**UNIT 1 IMPORTANT QUESTIONS WITH ANSWERS**

**Two marks**

**1. Define DBMS & RDBMS.**

A **Database management System** consists of a collection interrelated data and set of programs to access those data. The collection of data, usually referred to as the database, contains information about one particular enterprise.

A DBMS is defined as system software that enables to store, modify, manipulate and extract data from a database.

**Relational Database Management system** also allows the organization to access data more efficiently then DBMS.

RDBMS is a software system which is used to store only data which need to be stored in the form of tables. In this kind of system, data is managed and stored in rows and columns which is known as tuples and attributes.

**2. Define Database.**

A database is organized collection of related data of an organization stored in formatted way which is shared by multiple users.

The main feature of data in a database is:

1. It must be well organized

2. It is related

3. It is accessible in a logical order without any difficulty

4. It is stored only once

**3. Write about the role of Transaction manager.**

Transaction manager is responsible for ensuring that the database remains in a consistent state despite system failures. The TM also ensures that concurrent transaction executions proceed without conflict.

**4. Define instance and schema.**

**Instance**: Collection of data stored in the data base at a particular moment is called an Instance of the database.

**Schema**: The overall design of the data base is called the data base schema.

**5. What are storage manager and their components?**

A storage manager is a program module that provides the interface between the low level data stored in a database and the application programs and queries submitted to the system.

The storage manager components include

* Authorization and integrity manager
* Transaction manager
* File manager
* Buffer manager

**5. Define Meta-data.**

A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the DBMS catalogue which is also called meta-data.

**6. Define Data independence.**

The ability to modify a schema definition in only level without affecting a Schema definition in the next higher level is called data independence.

* Physical independence
* Logical independence

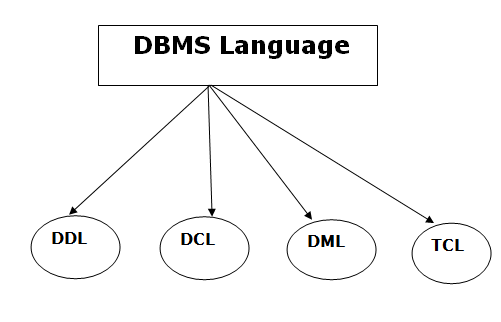
**1. Explain about Database Language.**

**Database Language**

A DBMS has appropriate languages and interfaces to express database queries and updates.

Database languages can be used to read, store and update the data in the database.

**Types of Database Language**



**1. Data Definition Language**

* **DDL** stands for **D**ata **D**efinition **L**anguage. It is used to define database structure or pattern.
* It is used to create schema, tables, indexes, constraints, etc. in the database.
* Using the DDL statements, you can create the skeleton of the database.
* Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Here are some tasks that come under DDL:

**Create:** It is used to create objects in the database.

**Alter:** It is used to alter the structure of the database.

**Drop:** It is used to delete objects from the database.

**Truncate:** It is used to remove all records from a table.

**Rename:** It is used to rename an object.

**Comment:** It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

**2. Data Manipulation Language**

* **DML** stands for **D**ata **M**anipulation **L**anguage. It is used for accessing and manipulating data in a database. It handles user requests.

Here are some tasks that come under DML:

**Select:** It is used to retrieve data from a database.

**Insert:** It is used to insert data into a table.

**Update:** It is used to update existing data within a table.

**Delete:** It is used to delete all records from a table.

**Merge:** It performs UPSERT operation, i.e., insert or update operations.

**Call:** It is used to call a structured query language or a Java subprogram.

**Lock Table:** It controls concurrency.

**3. Data Control Language**

* **DCL** stands for **D**ata **C**ontrol **L**anguage. It is used to retrieve the stored or saved data.
* The DCL execution is transactional. It also has rollback parameters.

(But in Oracle database, the execution of data control language does not have the feature of rolling back.)

Here are some tasks that come under DCL:

**Grant:** It is used to give user access privileges to a database.

**Revoke:** It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke:

CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

**4. Transaction Control Language**

* TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

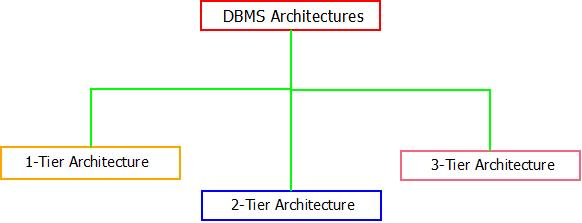
Here are some tasks that come under TCL:

**Commit:** It is used to save the transaction on the database.

**Rollback:** It is used to restore the database to original since the last Commit.

**2. Explain about Database Architecture.**

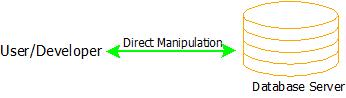
* The DBMS architecture is the foundation of any database management system, which lets the DBMS perform the functions efficiently and effectively. The whole concept of DBMS revolves around its architecture. Depending upon the architecture, database management systems are designed as centralized, decentralized and hierarchical.
* The centralized DBMS design can be correlated with the architecture of a simple computer system such as personal computer or laptop. The decentralized DBMS design can be correlated with client server architecture and hierarchical DBMS can be correlated with n-Tier architecture.
* The DBMS architecture can be classified into three types as per the uses and requirements of the users.



DBMS Architecture: Types

**DBMS Architecture : 1-Tier**

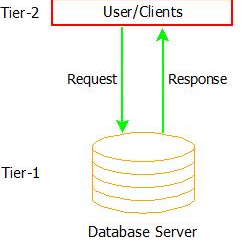
* Tier means “level/layer”. The tier-1 architecture of DBMSs is the simplest among all, which gives the user/developers the ability to let the communicate directly to the database without any intervention of 3rd It also enables the developers/users to make changes, manipulate and manage database directly. This type of architecture is mainly used by developers for their testing and data management purpose.
* Cloud based drives and Personal computer systems can be the possible examples of 1-Tier architecture.



DBMS Architecture: 1-Tier

**DBMS Architecture : 2-Tier**

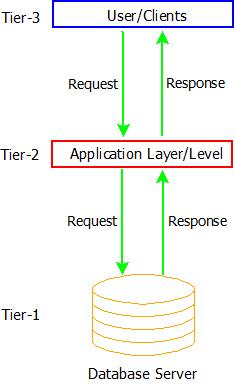
* The 2-Tier architecture of DBMS consists of two tiers. Tier-1 being the database server and Tier-2 being the users or clients of the application.
* In this type of DBMS architecture, the users of the software application deals with the database software and can find their response to the requests made by them. The user cannot manipulate the data inside the database without the permission.
* Client-Server architecture can be a possible example of 2-Tier DBMS architecture.



DBMS Architecture : 2-Tier

**DBMS Architecture : 3-Tier**

* The 3-Tier architecture of DBMS is a fully fledged software system that is responsible for generating response to user queries in the most efficient and suitable manner.
* 3-Tier architecture is the most complex among all three, but solves almost all the issues that occurs in 2-Tier and 1-Tier architecture.
* Security, Data Backup, Recovery, Concurrency Control and Low Data Redundancy are some of the features of a 3-Tier architecture, which makes it the most widely used database architecture.



DBMS Architecture: 3-Tier

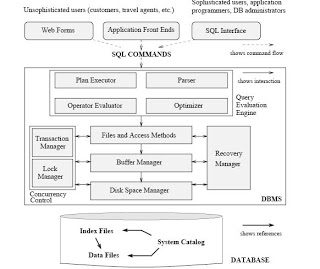
* The levels that are used in 3-Tier DBMS architecture are:
  + **Database Server(Tier-1)** : This tier contains and deals with all the data and information. Also, it guarantees that all the data is stored in a secured manner and there might not occur a situation of data inconsistency or data redundancy.
  + **Application Layer(Tier-2)** : The application layer acts as an intermediate between the User/Client and the actual database. But it ensures to present the abstracted view of the database and provides a way to respond to the queries requested by the user by fetching the response from the database in tier-1.
  + **User/Client Layer(Tier-3)** : This is the top most layer from where users/clients can request data and in response the data is fetched from the database tier and passed on to the user/client tier via the application layer in the most meaningful way possible. It also provides the graphical user interface(GUI) to the users.

**STRUCTURE OF A DBMS:**

When a user issues a query, the parsed query is presented to a query optimizer, which uses information about how the data is stored to produce an effcient execution plan for evaluating the query. An execution plan is a blueprint for evaluating a query, and is usually represented as a tree of relational operators.

The code that implements relational operators sits on top of the le and access methods layer. This layer includes a variety of software for supporting the concept of a le, which, in a DBMS, is a collection of pages or a collection of records. This layer typically supports a heap le, or le of unordered pages, as well as indexes. In addition to keeping track of the pages in a le, this layer organizes the information within a page.The les and access methods layer code sits on top of the buer manager, which brings pages in from disk to main memory as needed in response to read requests.

The lowest layer of the DBMS software deals with management of space on disk, where the data is stored. Higher layers allocate, deallocate, read, and write pages through (routines provided by) this layer, called the disk space manager.   
The DBMS supports concurrency and crash recovery by carefully scheduling user requests and maintaining a log of all changes to the database. DBMS components associated with concurrency control and recovery include the transaction manager, which ensures that transactions request and release locks according to a suitable locking protocol and schedules the execution transactions; the lock manager, which keeps track of requests for locks and grants locks on database objects when they become available; and the recovery manager, which is responsible for maintaining a log, and restoring the system to a consistent state after a crash. The disk space manager, buer manager, and le and access method layers must interact with these components.

****

**3. Write a note on the purpose of Database Systems & Advantages of Database System.**

**PURPOSE OF DATABASE MANAGEMENT SYSTEM**

To see why database management systems are necessary, let's look at a typical ``File-Processing System'' supported by a conventional operating system.

The application is a savings bank :

 Savings account and customer records are kept in permanent system files.

 Application programs are written to manipulate files to perform the following **tasks** :

* + - * Debit or credit an account.
      * Add a new account.
      * Find an account balance.
      * Generate monthly statements.
    1. Development of the System proceeds as follows :

New application programs must be written as the need arises.



 New permanent files are created as required.

 but over a long period of time files may be in different formats, and

 Application programs may be in different languages.

So we can see there are problems with the Straight File-Processing Approach

**1. Data Redundancy**

In previous data management system (file processing system), different programmer creates files and writes application programs to access it. In file system, each application accessed its own private files which cannot be shared between different applications. If two applications needs same file then there must be two files for both application. This caused data redundancy.  
For example, ‘Need For Speed’ game needs ‘direct X 9’ file and ‘counter strike’ game needs it too. Now a days, if we install direct x 9 once then it will be accessed by many programs. This minimized data redundancy.

**2. Data Integrity is Enforced**

Accurate data is maintained in database management system. Some kind of constraints are enforced on the database to maintain integrity. For example, If the data type is numbers (age) in the table, then you cannot enter text.

**3. Inconsistency can be avoided**

Data is accessed and manipulated frequently. If two programs are accessing the same data and one program changed it. Then another programs process with wrong data. Database management should avoid the chances of inconsistency.

**4. Data can be shared**

Data can be shared in centralized database management system. Because of sharing of data, different applications can access the centralized data.

**5. Restriction for unauthorized access**

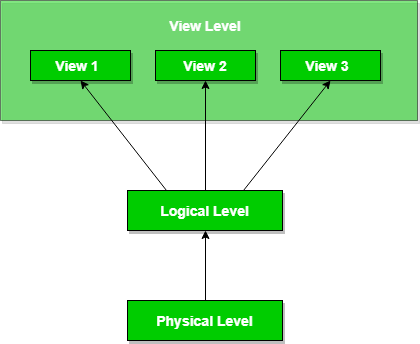
As data can be shared from database, there must be different users. All the users must not have full access to database. For example, If you could access my database then, you can get my password and vital information. You can access only posts and public pages of my website. Database provides access to authorized user only on different level. This increases security of the database.

**6. Provide Backup and Recovery**

As database has vital and important data, there is need of backup and recovery. Backup and recovery must be reliable and possible when needed. Software and hardware failure is often and database backup and recovery is important in case of failure.

**4. Explain about Data Abstraction**

For the system to be usable, it must retrieve data efficiently. The need for efficiency has led designers to use complex data structures to represent data in the database. Developers hide the complexity from users through several levels of abstraction to simplify users interactions with the system.



Database systems comprise of complex data-structures. In order to make the system efficient in terms of retrieval of data, and reduce complexity in terms of usability of users, developers use abstraction i.e. hide irrelevant details from the users. This approach simplifies database design.

There are mainly **3**levels of data abstraction:

**Physical**: This is the lowest level of data abstraction. It tells us how the data is actually stored in memory. The access methods like sequential or random access and file organisation methods like B+ trees, hashing used for the same. Usability, size of memory, and the number of times the records are factors which we need to know while designing the database.  
Suppose we need to store the details of an employee. Blocks of storage and the amount of memory used for these purposes is kept hidden from the user.

The lowest level of abstraction describes **how** the data are actually stored.

**Logical**: This level comprises of the information that is actually stored in the database in the form of tables. It also stores the relationship among the data entities in relatively simple structures. At this level, the information available to the user at the view level is unknown.  
We can store the various attributes of an employee and relationships, e.g. with the manager can also be stored.

The next-higher level of abstraction describes **what** data are stored in the database, and what relationships exist among those data.

**View**: This is the highest level of abstraction. Only a part of the actual database is viewed by the users. This level exists to ease the accessibility of the database by an individual user. Users view data in the form of rows and columns. Tables and relations are used to store data. Multiple views of the same database may exist. Users can just view the data and interact with the database, storage and implementation details are hidden from them.

The view level of abstraction exists to simplify their interaction with the system. The system may provide many views for the same database.

The main purpose of data abstraction is achieving data independence in order to save time and cost required when the database is modified or altered.

**Data Independence**

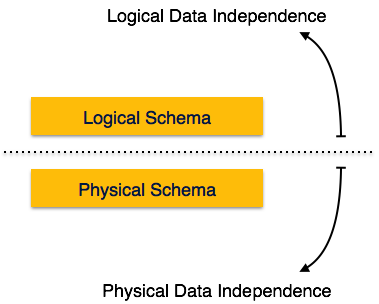
Data Independence can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level.

Data Independence occurs because when the schema is changed at some level, the schema at the next higher level remains unchanged; only the mapping between the two levels is changed.

There are two types of data independence:

1. Logical Data Independence
2. Physical Data Independence

A database system normally contains a lot of data in addition to users’ data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.



Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.

## Logical Data Independence

Logical data is data about database, that is, it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation.

Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

## Physical Data Independence

All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data.

For example, in case we want to change or upgrade the storage system itself − suppose we want to replace hard-disks with SSD − it should not have any impact on the logical data or schemas.

**5. Explain about Database Users and Administrator.**

**Application Programmers** are [computer](http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) professionals interacting with the system through DML calls embedded in a program written in a host language (e.g. C, PL/1, Pascal):

* These programs are called Application Programs.
* The DML Precompiled converts DML calls (prefaced by a special character like $, #, etc.) to normal procedure calls in a host language.
* The host language compiler then generates the object code. Some special types of programming languages combine Pascal-like control structures with control structures for the manipulation of a [database](http://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms).  These are sometimes called Fourth-Generation Languages. They often include features which to generate forms and display data.

• **Sophisticated Users** interact with the system without writing programs :

* They form requests by writing queries in a database query language.
* These are submitted to a **query processor** that breaks a DML statement down into instructions for the database manager module.

• **Specialized Users** are sophisticated users writing special database application programs.

* These may be CADD systems, knowledge-based and expert systems, complex data systems (audio/video), etc.

• **Naive Users** are unsophisticated users who interact with the system by using permanent application programs (e.g. automated teller machine)

### Database Administrators (DBA)

This may be one person or a group of people in an organization responsible for authorizing access to the database, monitoring its use and managing all of the resources to support the use of the entire database system.

**Database Administrators**

The person who has the central control over a database system is called Database Administrator (DBA).

The database administrator has the following functions in a database system.

**-Schema Definition:**The database administrator creates the original database schema by executing a set of data definition statements in DDL.

**-Storage structures an access method definition.**

**-Schema and physical or organization modification:**The database administrator performs the changes to the schema according to the needs of organizations or physical needs to improve the database performance.

**-Provide the granting of authorization to access data:**The database administrator can decide which parts of the database can be accessed by a user, by using the different types of authorization methods.

**-Database maintenance:**The database maintenance includes the following processes.

* Regular backing up of the database.
* Ensuring the disk space for performing the required operations.
* Monitoring the jobs running on the database.

**6. Explain about Application of Database.**

**Applications of DBMS**

Database is widely used. The some of the representative applications are:

**Banking** : for customer information, accounts and loans and banking transactions.

**Universities** : for student registrations and grades.

**Online shopping** : Everyone wants to shop from home. Everyday new products are added and sold only with the help of DBMS. Purchase information, invoice bills and payment, all of these are done with the help of DBMS.

**Airlines** : for reservations and schedule information.

**Credit card transactions**: for purchases on credit cards and generation of monthly statements.

**Library Management System**: maintain all the information relate to book issue dates, name of the book, author and availability of the book.

**Telecommunications**: for keeping records of call made, generating monthly bills, maintaining balances on prepaid calling cards.

**Sales** : for customer, product and purchase information.

**Finance** : for storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds.

**Manufacturing** : for management of supply chain and for tracking production of items in factories, inventories of items and orders for items.

**Human Resource** : for information about employees, salaries, payroll taxes and benefits.

**7. Write about Advantages of DBMS over File system.**

**File System**A File Management system is a DBMS that allows acces to single files or tables at a time. In a File System, data is directly stored in set of files. It contains flat files that have no relation to other files (when only one table is stored in single file, then this file is known as flat file).

**DBMS**A Database Management System (DBMS) is system software that allows users to efficiently define, create, maintain and share databases. Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database. Creating a database involves storing the data on some storage medium that is controlled by DBMS. Maintaining a database involves updating the database whenever required to evolve and reflect changes in the miniworld and also generating reports for each change. Sharing a database involves allowing multiple users to access the database. DBMS also serves as an interface between the database and end users or application programs. It provides control access to the data and ensures that data is consistent and correct by defining rules on them.  
An application program accesses the database by sending queries or requests for data to the DBMS. A query causes some data to be retrieved from database.

**Advantages of DBMS over File system –**

* **Data redundancy and inconsistency –** Redundancy is the concept of repetition of data i.e. each data may have more than a single copy. The file system cannot control redundancy of data as each user defines and maintains the needed files for a specific application to run. There may be a possibility that two users are maintaining same files data for different applications. Hence changes made by one user does not reflect in files used by second users, which leads to inconsistency of data. Whereas DBMS controls redundancy by maintaining a single repository of data that is defined once and is accessed by many users. As there is no or less redundancy, data remains consistent.
* **Data sharing –** File system does not allow sharing of data or sharing is too complex. Whereas in DBMS, data can be shared easily due to centralized system.
* **Data concurrency –** Concurrent access to data means more than one user is accessing the same data at the same time. Anomalies occur when changes made by one user gets lost because of changes made by other user. File system does not provide any procedure to stop anomalies. Whereas DBMS provides a locking system to stop anomalies to occur.
* **Data searching –** For every search operation performed on file system, a different application program has to be written. While DBMS provides inbuilt searching operations. Users only have to write a small query to retrieve data from database.
* **Data integrity –** There may be cases when some constraints need to be applied on the data before inserting it in database. The file system does not provide any procedure to check these constraints automatically. Whereas DBMS maintains data integrity by enforcing user defined constraints on data by itself.
* **Data Independence-** In a database system, the database management system provides the interface between the application programs and the data. When changes are made to the data representation, the metadata obtained by the DBMS is changed but the DBMS is continues to provide the data to application program in the previously used way. The DBMs handles the task of transformation of data wherever necessary.

#### Reduced Application Development and Maintenance Time-DBMS supports many important functions that are common to many applications, accessing data stored in the DBMS, which facilitates the quick development of application.

## Disadvantages of DBMS

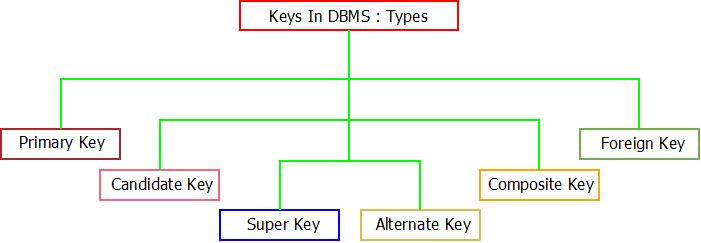
1. Increased Complexity
2. Requirement of New and Specialized Manpowers
3. Large Size of DBMS

**8. Explain about Keys in DBMS.**

**Keys in DBMS: Types**

**A major of six types of keys in DBMS are present. These keys are used according to the type of relation and database available. In some cases, multiple keys are also used in order to link two or more than two tables containing data.**

**The objective of introduction of keys is – They help in relationship identification among tables. All the keys are discussed below along with the examples. Stick to this tutorial and you will surely get the concept of keys.**



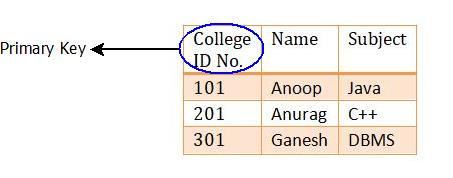
**Keys In DBMS : Types**

**1. Primary Key**

**Primary key is the one which helps in unique identification of each data record present in the tables. It can be a single attribute of the table or combination of attribute.**

**Primary key can also be called as a key which is most suitably chosen from set of candidate keys. Let’ make it this concept of primary key clear with an example.**

**For example : Consider the data of professors of a college.**



**Keys In DBMS : Primary Key Example**

**Here, “College ID No.” can be termed as a primary key because it can uniquely identify all the details of the professors easily. Two professors can have same name that is why name is not a primary key.**

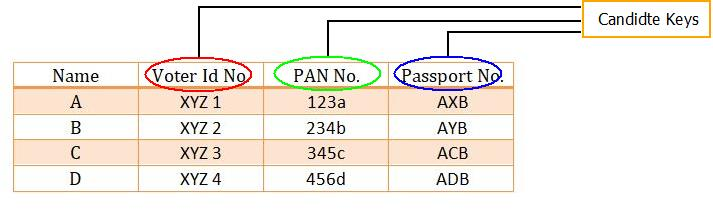
**2. Candidate Key**

**Candidate keys are similar to primary key. The only difference is, primary key for a table can only be one but candidate key can be more than one.**

**Candidate key is also called as minimal super key with each tuple having unique value.**

**The set of multiple keys possible for a relation is called candidate key and when suitable key is selected from that set, it is called primary key.**

**For example : Consider the data of members in a family. If one needs to be identified uniquely, it can be done by the concept of keys.**



**Keys In DBMS : Candidate Key Example**

**Here, Voter Id No, Passport No & PAN No can be termed as candidate keys because each of these can uniquely identify all the records present in the table.**

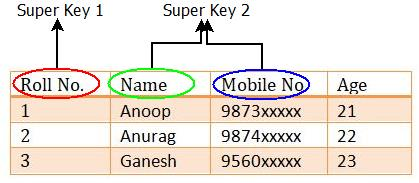
**Note: Out of  Voter Id No, Passport No & PAN No, any of can be selected as a primary key because of their uniqueness.**

**3. Super Key**

**Super key is the one which is able to determine any tuples/data/entity from record table.**

**Super key can be a single attribute or a set of attributes.**

**For example : Consider the data of students in a class.**



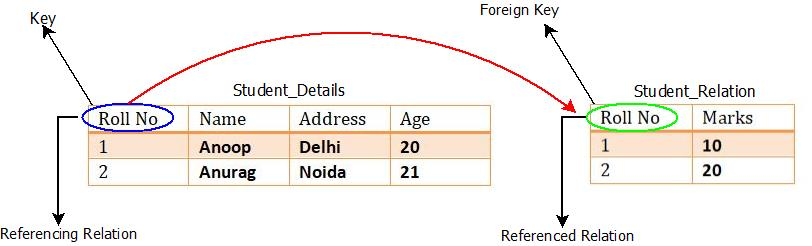
**Keys In DBMS : Super Key Example**

**Here, super key can be a single attribute such as “(Roll No.)” or a collection of multiple attributes such as “(Name, Mobile No)” which can help in identification of record of every student.**

**4. Foreign Key**

**A foreign key is nothing but an attribute that is commonly linked between two relation using that same attribute. Both the relations/tables must contain the same attribute.**

**For example : Consider two relations R1(Roll No, Name, Marks) and R2(Roll No, Address, Age) of students of a school. Roll No being the common attribute will act as a key for relation R1and can reference to Roll No attribute of relation R2with the help of which any tuple can be retrieved from the tables.**



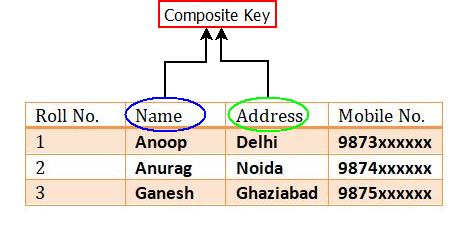
**Keys In DBMS : Foreign Key Example**

**5. Composite Key**

**Composite key is the one having combination of more than one attribute to uniquely determine the records/tuples in a table.**

**The only condition that is necessary before choosing composite key is, the attributes which are responsible for forming the composite key must not be a key individually.**

**For example : Consider the table of student having their details. If we wish to choose a composite key, it can be done as:**



**Keys In DBMS : Composite Key Example**

**6. Alternate Key**

**Candidate keys can be more than one. Out of which only one is chosen as a primary key. The remaining keys which are not selected are called as alternate key.**

**16. Briefly explain about the data models.**

A **database model** is the theoretical foundation of a database and fundamentally determines in which manner data can be stored, organized, and manipulated in a database system. It thereby defines the infrastructure offered by a particular database system. The most popular example of a database model is the relational model.

**types of data model used**

* + Hierarchical model
  + Network model
  + Relational model
  + Entity-relationship
  + Object-relational model
  + Object model
  + Semi-structured Model

**1. Hierarchical Model**

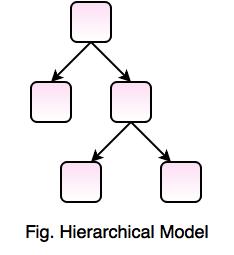
Hierarchical model was developed by IBM and North American Rockwell known as Information Management System.

It represents the data in a hierarchical tree structure.

This model is the first DBMS model.

In this model, the data is sorted hierarchically.

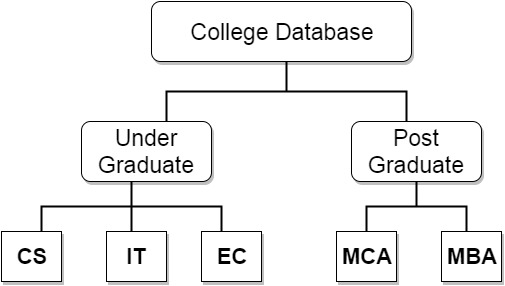
It uses pointer to navigate between the stored data.



In this model a database record is a tree that consists of one or more groupings of fields called segments, which makeup the individual nodes of the tree. This model use one-to-many relationship

**Advantage** : Data access is quite predictable in structure and retrieval and updates can be highly optimized by a DBMS.

**Disadvantage** : The link is permanently established and cannot be modified which makes this model rigid.



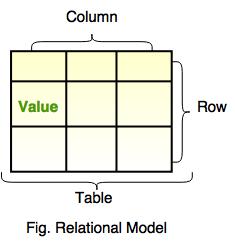
1. **Relational Model**

Record- and table-based model

**Relational database modeling is a *logical-level* model**

* Based on mathematical relations
* Uses relations, represented as tables
* Columns of tables represent attributes
* Tables represent relationships as well as entities
* It represents data as relations or tables.

Relational database simplifies the database structure by making use of tables and columns.



A relational database represents all data in the database as simple two-dimensional tables called relations. Each row of a relational table, called tuple, represents a data entity with columns of the table representing attributes(fields). The allowable values for these attributes are called the domain

Each row in a relational table must have a unique primary key and also has some secondary keys which correspond with primary keys in other tables.

**Advantage**: Provides flexibility that allows changes to the database structure to be easily accommodated. It facilitates multiple views of the same database for different users.

**For example**: COLLEGE table has Batch\_Year as primary key and has secondary keys Student\_ID and Course\_ID, these keys serve as primary keys for STUDENT and COURSE tables.

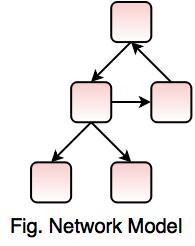
|  |  |
| --- | --- |
| Student Table | |
| Student\_ID | Student\_Name |
| 101 | Shubham |
| 102 | Rajat |

|  |  |
| --- | --- |
| Course Table | |
| Course\_ID | Course\_Name |
| 14 | Java |
| 16 | Android |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| College Table | | | | |
| Batch\_Year | Student\_ID | Course\_ID | Teacher\_Name | Teacher\_Number |
| 2012-16 | 101 | 14 | Jack | 9876543 |
| 2013-17 | 102 | 16 | Tom | 9823451 |

**3. Network Database Model**

* Network Database Model is same like Hierarchical Model, but the only difference is that it allows a record to have more than one parent.
* In this model, there is no need of parent to child association like the hierarchical model.
* It replaces the hierarchical tree with a graph.
* It represents the data as record types and one-to-many relationship.
* This model is easy to design and understand.

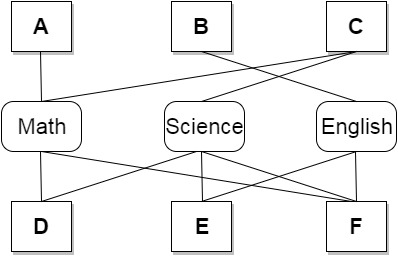


The Network database model was developed as an alternative to the hierarchical database.

This model expands on the hierarchical model by providing multiple paths among segments i.e more than one parent-child relationship.

Hence this model allows one-to-one, one-to-many and many-to-many relationships Supporting multiple paths in the data structure eliminates some of the drawbacks of the hierarchical model, the network model is not very practical.

**Disadvantage** : It can be quite complicated to maintain all the links.



**4. Entity Relationship Model**

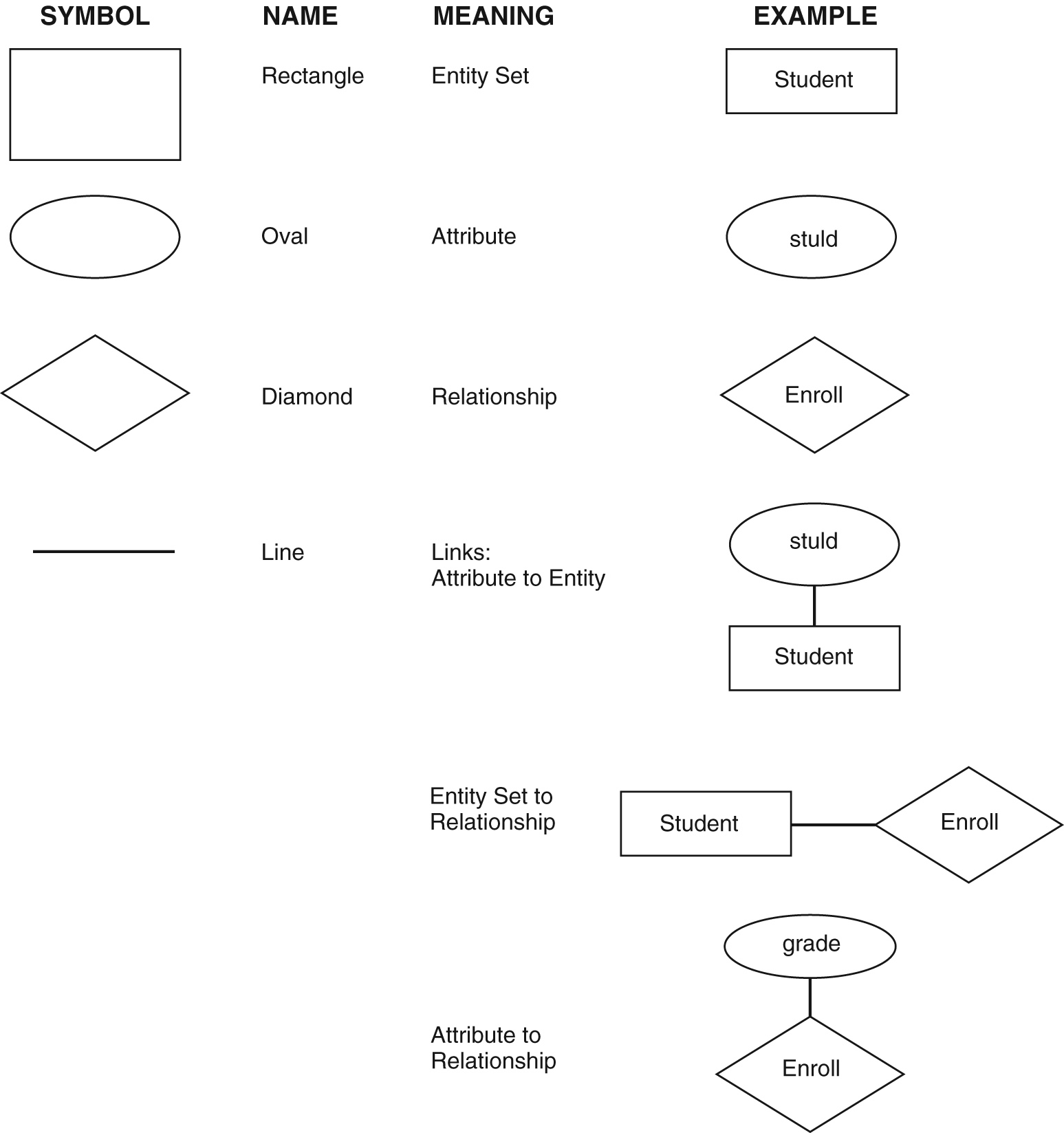
A semantic model captures meanings and intents

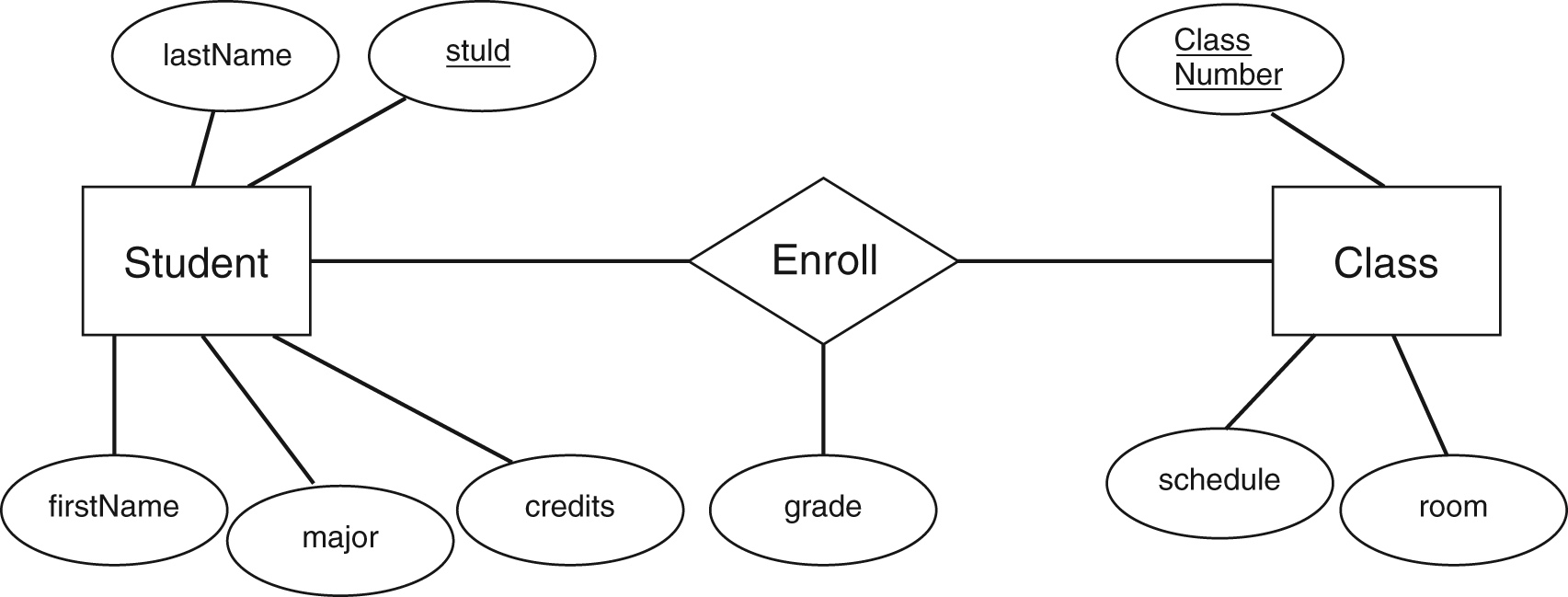
### E-R modeling is a conceptual level model

* **Entities** are real-world objects about which we collect data
* **Attributes** further describe the entities with particular values
* **Relationships** are associations among entities
* **Entity set**– set of entities of the same type
* **Relationship set** – set of relationships of same type

Relationships sets may also have descriptive attributes

Represented by E-R diagrams





## Object-oriented Model

Uses the E-R modeling as a basis but extended to include **encapsulation**, **inheritance**

Objects have both state and behavior

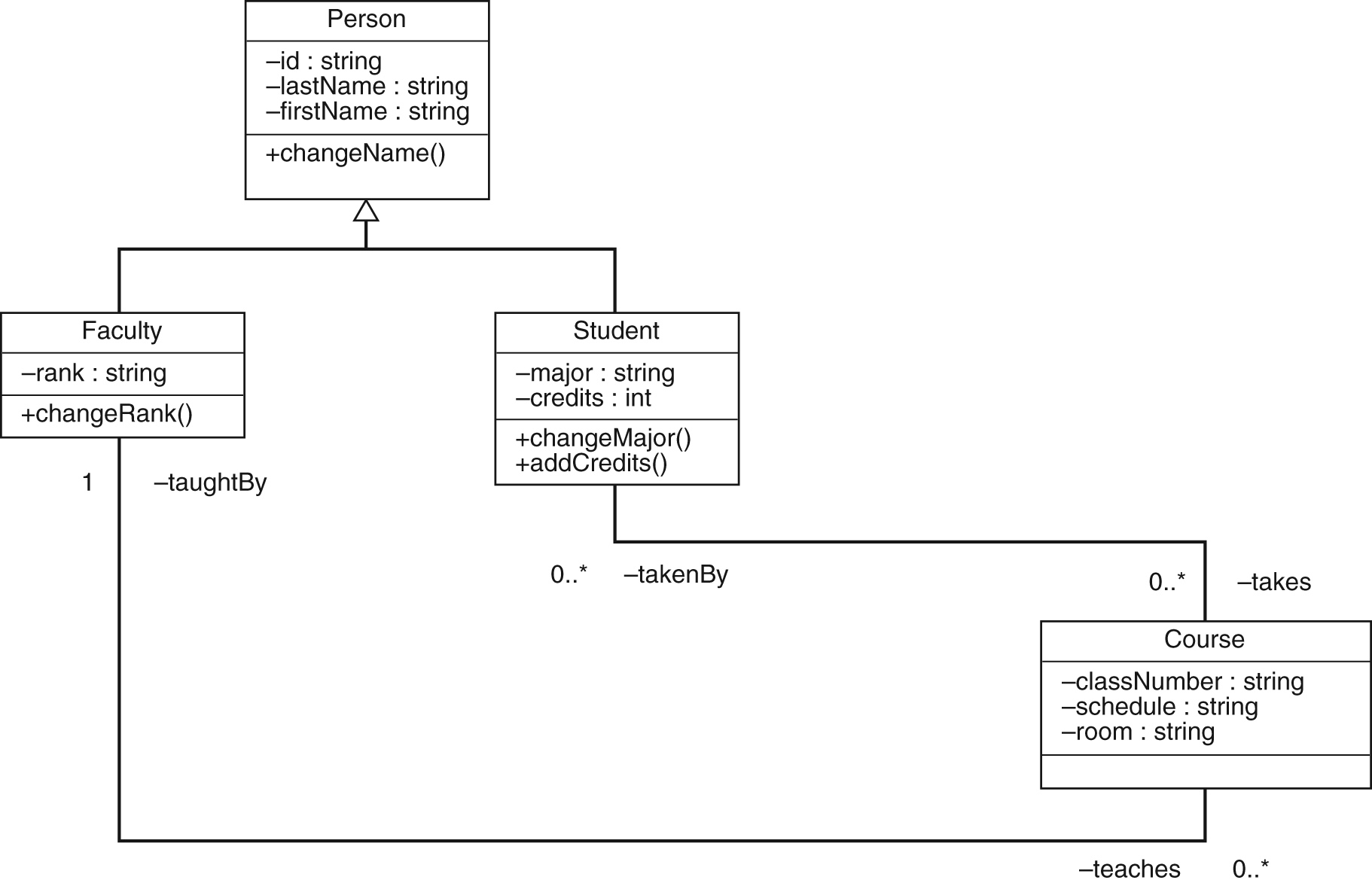
* **State** is defined by attributes
* **Behavior** is defined by methods (functions or procedures)

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
* Database objects have persistence

### Both conceptual-level and logical-level model



## Object-relational model

Adds new complex datatypes to relational model

Adds objects with attributes and methods

Adds inheritance

SQL extended to handle objects in SQL:1999

## Semi-structured Model

Collection of nodes, each with data, and with different schemas

Each node contains a description of its own contents

Can be used for integrating existing databases

XML tags added to documents to describe structure

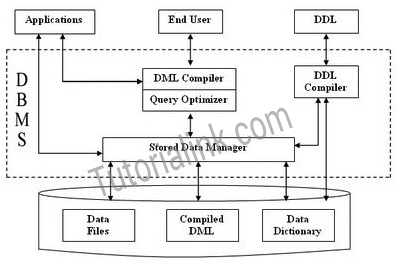
XML tags identify elements, sub-elements, attributes in documents

XML DTD (Document Type Definition) or XML Schema used to define structure

1. **Explain about Architecture of Database system.**

## Structure of DBMS:

* DBMS (Database Management System) acts as an interface between the user and the database. The user requests the DBMS to perform various operations such as insert, delete, update and retrieval on the database.
* The components of DBMS perform these requested operations on the database and provide necessary data to the users.
* The various components of DBMS are described below:



## Components of a DBMS

The components of DBMS can be divided into two parts:

### Function and Services of DBMS

1. **DDL Compiler:**
   * Data Description Language compiler processes schema definitions specified in the DDL.
   * It includes metadata information such as the name of the files, data items, storage details of each file, mapping information and constraints etc.
2. **DML Compiler and Query optimizer:**
   * The DML commands such as insert, update, delete, retrieve from the application program are sent to the DML compiler for compilation into object code for database access.
   * The object code is then optimized in the best way to execute a query by the query optimizer and then send to the data manager.
3. **Data Manager:**
   * The Data Manager is the central software component of the DBMS also knows as Database Control System.
   * The Main Functions Of Data Manager Are:
     1. Convert operations in user's Queries coming from the application programs or combination of DML Compiler and Query optimizer which is known as Query Processor from user's logical view to physical file system.
     2. Controls DBMS information access that is stored on disk.
     3. It also controls handling buffers in main memory.
     4. It also enforces constraints to maintain consistency and integrity of the data.
     5. It also synchronizes the simultaneous operations performed by the concurrent users.
     6. It also controls the backup and recovery operations.
4. Data Dictionary:
   * Data Dictionary, which stores metadata about the database, in particular the schema of the database.
   * names of the tables, names of attributes of each table, length of attributes, and number of rows in each table.
   * Detailed information on physical database design such as storage structure, access paths, files and record sizes.
   * Usage statistics such as frequency of query and transactions.
   * Data dictionary is used to actually control the data integrity, database operation and accuracy. It may be used as a important part of the DBMS
5. Data Files:
   * Which store the database itself.
6. Compiled DML:
   * The DML complier converts the high level Queries into low level file access commands known as compiled DML.
7. End Users:
   * The second class of users then is end user, who interacts with system from online workstation or terminals.
   * Use the interface provided as an integral part of the database system software.
   * User can request, in form of query, to access database either directly by using particular language, such as SQL, or by using some pre-developed application interface.
   * Such request are sent to query evaluation engine via DML pre-compiler and DML compiler
   * The query evaluation engine accepts the query and analyses it.
   * It finds the suitable way to execute the compiled SQL statements of the query.
   * Finally, the compiled SQL statements are executed to perform the specified operation
   * Query Processor Units:

Interprets DDL statements into a set of tables containing metadata.  
Translates DML statements into low level instructions that the query evaluation engine understands.  
Converts DML statements embedded in an application program into procedure calls int he host language.  
Executes low level instructions generated by DML compiler.

* + 1. DDL Interpreter
    2. DML Compiler
    3. Embedded DML Pre-compiler
    4. Query Evalution Engine
  + Storage Manager Units

Checks the authority of users to access data.

Checks for the satisfaction of the integrity constraints.

Preserves atomicity and controls concurrency.

Manages allocation of splace on disk.

#### Fetches data from disk storage to memory for being used.

* + 1. Authorization Manager
    2. Integrity Manager
    3. Transaction Manager
    4. File manager
    5. Buffer Manager
  + Functions of DBMS:
    1. DBMS free the programmers from the need to worry about the organization and location of the data i.e. it shields the users from complex hardware level details.
    2. DBMS can organize process and present data elements from the database. This capability enables decision makers to search and query database contents in order to extract answers that are not available in regular Reports.
    3. Programming is speeded up because programmer can concentrate on logic of the application.
    4. It includes special user friendly query languages which are easy to understand by non programming users of the system.
  + The service provided by the DBMS includes :-
    1. Authorization services like log on to the DBMS start the database stop the Database etc.
    2. Transaction supports like Recovery, Rollback etc,
    3. Import and Export of Data.
    4. Maintaining data dictionary
    5. User's Monitoring

1. **Write about History of Database Systems.**

Data are raw facts that constitute building blocks of information. Database is acollection of information and a means to manipulate data in a useful way, which must provide proper storage for large amounts of data, easy and fast access and facilitate the processing of data. Database Management System (DBMS) is a set of software that isused to define, store, manipulate and control the data in a database.Databases have been in use since the earliest days of electronic computing. Unlikemodern systems which can be applied to widely different databases and needs, the vastmajority of older systems were tightly linked to the custom databases in order to gainspeed at the expense of flexibility. Originally DBMSs were found only in largeorganizations with the computer hardware needed to support large data sets. From pre-stage flat-file system, to relational and object-relational systems, database technology hasgone through several generations and its 40 years history.

**The Evolution of the DatabaseAncient History:**

Data are not stored on disk; programmer defines both logical datastructure and physical structure, such as storage structure, access methods, I/O modes etc.One data set per program: high data redundancy. There is no persistence; Random accessmemory (RAM) is expensive and limited, programmer productivity low.

**1968 File-Based:**predecessor of database,Data maintained in a flat file. Processingcharacteristics determined by common use of magnetic tape medium. Data are stored infiles with interface between programs and files. Mapping happens between logical filesand physical file, one file corresponds to one or several programs. Various accessmethods exits, e.g., sequential, indexed, random. Requires extensive programming inthird-generation language such as COBOL, BASIC.

**Limitations:**

•Separation and isolation: Each program maintains its own set of data, users of one program may not aware of holding or blocking by other programs.

•Duplication: Same data is held by different programs, thus, wastes space andresources.

•High maintenance costs such as ensuing data consistency and controlling access

•Sharing granularity is very coarse

•Weak security.

**1968-1980 Era of non-relational database:**

A database provides integrated and structured collection of stored operational datawhich can be used or shared by application systems. Prominent hierarchical databasemodel was IBM’s first DBMS called IMS. Prominent network database model wasCODASYL DBTG model; IDMS was the most popular network DBMS.

# Data Abstraction

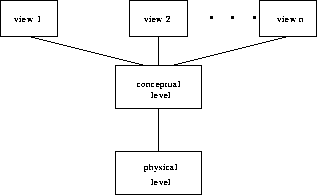
1. The major purpose of a database system is to provide users with an **abstract view** of the system.

The system hides certain details of how data is stored and created and maintained

Complexity should be hidden from database users.

1. There are several levels of abstraction:
   1. Physical Level:
      * How the data are stored.
      * E.g. index, B-tree, hashing.
      * Lowest level of abstraction.
      * Complex low-level structures described in detail.
   2. Conceptual Level:
      * Next highest level of abstraction.
      * Describes what data are stored.
      * Describes the relationships among data.
      * Database administrator level.
   3. View Level:
      * Highest level.
      * Describes part of the database for a particular group of users.
      * Can be many different views of a database.
      * E.g. tellers in a bank get a view of customer accounts, but not of payroll data.

Fig. [1.1](http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter1/node4.html#fig11abs) (figure 1.1 in the text) illustrates the three levels.

     
**Figure 1.1:** The three levels of data abstraction

**1.Explain about Relational Database**

A Relational database consist of a collection of tables, each of which is assigned a unique name. A row in a table represents a relationship among a set of values.

**Basic Terms in Relational Model**

**1. Relation**

A Relation is defined as a table with column and rows. We can use the term relation and tuple in place of the terms table and rows. Consider a relation **Student**

|  |  |  |
| --- | --- | --- |
| Student | | |
| **Roll\_number** | **Branch** | **Marks** |
| 101 | CS | 65 |
| 102 | IT | 88 |
| 103 | EC | 55 |

**2. Attributes**

The columns of a relation are known as attributes. Example: Roll\_number, Branch, Marks

**3. Domain**

For each attribute, there is a set of permitted values called domain of that attribute. Example: for attribute Branch, the domain is the set of all branches.

**4. Tuple**

It is a row of a relation.

**5. Degree**

The degree of a relation is equal to number of attributes. Example: the number of attributes in Student relation is 3, so degree is 3.

**6. Cardinality**

The number of tuples in a relation is called Cardinality.

**Merits of Relational Approach**

1. Easy to use
2. Flexibility
3. Security
4. Data Independence
5. Data Manipulation Language

**Demerits of Relational Approach**

1. Cost of setting up and maintaining the database system
2. They are not at storing very large records i.e. 100s of MBs or GBs

They tend to be slow and not scalable

**2. Explain about Data Storage and Querying.**

A database system is partitioned into modules that deal with each of the responsibilities 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. Corporate databases range in size from hundreds of  
gigabytes to, for the largest databases, terabytes of data. A gigabyte is approximately  
1000 megabytes (actually 1024) (1 billion bytes), and a terabyte is 1 million  
megabytes (1 trillion bytes). Since the main memory of computers cannot store  
this much information, the information is stored on disks. Data are moved between  
disk storage and main memory as needed. Since the movement of data  
to and from disk is slow relative to the speed of the central processing unit, it is  
imperative that the database system structure the data so as to minimize the need  
to move data between disk and main memory.

The query processor is important because it helps the database system to  
simplify and facilitate access to data. The query processor allows database users  
to obtain good performance while being able to work at the view level and not be  
burdened with understanding the physical-level details of the implementation of  
the system. It is the job of the database system to translate updates and queries  
written in a nonprocedural language, at the logical level, into an efficient sequence  
of operations at the physical level.

**1. Storage Manager**

The *storage manager* is the component of a database system 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 provided by the operating system. The storage manager  
translates the various DML statements 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 in 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.

The storage manager implements several data structures as part of the physical  
system implementation:

**• Data files**, which store the database itself.

**• Data dictionary,** which stores metadata about the structure of the database,  
in particular the schema of the database.

• **Indices,** which can provide fast access to data items. Like the index in this  
textbook, a database index provides pointers to those data items that hold a  
particular value. For example, we could use an index to find the instructor  
record with a particular ID, or all instructor records with a particular name.  
Hashing is an alternative to indexing that is faster in some but not all cases.

**2. The 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  
evaluation plan consisting of low-level instructions that the query evaluation  
engine understands.

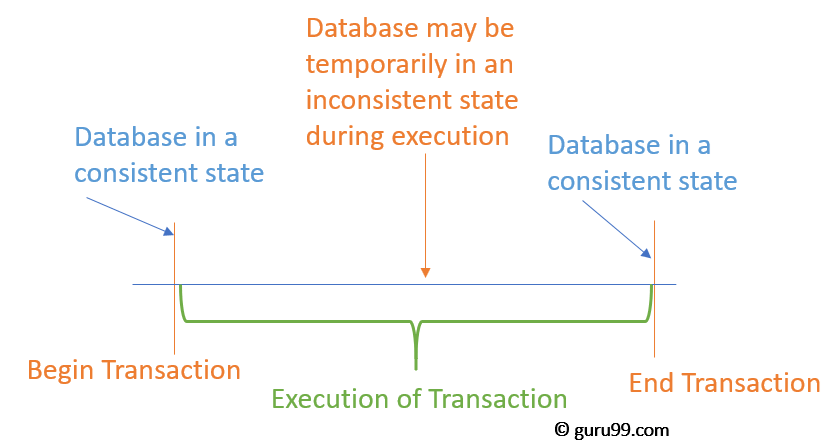
     A query can usually be translated into any of a number of alternative  
evaluation plans that all give the same result. The DMLcompiler also performs  
**query optimization;** that is, it picks the lowest cost evaluation plan from  
among the alternatives.

• **Query evaluation engine,** which executes low-level instructions generated  
by the DML compiler.

TransactionManagement:

A transaction is a logical unit of processing in a DBMS which entails one or more database access operation. In a nutshell, database transactions represent real-world events of any enterprise.

All types of database access operation which are held between the beginning and end transaction statements are considered as a single logical transaction. During the transaction the database is inconsistent. Only once the database is committed the state is changed from one consistent state to another.

[](https://www.guru99.com/images/1/100518_0500_DBMSTransac1.png)

## Database System versus File System

|  |  |
| --- | --- |
| **DBMS** | **File Processing System** |
| Minimal data redundancy problem in DBMS | Data Redundancy problem exits |
| Data Inconsistency does not exist | Data Inconsistency exist here |
| Accessing database is easier | Accessing is comparatively difficult |
| The problem of data isolation is not found in database | Data is scattered in various files and files may be of different format, so data isolation problem exists |
| Transactions like insert, delete, view, updating, etc are possible in database | In file system, transactions are not possible |
| Concurrent access and recovery is possible in database | Concurrent access and recovery is not possible |
| Security of data | Security of data is not good |
| A database manager (administrator) stores the relationship in form of structural tables | A file manager is used to store all relationships in directories in file systems. |