

4.5: The Difference between a Database and a Spreadsheet

When introducing the concept of databases to students, they quickly decide that a database is similar to a spreadsheet. There are some similarities, but we will review some big differences. A spreadsheet hopes to grow into a database one day.

Let's start with the spreadsheet. It is easy to create, edit, and format. It is simple to use for beginners. It comprises columns and rows and stores data in an organized fashion, like a database table. The two leading spreadsheet applications are Google Sheets and Microsoft Excel. One of the very convenient things about spreadsheets is shared accessibility with multiple users. This is not the case with a database.

For simple uses, a spreadsheet can substitute for a database quite well. Creating a database is probably overkill if a simple listing of rows and columns (a single table) is all that is needed. In our Student Clubs example, we could get away with a single spreadsheet if we only needed to track a listing of clubs, the number of members, and the president's contact information. However, the need to include a listing of events and members' names would be problematic if tracked with a spreadsheet.

When several types of data must be mixed, or when the relationships between these types of data are complex, then a spreadsheet is not the best solution. A database allows data from several entities (such as students, clubs, memberships, and events) to be related together into one whole. While a spreadsheet does allow you to define what kinds of values can be entered into its cells, a database provides more intuitive and powerful ways to define the types of data that go into each field, reducing possible errors and allowing for easier analysis. Though not good for replacing databases, spreadsheets can be ideal tools for analyzing the data stored in a database. A spreadsheet package can be connected to a specific table or query in a database and used to create charts or analyze that data.

A database has many similarities in looks of a spreadsheet utilizing tables that are made up of columns and rows. The database is a collection of structured raw material. The information is stored on the computer. A spreadsheet is easily editable with its rows and columns; this is not the case of a database. The database is formatted, so the field (column) is pre-configured. The database is also relational in that it has the ability to create relationships between records and tables. Spreadsheets and databases can both be edited by multiple authors. In a database, a log is created as changes are made. This is not the case with a spreadsheet. A spreadsheet is terrific for small projects, but a database would become more useful as the project grows.

✓ Example: Why use Databases? 4.5.1

A company uses a spreadsheet to store every purchase amount from a customer. The spreadsheet would look like this:

Clients Purchases

Date	Client Name	Amount
1/1/2023	Jane Doe	\$100
1/5/2023	John Smith	\$50
1/7/2023	Jane Doe	\$25
1/10/2023	Bob Jones	\$200

Each transaction is entered on a separate row, with the client name typed out fully each time. Jane made two purchases on two different days; hence there are two rows for her.

Problem: If Jane Doe changed her name to Jane Johnson, her name would need to be updated in multiple rows. Imagine if you have a spreadsheet that has thousands of rows for Jane Doe (she loves your products and has been a customer for the last 5

years!). It would take a lot of time to update each row for her, and it is error-prone because some rows could be missed. How would the use of a database be more efficient?

Solution

Here is the structure that we would design for this database.

First, we define a table "Clients" to keep the clients' names, and assign a unique key (client id) for each client.

Clients Table

Client-ID (key)	Client Name
1	Jane Johnson
2	John Smith
3	Bob Jones

Next, we define a table "Transactions" to keep track of money spent per client

Transaction Table

Date	Client-ID	Amount
1/1/2023	1	\$100
1/5/2023	2	\$50
1/7/2023	1	\$25
1/10/2023	3	\$200

In the database, the Clients table stores the master list of clients. The Transactions table references the ClientID as a unique key to 'relate' the two tables.

To change Jane Doe's name, only change is made at one place: only the name field in the Clients table needs to be updated. The Transactions table automatically reflects this change via the ClientID key.

The company now has a database structure that maintains data integrity and normalization much more efficiently than the spreadsheet when client data needs to be updated.

4.5.1: Streaming

Streaming is a new easy way to view on-demand audio or video from a remote server. Companies offer audio and video files from their server that can be accessed remotely by the user. The data is transmitted from their server directly and continuously to your device. Streaming can be accessed by any device that connects to the internet. There is no need for large memory or having to wait for a large file to download. Stream technology is becoming very popular because of its convenience and accessibility. An example of some streaming services is Netflix, iTunes, and YouTube.

4.5.2: Other Types of Databases

The relational database model is the most used today. However, many other database models exist that provide different strengths than the relational model. In the 1960s and 1970s, the hierarchical database model connected data in a hierarchy, allowing for a parent/child relationship between data. The document-centric model allowed for more unstructured data storage by placing data into “documents” that could then be manipulated.

However, the relational database model does not scale well. The term scale here refers to a larger and larger database being distributed on a larger number of computers connected via a network.

To support large scale models, Some companies are providing new non-relational databases as shown in the Figure 4.5.2

Figure 4.5.2: Non-Relational databases, by Ly-Huong Pham, Ph.D., is s licensed under [CC BY 4.0](#)

Non-Relational Databases

Closed Source (or hybrid) non-relational databases	Open Source non-relational databases
<ul style="list-style-type: none"> • MongoDB - MongoDB Inc. (public company) • Redis - Redis Labs (private company) 	<ul style="list-style-type: none"> • Cassandra - Apache Software Foundation • HBase - Apache Software Foundation
<ul style="list-style-type: none"> • Neo4j - Neo4j Inc. (private company) 	<ul style="list-style-type: none"> • Elasticsearch is open source software under the Apache license and managed by the non-profit Elasticsearch BV organization, not owned by them
<ul style="list-style-type: none"> • DynamoDB is a fully proprietary database service owned and managed by Amazon. But the DynamoDB software itself is not strictly closed source 	<ul style="list-style-type: none"> • Google' solutions: Bigtable, Cloud Datastore, Cloud Firestore, Cloud Memorystore

Definition: NoSQL

NoSQL databases are non-tabular data stores that store and access data in ways other than traditional row-column relations used in relational databases. Definition text

The concept of NoSQL (from the phrase “not only SQL”) arose from the need to solve large-scale databases spread over several servers or worldwide. For a relational database to work properly, only one person must be able to manipulate a piece of data at a time, a concept known as record-locking. But this is not possible with today’s large-scale databases (think Google and Amazon). A NoSQL database can work with data more loosely, allowing for a more unstructured environment communicating changes to the data over time to all the servers that are part of the database. Many companies collect data for all sorts of reasons, from how many times you visit a site to what you are viewing at the site. Without complex SQL queries, NoSQL databases typically access data through simple APIs and operate in a non-tabular format to allow for efficient handling of semi-structured and unstructured data at very large scale. Their dynamic and distributed nature makes NoSQL databases well-suited for contemporary big data and real-time web applications.

This page titled [4.5: The Difference between a Database and a Spreadsheet](#) is shared under a [CC BY 4.0](#) license and was authored, remixed, and/or curated by [Ly-Huong T. Pham and Tejal Desai-Naik \(Evergreen Valley College\)](#) .

- [4.5: Sidebar- The Difference between a Database and a Spreadsheet](#) by Ly-Huong T. Pham, Tejal Desai-Naik, Laurie Hammond, & Wael Abdeljabbar is licensed [CC BY 3.0](#).