Practical 01

Data Exploration

Data Exploration in SQLite

We will be using SQLite for this exercise; you can do this on your own machine or at

https://sqliteonline.com/.

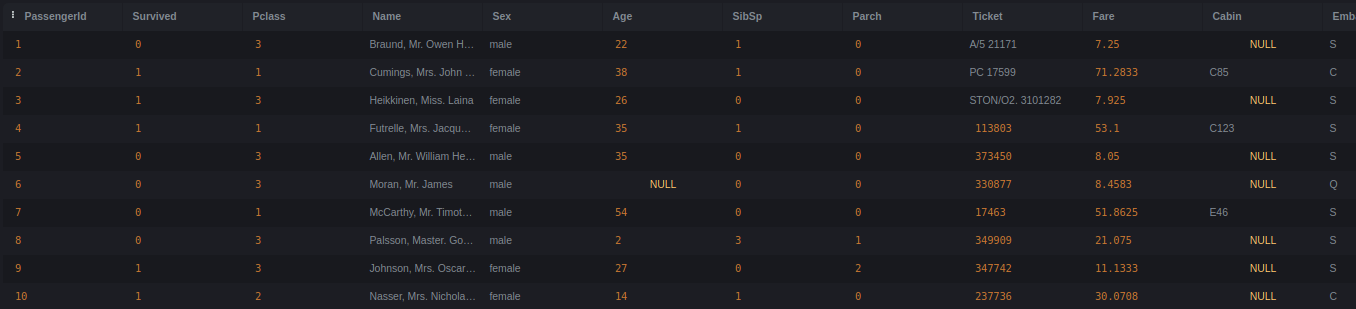
1. Download the file titanic.db file using this link:

https://github.com/PaulHancock/COMP5009\_pracs/raw/main/data/titanic.db

2. Load the database into your sqlite3 browser of choice.

3. Create a query which will SELECT all columns from the manifest table, and LIMIT the results to the 10 rows

SELECT \* FROM manifest LIMIT 10;



4. Examine each of the column names and content and determine an appropriate data type for each.

PassengerId: Numeric

Survived: Categorical/Binary -> Binary

Pclass: Categorical -> Numeric

Name: Text

Sex: Categorical

Age: Numeric

SibSp: Categorical

Parch: Categorical

Ticket: Text

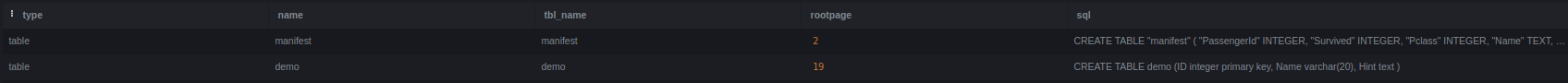
Fare: Numeric

Cabin: Text

Embarked: Categorical

5. Look at the database schema and compare the data types with those that you suggested above.

SELECT \* FROM sqlite\_schema;



CREATE TABLE "manifest" (  
"PassengerId" INTEGER,  
 "Survived" INTEGER,  
 "Pclass" INTEGER,  
 "Name" TEXT,  
 "Sex" TEXT,  
 "Age" REAL,  
 "SibSp" INTEGER,  
 "Parch" INTEGER,  
 "Ticket" TEXT,  
 "Fare" REAL,  
 "Cabin" TEXT,  
 "Embarked" TEXT  
)

6. Determine whether there are any missing values in this data set

a. Note which columns have data which IS NULL

SELECT count(\*),

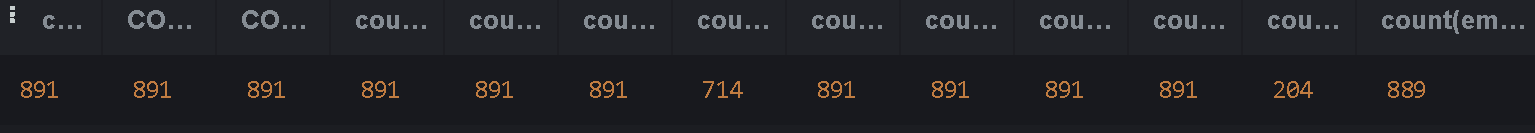
COUNT(passengerid), COUNT(survived), count(pclass),

count(name), count(sex), count(age), count(sibsp),

count(parch), count(ticket), count(fare), count(cabin),

count(embarked)

from manifest;



~~SELECT MAX(PassengerId IS NULL) PassengerId,~~

~~MAX(Survived IS NULL) Survived,~~

~~MAX(Pclass IS NULL) Pclass,~~

~~MAX(Name IS NULL) Name,~~

~~MAX(Sex IS NULL) Sex,~~

~~MAX(Age IS NULL) Age,~~

~~MAX(SibSp IS NULL) SibSp,~~

~~MAX(Parch IS NULL) Parch,~~

~~MAX(Ticket IS NULL) Ticket,~~

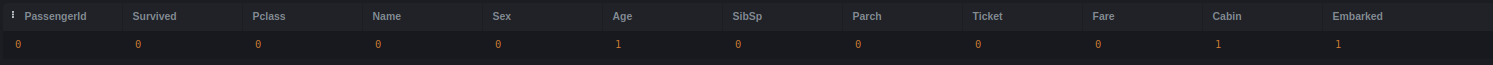
~~MAX(Fare IS NULL) Fare,~~

~~MAX(Cabin IS NULL) Cabin,~~

~~MAX(Embarked IS NULL) Embarked~~

~~FROM manifest;~~

~~Age/Cabin/Embarked = 1, others = 0~~



b. COUNT the number of entries which IS NULL and which IS NOT NULL

Total=891, Age IS NULL=177, 714, Cabin IS NULL=687, 204, Embarked IS NULL=2, 889

SELECT count(\*) FROM manifest WHERE embarked IS NULL

UNION ALL

SELECT count(\*) FROM manifest WHERE embarked IS NOT NULL;



7. For each of the numeric data columns create a query which aggregates the data to find:

SELECT MIN/MAX/AVG/SUM(age) FROM manifest

a. The minimum value

b. The maximum value

c. The average value

d. The sum of all values

SELECT MIN(age) FROM manifest

UNION ALL

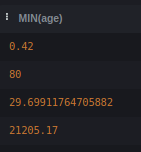
SELECT MAX(age) FROM manifest

UNION ALL

SELECT AVG(age) FROM manifest

UNION ALL

SELECT SUM(age) FROM manifest;

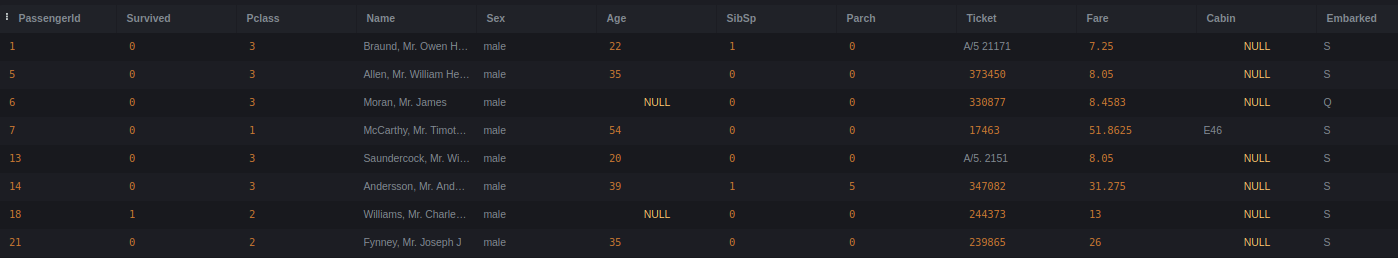


SELECT MIN(age), MAX(age), AVG(age), SUM(age) FROM manifest;

8. Create a query that will return all passengers with a name that is LIKE

“<something>Dr.<something>”

SELECT \* FROM manifest WHERE name LIKE '%Dr.%';



a. Modify this query to COUNT number of doctors and GROUP the results BY sex

SELECT sex, COUNT(\*) FROM manifest WHERE name LIKE '%Dr.%' GROUP BY sex;



9. Create a query that will return the average ticket cost

SELECT avg(fare) FROM manifest;



a. Modify this query to show this average GROUPed BY the different classes of ticket

SELECT pclass, avg(fare) FROM manifest GROUP BY pclass;



b. Further modify this query to GROUP BY the embarkation port.

SELECT pclass, embarked, avg(fare) FROM manifest GROUP BY pclass,embarked;



Remember to record your work in a logbook so that you can refer to it later in the course.

Useful references:

Software Carpentry intro to SQL: https://swcarpentry.github.io/sql-novice-survey/

Short data description: https://www.kaggle.com/c/titanic/data?select=train.csv

SQLite documentation: https://www.sqlite.org/docs.html