~~*Database Management System*~~
~~*(CS403)*~~

Database definitions:

A shared collection of logically related data, designed to meet the information needs of multiple users in an organization.

- **Database Application:**

Database Application is a program or group of programs which is used for performing certain operations on the data stored in the database. These operations may contain insertion of data into a database or extracting some data from the database based on a certain condition, updating data in the database, producing the data as output on any device such as Screen, disk or printer.

- **Database Management Systems:**

Database management system is software of collection of small programs to perform certain operation on data and manage the data.

Two basic operations performed by the DBMS are:

- **Management of Data in the Database**
- **Management of Users associated with the database.**

Further Advantages of Database Systems:

Database systems are very much beneficent to enterprises and businesses, some of the advantages are listed below:

- Data consistency
- Better data security
- Faster development of new applications
- Economy of scale
- Better concurrency control
- Better backup and recovery procedures

- **Data Consistency:**

Data consistency means that the changes made to different occurrence of data should be controlled and managed in such a way that all the occurrences have same value for any specific data item

Better Data Security:

All application programs access data through DBMS, So DBMS can very efficiently check that which user is performing which action and accessing which part of data , So A DBMS is the most effectively control and maintain security of Data stored in a database.

- **Economy of Scale:**

Databases and database systems are designed to share data stored in one location for many different purposes, So it needs not be stored as many number of times in different forms as it is used, for example the data used by Admission Department of any education institution can be used to maintain the attendance record of the students as well as the examination records of the students. So it saves us lots of efforts and finances providing

- **Better Concurrency Control:**

Concurrency means the access of database form as number of points simultaneously.

Concurrency control means to access the database in such a way that all the data accesses are completed correctly and transparently. One example of controlled concurrency is the
use of ATM Machine for withdrawal of money (cash). All ATM machines of a bank are interconnected to a central database system worldwide, so that a user can access its account from anywhere in the world and can get cash from any ATM terminal. As there are thousands of ATM terminal across the world for a specific bank so as a result thousands of user process and access the bank's database. All this process is managed concurrently using the database systems and is done in such an efficient manner that no two user face any delay in the processing of their requests.

Better Backup and Recovery Facility:

Data is a very important resource and is very much valuable for any organization, loss of such a valuable resource can result in a huge strategic disasters. As Data is stored on today's' storage devices like hard disks etc., It is necessary to take periodic backups of data so that in case a storage device looses the data due to any damage we should be ableto restore the data a nearest point, Database systems offer excellent facilities for taking backup of data and good mechanism of restoring those backups to get back the backed-up data.

- **High Cost:**

Database Systems have a number of inherent charges which are to be born by any organization that is going to adopt it. High Cost is one of these inherent charges, it includes the need for specialized software which is used to run database systems, Additional and specialized hardware and technically qualified staff are the requirements for adopting to the database system, all these requirements need an organization to invest handsome amount of money to have all the requirements of the database systems.

- **Conversion Cost:**

Once an organization has decided to adopt database system for its operations, it is not only the finance and technical man-power which is required for switching on to database system, it further has some conversion charges needed for adopting the database system,

- **Difficult Recovery Procedures:**

Although the database systems and database management systems provide very efficient ways of data recovery in case of any disaster, still the process of recovering a crashed database is very much technical and needs good professional skills to perform a perfect recovery of the database.

Importance of Data

- **Data as a Resource:**

A resource is anything which is valuable for an organization. There can be a number of resources in any organization, for example, Buildings, Furniture, Vehicle,
Technical Staff,
Managers, supporting staff and Machinery etc. As all these are resources for organizations and are consumed very much carefully to get full benefit out of them.
Data
in the same way is a very important resource and needs to considered equally important as other resource are considered.

Levels of Data

- **Real World Data**

The real world level of data means that level of data at which entities or objects exist in reality, it means that any object existing in reality have a name and other identifiable attributes through which we can identify that specific object or entity.

Example:

Any Student

- Meta Data:

For storage of the data related to any entity or object existing at real world level we define the way the data will be stored in the database. This is called Meta data. Meta data is also known as schema for the real world data.

Example: Name , Character Type, 25 character size field,
Age, Date type, 8 bytes size
Class, Alpha Numeric, 8 byte size field

- **Existence of Data:**

Existence of the data level shows the actual data regarding the entities as real world level according to the rules define at the Meta Data level.

Example:

According to the definition given in the Meta data level the Actual data or Data occurance for the entity at real world level is shown below:

Name Age Class
Ali 20/8/1979 MCS-I
Amir 22/3/1978 MCS-II etc…

Users of Database Systems:

- **Application Programmers**
- **End Users**

 Naïve

 Sophisticated

 Naïve Users

This category of users is that category who simply use the application database programs created by the programmers. This group has no interaction with other parts of there database and only use the programs meant for them. They have not to worry about the further working of the database.

 Sophisticated Users:

This type of users has some additional rights over the Naïve users, which means that they can access the data stored in the database any of their desired way. They can access data using the application programs as well as other ways of accessing data.

- **Database Administrators (DBA):**

This class of database users is the most technical class of db users. They need to have the knowledge of how to design and manage the database use as well as to manage the data in the database. DBA is a very responsible position in an organization. He is responsible for proper working of the database and DBMS, has the responsibility of making proper database backups and make necessary actions for recovering the database in case of a database crash.

 Duties of the DBA

- **Schema definition**
- **Granting data access**
- **Routine Maintenance**

 Backups

 Monitoring disk space

 Monitoring jobs running

- **Schema Design**

DBA is the person who create all the meta data information for the organization

- **Granting Access to Users:**

DBA is also responsible for grant of access rights to the database users.

- **Monitoring Disk Space :**

The DBA has to monitor the
disk space usage and statistics to ensure that no data over flow occurs at any stage.

- Monitoring Running Jobs:

To ensure the secure and proper functioning of the database system

Data Independence:

- **Logical Data Independence**
- **Physical Data Independence**

Logical data independence

Logical data independence provides the independence in a way that changes in conceptual model do not affect the external views. Or simply it can be stated at the Immunity of external level from changes at conceptual level.

Physical Data Independence

Physical data independence is that type of independence that provides us changes transparency between the conceptual and internal levels.

Functions of DBMS

- **Data Processing**
- **A user accessible Catalog**
- **Transaction Support**
- **Concurrency Control Services**
- **Recovery Services**
- **Authorization Services**
- **Support for Data Communication**
- **Integrity Services**

Two important functions that the DBMS performs are:

User management

Data Management

- **Data Processing**

creation of data, Storing of the data in the database, arrangement of the data in the databases and data-stores

- **A User Accessible Catalog**

DBMS which stores almost all of the information of the database, including schema information, user information right of the users

- **Transaction Support**

Transaction is an action that is used to perform some manipulation on the data stored in the database.

- **Concurrency Support**

Concurrency support means to support a number of transactions to

be executed simultaneously

- **Recovery Services**

Recovery services mean that in case a database gets an inconsistent state to get corrupted due to any invalid action of someone, the DBMS should be able to recover itself to a consistent state, ensuring that the data loss during the recovery process of the database remains minimum.

- **Authorization Services**

The database is intended to be used by a number of users, who will perform a number of actions on the database and data stored in the database, The DBMS is used to allow or restrict different database users to interact with the database. It is the responsibility of the database to check whether a user intending to get access to database is authorized to do so or not.

- **Support for Data Communication**

The DBMS should also have the support for communication of the data indifferent ways. For example if the system is working for such an organization which is spread across the country and it is deployed over a number of offices throughout the country, then the DBMS should be able to communicate to the central database station. Or if the data regarding a product is to be sent to the customers worldwide it should have the facility of sending the data of the product in the form of a report or offer to its valued customers.

- **Integrity Services**

Integrity means to maintain something in its truth or originality. The same concept applies to the integrity in the DBMS environment. Means the DBMS should allow the operation on the database which are real for the specific organization and it should not allow the false information or incorrect facts.

 Teleprocessing

database systems processes the user
requests at a central computer, all requests are carried to the central computer where the database is residing, transactions are carried out and the results transported back to the terminals (literally dumb terminals).

 File Servers

multi-user database environment Each client of the network runs its own copy of the DBMS and the database resides on the file server.

 Client-Server

multi-user environment It has a DBMS server machine
which runs the DBMS and to this machine are connected the clients having application programs running for each user. Once a users wants to perform a certain operation on data in the database it sends its requests to the DBMS through its machine's application Software This environment is best suited for large
enterprises where bulk of data is processed and requests are very

much frequent.

Database Development Process:

- **Analyze User Environment**
- **Develop Conceptual Model**
- **Map Conceptual Model to Logical**
- **Choose DBMS**
- **Develop Physical Design**
- **Implement System**
- **Test System**
- **Operational Maintenance**

Data Flow Diagrams:

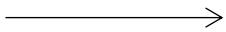
DFDs show the flow of data between different processes o a specific system.
DFDs are simple, and hide complexities.
DFDs are Descriptive and links between processes describe the information flow.

- **Limitation of DFDs**

They do not provide us a way of expressing decision points.
DFDs are focused on flow of information only.

- **Symbols used in DFD:**
- **DATAFLOW:**

The purpose of the dataflow in a DFD is to express the flow of information from one entity to another entity in the system



- **DATA STORE:**

Data store is a repository for the storage of the data.

- **Processes:**

Processes are expressed with ovals or rounded rectangles

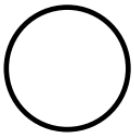


Fig: 6a Process

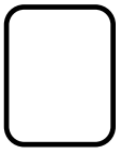


Fig-6b

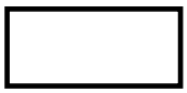
- **DFD-Process:**

In DFD processes are numbered for expressing their existence at a certain level in the system.

: 7. Numbered DFD Processes

- **External Entities:**

These are the entities interacting with the system in any of two different ways. They may be either receiving the data from the system, or may be producing the data for the system to consume



- **Collector:**

This DFD shape is used to express several dataflow connections terminating at a single location.

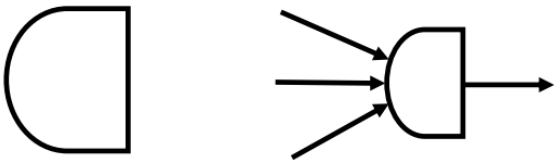


Fig: 9a Collector Fig 9b. Collector with Multiple Dataflow

- **Separator:**

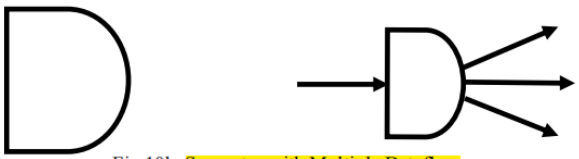


Fig: 10a Separator Fig 10b. Separator with Multiple Dataflow

- **Ring Sum Operator:**

Fig: 11a Ring sum operator Fig 11b. Separator with Ring sum operator

- **AND Operator:**

'a AND operator

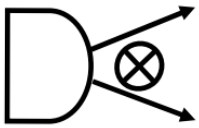
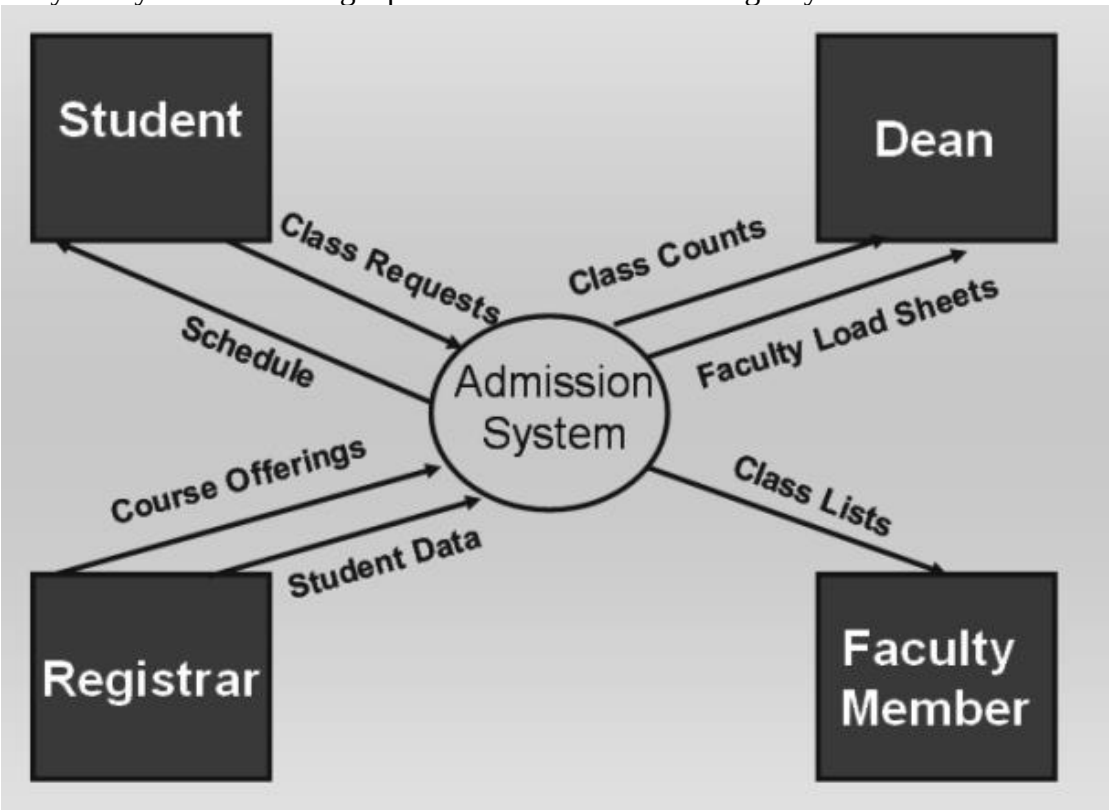


Fig 12b. Separator with AND operator

Types of DFD

- **Context diagram**
- **Level 0 diagram**
- **Detailed diagram**
- **Context Diagram:**

This is the level of DFD which provides the least amount of details about the working of the system. Context DFDs have the following properties:
They always consist of single process and describe the single system.



Example Context DFD Diagram

Types of Data Models

- **ER- Data Model**
- **Object oriented data model**

 Record Based Data Model
This is the second type of data models available to use and has three basic types
 Hierarchical Data Model
 Network Data model
 Relational Data model

Types of Database Design

Conceptual database design
Logical Database design
Physical Database Design:

Entity-Relationship Data Model

It is a semantic data model that is used for the graphical representation of the conceptual database design

Constructs in E-R Data Model

The E-R data model supports following major constructs:
 Entity
 Attribute
 Relationship

The Entity

Entity is basic building block of the E-R data model. The term entity is used in three different meanings or for three different terms and that are:
 Entity type (person, place, event)
 Entity instance(A particular object belonging to a particular entity type and how does an item becomes an instance of or belongs to an entity type)

Entity Types	Properties	Instances
EMPLOYEE	Human being, has name, has father name, has a registration number, has qualification, designation	M. Sharif, Sh. Akmal and many others

 Entity set

Classification of entity types

Weak Entity Types

An entity type whose instances cannot exist without being linked with instances of some other entity type,For example, in an organization
we want to maintain data about the vehicles owned by the employees. Now a particular vehicle can exist in this organization only if the owner already exists there as employee. Similarly, if employee leaves the job and the organization decides to delete the record of the employee then the record of the vehicle will also be deleted since it cannot exist without being linked to an instance of employee.

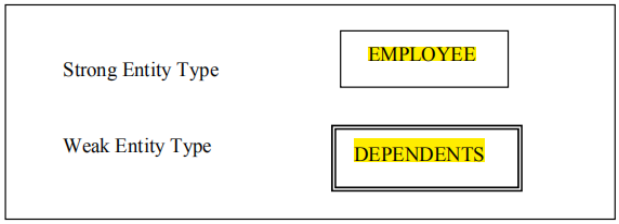
Strong Entity Type

An entity type whose instances can exist independently, that is, without being linked to the instances of any other entity type is called strong entity type. A major property of the strong entity types is that they have their own identification, which is not always the case with weak entity types. For example, employee in the previous example is an independent or strong entity type, since its instances can exist independently.

Naming Entity Types

Following are some recommendations for naming entity types. But they are just

recommendations; practices considered good in general. If one, some or all of them are ignored in a design, the design will still be valid if it satisfies the requirements otherwise, but good designs usually follow these practices:
 Singular noun recommended, but still plurals can also be used
 Organization specific names, like customer, client, gahak anything will work
 Write in capitals, yes, this is something that is generally followed, otherwise will also work.
 Abbreviations can be used, be consistent. Avoid using confusing abbreviations, if they are confusing for others today, tomorrow they will confuse you too.



Attribute ((E.g. Student Identification, Student Name).Like, one instance may have the values (M. Hafeez, Noor Muhammad, 37)

Domain of an Attribute

Domain is normally defined in form of data type and some additional constraints like the range constraint
Some common data types are Integer, Float, Varchar, Char,
String, etc.

Types of Attributes

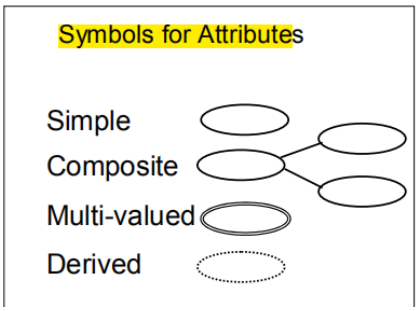
 Simple or Composite(stName, stFatherName, stDateOfBorth
 Single valued or multi-valued
 Stored or Derived

 Single valued or multi-valued

Some attribute have single value at a time, whereas some others may have multiple values. For example, hobby attribute of STUDENT or skills attribute of EMPLOYEE, since a student may have multiple hobbies, likewise an employee may have multiple skills so they are multi-valued attributes. On the other hand, name, father name, designation are generally single valued attributes.

 Stored or Derived

Normally attributes are stored attributes, that is, their values are stored and accessed as such from the database. However, sometimes attributes' values are not stored as such, rather they are computed or derived based on some other value. This other value may be stored in the database or obtained some other way. For example, we may store the name, father name, address of employees, but age can be computed from date of birth.



Attributes

Def 1:

An attribute is any detail that serves to identify, qualify, classify, quantify, or otherwise express the state of an entity occurrence or a relationship.

Def 2:

Attributes are data objects that either identify or describe entities.

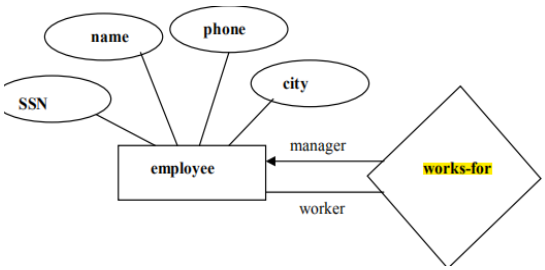
The Keys

A key is a set of attributes that can be used to identify or access a particular entity instance of an entity type (or out of an entity set). The concept of key is beautiful and very useful; why and how.

- **Super Key**
- **Candidate Key**
- **Primary Key**
- **Alternate Key**
- **Secondary Key**
- **Foreign Key**

Naming Relationships:

If there is no proper name of the association in the system then participants' names of abbreviations are used. STUDENT and CLASS have ENROLL relationship. However, it can also be named as STD_CLS.



Symbol for Relationships:

- **Shown as a Diamond**
- **Diamond is doubled if one of the participant is dependent on the other**
- **Participants are connected by continuous lines, labeled to indicate cardinality.**
- In partial relationships roles (if identifiable) are written on the line connecting the partially participating entity rectangle to the relationship diamond.
- Total participation is indicated by double lines

Types of Relationships

- **Unary Relationship**

An ENTITY TYPE linked with itself, also called recursive relationship. Example Roommate, where STUDENT is linked with STUDENT

- **Binary relationship**

A Binary relationship is the one that links two entities sets e.g. STUDENT-CLASS. Relationships can be formally described in an ordered pair form.
Enroll = {(S1001, ART103A), (S1020, CS201A), (S1002, CSC201A)}
Entire set is relationship set and each ordered pair is an instance of the relationship.

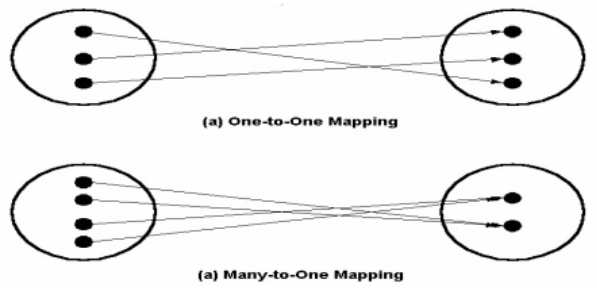
- **Ternary Relationship**

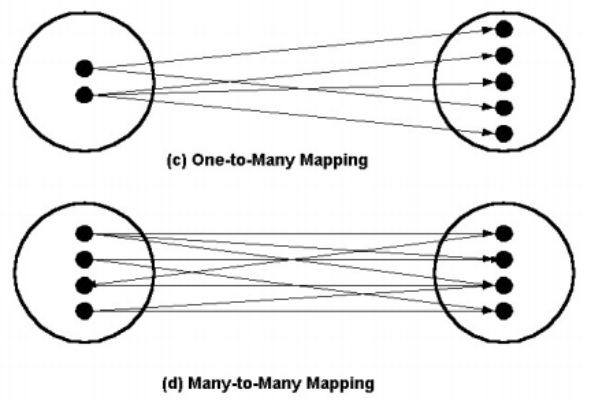
A Ternary relationship is the one that involves three entities

- **N-ary Relationship**

Most relationships in data model are binary or at most ternary but we could define a relationship set linking any number of entity sets i.e. n-ary relationship
Relationship Cardinalities
The cardinality of a relationship is the number of entities to which another entity can map under that relationship. Symbols for maximum and minimum cardinalities are:

- **One-to-One mapping:**
- **Many-to-One mapping:**
- **One-to-Many mapping:**
- **Many-to-Many mapping:**





Roles in Relationships

The way an entity is involved in a relationship is called the role of the entity in the Relationship

Dependencies

dependency is a constraint. There are a number of dependency types

The Existence dependency:

This is the type of dependency which exists when one entity instance needs instance of another entity for its existence. employee of and organization and the projects

Identifier Dependency:

It means that the dependent entity incase of existence dependency does not have its **own identifier**

Referential Dependency:

This is the situation when the dependent entity has it own key for unique identification but the key used to show the reference with the parent entity is shown with the help of an attribute of the parent entity.