SQL

Learning Objectives

- Understand what SQL is
- Understand how to write basic SQL queries

Material

What is SQL?

Once we have created our relational database, we need to read and write values to it. To do this, we use Structured Query Language (SQL). SQL is a language designed for handling relational databases. Relational Database Management Systems (RDBMS) which use SQL include: PostgreSQL, MySQL and SQLite.

SQL Queries

Let's look at some example queries using SQLite. Here is a table of tech companies:

```
companies
| id | name   | location      | numEmployees |
|----|--------|---------------|--------------|
| 1  | Google | Mountain View | 135000       |
| 2  | Apple  | Cupertino     | 147000       |
```

Each column in an SQL table has a specific data type. The data types for different SQL databases vary slightly; for SQLite they are as follows:

Data Type	**Description**
Null	Null value
Integer	Integer value
Real	Floating-point value
Text	String value
Blob	Blob of data (e.g. file)

The SQL command to create the COMPANIES table would be:

```
CREATE TABLE companies(id INTEGER PRIMARY KEY,
                       name TEXT,
                       location TEXT,
                       numEmployees INTEGER);
```

It is conventional (though not strictly necessary) to put SQL keywords in all-caps.

Inserting Rows

Create new rows in the database with the following;

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

For example:

```
INSERT INTO companies (name, location, numEmployees)
VALUES ("Amazon", "Seattle", 1298000);
```

Retrieving Rows

To read values out of the database you can use the following:

```
SELECT column1, column2, ...
FROM table_name;
```

For example:

```
SELECT name, numEmployees
FROM companies;
```

This statement can be extended with the `WHERE` keyword to be more selective in what is read from the database. The `WHERE` clause can filter records for you. An example is below.

```
SELECT * FROM companies
WHERE numEmployees > 100000;
```

The asterisk (*) is used to get *all* the columns. The `WHERE` clause identifies the column numEmployees and the condition is to only return rows where the value for this field is greater than 100000.

Updating a Table

The UPDATE keyword is used to update tables:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

For example, when Google grows, we might want to run:

```
UPDATE companies SET numEmployees=136000 WHERE name="Google";
```

Deleting Rows and Tables

We can delete rows like so:

```
DELETE FROM table_name WHERE condition;
```

This would delete all the companies whose location field has the value of "California".

```
DELETE FROM companies WHERE location="California";
```

You can also delete entire tables;

```
DROP TABLE companies;
```

Or even entire databases:

```
DROP DATABASE main;
```

Core Assignments

Restaurant Tables

You're going to create a relational database to model a series of restaurant chains. You should have tables for **companies**, **locations** and **menus**. Locations and menus both belong to a particular company. For example, *McDonalds* is a company with a *Drinks* menu and locations in *Cheltenham* and *Cirencester*. You will need to represent these relationships using the appropriate primary and foreign keys. Along with these keys:

Companies should also have:

- name: string
- logoUrl: string

Menus should also have:

- title: string

Locations should also have:

- name: string
- capacity: integer
- manager: string

Using a SQLite Repl, compose the SQL queries to do the following:

- Create the companies, menus and locations tables
- Insert rows into each table using "INSERT"
- Read back your row data using the SELECT command
- Update your rows using the UPDATE command
- Delete your rows using DELETE command
- Use the WHERE clause to filter results

You should also be aware of some SQLite-specific SQL features which will help you. The PRAGMA command describes a table by listing its columns and indicating which column is the primary key.

Put this at the top of your file to make the `SELECT` output more readable:

```
.mode column
.headers on
```

```
PRAGMA table_info(companies);
```

The following command lists all the tables in the database:

```
SELECT name FROM sqlite_master WHERE type='table'
```

Additional Resources

- W3schools SQL tutorial