SET-7

CUSTOMER (cid, fname, lname, city, country, phone)

CREATE TABLE CUSTOMER

(

cid NUMBER PRIMARY KEY,

fname VARCHAR2(20),

lname VARCHAR2(20),

city VARCHAR2(20),

country VARCHAR2(20),

phone VARCHAR2(15)

);

ORDER (oid, oDate, oNumber, cid, oTotalAmount)

CREATE TABLE ORDERS

(

oid NUMBER PRIMARY KEY,

cid NUMBER,

oDate DATE,

oTotalAmount NUMBER,

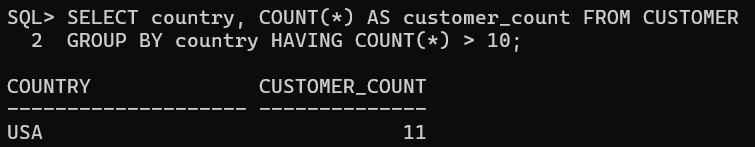
FOREIGN KEY (cid) REFERENCES CUSTOMER(cid)

);

1.List the number of customers in each country. Only include countries with more than 100 customers.

SELECT country, COUNT(\*) AS customer\_count FROM CUSTOMER

GROUP BY country HAVING COUNT(\*) > 10;



2.List the number of customers in each country, except China, sorted high to low. Only include countries with 5 or more customers.

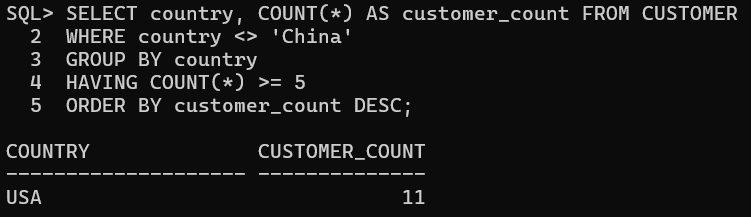
SELECT country, COUNT(\*) AS customer\_count FROM CUSTOMER

WHERE country <> 'China'

GROUP BY country

HAVING COUNT(\*) >= 5

ORDER BY customer\_count DESC;



3.List all customers with average orders between Rs.5000 and Rs.6500.

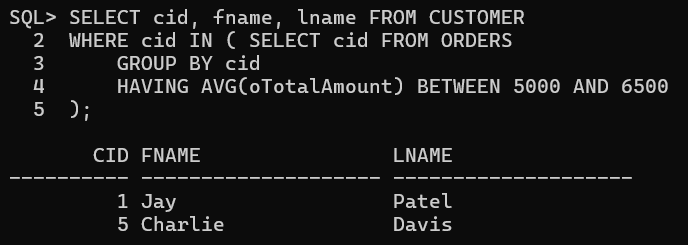
SELECT cid, fname, lname FROM CUSTOMER

WHERE cid IN ( SELECT cid FROM ORDERS

GROUP BY cid

HAVING AVG(oTotalAmount) BETWEEN 5000 AND 6500

);



4.Create a trigger that executes whenever country is updated in CUSTOMER table.

CREATE OR REPLACE TRIGGER country\_update\_trigger

AFTER UPDATE OF country ON CUSTOMER

FOR EACH ROW

BEGIN

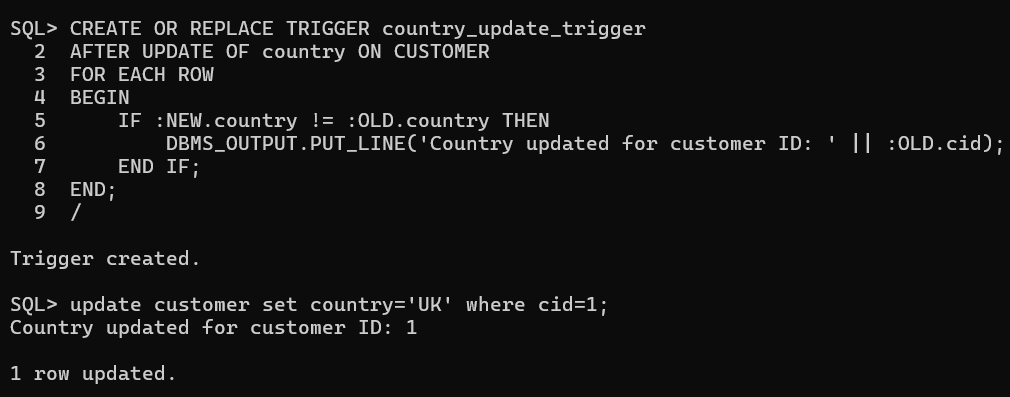
IF :NEW.country != :OLD.country THEN

DBMS\_OUTPUT.PUT\_LINE('Country updated for customer ID: ' || :OLD.cid);

END IF;

END;

/



5.Create a function to return customer with maximum orders.

**Declaration:**

CREATE OR REPLACE FUNCTION get\_customer\_with\_max\_orders

RETURN VARCHAR2 IS

customer\_name VARCHAR2(100);

BEGIN

SELECT fname || ' ' || lname INTO customer\_name

FROM CUSTOMER

WHERE cid = (

SELECT cid

FROM (

SELECT cid, COUNT(\*) AS order\_count FROM ORDERS

GROUP BY cid

ORDER BY order\_count DESC

)

WHERE ROWNUM = 1

);

RETURN customer\_name;

END;

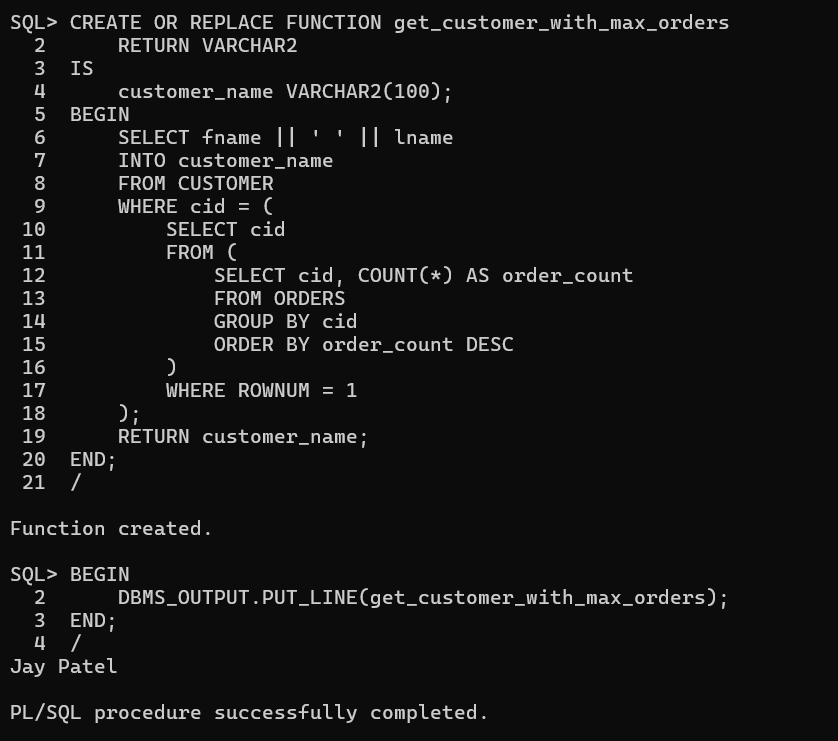
**Function Call:**

BEGIN

DBMS\_OUTPUT.PUT\_LINE(get\_customer\_with\_max\_orders);

END;

/



6.Create a procedure to display month names of dates of ORDER table. The month names should be unique.

**Procedure Create:**

CREATE PROCEDURE display\_unique\_month IS

CURSOR month\_cursor IS

SELECT DISTINCT TO\_CHAR(oDate, 'Month') AS month\_name

FROM ORDERS;

BEGIN

FOR month\_record IN month\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Month: ' || month\_record.month\_name);

END LOOP;

END;

**Call Procedure:**

BEGIN

display\_unique\_month;

END;

