บทที่4

การทดลองและผลการทดลอง

จากบทที่3 ที่ได้กล่าวถึงหน้าที่ของออพติไมเซอร์ (Optimizer) ที่จะทำการวิเคราะห์หาแผนประมวลผล (Execution Plan) ที่ดีที่สุด โดยได้นำแนวทางการคำนวณจำนวนแถวแบบเอ็กแซกท์ (Exact Cardinality Computing) [1] ซึ่งเป็นวิธีที่แตกต่างจากออพติไมเซอร์ทั่วไปที่เลือกใช้การคำนวณจำนวนแถวโดยการประมาณ (Estimation Cardinality) ที่ช่วยให้การทำงานของออพติไมเซอร์เร็วขึ้น แต่ผลลัพธ์ที่ได้กลับไม่มีประสิทธิภาพ การคำนวณจำนวนแถวแบบเอ็กแซกท์เป็นวิธีที่ช่วยให้ออพติไมเซอร์วิเคราะห์แผนประมวลผล(Execution plan) ที่ดีที่สุดออกมาอย่างมีประสิทธิภาพ

โดยในงานวิจัย [1] ได้ทำการเพิ่มประสิทธิภาพการทำงานของการคำนวณจำนวนแถวแบบเอ็กแซกท์เพิ่มลงไปให้แก่ออพติไมเซอร์ ก็คือ การหาความสัมพันธ์ของการสอบถาม (Query) ที่มีอยู่ในอินพุตที่เรียกว่า เวิร์คโหลด (Workload) แล้วนำการสอบถามที่มีความสัมพันธ์กันเหล่านั้นมาจัดกลุ่มแล้วทำการประมวลผล (Execute) หาจำนวนแถว โดยทำการประมวลผลแค่ครั้งเดียว จึงทำให้เวลาของการคำนวณจำนวนแถวแบบเอ็กแซกท์แบบเดิมมีความเร็วยิ่งขึ้น นอกจากนี้ยังมีการเพิ่มขั้นตอนวิธี (Algorithm) ลงไปในออพติไมเซอร์ ที่จะทำหน้าที่จัดลำดับกลุ่มการสอบถามในเวิร์คโหลด ให้กลุ่มการสอบถามที่มีประสิทธิภาพดีที่สุดทำการประมวลผลก่อน

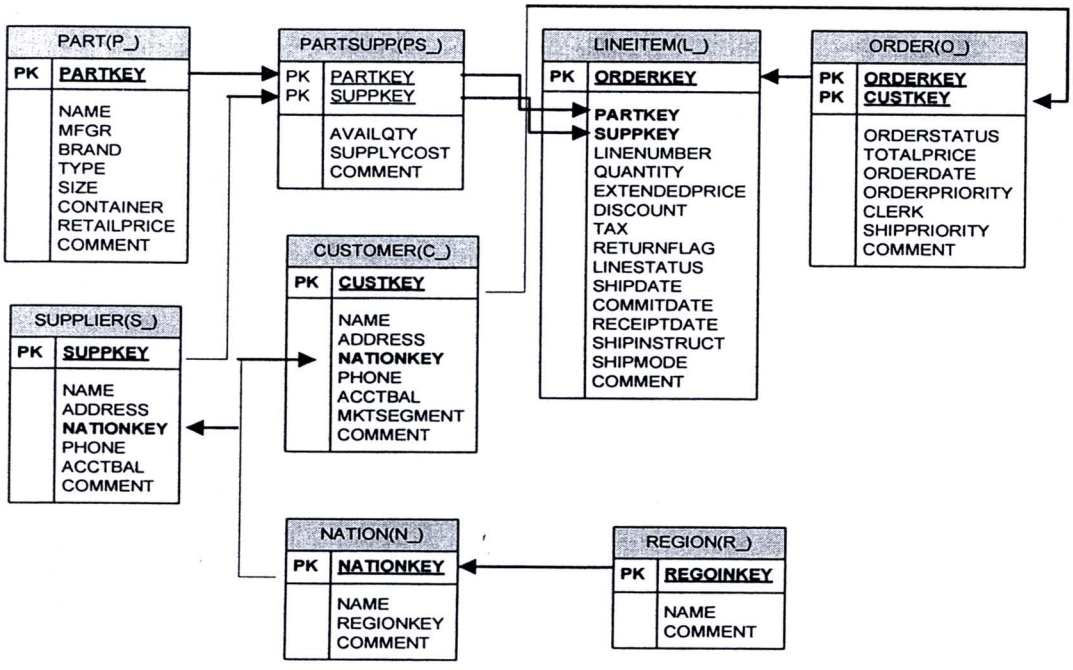
แนวทางในการวิจัยของวิทยานิพนธ์ฉบับนี้ ต้องการที่จะศึกษาการสอบถามที่มีอยู่ในเวิร์คโหลด โดยทำการแปลงการสอบถาม (Query Transformation) และทำการศึกษาผลกระทบที่เกิดขึ้น โดยทำการเปรียบเทียบกันระหว่างเวิร์คโหลดที่ไม่ได้ทำการแปลงการสอบถามกับเวิร์คโหลดที่ทำการแปลงการสอบถาม โดยทำการเปรียบเทียบผลการทดลองในรูปแบบของเวลา

ซึ่งจะนำรูปแบบเทคนิคการแปลงการสอบถามซึ่งมีด้วยกัน 4 รูปแบบมาทำการศึกษาซึ่งได้แก่

1. Subquery Unnestting
2. Group-By Distinct View Merging
3. Join Predicate Pushdown
4. Join Factorization

4.1 ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลองการแปลงการสอบถามสำหรับการคำนวณจำนวนแถวแบบเอ็กแซกท์ เป็นฐานข้อมูล (Database) ที่มีชื่อว่า TPCH โดยฐานข้อมูล TPCH เป็นฐานข้อมูลที่เกี่ยวกับการขายสินค้า โดยในฐานข้อมูลจะประกอบไปด้วยข้อมูลลูกค้า, ข้อมูลของผู้ผลิตสินค้า, ข้อมูลการสั่งซื้อสินค้าของลูกค้า เป็นต้น ซึ่งในรูปที่ 4.1 แสดงสกีมา (Schema) ของฐานข้อมูลดังกล่าว



รูปที่ 4.1 แสดงสกีมาของฐานข้อมูล

โดยรายละเอียดแต่ละตารางเป็นดังนี้

ตาราง REGION เป็นตารางที่จะเก็บทวีปต่างๆ ในโลก ซึ่งประกอบไปด้วยแอตทริบิวต์

REGIONKEY ทำหน้าที่เก็บรหัสทวีป

NAME ทำหน้าที่เก็บชื่อทวีป

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

ตาราง NATION เป็นตารางที่จะเก็บรายละเอียดประเทศที่ทำการค้า ซึ่งประกอบไปด้วยแอตทริบิวต์

NATIONKEY ทำหน้าที่เก็บรหัสของประเทศ

NAME ทำหน้าที่เก็บชื่อประเทศ

REGIONKEY ทำหน้าที่เก็บรหัสทวีป

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

ตาราง SUPPLIER เป็นตารางที่จะเก็บรายละเอียดของผู้ผลิตสินค้า ซึ่งประกอบไปด้วยแอตทริบิวต์

SUPPKEY ทำหน้าที่เก็บรหัสของผู้ผลิตสินค้า

NAME ทำหน้าที่เก็บชื่อผู้ผลิตสินค้า

ADDRESS ทำหน้าที่เก็บที่อยู่ของผู้ผลิตสินค้า

NATIONKEY ทำหน้าที่เก็บรหัสประเทศของผู้ผลิตสินค้า

PHONE ทำหน้าที่เก็บเบอร์โทรศัพท์ของผู้ผลิตสินค้า

ACCTBAL ทำหน้าที่เก็บ ยอดเงินคงเหลือในบัญชีของผู้ผลิต

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

ตาราง CUSTOMER เป็นตารางที่จะเก็บรายละเอียดของลูกค้า ซึ่งประกอบไปด้วยแอตทริบิวต์

CUSTKEY ทำหน้าที่เก็บรหัสของลูกค้า

NAME ทำหน้าที่เก็บชื่อของลูกค้า

ADDRESS ทำหน้าที่เก็บที่อยู่ของลูกค้า

NATIONKEY ทำหน้าที่เก็บรหัสประเทศของลูกค้า

PHONE ทำหน้าที่เก็บเบอร์โทรศัพท์ของลูกค้า

ACCTBAL ทำหน้าที่เก็บยอดเงินคงเหลือในบัญชีของลูกค้า

MKTSEGMENT ทำหน้าที่เก็บประเภทสินค้าของลูกค้า

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

ตาราง PART เป็นตารางที่จะเก็บส่วนประกอบของสินค้า ซึ่งประกอบไปด้วยแอตทริบิวต์

PARTKEY ทำหน้าที่เก็บรหัสของส่วนประกอบของสินค้า

NAME ทำหน้าที่เก็บชื่อของส่วนประกอบของสินค้า

MFGR ทำหน้าที่เก็บชื่อโรงงานที่ผลิตสินค้า

BRAND ทำหน้าที่เก็บยี่ห้อของสินค้า

TYPE ทำหน้าที่เก็บชนิดของสินค้า

SIZE ทำหน้าที่เก็บขนาดของสินค้า

CONTIANER ทำหน้าที่เก็บประเภทของการบรรจุสินค้า

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

RETAILPRICE ทำหน้าที่เก็บราคาขายปลีกของส่วนประกอบ

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

ตาราง PARTSUPP เป็นตารางที่จะเก็บส่วนประกอบของสินค้าของผู้ผลิต ซึ่งประกอบไปด้วยแอตทริบิวต์

PARTKEY ทำหน้าที่เก็บรหัสของส่วนประกอบของสินค้า

'SUPPKEY ทำหน้าที่เก็บรหัสของผู้ผลิต

AVAILQTY ทำหน้าที่เก็บวันที่จะได้รับของของลูกค้า

SUPPLYCOST ทำหน้าที่เก็บจำนวนเงินของผู้ผลิต

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

ตาราง ORDER เป็นตารางที่จะเก็บรายละเอียดในการสั่งสินค้า ซึ่งประกอบไปด้วยแอตทริบิวต์

ORDERKEY ทำหน้าที่เก็บรหัสของใบสั่งสินค้า

CUSTKEY ทำหน้าที่เก็บรหัสของลูกค้า

ORDERSTATUS ทำหน้าที่เก็บสถานะของลูกค้า

TOTALPRICE ทำหน้าที่เก็บราคาสินค้าทั้งหมดของลูกค้า

ORDERDATE ทำหน้าที่เก็บวันที่ทำการสั่งสินค้า

ORDERPRIORITY ทำหน้าที่เก็บความสำคัญของการสั่งสินค้า

CLERK ทำหน้าที่เก็บชื่อของเสมียนที่ทำการรับการสั่งสินค้า

SHIPPRIORITY ทำหน้าที่เก็บความสำคัญของการส่งสินค้า

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

ตาราง LINEITEM เป็นตารางที่จะเก็บรายละเอียดของการส่งสินค้า ซึ่งประกอบไปด้วยแอตทริบิวต์

ORDERKEY ทำหน้าที่เก็บรหัสของใบสั่งสินค้า

PARTKEY ทำหน้าที่เก็บรหัสของส่วนประกอบของสินค้า

SUPPKEY ทำหน้าที่เก็บรหัสของผู้ผลิต

LINENUMBER ทำหน้าที่เก็บหมายเลขของการส่งสินค้า

QUANTITY ทำหน้าที่เก็บจำนวนของสินค้า

EXTENDEDPRICE ทำหน้าที่เก็บจำนวนเงินที่เพิ่มในสินค้า

DISCOUNT ทำหน้าที่เก็บส่วนลดของสินค้า

TAX ทำหน้าที่เก็บภาษีของสินค้า

RETURNFLAG ทำหน้าที่เก็บสัญลักษณ์การยืนยันการส่งสินค้า

LINESTATUS ทำหน้าที่เก็บสถานะของการส่ง

SHIPDATE ทำหน้าที่เก็บวันที่ทำการส่งสินค้า

COMMITDATE ทำหน้าที่เก็บวันที่ทำสัญญาในการสั่งสินค้า

RECEIPTDATE ทำหน้าที่เก็บวันที่ทำการรับสินค้า

SHIPINSTRUCT ทำหน้าที่เก็บลักษณะการส่งค้า

SHIPMODE ทำหน้าที่เก็บประเภทของการส่งสินค้า

COMMENT ทำหน้าที่เก็บข้อเสนอแนะ

จากฐานข้อมูลดังกล่าวจะทำการทดลองโดยมีการสอบถามด้วยกัน 4 รูปแบบ ดังที่ได้กล่าวมาแล้วข้างต้น ซึ่งแต่ละการสอบถามจะประกอบไปด้วยข้อมูล 5 ชุด โดยจะมีข้อมูลดั้งเดิม 1 ชุด ข้อมูลที่จะทำการลดเซ็ตข้อมูล (Dataset) ที่จะทำให้จำนวนผลลัพธ์ของข้อมูลลดลงจำนวน 2 ชุด และข้อมูลที่ทำการเพิ่มเซ็ตข้อมูลที่จะทำให้ผลลัพธ์ของข้อมูลเพิ่มขึ้นจำนวน 2 ชุด โดยข้อมูลดั้งเดิมที่ไม่ได้ทำการเพิ่มหรือลดเซ็ตข้อมูล จะทำการบอกเป็นเปอร์เซ็นต์ ซึ่งข้อมูลดั้งเดิมจะเป็น 100% ส่วนเซ็ตข้อมูลที่เพิ่มหรือลดก็จะเป็นเปอร์เซ็นต์ที่เพิ่มหรือลดตามลำดับ

4.2 ผลการทดลอง

จากรูปแบบการแปลงการสอบถามที่มี 4 รูปแบบด้วยกัน ได้ผลการทดลองดังนี้

4.2.1 รูปแบบ Subquery Unnesting

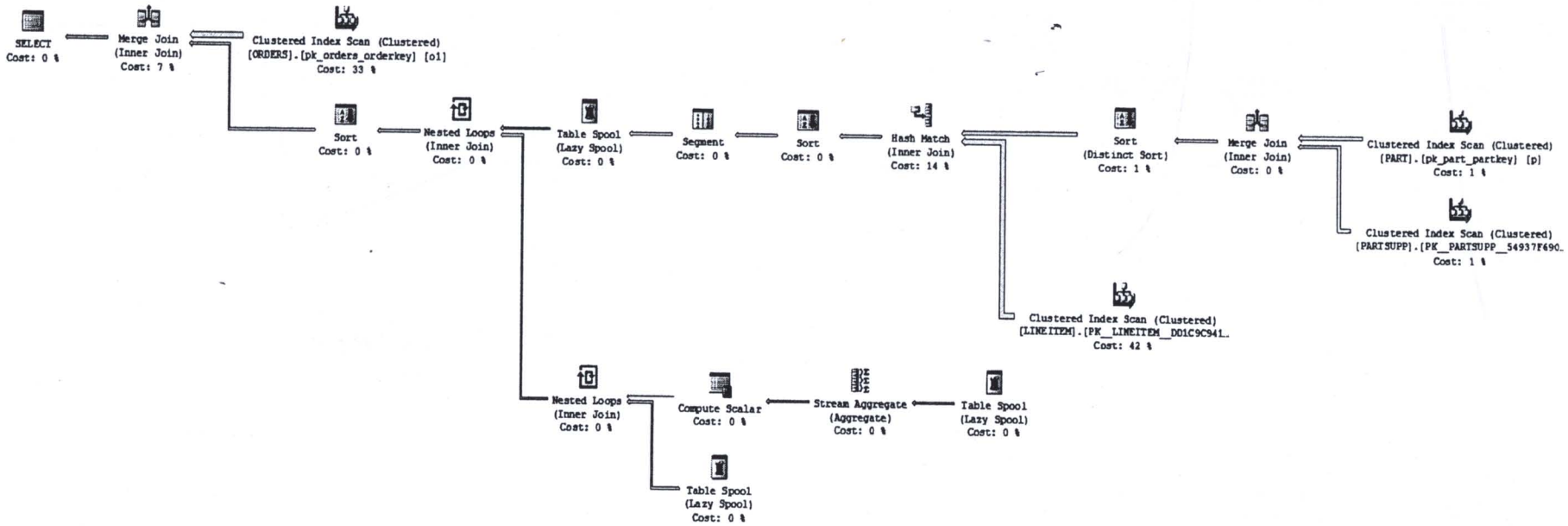
รูปแบบ Subquery Unnesting เป็นรูปแบบที่มีการสอบถามย่อยหลายๆ การสอบถามซ้อนอยู่ในการสอบถามหลักและมีความสัมพันธ์กันระหว่างการสอบถามย่อยเหล่านั้น ซึ่งเป็นเรื่องที่ยุ่งยากสำหรับออพติไมเซอร์ที่จะทำการเมิร์จ (Merge) กันระหว่างตารางที่มีความสัมพันธ์ของการสอบถามย่อยเหล่านั้นการแปลงการสอบถามรูปแบบนี้จะทำการแปลงโดยจะนำการสอบถามย่อยไปไว้ในวิว (View) เพื่อที่จะง่ายต่อการประมวลผล จากรูปที่ 4.2 แสดงการสอบถามรูปแบบดังกล่าวที่จะใช้นำมาทดลอง การสอบถามดังกล่าวต้องการที่จะทราบถึงส่วนลดของลูกค้าและรหัสของลูกค้าโดยที่มีการสั่งสินค้าตั้งแต่วันที่ 1 มกราคม ค.ศ. 1990 เป็นต้นไปและมีส่วนเพิ่มของราคาสินค้าต้องมากกว่าค่าเฉลี่ยของส่วนเพิ่มของราคาสินค้าโดยที่มีการค่าใช้จ่ายแก่ผู้ผลิตมากกว่า 300 ขึ้นไป โดยจากรูปที่ 4.3 แสดงการแปลงการสอบถามจากรูปที่ 4.2 ซึ่งนำมาทดลองเช่นเดียวกัน ซึ่งจากรูปที่ 4.3 ได้ทำการนำการสอบถามย่อยที่มีการคิดค่าเฉลี่ยของส่วนเพิ่มของราคาสินค้าได้นำไปไว้ในวิวเพื่อที่จะทำการคำนวณค่าเฉลี่ยทำเพียงครั้งเดียวแล้วค่อยนำไปเปรียบเทียบค่าในตาราง lineitem โดยในรูปที่ 4.4 และรูปที่ 4.5 แสดงแผนประมวลผลของรูปแบบ Subquery Unnesting แบบไม่แปลงการสอบถามและแบบแปลงการสอบถามตามลำดับ ซึ่งจากรูปทั้งสองจะเห็นได้ว่าขั้นตอนในการเมิร์จนั้น มีลำดับการเมิร์จที่เปลี่ยนไปในการคำนวณแบบจัดกลุ่ม

```
SELECT l1.l_discount, o1.o_custkey
       FROM lineitem l1, orders o1
       WHERE l1.l_orderkey=o1.o_orderkey
             AND o_orderdate > '1990-01-01'
             AND l1.l_extendedprice >
                        (SELECT
                   AVG(l_extendedprice)
                          FROM lineitem l2
                          WHERE
                    l2.l_suppkey=l1.l_suppkey)
             AND l1.l_suppkey IN (SELECT ps_suppkey
                  FROM partsupp ps, part p
                 WHERE ps.ps_partkey=p.p_partkey
                     AND ps.ps_supplycost > 300)
```

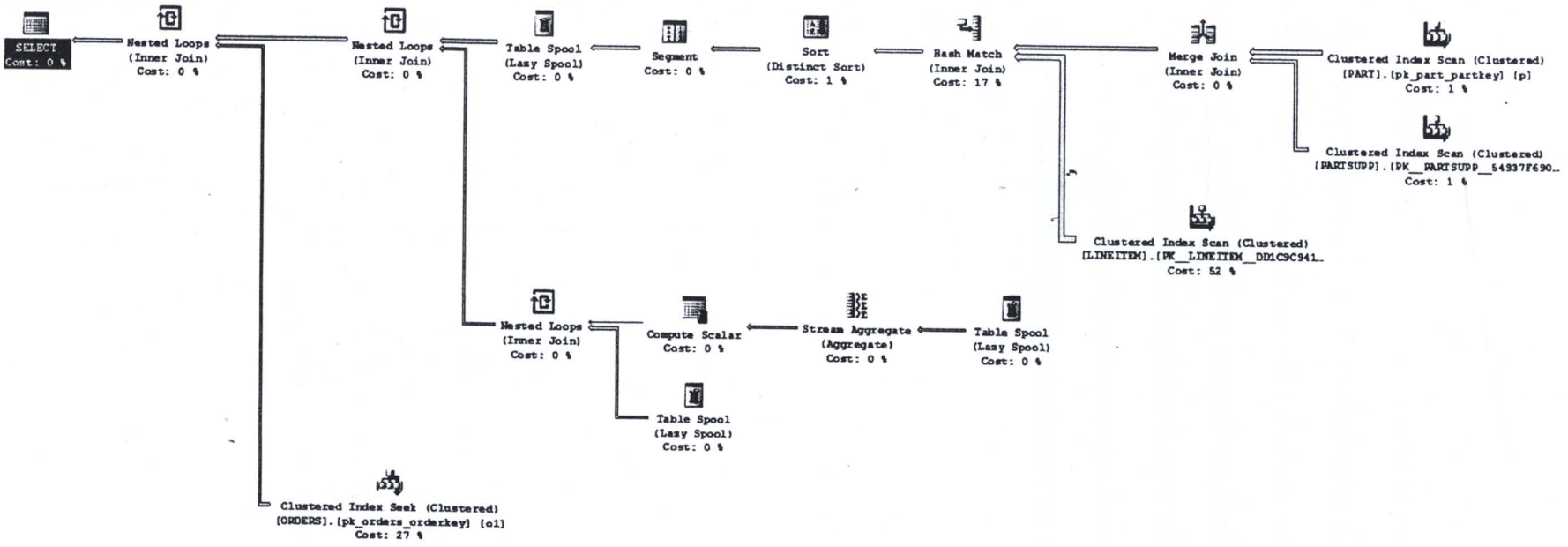
รูปที่ 4.2 แสดงการสอบถามแบบ Subquery Unnesting

```
SELECT l1.l_discount, o1.o_custkey
        FROM lineitem l1, orders o1, (SELECT
               AVG(l_extendedprice)
                                    Avg_exprice,
                                     l_suppkey
                          FROM lineitem l2
                          GROUP BY l_suppkey)
       WHERE l1.l_orderkey=o1.o_orderkey
       AND o_orderdate > '1990-01-01'
       AND l1.l_suppkey=v.l_suppkey
       AND l1.l_extendedprice > v.avg_exprice
       AND l1.l_suppkey IN (   SELECT ps_suppkey
                      FROM partsupp ps, part p
                                  WHERE
                        ps.ps_partkey=p.p_partkey
                       AND ps.ps_supplycost > 300)
```

รูปที่ 4.3 แสดงการแปลงการสอบถามแบบ Subquery Unnesting



รูปที่ 4.4 แสดงแผนประมวลผลแบบไม่แปลงการสอบถามของ Subquery Unnesting



รูปที่ 4.5 แผนประมวลผลแบบแปลงการสอบถามของ Subquery Unnesting

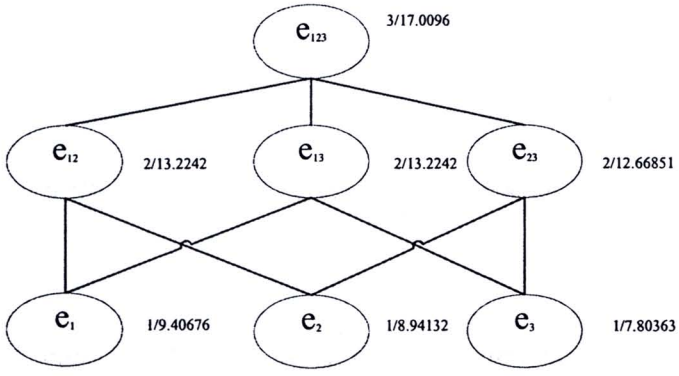
โดยการทดลองจะทำการนำการสอบถามที่มีรูปแบบคล้ายกันอีก 2 การสอบถามนำมาทดลอง เมื่อการสอบถามทั้ง 3 อยู่ในเวิร์คโหลดจะถูกจัดกลุ่มให้อยู่ในซิกเนเจอร์ (Signature) เดียวกันเพราะ มีรูปแบบการสอบถามที่คล้ายกันและจะนำไปทำขั้นตอนการหาจำนวนแถวแบบเอ็กแซกท์ดังที่เคยกล่าวมาแล้วในบทที่ 3 เมื่อมีการสอบถาม 3 การสอบถามขั้นตอนการหาแคนดิเดตเจเนเรชันจะเป็นไปดังรูปที่ 4.7 และรูปที่ 4.6 โดยเป็นของการสอบถามที่ยังไม่ได้ทำการแปลงการสอบถามกับรูปแบบการสอบถามที่ทำการแปลงการสอบถามตามลำดับ และตารางที่ 4.1 และ ตารางที่ 4.2 แสดงค่าประสิทธิภาพของการสอบถามโดยการประมาณค่าของออพติไมเซอร์ โดยเป็นการสอบถามที่ไม่ได้แปลงการสอบถามกับการสอบถามที่ทำการแปลงการสอบถามตามลำดับอีกเหมือนกัน

ตารางที่ 4.1 แสดงค่าประสิทธิภาพของอันเนสติงแบบไม่แปลงการสอบถาม

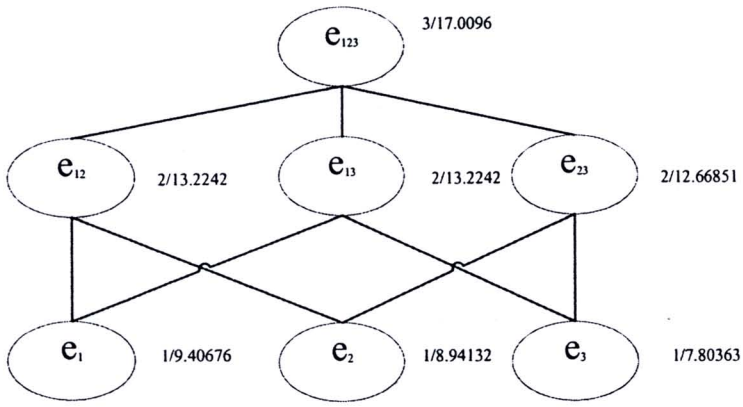
การสอบถาม	ค่าประสิทธิภาพ
e_1	9.40676
e_2	8.94132
e_3	7.80363
e_{12}	13.2242
e_{13}	13.2242
e_{23}	12.66851
e_{123}	17.0096

ตารางที่ 4.2 แสดงค่าประสิทธิภาพของอันเนสติงแปลงการสอบถาม

การสอบถาม	ค่าประสิทธิภาพ
e_1	4.97721
e_2	4.65191
e_3	4.08262
e_{12}	8.62400
e_{13}	8.62400
e_{23}	8.26044
e_{123}	9.12218

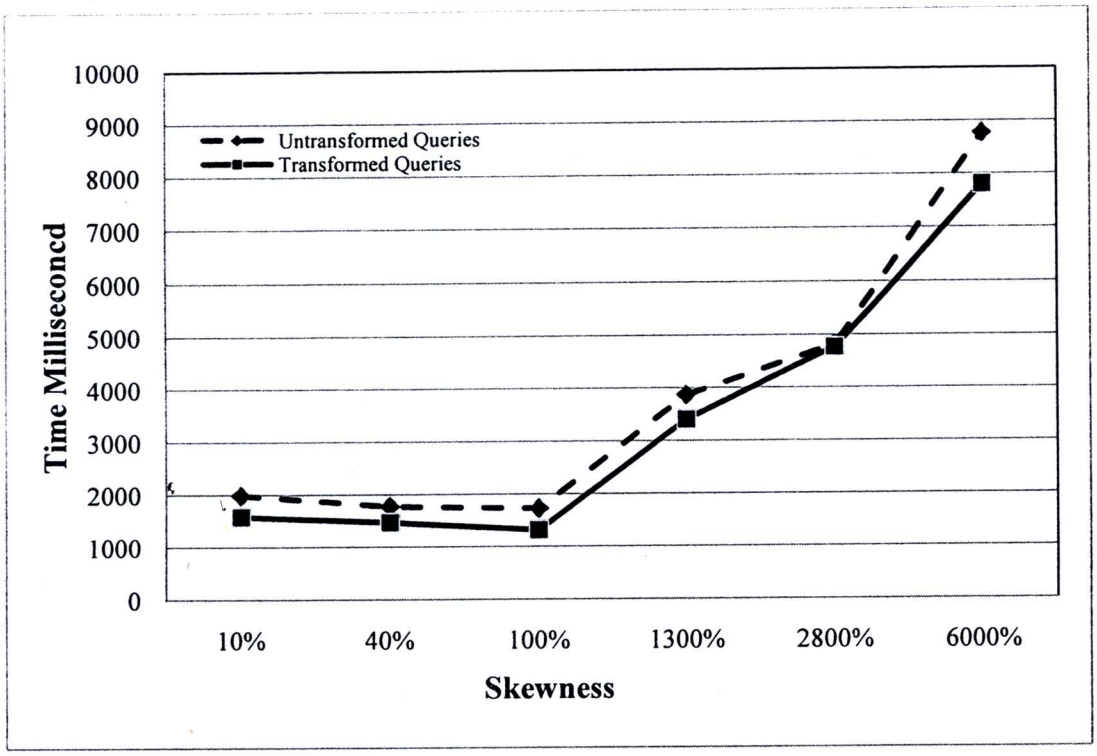


รูปที่ 4.6 แสดงขั้นตอนแคนดิเดตเจเนเรชัน Subquery Unnesting



รูปที่ 4.7 แสดงขั้นตอนแคนดิเดตเจเนเรชัน Subquery Unnesting

เมื่อถึงขั้นตอนแคนดิเดตเจเนเรชันดังรูปที่ 4.6 โดยในรูปจะเป็นการสอบถามแบบที่ยังไม่ได้ทำการแปลงการสอบถาม ขั้นตอนหลักๆ จะทำการเลือกรูปแบบว่าจะนำรูปแบบใดไปทำเคสคิวรี่ เมื่อได้ค่าประสิทธิภาพและนำไปผ่านขั้นตอนการคัดเลือกแคนดิเดตเจเนเรชันดังที่ได้เคยกล่าวมาแล้วในบทที่ 3 สำหรับรูปแบบอันเนสติงที่ยังไม่ได้ทำการแปลงการสอบถามรูปแบบที่จะนำไปทำเคสคิวรี่คือ รูปแบบ e_{123}และเช่นกันในรูปที่ 4.7 ก็ทำขั้นตอนที่เหมือนกันแต่เป็นอันเนสติงแบบแปลงการสอบถามโดยรูปแบบการเลือกแคนดิเดตเจเนเรชันจะเป็นรูปแบบ e_{123} เช่นกัน จะเห็นได้ว่าจากตารางที่ 4.2 เปรียบเทียบกับตารางที่ 4.1 ค่าประสิทธิภาพของการแปลงการสอบถามมีค่าที่ดีกว่าการสอบถามที่ไม่แปลงการสอบถามทุกรูปแบบ การแปลงการสอบถามแบบอันเนสติงช่วยเพิ่มประสิทธิภาพให้แก่ออพติไมเซอร์ในการประมวลผล ต่อจากนั้นจะนำคิวรี่ทั้ง 3 มาทำรูปแบบเคสคิวรี่ที่มีการสอบถามเดียวเพื่อที่จะให้ออพติไมเซอร์คำนวณจำนวนแถวซึ่ง โดยผลการทดลองเป็นกราฟได้แสดงในรูปที่ 4.8 โดยมีแกน y เป็นเวลาวัดเป็นหน่วยมิลลิวินาที และแกน x เป็นเซ็ตข้อมูลของข้อมูล 5 ชุด โดยกราฟแสดงการเปรียบเทียบเวลาระหว่างการสอบถามที่มีการแปลงการสอบถามกับการสอบถามที่ไม่ได้ทำการแปลงการสอบถาม



รูปที่ 4.8 แสดงกราฟผลการทดลอง Subquery Unnesting

การแปลงการสอบถามในรูปแบบอันเนสติง เมื่อพิจารณาจากจากคิวรี่แบบไม่แปลงการสอบถามนั้นจะทำงานได้ดีเมื่อข้อมูลจากตาราง lineitem มีจำนวนข้อมูลที่มีค่า l_extendedprice มากกว่าค่าเฉลี่ยที่ทำการคำนวณมาแล้วและถ้าคอลัมน์ที่มีความสัมพันธ์กับการสอบถามย่อยอื่นมีการสร้างดัชนี (Index) ในที่นี้คือ l2.l_suppkey การทำงานของการสอบถามแบบไม่แปลงการสอบถามจะดีกว่าแบบการสอบถามแบบแปลงการสอบถาม ส่วนในกรณีของการแปลงการสอบถาม จากคิวรี่จะสังเกตได้ว่าได้ทำการนำส่วนที่ทำการคำนวณค่าเฉลี่ย ไปไว้ในวิวเพื่อที่จะทำการจัดกลุ่ม(Group-by) และคำนวณเพียงแค่ครั้งเดียว

พิจารณาจากกราฟผลการทดลองช่วงแรกข้อมูลอยู่ที่ 10% จนถึงที่ 1300% นั้นแบบแปลงการสอบถามทำงานได้เร็วกว่าแบบไม่แปลงการสอบถาม แต่สังเกตได้ว่าที่ 2800% เวลาของการแปลงแบบสอบถามและแบบไม่แปลงการสอบถามใกล้เคียงกัน ที่เป็นเช่นนั้นเพราะว่าข้อมูลในเซ็ตข้อมูลที่ 2800% มีจำนวนข้อมูลในคอลัมน์ ของ l_extendedprice มีข้อมูลที่มากกว่าค่าเฉลี่ยใกล้เคียงกับข้อมูลที่น้อยกว่าค่าเฉลี่ย อย่างไรก็ตามในเซ็ตข้อมูลที่ 6000% แบบแปลงการสอบถามมีเวลาที่ดีกว่าแบบไม่แปลงการสอบถามแสดงให้เห็นว่าจำนวนของข้อมูลที่มากกว่าค่าเฉลี่ยกลับมามากกว่า

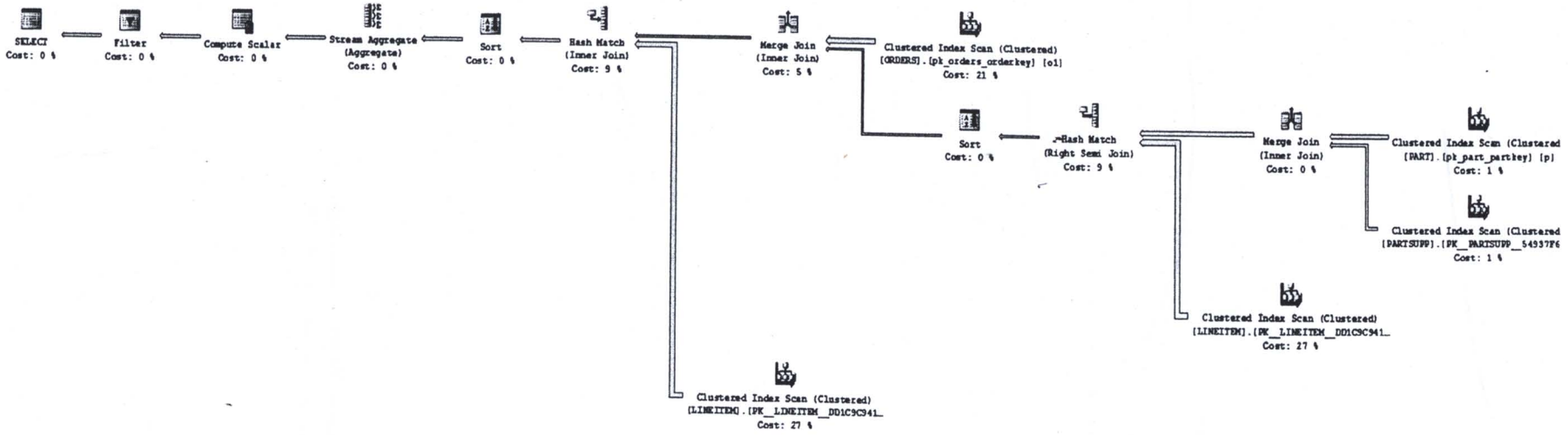
อีกครั้ง จากกราฟทำให้ทราบว่าการแปลงการสอบถามในรูปแบบอันเนสติงนั้นต้องพิจารณาข้อมูลในดาต้าเบสด้วยว่า มีความสัมพันธ์กับรูปแบบใดดังที่กล่าวข้างต้น แล้วจึงตัดสินใจว่าจะทำการแปลงการสอบถามหรือไม่แปลงการสอบถาม

4.2.2 รูปแบบ Group-by and Distinct View Merging

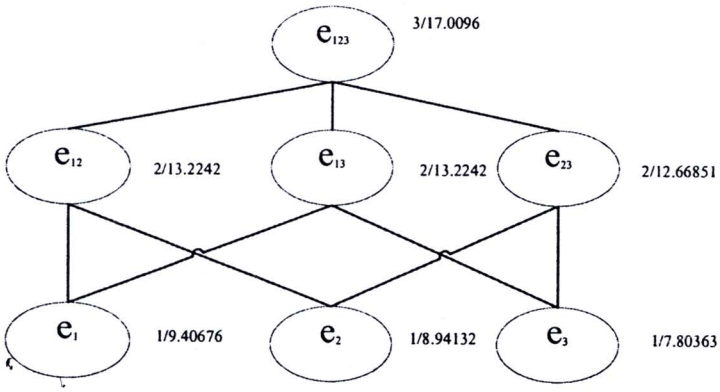
การทดลองในรูปแบบ Group-by and Distinct View Merging นี้จะจะใช้เซ็ตข้อมูลชุดเดียวกับอันเนสติ้ง เนื่องจากรูปแบบดังกล่าวเป็นรูปแบบที่จะทำการแปลงการสอบถามสำหรับการสอบถามที่มี Group by หรือ Distinct อยู่ในวิว ซึ่งการสอบถามที่จะนำมาทดลองคือการสอบถามรูปที่ 4.3 ซึ่งตรงกับรูปแบบของ Group-by and Distinct View Merging โดยในรูปที่ 4.9 แสดงการแปลงการสอบถามที่จะนำมาทำการทดลอง ซึ่งรูปที่ 4.11 แสดงการเลือกแคนดิเดตเจเนเรชันของ Group-by and Distinct View Merging แบบที่เป็นการแปลงการสอบถามและในรูปที่ 4.10 แสดงแผนประมวลของรูปแบบ Group-by and Distinct View Merging ที่ทำการแปลงการสอบถามมาจากแผนประมวลผลในรูปที่ 4.5 ซึ่งจะเห็นได้ว่าขั้นตอนของการคำนวณแบบจัดกลุ่มจะนำมาพิจารณาเกือบจะเป็นขั้นตอนท้ายสุด ซึ่งจากแผนประมวลผลจะทำให้การทำงานของออพติไมเซอร์เปลี่ยนแปลงไปจากเดิมและประสิทธิภาพก็เปลี่ยนแปลงไปจากเดิมอีกด้วย

```
SELECT l1.l_discount, o1.o_custkey
FROM lineitem l1, orders o1, lineitem l2
WHERE l1.l_orderkey=o1.o_orderkey
    AND o_orderdate > '1990-01-01'
    AND l2.l_suppkey=l1.l_suppkey
 AND l1.l_suppkey IN (SELECT ps_suppkey
                      FROM partsupp ps, part p
                     WHERE ps.ps_partkey=p.p_partkey
                           AND ps.ps_supplycost > 300)
      GROUP BY l2.l_suppkey, l1.l_suppkey,
      o1.o_custkey,              l1.l_discount,
      l1.l_extendedprice
HAVING l1.l_extendedprice > AVG(l2.l_extendedprice)
```

รูปที่ 4.9 แสดงการแปลงการสอบถามรูปแบบ Group-by and Distinct View Merging

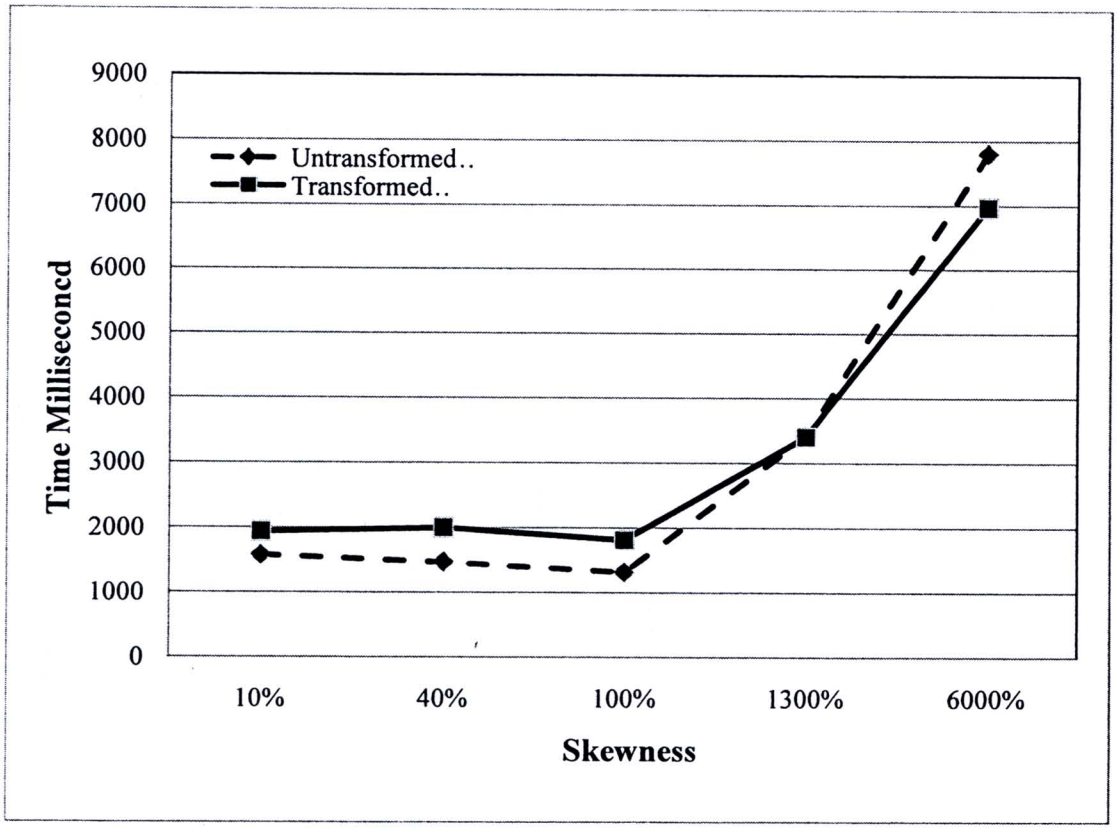


รูปที่ 4.10 แสดงแผนประมวลแบบแปลงการสอบถามของ Group-by and Distinct View Merging



รูปที่ 4.11 แสดงขั้นตอนแคนดิเดตเจเนเรชันของ Group-by and Distinct View Merging

ในขั้นตอนการเลือกแคนดิเดตเจเรชันของรูปแบบ Group-by and Distinct View Merging ได้เป็นรูปแบบ e_{123} ซึ่งรูปที่ 4.12 แสดงกราฟผลการทดลอง



รูปที่ 4.12 แสดงกราฟผลการทดลอง Group-by and Distinct View Merging

เทคนิคการแปลงการสอบถามแบบนี้เป็นรูปแบบที่มีการจัดกลุ่ม(Group-by) หรือไม่เอาข้อมูลที่ซ้ำ (Distinct) อยู่ในวิว โดยจะนำการสอบถามที่มีการจัดกลุ่มที่อยู่ในวิวนำออกไปไว้นอกวิว เพื่อที่จะทำการเมิร์จ (merge) กันในการสอบถามที่อยู่นอกวิวได้ การแปลงการสอบถามในรูปแบบนี้อาจจะดีหรือแย่กว่าแบบไม่แปลงการสอบถามก็ได้ ซึ่งในส่วนของการสอบถามที่ไม่มีการแปลงการสอบถามนั้นมีการ aggregation เพียงแค่ตาราง lineitem ตารางเดียวแต่ในส่วนของการแปลงการสอบถามต้องหาความสัมพันธ์เพื่อสร้างตารางใหม่ของตาราง lineitem ทั้ง 2 ตาราง ดังรูปที่ 4.9

เมื่อพิจารณาจากกราฟจะเห็นได้ว่าช่วงแรกแบบแปลงการสอบถามทำงานได้ช้ากว่าแบบไม่แปลงการสอบถาม นั่นเป็นเพราะผลลัพธ์ของข้อมูลที่จะนำมาจอย (Join) กันนั้นยังมีไม่สูงมากนัก แต่เมื่อข้อมูลเพิ่มมากขึ้นที่ 6000% แบบแปลงการสอบถามทำงานได้เร็วกว่าแบบไม่แปลงการสอบถาม จึงสามารถสรุปได้ว่าการแปลงการสอบถามรูปแบบนี้เหมาะสำหรับชุดข้อมูลที่มีจำนวนแถวมากๆ ถึงจะทำงานได้ดีกว่าแบบไม่แปลงการสอบถาม

4.2.3 รูปแบบ Join Predicate Pushdown

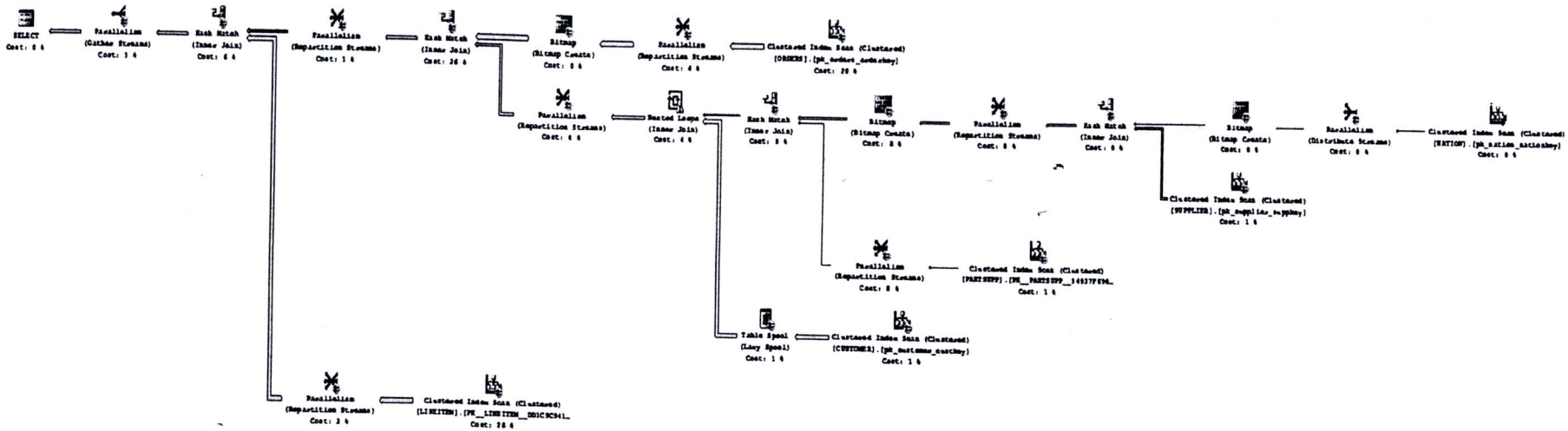
การแปลงการสอบถามในรูปแบบนี้ เป็นรูปแบบการสอบถามที่จะนำเงื่อนไขพรีดิเคตที่มีอยู่ในวิวแล้วมีความสัมพันธ์กับเงื่อนไขพรีดิเคตที่อยู่นอกวิว นำพรีดิเคตที่อยู่นอกวิวนั้นนำมาไว้ในวิวเพื่อที่จะทำให้การประมวลซับซ้อนน้อยลง ในส่วนการทดลอง รูปที่ 4.13 และ รูปที่ 4.14 แสดงการสอบถามที่นำมาทดลองของการสอบถามแบบไม่แปลงการสอบถามและแบบแปลงการสอบถามตามลำดับ โดยการสอบถามดังกล่าวต้องการที่ทราบถึง ชื่อของลูกค้า, วันที่ลูกค้าทำการสั่งสินค้าและวันที่จะทำการส่งสินค้า โดยต้องเป็นสินค้าที่มีผู้ผลิตอยู่ในประเทศอังกฤษ และรูปที่ 4.17 และรูปที่ 4.18 แสดงแคนดิเดตเจเนเรชันของการสอบถามแบบไม่แปลงการสอบถามและแบบแปลงการสอบถามตามลำดับ ซึ่งในรูปที่ 4.15 และรูปที่ 4.16 แสดงแผนประมวลผลแบบไม่แปลงการสอบถามและแบบแปลงการสอบถามตามลำดับ

```
        SELECT c.c_name, o.o_orderdate,
l.l_shipdate
        FROM customer c, lineitem l, partsupp
        ps, orders o,
        (SELECT distinct s.s_suppkey,
          FROM nation n, supplier s
          WHERE n.n_nationkey=s.s_nationkey
       AND n.n_nationkey = 'united kingdom') v
        WHERE v.s_suppkey=ps_suppkey
              AND c.c_custkey=o.o_custkey
              AND l.l_orderkey=o.o_orderkey
              AND l.l_shipdate > '1995-01-01'
```

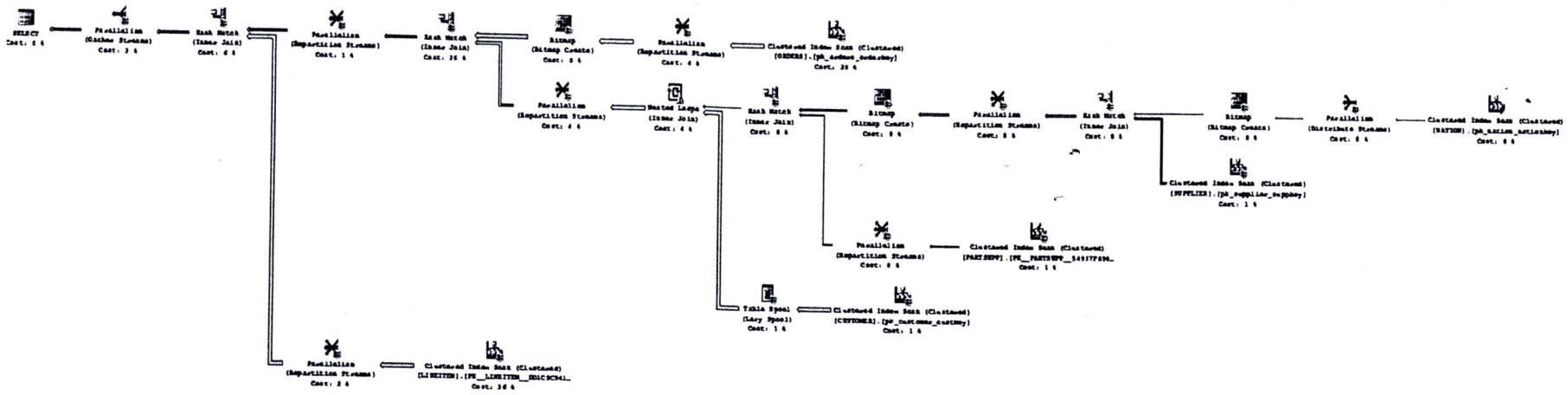
รูปที่ 4.13 แสดงการสอบถามแบบ Join Predicate Pushdown

```
SELECT c.c_name, o.o_orderdate, l.l_shipdate
       FROM customer c, lineitem l, partsupp ps,
       orders o, (SELECT distinct s.s_suppkey,
             FROM nation n, supplier s, partsupp ps
             WHERE n.n_nationkey=s.s_nationkey
                   AND ps.ps_suppkey=s.s_suppkey
                   AND n.n_nationkey = 'united
             kingdom')v
       WHERE c.c_custkey=o.o_custkey
             AND l.l_orderkey=o.o_orderkey
             AND l.l_shipdate > '1995-01-01'
```

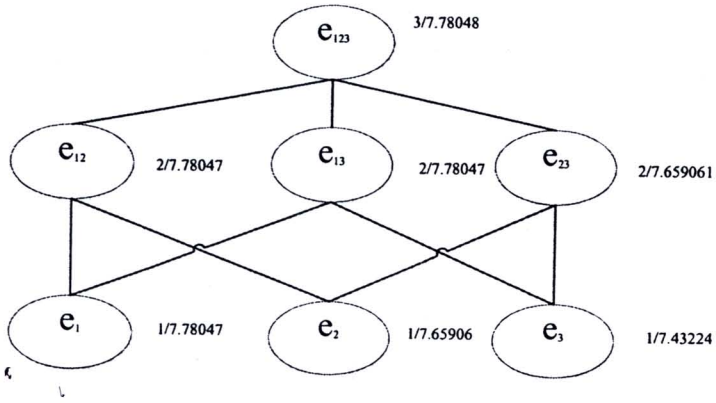
รูปที่ 4.14 แสดงการแปลงการสอบถามแบบ Join Predicate Pushdown



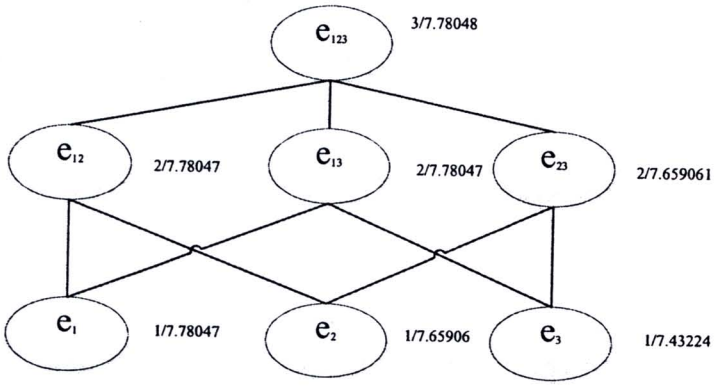
รูปที่ 4.15 แสดงแผนประมวลผลแบบไม่แปลงการสอบถามของ Join Predicate Pushdown



รูปที่ 4.16 แสดงแผนประมวลผลแบบแปลงการสอบถามของ Join Predicate Pushdow



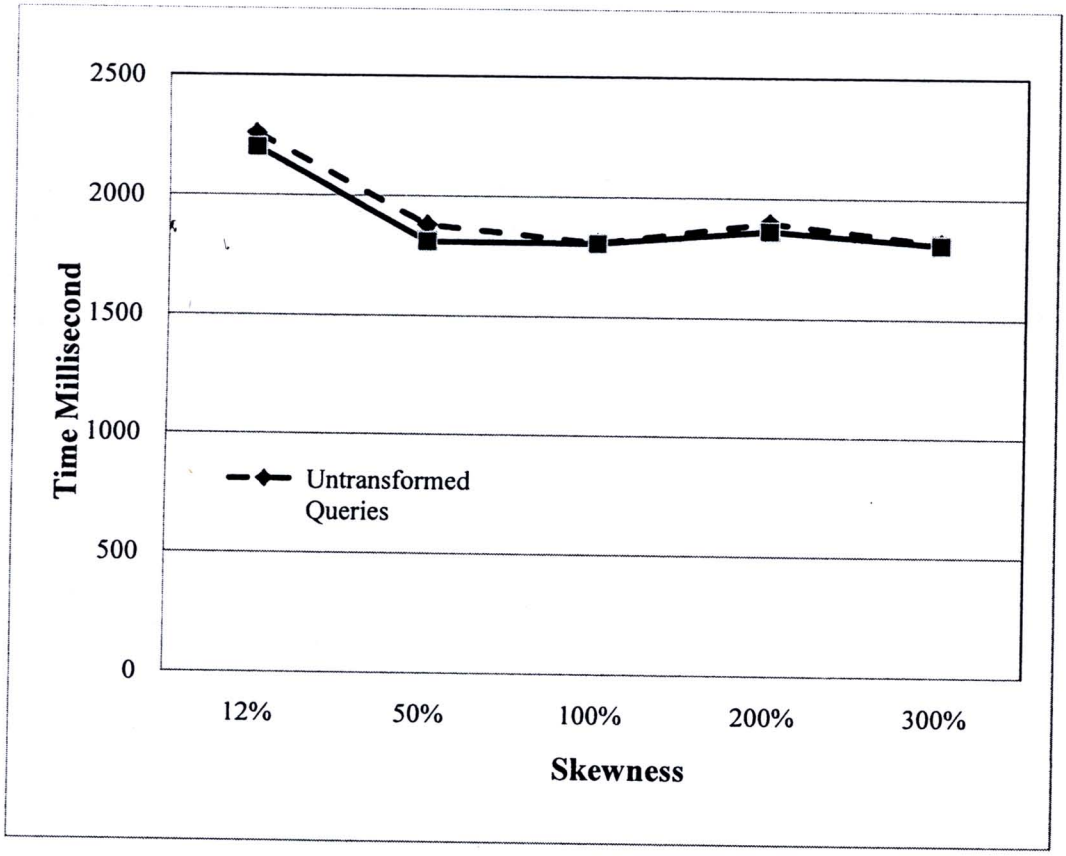
รูปที่ 4.17 แสดงแคนดิเดตเจเนเรชัน Join Predicate Pushdown



รูปที่ 4.18 แสดงแคนดิเดตเจเนเรชัน Join Predicate Pushdown

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

ซึ่งรูปแบบที่จะนำไปทำเคสดิวรี่ทั้งแบบไม่แปลงการสอบถามและแบบแปลงการสอบถามคือ e_{123} โดยกราฟผลการทดลองของการสอบถามรูปแบบนี้ได้แสดงในรูปที่ 4.19



รูปที่ 4.19 แสดงกราฟผลการทดลอง Join Predicate Pushdown

การแปลงการสอบถามรูปแบบ Join Predicate Pushdown การแปลงการสอบถามรูปแบบนี้จะดีกว่าแบบไม่แปลงการสอบถามเมื่อมีการทำดัชนี (Index) ให้แก่พรีดิเคตที่มีความสัมพันธ์กับพรีดิเคตในวิวในการสอบถามนี้คือ ps_suppkey และจำนวนผลลัพธ์มีจำนวนที่น้อยๆ ซึ่งจะต้องดูข้อมูลของฐานข้อมูลที่นำมาใช้ว่ามีลักษณะแบบใด

จากรูปที่ 4.19 ซึ่งแสดงกราฟผลการทดลอง สังเกตได้ว่า ในเซ็ตข้อมูลของข้อมูลที่ 12% และที่ 50% แบบแปลงการสอบถามทำงานได้ดีกว่าแบบไม่แปลงการสอบถาม ทั้งนี้

เพราะผลลัพธ์ของข้อมูลมีจำนวนไม่มาก แต่เมื่อเซ็ตข้อมูลของข้อมูลตั้งแต่ 100% ขึ้นไป แบบแปลงการสอบถามเริ่มทำงานได้ช้ากว่าแบบไม่แปลงการสอบถาม ทั้งนี้เพราะข้อมูลมากขึ้นนั่นเอง ดังนั้นรูปแบบการสอบถามแบบ Join Predicate Pushdown นี้สมควรใช้รูปแบบการแปลงการสอบถามสำหรับฐานข้อมูลที่มีขนาดไม่ใหญ่มากนัก

4.2.4 รูปแบบ Join Factorization

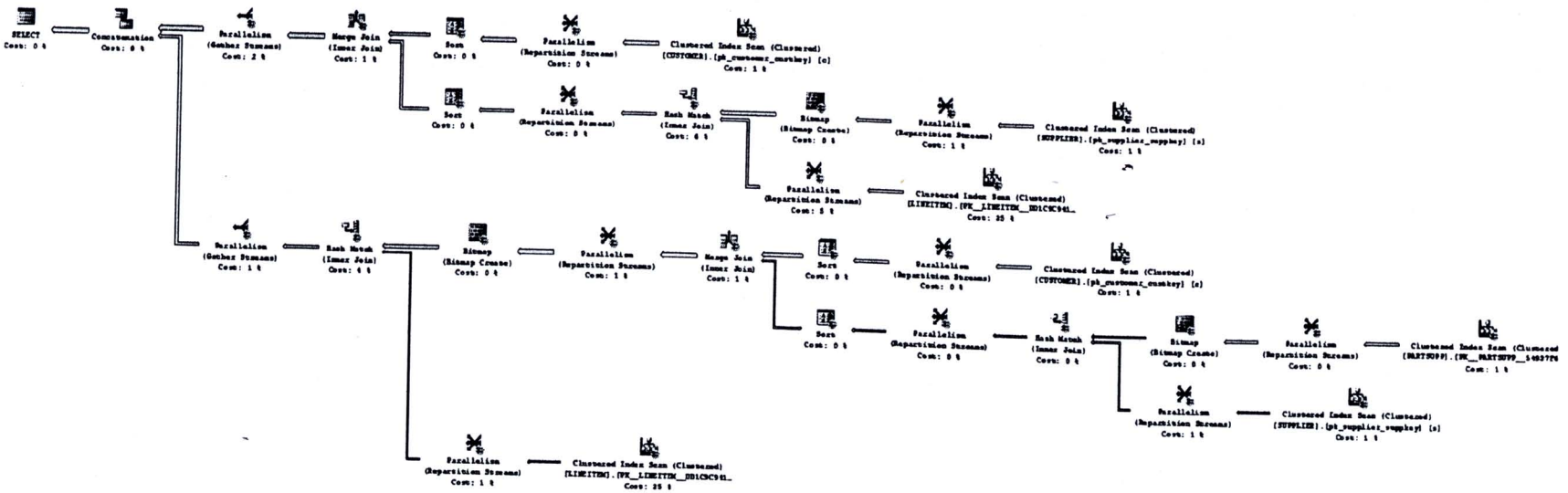
รูปแบบ Join Factorization เป็นการประยุกต์สำหรับการสอบถามที่มี Union และ Union all อยู่ในการสอบถาม โดยการแปลงการสอบถามจะนำการสอบถามที่มี Union และ Union all ที่มีการเรียกใช้ตารางเหมือนกันนำไปไว้ในวิว เพื่อที่จะทำให้ออพติไมเซอร์ทำการวิเคราะห์การสอบถามที่จะนำมารวมกันเพียงครั้งเดียว ต่างจากแบบไม่แปลงการสอบถามคือ ต้องทำการประมวลผลการสอบถามการสอบถามทีละการสอบถามแล้วจึงนำผลลัพธ์มารวมกัน ในส่วนการทดลอง รูปที่ 4.20 และ รูปที่ 4.21 แสดงการสอบถามที่นำมาทดลองของการสอบถามแบบไม่แปลงการสอบถามและแบบแปลงการสอบถามตามลำดับ โดยการสอบถามดังกล่าวต้องการที่ทราบถึง วันที่ขนส่งสินค้าและรหัสของผู้ผลิตสินค้าโดยที่ต้องมีส่วนเพิ่มของราคาสินค้ามากกว่า 5000ขึ้นไป และมีค่าใช้จ่ายผู้ผลิตมากกว่า 600 ขึ้นไป และรูปที่ 4.24 และรูปที่ 4.25 แสดงแคนดิเดตเจเนเรชันของการสอบถามแบบไม่แปลงการสอบถามและแบบแปลงการสอบถามตามลำดับ โดยในรูปที่ 4.22 และรูปที่ 4.23 แสดงแผนประมวลผลแบบไม่แปลงการสอบถามและแบบแปลงการสอบถาม ซึ่งเมื่อวิเคราะห์จากรูปแผนประมวลผลจะเห็นว่าแบบไม่แปลงการสอบถามจะทำการเมิร์จและจอยกันมากกว่าแบบแปลงการสอบถาม

```
SELECT l.l_shipdate, s.s_suppkey
    FROM supplier s, lineitem l, customer c
  WHERE l.l_suppkey=s.s_suupkey
AND s.s_nationkey=c.c_nationkey
            AND l.l_extendedprice > 5000

UNION ALL

SELECT l.l_shipdate, ps.s_suppkey
    FROM partsupp ps, supplier s, lineitem l,
                customer c
              WHERE l.l_partkey=ps.ps_partkey
  AND ps.ps_suppkey=s.s_suppkey
  AND s.s_nationkey=c.c_nationkey
               AND ps.ps_supplycost > 600
```

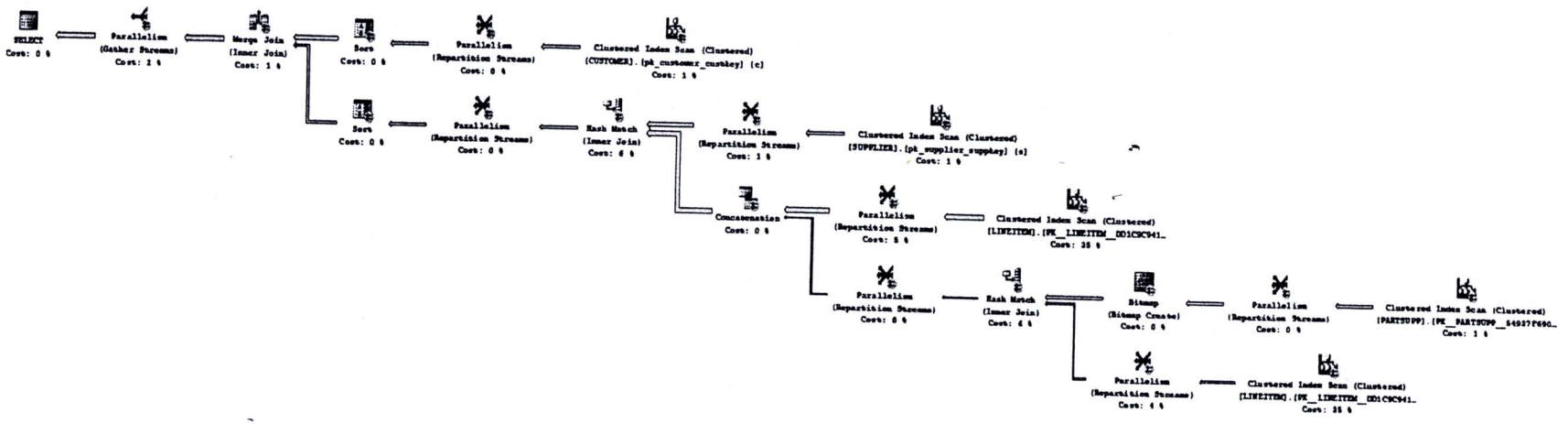
รูปที่ 4.20 แสดงการสอบถามแบบ Join Factorization

```
SELECT v.l_shipdate, s.s_suppkey
FROM supplier s, customer c,
      (SELECT l.l_shipdate, l.l_suppkey
         FROM lineitem l
         WHERE l.l_extendedprice > 5000

         UNION ALL

        SELECT l.l_shipdate,
ps.ps_suppkey
         FROM partsupp ps, lineitem l
         WHERE
l.l_partkey=ps.ps_partkey
             AND ps.ps_supplycost > 600)
  WHERE s.s_suppkey=l.l_suppkey
AND s.s_nationkey=c.c_nationkey
```

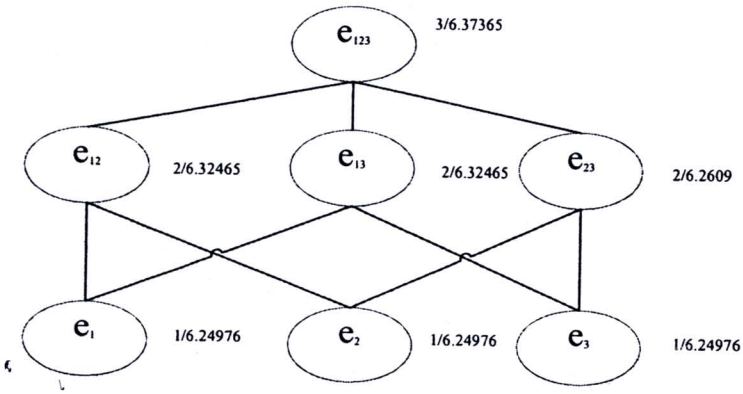
รูปที่ 4.21 แสดงการแปลงการสอบถามแบบ Join Factorization



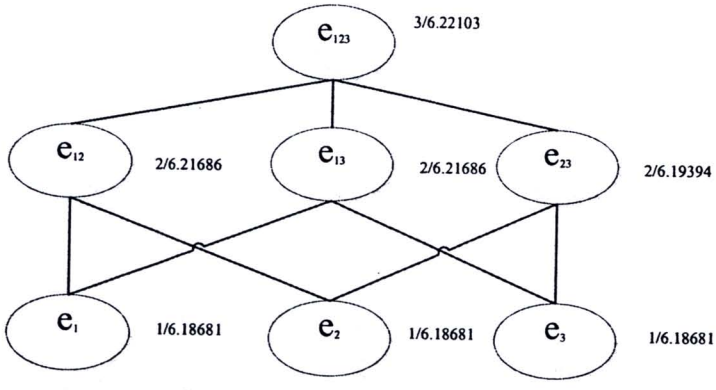
รูปที่ 4.22 แสดงแผนประมวลผลแบบไม่แปลงการสอบถามของ Join Factorization



รูปที่ 4.23 แสดงแผนประมวลผลแบบแปลงการสอบถามของ Join Factorization

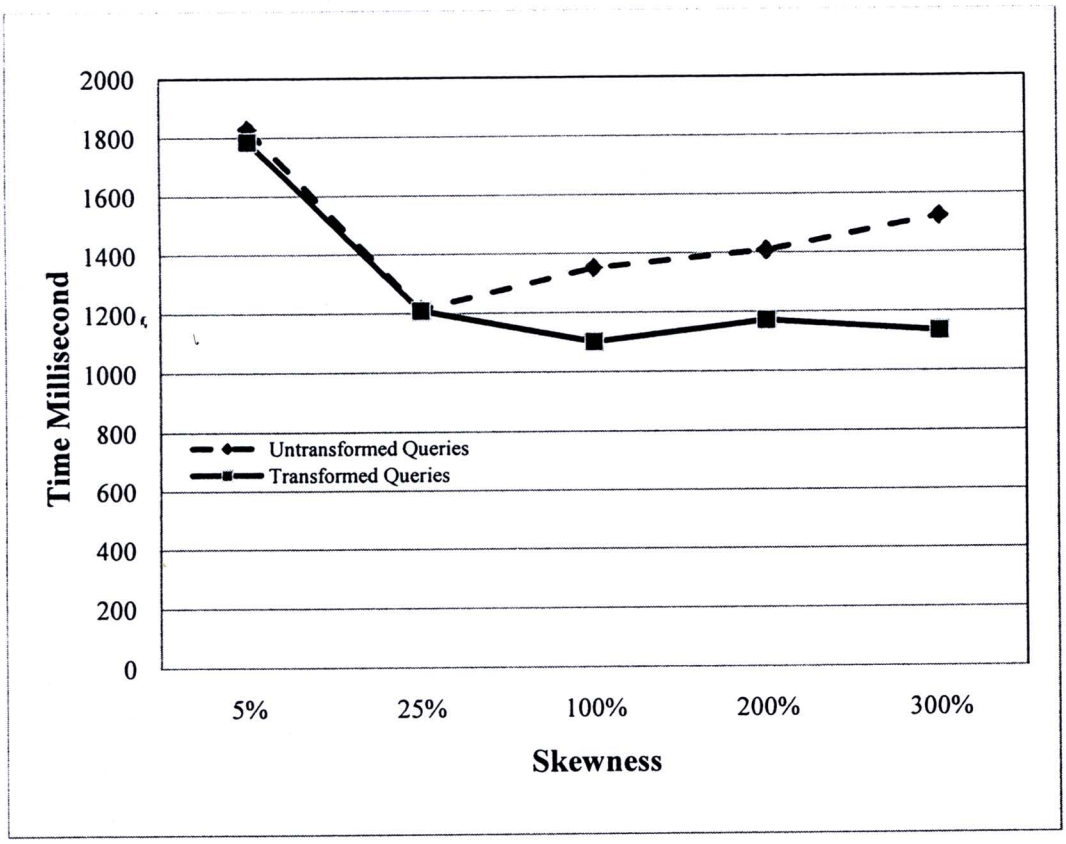


รูปที่ 4.24 แสดงแคนดิเดตเจเนเรชัน Join Factorization



รูปที่ 4.25 แสดงแคนดิเดตเจเนเรชัน Join Factorization

เมื่อทำการคำนวณแคนดิเดตเจเนเรชันดังรูปที่ 4.24 และรูปที่ 4.25 รูปแบบที่จะนำมาทำเคสดิวรี่คือ e_{123} โดยผลการทดลองแสดงเป็นกราฟได้แสดงดังรูปที่ 4.26



รูปที่ 4.26 แสดงกราฟผลการทดลอง Join Factorization

การแปลงการสอบถามแบบ Join Factorization พิจารณาจากกราฟที่เซ็ตข้อมูลของข้อมูลที่ 5% การสอบถามแบบแปลงการสอบถามมีประสิทธิภาพการทำงานที่ใกล้เคียงกับการสอบถามแบบที่ไม่แปลงการสอบถาม แต่แบบแปลงการสอบถามทำงานได้เร็วขึ้นและตั้งแต่ที่ 100% ขึ้นไป เมื่อเซ็ตข้อมูลของข้อมูลเพิ่มมากขึ้น จะเห็นได้ว่าแบบแปลงการสอบถามในรูปแบบ Join Factorization การทำงานที่เซ็ตข้อมูลที่มีข้อมูลไม่มากไม่ได้แตกต่างไปจากการไม่แปลงการสอบถามเลย แต่เมื่อมีเซ็ตข้อมูลของข้อมูลเพิ่มการทำงานของการสอบถามแบบแปลงการสอบถามเร็วขึ้นอย่างชัดเจน ดังนั้นการแปลงการสอบถามรูปแบบจึงเหมาะสมสำหรับเซ็ตข้อมูลที่มีขนาดใหญ่

4.3 สรุป

จากผลการทดลองทั้งหมดสามารถสรุปได้ว่าการนำการแปลงการสอบถามมาช่วยในการคำนวณจำนวนแถวแบบเอ็กแซกท์นั้นสามารถที่จะช่วยเพิ่มประสิทธิภาพได้นั้น จะต้องดูลักษณะของเซ็ตข้อมูลของฐานข้อมูลว่าการแปลงการสอบถามรูปแบบใดที่เหมาะต่อฐานข้อมูลขนาดเล็กหรือรูปแบบใดที่เหมาะกับฐานข้อมูลขนาดใหญ่ หรืออาจจะดูไปถึงผลลัพธ์ของข้อมูลด้วยว่าเหมาะสมที่จะทำการแปลงการสอบถามหรือไม่ เพราะถ้าสังเกตจากผลการทดลองทั้ง 4 รูปแบบนั้น จะทราบว่าการแปลงการสอบถามในบางเซ็ตข้อมูลนั้นไม่ได้ทำงานมีประสิทธิภาพที่ดีไปกว่าการไม่แปลงการสอบถามเลย บางรูปแบบทำงานได้ช้ากว่าการไม่แปลงการสอบถาม โดยกราฟผลการทดลองสามารถบอกได้อย่างชัดเจนว่ารูปแบบการสอบถามรูปแบบใดที่สมควรจะใช้การแปลงการสอบถามในฐานข้อมูลลักษณะใด