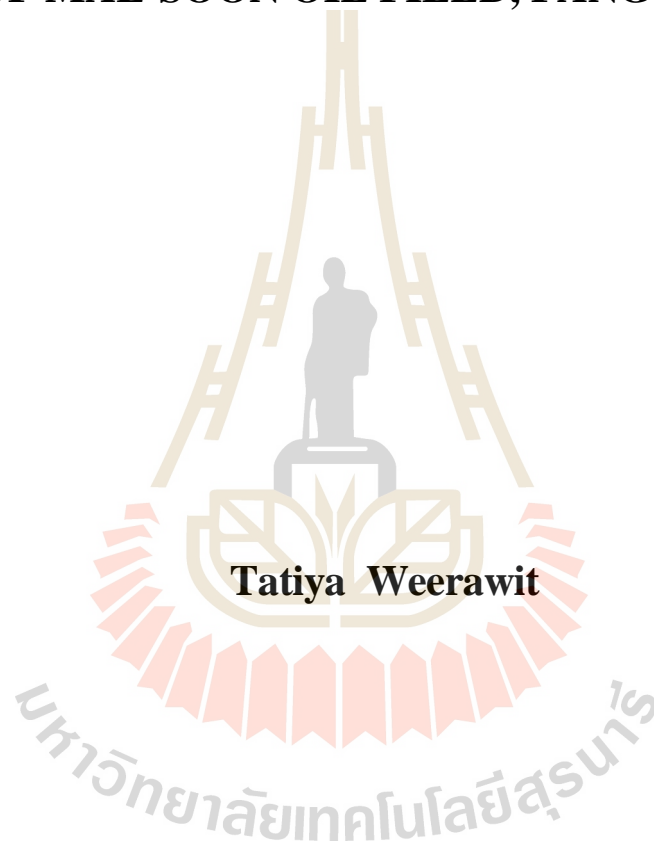


การพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการทำงานดัชนีชี้วัดการผลิต
โดยใช้วิธีการวิเคราะห์ถดถอยเชิงเส้นแบบพหุ
ในแหล่งน้ำมันแม่ฐาน แอ่งผาง



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีธรณี
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2556

**COMPUTER PROGRAM DEVELOPMENT FOR
PRODUCTIVITY INDEX PREDICTION USING
MULTIPLE LINEAR REGRESSION METHOD
OF MAE-SOON OIL FIELD, FANG BASIN**



**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Doctor of Master of Engineering in Geotechnology**

Suranaree University of Technology

Academic Year 2013

**COMPUTER PROGRAM DEVELOPMENT FOR
PRODUCTIVITY INDEX PREDICTION USING
MULTIPLE LINEAR REGRESSION METHOD
OF MAE-SOON OIL FIELD, FANG BASIN**

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

(Prof. Dr. Kittitep Fuenkajorn)

Chairperson

(Asst. Prof. Dr. Akkhapun Wannakomol)

Member (Thesis Advisor)

(Assoc. Prof. Kriangkrai Trisarn)

Member

(Prof. Dr. Sukit Limpijumngong)

Vice Rector for Academic Affairs
and Innovation

(Assoc. Prof. Flt. Lt. Dr. Kontorn Chamniprasart)

Dean of Institute of Engineering

ศศิธร วีระวิทย์ : การพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการทำงานดัชนีชี้วัดการผลิต
โดยใช้วิธีการวิเคราะห์ถดถอยเชิงเส้นแบบพหุในแหล่งน้ำมันแม่สูน แอ่งฝาง
(COMPUTER PROGRAM DEVELOPMENT FOR PRODUCTIVITY INDEX
PREDICTION USING MULTIPLE LINEAR REGRESSION METHOD OF MAE-
SOON OIL FIELD, FANG BASIN) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.อัมพรศักดิ์
วรรณโกมล, 198 หน้า.

วัตถุประสงค์ของการศึกษานี้ เพื่อพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการทำงานดัชนีชี้วัดการผลิตโดยใช้วิธีการวิเคราะห์ถดถอยเชิงเส้นแบบพหุเป็นพื้นฐานในการคำนวณ โปรแกรมดังกล่าวได้ถูกพัฒนาขึ้นบน MICROSOFT VISUAL BASIC และต่อไปจะถูกเรียกว่า PIP (PRODUCTIVITY INDEX PREDICTION) ข้อมูลที่ต้องการใช้และพื้นที่ศึกษาจะอ้างอิงมาจากแหล่งน้ำมันแม่สูนซึ่งตั้งอยู่ที่ แอ่งฝาง จังหวัดเชียงใหม่ ประเทศไทย สิ่งที่สำคัญในการศึกษานี้คือการหาตัวแปรที่เหมาะสมที่สุดสำหรับการสร้างสมการการทำงานดัชนีชี้วัดการผลิต เนื่องจากมีตัวแปรจำนวนมากที่มีผลต่อดัชนีชี้วัดการผลิต นอกจากนั้นดัชนีชี้วัดการผลิตและกลุ่มตัวแปรที่ส่งผลกระทบต่อเหล่านี้ มีความสัมพันธ์ที่ไม่สามารถคำนวณได้โดยตรงอีกด้วย การศึกษาครั้งนี้ได้ใช้ข้อมูลจากหลุมผลิตทั้งหมด 22 หลุมในแหล่งน้ำมันแม่สูน และ 10 ตัวแปรที่ส่งผลกระทบต่อถูกนำมาพิจารณาหลังจากการทดสอบกับโปรแกรม PIP หลาย ๆ ครั้ง ผลการศึกษาได้ชี้ให้เห็นว่า 8 ตัวแปรที่มีอิทธิพลต่อดัชนีชี้วัดการผลิต ประกอบด้วย ปริมาณน้ำที่ผลิตได้ ปริมาณน้ำมันดิบทั้งหมดที่ผลิตได้ ความดันใต้หลุม ค่าปัจจัยที่ใช้คำนวณปริมาตรภายในชั้นหิน ความหนืดของน้ำมันดิบ ความหนาแน่นของน้ำมันดิบ ระดับของเหลวภายในหลุมและความดันภายในแหล่งกักเก็บ ค่าดัชนีชี้วัดการผลิตที่คำนวณได้จากโปรแกรม PIP ได้ถูกนำมาเปรียบเทียบกับค่าทำนายดัชนีชี้วัดการผลิตที่คำนวณจากวิธีมาตรฐานทั่วไป ผลลัพธ์ที่ได้แสดงให้เห็นว่าค่าความคลาดเคลื่อนที่มากที่สุดเท่ากับ 15.08% ค่าความคลาดเคลื่อนที่น้อยที่สุดเท่ากับ 0.24% และค่าความคลาดเคลื่อนเฉลี่ยทั้งหมดเท่ากับ 2.27% ตามลำดับ ดังนั้นจากผลที่น่าพอใจดังกล่าวชี้ให้เห็นว่าโปรแกรมที่พัฒนาขึ้นนี้สามารถนำไปใช้ในการทำนายดัชนีชี้วัดการผลิตของพื้นที่ศึกษาหรือแหล่งน้ำมันอื่น ๆ ที่มีระบบปิโตรเลียมที่คล้ายคลึงกันได้ นอกจากนั้นค่าดัชนีชี้วัดการผลิตที่คาดการณ์ได้นี้ สามารถนำมาใช้สำหรับวางแผนการผลิตของพื้นที่ศึกษาต่อไปได้อีกด้วย

สาขาวิชา เทคโนโลยีธรณี

ปีการศึกษา 2556

ลายมือชื่อนักศึกษา _____

ลายมือชื่ออาจารย์ที่ปรึกษา _____

TATIYA WEERAWIT : COMPUTER PROGRAM DEVELOPMENT FOR
PRODUCTIVITY INDEX PREDICTION USING MULTIPLE LINEAR
REGRESSION METHOD OF MAE-SOON OIL FIELD, FANG BASIN.
THESIS ADVISOR : ASST. PROF. AKKHAPUN WANNAKOMOL, Ph.D.,
198 PP.

PRODUCTIVITY INDEX PREDICTION PROGRAM/MULTIPLE LINEAR
REGRESSION METHOD/MAE-SOON OIL FIELD/FANG BASIN

The objective of the study is to develop a computer program for productivity index (PI) calculation based on the multiple linear regression theory. The mentioned program was developed on the Microsoft Visual Basic software and hereafter is called PIP (Productivity index prediction). Required data and study area are referred from to the Mae-Soon oil field, located in the Fang basin, Chang Mai province, Thailand. The challenge in this study is to find the best fit parameters for PI prediction equation since there are numerous parameters affect in PI calculation. Moreover, PI and its affecting parameters have the relationship that cannot be calculated directly. This study collected the required data from a total of 22 production well data from Mae-Soon oil field. The 10 affecting parameters were considered after several screening tests with the PIP program. Results of the study indicated that 8 parameters have been identified as the most significant parameter, including water cut, gross oil production, bottom hole pressure, formation volume factor, oil viscosity, oil density, fluids level, and reservoir pressure. The estimated PI from PIP was then compared to those predicted by conventional method. Results indicated that the maximum error is

15.08% whilst the minimum is 0.24%, and the overall average error is 2.27%, respectively. Therefore, the satisfied results indicated that the developed PIP is capable to calculate a PI of the study area or any oil field where it has similar petroleum system. Moreover, estimated PI can be used for production planning of the study area further.



School of Geotechnlogy

Academic Year 2013

Student's Signature _____

Advisor's Signature _____

ACKNOWLEDGEMENTS

I wish to acknowledge the funding supported by Suranaree University of Technology (SUT).

In addition, the author expresses special gratitude and appreciation to Asst. Prof. Dr. Akkhapun Wannakomal, for his patience, guidance, knowledge and constant support during my graduate study.

The special appreciation is also extending to the teachers of the School of Geotechnology, including Prof. Dr. Kittitep Fuenkajorn, Assoc. Prof. Kriangkrai Trisarn, Dr. Chongpan Chonglakmani, Dr. Bantita Terakulsatit, Mr. Chatetha Chumkratoke and other persons, for knowledge and helpful suggestion to steer my research to the right path.

Finally, I most gratefully acknowledge my parents and everyone around me for all their help and support throughout the period of this research.

Tatiya Weerawit

TABLE OF CONTENTS

	Page
ABSTRACT (THAI)	I
ABSTRACT (ENGLISH).....	II
ACKNOWLEDGEMENTS	IV
TABLE OF CONTENTS.....	V
LIST OF TABLES	VIII
LIST OF FIGURES	IX
SYMBOLS AND ABBREVIATIONS.....	XI
CHAPTER	
I INTRODUCTION.....	1
1.1 Background and rationale.....	1
1.2 Research objectives.....	2
1.3 Research methodology	2
1.3.1 Literature review.....	2
1.3.2 Required data acquisition and preparation	2
1.3.3 A productivity index prediction program design.....	3
1.3.4 Program developing and testing	4
1.3.5 Conclusions and thesis writing	4
1.4 Scope and limitations	4
1.5 Thesis contents.....	4

TABLE OF CONTENTS (Continued)

	Page
II LITERATURE REVIEW	6
2.1 Productivity index	6
2.2 Multiple linear regression theory	11
2.2.1 T-Statistic method	11
2.2.2 Coefficient of determination & F-Statistic methods	13
2.3 Reviews on Fang oil field	16
2.3.1 The Fang oil field history	16
2.3.2 General geology	16
2.3.3 Source rock	17
2.3.4 Reservoir rock	19
2.4 Commercial program for multi linear regression	19
2.4.1 XLSTAT	19
2.4.2 GRetl	20
III METHODOLOGY	23
3.1 Program developing concepts	23
3.2 Data acquisition and preparation	24
3.2.1 Field production data	25
3.2.2 Laboratory analysis data	27
3.3 Program developing	36
3.3.1 Problem analysis	37
3.3.2 Flowchart	46

TABLE OF CONTENTS (Continued)

	Page
3.3.3 PIP Programming.....	46
3.3.4 Program system development.....	46
IV RESULTS AND DISCUSSIONS	51
4.1 Multiple linear regression equation.....	51
4.2 Relationship between productivity index and input parameters	58
V CONCLUSIONS AND RECOMMENDATIONS.....	65
5.1 Productivity index calculated from PIP	65
5.2 Recommendations.....	68
REFERENCES	69
APPENDICES	
APPENDIX A THE PARAMETERS INPUT DATA.....	73
APPENDIX B SOURCE CODE OF THE PIP PROGRAM.....	96
BIOGRAPHY	198

LIST OF TABLES

Table	Page
4.1 The best fit multiple linear regression equation calculated from PIP program of 22 selected production well of Mae-soon oil field, Fang basin (well MS-33 to MS-54).....	52
4.2 Relationship between PI and its related input parameters (well MS-33 to MS-66).....	59
5.1 Differentiated percentage between conventional calculated PI and PI calculated from the PIP program.....	66
5.2 List of significant related input parameters in the multiple linear regression equation for PI prediction of each selected production well of Mae-soon oil field, Fang basin.....	67

LIST OF FIGURES

Figure	Page
1.1	Flowchart showing steps of work of the study 3
2.1	Sequence analysis of multiple linear regression method 12
2.2	Fang basin geology and sub-basin division..... 18
2.3	XLSTAT/Modeling data/Regression commands 20
2.4	Linear regression dialog box 21
2.5	Gretl screenshot 22
3.1	Reference depth, pump depth, and fluid levels of the 22 selected production wells of Mae-soon oil field, Fang basin 26
3.2	Gross production rate, water cut, and net oil production rate of the 22 selected production wells of Mae-soon oil field, Fang basin 27
3.3	Oil specific gravity, oil viscosity, and oil formation volume factor of the 22 selected production wells of Mae-soon oil field, Fang basin 28
3.4	Reservoir pressure, bottom-hole pressure, and pressure drop in tubing of the 22 selected production wells of Mae-soon oil field, Fang basin 29
3.5	Bottom-hole pressure calculations 30
3.6	The relationship between pressure and oil density 33
3.7	The relationship between pressure and oil viscosity..... 34

LIST OF FIGURES (Continued)

Figure	Page
3.8 The relationship between pressure and formation volume factor.....	34
3.9 Parameters in radius flow calculation	36
3.10 Parameter view page of the PIP program.....	39
3.11 Data view page of the PIP program	40
3.12 Edit page of data view in the PIP program.....	41
3.13 Report data of data view page of the PIP program	42
3.14 Calculation view page of the PIP program.....	43
3.15 Result view page of the PIP program.....	44
3.16 Handbook view page of the PIP program showing program handling calculation and statistic theories.....	45
3.17 The main PIP program flowchart.....	47
3.18 The PIP program sub-flowchart part 1.....	48
3.19 The PIP program sub-flowchart part 2.....	49

SYMBOLS AND ABBREVIATIONS

BHP	=	Bottom-hole pressure (psi)
Bo	=	Formation volume factor (STB/bbl)
bbl/d	=	Barrel per day
D	=	Oil density (g/cc)
FL	=	Fluids levels (ft)
GOP	=	Gross oil production (bbl/d)
h	=	Reservoir thickness (ft)
J	=	Productivity index (bbl/d/psi)
k	=	Permeability (md)
NOP	=	Net oil production (bbl/d)
PD	=	Pump depth (ft)
PDT	=	Pressure drop in tubing (psi)
PI	=	Productivity index, STB/d/psi
PR	=	Reservoir Pressure (psi)
Pe	=	Pressure at external boundary
Pw	=	Bottom-hole producing pressure, BHP (psi)
RD	=	Reference depth (ft)
q	=	Gross production rate (bbl/d)
r _e	=	Boundary radius (ft)
r _w	=	Well bore radius (ft)

SYMBOLS AND ABBREVIATIONS (Continued)

SSE	=	Sum of squares, error
SST	=	Sum of squares, total
SSR	=	Sum of squares, regression
V	=	Oil viscosity (cp)
WC	=	Water cut (bbl/d)
X	=	Value of k^{th} predictor in year i
Y	=	Predicted in year i
β_0	=	Regression constant
β_k	=	Coefficient on the k^{th} predictor
μ	=	Oil viscosity (cp)
ε	=	Error term
k	=	Total number of regression

CHAPTER I

INTRODUCTION

1.1 Background and rationale

Productivity Index (PI) describes the relationship of production rate with bottom hole flowing pressure (P_{wf}). Its estimating accuracy is essential for production planning and well completion design in a new well. In general, the conventional methods are the best but sometimes these conventional methods cannot apply when the bottom hole flowing pressure is not known, e.g. in a new drilling well. Moreover, these methods may give high erroneous results when some required parameters are not exactly known. To minimize an inaccurate in productivity index determination, the multiple linear regression method might be applied.

The multiple linear regressions are a model of a linear relationship between a dependent variable and one or more independent variables. This method is based on least squares theory. The model is fit such that the sum-of-squares of differences of observed and predicted values is minimized. This study represents a developed computer program on Microsoft Visual Basic software for the productivity index prediction of Mae-Soon oil field, Fang basin. This field had been chosen for study because it has sufficient and available required data. Consequently, the productivity index from the developed program was compared to PI which were calculated by a conventional calculation method to see any differences.

1.2 Research objectives

The objective of the study is to develop a computer program on Microsoft Visual Basic software for productivity index calculation based on multiple linear regression theory.

1.3 Research methodology

The research methodology of the study is depicted in Figure 1, including literature review, required data acquisition and preparation, a productivity index prediction program design, program developing and testing, results conclusions and discussions, and thesis writing.

1.3.1 Literature review

Literature review had been carried out in order to study the previous researches on other subjects. The sources of information were from internet, text books, journals and conference papers. A summary of the literature review which are given in the thesis are as follows;

1.3.1.1 Productivity index prediction method

1.3.1.2 Multiple linear regression theory

1.3.1.3 Geology and petroleum system of Fang basin

1.3.1.4 Other similar commercial program

1.3.2 Required data acquisition and preparation

Required data for productivity index calculation had been gathered from Mae-Soon oil field, Fang basin, with the courtesy of the Department of Energy Defense (DED). The input parameters were hierarchically characterized into several

groups using various criteria, e.g. geology, production history, reservoir and well conditions, petroleum engineering requirements, design constraints, and project goals. All collected data had been prepared and arranged in suitable format for the program input process.

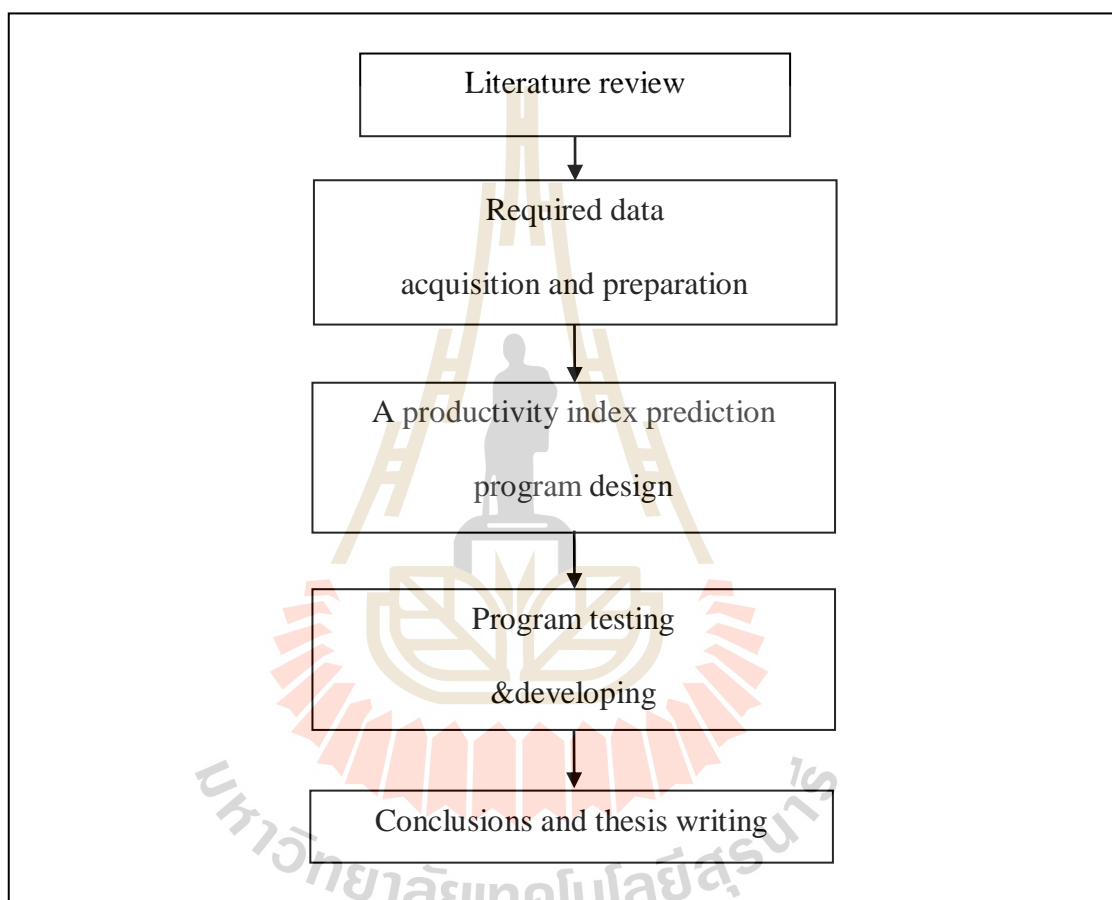


Figure 1.1 Flowchart showing steps of work of the study

1.3.3 A productivity index prediction program design

In this step the Visual Basic software had been used for developing a computer program for the productivity index prediction based on the multiple linear regression criterions. The productivity index prediction program had been designed

as a user-friendly program that includes necessary command button, such as “check”, “edit”, “save”, and simple format as same as any commercial programs.

1.3.4 Program developing and testing

In term of technical considerations, production, reservoir and well condition data had been collected, analyzed, and input to this computer program. Productivity index prediction will determine the most suitable equation for production. The developed program had been developed and tested in this step to get more reliable and accuracy for productivity index calculation.

1.3.5 Conclusions and thesis writing

Consequently, all research activities, methods, and results had been documented and complied in the thesis.

1.4 Scope and limitations

The scope and limitations of the study are as follows;

1.4.1 The productivity index prediction program had been developed on Microsoft Visual Basic software.

1.4.2 The required data were collected only from Mae-Soon oil field, Fang basin.

1.5 Thesis contents

This thesis is divided into five chapters. The first chapter includes background and rationale, research objectives, research methodology, scope and limitations, and expected results. Chapter II presents results of the literature review which are necessary for improving the productivity index prediction program. Chapter III

presents data acquisition and data preparation, describes program developing and program testing steps. Chapter IV presents results and discussions of the study. Chapter V presents conclusions and recommendations for future studies, respectively. Appendix A presents the collected essential data for thesis study. Appendix B depicts source code of the developed program.



CHAPTER II

LITERATURE REVIEW

This chapter summarizes the results of literature review which are helpful for the productivity index prediction program developing. The topics reviewed here include productivity index description, multiple linear regression theory, geology and petroleum system of Fang basin, and other similar commercial program for multiple linear regression calculation.

2.1 Productivity index (PI)

Productivity Index is a method of measuring the relative ability of wells to produce without open flow that shows considerable promise. This method was first suggested by Moore (1930) and has since been strongly advocated by many engineers and producers. The method involves measurement of the static, or shut-in, bottom-hole pressure while the well is flowing. The productivity index is defined as the barrels of oil per day per pound differential between static and flowing bottom-hole pressure. In determining the index of a well, the well is produced at low rates of production, thus eliminating gas wastage, the possibility of water conning, and unnecessary oversize expensive equipment required for open flow potential. The productivity index has been widely discussed, and engineers are in agreement as to its many advantages, but there has not been sufficient field testing to determine its practicability.

Theoretically, all that is necessary to determine the index is to measure the static bottom-hole pressure and the produce the well, measure the production and the flowing bottom-hole pressure, and calculate the productivity index. Actually, it is necessary to measure the bottom-hole pressure at various rates of production in order to eliminate the possibility of errors that may be incurred in determining only one value when plotting the indicates thus obtained, in many instances a different value is obtained for the index at each rate of production. This naturally gives rise to the question as to whether or not the index is a constant with rate of production, or whether we should expect to find a different value for each rate. In order to be able to make practical application of the index in the allocation of production, it should be a constant with rate of production, since if a different value is obtained at each rate it would be extremely difficult to determine at which rate the value of the index should be taken. In order that the index be a constant, the pressure differential must be a straight-line function of the rate production according to the equation $y = mx + c$ with c equal to zero, y equal to the rate of production and x is then equal to the pressure differential. The slope of the line m is then equal to the productivity index, and is a constant.

The productivity index is usually measured on a procuring well by dividing the production rate by the pressure drawdown as follows.

$$PI = \frac{q_o}{(P_e - P_w)} \quad (2.1)$$

Where PI = productivity index, STB/d/psi
 P_e = pressure at external boundary, psi
 P_w = bottom-hole producing pressure (BHP), psi

Using Darcy's law for steady-state radial flow gives

$$PI = \frac{0.00708kh}{\mu_o B_o \ln(r_e / r_w)} \quad (2.2)$$

Where $(P_e - P_w)$ has cancelled out

Equation 2.1 and 2.2 show that the productivity index can be measured in the field by equation 2.1, and is related to reservoir and fluid properties through equation 2.2. After actual production rate and pressure drawdown are measured for computing the productivity index with equation 2.1, equation 2.2 can be used to estimate an unknown reservoir parameter, such as permeability (k).

In some wells the productivity index may remain constant over a wide range of production rates, which is useful for predicting future production rates at various pressure drawdowns. However, in many wells the productivity index declines at higher production rates or at later times as the reservoir pressure declines. Several causes of declination productivity indexes are listed as follows:

- a. Non-laminar, or turbulent, flow at high rates of production,
- b. Decline in relative permeability to oil (k_{ro}) due to increasing gas saturation as the pressure drops below bubble point, especially near the well bore,
- c. Increase in oil viscosity at pressure below bubble point, and
- d. Decline in absolute permeability due to formation compaction as pressure decline.

Krirk (1972) presented a method of increasing the productivity index of producing wells. This invention relates to improving the injectivity index of injection

wells. About 1-500 gallons of micellar dispersion per vertical foot of hydrocarbon was injected into formation followed by about 1-500 gallons of a mobility buffer per vertical foot of the formation, and thereafter injecting a displacing fluid to displace the micelle dispersion to a radius of at least about 7.5 feet from the well bore and injected a fluid to increase permeability. The displacing fluid was followed by a fluid to increase the relative permeability to the flow of fluids characteristic of oil-wet reservoir formation.

Junkui (1983) reported an effective method used to identify the type of decline curve for prediction of the reservoir performance. The effectiveness of this method was illustrated by an actual field data. A computer program was presented in this method, was simple, quick, and a satisfactory result.

Xudong (1986) gave a prediction of the future production and cumulative recoverable reserves of an oil field by means of a model (Weng Cycle) and a prediction of the cumulative water-oil ratio of the same oil field by another model (Logistic Cycle). The results of calculations from the data of various oil fields both in China and abroad indicate that the predictions based on two models are acceptable, and some helpful ideas can be extracted.

Spath and Mach (1999) presented a method to estimate the productivity index and skin factor by pump runtime and downtime data. The proposed estimation was an economical and accurate method for monitoring the behavior of a well reservoir system during runtime model database points were generated to simulate runtime and downtime during production of a well. The model was computed assuming initial values of productivity index and skin. The actual run time and downtime is constantly and automatically recorded. The model was then compared, in a least

squares sense, with the actual data runtime data. The value of the productivity index and skin were updated until the model matches the actual data.

Ibelegbu (2004) studied the effect of reservoir and well parameter on the productivity index of a horizontal well. He analyzed the effect of skin due to partial completion on the productivity index using the three partial well completion configurations. The productivity index increased with increasing lateral length and isotropy. This is explained by the fact that a large portion of the reservoir has been contracted and the pressure drop along the well bore is reduced. The horizontal wells are more suitable for reservoir with high vertical permeability as this increases horizontal well productivity index. Considering productivity index ration between horizontal and vertical direction (J_h/J_v) for reservoir thickness, a thin reservoir produces fluid more than a thick reservoir. This is as a result of more gain in contact area, which can be achieved in a thin reservoir than in thick reservoir. The factors (well length, permeability, reservoir thickness, drainage area, penetration ration etc.) that affect the pressure drop between the reservoir and the well bore are also affected productivity index in horizontal wells. Type and efficiency of completion job in well and skin also affect productivity index too. The wells that are perforated at equal intervals along the well bore experience little or no skin effect, thereby enhancing productivity

Yanfeng and Xiaodong (2008) presented that the production index predicting method has good effect on the new wells in new blocks. Since the old block has been produced for a long time, the oil characteristics and other parameters of the block change greatly. As a result, the use of conventional productivity index predicting method is limited and the predicting results are not so accurate. Based on the pseudo-

steady state production capacity equation of the circular closed reservoir, combined with the production data of oilfields, a new predicting method is developed. The result based on the new method has been compared with that of the conventional one and it shows that the new method is more accurate.

2.2 Multiple linear regression theory

Regression analysis is a statistical methodology that utilizes the relation between two or more quantitative variables so that a response or outcome variable can be predicted from the other.

Sequence analysis of multiple linear regression method is showed in Figure 2.1 and listed as follows:

2.2.1 T-Statistic method

T-Statistic method is a statistical processing used to prove the relationship between explanatory (X) and the dependent variable (Y). The first model is estimated by the least squares criterion which yields parameter estimates such that the sum of squares of errors is minimized. The coefficients are considered as follows;

$$H_0 : \beta_i = 0 \quad (2.4)$$

$$H_a : \beta_i \neq 0 \quad (2.5)$$

H_0 is the coefficient relating the explanatory (x) variable to the 0 dependent (y) variable is zero. In other words, there is no relationship between the explanatory variable and the dependent variable (equation 2.4). The alternative hypothesis H_a is one that the coefficient relating the x variable to the y variable is not

equal to zero. In other words, there is some kind of relationship between x and y (equation 2.5).

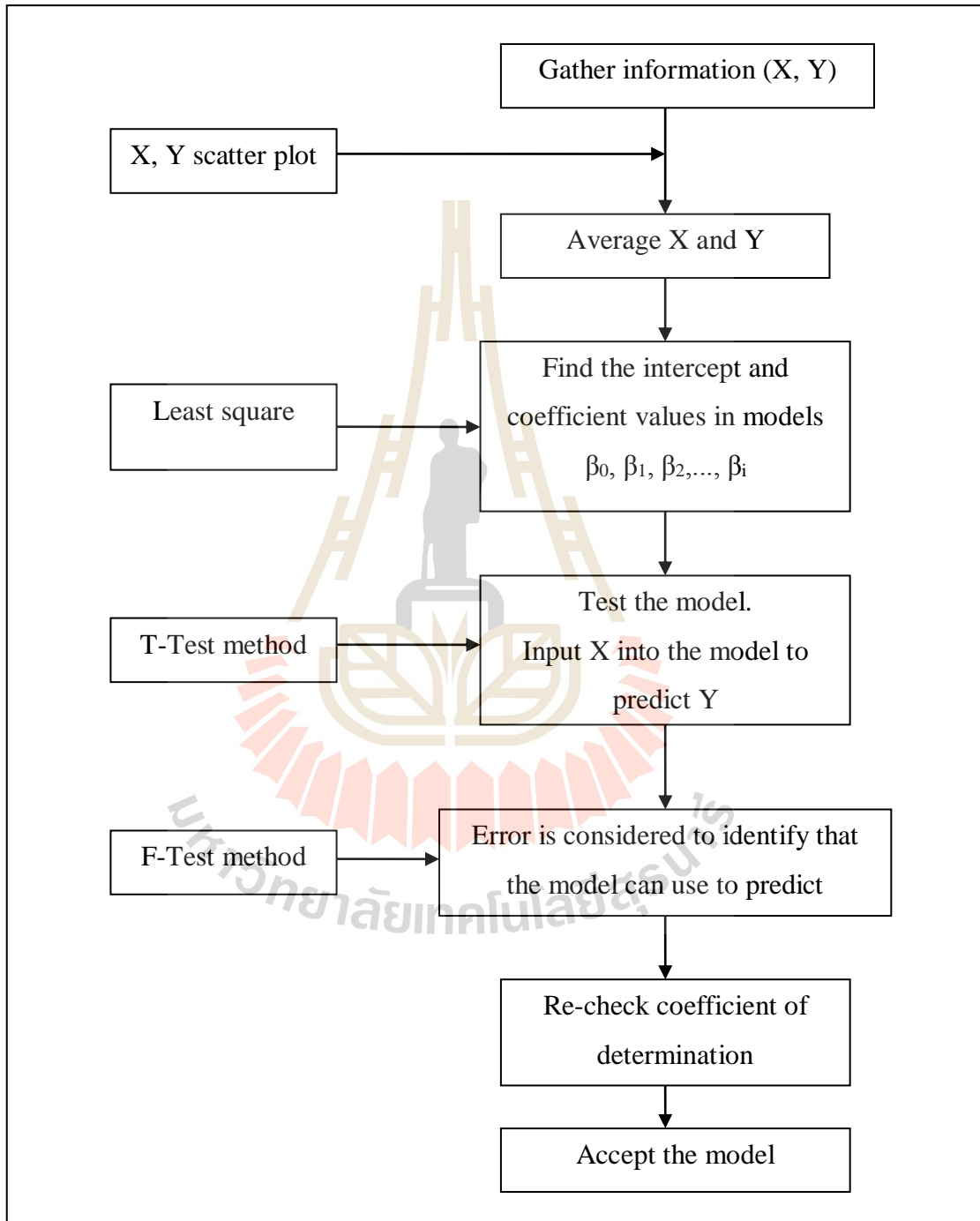


Figure 2.1 Sequence analysis of multiple linear regression method

2.2.2 Coefficient of determination & F-Statistic methods

Several regression statistics are computed as functions of the sums-of-squares terms (SS), including

$$SSE = \sum_{i=1}^n \hat{\epsilon}_i^2 \quad \text{sum of squares, error} \quad (2.6)$$

$$SST = \sum_{i=1}^n (y_i - \bar{y})^2 \quad \text{sum of squares, total} \quad (2.7)$$

$$SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \quad \text{sum of squares, regression} \quad (2.8)$$

Where

- n = sample size (number of observation in calibration period)
- y_i = observed value of predict in year i
- \hat{y}_i = predicted value of predict in year i
- \bar{y} = average predict in year i
- $\hat{\epsilon}_i$ = error term ($y_i - \hat{y}_i$)

Partitioning of variation, the regression equation is estimated such that the total sum-of-squares can be partitioned into components due to regression and residuals:

$$SST = SSR + SSE \quad (2.9)$$

Coefficient of determination, the explanatory power of the regression is summarized by its “R-squared, R^2 ” value, which is computed from the sums-of-squares term. It is often described as the proportion of variance described. It is important to keep in mind that a high R^2 does not imply causation. The relative sizes of the sums-of-squares terms indicate how “good” the regression is in term of fitting the calibration data. If the regression is “perfect”, all residuals are zero, SSE is zero, and R^2 is 1. If the regression is a total failure, the sum-of-squares of residuals equals the total sum-of-squares, no variance is accounted for by regression, and R^2 is zero.

Xianghua (1997) presented the simple and practical method to solve the Weng cycle parameters, of which need only two times of a simple linear regression to get the results. This method can overcome the disadvantage of the complexities in solving the Weng cycle parameters and bring the convenience to the onsite hydrocarbon reservoir engineers.

He, Chen, and Ping (2009) used the diagnosis methods for different types of production rate decline. The production rate decline theory by applying the linear relationship between output and time. The application result proved that this approach is simple and feasible to determine the production rate decline type.

Deng (2009) predicted the dynamic output of an oilfield, some relevant factors with oil output were chosen on the basis of actual manufacture experience. The forecast model of linear regression analysis for an oilfield was built according to the important factors of influencing oilfield output which was obtained with the synthetic regression analysis. The improved regression model for predicting annual output of an oilfield was built up by analyzing the statistic information in the solving process of regression parameters. Meanwhile, the two forecast models were used to

predict the output of an oilfield. As a result, the improved multiple linear regression models were better than the multiple linear regression models in the forecast accuracy of the oilfield output.

Mustafar and Radzuan (2011) presented a method for predicting the oil field output. The variables were analyzed to the relationships those were unsystematic and unexpected. Multiple linear regression method reduced several input variables in the models and considered factors of dynamic system while others could not. Eight parameters were identified to predict the oilfield output that can also be obtained by using MATLAB simulation. From the result and discussion it showed that the erroneous percentage of predicted value comparing to the actual output was 6.538 percent for eight parameters and 4.575 percent for four parameters after such a screening on multiple linear regression method had been made. The four parameters were the startup number of wells, the recovery percent of previous year, the injected water volume last year, and the oil moisture content of previous year. This method gave a satisfy results and could be implemented to forecast the oil field output.

Chitsiripanich and Pummarapanthu (2011) presented the estimation of productivity index which was a significant factor influencing reservoir performance. The analysis was divided into two parts; the analysis on the new wells and the existing wells. A total of 192 field data set was collected from Sirikit oil field. Twenty three parameters were considered and found that only 6 parameters which were the effecting factor. They were liquid rate, total gas-oil ratio, water cut, flowing tubing head pressure, perforation length, and reservoir pressure. In their study the multiple linear regression models were constructed for productivity index (PI) prediction. The estimated PI calculated from multiple linear regression models was

compared to those predicted by conventional method. From the result and discussion it showed that the multiple linear regression method failed for the new wells PI prediction due to production data were unavailable. Conversely, this method could be useful for the productivity index estimation in the existing wells part.

2.3 Reviews on Fang oil field

2.3.1 The Fang oil field history

The Fang oil field is located about 850 km north of Bangkok in north plains of Thailand, which covered an area about 600 km² in Mae Eye, Fang, and Chai Pra Karn district, Chang Mai province. The Fang oil field, Thailand's first significant oil found, was discovered oil seep in late 1949 by the Department of Energy, Thailand, with its second exploration well. The Department of Energy took only two years to design and install the production station, and organize an unconventional evacuation system (road tanker and railways) before oil came on stream in 1951. A series of facility upgrading kept pace with the production buildup to a plateau about 1,000 bbl/d.

2.3.2 General geology

The field is a NNE-SSW trending, small intracratonic basin (Figure 2.2). It formed in the Early Tertiary and evolved in the Middle Tertiary in a transtensional regime followed by Pliocene to Pleistocene compression in a transtensional/transpressional left-lateral tectonic system by Zollner and Moller, (1996). The twenty depositional environments in the Tertiary were fluvial-lacustrine and changed to fluvial and alluvial in the Quaternary by Settakul (1985). It is on the western margin of the Sokhothai Fold belt, which comprises Paleozoic and Triassic

strata and volcanic rocks that accumulated on the eastern margin of the Shan-Thai Craton prior to the Indosinian Orogeny. This fold belt is complex and deformed by granitic intrusions during the collision of the Indochina and Shan-Thai cratons by Bunopas and Vella (1983).

The Pre-Tertiary basement rocks consist of sedimentary, metamorphic and igneous rocks. On the western side of the basin, the rocks are Cambrian-Permian age, and include Carboniferous granite. On the eastern side of the basin, the rocks are Silurian-Devonian and Jurassic, with Triassic granite. The Tertiary rocks of the Fang basin are conglomerate, sandstone, claystone and shale. The Quaternary deposits are silt, clay, sand and gravel and occur as stream channels, terrace deposits and alluvial fans. These sediments are covered by recent soil and lateritic sand.

2.3.3 Source rock

Zollner and Moller (1996) described the hydrocarbon system of Fang basin and suggested that the source rocks were deposited in a lacustrine environment. Potential source rocks include bituminous shale and lignite of the Lower Mae Sod formation. Source rocks were within the oil window during this time. Top of the oil window is interpreted to be located at a depth of approximately 2,750 ft (temperature gradient is 7.2 Co/100 m). Migration pathways are interpreted to have been updip and along faults.

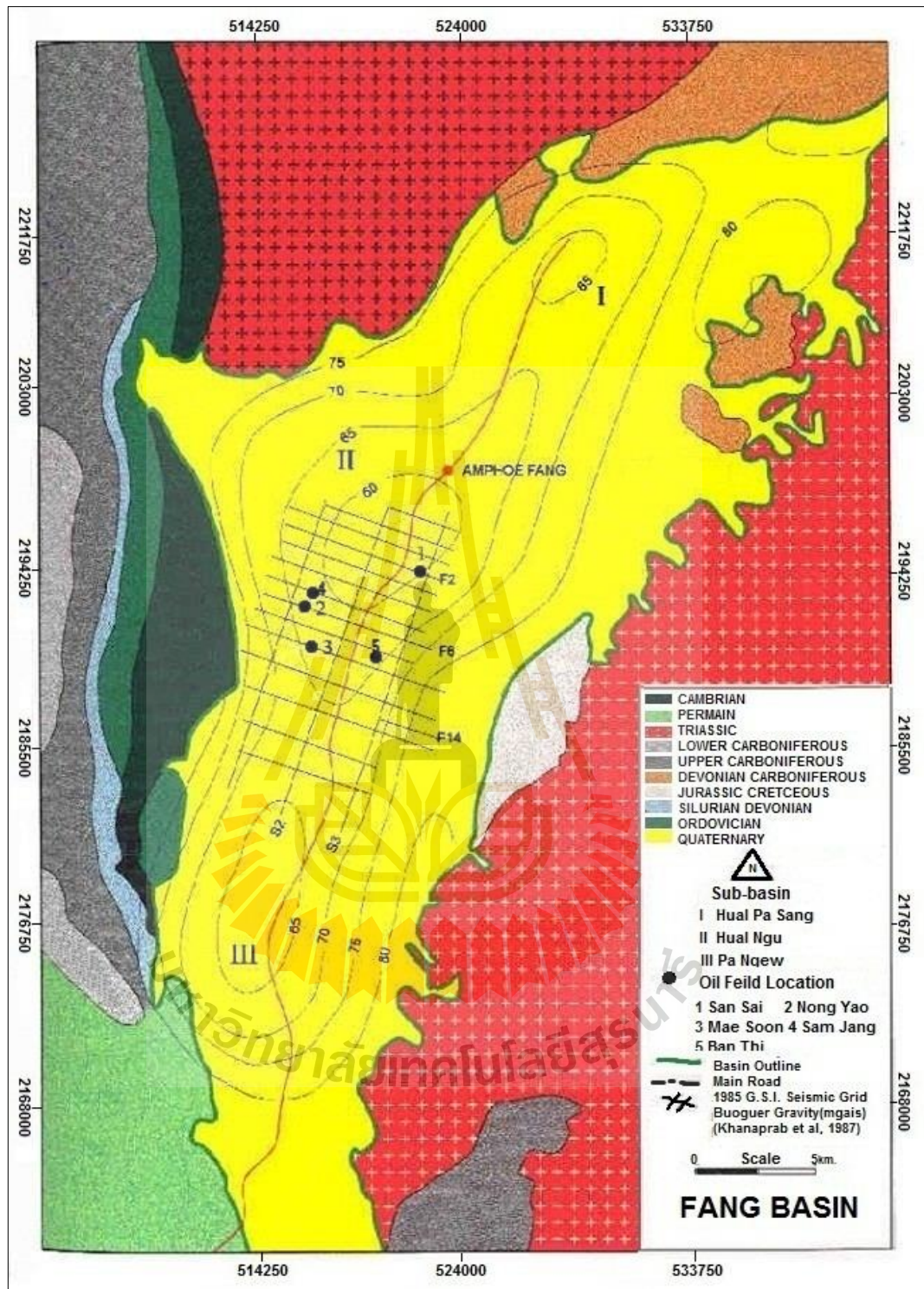


Figure 2.2 Fang basin geology and sub-basin division (modified after Petro-Canada Resources, 1988)

2.3.4 Reservoir rock

Reservoir rocks in the Mae-Soon and the San Sai structures are infill sands of the Upper Mae Sod formation, whereas in the Pong Nok area they are westward prograding sandstone that pinches out towards the east. Mostly intra-formation shale layers are interpreted to act as seals. The generation of structures was related to the compression during the Pliocene to Pleistocene.

2.4 Commercial program for multi linear regression

There are some commercial programs for analyzing multi linear regression. This study introduces two of them, XLSTAT and Gretl. XLSTAT is the leading data analysis and statistical solution for Microsoft Excel, whereas Gretl is an open-source statistical package, mainly for econometrics. However, there are other similar commercial programs that do not introduce in this section, e.g. Minitab 15, Wessa 2009, and X2.

2.4.1 XLSTAT

The XLSTAT statistical analysis add-in offers a wide variety of functions to enhance the analytical capabilities of Excel. XLSTAT's statistical analysis program is compatible with all Excel versions from version 97 to version 2011 (except 2008 for Mac), and is compatible with the Windows 9x till Windows 7 systems, as well as with the Power PC and Intel based Mac systems. Today, XLSTAT is used by over 30,000 customers, businesses and universities, in over 100 countries throughout the world. Annual licenses start as low as \$50 USD for students, \$165 USD for academics and \$275 USD for other users. XLSTAT can be purchased

online. Both electronic and CD-Rom versions are available. Example of XLSTAT features are showed in Figure 2.3 and Figure 2.4, respectively.

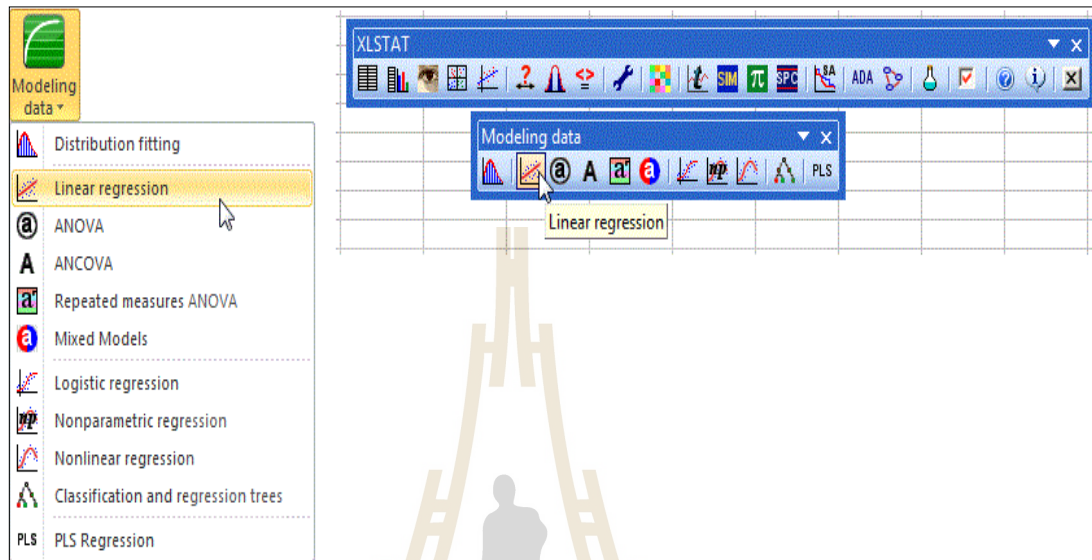


Figure 2.3 XLSTAT/Modeling data/Regression commands (after <http://www.xlstat.com/en/learning-center/tutorials/multipler-linear-regression-in-xlastat.html>)

2.4.2 GRetl

The name is an acronym for Gnu Regression, Econometrics and Time-series Library. It has a graphical user interface and can be used together with X-12-ARIMA, TRAMO/SEATS, R, Octave, and Ox. It is written in C, uses GTK as widget toolkit for creating its GUI, and uses gnu plot for generating graphs. As a complement to the GUI, it also has a command line interface. Some examples of GRetl screenshot are depicted in Figure 2.5.

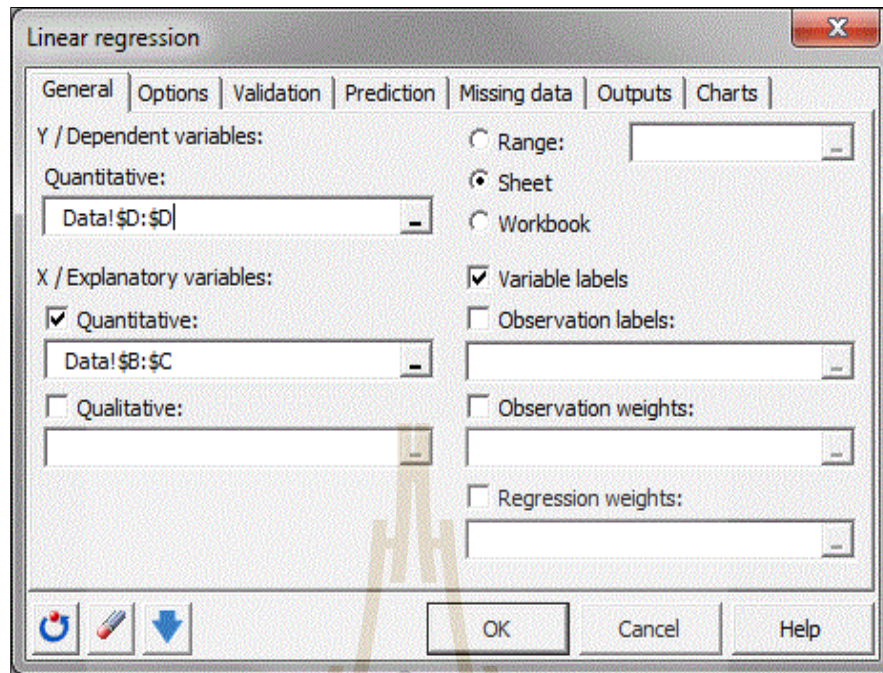


Figure 2.4 Linear regression dialog box of XLSTAT (<http://www.xlstat.com/en/Learning-Center/tutorials/multiple-linear-regression-in-xlstat.html>)

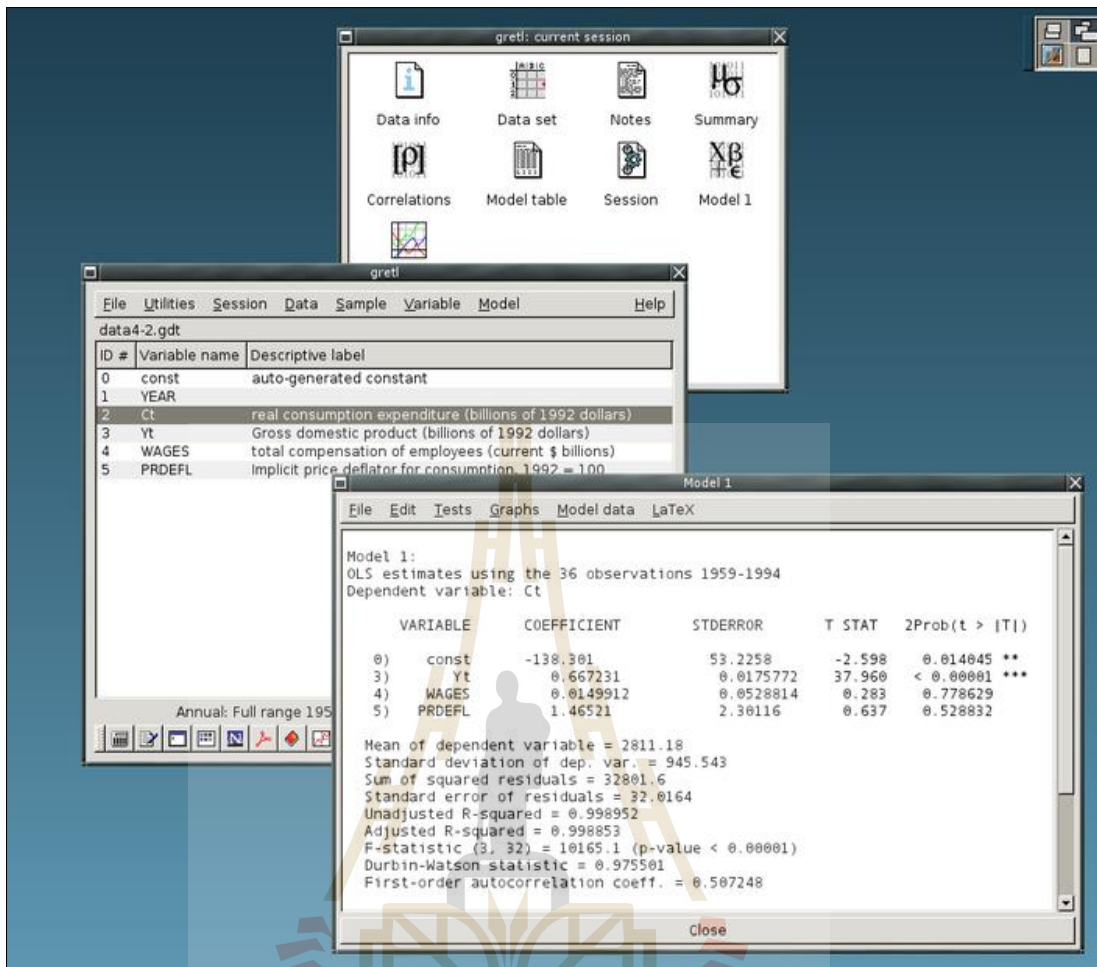


Figure 2.5 Gretl screenshot (after http://en.wikipedia.org/wiki/File:Gretl_screenshot.png)

CHAPTER III

METHODOLOGY

3.1 Program developing concepts

In this study, the computer program development for productivity index prediction using multiple linear regression method was separated into two main parts. The first part was for data recording and data modeling. Another part was for productivity index calculation and planning.

Linear regression method had been applied for productivity index calculation on a computer program which was developed on the Microsoft Visual Basic software. Productivity index predicted from the developed program then had been compared to the production index which was calculated from the conventional method to see the differences. Generally, the required variables used in this study are numerical value in the oil field unit.

The program predicts the values of a dependent variable, Y , given a set of n explanatory variables (x_1, x_2, \dots, x_n) . The explanatory variables, which were called the related parameters, were analyzed from the Darcy inflow equation and vertical lift correlations. Computer program development steps were as follows;

1) The input data set as mentioned in section 2.1 had been collected to develop multiple linear regression models and expressed on the basic form of multiple linear regression equation as equation (3.1);

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \quad (3.1)$$

Where	β_0	=	regression constant
	β_k	=	coefficient on the k^{th} predictor
	k	=	total number of regression
	ε	=	error term
	y	=	predictand in year i
	x	=	value of k^{th} predictor in year i

2) Multiple linear regression method was then applied again to find the less significant or negligible variables which could be rejected by applying T-statistic and F-test method. Therefore, the new mathematical model is rewritten.

3) Then resulted model from step 1 and step 2 had been compared to each other to see the erroneous percentage difference.

4) The best fit model resulted from step 2 and step 3 was then used for productivity index and its corresponding parameters prediction. In this step, user can manually adjust some parameters to test the output productivity index which is corresponded to the adjusted parameters.

3.2 Data acquisition and preparation

This topic describes the required data acquisition and preparation for using as input parameter of the developing computer program. The program was created to indicate the relationship between productivity index and its related parameters. The tested data here were obtained and courtesy from the Northern Petroleum Development Center, Defense Energy Department (DED), Fang district, Chang Mai province.

According to the data confidential limitation, the correlation between productivity index and others input parameters of Fang basin in this study were determined only for Mae-soon production wells. In this study twenty two production wells were selected, including FA-MS-25-33, FA-MS-27-45, FA-MS-28-47, FA-MS-28-50, FA-MS-30-54, FA-MS-32-56, FA-MS-32-57, FA-MS-34-60, FA-MS-35-62, FA-MS-35-64, FA-MS-36-66, FA-MS-47-71, FA-MS-48-73, FA-MS-48-74, FA-MS-49-77, FA-MS-51-78, FA-MS-51-79, FA-MS-52-80, FA-MS-53-81, FA-MS-53-82, FA-MS-53-83, and FA-MS-54-86. Well position, completion equipment data, casing data, sucker rod number, tubular and surface equipment data of the twenty two selected wells were collected and then divided into 2 data parts as field production data and laboratory analysis data.

3.2.1 Field production data

Field production data, including reference depth, pump depth, fluid level, gross production rate, water cut, and net oil production rate, had been collected back to the last four years (from the third quarter of 2009 to the third quarter of 2012). These required input parameters are summarized and presented in Figure 3.1 to Figure 3.2.

Fluids level, gross oil production, net oil production and water cut were routine recorded. These data were grouped and averaged each 3 months. Fluid level was range between 1931 and 2042 ft (average 2002.34 ft). Gross oil production was range between 49.69 and 56.12 bbl/day (average 53.76 bbl/d). Net oil production was range between 4.55 and 12.95 bbl/day (average 10.42 bbl/d). Water cut was range between 38.34 and 50.96 bbl/day (average 43.33 bbl/d). Others production well data are presented in Appendix A.

Pump depth and reference depth of every well in this oil field were the same and was recorded in the database of Fang oil field after well drilling operation had been completed. Reference depth was picked from the average perforation depth.

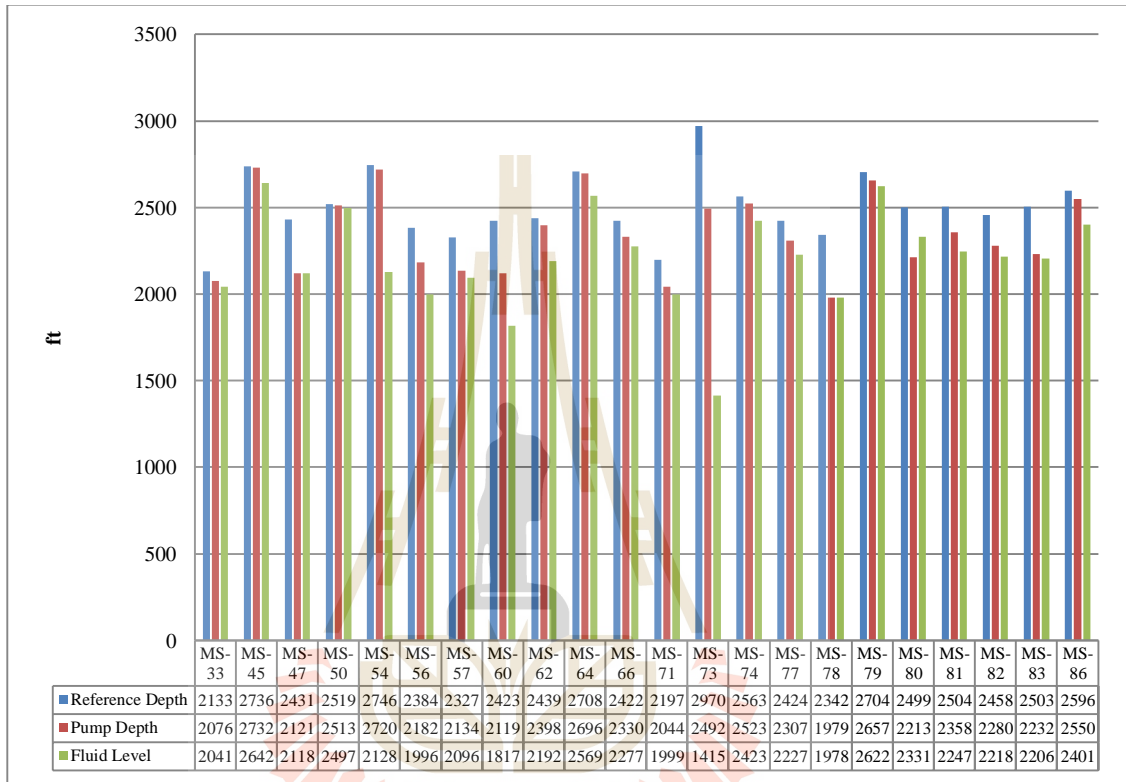


Figure 3.1 Reference depth, pump depth, and fluid levels of the 22 selected production wells of Mae-soon oil field, Fang basin

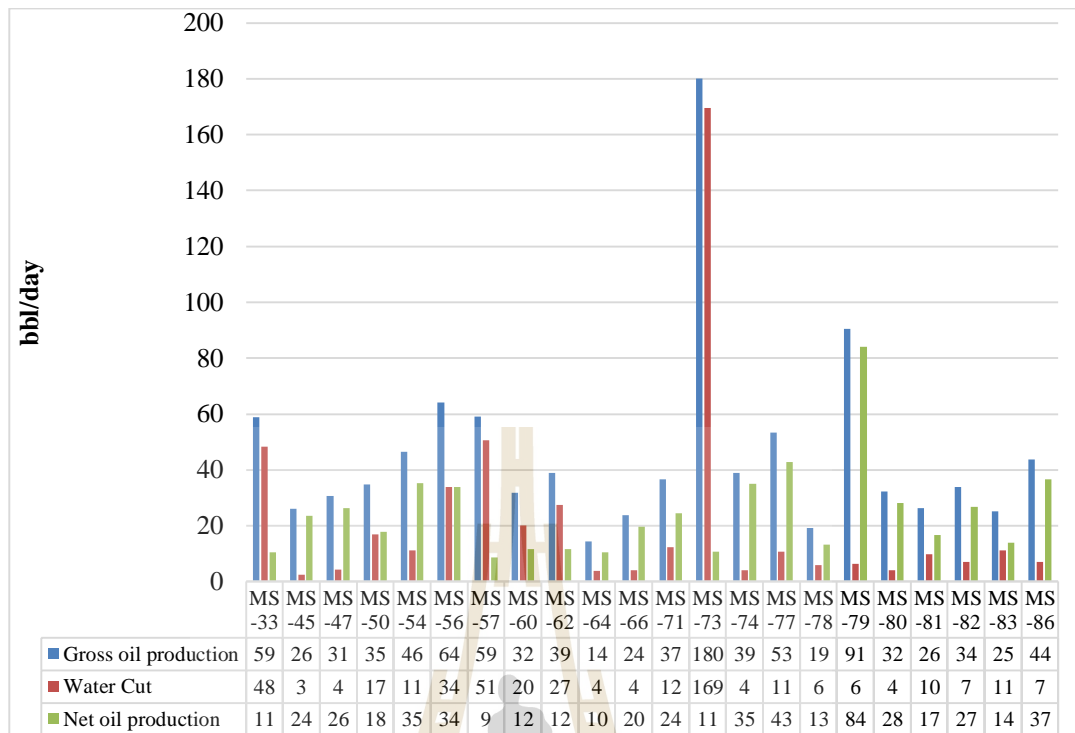


Figure 3.2 Gross production rate, water cut, and net oil production rate of the 22 selected production wells of Mae-soon oil field, Fang basin

3.1.2 Laboratory analysis data

Required laboratory analysis data, including reservoir pressure, bottom-hole pressure, tubing pressure drop, oil specific gravity, oil viscosity, and oil formation volume factor of the 22 selected production wells of Mae-soon oil field, Fang basin, that had been collected during the third quarter of 2009 and the third quarter of 2012, which were grouped and averaged each 3 months, are presented in Figure 3.3 and Figure 3.4, respectively.

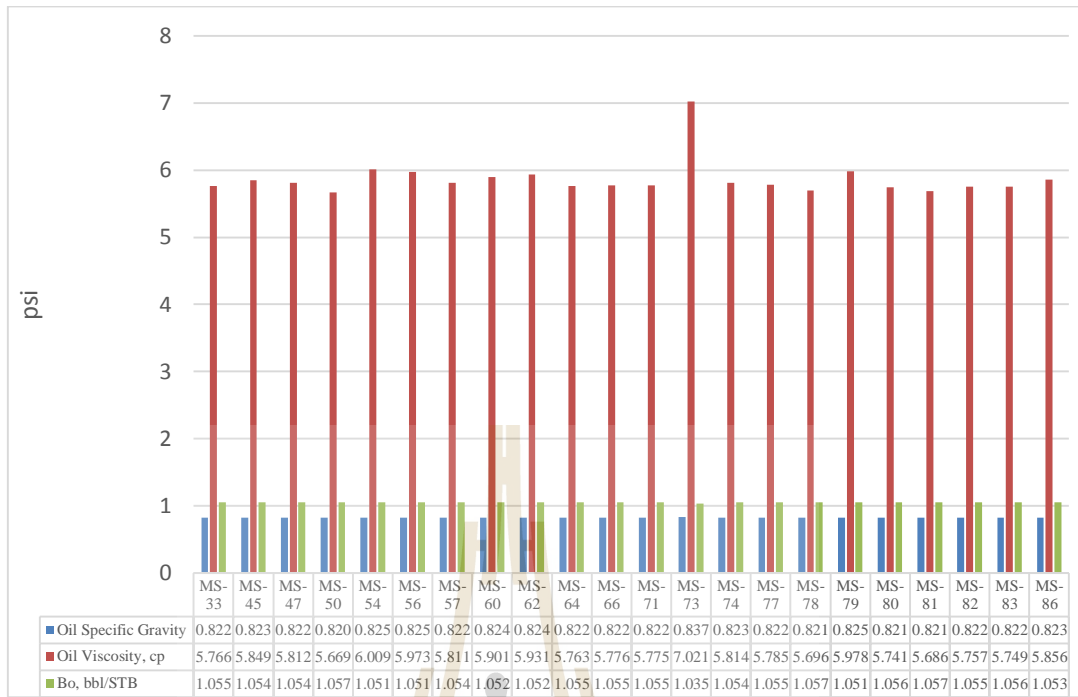
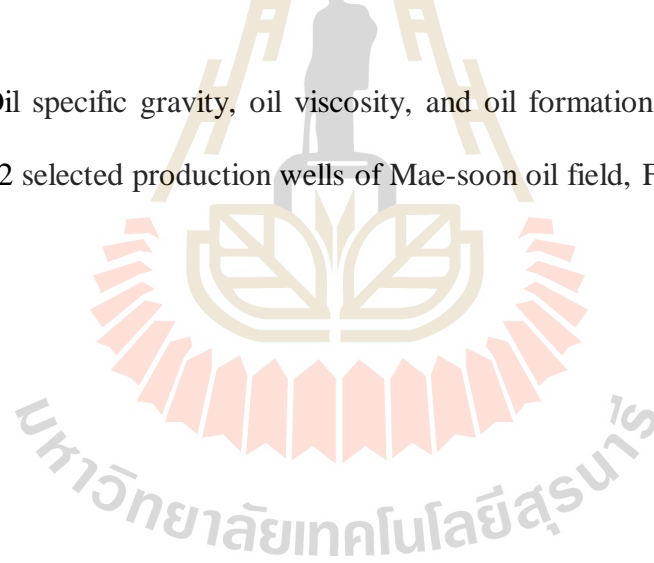


Figure 3.3 Oil specific gravity, oil viscosity, and oil formation volume factor of the 22 selected production wells of Mae-soon oil field, Fang basin



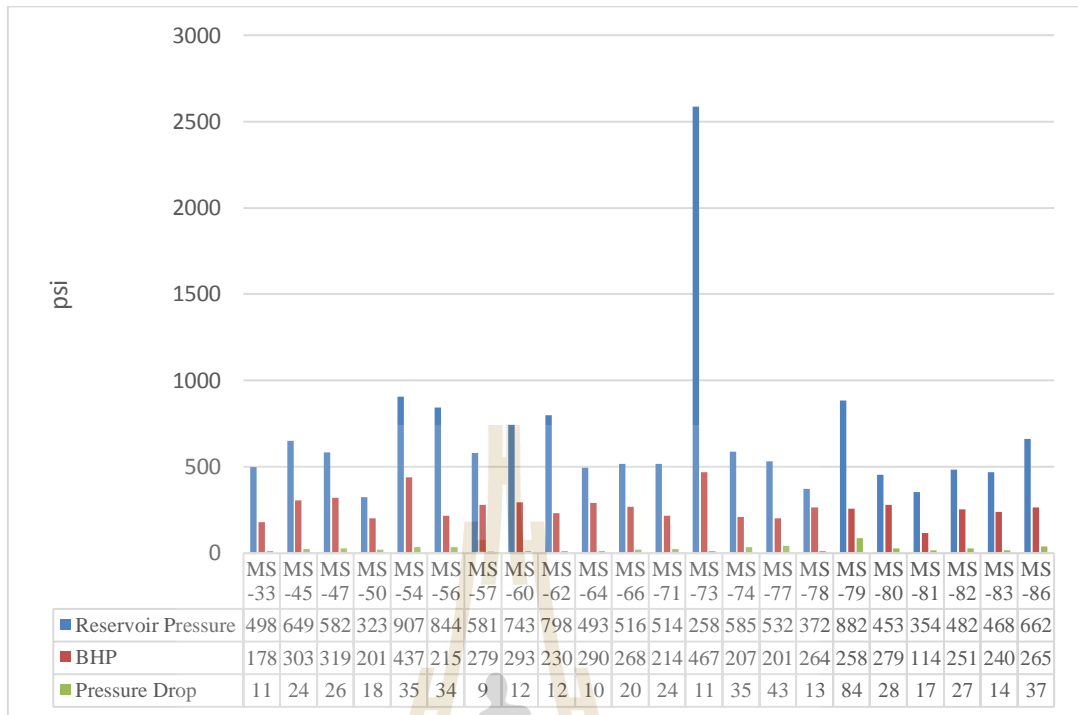


Figure 3.4 Reservoir pressure, bottom-hole pressure, and pressure drop in tubing of the 22 selected production wells of Mae-soon oil field, Fang basin

The classic approach to the shut-in bottom-hole pressure (BHP) calculation is presented graphically in Figure 3.5.

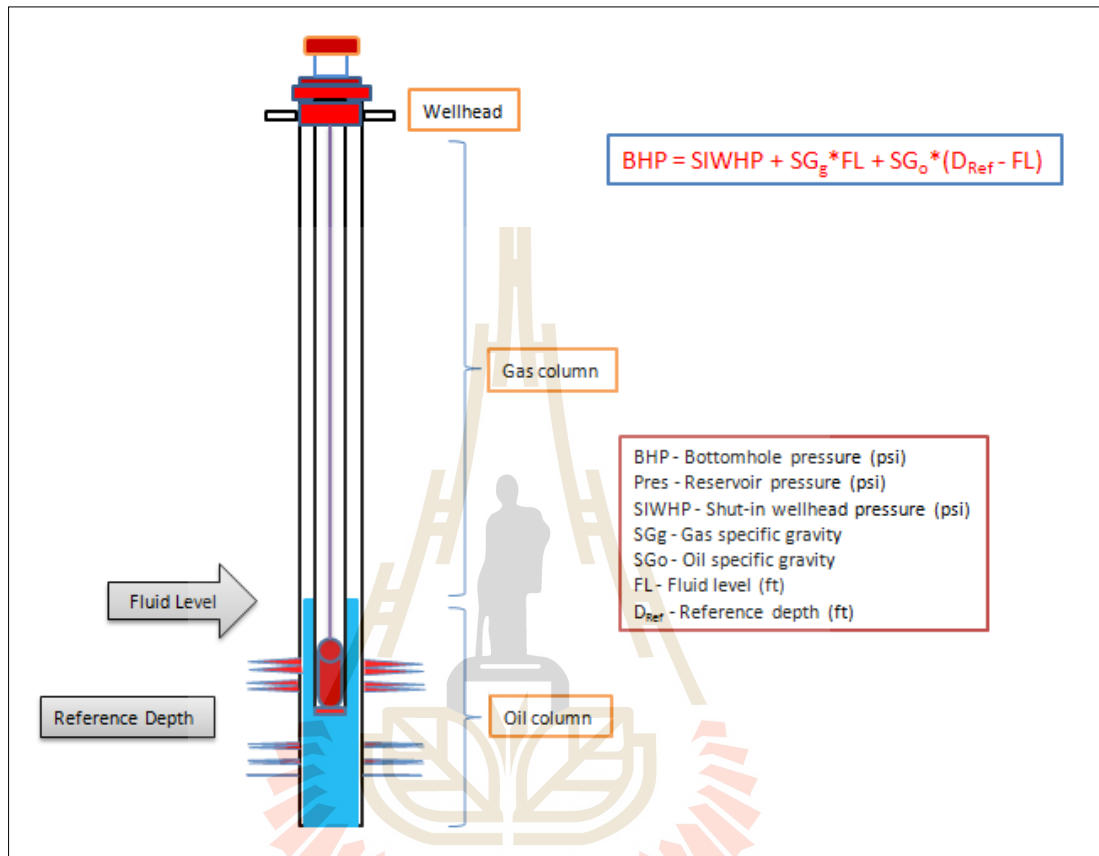


Figure 3.5 Bottom-Hole Pressure calculations

Bottom-hole pressure originated from the pressure gradient in oil and gas column of each selected production well in this study had been calculated by using the following equation.

$$BHP = SIWHP + SG_g * FL + SG_o * (D_{Ref} - FL) \quad (3.2)$$

Where

BHP	=	Bottom-hole pressure (psi)
SIWHP	=	Shut-in wellhead pressure (psi)

SGg	=	Gas specific gravity
SGo	=	Oil specific gravity
FL	=	Fluid level (ft)
DRef	=	Reference depth (ft)

In general pressure drop in tubing is decrease in pressure from one point in a pipe or tube to another point downstream due to the frictional forces on a fluid as it flows through the tube. The frictional forces are caused by the resistance to flow. The main determinants of resistance to fluid flow are fluid velocity through the pipe and fluid viscosity. The flow of any liquid or gas will always flow in the direction of least resistance (less pressure). Pressure drop increases proportional to the frictional share force within the piping network. A piping network containing a high relative roughness rating as well as many pipe fittings and joints, tube convergence, divergence, turns, surface roughness and other physical properties will affect the pressure drop. High flow velocities and/or high fluid viscosities result in a larger pressure drop across a section of pipe or a valve or elbow. Low velocity will result in lower or no pressure drop. The study considered especially in pressure drop in tubing because all of twenty two selected production wells have a low production flow rate.

In this study pressure drop in tubing was determined by the following steps;

First, the Reynolds number (N_{Re}) must be calculated to determine if the flow is laminar or turbulent by using Equation 3.3

$$N_{Re} = 1.48 \frac{q_{(bbl/d)} \rho_{(lb/ft^3)}}{\mu_{(cp)} ID_{(in)}} \quad (3.3)$$

Where	N_{Re}	=	Reynolds number
	q	=	Production rate (bbl/d)
	ρ	=	Produced fluid density (lb/ft ³)
	μ	=	Produced fluid viscosity (cp)
	ID	=	Pipe inside diameter (in)

As a result of the Reynolds number calculation of all twenty two selected wells, this number is below 2,100 and then the fluid flow was interpreted as laminar flow. Then either the Moody diagram or the Chen equation (equation 3.4) can be used to determine the friction factor. Using the Chen equation,

$$f_f = \frac{16}{N_{Re}} \quad (3.4)$$

Then,

$$u_{(ft/s)} = \frac{4q_{(bbl/d)}}{\pi ID_{(ft)}^2} \quad (3.5)$$

Finally,

$$\Delta P_{(psi)} = \frac{2\rho_{(lb/ft^3)} u_{(ft/s)}^2 l_{(ft)}}{32.17 * ID_{(ft)}} \quad (3.6)$$

Where	u	=	Fluid flow velocity (ft/s)
	ΔP	=	Pressure drop in pipe (psi)
	l	=	Length of pipe (ft)
	f_f	=	Friction factor

ID	=	Pipe inside diameter (ft)
q	=	Fluid flow rate (bbl/d)
ρ	=	Fluid density (lb/ft ³)

Oil specific gravity, oil viscosity, and oil formation volume factor, which were used in this study, were obtained from the data base of Fang basin. The relationship between pressure and oil viscosity, and pressure and formation volume factor of Fang basin are presented graphically in Figure 3.6, Figure 3.7, and Figure 3.8, respectively.

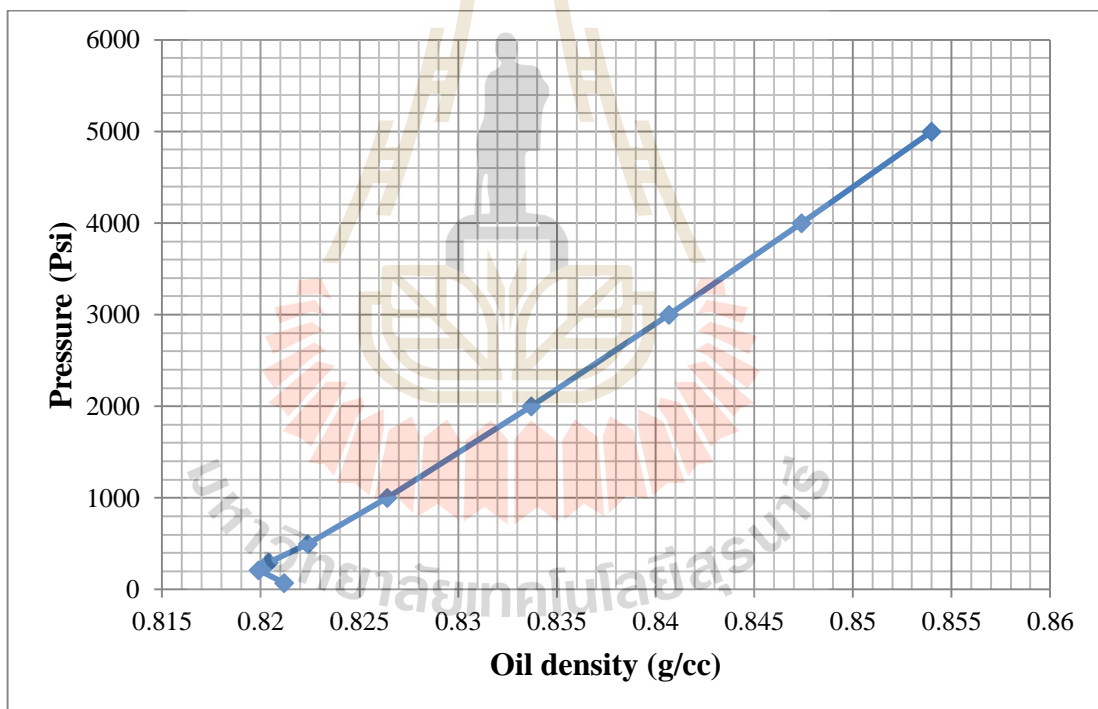


Figure 3.6 The relationship between pressure and oil density

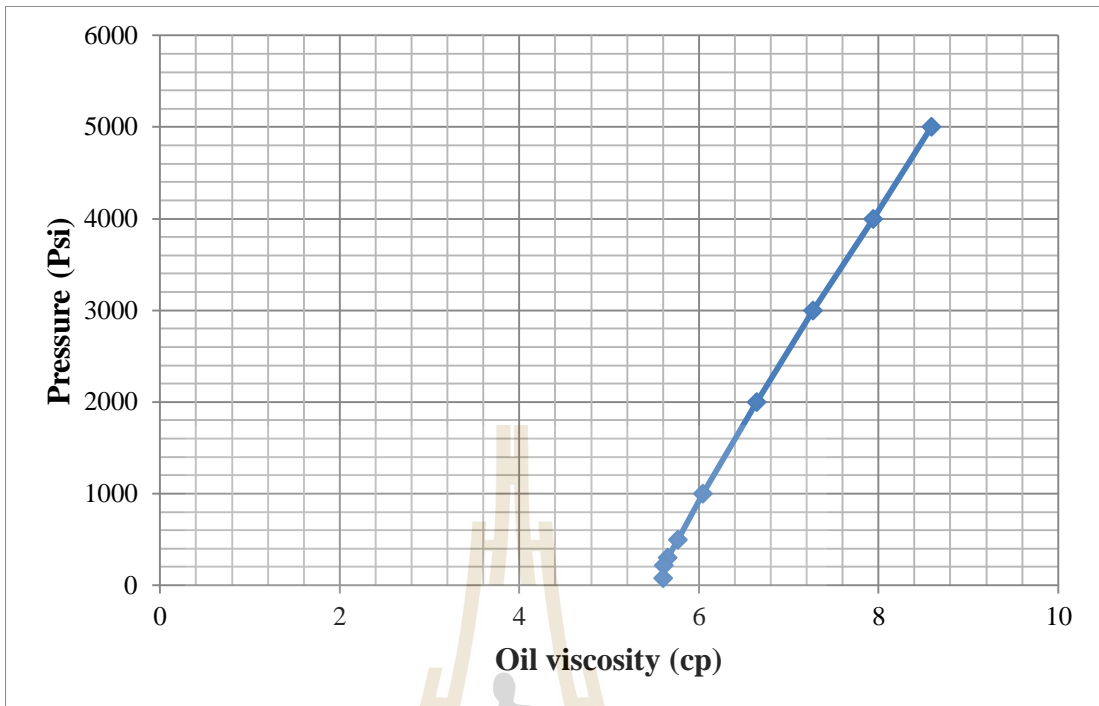


Figure 3.7 The relationship between pressure and oil viscosity

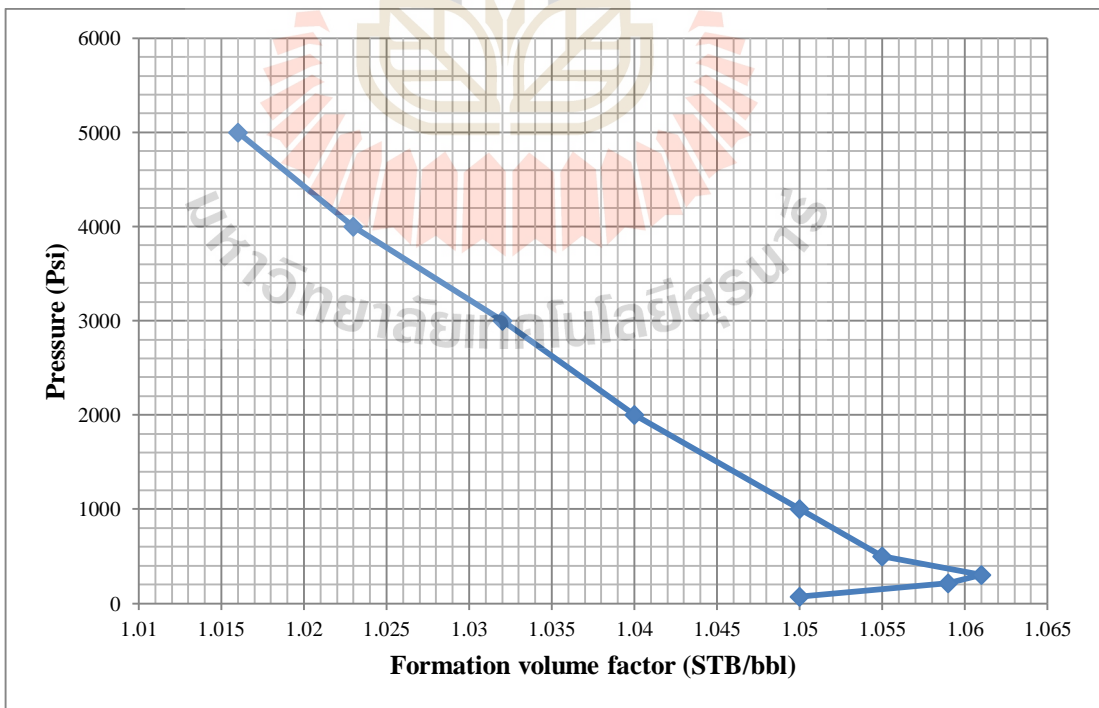


Figure 3.8 The relationship between pressure and formation volume factor

Reservoir pressure influences the production rate, reserves calculation, and the recovery of these calculated reserves. The average reservoir pressure can be evaluated from routinely available rate and flowing pressure production data. Traditionally, reservoir average pressure could be determined from an extended duration build-up test or from the radial flow equation by Darcy's law. This study adopted Darcy's law for calculating the average reservoir pressure because of lacking of any well test data in this field. Darcy's law in field units in homogenous radial flow at steady state conditions (SI units) can be expressed as the following equation.

$$q = 0.00708 \frac{kh(p_e - p_w)}{\mu B \ln\left(\frac{r_e}{r_w}\right)} \quad (3.6)$$

Where	q	=	Gross production rate (bbl/d)
	k	=	Permeability (md)
	h	=	Reservoir thickness (ft)
	P_e	=	Pressure at external reservoir boundary (psi)
	P_w	=	Bottom-hole producing pressure, BHP (psi)
	μ	=	Oil viscosity (cp)
	B	=	Formation volume factor (STB/bbl)
	r_e	=	Boundary radius (ft)
	r_w	=	Well bore radius (ft)

For easy understanding, Figure 3.9 expresses each parameter meaning graphically.

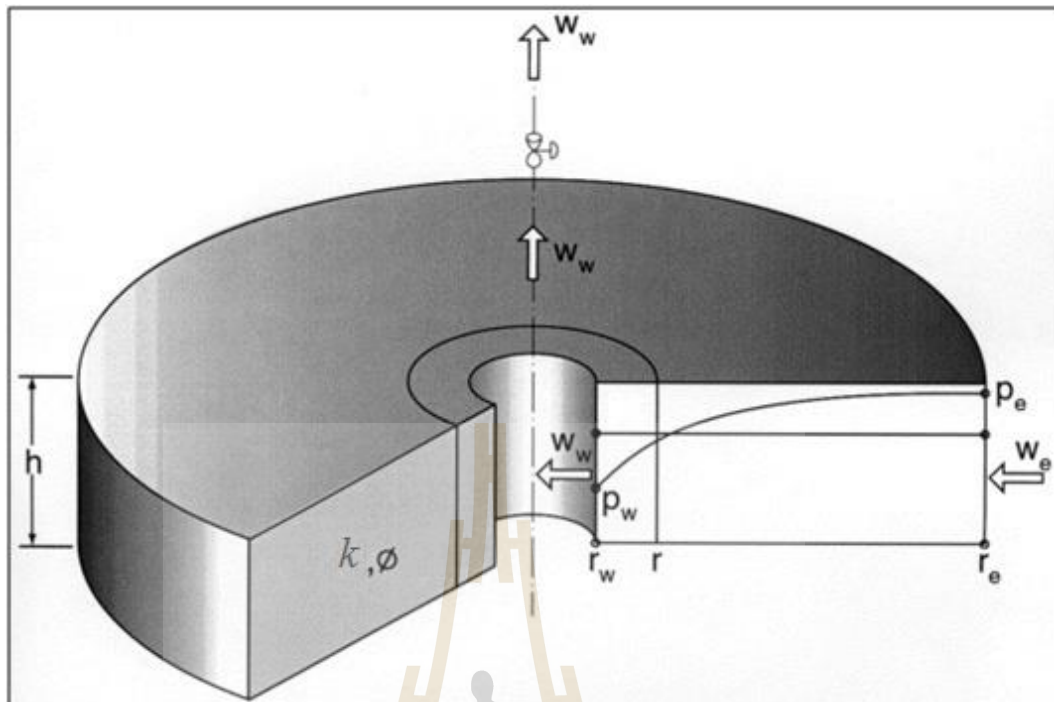


Figure 3.9 Parameters in radius flow calculation

Oil viscosity, permeability, and formation volume factor were calculated to find its average value of each well. Gross production rate were obtained from Fang basin production recorded data. Well bore radius were obtained from casing size of each well. Boundary radius and reservoir thickness were collected from work of Laikanok (2013) (unpublished).

3.3 Program developing

This topic describes the concept and steps used in the computer development for productivity index prediction in Fang oil field under various geological conditions and petroleum engineering requirements. The program hereafter is called PIP. The proposed system is based on the known analytical solutions and theories, but does not based on the heuristic knowledge, inference procedure and experience of productivity

index expert backed up by the rationale and logic. The concepts and steps include problem analysis, flowchart developing, programming, program testing, conclusion and discussion, respectively.

3.3.1 Problem analysis

Problem analysis is a primary step for a computer program development which identifies statement of problem, solution, procedure and result. The problem analysis can be divided into five sub-topics as program requirement, input data, output, variable declaration, and procedure, respectively.

3.3.1.1 Program requirement

The primary requirement of a computer program consists of program display details, basic facilities design, and necessary information that can be saved and printed in form of file and documents.

3.3.1.2 Input data

The PIP program records all required input parameters in SQL server in the program file folder of drive C. The PIP then calculates the productivity index from the maximum fifteen unfixed parameters, which are filed data and laboratory analysis data as previous mentioned in Chapter 2. However, all input unfixed parameters should have some relationships with the numerical productivity index. All required parameters are input to PIP through two pages, Parameter view page (Figure 3.10) and Data view page (Figure 3.11).

In Parameter view Page the required data parameters are assigned, including well name, production data, and laboratory analysis data. After raw data or calculated data have been input, all input data then are displayed in form of table on the Data view page automatically.

3.3.1.3 Output

After the calculation processes have been completed, results of the calculation are sent to the two output pages, Calculator page and Result page. The Calculator page has 2 parts; 1) graphs showing the relationship between PI and each input parameter, and 2) statistic tables (Root mean square or R-table, T-test table, F-test table). The Result page shows PI prediction equation, R-square and erroneous percent of its corresponding equation. Moreover, there is a Report button to report output data in form of Microsoft Office Word files too.

3.3.1.4 Variable declaration

Input parameters, output data, calculation, and processing symbols used in this developed program are declared and listed in Appendix A.

3.3.1.5 Procedure and Testing

This part demonstrates procedures to use the PIP program step by step as follows.

Step 1

In Parameter view page (Figure 3.10) user input the well name in title text box. In Factor block below, input factor or parameter and unit of that factor, click “save” icon to save the well name and all input factors. User can change or add parameters by click “additional” icon on the upper right corner.

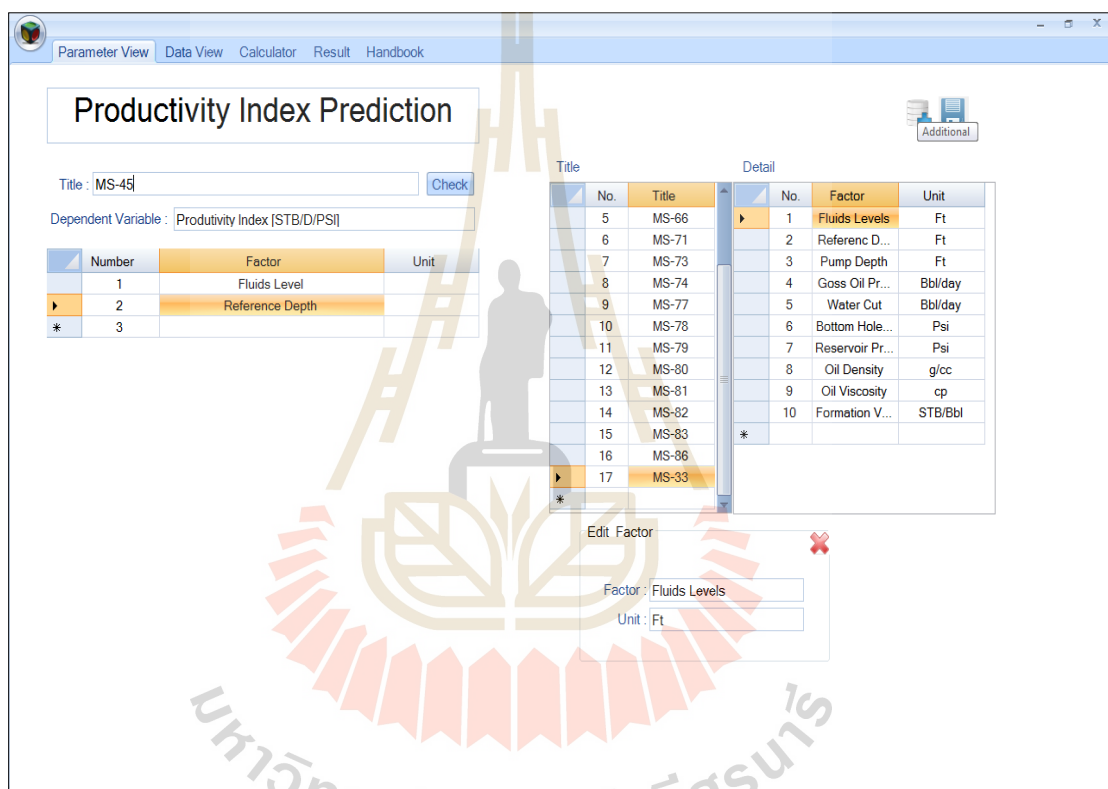


Figure 3.10 Parameter view page of the PIP program

Step 2

In Data view page (Figure 3.11) user input date and numerical data of each parameter, and click “save” icon to save. User can change or add parameters by click “additional” icon and “edit” icon. User can change the project well by click check listed block on the upper left. User can report the input data by click “Report data view” to report data in form of the Microsoft Office Word file.

Period	Productivity Index [STB/D/PS]	Fluids Levels	Referenc Depth	Pump Depth	Goss Oil Production	Water Cut	Bottom Hole Pressure	Reservoir Pressure	Oil Density	Oil Viscosity	Formation Volume Factor	
2009/4	0.037	2035.1	2133	2076	55.22	44.39	179.58	479.77	0.822	5.756	1.056	
2010/1	0.031	2028.4	2133	2076	56.51	47.39	181.44	488.64	0.822	5.761	1.056	
2010/2	0.031	2032	2133	2076	55.42	46.51	180.44	481.73	0.822	5.757	1.056	
2010/3	0.018	2028	2133	2076	56.12	50.96	181.55	486.65	0.822	5.76	1.056	
2010/4	0.016	1964.3	2133	2076	53.06	48.51	199.34	487.79	0.822	5.76	1.056	
2011/1	0.047	1954.8	2133	2076	53.12	40.17	202.01	490.79	0.822	5.762	1.056	
2011/2	0.046	1961.6	2133	2076	49.69	37.6	200.11	470.21	0.822	5.751	1.056	
2011/3	0.045	1931.2	2133	2076	51.69	39.56	208.6	489.57	0.822	5.761	1.056	
2011/4	0.046	1991.5	2133	2076	51.43	38.96	191.76	471.36	0.822	5.751	1.056	
2012/1	0.046	2034.7	2133	2076	50.36	38.34	179.7	453.47	0.822	5.742	1.056	
2012/2	0.046	2026.1	2133	2076	54.24	41.08	182.09	476.94	0.822	5.754	1.056	
2012/3	0.034	2041.5	2133	2076	58.98	48.35	177.79	498.4	0.822	0.818	1.056	
No	Preiod	Productivity Index [STB/D/PS]	Fluids Levels	Referent Depth	Pump Depth	Goss Oil Production	Water Cut	Bottom Hole Pressure	Reservo Pressure	Oil Density	Oil Viscosity	Formatic Volume Factor
*	13											

Figure 3.11 Data view page of the PIP program

In Edit page of the Data view page (Figure 3.12) user can edit each parameter manually, click “save” icon to save after finish editing. User can report the edited input data by click the “Report Data View” button to report data in form of the Microsoft Office Word file.

No.	Period	Productivity Index [STB/D/PSI]	Fluids Levels	Referenc Depth	Pump Depth	Goss Oil Production	Water Cut	Bottom Hole Pressure	Reservoir Pressure	Oil Density	Oil Viscosity	Formation Volume Factor
1	2009/4	0.037	2035.1	2133	2076	55.22	44.39	179.58	479.77	0.822	5.756	1.056
2	2010/1	0.031	2028.4	2133	2076	56.51	47.39	181.44	488.64	0.822	5.761	1.056
3	2010/2	0.031	2032	2133	2076	55.42	46.51	180.44	481.73	0.822	5.757	1.056
4	2010/3	0.018	2028	2133	2076	56.12	50.96	181.55	486.65	0.822	5.76	1.056
5	2010/4	0.016	1964.3	2133	2076	53.06	48.51	199.34	487.79	0.822	5.76	1.056
6	2011/1	0.047	1954.8	2133	2076	53.12	40.17	202.01	490.79	0.822	5.762	1.056
7	2011/2	0.046	1961.6	2133	2076	49.69	37.6	200.11	470.21	0.822	5.751	1.056
8	2011/3	0.045	1931.2	2133	2076	51.69	39.56	208.6	489.57	0.822	5.761	1.056
9	2011/4	0.046	1991.5	2133	2076	51.43	38.96	191.76	471.36	0.822	5.751	1.056
10	2012/1	0.046	2034.7	2133	2076	50.36	38.34	179.7	453.47	0.822	5.742	1.056
11	2012/2	0.046	2026.1	2133	2076	54.24	41.08	182.09	476.94	0.822	5.754	1.056
* 12	2012/3	0.034	2041.5	2133	2076	58.98	48.35	177.79	498.4	0.822	0.818	1.056

Figure 3.12 Edit page of Data view in the PIP program

In Report data of Data view page (Figure 3.13) user can consider the conclusion of each input parameter with its corresponding statistical meaning, including R-square, T-statistic, F-statistic. After finish input data examining, click “X” icon to save like other pages in this section, user can report the edited input data by click the “Report Data View” button to report data in form of the Microsoft Office Word file.

The screenshot shows the 'Data View' page of the PIP program. At the top, there are tabs for 'Parameter View', 'Data View', 'Calculator', 'Result', and 'Handbook'. A 'Report Data View' button is visible in the top right. The main data table has the following columns: Period, Productivity Index [STB/D/PSI], Fluids Levels, Referenc Depth, Pump Depth, Goss Oil Production, Water Cut, Bottom Hole Pressure, Reservoir Pressure, Oil Density, Oil Viscosity, and Formation Volume Factor. Below the main table are three sections labeled R, T, and F, each containing a smaller table with the same columns as the main table. The 'R' section has a value of 0.056222 in the Fluids Levels column. The 'T' section has a value of 0.771824. The 'F' section has a value of 0.595712. A 'Report RTF' button with a red 'X' icon is located at the bottom right of the R section.

Period	Productivity Index [STB/D/PSI]	Fluids Levels	Referenc Depth	Pump Depth	Goss Oil Production	Water Cut	Bottom Hole Pressure	Reservoir Pressure	Oil Density	Oil Viscosity	Formation Volume Factor
2009/4	0.037	2035.1	2133	2076	55.22	44.39	179.58	479.77	0.822	5.756	1.056
2010/1	0.031	2028.4	2133	2076	56.51	47.39	181.44	488.64	0.822	5.761	1.056
2010/2	0.031	2032	2133	2076	55.42	46.51	180.44	481.73	0.822	5.757	1.056
2010/3	0.018	2028	2133	2076	56.12	50.96	181.55	486.65	0.822	5.76	1.056
2010/4	0.016	1964.3	2133	2076	53.06	48.51	199.34	487.79	0.822	5.76	1.056
2011/1	0.047	1954.8	2133	2076	53.12	40.17	202.01	490.79	0.822	5.762	1.056
2011/2	0.046	1961.6	2133	2076	49.69	37.6	200.11	470.21	0.822	5.751	1.056
2011/3	0.045	1931.2	2133	2076	51.69	39.56	208.6	489.57	0.822	5.761	1.056
2011/4	0.046	1991.5	2133	2076	51.43	38.96	191.76	471.36	0.822	5.751	1.056

Fluids Levels	Referenc Depth	Pump Depth	Goss Oil Production	Water Cut	Bottom Hole Pressure	Reservoir Pressure	Oil Density	Oil Viscosity	Formation Volume Factor
0.056222			0.277706	0.840217	0.056481	0.195684		0.006462	

Fluids Levels	Referenc Depth	Pump Depth	Goss Oil Production	Water Cut	Bottom Hole Pressure	Reservoir Pressure	Oil Density	Oil Viscosity	Formation Volume Factor
0.771824			1.960812	7.251536	0.773706	1.559785		0.255031	

Fluids Levels	Referenc Depth	Pump Depth	Goss Oil Production	Water Cut	Bottom Hole Pressure	Reservoir Pressure	Oil Density	Oil Viscosity	Formation Volume Factor
0.595712			3.844783	52.5847...	0.598621	2.432928		0.065041	

Figure 3.13 Report data of Data View page of the PIP program

Step 3

In Calculator page (Figure 3.14) user can consider the plotted graph between the calculated PI and each its corresponding input parameter. The graph displays a straight line with its corresponding linear equation. Moreover, user can also consider the statistical relationship between the calculated PI and each displayed input parameter simultaneously. User can change the projected well by checking the well name block at the upper left corner.

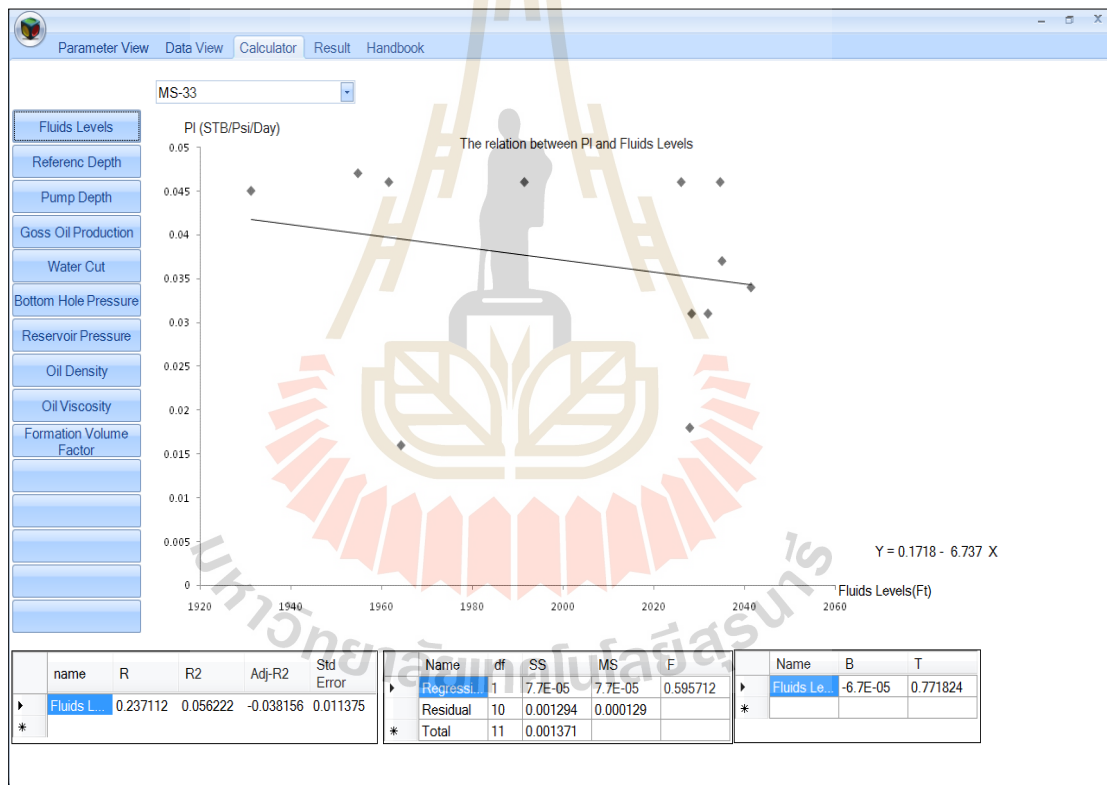


Figure 3.14 Calculation view page of the PIP program

Step 4

In Result page (Figure 3.15) the PIP shows the optimized multiple linear regression equation for the PI prediction. User can change the projected well by checking the well name block at the upper left corner and user can report the result of calculation and the optimized multiple linear regression equation for the PI prediction by click the “Report Result” button to report data in form of the Microsoft Office Word file.

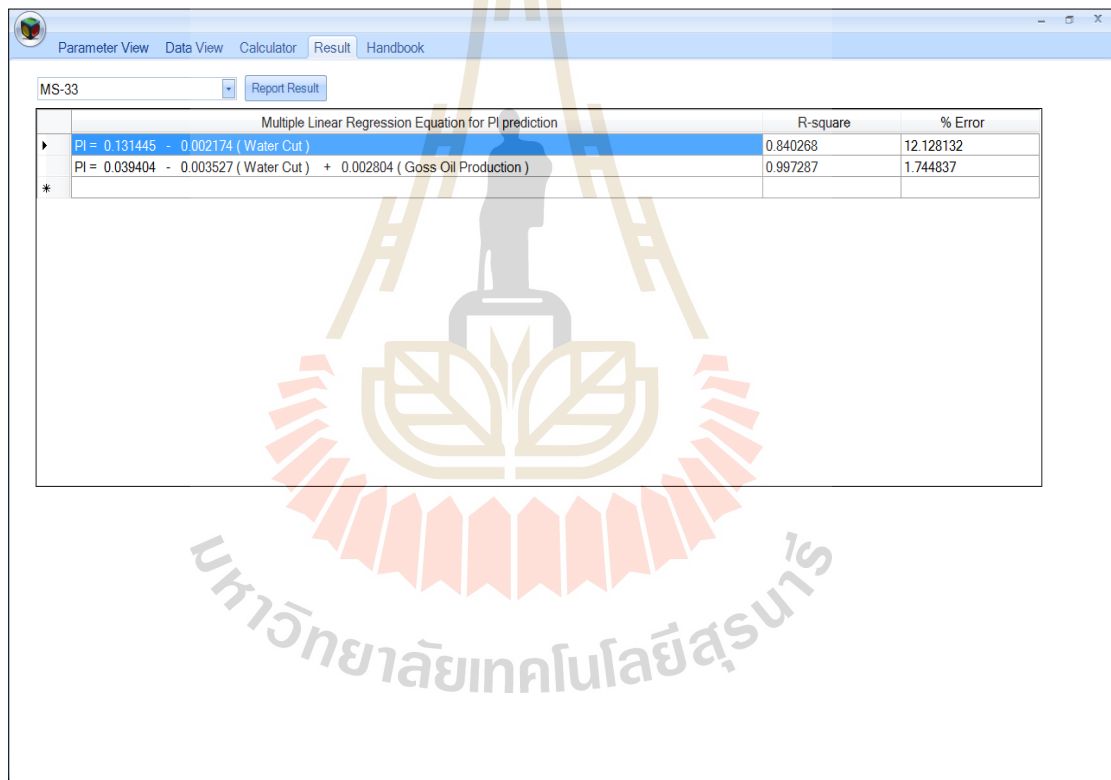


Figure 3.15 Result view page of the PIP program

PIP program has a help section for user called “Handbook” (Figure 3.16). In this section user can find some hints for any question concerned with the program and it also has some statistics theory in brief as a user guide.

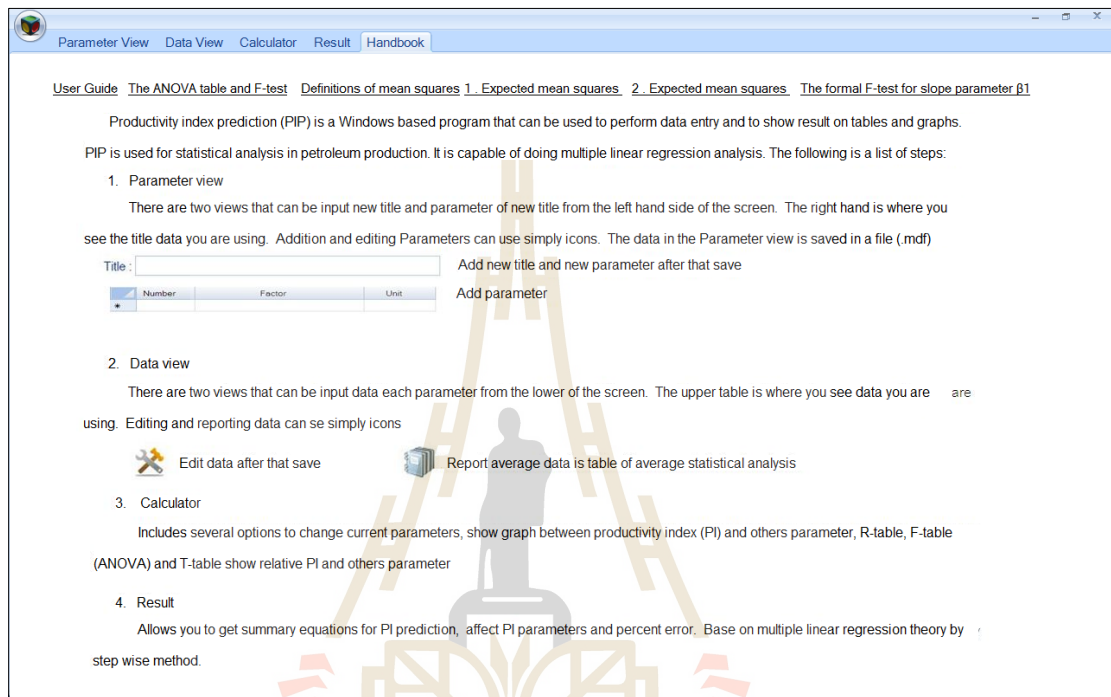


Figure 3.16 Handbook view page of the PIP program showing program handling, calculation and statistic theories

3.3.2 Flowchart

This part shows and explains the flowchart of PIP program developing. The main process includes data input, input checking, calculation, data base in SQL server linking, and output checking. These components sometimes work concurrently. The system uses forward chaining strategy. The input data are compiled and subjected to rules and conditions to obtain specific answers. This approach is necessary because the PIP records various data and need to be designed to the simply use.

The main flowchart that was developed for description the compiling process of PIP is as showed in Figure 3.17. The detail of sub-flowchart 1 and sub-flowchart 2 are presented in Figure 3.18 and Figure 3.19 respectively.

3.3.3 PIP Programming

Source codes of PIP program both for the main menu and sub-menu in each module following the flowchart structure are presented in detail in Appendix B.

3.3.4 Program system development

The PIP program system development can be divided to into three phases; 1) system shell development, 2) system control development, and 3) data base system development, respectively. In general, the system shell is used as the program structure. The system control directs the paths and flows of the program. Whereas the data base system stores the rules and conditions of statistic theories.

3.3.4.1 System shell

The system shell of PIP was developed on Microsoft Visual Basic software. The advantages of Microsoft Visual Basic are 1) equipped with GUI-Graphical User Interface, 2) ease of application, 3) quick construction, 4) supporting the management data base system, and 5) compile of complex calculation.

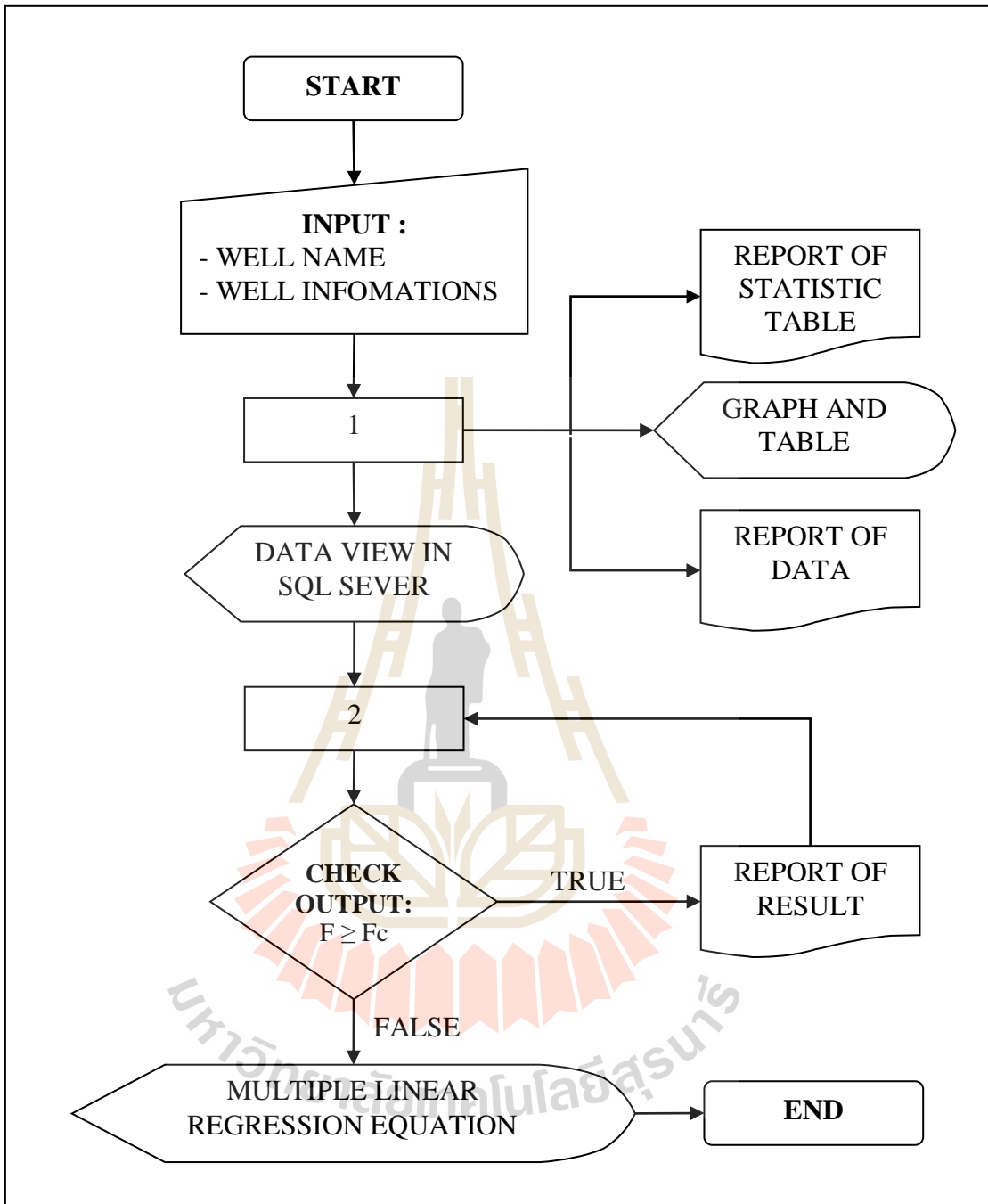


Figure 3.17 The main PIP program flowchart

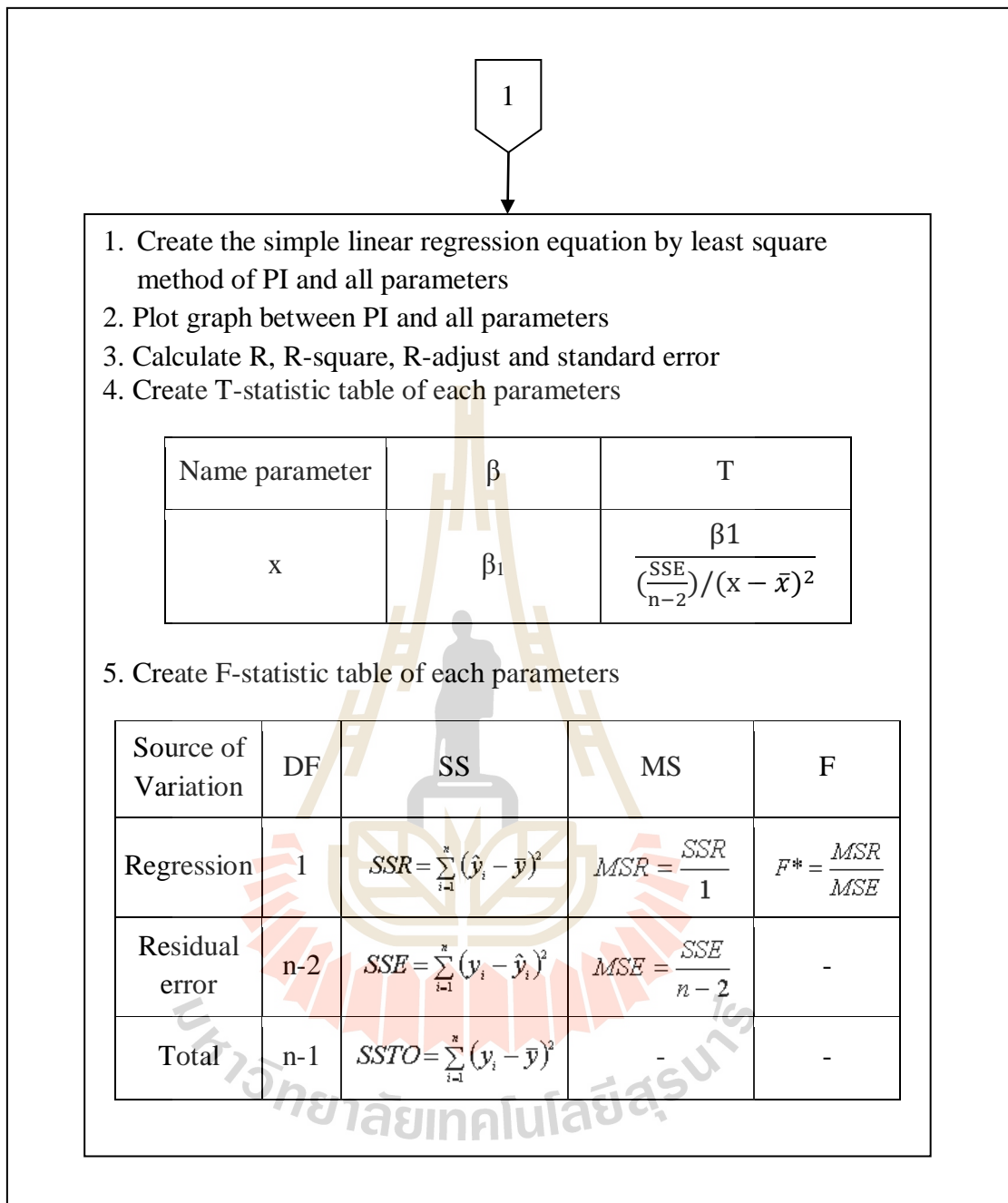


Figure 3.18 The PIP program sub-flowchart part 1

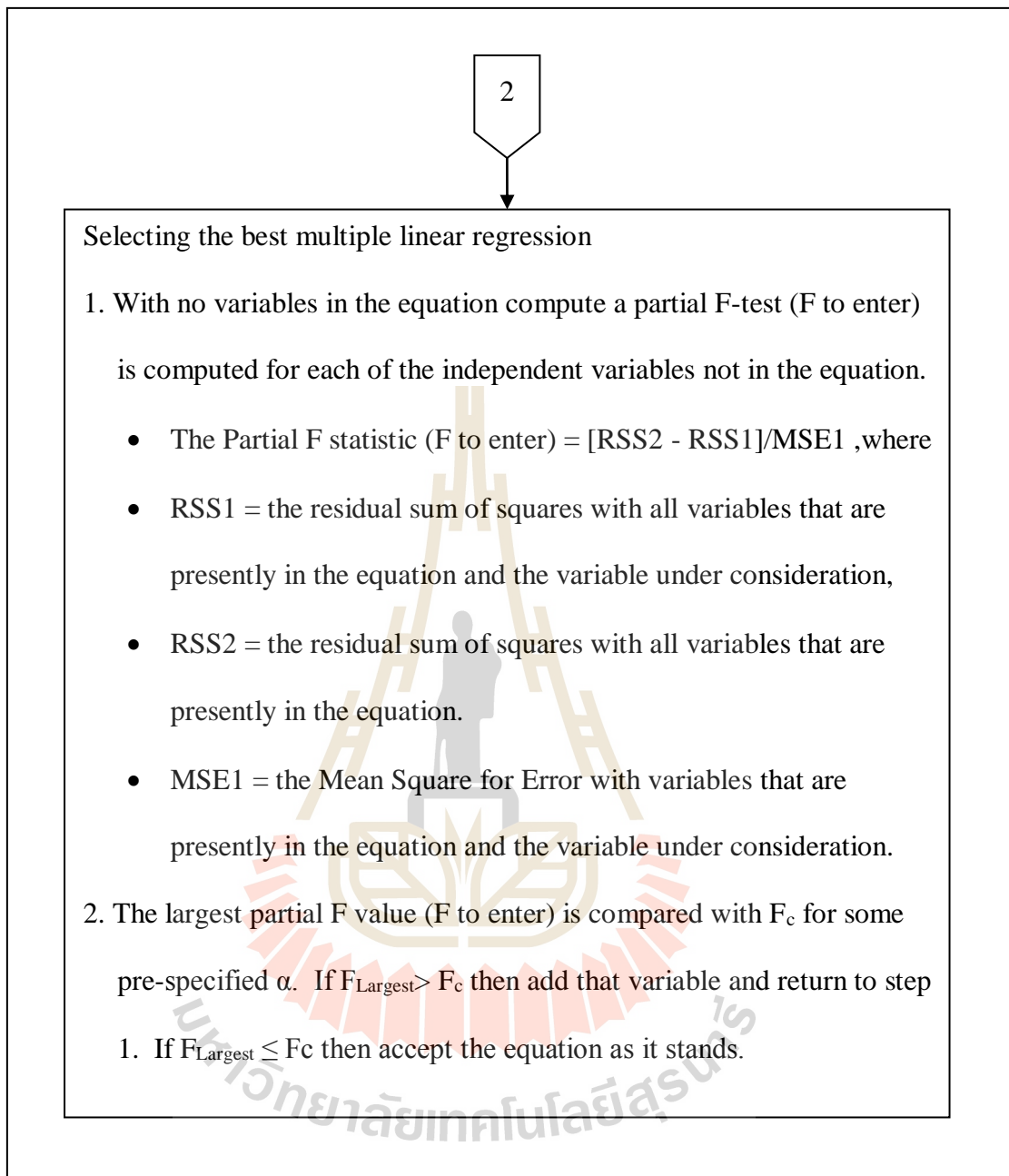


Figure 3.19 The PIP program sub-flowchart part 2

3.3.4.2 System control

The main processes for control functions are the decision making, iteration, array and procedure. The main structures of program developments in system control are as follows.

- 1) Decision structures
 - Two-way decision making; “if...Then...Else”
 - More than two-way decision making; “Select...Case”
- 2) Iteration structures
 - Known number of interaction; “For...Next”
 - Unknown number of interaction; “While...When”
 - Unknown number of interaction and go out from iteration; “Do/While...Unit/Loop”
- 3) Array and Dynamic.
 - Array structures are parts of permanent and non permanent storage data that are used for calculation.
- 4) Procedure structure; include
 - Sub program (sub routine)
 - Function (sub function)

3.3.4.3 Data base system

In data base system of PIP the input data have been compiled and stored in form of Microsoft access. They can be searched by Data Query Language (SQL) and data control function which are contained in Microsoft Visual Basic.

CHAPTER IV

RESULTS AND DISCUSSIONS

4.1 Multiple linear regression equation

During PIP is fitting and testing all related input parameters to PI by using a multiple linear regression model, it tries to eliminate insignificant parameters to form the final or the optimized equation. Degree of significant of each related input parameter can be considered from the statistical test, including T-test (T), F-test (F), and Root mean square test (R) when the less result number of these test mean the more significant to the best fit multiple linear regression equation. After the PIP has finished its processing, the final or the optimized multiple linear regression equation for the productivity index calculation of each tested well is obtained. Consequently, the best fit multiple linear regression equations together with their corresponding root mean square erroneous percentage of the twenty two selected production wells of Mae-soon oil field, Fang basin, are presented in Table 4.1.

Table 4.1 The best fit multiple linear regression equation calculated from PIP program of 22 selected production well of Mae-soon oil field, Fang basin (well MS-33 to MS-47).

Well name	Multiple linear regression equation for PI prediction	R-Square	Percent error
MS-33	PI = 0.131445 – 0.002174 (Water cut)	0.840268	12.128132
	PI = 0.039404 – 0.003527 (Water cut) + 0.002804 (Gross oil production)	0.997287	1.744837
MS-45	PI = 0.125323 – 0.000183 (Bottom-hole pressure)	0.535813	1.263371
	PI = 0.112065 – 0.000130 (Bottom-hole pressure) – 0.001126 (Water cut)	0.713514	1.002974
	PI = 1.671312 – 0.000074 (Bottom-hole pressure) – 0.002018 (Water cut) – 1.493563 (Formation volume factor)	0.934583	0.503526
MS-47	PI = 0.118035 – 0.003320 (Water cut)	0.961698	0.515314
	PI = 0.102511 – 0.003531 (Water cut) + 0.000507 (Gross oil production)	0.993813	0.246594

Table 4.1 The best fit multiple linear regression equation calculated from PIP program of 22 selected production well of Mae-soon oil field, Fang basin (well MS-50 to MS-57) (Continued).

Well name	Multiple linear regression equation for PI prediction	R-Square	Percent error
MS-50	PI = 0.260277 – 0.005756 (Water cut)	0.864749	3.699158
	PI = 0.152407 – 0.008239 (Water cut) + 0.004161 (Gross oil production)	0.997946	0.423485
MS-54	PI = 0.099196 – 0.001725 (Water cut)	0.649825	2.899960
	PI = –2.544869 – 0.001465 (Water cut) + 3.201941 (Oil density)	0.882484	1.828873
	PI = –2.641622 – 0.001547 (Water cut) + 3.296073 (Oil density) + 0.000008 (Fluids levels)	0.940396	1.311020
MS-56	PI = 2.350383 – 2.198238 (Formation volume factor)	0.389117	6.887467
MS-57	N/A	N/A	N/A

Table 4.1 The best fit multiple linear regression equation calculated from PIP program of 22 selected production well of Mae-soon oil field, Fang basin (well MS-60 to MS-71) (Continued).

Well name	Multiple linear regression equation for PI prediction	R-Square	Percent error
MS-60	PI = 0.003075 + 0.000032 (Reservoir pressure)	0.345900	9.375141
	PI = -0.000919 + 0.000076 (Reservoir pressure) - 0.001320 (Water cut)	0.730088	6.182086
MS-62	N/A	N/A	N/A
MS-64	PI = 0.071049 - 0.005216 (Water cut)	0.932118	3.131078
MS-66	PI = 0.091776 - 0.001963 (Water cut)	0.871817	1.767748
	PI = 0.080665 - 0.003170 (Water cut) + 0.000577 (Gross oil production)	0.987241	0.534352
MS-71	PI = 0.107754 - 0.002018 (Water cut)	0.792012	2.303390
	PI = 0.076901 - 0.003155 (Water cut) + 0.001224 (Gross oil production)	0.995929	0.357962

Table 4.1 The best fit multiple linear regression equation calculated from PIP program of 22 selected production well of Mae-soon oil field, Fang basin (well MS-73 to MS-77) (Continued).

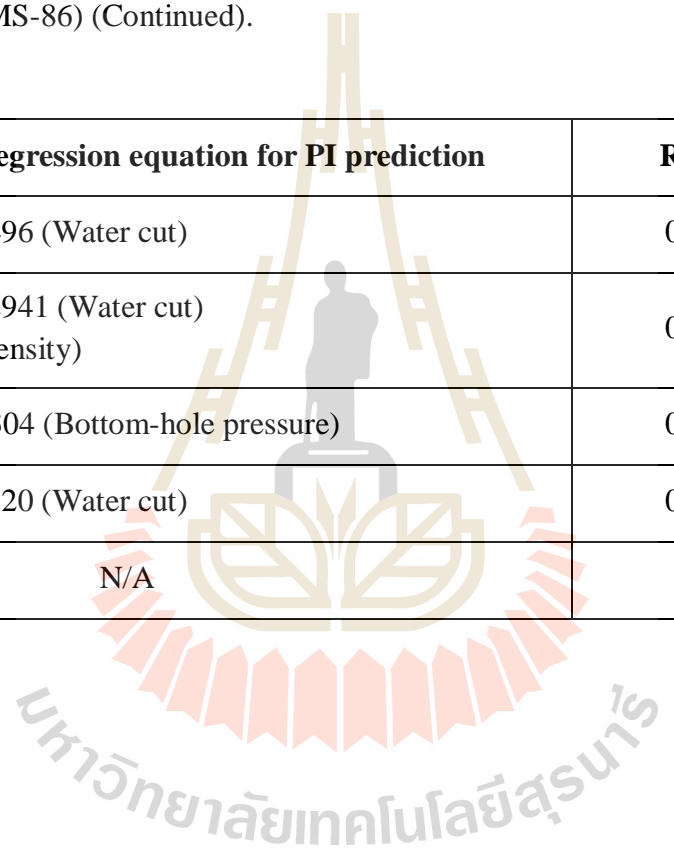
Well name	Multiple linear regression equation for PI prediction	R-Square	Percent error
MS-73	PI = 0.052506 – 0.000279 (Water cut)	0.679067	13.891843
	PI = 0.012838 – 0.000502 (Water cut) + 0.000428 (Gross oil production)	0.994889	3.355485
	PI = 0.332875 – 0.000518 (Water cut) + 0.000800 (Gross oil production) – 0.054701 (Oil viscosity)	0.997265	2.363879
MS-74	N/A	N/A	N/A
MS-77	PI = 0.162455 – 0.002878 (Water cut)	0.431295	2.104699
	PI = 0.130621 – 0.003478 (Water cut) + 0.000755 (Gross oil production)	0.982861	0.353049

Table 4.1 The best fit multiple linear regression equation calculated from PIP program of 22 selected production well of Mae-soon oil field, Fang basin (well MS-78 to MS-80) (Continued).

Well name	Multiple linear regression equation for PI prediction	R-Square	Percent error
MS-78	PI = 0.182471 – 0.008179 (Water cut)	0.553301	2.889328
	PI = –2.713506 – 0.007655 (Water cut) + 0.506802 (Oil viscosity)	0.896196	1.248794
	PI = –3.621990 – 0.009036 (Water cut) + 0.684963 (Oil viscosity) – 0.000371 (Bottom-hole pressure)	0.996470	0.240979
MS-79	PI = 0.144965 – 0.001554 (Water cut)	0.921337	2.120023
	PI = 3.075169 – 0.001869 (Water cut) – 2.779916 (Formation volume factor)	0.984397	0.962857
MS-80	PI = 0.082967 + 0.000316 (Bottom-hole pressure)	0.839941	1.609269
	PI = 0.114448 + 0.000229 (Bottom-hole pressure) – 0.001525 (Fluids levels)	0.913400	1.231164

Table 4.1 The best fit multiple linear regression equation calculated from PIP program of 22 selected production well of Mae-soon oil field, Fang basin (well MS-81 to MS-86) (Continued).

Well name	Multiple linear regression equation for PI prediction	R-Square	Percent error
MS-81	PI = 0.119079 – 0.004496 (Water cut)	0.909898	4.201183
	PI = –4.497994 – 0.002941 (Water cut) + 5.604343 (Oil density)	0.955133	2.871488
MS-82	PI = 0.042592 + 0.000304 (Bottom-hole pressure)	0.529628	3.213550
MS-83	PI = 0.100922 – 0.002320 (Water cut)	0.696079	15.079339
MS-86	N/A	N/A	N/A



4.2 Relationship between productivity index and input parameters

Results from productivity index and its related input parameters relationship testing by using Root mean square (R), T-testing (T), and F-testing (F) methods indicate that some input parameters play an important role to PI (R, T, F value is closed to zero) whereas some input parameter are not significant (R, T, F value is greater than 1.0). However, the relationship between PI and its related input parameter is vary and independent from well to well as depicted in Table 4.2.

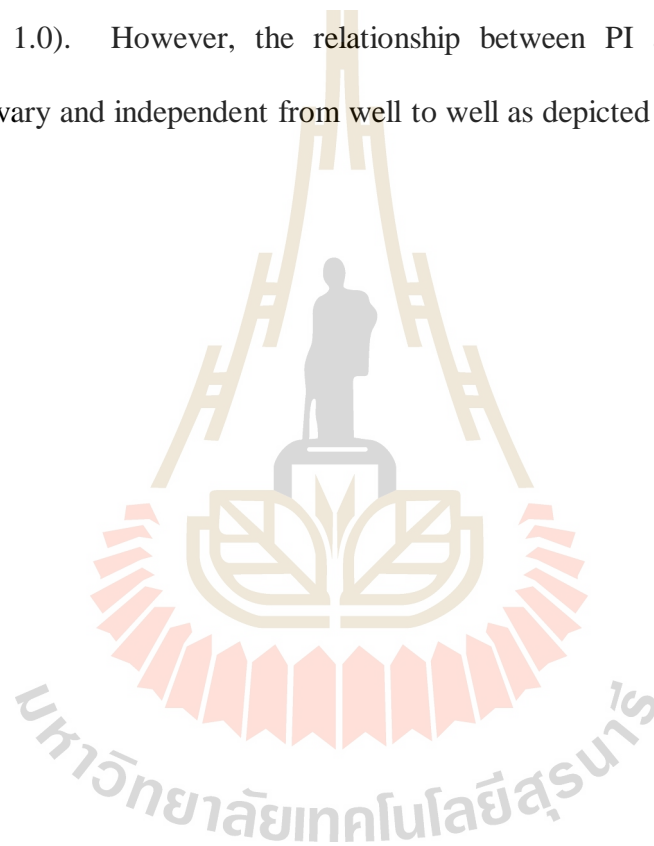


Table4.2 Relationship between PI and its related input parameters (well MS-33 to MS-66).

Parameter		Production well (MS-33 to MS-66)										
		MS-33	MS-45	MS-47	MS-50	MS-54	MS-56	MS-57	MS-60	MS-62	MS-64	MS-66
Fluids level	R	0.0562	0.5188	0.0785	0.0006	0.0050	0.3807	0.2214	0.0088	0.0874	0.0610	2.7E-05
	T	0.7718	3.2840	0.9230	0.0759	0.2237	2.4796	1.6862	0.2981	0.9786	0.8059	0.0164
	F	0.5957	10.7847	0.8520	0.0058	0.0501	6.1484	2.8431	0.0889	0.9576	0.6494	0.0003
Reference depth	R	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	T	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	F	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Pump depth	R	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	T	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	F	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Gross oil production	R	0.2777	0.0002	0.0238	0.1957	0.4324	0.3796	0.0025	0.3041	0.1947	0.0002	0.3978
	T	1.9608	0.0451	0.4937	1.559867	2.7604	2.4735	0.1577	2.0905	1.5550	0.0465	2.5703
	F	3.8447	0.0020	0.2438	2.433185	7.6200	6.1185	0.0249	4.3702	2.4180	0.0022	6.6063

Table4.2 Relationship between PI and its related input parameters (well MS-33 to MS-66) (continued).

Parameter		Production well (MS-33 to MS-66)										
		MS-33	MS-45	MS-47	MS-50	MS-54	MS-56	MS-57	MS-60	MS-62	MS-64	MS-66
Water cut	R	0.8402	0.4925	0.9629	0.86482	0.64961	0.1160	0.2474	0.0079	0.0167	0.9325	0.8715
	T	7.2515	3.1154	16.1217	7.9985	4.3059	1.1457	1.8133	0.2823	0.4120	11.7534	8.2373
	F	52.5847	9.7057	259.9081	63.9752	18.5405	1.3126	3.2883	0.0797	0.1698	138.1426	67.8538
Bottom-hole pressure	R	0.0565	0.5303	0.0782	0.0099	0.0017	0.0080	0.2213	0.0043	0.0871	0.0536	2.9E-05
	T	0.7737	3.3607	0.9212	0.3155	0.1312	0.2832	1.6859	0.2086	0.9767	0.7523	0.0169
	F	0.5983	11.294	0.8487	0.0996	0.0172	0.0802	2.8422	0.0435	0.9539	0.5659	0.0003
Reservoir pressure	R	0.1957	0.0071	0.0103	0.1105	0.4159	0.2746	0.0312	0.3451	0.1897	0.0270	0.3932
	T	1.5598	0.2667	0.3219	1.1144	2.6687	1.9455	0.5675	2.2958	1.5299	0.5266	2.5458
	F	2.433	0.0711	0.1036	1.2418	7.1220	3.7850	0.3220	5.2707	2.3407	0.2773	6.4812
Oil specific gravity	R	N/A	0.0101	N/A	N/A	0.4691	0.2681	0.0040	0.1753	0.2186	0.0287	0.2847
	T	N/A	0.3194	N/A	N/A	2.9726	1.9138	0.2009	1.4581	1.6726	0.5432	1.9950
	F	N/A	0.1020	N/A	N/A	8.8364	3.6628	0.0404	2.1260	2.7977	0.2951	3.9801

Table4.2 Relationship between PI and its related input parameters (well MS-33 to MS-66) (continued).

Parameter		Production well (MS-33 to MS-66)										
		MS-33	MS-45	MS-47	MS-50	MS-54	MS-56	MS-57	MS-60	MS-62	MS-64	MS-66
Oil viscosity	R	0.0065	0.0071	0.0088	0.1163	0.4151	0.2756	0.0309	0.3447	0.1921	0.0235	0.3928
	T	0.2550	0.2686	0.2979	1.1473	2.6641	1.9504	0.5643	2.2937	1.5419	0.4906	2.5434
	F	0.0650	0.0721	0.0887	1.3162	7.0974	3.8041	0.3184	5.2613	2.3776	0.2407	6.4690
Formation volume factor	R	N/A	0.0545	0.2934	0.0525	0.3263	0.3951	0.0031	0.3182	0.1852	0.0199	0.1203
	T	N/A	0.7596	2.0377	0.7447	2.2005	2.5556	0.1775	2.1602	1.5077	0.4506	1.1693
	F	N/A	0.5769	4.1523	0.5547	4.8424	6.5309	0.0315	4.6667	2.2731	0.2031	1.367



Table4.2 Relationship between PI and its related input parameters (well MS-71 to MS-86).

Parameter		Production well (MS-33 to MS-66)										
		MS-71	MS-73	MS-74	MS-77	MS-78	MS-79	MS-80	MS-81	MS-82	MS-83	MS-86
Fluids level	R	0.0010	0.2650	1E-06	0.0129	0.0976	0.1737	0.84014	0.7450	0.5289	0.2837	0.0787
	T	0.0980	1.8987	0.0035	0.3613	1.0397	1.4498	7.24935	5.1280	2.9971	1.5417	0.5846
	F	0.00961	3.6050	1E-05	0.1305	1.0811	2.1018	52.5531	26.296	8.9828	2.3769	0.3418
Reference depth	R	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	T	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	F	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Pump depth	R	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	T	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	F	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Gross oil production	R	0.1289	0.0695	0.0417	0.3736	0.3211	0.1143	0.65456	0.5961	0.3984	0.2329	0.0310
	T	1.2167	0.8641	0.6600	2.4422	2.1750	1.1360	4.35299	3.6444	2.3018	1.3497	0.3578
	F	1.4804	0.7466	0.4356	5.9643	4.7304	1.2904	18.9485	13.2820	5.2982	1.8217	0.1281

Table4.2 Relationship between PI and its related input parameters (well MS-71 to MS-86) (continued).

Parameter		Production well (MS-33 to MS-66)										
		MS-71	MS-73	MS-74	MS-77	MS-78	MS-79	MS-80	MS-81	MS-82	MS-83	MS-86
Water cut	R	0.7913	0.6779	0.1302	0.4313	0.5532	0.9215	0.67873	0.9097	0.0069	0.6962	0.6069
	T	6.1575	4.5872	1.2234	2.7539	3.5188	10.8332	4.59636	9.5248	0.2357	3.7078	2.4851
	F	37.915	21.043	1.4968	7.5838	12.382	117.3587	21.1265	90.7228	0.0555	13.7482	6.1757
Bottom-hole pressure	R	0.0009	0.2647	1E-06	0.0150	0.0976	0.1736	0.8402	0.7579	0.5292	0.2839	0.0787
	T	0.0971	1.8973	0.0025	0.3908	1.0403	1.4495	7.2504	5.3086	2.9986	1.5424	0.5845
	F	0.0094	3.5999	6E-06	0.1528	1.0821	2.1009	52.5682	28.182	8.9918	0.2839	0.3416
Reservoir pressure	R	0.0955	0.0498	0.0422	0.2714	0.4129	0.1486	0.6859	0.6614	0.4186	0.2385	0.0436
	T	1.0276	0.7239	0.6636	1.9301	2.6522	1.3213	4.6726	4.1928	2.3998	1.3707	0.4270
	F	1.0560	0.5241	0.4403	3.7251	7.0343	1.7459	21.8332	17.579	5.7592	1.8790	0.1823
Oil specific gravity	R	0.3594	0.0479	0.1196	0.0432	0.0405	0.0598	0.7077	0.8359	0.3190	0.2347	0.0381
	T	2.3685	0.7091	1.1656	0.6724	0.6496	0.7979	4.9211	6.7704	1.9357	1.3564	0.3980
	F	5.6101	0.5028	1.3587	0.4521	0.4220	0.6366	24.2172	45.838	3.7470	1.8399	0.1584

Table4.2 Relationship between PI and its related input parameters (well MS-71 to MS-86) (continued).

Parameter		Production well (MS-33 to MS-66)										
		MS-71	MS-73	MS-74	MS-77	MS-78	MS-79	MS-80	MS-81	MS-82	MS-83	MS-86
Oil viscosity	R	0.0953	0.0508	0.0432	0.2663	0.4154	0.1480	0.6856	0.6601	0.4187	0.2374	0.0441
	T	1.0263	0.7315	0.6721	1.9052	2.6655	1.3178	4.6694	4.1811	2.4006	1.3669	0.4297
	F	1.0532	0.5350	0.4517	3.6298	7.1047	1.7366	21.8034	17.4819	5.7630	1.8683	0.1846
Formation volume factor	R	0.0216	0.0538	0.0526	0.1946	N/A	0.1498	0.6700	0.6941	0.3568	0.1955	0.0305
	T	0.4694	0.7545	0.7453	1.5546	N/A	1.3276	4.5059	4.5194	2.1067	1.2075	0.3546
	F	0.2204	0.5692	0.5556	2.4167	N/A	1.7625	20.3037	20.4251	4.4381	1.4580	0.125737



CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

5.1 Productivity index calculated from PIP

The objective of the study is to development a computer program for productivity index calculation based on multiple linear regression theory. The PIP program has been developed for predicting productivity index of Fang basin under various geology and petroleum engineering requirements, including fluids level, gross oil production, water cut, pump depth, reference depth, reservoir pressure, bottom-hole pressure, oil specific gravity, oil viscosity, and Formation volume factor.

Based on recorded petroleum data from twenty-two selected production wells of Mae-soon located in Fang basin, the productivity index prediction for this oil field could be created together with the PIP program in multiple linear regression theory.

To examine the efficiency and accuracy of the PI calculated from PIP program, these PIs were compared to the PIs which were calculated from the manual conventional method. Result of the comparison is presented in Table 5.1. It is indicated that there is only a few different between these two groups of PI as an average differentiated percentage of 0.5035. Therefore, PI calculated from PIP program are reliable and can be compared to the conventional PI.

This may prove and assure that the prediction of PI by the PIP program can use confidently in Fang basin. However, this relationship can only be applied to Fang basin due to its data source and geological characters of study area.

Table 5.1 Differentiated percentage between conventional calculated PI and PI calculated from the PIP program.

Period	PI from conventional calculation (bbl/psi/d)	PI from PIP (bbl/psi/d)	Differentiated Percent
2009/4	0.070	0.06950	0.71
2010/1	0.067	0.06699	0.01
2010/2	0.070	0.06993	0.09
2010/3	0.069	0.06864	0.52
2010/4	0.065	0.06525	0.39
2011/1	0.070	0.07059	0.84
2011/2	0.071	0.07073	0.38
2011/3	0.071	0.07081	0.27
2011/4	0.070	0.06921	1.13
2012/1	0.070	0.07060	0.85
2012/2	0.070	0.06984	0.23
2012/3	0.069	0.06949	0.72
Average			0.5035

Moreover, result from this study can be observed the degree of significant of each input parameter for the PI calculation by PIP program. As a result, list of significant parameters in multiple linear regression equation for PI prediction by PIP program of twenty two selected production wells of Mae-soon oil field, Fang basin, is presented in Table 5.2.

Table 5.2 List of significant related input parameters in the multiple linear regression equation for PI prediction of each selected production well of Mae-soon oil field, Fang basin.

Production well	Input parameters in multiple linear regression equation for PI prediction
MS-33	Water cut, Gross oil production
MS-45	Bottom-hole pressure, Water cut, Formation volume factor
MS-47	Water cut, Gross oil production
MS-50	Water cut, Gross oil production
MS-54	Water cut, Oil density, Fluids levels
MS-56	Formation volume factor
MS-57	N/A
MS-60	Reservoir pressure, Water cut
MS-62	N/A
MS-64	Water cut
MS-66	Water cut, Gross oil production
MS-71	Water cut, Gross oil production
MS-73	Water cut, Gross oil production, Oil viscosity
MS-74	N/A
MS-77	Water cut, Gross oil production
MS-78	Water cut, Oil viscosity, Bottom-hole pressure
MS-79	Water cut, Formation volume factor
MS-80	Bottom-hole pressure, Fluids levels

Table 5.2 List of significant related input parameters in the multiple linear regression equation for PI prediction of each selected production well of Mae-soon oil field, Fang basin (continued).

Production well	Input parameters in multiple linear regression equation for PI prediction
MS-81	Water cut, Oil density
MS-82	Bottom-hole pressure
MS-83	Water cut
MS-86	N/A

5.2 Recommendations

1. This study used only old recorded data for calculation because of the lacking of new recorded data and data confidential, therefore PI calculated from PIP program might not up to date. Future study, if possible, should use more recent data for PIP program testing or for the new program developing in order to get more result accuracy.
2. The design about predicting productivity index is very difficult and complex because the limitation of some input data is not distinct and clearly explained. The complete data of Fang basin should be collected and used. Moreover, actual production data should be compared to the calculated PI for PIP program adjusting purpose.

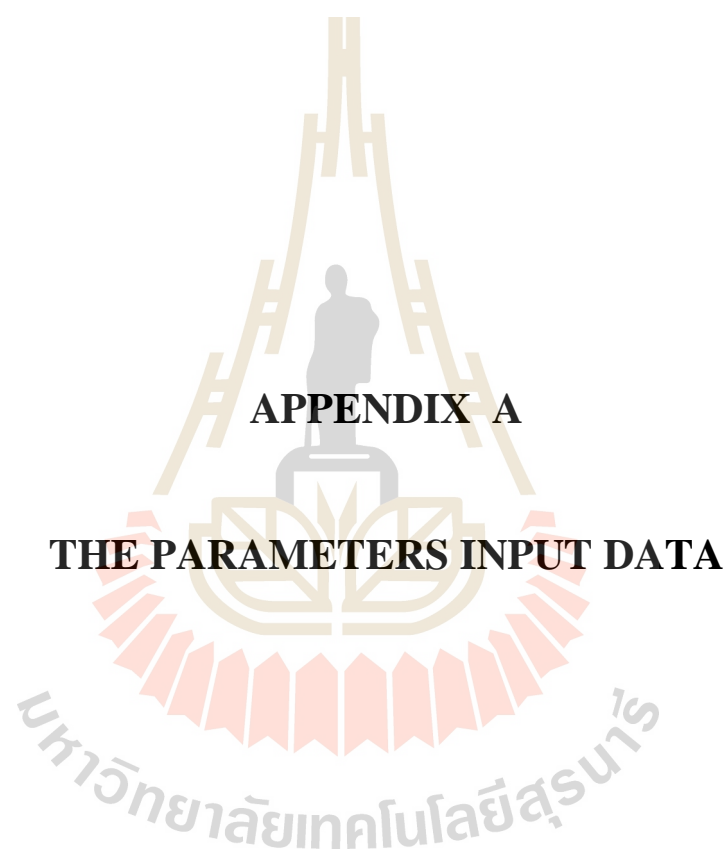
REFERENCES

- Agarwal, R.G. and Carter, R.D. (1950). **The Estimation of Permeability and Reservoir Pressure from Bottom Hole Pressure Buildup Characteristics.** Trans., AIME 91-104p.
- Ahmed, T. and McKinney, P.D. (2005). **Advance Reservoir Engineering.** Burlington, Master's thesis: Elsevier. 407 pp.
- Akhimina, N.W. (2005). **Analysis of Horizontal Gas Well Performance.** Master's thesis, The University Of Oklahoma, U.S.A.
- Babu, D.K. and Odeh, A.S. (1989). **Productivity of a Horizontal Well.** SPE paper, Reservoir Engineering. 417-421p.
- Bradley, H. B. (ed.). (1987). **Petroleum engineering handbook.** Richardson, TX, U.S.A. : Society of Petroleum Engineers.
- Chitsiripanich, S. and Pummarapanthu, T. (2011). Estimation of S1 Productivity Index from Production Data Using Multi-Linear Regression Method. **15th PTTEP TECHNICAL FORUM,** Bangkok, Thailand.
- Craft, B. C. and Hawkins, M. F. (1990). **Applied Petroleum Reservoir Engineering** (2nd ed.). New Jersey: Prentice-Hall, Eaglewood Cliffs. 431 pp.
- Deng, Liang Guo X. (2009). Application of improved multiple linear regression method in Oilfield output forecasting. **Innovation Management and Industrial Engineering 2009 Conference** (pp. 133), Department of Economic & Management, Xian Technology of University, Xian, China.

- Durongvattana, S. (1994). **Statistic And Regression Analysis**. Chulalongkorn University. Bangkok, Thailand.
- Economides, Hill A.D. (1993). **Petroleum Production System**, Prentice Hall, Inc.
- Evnger, H.H. and Muskat, M. (1942). **Calculation of Theoretical Productivity Factor**, Trans., AIME 126-139 p.
- Gentil, P. (2005). **The Use of Multi Linear Regression Model in Pattern Water flood : Physical Meaning of The Regression Coefficient**. Master's thesis. The University of Texas , Austin, U.S.A.
- Haider, M. (1936). **The productivity Index**. Sum Oil Co. 112-119p.
- He Jun, Chen, X., and Ping, Y. (2009). Application of linear regression method to analysis of oil and gas production rate decline. **Lithologic Reservoir**, Southwest Petroleum University, Chengdu, China.
- Ibelegbu C. (2004). Productivity index in Horizontal Wells. **The Journal of Scientific and Industrial Research Vol. 63** (pp. 979-984). Department of Petroleum & Gas Engineering, University of Port Harcourt, Nigeria.
- Ilk D., Velazquez R.C., Blasingame T.A. (2007). Inflow Performance Relationship For Solution Gas-Drive Reservoir Analytical Considerations. SPE paper 110821, **The 2007 SPE Annual Technical Conference and Exhibition held in Anaheim**, California, U.S.A.
- Junkui, W. (1983). Identification of types of decline curve and its application in oil field production performance. **Petroleum Expoloration and Development**, Institute of Daqing Oilfield, China.
- Junkui, W. (1993). **A Therotical Discusson of Production Decline Curve of Oil and Gas Reservoir**, Petroleum Exploration and Development, China.

- Juntalukkana, J. (2006). **Statistical Analysis and Research Planning**. (1st ed.). Faculty of sciences: Kasetsart University, Bangkok, Thailand.
- Kamkom, R. (2004). **Analysis of Two-Phase Inflow Performance in Horizontal Wells**. Master's thesis of University Of Texas, Austin, D. Hill., U.S.A.
- Katz. D.L., (1959). **Handbook of Natural Gas Engineering**, McGraw Hill Book Co., Inc., New York, U.S.A.
- Kirk, Walter B (1972). Increasing the injectivity index and productivity of wells. **United States Patent**, Marathon Oil Company, Findlay, Ohio, U.S.A.
- Kutner, M.H. and Nachtsheim, C.J. and Neter J.(2004). **Applied Linear Regression Models**. (4th ed.). McGraw-Hill education, Singapore.
- Matthews. C.S. and Russel, D.G. (1967). **Pressure Build up and Flow test in Wells, MonoGraph Series Vol. 1**. SPE of AIME, Dallas, TX, U.S.A.
- Mian, M. A. (1992). **Petroleum Engineering Handbook for the Practicing Engineer (Vol. 1)**. Tulsa: Penn Well Book. (n.p.).
- Montgomery, D.C., and Runger G.C. (1999). **Applies Statistics And Probability For Engineers**. (2nd ed.). John Wiley & Sons, Inc., New York, U.S.A.
- Mustafar, I., and Radzuan, R. (2011). A Study on Prediction of Output in Oilfield Using Multiple Linear Regression. **International Journal of Applied Science and Technology**. The university of Technology PETRONAS.
- Pongvichai, S. (2004). **Analysis of Statistic by Computer**. Chulalongkorn University.Bangkok, Thailand.
- Spath, J. and Mach, Joe M. (1999) Method for determining well productivity using automatic downtime data. **United States Patent**, Houston, Texas, U.S.A.

- Sinsombun, S. (2004). **Engineering Statistics**. (2nd ed.). Faculty of sciences: King Mongkut's Institute of Technology Ladkabung. Bangkok, Thailand.
- Taesombut, S. (2006), **Qualitative forecasting**. (1st ed.). Faculty of sciences: Kasetsert University Bangkok, Thailand.
- Vogel, J.V. (1968). Inflow Performance Relationships for Solution-Gas Drive Wells. SPE paper 1476, **The SPE Annual Fall Meeting held in Dallas**, Texas, USA
- Wang, H. (1996). **Analysis of Horizontal oil Well performance**, Master's thesis of The university of Oklahoma, Norman, U.S.A.
- Weisberg, S. (1985). **Applied Linear Regression**, 2nd ed., John Wiley, New York, U.S.A. 324 pp.
- Xianghua, Zhang Hujun X. (1997). A simple method to solve the Weng cycle parameters. **Acta Petrolei Sinica**, Yumen Petroleum administration, China.
- Xudong, Z. (1986). Prediction of the Productivity and the cumulative recoverable reserves of oil fields. **Petroleum Expoloration and Development**, Scientific Research Institute of Petroleum Exploration and Development, Beijing, China.
- Yangfeng, H. and Xiaodongel, WU (2008). New predicting method for fluid productivity indexes. **Oil Drilling & Production Technology**, MOE Key Laboratory of Petroleum in China University, Beijing, China.



APPENDIX A

THE PARAMETERS INPUT DATA

Table A1 The input data in MS-33 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2035.1	2133.0	2076.0	55.22	44.39	179.58	479.77	0.822	5.756	1.056	11.07	10.83	0.037
2010	1	2028.4	2133.0	2076.0	56.51	47.39	181.44	488.64	0.822	5.761	1.056	11.32	9.12	0.031
	2	2032.0	2133.0	2076.0	55.42	46.51	180.44	481.73	0.822	5.757	1.056	11.11	8.92	0.031
	3	2028.0	2133.0	2076.0	56.12	50.96	181.55	486.65	0.822	5.760	1.056	11.25	5.17	0.018
	4	1964.3	2133.0	2076.0	53.06	48.51	199.34	487.79	0.822	5.760	1.056	10.63	4.55	0.016
2011	1	1954.8	2133.0	2076.0	53.12	40.17	202.01	490.79	0.822	5.762	1.056	10.65	12.95	0.047
	2	1961.6	2133.0	2076.0	49.69	37.60	200.11	470.21	0.822	5.751	1.056	9.96	12.09	0.046
	3	1931.2	2133.0	2076.0	51.69	39.56	208.60	489.57	0.822	5.761	1.056	10.36	12.12	0.045
	4	1991.5	2133.0	2076.0	51.43	38.96	191.76	471.36	0.822	5.751	1.056	10.31	12.47	0.046
2012	1	2034.7	2133.0	2076.0	50.36	38.34	179.70	453.47	0.822	5.742	1.056	10.09	12.03	0.046
	2	2026.1	2133.0	2076.0	54.24	41.08	182.09	476.94	0.822	5.754	1.056	10.87	13.16	0.046
	3	2041.5	2133.0	2076.0	58.98	48.35	177.79	498.40	0.822	5.766	1.056	11.82	10.63	0.034

Table A2 The input data in MS-45 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2622.7	2736.0	2732.0	32.65	3.15	307.15	739.37	0.824	5.899	1.053	8.39	29.50	0.070
2010	1	2619.4	2736.0	2732.0	33.13	4.36	308.07	746.66	0.824	5.903	1.053	8.52	28.77	0.067
	2	2605.4	2736.0	2732.0	29.04	2.76	311.98	696.45	0.824	5.876	1.053	7.47	26.28	0.070
	3	2630.8	2736.0	2732.0	28.50	2.92	304.88	682.15	0.824	5.868	1.054	7.33	25.57	0.069
	4	2564.5	2736.0	2732.0	21.22	3.18	323.40	604.27	0.823	5.825	1.055	5.45	18.04	0.065
2011	1	2654.7	2736.0	2732.0	25.02	2.20	298.21	629.50	0.823	5.839	1.054	6.43	22.82	0.070
	2	2654.6	2736.0	2732.0	27.15	2.13	298.24	657.70	0.824	5.854	1.054	6.98	25.02	0.071
	3	2618.3	2736.0	2732.0	22.69	1.72	308.37	608.81	0.823	5.827	1.054	5.83	20.97	0.071
	4	2637.8	2736.0	2732.0	22.26	1.97	302.92	597.67	0.823	5.821	1.055	5.72	20.29	0.070
2012	1	2640.8	2736.0	2732.0	23.92	2.06	301.92	618.63	0.823	5.833	1.054	6.15	21.86	0.070
	2	2638.3	2736.0	2732.0	25.28	2.43	302.12	636.84	0.823	5.843	1.054	6.50	22.85	0.070
	3	2641.6	2736.0	2732.0	26.13	2.57	302.92	648.86	0.823	5.849	1.054	6.72	23.56	0.069

Table A3 The input data in MS-47 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2104.9	2431.1	2121.0	32.67	4.76	322.44	602.28	0.823	5.824	1.055	7.46	27.91	0.102
2010	1	2111.4	2431.1	2121.0	33.66	7.86	320.63	608.98	0.823	5.827	1.054	7.69	25.80	0.092
	2	2100.6	2431.1	2121.0	33.33	6.26	323.65	609.15	0.823	5.827	1.054	7.61	27.07	0.097
	3	2117.1	2431.1	2121.0	33.76	4.93	319.03	608.21	0.823	5.827	1.054	7.71	28.83	0.102
	4	2103.5	2431.1	2121.0	33.52	4.59	322.82	609.92	0.823	5.828	1.054	7.66	28.93	0.104
2011	1	2096.3	2431.1	2121.0	32.93	4.82	324.85	606.91	0.823	5.826	1.055	7.52	28.11	0.102
	2	2108.0	2431.1	2121.0	34.33	4.17	321.57	615.59	0.823	5.831	1.054	7.84	30.16	0.105
	3	2091.5	2431.1	2121.0	33.05	4.30	326.18	609.24	0.823	5.827	1.054	7.55	28.75	0.104
	4	2092.0	2431.1	2121.0	28.76	4.19	326.04	572.38	0.823	5.807	1.055	6.57	24.57	0.102
2012	1	2105.5	2431.1	2121.0	33.19	3.49	322.28	606.59	0.823	5.826	1.055	7.58	29.70	0.107
	2	2087.7	2431.1	2121.0	31.43	3.26	327.25	596.47	0.823	5.820	1.055	7.18	28.17	0.107
	3	2118.5	2431.1	2121.0	30.77	4.43	318.65	582.18	0.823	5.813	1.055	7.03	26.34	0.103

Table A4 The input data in MS-50 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2408.2	2519.0	2513.0	39.80	19.79	225.96	364.98	0.821	5.693	1.057	9.42	20.01	0.154
2010	1	2474.7	2519.0	2513.0	39.81	21.20	207.36	346.43	0.821	5.682	1.057	9.42	18.61	0.144
	2	2468.4	2519.0	2513.0	37.99	21.25	209.14	341.85	0.821	5.680	1.058	8.99	16.74	0.135
	3	2472.5	2519.0	2513.0	39.37	23.95	207.97	345.49	0.821	5.682	1.058	9.32	15.42	0.120
	4	2425.2	2519.0	2513.0	37.45	23.26	221.19	352.02	0.821	5.686	1.057	8.86	14.19	0.116
2011	1	2422.3	2519.0	2513.0	40.23	20.74	222.00	362.54	0.821	5.691	1.057	9.52	19.49	0.149
	2	2467.3	2519.0	2513.0	37.57	18.52	209.43	340.67	0.821	5.679	1.058	8.89	19.05	0.156
	3	2477.7	2519.0	2513.0	34.99	17.97	206.53	328.74	0.821	5.673	1.058	8.28	17.02	0.149
	4	2586.2	2519.0	2513.0	39.40	21.58	204.28	341.90	0.821	5.680	1.058	9.32	17.82	0.139
2012	1	2487.5	2519.0	2513.0	37.79	17.76	203.79	335.78	0.821	5.677	1.058	8.94	20.02	0.163
	2	2446.7	2519.0	2513.0	33.74	14.71	215.19	333.04	0.821	5.675	1.058	7.98	19.03	0.173
	3	2496.8	2519.0	2513.0	34.86	17.01	201.20	322.99	0.821	5.670	1.058	8.25	17.85	0.157

Table A5 The input data in MS-54 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2645.2	2720.0	2746.0	67.33	12.14	292.89	974.73	0.826	6.048	1.051	17.21	55.19	0.083
2010	1	2670.1	2720.0	2746.0	68.19	7.92	285.93	976.42	0.826	6.049	1.051	17.43	60.27	0.090
	2	2658.5	2720.0	2746.0	62.19	17.48	289.17	918.95	0.825	6.016	1.052	15.89	44.72	0.073
	3	2660.2	2720.0	2746.0	55.82	13.71	288.70	853.93	0.825	5.979	1.052	14.26	42.11	0.076
	4	2662.9	2720.0	2746.0	50.61	11.03	287.94	800.44	0.824	5.948	1.053	12.93	39.58	0.079
2011	1	2674.9	2720.0	2746.0	50.93	12.32	284.59	800.36	0.824	5.948	1.053	13.02	38.62	0.077
	2	2671.1	2720.0	2746.0	48.92	11.96	285.67	781.05	0.824	5.937	1.053	12.50	36.96	0.077
	3	2664.3	2720.0	2746.0	53.52	14.01	287.55	829.50	0.825	5.965	1.052	13.68	39.51	0.075
	4	2635.2	2720.0	2746.0	51.34	11.68	295.69	815.56	0.825	5.957	1.053	13.12	39.66	0.078
2012	1	2669.9	2720.0	2746.0	48.57	13.53	306.74	798.60	0.824	5.947	1.053	12.41	35.05	0.073
	2	2595.6	2720.0	2746.0	48.10	13.40	271.75	758.87	0.824	5.924	1.053	12.29	34.71	0.073
	3	2127.7	2720.0	2746.0	46.40	11.14	437.48	907.38	0.825	6.010	1.051	11.86	35.27	0.077

Table A6 The input data in MS-56 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2045.0	2383.9	2182.0	69.87	27.80	200.72	1122.03	0.827	6.133	1.049	15.65	42.07	0.046
2010	1	2071.8	2383.9	2182.0	69.41	28.77	193.23	1108.50	0.827	6.126	1.049	15.55	40.64	0.045
	2	2027.8	2383.9	2182.0	73.50	34.98	205.51	1174.65	0.827	6.164	1.049	16.46	38.52	0.040
	3	2013.6	2383.9	2182.0	67.59	34.97	209.48	1100.68	0.827	6.121	1.050	15.14	32.61	0.037
	4	1976.9	2383.9	2182.0	61.89	27.50	219.73	1035.86	0.826	6.084	1.050	13.86	34.39	0.043
2011	1	1830.9	2383.9	2182.0	42.70	24.43	260.52	823.52	0.825	5.961	1.053	9.56	18.26	0.033
	2	2045.3	2383.9	2182.0	66.51	24.26	200.61	1077.61	0.827	6.108	1.050	14.90	42.25	0.049
	3	1999.7	2383.9	2182.0	73.14	26.26	213.36	1177.79	0.827	6.166	1.049	16.38	46.88	0.049
	4	2047.2	2383.9	2182.0	72.05	32.02	200.10	1150.21	0.827	6.150	1.049	16.14	40.03	0.043
2012	1	2009.6	2383.9	2182.0	73.71	33.66	210.60	1182.55	0.827	6.168	1.049	16.51	40.05	0.042
	2	1995.0	2383.9	2182.0	66.68	31.12	214.69	1093.90	0.827	6.117	1.050	14.93	35.56	0.041
	3	1996.4	2383.9	2182.0	64.04	33.99	N/A	844.41	0.825	5.973	1.052	14.34	33.99	0.041

Table A7 The input data in MS-57of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2181.5	2327.4	2134.0	76.90	57.89	254.76	647.73	0.823	5.849	1.054	16.81	19.01	0.051
2010	1	2181.4	2327.4	2134.0	72.99	64.95	254.79	627.80	0.823	5.838	1.054	15.96	8.04	0.023
	2	2201.3	2327.4	2134.0	68.76	55.14	249.24	600.62	0.823	5.823	1.055	15.04	13.62	0.040
	3	2214.5	2327.4	2134.0	68.56	59.92	245.54	595.92	0.823	5.820	1.055	14.99	8.64	0.026
	4	2197.0	2327.4	2134.0	67.87	53.47	250.42	597.24	0.823	5.821	1.055	14.84	14.40	0.043
2011	1	2196.9	2327.4	2134.0	61.98	53.63	250.47	567.21	0.823	5.804	1.055	13.55	8.35	0.028
	2	2202.7	2327.4	2134.0	65.75	56.68	248.84	584.81	0.823	5.814	1.055	14.38	9.07	0.028
	3	2096.3	2327.4	2134.0	60.71	39.65	278.55	588.78	0.823	5.816	1.055	13.27	21.06	0.071
	4	2202.5	2327.4	2134.0	50.43	41.46	248.89	506.60	0.822	5.771	1.056	11.03	8.97	0.036
2012	1	2184.6	2327.4	2134.0	49.52	41.46	253.89	506.97	0.822	5.771	1.056	10.83	8.06	0.033
	2	2164.3	2327.4	2134.0	60.43	51.61	259.56	568.39	0.823	5.805	1.055	13.21	8.83	0.030
	3	2095.8	2327.4	2134.0	59.17	50.54	278.69	581.04	0.823	5.812	1.055	12.94	8.63	0.030

Table A8 The input data in MS-60 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2010.4	2422.5	2119.0	27.88	17.03	238.84	632.39	0.823	5.840	1.054	6.34	10.85	0.028
2010	1	2031.9	2422.5	2119.0	24.97	17.60	232.83	585.33	0.823	5.814	1.055	5.68	7.37	0.021
	2	1912.7	2422.5	2119.0	23.07	15.73	266.12	591.88	0.823	5.818	1.055	5.25	7.35	0.023
	3	1933.8	2422.5	2119.0	21.34	16.00	260.22	561.54	0.823	5.801	1.055	4.86	5.34	0.018
	4	1907.2	2422.5	2119.0	27.94	17.95	267.66	662.16	0.824	5.857	1.054	6.36	9.99	0.026
2011	1	1981.1	2422.5	2119.0	34.62	18.58	247.01	735.81	0.824	5.897	1.053	7.88	16.04	0.033
	2	1968.1	2422.5	2119.0	33.29	20.60	250.63	720.69	0.824	5.889	1.053	7.58	12.69	0.027
	3	1938.2	2422.5	2119.0	35.56	23.68	259.00	760.99	0.824	5.911	1.053	8.09	11.88	0.024
	4	1927.6	2422.5	2119.0	32.16	21.09	261.96	716.01	0.824	5.886	1.053	7.32	11.07	0.025
2012	1	2088.1	2422.5	2119.0	35.24	23.41	217.11	714.62	0.824	5.886	1.053	8.02	11.83	0.024
	2	2160.5	2422.5	2119.0	34.11	23.39	217.97	699.61	0.824	5.877	1.053	7.76	10.72	0.023
	3	1817.1	2422.5	2119.0	31.91	20.19	292.82	743.36	0.824	5.901	1.053	7.26	11.72	0.026

Table A9 The input data in MS-62 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2433.0	2439.0	2192.0	42.09	24.07	220.55	832.77	0.825	5.951	1.052	9.64	18.02	0.030
2010	1	2429.6	2439.0	2192.0	39.97	25.29	221.49	802.82	0.825	5.934	1.052	9.16	14.67	0.026
	2	2426.7	2439.0	2192.0	45.61	28.76	222.31	885.74	0.825	5.980	1.051	10.45	16.85	0.026
	3	2430.3	2439.0	2192.0	46.72	30.19	221.29	900.86	0.826	5.988	1.051	10.71	16.53	0.025
	4	2432.5	2439.0	2192.0	49.85	31.87	220.66	945.74	0.826	6.013	1.051	11.42	17.98	0.025
2011	1	2422.0	2439.0	2192.0	49.00	26.91	223.60	936.39	0.826	6.008	1.051	11.23	22.09	0.031
	2	2433.3	2439.0	2192.0	45.17	27.48	220.45	877.45	0.825	5.975	1.051	10.35	17.69	0.027
	3	2305.4	2439.0	2192.0	30.65	19.57	256.20	702.00	0.824	5.879	1.053	7.02	11.08	0.025
	4	2395.9	2439.0	2192.0	25.10	19.44	230.89	595.98	0.823	5.820	1.055	5.75	5.66	0.016
2012	1	2382.7	2439.0	2192.0	47.79	33.05	234.59	929.79	0.826	6.004	1.051	10.95	14.75	0.022
	2	2403.3	2439.0	2192.0	44.84	34.83	228.82	881.08	0.825	5.977	1.051	10.28	10.02	0.016
	3	2398.0	2439.0	2192.0	39.01	27.46	230.32	797.81	0.825	5.931	1.052	8.94	11.55	0.021

Table A10 The input data in MS-64 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2603.7	2707.5	2696.0	12.25	3.47	279.83	453.24	0.822	5.741	1.056	3.12	8.78	0.052
2010	1	2628.2	2707.5	2696.0	12.63	3.75	272.98	451.75	0.822	5.741	1.056	3.21	8.87	0.051
	2	2562.0	2707.5	2696.0	12.64	3.62	291.49	470.47	0.822	5.751	1.056	3.22	9.02	0.051
	3	2591.2	2707.5	2696.0	13.40	7.16	283.33	473.11	0.822	5.752	1.056	3.41	6.24	0.033
	4	2570.3	2707.5	2696.0	13.78	5.89	289.16	484.21	0.822	5.758	1.056	3.50	7.89	0.041
2011	1	2561.4	2707.5	2696.0	11.74	2.73	291.66	457.90	0.822	5.744	1.056	2.99	9.01	0.055
	2	2547.8	2707.5	2696.0	11.86	2.64	295.44	463.31	0.822	5.747	1.056	3.02	9.21	0.056
	3	2334.9	2707.5	2696.0	12.50	5.90	354.94	531.94	0.823	5.785	1.055	3.18	6.60	0.038
	4	2491.6	2707.5	2696.0	12.62	3.15	311.16	489.90	0.822	5.762	1.056	3.21	9.47	0.054
2012	1	2525.6	2707.5	2696.0	15.80	3.81	309.60	533.26	0.823	5.786	1.055	4.02	11.99	0.055
	2	2540.3	2707.5	2696.0	15.43	4.14	290.60	509.05	0.822	5.772	1.056	3.92	11.29	0.053
	3	2568.7	2707.5	2696.0	14.39	3.89	289.60	493.33	0.822	5.764	1.056	3.66	10.50	0.052

Table A11 The input data in MS-66 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2257.9	2422.0	2330.0	38.72	7.79	273.25	676.82	0.824	5.865	1.054	8.81	30.94	0.078
2010	1	2254.4	2422.0	2330.0	39.15	8.62	274.24	682.22	0.824	5.868	1.054	8.91	30.53	0.076
	2	2278.5	2422.0	2330.0	38.74	9.45	267.49	671.25	0.824	5.862	1.054	8.82	29.29	0.074
	3	2276.1	2422.0	2330.0	36.09	7.20	268.17	644.22	0.823	5.847	1.054	8.21	28.88	0.079
	4	2284.1	2422.0	2330.0	33.46	5.03	265.92	614.65	0.823	5.830	1.054	7.61	28.44	0.083
2011	1	2279.6	2422.0	2330.0	30.87	8.43	267.19	588.87	0.823	5.816	1.055	7.02	22.44	0.071
	2	2275.7	2422.0	2330.0	28.44	3.49	268.29	564.68	0.823	5.803	1.055	6.47	24.95	0.086
	3	2250.1	2422.0	2330.0	25.79	4.27	275.43	544.25	0.823	5.792	1.055	5.87	21.52	0.082
	4	2281.0	2422.0	2330.0	26.40	3.24	266.79	541.95	0.823	5.790	1.055	6.01	23.16	0.086
2012	1	2266.0	2422.0	2330.0	28.88	3.82	271.00	571.98	0.823	5.807	1.055	6.57	25.06	0.085
	2	2264.5	2422.0	2330.0	24.09	2.95	271.41	522.49	0.822	5.780	1.055	5.48	21.14	0.086
	3	2276.5	2422.0	2330.0	23.81	4.15	268.05	516.18	0.822	5.776	1.056	5.42	19.66	0.081

Table A12 The input data in MS-71 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2036.2	2196.7	2044.0	30.34	11.34	203.54	452.17	0.822	5.741	1.056	6.26	18.99	0.078
2010	1	1959.3	2196.7	2044.0	42.66	18.19	225.03	574.68	0.823	5.808	1.055	8.80	24.47	0.072
	2	2042.2	2196.7	2044.0	41.64	15.97	201.87	543.15	0.823	5.791	1.055	8.59	25.68	0.077
	3	2010.9	2196.7	2044.0	40.60	19.25	210.61	543.33	0.823	5.791	1.055	8.38	21.35	0.066
	4	1988.8	2196.7	2044.0	40.01	16.79	216.79	544.69	0.823	5.792	1.055	8.26	23.22	0.073
2011	1	2008.0	2196.7	2044.0	37.85	12.28	211.42	521.64	0.822	5.779	1.055	7.81	25.57	0.085
	2	2002.3	2196.7	2044.0	37.70	13.01	213.01	522.02	0.822	5.779	1.055	7.78	24.70	0.082
	3	1978.8	2196.7	2044.0	40.40	13.40	219.57	550.71	0.823	5.795	1.055	8.34	27.00	0.084
	4	1986.8	2196.7	2044.0	39.94	14.29	217.34	544.64	0.823	5.792	1.055	8.24	25.65	0.080
2012	1	2014.9	2196.7	2044.0	38.23	13.93	209.50	522.83	0.822	5.780	1.055	7.89	24.30	0.080
	2	1983.0	2196.7	2044.0	37.08	12.30	218.40	522.32	0.822	5.780	1.055	7.65	24.78	0.084
	3	1998.6	2196.7	2044.0	36.64	12.25	214.05	514.37	0.822	5.775	1.056	7.56	24.39	0.083

Table A13 The input data in MS-73 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	1362.3	2970.0	2492.0	179.15	141.00	481.72	2589.50	0.838	7.023	1.036	49.99	38.16	0.019
2010	1	1356.9	2970.0	2492.0	177.62	160.52	483.24	2573.04	0.838	7.013	1.036	49.56	17.10	0.008
	2	1347.0	2970.0	2492.0	176.71	155.88	486.00	2564.99	0.838	7.009	1.036	49.31	20.82	0.010
	3	1322.3	2970.0	2492.0	177.36	162.08	492.91	2579.55	0.838	7.017	1.036	49.49	15.28	0.007
	4	1397.9	2970.0	2492.0	176.02	161.76	471.77	2542.63	0.837	6.995	1.036	49.12	14.25	0.007
2011	1	1286.6	2970.0	2492.0	150.00	131.88	502.88	2267.71	0.835	6.830	1.039	41.86	18.13	0.011
	2	1390.2	2970.0	2492.0	179.66	164.08	473.94	2587.68	0.838	7.022	1.036	50.13	15.58	0.008
	3	1399.1	2970.0	2492.0	179.67	168.55	471.45	2585.32	0.838	7.021	1.036	50.14	11.12	0.005
	4	1411.5	2970.0	2492.0	179.53	171.02	467.98	2580.16	0.838	7.018	1.036	50.10	8.51	0.004
2012	1	1402.1	2970.0	2492.0	179.89	169.73	470.62	2587.03	0.838	7.022	1.036	50.20	10.16	0.005
	2	1404.5	2970.0	2492.0	178.68	168.48	469.95	2572.20	0.838	7.013	1.036	49.86	10.20	0.005
	3	1415.0	2970.0	2492.0	180.17	169.48	466.99	2586.77	0.838	7.022	1.036	50.28	10.69	0.005

Table A14 The input data in MS-74 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2397.5	2563.0	2523.0	69.07	7.23	214.13	884.15	0.825	5.979	1.051	16.63	61.85	0.095
2010	1	2367.8	2563.0	2523.0	57.17	5.20	222.45	776.99	0.825	5.920	1.053	13.77	51.97	0.096
	2	2393.5	2563.0	2523.0	68.37	7.14	215.26	878.47	0.825	5.976	1.051	16.46	61.23	0.095
	3	2381.6	2563.0	2523.0	75.72	7.93	218.58	953.09	0.826	6.017	1.051	18.23	67.79	0.095
	4	2384.0	2563.0	2523.0	72.22	5.68	217.92	918.49	0.826	5.998	1.051	17.39	66.54	0.097
2011	1	2363.7	2563.0	2523.0	59.23	4.59	223.58	798.12	0.825	5.932	1.052	14.26	54.64	0.098
	2	2374.0	2563.0	2523.0	54.08	4.37	220.72	745.31	0.824	5.903	1.053	13.02	49.71	0.097
	3	2356.8	2563.0	2523.0	54.47	6.67	225.51	753.91	0.824	5.907	1.053	13.12	47.81	0.093
	4	2434.6	2563.0	2523.0	67.31	6.08	203.78	856.72	0.825	5.964	1.052	16.21	61.24	0.096
2012	1	2370.7	2563.0	2523.0	44.28	4.40	221.63	651.11	0.824	5.851	1.054	10.66	39.88	0.095
	2	2373.4	2563.0	2523.0	39.34	3.82	220.89	602.46	0.823	5.824	1.055	9.47	35.51	0.095
	3	2423.3	2563.0	2523.0	38.97	3.99	206.93	584.98	0.823	5.814	1.055	9.39	34.99	0.095

Table A15 The input data in MS-77 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2340.0	2423.5	2307.0	56.39	8.60	169.49	519.47	0.822	5.778	1.056	12.84	47.78	0.142
2010	1	2333.2	2423.5	2307.0	58.12	11.37	171.40	532.13	0.823	5.785	1.055	13.23	46.75	0.135
	2	2311.9	2423.5	2307.0	48.04	10.63	177.36	475.55	0.822	5.754	1.056	10.94	37.41	0.130
	3	2244.7	2423.5	2307.0	47.59	11.28	196.14	491.53	0.822	5.763	1.056	10.84	36.32	0.128
	4	2295.2	2423.5	2307.0	39.05	10.00	182.02	424.42	0.822	5.726	1.057	8.89	29.05	0.124
2011	1	2342.6	2423.5	2307.0	49.53	8.76	166.01	473.44	0.822	5.753	1.056	11.28	40.77	0.138
	2	2352.5	2423.5	2307.0	49.36	8.40	166.01	472.39	0.822	5.752	1.056	11.24	40.96	0.139
	3	2173.1	2423.5	2307.0	50.95	8.69	216.14	532.39	0.823	5.785	1.055	11.60	42.26	0.139
	4	2254.7	2423.5	2307.0	45.04	8.17	193.33	472.87	0.822	5.752	1.056	10.25	36.87	0.137
2012	1	2340.9	2423.5	2307.0	47.76	9.47	169.26	465.68	0.822	5.748	1.056	10.87	38.29	0.134
	2	2338.3	2423.5	2307.0	53.18	10.52	169.98	500.05	0.822	5.767	1.056	12.11	42.66	0.134
	3	2227.5	2423.5	2307.0	53.40	10.67	200.95	532.37	0.823	5.785	1.055	12.16	42.72	0.134

Table A16 The input data in MS-78 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	1944.8	2342.0	1978.0	23.24	5.51	273.08	403.78	0.821	5.714	1.057	5.11	17.73	0.141
2010	1	1965.8	2342.0	1978.0	24.35	5.42	267.22	404.15	0.821	5.714	1.057	5.36	18.92	0.144
	2	1947.5	2342.0	1978.0	19.03	4.26	272.34	379.35	0.821	5.701	1.057	4.19	14.77	0.144
	3	1973.3	2342.0	1978.0	24.06	5.84	265.14	400.43	0.821	5.712	1.057	5.29	18.22	0.140
	4	1956.5	2342.0	1978.0	25.56	5.43	269.83	413.55	0.822	5.720	1.057	5.62	20.12	0.146
2011	1	1930.4	2342.0	1978.0	21.50	4.41	277.12	398.01	0.821	5.711	1.057	4.73	17.09	0.147
	2	1886.5	2342.0	1978.0	21.84	5.62	289.39	412.20	0.822	5.719	1.057	4.81	16.22	0.137
	3	1909.6	2342.0	1978.0	20.84	5.04	282.93	400.13	0.821	5.712	1.057	4.59	15.80	0.140
	4	1929.9	2342.0	1978.0	20.91	5.87	277.26	394.83	0.821	5.709	1.057	4.60	15.04	0.133
2012	1	1977.6	2342.0	1978.0	20.53	6.76	263.92	379.39	0.821	5.701	1.057	4.52	13.77	0.124
	2	1968.6	2342.0	1978.0	18.06	5.28	266.44	368.00	0.821	5.694	1.057	3.97	12.78	0.131
	3	1978.6	2342.0	1978.0	19.19	5.93	263.65	371.58	0.821	5.696	1.057	4.22	13.27	0.128

Table A17 The input data in MS-79 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2651.8	2703.5	2657.0	54.04	4.13	249.69	622.16	0.82	5.835	1.054	13.726	49.90	0.139
2010	1	2604.2	2703.5	2657.0	69.09	7.48	262.99	739.17	0.82	5.899	1.053	17.548	61.60	0.134
	2	2641.9	2703.5	2657.0	68.08	7.16	252.47	721.74	0.82	5.890	1.053	17.293	60.92	0.135
	3	2537.2	2703.5	2657.0	87.99	28.73	281.71	888.17	0.83	5.981	1.051	22.349	59.26	0.101
	4	2569.7	2703.5	2657.0	91.16	17.31	272.63	900.98	0.83	5.988	1.051	23.156	73.85	0.122
2011	1	2600.4	2703.5	2657.0	69.94	18.08	264.07	746.11	0.82	5.903	1.053	17.764	51.86	0.112
	2	2604.1	2703.5	2657.0	58.21	9.03	263.03	664.21	0.82	5.858	1.054	14.784	49.18	0.127
	3	2608.0	2703.5	2657.0	50.69	4.89	261.93	611.32	0.82	5.829	1.054	12.876	45.80	0.136
	4	2521.6	2703.5	2657.0	45.75	4.04	286.08	601.43	0.82	5.823	1.055	11.621	41.71	0.137
2012	1	2517.9	2703.5	2657.0	49.69	6.77	287.12	629.62	0.82	5.839	1.054	12.622	42.92	0.130
	2	2612.3	2703.5	2657.0	87.02	7.29	260.73	860.55	0.83	5.966	1.052	22.105	79.73	0.138
	3	2621.7	2703.5	2657.0	90.54	6.44	258.12	882.20	0.83	5.978	1.051	22.999	84.11	0.140

Table A18 The input data in MS-80 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2009	4	2038.0	2499.3	2213.0	136.67	0.90	360.47	1102.88	0.827	6.100	1.049	32.09	135.77	0.191
2010	1	2157.1	2499.3	2213.0	108.56	1.02	327.21	916.93	0.826	5.997	1.051	25.49	107.54	0.191
	2	2172.1	2499.3	2213.0	77.96	1.33	323.01	746.49	0.824	5.903	1.053	18.31	76.63	0.189
	3	2214.4	2499.3	2213.0	70.84	3.53	311.20	696.02	0.824	5.875	1.053	16.64	67.32	0.183
	4	2320.9	2499.3	2213.0	64.62	9.03	281.45	632.48	0.823	5.840	1.054	15.17	55.59	0.166
2011	1	2329.9	2499.3	2213.0	45.68	4.88	278.94	527.08	0.822	5.782	1.055	10.73	40.80	0.172
	2	2324.2	2499.3	2213.0	41.20	4.97	280.53	504.32	0.822	5.770	1.056	9.67	36.23	0.169
	3	2308.3	2499.3	2213.0	38.46	3.12	284.98	493.91	0.822	5.764	1.056	9.03	35.34	0.177
	4	2321.5	2499.3	2213.0	36.88	3.72	281.28	481.62	0.822	5.757	1.056	8.66	33.16	0.173
2012	1	2293.3	2499.3	2213.0	34.28	3.14	289.16	475.40	0.822	5.754	1.056	8.05	31.14	0.175
	2	2307.8	2499.3	2213.0	30.18	3.00	285.10	449.06	0.822	5.739	1.056	7.09	27.18	0.173
	3	2330.9	2499.3	2213.0	32.18	4.07	278.66	453.48	0.822	5.742	1.056	7.56	28.11	0.168

Table A19 The input data in MS-81 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2010	1	2114.2	2504.0	2247.0	63.49	4.35	186.68	764.19	0.824	5.913	1.053	14.94	59.14	0.105
	2	2179.8	2504.0	2247.0	40.09	0.46	168.35	533.04	0.823	5.785	1.055	9.43	39.63	0.112
	3	2167.2	2504.0	2247.0	43.25	4.24	171.88	565.27	0.823	5.803	1.055	10.17	39.01	0.102
	4	2325.7	2504.0	2247.0	42.76	7.09	127.59	516.49	0.822	5.776	1.056	10.06	35.67	0.094
2011	1	2296.6	2504.0	2247.0	33.44	4.54	135.73	439.85	0.822	5.734	1.056	7.87	28.90	0.098
	2	2268.9	2504.0	2247.0	31.04	7.68	143.48	425.83	0.822	5.726	1.057	7.30	23.62	0.086
	3	2328.6	2504.0	2247.0	27.52	8.68	126.78	377.06	0.821	5.699	1.057	6.47	17.81	0.073
	4	2350.4	2504.0	2247.0	28.27	10.48	120.69	377.84	0.821	5.700	1.057	6.65	19.09	0.076
2012	1	2381.5	2504.0	2247.0	29.07	9.83	112.00	376.42	0.821	5.699	1.057	6.84	19.24	0.075
	2	2371.6	2504.0	2247.0	27.26	9.41	113.00	360.94	0.821	5.690	1.057	6.41	17.85	0.074
	3	2358.1	2504.0	2247.0	26.41	9.73	114.00	354.19	0.821	5.687	1.057	6.21	16.68	0.071

Table A20 The input data in MS-82 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2010	2	1986.8	2458.0	2280.0	124.62	10.21	315.98	1166.10	0.828	6.135	1.048	28.78	114.41	0.139
	3	2172.5	2458.0	2280.0	97.11	17.22	264.09	926.59	0.826	6.003	1.051	22.43	79.89	0.125
	4	2149.4	2458.0	2280.0	57.72	11.62	270.55	664.33	0.824	5.858	1.054	13.33	46.10	0.121
2011	1	2163.1	2458.0	2280.0	47.68	7.93	266.74	592.03	0.823	5.818	1.055	11.01	39.75	0.126
	2	2174.6	2458.0	2280.0	43.09	6.81	263.53	557.48	0.823	5.799	1.055	9.95	36.28	0.128
	3	2180.5	2458.0	2280.0	39.82	9.70	261.86	533.48	0.823	5.786	1.055	9.20	30.11	0.115
	4	2178.8	2458.0	2280.0	39.38	9.71	262.34	531.02	0.823	5.784	1.055	9.10	29.67	0.114
2012	1	2180.4	2458.0	2280.0	36.66	5.56	261.90	511.99	0.822	5.774	1.056	8.47	31.10	0.129
	2	2169.2	2458.0	2280.0	32.51	5.75	265.02	486.78	0.822	5.760	1.056	7.51	26.76	0.125
	3	2217.9	2458.0	2280.0	33.81	7.09	251.42	482.05	0.822	5.757	1.056	7.81	26.72	0.120

Table A21 The input data in MS-83 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2010	4	2157.7	2503.0	2232.0	26.35	1.04	253.07	492.78	0.822	5.763	1.056	6.20	25.31	0.108
2011	1	2011.2	2503.0	2232.0	83.86	1.12	294.00	1056.77	0.827	6.074	1.049	19.72	82.74	0.111
	2	2110.3	2503.0	2232.0	57.42	23.23	266.30	788.58	0.825	5.926	1.052	13.50	34.18	0.067
	3	2122.2	2503.0	2232.0	39.89	22.34	262.98	625.78	0.823	5.837	1.054	9.38	17.55	0.050
	4	2147.1	2503.0	2232.0	29.51	13.80	256.02	524.40	0.822	5.781	1.055	6.94	15.71	0.060
2012	1	2168.7	2503.0	2232.0	26.18	11.24	249.99	488.15	0.822	5.761	1.056	6.16	14.94	0.064
	2	2184.9	2503.0	2232.0	26.00	11.46	245.47	481.91	0.822	5.757	1.056	6.11	14.53	0.063
	3	2205.7	2503.0	2232.0	25.08	11.18	239.64	467.80	0.822	5.749	1.056	5.90	13.91	0.063

Table A22 The input data in MS-86 of Mae-soon oil field, Fang basin

Year	3 month	FL	RD	PD	GOP	WC	BHP	PR	D	V	Bo	PDT	NOP	J
2011	3	1910.9	2596.0	2550.0	75.71	10.53	401.69	1090.29	0.827	6.093	1.049	18.46	65.18	0.097
	4	2407.9	2596.0	2550.0	42.65	3.07	262.83	650.80	0.824	5.850	1.054	10.40	39.58	0.105
2012	1	2333.0	2596.0	2550.0	48.86	2.99	283.76	728.19	0.824	5.893	1.053	11.92	45.87	0.106
	2	2332.5	2596.0	2550.0	42.46	5.20	283.90	670.15	0.824	5.861	1.054	10.36	37.26	0.099
	3	2349.7	2596.0	2550.0	40.44	5.84	279.08	646.88	0.823	5.848	1.054	9.86	34.60	0.097
	4	2400.5	2596.0	2550.0	43.71	6.98	264.89	662.49	0.824	5.857	1.054	10.66	36.73	0.095



The logo of Sakon Nakhon Rajabhat University is a large, faint watermark in the background. It features a central figure of a person standing on a pedestal, surrounded by a circular emblem with a gear-like border. The text 'มหาวิทยาลัยเทคโนโลยีสุรนารี' is written in Thai script along the bottom curve of the emblem.

APPENDIX B

SOURCE CODE OF THE PIP PROGRAM

มหาวิทยาลัยเทคโนโลยีสุรนารี

```

Imports System.IO
Imports
System.Data.SqlClient
Imports
Microsoft.Office.Tools.Excel
Imports
Microsoft.Office.Tools.Excel.Controls
Imports
Microsoft.Office.Interop
Imports
Microsoft.Office.Interop.Excel
Imports
Super2d3dGraphLibrary
Imports
System.Web.UI.WebControls
Imports
CrystalDecisions.CrystalReports.Engine
Imports
CrystalDecisions.ReportSource
Imports
CrystalDecisions.Shared
Imports
CrystalDecisions.Windows.Forms
Imports CrystalDecisions
Imports
CrystalDecisions.CrystalReports
Imports
CrystalDecisions.CrystalReports.Engine.ReportDocument
Imports System.Data
Imports
System.Globalization
Imports Word =
Microsoft.Office.Interop.Word

Public Class mdi_rgs
    Dim xlsSheet As
Excel.Worksheet
    Dim xlsBook As
Excel.Workbook
    Dim aaa As String
    Dim myFont As
System.Drawing.Font
    Dim r As String
    Dim r1 As Integer = 0
    Dim tag_result, r_no, str
As String

    Dim add_x As String
    Dim add_x1 As Integer =
0

    Private Sub
ButtonItem2_Click(ByVal
sender As System.Object,
ByVal e As
System.EventArgs)
        End
    End Sub

    Private Sub
ButtonItem5_Click(ByVal
sender As System.Object,
ByVal e As
System.EventArgs) Handles
ButtonItem5.Click
        pic_close.Visible =
False

        RibbonTabItem1.Select()
    End Sub

    Private Sub
mdi_rgs_Load(ByVal
sender As System.Object,
ByVal e As
System.EventArgs) Handles
 MyBase.Load
        Timer1.Enabled = True

        With conn1
            If .State =
ConnectionState.Open Then
                .Close()
                .ConnectionString =
strConn1
                .Open()
            End With
            show_data()

            ButtonX1.Text = "" :
            ButtonX2.Text = "" :
            ButtonX3.Text = "" :
            ButtonX4.Text = "" :
            ButtonX5.Text = "" :
            ButtonX6.Text = "" :
            ButtonX7.Text = ""
            ButtonX8.Text = "" :
            ButtonX9.Text = "" :
            ButtonX10.Text = "" :
            ButtonX11.Text = "" :
            ButtonX12.Text = "" :
            ButtonX13.Text = "" :
            ButtonX14.Text = "" :
            ButtonX15.Text = ""

            ButtonX1.Visible =
False : ButtonX2.Visible =
False : ButtonX3.Visible =
False : ButtonX4.Visible =
False : ButtonX5.Visible =
False : ButtonX6.Visible =
False : ButtonX7.Visible =
False
            ButtonX8.Visible =
False : ButtonX9.Visible =
False : ButtonX10.Visible =
False : ButtonX11.Visible =
False : ButtonX12.Visible =
False : ButtonX13.Visible =
False : ButtonX14.Visible =
False : ButtonX15.Visible =
False

            data_R.Visible = False
            : data_F.Visible = False :
            data_T.Visible = False

            RibbonControl2.MinimumS
ize.Height.ToString()

            myFont = New
System.Drawing.Font("Micro
soft Sans Serif", 11)
            RibbonControl2.Font =
myFont

            Me.WindowState =
FormWindowState.Maximiz
ed
        End Sub

    Protected Sub
show_data()
        ds1.Clear()
        sql = "SELECT
id_vrb,row_number() over
(order by id_vrb)
as[No.],vrb_name
as'Title',vrb_y FROM
VRB_X with(nolock)group
by id_vrb,vrb_name,vrb_y "
        da.SelectCommand =
New SqlCommand(sql,
conn1)
        da.Fill(ds1, "Title")
        data_x.DataSource =
ds1.Tables("Title")

        data_x.Columns(0).Visible
= False

```


<pre> data_x.Columns(3).Visible = False data_x.Columns(1).Width = 50 End Sub Private Sub txtCheck_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtCheck.Click check_data() End Sub Protected Sub check_data() If txt_name.Text = "" Then lb_check.Text = "Please enter the message." Exit Sub End If Dim sql As String = "SELECT * FROM VRB_X with(nolock) where vrb_name ='" & txt_name.Text & """" With com .Parameters.Clear() .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql dr = .ExecuteReader End With If dr.HasRows Then dr.Read() dr.Close() lb_check.Text = "Have been used. Please try again later." Exit Sub Else dr.Close() lb_check.Text = "Allowed to be used." End If End Sub Private Sub data_x_Cellmouseup(ByVal sender As System.Object, ByVal e As </pre>	<pre> System.Windows.Forms.Da taGridViewCellMouseEvent Args) Handles data_x.CellMouseUp lb_check.Visible = False dgv_db1.Visible = False If e.RowIndex = -1 Then Exit Sub Try With data_x max_id_name = .Rows.Item(e.RowIndex).C ells(0).Value.ToString() cmb_name.SelectedValue = .Rows.Item(e.RowIndex).C ells(0).Value.ToString() TextBox1.Text = .Rows.Item(e.RowIndex).C ells(2).Value.ToString() TextBox2.Text = .Rows.Item(e.RowIndex).C ells(3).Value.ToString() End With Catch ex As Exception End Try If max_id_name = "" Then Else dss.Clear() sql = "SELECT id_no,id_vrb,row_number() over (order by convert(int,id) asc) as[No.],vrb_x as'Factor' ,vrb_unit as'Unit',id_vrb,vrb_name,vr b_y FROM VRB_X with(nolock) where id_vrb='" & max_id_name & "" order by convert(int,id) asc " da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(dss, "Unit") DataGridViewX4.DataSour ce = dss.Tables("Unit") </pre>	<pre> DataGridViewX4.Columns(0).Visible = False DataGridViewX4.Columns(1).Visible = False DataGridViewX4.Columns(5).Visible = False DataGridViewX4.Columns(6).Visible = False DataGridViewX4.Columns(7).Visible = False DataGridViewX4.Columns(2).Width = 50 'RibbonTabItem2.Select() End If ButtonX16.Visible = True GroupBox3.Visible = False End Sub Private Sub DataGridViewX4_Cellmous eup(ByVal sender As System.Object, ByVal e As System.Windows.Forms.Da taGridViewCellMouseEvent Args) Handles DataGridViewX4.CellMous eUp lb_check.Visible = False dgv_db1.Visible = False If e.RowIndex = -1 Then Exit Sub Try With DataGridViewX4 TextBoxX3.Text = .Rows.Item(e.RowIndex).C ells(0).Value.ToString() 'id_no TextBoxX4.Text = .Rows.Item(e.RowIndex).C ells(3).Value.ToString() 'vrb_x </pre>
--	---	--

```

        TextBoxX5.Text
    =
    .Rows.Item(e.RowIndex).Cells(4).Value.ToString()
    'unit
        TextBoxX6.Text
    =
    .Rows.Item(e.RowIndex).Cells(1).Value.ToString()
    'id_vrb
        TextBoxX7.Text
    =
    .Rows.Item(e.RowIndex).Cells(6).Value.ToString()
    'vrb_name
        TextBoxX8.Text
    =
    .Rows.Item(e.RowIndex).Cells(7).Value.ToString()
    'vrb_y
        End With
        Catch ex As Exception
        End Try
        GroupBox3.Visible = True
        ButtonX16.Visible = False
        GroupBox4.Visible = False
    End Sub

    Private Sub pic_save_variable_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pic_save_variable.Click
        If
        dgv_db1.Rows(0).Cells(1).Value = "" Then
            Exit Sub
        End If

        txt_name.Focus()
        Tag = "add"

        If txt_name.Text = "" Then
            lb_check.Text = "Please enter the message."
            Exit Sub
        End If

        Dim sql As String =
        "SELECT * FROM VRB_X
        with(nolock) where
        vrb_name = " &
        txt_name.Text & ""

        With com
            .Parameters.Clear()
            .Connection = conn1
            .CommandType = CommandType.Text
            .CommandText = sql
            dr = .ExecuteReader
        End With

        If dr.HasRows Then
            dr.Read()
            dr.Close()
            lb_check.Text =
            "Have been used. Please try
            again later."
            Exit Sub
        Else
            dr.Close()
            lb_check.Text =
            "Allowed to be used."
        End If

        If
        (dgv_db1.Rows.Count =
        "1") Or (txt_name.Text =
        "") Or (txt_y.Text = "")
        Then
            MsgBox("data is not
            complete.")
            Exit Sub
        Else
            max_s()

            For i = 0 To
            dgv_db1.Rows.Count - 2

                Dim sqlx1 As
                String = " select max(id) as
                n from VRB_X where
                id_vrb=" & max_id_name
                & ""

                Try
                    With com
                        .CommandType =
                        CommandType.Text
                        .CommandText = sqlx1
                    End With

                    .Connection
                    = conn1
                    dr =
                    .ExecuteReader()
                    dr.Read()
                    id_x =
                    (CInt(dr.Item("n")) +
                    1).ToString("00")
                End With
                Catch
                    id_x = "01"
                End Try
                dr.Close()

                sql = "insert into
                VRB_X (
                id_vrb,vrb_name,vrb_y,vrb
                _x,vrb_unit,id)values(@id_
                vrb,@vrb_name,@vrb_y,@
                vrb_x,@vrb_unit,@id)"
                With com
                    .Connection =
                    conn1
                    .CommandType
                    = CommandType.Text
                    .CommandText
                    = sql
                    .Parameters.Clear()
                    .Parameters.Add("@id_vrb"
                    ,
                    SqlDbType.VarChar).Value
                    = max_id_name
                    .Parameters.Add("@vrb_na
                    me",
                    SqlDbType.VarChar).Value
                    = txt_name.Text.ToString
                    .Parameters.Add("@vrb_y",
                    SqlDbType.VarChar).Value
                    = txt_y.Text.ToString
                End With
                Try
                    If
                    String.IsNullOrEmpty(Trim
                    (CStr(dgv_db1.Rows(i).Cell
                    s(1).Value))) Then
                        .Parameters.Add("@vrb_x",
                        SqlDbType.VarChar).Value
                        = System.DBNull.Value
                    Else
                        .Parameters.Add("@vrb_x",
                    End With
            End Sub
    End Sub

```

<pre> SqlDbType.VarChar).Value = dgv_db1.Rows(i).Cells(1).V alue.ToString End If Catch ex As Exception .Parameters.Add("@vrb_x", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_db1.Rows(i).Cell s(2).Value))) Then .Parameters.Add("@vrb_uni t", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_uni t", SqlDbType.VarChar).Value = dgv_db1.Rows(i).Cells(2).V alue.ToString End If Catch ex As Exception .Parameters.Add("@vrb_uni t", SqlDbType.VarChar).Value = System.DBNull.Value End Try .Parameters.Add("@id", SqlDbType.VarChar).Value = id_x .ExecuteNonQuery() End With Next MsgBox("Complete....") End If show_data() </pre>	<pre> dgv_db1.Rows.Clear() txt_name.Clear() max_id_name = "" lb_check.Text = "" End Sub Private Sub dgv_db1_RowPostPaint(By Val sender As Object, ByVal e As System.Windows.Forms.Da taGridViewRowPostPaintE ventArgs) Handles dgv_db1.RowPostPaint If dgv_db1.Rows.Count = "16" Then dgv_db1.AllowUserToAdd Rows = False End If Grid With dgv_db1 If .RowCount > 0 Then For i = 0 To dgv_db1.Rows.Count - 1 .Columns(0).ReadOnly = False dgv_db1.Rows(i).Cells(0).V alue = i + 1 Next End If End With End Sub Private Sub but_Search_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) GroupBox2.Visible = False If cmb_name.Text = "" Then Exit Sub End If pt_save.Visible = True pt_add.Visible = True </pre>	<pre> pt_report.Visible = True pt_edit.Visible = True show_data_view() show_head() End Sub Protected Sub show_head() If (cmb_name.Text = "") Or (lbl_id.Text = "") Then Exit Sub End If max_s() DataGridView1.Columns(0) .HeaderText = "No." DataGridView1.Columns(0) .Width = 50 DataGridView1.Columns(1) .HeaderText = "Period" DataGridView1.Columns(2) .HeaderText = DataGridViewX2.Rows(1). Cells(0).Value.ToString DataGridView2.Columns(2) .HeaderText = DataGridViewX2.Rows(1). Cells(0).Value.ToString Dim m As Integer m = max_id + 1 i = 1 For j = 1 To m - 1 i = i + 1 Try DataGridView1.Columns(i + 1).HeaderText = DataGridViewX2.Rows(j - 1).Cells(1).Value.ToString DataGridView2.Columns(i + 1).HeaderText = DataGridViewX2.Rows(j - 1).Cells(1).Value.ToString Catch ex As Exception End Try </pre>
--	---	--

```

Next
Dim ij As Integer
max_id = max_id + 2
For ij = max_id To 17
DataGridView1.Columns(max_id).HeaderText = ""
max_id = max_id + 1
Next
End Sub
Protected Sub show_data_view() If (cmb_name.Text = "") Or (lbl_id.Text = "") Then
Exit Sub
End If
datagrid1()
lbl_name.Text = cmb_name.Text
Dim aa As String
aa = cmb_name.SelectedValue
sql = "SELECT distinct vrb_y FROM VRB_X where id_vrb = " & cmb_name.SelectedValue & ""
With com
.CommandType = CommandType.Text
.CommandText = sql
.Connection = conn1
dr = .ExecuteReader()
dr.Read()
lbl_y.Text = dr.Item("vrb_y")
End With
dr.Close()
ds.Clear()
sql = "SELECT row_number() over (order by id_no) as[No.],vrb_date ,vrb_y ,vrb_x1 ,vrb_x2 ,vrb_x3 ,vrb_x4 ,vrb_x5 ,vrb_x6 ,vrb_x7 ,vrb_x8 ,vrb_x9 " & _
" ,vrb_x10 ,vrb_x11 ,vrb_x12 ,vrb_x13 ,vrb_x14 ,vrb_x15 " & _
Next
Dim sql As String = "select count(*) as n from VRB_X where vrb_name = " & lbl_name.Text & ""
With com
.CommandType = CommandType.Text
.CommandText = sqlx
.Connection = conn1
dr = .ExecuteReader()
dr.Read()
max_id = CInt(dr.Item("n")) + 1
maxs = CInt(dr.Item("n"))
max_id_vrb = CInt(dr.Item("n"))
max_id_vrb = (CInt(15)) - CInt(dr.Item("n"))
max_id_vrb = CInt(max_id_vrb + 1)
max_id_vrb1 = CInt(dr.Item("n")) + 3
End With
dr.Close()
Dim sqlr As String = "select count(r_no) as n from VRB_r where substring(r_no,1,4) = " & lbl_id.Text & "" and (r <> '0') "
With com
.CommandType = CommandType.Text
.CommandText = sqlr
.Connection = conn1
dr = .ExecuteReader()
dr.Read()
.CommandText = sqlVERIABLE
.Connection = conn1
dr = .ExecuteReader()
dr.Read()
max_vrb = CInt(dr.Item("n"))
End With
dr.Close()
DataGridView1.DataSource = ds.Tables("aaaa")
End Sub
Protected Sub max_s()
If Tag = "add" Then
sql = "SELECT distinct MAX(id_vrb) AS max_id_vrb FROM VRB_X"
Try
With com
.CommandType = CommandType.Text
.CommandText = sql
.Connection = conn1
dr = .ExecuteReader()
dr.Read()
max_id_name = CInt(dr.Item("max_id_vrb"))
max_id_name = (CInt(max_id_name) + 1).ToString("0000")
End With
Catch
max_id_name = "0001"
End Try
dr.Close()
End If
Dim sqlVERIABLE As String = "select count(*) as n from VRB_VARIABLE where id_vrb = " & lbl_id.Text & ""
With com
.CommandType = CommandType.Text
.CommandText =

```

```

        max_r =
CInt(dr.Item("n"))
    End With
    dr.Close()

    End Sub
    Protected Sub datagrid1()
'แสดง หัวตาราง (ชื่อ x)
        Dim ds As New
DataSet
        ds.Clear()
        'sql = "select
vrb_y,vrb_x,id_no from
VRB_X with(nolock)
where vrb_name= '' &
cmb_name.Text & '' order
by id_no"
        sql = "select
vrb_y,vrb_x,id_no from
VRB_X with(nolock)
where id_vrb= '' &
lbl_id.Text & '' order by
id_no"
        da.SelectCommand =
New SqlCommand(sql,
conn1)

        da.Fill(ds, "Data")

DataGridViewX2.DataSource = ds.Tables("Data")

        For i = 1 To
dgv_save.Rows.Count - 1
            For j = 1 To 18 - 1
                ,
dgv_save.Columns(j).HeaderText = ""
                ' Next
                'Next

                For i = 1 To
dgv_save.Rows.Count - 1
                    For j = 1 To 18 - 1
                        ,
dgv_save.Columns(j).HeaderText = ""
                        ' Next
                        'Next

                        Dim ij As Integer
                        max_id = max_id + 2
                        For ij = max_id To 17

dgv_save.Columns(max_id)
.HeaderText = ""
                    max_r =
                    max_id = max_id +
                    1
                    Next

                    For i = 0 To
DataGridViewX2.Rows.Count - 1

                        With dgv_save

                            If i = 0 Then
                                If
DataGridViewX2.Rows(i).
Cells(0).Value IsNot
Nothing Then

                                    .Columns(2).HeaderText =
DataGridViewX2.Rows(i).
Cells(0).Value.ToString
                                End If
                                If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                    .Columns(3).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                End If
                                If i = 1 Then
                                    If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                        .Columns(4).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                    End If
                                End If
                                If i = 2 Then
                                    If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                        .Columns(5).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                    End If
                                End If
                                If i = 3 Then

                                        .Columns(6).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                    End If
                                If i = 4 Then
                                    If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                        .Columns(7).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                    End If
                                End If
                                If i = 5 Then
                                    If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                        .Columns(8).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                    End If
                                End If
                                If i = 6 Then
                                    If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                        .Columns(9).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                    End If
                                End If
                                If i = 7 Then
                                    If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                        .Columns(10).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                    End If
                                End If
                                If
DataGridViewX2.Rows(i).
Cells(1).Value IsNot
Nothing Then

                                    .Columns(11).HeaderText =
DataGridViewX2.Rows(i).
Cells(1).Value.ToString
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End Sub

```

End If	DataGridViewX2.Rows(i).Cells(1).Value.ToString	dgv_db.Rows(i).Cells(0).Value = i + 1
If i = 8 Then	End If	
If	End If	
DataGridViewX2.Rows(i).Cells(1).Value IsNot Nothing Then		Next
	If i = 13 Then	End If
	If	End With
.Columns(11).HeaderText = DataGridViewX2.Rows(i).Cells(1).Value.ToString	DataGridViewX2.Rows(i).Cells(1).Value IsNot Nothing Then	End Sub
End If		Private Sub
End If	.Columns(16).HeaderText = DataGridViewX2.Rows(i).Cells(1).Value.ToString	dgv_save_RowPostPaint(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewRowPostPaintEventArgs) Handles dgv_save.RowPostPaint
If i = 9 Then	End If	If
If	End If	dgv_save.Rows.Count = "31" Then
DataGridViewX2.Rows(i).Cells(1).Value IsNot Nothing Then	If i = 14 Then	
	If	dgv_save.AllowUserToAddRows = False
.Columns(12).HeaderText = DataGridViewX2.Rows(i).Cells(1).Value.ToString	DataGridViewX2.Rows(i).Cells(1).Value IsNot Nothing Then	End If
End If		For j = 2 To 18 - 1
End If	.Columns(17).HeaderText = DataGridViewX2.Rows(i).Cells(1).Value.ToString	dgv_save.Columns.Item(j).DefaultCellStyle.Format = "n4"
If i = 10 Then	End If	dgv_save.Columns.Item(j).ValueType = GetType(Double)
If	End If	Next
DataGridViewX2.Rows(i).Cells(1).Value IsNot Nothing Then	End With	With dgv_save
	Next	If .RowCount > 0
	End Sub	Then
.Columns(13).HeaderText = DataGridViewX2.Rows(i).Cells(1).Value.ToString	Private Sub	For i = 0 To dgv_save.Rows.Count - 1
End If	dgv_db_RowPostPaint(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewRowPostPaintEventArgs) Handles dgv_db.RowPostPaint	.Columns(0).ReadOnly = False
End If	If dgv_db.Rows.Count = max_id_vrb Then	dgv_save.Rows(i).Cells(0).Value = i + add_x1
If i = 11 Then	End If	Next
If	End If	End If
DataGridViewX2.Rows(i).Cells(1).Value IsNot Nothing Then	dgv_db.AllowUserToAddRows = False	End With
	End If	
.Columns(14).HeaderText = DataGridViewX2.Rows(i).Cells(1).Value.ToString	With dgv_db	
End If	If .RowCount > 0	
End If	Then	
If i = 12 Then	For i = 0 To dgv_db.Rows.Count - 1	
If	End Sub	
DataGridViewX2.Rows(i).Cells(1).Value IsNot Nothing Then		
	.Columns(0).ReadOnly = False	
.Columns(15).HeaderText =		Private Sub
		DataGridView2_RowPostP

<pre> aint(ByVal sender As Object, ByVal e As System.Windows.Forms.Da taGridViewRowPostPaintE ventArgs) Handles DataGridView2.RowPostPa int If DataGridView2.Rows.Coun t = max_vrb + 1 Then DataGridView2.AllowUser ToAddRows = False End If For j = 1 To 16 DataGridView2.Columns.It em(j).DefaultCellStyle.For mat = "n4" DataGridView2.Columns.It em(j).ValueType = GetType(Double) Next End Sub Private Sub dgv_save_DataError(ByVal sender As System.Object, ByVal e As System.Windows.Forms.Da taGridViewDataErrorEvent Args) Handles dgv_save.DataError MessageBox.Show(e.Excep tion.Message) End Sub Private Sub DataGridView2_DataError(ByVal sender As System.Object, ByVal e As System.Windows.Forms.Da taGridViewDataErrorEvent Args) Handles DataGridView2.DataError MessageBox.Show(e.Excep tion.Message) End Sub Private Sub pt_save_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pt_save.Click </pre>	<pre> If dgv_save.Rows.Count = 1 Then MsgBox("Please enter the message.") Exit Sub End If If lbl_id.Text = "" Then If cmb_name.SelectedValue = "" Then MsgBox("Please select the Parameter....") Exit Sub End If End If pt_save.Focus() max_s() For i = 0 To dgv_save.Rows.Count - 2 For j = 0 To max_id_vrb1 - 1 Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(j).Value))) Then MsgBox("Data Deficient....") Exit Sub End If Catch ex As Exception MsgBox("Data Deficient....") Exit Sub End Try Next Next For i = 0 To dgv_save.Rows.Count - 2 sql = "insert into VRB_VARIABLE " & _ " (id_vrb,vrb_date,vrb_y,vrb_ x1,vrb_x2,vrb_x3,vrb_x4,vr b_x5,vrb_x6,vrb_x7,vrb_x8 ,vrb_x9,vrb_x10,vrb_x11,vr </pre>	<pre> b_x12,vrb_x13,vrb_x14,vrb _x15," & _ " yx1,yx2,yx3,yx4,yx5,yx6,y x7,yx8,yx9,yx10,yx11,yx12 ,yx13,yx14,yx15," & _ " y_1,x_1,x_2,x_3,x_4,x_5,x _6,x_7,x_8,x_9,x_10,x_11, x_12,x_13,x_14,x_15)" & - " values " & _ " (@id_vrb,@vrb_date,@vrb _y,@vrb_x1,@vrb_x2,@vr b_x3,@vrb_x4,@vrb_x5,@ vrb_x6,@vrb_x7,@vrb_x8, @vrb_x9,@vrb_x10,@vrb_ x11,@vrb_x12,@vrb_x13, @vrb_x14,@vrb_x15," & _ " @yx1,@yx2,@yx3,@yx4, @yx5,@yx6,@yx7,@yx8, @yx9,@yx10,@yx11,@yx1 2,@yx13,@yx14,@yx15," & _ " @y_1,@x_1,@x_2,@x_3, @x_4,@x_5,@x_6,@x_7, @x_8,@x_9,@x_10,@x_11 ,@x_12,@x_13,@x_14,@x _15)" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() .Parameters.Add("@id_vrb" , SqlDbType.VarChar). Value = lbl_id.Text.ToString End With If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(1).Value))) Then .Parameters.Add("@vrb_dat e", SqlDbType.VarChar). Value = System.DBNull.Value </pre>
---	--	--

<pre> SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx3", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x3 ", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(5). Value.ToString .Parameters.Add("@yx3", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(5). Value.ToString) .Parameters.Add("x_3", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(5). Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x3 ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx3", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_3", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(6).Value))) Then .Parameters.Add("@vrb_x4 ", </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx4", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x4 ", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(6). Value.ToString .Parameters.Add("@yx4", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(6). Value.ToString) .Parameters.Add("x_4", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(6). Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x4 ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx4", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_4", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(7).Value))) Then .Parameters.Add("@vrb_x5 ", </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx5", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x5 ", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(7). Value.ToString .Parameters.Add("@yx5", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(7). Value.ToString) .Parameters.Add("x_5", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(7). Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x5 ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx5", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_5", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(8).Value))) Then .Parameters.Add("@vrb_x6 ", </pre>
---	---	---

<pre> SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx6", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x6 ", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(8). Value.ToString .Parameters.Add("@yx6", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(8). Value.ToString) .Parameters.Add("x_6", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(8). Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x6 ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx6", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_6", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(9).Value))) Then .Parameters.Add("@vrb_x7 ", </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx7", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x7 ", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(9). Value.ToString .Parameters.Add("@yx7", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(9). Value.ToString) .Parameters.Add("x_7", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(9). Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x7 ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx7", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_7", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(10).Value))) Then .Parameters.Add("@vrb_x8 ", </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx8", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x8 ", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(10). Value.ToString .Parameters.Add("@yx8", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(10)).Value.ToString) .Parameters.Add("x_8", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(10)).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x8 ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx8", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_8", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(11).Value))) Then .Parameters.Add("@vrb_x9 ", </pre>
---	--	---

<pre> ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx9", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x9 ", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(11). Value.ToString .Parameters.Add("@yx9", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(11).Value.ToString) .Parameters.Add("x_9", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(11).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x9 ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx9", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_9", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(12).Value))) Then .Parameters.Add("@vrb_x1 </pre>	<pre> 0", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx10", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x1 0", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(12). Value.ToString .Parameters.Add("@yx10", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(12).Value.ToString) .Parameters.Add("x_10", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(12).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x1 0", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx10", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_10", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(13).Value))) Then .Parameters.Add("@vrb_x1 </pre>	<pre> 1", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx11", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x1 1", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(13). Value.ToString .Parameters.Add("@yx11", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(13).Value.ToString) .Parameters.Add("x_11", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(13).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x1 1", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx11", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_11", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(14).Value))) Then </pre>
---	--	---

<pre> .Parameters.Add("@vrb_x1 2", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx12", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x1 2", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(14). Value.ToString .Parameters.Add("@yx12", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(14).Value.ToString) .Parameters.Add("x_12", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(14).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x1 2", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx12", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_12", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim </pre>	<pre> (CStr(dgv_save.Rows(i).Cel ls(15).Value))) Then .Parameters.Add("@vrb_x1 3", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx13", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x1 3", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(15). Value.ToString .Parameters.Add("@yx13", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(15).Value.ToString) .Parameters.Add("x_13", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(15).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x1 3", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx13", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_13", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cel ls(16).Value))) Then .Parameters.Add("@vrb_x1 4", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx14", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x1 4", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(16). Value.ToString .Parameters.Add("@yx14", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2). Value.ToString) * (dgv_save.Rows(i).Cells(16).Value.ToString) .Parameters.Add("x_14", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(16).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x1 4", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx14", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_14", SqlDbType.VarChar).Value = System.DBNull.Value End Try </pre>
--	--	--

<pre> Try If String.IsNullOrEmpty(Trim (CStr(dgv_save.Rows(i).Cells(17).Value))) Then .Parameters.Add("@vrb_x15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x15", SqlDbType.VarChar).Value = dgv_save.Rows(i).Cells(17).Value.ToString .Parameters.Add("@yx15", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(2).Value.ToString) * (dgv_save.Rows(i).Cells(17).Value.ToString) .Parameters.Add("x_15", SqlDbType.VarChar).Value = (dgv_save.Rows(i).Cells(17).Value.ToString) ^ 2 End If Catch ex As Exception .Parameters.Add("@vrb_x15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("x_15", SqlDbType.VarChar).Value = System.DBNull.Value End Try </pre>	<pre> .ExecuteNonQuery() End With Next show_save() sql = "SELECT * FROM VRB_cal with(nolock) where id_vrb =" & lbl_id.Text & "" With com .Parameters.Clear() .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql dr = .ExecuteReader End With If dr.HasRows Then dr.Read() dr.Close() up_cal() Else dr.Close() sql = "insert into VRB_cal " & _ (id_vrb,yx1,yx2,yx3,yx4,yx5,yx6,yx7,yx8,yx9,yx10,yx11,yx12,yx13,yx14,yx15 " & _ " ,xx1,xx2,xx3,xx4,xx5,xx6,xx7,xx8,xx9,xx10,xx11,xx12,xx13,xx14,xx15 " & _ " ,yy1,yy2,yy3,yy4,yy5,yy6,yy7,yy8,yy9,yy10,yy11,yy12,yy13,yy14,yy15 " & _ " ,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15 " & _ " ,b0_1,b0_2,b0_3,b0_4,b0_5,b0_6,b0_7,b0_8,b0_9,b0_10,b0_11,b0_12,b0_13,b0_14,b0_15)" & _ "values " & _ " (@id_vrb,@yx1,@yx2,@yx3,@yx4,@yx5,@yx6,@yx7,@yx8,@yx9,@yx10,@yx11 </pre>	<pre> ,@yx12,@yx13,@yx14,@yx15 " & _ " ,@xx1,@xx2,@xx3,@xx4,@xx5,@xx6,@xx7,@xx8,@xx9,@xx10,@xx11,@xx12,@xx13,@xx14,@xx15 " & _ " ,@yy1,@yy2,@yy3,@yy4,@yy5,@yy6,@yy7,@yy8,@yy9,@yy10,@yy11,@yy12,@yy13,@yy14,@yy15 " & _ " ,@b1,@b2,@b3,@b4,@b5,@b6,@b7,@b8,@b9,@b10,@b11,@b12,@b13,@b14,@b15 " & _ " ,@b0_1,@b0_2,@b0_3,@b0_4,@b0_5,@b0_6,@b0_7,@b0_8,@b0_9,@b0_10,@b0_11,@b0_12,@b0_13,@b0_14,@b0_15)" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() .Parameters.Add("@id_vrb", SqlDbType.VarChar).Value = lbl_id.Text If String.IsNullOrEmpty(Trim (CStr(yx1))) Then .Parameters.Add("@yx1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx1", SqlDbType.VarChar).Value = yx1 End If </pre>
--	---	--

<pre> If String.IsNullOrEmpty(Trim (CStr(yx2))) Then .Parameters.Add("@yx2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx2", SqlDbType.VarChar).Value = yx2 End If If String.IsNullOrEmpty(Trim (CStr(yx3))) Then .Parameters.Add("@yx3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx3", SqlDbType.VarChar).Value = yx3 End If If String.IsNullOrEmpty(Trim (CStr(yx4))) Then .Parameters.Add("@yx4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx4", SqlDbType.VarChar).Value = yx4 End If If String.IsNullOrEmpty(Trim (CStr(yx5))) Then .Parameters.Add("@yx5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx5", SqlDbType.VarChar).Value = yx5 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx6))) Then .Parameters.Add("@yx6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx6", SqlDbType.VarChar).Value = yx6 End If If String.IsNullOrEmpty(Trim (CStr(yx7))) Then .Parameters.Add("@yx7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx7", SqlDbType.VarChar).Value = yx7 End If If String.IsNullOrEmpty(Trim (CStr(yx8))) Then .Parameters.Add("@yx8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx8", SqlDbType.VarChar).Value = yx8 End If If String.IsNullOrEmpty(Trim (CStr(yx9))) Then .Parameters.Add("@yx9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx9", SqlDbType.VarChar).Value = yx9 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx10))) Then .Parameters.Add("@yx10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx10", SqlDbType.VarChar).Value = yx10 End If If String.IsNullOrEmpty(Trim (CStr(yx11))) Then .Parameters.Add("@yx11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx11", SqlDbType.VarChar).Value = yx11 End If If String.IsNullOrEmpty(Trim (CStr(yx12))) Then .Parameters.Add("@yx12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx12", SqlDbType.VarChar).Value = yx12 End If If String.IsNullOrEmpty(Trim (CStr(yx13))) Then .Parameters.Add("@yx13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx13", SqlDbType.VarChar).Value = yx13 End If </pre>
--	---	--

<pre> If String.IsNullOrEmpty(Trim (CStr(yx14))) Then .Parameters.Add("@yx14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx14", SqlDbType.VarChar).Value = yx14 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx3))) Then .Parameters.Add("@xx3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx3", SqlDbType.VarChar).Value = xx3 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx7))) Then .Parameters.Add("@xx7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx7", SqlDbType.VarChar).Value = xx7 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yx15))) Then .Parameters.Add("@yx15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx15", SqlDbType.VarChar).Value = yx15 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx4))) Then .Parameters.Add("@xx4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx4", SqlDbType.VarChar).Value = xx4 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx8))) Then .Parameters.Add("@xx8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx8", SqlDbType.VarChar).Value = xx8 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yx1))) Then .Parameters.Add("@xx1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx1", SqlDbType.VarChar).Value = xx1 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx5))) Then .Parameters.Add("@xx5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx5", SqlDbType.VarChar).Value = xx5 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx9))) Then .Parameters.Add("@xx9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx9", SqlDbType.VarChar).Value = xx9 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx2))) Then .Parameters.Add("@xx2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx2", SqlDbType.VarChar).Value = xx2 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx6))) Then .Parameters.Add("@xx6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx6", SqlDbType.VarChar).Value = xx6 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx10))) Then .Parameters.Add("@xx10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx10", SqlDbType.VarChar).Value = xx10 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(xx11))) Then .Parameters.Add("@xx11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx11", SqlDbType.VarChar).Value = xx11 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx15))) Then .Parameters.Add("@xx15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx15", SqlDbType.VarChar).Value = xx15 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy4))) Then .Parameters.Add("@yy4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy4", SqlDbType.VarChar).Value = yy4 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx12))) Then .Parameters.Add("@xx12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx12", SqlDbType.VarChar).Value = xx12 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx1))) Then .Parameters.Add("@yy1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy1", SqlDbType.VarChar).Value = yy1 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy5))) Then .Parameters.Add("@yy5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy5", SqlDbType.VarChar).Value = yy5 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx13))) Then .Parameters.Add("@xx13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx13", SqlDbType.VarChar).Value = xx13 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy2))) Then .Parameters.Add("@yy2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy2", SqlDbType.VarChar).Value = yy2 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy6))) Then .Parameters.Add("@yy6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy6", SqlDbType.VarChar).Value = yy6 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx14))) Then .Parameters.Add("@xx14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx14", SqlDbType.VarChar).Value = xx14 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy3))) Then .Parameters.Add("@yy3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy3", SqlDbType.VarChar).Value = yy3 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy7))) Then .Parameters.Add("@yy7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy7", SqlDbType.VarChar).Value = yy7 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(yy8))) Then .Parameters.Add("@yy8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy8", SqlDbType.VarChar).Value = yy8 End If If String.IsNullOrEmpty(Trim (CStr(yy9))) Then .Parameters.Add("@yy9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy9", SqlDbType.VarChar).Value = yy9 End If If String.IsNullOrEmpty(Trim (CStr(yy10))) Then .Parameters.Add("@yy10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy10", SqlDbType.VarChar).Value = yy10 End If If String.IsNullOrEmpty(Trim (CStr(yy11))) Then .Parameters.Add("@yy11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy11", SqlDbType.VarChar).Value = yy11 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy12))) Then .Parameters.Add("@yy12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy12", SqlDbType.VarChar).Value = yy12 End If If String.IsNullOrEmpty(Trim (CStr(yy13))) Then .Parameters.Add("@yy13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy13", SqlDbType.VarChar).Value = yy13 End If If String.IsNullOrEmpty(Trim (CStr(yy14))) Then .Parameters.Add("@yy14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy14", SqlDbType.VarChar).Value = yy14 End If If String.IsNullOrEmpty(Trim (CStr(yy15))) Then .Parameters.Add("@yy15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy15", SqlDbType.VarChar).Value = yy15 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb1))) Then .Parameters.Add("@b1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b1", SqlDbType.VarChar).Value = bb1 End If If String.IsNullOrEmpty(Trim (CStr(bb2))) Then .Parameters.Add("@b2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b2", SqlDbType.VarChar).Value = bb2 End If If String.IsNullOrEmpty(Trim (CStr(bb3))) Then .Parameters.Add("@b3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b3", SqlDbType.VarChar).Value = bb3 End If If String.IsNullOrEmpty(Trim (CStr(bb4))) Then .Parameters.Add("@b4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b4", SqlDbType.VarChar).Value = bb4 End If </pre>
--	--	--

<pre> If String.IsNullOrEmpty(Trim (CStr(bb5))) Then .Parameters.Add("@b5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b5", SqlDbType.VarChar).Value = bb5 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb9))) Then .Parameters.Add("@b9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b9", SqlDbType.VarChar).Value = bb9 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb13))) Then .Parameters.Add("@b13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b13", SqlDbType.VarChar).Value = bb13 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb6))) Then .Parameters.Add("@b6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b6", SqlDbType.VarChar).Value = bb6 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb10))) Then .Parameters.Add("@b10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b10", SqlDbType.VarChar).Value = bb10 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb14))) Then .Parameters.Add("@b14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b14", SqlDbType.VarChar).Value = bb14 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb7))) Then .Parameters.Add("@b7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b7", SqlDbType.VarChar).Value = bb7 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb11))) Then .Parameters.Add("@b11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b11", SqlDbType.VarChar).Value = bb11 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb15))) Then .Parameters.Add("@b15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b15", SqlDbType.VarChar).Value = bb15 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb8))) Then .Parameters.Add("@b8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b8", SqlDbType.VarChar).Value = bb8 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb12))) Then .Parameters.Add("@b12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b12", SqlDbType.VarChar).Value = bb12 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb0_1))) Then .Parameters.Add("@b0_1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_1", SqlDbType.VarChar).Value = bb0_1 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(bb0_2))) Then .Parameters.Add("@b0_2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_2", SqlDbType.VarChar).Value = bb0_2 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb6))) Then .Parameters.Add("@b0_6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_6", SqlDbType.VarChar).Value = bb0_6 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb10))) Then .Parameters.Add("@b0_10" , SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_10" , SqlDbType.VarChar).Value = bb0_10 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb3))) Then .Parameters.Add("@b0_3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_3", SqlDbType.VarChar).Value = bb0_3 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb7))) Then .Parameters.Add("@b0_7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_7", SqlDbType.VarChar).Value = bb0_7 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb11))) Then .Parameters.Add("@b0_11" , SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_11" , SqlDbType.VarChar).Value = bb0_11 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb4))) Then .Parameters.Add("@b0_4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_4", SqlDbType.VarChar).Value = bb0_4 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb8))) Then .Parameters.Add("@b0_8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_8", SqlDbType.VarChar).Value = bb0_8 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb12))) Then .Parameters.Add("@b0_12" , SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_12" , SqlDbType.VarChar).Value = bb0_12 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb5))) Then .Parameters.Add("@b0_5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_5", SqlDbType.VarChar).Value = bb0_5 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb9))) Then .Parameters.Add("@b0_9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_9", SqlDbType.VarChar).Value = bb0_9 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb13))) Then .Parameters.Add("@b0_13" , SqlDbType.VarChar).Value = System.DBNull.Value Else </pre>

```

.Parameters.Add("@b0_13"
,
SqlDbType.VarChar).Value
= bb0_13
    End If

    If
String.IsNullOrEmpty(Trim
(CStr(bb14))) Then

.Parameters.Add("@b0_14"
,
SqlDbType.VarChar).Value
= System.DBNull.Value
    Else

.Parameters.Add("@b0_14"
,
SqlDbType.VarChar).Value
= bb0_14
    End If
    If
String.IsNullOrEmpty(Trim
(CStr(bb15))) Then

.Parameters.Add("@b0_15"
,
SqlDbType.VarChar).Value
= System.DBNull.Value
    Else

.Parameters.Add("@b0_15"
,
SqlDbType.VarChar).Value
= bb0_15
    End If

.ExecuteNonQuery()
    End With
    End If

    For i = 1 To maxs
        sql = "SELECT
distinct MAX(r_no) AS
maxr_no FROM VRB_x1
where
substring(r_no,1,4)=" &
lbl_id.Text & ""
    Try
        With com
            .CommandType
= CommandType.Text
            .CommandText
= sql
            .Connection =
conn1
            dr =
            .ExecuteReader()
            dr.Read()
            max_r_no =
CInt(dr.Item("maxr_no"))
            max_r_no =
(CInt(max_r_no) +
"01").ToString("00000")
            End With
        Catch
            max_r_no =
lbl_id.Text & "01"
        End Try
        dr.Close()
        sql = "select case
when d.[1] is null then'0'else
d.[1] end as'1', " & _
            "case when
d.[2] is null then'0'else d.[2]
end as'2', " & _
            "case when
d.[3] is null then'0'else d.[3]
end as'3', " & _
            "case when
d.[4] is null then'0'else d.[4]
end as'4', " & _
            "case when
d.[5] is null then'0'else d.[5]
end as'5', " & _
            "case when
d.[6] is null then'0'else d.[6]
end as'6', " & _
            "case when
d.[7] is null then'0'else d.[7]
end as'7', " & _
            "case when
d.[8] is null then'0'else d.[8]
end as'8', " & _
            "case when
d.[9] is null then'0'else d.[9]
end as'9', " & _
            "case when
d.[10] is null then'0'else
d.[10] end as'10', " & _
            "case when
d.[11] is null then'0'else
d.[11] end as'11', " & _
            "case when
d.[12] is null then'0'else
d.[12] end as'12', " & _
            "case when
d.[13] is null then'0'else
d.[13] end as'13', " & _
            "case when
d.[14] is null then'0'else
d.[14] end as'14', " & _
            "case when
d.[15] is null then'0'else
d.[15] end as'15', " & _
            "case when
d.[16] is null then'0'else
d.[16] end as'16', " & _
            "case when
d.[17] is null then'0'else
d.[17] end as'17', " & _
            "case when
d.[18] is null then'0'else
d.[18] end as'18', " & _
            "case when
d.[19] is null then'0'else
d.[19] end as'19', " & _
            "case when
d.[20] is null then'0'else
d.[20] end as'20', " & _
            "case when
d.[21] is null then'0'else
d.[21] end as'21', " & _
            "case when
d.[22] is null then'0'else
d.[22] end as'22', " & _
            "case when
d.[23] is null then'0'else
d.[23] end as'23', " & _
            "case when
d.[24] is null then'0'else
d.[24] end as'24', " & _
            "case when
d.[25] is null then'0'else
d.[25] end as'25', " & _
            "case when
d.[26] is null then'0'else
d.[26] end as'26', " & _
            "case when
d.[27] is null then'0'else
d.[27] end as'27', " & _
            "case when
d.[28] is null then'0'else
d.[28] end as'28', " & _
            "case when
d.[29] is null then'0'else
d.[29] end as'29', " & _
            "case when
d.[30] is null then'0'else
d.[30] end as'30' from " & _
            " (select
row_number() over (order
by id_no)as num
,convert(float,vrb_x" & i &
")as ab from
VRB_VARIABLE where

```

<pre> id_vrb= "" & lbl_id.Text & "" as aa " & _ " PIVOT (sum(ab)FOR num in ([1],[2],[3],[4],[5],[6],[7],[8] ,[9],[10],[11],[12],[13],[14], [15],[16],[17],[18],[19],[20] , " & _ " [21],[22],[23],[24],[25],[26] ,[27],[28],[29],[30])) as d" With com .CommandType = CommandType.Text .CommandText = sql .Connection = conn1 dr = .ExecuteReader() dr.Read() xr1 = dr.Item("1") : xr2 = dr.Item("2") : xr3 = dr.Item("3") : xr4 = dr.Item("4") : xr5 = dr.Item("5") xr6 = dr.Item("6") : xr7 = dr.Item("7") : xr8 = dr.Item("8") : xr9 = dr.Item("9") : xr10 = dr.Item("10") xr11 = dr.Item("11") : xr12 = dr.Item("12") : xr13 = dr.Item("13") : xr14 = dr.Item("14") : xr15 = dr.Item("15") xr16 = dr.Item("16") : xr17 = dr.Item("17") : xr18 = dr.Item("18") : xr19 = dr.Item("19") : xr20 = dr.Item("20") xr21 = dr.Item("21") : xr22 = dr.Item("22") : xr23 = dr.Item("23") : xr24 = dr.Item("24") : xr25 = dr.Item("25") xr26 = dr.Item("26") : xr27 = dr.Item("27") : xr28 = dr.Item("28") : xr29 = dr.Item("29") : xr30 = dr.Item("30") End With </pre>	<pre> dr.Close() sql = "SELECT distinct count(r_no)as r_no FROM VRB_x1 with(nolock) where substring(r_no,1,4)="" & lbl_id.Text & "" With com .Parameters.Clear() .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql dr = .ExecuteReader() dr.Read() If dr.Item("r_no") = maxs Then dr.Close() sql = "update VRB_x1 set x1="" & xr1 & "" ,x2="" & xr2 & "" ,x3="" & xr3 & "" ,x4="" & xr4 & "" ,x5="" & xr5 & "" ,x6="" & xr6 & "" ,x7="" & xr7 & "" , & _ " x8="" & xr8 & "" ,x9="" & xr9 & "" ,x10="" & xr10 & "" ,x11="" & xr11 & "" ,x12="" & xr12 & "" ,x13="" & xr13 & "" ,x14="" & xr14 & "" ,x15="" & xr15 & "" , " & _ " x16="" & xr16 & "" ,x17="" & xr17 & "" ,x18="" & xr18 & "" ,x19="" & xr19 & "" ,x20="" & xr20 & "" ,x21="" & xr21 & "" ,x22="" & xr22 & "" ,x23="" & xr23 & "" , " & _ " x24="" & xr24 & "" ,x25="" & xr25 & "" ,x26="" & xr26 & "" ,x27="" & xr27 & "" ,x28="" & xr28 & "" ,x29="" & xr29 & "" ,x30="" & xr30 & "" "" & _ " where r_no="" & lbl_id.Text & i & "" With com .Connection = conn1 </pre>	<pre> .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With Else dr.Close() sql = "insert into VRB_x1" & _ " (r_no,x1,x2,x3,x4,x5,x6,x7, x8,x9,x10,x11,x12,x13,x14, x15,x16,x17,x18,x19,x20 " & _ " ,x21,x22,x23,x24,x25,x26,x 27,x28,x29,x30)" & _ " values " & - " (" & max_r_no & "" ,"" & xr1 & "" ,"" & xr2 & "" ,"" & xr3 & "" ,"" & xr4 & "" ,"" & xr5 & "" ,"" & xr6 & "" ,"" & xr7 & "" " & _ " ,"" & xr8 & "" ,"" & xr9 & "" ,"" & xr10 & "" ,"" & xr11 & "" ,"" & xr12 & "" ,"" & xr13 & "" ,"" & xr14 & "" ,"" & xr15 & "" ,"" & xr16 & "" " & _ " ,"" & xr17 & "" ,"" & xr18 & "" ,"" & xr19 & "" ,"" & xr20 & "" ,"" & xr21 & "" ,"" & xr22 & "" ,"" & xr23 & "" ,"" & xr24 & "" ,"" & xr25 & "" " & _ " ,"" & xr26 & "" ,"" & xr27 & "" ,"" & xr28 & "" ,"" & xr29 & "" ,"" & xr30 & "")" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With </pre>
--	---	--

<pre> End If End With dst.Clear() sql = "select * from (select distinct " & _ "case when((convert(float,c.yx1) ='0') or (convert(float,c.b1='0'))the n('0')else(convert(float,c.b1) *convert(float,c.yx1)/ convert(float,c.yy1))end as r1, " & _ "case when((convert(float,c.yx2) ='0') or (convert(float,c.b2='0'))the n('0')else(convert(float,c.b2) *convert(float,c.yx2)/ convert(float,c.yy2))end as r2, " & _ "case when((convert(float,c.yx3) ='0') or (convert(float,c.b3='0'))the n('0')else(convert(float,c.b3) *convert(float,c.yx3)/ convert(float,c.yy3))end as r3, " & _ "case when((convert(float,c.yx4) ='0') or (convert(float,c.b4='0'))the n('0')else(convert(float,c.b4) *convert(float,c.yx4)/ convert(float,c.yy4))end as r4, " & _ "case when((convert(float,c.yx5) ='0') or (convert(float,c.b5='0'))the n('0')else(convert(float,c.b5) *convert(float,c.yx5)/ convert(float,c.yy5))end as r5, " & _ "case when((convert(float,c.yx6) ='0') or (convert(float,c.b6='0'))the n('0')else(convert(float,c.b6) *convert(float,c.yx6)/ convert(float,c.yy6))end as r6, " & _ </pre>	<pre> "case when((convert(float,c.yx7) ='0') or (convert(float,c.b7='0'))the n('0')else(convert(float,c.b7) *convert(float,c.yx7)/ convert(float,c.yy7))end as r7, " & _ "case when((convert(float,c.yx8) ='0') or (convert(float,c.b8='0'))the n('0')else(convert(float,c.b8) *convert(float,c.yx8)/ convert(float,c.yy8))end as r8, " & _ "case when((convert(float,c.yx9) ='0') or (convert(float,c.b9='0'))the n('0')else(convert(float,c.b9) *convert(float,c.yx9)/ convert(float,c.yy9))end as r9, " & _ "case when((convert(float,c.yx10) ='0') or (convert(float,c.b10='0'))th en('0')else(convert(float,c.b1 0)*convert(float,c.yx10)/ convert(float,c.yy10))end as r10, " & _ "case when((convert(float,c.yx11) ='0') or (convert(float,c.b11='0'))th en('0')else(convert(float,c.b1 1)*convert(float,c.yx11)/ convert(float,c.yy11))end as r11, " & _ "case when((convert(float,c.yx12) ='0') or (convert(float,c.b12='0'))th en('0')else(convert(float,c.b1 2)*convert(float,c.yx12)/ convert(float,c.yy12))end as r12, " & _ "case when((convert(float,c.yx13) ='0') or (convert(float,c.b13='0'))th en('0')else(convert(float,c.b1 3)*convert(float,c.yx13)/ convert(float,c.yy13))end as r13, " & _ </pre>	<pre> "case when((convert(float,c.yx14) ='0') or (convert(float,c.b14='0'))th en('0')else(convert(float,c.b1 4)*convert(float,c.yx14)/ convert(float,c.yy14))end as r14, " & _ "case when((convert(float,c.yx15) ='0') or (convert(float,c.b15='0'))th en('0')else(convert(float,c.b1 5)*convert(float,c.yx15)/ convert(float,c.yy15))end as r15 " & _ "from VRB_cal c left join VRB_VARIABLE v with(nolock) on c.id_vrb=v.id_vrb where c.id_vrb = " & lbl_id.Text & ")datatable " & _ "UNPIVOT (r for names in(r1,r2,r3,r4,r5,r6,r7,r8,r9,r 10,r11,r12,r13,r14,r15) as vv" da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(dst, "vv") DataGridView5.DataSource = dst.Tables("vv") dsf.Clear() sql = " select id_no,r_no from VRB_x1 where substring (r_no,1,4)=" & lbl_id.Text & " order by id_no asc " da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(dsf, "r_no") DataGridView6.DataSource = dsf.Tables("r_no") sql = "SELECT distinct count(r_no)as r_no FROM VRB_r with(nolock) where substring(r_no,1,4)=" & lbl_id.Text & "" With com </pre>
---	---	---

.Parameters.Clear() .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql dr = .ExecuteReader() dr.Read() If dr.Item("r_no") = maxs Then dr.Close() r = r + 1 r = 0 & r sql = "update VRB_r set r=@r where r_no= " & lbl_id.Text & r & "" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() n = i - 1 If .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() n = i - 1 If String.IsNullOrEmpty(Trim (CStr(DataGridView6.Rows (i - 1).Cells(1).Value))) Then .Parameters.Add("@r", SqlCommandType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@r", SqlCommandType.VarChar).Value = DataGridView5.Rows(i - 1).Cells(0).Value.ToString End If .ExecuteNonQuery() End With Else	dr.Close() sql = "insert into VRB_r (r_no,r)values (@r_no,@r)" Dim aa, bb As String aa = DataGridView6.Rows(i - 1).Cells(1).Value.ToString bb = DataGridView5.Rows(i - 1).Cells(0).Value.ToString With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() n = i - 1 If String.IsNullOrEmpty(Trim (CStr(DataGridView6.Rows (i - 1).Cells(1).Value))) Then .Parameters.Add("@r_no", SqlCommandType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@r", SqlCommandType.VarChar).Value = DataGridView6.Rows(i - 1).Cells(1).Value.ToString .Parameters.Add("@r", SqlCommandType.VarChar).Value = DataGridView5.Rows(i - 1).Cells(0).Value.ToString End If .ExecuteNonQuery() End With End If End With	Next show_data_view() dgv_save.Rows.Clear() MsgBox("Complete....") End Sub Protected Sub show_save() yx1 = "" : yx2 = "" : yx3 = "" : yx4 = "" : yx5 = "" : yx6 = "" : yx7 = "" : yx8 = "" : yx9 = "" : yx10 = "" : yx11 = "" : yx12 = "" : yx13 = "" : yx14 = "" : yx15 = "" yy1 = "" : yy2 = "" : yy3 = "" : yy4 = "" : yy5 = "" : yy6 = "" : yy7 = "" : yy8 = "" : yy9 = "" : yy10 = "" : yy11 = "" : yy12 = "" : yy13 = "" : yy14 = "" : yy15 = "" bb1 = "" : bb2 = "" : bb3 = "" : bb4 = "" : bb5 = "" : bb6 = "" : bb7 = "" : bb8 = "" : bb9 = "" : bb10 = "" : bb11 = "" : bb12 = "" : yx13 = "" : bb14 = "" : bb15 = "" bb0_1 = "" : bb0_2 = "" : bb0_3 = "" : bb0_4 = "" : bb0_5 = "" : bb0_6 = "" : bb0_7 = "" : bb0_8 = "" : bb0_9 = "" : bb0_10 = "" : bb0_11 = "" : bb0_12 = "" : bb0_13 = "" : bb0_14 = "" : bb0_15 = "" For j = 1 To maxs sql = "select case when(aa.ssyx is null)then(0)else(aa.ssyx)end as ssyx from (" & _ "select round((sum(convert(float,yx " & j & "))),6) - round((((sum(convert(float, vr_b_y))))*(sum(convert(float ,vr_b_x" & j & ")))) / " & _ "count(vr_b_y)),6) as ssyx " & _ " from VRB_VARIABLE where id_vrb=" & lbl_id.Text & ") as aa"
---	--	---

```

.CommandType =          yx7 = ""          End If
CommandType.Text        End If          End If
.CommandText =          ElseIf j = 8 Then
sql                      yx8 =          End With
.Connection =          dr.Item("ssyx")        dr.Close()
conn1
  dr =                  Then
.ExecuteReader()        yx8 = ""
  dr.Read()             End If
                        ElseIf j = 9 Then
                        yx9 =
dr.Item("ssyx")         dr.Item("ssyx")        If yx9 = "0"
                        Then
Then                    yx9 = ""
                        End If
                        ElseIf j = 10 Then
                        yx10 =
dr.Item("ssyx")         dr.Item("ssyx")        If yx10 = "0"
                        Then
Then                    yx10 = ""
                        End If
                        ElseIf j = 11 Then
                        yx11 =
dr.Item("ssyx")         dr.Item("ssyx")        If yx11 = "0"
                        Then
Then                    yx11 = ""
                        End If
                        ElseIf j = 12 Then
dr.Item("ssyx")         dr.Item("ssyx")        If yx12 = "0"
                        Then
Then                    yx12 = ""
                        End If
                        ElseIf j = 13 Then
dr.Item("ssyx")         dr.Item("ssyx")        yx13 =
                        If yx13 = "0"
Then                    Then
                        yx13 = ""
                        End If
                        ElseIf j = 14 Then
dr.Item("ssyx")         dr.Item("ssyx")        yx14 =
                        If yx14 = "0"
Then                    Then
                        yx14 = ""
                        End If
                        ElseIf j = 15 Then
dr.Item("ssyx")         dr.Item("ssyx")        yx15 =
                        If yx15 = "0"
Then                    Then
Then                    yx15 = ""

```

```

sql = "select case
when(aa.sxxx is
null)then(0)else(aa.sxxx)end
as sxxx from ( " & _
" select
round(sum(convert
(float,x_" & j & ")),6) -
round(((POWER
(sum(convert(float,vrb_x"
& j & ")),2) / " & _
" count(vrb_y
),6) as sxxx " & _
" from
VRB_VARIABLE where
id_vrb = " & lbl_id.Text &
" )as aa"

With com
.CommandType =
CommandType.Text
.CommandText =
sql
.Connection =
conn1
  dr =
.ExecuteReader()
  dr.Read()
  If j = 1 Then
xx1 =
dr.Item("sxxx")
  ElseIf j = 2 Then
xx2 =
dr.Item("sxxx")
  ElseIf j = 3 Then
xx3 =
dr.Item("sxxx")
  ElseIf j = 4 Then
xx4 =
dr.Item("sxxx")
  ElseIf j = 5 Then
xx5 =
dr.Item("sxxx")
  ElseIf j = 6 Then
xx6 =
dr.Item("sxxx")
  ElseIf j = 7 Then
xx7 =
dr.Item("sxxx")
  ElseIf j = 8 Then
xx8 =
dr.Item("sxxx")

```



```

ElseIf j = 9 Then
    xx9 =
dr.Item("ssxx")
    ElseIf j = 10 Then
        xx10 =
dr.Item("ssxx")
    ElseIf j = 11 Then
        xx11 =
dr.Item("ssxx")
    ElseIf j = 12 Then
        xx12 =
dr.Item("ssxx")
    ElseIf j = 13 Then
        xx13 =
dr.Item("ssxx")
    ElseIf j = 14 Then
        xx14 =
dr.Item("ssxx")
    ElseIf j = 15 Then
        xx15 =
dr.Item("ssxx")
    End If

End With
dr.Close()

sql = "select case
when(aa.ssyy is
null)then(0)else(aa.ssyy)end
as ssyy from ( " & _
" select
round((sum(convert(float,y_
1))),6) - round(((POWER
(sum(convert(float,vrb_y)),
2))/count(vrb_y) ),6) as ssyy
" & _
" from
VRB_VARIABLE where
id_vrb ='" & lbl_id.Text &
'" )as aa"

With com
.CommandType =
CommandType.Text
.CommandText =
sql
.Connection =
conn1
dr =
.ExecuteReader()
dr.Read()
If j = 1 Then
    yy1 =
dr.Item("ssyy")
    ElseIf j = 2 Then
        yy2 =
dr.Item("ssyy")
    ElseIf j = 3 Then
        yy3 =
dr.Item("ssyy")
    ElseIf j = 4 Then
        yy4 =
dr.Item("ssyy")
    ElseIf j = 5 Then
        yy5 =
dr.Item("ssyy")
    ElseIf j = 6 Then
        yy6 =
dr.Item("ssyy")
    ElseIf j = 7 Then
        yy7 =
dr.Item("ssyy")
    ElseIf j = 8 Then
        yy8 =
dr.Item("ssyy")
    ElseIf j = 9 Then
        yy9 =
dr.Item("ssyy")
    ElseIf j = 10 Then
        yy10 =
dr.Item("ssyy")
    ElseIf j = 11 Then
        yy11 =
dr.Item("ssyy")
    ElseIf j = 12 Then
        yy12 =
dr.Item("ssyy")
    ElseIf j = 13 Then
        yy13 =
dr.Item("ssyy")
    ElseIf j = 14 Then
        yy14 =
dr.Item("ssyy")
    ElseIf j = 15 Then
        yy15 =
dr.Item("ssyy")
    End If
End With
dr.Close()

'B1
sql = "select " & _
"round((sum(convert(float,y
x" & j & ")),6) - " & _
"round((((sum(Convert(float
, vrb_y))) *
(sum(Convert(float, vrb_x"
& j & ")))) / " & _
"count(vrb_y)),6) as ab , "
& _
"round(sum(convert(float,x
_ " & j & ")),6) - " & _
"round((power((sum(conver
t(float,vrb_x" & j & "))),2)/
count(vrb_y)),6) as ab1 " &
_
"from
VRB_VARIABLE where
id_vrb ='" & lbl_id.Text &
'" "

With com
.CommandType =
CommandType.Text
.CommandText =
sql
.Connection =
conn1
dr =
.ExecuteReader()
dr.Read()

Dim a_b As
String
Try
    a_b =
dr.Item("ab") /
dr.Item("ab1")

Catch ex As
Exception
    str = "Data
Deficient."
    MsgBox("Data
Deficient.")
End Try
If (a_b = "NaN")
Or (a_b = "-NaN") Or (a_b
= "Infinity") Or (a_b = "-
Infinity") Or (a_b = "")
Then
    a_b = "0"
End If

If j = 1 Then
    bb1 = a_b
    bb2 = "0"
    bb3 = "0"
    bb4 = "0"
    bb5 = "0"
    bb6 = "0"
    bb7 = "0"
    bb8 = "0"
    bb9 = "0"
    bb10 = "0"

```

```

bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 2 Then
bb2 = a_b
bb3 = "0"
bb4 = "0"
bb5 = "0"
bb6 = "0"
bb7 = "0"
bb8 = "0"
bb9 = "0"
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 3 Then
bb3 = a_b
bb4 = "0"
bb5 = "0"
bb6 = "0"
bb7 = "0"
bb8 = "0"
bb9 = "0"
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 4 Then
bb4 = a_b
bb5 = "0"
bb6 = "0"
bb7 = "0"
bb8 = "0"
bb9 = "0"
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 5 Then
bb5 = a_b
bb6 = "0"
bb7 = "0"
bb8 = "0"
bb9 = "0"
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 6 Then
bb6 = a_b
bb7 = "0"
bb8 = "0"
bb9 = "0"
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 7 Then
bb7 = a_b
bb8 = "0"
bb9 = "0"
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 8 Then
bb8 = a_b
bb9 = "0"
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 9 Then
bb9 = a_b
bb10 = "0"
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 10 Then
bb10 = a_b
bb11 = "0"
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 11 Then
bb11 = a_b
bb12 = "0"
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 12 Then
bb12 = a_b
bb13 = "0"
bb14 = "0"
bb15 = "0"
ElseIf j = 13 Then
bb13 = a_b
bb14 = "0"
bb15 = "0"
ElseIf j = 14 Then
bb14 = a_b
bb15 = "0"
ElseIf j = 15 Then
bb15 = a_b
End If
End With
dr.Close()
Next

sql = " select " & _
"case
when(convert(float," & bb1
& ")=0)then(0)else(
(sum(convert(float,v.vrb_y)
)/count(v.vrb_y))-
((convert(float," & bb1 &
"))*(sum(convert(float,v.vrb
_x1)/count(v.vrb_x1)
)))end as Bb0_1, " & _
"case
when(convert(float," & bb2
&
")=0)then(0)else((sum(conv
ert(float,v.vrb_y)/count(v.v
rb_y)- ((convert(float," &
bb2 &
"))*(sum(convert(float,v.vrb
_x2)/count(v.vrb_x2)
))))end as Bb0_2, " & _
"case
when(convert(float," & bb3
&
")=0)then(0)else((sum(conv
ert(float,v.vrb_y)/count(v.v
rb_y)- ((convert(float," &
bb3 &
"))*(sum(convert(float,v.vrb
_x3)/count(v.vrb_x3)
))))end as Bb0_3, " & _
"case
when(convert(float," & bb4
&
")=0)then(0)else((sum(conv
ert(float,v.vrb_y)/count(v.v
rb_y)- ((convert(float," &
bb4 &
"))*(sum(convert(float,v.vrb
_x4)/count(v.vrb_x4)
))))end as Bb0_4, " & _

```

<pre> "case when(convert(float," & bb5 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb5 & "))*(sum(convert(float,v.vrb _x5))/count(v.vrb_x5))))end as Bb0_5, " & _ "case when(convert(float," & bb6 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb6 & "))*(sum(convert(float,v.vrb _x6))/count(v.vrb_x6))))end as Bb0_6, " & _ "case when(convert(float," & bb7 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb7 & "))*(sum(convert(float,v.vrb _x7))/count(v.vrb_x7))))end as Bb0_7, " & _ "case when(convert(float," & bb8 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb8 & "))*(sum(convert(float,v.vrb _x8))/count(v.vrb_x8))))end as Bb0_8, " & _ "case when(convert(float," & bb9 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb9 & "))*(sum(convert(float,v.vrb _x9))/count(v.vrb_x9))))end as Bb0_9, " & _ "case when(convert(float," & bb10 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb10 & </pre>	<pre> "))*(sum(convert(float,v.vrb _x10))/count(v.vrb_x10))))end as Bb0_10, " & _ "case when(convert(float," & bb11 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb11 & "))*(sum(convert(float,v.vrb _x11))/count(v.vrb_x11))))end as Bb0_11, " & _ "case when(convert(float," & bb12 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb12 & "))*(sum(convert(float,v.vrb _x12))/count(v.vrb_x12))))end as Bb0_12, " & _ "case when(convert(float," & bb13 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb13 & "))*(sum(convert(float,v.vrb _x13))/count(v.vrb_x13))))end as Bb0_13, " & _ "case when(convert(float," & bb14 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb14 & "))*(sum(convert(float,v.vrb _x14))/count(v.vrb_x14))))end as Bb0_14, " & _ "case when(convert(float," & bb15 & ")=0)then(0)else((sum(conv ert(float,v.vrb_y))/count(v.v rb_y))- ((convert(float," & bb15 & "))*(sum(convert(float,v.vrb _x15))/count(v.vrb_x15))))end as Bb0_15 " & _ "from VRB_VARIABLE v with(nolock) left join VRB_cal c with(nolock) on </pre>	<pre> v.id_vrb=c.id_vrb where v.id_vrb = "" & lbl_id.Text & """" With com .CommandType = CommandType.Text .CommandText = sql .Connection = conn1 dr = .ExecuteReader() dr.Read() Try bb0_1 = dr.Item("Bb0_1") Catch ex As Exception bb0_1 = "0" End Try Try bb0_2 = dr.Item("Bb0_2") Catch ex As Exception bb0_2 = "0" End Try Try bb0_3 = dr.Item("Bb0_3") Catch ex As Exception bb0_3 = "0" End Try Try bb0_4 = dr.Item("Bb0_4") Catch ex As Exception bb0_4 = "0" End Try Try bb0_5 = dr.Item("Bb0_5") Catch ex As Exception bb0_5 = "0" End Try Try bb0_6 = dr.Item("Bb0_6") Catch ex As Exception bb0_6 = "0" End Try Try bb0_7 = dr.Item("Bb0_7") </pre>
--	---	--

<pre> Catch ex As Exception bb0_7 = "0" End Try Try bb0_8 = dr.Item("Bb0_8") Catch ex As Exception bb0_8 = "0" End Try Try bb0_9 = dr.Item("Bb0_9") Catch ex As Exception bb0_9 = "0" End Try Try bb0_10 = dr.Item("Bb0_10") Catch ex As Exception bb0_10 = "0" End Try Try bb0_11 = dr.Item("Bb0_11") Catch ex As Exception bb0_11 = "0" End Try Try bb0_12 = dr.Item("Bb0_12") Catch ex As Exception bb0_12 = "0" End Try Try bb0_13 = dr.Item("Bb0_13") Catch ex As Exception bb0_13 = "0" End Try Try bb0_14 = dr.Item("Bb0_14") Catch ex As Exception bb0_14 = "0" End Try Try bb0_15 = dr.Item("Bb0_15") </pre>	<pre> Catch ex As Exception bb0_15 = "0" End Try End With dr.Close() End Sub Private Sub PictureBox8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox8.Click GroupBox1.Visible = False Exit Sub End Sub Private Sub PictureBox6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox6.Click GroupBox2.Visible = False Exit Sub End Sub Private Sub PictureBox7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox7.Click GroupBox1.Visible = False lbl_name.Focus() If String.IsNullOrEmpty(Trim (CStr(dgv_db.Rows(0).Cell s(1).Value))) Then Exit Sub End If For j = 0 To dgv_db.Rows.Count - 2 Dim sql1 As String = "insert into VRB_X (id_vrb,vrb_name,vrb_y,vrb _x,vrb_x_unit)values(@id_ vrb,@vrb_name,@vrb_y,@ vrb_x,@vrb_x_unit)" With com </pre>	<pre> .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql1 .Parameters.Clear() .Parameters.Add("@id_vrb" , SqlDbType.VarChar).Value = lbl_id.Text.ToString .Parameters.Add("@vrb_na me", SqlDbType.VarChar).Value = lbl_name.Text.ToString .Parameters.Add("@vrb_y", SqlDbType.VarChar).Value = lbl_y.Text If String.IsNullOrEmpty(Trim (CStr(dgv_db.Rows(j).Cells (1).Value))) Then .Parameters.Add("@vrb_x", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x", SqlDbType.VarChar).Value = dgv_db.Rows(j).Cells(1).Va lue.ToString End If If String.IsNullOrEmpty(Trim (CStr(dgv_db.Rows(j).Cells (2).Value))) Then .Parameters.Add("@vrb_x_ unit", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x_ unit", SqlDbType.VarChar).Value = dgv_db.Rows(j).Cells(2).Va lue.ToString </pre>
---	---	---

End If	convert(float,c.yx6)) / convert(float,c.yy6),6) as r6, " & _	da.SelectCommand = New SqlCommand(sql, conn1)
.ExecuteNonQuery() End With	"round((convert(float,c.b7)* convert(float,c.yx7)) / convert(float,c.yy7),6) as r7, " & _	da.Fill(dsr, "RR") dgv_R.DataSource = dsr.Tables("RR")
Next show_data_view() show_head() End Sub	"round((convert(float,c.b8)* convert(float,c.yx8)) / convert(float,c.yy8),6) as r8, " & _	dst.Clear() sql = "select " & _ "round(case
Private Sub pt_report_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pt_report.Click	"round((convert(float,c.b9)* convert(float,c.yx9)) / convert(float,c.yy9),6) as r9, " & _	when(((1- ((convert(float,c.b1) * convert(float,c.yx1))/conver t(float,c.yy1))))<'0') or (((convert(float,c.b1) " & _
If lbl_id.Text = "" Then	"round((convert(float,c.b10) *convert(float,c.yx10)) / convert(float,c.yy10),6) as r10, " & _	_* convert(float,c.yx1))/conver t(float,c.yy1))<'0'))then('0') else(" & _
Exit Sub End If GroupBox2.Visible = True	"round((convert(float,c.b11) *convert(float,c.yx11)) / convert(float,c.yy11),6) as r11, " & _	" case when(power(((1- (((convert(float,c.b1) * convert(float,c.yx1))/conver t(float,c.yy1))))),0.5) <= 0) then (0) " & _
show_data_view() Try	"round((convert(float,c.b12) *convert(float,c.yx12)) / convert(float,c.yy12),6) as r12, " & _	" else ((power(((convert(float,c.b1)* convert(float,c.yx1))/conver t(float,c.yy1)),0.5) " & _
dsr.Clear() sql = "select distinct " & _	"round((convert(float,c.b13) *convert(float,c.yx13)) / convert(float,c.yy13),6) as r13, " & _	" * power(convert(flo at,(count(v.vrb_y)-2)),0.5)) " & _
"round((convert(float,c.b1)* convert(float,c.yx1)) / convert(float,c.yy1),6) as r1, " & _	"round((convert(float,c.b14) *convert(float,c.yx14)) / convert(float,c.yy14),6) as r14, " & _	" /power(((1- (((convert(float,c.b1) * convert(float,c.yx1))/conver t(float,c.yy1))))),0.5) end end ,6) as T1 " & _
"round((convert(float,c.b2)* convert(float,c.yx2)) / convert(float,c.yy2),6) as r2, " & _	"round((convert(float,c.b15) *convert(float,c.yx15)) / convert(float,c.yy15),6) as r15 " & _	" ,round(case when(((1- ((convert(float,c.b2) * convert(float,c.yx2))/conver t(float,c.yy2))))<'0') or (((convert(float,c.b2) " & _
"round((convert(float,c.b3)* convert(float,c.yx3)) / convert(float,c.yy3),6) as r3, " & _	"from VRB_cal c left join VRB_VARIABLE v " & _	" * convert(float,c.yx2))/conver t(float,c.yy2))<'0'))then('0') else(" & _
"round((convert(float,c.b4)* convert(float,c.yx4)) / convert(float,c.yy4),6) as r4, " & _	"with(nolock) on c.id_vrb=v.id_vrb where c.id_vrb = " & lbl_id.Text.ToString & " " & _	" case when(power(((1- (((convert(float,c.b2) * convert(float,c.yx2))/conver t(float,c.yy2))<'0'))<'0'))then('0') else(" & _
"round((convert(float,c.b5)* convert(float,c.yx5)) / convert(float,c.yy5),6) as r5, " & _		"
"round((convert(float,c.b6)* convert(float,c.yx6)) / convert(float,c.yy6),6) as r6, " & _		"

```

convert(float,c.yx2))/convert
(float,c.yy2) ))),0.5) <= 0)
then (0) " & _
" else
((power(((convert(float,c.b2
) *
convert(float,c.yx2))/convert
(float,c.yy2) ),0.5) " & _
"
*power(convert(float,
(count(v.vrb_y)-2)),0.5))
" & _
"
/power(((1-
(((convert(float,c.b2) *
convert(float,c.yx2))/convert
(float,c.yy2) ))),0.5) end
)end ,6) as T2 " & _
" ,round(case
when((((1-
((convert(float,c.b3) *
convert(float,c.yx3))/convert
(float,c.yy3) )))<'0') or
((((convert(float,c.b3) " &
_
"
*
convert(float,c.yx3))/convert
(float,c.yy3))<'0'))then('0')
else( " & _
" case
when(power(((1-
(((convert(float,c.b3) *
convert(float,c.yx3))/convert
(float,c.yy3) ))),0.5) <= 0)
then (0) " & _
" else
((power(((convert(float,c.b3
) *
convert(float,c.yx3))/convert
(float,c.yy3) ),0.5) " & _
"
*power(convert(float,
(count(v.vrb_y)-2)),0.5))
" & _
"
/power(((1-
(((convert(float,c.b3) *
convert(float,c.yx3))/convert
(float,c.yy3) ))),0.5) end
)end ,6) as T3 " & _
" ,round(case
when((((1-
((convert(float,c.b4) *
convert(float,c.yx4))/convert
(float,c.yy4) )))<'0') or
((((convert(float,c.b4) " &
_
"
*
convert(float,c.yx4))/convert
(float,c.yy4) ))),0.5) <= 0)
then (0) " & _
" else
((power(((convert(float,c.b4)
) *
convert(float,c.yx4))/convert
(float,c.yy4) ),0.5) " & _
"
when(power(((1-
(((convert(float,c.b4) *
convert(float,c.yx4))/convert
(float,c.yy4) ))),0.5) <= 0)
then (0) " & _
" else
((power(((convert(float,c.b4)
) *
convert(float,c.yx4))/convert
(float,c.yy4) )),0.5) " & _
"
*power(convert(float,
(count(v.vrb_y)-2)),0.5))
" & _
"
/power(((1-
(((convert(float,c.b4) *
convert(float,c.yx4))/convert
(float,c.yy4) ))),0.5) end
)end ,6) as T4 " & _
" ,round(case
when((((1-
((convert(float,c.b5) *
convert(float,c.yx5))/convert
(float,c.yy5) )))<'0') or
((((convert(float,c.b5) " &
_
"
*
convert(float,c.yx5))/convert
(float,c.yy5))<'0'))then('0')
else( " & _
" case
when(power(((1-
(((convert(float,c.b5) *
convert(float,c.yx5))/convert
(float,c.yy5) ))),0.5) <= 0)
then (0) " & _
" else
((power(((convert(float,c.b5)
) *
convert(float,c.yx5))/convert
(float,c.yy5) ),0.5) " & _
"
*power(convert(float,
(count(v.vrb_y)-2)),0.5))
" & _
"
/power(((1-
(((convert(float,c.b5) *
convert(float,c.yx5))/convert
(float,c.yy5) ))),0.5) end
)end ,6) as T5 " & _
" ,round(case
when((((1-
((convert(float,c.b6) *
convert(float,c.yx6))/convert
(float,c.yy6) )))<'0') or
((((convert(float,c.b6) " &
_
"
*
convert(float,c.yx6))/convert
(float,c.yy6))<'0'))then('0')
else( " & _
" case
when(power(((1-
(((convert(float,c.b6) *
convert(float,c.yx6))/convert
(float,c.yy6) ))),0.5) <= 0)
then (0) " & _
" else
((power(((convert(float,c.b6)
) *
convert(float,c.yx6))/convert
(float,c.yy6) ),0.5) " & _
"
*power(convert(float,
(count(v.vrb_y)-2)),0.5))
" & _
"
/power(((1-
(((convert(float,c.b6) *
convert(float,c.yx6))/convert
(float,c.yy6) ))),0.5) end
)end ,6) as T6 " & _
" ,round(case
when((((1-
((convert(float,c.b7) *
convert(float,c.yx7))/convert
(float,c.yy7) )))<'0') or
((((convert(float,c.b7) " &
_
"
*
convert(float,c.yx7))/convert
(float,c.yy7))<'0'))then('0')
else( " & _
" case
when(power(((1-
(((convert(float,c.b7) *
convert(float,c.yx7))/convert
(float,c.yy7) ))),0.5) <= 0)
then (0) " & _
" else
((power(((convert(float,c.b7
) *
convert(float,c.yx7))/convert
(float,c.yy7) )),0.5) " & _
"
*power(convert(float,
(count(v.vrb_y)-2)),0.5))
" & _
"
/power(((1-

```

```

) *
convert(float,c.yx7))/conver
t(float,c.yy7 ),0.5) " & _
"
      *power(convert(flo
at,(count(v.vrb_y)-2)),0.5))
" & _
"
      /power(((1-
(((convert(float,c.b7) *
convert(float,c.yx7))/conver
t(float,c.yy7) )))),0.5)) end
end ,6) as T7 " & _
" ,round(case
      when((((1-
((convert(float,c.b8) *
convert(float,c.yx8))/conver
t(float,c.yy8) ))<'0') or
((((convert(float,c.b8) " &
_
"
      *
convert(float,c.yx8))/conver
t(float,c.yy8))<'0'))then('0')
else( " & _
" case
      when(power(((1-
(((convert(float,c.b8) *
convert(float,c.yx8))/conver
t(float,c.yy8) )))),0.5) <= 0)
then (0) " & _
" else
      ((power(((convert(float,c.b8
) *
convert(float,c.yx8))/conver
t(float,c.yy8)),0.5) " & _
"
      *power(convert(flo
at,(count(v.vrb_y)-2)),0.5))
" & _
"
      /power(((1-
(((convert(float,c.b8) *
convert(float,c.yx8))/conver
t(float,c.yy8) )))),0.5)) end
end ,6) as T8 " & _
" ,round(case
      when((((1-
((convert(float,c.b9) *
convert(float,c.yx9))/conver
t(float,c.yy9) ))<'0') or
((((convert(float,c.b9) " &
_
"
      *
convert(float,c.yx9))/conver
t(float,c.yy9))<'0'))then('0')
else( " & _
" case
      when(power(((1-
(((convert(float,c.b10) *
convert(float,c.yx10))/conve
rt(float,c.yy10) ))))<'0') or
((((convert(float,c.b10) " &
_
"
      *
convert(float,c.yx10))/conve
rt(float,c.yy10))<'0'))then('
0') else( " & _
" case
      when(power(((1-
(((convert(float,c.b10) *
convert(float,c.yx10))/conve
rt(float,c.yy10) )))),0.5) <=
0) then (0) " & _
" else
      ((power(((convert(float,c.b1
0) *
convert(float,c.yx10))/conve
rt(float,c.yy10) ),0.5) " & _
"
      *power(convert(flo
at,(count(v.vrb_y)-2)),0.5))
" & _
"
      /power(((1-
(((convert(float,c.b10) *
convert(float,c.yx10))/conve
rt(float,c.yy10) )))),0.5))
end end ,6) as T10 " & _
" ,round(case
      when((((1-
((convert(float,c.b11) *
convert(float,c.yx11))/conve
rt(float,c.yy11) ))))<'0') or
((((convert(float,c.b11) " &
_
"
      *
convert(float,c.yx11))/conve
rt(float,c.yy11))<'0'))then('
0') else( " & _
" case
      when(power(((1-
(((convert(float,c.b11) *
convert(float,c.yx11))/conve
rt(float,c.yy11) )))),0.5) <=
0) then (0) " & _
" else
      ((power(((convert(float,c.b1
1) *
convert(float,c.yx11))/conve
rt(float,c.yy11) ),0.5) " & _
"
      *power(convert(flo
at,(count(v.vrb_y)-2)),0.5))
" & _
"
      /power(((1-
(((convert(float,c.b11) *
convert(float,c.yx11))/conve
rt(float,c.yy11) )))),0.5))
end end ,6) as T11 " & _
" ,round(case
      when((((1-
((convert(float,c.b12) *
convert(float,c.yx12))/conve
rt(float,c.yy12) ))))<'0') or
((((convert(float,c.b12) " &
_
"
      *
convert(float,c.yx12))/conve
rt(float,c.yy12))<'0'))then('
0') else( " & _
" case
      when(power(((1-
(((convert(float,c.b12) *
convert(float,c.yx12))/conve
rt(float,c.yy12) )))),0.5) <=
0) then (0) " & _
" else
      ((power(((convert(float,c.b1
2) *
convert(float,c.yx12))/conve
rt(float,c.yy12) ),0.5) " &
_

```

```

"
    *power(convert(float,(count(v.vrb_y)-2)),0.5))
" & _
"
    /power(((1-
(((convert(float,c.b12) *
convert(float,c.yx12))/conve
rt(float,c.yy12) ))),0.5))
end )end ,6) as T12 " & _
"
    ",round(case
when((((1-
((convert(float,c.b13) *
convert(float,c.yx13))/conve
rt(float,c.yy13) ))<'0') or
((((convert(float,c.b13) " &
_
"
*
convert(float,c.yx13))/conve
rt(float,c.yy13))<'0'))then('
0') else( " & _
"
    case
when(power(((1-
(((convert(float,c.b13) *
convert(float,c.yx13))/conve
rt(float,c.yy13) ))),0.5) <=
0) then (0) " & _
"
    else
((power(((convert(float,c.b1
3) *
convert(float,c.yx13))/conve
rt(float,c.yy13) ),0.5) " & _
"
    *power(convert(flo
at,(count(v.vrb_y)-2)),0.5))
" & _
"
    /power(((1-
(((convert(float,c.b13) *
convert(float,c.yx13))/conve
rt(float,c.yy13) ))),0.5))
end )end ,6) as T13 " & _
"
    ",round(case
when((((1-
((convert(float,c.b14) *
convert(float,c.yx14))/conve
rt(float,c.yy14) ))<'0') or
((((convert(float,c.b14) " &
_
"
*
convert(float,c.yx14))/conve
rt(float,c.yy14))<'0'))then('
0') else( " & _
"
    case
when(power(((1-
(((convert(float,c.b14) *
convert(float,c.yx14))/conve
rt(float,c.yy14) ))),0.5) <=
0) then (0) " & _
"
    else
((power(((convert(float,c.b1
4) *
convert(float,c.yx14))/conve
rt(float,c.yy14) ),0.5) " & _
"
    *power(convert(flo
at,(count(v.vrb_y)-2)),0.5))
" & _
"
    /power(((1-
(((convert(float,c.b14) *
convert(float,c.yx14))/conve
rt(float,c.yy14) ))),0.5))
end )end ,6) as T14 " & _
"
    ",round(case
when((((1-
((convert(float,c.b15) *
convert(float,c.yx15))/conve
rt(float,c.yy15) ))<'0') or
((((convert(float,c.b15) " &
_
"
*
convert(float,c.yx15))/conve
rt(float,c.yy15))<'0'))then('
0') else( " & _
"
    case
when(power(((1-
(((convert(float,c.b15) *
convert(float,c.yx15))/conve
rt(float,c.yy15) ))),0.5) <=
0) then (0) " & _
"
    else
((power(((convert(float,c.b1
5) *
convert(float,c.yx15))/conve
rt(float,c.yy15) ),0.5) " & _
"
    *power(convert(flo
at,(count(v.vrb_y)-2)),0.5))
" & _
"
    /power(((1-
(((convert(float,c.b15) *
convert(float,c.yx15))/conve
rt(float,c.yy15) ))),0.5))
end )end ,6) as T15 " & _
"
    "from VRB_cal c
left join VRB_VARIABLE
v with(nolock) on
c.id_vrb=v.id_vrb where
c.id_vrb =" &
lbl_id.Text.ToString & "' "
& _
"
    "group by
c.b1,c.yx1,c.yy1,c.b2,c.yx2,
c.yy2,c.b3,c.yx3,c.yy3,c.b4,
c.yx4,c.yy4,c.b5,c.yx5,c.yy5
,c.b6,c.yx6,c.yy6,c.b7,c.yx7,
c.yy7, c.b8, c.yx8, c.yy8,
c.b9, c.yx9, c.yy9, c.b10,
c.yx10, c.yy10, c.b11,
c.yx11, c.yy11, c.b12,
c.yx12, c.yy12, c.b13,
c.yx13, c.yy13, c.b14,
c.yx14, c.yy14, c.b15,
c.yx15, c.yy15 "
"
    da.SelectCommand
= New SqlCommand(sql,
conn1)
da.Fill(dst, "TT")
dgv_T.DataSource =
dst.Tables("TT")
Catch ex As Exception
' Exit Sub
End Try
"
    dsf.Clear()
sql = "select " & _
"
    "round((convert(float,c.b1)
* convert(float,c.yx1) )/
(case
when((convert(float,c.yy1) -
(convert(float,c.b1) *
convert(float,c.yx1) ))='0')
then('1') " & _
"
    else
((convert(float,c.yy1) -
(convert(float,c.b1) *
convert(float,c.yx1)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2)end),6) as F1, " &
_
"
"
    round((convert(float,c.b2) *
convert(float,c.yx2) )/ (case
when((convert(float,c.yy2) -
(convert(float,c.b2) *
convert(float,c.yx2) ))='0')
then('1') " & _
"
    else
((convert(float,c.yy2) -
(convert(float,c.b2) *
convert(float,c.yx2)
)))end/case

```



```

when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F2, " &
-
"
round((convert(float,c.b3) *
convert(float,c.yx3) )/ (case
when((convert(float,c.yy3) -
(convert(float,c.b3) *
convert(float,c.yx3) ))='0')
then('1') " & _
" else
((convert(float,c.yy3) -
(convert(float,c.b3) *
convert(float,c.yx3)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F3, " &
-
"
round((convert(float,c.b4) *
convert(float,c.yx4) )/ (case
when((convert(float,c.yy4) -
(convert(float,c.b4) *
convert(float,c.yx4) ))='0')
then('1') " & _
" else
((convert(float,c.yy4) -
(convert(float,c.b4) *
convert(float,c.yx4)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F4, " &
-
"
round((convert(float,c.b5) *
convert(float,c.yx5) )/ (case
when((convert(float,c.yy5) -
(convert(float,c.b5) *
convert(float,c.yx5) ))='0')
then('1') " & _
" else
((convert(float,c.yy5) -
(convert(float,c.b5) *
convert(float,c.yx5)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F5, " &
-
"
round((convert(float,c.b6) *
convert(float,c.yx6) )/ (case
when((convert(float,c.yy6) -
(convert(float,c.b6) *
convert(float,c.yx6) ))='0')
then('1') " & _
" else
((convert(float,c.yy6) -
(convert(float,c.b6) *
convert(float,c.yx6)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F6, " &
-
"
round((convert(float,c.b7) *
convert(float,c.yx7) )/ (case
when((convert(float,c.yy7) -
(convert(float,c.b7) *
convert(float,c.yx7) ))='0')
then('1') " & _
" else
((convert(float,c.yy7) -
(convert(float,c.b7) *
convert(float,c.yx7)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F7, " &
-
"
round((convert(float,c.b8) *
convert(float,c.yx8) )/ (case
when((convert(float,c.yy8) -
(convert(float,c.b8) *
convert(float,c.yx8) ))='0')
then('1') " & _
" else
((convert(float,c.yy8) -
(convert(float,c.b8) *
convert(float,c.yx8)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F8, " &
-
"
round((convert(float,c.b9) *
convert(float,c.yx9) )/ (case
when((convert(float,c.yy9) -
(convert(float,c.b9) *
convert(float,c.yx9) ))='0')
then('1') " & _
" else
((convert(float,c.yy9) -
(convert(float,c.b9) *
convert(float,c.yx9)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F9, " &
-
"
round((convert(float,c.b10)
* convert(float,c.yx10) )/
(case
when((convert(float,c.yy10)
-(convert(float,c.b10) *
convert(float,c.yx10) ))='0')
then('1') " & _
" else
((convert(float,c.yy10) -
(convert(float,c.b10) *
convert(float,c.yx10)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F10, "
& _
"
round((convert(float,c.b11)
* convert(float,c.yx11) )/
(case
when((convert(float,c.yy11)
-(convert(float,c.b11) *
convert(float,c.yx11) ))='0')
then('1') " & _
" else
((convert(float,c.yy11) -
(convert(float,c.b11) *
convert(float,c.yx11)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F11, "
& _
"
round((convert(float,c.b12)
* convert(float,c.yx12) )/
(case
when((convert(float,c.yy12)
-(convert(float,c.b12) *
convert(float,c.yx12) ))='0')
then('1') " & _
" else
((convert(float,c.yy12) -
(convert(float,c.b12) *
convert(float,c.yx12)
)))end/case
when((count(v.vrb_y)-
2)='0')then('1')else((count(v.
vrb_y)-2))end),6) as F12, "
& _
"
round((convert(float,c.b13)
* convert(float,c.yx13) )/
(case

```

when((convert(float,c.yy13) -(convert(float,c.b13) * convert(float,c.yx13))='0') then('1') " & _ " else ((convert(float,c.yy13) - (convert(float,c.b13) * convert(float,c.yx13)))end/ case when((count(v.vrb_y)- 2)=0)then('1')else((count(v. vrb_y)-2))end),6) as F13, " & _ " round((convert(float,c.b14) * convert(float,c.yx14))/ (case when((convert(float,c.yy14) -(convert(float,c.b14) * convert(float,c.yx14))='0') then('1') " & _ " else ((convert(float,c.yy14) - (convert(float,c.b14) * convert(float,c.yx14)))end/ case when((count(v.vrb_y)- 2)=0)then('1')else((count(v. vrb_y)-2))end),6) as F14, " & _ " round((convert(float,c.b15) * convert(float,c.yx15))/ (case when((convert(float,c.yy15) -(convert(float,c.b15) * convert(float,c.yx15))='0') then('1') " & _ " else ((convert(float,c.yy15) - (convert(float,c.b15) * convert(float,c.yx15)))end/ case when((count(v.vrb_y)- 2)=0)then('1')else((count(v. vrb_y)-2))end),6) as F15 " & _ " from VRB_cal c left join VRB_VARIABLE v with(nolock) on c.id_vrb=v.id_vrb where c.id_vrb= "" & lbl_id.Text.ToString & "" " & _ " group by c.b1,c.yx1,c.b2,c.yx2,c.b3,c. yx3,c.b4,c.yx4,c.b5,c.yx5,c.	b6,c.yx6,c.b7,c.yx7,c.b8,c.y x8,c.b9,c.yx9,c.b10,c.yx10 " & _ " ,c.b11,c.yx11,c.b12,c.yx12,c .b13,c.yx13,c.b14,c.yx14,c. b15,c.yx15,yy1,yy2,yy3,yy4 ,yy5,yy6,yy7,yy8,yy9,yy10, yy11,yy12,yy13,yy14,yy15" da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(dsf, "FF") dgv_F.DataSource = dsf.Tables("FF") max_s() For j = 1 To max_id - 1 dgv_R.Columns(j - 1).HeaderText = DataGridViewX2.Rows(j - 1).Cells(1).Value.ToString dgv_T.Columns(j - 1).HeaderText = DataGridViewX2.Rows(j - 1).Cells(1).Value.ToString dgv_F.Columns(j - 1).HeaderText = DataGridViewX2.Rows(j - 1).Cells(1).Value.ToString Next Dim ij As Integer max_id = max_id - 1 For ij = max_id To 15 Try dgv_R.Columns(max_id).H eaderText = "" Catch ex As Exception End Try Try dgv_T.Columns(max_id).H eaderText = "" Catch ex As Exception End Try Try dgv_F.Columns(max_id).H eaderText = "" Catch ex As Exception End Try	End Try max_id = max_id + 1 Next Dim ij As Integer For ij = 0 To 14 dgv_R.Columns(ij).Width = "75" dgv_T.Columns(ij).Width = "75" dgv_F.Columns(ij).Width = "75" Next End Sub Private Sub pt_edit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pt_edit.Click data_x.Refresh() DataGridView2.Rows.Clear () show_data_view() Tags = "Show" max_s() max_vrbs.Text = max_vrb If max_vrb = "0" Then MsgBox("Please enter the message.") Exit Sub End If If max_vrb = "1" Then DataGridView2.Rows.Add(max_vrb) Else DataGridView2.Rows.Add(max_vrb - 1)
---	--	--

End If	dgv_save.Visible = False	,vrb_x3=@vrb_x3 ,vrb_x4=@vrb_x4 ,vrb_x5=@vrb_x5 ,vrb_x6=@vrb_x6 ,vrb_x7=@vrb_x7 , " & _ "
DataGridView2.Columns(0) .ReadOnly = True	'but_Search.Visible = False	
DataGridView2.Columns(1) .ReadOnly = True	cmb_name.Visible = False	
DataGridView2.Columns(3) .ReadOnly = True	End Sub	vrb_x8=@vrb_x8 ,vrb_x9=@vrb_x9 ,vrb_x10=@vrb_x10 ,vrb_x11=@vrb_x11 ,vrb_x12=@vrb_x12 ,vrb_x13=@vrb_x13 ,vrb_x14=@vrb_x14 ,vrb_x15=@vrb_x15 , " & _ "
Try	Private Sub	
For i = 0 To	pt_save_edit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pt_save_edit.Click	
DataGridView1.Rows.Count - 2		
Dim ma As Integer	If str = "Data Deficient." Then	yx1=@yx1,yx2=@yx2,yx3 =@yx3,yx4=@yx4,yx5=@yx5, yx6=@yx6,yx7=@yx7,yx8= @yx8,yx9=@yx9,yx10=@yx10, yx11=@yx11,yx12=@yx12, yx13=@yx13,yx14=@yx14, yx15=@yx15 , " & _ "
ma = maxs + 1	str = ""	
For j = 0 To ma + 1	Exit Sub	
1	End If	
DataGridView2.Rows(i).Cells(j).Value =	pt_save_edit.Focus()	
DataGridView1.Rows(i).Cells(j).Value.ToString	lbl_id.Text = cmb_name.SelectedValue	
Next	If lbl_id.Text = ""	y_1=@y_1,x_1=@x_1,x_2=@x_2, x_3=@x_3,x_4=@x_4,x_5=@x_5, x_6=@x_6,x_7=@x_7,x_8=@x_8, x_9=@x_9,x_10=@x_10,x_11=@x_11, x_12=@x_12,x_13=@x_13,x_14=@x_14, x_15=@x_15 " & _ "
Next	Then	
Catch ex As Exception	MsgBox("Please select the Parameter....")	
End Try	Exit Sub	
Dim ij As Integer	End If	
max_id = max_id + 2	For i = 0 To	" where id_vrb =" & lbl_id.Text & " and vrb_date =" &
For ij = max_id To 17	DataGridView2.Rows.Count - 1	& DataGridview2.Rows(i).Cells(1).Value.ToString & ""
DataGridView2.Columns(max_id).HeaderText = ""	Try	
max_id = max_id + 1	Convert.ToDouble(DataGridView2.Rows(i).Cells(2).Value)	With com
Next	Catch ex As Exception	.Connection = conn1
DataGridView2.Height = "535"	MsgBox("Only numbers.")	.CommandType = CommandType.Text
pt_close.Visible = True	Exit Sub	.CommandText = sql
DataGridView2.Visible = True	End Try	.Parameters.Clear()
pt_save_edit.Visible = True	Next	
pt_save.Visible = False	For i = 0 To	If
pt_add.Visible = False	DataGridView2.Rows.Count - 1	String.IsNullOrEmpty(Trim(CStr(DataGridview2.Rows(i).Cells(2).Value))) Then
pt_report.Visible = False	sql = "update VRB_VARIABLE " & _ " set	.Parameters.Add("@vrb_y", SqlDbType.VarChar).Value = System.DBNull.Value
pt_edit.Visible = False	vrb_y=@vrb_y ,vrb_x1=@vrb_x1 ,vrb_x2=@vrb_x2	

<pre>.Parameters.Add("y_1", SqlDbType.VarChar).Value = System.DBNull.Value Else</pre>	<pre>(DataGridView2.Rows(i).Cells(3).Value.ToString) ^ 2 End If</pre>	<pre>.Parameters.Add("@yx3", SqlDbType.VarChar).Value = System.DBNull.Value</pre>
<pre>.Parameters.Add("@vrb_y", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(2).Value.ToString</pre>	<pre>If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(4).Value))) Then</pre>	<pre>.Parameters.Add("@x_3", SqlDbType.VarChar).Value = System.DBNull.Value Else</pre>
<pre>.Parameters.Add("@y_1", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) ^ 2 End If</pre>	<pre>.Parameters.Add("@vrb_x2", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx2", SqlDbType.VarChar).Value = System.DBNull.Value</pre>	<pre>.Parameters.Add("@vrb_x3", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(5).Value.ToString</pre>
<pre>If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(3).Value))) Then</pre>	<pre>.Parameters.Add("@x_2", SqlDbType.VarChar).Value = System.DBNull.Value Else</pre>	<pre>.Parameters.Add("@yx3", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(5).Value.ToString)</pre>
<pre>.Parameters.Add("@vrb_x1", SqlDbType.VarChar).Value = System.DBNull.Value</pre>	<pre>.Parameters.Add("@vrb_x2", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(4).Value.ToString</pre>	<pre>.Parameters.Add("@x_3", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(5).Value.ToString) ^ 2 End If</pre>
<pre>.Parameters.Add("@yx1", SqlDbType.VarChar).Value = System.DBNull.Value</pre>	<pre>.Parameters.Add("@yx2", SqlDbType.VarChar).Value =</pre>	<pre>If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(6).Value))) Then</pre>
<pre>.Parameters.Add("@x_1", SqlDbType.VarChar).Value = System.DBNull.Value Else</pre>	<pre>(DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(4).Value.ToString)</pre>	<pre>.Parameters.Add("@vrb_x4", SqlDbType.VarChar).Value = System.DBNull.Value</pre>
<pre>.Parameters.Add("@vrb_x1", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(3).Value.ToString</pre>	<pre>.Parameters.Add("@x_2", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(4).Value.ToString) ^ 2</pre>	<pre>.Parameters.Add("@yx4", SqlDbType.VarChar).Value = System.DBNull.Value</pre>
<pre>.Parameters.Add("@yx1", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(3).Value.ToString)</pre>	<pre>End If If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(5).Value))) Then</pre>	<pre>.Parameters.Add("@x_4", SqlDbType.VarChar).Value = System.DBNull.Value Else</pre>
<pre>.Parameters.Add("@x_1", SqlDbType.VarChar).Value =</pre>	<pre>.Parameters.Add("@vrb_x3", SqlDbType.VarChar).Value = System.DBNull.Value</pre>	<pre>.Parameters.Add("@vrb_x4", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(6).Value.ToString</pre>

<pre>.Parameters.Add("@yx4", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(6).Value.ToString) .Parameters.Add("@x_4", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(6).Value.ToString) ^ 2 End If</pre>	<pre>If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(8).Value))) Then .Parameters.Add("@vrb_x6", ", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx6", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_6", SqlDbType.VarChar).Value = System.DBNull.Value Else</pre>	<pre>SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x7", ", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(9).Value.ToString .Parameters.Add("@yx7", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(9).Value.ToString)</pre>
<pre>If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(7).Value))) Then .Parameters.Add("@vrb_x5", ", SqlDbType.VarChar).Value = System.DBNull.Value</pre>	<pre>.Parameters.Add("@vrb_x6", ", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(8).Value.ToString</pre>	<pre>.Parameters.Add("@x_7", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(9).Value.ToString) ^ 2 End If</pre>
<pre>.Parameters.Add("@yx5", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_5", SqlDbType.VarChar).Value = System.DBNull.Value Else</pre>	<pre>.Parameters.Add("@yx6", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(8).Value.ToString)</pre>	<pre>If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(10).Value))) Then .Parameters.Add("@vrb_x8", ", SqlDbType.VarChar).Value = System.DBNull.Value</pre>
<pre>.Parameters.Add("@vrb_x5", ", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(7).Value.ToString</pre>	<pre>.Parameters.Add("@x_6", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(8).Value.ToString) ^ 2 End If</pre>	<pre>.Parameters.Add("@yx8", SqlDbType.VarChar).Value = System.DBNull.Value</pre>
<pre>.Parameters.Add("@yx5", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(7).Value.ToString)</pre>	<pre>If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(9).Value))) Then .Parameters.Add("@vrb_x7", ", SqlDbType.VarChar).Value = System.DBNull.Value</pre>	<pre>Else .Parameters.Add("@vrb_x8", ", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(10).Value.ToString</pre>
<pre>.Parameters.Add("@x_5", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(7).Value.ToString) ^ 2 End If</pre>	<pre>.Parameters.Add("@yx7", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_7",</pre>	<pre>.Parameters.Add("@yx8", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) *</pre>

(DataGridView2.Rows(i).Cells(10).Value.ToString)	0", SqlDbType.VarChar).Value = System.DBNull.Value	= DataGridView2.Rows(i).Cells(13).Value.ToString
.Parameters.Add("@x_8", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(10).Value.ToString) ^ 2 End If	.Parameters.Add("@yx10", SqlDbType.VarChar).Value = System.DBNull.Value	.Parameters.Add("@yx11", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(13).Value.ToString)
If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(11).Value))) Then	.Parameters.Add("@x_10", SqlDbType.VarChar).Value = System.DBNull.Value Else	.Parameters.Add("@x_11", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(13).Value.ToString) ^ 2 End If
.Parameters.Add("@vrb_x9", SqlDbType.VarChar).Value = System.DBNull.Value	.Parameters.Add("@vrb_x10", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(12).Value.ToString	If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(14).Value))) Then
.Parameters.Add("@yx9", SqlDbType.VarChar).Value = System.DBNull.Value	.Parameters.Add("@yx10", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(12).Value.ToString)	.Parameters.Add("@vrb_x12", SqlDbType.VarChar).Value = System.DBNull.Value
.Parameters.Add("@x_9", SqlDbType.VarChar).Value = System.DBNull.Value Else	.Parameters.Add("@x_10", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(12).Value.ToString) ^ 2 End If	.Parameters.Add("@yx12", SqlDbType.VarChar).Value = System.DBNull.Value
.Parameters.Add("@vrb_x9", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(11).Value.ToString	If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(13).Value))) Then	.Parameters.Add("@x_12", SqlDbType.VarChar).Value = System.DBNull.Value Else
.Parameters.Add("@yx9", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(11).Value.ToString)	.Parameters.Add("@vrb_x11", SqlDbType.VarChar).Value = System.DBNull.Value	.Parameters.Add("@vrb_x12", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(14).Value.ToString
.Parameters.Add("@x_9", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(11).Value.ToString) ^ 2 End If	.Parameters.Add("@yx11", SqlDbType.VarChar).Value = System.DBNull.Value	.Parameters.Add("@yx12", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).Cells(2).Value.ToString) * (DataGridView2.Rows(i).Cells(14).Value.ToString)
If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(12).Value))) Then	.Parameters.Add("@x_11", SqlDbType.VarChar).Value = System.DBNull.Value Else	.Parameters.Add("@x_12", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Cells(14).Value.ToString
.Parameters.Add("@vrb_x1	.Parameters.Add("@vrb_x1", SqlDbType.VarChar).Value	

<pre> (DataGridView2.Rows(i).Cells(14).Value.ToString) ^ 2 End If If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(15).Value))) Then .Parameters.Add("@vrb_x1 3", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx13", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_13", SqlDbType.VarChar).Value = System.DBNull.Value Else Dim aa As String aa = DataGridView2.Rows(i).Ce lls(15).Value.ToString .Parameters.Add("@vrb_x1 3", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(15).Value.ToString .Parameters.Add("@yx13", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(2).Value.ToString) * (DataGridView2.Rows(i).C ells(15).Value.ToString) .Parameters.Add("@x_13", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(15).Value.ToString) ^ 2 End If If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(16).Value))) Then .Parameters.Add("@vrb_x1 4", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@vrb_x1 4", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(16).Value.ToString .Parameters.Add("@yx14", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(2).Value.ToString) * (DataGridView2.Rows(i).C ells(16).Value.ToString) .Parameters.Add("@x_14", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(16).Value.ToString ^ 2 End If If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(17).Value))) Then .Parameters.Add("@vrb_x1 5", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@vrb_x1 5", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(17).Value.ToString .Parameters.Add("@yx14", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(2).Value.ToString) * (DataGridView2.Rows(i).C ells(16).Value.ToString) .Parameters.Add("@x_14", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(16).Value.ToString ^ 2 End If .ExecuteNonQuery() End With Next show_save() up_cal() For i = 0 To DataGridView2.Rows.Coun t - 2 sql = "select case when d.[1] is null then'0'else d.[1] end as'1', " & _ "case when d.[2] is null then'0'else d.[2] end as'2', " & _ "case when d.[3] is null then'0'else d.[3] end as'3', " & _ "case when d.[4] is null then'0'else d.[4] end as'4', " & _ "case when d.[5] is null then'0'else d.[5] end as'5', " & _ "case when d.[6] is null then'0'else d.[6] end as'6', " & _ "case when d.[7] is null then'0'else d.[7] end as'7', " & _ "case when d.[8] is null then'0'else d.[8] end as'8', " & _ </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx14", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@vrb_x1 4", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(16).Value.ToString .Parameters.Add("@yx14", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(2).Value.ToString) * (DataGridView2.Rows(i).C ells(16).Value.ToString) .Parameters.Add("@x_14", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(16).Value.ToString ^ 2 End If If String.IsNullOrEmpty(Trim (CStr(DataGridView2.Rows (i).Cells(17).Value))) Then .Parameters.Add("@vrb_x1 5", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@yx15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@x_15", SqlDbType.VarChar).Value = System.DBNull.Value .Parameters.Add("@vrb_x1 5", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(17).Value.ToString .Parameters.Add("@yx14", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(2).Value.ToString) * (DataGridView2.Rows(i).C ells(16).Value.ToString) .Parameters.Add("@x_14", SqlDbType.VarChar).Value = DataGridView2.Rows(i).Ce lls(16).Value.ToString ^ 2 End If .ExecuteNonQuery() End With Next show_save() up_cal() For i = 0 To DataGridView2.Rows.Coun t - 2 sql = "select case when d.[1] is null then'0'else d.[1] end as'1', " & _ "case when d.[2] is null then'0'else d.[2] end as'2', " & _ "case when d.[3] is null then'0'else d.[3] end as'3', " & _ "case when d.[4] is null then'0'else d.[4] end as'4', " & _ "case when d.[5] is null then'0'else d.[5] end as'5', " & _ "case when d.[6] is null then'0'else d.[6] end as'6', " & _ "case when d.[7] is null then'0'else d.[7] end as'7', " & _ "case when d.[8] is null then'0'else d.[8] end as'8', " & _ </pre>	<pre> DataGridView2.Rows(i).Ce lls(17).Value.ToString .Parameters.Add("@yx15", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(2).Value.ToString) * (DataGridView2.Rows(i).C ells(17).Value.ToString) .Parameters.Add("@x_15", SqlDbType.VarChar).Value = (DataGridView2.Rows(i).C ells(17).Value.ToString) ^ 2 End If .ExecuteNonQuery() End With Next show_save() up_cal() For i = 0 To DataGridView2.Rows.Coun t - 2 sql = "select case when d.[1] is null then'0'else d.[1] end as'1', " & _ "case when d.[2] is null then'0'else d.[2] end as'2', " & _ "case when d.[3] is null then'0'else d.[3] end as'3', " & _ "case when d.[4] is null then'0'else d.[4] end as'4', " & _ "case when d.[5] is null then'0'else d.[5] end as'5', " & _ "case when d.[6] is null then'0'else d.[6] end as'6', " & _ "case when d.[7] is null then'0'else d.[7] end as'7', " & _ "case when d.[8] is null then'0'else d.[8] end as'8', " & _ </pre>
---	---	--

```

"case when
d.[9] is null then'0'else d.[9]
end as'9', " & _
"case when
d.[10] is null then'0'else
d.[10] end as'10', " & _
"case when
d.[11] is null then'0'else
d.[11] end as'11', " & _
"case when
d.[12] is null then'0'else
d.[12] end as'12', " & _
"case when
d.[13] is null then'0'else
d.[13] end as'13', " & _
"case when
d.[14] is null then'0'else
d.[14] end as'14', " & _
"case when
d.[15] is null then'0'else
d.[15] end as'15', " & _
"case when
d.[16] is null then'0'else
d.[16] end as'16', " & _
"case when
d.[17] is null then'0'else
d.[17] end as'17', " & _
"case when
d.[18] is null then'0'else
d.[18] end as'18', " & _
"case when
d.[19] is null then'0'else
d.[19] end as'19', " & _
"case when
d.[20] is null then'0'else
d.[20] end as'20', " & _
"case when
d.[21] is null then'0'else
d.[21] end as'21', " & _
"case when
d.[22] is null then'0'else
d.[22] end as'22', " & _
"case when
d.[23] is null then'0'else
d.[23] end as'23', " & _
"case when
d.[24] is null then'0'else
d.[24] end as'24', " & _
"case when
d.[25] is null then'0'else
d.[25] end as'25', " & _
"case when
d.[26] is null then'0'else
d.[26] end as'26', " & _
"case when
d.[27] is null then'0'else
d.[27] end as'27', " & _

"case when
d.[28] is null then'0'else
d.[28] end as'28', " & _
"case when
d.[29] is null then'0'else
d.[29] end as'29', " & _
"case when
d.[30] is null then'0'else
d.[30] end as'30' from " & _
" (select
row_number() over (order
by id_no)as num
,convert(float,vrb_x" & i +
1 & ")as ab from
VRB_VARIABLE where
id_vrb= " & lbl_id.Text &
") as aa " & _
" PIVOT
(sum(ab)FOR num in
([1],[2],[3],[4],[5],[6],[7],[8]
,[9],[10],[11],[12],[13],[14],
[15],[16],[17],[18],[19],[20]
," & _
"
[21],[22],[23],[24],[25],[26]
,[27],[28],[29],[30])) as d"
With com
.CommandType =
CommandType.Text
.CommandText =
sql
.Connection =
conn1
dr =
.ExecuteReader()
dr.Read()
xr1 = dr.Item("1")
: xr2 = dr.Item("2") : xr3 =
dr.Item("3") : xr4 =
dr.Item("4") : xr5 =
dr.Item("5")
xr6 = dr.Item("6")
: xr7 = dr.Item("7") : xr8 =
dr.Item("8") : xr9 =
dr.Item("9") : xr10 =
dr.Item("10")
xr11 =
dr.Item("11") : xr12 =
dr.Item("12") : xr13 =
dr.Item("13") : xr14 =
dr.Item("14") : xr15 =
dr.Item("15")
xr16 =
dr.Item("16") : xr17 =
dr.Item("17") : xr18 =
dr.Item("18") : xr19 =
dr.Item("19") : xr20 =
dr.Item("20")
xr21 =
dr.Item("21") : xr22 =
dr.Item("22") : xr23 =
dr.Item("23") : xr24 =
dr.Item("24") : xr25 =
dr.Item("25")
xr26 =
dr.Item("26") : xr27 =
dr.Item("27") : xr28 =
dr.Item("28") : xr29 =
dr.Item("29") : xr30 =
dr.Item("30")
End With
dr.Close()

sql = "update
VRB_x1 set x1=" & xr1 &
"" ,x2=" & xr2 & "" ,x3=" &
xr3 & "" ,x4=" & xr4 &
"" ,x5=" & xr5 & "" ,x6=" &
xr6 & "" ,x7=" & xr7 & "" ,
" & _
" x8=" & xr8 &
"" ,x9=" & xr9 & "" ,x10=" &
xr10 & "" ,x11=" & xr11 &
"" ,x12=" & xr12 & "" ,x13=" &
xr13 & "" ,x14=" & xr14 &
"" ,x15=" & xr15 & "" , " &
-
" x16=" & xr16
& "" ,x17=" & xr17 &
"" ,x18=" & xr18 & "" ,x19=" &
xr19 & "" ,x20=" & xr20 &
"" ,x21=" & xr21 &
"" ,x22=" & xr22 & "" ,x23=" &
xr23 & "" , " & _
" x24=" & xr24
& "" ,x25=" & xr25 &
"" ,x26=" & xr26 & "" ,x27=" &
xr27 & "" ,x28=" & xr28
& "" ,x29=" & xr29 &
"" ,x30=" & xr30 & "" & _
" where r_no= ""
& lbl_id.Text & i + 1 & ""

With com
.Connection =
conn1
.CommandType =
CommandType.Text
.CommandText =
sql
.ExecuteReaderNonQuery()
End With

```


<pre> Next For i = 1 To maxs sql = " select round(power(power(((conve rt(float,c.b1) * case when(convert(float,c.yx1) <='0')then('0')else (convert(float,c.yx1))end) / " & _ " (case when(convert(float,c.yy" & i & ") = '0')then('1')else (convert(float,c.yy" & i & "))end)),0.5),2),6) as r " & - " from VRB_cal c left join VRB_VARIABLE v with(nolock) on c.id_vrb=v.id_vrb where c.id_vrb= '" & lbl_id.Text & "'" With com .CommandType = CommandType.Text .CommandText = sql .Connection = conn1 dr = .ExecuteReader() dr.Read() Rr = dr.Item("r") End With dr.Close() dst.Clear() sql = "select * from (select distinct " & _ "case when((convert(float,c.yx1) ='0') or (convert(float,c.b1)='0'))the n('0')else(convert(float,c.b1) *convert(float,c.yx1)/ convert(float,c.yy1))end as r1, " & _ "case when((convert(float,c.yx2) ='0') or (convert(float,c.b2)='0'))the n('0')else(convert(float,c.b2) *convert(float,c.yx2)/ convert(float,c.yy2))end as r2, " & _ </pre>	<pre> "case when((convert(float,c.yx3) ='0') or (convert(float,c.b3)='0'))the n('0')else(convert(float,c.b3) *convert(float,c.yx3)/ convert(float,c.yy3))end as r3, " & _ "case when((convert(float,c.yx4) ='0') or (convert(float,c.b4)='0'))the n('0')else(convert(float,c.b4) *convert(float,c.yx4)/ convert(float,c.yy4))end as r4, " & _ "case when((convert(float,c.yx5) ='0') or (convert(float,c.b5)='0'))the n('0')else(convert(float,c.b5) *convert(float,c.yx5)/ convert(float,c.yy5))end as r5, " & _ "case when((convert(float,c.yx6) ='0') or (convert(float,c.b6)='0'))the n('0')else(convert(float,c.b6) *convert(float,c.yx6)/ convert(float,c.yy6))end as r6, " & _ "case when((convert(float,c.yx7) ='0') or (convert(float,c.b7)='0'))the n('0')else(convert(float,c.b7) *convert(float,c.yx7)/ convert(float,c.yy7))end as r7, " & _ "case when((convert(float,c.yx8) ='0') or (convert(float,c.b8)='0'))the n('0')else(convert(float,c.b8) *convert(float,c.yx8)/ convert(float,c.yy8))end as r8, " & _ "case when((convert(float,c.yx9) ='0') or (convert(float,c.b9)='0'))the n('0')else(convert(float,c.b9) *convert(float,c.yx9)/ convert(float,c.yy9))end as r9, " & _ </pre>	<pre> "case when((convert(float,c.yx10) ='0') or (convert(float,c.b10)='0'))th en('0')else(convert(float,c.b1 0)*convert(float,c.yx10)/ convert(float,c.yy10))end as r10, " & _ "case when((convert(float,c.yx11) ='0') or (convert(float,c.b11)='0'))th en('0')else(convert(float,c.b1 1)*convert(float,c.yx11)/ convert(float,c.yy11))end as r11, " & _ "case when((convert(float,c.yx12) ='0') or (convert(float,c.b12)='0'))th en('0')else(convert(float,c.b1 2)*convert(float,c.yx12)/ convert(float,c.yy12))end as r12, " & _ "case when((convert(float,c.yx13) ='0') or (convert(float,c.b13)='0'))th en('0')else(convert(float,c.b1 3)*convert(float,c.yx13)/ convert(float,c.yy13))end as r13, " & _ "case when((convert(float,c.yx14) ='0') or (convert(float,c.b14)='0'))th en('0')else(convert(float,c.b1 4)*convert(float,c.yx14)/ convert(float,c.yy14))end as r14, " & _ "case when((convert(float,c.yx15) ='0') or (convert(float,c.b15)='0'))th en('0')else(convert(float,c.b1 5)*convert(float,c.yx15)/ convert(float,c.yy15))end as r15 " & _ "from VRB_cal c left join VRB_VARIABLE v with(nolock) on c.id_vrb=v.id_vrb where c.id_vrb = '" & lbl_id.Text & "'" datatable " & _ "UNPIVOT (r for names </pre>
---	--	--

<pre> in(r1,r2,r3,r4,r5,r6,r7,r8,r9,r 10,r11,r12,r13,r14,r15))as vv" da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(dst, "vv") DataGridView5.DataSource = dst.Tables("vv") dsf.Clear() sql = " select id_no,r_no from VRB_x1 where substring (r_no,1,4)="" & lbl_id.Text & "" order by id_no asc " da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(dsf, "r_no") DataGridView6.DataSource = dsf.Tables("r_no") If DataGridView6.Rows.Coun t = 0 Then Else r = r + 1 r = 0 & r sql = "update VRB_r set r=@r where r_no="" & lbl_id.Text & r & "" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() Try n = i - 1 If String.IsNullOrEmpty(Trim (CStr(DataGridView6.Rows (i - 1).Cells(1).Value))) Then .Parameters.Add("@r", </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@r", SqlDbType.VarChar).Value = DataGridView5.Rows(i - 1).Cells(0).Value.ToString End If .ExecuteNonQuery() Catch ex As Exception End Try End With End If Next show_data() show_data_view() dgv_save.Rows.Clear() close_edit() End Sub Protected Sub up_cal() sql = "update VRB_cal " & _ " set yx1=@yx1,yx2=@yx2,yx3 =@yx3,yx4=@yx4,yx5=@y x5,yx6=@yx6,yx7=@yx7,y x8=@yx8,yx9=@yx9,yx10 =@yx10,yx11=@yx11,yx12 =@yx12,yx13=@yx13,yx14 =@yx14,yx15=@yx15 " & _ " ,xx1=@xx1,xx2=@xx2,xx3 =@xx3,xx4=@xx4,xx5=@x x5,xx6=@xx6,xx7=@xx7,x x8=@xx8,xx9=@xx9,xx10 =@xx10,xx11=@xx11,xx12 =@xx12,xx13=@xx13,xx14 =@xx14,xx15=@xx15 " & _ " ,yy1=@yy1,yy2=@yy2,yy3 =@yy3,yy4=@yy4,yy5=@y y5,yy6=@yy6,yy7=@yy7,y y8=@yy8,yy9=@yy9,yy10 =@yy10,yy11=@yy11,yy12 =@yy12,yy13=@yy13,yy14 =@yy14,yy15=@yy15 " & _ </pre>	<pre> " ,b1=@b1,b2=@b2,b3=@b3, b4=@b4,b5=@b5,b6=@b6, b7=@b7,b8=@b8,b9=@b9, b10=@b10,b11=@b11,b12 =@b12,b13=@b13,b14=@b 14,b15=@b15 " & _ " ,b0_1=@b0_1,b0_2=@b0_2 ,b0_3=@b0_3,b0_4=@b0_4 ,b0_5=@b0_5,b0_6=@b0_6 ,b0_7=@b0_7,b0_8=@b0_8 ,b0_9=@b0_9,b0_10=@b0_ 10,b0_11=@b0_11 " & _ " ,b0_12=@b0_12,b0_13=@b 0_13,b0_14=@b0_14,b0_1 5=@b0_15 " & _ " where id_vrb =" & lbl_id.Text.ToString & "" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() If String.IsNullOrEmpty(Trim (CStr(yx1))) Then .Parameters.Add("@yx1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx1", SqlDbType.VarChar).Value = yx1 End If If String.IsNullOrEmpty(Trim (CStr(yx2))) Then .Parameters.Add("@yx2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx2", SqlDbType.VarChar).Value = yx2 End If </pre>
---	--	--

<pre> If String.IsNullOrEmpty(Trim (CStr(yx3))) Then .Parameters.Add("@yx3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx3", SqlDbType.VarChar).Value = yx3 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx7))) Then .Parameters.Add("@yx7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx7", SqlDbType.VarChar).Value = yx7 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx11))) Then .Parameters.Add("@yx11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx11", SqlDbType.VarChar).Value = yx11 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yx4))) Then .Parameters.Add("@yx4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx4", SqlDbType.VarChar).Value = yx4 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx8))) Then .Parameters.Add("@yx8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx8", SqlDbType.VarChar).Value = yx8 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx12))) Then .Parameters.Add("@yx12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx12", SqlDbType.VarChar).Value = yx12 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yx5))) Then .Parameters.Add("@yx5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx5", SqlDbType.VarChar).Value = yx5 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx9))) Then .Parameters.Add("@yx9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx9", SqlDbType.VarChar).Value = yx9 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx13))) Then .Parameters.Add("@yx13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx13", SqlDbType.VarChar).Value = yx13 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yx6))) Then .Parameters.Add("@yx6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx6", SqlDbType.VarChar).Value = yx6 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx10))) Then .Parameters.Add("@yx10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx10", SqlDbType.VarChar).Value = yx10 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx14))) Then .Parameters.Add("@yx14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx14", SqlDbType.VarChar).Value = yx14 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(yx15))) Then .Parameters.Add("@yx15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yx15", SqlDbType.VarChar).Value = yx15 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx4))) Then .Parameters.Add("@xx4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx4", SqlDbType.VarChar).Value = xx4 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx8))) Then .Parameters.Add("@xx8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx8", SqlDbType.VarChar).Value = xx8 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yx1))) Then .Parameters.Add("@xx1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx1", SqlDbType.VarChar).Value = xx1 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx5))) Then .Parameters.Add("@xx5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx5", SqlDbType.VarChar).Value = xx5 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx9))) Then .Parameters.Add("@xx9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx9", SqlDbType.VarChar).Value = xx9 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx2))) Then .Parameters.Add("@xx2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx2", SqlDbType.VarChar).Value = xx2 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx6))) Then .Parameters.Add("@xx6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx6", SqlDbType.VarChar).Value = xx6 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx10))) Then .Parameters.Add("@xx10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx10", SqlDbType.VarChar).Value = xx10 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx3))) Then .Parameters.Add("@xx3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx3", SqlDbType.VarChar).Value = xx3 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx7))) Then .Parameters.Add("@xx7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx7", SqlDbType.VarChar).Value = xx7 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(xx11))) Then .Parameters.Add("@xx11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx11", SqlDbType.VarChar).Value = xx11 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(xx12))) Then .Parameters.Add("@xx12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx12", SqlDbType.VarChar).Value = xx12 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yx1))) Then .Parameters.Add("@yy1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy1", SqlDbType.VarChar).Value = yy1 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy5))) Then .Parameters.Add("@yy5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy5", SqlDbType.VarChar).Value = yy5 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx13))) Then .Parameters.Add("@xx13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx13", SqlDbType.VarChar).Value = xx13 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy2))) Then .Parameters.Add("@yy2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy2", SqlDbType.VarChar).Value = yy2 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy6))) Then .Parameters.Add("@yy6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy6", SqlDbType.VarChar).Value = yy6 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx14))) Then .Parameters.Add("@xx14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx14", SqlDbType.VarChar).Value = xx14 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy3))) Then .Parameters.Add("@yy3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy3", SqlDbType.VarChar).Value = yy3 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy7))) Then .Parameters.Add("@yy7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy7", SqlDbType.VarChar).Value = yy7 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(xx15))) Then .Parameters.Add("@xx15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@xx15", SqlDbType.VarChar).Value = xx15 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy4))) Then .Parameters.Add("@yy4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy4", SqlDbType.VarChar).Value = yy4 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy8))) Then .Parameters.Add("@yy8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy8", SqlDbType.VarChar).Value = yy8 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(yy9))) Then .Parameters.Add("@yy9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy9", SqlDbType.VarChar).Value = yy9 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy13))) Then .Parameters.Add("@yy13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy13", SqlDbType.VarChar).Value = yy13 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb2))) Then .Parameters.Add("@b2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b2", SqlDbType.VarChar).Value = bb2 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yy10))) Then .Parameters.Add("@yy10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy10", SqlDbType.VarChar).Value = yy10 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy14))) Then .Parameters.Add("@yy14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy14", SqlDbType.VarChar).Value = yy14 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb3))) Then .Parameters.Add("@b3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b3", SqlDbType.VarChar).Value = bb3 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yy11))) Then .Parameters.Add("@yy11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy11", SqlDbType.VarChar).Value = yy11 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(yy15))) Then .Parameters.Add("@yy15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy15", SqlDbType.VarChar).Value = yy15 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb4))) Then .Parameters.Add("@b4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b4", SqlDbType.VarChar).Value = bb4 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(yy12))) Then .Parameters.Add("@yy12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@yy12", SqlDbType.VarChar).Value = yy12 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb1))) Then .Parameters.Add("@b1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b1", SqlDbType.VarChar).Value = bb1 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb5))) Then .Parameters.Add("@b5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b5", SqlDbType.VarChar).Value = bb5 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(bb6))) Then .Parameters.Add("@b6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b6", SqlDbType.VarChar).Value = bb6 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb10))) Then .Parameters.Add("@b10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b10", SqlDbType.VarChar).Value = bb10 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb14))) Then .Parameters.Add("@b14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b14", SqlDbType.VarChar).Value = bb14 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb7))) Then .Parameters.Add("@b7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b7", SqlDbType.VarChar).Value = bb7 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb11))) Then .Parameters.Add("@b11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b11", SqlDbType.VarChar).Value = bb11 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb15))) Then .Parameters.Add("@b15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b15", SqlDbType.VarChar).Value = bb15 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb8))) Then .Parameters.Add("@b8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b8", SqlDbType.VarChar).Value = bb8 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb12))) Then .Parameters.Add("@b12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b12", SqlDbType.VarChar).Value = bb12 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb0_1))) Then .Parameters.Add("@b0_1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_1", SqlDbType.VarChar).Value = bb0_1 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb9))) Then .Parameters.Add("@b9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b9", SqlDbType.VarChar).Value = bb9 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb13))) Then .Parameters.Add("@b13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b13", SqlDbType.VarChar).Value = bb13 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb0_2))) Then .Parameters.Add("@b0_2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_2", SqlDbType.VarChar).Value = bb0_2 End If </pre>

<pre> If String.IsNullOrEmpty(Trim (CStr(bb3))) Then .Parameters.Add("@b0_3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_3", SqlDbType.VarChar).Value = bb0_3 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb7))) Then .Parameters.Add("@b0_7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_7", SqlDbType.VarChar).Value = bb0_7 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb11))) Then .Parameters.Add("@b0_11" , SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_11" , SqlDbType.VarChar).Value = bb0_11 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb4))) Then .Parameters.Add("@b0_4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_4", SqlDbType.VarChar).Value = bb0_4 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb8))) Then .Parameters.Add("@b0_8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_8", SqlDbType.VarChar).Value = bb0_8 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb12))) Then .Parameters.Add("@b0_12" , SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_12" , SqlDbType.VarChar).Value = bb0_12 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb5))) Then .Parameters.Add("@b0_5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_5", SqlDbType.VarChar).Value = bb0_5 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb9))) Then .Parameters.Add("@b0_9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_9", SqlDbType.VarChar).Value = bb0_9 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb13))) Then .Parameters.Add("@b0_13" , SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_13" , SqlDbType.VarChar).Value = bb0_13 End If </pre>
<pre> If String.IsNullOrEmpty(Trim (CStr(bb6))) Then .Parameters.Add("@b0_6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_6", SqlDbType.VarChar).Value = bb0_6 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb10))) Then .Parameters.Add("@b0_10" , SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@b0_10" , SqlDbType.VarChar).Value = bb0_10 End If </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(bb14))) Then .Parameters.Add("@b0_14" , SqlDbType.VarChar).Value = System.DBNull.Value </pre>

Else	cmb_name.Visible = True	"='0')then('1')else(convert(float,c.yy" & name_id & ")end)/(count(v.vrb_y) - 1),6) as 'Adj-R2' " & _
.Parameters.Add("@b0_14", SqlDbType.VarChar).Value = bb0_14	End Sub	
End If	Protected Sub show_data_cal()	",round(power(case when(" & _
If String.IsNullOrEmpty(Trim(CStr(bb15))) Then	PictureBox13.Visible = True	"case
.Parameters.Add("@b0_15", SqlDbType.VarChar).Value = System.DBNull.Value	Label7.Visible = True	when((count(v.vrb_y)-2) <= 0)then
Else	Label6.Visible = True	(0)else((convert(float,c.yy" & name_id & ") -
.Parameters.Add("@b0_15", SqlDbType.VarChar).Value = System.DBNull.Value	Try	convert(float,c.b" & name_id & "
End If	'R	")*convert(float,c.yx" & name_id & ")
.Parameters.Add("@b0_15", SqlDbType.VarChar).Value = bb0_15	dsr.Clear()	/(count(v.vrb_y)-2)end " & _
End If	sql = "select x.vrb_x as name, " & _	-
.ExecuteNonQuery()	"round(power((case when((convert(float,c.b" & name_id & ")*convert(float,c.yx" & name_id & "))/convert(float,c.yy" & name_id & ")))<'0' then('0' " & _	"<= 0)then'0'
End With	"	else ((convert(float,c.yy" & name_id & ") -
End Sub	else(convert(float,c.b" & name_id & ")*convert(float,c.yx" & name_id & "))/convert(float,c.yy" & name_id & ")end),0.5),6) as R, " & _	convert(float,c.b" & name_id & ")*convert(float,c.yx" & name_id & ")
Private Sub pt_close_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pt_close.Click	close_edit()	/(count(v.vrb_y) - 2)end ,0.5),6) as 'Std Error' " & _
End Sub	Protected Sub close_edit()	"from VRB_cal
DataGridview2.Visible = False	"round(((convert(float,c.b" & name_id & ")*convert(float,c.yx" & name_id & "))/convert(float,c.yy" & name_id & ")),6) as R2, " & _	c left join
pt_save_edit.Visible = False	"round(1-(convert(float,c.yy" & name_id & ")-	VRB_VARIABLE v
pt_close.Visible = False	(convert(float,c.b" & name_id & ")*convert(float,c.yx" & name_id & "))/((case when((count(v.vrb_y) - 2)=0) " & _	with(nolock) on
pt_save.Visible = True	"then('1')else((count(v.vrb_y) - 2))end)/ " & _	c.id_vrb=v.id_vrb " & _
pt_add.Visible = True	"((case when(convert(float,c.yy" & name_id & ")	"left join
pt_report.Visible = True	when(convert(float,c.yx" & name_id & " " & _	VRB_X x with(nolock) on
pt_edit.Visible = True	"but_Search.Visible = True	c.id_vrb=x.id_vrb " & _
Dim iij As Integer		"where
For iij = 0 To 4		c.id_vrb=" " & lbl_id1.Text & " " and x.vrb_x=" " & name_x & " " " & _
		"group by c.b" & name_id & ",c.yx" & name_id & ",c.yy" & name_id & ",x.vrb_x "
		da.SelectCommand = New SqlCommand(sql, conn1)
		da.Fill(dsr, "aaaa")
		data_R.DataSource = dsr.Tables("aaaa")
		data_R.Columns(iij).Width = "45"

loat,c.yy" & name_id & ")))end))),0.5) " & _ " * power(convert(float,(count(v.vrb_y)-2)),0.5))/ " & _ " case when(power ((1-(case when((convert(float,c.b" & name_id & "))*convert(float,c.yx" & name_id & ")))='0')then('1')else((convert(float,c.b" & name_id & "))*convert(float,c.yx" & name_id & ")))end / " & _ " (case when((convert(float,c.yy" & name_id & ")))='0')then('1')else((convert(float,c.yy" & name_id & ")))end))),0.5)='0')then ('1') " & _ " else(power ((1-(case when((convert(float,c.b" & name_id & "))*convert(float,c.yx" & name_id & ")))='0')then('1')else((convert(float,c.b" & name_id & "))*convert(float,c.yx" & name_id & ")))end / " & _ " (case when((convert(float,c.yy" & name_id & ")))='0')then('1')else((convert(float,c.yy" & name_id & ")))end))),0.5)end ,6) as T , " & _ " case when(substring(c.b" & name_id & ",1,6)=convert(float,0))then('0.000')else(substring(c.b" & name_id & ",1,6))end as b1 " & _ " from VRB_cal c left join VRB_VARIABLE v with(nolock) on c.id_vrb=v.id_vrb left join VRB_X x with(nolock) on c.id_vrb=x.id_vrb " & _ " where c.id_vrb="" & lbl_id1.Text & "" and x.vrb_x="" & name_x & "" group by c.b"	& name_id & ",c.yx" & name_id & ",c.yy" & name_id & ",x.vrb_x " With com .CommandType = CommandType.Text .CommandText = sql .Connection = conn1 dr = .ExecuteReader() If dr.HasRows Then dr.Read() data_T.Rows(0).Cells(0).Value = dr.Item("name") data_T.Rows(0).Cells(1).Value = dr.Item("b1") data_T.Rows(0).Cells(2).Value = "TC" data_T.Rows(0).Cells(3).Value = dr.Item("T") data_T.Columns(0).HeaderText = "Name" data_T.Columns(1).HeaderText = "B" data_T.Columns(2).HeaderText = "TC" data_T.Columns(3).HeaderText = "T" dr.Close() Else dr.Close() data_T.Rows(0).Cells(0).Value = 0 data_T.Rows(0).Cells(1).Value = 0 data_T.Rows(0).Cells(2).Value = 0 data_T.Rows(0).Cells(3).Value = 0	data_T.Columns(0).HeaderText = "Name" data_T.Columns(1).HeaderText = "B" data_T.Columns(2).HeaderText = "TC" data_T.Columns(3).HeaderText = "T" 'data_T.Rows.Clear() End If End With Dim iif As Integer For iif = 0 To 4 data_R.Columns(iif).Width = "76" data_R.Columns(iif).Width = "76" data_R.Columns(iif).Width = "76" Next data_R.Visible = True : data_F.Visible = True : data_T.Visible = True show_xy1() End Sub Protected Sub show_xy1() Dim xlsApp As New Application xlsApp.Quit() xlsBook = xlsApp.Workbooks.Open("C:/vrb.xlsx") xlsSheet = xlsBook.Worksheets("Data") ds2.Clear() DataGridView10.DataSource = Nothing
---	---	--

```

Dim sql1 As String =
"select vrb_y,vrb_x" &
name_id & " as a from
VRB_VARIABLE where
id_vrb="" & lbl_id1.Text &
"" order by id_no asc"
da.SelectCommand =
New SqlCommand(sql1,
conn1)
da.Fill(ds2, "vrb_y")

DataGridView10.DataSou
rce = ds2.Tables("vrb_y")

Dim a As Integer = 1
For i = 0 To
DataGridView10.Rows.Cou
nt - 2

Try

With
DataGridView10
a = a + 1
If
String.IsNullOrEmpty(Trim
(CStr(DataGridView10.Ro
ws(i).Cells(0).Value))) Then
xlsSheet.Cells(a, 2) = "0"
Else
xlsSheet.Cells(a, 2) =
DataGridView10.Rows(i).C
ells(0).Value.ToString
End If

If
String.IsNullOrEmpty(Trim
(CStr(DataGridView10.Ro
ws(i).Cells(1).Value))) Then
xlsSheet.Cells(a, 3) = "0"
Else
xlsSheet.Cells(a, 3) =
DataGridView10.Rows(i).C
ells(1).Value.ToString
End If

End With
Catch ex As
Exception

End Try
Next

Try
xlsSheet =
xlsBook.Worksheets("Data"
).ChartObjects("v1").Chart.
Export("C:\Chart2.gif",
"gif")
Catch ex As Exception

End Try

Dim procs1 As
Process() =
Process.GetProcessesByNa
me("EXCEL")
For Each p As Process
In procs1
p.Kill()
Next

PictureBox13.ImageLocatio
n = ("C:\Chart2.gif")

sql = "select distinct
v.id_no,convert(float,v.vrb_
y) as
y,convert(float,v.vrb_x" &
name_id & ") as x " & _
",case
when(substring(c.b0_" &
name_id &
",1,6)=convert(float,0))then(
'0.000')else(substring(c.b0_"
& name_id & ",1,6))end as
b0 " & _
",case
when(substring(c.b" &
name_id &
",1,6)=convert(float,0))then(
'0.000')else(substring(c.b" &
name_id & ",1,6))end as b1
" & _
",x.vrb_x as
names,x.vrb_unit as unit "
& _
" from
VRB_VARIABLE v
with(nolock) left join
VRB_cal c with(nolock) on
v.id_vrb=c.id_vrb left join
VRB_X x with(nolock) on
c.id_vrb=x.id_vrb " & _
" where v.id_vrb = ""
& lbl_id1.Text & "" and
x.id="" & name_xs & ""
order by v.id_no asc"

da.SelectCommand =
New SqlCommand(sql,
conn1)
Dim ds As New
DataSet
da.Fill(ds, "Data")

DataGridView7.DataSource
= ds.Tables("Data")

Try
Label6.Text =
DataGridView7.Rows(1).Ce
lls(5).Value.ToString & "("
&
DataGridView7.Rows(1).Ce
lls(6).Value.ToString & ")"
Label4.Text = "The
relation between PI and " &
DataGridView7.Rows(1).Ce
lls(5).Value.ToString
Label5.Text = "PI
(STB/Psi/Day)"

Dim n, m As String
Dim abs3 As Double
abs3 =
DataGridView7.Rows(1).Ce
lls(4).Value

If
DataGridView7.Rows(1).Ce
lls(4).Value.ToString < 0
Then
abs3 =
Math.Abs(abs3)
If abs3 = 0.0 Then
m = "0.000"
Else
m = abs3
End If
n = " - " & abs3
& " X "
Else
abs3 =
Math.Abs(abs3)
If abs3 = 0.0 Then
m = "0.000"
Else
m = abs3
End If
n = " + " & m & "
X "

End If
Label7.Text = "Y =
" &

```

```

DataGridView7.Rows(1).Cells(3).Value.ToString & n
'DataGridView7.Rows(1).Cells(4).Value.ToString & "
X "

    Catch ex As Exception
        Label6.Text = ""
        Label7.Text = ""
    End Try

End Sub
Private Sub
ButtonX1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX1.Click
    name_x =
ButtonX1.Text
    name_id = "1"
    name_xs = "01"
    show_data_cal()
End Sub
Private Sub
ButtonX2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX2.Click
    name_x =
ButtonX2.Text
    name_id = "2"
    name_xs = "02"
    show_data_cal()
End Sub

Private Sub
ButtonX3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX3.Click
    name_x =
ButtonX3.Text
    name_id = "3"
    name_xs = "03"
    show_data_cal()
End Sub

Private Sub
ButtonX4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX4.Click
    name_x =
ButtonX4.Text
    name_id = "4"
    name_xs = "04"
    show_data_cal()
End Sub

Private Sub
ButtonX5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX5.Click
    name_x =
ButtonX5.Text
    name_id = "5"
    name_xs = "05"
    show_data_cal()
End Sub

Private Sub
ButtonX6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX6.Click
    name_x =
ButtonX6.Text
    name_id = "6"
    name_xs = "06"
    show_data_cal()
End Sub

Private Sub
ButtonX7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX7.Click
    name_x =
ButtonX7.Text
    name_id = "7"
    name_xs = "07"
    show_data_cal()
End Sub

Private Sub
ButtonX8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX8.Click
    name_x =
ButtonX8.Text
    name_id = "8"
    name_xs = "08"
    show_data_cal()
End Sub

End Sub

Private Sub
ButtonX9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX9.Click
    name_x =
ButtonX9.Text
    name_id = "9"
    name_xs = "09"
    show_data_cal()
End Sub

Private Sub
ButtonX10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX10.Click
    name_x =
ButtonX10.Text
    name_id = "10"
    name_xs = "10"
    show_data_cal()
End Sub

Private Sub
ButtonX11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX11.Click
    name_x =
ButtonX11.Text
    name_id = "11"
    name_xs = "11"
    show_data_cal()
End Sub

Private Sub
ButtonX12_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX12.Click
    name_x =
ButtonX12.Text
    name_id = "12"
    name_xs = "12"
    show_data_cal()
End Sub

Private Sub
ButtonX13_Click(ByVal sender As System.Object,

```

<pre> ByVal e As System.EventArgs) Handles ButtonX13.Click name_x = ButtonX13.Text name_id = "13" name_xs = "13" show_data_cal() End Sub Private Sub ButtonX14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX14.Click name_x = ButtonX14.Text name_id = "14" name_xs = "14" show_data_cal() End Sub Private Sub ButtonX15_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX15.Click name_x = ButtonX15.Text name_id = "15" name_xs = "15" show_data_cal() End Sub Private Sub but_Search1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) If cmb_name.SelectedValue = "" Then Exit Sub End If lbl_id1.Text = cmb_name.SelectedValue.T oString If cmb_name.Text = "" Then Exit Sub End If ButtonX1.Text = "" : ButtonX2.Text = "" : </pre>	<pre> ButtonX3.Text = "" : ButtonX4.Text = "" : ButtonX5.Text = "" : ButtonX6.Text = "" : ButtonX7.Text = "" ButtonX8.Text = "" : ButtonX9.Text = "" : ButtonX10.Text = "" : ButtonX11.Text = "" : ButtonX12.Text = "" : ButtonX13.Text = "" : ButtonX14.Text = "" : ButtonX15.Text = "" ButtonX1.Visible = True : ButtonX2.Visible = True : ButtonX3.Visible = True : ButtonX4.Visible = True : ButtonX5.Visible = True : ButtonX6.Visible = True : ButtonX7.Visible = True ButtonX8.Visible = True : ButtonX9.Visible = True : ButtonX10.Visible = True : ButtonX11.Visible = True : ButtonX12.Visible = True : ButtonX13.Visible = True : ButtonX14.Visible = True : ButtonX15.Visible = True datagrid_cal() End Sub Protected Sub datagrid_cal() sql = "select vrb_y,vrb_x,id_no from VRB_X with(nolock) where vrb_name= "" & cmb_name1.Text & "" order by id_no" da.SelectCommand = New SqlCommand(sql, conn1) Dim ds As New DataSet da.Fill(ds, "Data") DataGridViewX3.DataSou rce = ds.Tables("Data") </pre>	<pre> For i = 0 To DataGridViewX3.Rows.Co unt - 1 If i = 0 Then If DataGridViewX3.Rows(i). Cells(1).Value IsNot Nothing Then ButtonX1.Text = DataGridViewX3.Rows(i). Cells(1).Value.ToString End If End If If i = 1 Then If DataGridViewX3.Rows(i). Cells(1).Value IsNot Nothing Then ButtonX2.Text = DataGridViewX3.Rows(i). Cells(1).Value.ToString End If End If If i = 2 Then If DataGridViewX3.Rows(i). Cells(1).Value IsNot Nothing Then ButtonX3.Text = DataGridViewX3.Rows(i). Cells(1).Value.ToString End If End If If i = 3 Then If DataGridViewX3.Rows(i). Cells(1).Value IsNot Nothing Then ButtonX4.Text = DataGridViewX3.Rows(i). Cells(1).Value.ToString End If End If If i = 4 Then If DataGridViewX3.Rows(i). Cells(1).Value IsNot Nothing Then </pre>
---	---	---

```

        ButtonX5.Text
    =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 5 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX6.Text
    =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 6 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX7.Text
    =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 7 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX8.Text
    =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 8 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX9.Text
    =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 9 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX10.Text =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 10 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX11.Text =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 11 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX12.Text =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 12 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX13.Text =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    If i = 13 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX14.Text =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    End If

    If i = 14 Then
        If
    DataGridViewX3.Rows(i).
    Cells(1).Value IsNot
    Nothing Then
        ButtonX15.Text =
    DataGridViewX3.Rows(i).
    Cells(1).Value.ToString
        End If
    End If

    Next

    End Sub

    Private Sub
    RibbonTabItem2_Click(By
    Val sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    RibbonTabItem2.Click
        dgv_save.Visible =
    False
        tag_result = ""

        lbl_id.Text =
    max_id_name
        GroupBox2.Visible =
    False "ไม่แสดงค่า R T F

        cmb_name.ContextMenu =
    New ContextMenu()
        Dim sqlcmb_name As
    String = "select distinct
    id_vrb,vrb_name,vrb_y
    from VRB_X order by
    id_vrb desc" 'group by
    id_vrb,vrb_name"
        dacmb_name = New
    SqlDataAdapter(sqlcmb_na
    me, conn1)
        dscmb_name.Clear()

        dacmb_name.Fill(dscmb_na
    me, "VRB_X")
        With cmb_name
            .DataSource =
    dtcmb_name
            .DisplayMember =
    "vrb_name"
            .ValueMember =
    "id_vrb"

```

```

.DataSource = dr.Close()
dscmb_name.Tables("VRB_X")
End With
pt_save.Visible = True
pt_add.Visible = True
pt_report.Visible = True
pt_edit.Visible = True
show_data_view()
show_head()
End Sub
Private Sub pt_add_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pt_add.Click
dgv_save.Visible = True
sql = "SELECT distinct count(id_no) AS max_id_vrb FROM VRB_VARIABLE where id_vrb= '" & cmb_name.SelectedValue & "'"
Try
With com
.CommandType = CommandType.Text
.CommandText = sql
.Connection = conn1
dr = .ExecuteReader()
dr.Read()
add_x1 = CInt(dr.Item("max_id_vrb") + 1)
add_x = CInt(dr.Item("max_id_vrb") + 1) + 1 * (CInt(max_id_name) + 1).ToString("0000")
End With
Catch
End Try
dr.Close()
datagrid1()
End Sub
Private Sub RibbonTabItem3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RibbonTabItem3.Click
tag_result = ""
cmb_name1.ContextMenu = New ContextMenu()
Dim sqlcmb_name As String = "select distinct id_vrb,vrb_name,vrb_y from VRB_X order by id_vrb desc"
dacmb_name = New SqlDataAdapter(sqlcmb_name, conn1)
dscmb_name.Clear()
dacmb_name.Fill(dscmb_name, "VRB_X")
With cmb_name1
.DataSource = dtcmb_name
.DisplayMember = "vrb_name"
.ValueMember = "id_vrb"
.DataSource = dscmb_name.Tables("VRB_X")
End With
If max_id_name = ""
Then
Else
cmb_name1.SelectedValue = max_id_name
End If
lbl_id.Text = max_id_name
If lbl_id.Text = ""
Then
If
cmb_name1.SelectedValue = "" Then
Exit Sub
Else
lbl_id.Text = cmb_name1.SelectedValue
End If
End Try

```



```

        lbl_id1.Text =
cmb_name1.SelectedValu
End If

End If

ButtonX1.Text = "" :
ButtonX2.Text = "" :
ButtonX3.Text = "" :
ButtonX4.Text = "" :
ButtonX5.Text = "" :
ButtonX6.Text = "" :
ButtonX7.Text = ""
ButtonX8.Text = "" :
ButtonX9.Text = "" :
ButtonX10.Text = "" :
ButtonX11.Text = "" :
ButtonX12.Text = "" :
ButtonX13.Text = "" :
ButtonX14.Text = "" :
ButtonX15.Text = ""

ButtonX1.Visible =
True : ButtonX2.Visible =
True : ButtonX3.Visible =
True : ButtonX4.Visible =
True : ButtonX5.Visible =
True : ButtonX6.Visible =
True : ButtonX7.Visible =
True
ButtonX8.Visible =
True : ButtonX9.Visible =
True : ButtonX10.Visible =
True : ButtonX11.Visible =
True : ButtonX12.Visible =
True : ButtonX13.Visible =
True : ButtonX14.Visible =
True : ButtonX15.Visible =
True

datagrid_cal()

End Sub

Private Sub
cmb_name_SelectedIndexC
hanged(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
cmb_name.SelectedIndexCh
anged

dgv_save.Visible =
False

GroupBox2.Visible =
False "ไม่แสดงค่า R T F
        lbl_id.Text =
cmb_name.SelectedValu

If cmb_name.Text = ""
Then

If
cmb_name.SelectedValu =
"" Then
Exit Sub

Else
lbl_id.Text =
cmb_name.SelectedValu
End If

End If

sql = "SELECT
distinct count(id) AS
max_id_vrb FROM VRB_X
where id_vrb= " &
max_id_name & """
Try
With com
.CommandType =
CommandType.Text
.CommandText =
sql
.Connection =
conn1
dr =
.ExecuteReader()
dr.Read()
add_x1 =
CInt(dr.Item("max_id_vrb")
) + 1
add_x =
CInt(dr.Item("max_id_vrb")
) + 1 *(CInt(max_id_name)
+ 1).ToString("0000")
End With
Catch

End Try
dr.Close()

pt_save.Visible = True
pt_add.Visible = True
pt_report.Visible =
True
pt_edit.Visible = True

show_data_view()
show_head()
End Sub

Private Sub
cmb_name1_SelectedIndex
Changed(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
cmb_name1.SelectedIndexC
hanged

lbl_id.Text =
max_id_name

PictureBox13.Visible =
False
Label7.Visible = False
Label6.Visible = False

If lbl_id.Text = ""
Then
If
cmb_name1.SelectedValu
= "" Then

Exit Sub
Else
lbl_id.Text =
cmb_name1.SelectedValu
End If

End If

lbl_id1.Text =
cmb_name1.SelectedValu

ButtonX1.Text = "" :
ButtonX2.Text = "" :
ButtonX3.Text = "" :
ButtonX4.Text = "" :
ButtonX5.Text = "" :
ButtonX6.Text = "" :
ButtonX7.Text = ""
ButtonX8.Text = "" :
ButtonX9.Text = "" :
ButtonX10.Text = "" :
ButtonX11.Text = "" :
ButtonX12.Text = "" :
ButtonX13.Text = "" :
ButtonX14.Text = "" :
ButtonX15.Text = ""

ButtonX1.Visible =
True : ButtonX2.Visible =
True : ButtonX3.Visible =
True : ButtonX4.Visible =
True : ButtonX5.Visible =
True : ButtonX6.Visible =
True : ButtonX7.Visible =
True

```

```

ButtonX8.Visible =
True : ButtonX9.Visible =
True : ButtonX10.Visible =
True : ButtonX11.Visible =
True : ButtonX12.Visible =
True : ButtonX13.Visible =
True : ButtonX14.Visible =
True : ButtonX15.Visible =
True

datagrid_cal()
End Sub
Private Sub
ComboBoxEx1_SelectedInd
exChanged(ByVal sender
As System.Object, ByVal e
As System.EventArgs)
    If
cmb_name1.SelectedValue
= "" Then
        Exit Sub
    End If
    lbl_id1.Text =
cmb_name1.SelectedValue.
ToString

    If cmb_name1.Text =
"" Then
        Exit Sub
    End If
End Sub
Private Sub
cmb_name2_SelectedIndex
Changed(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
cmb_name2.SelectedIndexC
hanged
    Button7.Visible =
False

    If tag_data_x = "show"
Then
    Else
        tag_result = "Start"

        vrbx = 0
        count_x = 0
        Column_ex = 3

DataGridViewX1.Rows.Cle
ar()

DataGridView4.ClearSelect
ion()

        max_s()
        Dim xlsApp As New
Application

        xlsApp.Quit()
        xlsBook =
xlsApp.Workbooks.Open("
C:/vrb.xlsx")

        max_s()

DataGridView8.ClearSelect
ion()

        dx.Clear()
        sql = " select
x.vrb_x,r.r from VRB_r r
with(nolock) " & _
" left join
VRB_X x with(nolock) on
substring(r.r_no,5,2)=x.id
and substring(r.r_no,1,4)=
x.id_vrb " & _
" where
x.id_vrb= " & lbl_id.Text
& "" and (r.status is null or
r.status=)and (r.r <>'0') "
& _
" order by r.r
desc"
        da.SelectCommand
= New SqlCommand(sql,
conn1)
        da.Fill(dx, "bx")

DataGridView8.DataSource
= dx.Tables("bx")

Try

        For i = 0 To 5
            name_x1 =
DataGridView8.Rows(0).Ce
lls(0).Value.ToString
            name_x2 =
DataGridView8.Rows(1).Ce
lls(0).Value.ToString
            name_x3 =
DataGridView8.Rows(2).Ce
lls(0).Value.ToString
            name_x4 =
DataGridView8.Rows(3).Ce
lls(0).Value.ToString
            name_x5 =
DataGridView8.Rows(4).Ce
lls(0).Value.ToString

        DataGrid View3.ClearSelect
ion()

        If lbl_id.Text = ""
Then
            If
cmb_name2.SelectedValue
= "" Then
                Exit Sub
            Else
                lbl_id.Text =
cmb_name2.SelectedValue.
ToString
            Catch ex As
Exception
                Exit Sub
            End Try

            sql = "update VRB_r
set status=" where
substring(r_no,1,4) = " &
lbl_id.Text & ""
            With com
                .Connection =
conn1
                .CommandType =
CommandType.Text
                .CommandText =
sql
            End With

            ds2.Clear()
            sql = "select vrb_y
from VRB_VARIABLE
where id_vrb=" &
lbl_id.Text & "" order by
id_no asc"
            da.SelectCommand
= New SqlCommand(sql,
conn1)
            da.Fill(ds2, "vrb_y")

DataGridView3.DataSource
= ds2.Tables("vrb_y")

            show_x()

```

```

        Next
        Catch ex As
Exception
        End Try
        vrb_y()

        Dim procs As
Process() =
Process.GetProcessesByNa
me("EXCEL")
        For Each p As
Process In procs
            p.Kill()
            Next
            End If

            If
DataGridViewX1.Rows.Co
unt = 1 Then
                Else
                    Button7.Visible =
True
                End If

                End Sub
                Private Sub
RibbonTabItem4_Click(By
Val sender As
System.Object, ByVal e As
System.EventArgs) Handles
RibbonTabItem4.Click
                    Button7.Visible =
False

                    tag_result = "Start"
                    tag_data_x = "show"

                    cmb_name2.ContextMenu =
New ContextMenu()
                    Dim sqlcmb_name As
String = "select distinct
id_vrb,vrb_name,vrb_y
from VRB_X order by
id_vrb desc"
                    dacmb_name = New
SqlDataAdapter(sqlcmb_na
me, conn1)
                    dscmb_name.Clear()

                    dacmb_name.Fill(dscmb_na
me, "VRB_X")
                    With cmb_name2
                        .DataSource =
dscmb_name
                        .DisplayMember =
"vrb_name"
                        .ValueMember =
"id_vrb"
                        .DataSource =
dscmb_name.Tables("VRB
_X")
                        End With

                        If max_id_name = ""
Then
                            Else
                                cmb_name2.SelectedValue
= max_id_name
                                End If

                                vrbx = 0
                                count_x = 0
                                Column_ex = 3

                                DataGridViewX1.Rows.Cle
ar()

                                DataGridView4.ClearSelect
ion()

                                DataGridView3.ClearSelect
ion()

                                If lbl_id.Text = ""
Then
                                    If
                                        cmb_name2.SelectedValue
= "" Then
                                            MsgBox("Please
enter the message.")
                                            Exit Sub
                                        Else
                                            lbl_id.Text =
cmb_name2.SelectedValue
                                            End If
                                        End If

                                        Try
                                            lbl_name.Text =
cmb_name2.Text
                                            lbl_id.Text =
cmb_name2.SelectedValue.
ToString
                                            Catch ex As Exception
                                                Exit Sub
                                            End Try

                                            sql = "update VRB_r
set status=" where
                                substring(r_no,1,4) = "" &
lbl_id.Text & ""
                                With com
                                    .Connection = conn1
                                    .CommandType =
CommandType.Text
                                    .CommandText = sql
                                    .ExecuteNonQuery()
                                End With

                                ds2.Clear()
                                sql = "select vrb_y
from VRB_VARIABLE
where id_vrb="" &
lbl_id.Text & "" order by
id_no asc"
                                da.SelectCommand =
New SqlCommand(sql,
conn1)
                                da.Fill(ds2, "vrb_y")

                                DataGridView3.DataSource
= ds2.Tables("vrb_y")

                                show_x()
                                max_s()
                                Dim xlsApp As New
Application

                                xlsApp.Quit()
                                xlsBook =
xlsApp.Workbooks.Open("
C:/vrb.xlsx")

                                max_s()

                                DataGridView8.ClearSelect
ion()

                                dx.Clear()
                                sql = " select
x.vrb_x,r.r from VRB_r r
with(nolock) " & _
                                " left join VRB_X
x with(nolock) on
substring(r.r_no,5,2)=x.id
and substring(r.r_no,1,4)=
x.id_vrb " & _
                                " where x.id_vrb=
"" & lbl_id.Text & "" and
(r.status is null or
r.status='')and (r.r <>'0') "
                                & _
                                " order by r.r desc"

```

da.SelectCommand = New SqlCommand(sql, conn1)	x15,x.x16,x.x17,x.x18,x.x1 9,x.x20 " & _	End If End With Next
da.Fill(dx, "bx")	",x.x21,x.x22,x.x23,x.x24,x. x25,x.x26,x.x27,x.x28,x.x2 9,x.x30,r.r_no,xx.vrb_x " & _	vrb_x() End Sub Protected Sub vrb_x()
DataGridView8.DataSource = dx.Tables("bx")	" from VRB_r r with(nolock) " & _	For n = count_x To count_x With DataGridView4
Try	" left join VRB_x1 x with(nolock) on r.r_no=x.r_no " & _	Dim ij As Integer = 1
For i = 0 To 5 name_x1 = DataGridView8.Rows(0).Ce lls(0).Value.ToString	" left join VRB_X xx with(nolock) on substring(x.r_no,1,4)=xx.id _vrb and substring(x.r_no,5,2)=xx.id " & _	max_s() For j = 0 To max_vrb - 1 ij = ij + 1
DataGridView8.Rows(1).Ce lls(0).Value.ToString name_x2 = DataGridView8.Rows(2).Ce lls(0).Value.ToString	" where substring(r.r_no,1,4)=" & lbl_id.Text & " and (r.status is null or r.status=")and (r.r <>'0') order by r.r desc"	If String.IsNullOrEmpty(Trim (CStr(DataGridView4.Rows (count_x).Cells(j).Value))) Then
DataGridView8.Rows(3).Ce lls(0).Value.ToString name_x3 = DataGridView8.Rows(4).Ce lls(0).Value.ToString	da.SelectCommand = New SqlCommand(sql, conn1)	xlsSheet.Cells(ij, Column_ex) = "0"
Next Catch ex As Exception	da.Fill(ds3, "xx")	r_no = "" Else
End Try	DataGridView4.DataSource = ds3.Tables("xx")	xlsSheet.Cells(ij, Column_ex) = DataGridView4.Rows(count _x).Cells(j).Value.ToString
vrb_y()	End Sub Protected Sub vrb_y()	r_no = DataGridView4.Rows(count _x).Cells(30).Value.ToStrin g
Dim procs As Process() = Process.GetProcessesByNa me("EXCEL")	xlsSheet = xlsBook.Worksheets("Data")	End If
For Each p As Process In procs	Dim a As Integer = 1 For i = 0 To	Next
p.Kill() Next	DataGridView3.Rows.Coun t - 2	Dim oo As String oo =
tag_data_x = "" Button7.Visible = True	With DataGridView3	xlsSheet.Cells(ij, Column_ex).Value.ToString
End Sub Protected Sub show_x()	a = a + 1 If String.IsNullOrEmpty(Trim (CStr(DataGridView3.Rows (i).Cells(0).Value))) Then	End With Next
DataGridView4.ClearSelect ion()	xlsSheet.Cells(a, 2) = "0" Else	If (max_vrb = "0") Then
ds3.Clear() sql = "select x.x1,x.x2,x.x3,x.x4,x.x5,x.x 6,x.x7,x.x8,x.x9,x.x10 " & _ " ,x.x11,x.x12,x.x13,x.x14,x.	xlsSheet.Cells(a, 2) = DataGridView3.Rows(i).Ce lls(0).Value.ToString	Exit Sub End If If (vrbx = "0") Then vrbx1() 'สมการ 1 End If

```

End Sub
Protected Sub
show_name_x()

    dx.Clear()
    sql = " select
x.vrb_x,r.r from VRB_r r
with(nolock) " & _
        " left join VRB_X
x with(nolock) on
substring(r.r_no,5,2)=x.id
and substring(r.r_no,1,4)=
x.id_vrb " & _
        " where x.id_vrb=
" & lbl_id.Text & " and
(r.status is null or
r.status='')and (r.r <>'0') "
& _
        " order by r.r desc"
da.SelectCommand =
New SqlCommand(sql,
conn1)
da.Fill(dx, "bx")

DataGridView8.DataSource
= dx.Tables("bx")

Try

    For i = 0 To 5
        name_x1 =
DataGridView8.Rows(0).Ce
lls(0).Value.ToString
        name_x2 =
DataGridView8.Rows(1).Ce
lls(0).Value.ToString
        name_x3 =
DataGridView8.Rows(2).Ce
lls(0).Value.ToString
        name_x4 =
DataGridView8.Rows(3).Ce
lls(0).Value.ToString
        name_x5 =
DataGridView8.Rows(4).Ce
lls(0).Value.ToString
    Next
Catch ex As Exception

End Try
End Sub
Protected Sub vrbx1()

    vrbx = vrbx + 1
    If vrbx <> max_r Then

        xlsSheet =
xlsBook.Worksheets("1")

        Dim aa As String
        aa =
xlsSheet.Cells(17,
19).Value.ToString
        If xlsSheet.Cells(17,
19).Value.ToString = "OK"
Then

            DataGridViewX1.Rows.Ad
d()

            Dim abs, con As
String
            abs =
CDBl(Math.Abs(xlsSheet.C
ells(11, 14).Value))

            If
xlsSheet.Cells(11,
14).Value < 0 Then
                con = " - "
            Else
                con = " + "
            End If

            show_name_x()

            DataGridViewX1.Rows(0).
Cells(0).Value = "PI = " &
Format(CDBl(xlsSheet.Cells
(11, 12).Value),
"#,##0.000000") & con & _
Format(CDBl(abs),
"#,##0.000000") & " (" &
name_x1 & ") "

            DataGridViewX1.Rows(0).
Cells(2).Value =
Format(CDBl(xlsSheet.Cells
(26, 12).Value),
"#,##0.000000") 'R2

            Dim YY, XX, err,
err1, ca As String
            Dim sum_err As
Double

            Dim ij As Integer
            = 1
            max_s()
            For j = 0 To
max_vrb - 1            ij
            = ij + 1

            xlsSheet =
xlsBook.Worksheets("Data"
)
            XX =
CDBl(xlsSheet.Cells(ij,
3).Value)
            YY =
CDBl(xlsSheet.Cells(ij,
2).Value)

            xlsSheet =
xlsBook.Worksheets("1")

            Dim aaa, bbb,
cal As String
            aaa =
CDBl(xlsSheet.Cells(11,
12).Value)
            bbb =
CDBl(Math.Abs(xlsSheet.C
ells(11, 14).Value))

            If
xlsSheet.Cells(11,
14).Value < 0 Then
                cal = " - "
            Else
                cal = " + "
            End If

            If cal = " - "
Then
                err =
Val(CDBl(xlsSheet.Cells(11
, 12).Value)) -
Val((CDBl(Math.Abs(xlsSh
eet.Cells(11, 14).Value)) *
XX)) 'error
            Else
                err =
Val(CDBl(xlsSheet.Cells(11
, 12).Value)) +
Val((CDBl(Math.Abs(xlsSh
eet.Cells(11, 14).Value)) *
XX)) 'error
            End If

            err1 =
(Math.Abs((YY - err)) *
100) / YY

            sum_err =
sum_err + err1
        Next
    End Sub

```

DataGridViewX1.Rows(0).Cells(4).Value = Format(CDbl(sum_err / max_vrb), "###0.000000")	xlsSheet = xlsBook.Worksheets("2")	"###0.000000") & " (" & name_x2 & ") " 'y
count_x = 1 Column_ex = 4 vrb_y() vrb_x() vrbx2()	Dim abs1, abs, con1, con As String abs = CDbl(Math.Abs(xlsSheet.Cells(4, 13).Value)) abs1 = CDbl(Math.Abs(xlsSheet.Cells(4, 16).Value))	DataGridViewX1.Rows(1).Cells(2).Value = Format(CDbl(xlsSheet.Cells(4, 21).Value), "###0.000000") 'R2 Dim YY, XX, XX1, err, err1 As String Dim sum_err As Double
ElseIf xlsSheet.Cells(17, 19).Value.ToString = "NO" Then	If xlsSheet.Cells(4, 13).Value < 0 Then con = " - " Else con = " + " End If	Dim ij As Integer = 1
Try	End If	max_s() For j = 0 To max_vrb -
where r_no = "" & DataGridView4.Rows(r1).Cells(30).Value.ToString & ""	If xlsSheet.Cells(4, 16).Value < 0 Then con1 = " - " Else con1 = " + " End If	ij = ij + 1 'loop
sql = "update VRB_r set status='N' where r_no = "" & r_no & "" "	show_name_x()	xlsSheet = xlsBook.Worksheets("Data")
With com .Connection	Dim aa, bb As String aa = xlsSheet.Cells(7, 20).Value.ToString bb = xlsSheet.Cells(9, 20).Value.ToString	XX = CDbl(xlsSheet.Cells(ij, 3).Value) XX1 = CDbl(xlsSheet.Cells(ij, 4).Value) YY = CDbl(xlsSheet.Cells(ij, 2).Value)
= conn1	End With	xlsSheet = xlsBook.Worksheets("2")
.CommandType = CommandType.Text	If (xlsSheet.Cells(7, 20).Value.ToString = "OK") And (xlsSheet.Cells(9, 20).Value.ToString = "OK") Then	Dim a, b, c, d, cal, cal1 As String a = CDbl(xlsSheet.Cells(4, 11).Value) b = CDbl(Math.Abs(xlsSheet.Cells(4, 13).Value)) c = CDbl(Math.Abs(xlsSheet.Cells(4, 16).Value))
.CommandText = sql	End If	If xlsSheet.Cells(4, 13).Value < 0 Then
.ExecuteNonQuery()	DataGridViewX1.Rows.Add()	cal = " - " Else cal = " + " End If
End With r1 = r1 + 1 Catch ex As Exception	DataGridViewX1.Rows(1).Cells(0).Value = "PI = " & Format(CDbl(xlsSheet.Cells(4, 11).Value), "###0.000000") & con & _	
End Try	Format(CDbl(abs), "###0.000000") & " (" & name_x1 & ") " & con1 & _	
show_x() vrb_y() vrb_x() vrbx1()	Format(CDbl(abs1),	
End If End If End Sub		
Protected Sub vrbx2()		
vrbx = vrbx + 1 If vrbx <> max_r Then		

<pre> If xlsSheet.Cells(4, 12).Value < 0 Then con = " - " Else con = " + " End If If xlsSheet.Cells(4, 15).Value < 0 Then con1 = " - " Else con1 = " + " End If If xlsSheet.Cells(4, 18).Value < 0 Then con2 = " - " Else con2 = " + " End If show_name_x() Dim aa, bb, cc As String aa = xlsSheet.Cells(7, 22).Value.ToString bb = xlsSheet.Cells(9, 22).Value.ToString cc = xlsSheet.Cells(11, 22).Value.ToString If (xlsSheet.Cells(7, 22).Value.ToString = "OK") And (xlsSheet.Cells(9, 22).Value.ToString = "OK") And (xlsSheet.Cells(11, 22).Value.ToString = "OK") Then DataGridViewX1.Rows.Ad d() DataGridViewX1.Rows(2). Cells(0).Value = "PI = " & Format(CDbl(xlsSheet.Cells (4, 10).Value.ToString()), "###0.000000") & con & _ Format(CDbl(abs), "###0.000000") & " (" & name_x1 & ") " _ & con1 & </pre>	<pre> Format(CDbl(abs1), "###0.000000") & " (" & name_x2 & ") " _ & con2 & Format(CDbl(abs2), "###0.000000") & " (" & name_x3 & ") " y DataGridViewX1.Rows(2). Cells(1).Value = Format(CDbl(xlsSheet.Cells (4, 21).Value.ToString()), "###0.000000") 'Fc DataGridViewX1.Rows(2). Cells(2).Value = Format(CDbl(xlsSheet.Cells (4, 23).Value.ToString()), "###0.000000") 'R2 DataGridViewX1.Rows(2). Cells(3).Value = Format(CDbl(xlsSheet.Cells (4, 24).Value.ToString()), "###0.000000") 'Strd Dim YY, XX, XX1, XX2, err, err1, err2 As String Dim sum_err As Double Dim ij As Integer = 1 max_s() For j = 0 To max_vrb - 1 ij = ij + 1 xlsSheet = xlsBook.Worksheets("Data") XX2 = CDbl(xlsSheet.Cells(ij, 5).Value) XX1 = CDbl(xlsSheet.Cells(ij, 4).Value) XX = CDbl(xlsSheet.Cells(ij, 3).Value) YY = CDbl(xlsSheet.Cells(ij, 2).Value) </pre>	<pre> xlsSheet = xlsBook.Worksheets("3") Dim a, b, c, d, cal, cal1, cal2 As String a = CDbl(xlsSheet.Cells(4, 10).Value) b = CDbl(Math.Abs(xlsSheet.C ells(4, 12).Value)) c = CDbl(Math.Abs(xlsSheet.C ells(4, 15).Value)) d = CDbl(Math.Abs(xlsSheet.C ells(4, 18).Value)) If xlsSheet.Cells(4, 12).Value < 0 Then cal = " - " Else cal = " + " End If If xlsSheet.Cells(4, 15).Value < 0 Then cal1 = " - " Else cal1 = " + " End If If xlsSheet.Cells(4, 18).Value < 0 Then cal2 = " - " Else cal2 = " + " End If If cal = " - " And cal1 = " - " And cal2 = " - " Then err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 15).Value)) * XX1)) - Val((CDbl(Math.Abs(xlsSh </pre>
---	--	---

<pre>eet.Cells(4, 18).Value)) * XX2)) 'error ElseIf cal = " - " And cal1 = " - " And cal2 = " + " Then err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 15).Value)) * XX1)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 18).Value)) * XX2)) 'error ElseIf cal = " - " And cal1 = " + " And cal2 = " + " Then err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 15).Value)) * XX1)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 18).Value)) * XX2)) 'error ElseIf cal = " + " And cal1 = " + " And cal2 = " + " Then err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 15).Value)) * XX1)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 18).Value)) * XX2)) 'error ElseIf cal = " + " And cal1 = " - " And cal2 = " - " Then err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) + </pre>	<pre>Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 15).Value)) * XX1)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 18).Value)) * XX2)) 'error ElseIf cal = " + " And cal1 = " - " And cal2 = " + " Then err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 15).Value)) * XX1)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 18).Value)) * XX2)) 'error ElseIf cal = " - " And cal1 = " + " And cal2 = " - " Then err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 18).Value)) * XX2)) 'error err = Val(CDbl(xlsSheet.Cells(4, 10).Value)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 12).Value)) * XX)) + Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 15).Value)) * XX1)) - Val((CDbl(Math.Abs(xlsSh eet.Cells(4, 18).Value)) * XX2)) 'error End If err1 = (Math.Abs((YY - err)) * 100) / YY sum_err = sum_err + err1 Next DataGridViewX1.Rows(2). Cells(4).Value =</pre>	<pre>Format(CDbl(sum_err / max_vrb), "#,##0.000000") count_x = 3 Column_ex = 6 vrb_y() vrb_x() vrbx4() Else If xlsSheet.Cells(7, 22).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx3() ElseIf xlsSheet.Cells(9, 22).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql</pre>
---	--	--

<pre> .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx3() 'vrbx4() Else Then If aaa = "1" Else Try sql = "update VRB_r set status='N' where r_no = "" & r_no & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx3() End If End If End If End Sub Protected Sub vrbx4() vrbx = vrbx + 1 </pre>	<pre> If vrbx <> max_r Then xlsSheet = xlsBook.Worksheets("4") Dim abs1, abs, abs2, abs3, con1, con, con2, con3 As String abs = CDBl(Math.Abs(xlsSheet.C ells(4, 15).Value)) abs1 = CDBl(Math.Abs(xlsSheet.C ells(4, 18).Value)) abs2 = CDBl(Math.Abs(xlsSheet.C ells(4, 21).Value)) abs3 = CDBl(Math.Abs(xlsSheet.C ells(4, 24).Value)) If xlsSheet.Cells(4, 15).Value < 0 Then con = " - " Else con = " + " End If If xlsSheet.Cells(4, 18).Value < 0 Then con1 = " - " Else con1 = " + " End If If xlsSheet.Cells(4, 21).Value < 0 Then con2 = " - " Else con2 = " + " End If If xlsSheet.Cells(4, 24).Value < 0 Then con3 = " - " Else con3 = " + " End If show_name_x() If (xlsSheet.Cells(7, 28).Value.ToString = "OK") And (xlsSheet.Cells(9, 28).Value.ToString = "OK") And (xlsSheet.Cells(11, 28).Value.ToString = "OK") And (xlsSheet.Cells(13, </pre>	<pre> 28).Value.ToString = "OK") Then DataGridViewX1.Rows.Ad d() DataGridViewX1.Rows(3). Cells(0).Value = "PI = " & Format(CDBl(xlsSheet.Cells (4, 13).Value.ToString()), "###0.000000") & con & _ Format(CDBl(abs), "###0.000000") & " (" & name_x1 & ") " & con1 & _ Format(CDBl(abs1), "###0.000000") & " (" & name_x2 & ") " & con2 & _ Format(CDBl(abs2), "###0.000000") & " (" & name_x3 & ") " & con3 & _ Format(CDBl(abs3), "###0.000000") & " (" & name_x4 & ") " 'y DataGridViewX1.Rows(3). Cells(1).Value = Format(CDBl(xlsSheet.Cells (4, 27).Value.ToString()), "###0.000000") 'Fc DataGridViewX1.Rows(3). Cells(2).Value = Format(CDBl(xlsSheet.Cells (4, 29).Value.ToString()), "###0.000000") 'R2 DataGridViewX1.Rows(3). Cells(3).Value = Format(CDBl(xlsSheet.Cells (4, 30).Value.ToString()), "###0.000000") 'Strd count_x = 4 Column_ex = 7 vrb_y() vrb_x() vrbx5() Else </pre>
--	--	--

<pre> If xlsSheet.Cells(7, 28).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx4() ElseIf xlsSheet.Cells(9, 28).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx4() Else Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try End Try End Try End Try End Try </pre>	<pre> show_x() vrb_y() vrb_x() vrbx4() ElseIf xlsSheet.Cells(11, 28).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx4() Else Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try End Try </pre>	<pre> End Try show_x() vrb_y() vrb_x() vrbx4() End If End If End Sub Protected Sub vrbx5() vrbx = vrbx + 1 If vrbx <> max_r Then xlsSheet = xlsBook.Worksheets("5") Dim abs1, abs, abs2, abs3, abs4, con1, con, con2, con3, con4 As String abs = CDBl(Math.Abs(xlsSheet.C ells(4, 17).Value)) abs1 = CDBl(Math.Abs(xlsSheet.C ells(4, 20).Value)) abs2 = CDBl(Math.Abs(xlsSheet.C ells(4, 23).Value)) abs3 = CDBl(Math.Abs(xlsSheet.C ells(4, 26).Value)) abs4 = CDBl(Math.Abs(xlsSheet.C ells(4, 29).Value)) If xlsSheet.Cells(4, 17).Value < 0 Then con = " - " Else con = " + " End If If xlsSheet.Cells(4, 20).Value < 0 Then con1 = " - " Else con1 = " + " End If If xlsSheet.Cells(4, 23).Value < 0 Then con2 = " - " Else con2 = " + " End If End If </pre>
--	--	--

<pre> If xlsSheet.Cells(4, 26).Value < 0 Then con3 = " - " Else con3 = " + " End If If xlsSheet.Cells(4, 29).Value < 0 Then con4 = " - " Else con4 = " + " End If 'แสดงชื่อ X show_name_x() If (xlsSheet.Cells(7, 33).Value.ToString = "OK") And (xlsSheet.Cells(9, 33).Value.ToString = "OK") And (xlsSheet.Cells(11, 33).Value.ToString = "OK") And (xlsSheet.Cells(13, 33).Value.ToString = "OK") - And (xlsSheet.Cells(15, 33).Value.ToString = "OK") Then DataGridViewX1.Rows.Ad d() DataGridViewX1.Rows(4). Cells(0).Value = "PI = " & Format(CDbl(xlsSheet.Cells (4, 15).Value.ToString()), "#,##0.000000") & con & _ Format(CDbl(abs), "#,##0.000000") & " (" & name_x1 & ") " & con1 & - Format(CDbl(abs1), "#,##0.000000") & " (" & name_x2 & ") " & con2 & - Format(CDbl(abs2), "#,##0.000000") & " (" & name_x3 & ") " & con3 & - Format(CDbl(abs3), </pre>	<pre> "#,##0.000000") & " (" & name_x4 & ") " & con4 & - Format(CDbl(abs4), "#,##0.000000") & " (" & name_x5 & ") " & 'y DataGridViewX1.Rows(4). Cells(1).Value = Format(CDbl(xlsSheet.Cells (4, 32).Value.ToString()), "#,##0.000000") 'Fc DataGridViewX1.Rows(4). Cells(2).Value = Format(CDbl(xlsSheet.Cells (4, 34).Value.ToString()), "#,##0.000000") 'R2 DataGridViewX1.Rows(4). Cells(3).Value = Format(CDbl(xlsSheet.Cells (4, 35).Value.ToString()), "#,##0.000000") 'Strd count_x = 5 Column_ex = 8 vrb_y() vrb_x() vrbx6() Else If xlsSheet.Cells(7, 33).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = "" & r_no & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try </pre>	<pre> End Try show_x() vrb_y() vrb_x() vrbx5() ElseIf xlsSheet.Cells(9, 33).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = "" & r_no & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx5() ElseIf xlsSheet.Cells(11, 33).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = "" & r_no & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 </pre>
--	--	--

Exception	<pre> Catch ex As End Try show_x() vrb_y() vrb_x() vrbx5() ElseIf xlsSheet.Cells(13, 33).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx5() Else Try sql = "update VRB_r set status='N' where r_no = " & r_no & " " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With </pre>	<pre> r1 = r1 + 1 Catch ex As Exception End Try show_x() vrb_y() vrb_x() vrbx5() End If End If End If End Sub Protected Sub vrbx6() vrbx = vrbx + 1 If vrbx <> max_r Then xlsSheet = xlsBook.Worksheets("6") Dim abs1, abs, abs2, abs3, abs4, abs5, con1, con, con2, con3, con4, con5 As String abs = CDBl(Math.Abs(xlsSheet.C ells(4, 17).Value)) abs1 = CDBl(Math.Abs(xlsSheet.C ells(4, 20).Value)) abs2 = CDBl(Math.Abs(xlsSheet.C ells(4, 23).Value)) abs3 = CDBl(Math.Abs(xlsSheet.C ells(4, 26).Value)) abs4 = CDBl(Math.Abs(xlsSheet.C ells(4, 29).Value)) abs5 = CDBl(Math.Abs(xlsSheet.C ells(4, 32).Value)) If xlsSheet.Cells(4, 17).Value < 0 Then con = " - " Else con = " + " End If If xlsSheet.Cells(4, 20).Value < 0 Then con1 = " - " Else </pre>	<pre> con1 = " + " End If If xlsSheet.Cells(4, 23).Value < 0 Then con2 = " - " Else con2 = " + " End If If xlsSheet.Cells(4, 26).Value < 0 Then con3 = " - " Else con3 = " + " End If If xlsSheet.Cells(4, 29).Value < 0 Then con4 = " - " Else con4 = " + " End If If xlsSheet.Cells(4, 32).Value < 0 Then con5 = " - " Else con5 = " + " End If 'แสดงชื่อ X show_name_x() If (xlsSheet.Cells(7, 36).Value.ToString = "OK") And (xlsSheet.Cells(9, 36).Value.ToString = "OK") And (xlsSheet.Cells(11, 36).Value.ToString = "OK") And (xlsSheet.Cells(13, 36).Value.ToString = "OK") - And (xlsSheet.Cells(15, 36).Value.ToString = "OK") And (xlsSheet.Cells(17, 36).Value.ToString = "OK") Then DataGridViewX1.Rows.Ad d() DataGridViewX1.Rows(5). Cells(0).Value = "PI = " & Format(CDBl(xlsSheet.Cells </pre>
-----------	--	---	---

```

(4, 15).Value.ToString()),
"#,##0.000000") & con & _
Format(CDbl(abs),
"#,##0.000000") & " ( " &
DataGridView8.Rows(0).Cells(0).Value.ToString & " )
" & con1 & _
Format(CDbl(abs1),
"#,##0.000000") & " ( " &
DataGridView8.Rows(1).Cells(0).Value.ToString & " )
" & con2 & _
Format(CDbl(abs2),
"#,##0.000000") & " ( " &
DataGridView8.Rows(2).Cells(0).Value.ToString & " )
" & con3 & _
Format(CDbl(abs3),
"#,##0.000000") & " ( " &
DataGridView8.Rows(3).Cells(0).Value.ToString & " )
" & con4 & _
Format(CDbl(abs4),
"#,##0.000000") & " ( " &
DataGridView8.Rows(4).Cells(0).Value.ToString & " )
" & con5 & _
Format(CDbl(abs5),
"#,##0.000000") & " ( " &
DataGridView8.Rows(5).Cells(0).Value.ToString & " )
" 'y'y
DataGridViewX1.Rows(5).Cells(1).Value =
Format(CDbl(xlsSheet.Cells(4, 35).Value.ToString()),
"#,##0.000000") 'Fc
DataGridViewX1.Rows(5).Cells(2).Value =
Format(CDbl(xlsSheet.Cells(4, 37).Value.ToString()),
"#,##0.000000") 'R2
DataGridViewX1.Rows(5).Cells(3).Value =
Format(CDbl(xlsSheet.Cells(4, 38).Value.ToString()),
"#,##0.000000") 'Strd
count_x = 6
Column_ex = 9
vrb_y()
vrb_x()
vrbx7()
Else
If
xlsSheet.Cells(7, 33).Value.ToString = "NO"
Then
Try
sql = "update
VRB_r set status='N' where
r_no = " &
DataGridView4.Rows(r1).Cells(30).Value.ToString & "
"
With com
.Connection = conn1
.CommandType =
CommandType.Text
.CommandText = sql
.ExecuteNonQuery()
End With
r1 = r1 + 1
Catch ex As
Exception
End Try
count_x = 6
show_x()
vrb_y()
vrb_x()
vrbx6()
ElseIf
xlsSheet.Cells(11, 36).Value.ToString = "NO"
Then
Try
sql = "update
VRB_r set status='N' where
r_no = " &
DataGridView4.Rows(r1).Cells(30).Value.ToString & "
"
With com
.Connection = conn1
.CommandType =
CommandType.Text
.CommandText = sql
.ExecuteNonQuery()
End With
r1 = r1 + 1
Catch ex As
Exception
End Try
count_x = 6
show_x()
vrb_y()
vrb_x()
vrbx6()
ElseIf
xlsSheet.Cells(13, 36).Value.ToString = "NO"
Then

```

<pre> Try sql = "update VRB_r set status='N' where r_no = '" & DataGridView4.Rows(r1).C ells(30).Value.ToString & '" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 6 show_x() vrb_y() vrb_x() vrbx6() Elseif xlsSheet.Cells(15, 36).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = '" & DataGridView4.Rows(r1).C ells(30).Value.ToString & '" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 6 </pre>	<pre> show_x() vrb_y() vrb_x() vrbx6() Else Try sql = "update VRB_r set status='N' where r_no = '" & DataGridView4.Rows(r1).C ells(30).Value.ToString & '" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 6 show_x() vrb_y() vrb_x() vrbx6() End If End If End Sub Protected Sub vrbx7() vrbx = vrbx + 1 If vrbx <> max_r Then xlsSheet = xlsBook.Worksheets("7") Dim abs1, abs, abs2, abs3, abs4, abs5, abs6, con1, con, con2, con3, con4, con5, con6 As String abs = Cdbl(Math.Abs(xlsSheet.C ells(4, 18).Value)) abs1 = Cdbl(Math.Abs(xlsSheet.C ells(4, 21).Value)) </pre>	<pre> abs2 = Cdbl(Math.Abs(xlsSheet.C ells(4, 24).Value)) abs3 = Cdbl(Math.Abs(xlsSheet.C ells(4, 27).Value)) abs4 = Cdbl(Math.Abs(xlsSheet.C ells(4, 30).Value)) abs5 = Cdbl(Math.Abs(xlsSheet.C ells(4, 33).Value)) abs6 = Cdbl(Math.Abs(xlsSheet.C ells(4, 36).Value)) If xlsSheet.Cells(4, 18).Value < 0 Then con = " - " Else con = " + " End If If xlsSheet.Cells(4, 21).Value < 0 Then con1 = " - " Else con1 = " + " End If If xlsSheet.Cells(4, 24).Value < 0 Then con2 = " - " Else con2 = " + " End If If xlsSheet.Cells(4, 27).Value < 0 Then con3 = " - " Else con3 = " + " End If If xlsSheet.Cells(4, 30).Value < 0 Then con4 = " - " Else con4 = " + " End If If xlsSheet.Cells(4, 33).Value < 0 Then con5 = " - " Else con5 = " + " End If </pre>
--	---	--

<pre> If xlsSheet.Cells(4, 36).Value < 0 Then con6 = " - " Else con6 = " + " End If show_name_x() If (xlsSheet.Cells(7, 38).Value.ToString = "OK") And (xlsSheet.Cells(9, 38).Value.ToString = "OK") And (xlsSheet.Cells(11, 38).Value.ToString = "OK") And (xlsSheet.Cells(13, 38).Value.ToString = "OK") - And (xlsSheet.Cells(15, 38).Value.ToString = "OK") And (xlsSheet.Cells(17, 38).Value.ToString = "OK") And (xlsSheet.Cells(19, 38).Value.ToString = "OK") Then DataGridViewX1.Rows.Add() DataGridViewX1.Rows(6). Cells(0).Value = "PI = " & Format(CDbl(xlsSheet.Cells (4, 16).Value.ToString()), "#,##0.000000") & con & _ Format(CDbl(abs), "#,##0.000000") & " (" & DataGridView8.Rows(0).Ce lls(0).Value.ToString & ") " & con1 & _ Format(CDbl(abs1), "#,##0.000000") & " (" & DataGridView8.Rows(1).Ce lls(0).Value.ToString & ") " & con2 & _ Format(CDbl(abs2), "#,##0.000000") & " (" & DataGridView8.Rows(2).Ce lls(0).Value.ToString & ") " & con3 & _ Format(CDbl(abs3), "#,##0.000000") & " (" & </pre>	<pre> DataGridView8.Rows(3).Ce lls(0).Value.ToString & ") " & con4 & _ Format(CDbl(abs4), "#,##0.000000") & " (" & DataGridView8.Rows(4).Ce lls(0).Value.ToString & ") " & con5 & _ Format(CDbl(abs5), "#,##0.000000") & " (" & DataGridView8.Rows(5).Ce lls(0).Value.ToString & ") " & con6 & _ Format(CDbl(abs6), "#,##0.000000") & " (" & DataGridView8.Rows(6).Ce lls(0).Value.ToString & ") " 'y'y DataGridViewX1.Rows(6). Cells(1).Value = Format(CDbl(xlsSheet.Cells (4, 39).Value.ToString()), "#,##0.000000") 'Fc DataGridViewX1.Rows(6). Cells(2).Value = Format(CDbl(xlsSheet.Cells (4, 41).Value.ToString()), "#,##0.000000") 'R2 DataGridViewX1.Rows(6). Cells(3).Value = Format(CDbl(xlsSheet.Cells (4, 42).Value.ToString()), "#,##0.000000") 'Strd count_x = 7 Column_ex = 10 vrb_y() vrb_x() vrbx8() Else If xlsSheet.Cells(7, 38).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & DataGridView4.Rows(r1).C </pre>	<pre> ells(30).Value.ToString & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 7 show_x() vrb_y() vrb_x() vrbx7() ElseIf xlsSheet.Cells(9, 38).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = " & DataGridView4.Rows(r1).C ells(30).Value.ToString & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 7 show_x() vrb_y() vrb_x() vrbx7() </pre>
--	--	---

End If	con2 = " + "	44).Value.ToString = "OK")
End If	End If	Then
End Sub		
Protected Sub vrbx8()	If xlsSheet.Cells(4,	DataGridViewX1.Rows.Ad
	28).Value < 0 Then	d()
	con3 = " - "	
vrbx = vrbx + 1	Else	DataGridViewX1.Rows(7).
If vrbx <> max_r Then	con3 = " + "	Cells(0).Value = "PI = " &
xlsSheet =	End If	Format(CDbl(xlsSheet.Cells
xlsBook.Worksheets("8")		(4, 17).Value.ToString()),
	If xlsSheet.Cells(4,	"#,##0.000000") & con & _
Dim abs1, abs, abs2,	31).Value < 0 Then	
abs3, abs4, abs5, abs6, abs7,	con4 = " - "	Format(CDbl(abs),
con1, con, con2, con3, con4,	Else	"#,##0.000000") & " (" &
con5, con6, con7 As String	con4 = " + "	DataGridView8.Rows(0).Ce
abs =	End If	lls(0).Value.ToString & ")
CDbl(Math.Abs(xlsSheet.C		" & con1 & _
ells(4, 19).Value))	If xlsSheet.Cells(4,	Format(CDbl(abs1),
abs1 =	34).Value < 0 Then	"#,##0.000000") & " (" &
CDbl(Math.Abs(xlsSheet.C	con5 = " - "	DataGridView8.Rows(1).Ce
ells(4, 22).Value))	Else	lls(0).Value.ToString & ")
abs2 =	con5 = " + "	" & con2 & _
CDbl(Math.Abs(xlsSheet.C	End If	
ells(4, 25).Value))	If xlsSheet.Cells(4,	Format(CDbl(abs2),
abs3 =	37).Value < 0 Then	"#,##0.000000") & " (" &
CDbl(Math.Abs(xlsSheet.C	con6 = " - "	DataGridView8.Rows(2).Ce
ells(4, 28).Value))	Else	lls(0).Value.ToString & ")
abs4 =	con6 = " + "	" & con3 & _
CDbl(Math.Abs(xlsSheet.C	End If	
ells(4, 31).Value))	If xlsSheet.Cells(4,	Format(CDbl(abs3),
abs5 =	40).Value < 0 Then	"#,##0.000000") & " (" &
CDbl(Math.Abs(xlsSheet.C	con7 = " - "	DataGridView8.Rows(3).Ce
ells(4, 34).Value))	Else	lls(0).Value.ToString & ")
abs6 =	con7 = " + "	" & con4 & _
CDbl(Math.Abs(xlsSheet.C	End If	
ells(4, 37).Value))	show_name_x()	Format(CDbl(abs4),
abs7 =		"#,##0.000000") & " (" &
CDbl(Math.Abs(xlsSheet.C	If (xlsSheet.Cells(7,	DataGridView8.Rows(4).Ce
ells(4, 40).Value))	44).Value.ToString = "OK")	lls(0).Value.ToString & ")
	And (xlsSheet.Cells(9,	" & con5 & _
If xlsSheet.Cells(4,	44).Value.ToString = "OK")	
19).Value < 0 Then	And (xlsSheet.Cells(11,	Format(CDbl(abs5),
con = " - "	44).Value.ToString = "OK")	"#,##0.000000") & " (" &
Else	And (xlsSheet.Cells(13,	DataGridView8.Rows(5).Ce
con = " + "	44).Value.ToString = "OK")	lls(0).Value.ToString & ")
End If		" & con6 & _
	And	
If xlsSheet.Cells(4,	(xlsSheet.Cells(15,	Format(CDbl(abs6),
22).Value < 0 Then	44).Value.ToString = "OK")	"#,##0.000000") & " (" &
con1 = " - "	And (xlsSheet.Cells(17,	DataGridView8.Rows(6).Ce
Else	44).Value.ToString = "OK")	lls(0).Value.ToString & ")
con1 = " + "		" & con7 & _
End If		
	If xlsSheet.Cells(4,	Format(CDbl(abs7),
If xlsSheet.Cells(4,	25).Value < 0 Then	"#,##0.000000") & " (" &
con2 = " - "	con2 = " - "	DataGridView8.Rows(7).Ce
Else	Else	lls(0).Value.ToString & ")
	And (xlsSheet.Cells(21,	

r_no = "" & DataGridView4.Rows(r1).Cells(30).Value.ToString & "" "	vrb_x() vrbx8()	Catch ex As Exception
With com	ElseIf xlsSheet.Cells(19, 44).Value.ToString = "NO" Then	End Try count_x = 8 show_x() vrb_y() vrb_x() vrbx8()
.Connection = conn1	Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).Cells(30).Value.ToString & "" "	End If End If End Sub Protected Sub vrbx9()
.CommandType = CommandType.Text	With com	vrbx = vrbx + 1 If vrbx <> max_r Then xlsSheet = xlsBook.Worksheets("9")
.CommandText = sql	.Connection = conn1	Dim abs1, abs, abs2, abs3, abs4, abs5, abs6, abs7, abs8, con1, con, con2, con3, con4, con5, con6, con7, con8 As String abs = CDBl(Math.Abs(xlsSheet.Cells(4, 20).Value)) abs1 = CDBl(Math.Abs(xlsSheet.Cells(4, 23).Value)) abs2 = CDBl(Math.Abs(xlsSheet.Cells(4, 26).Value)) abs3 = CDBl(Math.Abs(xlsSheet.Cells(4, 29).Value)) abs4 = CDBl(Math.Abs(xlsSheet.Cells(4, 32).Value)) abs5 = CDBl(Math.Abs(xlsSheet.Cells(4, 35).Value)) abs6 = CDBl(Math.Abs(xlsSheet.Cells(4, 38).Value)) abs7 = CDBl(Math.Abs(xlsSheet.Cells(4, 41).Value)) abs8 = CDBl(Math.Abs(xlsSheet.Cells(4, 44).Value))
.ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception	.CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception	
End Try count_x = 8 show_x() vrb_y() vrb_x() vrbx8()	End Try count_x = 8 show_x() vrb_y() vrb_x() vrbx8() Else Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).Cells(30).Value.ToString & "" "	
ElseIf xlsSheet.Cells(17, 44).Value.ToString = "NO" Then	Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).Cells(30).Value.ToString & "" "	
Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).Cells(30).Value.ToString & "" "	With com	
With com	Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).Cells(30).Value.ToString & "" "	
.Connection = conn1	With com	
.CommandType = CommandType.Text	.Connection = conn1	
.CommandText = sql	.CommandType = CommandType.Text	
.ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception	.CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1	
End Try count_x = 8 show_x() vrb_y()		If xlsSheet.Cells(4, 20).Value < 0 Then con = " - "

Else con = " + " End If	con8 = " + " End If	"#,##0.000000") & " (" & DataGridView8.Rows(3).Ce lls(0).Value.ToString & ") " & con4 & _
If xlsSheet.Cells(4, 23).Value < 0 Then con1 = " - " Else con1 = " + " End If	show_name_x() If (xlsSheet.Cells(7, 48).Value.ToString = "OK") And (xlsSheet.Cells(9, 48).Value.ToString = "OK") And (xlsSheet.Cells(11, 48).Value.ToString = "OK") And (xlsSheet.Cells(13, 48).Value.ToString = "OK")	Format(CDbl(abs4), "#,##0.000000") & " (" & DataGridView8.Rows(4).Ce lls(0).Value.ToString & ") " & con5 & _
If xlsSheet.Cells(4, 26).Value < 0 Then con2 = " - " Else con2 = " + " End If	- And (xlsSheet.Cells(15, 48).Value.ToString = "OK") And (xlsSheet.Cells(17, 48).Value.ToString = "OK") And (xlsSheet.Cells(19, 48).Value.ToString = "OK")	Format(CDbl(abs5), "#,##0.000000") & " (" & DataGridView8.Rows(5).Ce lls(0).Value.ToString & ") " & con6 & _
If xlsSheet.Cells(4, 29).Value < 0 Then con3 = " - " Else con3 = " + " End If	- And (xlsSheet.Cells(21, 48).Value.ToString = "OK") And (xlsSheet.Cells(23, 48).Value.ToString = "OK")	Format(CDbl(abs6), "#,##0.000000") & " (" & DataGridView8.Rows(6).Ce lls(0).Value.ToString & ") " & con7 & _
If xlsSheet.Cells(4, 32).Value < 0 Then con4 = " - " Else con4 = " + " End If	Then DataGridViewX1.Rows.Ad d()	Format(CDbl(abs7), "#,##0.000000") & " (" & DataGridView8.Rows(7).Ce lls(0).Value.ToString & ") " & con8 & _
If xlsSheet.Cells(4, 35).Value < 0 Then con5 = " - " Else con5 = " + " End If	DataGridViewX1.Rows(8). Cells(0).Value = "PI = " & Format(CDbl(xlsSheet.Cells (4, 18).Value.ToString()), "#,##0.000000") & con & _	Format(CDbl(abs8), "#,##0.000000") & " (" & DataGridView8.Rows(8).Ce lls(0).Value.ToString & ") "" 'y'y"
If xlsSheet.Cells(4, 38).Value < 0 Then con6 = " - " Else con6 = " + " End If	Format(CDbl(abs), "#,##0.000000") & " (" & DataGridView8.Rows(0).Ce lls(0).Value.ToString & ") " & con1 & _	DataGridViewX1.Rows(8). Cells(1).Value = Format(CDbl(xlsSheet.Cells (4, 47).Value.ToString()), "#,##0.000000") 'Fc
If xlsSheet.Cells(4, 41).Value < 0 Then con7 = " - " Else con7 = " + " End If	Format(CDbl(abs1), "#,##0.000000") & " (" & DataGridView8.Rows(1).Ce lls(0).Value.ToString & ") " & con2 & _	DataGridViewX1.Rows(8). Cells(2).Value = Format(CDbl(xlsSheet.Cells (4, 49).Value.ToString()), "#,##0.000000") 'R2
If xlsSheet.Cells(4, 44).Value < 0 Then con8 = " - " Else	Format(CDbl(abs2), "#,##0.000000") & " (" & DataGridView8.Rows(2).Ce lls(0).Value.ToString & ") " & con3 & _ Format(CDbl(abs3),	DataGridViewX1.Rows(8). Cells(3).Value = Format(CDbl(xlsSheet.Cells (4, 50).Value.ToString()), "#,##0.000000") 'Strd
		count_x = 9 Column_ex = 12 vrb_y() vrb_x()

<pre> show_x() vrb_y() vrb_x() vrbx9() ElseIf xlsSheet.Cells(17, 48).Value.ToString = "NO" Then 'update status = N Try sql = "update VRB_r set status='N' where r_no = '" & DataGridView4.Rows(r1).C ells(30).Value.ToString & '" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 9 show_x() vrb_y() vrb_x() vrbx9() ElseIf xlsSheet.Cells(19, 48).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = '" & DataGridView4.Rows(r1).C ells(30).Value.ToString & '" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 9 show_x() vrb_y() vrb_x() vrbx9() </pre>	<pre> .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 9 show_x() vrb_y() vrb_x() vrbx9() ElseIf xlsSheet.Cells(21, 48).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = '" & DataGridView4.Rows(r1).C ells(30).Value.ToString & '" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 9 show_x() vrb_y() vrb_x() vrbx9() </pre>	<pre> DataGridView4.Rows(r1).C ells(30).Value.ToString & '" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 9 show_x() vrb_y() vrb_x() vrbx9() End Try End If End If End Sub Protected Sub vrbx10() vrbx = vrbx + 1 If vrbx <> max_r Then xlsSheet = xlsBook.Worksheets("10") Dim abs1, abs, abs2, abs3, abs4, abs5, abs6, abs7, abs8, abs9, con1, con, con2, con3, con4, con5, con6, con7, con8, con9 As String abs = CDBl(Math.Abs(xlsSheet.C ells(4, 21).Value)) abs1 = CDBl(Math.Abs(xlsSheet.C ells(4, 24).Value)) abs2 = CDBl(Math.Abs(xlsSheet.C ells(4, 27).Value)) abs3 = CDBl(Math.Abs(xlsSheet.C ells(4, 30).Value)) abs4 = CDBl(Math.Abs(xlsSheet.C ells(4, 33).Value)) </pre>
--	--	---

abs5 = CDBl(Math.Abs(xlsSheet.Cells(4, 36).Value))	If xlsSheet.Cells(4, 39).Value < 0 Then con6 = " - " Else con6 = " + " End If	DataGridViewX1.Rows.Add()
abs6 = CDBl(Math.Abs(xlsSheet.Cells(4, 39).Value))		DataGridViewX1.Rows(9).Cells(0).Value = "PI = " & Format(CDBl(xlsSheet.Cells(4, 19).Value.ToString()), "#,##0.000000") & con & _
abs7 = CDBl(Math.Abs(xlsSheet.Cells(4, 42).Value))	If xlsSheet.Cells(4, 42).Value < 0 Then con7 = " - " Else con7 = " + " End If	Format(CDBl(abs), "#,##0.000000") & " (" & DataGridView8.Rows(0).Cells(0).Value.ToString & ")" & con1 & _
abs8 = CDBl(Math.Abs(xlsSheet.Cells(4, 45).Value))		Format(CDBl(abs1), "#,##0.000000") & " (" & DataGridView8.Rows(1).Cells(0).Value.ToString & ")" & con2 & _
abs9 = CDBl(Math.Abs(xlsSheet.Cells(4, 48).Value))	If xlsSheet.Cells(4, 45).Value < 0 Then con8 = " - " Else con8 = " + " End If	Format(CDBl(abs2), "#,##0.000000") & " (" & DataGridView8.Rows(2).Cells(0).Value.ToString & ")" & con3 & _
If xlsSheet.Cells(4, 21).Value < 0 Then con = " - " Else con = " + " End If	If xlsSheet.Cells(4, 48).Value < 0 Then con9 = " - " Else con9 = " + " End If	Format(CDBl(abs3), "#,##0.000000") & " (" & DataGridView8.Rows(3).Cells(0).Value.ToString & ")" & con4 & _
If xlsSheet.Cells(4, 24).Value < 0 Then con1 = " - " Else con1 = " + " End If	show_name_x() If (xlsSheet.Cells(7, 52).Value.ToString = "OK") And (xlsSheet.Cells(9, 52).Value.ToString = "OK") And (xlsSheet.Cells(11, 52).Value.ToString = "OK") And (xlsSheet.Cells(13, 52).Value.ToString = "OK")	Format(CDBl(abs4), "#,##0.000000") & " (" & DataGridView8.Rows(4).Cells(0).Value.ToString & ")" & con5 & _
If xlsSheet.Cells(4, 27).Value < 0 Then con2 = " - " Else con2 = " + " End If	And (xlsSheet.Cells(15, 52).Value.ToString = "OK") And (xlsSheet.Cells(17, 52).Value.ToString = "OK") And (xlsSheet.Cells(19, 52).Value.ToString = "OK")	Format(CDBl(abs5), "#,##0.000000") & " (" & DataGridView8.Rows(5).Cells(0).Value.ToString & ")" & con6 & _
If xlsSheet.Cells(4, 30).Value < 0 Then con3 = " - " Else con3 = " + " End If	And (xlsSheet.Cells(21, 52).Value.ToString = "OK") And (xlsSheet.Cells(23, 52).Value.ToString = "OK") And (xlsSheet.Cells(25, 52).Value.ToString = "OK")	Format(CDBl(abs6), "#,##0.000000") & " (" & DataGridView8.Rows(6).Cells(0).Value.ToString & ")" & con7 & _
If xlsSheet.Cells(4, 33).Value < 0 Then con4 = " - " Else con4 = " + " End If	Then	Format(CDBl(abs7), "#,##0.000000") & " (" & DataGridView8.Rows(7).Ce
If xlsSheet.Cells(4, 36).Value < 0 Then con5 = " - " Else con5 = " + " End If		

<pre> lls(0).Value.ToString & ") " & con8 & _ Format(CDbl(abs8), "#,##0.000000") & " (" & DataGridView8.Rows(8).Ce lls(0).Value.ToString & ") " & con9 & _ Format(CDbl(abs9), "#,##0.000000") & " (" & DataGridView8.Rows(9).Ce lls(0).Value.ToString & ") " 'y'y" DataGridViewX1.Rows(9). Cells(1).Value = Format(CDbl(xlsSheet.Cells (4, 51).Value.ToString()), "#,##0.000000") 'Fc DataGridViewX1.Rows(9). Cells(2).Value = Format(CDbl(xlsSheet.Cells (4, 53).Value.ToString()), "#,##0.000000") 'R2 DataGridViewX1.Rows(9). Cells(3).Value = Format(CDbl(xlsSheet.Cells (4, 54).Value.ToString()), "#,##0.000000") 'Strd count_x = 10 Column_ex = 13 vrb_y() vrb_x() Else If xlsSheet.Cells(7, 52).Value.ToString = "NO" Then 'update status = N Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).C ells(30).Value.ToString & "" " With com .Connection = conn1 .CommandText = sql .ExecuteNonQuery() End Try count_x = 10 show_x() vrb_y() vrb_x() vrbx10() ElseIf xlsSheet.Cells(11, 52).Value.ToString = "NO" Then </pre>	<pre> .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 10 show_x() vrb_y() vrb_x() vrbx10() ElseIf xlsSheet.Cells(9, 52).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).C ells(30).Value.ToString & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception End Try count_x = 10 show_x() vrb_y() vrb_x() vrbx10() ElseIf xlsSheet.Cells(13, 52).Value.ToString = "NO" Then Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).C ells(30).Value.ToString & "" " With com .Connection = conn1 .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception </pre>	<pre> 'update status = N Try sql = "update VRB_r set status='N' where r_no = "" & DataGridView4.Rows(r1).C ells(30).Value.ToString & "" " With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With r1 = r1 + 1 Catch ex As Exception </pre>
---	--	--


```

        TextBoxX5.Text = ""
        GroupBox3.Visible =
False
        ButtonX16.Visible =
False
        End Sub

        Private Sub
PictureBox9_Click(ByVal
sender As System.Object,
ByVal e As
System.EventArgs) Handles
PictureBox9.Click
        'update x
        PictureBox9.Focus()
        max_s()

        For i = 0 To
DataGridViewX5.Rows.Co
unt - 2

            Dim sqlx1 As String
= " select max(id) as n from
VRB_X where id_vrb="" &
max_id_name & """"
            Try
                With com
                    .CommandType
= CommandType.Text
                    .CommandText
= sqlx1
                    .Connection =
conn1
                    dr =
.ExecuteReader()
                    dr.Read()
                    id_x =
(CInt(dr.Item("n")) +
1).ToString("00")

                    End With
                Catch
                    id_x = "01"
                End Try
                dr.Close()

                sql = "insert into
VRB_X (
id_vrb,vrb_name,vrb_y,vrb
_x,vrb_unit,id)values (@id_
vrb,@vrb_name,@vrb_y,@
vrb_x,@vrb_unit,@id)"

                With com
                    .Connection =
conn1
                    .CommandType =
CommandType.Text
                    .CommandText =
sql
                    .Parameters.Clear()
                    .Parameters.Add("@id_vrb"
,
SqlDbType.VarChar).Value
= max_id_name
                    .Parameters.Add("@vrb_na
me",
SqlDbType.VarChar).Value
= TextBox1.Text
                    .Parameters.Add("@vrb_y",
SqlDbType.VarChar).Value
= TextBox2.Text

                    Try
                        If
String.IsNullOrEmpty(Trim
(CStr(DataGridViewX5.Ro
ws(i).Cells(1).Value))) Then
                            'ถ้าเป็นค่าว่าง
                            .Parameters.Add("@vrb_x",
SqlDbType.VarChar).Value
= System.DBNull.Value
                            Else
                                .Parameters.Add("@vrb_x",
SqlDbType.VarChar).Value
=
DataGridViewX5.Rows(i).
Cells(1).Value.ToString
                                End If
                                Catch ex As
Exception
                                    .Parameters.Add("@vrb_x",
SqlDbType.VarChar).Value
= System.DBNull.Value
                                End Try
                            Try
                                If
String.IsNullOrEmpty(Trim
(CStr(DataGridViewX5.Ro
ws(i).Cells(2).Value))) Then
                                    .Parameters.Add("@vrb_uni
t",
SqlDbType.VarChar).Value
= System.DBNull.Value
                                Else
                                    .Parameters.Add("@vrb_uni
t",
SqlDbType.VarChar).Value
=
DataGridViewX5.Rows(i).
Cells(2).Value.ToString
                                End If
                                Catch ex As
Exception
                                    .Parameters.Add("@vrb_uni
t",
SqlDbType.VarChar).Value
= System.DBNull.Value
                                End Try
                            .Parameters.Add("@id",
SqlDbType.VarChar).Value
= id_x
                            .ExecuteNonQuery()
                            End With
                        Next
                            show_data()
                            DataGridViewX5.Rows.Cle
ar()
                            dss1.Clear()
                            sql = "SELECT
id_no,id_vrb,row_number()
over (order by id_vrb)
as[No.],vrb_x as'Factor'
,vrb_unit
as'Unit',id_vrb,vrb_name,vr
b_y FROM VRB_X
with(nolock) where
id_vrb="" & max_id_name
& "" order by id_no "
                            da.SelectCommand =
New SqlCommand(sql,
conn1)
                            da.Fill(dss1, "Unit")

                            DataGridViewX4.DataSou
rce = dss1.Tables("Unit")

                            DataGridViewX4.Columns(
0).Visible = False

                            DataGridViewX4.Columns(
1).Visible = False
                    End Sub
    
```

DataGridViewX4.Columns(5).Visible = False	.Columns(0).ReadOnly = False	sql = "delete from VRB_VARIABLE where id_vrb= '' & max_id_name & ''"
DataGridViewX4.Columns(6).Visible = False	DataGridViewX5.Rows(i).Cells(0).Value = i + add_x1 Next	With com .Connection = conn1 .CommandType = CommandType.Text
DataGridViewX4.Columns(7).Visible = False	End If End With	.CommandText = sql .ExecuteNonQuery() End With
DataGridViewX4.Columns(2).Width = 50 GroupBox4.Visible = False End Sub Private Sub	End Sub Private Sub ButtonX16_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonX16.Click	sql = "delete from VRB_cal where id_vrb= '' & max_id_name & ''"
DataGridViewX5_RowPostPaint(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewRowPostPaintEventArgs) Handles DataGridViewX5.RowPostPaint	'delete Dim btncode As String btncode = MsgBox("Do you want to delete?", vbYesNo, "") If btncode = vbYes Then Else Exit Sub End If	With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With
max_s() If maxs = "16" Then	MsgBox("Do you want to delete?", vbYesNo, "") If btncode = vbYes Then Else Exit Sub End If	sql = "delete from VRB_r where substring(r_no,1,4)= '' & max_id_name & ''"
DataGridViewX5.AllowUserToAddRows = False End If	Then Else Exit Sub End If	With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With
DataGridViewX5.Rows.Count - 1 Dim ab As Integer ab =	GroupBox4.Visible = False sql = "delete from VRB_X where id_vrb= '' & max_id_name & ''"	show_data() dss1.Clear() sql = "SELECT id_no,id_vrb,row_number() over (order by id_vrb) as[No.],vrb_x as'Factor',vrb_unit as'Unit',id_vrb,vrb_name,vrb_y FROM VRB_X with(nolock) where id_vrb='' & max_id_name & '' order by id_no "
DataGridViewX5.Rows.Count - 1 If	With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With	da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(dss1, "Unit")
DataGridViewX5.Rows(ab).Cells(0).Value = "31" Then	sql = "delete from VRB_x1 where substring(r_no,1,4)= '' & max_id_name & ''"	
DataGridViewX5.AllowUserToAddRows = False End If	With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With	
With DataGridViewX5 If .RowCount > 0 Then For i = 0 To DataGridViewX5.Rows.Count - 1	With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .ExecuteNonQuery() End With	DataGridViewX4.DataSource = dss1.Tables("Unit")

DataGridViewX4.Columns(0).Visible = False	dgv_db1.Visible = False End Sub	gs) Handles LinkLabel3.LinkClicked
DataGridViewX4.Columns(1).Visible = False	Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked PictureBox20.Visible = True	PictureBox22.Visible = True
DataGridViewX4.Columns(5).Visible = False		PictureBox20.Visible = False
DataGridViewX4.Columns(6).Visible = False		PictureBox21.Visible = False
DataGridViewX4.Columns(7).Visible = False		PictureBox23.Visible = False
		PictureBox24.Visible = False
DataGridViewX4.Columns(2).Width = 50		PictureBox3.Visible = False
TextBoxX3.Text = ""	PictureBox21.Visible = False	PictureBox4.Visible = False
TextBoxX4.Text = ""	PictureBox22.Visible = False	End Sub
TextBoxX5.Text = ""	PictureBox23.Visible = False	Private Sub LinkLabel5_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel5.LinkClicked
GroupBox3.Visible = False	PictureBox3.Visible = False	
ButtonX16.Visible = False	PictureBox4.Visible = False	
End Sub	End Sub	
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick	Private Sub LinkLabel2_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel2.LinkClicked	PictureBox23.Visible = True
Timer1.Enabled = False	PictureBox21.Visible = True	PictureBox20.Visible = False
Panel1.Visible = False		PictureBox21.Visible = False
End Sub		PictureBox22.Visible = False
Private Sub txt_name_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt_name.TextChanged	PictureBox20.Visible = False	PictureBox24.Visible = False
dgv_db1.Visible = True	PictureBox22.Visible = False	PictureBox3.Visible = False
End Sub	PictureBox23.Visible = False	PictureBox4.Visible = False
Private Sub RibbonTabItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RibbonTabItem1.Click	PictureBox24.Visible = False	End Sub
	PictureBox3.Visible = False	Private Sub LinkLabel4_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel4.LinkClicked
	PictureBox4.Visible = False	
	End Sub	
	Private Sub LinkLabel3_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs)	PictureBox24.Visible = True

```

        PictureBox20.Visible =
False
        PictureBox21.Visible =
False
        PictureBox22.Visible =
False
        PictureBox23.Visible =
False
        PictureBox3.Visible =
False
        PictureBox4.Visible =
False
    End Sub

    Private Sub
LinkLabel6_LinkClicked(By
yVal sender As
System.Object, ByVal e As
System.Windows.Forms.Lin
kLabelLinkClickedEventAr
gs) Handles
LinkLabel6.LinkClicked

        PictureBox3.Visible =
True
        PictureBox4.Visible =
True

        PictureBox20.Visible =
False
        PictureBox21.Visible =
False
        PictureBox22.Visible =
False
        PictureBox23.Visible =
False
        PictureBox24.Visible =
False

    End Sub

    Private Sub
RibbonTabItem6_Click(By
Val sender As
System.Object, ByVal e As
System.EventArgs)

    End Sub

    Private Sub
Button2_Click(ByVal
sender As System.Object,
ByVal e As
System.EventArgs)
        If data_x.Rows.Count
= 0 Then
            Exit Sub
        End If

        If
        ((data_x.Columns.Count =
0) Or (data_x.Rows.Count =
0)) Then
            Exit Sub
        End If

        Dim dset As New
DataSet
        dset.Tables.Add()
        For i As Integer = 0
To data_x.ColumnCount - 1
            dset.Tables(0).Columns.Ad
d(data_x.Columns(i).Header
Text)
            Next

            Dim dr1 As DataRow
            For i As Integer = 0 To
data_x.RowCount - 1
                dr1 =
dset.Tables(0).NewRow
                For j As Integer = 0
To data_x.Columns.Count -
1
                    dr1(j) =
data_x.Rows(i).Cells(j).Val
ue
                Next

                dset.Tables(0).Rows.Add(dr
1)
            Next

            Dim excel As New
Microsoft.Office.Interop.Ex
cel.ApplicationClass
            Dim wBook As
Microsoft.Office.Interop.Ex
cel.Workbook
            Dim wSheet As
Microsoft.Office.Interop.Ex
cel.Worksheet

            System.Threading.Thread.C
urrentThread.CurrentCultur
e = New
System.Globalization.Cultur
eInfo("en-US")

            wBook =
excel.Workbooks.Add()
            wSheet =
wBook.ActiveSheet()

            Dim dt As
System.Data.DataTable =
dset.Tables(0)
            Dim dc As
System.Data.DataColumn
            Dim dr As
System.Data.DataRow
            Dim colIndex As
Integer = 0
            Dim rowIndex As
Integer = 0

            For Each dc In
dt.Columns
                colIndex = colIndex
+ 1
                excel.Cells(1,
colIndex) =
dc.ColumnName
            Next

            For Each dr In dt.Rows
                rowIndex =
rowIndex + 1
                colIndex = 0
                For Each dc In
dt.Columns
                    colIndex =
colIndex + 1
                    excel.Cells(rowIndex + 1,
colIndex) =
dr(dc.ColumnName)
                Next
            Next

            wSheet.Columns.AutoFit()
            Dim strFileName As
String = "C:\รายงาน
วัคซีน\VAC_" & Now.Day &
Now.Month & Now.Year &
".doc"
            Dim blnFileOpen As
Boolean = False
            Try
                Dim fileTemp As
System.IO.FileStream =
System.IO.File.OpenWrite(s
trFileName)
                fileTemp.Close()
            Catch ex As Exception
                blnFileOpen = False
            End Try
        End Sub
    End Sub

```

<pre> If System.IO.File.Exists(strFile eName) Then System.IO.File.Delete(strFile eName) End If wBook.SaveAs(strFileName) excel.Workbooks.Open(strF ileName) excel.Visible = True System.Threading.Thread.C urrentThread.CurrentCultur e = New System.Globalization.Cultur eInfo("th-TH") End Sub Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Dim objWordApp As Object Dim objWordDoc As Object objWordApp = CreateObject("Word.Applic ation") objWordDoc = objWordApp.documents.ad d() objWordDoc.tables.add(obj WordDoc.range(0, 0), 11, 3) With objWordDoc.tables(1) .borders.InsideLineStyle = 1 .borders.OutsideLineStyle = 1 For i As Integer = 0 To 10 .rows(i + 1).cells(1).range.text = </pre>	<pre> Me.data_x.Rows(i).Cells(" No.").Value .rows(i + 1).cells(2).range.text = Me.data_x.Rows(i).Cells("T itle").Value Next End With objWordApp.visible = True objWordApp = Nothing End Sub Public Sub exportToWord(ByVal dgv As DataGridView) Dim oWord As Word.Application = DirectCast(CreateObject(" Word.Application"), Word.Application) Dim oDoc As Word.Document = oWord.Documents.Add() oWord.Visible = True Const wdAlignParagraphCenter = 1 Dim MyRange1, MyRange2, MyRange3 As Microsoft.Office.Interop.W ord.Range MyRange1 = oDoc.Paragraphs.Add.Rang e With MyRange1 .ParagraphFormat.Alignmen t = wdAlignParagraphCenter .Font.Bold = True .InsertBefore(" Productivity Index Prediction Software" & _ vbCrLf & "Data (View)" & "Name: " & TextBox1.Text & vbCrLf & "Date: " & Date.Today) End With </pre>	<pre> Dim headers = (From ch In dgv.Columns _ Let header = DirectCast(DirectCast(ch, DataGridViewColumn).Hea derCell, DataGridViewColumnHead erCell) _ Select header.Value).ToArray() Dim headerText() As String = Array.ConvertAll(headers, Function(v) v.ToString) Dim items() = (From r In dgv.Rows _ Let row = DirectCast(r, DataGridViewRow) _ Where Not row.IsNewRow _ Select (From cell In row.Cells _ Let c = DirectCast(cell, DataGridViewCell) _ Select c.Value).ToArray()).ToArra y() Dim table As String = String.Join(vbTab, headerText) & Environment.NewLine For Each a In items Dim t() As String = Array.ConvertAll(a, Function(v) v.ToString) table &= String.Join(vbTab, t) & Environment.NewLine Next table = table.TrimEnd(CChar(Envir onment.NewLine)) Clipboard.SetText(table) Dim oTable As Word.Table = oDoc.Tables.Add(oDoc.Boo kmarks.Item("\endofdoc").R ange, items.Count + 1, headers.Count) </pre>
---	--	---

oTable.Range.Paste()	.Color = Word.WdColor.wdColorAutomatic	.Borders(Word.WdBorderType.wdBorderDiagonalDown).LineStyle = Word.WdLineStyle.wdLineStyleNone
oTable.Rows.Item(1).Range .Font.Bold = &H98967E	End With With	
oTable.Rows.Item(1).Range .Font.Size = 14	.Borders(Word.WdBorderType.wdBorderTop) .LineStyle =	.Borders(Word.WdBorderType.wdBorderDiagonalUp).LineStyle = Word.WdLineStyle.wdLineStyleNone
oTable.Rows.Item(1).Range .Font.Color = Word.WdColor.wdColorWhite	Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt	.Borders.Shadow = False
oTable.Rows.Item(1).Range .Shading.Texture = Word.WdTextureIndex.wdTextureNone	.Color = Word.WdColor.wdColorAutomatic End With With	End With MyRange2 = oDoc.Paragraphs.Add.Range
oTable.Rows.Item(1).Range .Shading.ForegroundPatternColor = Word.WdColor.wdColorAutomatic	.Borders(Word.WdBorderType.wdBorderBottom) .LineStyle = Word.WdLineStyle.wdLineStyleSingle	With MyRange2 .ParagraphFormat.Alignment = wdAlignParagraphCenter
oTable.Rows.Item(1).Range .Shading.BackgroundColor = Word.WdColor.wdColorPaleBlue	.LineWidth = Word.WdLineWidth.wdLineeWidth100pt .Color = Word.WdColor.wdColorAutomatic	.Font.Bold = True InsertBefore(vbCrLf & vbCrLf & " By Weerawit@2013")
With	End With With	End With
oTable.Range.Tables(1) With	.Borders(Word.WdBorderType.wdBorderHorizontal) .LineStyle = Word.WdLineStyle.wdLineStyleSingle	End Sub Public Sub exportToWord1(ByVal dgv As DataGridView)
.Borders(Word.WdBorderType.wdBorderLeft) .LineStyle = Word.WdLineStyle.wdLineStyleSingle	.LineWidth = Word.WdLineWidth.wdLineeWidth050pt	Dim oWord As Word.Application = DirectCast(CreateObject("Word.Application"), Word.Application)
Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt .Color = Word.WdColor.wdColorAutomatic	.Color = Word.WdColor.wdColorAutomatic End With With	Dim oDoc As Word.Document = oWord.Documents.Add() oWord.Visible = True
Word.WdColor.wdColorAutomatic	.Borders(Word.WdBorderType.wdBorderVertical) .LineStyle = Word.WdLineStyle.wdLineStyleSingle	Const wdAlignParagraphCenter = 1
End With With	.LineWidth = Word.WdLineWidth.wdLineeWidth050pt .Color = Word.WdColor.wdColorAutomatic	Dim MyRange1, MyRange2, MyRange3 As Microsoft.Office.Interop.Word.Range
.Borders(Word.WdBorderType.wdBorderRight) .LineStyle = Word.WdLineStyle.wdLineStyleSingle	Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt	
Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt	.Color = Word.WdColor.wdColorAutomatic End With	

<pre> MyRange1 = oDoc.Paragraphs.Add.Range With MyRange1 .ParagraphFormat.Alignment = wdAlignParagraphCenter .Font.Bold = True .InsertBefore(" Productivity Index Prediction Software" & _ vbCrLf & "Report average of Software " & "Name: " & TextBox1.Text & vbCrLf & "Date: " & Date.Today) End With Dim headers = (From ch In dgv.Columns _ Let header = DirectCast(DirectCast(ch, DataGridViewColumn).HeaderCell, DataGridViewColumnHeaderCell) _ Select header.Value).ToArray() Dim headerText() As String = Array.ConvertAll(headers, Function(v) v.ToString) Dim items() = (From r In dgv.Rows _ Let row = DirectCast(r, DataGridViewRow) _ Where Not row.IsNewRow _ Select (From cell In row.Cells _ Let c = DirectCast(cell, DataGridViewCell) _ Select c.Value).ToArray()).ToArray() Dim table As String = String.Join(vbTab, headerText) & Environment.NewLine For Each a In items </pre>	<pre> Dim t() As String = Array.ConvertAll(a, Function(v) v.ToString) table &= String.Join(vbTab, t) & Environment.NewLine Next table = table.TrimEnd(CChar(Environment.NewLine)) Clipboard.SetText(table) Dim oTable As Word.Table = oDoc.Tables.Add(oDoc.Bookmarks.Item("endofdoc").Range, items.Count + 1, headers.Count) oTable.Range.Paste() oTable.Rows.Item(1).Range.Font.Bold = &H98967E oTable.Rows.Item(1).Range.Font.Size = 14 oTable.Rows.Item(1).Range.Font.Color = Word.WdColor.wdColorWhite oTable.Rows.Item(1).Range.Shading.Texture = Word.WdTextureIndex.wdTextureNone oTable.Rows.Item(1).Range.Shading.ForegroundPatternColor = Word.WdColor.wdColorAutomatic oTable.Rows.Item(1).Range.Shading.BackgroundColor = Word.WdColor.wdColorPaleBlue With oTable.Range.Tables(1) With .Borders(Word.WdBorderType.wdBorderLeft) </pre>	<pre> .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineWidth100pt .Color = Word.WdColor.wdColorAutomatic End With With .Borders(Word.WdBorderType.wdBorderRight) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineWidth100pt .Color = Word.WdColor.wdColorAutomatic End With With .Borders(Word.WdBorderType.wdBorderTop) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineWidth100pt .Color = Word.WdColor.wdColorAutomatic End With With .Borders(Word.WdBorderType.wdBorderBottom) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineWidth100pt .Color = Word.WdColor.wdColorAutomatic End With With .Borders(Word.WdBorderType.wdBorderHorizontal) .LineStyle = Word.WdLineStyle.wdLineStyleSingle </pre>
---	--	--

<pre> .LineWidth = Word.WdLineWidth.wdLin eWidth050pt .Color = Word.WdColor.wdColorAu tomatic End With With .Borders(Word.WdBorderT ype.wdBorderVertical) .LineStyle = Word.WdLineStyle.wdLin eStyleSingle .LineWidth = Word.WdLineWidth.wdLin eWidth050pt .Color = Word.WdColor.wdColorAu tomatic End With .Borders(Word.WdBorderT ype.wdBorderDiagonalDow n).LineStyle = Word.WdLineStyle.wdLin eStyleNone .Borders(Word.WdBorderT ype.wdBorderDiagonalUp). LineStyle = Word.WdLineStyle.wdLin eStyleNone .Borders.Shadow = False End With MyRange2 = oDoc.Paragraphs.Add.Rang e With MyRange2 .ParagraphFormat.Alignmen t = wdAlignParagraphCenter .Font.Bold = True .InsertBefore(vbCrLf & vbCrLf & " By Weerawit@2013") End With End Sub </pre>	<pre> Public Sub exportToWord2(ByVal dgv As DataGridView) Dim oWord As Word.Application = DirectCast(CreateObject(" Word.Application"), Word.Application) Dim oDoc As Word.Document = oWord.Documents.Add() oWord.Visible = True Const wdAlignParagraphCenter = 1 Dim MyRange1, MyRange2, MyRange3 As Microsoft.Office.Interop.W ord.Range MyRange1 = oDoc.Paragraphs.Add.Rang e With MyRange1 .ParagraphFormat.Alignmen t = wdAlignParagraphCenter .Font.Bold = True .InsertBefore(" Productivity Index Prediction Software" & _ vbCrLf & "Result of Software" & " Name: " & TextBox1.Text & vbCrLf & "Date: " & Date.Today) End With Dim headers = (From ch In dgv.Columns _ Let header = DirectCast(DirectCast(ch, DataGridViewColumn).Hea derCell, DataGridViewColumnHead erCell) _ Select header.Value).ToArray() Dim headerText() As String = Array.ConvertAll(headers, Function(v) v.ToString) </pre>	<pre> Dim items() = (From r In dgv.Rows _ Let row = DirectCast(r, DataGridViewRow) _ Where Not row.IsNewRow _ Select (From cell In row.Cells _ Let c = DirectCast(cell, DataGridViewCell) _ Select c.Value).ToArray()).ToArra y() Dim table As String = String.Join(vbTab, headerText) & Environment.NewLine For Each a In items Dim t() As String = Array.ConvertAll(a, Function(v) v.ToString) table &= String.Join(vbTab, t) & Environment.NewLine Next table = table.TrimEnd(CChar(Envir onment.NewLine)) Clipboard.SetText(table) Dim oTable As Word.Table = oDoc.Tables.Add(oDoc.Boo kmarks.Item("\endofdoc").R ange, items.Count + 1, headers.Count) oTable.Range.Paste() oTable.Rows.Item(1).Range .Font.Bold = &H98967E oTable.Rows.Item(1).Range .Font.Size = 14 oTable.Rows.Item(1).Range .Font.Color = Word.WdColor.wdColorWh ite oTable.Rows.Item(1).Range .Shading.Texture = </pre>
--	--	---

Word.WdTextureIndex.wdTextureNone	End With With	MyRange2 = oDoc.Paragraphs.AddRange With MyRange2
oTable.Rows.Item(1).Range .Shading.ForegroundPattern Color = Word.WdColor.wdColorAutomatic	.Borders(Word.WdBorderType.wdBorderBottom) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt	.ParagraphFormat.Alignment = wdAlignParagraphCenter .Font.Bold = True
oTable.Rows.Item(1).Range .Shading.BackgroundPatternColor = Word.WdColor.wdColorPaleBlue	.Color = Word.WdColor.wdColorAutomatic End With With	.InsertBefore(vbCrLf & vbCrLf & " By Weerawit@2013")
With oTable.Range.Tables(1) With	.Borders(Word.WdBorderType.wdBorderHorizontal) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth050pt .Color = Word.WdColor.wdColorAutomatic	End With End Sub Private Sub Button5_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
.Borders(Word.WdBorderType.wdBorderLeft) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt .Color = Word.WdColor.wdColorAutomatic	Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt .Color = Word.WdColor.wdColorAutomatic End With With	exportToWord(DataGridView1) End Sub
End With With	.Borders(Word.WdBorderType.wdBorderVertical) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth050pt .Color = Word.WdColor.wdColorAutomatic	Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click
.Borders(Word.WdBorderType.wdBorderRight) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt .Color = Word.WdColor.wdColorAutomatic	Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt .Color = Word.WdColor.wdColorAutomatic End With With	ds1.Clear() sql = "delete from RFT " da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(ds1, "RFT")
End With With	.Borders(Word.WdBorderType.wdBorderDiagonalDown).LineStyle = Word.WdLineStyle.wdLineStyleNone	sql = "insert into RFT (x1,x2,x3,x4,x5,x6,x7,x8,x9, x10,x11,x12,x13,x14,x15)values " & _
.Borders(Word.WdBorderType.wdBorderTop) .LineStyle = Word.WdLineStyle.wdLineStyleSingle .LineWidth = Word.WdLineWidth.wdLineeWidth100pt .Color = Word.WdColor.wdColorAutomatic	.Borders(Word.WdBorderType.wdBorderDiagonalUp).LineStyle = Word.WdLineStyle.wdLineStyleNone .Borders.Shadow = False	"(@x1,@x2,@x3,@x4,@x5 ,@x6,@x7,@x8,@x9,@x10 ,@x11,@x12,@x13,@x14, @x15)" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql
End With With	End With	

<pre> .Parameters.Clear() Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (0).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x1", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(0).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x1", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (1).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x2", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(1).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x2", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim </pre>	<pre> (CStr(dgv_R.Rows(0).Cells (2).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x3", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(2).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x3", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (3).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x4", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(3).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x4", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (4).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x5", SqlDbType.VarChar).Value = System.DBNull.Value </pre>	<pre> Else .Parameters.Add("@x5", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(4).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x5", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (5).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x6", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(5).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x6", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (6).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x7", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(6).Va lue.ToString </pre>
--	---	---

<pre> End If Catch ex As Exception .Parameters.Add("@x7", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (7).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x8", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(7).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x8", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (8).Value))) Then 'ถ้าเป็นค่าว่าง .Parameters.Add("@x9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x9", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(8).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x9", SqlDbType.VarChar).Value = System.DBNull.Value </pre>	<pre> End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (9).Value))) Then .Parameters.Add("@x10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x10", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(9).Va lue.ToString End If Catch ex As Exception .Parameters.Add("@x10", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (10).Value))) Then .Parameters.Add("@x11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x11", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(10).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x11", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (11).Value))) Then </pre>	<pre> .Parameters.Add("@x12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x12", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(11).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x12", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (12).Value))) Then .Parameters.Add("@x13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x13", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(12).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x13", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (13).Value))) Then .Parameters.Add("@x14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x14", </pre>
--	---	---

<pre> SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(13).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x14", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_R.Rows(0).Cells (14).Value))) Then .Parameters.Add("@x15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x15", SqlDbType.VarChar).Value = dgv_R.Rows(0).Cells(14).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x15", SqlDbType.VarChar).Value = System.DBNull.Value End Try .ExecuteNonQuery() End With sql = "insert into RFT (x1,x2,x3,x4,x5,x6,x7,x8,x9, x10,x11,x12,x13,x14,x15)v alues " & _ "(@x1,@x2,@x3,@x4,@x5 ,@x6,@x7,@x8,@x9,@x10 ,@x11,@x12,@x13,@x14, @x15)" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() </pre>	<pre> Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(0).Value))) Then "ถ้าเป็นค่าว่าง .Parameters.Add("@x1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x1", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(0).Val ue.ToString End If Catch ex As Exception .Parameters.Add("@x1", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(1).Value))) Then .Parameters.Add("@x2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x2", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(1).Val ue.ToString End If Catch ex As Exception .Parameters.Add("@x2", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(2).Value))) Then .Parameters.Add("@x3", </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x3", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(2).Val ue.ToString End If Catch ex As Exception .Parameters.Add("@x3", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(3).Value))) Then .Parameters.Add("@x4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x4", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(3).Val ue.ToString End If Catch ex As Exception .Parameters.Add("@x4", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(4).Value))) Then .Parameters.Add("@x5", SqlDbType.VarChar).Value = System.DBNull.Value = </pre>
--	---	---

<pre> dgv_T.Rows(0).Cells(4).Value.ToString End If Catch ex As Exception .Parameters.Add("@x5", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(5).Value))) Then .Parameters.Add("@x6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x6", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(5).Value.ToString End If Catch ex As Exception .Parameters.Add("@x6", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(6).Value))) Then .Parameters.Add("@x7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x7", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(6).Value.ToString End If Catch ex As Exception .Parameters.Add("@x7", </pre>	<pre> SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(7).Value))) Then .Parameters.Add("@x8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x8", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(7).Value.ToString End If Catch ex As Exception .Parameters.Add("@x8", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(8).Value))) Then .Parameters.Add("@x9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x9", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(8).Value.ToString End If Catch ex As Exception .Parameters.Add("@x9", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim </pre>	<pre> (CStr(dgv_T.Rows(0).Cells(9).Value))) Then .Parameters.Add("@x10", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x10", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(9).Value.ToString End If Catch ex As Exception .Parameters.Add("@x10", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(10).Value))) Then .Parameters.Add("@x11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x11", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(10).Value.ToString End If Catch ex As Exception .Parameters.Add("@x11", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(11).Value))) Then .Parameters.Add("@x12", SqlDbType.VarChar).Value = System.DBNull.Value Else </pre>
--	---	--

<pre> .Parameters.Add("@x12", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(11).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x12", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(12).Value))) Then .Parameters.Add("@x13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x13", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(12).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x13", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(12).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x13", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(13).Value))) Then .Parameters.Add("@x14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x14", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(13).V alue.ToString End If </pre>	<pre> Catch ex As Exception .Parameters.Add("@x14", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_T.Rows(0).Cells(14).Value))) Then .Parameters.Add("@x15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x15", SqlDbType.VarChar).Value = dgv_T.Rows(0).Cells(14).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x15", SqlDbType.VarChar).Value = System.DBNull.Value End With .ExecuteNonQuery() End With sql = "insert into RFT (x1,x2,x3,x4,x5,x6,x7,x8,x9, x10,x11,x12,x13,x14,x15)v alues " & "(@x1,@x2,@x3,@x4,@x5 ,@x6,@x7,@x8,@x9,@x10 ,@x11,@x12,@x13,@x14, @x15)" With com .Connection = conn1 .CommandType = CommandType.Text .CommandText = sql .Parameters.Clear() Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(0).Value))) Then </pre>	<pre> .Parameters.Add("@x1", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x1", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(0).Val ue.ToString End If Catch ex As Exception .Parameters.Add("@x1", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(1).Value))) Then .Parameters.Add("@x2", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x2", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(1).Val ue.ToString End If Catch ex As Exception .Parameters.Add("@x2", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(2).Value))) Then .Parameters.Add("@x3", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x3", </pre>
---	--	---

<pre> SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(2).Value.ToString End If Catch ex As Exception .Parameters.Add("@x3", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(3).Value))) Then .Parameters.Add("@x4", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x4", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(3).Value.ToString End If Catch ex As Exception .Parameters.Add("@x4", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(4).Value))) Then .Parameters.Add("@x5", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x5", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(4).Value.ToString End If Catch ex As Exception </pre>	<pre> .Parameters.Add("@x5", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(5).Value))) Then .Parameters.Add("@x6", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x6", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(5).Value.ToString End If Catch ex As Exception .Parameters.Add("@x6", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(6).Value))) Then .Parameters.Add("@x7", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x7", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(6).Value.ToString End If Catch ex As Exception .Parameters.Add("@x7", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try </pre>	<pre> If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(7).Value))) Then .Parameters.Add("@x8", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x8", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(7).Value.ToString End If Catch ex As Exception .Parameters.Add("@x8", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(8).Value))) Then .Parameters.Add("@x9", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x9", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(8).Value.ToString End If Catch ex As Exception .Parameters.Add("@x9", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(9).Value))) Then .Parameters.Add("@x10", </pre>
--	---	--

<pre> SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x10", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(9).Val ue.ToString End If Catch ex As Exception .Parameters.Add("@x10", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(10).Value))) Then .Parameters.Add("@x11", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x11", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(10).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x11", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(11).Value))) Then .Parameters.Add("@x12", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x12", SqlDbType.VarChar).Value = </pre>	<pre> dgv_F.Rows(0).Cells(11).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x12", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(12).Value))) Then .Parameters.Add("@x13", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x13", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(12).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x13", SqlDbType.VarChar).Value = System.DBNull.Value End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(13).Value))) Then .Parameters.Add("@x14", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x14", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(13).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x14", SqlDbType.VarChar).Value = System.DBNull.Value </pre>	<pre> End Try Try If String.IsNullOrEmpty(Trim (CStr(dgv_F.Rows(0).Cells(14).Value))) Then .Parameters.Add("@x15", SqlDbType.VarChar).Value = System.DBNull.Value Else .Parameters.Add("@x15", SqlDbType.VarChar).Value = dgv_F.Rows(0).Cells(14).V alue.ToString End If Catch ex As Exception .Parameters.Add("@x15", SqlDbType.VarChar).Value = System.DBNull.Value End Try .ExecuteNonQuery() End With ds1.Clear() sql = "SELECT * from RFT " da.SelectCommand = New SqlCommand(sql, conn1) da.Fill(ds1, "RFT") DataGridView11.DataSou rce = ds1.Tables("RFT") max_s() For j = 1 To max_id - 1 DataGridView11.Columns(j - 1).HeaderText = DataGridViewX2.Rows(j - 1).Cells(1).Value.ToString Next Dim ij As Integer max_id = max_id - 1 For ij = max_id To 15 Try </pre>
--	--	--

DataGridView1.Columns(max_id).HeaderText = ""	SqlDbType.VarChar).Value = System.DBNull.Value	=
Catch ex As	Else	DataGridViewX1.Rows(i).Cells(4).Value.ToString
Exception	.Parameters.Add("@mult", SqlDbType.VarChar).Value =	End If
End Try	=	Catch ex As
max_id = max_id + 1	DataGridViewX1.Rows(i).Cells(0).Value.ToString	Exception
Next	End If	.Parameters.Add("@err", SqlDbType.VarChar).Value = System.DBNull.Value
	Catch ex As	End Try
	Exception	
exportToWord1(DataGridView1)	.Parameters.Add("@mult", SqlDbType.VarChar).Value = System.DBNull.Value	.ExecuteNonQuery()
End Sub	End Try	End With
Private Sub	Try	Next
Button7_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click	If	ds1.Clear()
ds1.Clear()	String.IsNullOrEmpty(Trim(CStr(DataGridViewX1.Rows(i).Cells(2).Value))) Then	sql = "SELECT mult as[Multiple Linear Regression Equation for PI prediction] ,r_squ as [R-square] ,err as [% Error] from Result "
sql = "delete from Result "	'ถ้าเป็นค่าว่าง	da.SelectCommand = New SqlCommand(sql, conn1)
da.SelectCommand = New SqlCommand(sql, conn1)	.Parameters.Add("@r_squ", SqlDbType.VarChar).Value = System.DBNull.Value	da.Fill(ds1, "Result")
da.Fill(ds1, "Result")	Else	DataGridView9.DataSource = ds1.Tables("Result")
For i = 0 To DataGridViewX1.Rows.Count - 2	.Parameters.Add("@r_squ", SqlDbType.VarChar).Value =	
sql = "insert into Result (mult,r_squ,err)values(@mult,@r_squ,@err)"	DataGridViewX1.Rows(i).Cells(2).Value.ToString	exportToWord2(DataGridView9)
	End If	End Sub
	Catch ex As	End Class
With com	Exception	
conn1	.Parameters.Add("@r_squ", SqlDbType.VarChar).Value = System.DBNull.Value	
.CommandType = CommandType.Text	End Try	
.CommandText = sql	Try	
.Parameters.Clear()	If	
Try	String.IsNullOrEmpty(Trim(CStr(DataGridViewX1.Rows(i).Cells(4).Value))) Then	
If	.Parameters.Add("@err", SqlDbType.VarChar).Value = System.DBNull.Value	
String.IsNullOrEmpty(Trim(CStr(DataGridViewX1.Rows(i).Cells(0).Value))) Then	Else	
.Parameters.Add("@mult",	.Parameters.Add("@err", SqlDbType.VarChar).Value	

BIOGRAPHY

Mr. Tatiya Weerawit was born on the 10th of May 1988 in Bangkok, Thailand. He earned his high school diploma in science-math from Boonwatana School in 2005 and received his Bachelor's Degree in Engineering (Geotechnology) from Suranaree University of Technology (SUT) in 2009. For his post-graduate, he continued to study with a Master's degree in the Petroleum Engineering Program, Institute of Engineering, SUT. During graduation, 2010-2013, he was a part time worker in position of laboratory assistant, teaching assistant and research assistant of SUT. His strong background is in drilling engineering, well logging and reservoir management.

