

SQL QUERIES and their Description

1. This SQL query selects all the vehicle details that belong to Haryana and orders them by fuel type, specifically in the order of "Diesel", "EV", "Petrol", and "CNG".

```
SELECT * FROM rapid_cab_new.vehicle WHERE Number_Plate LIKE 'HR%'  
ORDER BY Fuel;
```

2. This SQL query adds a column called "status" in the ride_request table that is associated with a trip, specifically present in the trip table. It then sets the status of each ride_request record that is also present in the trip table to "A".

```
ALTER TABLE ride_request ADD status CHAR(1) DEFAULT "U";
```

```
UPDATE ride_request SET status = "A" WHERE R_ID IN (SELECT R_ID FROM trip);
```

```
SELECT * FROM ride_request;
```

3. This SQL query selects the customer IDs with the total amount spent by all the customers. It makes use of aggregator functions like SUM and Group By operations.

```
SELECT Customer_ID, SUM(Amount) AS total_amount FROM payment GROUP BY  
Customer_ID;
```

4. This SQL query updates the fare price to 0 of all the trips where the driver status is "B". It makes use of nested queries to update the required tuples.

```
UPDATE Trip SET Fare_Price = 0 WHERE Driver_ID IN (SELECT Driver_ID FROM  
Driver WHERE Driver_Status = 'B');
```

5. This SQL query increases the amount of the payment where the pickup time is after 4pm. It makes use of a JOIN operation using the Payment and Ride_Request tables.

```
SET SQL_SAFE_UPDATES = 0;
```

**UPDATE Payment p JOIN Ride_Request r ON p.Customer_ID = r.Customer_ID SET
p.Amount = p.Amount + 1000 WHERE r.pickup_time > '16:00:00';**

6. This SQL query deletes the payment and trip records of drivers whose status is “O”. It makes use of double JOIN.

**DELETE p, t FROM payment p JOIN trip t ON p.T_ID = t.T_ID JOIN Driver d ON
t.Driver_ID = d.Driver_ID WHERE d.Driver_Status = 'O';**

7. This SQL query selects the payment details for all trips whose drivers are available. It makes use of a JOIN operation using the Payment, Trip, and Driver tables.

**SELECT * FROM payment p JOIN Trip t ON p.T_ID = t.T_ID JOIN Driver d ON
t.Driver_ID = d.Driver_ID WHERE d.Driver_Status = 'A';**

8. This SQL query selects all the customers who have had a trip in all fuel kinds. It makes use of the Relational Division operation using a WHERE NOT EXISTS subquery.

**SELECT c.Customer_ID, c.Name, c.Email FROM Customer c WHERE NOT EXISTS (
SELECT v.Fuel FROM Vehicle v WHERE NOT EXISTS (
SELECT t.Customer_ID FROM Trip t WHERE t.Customer_ID = c.Customer_ID AND
t.Driver_ID = v.Driver_ID)
);**

9. This SQL query selects the driver and fuel details of the vehicle with fuel = “CNG”. It makes use of a LEFT JOIN operation using the Driver and Vehicle tables.

**SELECT d.Driver_ID, d.Name, v.Fuel FROM Driver d LEFT JOIN Vehicle v ON
d.Driver_ID = v.Driver_ID WHERE v.Fuel = 'CNG' OR v.Fuel IS NULL;**

10. This SQL query selects the driver and fuel details of the vehicle with Maintenance_State = B. It makes use of an INNER JOIN operation using the Driver and Vehicle tables.

**SELECT d.Name, v.Number_Plate, v.Fuel FROM Driver d INNER JOIN Vehicle v ON
d.Driver_ID = v.Driver_ID WHERE v.Maintenance_State = 'B';**