**iNeuron\_SQL\_Master\_Project Completed**

**A**𝗻𝗮𝗹𝘆𝘇𝗶𝗻𝗴 𝗥𝗼𝗮𝗱 𝗦𝗮𝗳𝗲𝘁𝘆 𝗶𝗻 𝘁𝗵𝗲 𝗨**K**

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.

**ANSWER :-**

create database N\_Accidents ;

use N\_Accidents ;

==================================================================================

CREATE TABLE accident(

accident\_index VARCHAR(13),

accident\_severity INT

);

CREATE TABLE vehicles(

accident\_index VARCHAR(13),

vehicle\_type VARCHAR(50)

);

CREATE TABLE vehicle\_types(

vehicle\_code INT,

vehicle\_type VARCHAR(10)

);

LOAD DATA LOCAL INFILE 'C:\\study\\INeuron data analyst\\mysql\\projects\\iNeuron\_SQL\_Master\_Project\\project\_1\\dataset\\Accidents\_2015.csv'

INTO TABLE accident

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(@col1, @dummy, @dummy, @dummy, @dummy, @dummy, @col2, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy)

SET accident\_index=@col1, accident\_severity=@col2;

Select \* from Accident

LOAD DATA LOCAL INFILE 'C:\\study\\INeuron data analyst\\mysql\\projects\\iNeuron\_SQL\_Master\_Project\\project\_1\\dataset\\Vehicles\_2015.csv'

INTO TABLE vehicles

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(@col1, @dummy, @col2, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy)

SET accident\_index=@col1, vehicle\_type=@col2;

Select \* from vehicles

LOAD DATA LOCAL INFILE 'C:\\study\\INeuron data analyst\\mysql\\projects\\iNeuron\_SQL\_Master\_Project\\project\_1\\dataset\\vehicle\_types.csv'

INTO TABLE vehicle\_types

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES;

Select \* from vehicle\_types

==================================================================================

1. **Evaluate the median severity value of accidents caused by various Motorcycles. ( MYSQL and PYTHON )**

**In SQL Create a New Table :**

CREATE TABLE accidents\_median(

vehicle\_types VARCHAR(100),

severity INT

);

**In python :-**

1. **Install required libraries**

!pip3 install pymysql

1. **Connect python to MySQL :**



============================================================================

1. **Evaluate Accident Severity and Total Accidents per Vehicle Type**

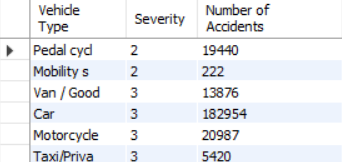
select vt.Vehicle\_Type as 'Vehicle Type', a.accident\_severity as 'Severity', COUNT(v.vehicle\_type) AS 'Number of Accidents'

from accident a

JOIN Vehicles v ON a.accident\_index = v.accident\_index

JOIN Vehicle\_Types vt ON v.vehicle\_type = vt.vehicle\_code

Group by 1

Order by 2 ;  
  
  
  
===========================================================================

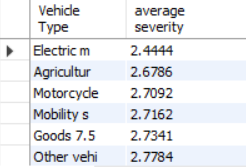
**3. Calculate the Average Severity by vehicle type.**

select vt.Vehicle\_Type as 'Vehicle Type', avg(a.accident\_severity) as 'average severity' from accident a

JOIN Vehicles v ON a.accident\_index = v.accident\_index

JOIN Vehicle\_Types vt ON v.vehicle\_type = vt.vehicle\_code

Group by 1

Order by 2 ;  
  
  
===========================================================================

**4. Calculate the Average Severity and Total Accidents by Motorcycle.**

select vt.Vehicle\_Type as 'Vehicle Type', avg(a.accident\_severity) as 'average severity' from accident a

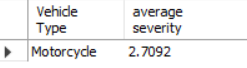
JOIN Vehicles v ON a.accident\_index = v.accident\_index

JOIN Vehicle\_Types vt ON v.vehicle\_type = vt.vehicle\_code

Group by 1

HAVING vt.Vehicle\_type = 'motorcycle'

Order by 2 ;



======================================================================================================================================================

𝗔𝗻𝗮𝗹𝘆𝘇𝗶𝗻𝗴 𝘁𝗵𝗲 𝗪𝗼𝗿𝗹𝗱 𝗣𝗼𝗽𝘂𝗹𝗮𝘁𝗶𝗼**n**

In this project, you will use the dataset by CIA World Factbook and explore how the world population spreads across different countries.  
  
The data has information from only 261 different countries. Dataset contains 11 rows and 262 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?

**(NOTE :- In the Source section** [**https://github.com/ptyadana/SQL-Data-Analysis-and-Visualization-Projects**](https://github.com/ptyadana/SQL-Data-Analysis-and-Visualization-Projects) **dataset is not found. So I’ve taken CIA population data from (source :-** [**https://www.openintro.org/data/index.php?data=cia\_factbook**](https://www.openintro.org/data/index.php?data=cia_factbook) **) (CSV). In this CSV, ‘ID’ and ‘code’ columns are not found so, I’ve replace ‘ID’ with ‘country’. The data has information from only 259 different countries)**

**ANSWER :-**

Create database CIA\_World\_Factbook ;

use CIA\_World\_Factbook ;

create table World\_Population (

Country Varchar(30),

Area decimal(38,2),

Birth\_Rate decimal(38,2),

Death\_Rate decimal(38,2),

Migration\_Rate decimal(38,2),

population int ,

population\_growth\_rate decimal(38,2)

);

LOAD DATA LOCAL INFILE 'C:\\study\\INeuron data analyst\\mysql\\projects\\iNeuron\_SQL\_Master\_Project\\project\_2\\datasets\\cia\_factbook.csv'

INTO TABLE World\_Population

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(@col1, @col2, @col3, @col4, @dummy, @dummy, @dummy, @dummy, @col5,@col6, @col7)

SET Country=@col1, Area=@col2, Birth\_Rate=@col3, Death\_Rate=@col4, Migration\_Rate=@col5,population=@col6, population\_growth\_rate=@col7;

select \* from World\_Population;

==================================================================================

**1. Which country has the highest population?**

select country , population from World\_Population

order by population desc

limit 1;

-----------OR--------------

SELECT country, population FROM World\_Population WHERE population = (SELECT MAX(population) FROM World\_Population)



==================================================================================

**2. Which country has the least number of people?**

SELECT country, population FROM World\_Population WHERE population = (SELECT MIN(population) FROM World\_Population)



==================================================================================

**3. Which country is witnessing the highest population growth?**

select country , population\_growth\_rate from World\_Population

order by population\_growth\_rate desc

limit 1;

----------------------OR---------------------

SELECT country, population\_growth\_rate FROM World\_Population WHERE population\_growth\_rate = (SELECT MAX(population\_growth\_rate) FROM World\_Population)



==================================================================================

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

select \* from world\_population

where population = (select min (population) from world\_population;

==================================================================================

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

select country, area, population, (population/area) as 'population density' from World\_Population

order by (population/area) desc

limit 1;



====================================================================================================================================================================