

SQL-QUERIES(AUTOMOBILE-DBMS-PROJECT)

QUERIES:

1) SELECT ALL VEHICLE NAME WHERE VEHICLE TYPE IS BIKE

```
SELECT V_NAME  
FROM VEHICLE  
WHERE V_TYPE = 'BIKE';
```

2) LIST ALL CUSTOMERS NAME WHERE ORDER VEHICLE ON DATE 16-JAN- 2016.

```
SELECT C.CU_NAME  
FROM CUSTOMER C  
JOIN SALES S ON C.CU_ID = S.CU_ID  
WHERE S.ORD_DAT = TO_DATE('16-JAN-2016', 'DD-MON-YYYY');
```

3) LIST VEHICLE NAME WHICH HAS HIGH COST AMONG ALL VEHICLES.

```
SELECT V_NAME  
FROM VEHICLE  
WHERE PRICE = (SELECT MAX(PRICE) FROM VEHICLE);
```

4) SELECT TAX DETAILS OF VEHICLE WHERE VEHICLE ID IS 11.

```
SELECT T.*  
FROM TAX T  
JOIN SALES S ON T.TAX_ID = S.TAX_ID  
WHERE S.V_ID = 11;
```

5) LIST ALL SHOWROOM NAME THAT ARE COMES UNDER DEALER ID=104.

```
SELECT SH_NAME  
FROM SHOWROOM  
WHERE D_ID = 104;
```

TRIGGERS:

1) WRITE A INSERTION TRIGGER TO CHECK DATE OF ORDER MUST BE LESS THAN CURRENT DATE.

```
CREATE OR REPLACE TRIGGER check_order_date  
BEFORE INSERT ON SALES
```

```

FOR EACH ROW
DECLARE
    current_date_check DATE;
BEGIN
    current_date_check := SYSDATE;
    IF :NEW.ORD_DAT >= current_date_check THEN
        RAISE_APPLICATION_ERROR(-20001, 'Order date must be less than the current date');
    END IF;
END;
/

```

2) Write a trigger to car should greater than 100000 and bike cost should be less than 100000.

```

CREATE OR REPLACE TRIGGER check_vehicle_price
BEFORE INSERT OR UPDATE ON VEHICLE
FOR EACH ROW
DECLARE
    car_price_threshold NUMBER := 100000;
    bike_price_threshold NUMBER := 100000;
BEGIN
    IF :NEW.V_TYPE = 'CAR' AND :NEW.PRICE <= car_price_threshold THEN
        RAISE_APPLICATION_ERROR(-20001, 'Car price must be greater than ' ||
car_price_threshold);
    ELSIF :NEW.V_TYPE = 'BIKE' AND :NEW.PRICE >= bike_price_threshold THEN
        RAISE_APPLICATION_ERROR(-20002, 'Bike price must be less than ' ||
bike_price_threshold);
    END IF;
END;
/

```