COMP0022: Database and information systems (PG – Term 1)

Query processing exercises - solutions

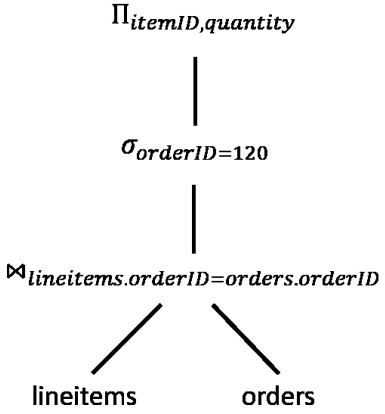
The following exercises make use of the schema of the Shop database from the SQL exercises:

customers (<u>custID</u>, firstname, familyname, town, state)
items (<u>itemID</u>, description, unitcost, stocklevel)
lineitems (<u>orderID</u>, <u>itemID</u>, quantity, despatched)
orders (<u>orderID</u>, custID, date)
supplieritems (<u>supplierID</u>, <u>itemID</u>)
suppliers (<u>supplierID</u>, supplierName, supplierAddress, phoneNo, Delivers)

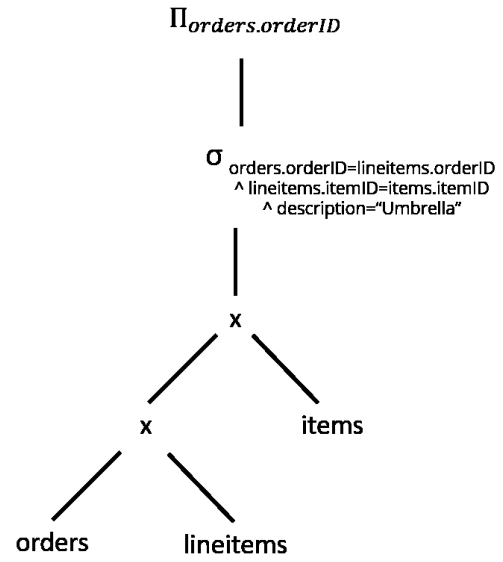
Part 1

Translate the following SQL queries into (a) a relational algebra statement, and (b) a corresponding relational algebra tree.

1. SELECT * FROM customers WHERE state = "Arizona"
   a. $\sigma_{state=Arizona}(customers)$
   b. 

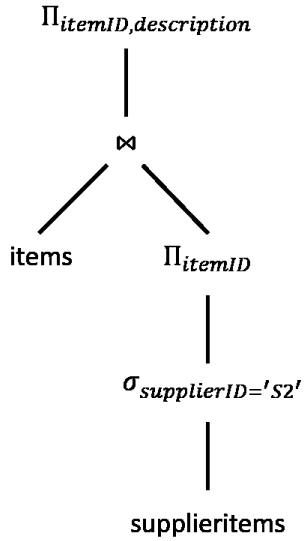
2. SELECT itemID, quantity
   FROM lineitems JOIN orders USING (orderID)
   WHERE orderID = 120
   a. One possibility:
   $\Pi_{itemID,quantity}\sigma_{orderID=120}(lineitems \bowtie_{lineitems.orderID=orders.orderID} orders)$
   b. 

3. SELECT o.orderID FROM orders o, lineitems l, items i
   WHERE o.orderID = l.orderID AND l.itemID = i.itemID AND description = "Umbrella"
   a. One possibility: $\Pi_{o.orderID}\sigma_{o.orderID=l.orderID\wedge l.itemID=i.itemID\wedge description="Umbrella"}(o \times l \times i)$



b.

4. SELECT itemID, description FROM items
   WHERE itemID IN (SELECT itemID FROM supplieritems WHERE supplierID = 's2')
   a. One possibility: $\Pi_{itemID,description}(items \bowtie \left(\Pi_{itemID}\sigma_{supplierID='s2'}(supplieritems)\right))$



b.

Use relational algebra transformation rules to generate at least two alternative executions of queries 2-4 above.

2. Some possibilities:
   $\Pi_{itemID,quantity}(\sigma_{orderID=120}(lineitems) \bowtie \sigma_{orderID=120}(orders))$
   $\Pi_{itemID,quantity}(\Pi_{itemID,quantity,orderID}(\sigma_{orderID=120}(lineitems)) \bowtie \Pi_{orderID}(\sigma_{orderID=120}(orders)))$

3. Some possibilities:
   $\Pi_{orderID}\sigma_{description=\text{"Umbrella"}}((o \bowtie l) \bowtie i)$
   $\Pi_{orderID}((o \bowtie l) \bowtie \sigma_{description=\text{"Umbrella"}}(i))$

4. Some possibilities:

$$(\Pi_{itemID,description}(items)) \bowtie \left(\Pi_{itemID}\sigma_{supplierID='s2'}(supplieritems)\right)$$

$$\sigma_{i.itemID=s.itemID}(\Pi_{itemID,description}(items) \times \Pi_{itemID}\sigma_{supplierID='s2'}(supplieritems))$$

Part 2

Finally, estimate the cost of the answers given in the previous sections in terms of number of disk accesses.

Assume the following:

There are 7000 rows in the customers table; 500 rows in the items table; 4000 rows in the lineitems table; 3000 rows in the orders table, 500 rows in the supplieritems table, and 5 rows in the suppliers table.
There are 50 states over which customers are uniformly distributed.
There are approximately 10 items with the description “Umbrella”.

No indexes or sort keys are used.
The results of all intermediate operations are stored on disc. The final result, however, is returned directly and not stored.
Tuples are accessed and written back one at a time, rather than in blocks.
Main memory is large enough to process entire tables for each relational algebra operation. No tuples, once loaded, will need to be kicked out.

1. 7000 disk accesses
2. $\Pi_{itemID,quantity}\sigma_{orderID=120}(lineitems \bowtie_{lineitems.orderID=orders.orderID} orders)$
   4000+3000 (reads) + 4000 (writes) + 4000 (reads) + 2† (writes) + 2 (reads) = 15004 disk accesses
   † Using 2 here because the number of items associated with orderID 120 is estimated to be somewhere between 1 and 2 on average based on the number of lineitems vs. number of orders
   $\Pi_{itemID,quantity}(\sigma_{orderID=120}(lineitems) \bowtie \sigma_{orderID=120}(orders))$
   4000+2 (reads/writes) + 3000+1 (reads/writes) + 3 (reads) + 3 (writes) + 3 (reads) = 7012 disk accesses
   $\Pi_{itemID,quantity}(\Pi_{itemID,quantity,orderID}(\sigma_{orderID=120}(lineitems)) \bowtie \Pi_{orderID}(\sigma_{orderID=120}(orders)))$
   4000+2 (reads/writes) + 2+2 (reads/writes) + 3000+1 (reads/writes) + 1+1 (reads/writes) + 3+3 (reads/writes) + 3 (reads) = 7018 disk accesses
3. $\Pi_{o.orderID}\sigma_{o.orderID=l.orderID \wedge l.itemID=i.itemID \wedge description=\text{Umbrella}}(o \times l \times i)$
   4000+3000+500 (reads) + 4000*3000*500 (writes) + 4000*3000*500 (reads) + 60 (writes) + 60 (reads) = 12,000,007,620 disk accesses
   (Assumes three-way product can be done in one operation. If only binary products can be performed, need extra work. Estimates 10 umbrellas out of 500 items are involved in 1/50 of the orders = ~60 orders.)
   $\Pi_{orderID}\sigma_{description=\text{"Umbrella"}}((o \bowtie l) \bowtie i)$
   3000+4000 (reads) + 4000 (writes) + 4000+500 (reads) + 4000 (writes) + 4000 (reads) + 60 (writes) + 60 (reads) = 23,620 disk accesses

$\Pi_{orderID}((o \bowtie l) \bowtie \sigma_{description=\text{"Umbrella"}}(i))$

3000+4000 (reads) + 4000 (writes) + 500 (reads) + 10 (writes) + 4000+10 (reads) + 60 (writes) + 60 (reads) = 15,640 disk accesses

4. $\Pi_{itemID,description}(items \bowtie \left(\Pi_{itemID}\sigma_{supplierID='s2'}(supplieritems)\right))$

   500 (reads) + 100‡ (writes) + 100+100 (reads/writes) + 500+100 (reads) + 100 (writes) + 100 (reads) = 1,600 disk accesses

   ‡ Estimated that S2, 1 supplier out of 5, will be involved with 1/5 of the listings of supplieritems = ~100 tuples

   $(\Pi_{itemID,description}(items)) \bowtie \left(\Pi_{itemID}\sigma_{supplierID='s2'}(supplieritems)\right)$

   500+500 (reads/writes) + 500 (reads) + 100 (writes) + 100+100 (reads/writes) + 500+100 (reads) = 2,400 disk accesses

   $\sigma_{i.itemID=s.itemID}(\Pi_{itemID,description}(items) \times \Pi_{itemID}\sigma_{supplierID='s2'}(supplieritems))$

   500+500 (read/writes) + 500 (reads) + 100 (writes) + 100+100 (reads/writes) + 500+100 (reads) + 500*100 (writes) + 500*100 (reads) = 102,400 disk accesses