1. Display the employee ename and sal using **case** and **when**

Ranges less than 1000 as less salary,between 1000 and 2000 as mediumsal,& 2500,3000, and 3500 as special case, if not falling in these three then other sal .

**SQL> select ename,sal, case when sal<1000 then 'low' when sal between 1000 and 2000 then 'medium' when sal in (2500,3000,3500) then 'special' else 'others' end from emp;**

1. Display the employees whose ename 2nd letter would be capital letter ‘LA’ using substring substr().

**SQL> select \* from emp where substr(ename,2,2) like'LA';**

1. Display the employees whose salary more than the ‘SMITH’ salary and less than the ‘BLAKE’salary.

**select \* from emp where sal >(select sal from emp where ename='SMITH') and sal < (select sal from emp where ename='BLAKE');**

1. Display the lowest average salary job.

**SQL> select avg(sal),job from emp group by job having avg(sal)=(select min(av) from (select avg(sal) as av ,job from emp group by job)) ;**

1. Display the deptno, maximum salary and also display those departments having more than 20th department minimum salary.

**SQL> select max(sal),deptno from emp group by deptno;**

**MAX(SAL) DEPTNO**

**---------- ----------**

**2850 30**

**3000 20**

**5000 10**

**SQL> select min(sal),deptno from emp group by deptno having min(sal)>(select min(sal) from emp group by deptno having deptno=20);**

**MIN(SAL) DEPTNO**

**---------- ----------**

**950 30**

**1300 10**

1. Display the employees whose job title same as 10th department job title.

**SQL> select \* from emp where job in (select job from emp where deptno=10);**

1. Convert 18-sep-17 to 18/sep/2017.

**select to\_char(to\_date('18-sep-17'),'dd/mon/yyyy') from dual;**

1. Display the employees who are joining in the month February using to\_char( ).

**SQL> select \* from emp where to\_char(hiredate,'MON')='FEB';**

1. Display those departments having more than 5 employees in them.

**SQL> select deptno from emp group by deptno having count(\*)>5;**

1. Display the deptno and number of employees who are drawing more than 1000 salary and only departments where department wise such employees count must be 3 or more.

**select deptno,count(\*) from (select \* from emp where sal>1000) group by deptno having count(\*)>=3;**

1. Display the employee salary department wise and job wise subtotal and grandtotal by using rollup( ).

**select deptno,job,sum(sal) from emp group by rollup(deptno,job);**

1. Display ename, sal,deptno,dname,loc by joining tables.

**SQL> select ename,sal,e.deptno,dname,loc from emp e,dept d where e.deptno=d.deptno;**

1. Display the employees who are joining before their managers.

**> select e.ename,e.hiredate,m.hiredate from emp e, emp m where e.mgr=m.empno and e.hiredate<m.hiredate**;

1. Print the sum of the salaries of all the employees who are working in department in which at least 2 managers exists.

**select sum(sal) from emp group by deptno having deptno in( select deptno from emp where job='MANAGER' group by deptno having count(\*)>2);**

1. Display the employees who are getting less than the average salary from EMP table.

**SQL> select \* from emp where sal<(select avg(sal) from emp);**

1. Display senior employee details from emp.

**select \* from (select \* from emp order by hiredate asc) where rownum=1;**

1. Display lowest average salary job.

**select job,avg(sal) from emp group by job having avg(sal)=(select min(avg(sal)) from emp group by job);**

1. Display first five highest salaries of employees using ROWNUM.

**> select \* from ( select sal,ename from emp order by sal desc) where rownum<=5;**

1. Display 3rd highest salary employee using ROWNUM.

select \* from ( select sal,ename from emp order by sal desc) where rownum<=3 minus select \* from ( select sal,ename from emp order by sal desc) where rownum<=2;

1. Display the rows of emp between 3 and 9 by using ROWNUM.

select \* from emp where rownum<=9 minus select \* from emp where rownum<3;

1. Display the last two rows from employee table using ROWNUM.

select \* from emp where rownum<=(select count(\*)from emp) minus select \* from emp where rownum<=(select count(\*)-2 from emp);

1. **Display employee who are getting 2nd highest sal in each group.**

**select \* from(select ename,sal,deptno,dense\_rank() over( partition by deptno order by sal desc) t from emp) where t=2;**

1. Display the employees who are getting more than the highest paid employee in 20th department.

select \* from emp where sal>(select max(sal) from emp group by deptno having deptno=20);

1. Display the employees who are getting more than the lowest paid employee in 10th department.

select \* from emp where sal>(select min(sal) from emp group by deptno having deptno=10);

1. Display 3rd highest salary employee using analytical function.

**select \* from (select ename,sal ,dense\_rank() over ( order by sal desc) t from emp)where t=3;**

1. Display the employees whose job, mgr match with the job, mgr of the employee number 7788.

**> select \* from emp where (job,mgr)=(select job,mgr from emp where empno=7788);**

1. Write a query to display ename,deptno,sal of any employee whose deptno,salary match with deptno,salary of the employee who are getting a commission.

**select \* from emp where (deptno,sal) in (select deptno,sal from emp where comm is not null and comm!=0);**

1. Write a query to display ename,dname,salary of any employee,comm. Match with the salary,comm of the employees located in DALLAS.

**select \* from emp where (deptno,NVL(comm,0)) in (select e.deptno,NVL(comm,0) from emp e,dept d where e.deptno=d.deptno and d.loc='DALLAS' );**

1. Display the employees whose salary are less than the average salaries of their job’s.

**select e.\* from emp e,( select avg(sal) a,job from emp group by job) p where e.job=p.job and e.sal<p.a;**

1. Display the first highest salary employee using coreleated subquerys.

**select \* from emp a where 1=(select count(distinct(sal)) from emp b where a.sal<=b.sal);**

1. Display those departments does not have any employees in them using non-coreleated sub-querys.

**select deptno from dept where deptno not in (select distinct(deptno) from emp);**

1. Display those departments does not have any employees in them using correleated sub-querys.

**select deptno from dept d where not exists(select distinct(deptno) from emp e where e.deptno=d.deptno);**

1. Display the last names and hire dates of all latest hires in their respective departments in the location DALLAS.

**select max(hiredate) from emp where deptno=(select deptno from dept where loc='DALLAS');**

1. Print the ename who are not working in any department.

**select ename from emp where deptno is null;**

1. Print employee names who is earning highest salary under each job category.

**select \* from emp where (sal,job) in (select max(sal),job from emp group by job);**

1. Print the employee who are not managers.

**select \* from emp where empno not in(select distinct(NVL(mgr,0)) from emp);**

………………………………………………………………………………………………………

1. Write a pl/sql program to list employee names whose salary is more than their manager salary.

**declare**

**cursor c is select e.\* from emp e,emp m where e.mgr=m.empno and e.sal>m.sal;**

**i emp%rowtype;**

**begin**

**open c;**

**fetch c into i;**

**loop**

**exit when c%notfound;**

**dbms\_output.put\_line(i.ename||i.sal);**

**fetch c into i;**

**end loop;**

**close c;**

**end;**

**/**

1. Write a pl/sql program to list names of employees in alphabetical order along with their position, which is the position of an employee in the list sorted by salary in descending order.

**declare**

**cursor c is select r.\* from (select rownum, ename,sal from (select ename,sal from emp order by sal desc)) r , (select ename from emp order by ename) e where e.ename=r.ename;**

**i c%rowtype;**

**begin**

**open c;**

**open c1;**

**fetch c into i;**

**loop**

**exit when c%notfound;**

**...**

1. Write a pl/sql program to display top 3 earners of the company.

**declare**

**cursor c is select \* from (select ename,sal from emp order by sal desc) where rownum<=3;**

1. Write a trigger on employee table whenever user delete some rows,those rows are stored in another table.

**create or replace trigger t after delete on fake for each row**

**begin**

**insert into fa values('haha',:old.empno);**

**end;**

**/**

1. Write a trigger program to ensure that dept table does not contain duplicate or null values in deptno.

**create or replace trigger t before insert on dept1 for each row**

**declare**

**i number;**

**begin**

**select count(\*) into i from dept where :new.deptno=deptno;**

**if i!=0 then**

**raise\_application\_error(-20015,'duplicat error violate atomicity u fool');**

**end if;**

**end;**

**/**

1. Write a trigger such that salary of an employee must be modified only on Monday in any month with an entry in the log table.

**create or replace trigger t before update of sal on emp1 for each row**

**declare**

**begin**

**if to\_char(sysdate,'dy')!='mon' then**

**insert into fdate values(sysdate);**

**end if;**

**end;**

**/**

1. Write a curor to display the list of employees who are working as a manager.

1. Display first 5 highest salary employees using cursor.

**declare**

**cursor c is select ename,sal from emp order by sal desc;**

**i c%rowtype;**

**begin**

**open c;**

**fetch c into i;**

**loop**

**exit when c%rowcount>5;**

**dbms\_output.put\_line(i.ename||i.sal);**

**fetch c into i;**

**end loop;**

**end;**

**/**

1. Write a pl/sql cursor to transfer ename, sal of employees to another table who are getting more than 2000 salary.
2. Write a pl/sql parameterized cursor program. Display the employees who are working under job as manager and display the employees who are working under analyst.

**declare**

**cursor c(j varchar2) is select ename,sal from emp where job=j;**

**i c%rowtype;**

**begin**

**open c('&j');**

**fetch c into i;**

**loop**

**exit when c%notfound;**

**dbms\_output.put\_line(i.ename||i.sal);**

**fetch c into i;**

**end loop;**

**end;**

**/**

1. Write pl/sql program using %found attribute of cursor whenever user enter the job, count number of these jobs.

**declare**

**cursor c(j varchar2) is select count(\*) as a from emp where job=j;**

**i c%rowtype;**

**begin**

**open c('&j');**

**fetch c into i;**

**loop**

**exit when c%found = false;**

**dbms\_output.put\_line(i.a);**

**fetch c into i;**

**end loop;**

**end;**

**/**

1. Display all employees using %found attribute of cursor ,and use while loop within a cursor.

**declare**

**cursor c(j varchar2) is select count(\*) as a from emp where job=j;**

**i c%rowtype;**

**begin**

**open c('&j');**

**fetch c into i;**

**while c%found=true**

**loop**

**dbms\_output.put\_line(i.a);**

**fetch c into i;**

**end loop;**

**end;**

**/**

1. Write a pl/sql cursor program modify the salaries in employee table based on the following condition.
2. If job=’clerk’ increment salary 100.
3. If job=’salesman’ decrement salary 200.
4. If job=’analyst’ increment salary 300.
5. Write a package i.e ‘emp pack’ with the following procedures
6. To list top workers in each dept.
7. To list designations that are unique to a dept.
8. To list names of employees to which maximum number of employees report.
9. Write a package i.e ‘emp pack1’ with the following procedures
10. To list names of employees who report to an emp with less salary.
11. To list chain of employee-managers.
12. Write a procedure and function to accept a empno and increase sal amount if empno is not found or current salary is null, then raise exception otherwise diplay salary + increment.

**create or replace procedure p(d number) as**

**a varchar2(20);**

**b number;**

**begin**

**select sal into b from emp1 where empno=d;**

**if b is null then**

**raise no\_data\_found;**

**else**

**update emp1 set sal=sal+100 where empno=d;**

**end if;**

**exception when no\_data\_found then**

**raise\_application\_error('errorr');**

**end;**

**/**

1. Develop pl/sql stored procedure for passing empno as parameter to display name of the employee and salary.

**create or replace procedure p(d number) as**

**a varchar2(20);**

**b number;**

**begin**

**select ename,sal into a,b from emp where empno=d;**

**dbms\_output.put\_line(a|| ' '||b);**

**end;**

**/**

1. Develop a stored procedure to insert a record into dept table using formal parameters.

**create or replace procedure p(d number,dn varchar2,lo varchar2) as**

**begin**

**insert into dept values(d,'dn','lo');**

**end;**

**/**

1. Write a pl/sql program to raise pre-defined exception when user try to access ename by entering invalid empno.

**declare**

**i number;**

**begin**

**select sal into i from emp where empno=120;**

**exception**

**when no\_data\_found then**

**dbms\_output.put\_line('its not possible for many rows');**

**end;**

**/**

1. Write a pl/sql program to raise pre-defined exception when user try to access salary of employee to handle multiple values instead of single value expected.

**declare**

**i number;**

**begin**

**select sal into i from emp;**

**exception**

**when too\_many\_rows then**

**dbms\_output.put\_line('its not possible for many rows');**

**end;**

**/**

1. Write a pl/sql program to raise an exception today.

**declare**

**z exception;**

**begin**

**if to\_char(sysdate,'dy')='mon' then**

**raise z;**

**end if;**

**exception**

**when z then**

**dbms\_output.put\_line('its today');**

**end;**

/

1. Write a user defined exception, raise exception if empno is 7902 salary is more than 4000 ,otherwise increase salary by 100.

declare

z exception;

i number;

begin

select sal into i from emp where empno=7902;

if i>4000 then

raise z;

else

update emp1 set sal=sal+100 where empno=7902;

end if;

exception

when z then

dbms\_output.put\_line('his sal is more than 4000');

end;

/

1. Generate form on emp table and apply different functions.
2. Generate report on one of the employee data base table with paper layout