

Videregående Database



Murat Kilic
mki@eaaa.dk



Rasmus Cederdorff
race@eaaa.dk

Agenda

1. Introduktion til kurset og indhold
2. Hvem er vi?
3. Hvem er I, og hvilke forudsætninger kommer I med?
4. Værktøjer
5. SQL vs NoSQL
6. Grundlæggende SQL

Videregående Database

Modulets formål er at give den studerende kendskab

- til forskellige databasemodeller,
- til databaseserveres virkemåde
- og til forskellige anvendelser af Database.

Videregående Database

Indhold:

- Databasemodeller
- Relationel Model
- Mapping og Mapping Frameworks
- Videregående SQL
- Transaktionshåndtering
- Fysisk Niveau: indeks, query-optimering
- Avancerede Databaseanvendelser

Videregående Database

- Databasemodeller
- Relationel Model
- Mapping og Mapping Frameworks
- Videregående SQL
- Transaktionshåndtering
- Fysisk Niveau: indeks, query-optimering
- Avancerede Databaseanvendelser

► RACE & MKI - Introduktion og opsamling på grundlæggende SQL - 23-10-2023

▼ MKI - Relationel Model og Videregående SQL - 30-10-2023

▼ MKI - Transaktionshåndtering, indeksering og query-optimering - 06-11-2023

▼ RACE - Mapping og Mapping Frameworks - 13-11-2023

▼ RACE - Introduktion til NoSQL og MongoDB - 20-11-2023

▼ RACE - MongoDB, Mongoose og Mapping - 27-11-2023

▼ RACE - NoSQL vs SQL, Databaseanvendelser og opsamling - 04-12-2023

Introduktion og opsamling på grundlæggende SQL

Formål

- Introducere jer til kurset og dets indhold
- Hvem er I, og hvem er underviserne
- Opsætte nødvendige værktøjer
- Indledende forståelse for de to primære database modeller: Relationelle (SQL) og Ikke-relationelle (NoSQL)
- Opsamling på grundlæggende SQL

Forberedelse

- **Læs:**
 - MongoDB, Inc. 2023. Relational vs. Non-Relational Databases: <https://www.mongodb.com/compare/relational-vs-non-relational-databases>
- **Se:**
 - MySQL vs MongoDB:
<https://www.youtube.com/watch?v=OdgZ0jr4jpM>
- **Gør:**
 - Sørg for at du har MySQL installeret på din maskine. Du kan eventuelt følge denne guide: [Installer MySQL](#)
 - Installer Database IDE'et DataGrip. Du kan anvende følgende guide: [Installer og opsæt DataGrip](#)

Materialer

- **Slides:**
 - Vil blive tilgængelige her
- **Opgaver og guides:**
 - [Installer MySQL](#)
 - [Installer og opsæt DataGrip](#)
- **Links:**
 - Hvem er I? Padlet: https://eaaa.padlet.org/race/hvem_er_i_au_vdb_e23

Agenda/ Emner

- Introduktion til kurset og dets indhold
- Præsentation af undervisere
- Hvem er I, og hvilke forudsætninger kommer I med?
- Opsætning af værktøjer

Videregående Database

Eksamens:

- Mundtlig prøve kombineret med et praktisk produkt og en disposition.

A.7 Mundtlig prøve kombineret med et praktisk produkt og en disposition

Et praktisk produkt er kendtegnet ved:

- at den studerende gennem det praktiske produkt viser sine evner til at omsætte teori til et praktisk produkt
- at den studerende præsenterer sin løsning, som fx kan være en opførelse, en udstilling, en praktiskøvelse, resultatet af en programmeringsopgave, en videopræsentation eller lignende
- at det sammen med dispositionen danner grundlag for den mundtlige eksamination.

Dispositionens indhold og omfang

En disposition er kendtegnet ved:

- at den har et omfang på maksimum én normalside og desuden er vedhæftet en litteraturliste
- at den i stikordsform angiver det teoretiske grundlag for det praktiske produkt
- at den sammen med det praktiske produkt danner grundlag for den mundtlige eksamination.

Bedømmelse

Bedømmelsesgrundlaget er en samlet helhedsvurdering af det praktiske produkt, dispositionen og den mundtlige præstation. Der gives en samlet karakter efter 7-trinsskalaen.

Mundtlig prøve

Den mundtlige prøve har en varighed på i alt 30 minutter, der har følgende om trentlige fordeling:

- 1/4 til et mundtligt oplæg
- 2/4 til eksaminationssamtale
- 1/4 til votering og tilbagemelding til den studerende.

I'm Rasmus Cederdorff (RACE)

Senior Lecturer

Freelance Web App Developer

- Programming with an eye for UI and UX.
- Web Development, JavaScript, React, BaaS & Node.js
- “I speak JavaScript”.
- Websites, Webshops, Web Apps, Mobile Apps, Server App and BaaS.







I'm Rasmus Cederdorff
Aarhus (Tilst)
Alicia & Ida

From Holstebro
I'm into sports
Love (apple) gadgets, to take
pictures & interior design
projects

What's with my arm?

Frontend (client)

User CRUD with Node.js REST API

User ID	Name	Title	Email
1	Birgitte Kirk Iversen	Senior Lecturer	bki@mail.dk
4	Rasmus Cederdorff	Senior Lecturer	race@mail.dk
5	Dan Okkels Brendstrup	Lecturer	dob@mail.dk
6	Kasper Fischer Topp	Lecturer	kato@mail.dk

Backend (Server)

```
Raw Parsed
```

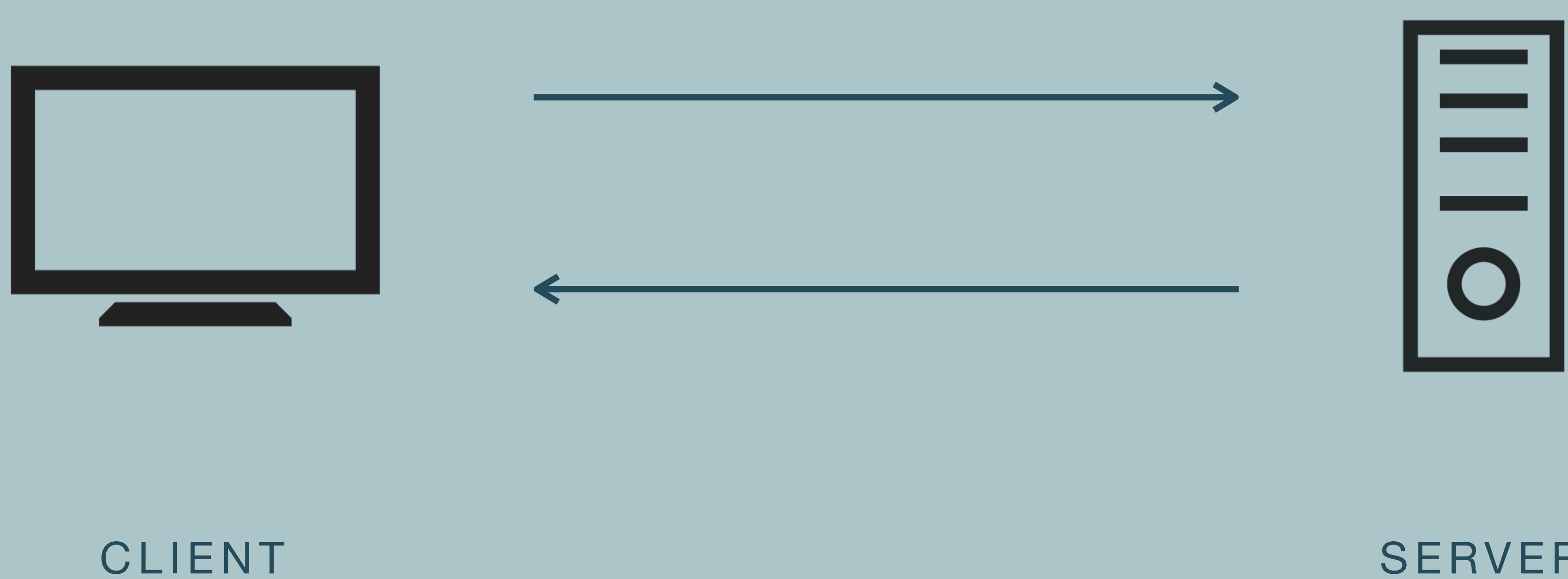
```
[{"id": 1, "name": "Birgitte Kirk Iversen", "mail": "bki@mail.dk", "title": "Senior Lecturer", "image": "https://www.baaa.dk/media/u4gorzsd/birgitte-kirk-iversen2.jpg?anchor=center&mode=crop&width=800&height=450&rnd=132792921541630000&format=webp"}, {"id": 4, "name": "Rasmus Cederdorff", "mail": "race@mail.dk", "title": "Senior Lecturer", "image": "https://www.baaa.dk/media/devlvgj/rasmus-cederdorff.jpg?anchor=center&mode=crop&width=800&height=450&rnd=132792921695570000&format=webp"}, {"id": 5, "name": "Dan Okkels Brendstrup", "mail": "dob@mail.dk", "title": "Lecturer", "image": "https://www.eaaa.dk/media/bdoje141/dan-okkels-brendstrup.jpg?anchor=center&mode=crop&width=800&height=450&rnd=132792921559630000&format=webp"}, {"id": 6, "name": "Kasper Fischer Topp", "mail": "kato@mail.dk", "title": "Lecturer", "image": "https://www.eaaa.dk/media/lxzcybme/kasper-topp.jpg?anchor=center&mode=crop&width=800&height=450&rnd=132792921618200000&format=webp"}, {"id": 7, "name": "Line Skjødt", "mail": "lskj@mail.dk", "title": "Senior Lecturer & Internship Coordinator", "image": "https://www.eaaa.dk/media/14qpfeq4/line-skj%C3%B8dt.jpg?anchor=center&mode=crop&width=800&height=450&rnd=132792921638700000&format=webp"}, {"id": 8, "name": "Martin Aagaard Nøhr", "mail": "mnor@mail.dk", "title": "Lecturer", "image": "https://www.eaaa.dk/media/oayjq02h/martin-n%C3%B8hr.jpg?anchor=center&mode=crop&width=800&height=450&rnd=132792921658800000&format=webp"}]
```

<https://github.com/cederdorff/rest-user-crud-frontend>

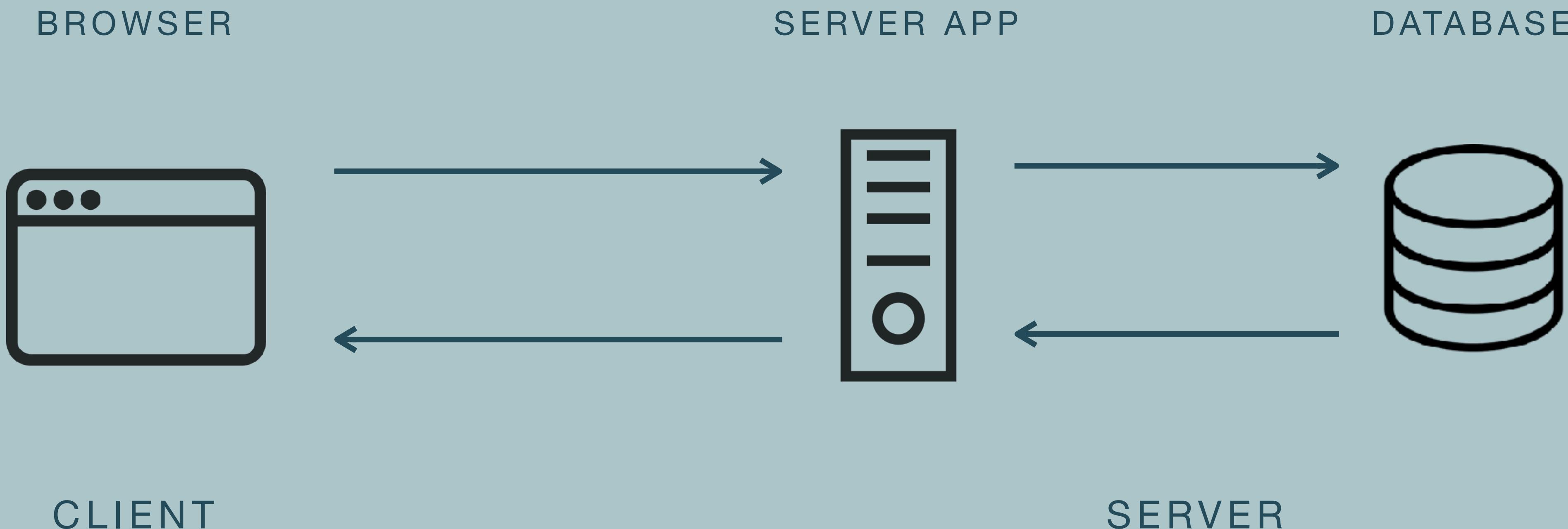
<https://github.com/cederdorff/rest-user-crud-backend>

Client-Server Model

Communication between web **clients** and web **servers**.



Web Dev Architecture



Hvem er I?

https://eaaa.padlet.org/race/hvem_er_i_au_vdb_e23

- Navn
- By
- Baggrund - Job og uddannelse
- Hvorfor AU i IT og Videregående Databaser?
- Tekniske forudsætninger og erfaring
 - Programmeringssprog? Databasesystemer?
 - Anvender du programmering og/eller databaser i forbindelse med dit job?
- Forventninger og ønsker til kurset?
- Drømmejob og fremtid
- Interesser og/eller facts du har lyst til at dele?
- Links - del gerne LinkedIn, website og/eller lignende
- Profilbillede - meget gerne for hukommelsens skyld

- Du bestemmer selv, hvad du vil dele.
- Det er blot så vi kan lære jer bedre at kende.
- Og tilpasse kursets indhold bedst muligt til jer.

Tools



SQL

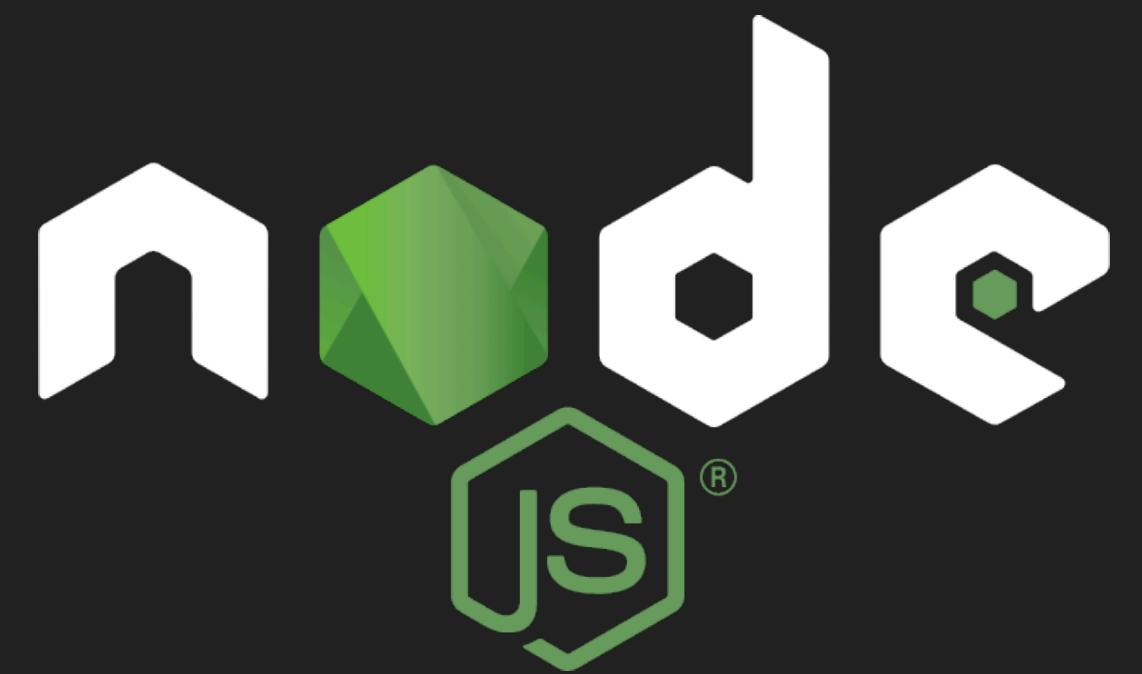


NoSQL

Tools



Tools



Express



Hvad skal I installere?



Installer MySQL



Installer og opset DataGrip



Vi anvender Atlas (online)

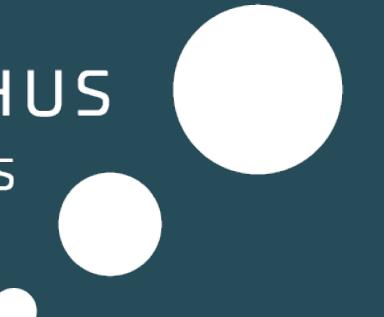


Installer Node og NPM

Databaser & SQL

Videregående Databaser

ERHVERVSAKADEMI AARHUS
BUSINESS ACADEMY AARHUS



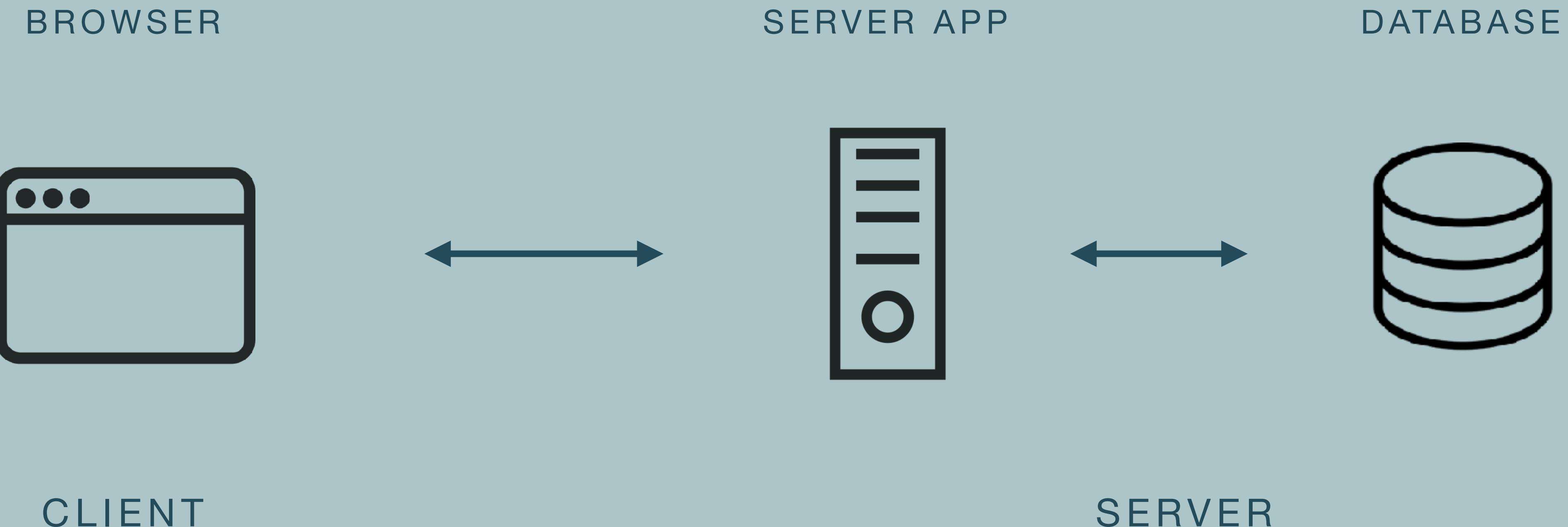
Content

- What is a Data Source?
- What is a Database?
- Relational Databases
- SQL & MySQL
 - SQL
 - MySQL
- Tables & Entities
- SQL statements, DDL, DML & DCL
 - CREATE database & table
 - Data Types
 - INSERT INTO (create)
 - SELECT (read)
 - WHERE
 - ORDER BY
 - UPDATE (update)
 - DELETE (delete)
- SQL Queries
 - GROUP BY
 - HAVING
- Keys & IDs
- Entity Relationships
 - One to many
 - Many to many
 - JOINS

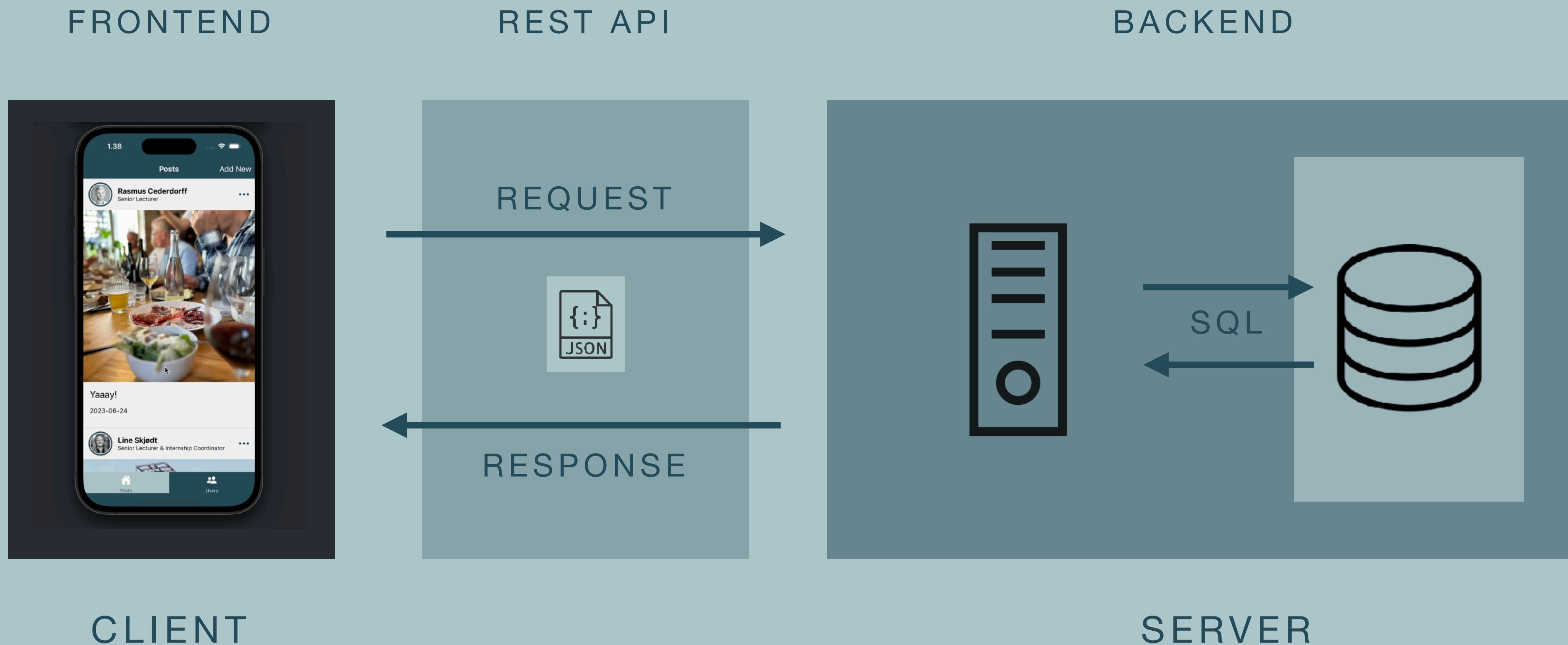
Grundlæggende SQL

- SQL statements, DDL, DML & DCL
 - CREATE database, table & Data Types
 - INSERT INTO (create)
 - SELECT (read)
 - WHERE
 - ORDER BY
 - UPDATE (update)
 - DELETE (delete)
- SQL Queries
 - GROUP BY
 - HAVING
 - Keys & IDs
 - Entity Relationships
 - One to many
 - Many to many
 - JOINS

Web Dev Architecture



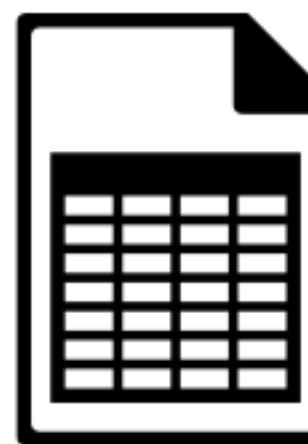
Web Dev Architecture



What is a Data Source?

What is a Data Source?

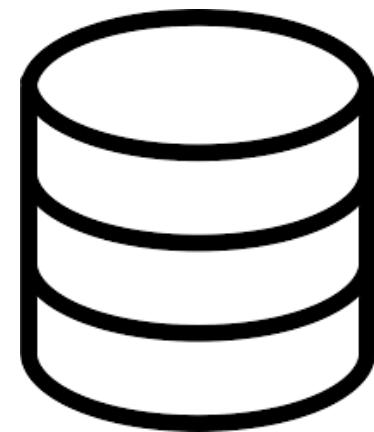
- Data sources are places or systems where data is collected or accessed.



{JSON}

What is a Data Source?

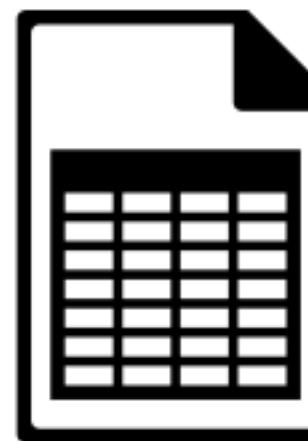
- Location of data
- Where data comes from
- Can be any kind of data of any file format
- Database, a file, data sheet, spreadsheet, XML, JSON



{JSON}

What is a Data Source?

- A data source is where we get our data. It can be any place or system we use to collect, find, or access information.



{JSON}

Types

- Data sources can be diverse and can include:

- Databases
- Spreadsheets
- APIs
- Files
- Web Scraping
- Sensors/IoT Devices
- Cloud Services
- Legacy Systems
- External Partners
- Logs/Event Streams
- Social Media
- Public Datasets



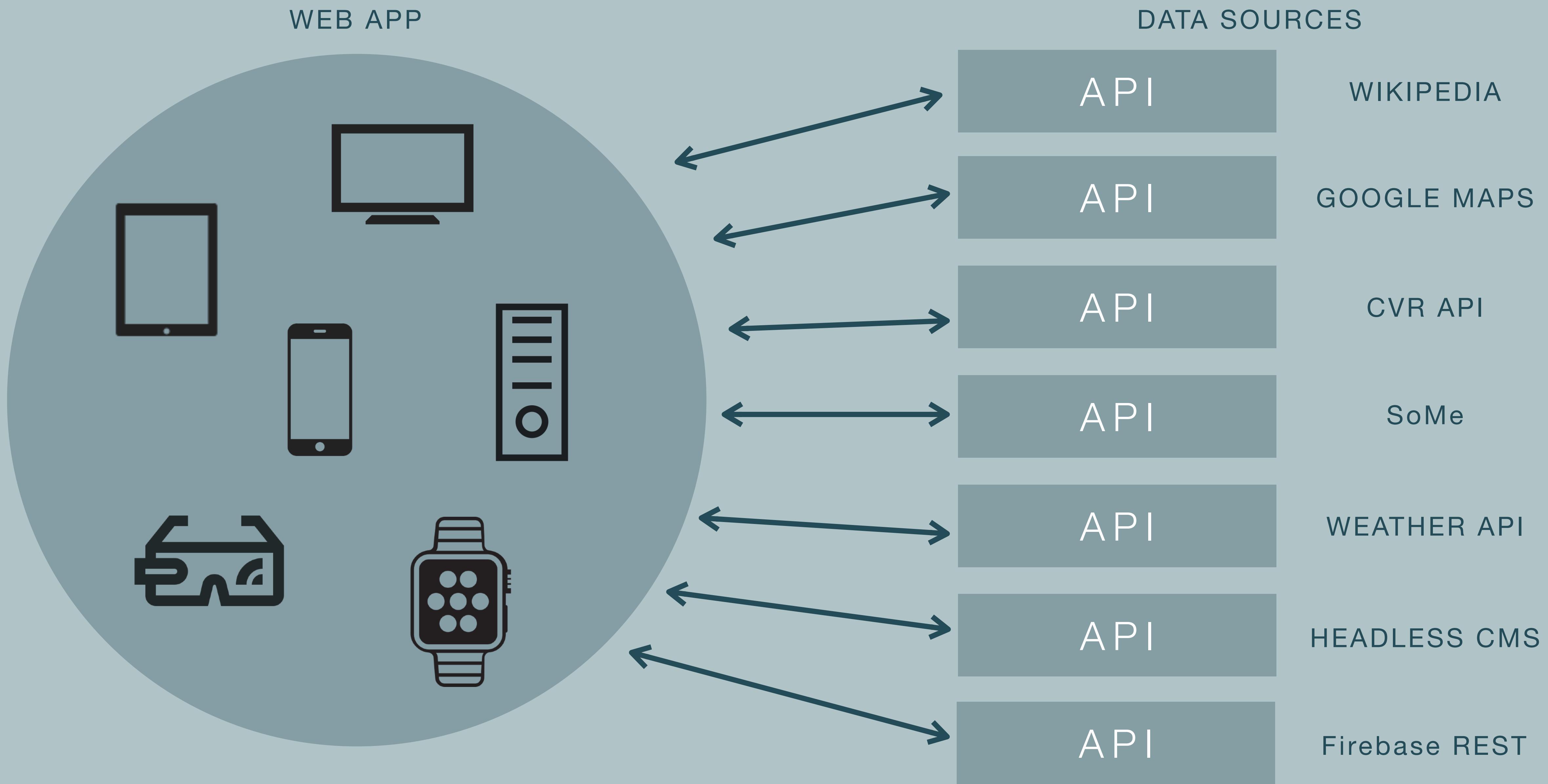
{JSON}

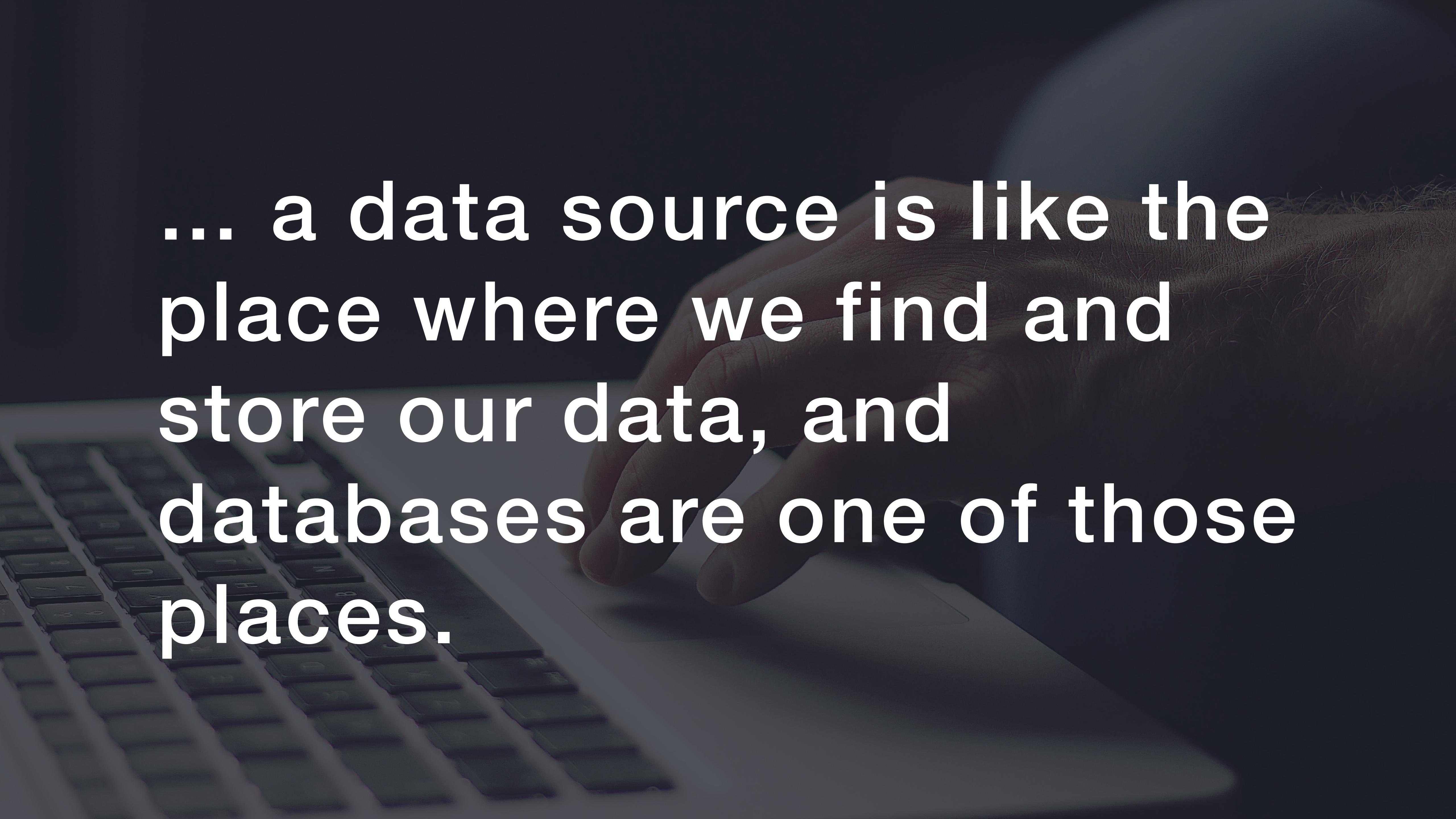
Data Sources

- Purpose: Data sources are used for analysis, reporting, and decision-making.
- Integration: Often, data from multiple sources is integrated into a central repository (data warehouse or data lake) for analysis.
- Data Variety: Data sources can differ in format, structure, and accessibility.
- Data Integration: The process of combining and preparing data from diverse sources is known as data integration.

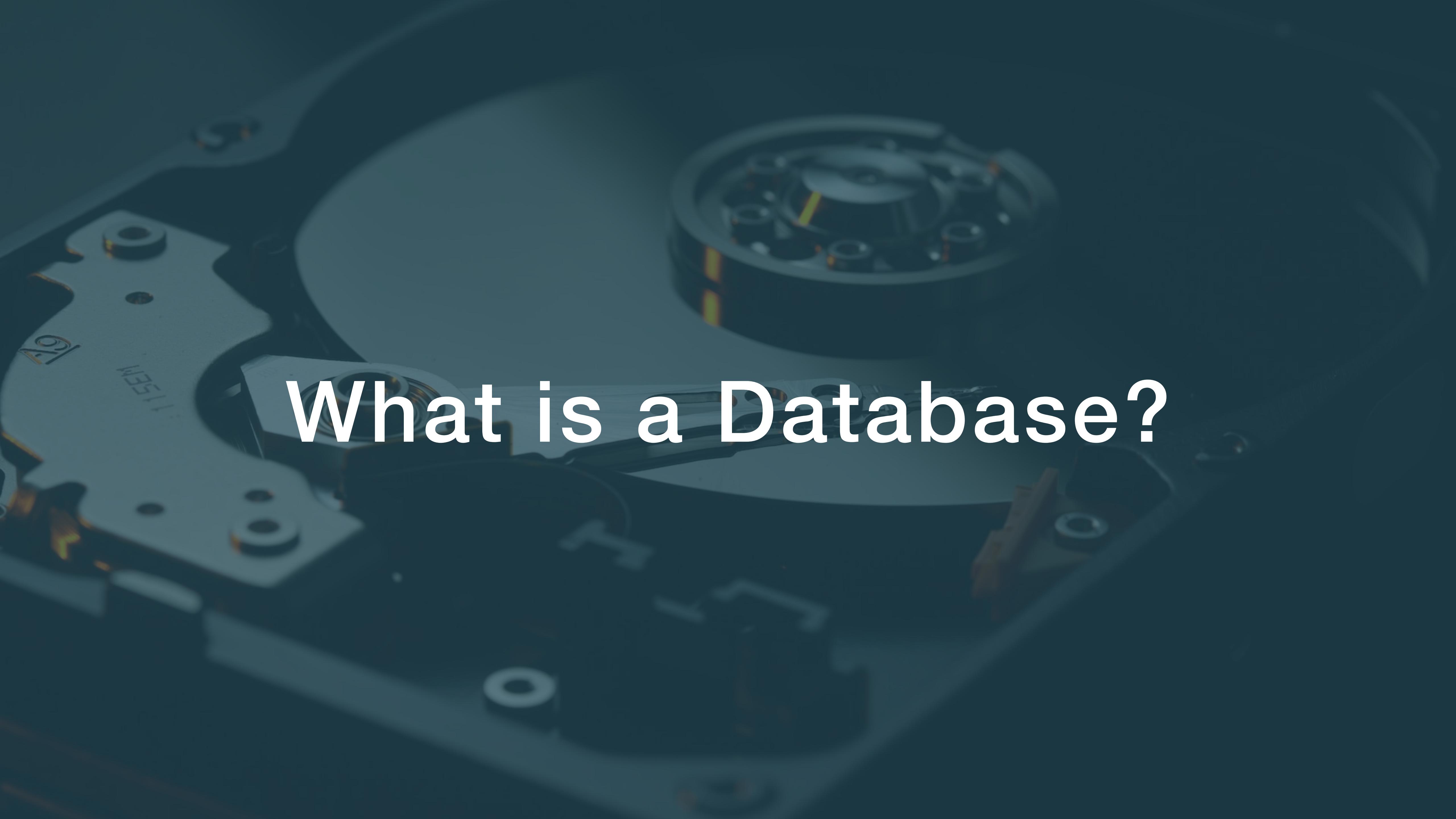


API





... a data source is like the place where we find and store our data, and databases are one of those places.



What is a Database?

Database

A database is an **organized collection of data** stored in a computer system. It's designed for efficient **data management, retrieval, and manipulation**. Databases are used in various fields, ensuring data structure, integrity, and security while allowing multiple users to access and update information concurrently. They come in different types to suit specific needs, like relational and NoSQL databases.



Key characteristics of a database

- **Data Organization**

Data is structured into tables, rows, and columns, ensuring integrity.

- **Data Rules**

Enforced rules maintain data accuracy, including data types, uniqueness, and relationships.

- **Data Retrieval**

Users efficiently extract specific data through queries.

- **Data Manipulation**

Support for adding, modifying, and deleting data keeps the database current.

- **Concurrent Access**

Multiple users can access and modify data simultaneously with consistency.

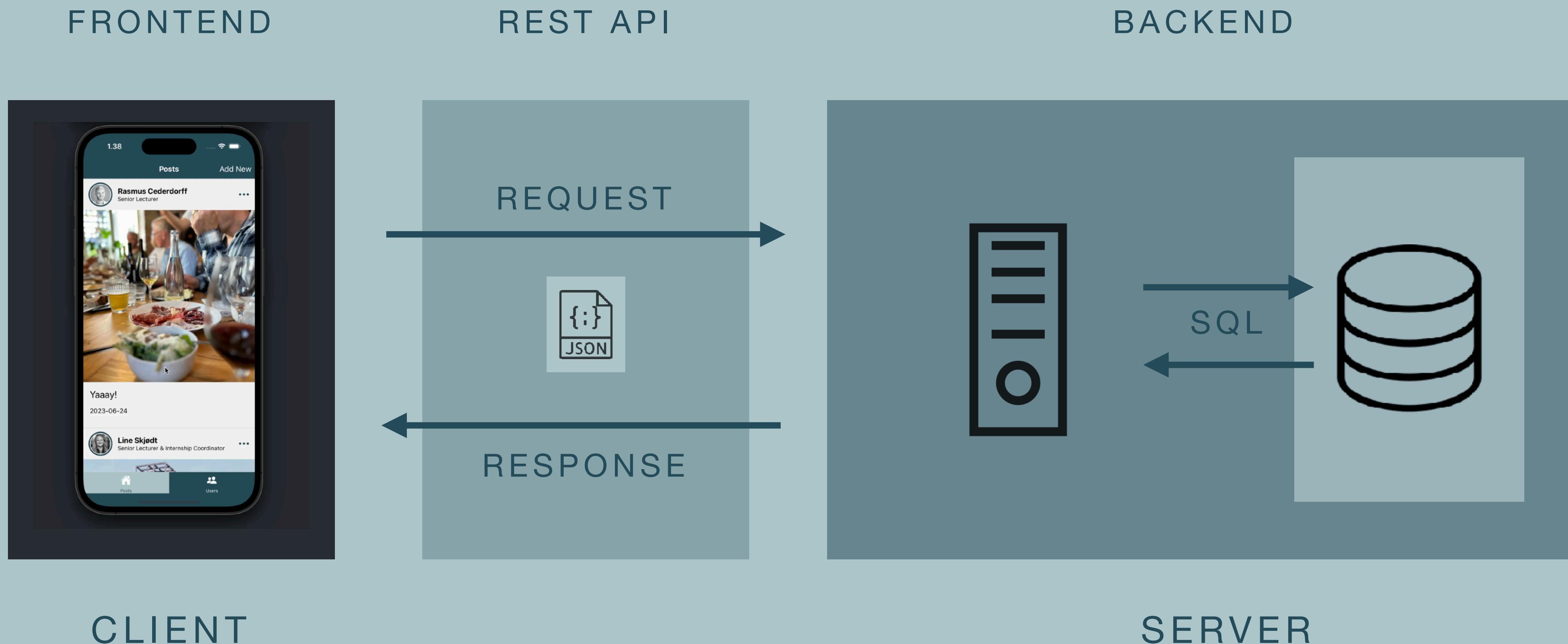
- **Security**

Features protect data with access control, authentication, and encryption.

- **Scalability**

Databases can grow to handle more data and users by adding resources or servers.

Web Dev Architecture



Types

There are various types of databases, including relational databases (such as MySQL, PostgreSQL, and Oracle), NoSQL databases (like MongoDB and Cassandra), and other specialized databases tailored to specific use cases.



ORACLE





Most Popular Databases



<https://survey.stackoverflow.co/2023/#section-most-popular-technologies-databases>

Most Popular Databases



<https://survey.stackoverflow.co/2023/#section-most-popular-technologies-databases>

FULLSTACK DEVELOPER



FRONTEND



BACKEND



DATABASE



TOOLS



HTML



PYTHON



CSS



JAVA



JAVASCRIPT



PHP



REACT



RUBY



ANGULAR



DJANGO



MY SQL



MONGODB



POSTGRE SQL



ORACLE



FIREBASE



VS CODE



GIT



GITHUB

100 *SECONDS OF*

SQL



<https://www.youtube.com/watch?v=zsjvFFK0m3c>

100 *SECONDS OF*



mongoDB

https://www.youtube.com/watch?v=-bt_y4Loofg

MySQL vs MongoDB

1. Definer med keywords MySQL og MongoDB
2. Definer ligheder og forskelle

- I skal gå sammen to og to.
- En tremandsgruppe
- Brug artikel og video

Relational Databases

A structured system for storing and managing data.

Key characteristics of relational database

- **Tables:** Data is organized into tables with rows and columns, defining the structure and data types.
- **Primary Keys:** Each table has a unique identifier (primary key) for distinct referencing.
- **Relationships:** Tables link through foreign keys, forming relationships like one-to-one, one-to-many, or many-to-many.
- **SQL:** SQL is used for data retrieval, manipulation, and management.
- **Data Integrity:** Constraints (e.g., unique, referential) maintain data accuracy and consistency.
- **ACID:** Relational databases follow ACID principles for reliable transactions.
- **Normalization:** Data is organized into related tables to reduce redundancy and improve integrity.
- **Indexing:** Indexes speed up data retrieval by locating specific data efficiently.
- **Transactions:** Sequences of SQL statements are treated as a single unit for consistent and reliable database changes.

SQL & MySQL



SQL

A standard language for storing, manipulating and retrieving data in and from a database.



Learn SQL In 60 Minutes

https://www.youtube.com/watch?v=p3qvj9hO_Bo

SQL

SQL stands for Structured Query Language

It is used for managing and manipulating data stored in relational databases.

SQL is used to interact and analyze large amounts of data. It is a key tool for Data Analysts and Scientists.

SQL is used for

- Creating, modifying, and deleting databases and tables
- Inserting, updating, and deleting data in databases
- Retrieving data from databases
- Setting permissions on database objects
- Creating and executing stored procedures

Basic SQL commands

- Some important SQL commands are SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, and ALTER.
- These commands allow you to retrieve, modify, and structure data in your database.

```
-- Create a New Table
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(256) NOT NULL,
    mail VARCHAR(256),
    title VARCHAR(256),
    image TEXT
);

-- Create New Row (entity or record)
INSERT INTO users (name, mail, title, image)
VALUES ('John Doe', 'johndoe@example.com', 'Software Eng');

-- Update the title column for user with id 1
UPDATE users
SET title = 'Senior Software Engineer'
WHERE id = 1;

-- Delete row (user) with id 2
DELETE FROM users
WHERE id = 2;
```

MySQL

A SQL Dialect.

An open-source relational database management system for storing and managing structured data.

MySQL

- **Relational Model:** Organizes data into tables for structured data management.
- **Open Source:** Freely available with accessible source code, fostering a large user and developer community.
- **Cross-Platform:** Compatible with various operating systems, making it versatile.
- **Scalable:** Handles both small and large databases, supporting high availability and scalability.
- **Performance:** Optimized with indexing, caching, and query optimization.
- **Security:** Offers robust security features like authentication, access control, encryption, and auditing.
- **ACID Compliance:** Ensures data consistency and reliability.
- **SQL Support:** Utilizes SQL for data manipulation.
- **Storage Engines:** Provides various options with distinct features.
- **Community and Support:** Benefits from a large user community and commercial support by Oracle.

MySQL - The Basics // Learn SQL in 23 Easy Steps



<https://www.youtube.com/watch?v=Cz3WcZLRaWc>

Tables & Entities

Tables and entities are core concepts in relational databases like MySQL. Tables organize data into rows and columns, where each row represents an instance of an entity, and each column represents an attribute of that entity. MySQL provides tools and SQL commands to work with these tables and entities effectively.

id	name	mail	title	image
1	Peter Lind	petl@kea.dk	Senior Lecturer	https://share.cederdorff.com/images/petl.jpg
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...
4	Edith Terte	edan@kea.dk	Lecturer	https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g/...
5	Frederikke Ben... ...n	fbe@kea.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...

Tables

In a relational database, a table is a two-dimensional data structure that consists of rows and columns. Each table is designed to store a specific type of data, and it represents an entity or concept in the real world.

 id	 name	 mail	 title	 image
1	Peter Lind	petl@kea.dk	Senior Lecturer	https://share.cederdorff.com/images/petl.jpg
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...
4	Edith Terte	edan@kea.dk	Lecturer	https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g...
5	Frederikke Ben... ...n	fbe@kea.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...

ROWS

Each row in a table represents a single record or instance of the entity being modeled. For example, in a "Users" table, each row could represent a different user.

 id	name	mail	title	image
1	Peter Lind	petl@kea.dk	Senior Lecturer	https://share.cederdorff.com/images/petl.jpg
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...
4	Edith Terte	edan@kea.dk	Lecturer	https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g/...
5	Frederikke Ben... <small>Frederikke Bensdorp</small>	fbe@kea.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...

Columns

Columns define the attributes or properties of the entity. Each column holds a specific type of data, such as text, numbers, dates, or even binary data.

id	name	mail	title	image
1	Peter Lind	petl@kea.dk	Senior Lecturer	https://share.cederdorff.com/images/petl.jpg
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...
4	Edith Terte	edan@kea.dk	Lecturer	https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g...
5	Frederikke Ben... ...nzen	fbe@kea.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...

Primary Key

A primary key is a column (or set of columns) in a table that uniquely identifies each row. It ensures that there are no duplicate records in the table.

id	name	mail	title	image
1	Peter Lind	petl@kea.dk	Senior Lecturer	https://share.cederdorff.com/images/petl.jpg
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...
4	Edith Terte	edan@kea.dk	Lecturer	https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g/...
5	Frederikke Ben... ...nzen	fbe@kea.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...

Entities

An entity in the context of databases refers to an object, concept, or thing that has attributes or properties. Entities are typically mapped to tables in a relational database. Each row in the table represents an instance of the entity, and each column corresponds to an attribute of the entity.

 id	 name	 mail	 title	 image
1	Peter Lind	petl@kea.dk	Senior Lecturer	https://share.cederdorff.com/images/petl.jpg
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...
4	Edith Terte	edan@kea.dk	Lecturer	https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g/...
5	Frederikke Ben... ...n	fbe@kea.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...

Entities

Let's say you are designing a database for a library. In this case, you might have an entity called "Books." Each book in the library is represented by a row in the "Books" table, and the attributes of a book (e.g., title, author, publication date) are represented as columns in the table.

 id	 name	 mail	 title	 image
1	Peter Lind	petl@kea.dk	Senior Lecturer	https://share.cederdorff.com/images/petl.jpg
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...
4	Edith Terte	edan@kea.dk	Lecturer	https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g/...
5	Frederikke Ben... ...n	fbe@kea.dk	Head of Education	https://kea.dk/slir/w200-c1x1/images/user-profile/chef...

SQL Statements

Commands or instructions that you use to interact with a relational database. They are used to perform various database operations, such as creating, retrieving, updating, and deleting data

```
-- Create a New Table
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(256) NOT NULL,
    mail VARCHAR(256),
    title VARCHAR(256),
    image TEXT
);

-- Create New Row (entity or record)
INSERT INTO users (name, mail, title, image)
VALUES ('John Doe', 'johndoe@example.com', 'Software
-- Update the title column for user with id 1
UPDATE users
SET title = 'Senior Software Engineer'
WHERE id = 1;

-- Delete row (user) with id 2
DELETE FROM users
WHERE id = 2;
```

DDL Statements (Data Definition Language)

CREATE TABLE: Creates a new table in the database.

ALTER TABLE: Modifies the structure of an existing table (e.g., adding or dropping columns).

DROP TABLE: Deletes an existing table and its data.

CREATE INDEX: Creates an index on one or more columns for performance optimization.

DML Statements (Data Manipulation Language)

SELECT: Retrieves data from one or more tables (SQL queries).

INSERT INTO: Adds new records (rows) to a table.

UPDATE: Modifies existing records in a table.

DELETE FROM: Removes records from a table.

DCL Statements (Data Control Language)

GRANT: Gives specific privileges to database users.

REVOKE: Revokes previously granted privileges.

CREATE DATABASE

... used to create a new SQL database.

```
CREATE DATABASE users_db;
```

... drop (delete) an existing SQL database.

```
DROP DATABASE users_db;
```

CREATE TABLE

Create a new table

```
-- Create a New Table
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    mail VARCHAR(255),
    title VARCHAR(255),
    image VARCHAR(255)
);
```

MySQL Data Types

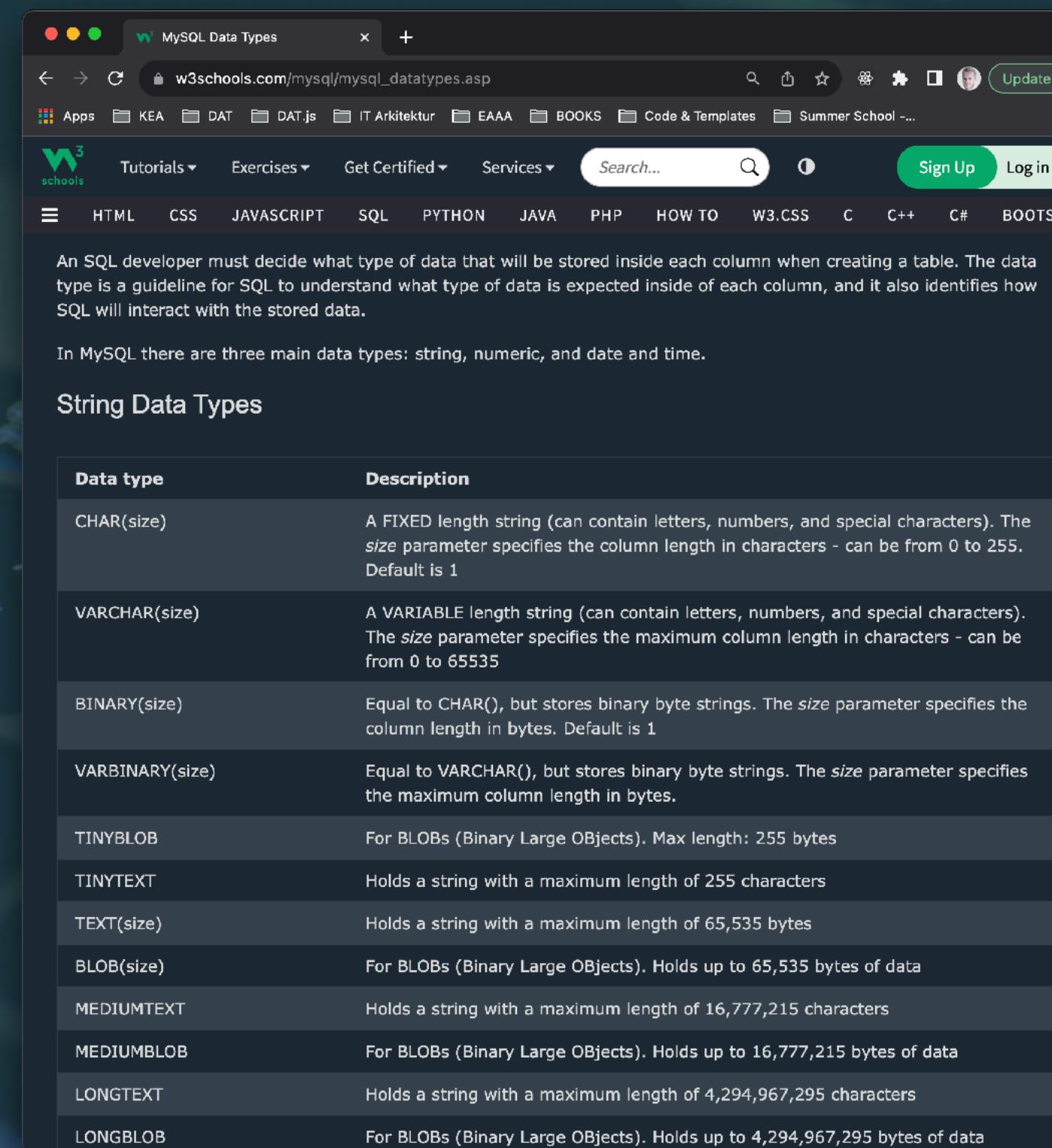
Every column in a table is required to have a data type

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The size parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The size parameter specifies the maximum column length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The size parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The size parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large OBjects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    mail VARCHAR(255) UNIQUE NOT NULL,
    title VARCHAR(100),
    image VARCHAR(500)
);
```

MySQL Data Types

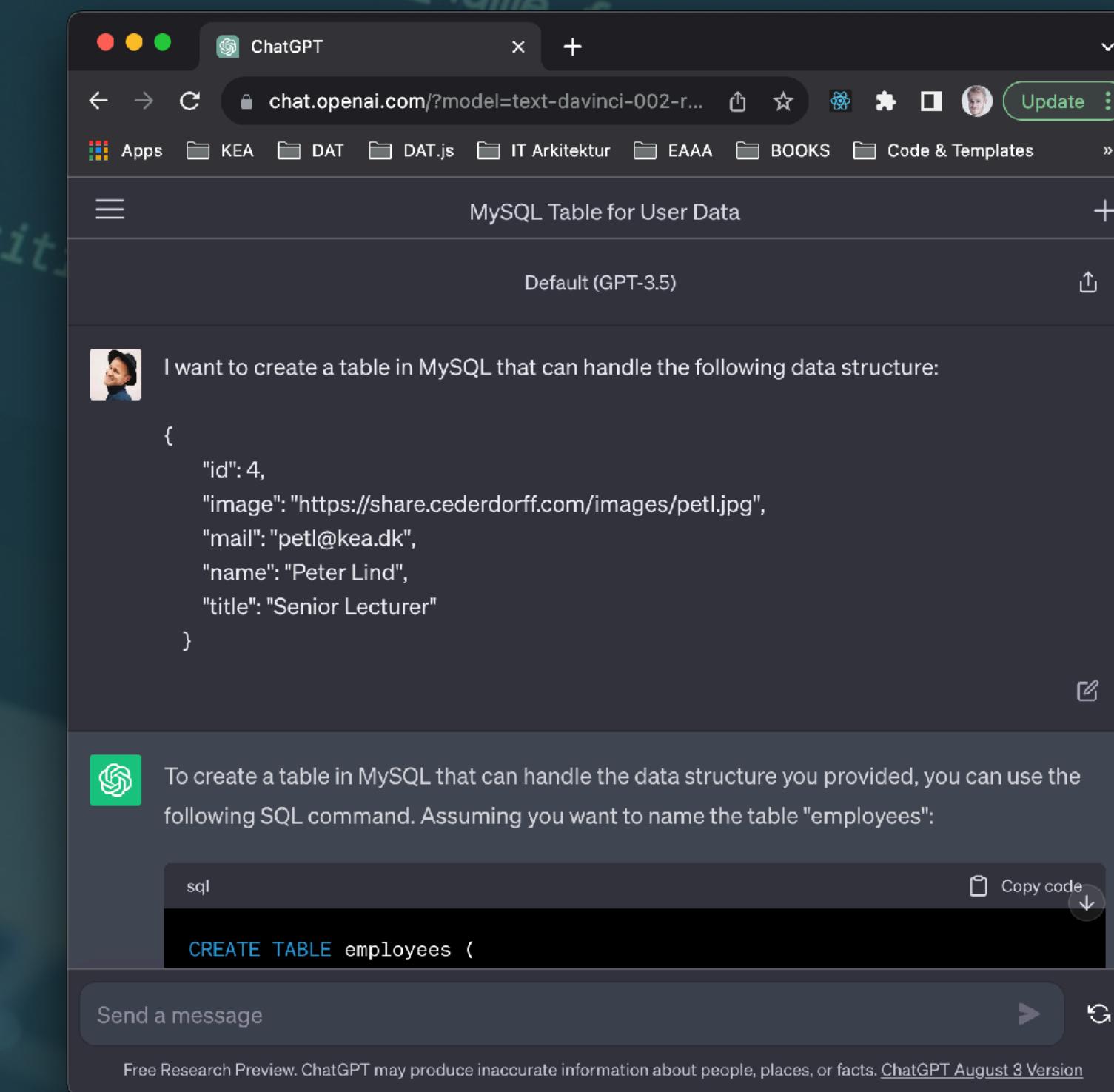
Ask your best friend!



The screenshot shows the MySQL Data Types page from w3schools.com. The page title is "MySQL Data Types". It starts with a brief introduction: "An SQL developer must decide what type of data that will be stored inside each column when creating a table. The data type is a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data." Below this, it states: "In MySQL there are three main data types: string, numeric, and date and time." The main content is titled "String Data Types" and contains a table with 14 rows, each listing a data type and its description:

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The size parameter specifies the column length in characters - can be from 0 to 255. Default is 1.
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The size parameter specifies the maximum column length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The size parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The size parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large OBjects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data

https://www.w3schools.com/mysql/mysql_datatypes.asp



The screenshot shows a ChatGPT session. The user asks: "I want to create a table in MySQL that can handle the following data structure:" followed by a JSON object:

```
{  
  "id": 4,  
  "image": "https://share.cederdorff.com/images/petl.jpg",  
  "mail": "petl@kea.dk",  
  "name": "Peter Lind",  
  "title": "Senior Lecturer"  
}
```

ChatGPT responds with: "To create a table in MySQL that can handle the data structure you provided, you can use the following SQL command. Assuming you want to name the table "employees":"

```
sql  
CREATE TABLE employees (
```

<https://chat.openai.com/share/23dd5dd2-abbe-45c1-b009-4010be8f5029>

INSERT

Insert a new record
(entity) into a given
table.

```
-- insert
INSERT INTO users_db.users (name, mail, title, image)
VALUES ('Peter Lind',
        'petl@kea.dk',
        'Senior Lecturer',
        'https://share.cederdorff.com/images/petl.jpg');
```

INSERT

Insert a new record (entity) into a table.

```
INSERT INTO [table name]
  (attribute1, attribute2, attribute3, etc.)
VALUES
  ("value1", "value2", "value3", etc.)
;
```

```
INSERT INTO users
  (firstname, lastname, age)
VALUES
  ("Kasper", "Topp", "34")
;
```

SELECT

Select data from a given database.

Specify what columns you want to.

```
-- select all  
SELECT * FROM users;
```

```
-- select name and title  
SELECT name, title  
FROM users;
```

WHERE

- The WHERE clause serves as a filter for database records.
- Its primary purpose is to extract and retrieve only those records that meet a specific set of conditions.

```
SELECT *
FROM users
WHERE title = 'Lecturer';
```

```
SELECT *
FROM users
WHERE title = 'Senior Lecturer';
```

```
SELECT *
FROM users
WHERE name = 'Edith Terte';
```

ORDER BY

- Sort the resultset in ascending or descending order

```
-- Select all and order by name  
SELECT *  
FROM users  
ORDER BY name;
```

```
-- Select all and order by title  
SELECT *  
FROM users  
ORDER BY title;
```

```
-- Select all and order by name, descending order  
SELECT *  
FROM users  
ORDER BY name DESC;
```

UPDATE

Modify the existing records
(entities) in a table.

Specify the columns you
want to update.

```
-- Update the title column for user with id 1
UPDATE users
SET title = 'Senior Software Engineer'
WHERE id = 1;

-- Update title and mail for user with id 2
UPDATE users
SET title = 'Senior Software Engineer',
    mail = 'race@eaaa.dk'
WHERE id = 2;
```

DELETE

Delete existing records in a table.

In this case by a specified id.

```
-- Delete row (user) with id 2  
DELETE FROM users  
WHERE id = 2;
```

SQL Queries

A specific type of SQL statement used to retrieve data from one or more tables in a database. SQL queries are used to answer questions or extract information from the database. They are primarily associated with the SELECT statement, which is the most commonly used SQL query.

A SQL query typically specifies the following: Columns, Table(s), Filtering Criteria (WHERE), Sorting Order (ORDER BY), Aggregation and Grouping (GROUP BY, SUM, COUNT, AVG, etc.)

SQL Queries

Retrieving Data (SELECT)

```
-- Read data (name and mail column) from table (user)
SELECT name, mail
FROM users
WHERE title = 'Software Engineer'
ORDER BY name;
```

Retrieves the "name" and "mail" columns from the "users" table for users with the title "Software Engineer" and orders the results by the "name" column.

SQL Queries

Search & Retrieving Data (SELECT & LIKE)

```
SELECT *  
FROM your_table_name  
WHERE name_column_name LIKE '%search_term%';
```

- LIKE Operator: Used to search for patterns in a column.
- % Wildcard Character: Used with LIKE to match any characters (including none) in a pattern.
- In the SQL query, % is used to search for rows where the email column ends with "@kea.dk," regardless of what precedes the "@" symbol.

SQL Queries

Search & Retrieving Data (SELECT & LIKE)

```
SELECT *  
FROM users  
WHERE mail LIKE '%@kea.dk';
```

- LIKE Operator: Used to search for patterns in a column.
- % Wildcard Character: Used with LIKE to match any characters (including none) in a pattern.
- In the SQL query, % is used to search for rows where the email column ends with "@kea.dk," regardless of what precedes the "@" symbol.

GROUP BY

Group rows with the same values into summary rows, like find the number of users with each title

```
-- counts the number of different titles  
SELECT title, COUNT(title) as count  
FROM users  
GROUP BY title;
```

	title	count
1	Senior Lecturer	2
2	Head of Education	2
3	Lecturer	1

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

GROUP BY

Group rows with the same values into summary rows, like find the number of users with each title

```
-- counts the number of different titles  
SELECT title, COUNT(title) as count  
FROM users  
GROUP BY title;
```

	title	count
1	Senior Lecturer	2
2	Head of Education	2
3	Lecturer	1

counts the number of users in each job title and groups the results by the "title" column.

HAVING

You cannot use WHERE with an aggregate function (COUNT(), MAX(), MIN(), SUM(), AVG()). Therefor HAVING was added to SQL.

```
SELECT title, COUNT(title) as count  
FROM users  
GROUP BY title  
HAVING count > 1;
```

	title	count
1	Senior Lecturer	2
2	Head of Education	2

counts the number of users in each job title and groups the results by the "title" column - but only the counts bigger than 1

SQL Queries

Joining Tables (INNER JOIN)

```
SELECT *  
FROM posts  
INNER JOIN users ON posts.createdBy = users.id;
```

Selects and joins "posts" and "users" tables based on the "createdBy" column

#	caption	postImage	createdAt	name	title	ht
1	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Maria Louise Bendixen	Senior Lecturer	ht
2	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Rasmus Cederdorff	Senior Lecturer	ht
3	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Rasmus Cederdorff	Senior Lecturer	ht
4	A cozy morning with coffee	https://images.unsplash.com/photo-1545319261-f3760f9dd64d?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Rasmus Cederdorff	Senior Lecturer	ht
5	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-dbb3de50b8b2?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Anne Kirketerp	Head of Department	ht
6	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Line Skjødt	Senior Lecturer & Internship Coordinator	ht
7	Serenity of the forest	https://images.unsplash.com/photo-1661505216710-32316e7b5bb3?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Dan Okkels Brendstrup	Lecturer	ht
8	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Dan Okkels Brendstrup	Lecturer	ht

Keys & IDs

A key is a field or combination of fields that is used to uniquely identify records (rows) within a table. Keys are essential for maintaining data integrity and for establishing relationships between tables.

An ID, short for identifier, is a unique value that is assigned to each record in a table. It is often used as the primary key to uniquely identify records within the table.

Types of Keys

- Primary Key: A primary key is a special type of key that uniquely identifies each row in a table. It ensures that there are no duplicate records in the table. There can be only one primary key in a table.
- Foreign Key: A foreign key is a field in one table that is used to establish a link between the data in two related tables. It typically refers to the primary key of another table and enforces referential integrity.
- Unique Key: A unique key ensures that the values in a particular field or combination of fields are unique across all rows in the table, but unlike the primary key, it allows for one or more rows to have null values.
- Composite Key: A composite key is a key that consists of two or more columns used together to uniquely identify rows in a table.

Primary Key

```
-- Create a New Table
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    mail VARCHAR(255),
    title VARCHAR(100),
    image VARCHAR(500)
);
```

- Unique identifier for each entity (row)
- Used to find an entity in a table.
- Entity must ALWAYS have a Primary Key!

Auto-Increment

In many database systems, including MySQL, you can define a primary key column with an auto-increment attribute. This means that the database system automatically assigns a unique ID to each new record added to the table. It simplifies the process of inserting new records and ensures that each record has a unique identifier.

✓	users
✓	columns 5
	id int (auto increment)
	name varchar(255)
	mail varchar(255)
	title varchar(255)
	image varchar(255)

Auto-Increment

```
-- Create a New Table
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    mail VARCHAR(255),
    title VARCHAR(100),
    image VARCHAR(500)
);
```

users	
columns 5	
o♀	id int (auto increment)
□	name varchar(255)
□	mail varchar(255)
□	title varchar(255)
□	image varchar(255)

Foreign Key

A foreign key is a field in one table that is used to establish a link between the data in two related tables. It typically refers to the primary key of another table and enforces referential integrity.

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    mail VARCHAR(255) UNIQUE NOT NULL,
    title VARCHAR(100),
    image TEXT
);

CREATE TABLE posts(
    id INT AUTO_INCREMENT PRIMARY KEY,
    caption VARCHAR(255) NOT NULL,
    image TEXT NOT NULL,
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    createdBy INT,
    FOREIGN KEY (createdBy) REFERENCES users(id)
);
```

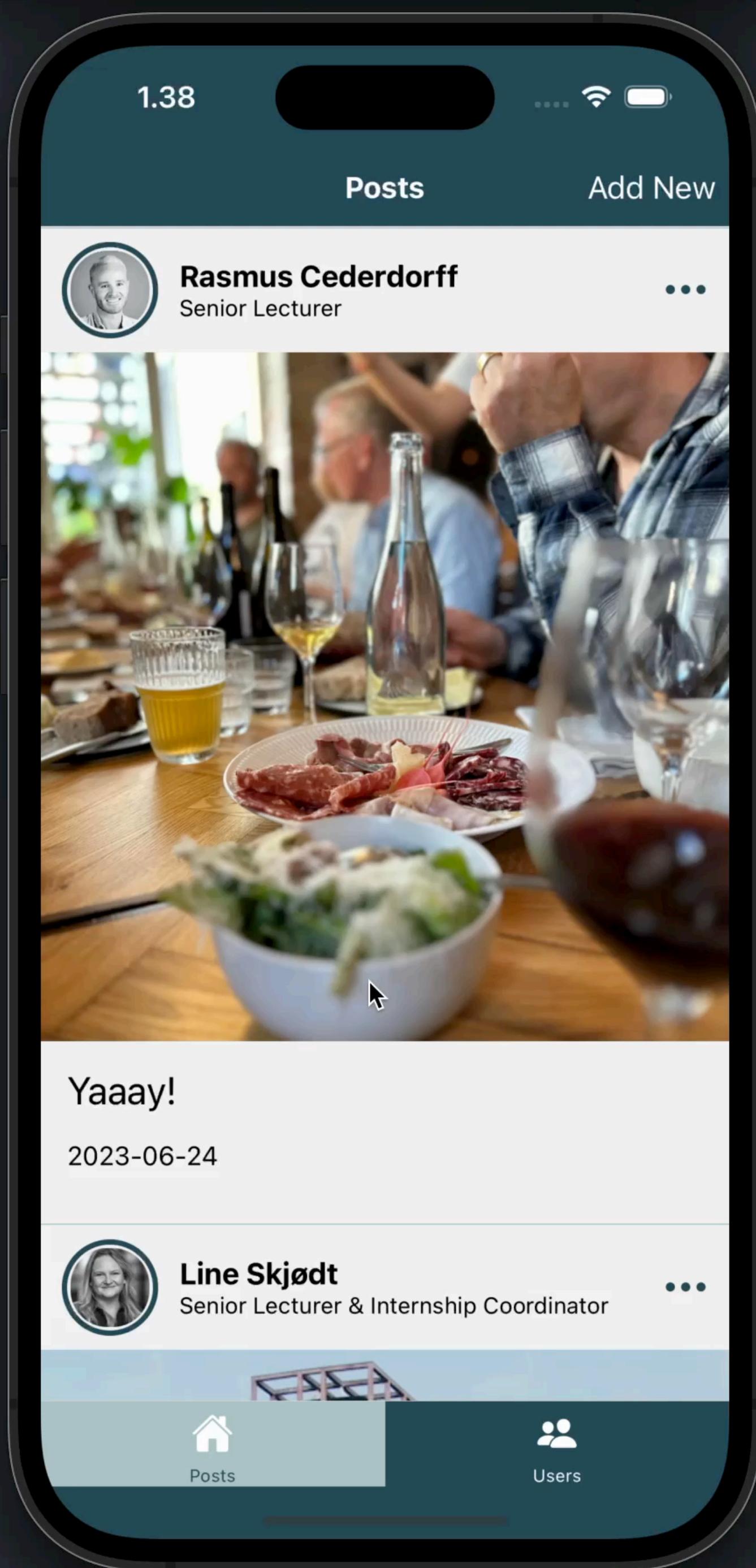
Unique Key

```
-- Create a New Table
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    mail VARCHAR(255) UNIQUE NOT NULL,
    title VARCHAR(100),
    image VARCHAR(500)
);
```

A unique key ensures that the values in a particular field or combination of fields are unique across all rows in the table, but unlike the primary key, it allows for one or more rows to have null values.

Entity Relationships

In many cases, entities in a database are related to each other. MySQL supports defining and maintaining relationships between tables through concepts like foreign keys, which ensure data integrity and enable complex querying.



Posts & Users

React Firebase REST Post App https://race-rest.web.app

POSTS CREATE

Morten Algy Bonderup
Senior Lecturer

Qui est esse

Est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

Dan Okkels Brendstrup
Lecturer

Consequuntur deleniti eos quia temporibus ab aliquid at

Voluptatem cumque tenetur consequatur expedita ipsum nemo quia explicabo aut eum minima consequatur tempore cumque quae est et et in consequuntur voluptatem voluptates aut

Kim Elkjær Marcher-Jepsen
Senior Lecturer

At nam consequatur ea labore ea harum

Cupiditate quo est a modi nesciunt soluta ipsa voluptas error itaque dicta in autem qui minus magnam et distinctio eum accusamus ratione error aut

Birgitte Kirk Iversen
Senior Lecturer

Jes Arbov
Lecturer

Maria Louise Bendixen
Senior Lecturer

posts users

1.38



Posts Add New

Rasmus Cederdorff
Senior Lecturer

Yaaay!

2023-06-24

Line Skjødt
Senior Lecturer & Internship Coordinator

Posts Users

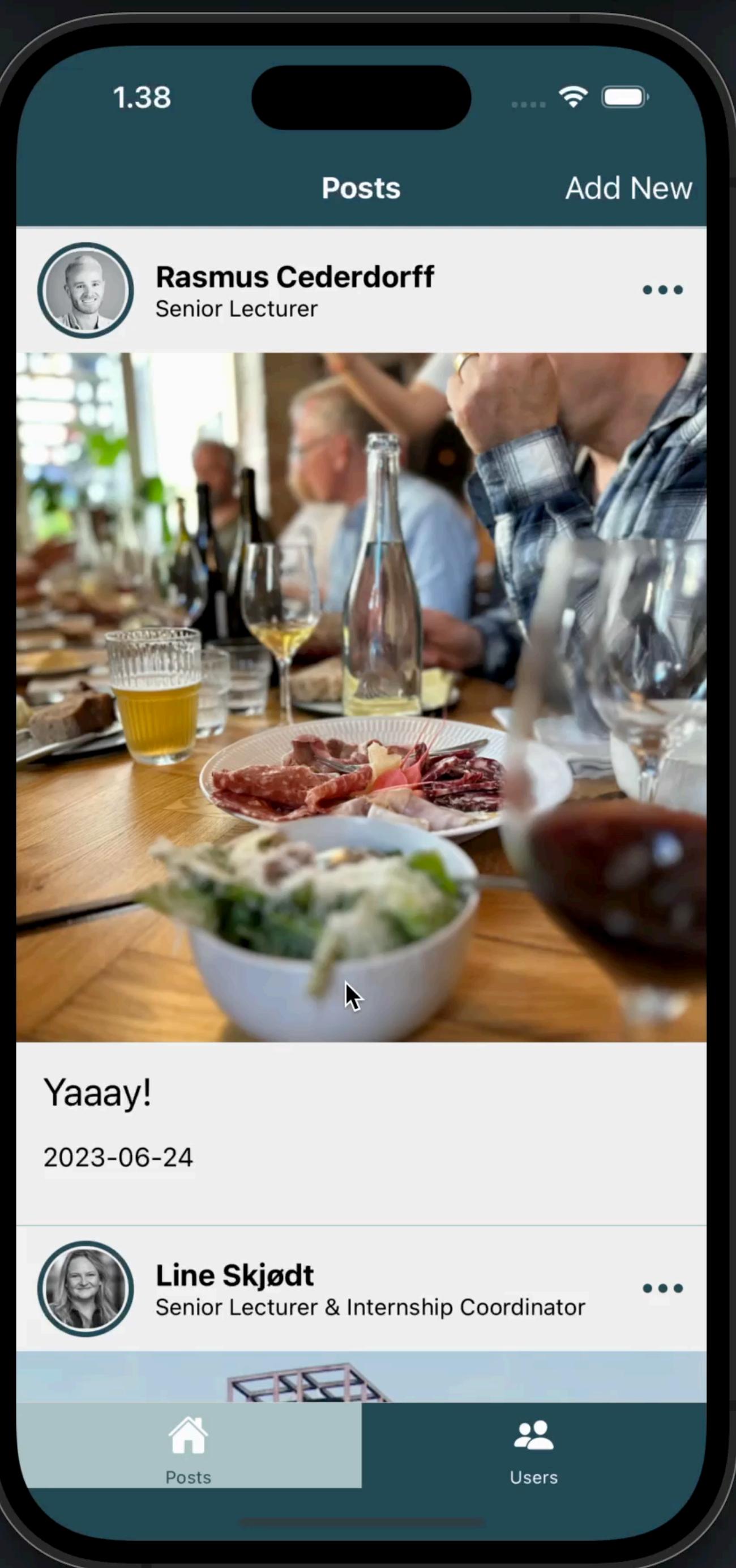
How are they related?

Posts

	id	caption	image	createdAt
1	1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07
2	2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07
3	3	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07
4	4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07
5	5	A cozy morning with coffee	https://images.unsplash.com/photo-1545319261-f3760f9dd64d?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07
6	6	Serenity of the forest	https://images.unsplash.com/photo-1661505216710-32316e7b5bb3?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07
7	7	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07
8	8	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-dbb3de50b8b2?ixlib=rb-4.0.3&ixid=	2023-09-13 16:51:07

Users

	id	name	mail	title	image
1	1	Maria Louise Bendixen	mlbe@eaaa.dk	Senior Lecturer	https://www.baaa.dk/media/b5ahrlra/maria-loui...
2	2	Rasmus Cederdorff	race@eaaa.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	3	Anne Kirketerp	anki@eaaa.dk	Head of Department	https://www.baaa.dk/media/5buh1xeo/anne-kirke...
4	4	Line Skjødt	lskj@eaaa.dk	Senior Lecturer & Internship Coord...	https://www.eaaa.dk/media/14qpfeq4/line-skjod...
5	5	Dan Okkels Brendstrup	dob@eaaa.dk	Lecturer	https://www.eaaa.dk/media/bdojel41/dan-okkels...



How are they related?

Posts

	<input type="checkbox"/> id	<input type="checkbox"/> caption	<input type="checkbox"/> image	<input type="checkbox"/> createdAt	<input type="checkbox"/> createdBy
1	1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3	2023-09-13 17:08:22	2
2	2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3	2023-09-13 17:08:22	4
3	3	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3	2023-09-13 17:08:22	1
4	4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3	2023-09-13 17:08:22	2
5	5	A cozy morning with coffee	https://images.unsplash.com/photo-1545319261-f3760f9dd64d?ixlib=rb-4.0.3	2023-09-13 17:08:22	2
6	6	Serenity of the forest	https://images.unsplash.com/photo-1661505216710-32316e7b5bb3?ixlib=rb-4.0.3	2023-09-13 17:08:22	5
7	7	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=rb-4.0.3	2023-09-13 17:08:22	5
8	8	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-dbb3de50b8b2?ixlib=rb-4.0.3	2023-09-13 17:08:22	3

Users

	<input type="checkbox"/> id	<input type="checkbox"/> name	<input type="checkbox"/> mail	<input type="checkbox"/> title	<input type="checkbox"/> image
1	1	Maria Louise Bendixen	mlbe@eaaa.dk	Senior Lecturer	https://www.baaa.dk/media/b5ahrlra/maria-loui...
2	2	Rasmus Cederdorff	race@eaaa.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	3	Anne Kirketerp	anki@eaaa.dk	Head of Department	https://www.baaa.dk/media/5buh1xeo/anne-kirke...
4	4	Line Skjødt	lskj@eaaa.dk	Senior Lecturer & Internship Coord...	https://www.eaaa.dk/media/14qpfeq4/line-skjod...
5	5	Dan Okkels Brendstrup	dob@eaaa.dk	Lecturer	https://www.eaaa.dk/media/bdojel41/dan-okkels...

Entity Relationships

Posts

	id	caption	image	createdAt	createdBy
1	1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=	2023-09-13 17:08:22	2
2	2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=	2023-09-13 17:08:22	4
3	3	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=	2023-09-13 17:08:22	1
4	4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=	2023-09-13 17:08:22	2
5	5	A cozy morning with coffee	https://images.unsplash.com/photo-1545319261-f3760f9dd64d?ixlib=	2023-09-13 17:08:22	2
6	6	Serenity of the forest	https://images.unsplash.com/photo-1661505216710-32316e7b5bb3?ixlib=	2023-09-13 17:08:22	5
7	7	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=	2023-09-13 17:08:22	5
8	8	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-dbb3de50b8b2?ixlib=	2023-09-13 17:08:22	3

Foreign key

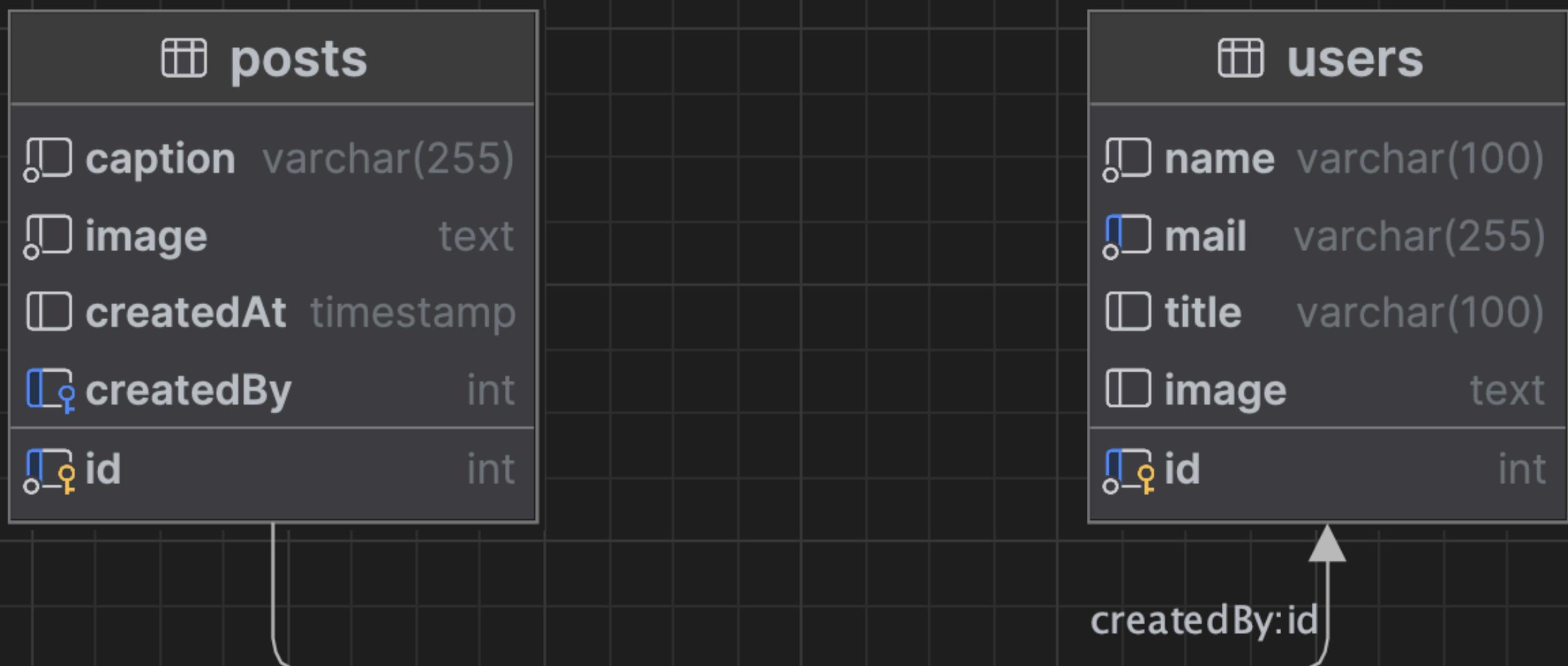
- The foreign links posts createdBy column to users table's primary key.
- It establishes a relationship between posts and users

Users

	id	name	mail	title	image
1	1	Maria Louise Bendixen	mlbe@eaaa.dk	Senior Lecturer	https://www.baaa.dk/media/b5ahrlra/maria-loui...
2	2	Rasmus Cederdorff	race@eaaa.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	3	Anne Kirketerp	anki@eaaa.dk	Head of Department	https://www.baaa.dk/media/5buh1xeo/anne-kirke...
4	4	Line Skjødt	lskj@eaaa.dk	Senior Lecturer & Internship Coord...	https://www.eaaa.dk/media/14qpfeq4/line-skjod...
5	5	Dan Okkels Brendstrup	dob@eaaa.dk	Lecturer	https://www.eaaa.dk/media/bdojel41/dan-okkels...

Entity Relationships

One to many



One to many

Posts

	id	caption	image	createdAt	createdBy
1	1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3	2023-09-13 17:08:22	2
2	2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3	2023-09-13 17:08:22	4
3	3	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3	2023-09-13 17:08:22	1
4	4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3	2023-09-13 17:08:22	2
5	5	A cozy morning with coffee	https://images.unsplash.com/photo-1545319261-f3760f9dd64d?ixlib=rb-4.0.3	2023-09-13 17:08:22	2
6	6	Serenity of the forest	https://images.unsplash.com/photo-1661505216710-32316e7b5bb3?ixlib=rb-4.0.3	2023-09-13 17:08:22	5
7	7	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=rb-4.0.3	2023-09-13 17:08:22	5
8	8	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-db3de50b8b2?ixlib=rb-4.0.3	2023-09-13 17:08:22	3

Users

	id	name	mail	title	image
1	1	Maria Louise Bendixen	mlbe@aaaa.dk	Senior Lecturer	https://www.baaa.dk/media/b5ahr1ra/maria-loui...
2	2	Rasmus Cederdorff	race@aaaa.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	3	Anne Kirketerp	anki@aaaa.dk	Head of Department	https://www.baaa.dk/media/5buh1xeo/anne-kirke...
4	4	Line Skjødt	lskj@aaaa.dk	Senior Lecturer & Internship Coord.	https://www.eaaa.dk/media/14qpfeq4/line-skjod...
5	5	Dan Okkels Brendstrup	dob@aaaa.dk	Lecturer	https://www.eaaa.dk/media/bdoje141/dan-okkels...

```
CREATE TABLE users (
```

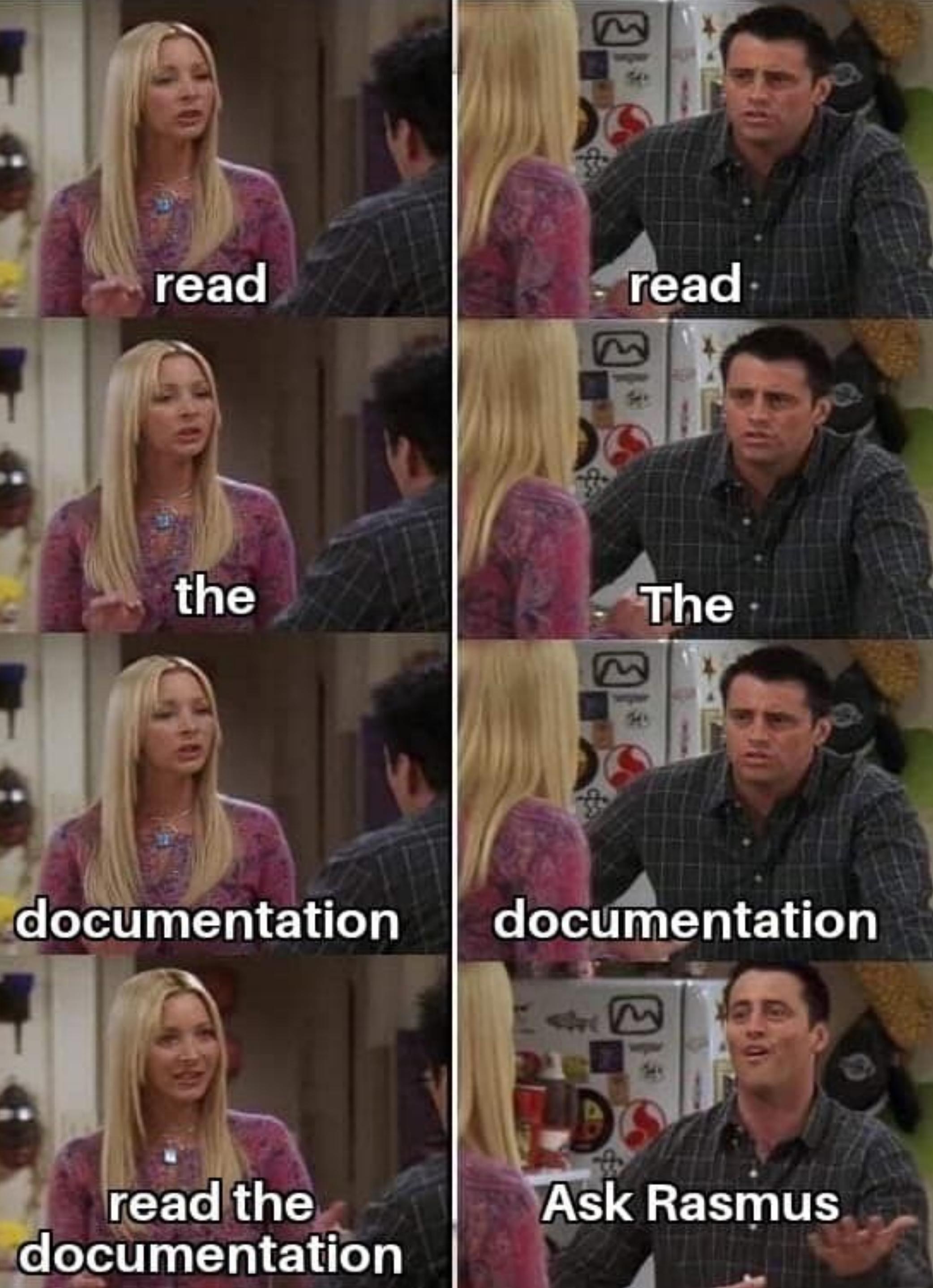
```
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    mail VARCHAR(255) UNIQUE NOT NULL,  
    title VARCHAR(100),  
    image TEXT  
);
```

```
CREATE TABLE posts(
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,  
    caption VARCHAR(255) NOT NULL,  
    image TEXT NOT NULL,  
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    createdBy INT,  
    FOREIGN KEY (createdBy) REFERENCES users(id)  
);
```

How do we select ALL the data, then?

```
SELECT *  
FROM posts, users  
WHERE posts.createdBy = users.id;
```



Using WHERE

```
-- Select all (posts and users) where createBy is equal to user id
SELECT *
FROM posts,
      users
WHERE posts.createdBy = users.id;
```

```
-- Select selected column with aliases
SELECT posts.caption,
       posts.image AS postImage,
       posts.createdAt,
       users.name,
       users.title,
       users.image AS userImage
FROM posts, users
WHERE posts.createdBy = users.id;
```

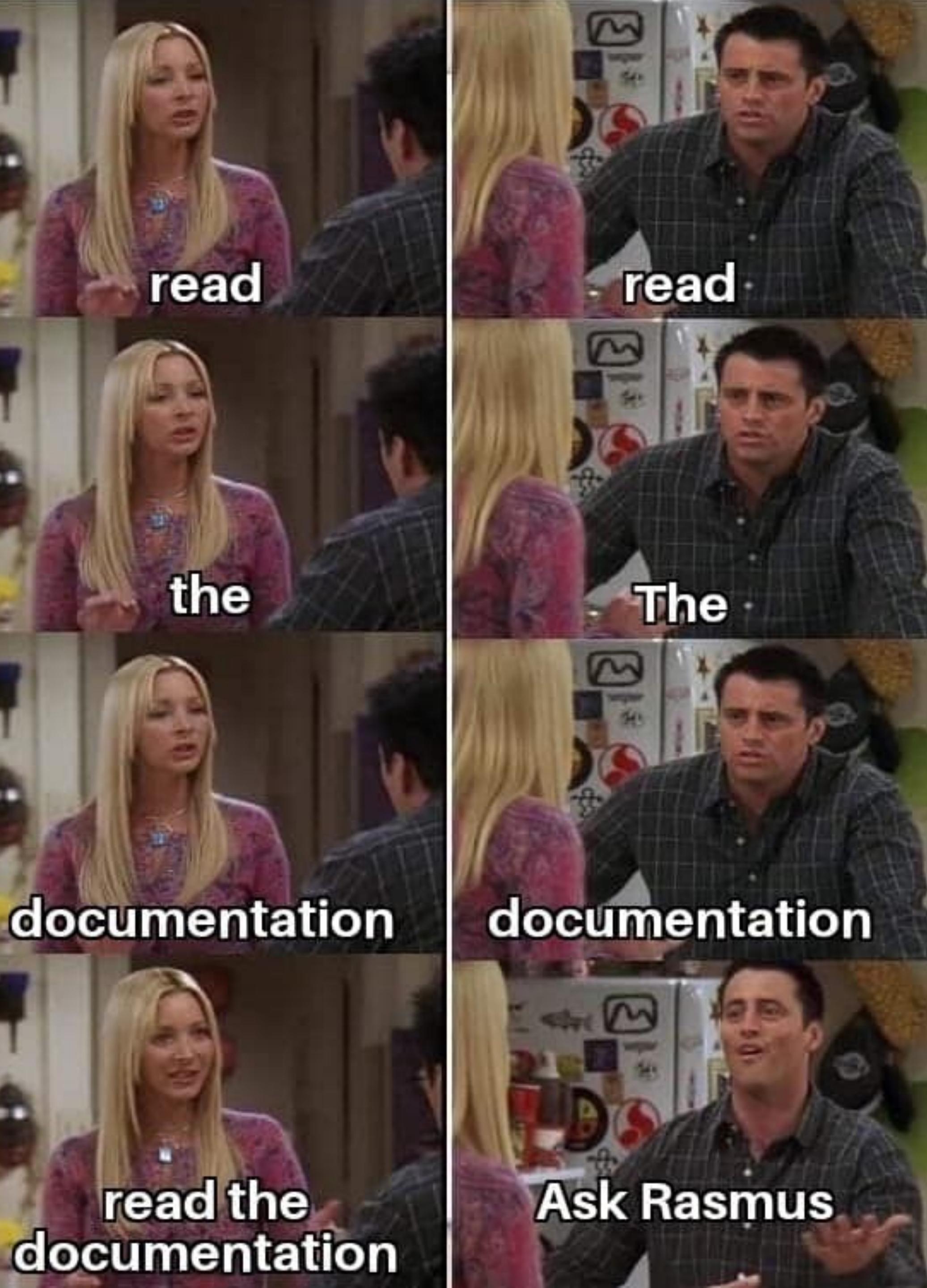
```
-- Select Posts by a Specific User (user with id 2)
SELECT * FROM posts
WHERE createdBy = 2;
```

	caption	postImage	createdAt	name	title	
1	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Maria Louise Bendixen	Senior Lecturer	ht
2	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Rasmus Cederdorff	Senior Lecturer	ht
3	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Rasmus Cederdorff	Senior Lecturer	ht
4	A cozy morning with coffee	https://images.unsplash.com/photo-1545319261-f3760f9dd64d?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Rasmus Cederdorff	Senior Lecturer	ht
5	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-dbb3de50b8b2?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Anne Kirketerp	Head of Department	ht
6	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Line Skjødt	Senior Lecturer & Internship Coordinator	ht
7	Serenity of the forest	https://images.unsplash.com/photo-1661505216710-32316e7b5bb3?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Dan Okkels Brendstrup	Lecturer	ht
8	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=rb-4.0.3&ixid=...	2023-09-13 18:08:13	Dan Okkels Brendstrup	Lecturer	ht

But JOIN is be a better solution



```
SELECT *  
FROM posts  
INNER JOIN users ON posts.createdBy = users.id;
```



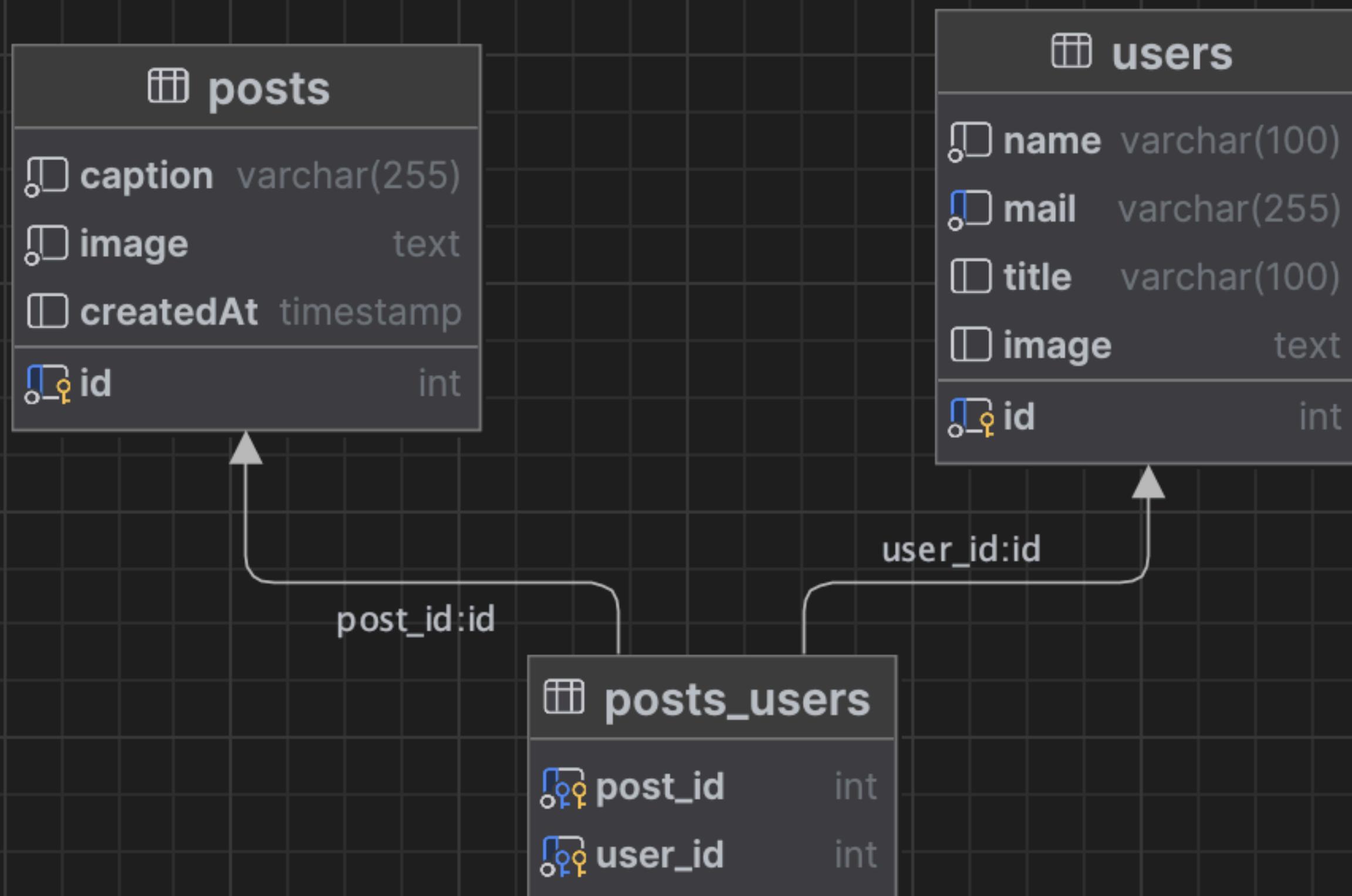
But what if a post could have several users as author?

Imagine a post or a book could be written by several writers. Or a song could be written or recorded by several artists.

Then we have another relationship between the twos.

Entity Relationships

Many to many



Junction Table

posts_users

	post_id	user_id
1	2	1
2	1	2
3	2	2
4	1	4
5	2	5
6	4	5

posts

	id	caption	image	createdAt
1	1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
2	2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
3	3	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
4	4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
5	5	A cozy morning with coffee	https://images.unsplash.com/photo-1545319261-f3760f9dd64d?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
6	6	Serenity of the forest	https://images.unsplash.com/photo-1661505216710-32316e7b5bb3?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
7	7	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
8	8	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-dbb3de50b8b2?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07

users

	id	name	mail	title	image
1	1	Maria Louise Bendixen	mlbe@eaaa.dk	Senior Lecturer	https://www.baaa.dk/media/b5ahrlra/maria-loui...
2	2	Rasmus Cederdorff	race@eaaa.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	3	Anne Kirketerp	anki@eaaa.dk	Head of Department	https://www.baaa.dk/media/5buh1xeo/anne-kirke...
4	4	Line Skjødt	lskj@eaaa.dk	Senior Lecturer & Internship Coord...	https://www.eaaa.dk/media/14qpfeq4/line-skjod...
5	5	Dan Okkels Brendstrup	dob@eaaa.dk	Lecturer	https://www.eaaa.dk/media/bdojel41/dan-okkels...

Junction Table

posts_users

	post_id	user_id
1	2	1
2	1	2
3	2	2
4	1	4
5	2	5
6	4	5

posts

	id	caption	image	createdAt
1	1	Beautiful sunset at the beach	https://images.unsplash.com/photo-15442183278117e0530b2c21xlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
2	2	Fishing boat in the harbor	https://images.unsplash.com/photo-15442183671117e0530b2c21xlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
3	3	Delicious food at the restaurant	https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
4	4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
5	5	A cozy morning with coffee	https://images.unsplash.com/photo-1543122411-f7760f9dd4?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
6	6	Serenity on the lake	https://images.unsplash.com/photo-1561521523167b7bb7?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
7	7	A beautiful morning in Aarhus	https://images.unsplash.com/photo-1573997953524-efed43db70a0?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07
8	8	Rainbow reflections of the city of Aarhus	https://images.unsplash.com/photo-1558443336-dbb3de50b8b2?ixlib=rb-4.0.3&ixid...	2023-09-13 16:51:07

A junction table contains the primary key columns of two related tables, facilitating many-to-many relationships between them.

	id	name	mail	title	image
1	1	Maria Louise Bendixen	mlbe@aaaa.dk	Senior Lecturer	https://www.baaa.dk/media/b5ahrlra/maria-loui...
2	2	Rasmus Cederdorff	race@aaaa.dk	Senior Lecturer	https://share.cederdorff.com/images/race.jpg
3	3	Anne Kirketerp	anki@aaaa.dk	Head of Department	https://www.baaa.dk/media/5buh1xeo/anne-kirke...
4	4	Line Skjødt	lskj@aaaa.dk	Senior Lecturer & Internship Coord.	https://www.eaaa.dk/media/14qpfeq4/line-skjod...
5	5	Dan Okkels Brendstrup	dob@aaaa.dk	Lecturer	https://www.eaaa.dk/media/bdojel41/dan-okkels...

Many to many

posts

posts_users

	post_id	user_id
1	2	1
2	1	2
3	2	2
4	1	4
5	2	5
6	4	5

users

	id	name	mail
1	1	Maria Louise Bendixen	mlbe@eaaa
2	2	Rasmus Cederdorff	race@eaaa
3	3	Anne Kirketerp	anki@eaaa
4	4	Line Skjødt	lskj@eaaa
5	5	Dan Okkels Brendstrup	dob@eaaa

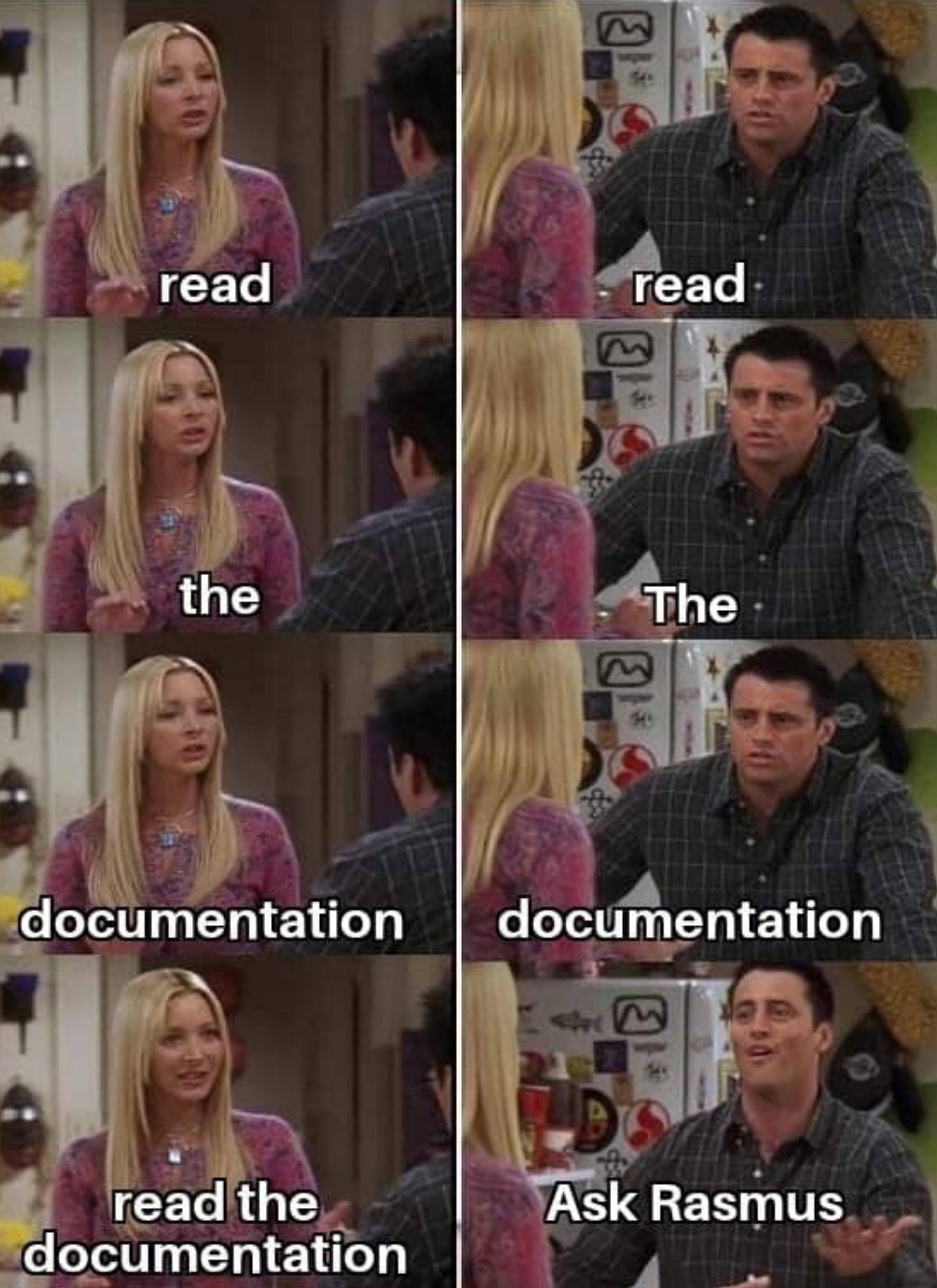
```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    mail VARCHAR(255) UNIQUE NOT NULL,
    title VARCHAR(100),
    image TEXT
);
```

```
CREATE TABLE posts(
    id INT AUTO_INCREMENT PRIMARY KEY,
    caption VARCHAR(255) NOT NULL,
    image TEXT NOT NULL,
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
-- Create a Junction Table to link posts and users
CREATE TABLE posts_users (
    post_id INT,
    user_id INT,
    PRIMARY KEY (post_id, user_id),
    FOREIGN KEY (post_id) REFERENCES posts(id),
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

How do we select ALL the data now?

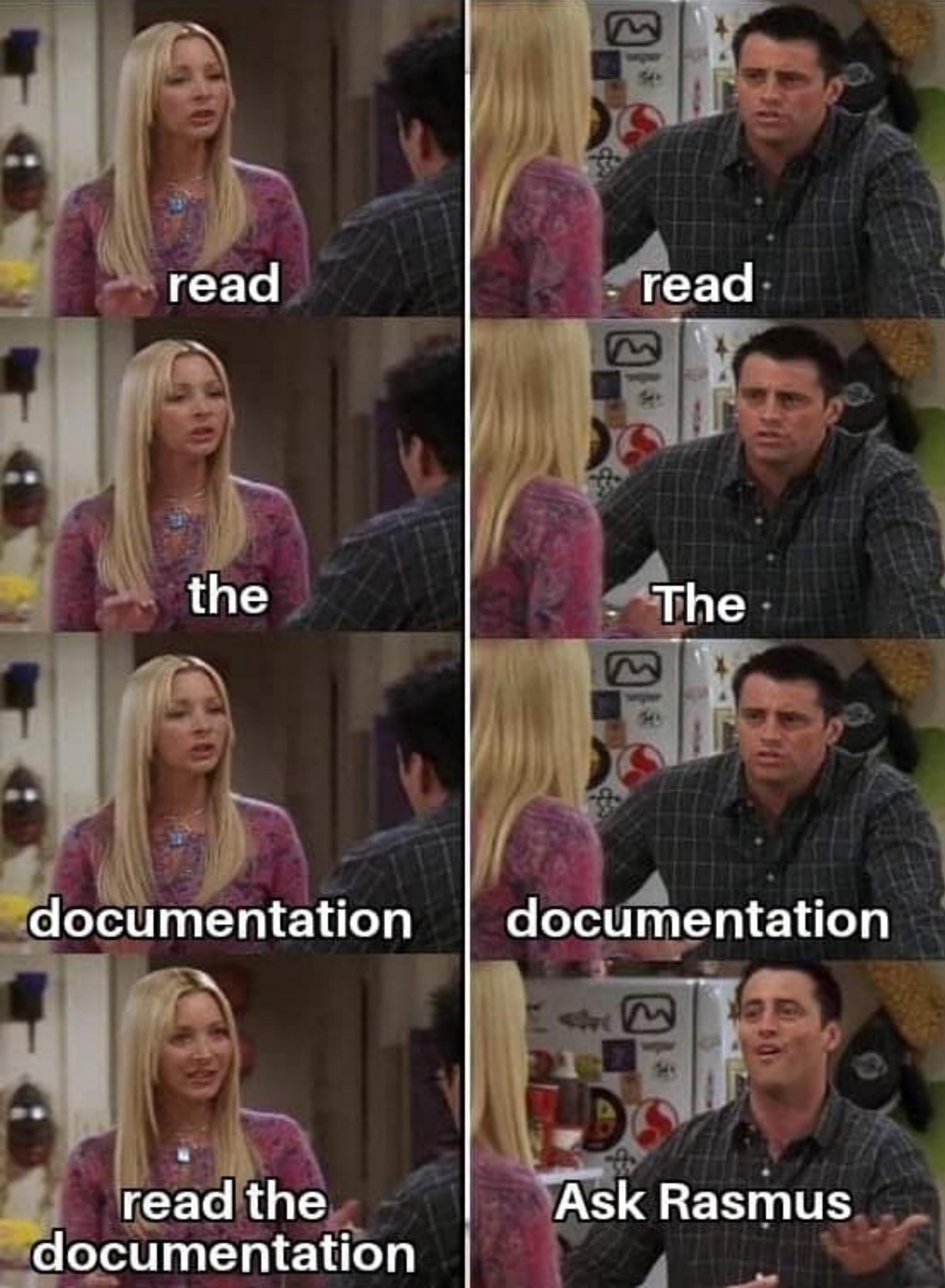
```
SELECT *  
FROM posts,  
     users,  
     posts_users  
WHERE posts.id = posts_users.post_id  
      AND posts_users.user_id = users.id;
```



But, again, JOIN is a better solution



```
SELECT *  
FROM posts  
INNER JOIN posts_users  
    ON posts.id = posts_users.post_id  
INNER JOIN users  
    ON posts_users.user_id = users.id;
```



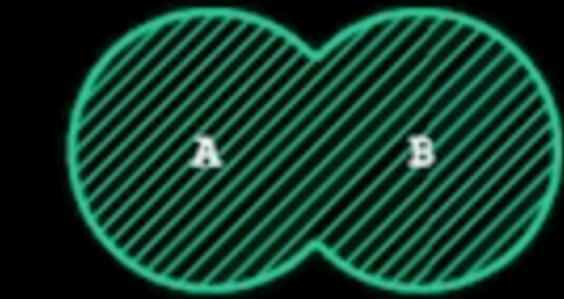
JOINS

- JOINS are SQL statements allowing us to join two (or more) tables in a database on a common column in a particular way.
- Different types of JOINS:
 - (INNER) JOIN
 - LEFT (OUTER) JOIN
 - RIGHT (OUTER) JOIN
 - FULL (OUTER) JOIN

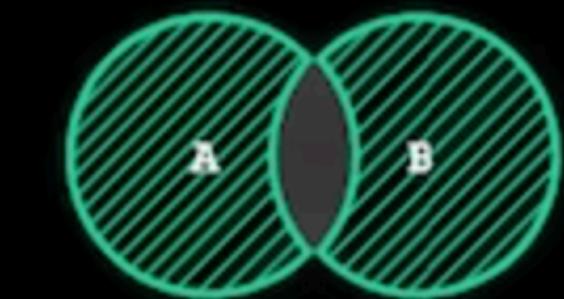
[The Ultimate Guide to All Types of SQL JOINS](#)



```
1 SELECT *\n2 FROM A\n3 INNER JOIN B ON A.key = B.key
```



```
1 SELECT *\n2 FROM A\n3 FULL JOIN B ON A.key = B.key
```



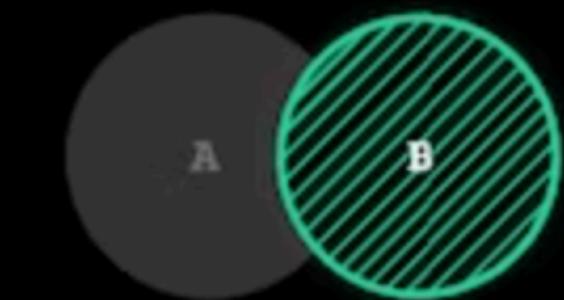
```
1 SELECT *\n2 FROM A\n3 FULL JOIN B ON A.key = B.key\n4 WHERE A.key IS NULL OR\n5 B.key IS NULL
```



```
1 SELECT *\n2 FROM A\n3 LEFT JOIN B ON A.key = B.key
```



```
1 SELECT *\n2 FROM A\n3 LEFT JOIN B ON A.key = B.key\n4 WHERE B.key IS NULL
```



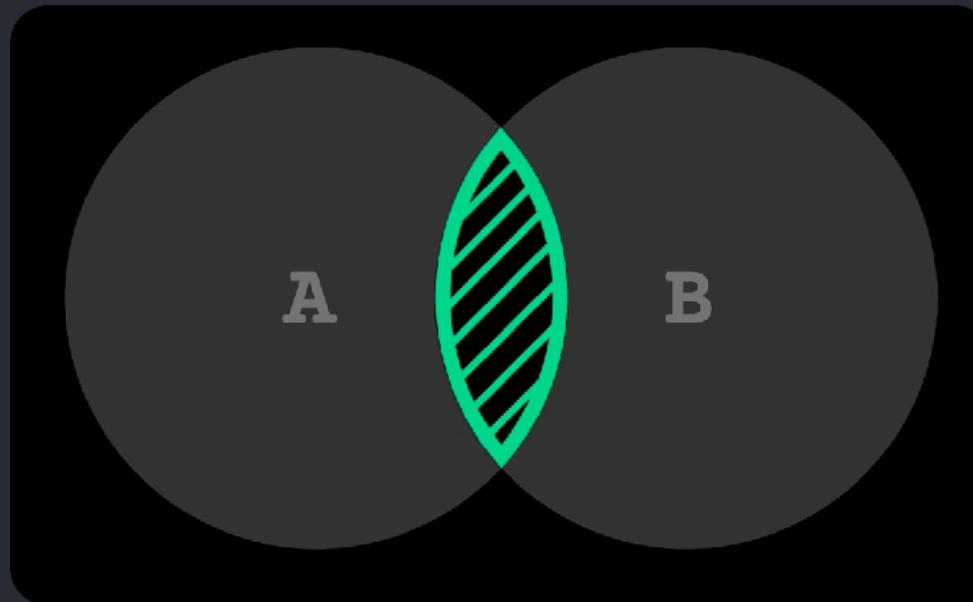
```
1 SELECT *\n2 FROM A\n3 RIGHT JOIN B ON A.key = B.key
```



```
1 SELECT *\n2 FROM A\n3 RIGHT JOIN B ON A.key = B.key\n4 WHERE B.key IS NULL
```

Using (INNER) JOIN

- The INNER JOIN keyword selects records that have a common matching value between them (in both tables).



```
-- Select Posts with Users' Names and Titles
-- (Join Users, Posts, and posts_users Tables)
SELECT *
FROM posts
INNER JOIN posts_users
    ON posts.id = posts_users.post_id
INNER JOIN users
    ON posts_users.user_id = users.id;
```

posts.id	caption	posts.image	createdAt	post_id	user_id	users.id	name
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid...	2023-09-13 19:02:29	2	1	1	Maria Louise Bendixen
1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&i...	2023-09-13 19:02:29	1	2	2	Rasmus Cederdorff
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid...	2023-09-13 19:02:29	2	2	2	Rasmus Cederdorff
1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&i...	2023-09-13 19:02:29	1	4	4	Line Skjødt
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid...	2023-09-13 19:02:29	2	5	5	Dan Okkels Brendstrup
4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&i...	2023-09-13 19:02:29	4	5	5	Dan Okkels Brendstrup

Using JOIN

```
-- Same, but with aliases
SELECT posts.id AS post_id,
       posts.caption,
       posts.image AS post_image,
       users.id AS user_id,
       users.name AS user_name,
       users.mail,
       users.title,
       users.image AS user_image
  FROM posts
 INNER JOIN posts_users
    ON posts.id = posts_users.post_id
 INNER JOIN users
    ON posts_users.user_id = users.id;
```

post_id	caption	post_image	user_id	user_name	mail	title
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wx...	1	Maria Louise Bendixen	mlbe@eaaa.dk	Senior Lecturer
1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M...	2	Rasmus Cederdorff	race@eaaa.dk	Senior Lecturer
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wx...	2	Rasmus Cederdorff	race@eaaa.dk	Senior Lecturer
1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M...	4	Line Skjødt	lskj@eaaa.dk	Senior Lecturer & Internship Co...
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wx...	5	Dan Okkels Brendstrup	dob@eaaa.dk	Lecturer
4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=M...	5	Dan Okkels Brendstrup	dob@eaaa.dk	Lecturer

Using JOIN

```
-- Same, but with other aliases
SELECT posts.*,
       users.name AS user_name,
       users.title AS user_title
  FROM posts
 INNER JOIN posts_users
    ON posts.id = posts_users.post_id
 INNER JOIN users
    ON posts_users.user_id = users.id;
```

□ id	□ caption	□ image	□ createdAt	□ user_name	□ user_title
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wx...	2023-09-13 19:02:29	Maria Louise Bendixen	Senior Lecturer
1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M...	2023-09-13 19:02:29	Rasmus Cederdorff	Senior Lecturer
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wx...	2023-09-13 19:02:29	Rasmus Cederdorff	Senior Lecturer
1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M...	2023-09-13 19:02:29	Line Skjødt	Senior Lecturer & Internship Coordinator
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wx...	2023-09-13 19:02:29	Dan Okkels Brendstrup	Lecturer
4	Exploring the city center of Aarhus	https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=M...	2023-09-13 19:02:29	Dan Okkels Brendstrup	Lecturer

Using JOIN

And specific
user (id)

```
-- Select Posts by a User with Their Name
-- (Replace X with the user's ID)
SELECT posts.*,
       users.name AS user_name
FROM posts
INNER JOIN posts_users
        ON posts.id = posts_users.post_id
INNER JOIN users
        ON posts_users.user_id = users.id
WHERE users.id = 2;
```

□ id	□ caption	□ image	□ createdAt	□ user_name
1	Beautiful sunset at the beach	https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M...	2023-09-13 19:02:29	Rasmus Cederdorff
2	Exploring the city streets of Aarhus	https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wx...	2023-09-13 19:02:29	Rasmus Cederdorff

Functions

- MySQL functions are predefined commands that perform specific operations on data, such as calculations or transformations.
- They enable you to work with data more efficiently and can be used for various tasks, including text manipulation, mathematical operations, and date handling.
- Some return single results (scalar functions), while others work with groups of data (aggregate functions).
- You can also create your own functions if needed.

MySQL MIN() and MAX() Functions

◀ Previous Next ▶

MySQL MIN() and MAX() Functions

The `MIN()` function returns the smallest value of the selected column.

The `MAX()` function returns the largest value of the selected column.

MIN() Syntax

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

MAX() Syntax

```
SELECT MAX(column_name)
```

MySQL COUNT(), AVG() and SUM() Functions

◀ Previous Next ▶

MySQL COUNT(), AVG() and SUM() Functions

The `COUNT()` function returns the number of rows that matches a specified criterion.

COUNT() Syntax

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

The `AVG()` function returns the average value of a numeric column.

AVG() Syntax

Normalization

- Normalization is a database design process that minimizes data redundancy and ensures data integrity.
- It involves organizing data into multiple related tables, reducing duplicate information, and establishing relationships between them. This process typically involves dividing a database into smaller, more manageable tables and using foreign keys to link related data.
- Normalization helps maintain data accuracy, consistency, and efficiency in relational databases.

Normalization

1. **First Normal Form (1NF)**: Make sure each column in your table holds simple, individual pieces of data. No repeated groups of data.
2. **Second Normal Form (2NF)**: Build on 1NF by making sure each column depends on the whole primary key. Break down composite (multi-column) keys.
3. **Third Normal Form (3NF)**: Go a step further by removing indirect dependencies. Data in a table shouldn't rely on things that aren't part of the primary key. This reduces data duplication and keeps things tidy.

Normalization

1. First Normal Form (1NF):

- Ensures that each column in a table contains atomic (indivisible) values.
- Eliminates repeating groups of data.
- Creates a separate table for related data and establishes primary keys.

2. Second Normal Form (2NF):

- Builds upon 1NF by ensuring that non-key attributes are functionally dependent on the entire primary key.
- Removes partial dependencies by breaking down composite keys.
- Ensures that each column is directly related to the primary key.

3. Third Normal Form (3NF):

- Further refines data integrity by eliminating transitive dependencies.
- Data in a table should not depend on non-key attributes.
- Reduces redundancy and enhances data consistency.