Step 1: Saved each spreadsheet as a CSV

Notes:

* In the risk table, there was one cell with no value at all. I inserted a NULL
* In driver table, many columns have empty values for MilesToWork and DOB. I changed these cells to NULL using R

Step 2: Schema Creation [In sqlite syntax]

AGENCY TABLE

CREATE TABLE AGENCY

( id INT,

Name TEXT NOT NULL,

City TEXT NOT NULL,

State TEXT NOT NULL,

PRIMARY KEY (id))

COMPANY TABLE

CREATE TABLE COMPANY

( id INT,

UnderwritingAgencyName TEXT NOT NULL,

City TEXT NOT NULL,

State TEXT NOT NULL,

PRIMARY KEY(id))

POLICY TABLE

\*Note that Sqlite3 does not have a date data type. Hence dates are text. [Yet sqlite3 does support date manipulation with strftime]

CREATE TABLE POLICY

( number INT,

StartDate TEXT NOT NULL,

EndDate TEXT NOT NULL,

AgencyID INT NOT NULL,

CompanyID INT NOT NULL,

CancelDate TEXT,

PRIMARY KEY(number),

FOREIGN KEY (AgencyID) REFERENCES AGENCY,

FOREIGN KEY (CompanyID) REFERENCES COMPANY)

DRIVER TABLE

\*Vehicle Id comes from the risk table. In R, I do a join between the driver table and the risk table to get the vehicle id from the risk table and put it into the driver table. The join variable is Driver.ID. Driver.ID is unique in both tables. Note that there are more drivers than risks [7,940]. Hence there are 7,940 vehicle Ids that are NULL in the driver table.

CREATE TABLE DRIVER

(id INT,

Fname TEXT NOT NULL,

Lname TEXT NOT NULL,

Violations INT NOT NULL,

Accidents INT NOT NULL,

MaritalStatus TEXT NOT NULL,

Gender TEXT NOT NULL,

MilesToWork INT,

PrimaryVehicleUsage TEXT NOT NULL,

DateOfBirth TEXT,

VehicleID INT,

FOREIGN KEY (VehicleID) REFERENCES VEHICLE,

PRIMARY KEY(id))

VEHICLE TABLE

CREATE TABLE VEHICLE

(id INT,

Make TEXT NOT NULL,

Model TEXT NOT NULL,

AvgPrice INT NOT NULL,

PRIMARY KEY(id))

CLAIM TABLE

\*Assuming that claimant refers to a driver. Not 100% sure that this is true

CREATE TABLE CLAIM

(id INT,

PolicyID INT NOT NULL,

DriverID INT NOT NULL,

VehicleID INT NOT NULL,

Amount INT NOT NULL,

Description TEXT NOT NULL,

PRIMARY KEY(id),

FOREIGN KEY (PolicyID) REFERENCES POLICY,

FOREIGN KEY (DriverID) REFERENCES DRIVER,

FOREIGN KEY (VehicleID) REFERENCES VEHICLE)

LOCATION TABLE

Population Distribution and Location excel files were joined using the zip code. This was done in R, so that we don’t have to complete this join everytime we want to query population distribution information from SQL

CREATE TABLE LOCATION

( ZipCode INT NOT NULL,

id INT,

City TEXT NOT NULL,

State TEXT NOT NULL,

County TEXT NOT NULL

Population INT NOT NULL,

Percent0to15 REAL NOT NULL,

Percent15to25 REAL NOT NULL,

Percent25to40 REAL NOT NULL,

Percent50 REAL NOT NULL,

PRIMARY KEY(`id`) )

DRIVERVEH TABLE

The vehicle model year is oddly placed in the risk table. Instead of putting it here, I create a Driver-Vehicle table that associates a driver with a particular vehicle. The model year for the vehicle is stored in this table

CREATE TABLE DRIVERVEH

(

DriverID INT,

VehicleID INT,

VehicleModelYear INT,

PRIMARY KEY (DriverID,VehicleID),

FOREIGN KEY (DriverID) REFERENCES DRIVER,

FOREIGN KEY (VehicleID) REFERENCES VEHICLE

)

RISK TABLE

Model.Year has been taken out.

CREATE TABLE RISK

(

id INT,

PolicyID INT NOT NULL,

DriverID INT NOT NULL,

VehicleID INT NOT NULL,

LocationID INT,

Premium INT,

PRIMARY KEY (id),

FOREIGN KEY (PolicyID) REFERENCES POLICY,

FOREIGN KEY (DriverID) REFERENCES DRIVER,

FOREIGN KEY (VehicleID) REFERENCES VEHICLE,

FOREIGN KEY (LocationID) REFERENCES LOCATION)

Notes: In my asCSV folder:

If a sheet has an adj file, then the adj file was inserted into the DB. Otherwise, the file without the adj was inserted into the db.

In general: To load a file into the db

.headers on

.mode csv

.import <FILE> <TABLE>

\*For some reason, there was an issue with loading the DriverVeh table into the DB. So I had to manually delete the first row (which was the header) when using the above method for import

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Creating Views

* The below tables represent convenient views for querying. These tables are now stored in the database

Below is the code for these views:

**CLAIM\_RISK: Claim joined with risk**

create table claim\_risk as select RISK.\*,sum(CLAIM.Amount) as ClaimsAmount from RISK LEFT JOIN CLAIM on CLAIM.DriverID=RISK.DriverID group by RISK.DriverID;

**DRIVER\_CLAIM\_RISK: Driver joined with Claim and Risk**

create table driver\_claim\_risk as select claim\_risk.\*, DRIVER.\* from claim\_risk LEFT JOIN DRIVER on claim\_risk.DriverID=Driver.id;

select count(\*) from driver\_claim\_risk;

**LOCATION\_DRIVER\_CLAIM\_RISK: Location joined with driver joined with claim joined with risk**

create table location\_driver\_claim\_risk as select driver\_claim\_risk.\*,LOCATION.\* from driver\_claim\_risk LEFT JOIN LOCATION on LocationID=LOCATION.id;

select count(\*) from location\_driver\_claim\_risk

**LOCATION\_DRIVER\_CLAIM\_RISK: Vehicle joined with location joined with driver joined with claim joined with risk**

create table vehicle\_location\_driver\_claim\_risk as select location\_driver\_claim\_risk.\*, VEHICLE.\* from location\_driver\_claim\_risk LEFT JOIN VEHICLE on VehicleID=VEHICLE.id;

select count(\*) from vehicle\_location\_driver\_claim\_risk;

**POLICY\_COMPANY: Policy joined with company**

create table policy\_company as select POLICY.\*, COMPANY.\* from POLICY LEFT JOIN COMPANY on CompanyID=COMPANY.id;

**POLICY\_COMPANY\_AGENCY: Policy joined with company with agency**

create table policy\_company\_agency as select policy\_company.\*, AGENCY.\* from policy\_company LEFT JOIN AGENCY on policy\_company.AgencyID= AGENCY.id;

select count(\*) from policy\_company\_agency;

**ALL\_Data:**

* **Each row: A given driver, with total claims, vehicle info., policy info, location info. This is the final table.**

create table All\_DATA as select vehicle\_location\_driver\_claim\_risk.\*, policy\_company\_agency.\* from vehicle\_location\_driver\_claim\_risk LEFT JOIN policy\_company\_agency on vehicle\_location\_driver\_claim\_risk.PolicyID=policy\_company\_agency.number;

select count(\*) from All\_DATA;