**CODING SNIPPET:**

-- 1. Write a simple PL/SQL (Anonymous block) program to generate multiplication table for a given number.

DECLARE

I NUMBER:=2;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('- - - - - - - - - -');

WHILE (I<=4) LOOP

DBMS\_OUTPUT.PUT\_LINE(I||' MULTIPLICATION TABLE');

DBMS\_OUTPUT.PUT\_LINE('- - - - - - - - - -');

FOR J IN 1 .. 10 LOOP

DBMS\_OUTPUT.PUT\_LINE(I||'\*'||J||'='||(I\*J));

END LOOP;

I:=I+1;

DBMS\_OUTPUT.PUT\_LINE('- - - - - - - - - -');

END LOOP;

END;

/

--2. Write a PL/SQL code to print first 50 whole numbers. Also insert the list in temp table which is created with only one column of number datatype.

CREATE TABLE TEMP(WHOLENUMBER NUMBER(2));

BEGIN

FOR LOOPCOUNT IN 1..50 LOOP

INSERT INTO TEMP VALUES(LOOPCOUNT);

END LOOP;

END;

/

SELECT \* FROM TEMP;

-- 4. Create a trigger for the employee table, which makes the entry in ENAME column in uppercase.

CREATE TABLE EMPLOYEES

(EMP\_ID NUMBER(4) PRIMARY KEY,

EMP\_NAME VARCHAR(25),

EMAIL VARCHAR(20),

PHONE\_NUMBER VARCHAR(10),

COMM DECIMAL(8,2),

SALARY DECIMAL(8,2)

);

INSERT INTO EMPLOYEES VALUES(101,’Babu’,’[babu101@gmail.com](mailto:babu101@gmail.com)’,9878987898,1000,8000.00);

INSERT INTO EMPLOYEES VALUES(102,’Somu’,’somu102@gmail.com’,9448534120 ,100,NULL);

INSERT INTO EMPLOYEES VALUES(103,’Ramu’,’ramu103@gmail.com’,9448886634 ,300, 19000.00);

INSERT INTO EMPLOYEES VALUES (104,’Gobu’,’gobu104@gmail.com’,9447783412 ,600,NULL);

INSERT INTO EMPLOYEES VALUES(105,’Anbu’,’anbu105@gmail.com’,9848341200 ,2300,53000.00);

CREATE TRIGGER EMP\_NAME

BEFORE UPDATE OR INSERT ON EMPLOYEES

FOR EACH ROW

BEGIN

:NEW.EMP\_NAME := UPPER( :NEW.EMP\_NAME );

DBMS\_OUTPUT.PUT\_LINE('TRIGGER FIRED');

END;

/

--5. Write a PL/SQL block to fire any two built in exceptions in ORACLE by assuming your own data in a table.

1. CREATE TABLE DEPART2

(DEPART\_ID NUMBER(4) NOT NULL PRIMARY KEY,

DEPART\_NAME VARCHAR(30)NOT NULL ,

MANAGER\_ID NUMBER (6) UNIQUE

);

INSERT INTO DEPART2 VALUES(10,'Purchase ',123);

INSERT INTO DEPART2 VALUES(20,'Inventory ',345);

INSERT INTO DEPART2 VALUES(30,'Sales',567);

INSERT INTO DEPART2 VALUES(40,'Materials ',789);

INSERT INTO DEPART2 VALUES(50,'Production ',911);

DECLARE

d\_id depart2.depart\_id%type := 5;

d\_name depart2.depart\_name%type;

d\_mid depart2.manager\_id%type;

BEGIN

SELECT depart\_id,depart\_name,manager\_id into d\_id,d\_name,d\_mid from depart2

WHERE depart\_id=d\_id;

DBMS\_OUTPUT.PUT\_LINE ('Dept Id: '|| d\_id);

DBMS\_OUTPUT.PUT\_LINE ('Dept Name: '|| d\_name);

DBMS\_OUTPUT.PUT\_LINE ('Manager Id: ' || d\_mid);

EXCEPTION

WHEN no\_data\_found THEN

dbms\_output.put\_line('No such Record with Dept\_Id =5!');

WHEN others THEN

dbms\_output.put\_line('Error!');

END;

/

(2).

DECLARE

a int:=20;

b int:=0;

answer int;

BEGIN

answer:=a/b;

dbms\_output.put\_line('Result after division:'||answer);

exception

WHEN zero\_divide THEN

dbms\_output.put\_line('Dividing by zero, Please check the values again!!');

dbms\_output.put\_line('Value of a:'||a);

dbms\_output.put\_line('Value of b:'||b);

END;

/

-- 6. Write a function in PL/SQL to check whether the given string is a palindrome or not. (Use any appropriate built-in functions available)

DECLARE

S VARCHAR2(10) := 'REFER';

L VARCHAR2(20);

T VARCHAR2(10);

BEGIN

FOR I IN REVERSE 1..LENGTH(S) LOOP

L := SUBSTR(S, I, 1);

T := T

||''

||L;

END LOOP;

IF T = S THEN

DBMS\_OUTPUT.PUT\_LINE(T

||''

||' IS A PALINDROME STRING.');

ELSE

DBMS\_OUTPUT.PUT\_LINE(T

||''

||' IS NOT A PALINDROME STRING!!');

END IF;

END;

/

--7. Write a PL/SQL block to fire any one user defined exception by assuming your own data in a table.

DECLARE

d\_id depart2.depart\_id%type := &dd\_id;

d\_name depart2.depart\_name%type;

d\_mid depart2.manager\_id%type;

-- user defined exception

ex\_invalid\_id EXCEPTION;

BEGIN

IF d\_id <= 0 THEN

RAISE ex\_invalid\_id;

ELSE

SELECT depart\_id,depart\_name,manager\_id into d\_id,d\_name,d\_mid from depart2

WHERE depart\_id=d\_id;

DBMS\_OUTPUT.PUT\_LINE ('Dept Id: '|| d\_id);

DBMS\_OUTPUT.PUT\_LINE ('Dept Name: '|| d\_name);

DBMS\_OUTPUT.PUT\_LINE ('Manager Id: ' || d\_mid);

END IF;

EXCEPTION

WHEN ex\_invalid\_id THEN

dbms\_output.put\_line('ID must be greater than zero!');

WHEN no\_data\_found THEN

dbms\_output.put\_line('No such customer!');

WHEN others THEN

dbms\_output.put\_line('Error!');

END;

/

-- 8. Write a named procedure to delete records available in your table.

DECLARE

A NUMBER:=&A;

PROCEDURE DELETE\_RECORD(X IN NUMBER) IS

BEGIN

DELETE FROM EMPLOYEES WHERE EMP\_ID=X;

END;

BEGIN

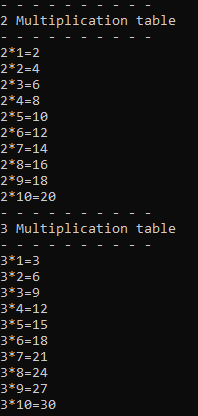
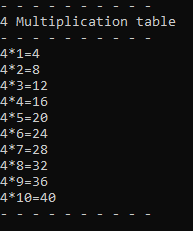
DELETE\_RECORD(A);

END;

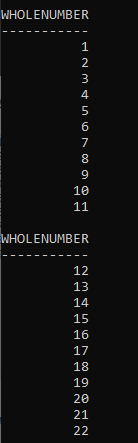
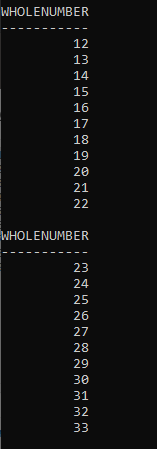
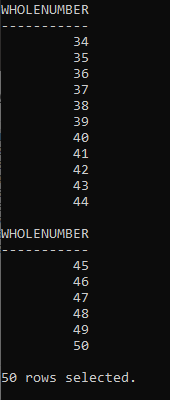
/

**OUTPUT SCREENSHOTS:**

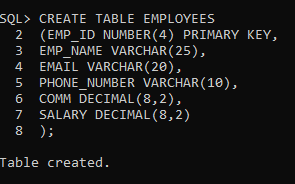
**1.**

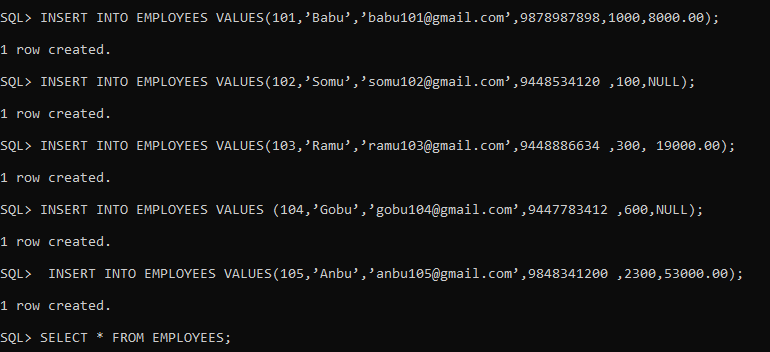
**** 

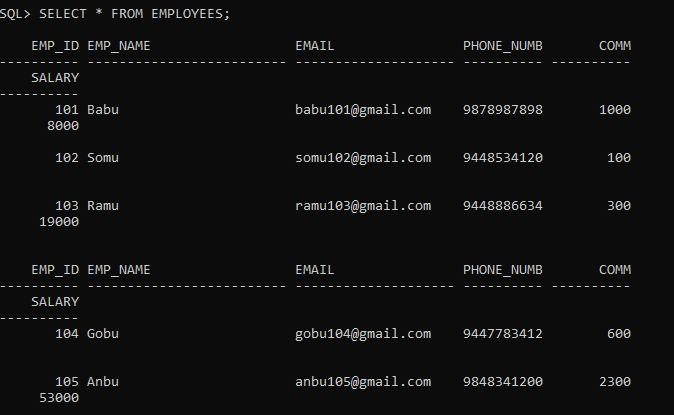
2.

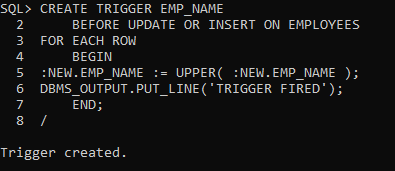
  

4.

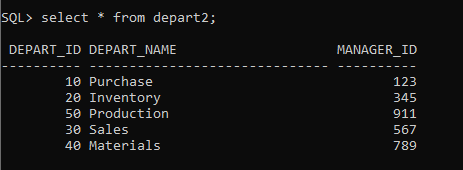








5.



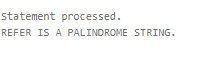
(1)



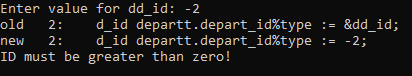
(2)



6.

7.



8.

