DBMS LAB MANUAL

IV. Write SQL queries for subqueries, nested queries.

Given:

EMPLOYEE TABLE

EMPNO	ENAME	JOB	DEPTNO	SALARY
1	Mathi	AP	1	30000
2	Arjun	ASP	2	32000
3	Gugan	ASP	2	40000
4	Karthik	AP	1	35000

Q1. Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with "A".

SQL>select ename,sal from emp where sal>(select min(sal) from emp where job like 'A%');
Output:

ENAME	SALARY
Arjun	32000
Gugan	40000
Karthik	35000

Q2. Display all employee names and salary whose salary is greater than minimum salary of the company and job title is AS

V.Write an SQL query to implement JOINS in relational Algebra.

Given:

EMPLOYEE TABLE

EMPNO	ENAME	JOB	DEPTNO	SALARY
1	Mathi	AP	1	30000
2	Arjun	ASP	2	32000
3	Gugan	ASP	2	40000
4	Karthik	AP	1	35000

DEPARTMENT TABLE

DEPTNO	DNAME	LOCATION
1	ACCOUNTING	NEW YORK
2	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Q1. Display the employee details, departments that the departments are same in both the emp and dept.

SQL>select * from emp,dept where emp.deptno=dept.deptno;

Output:

EMP NO	ENAME	JOB	DEPT NO	SALARY	DEPT NO	DNAME	LOCATIO N
1	Mathi	AP	1	30000	1	ACCOUNTIN G	NEW YORK
2	Arjun	ASP	2	32000	2	RESEARCH	DALLAS
3	Gugan	ASP	2	40000	2	RESEARCH	DALLAS
4	Karthik	AP	1	35000	1	ACCOUNTNG	NEW YORK

VI. Write program by the use of PL/SQL.

Query:

1. Write PL/SQL Program to find the sum two numbers.

```
DECLARE
   a number:=2;
  b number:=3;
  c number;
  begin
  c := a+b;
 dbms_output.put_line('Sum of two numbers:='||c);
 END;
 /
```

Output:
Sum of two numbers: =5

2. Write PL/SQL Program to find the greater number.

```
DECLARE
a number (2) := 21;
b number (2) := 10;
BEGIN
IF (a = b) then
dbms_output.put_line('Line 1 - a is equal to b');
ELSE
dbms_output.put_line('Line 1 - a is not equal to b');
END IF;
IF (a < b) then
dbms_output.put_line('Line 2 - a is less than b');
ELSE
dbms_output.put_line('Line 2 - a is not less than b');
END IF;
IF ( a> b ) THEN
dbms_output.put_line('Line 3 - a is greater than b');
ELSE
dbms_output.put_line('Line 3 - a is not greater than b');
END IF;
END;
/
```

Output:
Line 1 - a is not equal to b
Line 2 - a is not less than b
Line 3 - a is greater than b

VII. Write SQL queries to create views.

Syntax:
CREATE VIEW view_name AS
SELECT column1,column2,....
FROM table_name
WHERE condition;

Query:
Q1. Write a SQL query to create a view of the customer table created in PRACTICAL no 1.
SQL>CREATE VIEW CUST as
Select ID, Name, Address
From Customer;

CUSTOMER TABLE

ID	NAME	AGE	ADDRESS	SALARY
1	Akshay	25	Delhi	30000
2	Manish	27	Mumbai	35000
3	Kushagra	26	Kolkata	30000
4	Mukesh	31	Hyderabad	32000
5	Himanshu	29	Chennai	40000
6	Neeraj	30	Noida	36000
7	Nishant	32	Delhi	30000

Output

ID	NAME	ADDRESS
1	Akshay	Delhi
2	Manish	Mumbai
3	Kushagra	Kolkata
4	Mukesh	Hyderabad
5	Himanshu	Chennai
6	Neeraj	Noida
7	Nishant	Delhi

VIII. Write a query for extracting data from more than one table.

Query:

EMPLOYEE TABLE

EMPNO	ENAME	JOB	DEPTNO	SALARY
1	Mathi	AP	1	30000
2	Arjun	ASP	2	32000
3	Gugan	ASP	2	40000
4	Karthik	AP	1	35000

DEPARTMENT TABLE

DEPTNO	DNAME	LOCATION
1	ACCOUNTING	NEW YORK
2	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Q1. Write a query to extract empno, ename, salary, dname and location from employee and department table where empno = deptno without using joins.

SQL> select employee.empno, employee.ename, employee.salary, department.dname, department.location
From department, employee
Where department.deptno = employee.empno;
Output:

EMPNO	ENAME	SALARY	DNAME	LOCATION
1	Mathi	30000	ACCOUNTING	NEW YORK
2	Arjun	32000	RESEARCH	DALLAS

2 rows selected.

Q2. Write a query to extract ename, salary and location from employee and department table where is like (30, 40).

SQL> select employee. ename, employee.salary, department.location
From department, employee
Where department.deptnoIN (30,40);

Output:
No rows Selected.

IX. Write a query to understand the concepts for ROLLBACK, COMMIT & CHECK POINTS.

QUERY:

Q1. Write a query to implement the save point.

```
SQL> select employee.empno, employee.ename, employee.salary, department.dname,
department.location
From department, employee
Where department.deptno = employee.empno;
SQL> SAVEPOINT S1;
Savepoint created.
```

Q2. Write a query to implement the Rollback.

```
SQL>ROLL BACK S1;
```