Countries Of The World App 2.5 - Java

CS 3310 – Dr. Donna Kaminski

Connecting to MySQL...

OK, the DB Connection is OPENED

SELECT Name, Language, Percentage FROM Country, CountryLanguage WHERE Code = 'USA' AND Code = CountryCode

Fix & Format (FF).

SELECT Name, LifeExpectancy, IndepYear FROM country WHERE continent = 'Europe' ORDER BY lifeexpectancy

Fix & Format (FF).

SELECT Name, LifeExpectancy FROM Country WHERE LifeExpectancy IN (SELECT MIN(LifeExpectancy) FROM country UNION SELECT MAX(LifeExpectancy) FROM Country) OR Code = 'USA'

Fix & Format (FF).

SELECT Name, Percentage FROM Country, CountryLanguage WHERE Language = 'English' AND Percentage > 1.0 AND Code = CountryCode

Fix & Format (FF).

SELECT Population FROM Country WHERE Name = 'Iceland'

Fix & Format (FF).

SELECT COUNT(\*) FROM Country

Fix & Format (FF).

SELECT COUNT(\*) FROM CountryLanguage

Fix & Format (FF).

I CountryLanguage|'USA','C#','F',0.01

SQL: INSERT INTO VALUES (C);

ERROR on INSERT, not done.

I CountryLanguage|'GBR','Java','F',0.01

SQL: INSERT INTO VALUES (C);

ERROR on INSERT, not done.

I CountryLanguage|CountryCode, Language|'DEU','Java'

SQL: INSERT INTO VALUES (C);

ERROR on INSERT, not done.

SELECT \* FROM CountryLanguage WHERE Language = 'C#' OR Language = 'Java'

Fix & Format (FF).

I Country|Code, Name, Population|'KZO','Kalamazoo',85000

SQL: INSERT INTO VALUES (C);

ERROR on INSERT, not done.

I Country|Code, Name, HeadOfState, Continent, Region|'DIS','Disneyland','Mickey Mouse','North America','North America'

SQL: INSERT INTO VALUES (C);

ERROR on INSERT, not done.

SELECT Code, Name, Population, HeadOfState, Continent, Region FROM Country WHERE code = 'KZO' OR code = 'DIS'

Fix & Format (FF).

D Country|Name|'Iceland'

SQL: DELETE FROM WHERE C = o;

ERROR on DELETE, not done.

D Country|Name|'Faroe Islands'

SQL: DELETE FROM WHERE C = o;

ERROR on DELETE, not done.

D Country|Code|'SJM'

SQL: DELETE FROM WHERE C = o;

ERROR on DELETE, not done.

D CountryLanguage|Language|'Swedish'

SQL: DELETE FROM WHERE C = o;

ERROR on DELETE, not done.

D CountryLanguage|Language|'Russian'

SQL: DELETE FROM WHERE C = o;

ERROR on DELETE, not done.

SELECT Name FROM Country WHERE Region = 'Nordic Countries'

Fix & Format (FF).

SELECT Name, Language FROM Country, CountryLanguage WHERE Region = 'Nordic Countries' AND Code = CountryCode

Fix & Format (FF).

SELECT Name, Language FROM Country, CountryLanguage WHERE Region = 'Nordic Countries' AND Percentage > 99.5

Fix & Format (FF).

UPDATE Country SET HeadOfState = 'Barack Obama' WHERE Name = 'United States'

SQL: UPDATE Country SET HeadOfState = 'Barack Obama' WHERE Name = 'United States'

OK, UPDATE done.

UPDATE CountryLanguage SET language = 'American English' WHERE countryCode = 'USA' AND language = 'English'

SQL: UPDATE CountryLanguage SET language = 'American English' WHERE countryCode = 'USA' AND language = 'English'

SELECT Name, HeadOfState FROM Country WHERE Code = 'USA'

Fix & Format (FF).

SELECT \* FROM countryLanguage WHERE CountryCode = 'USA' AND Percentage > 0.5

Fix & Format (FF).

SELECT COUNT(\*) FROM Country

Fix & Format (FF).

SELECT COUNT(\*) FROM CountryLanguage

Fix & Format (FF).

EXITING PROGRAM

**package** mySQL;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** Main {

/\*\*

\* **@param** args

\* **@throws** IOException

\*/

**public** **static** **void** main(String[] args) **throws** IOException {

String url = "jdbc:mysql://localhost:3306/world";

String user = "kaminski";

String password = "cs3310";

UI ui = **new** UI("WorldTrans.txt");

ui.println("Connecting to MySQL...");

**try** {

Connection conn = DriverManager.*getConnection*(url, user, password);

ui.println("OK, the DB Connection is OPENED");

String line;

**while**(ui.hasNextLine()){

line = ui.nextLine();

**switch** (line.charAt(0)) {

**case** 'S':

ui.println(line);

line = UserApp.*selectHandler*(line);

DBAccess.*retrieveData*(conn, ui, line);

**break**;

**case** 'I':

ui.println(line);

line = UserApp.*insertHandler*(line);

DBAccess.*changeData*(conn, ui, line);

**break**;

**case** 'D':

ui.println(line);

line = UserApp.*deleteHandler*(line);

DBAccess.*changeData*(conn, ui, line);

**break**;

**case** 'U':

ui.println(line);

line = UserApp.*updateHandler*(line);

DBAccess.*changeData*(conn, ui, line);

**break**;

}

ui.println("");

}

conn.close();

} **catch** (SQLException e) {

ui.println("ERROR, DB Connection didn't work - no trans done");

}

ui.println("EXITING PROGRAM");

ui.finishUp();

}

}

**package** mySQL;

**import** java.util.Arrays;

**public** **class** UserApp {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Build appropiate sqlString for SELECT command.

\* **@param** sqlString

\* **@return**

\*/

**public** **static** String selectHandler(String sqlString){

**return** sqlString;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Build appropiate sqlString for INSERT command.

\* **@param** sqlString

\* **@return**

\*/

**public** **static** String insertHandler(String sqlString){

String[] temp;

sqlString = sqlString.substring(2,sqlString.length());

sqlString.replace("','","MMG");

System.*out*.println(sqlString);

temp = sqlString.split("|");

/\*It doesn't want to split by that and it makes no sense.\*/

**if** (temp.length > 2)

**return** String.*format*("INSERT INTO %s VALUES (%s);", temp[0], temp[1]);

**return** String.*format*("INSERT INTO %s(%s) VALUES (%s);", temp[0], temp[1], temp[2]);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Build appropiate sqlString for DELETE command.

\* **@param** sqlString

\* **@return**

\*/

**public** **static** String deleteHandler(String sqlString){

String[] temp;

sqlString = sqlString.substring(2,sqlString.length());

temp = sqlString.split("|");

**return** String.*format*("DELETE FROM %s WHERE %s = %s;", temp[0], temp[1], temp[2]);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Build appropiate sqlString for UPDATE command.

\* **@param** sqlString

\* **@return**

\*/

**public** **static** String updateHandler(String sqlString){

**return** sqlString;

}

}

**package** mySQL;

**import** java.io.IOException;

**import** java.sql.\*;

**public** **class** DBAccess {

**private** **static** String *command*;

**public** **static** **void** retrieveData(Connection conn, UI ui, String sqlString) **throws** IOException{

*getCommand*(sqlString);

**if** (*command*.equals("INSERT") || *command*.equals("DELETE"))

ui.println(String.*format*("SQL: %s", sqlString));

**try** {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sqlString);

ui.println("Fix & Format (FF).");

stmt.close();

} **catch** (SQLException e) {

ui.println("ERROR on "+ *command* + ", not done.");

}

}

**public** **static** **void** changeData(Connection conn, UI ui, String sqlString) **throws** IOException{

*getCommand*(sqlString);

ui.println(String.*format*("SQL: %s", sqlString));

**try** {

Statement stmt = conn.createStatement();

**int** result = stmt.executeUpdate(sqlString);

**if** (result != 0){

ui.println("OK, " + *command* + " done.");

}

stmt.close();

} **catch** (SQLException e) {

ui.println("ERROR on " + *command* + ", not done.");

}

}

**private** **static** **void** getCommand(String sqlString){

*command* = sqlString.substring(0,6);

}

}

**package** edu.wmich.cs3310.asgn4;

**package** mySQL;

**import** java.io.IOException;

**import** java.sql.\*;

**public** **class** DBAccess {

**private** **static** String *command*;

**public** **static** **void** retrieveData(Connection conn, UI ui, String sqlString) **throws** IOException{

*getCommand*(sqlString);

**if** (*command*.equals("INSERT") || *command*.equals("DELETE"))

ui.println(String.*format*("SQL: %s", sqlString));

**try** {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sqlString);

ui.println("Fix & Format (FF).");

stmt.close();

} **catch** (SQLException e) {

ui.println("ERROR on "+ *command* + ", not done.");

}

}

**public** **static** **void** changeData(Connection conn, UI ui, String sqlString) **throws** IOException{

*getCommand*(sqlString);

ui.println(String.*format*("SQL: %s", sqlString));

**try** {

Statement stmt = conn.createStatement();

**int** result = stmt.executeUpdate(sqlString);

**if** (result != 0){

ui.println("OK, " + *command* + " done.");

}

stmt.close();

} **catch** (SQLException e) {

ui.println("ERROR on " + *command* + ", not done.");

}

}

**private** **static** **void** getCommand(String sqlString){

*command* = sqlString.substring(0,6);

}

}