

DATABASE SYSTEM

LAB No: 03

Instructor: Marina Gul

Objective of Lab No. 3:

After performing lab 3, students will be able to:

- SQL Functions
- Type of Scalar Functions
- Character Functions
- Number Functions
- Date Functions
- Conversion Functions
- General Functions

SQL Functions

There are two types of SQL functions, aggregate functions, and scalar (non-aggregate) functions.

1. **Scalar(Non Aggregate)/Single Row Functions:**

Non-aggregate functions operate on each record independently. OR These functions operate on single rows only and return one result per row and can be used in SELECT, WHERE and ORDER BY clause.

2. **Aggregate/Multiple Rows Functions:**

Aggregate functions operate on many records and produce a summary, works with GROUP BY. OR These functions manipulate group of rows to give one and only one result per group of rows. There are so many built-in functions in SQL to do various calculations on data.

Types of Scalar Functions

1. **Character Functions:** Accept character input and can return number or character value.
2. **Numeric 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:** Usually contains NULL handling functions.

Character Functions

LOWER()/LCASE() function

LOWER() function is used to convert all characters of a string to lower case.

Syntax:

```
SELECT LOWER/LCASE(string)
```

UPPER()/UCASE() function

UPPER() function is used to convert all characters of a string to uppercase.

Syntax:

```
UPPER/UCASE(string)
```

TRIM() function

The TRIM() function removes leading and trailing spaces from a string.

Syntax:

```
TRIM(string)
```

CONCAT() function

The CONCAT() function adds two or more expressions together.

Syntax:

```
CONCAT(expression1, expression2, expression3,...)
```

LENGTH Function

The LENGTH() function returns the length of a string.

Syntax:

```
LENGTH(string)
```

REPLACE() Function

The REPLACE() function replaces all occurrences of a substring within a string, with a new substring.

Note: This function performs a case-sensitive replacement.

Syntax:

```
REPLACE(string, from_string, new_string)
```

SUBSTRING()/SUBSTR()/MID() Functions

These functions extract a substring from a string (starting at any position).

Syntax:

```
SELECT SUBSTRING/SUBSTR/MID(string, start, length)
```

Parameter	Description
string	Required. The string to extract from
start	Required. The start position. Can be both a positive or negative number. If it is a positive number, this function extracts from the beginning of the string. If it is a negative number, this function extracts from the end of the string
length	Required. The number of characters to extract

LEFT() Function

The LEFT() function extracts a number of characters from a string (starting from left).

Syntax:

```
LEFT(string, number_of_chars)
```

RIGHT() Function

The RIGHT() function extracts a number of characters from a string (starting from right).

Syntax:

```
RIGHT(string, number_of_chars)
```

STRCMP() Function

The STRCMP() function compares two strings.

```
STRCMP(string1, string2)
```

Return Values:

- If string1 = string2, this function returns 0
- If string1 < string2, this function returns -1
- If string1 > string2, this function returns 1

INSTR() Function

The INSTR() function returns the position of the first occurrence of a string in another string. This function performs a case-insensitive search.

Syntax:

INSTR(string1, string2)	
Parameter	Description
string1	Required. The string that will be searched
string2	Required. The string to search for in string1. If string2 is not found, this function returns 0

Number Functions

MOD() Function

The MOD() function returns the remainder of a number divided by another number.

Syntax:

```
MOD(x, y)
```

ROUND() Function

The ROUND() function rounds a number to a specified number of decimal places.

Syntax:

```
ROUND(number, decimals)
```

TRUNCATE() Function

The TRUNCATE() function truncates a number to the specified number of decimal places.

Syntax:

```
TRUNCATE(number, decimals)
```

CEILING/CEIL() Function

The CEILING/CEIL() function returns the smallest integer value that is bigger than or equal to a number.

Syntax:

```
CEILING/CEIL(number)
```

Date Functions

DATEIF() Function

The DATEDIFF() function returns the number of days between two date values.

Syntax:

```
DATEDIFF(date1, date2)
```

ADDDATE() Function

The ADDDATE() function adds a time/date interval to a date and then returns the date.

```
ADDDATE(date, INTERVAL value addunit)
```

OR

```
ADDDATE(date, days)
```

Example:

```
ADDDATE("2017-06-15 09:34:21", INTERVAL 15 MINUTE)
```

[For more details](#)

CURRENT_DATE() Function

The CURRENT_DATE() function returns the current date.

Syntax:

```
CURRENT_DATE()
```

CURRENT_TIME() Function

The CURRENT_TIME() function returns the current time.

Syntax:

```
CURRENT_TIME()
```

CURRENT_TIMESTAMP() Function

The CURRENT_TIMESTAMP() function returns the current date and time.

Syntax:

```
CURRENT_TIMESTAMP()
```

DATE() Function

The DATE() function extracts the date part from a datetime expression.

Syntax:

```
DATE(expression)
```

Other Functions

The **MINUTE**(*datetime*) function returns the minute part of a time/datetime (from 0 to 59).

The **HOURL**(*datetime*) function returns the hour part for a given date (from 0 to 838).

The **DAY**(*datetime*) function returns the day of the month for a given date (a number from 1 to 31).

The **MONTH**(*date*) function returns the month part for a given date (a number from 1 to 12).

The **YEAR**(*date*) function returns the year part for a given date (a number from 1000 to 9999).

Conversion Functions

DATE_FORMAT() Function

The DATE_FORMAT() function formats a date as specified.

Syntax:

```
DATE_FORMAT(date, format)
```

Example:

```
DATE_FORMAT("2017-06-15", "%M %d %Y")
```

[For more details](#)

FORMAT() Function

The FORMAT() function formats a number to a format like "#,###,###.##", rounded to a specified number of decimal places, then it returns the result as a string.

Syntax:

```
FORMAT(number, decimal_places)
```

CONVERT() Function

The CONVERT() function converts a value into the specified datatype or character set.

Syntax:

```
CONVERT(value, type)
```

[For more details](#)

General Functions

NULLIF() Function

The NULLIF() function compares two expressions and returns NULL if they are equal. Otherwise, the first expression is returned.

```
NULLIF(expr1, expr2)
```

COALESCE() Function

The COALESCE() function returns the first non-null value in a list.

```
COALESCE(val1, val2, ..., val_n)
```

IFNULL() Function

The IFNULL() function returns a specified value if the expression is NULL.

If the expression is NOT NULL, this function returns the expression.

Syntax:

```
IFNULL(expression, alt_value)
```

Lab Tasks

1. Write a query to generate new names of the employees by combining the first 3 characters of the First_Name and last 3 characters of the job.
2. Generate new jobs of the employees by changing letter E with A in the existing jobs.
3. Write a query to Display Names, hire date years from the employees.
4. Write a query to display names, hire date of all the employees who were hired before July 30, 1987. Keeping this in mind that hire dates should be displayed in the format "MONTH DATE, YEAR". Also date in the where clause should be in the format "MONTH DATE, YEAR".
5. Write a query to display the last day of the current month three months before the current month.
6. Write a query to get a day of the current year.
7. Write a query to get the current date in the following format.
Sample date : 2014-09-04
Output : September 4, 2014
8. Write a query to get the current date in Thursday September 2014 format.
9. Write a query to get the first name and hire date from employees table where hire date between '1987-06-01' and '1987-07-30'.
10. Write a query to display the current date in the 05/09/2014 format.
11. Write a query to get the firstname, lastname who joined in the month of June.
12. Write a query to append '@iba-suk.edu.pk' to email field.
13. Write a query to get the employee id, first name and hire month using MID().
14. Write a query to extract the last 4 character of phone numbers.
15. Write a query to get the last word of the street address(from locations able)
16. Write a query to get the locations that have minimum street length. (locations tbl).
17. Write a query to display the first word from those job titles which contains more than one words.
18. Write a query to display the length of first name for employees where last name contain character 'c' after 2nd position.