DATA202/STAT483 Assignment 3

Due: Thursday 5 September 2019, Worth 10%

Instructions

- Prepare your assignment using Rmarkdown.
- Submit your solutions in a single file named `assignment3.Rmd` through the ECS submission system
- The YAML header of your Rmarkdown file must contain your name and ID number in the author field, and should have the output format set to `pdf_document`. For example:

```
---
title: "DATA202 Assignment 3"
name: "Richard Arnold. 30XXXXXXX"
date: "25 July 2019"
output: "pdf_document"
---
```

- While you are developing your code you may find it easiest to have the output set to `html_document`, but change it to `pdf_document` when you submit.
- In your submission embed any executable R code in code chunks, and make sure both the R code and the output is displayed correctly when you knit the document.

````
```{r}
your R code
```
````

- All of your R code should execute on the server when you submit it - don't include your output in the .Rmd file (though you may include **commments** on the output).
- If there are any R code errors, then the Rmarkdown file will not knit, and no output will be created at all. So if you can't get your code to work, but want to show your attempted code, then put `eval=FALSE` in the header of the R code chunk that is failing.

````
```{r eval=FALSE}
your imperfect R code
```
````

- Where appropriate, make sure you include your comments on the output within the R markdown document.
- Make sure you check the `.pdf` file that is created when you submit, to see if it has knitted correctly with no errors.
- It can be difficult to get the automated numbering to work in Rmarkdown in a long document like this. So title each question answer with its question number as `Q1.`, `Q2.`, … etc. instead of `1.`, `2.`, … .
- Do **not** use any `setwd()` commands inside your .Rmd code: your code will be running on a different computer when you submit it, so there is no point including any references to your own folders
- Make sure you don't have any `install.packages()` commands in your code: when it executes it will execute on our server, and you don't have privilege to install packages. However all packages you need will be already installed - you just need to run the appropriate `library()` commands to load them.

- When you are developing your code make sure any external files (such as data files) that you need to access are in the same folder as your .Rmd file.
- When you check your .pdf make sure it is only a few pages long: it should **not** contain pages and pages of output which is just a whole lot rows of a dataset. Only include output that is relevant to the question.

Assignment Questions

Q1. (4 Marks)

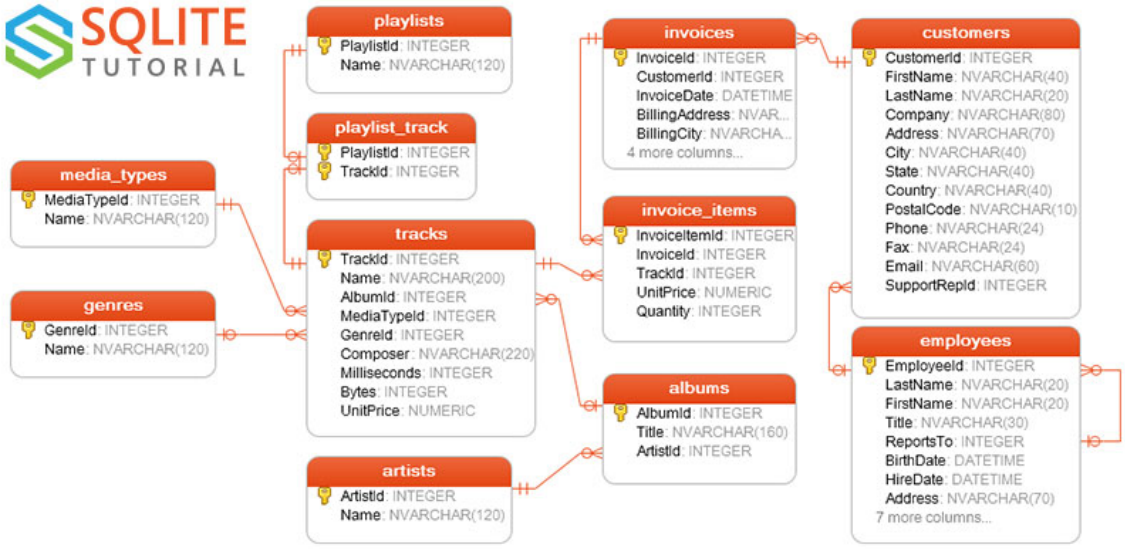
a. Write code to interpret the following strings as dates in R:

- `23-January-2001`
- `23/01/2001`
- `01/23/01`
- `day 23 of January in the year 2001`

b. What day and time was it in New Zealand when Neil Armstrong first walked on the moon?

c. Write (and execute) R code to find the day of the week it was in New Zealand when the first moonwalk happened.

Q2. (14 Marks) Download the `chinook.zip` file from the Assignments page. It is the same data provided in the SQLite tutorial http://www.sqlitetutorial.net/sqlite-sample-database/ (http://www.sqlitetutorial.net/sqlite-sample-database/) and that we used in Lab 6. A diagram of the tables in the database is given below:



Answer the following questions using SQL (use R to read the files, create a database, and load the tables into it - everything else should be in SQL). Your R and SQL code should execute in the submission system: when you submit your assignment, the chinook files will be provided for you in the **same directory** as your .Rmd code, so make sure your `read.csv()` commands refer to the files in that location.

a. In a few sentences, give a brief explanation of what Primary and Foreign Keys are and how they differ.
b. Why are Primary and Foreign Keys an important part of a database?
c. Identify the Primary and Foreign Keys (if any) in the the tables
    - `artists`
    - `invoices`
    - `albums`
    - `tracks`
d. How many different genres are there in the database?

e. Make a list of the number of tracks by genre, giving the name of each genre (as well as its identifier) and the number of tracks. Order the list in descending number of tracks.
f. Are there any genres in the database with no tracks? How do you know?
g. Calculate the percentages of tracks in each genre, and order by decreasing percentage
h. List only those genres that have ten percent or more of the tracks in the database.

Q3. (11 Marks) In this question continue using the Chinook database from the previous question.

a. In a single SQL statement, create a table call `ArtistInfo` with the contents `ArtistId`, `ArtistName`, `NumAlbums` (a count of the number of albums recorded by each artist). Make sure only Artists who have at least one album are included.
b. Delete the table `ArstistInfo`, and re-create it using a CREATE TABLE statement followed by an INSERT statement.
c. Now create a table `ArtistGenres` which for each artist counts the number of **tracks** grouped by **genre**. Do this with a CREATE+INSERT pair of statements. The table should contain the columns `ArtistId`, `ArtistName`, `GenreId` and `NumTracks`. When you create the table, place appropriate Foreign and Primary Key constraints on the table.
d. Which artists have recorded tracks in more than one genre?
e. How much money has the band (Artist) Jamiroquai made?

Q4. (11 Marks) (Answer this question in R.) The file `birthvacc.zip` contains a set of **simulated** data files on vaccination in the South Island of New Zealand. There are eight files: `birthfile.csv` contains all South Island birth records in during the period 2014-2017, `birthfile2018.csv` contains South Island births from the year 2018. `RegionCodes.csv` contains the list of NZ regions (including the 7 in the South Island). In the births tables Mother's Age has been rounded **down** to the nearest multiple of 5, except that all births to girls under the age of 15 are recorded in the 15-19 age group, and all births to mothers 50 or over are recorded in the 45-49 age group. There are five files of vaccination records, one for each year. The vaccination files record vaccination dates for children at the first three stages of the infant vaccination schedule. Note that the layout of some data files changed in 2018 - this includes names of columns.

a. Unzip the file, and load the eight files into R.
b. Convert all of the dates in all of the eight data frames from character to date format.
c. `Sex` is coded differently in 2018 birth data (Male/Female) to the other years (M/F) - and note the upper case `Sex` compared to `sex`. Change "Male" to "M" and "Female" to "F".
d. In the 2018 births data we have the region name in **text** rather than as a numerical identifier. Add the region ID to the `births2018` table using the `regions` table.
e. The column names in the `births2018` data differ from those of the births data. Rename them to match - and eliminate any columns from `births2018` which don't exist in the `births` table.
f. Combine the vaccine date data into a single data frame using `rbind()`
g. Combine the births into a single dataset with all births from 2014-2018 using `rbind()`.
h. Add a year column to the data set, and count the number of births per year.
i. Reshape the all vaccines data table from its current long format into wide format.
j. Finally merge the births and vaccines data set together. Ensure that you retain any unvaccinated children from the births data set.
k. Print out the first three lines of the combined births-vaccine data set, and print out the **dimensions** of the data set.

(Assignment total: 40 Marks)