# BASIC CONCEPT OF DBMS

## DATA: -

Data are the known parts can be recorded and have an implicit meaning.

## DATABASE: -

Database is a repository form of related information.

- Simply a database is consisted of some related information.
- A database is consisted of so many tables.
- A table consist of so many rows and columns.
- In database the columns are known are field and rows are known as record.
    - Or
- In the other words a database consisted of so many tables which are in the form of fields and records.

Ex- Dictionary, Student database etc.

## DATABASE MANAGEMENT SYSTEMS: -

- Database management system (DBMS) is a software package or a set of programs which is used to manipulate the database.
- Manipulate means create, insert, modify, alter and delete etc.

Ex- FoxPro, Ms access, Database (D BASE) etc.

# 1$^{ST}$ CHAPTER (BASIC CONCEPT OF DBMS)

## PURPOSE OF DATABASE SYSTEM: -

- To overcome the following problems, we use a database-

i. Reduce the data redundancy and inconsistency.
ii. Difficulty in accessing data.
iii. Data isolation.
iv. Concurrent access anomalies.
v. Atomicity problems.
vi. Integrity problems.
vii. Security problems.

## REDUCE THE DATA REDUNDANCY AND INCONSISTANCY: -

- Different programmers create the file and application programs.
- If the various files are likely to Have different structure and the programs may be written in several program language
- The same information may be duplicated in several places or files. The duplication of data leads to data redundancy.
- The redundancy leads to higher storage and access cost.
- It may lead to data consistency that tends to be present in redundant data files.
- The centralised control of data by the DBA (database administrator) avoids on access duplication of data and reduces the total amount of data storage required.
- The advantages of avoiding duplication is elimination of inconsistency.
- Any redundancy that exists in the DBMS is controlled and the system ensures that these multiple copies are consistent.

## DIFFICULTY IN ACCESSING DATA: -

- The conventional life processing environment does not allow needed data to be retrieved in a convenient and efficient manner.
- More responsive data retrieved systems are required for general use.

## DATA ISOLATION: -

- Writing a new application program to retrieve the appropriate data is difficult because data scattered in various fields may be in difficult formats.

## CONCURRENT ACCESS ANOMALIS: -

- For the purpose of overall performance of the systems and faster response many systems allow multiple users to update the data simultaneously.
- Today the largest internet retailers may have millions of accesses per day to their shoppers.
- For this cause interaction of concurrent updates is possible and this may lead to Inconsistency data.
- To avoid such problems the system must maintain some form of supervision but supervision is difficult to provide because data may be accessed by many difficult application programs that have not been coordinated previously.

## ATOMICITY PROBLAMS: -

- Like any other mechanical and electrical device, a computer system is also subjected to failure.

- In many applications it is crucial that if a failure occurs, the data be restored to the consistent state that existed prior to the failure.
- The font transfer must be atomic; it must happen entirely or not at all.
- It is difficult to ensure atomicity in a conventional file processing system.

Ex-

Let us consider a program to transfer 150 from account to account be if a system error occurs during the execution of program, then it is possible that the 150 was removed from account air what was not get it to account be resulting in consistency in data pasted it is essential to database consistency that either both the credit and debit for that neither access.

## INTEGRITY PROBLEMS: -

- The data values stored in the database must satisfy certain types of consistency constraints.
- Developers force date constraints in the system by adding appropriate code in the various application programs.
  When new constraints are added it is difficult to enforce them.
- It is compounded when constraints involve several data items from difficult files.

## SECURITY PROBLEMS: -

- Every user of the database system should not be able to access all the data.
- the application programs are added to the file processing system in an ad HOC manager coma enforcing sorts security constraints is difficult.

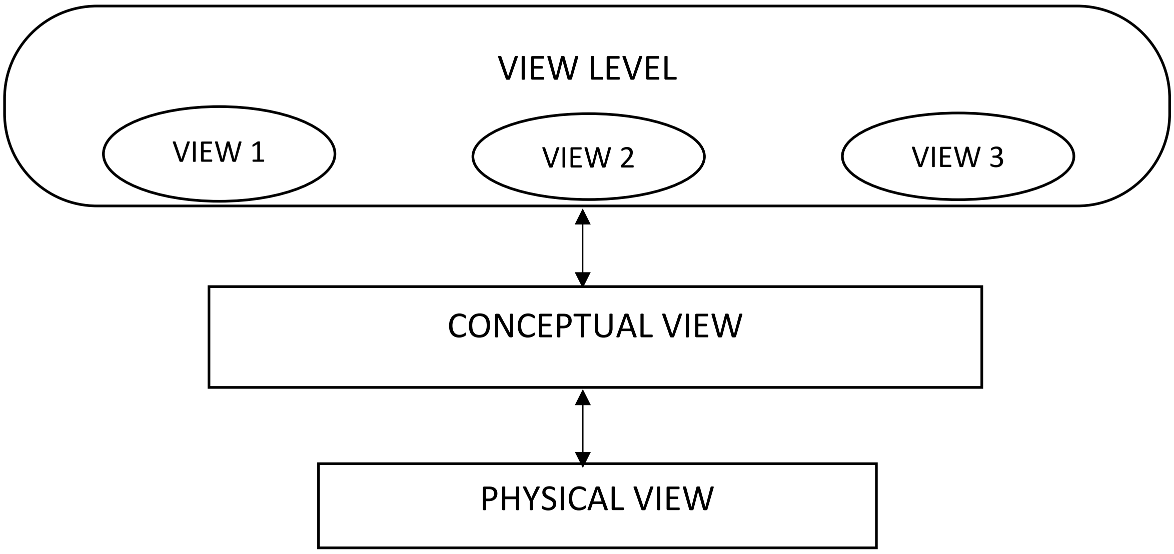
## DATA ABSTRACTION: -

- It is the way of representing data without including its background details.
- The characteristics that allow programs that are and programs operate in independence is called data abstraction.
- Database system provides the user an abstract view of the data.
- System Heights software details of how the data are stored and maintained.
- To simplify the user's interaction with the system developers, hide the complexity from the user through several levels of abstraction.
- Data can be abstracted by 3 levels: -

1. Physical level
2. Conceptual level
3. View level

## 1.PHYSICAL LEVEL: -

- Physical level is the lowest level of the data abstraction.
- It is the closest label to the database.
- It describes how the data is actually stored.
- The actual database is available at the physical level.
- Physical level describes Complex level structure in detail.
- We cannot see the physical level of data in the database.



**(THREE LEVELS OF DATA ABSTRACTION)**

## 2.CONCEPTUAL LEVEL: -

- The next higher level of data abstraction is conceptual level.
- This level is used to link physical level and view level.
- This level describes What data are stored in the database, What relationship exists among those data.
- This conception level hides the actual data of the database.
- Database administrators use the conceptual level of data abstraction.

## 3.VIEW LEVEL: -

- View level is the higher level of the data abstraction.
- It describes only a part of the database.
- Users can work at this level.
- View level of data structures exists to simplify the user interaction with the system.
- The system may provide many views for the same database.
- Users can see the copy of the database in the level but cannot see the actual data.
- It is also known as the usual level or external level.

## DATABASE USERS: -

- The person who handles/interacts with the database is known as database user.
- There are different types of database users.
- The user is different by the way expert to interact with the system. The users are: -

a) Application programmer
b) Sophisticated User
c) Specified User
d) Naive User/End user

## APPLICATION PROGRAMMER: -

- The computer professional program or who is responsible for developing applications are known as application programmers.
- Application programmers can choose from tools to develop user interfaces.
- Application programs of code be returned in purpose programming language, such as documents required to manipulate the database.

- Rapid application program development tools are the tools that allow an application program to construct form and reports with minimum programming efforts.

## SOPHISTICATED USERS: -

- The user who interacts with the system without writing programs is known as a sophisticated user.
- Sophisticated users submit their request in a database query language.
- They submit each such query to the query processor.
- The function of the query processor is to break DML statements into instruction.
- Ex: -DBA (Database Administrator).

## SPECIALISED USER: -

- Specialised users other sophisticated users who write specialised database applications that don't fit into the traditional data processing form work.
- Search type of applications are Computer aided design system, knowledge-based and expert system that stored data with Complex data types and environment modelling system.

## NAIVE USER/END USER: -

- Naive User on sophisticated users who interact with the system by involving one of the application programs that has been written previously.
- Naive users simply read reports generated from the database.
- Naive user is also known as an end user.
- End users of the database works through a menu-oriented application program where the type and range of recent is always indicated to the user.

## DATA DEFINITION LANGUAGE: -

- The Structured query language (SQL) of three types: -

I. Data Definition Language (DDL)
II. Data Manipulation Language (DML)
III. Data Control Language (DCL)

## DATA DEFINITION LANGUAGE(DDL)(Create, Alter, Drop):-

- The language which defines the database is known as data definition.
- Data definition language is also used for additional properties of data.
- These definitions include all the entity sets, associate attributes as well as the relative among them.
- Just like any other programming language DDL, guess as input some instructions are some outputs.
- The output of DDL placed in the data.

## DATA MANIPULATION LANGUAGE (DML) (Update, Insert, Select, Delete):-

- The language used to manipulate the database is called data manipulation language.
- data manipulation in valves retrieval the database, insertion of new data in database and deletion or modification of data in database.
- DML Provides commands to select and from the database.

## DATA CONTROL LANGUAGE(DCL): -

- Data control language is used to control the database.
- DATA DICTIONARY: -
- Data dictionary is a database itself.
- The Information about the structure of the database the uses of the contain in the database. The data is maintained in a data dictionary.
- The Data Dictionary contains detailed information which was stored in the database.

- The Detailed information about the database or the structure of the database is known as a data dictionary.
- Note: -Collection of metadata is known as a data dictionary.

## ADVANTAGES OF DBMS: -

- Data redundancy and inconsistency are avoided.
- Data can be easily accessed.
- Data integrity is maintained.
- Atomicity properties satisfied.
- Security is applied.

## DISADVANTAGES OF DBMS: -

- Cost of Software and Hardware migration.
- Complexity of backup and recovery.

# 2ND CHAPTER

# DATA MODELS

## DATA INDEPENDENCE: -

- The ability to modify at schema definition level without affecting the schema define the next higher level is called Data Independence.
- There are two types of levels of data independence.

I. PHYSICAL DATA INDEPENDENCE

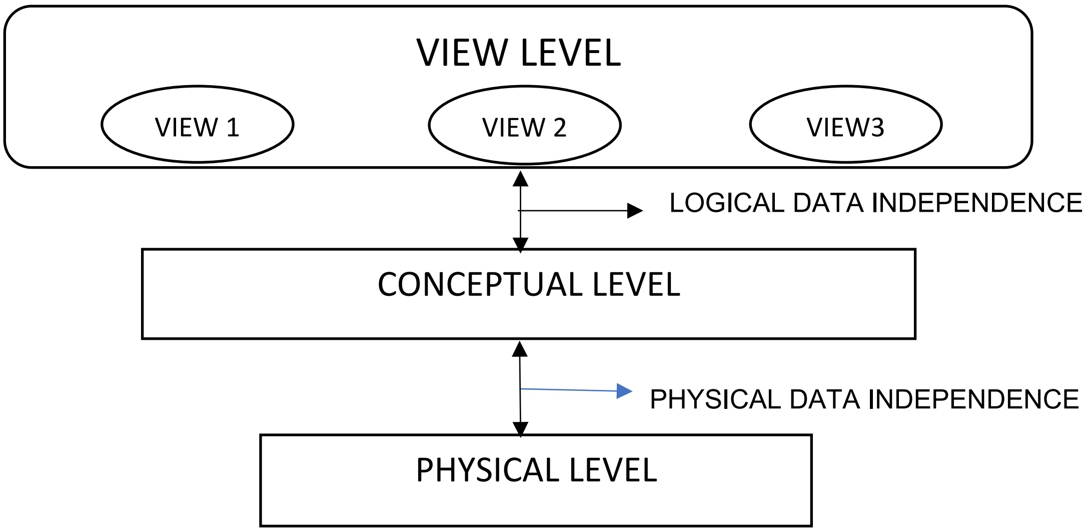
II. LOGICAL DATA INDEPENDENCE

## PHYSICAL DATA INDEPENDENCE: -

- Physical data Independence is ability to the physical schema without changing application programs should be rewritten.
- Modification at the physical level is necessary in order to improve the performance.

## LOGICAL DATA INDEPENDENCE: -

- It is the ability for the conceptual schema without changing application programs to be rewritten.
- Modification of the conceptual level is necessary in the logic of the database.
- Logical data Independence is more difficult application programs are heavily dependent on the logical structure of the data they access.



## ENTITY RELATIONSHIP MODEL: -

- The Entity relationship model consists of entities and the relationship.
- It was developed to facilitate database design by allowing the specification of an enterprise schema.
- The enterprise schema consists of the overall logical structure of a database.
- The E-R model is very useful in mapping the meanings and interaction of the real-world enterprise on a conceptual level.
- The E-R data model employees have three basic options: -

i. ENTITY SET

ii. RELATIONSHIP SET

iii. ATTRIBUTES SET

- It is used for the conceptual design of database applications.

## ENTITY: -

- Entity is simply an object or element or item which has existence and that is distinguishable from others.

Or

- Any object which has physical existence that is known as an entity.

Ex: - A person, A bank account, A table etc.

## ENTITY SETS AND RELATIONSHIP SETS: -

### Entity Sets: -

- A group of similar types of entities is called a constituent entity set.
- Let ‘t’ be a entity set then

$t$~$e_1$, $e_2$, $e_3$, ..............$e_n$

$e_1$, $e_2$, $e_3$, .............$e_n$ → all are similar types of entities.

Or

- An entity set is a set of entities of the same type that share the same properties.

Ex: - The Setup on portions to our customers at a given Bank can be defined as an entity set of customers.

- All the individual Bank Customers at the extension of the entity set customers.
- Entity sets do not need to be disjoint.
- An entity is represented by a set of attributes.
- Each entity has a value for each of its attributes.
- The database includes a collection of entity sets each of which contains many numbers of entities of the same type.

| CUSTOMER ID | NAME | STREET | CITY |
|---|---|---|---|
| 321-12-3121 | JONES | MAIN | HARRISON |
| 019-28-3746 | SMITH | NORTH | RAY |
| 677-89-9011 | HAYES | MAIN | HARRISON |
| 555-55-5555 | JACKSON | DUPONT | WOOODSIDE |
| 244-66-8800 | IRFAN | BARODA | GUJARAT |
| 963-96-3963 | DELE | MASSAU | PRINCETON |
| 335-57-7941 | ADAMS | SPRING | PITTSFIELD |
| 943-72-2661 | CURRY | NORTH | RAY |

CUSTOMER

| LOAN NO | AMOUNT |
|---|---|
| L-17 | 1000 |
| L-23 | 2000 |
| L-15 | 1500 |
| L-14 | 1500 |
| L-19 | 500 |
| L-11 | 900 |
| L-16 | 1300 |
| L-20 | 3000 |

LOAN

**(ENTITY SET OF CUSTOMER AND LOAN)**

## RELATIONSHIP SET: -

- A relationship is an association among several entities.
- Group of relationships is known as relationship sets.

Ex: - A relationship that associates the customer Irfan with loan L-19, this relationship specifies that Irfan is a customer with loan number L-19.

- We can have more than one relationship that can be defined over the same set.
- A relationship set is a set of relationships of the same type.

## ATTRIBUTES: -

- Properties or characteristics of an entity are called attributes.
- Attributes are the descriptive properties processed by each member of an entity set.
- Simply the characteristics which distinguish entities from the others, those characteristics are called attributes.

Ex: - The possible attributes of customer entity set or customer ID, customer name, street and costumer city etc.

- There are mainly four types of attributes: -

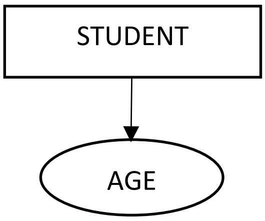
I. Simple and Composite Attributes
II. Single value and Multi valued Attributes
III. Derived Attributes
IV. Null Attributes

## SIMPEL AND COMMPOSITE ATTRIBUTES: -

- The simple attribute can't be divided into sub-parts.
- On the other hand, composite attributes can be divided into sub-parts.

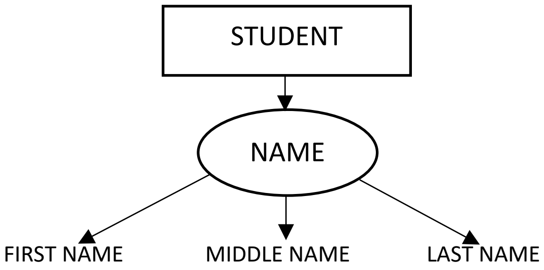
Ex: - Student name could be structured as a composite attribute consisting of first name, middle name and last name.

Ex-



(SIMPLE ATTRIBUTES)

Ex-



(COMPOSITE ATTRIBUTES)

## SINGLE VALUE AND MULTI VALUR ATTRIBUTES: -

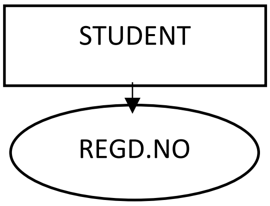
- In a single value attribute, there is a single value for a particular entity.

Ex: - The registration number attributes for a specific student and it refers to only one registration number South Africa beauties are said to be single value.

- In multivalued attributes the attribute has more than two values for a specific entity.

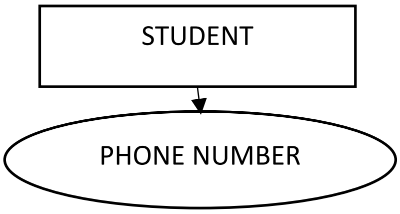
Ex: - Let us consider the student entity set with the attributes phone number. students can have more than one phone number. which type of attributes are said to be multivalued attributes.

Ex-



(EX. OF SINGLE VALUE ATTRIBUTES)

Ex-



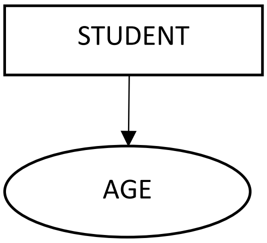
(EX. OF MULTI-VALUED ATTRIBUTES)

## DERIVED ATTRIBUTES: -

- The value for derived attribute can be derived from the value of other related attributes or entities.
- The value of the related attributes is known as a base attribute or stored attribute.

Ex: - Suppose the student entity set has an attribute "age", that indicates the student age. If the student entity also has an attribute "date of birth". Then we can calculate age from "date of birth and the current date".

- This age is a derived attribute and date of birth is the base or stored attributes.



(EX. OF DERIVED ATTRIBUTES)

## NULL ATTRIBUTES: -

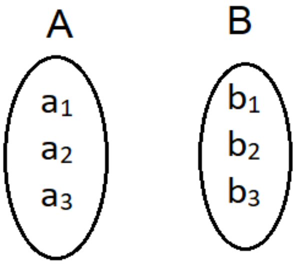
- A null value is used when an entire value does not value an attribute.

## MAPPING CONSTRAINTS: -

- Mapping constraints express the number of entities to which another entity can be associated via a relationship set. Mapping constraints are most useful in describing binary relationship sets, although they are also useful in describing relationship sets that involve more than two entity sets. There may be four types of constraints: -
    1. ONE-TO-ONE
    2. ONE-TO-MANY
    3. MANY-TO-ONE
    4. MANY-TO-MANY

## ONE-TO-ONE: -

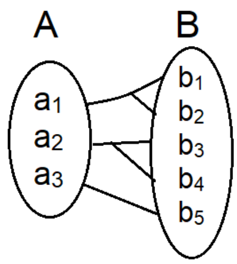
- Let us consider two entity sets 'A' and 'B'. An entity in 'A' is associated with almost one entity in 'B' and an entity in 'B' is associated with almost one entity in 'A'.



(ONE-TO-ONE)

## ONE-TO-MANY: -

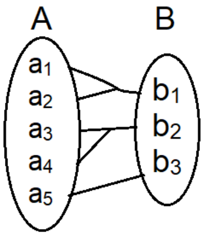
- Let us consider two entity sets ‘A’ and ‘B’. An entity set is associated with any number (0 or more) of entities in ‘B’. An entity in ‘B’ can be associated with one entity in ‘A’.



(ONE-TO- MANY)

## MANY-TO-ONE: -

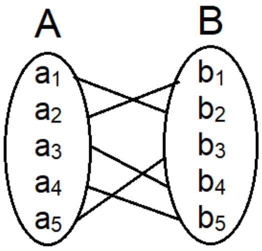
- An entity in ‘A’ is associated with at most one entity in ‘B’. An entity in ‘B’ can be associated with any number of entities in ‘A’.



(MANY - TO - ONE)

## MANY-TO-MANY: -

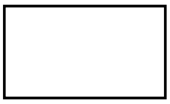
- An entity in ‘A’ is associated with any number of entities in ‘B’ and an entity in ‘B’ is associated with any number of entities in ‘A’.



(MANY - TO - MANY)

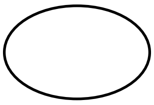
## ER - DIAGRAM: -

- The E-R diagram is the entity relationship diagram.
- It can express the logical structure of a database graphically.
- The E-R diagram consists of following major components.



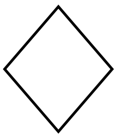
(RECTANGULAR)

The entity Sets are represented by rectangles.



(ELIPSE)

Attributes are represented by ellipse.

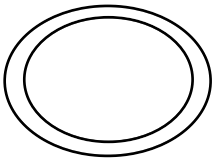


(DIAMONDS)

The relationship sets are represented by Diamond or rhombus.

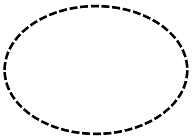
(LINES)

It is used to link attributes to entity sets and entity sets to relationship sets.



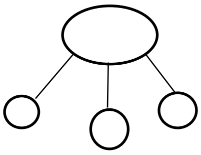
(DOUBLE ELIPSE)

Double ellipse is represented by multivalued attributes.

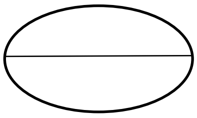


(DOTTED ELIPSE)

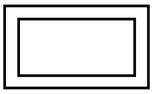
Dotted ellipses are represented by derived attributes.



It represents composite attributes.

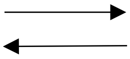


It represents key attributes.



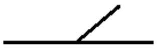
(DOUBLE RECTANGULAR)

It represents the weak entity sets.

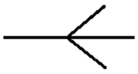


(DOUBLE LINES)

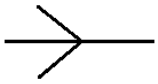
It indicates the total participation sets of an entity in a relationship set.



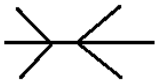
ONE-TO-ONE



ONE-TO-MANY



MANY-TO-ONE



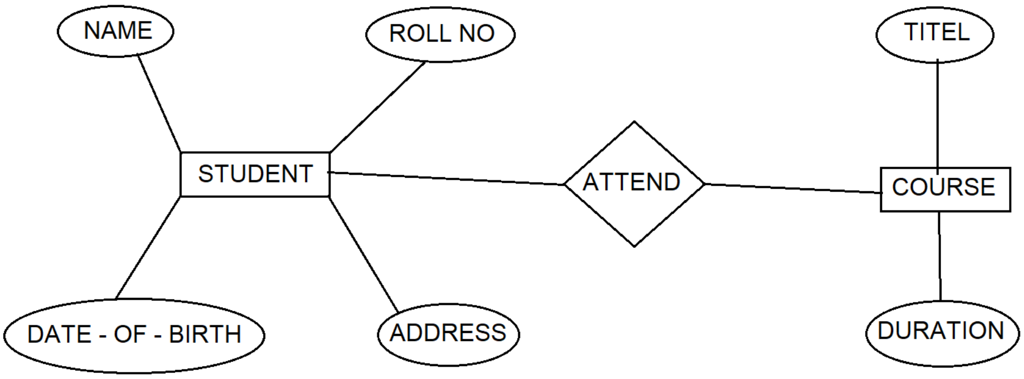
MANY- TO-MANY

- At consider University conduct various question students. Different faculty members are taking different classes for different courses. Different payments are received by the University from the students. Different study loans are also available for the student. Draw an E-R diagram assuming the necessary facts.

| ENTITY | | ATTRIBUTES |
|---|---|---|
| i. University | | |
| ii. Courses | A. Underground | +2, Diploma, ITI |
| | B. Batchelor | +3, B. pharma, degree |
| | C. Master degree | H-tech, MCB |
| | D. PHD | |
| | E. Professional | PGDCA, MSE, DCA, CCNA Management |
| iii. Teacher | | Teacher. Id, teacher -name, engaged-in-course |
| iv. Student | | Roll no, name |
| v. Payment | | Sponsored, Direct, free, pay |
| vi. Loan | | Study loan |

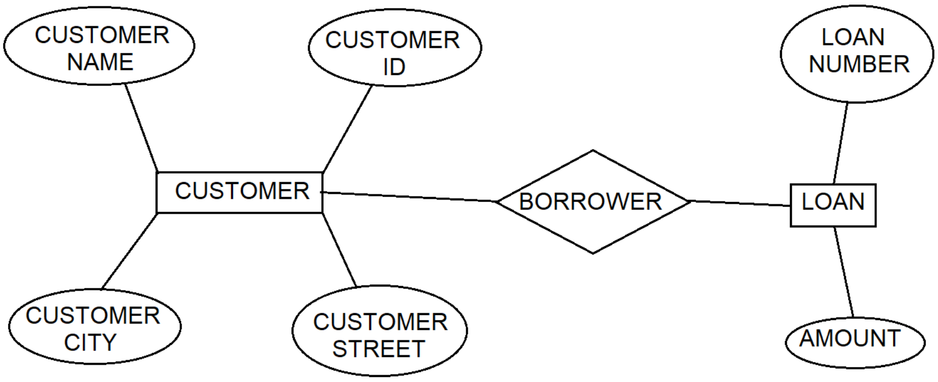
- Consider the ESCI and IT students attending many courses. a set of students have a set of courses. Each student has a name, roll number, address and date-of-birth and each course has title and duration. draw an E-R diagram.

| ENTITY | ATTRIBUTES |
|---|---|
| i. Student | Name, roll no, address, data-of-birth, duration |
| ii. course | Title, duration |



Draw an E-R diagram for customer and loan.

| | ENTITY | ATTRIBUTES |
|---|---|---|
| i. | Customer | Customer -name, Customer Id,<br>Customer – city, Customer Street |
| ii. | Loan | Loan number, amount. |



- The entity-relationship diagram in figure (6.7), which consists of two entity sets, customer and loan related, throw a binary relationship set borrower.
- The attributes associated with customer are customer-ID, customer -Name, customer -Street and customer-City.
- The attributes associated with loan are Loan-number and amount.
- The relationship set borrower may be many-to-many, many-to-one. To distinguish among these types, we draw either a directed line (→) or an undirected line (-) between the relationship set and the entity set.

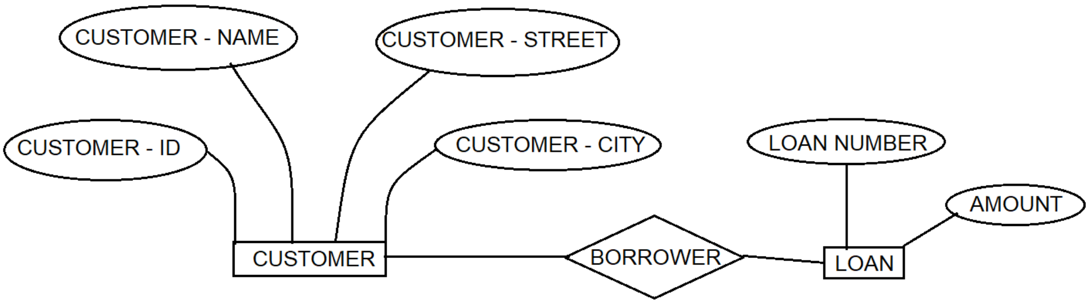


Fig-(6.7) E-R Diagram corresponding to customers and loan

- If the relationship set borrower were one-to-many from customer to loan, then the line from borrower to customer would be directed, with an arrow pointing to the customer entity fig 6.8(a).

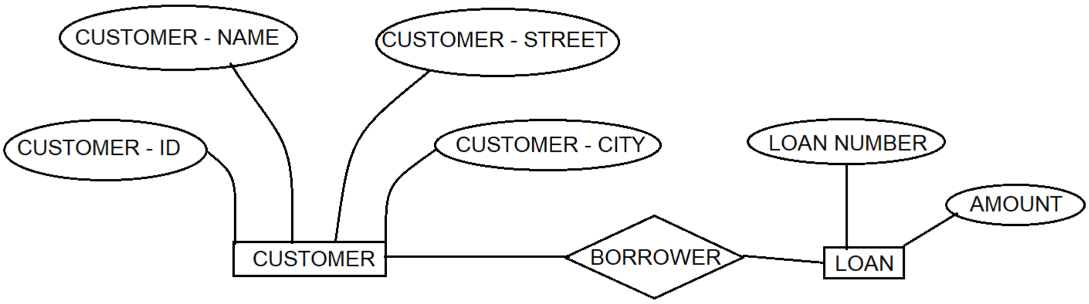


Fig- (6.8) (a)  ( ONE - TO - MANY)

- Similarly, if the relationship set borrower were many-to-one from customer to loan, then the line from borrower to loan would have an arrow pointing to the loan entity figure 6.8 (b).

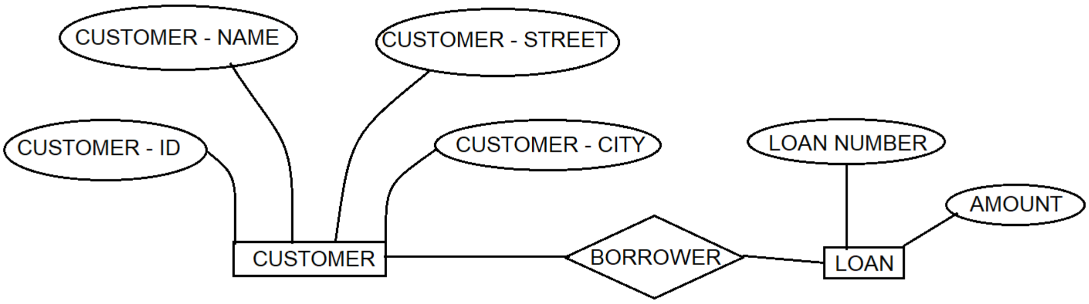


Fig- 6.8 (b) (MANY - TO - ONE)

- Finally, if the relationship set borrower
- Were one-to-one, then both lines from the borrower would have arrows, one pointing to the customer.
- Entity set Fig 6.8(c).

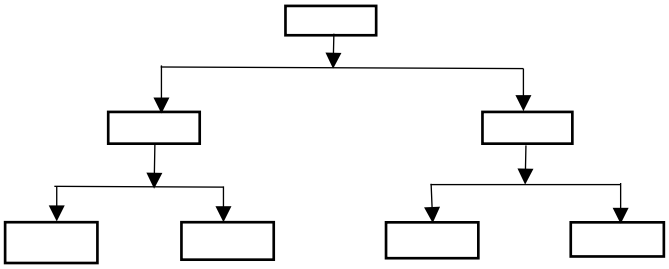
## DATA MODEL: -

- Data model is used to describe the structure of a database and to hide storage details and present the users with a conceptual view of the database.
- In other words, how to store the data in a database is known as a data model.
- There are three types of data model: -

1. Hierarchical data model
2. Network data model
3. Relational data model

## HIERARCHACAL DATA MODEL: -

- Hierarchical data model is nothing but an inverted tree-like structure.
- It represents data and relationships between data items in the model of inverted tree.
- It is the most popular model of the system based on this model.
- It is very simple to construct and operate on simple language.



(STRUCTURE OF HIERARCHICAL DATA MODEL)

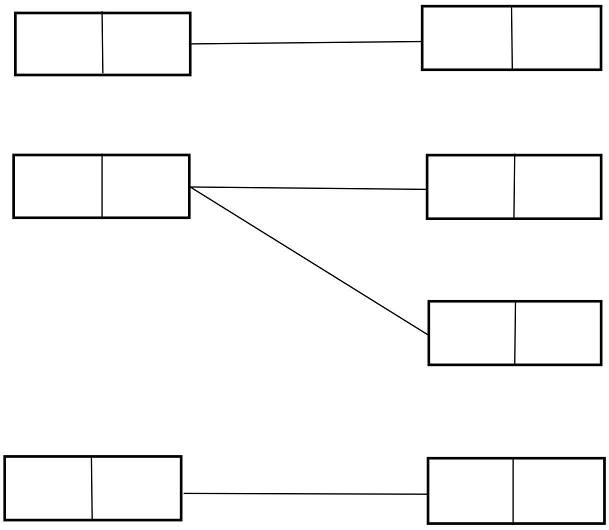
## NETWORK DATA MODEL: -

- It represents data in a network structure.
- In network data model, the data are in link wire.
- It is able to model Complex relationships and represents semantics of add or Delete on the relationship.

- The language is navigational.

Ex:

1. one-to-one
2. one-to-many



(STRUCTURE OF NETWORK DATA MODEL)

## RELATIONAL DATA MODEL: -

- To overcome the drawbacks of hierarchical data models and network data model we use relational data models.
- Relational data model is nothing but a table.
- In the table all the columns are known as attributes.
- All the rows are known as tuples or tools.

|  | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| $R_1$ |  |  |  |  |
| $R_2$ |  |  |  |  |
| $R_3$ |  |  |  |  |
| $R_4$ |  |  |  |  |

Attributes (COLUMANS)

Tuples or tools (ROWS)

(Structure of relational data model)