Databases Lecture 1

Objectives

From this lecture, you should be able to:

- Compare data, information, and knowledge
- Explain why a database is useful
- Explain what Structured Query Language (SQL) is
- Use SQL to retrieve data from a database
  - SELECT using multiple columns, calculated columns (AS), filtered data (WHERE) using multiple conditions, aggregate functions (Count, Sum, Avg, Min, Max) along with GROUP BY.
- Use SQL to manipulate a database
  - INSERT rows into tables
  - CREATE table with a primary key, UPDATE data using calculated columns and WHERE clause.
  - DELETE rows using a WHERE clause

Introduction to Databases

Data vs Information

Data are raw, unprocessed facts. By processing the raw facts (the data), information is created. Data are the raw building blocks for information. Information is created by processing the data and adding context. Information often reveals meaning in the data.

The goal of information is to provide enough context to the data so that meaningful decisions can be made.

Database Management System

A DBMS (DataBase Management System) is an application that allows for the creation of databases and the manipulation of the data therein. The DBMS ensures that the rules and constraints set by the database design is properly enforced and, by doing so, ensuring data integrity.

A DBMS can manage multiple databases, each with their own access rights.

SQLite

Unlike larger DBMS (DataBase Management Systems), SQLite is a small, self-contained, fully-featured SQL database system. An entire database can be contained in a single `*.db` file, which makes it easily copied.

> 🛑 You will mostly come across MySQL in the Laravel course. Keep in mind that there might be small differences.

The Database

A database contains one or more objects called **tables**. Data and information in a database are stored in the tables.

Table

Also sometimes known as a **Relation**. Tables that are related to each other is called a **schema**. Tables are identified by names and are comprised of columns (**field** or **attribute**) and the data in rows (**tuple** or **record**).

Each column must have a datatype (yes, similar to TypeScript), and all data in that column be of that specific data type.

Primary Key

Each table should have one column defined as the **primary key**. The role of the primary key is to uniquely identify a row within the table. A primary key must be **unique** and must **never change.**

Primary keys are often used to retrieve all data related to that specific key or to link multiple tables together.

Relational Notation

Relational notation is a means to describes a table's design. It can describe a table in a single line, as follows:

```
TableName(PrimaryKeyColumn, ColumnName, ColumnName) Employee(EmployeeNumber, Name, Surname, Salary, Address) Song(SongID, SongName, Artist, Album, ReleaseYear, Length, BPM)
```

Starting with the table's name and then all the columns within the table. Primary keys a **bold** and **underlined.**

Entity Relationship Diagram

Tables often have a relationship with other tables (which we will discuss next week), you want to show their relation in an ERD (Entity Relationship Diagram). Tables look like this in an ERD:

TableName	
PK	PrimaryKeyColumn
	ColumnName
	ColumnName

Employee	
PK	EmployeeNumber
	Name
	Surname
	Salary
	Address

We will show how relationships are shown in these diagrams next week.

Introduction to SQL

Structured Query Language (SQL, pronounced *see-kwel*) is used to communicate with a database. It allows databases to be easily manipulated and data retrieved. Its vocabulary consists of fewer than 100 words.

SQL Tutorial

SQL is a standard language for storing, manipulating and retrieving data in databases. Our SQL tutorial will teach you how

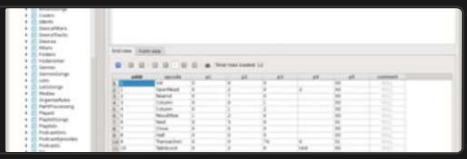
https://www.w3schools.com/sql/default.asp

SQLiteStudio

For this series of lectures, we will use SQLiteStudio. You are free to use any other SQLite tool, such as SQLiteBrowser or phpLiteAdmin.

SQLiteStudio

https://sqlitestudio.pl/



Data Types in SQLite

It is very important for databases to know exactly what type of data will be stored to determine how much space to reserve it. mySQL has many different data types. Luckily, SQLite only has the following:

- **NULL**. The value is NULL.
- **INTEGER**. The value is any signed integer. Can take 0-8 bytes depending on the size.
- **REAL**. A floating point number. 8-bytes.
- **TEXT**. Text using the encoding selected on the database (UTF-8, or UTF-16LE/BE)
- **BLOB**. Raw data.

Boolean?

Booleans are stored as an INTEGER. 0 (false) and 1 (true).

DATE/TIME

There are three ways to store a DATE/TIME in SQLite

- **TEXT** stored in ISO8601 (YYYY-MM-DD HH:MM:SS.SSS) format.
- **INTEGER** storing the Unix Time.
- **REAL** storing Julian day numbers.

🛑 When working on your Laravel project, you will be using many more different data types. Read about them here

Creating a Table

To create a new table, you use the following command:

```
CREATE TABLE tablename ( column1 datatype [constraint], column2 datatype [constraint], column3 datatype [constraint], [PRIMARY KEY (column1)] );
```

- Start with `CREATE TABLE`
- Give the table a name. It has to be unique within the schema
- List each column to be created. Each column must have
  - a name, which is unique to the table
  - a data type
  - an *optional* constraint
    - UNIQUE: no other row may have the same value
    - NOT NULL: a value is mandatory
- Table constraints
  - The primary key for the table
    - Start with `PRIMARY KEY` and then use the column name that will be the primary key
  - Any foreign keys for the table (more on this later in the series).
    - Start with `FOREIGN KEY` and then select the column that will be used as the foreign key, and which table it references.

An example table will be:

```
CREATE TABLE Employees ( EmployeeNumber TEXT NOT NULL, Surname TEXT, Name TEXT, Salary REAL, Address TEXT, PRIMARY KEY(EmployeeNumber) ); CREATE TABLE Songs ( SongID INTEGER, SongName TEXT, Artist TEXT, Album TEXT, ReleaseYear INTEGER, Length INTEGER, BPM INTEGER, PRIMARY KEY(SongID) );
```

Inserting Data

To insert data use the following command:

```
INSERT INTO tablename (column1, column2) VALUES (value1, value2);
```

- Start with `INSERT INTO` and select the table that you want to insert data into
- Within brackets, list the columns that you will be entering data into. It doesn't have to be all the columns in the table, but it must be columns that do exist in the table.
- Then `VALUES` followed by the values in the order that you selected in the columns.

An example insert will be:

```
INSERT INTO Employees (EmployeeNumber, Surname, Name, Salary) VALUES ("15", "Jansen", "Jan", 15938.22);
```

Retrieving Data

This is where the real power of SQL now starts. Retrieving data is done using the `SELECT` query.

```
SELECT [column names, *] FROM tablename
```

- Start with `SELECT` and select the column names you want to retrieve. You can use `*` for all columns.
- Add `FROM` and select the table where you want to get the data from.

For example:

```
SELECT * FROM Employees; SELECT EmployeeNumber, Name FROM Employees
```

SQL SELECT Statement

The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.

https://www.w3schools.com/sql/sql_select.asp

Filtering

You can add conditions using the `WHERE` clause. For example

```
SELECT * FROM Employees WHERE Salary > 5000; SELECT EmployeeNumber, Name F
ROM Employees WHERE Salary <= 5000;
```

SQL WHERE Clause

The WHERE clause is used to filter records. It is used to extract only those records that fulfill a specified condition. Note: The

https://www.w3schools.com/sql/sql_where.asp

In there `WHERE` clause you can add multiple conditions using `AND`, `OR`, or `NOT` operators.

```
SELECT * FROM Employees WHERE Salary <= 5000 AND Salary > 10000;
```

SQL AND, OR, NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators. The AND and OR operators are used to filter records

https://www.w3schools.com/sql/sql_and_or.asp

Or just using a range using `BETWEEN.`

```
SELECT * from Employees WHERE Salary BETWEEN 5000 AND 10000;
```

SQL BETWEEN Operator

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates. The BETWEEN operator is

https://www.w3schools.com/sql/sql_between.asp

Selecting from a list is also possible using `IN`

SQL IN Operator

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects

https://www.w3schools.com/sql/sql_in.asp

Using wildcards are also possible using the `LIKE` operator.

```
SELECT * FROM Employees WHERE Surname LIKE 'Smith%'
```

SQL Wildcard Characters

A wildcard character is used to substitute one or more characters in a string. Wildcard characters are used with the operator. The

https://www.w3schools.com/sql/sql_wildcards.asp

Ordering

You can sort your data using the `ORDER BY` clause

```
SELECT * FROM Employees ORDER BY Salary; SELECT Name, Surname FROM Employe
es ORDER BY Surname; SELECT * FROM Employees WHERE Salary > 5000 ORDER BY
Salary DESC;
```

`DESC` means descending or reverse order.

SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set in ascending or descending order. The ORDER BY keyword sorts

https://www.w3schools.com/sql/sql_orderby.asp

Limiting Output

You can limit the number of rows returned as well by using `LIMIT`

```
SELECT * from Employees ORDER BY Salary LIMIT 3;
```

MySQL LIMIT

The LIMIT clause is used to specify the number of records to return. The LIMIT clause is useful on large tables with thousands

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

Calculated Columns

Instead of selecting data directly from the table, you can calculate a new value from existing data in the table.

```
SELECT (column1*column2) AS MyNewColumn from TABLE
```

SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name. Aliases are often used to make column names

https://www.w3schools.com/sql/sql_alias.asp

Aggregating

You can also get the largest, smallest, average, count and sum of columns from a table.

www.w3schools.com

https://www.w3schools.com/sql/sql_min_max.asp

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

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

https://www.w3schools.com/sql/sql_count_avg_sum.asp

SQL GROUP BY Statement

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each

https://www.w3schools.com/sql/sql_groupby.asp

Updating

You can update data using:

```
UPDATE tablename SET column1=newvalue, column2="another value" [WHERE cond
itions]
```

Start with the `UPDATE` and select the table you want to update. Set the new values. You may use a `WHERE` clause, but if you don't, ALL rows will be updated with the new value 🙂

SQL UPDATE Statement

The UPDATE statement is used to modify the existing records in a table. Note: Be careful when updating records in a table!

https://www.w3schools.com/sql/sql_update.asp

Deleting

And, finally, to delete data:

```
DELETE FROM tablename [WHERE conditions]
```

Start with `DELETE FROM` and select the table you wish to delete data from. It is highly suggested you use a `WHERE` clause, otherwise everything will be deleted.

SQL DELETE Statement

The DELETE statement is used to delete existing records in a table. Note: Be careful when deleting records in a table! Notice

https://www.w3schools.com/sql/sql_delete.asp

🥎 Week 1 Exercises

Supplementary Tutorial

If none of this made sense to you, here's another Tutorial