



Databases

Databases

Efficient

Can store lots of data

Tables, Columns, Rows

2 types: Relational DB using SQL, Document using an adapter



Relational DB

Relationships define structure

User ID 5 posts 56 tweets

56 rows in the TWEETS table are related to User 5



Relational DB

ID	Content	User_ID
1	My first tweet	5
2	Great coffee @ozone	5
3	Check out @CodersInHoods	5

Relationships

=> one-to-one

=> one-to-many

=> many-to-many

One-to-One

This type of relationship allows only one record on each side of the relationship. The primary key relates to only one record—or none—in another table.



Example

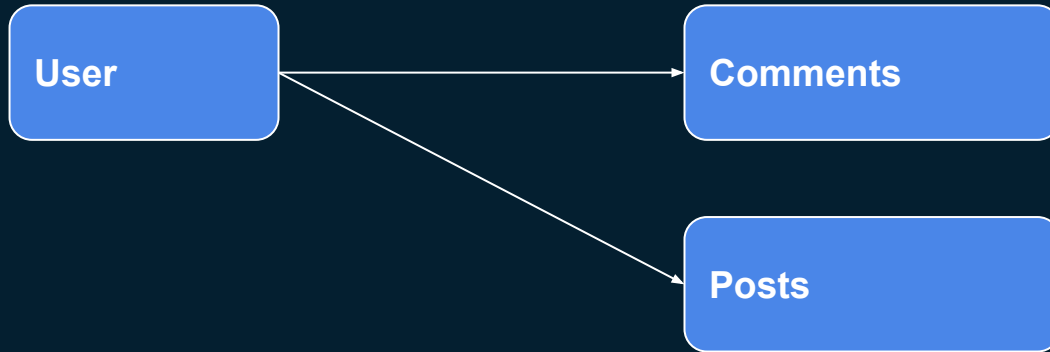
ID	full_name
1	Vasile Cojusco
2	John Smith
3	Tom Smith

ID	number	size	student_id
1	322	25	1
2	114	20	3
3	232	24	2



One-to-many

A one-to-many relationship allows a single record in one table to be related to multiple records in another table.



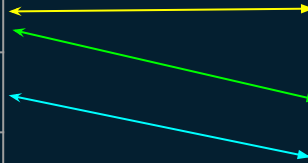
Example

Users

ID	name
1	mrPotato
2	josh777
3	matSmith

Comments

ID	text	user_id
1	Great article	1
2	Want to read more	1
3	Like!	2



Example

Users

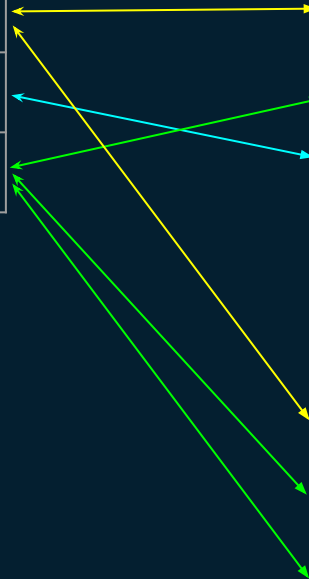
ID	name
1	mrPotato
2	josh777
3	matSmith

Comments

ID	text	user_id
1	Great article	1
2	Want to read more	3
3	Like!	2

Posts

ID	title	user_id
1	Best car	1
2	What is MySQL	3
3	JS for experts	3



Many-to-many

In this relationship many records in a table can link to many records in another table.



Example

authors

ID	full_name
1	Josh Smith
2	Andrew Butkevich
3	Albert Bow

author_book

ID	author_id	book_id
1	1	2
2	1	1
3	2	2

books

ID	title
1	JS: Book 1
2	JS: Book 2
3	Frontend in 2020
4	NodeJS: A-Z

Primary key

A primary key is a special relational database table column (or combination of columns) designated to uniquely identify each table record.

*Primary keys must contain **UNIQUE** values, and cannot contain **NULL** values.*

NOTE: you need to define a Primary key when you create a table



SQL

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

SQL Cheat sheet

<https://hackr.io/blog/sql-cheat-sheet>



MySQL setup

1. Install [MySQL](#).
2. Install [mysql package](#) from npm.
3. Install [TablePlus](#) - GUI for DB.



To code editor



Sequelize

Sequelize. Sequelize is a promise-based Node.js ORM for Postgres, MySQL, MariaDB, SQLite and Microsoft SQL Server.*

****ORM** - Object-relational mapping in computer science is a programming technique for converting data between incompatible type systems using object-oriented programming languages.*



Basic setup

Postgres or sqlite with URI

```
1 const { Sequelize } = require('sequelize');  
2  
3 // Option 1: Passing a connection URI  
4 const sequelize = new Sequelize('sqlite::memory:') // Example for sqlite  
5 const sequelize = new Sequelize('postgres://user:pass@example.com:5432/dbname') //  
   Example for postgres
```

Basic setup

sqlite with params



```
1 const { Sequelize } = require('sequelize');  
2  
3 // Option 2: Passing parameters separately (sqlite)  
4 const sequelize = new Sequelize({  
5   dialect: 'sqlite',  
6   storage: 'path/to/database.sqlite'  
7 });
```

Basic setup

Other with params



```
1 const { Sequelize } = require('sequelize');  
2  
3 // Option 2: Passing parameters separately (other dialects)  
4 const sequelize = new Sequelize('database', 'username', 'password', {  
5   host: 'localhost',  
6   dialect: /* one of 'mysql' | 'mariadb' | 'postgres' | 'mssql' */  
7 });
```

To code editor



Practice

Create API using express and sequelize.

End points:

/users

`GET` - return all users

`POST` - add a new user

/pets

`GET` - return all pets

/users/:id

`GET` - return user by id

`DELETE` - delete user by id

