# Database Management Systems

# UNIT-I

**Introduction to DBMS**

Data are simply facts or figures, bits of information. When data are processed, interpreted, organized, structured or presented so as to make them meaningful or useful, they are called information. Information provides context for data.

Data usually refers to raw data, or unprocessed data. It is the basic form of data, data that hasn’t been analysed or processed in any manner. Once the data is analysed, it is considered as information. Information is "knowledge communicated or received concern- ing a particular fact or circumstance." Information is a sequence of symbols that can be interpreted as a message. It provides knowledge or insight about a certain matter.

Some differences between data and information:

* Data is used as input for the computer system. Information is the output of data.
* Data is unprocessed facts figures. Information is processed data.
* Data doesn’t depend on Information. Information depends on data.
* Data is not specific. Information is specific.
* Data is a single unit. A group of data which carries news and meaning is called Information.
* Data doesn’t carry a meaning. Information must carry a logical meaning.
* Data is the raw material. Information is the product.

## Database and Database Management System

Database

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc. Using the database, you can easily retrieve, insert, and delete the infor- mation.

Database Management System

A DBMS is a collection of inter related data and a set of programs to manipulate those data. Database management system is software which is used to manage the database.

DBMS = Database + set of programs

A database management system (DBMS) is a computerized system that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

An application program accesses the database by sending queries or requests for data to the DBMS. A query typically causes some data to be retrieved; a transaction may cause some data to be read and some data to be written into the database. Other important functions provided by the DBMS include protecting the database and maintaining it over a long period of time. Protection includes system protection against hardware or software malfunction (or crashes) and security protection against unauthorized or malicious access. A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.

There are many different types of database management systems, ranging from small systems that run on personal computers to huge systems that run on mainframes.

DBMS applications

There are different fields where a database management system is utilized. Following are a few applications which utilize the information base administration framework –

* Railway Reservation System
* Library Management System
* Banking
* Education Sector
* Credit card exchanges
* Social Media Sites
* Broadcast communications
* Account
* Online Shopping
* Human Resource Management
* Manufacturing

Examples of DBMS

* IBM DB2
* Microsoft Access
* Mango DB
* Microsoft SQL Server
* MySQL
* Oracle RDBMS
  + 1. **Function of DBMS**
       1. Defining database schema: it must give facility for defining the database structure also specifies access rights to authorized users
       2. Manipulation of the database: The dbms must have functions like insertion of record into database updation of data, deletion of data, retrieval of data
       3. Sharing of database: The DBMS must share data items for multiple users by main- taining consistency of data
       4. Protection of database: It must protect the database against unauthorized users
       5. Database recovery: If for any reason the system fails DBMS must facilitate data base recovery

##### File oriented approach

The traditional file oriented approach to information processing has for each application a separate master file and its own set of personal file. In file oriented approach the program dependent on the files and files become dependent on the files and files become dependents upon the programs.

Disadvantages of file oriented approach

* + - 1. **Data redundancy and inconsistency:**

The same information may be written in several files. This redundancy leads to higher storage and access cost. It may lead data inconsistency that is the various copies of the same data may longer agree for example a changed customer address may be reflected in single file but not elsewhere in the system.

* + - 1. Difficulty in accessing data:

The conventional file processing system does not allow data to retrieve in a conve - nient and efficient manner according to user choice.

* + - 1. Data isolation:

Because data are scattered in various file and files may be in different formats with new application programs to retrieve the appropriate data is difficult.

* + - 1. Integrity Problems:

Developers enforce data validation in the system by adding appropriate code in the various application program. However when new constraints are added, it is difficult to change the programs to enforce them.

* + - 1. Atomicity:

It is difficult to ensure atomicity in a file processing system when transaction fail - ure occurs due to power failure, networking problems etc. (Atomicity: either all operations of the transaction are reflected properly in the database or non are)

* + - 1. Concurrent access:

In the file processing system it is not possible to access a same file for transaction at same time.

* + - 1. Security problems:

There is no security provided in file processing system to secure the data from unauthorized user access.

##### Difference between File System and DBMS

|  |  |  |
| --- | --- | --- |
| **Basis** | **File System** | **DBMS** |
| Structure | File system is a software that manages and organizes the files in a storage medium within  a computer. | DBMS is a software for managing the database. |
| Data  Redundancy | Redundant data can be  present in a file system. | In DBMS there is no  redundant data. |
| Backup and Recovery | It doesn’t provide backup and recovery of data if it is lost. | It provides backup and recovery of data even  if it is lost. |
| Query  processing | There is no efficient query  processing in file system. | Efficient query processing  is there in DBMS. |
| Consistency | There is less data consistency in file system. | There is more data consistency because of the  process of normalization |
| Complexity | It is less complex as compared to DBMS. | It has more complexity in handling as compared  to file system. |
| Security Constraints | File systems provide less security in comparison to DBMS. | DBMS has more security  mechanisms as compared to file system. |
| Cost | It is less expensive  than DBMS. | It has a comparatively  higher cost than a file system. |

Table 1.1: Difference between File System and DBMS.

Advantages of DBMS:

* + - 1. **Data Independence**

Data independence refers to characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

* + - 1. Reducing Data Redundancy

The file based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which lead to data redundancy. This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

* + - 1. Sharing of Data

In a database, the users of the database can share the data among themselves. There are various levels of authorisation to access the data, and consequently the data can only be shared based on the correct authorisation protocols being followed. Many remote users can also access the database simultaneously and share the data between themselves.

* + - 1. Data Security

Data Security is vital concept in a database. Only authorised users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorised users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

* + - 1. Privacy

The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

* + - 1. Backup and Recovery

Database Management System automatically takes care of backup and recovery. The users don’t need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

* + - 1. Data Consistency

Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

Disadvantage of DBMS:

1. DBMS software and hardware (networking installation) cost is high
2. The processing overhead by the dbms for implementation of security, integrity and

sharing of the data.

1. Centralized database control
2. Setup of the database system requires more knowledge, money, skills, and time.
3. The complexity of the database may result in poor performance.

## Queries in DBMS

A query is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a query language. A DBMS provides a specialized language, called the query language, in which queries can be posed. A very attractive feature of the relational model is that it supports powerful query languages.

Different types of queries in DBMS are:

* Data definition language queries (DDL)
* Data manipulation language queries (DML)
* Data control language queries (DCL)
* Transaction language queries (TCL)

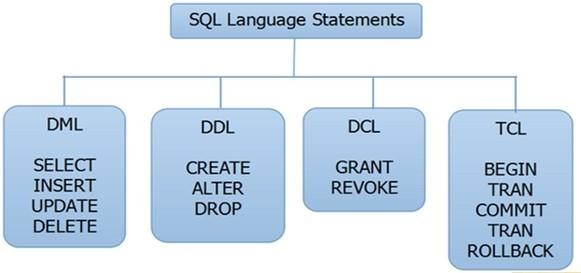


Figure 1.1: DBMS Queries

## Transaction Management

A transaction is a set of logically related operations. For example, you are transferring money from your bank account to your friend’s account, the set of operations would be:

* Read your account balance
* Deduct the amount from your balance
* Write the remaining balance to your account
* Read your friend’s account balance
* Add the amount to his account balance
* Write the new updated balance to his account

This whole set of operations can be called a transaction. The main problem that can happen during a transaction is that the transaction can fail before finishing the all the operations in the set. This can happen due to power failure, system crash etc. This is a serious problem that can leave database in an inconsistent state. Assume that transaction fail after third operation (see the example above) then the amount would be deducted from your account but your friend will not receive it. To solve this problem, we have the following two operations:

1. **Commit:** If all the operations in a transaction are completed successfully then commit those changes to the database permanently.
2. **Rollback:** If any of the operation fails then rollback all the changes done by previous operations.

Even though these operations can help us avoiding several issues that may arise during transaction but they are not sufficient when two transactions are running concurrently. To handle those problems the database system maintains the ACID properties.

1. **Atomicity:** This property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in a database where a transaction is left partially completed.
2. **Consistency:** A transaction enforces consistency in the system state by ensuring that at the end of any transaction the system is in a valid state.
3. **Isolation:** For every pair of transactions, one transaction should start execution only when the other finished execution. I have already discussed the example of Isolation in the Consistency property above.
4. **Durability:** Once a transaction completes successfully, the changes it has made into the database should be permanent even if there is a system failure. The recovery- management component of database systems ensures the durability of transaction



Figure 1.2: Transaction state diagram

1. **Active:** In this state, the transaction is being executed. This is the initial state of every transaction.
2. **Partially Committed:** When a transaction executes its final operation, it is said to be in a partially committed state.
3. **Failed:** A transaction is said to be in a failed state if any of the checks made by the database recovery system fails. A failed transaction can no longer proceed further.
4. **Aborted:** If any of the checks fails and the transaction has reached a failed state, then the recovery manager rolls back all its write operations on the database to bring the database back to its original state where it was prior to the execution of the transaction. Transactions in this state are called aborted. The database recovery module can select one of the two operations after a transaction aborts
   * Re-start the transaction
   * Kill the transaction
5. **Committed:** If a transaction executes all its operations successfully, it is said to be committed. All its effects are now permanently established on the database system.

## DBMS Structure

* A database system is partitioned into modules that deal with each of the responsi- bilities of the overall system.
* The functional components of a database system can be broadly divided into the storage manager and the query processor components.
* The storage manager is important because databases typically require a large amount of storage space.
* The query processor is important because it helps the database system to simplify and facilitate access to data.
* It is the job of the database system to translate updates and queries written in a non-procedural language, at the logical level, into an efficient sequence of operations at the physical level.

Query Processor

The query processor components include

* DDL interpreter, which interprets DDL statements and records the definitions in the data dictionary.
* DML compiler, which translates DML statements in a query language into an eval - uation plan consisting of low-level instructions that the query evaluation engine understands.
* A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs query optimiza - tion, that is, it picks the lowest cost evaluation plan from among the alternatives.
* Query evaluation engine, which executes low- level instructions generated by the DML compiler

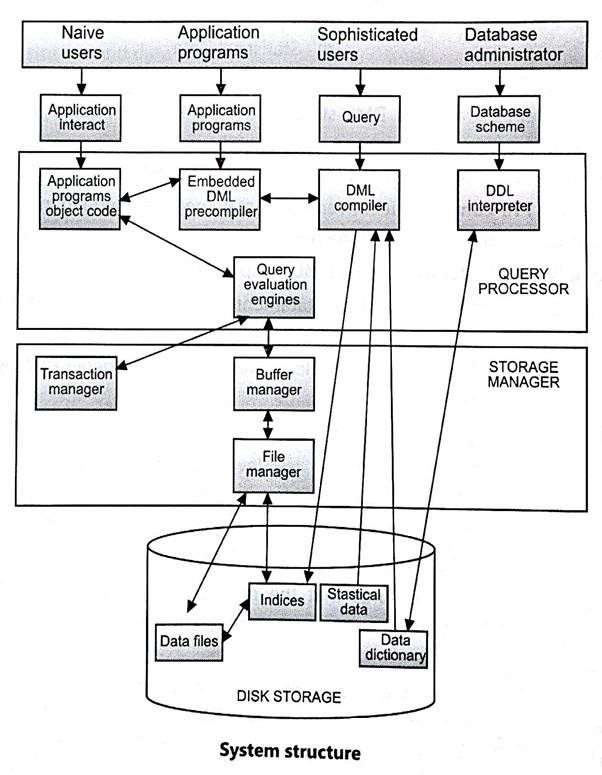


Figure 1.3: Structure of DBMS

* + 1. **Storage Manager**

A storage manager is a program module that provides the interface between the low -level data stored in the database and the application programs and queries submitted to the system. The storage manager is responsible for the interaction with the file manager. The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system. The storage manager translates the various DML state- ments into low- level file-system commands. Thus, the storage manager is responsible for storing, retrieving, and updating data in the database. The storage manager components include:

**Authorization and integrity manager:** which tests for the satisfaction of integrity

constraints and checks the authority of users to access data

**Transaction manager:** which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.

**File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

**Buffer manager**, which is responsible for fetching data from disk storage into main memory, and deciding what data to cache i main memory. The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.

**Transaction Manager**, A transaction is a collection of operations that performs a single logical function in a database application. Each transaction is a unit of both atomicity and consistency. Thus, we require that transactions do not violate any database-consistency constraints. That is, if the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates. Transaction manager ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.

##### Types of Database users

* + - 1. Database Administrator (DBA)

DBA is responsible for:

* + - * + Deciding the instances for the database.
        + Defining the Schema
        + Liaising with Users
        + Define Security
        + Back-up and Recovery
        + Monitoring the performance
      1. Database Designers

Database designers design the appropriate structure for the database, where we share data.

* + - 1. System Analyst

System analyst analyses the requirements of end users, especially naïve and para- metric end users.

* + - 1. Application Programmers

Application programmers are computer professionals, who write application pro- grams.

* + - 1. Naïve Users / Parametric Users

Naïve Users are Un-sophisticated users, which has no knowledge of the database. These users are like a layman, which has a little bit of knowledge of the database. Naive Users are just to work on developed applications and get the desired result. Example: Railway’s ticket booking users are naive users. Or Clerical staff in any

bank is a naïve user because they don’t have any DBMS knowledge but they still use the database and perform their given task.

* + - 1. Sophisticated Users

Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. These users interact with the database but they do not write programs.

* + - 1. Casual Users / Temporary Users

These types of users communicate with the database for a little period of time.

## Storage Data

A database system provides an ultimate view of the stored data. However, data in the form of bits, bytes get stored in different storage devices. In this section, we will take an overview of various types of storage devices that are used for accessing and storing data.

##### Types of Data Storage

For storing the data, there are different types of storage options available. These storage types differ from one another as per the speed and accessibility. There are the following types of storage devices used for storing the data:

* Primary Storage
* Secondary Storage
* Tertiary Storage

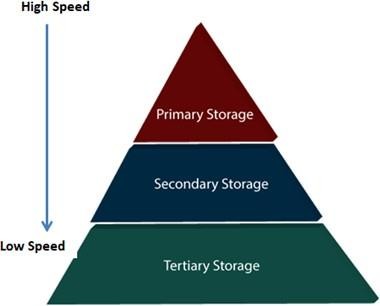


Figure 1.4: Types of data storage

* + - 1. **Primary Storage** It is the primary area that offers quick access to the stored data. We also know the primary storage as volatile storage. It is because this type of memory does not permanently store the data. As soon as the system leads to a power cut or a crash, the data also get lost. Main memory and cache are the types of primary storage.
         * **Main Memory:** It is the one that is responsible for operating the data that is available by the storage medium. The main memory handles each instruction of a computer machine. This type of memory can store gigabytes of data on a system but is small enough to carry the entire database. At last, the main memory loses the whole content if the system shuts down because of power failure or other reasons.
         * **Cache:** It is one of the costly storage media. On the other hand, it is the fastest one. A cache is a tiny storage media which is maintained by the computer hardware usually. While designing the algorithms and query processors for the data structures, the designers keep concern on the cache effects.
      2. **Secondary Storage** Secondary storage is also called as Online storage. It is the storage area that allows the user to save and store data permanently. This type of memory does not lose the data due to any power failure or system crash. That’s why we also call it non-volatile storage.

There are some commonly described secondary storage media which are available in almost every type of computer system:

* + - * + **Flash Memory:** A flash memory stores data in USB (Universal Serial Bus) keys which are further plugged into the USB slots of a computer system. These USB keys help transfer data to a computer system, but it varies in size limits. Unlike the main memory, it is possible to get back the stored data which may be lost due to a power cut or other reasons. This type of memory storage is most commonly used in the server systems for caching the frequently used data. This leads the systems towards high performance and is capable of storing large amounts of databases than the main memory.
        + **Magnetic Disk Storage:** This type of storage media is also known as online storage media. A magnetic disk is used for storing the data for a long time. It is capable of storing an entire database. It is the responsibility of the computer system to make availability of the data from a disk to the main memory for further accessing. Also, if the system performs any operation over the data, the modified data should be written back to the disk. The tremendous capability of a magnetic disk is that it does not affect the data due to a system crash or failure, but a disk failure can easily ruin as well as destroy the stored data.
      1. Tertiary Storage

It is the storage type that is external from the computer system. It has the slowest speed. But it is capable of storing a large amount of data. It is also known as Offline storage. Tertiary storage is generally used for data backup.

There are following tertiary storage devices available:

* + - * + **Optical Storage:** An optical storage can store megabytes or gigabytes of

data. A Compact Disk (CD) can store 700 megabytes of data with a playtime of around 80 minutes. On the other hand, a Digital Video Disk or a DVD can store 4.7 or 8.5 gigabytes of data on each side of the disk.

* + - * + **Tape Storage:** It is the cheapest storage medium than disks. Generally, tapes are used for archiving or backing up the data. It provides slow access to data as it accesses data sequentially from the start. Thus, tape storage is also known as sequential-access storage. Disk storage is known as direct-access storage as we can directly access the data from any location on disk.

##### Storage Hierarchy

Besides the above, various other storage devices reside in the computer system. These storage media are organized on the basis of data accessing speed, cost per unit of data to buy the medium, and by medium’s reliability. Thus, we can create a hierarchy of storage media on the basis of its cost and speed.

Thus, on arranging the above-described storage media in a hierarchy according to its speed and cost, we conclude the below-described image:

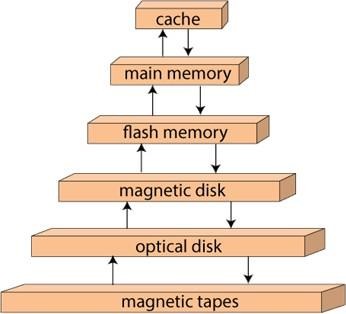


Figure 1.5: Storage device hierarchy

In the Figure. [1.5](#_bookmark0), the higher levels are expensive but fast. On moving down, the cost per bit is decreasing, and the access time is increasing. Also, the storage media from the main memory to up represents the volatile nature, and below the main memory, all are non-volatile devices.

## Data Models in DBMS

A DBMS allows a user to define the data to be stored in terms of a data model. A data model is a collection of high-level data description constructs that hide many low-level storage details. A data model is a collection of concepts that can be used to describe the structure of a database We can categorize data models according to the types of concepts they use to describe the database structure.

* High-level or conceptual data models provide concepts that are close to the way many users perceive data.
* Low-level or physical data models provide concepts that describe the details of how data is stored on the computer storage media, typically magnetic disks.

Concepts provided by physical data models are generally meant for computer spe- cialists, not for end users.

Between these two extremes is a class of Representational or implementation data models, which provide concepts that may be easily understood by end users.

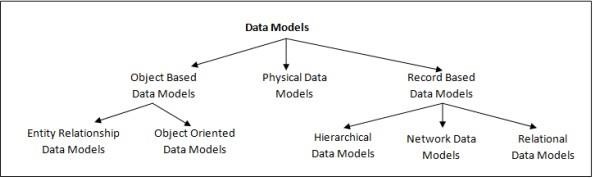


Figure 1.6: Data models classification

##### Entity - Relationship (E-R) Model

Conceptual data models or semantic data model is a more abstract, high-level data model that makes it easier for a user to come up with a good initial description of the data in an enterprise. A database design in terms of a semantic model serves as a useful starting point and is subsequently translated into a database design in terms of the data model the DBMS actually supports. A widely used semantic data model called the entity- relationship (ER) model allows us to pictorially denote entities and the relationships among them. It use concepts such as entities, attributes, and relationships.

An entity represents a real-world object or concept, such as an employee or a project from the world that is described in the database. An attribute represents some property of interest that further describes an entity, such as the employee’s name or salary. A relationship among two or more entities represents an association among the entities, for example,

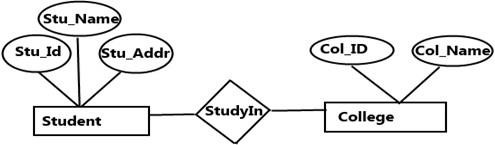


Figure 1.7: Sample E–R Diagram

Entity Relationship Model Advantages:

* Visual modelling yields conceptual simplicity
* Visual representation makes it an effective communication tool
* Is integrated with the dominant relational model

Disadvantages:

* Limited constraint representation
* Limited relationship representation
* No data manipulation language
* Loss of information content occurs when attributes are removed from entities to avoid crowded displays

##### Relational model

In relational database models, three key terms are used extensively: relations, attributes, and domains. A relation is a table with columns and rows. The named columns of the relation are called attributes, and the domain is the set of values the attributes are allowed to take.

Relational Model Advantages

* Structural independence is promoted using independent tables
* Tabular view improves conceptual simplicity
* Adhoc query capability is based on SQL
* Isolates the end user from physical-level details
* Improves implementation and management simplicity

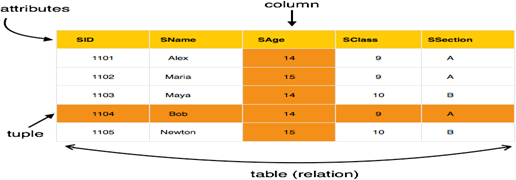


Figure 1.8: Sample Relational model

Disadvantages

* Requires substantial hardware and system software overhead
* Conceptual simplicity gives untrained people the tools to use a good system poorly
* May promote information problems

##### Hierarchical model

In a hierarchical model, data is organized into a tree-like structure, implying a single parent for each record. Hierarchical structures were widely used in the early mainframe database management systems, This structure allows one one-to-many relationship be- tween two types of data. This structure is very efficient to describe many relationships in the real world. The main drawback of this model is that, it can have only one to many relationships between nodes.

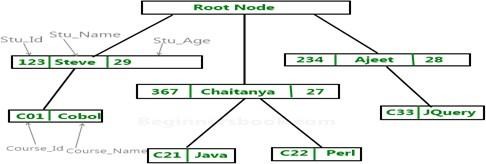


Figure 1.9: Sample Heirarchical model

Hierarchical Model Advantages

* Promotes data sharing
* Parent/child relationship promotes conceptual simplicity and data integrity
* Database security is provided and enforced by DBMS
* Efficient with 1:M relationships

Disadvantages

* Requires knowledge of physical data storage characteristics
* Navigational system requires knowledge of hierarchical path
* Changes in structure require changes in all application programs
* Implementation limitations
* No data definition
* Lack of standards

##### Network model

The network model expands upon the hierarchical structure, allowing many-to-many re- lationships in a tree-like structure that allows multiple parents. A record may be an owner in any number of sets, and a member in any number of sets. It was most popular before being replaced by the relational model. The network model is able to represent redundancy in data more efficiently than in the hierarchical model, and there can be more than one path from an ancestor node to a descendant.

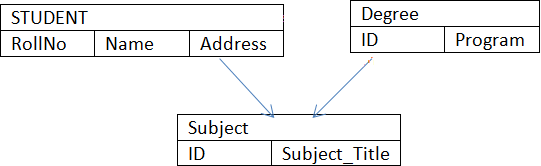


Figure 1.10: Sample Network model

Network Model Advantages

* Conceptual simplicity
* Handles more relationship types
* Data access is flexible
* Data owner/member relationship promotes data integrity
* Conformance to standards
* Includes data definition language (DDL) and data manipulation language (DML)

Disadvantages

* Navigational system yields complex implementation, application development, and management
* Structural changes require changes in all application programs

##### Object oriented data model

Object oriented data models are also frequently utilized as high-level conceptual models, particularly in the software engineering domain. Uses the E-R modelling as a basis but extended to include encapsulation, inheritance.

* Objects have both state and behaviour. State is defined by attributes. Behaviour is defined by methods (functions or procedures)
* Designer defines classes with attributes, methods, and relationships
* Class constructor method creates object instances
* Each object has a unique object ID
* Classes related by class hierarchies

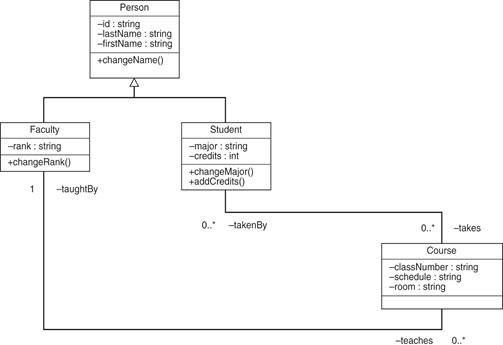


Figure 1.11: Sample Object Oriented model

Advantages

* Semantic content is added
* Visual representation includes semantic content
* Inheritance promotes data Integrity

Disadvantages

* Slow development of standards caused vendors to supply their own enhancements
* Compromised widely accepted standard
* Complex navigational system
* Learning curve is steep
* High system overhead slows transaction

##### Physical data model

Physical data models describe how data is stored as files in the computer by representing information such as record formats, record orderings, and access paths. Physical data model represent the model where it describes how data are stored in computer memory, how they are scattered and ordered in the memory, and how they would be retrieved from memory. Basically physical data model represents the data at data layer or internal layer. It represents each table, their columns and specifications, constraints like primary key, foreign key etc. It basically represents how each tables are built and related to each other in DB.

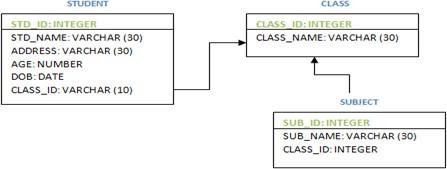


Figure 1.12: Sample Physical model

Figure [1.12](#_bookmark1) shows how physical data model is designed. It is represented as UML diagram along with table and its columns. Primary key is represented at the top. The relationship between the tables is represented by interconnected arrows from table to table. Above STUDENT table is related to CLASS and SUBJECT is related to CLASS. The above diagram depicts CLASS as the parent table and it has 2 child tables – STUDENT and SUBJECT.

Importance of Data Models:

* + - 1. Are a communication tool. Data models can facilitate interaction among the de- signer, the applications programmer, and the end user.
      2. Give an overall view of the database
      3. Organize data for various users
      4. Are an abstraction for the creation of good data base.

##### Evolution of Data Models

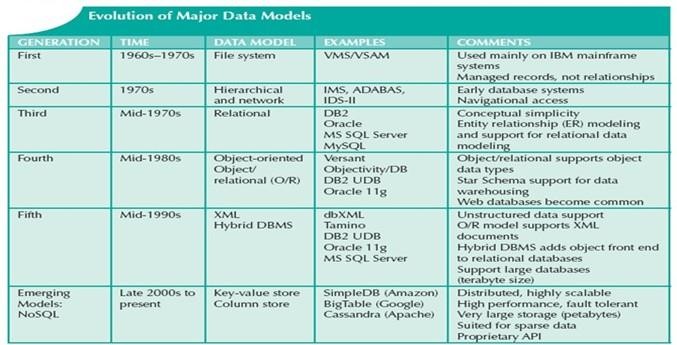


Figure 1.13: Evolution of data models

## Data abstraction

It is a process of hiding unwanted or irrelevant details from the end user. It provides a different view and helps in achieving data independence which is used to enhance the security of data. Mainly there are three levels of abstraction for DBMS, which are as follows:

1. Physical or Internal Level
2. Logical or Conceptual Level
3. View or External Level

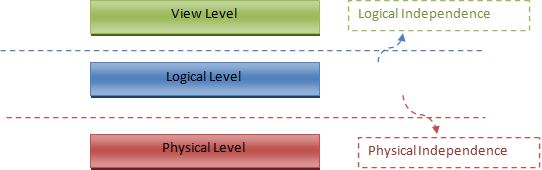


Figure 1.14: Data abstraction

Physical or Internal Level

It is the lowest level of abstraction for DBMS which defines how the data is actually stored, it defines data-structures to store data and access methods used by the database. Actually, it is decided by developers or database application programmers how to store the data in the database.

So, overall, the entire database is described in this level that is physical or internal level. It is a very complex level to understand. For example, customer’s information is stored in tables and data is stored in the form of blocks of storage such as bytes, gigabytes etc.

Logical or Conceptual Level

Logical level is the intermediate level or next higher level. It describes what data is stored in the database and what relationship exists among those data. It tries to describe the entire or whole data because it describes what tables to be created and what are the links among those tables that are created.

It is less complex than the physical level. Logical level is used by developers or database administrators (DBA). So, overall, the logical level contains tables (fields and attributes) and relationships among table attributes.

View or External Level

It is the highest level. In view level, there are different levels of views and every view only defines a part of the entire data. It also simplifies interaction with the user and it provides many views or multiple views of the same database.

View level can be used by all users (all levels’ users). This level is the least complex and easy to understand.

For example, a user can interact with a system using GUI that is view level and can enter details at GUI or screen and the user does not know how data is stored and what data is stored, this detail is hidden from the user.

Internal level or Physical level

It is the lowest level of abstraction and External or View level of abstraction is the highest level of abstraction. Based on these levels of abstraction, we have two types of data independence.

**Physical Data Independence** Physical Data Independence means changing the phys- ical level without affecting the logical level or conceptual level. Using this property, we

can change the storage device of the database without affecting the logical schema. The changes in the physical level may include changes using the following

* A new storage device like magnetic tape, hard disk, etc.
* A new data structure for storage.
* A different data access method or using an alternative files organization technique.
* Changing the location of the database.

Logical Data Independence

Logical view of data is the user view of the data. It presents data in the form that can be accessed by the end users. Logical Data Independence says that users should be able to manipulate the Logical View of data without any information of its physical storage. Software or the computer program is used to manipulate the logical view of the data.

Database administrator is the one who decides what information is to be kept in the database and how to use the logical level of abstraction. It provides the global view of Data. It also describes what data is to be stored in the database along with the relationship. The data independence provides the database in simple structure. It is based on application domain entities to provide the functional requirement. It provides abstraction of system functional requirements. Static structure for the logical view is defined in the class object diagrams. Users cannot manipulate the logical structure of the database.

The changes in the logical level may include:

* Change the data definition.
* Adding, deleting, or updating any new attribute, entity or relationship in the database.