You've learned what a database is and how data is related in a database. You've also learned about common database projects and alternative types of databases. In this reading, you'll find out more about the history of database technology and how it evolved.

The history

The history of databases begins in the 1960s with the computerization of databases. Computers emerged as a more cost-effective option for organizations. It also became easier to shift data storage and databases to computers.

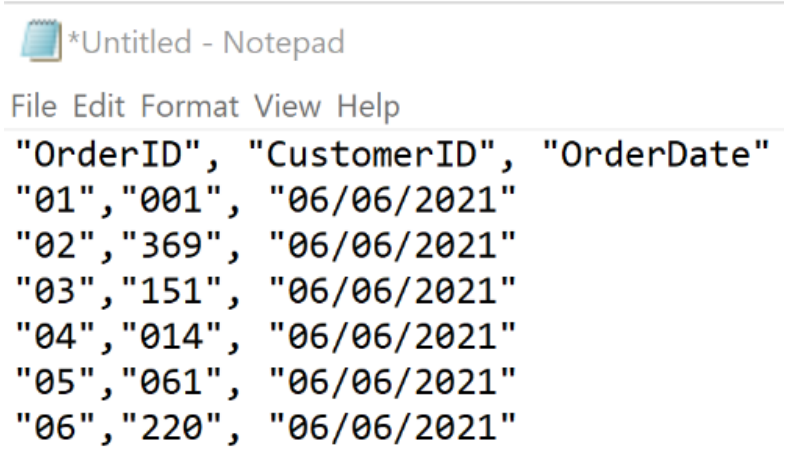
The chronological order of the development of databases is as follows:

- (1970s-1990s) - Flat files, hierarchical and network
- (1980s-present) - Relational
- (1990s- present) - Object-oriented, object-relational, web-enabled

Flat files

Flat files databases were used during the 1970s-1990s. This is a type of database system that stores data in a single file or table. They are basically text files, where every line contains one record and fields either have fixed lengths or are separated by commas, whitespaces and tabs. Such a file cannot contain multiple tables.

Below is an example of what a flat file database looks like. This text file stores lines of data, where each line represents a record. The fields OrderID, CustomerID and OrderDate are separated by commas.



Hierarchical database systems

Hierarchical database systems that were in use during the same era store data in a hierarchically arranged manner.

Think about it this way: parents can have many children, but one child can only have one parent. In other words, the database represents a one-to-many relationship: all attributes of a specific record are listed under an entity type.

Below is an example of how data is stored in a hierarchical database. In this case, it is data on college students who are taking different courses. A course can be assigned to only a single student, but a student can take as many courses as they want. Thus, there is a one-to-many relationship.

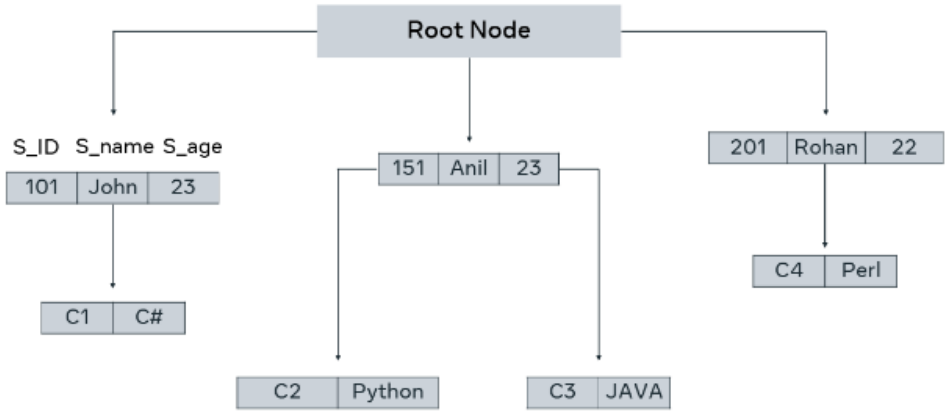
There are three students:

- John,
- Anil
- and Rohan

And there are four courses:

- C#,
- Perl,
- Python
- and Java.

Student and Course are the entity types. John takes C# and Anil takes both Python and Java. Rohan takes Perl.

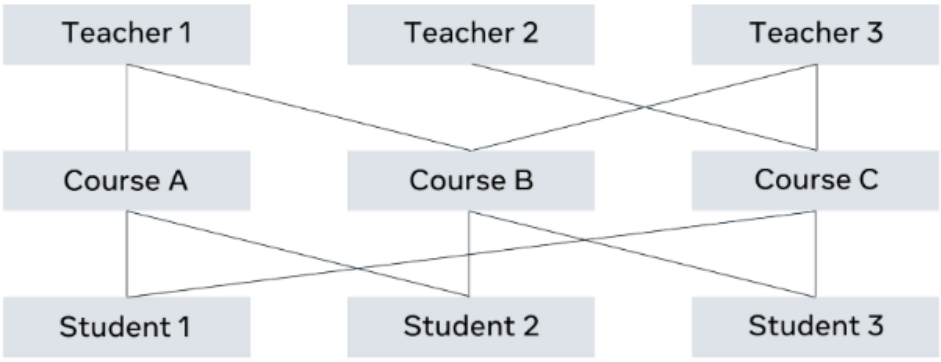


Network databases

Network databases were introduced by Charles Bachmann. Unlike the hierarchical database model, a network database allows multiple parent and child relationships. In other words, many-to many relationships. In network database terminology, a child record is known as a member. A member or child can be reached through more than one parent, which is called an owner.

A network database has a graph-like structure, and it allows you to represent more complex relationships among data.

Here's an example of a network database. A teacher can teach multiple courses and a course can have multiple teachers teaching it.

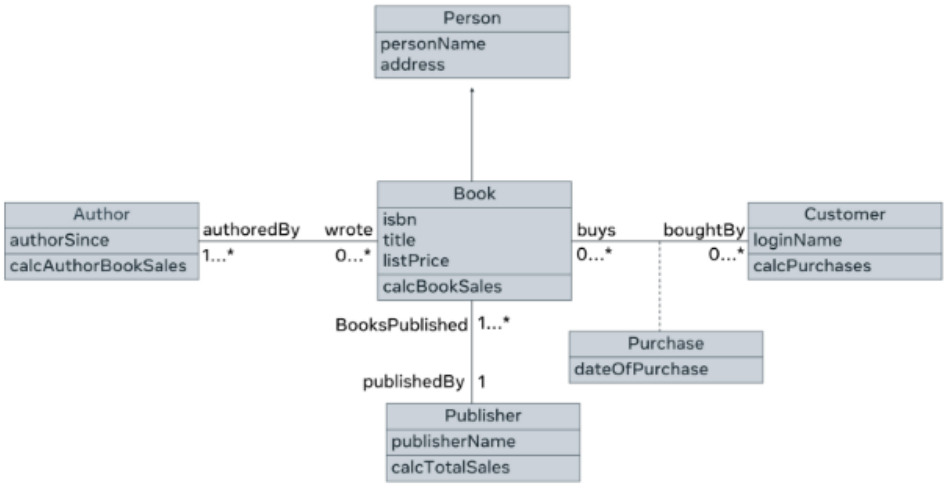


In this era, a language known as the SEQUEL query language was used to work with databases. Later on, with relational databases, this developed into SQL (Structured Query Language) which was made a standard query language to work with databases by the American National Standards Institute once relational database systems were introduced.

Relational database system

The relational database system that was introduced in the 1980s is still the most used database system. It was invented by E. F. Codd and it's the successor of hierarchical and network database systems. It was viewed as a major paradigm shift in database technology. In a relational database system, data is stored in tables. The columns of the table hold attributes of the data. Each record usually has a value for each attribute, making it easy to establish the relationships between data points. In a relational database, each row in the table is a record with a unique ID attribute called the primary key. A relational database stores and provides access to data that are related to one another using an attribute known as a foreign key.
Here's an example of what a Relational Database would look like. Here, there are tables with attributes/ columns that store rows/records of data in them. The relationships between data in tables are established using key columns known as foreign keys that are themselves primary key(s) of a given table. For example, the primary key of the PROFESSOR table is PROF_ID and in the CLASS table, it's there as a foreign key. It creates the relationship between the PROFESSOR table and the CLASS table. Another example, the COURSE_ID is the primary key of the COURSE table, and it is there in the CLASS table as a foreign key. It establishes the relationship between the COURSE table and the CLASS table.

Object-oriented databases
In the 1990s, object-oriented databases were introduced. This was when the object-oriented (OO) programming paradigm became popular and there was a need to represent data in a system as objects as well. Unlike relational databases, object-oriented databases work in the framework of real programming languages like Java and C++, for example.
Below is what an object-oriented database looks like. Instead of tables, there are entities or classes like Author, Book and Customer with their attributes and behaviors.
It's possible to represent data according to OO concepts like inheritance and parent-child relationships among data. For example, an Author and Customer are both descendants of Person. Thus, a person is a generic entity that can represent both an Author and a Customer.



NoSQL databases
Relational databases that are widely used even at present only allows to store structured data. Later on, there was a need to work more and more with unstructured data. This was when NoSQL databases came about as a response to the Internet and the need for faster speed and the processing of unstructured data. NoSQL databases are preferred over relational databases

because of their speed and flexibility in storing data. It does not store data in relations or tables that belong to a strict structure. Data can be stored in an ad-hoc manner and they allow to store and process high volumes of different kinds of data. NoSQL databases are capable of processing unstructured big data that's generated by social media, IoT and others. Therefore, social platforms like Twitter, LinkedIn, Facebook, and Google for example makes use of NoSQL databases.

These are some of the advantages of NoSQL databases:

- Higher scalability
- Distributed
- Lower costs
- A flexible schema
- Can process unstructured and semi-structured data
- Has no complex relationships

Over time there were different types of NoSQL databases that were introduced:

- Document databases store data in documents similar to JSON (JavaScript Object Notation) objects. Each document contains pairs of fields and values. The values can typically be a variety of types including things like strings, numbers, booleans, arrays, or objects.
- Key-value databases are a simpler type of database where each item contains keys and values.
- Wide-column databases store data in tables, rows, and dynamic columns.
- Graph databases store data in nodes and edges. Nodes typically store information about people, places, and things, while edges store information about the relationships between the nodes.