HOMEWORKS

Homework 6

Views

When you create a view, say `my_view`, you need to define a "SELECT-statement". This is used to define what the contents of `my_view` will be, similar to what a label does to a box.

Now, consider the following:

```
// The `accounts` relation contains 100 records for age > 50
CREATE VIEW my_view AS SELECT * FROM account WHERE age > 50;
CREATE VIEW MATERIALIZED my_mat_view AS SELECT * FROM account
WHERE age > 50;
```

Say that these two views are created virtually at the same time, and that the number of records of `accounts` did not change in-between. Now, we execute the following two queries:

```
SELECT * FROM my_view;      -- 1
SELECT * FROM my_mat_view;  -- 2
```

Answer the following:

1. Will the result of both queries be the same?
2. Now, say that we add 100 more records to `account` with age > 50, after the CREATE VIEW-statements. If we then run the two queries 1 and 2, are they still the same? Why/why not?
3. When would it be favourable to use a VIEW over a Query? What about a materialised VIEW over a regular VIEW?

Limit your answer to max 150 words (for all three points).

HOMEWORKS

Homework 6 P+

Booking race

When working with a library system a key functionality is the ability to rent books. Back in the day this was rather easy to handle, someone showed up to simply say “Hey I want this book!” and a librarian would mark it down. Now, in our case the booking should be more sophisticated where people could rent a book on their own, remotely. This introduces a range of interesting problems.

Consider the following scenario: “1000 students are using your LMS at the same time (all within a short time *d*), trying to lend your books; but you only have 100 books in your DB”. The pigeon-hole principle says that there should be some potential double-bookings, how do we solve this? Please answer the following:

- Utilising a transaction is most likely the go-to solution for this case. But which of the four isolation levels: Serializable, Repeatable reads, Read committed, Read uncommitted, should we utilise? **For each isolation level, please describe if the isolation level prevents double-bookings.**
- In some cases however transactions (read isolation levels) will infer some hard constraints on the usability of the lending. **What major side-effect will a transaction of “isolation level Serializable” come with?**

Please limit your answers to 300 words (total; for both questions)