PSTAT 10 Worksheet 8

Setup

```
setwd('C:/Users/visha/OneDrive/Documents/School/Summer 2024/PSTAT 10/PSTAT10-Worksheets-and-HW')
library(RSQLite)
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
library(DBI)
chinook_db <- dbConnect(SQLite(), "Chinook_Sqlite.sqlite")
```

Problem 1

1.

```
dbGetQuery(chinook_db, "SELECT GenreId, sum(milliseconds)/60000
                        FROM Track
                        WHERE GenreId = 2")
```

```
##   GenreId sum(milliseconds)/60000
## 1       2                     632
```

2.

```
dbGetQuery(chinook_db, "SELECT Genre.Name, Genre.GenreId, sum(milliseconds)/60000
                        FROM Track JOIN Genre
                        ON Track.GenreId = Genre.GenreId
                        WHERE Genre.GenreId = 2")
```

```
##   Name GenreId sum(milliseconds)/60000
## 1 Jazz       2                     632
```

3.

```
dbGetQuery(chinook_db, "SELECT Genre.Name, Genre.GenreId, sum(milliseconds)/60000
                        FROM Track JOIN Genre
                        ON Track.GenreId = Genre.GenreId
                        GROUP BY Genre.GenreId")
```

```
##                  Name GenreId sum(milliseconds)/60000
## 1                Rock       1                    6137
## 2                Jazz       2                     632
## 3               Metal       3                    1930
## 4  Alternative & Punk       4                    1296
## 5       Rock And Roll       5                      26
## 6               Blues       6                     364
## 7               Latin       7                    2247
## 8              Reggae       8                     238
## 9                 Pop       9                     183
## 10         Soundtrack      10                     175
## 11         Bossa Nova      11                      54
## 12     Easy Listening      12                      75
## 13        Heavy Metal      13                     138
## 14           R&B/Soul      14                     223
## 15  Electronica/Dance      15                     151
## 16              World      16                     104
## 17        Hip Hop/Rap      17                     103
## 18    Science Fiction      18                     568
## 19           TV Shows      19                    3324
## 20   Sci Fi & Fantasy      20                    1261
## 21              Drama      21                    2746
## 22             Comedy      22                     449
## 23        Alternative      23                     176
## 24          Classical      24                     362
## 25              Opera      25                       2
```

Problem 2

```
dbGetQuery(chinook_db, "SELECT Employee.EmployeeId AS RepId, Employee.FirstName as RepFirstName,
                        Employee.LastName as RepLastName, CustomerId, Customer.FirstName as CustFirstNa
                        Customer.LastName as CustLastName
                        FROM Customer JOIN Employee
                        ON Customer.SupportRepId = Employee.EmployeeId
                        WHERE Customer.FirstName = 'Heather' AND Customer.LastName = 'Leacock'")
```

```
##   RepId RepFirstName RepLastName CustomerId CustFirstName CustLastName
## 1     4     Margaret        Park         22       Heather      Leacock
```

Problem 3

1.

```
dbGetQuery(chinook_db, "SELECT count(DISTINCT Country)
                        FROM Customer")
```

```
##   count(DISTINCT Country)
## 1                      24
```

Customers are from 24 different countries.

2.

```
dbGetQuery(chinook_db, "SELECT DISTINCT Title
                        FROM Employee")
```

```
##                 Title
## 1     General Manager
## 2       Sales Manager
## 3 Sales Support Agent
## 4          IT Manager
## 5            IT Staff
```

3. No, distinct was not necessary here. GenreId is the primary key of Genre, and in order to maintain entity integrity every row would need to have a *unique* value for GenreId - thus every value for GenreId is inherently distinct.