Data Analytic SQL Mini-Project

(Datasets Description / META DATA:)

■ The Dataset contains the following columns:

'COLUMN'	'DESCRIPTION'
1. name:	*The name or model of the car.*
2. year:	*The manufacturing year of the car.*
3. selling_price:	*The price at which the car was sold.*
4. km_driven	*The number of kilometers driven by the car.*
5. fuel:	*The type of fuel the car uses.*
6. seller_type:	*The type of seller (individual, dealer, or Trust mark dealer).*
7. transmission:	*The type of transmission (manual or automatic).*
8. owner:	*The number of previous owners of the car.*
9. mileage:	*The mileage of the car in kilometers per liter.*
10. engine [CC]:	*The engine displacement in cubic centimeters (CC).*
11. max_power:	*The maximum power output of the car's engine.*
12. seats:	*The number of seats in the car.*

MYSQL query for meaningful insights:

For reading whole table:

Using query:

```
USE shubhdb;

SELECT *
FROM car_info;
```

1. Maximum selling price:

- The query you've written will retrieve the maximum selling price from the ***car_info*** table. This SQL statement selects the highest value present in the ***selling_price*** column:

```
SELECT  MAX(selling_price) as max_sell_price
FROM car_info;
```

Output:

max_sell_price
10000000

- This query will return the single highest selling price among all the cars listed in the ***car_info*** table.

(shubham99singh@gmail.com)

2. Selling price worth 1000000

- Certainly! This query will retrieve distinct combinations of the year and selling price where the selling price is equal to 10,000,000 (10 million):

```
SELECT DISTINCT year, selling_price
FROM car_info
WHERE selling_price = 10000000;
```

- This query will display unique pairs of year and selling price where the selling price matches the specified value of 10,000,000 in the ***car_info*** table.

3. Count name from year 2017

- It seems like you're attempting to sum the ***name*** column, which typically contains text data representing the car's model or name. However, summing up textual data might not yield meaningful results. If you're looking to retrieve the total count of cars sold by name in the year 2017, you'd want to use ***COUNT*** instead of ***SUM***. Here's how you could adjust your query:

```
 SELECT name, year, COUNT(name) AS total_count
FROM car_info
WHERE year = 2017
GROUP BY name, year;
```

- This query will count the occurrences of each car model or name in the year 2017 and provide the total count for each name in that specific year.

4. Count name where total cars sold in which year

- This query aims to find the year with the highest count of cars sold from the ***car_info*** table. It selects the year and the count of cars sold in each year, orders the result by the total cars sold in descending order, and limits the output to only one result—the year with the highest count of cars sold.
- This query will return the year with the highest count of cars sold:

```
SELECT year, COUNT(name) AS total_cars_sold
FROM car_info
GROUP BY year
ORDER BY total_cars_sold DESC
LIMIT 1;
```

(shubham99singh@gmail.com)

- It will display the year and the total number of cars sold in that year, showing the year with the most significant sales volume.
-

5. MAXIMUM POWER OUTPUT BY FUEL TYPE

This query identifies the maximum power output for each fuel type. It helps in comparing the performance attributes across different fuel categories.

```
SELECT fuel, MAX(max_power) AS max_power_output
FROM car_info
GROUP BY fuel
```

6. Average selling price

- This query calculates the average selling price of cars for each year. It helps in understanding the trend of price variations over different manufacturing years.

```
 SELECT year, AVG(selling_price) AS avg_selling_price
FROM car_info
GROUP BY year;
```

7. Percentage of cars sold by the fuel type:

- This query calculates the percentage of cars sold based on the type of fuel they use. It helps in understanding the market share of different fuel types in the car sales.

```
SELECT fuel, COUNT(*) AS total_cars,
    ROUND((COUNT(*) * 100 / (SELECT COUNT(*) FROM car_info)), 2) AS percentage_sold
FROM car_info
GROUP BY fuel;
```

8. Average kilometre driven by the seller type:

- This query calculates the average kilometers driven for each seller type. It provides insights into the typical usage of cars sold by different seller categories.

```
SELECT seller_type, AVG(km_driven) AS avg_km_driven
FROM car_info
GROUP BY seller_type;
```

(shubham99singh@gmail.com)

9. Count of maximum cars sold by transmission type and years:

- This query provides the count of cars sold based on transmission type for each year. It helps in analyzing the preference for manual or automatic transmissions over different years.

```
 SELECT year, transmission, COUNT(*) AS total_cars_sold
FROM car_info
GROUP BY year, transmission
HAVING  total_cars_sold > 100
ORDER BY  total_cars_sold desc;
```

10. AVERAGE MILEAGE PER ENGINE [CC]:

- This query calculates the average mileage based on the engine displacement. It can uncover insights into the relationship between engine size and fuel efficiency.

```
SELECT engine_cc, AVG(mileage) AS avg_mileage
FROM car_info
GROUP BY engine_cc;
```

11. Most common number of seat in car:

- This query identifies the most common number of seats in cars based on the dateset. It offers insights into the prevalent seating capacity in the cars sold

```
SELECT seats, COUNT(*) AS total_cars
FROM car_info
GROUP BY seats
ORDER BY total_cars DESC
LIMIT 1;
```

These queries help extract valuable insights from the dataset, shedding light on various aspects of the car sales data such as pricing trends, fuel preferences, seller patterns, transmission preferences over time, and the relationship between engine size and mileage.

(shubham99singh@gmail.com)

(shubham99singh@gmail.com)