Introduction to SQL

Qitian Liao

University of California, Berkeley

Contents

1 Manipulation

Get up and running with SQL by learning commands to manipulate data stored in relational databases.

1.1 Introduction to SQL

SQL, **S**tructured **Q**uery **L**anguage, is a programming language designed to manage data stored in relational databases. SQL operates through simple, declarative statements. This keeps data accurate and secure, and helps maintain the integrity of databases, regardless of size.

The SQL language is widely used today across web frameworks and database applications. Knowing SQL gives you the freedom to explore your data, and the power to make better decisions. By learning SQL, you will also learn concepts that apply to nearly every data storage system.

The statements covered in this course use SQLite Relational Database Management System (RDBMS). You can also access a glossary of all the SQL commands taught in this chapter.

1.2 Relational Databases

In one line of code, we can return information from a relational database.

```
SELECT * FROM celebs;
```

A *relational database* is a database that organizes information into one or more tables. Here, the relational database contains one table.

A *table* is a collection of data organized into rows and columns. Tables are sometimes referred to as *relations*. Here the table is `celebs`.

A *column* is a set of data values of a particular type. Here, `id`, `name`, and `age` are the columns.

A *row* is a single record in a table.

All data stored in a relational database is of a certain data type. Some of the most common data types are:

- `INTEGER`, a positive or negative whole number
- `TEXT`, a text string
- `DATE`, the date formatted as YYYY-MM-DD
- `REAL`, a decimal value

1.3 Statements

The code below is a SQL statement. A *statement* is text that the database recognizes as a valid command. Statements always end in a semicolon `;`.

```
CREATE TABLE table_name (
   column_1 data_type,
   column_2 data_type,
   column_3 data_type
);
```

Let us break down the components of a statement:

1. `CREATE TABLE` is a *clause*. Clauses perform specific tasks in SQL. By convention, clauses are written in capital letters. Clauses can also be referred to as commands.
2. `table_name` refers to the name of the table that the command is applied to.
3. `(column_1 data_type, column_2 data_type, column_3 data_type)` is a *parameter*. A parameter is a list of columns, data types, or values that are passed to a clause as an argument. Here, the parameter is a list of column names and the associated data type.

The structure of SQL statements vary. The number of lines used does not matter. A statement can be written all on one line, or split up across multiple lines if it makes it easier to read.

1.4 Create

`CREATE` statements allow us to create a new table in the database. You can use the `CREATE` statement anytime you want to create a new table from scratch. The statement below creates a new table named celebs.

```
1 CREATE TABLE celebs (
2    id INTEGER,
3    name TEXT,
4    age INTEGER
5 );
```

1. `CREATE TABLE` is a clause that tells SQL you want to create a new table.
2. `celebs` is the name of the table.
3. `(id INTEGER, name TEXT, age INTEGER)` is a list of parameters defining each column, or attribute in the table and its data type:
   - `id` is the first column in the table. It stores values of data type `INTEGER`
   - `name` is the second column in the table. It stores values of data type `TEXT`
   - `age` is the third column in the table. It stores values of data type `INTEGER`

1.5 Insert

The `INSERT` statement inserts a new row into a table. You can use the `INSERT` statement when you want to add new records. The statement below enters a record for Justin Bieber into the celebs table.

```
1 INSERT INTO celebs (id, name, age)
2 VALUES (1, "Justin Bieber", 22);
```

1. `INSERT INTO` is a clause that adds the specified row or rows.
2. `celebs` is the name of the table the row is added to.
3. `(id, name, age)` is a parameter identifying the columns that data will be inserted into.
4. `VALUES` is a clause that indicates the data being inserted. `(1, "Justin Bieber", 22)` is a parameter identifying the values being inserted.

- `1` is an integer that will be inserted into the `id` column
- `"Justin Bieber"` is text that will be inserted into the `name` column
- `22` is an integer that will be inserted into the `age` column

1.6 Select

`SELECT` statements are used to fetch data from a database. In the statement below, `SELECT` returns all data in the name column of the celebs table.

```
1 SELECT name FROM celebs;
```

1. `SELECT` is a clause that indicates that the statement is a query. You will use `SELECT` every time you query data from a database.
2. `name` specifies the column to query data from.
3. `FROM celebs` specifies the name of the table to query data from. In this statement, data is queried from the `celebs` table.

You can also query data from all columns in a table with `SELECT`.

```
1 SELECT * FROM celebs;
```

`*` is a special wildcard character that we have been using. It allows you to select every column in a table without having to name each one individually. Here, the result set contains every column in the `celebs` table.

`SELECT` statements always return a new table called the *result set*.