Lab Manual for Database Management System

Lab No. 3

SQL Functions

LAB 3: SQL FUNCTIONS

1. INTRODUCTION:

SQL functions are essential elements in database queries, designed to perform specific tasks on data stored in a database. These functions enable you to manipulate, analyze, and retrieve information with ease. Whether you're working with numerical calculations, string manipulations, or date/time operations, SQL functions provide a versatile toolkit for tailoring your queries to meet specific requirements.

2. OBJECTIVE:

After completing this lab the student should be able to:

a. *Apply various SQL functions for efficient data manipulations.*
b. *Employ functions to streamline precise data retrieval in SQL queries.*
c. *Leverage SQL functions for improved data analysis and reporting.*

3. THEORY

SQL functions are classified into different categories based on their functionality. We will explore them one by one.

i. Aggregate Functions *(Since we have already addressed and examined this topic in the previous lab, let's move on to the next category)*

ii. Null Functions

In SQL, a NULL operation refers to an operation that involves handling NULL values. NULL is a special marker in SQL that represents a missing, undefined, or unknown value. It is not the same as an empty string or zero; it is distinct and indicates the absence of a value.

Functions	Description	Syntax
COALESCE()	*Returns the first non-NULL expression in the list*	***COALESCE(expr1, expr2, ...)***
ISNULL()	*Returns replacement_value if the expression is NULL*	***ISNULL(expression, replacement_value)***
IS NOT NULL	*Tests whether a value is not NULL*	***column_name IS NOT NULL***

Examples:

1. ***SELECT ProductName, COALESCE(StockQuantity, 0) [StockQuantity] FROM Products;***
2. ***SELECT ProductName, ISNULL(StockQuantity, 0) AS IsOutOfStock FROM Products;***
3. ***SELECT ProductName, StockQuantity FROM Products WHERE StockQuantity IS NOT NULL;***

iii. Date Functions

In SQL, date and time functions play a crucial role in handling and manipulating temporal data. They allow you to perform various operations on date and time values, such as extracting components, formatting, arithmetic operations, and more.

Functions	Description
CURRENT_TIMESTAMP	*Returns the current date and time.*
GETDATE()	*Returns the current date and time.*
GETUTCDATE()	*Returns the current UTC date and time.*
SYSDATETIME()	*Returns the current date and time in the system's time zone.*
DATEADD()	*Adds a specified interval to a date or time.*
DATEDIFF()	*Calculates the difference between two dates or times.*
FORMAT()	*Formats a date or time value.*
DATEPART()	*Returns a specific part of a date or time.*
DATENAME()	*Returns a specified part of a date as a character string.*
DAY() | MONTH() | YEAR()	*Returns the day of the month | month | year from a date.*

<u>Examples:</u>

1. ***SELECT CURRENT_TIMESTAMP AS CurrentTimestamp;***
2. ***SELECT GETDATE() AS CurrentDateTime;***
3. ***SELECT GETUTCDATE() AS CurrentUTCDateTime;***
4. ***SELECT SYSDATETIME() AS SystemDateTime;***
5. ***SELECT DATEADD(DAY, 7, OrderDate) AS FutureDate FROM Orders;***
6. ***SELECT DATEDIFF(DAY, OrderDate, ShippedDate) AS DaysToShip FROM Orders;***
7. ***SELECT FORMAT(OrderDate, 'MM/dd/yyyy') AS FormattedDate FROM Orders;***
8. ***SELECT DATEPART(YEAR, OrderDate) AS OrderYear FROM Orders;***
9. ***SELECT DATENAME(MONTH, OrderDate) AS MonthName FROM Orders;***
10. ***SELECT DAY(OrderDate) AS OrderDay, MONTH(OrderDate) AS OrderMonth, YEAR(OrderDate) AS OrderYear FROM Orders;***

iv. Mathematical Functions

Mathematical functions in SQL are essential for performing numerical operations and calculations within a database. These functions facilitate tasks such as rounding numbers, finding absolute values, and executing various mathematical operations.

Functions	Description	Syntax
ABS	*Returns the absolute value of a numeric expression.*	***ABS(number)***
ROUND	*Rounds a numeric value to the nearest integer or decimals.*	***ROUND(number, [decimal_places])***
CEIL	*Rounds a numeric value up to the nearest integer.*	***CEIL(number) or CEILING(number)***
FLOOR	*Rounds a numeric value down to the nearest integer.*	***FLOOR(number)***
POWER	*Raises a numeric value to the power of another.*	***POWER(base, exponent)***
SQRT	*Returns the square root of a numeric value.*	***SQRT(number)***
EXP	*Returns the exponential value of a numeric expression.*	***EXP(number)***
LOG	*Returns the natural logarithm of a numeric expression.*	***LOG(number)***

Examples:

1. ***SELECT ABS(Freight) AS AbsoluteFreight FROM Orders;***
2. ***SELECT ROUND(Freight, 2) AS RoundedFreight FROM Orders;***
3. ***SELECT CEILING(Freight) AS CeilFreight FROM Orders;***
4. ***SELECT FLOOR(Freight) AS FloorFreight FROM Orders;***
5. ***SELECT POWER(Freight, 2) AS PoweredFreight FROM Orders;***
6. ***SELECT SQRT(Freight) AS SquareRootFreight FROM Orders;***
7. ***SELECT EXP(Freight) AS ExponentialFreight FROM Orders;***
8. ***SELECT LOG(Freight) AS LogarithmicFreight FROM Orders;***

v. String Functions

String functions in SQL are operations specifically designed to manipulate character or text data. These functions enable the transformation, extraction, or combination of strings, offering a versatile toolkit for working with textual information within a database.

Functions	Description	Syntax
CONCAT	*Concatenates two or more strings.*	***CONCAT(string1, string2, ...)***
UPPER	*Converts characters to uppercase.*	***UPPER(string)***
LOWER	*Converts characters to lowercase.*	***LOWER(string)***
LEFT	*Returns a specified number of characters from the left of a string.*	***LEFT(string, length)***
RIGHT	*Returns a specified number of characters from the right of a string.*	***RIGHT(string, length)***
LEN (or LENGTH)	*Returns the length of a string.*	***LEN(string) or LENGTH(string)***
SUBSTRING	*Extracts a substring from a string.*	***SUBSTRING(string, start_position, length)***
REPLACE	*Replaces occurrences of a specified substring.*	***REPLACE(string, old_substring, new_substring)***
CHARINDEX	*Returns the starting position of a substring.*	***CHARINDEX(substring, string)***
TRIM	*Removes specified characters from the beginning or end of a string.*	

Examples:

1. SELECT CONCAT(CompanyName, ' - ', ContactName) AS FullContact FROM Customers;
2. SELECT UPPER(ProductName) AS UppercaseProductName FROM Products;
3. SELECT LOWER(CompanyName) AS LowercaseCompany FROM Customers;
4. SELECT LEFT(ContactName, 3) AS LeftThreeChars FROM Customers;
5. SELECT RIGHT(ProductName, 5) AS RightFiveChars FROM Products;
6. SELECT LEN(ContactName) AS ContactNameLength FROM Customers;
7. SELECT SUBSTRING(ProductName, 1, 3) AS SubstringProductName FROM Products;
8. SELECT REPLACE(CompanyName, ' ', '-') AS CompanyNameWithoutSpaces FROM Customers;
9. SELECT CHARINDEX('land', 'Finland') AS PositionInString;
10. SELECT TRIM('!' FROM '! Hello !') as [Trimmed String];

4. ACTIVITY TIME BOXING

Activity Name	Activity Time	Total Time
Instruments Allocation + Setting up Lab	10 mints	10 mints
Walk through Theory & Tasks (Lecture)	60 mints	60 mints
Implementation & Practice time	90 mints	80 mints
Evaluation Time	20 mints	20 mints
	Total Duration	180 mints

5. EXERCISE:

PERFORMANCE TASKS:

1. ***All Example Tasks.***
2. *Show the current timestamp and the order date of the latest order from the "sales" table.*
3. *Extract the day of the week for each order date.*
4. *Display the first 5 characters of the company name for each customer.*
5. *Display the product of the quantity and price raised to the power of 2 for each order detail.*

LAB FILE TASKS:

1. *Retrieve the title and publication date of books. If the publication date is NULL, display "Not available" instead.*
2. *Combine the author's first name and last name into a single column and display it in uppercase.*
3. *Find the difference in days between the current date and the hire date of employees in the "employee" table.*
4. *Determine the age of each author based on their birthdate.*
5. *Replace the substring "St" with "Street" in the authors' addresses.*
6. *Show the order date and the order month for all orders in the "orders" table.*
7. *List the products along with their units in stock. If units in stock are NULL, display "Out of stock.*
8. *Display the absolute value of the difference between the units in stock and the units on order for each product.*
9. *Combine the first three letters of the customer's city and the first two letters of the country for each customer.*
10. *Combine the first three letters of the customer's city and the first two letters of the country for each customer.*
11. *Use getdate() function to get system date then use datepart() function to extract month, year and date from it.*
12. *Use getdate() function to get system date then use day(), month() and year() functions to extract month, year and date from it.*