POSTGRE SQL QUERIES

INTERACTIVE TRAFFIC ACCIDENT SEVERITY DASHBOARD

1.Get the total number of accidents:

SELECT SUM(total\_accidents) AS total\_accidents FROM accidents\_data;

2.List all types of weather conditions recorded in accidents:

SELECT DISTINCT weather\_condition FROM accidents\_data;

3.Find the total fatal casualties in the current year:

SELECT SUM(fatal\_casualties) AS total\_fatal\_casualties FROM accidents\_data WHERE year = 2019;

4.Retrieve the number of accidents by road surface condition:

SELECT road\_surface, COUNT(\*) AS total\_accidents FROM accidents\_data GROUP BY road\_surface;

5.Display the total number of serious casualties:

SELECT SUM(serious\_casualties) AS serious\_casualties FROM accidents\_data;

6.Find the percentage of fatal casualties caused by weather conditions:

SELECT weather\_condition,

(SUM(fatal\_casualties) \* 100.0 / SUM(SUM(fatal\_casualties)) OVER()) AS percentage

FROM accidents\_data

GROUP BY weather\_condition;

7.Get the number of accidents grouped by vehicle type:

SELECT vehicle\_type, COUNT(\*) AS total\_accidents FROM accidents\_data GROUP BY vehicle\_type;

8.Find the average number of accidents by month:

SELECT EXTRACT(MONTH FROM accident\_date) AS month, AVG(total\_accidents) AS avg\_accidents

FROM accidents\_data

GROUP BY EXTRACT(MONTH FROM accident\_date);

9.Display the total casualties by road type:

SELECT road\_type, SUM(total\_casualties) AS casualties

FROM accidents\_data

GROUP BY road\_type;

10.Find the top 3 states with the highest number of fatal casualties:

SELECT state, SUM(fatal\_casualties) AS total\_fatal

FROM accidents\_data

GROUP BY state

ORDER BY total\_fatal DESC

LIMIT 3;

11.Compare the total number of accidents between the current and previous year:

SELECT year, SUM(total\_accidents) AS accidents

FROM accidents\_data

WHERE year IN (2019, 2020)

GROUP BY year;

12.Identify the road surface condition with the highest accident rate:

SELECT road\_surface, COUNT(\*) AS accident\_count

FROM accidents\_data

GROUP BY road\_surface

ORDER BY accident\_count DESC

LIMIT 1;

13.Retrieve the total accidents and casualties by month:

SELECT EXTRACT(MONTH FROM accident\_date) AS month,

SUM(total\_accidents) AS accidents,

SUM(total\_casualties) AS casualties

FROM accidents\_data

GROUP BY EXTRACT(MONTH FROM accident\_date);

14.Get the number of accidents caused by "Snow/Fog":

SELECT COUNT(\*) AS accidents\_in\_snow FROM accidents\_data WHERE weather\_condition = 'Snow/Fog';

15.Display accident details for "Single Carriageway" road type:

SELECT \* FROM accidents\_data WHERE road\_type = 'Single Carriageway';

16.Calculate the year-over-year percentage change in total accidents:

SELECT a.year,

((b.total\_accidents - a.total\_accidents) \* 100.0 / a.total\_accidents) AS yoy\_change

FROM (SELECT year, SUM(total\_accidents) AS total\_accidents FROM accidents\_data GROUP BY year) a

JOIN (SELECT year, SUM(total\_accidents) AS total\_accidents FROM accidents\_data GROUP BY year) b

ON a.year = b.year - 1;

17.Find the top 3 road types contributing to the highest percentage of fatal casualties:

SELECT road\_type,

(SUM(fatal\_casualties) \* 100.0 / SUM(SUM(fatal\_casualties)) OVER()) AS percentage

FROM accidents\_data

GROUP BY road\_type

ORDER BY percentage DESC

LIMIT 3;

18.Identify the state with the highest fatal casualty-to-accident ratio:

SELECT state,

(SUM(fatal\_casualties) \* 1.0 / SUM(total\_accidents)) AS fatality\_ratio

FROM accidents\_data

GROUP BY state

ORDER BY fatality\_ratio DESC

LIMIT 1;

19.Rank states by total serious casualties using a window function:

SELECT state, SUM(serious\_casualties) AS total\_serious,

RANK() OVER (ORDER BY SUM(serious\_casualties) DESC) AS rank

FROM accidents\_data

GROUP BY state;

20.Compare fatal casualties in wet vs dry road conditions:

SELECT road\_surface, SUM(fatal\_casualties) AS fatal\_casualties

FROM accidents\_data

WHERE road\_surface IN ('Wet', 'Dry')

GROUP BY road\_surface;

21.Find the month with the highest number of serious casualties over all years:

SELECT EXTRACT(MONTH FROM accident\_date) AS month,

SUM(serious\_casualties) AS total\_serious

FROM accidents\_data

GROUP BY EXTRACT(MONTH FROM accident\_date)

ORDER BY total\_serious DESC

LIMIT 1;

22.Determine the correlation between weather conditions and fatal casualties:

SELECT weather\_condition, AVG(fatal\_casualties) AS avg\_fatal

FROM accidents\_data

GROUP BY weather\_condition;

23.List road types where accidents increased by more than 20% year-over-year:

SELECT a.road\_type

FROM (SELECT road\_type, year, SUM(total\_accidents) AS total\_accidents FROM accidents\_data GROUP BY road\_type, year) a

JOIN (SELECT road\_type, year, SUM(total\_accidents) AS total\_accidents FROM accidents\_data GROUP BY road\_type, year) b

ON a.road\_type = b.road\_type AND a.year = b.year - 1

WHERE (b.total\_accidents - a.total\_accidents) \* 100.0 / a.total\_accidents > 20;

24.Find states with above-average fatal casualties:

WITH avg\_fatal AS (

SELECT AVG(SUM(fatal\_casualties)) OVER() AS avg\_fatal

FROM accidents\_data

)

SELECT state, SUM(fatal\_casualties) AS total\_fatal

FROM accidents\_data, avg\_fatal

WHERE SUM(fatal\_casualties) > avg\_fatal

GROUP BY state;

25.Analyze the distribution of fatal casualties by vehicle type over months:

SELECT vehicle\_type, EXTRACT(MONTH FROM accident\_date) AS month,

SUM(fatal\_casualties) AS fatal\_casualties

FROM accidents\_data

GROUP BY vehicle\_type, EXTRACT(MONTH FROM accident\_date);

THANKYOU