[Team Cool Rabbit]
Team LinkedIn Account

[Alaa Magdy] [Link]

[Ahmed Hisham] [Link]

[Esraa Dahy] [www.linkedin.com/in/esraa-dahy-801349245]

[Hassan Ezz] [Link]

Difference between SQL and NoSQL

SQL	NoSQL
Relational-Database	Non-relational or Distributed Database
Have fixed or predefined schema	Have dynamic schema
Not suited for hierarchical data storage	Best suited for hierarchical data storage
Are best suited for complex queries	Are not so good for complex queries
Vertically Scalable	Horizontally scalable
Follows ACID property	Follows CAP(consistency, Availability, partition tolerance)
Are table-based	Are document, key-value, graph, or wide-column stores.
Examples: MySQL, PostgreSQL, Oracle, MS-SQL Server etc	**Examples:** MongoDB, GraphQL, HBase Neo4j , Cassandra

What is PostgreSQL?

PostgreSQL is an advanced, enterprise class open source relational database that supports both SQL (relational) and JSON (non-relational) querying. It is a highly stable database management system, backed by more than 20 years of community development which has contributed to its high levels of resilience, integrity, and correctness. PostgreSQL is used as the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications.

Why We Use PostgreSQL?

Data types: PostgreSQL supports all needed data types such as documents, primitives, geometry, structures etc

Data integrity: Postgres provides your data integrity by introducing constraints and regulating data you add. With PostgreSQL, you can forget about invalid or orphan records.

Performance: Parallelization of read queries, powerful indexing methods, Multiversion concurrency control. These are only a few of numerous features implemented by PostgreSQL to boost and optimize its performance.

Disaster Recovery & Reliability: PostgreSQL cares to provide the highest level of reliability for your data. With its sophisticated replication options, your data are totally safe. Besides, you can always backup the most valuable information.

Extensibility: In this database, you don't have to limit yourself to certain types of documents. The database offers you a wide selection of data types for your disposal.

Internationalization & Text Search: Postgres supports international character sets. It also enables full-text search to speed up finding process and integrates case-insensitive and accent-insensitive collations.

Support of non-relational data: It is probably the most important update of the database. Support of JSON, XML, Hstore and Cstore documents actually turns Postgres into NoSQL database

---

What is CRUD?

CRUD is an acronym that refers to four necessary basic functions (CREATE, READ, UPDATE, DELETE) of a SQL database.

Creating a table called users in a database.

```
                            ^
postgres=# CREATE TABLE users (id SERIAL PRIMARY KEY NOT NULL, name VARCHAR(100) NOT NULL, age INT NOT NULL);
CREATE TABLE
postgres=#
```

.CREATE user in the table

```
postgres=# INSERT INTO users (name, age) VALUES ('Ahmed Hisham', 22);
INSERT 0 1
postgres=#
```

.UPDATE user with id=1 to be with age=30, and then READ values in users table

```
postgres=# UPDATE users SET age=30 WHERE id=1;
UPDATE 1
postgres=# SELECT * FROM users;
 id |      name      | age
----+----------------+-----
  2 | Yousef         |  20
  3 | Omar Ahmed     |  20
  1 | ahmed hisham   |  30
(3 rows)
```

-DELETE user with id=2

```
postgres=# DELETE from users where id=2;
DELETE 1
postgres=# SELECT * FROM users;
 id |      name      | age
----+----------------+-----
  3 | Omar Ahmed     |  20
  1 | ahmed hisham   |  30
(2 rows)
```

SQL filters

SQL filters are text strings that you use to specify a subset of the data items in an internal or SQL database data type.

For SQL database and internal data types, the filter is an SQL `WHERE` clause that provides a set of comparisons that must be true in order for a data item to be returned. These comparisons are typically between field names and their corresponding values.

Practical

WHERE

```
postgres=# UPDATE students SET sighting_date='1/9/2022' WHERE id=1;
UPDATE 1
```

LIMIT

```
postgres=# SELECT * FROM students LIMIT 4;
 id |     name     | sighting_date
----+--------------+---------------
  2 | ahmed hesham |
  3 | esraa        |
  1 | alaa magdy   | 2022-01-09
  4 | hassan       |
(4 rows)
```

LIKE

```
postgres=# SELECT * FROM students WHERE name LIKE 'a%';
 id |     name     | sighting_date
----+--------------+---------------
  1 | alaa magdy   | 2022-01-09
  2 | ahmed hesham | 2022-10-10
(2 rows)
```

NULL

```
postgres=# SELECT * FROM students WHERE sighting_date ISNULL;
 id |  name  | sighting_date
----+--------+---------------
  4 | hassan |
(1 row)
```

NOTNULL

```
postgres=# SELECT * FROM students WHERE sighting_date IS NOT NULL;
 id |     name     | sighting_date
----+--------------+---------------
  1 | alaa magdy   | 2022-01-09
  2 | ahmed hesham | 2022-10-10
  3 | esraa        | 2022-10-09
(3 rows)
```

Difference Between Primary Key and Foreign key

PRIMARY KEY	FOREIGN KEY
A primary key is used to ensure data in the specific column is unique.	A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables.
It uniquely identifies a record in the relational database table.	It refers to the field in a table which is the primary key of another table .
Only one primary key is allowed in a table.	Whereas more than one foreign key are allowed in a table.
It is a combination of UNIQUE and Not Null constraints.	It can contain duplicate values and a table in a relational database
It does not allow NULL values.	It can also contain NULL values.
Its value cannot be deleted from the parent table.	Its value can be deleted from the child table.

Database migrations

What is a db migration?

Database migration is **the process of migrating data from one or more source databases to one or more target databases by using a database migration service**. When a migration is finished, the dataset in the source databases resides fully, though possibly restructured, in the target databases.

To make this work we will use the "***db-migrate***" package, it is a database migration framework for node js.

1. First of all we install the package by running *"**npm i db-migrate -g**"*,
2. We create the configuration file "database.json" and configure the development database and the testing database.
3. We create our first table by running "***db-migrate create users***", then a migrations folder will appear, and you will see two files for every table one for creating the table and one for dropping the table or "up" and "down"
4. We add our SQL code that creates the table and its columns in "***migrations/sql***" folder, in the user up SQIL file
5. We add our SQL code that drops the table in "***migrations/sql***" folder, in the user down SQIL file

Now we finished up configuring the migrations process, we now can run "***db-migrate up***" to create the mentioned tables, then we can run "***db-migrate down***" to drop a table from the database.

Note: if you want to use the testing database add "***-e test***" to the up or down command, example: "***db-migrate up -e test***"

References:
https://www.geeksforgeeks.org/difference-between-sql-and-nosql/
https://fulcrum.rocks/blog/why-use-postgresql-database