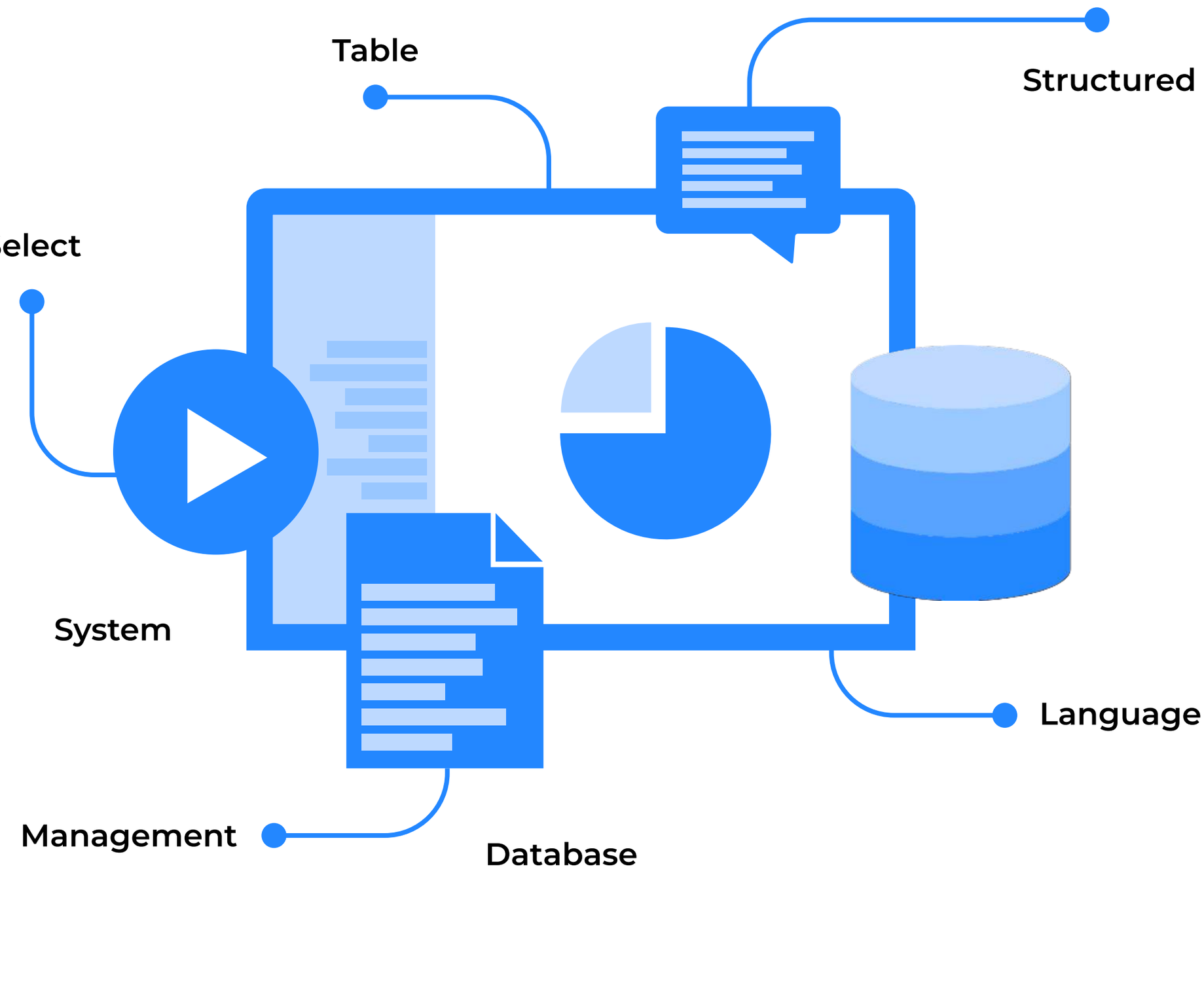
BOSSCODER ACADEMY

A Guide to Top 10

SQL COMMANDS



Introduction to SQL

Today the universal language for managing relational database is SQL. It's both powerful and relatively easy to learn.

However, SQL can get tricky as your data structures become more complex. But you can be a pro at it with some practice.

So, here are essential commands that you'll use frequently. This doc doesn't contains all the commands but includes few of the essential commands:

- DDL (Data Definition Language),
- DML (Data Manipulation Language),
- DQL (Data Query Language), and
- DCL (Data Control Language).

Data Definition Language (DDL)

DDL is a subset of SQL primarily used for defining and managing the structure of a database.

- DDL commands are non-transactional, meaning they immediately affect the database structure.
- Key DDL commands include CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, DROP INDEX, CREATE VIEW, and DROP VIEW.

CREATE TABLE Command

- **Syntax:** `CREATE TABLE table_name (columnname1 datatype, columnname2 datatype, ...);`
- **Purpose:** Creates a new table with specified columns and data types.
- **Example:** `CREATE TABLE employees (id INT, name VARCHAR(50), salary DECIMAL);`

This SQL command creates a table named "employees" with three columns:

`id (INT)`	An integer column to store unique employee IDs.
`name (VARCHAR(50))`	A variable-length character column for employee names with a maximum length of 50 characters.
`salary (DECIMAL)`	A decimal number column to store employee salaries.

ALTER TABLE Command

- **Syntax:** `ALTER TABLE table_name ADD|MODIFY|DROP column_name datatype;`

- **ADD :**
    - Example: `ALTER TABLE employees ADD department VARCHAR(30);`

    - This command adds a new column named "department" to the "employees" table with a data type of VARCHAR(30).
- **MODIFY:**
    - Example: `ALTER TABLE employees MODIFY salary DECIMAL(10, 2);`
    - In this example, the command modifies the data type of the "salary" column in the "employees" table. It changes the data type to DECIMAL with a precision of 10 and a scale of 2.
- **DROP:**
    - Example: `ALTER TABLE employees DROP department;`
    - The DROP command removes the "department" column from the "employees" table.

DROP TABLE Command

- **Syntax:** `DROP TABLE table_name;`
- **Purpose:** Deletes an entire table and all its data.
- **Example:** `DROP TABLE employees;`

The purpose of this command is to remove the entire "employees" table, including all the data it contains. This action permanently deletes the table and all the records (rows) stored in it.

CREATE INDEX Command

- **Syntax:** `CREATE INDEX index_name ON table_name (column1, column2, ...);`
- **Purpose:** This command is used to create an index on one or more columns within a database table, improving the speed of data retrieval operations.
- **Example:** `CREATE INDEX idx_name ON employees (id);`

The provided example creates an index named "idx_name" on the "employees" table, specifically on the "id" column. This indexing will optimize the retrieval of data based on the "id" column.

DROP INDEX Command

- **Syntax:** `DROP INDEX index_name;`

- **Purpose:** The primary purpose of the DROP INDEX command is to eliminate an existing index associated with a database table.
- **Example:** To illustrate, let's say you want to remove an index named "idx_name" from a table. You would execute the following command: `DROP INDEX idx_name;`.

CREATE VIEW Command

- **Syntax:** `CREATE VIEW view_name AS SELECT column1, column2 FROM table_name WHERE condition;`
- **Purpose:**
  - This command serves the purpose of establishing a virtual table that is generated from the results of a SELECT query.
  - It offers a way to encapsulate complex queries or provide a simplified view of your data without physically storing it.

- **Example**: For instance, you can create a view named "high_salary_employees" by executing the following command: `CREATE VIEW high_salary_employees AS SELECT name, salary FROM employees WHERE salary > 50000;`. This view gathers and displays the names and salaries of employees with salaries exceeding Rs. 50,000, providing a convenient way to access this specific dataset.

DROP VIEW Command

- **Syntax:** `DROP VIEW view_name;`
- **Purpose:**
    - The primary purpose of the DROP VIEW command is to remove a view from the database and eliminate its associated definition.
    - This action effectively erases the virtual table, which can be beneficial when a view is no longer required or needs to be replaced.
- **Example:** As an illustration, consider the need to remove a view named "high_salary_employees." You can achieve this by executing the following command: `DROP VIEW high_salary_employees;`

TRUNCATE Command

- **Syntax:** `TRUNCATE TABLE table_name;`
- **Purpose:** The TRUNCATE command is designed to provide a fast and efficient way to delete all records from a table, maintaining the table structure for future data insertion.
- **Example:** As an example, to clear all data from a table called "employee_data," you can execute the command: `TRUNCATE TABLE employee_data;`.

RENAME Command

- **Syntax:** `RENAME old_table_name TO new_table_name;`
- **Purpose:** The primary purpose of the RENAME command is to alter the name of an existing table in the database.
- **Example:** For instance, if you wish to rename a table from "old_employees" to "new_employees," you can execute the command: `RENAME old_employees TO new_employees;`

Data Manipulation Language (DML)

DML, or Data Manipulation Language, is a subset of SQL used to manage and manipulate data within a database.

SELECT Statement

- **Syntax:** `SELECT column1, column2 FROM table WHERE condition;`
- **Purpose:** Retrieves data from a database.
- **Example:** `SELECT first_name, last_name FROM employees;`

This command selects the first name and last name of employees working in the Sales department.

SELECT - Retrieving Data

You can select specific columns or use '*' to retrieve all columns.

Filtering is done using the WHERE clause.

Example: `SELECT product_name, price FROM products WHERE category = 'Electronics';`

Here, we retrieve the product name and price from the 'products' table for items in the 'Electronics' category. The WHERE clause filters the results to include only Electronics products.

Example: `SELECT * FROM products;`

In this example, the `SELECT` statement is used with `*`, as the column list. When you use *, it's a shorthand way of selecting all columns from the specified table, in this case, the `products` table.

INSERT Statement

- **Syntax:** `INSERT INTO table (column1, column2, ...) VALUES (value1, value2, ...);`
- **Purpose:** Adds new data to a table.
- **Example:** `INSERT INTO customers (first_name, last_name, email) VALUES ('John', 'Doe', 'john.doe@email.com');`

Explanation: This command inserts a new customer, 'John Doe', into the 'customers' table with the specified first name, last name, and email.

INSERT - Adding Multiple Rows

You can insert multiple rows in a single statement in the following way.

Example:

```
INSERT INTO orders (order_id, customer_id,
order_date) VALUES (1, 101, '2023-10-18'), (2, 102,
'2023-10-19');
```

Multiple rows can be inserted in a single command. Here, two orders are added to the 'orders' table with their respective order IDs, customer IDs, and order dates.

UPDATE Statement

- **Syntax:** `UPDATE table SET column1 = value1, column2 = value2 WHERE condition;`
- **Purpose:** Modifies existing data in a table.

Example:

```
UPDATE products SET price = 49.99 WHERE product_id =
101;
```

This command updates the price of a product with 'product_id' 101 in the 'products' table to Rs. 49.99.

Example:

```
UPDATE employees SET salary = 60000 WHERE department
= 'HR';
```

The command updates the salary of employees in the 'HR' department to 60,000. The WHERE clause ensures that only HR department employees are affected.

DELETE Statement

- **Syntax:** `DELETE FROM table WHERE condition;`
- **Purpose:** Removes data from a table.

Example:

```
DELETE FROM customers WHERE customer_id = 105;
```

This command deletes the customer with 'customer_id' 105 from the 'customers' table.

Example:

```
DELETE FROM orders WHERE order_date < '2023-01-01';
```

The command deletes all orders with an 'order_date' before January 1, 2023.

It's essential to be cautious with DELETE as it permanently removes data.

Transaction Control Language (TCL)

- Transaction Control Language (TCL) commands are a subset of SQL commands used to manage and control database transactions.
- They include commands like COMMIT, ROLLBACK, and SAVEPOINT, allowing you to control the integrity and consistency of your database.

COMMIT Command

- **Syntax:** `COMMIT;`
- **Purpose:** Commits the current transaction, making all changes permanent in the database.
- **Example:**

```
BEGIN;
-- Perform SQL operations
COMMIT;
```

In this example, the BEGIN initiates a transaction, and COMMIT makes all changes within that transaction permanent.

ROLLBACK Command

- **Syntax:** `ROLLBACK;`
- **Purpose:** Rolls back the current transaction, undoing all changes made within the transaction.
- **Example:**

```
BEGIN;
-- Perform SQL operations
ROLLBACK;
```

When ROLLBACK is executed, all changes made within the transaction are undone, and the database returns to its previous state.

SAVEPOINT Command

- **Syntax:** `SAVEPOINT savepoint_name;`
- **Purpose:** Creates a savepoint within a transaction to which you can later roll back.

- Example:

```
BEGIN;
-- Perform SQL operations
SAVEPOINT sp1;
-- More SQL operations
ROLLBACK TO sp1;
```

SAVEPOINT creates a point in the transaction where you can later roll back, preserving changes made before the savepoint.

SET TRANSACTION Command

- **Syntax:** `SET TRANSACTION [transaction_properties];`
- **Purpose:** Sets various transaction properties such as isolation level or access mode.
- **Example:** `SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;`

This example sets the isolation level to SERIALIZABLE, which ensures that concurrent transactions do not interfere with the current one, providing high data consistency.

BOSSCODER ACADEMY

WHY BOSSCODER?

1000+ Alumni placed at Top Product-based companies.

More than **136% hike** for every **2 out of 3** working professional.

Average package of **24LPA.**

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

Lavanya
Meta

Course is very well **structured** and **streamlined** to crack any **MAANG** company

Rahul .
Google

EXPLORE MORE