Adam Norton, Guanglei Cao
7 November 2013

Project 3

Exercise 1:

a. One

b.

Relation	Disk Pages
customers	471
orderlines	384

Index	Disk Pages
Customers_pkey	57
Orders_pkey	35

c. Two mode btree index: city, creditcard

d.

```
COMPARE:
       attname         |    with Postgres catalog       without Postgres
catalog
----------------------+------------------
 address1              |          20000               20000
 address2              |              1               1
 age                   |             73               73
 city                  |          20000               20000
 country               |             11               11
 creditcard            |          20000               20000
 creditcardexpiration  |             60               60
 creditcardtype        |              5               5
 customerid            |          20000               20000
 email                 |          20000               20000
 firstname             |          20000               20000
 gender                |              2               2
 income                |              5               5
 lastname              |          20000               20000
 password              |              1               1
 phone                 |          20000               20000
 region                |              2               2
 state                 |             52               52
 username              |          20000               20000
 zip                   | 9499.99988079071             9500
```

Exercise 2:

a. The estimated number of rows is 995. This is the same as the actual value.

b. The optimizer calculated the expected execution cost of 721 as follows:

(disk_pages_read * seq_page_cost) + (rows_scanned * cpu_tuple_cost) + (rows_scanned * cpu_operator_cost)
= (471 * 1) + (20000 * 0.01) + (20000 * 0.0025)
= 471 + 200 + 50
= 721

c.

$$\sigma_{country='Japan'}(\text{Customers})$$

Exercise 3:

a. customers_country occupies 59 disk pages.

b. The optimizer has chosen a sequential scan. It has an expected execution cost of 721.

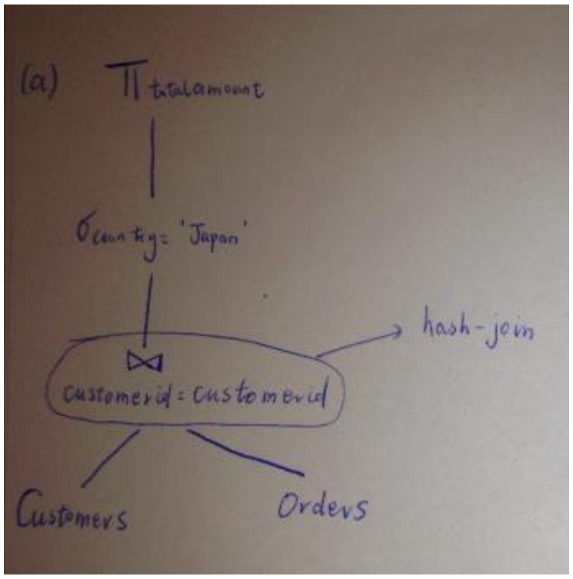
c. Adding this non-clustered index did not change the execution plan. This is because a large number of rows were expected to be returned. Therefore, the optimizer decided to do a sequential scan to avoid potentially having to read a new page for every tuple.

d. The query optimizer chose to do an index scan with an expected cost of 56.66.

e. The new plan, using the clustered index, is expected to be significantly faster since it will use the index rather than running a sequential scan. It has chosen to do this since the index is now clustered and the ordering of the data entries in the index is the same as the ordering of the data records on disk. This means that we will read each data entry on one page before moving on to the next page.

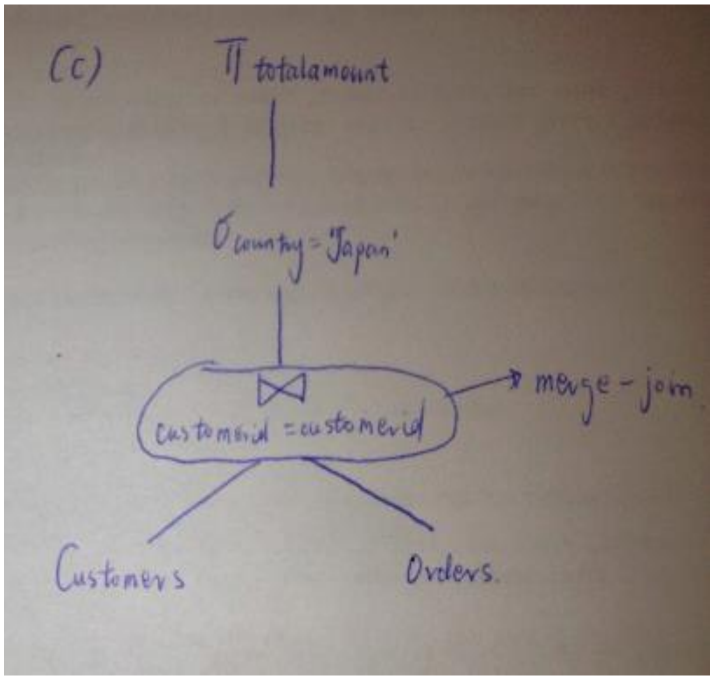
Exercise 4:

a.



b. use hash join; cost is 1004.41. Cardinality is 597.

c. use merge join, cost is 1874.53.



d. use nested loop, cost is 5749.36

Exercise 5:
a. hash join algorithm; cost is 1501.36
b. The query optimizer uses merge join algorithm. The total expected cost is 2265.19.

After disabling the merge join algorithm, the query optimizer uses hash join algorithm. The total expected cost is 3813.54.
c. Merge join is achieved by first sorting the two tables and then joining them, while hash join is first joining and then sorting. For the first query, the merge join does not give the same sorted order as required in the query, therefore wastes time on meaningless sorting; for the second query, the sorted order of merge join is consistent with that in the query. Moreover, the number of elements is too large for an efficient sort after hash join. In conclusion, hash join is best for the first query and merge join is best for the second query.

Exercise 6:
a. The estimated execution cost is 5001021.
b. Solution is in the P3queries.sql file.
c. Solution is in the P3queries.sql file.
d. The estimated startup cost is 2452 and the estimated execution cost is 2924.29. Even though there is a higher startup cost, the total estimated time is faster than the previous query since it only needs to scan through the orders table once when creating the view.

<u>Exercise 7</u>:

a. The estimated startup cost is 614926.51 and the estimated execution cost is 614927.01.

b. Solution is in the P3queries.sql file.

c. Solution is in the P3queries.sql file.

d. The estimated startup cost is 12893.86 and the estimated execution cost is 12909.6. This query is much better than the original version since we have eliminated the correlated subqueries and thus avoided repeated scanning though the customer and order tables. The join optimizer is much more efficient than the nested subqueries.