**1.Display the names and ages of programmers**

select pname,extract(year from sysdate)-extract(year from dob) as age from programmer;

**2. Which female programmer earning more than 3000 doesn’t know C++, Oracle or dbase**

select pname from programmer where gender='f' and salary>3000 and prof1 not in ('c++','dbase','oracle') and prof2 not in ('c++','dbase','oracle');

1. **Which institute has the most number of students**

select splace,count(splace) from studies group by splace;

**4. Display language wise count of prof1**

select prof1,count(prof1) from programmer group by prof1;

**5.Display the number of people born in each year**

select extract(year from dob),count(extract(year from dob)) from programmer group by extract(year from dob);

**6. Display details of programmers draw 2000-4000**

select pname from programmer where salary between 2000 and 4000;

**7. Display the programmers name and their software’s**

select pname,title from software;

**8. What is the highest numbers of copies sold by a package**

SELECT MAX(no\_of\_products\_sold) AS highest\_sold FROM Software;

**9.Display each institute name with number of students**

SELECT study\_place, COUNT(\*) AS student\_count FROM Studies GROUP BY study\_place;

**10.Display the number of people born in each month**

SELECT MONTH(date\_of\_birth) AS birth\_month, COUNT(\*) AS count FROM Programmer GROUP BY birth\_month;

**11. What are the languages known by male programmers**

select prof1 from programmer where gender='m' union select prof2 from programmer where gender='m';

**12. Display lowest course fee**

select min(cost) from studies;

**13. Which course has been done by most of the students**

select course,count(course) from studies group by course having count(course)=(select max(student) from(select count(course) as student from studies group by course));

**14. Display each institute name with number of courses, average cost per course**

SELECT study\_place, COUNT(course\_name) AS number\_of\_courses, AVG(course\_cost) AS average\_cost FROM Studies GROUP BY study\_place;

**15. Display the number of people studied in each institute**

select splace,count(course) from studies group by splace;

**16. Display the details of software developed by Ramesh?**

select \* from software where pname='ramesh';

**17. How many people draw 2000-4000**

SELECT COUNT(\*) AS count FROM Programmer WHERE salary BETWEEN 2000 AND 4000;

**18. Which is the costliest software developed in .NET**

select max(scost) from software where dev='.net' group by dev;

**19. Display the sales cost of the software’s developed by each programmer**

select pname,sum(dcost) from software group by pname;

**20. Which language have been stated as prof1 by most of the programmers**

select prof1,count(prof1) from programmer group by prof1 having count(prof1)=(select max(prof1\_count) from (select count(prof1) as prof1\_count from programmer group by prof1));

**21. What is the price of the costliest software developed in BASIC?**

select max(scost) from software where dev='basic' group by dev;

**22.Display the details of programmers who don’t know Fortran, Cobol or .NET**

select \* from programmer where prof1 not in ('fortran','cobol','.net') and prof2 not in ('fortran','cobol','.net');

**23.** **Display the names of programmers whose names contains up to 5 characters**

select pname from programmer where length(pname)<=5;

**24. Display the number of softwares sold by each programmer**

select pname,count(sold) from software group by pname;

**25. How old is the oldest male programmer**

select pname,extract(year from sysdate)-extract(year from dob) as age from programmer where extract(year from sysdate)-extract(year from dob)=(select max(extract(year from sysdate)-extract(year from dob)) from programmer);

**26. Identify and include the keys (Primary Keys & Foreign Keys) in the tables**

**27. Display the names of programmers who have done the DAP course**

select pname from studies where course='dap';

**28. How many female programmers knowing Cobol have more than 2 years experience**

select pname from programmer where gender='f' and (prof1='cobol' or prof2='cobol') and extract(year from sysdate)-extract(year from doj)>2;

**29.** **Display the number of software’s in each language except C and C++**

select dev,count(title),count(dev) from software where dev not in ('c','c++') group by dev;

**30. How many programmers don’t know .NET and C**

select \* from programmer where prof1 not in ('.net','c') and prof2 not in ('.net','c');

**31. Display the details of software’s whose sales have been crossed 2000 mark**

select title from software where scost\*sold-dcost>2000;

**32. Display the names of male and female programmers**

select pname from programmer where gender in ('m','f');

**33. Display the name of the highest paid programmers for each language**

SELECT profession1, programmer\_name, MAX(salary) AS highest\_salary

FROM Programmer GROUP BY profession1 ORDER BY highest\_salary DESC;

**34. Display the selling cost of software’s developed in each language**

SELECT platform\_used, SUM(selling\_cost) AS total\_selling\_cost FROM Software

GROUP BY platform\_used;

**35. Calculate the experience in years for each programmer and display along with their names in descending order**

SELECT programmer\_name, DATEDIFF(CURDATE(), date\_of\_joining) / 365 AS experience\_years FROM Programmer ORDER BY experience\_years DESC;

**PL/SQL**

**EXP 1:**

SET SERVEROUTPUT ON

DECLARE

A NUMBER;

B NUMBER DEFAULT 0;

C B%TYPE;

PI CONSTANT NUMBER(3,2):=3.14;

BEGIN

A:=&n;

C:=&j;

dbms\_output.put\_line('tHE VALUE OF A IS '||A);

DBMS\_OUTPUT.PUT\_LINE('tHE VALUE OF B IS '||B);

DBMS\_OUTPUT.PUT\_LINE('tHE VALUE OF C IS '||C);

DBMS\_OUTPUT.PUT\_LINE('tHE VALUE OF PI IS '||PI);

END;

/

**OUTPUT:**

The Value of a is10

The Value of b is0

The Value of c is0

The Value of pi is3.14

**EXP 2:**

set serveroutput ON

VARIABLE numb NUMBER;

DECLARE

a number;

b number default 0;

c b%TYPE;

Pi constant number(3,2):=3.14;

BEGIN

a:=&A;

c:=b;

:numb:=c;

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

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

dbms\_output.put\_line('The Value of c is'||c);

dbms\_output.put\_line('The Value of Constant pi is'||Pi);

dbms\_output.put\_line('The Value of Bind variable numb is'||:numb);

END;

/

**OUTPUT:**

Enter value for a: 20

old 7: a:=&A;

new 7: a:=20;

The Value of a is20

The Value of b is0

The Value of c is0

The Value of Constant pi is3.14

The Value of Bind variable numb is0

PL/SQL procedure successfully completed

EXP 3(SIMPLE IF):

SET SERVEROUTPUT ON

DECLARE

A NUMBER;

B NUMBER;

C NUMBER;

BEGIN

A:=&A;

B:=&B;

IF A>B THEN

C:=A;

A:=B;

B:=C;

END IF;

DBMS\_OUTPUT.PUT\_LINE('VALUE OF A '||A);

DBMS\_OUTPUT.PUT\_LINE('VALUE OF B '||B);

END;

/

OUTPUT 1:

Enter value for a: 10

old 6: A:=&A;

new 6: A:=10;

Enter value for b: 15

old 7: B:=&B;

new 7: B:=15;

VALUE OF A 10

VALUE OF B 15

OUTPUT 2:

Enter value for a: 20

old 6: A:=&A;

new 6: A:=20;

Enter value for b: 10

old 7: B:=&B;

new 7: B:=10;

VALUE OF A 10

VALUE OF B 20

EXP 3(IF ELSE):

SET SERVEROUTPUT ON

DECLARE

A NUMBER;

B NUMBER;

BEGIN

A:=&A;

B:=&B;

IF MOD(A,2)=0 THEN

dbms\_output.put\_line('A is even number');

ELSE

DBMS\_OUTPUT.PUT\_LINE('A IS ODD NUMBER');

END IF;

IF MOD(B,2)=0 THEN

DBMS\_OUTPUT.PUT\_LINE('B IS EVEN NUMBER');

ELSE

DBMS\_OUTPUT.PUT\_LINE('B IS ODD NUMBER');

END IF;

END;

/

CASE

SET SERVEROUTPUT ON

DECLARE

A NUMBER;

BEGIN

A :=&ENTER\_VALUE\_OF\_A;

CASE

WHEN A>0 THEN

DBMS\_OUTPUT.PUT\_LINE('A IS POSITIVE');

WHEN A<0 THEN

DBMS\_OUTPUT.PUT\_LINE('A IS NEGATIVE');

WHEN A=0 THEN

DBMS\_OUTPUT.PUT\_LINE('A IS ZERO');

END CASE;

END;

/

FOR LOOP

SETSERVEROUTPUT ON

DECLARE

I NUMBER;

BEGIN

I:=&ENTER\_NUMBER\_OF\_ITERATIONS;

FOR K IN REVERSE 1..I

LOOP

DBMS\_OUTPUT.PUT\_LINE(K);

END LOOP;

END;

/

WHILE LOOP

SET SERVEROUTPUT ON

DECLARE

I NUMBER;

K NUMBER :=0;

BEGIN

I :=&ENTER\_NO\_OF\_ITERATIONS;

WHILE K<=I

LOOP

DBMS\_OUTPUT.PUT\_LINE(K);

K:=K+1;

END LOOP;

END;

/

CURSORS

SET SERVEROUTPUT ON

DECLARE

C\_PNAME STUDIES.PNAME%TYPE;

C\_SPLACE STUDIES.SPLACE%TYPE;

C\_COURSE STUDIES.COURSE%TYPE;

C\_COST STUDIES.COST%TYPE;

CURSOR C\_STUDIES IS SELECT PNAME,SPLACE,COURSE,COST FROM STUDIES;

BEGIN

OPEN C\_STUDIES;

LOOP

FETCH C\_STUDIES INTO C\_PNAME,C\_SPLACE,C\_COURSE,C\_COST;

EXIT WHEN C\_STUDIES%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(C\_PNAME||' '||C\_SPLACE||' '||C\_COURSE||' '||C\_COST);

END LOOP;

CLOSE C\_STUDIES;

END;

/

CURSOR PRACTICE PROGRAMS

1.

SET SERVEROUTPUT ON

DECLARE

C\_PNAME STUDIES.PNAME%TYPE;

CURSOR C\_STUDIES IS SELECT PNAME FROM STUDIES;

BEGIN

OPEN C\_STUDIES;

LOOP

FETCH C\_STUDIES INTO C\_PNAME;

EXIT WHEN C\_STUDIES%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(C\_PNAME);

END LOOP;

CLOSE C\_STUDIES;

END;

/

2.

SET SERVEROUTPUT ON

DECLARE

C\_SPLACE STUDIES.SPLACE%TYPE;

CURSOR C\_STUDIES IS SELECT SPLACE FROM STUDIES;

BEGIN

OPEN C\_STUDIES;

LOOP

FETCH C\_STUDIES INTO C\_SPLACE;

EXIT WHEN C\_STUDIES%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(C\_SPLACE);

END LOOP;

CLOSE C\_STUDIES;

END;

/

EXCEPTIONS

SET SERVEROUTPUT ON

DECLARE

A NUMBER;

B NUMBER;

BEGIN

A :=&ENTER\_A;

B :=&ENTER\_B;

DBMS\_OUTPUT.PUT\_LINE('A/B IS '||A/B);

EXCEPTION WHEN ZERO\_DIVIDE THEN

DBMS\_OUTPUT.PUT\_LINE('CANNOT DIVIDE WITH ZERO');

END;

/

RECORD

SET SERVEROUTPUT ON

DECLARE

STUDIES\_REC STUDIES%ROWTYPE;

BEGIN

SELECT \* INTO STUDIES\_REC FROM STUDIES WHERE PNAME='MARY';

DBMS\_OUTPUT.PUT\_LINE('NAME : '||STUDIES\_REC.PNAME);

DBMS\_OUTPUT.PUT\_LINE('SPLACE : '||STUDIES\_REC.SPLACE);

DBMS\_OUTPUT.PUT\_LINE('COURSE : '||STUDIES\_REC.COURSE);

DBMS\_OUTPUT.PUT\_LINE('COST : '||STUDIES\_REC.COST);

END;

/

PRACTICE PROGRAMS FOR RECORDS

1.

SET SERVEROUTPUT ON

DECLARE

STUDIES\_REC STUDIES%ROWTYPE;

BEGIN

SELECT \* INTO STUDIES\_REC FROM STUDIES WHERE PNAME='VIJAYA';

DBMS\_OUTPUT.PUT\_LINE('NAME : '||STUDIES\_REC.PNAME);

DBMS\_OUTPUT.PUT\_LINE('SPLACE : '||STUDIES\_REC.SPLACE);

DBMS\_OUTPUT.PUT\_LINE('COURSE : '||STUDIES\_REC.COURSE);

DBMS\_OUTPUT.PUT\_LINE('COST : '||STUDIES\_REC.COST);

END;

/

VARRAY

SET SERVEROUTPUT ON

DECLARE

STUDIES\_REC STUDIES%ROWTYPE;

BEGIN

SELECT \* INTO STUDIES\_REC FROM STUDIES WHERE PNAME='VIJAYA';

DBMS\_OUTPUT.PUT\_LINE('NAME : '||STUDIES\_REC.PNAME);

DBMS\_OUTPUT.PUT\_LINE('SPLACE : '||STUDIES\_REC.SPLACE);

DBMS\_OUTPUT.PUT\_LINE('COURSE : '||STUDIES\_REC.COURSE);

DBMS\_OUTPUT.PUT\_LINE('COST : '||STUDIES\_REC.COST);

END;

/

SET SERVEROUTPUT ON

DECLARE

TYPE NAMESARRAY IS VARRAY(3) of VARCHAR(10);

NAMES NAMESARRAY;

BEGIN

NAMES := NAMESARRAY('RAMESH','SURESH','KAMLESH');

FOR I IN 1..NAMES.COUNT

LOOP

DBMS\_OUTPUT.PUT\_LINE(NAMES(I));

END LOOP;

END;

/

PRACTICE PROGRAMS

1.

SET SERVEROUTPUT ON

DECLARE

TYPE NAMESARRA IS VARRAY(3) OF VARCHAR(20);

NAMES NAMESARRA;

BEGIN

NAMES:= NAMESARRA('rASHEED','HEMANTH','NAYUM');

FOR I IN 1..NAMES.COUNT

LOOP

DBMS\_OUTPUT.PUT\_LINE(NAMES(I));

END LOOP;

END;

/

2.

DECLARE

TYPE RASHEED IS VARRAY(10) OF INTEGER(2);

NAMES RASHEED;

BEGIN

NAMES := RASHEED(1,2,5,6,7,11,13,15,27,28);

FOR I IN 1..NAMES.COUNT

LOOP

DBMS\_OUTPUT.PUT\_LINE(NAMES(I));

END LOOP;

END;

/

PROCEDURES

DECLARE

A NUMBER;

B NUMBER;

C NUMBER;

PROCEDURE ADD(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER) IS

BEGIN

Z:=X+Y;

END;

BEGIN

A:=10;

B:=20;

ADD(A,B,C);

DBMS\_OUTPUT.PUT\_LINE('ADDITION OD A AND B IS '||C);

END;

/

CREATION OF A PROCEDURE

CREATE OR REPLACE PROCEDURE ADDITION (X IN NUMBER, Y IN NUMBER, Z OUT NUMBER) IS

BEGIN

Z:=X+Y;

END;

/

DECLARE

A NUMBER;

B NUMBER;

C NUMBER;

BEGIN

A:=10;

B:=20;

ADDITION(A,B,C);

DBMS\_OUTPUT.PUT\_LINE(C);

END;

/

FUNCTIONS

SET SERVEROUTPUT ON

DECLARE

A NUMBER;

B NUMBER;

C NUMBER;

FUNCTION SAMPLE(X IN NUMBER,Y IN NUMBER) RETURN NUMBER IS

Z NUMBER;

BEGIN

Z:=X+Y;

RETURN Z;

END;

BEGIN

A:=10;

B:=20;

C:=SAMPLE(A,B);

DBMS\_OUTPUT.PUT\_LINE(C);

END;

/

FUNCTION DECLARATION

CREATE OR REPLACE FUNCTION ADDITION2(X IN NUMBER , Y IN NUMBER) RETURN NUMBER AS

Z NUMBER;

BEGIN

Z:=X+Y;

RETURN Z;

END;

/