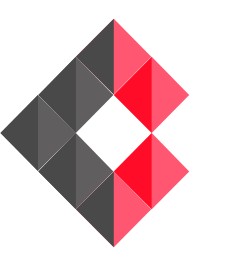
**Internship Task** - **RDBMS and SQL Task #2**



SQL case-based assignment with a scenario and 10 queries for an application like **Ola** (a ride-hailing service). **Scenario**

You are managing a database for a ride-hailing application like Ola. The database includes the following tables:

1. **Drivers** o DriverID (Primary Key)
   * FirstName o LastName o Phone
   * City
   * VehicleType (e.g., 'Sedan', 'Hatchback', 'SUV') o Rating (out of 5)
2. **Riders** o RiderID (Primary Key)
   * FirstName o LastName o Phone o City o JoinDate
3. **Rides** o RideID (Primary Key) o RiderID (Foreign Key) o DriverID (Foreign Key)
   * RideDate o PickupLocation o DropLocation o Distance (in km) o Fare
   * RideStatus (e.g., 'Completed', 'Cancelled', 'Ongoing')
4. **Payments** o PaymentID (Primary Key)
   * RideID (Foreign Key)
   * PaymentMethod (e.g., 'Card', 'Cash', 'Wallet')
   * Amount o PaymentDate

# Assignment Queries

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

**Ans :**

SELECT FirstName, LastName, Phone

FROM Drivers

WHERE Rating >= 4.5;

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

**Ans :**

SELECT driver.FirstName, driver.LastName, COUNT(ride.RideID) AS RidesCompleted

FROM Drivers driver

LEFT JOIN Rides ride ON driver.DriverID = ride.DriverID

WHERE ride.RideStatus = 'Completed'

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

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

**Ans :**

SELECT Riders.FirstName, Riders.LastName

FROM Riders

WHERE Riders.RiderID NOT IN (SELECT DISTINCT Rides.RiderID FROM Rides);

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

**Ans :**

SELECT driver.FirstName, driver.LastName, SUM(ride.Fare) AS TotalEarnings

FROM Drivers driver

JOIN Rides ride ON driver.DriverID = ride.DriverID

WHERE ride.RideStatus = 'Completed'

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

1. **Retrieve the most recent ride for each rider.**

**Ans :**

SELECT ride1.RiderID, ride1.RideDate

FROM Rides ride1

WHERE ride1.RideDate = (SELECT MAX(ride2.RideDate) FROM Rides ride2 WHERE ride2.RiderID = ride1.RiderID);

1. **Count the number of rides taken in each city.**

**Ans :**

SELECT City, COUNT(RideID) AS RidesTaken

FROM Rides

GROUP BY PickupLocation;

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

**Ans :**

SELECT RideID, PickupLocation, DropLocation, Distance

FROM Rides

WHERE Distance > 20;

1. **Identify the most preferred payment method.**

**Ans :**

SELECT PaymentMethod, COUNT(PaymentID) AS PaymentCount

FROM Payments

GROUP BY PaymentMethod

ORDER BY PaymentCount DESC

LIMIT 1;

1. **Find the top 3 highest-earning drivers.**

**Ans :**

SELECT driver.FirstName, driver.LastName, SUM(ride.Fare) AS TotalEarnings

FROM Drivers driver

JOIN Rides ride ON driver.DriverID = ride.DriverID

WHERE ride.RideStatus = 'Completed'

GROUP BY driver.DriverID, driver.FirstName, driver.LastName

ORDER BY TotalEarnings DESC

LIMIT 3;

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

**Ans :**

SELECT ride.RideID, ride.RideDate, ride.PickupLocation, ride.DropLocation, riderInformation.FirstName AS RiderFirstName, riderInformation.LastName AS RiderLastName, driverInformation.FirstName AS DriverFirstName, driverInformation.LastName AS DriverLastName

FROM Rides ride

JOIN Riders riderInformation ON ride.RiderID = riderInformation.RiderID

JOIN Drivers driverInformation ON ride.DriverID = driverInformation.DriverID

WHERE ride.RideStatus = 'Cancelled';