**#Task-2**

**#Create Data Base :**

Create database ola\_db;

**#Use Data Base :**

Use ola;

**#Create Table Structure of Drivers:**

CREATE TABLE Drivers (

DriverID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Phone VARCHAR(15) NOT NULL UNIQUE,

City VARCHAR(50) NOT NULL,

VehicleType ENUM('Sedan', 'Hatchback', 'SUV') NOT NULL,

Rating DECIMAL(2, 1) CHECK (Rating BETWEEN 0 AND 5)

);

**#Insert Data Into Drivers Table :**

INSERT INTO Drivers (FirstName, LastName, Phone, City, VehicleType, Rating)

VALUES

('Ravi', 'Kumar', '9876543210', 'Mumbai', 'Sedan', 4.5),

('Anjali', 'Sharma', '8765432109', 'Delhi', 'SUV', 4.8),

('Priya', 'Singh', '7654321098', 'Bangalore', 'Hatchback', 4.2),

('Amit', 'Verma', '6543210987', 'Chennai', 'Sedan', 4.0),

('Rajesh', 'Gupta', '5432109876', 'Kolkata', 'SUV', 3.9),

('Neha', 'Mehta', '9123456780', 'Pune', 'Hatchback', 4.6),

('Suresh', 'Yadav', '9988776655', 'Hyderabad', 'Sedan', 4.3),

('Pooja', 'Reddy', '9876123456', 'Ahmedabad', 'SUV', 4.7),

('Manoj', 'Patel', '7890654321', 'Jaipur', 'Sedan', 3.8),

('Kavita', 'Chauhan', '8901234567', 'Lucknow', 'Hatchback', 4.4);

**#Create Table Structure of Riders :**

CREATE TABLE Riders (

RiderID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Phone VARCHAR(15) NOT NULL UNIQUE,

City VARCHAR(50) NOT NULL,

JoinDate DATE NOT NULL

);

**#Insert Data Into Riders Table :**

INSERT INTO Riders (FirstName, LastName, Phone, City, JoinDate)

VALUES

('Arjun', 'Kumar', '9876543210', 'Mumbai', '2023-01-15'),

('Sneha', 'Sharma', '8765432109', 'Delhi', '2023-02-10'),

('Rahul', 'Singh', '7654321098', 'Bangalore', '2023-03-05'),

('Pooja', 'Verma', '6543210987', 'Chennai', '2023-04-20'),

('Vikram', 'Gupta', '5432109876', 'Kolkata', '2023-05-15'),

('Anita', 'Mehta', '9123456780', 'Pune', '2023-06-25'),

('Siddharth', 'Yadav', '9988776655', 'Hyderabad', '2023-07-30'),

('Priyanka', 'Reddy', '9876123456', 'Ahmedabad', '2023-08-18'),

('Manish', 'Patel', '7890654321', 'Jaipur', '2023-09-12'),

('Kajal', 'Chauhan', '8901234567', 'Lucknow', '2023-10-05');

***#Create Table Structure Of Rides:***

CREATE TABLE Rides (

RideID INT PRIMARY KEY AUTO\_INCREMENT,

RiderID INT,

DriverID INT,

RideDate DATETIME NOT NULL,

PickupLocation VARCHAR(100) NOT NULL,

DropLocation VARCHAR(100) NOT NULL,

Distance DECIMAL(5, 2) NOT NULL,

Fare DECIMAL(10, 2) NOT NULL,

RideStatus ENUM('Completed', 'Cancelled', 'Ongoing') NOT NULL,

FOREIGN KEY (RiderID) REFERENCES Riders(RiderID),

FOREIGN KEY (DriverID) REFERENCES Drivers(DriverID)

);

**#Insert Data Into Rides Table :**

INSERT INTO Rides (RiderID, DriverID, RideDate, PickupLocation, DropLocation, Distance, Fare, RideStatus)

VALUES

(1, 2, '2023-11-01 10:30:00', 'Andheri, Mumbai', 'Bandra, Mumbai', 12.5, 250.00, 'Completed'),

(2, 3, '2023-11-02 14:00:00', 'Connaught Place, Delhi', 'Saket, Delhi', 15.0, 300.00, 'Completed'),

(3, 1, '2023-11-03 09:15:00', 'MG Road, Bangalore', 'Whitefield, Bangalore', 18.2, 400.00, 'Cancelled'),

(4, 6, '2023-11-04 17:45:00', 'T Nagar, Chennai', 'Adyar, Chennai', 10.0, 200.00, 'Completed'),

(5, 4, '2023-11-05 20:30:00', 'Salt Lake, Kolkata', 'Howrah, Kolkata', 8.5, 150.00, 'Ongoing'),

(6, 7, '2023-11-06 12:00:00', 'Koregaon Park, Pune', 'Hinjewadi, Pune', 20.0, 500.00, 'Completed'),

(7, 5, '2023-11-07 11:00:00', 'Banjara Hills, Hyderabad', 'Gachibowli, Hyderabad', 25.0, 600.00, 'Ongoing'),

(8, 10, '2023-11-08 08:30:00', 'SG Highway, Ahmedabad', 'Maninagar, Ahmedabad', 13.0, 275.00, 'Cancelled'),

(9, 9, '2023-11-09 15:30:00', 'C Scheme, Jaipur', 'Malviya Nagar, Jaipur', 16.0, 350.00, Ongoing),

(10, 8, '2023-11-10 19:00:00', 'Hazratganj, Lucknow', 'Gomti Nagar, Lucknow', 9.0, 180.00, 'Completed');

**#Create Table Structure Of Payments:**

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY AUTO\_INCREMENT,

RideID INT NOT NULL,

PaymentMethod ENUM('Card', 'Cash', 'Wallet') NOT NULL,

Amount DECIMAL(10, 2) NOT NULL CHECK (Amount > 0),

PaymentDate DATETIME NOT NULL,

FOREIGN KEY (RideID) REFERENCES Rides(RideID)

);

**#Insert Data Into Payments Table :**

INSERT INTO Payments (RideID, PaymentMethod, Amount, PaymentDate)

VALUES

(1, 'Card', 250.00, '2023-11-01 11:00:00'),

(2, 'Cash', 300.00, '2023-11-02 14:30:00'),

(3, 'Wallet', 400.00, '2023-11-03 09:45:00'),

(4, 'Card', 200.00, '2023-11-04 18:00:00'),

(5, 'Cash', 150.00, '2023-11-05 21:00:00'),

(6, 'Wallet', 500.00, '2023-11-06 12:30:00'),

(7, 'Card', 600.00, '2023-11-07 11:30:00'),

(8, 'Cash', 275.00, '2023-11-08 09:00:00'),

(9, 'Wallet', 350.00, '2023-11-09 16:00:00'),

(10, 'Card', 180.00, '2023-11-10 19:30:00');

***QUERIES🡺***

1. Retrieve the names and contact details of all drivers with a rating of 4.5 or higher.

🡪 SELECT FirstName, LastName, Phone, Rating

FROM Drivers

WHERE Rating >= 4.5;

1. Find the total number of rides completed by each driver.

🡪 SELECT DriverID, COUNT(RideID) AS TotalRidesCompleted FROM Rides

WHERE RideStatus = 'Completed'

GROUP BY DriverID;

1. List all riders who have never booked a ride.

🡪 SELECT RiderID, CONCAT(FirstName, LastName) AS RiderName

FROM Riders

WHERE RiderID NOT IN (SELECT RiderID FROM Rides);

1. Calculate the total earnings of each driver from completed rides.

🡪SELECT Drivers.DriverID, Drivers.FirstName, Drivers.LastName ,

SUM(Rides.Fare) AS TotalEarnings FROM Rides

INNER JOIN Drivers ON Rides.DriverID = Drivers.DriverID

WHERE Rides.RideStatus = 'Completed'

GROUP BY Drivers.DriverID, Drivers.FirstName, Drivers.LastName;

1. Retrieve the most recent ride for each rider.

🡪 SELECT RiderID, MAX(RideDate) AS MostRecentRide

FROM Rides

GROUP BY RiderID;

6. Count the number of rides taken in each city.

🡪 SELECT city, COUNT(RideId) As TotalRides from Rides JOIN Riders Using (RiderId)

GROUP by city;

7. List all rides where the distance was greater than 20 km.

🡪 SELECT RideID ,RiderID ,DriverID ,PickupLocation ,DropLocation ,RideDate,Distance

FROM Rides

WHERE Distance > 20;

8. Identify the most preferred payment method.

🡪SELECT PaymentMethod,COUNT(PaymentID) AS UsageCount

FROM Payments

GROUP BY PaymentMethod

ORDER BY UsageCount DESC

LIMIT 1;

9. Find the top 3 highest-earning drivers.

🡪 SELECT DriverID, FirstName, LastName,

SUM(Fare) AS TotalEarnings

FROM Rides

JOIN Drivers USING (DriverID)

WHERE RideStatus = 'Completed'

GROUP BY DriverID, FirstName, LastName

ORDER BY TotalEarnings DESC

LIMIT 3;

10. Retrieve details of all cancelled rides along with the rider's and driver's names.

🡪 SELECT

Rides.RideID,

concat(Riders.FirstName ," ",Riders.LastName)AS CoustmerName,

concat(Drivers.FirstName," ",Drivers.LastName) AS DriverName,

Rides.PickupLocation,

Rides.DropLocation,

Rides.RideDate,

rides.RideStatus

FROM Rides

JOIN Riders USING (RiderID)

JOIN Drivers USING (DriverID)

WHERE Rides.RideStatus = 'Cancelled';