

Homework #5  
CSC 452  
Zehong Zhuang

**AgeTable (AgeNum, AgeFact)**

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Building new AgeTable table...

Inserting rows in Age table...

Display Age table...

1: Under 18

18: 1-24

25: 25-34

35: 35-44

45: 45-49

50: 50-55

56: 56+

### GenreTable (GenreID, Genre)

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Building new GenreTable table...

Inserting rows Genre table...

Display Genre table...

- 1: Action
- 2: Adventure
- 3: Animation
- 4: Childrens
- 5: Comedy
- 6: Crime
- 7: Documentary
- 8: Drama
- 9: Fantasy
- 10: Film-Noir
- 11: Horror
- 12: Musical
- 13: Mystery
- 14: Romance
- 15: Sci-Fi
- 16: Thriller
- 17: War
- 18: Western

**OccupationTable (OccupationID, Occupation)**

---

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Building new OccupationTable table...

Inserting rows OccupationTable table...

Display Occupation table...

0: other  
1: academic/educator  
2: artist  
3: clerical/admin  
4: college/grad student  
5: customer service  
6: doctor/health care  
7: executive/managerial  
8: farmer  
9: homemaker  
10: K-12 Student  
11: lawyer  
12: programmer  
13: retired  
14: sales/marketing  
15: scientist  
16: self-employed  
17: technician/engineer  
18: tradesman/craftsman  
19: unemployed  
20: writer

# RatingsTable (UserID, MovieID, Ratings, Timestamp)

Loading JDBC driver...

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Building new Ratings table...

Inserting rows in Ratings table...

|         |   |          |      |          |   |            |                         |
|---------|---|----------|------|----------|---|------------|-------------------------|
| UserID: | 1 | MovieID: | 1193 | Ratings: | 5 | Timestamp: | 1970-01-12 01:45:00.76  |
| UserID: | 1 | MovieID: | 661  | Ratings: | 3 | Timestamp: | 1970-01-12 01:45:02.109 |
| UserID: | 1 | MovieID: | 914  | Ratings: | 3 | Timestamp: | 1970-01-12 01:45:01.968 |
| UserID: | 1 | MovieID: | 3408 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:00.275 |
| UserID: | 1 | MovieID: | 2355 | Ratings: | 5 | Timestamp: | 1970-01-12 01:53:44.291 |
| UserID: | 1 | MovieID: | 1197 | Ratings: | 3 | Timestamp: | 1970-01-12 01:45:02.268 |
| UserID: | 1 | MovieID: | 1287 | Ratings: | 5 | Timestamp: | 1970-01-12 01:45:02.039 |
| UserID: | 1 | MovieID: | 2804 | Ratings: | 5 | Timestamp: | 1970-01-12 01:45:00.719 |
| UserID: | 1 | MovieID: | 594  | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:02.268 |
| UserID: | 1 | MovieID: | 919  | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:01.368 |
| UserID: | 1 | MovieID: | 595  | Ratings: | 5 | Timestamp: | 1970-01-12 01:53:44.268 |
| UserID: | 1 | MovieID: | 938  | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:01.752 |
| UserID: | 1 | MovieID: | 2398 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:02.281 |
| UserID: | 1 | MovieID: | 2918 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:02.124 |
| UserID: | 1 | MovieID: | 1035 | Ratings: | 5 | Timestamp: | 1970-01-12 01:45:01.753 |
| UserID: | 1 | MovieID: | 2791 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:02.188 |
| UserID: | 1 | MovieID: | 2687 | Ratings: | 3 | Timestamp: | 1970-01-12 01:53:44.268 |
| UserID: | 1 | MovieID: | 2018 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:01.777 |
| UserID: | 1 | MovieID: | 3105 | Ratings: | 5 | Timestamp: | 1970-01-12 01:45:01.713 |
| UserID: | 1 | MovieID: | 2797 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:02.039 |
| UserID: | 1 | MovieID: | 2321 | Ratings: | 3 | Timestamp: | 1970-01-12 01:45:02.205 |
| UserID: | 1 | MovieID: | 720  | Ratings: | 3 | Timestamp: | 1970-01-12 01:45:00.76  |
| UserID: | 1 | MovieID: | 1270 | Ratings: | 5 | Timestamp: | 1970-01-12 01:45:00.055 |
| UserID: | 1 | MovieID: | 527  | Ratings: | 5 | Timestamp: | 1970-01-12 01:53:44.195 |
| UserID: | 1 | MovieID: | 2340 | Ratings: | 3 | Timestamp: | 1970-01-12 01:45:00.103 |
| UserID: | 1 | MovieID: | 48   | Ratings: | 5 | Timestamp: | 1970-01-12 01:53:44.351 |
| UserID: | 1 | MovieID: | 1097 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:01.953 |
| UserID: | 1 | MovieID: | 1721 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:00.055 |
| UserID: | 1 | MovieID: | 1545 | Ratings: | 4 | Timestamp: | 1970-01-12 01:53:44.139 |
| UserID: | 1 | MovieID: | 745  | Ratings: | 3 | Timestamp: | 1970-01-12 01:53:44.268 |
| UserID: | 1 | MovieID: | 2294 | Ratings: | 4 | Timestamp: | 1970-01-12 01:53:44.291 |
| UserID: | 1 | MovieID: | 3186 | Ratings: | 4 | Timestamp: | 1970-01-12 01:45:00.019 |
| UserID: | 1 | MovieID: | 1566 | Ratings: | 4 | Timestamp: | 1970-01-12 01:53:44.33  |
| UserID: | 1 | MovieID: | 588  | Ratings: | 4 | Timestamp: | 1970-01-12 01:53:44.268 |
| UserID: | 1 | MovieID: | 1907 | Ratings: | 4 | Timestamp: | 1970-01-12 01:53:44.33  |

UsersTable (UserID, Gender, AgeCode, Occupation, Zipcode)

Loading JDBC driver...

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Building new Users table...

Inserting rows in Users table...

|         |    |          |            |               |               |
|---------|----|----------|------------|---------------|---------------|
| UserID: | 1  | Gender:F | AgeCode:1  | Occupation:10 | Zipcode:48067 |
| UserID: | 2  | Gender:M | AgeCode:56 | Occupation:16 | Zipcode:70072 |
| UserID: | 3  | Gender:M | AgeCode:25 | Occupation:15 | Zipcode:55117 |
| UserID: | 4  | Gender:M | AgeCode:45 | Occupation:7  | Zipcode:02460 |
| UserID: | 5  | Gender:M | AgeCode:25 | Occupation:20 | Zipcode:55455 |
| UserID: | 6  | Gender:F | AgeCode:50 | Occupation:9  | Zipcode:55117 |
| UserID: | 7  | Gender:M | AgeCode:35 | Occupation:1  | Zipcode:06810 |
| UserID: | 8  | Gender:M | AgeCode:25 | Occupation:12 | Zipcode:11413 |
| UserID: | 9  | Gender:M | AgeCode:25 | Occupation:17 | Zipcode:61614 |
| UserID: | 10 | Gender:F | AgeCode:35 | Occupation:1  | Zipcode:95370 |
| UserID: | 11 | Gender:F | AgeCode:25 | Occupation:1  | Zipcode:04093 |
| UserID: | 12 | Gender:M | AgeCode:25 | Occupation:12 | Zipcode:32793 |
| UserID: | 13 | Gender:M | AgeCode:45 | Occupation:1  | Zipcode:93304 |
| UserID: | 14 | Gender:M | AgeCode:35 | Occupation:0  | Zipcode:60126 |
| UserID: | 15 | Gender:M | AgeCode:25 | Occupation:7  | Zipcode:22903 |
| UserID: | 16 | Gender:F | AgeCode:35 | Occupation:0  | Zipcode:20670 |
| UserID: | 17 | Gender:M | AgeCode:50 | Occupation:1  | Zipcode:95350 |
| UserID: | 18 | Gender:F | AgeCode:18 | Occupation:3  | Zipcode:95825 |
| UserID: | 19 | Gender:M | AgeCode:1  | Occupation:10 | Zipcode:48073 |
| UserID: | 20 | Gender:M | AgeCode:25 | Occupation:14 | Zipcode:55113 |
| UserID: | 21 | Gender:M | AgeCode:18 | Occupation:16 | Zipcode:99353 |
| UserID: | 22 | Gender:M | AgeCode:18 | Occupation:15 | Zipcode:53706 |
| UserID: | 23 | Gender:M | AgeCode:35 | Occupation:0  | Zipcode:90049 |
| UserID: | 24 | Gender:F | AgeCode:25 | Occupation:7  | Zipcode:10023 |
| UserID: | 25 | Gender:M | AgeCode:18 | Occupation:4  | Zipcode:01609 |
| UserID: | 26 | Gender:M | AgeCode:25 | Occupation:7  | Zipcode:23112 |
| UserID: | 27 | Gender:M | AgeCode:25 | Occupation:11 | Zipcode:19130 |
| UserID: | 28 | Gender:F | AgeCode:25 | Occupation:1  | Zipcode:14607 |
| UserID: | 29 | Gender:M | AgeCode:35 | Occupation:7  | Zipcode:33407 |
| UserID: | 30 | Gender:F | AgeCode:35 | Occupation:7  | Zipcode:19143 |
| UserID: | 31 | Gender:M | AgeCode:56 | Occupation:7  | Zipcode:06840 |
| UserID: | 32 | Gender:F | AgeCode:25 | Occupation:0  | Zipcode:19355 |
| UserID: | 33 | Gender:M | AgeCode:45 | Occupation:3  | Zipcode:55421 |
| UserID: | 34 | Gender:F | AgeCode:18 | Occupation:0  | Zipcode:02135 |
| UserID: | 35 | Gender:M | AgeCode:45 | Occupation:1  | Zipcode:02482 |
| UserID: | 36 | Gender:M | AgeCode:25 | Occupation:3  | Zipcode:94123 |
| UserID: | 37 | Gender:F | AgeCode:25 | Occupation:9  | Zipcode:66212 |
| UserID: | 38 | Gender:F | AgeCode:18 | Occupation:4  | Zipcode:02215 |
| UserID: | 39 | Gender:M | AgeCode:18 | Occupation:4  | Zipcode:61820 |

## MoviesTable (MovieID, Title, Years)

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Building new MoviesTable table...

Inserting rows in Movies table...

Display Movies table...

|              |   |               |
|--------------|---|---------------|
| MovieID: 181 | Title: Mighty Morphin Power Rangers: The Movie                    | Years: (1995) |
| MovieID: 190 | Title: Safe   | Years: (1995) |
| MovieID: 199 | Title: Umbrellas of Cherbourg, The (Parapluies de Cherbourg, Les) | Years: (1964) |
| MovieID: 8   | Title: Tom and Huck   | Years: (1995) |
| MovieID: 22  | Title: Copycat  | Years: (1995) |
| MovieID: 39  | Title: Clueless   | Years: (1995) |
| MovieID: 65  | Title: Bio-Dome   | Years: (1996) |
| MovieID: 71  | Title: Fair Game  | Years: (1995) |
| MovieID: 74  | Title: Bed of Roses   | Years: (1996) |
| MovieID: 75  | Title: Big Bully  | Years: (1996) |
| MovieID: 89  | Title: Nick of Time   | Years: (1995) |
| MovieID: 101 | Title: Bottle Rocket  | Years: (1996) |
| MovieID: 109 | Title: Headless Body in Topless Bar                               | Years: (1995) |
| MovieID: 113 | Title: Before and After   | Years: (1996) |
| MovieID: 135 | Title: Down Periscope   | Years: (1996) |
| MovieID: 143 | Title: Gospa  | Years: (1995) |
| MovieID: 144 | Title: Brothers McMullen, The                                     | Years: (1995) |
| MovieID: 149 | Title: Amateur  | Years: (1994) |
| MovieID: 158 | Title: Casper   | Years: (1995) |
| MovieID: 174 | Title: Jury Duty  | Years: (1995) |
| MovieID: 183 | Title: Mute Witness   | Years: (1994) |
| MovieID: 186 | Title: Nine Months  | Years: (1995) |
| MovieID: 11  | Title: American President, The                                    | Years: (1995) |
| MovieID: 16  | Title: Casino   | Years: (1995) |
| MovieID: 18  | Title: Four Rooms   | Years: (1995) |
| MovieID: 19  | Title: Ace Ventura: When Nature Calls                             | Years: (1995) |
| MovieID: 20  | Title: Money Train  | Years: (1995) |
| MovieID: 26  | Title: Othello  | Years: (1995) |
| MovieID: 34  | Title: Babe   | Years: (1995) |
| MovieID: 36  | Title: Dead Man Walking   | Years: (1995) |
| MovieID: 41  | Title: Richard III  | Years: (1995) |
| MovieID: 48  | Title: Pocahontas   | Years: (1995) |
| MovieID: 50  | Title: Usual Suspects, The  | Years: (1995) |
| MovieID: 62  | Title: Mr. Holland's Opus   | Years: (1995) |
| MovieID: 66  | Title: Lawnmower Man 2: Beyond Cyberspace                         | Years: (1996) |
| MovieID: 69  | Title: Friday   | Years: (1995) |
| MovieID: 72  | Title: Kicking and Screaming                                      | Years: (1995) |
| MovieID: 78  | Title: Crossing Guard, The  | Years: (1995) |
| MovieID: 84  | Title: Last Summer in the Hamptons                                | Years: (1995) |
| MovieID: 87  | Title: Dunston Checks In  | Years: (1996) |

### **MovieGenre (MovieID, GenreID)**

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Building new MoviesGenreTable table...

Inserting rows in Genre table...

Display Movies table...

|          |     |          |      |
|----------|-----|----------|------|
| MovieID: | 20  | GenreID: | 1    |
| MovieID: | 21  | GenreID: | 1    |
| MovieID: | 42  | GenreID: | 1    |
| MovieID: | 139 | GenreID: | 1    |
| MovieID: | 2   | GenreID: | 2    |
| MovieID: | 60  | GenreID: | 2    |
| MovieID: | 13  | GenreID: | 3    |
| MovieID: | 19  | GenreID: | 5    |
| MovieID: | 104 | GenreID: | 5    |
| MovieID: | 174 | GenreID: | 5    |
| MovieID: | 195 | GenreID: | 5    |
| MovieID: | 39  | GenreID: | 5    |
| MovieID: | 129 | GenreID: | 5    |
| MovieID: | 77  | GenreID: | 7    |
| MovieID: | 162 | GenreID: | 7    |
| MovieID: | 53  | GenreID: | 8    |
| MovieID: | 78  | GenreID: | 8    |
| MovieID: | 159 | GenreID: | 8    |
| MovieID: | 167 | GenreID: | 8    |
| MovieID: | 175 | GenreID: | 8    |
| MovieID: | 17  | GenreID: | 8    |
| MovieID: | 140 | GenreID: | 8    |
| MovieID: | 66  | GenreID: | 15   |
| MovieID: | 103 | GenreID: | 16   |
| MovieID: | 200 | GenreID: | 16   |
| MovieID: | 112 | GenreID: | 1    |
| MovieID: | 15  | GenreID: | 1    |
| MovieID: | 107 | GenreID: | 2    |
| MovieID: | 86  | GenreID: | 2    |
| MovieID: | 54  | GenreID: | null |
| MovieID: | 5   | GenreID: | 5    |
| MovieID: | 119 | GenreID: | 5    |
| MovieID: | 125 | GenreID: | 5    |
| MovieID: | 171 | GenreID: | 5    |
| MovieID: | 176 | GenreID: | 5    |
| MovieID: | 178 | GenreID: | 5    |
| MovieID: | 180 | GenreID: | 5    |
| MovieID: | 45  | GenreID: | 5    |
| MovieID: | 84  | GenreID: | 5    |
| MovieID: | 157 | GenreID: | 5    |

### Query Question

Connecting to ACADPRD0 database...

Connected to database ACADPRD0...

Display query question...

Number: 4331 Gender: M

Number: 1709 Gender: F

The majority of the users are males. Based on the result, numbers of males are way much more than females, which cause unbalancing of datasets. There is a risk that any analysis based on this dataset is going to be biased. Analyst should pay attentions when performing analysis. Also, for marketing department, it is important to collect information from both parties.



## AgeTable Codes

```
import java.sql.*;
import java.io.*;

public class AgeTable {

    public static void main( String[] args ) {

        String ageTable = "AgeTable";

        Connection conn = null;
        Statement stmt = null;

        /*****
        * determine if the JDBC driver exists and load it...
        *****/

        /*****
        * establish a connection to the database...
        *****/
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn =
DriverManager.getConnection("jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:AC
ADPRD0", "ZZHUANG", "cdm1379457");

            /*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
            : DriverManager.getConnection(url, userName, password );*/
            System.out.println( "Connected to database ACADPRD0..." );

            /*****
            * create an object by which we will pass SQL stmts to the database...
            *****/
            stmt = conn.createStatement();
        }
        catch (SQLException se) {
            System.out.println(se);
            System.exit(1);
        }
        try {
```

```

String dropString = "DROP TABLE " + ageTable;
stmt.executeUpdate(dropString);
}
catch (SQLException se) { /*do nothing*/ } // table doesn't exist

try {
    /*****
    * finally, display all the rows in the database...
    *****/

    System.out.print( "Building new " + ageTable + " table...\n\n" );
    String createString =
        "CREATE TABLE " + ageTable + "(AgeNum Number PRIMARY KEY, "
        + "AgeFact VARCHAR2(30))";
    stmt.executeUpdate(createString);

    System.out.print( "Inserting rows in Age table...\n\n" );
    String insertString =
        "INSERT INTO " + ageTable + " VALUES (01,'Under 18)";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + ageTable + " VALUES (18,'1-24)";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + ageTable + " VALUES (25,'25-34)";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + ageTable + " VALUES (35,'35-44)";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + ageTable + " VALUES (45,'45-49)";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + ageTable + " VALUES (50,'50-55)";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + ageTable + " VALUES (56,'56+)";
    stmt.executeUpdate(insertString);

```

```
        System.out.print( "Display Age table...\n\n");
        ResultSet rset = stmt.executeQuery( " SELECT * FROM " + ageTable );
        while( rset.next() )
            System.out.println( rset.getString("AgeNum") + ": " +
                                rset.getString("AgeFact"));

        rset.close();
        stmt.close();
        conn.close();
    }
    catch (SQLException se) {
        System.out.println( "SQL ERROR: " + se );
    }
} // end main

} // end class
```

## GenreTable Code

```
import java.sql.*;
import java.io.*;

public class GenreTable {

    public static void main( String[] args ) {

        String genreTable = "GenreTable";

        Connection conn = null;
        Statement stmt = null;

        /*****
        * determine if the JDBC driver exists and load it...
        *****/

        /*****
        * establish a connection to the database...
        *****/
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn =
DriverManager.getConnection("jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:AC
ADPRD0", "ZZHUANG", "cdm1379457");

            /*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
            : DriverManager.getConnection(url, userName, password );*/
            System.out.println( "Connected to database ACADPRD0..." );

            /*****
            * create an object by which we will pass SQL stmts to the database...
            *****/
            stmt = conn.createStatement();
        }
        catch (SQLException se) {
            System.out.println(se);
            System.exit(1);
        }
        try {
            String dropString = "DROP TABLE " + genreTable;
```

```

    stmt.executeUpdate(dropString);
}
catch (SQLException se) { /*do nothing*/ } // table doesn't exist

try {
    /*****
    * finally, display all the rows in the database...
    *****/
    System.out.print( "Building new " + genreTable + " table...\n\n" );
    String createString =
        "CREATE TABLE " + genreTable + "(GenreID Number PRIMARY KEY, "
        + "Genre VARCHAR2(30))";
    stmt.executeUpdate(createString);

    System.out.print( "Inserting rows Genre table...\n\n" );
    String insertString =
        "INSERT INTO " + genreTable + " VALUES (1,'Action')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + genreTable + " VALUES (2,'Adventure')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + genreTable + " VALUES (3,'Animation')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + genreTable + " VALUES (4,'Childrens')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + genreTable + " VALUES (5,'Comedy')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + genreTable + " VALUES (6,'Crime')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + genreTable + " VALUES (7,'Documentary')";
    stmt.executeUpdate(insertString);

    insertString =

```

```
        "INSERT INTO " + genreTable + " VALUES (8,'Drama')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (9,'Fantasy')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (10,'Film-Noir')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (11,'Horror')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (12,'Musical')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (13,'Mystery')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (14,'Romance')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (15,'Sci-Fi')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (16,'Thriller')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (17,'War')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + genreTable + " VALUES (18,'Western')";
stmt.executeUpdate(insertString);

System.out.print( "Display Genre table...\n\n");
```

```
ResultSet rset = stmt.executeQuery( " SELECT * FROM " + genreTable );
while( rset.next() )
    System.out.println( rset.getString("GenreID") + ": " +
        rset.getString("Genre"));

    rset.close();
    stmt.close();
    conn.close();
}
catch (SQLException se) {
    System.out.println( "SQL ERROR: " + se );
}

} // end main

} // end class
```

## OccupationTable Code

```
import java.sql.*;
import java.io.*;

public class OccupationTable {

    public static void main( String[] args ) {

        String occupationTable = "OccupationTable";

        Connection conn = null;
        Statement stmt = null;

        /*****
        * determine if the JDBC driver exists and load it...
        *****/

        /*****
        * establish a connection to the database...
        *****/
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn =
DriverManager.getConnection("jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:AC
ADPRD0", "ZZHUANG", "cdm1379457");

            /*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
            : DriverManager.getConnection(url, userName, password );*/
            System.out.println( "Connected to database ACADPRD0..." );

            /*****
            * create an object by which we will pass SQL stmts to the database...
            *****/
            stmt = conn.createStatement();
        }
        catch (SQLException se) {
            System.out.println(se);
            System.exit(1);
        }
        try {
            String dropString = "DROP TABLE " + occupationTable;
```



```

    stmt.executeUpdate(dropString);
}
catch (SQLException se) { /*do nothing*/ } // table doesn't exist

try {
    /*****
    * finally, display all the rows in the database...
    *****/
    System.out.print( "Building new " + occupationTable + " table...\n\n" );
    String createString =
        "CREATE TABLE " + occupationTable + "(OccupationID Number
PRIMARY KEY, "
        + "JobDescription VARCHAR2(30))";
    stmt.executeUpdate(createString);

    System.out.print( "Inserting rows OccupationTable table...\n\n" );
    String insertString =
        "INSERT INTO " + occupationTable + " VALUES (0,'other')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + occupationTable + " VALUES (1,'academic/educator')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + occupationTable + " VALUES (2,'artist')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + occupationTable + " VALUES (3,'clerical/admin')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + occupationTable + " VALUES (4,'college/grad student')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + occupationTable + " VALUES (5,'customer service')";
    stmt.executeUpdate(insertString);

    insertString =
        "INSERT INTO " + occupationTable + " VALUES (6,'doctor/health care')";
    stmt.executeUpdate(insertString);

```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (7,'executive/managerial')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (8,'farmer')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (9,'homemaker')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (10,'K-12 Student')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (11,'lawyer')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (12,'programmer')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (13,'retired')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (14,'sales/marketing')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (15,'scientist')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (16,'self-employed')";  
stmt.executeUpdate(insertString);
```

```
insertString =  
    "INSERT INTO " + occupationTable + " VALUES (17,'technician/engineer')";  
stmt.executeUpdate(insertString);
```

```

insertString =
    "INSERT INTO " + occupationTable + " VALUES (18,'tradesman/craftsman')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + occupationTable + " VALUES (19,'unemployed')";
stmt.executeUpdate(insertString);

insertString =
    "INSERT INTO " + occupationTable + " VALUES (20,'writer')";
stmt.executeUpdate(insertString);

    System.out.print( "Display Occupation table...\n\n");
    ResultSet rset = stmt.executeQuery( " SELECT * FROM " + occupationTable );
    while( rset.next() )
        System.out.println( rset.getString("OccupationID") + ": " +
            rset.getString("JobDescription"));

    rset.close();
    stmt.close();
    conn.close();
}
catch (SQLException se) {
    System.out.println( "SQL ERROR: " + se );
}

} // end main

} // end class

```

## RatingsTable Code

```
import java.util.List;
import java.sql.*;
import java.util.Arrays;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.Connection;
import java.io.FileNotFoundException;

public class RatingsTable {

    public static void main( String[] args ) throws
    FileNotFoundException, NumberFormatException, IOException {

        String tableName = "Ratings";

        Connection conn = null;
        Statement stmt = null;
        /*****
        * determine if the JDBC driver exists and load it...
        *****/
        System.out.print( "\nLoading JDBC driver...\n\n" );
        try {
            Class.forName("oracle.jdbc.OracleDriver");
        }
        catch(ClassNotFoundException e) {
            System.out.println(e);
            System.exit(1);
        }

        /*****
        * establish a connection to the database...
        *****/
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn = DriverManager.getConnection(
                "jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:ACADPRD0",
```

```

        "ZZHUANG", "cdm1379457");

/*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
 : DriverManager.getConnection(url, userName, password );*/
System.out.println( "Connected to database ACADPRD0..." );

/*****
 * create an object by which we will pass SQL stmts to the database...
 *****/
stmt = conn.createStatement();
}
catch (SQLException se) {
    System.out.println(se);
    System.exit(1);
}

/*****
 * in the event that this table already exists, we want to delete it
 * and build a new table from scratch... if the table doesn't exist,
 * an SQLException would be thrown when the DROP TABLE stmt below is
 * executed. We catch that exception, but we don't need to do anything
 * because we expect the error to occur if the table doesn't exist...
 *****/
try {
    String dropString = "DROP TABLE " + tableName;
    stmt.executeUpdate(dropString);
}
catch (SQLException se) { /*do nothing*/ } // table doesn't exist

try {
/*****
 * create the new table...
 *****/
System.out.print( "Building new " + tableName + " table...\n\n" );
String createString =
    "CREATE TABLE " + tableName +
    " (UserID INT NOT NULL,"
    + "MovieID INT NOT NULL,"
    + "Ratings INT NOT NULL,"
    + "TimeStamp TIMESTAMP NOT NULL,"
    + "PRIMARY KEY (UserID, MovieID))";
stmt.executeUpdate(createString);

```

```

/*****
* now populate the table...
*****/
System.out.print( "Inserting rows in Ratings table...\n\n" );

    PreparedStatement updateRating =
    conn.prepareStatement( "INSERT INTO " + tableName + " VALUES (?, ?, ?, ? )" );

    conn.setAutoCommit(false);

    File file = new File("/Users/zehongzhuang/Desktop/ratings.dat");
    FileInputStream fstream = new FileInputStream(file);
    BufferedReader br = new BufferedReader (new InputStreamReader(fstream));

    String content;

    int i = 0;

    while ((content = br.readLine()) !=null && i<=200000){
        List<String> ratingData = Arrays.asList(content.split("::"));
        updateRating.setInt(1, Integer.parseInt(ratingData.get(0)));
        updateRating.setInt(2, Integer.parseInt(ratingData.get(1)));
        updateRating.setInt(3, Integer.parseInt(ratingData.get(2)));
        updateRating.setTimestamp(4, new
Timestamp(Long.parseLong(ratingData.get(3))));
        updateRating.executeUpdate();
        i++;
    }
    conn.commit();

/*****
* finally, display all the rows in the database...
*****/
    ResultSet rset = stmt.executeQuery( "SELECT * FROM " + tableName);
    while( rset.next() )
        System.out.println("UserID: "+ rset.getString("UserID") + " MovieID: " +
            rset.getString("MovieID")+" Ratings:"+rset.getString("Ratings")+"
Timestamp:"+rset.getString("TimeStamp"));
    br.close();
    rset.close();
    stmt.close();
    conn.close();
}
catch (SQLException se) {

```

```
        System.out.println( "SQL ERROR: " + se );  
    }
```

```
} // end main
```

```
} // end class
```

## UsersTable Code

```
import java.util.List;
import java.sql.*;
import java.util.Arrays;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.Connection;
import java.io.FileNotFoundException;

public class UsersTable {

    public static void main( String[] args ) throws
    FileNotFoundException, NumberFormatException, IOException {

        String tableName = "Users";

        Connection conn = null;
        Statement stmt = null;
        /*****
        * determine if the JDBC driver exists and load it...
        *****/
        System.out.print( "\nLoading JDBC driver...\n\n" );
        try {
            Class.forName("oracle.jdbc.OracleDriver");
        }
        catch(ClassNotFoundException e) {
            System.out.println(e);
            System.exit(1);
        }

        /*****
        * establish a connection to the database...
        *****/
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn = DriverManager.getConnection(
                "jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:ACADPRD0",
```



```

        "ZZHUANG", "cdm1379457");

/*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
 : DriverManager.getConnection(url, userName, password );*/
System.out.println( "Connected to database ACADPRD0..." );

/*****
 * create an object by which we will pass SQL stmts to the database...
 *****/
stmt = conn.createStatement();
}
catch (SQLException se) {
    System.out.println(se);
    System.exit(1);
}

/*****
 * in the event that this table already exists, we want to delete it
 * and build a new table from scratch... if the table doesn't exist,
 * an SQLException would be thrown when the DROP TABLE stmt below is
 * executed. We catch that exception, but we don't need to do anything
 * because we expect the error to occur if the table doesn't exist...
 *****/
try {
    String dropString = "DROP TABLE " + tableName;
    stmt.executeUpdate(dropString);
}
catch (SQLException se) { /*do nothing*/ } // table doesn't exist

try {
/*****
 * create the new table...
 *****/
System.out.print( "Building new " + tableName + " table...\n\n" );
String createString =
    "CREATE TABLE " + tableName +
    " (UserID INT NOT NULL,"
    + "Gender VARCHAR2(1),"
    + "AgeCode INT,"
    + "Occupation VARCHAR(20),"
    + "Zipcode VARCHAR2(100))";
stmt.executeUpdate(createString);

```

```

/*****
* now populate the table...
*****/
System.out.print( "Inserting rows in Users table...\n\n" );

        PreparedStatement updateUsers =
        conn.prepareStatement( "INSERT INTO " + tableName + " VALUES
(?, ?, ?, ?, ?)");

        conn.setAutoCommit(false);

        File file = new File("/Users/zehongzhuang/Desktop/users.dat");
        FileInputStream fstream = new FileInputStream(file);
        BufferedReader br = new BufferedReader (new InputStreamReader(fstream));

        String content;

        int i = 0;

        while ((content = br.readLine()) != null && i <= 200000){
            List<String> usersData = Arrays.asList(content.split(":"));
            updateUsers.setInt(1, Integer.parseInt(usersData.get(0)));
            updateUsers.setString(2, usersData.get(1));
            updateUsers.setInt(3, Integer.parseInt(usersData.get(2)));
            updateUsers.setString(4, usersData.get(3));
            updateUsers.setString(5, usersData.get(4));
            updateUsers.executeUpdate();
            i++;
        }
        conn.commit();

/*****
* finally, display all the rows in the database...
*****/
        ResultSet rset = stmt.executeQuery( "SELECT * FROM " + tableName);
        while( rset.next() )
            System.out.println("UserID: "+ rset.getString("UserID") + " Gender:" +
            rset.getString("Gender")+" AgeCode:"+rset.getString("AgeCode")+"
Occupation:"+rset.getString("Occupation")+" Zipcode:"+rset.getString("Zipcode"));
        br.close();
        rset.close();

```

```
        stmt.close();
        conn.close();
    }
    catch (SQLException se) {
        System.out.println( "SQL ERROR: " + se );
    }

} // end main

} // end class
```

## MoviesTable Code

(Upload the movies.dat first)

```
import java.util.List;
import java.sql.*;
import java.util.Arrays;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.Connection;
import java.io.FileNotFoundException;

public class Movies {

    public static void main( String[] args ) throws
    FileNotFoundException, NumberFormatException, IOException {

        String tableName = "Movies";

        Connection conn = null;
        Statement stmt = null;
        /*****
        * determine if the JDBC driver exists and load it...
        *****/
        System.out.print( "\nLoading JDBC driver...\n\n" );
        try {
            Class.forName("oracle.jdbc.OracleDriver");
        }
        catch(ClassNotFoundException e) {
            System.out.println(e);
            System.exit(1);
        }

        /*****
        * establish a connection to the database...
        *****/
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn = DriverManager.getConnection(
```

```
"jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:ACADPRD0",  
"ZZHUANG", "cdm1379457");
```

```
/*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)  
: DriverManager.getConnection(url, userName, password );*/  
System.out.println( "Connected to database ACADPRD0..." );
```

```
/*  
* create an object by which we will pass SQL stmts to the database...  
*/
```

```
stmt = conn.createStatement();  
}
```

```
catch (SQLException se) {  
    System.out.println(se);  
    System.exit(1);  
}
```

```
/*  
* in the event that this table already exists, we want to delete it  
* and build a new table from scratch... if the table doesn't exist,  
* an SQLException would be thrown when the DROP TABLE stmt below is  
* executed. We catch that exception, but we don't need to do anything  
* because we expect the error to occur if the table doesn't exist...  
*/
```

```
try {  
    String dropString = "DROP TABLE " + tableName;  
    stmt.executeUpdate(dropString);  
}  
catch (SQLException se) { /*do nothing*/ } // table doesn't exist
```

```
try {  
/*  
* create the new table...  
*/
```

```
System.out.print( "Building new " + tableName + " table...\n\n" );  
String createString =  
    "CREATE TABLE " + tableName +  
    " (MovieID INT NOT NULL,"  
    + "Title VARCHAR2(500),"  
    + "Genres VARCHAR2 (500))";  
stmt.executeUpdate(createString);
```

```

/*****
* now populate the table...
*****/
System.out.print( "Inserting rows in Movies table...\n\n" );

    PreparedStatement updateMovies =
    conn.prepareStatement( "INSERT INTO " + tableName + " VALUES (?, ?, ? )" );

    conn.setAutoCommit(false);

    File file = new File("/Users/zehongzhuang/Desktop/movies.dat");
    FileInputStream fstream = new FileInputStream(file);
    BufferedReader br = new BufferedReader (new InputStreamReader(fstream));

    String content;

    int i = 0;

    while ((content = br.readLine()) !=null && i<=200000){
        List<String> moviesData = Arrays.asList(content.split("::"));
        updateMovies.setInt(1, Integer.parseInt(moviesData.get(0)));
        updateMovies.setString(2, moviesData.get(1));
        updateMovies.setString(3, moviesData.get(2));
        updateMovies.executeUpdate();
        i++;
    }
    conn.commit();

/*****
* finally, display all the rows in the database...
*****/
ResultSet rset = stmt.executeQuery( "SELECT * FROM " + tableName);
while( rset.next() )
    System.out.println("MovieID: "+ rset.getString("MovieID") + " Title:" +
        rset.getString("Title")+" Genres:"+rset.getString("Genres"));
br.close();
rset.close();
stmt.close();
conn.close();
}
catch (SQLException se) {
    System.out.println( "SQL ERROR: " + se );
}

```

```
}
```

```
} // end main
```

```
} // end class
```

**(Create the MoviesTable)**

```
import java.sql.*;
```

```
import java.io.*;
```

```
public class MoviesTable {
```

```
    public static void main( String[] args ) {
```

```
        String moviesTable = "MoviesTable";
```

```
        Connection conn = null;
```

```
        Statement stmt = null;
```

```
        /*****
```

```
        * determine if the JDBC driver exists and load it...
```

```
        *****/
```

```
        /*****
```

```
        * establish a connection to the database...
```

```
        *****/
```

```
        try {
```

```
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
```

```
            //String url = dataSource + dbName;
```

```
            conn =
```

```
DriverManager.getConnection("jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:ACADPRD0", "ZZHUANG", "cdm1379457");
```

```
        /*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
```

```
        : DriverManager.getConnection(url, userName, password );*/
```

```
        System.out.println( "Connected to database ACADPRD0..." );
```

```
        /*****
```

```
        * create an object by which we will pass SQL stmts to the database...
```

```
        *****/
```

```
        stmt = conn.createStatement();
```

```

    }
    catch (SQLException se) {
        System.out.println(se);
        System.exit(1);
    }
    try {
        String dropString = "DROP TABLE " + moviesTable;
        stmt.executeUpdate(dropString);
    }
    catch (SQLException se) { /*do nothing*/ } // table doesn't exist

    try {
        /*****
        * finally, display all the rows in the database...
        *****/
        System.out.print( "Building new " + moviesTable + " table...\n\n" );
        String createString =
            "CREATE TABLE " + moviesTable + "(MovieID Number PRIMARY KEY, "
            + "Title VARCHAR2(500),"
            + "Years VARCHAR2(10))";
        stmt.executeUpdate(createString);

        System.out.print( "Inserting rows in Movies table...\n\n" );
        String insertString=
            "INSERT INTO MoviesTable (MovieID, Title, Years)"
            + "SELECT DISTINCT MOVIEID, SUBSTR(TITLE,0, LENGTH(TITLE)-6),
SUBSTR(TITLE,-6) "
            + "FROM MOVIES";
        stmt.executeUpdate(insertString);

        System.out.print( "Display Movies table...\n\n");
        ResultSet rset = stmt.executeQuery( " SELECT * FROM " + moviesTable );
        while( rset.next() )
            System.out.println("MovieID: "+ rset.getString("MovieID") + " Title: " +
                rset.getString("Title") + " Years: "+ rset.getString("Years"));

        rset.close();
        stmt.close();
        conn.close();
    }
    catch (SQLException se) {

```



```
        System.out.println( "SQL ERROR: " + se );  
    }
```

```
} // end main
```

```
} // end class
```

## Movies-Genre Table Code

```
import java.sql.*;
import java.io.*;

public class MoviesGenreTable {

    public static void main( String[] args ) {

        String moviesGenreTable = "MoviesGenreTable";

        Connection conn = null;
        Statement stmt = null;

        /**
         * determine if the JDBC driver exists and load it...
         */

        /**
         * establish a connection to the database...
         */
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn =
DriverManager.getConnection("jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:AC
ADPRD0", "ZZHUANG", "cdm1379457");

            /*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
            : DriverManager.getConnection(url, userName, password );*/
            System.out.println( "Connected to database ACADPRD0..." );

            /**
             * create an object by which we will pass SQL stmts to the database...
             */

            stmt = conn.createStatement();
        }
        catch (SQLException se) {
            System.out.println(se);
            System.exit(1);
        }
        try {
            String dropString = "DROP TABLE " + moviesGenreTable;
```

```

        stmt.executeUpdate(dropString);
    }
    catch (SQLException se) { /*do nothing*/ } // table doesn't exist

try {
    /*****
    * finally, display all the rows in the database...
    *****/
    System.out.print( "Building new " + moviesGenreTable + " table...\n\n" );
    String createString =
        "CREATE TABLE " + moviesGenreTable + "(MovieID INT PRIMARY KEY, "
        + "GenreID INT,"
        + "Genre VARCHAR2(100))";
    stmt.executeUpdate(createString);

    System.out.print( "Inserting rows in Genre table...\n\n" );
    String insertString=
        "INSERT INTO "+ moviesGenreTable+ "(MovieID,Genre) SELECT DISTINCT
    MOVIEID, trim(regexp_substr(Genres, '[^|]+' , LEVEL)) FROM MOVIES CONNECT BY
    instr(regexp_substr(Genres, '[^|]+' ,LEVEL),'|', 1, level -1)>0";
    stmt.executeUpdate(insertString);

    String mergeString=
        "MERGE INTO MoviesGenreTable USING GenreTable ON
    (MoviesGenreTable.Genre=GenreTable.Genre) WHEN MATCHED THEN UPDATE SET
    MoviesGenreTable.GenreID=GenreTable.GenreID";
    stmt.executeUpdate(mergeString);

    String dropString=
        "ALTER TABLE moviesGenreTable DROP COLUMN Genre";
    stmt.executeUpdate(dropString);

    System.out.print( "Display Movies table...\n\n");
    ResultSet rset = stmt.executeQuery( " SELECT * FROM " + moviesGenreTable);
    while( rset.next() )
        System.out.println("MovieID: " + rset.getString("MovieID") + " GenreID: " +
            rset.getString("GenreID"));

    rset.close();
    stmt.close();
    conn.close();
}

```

```
    catch (SQLException se) {  
        System.out.println( "SQL ERROR: " + se );  
    }  
  
} // end main  
  
} // end class
```

## Query Question Code

```
import java.sql.*;
import java.io.*;

public class Query {

    public static void main( String[] args ) {

        Connection conn = null;
        Statement stmt = null;

        /*****
        * determine if the JDBC driver exists and load it...
        *****/

        /*****
        * establish a connection to the database...
        *****/
        try {
            System.out.print( "Connecting to ACADPRD0 database...\n\n" );
            //String url = dataSource + dbName;

            conn =
            DriverManager.getConnection("jdbc:oracle:thin:@acadoradbprd01.dpu.depaul.edu:1521:AC
            ADPRD0", "ZZHUANG", "cdm1379457");

            /*conn = dbms.equals("localAccess") ? DriverManager.getConnection(url)
            : DriverManager.getConnection(url, userName, password );*/
            System.out.println( "Connected to database ACADPRD0..." );

            /*****
            * create an object by which we will pass SQL stmts to the database...
            *****/

            stmt = conn.createStatement();
        }
        catch (SQLException se) {
            System.out.println(se);
            System.exit(1);
        }

        try {
            /*****
            * finally, display all the rows in the database...
            *****/
```

```

*****/
    String query="SELECT count(gender),gender FROM UsersTable group by gender";
    System.out.print( "Display query question...\n\n");
    ResultSet rset = stmt.executeQuery(query);
    while( rset.next() )
        System.out.println("Number: "+ rset.getInt(1) + " Gender: " +
            rset.getString(2));

    rset.close();
    stmt.close();
    conn.close();
}
catch (SQLException se) {
    System.out.println( "SQL ERROR: " + se );
}

} // end main

} // end class

```

