**Join, Metadata & view in jdbc**

**--------------------------------------------🡪**

**Methods**

**--------------------------------------------🡪**

**package** com.company;  
  
**import** java.sql.\*;  
**import** java.util.ArrayList;  
**import** java.util.Iterator;  
**import** java.util.List;  
  
**public class** Methods {  
 Connection **connection** = **null**;  
 Statement **statement** = **null**;  
 PreparedStatement **preparedStatement** = **null**;  
 ResultSet **resultSet** = **null**;  
 String **select\_all** = **"select \* from songs"**;  
 String **select\_artist\_name** = **"select \* from artists"**;  
 String **join\_song\_alb** = **"select songs.track, songs.title, albums.name from songs songs inner join albums albums on songs.album=albums.\_id"**;  
 String **Join\_song\_alb\_orderBy** = **" select songs.track, songs.title, albums.name from songs songs inner join albums albums on songs.album=albums.\_id order by songs.title asc;"**;  
 String **create\_view** = **"create view if not exists catch as select songs.title, albums.name, albums.artist from songs songs inner join albums albums on songs.album=albums.\_id;"**;  
 String **query\_view\_catch** = **"select title, name from catch order by title asc;"**;  
  
  
 *//fetching data* **public void** show\_data(){  
 **try** {  
 Connections c = **new** Connections(); *//connection class under com.company* connection = c.getConn();  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(select\_all);  
 **while** (resultSet.next()){  
 System.out.println(**"Id : "** + resultSet.getInt(1) + **" track : "** + resultSet.getInt(2)  
 + **" Title : "** + resultSet.getString(3) + **" Album : "** + resultSet.getInt(4));  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 *//joins songs -> albums* **public void** joinSongs\_Album(){  
 **try** {  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 statement = connection.createStatement();  
 *//no value will be passed so no need for prepared statement* resultSet = statement.executeQuery(join\_song\_alb);  
  
 **while** (resultSet.next()){  
 System.out.println(**"Song name : "** + resultSet.getString(2)+ **" || Artist : "** + resultSet.getString(3));  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public void** joinSongs\_Album\_orderby(){  
 **try** {  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 statement = connection.createStatement();  
 *//no value will be passed so no need for prepared statement* resultSet = statement.executeQuery(Join\_song\_alb\_orderBy);  
  
 **while** (resultSet.next()){  
 System.out.println(**"Song name : "** + resultSet.getString(2)+ **" || Artist : "** + resultSet.getString(3));  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public void** getByArtist(String x){  
 **try** {  
 Connections c = **new** Connections();  
 connection = c.getConn();  
*// preparedStatement = connection.prepareStatement(album\_by\_artist);  
// preparedStatement.setInt(1, 8);  
//// resultSet = (ResultSet)preparedStatement.execute();  
//// connection.commit();* String album\_by\_artist = **"select \* from albums where artist="** + x;  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(album\_by\_artist);  
  
 **while** (resultSet.next()){  
 System.out.println(resultSet.getString(2));  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public void** getByArtistFilter(String x){  
 **try** {  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 String album\_and\_songs\_by\_artist = **"select songs.title, albums.name from "** +  
 **"songs songs inner join albums albums on songs.album=albums.\_id "** +  
 **"where albums.artist = "** + x + **" order by songs.title asc;"**;  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(album\_and\_songs\_by\_artist);  
  
 **while** (resultSet.next()){  
 System.out.println(**"Song name : "** + resultSet.getString(1) + **"|| Album : "** + resultSet.getString(2));  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public void** getByArtistFilterwith3tables(String x){  
 **try** {  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 String album\_and\_songs\_name\_by\_artist = **"select songs.title, albums.name, artists.name "** +  
 **"from songs songs inner join albums albums on songs.album=albums.\_id "** +  
 **"inner join artists artists on albums.artist=artists.\_id "** +  
 **"where albums.artist = "**+ x +**" order by songs.title asc;"**;  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(album\_and\_songs\_name\_by\_artist);  
  
 **while** (resultSet.next()){  
 System.out.println(**"Song name : "** + resultSet.getString(1) + **"|| Album : "** + resultSet.getString(2) + **"|| Artist : "** + resultSet.getString(3));  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public void** putArtistnameinlist(){  
 **try**{  
 List<String> list = **new** ArrayList<>();  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(select\_artist\_name);  
  
 **while** (resultSet.next()){  
 list.add(resultSet.getString(2));  
 }  
  
 *// iterator : to iterate through the list* Iterator iterator = list.iterator();  
 **while** (iterator.hasNext()){  
 Object element = iterator.next();  
 System.out.println(element);  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }

**public boolean** putArtistnameinlistCheck(String name){  
 **try**{  
 List<String> list = **new** ArrayList<>();  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(select\_artist\_name);  
  
 **while** (resultSet.next()){  
 list.add(resultSet.getString(2));  
 }  
  
 *// iterator : to iterate through the list* Iterator iterator = list.iterator();  
 **while** (iterator.hasNext()){  
 Object element = iterator.next();  
 **if** (element.equals(name)) {  
 **return true**;  
 }  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 **return false**;  
 }  
  
 *// getting the informations about the columns in songs table* **public void** querySongsMetadata(){  
 **try** {  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(select\_all);  
 ResultSetMetaData resultSetMetaData = resultSet.getMetaData(); *//fetching col names* **int** columns = resultSetMetaData.getColumnCount();  
 **for** (**int** i = 1; i <= columns; i++){  
 String type = resultSetMetaData.getColumnTypeName(i);  
 System.out.println(**"Column : "** + i + **" name : "** + resultSetMetaData.getColumnName(i) + **" type : "** + type);  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 *// Aggregation functions* **public void** aggregationFunction(){  
 String select = **"select count(\*), min(\_id) from songs"**;  
 **try** {  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(select);  
 **int** count = resultSet.getInt(1); *// in this type of queries we need to use COLUMN NUMBER as COUNT PARAMETER* **int** lowest\_id = resultSet.getInt(2); *// for min* System.out.println(**"no of element is : "** + count + **" Min \_id is : "** + lowest\_id);  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 *// create view* **public boolean** createsView(){  
 **try** {  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 **statement**.execute(**create\_view**); *//no need to put it in result set as it returns nothing* **return true**;  
 } **catch** (Exception e){  
 e.printStackTrace();  
 **return false**;  
 }  
 }

*// query view* **public void** queryView(){  
 **try**{  
 Connections c = **new** Connections();  
 connection = c.getConn();  
 statement = connection.createStatement();  
 resultSet = statement.executeQuery(query\_view\_catch);  
  
 **while** (resultSet.next()){  
 System.out.println(**"Song : "** + resultSet.getString(1) + **" || Artist : "** + resultSet.getString(2));  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
}

**------------------------------🡪**

**Connection**

**-------------------------------🡪**

**package** com.company;  
  
**import** java.sql.\*;  
**public class** Connections {  
 Connection **con**=**null**;  
 **public final** String **Db\_name** = **"music.db"**;  
 *//find the .jar here -> D:\apps\sqlite-jdbc-3.20.1.jar* **public** Connection getConn() {  
 **try** {  
 **con**=DriverManager.*getConnection*(**"jdbc:sqlite:C:\\Java\\JDBC\\"** + **Db\_name**);  
 } **catch**(Exception se) {  
 se.printStackTrace();  
 }  
 **return con**;  
 }  
}

**------------------------------🡪**

**Main**

**-------------------------------🡪**

**package** com.company;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
 Methods methods = **new** Methods();  
 methods.show\_data();  
 methods.joinSongs\_Album();  
 methods.joinSongs\_Album\_orderby();  
 methods.getByArtist(**"8"**);  
 methods.getByArtistFilter(**"8"**);  
 methods.getByArtistFilterwith3tables(**"8"**);  
 methods.putArtistnameinlist();  
 **if** (methods.putArtistnameinlistCheck(**"MetallicaP"**))  
 System.***out***.println(**"found"**);  
 **else** System.***out***.println(**"not found"**);  
 methods.querySongsMetadata();  
 methods.aggregationFunction();  
 **if** (methods.createsView())  
 System.***out***.printf(**"View is created"**);  
 **else** System.***out***.println(**"View exists"**);  
 methods.queryView();  
 }  
}