



Basic Commands & Syntax

Connecting & Creating Databases

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

Basic SQL Commands

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

Data Types

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

Advanced Querying

Filtering and Sorting

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

Joins

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

Aggregate Functions

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

Transactions & Concurrency

Transactions

| | |
|--|---|
| <code>BEGIN;</code> | Starts a transaction block. |
| <code>COMMIT;</code> | Saves the changes made during the transaction. |
| <code>ROLLBACK;</code> | Discards the changes made during the transaction. |
| <code>SAVEPOINT savepoint_name;</code> | Sets a savepoint within a transaction. |
| <code>ROLLBACK TO savepoint_name;</code> | Rolls back to a specific savepoint. |

Concurrency Control

| |
|--|
| Isolation Levels: |
| <code>READ UNCOMMITTED</code> , <code>READ COMMITTED</code> , <code>REPEATABLE READ</code> , <code>SERIALIZABLE</code> |
| <code>SET TRANSACTION ISOLATION LEVEL level1;</code> - Sets the transaction isolation level. |
| <code>LOCK TABLE table_name IN lock_mode MODE;</code> - Explicitly locks a table. |
| Lock Modes: |
| <code>ACCESS SHARE</code> , <code>ROW SHARE</code> , <code>ROW EXCLUSIVE</code> , <code>SHARE UPDATE EXCLUSIVE</code> , <code>SHARE</code> , <code>SHARE ROW EXCLUSIVE</code> , <code>EXCLUSIVE</code> , <code>ACCESS EXCLUSIVE</code> |

Functions & Operators

String Functions

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

Date/Time Functions

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

Operators

| |
|---|
| Comparison: <code>=</code> , <code><></code> , <code><</code> , <code>></code> , <code><=</code> , <code>>=</code> |
| Logical: <code>AND</code> , <code>OR</code> , <code>NOT</code> |
| String Concatenation: <code> </code> |
| Mathematical: <code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>^</code> (exponentiation), <code> /</code> (square root) |
| Pattern Matching: <code>LIKE</code> , <code>ILIKE</code> , <code>SIMILAR TO</code> |