Lab Session 02

(*Functions, Dates,Operators and Group of Data)*

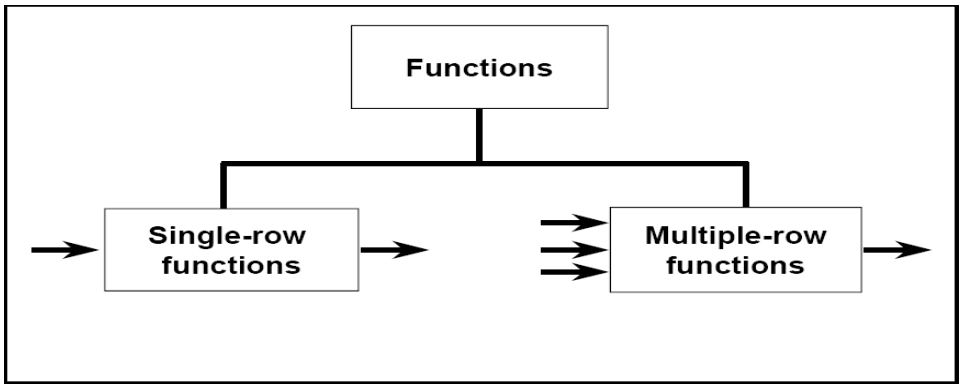
ORACLE BUILT-IN FUNCTIONS:

Functions are a very powerful feature of SQL and can be used to do the following:

- Perform calculations on data
- Modify individual data items
- Manipulate output for groups of rows
- Format dates and numbers for display
- Convert column data types

Types of Functions:

- Single-row functions
- Multiple-row functions/Group Functions/Aggregate Functions



Types of Single Row Functions:

1) Numeric Functions

These are functions that accept numeric input and return numeric values.

2) Character or Text Functions

These are functions that accept character input and can return both character and number values.

3) Date Functions

These are functions that take values that are of datatype DATE as input and return values of datatype DATE, except for the MONTHS_BETWEEN function, which returns a number.

4) Conversion Functions

These are functions that help us to convert a value in one form to another form. For Example: a null value into an actual value, or a value from one datatype to another datatype like NVL, TO_CHAR, TO_NUMBER, TO_DATE etc.

Numeric Functions

Function Name	Return Value
ABS (x)	Absolute value of the number 'x'
CEIL (x)	Integer value that is Greater than or equal to the number 'x'
FLOOR (x)	Integer value that is Less than or equal to the number 'x'
TRUNC (x, y)	Truncates value of number 'x' up to 'y' decimal places
ROUND (x, y)	Rounded off value of the number 'x' up to the number 'y' decimal places

Function Name	Examples	Return Value
ABS (x)	ABS (1) ABS (-1)	1 1
GREATEST(value1, value2, ...)	GREATEST(7,8,10)	10
LEAST(value1, value2, ...)	LEAST(7,8,10)	7
CEIL (x)	CEIL (2.83) CEIL (2.49) CEIL (-1.6)	3 3 -1
FLOOR (x)	FLOOR (2.83) FLOOR (2.49) FLOOR (-1.6)	2 2 -2
TRUNC (x, y)	ROUND (125.456, 1) ROUND (125.456, 0) ROUND (124.456, -1)	125.4 125 120
ROUND (x, y)	TRUNC (140.234, 2) TRUNC (-54, 1) TRUNC (5.7) TRUNC (142, -1)	140.23 54 5 140

Character Functions

Function Name	Return Value
LOWER (string_value)	All the letters in 'string_value' is converted to lowercase.
UPPER (string_value)	All the letters in 'string_value' is converted to uppercase.
INITCAP (string_value)	All the letters in 'string_value' is converted to mixed case.
LTRIM (string_value, trim_text)	All occurrences of 'trim_text' is removed from the left of 'string_value'.
RTRIM (string_value, trim_text)	All occurrences of 'trim_text' is removed from the right of'string_value' .
TRIM (trim_text FROM string_value)	All occurrences of 'trim_text' from the left and right of 'string_value' ,'trim_text' can also be only one character long .
SUBSTR (string_value, m, n)	Returns 'n' number of characters from'string_value' starting from the 'm'position.
LENGTH (string_value)	Number of characters in 'string_value'in returned.
LPAD (string_value, n, pad_value)	Returns 'string_value' left-padded with'pad_value' . The length of the whole string will be of 'n' characters.
RPAD (string_value, n, pad_value)	Returns 'string_value' right-padded with 'pad_value' . The length of the whole string will be of 'n' characters.

Function Name	Examples	Return Value
LOWER(string_value)	LOWER('Good Morning')	good morning
UPPER(string_value)	UPPER('Good Morning')	GOOD MORNING
INITCAP(string_value)	INITCAP('GOOD MORNING')	Good Morning
LTRIM(string_value, trim_text)	LTRIM ('Good Morning', 'Good)	Morning
RTRIM (string_value, trim_text)	RTRIM ('Good Morning', ' Morning')	Good
TRIM (trim_text FROM string_value)	TRIM ('o' FROM 'Good Morning')	Gd Mrning
SUBSTR (string_value, m, n)	SUBSTR ('Good Morning', 6, 7)	Morning
LENGTH (string_value)	LENGTH ('Good Morning')	12
LPAD (string_value, n, pad_value)	LPAD ('Good', 6, '*')	**Good
RPAD (string_value, n, pad_value)	RPAD ('Good', 6, '*')	Good**

Conversion Functions

Function Name	Return Value
TO_CHAR (x [,y])	Converts Numeric and Date values to a character string value. It cannot be used for calculations since it is a string value.
TO_DATE (x [, date_format])	Converts a valid Numeric and Character values to a Date value. Date is formatted to the format specified by 'date_format'.
NVL (x, y)	If 'x' is NULL, replace it with 'y'. 'x' and 'y'must be of the same datatype.

Function Name	Examples	Return Value
TO_CHAR ()	TO_CHAR (3000, '$9999') TO_CHAR (SYSDATE, 'Day, Month YYYY')	$3000 Monday, June 2008
TO_DATE ()	TO_DATE ('01-Jun-08')	01-Jun-08
NVL ()	NVL (null, 1)	1

Date Functions

Function	Result	Description
MONTHS_BETWEEN('01-SEP-95', '11-JAN-94')	19.6774194	Number of months between two dates
ADD_MONTHS('11-JAN-94', 6)	'11-JUL-94'	Add calendar months to dates
NEXT_DAY('01-SEP-95', 'FRIDAY')	'08-SEP-95'	Next day of the date specified
LAST_DAY('01-SEP-95')	'30-SEP-95'	Last day of the month
ROUND(TO_DATE('25-JUL-95', 'DD-MON-YY'), 'MONTH')	01-AUG-95	Round date
ROUND(TO_DATE('25-JUL-95', 'DD-MON-YY'), 'YEAR')	01-JAN-96	Round date
TRUNC(TO_DATE('25-JUL-95', 'DD-MON-YY'), 'MONTH')	01-JUL-95	Truncate date
TRUNC(TO_DATE('25-JUL-95', 'DD-MON-YY'), 'YEAR')	01-JAN-95	Truncate date

NOTE: Date functions operate on Oracle dates. All date functions return a value of DATE data type except MONTHS_BETWEEN, which returns a numeric value.

SYSDATE:

SYSDATE is a date function that returns the current date and time. The current date can be displayed by selecting SYSDATE from a table. It is customary to select SYSDATE from a dummy table called **DUAL.**

The **DUAL** table is owned by the user **SYS** and can be accessed by all users. It contains one column, **DUMMY**, and one row with the value **X**. It is useful for returning a value once only

– for instance, the value of a constant, pseudocolumn, or expression that is not derived from a table with user data.

For example, to display the current date using the DUAL table as

SELECT SYSDATE FROM DUAL;

Arithmetic with Dates

We can add or subtract a number to or from a date for a resultant date value.

For example, to display the name and the number of weeks employed for all employees in department 10.

SELECT ENAME, (SYSDATE – HIREDATE) / 7 “NUMBER OF WEEKS” FROM

EMP WHERE DEPTNO = 10;

2. SELECT TO_DATE (’31-JUL-2012’) + 2 FROM DUAL;

3. SELECT TO_DATE (’02-AUG-2012’) – TO_DATE (’31-JUL-2012’) FROM DUAL;

Rownum Operator:

Used to display certain number of rows

SELECT column_name(s) FROM table_name WHERE ROWNUM <=

number SELECT EMPNO, ENAME, DEPTNO FROM EMP WHERE

ROWNUM <= 3;

Conversion Of NULL values:

To convert Null Values two functions are used:

Take an example query:

SELECT ENAME, NVL (COMM, 0), NVL2 (COMM, SAL+COMM, SAL) AS INCOME FROM EMP;

- NVL converts a NULL value to an actual value in this e.g to zero
- NVL2: if commission is not null then return salary + commission else if commission is Null then return salary

Group By Statement

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

Syntax

SELECT column_name,
aggregate_function(column_name) FROM table_name
WHERE column_name operator
value GROUP BY column_name

Examples

- **To show the department-wise average**

salary SELECT deptno, AVG(sal)

AVERAGE_SALARY FROM emp GROUP BY

deptno;

- **To show the job-wise total salary for each department**

SELECT deptno, job, sum(sal) FROM

emp GROUP BY deptno, job;

Having Clause

In the same way that we use the WHERE clause to restrict the rows that we select, the HAVING clause is used to restrict groups. First the group function is applied and the groups matching the HAVING clause are displayed.

Syntax

SELECT column,

group_function FROM table

[WHERE condition]

[GROUP BY

group_by_expression] [HAVING

group_condition] [ORDER BY

column];

Examples:

- **To show the department-wise average and maximum salary, in the descending order of average salary, for all departments having average salary higher than 2000.**

SELECT DEPTNO, AVG (SAL), MAX (SAL) FROM

EMP GROUP BY DEPTNO HAVING AVG (SAL) >

2000 ORDER BY AVG (SAL)

- **To display the job title and total monthly salary for each job title with a total payroll exceeding 5000.**

SELECT JOB, SUM (SAL) PAYROLL FROM EMP GROUP BY JOB HAVING SUM (SAL) > 5000 ORDER BY SUM (SAL);

Exercise

1. Print an employee name (first letter capital) and job title (lower Case).
2. Display the employee number, name (IN UPPER CASE) and department number for employee Blake..
3. To display the employee number, the month number and year of hiring.
4. List ANNUAL salary of all employees along with their names and job type.
5. List the employees in the ascending order of their salaries.
6. Display the salary of employee SCOTT with $ sign preceded.
7. List the employee's information whose daily salary is more than Rs.100.
8. Display the employee number, name, salary, salary increase by 15% expressed as a whole number (labeled as New Salary), the difference between old salary and new salary (labeled as Increment).
9. Write a query that displays the employee name and commission amounts. If an employee doesn't earn commission, put "No commission".
10. Display each employee name, hire date, and the day of the week on which the employee started, Label the column day.
11. Display the employee's name (labeled name) with the first letter capitalized and all other letters lowercase and the length of their name (labeled length), for all employees whose name starts with J, A or M
12. Display the name of the employee who are working in the company for the past 35 years.
13. Display the total salary being paid to all employees.
14. Display department number and total number of employees within each department.
15. Display department no. and max. Salary of those departments whose max. salary is greater than $2900
16. Display the department numbers with more than 3 employees in each department.