CASE 1

🎯Analyzing Road Safety in the UK

Business problem:

The UK Department of Transport provides open datasets on road safety and casualties, and one can use these datasets to analyze how safe the roads in the UK are. This project will help you answer a few questions using their 2015 dataset.

The dataset has 3 tables i.e Accident, vehicle, Vehicle_type

Approach/Project Idea

Use aggregate functions in SQL and Python to answer the following sample questions:

1. Evaluate the median severity value of accidents caused by various Motorcycles.
2. Evaluate Accident Severity and Total Accidents per Vehicle Type
3. Calculate the Average Severity by vehicle type.
4. Calculate the Average Severity and Total Accidents by Motorcycle.

Solution

We will first start by creating a database and using that database as a first step.

STEP1

```
-- CREATE DATABASE NAMED INEURON_ASSIGNMENT
CREATE DATABASE INEURON_ASSIGNMENT;

-- USE DATABASE NAMED INEURON_ASSIGNMENT
USE INEURON_ASSIGNMENT;
```

Create all three tables given i.e. Accidents_2015, Vehicles_2015 and vehicle_types. To get the dataset creation format we can use Anaconda Powershell Prompt. The code which we will write is as follows:

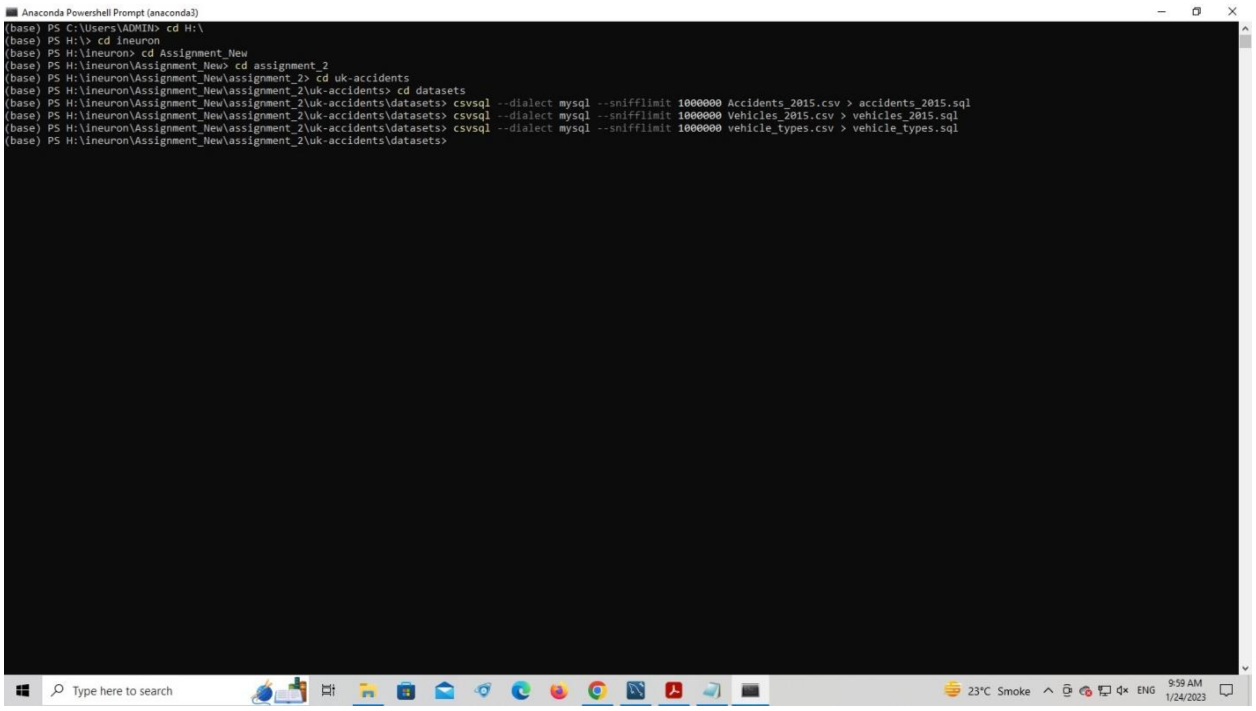
STEP 2

csvsql --dialect mysql --snifflimit Accidents_2015.csv > Accidents_2015.sql

csvsql --dialect mysql --snifflimit Vehicles_2015.csv > Vehicles_2015.sql

csvsql --dialect mysql --snifflimit vehicles_type.csv > vehicles_type.sql

Note: .sql file will be created in same directory where .csv files are located



```
(base) PS C:\Users\ADMIN> cd H:\
(base) PS H:\> cd ineuron
(base) PS H:\ineuron> cd Assignment_New
(base) PS H:\ineuron\Assignment_New> cd assignment_2
(base) PS H:\ineuron\Assignment_New\assignment_2> cd uk-accidents
(base) PS H:\ineuron\Assignment_New\assignment_2\uk-accidents> cd datasets
(base) PS H:\ineuron\Assignment_New\assignment_2\uk-accidents\datasets> csvsql --dialect mysql --snifflimit 1000000 Accidents_2015.csv > accidents_2015.sql
(base) PS H:\ineuron\Assignment_New\assignment_2\uk-accidents\datasets> csvsql --dialect mysql --snifflimit 1000000 Vehicles_2015.csv > vehicles_2015.sql
(base) PS H:\ineuron\Assignment_New\assignment_2\uk-accidents\datasets> csvsql --dialect mysql --snifflimit 1000000 vehicle_types.csv > vehicle_types.sql
(base) PS H:\ineuron\Assignment_New\assignment_2\uk-accidents\datasets>
```

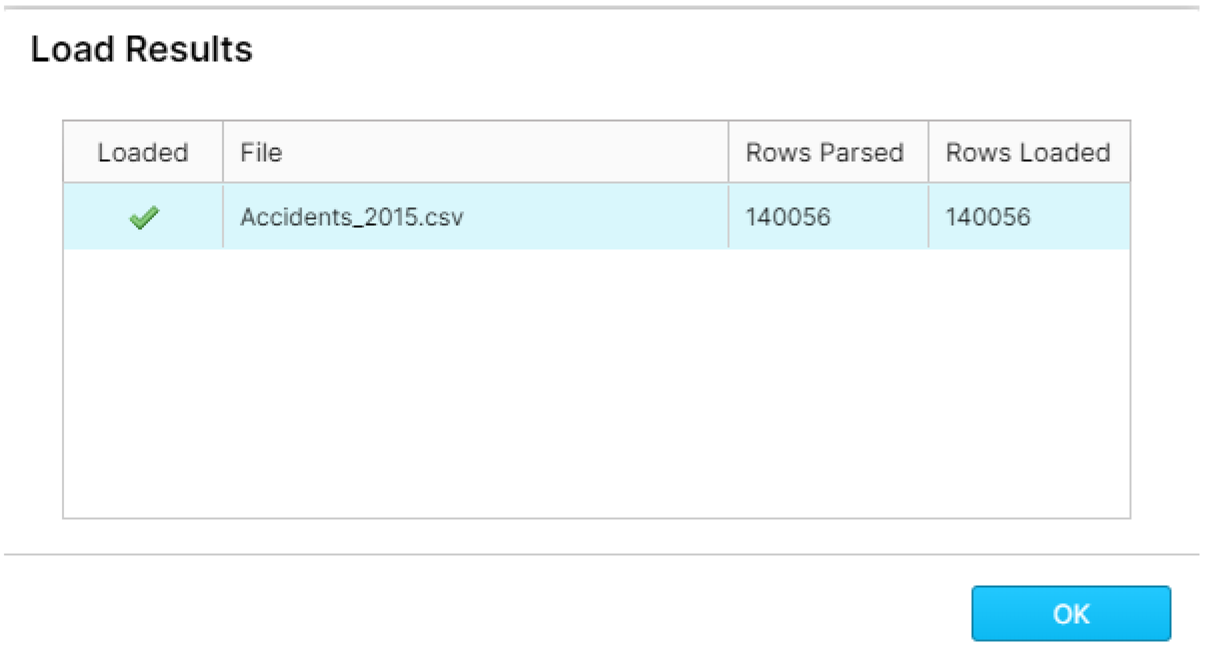
After Running the above command we will open the .sql file in Notepad ++ to see the table structure which is automatically created and copy paste the same in Snowflake.

Step 3

a. Create and Load Data in Accidents_2015 table

```
CREATE TABLE Accidents_2015 (
    Accident_Index VARCHAR(50) NOT NULL,
    Location_Easting_OSGR VARCHAR(50),
    Location_Northing_OSGR VARCHAR(50),
    Longitude VARCHAR(50),
    Latitude VARCHAR(50),
    Police_Force INTEGER NOT NULL,
    Accident_Severity INTEGER NOT NULL,
    Number_of_Vehicles INTEGER NOT NULL,
    Number_of_Casualties INTEGER NOT NULL,
    `Date` DATE NOT NULL,
    Day_of_Week INTEGER NOT NULL,
    `Time` TIME,
    Local_Authority_District INTEGER NOT NULL,
    Local_Authority_Highway VARCHAR(9) NOT NULL,
    `1st_Road_Class` INTEGER NOT NULL,
    `1st_Road_Number` INTEGER NOT NULL,
    Road_Type INTEGER NOT NULL,
    Speed_limit INTEGER NOT NULL,
    Junction_Detail INTEGER NOT NULL,
    Junction_Control INTEGER NOT NULL,
    `2nd_Road_Class` INTEGER NOT NULL,
    `2nd_Road_Number` INTEGER NOT NULL,
    Pedestrian_Crossing_Human_Control INTEGER NOT NULL,
    Pedestrian_Crossing_Physical_Facilities INTEGER NOT NULL,
    Light_Conditions INTEGER NOT NULL,
    Weather_Conditions INTEGER NOT NULL,
    Road_Surface_Conditions INTEGER NOT NULL,
    Special_Conditions_at_Site INTEGER NOT NULL,
    Carriageway_Hazards INTEGER NOT NULL,
    Urban_or_Rural_Area INTEGER NOT NULL,
    Did_Police_Officer_Attend_Scene_of_Accident INTEGER NOT NULL,
    LSOA_of_Accident_Location VARCHAR(50)
);
```

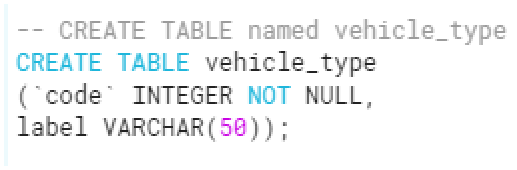
Data Loaded Successfully in snowflake for table Accidents_2015



Load Results

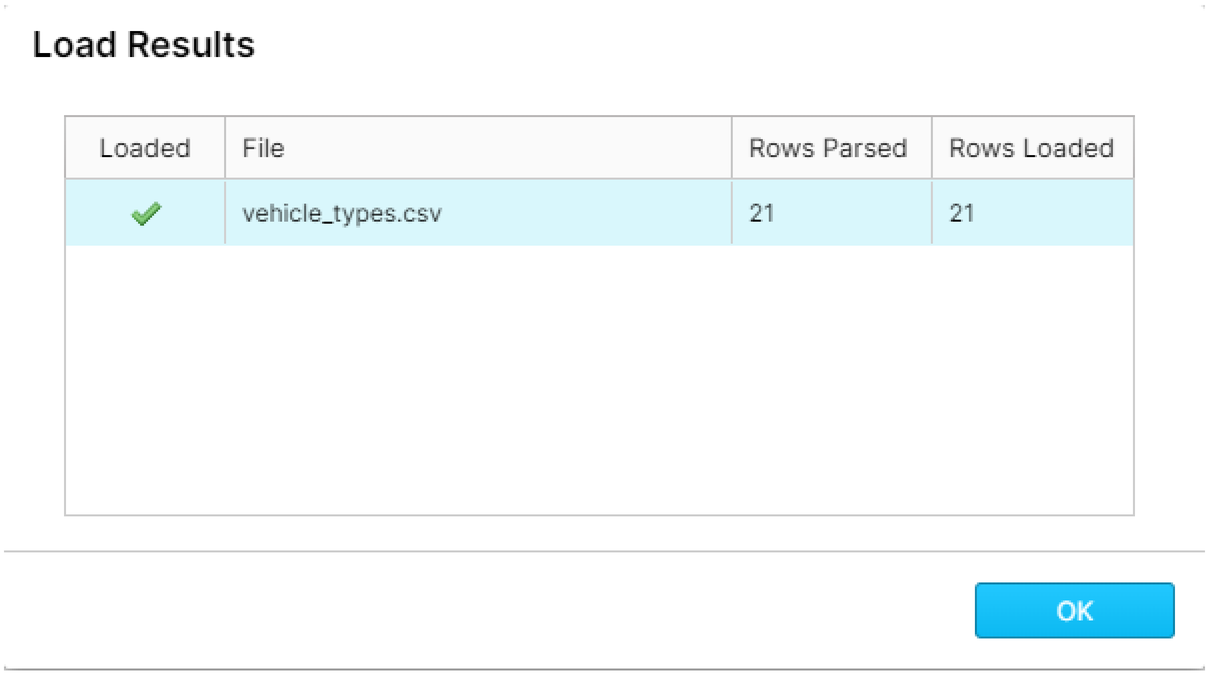
Loaded	File	Rows Parsed	Rows Loaded
✓	Accidents_2015.csv	140056	140056

b. Create and Load Data in vehicles_type table



```
-- CREATE TABLE named vehicle_type
CREATE TABLE vehicle_type
(`code` INTEGER NOT NULL,
label VARCHAR(50));
```

Data Loaded Successfully in snowflake for table vehicle_types



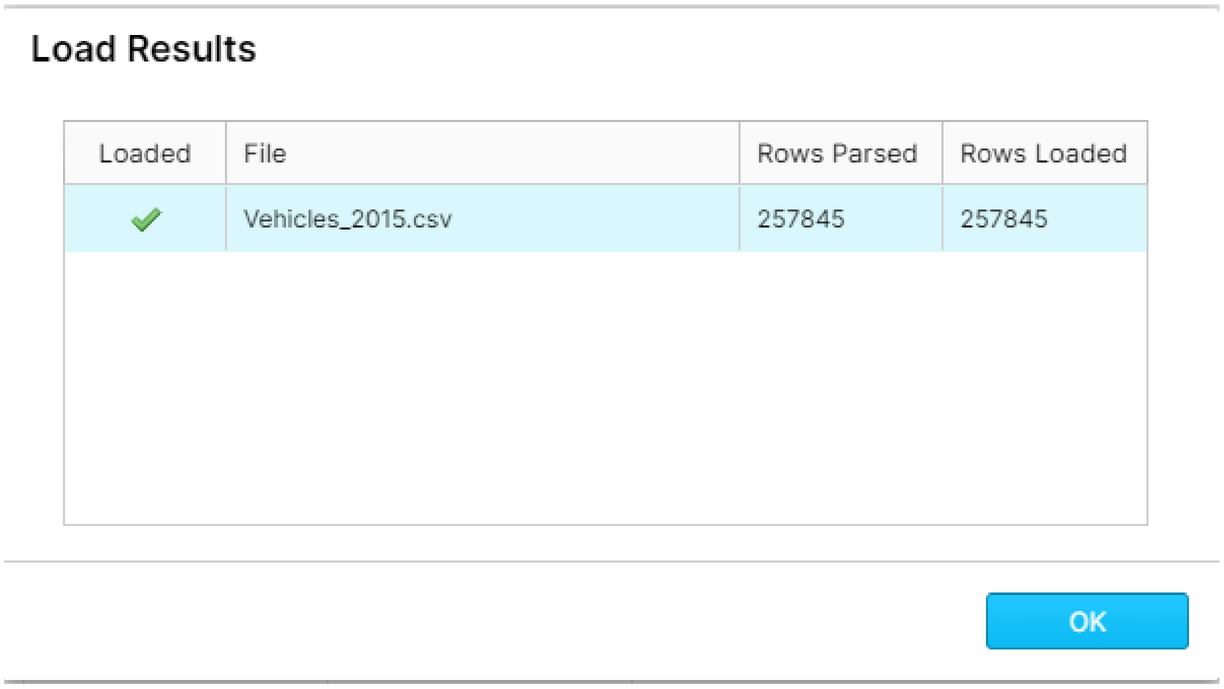
Load Results

Loaded	File	Rows Parsed	Rows Loaded
✓	vehicle_types.csv	21	21

c. Create and Load Data in Vehicles_2015 table

```
-- CREATE TABLE NAMED Vehicles_2015
CREATE TABLE Vehicles_2015 (
    Accident_Index VARCHAR(13) NOT NULL,
    Vehicle_Reference INTEGER NOT NULL,
    Vehicle_Type INTEGER NOT NULL,
    Towing_and_Articulation INTEGER NOT NULL,
    Vehicle_Manoeuvre INTEGER  NOT NULL,
    Vehicle_Location_Restricted_Lane INTEGER NOT NULL,
    Junction_Location INTEGER NOT NULL,
    Skidding_and_Overturning INTEGER NOT NULL,
    Hit_Object_in_Carriageway INTEGER NOT NULL,
    Vehicle_Leaving_Carriageway INTEGER NOT NULL,
    Hit_Object_off_Carriageway INTEGER NOT NULL,
    `1st_Point_of_Impact` INTEGER NOT NULL,
    Was_Vehicle_Left_Hand_Drive INTEGER NOT NULL,
    Journey_Purpose_of_Driver INTEGER NOT NULL,
    Sex_of_Driver INTEGER NOT NULL,
    Age_of_Driver INTEGER NOT NULL,
    Age_Band_of_Driver INTEGER NOT NULL,
    Engine_Capacity_CC INTEGER NOT NULL,
    Propulsion_Code INTEGER NOT NULL,
    Age_of_Vehicle INTEGER NOT NULL,
    Driver_IMD_Decile INTEGER NOT NULL,
    Driver_Home_Area_Type INTEGER NOT NULL,
    Vehicle_IMD_Decile INTEGER NOT NULL
);
```

Data Loaded Successfully in snowflake for table Vehicles_2015



Load Results

Loaded	File	Rows Parsed	Rows Loaded
✓	Vehicles_2015.csv	257845	257845

STEP 4

Create a Master Dataset named Master_Accidents by performing multiple INNER JOINS as we need exact matching records

```
-- LETS CREATE A MASTER DATASET BY PERFORMING INNER JOIN ON Accidents_2015,Vehicles_2015, vehicle_type
CREATE TABLE MASTER_ACCIDENTS AS
SELECT a.*, b.Vehicle_Type, c.label FROM Accidents_2015 AS a
INNER JOIN
Vehicles_2015 as b ON a.Accident_Index = b.Accident_Index
INNER JOIN vehicle_type c ON b.Vehicle_Type = c.`code`;
```

STEP 5

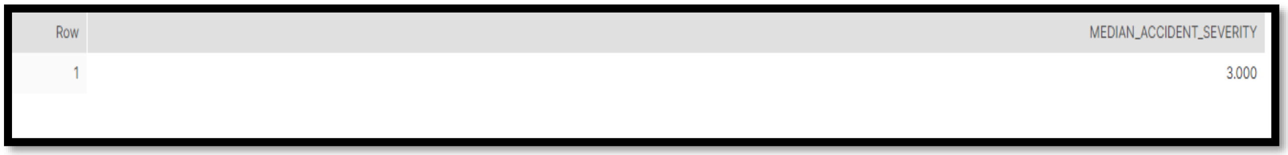
Now we are in a position to address the problem stated above in the problem statement

1. Evaluate the median severity value of accidents caused by various Motorcycles

```
-- Evaluate the median severity value of accidents caused by various Motorcycles

SELECT MEDIAN(Accident_Severity) AS MEDIAN_ACCIDENT_SEVERITY
FROM MASTER_ACCIDENTS
WHERE label LIKE '%Motorcycle%'
ORDER BY Accident_Severity;
```

Output by running above query we got:



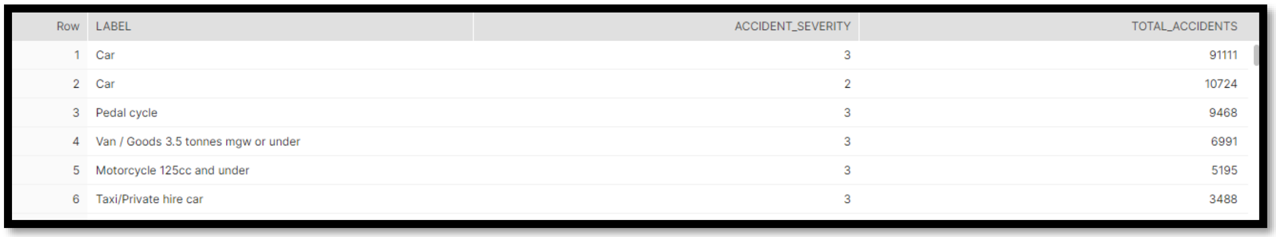
Row	MEDIAN_ACCIDENT_SEVERITY
1	3.000

median severity value of accidents caused by various Motorcycles is ***3*** which means it is ***Slight***

2. Evaluate Accident Severity and Total Accidents per Vehicle Type

```
-- Evaluate Accident Severity and Total Accidents per Vehicle Type
SELECT label, Accident_Severity,COUNT(Accident_Index) As Total_Accidents
FROM MASTER_ACCIDENTS
GROUP BY label,  Accident_Severity
ORDER BY Total_Accidents DESC;
```

Output by running above query we got:



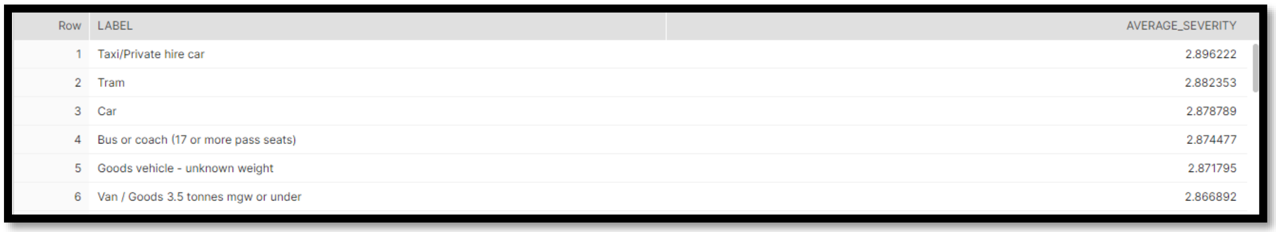
Row	LABEL	ACCIDENT_SEVERITY	TOTAL_ACCIDENTS
1	Car	3	91111
2	Car	2	10724
3	Pedal cycle	3	9468
4	Van / Goods 3.5 tonnes mgw or under	3	6991
5	Motorcycle 125cc and under	3	5195
6	Taxi/Private hire car	3	3488

It gives us a clear picture that ***maximum number of accidents*** have ***happened with car*** with a ***accident severity of 3*** which is ***Slight***

3. Calculate the Average Severity by vehicle type

```
-- Calculate the Average Severity by vehicle type
SELECT label, AVG(Accident_Severity) AS Average_Severity
FROM MASTER_ACCIDENTS
GROUP BY label
ORDER BY Average_Severity DESC;
```

Output by running above query we got:



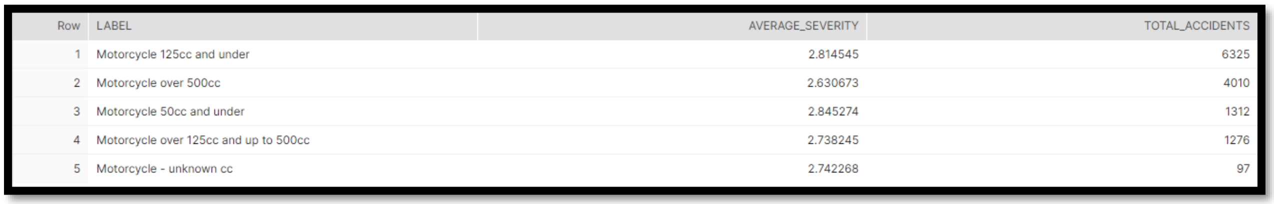
Row	LABEL	AVERAGE_SEVERITY
1	Taxi/Private hire car	2.896222
2	Tram	2.882353
3	Car	2.878789
4	Bus or coach (17 or more pass seats)	2.874477
5	Goods vehicle - unknown weight	2.871795
6	Van / Goods 3.5 tonnes mgw or under	2.866892

It gives us the ***average severity per vehicle type*** and we can see that ***max average 2.90 accident severity is with Taxi/Private hire car.***

4. Calculate the Average Severity and Total Accidents by Motorcycle

```
-- Calculate the Average Severity and Total Accidents by Motorcycle
SELECT label, AVG(Accident_Severity) AS AVERAGE_SEVERITY, COUNT(Accident_Index) AS Total_Accidents
FROM MASTER_ACCIDENTS
WHERE label LIKE '%Motorcycle%'
GROUP BY label
ORDER BY Total_Accidents DESC;
```

Output by running above query we got:



Row	LABEL	AVERAGE_SEVERITY	TOTAL_ACCIDENTS
1	Motorcycle 125cc and under	2.814545	6325
2	Motorcycle over 500cc	2.630673	4010
3	Motorcycle 50cc and under	2.845274	1312
4	Motorcycle over 125cc and up to 500cc	2.738245	1276
5	Motorcycle - unknown cc	2.742268	97

As we have taken vehicle type as ***motocycle***. ***Highest total number of accidents*** with motocycle is ***6325*** caused by ***Motorcycle 125cc*** and under while ***Average Severity*** is highest in ***Motorcycle 50cc and under*** which is **2.85.**

CASE 2

🎯Analyzing the World Population

In this project, you will use the dataset by CIA World Factbook and explore how the world population spreads across different countries.

Dataset contains columns like id, code, area, population, birth_rate, death_rate, migration_rate, population_growth.

Approach/Project Idea

You will learn how to use SQL to answer the following analytical questions:

1. Which country has the highest population?

2. Which country has the least number of people?

3. Which country is witnessing the highest population growth?

4. Which country has an extraordinary number for the population?

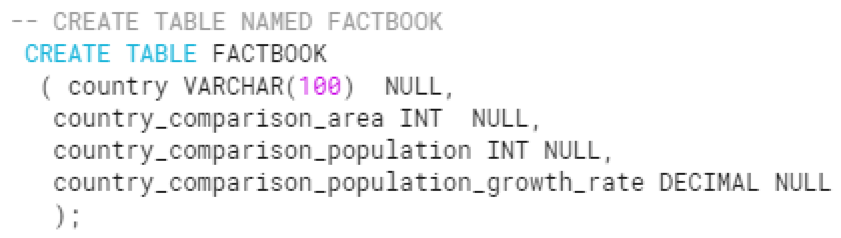
5. Which is the most densely populated country in the world?

SOLUTION

In this dataset we are given a whole lot of information in the Factbook table which is related to world population and we don't require all the columns in the csv file. So we will only select those columns which we need to answer the problem statement defined above

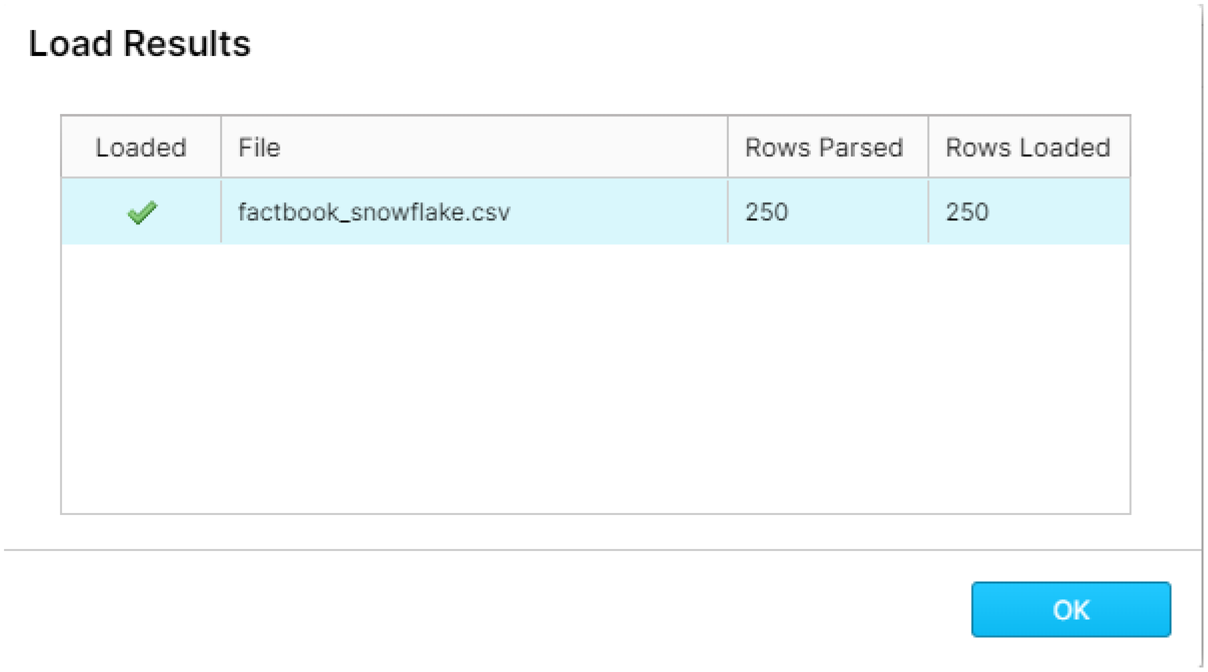
STEP 1

Create a table named Factbook



```
-- CREATE TABLE NAMED FACTBOOK
 CREATE TABLE FACTBOOK
  ( country VARCHAR(100)  NULL,
   country_comparison_area INT  NULL,
   country_comparison_population INT NULL,
   country_comparison_population_growth_rate DECIMAL NULL
   );
```

Data Loaded Successfully in snowflake for table Vehicles_2015



Load Results

Loaded	File	Rows Parsed	Rows Loaded
✓	factbook_snowflake.csv	250	250

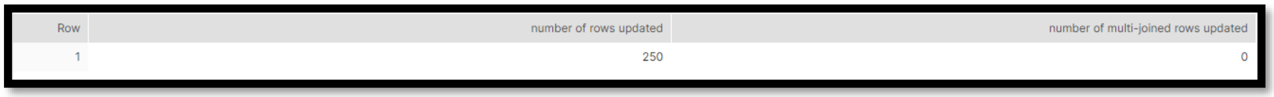
STEP 2

We need to calculate density to see which country is densely populated. For that we will create a new column named density i.e. country_comparison_population/country_comparison_area

```
-- CREATE NEW COLUMN NAMED density
 ALTER TABLE FACTBOOK
 ADD COLUMN density int NULL;

 -- UPDATE THE RECORDS IN NEW COLUMN density
 UPDATE FACTBOOK
 SET density = country_comparison_population/country_comparison_area;
```

Records are successfully updated



Row	number of rows updated	number of multi-joined rows updated
1	250	0

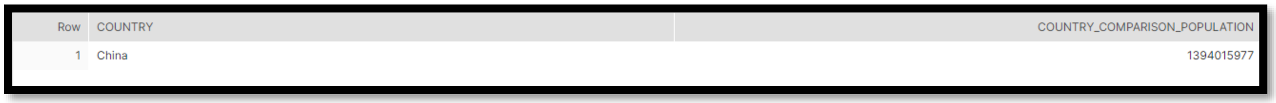
STEP 3

After the records are successfully updated we are now fully ready to give answers to the above stated problem statements:

1. Which country has the highest population ?

```
 -- Which country has the highest population
SELECT country , country_comparison_population
FROM FACTBOOK
WHERE country_comparison_population = (SELECT MAX(country_comparison_population) FROM FACTBOOK);
```

Output by running above query we got:



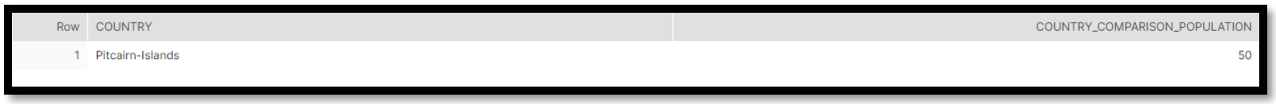
Row	COUNTRY	COUNTRY_COMPARISON_POPULATION
1	China	1394015977

China has highest population in the world i.e. ***134015977***

2. Which country has the least number of people ?

```
-- Which country has the least number of people
SELECT country, country_comparison_population
FROM FACTBOOK
WHERE country_comparison_population = (SELECT MIN(country_comparison_population) FROM FACTBOOK);
```

Output by running above query we got:



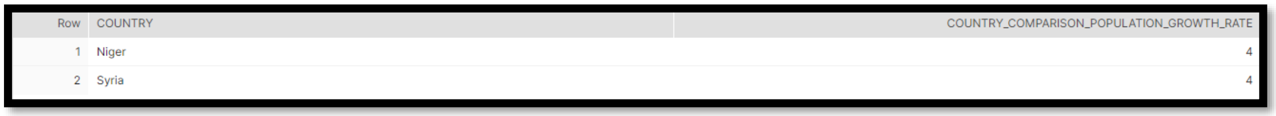
Row	COUNTRY	COUNTRY_COMPARISON_POPULATION
1	Pitcairn-Islands	50

Pitcairn-Islands has least number of people i.e. ***50***

3. Which country is witnessing the highest population growth ?

```
-- Which country is witnessing the highest population growth
SELECT country, country_comparison_population_growth_rate
FROM FACTBOOK
WHERE country_comparison_population_growth_rate = (SELECT MAX(country_comparison_population_growth_rate) FROM FACTBOOK);
```

Output by running above query we got:



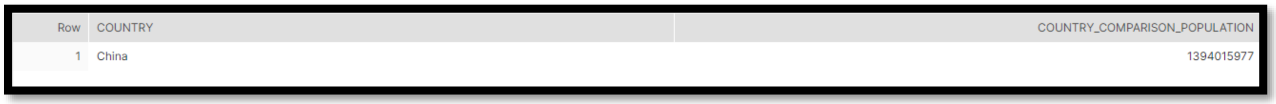
Row	COUNTRY	COUNTRY_COMPARISON_POPULATION_GROWTH_RATE
1	Niger	4
2	Syria	4

Niger and ***Syria*** are the once which has ***highest population growth*** of ***4%***

4. Which country has an extraordinary number for the population?

```
 -- Which country has an extraordinary number for the population
 SELECT country , country_comparison_population
FROM FACTBOOK
WHERE country_comparison_population = (SELECT MAX(country_comparison_population) FROM FACTBOOK);
```

Output by running above query we got:



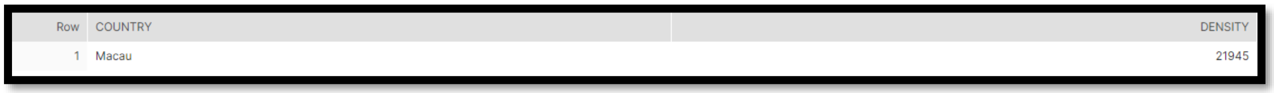
Row	COUNTRY	COUNTRY_COMPARISON_POPULATION
1	China	1394015977

China is the country which has ***extra-ordinary population*** of ***134015977 people***

5. Which is the most densely populated country in the world?

```
-- Which is the most densely populated country in the world
SELECT country, density
FROM FACTBOOK
WHERE density = (SELECT MAX(density) FROM FACTBOOK);
```

Output by running above query we got:



Row	COUNTRY	DENSITY
1	Macau	21945

Macau is most ***densely populated country*** in the world with ***density of 21945***