1. **Display name of employees , department name and job name for each employee**

SQL> select ename,job,DNO from sapna\_emp;

ENAME JOB DNO

----------- -------- --------

mr.smith clerk 20

mr.allen salesman 30

mr.ward salesman 30

mr.martin salesman 30

mr.blake manager 30

mr.clark manager 10

mr.jones manager 20

mr.kong president 10

8 rows selected.

1. **Display the department name along with no of employees and average salary of that department**

SQL> select DNO,count(empno),avg(sal) from sapna\_emp group by DNO;

DNO COUNT(EMPNO) AVG(SAL)

------- ---------------------- ---------------

30 4 1737.5

20 2 1887.5

10 2 3725

1. **For each department, find out no. of jobs the employees are assigned to.**

SQL> select DNO,count(job) as number\_of\_jobs from sapna\_emp group by DNO

DNO NUMBER\_OF\_JOBS

---------- ----------------------------

30 4

20 2

10 2

1. **Check for correctness of the above queries in terms of count, if you want to bring in all entries, would you achieve the same?**

SQL> select job,count(empno) as no\_of\_jobs from sapna\_emp group by job;

JOB NO\_OF\_JOBS

------ --------------------------

salesman 3

clerk 1

president 1

manager 3

SQL> select DNO,job,count(empno) as number\_of\_jobs from sapna\_emp group by DNO,job;

DNO JOB NUMBER\_OF\_JOBS

------- ------- ------------------------

10 manager 1

20 clerk 1

30 manager 1

20 manager 1

30 salesman 3

10 president 1

6 rows selected.

1. **Group by the employees based on the first character of employee first name. Display the results in alphabetic order (descending) of first character.**

SQL> SELECT SUBSTR(ename, 1) as first\_char, count(\*) as no\_of\_emp from sapna\_emp group by SUBSTR(ename,1) order by first\_char desc;

FIRST\_CHAR NO\_OF\_EMP

-------------------- ----------------------

mr.ward 1

mr.smith 1

mr.martin 1

mr.kong 1

mr.jones 1

mr.clark 1

mr.blake 1

mr.allen 1

8 rows selected.

1. **Display name of those employees who get a salary more than the avg salary**

SQL>SELECT ename, sal FROM sapna\_emp WHERE sal > (SELECT AVG(sal) FROM sapna\_emp);

ENAME SAL

--------- -------

mr.blake 2850

mr.clark 2450

mr.jones 2975

mr.kong 5000

1. **Display name of the all the employees who are ‘manager’, except the one who gets the minimum salary.**

SQL> select ename as name from sapna\_emp where job='manager' and sal>(select min(sal) from sapna\_emp where job='manager');

NAME

----------

mr.blake

mr.jones

1. **Display firstname, lastname, salary of those sales representatives who earns a higher salary than the minimum salary a sales manager receives.**

SQL> select ename,sal from sapna\_emp where job='manager' and sal>(select min(sal) from sapna\_emp where job='salesman');

ENAME SAL

------------ -------

mr.blake 2850

mr.clark 2450

mr.jones 2975

1. **Display the name of the employees/employee who gets the second highest salary. (sub query)**

SQL> select ename from sapna\_emp where sal=(select max(sal) from sapna\_emp where sal<(select max(sal) from sapna\_emp));

ENAME

-------------

mr.jones

1. **Come up with the query for previous question using set operators**

SELECT ename, sal FROM sapna\_emp WHERE sal = ( SELECT MAX(sal) FROM employees WHERE sal < ( SELECT MAX(sal) FROM sapna\_emp ) );

ENAME

-------------

mr.jones

1. **Display the name of the employee (manager) who has the maximum no. of employees reporting to Him.**

SQL>select ename from sapna\_emp where job = 'manager' AND emp\_under=max(emp\_under);

ENAME

------------

mr.clark

1. **Display the name of those employees, who are in the same department as Timothy Gates and gets a salary more than the average salary of all the employees**

SQL> SELECT ename FROM sapna\_emp WHERE DNO = (SELECT DNO FROM sapna\_emp WHERE ename = 'Mr. Blake') AND sal > ( SELECT AVG(sal) FROM sapna\_emp);

no rows selected

1. **If an employee has spent less than 5 years then he is considered entry level id 5 – 10 then midlevel else a senior employee. Write a query, which will label the employees in either of the above categories**

SELECT ename, hiredate, ROUND(MONTHS\_BETWEEN(SYSDATE, hiredate) / 12) AS years\_experience,CASE WHEN ROUND(MONTHS\_BETWEEN(SYSDATE, hiredate) / 12) < 5 THEN 'Entry Level' WHEN ROUND(MONTHS\_BETWEEN(SYSDATE, hiredate) / 12) BETWEEN 5 AND 10

THEN 'Mid Level' ELSE 'Senior'

END AS experience\_level FROM sapna\_emp ;

ENAME HIREDATE YEARS\_EXPERIENCE EXPERIENCE

---------- --------------- ----------------------------- --------------------

mr. smith 17-DEC-80 44 Senior

mr. allen 20-FEB-81 44 Senior

mr. ward 22-FEB-81 44 Senior

mr. jones 02-APR-81 43 Senior

mr. martin 28-SEP-81 43 Senior

mr. blake 01-MAY-81 43 Senior

mr. clark 09-JUN-81 43 Senior

mr. kong 17-NOV-81 43 Senior

8 rows selected.

1. **Write query to find out any departments that are present in department table but does not have employees**

**Ans:SQL> SELECT d.department\_id, d.department\_name FROM department d LEFT JOIN employee e ON d.department\_id = e.department\_id WHERE e.employee\_id IS NULL;**

no rows selected.

1. **Write a query which will display job id , which are present in both job and employee columns SELECT job FROM job\_table INTERSECT SELECT job\_id FROM sapna\_emp;**

JOB

-----

clerk

salesman

manager

president

1. **Increase salary of each employee of all the department who draws the minimum salary by 100$.**

**SQL>UPDATE sapna\_emp e SET sal = sal+ 100 WHERE sal= ( SELECT MIN(sal) FROM sapna\_emp WHERE DNO = e.DNO);**

4 rows updated.

Commit complete.