**Topic: Transactions**

**Class Participation for Week 8**

**Learning objective:**

* Gain a deeper understanding about concurrency control and protection against failure provided by Transaction Manager in DBMS.

**Instruction:** Answer all questions to receive full credits for this class participation.

See Program 1 and Program 2 below. This program updates the Food database instance (Fig. 1) that is stored in a DBMS that supports ACID properties. A database transaction is an execution of a user program in a DBMS. Each run of a program is considered one transaction.

The Connection.TRANSACTION\_SERIALIZABLE isolation level provides the maximum protection against interference by other concurrently running programs as if all these concurrently running programs run in a serial order.

**try** {

// conn is a valid connection

// fname is a String and has the food name provided by the user.

//

String fname=JOptionPane.*showInputDialog*("Enter food name:");

conn.setAutoCommit(**false**);

conn.setTransactionIsolation(Connection.TRANSACTION\_SERIALIZABLE);

Statement stmt = conn.createStatement();

ResultSet rs;

**int** id=0;

rs = stmt.executeQuery("select max(fid) from food");

**while** (rs.next()) {

id = rs.getInt(1);

}

rs.close(); stmt.close();

// ? indicating that it is a parameterized SQL query

PreparedStatement inststmt =

conn.prepareStatement("insert into food (fid,fname) values(?,?) ");

inststmt.setInt(1, id+1);

inststmt.setString(2, fname);

inststmt.executeUpdate();

inststmt.close();

conn.commit();

} **catch** (SQLException e) {

// when setautocommit is false, if commit() is not called, the

// all SQL statements between the point of failure to the prior commit

// are undone

}

Program 1

**try** {

conn.setAutoCommit(**false**);

conn.setTransactionIsolation(Connection.TRANSACTION\_SERIALIZABLE);

// Static SQL statement uses Statement object

Statement stmt = conn.createStatement();

ResultSet rs;

String tfname;

**int** id=0;

// this query has a problem that fname may not correspond to the

// maximum fid value; however, we do not use tfname for anything

rs = stmt.executeQuery("select max(fid), fname from food");

**while** (rs.next()) {

id = rs.getInt(1);

tffname = rs.getString(2);

}

rs.close();

stmt.close();

// use of a parameterized SQL statement which is a statement with

// the question mark whose value is to be replaced by the

// parameter values given by a user

PreparedStatement inststmt =

conn.prepareStatement("update food set fname=? where fid=?");

String fname=JOptionPane.*showInputDialog*("Enter food name:");

inststmt.setString(1, fname.toUpperCase());

inststmt.setInt(2, id);

inststmt.executeUpdate();

inststmt.close();

conn.commit();

} **catch** (SQLException e) {}

Program 2

**Questions**:

1. What does Program 1 do? **Program 1 Insert a new food item into the food table with the fid value as the highest fid value in the food table + 1.**
2. What does Program 2 do? **Program 2 Changes the fname value of the row with the highest fid value to the uppercase of the input given by the user.**
3. Suppose there are three users: A, B, and C, running the two programs simultaneously. Users A and B run Program 1. User A adds the food name “Hawaiian Pizza” while user B adds the food name “Hummus”. User C runs Program 2 with the food name of “Pad Thai”. There are no other transactions this DBMS is running except these three transactions. What is the content of the Food table if the three transactions complete successfully?

A: fid, fname

30, “Hawaiian Pizza”

B: 31, “Hummus”

C: 32, “PAD THAI”

A: 30, “Hawaiian Pizza”

C: 31, “PAD THAI”

B: 32, “Hummus”

B: 30, “Hummus”

A 31, “Hawaiian Pizza”

C 32, “PAD THAI”

B: 30, “Hummus”

C: 31, “PAD THAI”

A: 32, “Hawaiian Pizza”

C: 30, “PAD THAI”

A: 31, “Hawaiian Pizza”

B: 32, “Hummus”

C: 30, “PAD THAI”

B: 31, “Hummus”

A: 32, “Hawaiian Pizza”

1. Starting with the instances of the database in Figure 1, what is the content of the Food table if only User A (running Program 1 adding Hawaiian Pizza) and User C (running Program 2 with “Pad Thai”) completed the transactions successfully, i.e., the programs have passed the commit statement before the system crashed but User B’s program (running Program 1 adding Hummus) was still running and has not reached the commit statement when the system crashed? Once the DBMS restarts, what is the content of the Food table?

C: 30, “PAD THAI”

A: 31, “Hawaiian Pizza”

Or

A: 31, “Hawaiian Pizza”

C: 31, “PAD THAI”

|  |  |
| --- | --- |
| food | ingredient |
| Recipe | food(fid:int, fname:varchar(45));  ingredient(iid:int, iname:varchar(45), type: varchar(30));  recipe(amount:varchar(10), fid:int, iid: int, calorie:int)  recipe.fid is a foreign key to food.fid  recipe.iid is a foreign key to Ingredient.iid  Primary key attributes are underlined. |

**Fig. 1. Current instances of the three tables**