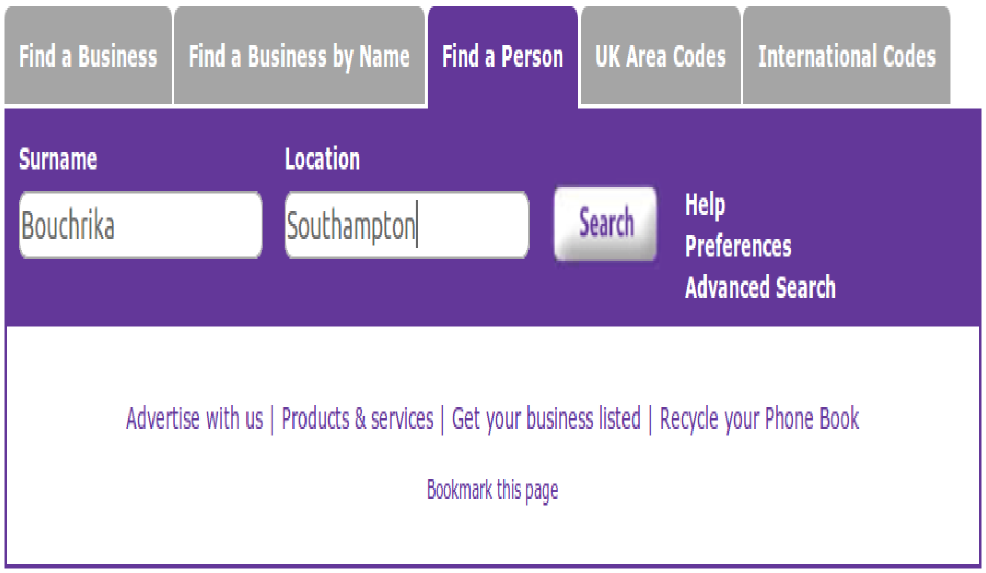
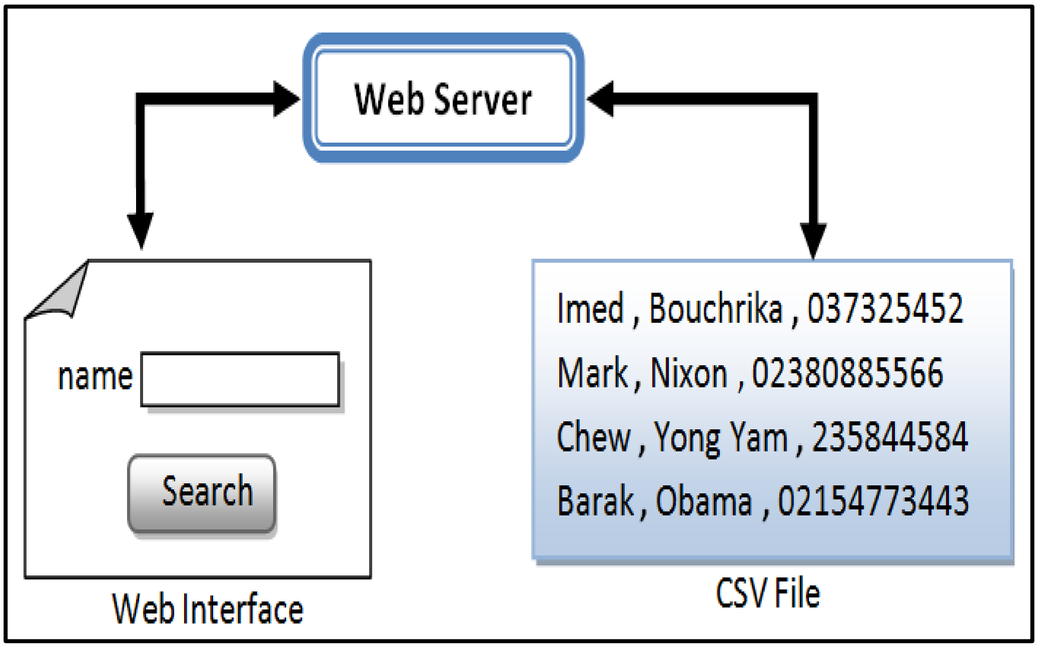
Introduction

Motivations of why to Learn Databases

Before we define or explain what a database or even argue whether it is important for software development, let us ask the simple question: *Can we create a software application without the use of a database system.* To make it easier for you to answer such question, we take the simple example for an online phone directory which can be developed using a web scripting programming language such as PHP. The application is normally used by online visitors to search for phone numbers by typing the person surname as well as the address. The Figure below shows an example for the online phone directory provided by British Telecom to search for people phone numbers within the United Kingdom.



The question that poses is where and how we can store the list of people’s names and their phone numbers permanently as well as retrieve them easily. In other words, how and where to store data such that it can be accessed by the web application. Trivially, the data would never be stored inside the application code as any modification needed for the data requires altering the code and therefore exacerbating compilation and deployment again which is an expensive and cumbersome process. The simplest solution would be to store data inside a normal text file whereby we format the data using a simple pre-defined structure. For instance, we store each record for a new person on a new line whilst the values for a record including name, surname and phone number should separate by a comma “,”. That’s what we call a CSV file “comma separated values” as illustrated in the following Figure which shows a simple architecture for the phone directory. There is a web interface written usually in HTML having a simple form to search for a person by name. Upon clicking the “search button” within the form, data is transmitted into the web server for processing. The server handles search and retrieval of data from the csv text file.


Web Server
name
Search
Web Interface
Imed , Bouchrika , 037325452
Mark , Nixon , 02380885566
Chew , Yong Yam , 235844584
Barak , Obama , 02154773443
CSV File

What if we want to enable concurrent or distributed management by different users connecting from remote different locations to the storage facility or files?

What about if we want to grant different access rights or roles to different users. For example, we grant read-only for guests and read-write for administrators.

What about the scalability and availability of the system when adding over a million of records where the performance of the search suffers greatly as it would scan a large number of lines within the file?

Further and critical issues and challenges that needs to be addressed for software applications using flat-files for storage includes: File management, concurrency and multiple access, advanced search or operations, scalability, adding new functionalities, integrity checking, maintenance and backup and security.

Back to the first question of whether we can develop a software application without the use of an advanced database system. Yes, it can be done easily as shown earlier, but life would difficult and tough to address a large number of issues and functionalities for the system. This signifies the importance of using database systems and their critical role for software application as they were invented primarily to address the limitations encountered when using files as a direct storage facility. To further motivate you to learn databases in terms of money and job potentials, a comparative analysis for the average salary for different job roles within the United States of America, have showed that the average salary for an Oracle Database Administrator exceeds the average salary for Java Programmer, Research fellow and even a dentist. The job role for a database administrator is usually to ensure the smooth functioning of an existing database system including maintenance and security hardening for the system. Other database related job roles include database designer, analyst and developer.

1. Data and Databases

1.1 Preface:

You have already been introduced to hardware and software. However, those two components by themselves do not make a computer useful. Imagine if you turned on a computer, started the word processor, but could not save a document. Imagine if you opened a music player but there was no music to play. Imagine opening a web browser but there were no web pages. Without data, hardware and software are not very useful! Data is the third component of an information system.

1.2 Data, Information, and Knowledge

Data are the raw bits and pieces of information with no context. If I told you, “15, 23, 14, 85,” you would not have learned anything. But I would have given you data.

Data can be quantitative or qualitative. Quantitative data is numeric, the result of a measurement, count, or some other mathematical calculation. Qualitative data is descriptive. “Ruby Red,” the color of a 2013 Ford Focus, is an example of qualitative data. A number can be qualitative too: if I tell you my favorite number is 5, that is qualitative data because it is descriptive, not the result of a measurement or mathematical calculation.

By itself, data is not that useful. To be useful, it needs to be given context. Returning to the example above, if I told you that “15, 23, 14, and 85″ are the numbers of students that had registered for upcoming classes, that would be information. By adding the context – that the numbers represent the count of students registering for specific classes – I have converted data into information.

Once we have put our data into context, aggregated and analyzed it, we can use it to make decisions for our organization. We can say that this consumption of information produces knowledge. This knowledge can be used to make decisions, set policies, and even spark innovation.

The final step up the information ladder is the step from knowledge (knowing a lot about a topic) to wisdom. We can say that someone has wisdom when they can combine their knowledge and experience to produce a deeper understanding of a topic. It often takes many years to develop wisdom on a particular topic and requires patience.

1.2.1 Examples of Data

Almost all software programs require data to do anything useful. For example, if you are editing a document in a word processor such as Microsoft Word, the document you are working on is the data. The word-processing software can manipulate the data: create a new document, duplicate a document, or modify a document.

Some other examples of data are: an MP3 music file, a video file, a spreadsheet, a web page, and an e-book. In some cases, such as with an e-book, you may only have the ability to read the data.

1.3 Databases

The goal of many information systems is to transform data into information in order to generate knowledge that can be used for decision making. In order to do this, the system must be able to take data, put the data into context, and provide tools for aggregation and analysis. A database is designed for just such a purpose.

A database is an organized collection of related information. It is an organized collection, because in a database, all data is described and associated with other data. All information in a database should be related as well; separate databases should be created to manage unrelated information. For example, a database that contains information about students should not also hold information about company stock prices. Databases are not always digital – a filing cabinet, for instance, might be considered a form of a database. For the purposes of this text, we will only consider digital databases.

Finally, the name indicates what the database is. A database is one of the essential components for many applications and is used for storing a series of data in a single set. In other words, it is a group/package of information that is put in order so that it can be easily accessed, manage, and update.

In a database, even the smallest portion of information becomes the data. For example, a Student is a data, a roll number is a data, and the address is data, height, weight, marks everything is data. In brief, all the living and non-living objects in this world are data. In this chapter of the database, you will learn about the fundamental terminologies that are used in DBMS.

A database management system is a software to store, organize, manage, and retrieve data. Think of it like a group of massive spreadsheets that organize information. There's more than one type of database management system, and each is housed on servers, whether in a data center or virtually, on cloud infrastructure (cloud database).

Database management systems come in a variety of shapes, sizes, and flavors, each designed to do different things with different kinds of data. MongoDB is a general-purpose, document-based, distributed database management system built for modern application developers.

What type of information is stored in a database?

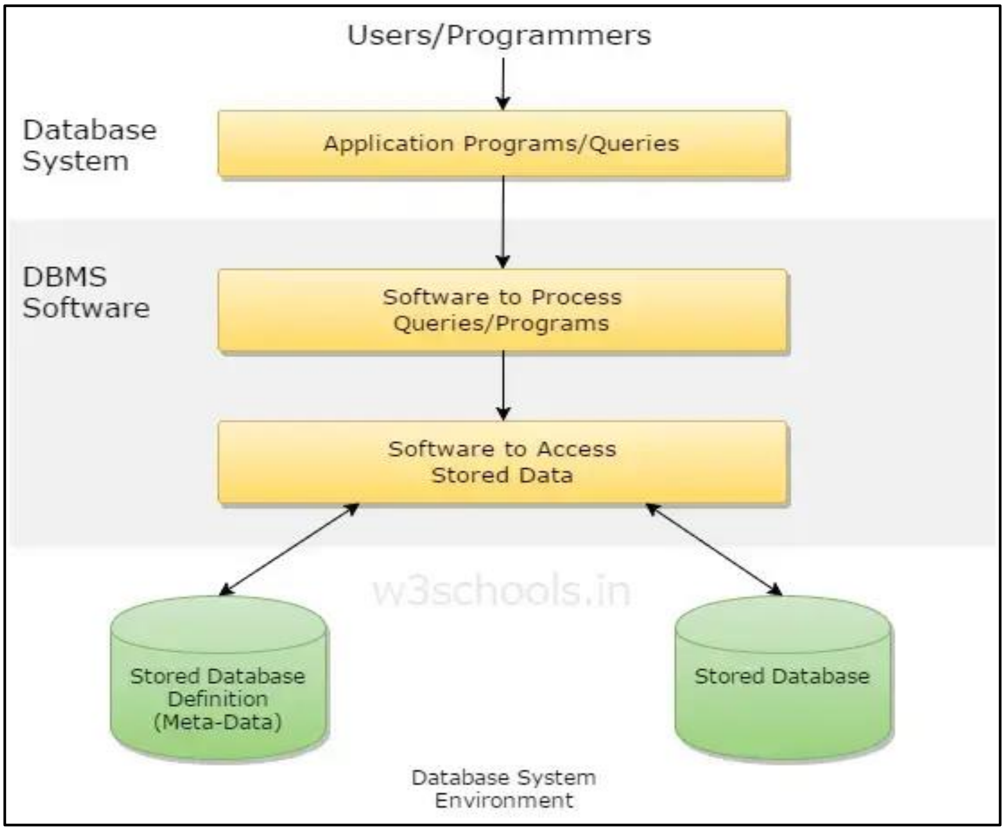
Databases are used in most modern applications, whether the database is on your personal phone, computer, or the internet. An operational database system will store much of the data an application needs to function, keeping the data organized and allowing users to access the data.

If you were building an eCommerce app, some of the data you might access and store in your operational database system includes:

- *Customer data*, like usernames, email addresses, and preferences.
- *Business data*, like product colors, prices, and ratings.
- *Relationship data*, like the locations of stores with a specific product in stock.

1.3.1 Database environment

One of the primary aims of a database is to supply users with an abstract view of data, hiding a certain element of how data is stored and manipulated. Therefore, the starting point for the design of a database should be an abstract and general description of the information needs of the organization that is to be represented in the database. And hence you will require an environment to store data and make it work as a database. In this chapter, you will learn about the database environment and its architecture.

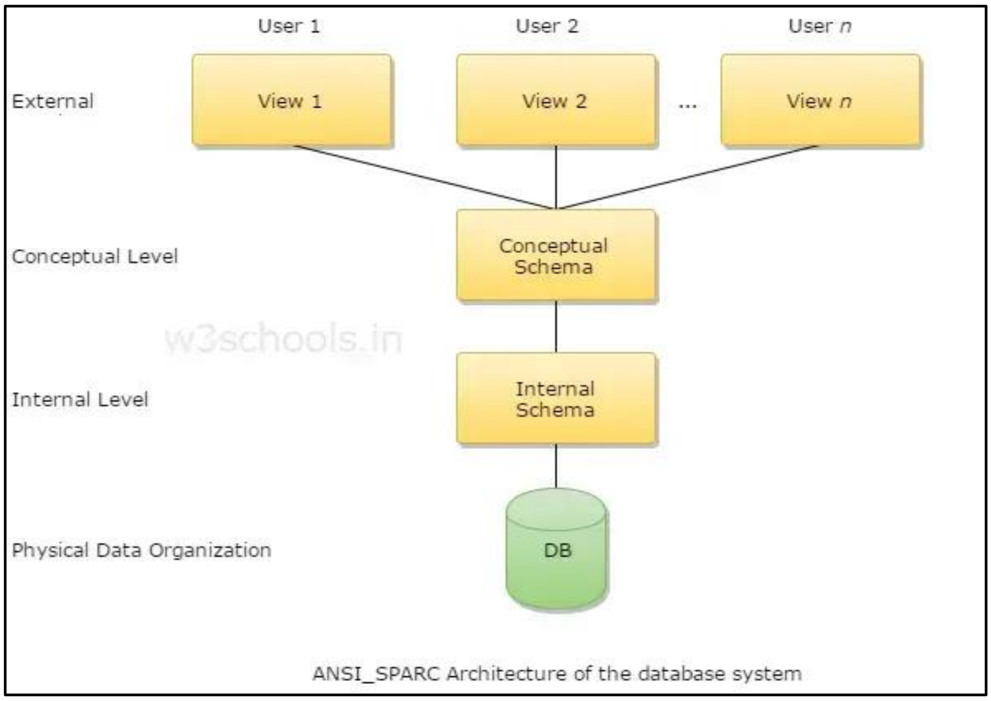

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
w3schools.in
Stored Database Definition (Meta-Data)
Stored Database
Database System Environment

A database environment is a collective system of components that comprise and regulates the group of data, management, and use of data, which consist of software, hardware, people, techniques of handling database, and the data also.

Here, the hardware in a database environment means the computers and computer peripherals that are being used to manage a database, and the software means the whole thing right from the operating system (OS) to the application programs that include database management software like M.S. Access or SQL Server. Again, the people in a database environment include those people who administrate and use the system. The techniques are the rules, concepts, and instructions given to both the people and the software along with the data with the group of facts and information positioned within the database environment.

1.3.2 Database architecture.

An early proposal for a standard terminology and general architecture for database systems was produced in 1971 by the DBTG (Data Base Task Group) appointed by the Conference on Data Systems and Languages (CODASYL, 1971). The DBTG recognized the need for a two-level approach with a system view called the schema and user views called sub-schemas.



Here is the figure showing the ANSI_SPARC Architecture of the database system:

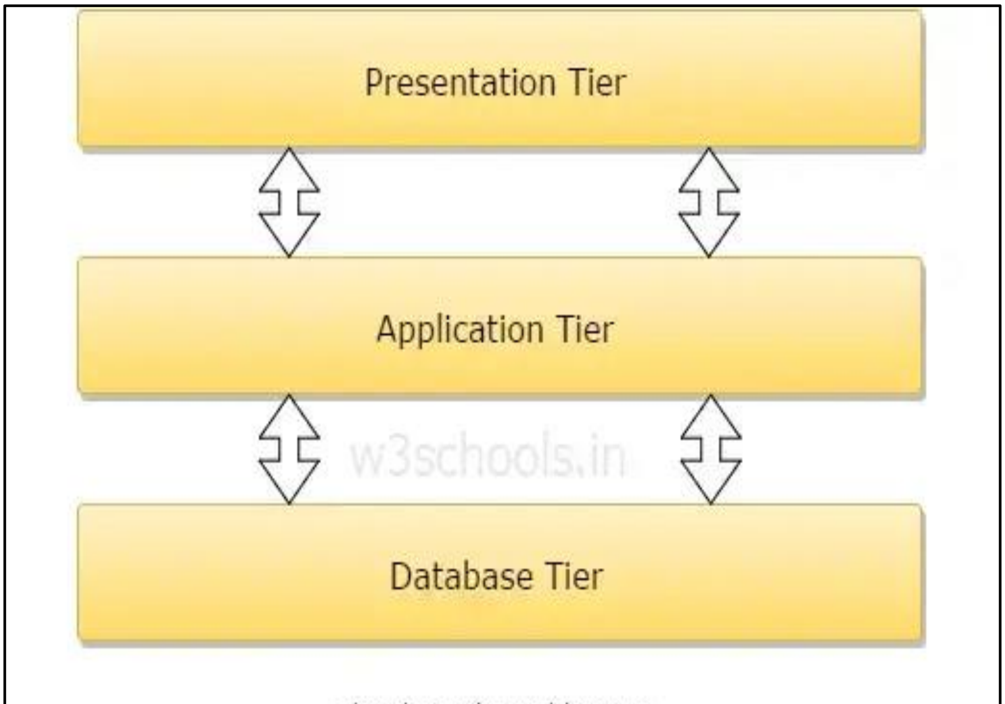
The levels form a three-level architecture that includes an external, a conceptual, and an internal level. The way users recognize the data is called the external level. The way the DBMS and the operating system distinguish the data is the internal level, where the data is stored using the data structures and file. The conceptual level offers both the mapping and the desired independence between the external and internal levels.

A DBMS architecture is depending on its design and can be of the following types:

- Centralized
- Decentralized
- Hierarchical

DBMS architecture can be seen as either a single-tier or multi-tier. An architecture having n-tier splits the entire system into related but independent n modules that can be independently customized, changed, altered, or replaced.

The architecture of a database system is very much influenced by the primary computer system on which the database system runs. Database systems can be centralized, or client-server, where one server machine executes work on behalf of multiple client machines. Database systems can also be designed to exploit parallel computer architectures. Distributed databases span multiple geographically separated machines.



The Three Tier Architecture

A 3-tier application is an application program that is structured into three major parts; each of them is distributed to a different place or places in a network. These three divisions are as follows:

- ✓ The workstation or presentation layer
- ✓ The business or application logic layer
- ✓ The database and programming related to managing layer

1.3.3 Types of Databases

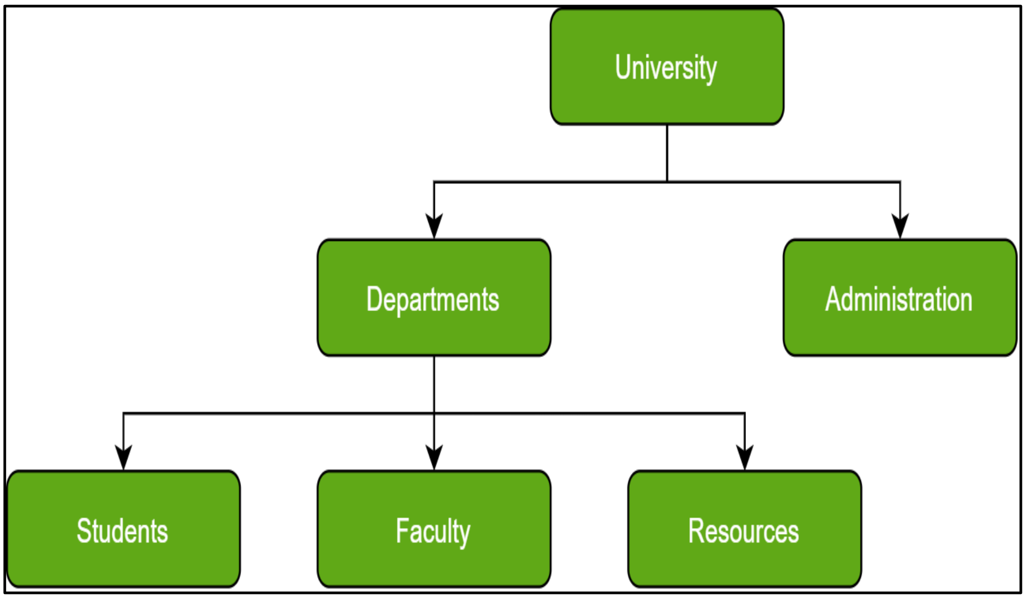
There are several types of databases, that are briefly explained below.

- o Hierarchical databases
- o Network databases

- Object-oriented databases
- Relational databases
- Cloud Database
- Centralized Database
- Operational Database
- NoSQL databases

- ***Hierarchical Databases***

Just as in any hierarchy, this database follows the progression of data being categorized in ranks or levels, wherein data is categorized based on a common point of linkage. As a result, two entities of data will be lower in rank and the commonality would assume a higher rank. Refer to the diagram below :

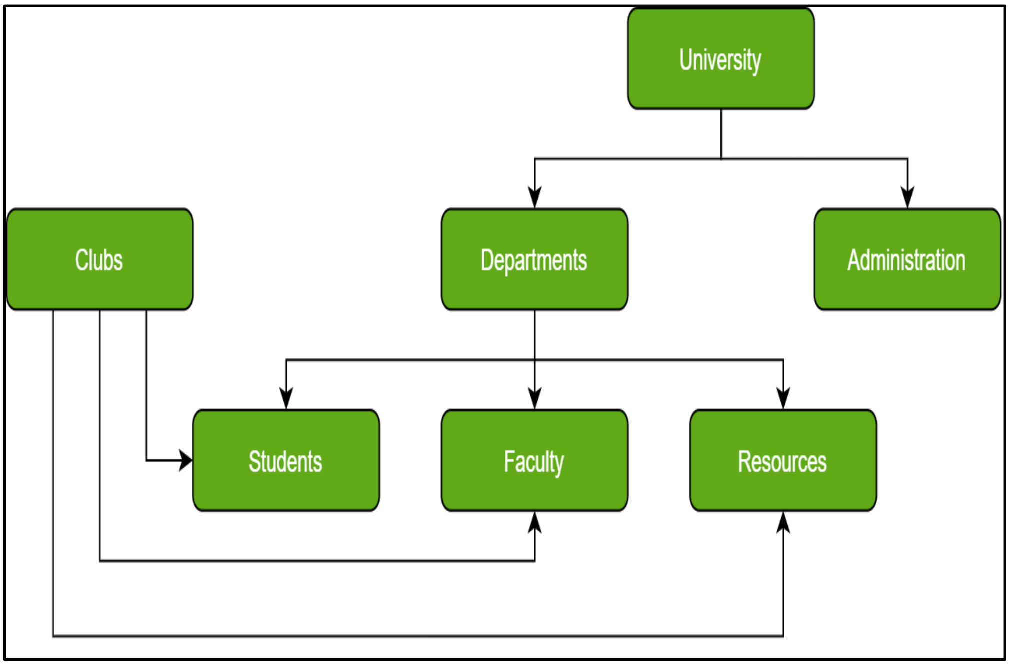


Do note how Departments and Administration are entirely unlike each other and yet fall under the domain of a University. They are elements that form this hierarchy .

Another perspective advises visualizing the data being organized in a parent-child relationship, which upon addition of multiple data elements would resemble a tree. The child records are linked to the parent record using a field, and so the parent record is allowed multiple child records. However, vice versa is not possible .

Notice that due to such a structure, hierarchical databases are not easily salable; the addition of data elements requires a lengthy traversal through the database .

○ ***Network Databases***



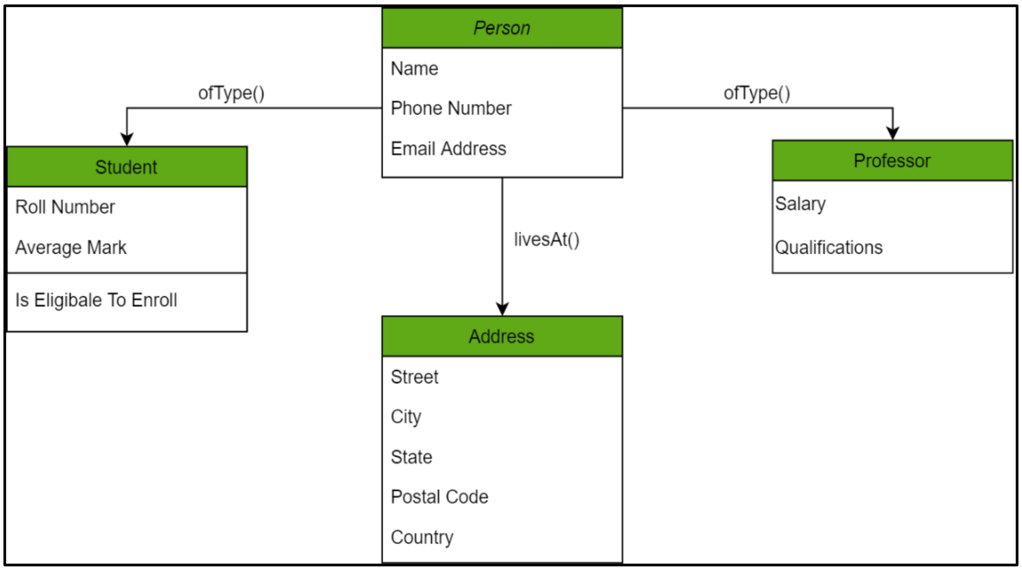
In Layman’s terms, a network database is a hierarchical database, but with a major tweak. The child records are given the freedom to associate with multiple parent records. As a result, a network or net of database files linked with multiple threads is observed. Notice how the Student, Faculty, and Resources elements each have two-parent records, which are Departments and Clubs .

Certainly, a complex framework, network databases are more capable of representing two-directional relationships. Also, conceptual simplicity favors the utilization of a simpler database management language .

The disadvantage lies in the inability to alter the structure due to its complexity and also in it being highly structurally dependent .

○ ***Object-Oriented Databases***

Those familiar with the Object-Oriented Programming Paradigm would be able to relate to this model of databases easily. Information stored in a database is capable of being represented as an object which response as an instance of the database model. Therefore, the object can be referenced and called without any difficulty. As a result, the workload on the database is substantially reduced .



In the chart above, we have different objects linked to one another using methods; one can get the address of the Person (represented by the Person Object) using the livesAt() method. Furthermore, these objects have attributes which are in fact the data elements that need to be defined in the database .

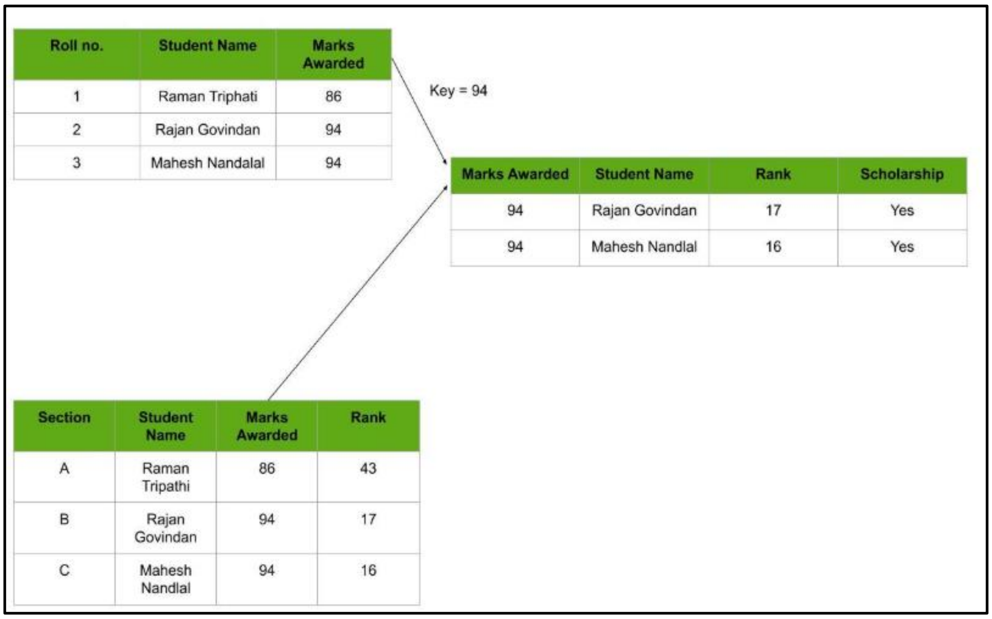
An example of such a model is the Berkeley DB software library which uses the same conceptual background to deliver quick and highly efficient responses to database queries from the embedded database .

○ ***Relational Databases***

Considered the most mature of all databases, these databases lead in the production line along with their management systems. In this database, every piece of information has a relationship with every other piece of information. This is on account of every data value in the database having a unique identity in the form of a record .

Note that all data is tabulated in this model. Therefore, every row of data in the database is linked with another row using a primary key. Similarly, every table is linked with another table using a foreign key .

Refer to the diagram below and notice how the concept of 'Keys' is used to link two tables .



Roll no.	Student Name	Marks Awarded
1	Raman Triphati	86
2	Rajan Govindan	94
3	Mahesh Nandalal	94

Marks Awarded	Student Name	Rank	Scholarship
94	Rajan Govindan	17	Yes
94	Mahesh Nandlal	16	Yes

Section	Student Name	Marks Awarded	Rank
A	Raman Tripathi	86	43
B	Rajan Govindan	94	17
C	Mahesh Nandlal	94	16

Due to this introduction of tables to organize data, it has become exceedingly popular. In consequence, they are widely integrated into Web-Ap interfaces to serve as ideal repositories for user data. What makes it further interesting is the ease in mastering it, since the language used to interact with the database is simple (SQL in this case) and easy to comprehend.

It is also worth being aware of the fact that in Relational databases, scaling and traversing through data is quite a light-weighted task in comparison to Hierarchical Databases .

○ ***Cloud Databases***

A cloud database is used where data requires a virtual environment for storing and executing over the cloud platforms and there are so many cloud computing services for accessing the data from the databases (like SaaS, Paas, etc).

There are some names of cloud platforms are-

- *Amazon Web Services (AWS)*
- *Google Cloud Platform (GCP)*
- *Microsoft Azure*
- *ScienceSoft, etc.*

○ ***Centralized Databases***

A centralized database is basically a type of database that is stored, located as well as maintained at a single location and it is more secure when the user wants to fetch the data from the Centralized Database.

Advantages

- Data Security

- Reduced Redundancy
- Consistency

Disadvantages

- The size of the centralized database is large which increases the response and retrieval time.
- It is not easy to modify, delete and update.

○ ***Personal Databases***

Collecting and Storing the data on its own System and this type of databases is basically designed for the single user.

Advantages

- It is easy to handle
- It occupies less space

○ ***Operational Databases***

It is used for creating, updating, and deleting the database in real-time and it is basically designed for executing and handling the daily data operation in organizations and businesses purposes.

Advantages

- easy to fetch.
- Structured data
- Real-time processing

○ ***NoSQL Databases***

A NoSQL originally referring to non-SQL or non-relational is a database that provides a mechanism for storage and retrieval of data. This data is modeled in means other than the tabular relations used in relational databases .

A NoSQL database includes simplicity of design, simpler horizontal scaling to clusters of machines, and finer control over availability. The data structures used by NoSQL databases are different from those used by default in relational databases which makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it should solve. Data structures used by NoSQL databases are sometimes also viewed as more flexible than relational database tables .

MongoDB falls in the category of NoSQL document-based database .

Advantages of NoSQL

- There are many advantages of working with NoSQL databases such as MongoDB and Cassandra. The main advantages are high scalability and high availability .

Disadvantages of NoSQL

- NoSQL is an open-source database.
- GUI is not available
- Backup is a weak point for some NoSQL databases like MongoDB.

- Large document size .

These are but a few types of database structures which represent the fundamental concepts extensively used in the industry. However, as mentioned earlier, clients tend to focus on creating databases which would suit their own needs; to store data in a schema which showcases a variable functionality based on its blueprint. Hence, the scope for development in reference to databases and database management systems is bright.