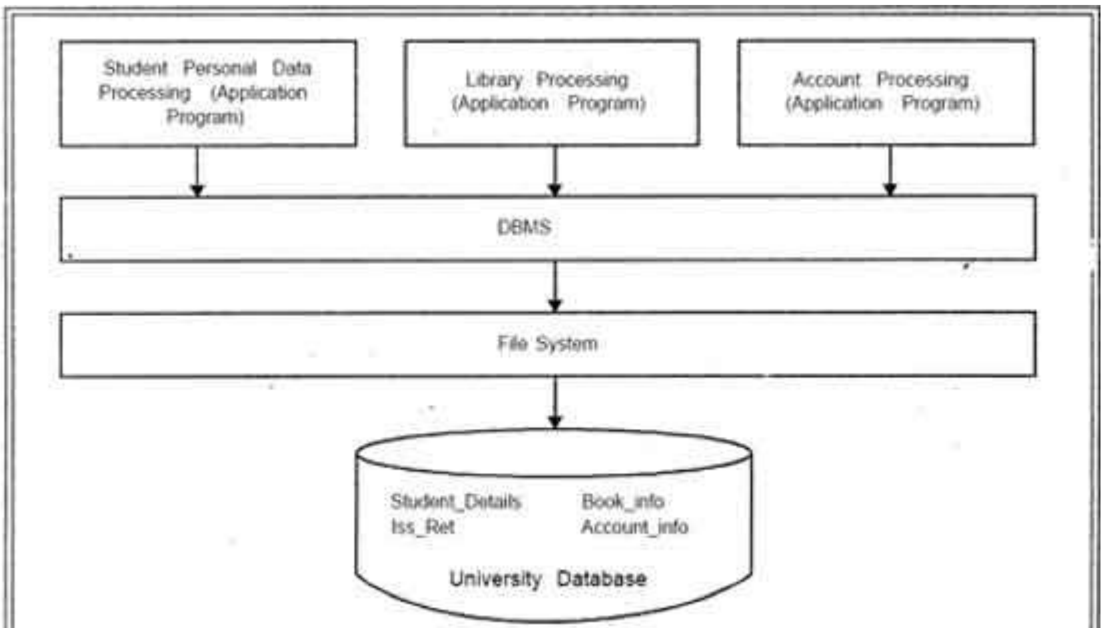
Database Approach

In order to remove all limitations of the File Based Approach, a new approach was required that must be more effective known as Database approach

The Database is a shared collection of logically related data, designed to meet the information needs of an organization. A database is a computer based record keeping system whose over all purpose is to record and maintains information. The database is a single, large repository of data, which can be used simultaneously by many departments and users. Instead of disconnected files with redundant data, all data items are integrated with a minimum amount of duplication.

The database is no longer owned by one department but is a shared corporate resource. The database holds not only the organization's operational data but also a description of this data. For this reason, a database is also defined as a self-describing collection of integrated records. The description of the data is known as the Data Dictionary or Meta Data (the 'data about data'). It is the self-describing nature of a database that provides program-data independence.

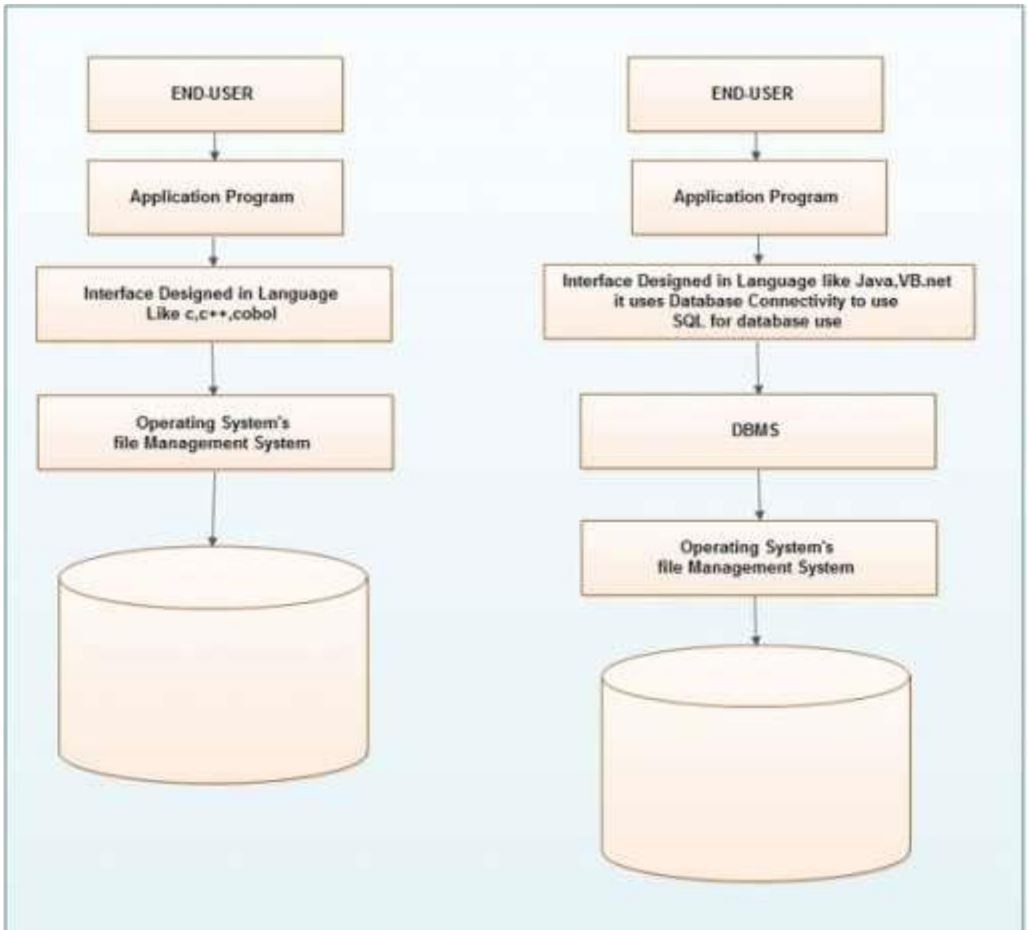


DBMS based implementation of University Management System

A database implies separation of physical storage from use of the data by an application program to achieve program/data independence. Using a database system, the user or programmer or application specialist need not know the details of how the data are stored and such details are "transparent to the user". Changes (or updating) can be made to data without affecting other components of the system. These changes include, for example, change of data format or file structure or relocation from one device to another.

In the DBMS approach, application program written in some programming language like Java, Visual Basic.Net, and Developer 2000 etc. uses database connectivity to access the database stored in the disk with the help of operating system's file management system.

The file system interface and DBMS interface for the university management system is shown.



File System Interface versus DBMS Interface

Building blocks of a Database

The following three components form the building blocks of a database. They store the data that we want to save in our database.

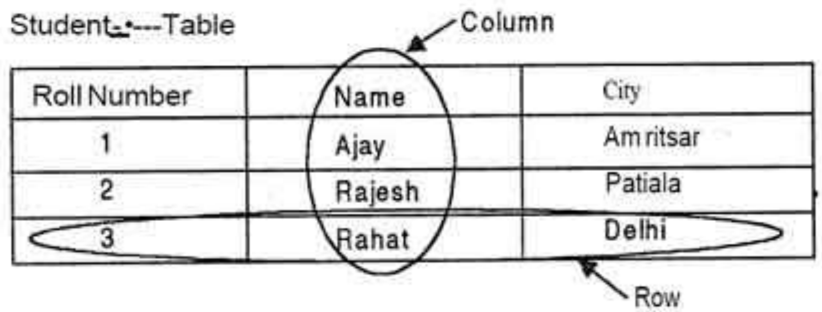
Columns. Columns are similar to fields, that is, individual items of data that we wish to store. A Student' Roll Number, Name, Address etc. are all examples of columns. They are also similar to the columns found in spreadsheets (the A, B, C etc. along the top).

Rows. Rows are similar to records as they contain data of multiple columns (like the 1, 2, 3 etc. in a spreadsheet). A row can be made up of as many or as few columns as you want. This makes reading data much more efficient - you fetch what you want.

Tables. A table is a logical group of columns. For example, you may have a table that stores details of customers' names and addresses. Another table would be used to store details of parts and yet another would be used .for supplier's names and addresses.

It is the tables that make up the entire database and it is important that we do not duplicate data at all.

Example:



Characteristics of database

The data in a database should have the following features:

- **Organized/Related.** It should be well organized and related.
- **Shared.** Data in a database are shared among different users and applications.
- **Permanent or Persistence.** Data in a database exist permanently in the sense the data can live beyond the scope of the process that created it.
- **Validity/integrity/Correctness.** Data should be correct with respect to the real world entity that they represent.
- **Security.** Data should be protected from unauthorized access.
- **Consistency.** Whenever more than one data element in a database represents related real world values, the values should be consistent with respect to the relationship.
- **Non-redundancy:** No two data items in a database should represent the same real world entity.
- **Independence.** Data at different levels should be independent of each other so that the changes in one level should not affect the other levels.
- **Easily Accessible.** It should be available when and where it is needed i.e. it should be easily accessible.
- **Recoverable.** It should be recoverable in case of damage.
- **Flexible to change.** It should be flexible to change.

To create, manage and manipulate data in databases, a management system known as database management system was developed.

Database Languages

A DBMS must provide appropriate languages and interfaces for each category of users to express database queries and updates. Database Languages are used to create and maintain database on computer. There are large numbers of database languages like Oracle, MySQL, MS Access, dBase, FoxPro etc. SQL statements commonly used in Oracle and MS Access can be categorized as data definition language (DDL), data control language (DCL) and data manipulation language (DML).

Data Definition Language (DDL)

It is a language that allows the users to define data and their relationship to other types of data. It is mainly used to create files, databases, data dictionary and tables within databases.

It is also used to specify the structure of each table, set of associated values with each attribute, integrity constraints, security and authorization information for each table and physical storage structure of each table on disk.

The following table gives an overview about usage of DDL statements in SQL

S.No	Need and Usage	The SQL DDL statement
1	Create schema objects	CREATE
2	Alter schema objects	ALTER
3	Delete schema objects	DROP
4	Reneme schema objects	RENAME

Data Manipulation Language (DML)

It is a language that provides a set of operations to support the basic data manipulation operations on the data held in the databases. It allows users to insert, update, delete and retrieve data from the database. The part of DML that involves data retrieval is called a query language.

The following table gives an overview about the usage of DML statements in SQL:

S. No	Need and Usage	The SQL DML statement
1	Remove rows from tables or views	DELETE
2	Add new rows of data into table or view	INSERT
3	Retrieve data from one or more tables	SELECT
4	change column values In existing rows of a table or view	UPDATE

Data Control Language (DCL)

DCL statements control access to data and the database using statements such as GRANT and REVOKE. A privilege can either be granted to a User with the help of GRANT statement. The privileges assigned can be SELECT, ALTER, DELETE, EXECUTE, INSERT, INDEX etc. In addition to granting of privileges, you can also revoke (taken back) it by using REVOKE command.

The following table gives an overview about the usage of DCL statements in SQL:

S. No.	Need And Usage	Age
1	Grant and take away priviliges and roles	Grant Revoke
2	Add a comment to the data dictionary	Comment

In practice, the data definition and data manipulation languages are not two separate languages. Instead they simply form parts of a single database language such as Structured Query Language (SQL). SQL represents combination of DDL and DML, as well as statements for constraints specification and schema evaluation.

What is Database?

A database document is just a collection of information stored in computerized form. The simplest way to understand a database is to think of it like a set of 3 x 5 cards. Since the information is on your computer, though, a dick of the mouse or the stroke of a key can alphabetize those "cards," or find just the names of the people on the cards who live in a certain town, or tell you who owes how much money, and so on.

Computer databases can be highly structured, storing the same kind of information about each item in the database in well-defined compartments. This works as if you printed a standard form on each of your 3 x 5 cards perhaps with one space for a name, one space for an address, and one space for a telephone number. In a structured computer database, the "space" for a name, a part number, a price, is called a field. A record corresponds to one of the individual 3 x 5 cards. The record contains a complete set of fields, all filled with information corresponding to a particular item: if your database is a name-and-address list, each record represents a person; if your database is a parts catalog, each record represents one part.

A specific set of fields and records organized in a specific order, including the information they contain, is called a table. In fact, tables are often displayed on the screen with each item, or record, in a row, and each field as a column.

Structured databases can be either flat file databases or relational databases. In a flat file database, you can work with only one data table-one set of fields-at a time. In a relational database, you can use multiple tables (multiple database documents) at once. Flat file databases are much easier to understand and use, but relational databases are much more efficient for many things you commonly do with data, especially in businesses.

A database can also be simply a free-form collection of information, without any particular structure. In this case, the analogy would be to a pile of notes you've written on whatever paper was handy at the time the information on each piece of paper doesn't have to be organized in the same way.

The term database can also refer to the software package itself that you use to create the database. More often, the software is called a "database program" ("database application" is more specific) or a "database management system" (DBMS).

A database application is one of the most useful tools on the computer, and is actually an incredible amount of fun.

What is a Database Object

A database object in a relational database is a data structure used to either store or reference data. The most common object that most people interact with is the table. Other objects are indexes, stored procedures, sequences, views and many more.

When a database object is created, a new object type cannot be created because all the various object types created are restricted by the very nature, or source code, of the relational database model being used, such as Oracle, SQL Server or Access. What are being created are instances of the objects, such as a new table, an index on that table or a view on the same table.

Most of the major database engines offer the same set of major database object types:

- Tables
- Indexes
- Sequences
- Views
- Synonyms

Who makes this database software?

There are a lot of database software manufacturers out there and a wide range of prices, sizes, speeds and functionalities. At the lower end of the scale are personal database software products like Microsoft Access, which is designed to be used by individuals or small companies relatively little data. User friendliness and ease of use are the priority rather than speed and scalability (in other words, it works well when you have 100 records but not when you have 100,000). At the higher end are full-fledged enterprise solutions, such as Oracle Enterprise Edition. These database software products can handle millions of data entries and are fast and efficient. They have ·many optimization and performance tools and generally require a Database Administrator (DBA) to look after them. Products in this range can also be very expensive.

In the middle are products like Microsoft SQL Server, which is a logical upgrade from Microsoft Access for Windows users. There are also several very good free database software products, such as MySQL and PostgreSQL. These are lacking on the user interface side, but can certainly compete on speed and scalability.

Developments and Evolution of DBMS Concept

We have already seen that the predecessor to the DBMS was the file-based system. However, there was never a time when the database approach began and the file-based system ceased. In fact, the file-based system still exists in specific areas. It has been suggested that the DBMS has its roots in the 1960s Apollo moon-landing project, which was initiated in response to USA's President Kennedy's objective of landing a man on the moon by the end of that decade. At that time there was no system available that would be able to handle and manage the vast amounts of information that the project would generate. As a result, North American Aviation (NAA, now Rockwell International), the prime contractor for the project, developed software known as GUAM (Generalized Update Access Method). GUAM was based on the concept that smaller components come together as parts of larger components, and so on, until the final product is assembled. This structure, which confirms to an upside down tree, is also known as a hierarchical structure.

In the mid 1960s, IBM joined NAA to develop GUAM into what is now known as IMS (Information Management System). The reason why IBM restricted IMS to the management of hierarchies of records was to allow the use of serial storage devices, most notably magnetic tape, which was a market requirement at that time. This restriction was subsequently dropped. Although one of the earliest commercial DBMS, IMS is still main hierarchical DBMS used by most large mainframe installations.

In the mid-1960s, another significant development was the emergency of IDS (Integrated Data Store) from General Electric. This work was headed by one of the early pioneers of database systems, Charles Bachmann. This development led to a new type of database system known as the network DBMS, which had a profound effect on the information systems of that generation. The network database was developed partly to address the need to represent more complex data relationships that could be modeled with hierarchical structures, and partly to impose a database standard. To help establish such standards, the Conference on Data Systems Languages (CODASYL), comprising representatives of the US government and the world of business and commerce formed a List Processing Task Force in 1965, subsequently renamed the Data Base Task Group (DBTG) in 1967. The terms of reference for the DBTG were to define standard specifications for an environment that would allow database creation and data manipulation. A draft report was issued in 1969 and the first definitive report in 1971.

Although, the report, was not formally adopted by the American National Standards Institute (ANSI), a number of systems were subsequently developed following the DBTG proposal. These systems are now known as CODASYL or DBTG systems. The CODASYL and hierarchical approaches represented the first-generation of DBMSs.

In 1970 E.F. Codd of the IBM Research Laboratory produced his highly influential paper on the relational data model. This paper was very timely and addressed the disadvantages of the former approaches. Many experimental relational DBMS's were implemented there after, with the first commercial products appearing in the late 1970s and early 1980s. Of particular note is the System R project at IBM's San Jose Research Laboratory in California, which was developed during the late 1970s.This project was designed to prove the practicality of the relational model by providing an implementation of its data structures and operations, and led to two major developments:

• The development 'of a structure query language called SQL, which has since become the standard language for relational DBMS's.

• The production of various commercial relational DBMS products during the 1980s, for example DB2 and SQL/DS from IBM and Oracle Corporation.

Now there are several hundred relational DBMSs for both mainframe and PC environments, though many are stretching the definition of the relational model. Other examples of multi-user relational DBMSs are INGRES-II from Computer Associates, and Informix Software, Inc. Examples of PC-based relational l)BMSs are Access and FoxPro from Microsoft, Paradox from Corel Corporation, InterBase and BDE from Borland, and R:Base from R:Base Technologies. Relational DBl\1Ss are referred to as second generation DBMSs

The relational model is not without its failures, and in particular its limited modeling capabilities. There has been much research since then attempting to address this problem. In 1976, Chen presented the Entity-Relationship model, which are now a widely accepted technique for database design and the basis for the methodology.

In 1979, Codd himself attempted to address some of the failures in-his original work with an extended version of the relational model called RM/T (1979) and subsequently RM/V2 (1990).The attempts to provide a data model that represents the 'real world' more closely have been loosely classified as semantic data modeling.

In response to the increasing complexity of database applications, two new systems have emerged: the Object Oriented DBMS (OODBMS) and the Object-Relational DBMS (ORDBMS). This evolution represents third-generation DBMSs

Components of DBMS

A database management system (DBMS) consists of several components. Each component plays very important role in the database management system environment. The major components of database management system are:

- Software
- Hardware
- Data
- Procedures
- Database Access Language

Software

The main component of a DBMS is the software. It is the set of programs used to handle the database and to control and manage the overall computerized database

- DBMS software itself, is the most important software component in the overall system
- Operating system including network software being used in network, to share the data of database among multiple users.
- Application programs developed in programming languages such as C++, Visual Basic that are used to to access database in database management system. Each program contains statements that request the DBMS to perform operation on database. The operations may include retrieving, updating, deleting data etc. The application program may be conventional or online workstations or terminals.

Hardware

Hardware consists of a set of physical electronic devices such as computers (together with associated I/O devices like disk drives), storage devices, I/O channels, electromechanical devices that make interface between computers and the real world systems etc, and so on. It is impossible to implement the DBMS without the hardware devices, In a network, a powerful computer with high data processing speed and a storage device with large storage capacity is required as database server.

Data

Data is the most important component of the DBMS. The main purpose of DBMS is to process the data. In DBMS, databases are defined, constructed and then data is stored, updated and retrieved to and from the databases. The database contains both the actual (or operational) data and the metadata (data about data or description about data).

Procedures

Procedures refer to the instructions and rules that help to design the database and to use the DBMS. The users that operate and manage the DBMS require documented procedures on hot use or run the database management system. These may include.

1. Procedure to install the new DBMS.
2. To log on to the DBMS.
3. To use the DBMS or application program.
4. To make backup copies of database.
5. To change the structure of database.
6. To generate the reports of data retrieved from database.

Database Access Language

The database access language is used to access the data to and from the database. The users use the database access language to enter new data, change the existing data in database and to retrieve required data from databases. The user write a set of appropriate commands in a database access language and submits these to the DBMS. The DBMS translates the user commands and sends it to a specific part of the DBMS called the Database Jet Engine. The database engine generates a set of results according to the commands submitted by user, converts these into a user readable form called an Inquiry Report and then displays them on the screen. The administrators may also use the database access language to create and maintain the databases.

The most popular database access language is SQL (Structured Query Language). Relational databases are required to have a database query language.

Users

The users are the people who manage the databases and perform different operations on the databases in the database system.There are three kinds of people who play different roles in database system

1. Application Programmers

2. Database Administrators
3. End-Users

Application Programmers

The people who write application programs in programming languages (such as Visual Basic, Java, or C++) to interact with databases are called Application Programmer.

Database Administrators

A person who is responsible for managing the overall database management system is called database administrator or simply DBA.

End-Users

The end-users are the people who interact with database management system to perform different operations on database such as retrieving, updating, inserting, deleting data etc.

Database technology for Geospatial Data

Spatial Database:

A **spatial database** is a database that is optimized to store and query data that represents objects defined in a geometric space. Most spatial databases allow representing simple geometric objects such as points, lines and polygons. Some spatial databases handle more complex structures such as 3D objects, topological coverages, linear networks, and TINs. While typical databases have developed to manage various numeric and character types of data, such databases require additional functionality to process spatial data types efficiently, and developers have often added *geometry* or *feature* data types.

The Open Geospatial Consortium developed the Simple Features specification (first released in 1997) and sets standards for adding spatial functionality to database systems.

The *SQL/MM Spatial* ISO/EIC standard is a part the SQL/MM multimedia standard and extends the Simple Features standard with data types that support circular interpolations.

Geodatabase:

A geodatabase (also geographical database and geospatial database) is a database of geographic data, such as countries, administrative divisions, cities, and related information. Such databases can be useful for websites that wish to identify the locations of their visitors for customization purposes.

Features of Spatial Database:

Database systems use indexes to quickly look up values and the way that most databases index data is not optimal for spatial queries. Instead, spatial databases use a spatial index to speed up database operations.

In addition to typical SQL queries such as SELECT statements, spatial databases can perform a wide variety of spatial operations. The following operations and many more are specified by the Open Geospatial Consortium standard:

- Spatial Measurements: Computes line length, polygon area, the distance between geometries, etc.
- Spatial Functions: Modify existing features to create new ones, for example by providing a buffer around them, intersecting features, etc.
- Spatial Predicates: Allows true/false queries about spatial relationships between geometries. Examples include "do two polygons overlap" or 'is there a residence located within a mile of the area we are planning to build the landfill?
- Geometry Constructors: Creates new geometries, usually by specifying the vertices (points or nodes) which define the shape.
- Observer Functions: Queries which return specific information about a feature such as the location of the center of a circle