task-2

create database task-2

CREATE TABLE Drivers (

DriverID INT PRIMARY KEY,

FirstName VARCHAR(100),

LastName VARCHAR(100),

Phone VARCHAR(20),

City VARCHAR(100),

VehicleType VARCHAR(50),

Rating DECIMAL(2, 1)

);

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

VALUES

(1, 'Ravi', 'Sharma', '9876543210', 'Delhi', 'SUV', 4.5),

(2, 'Amit', 'Patel', '9876543211', 'Mumbai', 'Sedan', 4.2),

(3, 'Vikram', 'Kumar', '9876543212', 'Bangalore', 'Hatchback', 3.9),

(4, 'Suresh', 'Reddy', '9876543213', 'Hyderabad', 'SUV', 4.7),

(5, 'Rajesh', 'Singh', '9876543214', 'Chennai', 'Sedan', 4.0),

(6, 'Praveen', 'Joshi', '9876543215', 'Kolkata', 'Hatchback', 4.3),

(7, 'Manoj', 'Mehta', '9876543216', 'Ahmedabad', 'SUV', 4.6),

(8, 'Arun', 'Shukla', '9876543217', 'Lucknow', 'Sedan', 3.8),

(9, 'Rakesh', 'Yadav', '9876543218', 'Noida', 'Hatchback', 4.4),

(10, 'Kiran', 'Nair', '9876543219', 'Pune', 'SUV', 4.1);

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

CREATE TABLE Riders (

RiderID INT PRIMARY KEY,

FirstName VARCHAR(100),

LastName VARCHAR(100),

Phone VARCHAR(20),

City VARCHAR(100),

JoinDate DATE

);

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

VALUES

(1, 'Manisha', 'Rathore', '9876543201', 'Delhi', '2023-05-15'),

(2, 'Anjali', 'Prajapti', '9876543202', 'Mumbai', '2022-03-25'),

(3, 'Maya', 'Rathore', '9876543203', 'Bangalore', '2021-11-05'),

(4, 'Mahi', 'Sharma', '9876543204', 'Hyderabad', '2024-01-10'),

(5, 'Neha', 'Kushwah', '9876543205', 'Chennai', '2023-08-20'),

(6, 'Atishay', 'Jain', '9876543206', 'Pune', '2022-06-18'),

(7, 'Sunita', 'Choudhury', '9876543207', 'Kolkata', '2023-03-12'),

(8, 'Manoj', 'Singh', '9876543208', 'Lucknow', '2024-02-05'),

(9, 'Sonia', 'Shukla', '9876543209', 'Jaipur', '2023-09-25'),

(10, 'Ajay', 'Reddy', '9876543210', 'Chandigarh', '2023-11-15');

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

CREATE TABLE Rides (

RideID INT PRIMARY KEY,

RiderID INT,

DriverID INT,

RideDate DATE,

PickupLocation VARCHAR(100),

DropLocation VARCHAR(100),

Distance DECIMAL(5, 2),

Fare DECIMAL(10, 2),

RideStatus VARCHAR(50),

FOREIGN KEY (RiderID) REFERENCES Riders(RiderID),

FOREIGN KEY (DriverID) REFERENCES Drivers(DriverID)

);

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

VALUES

(1, 1, 1, '2023-06-01', 'Connaught Place', 'IGI Airport', 20.5, 350.00, 'Completed'),

(2, 2, 2, '2023-07-10', 'Marine Drive', 'Andheri', 15.0, 250.00, 'Ongoing'),

(3, 3, 3, '2023-08-15', 'MG Road', 'Electronic City', 10.0, 150.00, 'Completed'),

(4, 4, 4, '2023-09-05', 'Banjara Hills', 'HITEC City', 12.5, 180.00, 'Cancelled'),

(5, 5, 5, '2023-12-25', 'Tidel Park', 'Parry’s Corner', 18.0, 270.00, 'Completed'),

(6, 6, 6, '2023-06-20', 'Koramangala', 'Whitefield', 22.0, 380.00, 'Completed'),

(7, 7, 7, '2023-08-25', 'Salt Lake', 'Howrah Bridge', 17.5, 290.00, 'Ongoing'),

(8, 8, 8, '2023-09-18', 'Bani Park', 'Jaipur Junction', 10.0, 160.00, 'Cancelled'),

(9, 9, 9, '2023-11-10', 'Aundh', 'Shivaji Nagar', 14.5, 240.00, 'Completed'),

(10, 10, 10, '2023-12-05', 'Sector 17', 'Sector 35', 9.0, 120.00, 'Completed');

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

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY,

RideID INT,

PaymentMethod VARCHAR(50),

Amount DECIMAL(10, 2),

PaymentDate DATE,

FOREIGN KEY (RideID) REFERENCES Rides(RideID)

);

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

VALUES

(1, 1, 'Card', 350.00, '2023-06-01'),

(2, 2, 'Cash', 250.00, '2023-07-10'),

(3, 3, 'Wallet', 150.00, '2023-08-15'),

(4, 4, 'Card', 180.00, '2023-09-05'),

(5, 5, 'Cash', 270.00, '2023-12-25'),

(6, 6, 'Card', 380.00, '2023-06-20'),

(7, 7, 'Wallet', 290.00, '2023-08-25'),

(8, 8, 'Cash', 160.00, '2023-09-18'),

(9, 9, 'Card', 240.00, '2023-11-10'),

(10, 10, 'Wallet', 120.00, '2023-12-05');

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

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

SELECT DriverID, FirstName, LastName, Phone, City, VehicleType, Rating

FROM Drivers

WHERE Rating >= 4.5;

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

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

SELECT r.DriverID, COUNT(r.RideID) AS TotalCompletedRides

FROM Rides r

WHERE r.RideStatus = 'Completed'

GROUP BY r.DriverID;

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

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

SELECT r.RiderID, r.FirstName, r.LastName, r.Phone, r.City

FROM Riders r

LEFT JOIN Rides ri ON r.RiderID = ri.RiderID

WHERE ri.RideID IS NULL;

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

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

SELECT r.DriverID, SUM(p.Amount) AS TotalEarnings

FROM Rides r

JOIN Payments p ON r.RideID = p.RideID

WHERE r.RideStatus = 'Completed'

GROUP BY r.DriverID;

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

5. Retrieve the most recent ride for each rider.

SELECT r.RiderID, r.RideID, r.RideDate, r.PickupLocation, r.DropLocation, r.Distance, r.Fare, r.RideStatus

FROM Rides r

INNER JOIN (

SELECT RiderID, MAX(RideDate) AS MostRecentRide

FROM Rides

GROUP BY RiderID

) recent\_rides ON r.RiderID = recent\_rides.RiderID AND r.RideDate = recent\_rides.MostRecentRide;

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

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

SELECT r.City, COUNT(ri.RideID) AS TotalRides

FROM Riders r

JOIN Rides ri ON r.RiderID = ri.RiderID

GROUP BY r.City;

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

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

SELECT RideID, RiderID, DriverID, RideDate, PickupLocation, DropLocation, Distance, Fare, RideStatus

FROM Rides

WHERE Distance > 20;

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

8. Identify the most preferred payment method.

SELECT PaymentMethod, COUNT(PaymentMethod) AS PaymentCount

FROM Payments

GROUP BY PaymentMethod

ORDER BY PaymentCount DESC

LIMIT 1;

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

9. Find the top 3 highest-earning drivers.

SELECT r.DriverID, SUM(p.Amount) AS TotalEarnings

FROM Rides r

JOIN Payments p ON r.RideID = p.RideID

WHERE r.RideStatus = 'Completed'

GROUP BY r.DriverID

ORDER BY TotalEarnings DESC

LIMIT 3;

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

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

SELECT r.RideID, r.RideDate, r.PickupLocation, r.DropLocation, r.Distance, r.Fare, r.RideStatus,

ri.FirstName AS RiderFirstName, ri.LastName AS RiderLastName,

d.FirstName AS DriverFirstName, d.LastName AS DriverLastName

FROM Rides r

JOIN Riders ri ON r.RiderID = ri.RiderID

JOIN Drivers d ON r.DriverID = d.DriverID

WHERE r.RideStatus = 'Cancelled';