

Query Optimization Tutorial

Database Schema

Customer(<u>ID</u> , Name, Type)	10,000
(8, 30, and 5 bytes each)	
Invoice(<u>InvID</u> , CustID, Date, Amount)	10 per customer per month
(12, 8, 10, and 5 bytes each)	
LineItem(<u>InvID</u> , <u>LineNo</u> , ProdID, Qty)	10 per invoice
(12, 5, 20, and 5 bytes each)	
Product(<u>ProdID</u> , Description, ProdType)	1000
(20, 40, and 5 bytes each)	

Assume that each pointer uses 10 bytes. Each block is 256 bytes. Further, there are 10 distinct values of ProdType in Product, and 5 distinct values of Type in Customer.

- a) How many tuples/blocks are there in each table?

Customer
Invoice
LineItem
Product

- b) How many tuples are there in the natural join of Customer, Invoice, LineItem, and Product? Also, how many block in each join?

1. |Customer join Invoice| =
2. (1.) join |LineItem| =
3. (2.) join Product| =

Which relation affects the size of the result the most?

- c) What would be the cost (in block reads) of computing this natural join, step by step, in the sequence indicated (1->2->3)? Is there another sequence that would cost more?

Note 1 The sort-merge join is to be used ($n + n \log n + m + m \log m$) with log based 2.

Note 2 Also, assume that the intermediate query result is written to the disk each time.

Query Optimization Tutorial

- d) Suppose that a flat-file index is to be created for Invoice.CustID, re-assess the cost of the join number 1 (Customer join Invoice).

Note from now on, the join number 1 can use index if needed.

Cost index = $n + 10k(\log_2(i) + 9 + m)$ and $N = 2000$ blocks

BFR I = 14

Entry invoice = 1,200,000

$I = 1,200,000 / 14$

= 85,715 blocks

Index block = $120 / 14 = 9$

Entry M = 120

the cost is 1,462,000.00

- e) Suppose we want to know the types of customers which have bought a given type of product (widget) in July. How many tuples would you expect in the result?

Ans. at most 5

- f) From the query in e), what joins are remaining, in increasing order of cost?

cus = 10,000
 invoice = 100,000
 line = 100,000
 product = 1000

invoice cus
 (10 x 10,000)
 line invoice
 (10 x 100,000)

A = Customer join invoices (for July),
 cost = Join(2000, 14,286) = 235,398.00,
 the number of tuples/blocks in result 14,286.
 (above join can use index if more efficient)

B = invoices (for July) join LineItem,
 cost = Join(14,286, 2,000,000) = 44,074,604.00,
 the number of tuples/blocks in result 2,000,000.

C = (LineItem for July) join (products of given type),
 cost = Join(166,667, 334) = 3,057,982
 the number of tuples/blocks in result 166,667.

The final join can be done on A and C:
 A join C, cost = Join(14,286, 166,667) = 3,269,241
 the number of tuples/blocks in result 166,667.

Query Optimization Tutorial

g) From the query in d), its SQL statement is:

```
SELECT Type
FROM   Customer, invoice, LineItem, Product
WHERE  Invoice.CustID = Customer.ID AND
       LineItem.InvID = Invoice.InvID AND
       Product.ProdID = LineItem.ProdID AND
       Product.ProdType = Widget AND
       Invoice.Date.Month = July
```

Construct a query tree for this query, and show the steps in heuristic optimization.

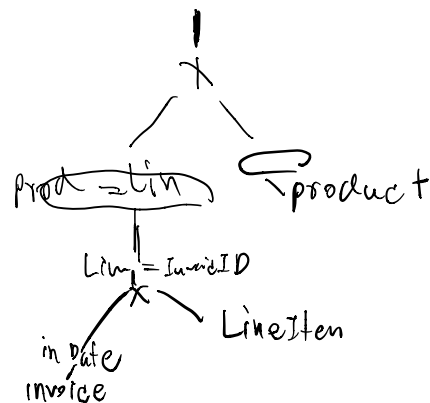
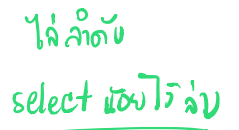
1. $\eta =$ เลื่อนเฉพาะจุดที่บิดหัก

កំណត់សម្គាល់
៥២៥២

①

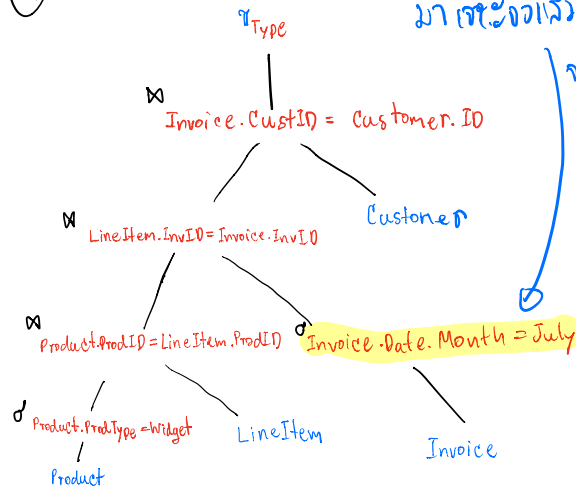


③

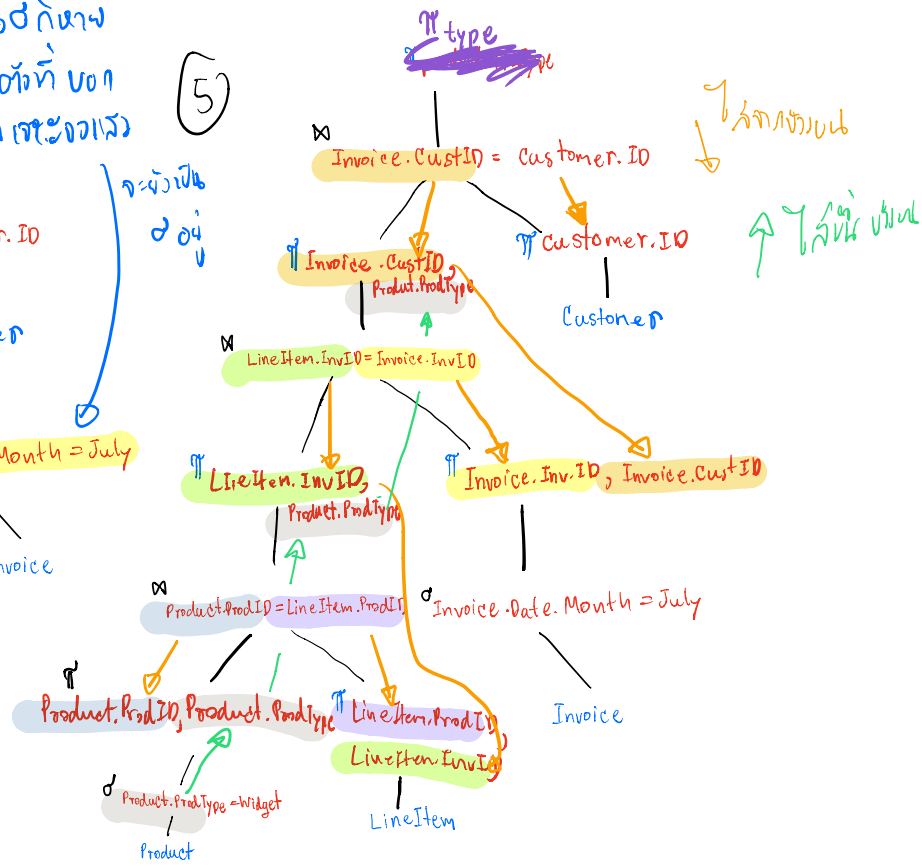


④

เปลี่ยน $\times \rightarrow \bowtie$ แล้วต่อสัปดาห์
ทุกวัน ต่อที่ พอ
มาเจอข้อแล้ว



⑤



Query Optimization Tutorial

- h) From the optimized query tree in g), suppose that the system can read and write data from disk with 40,000 and 20,000 block/second respectively. How long does it take to execute the query?

From step e in g)

step		cost
1	$O(n+\log N)$	343.00
2	$O(n)$	334.00
3	$O(n)$	166,667.00
4	join	3,060,909.00
5	$O(n + \log N)$	14,300.00
6	$O(n)$	14,286.00
7	$O(n)$	166,667.00
8	join	3,269,241.00
9	$O(n)$	166,667.00
10	$O(n)$	2,000.00
11	join	3,081,706.00
12	$O(n)$	166,667.00
total cost		1,384,647 block

④ 166

10,09787

Read disk take time $1,384,647/40,000 = 35$ second

25.7

Write disk take time $1,384,647/20,000 = 70$ second

50.5

Ans. This query take time ~~105~~ second

$$25.7 + 50.5 = 76.2 \text{ seconds}$$

① ลำดับ 1 ถึง 12

② คัดทุก = step



วิธีที่ง่าย
. สักที่ พอ
มาเจอข้อนี้

(5)

er.ID

ver

Month = July

Invoice

2

1

Product

π type

12

11

9

8

7

4

3

1

Invoice.CustID = Customer.ID

Invoice.CustID, Product.ProdType

LineItem.InvID = Invoice.InvID

LineItem.InvID, Product.ProdType

Product.ProdID = LineItem.ProdID

Product.ProdID, Product.ProdType

Product.ProdType - widget

customer.ID

Invoice.InvID

Invoice.InvID, Invoice.CustID

Invoice.Date.Month = July

LineItem.ProdID

LineItem.InvID

Customer

Invoice

Invoice

Invoice

Invoice

Invoice

↓ สักที่ พอ

↑ สักที่ พอ

