Lab 03: SQL Functions

Objective(s):

To learn the use of Single Row Functions in SELECT statement.

SQL Functions

There are two types of SQL functions.

1) Single Row Functions:

- These functions operate of single rows only and return one result per row.
- Can be used in SELECT,WHERE and ORDER BY clause.

Example:

select empno, lower(ename) from emp;

EMPNO	LOWER(ENAM
7369	smith
7499	allen
7521	ward
7566	jones

2) Multiple Row Functions:

These functions manipulate group of rows to give one and only one result per group of rows.

Example:

Select deptno,sum(sal) from emp group by deptno;

DEPTNO	SUM(SAL)
10	8750
20	10875
30	9400

Type of Single Row Functions

1) Character Functions:

Accept character input and can return both character and number values.

2) Number Functions:

Accept numeric input and returns numeric values.

3) Date Functions:

Operate on values of date data type.It always return a date value except the **MONTHS_BETWEEN** function, which returns a number.

4) Conversion Functions:

Converts a value from one data type to another.

5) General Functions:

- NVL function.

- DECODE function etc.

Character Function:

1) LOWER(column | expression):

Converts mixed case or upper case character string to lower case.

Example:

Select lower(job) from emp;

2) UPPER(Column | Expression):

Converts mixed case or lower case character string to upper case.

Example:

Select upper(job) from emp;

3) INITCAP(column | expression):

Converts first letter of each word to upper case and remaining letters to lower case.

Example:

Select initcap(job) from emp;

4) CONCAT(col 1 | exp 1, col 2 | exp 2):

Concatenate the first character value to the second character value.

5) SUBSTR(col | exp, m , [n]):

Returns specified character from the character value starting at character position ***m*** up to ***n*** characters.

6) LENGTH(col | exp):

Returns the number of characters in value.

7) INSTR(col | exp, m):

Return the numeric position of a named character.

Example:

Select concat(ename,job), length(ename), instr(ename,‟A‟) from emp

where substr(job,1,5) = „„SALES‟‟;

8) TRIM(leading|trailing|both trim_character from trim source):

Enables you to trim heading or trailing characters (or both) from a character string. If trim_source is character literal, you must enclose it in single quotes.

9) REPLACE(text,search_string,replacement_string):

Searches a text expression for a character string and, if found, replaces it with a specified replacement string.

NUMBER FUNCTIONS:

1) MOD(m,n):

Returns the remainder of ***m*** divided by ***n.***

Example:

Select ename,sal,mod(sal,comm) from emp where job = "SALESMAN";

2) ROUND(m,n):

Rounds the value ***m*** to specified ***n*** decimal places.

3) TRUNC(m,n):

Truncates the value ***m*** to specified ***n*** decimal places.

Example

Select round(45.269,2),trunc(45.269,2) from dual;

OUTPUT:

ROUND(45.269,2) TRUNC(45.269,2)
---------- ----------

45.27 45.26

Arithmetic with Dates:

 Add or subtract a number to or from a date for a resultant date value.

 SYSDATE is a function that returns:

o Date

o Time

Example:

1) Select sysdate+10 from dual;

2) Select sysdate-10 from dual;

 Dual is a dummy table you can use to view results from functions and calculations.

 Subtract two dates to find the number of days b/w those dates.

Example:

Select sysdate – hiredate from emp;

Date Functions:

1)Months_Between(date1,date2):

Number of months b/w two months.

2) Add_Months(date,n):

Add calendar months to date.

3) Next_day(date,'char'):

Date of the next specified day of the week(„char‟);

4) Last_day(date):

Finds the date of the last date of the month that contains date.

Example:

select empno,hiredate, months_between(sysdate,hiredate), add_months(hiredate,6), next_day(hiredate,‟FRIDAY‟), last_day(hire_date)

from emp;

Conversion Functions

1)TO_CHAR(number|date,'fmt'):

Converts a number or date value to a varchar2 character string with format „fmt‟.

Example:

- Select empno, to_char(hiredate,‟mm/yy‟) Month_hired from emp;
- Select to_char(hiredate,‟dd-month-yyyy‟) from emp;
- Select to_char(sal,‟$99,999‟) from emp;
- Select to_char(17145,‟$099,999‟) from emp;

2)TO_NUMBER(char,'fmt'):

Convert a character string to a number format.

Example:

Select sal+to_number(substr("$100",2,3)) from emp;

3)TO_DATE(char,'fmt'):

Convert a character string to a date format.

Example:

Select ename,hiredate from emp where hiredate = to_date("Feb 22, 1981", "Mon dd, yyyy");

General Functions

NVL Function :

Syntax:

NVL(expr1, expr2)

 If ***expr1*** is null, NVL returns ***expr2.***

 The argument ***expr1*** can have any data type.

 Datatypes of ***expr1*** and ***expr2*** must match.

Examples:

1) Select ename,sal,comm, (sal * 12) + NVL(Comm,0) from emp;

2) Select nvl(job,"No Job Yet") from emp;

3) Select ename,NVL(to_char(manager_id),"No Manager") from employees where mgr is NULL;

NVL2 Function :

Syntax:

NVL2(expr1, expr2, expr3)

 If ***expr1 is not null***,NVL2 returns ***expr2***. If ***expr1 is null***, NVL2 returns ***expr3.***

 The argument ***expr1*** can have any data type.

 Datatypes of ***expr1,expr2*** and ***expr3*** must match.

Examples:

1)Select ename,sal,comm, NVL2(Comm,sal+comm,sal) from emp;

NULLIF Function:

Syntax:

NULLIF(expr1,expr2)

NULLIF function compares two expressions. If they are equal, the function returns NULL else function returns the first expression.

COALESCE Function:

 COALESCE is just like NVL function but this function can take multiples alternative values.

 If the first expression is not null, it returns that expression; otherwise, it does a COALESCE of the remaining expressions.

Syntax:

COALESCE(expr1,expr2,…..,exprn)

In the syntax:

Expr1 returns this expression if it is not null.

Expr2 returns this expression if the first expression is null and this expression is not null.

Exprn returns this expression if the preceding expressions are null.

Conditional Expressions:

Two methods used to implement conditional processing (if-then-else logic) within the SQL statements are:

CASE expression.

Example:

```
Select ename,job,sal,
Case deptno
when 10 then ‘ACCOUNTS’
when 20 then ‘MARKETING’
else ‘ADMIN’
end “DNAME”
from emp;
```

Exercise

Now by using the EMP table, formulate the following queries in SQL

1. Display ename, job, sal of all the employees whose job is “CLERK”. Keeping this in mind that CLERK may be in capital, small or combination of small capital characters in the table.

2. Generate new names of the employees by combining the first 3 characters of the ename and last 3 characters of the job.

3. Generate new jobs of the employees by changing letter E with A in the existing jobs.

4. Display ename, total years in the department.

5. Display ename, hiredate of all the employees who were hired before February 20, 1981. Keeping this in mind that hiredates should be displayed in the format “MONTH DATE, YEAR”. Also date in the where clause should be in the format “MONTH DATE, YEAR”.

6. Display employee name,sal,comm and total salary (Comm + Sal) of all the employees. Keeping this in mind that comm of some of the employee might be null but total salary of the employees should not be null.

7. Display employee name,sal,comm and total salary (Comm + Sal) of all the employees. Keeping this in mind that comm or sal of some of the employee might be null but total salary of the employees should not be null.In case of comm is null, total salary should be the original salary or if the salary is null, total salary should be the original comm.

8. Display ename,sal,comm and bonus of all the employees. If job = CLERK then bonus = 20% of the salary, if job = SALESMAN then bonus = 30% of the salary, if job = MANAGER then bonus = 50% of the salary else bonus = 10% of the salary