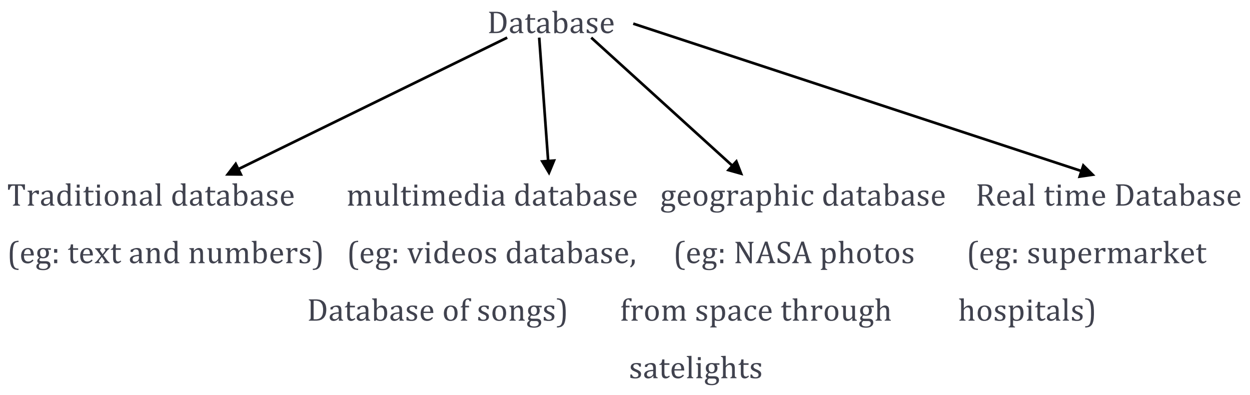
UNIT-1

Data is a fact that should be recorded like name, number, address, phone number, videos, speeches etc

Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

Various Databases


Database
Traditional database (eg: text and numbers)
multimedia database (eg: videos database, Database of songs)
geographic database (eg: NASA photos from space through satelights
Real time Database (eg: supermarket hospitals)

in earlier when we want to use the database first we have to call the operating system due to this reason Operating system switches from kernel to user mode in order to access the data

it is more complicated . so for that reason we design a program to define the data in database in a software and we manipulate that and that program is called **DBMS**

DBMS is a software

The combination of Database and DBMS is called DBS(Database Systems)

File System Vs DBMS

There are following differences between DBMS and File system:

DBMS	File System
DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	File system is a collection of data. In this system, the user has to write the procedures for managing the database.
DBMS gives an abstract view of data that hides the details.	File system provides the detail of the data representation and storage of data.
DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from the system failure.	File system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will lost.
DBMS provides a good protection mechanism.	It is very difficult to protect a file under the file system.
DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	File system can't efficiently store and retrieve the data.
DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while other deleting some information or updating some information.

Drawbacks of File system

- **Data redundancy:** Data redundancy refers to the duplication of data, lets say we are managing the data of a college where a student is enrolled for two courses, the same student details in such case will be stored twice, which will take more storage than needed. Data redundancy often leads to higher storage costs and poor access time.

- **Data inconsistency:** Data redundancy leads to data inconsistency, lets take the same example that we have taken above, a student is enrolled for two courses and we have student address stored twice, now lets say student requests to change his address, if the address is changed at one place and not on all the records then this can lead to data inconsistency.
- **Data Isolation:** Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.
- **Dependency on application programs:** Changing files would lead to change in application programs.
- **Atomicity issues:** Atomicity of a transaction refers to "All or nothing", which means either all the operations in a transaction executes or none.

  For example: Lets say Steve transfers 100$ to Negan's account. This transaction consists multiple operations such as debit 100$ from Steve's account, credit 100$ to Negan's account. Like any other device, a computer system can fail lets say it fails after first operation then in that case Steve's account would have been debited by 100$ but the amount was not credited to Negan's account, in such case the rollback of operation should occur to maintain the atomicity of transaction. It is **difficult to achieve atomicity in file processing systems**.

- **Data Security:** Data should be secured from unauthorised access, for example a student in a college should not be able to see the payroll details of the teachers, such kind of security constraints are difficult to apply in file processing systems.

Advantage of DBMS over file system

There are several advantages of Database management system over file system. Few of them are as follows:

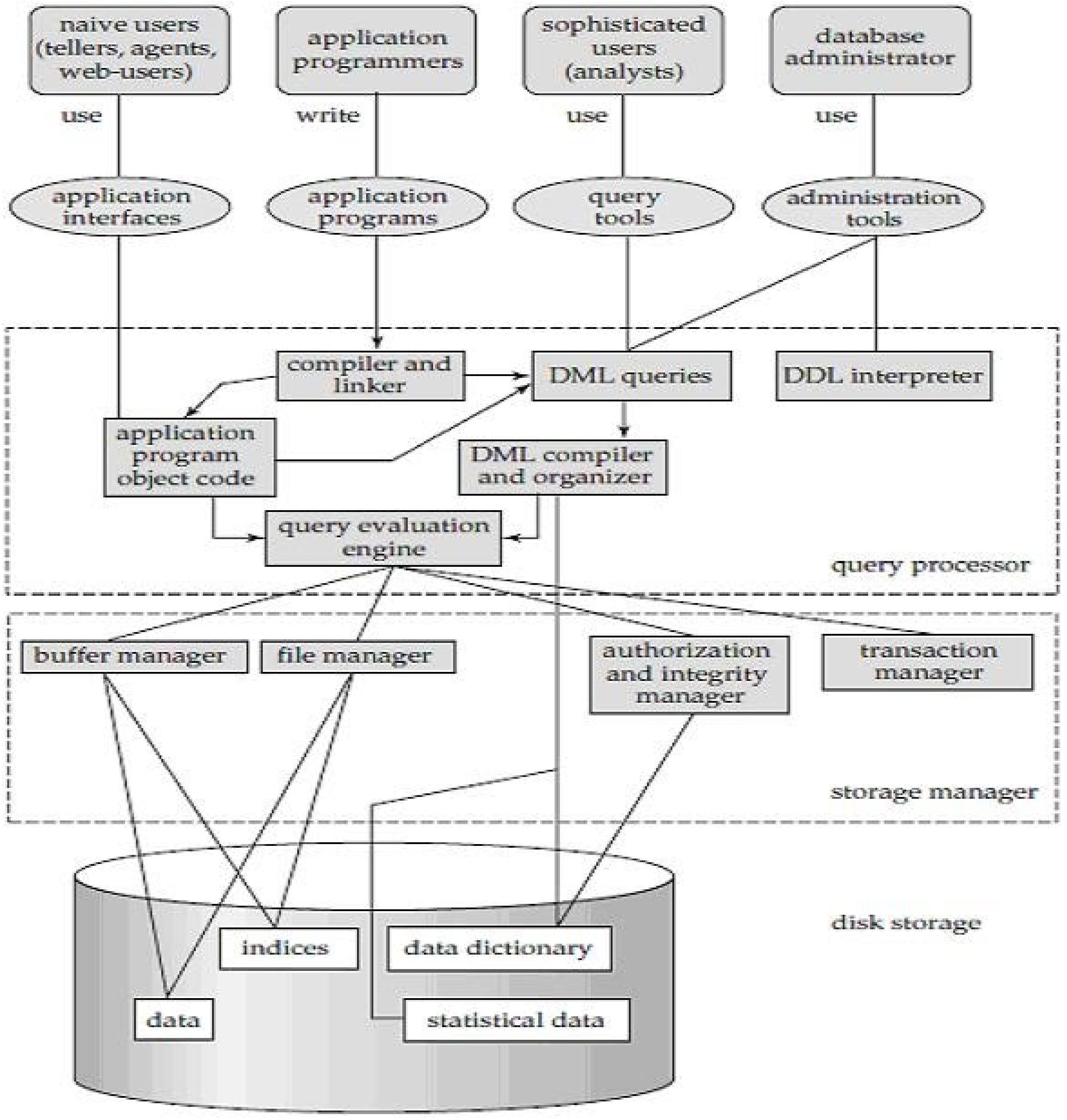
- **No redundant data**: Redundancy removed by data normalization. No data duplication saves storage and improves access time.
- **Data Consistency and Integrity**: As we discussed earlier the root cause of data inconsistency is data redundancy, since data normalization takes care of the data redundancy, data inconsistency also been taken care of as part of it
- **Data Security**: It is easier to apply access constraints in database systems so that only authorized user is able to access the data. Each user has a different set of access thus data is secured from the issues such as identity theft, data leaks and misuse of data.
- **Privacy**: Limited access means privacy of data.

- **Easy access to data –** Database systems manages data in such a way so that the data is easily accessible with fast response times.
- **Easy recovery**: Since database systems keeps the backup of data, it is easier to do a full recovery of data in case of a failure.
- **Flexible**: Database systems are more flexible than file processing systems.

Disadvantages of DBMS:

- DBMS implementation cost is high compared to the file system
- Complexity: Database systems are complex to understand
- Performance: Database systems are generic, making them suitable for various applications. However this feature affect their performance for some applications
- 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. The query processor is important because it helps the database system simplify and facilitate access to data.

•



- 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.
- **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.

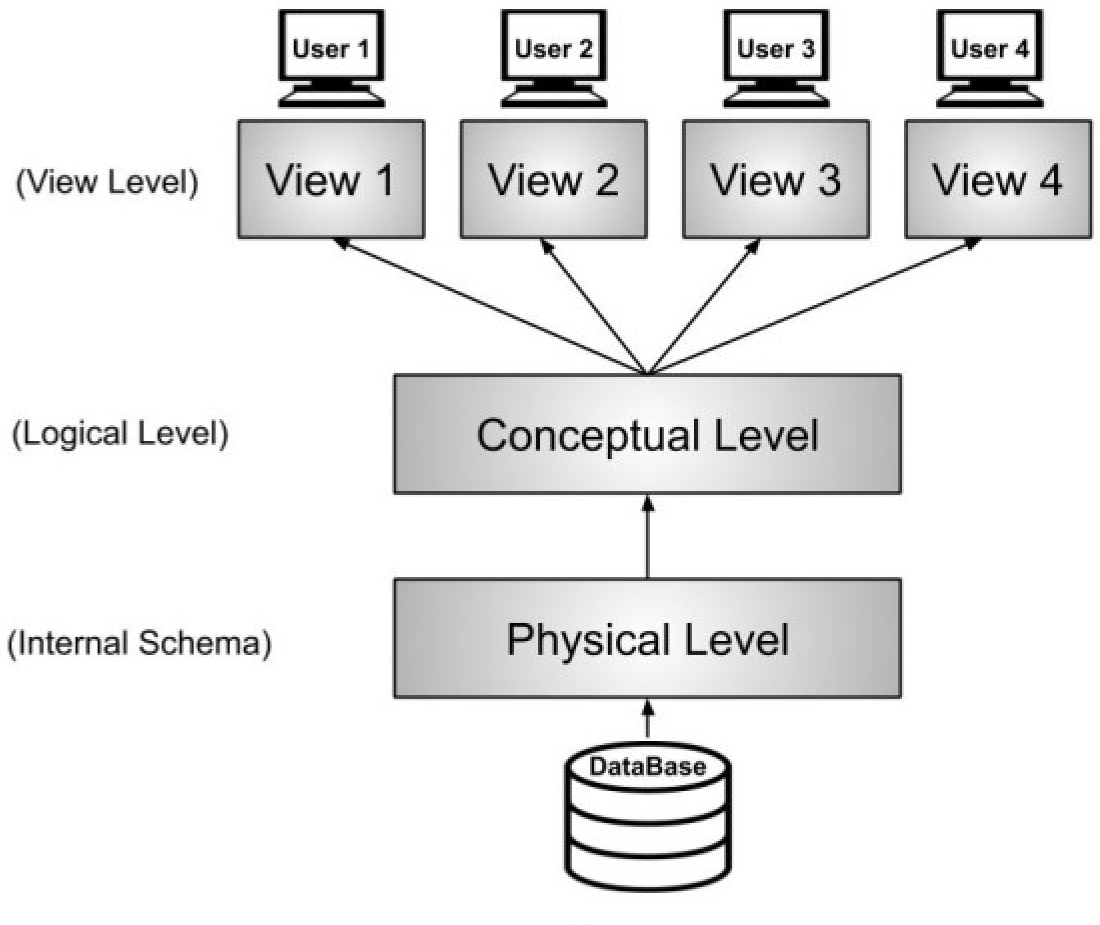
- A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs **query optimization**, that is, it picks the lowest cost evaluation plan from among the alternatives.
- · **Query evaluation engine,** which executes low-level instructions generated by the DML compiler.
- **Storage Manager**
- A *storage manager* is a program module that provides the interface between the lowlevel 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 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.
- **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.

Level of Data Abstraction

Data Abstraction refers to the process of hiding irrelevant details from the user. So, what is the meaning of irrelevant details? Let's understand this with one example. ***Example:*** If we want to access any mail from our Gmail then we don't know where that data is physically stored i.e is the data present in India or USA or what data model has been used to store that data? We are not concerned about these things. We are only concerned with our email. So, information like these i.e. location of data and data models are irrelevant to us and in data abstraction, we do this only. Apart from the location of data and data models, there are other factors that we don't care of. We hide the unnecessary data from the user and this process of hiding unwanted data is called Data Abstraction.

There are mainly three levels of data abstraction and we divide it into three levels in order to achieve *Data Independence*. Data Independence means users and data should not directly interact with each other. The user should be at a different level and the data should be present at some other level. By doing so, Data Independence can be achieved. So, let's see in details what are these three levels of data abstraction:

1. View Level
2. Conceptual Level
3. Physical Level



Levels of Data Abstraction

View Level or External Schema

This level tells the application about how the data should be shown to the user. ***Example:*** If we have a login-id and password in a university system, then as a student, we can view our marks, attendance, fee structure, etc. But the faculty of the university will have a different view. He will have options like salary, edit marks of a student, enter attendance of the students, etc. So, both the student and the faculty have a different view. By doing so, the security of the system also increases. In this example, the student can't edit his marks but the faculty who is authorized to edit the marks can edit the student's marks. Similarly, the dean of the college or university will have some more authorization and accordingly, he will has his view. So, different users will have a different view according to the authorization they have.

Conceptual Level or Logical Level

This level tells how the data is actually stored and structured. We have different data models by which we can store the data(You can read more about the different types of data model from here). ***Example***: Let us take an example where we use the relational model for storing the data. We have to store the data of a student, the columns in the student table will be student_name, age, mail_id, roll_no etc. We have to define all these at this level while we are creating the database. Though the data is stored in the database but the structure of the tables like the student table, teacher table, books table, etc are defined here in the conceptual level or logical level. Also, how the tables are related to each other are defined here. Overall, we can say that we are creating a blueprint of the data at the conceptual level.

Physical Level or Internal Schema

As the name suggests, the Physical level tells us that where the data is actually stored i.e. it tells the actual location of the data that is being stored by the user. The Database Administrators(DBA) decide that which data should be kept at which particular disk drive, how the data has to be fragmented, where it has to be stored etc. They decide if the data has to be centralized or distributed. Though we see the data in the form of tables at view level the data here is actually stored in the form of files only. It totally depends on the DBA, how he/she manages the database at the physical level.

So, the Data Abstraction provides us with a different view and help in achieving Data Independence

Database users and Administrators

The primary goal of any database is to provide its users an environment for retrieving and storing new information into the database. An important aspect of the DBA‘s working is that, it has liaison with users at all levels of the organization. The DBA must be able to communicate effectively with the users and must be conversant with the technical aspects of the system as well as the working procedures of the organization.

Depending on the way that the users expect to interact with the database system, the users are classified in to

1. Application Programmers : Application Programmers are computer professionals interacting with the system through DML calls, embedded in a program written in a language like high level languages like COBOL, C, etc.

2. Sophisticated users : These users interact with system without writing programs. They form their request by writing queries in a database query language. Those are submitted to a query processor that breaks a DML statement down in to instructions for the database manager's module.

3. Unsophisticated users : Who interact with the system by using permanent applications. Ex ATM

4. Specialized users : These users write specialized database applications that do not fir in to the traditional data processing frame work. These applications include Computer –aided design (CAD) systems, Knowledge base expert systems etc.

Database Administrator

A person who has central control of both the data and programs over the system is called database administrator. Or

Database administrator (DBA) is the person or group , who is responsible for the supervision and control of the databases, within on organization.

The functions or responsibilities of DBA includes

1. Schema Definition : The DBA creates the original database schema by writing a set of definitions which are translated by DDL compiler to a set of tables that is stored permanently in the data dictionary.

2. Storage structure and access Method Definition :A DBA creates appropriate storage structures and access methods by writing a set of definitions ,which are translated by the data-storage and DDL compiler.

3. Schema Physical Organization and Modification : The DBA allows the users to accomplish modifications in the database scheme or storage organization through a set of definitions and make these modifications to be appropriate in the internal system tables.

4. Granting of Authorization for data access : By granting different types of authorization, the database administrator can regulate which of the database various users can access.

5. Routine Maintenance :DBA is the final authority to regulate the daily activities.

As a whole, the DBA jobs are

- Creating primary database storage structures
- Modifying the structure of the database
- Monitoring database performance and efficiently
- Transferring data between the database and external file

• Monitoring and re-establishing database consistency

E-R model:

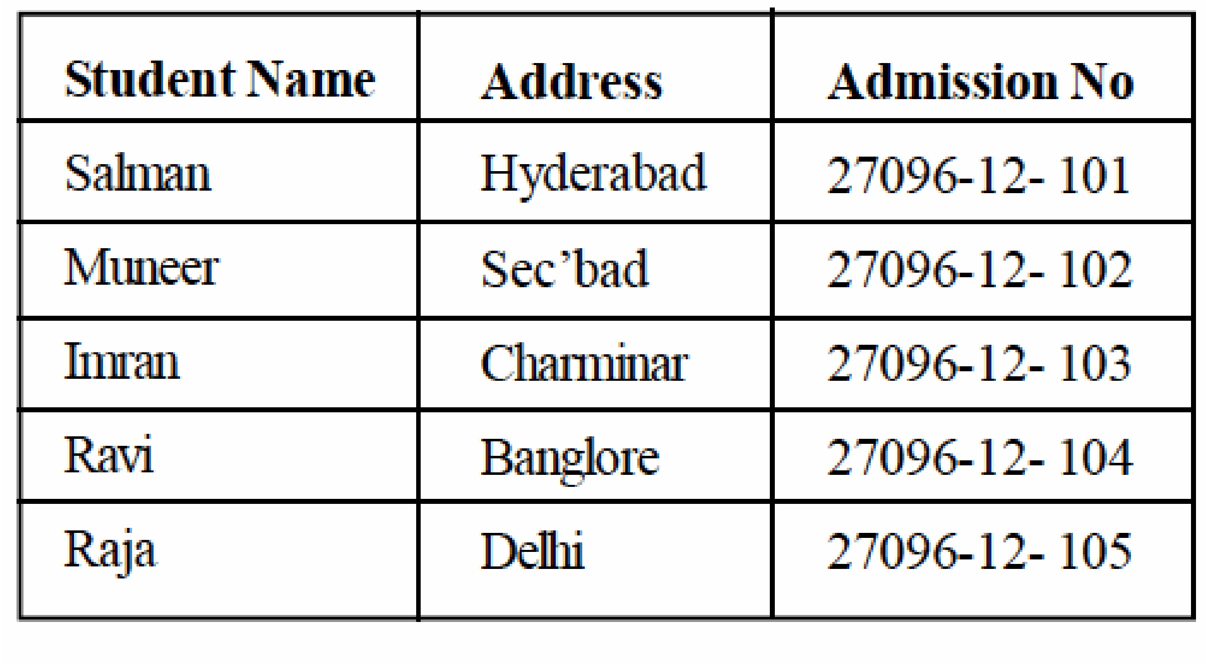
An **entity relationship model**, also called an **entity-relationship** (**ER**) **diagram**, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems.

Entity Sets

The Entity Set is a set of entities of the same type, that share the sameproperties or Attributes.

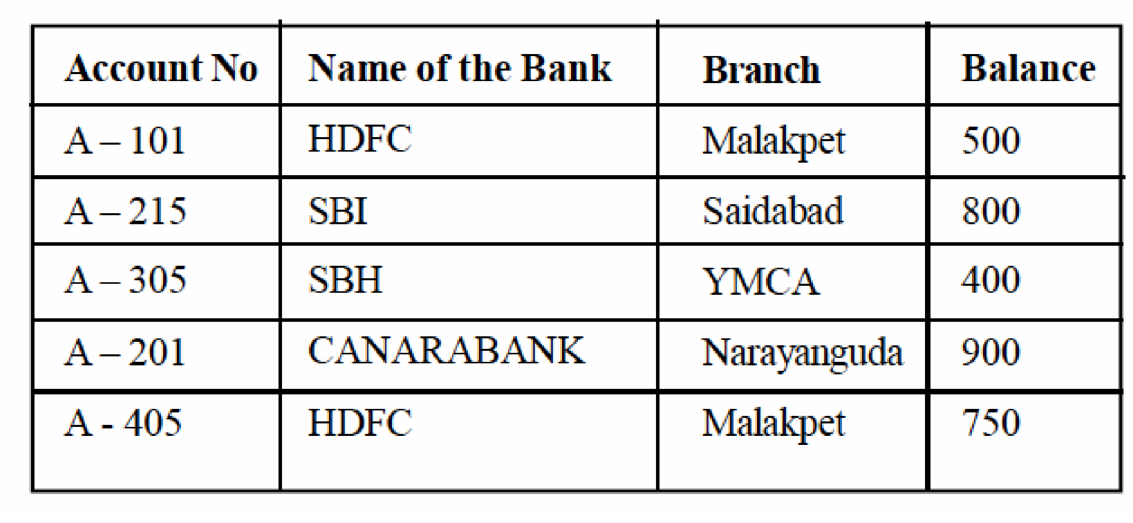
Ex : Persons having an account at bank. or Eachstudent having Admission Number.The set of all students in a class., can be defined as the entity-set. Similarly, the entity set Admission might represent the set of all admissionawarded by a particular university.

Example : Student Table



Student Name	Address	Admission No
Salman	Hyderabad	27096-12- 101
Muneer	Sec'bad	27096-12- 102
Imran	Charminar	27096-12- 103
Ravi	Banglore	27096-12- 104
Raja	Delhi	27096-12- 105

Bank Account



Account No	Name of the Bank	Branch	Balance
A – 101	HDFC	Malakpet	500
A – 215	SBI	Saidabad	800
A – 305	SBH	YMCA	400
A – 201	CANARABANK	Narayanguda	900
A - 405	HDFC	Malakpet	750

Attributes : Each entity has certain characteristics knows as attributes.

For instancethe student entity might include the following attributes, Student name, RollNumber etc. For each attribute, there is a set of permitted values, called thedomain, or value set, of that attribute. An attribute of an entity set is a functionthat maps from the entity set into adomain. Since an entity set may have severalattributes, each entity can be described by a set of (attribute, data value) pairs,one pair for each attribute of the entity set.

The attributes can be classified into

1. Simple attributes

2. Complex/ composite attributes

3. Single – valued attributes

4. Multi - valued attributes

5. Derived attribute

6. Null Attribute

An attribute, as used in the E – R model, can be characterized by thefollowing attribute types.

• **Simple attributes**: The attributes have been simple; that is, theyhave not been divided into subparts.

Example : Student class Roll Number.

• **Composite attributes** : The attributes, which can be sub divided into sub parts.

Example : Student Name, Which can be divided in parts like Firstname, Middle name and Last name. Note also that a composite attribute may appear as a hierarchy. In the composite attribute address, its component attributestreet can be further divided into street_number, street_name, and Door _ numberetc.

• **Single valued attributes** : The attribute which contain/ accept onlyone value/character.

ExampleSex : Male or Female

Marital status : Married or Unmarried

• **Multivalued attributes** : The attributes which has set of values for

a specific entity. in our example all have a single value for a particular entity.

Example 1 : Number of dependents in a family may 0,1,2,3,4…..

Example 2 : A student may have several phone numbers, and differentstudents may have different numbers of phones.

• **Derived attribute** : The value of this type of attribute can be derivedfrom the values of other related attributes or entities. The value of a derivedattribute is not stored but is computed when required.

• **Null Attributes** : An attribute takes a null value when an entitydoes not have a value for it. The null value may indicate "not applicable" – thatis, that the value does not exist for the entity.

Relationship Sets

A Relationship is an association among several entities. For example,we can define a relationship between student entity and bank account entity.The student named Rahul having bank account in HDFC ,Malakpet branchwith an account number A-101.

Mapping Constraints

Mapping Cardinalities, or cardinality ratios, expressthe number of entities to which another entity can be associated via a relationshipetc.

Mapping Cardinalities are most useful in describing binary relationshipsets, although they can contribute to the description of relationship sets thatinvolve more than two entity sets.

For a binary relationship set R between entity sets A and B, the mappingcardinality must be one of the following.

There are 4 types of mapping cardinalities.

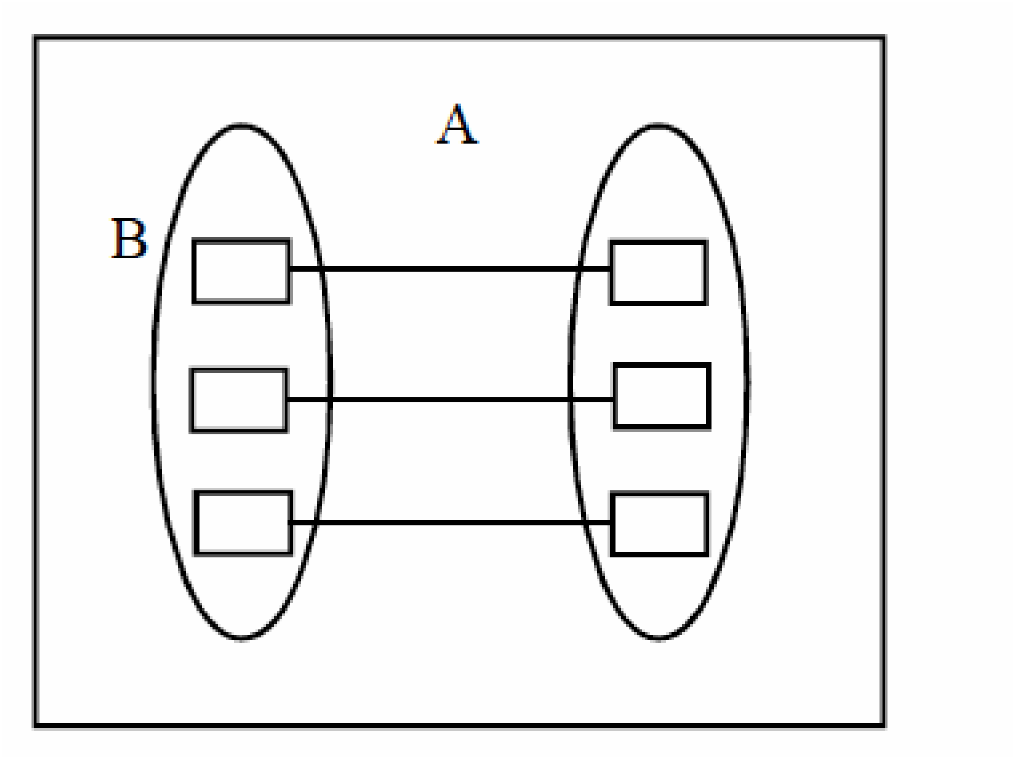
1. ONE – to – ONE Relationship

2. MANY – to – MANY Relationship

3. ONE – to – MANY Relationship

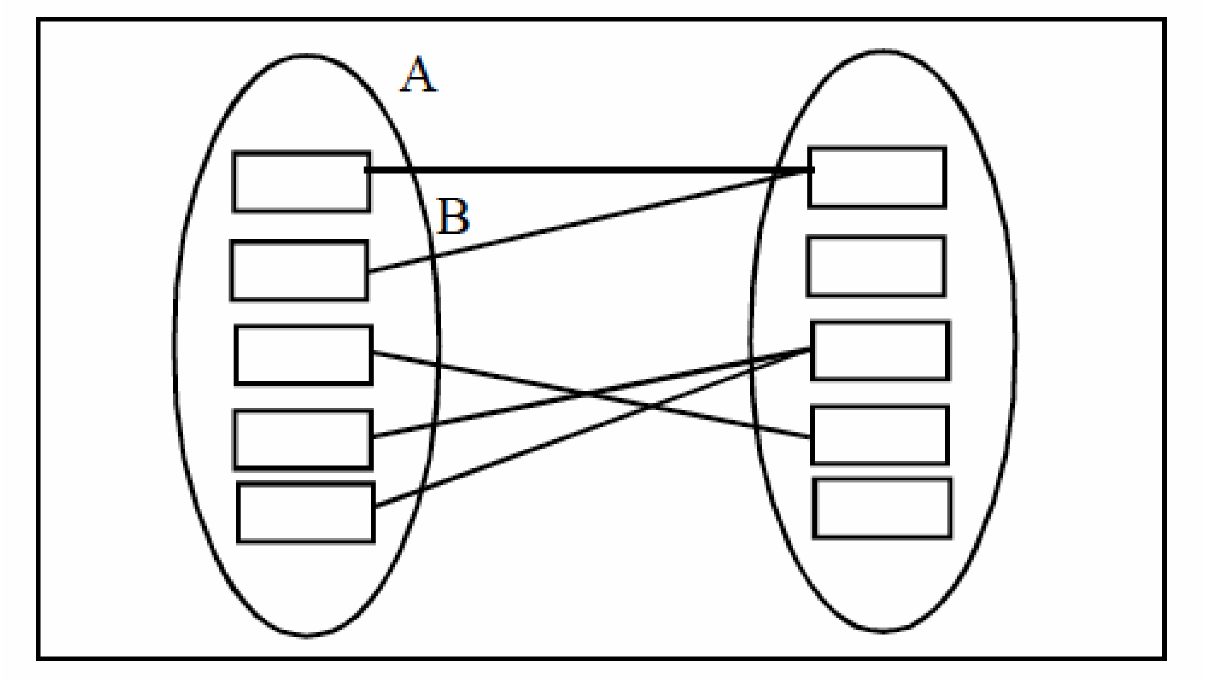
4. MANY – to – MANY Relationship

1. ONE – to – ONE Relationship : An entity in A is associated with atmost one entity in B is also associated with at most one entity in A.



Example : Relationship between the entities principal and college. i.e.,Principals can lead a single college and a principal can have only one college

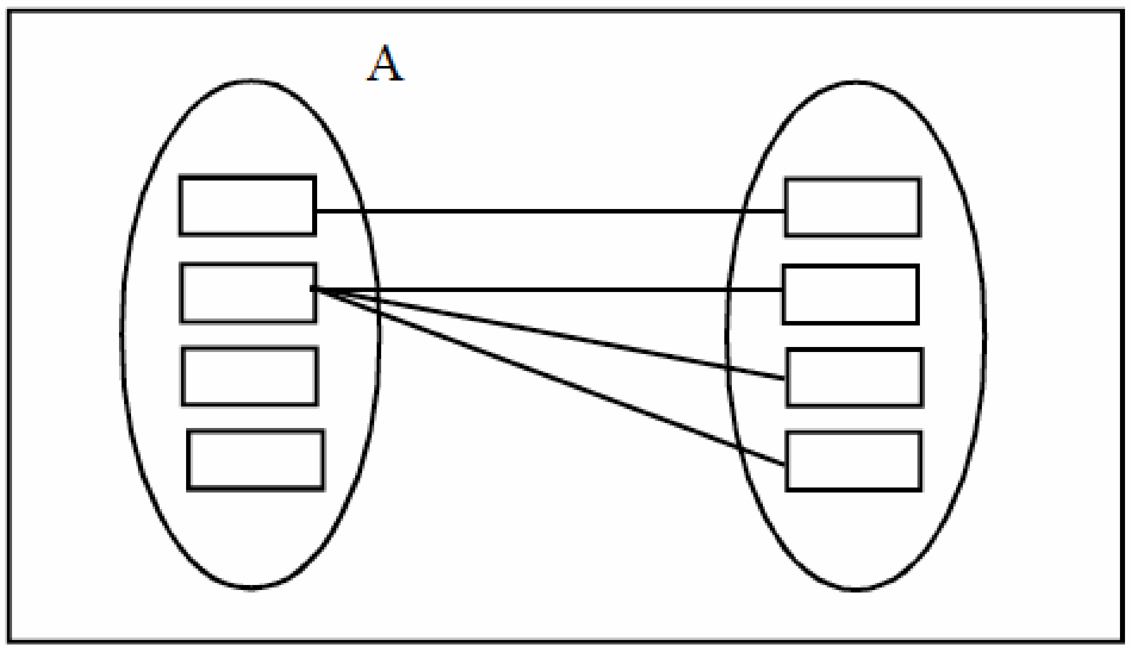
2. Many – to – One Relationship : An entity set in A is associatedwith at most one entity in B, An entity in B however can be associated with anynumber of entities in A.



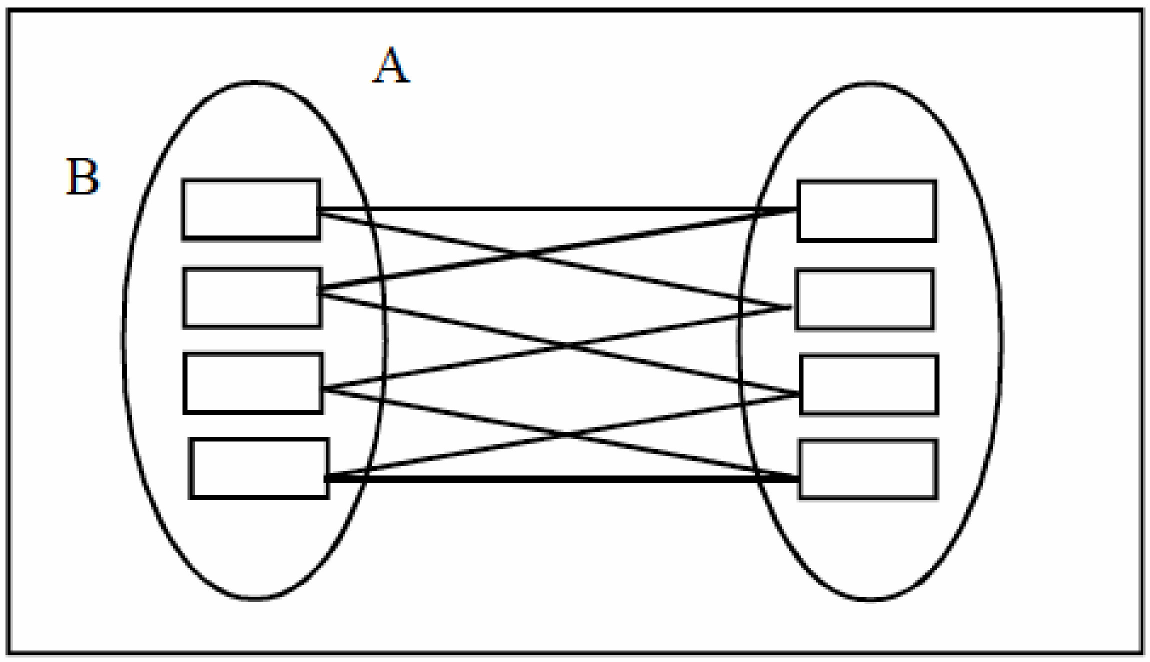
Example : Relationship between the entities Districts and state .i.e.many districts belong to a single state but many states cannot belong to singledistrict.

3**. ONE – to - MANY Relationship** : An entity set A is associatedwith any number of entities in B. An entity in B, however can be associated withat most one entity in A.

Example : Relationship between the entities class and student i.e., aclass can have many students but a student cannot be in more than one class ata time.



4. **MANY – to – MANY Relationship**: An entity set in A is associatedwith any number of entities in B and an entity set in B is associated with anynumber of entities in A.



Example : Relationship between the Entities College and course .i.e. acollege can have many courses and course can be offered by many collegesThe appropriate mapping cardinality for a particular relationship setobviously depends on the real – world situation that the relationship set ismodeling.

An E-R diagram can express the overall logical structure of a databasegraphically. E-R diagrams are simple and clear – qualities that may well account in large part for the widespread use of the E-R model. Such diagram consists ofthe following major components.

An E-R diagram can express the overall logical structure of a databasegraphically. E-R diagrams are simple and clear – qualities that may well account in large part for the widespread use of the E-R model. Such diagram consists ofthe following major components.

• Rectangles : Which represent entity sets.

• Ellipses : Which represent attributes

• Diamonds : Which represent relationship sets

• Lines : Which link attributes to entity sets and entity sets to relationshipsets

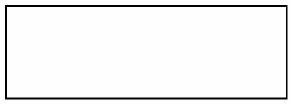
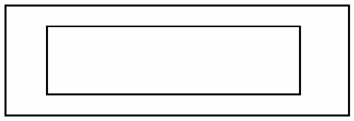
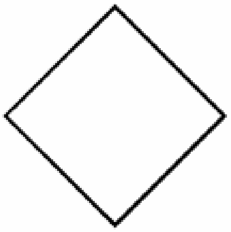
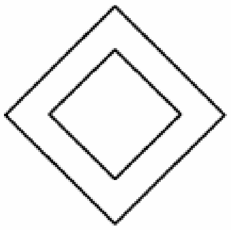
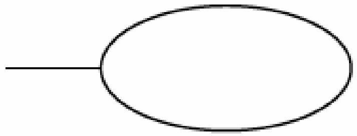
• Double ellipses : Which represents multivalued attributes

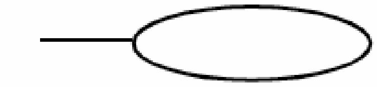
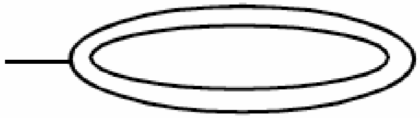
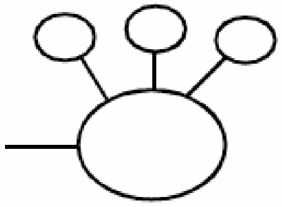
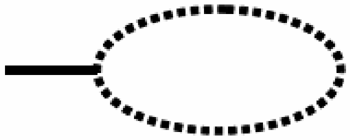
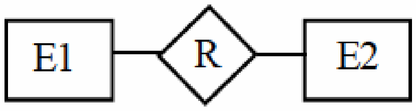
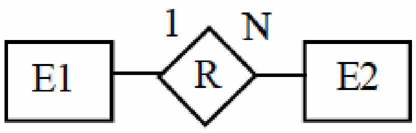
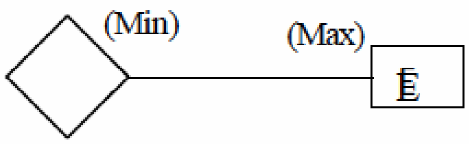
• Dashed ellipses : Which denote derived attributes.

• Double Lines : Which indicate total participation of an entity in arelationship set.

• Double Rectangles : Which represent weak entity sets

Symbols used in E-R diagram representation

Symbol	Represented ERD Property
	**Entity**
	**Weak Entity**
	**Relation Ship**
	**Identifying Relationship**
	**Attribute**

	**Key Attribute**
	**Multivalued Attribute**
	**Composite Attribute**
	**Derived Attribute**
	**Total Participation of E 1 in R**
	**Cardinality Ratio 1: N for E 1, E2 in R**
	**Structural constraint (Min, Max) on Participation of E in R**

Steps to draw ER diagram::

Example: Construct an ER diagram for a car-insurance company whose customer own one or more cars each. Each car has associated with it zero to any number of recorded accidents

1.identify entities

2. Identify relationships using relationship matrix

3.draw rough ER diagram

4.describe relationship (Cardinality)

5.identify essential attributes and Primary key

6. Draw actual diagram

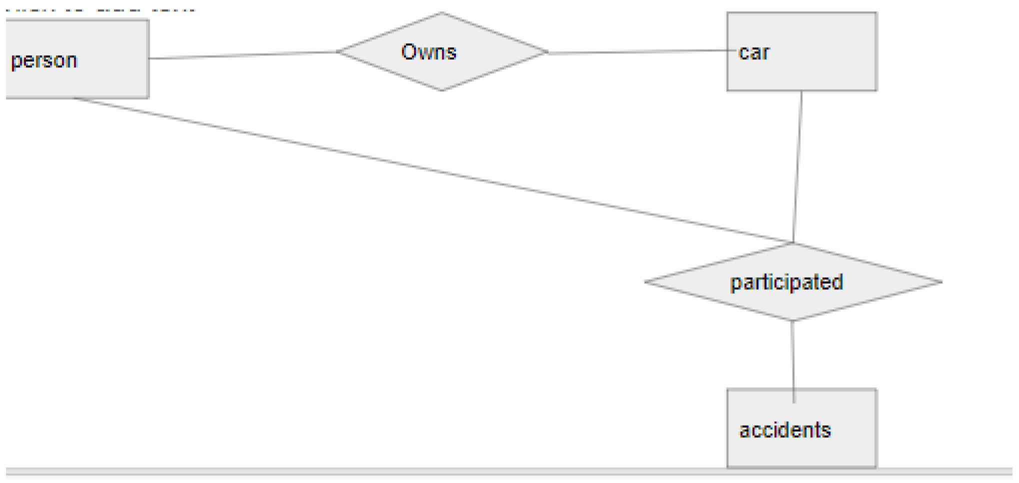
Identify entities

- Person
- Car
- Accidents

Identify relationships using relationship matrix

	Person	**car**	**accidents**
person	-	owns	participated
car	Owns	-	participated
accidents	Done by	Done by	-

draw rough ER diagram



describe relationship (Cardinality)

1. Person own one or more cars
2. Car own by only one person
3. Car has associated with zero to many accidents
4. Accident is done by only one person

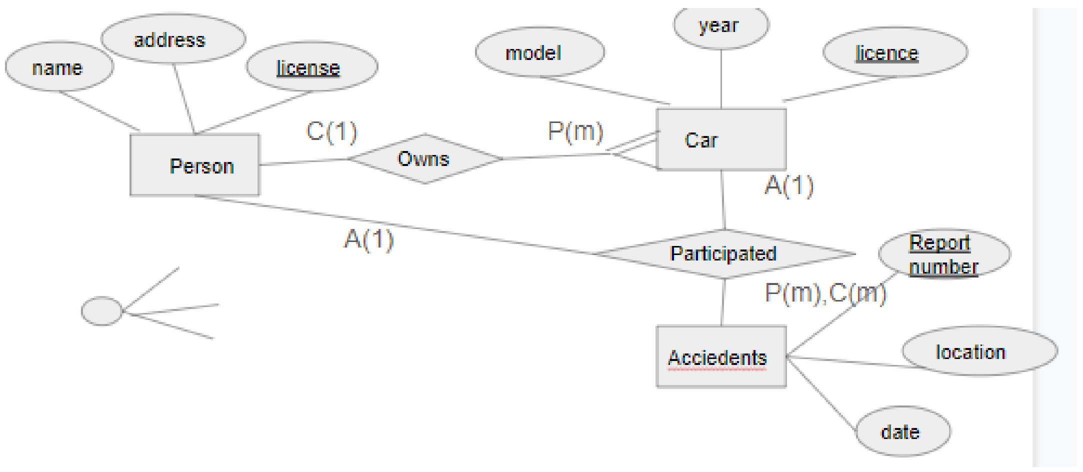
Identify essential attributes and Primary key

1.person -name,address,<u>licence-id</u>

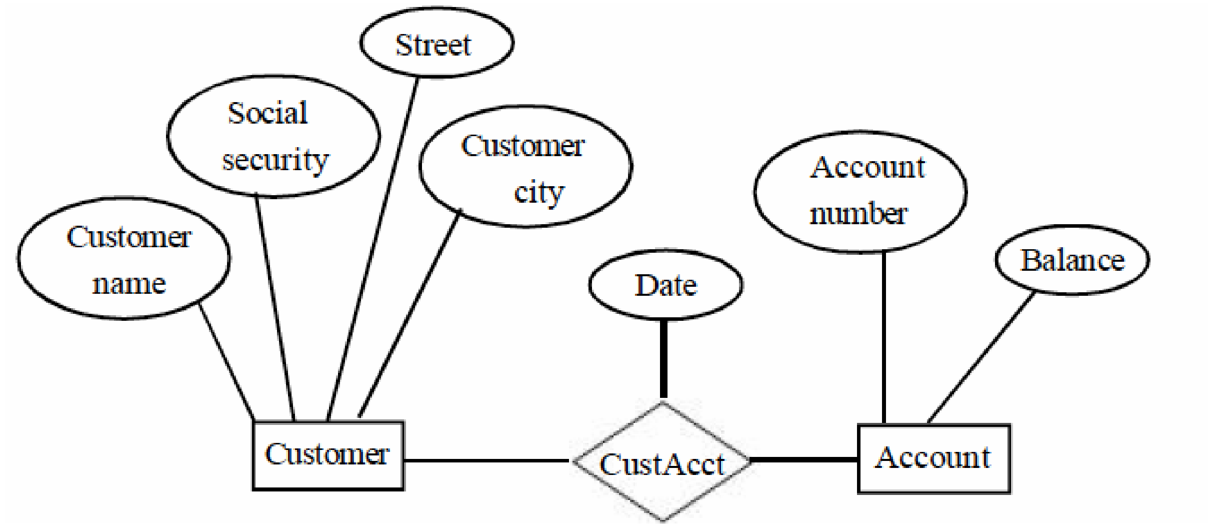
2. Car- model,year,<u>licence</u>

3.accidents- location, date, <u>report number</u>

Draw actual diagram

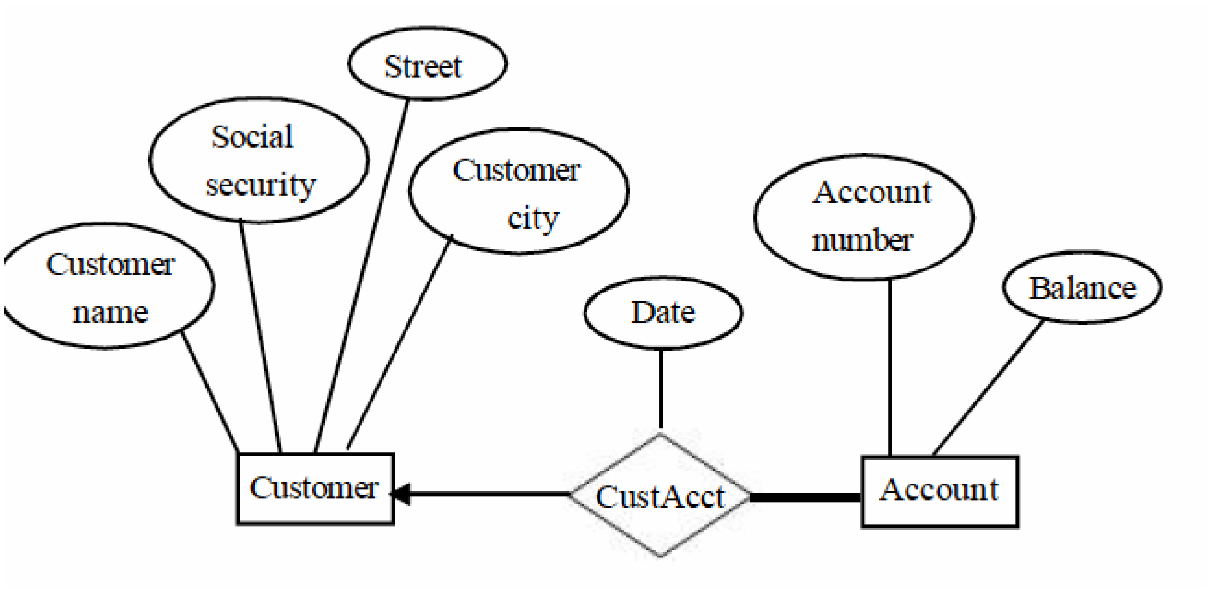


Example I - ER diagram with the entity sets Customer (Customer-name, Socialsecurity,Street, Customer-city), Account (Account-Number, Balance with arelationship Custacct (date), i.e. date is attribute of relationship as shown below.



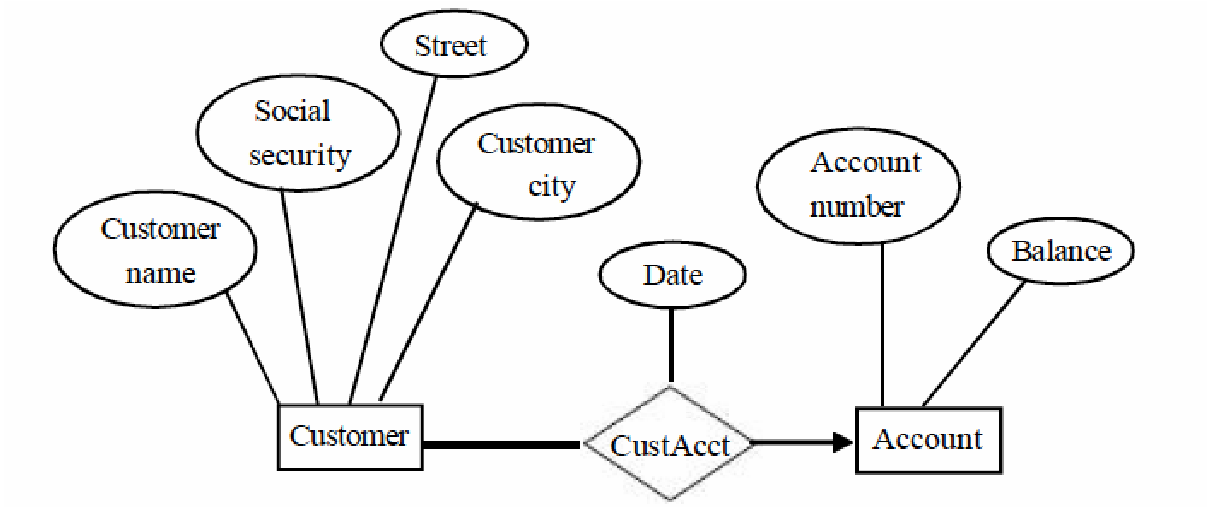
Example : 2

ER diagram showing the cardinality of the customer account relationshipas one-to-many is as shown below.



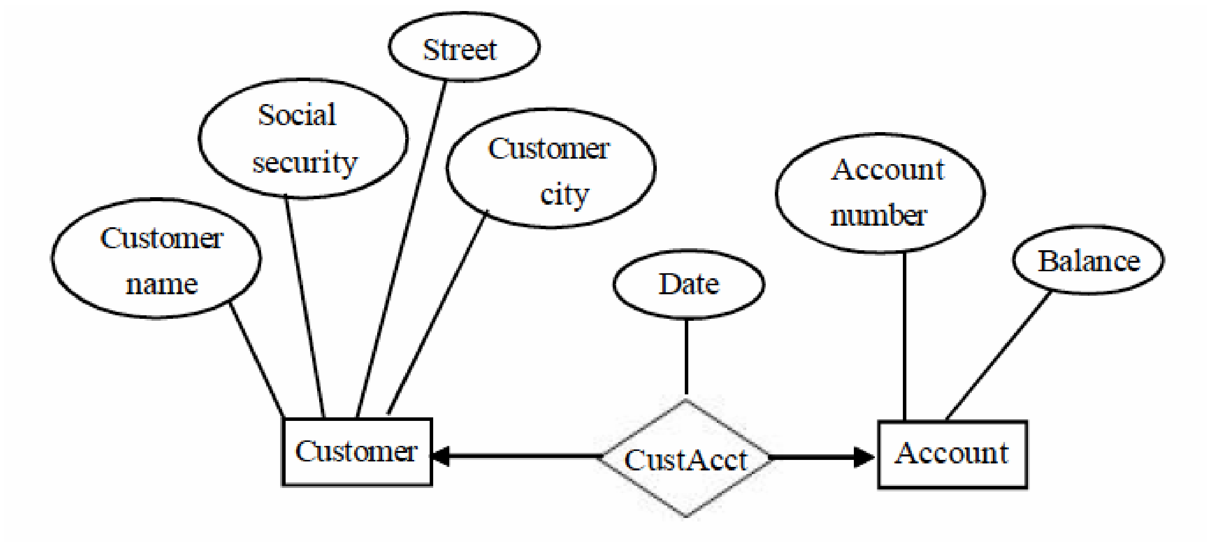
Note : Oserve an arrow mark towards customer on the line joiningcustomer and custacc. i.e, arrow mark indicates the cardinality one, on arrowmark indicates many.

Example : 3ER diagram showing the many - to one relationship between customerand account is as shown below. (Observe the arrow mark is towards account)



Example : 4

ER diagram showing the one -to one relationship between customerand account is as shown below.(Observe the arrow mark is towards accountboth customer as well as account).

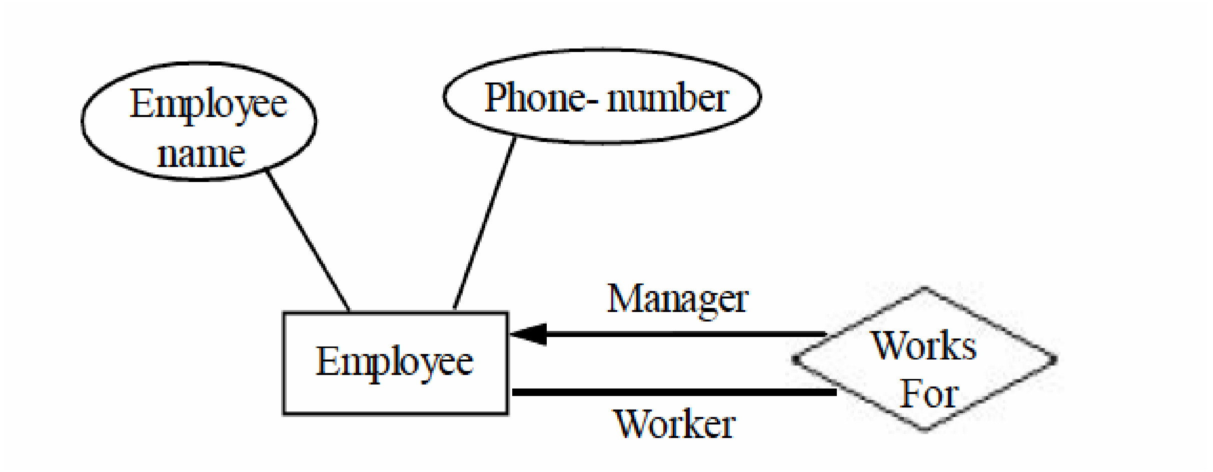


Note : In the first ER diagram, the cardinality is many - to -many, because there are no arrows towards both the entity sets.

Role in E-R DiagramsThe function that an entity plays in a relationship is called its role. Rolesare normally explicit and not specified.They are useful when the meaning of a relationship set needs clarification.For example, the entity sets of a relationship may not be distinct. Therelationship works - for might be ordered pairs of employee (first is manager,

second is worker).

In the E-R diagram, this can be shown by labeling the lines connectingentities (rectangles) to relationships (diamonds).



Weak Entity sets E-R Diagrams

Weak entity set : An entity set that does not possess sufficient attributesto form a primary key is called a weak entity set.

Strong Entity set : An entity set that has a primary key is called astrong entity set.

For example

The entity set transaction has attributes transaction - number, date andamount. Different transaction on different accounts could share the same number.These are not sufficient to form a primary key (uniquely identify a transaction).Thus transaction is a weak entity set.

For weak entity set to be meaningful, it must be part of a one – many relationships set. This relationship set should have no descriptive attributes.The idea of strong and weak entity sets is related to the existencedependencies seen earlier.

Member of a strong entity set a dominant entity. Member of a weakentity set is a subordinate entity.

A weak entity set does not have a primary key, but we need a means of distinguishing among the entities. The discriminator of a weak entity set is a setof attributes that allows this distinction to be made.The primary key of a weak entity set is formed by taking the primary

key of the strong entity set on which its existence depends (see MappingConstraints) plus its discriminator.

Example : 5

Transaction is a weak entity. It is existence - dependent on account.The primary key of account is account - number.Transaction - number distinguishes transaction entities with in the sameaccount (and is thus the discriminator).So the primary key for transaction will be (account - number, transaction

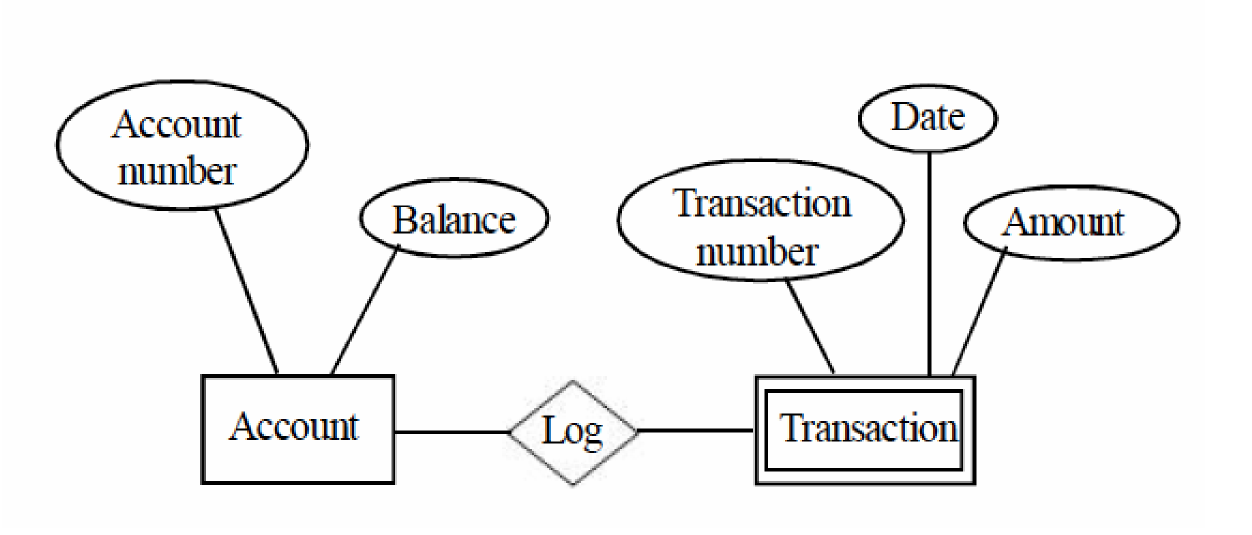
- number).

Note : The primary key of a weak entity is found by taking the primarykey of the strong entity on which it is existence - dependent, plus the discriminatorof the weak entity set.

A weak entity set is indicated by a doubly - outlined box. For example,the previously - mentioned weak entity set transaction is dependent on the strongentity set account via the relationship set log.

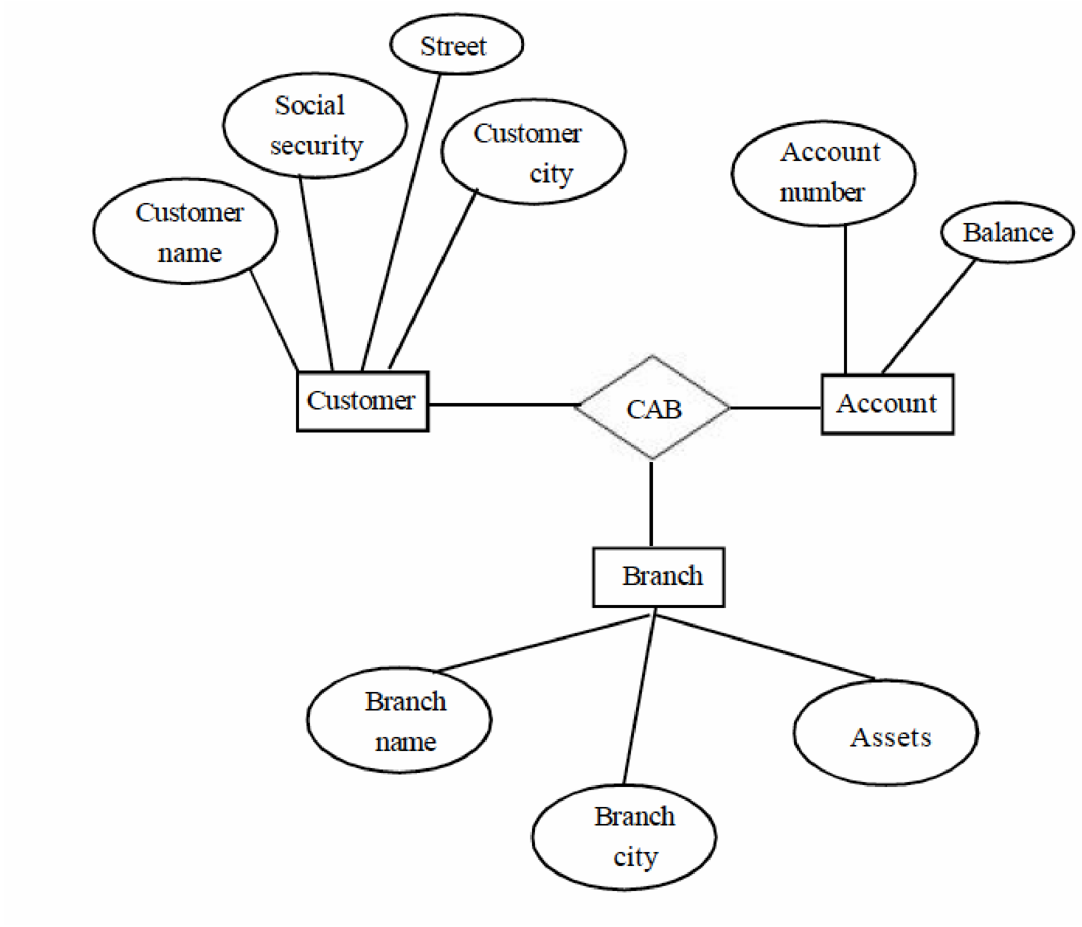
Example : 6

E-R diagram of weak entities. Observe that transaction is placed indouble rectangle.



Example : 7

Non - binary ER diagram (ternary) between three entity sets a customer,Account, Branch is as shown below which says that a customer may have severalaccounts, each located in a specific bank branch, and that an account maybelong to several different customers.


Street
Social security
Customer city
Customer name
Account number
Balance
Customer
CAB
Account
Branch
Branch name
Assets
Branch city