

# Databases & SQL

Web Development

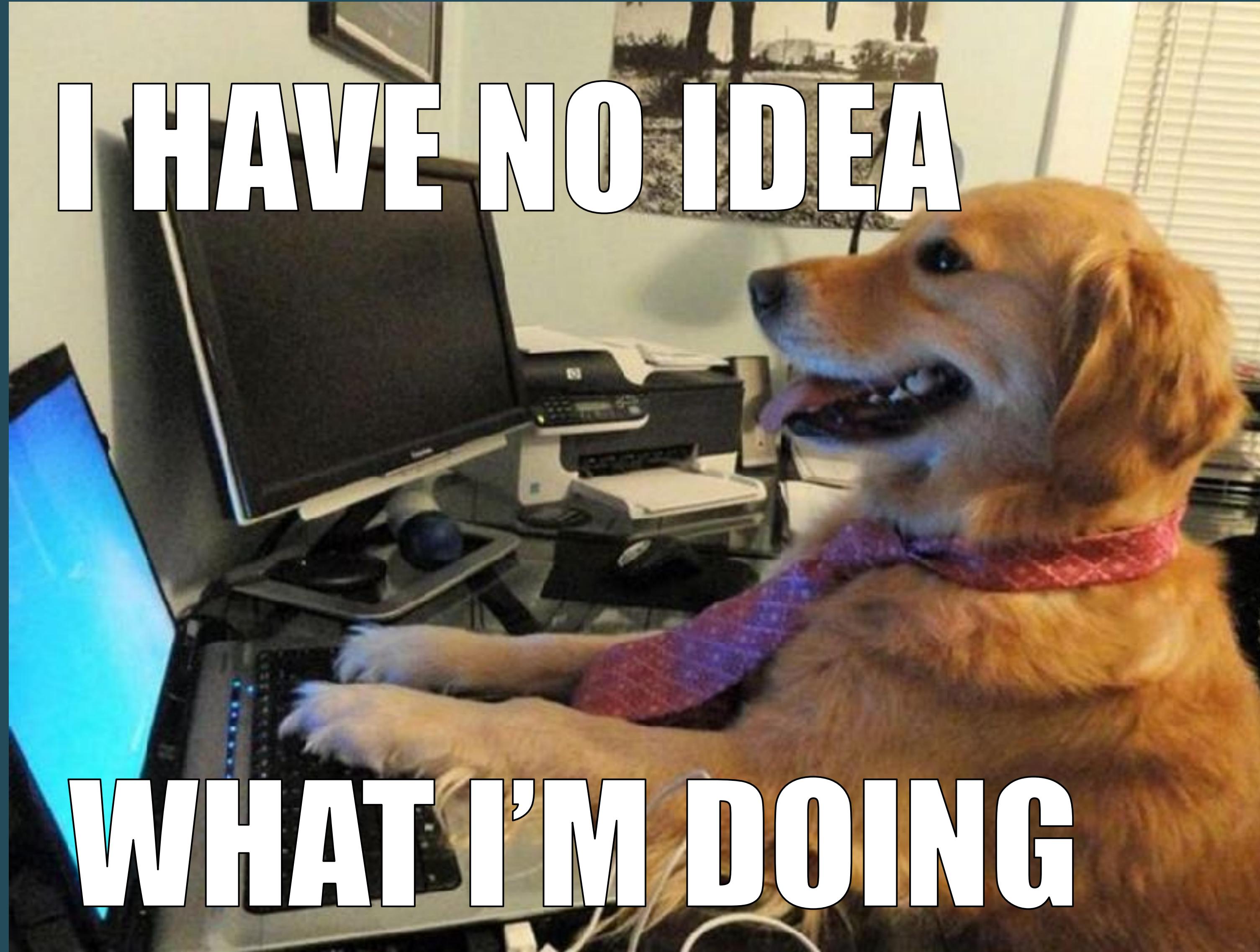
kea  
KØBENHAVNS ERHVERVSAKADEMI

# Turn on the Database



I HAVE NO IDEA

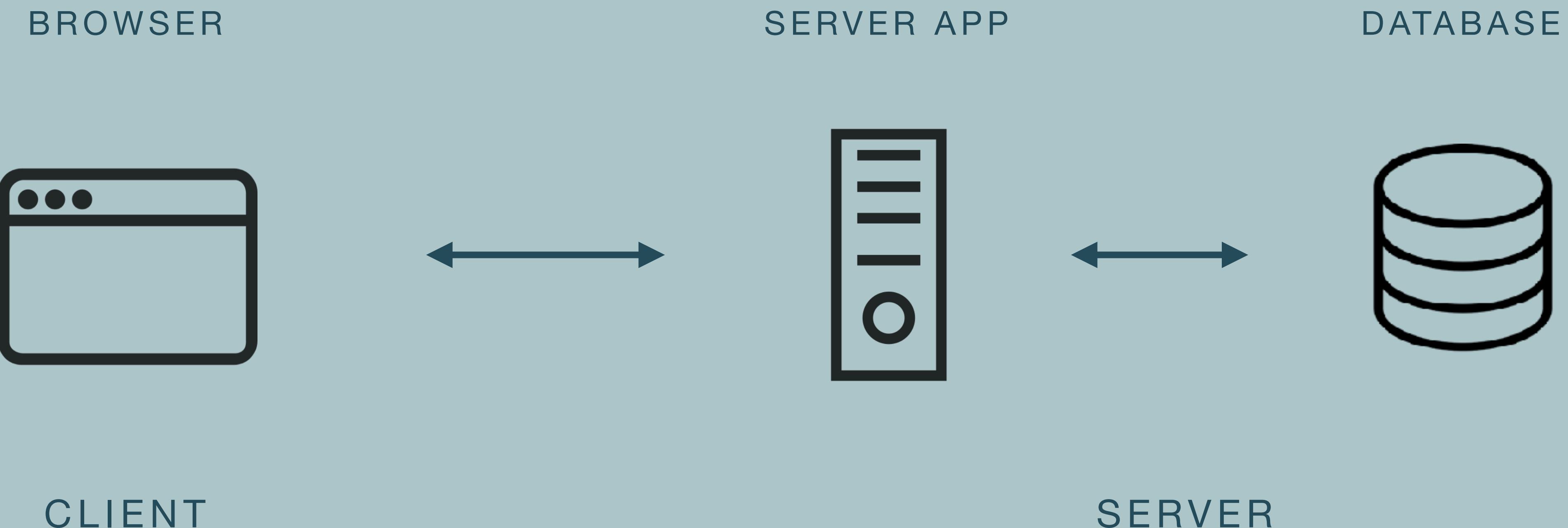
WHAT I'M DOING



# Content

- What is a Data Source?
- What is a Database?
- Relational Databases
- SQL & MySQL
  - SQL
  - MySQL
- Tables & Entities
- SQL statements
  - CREATE database & table
  - Data Types
  - INSERT INTO (create)
  - SELECT (read)
  - UPDATE (update)
  - DELETE (delete)
- SQL Queries
- Keys & IDs
- Entity Relationships

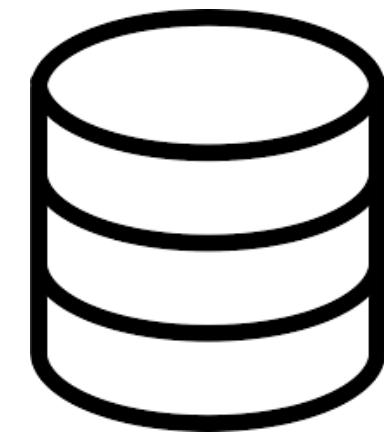
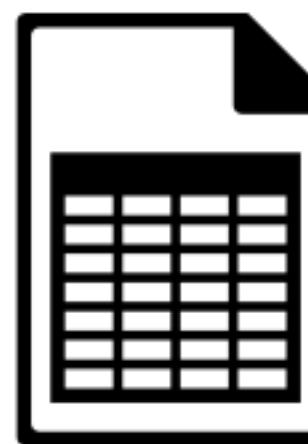
# Web Dev Architecture



# 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



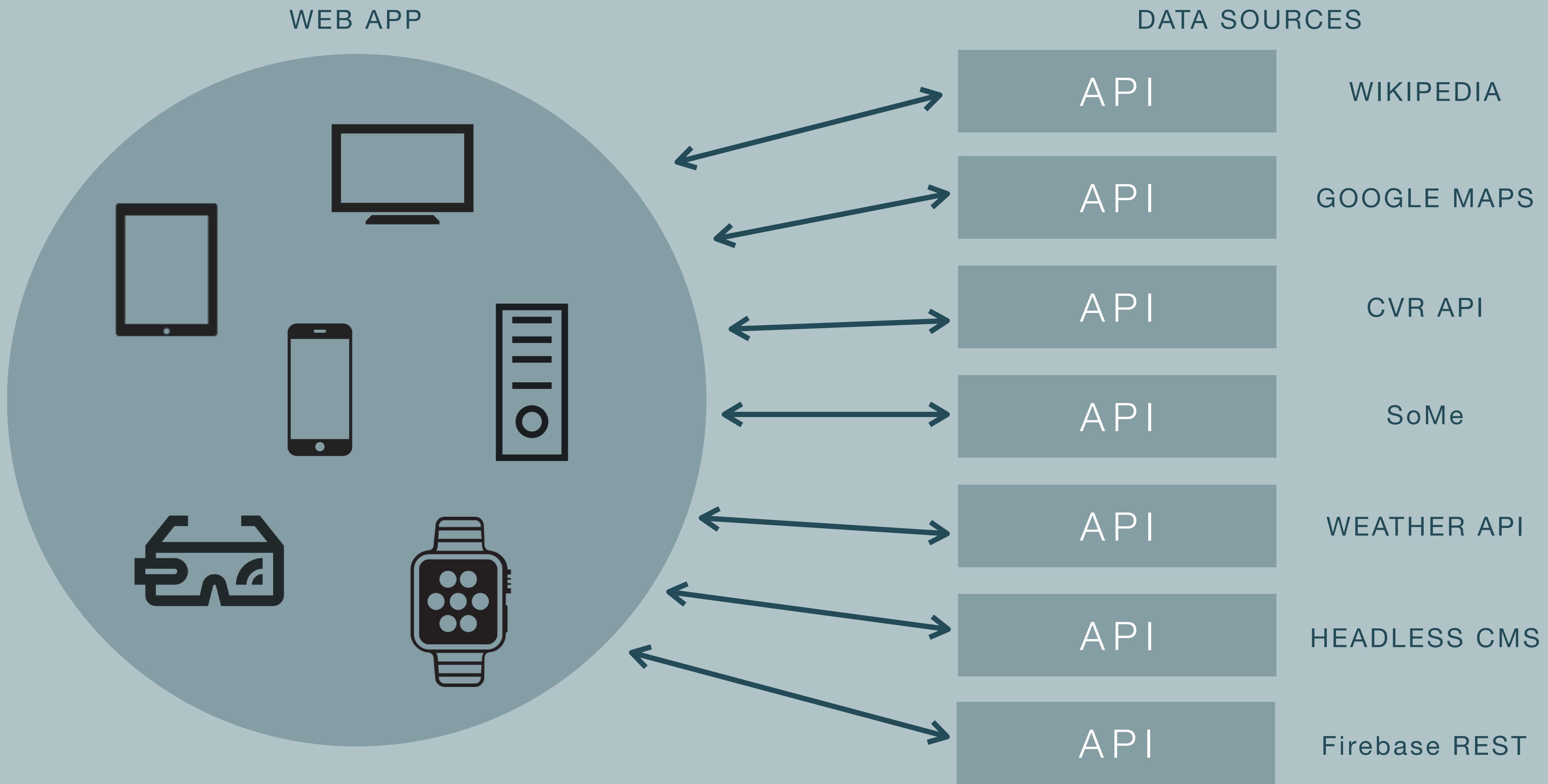
{JSON}

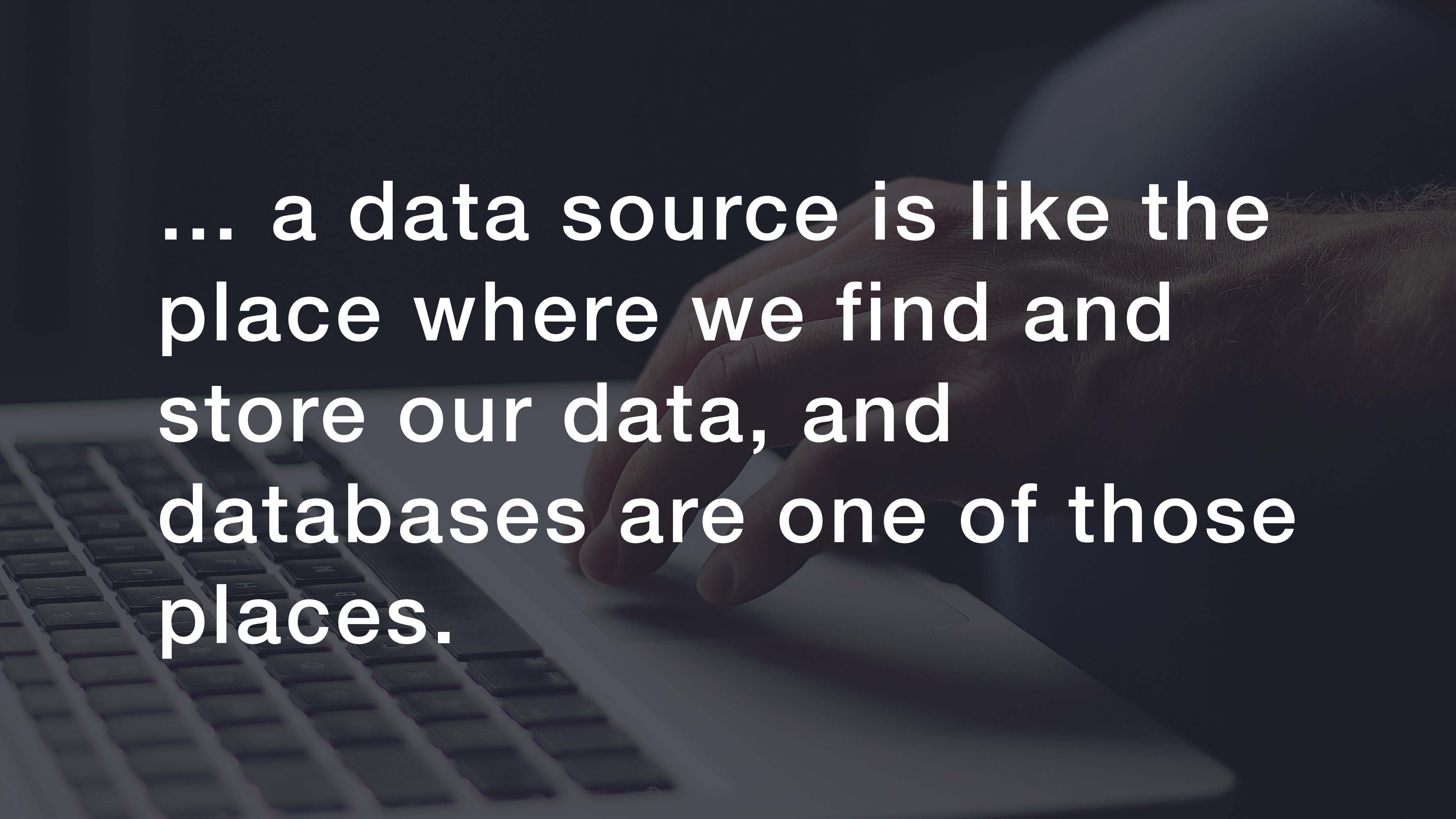
# Data Sources

- 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 dark, moody photograph showing a close-up of a person's hand resting on a laptop keyboard. The hand is positioned with the fingers slightly spread, resting on the keys. The background is dark and out of focus, creating a dramatic lighting effect on the hand and the keyboard.

... 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.

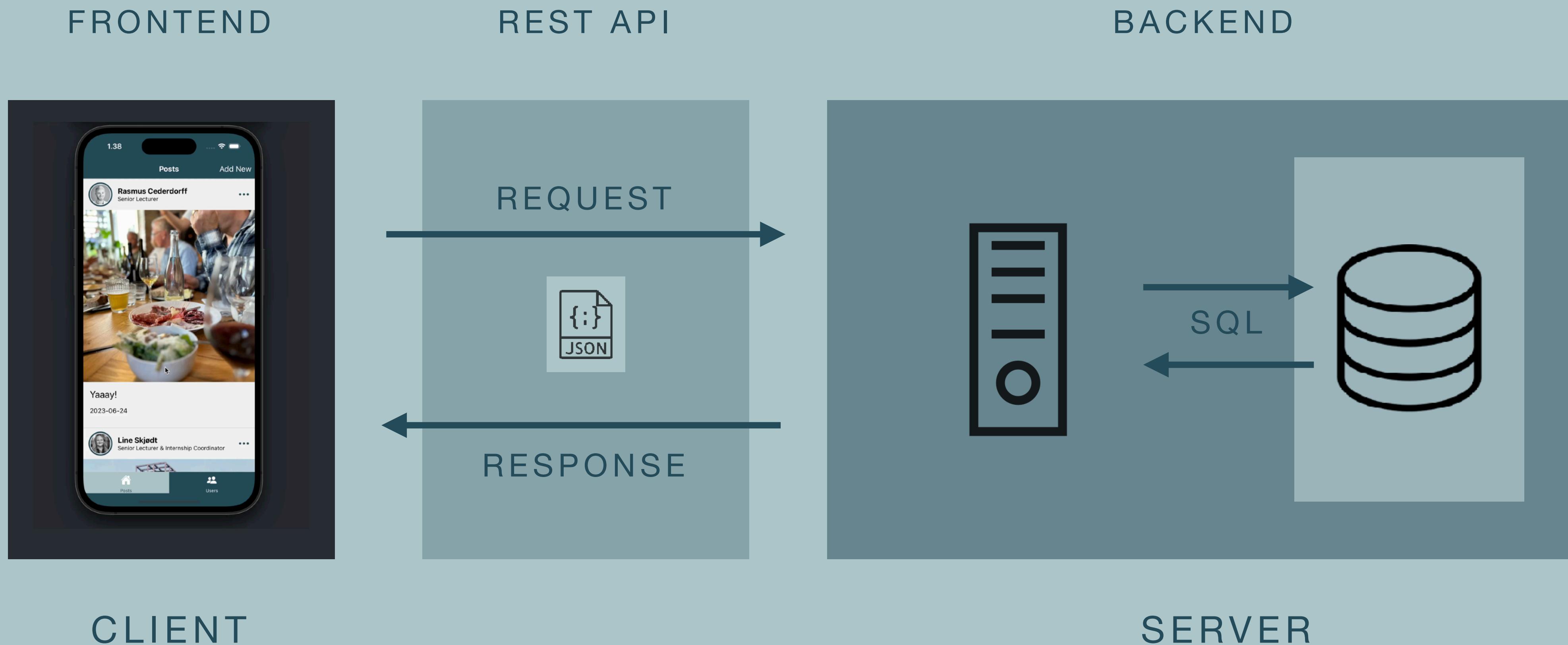
- **Security**

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



# FULLSTACK DEVELOPER



FRONTEND



BACKEND



DATABASE



TOOLS



HTML



PYTHON



CSS



JAVA



JAVASCRIPT



PHP



REACT



RUBY



ANGULAR



MY SQL



MONGODB



POSTGRE SQL



ORACLE



DJANGO



VS CODE



GIT



GITHUB



FIREBASE

# 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.

# 100 *SECONDS OF*

# SQL



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



# Learn SQL In 60 Minutes

[https://www.youtube.com/watch?v=p3qvj9hO\\_Bo](https://www.youtube.com/watch?v=p3qvj9hO_Bo)

# 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	<a href="https://share.cederdorff.com/images/petl.jpg">https://share.cederdorff.com/images/petl.jpg</a>
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	<a href="https://share.cederdorff.com/images/race.jpg">https://share.cederdorff.com/images/race.jpg</a>
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>
4	Edith Terte	edan@kea.dk	Lecturer	<a href="https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g...">https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g...</a>
5	Frederikke Ben... ...nnerup	fbe@kea.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>

# 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	<a href="https://share.cederdorff.com/images/petl.jpg">https://share.cederdorff.com/images/petl.jpg</a>
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	<a href="https://share.cederdorff.com/images/race.jpg">https://share.cederdorff.com/images/race.jpg</a>
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>
4	Edith Terte	edan@kea.dk	Lecturer	<a href="https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g...">https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g...</a>
5	Frederikke Ben... ...n	fbe@kea.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>

# 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	<a href="https://share.cederdorff.com/images/petl.jpg">https://share.cederdorff.com/images/petl.jpg</a>
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	<a href="https://share.cederdorff.com/images/race.jpg">https://share.cederdorff.com/images/race.jpg</a>
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>
4	Edith Terte	edan@kea.dk	Lecturer	<a href="https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g/">https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g/...</a>
5	Frederikke Ben...  ...	fbe@kea.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>

# 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	<a href="https://share.cederdorff.com/images/petl.jpg">https://share.cederdorff.com/images/petl.jpg</a>
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	<a href="https://share.cederdorff.com/images/race.jpg">https://share.cederdorff.com/images/race.jpg</a>
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>
4	Edith Terte	edan@kea.dk	Lecturer	<a href="https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g...">https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g...</a>
5	Frederikke Ben... ...nzen	fbe@kea.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>

# 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	<a href="https://share.cederdorff.com/images/petl.jpg">https://share.cederdorff.com/images/petl.jpg</a>
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	<a href="https://share.cederdorff.com/images/race.jpg">https://share.cederdorff.com/images/race.jpg</a>
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>
4	Edith Terte	edan@kea.dk	Lecturer	<a href="https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g/">https://media.licdn.com/dms/image/C4E03AQE6nx7oUPqo_g/...</a>
5	Frederikke Ben... ...nzen	fbe@kea.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>

# 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	<a href="https://share.cederdorff.com/images/petl.jpg">https://share.cederdorff.com/images/petl.jpg</a>
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	<a href="https://share.cederdorff.com/images/race.jpg">https://share.cederdorff.com/images/race.jpg</a>
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>
4	Edith Terte	edan@kea.dk	Lecturer	<a href="https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g/...">https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g/...</a>
5	Frederikke Ben... ...n	fbe@kea.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>

# 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	<a href="https://share.cederdorff.com/images/petl.jpg">https://share.cederdorff.com/images/petl.jpg</a>
2	Rasmus Cederdorff	race@dev.dk	Senior Lecturer	<a href="https://share.cederdorff.com/images/race.jpg">https://share.cederdorff.com/images/race.jpg</a>
3	Lars Bogetoft	larb@eaaa.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>
4	Edith Terte	edan@kea.dk	Lecturer	<a href="https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g/">https://media.linkedin.com/dms/image/C4E03AQE6nx7oUPqo_g/...</a>
5	Frederikke Ben... ...n	fbe@kea.dk	Head of Education	<a href="https://kea.dk/slir/w200-c1x1/images/user-profile/chef...">https://kea.dk/slir/w200-c1x1/images/user-profile/chef...</a>

# 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

# SQL Statements

```
-- 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 Engineer', 'profile.jpg');

-- 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

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.

In MySQL there are three main data types: string, numeric, and date and time.

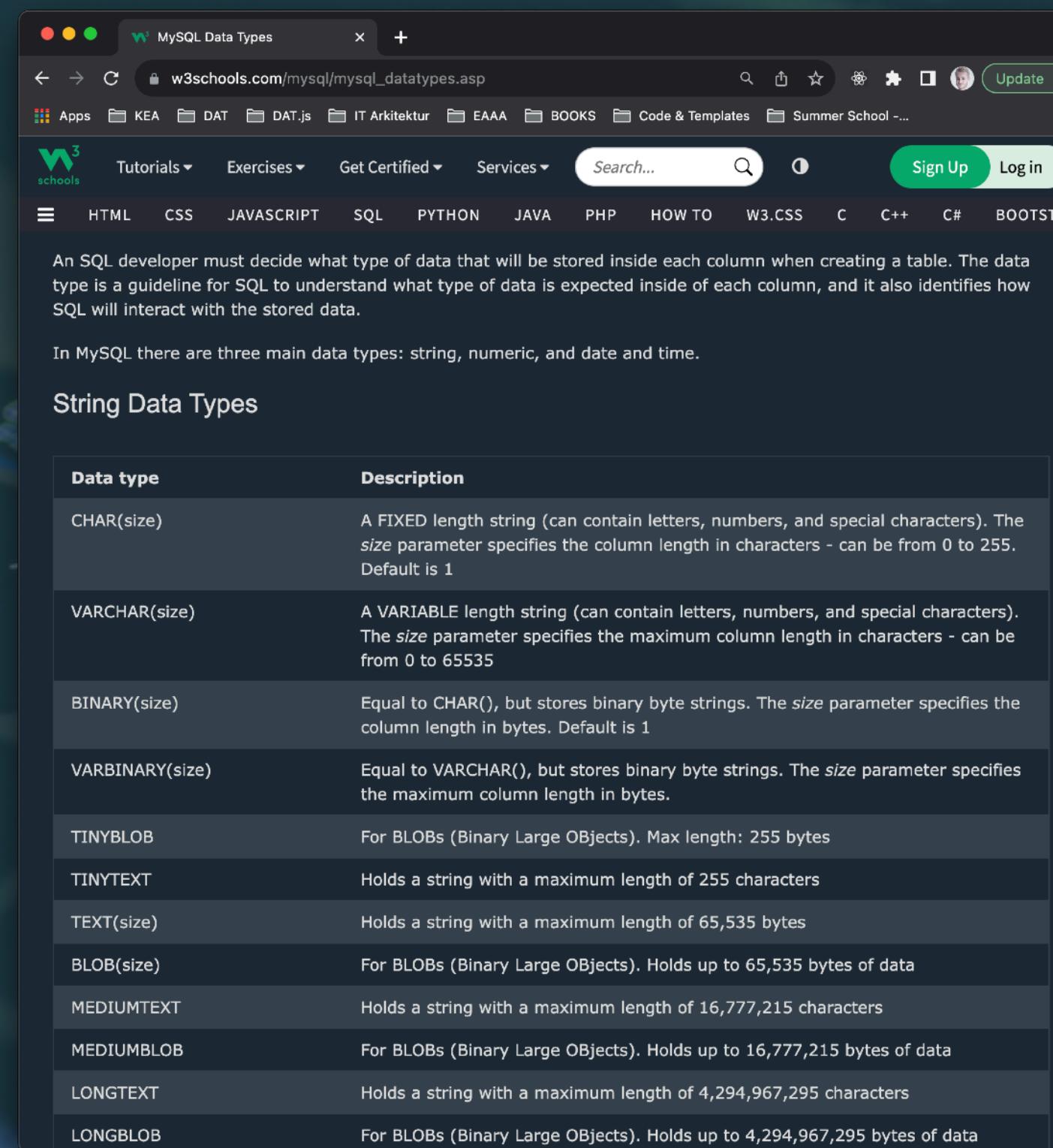
### String Data Types

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 NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(256) NOT NULL,
    mail VARCHAR(256),
    title VARCHAR(256),
    image TEXT
);
```

# MySQL Data Types

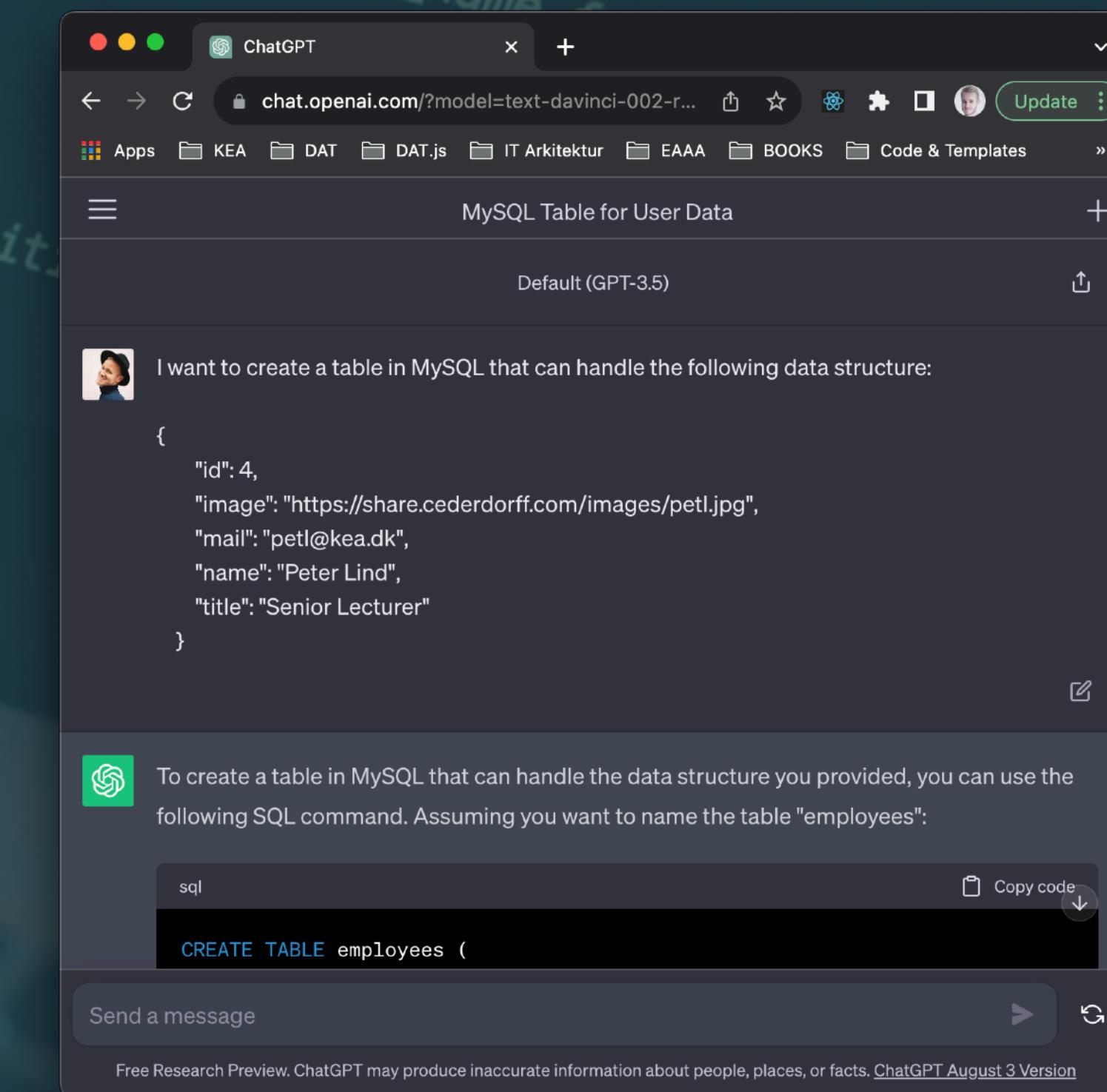
Ask your best friend!



The screenshot shows the MySQL Data Types page from w3schools.com. It features a header with the w3schools logo and navigation links for Tutorials, Exercises, Get Certified, Services, and a search bar. Below the header, a section titled "String Data Types" provides a table comparing various string data types based on their descriptions and characteristics.

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](https://www.w3schools.com/mysql/mysql_datatypes.asp)



The screenshot shows a ChatGPT session. The user asks to create a table in MySQL that can handle a specific data structure. ChatGPT provides the SQL code to create the table based on the provided JSON data.

I want to create a table in MySQL that can handle the following data structure:

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

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');
```

# 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;
```

# 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.

# SQL Queries

## Aggregating Data (GROUP BY, SUM / COUNT)

```
-- 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.

# SQL Queries

## Joining Tables (INNER JOIN)

```
-- join "users" and "departments" tables
SELECT users.name, departments.department_name
FROM users
INNER JOIN departments ON users.department_id = departments.department_id;
```

Selects and joins "users" and "departments" tables based on the "department\_id" column and retrieves user names along with their corresponding department names

# 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.

# 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.

# Keys & IDs

```
CREATE TABLE users (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(256) NOT NULL, [REDACTED]
    mail VARCHAR(256),
    title VARCHAR(256),
    image TEXT
);
```

# Primary Key

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

Unique identifier for each entity (row)

Used to find an entity in a table

Entity must ALWAYS have a Primary Key!

# 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.

# 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.