

### Ex.No. 3

### Inbuilt functions in SQL

#### Aim:

To perform manipulate records of table using inbuilt functions in SQL.

SQL functions are of two types

#### (i) Single row functions or scalar functions

- Returns only one value for every row queried in the table
- Can be used in Select clause and where clause
- It can be broadly classified into 5 categories
  - Date Functions
  - Character Functions
  - Conversion functions
  - Numeric functions
  - Miscellaneous functions

#### (ii) Group functions or multiple-row functions

Discussed in the next exercise (ie.; Ex. No.6)

**Note :** The exercises that follow mostly uses system table 'dual'. It is a table which is automatically created by Oracle along with the data dictionary. Dual table has one column defined to be of varchar2 type and contains only one row with value 'x'.

### SCALAR FUNCTIONS

**Q1)** List the hiredate of employees who work in deptno 20 in a format like 'WEDNESDAY JANUARY 12, 1983'

(Hint: DAY : Day of the week, MONTH : Name of the month, DD: Day of the month, and YYYY : Year)

SQL>

**Q2)** Display the hiredate with time of employess who work in deptno 20.

SQL>

**Q3)** Each employee receives a salary review after every 150 days of service. Now list employee name, hiredate and first salary review date of each employee who work in dept no 20.

SQL>

#### Q4. Date Functions

Functions	Value Returned	Input	Output
add_months(d,n)	'n' months added to date 'd'.	Select add_months(sysdate,2) from dual;	
last_day(d)	Date corresponding to the last day of the month	Select last_day(sysdate) from dual;	
to_date(str,'format')	Converts the string in a given format into Oracle date.	Select to_date('10-02-09','dd-mm-yy') from dual;	
to_char(date,'format')	Reformats date according to format	Select to_char(sysdate,'dy dd mon yyyy') from dual;	
months_between(d1,d2)	No. of months between two dates	Select months_between(sysdate, to_date('10-10-07','dd-mm-yy')) from dual;	
next_day(d,day)	Date of the 'day' that immediately follows the date 'd'	Select next_day(sysdate,'wednesday') from dual;	
round(d,'format')	Date will be rounded to the nearest day.	Select round(sysdate,'year') from dual;	
		Select round(sysdate,'month') from dual;	
		Select round(sysdate,'day') from dual;	
		Select round(sysdate) from dual;	
trunc(d,'format');	Date will be truncated to the nearest day.	Select trunc(sysdate,'year') from dual;	
		Select trunc(sysdate,'month') from dual;	
		Select trunc(sysdate,'day') from dual;	
		Select trunc(sysdate) from dual;	
greatest(d1,d2,...)	Picks latest of list of dates	Select greatest(sysdate, to_date('02-10-06','dd-mm-yy'), to_date('12-07-12','dd-mm-yy')) from dual;	
Date Arithmetic	Add /Subtract no. of days to a date	Select sysdate+25 from dual;	
		Select sysdate-25 from dual;	
	Subtract one date from another, producing a no. of days	Select sysdate - to_date('02-10-06','dd-mm-yy') from dual;	

**Q5. Character Functions**

Functions	Value Returned	Input	Output
initcap(char)	First letter of each word capitalized	Select initcap('jesus christ') from dual;	
lower(char)	Lower case	Select lower('DIED') from dual;	
upper(char)	Upper case	Select upper('for Us') from dual;	
ltrim(char, set)	Initial characters removed up to the character not in set.	Select ltrim('lordourgod','lord') from dual;	
rtrim(char, set)	Final characters removed after the last character not in set.	Select rtrim('godlovesyou','you') from dual;	
translate(char, from, to)	Translate 'from' by 'to' in char.	Select translate('jack','j','b') from dual;	
replace(char, search, repl)	Replace 'search' string by 'repl' string in 'char'.	Select replace('jack and jue','j','bl') from dual;	
substr(char, m, n)	Substring of 'char' at 'm' of size 'n' char long.	Select substr('wages of sin is death',10,3) from dual;	

**Q6. Conversion Functions**

Functions	Value Returned	Input	Output
to_date(str,'format')	Converts the string in a given format into Oracle date.	Select to_date('10-02-09','dd-mm-yy') from dual;	
to_char(date,'format')	Reformats date according to format	Select to_char(sysdate,'dy dd mon yyyy') from dual;	
to_char(number,'format')	Display number value as a char.	Select to_char(12345.5,'L099,999.99') from dual;	
to_number(char)	Char string to number form	Select to_number('123') from dual;	

**Q7. Numeric Functions**

Functions	Value Returned	Input	Output
Abs(n)	Absolute value of n	Select abs(-15) from dual;	
Ceil(n)	Smallest int >= n	Select ceil(33.645) from dual;	
Cos(n)	Cosine of n	Select cos(180) from dual;	
Cosh(n)	Hyperbolic cosine of n	Select cosh(0) from dual;	
Exp(n)	e <sup>n</sup>	Select exp(2) from dual;	
Floor(n)	Largest int <= n	Select floor(100.2) from dual;	
Ln(n)	Natural log of n (base e)	Select ln(5) from dual;	
Log(b,n)	Log n base b	Select log(2,64) from dual;	
Mod(m,n)	Remainder of m divided by n	Select mod(17,3) from dual;	
Power(m,n)	m power n	Select power(5,3) from dual;	
Round(m,n)	m rounded to n decimal places	Select round(125.67854,2) from dual;	
Sign(n)	If n<0, -1 if n=0, 0 otherwise 1.	Select sin(-19) from dual;	
Sin(n)	Sin of n	Select sin(90) from dual;	
Sinh(n)	Hyperbolic sin of n	Select sinh(45) from dual;	
Sqrt(n)	Square root of n	Select sqrt(7) from dual;	
Tan(n)	Tangent of n	Select tan(45) from dual;	
Tanh(n)	Hyperbolic tangent of n	Select tanh(60) from dual;	
Trunc(m,n)	m truncated to n decimal places	Select trunc(125.5764,2) from dual;	

**Q8. Miscellaneous Functions**

Functions	Value Returned	Input	Output
Uid	User id	Select uid from dual;	
User	User name	Select user from dual;	
Vsize(n)	Storage size of v	Select vsize('hello') from dual;	
NVL(exp1,exp2)	Returns exp1 if not null, otherwise returns exp2.	Select nvl(comm,50) from emp where empno=7369;	



## **GROUP FUNCTIONS**

### **Common Group Functions**

- AVG : Average value of a set
- COUNT : Numbers of non null values
- MAX : Maximum of a set
- MIN : Minimum of a set
- STDDEV : Standard Deviation of a set
- SUM : Sum of a set
- VARIANCE : Variance of a set

### **Syntax :**

```
SELECT    column, group_function(column)
FROM      table
[WHERE    condition]
[GROUP BY group_column_or_expression]
[HAVING   group_condition]
[ORDER BY column];
```

- Group functions ignore null values
- *Group by* Clause is used to modularize rows in a table into smaller groups
- Columns that are not a part of the Group Functions should be included in the Group by clause
- Any column or expression in the SELECT list that is not an aggregate function must be in the GROUP BY clause
- Group Functions cannot be placed in the where clause
- HAVING clause is to restrict groups Groups satisfying the HAVING condition are displayed
- Order of evaluation of the clauses :
  - WHERE clause
  - GROUP BY clause
  - HAVING clause

**Q9)** Find number of rows in the table EMP  
**SQL >**

- Q10)** Find number of designations available in EMP table.  
**SQL>**
- Q11)** Find number of employees who earn commission in EMP table.  
**SQL>**
- Q12)** What is the difference between the following queries  
**SQL > select count(comm) from emp;**  
**SQL > select count(nvl(comm,0)) from emp;**
- Q13)** Find the total salary paid to the employees.  
**SQL>**
- Q14)** Find maximum, minimum and average salary in EMP table.  
**SQL>**
- Q15)** Find number of employees who work in department number 30  
**SQL> select count(\*) from emp**  
**where deptno=30;**
- Q16)** Find the maximum salary paid to a 'CLERK'  
**SQL>**
- Q17)** List the department numbers and number of employees in each department  
**SQL>**
- Q18)** List the jobs and number of employees in each job. The result should be in the descending order of the number of employees.  
**SQL> select job, count(\*) from emp**  
**group by job**  
**order by count(\*) desc;**

**Q19)** List the total salary, maximum and minimum salary and average salary of the employees jobwise.  
**SQL>**

**Q20)** List the total salary, maximum and minimum salary and average salary of the employees jobwise, for department 20 and display only those rows having an average salary > 1000.

**SQL> select job,sum(sal), max(sal), min(sal), avg(sal)**  
**from emp**  
**group by job, deptno**  
**having deptno=20 and avg(sal) > 1000;**

**Q21)** List the job and total salary of employees jobwise, for jobs other than 'PRESIDENT' and display only those rows having total salary > 5000.

**SQL>**

**Q22)** List the job, number of employees and average salary of employees jobwise. Display only the rows where the number of employees in each job is more than two.

**SQL>**