# Database System

Collection of related data

Database

RDBMS

Structured
IRCTC, University

Unstructured
Webpages

(operations)

Database management system

SQL Server
Oracle 9i, 11, 12c
My SQL
$DB_2$

# Structured database: The database for which we have a proper structured named as Relational Database Management system (RDBMS).

→ Data store in the form of Relation Or Table.

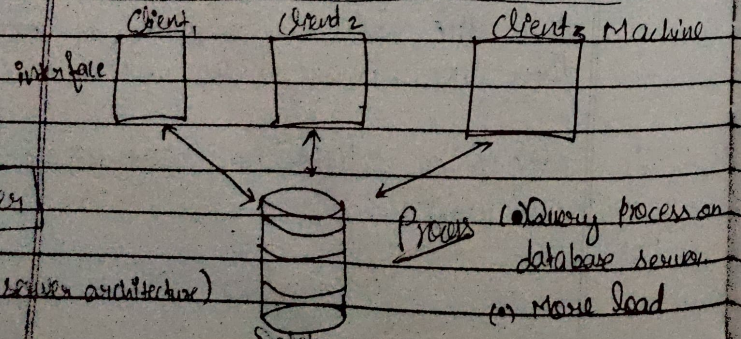
# Unstructured database: In this we do not have a proper structure to store unstructured data because in this the data is majority of no format data like, image, video, etc.

∴ 90% of data is unstructure in the world.

## File system Vs DBMS :-

| | DBMS |
|---|---|
| 1. | Searching Fast, easy. |
| 2. | Memory utilisation is more efficient. |
| 3. | Totally Independent |
| 4. | DBMS is more concurrent in nature. |
| 5. | Provide proper protocols. → RR, RW, WR, WW. |
| 6. | DBMS is more secure. (Role base security) → In which we can make the different users and their different roles to them to access provide the data. |
| 7. | Data Redundancy Or Data duplicating. → Their are lot of constrains like primary key, unique identity must be store. |

## 2-tier and 3-tier Architecture :-




2-Tier
(client server architecture)

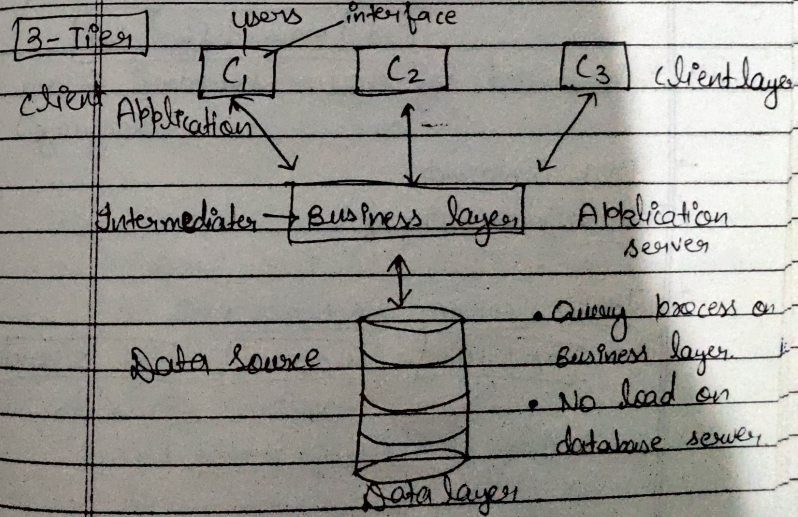
(+) Query process on database server.
(-) More load

→ ADVANTAGE OF 2-Tier :-

(+) 2 layer server
(+) Easy maintainence
(+) Limited client / limited database which we want to access.

→ Draw back :-

(-) It will fails if there are too many clients are present and they want to access data at any time from any location.
(-) Scalability.
(-) It doesn't secure as compare to 3-tier.



→ ADVANTAGE OF 3-Tier:
(•) Scalability
(•) Security

→ DISADVANTAGE:
(•) High maintainance
(•) Complex system

◆ SCHEMA:
Logical representation of a database.

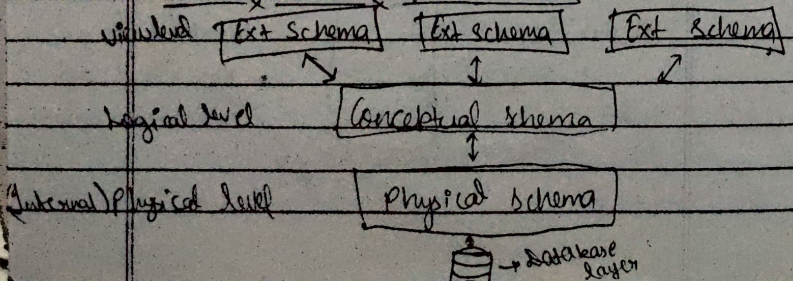
Student
| RollNo | name | addresses |

Course
| Cid | name | Duration |

→ Implemented by SQL. (Structured Query Language)

→ DDL commands
(Data definition language)

◆ THREE SCHEMA ARCHITECTURE:




◆ EXAMPLE OF THREE LEVELS:

External
Customer_Loan:
Cust_ID : 101
Loan_No : 0111
Amount_in_Dollars : 8255.00

Conceptual
CREATE TABLE Customer_Loan(
Cust_ID Number(4),
Loan_NO Number(4)
Amount_in_Dollars Number(7,2),

Internal
Cust_ID TYPE = BYTE(4), OFFSET = 0
Loan_NO TYPE = BYTE(4), OFFSET = 4
Amount_in_Dollars TYPE = BYTE(7); OFFSET = 8

◆ Instances:
• Instances are the collection of information stored at a particular moment.
• These instances can be changed by certain CRUD (Create, Read, update, Delete) operation as like addition, deletion of data.

◆ Data Independence:
• It can be explained by three-Schema Architecture.
• It is being able to modify the schema at one level of database system without altering the schema at higher level.

- There are two types:-
1. Logical data Independence
2. Physical data Independence

1- Logical data Independence:

(•) It can be able to change the conceptual schema without having to change the external schema.

(•) It is used to separate the conceptual view from the external level.

(•) If we do changes in conceptual data then there is no affect on the user view data.

(•) It occurs at the user Interface level.

2. Physical Data Independence:

(•) It can be able to change the internal schema without having to change the conceptual schema.

(•) If we do any changes in the storage size of database system server, then the conceptual structure of database will not be affected.

(•) It is used to separate conceptual levels from the internal level.

(•) It occurs at the Logical Interface level.

Database Models:

(•) Data models gives us an idea that how the final system will look like after its complete implementation.

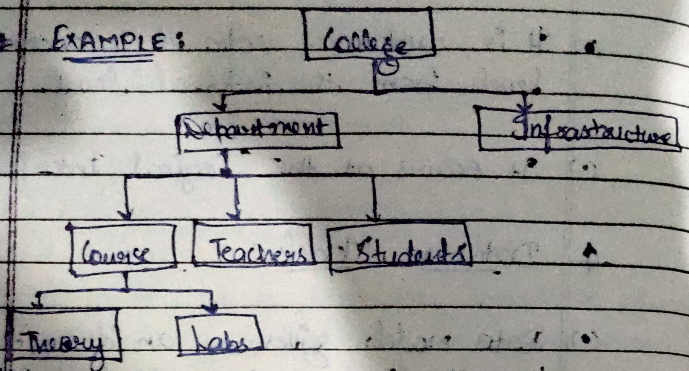
(•) It defines the data elements and their relationships.

(•) It shows that how data is stored, connected, accessed & updated in DBMS.

→ Some models are:

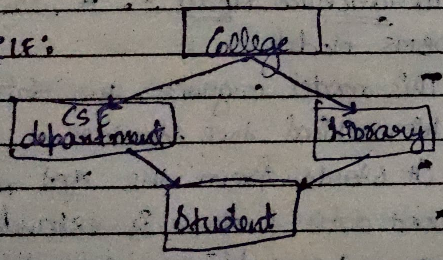
1. Hierarchical model:- It was the first DBMS model.
- This model organizes the data in the hierarchical tree structure.
- It starts from the root which has root data & then it expands in the form of a tree adding child node to parent node.

# EXAMPLE:




2. Network Model: It is an extension of hierarchical model.
- It was the more popular model before the relational model.
- It is as same as hierarchical model, the only difference is that a record can have more than one parent.
- It replaces the hierarchical tree with a graph.

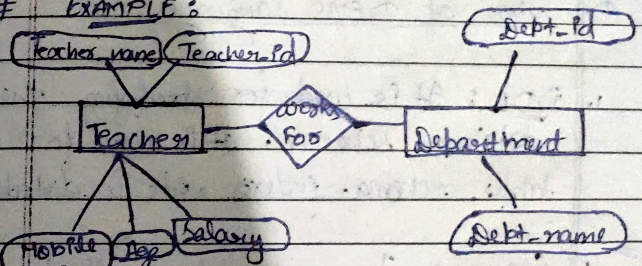
# EXAMPLE:




Network Model

3. Entity - Relationship Model:
- ER model is a high-level data model diagram.
- In this, we represent the real world problem in the pictorial form to make it easy for the stakeholders to understand.
- It is also easy for the developers to understand the system by just looking at the ER diagram.
- We use ER diagram as a visual tool to represent an ER Model.
- It has three components: Entity, Attribute, Relationship.

# EXAMPLE:




4. Relational Model: It is introduced by Dr. EF Codd.
- It is the most widely used model.
- In this, the data is maintained in the form of 2D table.
- All information stored in the form of rows & columns.

- Basic structure of relational model is tables.
- So, tables are also called relations in the relational model.

Column ↓

# EXAMPLE:

Attribute →

| SID | SName | SAge | SClass | Section |
|---|---|---|---|---|
| 1101 | Alex | 26 | 12 | A |
| 1102 | Mahima | 14 | 10 | B |
| 1103 | Rohit | 14 | 9 | A |
| 1104 | Newton | 15 | 10 | B |

tuple →

table (relation)

# Types of DBMS languages:

1. DDL: It is used for specifying the database schema: It is used for creating tables, schema, indexes, etc in database.

CREATE → To create database instance
ALTER → To alter structure of database instance
TRUNCATE → To delete tables in a database instance.
RENAME → To rename database instance.
DROP → To drop objects from database such as tables.

2. DML: It is used for accessing & manipulating data in Database.

SELECT → To read records from table(s).
INSERT → To insert record(s) into the table(s).
UPDATE → update the record in table(s).
DELETE → Delete all the records from table.

# TWO TYPES OF DML:

1. High-level or Non-procedural languages: which require a user what data is needed without specifing how to get it.

2. Low-level or Procedural languages: which requires a user what data is needed and how to get it.

3. DCL: It is used for granting & revoking user access on a database.

GRANT: To grant access the user.
REVOKE: To revoke access from user.

## Keys:

A key in DBMS is an attribute or a set of attributes that help to uniquely identify a tuple (or row) in a relation (or table).

1. Primary key:
- A primary key is an attribute which is uniquely identified in the entire table.
- It can be only one in a table.
- Every value of the primary key has to be different with no repetitions.

2. Candidate key:
- Those attributes which are uniquely identify rows of a table.
- The PK of a table is also selected from one of the candidate key.
- There can be more than one candidate key in a table.

3. Alternate key:
All the keys which do not become the primary key are called alternate key.

4. Super key:
- Super key is a set of all the keys which help to identify rows in a table uniquely.
- It means that all those columns that are capable of identifying the other columns of that table uniquely will all be considered super keys.
- Super key is the superset of Candidate key.

5. Composite key:
- It is a set of two or more attributes that help identify each table uniquely.

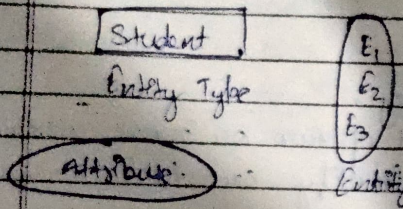
6. Unique key:
- It is a column or set of columns that uniquely identify each record in a table.
- All values will have to be unique in this key.

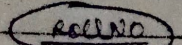
7. Foreign key: It is used to establish relationship between two tables.

## E-R Diagram:

- It stands for Entity- Relationship diagram.
- It shows the relationship among entity sets.
- It shows the complete logical structure of a database.
- An entity set is a group of similar entities & these entities can have attributes.

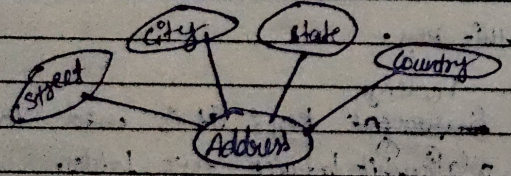
→ Entity, Entity Type, Entity set:



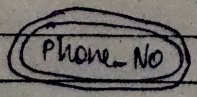

1. 

key attribute

2. Composite attribute



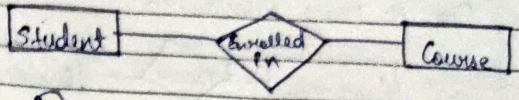

3. Multivalued attribute

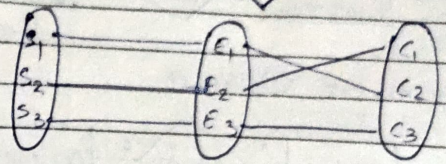



4. Derived Attribute




→ Relationship Type and Relationship set:

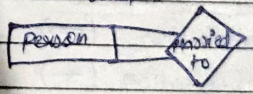


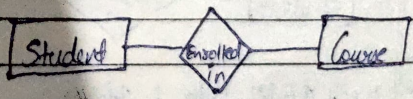

→ Degree of Relationship set:

The no. of different entity set participating in a relationship set is called as degree of relationship set.

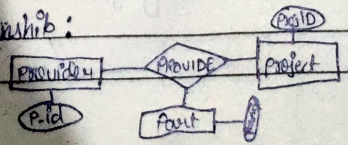
1. Unary Relationship:




2. Binary Relationship:

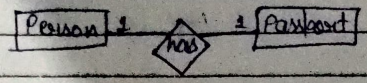



3. n-ary Relationship:

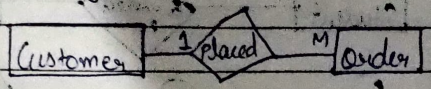


## CARDINALITY:

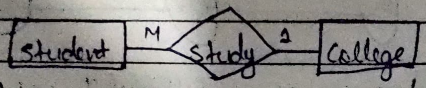
1. One to one Relationship



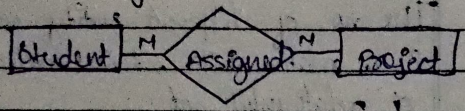

2. One to Many Relationship




3. Many to one Relationship

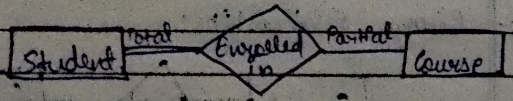



4. Many to many Relationship.

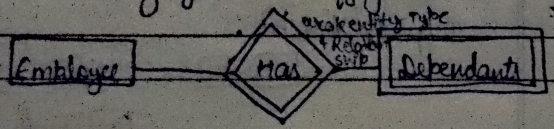



## PARTICIPATION CONSTRAINTS:

- Total participation (═══)
- Partial Participation (———)




## WEAK ENTITY TYPE & IDENTIFYING RELATIONSHIP:

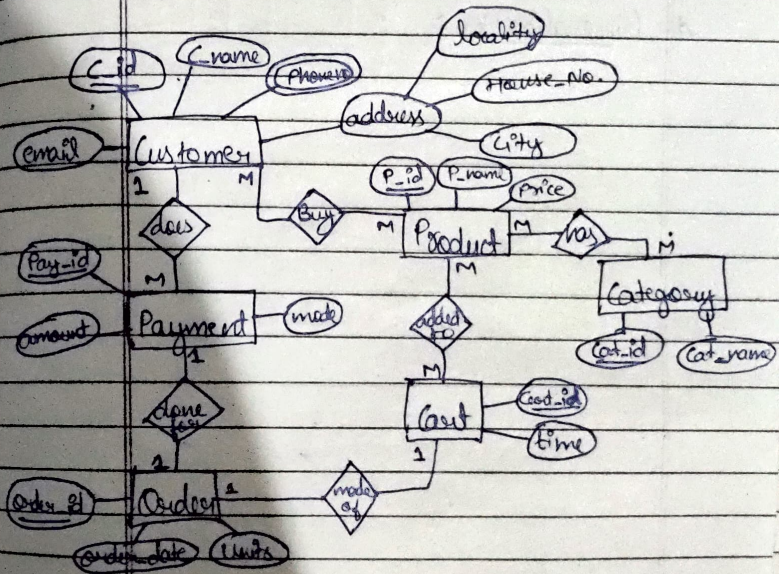



## EXAMPLE:

E-R Diagram for online shopping system.

### Entities:

1. Customer
2. Product
3. Category
4. Payment
5. order
6. Cart



## EER Diagram: (Extended Entity Relationship)

- High level data model that represent the requirements and complexities of complex database.
- In addition to ER model concepts EER includes specialization, Generalization and Aggregation.

## Generalization: