

Databases in the Real World

by Sophia



WHAT'S COVERED

In this lesson, you will explore real-life scenarios of how databases are used in businesses. People and companies interact with many databases every day. People and customers will interact with databases to get information about products or services, often unaware of the systems they are interacting with. Businesses interact with databases to provide data, get information, make decisions, and keep track of payments in and out. Sometimes when a business is rolling out a new technology, it must decide whether to create its own database, expand an existing one, or avoid databases altogether. You will learn about interacting with a database, in two parts. Specifically, this lesson will cover:

1. Daily Interactions With Databases

2. To Build or Not to Build

1. Daily Interactions With Databases

You interact with data and databases throughout your life. Suppose you ordered a product online, used a streaming service to watch a movie, played a video game, used a cloud storage to save or back up files, played fantasy football, interacted with a bank, or made payments using a debit or credit card; in each case, you have interacted with a database. AI (artificial intelligence), social media, and e-commerce are the biggest systems using databases today.

Let's walk through a day with Carlos and see how he uses a database in just half of a day.



IN CONTEXT

Carlos wakes up in the morning and asks Alexa to play his morning song list. (Alexa uses machine learning to know what Carlos likes and stores that data in the cloud for later retrieval, as well as breaks down each of Carlos's words and matches them against a database of sounds so it knows what Carlos is saying.)

Carlos opens his phone and checks his email (email is stored in a type of database) and his TikTok channel (TikTok uses MySQL, Mongo, ByteGraph, and other databases) for new information. He looks at his Twitch channel (PostgreSQL and others) to see who did the last game run and how it went. Then he gets ready to go to work.

Carlos goes to his car, looks up his driving route to see what traffic looks like on Google Maps (Bigtable) and then heads off to work. On the way to work, he asks Siri to order coffee for him at the local coffee shop (Oracle Exadata database) so he can swing through the drive-through and get his favorite coffee. Google Maps will automatically update his route to work for him and give him a new traffic map.

Once at work, Carlos starts to work on customer orders (the company uses PostgreSQL) and coordinating global shipping for orders. On his cell phone, he gets a reminder that he has an online

test at 11:00 AM for his math class using the school's learning management system (LMS). He acknowledges the alert, saying he will be online then and ready for the test (PostgreSQL).

At 10:50, he logs into the school and goes to the class to prepare for the math test. He told his boss he needed to take the test (Outlook Calendar database) by blocking off the hour. The LMS takes his login and lets him know the test is ready early. Does he want to take it now?

At 11:55, he completes the test and gets instant feedback that he passed the test with a 92% (PostgreSQL). Carlos takes a picture of the passing grade and posts it to Instagram (PostgreSQL, Redis, and Memcache) to let his friends and family know he passed the test.

This is just part of Carlos's day, and he has interfaced with many different kinds of databases throughout his morning. Databases and their design are structured by the companies that make the software and provide the service to people. Carlos's typical morning reflects how databases have influenced many systems we use and how we interact with others globally.



THINK ABOUT IT

As a fun exercise, try to map out where you may have interacted with a database today. Are there apps that you use or games that you play that track your progress through the game, or apps that track your progress through their program? How does your news get delivered to you? What about the messages that you send to your friends and family? You'll probably be surprised at how many different kinds of databases you work with every day.

2. To Build or Not to Build

Businesses must consider many things when building a database. They must evaluate the database's desired performance and the database capabilities to solve the business problem. They must also determine when to build a database, and when it is unnecessary to have one. In our example, we showed that Carlos uses many databases. However, some of those same systems utilize cloud storage for media and databases to store other information. Databases should be considered if many users interact with them, especially if some are customers rather than employees. This is because databases have very effective security and privacy built into them. Some users may only need to read data in the database, while others may also have to insert, update, and delete data from the business database.

If a company already has a database, a separate new database may not be needed. However, the existing database must have the hardware and software capacity for the addition.

⇒ **EXAMPLE** If a company had an e-commerce site with a database that stores user names and email addresses, it would not make sense to build a separate new database to track newsletter signups from its customers. Adding a field or table to the existing database would make more sense to collect email addresses as part of the newsletter signup process. That way, they could track the users who sign up for the newsletter and those who do not.

Although the company in that example could certainly add its newsletter mailing list to its existing database, that might not be the best solution for all companies. For example, if the current database is nearing its maximum capacity in terms of data storage space or processing power, or if it cannot be modified to meet the new use, setting up a separate new database in parallel to the original one may make sense. The new database could be set up to pull data from a certain table in the existing database to avoid redundancy.

A company that does not currently have a database—or a database administrator to create one—may prefer to outsource its newsletter operations to a third-party service that can manage recipient names and addresses and send the newsletters out to them. The cost to develop a database, or to add additional capabilities to an existing database, may be greater than the cost of a subscription to a third-party service.

Depending on the size and scope of the project, a database may not be required at all. For a very small mailing list, a company might get along just fine maintaining the list in a spreadsheet and using the Mail Merge feature in an email app such as Microsoft Outlook to mail out a monthly newsletter. Such a spreadsheet would be considered a flat file, and that's the subject of the next lesson.

Let's recap.

Reasons for a business to build a new database include:

- They cannot change their existing database to suit their new needs.
- The business needs to construct a new application that cannot connect to an existing database.
- The current database is at capacity and cannot take on additional roles without upgrading hardware, which can be expensive.

Reasons to avoid building a new database include:

- The existing database can be adapted by adding a table or field to satisfy the need.
- A simpler or cheaper tool can satisfy the business need.
- A third-party business can provide a service to fill the need.
- The business needs to access the existing data in a new way rather than to store new data.



SUMMARY

In this lesson, you learned that businesses have many reasons to store customer data in a person's **daily interactions**. There are many different scenarios in which a business may or may not want to use a database. You also saw some of the considerations for using a database in a business. You also got to see how many different kinds of databases you or others interact with every day as you go about your normal lives. These insights will help you determine whether to **build or not to build** the database. Next time, you will learn about the differences between a database and another means of storing data, the flat file.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND FAITHE WEMPEN (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR [TERMS OF USE](#).