Triggers(3)

- DELIMITER $$
  CREATE TRIGGER customer_wallet_updation
  BEFORE UPDATE ON customer
  FOR EACH ROW
  BEGIN
    IF NEW.type = 'elite' AND OLD.type = 'normal' THEN
        SET NEW.wallet = OLD.wallet - 5000;
    END IF;
  END $$
  DELIMITER ;

  UPDATE customer SET type ='elite' WHERE Customer_ID=50012;

  select wallet from customer where Customer_ID=50012;
  Drop TRIGGER customer_wallet_updation;

Explanation: An amount of rupees 5000 is subtracted from the customer wallet if he buys the elite customer plan.

- DELIMITER //
  CREATE TRIGGER check_gender
  before insert on driver
  for each row
  begin
  if new.Gender is null then
  set new.Gender = 'U';
  end if ;
  end //
  DELIMITER ;

  INSERT INTO
  driver(Driver_ID,Email_ID,Name,Phone_Number,Gender,Address,Avg_Rating,Driver_St
  atus) VALUES(10301,"cmeo0@hellouardian.com","Corny_1
  Meo",8010544451,NULL,"557 Ilene new Way",1.9,"B");

  select Gender from driver where Driver_ID=10301;

Explanation: When a driver entry is inserted in the database with Gender value NULL then , the above trigger check_gender will add the value of the corresponding gender as ‘U’ for unknown.

- DELIMITER $$
  CREATE TRIGGER vehicle_insertion
  after insert on driver
  for each row
  begin
  DECLARE mystring varchar(12);
  select concat("HR",floor(rand()*100),"BF",floor(rand()*10000)) into mystring;
  INSERT INTO
  vehicle(Driver_ID,Number_Plate,Seats_accomadation,Fuel,Color,Maintainance_state)
  VALUES(new.Driver_ID,mystring,6,"EV","White","G");
  end $$
  DELIMITER $$

  INSERT INTO
  driver(Driver_ID,Email_ID,Name,Phone_Number,Gender,Address,Avg_Rating,Driver_St
  atus) VALUES(10301,"dokennavain1s@t3imenline.co.u","Debbie
  O'Kennavain",8378131094,"F","99 Butterfield Alley",4.54,"O");

  select * from vehicle where Driver_ID=10301;

The above trigger vehicle_insertion will insert an entry in the vehicle table for a corresponding insertion of a driver entry in the driver table.

OLAP Query:

- WITH table1 (amnt, t_id, pickup_time, date) AS (
    SELECT
      p.Amount, p.T_ID, r.Pickup_Time, r.Date FROM payment p
    JOIN ride_request r ON p.Customer_ID = r.Customer_ID
)
  SELECT
    pickup_time,
    date,
    SUM(amnt) AS total_amount
  FROM table1
  GROUP BY date,pickup_time with rollup;

- With table2(no_seats, fuel_type , color , amount) as(
  select v.Seats_accomadation, v.Fuel, v.Color , p.amount
  from vehicle v , payment p , trip t where p.T_ID=t.T_ID and v.Driver_ID=t.Driver_ID)

  select no_seats,fuel_type , color , sum(amount) from table2
  group by no_seats, fuel_type , color with rollup;

CUBE:
(Not working in sql)

```
SELECT t.T_ID, c.Customer_ID, AVG(p.Amount) AS AvgAmount, AVG(t.time_taken) AS
AvgTime, AVG(Credit_Score) AS AvgCreditScore
        FROM customer AS c
        JOIN payment AS p ON c.Customer_ID = p.Customer_ID
        JOIN trip AS t ON p.T_ID = t.T_ID
        GROUP BY CUBE(t.T_ID, c.Customer_ID)
        ORDER BY t.T_ID, c.Customer_ID;
```

slicing

- With table3(no_seats, fuel_type , color , amount, distance) as(
  select v.Seats_accomadation, v.Fuel, v.Color , p.amount, ABS(t.Pickup_x -
  t.Destination_x) + ABS(t.Pickup_y - t.Destination_y)
  from vehicle v , payment p , trip t where p.T_ID=t.T_ID and v.Driver_ID=t.Driver_ID)

  select no_seats,sum(amount) from table3
  where fuel_type="Petrol" and color ="Black" and distance > 100 group by no_seats;

PIVOT

- select fuel ,
  sum(case when fuel='Diesel' then 1 else 0 end) as Diesel_vehicle,
  sum(case when fuel='CNG' then 1 else 0 end) as CNG_vehicle,
  sum(case when fuel='EV' then 1 else 0 end) as EV_vehicle,
  sum(case when fuel='Petrol' then 1 else 0 end) as Petrol_vehicle
  from vehicle
  group by fuel;

Embedded Query

```
import mysql.connector as sql
```

```
def main():
    db = sql.connect(host="localhost",user = "root", password =
"Madhur@02", database = "Rapid_Cab_new")
    print(db)
    cur = db.cursor()
    dict = {
        1:"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'",
        2:"SELECT * FROM rapid_cab_new.vehicle where Number_Plate Like
'HR%' order by Fuel",
        3:"select Customer_ID ,SUM(Amount) as total_amount from payment
Group BY Customer_ID",
        4:"SELECT d.Name, v.Number_Plate, v.Fuel FROM Driver d INNER JOIN
Vehicle v ON d.Driver_ID = v.Driver_ID WHERE v.Maintainance_state = 'B'"
    }
    while(True):
        lol = int(input("Enter the query number: "))
        cur.execute(dict[lol])
        for i in cur:
            print(i)
        print("Do you want to continue? (y/n)")
        ch = input()
        if ch == 'n':
            break
    db.close()
if __name__ == "__main__":
    main()
```