

SQL: structured query language

SQL (Structured Query Language) is a **domain-specific programming language** designed to handle data in tables.

It was developed in the 1970s. To this day, SQL-like interfaces are very popular in various data management systems, not only the ones based on tables!

If you are a software engineer, it's good to learn it because many systems store and process business data via services that support SQL.

For example, the backend of an insurance company's information system may use SQL to extract and update data about their clients.

Calculating statistics

SQL makes aggregating data and calculating statistics easier. Suppose you need to evaluate changes in the popularity of the name Jessie between 1920 and 2000 (inclusive) based on census data.

It selects records about individuals named 'Jessie' who were born BETWEEN 1920 and 2000. It groups them by year and gender, counts the number of records in each group via COUNT(*), and generates a table with columns named year, gender, and cnt. It also sorts the table by year and gender in descending order.

```
SELECT
  year, gender, COUNT(*) as cnt
FROM
  census
WHERE
  year BETWEEN 1920 and 2000
  AND name = 'Jessie'
GROUP BY
  year, gender
ORDER BY
  year, gender DESC
```

SQL is the standard **data manipulation language** used by data-driven companies around the world.

S is for Structured

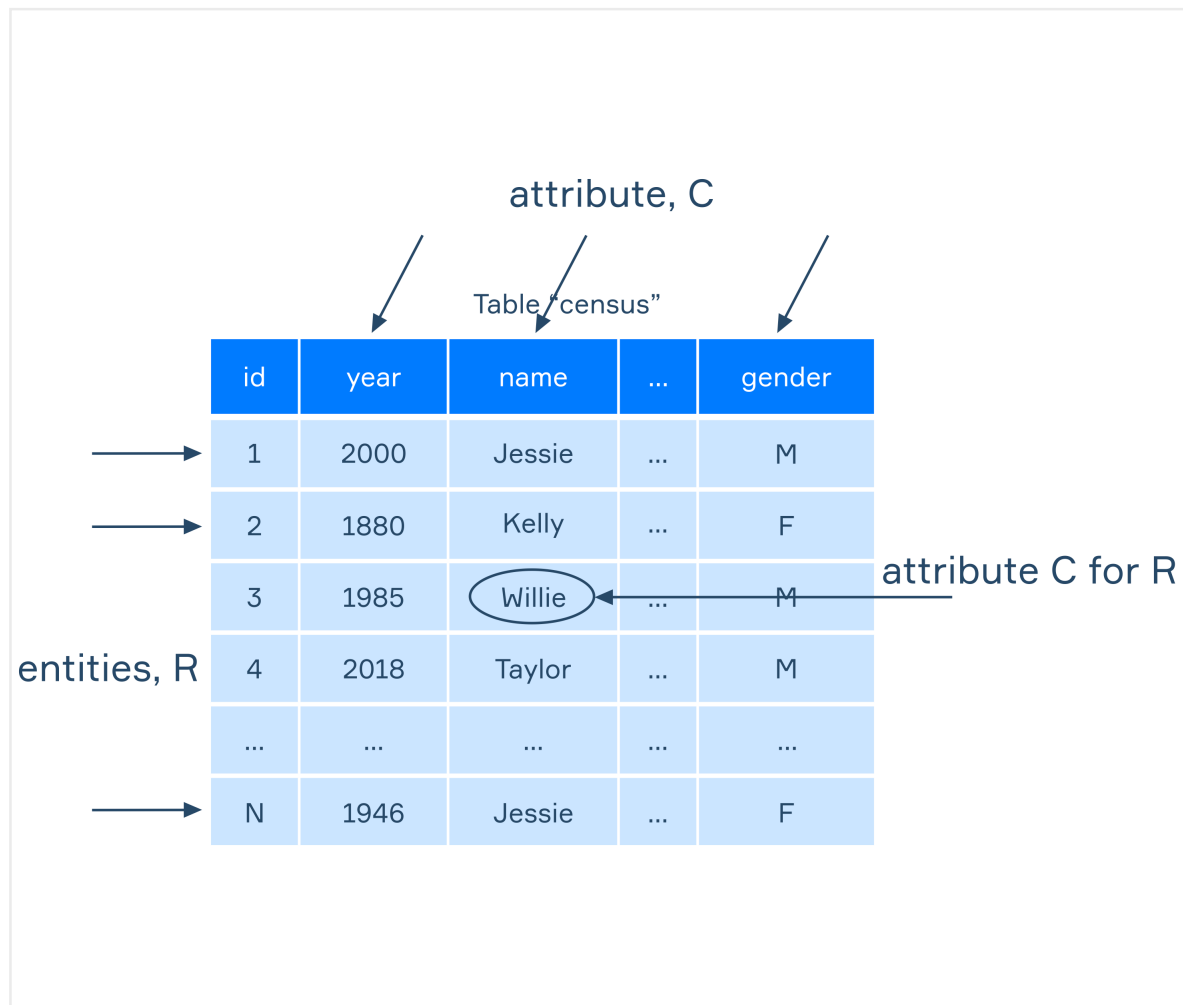
SQL is a language used to extract and update data **structured** as tables.

This kind of data appears in various application areas, such as Excel

spreadsheets containing accounting data, or census statistics in Google BigQuery.

Another example is an online store that utilizes a special software system to store and access tables, known as a **Relational Database Management System (RDBMS)**. These can help to process the information on goods, orders, and customers.

SQL is intended for use with tables that have a particular structure:



The tables contain **rows** and **columns**. Each row is an object or entity that has a set of properties or attributes. For instance, the third row contains data about Willie, a man born in 1985.

Data is often organized into a set of tables, known as a **database**. Then it's possible to access these tables using their individual names.

For example, in an online store's database, a table called Customers would probably contain general information about the company's customers: their names and contact details.

The Orders table would store information about the specific orders they place: customer names, goods, and payment details.

Q is for Query

SQL is a programming language with a large set of data processing features. It is **declarative**, meaning that a statement written in SQL can be a **query** that tells the system what should be done or evaluated but doesn't specify how.

In the next example, a query extracts all rows and columns from the table Census:

```
SELECT * FROM Census;
```

The * symbol is used to select all the columns from the table.

It's necessary to end each SQL statement with a semicolon, also known as a **statement terminator**.

Else, there will be an error. A semicolon may be omitted only if you make one query at a time, but it is a good practice to always put it at the end of your query.

Keywords such as SELECT are not case sensitive in the SQL language. They can be in any letter case but are often written in all caps to make them more visible.

L is for Language

You can read the simple query in the above example as "select everything from the census." SQL was designed to be as similar as possible to a natural **language**.

Its declarative nature helps to hide the operation's complexities, letting the user define what is required in a relatively straightforward way.

The system then analyzes the query, chooses the control flow, and executes it. SQL was originally adopted as a standard by the **American National Standards Institute (ANSI)** in the 1980s.

There are many dialects implemented by software vendors that support it. Dialects are based on the ANSI standard but have some technical differences. For example, they might process dates or strings differently. This means that SQL queries written in different dialects are not compatible. However, once you know the SQL basics, it's possible to adapt to dialects fairly easily, like with written American and British English.

MySQL syntax has been used in this topic and elsewhere on this website, so the examples may not be compatible with other SQL dialects.

