**JDBC full**

**-----------------------🡪Method**

**package** com.company;  
  
**import** java.sql.\*;  
**import** java.util.\*;  
  
**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;"**;  
 String **tomaps** = **"select** *count***(***\****) from catch;"**;  
 String **join\_three** = **"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 artists.name='Metallica' order by songs.title asc;"**;  
 String **insert\_into\_songs** = **"insert into songs values(?,?,?,?)"**;  
 String **insert\_into\_albums** = **"insert into albums values(?,?,?)"**;  
 String **insert\_into\_artists** = **"insert into artists values(?,?)"**;  
  
  
 *//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();  
 }  
 }  
  
 *//pushing in map to check availability* **public boolean** queryViewMap(Object name){  
 **try**{  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 **resultSet** = **statement**.executeQuery(**tomaps**);  
 **int** counts\_catch = **resultSet**.getInt(1); *// as we know AGGREGATE function can appear like this way* ResultSet resultSet1 = **statement**.executeQuery(**query\_view\_catch**);  
 Map map = **new** HashMap();  
 **int** j = 1;  
 **while** (resultSet1.next()){  
 String grant = resultSet1.getString(2);  
 map.put(j, grant);  
 j++;  
 }  
  
 System.***out***.println(map);  
  
 **if** (map.containsValue(name))  
 **return true**;  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 **return false**;  
 }  
  
 *// inserting values in 3 tables in one shot - transaction -- LET THE SINGER HAS NO ENTRY BEFORE* **public void** addSong(String Song\_title,String Song\_track, String album\_name, String artist\_name){  
 String song\_id\_count = **"select** *max***(\_id) from songs;"**;  
 String albums\_id\_count = **"select** *max***(\_id) from albums;"**;  
 String artist\_id\_count = **"select** *max***(\_id) from artists;"**;  
  
 **try** {  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 ResultSet resultSet1, resultSet2;  
 PreparedStatement preparedStatement1, preparedStatement2;  
  
 **resultSet** = **statement**.executeQuery(song\_id\_count);  
 **int** resultset\_song = **resultSet**.getInt(1);  
 resultSet1 = **statement**.executeQuery(albums\_id\_count);  
 **int** resultset\_albums = resultSet1.getInt(1);  
 resultSet2 = **statement**.executeQuery(artist\_id\_count);  
 **int** resultset\_artists = resultSet2.getInt(1);  
  
 **preparedStatement** = **connection**.prepareStatement(**insert\_into\_artists**);  
 **preparedStatement**.setInt(1, resultset\_artists+1);  
 **preparedStatement**.setString(2, artist\_name);  
 **preparedStatement**.executeUpdate();  
  
 preparedStatement1 = **connection**.prepareStatement(**insert\_into\_albums**);  
 preparedStatement1.setInt(1, resultset\_albums+1);  
 preparedStatement1.setString(2, album\_name);  
 preparedStatement1.setInt(3, resultset\_artists+1);  
 preparedStatement1.executeUpdate();  
  
 preparedStatement2 = **connection**.prepareStatement(**insert\_into\_songs**);  
 preparedStatement2.setInt(1, resultset\_song+1);  
 preparedStatement2.setInt(2, Integer.*parseInt*(Song\_track));  
 preparedStatement2.setString(3, Song\_title);  
 preparedStatement2.setInt(4, resultset\_albums+1);  
 preparedStatement2.executeUpdate();  
  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public int** checkArtist(String name){  
 **int** i = 0;  
 **try** {  
 String q1 = **"select \_id from artists where name = '"** + name + **"';"**;  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 **resultSet** = **statement**.executeQuery(q1);  
  
 **while** (**resultSet**.next()){  
 i = **resultSet**.getInt(1);  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
  
 }  
 **return** i;  
 }  
  
 **public int** checkAlbums(String albumsname){  
 **int** i = 0;  
 **try** {  
 String q1 = **"select \_id from albums where name = '"** + albumsname + **"';"**;  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 **resultSet** = **statement**.executeQuery(q1);  
  
 **while** (**resultSet**.next()){  
 i = **resultSet**.getInt(1);  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
  
 }  
 **return** i;  
 }  
  
 **public int** checkSongs(String songsname){  
 **int** i = 0;  
 **try** {  
 String q1 = **"select \_id from albums where name = '"** + songsname + **"';"**;  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 **resultSet** = **statement**.executeQuery(q1);  
  
 **while** (**resultSet**.next()){  
 i = **resultSet**.getInt(1);  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
  
 }  
 **return** i;  
 }  
  
 **public void** triggerInsert(String S\_track, String S\_title, String Alb\_name, String Art\_name){  
  
 **try** {  
 **int** a = checkArtist(Art\_name); *//artist.\_id* **int** b = checkAlbums(Alb\_name); *//albums.\_id* **int** c = checkSongs(S\_title); *// songs.id* Connections con = **new** Connections();  
 **connection** = con.getConn();  
 **statement** = **connection**.createStatement();  
 PreparedStatement preparedStatement1, preparedStatement2, preparedStatement3;  
 **int** maxAlbum\_id = maxAlbum\_id();  
 **int** maxSongs\_id = maxSong\_id();  
  
 **if** (a == 0) {  
 addSong(S\_title, S\_track, Alb\_name, Art\_name);  
 **return**;  
 } **else** {  
 **if** (b == 0) {  
 *//has no album - add  
 //updating album* preparedStatement1 = **connection**.prepareStatement(**insert\_into\_albums**);  
 preparedStatement1.setInt(1, maxAlbum\_id+1);  
 preparedStatement1.setString(2, Alb\_name);  
 preparedStatement1.setInt(3, a);  
 preparedStatement1.executeUpdate();  
  
 *//updating songs* preparedStatement2 = **connection**.prepareStatement(**insert\_into\_songs**);  
 preparedStatement2.setInt(1, maxSongs\_id+1);  
 preparedStatement2.setInt(2, Integer.*parseInt*(S\_track));  
 preparedStatement2.setString(3, S\_title);  
 preparedStatement2.setInt(4, maxAlbum\_id+1);  
 preparedStatement2.executeUpdate();  
 **return**;  
 } **else** {  
 **int** Alb\_id = b;  
 **int** Alb\_art = a;  
 **if** (c == 0) {  
 *// has nom song - add* preparedStatement3 = **connection**.prepareStatement(**insert\_into\_songs**);  
 preparedStatement3.setInt(1, maxSongs\_id+1);  
 preparedStatement3.setInt(2, Integer.*parseInt*(S\_track));  
 preparedStatement3.setString(3, S\_title);  
 preparedStatement3.setInt(4, maxAlbum\_id+1);  
 preparedStatement3.executeUpdate();  
 **return**;  
 } **else** {  
 System.***out***.println(**"Song is in the list with id : "** + c);  
 }  
 }  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public int** maxAlbum\_id(){  
 **int** x = 0;  
 **try** {  
 String q1 = **"select** *max***(\_id) from albums"**;  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 **resultSet** = **statement**.executeQuery(q1);  
 x = **resultSet**.getInt(1);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return** x;  
 }  
  
 **public int** maxSong\_id(){  
 **int** x = 0;  
 **try** {  
 String q1 = **"select** *max***(\_id) from songs"**;  
 Connections c = **new** Connections();  
 **connection** = c.getConn();  
 **statement** = **connection**.createStatement();  
 **resultSet** = **statement**.executeQuery(q1);  
 x = **resultSet**.getInt(1);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return** x;  
 }  
}

**------------------------------🡪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();  
 ------------------------------------------------------------------------------  
 **if** (methods.queryViewMap(**"Rainbow"**))  
 System.***out***.println(**"Exists"**);  
 **else** System.***out***.println(**"Doesn't exists"**);  
 ---------------------------------------------------------------------------------  
 methods.addSong(**"Despacito"**, **"100"**, **"Fonsi"**, **"DY"**);  
 --------------------------------------------------------------------------------------  
 NOT WORKING>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  
 methods.addSong(**"love me like you do"**, **"100"**, **"Gray"**, **"Golding"**);  
 methods.triggerInsert(**"100"**, **"love me like you do"**, **"Gray"**, **"Golding"**);  
 *//--------------------------------------------------------------------------------------* }  
}

**-----------------🡪Connections**

**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**;  
 }  
}