ComS 363 Fall 2022
Class Participation for Week 9

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

Terms of use: This document is not to be shared with anyone else outside of ComS 363 Fall 2022 without the instructor's written approval.

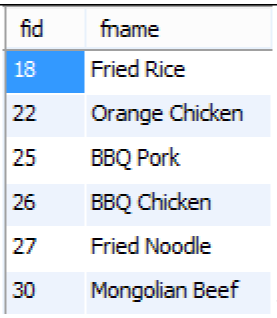
```
try {

    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;
    String sqlstr="select fid, fname from food where fid = (select max(fid)from food)";

    // 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(sqlstr);

    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:

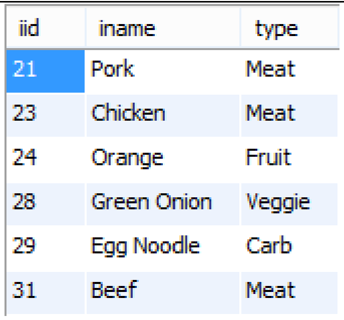
1. What does Program 1 do?
2. What does Program 2 do?
3. Suppose there are three users: A, B, and C, running the two programs simultaneously. Users A and B run Program 1. User A adds the food name "Hawaiian Pizza" while user B adds the food name "Hummus". User C runs Program 2 with the food name of "Pad 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?
4. 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?

Terms of use: This document is not to be shared with anyone else outside of ComS 363 Fall 2022 without the instructor's written approval.



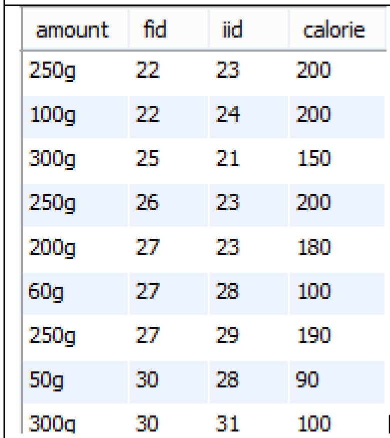
fid	fname
18	Fried Rice
22	Orange Chicken
25	BBQ Pork
26	BBQ Chicken
27	Fried Noodle
30	Mongolian Beef

food



iid	iname	type
21	Pork	Meat
23	Chicken	Meat
24	Orange	Fruit
28	Green Onion	Veggie
29	Egg Noodle	Carb
31	Beef	Meat

ingredient



amount	fid	iid	calorie
250g	22	23	200
100g	22	24	200
300g	25	21	150
250g	26	23	200
200g	27	23	180
60g	27	28	100
250g	27	29	190
50g	30	28	90
300g	30	31	100

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