(



When to Use Single Quotes, Double Quotes, and Backticks in SQL?

Last Updated: 14 May, 2024



It is very important to know the use of **single quotes**, **double quotes**, and **backticks** in SQL, as it helps in writing error-free SQL queries and creating an effective database. All these quotes and Backticks enhance the effectiveness of the database and help in giving precise information to the users.

The usage of Quotation and Backticks in SQL holds significant importance. Incorrect or wrong use of these will lead to syntactic **errors**, **security risks**, and **issues** with **data integrity** in your database. So, SQL developers must learn how to use quotations and backticks correctly to write error-free, understandable, and effective codes.

Prerequisites

You should know the **fundamentals of SQL** syntax and database concepts before diving into the use of quotes in SQL. It will also help if you are familiar with the database management system you are using, such as <u>MySQL</u>, <u>PostgreSQL</u>, or <u>SQL Server</u>.

Single Quotes, Double Quotes, and Backticks

String literals are often delimited in SQL by **single quotes** ('). When identifiers like table names or column names contain reserved keywords or special characters, they should be enclosed in **double quotations** ("). Particular to MySQL, **backticks** (') are utilized as identifier separators in a manner akin to that of double quotes.

Examples of Using Quotes in SQL

Single Quotes(' ')

Single quotes are primarily used to denote string literals and character values in <u>SQL</u> queries.

Example 1: Using Single Quotes for String Literals

Syntax:

```
SELECT * FROM table WHERE column = 'value';
```

```
CREATE TABLE users (
    id INT,
    username VARCHAR(50)
);
INSERT INTO users (id, username) VALUES (1, 'John');
SELECT * FROM users WHERE username = 'John';
```

Output:

```
Databases SQL MySQL PostgreSQL PL/SQL MongoDB SQL Cheat Sheet SQL Interview Questions Mys

1 John
```

In this, a table named "users" has been created with columns named 'id' and 'username' and values have been inserted in respective columns. Here, 'John' is a string that's why single quotation is used to call it.

Example 2: Another Example of Single Quotes(' ')

```
CREATE TABLE products (
   id INT,
   name VARCHAR(50),
   category VARCHAR(50)
);
INSERT INTO products (id, name, category) VALUES
   (1, 'Laptop', 'Electronics'),
   (2, 'Smartphone', 'Electronics'),
   (3, 'Smart TV', 'Electronics');
```

```
SELECT * FROM products WHERE category = 'Electronics';
```

Output:

id	name	category
1	Laptop	Electronics
2	Smartphone	Electronics
3	Smart TV	Electronics

Here as well, the first table named "products" has been created with columns 'id', 'name', and 'category'. Then Values have been inserted in respective columns. After that value has been displayed by calling them and the single quotation is used in 'Electronics' as it is a string. You can use 'laptop' or any other string values from table as well and can see the respective outputs.

Double Quotes(" ")

Double quotes are used in certain database systems like <u>PostgreSQL</u> to specify identifiers such as table names or column names. They are particularly useful when dealing with identifiers that contain special characters or are **case-sensitive**.

Example: Using Double Quotes for Identifiers with Special Characters Syntax:

```
SELECT "FirstName", "LastName" FROM "Employee";
```

```
CREATE TABLE "Employee" (
"ID" INT,
```

```
"FirstName" VARCHAR(50),
   "LastName" VARCHAR(50)
);
INSERT INTO "Employee" ("ID", "FirstName", "LastName") VALUES (1,
'Alice', 'Smith');
SELECT "FirstName", "LastName" FROM "Employee";
```

Output:

FirstName	LastName
Alice	Smith

By using this query, all values present in given columns get displayed, along with the column in which they are present.

Backticks (``)

Backticks are commonly used in <u>MySQL</u> to enclose **identifiers**, especially when dealing with database objects with **unconventional** names.

Example: Using Backticks in MySQL for Identifiers

Syntax:

```
SELECT `column_name`FROM`table_name`;
```

```
CREATE TABLE `Product Sales` (
    `ID` INT,
    `Product Name` VARCHAR(50),
    `Units Sold` INT
);
INSERT INTO `Product Sales` (`ID`, `Product Name`, `Units Sold`)
VALUES (1, 'Laptop', 100);
SELECT `Product Name`, `Units Sold` FROM `Product Sales`;
```

Output:

Product Name	Units Sold
Laptop	100

By this query, Values present in respective columns get displayed along with their columns. You can check by inserting more values in your table.

Conclusion

For developers to effectively conquer database administration challenges, enhance data integrity, query efficiency, cross-platform compatibility, and syntactical accuracy, correct and precise use of <u>SQL</u> quotation must be known. As we know SQL is still a vital part of data-driven systems, software developers would benefit from understanding these nuances and becoming more competent and excellent in their work.



Similar Reads

When to Use Single Quotes, Double Quotes, and Backticks in SQL?

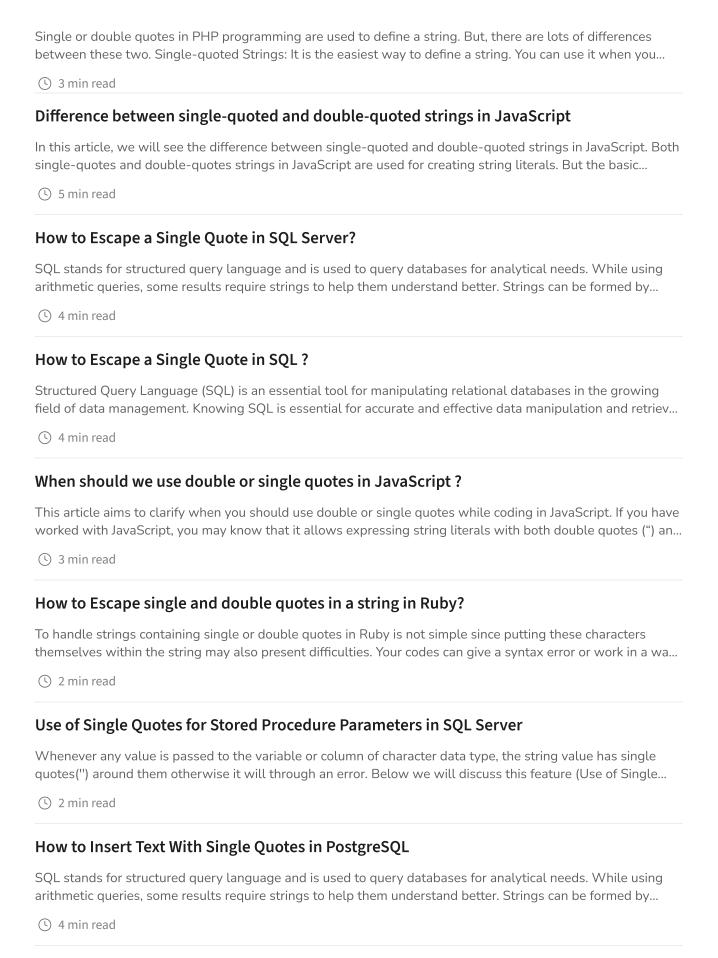
It is very important to know the use of single quotes, double quotes, and backticks in SQL, as it helps in writing error-free SQL queries and creating an effective database. All these quotes and Backticks enhance th...

(4 min read

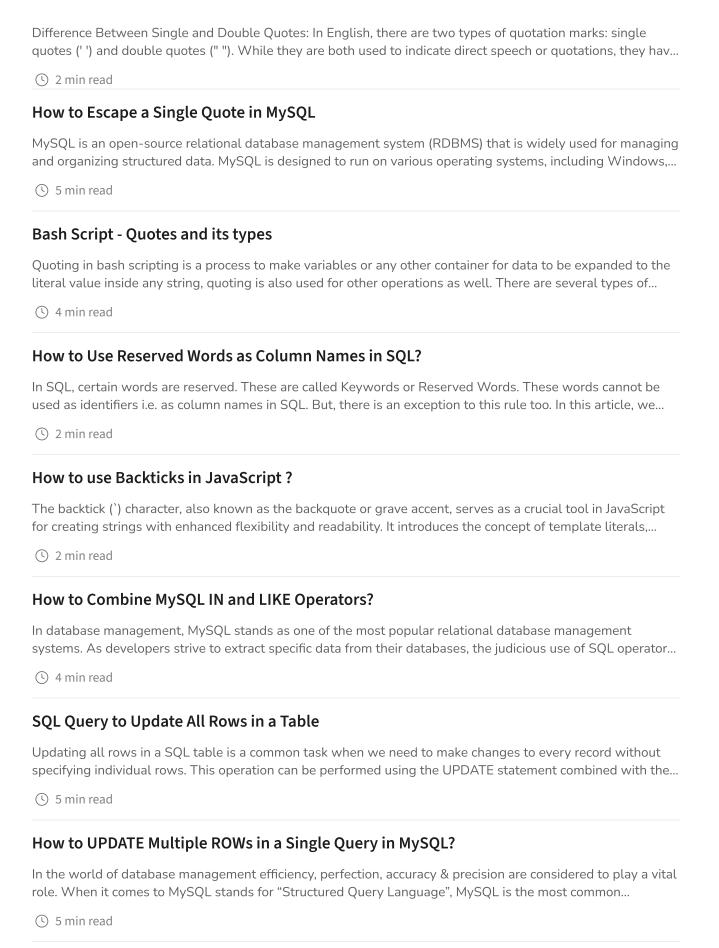
Single Quote, Double Quote, and Backticks in MySQL Queries

In MySQL, quotes and backticks are used to specify and handle different types of data and identifiers in queries. Single quotes (' ') are mainly for enclosing text values, double quotes (" ") are similar but less...

(4 min read



Difference Between Single and Double Quotes



How to Use SQL for Social Media Data Analysis.

Social media has enormous data possibilities for corporations, marketers, and researchers. SQL effectively extracts and alters data for analysis. Customer behavior and market trends were among the insights gained....

(6 min read

Article Tags:

SQL

Databases



Orporate & Communications Address:-A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305















Company

About Us

Legal

In Media

Contact Us

Advertise with us

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects