

Database Systems

Project Deliverable 4

Part A :

- Given two large relations $R(X_1, X_2, X_3, \dots, Y, \dots)$ and $S(\dots, Y, Z_1, Z_2, Z_3, \dots)$, implement Sort-Merge join on R,S such that $R.Y = S.Y$

Part B :

- Calculate $V(R,a) \Rightarrow$ Number of unique values for attribute 'a' in Relation 'R'.
- $V(R,a)$ should be updated if insertion(s) are made to the relation.
 - You are not supposed to scan complete table again to get updated value of $V(R,a)$ after insertion.
 - Store required meta-data, which can help you in estimating updated $V(R,a)$ value without scanning completing table.
- Implement a method "int V(String tableName, String attributeName)" in DBSystem class which will return the value of $V(R,a)$.
- No error handling is required. You can assume valid inputs.

Part C :

- Suppose we join more than two relations like
 - Join R , S , T \Rightarrow
 - $TEMP \Leftarrow Join\ R,\ S$
 - $ANS \Leftarrow Join\ Temp,\ T$
 - OR
 - $TEMP \Leftarrow Join\ S,\ T$
 - $ANS \Leftarrow Join\ R,\ Temp$
- To get which of the available order(s) of join is better, we need to do cost estimation.
- Implement a cost estimation algorithm which can predict the optimum order of joining multiple tables (without actually joining/scanning them, off course :P)

Sample input and output :

PART - A

Input

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID=Customers.CustomerID;
```

Output: the result of the join.

Part B

We will be calling the method $V(R,a)$ defined in DBSystem to evaluate.

Part C

Note: If any of input query contains join of more than two tables in FOR clause, output only the order of JOIN (Part -C). Otherwise, output result (Part -A) of the query.

Input

```
SELECT Items.ItemID, Orders.OrderID, Customers.CustomerName,  
Orders.OrderDate  
FROM Orders  
JOIN Customers  
ON Orders.CustomerID=Customers.CustomerID  
JOIN Items  
ON Items.ItemID=Order.ItemID;
```

Output

- A bracketed expression representing the order of join with its cost.

((Items,Orders), Customers)

xxxx - where xxxx is the cost of joining