Introduction to SQL

The main objective of this section to develop a solid understanding of SQL (Structured Query Language), its purpose, and its role in working with relational databases. Explore the various components and constructs of SQL, and gain practical skills in writing SQL queries for data manipulation and retrieval.

Concept Overview

Topics

* Introduction to Relational Database Concepts
* Introduction to SQL
* SQL Syntax and Structure
* Basic SQL Operations
* Installing MySQL

Learning Objectives

* Understand the purpose and significance of SQL (Structured Query Language).
* Learn the basic syntax of SQL queries for data retrieval, manipulation, and database schema management.

Introduction to Relational Database Concepts

Tables

* A table is a fundamental component of a relational database, which stores data in a structured format.
* Tables consist of rows (records) and columns (fields).
* Each table represents a specific entity or concept, such as “Customers,” “Orders,” or “Products.”
* Tables are defined with a schema that specifies the names of columns, data types, and constraints.

Rows

* Rows, also known as records, represent individual instances or occurrences of the entity represented by the table.
* Each row in a table contains a unique set of values for the columns defined in the table’s schema.
* For example, in a “Customers” table, each row would represent a single customer with their respective data (name, address, email, etc.).

Columns

* Columns, also known as fields, define the attributes or characteristics of the entity represented by the table.
* Each column has a name and a specific data type (e.g., text, numeric, date, etc.) that defines the type of data it can store.
* Columns often have constraints, such as “NOT NULL” to ensure that a value is always provided, or “UNIQUE” to prevent duplicate values.
* For example, in a “Customers” table, columns might include “CustomerID,” “FirstName,” “LastName,” “Email,” and “Phone.”

Relationships

* Relational databases allow tables to be related to each other through the use of primary keys and foreign keys.
* A primary key is a column or a combination of columns that uniquely identifies each row in a table.
* A foreign key is a column or a combination of columns that references the primary key of another table, establishing a relationship between the two tables.
* Relationships enable data to be efficiently queried and combined from multiple tables, ensuring data integrity and eliminating redundancy.

Introduction to SQL

SQL (Structured Query Language) is a programming language designed for managing and manipulating relational databases. It serves as the standard language for interacting with SQL databases, such as MySQL, PostgreSQL, Oracle, and Microsoft SQL Server. SQL provides a standardized way to perform various operations on databases, including creating, modifying, and querying data. It uses a structured, English-like syntax, making it relatively easy to learn and understand, even for non-programmers

SQL is a declarative language, meaning that you specify what data you want to retrieve or manipulate, rather than providing step-by-step instructions on how to do it. It is widely adopted and supported by various database management systems (DBMS), ensuring portability and compatibility across different platforms. SQL is essential for data analysts, database administrators, and developers working with relational databases and data-driven applications.

SQL Syntax and Structure

SQL follows a specific syntax and structure, consisting of various statements and clauses. Understanding the syntax and structure is crucial for writing correct and efficient SQL queries.

SQL statements can be categorized into several types, including:

* **Data Definition Language (DDL)**: Used for creating, modifying, and deleting database objects like tables and indexes (e.g., CREATE, ALTER, DROP).
* **Data Manipulation Language (DML)**: Used for inserting, updating, and deleting data within tables (e.g., INSERT, UPDATE, DELETE).
* **Data Query Language (DQL)**: Used for retrieving data from tables (e.g., SELECT).
* **Data Control Language (DCL)**: Used for managing user access and permissions (e.g., GRANT, REVOKE).

SQL Data Types

SQL supports various data types to store different kinds of data in a database. Here are some common SQL data types:

* **Numeric Types**: Used to store numbers. Examples: INTEGER, SMALLINT, BIGINT, DECIMAL, NUMERIC, FLOAT.
* **Character Types**: Used to store text data. Examples: CHAR, VARCHAR, TEXT.
* **Date and Time Types**: Used to store date and time values. Examples: DATE, TIME, DATETIME, TIMESTAMP.
* **Boolean Type**: Used to store logical values (true or false). Example: BOOLEAN.

Data types are important because they define the kind of data that can be stored in a column and ensure data integrity by preventing invalid data from being inserted.

Basic SQL Operations

Creating a Table

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Email VARCHAR(100) UNIQUE,

HireDate DATE

);

This statement creates a new table named “Employees” with columns for EmployeeID (as the primary key), FirstName, LastName, Email (with a unique constraint), and HireDate. The data types for each column are specified, and constraints like NOT NULL and UNIQUE are applied.

**Inserting Data into a Table**

INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, HireDate)

VALUES (1, 'John', 'Doe', 'john.doe@example.com', '2022-05-15');

INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, HireDate)

VALUES (2, 'Jane', 'Smith', 'jane.smith@example.com', '2021-08-22'),

(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '2023-02-01');

These statements insert new records (rows) into the “Employees” table. The first statement inserts a single record, while the second statement demonstrates how to insert multiple records in one statement by separating the value sets with commas.

**Selecting Data from a Table**

SELECT \* FROM Employees;

This statement retrieves all columns and rows from the “Employees” table.

SELECT FirstName, LastName, Email FROM Employees;

This statement retrieves only the FirstName, LastName, and Email columns from the “Employees” table.

Installing MySQL

Installing on Windows

Download MySQL

Download MySQL installer from [https://dev.mysql.com/downloads/installer/](https://savanna.alxafrica.com/rltoken/__e9lsqjaqU5daFCW2TTWA) and execute it. On this page you can select the version you would like to install. You can choose the web installer or the offline installer.

Install MySQL

Follow the below steps to install MySQL on Windows

Running MySQL

Installing MySQL on Linux

Below instruction is for **Ubuntu** distribution

1. Update the package index
2. suo apt update
3. Install mysql-server
4. sudo apt install mysql-server
5. Ensure the server is running - it should start automatically but to make sure it is running execute the below command
6. sudo systemctl start mysql.service
7. Run MySQL
8. sudo mysql
9. Setup password for root user. Copy paste the below command and change the password string to your password
10. sudo mysql
11. mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password';
12. Access MySQL with your password with the following command and enter your password when asked.
13. mysql -u root -p

Practice Exercises

* **Exercise 1:** Write a SQL statement to create a table called “Employees” with columns for employee ID, first name, last name, department, and hire date.
* **Exercise 2:** Insert a few sample records into the “Employees” table using SQL INSERT statements.
* **Exercise 3:** Write a SQL query to retrieve all employees from the “Employees” table who were hired after a specific date, ordered by their last name in ascending order.

Additional Resources

* [What is SQL?](https://savanna.alxafrica.com/rltoken/jT1nEzy8KMKB6ObKUMUMqw)
* [SQLBolt](https://savanna.alxafrica.com/rltoken/oN4XIphbJmSYf2C7S_vwMg)

Top of Form

Bottom of Form