**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).**

CREATE DATABASE ola;

CREATE TABLE drivers(

driver\_id int PRIMARY key,

first\_name varchar(100),

last\_name varchar(100),

phone VARCHAR(20),

city VARCHAR(100),

VehicleType VARCHAR(100),

rating int

);

INSERT INTO drivers (driver\_id, first\_name, last\_name, phone, city, VehicleType, rating)

VALUES

(1, 'Puja', 'Singh', '843210', 'Indore', 'Electric', 5),

(2, 'Arun', 'Gupta', '8732109', 'Bhopal', 'Eledctric', 4),

(3, 'Yesh', 'Rajput', '76321098', 'Panna', 'Electric', 4),

(4, ''Sneha, 'Gupta', '65410987', 'Raipur', 'Electric', 5),

(5, 'Aman', 'Kumar', '54109876', 'Satna', 'Electric', 3),

(6, 'Vikram', 'Rao', '4321098765', 'Pune', 'Electric', 4),

(7, 'Sunita', 'Mishra', '3210987654', 'Kolkata', 'Electric', 5),

(8, 'Deepak', 'Yadav', '2109876543', 'Ahmedabad', 'Electric', 3),

(9, 'Meena', 'Jain', '1098765432', 'Jaipur', 'Electric', 4),

(10, 'Nikhil', 'Patel', '9988776655', 'Lucknow', 'Electric', 4);

CREATE TABLE riders(

rider\_id int PRIMARY key,

first\_name varchar(100),

last\_name varchar(100),

phone VARCHAR(20),

city VARCHAR(100),

join\_date date

);

INSERT INTO riders (rider\_id, first\_name, last\_name, phone, city, join\_date)

VALUES

(1, 'Karan', 'Mehta', '9876541230', 'Delhi', '2023-01-15'),

(2, 'Sneha', 'Kapoor', '8765432129', 'Mumbai', '2023-03-10'),

(3, 'Ajay', 'Shah', '7654321019', 'Bangalore', '2023-05-20'),

(4, 'Divya', 'Reddy', '6543210118', 'Chennai', '2023-07-25'),

(5, 'Rahul', 'Saxena', '5432101217', 'Hyderabad', '2023-09-05'),

(6, 'Neha', 'Chauhan', '4321011316', 'Pune', '2023-11-10'),

(7, 'Vivek', 'Agarwal', '3210911415', 'Kolkata', '2023-12-01'),

(8, 'Pooja', 'Roy', '2109811514', 'Ahmedabad', '2023-06-18'),

(9, 'Alok', 'Sinha', '1098711613', 'Jaipur', '2023-08-30'),

(10, 'Simran', 'Bhatia', '9988611712', 'Lucknow', '2023-10-22');

CREATE TABLE rides(

ride\_id int PRIMARY key,

rider\_id int,

FOREIGN key (rider\_id) REFERENCES riders(rider\_id),

driver\_id int,

FOREIGN key (driver\_id ) REFERENCES drivers(driver\_id ),

ride\_date date,

pickupLocation varchar(100),

dropLocation varchar(100),

distance DECIMAL(10, 2),

Fare DECIMAL(10, 2)

RideStatus varchar(70)

);

INSERT INTO rides (ride\_id, rider\_id, driver\_id, ride\_date, pickupLocation, dropLocation, distance, Fare,RideStatus)

VALUES

(1, 1, 1, '2024-01-01', 'Connaught Place', 'Indira Gandhi Intl Airport', 15.50, 350.00,'Completed'),

(2, 2, 2, '2024-01-02', 'Bandra', 'Gateway of India', 18.00, 400.00,'Cancelled'),

(3, 3, 3, '2024-01-03', 'MG Road', 'Electronic City', 25.00, 600.00,'Completed'),

(4, 4, 4, '2024-01-04', 'T Nagar', 'Marina Beach', 12.00, 250.00,'Cancelled'),

(5, 5, 5, '2024-01-05', 'Hitech City', 'Charminar', 20.00, 500.00,'Completed'),

(6, 6, 6, '2024-01-06', 'Koregaon Park', 'Shaniwar Wada', 8.00, 200.00,'Ongoing'),

(7, 7, 7, '2024-01-07', 'Salt Lake', 'Howrah Bridge', 10.00, 300.00,'Ongoing'),

(8, 8, 8, '2024-01-08', 'Navrangpura', 'Sabarmati Ashram', 7.50, 150.00,'Cancelled'),

(9, 9, 9, '2024-01-09', 'Malviya Nagar', 'Amber Fort', 15.00, 350.00,'Ongoing'),

(10, 10, 10, '2024-01-10', 'Hazratganj', 'Bara Imambara', 10.00, 250.00,'Cancelled');

CREATE TABLE payments(

payment\_iD int Primary Key,

ride\_id int,

FOREIGN key (ride\_id) REFERENCES rides(ride\_id),

paymentMethod VARCHAR(50),

Amount DECIMAL(10, 2),

PaymentDate date

);

INSERT INTO payments (payment\_iD, ride\_id, paymentMethod, Amount, PaymentDate)

VALUES

(1, 1, 'Credit Card', 350.00, '2024-01-01'),

(2, 2, 'UPI', 400.00, '2024-01-02'),

(3, 3, 'Cash', 600.00, '2024-01-03'),

(4, 4, 'Credit Card', 250.00, '2024-01-04'),

(5, 5, 'Debit Card', 500.00, '2024-01-05'),

(6, 6, 'UPI', 200.00, '2024-01-06'),

(7, 7, 'Cash', 300.00, '2024-01-07'),

(8, 8, 'UPI', 150.00, '2024-01-08'),

(9, 9, 'Credit Card', 350.00, '2024-01-09'),

(10, 10, 'Debit Card', 250.00, '2024-01-10');

---------------------------------------------------------------------------------------------------------

**#Assignment Queries#**

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

Ans=

SELECT first\_name, last\_name, phone, rating FROM drivers where rating>=4.5

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

Ans=

SELECT driver\_id, COUNT(ride\_id) as totalRides from rides where driver\_id in(SELECT driver\_id FROM drivers)GROUP by driver\_id

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

Ans=

SELECT first\_name, last\_name FROM riders where rider\_id not in(SELECT rider\_id from rides)

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

Ans=

SELECT d.driver\_id ,d.first\_name, d.last\_name, sum(Amount) as totalEarnings from drivers as d left join rides as r on d.driver\_id=r.driver\_id left join payments as p on r.ride\_id=p.ride\_id where d.driver\_id in( SELECT r.driver\_id from rides) GROUP by d.driver\_id, d.first\_name, d.last\_name

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

Ans=

SELECT

r.rider\_id,

r1.first\_name,

r1.last\_name,

r.ride\_date AS mostRecentRide

FROM

rides AS r

JOIN

riders AS r1 ON r.rider\_id = r1.rider\_id

WHERE

r.ride\_date = (

SELECT MAX(ride\_date)

FROM rides AS r2

WHERE r2.rider\_id = r.rider\_id

);

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

Ans=

select d.driver\_id, d.city, count(r.ride\_id)as noOfrides from drivers as d left JOIN rides as r on d.driver\_id=r.driver\_id GROUP by d.driver\_id, d.city

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

Ans=

select ride\_id FROM rides where distance>20

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

Ans=

SELECT

paymentMethod,

COUNT(paymentMethod) AS usageCount

FROM

payments

GROUP BY

paymentMethod

ORDER BY

usageCount DESC

LIMIT 1;

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

**Ans=**

SELECT

d.first\_name,

d.last\_name,

SUM(p.amount) AS total\_earnings

FROM

drivers AS d

LEFT JOIN

rides AS r

ON

d.driver\_id = r.driver\_id

LEFT JOIN

payments AS p

ON

r.ride\_id = p.ride\_id

GROUP BY

d.first\_name, d.last\_name

ORDER BY

total\_earnings DESC

LIMIT 3;

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

Ans=

SELECT r.ride\_id,r.ride\_date,r.pickupLocation ,r.dropLocation, r.distance ,r.Fare,r.RideStatus,d.first\_name,d.last\_name, r1.first\_name,r1.last\_name FROM rides as r join riders as r1 on r.rider\_id=r1.rider\_id join drivers as d on r.driver\_id=d.driver\_id WHERE r.RideStatus='cancelled' GROUP by r.ride\_id