**-- Step 1: Create the database**

CREATE DATABASE GlobalCarDetails;

**-- Step 2: Show the database**

SHOW DATABASES;

**-- Step 3: Create the Manufacturers table(Table1)**

**#IF NOT EXISTS** is a conditional clause used in the CREATE TABLE statement.

**#It ensures** that a table is created only if it does not already exist in the database.

**#A PRIMARY KEY** is a column or a set of columns in a table that uniquely identifies each row in that table.

#Each table can have **only one PRIMARY KEY**,

**#AUTO\_INCREMENT** ensures that each new record gets a unique identifier without requiring manual input.

#This is especially useful for primary key **columns where uniqueness is essential.**

USE globalcardetails;

CREATE TABLE IF NOT EXISTS MANUFACTURERS (

ManufacturerID INT AUTO\_INCREMENT PRIMARY KEY, *-- Unique ID for each manufacturer*

Name VARCHAR(100) NOT NULL, *-- Name of the manufacturer*

Country VARCHAR(100) NOT NULL *-- Country where the manufacturer is* based

);

**-- Step 4: Create the Models table(Table2)**

USE globalcardetails;

CREATE TABLE IF NOT EXISTS Models

(

ModelID INT AUTO\_INCREMENT PRIMARY KEY,  *-- Unique ModelID for each manufacturer* ModelName VARCHAR(100) NOT NULL *-- Model Name of the car*

);

-**- Step 5: Create the Cars table(Table3)**

USE globalcardetails;

CREATE TABLE IF NOT EXISTS Cars (

CarID INT AUTO\_INCREMENT PRIMARY KEY,

ManufacturerID INT,

ModelID INT,

Year INT,

Price DECIMAL(15, 2),

**-- Define foreign key constraints to enforce referential integrity:**

/\***A FOREIGN KEY** is a critical concept in relational database design used to maintain referential integrity between tables.

Here's an overview of what a FOREIGN KEY is and its uses:\*/

FOREIGN KEY (ManufacturerID) REFERENCES Manufacturers(ManufacturerID), -- *Ensures that the value in 'ManufacturerID' must match a valid 'ManufacturerID' in the 'Manufacturers' table.*

FOREIGN KEY (ModelID) REFERENCES Models(ModelID) -- Ensures that the

value in 'ModelID' must match a valid 'ModelID' in the 'Models' table.

);

**-- Step 6: Create the Countries table(Table4)**

USE globalcardetails;

CREATE TABLE IF NOT EXISTS Countries (

CountryID INT AUTO\_INCREMENT PRIMARY KEY,

CountryName VARCHAR(100) NOT NULL

);

**-- Step 7: Create the Car\_Sales table(Table5)**

USE globalcardetails; .

CREATE TABLE IF NOT EXISTS Car\_Sales (

SaleID INT AUTO\_INCREMENT PRIMARY KEY,

CarID INT,

SaleDate DATE,

SalePrice DECIMAL(15, 2),

CountryID INT,

**-- Define foreign key constraints to enforce referential integrity:**

FOREIGN KEY (CarID) REFERENCES Cars(CarID), FOREIGN KEY (CountryID) REFERENCES Countries(CountryID) -- Ensures that the value in 'CountryID' must match a valid 'CountryID' in the 'Countries' table.

);

**-- Step 8: Insert data into Manufacturers table(Table1)**

INSERT INTO MANUFACTURERS (ManufacturerID, Name, Country)

VALUES

(101, 'Audi', 'Germany'),

(102, 'BMW', 'Germany'),

(103, 'Toyota', 'Japan'),

(104, 'Ford', 'USA'),

(105, 'Hyundai', 'South Korea'),

(106, 'Fiat', 'Italy');

DELETE FROM MANUFACTURERS

WHERE ManufacturerID IN (1, 2, 3, 1006)

OR (ManufacturerID IS NULL AND Name IS NULL AND Country IS NULL);

DELETE FROM MANUFACTURERS

WHERE ManufacturerID IS NULL

AND Name IS NULL

AND Country IS NULL;

SELECT \* FROM MANUFACTURERS;

**#Step9 Insert values into Model(table2)**

INSERT INTO Models (ModelID, ModelName) --

VALUES

(1001, 'Q5'),

(1002, 'X5'),

(1003, 'Corolla'),

(1004, 'Mustang'),

(1005, 'Verna'),

(1006, 'CR-V');

SELECT \* FROM Models;

**-- Step 10: Insert data into Cars table(table3)**

INSERT INTO Cars (CarId,ManufacturerID, ModelID, Year, Price)

VALUES (1,101, 1001, 1990, 2500000),

(2,102, 1002, 1987, 3500000),

(3,103, 1003, 1965, 5000000),

(4,104, 1004, 1999, 2200000),

(5,105, 1005, 2000, 3500000),

(6,106, 1006, 1980, 5500000);

SELECT \* FROM Cars;

**-- Step 11: Insert data into Countries table(table4)**

INSERT INTO Countries (CountryId,CountryName)

VALUES (11,'Germany'),

(12,'Germany'),

(13,'Japan'),

(14,'USA'),

(15,'South Korea'),

(16,'Italy');

DELETE FROM Countries

WHERE CountryId IN (1, 2, 3,4,5,6);

SELECT \* FROM Countries;

**-- Step 12: Insert data into Car\_Sales table(table5)**

INSERT INTO Car\_Sales (SaleID, CarId, SaleDate, SalePrice, CountryID)

VALUES (221, 1, '2024-01-15', 1900000, 11),

(222, 2,'2024-02-20', 2900000, 12),

(223, 3, '2024-03-10', 430000, 13),

(224, 4, '2024-01-25', 1700000, 14),

(225, 5, '2024-02-15', 4300000, 15);

SELECT \* FROM Car\_Sales;

**-- Query1: Basic Query - Select all cars**

SELECT \* FROM Cars;

**-- Query2: Basic Query - Find cars with price greater than $250,000**

SELECT \* FROM Cars WHERE Price > 2500000;

**-- Query3: Basic Query - Find cars with price lesser than $500,000**

SELECT \* FROM Cars WHERE Price < 5000000;

**-- Query4: Basic Query- Find only 3car sales Limit**

SELECT \* FROM car\_sales LIMIT 3;

**-- Query5: Basic Query- Alias function**

SELECT `SaleDate` AS SD FROM car\_sales;

**-- Query6: Basic Query- List all car sales, starting with the most expensive sale.**

SELECT \* FROM car\_sales

ORDER BY SalePrice DESC;

**# All Sql Queries**

**-- 1)Display all records from the Manufacturers table.**

SELECT \* FROM MANUFACTURERS;

**-- 2)Show only the Name and Country of all manufacturers.**

SELECT Name,Country FROM MANUFACTURERS;

**-- 3)Retrieve the ModelID and ModelName from the Models table**

SELECT ModelID, ModelName FROM Models;

**-- 4)Display all car models from the Cars table manufactured after the year 2000.**

SELECT \* FROM Cars

WHERE Year > 1995;

**-- 5)Find all cars with a price greater than 3,000,000**

SELECT \* FROM Cars

WHERE Price > 3000000;

**-- 6) Count the number of manufacturers in the Manufacturers table.**

SELECT COUNT(\*) FROM Manufacturers;

**-- 7) Find the maximum price of all cars in the Cars table.**

SELECT MAX(Price) FROM Cars;

**-- 8) Retrieve the minimum price from the Car\_Sales table.**

SELECT MIN(SalePrice) FROM car\_sales;

**-- 9) Display distinct countries from the Countries table.**

SELECT DISTINCT(CountryName) FROM Countries;

**-- 10) Find all car models that start with the letter 'C'.**

SELECT \* FROM Models

WHERE ModelName LIKE 'C%';

**-- 11) Retrieve all cars sold in 2024.**

SELECT \* FROM Car\_Sales

WHERE SaleDate BETWEEN '2024-01-01' AND '2024-12-31';

**-- 12) Count the number of car models by each manufacturer.**

SELECT ManufacturerID, COUNT(\*) AS ModelCount

FROM Cars GROUP BY ManufacturerID;

**-- 13) Find the total sales price of all cars sold.**

SELECT SUM(SalePrice)

FROM Car\_Sales;

**-- 14) Display all records from the Countries table where the country name is not 'Germany'.**

SELECT \* FROM Countries

WHERE CountryName = 'Germany';

**-- 15) Retrieve the first 3 manufacturers from the Manufacturers table.**

SELECT \* FROM Manufacturers LIMIT 3;

**-- 16) Display all manufacturers in alphabetical order by their name.**

SELECT \* FROM Manufacturers

ORDER BY Name ASC;

**-- 17) Find the average price of cars from the Cars table.**

SELECT AVG(Price) AS AvgPrice

FROM cars;

**-- 18) Retrieve car models that have the letter 'a' in their name**

SELECT \* FROM Models

WHERE ModelName LIKE '%a';

**-- 19) Count the number of cars for each model.**

SELECT ModelID, COUNT(\*) AS CarCount

FROM Models

GROUP BY ModelID;

**-- 20) Find the maximum year of manufacture for each manufacturer.**

SELECT ManufacturerID, MAX(Year) AS LatestYear

FROM Cars

Group By ManufacturerID;

**-- 21)Display cars with a price between 2,000,000 and 5,000,000.**

SELECT \* FROM Cars

WHERE Price BETWEEN 2000000 AND 5000000;

**-- 22) Show cars that have either a ModelID of 1001 or 1004.**

SELECT \* FROM Models

WHERE ModelID IN(1001,1004);

**-- 23) Find all sales that occurred in January 2024.**

SELECT \* FROM Car\_Sales

WHERE MONTH(SaleDate)=1 AND Year(SaleDate)=2024;

**-- 24) Retrieve the details of cars with a CarID of 1 or 2.**

SELECT \* FROM Cars

WHERE CarId IN(1,2);

**-- 25) Display all manufacturers and order them by Country in descending order.**

SELECT \* FROM Manufacturers

ORDER BY Country DESC;

**-- 26) Join the Cars table with the Manufacturers table to show car details along with the manufacturer name.**

SELECT Cars.CarID, Cars.ManufacturerID, Cars.ModelID, Cars.Year, Cars.Price,

Manufacturers.Name as ManufacturerName FROM Cars

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID;

**-- 27)Find the total number of cars manufactured by each manufacturer.**

SELECT ManufacturerID, COUNT(CarID) AS `Total number of cars`

FROM Cars

Group By ManufacturerID;

**-- 28)Calculate the total sales for each car model.**

SELECT ModelId, SUM(SalePrice) AS TotalSales

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

GROUP BY ModelID;

**-- 29) Retrieve the most recent sale date for each car.**

SELECT CarID,SaleID, MAX(SaleDate) AS LatestSaleDate

FROM Car\_Sales

GROUP BY SaleID;

**-- 30) List all cars that have not been sold.**

SELECT \* FROM Cars

WHERE CarID NOT IN

(SELECT CarID FROM Car\_Sales);

**-- 31) Find the average sale price for each country.**

SELECT ROUND(AVG(SalePrice),2) AS AvgSalePrice

FROM Car\_Sales

GROUP BY CountryID;

**-- 32) Count the number of sales for each car.**

SELECT CarID, COUNT(SaleID) As `Number Of Sales For Each Car`

FROM Car\_Sales

GROUP BY CarID;

**-- 33)Find the most expensive car sold in each country.**

SELECT CountryID,

MAX(SalePrice) AS MaxSalePrice

FROM Car\_Sales

GROUP BY CountryID;

**-- 34)Join the Car\_Sales table with Countries to show sales details along with the country name.**

SELECT Car\_Sales.SaleID, Car\_Sales.CarID, Car\_Sales.SaleDate, Car\_Sales.SalePrice, Car\_Sales.CountryID, Countries.CountryName

FROM Car\_Sales

INNER JOIN Countries ON Car\_Sales.CountryID = Countries.CountryID;

**-- 35)Retrieve all car models along with their manufacturer's name.**

SELECT Models.ModelName, Manufacturers.Name AS ManufacturerName

FROM Models

INNER JOIN Cars ON Models.ModelID = Cars.ModelID

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID;

**-- 36) Display all car sales where the sale price is higher than the car's original price.**

SELECT Car\_Sales.SaleID, Car\_Sales.SalePrice, Cars.Price

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

WHERE Car\_Sales.SalePrice > Cars.Price;

**-- 37) Find the manufacturer with the most car models.**

SELECT ManufacturerID, COUNT(ModelID) AS `Manufacturers with most car models`

FROM Cars

GROUP BY ManufacturerID

ORDER BY `Manufacturers with most car models` DESC

LIMIT 1;

**-- 38) List the car models that have been sold more than once.**

SELECT ModelID, COUNT(SaleID) AS SaleCount

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

GROUP BY ModelID

HAVING COUNT(SaleID) > 1;

**-- 39) Find all car models that were sold in Germany.**

SELECT Models.ModelName, Countries.CountryName

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

INNER JOIN Models ON Cars.ModelID = Models.ModelID

INNER JOIN Countries ON Car\_Sales.CountryID = Countries.CountryID

WHERE Countries.CountryName = 'Germany';

**-- 40) Retrieve the total revenue generated from car sales in each year.**

SELECT YEAR(SaleDate) AS SaleYear, SUM(SalePrice) AS TotalRevenue

FROM Car\_Sales

GROUP BY YEAR(SaleDate);

**-- 41) Find all manufacturers that have produced cars in the year 2000.**

SELECT DISTINCT Manufacturers.Name,Year

FROM Cars

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID

WHERE Cars.Year = 2000;

**-- 42)Count the number of cars sold for each year.**

SELECT YEAR(SaleDate) AS SaleYear, COUNT(CarID) AS TotalCarsSold

FROM Car\_Sales

GROUP BY YEAR(SaleDate);

**-- 43) Show all car models that have been sold in multiple countries.**

SELECT Models.ModelName, COUNT(DISTINCT Countries.CountryID) AS CountryCount

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

INNER JOIN Models ON Cars.ModelID = Models.ModelID

INNER JOIN Countries ON Car\_Sales.CountryID = Countries.CountryID

GROUP BY Models.ModelName

HAVING COUNT(DISTINCT Countries.CountryID) > 1;

**-- 44)List all sales with the price higher than the average price of that model.**

SELECT Car\_Sales.SaleID, Car\_Sales.SalePrice, AVG(Cars.Price) AS AveragePrice

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

GROUP BY Car\_Sales.SaleID

HAVING Car\_Sales.SalePrice > AVG(Cars.Price);

**-- 45)Retrieve the number of cars manufactured by each manufacturer after the year 2010.**

SELECT ManufacturerID,COUNT(CarID) AS TotalCars

FROM Cars

WHERE Year > 2010

GROUP BY ManufacturerID;

**-- 46)Find the number of cars sold by each manufacturer in 2024.**

SELECT Manufacturers.Name, COUNT(Car\_Sales.CarID) AS CarsSold

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID

WHERE YEAR(Car\_Sales.SaleDate) = 2024

GROUP BY Manufacturers.Name;

**-- 47) Display the highest and lowest sale prices for each manufacturer.**

SELECT Manufacturers.Name AS `Manufacture Name`, MAX(Car\_Sales.SalePrice) AS `Highest Sale Price`,

MIN(Car\_Sales.SalePrice) AS `Lowest Sale Price`

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID

GROUP BY `Manufacture Name`;

**-- 48) List all cars that have a higher sale price than the maximum price of any car manufactured by the same manufacturer.**

SELECT Car\_Sales.SaleID, Car\_Sales.SalePrice

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

WHERE Car\_Sales.SalePrice > (SELECT MAX(Price) FROM Cars WHERE ManufacturerID = Cars.ManufacturerID);

**-- 49) Retrieve all car models that have not been sold in 2024.**

SELECT Models.ModelName

FROM Models

LEFT JOIN Cars ON Models.ModelID = Cars.ModelID

LEFT JOIN Car\_Sales ON Cars.CarID = Car\_Sales.CarID AND YEAR(Car\_Sales.SaleDate) = 2024

WHERE Car\_Sales.CarID IS NULL;

**-- 50) List the top 3 car models with the highest total sales revenue.**

SELECT Models.ModelName, SUM(Car\_Sales.SalePrice) AS TotalRevenue

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

INNER JOIN Models ON Cars.ModelID = Models.ModelID

GROUP BY Models.ModelName

ORDER BY TotalRevenue DESC

LIMIT 3;

**-- 51) Find the top 5 countries with the highest number of car sales in 2024.**

SELECT Countries.CountryName, COUNT(Car\_Sales.SaleID) AS SalesCount

FROM Car\_Sales

INNER JOIN Countries ON Car\_Sales.CountryID = Countries.CountryID

WHERE YEAR(Car\_Sales.SaleDate) = 2024

GROUP BY Countries.CountryName

ORDER BY SalesCount DESC

LIMIT 5;

**-- 52) Calculate the average sale price for each manufacturer in 2024 and compare it to the average sale price across all manufacturers.**

WITH ManufacturerAvg AS (

SELECT Manufacturers.Name, AVG(Car\_Sales.SalePrice) AS AvgSalePrice

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID

WHERE YEAR(Car\_Sales.SaleDate) = 2024

GROUP BY Manufacturers.Name

)

SELECT Name, AvgSalePrice,

(SELECT AVG(Car\_Sales.SalePrice) FROM Car\_Sales WHERE YEAR(SaleDate) = 2024) AS OverallAvg

FROM ManufacturerAvg;

**-- 53) List the top 3 manufacturers by the total number of cars produced, with each manufacturer having at least one car sold in 2024.**

SELECT Manufacturers.Name, COUNT(Cars.CarID) AS TotalCarsProduced

FROM Manufacturers

INNER JOIN Cars ON Manufacturers.ManufacturerID = Cars.ManufacturerID

INNER JOIN Car\_Sales ON Cars.CarID = Car\_Sales.CarID AND YEAR(Car\_Sales.SaleDate) = 2024

GROUP BY Manufacturers.Name

HAVING COUNT(Car\_Sales.CarID) > 0

ORDER BY TotalCarsProduced DESC

LIMIT 3;

**-- 54)Identify the month in 2024 with the highest average sale price.**

SELECT MONTH(SaleDate) AS SaleMonth, AVG(SalePrice) AS AverageSalePrice

FROM Car\_Sales

WHERE YEAR(SaleDate) = 2024

GROUP BY MONTH(SaleDate)

ORDER BY AverageSalePrice DESC

LIMIT 1

**-- 55) Identify the top 3 manufacturers that have the highest percentage of cars sold in 2024 relative to their total production.**

WITH ManufacturerSales AS (

SELECT Manufacturers.ManufacturerID, COUNT(Cars.CarID) AS TotalCarsProduced,

SUM(CASE WHEN YEAR(Car\_Sales.SaleDate) = 2024 THEN 1 ELSE 0 END) AS CarsSoldIn2024

FROM Manufacturers

INNER JOIN Cars ON Manufacturers.ManufacturerID = Cars.ManufacturerID

LEFT JOIN Car\_Sales ON Cars.CarID = Car\_Sales.CarID

GROUP BY Manufacturers.ManufacturerID

)

SELECT ManufacturerID, (CarsSoldIn2024 / TotalCarsProduced) \* 100 AS SalesPercentage

FROM ManufacturerSales

ORDER BY SalesPercentage DESC

LIMIT 3;

**-- 56) Find the country that has the most significant increase in total car sales from 2023 to 2024.**

WITH Sales2023 AS (

SELECT CountryID, COUNT(SaleID) AS SalesCount2023

FROM Car\_Sales

WHERE YEAR(SaleDate) = 2023

GROUP BY CountryID

),

Sales2024 AS (

SELECT CountryID, COUNT(SaleID) AS SalesCount2024

FROM Car\_Sales

WHERE YEAR(SaleDate) = 2024

GROUP BY CountryID

)

SELECT Sales2024.CountryID, (SalesCount2024 - COALESCE(SalesCount2023, 0)) AS SalesIncrease

FROM Sales2024

LEFT JOIN Sales2023 ON Sales2024.CountryID = Sales2023.CountryID

ORDER BY SalesIncrease DESC

LIMIT 1;

**-- 57) Find the country with the highest average sale price for cars in 2024.**

SELECT Countries.CountryName, AVG(Car\_Sales.SalePrice) AS AverageSalePrice

FROM Car\_Sales

INNER JOIN Countries ON Car\_Sales.CountryID = Countries.CountryID

WHERE YEAR(Car\_Sales.SaleDate) = 2024

GROUP BY Countries.CountryName

ORDER BY AverageSalePrice DESC

LIMIT 1;

**-- 58)Retrieve the manufacturer with the highest total sales revenue in 2024.**

SELECT Manufacturers.Name AS `Manufacturer Name`, SUM(Car\_Sales.SalePrice) AS TotalRevenue

FROM Car\_Sales

INNER JOIN Cars ON Car\_Sales.CarID = Cars.CarID

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID

WHERE YEAR(Car\_Sales.SaleDate) = 2024

GROUP BY `Manufacturer Name`

ORDER BY TotalRevenue DESC

LIMIT 1;

**-- 59)List the top 5 countries with the highest average car sale price in 2024.**

SELECT Countries.CountryName, ROUND(AVG(Car\_Sales.SalePrice),2)AS AverageSalePrice

FROM Car\_Sales

INNER JOIN Countries ON Car\_Sales.CountryID = Countries.CountryID

WHERE YEAR(Car\_Sales.SaleDate) = 2024

GROUP BY Countries.CountryName

ORDER BY AverageSalePrice DESC

LIMIT 5;

**-- 60) Conactinate Query**

SELECT CONCAT(Manufacturers.Name, ' ', Models.ModelName, ' (', Cars.Year, ')') AS CarDetails

FROM Cars

INNER JOIN Manufacturers ON Cars.ManufacturerID = Manufacturers.ManufacturerID

INNER JOIN Models ON Cars.ModelID = Models.ModelID;

SELECT ROUND(SalePrice, -4) AS bucket,

COUNT(\*) AS count,

RPAD('', COUNT(\*), '\*') AS bar

FROM Car\_Sales

GROUP BY bucket

ORDER BY bucket;

SELECT \* FROM Cars;

SELECT

Year as full\_date,

DATE(Year) AS date\_only,

TIME(Year) AS time\_only,

HOUR(Year) AS hour,

MONTH(Year) AS month,

SECOND(Year) AS second,

MINUTE(Year) AS minute

FROM Cars;

**/\* Replace Query\*/**

SELECT

REPLACE(Name, 'Audi', 'Porche') AS name

FROM Manufacturers;

SELECT

CONCAT('This car ', Name, ' was manufactured in ', Country, '.') AS battle\_statement

FROM Manufacturers;

**#61) CASE Statements Query**

SELECT

CASE

WHEN SalePrice < 10000 THEN 'Below 10K'

WHEN SalePrice BETWEEN 10000 AND 50000 THEN '10K-50K'

WHEN SalePrice BETWEEN 50001 AND 100000 THEN '50K-100K'

WHEN SalePrice BETWEEN 100001 AND 200000 THEN '100K-200K'

ELSE 'Above 200K'

END AS price\_range,

COUNT(\*) AS count

FROM Car\_Sales

GROUP BY price\_range

ORDER BY count DESC;

**#62)Having Clause**

SELECT Countries.CountryName, SUM(Car\_Sales.SalePrice) AS TotalSales

FROM Car\_Sales

INNER JOIN Countries ON Car\_Sales.CountryID = Countries.CountryID

GROUP BY Countries.CountryName

HAVING SUM(Car\_Sales.SalePrice) > 50000;

#63)Window Function.

SELECT

CountryID,

SalePrice,

COUNT(\*) OVER (PARTITION BY CountryID) AS TotalSalesPerCountry,

SUM(SalePrice) OVER (PARTITION BY CountryID) AS TotalRevenuePerCountry,

ROW\_NUMBER() OVER (PARTITION BY CountryID ORDER BY SaleDate DESC) AS SaleRank

FROM

Car\_Sales;

**#Rank Functions**

SELECT CountryID, SalePrice,

RANK() OVER(PARTITION BY CountryID ORDER BY SalePrice DESC)

FROM Car\_Sales;

**#Dense Rank Functions**

SELECT CountryID, SalePrice,

DENSE\_RANK() OVER(PARTITION BY CountryID ORDER BY SalePrice DESC) AS DenseRank

FROM Car\_Sales;

**#Moving Avearge Functions**

SELECT SaleDate, SalePrice,

AVG(ROUND (SalePrice)) OVER (ORDER BY SaleDate ROWS BETWEEN 5 PRECEDING AND CURRENT ROW) AS MovingAverage

FROM Car\_Sales;

**#Regular Expression**

SELECT

SaleID,

REGEXP\_REPLACE(SaleID, '[^0-9]', '') AS NumericPart

FROM

Car\_Sales;

**#Sub Queries**

SELECT \*

FROM Car\_Sales

WHERE CountryID IN

(SELECT CountryID

FROM Car\_Sales

GROUP BY CountryID

HAVING AVG(SalePrice) > 100000);

**#View queries**

CREATE VIEW high\_priced\_cars\_by\_country AS

SELECT CountryID

FROM Car\_Sales

GROUP BY CountryID

HAVING AVG(SalePrice) > 3000000;

SELECT \* FROM high\_priced\_cars\_by\_country;

**#Stored Procedure**

SELECT \* FROM information\_schema.routines;

DELIMITER $$

CREATE PROCEDURE sp\_all\_rows\_car\_sales()

BEGIN

SELECT \* FROM Car\_Sales;

END $$

DELIMITER ;

SELECT \* FROM information\_schema.routines;

CALL sp\_all\_rows\_car\_sales();