Lecture 11 Notes - Monday 10/31/16

Reading Quiz

Question 1: Ans = B
Question 2: Ans = D
Question 3: Ans = D
Question 4: Ans = E
Question 5: Ans = C

Notes

A *view* is basically a way to name a SQL query.

Concept Question 1: E. In the first view (*All_Employee_View*) we are filtering out *ssn* and *salary*. In the second view (*Manager_Employee_View*) we are filtering out records with "Executive" *role*s.

So far we've seen two applications for views: "security" views -- hiding certain columns or records from the original table in the view; and "complexity hiding" views -- these let us see the results of a common but complex query easily by giving it a single name, instead of writing a whole long query every time, and people just using the view don't even need to understand all of the SQL of the complex query to use the view.

Concept Question 2: A. The query optimizer is smart enough to see that we don't need to use both of the tables that make up the view *CustomerSales* to complete this query. In fact, we only need to use the *Orders* table, as we're just selecting *customer_id* and filtering using *store*, both of which columns are in *Orders.* So only *Orders* will be used to complete this query. The query optimizer decides to only use the tables we need to complete the query, as we just saw in the slides before (*Query Modification*).

So far we've just talked about *virtual views*, which is just when we consider a view as a name placed on a query -- when we query this view, we're running the select statement that defines the view as a sort of subquery in the *from* clause of our query.

But there are also *materialized views*, which are stored on their own, kind of just like a brand new table, but created based on old tables.