

```
-- TIME => provides the time
-- TIMESTAMP => combination of the date and time
-- ARRAY => ordered collection of elements of the same datatype
-- JSON => internal collection equivalent to the JSON datatype, allowing for manipulation of flexible data structures
-- POINT, LINE, POLYGON => used to represent geometric objects
```

Quickstart

```
-- ----- QUICKSTART -----
-- PostgreSQL uses SQL as its query language, below is a quick rundown of general SQL syntax
-- SQL is a semicolon language

-- ---- GENERAL ----
-- CREATE DATABASE <databaseName> => creates a database with the specified name
-- USE <database> => specifies the named database for use presently in the SQL queries
-- CREATE TABLE <tableName> (<columnName> <datatype> <augmenter(s)>) => creates a table under the specified name and assigns it the specified values with as many columns as the table has
-- INSERT INTO <tableName> <columnTypeSignature> VALUES <valuesToBeInserted> => inserts a record (row) of the specified values to the given table, SQL is strict about the type signature of the table and values must be appropriate as to the type signature of the columns, otherwise an error is thrown
-- UPDATE <tableName> SET <changedColumnValue> WHERE <predicate(s)> => updates the specified record which matches the predicate check with the new value in the given table
-- DELETE FROM <tableName> WHERE <predicate(s)> => deletes records from the specified table when the record matches the predicate check
-- SELECT <sqlQuery> FROM <tableName(s)> WHERE <predicates(s)> => the foundation of any sql query, used to display the queried tables, records or results after augmenting the query with the specified predicates

CREATE DATABASE mydatabase;
USE mydatabase;

CREATE TABLE users (
  id SERIAL PRIMARY KEY, -- Auto-incrementing integer for unique ID
  username VARCHAR(50) NOT NULL UNIQUE, -- Username, max 50 characters, unique
  email VARCHAR(100) NOT NULL UNIQUE -- Email address, max 100 characters, unique
);

INSERT INTO users (name, age, email) VALUES ('Alice', 30, 'alice@example.com');
INSERT INTO users (name, age, email) VALUES ('Bob', 25, 'bob@example.com');

UPDATE users SET age = 31 WHERE name = 'Alice';

DELETE FROM users WHERE name = 'Bob';

SELECT * FROM users;
SELECT name, age FROM users WHERE age > 25;
```

More on

- [sql aggregate functions](#)
- [sql joins](#)
- [sql subqueries](#)