select upper(empname), lower(emailid) from emp;

select initcap('i am there') from dual;

select initcap(empname) from emp;

select concat(empname,' is an Employee') from emp;--fun can work with only 2

select firstname || ' '|| lastname from employee; --operator can works with any

select Email, replace(Email, '.com', '.co.in') from employee;

select \* from employee;

--to get actual replace values in ther table then use update

update emp

set emailid=replace(emailid,'.com','.co.in')

where empid=11;

select length(emailid) from emp;

select Substr('It is very hot today',7,8) from dual;

select Substr('It is very hot today',7) from dual;

select Substr('It is very hot today',-11,5) from dual; --y hot negative

-- indexing start from right hand side and then next 5 char taken

select

Instr('it is very hot','i',1,2)

from dual;

select Emailid, instr(emailid,'@',1,1) from Emp;

desc employee;

select \* from employee;

select \* from emp;

select email,instr(email,'@',1,1) from Employee;

select substr(email, 1 , instr(email,'@',1,1)-1 ) from Employee;

select Mod(27,4)  from dual;

select hiredate,instr(hiredate,'-',1,2) from emp;

-- select sbstr(hiredate,instr(hiredate('-',1,2))+1) from emp;

select trunc(25.90327850000,4) from dual;  --truncate till 4 digits

select trunc(25.90327850000,1) from dual;

select trunc(25.90327850000,0) from dual;

select round(25.90327850000,4) from dual;

select round(25.90327850000,0) from dual;

select sysdate from dual;

select sysdate -7 from dual;

select sysdate +4 from dual;

select add\_Months(sysdate,2) from dual;

select hiredate, add\_months(hiredate,6) from employee; -- date after 6 months

--select hiredate,

--(months\_between(sysdate, hiredate)) from employee;

select hiredate,

  trunc(months\_between(sysdate, hiredate)) from employee; --in readable format

select next\_day(sysdate,'Monday') from dual; -- whenever next monday after date

select next\_day(sysdate,'Thursday') from dual;

select last\_day(sysdate) from dual; --last  day of month

SELECT ROUND(TO\_DATE ('16-SEP-2015'),'MONTH')   "New Month",

ROUND(TO\_DATE ('16-SEP-2015'),'YEAR')   "New Year"

FROM DUAL;

--rounds a date to the first day of the month and year

SELECT TRUNC(TO\_DATE('02-MAR-15','DD-MON-YY'), 'Year')

  "New Year" FROM DUAL;--The value returned is always of datatype DATE

  --The TRUNC (date) function is used to get the date with the time portion

  --of the day truncated to a specific unit of measure.

select \* from employee;

--don't want null values. Instead some default value

select Employeeid, firstname, salary, nvl(manager\_id,0),

nvl(department\_id,0),

NVL(commitionpct,0) from employee;

select \* from employee;

--use of nvl2

select Employeeid, firstname, salary, nvl(manager\_id,0), nvl(department\_id,0), commitionpct from employee;

select Employeeid, firstname, salary, nvl(manager\_id,0), nvl(department\_id,0),

NVL2(commitionpct , salary \* commitionpct  , 0) as comm\_value from employee;

select \* from emp;

select to\_char(sysdate,'Month DD, YY') from dual;

select to\_char(sysdate,'fm Day, Month DD, YYYY') from dual;

select to\_char(sysdate,'fm Ddth Month YY') from dual; -- to remove the space between  2 fields

select to\_char(sysdate,'Ddth Month YY') from dual;

select to\_char(sysdate,'Day, DD Month YYYY HH:MI:SS') from dual;

select salary from Employee;

select to\_char(salary,'99,99,999.99') from Employee; -- to get in proper numeric format

select to\_char(sysdate,'Ddth Month YYYY') from dual;

select to\_char(sysdate,'Ddsp Month YYYY') from dual;

select to\_char(sysdate,'Ddspth Month YYYY') from dual;

select to\_char(sysdate,'Day') from dual;

select to\_char(sysdate,'YYYY') from dual;

select salary from emp; -- to apply format to numbers

select to\_char(salary,'99,99,999.99') from emp;

select to\_date('June 10, 2002','Month DD, YYYY') from dual;-- to get date in dd-mm-yy format

select \* from Employee

Where hiredate = to\_date('June 10, 2002','Month DD, YYYY');  -- if date is not in dd-MON-yyyy format the need to convert

select '$68,000' + 1800 from dual;

select to\_number('$68,000','$99,999') + 1800   from dual;  --conveted amount to the number

--avg and sum workd on numberic values only

select max(salary), min(salary) from employee;

select sum(salary) from employee;

desc emp;

select \* from emp;

select max(salary), min(salary) from emp;

select sum(salary) from emp;

select max(hiredate), min(hiredate) from emp;

select count(\*) from Emp;

--All the group functions ignore null values in the column

select count(manager\_id) from Employee;

desc employee;

select count(employeeid) from employee; -- but the count have variations as given below

select count(\*) from employee;

select count(distinct(department\_id)) from employee; --distinct is used to get unique values

select Department\_Id, count(employeeid)

from employee

Group by Department\_Id; --to segregate the data

select Department\_Id, max(salary), min(salary), sum(salary)

from employee

Group by Department\_Id;

select job\_id, max(salary)

from employee

Group by Job\_id;

select Department\_Id, job\_id, sum(salary)

from employee

Group by Department\_Id, job\_id  --can have multiple columns for group by clause

order by Department\_id desc,job\_id;

select Department\_Id, sum(salary) tot\_salary\_paid

from employee

Group by Department\_Id

having department\_id=10;

--Having sum(salary)> 95000;

select Department\_Id as deptid, sum(salary) as sum\_of\_sal

from employee

where manager\_id is not null

group by department\_id

Having sum(salary)> 95000; --apply filter condition depending on select only not on other columns

-- can't use column alias in where, group by and having clause

select department\_id, sum(salary) as total\_sal    -- column list

from Employee                                     -- table names

Where department\_id is not null                   -- row level filter

group by department\_id                            -- group the data

Having sum(salary)>=95000                         -- group level filter

order by total\_sal;    -- sort the data

-- need to remember sequence on 6 clauses in the query

select \* from Employee;

--simple expression

select Employeeid, firstname, lastname, job\_id, department\_id, salary,

CASE Job\_id

When 'IT\_REP' Then salary+4000

When 'SALES\_REP' then salary+3000

When 'HR' Then salary+1000

ELSE  Salary

END as caculated\_salary

from Employee;

----Searched case Expression

select Employeeid, firstname, lastname, job\_id, department\_id, salary,

CASE

When job\_id = 'IT\_REP' and Department\_id=20 Then salary+4000 -- applied condition on multiple columns

When job\_id= 'SALES\_REP' then salary+3000

ELSE  Salary

END as caculated\_salary

from Employee;