1. Queries to facilitate acquaintance of Built-In Functions, String Functions, Numeric

Functions, Date Functions and Conversion Functions.

SQL FUNCTIONS:

SQL Functions are used to perform m calculations on data. Manipulate output from groups of rows. It can also format date members for display. It can also used for modifying individual data items. SQL function sometimes takes arguments and always returns value.

There are two distinct types of functions:

1. Single Row functions
2. Multiple Row functions

Single Row functions:

Single row functions operate on single rows only and return one result per row.

The different type single row functions are

1. Character functions
2. Number Functions
3. Date functions.
4. Conversion Function
5. General Function

1 .Character Functions:

Upper	Returns char with all letters into upper case
lower	Converts the mixed case or uppercase character strings to lowercase
Initcap	Converts the first letter of each word to upper case and remaining letters to lowercase
Concat	Joins values together you are limited to two arguments with concat

Substr	This extracts a string of determined length
Length	Shows the length of a string as a numeric value
Instr	Finds numeric position of named character
Lpad	Pads the character value right justified
rpad	Pads the character value left justified
Trim	Trims heading or trailing characters from a character string
Raplace	To replace a set of character (String based)
Translate	Change a character to a new described character(character based)

```
SQL> select upper ('oracle') "UPPER" from dual;
UPPER
------
ORACLE

SQL> select lower ('ORACLE') "LOWER" from dual;
LOWER
------
Oracle

SQL> select initcap('sql functions') "initcap" from dual;
initcap
-------------
Sql Functions

SQL> select concat('sql','functions') from dual;

    CONCAT('SQL'
    ------------
    Sqlfunctions

    SQL> select substr('sqlfunctions',1,5) from dual;
    SUBST
    -----
```

```
sqlfu

SQL> select substr('sqlfunctions',2,5) from dual;
SUBST
-----
qlfun
SQL> select substr('sqlfunctions',4,5) from dual;
SUBST
-----
funct

SQL> select length('sqlfunctions') from dual;

LENGTH('SQLFUNCTIONS')
----------------------
               12




SQL> select instr('sqlfunctions','f') from dual;
INSTR('SQLFUNCTIONS','F')
-------------------------
                  4

SQL> select lpad(sal,15,'*') from emp;
LPAD(SAL,15,'*'
---------------
************800
***********1600
***********1250
***********2975
***********1250
***********2850
***********2450
***********3000
***********5000
***********1500
***********1100
************950
***********3000
```

```
***********1300

SQL> select rpad(sal,15,'*') from emp;
RPAD(SAL,15,'*'
---------------
800************
1600***********
1250***********
2975***********
1250***********
2850***********
2450***********
3000***********
5000***********
1500***********
1100***********
950************
3000***********
1300***********


SQL> select trim('s' from 'ssmiths') from dual;
TRIM
----
mith
SQL> select ltrim('ssmiths','s') from dual;
LTRIM
-----
miths


SQL> select rtrim('ssmiths','s') from dual;
RTRIM(
------
ssmith
SQL> select replace('jack and jue','j','bl') from dual;
REPLACE('JACKA
--------------
black and blue
SQL> select translate('jack','j','b') from dual;
TRAN
```

----BACK

Number Functions:

Round	Rounds the value to specified decimal
Trunc	Truncates the column, expression, or value to n decimal places
Power	Calculates the power of the given value
Mod	Finds the remainder of value1 divided by value1
Ceil	Takes the height decimal value
Floor	Takes the lowest decimal value

```
SQL> select round(35.823,2), round(35.823,0), round(35.823,-1) from dual;
ROUND(35.823,2) ROUND(35.823,0) ROUND(35.823,-1)
--------------- --------------- ----------------
          35.82              36               40

SQL> select trunc(35.823,2), trunc(35.823), trunc(35.823,-2) from dual;
TRUNC(35.823,2) TRUNC(35.823) TRUNC(35.823,-2)
--------------- ------------- ----------------
          35.82            35                0

SQL> select mod(5,2) from dual;
  MOD(5,2)
----------
         1

SQL> select mod(sal,2000) from emp where job like 'SALESMAN';
MOD(SAL,2000)
-------------
         1600
         1250
         1250
         1500
SQL> select sal from emp where job like 'SALESMAN';
       SAL
----------
      1600
      1250
      1250
      1500

SQL> select ceil(35.23), ceil(35.5), ceil(35.6) from dual;

CEIL(35.23) CEIL(35.5) CEIL(35.6)
----------- ---------- ----------
         36         36         36

SQL> select floor(35.23), floor(35.5), floor(35.6) from dual;

FLOOR(35.23) FLOOR(35.5) FLOOR(35.6)
------------ ----------- -----------
```

```
        35         35         35
```

Date Functions:

SYSDATE is a pseudo column that returns the current date and time. When we select sysdate it will display in a dummy table called DUAL. Oracle date range between 1st jan 4712 BC and 31st Dec 4712 AD.

Months_between	It returns the numeric value. Finds the no. of months between date1 and date2, result may be positive or negative.
Add_months	It returns the date datatype. Adds n number of calendar months to date, n must be an integer and it can be negative
Last_day	It returns the date datatype. Date of the
Next_day	It returns the date datatype. Date of the next specified day of the week following date1, char may be number representing a day, or a character

```
SQL> select sysdate from dual;


SYSDATE

---------

08-JUL-10


SQL> select months_between(sysdate, hiredate) from emp;


MONTHS_BETWEEN(SYSDATE,HIREDATE)

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

                354.728983

                352.632208

                352.567692

                351.212854
```

```
          345.374144

          350.245112

          348.987047

          278.664466

          343.728983

                 346

          277.535434

          343.180596

          343.180596

          341.535434


14 rows selected.


SQL> select months_between('01-jan-2010', sysdate) from dual;

MONTHS_BETWEEN('01-JAN-2010',SYSDATE)

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

                           -6.2451325


SQL> select last_day(sysdate) from dual;

LAST_DAY(

---------

31-JUL-10


SQL> select last_day(hiredate),last_day('15-feb-88') from emp;

LAST_DAY( LAST_DAY(

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

31-DEC-80 29-FEB-88

28-FEB-81 29-FEB-88

28-FEB-81 29-FEB-88
```

```
30-APR-81 29-FEB-88

30-SEP-81 29-FEB-88

31-MAY-81 29-FEB-88

30-JUN-81 29-FEB-88

30-APR-87 29-FEB-88

30-NOV-81 29-FEB-88

30-SEP-81 29-FEB-88

31-MAY-87 29-FEB-88

31-DEC-81 29-FEB-88

31-DEC-81 29-FEB-88

31-JAN-82 29-FEB-88
```

Sunday	1
Monday	2
Tuesday	3
Wednesday	4
Thursday	5
Friday	6
Saturday	7

```
SQL> select last_day(hiredate),last_day('15-feb-88') from emp;

LAST_DAY( LAST_DAY(

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

31-DEC-80 29-FEB-88

28-FEB-81 29-FEB-88

28-FEB-81 29-FEB-88

30-APR-81 29-FEB-88

30-SEP-81 29-FEB-88
```

```
31-MAY-81 29-FEB-88
30-JUN-81 29-FEB-88
30-APR-87 29-FEB-88
30-NOV-81 29-FEB-88
30-SEP-81 29-FEB-88
31-MAY-87 29-FEB-88
31-DEC-81 29-FEB-88
31-DEC-81 29-FEB-88
31-JAN-82 29-FEB-88
14 rows selected.
SQL> select next_day(sysdate, 'friday') from dual;
NEXT_DAY(
---------
09-JUL-10
```

```
SQL> select next_day(hiredate,'friday'), next_day(hiredate,6) from emp where deptno=10;
NEXT_DAY( NEXT_DAY(
--------- ---------
12-JUN-81 12-JUN-81
20-NOV-81 20-NOV-81
29-JAN-82 29-JAN-82
```

Month mid value	1-15
Day mid value	Sunday
Year mid value	30-jun

```
SQL> select round(sysdate,'day') from dual;
ROUND(SYS
---------
11-JUL-10

SQL> select round(sysdate,'year') from dual;
ROUND(SYS
---------
01-JAN-11

SQL> select round(sysdate,'month') from dual;
ROUND(SYS
---------
01-JUL-10

SQL> select trunc(sysdate,'month'), trunc(sysdate,'year') from dual;
TRUNC(SYS TRUNC(SYS
--------- ---------
01-JUL-10 01-JAN-10
```

Conversion Functions:

To_char(number | date,[‘fmt’]	Converts numbers or date to character format fmt
To_number(char)	Converts char, which contains a number to a NUBER
To_date	Converts the char value representing date, into a date value according to fmt specified. If fmt is omitted, format is DD-MM-YYYY

```
SQL> select to_char(3000, '$9999.99') from dual;
TO_CHAR(3
---------
 $3000.00

SQL> select to_char(sysdate, 'fmday, ddth month yyyy') from dual;
TO_CHAR(SYSDATE,'FMDAY,DDTHMON
------------------------------
thursday, 8th july 2010

SQL> select to_char(sysdate, 'hh:mi:ss') from dual;
TO_CHAR(
--------
03:04:27

SQL> select to_char(sal,'$9999.99') from emp;
TO_CHAR(S
---------
  $800.00
 $1600.00
 $1250.00
 $2975.00
 $1250.00
 $2850.00

 $2450.00

 $3000.00

 $5000.00

 $1500.00

 $1100.00

  $950.00

 $3000.00

 $1300.00
```

```
SQL> select empno,ename, job,sal from emp where sal>to_number('1500');

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7499 ALLEN      SALESMAN        1600
      7566 JONES      MANAGER         2975
      7698 BLAKE      MANAGER         2850
      7782 CLARK      MANAGER         2450
      7788 SCOTT      ANALYST         3000
      7839 KING       PRESIDENT       5000
      7902 FORD       ANALYST         3000
```

```
SQL> update emp set hiredate=to_date('1998 05 20', 'yyyy mm dd') where ename='SMITH';

1 row updated.
```

General Functions:

Uid	This function returns the integer value corresponding to the user currently logged in

User	This function returns the login user name, which is in varchar2 datatype
Nvl	This function is used in case where we want to consider null values
Vsize	This function returns the number of bytes in the expression, if expression is null it returns zero.
Case	Case expression let you use IF-THEN-ELSE logic in SQL statements without having invoke procedures
Decode	Decodes and expression in a way similar IF-THEN-ELSE logic. Decodes and expression after comparing it to each search value.

```
SQL> select uid from dual;

      UID
----------
       59


SQL> select user from dual;

USER
------------------------------
SCOTT


SQL>  select ename, nvl(comm,0) from emp;

ENAME      NVL(COMM,0)
---------- -----------
SMITH              0
ALLEN            300
WARD             500
JONES              0
MARTIN          1400
BLAKE              0
CLARK              0
```

```
SCOTT           0
KING            0
TURNER          0
ADAMS           0
JAMES           0
FORD            0
MILLER          0

SQL> select vsize('hello') from dual;
VSIZE('HELLO')
--------------
             5

SQL> select vsize(ename) from emp;
VSIZE(ENAME)
------------
           5
           5
           4
           5
           6
           5
           5
           5
           4
           6
           5
           5
           4
```

```
      6

SQL> select ename,job,sal ,
  case job when 'CLERK' then 1.10*sal
    when 'MANAGER' then 1.15*sal
    else sal end "revised salary" from emp;

ENAME      JOB              SAL revised salary
---------- --------- ---------- --------------
SMITH      CLERK            800            880
ALLEN      SALESMAN        1600           1600
WARD       SALESMAN        1250           1250
JONES      MANAGER         2975        3421.25
MARTIN     SALESMAN        1250           1250
BLAKE      MANAGER         2850         3277.5
CLARK      MANAGER         2450         2817.5
SCOTT      ANALYST         3000           3000
KING       PRESIDENT       5000           5000
TURNER     SALESMAN        1500           1500
ADAMS      CLERK           1100           1210
JAMES      CLERK            950           1045
FORD       ANALYST         3000           3000
MILLER     CLERK           1300           1430

SQL> select ename,job,sal ,
   decode(job,'CLERK',1.10*sal,'MANAGER',1.15*sal,'SALESMAN',1.20*sal,sal) "revised
salary" from
emp;

ENAME      JOB              SAL revised salary
```

```
---------- --------- ---------- --------------
SMITH     CLERK          800          880
ALLEN     SALESMAN       1600         1920
WARD      SALESMAN       1250         1500
JONES     MANAGER        2975       3421.25
MARTIN    SALESMAN       1250         1500
BLAKE     MANAGER        2850        3277.5
CLARK     MANAGER        2450        2817.5
SCOTT     ANALYST        3000         3000
KING      PRESIDENT      5000         5000
TURNER    SALESMAN       1500         1800
ADAMS     CLERK          1100         1210
JAMES     CLERK          950          1045
FORD      ANALYST        3000         3000
MILLER    CLERK          1300         1430
```

Multiple Row functions:

A group function returns a result based on a group of rows. Some of these are just purely mathematical functions.

This group function operate on sets of rows of rows to give one result per group. These sets may be the whole table or the table split into groups.

Sum	To obtain the sum of a range of values of a record set
Avg	This function will return the average of values of the column specified in the argument of column
Min	This function will give the least value of all values of the column present in the argument.
Max	This function will give the maximum value of all values of the column present in the argument.
Count	This function will return the number of rows contained to the related column

```
SQL> select sum(sal) from emp;
  SUM(SAL)
----------
     29025U

SQL> select avg(Sal) from emp;
  AVG(SAL)
----------
2073.21429

SQL> select min(sal) from emp;
  MIN(SAL)
----------
       800

SQL> select max(sal) from emp;
  MAX(SAL)
----------
      5000

SQL> select count(*) from emp;
  COUNT(*)
----------
        14
```