MONDAY

Date 03/05/2021
Page 1

DATABASE MANAGEMENT SYSTEM (DBMS)

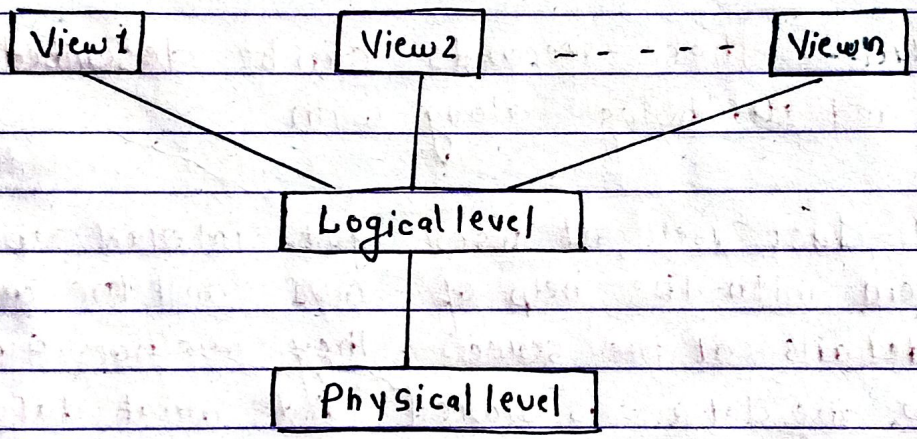
Unit-1:

Database System Architecture.

Data Abstraction in DBMS.

The process of hiding irrelevant details from user is called data Abstraction.

Three levels of Abstraction.



Three Levels of Abstractions:-

1) Physical level:- This is the lowest level of data Abstraction. It describes how data is actually stored in database.

2) Logical level:- This is the middle level of 3-level data abstraction architecture. It describes what data is stored in database.

3) View level:- Highest level of Abstraction. This level describes the

user interaction with database system.

Example:- We are storing customer information in a customer table.

At physical level:- These records can be described as block of storage (bytes, GB, TB etc) in memory.

At Logical level:- These records can be described as fiel and attributes along with

At View Level:- ~~Just user set~~ user just interact with system with the help of GUI and the entey details at the screen, they are not aware of how the data is stored and what data is stored; such details are hidden from them.

DATA INDEPENDENCE:-

Definations:- Data independence is defined as property of DBMS that helps you to change the Database schema at one level of a database system without requiring to change the schema at the next higher level.

OR

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

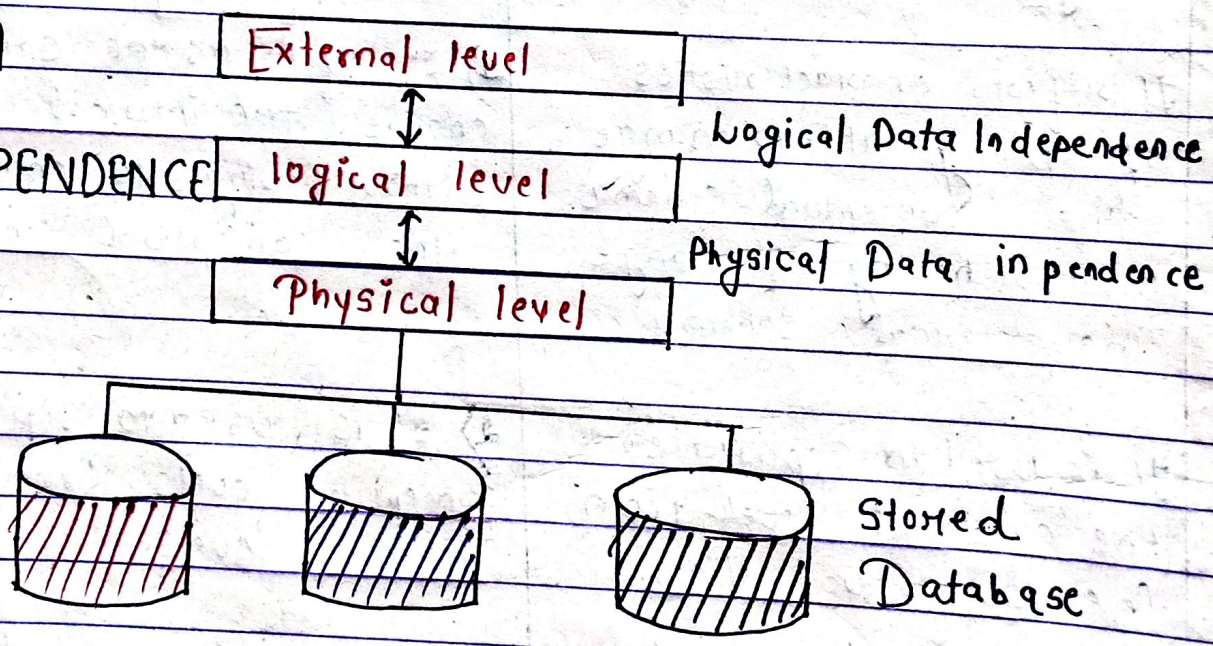
DATA INDEPEDENCE

Types of Data Independence.

Logical Data Independence	Physical Data Inpedence
1) It refers characteristics of being able to change the conceptual schema without having to change the external schema.	1) It refers as the capacity to change the internal schema without having to change the conceptual schema.
2) It is used to separate the external level from the conceptual view.	2) It is used to separate conceptual levels from the internal levels.
3) Logical Data Independence occurs at the user interface level.	3) Physical Data Independence occurs at the logical interface level.
4) if we do ~~not~~ any change the conceptual view of the data, then the user view	4) if we do not any changes in the storage size of the database system servers, then the

of the data would not be affected.	conceptual structure of the database will not be affected.

DATA INDEPENDENCE



- Metadata Data about data

DATA DEFINATION LANGUAGE (DDL)

1) DDL stands for Data Defination Language.
2) It is used to create schema, tables, indexes, constraints etc. in the database.
3) Data Defination language is used to store the information of metadata like the number of ~~data~~ tables and schemas, their names, indexes, column in each table, constraints etc.

- 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 in a table
- Rename:- It is used to rename an object.
- Comment:- It is used to comment on data dictionary.

DATA MANIPULATION LANGUAGE (DML)

1) DML stands for Data Manipulation Language.
2) It is used for accessing and manipulating data in a database. It handles user request.

- SELECT It is used to retrieve data from a database.
- INSERT It is used to insert data within a table.
- UPDATE It is used to update existing data within a table
- DELETE It is used to ~~records~~ delete all records in a table.
- MERGE It performs upsert operation. i.e insert or update.

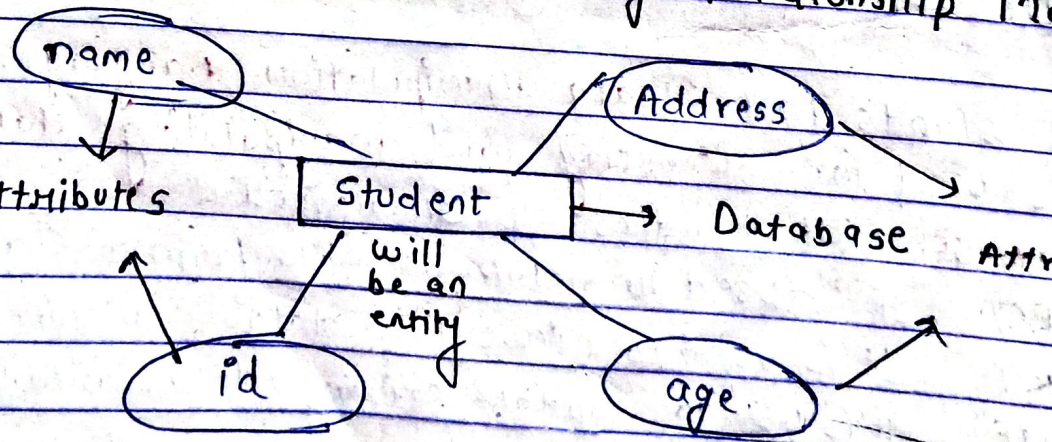
- CALL : It is used to call a structured query language or Java Subprogram.
- Explain plan : It has the parameter of explaining data.
- Lock Table : It controls Concurrency.

DATA MODELS

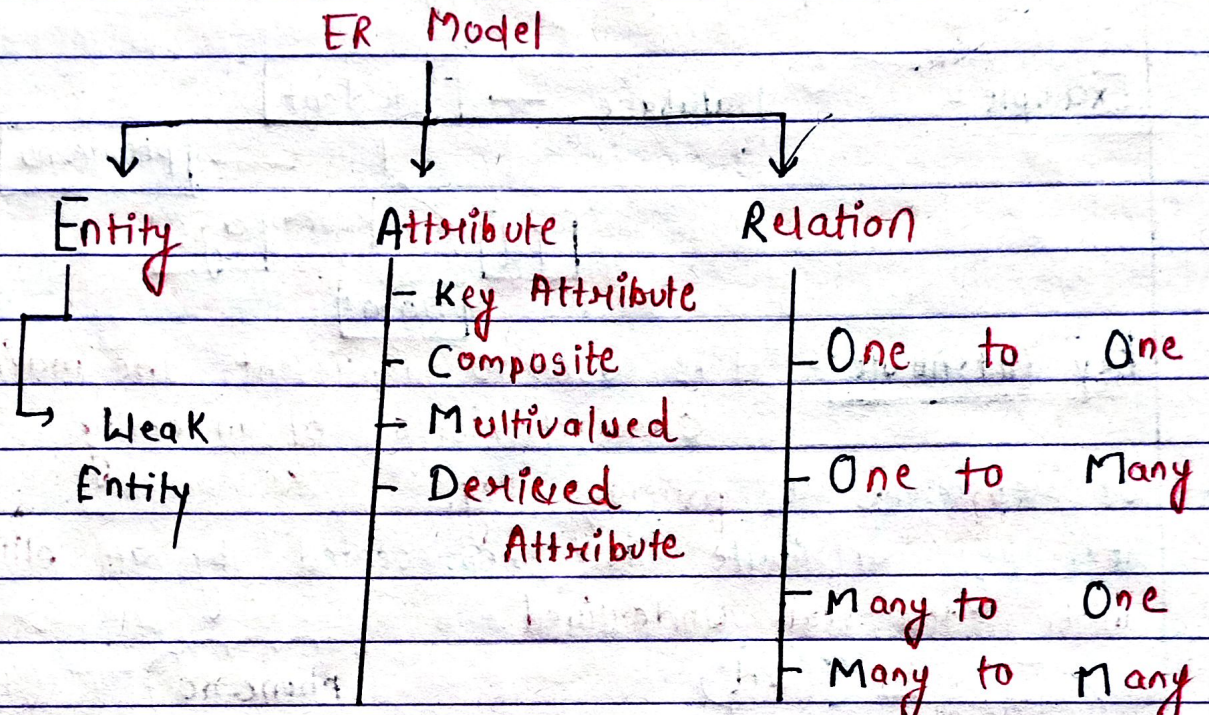
ER Models :-

→ ER models stands for Entity-Relationship Models.
→ It is high-level model.
→ This model defines the data elements and relationship for a specified system.
→ It develops a conceptual design for the database.
→ It also develops a very simple and easy to design view data.
→ In ER modelling, the database structure is portrayed as a diagram called an Entity-Relationship Model.

Example :-

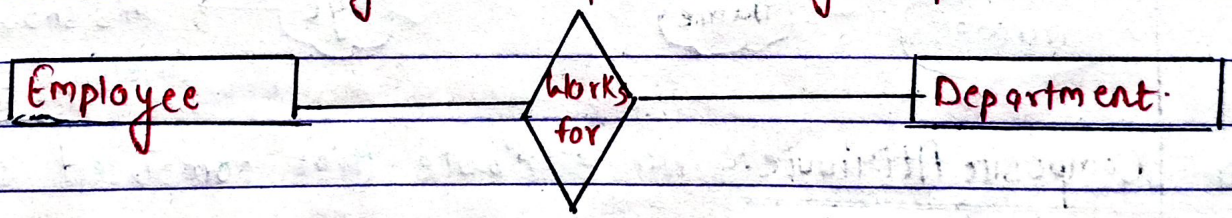


Component of ER Diagram.



1. **Entity**: An entity may be any object, class, person or place. In the ER model, an entity can be represented as Rectangle.

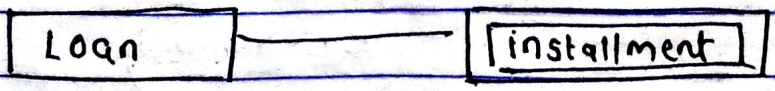
Example:- Consider an org. as example- manager, product etc.



(a) **Weak Entity**:- An entity that depends on other entity called a weak entity.

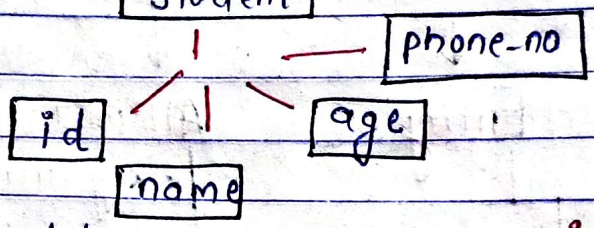
The weak entity that doesn't contain any key attribute of its own. It is represented by Double rectangle.

Ex:



2. **Attribute:-** The attribute is used to display describe the property of an entity.

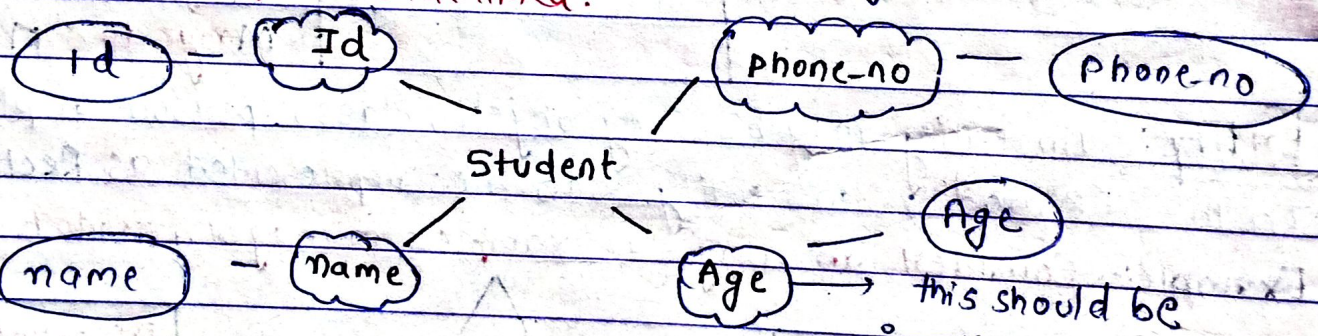
Example :- Database — Student



a. **Key Attribute:-** It is used to represent the main characteristics of an entity.

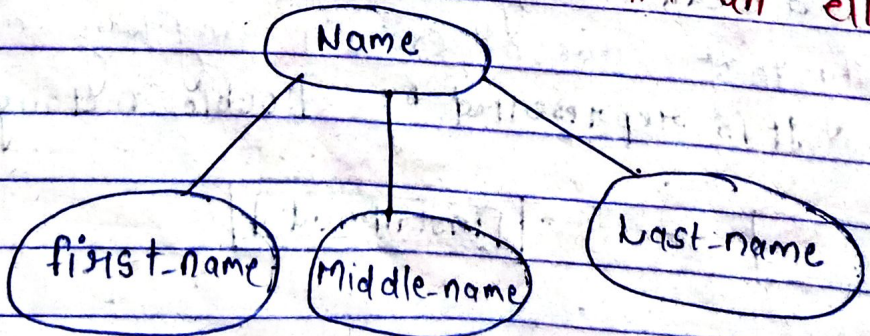
→ It represents a primary key.

→ The key attribute is represented by an ellipse with the text Underlined.



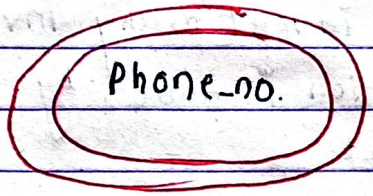
b. **Composite Attribute:-** An attribute that composed of many other attribute.

The composite Attribute is represented by an ellipse, and those ellipse are connected with an ellipse.



c. **Multivalued Attribute**:- An attribute can have more than one value. These attributes are known as a Multivalued Attribute. The double oval is used to represent multivalued attribute.

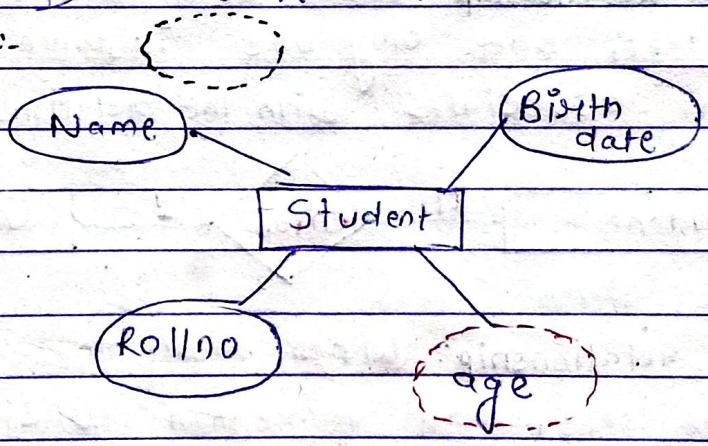
Example :- A student can have more than one phone no.



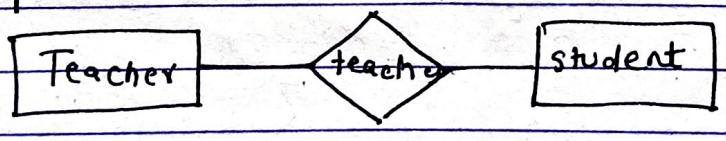
d. **Derived Attribute** :- An attribute that can be derived from other attribute is called Derived Attribute

Represented by :-

Example :-



3. **Relationship**:- It is used to describe the relation between entities.

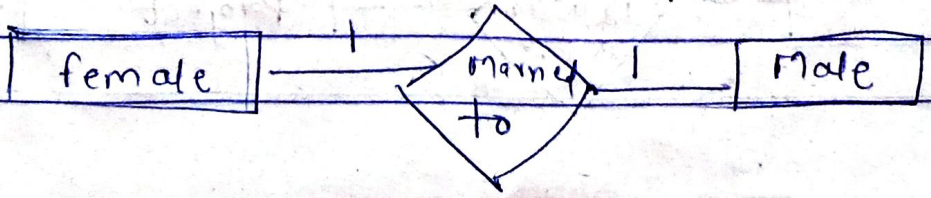


Types of Relationship :-

a) one-to-one when one instance of an entity is associated with the relationship.

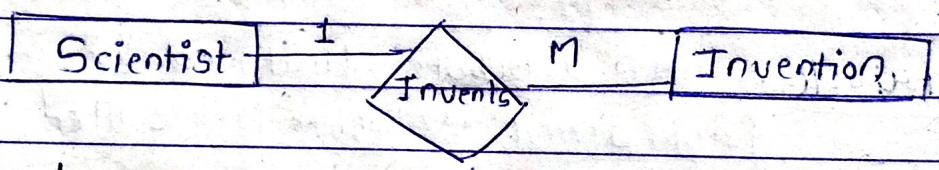
b)

Ex:-



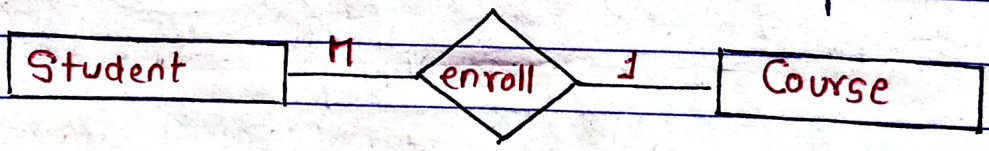
b. One-to-many Relationship:- When one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship

Example, Scientists can invent many inventions but the invention is done by the only specific scientist.



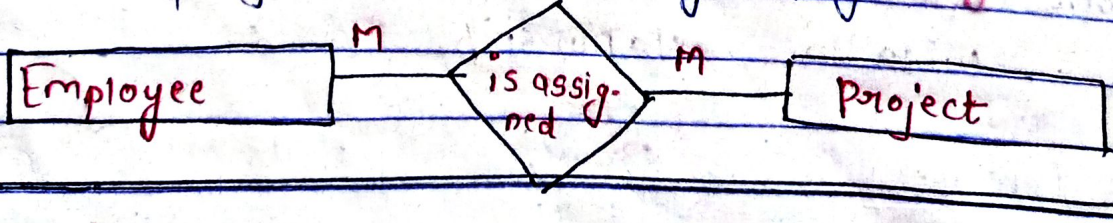
(c) Many to one Relationship:- When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship

Example:-



(d) Many to many relationship:- When more than one instance of the entity on the left, and more than one instances of an entity on the right associates with the relationship

Example- Employee can assign by many projects and Project can have many employees.



NETWORK MODEL :-

The network model is the extension of the hierachial structure because it allows many-to many relationship to be managed in a tree like structure that allow multiple parents.

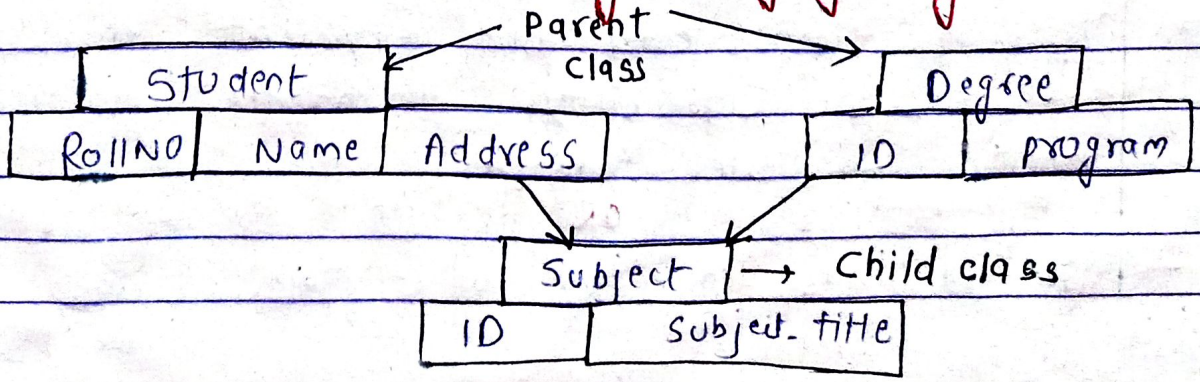
⇒ There are two fundamental concept of a network model :-

- Records contain fields which need hierarchial organization.
- sets are used to define one-to-many relationships between records that contain one owner, many members.

⇒ There are the following model major features :-

1) It can represent redundancy in data more efficient than that in the hierachial model.
2) There can be more than one path from a previous mode to successor node/s.
3) The operations of the network model are maintained by indexing structure of linked list (circular) where a program maintains a current position and navigate from one record to another by following the relationships in which the record participates
4) Records can also be located by supplying Key Values

Example :



Characteristics of network Model are:- (IN SHORT).

1. Network model is better than a hierarchial model.
2. Supports many to many relationship.
3. many children can have many parents.
4. many parents can have many children.
5. High performance
6. Entity are represented as a connected network with each other.
7. Not very flexible to reorganize the model.

Relational and object Oriented data

RELATIONAL DATA MODEL

→ The relational data model is the most famous data model and is used by majority around the world.

→ This is simple and but efficient data model and has the capability to handle data in the best possible manner

OR

→ The database collection of relation

⇒ Tables are used to handle the data in the relational data models:-

<Employee>

Emp-No.	Emp-Name	Emp-Desi.	Emp-Age	Emp-Salary
1	Jack	Manager	35	500
2	Tom	Technician	25	600
3	Henry	Secretary	50	700

Some popular Relational Database management system are.

- DB2 and Informix Dynamic Server - IBM
- Oracle and RDB - Oracle
- SQL server and Access - Microsoft

NOTE:→ A relation is nothing but a table of values.

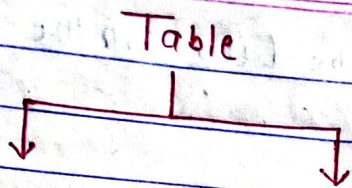
→ Every row in the table represents a collection of related data values.

→ These rows in the table denote a real-world entity or relationship.

RELATIONAL MODEL CONCEPTS.

1) Attribute:- Each column in a table. Attributes are the properties which define a relation.
Ex:- Student_Rollno, Name, etc.

2) Tables:- In the relational model the relation are stored stored along with its entity.



3. **Tuple**:- It is nothing but a single row of a table, which contains a single record.

4. **Relation Schema**: Represents the name of the relation with its attributes.

5. **Degree**: The total number of attributes which in the relation

6. **Cardinality**:- Total no. of rows present in a table.

7. **Relation Instance**:- finite set of tuples in RDBMS Relation instances never have duplicate tuples.

8. **Column**:- The set of values for a specific attribute.

9. **Relation key**:- Every row has one, two or multiple attributes which is called relation key.

10. **Attribute domain**:- Every attribute has some pre-defined values and scope.

RELATIONAL INTEGRITY CONSTRAINTS

(Conditions must be present for a valid relation)

Domain Constraints

↳ It specify that within each tuple, and the value of each attribute must be unique.

↳ This is specified as data types which include standard data types, real no, characters, Booleans, variable length strings etc.

Example

Create DOMAIN CustomerName CHECK (Value not NULL)

↓

Here CustomerName is not nut (Creating a domain constraints)

Key Constraints

An attribute that can uniquely identify a tuple in a relation. The value of the attribute for different tuples in the relation has to be unique.

Example:-

Customer ID	Customer Name	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

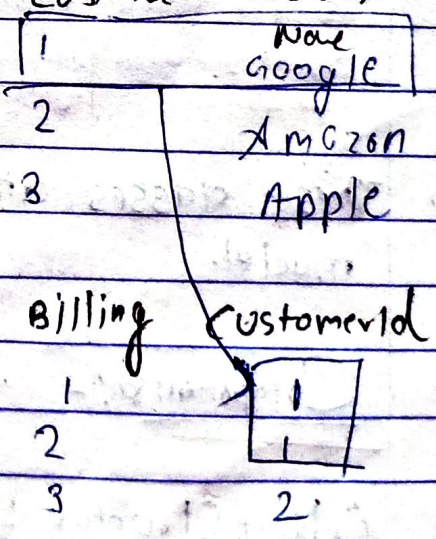
Here, customerId is Key

Referential Integrity

Referential Integrity constraint in DBMS are based on the concept of foreign keys.

Foreign Key is an important attribute of a relation which should be referred to in other relationship.

Ex:



⇒ Operations in Relational model.

- INSERT
- DELETE
- MODIFY
- SELECT

⇒ Advantage

→ Simplicity (compare Hierarchical)
→ Structural independence
→ Easy to use
→ Query capability
→ Data independence
→ Scalable

Disadvantage:

→ few relational databases have limits on field lengths which can't be exceeded.
→ Relational database can sometimes becomes complex as the amount of data grows.

ODBMS

ODBMS stands for Object oriented database system is the data model in which data is stored in form of objects, which are instances of classes.

→ These classes and objects together make an object oriented model.

Components :-

1) Object Structure :- The structure of an object refers to the properties that an object is made up of.

→ These properties of an object are referred as an attribute.

→ Thus,
an object is a real world entity with certain attribute that makes up the object structure

Object Structure

Messages
provides an interface or acts as a communication medium between an object and the outside world.

- **Read-only message**: If the invoked method does not change the value of a variable
- **Update message**: changes the value of a variable

Methods
When a message is passed then the body of code that is executed.
→ Every time when a method is executed, it returns a value as output.

- **Read only method**: When the value of a var. is not affected by a method
- **Update method**: changes by a method

Variables
→ It stores the data of an object.
The data stored in the variables make the object distinguishable from one other.

Object Oriented Data Model :-

→ Object oriented data model is based upon real world situations.

→ These situations are represented as objects, with different attributes.

Elements of Object Oriented Data model

↳ **Objects:-** The real worlds entities and situations are represented as objects.

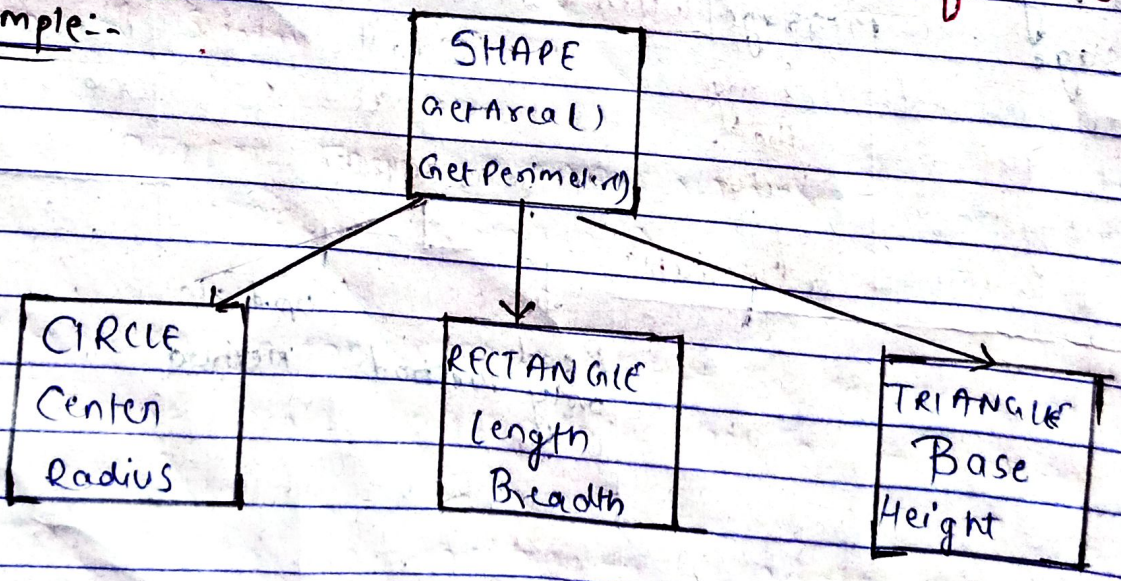
↳ **Attributes :-** Every objects has certain characteristics. are represented using Att.

Methods:- The behaviour of the objects.

↳ **class :-** Similar attributes and methods are grouped together using a class. An object can be called as instance of the class.

↳ **Inheritance:-** A new class can be derived from the class.

Example:-



Integrity Constraints:-

1. Integrity Constraints are a set of rules. It is used to maintain the quality of information.

2. Integrity constraints ensure that the data insertion, updating and other processes have to be performed in such a way that data integrity is not affected.

3. Thus, integrity constraints is used to guard against accidental damage to the database.

Types of Integrity Constraints

- Domain Constraint
- Entity Integrity Constraint
- Referential Integrity Constraint
- Key Constraint.

Data Manipulation Constraints:- Operations:

Data Manipulation Commands are used to manipulate data in the database.

Some of the data Manipulation Commands are-

1. **SELECT**: Select statement retrieves the data from database

```
SELECT <COLUMNNAME>
FROM <TABLE NAME>
WHERE <CONDITIONS> ;
```

(2) **INSERT** :- Insert statement is used to insert data into database tables.

General Syntax :-

```
INSERT INTO <TABLENAME>
(<COLUMN TO INSERT>).
```

(3) **UPDATE** :- The update command updates existing data within a table.

General Syntax :-

```
UPDATE <TABLE NAME>
SET <COLUMN NAME>= <UPDATE COLUMN NAME>
WHERE <CONDITION>;
```

(4) **DELETE** :- Delete records from the database table according to the given constraints.

General Syntax :-

```
DELETE FROM <TABLE NAME>
WHERE <CONDITION>;
```

(5) **MERGE** :- Use the MERGE statement to select rows from one table for update or insertion into another table.

The decision whether to update or insert into the target table based on a condition in the ON clause.

It is also known as UPSERT. i.e. Combination of Update and insert.

General Syntax:-

```
MERGE <TARGET TABLE> [AS TARGET]
USING <SOURCE TABLE> [AS SOURCE]
ON <SEARCH CONDITION>
[WHEN MATCHED
   THEN <MERGE MATHCHED>]
[WHEN NOT MATCHED [BY TARGET]
THEN <MERGE NOT MATCHED>];
```