CS-245 Database System Principles – Winter 2002
Assignment 4

Due at the beginning of class on Tuesday, February 19

- State all assumptions and show all work.
- Subscribe to cs245@lists.stanford.edu to receive clarifications and changes.
- You can email questions to cs245-staff@cs.stanford.edu

Problem 1. (15 points)

Assume that we have two relations, R and S. Let p be a predicate containing only R attributes. Let q be a predicate containing only S attributes. Let m be a predicate containing only attributes from both R and S. Demonstrate how to derive the following relational algebra equivalence rules from simpler rules. You may use any rule mentioned in the book or in lecture, but state explicitly which rule you are using.

a. $\sigma_{p \wedge q \wedge m} (R \bowtie S) = \sigma_m [(\sigma_p R) \bowtie (\sigma_q S)]$

b. $\sigma_{p \vee q} (R \bowtie S) = [(\sigma_p R) \bowtie S] \cup [R \bowtie (\sigma_q S)]$

Problems 2-4.

For the next three problems, use the following schema, for an on-line bookstore:

```
Cust (CustID, Name, Address, State, Zip)
Book (BookID, Title, Author, Price, Category)
Order (OrderID, CustID, BookID, ShipDate)
Inventory (BookID, Quantity, WarehouseID, ShelfLocation)
Warehouse (WarehouseID, State)
```

This schema represents customers (Cust) and books (Book). When a customer buys a book, a tuple is entered into the Order table. The inventory of books is recorded in the Inventory table, which records the quantity, warehouse and shelf location for each different book. The Warehouse table records the state where each warehouse is located in. You may assume that these relations are sets. You may assume Price is a numeric value. You may also assume that ShipDate is an integer representation of a date. Finally, you may assume that there is a constant, `:today`, that contains the integer representation of today's date.

Problem 2. (15 points)

Write relational algebra expressions for the following SQL queries.

a. Find books under $10:

```
SELECT Title, Author
FROM Book
WHERE Price < 10.00;
```

b. Find books ordered by customer 53:

```
SELECT Title, Author
FROM Cust NATURALJOIN Order NATURALJOIN Book
WHERE CustID = 53;
```

c. Find other books purchased by the same customers that purchased book 1085:

```
SELECT B.Title, B.Author, B.BookID
FROM Book B, Order Q, Order R
WHERE Q.BookID=1085 AND Q.CustID=R.CustID AND R.BookID=B.BookID AND
B.BookID <> 1085;
```

Problem 3. (30 points)

For each of the following relational algebra expressions:

- Show the logical query plan for the given expression, drawn as a tree of relational operators with relations at the leaves.
- Transform the logical query plan you have drawn by pushing selections and projections down as far as you can. Show the result as another logical query plan drawn as a tree of relational operators with relations at the leaves. (For simplicity, you should not introduce projections over attributes other than those attributes already projected.)
- Write the transformed query plan (with the selections and projections pushed down) as a relational algebra expression.

a. Find books by "Oprah Winfrey" bought by customers in California for under $20.

$\pi_{\text{Title,BookID}}$ ($\sigma_{\text{State='CA'} \wedge \text{Author='Oprah Winfrey'} \wedge \text{Price<20}}$ (Cust $\bowtie$ Order $\bowtie$ Book))

b. Find books in warehouse 12 that are supposed to be shipped today.

$\pi_{\text{BookID,ShelfLocation}}$ ($\sigma_{\text{ShipDate=:today} \wedge \text{WarehouseID=12}}$(Order $\bowtie$ Inventory))

c. Find books that customer 89 has not bought yet but that are in the same category as some book customer 89 has bought already.

$\pi_{Title,Author,BookID}(\sigma_{CustID=89}[(Book \rhd\lhd (\pi_{Category,CustID}(Order \rhd\lhd Book)\ )\ ) -$
$(\pi_{CustID,BookID,Title,Author,Price,Category}(Order \rhd\lhd Book))])$

Problem 4. (30 points)

For each of the following logical query plans, label each relational operator with the expected result size: the number of tuples. You may assume "containment of value sets" and "preservation of value sets." You may also make the assumption stated in the lecture notes, that "Values in select expression Z=val are uniformly distributed over possible V(R,Z) values." Be sure to justify each value you calculate. Note that since we are only asking for the number of tuples, you do not need to examine the effect of projections.

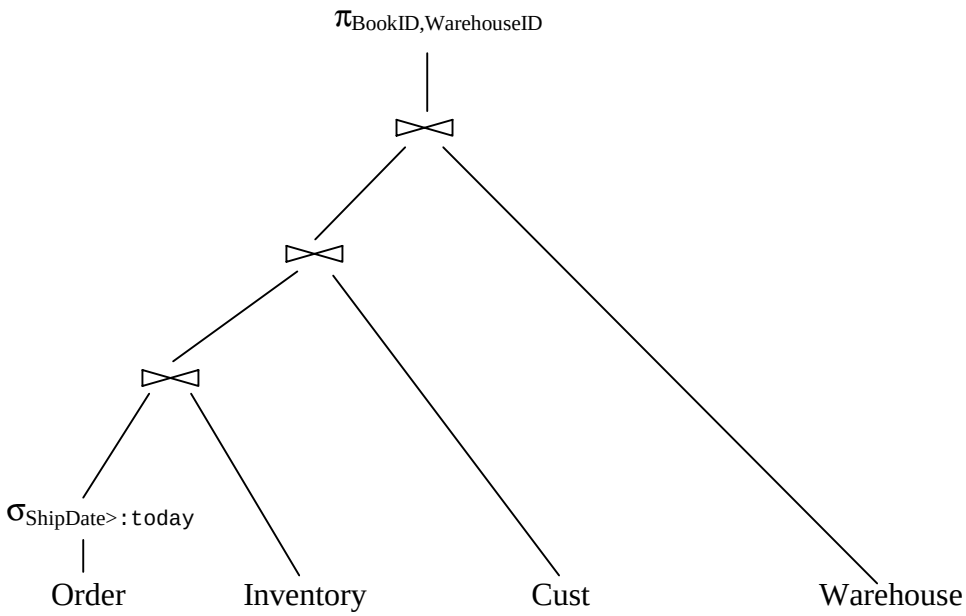
Now make the assumption stated in lecture that "Values in select expression Z=val are uniformly distributed over a domain with DOM(R,Z) values." Which, if any, of the values you calculated change? Give the new values.

Use the following statistics:

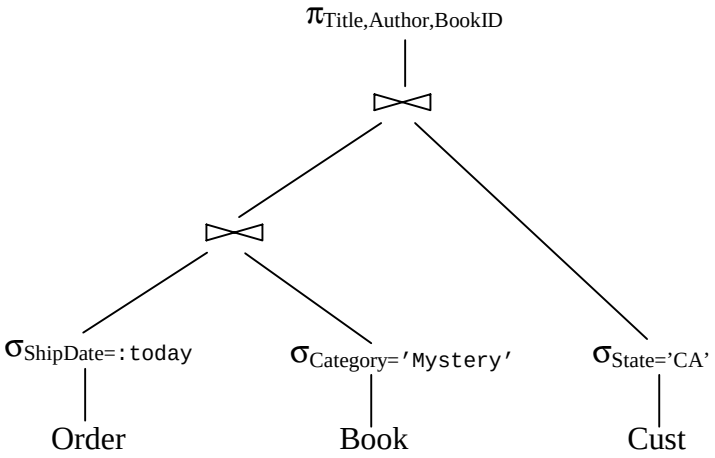
- T(Order) = 30,000
- V(Order,BookID) = 1,000
- V(Order,CustID) = 3,000
- V(Order,ShipDate) = 2,500
- DOM(Order,ShipDate) = 10,000
- T(Inventory) = 100,000
- V(Inventory,BookID) = 2,000
- V(Inventory,WarehouseID) = 50
- T(Cust) = 3,000
- V(Cust,CustID) = 3,000
- V(Cust,State) = 50
- DOM(Cust,State) = 50
- T(Warehouse) = 50
- V(Warehouse,WarehouseID) = 50
- V(Warehouse,State) = 50
- T(Book) = 1,000
- V(Book,BookID) =1,000
- V(Book,Category) = 20
- DOM(Book,Category) = 25

(HINT: You have enough information in these statistics to solve this problem.)

a. Find books that have yet to be shipped that are in a warehouse in the same state as the customer that ordered them.



b. Find books from category “Mystery” ordered by people in California that shipped today.



Problem 5. (10 points)

Imagine a large relation R stored in *m* disk blocks where *n* tuples are stored per block. Consider a query that involves a selection over R. Using statistics, we deduce that the selection is likely to return at least *m* tuples, and we know that the tuples are uniformly distributed throughout the disk blocks. In other words, we can reasonably expect that each disk block will contain at least one tuple that will be a part of our selection result. The basic way to answer the selection is to scan the table, one block at a time, retrieving tuples that match the selection; the cost of this plain table scan operation is *m* I/Os. How much improvement can we expect in performance if we use an index to support our selection instead of a plain table scan?