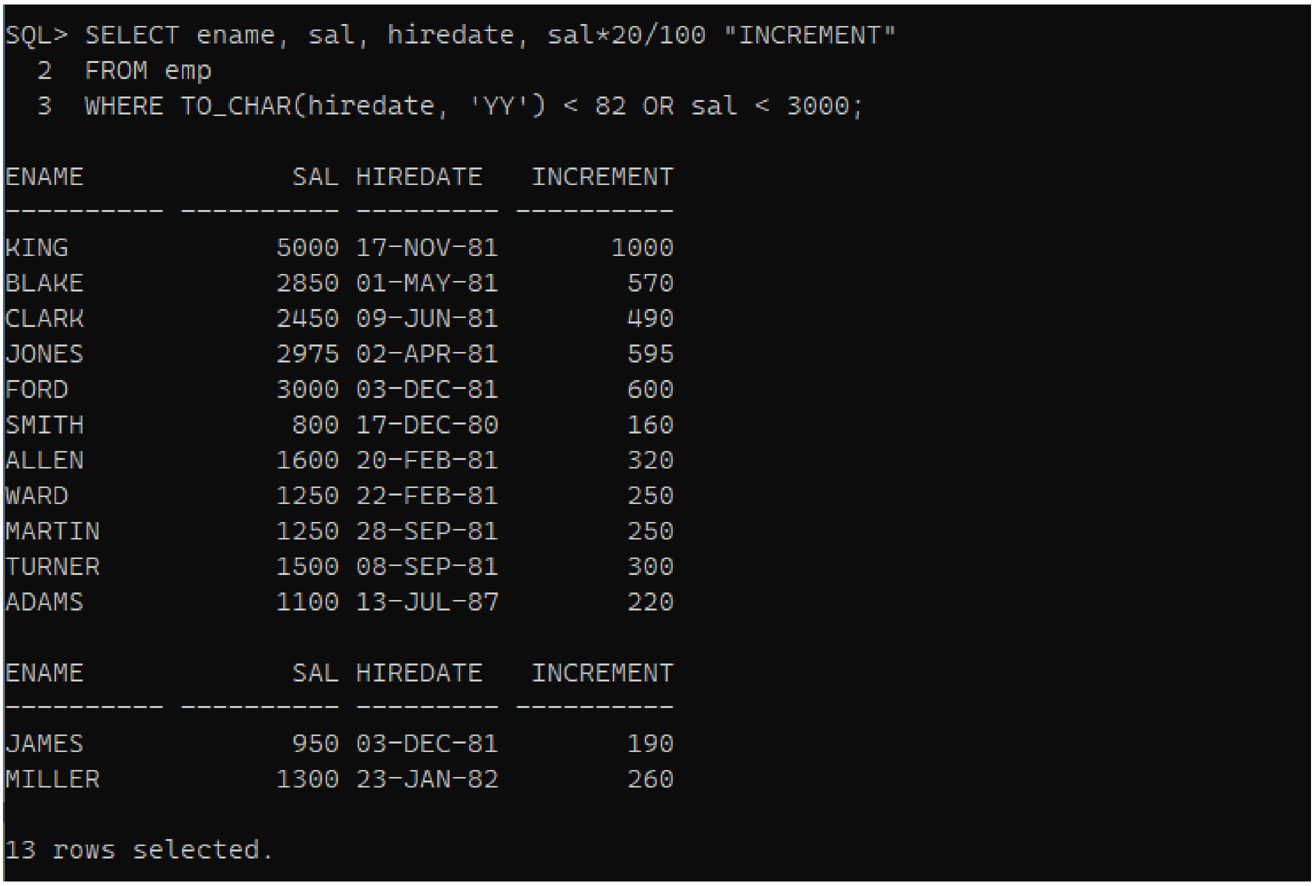
SP_Nov04

```sql
create table dept(deptno number(2,0), dname varchar2(14), loc varchar2(13),
constraint pk_dept primary key (deptno));


create table emp(empno number(4,0), ename varchar2(10), job varchar2(9), mgr
number(4,0), hiredate date, sal number(7,2), comm number(7,2), deptno
number(2,0), constraint pk_emp primary key (empno), constraint fk_deptno foreign
key (deptno) references dept (deptno));
insert into DEPT (DEPTNO, DNAME, LOC) values(10, 'ACCOUNTING', 'NEW YORK');
insert into dept values(20, 'RESEARCH', 'DALLAS');
insert into dept values(30, 'SALES', 'CHICAGO');
insert into dept values(40, 'OPERATIONS', 'BOSTON');

insert into emp values(7839, 'KING', 'PRESIDENT', null, '17-NOV-1981', 5000,
null, 10);
insert into emp values(7698, 'BLAKE', 'MANAGER', 7839, '1-MAY-1981', 2850, null,
30);
insert into emp values(7782, 'CLARK', 'MANAGER', 7839, '09-JUN-1981', 2450,
null, 10);
insert into emp values(7566, 'JONES', 'MANAGER', 7839, '2-APR-1981', 2975, null,
20);
insert into emp values(7788, 'SCOTT', 'ANALYST', 7566, '13-JUL-1987', 3000,
null, 20);
insert into emp values(7902, 'FORD', 'ANALYST', 7566, '3-DEC-1981', 3000, null,
20);
insert into emp values(7369, 'SMITH', 'CLERK', 7902, '17-DEC-1980', 800, null,
20);
insert into emp values(7499, 'ALLEN', 'SALESMAN', 7698, '20-FEB-1981', 1600,
300, 30);
insert into emp values(7521, 'WARD', 'SALESMAN', 7698, '22-FEB-1981', 1250, 500,
30);
insert into emp values(7654, 'MARTIN', 'SALESMAN', 7698, '28-SEP-1981', 1250,
1400, 30);
insert into emp values(7844, 'TURNER', 'SALESMAN', 7698, '8-SEP-1981', 1500, 0,
30);
insert into emp values(7876, 'ADAMS', 'CLERK', 7788, '13-JUL-1987', 1100, null,
20);
insert into emp values(7900, 'JAMES', 'CLERK', 7698, '3-DEC-1981', 950, null,
30);
insert into emp values(7934, 'MILLER', 'CLERK', 7782, '23-JAN-1982', 1300, null,
10);
```

(Continuation from Oct 28)

Question 10: Give an increment of 20% to all employees joined before 1/1/82 or earner less than Rs. 3000.

```
SELECT ename, sal, hiredate, sal*20/100 "INCREMENT"
        FROM emp
        WHERE TO_CHAR(hiredate, 'YY') < 82 OR sal < 3000;
```

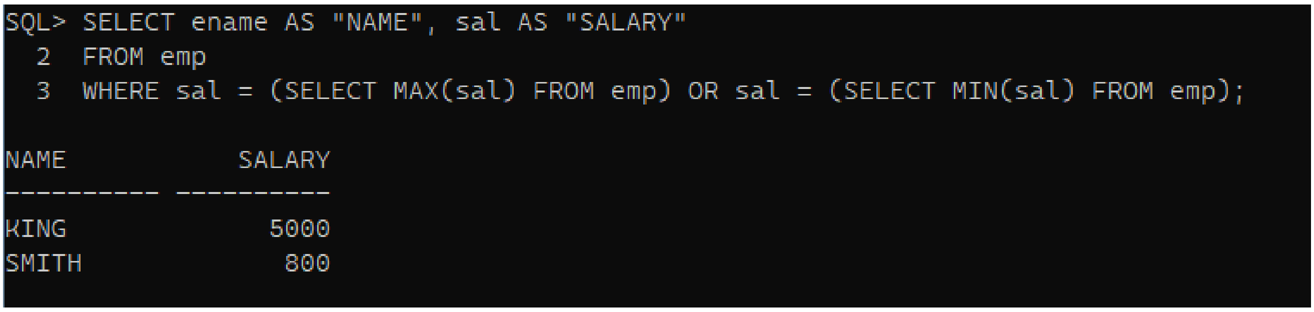


```
SQL> SELECT ename, sal, hiredate, sal*20/100 "INCREMENT"
  2  FROM emp
  3  WHERE TO_CHAR(hiredate, 'YY') < 82 OR sal < 3000;

ENAME             SAL HIREDATE   INCREMENT
---------- ---------- --------- ----------
KING             5000 17-NOV-81       1000
BLAKE            2850 01-MAY-81        570
CLARK            2450 09-JUN-81        490
JONES            2975 02-APR-81        595
FORD             3000 03-DEC-81        600
SMITH             800 17-DEC-80        160
ALLEN            1600 20-FEB-81        320
WARD             1250 22-FEB-81        250
MARTIN           1250 28-SEP-81        250
TURNER           1500 08-SEP-81        300
ADAMS            1100 13-JUL-87        220

ENAME             SAL HIREDATE   INCREMENT
---------- ---------- --------- ----------
JAMES             950 03-DEC-81        190
MILLER           1300 23-JAN-82        260

13 rows selected.
```

Question 11: Display the name and salary of the employees who receive maximum and minimum salary in one row with proper column heading.

- Not in one row:

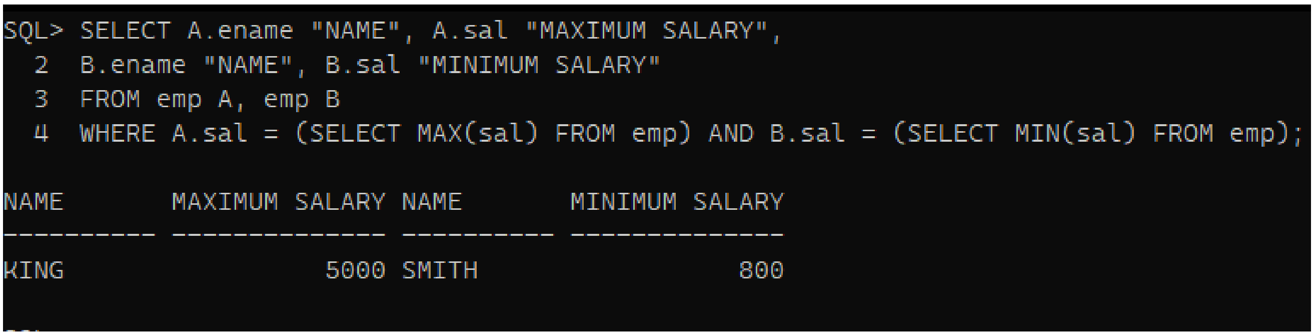
```
SELECT ename AS "NAME", sal AS "SALARY"
        FROM emp
        WHERE sal = (SELECT MAX(sal) FROM emp) OR sal = (SELECT MIN(sal) FROM
emp);
```



```
SQL> SELECT ename AS "NAME", sal AS "SALARY"
  2  FROM emp
  3  WHERE sal = (SELECT MAX(sal) FROM emp) OR sal = (SELECT MIN(sal) FROM emp);

NAME           SALARY
---------- ----------
KING             5000
SMITH             800
```

- In one row:

```
SELECT A.ename "NAME", A.sal "MAXIMUM SALARY",
        B.ename "NAME", B.sal "MINIMUM SALARY"
        FROM emp A, emp B
        WHERE A.sal = (SELECT MAX(sal) FROM emp) AND B.sal = (SELECT MIN(sal)
FROM emp);
```

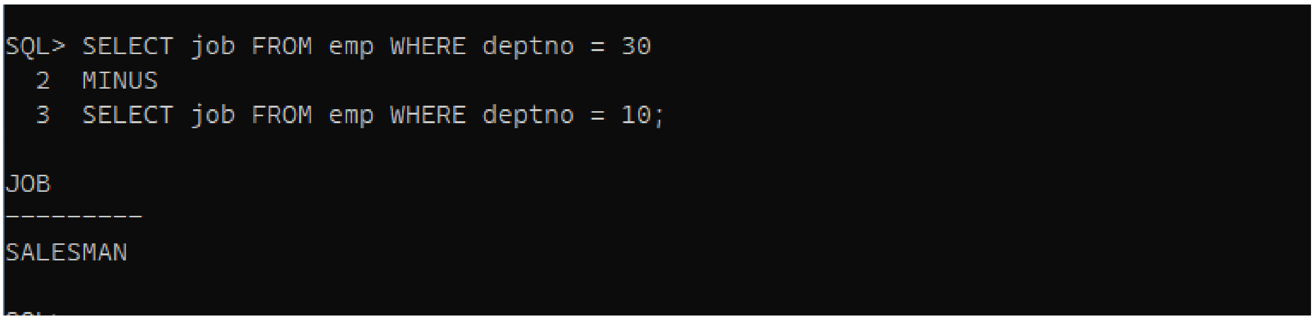


```
SQL> SELECT A.ename "NAME", A.sal "MAXIMUM SALARY",
  2  B.ename "NAME", B.sal "MINIMUM SALARY"
  3  FROM emp A, emp B
  4  WHERE A.sal = (SELECT MAX(sal) FROM emp) AND B.sal = (SELECT MIN(sal) FROM emp);

NAME       MAXIMUM SALARY NAME       MINIMUM SALARY
---------- -------------- ---------- --------------
KING                 5000 SMITH                 800
```

- This is an example of a self-join operation, where a Certesian product is taken with itself.

Question 12: Find the job that exists in dept. no. 30 but not in dept. no. 10.

```
SELECT job FROM emp WHERE deptno = 30
        MINUS
        SELECT job FROM emp WHERE deptno = 10;
```

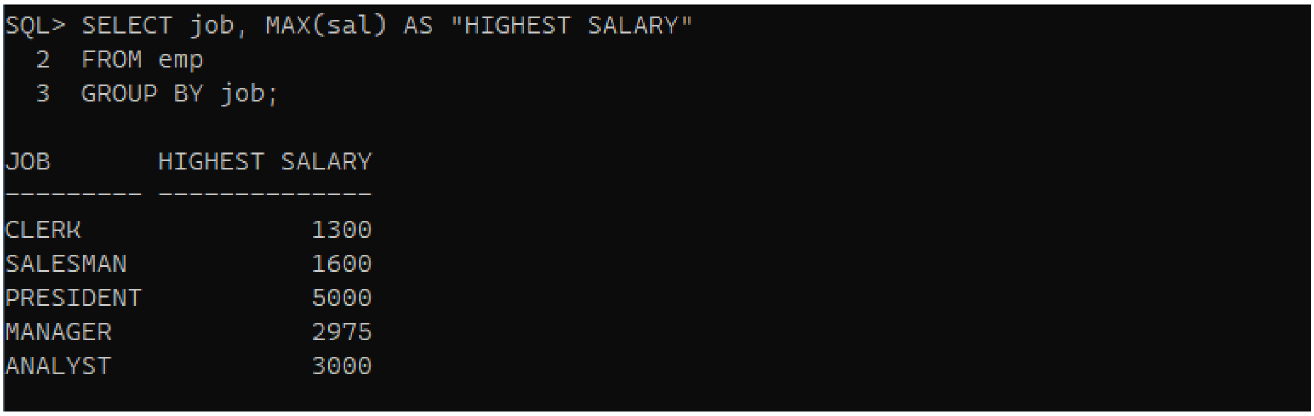


```
SQL> SELECT job FROM emp WHERE deptno = 30
  2  MINUS
  3  SELECT job FROM emp WHERE deptno = 10;

JOB
---------
SALESMAN
```

- Note the `MINUS` clause used here.

Question 13: Find the highest salary in each job type.

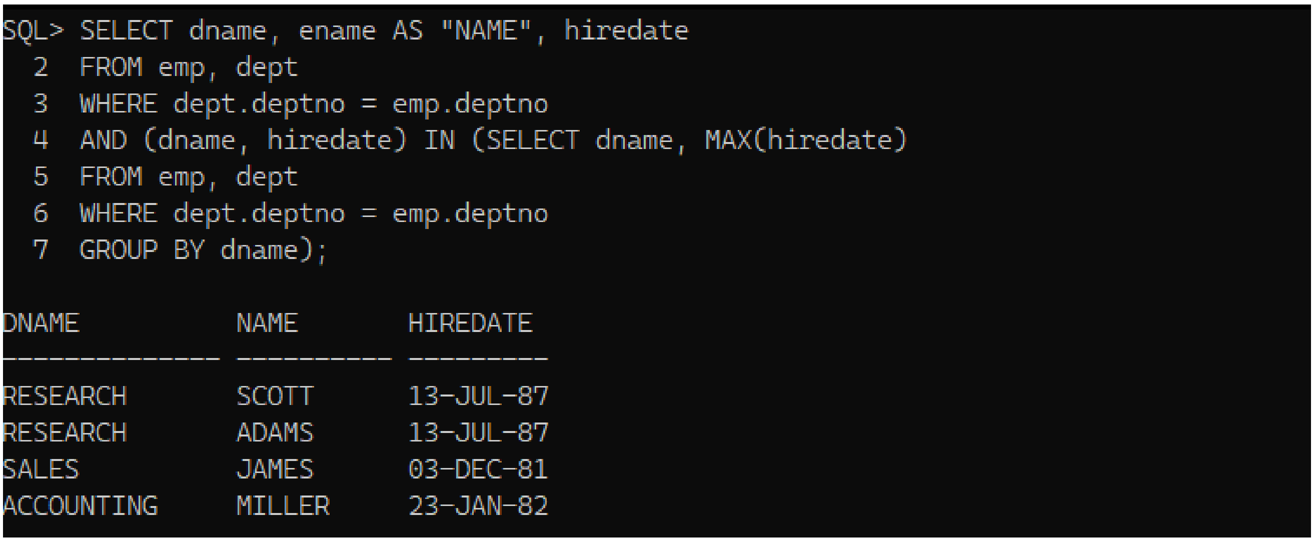
```
SELECT job, MAX(sal) AS "HIGHEST SALARY"
        FROM emp
        GROUP BY job;
```



```
SQL> SELECT job, MAX(sal) AS "HIGHEST SALARY"
  2  FROM emp
  3  GROUP BY job;

JOB       HIGHEST SALARY
--------- --------------
CLERK               1300
SALESMAN            1600
PRESIDENT           5000
MANAGER             2975
ANALYST             3000
```

Question 14: Find the most recently hired employees in each department.

```
SELECT dname, ename AS "NAME", hiredate
        FROM emp, dept
        WHERE dept.deptno = emp.deptno
        AND (dname, hiredate) IN (SELECT dname, MAX(hiredate)
                FROM emp, dept
                WHERE dept.deptno = emp.deptno
        GROUP BY dname);
```



```
SQL> SELECT dname, ename AS "NAME", hiredate
  2  FROM emp, dept
  3  WHERE dept.deptno = emp.deptno
  4  AND (dname, hiredate) IN (SELECT dname, MAX(hiredate)
  5  FROM emp, dept
  6  WHERE dept.deptno = emp.deptno
  7  GROUP BY dname);

DNAME          NAME       HIREDATE
-------------- ---------- ---------
RESEARCH       SCOTT      13-JUL-87
RESEARCH       ADAMS      13-JUL-87
SALES          JAMES      03-DEC-81
ACCOUNTING     MILLER     23-JAN-82
```

Create the following tables:

1. Borrower(Id: data type = varchar, size = 3, primary key
   Name: data type = varchar, size = 20)

```
CREATE TABLE borrower(Id VARCHAR(3), name VARCHAR(20), CONSTRAINT pk_bowworer
PRIMARY KEY (Id));
```

2. Book(Id: data type = varchar, size = 3, primary key
   Title: data type = varchar, size = 20

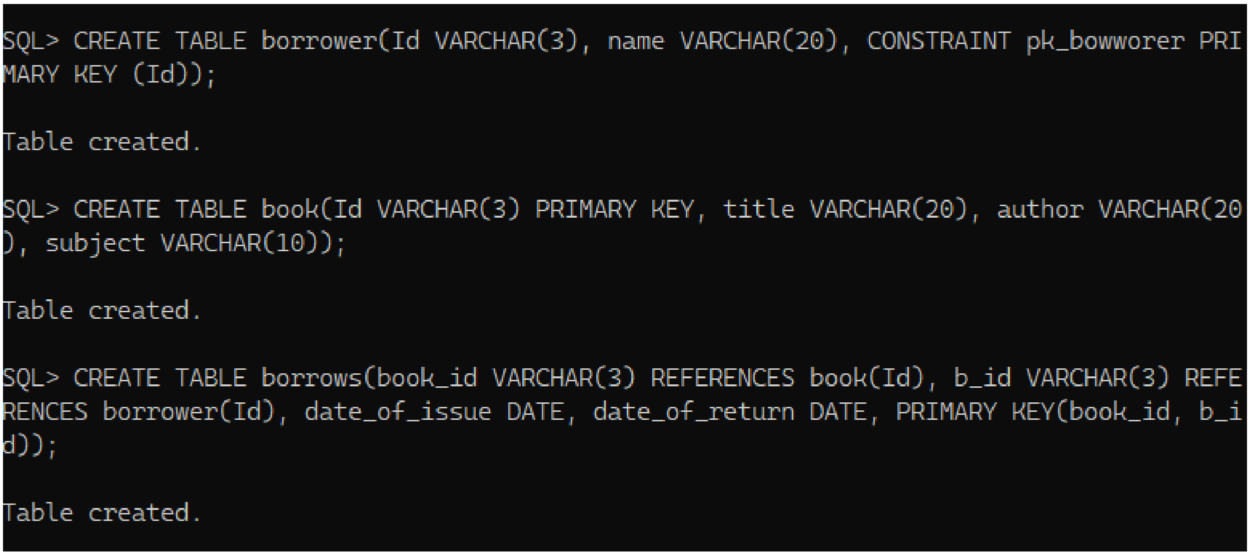
Author: data type = varchar, size = 20
Subject: data type = varchar, size = 10)

```
CREATE TABLE book(Id VARCHAR(3) PRIMARY KEY, title VARCHAR(20), author
VARCHAR(20), subject VARCHAR(10));
```

3. Borrows(Book_Id: data type = varchar, size = 3, key attribute, foreign key depends on Id attribute of Book table
B_Id: data type = varchar, size = 3, key attribute, foreign key depends on Id attribute of Borrower table
Data_of_Issue: data type = date
Date_of_return: data type = date)

```
CREATE TABLE borrows(book_id VARCHAR(3) REFERENCES book(Id), b_id VARCHAR(3)
REFERENCES borrower(Id), date_of_issue DATE, date_of_return DATE, PRIMARY
KEY(book_id, b_id));
```

- *Here the primary key has more than one attribute, hence has to be declared in table level.*



```
SQL> CREATE TABLE borrower(Id VARCHAR(3), name VARCHAR(20), CONSTRAINT pk_bowworer PRI
MARY KEY (Id));

Table created.

SQL> CREATE TABLE book(Id VARCHAR(3) PRIMARY KEY, title VARCHAR(20), author VARCHAR(20
), subject VARCHAR(10));

Table created.

SQL> CREATE TABLE borrows(book_id VARCHAR(3) REFERENCES book(Id), b_id VARCHAR(3) REFE
RENCES borrower(Id), date_of_issue DATE, date_of_return DATE, PRIMARY KEY(book_id, b_i
d));

Table created.
```

4. Borrower table data:

Id	NAME
BR1	Sohini Rai
BR2	Suman Chandra
BR3	Karan Doshi

(not complete)