# EX 3 SubQueries, Views and Joins

## Aim:

To view implement SubQueries, Views and Joins.

## Create employee Table

```
CREATE TABLE EMP (EMPNO NUMBER(4) PRIMARY KEY,ENAME VARCHAR2(10),JOB VARCHAR2(9),MGR N
```

## Insert the values

```
INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7369, 'SMITH', 'CLERK', 7902, '17-DEC-80', 800, NULL, 20);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7499, 'ALLEN', 'SALESMAN', 7698, '20-FEB-81', 1600, 300, 30);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7521, 'WARD', 'SALESMAN', 7698, '22-FEB-81', 1250, 500, 30);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7566, 'JONES', 'MANAGER', 7839, '02-APR-81', 2975, NULL, 20);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7654, 'MARTIN', 'SALESMAN', 7698, '28-SEP-81', 1250, 1400, 30);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7698, 'BLAKE', 'MANAGER', 7839, '01-MAY-81', 2850, NULL, 30);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7782, 'CLARK', 'MANAGER', 7839, '09-JUN-81', 2450, NULL, 10);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7788, 'SCOTT', 'ANALYST', 7566, '19-APR-87', 3000, NULL, 20);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7839, 'KING', 'PRESIDENT', NULL, '17-NOV-81', 5000, NULL, 10);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7844, 'TURNER', 'SALESMAN', 7698, '08-SEP-81', 1500, 0, 30);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7876, 'ADAMS', 'CLERK', 7788, '23-MAY-87', 1100, NULL, 20);
```

```
INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7900, 'JAMES', 'CLERK', 7698, '03-DEC-81', 950, NULL, 30);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7902, 'FORD', 'ANALYST', 7566, TO_DATE('03-DEC-81', 'DD-MON-RR'), 3000, 20, 20

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES (7934, 'MILLER', 'CLERK', 7782, TO_DATE('23-JAN-82', 'DD-MON-RR'), 1300, 10, 10
```

## Create department table

```
CREATE TABLE DEPT (DEPTNO NUMBER(2) PRIMARY KEY,DNAME VARCHAR2(14),LOC VARCHAR2(13));
```

## Insert the values in the department table

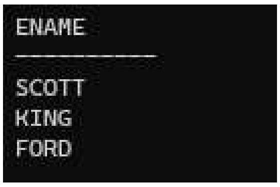
```
INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES (10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES (20, 'RESEARCH', 'DALLAS');

INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES (40, 'OPERATIONS', 'BOSTON');
```

### Q1) List the name of the employees whose salary is greater than that of employee with empno 7566.

### QUERY:

```
SELECT ename FROM EMP WHERE sal > (SELECT sal FROM EMP WHERE empno = 7566);
```

### OUTPUT:



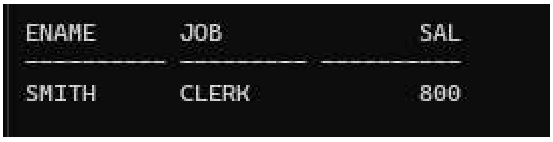
```
ENAME
----------
SCOTT
KING
FORD
```

### Q2) List the ename,job,sal of the employee who get minimum salary in the company.

### QUERY:

```
SELECT ename,job,sal FROM EMP WHERE sal = (SELECT MIN(sal) FROM EMP);
```

### OUTPUT:



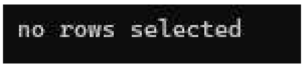
```
ENAME      JOB              SAL
---------- --------- ----------
SMITH      CLERK            800
```

## Q3) List ename, job of the employees who work in deptno 10 and his/her job is any one of the job in the department ‘SALES’.

### QUERY:

```
SELECT ename,job FROM EMP WHERE deptno = 10 AND job IN (SELECT job FROM EMP WHERE
job = 'sales');
```

### OUTPUT:



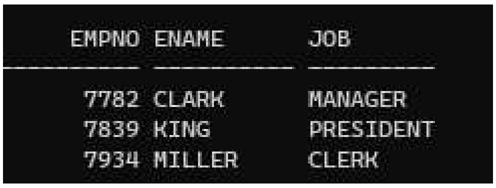
```
no rows selected
```

## Q4) Create a view empv5 (for the table emp) that contains empno, ename, job of the employees who work in dept 10.

### QUERY:

```
create view empv5 as select EMPNO,ENAME,JOB from EMP where DEPTNO=10;
SELECT * FROM empv5;
```

### OUTPUT:



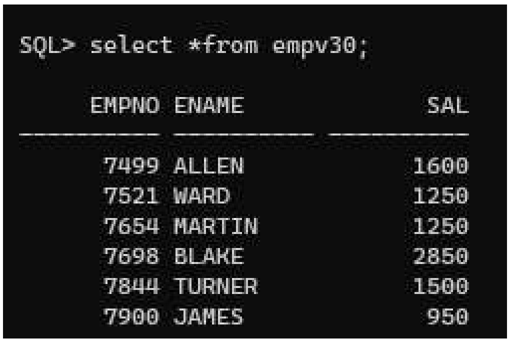
```
     EMPNO ENAME      JOB
---------- ---------- ---------
      7782 CLARK      MANAGER
      7839 KING       PRESIDENT
      7934 MILLER     CLERK
```

## Q5) Create a view with column aliases empv30 that contains empno, ename, sal of the employees who work in dept 30. Also display the contents of the view.

### QUERY:

```
create view empv30 AS select EMPNO,ENAME,SAL from EMP where DEPTNO=30;
```

### OUTPUT:



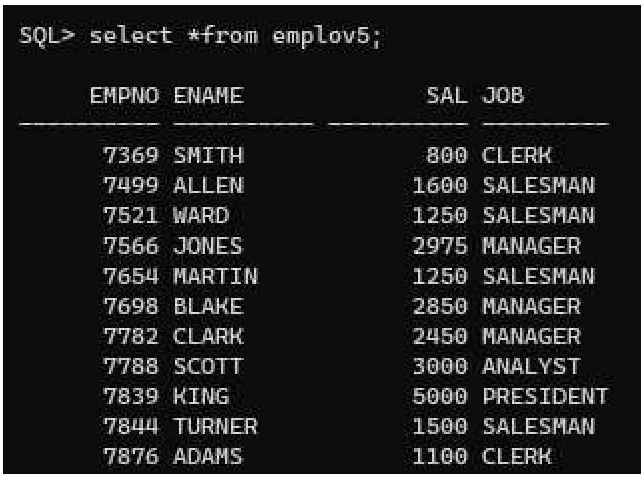
```
SQL> select *from empv30;

     EMPNO ENAME             SAL
---------- ---------- ----------
      7499 ALLEN            1600
      7521 WARD             1250
      7654 MARTIN           1250
      7698 BLAKE            2850
      7844 TURNER           1500
      7900 JAMES             950
```

## Q6) Update the view empv5 by increasing 10% salary of the employees who work as ‘CLERK’. Also confirm the modifications in emp table

### QUERY:

```
update EMP set SAL=SAL*1.1 WHERE JOB='clerk';
create view emplov5 as select EMPNO,ENAME,SAL,JOB from EMP;
```

### OUTPUT:



```
SQL> select *from emplov5;

     EMPNO ENAME             SAL JOB
---------- ---------- ---------- ---------
      7369 SMITH             800 CLERK
      7499 ALLEN            1600 SALESMAN
      7521 WARD             1250 SALESMAN
      7566 JONES            2975 MANAGER
      7654 MARTIN           1250 SALESMAN
      7698 BLAKE            2850 MANAGER
      7782 CLARK            2450 MANAGER
      7788 SCOTT            3000 ANALYST
      7839 KING             5000 PRESIDENT
      7844 TURNER           1500 SALESMAN
      7876 ADAMS            1100 CLERK
```

## Create a Customer1 Table

```
CREATE TABLE Customer1 (customer_id INT,cust_name VARCHAR(20),city VARCHAR(20),grade I
```

## Inserting Values to the Table

```
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3002,
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3007,
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3005,
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3008,
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3004,
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3009,
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3003,
INSERT INTO Customer1 (customer_id, cust_name, city, grade, salesman_id) VALUES(3001,
```

## Create a Salesperson1 table

```
CREATE TABLE Salesman1 (salesman_id INT,name VARCHAR(20),city VARCHAR(20),commission D
```

## Inserting Values to the Table

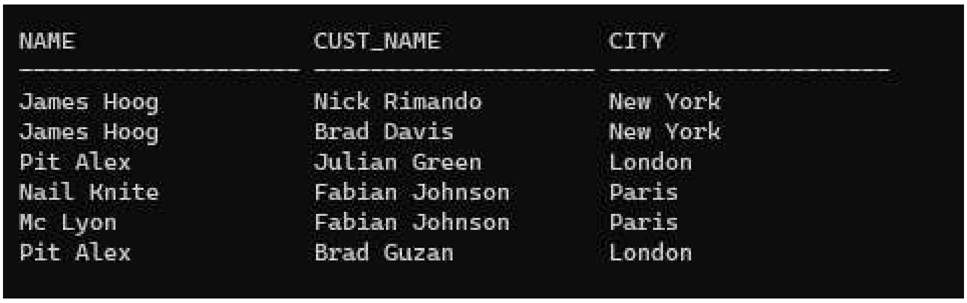
```
INSERT INTO Salesman1 (salesman_id, name, city, commission) VALUES(5001, 'James Hoog',
INSERT INTO Salesman1 (salesman_id, name, city, commission) VALUES(5002, 'Nail Knite',
INSERT INTO Salesman1 (salesman_id, name, city, commission) VALUES(5005, 'Pit Alex', '
INSERT INTO Salesman1 (salesman_id, name, city, commission) VALUES(5006, 'Mc Lyon', 'P
INSERT INTO Salesman1 (salesman_id, name, city, commission) VALUES(5007, 'Paul Adam',
INSERT INTO Salesman1 (salesman_id, name, city, commission) VALUES(5003, 'Lauson Hen',
```

## Q7) Write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust_name and city.

### QUERY:

```
select s.name,c.cust_name,s.city from salesman1 s ,customer1 c where s.city=c.city;
```

### OUTPUT:



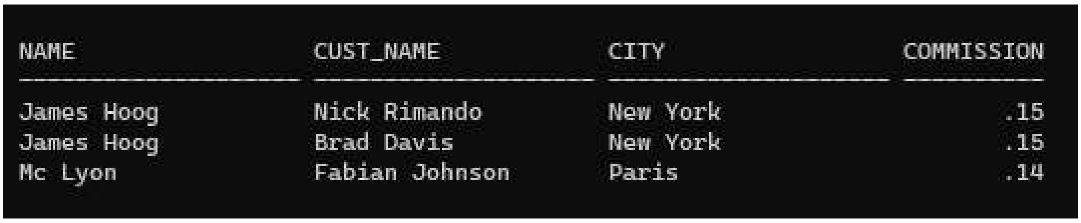
```
NAME                 CUST_NAME            CITY
-------------------- -------------------- --------------------
James Hoog           Nick Rimando         New York
James Hoog           Brad Davis           New York
Pit Alex             Julian Green         London
Nail Knite           Fabian Johnson       Paris
Mc Lyon              Fabian Johnson       Paris
Pit Alex             Brad Guzan           London
```

## Q8) Write a SQL query to find salespeople who received commissions of more than 13 percent from the company. Return Customer Name, customer city, Salesman, commission.

### QUERY:

```
select s.name,c.cust_name,c.city,s.commission from salesman1 s inner join customer1
c on s.city=c.city where s.commission>0.13;
```

### OUTPUT:



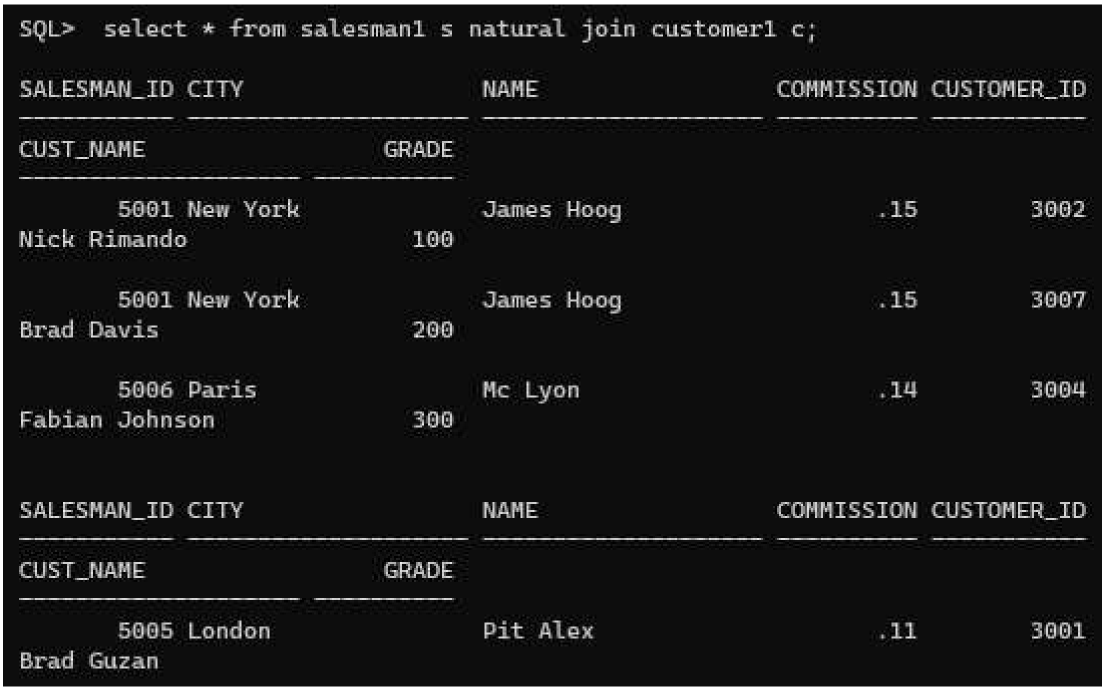
```
NAME                 CUST_NAME            CITY                 COMMISSION
-------------------- -------------------- -------------------- ----------
James Hoog           Nick Rimando         New York                    .15
James Hoog           Brad Davis           New York                    .15
Mc Lyon              Fabian Johnson       Paris                       .14
```

## Q9) Perform Natural join on both tables

### QUERY:

```
select * from salesman1 s natural join customer1 c;
```

### OUTPUT:



```
SQL>  select * from salesman1 s natural join customer1 c;

SALESMAN_ID CITY                 NAME                 COMMISSION CUSTOMER_ID
----------- -------------------- -------------------- ---------- -----------
CUST_NAME                 GRADE
-------------------- ----------
       5001 New York             James Hoog                  .15        3002
Nick Rimando                100

       5001 New York             James Hoog                  .15        3007
Brad Davis                  200

       5006 Paris                Mc Lyon                     .14        3004
Fabian Johnson              300


SALESMAN_ID CITY                 NAME                 COMMISSION CUSTOMER_ID
----------- -------------------- -------------------- ---------- -----------
CUST_NAME                 GRADE
-------------------- ----------
       5005 London               Pit Alex                    .11        3001
Brad Guzan
```

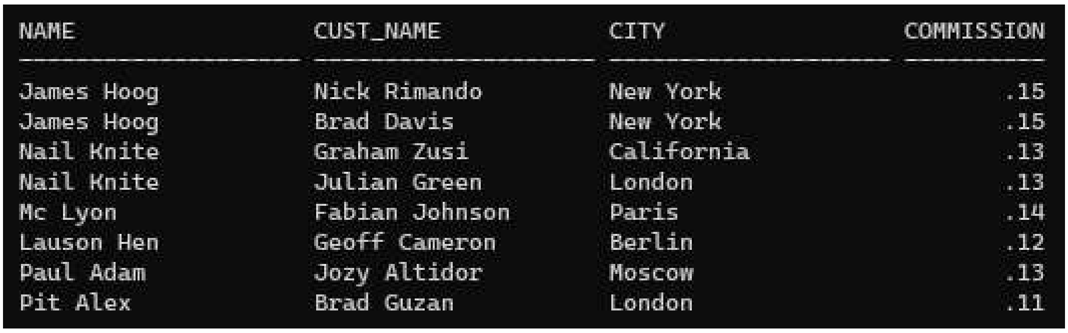
## Q10) Perform Left and right join on both tables

## QUERY:

```
select s.name,c.cust_name,c.city,s.commission from salesman1 s left join customer1 c on s.salesman_id=c.salesman_id;

select s.name,c.cust_name,c.city,s.commission from salesman1 s right join customer1 c on s.salesman_id=c.salesman_id;
```

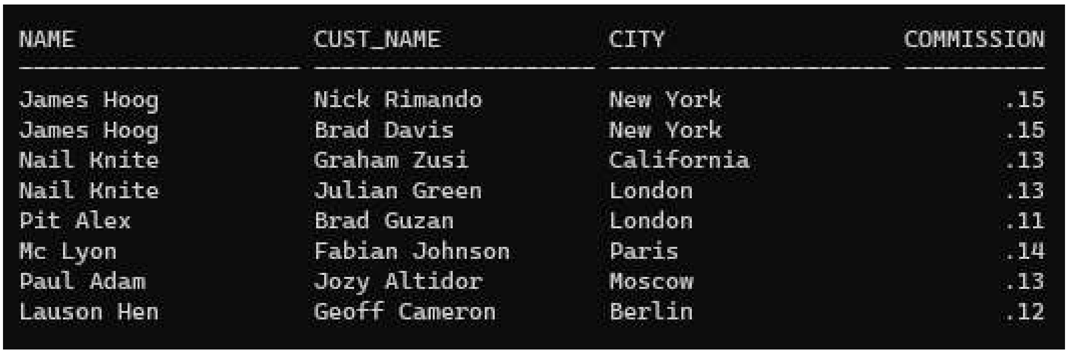
## OUTPUT:

## Left Join:



```
NAME                 CUST_NAME            CITY                 COMMISSION
-------------------- -------------------- -------------------- ----------
James Hoog           Nick Rimando         New York                    .15
James Hoog           Brad Davis           New York                    .15
Nail Knite           Graham Zusi          California                  .13
Nail Knite           Julian Green         London                      .13
Mc Lyon              Fabian Johnson       Paris                       .14
Lauson Hen           Geoff Cameron        Berlin                      .12
Paul Adam            Jozy Altidor         Moscow                      .13
Pit Alex             Brad Guzan           London                      .11
```

## Right Join:



```
NAME                 CUST_NAME            CITY                 COMMISSION
-------------------- -------------------- -------------------- ----------
James Hoog           Nick Rimando         New York                    .15
James Hoog           Brad Davis           New York                    .15
Nail Knite           Graham Zusi          California                  .13
Nail Knite           Julian Green         London                      .13
Pit Alex             Brad Guzan           London                      .11
Mc Lyon              Fabian Johnson       Paris                       .14
Paul Adam            Jozy Altidor         Moscow                      .13
Lauson Hen           Geoff Cameron        Berlin                      .12
```

## Result:

Thus, a database is created and implementation of views,subqueries and joins is executed successfully.