

EX 4. JOINS, SET OPERATORS, NESTED QUERIES

Joins

1. List the Employee names and their department names. (Use Equi-Join)

```
SQL> select empname,depname from employee,department where employee.deptno=department.deptno;
```

EMPNAME	DEPTNAME
Siva	accounts
Mani	sales
Raju	sales
Babu	purchase
Ram	production
Velu	software
Ravi	accounts
Balan	production
Mahesh	purchase
Kumar	software

10 rows selected.

2. List the Employees who earn more than Raju. (Use Self-Join)

```
SQL> select e1.empname
2   from employee e1
3   join employee e2 on e1.salary > e2.salary
4   where e2.empname = 'raju';
```

EMPNAME
velu
ravi
mahesh
kumar

3. List the Employee names and their department names.(Use Outer join)

```
SQL> select e.empname, d.depname
  2   from employee e
  3  left outer join department d on e.depno= d.depno;
```

EMPNAME	DEPNAME
Mani	sales
Raju	sales
Babu	purchase
Mahesh	purchase
Ram	production
Balan	production
Siva	accounts
Ravi	accounts
Velu	software
Kumar	software

10 rows selected.

4. List the employees who belongs to the department
“Production”,”Sales”,”Software”.

(Use Non-Equi join)

* Use ‘Employee’ and ‘Department’ tables .

```
SQL> select empname, depname
  2   from employee, department
  3  where employee.depno = department.depno
  4    and department.depname in ('production', 'sales', 'software');
```

EMPNAME	DEPNAME
Mani	sales
Raju	sales
Ram	production
Velu	software
Balan	production
Kumar	software

6 rows selected.

Set Operators

1. List the products available in both the tables without duplication.

```
SQL> create table product1 (  
2      item_code varchar2(5),  
3      item_name varchar2(10)  
4  );
```

Table created.

```
SQL> v  
SP2-0042: unknown command "v" - rest of line ignored.
```

```
SQL> create table product2 (  
2      item_code varchar2(5),  
3      item_name varchar2(10)  
4  );
```

Table created.

```
SQL>
SQL> insert into product1 values ('P001', 'Laptop');
1 row created.

SQL> insert into product1 values ('P002', 'Mouse');
1 row created.

SQL> insert into product1 values ('P003', 'Keyboard');
1 row created.

SQL> insert into product1 values ('P004', 'Monitor');
1 row created.

SQL> insert into product1 values ('P005', 'Tablet');
1 row created.

SQL>
SQL>
SQL> insert into product2 values ('P003', 'Keyboard');
1 row created.

SQL> insert into product2 values ('P004', 'Monitor');
1 row created.

SQL> insert into product2 values ('P006', 'Printer');
1 row created.

SQL> insert into product2 values ('P007', 'Scanner');
1 row created.

SQL> insert into product2 values ('P005', 'Tablet');
1 row created.
```

```
SQL> select item_name from product1
2 union
3 select item_name from product2;
```

```
ITEM_NAME
-----
Laptop
Mouse
Keyboard
Monitor
Tablet
```

2. List the products available in both the tables with duplication.

```
SQL> select item_name from product1
2 union all
3 select item_name from product2;
```

```
ITEM_NAME
-----
Laptop
Mouse
Keyboard
Monitor
Tablet
Keyboard
Monitor
Printer
Scanner
Tablet

10 rows selected.
```

3. List the products common to both the tables .

```
SQL> select item_name from product1
  2  intersect
  3  select item_name from product2;

ITEM_NAME
-----
Keyboard
Monitor
Tablet
```

4. List the products available in 'Product 1' table but not available in 'Product 2' table.

```
SQL> select item_name from product1
  2  minus
  3  select item_name from product2;

ITEM_NAME
-----
Laptop
Mouse
```

Sub Queries:

1. Find the employee with the maximum salary.

```
SQL> select empname, salary
  2  from employee
  3  where salary = (select max(salary) from employee);

EMPNAME                SALARY
-----
Siva                    15000.00
```

2. List the employees having salary greater than the lowest of the average salary of departments.(Use ANY operator)

```
SQL> select empname, salary
  2  from employee
  3  where salary > any (
  4      select avg(salary)
  5      from employee
  6      group by depno
  7  );
```

EMPNAME	SALARY
Siva	15000.00
Raju	7000.00
Velu	10000.50
Ravi	8000.25
Balan	12000.75
Mahesh	10000.50
Kumar	14500.00

7 rows selected.

3. List the employees having salary lower than the greatest of the averagesalary of all departments.(Use ALL operator).

```
SQL> select empname, salary
  2  from employee
  3  where salary < all (
  4      select avg(salary)
  5      from employee
  6      group by depno
  7  );
```

EMPNAME	SALARY
Mani	5000.75
Babu	5000.50
Ram	3000.25

4. List the departments where there are employees functioning. (Use EXISTS operator)

```
SQL> select depno, depname
  2   from department d
  3   where exists (
  4       select 1
  5       from employee e
  6       where e.depno = d.depno
  7   );
```

DEPNO	DEPNAME
5	accounts
1	sales
2	purchase
3	production
6	software

5. List the details of the employees belonging to 'Software' department.

```
SQL> select e.*
  2   from employee e
  3   join department d on e.depno = d.depno
  4   where d.depname = 'software';
```

EMPNO	EMPNAME	DESIGNATIO	DATEJOIN	SALARY	DEPNO	PHONENO	G
6	Velu	Programmer	24-FEB-02	10000.50	6		F
10	Kumar	Analyst	15-JAN-95	14500.00	6		C

6. Display designation, department number, total number of employees designation wise and department wise and the number of employees in each department and for each job must greater than 2 (use group by having)

```
SQL> insert into employee (empno, empname, designation, datejoin, salary, depno, grade)
  2  values (11, 'Ajay', 'Clerk', to_date('01-Jan-2000', 'DD-MON-YYYY'), 5500, 1, 'E');

1 row created.

SQL>
SQL> insert into employee (empno, empname, designation, datejoin, salary, depno, grade)
  2  values (12, 'Vijay', 'Clerk', to_date('15-Feb-2001', 'DD-MON-YYYY'), 6000, 1, 'E');

1 row created.

SQL> select designation, depno, count(*) as total_employees
  2  from employee
  3  group by designation, depno
  4  having count(*) > 2;
```

DESIGNATIO	DEPNO	TOTAL_EMPLOYEES
Clerk	1	3