

Databases vs. Flat Files

by Sophia



WHAT'S COVERED

This tutorial explores the differences in flat files and databases in terms of flexibility, data consistency, security, and data redundancy in three parts:

1. Introduction
2. Flat File History
3. Weaknesses of Flat Files

1. Introduction

Databases evolved from the use of **flat files**. A flat file is a data file that is not related to or does not contain any linkages to another file. As you have seen, relational databases provide a means of organizing various topics into tables that relate to one another through the primary key and foreign keys. Flat files, on the other hand, store data in a single table. Although this can work in certain cases, problems occur as the relationships of the data become more complex.



TERM TO KNOW

Flat file

A data file that is not related to or does not contain any linkages to another file.

2. Flat File History

Physical file systems are where the idea of databases began. Those old systems were handled with paper and pencil. These types of paper systems were organized through a system of file folders and filing cabinets. Typically, there wasn't a significant amount of data collection required and reporting was quite limited. This type of system worked well as a data repository. However, as companies started to become larger and needed to report on their data, managing the data in these physical file systems became too difficult.

Computerized file systems came next. Generating reports from physical file systems were quite difficult to run on a regular basis and were error-prone. Computer files in early computerized file systems were similar to physical files, but stored on a computer similar to a simple spreadsheet. If a business user needed information, there were central individuals that handled all of the data processing requests. These early computerized flat-file systems became just as much of a burden to manage due to related and overlapping data, with no means of controlling or managing the data across each of the files.

The shortcomings of the physical and early computerized file systems are one of the big reasons for creating database systems. Those old systems took longer to develop, still had challenges to returning quick answers and making sense of the data, and had very complex administration. They also lacked security.

3. Weaknesses of Flat Files

To illustrate the difficulties associated with flat files, let us consider the Postgres database that you have been working with. Look at the invoice table and customer table. In the database, invoices are stored in the invoice table and related to the customer through the customer_id. If you had to use a flat-file database, you could only have a single table in which to store both the customer and invoice data. All of the information would be stored in a single record. The customer's information, like the name, address, phone, and email, would be stored in the same row as the invoice information. If the customer makes more than one purchase, all of that customer information would have to be re-entered for each item they order.



BRAINSTORM

Imagine if you had to do that every time you made an online purchase from a company. You could enter a different address or a nickname on some of the items. How would the company identify which orders belong to you? What if you moved and needed to change your address? What additional kinds of problems can you anticipate with this system?

Keeping customer data in flat files creates a lot of data redundancy, as a company would have the same data stored multiple times not only in the same flat file but perhaps in other flat files for other purposes.

If you have data redundancy due to repeated data, then you have poor data security. Having multiple copies of the same data increases the chance of unauthorized access to the data. For flat files, the operating system would have to handle the security rather than the database itself.

There would also be potential data inconsistency, like the address scenario considered above. You could have different and conflicting versions of data appearing in various locations in your records. Data inconsistency can further increase if you have data-entry issues that go undetected. You could also have data integrity issues, as you may have products being entered in that do not exist at the company.

Finally, since you would have repeated information, the data would become a lot larger and more difficult to manage and maintain. You would also lack the flexibility to manage the data in a flat file, as you could only enter in data that fit the data structure. In the example, you would only have one customer and one invoice inserted together in the same flat file table. If you later wanted to record other information about customer transactions, you might not be able to change the available columns to include those options.



SUMMARY

Databases have evolved from physical file systems and computerized flat file systems. In doing so, they improved flexibility, data consistency, and security, and reduced data redundancy.

Next time, you will learn more about the variety of parts of a database system.

Source: Authored by Vincent Tran

**Flat file**

A data file that is not related to or does not contain any linkages to another file.