CHAMELEON

FOR OUR SMARTER WORLD

SQL Documentation

By

ANENA GHOSH

S223792684

TABLE OF CONTENTS

Introduction:

Purpose of SQL Documentation:

SQL documentation is a complete resource for learning and applying SQL (Structured Query Language) to manage and manipulate relational databases. Developers, database administrators, and anybody involved in data administration must have clear, detailed documentation for reference. The documentation contributes to effective and accurate database operations, ranging from basic data retrieval to complicated transactions and data design.

Audience of this documentation:

- Students and Learners: Gaining a fundamental understanding of SQL and its applications.
- Developers: Writing and optimizing SQL queries to interact with databases.
- Data Analysts: Retrieving and analyzing data to support business decisions.
- Database Administrators (DBAs): Those who manages database structures, security, and performance.

Importance of SQL Documentation:

- Consistency and Accuracy: Ensures that all users adhere to the same standards and practices, resulting in consistent and accurate database operations.
- Efficiency: Allows users to easily discover information and answers, decreasing time spent troubleshooting and increasing productivity.
- Learning and Development: Offers an organized learning path for novice users, as well as a resource for experienced professionals looking to improve their skills.
- Collaboration: Provides a common knowledge base, allowing team members to communicate and collaborate more effectively.

SQL Basics:

What is SQL:

SQL (Structured Query Language) is a programming language that is specifically developed to manage and manipulate relational databases. SQL is used to query data, update records, insert new data, and delete old data from databases. It is the standard language for relational database management systems (RDBMS), including MySQL, PostgreSQL, Oracle, and Microsoft SQL Server.

Key components of SQL:

- Data Query Language (DQL): Used to retrieve data from the database.
  - SELECT: Extracts data from a database.
- Data Manipulation Language (DML): Used to modify data in the database.
  - INSERT: It adds new data to the database.
  - UPDATE: It modifies existing data.
  - DELETE: It removes data.
- Data Definition Language (DDL): Used to define and manage database structures.
  - CREATE: It creates new tables, databases, indexes, and other database objects.
  - ALTER: It modifies existing database structures.
  - DROP: It deletes database objects.

- Data Control Language (DCL): It is used to control access to data in the database.
  - GRANT: It gives user access privileges.
  - REVOKE: It removes user access privileges.

Example:

Table: Employees

id	first_name	last_name	salary
1	Anabelle	Smith	60000
2	Mary	Warne	70000
3	Jean	Johnson	90000

Retrieving Data: To retrieve all records from the employees table, you use the SELECT statement-

SELECT * FROM employees;
Inserting Data: To add a new employee to the employees table, you use the INSERT statement-
INSERT INTO employees (first_name, last_name, salary) VALUES ('Bob', 'Brown', 55000);
Output: After execution, the employees table will include the new record:

id	First_name	Last_name	Salary
1	Anabelle	Smith	60000
2	Mary	Warne	70000
3	Jean	Johnson	90000
4	Bob	Brown	550000

Basic SQL Query Structures:

SELECT column1, column2 FROM table_name;

Example: SELECT name, age FROM users;

SQL Data Types:

- Numeric Types: INT, FLOAT, DECIMAL
- Character Types: CHAR, VARCHAR, TEXT
- Date and Time Types: DATE, TIME, DATETIME
- Boolean Types: BOOLEAN

SQL Commands:

- SELECT: It retrieves data from a database.
- INSERT: It inserts new data into a database.
- UPDATE: It updates existing data within a database.
- DELETE: It deletes data from a database.
- CREATE: It creates a new table or database.
- ALTER: It modifies an existing database structure.
- DROP: It deletes tables or databases.

SQL Constraints:

- NOT NULL: It ensures that a column cannot have a NULL value.
- UNIQUE: It ensures all values in a column are unique.
- PRIMARY KEY: A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.

- FOREIGN KEY: It prevents actions that would destroy links between tables.
- CHECK: It ensures that the values in a column satisfy a specific condition.
- DEFAULT: It sets a default value for a column when no value is specified.

SQL Functions:

- Aggregate Functions: COUNT, SUM, AVG, MAX, MIN
- String Functions: CONCAT, LENGTH, SUBSTRING
- Date Functions: NOW, CURDATE, DATEADD

Example of SQL Functions:
Query:
SELECT COUNT(*), AVG(salary)
FROM employees
WHERE department = 'Sales';
It Counts the number of employees and calculates the average salary in the Sales department.

Best Practices:

- We can use meaningful table and column names.
- We should keep queries simple and readable.
- We can use joins instead of subqueries where possible.
- Index columns that are frequently used in WHERE clauses.
- We should regularly backup your database.

Examples:

1. Updating Data (UPDATE):
   UPDATE employees SET salary = 52000 WHERE id = 1;
   Output:

id	first_name	last_name	salary
1	Anabelle	Smith	52000
2	Mary	Warne	70000
3	Jean	Johnson	90000
4	Bob	Brown	550000

2. Deleting Data (DELETE):
   DELETE FROM employees WHERE id = 2;
   Output:

Id	first_name	last_name	salary
1	Anabelle	Smith	52000
3	Jean	Johnson	90000
4	Bob	Brown	550000

Tools and Resources:

- SQL Editors and IDEs: Tools like MySQL Workbench, SQL Server Management Studio.
- Online Resources: Documentation from database vendors, SQL tutorials, forums.

References:

- https://dev.mysql.com/doc/
- https://docs.microsoft.com/en-us/sql/sql-server/
- https://www.bing.com/ck/a?!&&p=19db7f50dda8bbfdJmltdHM9MTcxNTkwNDAwMCZpZ3VpZD0xZTk2ZTQzNi0yOGE4LTZkMWUtMTE3Yi1mN2U3Mjk3ZDZjZmUmaW5zaWQ9NTQ1NQ&ptn=3&ver=2&hsh=3&fclid=1e96e436-28a8-6d1e-117b-f7e7297d6cfe&psq=GeeksforGeeks+SQL+Tutorial&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvc3FsLXR1dG9yaWFsLw&ntb=1
- https://www.w3schools.com/sql/