PostgreSQL Cheat Sheet

A quick reference guide to PostgreSQL, covering essential commands, data types, functions, and syntax for database management and querying.

Basic Commands & Syntax

Connecting & Creating Databases

`psql -U username -d databasename -h hostname`	Connect to a PostgreSQL database.
`CREATE DATABASE database_name;`	Create a new database.
`\l`	List all databases.
`\c database_name`	Connect to a specific database.
`DROP DATABASE database_name;`	Delete a database.
`\dt`	List tables in the current database.

Basic SQL Commands

`SELECT column1, column2 FROM table_name;`	Retrieve data from a table.
`INSERT INTO table_name (column1, column2) VALUES (value1, value2);`	Insert new data into a table.
`UPDATE table_name SET column1 = value1 WHERE condition;`	Modify existing data in a table.
`DELETE FROM table_name WHERE condition;`	Remove data from a table.
`CREATE TABLE table_name (column1 datatype, column2 datatype);`	Create a new table.
`DROP TABLE table_name;`	Delete a table.

Data Types

Numeric: `INTEGER`, `BIGINT`, `REAL`, `DOUBLE PRECISION`, `NUMERIC`
Character: `VARCHAR(n)`, `TEXT`, `CHAR(n)`
Date/Time: `DATE`, `TIME`, `TIMESTAMP`, `TIMESTAMPTZ` (with timezone)
Boolean: `BOOLEAN`
Other: `UUID`, `JSON`, `JSONB`, `BYTEA`

Advanced Querying

Filtering and Sorting

`WHERE condition`	Filter rows based on a condition.
`ORDER BY column1 ASC|DESC`	Sort the result set by a column in ascending or descending order.
`LIMIT n`	Limit the number of rows returned.
`OFFSET n`	Skip the first n rows.
`LIKE pattern`	Pattern matching (`%` for any sequence, `_` for any single character).
`ILIKE pattern`	Case-insensitive pattern matching.

Joins

`INNER JOIN table2 ON table1.column = table2.column`	Returns rows only when there is a match in both tables.
`LEFT JOIN table2 ON table1.column = table2.column`	Returns all rows from table1 and matching rows from table2. If no match, `NULL` values are returned for table2 columns.
`RIGHT JOIN table2 ON table1.column = table2.column`	Returns all rows from table2 and matching rows from table1. If no match, `NULL` values are returned for table1 columns.
`FULL OUTER JOIN table2 ON table1.column = table2.column`	Returns all rows when there is a match in one of the tables.
`CROSS JOIN table2`	Returns the Cartesian product of the tables.

Aggregate Functions

`COUNT(column)` - Counts the number of rows.
`SUM(column)` - Sums the values in a column.
`AVG(column)` - Calculates the average of values in a column.
`MIN(column)` - Returns the minimum value in a column.
`MAX(column)` - Returns the maximum value in a column.
`GROUP BY column` - Groups rows with the same value in a column.
`HAVING condition` - Filters the results of a `GROUP BY` query.

Transactions & Concurrency

Transactions

`BEGIN;`	Starts a transaction block.
`COMMIT;`	Saves the changes made during the transaction.
`ROLLBACK;`	Discards the changes made during the transaction.
`SAVEPOINT savepoint_name;`	Sets a savepoint within a transaction.
`ROLLBACK TO savepoint_name;`	Rolls back to a specific savepoint.

Concurrency Control

Isolation Levels:
`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ`, `SERIALIZABLE`

`SET TRANSACTION ISOLATION LEVEL level;` - Sets the transaction isolation level.

`LOCK TABLE table_name IN lock_mode MODE;` - Explicitly locks a table.

Lock Modes:
`ACCESS SHARE`, `ROW SHARE`, `ROW EXCLUSIVE`, `SHARE UPDATE EXCLUSIVE`, `SHARE`, `SHARE ROW EXCLUSIVE`, `EXCLUSIVE`, `ACCESS EXCLUSIVE`

Functions & Operators

String Functions

`LENGTH(string)`	Returns the length of the string.
`UPPER(string)`	Converts the string to uppercase.
`LOWER(string)`	Converts the string to lowercase.
`TRIM(string)`	Removes leading and trailing spaces.
`SUBSTRING(string, start, length)`	Extracts a substring from the string.
`REPLACE(string, from, to)`	Replaces occurrences of `from` with `to`.

Date/Time Functions

`NOW()`	Returns the current timestamp.
`CURRENT_DATE`	Returns the current date.
`CURRENT_TIME`	Returns the current time.
`EXTRACT(field FROM timestamp)`	Extracts a field (e.g., year, month, day) from a timestamp.
`DATE_TRUNC(field, timestamp)`	Truncates a timestamp to a specified field.

Operators

Comparison: `=`, `<>`, `<`, `>`, `<=`, `>=`

Logical: `AND`, `OR`, `NOT`

String Concatenation: `||`

Mathematical: `+`, `-`, `*`, `/`, `^` (exponentiation), `|/` (square root)

Pattern Matching: `LIKE`, `ILIKE`, `SIMILAR TO`