```
 1 package assignment2;
 2
 3 /**Michael Masterson
10
11 import java.sql.*;
12 public class assignment2
13 {
14     public static void main(String args[])
15     {
16         //start connection to JDBC
17         Connection c = null;
18         Statement stmt = null;
19         try{
20             Class.forName("org.sqlite.JDBC");
21             c = DriverManager.getConnection("jdbc:sqlite:C:/Sqlite/chinook.db");
22             c.setAutoCommit(false);
23             System.out.println("Opened database");
24
25             //start creating queries, comment in
26             stmt = c.createStatement();
27             /**
28             //1.) Find the name of artists with more than 30 tracks
29             ResultSet rs = stmt.executeQuery("SELECT Name, COUNT(*)"
30                     +                        "FROM tracks, albums, artists"
31                     +                        "WHERE albums.AlbumId = tracks.AlbumId AND
   artists.ArtistId = ablums.ArtistId"
32                     +                        "GROUP BY artists.ArtistId, artists.Name"
33                     +                        "HAVING COUNT (*) >30");
34
35             while(rs.next())
36             {
37                 String Name = rs.getString("");
38
39                 System.out.println("Artist Name: " + Name);
40                 System.out.println();
41             }*/
42
43             /**
44             //2.) Find the most popular Rock playlist, i.e. playlist(s) which contain
45             //the largest number of Rock tracks
46             ResultSet rs = stmt.executeQuery("SELECT PlaylistId, Name"
47                     + "FROM playlists, playlist_track, tracks,media_types,genres"
48                     + "WHERE media_types.Name LIKE'%audio%' AND playlists.PlaylistId =
   playlist_track.PlaylistId"
49                     +                                    "AND playlist_track.TrackId =
   tracks.TrackId"
50                     +                                    "AND tracks.MediaTypeId =
   media_types.MediaTypeId"
51                     +                                    "AND tracks.GenreId =
   genres.GenreID AND genres.Name ='Rock'"
52                     +"GROUP BY PlaylistId, Name"
53                     + "HAVING COUNT(*) = (SELECT MAX(aux.count)"
54                                         +"FROM(SELECT COUNT(*) as count"
55                                           +   "FROM playlists, playlist_track,
   tracks,media_types,genres"
56                                           +   "media_types.Name LIKE'%audio%' AND
   playlists.PlaylistId = playlist_track.PlaylistId"
```

```
57                                          +   "          AND playlist_track.TrackId =
  tracks.TrackId"
58                                          +   "          AND tracks.MediaTypeId =
  media_types.MediaTypeId"
59                                          +   "          AND tracks.GenreId = genres.GenreID
  AND genres.Name ='Rock'"
60                                          +   "                                    GROUP BY
  P.PlaylistId )aux)");
61
62            while(rs.next())
63            {
64                String name = rs.getString("");
65
66                System.out.println("Most Popular Rock Playlist: " + name);
67                System.out.println();
68            }*/
69
70
71            /**
72            //3.) Find the playlist that contains most tracks by artists "Aerosmith" (no View)
73             ResultSet rs = stmt.executeQuery("SELECT Name, COUNT(*)"
74                    + "FROM playlists AS P, playlist_track AS PL, tracks,albums,artists"
75                    + "WHERE P.PlaylsitId = PL.PlaylistId AND PL.TrackId = tracks.TrackId AND
  albums.AlbumId = tracks.AlbumId"
76                    +                                   "AND artists.ArtistId =
  albums.ArtistId AND artists.Name='AC/DC'"
77                    +"GROUP BY PlaylistId, Name"
78                    + "HAVING COUNT(*) = (SELECT MAX(aux.count)"
79                    +                     "FROM(SELECT COUNT(*) as count"
80                    +                          "FROM playlists AS P, playlist_track AS PL,
  tracks,tracks,albums,artists"
81                    +                          "WHERE P.PlaylistId - PL.PlaylisteId AND
  PL.TrackId = tracks.TrackId AND ablums.AlbumId = tracks.AlbumId AND artists.ArtistId =
  albums.ArtistId AND artists.Name='AC/DC'"
82                    +                          "GROUP BY P.PlaylistId) aux");
83
84
85            while(rs.next())
86            {
87                String name = rs.getString("");
88
89                System.out.println("Playlist with most tracks with Aerosmith: " + name);
90                System.out.println();
91            }*/
92
93            /**
94            //4.) Find the genre of tracks which is contained in the most playlist (no View)
95            ResultSet rs = stmt.executeQuery("SELECT GenreID,Name, COUNT(Distinct PlaylistId"
96                                          + "FROM genres, tracks, playlist_track"
97                                          +"WHERE genres.GenreId = tracks.GenreID AND
  tracks.TrackId = playlist_track.TrackId"
98                                          +"GROUP BY genres.GenreId, genres.Name"
99                                          +"HAVING COUNT (DISTINCT PlaylistId) = (SELECT
  MAX(aux.count)"
100                                         +                                   "FROM(SELE
  CT COUNT(DISTINCT PlaylistId) as count"
101                                         +                                   "FROM
```

```
genres, tracks, playlist_track"
102                                         +                                   "WHERE
genres.GenreId = tracks.GenreID AND tracks.TrackId = playlist_track.TrackId"
103                                         +                                   "GROUP BY
genres.GenreId, genres.Name) aux");
104
105             while(rs.next())
106             {
107                 String name = rs.getString("");
108
109                 System.out.println("Genre with most tracks: " + name);
110                 System.out.println();
111             }*/
112
113
114             /**
115             //5.) Find the number of employees live in the same city with each customer
116             ResultSet rs = stmt.executeQuery("SELECT EmployeeId, City"
117                     + "FROM customers, employees"
118                     + "WHERE EmployeeId = City"
119                     + "GROUP BY City"
120                     + "COUNT (*)");
121
122             while(rs.next())
123             {
124                 String City = rs.getString("");
125
126                 System.out.println("Employees in City: " + City);
127                 System.out.println();
128             }*/
129
130             /**
131             //6.) Find artist has the most "Rocks" and "Metal" tracks combined
132             ResultSet rs = stmt.executeQuery("SELECT artists, genres"
133                                         + "FROM genres, artists"
134                                         + "WHERE artists == genres.Name ='Rock' AND
genres.Name ='Metal'");
135
136
137             while(rs.next())
138             {
139                 String art = rs.getString("");
140
141                 System.out.println("Artist with Metal and Rocks songs: " + art);
142                 System.out.println();
143             }*/
144
145             /**
146             //7.) Find MPEG/MPEG4 tracks which have a length longer than the
147             // average length of all the MPEG/MPEG4 tracks
148             ResultSet rs = stmt.executeQuery("SELECT Name AS name"
149                     + "FROM tracks, media_types"
150                     + "WHERE tracks.MediaTypeId = MediaTypeId AND media_types.Name LIKE
'%audio%'"
151                     +                                   "AND tracks.Milliseconds >
(SELECT AVG(Milliseconds)"
152                     +                                                       "FROM
```

```
   tracks, media_types"
153                     +                                                                   "WHERE
   tracks.MediaTypeId = media_types.MediaTypeId AND media_types.Name LIKE'%audio%')");
154
155             while(rs.next())
156             {
157                 String Name = rs.getString("");
158
159                 System.out.println("MPEG Song: " + Name);
160                 System.out.println();
161             }*/
162
163             /**
164             //8.) Find the name and phone of the third oldest (age) employee
165             ResultSet rs = stmt.executeQuery("SELECT a.Name,a.Age, a.Phone"
166                     + "FROM employees a"
167                     + "WHERE 3= (SELECT COUNT(DISTINCT(b.Age,b.Phone))"
168                     +           "FROM employees b"
169                     +           "WHERE a.Age <= b.Age AND a.Phone <= b.Phone");
170
171             while(rs.next())
172             {
173                 String name = rs.getString("");
174                 String phone = rs.getString("");
175
176                 System.out.println("Name: " + name);
177                 System.out.println("Phone: " + phone);
178                 System.out.println();
179             }*/
180             rs.close();
181             stmt.close();
182             c.close();
183         }catch (Exception e)
184         {
185             System.out.println(e.getClass().getName()+": "+e.getMessage());
186             System.exit(0);
187         }
188         System.out.println("opened");
189     }
190 }
```