Section 8

Lee Lee Choo

Feb 28th 2013

- Transaction Review
- Homework Introduction
- Common Questions and Mistakes
- CSE344 Video Rental Store in Action

ACID or ALKALINE

ACID!

A Atomicity
C Consistency
I Isolation
D Durability

Transactions

begin transaction;

...

...

...

commit; /rollback;

Isolation Levels in SQL

- SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
- SET TRANSACTION ISOLATION LEVEL READ COMMITTED
- SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
- SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

Homework 7 (1)

Files to modify:

- Query.java
- dbconn.properties

File to create:

- setup.sql

Homework 7 (2)

PreparedStatement, Connection, String

Example 1: A query that returns all manager.

```
Connection conn;
... ...
String  query = "select * from Employee where position='manager'";
PreparedStatement managers = conn. prepareStatement(query);
```

Homework 7 (3)

Example 2: Find all the employees that have ? as their last name.

String query = "select * from Employee where lname = ? ";

- ? represents parameter that we are going to fill in later.
- See code given under transaction_login

Homework 7 (4)

Forcing Transaction Interleaving:

```
Scanner console = new Scanner (System.in);
console.nextLine();
```

Things to keep in mind for the homework

common questions and mistakes…

1 Customer database design

- Plan is a keyword in SQL, so we cannot create a table name Plan, change it to any name that is relevant.
- If we have a primary key in the table, the clustered indices are created automatically.
- Minimum 8 tuples means 8 tuples in total not 8 tuples for each table.
- Make sure the statements in setup.sql are in the right order.
- The date field in rental database should be a valid SQL date type that includes time.

2 Java customer application

- Fastsearch is faster but it is for single word only.
- Print customerinfo = at least the customer name and the number of movies the user can still rent.

MISC

- It is common to get Socket Write Error Exception when running the application.

Fastsearch explanation (1)

Magda: "In fast search, you should really execute three queries only (forget about movie availability): the first query should compute the movie metadata for all movies that match the keyword search, the second query should find the directors for all movies that match the keyword search, and the third one should similarly find the actors for all the movies that match the keyword search. Execute each of these three queries separately. You then need to merge the results of the three queries *in* the Java code. The merge will be easier if you sort the results of the three queries.

There is also a way to actually merge all of this info in a single SQL query but don't worry about that because it's similarly easy to write a very expensive single SQL query. Best to try writing three queries exactly."

Fastsearch explanation (2)

Vaspol: "If you take a close look at the "search" method, you will see that the "search" method is iterating over all the mids returned when you search for the movie. Then, for each of the mids, it will issue 2 queries to the database: getting the actors for that mid, getting the director(s) for that mid. This results in number_of_mids * 2 queries when we do the "search" method, which is expensive.

The idea of "fastsearch" is that we want to reduce the queries being issued to the database. This will give you only a few queries (way less than that of "search" method.) Therefore, it will be faster in the sense that we don't need to connect to the database and run a lot of queries."