Khushboo
Shkya

DBMS

1

Introduction

- Data: is generally a raw format of any information.
- Information: is nothing that data being viewed in a structured format
- Database: is a collection of related data, organized in a structured and meaningful way.
- Data base management System: is software that ~~can~~ enables user to manage a database easily
  Ex. SQL, mySQL, Oracle etc
- Issue with File system (Why we shifted to DBMS)
  - ↳ Data Redundancy
  - ↳ Data Inconsistency
  - ↳ Data security
  - ↳ No backup and Recover
  - ↳ Data mapping and access
- Tuple = row = record = Each tuples represents a complete record of the specific data items, each tuple stores different data with same structure.

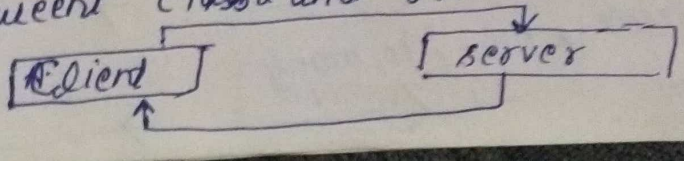
2

DBMS Architecture

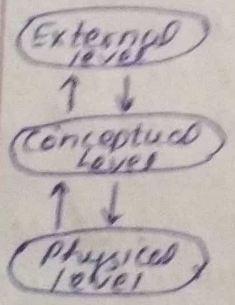
1 tier Architecture
2 tier Architecture
3 tier Architecture

- 1 tier :- Client server and database all present on the same machine.

2 tier :- Independent communication betweent client and server.



3 tier :-



Logical Data Independence

Physical Data Independence

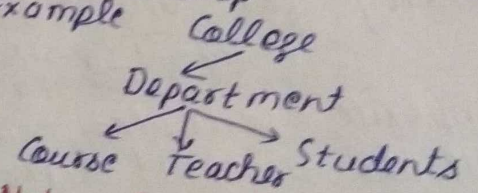
Data Abstraction and Data independence

- Data Abstraction refers to the hiding of details from user at certain levels for authentication and security purpose.
  - Conceptual Level
  - Internal Level
  - External Level
- Data Independence is all the transaction or changes made at one level are unaffected to other levels
  - Logical data Independence
  - Physical data independence

Data models in DBMS

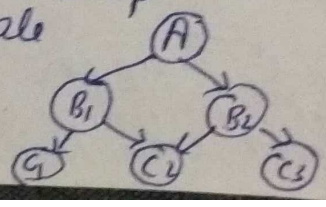
i) Hierarchical model:-
- ↳ data is organized like a tree structure
- ↳ based on one to many relationship.

Example



ii) Network model:-
- ↳ is just like a graph rather than a tree, a child record can have any number of parent ~~the~~ record.
- ↳ based on many to many relationship

Example



iii) Entity Relationship model (ER model) is a design or blueprint of a database that can later implemented as a database

iv) Relational Data Model :- This relational data model is simple 2-D tables of value

v) Object Oriented Database Model :- represents as object, with different attributes. All these object have multiple relationship between them

3

ER MODELS

Units components

- Entity :- real object representation in a ER diagram.

  Student → rectangle

- ~~En~~ Entity Set :- set of all entities together

  Student
  Courses
  Teachers
  → vertical Oval

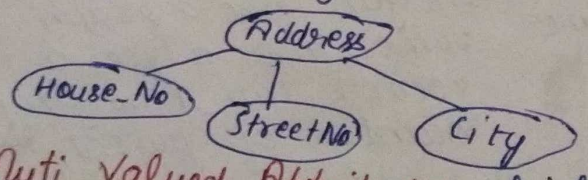
- Attributes :- are the properties of any given entity.

  ID → Horizontal Oval

- Key Attributes :- is one which will uniquely identify and associate identity for an entity.

  Student_ID → Horizontal oval with under line

- Composite Attribute :- composes of multiple units of attributes forming a larger one.



- Muti Valued Attribute :- which may have more than one value

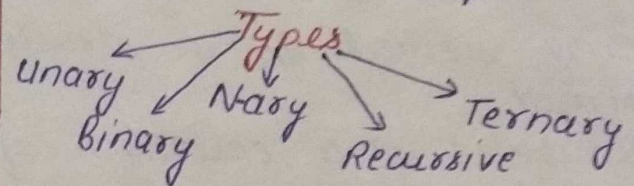
  Phone_No → Double Horizontal Oval

- Derived Attribute :- One that can derived from other attribute (Age)

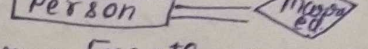
Relationships

↳ represents the association between two or more entities

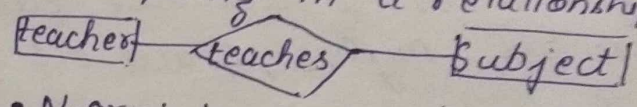
↳ shows the different entity sets that are participating in a relationship



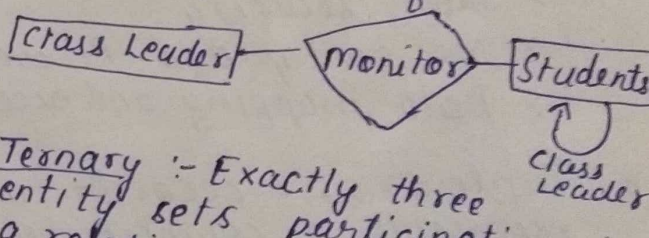
- Unary :- Only one entity set participating in a relationship



- Binary :- Exactly two entity sets participating in a relationship.



- N-ary :- large No of entity sets are participating in a relationship.
- Recursive :- Entity is having a relationship with self



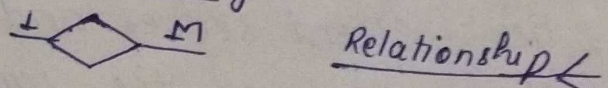
- Ternary :- Exactly three entity sets participating in a relationship.

Cardinality

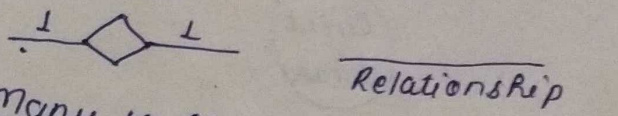
The number of entity that can participate in a relationship with another entity set.

Types

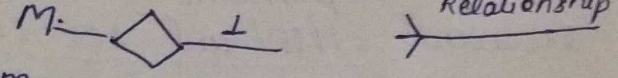
- One to many :-



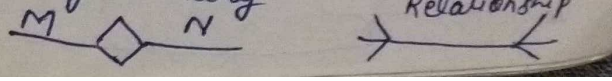
- One to one



- Many to one



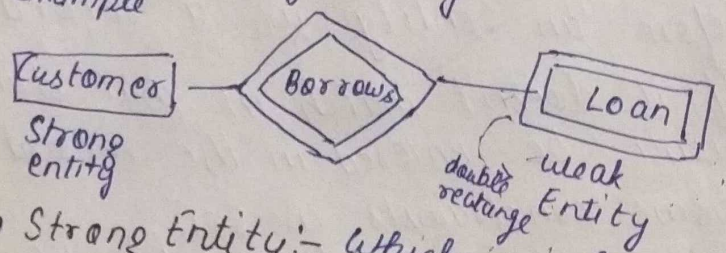
- Many to many



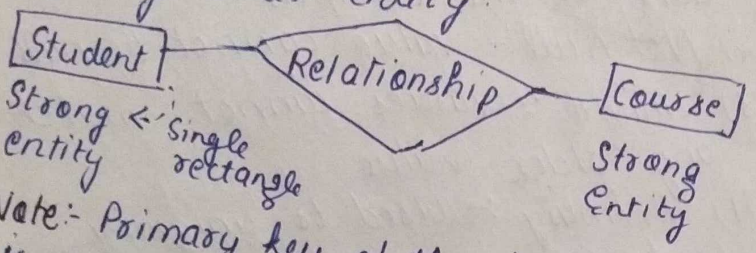
Weak Entity and Strong Entity

- Weak Entity :- which entity is depend-ent on strong entity.

Example



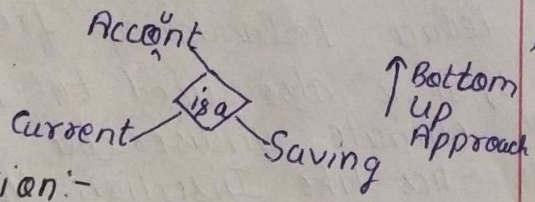
- Strong Entity :- Which is independent of any other entity.



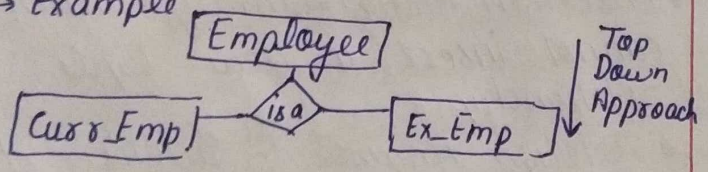
Note :- Primary key of the strong entity is represented by underlining it (Called strong attribute)

Generalization, Specialization, Aggregation

- Generalization :-
  - ↳ Combine lower level entity to higher level entity.
  - ↳ Bottom up approch.
  - ↳ Example

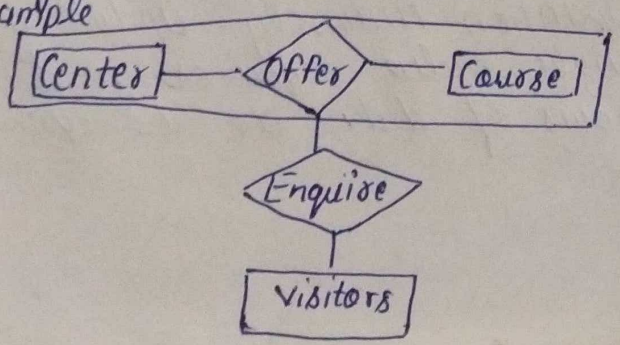


- Specialization :-
  - ↳ Opposite of Generalization
  - ↳ Example



- Aggregation :- the relation between two entities is treated as a single entity.

Example



4 RELATIONAL DATABASE MODEL

Codd's Rule

0) Foundation Rule
1) Information Rule
2) Guranteed Access
3) Systematic treatment of null values
4) Active online Catlouge
5) Powerful and well structured Language
6) View updation Rule
7) Relational Level Operation
8) Physical Data Independence
9) Logical Data Independence
10) Integrity Independence
11) Distribution Independence
12) Non-Subdivision Rule

RDMS

The software used to store, manage, query and retrieve data stored in a relational database
Example SQL, MYSQL, Oracle

ID	Name	Age	GPA
1	Arya	21	4
2	Bran	19	3
3	John	24	4.3

← field

← Tuple/Row/Record

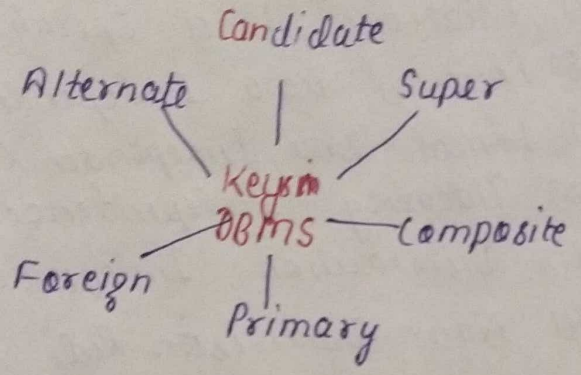
↑ Column/Attribute

Table

Terms used in the Relational Model

- Relation :- Representing data in the form of a table
- Schema :- Organization of data in a table.
- Instance :- unique values present in the table at present.

- Attribute:- An attribute defines the properties of a table. that means what type of data that a table is storing.
- Domain:- Set of a possible values that are allowed for a column in a database table.
- Null values:- Data may be unknown, missing or undefined which are represented by using this NULL



- Primary Key- To uniquely identify each tuple or row in a data table.
- Candidate Key- The minimum number of attributes that can uniquely identify a record for a data table
- Super Key:- uniquely identify a record or tuple in the database.
  - Candidate Key - Minimal
  - ↳ Super key - Doesn't say any no.
- Composite Key:- That can uniquely identify a record in a table.
- Foreign Key:- A foreign key is a column of a table that points towards the primary of another table

SQL Constraints

Constraints in DBMS are various checks invoked by the system before an entity in a record. If it doesn't match in that data won't be entered in the record

Some constraints in SQL

1) Default:- is used to add default data to the columns.
2) Not Null:- value cannot be null
3) Unique:- values cannot match any older value
4) Primary:- Used to uniquely identify a row.
   Not Null + Unique = Primary
5) Foreign:- References a row in another table
6) Check:- Checks for the predefined condition before inserting the data

5

NORMALIZATION

↳ It is a technique to Remove or Reduce Redundancy from a table

↳ It is also used to ~~minimize~~ elimate undiserable chactrecteristics like Insertion, Update and Delete Anomalies.

- Insertion ~~Anato~~ Anamaly:- when one cannot insert a new Tuple into relationship
- Delelion Anamoly:- Deletion of data results in the unintended loss of some other important data.
- Updation Anamoly:- update a single value requires multiple rows of data to be updated.

Normal Forms

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

- **1 Normal Form:** Should not have any multivalued attribute.
  ↳ (contains more than one value in a single cell)

Example

Roll No	Course
1	C, C++
2	C++

Not 1 NF

- **2 Normal:**
  ↳ Must be in 1 NF.
  ↳ No Partial Dependencies
  (Non Prime attribute is functionally dependent on part of a candidate key
  Eg. AB → B → C)

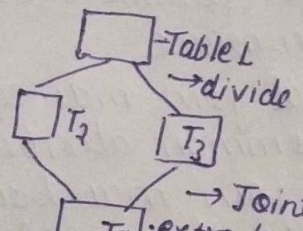
- **3 Normal Form:**
  ↳ Must be in 2NF
  ↳ No transitive Dependency
  (when non prime attribute depends non prime attribute.
  Roll No (Prime) → State (Non Prime) → City (Non Prime))

- **Boyee Codd Normal Form:**
  ↳ Must be in 3NF.
  ↳ L.H.S. of each functional Dependency should be candidate Key or Super Key.

  $X \rightarrow Y$
  should (CK, SK)

- **4 NF:-**
  ↳ Must be in BCNF
  ↳ No multivalued Dependencies
  (A Dependency A → B for a single value of A there exist multiple value of B)

- **5 NF:-**
  ↳ Must be in 4NF
  ↳ Lossless decomposition



· extra tuple will come after joining so for that we will make common attribute as candidate key.

Joins in DBMS

↳ Used to retrieve data from multiple tables i.e. it used for data merging.
↳ Joins = Cross product + Select Statement (Condition)

Joins
- Cross Joins
- Natural Joins
- Conditional Join
- Self Join
- Equi Join
- Outer Join
  - Right
  - Left
  - full

- **Natural Joins:-** It joins the table only when the two tables have at least one common attribute with same name and domain.
- **Self Joins:-** which a table is joined with itself.

Sytax: Select columnname(s) From table1 T1, table1 T2 where condition

- **Equi Join:-** performs a Join against equality or matching column values of the associated tables

Syntax: Select column_name from table1, table2 where table1.column_name = table2.column_name;

Note:- Natural join mai equal karta hai common attribute ko but Equi join mai koi bhi column ko equal karta hai

- Inner Join:- selects record that have matching values in both tables.

Syntax: Select column_name from table1 INNER JOIN table2 ON table1.column_name = table2.column_name;

- Left outer Joins- return all rows from the left table ~~table~~ and matching rows from right side table.

Syntax: Select column_name From table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;

- Right outer Join:- return all rows from the right table and matchig rows from left side table.

Syntax:- Select column_name from table1 RIGHT JOIN table 2 On table.column_name = table.column_name;

- Full outer join: Returns all the matched ~~or~~ records and unmatched records from both left side table and right side table.

Syntax: Select column_name from table1 FULL JOIN table2 ON table1.column_name = table2.column_name; ~~where~~

ACID PROPERTIES

To maintain consistency of the database before and after a transaction, specific properties are followed called acid properties.

Atomicity | Consistency | Isolation | Durability

- Atomicity:- transaction happens only if it can be completed and achieve its purpose or if it doesn't happen at all
  - all or nothing
- Consistency:- Before the transaction start and after the transaction completed, Sum of money should be same.
- Isolation:- ensures that multiple transaction can occur at the same time ~~per~~ provided each transaction is independent and shall not interfere in anothe transcd
- Durability:- Durability ensures the permanenty of something i.e. that the data after the successful execution of the operation becomes permanent in the database.