**MOVIE DATA ANALYSIS REPORT**

1. **Creating a new SQLite database**– Learn how to create a new SQLite database from a Java (any language) program using SQLiteJDBC driver or related driver.

Code:

package net.sqlitetutorial;

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.DriverManager;

import java.sql.SQLException;

public class Createdb {

public static void createNewDatabase(String fileName) {

String url = "jdbc:sqlite:C:\\sqlite\\db\\" + fileName;

try (Connection conn = DriverManager.getConnection(url)) {

if (conn != null) {

DatabaseMetaData meta = conn.getMetaData();

System.out.println("The driver name is " + meta.getDriverName());

System.out.println("A new database has been created.");

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

public static void main(String[] args) {

createNewDatabase("cinema\_fav.db");

}

}

1. **Connect to the SQLite database** (or any Database you know): Learn how to download SQLiteJDBC driver and connect to an existing SQLite database using JDBC.

Code:

package net.sqlitetutorial;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class Connect {

/\*\*

\* Connect to a sample database

\*/

public static void connect()

{

Connection conn = null;

try {

// db parameters

String url = "jdbc:sqlite:C:\\sqlite\\db\\cinema\_fav.db";

// create a connection to the database

conn = DriverManager.getConnection(url);

System.out.println("Connection to SQLite has been established.");

} catch (SQLException e) {

System.out.println(e.getMessage());

} finally {

try {

if (conn != null) {

conn.close();

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

}

}

}

/\*\*

\* args the command line arguments

\*/

public static void main(String[] args) {

connect();

}

}  
  
  
**3.** **Creating a new table (Movies)**using JDBC / Other Languages – before working with data, you need to create a table called Movies. Learn how to create a new table in an SQLite database from a Java (any language) program.

Code:

package net.sqlitetutorial;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

public class CreateTable {

/\*\*

\* Create a new table in the test database

\*

\*/

public static void createNewTable() {

// SQLite connection string

String url = "jdbc:sqlite:C:\\sqlite\\db\\cinema\_fav.db";

// SQL statement for creating a new table

String sql = "CREATE TABLE IF NOT EXISTS movies (\n"

+ " Movie\_id integer PRIMARY KEY,\n"

+ " Movie\_name text NOT NULL,\n"

+ " Lead\_Actor text NOT NULL,\n"

+ " Lead\_Actress text NOT NULL,\n"

+ " Director text NOT NULL,\n"

+ " Year\_of\_Release date NOT NULL\n"

+ ");";

try (Connection conn = DriverManager.getConnection(url);

Statement Stmt = conn.createStatement()) {

// create a new table

Stmt.execute(sql);

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

/\*\*

\* args the command line arguments

\*/

public static void main(String[] args) {

createNewTable();

}

}  
  
**4. Inserting data into Movies table** from a Java (any language) program  
Code:

package net.sqlitetutorial;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

/\*\*

\*

\*/

public class InsertData{

/\*\*

\* Connect to the cinema\_fav.db database

\*

\* @return the Connection object

\*/

private Connection connect() {

// SQLite connection string

String url = "jdbc:sqlite:C:\\sqlite\\db\\cinema\_fav.db";

Connection conn = null;

try {

conn = DriverManager.getConnection(url);

} catch (SQLException e) {

System.out.println(e.getMessage());

}

return conn;

}

/\*\*

\* Insert a new row into the MOVIES table

\*

\*/

public void insert(String Movie\_name, String Lead\_Actor, String Lead\_Actress, String Director, String Year\_of\_Release) {

String sql = "INSERT INTO movies(Movie\_name,Lead\_Actor,Lead\_Actress,Director,Year\_of\_Release) VALUES(?,?,?,?,?)";

try (Connection conn = this.connect();

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setString(1, Movie\_name);

pstmt.setString(2, Lead\_Actor);

pstmt.setString(3, Lead\_Actress);

pstmt.setString(4, Director);

pstmt.setString(5, Year\_of\_Release);

pstmt.executeUpdate();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

/\*\*

\* args the command line arguments

\*/

public static void main(String[] args) {

InsertData app = new InsertData();

// inserting 15 new rows

app.insert("K.G.F: Chapter 1","Yash","Srinidhi Shetty","Prashanth Neel","21-December-2018");

app.insert("Tum Bin","Priyanshu Chatterjee","Sandali Sinha","Anubhav Sinha","13-July-2001");

app.insert("Baahubali: The Beginning", "Prabhas", "Anushka Shetty", "S. S. Rajamouli","10-July-2015");

app.insert("Baahubali 2: The Conclusion","Prabhas","Anushka Shetty","S. S. Rajamouli","28-April-2017");

app.insert("Junglee","Vidyut Jammwal","Pooja Sawant","Chuck Russell","29-March-2019");

app.insert("Dangal","Aamir Khan","Fatima Sana Shaikh","Nitesh Tiwari","23-December-2016");

app.insert("3 Idiots","Aamir Khan","Kareena Kapoo","Rajkumar Hirani","25-December-2009");

app.insert("Thugs of Hindostan","Aamir Khan","Fatima Sana Shaikh","Vijay Krishna Acharya","8-November-2018");

app.insert("Lagaan","Aamir Khan","Gracy Singh","Ashutosh Gowariker","15-June-2001");

app.insert("Mangal Pandey: The Rising","Aamir Khan","Rani Mukerji","Ketan Mehta","12-August-2005");

app.insert("Drishyam","Ajay Devgan","Tabu","Nishikant Kamat","31-July-2015");

app.insert("Shivaay","Ajay Devgan","Erika Kaar","Ajay Devgn","28-October-2016");

app.insert("Raid","Ajay Devgan","Ileana D Cruz","Raj Kumar Gupta","16-March-2018");

app.insert("Tanhaji","Ajay Devgan","Kajol","Om Raut","10-January-2020");

app.insert("Kesari","Akshay Kumar","Parineeti Chopra","Anurag Singh","21-March-2019");

}

}

**5. Querying data from Movies table** with or without parameters – after having the movies data in the table, you need to query the movie details (name, actor, actress, director, year of release) using a SELECT statement. You will need to write a program to issue a simple SELECT statement to query all rows from the Movies table, as well as use a query with parameter like actor name to select movies based on the actor's name.

Code: a query without parameter

package net.sqlitetutorial;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class SelectData {

/\*\*

\* Connect to the cinema\_fav.db database

\* @return the Connection object

\*/

private Connection connect() {

// SQLite connection string

String url = "jdbc:sqlite:C:\\sqlite\\db\\cinema\_fav.db";

Connection conn = null;

try {

conn = DriverManager.getConnection(url);

} catch (SQLException e) {

System.out.println(e.getMessage());

}

return conn;

}

/\*\*

\* select all rows in the MOVIES table

\*/

public void selectAll(){

String sql = "SELECT Movie\_id, Movie\_name, Lead\_Actor, Lead\_Actress, Director, Year\_of\_Release FROM movies";

try (Connection conn = this.connect();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)){

// loop through the result set

while (rs.next()) {

System.out.println("\n" + rs.getInt("Movie\_id") + "\t" +

rs.getString("Movie\_name") + "\t" +

rs.getString("Lead\_Actor") + "\t" +

rs.getString("Lead\_Actress") + "\t" +

rs.getString("Director") + "\t" +

rs.getString("Year\_of\_Release"));

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

/\*\*

\* @args the command line arguments

\*/

public static void main(String[] args) {

SelectData app = new SelectData();

app.selectAll();

}

}

Code: a query with parameter

package net.sqlitetutorial;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class SelectdataWP{

/\*\*

\* Connect to the cinema\_fav.db database

\* @return the Connection object

\*/

private Connection connect() {

// SQLite connection string

String url = "jdbc:sqlite:C:\\sqlite\\db\\cinema\_fav.db";

Connection conn = null;

try {

conn = DriverManager.getConnection(url);

} catch (SQLException e) {

System.out.println(e.getMessage());

}

return conn;

}

/\*\*

\* select all rows from the MOVIES table with parameter

\*/

public void selectAll(){

String sql = "SELECT Movie\_id, Movie\_name, Lead\_Actor, Lead\_Actress, Director, Year\_of\_Release FROM movies WHERE Lead\_Actor like 'Ajay Devgan' ORDER BY Movie\_id ";

try (Connection conn = this.connect();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)){

// loop through the result set

while (rs.next()) {

System.out.println("\n" + rs.getInt("Movie\_id") + "\t" +

rs.getString("Movie\_name") + "\t" +

rs.getString("Lead\_Actor") + "\t" +

rs.getString("Lead\_Actress") + "\t" +

rs.getString("Director") + "\t" +

rs.getString("Year\_of\_Release"));

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

/\*\*

\* @args the command line arguments

\*/

public static void main(String[] args) {

SelectdataWP app = new SelectdataWP();

app.selectAll();

}

}

**In command prompt I have executed the following coding steps to compile all the above code.**

1. **cd C:\sqlite\java\connect\net\sqlitetutorial**
2. **javac Createdb.java**
3. **javac Connect.java**
4. **javac CreateTable.java**
5. **cd..**
6. **cd..**
7. **java -classpath ".;sqlite-jdbc-3.32.3.3.jar" net.sqlitetutorial.Createdb**
8. **java -classpath ".;sqlite-jdbc-3.32.3.3.jar" net.sqlitetutorial.Connect**
9. **java -classpath ".;sqlite-jdbc-3.32.3.3.jar" net.sqlitetutorial.CreateTable**
10. **cd C:\sqlite\java\connect\net\sqlitetutorial**
11. **javac InsertData.java**
12. **cd..**
13. **cd..**
14. **java -classpath ".;sqlite-jdbc-3.32.3.3.jar" net.sqlitetutorial.InsertData**
15. **cd C:\sqlite\java\connect\net\sqlitetutorial**
16. **javac SelectData.java**
17. **cd..**
18. **cd..**
19. **java -classpath ".;sqlite-jdbc-3.32.3.3.jar" net.sqlitetutorial.SelectData**
20. **cd C:\sqlite\java\connect\net\sqlitetutorial**
21. **javac SelectdataWP.java**
22. **cd..**
23. **cd..**
24. **java -classpath ".;sqlite-jdbc-3.32.3.3.jar" net.sqlitetutorial.SelectdataWP**

**-------------------+++++++/\*\+++++++-------------------**

**END**