การสร้างและการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม

นายจักรพันธ์ มหาวันตัง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2550

MATERIALIZED VIEW GENERATION AND SELECTION FOR QUERY REWRITING

Jackapan Mahavantang

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Computer Engineering

Suranaree University of Technology

Academic Year 2007

การสร้างและการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(ผศ. ดร.พิชโยทัย มหัทธนาภิวัฒน์)
ประธานกรรมการ

_____

(รศ. ดร.นิตยา เกิดประสพ)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

_____

(ผศ. ดร.คะชา ชาญศิลป์)
กรรมการ

_____

(ศ. ดร. ไพโรจน์ สัตยธรรม)
รักษาการแทนรองอธิการบดีฝ่ายวิชาการ

_____

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

จักรพันธ์ มหาวันตัง : การสร้างและการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม (MATERIALIZED VIEW GENERATION AND SELECTION FOR QUERY REWRITING) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.นิตยา เกิดประสพ, 90 หน้า.

ฐานข้อมูลในปัจจุบันเต็มไปด้วยข้อมูลมากมายทั้งที่เกิดประโยชน์ และไม่เกิดประโยชน์ จึงเกิดการขุดค้นข้อมูลเพื่อหาความรู้จากข้อมูลเหล่านั้น อย่างไรก็ตาม ในการขุดค้นข้อมูลยังมีปัญหาเรื่องการประมวลผลข้อคำถามที่ต้องใช้เวลานาน จึงได้มีความพยายามที่จะเพิ่มประสิทธิภาพของการประมวลผลข้อคำถามด้วยวิธีการต่าง ๆ วิวข้อมูลได้ถูกนำมาใช้ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม โดยนักวิจัยได้พยายามหาวิธีใช้ประโยชน์จากวิวข้อมูลมาเป็นระยะเวลานาน แต่วิธีการต่าง ๆ ที่เสนอยังมีข้อจำกัดที่วิวข้อมูลจะต้องตรงพอดีกับเงื่อนไขในข้อคำถาม แนวทางการวิจัยของโครงการวิจัยนี้พยายามลดข้อจำกัดดังกล่าว โดยเสนอการสร้างวิวข้อมูลและพยายามหาเกณฑ์คัดเลือกวิวเพื่อพิจารณาวิวข้อมูลที่มีความใกล้เคียงกับข้อคำถามมากที่สุด เพื่อแปลงรูปแบบข้อคำถามให้สามารถประมวลผลได้เร็วที่สุด โดยยังคงความถูกต้องของผลลัพธ์ในการตอบข้อคำถาม

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2550 ลายมือชื่ออาจารย์ที่ปรึกษา_____

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม_____

JACKAPAN MAHAVANTANG : MATERIALIZED VIEW GENERATION AND SELECTION FOR QUERY REWRITING. THESIS ADVISOR : ASSOC. PROF. NITTAYA KERDPRASOP, Ph.D., 90 PP.

MATERIALIZED VIEW SELECTION/QUERY REWRITING

Modern database contains a wealth of information waiting to be discovered and understood. However, finding and presenting this information in a timely fashion can be a major issue, especially when vast amount of data have to be searched. Materialized views help solving this problem. To realize this potential, the query optimizer should know how and when to exploit materialized views. This thesis presents algorithm for determining whether part or all of a query can be estimated from materialized views and describes how it can be combined to rewrite query. The objective of query rewriting is to speed up query evaluation whilst retaining consistent answer.

School of Computer Engineering

Academic Year 2007

Student's Signature_____

Advisor's Signature_____

Co-advisor's Signature_____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคลดังต่อไปนี้ ที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลือ อย่างดียิ่ง ทั้งในด้านวิชาการ และ ด้านการดำเนินงานวิจัย

- รองศาสตราจารย์ ดร.นิตยา เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์
- รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
- ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์ ผู้ช่วยศาสตราจารย์ ดร.คะชา ชาญศิลป์ และ ผู้ช่วยศาสตราจารย์ สมพันธุ์ ชาญศิลป์ อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
- ขอขอบคุณ คุณชลดา พรหมสุข คุณจริยาพร ศรีวิไลลักษณ์ และ คุณกัลญา พับโพธิ์ ที่ช่วยตรวจทานวิทยานิพนธ์ฉบับนี้ และขอบคุณเพื่อนนักศึกษาบัณฑิตศึกษาทุกท่านที่ให้กำลังใจและให้การสนับสนุนงานวิจัยนี้เป็นอย่างดี อันมีส่วนทำให้วิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์เล่มนี้ ผู้วิจัยขอมอบให้กับบิดา มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ผู้วิจัย จนทำให้ประสบความสำเร็จในชีวิตตลอดมา

จักรพันธ์ มหาวันตัง

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก

บทคัดย่อ (ภาษาอังกฤษ) ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่

1 บทนำ 1

1.1 ความสำคัญและที่มาของปัญหาการวิจัย 1

1.2 วัตถุประสงค์ของการวิจัย 3

1.3 ขอบเขตการวิจัย 4

1.4 ประโยชน์ที่คาดว่าจะได้รับ 4

2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 5

2.1 พีชคณิตเชิงสัมพันธ์ (Relational algebra) 5

2.1.1 Selection (σ) 6

2.1.2 Projection (Π) 7

2.1.3 Join ($\bowtie$) 7

2.2 การประมวลผลข้อคำถาม (Query processing) 8

2.2.1 ขั้นตอนการประมวลผลข้อคำถาม 9

2.2.2 การวิเคราะห์ค่าใช้จ่าย (Cost analysis) 12

2.3 การสร้างและการบำรุงรักษาวิวข้อมูล (Materialized view creation and maintenance) 15

2.3.1 วิว (View) และวิวข้อมูล (Materialized view) 15

2.3.2 การสร้างวิวข้อมูล (Materialized view creation) 20

2.3.3 การบำรุงรักษาและการฟื้นฟูข้อมูลของวิวข้อมูล (Maintenance and refreshes of materialized view) 22

สารบัญ (ต่อ)

หน้า

2.4 การเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล (Query optimization with materialized views) 24

2.5 การคัดเลือกวิวข้อมูลและการแปลงรูปแบบข้อคำถามด้วยวิวข้อมูล (Materialized view selection and query rewriting with materialized views) 35

2.5.1 การคัดเลือกวิวข้อมูล (Materialized view selection) 35

2.5.2 การแปลงรูปแบบข้อคำถามด้วยวิวข้อมูล (Query rewriting with materialized views) 37

3 ระเบียบวิธีวิจัย 44

3.1 ขั้นตอนการวิจัย 44

3.2 โปรแกรม MV4QR สำหรับสร้างและคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม 45

3.2.1 เทคนิคการพิจารณาข้อมูลเพื่อที่จะนำมาสร้างเป็นวิวข้อมูล 46

3.2.2 เทคนิคในการคัดเลือกวิวข้อมูลเพื่อการแปลงรูปแบบข้อคำถาม 56

3.3 การทำงานของโปรแกรม MV4QR 61

3.4 แหล่งที่มาของข้อมูล 64

3.5 วิธีการทดสอบ 66

4 ผลการวิเคราะห์ข้อมูลและการอภิปรายผล 67

4.1 ผลการวิเคราะห์สถิติการใช้ข้อคำถามเพื่อเตรียมวิวข้อมูล 67

4.2 ผลการทดสอบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล TPC-H 67

4.3 ผลการทดสอบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล SH Schema 72

5 สรุปผลการวิจัยและข้อเสนอแนะ 77

5.1 สรุปผลการวิจัย 78

5.1.1 การทดสอบโปรแกรม MV4QR ให้ทำการสร้างวิวข้อมูล 78

5.1.2 การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดเล็กและป้อนข้อคำถามตรงกับวิวข้อมูลที่มีอยู่ 78

สารบัญ (ต่อ)

หน้า

5.1.3 การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดเล็ก และป้อนข้อคำถามที่ไม่ตรงกับวิวข้อมูลที่มีอยู่ 78

5.1.4 การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดใหญ่ และป้อนข้อคำถามตรงกับวิวข้อมูลที่มีอยู่ 78

5.1.5 การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดใหญ่ และป้อนข้อคำถามที่ไม่ตรงกับวิวข้อมูลที่มีอยู่ 78

5.2 การประยุกต์งานวิจัย 79

5.3 ข้อเสนอแนะ 79

รายการอ้างอิง 80

ภาคผนวก

ภาคผนวก ก. บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 32 82

ประวัติผู้เขียน 90

สารบัญตาราง

ตารางที่ **หน้า**

3.1 แสดงตัวอย่างข้อมูลที่ได้จากวิเคราะห์สถิติการประมวลผลข้อคำถาม 48
3.2 แสดงตัวอย่างข้อมูลที่ได้จากการวิเคราะห์สถิติการเข้าถึงวิวข้อมูล 51
3.3 แสดงตัวอย่างข้อมูลที่ผ่านเงื่อนไข 53
3.4 แสดงรายละเอียดของข้อมูล TPC-H 65
3.5 แสดงรายละเอียดของข้อมูล SH Schema 65
4.1 ข้อคำถามที่ใช้ทดสอบกับชุดข้อมูล TPC-H 68
4.2 ตารางผลการทดสอบการประมวลผลข้อคำถาม กรณีแปลงรูปแบบข้อคำถามได้ 70
4.3 ตารางผลการทดสอบการประมวลผลข้อคำถาม กรณีไม่สามารถแปลงข้อคำถามได้ 71
4.4 ตารางข้อคำถามที่ทดสอบกับชุดข้อมูล SH Schema 73
4.5 ตารางผลการทดสอบการประมวลผลข้อคำถามกรณีตรงกับวิวข้อมูล 74
4.6 ตารางผลการทดสอบการประมวลผลข้อคำถาม กรณีไม่ตรงกับวิวข้อมูล 75

สารบัญรูป

รูปที่ **หน้า**

1.1 คำสั่งในการสร้างวิวข้อมูล C_KORAT_MV 2
1.2 ตัวอย่างเส้นทางในการเข้าถึงข้อมูล โดยใช้ (ก) ตารางข้อมูลพื้นฐาน (ข) วิวข้อมูล 3
2.1 ตัวอย่างการทำ Selection 6
2.2 ตัวอย่างการทำ Projection 7
2.3 ตัวอย่างการทำ Join 8
2.4 ขั้นตอนของการประมวลผลข้อคำถาม 9
2.5 เปรียบเทียบลำดับการประมวลผลข้อคำถามด้วยแผนที่ต่างกัน 12
2.6 อัลกอริทึม Nested loop join ด้วยเงื่อนไข R.A=S.B 14
2.7 ตัวอย่างการทำ Sort merge join ด้วยเงื่อนไข R.A=S.B 14
2.8 อัลกอริทึม Hash join ด้วยเงื่อนไขR.A=S.B 15
2.9 ตัวอย่างเงื่อนไขในการสร้างวิว 16
2.10 ตัวอย่างการสร้างวิว 16
2.11 ตัวอย่างการใช้คำสั่ง DESCRIBE กับวิว 17
2.12 ตัวอย่างการสอบถามข้อมูลจากวิว 17
2.13 ตัวอย่างการสอบถามข้อมูลจากตารางพื้นฐาน 18
2.14 วิวข้อมูลแบบง่ายและแบบซับซ้อน 19
2.15 รูปแบบการใช้คำสั่ง create materialized view 20
2.16 ตัวอย่างการสร้างวิวข้อมูล 21
2.17 การกำหนดการฟื้นฟูวิวข้อมูลโดยอัตโนมัติ 22
2.18 การกำหนดการฟื้นฟูวิวข้อมูลแบบทำด้วยมือ 23
2.19 การกำหนดการฟื้นฟูวิวข้อมูลแบบหลายวิว 23
2.20 ตัวอย่างแผนการบำรุงรักษาวิวข้อมูล 24
2.21 ตัวอย่างเส้นทางในการเข้าถึงข้อมูล โดยใช้ (ก) ตารางข้อมูลพื้นฐาน (ข) วิวข้อมูล 25
2.22 แผนการประมวลผลข้อคำถาม Query 1A 28
2.23 ผลจากการประมวลผลชุดคำสั่ง Query 1A 28

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

2.24 แผนการประมวลผลข้อคำถาม Query 1B 29
2.25 ผลจากการประมวลผลชุดคำสั่ง Query1B 29
2.26 แผนการประมวลผลข้อคำถาม Query 2A 31
2.27 ผลจากการประมวลผลชุดคำสั่ง Query2A 32
2.28 แผนการประมวลผลข้อคำถาม Query 2B 32
2.29 ผลจากการประมวลผลชุดคำสั่ง Query 2B 33
2.30 ตัวอย่างการคัดเลือกวิวข้อมูล 35
2.31 โครงสร้างสำหรับการคัดเลือกวิวข้อมูลแบบอัตโนมัติ 36
2.32 วิวข้อมูล all_cust_sales_mv 38
2.33 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Exact match 38
2.34 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Join back 39
2.35 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Rollup & aggregate rollup 39
2.36 วิวข้อมูล some_cust_sales_mv 40
2.37 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Data subset 41
2.38 วิวข้อมูล east_sales_mv 42
2.39 วิวข้อมูล north_sales_mv 42
2.40 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธีใช้หลายวิวข้อมูล 43
3.1 โครงสร้างของโปรแกรม MV4QR 45
3.2 ขั้นตอนการคัดเลือกข้อมูลเพื่อนำมาสร้างเป็นวิวข้อมูล 46
3.3 แสดงผังการทำงานส่วนวิเคราะห์สถิติการประมวลผลข้อคำถาม 47
3.4 อัลกอริทึมการวิเคราะห์สถิติการประมวลผลข้อคำถาม 48
3.5 แสดงผังการทำงานส่วนที่วิเคราะห์สถิติการเข้าถึงวิวข้อมูล 50
3.6 อัลกอริทึมการวิเคราะห์สถิติการเข้าถึงวิวข้อมูล 51
3.7 แสดงผังการทำงานส่วนตรวจสอบเงื่อนไขเพื่อนำไปสร้างวิวข้อมูล 52
3.8 อัลกอริทึมการตรวจสอบเงื่อนไขเพื่อนำไปสร้างวิวข้อมูล 53
3.9 แสดงผังการทำงานส่วนสร้างวิวข้อมูล 54
3.10 อัลกอริทึมการสร้างวิวข้อมูล 55

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

3.11 ตัวอย่างคำสั่งในการสร้างวิวข้อมูลจากข้อคำถาม 55
3.12 ตัวอย่างคำสั่งในการสร้างวิวข้อมูลจากวิว 56
3.13 แสดงผังการทำงานส่วนคัดเลือกวิวข้อมูล 56
3.14 ตัวอย่างการหาความสัมพันธ์ของข้อคำถามกับวิวข้อมูล 57
3.15 แสดงผังการทำงานส่วนการคัดเลือกวิวข้อมูลที่สามารถตอบข้อคำถามได้ 58
3.16 ตัวอย่างการหาความสัมพันธ์ของข้อคำถามย่อยกับวิวข้อมูล 59
3.17 แสดงผังการทำงานส่วนการคัดเลือกวิวข้อมูลที่สามารถตอบข้อคำถามได้ 60
3.18 อัลกอริทึมการคัดเลือกวิวข้อมูล (MVS) 60
3.19 อัลกอริทึมการคัดเลือกวิวข้อมูลที่ดีที่สุด (MvBestSelect) 61
4.1 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิม กับข้อคำถามที่แปลงรูปแบบแล้วในกรณีที่สามารถแปลงรูปแบบข้อคำถามได้ 71
4.2 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิม กับข้อคำถามที่แปลงรูปแบบแล้วในกรณีที่ไม่สามารถแปลงรูปแบบข้อคำถามได้ 72
4.3 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิม กับข้อคำถามที่แปลงรูปแบบแล้ว (ข้อมูล SH Schema) 75
4.4 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิม กับข้อคำถามที่แปลงรูปแบบแล้วในกรณีที่ไม่สามารถแปลงรูปแบบข้อคำถามได้ (ข้อมูล SH Schema) 76

บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงความสำคัญ ที่มาของปัญหาการวิจัย วัตถุประสงค์ของงานวิจัย ขอบเขตงานวิจัย และประโยชน์ที่จะได้รับจากงานวิจัย

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ปัจจุบันการจัดเก็บฐานข้อมูลของภาคธุรกิจนั้นมีระบบการจัดเก็บข้อมูลที่ซับซ้อนมากขึ้นและเต็มไปด้วยข้อมูลจำนวนมหาศาลทั้งที่เกิดประโยชน์และไม่เกิดประโยชน์ การเข้าถึงฐานข้อมูลให้ตรงกับความต้องการสำหรับการนำไปใช้งาน เพื่อให้ข้อมูลและสารสนเทศที่มีอยู่เกิดประโยชน์กับองค์กรได้อย่างมีประสิทธิภาพยังมีปัญหาเรื่องการประมวลผลข้อคำถามที่ต้องใช้เวลานาน ในปัจจุบันระบบจัดการฐานข้อมูลขนาดใหญ่เช่น Oracle, Microsoft SQL Server, IBM DB2 ได้เพิ่มส่วนที่เรียกว่า ส่วนเพิ่มประสิทธิภาพข้อคำถาม (Query optimizer) ไว้ในส่วนประมวลผลข้อคำถามเพื่อทำหน้าที่ปรับปรุงรูปแบบข้อคำถามให้สามารถประมวลผลได้เร็วขึ้น โดยที่ยังคงให้คำตอบที่ถูกต้องได้เช่นเดิม

ลักษณะที่สำคัญอย่างหนึ่งของระบบฐานข้อมูลคือ มีความเป็นอิสระของข้อมูลสูง (High degree of data independence) เป็นการพัฒนาโปรแกรมที่เป็นอิสระต่อการเปลี่ยนแปลงโครงสร้างของระบบฐานข้อมูล ไม่ว่าจะมีการเปลี่ยนแปลงโครงสร้างของข้อมูลแบบใดก็ตาม คำสั่งหรือโปรแกรมที่ได้พัฒนาไว้แล้วก็ยังคงสามารถใช้โปรแกรมนั้นต่อไปได้โดยไม่ต้องมีการพัฒนาโปรแกรมใหม่ ความเป็นอิสระของข้อมูลจำแนกได้เป็นสองระดับคือระดับตรรกะและระดับกายภาพ

1) ความเป็นอิสระของข้อมูลในระดับตรรกะ (Logical data independence)

ความเป็นอิสระของข้อมูลในระดับตรรกะ หมายถึงความเป็นอิสระของโครงสร้างข้อมูลในระดับแนวคิด (Conceptual level) กับระดับภายนอก (External level) ถ้ามีการเปลี่ยนแปลงโครงสร้างข้อมูลในระดับแนวคิดก็จะไม่มีผลต่อการใช้งานในระดับภายนอก ไม่ว่าจะมีการปรับโครงสร้างข้อมูลอย่างไรก็ตาม ความเป็นอิสระของข้อมูลในลักษณะนี้ ถ้ามีการเปลี่ยนแปลงโครงสร้างของข้อมูลที่ไม่เป็นโครงสร้างหลัก โปรแกรมที่ได้พัฒนาไว้แล้วก็ยังคงสามารถใช้ต่อไปได้

2) ความเป็นอิสระของข้อมูลในระดับกายภาพ (Physical data independence)

ความเป็นอิสระของข้อมูลในระดับกายภาพ หมายถึงความเป็นอิสระของข้อมูลในระดับภายใน (Internal level) กับระดับแนวคิด (Conceptual level) และระดับภายนอก (External level) เช่น การเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม โดยการปรับเปลี่ยนโครงสร้างภายในซึ่งจะไม่กระทบกระเทือนต่อเค้าร่าง (Schema) ในระดับแนวคิดหรือเค้าร่างในระดับภายนอก

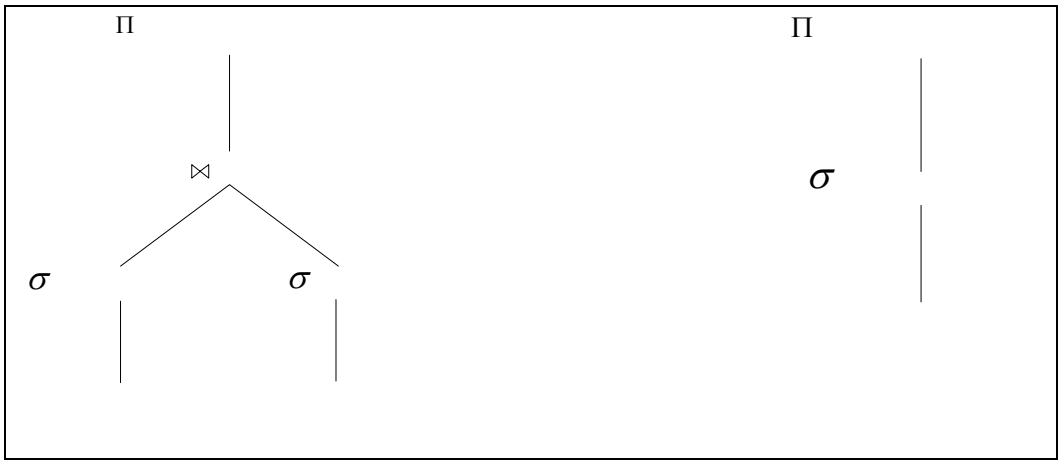
การเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล (Materialized view) เป็นการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามโดยการปรับเปลี่ยนโครงสร้างระดับกายภาพ โดยวิวข้อมูลสร้างขึ้นจากข้อมูลในตารางพื้นฐาน (Base table) หรือตารางหลักที่ถูกเก็บในฐานข้อมูล ข้อมูลที่ตรงกับเงื่อนไขที่กำหนดในคำสั่งสร้างวิวถูกนำมาเก็บไว้เป็นการถาวรเปรียบเหมือนตารางข้อมูลทั่วไป จึงช่วยให้การตอบข้อคำถามทำได้อย่างรวดเร็วยิ่งขึ้น ตัวอย่างเช่น มีข้อมูลชื่อและหมายเลขโทรศัพท์ของลูกค้าถูกเก็บไว้ที่ตาราง CUSTOMER ข้อมูลของรายชื่อจังหวัดถูกเก็บไว้ที่ตาราง PROVINCE จากตารางข้อมูลทั้งสองนี้สามารถนำมาสร้างเป็นวิวข้อมูล C_KORAT_MV ได้ตามคำสั่งดังแสดงในรูปที่ 1.1

```
CREATE MATERIALIZED VIEW  "C_KORAT_MV"
AS select        "CUSTOMER"  .  "C_ID" as "C_ID",
                 "CUSTOMER"  .  "C_NAME" as "C_NAME",
                 "CUSTOMER"  .  "C_ADDRESS" as "C_ADDRESS",
                 "CUSTOMER"  .  "C_PHONE" as "C_PHONE",
                 "CUSTOMER"  .  "C_GENDER" as "C_GENDER"
 from            "PROVINCE"     "PROVINCE",
                 "CUSTOMER"     "CUSTOMER"
where   "PROVINCE" . "PROVINCE_ID"  =  "CUSTOMER"."PROVINCE_ID"
and     "PROVINCE"."PROVINCE_NAME"  =  'NAKHONRATCHASIMA'
```

รูปที่ 1.1 คำสั่งในการสร้างวิวข้อมูล C_KORAT_MV

เมื่อต้องการสอบถามข้อมูลชื่อและหมายเลขโทรศัพท์ของลูกค้าที่อาศัยอยู่ที่จังหวัดนครราชสีมา (NAKHONRATCHASIMA) จะเห็นว่าทั้งตารางข้อมูลพื้นฐานและวิวข้อมูล C_KORAT_MV สามารถตอบข้อคำถามนี้ได้ ดังรูปที่ 1.2 เป็นการเปรียบเทียบเส้นทางในการเข้าถึงข้อมูลโดยการเรียกใช้ข้อมูลจากตารางพื้นฐานกับการเรียกใช้ข้อมูลผ่านทางวิวข้อมูล จากรูปจะเห็นว่า

เส้นทางในการเข้าถึงข้อมูลโดยใช้ตารางข้อมูลพื้นฐาน (รูปที่ 1.2 (ก)) จะมีกระบวนการเข้าถึงข้อมูลซับซ้อนมากกว่าเส้นทางในการเข้าถึงข้อมูลโดยใช้วิวข้อมูล (รูปที่ 1.2 (ข)) เนื่องจากวิวข้อมูลจะมีการประมวลผลไว้ก่อนแล้ว (Pre-computing) ทำให้กระบวนการเข้าถึงข้อมูลนี้ซับซ้อนน้อยลง นั่นคือมีประสิทธิภาพในการประมวลผลข้อคำถามที่ดีขึ้นกว่าการเข้าถึงข้อมูลในตารางข้อมูลพื้นฐาน



รูปที่ 1.2 ตัวอย่างเส้นทางในการเข้าถึงข้อมูล โดยใช้ (ก) ตารางข้อมูลพื้นฐาน (ข) วิวข้อมูล

วิวข้อมูลได้ถูกนำมาใช้ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม โดยนักวิจัยได้พยายามหาวิธีใช้ประโยชน์จากวิวข้อมูลมาเป็นระยะเวลานาน แต่วิธีการต่างๆที่เสนอยังมีข้อจำกัดที่วิวข้อมูลจะต้องมีโครงสร้างที่ตรงพอดีกับเงื่อนไขในข้อคำถาม

แนวทางการวิจัยของโครงการวิจัยนี้พยายามลดข้อจำกัดดังกล่าว โดยเสนอวิธีการสร้างวิวข้อมูลและพยายามหาเกณฑ์คัดเลือกวิวเพื่อพิจารณาวิวข้อมูลที่มีความเหมาะสมกับข้อคำถามมากที่สุด เพื่อแปลงรูปแบบข้อคำถามให้สามารถประมวลผลได้เร็วที่สุด

1.2 วัตถุประสงค์ของการวิจัย

- ศึกษาวิธีการเก็บสถิติของฐานข้อมูลที่สามารถช่วยในการเลือกข้อมูลมาสร้างเป็นวิวข้อมูล
- ศึกษาวิธีการนำวิวข้อมูลเข้ามาช่วยเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม
- พัฒนาแนวทางการนำวิวข้อมูล มาใช้ช่วยเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม
- กำหนดวิธีการคัดเลือกวิวข้อมูล และวิธีการใช้วิวข้อมูลเพื่อแปลงรูปแบบคำสั่ง SQL
- ออกแบบเกณฑ์ในการประเมินเวลาที่ใช้ในการประมวลผลข้อคำถามในกรณีที่มีการใช้วิวข้อมูล
- ทดสอบวิธีการใช้วิวข้อมูลเพื่อปรับปรุงประสิทธิภาพการประมวลผลข้อคำถาม

1.3 ขอบเขตการวิจัย

งานวิจัยนี้เป็นการศึกษาและพัฒนาแนวทางในการนำข้อมูลทางสถิติและวิวข้อมูลเข้ามาช่วยในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม และการคัดเลือกข้อมูลมาสร้างเป็นวิวข้อมูล ดังนั้นการออกแบบ การวิเคราะห์ และการทดสอบจะจำกัดเฉพาะในส่วน Query optimizer และวิธีการนำข้อมูลทางสถิติมาช่วยในการคัดเลือกข้อมูลเพื่อนำไปสร้างเป็นวิวข้อมูล

1.4 ประโยชน์ที่คาดว่าจะได้รับ

โครงการวิจัยนี้มีจุดมุ่งหมายที่จะพัฒนาแนวทางในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล สิ่งที่ได้จากการวิจัยนี้จะเป็นแนวทางใหม่ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม และแนวทางในการนำข้อมูลทางสถิติมาช่วยในการสร้างวิวข้อมูลเพื่อให้ได้วิวข้อมูลที่สามารถนำมาใช้ประโยชน์ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามได้อย่างเต็มที่

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะเป็นการนำเสนอทฤษฎีพื้นฐานของการประมวลผลข้อคำถาม วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับงานวิจัยด้านวิวข้อมูลและการประมวลผลข้อคำถาม โดยหัวข้อที่ 2.1 จะกล่าวถึงพีชคณิตเชิงสัมพันธ์ (Relational algebra) ซึ่งถือว่าเป็นพื้นฐานที่สำคัญของระบบฐานข้อมูลเชิงสัมพันธ์ โดยเฉพาะในด้านการประมวลผลข้อคำถาม ในหัวข้อที่ 2.2 จะกล่าวถึงวิธีการประมวลผลข้อคำถาม (Query processing) ในหัวข้อ 2.3 จะกล่าวถึงการสร้างและการบำรุงรักษาวิวข้อมูล (Materialized view creation and maintenance) ในหัวข้อที่ 2.4 จะกล่าวถึงการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล (Query optimization with materialized views) หัวข้อที่ 2.5 จะกล่าวถึงการคัดเลือกวิวข้อมูลและการแปลงรูปแบบข้อคำถามด้วยวิวข้อมูล (Materialized view selection and query rewriting with materialized views)

2.1 พีชคณิตเชิงสัมพันธ์ (Relational algebra)

แนวคิดของฐานข้อมูลเชิงสัมพันธ์ที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน เป็นแนวคิดที่มีหลักการซึ่งประกอบด้วยรูปแบบทางคณิตศาสตร์ที่สนับสนุนอยู่เบื้องหลัง คือ พีชคณิตเชิงสัมพันธ์

พีชคณิตเชิงสัมพันธ์ (Relational algebra) คือ ทฤษฎีทางภาษาที่ใช้หลักการพีชคณิตสำหรับการปฏิบัติการกับรีเลชันหลักเพื่อสร้างรีเลชันใหม่ขึ้นมา ผลของการปฏิบัติการใด ๆ ที่เกิดขึ้นจะไม่มีผลกระทบหรือเกิดการเปลี่ยนแปลงใด ๆ ในรีเลชันหลัก รูปแบบการสร้างประโยคภาษาในพีชคณิตเชิงสัมพันธ์จะมีอยู่หลายคำสั่งด้วยกัน ซึ่งได้กำหนดไว้ทั้งสิ้น 8 โอเปอเรชัน แต่อย่างไรก็ตาม ก็มีโอเปอเรชันที่ได้มีการพัฒนาขึ้นมาใหม่ แต่จะมีโอเปอเรชันพื้นฐาน 5 โอเปอเรชัน คือ

1. Selection
2. Projection
3. Cartesian product
4. Union
5. Set difference

และจะมีส่วนเพิ่มเติมอีก 3 โอเปอเรชัน คือ

1. Join
2. Intersection
3. Division

ในงานวิจัยนี้จะให้ความสนใจไปที่โอเปอเรชันที่ถูกใช้บ่อยในการสอบถามข้อมูลได้แก่ Selection, Projection และ Join ซึ่งทั้งสามโอเปอเรชันมีรายละเอียดดังนี้

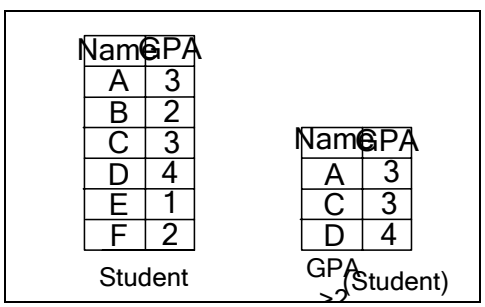
2.1.1 Selection (σ)

โอเปอเรชัน Selection เป็นการเลือกแถวจากความสัมพันธ์หรือรีเลชัน ซึ่งจะได้ผลลัพธ์เป็นรีเลชันใหม่โดยจะมีจำนวนทูเพิล (Tuple) หรือแถวจากรีเลชัน R ที่ตรงตามเงื่อนไข (Predicate) ที่กำหนดขึ้น ผลที่ได้อาจมีจำนวนทูเพิลตั้งแต่ศูนย์ขึ้นไป และผลลัพธ์อาจจะมีจำนวนแถวที่น้อยกว่าหรือเท่ากับจำนวนแถวทั้งหมดที่มีในความสัมพันธ์ R แต่จะไม่มากกว่ารูปแบบการเขียนคำสั่งแสดงได้ดังนี้

$$\sigma_{predicate}(R)$$

σ	คือ สัญลักษณ์แทนตัวกระทำการ Selection
Predicate	คือ ประโยควลีหรือเงื่อนไขที่กำหนดขึ้น
R	คือ รีเลชันที่มีการทำ Selection

ตัวอย่างเช่น $\sigma_{GPA>2}$ (Student) เป็นการเลือกแถวจากความสัมพันธ์ Student โดยมีเงื่อนไขว่าในแต่ละแถวที่เลือกมานั้นจะต้องมีค่า GPA > 2 ผลลัพธ์ที่ได้จะเป็นรีเลชันใหม่ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างการทำ Selection

2.1.2 Projection (Π)

โอเปอเรชัน Projection เป็นการเลือกเฉพาะคอลัมน์ที่ต้องการจากความสัมพันธ์ กล่าวคือ เป็นการเลือกแอททริบิวต์ที่ต้องการในรีเลชันที่กำหนด ซึ่งจะได้ผลลัพธ์ในรีเลชันใหม่ โดยจะมีแอททริบิวต์ที่ได้เลือกไว้ และข้อมูลที่ได้แสดงตามเงื่อนไข รูปแบบการเขียนคำสั่งแสดงได้ดังนี้

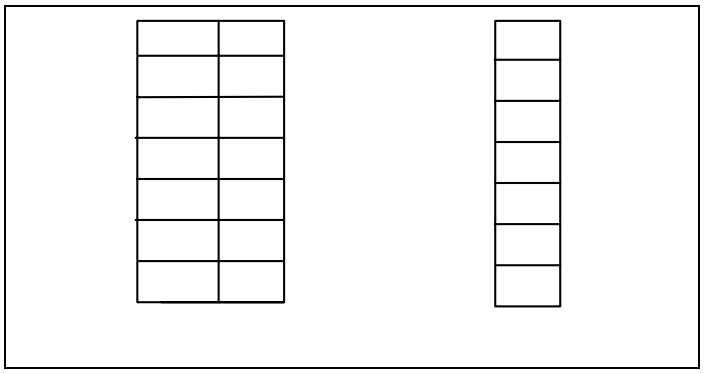
$$\Pi_{col1,...,coln}(R)$$

Π คือ สัญลักษณ์แทนตัวกระทำการ Projection

$col_{1,...,}col_n$ คือ คอลัมน์หรือแอททริบิวต์ที่เลือก

R คือ รีเลชันที่มีการทำ Projection

ตัวอย่างเช่น Π_{GPA} (Student) เป็นการเลือกเฉพาะคอลัมน์ GPA จากรีเลชัน Student ผลลัพธ์ที่ได้จะเป็นรีเลชันใหม่ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างการทำ Projection

2.1.3 Join ($\bowtie$)

โอเปอเรชัน Join เป็นการรวม 2 รีเลชันเพื่อสร้างเป็นรีเลชันใหม่ การ Join จะเป็นโอเปอเรชันที่ค่อนข้างยากสำหรับการพัฒนาการออกแบบเพื่อจัดการฐานข้อมูลเพื่อให้ได้มาซึ่งประสิทธิภาพ รูปแบบการเขียนคำสั่งแสดงได้ดังนี้

$R \bowtie_F S$

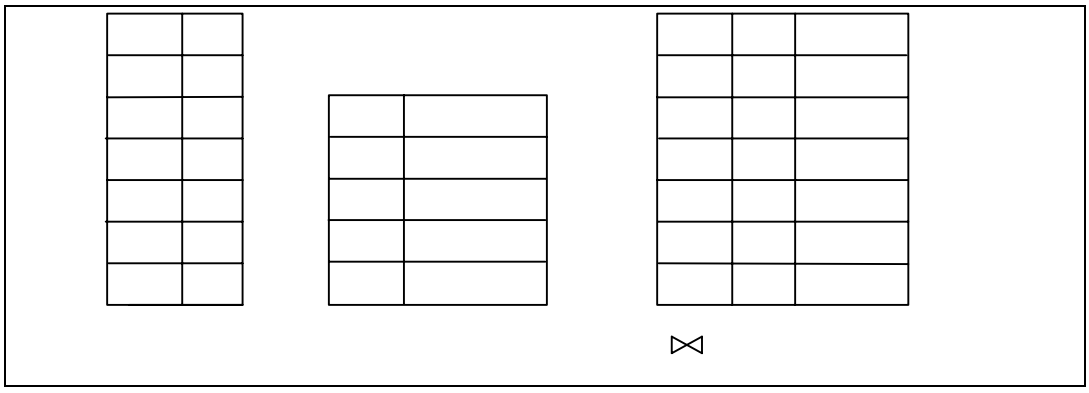
$\bowtie$ คือ สัญลักษณ์แทนตัวกระทำการ Join

R, S คือ รีเลชั่นที่ทำการ Join

F คือ เงื่อนไขที่กำหนดอยู่ในรูป R.ai θ S.bi

เมื่อ θ คือ โอเปอเรชันการเปรียบเทียบ

ตัวอย่างเช่น Student $\bowtie_{\text{Student.GPA=Division.GPA}}$ Division เป็นการรวมรีเลชัน Student เข้ากับรีเลชัน Division โดยใช้คอลัมน์ GPA เป็นเงื่อนไขในการสร้างความสัมพันธ์ขึ้นมาใหม่ระหว่างรีเลชัน Student กับรีเลชัน Division ผลลัพธ์ที่ได้จะเป็นรีเลชันใหม่ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างการทำ Join

2.2 การประมวลผลข้อคำถาม (Query processing)

การประมวลผลข้อคำถาม (Query processing) เป็นกระบวนการหนึ่งของระบบการจัดการฐานข้อมูลที่ใช้ในการเลือกแผนหรือกลยุทธ์ที่เหมาะสมในการสอบถามข้อมูลในฐานข้อมูล โดยระบบจัดการฐานข้อมูลจะมีตัวประมวลผลที่เรียกว่า ตัวประมวลผลข้อคำถาม (Query processor) ทำหน้าที่ในการเลือกแผนหรือกลยุทธ์ที่เหมาะสมในการเข้าถึงข้อมูลเพื่อให้ได้คำตอบที่ดีที่สุดและถูกต้องโดยข้อคำถามจะเกิดจากภาษาสอบถาม เช่น ภาษาเอสคิวแอล (Structured query language : SQL) ภาษาคิวบีอี (Query by example : QBE) เป็นต้น

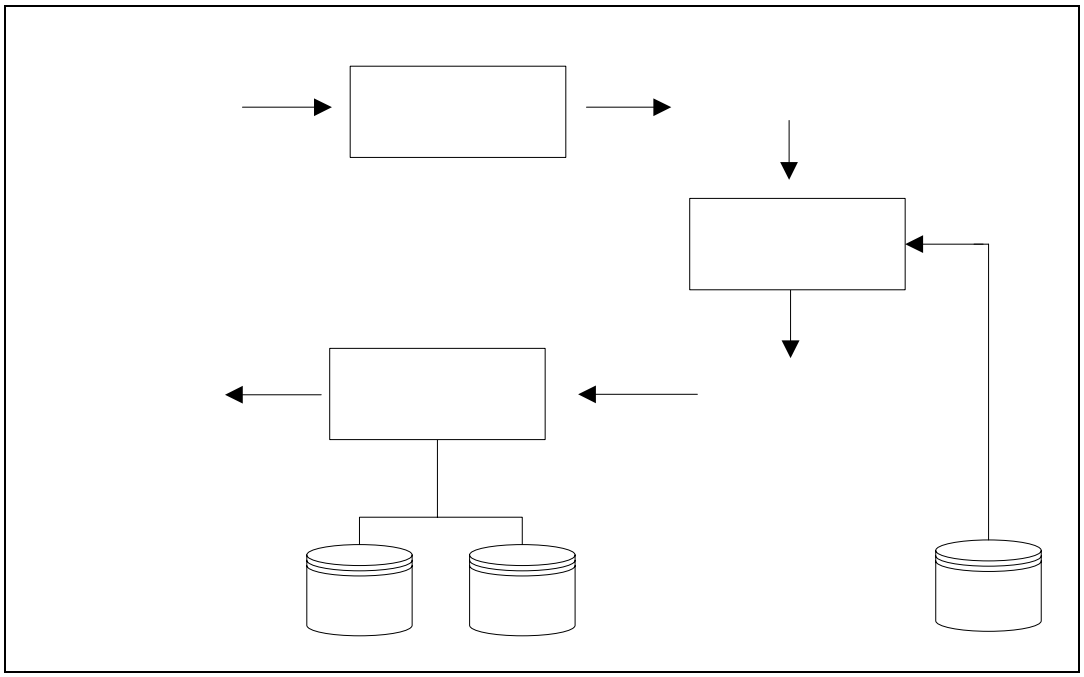
Name GPA
A 3
B 2

GPA Cla

2.2.1 ขั้นตอนการประมวลผลข้อคำถาม

ขั้นตอนของการประมวลผลข้อคำถามดังที่แสดงใน รูปที่ 2.4 มีสามขั้นตอน คือ

1) การตรวจสอบไวยากรณ์และการแปลความหมาย (Parsing and translation)
2) การปรับปรุงประสิทธิภาพ (Optimization)
3) การประมวลผล (Evaluation)



รูปที่ 2.4 ขั้นตอนของการประมวลผลข้อคำถาม

1) การตรวจสอบไวยากรณ์และการแปลความหมาย (Parsing and translation)

การตรวจสอบไวยากรณ์และการแปลความหมายด้วย Parser และ Translator เป็นส่วนหนึ่งของตัวประมวลผลข้อคำถามที่ทำหน้าที่ตรวจเช็คกฎและไวยากรณ์ (Syntax) ของข้อคำถาม รวมทั้งตรวจสอบความถูกต้องของชื่อข้อมูล โดยจะทำการตรวจสอบกับข้อมูลที่อยู่ในพจนานุกรมข้อมูล (Data dictionary) แล้วทำการแปลคำสั่งให้อยู่ในรูปของนิพจน์พีชคณิตเชิงสัมพันธ์ (Relational algebraic expression) ที่ระบบจัดการฐานข้อมูลสามารถเข้าใจได้ ก่อนที่การประมวลผลข้อคำถามจะเริ่มขึ้นระบบจะต้องแปลงข้อคำถามให้อยู่ในรูปแบบที่สามารถใช้งานได้ เนื่องจากภาษาเอสคิวแอล (SQL) เป็นภาษาระดับสูงที่มนุษย์ใช้ในการสอบถามข้อมูล แต่ไม่เหมาะสมกับการทำงานของระบบ จำเป็นต้องทำการแปลคำสั่งให้อยู่ในรูปของนิพจน์พีชคณิตเชิงสัมพันธ์ที่ระบบจัดการฐานข้อมูลสามารถเข้าใจได้ เช่น ตัวอย่างเช่น

Query Parser transla

Query output Evaluati engine

กำหนดให้มีข้อคำถามดังนี้

```
SELECT    balance
FROM      account
WHERE     balance<2500
```

ข้อคำถามนี้สามารถ เขียนให้อยู่ในรูปของนิพจน์พีชคณิตเชิงสัมพันธ์ได้สองรูปแบบดังนี้

$$\sigma_{balance<2500}(\prod_{balance}(account))$$

หรือ

$$\prod_{balance}(\sigma_{balance<2500}(account))$$

2) การปรับปรุงประสิทธิภาพและการประมวลผล (Optimization and evaluation)

การปรับปรุงประสิทธิภาพและการประมวลผลด้วย Optimizer และ Evaluation engine เป็นส่วนหนึ่งของตัวประมวลผลข้อคำถามที่ทำหน้าที่ในการหากลยุทธ์หรือวิธีการที่เหมาะสม เพื่อให้การเข้าถึงข้อมูลประหยัดเวลาที่สุด เมื่อคำสั่งในภาษาสอบถามผ่านตัวตรวจสอบภาษา (Parser) แล้วจะได้คำสั่งในการสอบถามที่อยู่ในรูปของนิพจน์พีชคณิตเชิงสัมพันธ์ซึ่งสามารถประมวลผลได้หลายวิธีด้วยเทคนิคต่าง ๆ ในการหาทางเลือกที่เข้าถึงข้อมูลได้อย่างเหมาะสม เช่น เทคนิคการซอร์ตและการเมอร์จ (Sorting and merge method) เป็นการเรียงลำดับข้อมูลในตารางข้อมูลโดยระบบจัดการฐานข้อมูลจะมีโปรแกรมสำหรับจัดเรียงลำดับข้อมูลเพื่อให้ใช้เวลาในการประมวลผลลดลง เทคนิคดีคอมโพสิชั่น (Decomposition method) เป็นการแตกคำสั่งในภาษาสอบถามให้เป็นส่วนย่อย ๆ เพื่อง่ายต่อการประมวลผล เทคนิคโอเปอร์เรเตอร์กราฟ (Operator graph) เป็นการนำนิพจน์พีชคณิตที่ได้จากการตรวจเช็คไวยากรณ์ของภาษาแล้วนำมาหาทางการประมวลผลที่ให้การประมวลผลนั้นใช้เวลาน้อยที่สุด เทคนิคการนำวิวข้อมูล (Materialized view) มาช่วยในการตอบคำถาม

เทคนิคต่าง ๆ เหล่านี้จะทำให้ได้แผนในการประมวลผลที่เหมาะสมเพื่อให้ได้วิธีการเข้าถึงข้อมูลที่มีประสิทธิภาพและใช้เวลาในการประมวลผลน้อยที่สุด เมื่อผู้ใช้ป้อนข้อคำถามที่อยู่ในรูปแบบภาษาเอสคิวแอลเข้ามา ขั้นตอนแรกตัวประมวลผลข้อคำถาม (Query processor) จะทำการตรวจสอบความถูกต้องตามไวยากรณ์ของข้อคำถามนั้น และทำการตรวจสอบความสัมพันธ์

และคุณลักษณะที่ถูกอ้างถึงในฐานข้อมูล ถ้าข้อคำถามที่เข้ามาถูกต้องตามข้อกำหนด ขั้นตอนต่อมาคือ กำหนดแผนการประมวลผลข้อคำถาม นั่นคือ การกำหนดลำดับขั้นตอนในการดำเนินการกับข้อคำถามนั้น ในแต่ละขั้นตอนจะมีลักษณะเช่นเดียวกันคือ การทำงานร่วมกันระหว่างโอเปอเรชันความสัมพันธ์กับวิธีการประมวลผล ตัวอย่างเช่น จอยน์ (Join) โอเปอเรชัน สามารถประมวลผลได้โดยวิธี Nested loop และ Sort merge โดยพิจารณาจากข้อคำถามต่อไปนี้

```
SELECT      *
FROM        R, S, T
WHERE       R.A > a AND R.B  = S.B AND S.C = T.C
```

เมื่อ R S และ T คือ ความสัมพันธ์ A B และ C คือ คุณลักษณะหรือแอททริบิวต์ (attribute) และ a คือ ค่าคงที่ที่ใช้กำหนดเงื่อนไข ข้อคำถามนี้จะประกอบไปด้วยการจอยน์ (Join) สองครั้งและการซีเลคท์ (Select) หนึ่งครั้ง ลำดับการประมวลผลข้อคำถามนี้จะมีสามขั้นตอนดังนี้

1) ทำการซีเลคท์ $\sigma_{A>a}(R)$ คือ การตรวจสอบและค้นหาคุณลักษณะ A ของ R ที่มีค่ามากกว่า a และเก็บผลลัพธ์ที่ได้จาก $\sigma_{A>a}(R)$ ไว้ใน R1

2) ทำการจอยน์ R1 $\bowtie_{R1.B = S.B}$ S โดยใช้วิธีการที่เรียกว่า Merge join และเก็บผลลัพธ์ไว้ที่ R2

3) ทำการจอยน์ R2 $\bowtie_{R2.C = T.C}$ T โดยใช้วิธีการที่เรียกว่า Nested loop

สำหรับข้อคำถามที่กำหนดให้นี้สามารถที่จะมีแผนการประมวลผลต่างจากนี้ได้แต่ผลลัพธ์ที่ได้จะเหมือนกัน จากตัวอย่างสามารถกำหนดแผนการประมวลผลที่แตกต่างกันได้โดยทำการเปลี่ยนแปลงดังนี้

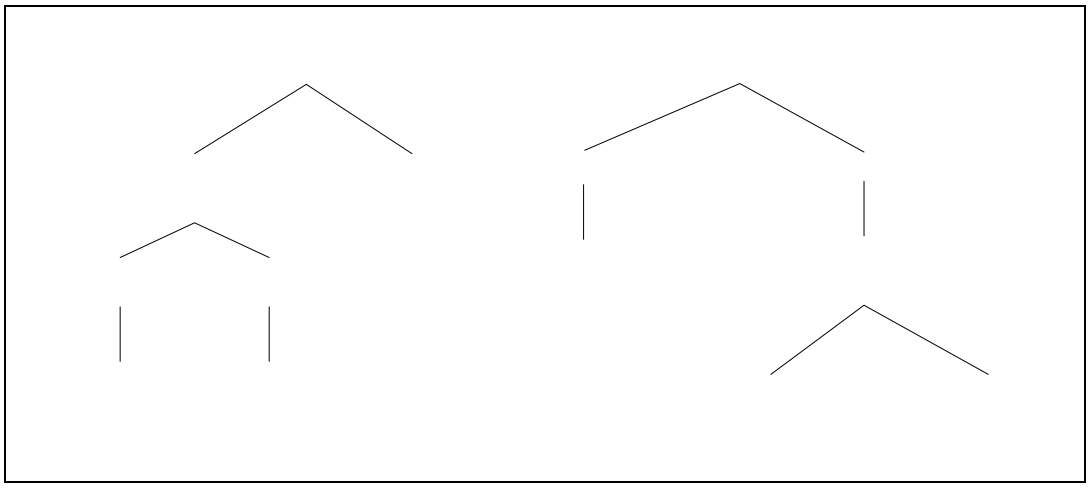
1) ทำการซีเลคท์ $\sigma_{A>a}(R)$ คือ การตรวจสอบและค้นหาคุณลักษณะ A ของ R ที่มีค่ามากกว่า a และเก็บผลลัพธ์ที่ได้จาก $\sigma_{A>a}(R)$ ไว้ใน R1

2) ทำการจอยน์ S $\bowtie_{S.C = T.C}$ T โดยใช้วิธีการที่เรียกว่า Nested loop แล้วเก็บผลลัพธ์ที่ได้ไว้ที่ R3

3) ทำการจอยน์ R1 $\bowtie_{R1.B = R3.B}$ R3 โดยใช้วิธีการที่เรียกว่า Merge join

จะเห็นว่าแผนการประมวลผลที่ต่างกันนี้สามารถให้ผลลัพธ์ที่เหมือนกันได้ หรือสามารถบอกได้ว่าทั้งสองวิธีนี้มีความเท่าเทียมกัน แต่อย่างไรก็ตามความแตกต่างของแผนการประมวลผลนี้จะอยู่ที่ลำดับขั้นตอนการทำงานดังที่แสดงเปรียบเทียบในรูปที่ 2.5 ซึ่งจะส่งผลให้ค่าใช้จ่ายในการประมวลผลหรือเวลาที่ใช้ในการประมวลผลต่างกัน ดังนั้นจึงควรเลือกแผนการประมวลผลที่มีค่าใช้จ่ายน้อยที่สุด

ในระบบฐานข้อมูลค่าใช้จ่ายของการประมวลผลข้อคำถามจะรวมไปถึงค่าใช้จ่ายที่เกิดจากหน่วยนำเข้าและส่งออก (I/O) และ ค่าใช้จ่ายที่เกิดจากหน่วยประมวลผลกลาง (CPU) ด้วย



รูปที่ 2.5 เปรียบเทียบลำดับการประมวลผลข้อคำถามด้วยแผนที่ต่างกัน (ก)

2.2.2 การวิเคราะห์ค่าใช้จ่าย (Cost analysis)

ในหัวข้อนี้จะกล่าวถึงเทคนิคการประมวลผลของโอเปอเรชันความสัมพันธ์โดยแยกวิเคราะห์ประกอบด้วย Selection, Projection และ Join พร้อมทั้งวิเคราะห์ค่าใช้จ่ายหรือวิธีที่ใช้ในแต่ละวิธีการ

ค่าใช้จ่ายของการทำ Selection

ค่าใช้จ่ายที่เกิดจากการใช้โอเปอเรชัน Selection จะขึ้นอยู่กับจำนวนของทูเพิล (Tuple) การวิเคราะห์จะวิเคราะห์ภายใต้ข้อกำหนด "A op a on R" เขียนในรูปแบบสัญลักษณ์ได้ดังนี้

$\sigma_{A\ op\ a}(R)$

เมื่อ A คือ คุณลักษณะหรือแอททริบิวต์ (Attribute)

op คือ โอเปอเรชันในการเปรียบเทียบได้แก่ $=, \neq, <, \leq, >, \geq$

a คือ ค่าคงที่

R คือ ความสัมพันธ์ที่มีจำนวน n ทูเพิล และในจำนวนนี้มี k ทูเพิลที่ตรงตามเงื่อนไข A op a

กรณีที่ 1 ไม่ใช้เส้นทางการเข้าถึงแบบรวดเร็ว (Fast access path not available or not use) กรณีนี้สามารถแบ่งย่อยได้เป็นสองกรณี คือ

1. กรณีที่ข้อมูลมีการจัดเรียงแล้ว ในกรณีนี้สามารถใช้การค้นหาแบบทวิภาค (Binary search) ได้ และค่าใช้จ่ายที่เกิดขึ้นสามารถหาได้ดังนี้ ค่าใช้จ่ายที่เกิดจาก CPU คือ O(log n + k) และค่าใช้จ่ายจาก I/O คือ $O(\log N + \lceil (k/n) \cdot N \rceil)$ เมื่อ N คือ จำนวนของหน้า (Page) หรือส่วนของหน่วยความจำที่บรรจุทูเพิลของ R และ $\lceil (k/n) \cdot N \rceil$ คือ จำนวนของหน้าในหน่วยความจำที่เก็บบรรจุทูเพิลไว้ k ทูเพิล ตามเงื่อนไขที่ได้กำหนดไว้

2. กรณีที่ข้อมูลยังไม่ได้จัดเรียง ในกรณีนี้จะใช้ การอ่านแบบลำดับ (Sequential scan) และค่าใช้จ่ายที่เกิดจาก CPU คือ $O(n)$ และค่าใช้จ่ายที่เกิดจาก I/O คือ $O(N)$

กรณีที่ 2 ใช้เส้นทางการเข้าถึงแบบรวดเร็ว (Fast access path is use) กรณีนี้สามารถแบ่งได้เป็นสองกรณี คือ

1. กรณีที่ข้อมูลมีการจัดเรียงแล้ว กรณีนี้จะเกิดขึ้นเมื่อมีการใช้ดรรชนีร่วม (clustered index) ซึ่งเป็นเส้นทางการเข้าถึงแบบรวดเร็ว ค่าใช้จ่ายที่เกิดจาก CPU คือ $O(k)$ และค่าใช้จ่ายที่เกิดจาก I/O คือ $O(\lceil (k/n) \cdot N \rceil)$

2. กรณีที่ข้อมูลยังไม่ได้จัดเรียง ค่าใช้จ่ายที่เกิดขึ้นสามารถหาได้ดังนี้ ค่าใช้จ่ายที่เกิดจาก CPU คือ $O(k)$ และค่าใช้จ่ายที่เกิดจาก I/O คือ $O(\min\{k, N\})$

ค่าใช้จ่ายของการทำ Projection

ค่าใช้จ่ายที่ประเมินจาก Projection สมมุติให้มีการใช้คำสั่ง $\prod_{A1,\ldots,At}(R)$ เมื่อ A1,…,At คือ คุณลักษณะหรือแอททริบิวต์ (attribute) ของความสัมพันธ์ R เป็นไปได้สองกรณีคือ

กรณีที่ 1 จำลองแถวแบบไม่เอาออก ในกรณีนี้สามารถประมวลผลได้โดยการอ่านทีละทูเพิลเพราะฉะนั้นจะได้ค่าใช้จ่ายที่เกิดจาก CPU เป็น O (n) และ จาก I/O เป็น O (N)

กรณีที่ 2 จำลองแถวแบบเอาออก ในเอสคิวแอลการจำลองแล้วเอาออกมีอยู่กรณีเดียวคือ เมื่อใช้คำสั่ง Select distinct จะมีการทำงานเป็นสามขั้นตอน ขั้นแรกคือ การอ่าน ขั้นที่สองคือ การจัดเรียง และสุดท้ายคือ การเอาออก จะได้ค่าใช้จ่ายที่เกิดจาก CPU เป็น $O(n \log n)$ ซึ่งค่าใช้จ่ายในส่วนนี้จะถูกควบคุมโดยค่าใช้จ่ายที่เกิดจากการจัดเรียง และ ค่าใช้จ่ายจาก I/O คิดเป็น $O(W \cdot \log W)$ เมื่อ $W = n \cdot (\sum_{i=1}^{t} \text{length}(A_i)) / \text{PageSize}$

ค่าใช้จ่ายของการทำ Join

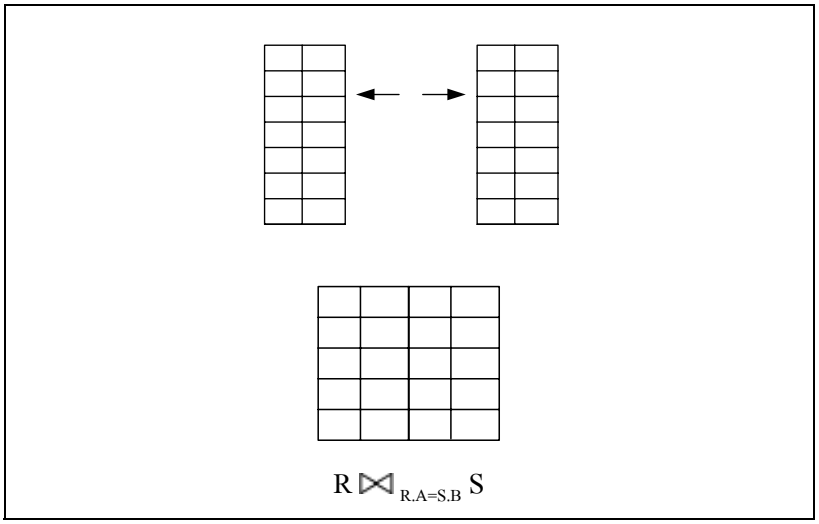
ค่าใช้จ่ายที่ประเมินจาก Join เป็นโอเปอเรชันที่มีค่าใช้จ่ายสูงกว่าทุกโอเปอเรชันที่ได้กล่าวมาแล้ว ซึ่งการหาค่าใช้จ่ายที่เกิดขึ้นสามารถหาได้จากวิธีดำเนินการ ในโอเปอเรชันนี้มีวิธีดำเนินการอยู่ 3 วิธีการคือ

1) Nested loop เป็นการดำเนินการ Join อย่างง่ายโดยทำไปตามข้อกำหนดของ Join operation โดยกำหนดให้ความสัมพันธ์ R เป็นความสัมพันธ์ภายนอก (Outer relation) มีจำนวน n ทูเพิลและความสัมพันธ์ S เป็นความสัมพันธ์ภายใน (Inner relation) มีจำนวน m ทูเพิลและทูเพิลทั้งหมดของความสัมพันธ์ภายในจะถูกเปรียบเทียบโดยทูเพิลจากความสัมพันธ์ภายนอก ดังตัวอย่างรูปที่ 2.6 และค่าใช้จ่ายที่เกิดจาก I/O จะขึ้นอยู่กับการลดหน้าหรือส่วนของหน่วยความจำ ส่วนค่าใช้จ่ายที่เกิดจาก CPU จะเป็น O (n · m)

```
for each tuple x in R
    for each tuple y in S
        if x [A] = y [B] then return ( x, y )
```

รูปที่ 2.6 อัลกอริทึม Nested loop join ด้วยเงื่อนไข R.A=S.B (Clement and Meng, 1997)

2) Sort merge ทำงานดังนี้ ขั้นที่หนึ่ง ข้อมูลในทุกความสัมพันธ์จะถูกจัดเรียง และในขั้นที่สอง ข้อมูลในความสัมพันธ์ทั้งสองจะถูกอ่านและถูกเปรียบเทียบ โดยเงื่อนไขในการ Join และเมื่อข้อมูลจากความสัมพันธ์แรก (R) ตรงกับข้อมูลในความสัมพันธ์ที่สอง (S) ตามเงื่อนไขก็จะส่งออกมาเป็นผลลัพธ์ ดังตัวอย่างรูปที่ 2.7 ค่าใช้จ่ายที่เกิดจาก I/O จะเป็น O (N + M) เมื่อ N คือจำนวน Page ของความสัมพันธ์ R และ M คือ จำนวน Page ของความสัมพันธ์ S



รูปที่ 2.7 ตัวอย่างการทำ Sort merge join ด้วยเงื่อนไข R.A=S.B (Clement and Meng, 1997)

3) Hash join พื้นฐานของอัลกอริทึมนี้จะประกอบด้วยขั้นตอนการทำงานสองขั้นตอนคือ ขั้นตอนแรก สร้าง Hash table สำหรับความสัมพันธ์ที่มีขนาดเล็กกว่า (กำหนดให้เป็น S) ดังตัวอย่างในรูปที่ 2.5 ภายใต้คุณลักษณะที่ต้องการเชื่อมต่อกัน ขั้นที่สอง ใช้ความสัมพันธ์ที่ใหญ่กว่า (กำหนดให้เป็น R) ตรวจสอบการเชื่อมต่อกันซึ่งมีรายละเอียดดังรูปที่ 2.8 และค่าใช้จ่ายที่เกิดจาก CPU คือ O (n + m· b) เมื่อ b คือ ค่าเฉลี่ยของจำนวนทูเพิลต่อบัคเก็ต (Bucket) ค่าใช้จ่ายที่เกิดจาก I/O คือ O (N + M)

```
for each tuple x
        in R
    {
 if the bucket is nonempty
            for every tuple y in the
                found bucket
             if x [A] = y [B] then return
                                ( x, y )
    }
```

รูปที่ 2.8 อัลกอริทึม Hash join ด้วยเงื่อนไข R.A=S.B (Clement and Meng, 1997)

2.3 การสร้างและการบำรุงรักษาวิวข้อมูล (Materialized view creation and maintenance)

2.3.1 วิว (View) และวิวข้อมูล (Materialized view)

วิว (View) คือ มุมมองเฉพาะหรือตารางจำลองที่สร้างขึ้นมาใหม่จากข้อคำถามที่ได้กำหนดไว้แล้ว โดยจะอ่านข้อมูลมาจากตารางจริงในฐานข้อมูลจากข้อคำถามที่กำหนด โดยวิวที่สร้างขึ้นมานั้นอาจจะอ่านข้อมูลมาจากตารางเดียวหรือหลาย ๆ ตารางรวมกันก็ได้ ซึ่งวิวนี้ไม่ได้เก็บข้อมูลอย่างตารางแต่ข้อมูลที่เห็นคือข้อมูลจากตารางหลักทั้งสิ้น วิวจะเก็บเพียงแต่โครงสร้างที่จะไปเรียกข้อมูลมาจากตารางเท่านั้น ตัวอย่างเช่น เมื่อต้องการสร้างวิวชื่อ INVASION ซึ่งเป็นวิวที่รวมข้อมูลจากตาราง WEATHER และ LOCATION โดยจะกำหนดเงื่อนไขในการสร้างดังรูปที่ 2.9

```
SELECT    WEATHER.City, Condition, Temperature, Latitude,
          NortSouth, Longitude, EastWest
FROM      WEATHER, LOCATION
WHERE     WEATHER.City = LOCATION.City;
```

รูปที่ 2.9 ตัวอย่างเงื่อนไขในการสร้างวิว

จากตัวอย่างในรูปที่ 2.9 จะเห็นว่าเป็นชุดคำสั่งในการจอยน์ (Join) ข้อมูลระหว่างตาราง WEATHER กับ LOCATION ซึ่งจะสามารถนำข้อกำหนดนี้ไปสร้างเป็นวิวได้ดังรูปที่ 2.10

```
CREATE    VIEW INVASION AS
SELECT    WEATHER.City, Condition, Temperature, Latitude,
          NortSouth, Longitude, EastWest
FROM      WEATHER, LOCATION
WHERE     WEATHER.City = LOCATION.City;
```

รูปที่ 2.10 ตัวอย่างการสร้างวิว

จะเห็นว่าคำสั่งในการสร้างวิวนั้นจะไม่เหมือนกับคำสั่งในการสร้างตาราง แต่อย่างไรก็ตามวิวที่ถูกสร้างขึ้นมานั้นจะสามารถถูกเรียกใช้ได้เหมือนกับตารางข้อมูลทั่วไป เช่น สามารถสอบถามโครงสร้างของวิวได้โดยใช้คำสั่ง Describe เหมือนกับการสอบถามโครงสร้างของตารางทั่วไป ดังแสดงในรูปที่ 2.11

```
DESCRIBE INVASION
Name                                      Null?            Type
----------------------------------------- ---------------- --------------------
CITY                                                       VACHAR2 (11)
CONDITION                                                  VACHAR2 (9)
TEMPERATURE                                                NUMBER
LATITUDE                                                   NUMBER
NORTHSOUTH                                                 CHAR (1)
LONGITUDE                                                  NUMBER
EASTWEST                                                   CHAR (1)
```

รูปที่ 2.11 ตัวอย่างการใช้คำสั่ง DESCRIBE กับวิว

การสอบถามข้อมูลจากวิวสามารถทำได้โดยใช้คำสั่ง Select เช่นเดียวกันกับที่ใช้ในการสอบถามข้อมูลจากตารางหลัก ดังตัวอย่างในรูปที่ 2.12

```
SELECT    City, Condition, Temperature, Latitude, NortSouth AS N,
          Longitude, EastWest AS E
FROM      INVASION;

CITY    CONDITION   TEMPERATURE  LATITUDE  N   LONGITUDE  E
-------------------------------------------------------------------------------
ATHENS  SUNNY       97           37.58     N   23.43      E
LIMA    RAIN        45           12.03     S   77.03      W
PARIS   CLOUDY      81           48.52     N   2.2        E
```

รูปที่ 2.12 ตัวอย่างการสอบถามข้อมูลจากวิว

จากตัวอย่างในรูปที่ 2.12 จะเห็นว่าผลลัพธ์ที่ได้นั้นจะเหมือนกับการสอบถามข้อมูลจากตารางหลักหรือตารางพื้นฐานโดยใช้เงื่อนไขในการสร้างวิวเป็นข้อคำถามซึ่งผลลัพธ์ที่ได้จากการสอบถามข้อมูลจากตารางพื้นฐานนั้นจะแสดงดังรูปที่ 2.13

```
SELECT  WEATHER.City,  Condition,  Temperature,  Latitude, NortSouth AS N,
                Longitude,  EastWest AS E
FROM            WEATHER,  LOCATION
WHERE  WEATHER.City  =  LOCATION.City;
```

CITY	CONDITION	TEMPERATURE	LATITUDE	N	LONGITUDE	E
ATHENS	SUNNY	97	37.58	N	23.43	E
LIMA	RAIN	45	12.03	S	77.03	W
PARIS	CLOUDY	81	48.52	N	2.2	E

รูปที่ 2.13 ตัวอย่างการสอบถามข้อมูลจากตารางพื้นฐาน

ประโยชน์ของการสร้างวิว คือ ช่วยทำให้เกิดความปลอดภัยของข้อมูลมากขึ้น โดยสามารถควบคุมสิทธิของผู้ใช้ในการมองเห็นข้อมูลจากตารางจริงได้ โดยสามารถกำหนดให้ผู้ใช้เห็นข้อมูลเฉพาะคอลัมน์ที่ต้องการให้เห็นเท่านั้น และวิวช่วยลดความซับซ้อนในการเขียนข้อคำถามเพื่อการอ่านข้อมูลในกรณีที่เรียกใช้ข้อคำถามนั้นบ่อย ๆ รวมทั้งช่วยให้การเรียกใช้คำสั่งเพื่ออ่านข้อมูลของผู้ใช้นั้นง่ายขึ้นเมื่อมีการอ่านข้อมูลจากวิวอีกด้วย

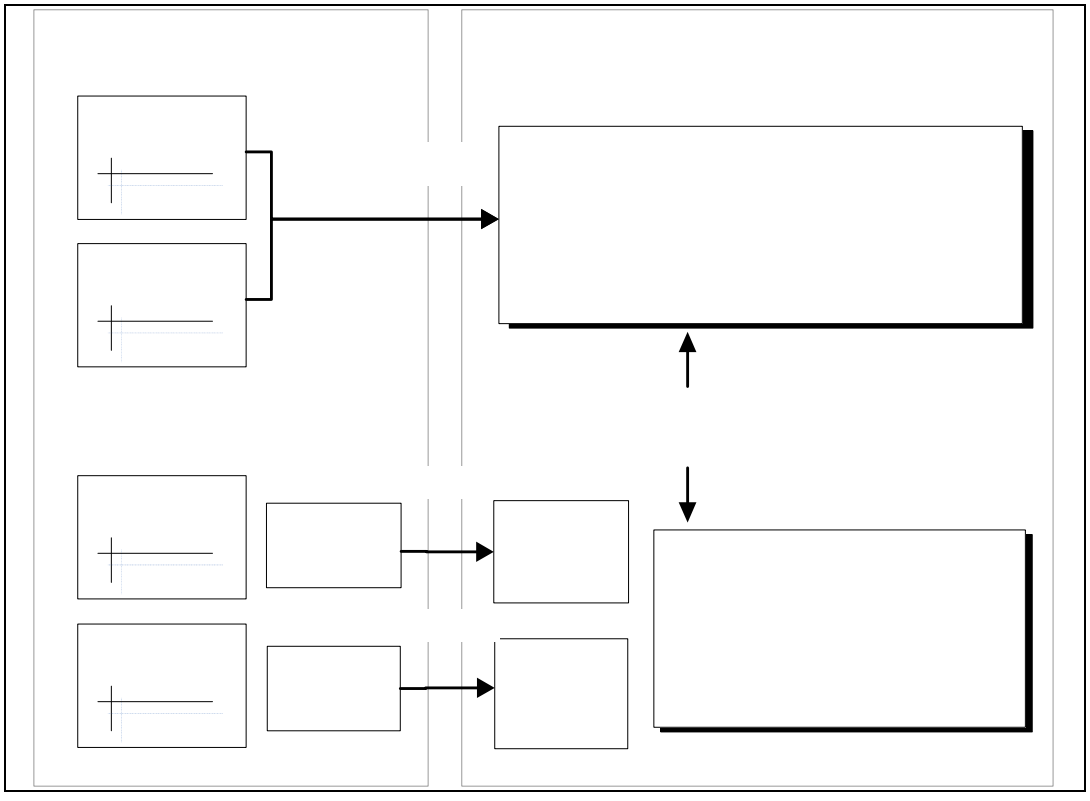
วิวข้อมูล (Materialized view) คือ ข้อมูลที่มีมุมมองเฉพาะหรือมีข้อมูลเฉพาะด้านที่ตรงกับความสนใจของผู้ใช้เหมือนกับวิวทั่วไปจะต่างกันที่วิวข้อมูลจะเก็บข้อมูลไว้จริง วิวข้อมูลสร้างขึ้นจากข้อมูลพื้นฐานที่ถูกเก็บในฐานข้อมูลตามเงื่อนไขและข้อกำหนดในการสร้างข้อมูล ประโยชน์หลักของวิวข้อมูลคือ ใช้ตอบคำถามเฉพาะเรื่องที่ผู้ใช้ส่วนใหญ่สนใจและมักสอบถามอยู่เสมอ วิวข้อมูลเป็นการนำข้อมูลจากข้อมูลพื้นฐานภายใต้เงื่อนไขที่กำหนดมาเก็บไว้เป็นการถาวรเปรียบเหมือนตารางข้อมูลทั่วไป จึงช่วยให้การตอบข้อคำถามทำได้อย่างรวดเร็วยิ่งขึ้น (Lane, 2003) การที่จะใช้วิวข้อมูลให้เกิดประสิทธิภาพขึ้นอยู่กับข้อมูลที่เก็บอยู่ในวิวข้อมูลว่าตรงตามความต้องการของผู้ใช้หรือไม่ โดยปกติแล้วการจะตอบข้อคำถามที่มีความซับซ้อน จะต้องใช้เวลาในการประมวลผลมาก เนื่องจากจะต้องไปค้นหาข้อมูลจากตารางต่าง ๆ ที่ตรงตามความต้องการของผู้ใช้ ปัญหานี้สามารถแก้ไขได้โดยการนำ

วิวข้อมูลมาเก็บข้อมูลชุดนั้นและเมื่อมีการสอบถามเกี่ยวกับข้อมูลชุดดังกล่าวก็สามารถสอบถามกับวิวข้อมูลได้ทันทีโดยไม่ต้องเข้าไปที่ตารางข้อมูลพื้นฐานและข้อมูลที่ได้มานั้นจะเหมือนกับการสอบถามจากตารางข้อมูลพื้นฐานโดยตรงซึ่งจะเป็นการลดภาระการทำงานลงได้และเป็นการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม

โดยทั่วไปแล้วรูปแบบของการนำวิวข้อมูลมาใช้จะมีอยู่สองรูปแบบ คือ วิวข้อมูลแบบซับซ้อน และวิวข้อมูลแบบง่าย ดังตัวอย่างรูปที่ 2.14

1) วิวข้อมูลแบบซับซ้อน (Complex materialized view : Method A) วิวข้อมูลแบบนี้จะมีประสิทธิภาพในการตอบคำถามมากเนื่องจากมีการจอยน์ (Join) ข้อมูลมาก่อนแล้วในขั้นตอนการสร้างวิวข้อมูล ดังนั้นเวลาตอบคำถามจึงจะไม่มีการจอยน์เกิดขึ้น แต่การฟื้นฟู (Refresh) ข้อมูลจะช้า ดังนั้นวิวข้อมูลในรูปแบบนี้จึงเหมาะกับข้อมูลไม่ค่อยมีการเปลี่ยนแปลง

2) วิวข้อมูลแบบง่าย (Simple materialized view : Method B) วิวข้อมูลแบบนี้จะมีประสิทธิภาพในการตอบคำถามไม่ค่อยดีนักเมื่อเทียบกับวิวข้อมูลแบบซับซ้อนเพราะจะมีการจอยน์เกิดขึ้น แต่อย่างไรก็ตามวิวข้อมูลแบบนี้จะฟื้นฟูข้อมูลได้เร็วกว่า ดังนั้นวิวข้อมูลในรูปแบบนี้จึงเหมาะกับข้อมูลที่มีการเปลี่ยนแปลงบ่อย ๆ



รูปที่ 2.14 วิวข้อมูลแบบง่ายและแบบซับซ้อน (Urbano, 2003)

2.3.2 การสร้างวิวข้อมูล (Materialized view creation)

วิวข้อมูลถูกกำหนดให้สามารถใช้งานฟังก์ชันในการรวม (Aggregate functions) เป็นกลุ่มฟังก์ชันที่ใช้กับข้อมูลที่เป็นตัวเลข ได้แก่ COUNT SUM AVG MAX และ MIN ทั้งที่เป็นข้อมูลตารางเดียว หรือ ตารางจากการจอยน์ (Join) ก็ตาม (Gupta, Harinarayan, and Quass, 1995) และวิวข้อมูลยังสามารถใช้ภาษานิยามข้อมูล (Data definition language : DDL) ได้เหมือนตารางทั่วไป อย่างเช่น CREATE ALTER และ DROP เป็นต้น

การสร้างวิวข้อมูลนั้นสามารถทำได้โดยใช้คำสั่ง CREATE MATERIALIZED VIEW ซึ่งจะมีรูปแบบคำสั่งดังรูปที่ 2.15

```
create materialized view [ user. ] name                                    1
[ organization index iot_clause ]                                          2
[  { { segment attribute clauses }                                         3
  | cluster cluster ( column [ , column ] … ) }                            4
  [ { partitioning clause | parallel clause | build clause } ]             5
  | on prebuilt table [ { with | without } reduced precision ] ]           6
[ using index                                                              7
  [ { physical attribute clauses | tablespace clause }                     8
    [ physical attribute clauses | tablespace clause ]                     9
    | using on index ]                                                     10
[ refresh clause ]                                                         11
[ for update ] [ { disable | enable } query rewrite ]                      12
as subquery;                                                               13
```

รูปที่ 2.15 รูปแบบการใช้คำสั่ง create materialized view (Hobbs, 2005)

รูปแบบคำสั่งในการสร้างวิวข้อมูลจะประกอบไปด้วย 4 ส่วนหลักดังนี้

ส่วนที่ 1 คือ ส่วนหัว (Header) ส่วนนี้จะระบุชื่อของวิวข้อมูล โดยจะปรากฏอยู่บรรทัดแรกของวิวข้อมูลจะถูกสร้างขึ้นมาในบัญชีผู้ใช้ (User account) หรือ สคีมา (Schema) ที่กำหนดไว้

ส่วนที่ 2 คือ ส่วนระบุพารามิเตอร์ในการเก็บข้อมูล (Storage parameter) ปรากฏอยู่ในบรรทัดที่ 2 ถึง บรรทัดที่ 10 พารามิเตอร์ในการเก็บจะเป็นตัวกำหนดลักษณะของวิวข้อมูล เช่น การกำหนดคอลัมน์ เป็นต้น

ส่วนที่ 3 คือ ส่วนที่ระบุชนิดการฟื้นฟูข้อมูล (Refresh) ของวิวข้อมูล โดยจะปรากฏอยู่ในบรรทัดที่ 11 รายละเอียดในส่วนนี้จะกล่าวถึงในหัวข้อ 2.3.3

ส่วนที่ 4 คือ ส่วนที่ระบุข้อคำถามซึ่งจะเป็นข้อกำหนดในการดึงข้อมูลมาสร้างเป็นวิวข้อมูลจะปรากฏอยู่ในบรรทัดที่ 12 ถึง บรรทัดที่ 13 ถ้านำข้อคำถามนี้ไปสอบถามข้อมูลจากตารางพื้นฐานก็จะได้ข้อมูลออกมาเท่ากับข้อมูลที่ถูกเก็บไว้ในวิวข้อมูล

ตัวอย่างในรูปที่ 2.16 เป็นตัวอย่างคำสั่งที่ใช้ในการสร้างวิวข้อมูลแบบอ่านอย่างเดียว (Read-only) ชื่อ LOCAL_CATEGORY_COUNT วิวข้อมูลนี้จะถูกสร้างขึ้นภายใต้ Tablespace ที่ชื่อ USERS กำหนดการฟื้นฟูข้อมูลเป็นแบบ Force และกำหนดให้มีการฟื้นฟูข้อมูลทุก ๆ เจ็ดวัน วิวข้อมูลนี้จะถูกสร้างขึ้นโดยนำข้อมูลมาจากตารางพื้นฐานชื่อ BOOKSHELF ตามเงื่อนไขที่ถูกกำหนด

```
create materialized view LOCAL_CATEGORY_COUNT
tablespace USERS
refresh force
start with SysDate next SysDate+7
with primary key
as
select CategoryName, count(*) CountPerCat
from BOOKSHELF@REMOTE_CONNECT
        group by CategoryName;
```

รูปที่ 2.16 ตัวอย่างการสร้างวิวข้อมูล

2.3.3 การบำรุงรักษาและการฟื้นฟูข้อมูลของวิวข้อมูล (Maintenance and refreshes of materialized view)

ปัญหาที่พบอย่างหนึ่งในการใช้ข้อมูลจากวิวข้อมูล คือ การถ่ายโอนข้อมูล และการปรับปรุงข้อมูลให้ใหม่อยู่เสมอ (Gupta, and Mumick, 1998) เป็นการจัดเตรียมการเพื่อกลไกต่อไปนี้

1) การฟื้นฟูปรับปรุงข้อมูลทั้งหมดให้ใหม่อยู่เสมอ (Fully refresh the data)
2) กระบวนการฟื้นฟูแบบเร็ว เช่น การเพิ่ม หรือ รวม ข้อมูล (Perform a fast refresh)
3) การปรับปรุงวิวข้อมูลอัตโนมัติทุกครั้งเมื่อมีการเปลี่ยนแปลง (Automatic update)

การฟื้นฟูข้อมูลของวิวข้อมูลนั้นสามารทำได้ 2 แบบดังนี้

1) การฟื้นฟูแบบอัตโนมัติ (Automatic refreshes)

พิจารณาจากชุดคำสั่งที่ใช้สร้างวิวข้อมูลที่ชื่อ LOCAL_BOOKSHELF จะเห็นว่ามีการกำหนดให้ฟื้นฟูวิวข้อมูลโดยอัตโนมัติ อยู่ในชุดคำสั่งนี้ด้วย ดังรูปที่ 2.17

```
Create materialized view LOCAL_BOOKSHELF tablespace USER
Refresh force
Start with SysDate next SysDate+7
with primary key
as
select * from BOOKSHELF@REMOTE_CONNECT;
```

รูปที่ 2.17 การกำหนดการฟื้นฟูวิวข้อมูลโดยอัตโนมัติ

คำสั่งที่ใช้ในการฟื้นฟูข้อมูลของวิวข้อมูลนั้นจะแบ่งออกเป็น 3 ส่วน คือ

- ส่วนแรก จะเป็นการกำหนดชนิดของการทำงานในการฟื้นฟูข้อมูลโดยจะระบุเป็น Fast, Complete, Never หรือ Force ซึ่งในแต่ละชนิดจะมีความแตกต่างกันดังนี้ Fast refresh เป็นการฟื้นฟูข้อมูลแบบเร็วโดยจะทำงานร่วมกับ Materialized view log โดย Materialized view log จะเป็นตารางที่เก็บข้อมูลของการเปลี่ยนแปลงข้อมูลในตารางพื้นฐานและจะนำข้อมูลนั้นไปปรับปรุงข้อมูลในวิวข้อมูล เมื่อมีข้อมูลเปลี่ยนแปลงในตารางพื้นฐานแม้เพียงแค่รายการเดียววิวข้อมูลก็จะถูกปรับปรุงข้อมูลไปด้วย Complete refresh การฟื้นฟูข้อมูลแบบนี้จะเป็นการลบข้อมูล

เก่าทั้งหมดที่มีในวิวข้อมูลและทำการคัดลอกข้อมูลจากตารางพื้นฐานมาใหม่ทั้งหมดตามเงื่อนไขที่ถูกกำหนดไว้ในขั้นตอนการสร้างวิวข้อมูล และ Force refresh คำสั่งนี้จะเป็นการบอกกับระบบเมื่อมีการฟื้นฟูข้อมูลให้ทำการฟื้นฟูข้อมูลแบบ Fast refresh ถ้าการฟื้นฟูข้อมูลในรูปแบบนี้ไม่สามารถทำงานได้ก็ให้ทำการฟื้นฟูข้อมูลในแบบ Complete refresh

- ส่วนที่สอง เป็นส่วนที่เริ่มคำสั่งด้วย Start with เป็นส่วนที่บอกฐานข้อมูลว่าเมื่อไหร่จะเริ่มทำงาน และเมื่อมีการกำหนดคำสั่งนี้จะต้องมีการกำหนดค่าในส่วนคำสั่ง Next ด้วย

- ส่วนที่สาม คำสั่ง Next เป็นส่วนที่บอกว่าอีกนานเท่าไหร่ถึงจะทำอีกครั้ง จากตัวอย่างนี้จะเห็นว่าใช้ Start with SysDate เป็นการเริ่มทำงาน ตามด้วย Next SysDate+7 ซึ่งจะหมายความว่าจะปรับปรุงทุกๆ 7 วัน

2) การทำด้วยมือ (Manual refreshes)

เราสามารถใช้คำสั่งเพื่อให้วิวข้อมูลมีการฟื้นฟูข้อมูลโดยจะมีรูปแบบของคำสั่งอยู่สองแบบ คือ การสั่งทีละวิวข้อมูลใช้ คำสั่ง DBMS.REFRESH ซึ่งจะมีพารามิเตอร์สองตัวคือ ตัวแรกจะเป็นชื่อของวิวข้อมูล ตัวที่สองจะเป็น ชนิดของการฟื้นฟูโดยจะระบุเป็น ‘f’ คือ fast ‘c’ คือ Complete และ ‘?’ คือ Force ตัวอย่างดังรูปที่ 2.18 เป็นการระบุ Complete refresh

```
execute DBMS.REFRESH(‘local bookshelf’,’c’)
```

รูปที่ 2.18 การกำหนดการฟื้นฟูวิวข้อมูลแบบทำด้วยมือ

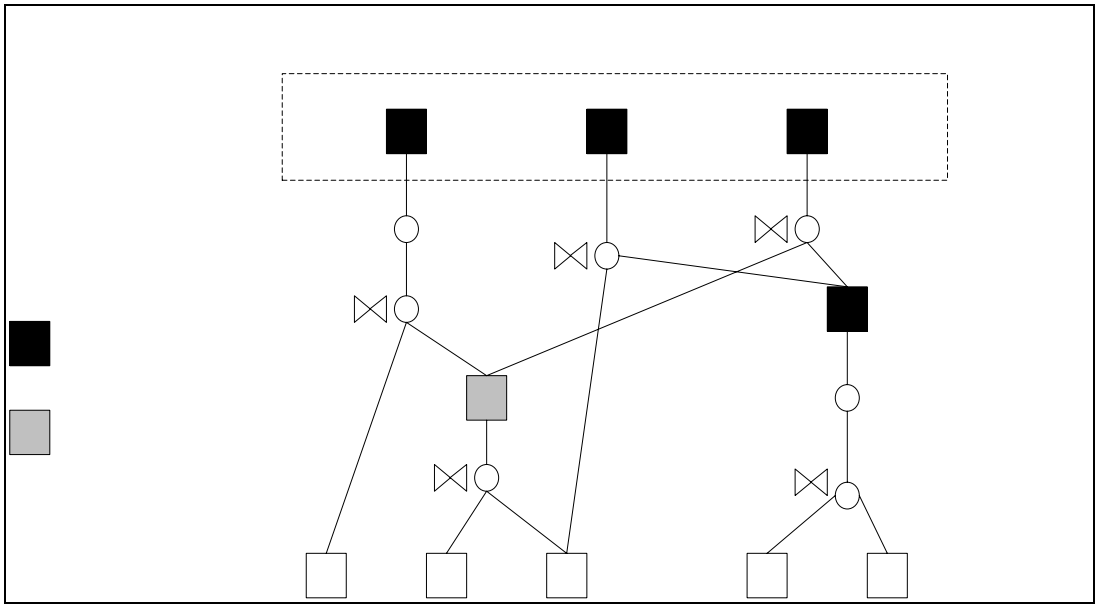
ถ้าต้องการใช้คำสั่งนี้กำหนดการฟื้นฟูทีละหลายวิวข้อมูลก็สามารถทำได้ดังรูปที่ 2.19

```
excute DBMS_MVIEW.REFRESH(‘local_bookshelf, local_category_count’,’?c’);
```

รูปที่ 2.19 การกำหนดการฟื้นฟูวิวข้อมูลแบบหลายวิว

จากคำสั่งในรูปที่ 2.19 จะเป็นการทำการฟื้นฟูสองวิวข้อมูลพร้อมกัน โดยวิวข้อมูลแรกใช้เป็น Force และวิวข้อมูลที่สองใช้ complete นอกจากนี้เรายังสามารถทำการฟื้นฟูวิวข้อมูลทั้งหมดพร้อมกันได้โดยใช้คำสั่ง REFRESH_ALL_MVIEWS

Mistry, Parsan and Ramamritham (2001) ได้แสดงการคัดเลือกและการบำรุงรักษาวิวข้อมูลเพื่อนำมาใช้ในการเพิ่มประสิทธิการประมวลผลข้อคำถาม ในส่วนของการคัดเลือก วิวข้อมูลได้กล่าวถึงรายละเอียดการคัดเลือกวิวข้อมูลร่วมกับการบำรุงรักษาวิวข้อมูลอย่างเหมาะสม กระบวนการนี้ใช้อัลกอริทึมกรีดี (Greedy algorithm) มาประยุกต์ใช้ในการคัดเลือกวิวข้อมูล และได้นำเสนอการบำรุงรักษาวิวข้อมูลดังตัวอย่างรูปที่ 2.20 และพวกเขาได้สรุปว่า เทคนิคของพวกเขาสามารถเพิ่มความเร็วในการบำรุงรักษาวิวข้อมูลได้ดี พร้อมทั้งสามารถใช้วิวข้อมูลที่ได้มาตอบข้อคำถามได้อย่างมีประสิทธิภาพและเวลาที่สูญเสียไปในกระบวนการเพิ่มประสิทธิภาพนั้นอยู่ในขั้นที่สามารถยอมรับได้

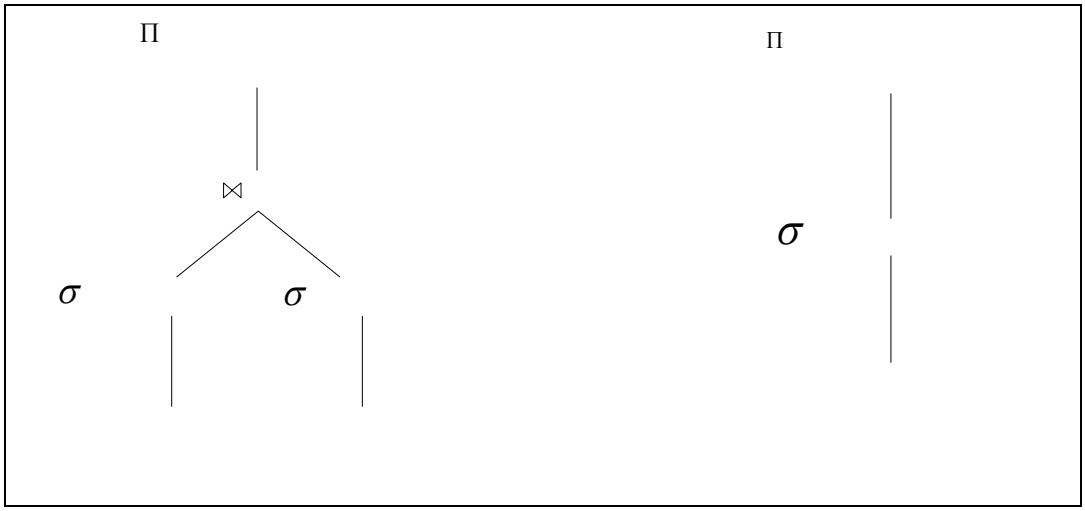


รูปที่ 2.20 ตัวอย่างแผนการบำรุงรักษาวิวข้อมูล

2.4 การเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล (Query optimization with materialized views)

การเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลเป็นการนำวิวข้อมูลเข้ามาใช้เพื่อช่วยตอบข้อคำถามโดยพยายามที่จะลดระยะเวลาที่ใช้ในการข้าถึงข้อมูล จากหัวข้อที่ผ่านมาได้กล่าวว่าวิวข้อมูลคือ การจำลองข้อมูลขึ้นมาใหม่จากกลุ่มข้อมูลที่เราสนใจ ดังนั้นจึงได้มีการนำวิวข้อมูล เข้ามาใช้โดยการนำข้อมูลที่ถูกสอบถามหรืออ้างถึงบ่อย ๆ มาสร้างเป็นวิวข้อมูลเอาไว้เพื่อตอบข้อคำถามเหล่านั้นแทนที่จะไปเรียกดูจากตารางข้อมูลพื้นฐานซึ่งจะเป็นการลดค่าใช้จ่ายได้

ดังตัวอย่างรูปที่ 2.21 เป็นการเปรียบเทียบเส้นทางในการเข้าถึงข้อมูลโดยการเรียกใช้ข้อมูลจากตารางข้อมูลพื้นฐานกับการเรียกใช้ข้อมูลผ่านทางวิวข้อมูล



รูปที่ 2.21 ตัวอย่างเส้นทางในการเข้าถึงข้อมูล โดยใช้ (ก) ตารางข้อมูลพื้นฐาน (ข) วิวข้อมูล

นักวิจัยได้พยายามมาเป็นระยะเวลานานที่จะนำวิวข้อมูลเข้ามาใช้ประโยชน์ในกระบวนการประมวลผลข้อคำถามโดยได้เสนอข้อคิดเห็นออกมามากมาย (Chang and Lee, 1998; Goldstein and Larson, 2001; Zhengxin, 2001) และมีหัวข้อที่ถูกหยิบยกขึ้นมาบ่อย ๆ อย่างเช่น คำจำกัดความ ข้อกำหนดของวิว องค์ประกอบของวิว และการบำรุงรักษาวิว แต่ในยุคนั้นยังไม่มีใครให้ความสำคัญเกี่ยวกับการนำวิวข้อมูลมาช่วยในการตอบคำถาม

Chaudhuri, Krishnamurthy, Potamianos, and Shim (1995) ได้เสนอแนวคิดของการนำวิวข้อมูล (Materialized views) มาช่วยในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม โดยได้นำเสนอในรูปของกฎที่เรียกว่า One-level rule ดังนี้

$$L(x, y) \longrightarrow V(x)$$

V คือ วิวข้อมูล

x คือ แอททริบิวต์ที่ต้องการนำไปสร้างวิวข้อมูล

y คือ ข้อกำหนด หรือ เงื่อนไข

C_ID C_NAME C_PHONE
PROVINCE_ID
C_NAME='Jirawan' PROVINCE_NAME=
SIM
CUSTOMER PROVINCE
(ก)

และได้แสดงตัวอย่างโดยกำหนดสคีมาของข้อมูลพื้นฐานให้ดังนี้

Emp(name, dno, sal, age)

Dep(dno, size, loc)

จากข้อมูลพื้นฐานที่กำหนดให้นำมาสร้างวิวข้อมูลชื่อ Executive ตาม One-level rule ได้ดังนี้

Emp(name, dno, sal, age), sal > 200k ⟶ Executive (name, dno, sal)

ถ้าต้องการสอบถามว่า ใครบ้างที่ทำงานในแผนกที่มีขนาดใหญ่กว่า 30 และมีรายได้มากกว่า 200 k จะสามารถระบุข้อคำถามในรูปแบบของ first-order logic ได้ดังนี้

Q(name) : - Emp(name, dno, sal, age), sal > 200k,
Dep(dno, size, loc), size > 30

แต่จากวิวข้อมูลที่มีสามารถเปลี่ยนแปลงข้อคำถามให้ซับซ้อนน้อยลงแต่ยังได้ผลลัพธ์เหมือนเดิมได้ดังนี้

Q(name) : - Executive(name, dno, sal),
Dep(dno, size, loc), size > 30

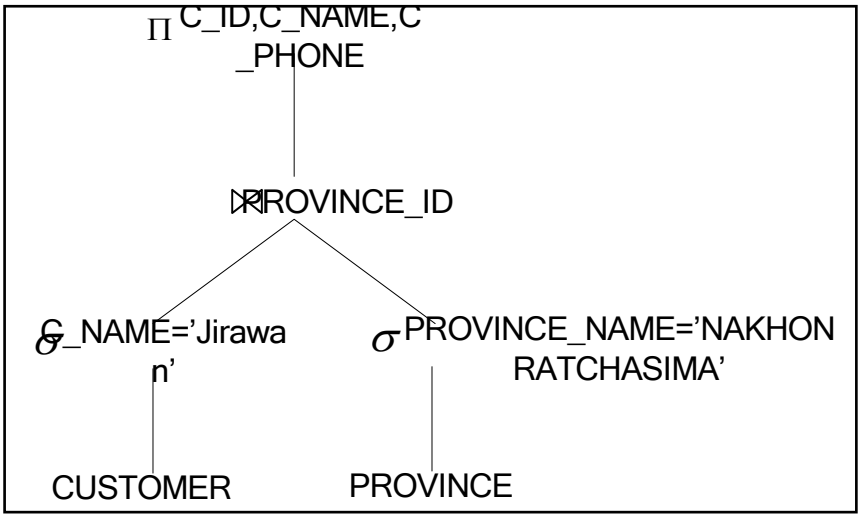
การเข้าถึงข้อมูลในวิวข้อมูล จะเหมือนกับการเข้าถึงข้อมูลจากตารางข้อมูลพื้นฐานทั่วไป การใช้วิวข้อมูลให้เกิดประสิทธิภาพนั้นขึ้นอยู่กับข้อมูลที่เก็บในวิวข้อมูลว่าตรงตามต้องการของผู้ใช้หรือไม่ การตอบข้อคำถามที่มีความซับซ้อนหรือข้อคำถามที่ต้องการข้อมูลจากหลายตารางนั้นต้องใช้เวลาในการประมวลผลมาก เพราะต้องค้นหาข้อมูลจากตารางต่าง ๆ ที่ตรงตามต้องการของผู้ใช้ ปัญหานี้แก้ไขได้โดยการนำวิวข้อมูลมาเก็บข้อมูลชุดนั้น และเมื่อมีการสอบถามเกี่ยวกับข้อมูลชุดดังกล่าวก็สามารถสอบถามกับวิวข้อมูลได้ทันทีโดยไม่ต้องเข้าไปที่ตารางข้อมูลพื้นฐาน และข้อมูลที่ได้มานั้นจะเหมือนกับการสอบถามจากตารางข้อมูลพื้นฐานโดยตรง ซึ่งลดภาระการทำงานลงได้ และเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม ตัวอย่างต่อไปนี้แสดงการใช้ Materialized view เพื่อลดภาระการทำงานของ Query processor

<u>ตัวอย่างที่ 1.</u> ตัวอย่างนี้เป็นการสร้างวิวข้อมูล (Materialized view) ที่เก็บรหัสลูกค้า ชื่อ ที่อยู่ หมายเลขโทรศัพท์ และ เพศ โดยจะเก็บข้อมูลเฉพาะลูกค้าที่อยู่จังหวัดนครราชสีมา (Nakhon Ratchasima) เท่านั้น ตัวอย่างนี้จะเป็นการสร้างวิวข้อมูลใหม่ด้วยข้อมูลสองตาราง คือ PROVINCE และ CUSTOMER โดยใช้ชุดคำสั่งดังนี้

```
CREATE MATERIALIZED VIEW  "C_KORAT_MV"
AS select       "CUSTOMER"."C_ID" as "C_ID",
                "CUSTOMER"."C_NAME" as "C_NAME",
                "CUSTOMER"."C_ADDRESS" as "C_ADDRESS",
                "CUSTOMER"."C_PHONE" as "C_PHONE",
                "CUSTOMER"."C_GENDER" as "C_GENDER"
from            "PROVINCE" "PROVINCE",
                "CUSTOMER" "CUSTOMER"
where           "PROVINCE"."PROVINCE_ID"="CUSTOMER"."PROVINCE_ID"
and             "PROVINCE"."PROVINCE_NAME" ='NAKHONRATCHASIMA'
```

หลังจากการประมวลผลชุดคำสั่งข้างบนนี้ จะได้วิวข้อมูลที่เก็บข้อมูลของลูกค้าซึ่งเป็นการเลือกข้อมูลบางส่วนมาจากตาราง Customer ทำให้วิวข้อมูลที่ได้เป็นชุดของข้อมูลที่มีขนาดเล็กกว่าชุดข้อมูลชุดเดิม ถ้ามีข้อคำถามเกี่ยวกับลูกค้าที่มีที่อยู่ ที่จังหวัดนครราชสีมาว่า ต้องการทราบหมายเลขโทรศัพท์ของลูกค้าที่ชื่อ Jirawan จะสามารถสอบถามได้ด้วยชุดคำสั่ง Query 1A และแสดงแผนงานการประมวลผลข้อคำถามได้ดังรูปที่ 2.22

```
Query1A :
SELECT          "CUSTOMER"."C_ID" as "C_ID",
                "CUSTOMER"."C_NAME" as "C_NAME",
                "CUSTOMER"."C_PHONE" as "C_PHONE"
FROM            "PROVINCE" "PROVINCE",
                 "CUSTOMER" "CUSTOMER"
WHERE            "CUSTOMER"."PROVINCE_ID"="PROVINCE"."PROVINCE_ID"
AND              "CUSTOMER"."C_NAME" ='Jirawan'
AND             "PROVINCE"."PROVINCE_NAME" ='NAKHONRATCHASIMA'
```



รูปที่ 2.22 แผนการประมวลผลข้อคำถาม Query 1A

หลังจากการประมวลผลชุดคำสั่งด้วย Query 1A จะได้ผลลัพธ์ดังนี้

C_ID	C_NAME	C_PHONE
7981	Jirawan	849849000

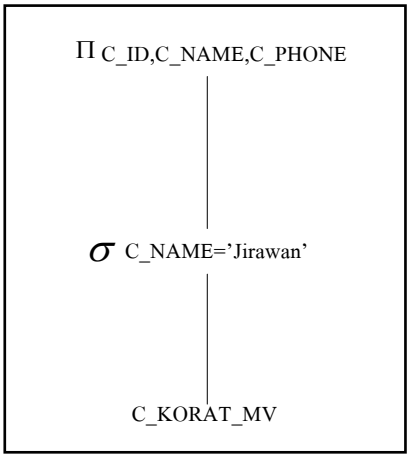
Execution time (3 times) 31 msecs., 32 msecs., 30 msecs.

รูปที่ 2.23 ผลจากการประมวลผลชุดคำสั่ง Query1A

จากวิวข้อมูล C_KORAT_MV ทำให้สามารถแปลงข้อคำถาม Query1A ได้เป็น Query 1B ที่มีแผนการประมวลผลดังรูปที่ 2.24

Query1B :

```
select    "C_KORAT_MV"."C_ID" as "C_ID",
          "C_KORAT_MV"."C_NAME" as "C_NAME",
          "C_KORAT_MV"."C_PHONE" as "C_PHONE"
from      "C_KORAT_MV" "C_KORAT_MV"
where     "C_KORAT_MV"."C_NAME" ='Jirawan'
```



รูปที่ 2.24 แผนการประมวลผลข้อคำถาม Query 1B

หลังจากการประมวลผลชุดคำสั่งด้วย Query 1B จะได้ผลลัพธ์ดังนี้

C_ID	C_NAME	C_PHONE
7981	Jirawan	849849000

Execution time (3 times) 16 msecs., 16 msecs., 15 msecs.

รูปที่ 2.25 ผลจากการประมวลผลชุดคำสั่ง Query1B

เมื่อเปรียบเทียบผลของการประมวลผลชุดคำสั่งทั้งสองจะเห็นว่าข้อมูลที่ได้ออกมานั้นเหมือนกัน แต่เวลาที่ใช้ประมวลผลนั้นต่างกัน โดยจะเห็นว่าเวลาที่ใช้ในการประมวลผลชุดคำสั่ง Query1B นั้นใช้เวลาน้อยกว่า Query1A

ตัวอย่างที่ 2. ตัวอย่างนี้เป็นการสร้างวิวข้อมูล (Materialized view) ที่เก็บข้อมูลการเช่า video ของลูกค้าที่อาศัยอยู่ที่จังหวัดนครราชสีมา โดยจะเก็บข้อมูล รหัสลูกค้า ชื่อลูกค้า ชื่อภาพยนตร์ที่ลูกค้าเช่าไป ตัวอย่างนี้เป็นตัวอย่างที่แสดงการสร้างวิวข้อมูลจากข้อมูลหลายตาราง โดยใช้ชุดคำสั่งดังนี้

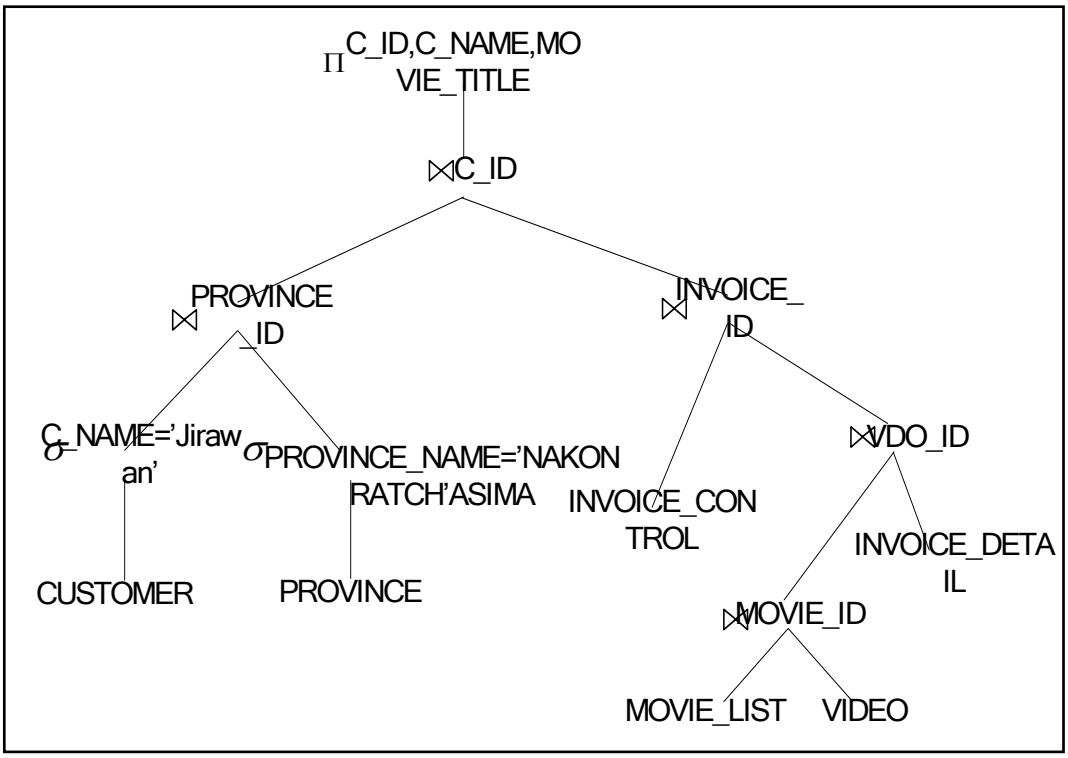
```
CREATE MATERIALIZED VIEW  "INV_BY_CUS_KORAT_MV"
AS select       "CUSTOMER"."C_ID" as "C_ID",
                "CUSTOMER"."C_NAME" as "C_NAME",
                "PROVINCE"."PROVINCE_NAME" as "PROVINCE_NAME",
                "MOVIE_LIST"."MOVIE_TITLE" as "MOVIE_TITLE"
from            "MOVIE_LIST" "MOVIE_LIST",
                "VIDEO" "VIDEO",
                "INVOICE_DETAIL" "INVOICE_DETAIL",
                "INVOICE_CONTROL" "INVOICE_CONTROL",
                "PROVINCE" "PROVINCE",
                "CUSTOMER" "CUSTOMER"
where           "PROVINCE"."PROVINCE_ID"="CUSTOMER"."PROVINCE_ID"
and             "CUSTOMER"."C_ID"="INVOICE_CONTROL"."C_ID"
and             "INVOICE_CONTROL"."INVOICE_ID"="INVOICE_DETAIL"
                ."INVOICE_ID"
and             "VIDEO"."VDO_ID"="INVOICE_DETAIL"."VDO_ID"
and             "MOVIE_LIST"."MOVIE_ID"="VIDEO"."MOVIE_ID"
and             "PROVINCE"."PROVINCE_NAME" ='NAKHONRATCHASIMA'
```

ถ้ามีข้อคำถามเกี่ยวกับลูกค้าที่มีที่อยู่ที่จังหวัดนครราชสีมาว่าต้องการทราบรายชื่อภาพยนตร์ที่ลูกค้าชื่อ Jirawan เช่าไป จะสามารถสอบถามได้ด้วยชุดคำสั่งต่อไปนี้

```
Query2A :
select          "CUSTOMER"."C_ID" as "C_ID",
                "CUSTOMER"."C_NAME" as "C_NAME",
                "MOVIE_LIST"."MOVIE_TITLE" as "MOVIE_TITLE"
from            "MOVIE_LIST" "MOVIE_LIST",
                "VIDEO" "VIDEO",
                "INVOICE_DETAIL" "INVOICE_DETAIL",
                "INVOICE_CONTROL" "INVOICE_CONTROL",
                "PROVINCE" "PROVINCE",
```

```
                "CUSTOMER" "CUSTOMER"
where           "PROVINCE"."PROVINCE_ID"="CUSTOMER"."PROVINCE_ID"
and             "INVOICE_DETAIL"."INVOICE_ID"="INVOICE_CONTROL"
                ."INVOICE_ID"
and             "CUSTOMER"."C_ID"="INVOICE_CONTROL"."C_ID"
and             "INVOICE_DETAIL"."VDO_ID"="VIDEO"."VDO_ID"
and             "VIDEO"."MOVIE_ID"="MOVIE_LIST"."MOVIE_ID"
and             "CUSTOMER"."C_NAME" ='Jirawan'
and             "PROVINCE"."PROVINCE_NAME" ='NAKHONRATCHASIMA'
```

แผนการประมวลผลข้อคำถามแสดงได้ดังตัวอย่างรูปที่ 2.26 และข้อมูลที่เป็นผลลัพธ์ของการประมวลผลปรากฏดังตัวอย่างรูปที่ 2.27



รูปที่ 2.26 แผนการประมวลผลข้อคำถาม Query 2A

C_ID	C_NAME	MOVIE_TITLE
7981	Jirawan	THE AMERICANIZATION OF EMILY
7981	Jirawan	ANATOMY OF A MURDER
7981	Jirawan	THE AFRICAN QUEEN

Execution time (3 times) 78 msecs., 63 msecs., 77 msecs.

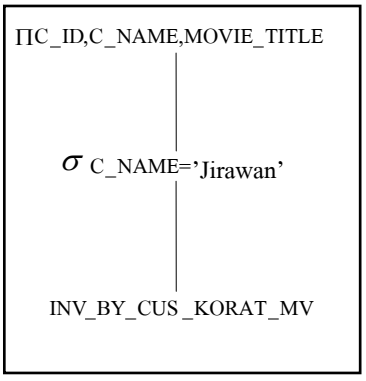
รูปที่ 2.27 ผลจากการประมวลผลชุดคำสั่ง Query2A

จากวิวข้อมูล INV_BY_CUS_KORAT_MV สามารถแปลง Query 2A เป็น Query 2B ดังนี้

Query 2B :

```
SELECT    "INV_BY_CUS_KORAT_MV"."C_ID" as "C_ID",
          "INV_BY_CUS_KORAT_MV"."C_NAME" as "C_NAME",
          "INV_BY_CUS_KORAT_MV"."MOVIE_TITLE" as "MOVIE_TITLE"
FROM      "INV_BY_CUS_KORAT_MV" "INV_BY_CUS_KORAT_MV"
WHERE     "INV_BY_CUS_KORAT_MV"."C_NAME" ='Jirawan'
```

แผนการประมวลผลข้อคำถาม (รูปที่ 2.29) จะมีขั้นตอนลดลง โดยผลลัพธ์ของการประมวลผลแสดงได้ดังรูปที่ 2.29



รูปที่ 2.28 แผนการประมวลผลข้อคำถาม Query 2B

C_ID	C_NAME	MOVIE_TITLE
7981	Jirawan	THE AMERICANIZATION OF EMILY
7981	Jirawan	ANATOMY OF A MURDER
7981	Jirawan	THE AFRICAN QUEEN

Execution time (3 times) 16 msecs., 16 msecs., 15 msecs.

รูปที่ 2.29 ผลจากการประมวลผลชุดคำสั่ง Query 2B

เมื่อเปรียบเทียบผลของการประมวลผลชุดคำสั่งทั้งสอง จะเห็นว่าข้อมูลที่ได้ออกมา นั้นเหมือนกัน แต่เวลาที่ใช้ประมวลผลนั้นต่างกัน โดยจะเห็นว่าเวลาที่ใช้ในการประมวลผล ชุดคำสั่ง Query 2B นั้นใช้เวลาน้อยกว่า Query 2A ซึ่งเป็นผลมาจากการลดขนาดของข้อมูล และลดความซับซ้อนของข้อคำถามโดยการใช้วิวข้อมูล

การนำวิวข้อมูลเข้ามาใช้ในการตอบคำถามแทนตารางข้อมูลพื้นฐานเป็นอีกวิธีการที่สำคัญที่จะช่วยเพิ่มประสิทธิภาพในการประมวลผลข้อคำถาม โดยพยายามหาเส้นทางที่จะเข้าถึงข้อมูลได้อย่างรวดเร็วที่สุด อย่างไรก็ตามการใช้วิวข้อมูลยังมีข้อจำกัดอยู่ที่การตัดสินใจว่า เมื่อไรที่เราควรจะใช้วิวข้อมูลในการประมวลผลข้อคำถามซึ่งปัญหานี้เป็นอุปสรรคเป็นอย่างมากต่อการนำวิวข้อมูลไปใช้ช่วยในการประมวลผลข้อคำถาม ในงานวิจัยก่อนหน้านี้จะมีการพิจารณาวิวข้อมูลเพียงวิวเดียวเท่านั้นที่มีความเกี่ยวข้องสัมพันธ์กันกับข้อคำถามที่จะถูกนำขึ้นมาช่วยเพิ่มประสิทธิภาพการประมวลผล ส่วนวิวข้อมูลที่เป็นประโยชน์อื่น ๆ จะถูกตัดออกจากการพิจารณา ต่อมามีนักวิจัยนำเสนอในบทความที่เป็นแนวทางใหม่ของการนำวิวข้อมูลมาใช้ในการตอบคำถาม ซึ่งสามารถตอบคำถามฟังก์ชันการรวม (Aggregate function) และสามารถกำหนดเงื่อนไขในที่ซึ่งวิวข้อมูลถูกใช้ในการสร้างข้อคำถามใหม่ได้

Chang and Lee (1998) ได้เสนอแนวคิดของการนำวิวข้อมูล (Materialized views) มาใช้ช่วยในการตอบข้อคำถาม โดยนำเสนอผ่านตัวอย่างต่อไปนี้ ซึ่งกำหนดสคีมาของข้อมูลพื้นฐานให้ดังนี้

Sales (<u>eid</u>, <u>item</u>, <u>vol</u>, <u>date</u>)

Emp (<u>eid</u>, name, dept, salary, age)

Dept (<u>dept</u>, manager, loc)

Item (<u>item</u>, category, size, weight)

ตัวอย่างชุดคำสั่งต่อไปนี้ทำหน้าที่สร้างวิวข้อมูล V (eid, name, item, category) ที่เก็บรายละเอียดข้อมูลการขายที่มีปริมาณการขายเกิน 100

```
CREATE VIEW V  AS
(SELECT     E.eid, E.name, I.item, I.category
FROM        Emp E, Sales S, Item I
WHERE       E.eid = S.eid AND  S.item = I.item AND  S.vol > 100)
```

ในกรณีที่มีข้อคำถาม สอบถามเกี่ยวกับพนักงาน (พร้อมทั้งให้แสดงชื่อแผนกที่สังกัดและชื่อผู้จัดการแผนก) ที่มีความสามารถขายคอมพิวเตอร์ได้มากกว่า 100 เครื่อง

```
SELECT      E.name, E.dept, D.manager
FROM        Emp E, Sales S, Dept D
WHERE       E.eid = S.eid  AND  E.dept = D.dept
            AND  S.item = 'computer'  AND  S.vol > 100
```

จากกรณีมีวิวข้อมูล V ทำให้สามารถแปลงข้อคำถามเดิมให้อยู่ในรูปแบบอื่นได้ดังนี้

```
SELECT      V.name, E.dept, D.manager
FROM        V, Emp E, Dept D
WHERE       V.eid = E.eid  AND  V.dept = D.dept
            AND  V.item = 'computer'
```

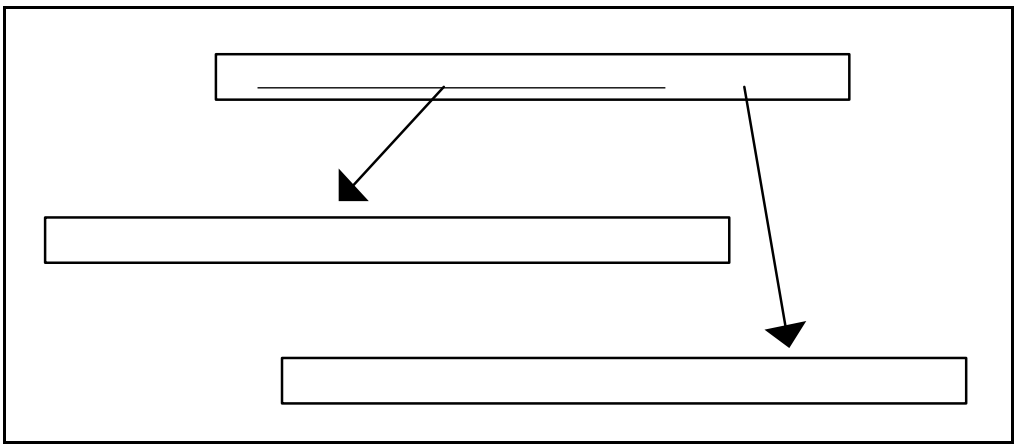
ดังนั้นการประมวลผลข้อคำถามที่ถูกเปลี่ยนแปลงแล้วจึงมีโอกาสที่จะได้รับการเพิ่มประสิทธิภาพให้ประมวลผลได้เร็วขึ้นด้วยการแปลงให้ไปสอบถามข้อมูลจากวิวข้อมูลแทน ถ้าหากขนาดของวิวข้อมูล V เล็กกว่าขนาดของข้อมูลพื้นฐาน Sales

2.5 การคัดเลือกวิวข้อมูลและการแปลงรูปแบบข้อคำถามด้วยวิวข้อมูล (Materialized view selection and query rewriting with materialized views)

การคัดเลือกวิวข้อมูลและการแปลงรูปแบบข้อคำถามด้วยวิวข้อมูลถือว่าเป็นขั้นตอนที่สำคัญมากที่สุดในกระบวนการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลในหัวข้อนี้จะกล่าวถึงรายละเอียดและวิธีการต่าง ๆ

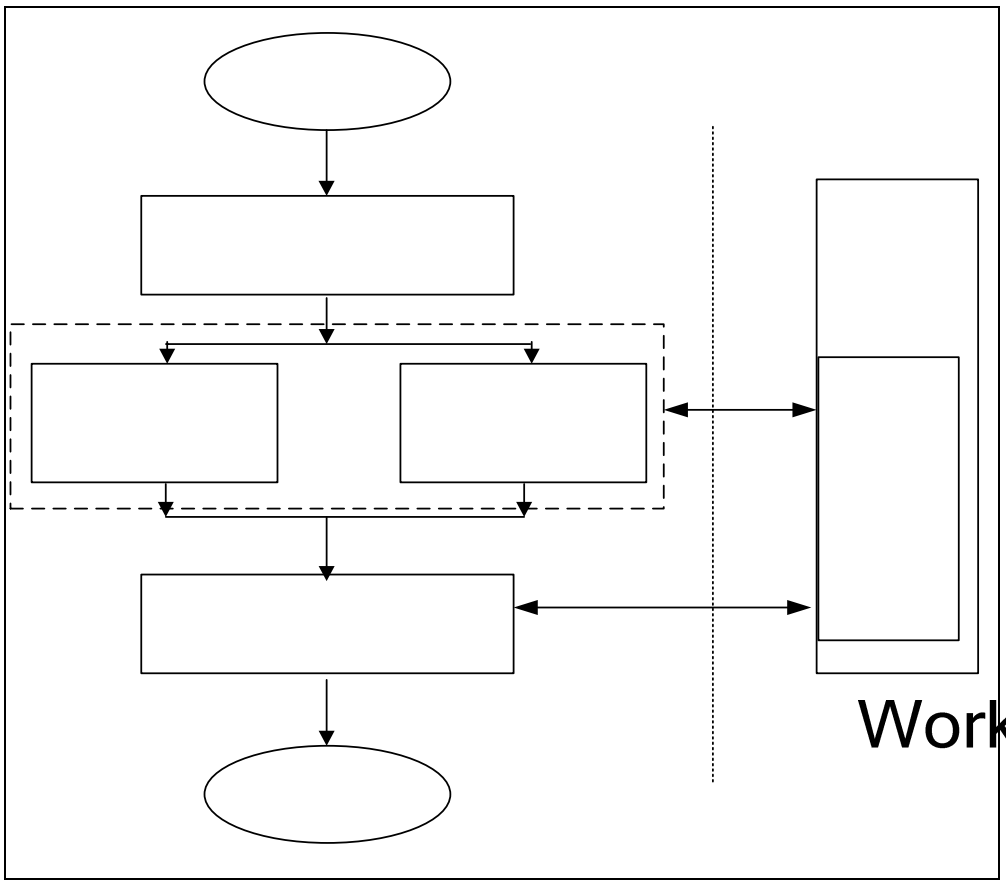
2.5.1 การคัดเลือก (Materialized view selection)

จากที่ได้กล่าวมาแล้ว ปัญหาหลักของการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลมักจะเกิดขึ้นในขั้นตอนการคัดเลือกวิวข้อมูล เมื่อมีการออกแบบระบบการเพิ่มประสิทธิภาพประมวลผลข้อคำถามสิ่งที่สำคัญที่สุดคือ การพยายามหาวิธีการที่จะช่วยลดเวลาที่ใช้ในการตอบสนองต่อข้อคำถามให้ได้มากที่สุด นั่นคือ ระบบจะต้องตอบข้อคำถามให้ได้เร็วที่สุดนั่นเอง โดยเฉพาะในระบบฐานข้อมูลที่มีขนาดใหญ่และข้อคำถามที่มีความซับซ้อนที่ถูกใช้บ่อย ๆ จำเป็นอย่างยิ่งที่ต้องมีระบบนี้เพื่อตอบสนองความต้องการของผู้ใช้ได้ดียิ่งขึ้น โดยปกติแล้วคำจำกัดความหรือข้อกำหนดของการใช้วิวข้อมูลอย่างมีประสิทธิภาพ และการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลจะกล่าวอ้างถึงปัญหาการคัดเลือกวิวข้อมูลอยู่เสมอจะเห็นว่าสิ่งสำคัญที่สุดในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลคือ การเลือกใช้วิวข้อมูล เช่น จากตัวอย่างการคัดเลือกวิวข้อมูลในรูปที่ 2.30 ข้อคำถาม Q มีความสัมพันธ์กับวิวข้อมูล MV1 และ MV2 และวิวข้อมูลทั้งสองจะมีข้อมูลรายชื่อของลูกค้า (C_NAME) เหมือนกัน สิ่งที่น่าสนใจคือ จะรู้ได้อย่างไรว่าควรจะเลือกวิวข้อมูลใดมาตอบข้อคำถามนั้น



รูปที่ 2.30 ตัวอย่างการคัดเลือกวิวข้อมูล

Chaudnturi, Narasayya and Agrawal (2000) ได้นำเสนอโครงสร้างและอัลกอริทึมที่เป็นพื้นฐานของเครื่องมือออกแบบฐานข้อมูลในระดับกายภาคของ Microsoft SQL Server 2000 ที่เกี่ยวกับการคัดเลือกวิวข้อมูลแบบอัตโนมัติ (รูปที่ 2.31) ซึ่งเป็นเทคนิคการกำหนดว่าวิวข้อมูลใดควรจะได้รับการคัดเลือก



รูปที่ 2.31 โครงสร้างสำหรับการคัดเลือกวิวข้อมูลแบบอัตโนมัติ (Chaudnturi, Narasayya and Agrawal, 2000)

Syntactic structu
selection

โครงสร้างและอัลกอริทึมนี้เป็นแนวความคิดที่เหมาะสมอย่างยิ่งที่จะนำไปเป็นพื้นฐานสำหรับการพัฒนาการออกแบบระดับกายภาพของฐานข้อมูลและการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล

Candidate
index
selection

vi

2.5.2 การแปลงรูปแบบข้อคำถามด้วยวิวข้อมูล (Query rewriting with materialized views)

ประโยชน์หลักของการสร้างและบำรุงรักษาวิวข้อมูล คือ สามารถนำมาใช้ในการแปลงรูปแบบข้อคำถามได้ การแปลงรูปแบบข้อคำถามเป็นเทคนิคหนึ่งในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม (Pottinger and Levy, 2000) โดยจะเป็นการเปลี่ยนรูปข้อคำถามที่ผู้ใช้เขียนสอบถามข้อมูลจากตาราง และวิวให้ไปใช้ข้อมูลจากวิวข้อมูลแทน ทำให้สามารถประมวลผลได้เร็วขึ้น

อย่างไรก็ตามก่อนที่จะทำการแปลงรูปแบบข้อคำถามจะต้องมีการตรวจสอบข้อคำถามว่าสามารถจะเปลี่ยนแปลงได้หรือไม่ ถ้าการตรวจสอบล้มเหลวหรือไม่สามารถเปลี่ยนแปลงรูปแบบข้อคำถามได้ จะทำให้ต้องใช้เวลาในการประมวลผลเพิ่มขึ้นโดยเปล่าประโยชน์จากส่วนการทำงานเหล่านี้

การแปลงรูปแบบข้อคำถามนั้นสามารถทำได้หลายวิธี วิธีที่พบเห็นบ่อย ๆ จะมีอยู่ 5 รูปแบบ (Hobbs, 2005) ดังนี้

1) การเปรียบเทียบ (Exact match)

การเปรียบเทียบ (Exact match) เป็นการเปลี่ยนแปลงรูปแบบข้อคำถามแบบง่าย โดยนำข้อคำถามไปเปรียบเทียบกับเงื่อนไขในการสร้างวิวข้อมูล การแปลงรูปแบบข้อคำถามชนิดนี้จะมีวิธีการเปรียบเทียบข้อคำถาม 2 ชนิดคือ

- การเปรียบเทียบทั้งหมด (Full text match)
- การเปรียบเทียบบางส่วน (Partial text match)

การแปลงรูปแบบข้อคำถามแบบง่ายนี้จะเริ่มการทำงานด้วยการเปรียบเทียบทั้งหมด (Full text match) คือ การนำข้อคำถามมาเปรียบเทียบกับเงื่อนไขในการสร้างวิวข้อมูล ถ้าตรงกันก็สามารถทำการแปลงรูปแบบข้อคำถามนี้ได้ แต่ถ้าไม่ตรงกันก็จะเริ่มการเปรียบเทียบบางส่วน (Partial text match) การเปรียบเทียบแบบนี้จะพิจารณาที่ส่วนของ FROM WHERE และ GROUP BY

ตัวอย่างต่อไปนี้จะแสดงความเป็นไปได้ในการใช้วิวข้อมูลในรูปที่ 2.32 เพื่อแปลงรูปแบบข้อคำถาม จากนั้นพิจารณาข้อคำถามจากรูปที่ 2.33 เมื่อทำการเปรียบเทียบ (Exact match) กับวิวข้อมูล all_cust_sales_mv จะเห็นว่าสามารถแปลงรูปแบบข้อคำถามนี้ได้

```
CREATE MATERIALIZED VIEW        all_cust_sales_mv
BUILD IMMEDIATE REFRESH
COMPLETE
ENABLE QUERY REWRITE
AS SELECT     c.cust_id, sum (s.amount_sold) AS bahts, p.prod_id ,
              sum (s.quantity_sold) as quantity
FROM          sales s, customers c, products p
WHERE         c.cust_id = s.cust_id AND s.prod_id = p.prod_id
GROUP BY      c.cust_id, p.prod_id;
```

รูปที่ 2.32 วิวข้อมูล all_cust_sales_mv

พิจารณาข้อคำถามจากรูปที่ 2.33 เมื่อได้เปรียบเทียบ (Exact match) กับวิวข้อมูล all_cust_sales_mv จะเห็นว่าสามารถแปลงรูปแบบข้อคำถามนี้ได้

```
SELECT        c.cust_id, sum(s.quantity_sold) as quantity
FROM          sales s , customers c, products p
WHERE         c.cust_id = s.cust_id AND s.prod_id = p.prod_id
GROUP BY      c.cust_id, p.prod_id;
```

รูปที่ 2.33 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Exact match

2) การรวมย้อนกลับ (Join back)

การรวมย้อนกลับ (Join back) เป็นการเปลี่ยนแปลงรูปแบบข้อคำถามในกรณีที่วิวข้อมูลไม่สามารถตอบคำถามได้ทั้งหมด โดยบางข้อคำถามจะอ้างถึงคอลัมน์ที่ไม่มีในวิวข้อมูล ดังนั้นจะนำวิวข้อมูลไปใช้ร่วมกับตารางข้อมูลในส่วนที่ข้อคำถามอ้างถึงแต่ไม่ได้ถูกเก็บไว้ในวิวข้อมูล เช่น เมื่อต้องการแปลงรูปแบบข้อคำถามในรูปที่ 2.34 โดยใช้วิวข้อมูล all_cust_sales_mv (รูปที่ 2.32) วิวข้อมูลจะตอบคำถามไม่ได้ทั้งหมด จะมีคอลัมน์ c.cust_last_name ที่วิวข้อมูลไม่

สามารถตอบคำถามได้ ดังนั้นข้อมูลในส่วนนี้จึงจำเป็นต้องไปสอบถามกับตารางพื้นฐานร่วมกับ การใช้วิวข้อมูล

```
SELECT      c.cust_last_name, sum (s.quantity_sold) as quantity
FROM        sales s, customers c, products p
WHERE       c.cust_id = s.cust_id AND s.prod_id = p.prod_id
GROUP BY    c.cust_id, p.prod_id;
```

รูปที่ 2.34 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Join back

3) การผนึกและการรวม (Rollup & aggregate rollup)

การผนึกและการรวม (Rollup & aggregate rollup) เป็นการเปลี่ยนแปลงรูปแบบข้อคำถามในกรณีที่ข้อคำถามต้องการใช้ฟังก์ชันการรวม (Aggregate function) เช่น ฟังก์ชัน SUM ฟังก์ชัน MIN ฟังก์ชัน MAX เป็นต้น

จากที่กล่าวมาแล้วว่าวิวข้อมูลสามารถเก็บข้อมูลที่อยู่ในรูปแบบฟังก์ชันการรวม (Aggregate function) ได้ ดังนั้นการเปลี่ยนแปลงรูปแบบข้อคำถามนี้จะแปลงข้อคำถามให้ไปใช้ข้อมูลที่ผ่านการประมวลผลด้วยฟังก์ชันการรวมไว้แล้วในวิวข้อมูลแทนการใช้ตารางข้อมูล ตัวอย่างเช่น เมื่อพิจารณาข้อคำถามในรูปที่ 2.35 จะเห็นว่ามีการใช้ฟังก์ชัน SUM อยู่และเมื่อนำข้อคำถามนี้ไปเปรียบเทียบกับวิวข้อมูล all_cust_sales_mv จะเห็นว่าวิวข้อมูลนี้สามารถตอบคำถามในส่วนที่เป็นการใช้ฟังก์ชัน SUM ได้ดังนั้นจึงสามารถแปลงรูปแบบข้อคำถามนี้ โดยใช้วิวข้อมูล all_cust_sales_mv ได้

```
SELECT      c.cust_province, sum(s.quantity_sold) as quantity
FROM        sales s , customers c, products p
WHERE       c.cust_id = s.cust_id AND s.prod_id = p.prod_id
GROUP BY    c.cust_province;
```

รูปที่ 2.35 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Rollup & aggregate rollup

4) การแบ่งข้อมูลเป็นกลุ่มย่อย (Data subsets)

การแบ่งข้อมูลเป็นกลุ่มย่อย (Data subsets) เป็นการเปลี่ยนแปลงรูปแบบข้อคำถามที่อ้างถึงวิวข้อมูลแบบมีเงื่อนไข ปกติแล้ววิวข้อมูลจะสามารถเก็บข้อมูลไว้เป็นจำนวนมาก ฉะนั้นเป็นไปได้ที่จะมีการกำหนดเงื่อนไขในการสร้างวิวข้อมูลเพื่อลดขนาดของข้อมูลโดยใช้ IN และ BETWEEN กำหนดช่วงของข้อมูล สมมุติถ้ามีข้อคำถามที่สอบถามถึงข้อมูลในช่วงข้อมูลนี้ จากตารางข้อมูลก็สามารถแปลงรูปแบบข้อคำถามให้มาใช้วิวข้อมูลนี้ได้ ดั้งนั้นการเปลี่ยนแปลงรูปแบบข้อคำถามนี้จะสามารถเพิ่มประสิทธิภาพในการประมวลผลได้ ตัวอย่างเช่น วิวข้อมูล some_cust_sales_mv (รูปที่ 2.36) เป็นวิวข้อมูลที่สร้างโดยมีการใช้ IN เพื่อกำหนดเงื่อนไขให้ข้อมูลในวิวข้อมูลมีจำนวนน้อยลงกว่าตารางพื้นฐาน ซึ่งส่งผลให้การประมวลผลข้อคำถามเมื่อสอบถามกับวิวข้อมูลนี้จะมีประสิทธิภาพมากกว่าการสอบถามข้อมูลโดยตรงกับตารางข้อมูลพื้นฐาน

```
CREATE MATERIALIZED VIEW        some_cust_sales_mv
BUILD IMMEDIATE
REFRESH COMPLETE
ENABLE QUERY REWRITE
AS
SELECT      c.cust_id, sum(s.amount_sold) AS bahts,
            p.prod_ id, sum(s.quantity_sold) as quantity
FROM        sales s , customers c, products p
WHERE       c.cust_id = s.cust_id AND s.prod_id = p.prod_id
AND         c.cust_state_province
IN          ('Bangkok', 'Phuket', 'Chiang Mai', 'Nakhon Ratchasima')
GROUP BY    c.cust_id, p.prod_id;
```

รูปที่ 2.36 วิวข้อมูล some_cust_sales_mv

พิจารณาข้อคำถามจากรูปที่ 2.37 ข้อคำถามนี้ต้องการสอบถามข้อมูลที่เป็น ข้อมูลย่อย (Data subset) ของวิวข้อมูล some_cust_sales_mv จะเห็นว่าสามารถแปลงรูปแบบข้อคำถามนี้ให้ไปสอบถามกับวิวข้อมูลแทนที่การสอบถามข้อมูลจากตารางพื้นฐานได้ ดังนั้นการประมวลผล ข้อคำถามที่ทำการแปลงรูปแบบข้อคำถามแล้วจะมีประสิทธิภาพมากขึ้น

```
SELECT      c.cust_province, sum(s.quantity_sold) as quantity
FROM        sales s , customers c, products p
WHERE       c.cust_id = s.cust_id AND s.prod_id = p.prod_id
AND         c.cust_province
IN          ('Bangkok','Phuket')
GROUP BY    c.cust_province;
```

รูปที่ 2.37 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธี Data subset

5) การใช้หลายวิวข้อมูล (Using multiple materialized views)

การใช้หลายวิวข้อมูล (Using multiple materialized views) ที่ผ่านมามีหลายครั้งที่วิวข้อมูลเพียงวิวเดียวไม่สามารถตอบคำถามของข้อคำถามได้ จำเป็นต้องใช้ข้อมูลจากหลาย วิวข้อมูลรวมกันจึงสามารถตอบคำถามได้ ตัวอย่างเช่น วิวข้อมูลในรูปที่ 2.38 และวิวข้อมูลในรูปที่ 2.39 เป็นวิวข้อมูลที่ถูกสร้างมาเพื่อช่วยในการแปลงรูปแบบข้อคำถามให้การประมวลผลมีประสิทธิภาพมากขึ้น เมื่อพิจารณาข้อคำถามจากรูปที่ 2.40 จะเห็นว่าไม่มีวิวข้อมูลใดเลยที่สามารถตอบข้อคำถามนี้ได้ทั้งหมด แต่ถ้านำข้อมูลจากวิวข้อมูลทั้งสองนี้มารวมกันก็จะสามารถตอบข้อคำถามได้ทั้งหมด ดังนั้นจึงต้องใช้วิวข้อมูลทั้งสองนี้มาช่วยในการแปลงรูปแบบข้อคำถาม

จากที่ได้กล่าวมาทั้งหมดนี้สามารถแสดงให้เห็นถึงความเป็นไปได้และประสิทธิภาพของการนำวิวข้อมูลเข้ามาช่วยในการตอบข้อคำถาม แต่อย่างไรก็ตามการนำวิวข้อมูลเข้ามาใช้จำเป็นต้องคำนึงถึงการใช้พื้นที่ในการเก็บข้อมูลที่เพิ่มขึ้น และค่าใช้จ่ายที่จะเพิ่มขึ้นมาในส่วนตรวจสอบและการแปลงข้อคำถาม การสร้างและการคัดเลือกวิวข้อมูลอย่างเหมาะสมเป็นขั้นตอนที่สำคัญอย่างยิ่งในการนำวิวข้อมูลมาช่วยในการตอบคำถามได้อย่างถูกต้องและมีประสิทธิภาพสูงสุด

งานวิจัยนี้มีจุดมุ่งหมายที่จะพัฒนาแนวทางในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล โดยจะเสนอแนวคิดและอัลกอริทึมในการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถามและแนวทางในการนำข้อมูลทางสถิติมาช่วยในการสร้างวิวข้อมูลเพื่อให้ได้วิวข้อมูลที่สามารถนำมาใช้ประโยชน์ในการเพิ่มประสิทธิภาพการประมวลผล ข้อคำถามได้อย่างเต็มที่

```
CREATE MATERIALIZED VIEW        east_sales_mv
BUILD IMMEDIATE
REFRESH COMPLETE
ENABLE QUERY REWRITE
AS
SELECT      c.cust_id, sum(s.amount_sold) AS bahts,
            p.prod_id, sum(s.quantity_sold) as quantity
FROM        sales s , customers c, products p
WHERE       c.cust_id = s.cust_id AND s.prod_id = p.prod_id
AND         c.region_id IN (52779, 52789,52770,52777)
GROUP BY    c.cust_id, p.prod_id;
```

รูปที่ 2.38 วิวข้อมูล east_sales_mv

```
CREATE MATERIALIZED VIEW        north_sales_mv
BUILD IMMEDIATE
REFRESH COMPLETE
ENABLE QUERY REWRITE
AS
SELECT      c.cust_id, sum(s.amount_sold) AS bahts,
            p.prod_id, sum(s.quantity_sold) as quantity
FROM        sales s , customers c, products p
WHERE       c.cust_id = s.cust_id AND s.prod_id = p.prod_id
AND         c.region_id IN (52790, 52772,52773)
GROUP BY    c.cust_id, p.prod_id;
```

รูปที่ 2.39 วิวข้อมูล north_sales_mv

```
SELECT      c.country_id, t.country_name, sum(s.amount_sold) AS bahts,
            p.prod_id, sum(s.quantity_sold) as quantity
FROM        sales s , customers c, products p , region t
WHERE       c.cust_id = s.cust_id AND s.prod_p = p.prod_id
AND         c.region_id
IN          (52790, 52789) and c.region_id=t.region_id
GROUP BY    c. region_id, p.prod_id, t. region_name;
```

รูปที่ 2.40 ตัวอย่างข้อคำถามที่สามารถแปลงรูปแบบด้วยวิธีใช้หลายวิวข้อมูล

บทที่ 3
ระเบียบวิธีวิจัย

งานวิจัยนี้มีจุดมุ่งหมายที่จะพัฒนาวิธีการพิจารณาข้อมูลเพื่อที่จะนำมาสร้างเป็นวิวข้อมูล และอัลกอริทึมที่ใช้ในการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม (Query rewriting) โปรแกรมที่พัฒนาขึ้นใช้ภาษาจาวา (Java) ในการพัฒนา โดยมุ่งเน้นให้มีวิธีการสร้างวิวข้อมูลที่เกิดประโยชน์โดยจะพิจารณาจากสถิติการเข้าใช้วิว หรือข้อคำถามที่ถูกใช้บ่อย ๆ และอัลกอริทึมสามารถคัดเลือกวิวข้อมูลที่จะนำมาแปลงรูปแบบข้อคำถามเพื่อเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม โดยเฉพาะข้อคำถามที่ถูกใช้บ่อย ๆ ได้อย่างเหมาะสม รายละเอียดในเนื้อหาบทนี้ประกอบด้วย ขั้นตอนการวิจัย ปรากฏอยู่ในหัวข้อ 3.1 โปรแกรม MV4QR ที่พัฒนาขึ้น ปรากฏอยู่ในหัวข้อ 3.2 คำอธิบายชุดข้อมูลที่ใช้ในการทดสอบอัลกอริทึม ปรากฏอยู่ในหัวข้อ 3.3 และหัวข้อ 3.4 เป็นรายละเอียดวิธีการเปรียบเทียบผลลัพธ์ที่ได้จากการทดสอบ

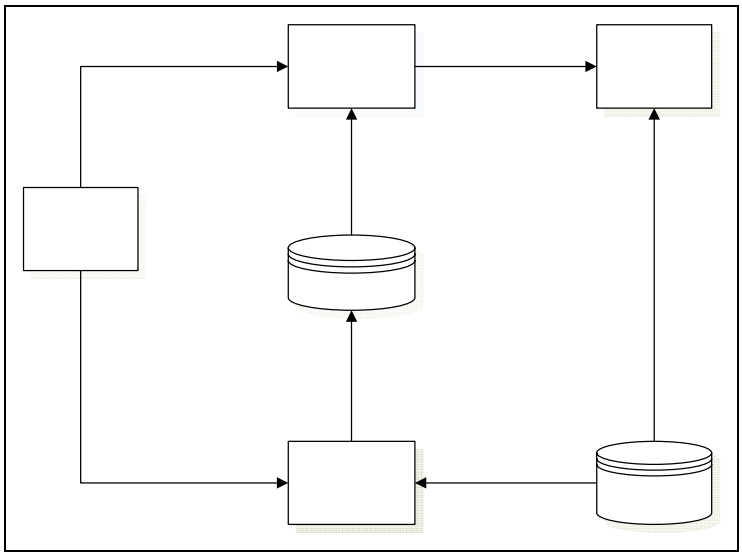
3.1 ขั้นตอนการวิจัย

1) ศึกษาค้นคว้าและรวบรวมสรุปงานวิจัยที่เกี่ยวข้อง
2) ศึกษาระบบงานการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลและการคัดเลือกวิวข้อมูล
3) ศึกษาและพัฒนาวิธีการวิเคราะห์และคัดเลือกข้อมูลมาสร้างเป็นวิวข้อมูล
4) กำหนดวิธีการเลือกวิวข้อมูล และวิธีพิจารณาความเข้ากันได้ของวิวข้อมูลกับข้อคำถามของผู้ใช้ เป็นขั้นตอนการออกแบบและกำหนดรายละเอียดของวิธีการใช้งานวิวข้อมูล โดยเริ่มตั้งแต่การเลือกวิวข้อมูลที่เกี่ยวข้องไปจนถึงเกณฑ์การพิจารณาว่าจะใช้วิวข้อมูลปรับปรุงข้อคำถามได้อย่างไรบ้าง
5) ศึกษาและกำหนดแนวทางในการนำวิวข้อมูลที่ได้จากการคัดเลือกไปใช้เพื่อการแปลงรูปแบบข้อคำถาม
6) รวบรวมข้อมูลที่จะใช้ในการทดสอบจาก Transaction Processing Performance Council : TPC (http://www.tpc.org) และ Oracle (http://www.oracle.com)
7) ออกแบบและพัฒนาอัลกอริทึมในการคัดเลือกวิวข้อมูลตามวิธีการที่ได้ออกแบบไว้
8) พัฒนาโปรแกรมตามอัลกอริทึมที่ได้ออกแบบไว้

9) ทดสอบประสิทธิภาพของอัลกอริทึมจากโปรแกรมที่ได้พัฒนาขึ้นมาโดยเครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบเป็นคอมพิวเตอร์ Laptop CPU Centrino Core 2 Duo ความเร็ว 1.8 GHz หน่วยความจำหลัก 1 GB ฮาร์ดดิสก์ความจุ 160 GB
10) วิเคราะห์และสรุปผลการวิจัย

3.2 โปรแกรม MV4QR สำหรับสร้างและคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม

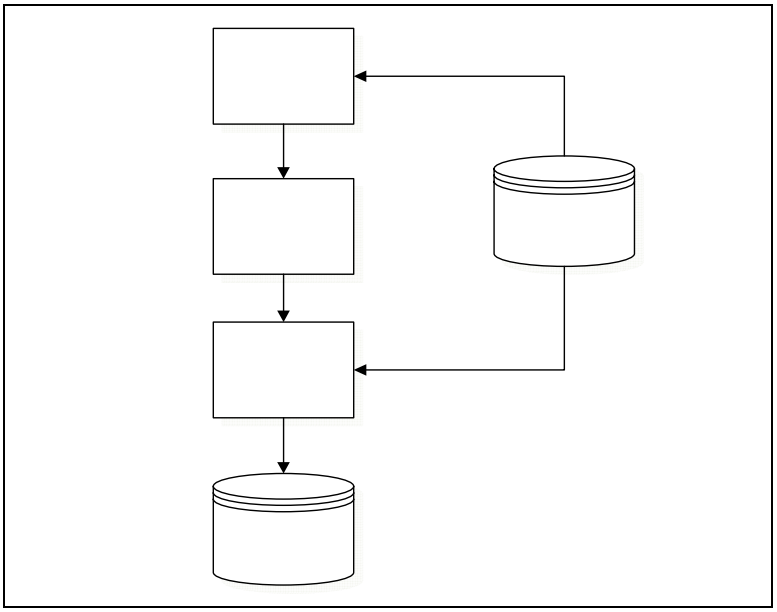
แนวทางการออกแบบระบบนี้จะแยกการทำงานออกเป็นสองส่วนใหญ่ ๆ คือ ส่วนที่ทำการวิเคราะห์ข้อคำถามและวิวเพื่อสร้างเป็นวิวข้อมูล และส่วนที่ทำหน้าที่คัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม การทำงานส่วนแรกของโปรแกรมคือ ในส่วนที่ทำการวิเคราะห์ข้อคำถามและ วิวเพื่อสร้างเป็นวิวข้อมูล (MV-generator) จะเริ่มการทำงานโดยการวิเคราะห์ข้อมูลทางสถิติของการประมวลผลข้อคำถามและการเข้าถึงวิวเพื่อนำมาเป็นข้อมูลในการสร้างวิวข้อมูลโดยจะเลือกข้อคำถามและวิวที่มีสถิติการประมวลผลและสถิติการเข้าถึงข้อมูลในวิวนั้นผ่านเงื่อนไขที่กำหนดมาสร้างเป็นวิวข้อมูลเพื่อรอการใช้งานต่อไป การทำงานในส่วนที่สองจะเป็นส่วนของการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม การทำงานในส่วนนี้จะเริ่มการทำงานเมื่อมีการป้อนข้อคำถามเข้ามา จะมีการนำข้อคำถามนั้นไปตรวจสอบกับวิวข้อมูลที่มีอยู่ว่ามีวิวข้อมูลใดบ้างที่สามารถใช้ตอบคำถามของข้อคำถามที่ถูกป้อนเข้ามาได้บ้าง และทำการคัดเลือกวิวข้อมูลที่สามารถตอบคำถามได้นั้นเพื่อทำการแปลงรูปแบบข้อคำถามและส่งข้อคำถามที่ถูกแปลงรูปแบบแล้วให้ระบบจัดการฐานข้อมูล (DBMS) เพื่อทำการประมวลผล โครงสร้างการทำงานนี้แสดงเป็นแผนภาพได้ดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างของโปรแกรม MV4QR

3.2.1 เทคนิคการพิจารณาข้อมูลเพื่อที่จะนำมาสร้างเป็นวิวข้อมูล

เทคนิคในการพิจารณานำข้อมูลที่มีอยู่มาสร้างเป็นวิวข้อมูลซึ่งเป็นการเตรียมวิวข้อมูลไว้ใช้สำหรับการแปลงรูปแบบข้อคำถามเพื่อให้วิวที่สร้างมานั้นเกิดประโยชน์สูงสุดและคุ้มค่ากับพื้นที่ใช้เก็บวิวข้อมูลที่เพิ่มขึ้นมา ซึ่งจะพิจารณาจากสถิติการใช้ข้อคำถามเพื่อสอบถามข้อมูลแบบซ้ำ ๆ โดยจะเน้นที่การพิจารณาการเรียกใช้วิวซึ่งเป็นตารางจำลองที่สร้างขึ้นมาใหม่จากเงื่อนไขที่ได้กำหนดไว้แล้ว โดยจะอ่านข้อมูลมาจากตารางจริงในฐานข้อมูลจากเงื่อนไขที่กำหนด โดยวิวที่สร้างขึ้นมานั้นอาจจะอ่านข้อมูลมาจากตารางเดียวหรือหลายๆตารางรวมกันก็ได้ โดยทั่วไปแล้วชุดข้อมูลที่ผู้ใช้สนใจและถูกเรียกดูบ่อย ๆ จะถูกสร้างเป็นวิวไว้แล้วเพื่อลดความซับซ้อนของข้อคำถามใน การสอบถามข้อมูลเนื่องจากการสร้างวิวนั้นไม่ได้เกิดความสิ้นเปลืองทรัพยากรเพราะไม่ได้เก็บข้อมูลจริง แต่ด้วยเหตุนี้การสอบถามข้อมูลจากวิวยังคงต้องมีการเข้าถึงตารางข้อมูลพื้นฐานทุกครั้งเมื่อมีการเรียกใช้วิวจึงจะเป็นประโยชน์อย่างยิ่งถ้ามีการเปลี่ยนวิวที่ถูกเรียกใช้บ่อย ๆ ให้เป็นวิวข้อมูล ซึ่งขั้นตอนการคัดเลือกข้อมูลเพื่อนำมาสร้างเป็นวิวข้อมูลสามารถแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 ขั้นตอนการคัดเลือกข้อมูลเพื่อนำมาสร้างเป็นวิวข้อมูล

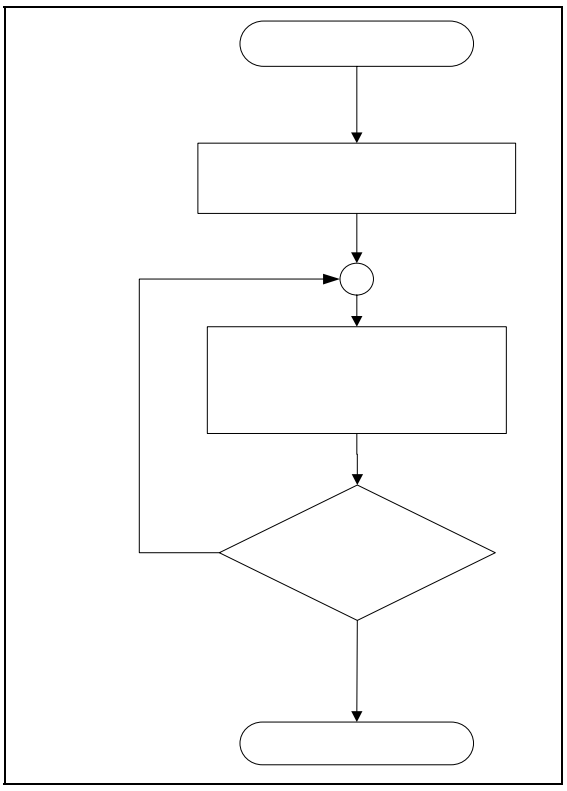
ปกติแล้วระบบจัดการฐานข้อมูลจะมีการเก็บข้อมูลเกี่ยวกับประวัติการใช้ฐานข้อมูลไว้เพื่อการตรวจสอบ และการวิเคราะห์การใช้ฐานข้อมูล ระบบจัดการฐานข้อมูลของ Oracle ก็เป็นระบบจัดการฐานข้อมูลที่มีการเก็บข้อมูลนั้นเช่นกัน วิวที่ชื่อ V_$SQLSTATS เป็นวิวที่เก็บข้อมูลของชุดคำสั่งที่ถูกประมวลผลไปแล้วของระบบจัดการฐานข้อมูล Oracle เราสามารถสอบถามข้อมูล

Analyze
data

จากวิวนี้เพื่อตรวจสอบได้ว่าชุดคำสั่งที่ถูกประมวลผลไปแล้วมีชุดคำสั่งใดบ้าง และสามารถนำมาใช้ตรวจสอบชุดคำสั่งที่เกี่ยวข้องกับวิวที่เราต้องการวิเคราะห์ได้ว่าวิวนั้นถูกเรียกใช้ไปกี่ครั้งแล้ว

1) การสร้างวิวข้อมูลโดยพิจารณาจากสถิติการประมวลผลข้อคำถาม

ในส่วนนี้จะเป็นโปรแกรมย่อยที่ใช้ในการพิจารณาข้อมูลที่จะนำมาสร้างเป็นวิวข้อมูลโดยพิจารณาจากสถิติการประมวลข้อคำถามซึ่งจะเป็นส่วนการทำงานส่วนแรกของรูปที่ 3.2 คือ ส่วนวิเคราะห์ข้อมูล (Analyze data) โปรแกรมย่อยนี้จะเริ่มต้นทำงานโดยจะตรวจสอบและดึงข้อมูลของตารางข้อมูลพื้นฐานหรือวิวจากสคีมา (Schema) ที่เลือกขึ้นมาแล้วนำข้อมูลที่ได้นี้ไปตรวจสอบกับสถิติการประมวลผลข้อคำถามกับวิวที่ชื่อ V_$SQLSTATS ว่ามีข้อคำถามใดบ้างที่มีความสัมพันธ์กับข้อมูลจากสคีมาที่ได้มาและดึงข้อมูลสถิติการประมวลผลข้อคำถามเหล่านั้นขึ้นมาเพื่อตรวจสอบกับเงื่อนไขที่กำหนด โดยข้อมูลที่นำออกมานั้นจะประกอบไปด้วย ข้อคำถาม จำนวนครั้งในการประมวลผลข้อคำถามนั้น และจำนวนแถวหรือทูเพิลของข้อมูลที่ได้มา การทำงานของฟังก์ชันในการวิเคราะห์ข้อมูลนี้แสดงในรูปที่ 3.3 หรือเขียนอยู่ในลักษณะของซูโดโค้ดแสดงได้ดังรูปที่ 3.4 ตัวอย่างการวิเคราะห์ข้อคำถามเพื่อบันทึกสถิติการประมวลผลแสดงได้ดังตารางที่ 3.1



รูปที่ 3.3 แสดงผังการทำงานส่วนวิเคราะห์สถิติการประมวลผลข้อคำถาม

Declare TV to be the set of tables and views to appear in schema,

P_i is number of query executions,

R_i is number of records query result

Q_i is SQL text

Begin

For each $TV_i \in TV$

Q_i = sqlText (TV_i)

P_i = executions (TV_i);

R_i = rowsProcessed (TV_i);

End For

End

รูปที่ 3.4 อัลกอริทึมการวิเคราะห์สถิติการประมวลผลข้อคำถาม

ตารางที่ 3.1 แสดงตัวอย่างข้อมูลที่ได้จากวิเคราะห์สถิติการประมวลผลข้อคำถาม

User' queries	Execution times	Rows processed
SELECT "E"."EMPLOYEE_ID" AS "EMPLOYEE_ID", "E"."FIRST_NAME" AS "FIRST_NAME", "E"."LAST_NAME" AS "LAST_NAME", "J"."JOB_TITLE" AS "JOB_TITLE", "E"."SALARY" AS "SALARY" FROM "JOBS" "J", "EMPLOYEES" "E" WHERE "E"."JOB_ID"="J"."JOB_ID"	1	11
SELECT "JH"."START_DATE" AS "START_DATE", "JH "."END_DATE" AS "END_DATE", "J"."JOB_TITLE" AS "JOB_TITLE", "D"."DEPARTMENT_NAME" AS "E"."FIRST_NAME" AS "FIRST_NAME", "E"."LAST_NAME" AS "LAST_NAME", "E"."EMPLOYEE_ID" AS "EMPLOYEE_ID"		

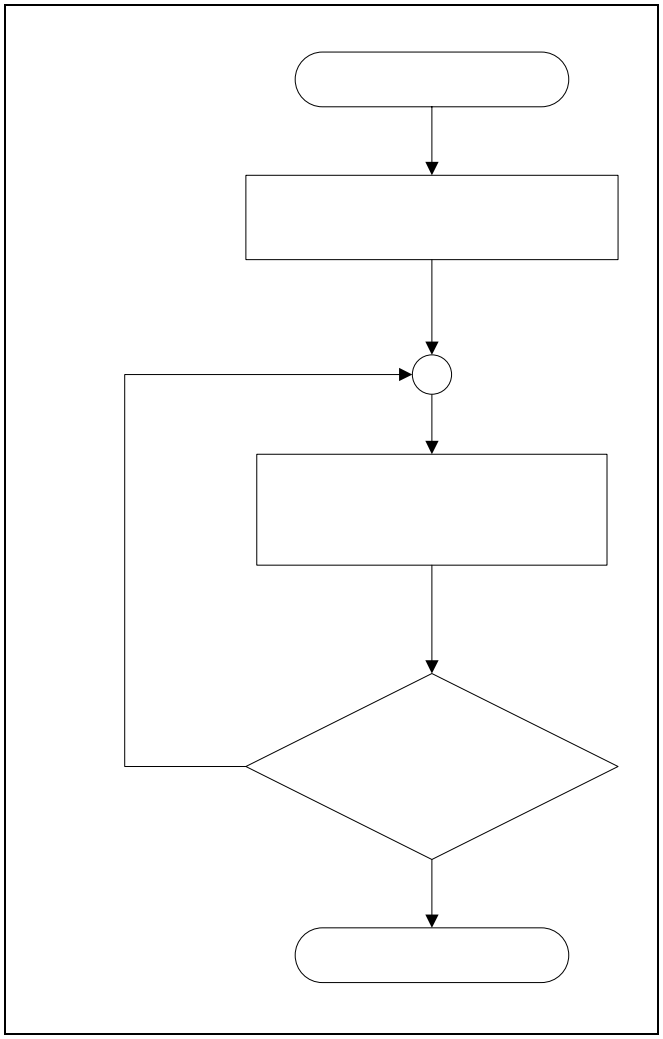
ตารางที่ 3.1 แสดงตัวอย่างข้อมูลที่ได้จากวิเคราะห์สถิติการประมวลผลข้อคำถาม (ต่อ)

User' queries	Execution times	Rows processed
FROM "JOBS" "J", "DEPARTMENTS" "D", "EMPLOYEES" "E", "JOB_HISTORY" "JH " WHERE "JH "."EMPLOYEE_ID"= "E"."EMPLOYEE_ID" AND "JH"."DEPARTMENT_ID"= "D"."DEPARTMENT_ID" AND "JH"."JOB_ID"="J"."JOB_ID"	1	11
SELECT "EMPLOYEE_ID","FIRST_NAME", "LAST_NAME","START_DATE", "END_DATE","JOB_TITLE", "DEPARTMENT_NAME" FROM "HR"."EMP_JOB_HISTORY_VIEW"	4	24
SELECT "C"."C_ID" as "C_ID", "C"."C_NAME" as "C_NAME", "C"."C_ADDRESS" as "C_ADDRESS", "C"."C_PHONE" as "C_PHONE", "C"."C_GENDER" as "C_GENDER" FROM "PROVINCE" "P", "CUSTOMER" "C" WHERE "P"."PROVINCE_ID"="C"."PROVINCE_ID" AND "P"."PROVINCE_NAME" ='NAKHONRATCHASIMA';	13	1227

2) การสร้างวิวข้อมูลโดยพิจารณาจากสถิติการเข้าใช้วิว

ในส่วนนี้จะเป็นโปรแกรมย่อยที่ใช้ในการพิจารณาข้อมูลที่จะนำมาสร้างเป็นวิวข้อมูลโดยพิจารณาจากสถิติการเข้าถึงวิวหรือการสอบถามข้อมูลจากวิวซึ่งจะเป็นส่วนการทำงานส่วนแรกของรูปที่ 3.2 คือส่วนวิเคราะห์ข้อมูล (Analyze data)โปรแกรมย่อยนี้จะเริ่มต้นทำงานโดยจะตรวจสอบและดึงข้อมูลของวิว จากสคีมา (Schema) ที่เลือกขึ้นมาแล้วนำข้อมูลที่ได้นี้ไปตรวจสอบ

กับสถิติการประมวลผลข้อคำถามจากวิวชื่อ V_$SQLSTATS ว่ามีข้อคำถามใดบ้างที่มีความสัมพันธ์กับข้อมูลของวิวที่ได้จากสคีมาและดึงข้อมูลสถิติการประมวลผลข้อคำถามที่เกี่ยวข้องกับวิวขึ้นมา ซึ่งประกอบไปด้วยชื่อวิว จำนวนการเข้าถึงข้อมูลหรือจำนวนการประมวลผลข้อคำถามที่เกี่ยวข้องกับวิวนั้น และจำนวนแถวของข้อมูลที่เป็นผลลัพธ์ของการประมวลผลข้อคำถามทั้งหมดที่เกี่ยวข้องกับวิวนั้นเพื่อตรวจสอบกับเงื่อนไขที่กำหนด การทำงานของฟังก์ชันในการวิเคราะห์ข้อมูลนี้แสดงในรูปที่ 3.5 โดยซูโดโค้ดแสดงได้ดังรูป 3.6 และแสดงตัวอย่างการวิเคราะห์ข้อมูลได้ดังตารางที่ 3.2



รูปที่ 3.5 แสดงผังการทำงานส่วนที่วิเคราะห์สถิติการเข้าถึงวิวข้อมูล

```
Declare        V to be the set of view to appear in schema,
               N_i to be the amount of data access to view,
               M_i to be number of records in summary view
Begin
    For each V_i ∈ V
        N_i = sumOfExecutions (V_i);
        M_i = sumOfRowsProcessed (V_i);
    End For
End
```

รูปที่ 3.6 อัลกอริทึมการวิเคราะห์สถิติการเข้าถึงวิวข้อมูล

ตารางที่ 3.2 แสดงตัวอย่างข้อมูลที่ได้จากการวิเคราะห์สถิติการเข้าถึงวิวข้อมูล

View	Data accessed times	Rows processed
EMP_DETAILS_VIEW	5	425
EMP_JOB_VIEW	6	642
EMP_JOB_HISTORY_VIEW	4	25

3) การกำหนดเงื่อนไขและการตรวจสอบเงื่อนไข

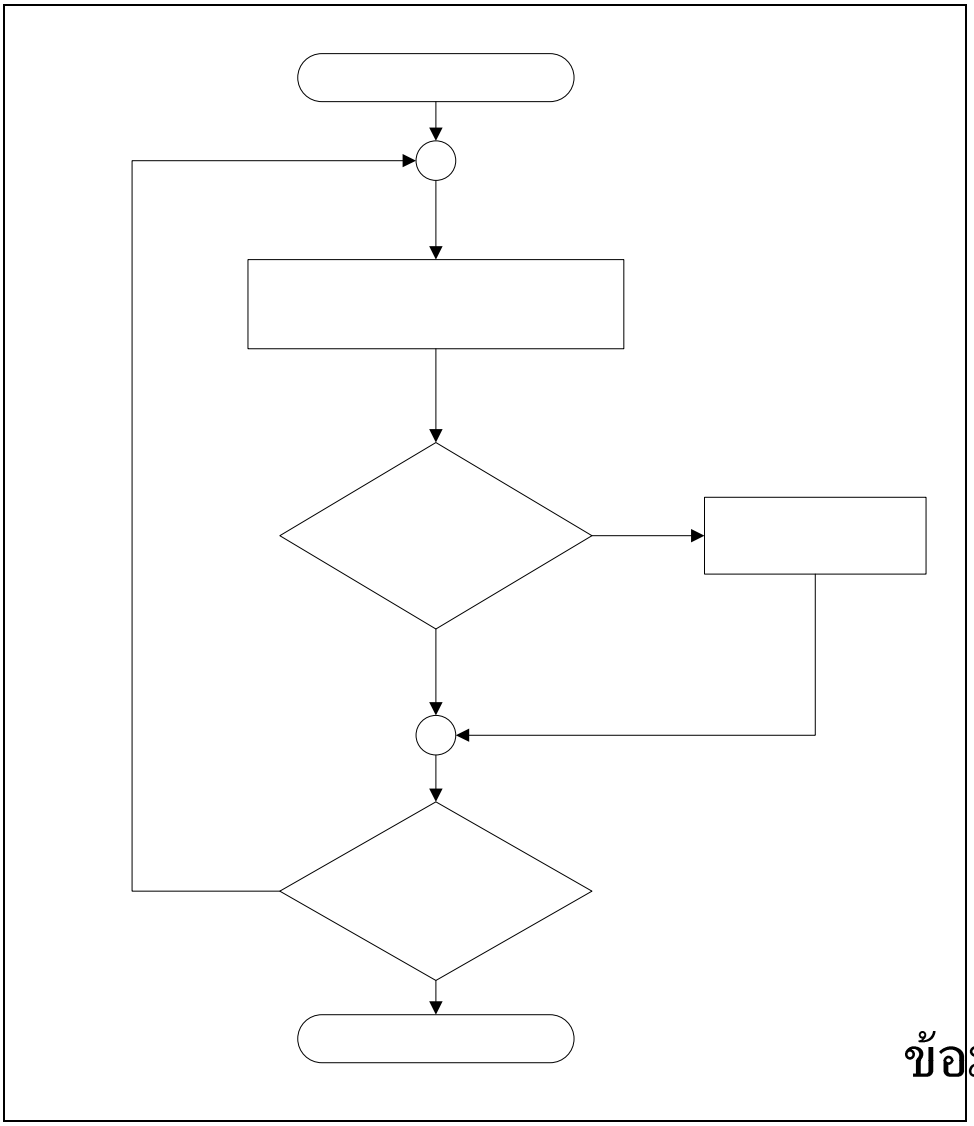
ในส่วนนี้จะเป็นการกำหนดเงื่อนไขในการพิจารณาข้อมูลและนำข้อมูลที่ได้จากส่วนแรกมาตรวจสอบกับเงื่อนไขที่กำหนดที่ได้จากส่วนแรกเพื่อสร้างเป็นวิวข้อมูล ซึ่งส่วนนี้จะอยู่ในส่วนที่สองของรูปที่ 3.2 คือส่วนการตรวจสอบ (Audit) การทำงานส่วนนี้จะเริ่มต้นขึ้นได้จำเป็นต้องมีการกำหนดเงื่อนไขในการพิจารณาไว้ก่อนโดยจะมีค่าที่จำเป็นต้องกำหนดดังนี้

3.1) ค่าขั้นต่ำของจำนวนการประมวลผลหรือการเข้าถึงวิว

3.2) จำนวนแถวของข้อมูลที่ได้จากผลลัพธ์

ตัวอย่างเช่น กำหนดค่าขั้นต่ำของจำนวนการประมวลผลหรือการเข้าถึงวิวไว้ที่ 5 และกำหนดจำนวนแถวของข้อมูลที่ได้จากผลลัพธ์ขั้นต่ำไว้ที่ 500 ถ้ามีข้อคำถามใดที่มีจำนวนครั้งของการประมวลผล หรือจำนวนการเข้าถึงข้อมูลของวิวและจำนวนแถวของข้อมูลผ่านเกณฑ์ขั้นต่ำที่กำหนดไว้ก็จะนำข้อมูลนั้นไปสร้างเป็นวิวข้อมูลจากค่าในตัวอย่างที่กำหนดไว้นั้นจะมีวิวชื่อ EMP_JOB_VIEW

ผ่านเงื่อนไขที่กำหนดไว้ดังนั้นวิวนี้จะถูกนำไปสร้างเป็นวิวข้อมูลต่อไป ผังการทำงานซูโดโค้ดและตัวอย่างแสดงได้ดังรูปที่ 3.7, 3.8 และตารางที่ 3.3 ตามลำดับ



รูปที่ 3.7 แสดงผังการทำงานส่วนตรวจสอบเงื่อนไขเพื่อนำไปสร้างวิวข้อมูล

```
Declare L = {} //L is the set of queries and views that meet the predefined
                //thresholds
ST = set of query and view statistics
Begin
        For i=1 to |ST|
            If audit (ST_i) = = accept
                L=L ⋃ ST_i ;
            End If
        End For
End
```

รูปที่ 3.8 อัลกอริทึมการตรวจสอบเงื่อนไขเพื่อนำไปสร้างวิวข้อมูล

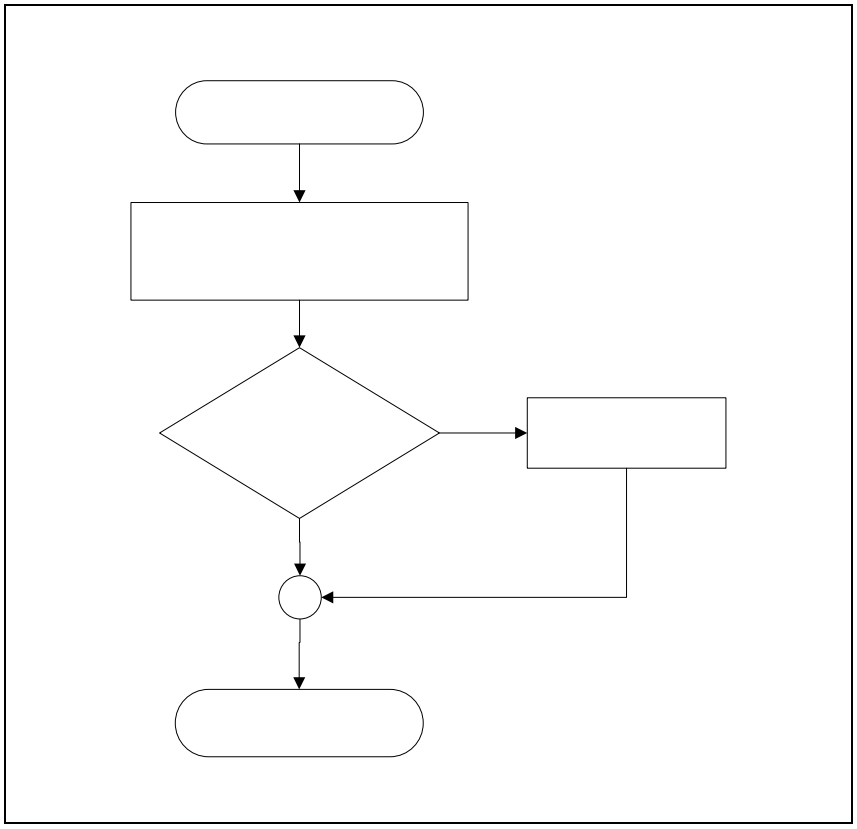
ตารางที่ 3.3 แสดงตัวอย่างข้อมูลที่ผ่านเงื่อนไข

View or query	Data accessed times	Rows processed
SELECT "C"."C_ID" as "C_ID", "C"."C_NAME" as "C_NAME", "C"."C_ADDRESS" as "C_ADDRESS", "C"."C_PHONE" as "C_PHONE", "C"."C_GENDER" as "C_GENDER" FROM "PROVINCE" "P", "CUSTOMER" "C" WHERE "P"."PROVINCE_ID"="C"."PROVINCE_ID" AND "P"."PROVINCE_NAME" ='NAKHONRATCHASIMA';	13	1227
EMP_JOB_VIEW	6	642

4) การสร้างวิวข้อมูล

ในส่วนนี้จะเป็นส่วนที่นำข้อคำถามหรือวิวที่ผ่านเงื่อนไขที่กำหนดมาสร้างเป็นวิวข้อมูลซึ่งส่วนนี้จะเป็นส่วนสุดท้ายที่ปรากฏในรูปที่ 3.2 คือ ส่วนการสร้างวิวข้อมูล (Generate MV) ขั้นตอนนี้ (แสดงดังรูปที่ 3.9) จะเริ่มการทำงานโดยการนำข้อคำถามหรือวิวที่ผ่านการตรวจสอบ

เงื่อนไขมาทำการตรวจสอบว่าได้ถูกสร้างเป็นวิวข้อมูลแล้วหรือยัง ถ้ามีการสร้างไปแล้วก็จะข้ามขั้นตอนการสร้างวิวข้อมูลนี้ไปแต่ถ้ายังไม่ถูกสร้างก็จะทำการสร้างวิวข้อมูล ซึ่งการสร้างวิวข้อมูลจากข้อคำถามนี้สามารถนำข้อคำถามที่ผ่านการตรวจสอบมาเป็นเงื่อนไขในการสร้างวิวข้อมูลได้ทันทีตามโครงสร้างของคำสั่งสร้างวิวข้อมูล และข้อมูลของวิวข้อมูลที่ได้จะเป็นข้อมูลเดียวกันกับข้อมูลที่ได้จากการประมวลผลข้อคำถามนั้น ๆ ส่วนการสร้างวิวข้อมูลจากวิวนั้นก็สามารถทำได้โดยการเลือกข้อมูลทั้งหมดของวิวมาสร้างเป็นวิวข้อมูล ดังนั้นวิวข้อมูลที่ได้จะมีข้อมูลที่เหมือนกันกับข้อมูลที่ได้จากวิว ขั้นตอนเหล่านี้เขียนเป็นซูโดโค้ดได้ดังรูปที่ 3.10



รูปที่ 3.9 แสดงผังการทำงานส่วนสร้างวิวข้อมูล

เริ่มต้น

อ่านข้อคำถามหรือวิวที่ผ่า

```
Declare L to be the set of queries and views that meet the predefined thresholds
Begin
        For i=1 to |L|
            IF L_i not existing
                CreateMvfrom(L_i);
            End If
        End For
End
```

รูปที่ 3.10 อัลกอริทึมการสร้างวิวข้อมูล

ข้อคำถามและวิวที่ผ่านเงื่อนไขที่กำหนดเกณฑ์ขั้นต่ำกว่าจำนวนการประมวลผลไม่ต่ำกว่า 5 ครั้ง และจำนวนผลลัพธ์ไม่ต่ำกว่า 500 ทูเพิล (ตามตารางที่ 3.3) จะถูกนำมาสร้างเป็นวิวข้อมูล ดังปรากฏในรูปที่ 3.11 และ 3.1

```
CREATE MATERIALIZED VIEW  "C_KORAT_MV"   AS
SELECT          "CUSTOMER"."C_ID" as "C_ID",
                "CUSTOMER"."C_NAME" as "C_NAME",
                "CUSTOMER"."C_ADDRESS" as "C_ADDRESS",
                "CUSTOMER"."C_PHONE" as "C_PHONE",
                "CUSTOMER"."C_GENDER" as "C_GENDER"
FROM            "PROVINCE" "PROVINCE",
                "CUSTOMER" "CUSTOMER"
WHERE           "PROVINCE"."PROVINCE_ID"="CUSTOMER"."PROVINCE_ID"
 AND            "PROVINCE"."PROVINCE_NAME" ='NAKHONRATCHASIMA';
```

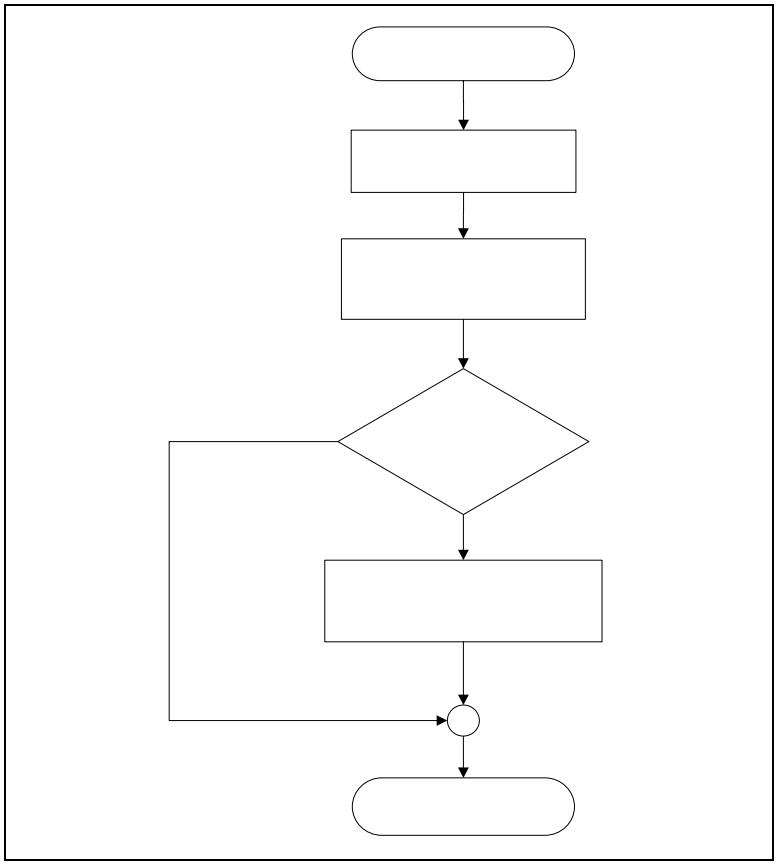
รูปที่ 3.11 ตัวอย่างคำสั่งในการสร้างวิวข้อมูลจากข้อคำถาม

```
CREATE MATERIALIZED VIEW  "EMP_JOB_MV"
AS
SELECT * FROM EMP_JOB_VIEW;
```

รูปที่ 3.12 ตัวอย่างคำสั่งในการสร้างวิวข้อมูลจากวิว

3.2.2 เทคนิคในการคัดเลือกวิวข้อมูลเพื่อการแปลงรูปแบบข้อคำถาม

ในส่วนนี้จะเป็นส่วนที่ทำการคัดเลือกวิวข้อมูลที่ถูกสร้างเอาไว้แล้ว และเป็นวิวข้อมูลที่มีความสัมพันธ์กับข้อคำถามที่ผู้ใช้ป้อนเข้ามา การทำงานในส่วนนี้จะเริ่มการทำงานเมื่อผู้ใช้ป้อนข้อคำถามเข้ามาโปรแกรมจะคัดเลือกวิวข้อมูลที่มีความสัมพันธ์กับข้อคำถามซึ่งการคัดเลือกวิวข้อมูลจะมีกระบวนการทำงานดังรูปที่ 3.13

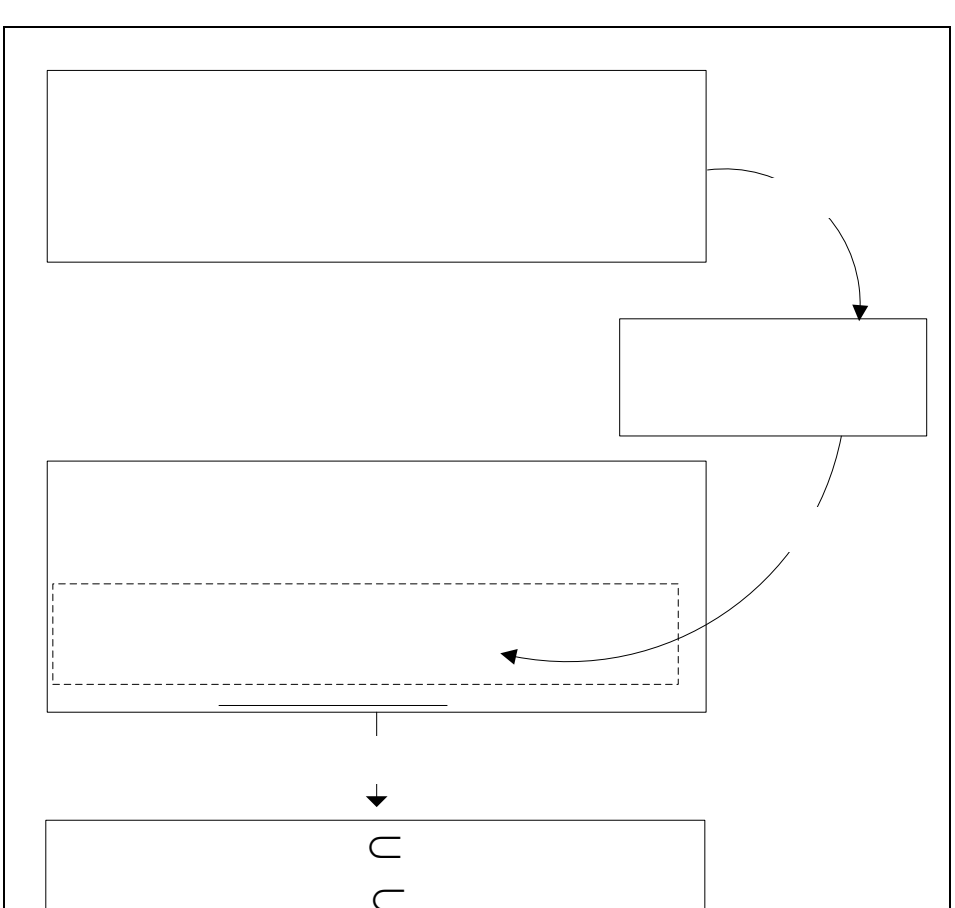


รูปที่ 3.13 แสดงผังการทำงานส่วนคัดเลือกวิวข้อมูล

เทคนิคในการคัดเลือกวิวข้อมูลเพื่อการแปลงรูปแบบข้อคำถาม สามารถแบ่งการทำงานออกเป็นสองขั้นตอนดังนี้

1) การคัดเลือกวิวข้อมูลที่สามารถตอบข้อคำถามได้

วิวข้อมูลที่สามารถตอบข้อคำถามได้หรือวิวข้อมูลที่มีความสัมพันธ์กับข้อคำถาม หมายถึง วิวข้อมูลที่สามารถตอบข้อคำถามได้ทั้งหมด หรือ บางส่วนของความต้องการของข้อคำถาม ในหนึ่งข้อคำถามอาจมีความต้องการทราบถึงข้อมูลหลาย ๆ ส่วนได้ ในขั้นตอนนี้จะทำการคัดเลือกวิวข้อมูลที่มีความสัมพันธ์กับข้อคำถามออกมาทั้งหมด โดยการตรวจสอบข้อคำถามว่าข้อคำถามนี้ต้องการสอบถามข้อมูลจากตารางใดและทำการแบ่งข้อคำถามนั้นออกเป็นข้อคำถามย่อย จากนั้นนำไปตรวจสอบกับวิวข้อมูลทั้งหมดที่มีอยู่ ซึ่งการเปรียบเทียบนี้จะนำข้อคำถามย่อยไปเปรียบเทียบกับส่วนที่เป็นการกำหนดเงื่อนไขในการสร้างวิวข้อมูลว่ามีความสัมพันธ์กับตารางที่ต้องการหรือไม่ ดังแสดงตัวอย่างในรูปที่ 3.14 และแสดงผังการทำงานได้ดังรูปที่ 3.15

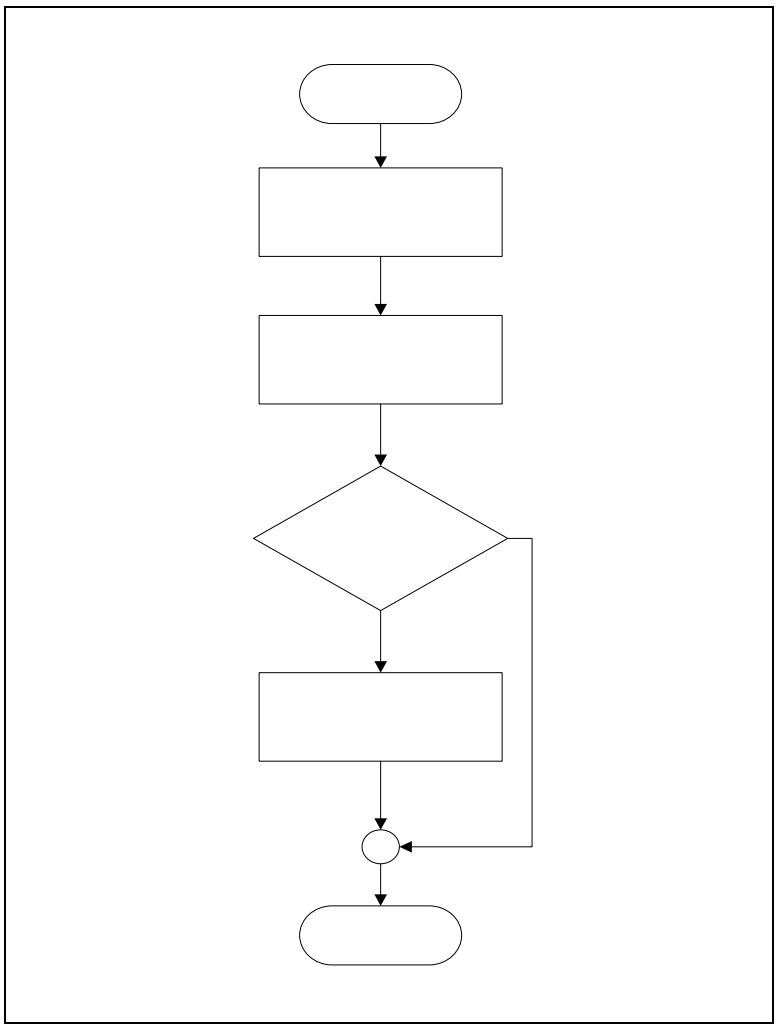


รูปที่ 3.14 ตัวอย่างการหาความสัมพันธ์ของข้อคำถามกับวิวข้อมูล

Query : Q

SELECT C.C_ID, C.CNAM

C_ADDRESS, MC

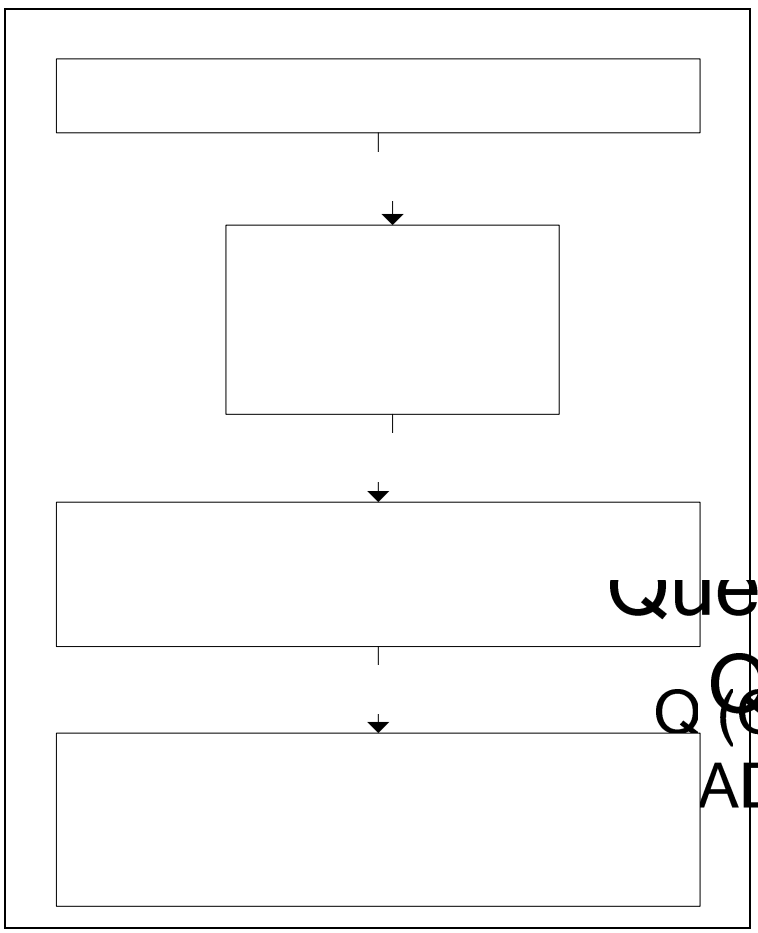


รูปที่ 3.15 แสดงผังการทำงานส่วนการคัดเลือกวิวข้อมูลที่สามารถตอบข้อคำถามได้

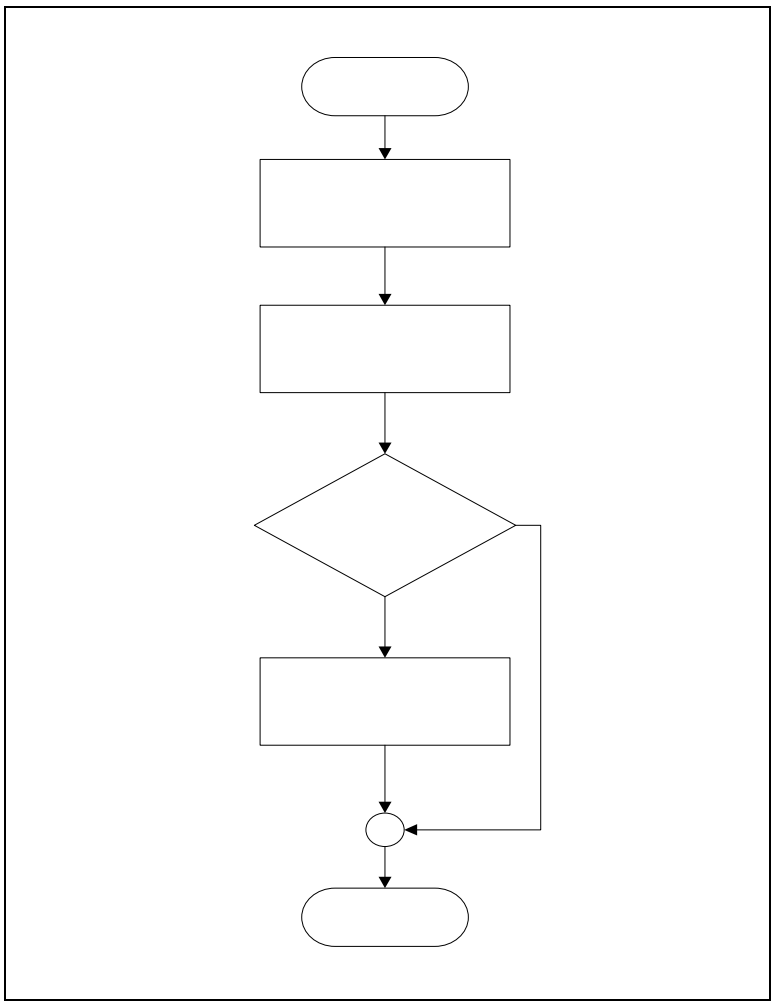
2) คัดเลือกวิวข้อมูลที่ตอบคำถามได้ดีที่สุด

จากขั้นตอนที่ 1 จะได้ของเซตของวิวข้อมูลที่มีความสัมพันธ์กับข้อคำถามออกมา แต่ข้อมูลในแต่ละวิวข้อมูลมีข้อมูลที่ซ้ำซ้อนกันอยู่ ในขั้นตอนนี้จะเป็นการคัดเลือกวิวข้อมูลที่ดีที่สุดที่สามารถใช้ตอบคำถามได้ซึ่งจะเป็นการตัดวิวข้อมูลที่ทำหน้าที่ในการตอบคำถามที่ซ้ำซ้อนออกไป โดยนำเอาข้อคำถามที่ถูกป้อนเข้ามาแปลงเป็นข้อคำถามย่อย (Sub query) แล้วนำแต่ละข้อคำถามย่อยไปตรวจสอบกับวิวข้อมูลทั้งหมดที่มีอยู่ ซึ่งการเปรียบเทียบนี้จะนำข้อคำถามย่อยไปเปรียบเทียบกับส่วนที่เป็นการกำหนดเงื่อนไขในการสร้างวิวข้อมูลเพื่อหาว่ามีวิวข้อมูลใดที่สามารถตอบข้อคำถามย่อยได้ ผลที่ได้จะเป็นความสัมพันธ์ของข้อคำถามย่อยกับวิวข้อมูลที่สามารถตอบคำถามได้ ดังแสดงตัวอย่างในรูปที่ 3.16 จากนั้นจะนำข้อมูลทางสถิติของความสัมพันธ์นี้ไปทำ

การวิเคราะห์หาวิวข้อมูลที่สามารถตอบคำถามได้ดีที่สุดตามอัลกอริทึม MVS ซึ่งเป็นอัลกอริทึมการคัดเลือกวิวข้อมูล คำตอบที่ได้จากขั้นตอนนี้จะเป็นเซตของวิวข้อมูลที่ดีที่สุดที่สามารถนำไปแปลงรูปแบบข้อคำถามได้ ขั้นตอนการคัดเลือกวิวข้อมูลที่ดีที่สุด แสดงเป็นแผนผังได้ดังรูปที่ 3.17 และอัลกอริทึม MVS แสดงได้ดังรูป 3.18



รูปที่ 3.16 ตัวอย่างการหาความสัมพันธ์ของข้อคำถามย่อยกับวิวข้อมูล



รูปที่ 3.17 แสดงผังการทำงานส่วนการคัดเลือกวิวข้อมูลที่สามารถตอบข้อคำถามได้

```
M = {} // M is the set of materialized views that is useful for at least one
       // query in workload W#
         For i = 1 to |W|
                  Let S_i = Set of materialized views proposed for Q_i
                  C = MvBestSelect(Q_i,S_i)
                  M = M ⋃ C;
         End For
         Return M
```

รูปที่ 3.18 อัลกอริทึมการคัดเลือกวิวข้อมูล (MVS)

อัลกอริทึม MvBestSelect เป็นอัลกอริทึมที่ใช้ในการคัดเลือกวิวข้อมูลที่สามารถตอบคำถามได้ดีที่สุดโดยจะนำไปใช้ร่วมกับอัลกอริทึมการคัดเลือกวิวข้อมูล (รูปที่ 3.18) อัลกอริทึม MvBestSelect สามารถแสดงได้ดังรูปที่ 3.19

```
Let S_i = Set of materialized views proposed for Q_i
If (|S_i| > 1)
        For each v ∈ S_i
                V_m = MaxOfFrequency(v)
        Return V_m
End If
Return S_i
```

รูปที่ 3.19 อัลกอริทึมการคัดเลือกวิวข้อมูลที่ดีที่สุด (MvBestSelect)

3.3 การทำงานของโปรแกรม MV4QR

คำอธิบายในส่วนนี้เป็นการจำลองการทำงานของโปรแกรม MV4QR ที่ทำการพัฒนาขึ้นเพื่อสร้างความเข้าใจการทำงานของอัลกอริทึมด้วยตัวอย่างดังต่อไปนี้

ตัวอย่าง ต้องการสอบถามข้อมูลเกี่ยวกับร้านเช่าภาพยนตร์โดยมีข้อคำถาม Q1 ต้องการทราบถึงข้อมูลที่เป็น รหัสลูกค้า ชื่อลูกค้า หมายเลขโทรศัพท์ ที่อยู่ และชื่อภาพยนตร์ที่ลูกค้าเช่า ซึ่งสามารถแสดงได้ดังนี้

ข้อคำถามที่ 1

Q1(C_ID, C_NAME, C_PHONE, C_ADDRESS, MOVIE_TITLE)

วิวข้อมูลที่สามารถตอบข้อคำถามนี้ได้ก็อาจจะมีหลายวิวข้อมูล จากตัวอย่างข้อคำถามนี้จะสมมุติวิวข้อมูลที่สามารถตอบข้อคำถามได้ โดยดูได้จากวิวข้อมูลที่ 1

วิวข้อมูลที่ 1

MV1(C_ID, C_NAME, C_PHONE, C_ADDRESS)

วิวข้อมูลที่ 1 จะประกอบไปด้วยข้อมูลที่เป็น รหัสลูกค้า ชื่อลูกค้า หมายเลขโทรศัพท์ และที่อยู่ของลูกค้า วิวข้อมูลนี้สามารถตอบคำถามของข้อคำถาม Q1 ได้บางส่วน นั่นคือ สามารถตอบรหัสลูกค้า ชื่อลูกค้า หมายเลขโทรศัพท์ ที่อยู่ของลูกค้าได้ แต่ไม่สามารถตอบได้ว่าชื่อภาพยนตร์ที่ลูกค้าเช่าไปนั้นชื่ออะไร

วิวข้อมูลที่ 2

MV2(C_ID, PROVINCE_NAME, MOVIE_TITLE)

วิวข้อมูลที่ 2 จะประกอบไปด้วยข้อมูลที่เป็น รหัสลูกค้า จังหวัดที่ลูกค้าอาศัยอยู่ และชื่อภาพยนตร์ที่เช่าวิวข้อมูลนี้สามารถตอบคำถามของข้อคำถาม Q1 ได้บางส่วน นั่นคือ รหัสลูกค้า และชื่อภาพยนตร์ที่เช่า แต่ไม่สามารถตอบคำถามเกี่ยวกับชื่อลูกค้าและหมายเลขโทรศัพท์ได้

วิวข้อมูลที่ 3

MV3(C_ID, C_NAME, C_PHONE, C_ADDRESS, MOVIE_TITLE)

วิวข้อมูลที่ 3 จะประกอบไปด้วยข้อมูลที่เป็น รหัสลูกค้า ชื่อลูกค้า หมายเลขโทรศัพท์ ที่อยู่ และชื่อภาพยนตร์ที่ลูกค้าเช่า วิวข้อมูลนี้จะสามารถตอบคำถามของข้อคำถาม Q1 ได้ทั้งหมด

จะเห็นว่าวิวข้อมูลทั้งหมดนี้สามารถตอบคำถามของข้อคำถาม Q1 ได้ซึ่งวิวข้อมูลบางวิวข้อมูลจะสามารถตอบคำถามได้ทั้งหมดหรือบางส่วนก็ตาม ดังนั้นในขั้นตอนนี้จะคัดเลือกวิวข้อมูลทั้งสามนี้ออกมาเป็นเซตของวิวข้อมูลที่มีความสัมพันธ์กับข้อคำถาม

Q_i คือ ข้อคำถามที่ถูกแบ่งย่อยออกมาจากข้อคำถาม Q สามารถหา Q_i ได้ดังนี้

$Q1_1$ (C_ID)
$Q1_2$ (C_NAME)
$Q1_3$ (C_PHONE)
$Q1_4$ (C_ADDRESS)
$Q1_5$ (MOVIE_TITLE)

S_i คือ เซตของวิวข้อมูลที่สามารถตอบคำถามของ Q_i ได้เซตของวิวข้อมูลแสดงได้ดังนี้

{MV1, MV2, MV3}	เป็นเซตที่ตอบคำถามของ $Q1_1$ (C_ID) ได้
{MV1, MV3}	เป็นเซตที่ตอบคำถามของ $Q1_2$ (C_NAME) ได้
{MV1, MV3}	เป็นเซตที่ตอบคำถามของ $Q1_3$ (C_PHONE) ได้
{MV1, MV3}	เป็นเซตที่ตอบคำถามของ $Q1_4$ (C_ADDRESS) ได้
{MV2}	เป็นเซตที่ตอบคำถามของ $Q1_5$ (MOVIE_TITLE) ได้

จากตัวอย่างจะเห็นว่าวิวข้อมูลทั้งสามสามารถตอบคำถามได้ดังนี้ วิวข้อมูลที่ 1 สามารถบอก รหัสลูกค้า ชื่อลูกค้า หมายเลขโทรศัพท์ และที่อยู่ของลูกค้าได้ วิวข้อมูลที่ 2 สามารถบอกรหัสของลูกค้าและชื่อภาพยนตร์ได้ วิวข้อมูลที่ 3 สามารถบอก รหัสของลูกค้า ชื่อลูกค้า หมายเลขโทรศัพท์ ที่อยู่ของลูกค้า และชื่อภาพยนตร์ได้ จะเห็นว่าไม่จำเป็นต้องใช้วิวข้อมูลทั้งหมดก็สามารถตอบข้อคำถามได้ทั้งหมดแล้ว โดยการคัดเลือกวิวข้อมูลที่ตอบคำถามได้ดีที่สุดนั้น จะเป็นการเลือกวิวข้อมูลที่ตอบข้อคำถามได้ทั้งหมด หรือตอบได้มากที่สุดก่อนซึ่งสามารถหาได้จากอัลกอริทึมการคัดเลือกวิวข้อมูล

จากตัวอย่าง เมื่อนำอัลกอริทึมนี้มาใช้จะสามารถแสดงผลได้ดังนี้

ค่าความถี่ที่ปรากฏ

MV1 = 4
MV2 = 2
MV3 = 5

ค่าความถี่นี้จะเป็นตัวตัดสินว่าจะเลือกใช้วิวข้อมูลใดในการตอบข้อคำถาม Q_i โดยการคัดเลือกวิวข้อมูลที่มีค่าความถี่สูงสุด

$Q1_1$ (C_ID) วิวข้อมูลที่สามารถตอบข้อคำถามนี้ได้คือ {MV1,MV2,MV3} เลือก {MV3}
$Q1_2$ (C_NAME) วิวข้อมูลที่สามารถตอบข้อคำถามนี้ได้คือ {MV1,MV3} เลือก {MV3}
$Q1_3$ (C_PHONE) วิวข้อมูลที่สามารถตอบข้อคำถามนี้ได้คือ {MV1,MV3} เลือก {MV3}
$Q1_4$ (C_ADDRESS) วิวข้อมูลที่สามารถตอบข้อคำถามนี้ได้คือ {MV1,MV3} เลือก {MV3}
$Q1_5$ (MOVIE_TITLE) วิวข้อมูลที่สามารถตอบข้อคำถามนี้ได้คือ { MV2,MV3} เลือก {MV3}

M คือ เซตของวิวข้อมูลที่ตอบคำถามได้ดีที่สุด จากตัวอย่างวิวข้อมูลที่สามารถตอบคำถามได้ดีที่สุดจะได้

$$M = \{MV3\} \cup \{MV3\} \cup \{MV3\} \cup \{MV3\} \cup \{MV3\} = \{MV3\}$$

ดังนั้นวิวข้อมูลที่สามารถตอบข้อคำถาม Q1 ได้ดีที่สุดคือ {MV3}

3.4 แหล่งที่มาของข้อมูล

การทดสอบอัลกอริทึมที่พัฒนาขึ้นนี้ใช้ข้อมูลในการทดสอบทั้งสิ้นจำนวน 2 ชุด โดยชุดแรกสืบค้นมาจาก Transaction Processing Performance Council : TPC (http://www.tpc.org) ซึ่งได้เลือกชุดข้อมูล TPC-H มาใช้ในการทดสอบ ชุดที่สองเป็นชุดข้อมูล SH Schema ของ Oracle (http://www.oracle.com)

1) ชุดข้อมูล TPC-H

ส่วนประกอบของฐานข้อมูล TPC-H ถูกนิยามขึ้นโดยประกอบไปด้วยส่วนประกอบ 8 ส่วน ในแต่ละตาราง รายละเอียดของข้อมูลสามารถแสดงได้ในตารางที่ 3.4

ตารางที่ 3.4 แสดงรายละเอียดของข้อมูล TPC-H

Table name	Cardinality (in rows)	Length (in bytes) of typical row	Typical table size (in KB)
SUPPLIER	10,000	159	2
PART	200,000	155	30
PARTSUPP	800,000	144	110
CUSTOMER	150,000	179	26
ORDERS	1,500,000	104	149
LINEITEM	6,001,215	112	641
NATION	25	128	< 1
REGION	5	124	< 1
Total	8,661,245	1,105	956

2) SH Schema

เป็นข้อมูลที่เก็บสถิติการขายเพื่อที่จะใช้ในการช่วยสนับสนุนการตัดสินใจรายละเอียดของข้อมูลสามารถแสดงได้ในตารางที่ 3.5

ตารางที่ 3.5 แสดงรายละเอียดของข้อมูล SH Schema

Table name	Cardinality (in rows)	Length (in bytes) of typical row	Typical table size (in KB)
CHANNELS	5	40	0.045
COSTS	82112	26	82.138
COUNTRIES	23	64	0.087
CUSTOMERS	55500	180	55.68
PRODUCTS	72	170	0.242
PROMOTIONS	503	97	0.6
SALES	918843	29	918.872
TIMES	1826	197	2.023
Total	1058884	803	1059.687

3.5 วิธีการทดสอบ

ในวิธีการทดสอบนี้จะแบ่งออกเป็นสองส่วนดังนี้

ส่วนแรก จะเป็นการเตรียมข้อมูล การทดสอบในส่วนนี้จะทำโดยการเรียกใช้โปรแกรม MV4QR ที่พัฒนาขึ้นเพื่อเตรียมวิวข้อมูลไว้เพื่อการทดสอบการคัดเลือกวิวข้อมูล โดยโปรแกรมจะติดต่อกับฐานข้อมูลและทำการวิเคราะห์ข้อมูลในส่วนที่จะนำมาสร้างเป็นวิวข้อมูล ในส่วนนี้จำเป็นต้องมีการกำหนดค่าพารามิเตอร์เข้าไปด้วยดังนี้ ค่าขั้นต่ำของจำนวนการประมวลผลหรือการเข้าถึงวิวและจำนวนแถวของข้อมูลที่ได้จากผลลัพธ์ จากขั้นตอนนี้จะได้วิวข้อมูลที่เหมาะสมสำหรับการนำไปแปลงรูปแบบข้อคำถาม

ส่วนที่สอง จะเป็นการทดสอบประสิทธิภาพของโปรแกรม MV4QR ที่พัฒนาขึ้นโดยทำการป้อนข้อคำถามที่ต้องการเข้าไป โปรแกรมจะทำการคัดเลือกวิวข้อมูลที่สามารถตอบข้อคำถามโดยจะให้ข้อคำถามไปสอบถามข้อมูลจากวิวข้อมูลแทนการสอบถามกับตารางพื้นฐานโดยตรง ในงานวิจัยนี้ต้องการทดสอบประสิทธิภาพการทำงานของโปรแกรม MV4QR โดยวิเคราะห์จากประเด็นต่าง ๆ ดังนี้

1) เวลาที่ใช้ในการประมวลผลข้อคำถาม
2) ปริมาณของข้อมูลที่ใช้ในการทดสอบ
3) ความซับซ้อนของข้อคำถาม

บทที่ 4
ผลการวิเคราะห์ข้อมูลและการอภิปรายผล

การทดลองในการวิจัยครั้งนี้จะแบ่งออกเป็นสองส่วนคือ การเตรียมข้อมูลจะเป็นการวิเคราะห์ค่าทางสถิติของการใช้ข้อคำถามเพื่อนำมาสร้างเป็นวิวข้อมูลไว้สำหรับการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม และการทดสอบประสิทธิภาพของอัลกอริทึมที่ใช้ในการคัดเลือกวิวข้อมูล โดยใช้โปรแกรม MV4QR ที่พัฒนาขึ้นในการทดสอบ ในการทดลองนี้จะทำการศึกษาประสิทธิภาพและความเหมาะสมในการใช้วิวข้อมูลเพื่อช่วยในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม เกณฑ์ในการพิจารณาความเหมาะสมประกอบด้วย เวลาที่ใช้ในการประมวลผล ปริมาณของข้อมูลที่ใช้ในการทดสอบ และความซับซ้อนของข้อคำถาม การทดสอบทำการประมวลผลบนเครื่องคอมพิวเตอร์ Laptop Centrino Core 2 Duo 1.8 GHz หน่วยความจำหลัก 1 GB ฮาร์ดดิสก์ความจุ 160 GB บนระบบปฏิบัติการ Windows XP SP2

สำหรับเนื้อหาของบทนี้จะเป็นการนำเสนอผลการทดลองจากการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมที่พัฒนาขึ้นโดยใช้โปรแกรม MV4QR เปรียบเทียบกับการประมวลผลข้อคำถามแบบปกติหรือข้อคำถามดั้งเดิมที่ไม่ได้ทำการแปลงรูปแบบข้อคำถามโดยจะนำเสนอตามลำดับดังนี้ ผลการวิเคราะห์สถิติการใช้ข้อคำถามเพื่อเตรียมวิวข้อมูล, ผลการเปรียบเทียบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล TPC-H และผลการเปรียบเทียบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล SH Schema

4.1 ผลการวิเคราะห์สถิติการใช้ข้อคำถามเพื่อเตรียมวิวข้อมูล

เมื่อทำการทดสอบโปรแกรม MV4QR ให้ทำการสร้างวิวข้อมูลโปรแกรม MV4QR สามารถอ่านค่าสถิติของการใช้ข้อคำถามและสถิติของการใช้วิวออกมา เพื่อวิเคราะห์ตามเงื่อนไขที่กำหนดและสามารถนำข้อคำถามและวิวที่ผ่านเงื่อนไขมาสร้างเป็นวิวข้อมูลได้

4.2 ผลการทดสอบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล TPC-H

ตารางที่ 4.1 เป็นตารางแสดงข้อคำถามที่ใช้ในการทดสอบการใช้โปรแกรม MV4QR กับชุดข้อมูล TPC-H ซึ่งเป็นฐานข้อมูลขนาดใหญ่โดยหลักในการพิจารณาคัดเลือกข้อคำถามจะพิจารณา

จากลักษณะการประมวลผลดังนี้

1) ข้อคำถามที่มีฟังก์ชันการรวมแบบซับซ้อน (Q1-TPC)

2) ข้อคำถามที่มีการจอยน์ (Q2-TPC, Q3-TPC, Q4-TPC, Q5-TPC)

ในการทดสอบนี้จะเป็นการเปรียบเทียบเวลาระหว่างการประมวลผลข้อคำถามเดิมกับการประมวลผลข้อคำถามโดยมีการแปลงรูปแบบข้อคำถามแล้วโดยใช้โปรแกรม MV4QR ในการทดสอบส่วนนี้จะแบ่งผลการทดสอบออกเป็นสองกรณีดังนี้

1) โปรแกรม MV4QR สามารถแปลงรูปแบบข้อคำถามได้

2) โปรแกรม MV4QR ไม่สามารถแปลงรูปแบบข้อคำถามได้

ตารางที่ 4.1 ข้อคำถามที่ใช้ทดสอบกับชุดข้อมูล TPC-H

ชื่อข้อคำถาม	ข้อคำถาม
Q1-TPC	SELECT L_RETURNFLAG, L_LINESTATUS, SUM(L_QUANTITY) AS SUM_QTY, SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE, SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS SUM_DISC_PRICE, SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)*(1+L_TAX)) AS SUM_CHARGE, AVG(L_QUANTITY) AS AVG_QTY, AVG(L_EXTENDEDPRICE) AS AVG_PRICE, AVG(L_DISCOUNT) AS AVG_DISC, COUNT(*) AS COUNT_ORDER FROM LINEITEM WHERE L_SHIPDATE <= DATE '1998-12-01' - INTERVAL '90' DAY (3) GROUP BY L_RETURNFLAG, L_LINESTATUS ORDER BY L_RETURNFLAG, L_LINESTATUS;
Q2-TPC	SELECT CUSTOMER.C_CUSTKEY AS C_CUSTKEY, CUSTOMER.C_NAME AS C_NAME, CUSTOMER.C_ADDRESS AS C_ADDRESS, CUSTOMER.C_NATIONKEY AS C_NATIONKEY, CUSTOMER.C_PHONE AS C_PHONE, CUSTOMER.C_ACCTBAL AS C_ACCTBAL, CUSTOMER.C_MKTSEGMENT AS C_MKTSEGMENT, CUSTOMER.C_COMMENT AS C_COMMENT,

ตารางที่ 4.1 ข้อคำถามที่ใช้ทดสอบกับชุดข้อมูล TPC-H (ต่อ)

ชื่อข้อคำถาม	ข้อคำถาม
	REGION.R_NAME AS R_NAME FROM REGION REGION, NATION NATION, CUSTOMER CUSTOMER WHERE NATION.N_NATIONKEY=CUSTOMER.C_NATIONKEY AND REGION.R_REGIONKEY=NATION.N_NATIONKEY AND REGION.R_NAME ='ASIA'
Q3-TPC	SELECT AVG(CUSTOMER.C_ACCTBAL) AS C_ACCTBAL FROM REGION REGION, NATION NATION, CUSTOMER CUSTOMER WHERE CUSTOMER.C_NATIONKEY=NATION.N_NATIONKEY AND NATION.N_REGIONKEY=REGION.R_REGIONKEY AND REGION.R_NAME = 'ASIA'
Q4-TPC	SELECT CUSTOMER.C_CUSTKEY AS C_CUSTKEY, CUSTOMER.C_NAME AS C_NAME, CUSTOMER.C_ADDRESS AS C_ADDRESS, ORDERS.O_ORDERKEY AS O_ORDERKEY, ORDERS.O_ORDERSTATUS AS O_ORDERSTATUS, ORDERS.O_TOTALPRICE AS O_TOTALPRICE, ORDERS.O_ORDERDATE AS O_ORDERDATE, ORDERS.O_ORDERPRIORITY AS O_ORDERPRIORITY, ORDERS.O_CLERK AS O_CLERK, ORDERS.O_SHIPPRIORITY AS O_SHIPPRIORITY, CUSTOMER.C_ACCTBAL AS C_ACCTBAL FROM ORDERS ORDERS, CUSTOMER CUSTOMER WHERE CUSTOMER.C_CUSTKEY=ORDERS.O_ORDERKEY AND CUSTOMER.C_ACCTBAL <0
Q5-TPC	SELECT CUSTOMER.C_CUSTKEY AS C_CUSTKEY, CUSTOMER.C_NAME AS C_NAME, CUSTOMER.C_ADDRESS AS C_ADDRESS, ORDERS.O_ORDERKEY AS O_ORDERKEY,

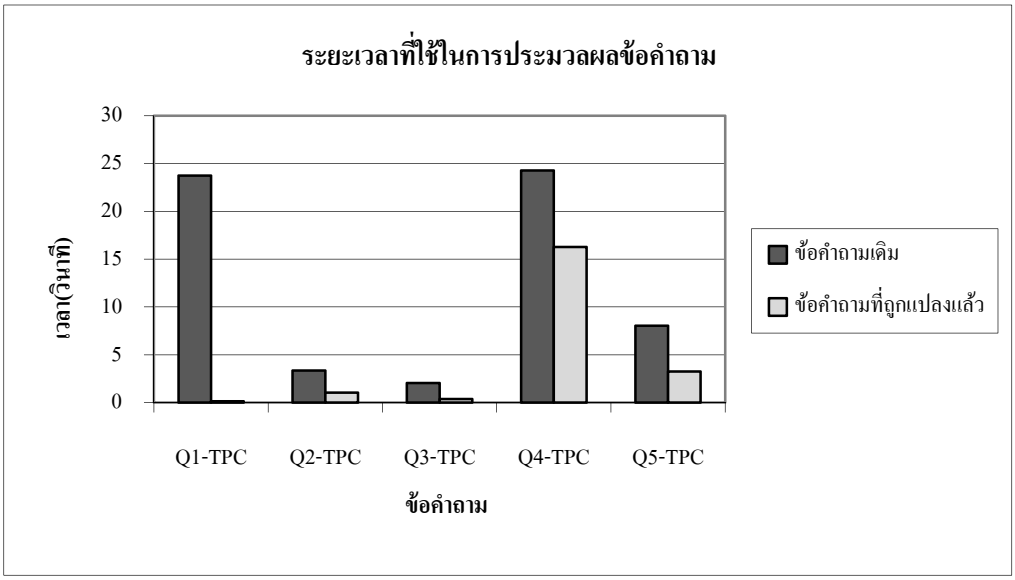
ตารางที่ 4.1 ข้อคำถามที่ใช้ทดสอบกับชุดข้อมูล TPC-H (ต่อ)

ชื่อข้อคำถาม	ข้อคำถาม
	ORDERS.O_ORDERSTATUS AS O_ORDERSTATUS, ORDERS.O_TOTALPRICE AS O_TOTALPRICE, ORDERS.O_ORDERDATE AS O_ORDERDATE, ORDERS.O_ORDERPRIORITY AS O_ORDERPRIORITY, ORDERS.O_CLERK AS O_CLERK, ORDERS.O_SHIPPRIORITY AS O_SHIPPRIORITY, CUSTOMER.C_ACCTBAL AS C_ACCTBAL FROM ORDERS ORDERS, CUSTOMER CUSTOMER WHERE CUSTOMER.C_CUSTKEY=ORDERS.O_ORDERKEY AND CUSTOMER.C_ACCTBAL <-900

<u>กรณีที่ 1</u> โปรแกรม MV4QR สามารถแปลงรูปแบบข้อคำถามได้ นั่นคือ ข้อคำถามที่ถูกป้อนเข้ามาไปตรงกับวิวข้อมูลที่ถูกสร้างไว้ ผลการเปรียบเทียบเวลาที่ใช้ประมวลผลข้อคำถามแสดงได้ดัง ตารางที่ 4.2 และ รูปที่ 4.1

ตารางที่ 4.2 ตารางผลการทดสอบการประมวลผลข้อคำถาม กรณีแปลงรูปแบบข้อคำถามได้

Query	Time(no mv) sec.	Time(use mv) sec.			Data (Rows)
		Pre-process	Query process	Total	
Q1-TPC	23.75	0.03	0.09	0.12	4
Q2-TPC	03.34	0.03	1.01	1.04	46
Q3-TPC	02.03	0.03	0.33	0.36	1
Q4-TPC	24.27	0.03	16.24	16.27	115460
Q5-TPC	8.05	0.03	3.21	3.24	23092

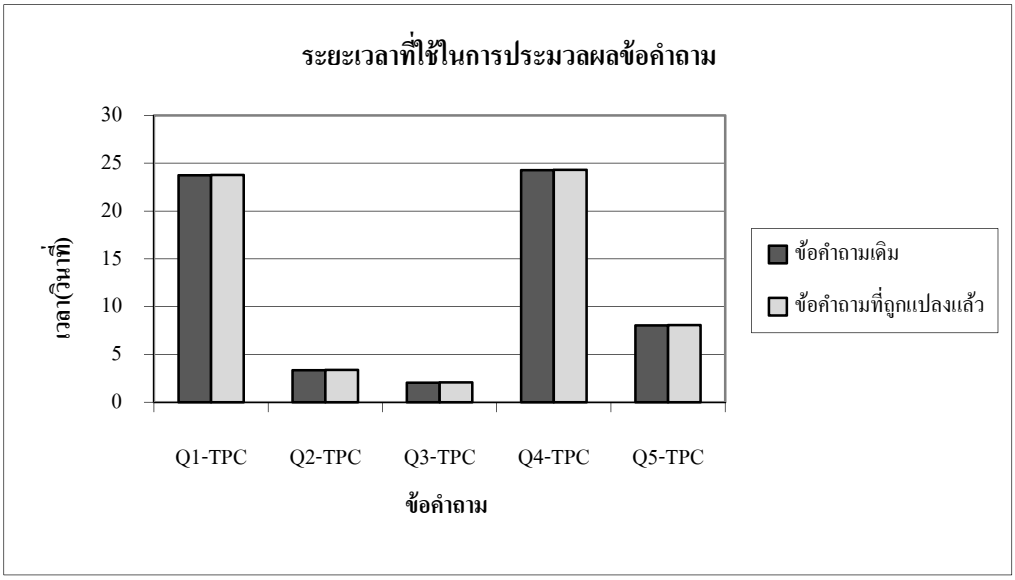


รูปที่ 4.1 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิมกับข้อคำถาม
ที่แปลงรูปแบบแล้วในกรณีที่สามารถแปลงรูปแบบข้อคำถามได้

<u>กรณีที่ 2</u> โปรแกรม MV4QR ไม่สามารถแปลงรูปแบบข้อคำถามได้ นั่นคือ ข้อคำถามที่ถูกป้อนเข้ามาไม่ตรงกับวิวข้อมูลที่ถูกสร้างไว้หรือวิวข้อมูลไม่ได้ถูกสร้างเก็บไว้ ผลการเปรียบเทียบเวลาที่ใช้ประมวลผลข้อคำถามแสดงได้ดัง ตารางที่ 4.3 และ รูปที่ 4.2

ตารางที่ 4.3 ตารางผลการทดสอบการประมวลผลข้อคำถาม กรณีไม่สามารถแปลงข้อคำถามได้

Query	Time(no mv) sec.	Time(use mv) sec.			Data (Rows)
		Pre-process	Query process	Total	
Q1-TPC	23.75	00.03	23.75	23.78	4
Q2-TPC	03.34	00.03	03.34	3.37	7268
Q3-TPC	02.03	00.03	02.03	2.06	1
Q4-TPC	24.27	00.03	24.27	24.3	115460
Q5-TPC	8.05	00.03	8.05	8.08	23092



รูปที่ 4.2 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิมกับข้อคำถามที่แปลงรูปแบบแล้วในกรณีที่ไม่สามารถแปลงรูปแบบข้อคำถามได้

จากผลการทดสอบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล TPC-H จะเห็นว่าในกรณีที่ข้อคำถามที่ถูกป้อนเข้ามาตรงกับวิวข้อมูลโปรแกรม MV4QR สามารถแปลงรูปแบบข้อคำถามได้ เมื่อพิจารณาผลต่างของเวลาจะเห็นว่าเวลาที่ใช้ในการประมวลผล ข้อคำถามที่ถูกแปลงรูปแบบแล้วนั้นน้อยกว่าเวลาที่ใช้ในการประมวลผลข้อคำถามเดิมมาก ในกรณีที่ข้อคำถามที่ถูกป้อนเข้ามาไม่ตรงกับวิวข้อมูล โปรแกรมจะไม่ทำการแปลงรูปแบบข้อคำถามได้และยังต้องสูญเสียเวลาไปในการตรวจสอบข้อคำถามกับโครงสร้างวิวข้อมูล จึงทำให้เวลาที่ใช้ในการประมวลผลข้อคำถามเดิมเร็วกว่าการประมวลผลข้อคำถามที่ผ่านการทำงานของโปรแกรม MV4QR แต่อย่างไรก็ตามเวลาที่สูญเสียไปเมื่อเทียบกับเวลาที่ใช้ในการประมวลผลข้อคำถามเป็นเวลาเพียงเล็กน้อยเท่านั้น พิจารณาเวลาที่สูญเสียไปเมื่อเทียบประสิทธิภาพในการประมวลผลข้อคำถามจะเห็นว่าโปรแกรม MV4QR สามารถเพิ่มประสิทธิภาพการประมวลผลข้อคำถามได้ดี

4.3 ผลการทดสอบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล SH Schema

ตารางที่ 4.4 เป็นตารางแสดงข้อคำถามที่ใช้ในการทดสอบการใช้โปรแกรม MV4QR กับชุดข้อมูล SH Schema ซึ่งเป็นฐานข้อมูลขนาดเล็ก ในการทดสอบนี้จะเป็นการเปรียบเทียบเวลาระหว่างการประมวลผลข้อคำถามเดิมกับการประมวลผลข้อคำถามโดยมีการแปลงรูปแบบข้อคำถามแล้ว โดยใช้โปรแกรม MV4QR

หลักในการพิจารณาคัดเลือกข้อคำถามจะพิจารณาจากลักษณะการประมวลผลดังนี้

1) ข้อคำถามที่มีฟังก์ชันการรวมแบบซับซ้อน (Q2-SH)
2) ข้อคำถามที่มีการจอยน์ (Q1-SH, Q2-SH, Q3-SH, Q5-SH)
3) ข้อคำถามที่มีการจอยน์และมีฟังก์ชันการรวม (Q1-SH, Q2-SH, Q3-S)

ในการทดสอบส่วนนี้จะแบ่งผลการทดสอบออกเป็นสองกรณีดังนี้

1) โปรแกรม MV4QR สามารถแปลงรูปแบบข้อคำถามได้
2) โปรแกรม MV4QR ไม่สามารถแปลงรูปแบบข้อคำถามได้

ตารางที่ 4.4 ตารางข้อคำถามที่ทดสอบกับชุดข้อมูล SH Schema

ชื่อข้อคำถาม	ข้อคำถาม
Q1-SH	SELECT T.CALENDAR_MONTH_DESC , SUM(S.AMOUNT_SOLD) AS DOLLARS FROM SH.SALES S, SH.TIMES T WHERE S.TIME_ID = T.TIME_ID GROUP BY T.CALENDAR_MONTH_DESC
Q2-SH	SELECT T.WEEK_ENDING_DAY, P.PROD_SUBCATEGORY, SUM(S.AMOUNT_SOLD) AS DOLLARS, S.CHANNEL_ID, S.PROMO_ID FROM SH.SALES S, SH.TIMES T, SH.PRODUCTS P WHERE S.TIME_ID = T.TIME_ID AND S.PROD_ID = P.PROD_ID GROUP BY T.WEEK_ENDING_DAY, P.PROD_SUBCATEGORY, S.CHANNEL_ID, S.PROMO_ID
Q3-SH	SELECT V.CUST_FIRST_NAME, SUM(S.QUANTITY_SOLD) FROM SH.CUST_AMOUNT_PAY_V V,SH.CHANNELS C,SH.SALES S WHERE V.CUST_ID=S.CUST_ID AND C.CHANNEL_ID = S.CHANNEL_ID AND C.CHANNEL_CLASS='DIRECT' GROUP BY V.CUST_FIRST_NAME

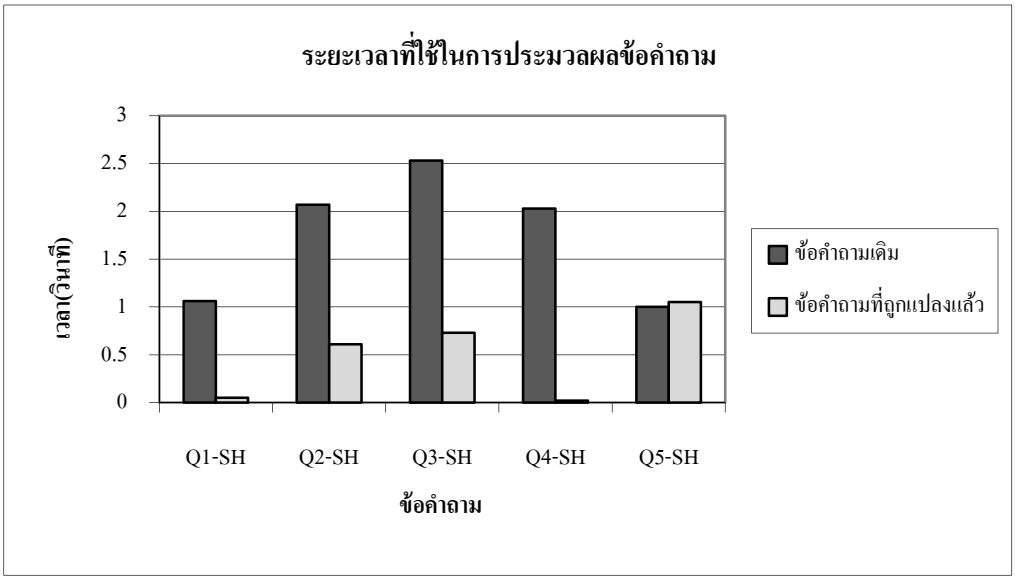
ตารางที่ 4.4 ตารางข้อคำถามที่ทดสอบกับชุดข้อมูล SH Schema (ต่อ)

ชื่อข้อคำถาม	ข้อคำถาม
Q4-SH	SELECT V.CUST_ID, V.CUST_FIRST_NAME, V.CUST_LAST_NAME, V.CUST_EMAIL,DOLLARS FROM SH.CUST_AMOUNT_PAY_V V WHERE V.DOLLARS>80000
Q5-SH	SELECT S.PROD_ID, S.CUST_ID, S.TIME_ID, S.QUANTITY_SOLD, C.CHANNEL_ID,C.CHANNEL_DESC FROM SH.CHANNELS C, SH.SALES S WHERE C.CHANNEL_ID = S.CHANNEL_ID AND C. CHANNEL_ID=9

กรณีที่ 1 โปรแกรม MV4QR สามารถแปลงรูปแบบข้อคำถามได้ นั่นคือ ข้อคำถามที่ถูกป้อนเข้ามาไปตรงกับวิวข้อมูลที่ถูกสร้างไว้ ผลการเปรียบเทียบเวลาที่ใช้ประมวลผลข้อคำถามแสดงได้ดังตารางที่ 4.5 และ รูปที่ 4.3

ตารางที่ 4.5 ตารางผลการทดสอบการประมวลผลข้อคำถามกรณีตรงกับวิวข้อมูล

Query	Time(no mv) sec.	Time(use mv) sec.			Data (Rows)
		Pre-process	Query process	Total	
Q1-SH	01.06	0.01	00.04	00.05	48
Q2-SH	02.07	0.01	00.60	00.61	11266
Q3-SH	02.53	0.03	00.70	00.73	1084
Q4-SH	02.03	0.01	00.01	00.02	30
Q5-SH	01.00	0.03	01.02	01.05	2074

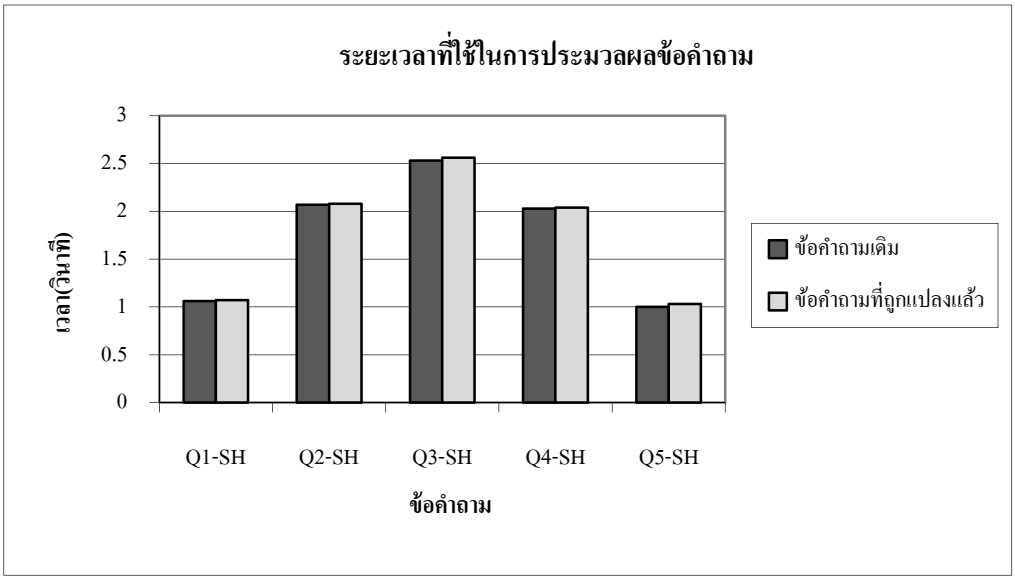


รูปที่ 4.3 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิมกับข้อคำถามที่แปลงรูปแบบแล้ว (ข้อมูล SH)

<u>กรณีที่ 2</u> โปรแกรม MV4QR สามารถแปลงรูปแบบข้อคำถามได้ นั่นคือ ข้อคำถามที่ถูกป้อนเข้ามาไม่ตรงกับวิวข้อมูลที่ถูกสร้างไว้ ผลการเปรียบเทียบเวลาที่ใช้ประมวลผลข้อคำถามแสดงได้ดัง ตารางที่ 4.6 และ รูปที่ 4.4

ตารางที่ 4.6 ตารางผลการทดสอบการประมวลผลข้อคำถาม กรณีไม่ตรงกับวิวข้อมูล

Query	Time(no mv) sec.	Time(use mv) sec.			Data (Rows)
		Pre-process	Query process	Total	
Q1-SH	01.06	00.01	01.06	01.07	48
Q2-SH	02.07	00.01	02.07	02.08	11266
Q3-SH	02.53	00.03	02.53	02.56	1084
Q4-SH	02.03	00.01	02.03	02.04	30
Q5-SH	01.00	00.03	01.00	01.03	2074



รูปที่ 4.4 กราฟเปรียบเทียบเวลาที่ใช้ในการประมวลผลข้อคำถามระหว่างข้อคำถามเดิมกับข้อคำถามที่แปลงรูปแบบแล้วในกรณีที่ไม่สามารถแปลงรูปแบบข้อคำถามได้ (ข้อมูล SH)

จากผลการทดสอบประสิทธิภาพการประมวลผลข้อคำถามโดยทดสอบกับชุดข้อมูล SH Schema จะเห็นว่าในกรณีที่ข้อคำถามที่ถูกป้อนเข้ามาตรงกับวิวข้อมูลโปรแกรม MV4QR สามารถแปลงรูปแบบข้อคำถามได้ เมื่อพิจารณาผลต่างของเวลาจะเห็นว่าเวลาที่ใช้ในการประมวลผลข้อคำถามที่ถูกแปลงรูปแบบแล้วนั้นน้อยกว่าเวลาที่ใช้ในการประมวลผลข้อคำถามเดิมเล็กน้อย ในกรณีที่ข้อคำถามที่ถูกป้อนเข้ามาไม่ตรงกับวิวข้อมูลโปรแกรมจะไม่ทำการแปลงรูปแบบข้อคำถามได้ และยังต้องสูญเสียเวลาไปในการตรวจสอบข้อคำถามกับโครงสร้างวิวข้อมูล จึงทำให้เวลาที่ใช้ในการประมวลผลข้อคำถามเดิมจึงเร็วกว่าการประมวลผลข้อคำถามที่ผ่านการทำงานของโปรแกรม MV4QR พิจารณาเวลาที่สูญเสียไปเมื่อเทียบประสิทธิภาพในการประมวลผลข้อคำถาม จะเห็นว่าโปรแกรม MV4QR สามารถเพิ่มประสิทธิภาพการประมวลผลข้อคำถามได้เพียงเล็กน้อยเมื่อทำงานกับชุดข้อมูลขนาดเล็ก

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

การเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลเป็นการนำวิวข้อมูลเข้ามาใช้เพื่อช่วยในการตอบข้อคำถามโดยจะพยายามลดระยะเวลาที่ใช้ในการเข้าถึงข้อมูลเมื่อมีการออกแบบระบบการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม สิ่งที่สำคัญที่สุดคือ การพยายามหาวิธีที่จะช่วยลดเวลาที่ใช้ในการตอบสนองต่อข้อคำถามให้ได้มากที่สุด โดยปกติแล้วคำจำกัดความหรือข้อกำหนดในการใช้วิวข้อมูลอย่างมีประสิทธิภาพและการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามจะกล่าวอ้างถึงปัญหาการคัดเลือกวิวข้อมูลอยู่เสมอ ดังนั้นการสร้างและการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถามถือว่าเป็นขั้นตอนที่สำคัญในกระบวนการเพิ่มประสิทธิภาพ การประมวลผลข้อคำถามด้วยวิวข้อมูล

งานวิจัยนี้มีจุดมุ่งหมายที่จะพัฒนาแนวทางในการเพิ่มประสิทธิการประมวลผลข้อคำถามด้วยวิวข้อมูล โดยจะเสนอแนวคิดและอัลกอริทึมในการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถามและแนวทางในการนำข้อมูลทางสถิติมาช่วยในการสร้างวิวข้อมูลเพื่อให้ได้วิวข้อมูลที่สามารถนำมาใช้ประโยชน์ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามได้อย่างเต็มที่

ขั้นตอนการดำเนินงานวิจัยแบ่งออกเป็น การพัฒนาวิธีการพิจารณาข้อมูลเพื่อที่จะนำมาสร้างเป็นวิวข้อมูลและอัลกอริทึมที่ใช้ในการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม (Query rewriting) โปรแกรมที่พัฒนาขึ้นใช้ภาษาจาวา (Java) ในการพัฒนา โดยมุ่งเน้นให้มีวิธีการสร้างวิวข้อมูลที่เกิดประโยชน์โดยจะพิจารณาจากสถิติการเข้าใช้วิว หรือข้อคำถามที่ถูกใช้บ่อย ๆ และอัลกอริทึมสามารถคัดเลือกวิวข้อมูลที่จะนำมาแปลงรูปแบบข้อคำถามเพื่อเพิ่มประสิทธิภาพการประมวลผลข้อคำถามโดยเฉพาะข้อคำถามที่ถูกใช้บ่อย ๆ ได้อย่างเหมาะสม

ขั้นตอนการทดสอบประสิทธิภาพอัลกอริทึมที่พัฒนาขึ้นนี้ใช้ข้อมูลในการทดสอบทั้งสิ้นจำนวน 2 ชุด โดยชุดแรกสืบค้นมาจาก Transaction Processing Performance Council : TPC (http://www.tpc.org) ซึ่งได้เลือกชุดข้อมูล TPC-H มาใช้ในการทดสอบ ชุดที่สองเป็นชุดข้อมูล SH Schema ของ Oracle (http://www.oracle.com) จากนั้นเปรียบเทียบผลที่ได้โดยพิจารณาจากระยะเวลาที่ใช้ในการประมวลผลเปรียบเทียบกันระหว่างข้อคำถามดั้งเดิมที่สอบถามข้อมูลจากตารางพื้นฐานกับข้อคำถามที่ถูกแปลงแล้ว ซึ่งเวลาที่ใช้ในการประมวลข้อคำถามที่ถูกแปลงเป็นเวลาที่ได้รวมเวลาในการแปลงรูปแบบข้อคำถามแล้ว

5.1 สรุปผลการวิจัย

ผลการทดสอบสรุปได้ดังนี้

1) การทดสอบโปรแกรม MV4QR ให้ทำการสร้างวิวข้อมูล โปรแกรม MV4QR สามารถอ่านค่าสถิติของการใช้ข้อคำถามและสถิติของการใช้วิวออกมาเพื่อวิเคราะห์ตามเงื่อนไขที่กำหนด และสามารถนำข้อคำถามและวิวที่ผ่านเงื่อนไขมาสร้างเป็นวิวข้อมูลได้

2) การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดเล็กและป้อนข้อคำถามตรงกับวิวข้อมูลที่มีอยู่ โปรแกรมสามารถทำงานได้ดีและเห็นผลต่างเพียงเล็กน้อยเท่านั้นระหว่างข้อคำถามดั้งเดิมกับข้อคำถามที่ถูกปรับปรุงโดยโปรแกรม MV4QR โดยข้อคำถามที่ถูกปรับปรุงแล้วจะใช้เวลาในการประมวลผลน้อยกว่าข้อคำถามดั้งเดิม

3) การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดเล็กและป้อนข้อคำถามที่ไม่ตรงกับวิวข้อมูลที่มีอยู่ โปรแกรมสามารถทำงานได้ดี ในกรณีนี้ข้อคำถามจะไม่ถูกแปลงรูปแบบดังนั้นเวลาที่ใช้ในการประมวลผลข้อคำถามจึงเท่ากันแต่เมื่อรวมเวลาที่ใช้ในกระบวนการตรวจเช็คผลต่างระหว่างข้อคำถามดั้งเดิมกับข้อคำถามที่ได้จากโปรแกรมต่างกันไม่มากนัก แต่เมื่อพิจารณาประสิทธิภาพที่ได้มากับเวลาที่ต้องเสียไปยังไม่เห็นผลที่ชัดเจนของการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามโดยใช้โปรแกรม MV4QR

4) การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดใหญ่และป้อนข้อคำถามตรงกับวิวข้อมูลที่มีอยู่ โปรแกรมสามารถทำงานได้ดีและเห็นผลต่างอย่างชัดเจนระหว่างข้อคำถามดั้งเดิมกับข้อคำถามที่ถูกปรับปรุงโดยโปรแกรม MV4QR โดยข้อคำถามที่ถูกปรับปรุงแล้วจะใช้เวลาในการประมวลผลน้อยกว่าข้อคำถามดั้งเดิม นั่นคือโปรแกรมนี้สามารถเพิ่มประสิทธิภาพการประมวลผลข้อคำถามได้ดี เวลาที่ใช้ในการประมวลผลข้อคำถามที่ถูกแปลงแล้วที่ใช้ทดสอบแต่ละข้อคำถามจะต่างกันไม่มากนักเมื่อเปรียบเทียบกับข้อคำถามที่ถูกแปลงแล้วเช่นกัน เวลาที่ต่างกันไปนั้นจะขึ้นอยู่กับจำนวนของข้อมูล

5) การทดสอบโปรแกรม MV4QR กับฐานข้อมูลขนาดใหญ่และป้อนข้อคำถามที่ไม่ตรงกับวิวข้อมูลที่มีอยู่ โปรแกรมสามารถทำงานได้ดี ในกรณีนี้ข้อคำถามจะไม่ถูกแปลงรูปแบบดังนั้นเวลาที่ใช้ในการประมวลผลข้อคำถามจึงเท่ากันแต่เมื่อรวมเวลาที่ใช้ในกระบวนการตรวจเช็คผลต่างระหว่างข้อคำถามดั้งเดิมกับข้อคำถามที่ได้จากโปรแกรมต่างกันไม่มากนัก โดยข้อคำถามที่ถูกแปลงรูปแบบจะใช้เวลามากกว่าข้อคำถามดั้งเดิม พิจารณาเวลาที่สูญเสียไปในการเปรียบเทียบกับเวลาที่ใช้ในการประมวลผลจะเห็นว่าเป็นเวลาที่เล็กน้อยเท่านั้น

5.2 การประยุกต์งานวิจัย

โปรแกรม MV4QR ที่ทำการพัฒนาขึ้นสามารถนำมาใช้ในการสร้างและการคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม ซึ่งเป็นขั้นตอนสำคัญในกระบวนการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล ในกรณีที่มีการใช้ข้อคำถามที่ซับซ้อน เช่น ข้อคำถามที่เป็นฟังก์ชันการรวมหรือข้อคำถามที่สอบถามข้อมูลจากหลายตาราง สามารถใช้โปรแกรม MV4QR เข้ามาช่วยในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามได้

5.3 ข้อเสนอแนะ

ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูล การสร้างและคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม เป็นกระบวนการที่สำคัญที่ทำให้การนำวิวข้อมูลมาช่วยในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามเกิดประสิทธิภาพสูงสุด โดยโปรแกรม MV4QR สามารถสร้างวิวข้อมูลจากข้อคำถามที่ถูกใช้บ่อย สามารถคัดเลือกวิวข้อมูลขึ้นมาเพื่อช่วยตอบข้อคำถามแทนตารางพื้นฐาน และสามารถแปลงรูปแบบข้อคำถามได้ ดังนั้นโปรแกรม MV4QR ที่พัฒนาขึ้นมานี้น่าจะเป็นประโยชน์แก่นักวิจัยท่านอื่นที่สนใจที่จะพัฒนางานทางด้านการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลต่อไปดังนี้

1) การวิจัยเพื่อเพิ่มประสิทธิภาพของโปรแกรม MV4QR โดยการปรับปรุงเทคนิคที่ใช้ในการพิจารณาข้อคำถามเพื่อนำมาสร้างวิวข้อมูล สามารถนำความซับซ้อนของข้อคำถามหรือแผนการประมวลผลมาช่วยในการตัดสินใจเพื่อให้ได้วิวข้อมูลที่สามารถนำไปใช้ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามได้ดียิ่งขึ้น

2) โปรแกรม MV4QR ที่ทำการพัฒนาขึ้นยังมีข้อจำกัดในการแปลงรูปแบบข้อคำถามในการทดสอบการทำงานของโปรแกรม MV4QR ยังพบปัญหาในการแปลความหมายบางคำสั่ง เช่น BETWEEN นั่นคือสามารถเพิ่มประสิทธิภาพในการแปลความหมายและการแปลงรูปแบบข้อคำถามให้โปรแกรม MV4QR ทำงานได้อย่างมีประสิทธิภาพมากขึ้น

3) สามารถพัฒนาอัลกอริทึมที่ใช้สำหรับในการคัดเลือกวิวข้อมูลเพื่อเพิ่มประสิทธิภาพการทำงานของโปรแกรม MV4QR

4) การพัฒนาโปรแกรม MV4QR เพิ่มเติมเพื่อนำไปสู่โปรแกรมที่สามารถทำการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลที่ทำงานแบบอัตโนมัติซึ่งอยู่บนระบบจัดการฐานข้อมูลได้

รายการอ้างอิง

Chang, J. -Y. and Lee, S. -G. (1998). Query reformulation using materialized views in data warehousing environment. In **Proceedings of the 1st International Workshop on Data Warehousing and OLAP (DOLAP)** : 54-59.

Chaudhuri, S., Krishnamurthy, R., Potamianos, S. and Shim, K. (1995). Optimizing queries with materialized views. In **Proceedings of the 11th International Conference on Data Engineering (ICDE)** : 77-92.

Chaudhuri, S., Narasayya, V., and Agrawal, S. (2000). Automated selection of materialized views and indexes for SQL databases. In **Proceedings of 26th International Conference on Very Lage Data Base** : 496-505.

Clement, T. Y. And Meng, Y. (1997). **Principles of Database Query Processing for Advanced Applications.** Morgan Kaufmann.

Goldstein, J. and Larson, P. (2001). Optimizing queries using materialized views: A practical, scalable solution, In **Proceedings of ACM SIGMOD International Conference on Management of Data** : 331-342.

Gupta, A., Harinarayan, V., and Quass, D. (1995). Aggregate-query processing in data warehousing environments. In **Proceedings of the 21th International Conference on Very Large Data Bases (VLDB)** : 358-369.

Gupta, A., and Mumick, I. S. (1998). Maintenance of materialized views: Problem, techniques, and application. In A. Gupta and I. S. Mumick, Eds., **Materialized Views** : 145-158.

Hobbs, L. (2005). **Oracle Materialized Views & Query Rewrite**. Oracle Corporation.

Mistry, H., Prasan, R., Ramamritham, S. (2001). Materialized view selection and maintenance using multi-query optimization**. In Proceedings of ACM SIGMOD International Conference on Management of Data** : 307-318.

Lane, P. (2003). **Oracle Database Data Warehousing Guide, 10g Release 1(10.1).** Oracle Corporation.

Pottinger, R., and Levy, A. Y. (2000). A scalable algorithm for answering queries using views. In **Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)** : 182-198.

Urbano, R. (2003). **Oracle Database Advanced Replication, 10g Release 1 (10.1).** Oracle Corporation.

Zhengxin, C. (2001). **Intelligent Data Warehousing: From Data Preparation to Data Mining,** CRC PRESS.

ภาคผนวก ก

บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 32

การคัดเลือกวิวข้อมูลเพื่อแปลงรูปแบบข้อคำถาม

MATERIALIZED VIEW SELECTION FOR QUERY REWRITING

จักรพันธ์ มหาวันตัง, นิตยา เกิดประสพ และ กิตติศักดิ์ เกิดประสพ

Jackapan Mahavantang, Nittaya Kerdprasop and Kittisak Kerdprasop
School of Computer Engineering, Suranaree University of Technology, Muang, Nakhon Ratchasima, 30000.
Email:jackapan@yahoo.com

บทคัดย่อ: ฐานข้อมูลในปัจจุบันเต็มไปด้วยข้อมูลมากมายทั้งที่เกิดประโยชน์ และไม่เกิดประโยชน์ จึงเกิดการขุดค้นข้อมูลเพื่อหาความรู้จากข้อมูลเหล่านั้น อย่างไรก็ตาม ในการขุดค้นข้อมูลยังมีปัญหาเรื่องการประมวลผลข้อคำถามที่ต้องใช้เวลานาน จึงได้มีความพยายามที่จะเพิ่มประสิทธิภาพของ การประมวลผลข้อคำถามด้วยวิธีการต่าง ๆ วิวข้อมูลได้ถูกนำมาใช้ในการเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม โดยนักวิจัยได้พยายามหาวิธีใช้ประโยชน์จากวิวข้อมูลมาเป็นระยะเวลานาน แต่วิธีการต่าง ๆ ที่เสนอยังมีข้อจำกัดที่วิวข้อมูลจะต้องตรงพอดีกับเงื่อนไขในข้อคำถาม แนวทางการวิจัยของโครงการวิจัยนี้พยายามลดข้อจำกัดดังกล่าว โดยเสนอการสร้างวิวข้อมูลและพยายามหาเกณฑ์คัดเลือกวิวเพื่อพิจารณาวิวข้อมูลที่มีความใกล้เคียงกับข้อคำถามมากที่สุด เพื่อแปลงรูปแบบข้อคำถามให้สามารถประมวลผลได้เร็วที่สุด โดยยังคงความถูกต้องของผลลัพธ์ใน การตอบข้อคำถาม

Abstract: Modern database contains a wealth of information waiting to be discovered and understood. However, finding and presenting this information in a timely fashion can be a major issue, especially when vast amount of data have to be searched. Materialized views help solve this problem. To realize this potential, the query optimizer should know how and when to exploit materialized views. This paper presents algorithm for determining whether part or all of a query can be estimated from materialized views and describes how it can be combined to rewrite query.

Introduction: Materialized views can provide massive improvements in query processing time, especially for aggregation queries over large data [2]. The materialized view should be thought of as a special kind of view, which physically exists inside in the database [3, 4]. We can improve query execution time by pre-computing expensive joins and aggregation operation prior to execution [6]. Then create a materialized view as a new physical table which consists of pre-computed data that are much smaller in size and able to answer the query rapidly [4]. Compared to the original source the need of physical space is very low, but the increased speed of the answer is substantial. For example, given a database containing a customer relation customer (C_ID, C_NAME, C_PHONE, PROVINCE_ID) and a province relation province (PROVINCE_ID, PROVINCE_NAME). Let c_korat_mv be a materialized view that contains all customers who live in nakhonratchasima (PROVINCE_NAME = nakhonratchasima). Consider the query that asks for customer whose name is jirawan and live in nakhonratchasima. We present accessed data path on a base table, compare to accessed data path on materialized view as follows:

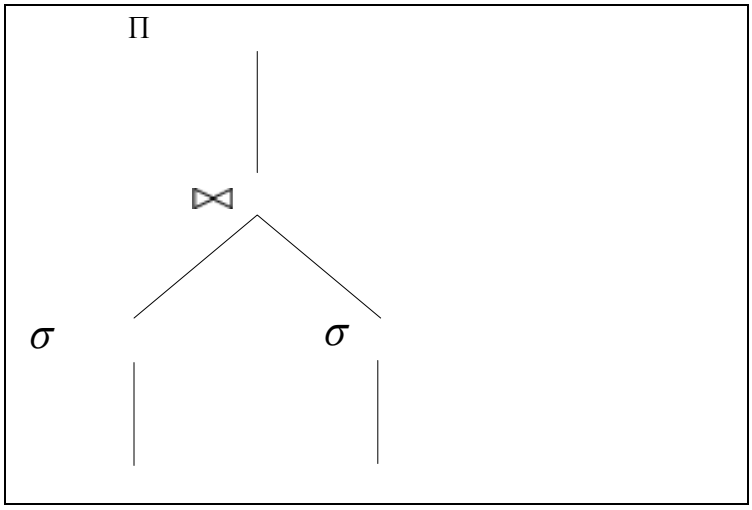


Figure 1 Access data path on base table **.**

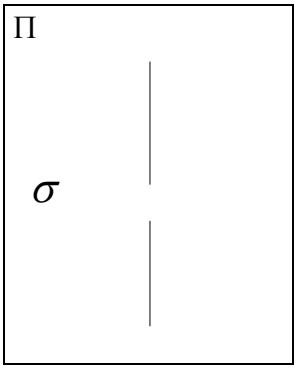


Figure 2 Access data path on materialized view.

In this example, execution time for accessing data on the materialized view is better than accessing data a base table. That means materialized views have been found to be very effective at speeding up query answering [5]. Essential for materialized view usage is view selection. The conceptual idea is selecting views that match the query′s constraints as much as possible. According to our sample, a view to be selected is MV1. Therefore, we use this view first, thin apply view MV2.

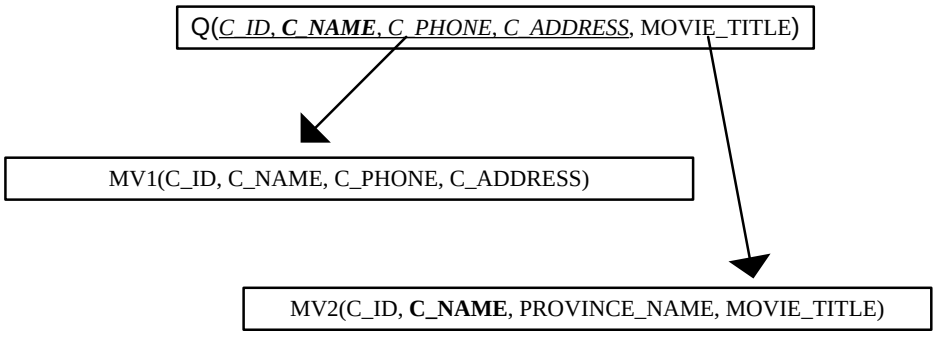


Figure 3 Materialized view selecting example.

Methodology: We use the basic architecture for the automated selection process as proposed in [1] and extend the algorithm for materialized view selection described step by step as follows:

Step1. Select the materialized view that correlated with the query. Materialized view which is correlated with the query can give the desired answer. For example, given query Q1 (C_ID, C_NAME, C_PHONE, C_ADDRESS, MOVIE_TITLE), there are chances that several materialized views can be applied. Assume MV1 (C_ID, C_NAME, C_PHONE, C_ADDRESS),

MV2 (C_ID, C_NAME, PROVINCE_NAME, MOVIE_TITLE), MV3 (C_ID, C_NAME, C_PHONE, C_ADDRESS, MOVIE_TITLE) are set of views that can give the answer for Q1.

Step2. Select the materialized views which can give the best answer. From step 1 we will get the answers which correlated with query but there are the overlapping data in each answer then in this step we will select the materialized view that can give the best answer. We've get a set of data from this step such as set 1, the materialized view indicates ID code, phone number and address. Set 2, the materialized view indicates ID code and name of the movie and also set 3, the materialized view indicates ID code, phone number, address and name of the movie. The example showed that we don't need to materialized views all the data but we can get the best answer for all queries by selecting the materialized views which can give the best answer. In this step we took advantage of mvBestselect algorithm and Cost-based pruning of syntactically relevant materialized views algorithm [1] present as figure4.

```
M = {} /* M is the set of materialized views that is useful for at least one
query in workload W */
For i = 1 to |W|
        Let Si = Set of materialized views proposed for Qi
        C = MvBestSelect(Qi,Si)
        M = M U C
End For
Return M
```

Figure 4 Cost-based pruning of syntactically relevant materialized views.

mvBestSelect is the algorithm that is use for selecting the materialized view that can give the best answer and used with Cost-based pruning of syntactically relevant materialized views algorithm.

```
Let Si = Set of materialized views proposed for Qi
If (|Si| > 1)
        For each v   Si
                Vm = MaxOfFequency(v)
        Return Vm
End If
Return Si
```

Figure 5 mvBestSelect algorithm.

Qi is the divided query and v is the member of set of the materialized view that can answer the query of Qi, Si is a set of the materialized view which can answer the query of Qi for example:

{MV1, MV2, MV3} is a set that can answer the query of $Q1_1$ (C_ID)

{MV1, MV2, MV3} is a set that can answer the query of $Q1_2$ (C_NAME)

{MV1, MV3} is a set that can answer the query of $Q1_3$ (C_PHONE)

{MV1, MV3} is a set that can answer the query of $Q1_4$ (C_ADDRESS)

{MV3} is a set that can answer the query of $Q1_5$ (MOVIE_TITLE)

Then, M is the materialized view that can give the best answer and M = {MV3}

Results, Discussion and Conclusion: The experiments were run on Pentium4 with CPU speed 3.2 GHz and 512 MB RAM. The databases used for our tests were stored on an internal 80GB hard drive.

Databases: The algorithms presented in this paper have been extensively tested on Oracle Database Sample Schemas 10g [6] (sales history (SH) schema) 0.893 GB database size and 128 MB Table spaces size.

The result (as shows in figure 6) reveals that response time of new query is better than original one. This process decreases compute and joins data, and thus shows the better performance.

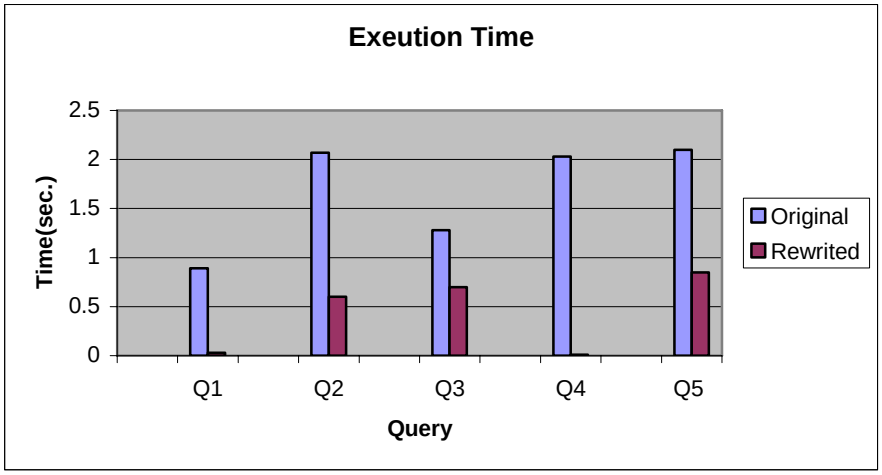


Figure 6 Execution time.

Materialized views can provide massive improvements in query processing time, especially for aggregation queries over large data. Essential for materialized view usage is views selection and physical design.

References:

1. Chaudhuri, S., Narasayya V., Agrawal S., Automated selection of materialized views and indexes for SQL databases, *Proc. 26th Int. Conf. Very Large Databases*, 496-505, 2000.
2. Goldstein, J. and Larson, P., Optimizing queries using materialized views: A practical, scalable polution, *Proc. ACM SIGMOD,* 331-342, 2001.
3. Gupta, A., and Mumick, I. S., Maintenance of materialized views: Problem, techniques, and application, in A. Gupta and I. S. Mumick, Eds., *materialized view*s, 145-158, 1998.
4. http://www.oracle.com/
5. Mistry, H., Prasan Roy, S., Ramamritham, S., Materialized view selection and maintenance using multi-query optimization, *Proc. SIGMOD*, 307-318, 2001.
6. Zhengxin, C., Intelligent data warehousing: From data preparation to data mining, CRC PRESS, 2001.

Keywords: Materialized view selection, query rewriting.

Acknowledgements: This work was supported by grant from National Research Council of Thailand (NRCT). The authors are member of Data Engineering and Knowledge Discovery (DEKD) Research Unit, which is fully supported by Suranaree University of Technology.

ประวัติผู้เขียน

นายจักรพันธ์ มหาวันตัง เกิดเมื่อวันที่ 25 ตุลาคม พ.ศ. 2525 ที่อำเภอนาดูน จังหวัดมหาสารคาม เริ่มการศึกษาระดับประถมศึกษาปีที่ 1-6 ที่โรงเรียนบ้านเหล่าจั่น ระดับมัธยมศึกษาปีที่ 1-3 ที่โรงเรียนนาดูนประชาสรรพ์ ระดับมัธยมศึกษาปีที่ 4-6 ที่โรงเรียนสารคามพิทยาคม จังหวัดมหาสารคาม และเข้าศึกษาในระดับปริญญาตรี ในปีการศึกษา 2545 สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาเมื่อปี พ.ศ. 2548ภายหลังสำเร็จการศึกษาได้เข้าศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีการศึกษา 2548

ในระหว่างการศึกษาได้รับความอนุเคราะห์จากคณาจารย์ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ โดยได้รับความไว้วางใจให้เป็นผู้ช่วยวิจัยและผู้สอนปฏิบัติการรายวิชา Database System