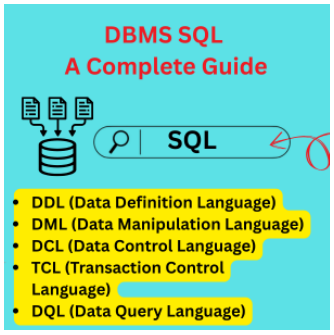

DBMS SQL
A Complete Guide
SQL
DDL (Data Definition Language)
DML (Data Manipulation Language)
DCL (Data Control Language)
TCL (Transaction Control Language)
DQL (Data Query Language)

Among all languages, Structured Query Language represents the one that is completely standard for the purposes of manipulation and management of a particular database under the control of a database management system or DBMS. It states the features that are supposed to be there in a user interface with the help of which a user can create, update, delete, and retrieve data with the most efficient manner. This particular tutorial is about different concepts, commands, and best practices that are significant while working through SQL representing learn SQL in DBMS.

"SQL is the standard language when it comes to the basic management and manipulation of databases under a Database Management System (DBMS). It allows one to create, update, delete, and access data in an efficient manner. Such kind of material will discuss the essential concepts, commands, and best practices of SQL."

Introduction to Javatpoint DBMS sql tutorials

DBMS is the most important part of modern computing as it considers efficient storage, retrieval, and management of data. This tutorial encapsulates all possible DBMS concepts like SQL, normalization, relational algebra, transactions, and keys. This is supposed to help students and candidates excel in the field of DBMS.

Javatpoint DBMS Tutorial

DBMS stands for Database Management System, which is software used to create and manage databases. The best sample DBMSs are MySQL, PostgreSQL, Oracle, and SQL Server. The tutorial will discuss the following topics with respect to DBMS topics:

- Introduction to DBMS
- Types of DBMS
- SQL in DBMS
- Keys in DBMS
- Normalization in DBMS
- Transactions in DBMS
- Relational Algebra in DBMS
- DBMS Interview Questions and MCQs

SQL in DBMS An Overview

SQL is an important query language in most relational database management systems (RDBMS), such as MySQL, PostgreSQL, Oracle, or SQL Server. The commands are commonly classified into:

DDL – Data Definition Language

DML – Data Manipulation Language

DCL – Data Control Language

TCL – Transaction Control Language

DQL – Data Query Language

SQL Commands in DBMS Javatpoint

1. Data Definition Language (DDL)

DDL commands define and manage database structures. Key DDL commands include:

CREATE – Creates a database or table.

```
CREATE TABLE Employees (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT,
    Department VARCHAR(50)
);
```

ALTER – Modifies an existing table.

```
ALTER TABLE Employees ADD Salary DECIMAL(10,2);
```

DROP – Deletes a table.

```
DROP TABLE Employees;
```

2. Data Manipulation Language (DML)

DML commands are used to manipulate data within tables.

INSERT – Adds new records to a table.

```
INSERT INTO Employees (ID, Name, Age, Department)
VALUES (1, 'John Doe', 30, 'IT');
```

UPDATE – Modifies existing records.

```
UPDATE Employees SET Age = 31 WHERE ID = 1;
```

DELETE – Removes records from a table.

```
DELETE FROM Employees WHERE ID = 1;
```

3. Data Query Language (DQL)

DQL commands retrieve data from the database.

SELECT – Fetches records from a table

```
SELECT * FROM Employees;
```

Using WHERE clause:

```
SELECT Name, Age FROM Employees WHERE Department = 'IT';
```

Using ORDER BY clause:

```
SELECT * FROM Employees ORDER BY Age DESC;
```

4. Data Control Language (DCL)

DCL commands control user access to the database.

GRANT – Gives privileges to users.

```
GRANT SELECT, INSERT ON Employees TO 'user1';
```

REVOKE – Removes privileges.

```
REVOKE INSERT ON Employees FROM 'user1';
```

Transaction Control Language (TCL)

TCL commands manage database transactions to ensure data consistency.

COMMIT – Saves all changes permanently.

```
COMMIT;
```

ROLLBACK – Undoes changes before committing.]

```
ROLLBACK;
```

SAVEPOINT – Creates a savepoint within a transaction.

```
SAVEPOINT sp1;
```

Javatpoint SQL Joins in DBMS

Joins in SQL are used to join data from more than one table based on a common attribute. They are helpful in retrieving meaningful data from relational databases. Here is a brief explanation of the various types of SQL joins:

INNER JOIN: Returns only the records that have matching values in both tables.

```
SELECT Employees.Name, Departments.DeptName
FROM Employees
INNER JOIN Departments ON Employees.DepartmentID = Departments.ID;
```

Retrieves only the employees that have a corresponding department.

LEFT JOIN (LEFT OUTER JOIN): Returns all records from the left table and matching records from the right table. If no match is found, NULL values are returned for columns from the right table.

```
SELECT Employees.Name, Departments.DeptName
FROM Employees
LEFT JOIN Departments ON Employees.DepartmentID = Departments.ID;
```

Ensures all employees are listed, even if they don't belong to a department.

RIGHT JOIN (RIGHT OUTER JOIN): Returns all records from the right table and matching records from the left table. If no match is found, NULL values are returned for columns from the left table.

```
SELECT Employees.Name, Departments.DeptName
FROM Employees
RIGHT JOIN Departments ON Employees.DepartmentID = Departments.ID;
```

Ensures all departments are listed, even if they don't have employees.

FULL JOIN (FULL OUTER JOIN): Returns all records when there is a match in either table. If there's no match, NULL values are returned for columns from the missing side.

```
SELECT Employees.Name, Departments.DeptName
FROM Employees
FULL JOIN Departments ON Employees.DepartmentID = Departments.ID;
```

Retrieves all employees and all departments, whether they have a match or not.

SQL Constraints in DBMS

Constraints play a vital role in dbms by enforcing rules on data in tables, thereby maintaining accuracy and integrity. They also ensure data consistency and prevent invalid entries. Here is a detailed description about these:

NOT NULL – Ensures that a column cannot have NULL values.

```
CREATE TABLE Students (
    ID INT PRIMARY KEY,
    Name VARCHAR(50) NOT NULL
);
```

This prevents inserting records without a name.

UNIQUE – Ensures all values in a column are unique.

```
CREATE TABLE Employees (
    ID INT PRIMARY KEY,
    Email VARCHAR(100) UNIQUE
);
```

No two employees can have the same email.

PRIMARY KEY – A combination of NOT NULL and UNIQUE, uniquely identifying each record.

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    ProductName VARCHAR(50)
);
```

Ensures each order has a unique identifier.

FOREIGN KEY – Establishes relationships between tables by referencing a primary key from another table.

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(ID)
);
```

Prevents adding orders with non-existing customers.

CHECK – Ensures values meet a specific condition.

```
CREATE TABLE Employees (
    ID INT PRIMARY KEY,
    Age INT CHECK (Age >= 18)
);
```

Ensures employees are at least 18 years old.

SQL Indexing in DBMS

Indexes improve query performance by allowing faster data retrieval.

CREATE INDEX – Creates an index on a table.

```
CREATE INDEX idx_name ON Employees (Name);
```

UNIQUE INDEX – Ensures uniqueness.

```
CREATE UNIQUE INDEX idx_unique_name ON Employees (Name);
```

Transactions in DBMS SQL

A sequence of operations functioning as a single logical unit of work constitutes a transaction. Transactions are defined by the ACID properties to maintain integrity and reliability of data:

Atomicity: One part of the transaction could go wrong, thereby making the whole transaction invalid with rollback.

Consistency: The database remains consistent before and after the transaction.

Isolation: Simultaneously running transactions do not hinder or interfere with one another.

Durability: The transaction, once committed, will remain forever in the database.

```
START TRANSACTION;
UPDATE Employees SET Age = 35 WHERE ID = 2;
COMMIT;
```

`START TRANSACTION`: Begins the transaction.

`UPDATE`: Modifies a record.

`COMMIT`: Saves the changes permanently.

If something goes wrong before `COMMIT`, we can use `ROLLBACK` to undo changes:

```
ROLLBACK;
```

Javatpoint DBMS SQL Interview Questions

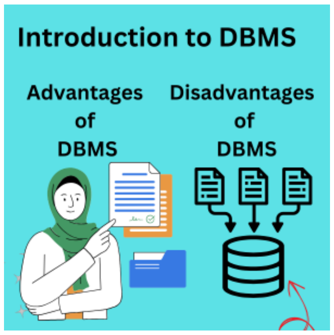
1. What is the difference between DELETE, TRUNCATE, and DROP?
2. Explain the different types of SQL joins with examples.
3. What are the different types of indexes in SQL, and when should they be used?
4. What are stored procedures and triggers in SQL? Provide examples.
5. How do you optimize SQL queries for better performance?
6. What is the difference between WHERE and HAVING clauses in SQL?
7. Explain normalization and denormalization in DBMS.
8. What are ACID properties in DBMS transactions, and why are they important?
9. What is the difference between clustered and non-clustered indexes?
10. How does the GROUP BY clause work in SQL? Provide an example.

Here are some commonly asked SQL interview questions:

1. What is the difference between DELETE, TRUNCATE, and DROP?
2. Explain the different types of SQL joins.
3. What are the different types of indexes in SQL?
4. What are stored procedures and triggers in SQL?
5. How do you optimize SQL queries for performance?

← Previous Post

Related Posts



Javatpoint Dbms Tutorials Beginners to Advance level Guide 2025

dbms / March 19, 2025

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..

Name*

Email*

Website

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment »