# SQL Commands Cheatsheet

Categorized by language: DDL, DML, DCL, TCL, DQL — definitions added. (v5)

This PDF lists common SQL commands grouped by category with a brief purpose and example syntax. Definitions for each language category are provided. Examples are ANSI-style; exact syntax may vary by RDBMS.

## Definitions — SQL Language Categories

| Category | Definition & Example |
|---|---|
| **DDL — Data Definition Language** | DDL statements define, modify, and remove database schema objects such as tables, views, indexes, and constraints. Common DDL commands: CREATE, ALTER, DROP, TRUNCATE, RENAME.<br>`Example: CREATE TABLE employees (id INT PRIMARY KEY, name VARCHAR(100));` |
| **DML — Data Manipulation Language** | DML statements are used to insert, update, delete, and query data stored in database tables. Common DML commands: INSERT, UPDATE, DELETE, MERGE/UPSERT.<br>`Example: INSERT INTO employees (id, name) VALUES (1, 'Alice');` |
| **DQL — Data Query Language** | DQL focuses on retrieving data (primarily the SELECT statement). It is often considered a subset of DML or an alternate name for querying.<br>`Example: SELECT id, name FROM employees WHERE salary > 50000;` |
| **DCL — Data Control Language** | DCL statements manage privileges, roles, and access control for database objects and users. Common commands: GRANT, REVOKE.<br>`Example: GRANT SELECT ON employees TO reporting_user;` |
| **TCL — Transaction Control Language** | TCL statements control transactions (groups of operations that must succeed or fail as a unit), allowing commit or rollback. Common commands: BEGIN/START TRANSACTION, COMMIT, ROLLBACK, SAVEPOINT.<br>`Example: BEGIN; UPDATE accounts SET balance = balance - 100 WHERE id = 1; COMMIT;` |

## DDL — Data Definition Language

| Command | Purpose | Syntax / Example |
|---|---|---|
| CREATE | Create a new table, view, index, or other object. | `CREATE TABLE employees (`<br>`id INT PRIMARY KEY,`<br>`name VARCHAR(100),`<br>`hire_date DATE`<br>`);` |
| ALTER | Modify an existing object (e.g., add column). | `ALTER TABLE employees ADD COLUMN salary DECIMAL(10,2);` |
| DROP | Remove an object (table, view, index). | `DROP TABLE employees;` |
| TRUNCATE | Remove all rows from a table (faster; may have different transactional behavior across DBs). | `TRUNCATE TABLE employees;` |

| Command | Purpose | Syntax / Example |
|---|---|---|
| RENAME | Rename an object (syntax varies by vendor). | `-- PostgreSQL: ALTER TABLE old_name RENAME TO new_name;` |

## DML — Data Manipulation Language

| Command | Purpose | Syntax / Example |
|---|---|---|
| SELECT | Retrieve rows from one or more tables. (Often called DQL - Data Query Language.) | `SELECT id, name FROM employees WHERE salary > 50000 ORDER BY hire_date DESC;` |
| INSERT | Insert new row(s) into a table. | `INSERT INTO employees (id, name, hire_date) VALUES (1, 'Alice', '2020-01-15');` |
| UPDATE | Modify existing row(s). | `UPDATE employees SET salary = salary * 1.05 WHERE hire_date < '2022-01-01';` |
| DELETE | Delete row(s) from a table. | `DELETE FROM employees WHERE id = 10;` |
| MERGE / UPSERT | Insert or update in a single statement (syntax varies by RDBMS). | `-- Example (pseudo): MERGE INTO target USING source ON (...) WHEN MATCHED THEN UPDATE WHEN NOT MATCHED THEN INSERT;` |

## DCL — Data Control Language

| Command | Purpose | Syntax / Example |
|---|---|---|
| GRANT | Give privileges to a user or role. | `GRANT SELECT, INSERT ON employees TO reporting_user;` |
| REVOKE | Remove privileges. | `REVOKE INSERT ON employees FROM some_user;` |

## TCL — Transaction Control Language

| Command | Purpose | Syntax / Example |
|---|---|---|
| BEGIN / START TRANSACTION | Start a transaction block. | `BEGIN; -- or START TRANSACTION;` |
| COMMIT | Commit the current transaction, making changes permanent. | `COMMIT;` |
| ROLLBACK | Rollback the transaction to undo changes. | `ROLLBACK;` |

| Command | Purpose | Syntax / Example |
|---|---|---|
| SAVEPOINT | Create a savepoint to partially roll back within a transaction. | `SAVEPOINT before_salary_update;`<br>`-- later: ROLLBACK TO SAVEPOINT before_salary_update;` |

## DQL — Data Query Language (commonly considered part of DML)

| Command | Purpose | Syntax / Example |
|---|---|---|
| SELECT (detailed) | Powerful query tool; supports JOINs, GROUP BY, HAVING, window functions. | `SELECT d.department, COUNT(*) AS cnt, AVG(salary) AS avg_sal`<br>`FROM employees e`<br>`JOIN departments d ON e.dept_id = d.id`<br>`GROUP BY d.department`<br>`HAVING AVG(salary) > 60000;` |
| UNION / UNION ALL | Combine results from multiple SELECT queries. UNION removes duplicates; UNION ALL keeps them. | `SELECT id, name FROM employees_us`<br>`UNION ALL`<br>`SELECT id, name FROM employees_eu;` |

## Other useful statements & clauses

| Command | Purpose | Syntax / Example |
|---|---|---|
| JOIN types | Combine rows from two or more tables. | `INNER JOIN, LEFT (OUTER) JOIN, RIGHT (OUTER) JOIN, FULL (OUTER) JOIN, CROSS JOIN` |
| WHERE | Filter rows. | `SELECT * FROM employees WHERE name LIKE 'A%';` |
| GROUP BY / HAVING | Aggregate rows and filter aggregates. | `GROUP BY dept_id HAVING COUNT(*) > 5;` |
| ORDER BY | Sort the result set. | `ORDER BY hire_date DESC;` |
| LIMIT / OFFSET | Limit rows returned (syntax varies). | `LIMIT 10 OFFSET 20; -- MySQL/Postgres`<br>`TOP 10 -- SQL Server` |

## Notes:

- Syntax can vary between RDBMS (MySQL, PostgreSQL, SQL Server, Oracle). Use vendor documentation for exact dialect features.
- Some statements (e.g., TRUNCATE, ALTER) may require extra privileges.
- For complex migrations and schema changes, consider transactional DDL support in your database.