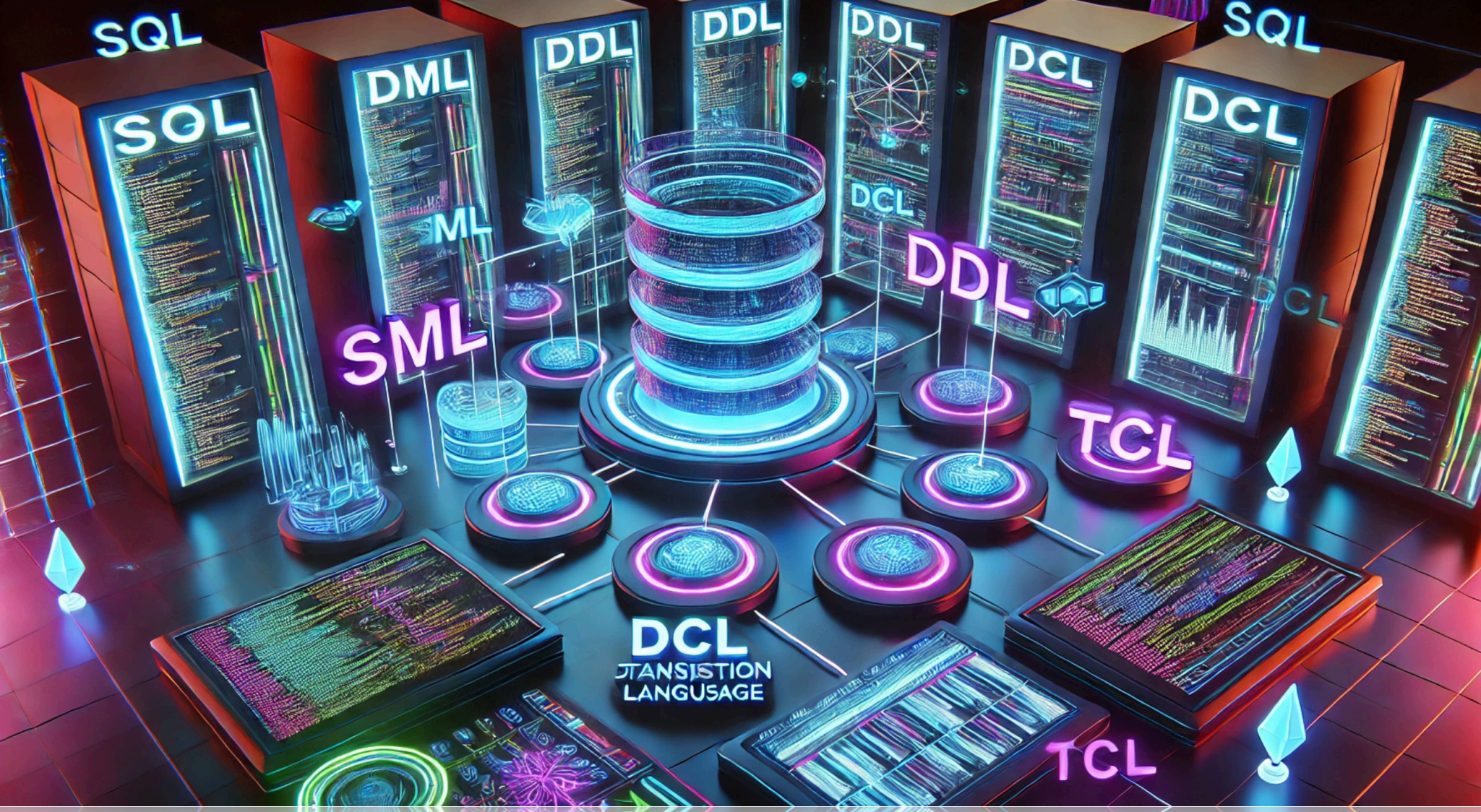
Ahmed Abdulwahid

The Ultimate Guide to SQL Commands

#DataGenius



SQL (Structured Query Language) is the powerhouse behind data management in relational databases. Whether you're managing business data, building dashboards, or performing complex queries, SQL has your back! 😎

Let's dive into the epic world of SQL commands, covering all the categories you need to master! 💡

📋 Categories of SQL Commands

SQL commands are grouped into five major categories:

- *DML (Data Manipulation Language)* ✍️
- *DDL (Data Definition Language)* 🏗️
- *DCL (Data Control Language)* 🔐
- *TCL (Transaction Control Language)* 🔄
- *DQL (Data Query Language)* 🔍

Data Manipulation Language (DML) ✍️

DML commands help you manipulate data stored in database tables. Think of DML as the "action" commands

1. SELECT 🔎 — Retrieves data from tables.

- *Syntax:* `SELECT column1, column2 FROM table_name WHERE condition;`

- *Example:* `SELECT * FROM employees WHERE department = 'Sales';`

- *Advanced Usage: Use* `JOIN` *to combine data from multiple tables:*

```
SELECT e.name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.id;
```

2. INSERT + — Adds new records.

- *Syntax:* `INSERT INTO table_name (column1, column2) VALUES (value1, value2);`

- *Example:* `INSERT INTO employees (name, role, salary) VALUES ('John Doe', 'Manager', 75000);`

- *Bulk Insert: Insert multiple records at once:*

```
INSERT INTO employees (name, role)
VALUES ('Jane Doe', 'Analyst'), ('Mark Smith', 'Developer');
```

3. UPDATE 🔄 — Modifies existing data.

- *Syntax:* `UPDATE table_name SET column1 = value1 WHERE condition;`

- *Example:* `UPDATE employees SET salary = salary * 1.1 WHERE department = 'Sales';`

- *Multi-Table Update (MySQL):*

```sql
UPDATE employees e
JOIN departments d ON e.department_id = d.id
SET e.salary = e.salary * 1.05
WHERE d.department_name = 'IT';
```

4. DELETE ❌ — Removes data from tables.

- *Syntax:* `DELETE FROM table_name WHERE condition;`

- *Example:* `DELETE FROM employees WHERE name = 'John Doe';`

- *Trick: Use* `LIMIT` *to delete specific numbers of rows:*

```sql
DELETE FROM employees WHERE department = 'Sales' LIMIT 5;
```

Key Tip: Always use `WHERE` with `UPDATE` and `DELETE` to avoid affecting all records accidentally. ⚠️

Data Definition Language (DDL) 

DDL commands are used to define, modify, or remove database structures like tables, indexes, and schemas.

1. CREATE 🆕 — Creates new tables, views, indexes, etc.

- *Syntax:* `CREATE TABLE table_name (column1 datatype, column2 datatype, ...);`

- *Example:*

```sql
CREATE TABLE employees (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50) NOT NULL,
  role VARCHAR(50),
  hire_date DATE DEFAULT CURRENT_DATE
);
```

- *Advanced: Create views for complex queries:*

```sql
CREATE VIEW active_employees AS
SELECT * FROM employees WHERE status = 'active';
```

2. ALTER ✏️ — Modifies an existing table's structure.

- *Add Column:* `ALTER TABLE employees ADD salary DECIMAL(10,2);`

- *Modify Column:* `ALTER TABLE employees MODIFY role VARCHAR(100);`

- *Drop Column:* `ALTER TABLE employees DROP COLUMN hire_date;`

3. DROP 🗑️ — Deletes tables, views, or databases permanently.

- *Example:* `DROP TABLE employees;`

- *Cascade Delete (PostgreSQL):*

```
DROP TABLE employees CASCADE;
```

4. TRUNCATE 🚮 — Deletes all data from a table but keeps the structure intact.

- *Example:* `TRUNCATE TABLE employees;`
- *Key Difference from DELETE:* `TRUNCATE` *is faster and doesn't log individual row deletions.*
- *Key Tip:* `DROP` *and* `TRUNCATE` *are irreversible—double-check before executing!* 🔒

📊 Key Differences Between DROP, TRUNCATE, and DELETE

🔑 Feature	🗑️ DROP	🚄 TRUNCATE	✏️ DELETE
Removes Data	✅ Yes	✅ Yes	✅ Yes
Removes Table Structure	✅ Yes	❌ No	❌ No
Can Be Rolled Back	❌ No *(unless in transaction)*	❌ No *(mostly)*	✅ Yes *(with transaction)*
Supports WHERE Clause	❌ No	❌ No	✅ Yes
Speed	⚡ **Fastest**	🚀 **Faster**	🦕 **Slower**
Affects Indexes/Constraints	✅ Yes *(drops them)*	❌ No *(keeps them)*	❌ No *(keeps them)*
Typical Use Case	**Permanently** delete a table	**Quickly** clear all data	**Selectively** delete specific rows

Data Control Language (DCL) 

DCL commands manage access control and permissions in the database.

1. GRANT ✅ — Gives users access privileges.

- *Syntax:* `GRANT privilege ON object TO user;`

- *Example:* `GRANT SELECT, INSERT ON employees TO ‘user123'@'localhost';`

- *Grant with Option:*

```
GRANT SELECT ON employees TO 'user123' WITH GRANT OPTION;
```

2. REVOKE ❌ — Removes previously granted permissions.

- *Syntax:* `REVOKE privilege ON object FROM user;`

- *Example:* `REVOKE INSERT ON employees FROM 'user123'@'localhost';`

Key Tip: Use DCL to enforce security policies and protect sensitive data. 🛡️

Transaction Control Language (TCL) 🔄

TCL commands help manage database transactions to ensure data integrity.

1. COMMIT 💾 — Saves changes made during a transaction.

- *Example:* `COMMIT;`

- *Auto-Commit Mode: By default, some databases auto-commit after each statement.*

2. ROLLBACK ⬅ BACK — Reverses changes made during a transaction.

- *Example:* `ROLLBACK;`

- *Partial Rollback with SAVEPOINT:*

```
SAVEPOINT sp1;
UPDATE employees SET salary = 50000 WHERE id = 1;
ROLLBACK TO sp1;
```

3. SAVEPOINT 📍 — Sets a point within a transaction to roll back to.

- *Example:* `SAVEPOINT before_update;`

4. SET TRANSACTION ⚙️ — Configures transaction settings.

- `Example: SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;`

Key Tip: Always pair `COMMIT` and `ROLLBACK` wisely when working with sensitive transactions. 💡

Data Query Language (DQL) 🔍

While `SELECT` is technically part of DML, it's often considered its own category because of its importance in querying data.

- *`SELECT` 🌟 - The superstar of SQL, used for querying data.*
- *Basic Query: `SELECT name, role FROM employees;`*
- *With Conditions: `SELECT * FROM employees WHERE role = 'Manager';`*
- *Aggregations: `SELECT COUNT(*) FROM employees;`*
- *Advanced: Using subqueries:*

```
SELECT name FROM employees WHERE department_id =
                                (SELECT id FROM departments WHERE name = 'IT');
```

Key Tip: Mastering `SELECT` with `JOINs`, `GROUP BY`, and `HAVING` clauses unlocks powerful data analysis capabilities. 📊

🚀 Wrapping Up

SQL commands are like the building blocks of data management. Mastering DML, DDL, DCL, TCL, and DQL will turn you into a SQL wizard 🧙, *capable of handling any data challenge with ease.*

Repost it

Thank you