INFO1903 Project Part 1

Data

The datasets that I have chosen are the Australian Road Deaths Database (ARDD) which contains information about the crashes and the fatalities, and the Bureau of Meteorology's Daily Rainfall Data (Flemington Station Rainfall Data). I plan to analyse this data and attempt to find a correlation between rainfall and fatal accidents, along with other statistics regarding gender and age. The data was obtained in CSV format and then imported into a PostgreSQL database for storage and queries.

Download crash database (csv)

Download crash database legend (pdf)

Download rain database (csv)

Cleaning

For `crashes.csv`

1. Replace "Unknown" fields with null: `sed -ir 's/Unknown//g' crashes.csv`
2. Create database schema using query in appendix 1.1
3. Import csv data into database using `copy` command in appendix 1.2

For `rainfall.csv`

1. Create database schema using query in appendix 2.1
2. Import csv data into database using `copy` command in appendix 2.2

Outputting to a clean CSV

Execute the following `\copy` command from the psql prompt:

```
\copy (
  SELECT
    crashID,
    rainfall.year,
    rainfall.month,
    rainfall.day,
    crashes.crashType,
    crashes.bus,
    crashes.heavyTruck,
    crashes.articulatedTruck,
    crashes.speedLimit,
    crashes.fatalities,
```

```
    rainfall.rainfall
  FROM crashes, rainfall
 WHERE rainfall.rainfall is not null
   AND rainfall.year = crashes.year
   AND rainfall.day = crashes.day
   AND rainfall.month = extract(
     MONTH FROM to_date(concat(crashes.month, ' 2000'), 'Month YYYY')
   )
   AND crashes.state = 'VIC'
) TO '/home/lyneca/out.csv' (format csv);
```

The `extract` statement converts the month field from month name to numerical format.

Analysis

For my analysis, I counted the number of crashes when there was some rain and the number of crashes where there was no rain. I selected only the crashes in Victoria, as that is where the BOM rain data was located. The query I used was this:

```
-- Where rainfall > 0
select count(crashes.crashID)
  from crashes, rainfall
 where rainfall.year = crashes.year
   and rainfall.month = extract(
     MONTH from to_date(concat(crashes.month, ' 2000'), 'Month YYYY')
   )
   and rainfall.day = crashes.day
   and rainfall.rainfall is not null
   and rainfall.rainfall > 0
   and crashes.state = 'VIC'
;
-- Where rainfall == 0
select count(crashes.crashID)
  from crashes, rainfall
 where rainfall.year = crashes.year
   and rainfall.month = extract(
     MONTH from to_date(concat(crashes.month, ' 2000'), 'Month YYYY')
   )
   and rainfall.day = crashes.day
   and rainfall.rainfall is not null
   and rainfall.rainfall = 0
   and crashes.state = 'VIC'
;
```

My hypothesis at the time was that there would be more crashes when raining then crashes when not, but this turned out to be wrong - there were 3373 rainy crashes and 5804 sunny crashes.

After some thought, I realised that this was most likely because there are more sunny crashes than there are rainy ones, thus a more complex query is needed to discover the actual ratio. However, the query served its purpose of confirming that the database is set up for querying and data manipulation.

Appendices

Appendix 1

1. `crashes` table creation query

```
CREATE TABLE crashes (
  crashid char(13),
  state varchar(3),
  day integer,
  month varchar(10),
  year integer,
  hour integer,
  minute integer,
  crashtype varchar(16),
  bus boolean,
  heavytruck boolean,
  articulatedtruck boolean,
  speedlimit integer,
  roaduser varchar(30),
  gender varchar(6),
  age integer
);
```

2. `crashes` copy query

```
COPY crashes FROM '/path/to/crashes.csv' WITH csv header;
```

Appendix 2

1. `rainfall` table creation query

```
CREATE TABLE rainfall (
  productcode char(11),
  stationno integer,
  year integer,
  month integer,
  day integer,
  rainfall float,
  period integer,
  quality char(1)
);
```

2. `rainfall` copy query

```
COPY rainfall FROM '/path/to/rainfall.csv' WITH csv header;
```