

Query processing exercises

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

customers (custID, firstname, familyname, town, state)
items (itemID, description, unitcost, stocklevel)
lineitems (orderID, itemID, quantity, despatched)
orders (orderID, custID, date)
supplieritems (supplierID, itemID)
suppliers (supplierID, 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"
2. SELECT itemID, quantity
FROM lineitems JOIN orders USING (orderID)
WHERE orderID = 120
3. SELECT o.orderID FROM orders o, lineitems l, items i
WHERE o.orderID = l.orderID AND l.itemID = i.itemID AND description = "Umbrella"
4. SELECT itemID, description FROM items
WHERE itemID IN (SELECT itemID FROM supplieritems WHERE supplierID = 's2')

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

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 line items 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.