PCA 2

MAKAUT ODD SEMESTER 2024

NAME:	**RUPAK SARKAR**
STREAM:	**MCA**
SEMESTER:	**1ST**
SUBJECT:	**RELATIONAL DATABASE MANAGEMENT SYSTEM**
SUBJECT	**CODE: MCAN-192**

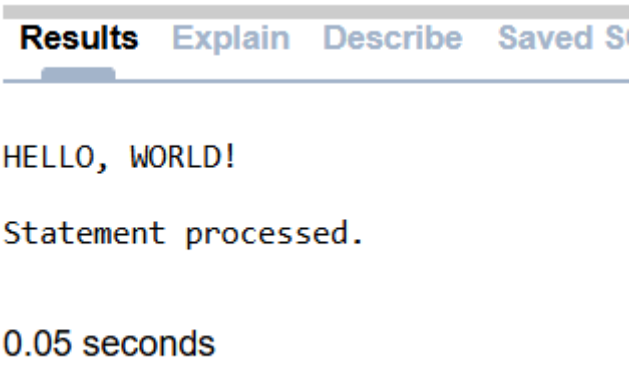
Q.1. Write a PL/SQL to print Hello World.

Ans:

SQL: DECLARE

```
  message VARCHAR2(20):= 'HELLO, WORLD!';
BEGIN
  dbms_output.put_line(message);
END;
```

Output:



Q.2. Write a PL/SQL program to find addition, subtraction, multiplication, division of two numbers.

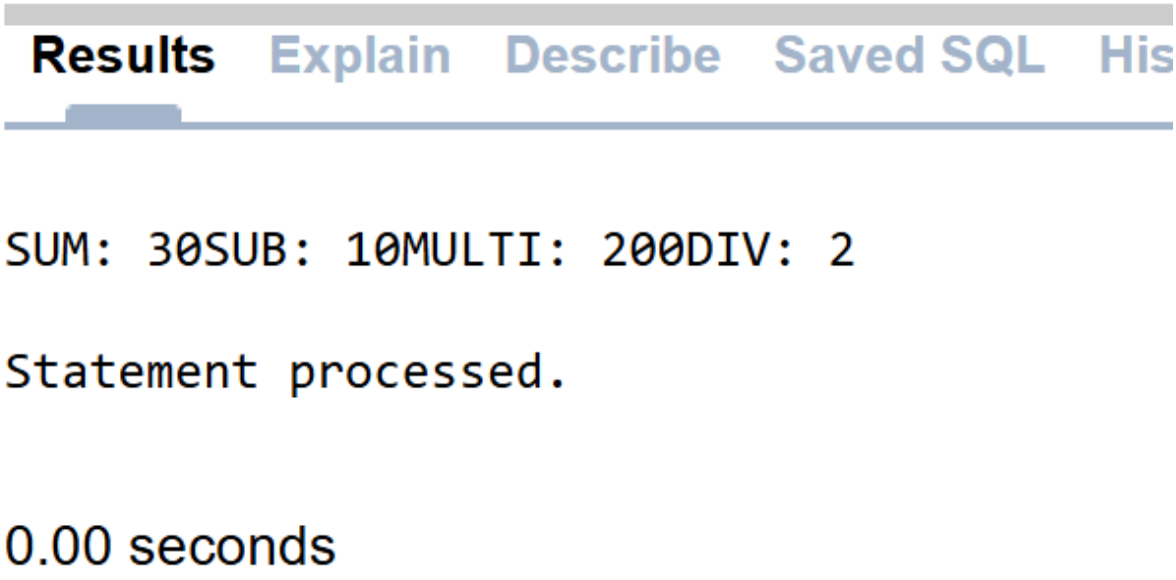
Ans:

SQL: DECLARE

```
A NUMBER(6);
B NUMBER(6);
S NUMBER(6);
SUB NUMBER(6);
M NUMBER(6);
D NUMBER(6);

BEGIN
```

```
A:=:A;

B:=:B;

S:=A+B;

SUB:=A-B;

M:=A*B;

D:=A/B;

DBMS_OUTPUT.PUT_LINE('SUM: '||S|| 'SUB: '||SUB||'MULTI: '||M||'DIV: '||D);

END;
```

Output:



Q.3. Write a PL/SQL program to print the maximum among three numbers.

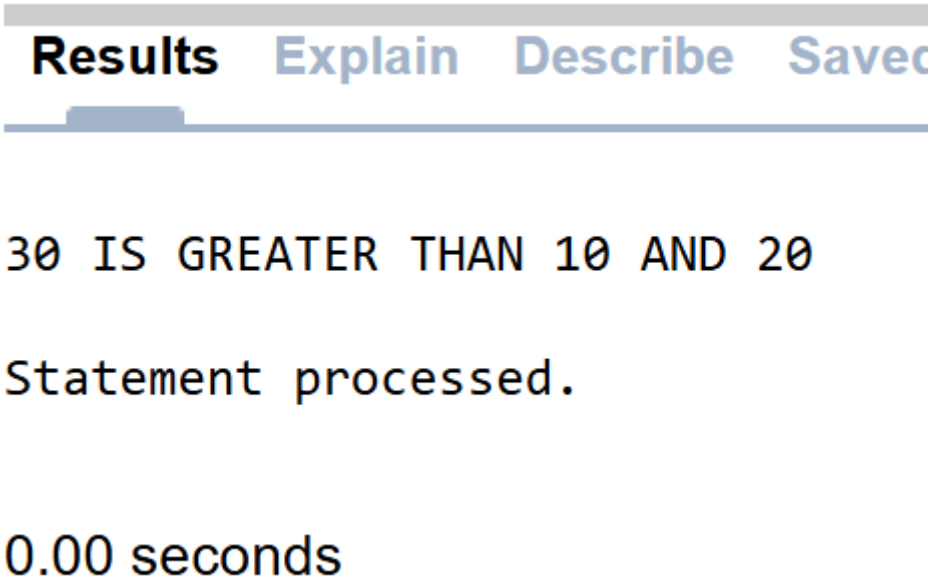
Ans:

SQL: DECLARE

```
A NUMBER(3); B NUMBER(3); C NUMBER(3);

BEGIN

A:=:A;

B:=:B;
```

```
C:=:C;

IF A>B AND A>C THEN
DBMS_OUTPUT.PUT_LINE(A||' IS GREATER THAN '||B||' AND '||C);
ELSE
IF B>C AND B>C THEN
DBMS_OUTPUT.PUT_LINE(B||' IS GREATER THAN '||A||' AND '||C); ELSE
DBMS_OUTPUT.PUT_LINE(C||' IS GREATER THAN '||A||' AND '||B); END IF;
END IF;
END;
```

Output:



Q.4. Write a PL/SQL program to print numbers from 1 -10 using different types of loop.

Ans:

SQL: DECLARE

```
A NUMBER(6):=0;
BEGIN
```

```
LOOP
IF A>=10 THEN
EXIT;
END IF;
A:=A+1;
DBMS_OUTPUT.PUT_LINE(A);
END LOOP;
END;
```

Output:

```
1
2
3
4
5
6
7
8
9
10

Statement processed.

0.01 seconds
```

WHILE LOOP:

SQL: DECLARE

```
A NUMBER(6):=1;
BEGIN
WHILE A<=10
LOOP
DBMS_OUTPUT.PUT_LINE(A);
A:=A+1;
```

END LOOP;

END;

Output:

```
1
2
3
4
5
6
7
8
9
10

Statement processed.

0.00 seconds
```

FOR LOOP:

SQL: DECLARE

A NUMBER(6):=0;

BEGIN

FOR A IN 0..10

LOOP

DBMS_OUTPUT.PUT_LINE(A);

END LOOP;

END;

Output:

```
0
1
2
3
4
5
6
7
8
9
10

Statement processed.

0.00 seconds
```

Q.5. Write a PL/SQL program to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named AREAS consisting of two columns RADIUS and AREA.

Ans:

SQL: DECLARE

PI CONSTANT NUMBER(4,2):=3.14;

V_RADIUS CIRCLE.RADIUS%TYPE;

V_AREA CIRCLE.AREA%TYPE;

BEGIN

V_RADIUS:=3;

WHILE V_RADIUS<=7

LOOP

```
V_AREA:=PI * POWER(V_RADIUS, 2);
INSERT INTO CIRCLE VALUES(V_RADIUS, V_AREA);
V_RADIUS:=V_RADIUS + 1;
END LOOP;
END;
```

Output:

Results Explain Describe Sa

RADIUS	AREA
3	28.26
4	50.24
5	78.5
6	113.04
7	153.86

5 rows returned in 0.01 seconds

Q.6. Write a PL/SQL program to Factorial of a number using function.

Ans:

SQL: DECLARE

```
NUM NUMBER;
FACTORIAL NUMBER;
```

```
FUNCTION FACT(X NUMBER)
RETURN NUMBER
IS
F NUMBER;

BEGIN
IF X=0 THEN
F:=1;
ELSE
F:=X * FACT(X-1);
END IF;
RETURN F;
DBMS_OUTPUT.PUT_LINE(F);
END;

BEGIN
NUM:=6;
FACTORIAL:=FACT(NUM);
DBMS_OUTPUT.PUT_LINE('FACTORIAL '||NUM|| ' IS '||FACTORIAL);
END;
```

OUTPUT:

Results Explain Desc

```
FACTORIAL 6 IS 720

Statement processed.
```

0.01 seconds

Q.7. Print HELLO WORLD using Procedure.

Ans:

SQL: CREATE OR REPLACE PROCEDURE GREETINGS

AS

BEGIN

DBMS_OUTPUT.PUT_LINE('HELLO WORLD');

END;

BEGIN

GREETINGS;

END;

OUTPUT:

Results Explain Des

```
Procedure created.
```

0.00 seconds

Results Explain Describe

```
HELLO WORLD

Statement processed.
```

0.00 seconds

Q.8. Write a PL/SQL script to create a table employee (emp_no, emp_name, salary, manager, dept_no). Write a program to create a row-level trigger for the employee table that would fire for INSERT or UPDATE operations performed on the employee table. This trigger will display the salary difference between the old values and new values.

Ans:

TABLE CREATION:

SQL: CREATE TABLE EMPLOYEE(EMP_NO NUMBER, EMP_NAME VARCHAR2(50), SALARY NUMBER, MANAGER VARCHAR2(50), DEPT_NO NUMBER);

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **EMPLOYEE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMP_NO	NUMBER	22	-	-	-	✓	-	-
	EMP_NAME	VARCHAR2	50	-	-	-	✓	-	-
	SALARY	NUMBER	22	-	-	-	✓	-	-
	MANAGER	VARCHAR2	50	-	-	-	✓	-	-
	DEPT_NO	NUMBER	22	-	-	-	✓	-	-

1 - 5

VALUE INSERTION:

SQL:

BEGIN

INSERT INTO employee VALUES(1001,'YYY5',18000,'AAA',20) ;

INSERT INTO employee VALUES(1003,'PPP5',20000,'BBB',10);

COMMIT;

END;

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	SALARY	MANAGER	DEPT_NO
1001	YYY5	18000	AAA	20
1003	PPP5	20000	BBB	10

2 rows returned in 0.00 seconds Download

TRIGGER CREATION:

SQL: CREATE OR REPLACE TRIGGER salary_difference

BEFORE INSERT OR UPDATE ON employee

FOR EACH ROW

WHEN (NEW.emp_no > 0)

DECLARE

sal_diff number;

BEGIN

sal_diff := :NEW.salary - :OLD.salary;

dbms_output.put_line('Old salary: ' || :OLD.salary);

dbms_output.put_line('New salary: ' || :NEW.salary);

dbms_output.put_line('Salary difference: ' || sal_diff);

END;

Results Explain Describe Saved SQL History

```
Trigger created.
```

0.02 seconds

TRIGGER CHECKING:

SQL: INSERT INTO employee (emp_no,emp_name,salary,manager,dept_no)

VALUES(1005,'RRR5',30000,'BBB',10);

OUTPUT:

Results Explain Describe Saved SQL History

```
Old salary:
New salary: 30000
Salary difference:

1 row(s) inserted.
```

0.00 seconds

SQL: UPDATE employee

SET salary = salary + 500

WHERE emp_no=1003;

OUTPUT:

Results Explain Describe Saved SQL History

```
Old salary: 20000
New salary: 20500
Salary difference: 500

1 row(s) updated.
```

0.00 seconds

Q.9. Write a PL/SQL script to print the details of the employee having the 2nd highest salary from employee table using explicit cursor.

Ans:

TABLE CREATION:

SQL: CREATE TABLE EMP_EC(emp_no number(8),emp_name varchar2(10),salary number(8));

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	SALARY
1001	SAYAK	25000
1002	OLI	35000
1005	PRITAM	25400
1003	SIMRAN	20000
1004	SATADRU	45000

5 rows returned in 0.00 seconds Download

CURSOR IMPLEMENTION:

SQL: DECLARE

```
 emp_no EMP_EC.emp_no%type;
 emp_name EMP_EC.emp_name%type;
 salary EMP_EC.salary%type;
 CURSOR employee is
 SELECT emp_no,emp_name,salary from EMP_EC
 where salary = (select max(salary) from EMP_EC where salary <
(select max(salary) from EMP_EC));
BEGIN
 OPEN employee;
 LOOP
 FETCH employee INTO emp_no,emp_name,salary;
 EXIT WHEN employee%NOTFOUND;
 dbms_output.put_line(emp_no||' '||emp_name||' '||salary);
 END LOOP;
 CLOSE employee;
END;
```

OUTPUT:

Results Explain Describe Saved SQL History

```
1002 OLI 35000

Statement processed.
```

0.01 seconds

Q.10. Write a PL/SQL script to implement Package.

Ans:

SQL: CREATE OR REPLACE PACKAGE my_package AS

```
  PROCEDURE greet(name VARCHAR2);
END my_package;


CREATE OR REPLACE PACKAGE BODY my_package AS
  PROCEDURE greet(name VARCHAR2) IS
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello, ' || name);
  END greet;
END my_package;
BEGIN
  my_package.greet('HEY THERE');
END;
```

OUTPUT:

Results Explain Describe Saved SQL History

```
Hello, HEY THERE

Statement processed.
```

0.00 seconds