

Search...

[Databases](#) [SQL](#) [MySQL](#) [PostgreSQL](#) [PL/SQL](#) [MongoDB](#) [SQL Cheat Sheet](#) [SQL Interview Questic](#)

What is SQL?

Last Updated : 12 Mar, 2025

SQL stands for **Structured Query Language**. It is a standardized programming language used to **manage** and manipulate relational databases. It enables users to perform a variety of tasks such as querying data, creating and modifying database structures, and managing access permissions. SQL is widely used across various relational database management systems such as **MySQL**, **PostgreSQL**, Oracle, and SQL Server.

In this article, we will learn about what SQL is, and its **functionality**, structure, and practical uses. We will also learn its **characteristics**, rules, commands, etc from basic to advanced technical features.

Table of Content

- [What is SQL?](#)
- [Components of a SQL System](#)
- [What are the Characteristics of SQL?](#)
- [How SQL Works?](#)
- [Rules for Writing SQL Queries](#)
- [What are SQL Commands?](#)
- [Benefits of SQL](#)
- [Limitations of SQL](#)
- [SQL Use Cases](#)
- [Conclusion](#)

What is SQL?

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

a small user database or analyzing **terabytes** of sales records, SQL allows efficient querying, **updating**, and **management** of relational databases.

When data needs to be retrieved from a database, SQL is used to construct and send the request. The Database Management System (DBMS) processes the **SQL query**, retrieves the requested data, and returns it to the user or application. Instead of specifying step-by-step procedures, SQL statements describe what data should be retrieved, **organized**, or modified, allowing the DBMS to handle how the operations are executed efficiently.

In common usage, SQL encompasses **DDL** and **DML** commands for **CREATE**, **UPDATE**, **MODIFY**, or other operations on database structure.

SQL History

- *SQL was invented in 1970s and was first commercially distributed by Oracle.*
- *The original name was given by IBM as Structured English Query Language, abbreviated by the acronym SEQUEL.*

Components of a SQL System

A SQL system consists of several key components that work together to enable efficient data storage, **retrieval**, and manipulation. Understanding these components is crucial for mastering **SQL** and its role in relational database systems. Some of the Key components of a SQL System are:

- **Databases:** Databases are structured collections of data organized into tables, rows, and columns. Databases serve as repositories for storing

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

- **Tables:** Tables are the fundamental building blocks of a database, consisting of rows (records) and columns (attributes or fields). Tables ensure data integrity and consistency by defining the structure and relationships of the stored information.
- **Queries:** Queries are SQL commands used to interact with databases. They enable users to retrieve, update, insert, or delete data from tables, allowing for efficient data manipulation and retrieval.
- **Constraints:** Constraints are rules applied to tables to maintain data integrity. [Constraints](#) define conditions that data must meet to be stored in the database, ensuring accuracy and consistency.
- **Stored Procedures:** Stored procedures are pre-compiled SQL statements stored in the database. [Stored procedures](#) can accept parameters, execute complex operations, and return results, enhancing efficiency, reusability, and security in database management.
- **Transactions:** Transactions are groups of SQL statements that are executed as a single unit of work. [Transactions](#) ensure data consistency and integrity by allowing for the rollback of changes if any part of the transaction fails.

Some other important components include:

- Data Types
- Indexes
- Views
- Security and Permissions
- Joins

***SQL Injection** is a cyberattack where malicious SQL queries are injected into input fields to manipulate a database, enabling unauthorized access, data modification, or corruption. Using parameterized queries and input validation helps prevent such attacks.*

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

What are the Characteristics of SQL?

- **User-Friendly and Accessible:** SQL is designed for a broad range of users, including those with minimal programming experience, making it approachable for non-technical individuals.
- **Declarative Language:** As a non-procedural language, SQL allows users to specify what data is needed rather than how to retrieve it, focusing on the desired results rather than the retrieval process.
- **Efficient Database Management:** SQL enables the creation, modification, and management of databases efficiently, saving time and simplifying complex database operations.
- **Standardized Language:** Based on ANSI (American National Standards Institute) and ISO (International Organization for Standardization) standards, SQL ensures consistency and stability across various database management systems (DBMS).
- **Command Structure:** SQL does not require a continuation character for multi-line queries, allowing flexibility in writing commands across one or multiple lines.
- **Execution Mechanism:** Queries are executed using a termination character (e.g., a semicolon ;), enabling immediate and accurate command processing.
- **Built-in Functionality:** SQL includes a rich set of built-in functions for data manipulation, aggregation, and formatting, empowering users to handle diverse data-processing needs effectively.

How SQL Works?

Structured Query Language (SQL) operates on a server machine, where it processes database queries and returns results efficiently. Below are the key software components involved in the SQL execution process.

1. **Input:** The process begins when a user submits an SQL query through a database interface or application. This query typically specifies the

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

2. **Parsing:** The query is passed to the query processor, which breaks it into smaller units called tokens. These tokens represent keywords, table names, column names, and other elements of the query. The processor then validates the syntax against SQL standards and the database schema to ensure the query is well-formed and executable.
3. **Optimization:** After parsing, the query is handed to the optimizer, which evaluates multiple ways to execute the query. The optimizer considers factors like indexes, table statistics, and available resources to generate the most efficient execution plan. This step ensures that the query runs with minimal resource consumption and maximum performance.
4. **Execution:** The execution engine follows the plan provided by the optimizer. It interacts with the storage engine, which retrieves, manipulates, or updates the required data from the database tables. During this step, SQL statements like SELECT, INSERT, UPDATE, or DELETE are translated into actions performed on the underlying data.
5. **Output:** Once the execution engine processes the query, the result is formatted and returned to the user. Depending on the query type, the output could be a result set (for SELECT queries) or an acknowledgment of the operation (for INSERT, UPDATE, or DELETE queries).

By combining these steps, SQL ensures the seamless interaction between users and relational databases, enabling efficient data manipulation and retrieval.

Rules for Writing SQL Queries

There are certain rules for SQL which would ensure consistency and functionality across databases. By following these rules, queries will be well formed and well executed in any database.

- **Statement Termination:** Every SQL statement ends with a semicolon (;), signaling the DBMS to execute the command.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

sensitive depending on the DBMS.

- **Whitespace Flexibility:** SQL statements can span multiple lines, but keywords and identifiers must be separated by at least one space.
- **Unique Identifiers:** Reserved words (e.g., SELECT, FROM) cannot be used as table or column names unless enclosed in double quotes (") or backticks (`), depending on the DBMS.
- **Comments:** Comments enhance readability:
 - Single-line comments: —
 - Multi-line comments: /* ... */
- **Data Integrity:** Constraints like NOT NULL, UNIQUE, and PRIMARY KEY must be defined correctly to maintain data consistency.
- **String Literals:** String values must be enclosed in single quotes (').
- **Valid Identifiers:** Table and column names must:
 - Begin with an alphabetic character.
 - Contain up to 30 characters.
 - Avoid special characters except underscores (_).

By following these rules, SQL users ensure reliable query execution and maintainable database structures.

What are SQL Commands?

Structured Query Language (SQL) commands are standardized instructions used by developers to interact with data stored in **relational databases**. These commands allow for the **creation**, manipulation, retrieval, and **control of data**, as well as database structures. SQL commands are categorized based on their specific functionalities:

1. Data Definition Language

These commands are used to **define** the structure of database objects by **creating, altering, and dropping** the database objects. Based on the needs

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

using DDL. The [CREATE](#) command, for instance, is used by the database engineer to create database objects like tables, views, and indexes.

Command	Description
CREATE	Creates a new table, a view on a table, or some other object in the database.
ALTER	Modifies an existing database object, such as a table
DROP	Deletes an entire table, a view of a table, or other objects in the database

2. Data Manipulation Language

A relational database can be updated with new data using data manipulation language ([DML](#)) statements. The [INSERT](#) command, for instance, is used by an application to add a new record to the database.

Command	Description
INSERT	Creates a record.
UPDATE	Modifies records.
DELETE	Deletes records.

3. Data Query Language

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

used by software programs to filter and return particular results from a SQL table.

4. Data Control language

DCL commands manage user access to the database by granting or revoking permissions. Database administrators use [DCL](#) to enforce security and control access to database objects.

Command	Description
GRANT	Gives a privilege to the user.
REVOKE	Takes back privileges granted by the user.

5. Transaction Control Language

TCL commands manage transactions in relational databases, ensuring data integrity and consistency. These commands are used to commit changes or roll back operations in case of errors.

Command	Description
COMMIT	Saves all changes made during the current transaction on a permanent basis. Some databases provide an auto-commit feature, which can be configured using settings.
ROLLBACK	Reverts changes made during the current transaction, ensuring no unwanted changes are saved.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Command	Description
SAVEPOINT	Sets a point within a transaction to which changes can be rolled back, allowing partial rollbacks

Benefits of SQL

- **Efficiency:** SQL is designed to handle complex queries and large datasets with optimal performance, making data retrieval and manipulation seamless.
- **Standardization:** As an ANSI and ISO standard language, SQL provides a universal method to interact with relational databases across platforms.
- **Scalability:** SQL supports databases ranging from small-scale applications to enterprise-level systems, ensuring smooth operations regardless of size.
- **Flexibility:** SQL can be extended with procedural programming (e.g., PL/SQL, T-SQL) to build complex business logic and custom functions.

Limitations of SQL

- **Complexity in Advanced Operations:** Advanced functionalities such as indexing, query optimization, and performance tuning require in-depth technical knowledge.
- **Scalability Concerns:** SQL performs best with structured data; handling unstructured data or massive distributed systems can pose challenges.
- **Platform-Specific Variations:** While SQL is standardized, many databases implement unique extensions, leading to portability and compatibility issues.

SQL Use Cases

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

- **Banking:** Analyze transaction histories and generate financial reports.
- **Web Development:** Power dynamic websites with user-specific content.
- **Machine Learning and Data Science:** Combine SQL databases with tools like Python, R, and TensorFlow to streamline machine learning workflows.

Conclusion

SQL(Structured Query Language) is a programming language designed for managing and manipulating data stored in relational databases. It is used for interacting with DBMS like MySQL, SQL Server, Oracle, and PostgreSQL.

In this article we have covered about SQL and understood it's characteristics, rules to write SQL queries, commands, uses and many important concepts. We also covered the SQL injection and how it can be harmful for database security. After completing this guide, you will be equipped with all necessary information about SQL.

[Comment](#)[More info](#)[Advertise with us](#)

Next Article

[SQL Data Types](#)

Similar Reads

SQL Clauses

Structured Query Language (SQL) is a powerful language used to manage and manipulate relational databases. One of the essential features of SQL i...

15+ min read

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).