# SP_Oct28

## FYI (Example 4, for today)

```
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);
```

- The expression `NUMBER(n, m)` specifies decimal formatting.
- Notice the *table-level-constrain*.

---

## Question 1: List tname of the employee who earns the minimum salary.

```
SELECT ename FROM emp WHERE sal = (SELECT MIN(sal) FROM emp);
```

```
SQL> SELECT ename FROM emp WHERE sal = (SELECT MIN(sal) FROM emp);

ENAME
----------
SMITH
```

- Here, the aggregate function `MIN` is used. remember, there are 5 aggregate functions.

## Question 2: List all employees who work in the same post as Smith.

```
SELECT ename FROM emp WHERE job = (SELECT job FROM emp WHERE ename = 'SMITH') AND
ename <> 'SMITH';
```

```
SQL> SELECT ename FROM emp WHERE job = (SELECT job FROM emp WHERE ename = 'SMITH') AND
 ename <> 'SMITH';

ENAME
----------
ADAMS
JAMES
MILLER
```

- The `=` is a relational equals-to operator.
- The `<>` is a relational not-equals-to operator.

## Question 3: List all employees who earn more than every employee in the 'Sales' dept.

```
SELECT ename FROM emp WHERE sal > (SELECT MAX(sal) FROM emp, dept WHERE emp.deptno =
dept.deptno AND dname = 'SALES');
```

```
SQL> SELECT ename FROM emp WHERE sal > (SELECT MAX(sal) FROM emp, dept WHERE emp.deptn
o = dept.deptno AND dname = 'SALES');

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

- `WHERE emp.deptno = dept.deptno` helps to temporarily join the two tables using primary key-foreign keys.

## Question 4: Find the job with the highest average salary.

```
SELECT A.job
FROM (SELECT job, AVG(sal) as avgsal
        FROM emp
        GROUP BY job)A,
        (SELECT MAX(AVG(sal)) AS maxsal
```

```
        FROM emp GROUP BY job)B
        WHERE A.avgsal = B.maxsal;
```

```
SQL> SELECT A.job
  2  FROM (SELECT job, AVG(sal) as avgsal
  3  FROM emp
  4  GROUP BY job)A,
  5  (SELECT MAX(AVG(sal)) AS maxsal
  6  FROM emp GROUP BY job)B
  7  WHERE A.avgsal = B.maxsal;

JOB
---------
PRESIDENT
```

- 'A' is used as an alias to refer to what is returned by `(SELECT job, AVG(sal) as avgsal FROM emp GROUP BY job)`; which is a temporary table with two columns: job and average salary.

## Question 5: Find the highest salary of each job.

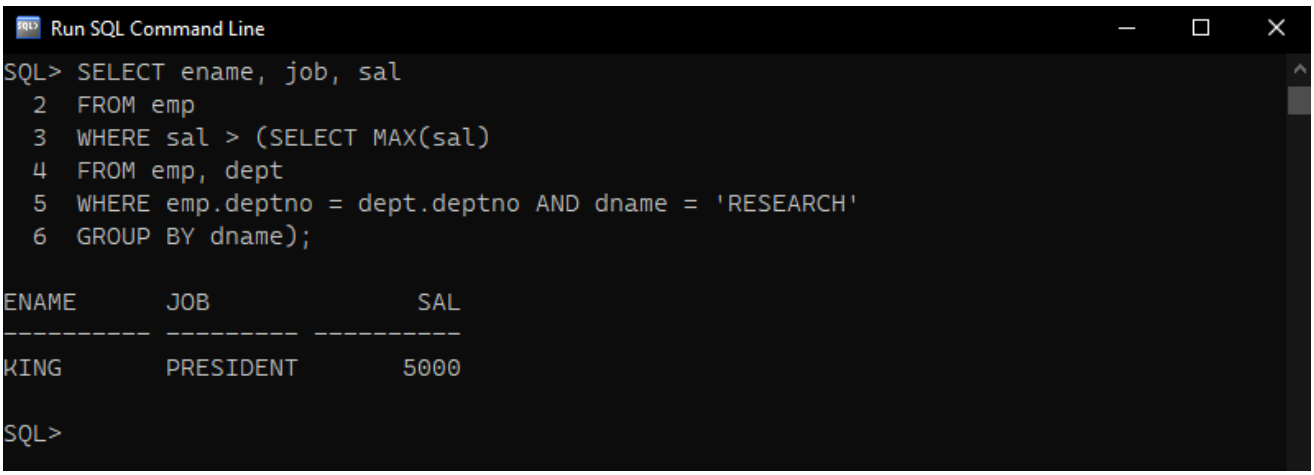
```
SELECT job, MAX(sal) as "HIGHEST SALARY"
FROM emp
GROUP BY job;
```

```
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 6: List the names, jobs and salaries of employees whose salary is greater than the highest salary in Research department.

```
SELECT ename, job, sal
FROM emp
WHERE sal > (SELECT MAX(sal)
                FROM emp, dept
                WHERE emp.deptno = dept.deptno AND dname = 'RESEARCH');
```




- The query also works without the `GROUP BY dname` part.

## Question 7: Find the department that is no having any emmployee.

```
SELECT dname
FROM dept
WHERE NOT EXISTS (SELECT job FROM emp WHERE emp.deptno = dept.deptno);
```

```
SQL> SELECT dname
  2  FROM dept
  3  WHERE NOT EXISTS (SELECT job FROM emp WHERE emp.deptno = dept.deptno);

DNAME
--------------
OPERATIONS
```

## Question 8: List the top 3 earners in the organisation.

```
SELECT E.ename, E.sal
FROM (SELECT *
          FROM emp
          ORDER BY sal DESC)E
          WHERE rownum < 4;
```

```
SQL> SELECT E.ename, E.sal
  2  FROM (SELECT *
  3    FROM emp
  4    ORDER BY sal DESC)E
  5    WHERE rownum < 4;

ENAME             SAL
---------- ----------
KING              5000
SCOTT             3000
FORD              3000
```

## Question 9: List the years and the number of people joining in that year.

```
SELECT TO_CHAR(hiredate, 'YYYY') AS "year", COUNT(ename) "NO OF EMPLOYEE"
FROM emp
GROUP BY TO_CHAR(hiredate, 'YYYY');
```

```
SQL> SELECT TO_CHAR(hiredate, 'YYYY') AS year, COUNT(ename) "NO OF EMPLOYEE"
  2  FROM emp
  3  GROUP BY TO_CHAR(hiredate, 'YYYY');

YEAR NO OF EMPLOYEE
---- --------------
1987              2
1980              1
1982              1
1981             10
```

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

```

```