

Database Basics: Concepts & Examples for Beginners

Get started with relational databases by understanding organized data storage, an overview of management and analytics, and how it all relates to spreadsheets!



10-minute read

Table of Contents

- What's organized data?
- How's organized data stored efficiently?
- How are databases and spreadsheets different?
- Taking your spreadsheets to the next level.

Let's get started...

The importance of well-presented data cannot be understated in today's digitally advanced landscape. Companies around the world are focusing their entire strategies based on data, so they can understand their customers well. Facebook, Amazon, Netflix, and Google are just some of the large corporations whose business model revolves around providing personalized recommendations to their users. This has been made possible only through organized data.

So, what is organized data?

Think of organized data as a good old file cabinet (but on your computer)

Organized data can be any representation of data that allows you to gather insights. What's more necessary is that it should be relevant to your department. If you work at an insurance firm, you'll want to have information that includes customer credit history, age, bank records, etc. What you won't be concerned with is their favorite TV show or what books they like to read.



All data is powerful, you just have to make sure that you're dealing with something that concerns your end goal.

Sometimes you will need to tackle multiple datasets together to form useful insights. When multiple datasets are concerned, things can get complicated very easily and it can become time-consuming to constantly move back and forth between heaps of data.

†

Database Basics: What is a Database?

The most efficient way to store data is with the help of a database. A database is made up of tables that contain columns and rows. Each category is given its own table. For example, a company may have a table for customer information and another for sales numbers. You can think of a table somewhat like a spreadsheet. Inside a spreadsheet there are columns and rows of data. *For a database however each row is called a record and each cell is called a field.*

When people talk about a database they are usually referring to a relational database. This is the oldest type of database and has been used for over 40 years.

A relational database consists of 3 high-level components:

- Tables
- Relationships
- Keys

Value		Metric	
1		B	byte
1000		kB	kilobyte
1000	²	MB	megabyte
1000	³	GB	gigabyte
1000	⁴	TB	terabyte
1000	⁵	PB	petabyte
1000	⁶	EB	exabyte
1000	⁷	ZB	zettabyte
1000	⁸	YB	yottabyte

A yottabyte is a lot of bytes!

With these assets, you can easily link zettabytes (1,000,000,000,000,000,000,000 bytes) of data into something meaningful that can easily be traversed at will to see everything you could want to.

Do you want to look at a specific segment of your data? Easily doable. Do you want to look at ONE particular result from a set of millions? No problem. How about looking at those 27 anomalies in your data that could be interesting to observe? [Relational databases](#) will always have your back. **The flexibility that comes from having a relational database is unparalleled. Nothing has come close to being as mainstream and useful as relational databases and for good reason.** Let's now discuss each of those 3 components in detail to make sense of what they are.

Tables

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7369	SMITH	CLERK	7902	17-DEC-80	800	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500

A very basic data table.

Tables are the Microsoft Excel equivalent of a single spreadsheet. They can also be classified as standalone datasets. Tables are used to organize the most closely related data together. A very basic example of a table could be a dataset about people that contains a bunch of people's names, job titles, manager numbers, hiring dates, salaries, and commissions. This information would be stored in a column and row format. Rows and columns also happen to be the very foundation of a table.

Where columns are used to store different information about one person, rows store information about different people. With both of them paired together, it ends up becoming a table full of information. Let's discuss both of them in more detail.

Columns

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7369	SMITH	CLERK	7902	17-DEC-80	800	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500

Some very basic columns.

Columns are used to differentiate the information we have on a single observable entity. In a Table that contains information about people, the columns would be used to hold different information.

If a Table, as mentioned above, contains people's names, job titles, manager numbers, hiring dates, salaries, and commissions, then that table will have 6 columns plus a Primary Key column that we will discuss in later sections.

Each column can be set up to allow only a specific type of information to be entered into it. This aspect allows for much-needed data integrity. For example, a column about salary should only contain numbers, right? While that is true, the people operating the databases are humans and can therefore accidentally enter something else in it. To prevent this from happening, columns can be designed to only let a specific type of information to be entered.

The same goes for an email column. Anything that does not end in the typical '@abc.com' should not be allowed inside that column.

The customization that goes into a column is pretty much endless. There are many presets available and custom options too.

Rows

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7369	SMITH	CLERK	7902	17-DEC-80	800	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500

Now, some very basic rows.

Rows of a table represent the number of observable entities we are looking at. To put it simply, if the people table has 3 rows, it means it has the data of 3 different people. Each row represents an individual person and the columns will display their respective information.

Rows allow us to see individual entries in the table. Each row also contains a Primary Key that allows us to search for individual entries with ease.

Keys

Keys allow unique identification for all rows in the table. Without keys there would be no way to differentiate between entries that have identical information in their columns. Two people in a table can have the same names and birthdays and without a unique key, it will be hard to differentiate between them and can lead to unnecessary confusion.

Suppose you're an HR person who has to send a termination letter to a guy named John Doe and a promotion letter to another person with the same name. Imagine if that gets mixed up, both receive the termination or promotion letter. Talk about a corporate nightmare, right?

There are two types of keys you should know: a primary key and a foreign key.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7369	SMITH	CLERK	7902	17-DEC-80	800	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500

You guessed it — a basic primary key.

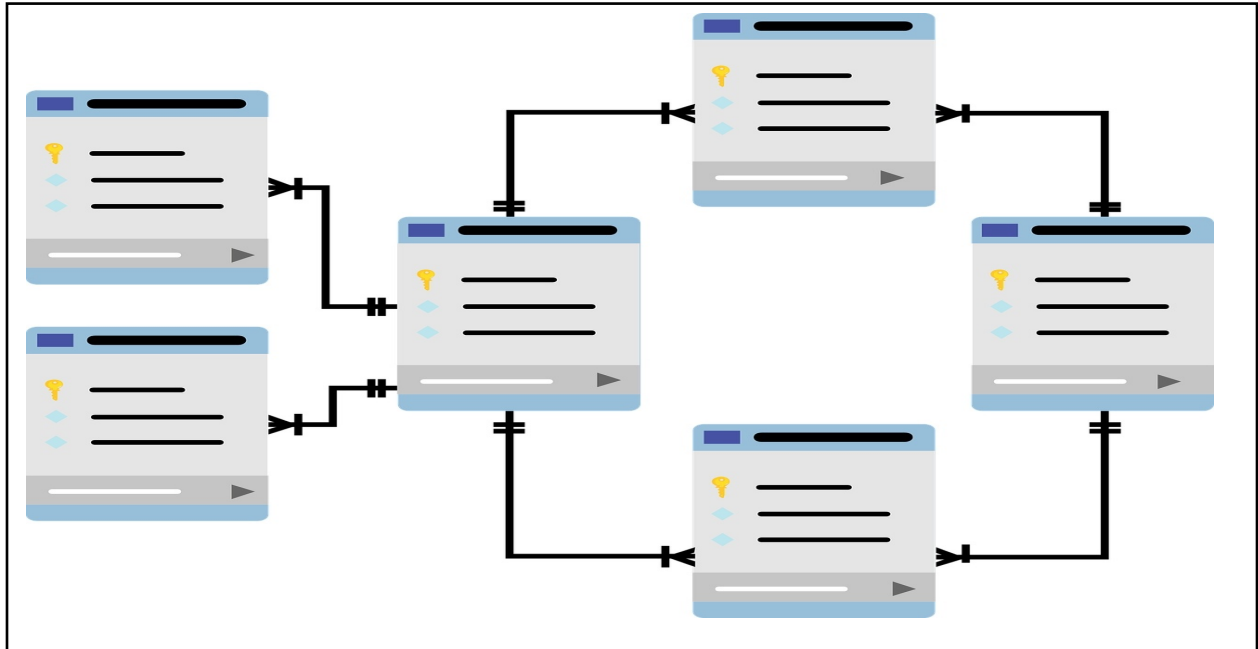
Primary Key

Primary Keys are how every row in the table is searchable. They can be a single column or a combination of columns that make up a unique identification number.

Foreign Key

Foreign Keys are used to link tables together within a database. These links are called relationships.

Relationships



Relationships help efficiency across teams. And, as we all know, teamwork makes the dreamwork.

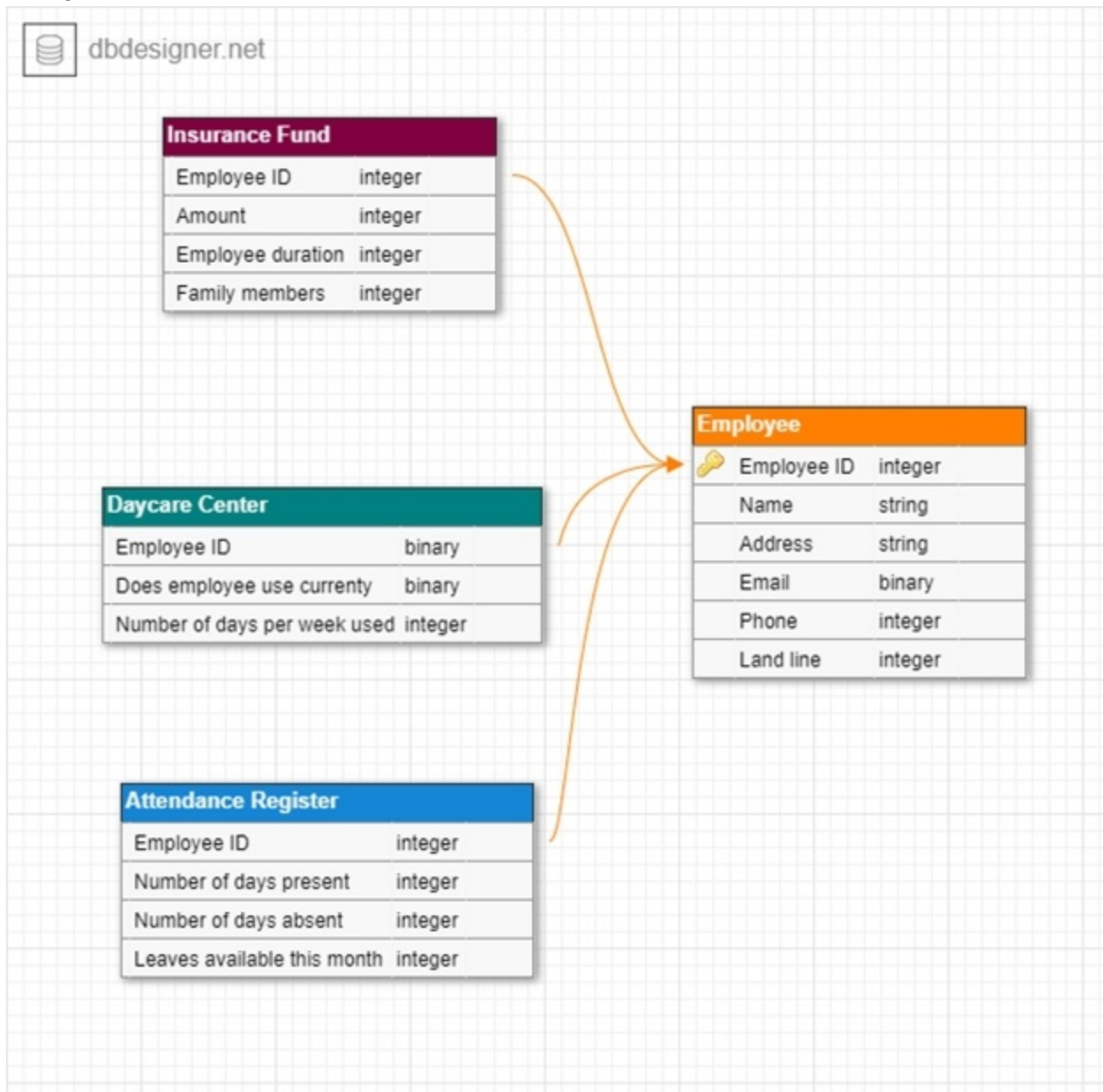
This is the part where the link between various tables starts to develop. Relationships allow a multitude of tables to contain different, but related kinds of information, while at the same time maintaining readability and optimizing space.

Imagine a small company that has different sections and departments for its employees like an insurance fund, a daycare center, an electronic attendance register.

While all this information may be useful, reading all of it together won't be. If the HR department would like to see the insurance information of a specific employee, they will not be concerned with the use of the daycare center. In fact, it will only become harder to read with so many columns in place.

Storing the databases is also not a superficial issue as they can demand a ton of space once they start to grow in size. It's not optimal for every computer in the company to have the entire database in terms of storage and security issues.

To tackle this, relationships are implemented between various tables. **Relationships essentially allow splitting up of information into useful components that emphasize readability and efficiency. This also means that different departments will only have access to what they need and the rest will not be available on their computers.**



Applying our small company scenario to a relationship flow chart.

Separating the important information

In the above small company example, the best way to form relationships would be to have 4 tables. Each to represent employees, the insurance fund, the daycare center, and the attendance register.

Now, the information is split quite nicely. The next step is to identify what information would be present in most areas. In this case, everything we split up would be used by employees so it makes sense to consider it as our main table.

Think about what sort of information an employee table should have from your own experience of working in an office. Every staff member has the basic name, address, phone, email, and age columns along with something known as an ID number. That ID number is unique to each worker and it also serves as the Primary Key of the Employee table.

Using the ID number, we can search for information on any employee we want, even if we have employees with the same name. If by some odd miracle all your staff have the same name and age, you can still tell them apart by their ID number. Does that mean we can use the Employee ID to somehow link to other tables? Absolutely.

Forming relations with other tables through Foreign Keys

The Employee ID column from that table will be used as the Primary Key for the other tables. When other tables are linked this way, they are said to have a Foreign Key. It simply means a key from outside. This Foreign Key lets us link all the other information to each worker. All the information we have in other tables is tied to specific people, therefore having the Employee ID in every other table makes perfect sense.

While many technicalities decide how relationships are made in complex systems, this is one of the most simple examples that you can find in any relational database book.

Common Terminology

When discussing relational databases, some of the most common terms you'll hear are SQL and synonyms for other properties. Columns may also be known as attributes, fields, or features. Rows may also be called records, entries, or tuples.

SQL is a programming language that was designed to make databases easier to work with. The power of this tool cannot be understated. With a strong grasp on its core concepts, you can do pretty much anything you want with the data when it comes to insights. *Most commonly, SQL is used to extract (or query) data from the database.* With this language, you can specify what data you want and what the output should look like. This is how you can take data from a database into Microsoft Excel or Google Sheets.

How are relational databases different from Excel/Google Spreadsheets?

On a very small scale, spreadsheet programs can work well. But the minute you start to think about scalability, security, and usability, spreadsheets no longer suffice.

Scalability

This deals with having the data available to more than one person. While you may be able to work with giving access to that online spreadsheet to a few people, it simply can't work when you need many different departments looking into it every day. The chances of two people working on the same thing are not negligible and can lead to serious problems. If you've ever worked with another person on [Google Docs](#) at the same time, chances are you've experienced small hiccups.

Security

Data is precious and giving all employees access to it can result in disaster. Not all employees need all the data and some of it needs to be kept confidential. For example, Amazon has a lot of sensitive customer data: addresses, phone numbers, credit card information, etc. They have over 50,000 corporate employees (over 700,000 corporate and non-corporate), but most do not have access to this information. Amazon uses databases to restrict access and protect customers.

Usability

If you have a ton of data but can't find an efficient way to get insights from it, it's pretty much useless. Tools other than relational databases just don't have that kind of power to extract such meaningful information that SQL and other database languages possess.

As your data increases, you might have to shift to a relational database forcefully. Spreadsheets can only handle so much data. Google Spreadsheets has a limit of 5 million cells and in today's time, that really isn't a lot. It won't take long for that to be filled up and migrating with 5 million cells to a database is going to be troublesome. It's recommended you start your migration the moment you observe growth. Databases can handle as much you can throw at them and this is why you seem them implemented anywhere data exists.

The flexibility that comes from having a relational database is unparalleled. Nothing has come close to being as mainstream and useful as relational databases and for good reason.

Summary

To recap, we explored why organized data is important and how you can organize data with a relational database. A relational database consists of multiple data tables linked together through keys and relationships.

Tables, keys, and relationships are the three core components of a relational database. Tables are made up of rows and columns. Rows represent individual entities in a table where columns represent their attributes. Keys (primary and foreign) are one of the key concepts of what makes relational databases work. **Relationships between tables are the link that makes the data much more meaningful.** They explain how things are actually connected and what connects them.

Without keys and relationships linking tables together, there is no significant difference between multiple spreadsheets and a relational database. Finally, we reviewed common database jargon that you should get familiar with. It's mostly synonyms of other things, but it can help to know what's being discussed.