

23/01/23

M	T	W	T	F	S	S
Page No.						
Date						

YOUNA

DATA

It is meaningful Known raw facts that can be processed and stored as information.

DATABASE

It is a collection of interrelated and organised data. Ex -> Student, book, library.

DATABASE MANAGEMENT SYSTEM -

It is a collection of interrelated data and set of program to access, update and manage those data.

- MS Access
- dBase
- File Marker Pro
- Fox BASE
- Oracle
- Ingress
- Informix
- MySQL
- MS SQL Server
- IBM DB2
- PostgreSQL

Primary Goals of Database Management System

- To provide a way to store and retrieve database information that is both convenient and efficient.
- To manage large and small bodies of information.

It involves defining structure for storage of information & providing mechanism for manipulation of information.

- It should ensure safety of information stored despite system crashes or attempts at unauthorized access.
- If data ~~box~~ are to be shared among several users then the system should avoid possible anomalies.

Disadvantages of File Processing System.

- Data Redundancy :
Same information may be duplicated in several files and this duplication of data over several files is known as data redundancy.
Ex → The address & telephone number of a particular customer may appear in a file that consists of saving account records and in a file that consists of current account records.
- Data Inconsistency :
Various copies of same data may no longer agree i.e. various copies of the same data may contain different information.

- Difficulty in Accessing data .

In conventional file processing system it is difficult to access data in specific manner and it is required to create an application program to carry out each new task.

- Data Isolation .

Because data are scattered in various files and files may be in different format writing new application programs to retrieve appropriate data is difficult.

- Integrity Constraints .

The data values stored in the database must satisfy certain types of consistency constraints .
Ex. - Balance of bank account may never fall below a prescribed amount.

- Atomicity Problem .

In many application it is crucial that if any failure occurs then the data may be restored to a consistent state that existed prior to the failure state.

It is difficult to ensure atomicity in a conventional data state .

INSTANCES

Instances are collection of information stored at a particular moment. The instances can be changed by certain operations like addition, deletion of data.

- Let's say a table teacher in our database whose name is school. Suppose the table has 50 records so the instance of the database has 50 records for now and for tomorrow we are going to add another 50 records. So tomorrow the instance has total 50 records. This is called Instances.

SCHEMA

This is the overall display of database, the basic structure of how the data will be stored in the database is called Schema.

* physical Schema →

It describes the database designed at physical level. It is hidden below the Logical Schema and can be changed easily without affecting the application program.

* Logical Schema →

It describes the database designed at logical level. Programmers construct applications using Logical Schema.

* Views Schema →

It designs the view of the database at the view level. It is the highest level of Schema which defines the views for end users.

Schema) Example → Teacher table .

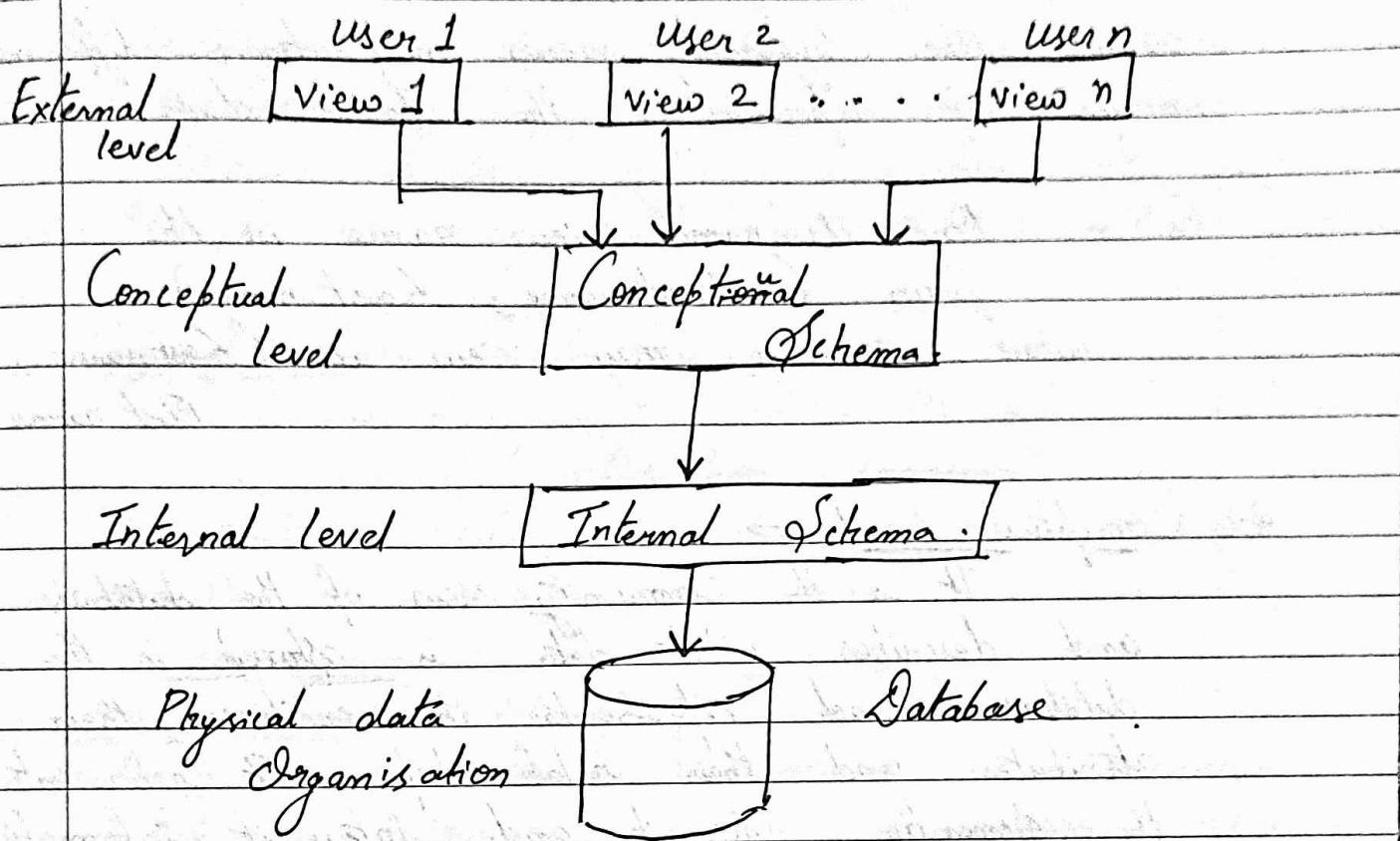
name : String .
doj : date .
dob : date .

Schema : Instances .

- 1.) It is the overall description → It is the collection of information stored in a database at a particular moment .
- 2.) Schema is same for the whole database . 2.) Data in instances can be changed using addition , deletion and updation .
- 3.) Does not change frequently . 3.) Changes frequently .
- 4.) Defines the basic structure of the database i.e. how the data will be stored in the database . 4.) It is the set of information stored at a particular time .
- 5.)

IMP.

ANSI / SPARK 3 level Architecture of DBMS



In 1971 DBTG (Database Task Group) realized the requirement for 2 level approach having views and schema and afterward in 1975 NC Spark realized the need for a 3 level - approach with the three levels of abstraction comprising of an external, conceptual and internal level. The aim is to separate each user's view of the database from the way the database is physically represented.

1) External Level →

It is the view, how the user views the database. The data of the database i.e relevant to that user is described at this level

M&H

2) In the external view only that entities, attributes and relationships are included that the user wants. The different views may have different ways of representing the same data.

Ex. → One User may view name in the form (First name, Last name) while another may view as (Lastname, First name)

2) Conceptual Level →

It is the community view of the database and describes what data is stored in the database and represents the entities, their attributes and their relationships. It represents the Semantic, Security and integrity information about the data. This level contains the logical structure of the entire database.

3) Internal Level →

At the internal level, the database is represented physically on the computer. It emphasizes the physical implementation of the database to do storage based utilization and to achieve the optimal run time performance and data in Encryption techniques.

Data Independence

It is the ability to modify the schema definition in one level without affecting a schema definition in higher level is called Data Independence. There are two types of Data Independencies :-

→ Physical Data Independence → Logical Data Independence

Physical Data Independence

It is the ability to modify the physical schema without causing application programs to be re-written. Modification at disk level are usually to improve performance.

Logical Data Independence

It is the ability to modify the conceptual schema without causing application programs to be re-written. It is usually done when logical structure of database is altered.

* Different types of DBMS Users

1) Naive User - Parametric User

They are also known as Parametric end users, they do not have any knowledge of DBMS but still frequently use the database applications to get the desired results.

With the help of interface provided by the DBMS application Naive user mostly use the database to fill in or retrieve the information. They do not need to be aware of the presence of the database system as they can

M.T.W.T.F.S
Page No. _____
Date _____

interact with Database with the help of
minute driven application interface.
Ex. → Railway Ticket Booking, Zomato.

2.) Application Programmers.

They are also known as Back-end developers, are computer professional users who are responsible for developing the application programs or the user interface so that the other users can use these applications to interact with the Database. They have deep knowledge of DBMS and database and know everything in detail. When needed Application Programmers also specify the modification needed in the Database structure for an application. They are efficient enough in designing or developing their database in any language they know.

3.) Sophisticated Users.

They are the types of database users who know the type and are familiar with the Database. They are usually business analyst, engineers, system analyst etc. They can develop or access their database applications according to the requirements without actually writing the program code for it.

4.) Specialized Users

They are also called Sophisticated user but they are responsible for writing the special database application programs. The specialized user are also responsible to develop the program according to the given requirement.

Ink * Database Administrator and it's functions

Database Administrator is a person having central control over data and programs accessing that data. He co ordinates all the activities of the Database system. He has a good understanding of the enterprise's information ; resources, and needs.

* functions of DBA ~

1.) Schema Definition :

The creation of original database Schema. This involves writing a set of definitions in a DDL (Data Definition Language) compiled by the DDL compiler into a set of tables stored in the data dictionary.

2.) Storage Structure and access method definition;

Writing a set of definitions translated by the data storage and Data Definition Language Compiler.

M T W T F S
Page No. _____
Date. _____

3.) Schema and physical Organization and modification :

Writing a set of definitions used by the DDM compiler to generate modifications to appropriate internal system tables this is done rarely but sometimes the database Schema or physical organization must be modified.

4.) Granting user authority to access database :

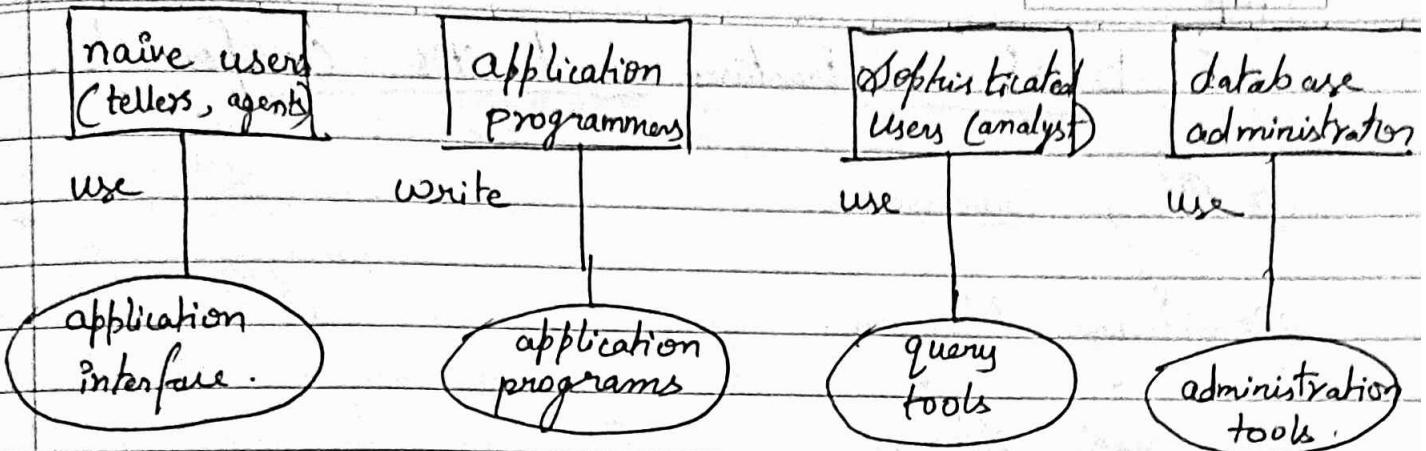
Granting different types of authorization for data access to various users.

5.) Specifying Integrity Constraints :

Generating Integrity Constraints . These are consulted by the Database Manager Module whenever updates occur.

6.) Routine Maintenance :

- Acting as liaison with users
- Monitoring performance and responding to changes in requirements.
- Periodically backing up the database.



Unit -

DBMS Structures and its Components.

1) Query Processor.

2) Storage Manager.

3) Disk Storage.

DDL = Data Definition Language.

Create, Drop, Alter, Truncate

DML = Data Manipulation Language.

Select, insert, update, Delete

→ Database systems are functional into modules for different functions, functional components of a DB processor are:

I) Query Processor ~ It interprets the request (Queries) received from end user via an application programs into instructions. It also executes the user request which is received from the DMATL Compiler. Query processor contains the following component

1) DML Compiler

It processes the DML statements into low level instructions so that they can be executed.

2) Query evaluation engine :-

It executes low level instructions generated by DML Compiler.

3) DDL interpreter -

It processes the DDL statements into a set of table containing metadata (Data over about data)

II Storage Manager ~

Storage Manager is a program that provides an interface between the data stored in the database and the Queries received. It is also known as Database control system. It maintains the consistency and integrity of the database by applying the constraints and executing the DCL statements. It is also responsible for updating, storing, deleting and retrieving data in the database. It contains the following components :-

a) Authorization Manager

It ensures role based access control i.e. it checks whether the particular person is privileged to perform the requested operation or not.

b) Integrity Manager

It checks the integrity constraints if the database is modified when

c) Transaction Manager

It controls concurrent access by performing the operations in a scheduled way that it receives the transaction. Thus it ensures that the database remains in the consistent state before and after the execution of the transaction.

d) File Manager.

It manages the file space and the data structure used to represent information in the database.

e) Buffer Manager.

It is responsible for cache memory and the transfer of data between Secondary Storage and main memory.

III Disk Storage ~

1.) Data files It have the following components.

The files which stores data (all files are storing data)

2.) Data dictionary

Stores These are the structures of database object.

3.) Indices

Indices provides buffer retrieval of data items.

1/2/23 Data Models in DBMS

- 1) Relational data model.
- 2) Entity Relationship (ER) Model.
- 3) Object Based Data Model
- 4) Semi Structured Data Model
- 5) Older Model : Hierarchical Network.

- A datamodel in DBMS is concept of tools that are developed to summarize the description of the database.
- It defines how the logical structure of the database is modelled.
- A data model is collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints.
- It describes the design of a database at each level of data abstraction.
- It defines how data is connected to each other and how they are processed and stored inside the system.

E Relational Data Model

Table of Student ← Editing Entity

Attributes:

Student Id	Student Name	Student Age
1	Ram	23
2	Sayam	22
3	Mohan	20
4	Soham	18

Records

Tuples

→ Most widely used model by commercial data processing application.

→ Use → collection of tables for representing data and relationships among those data.

→ Data is stored in tables called Relation.

→ Each table is a group of column and rows where column represents attributes of an entity and rows represents records (Tuples).

→ This model was initially described by Edgar P F. Codd in 1969.

→ Attribute or field ~

Each column in a relation is called an attribute. The values of attribute should be

from the same domain.

→ Tuples or Records ~

Each row in the relation is called Tuple.
A Tuple defines a collection of attribute values
Each row in a relation contains unique
values.

Advantage & Disadvantage —

Advantage of Relational Data Model.

1 → Structural independence.

2 → Simplicity.

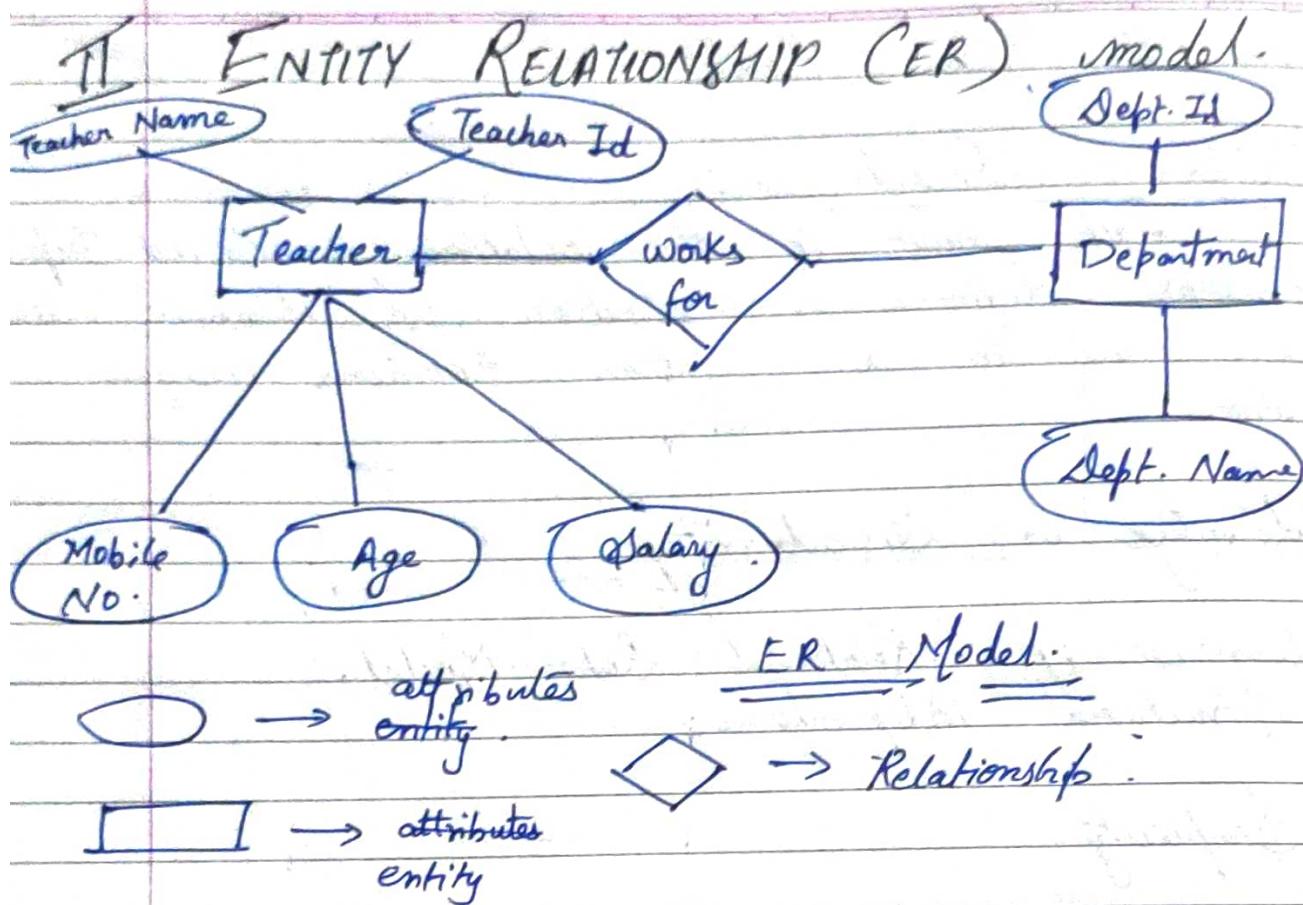
3 → Ease of Designing, Implementation, Maintenance,
and usage.

4 → ^{Adhoc} Adopt Query Capability.

Disadvantage of Relational Data Model.

1 → Hardware Overheads.

2 → Ease of Design can result in bad design



- ER model is a high level data model diagram
- Describes structures of the database with the help of diagram which is known as ER model.
- The ER model is a design or a blueprint of a database that can be latter be implemented as a database.
- It is based on the notion of real world entities and relationships among them.
- ER diagram has the following three components:

a) Entities ~ It is a real-world thing or objects. It can be a person, place or even a concept.

Eg. → Teachers, Students, Course, buildings, departments etc. are the entities of School Management system.

b) Attributes ~ An entity contains a real-world property known as Attributes.

Eg. → the entity teacher has Teacher name, the property like name, age, salary etc.

c) Relationships ~ It tells how two entities are related

Eg - Teacher works for a department.

Advantages ~

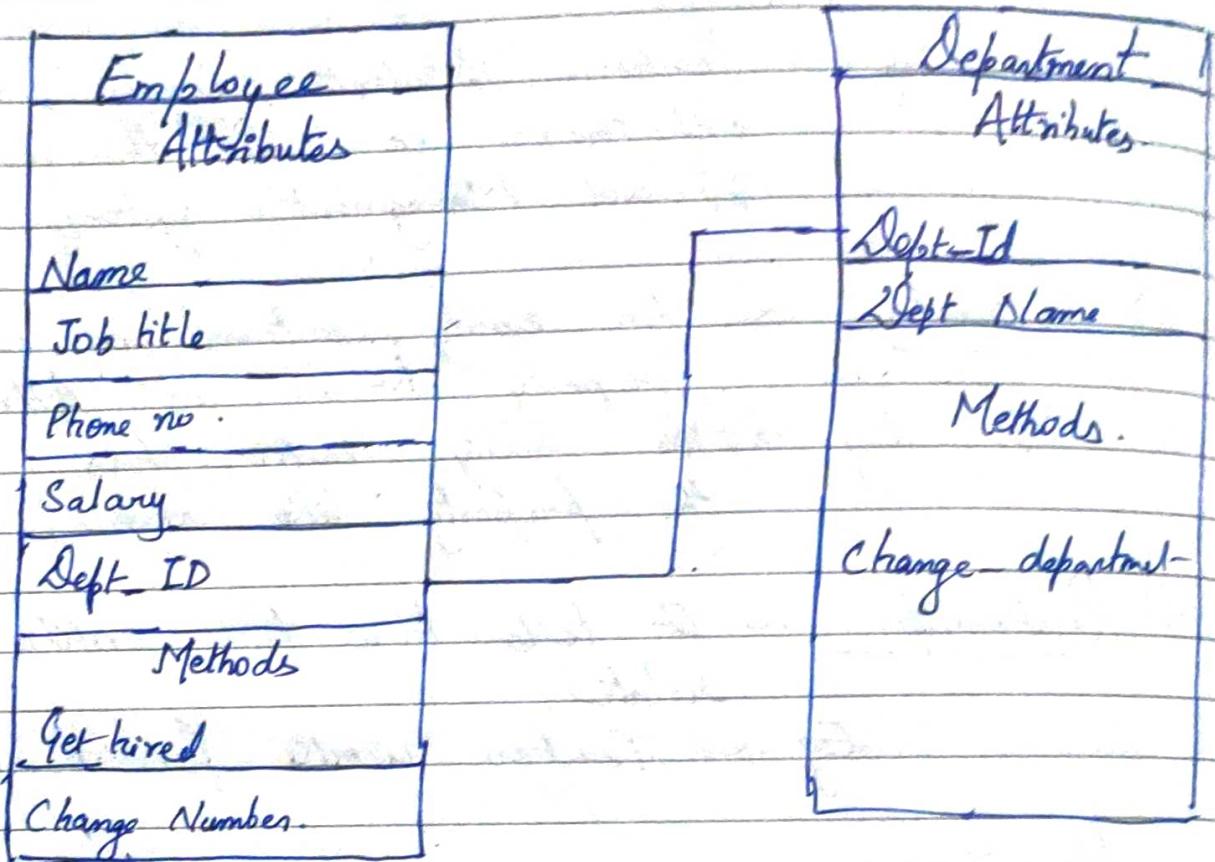
- 1) Straight forward relational representation.
- 2) Easy conversion from ER to other data model.
- 3) Graphical representation for better understanding.

Disadvantages ~

- 1) No industry standard for notation.
- 2) Popular for high level designing.

III OBJECT BASED DATA MODEL.

a) Object oriented data model.

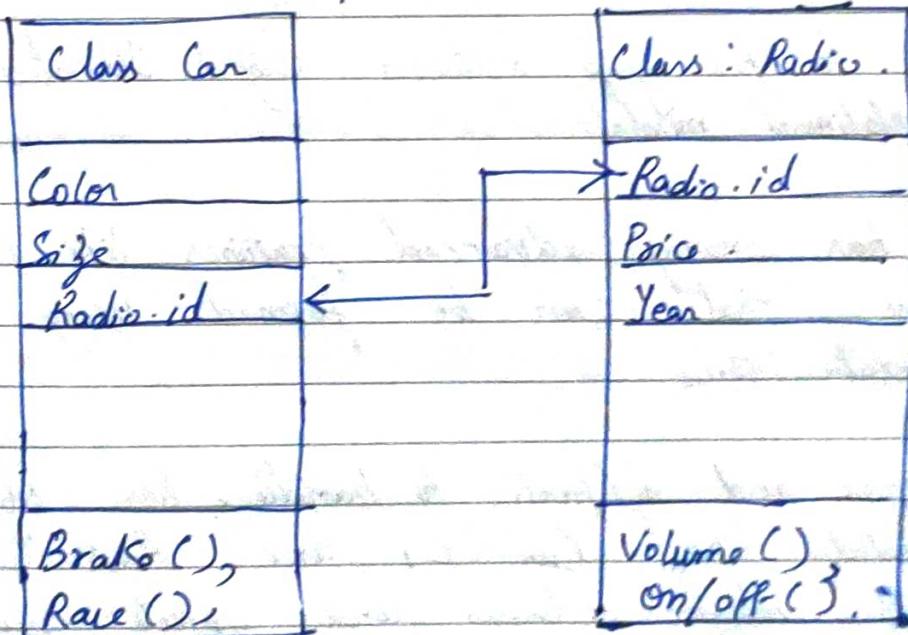
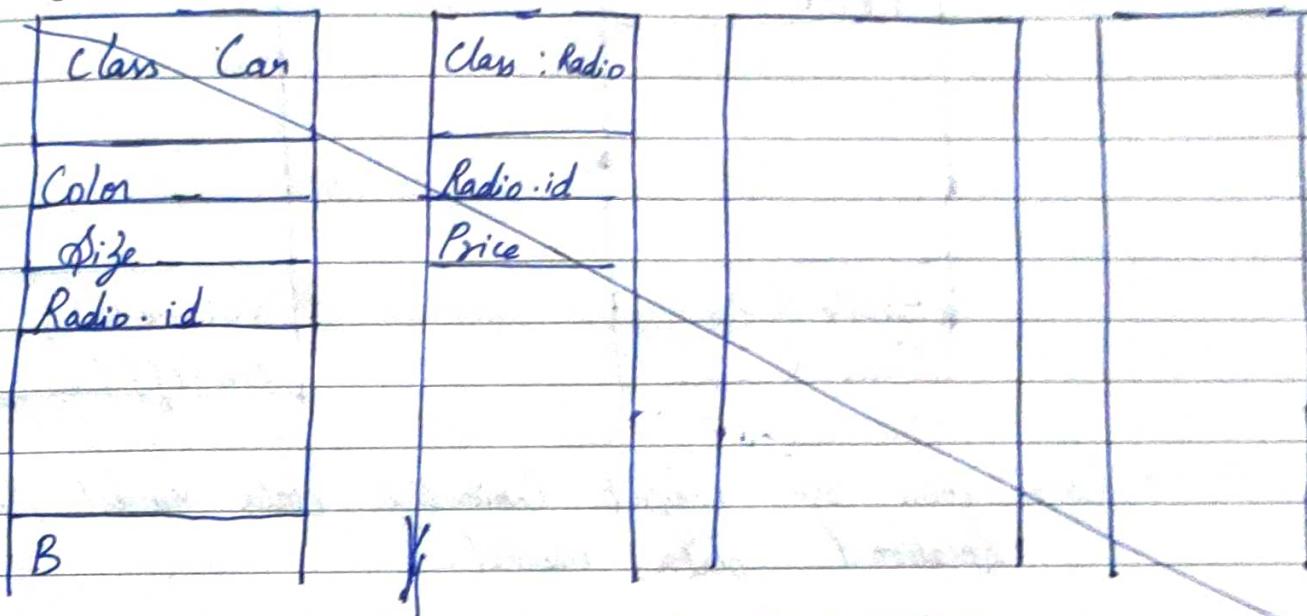


- An extension of ER model with notion of functions, encapsulation and object identity as well.
- Both the data and relationships are present in a single structure known as object.
- Two or more objects are connected through links we use this link to relate one object to other objects where employee and department are objects.

Two objects are connected through a common attribute i.e. dept-ID and the

communication between these two will be done through this, Id.

b) Object Relational Data Model -



[P-T.O]

Car : C ₁	Radio : R ₁
Red	9876
Small	500
9876	2019
Brake(), Brake(),	Volume(), On/Off()

- Combination of Object Oriented data model & Relational data model.
- Built to fill gap between Object Oriented model and Relational model.
- It has many advanced features like Complex data-types that can be formed using the existing data items.
- Complex and difficult to handle, here objects are created like Car : C₁ and Radio : R₁, by using above classes and objects we can find the year of radio i.e. 9876.

Advantage of Object Oriented model

- Capable of handling large variety of data types.
- Combining Object Oriented Programming with database technology.
- Improved productivity.
- Improved data access.

Disadvantages of Object Oriented model.

- No precise definition.
- Difficult to maintain.
- Not suited for all applications.

[P.T.O]

III SEMI - STRUCTURED DATA MODEL.

[P.T.O].

SEMI - STRUCTURED DATA MODEL

Example

{Represented in XML, json}

```
<student 1>
  <Roll No> . . . . . </Roll No>
  <Name> . . . . . </Name>
  <Class> . . . . . </Class>
  <Age> . . . . . </Age>

</Student 1>

<Student 2>
  <Name> . . . . . </Name>
  <Class> . . . . . </Class>
  <Age> . . . . . </Age>

</Student 2>
```

→ An evolved form of Relational Model.

→ This model allows the data specification at places where the individual data items of the same type may have different sets of attributes.

→ In this model some entities may have missing attributes while others may have extra attributes.

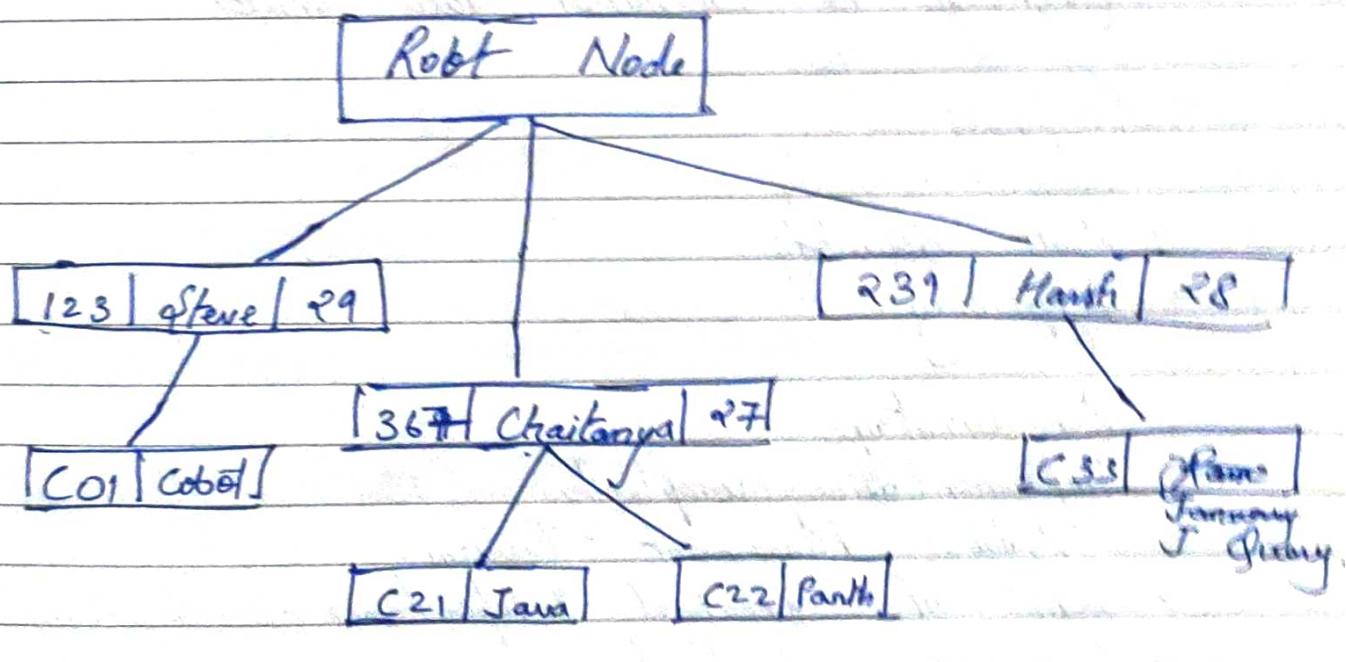
→ This model ~~also~~ gives flexibility in storing data, it also gives flexibility in storing attributes for ex. → If we are storing any value in any attribute then that value either be

atomic value or collection of values.

→ XML is used for representing the semi-structured data.

→ Here in Student 2 Roll no. is missing but in Relational model, it is not possible that Student 1 and Student 2 have different attributes.

V HIERARCHICAL DATA MODEL.



One to many relationships.

→ First DBMS Model.

→ Data is organized into a tree like structure with each record having one parent record and many children.

→ Drawback

- ① It can have only one to many relationship between nodes
- ② This model is rarely used now.
- ③ In the Dig., we have few students and few courses, a course can be assigned to single student only however student takes any no. of courses.

Advantage of hierarchical data model

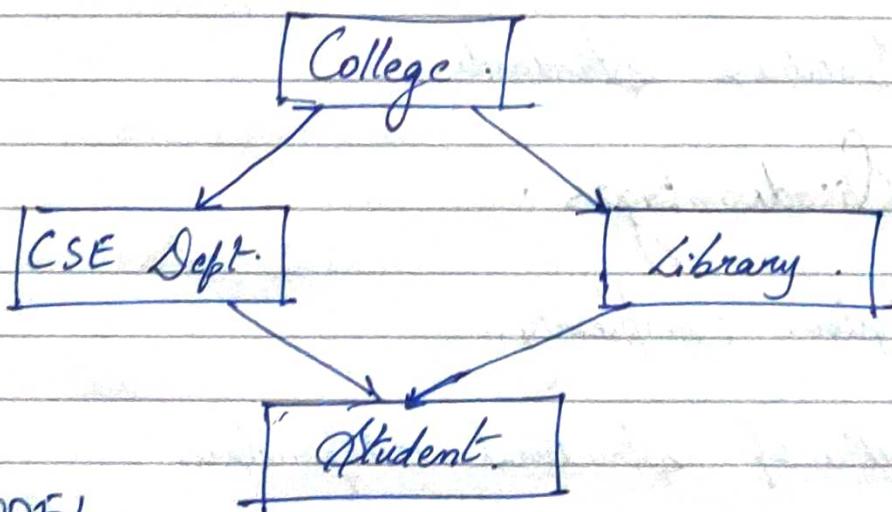
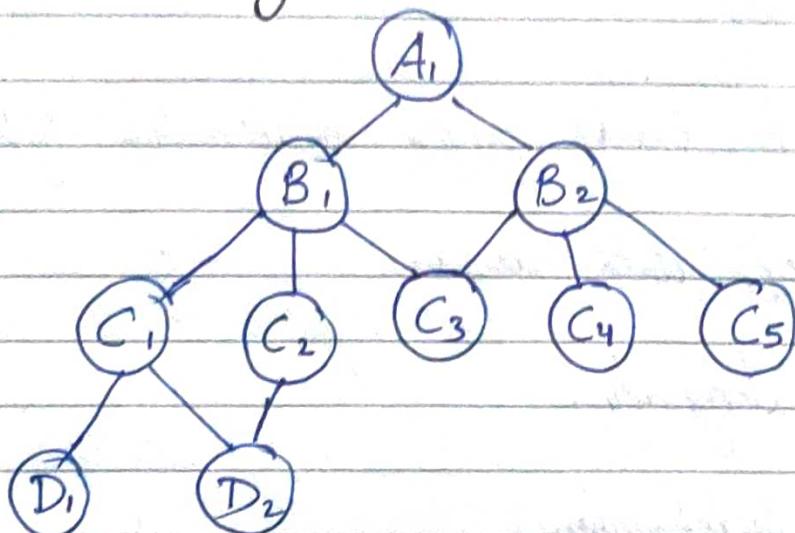
- 1 → Simplicity.
- 2 → Data integrity.
- 3 → Data Security.
- 4 → Efficiency.

Disadvantage

- Implementation complexity.
- Database Management Problem.
- Lack of Structural Independence.
- Programming Complexity.
- Implementation Limitation.

Network Complexity

Graph-like
Structure.



NETWORK MODEL.

- Extension of hierarchical model
- Most popular model before relational model
- Same as Hierarchical model except that it has graph like structure rather than tree based structure and are allowed to have more than one parent node.
- Supports many to many relationships. Here node 'Student' have two parent nodes i.e. 'CSE dept' and 'Library' nodes.

Advantages .

- Simplicity .
- Ability to handle more relationship types .
- Ease of data access .
- Data Integrity .
- Data Independence .
- Database standards .

Disadvantages .

- System complexity .
- Lack of structural independence .

ER MODEL IN DETAIL .

* ENTITY SETS .

Entity set is a set of entities of the same type that share same properties or attributes.

ex. → Student set , Teacher set

→ Entity set need not to be disjoint .

→ Entity set should be Capital Letter

→ Attributes →

→ It should be noun and first letter of each word should be capital

→ Should follow standard format

for ex. → Student_GPA not GPA_of_Student

Types of Attributes :

1.) Simple and Composite .

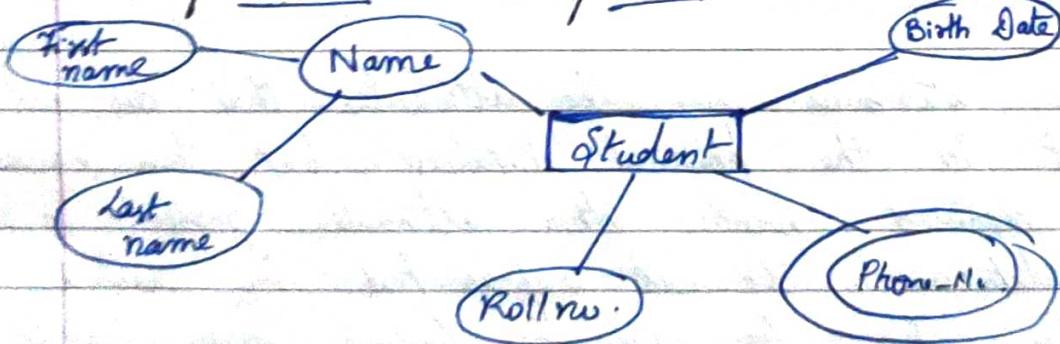
2.) Single valued or Multivalued .

3.) Stored or derived .

4.) NULL or Prime .

5.) Key attributes .

1.) Simple and Composite .

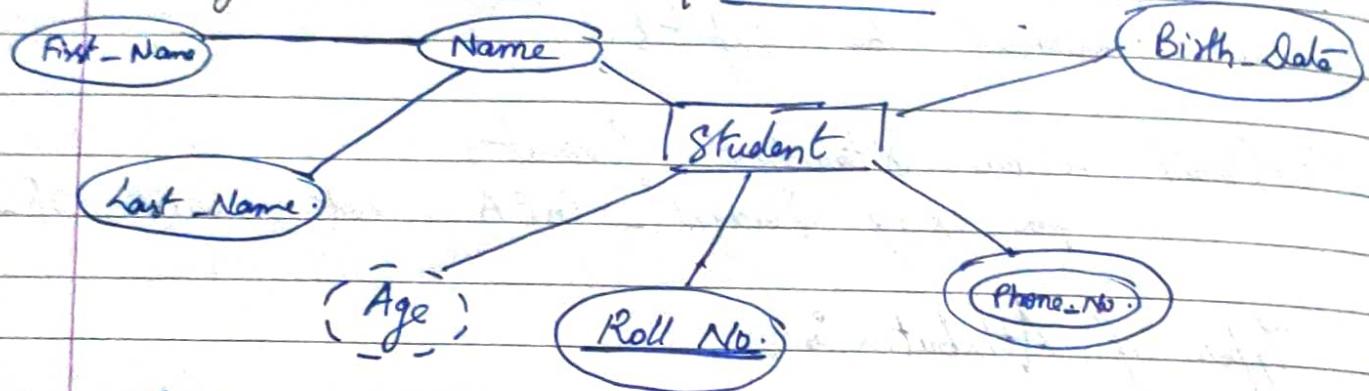


Simple-Attributes which are not divided into sub parts are called simple attributes ex. → Customer id , Customer - id .

Composite-Composite attributes are made up of more than one single simple attributes ex. → Student's complete

name may have First-name and Last-name.

2) Single Valued or Multivalued.



→ Single-valued attributes contain single value.
Ex. → Roll no., Aadhar card no.

→ Multi-valued attribute may contain more than one value and is represented by double-ellipse.
Ex. → Phone No.

3) Stored or Derived.

{refer same dig.}

→ Stored attributes are physically stored in a database.
Ex. → Roll No., Name etc.

→ Derived attributes are the attributes that do not exist in the physical database but their values are derived from other attributes present in the database. It is represented by dashed ellipse. Ex. → Age can be derived from Date of Birth.

4) NULL or Prime

NULL - An attribute takes NULL value when entity does not have a value for it. It may indicate not applicable N.A.

Ex. - One may have no middle name.

Prime - An attribute whose values are used to identify an entity within an entity set is known as Prime attribute. It is similar to Key attribute. Ex. → Registration - No., Roll - No.

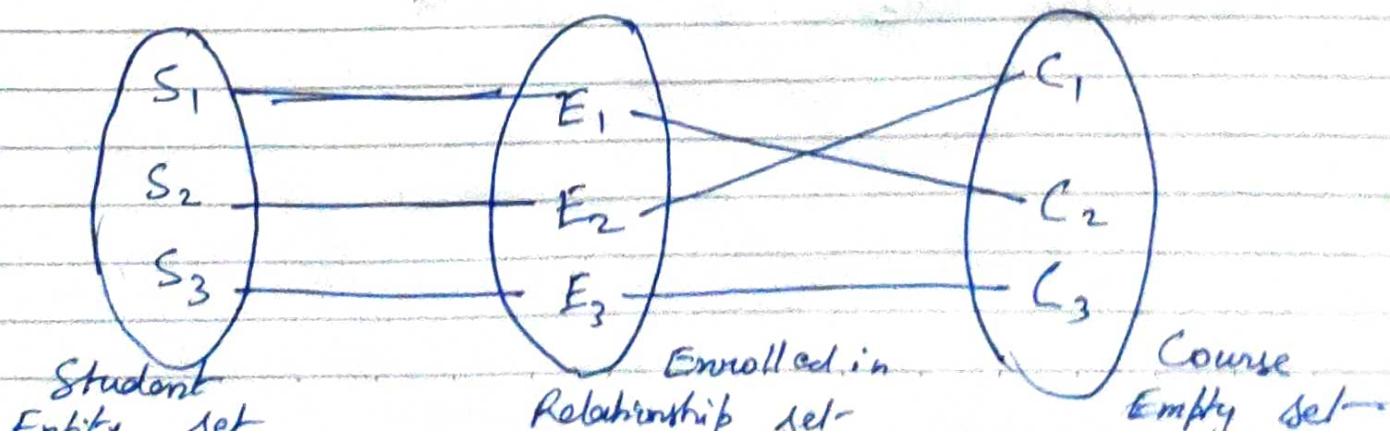
5) **Key attributes** attributes which uniquely identifies each entity in the entity set is called as Key attribute. If represents a primary key. It is represented by ellipse with a underline.

Ex. → Roll - No.

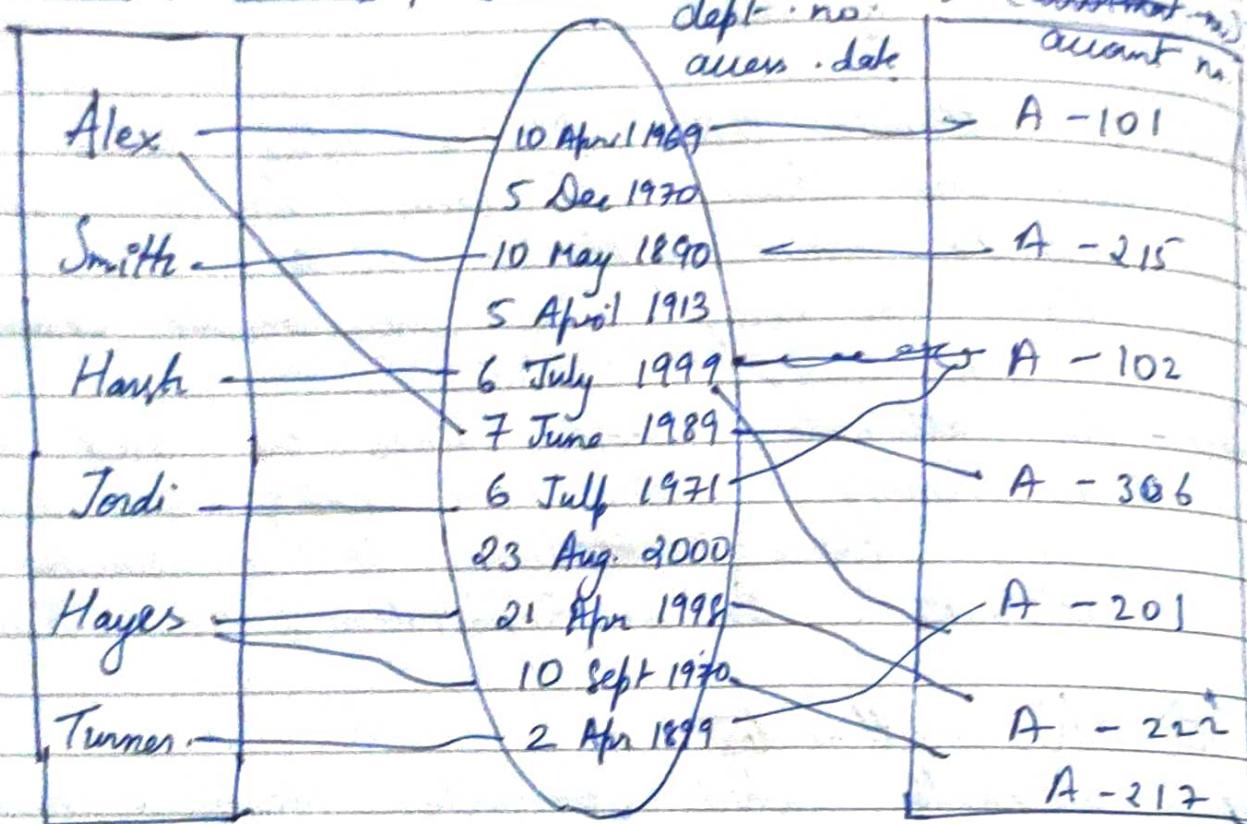
* RELATIONSHIP SET

Set of relationship of similar type is called Relationship set.

Ex. → Relationship Set Enrolls (E₁, E₂, E₃) .



Relationship Descriptive attributes



RELATIONSHIP DESCRIPTIVE ATTRIBUTES

These attributes are called Descriptive attributes.

Ex. →

The depositor relationship sets between Entity sets customer and account may have the attribute access date

Degree of Relationship

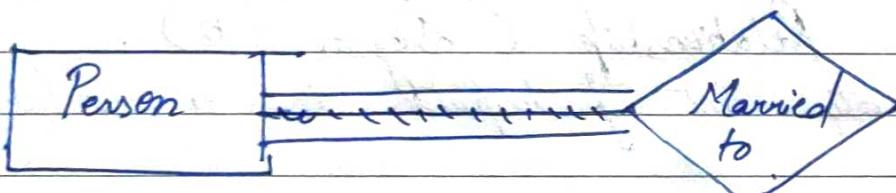
We have - Unary, Binary, Tertiary and A N-any degree of relationships.

The number of different entity sets participating in a relationship is called as Degree of Relationship set.

~ Unary Relationship (Degree = 1)

Here only one entity participate in a relationship.

Ex. →



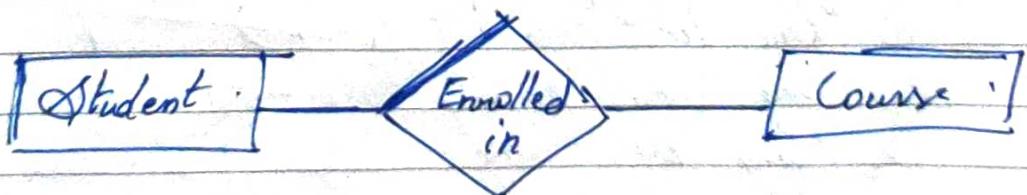
One person is married to only person.

~ Binary Relationship (Degree = 2)

Here two entities participate in a relationship and is the most common Relationship degree.

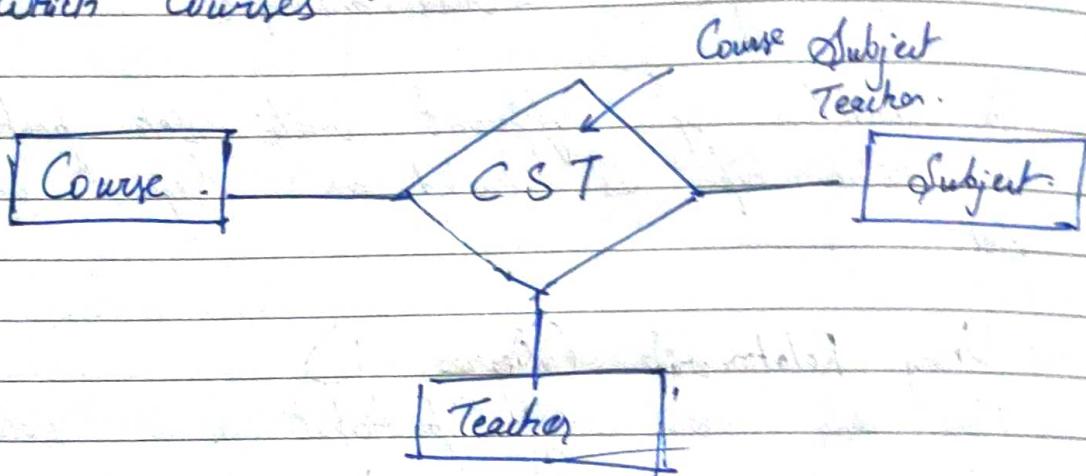
Ex. →

Student is enrolled in course.



→ Ternary Relationship (Degree = 3).

Here three entities participate in the relationship.
 Ex. → The university might need to record which teacher taught which subject in which courses.



→ N-ary Relationship (Degree = n).

N entity set participate in a relation.

MAPPING CARDINALITIES.

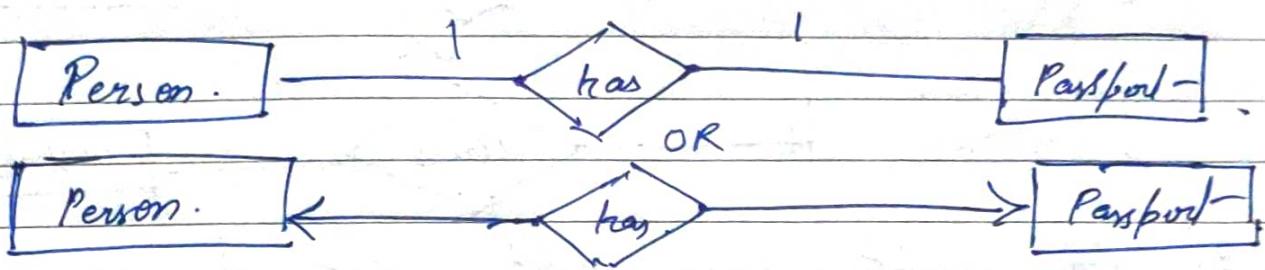
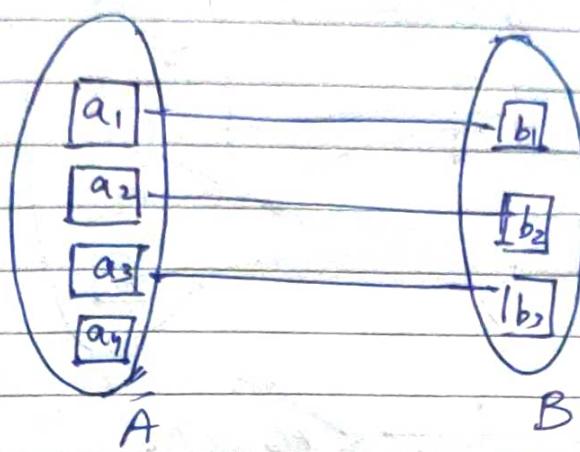
→ Cardinality defines the number of entity of an entity set participates in a relationship set.

→ Most useful in describing binary relationship.

→ Different type of cardinalities or relationship are :-

- 1.) One to One (1 - 1)
- 2.) One to many (1 - M).
- 3.) Many to one (M - 1).
- 4.) Many to many (M - M).

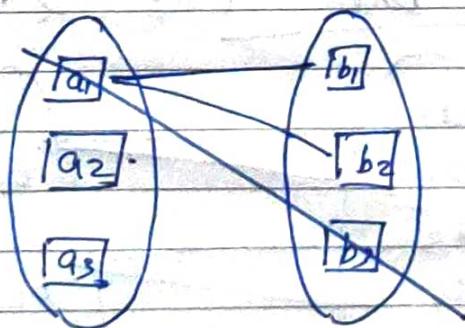
1) One to One (1-1)



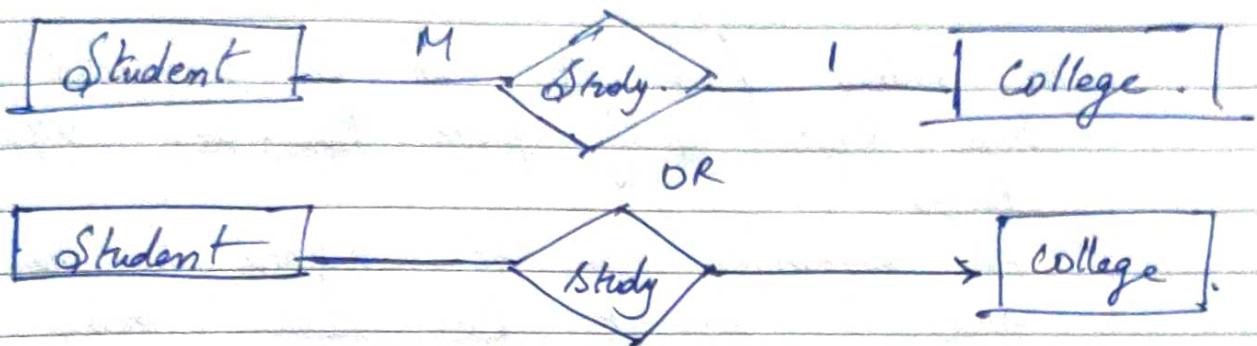
→ One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

Ex. → Person has ^{only} one passport and a passport is given to one person.

2) One to Many (1-M)



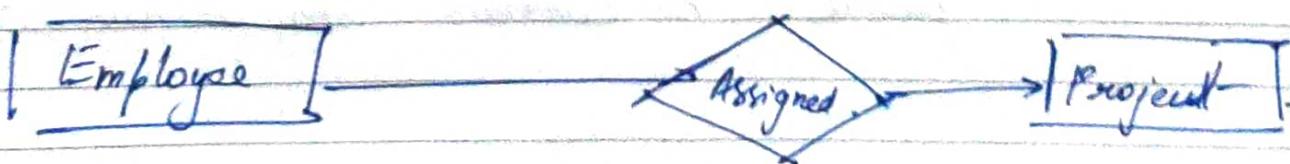
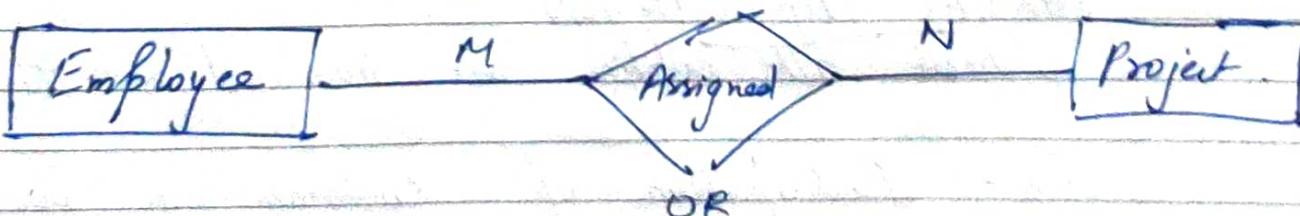
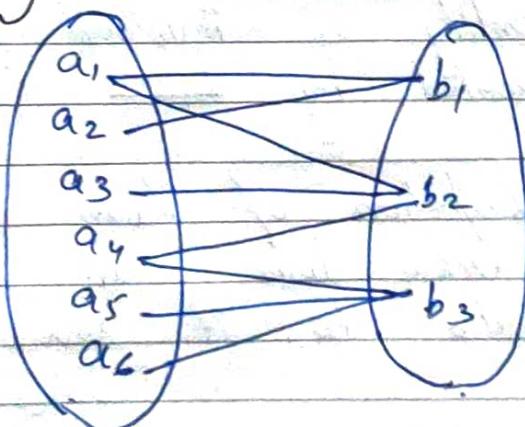
[P.T.O]



More than one entities from entity set A can be associated with almost one entity of entity set B, However an entity from entity set B can be associated with more than one entity from entity set A.

Ex. → Many students can study in a single college but a student cannot study in many colleges at the same time.

Many -to - Many (M-M).

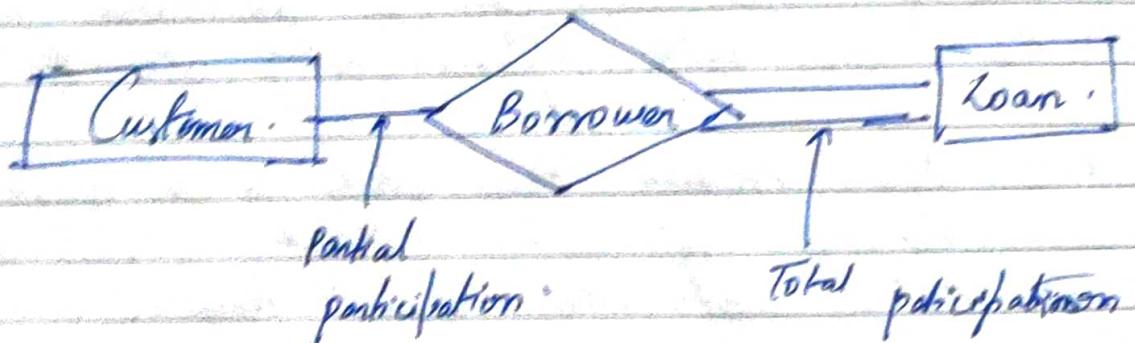


One entity from A can be associated with more than one entity from B, and vice versa.

Ex:-

An employee can be assigned to many projects, and a project can have many employees.

PARTICIPATION CONSTRAINTS



TOTAL PARTICIPATION

Each entity is involved in the relationship represented by == (double line)

Ex:-

Participation of Loan in Borrower is Total, i.e. every loan must have a customer associated to it via Borrower.

PARTIAL PARTICIPATION

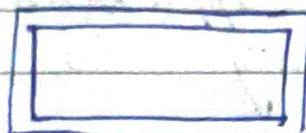
Not all entities are involved in the relationship

Ex:- Participation of customer in borrower is partial. A customer may have no loan.

STRONG & WEAK ENTITY SET



Strong



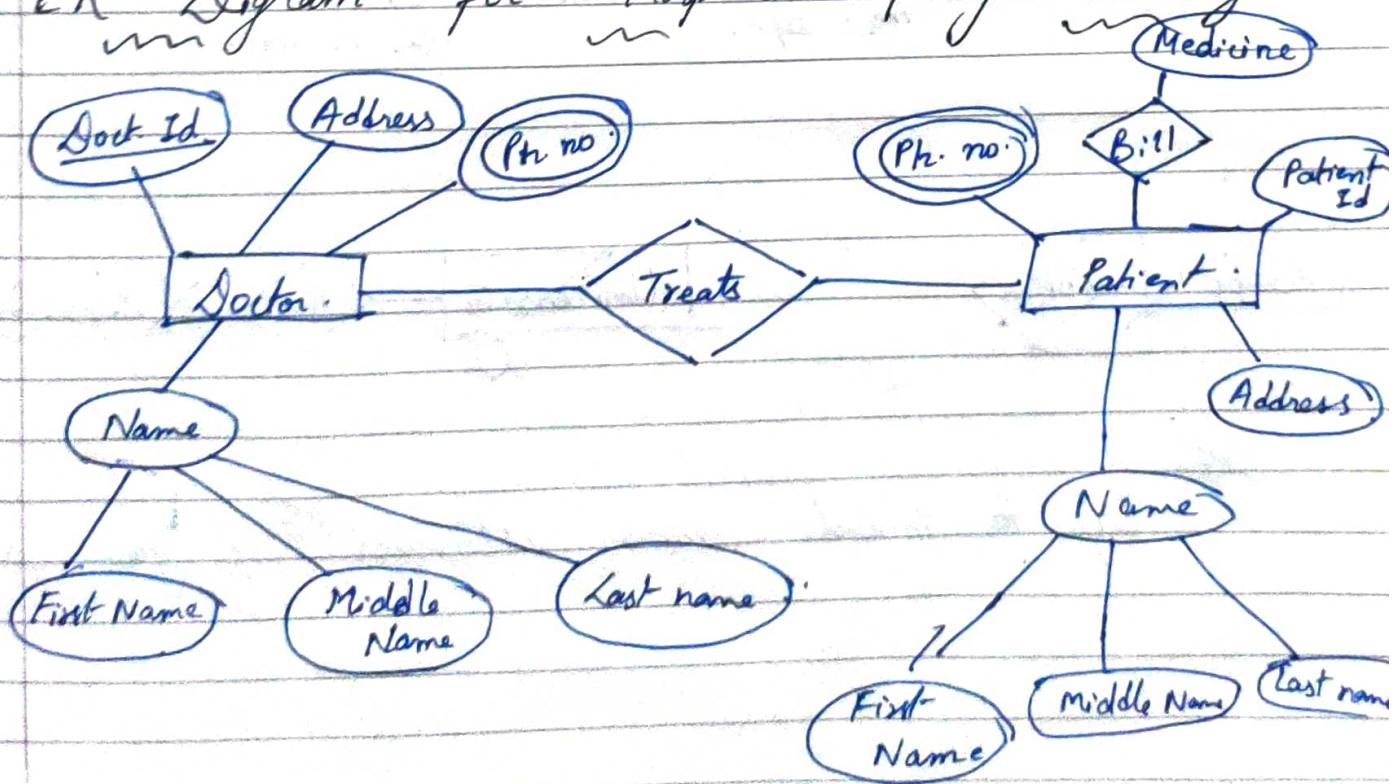
Weak

Strong entity always have a primary key, it's existence is not dependent on any other entity, a set of Strong entity is called Strong entity set.

Weak entity does not have sufficient attributes to form a primary key i.e. it has no primary key. It is dependent on Strong entity to ensure its existence. Set of weak entities is known as Weak entity set represented by double rectangle.



ER Diagram for Hospital Management System.

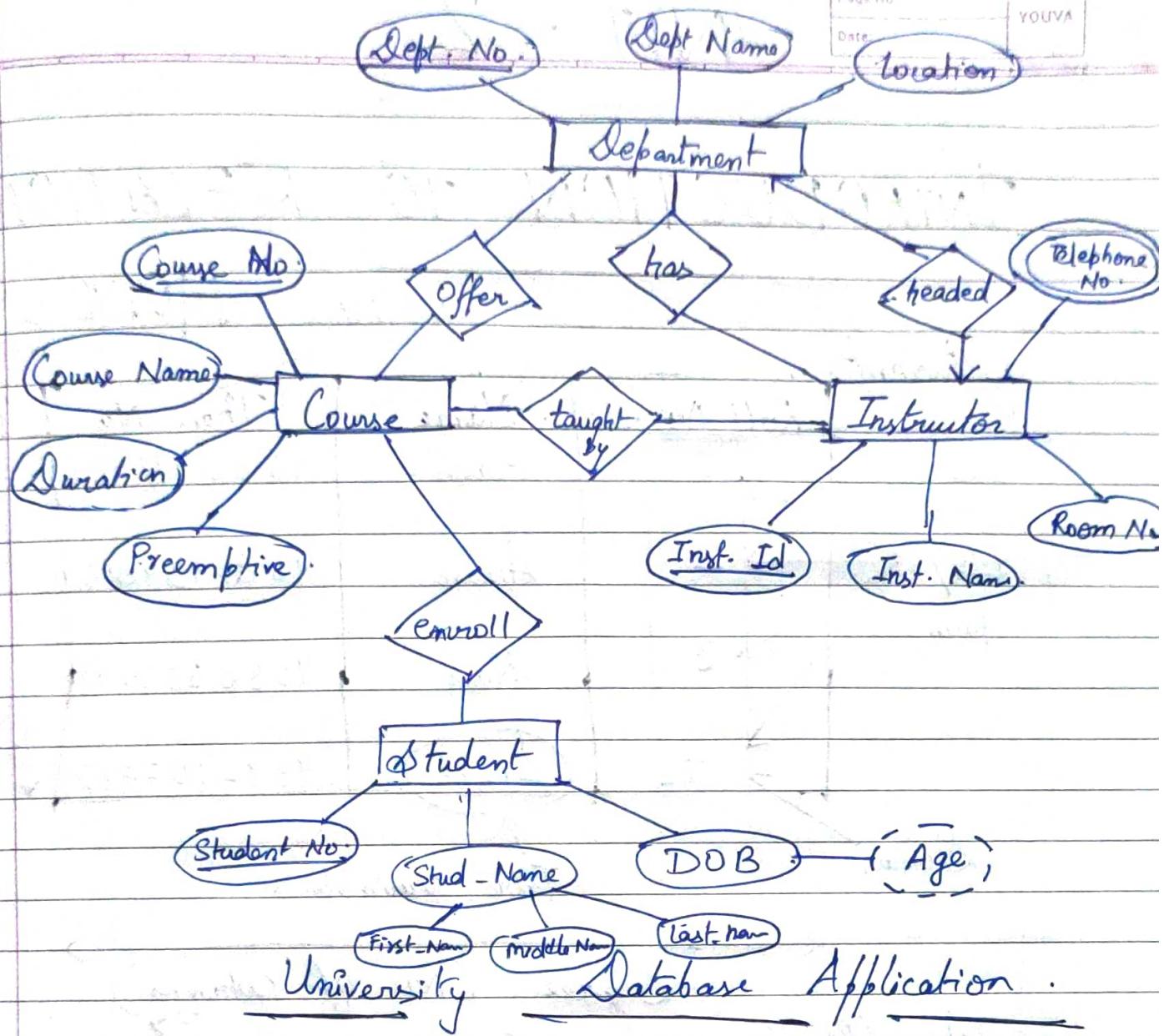


- Q) Draw an ER Diagram for an university database application where:
- A university has many departments.
 - Each department has multiple instructors, one among them is the Head Of the Department.
 - An instructor belongs to only one department.
 - Each department offers multiple courses, each of which is taught by a single instructor.
 - A student may enroll for many courses offered by different departments.

Relationship \rightarrow Offer^{dept, course}, has^{course instr.}, taught by^{ins. dept.}, headed by^{dept. or inst.}, enroll^{stud, exam}.



* Try to include more attributes ~~if you can~~ if you can.



[P.7.0]

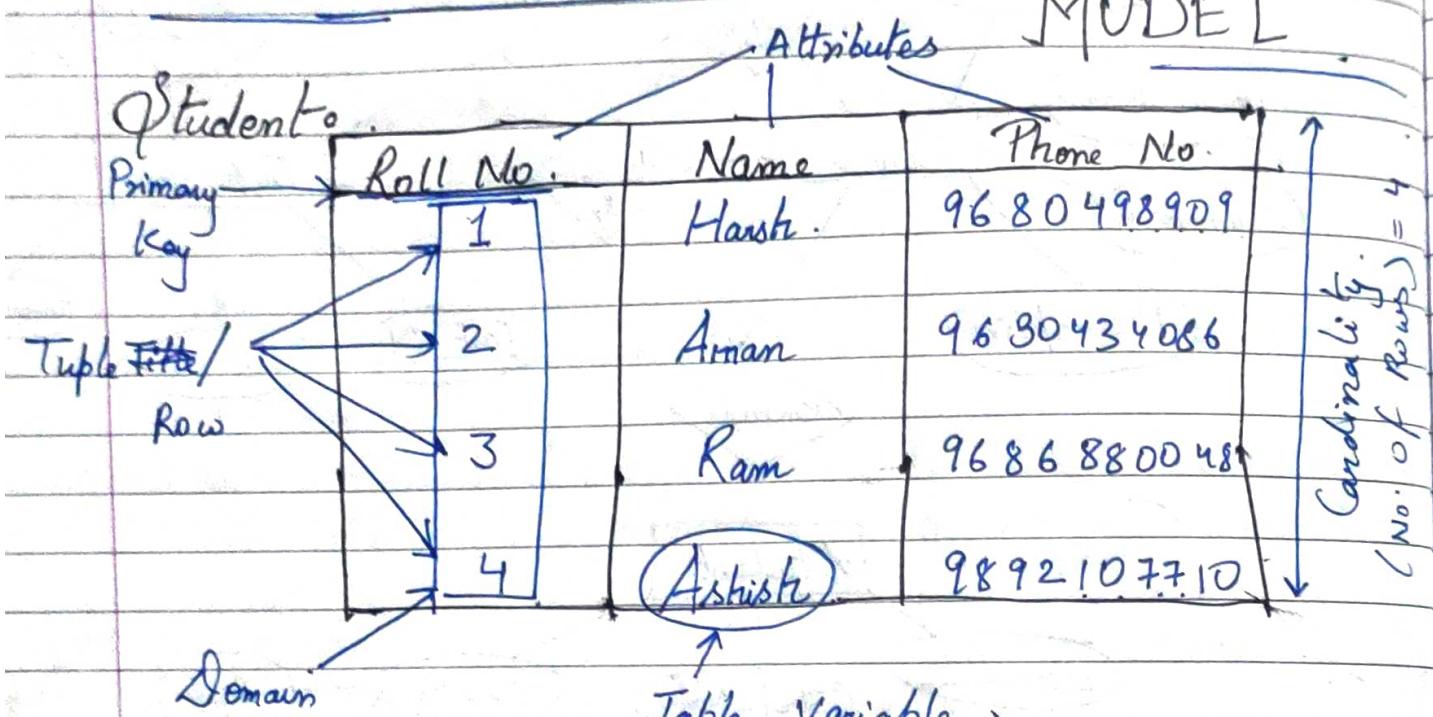
✓
Monday
4

UNIT - 2

M	T	W	T	F	S	S
Page No.						
Date:						YOUVA

INTRODUCTION TO RELATIONAL

MODEL



Degree (No. of columns) = 3

Table / Relation

→ Student.

Relational Schema :

Student (Roll No., Name, Phone No.)

→ The relational model is the theoretical basis of Relational databases

→ The Relational model of data is based on the concept of relations.

→ A Relation is a mathematical concept - based on the ideas of sets.



- After designing the conceptual design of DB using ER diagram, we need to convert the conceptual model in the Relational model which can be implemented using RDBMS languages like Oracle, SQL, MySQL etc.
- RDBMS is a database management system i.e. based on the Relational model as introduced by E.F. Codd.
- Currently popular RDBMS includes:
 - 1) DB 2 and Informix Dynamic System from IBM.
 - 2) Oracle & Rdb - from Oracle.
 - 3) SQL Server & MS Access - from Microsoft.

Domain - A domain is the set of allowable values for one or more attributes.
 Ex. → Phone no. Should contain numeric values only.

Relational Schema - It represents the name of the relation with its attributes.

~~Keys & its~~

Keys and its types in DBMS.

Keys → An attribute or set of attributes that uniquely identifies any record (Tuple) from the table.

1.) Super Key -

Employee -

Emp_id	Name	Aadhar - No.	Email_id	Dept.
1	Harsh	4956 2019 8909	Harsh@gmail.com	1
2	Harshit	4986 2020 8505	harsh03@gmail.com	2
3	Ragini	5085 2003 9003	R.Kumari@gmail.com	2
4	Shruti	7078 2013 9008	Shruti@gmail.com	3

1. { Emp_id }
2. { Aadhar - No. }
3. { Email_id }
4. { Email_id , Aadhar - No. }
5. { Emp_id , Aadhar - No. }
6. { Emp_id , Email_id }
7. { Emp_id , Aadhar - No. , Email - No. }

etc -

Super Key - A Combination of all possible attributes that can uniquely identify the rows (tuples) in the given relation.

- Super Key is a Super set of candidate Key
- A table may have many Super Keys.
- A Super Key may have all additional attributes that are not needed for unique identity.

2.) Candidate Key -

- It is an attribute or set of attributes which can uniquely identify a tuple
- Candidate Key is a minimum Super Key or Super Key with no redundant attribute.
- It is called a Minimal Super Key because we select a Candidate Key from a set of Super Key such that Selected Candidate Key is the minimum attribute required to uniquely identify a table.
- Candidate Keys are defined as distinct set of attributes from which primary key can be selected.
- Candidate Keys are not allowed to have Null values.

→ Ex. of Candidate Key
{ Emp-id }, { Email-id }, { Aadhar-No }

3) Primary Key =

- Primary Key is one of the Candidate Key chosen by the database designer to uniquely identify the tuple in the relation.
- The value of primary key can never be Null.
- Value of Primary Key must always be unique (not duplicate data).
- The value of this key can never be changed i.e. no updation is possible.
- The value of this key must be assigned when inserting a record.
- A relation is allowed to have only one Primary Key.
Ex - { Emp-id },

4.) Alternate Keys -

Out of all Candidate Keys only one gets selected as a primary key, remaining keys are known as Alternate Keys.
Ex:- Aadhar-No. {, Email-ID }.

5.) Foreign Key -

Employee Table (Referencing relation).

Emp-id	Name	Aadhar-No.	Email-No.	Dept-id
1	Harsh			1
2	Harshit			2
3	Ragini			2
4	Shruti			3

Department Table (Referenced Relation).

Primary Key →

Dept-id	Dept. Name
1	Sales
2	Marketing
3	HR

- Foreign Key is a key used to link from two tables together. It is an attribute or set of attributes in one table that refers to the Primary Key in another table.
- Purpose of the Foreign Key is to ensure Referential Integrity of the data.
- Department table is primary table / Master table / Parent table.
- Employee table is Foreign table / Child table.
- Foreign Key can take only those values which are present in the Primary Key of the Referenced relation.
- Imp. → Foreign Key may have a name other than that of a Primary Key.
- Foreign Key can take Null values.
- There is no restriction of on a Foreign key to be unique; in fact Foreign Key is not unique most of the time.

-6) Composite Key -

Cust. Emp. id	Error. id	Product. id	Product. Cols.	Product. Count
Customer C01	001	P111		5
C02	012	P111		8
C02	012	P222		6
C01	001	P333		9

→ A Key that has more than one attributes is known as Composite Key also called Compound Key.

→ Any of the above attributes cannot be taken as Primary Keys so combination of two attributes can uniquely identify this relation. So it is called Composite Key.
 eg → { Cust. - Id } { Product - Cols }

[P . 7 . 0]

INTEGRITY CONSTRAINTS OVER RELATION

- Integrity constraints are used to ensure accuracy and consistency of data in a Relational database.
- Integrity constraints are set of rules that the database is not permitted to violate.

* Types of Integrity Constraints :

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

→ Domain Constraint :

Defines the domain or the valid set of values for an attribute. The datatypes of domain are → String, integer, Character, -datatype, date, currency etc.

Ex. → Age of Student contains only integer value.

→ Entity Integrity Constraint. : value be

If states that primary key cannot be NULL. This is because primary key value is used to identify individual rows in a relation, and if the primary key has NULL value then we cannot identify those rows.

Page No. _____ Foreign key

Emp (Table 1) Related / Referencing table			
Emp - id	Emp - Name	Age	Dept - No.
111	Mohan	21	1
112	Rohan	33	2
113	Sohan	27	3
114	Ram	25	5

Not allowed as Primary
Dept. No. 5 is Key.
not defined as primary key in
Department table.

↓ Dept. Table (Table - 2)

Dept. No.	Location
1	Mumbai
2	Delhi
3	Abida

(Referenced Table)

- Referential Integrity constraint is specified between two tables.
- It is enforced when a Foreign Key references the Primary Key of a table.
- If a Foreign key refers to the Primary Key of Table - 2 then either every value of the Foreign Key in table 1 must be available in Primary Key value of table - 2 or it must be NULL.
- Some more rules are
 - We cannot delete a record from Primary table, If the matching record exist in the Related table.

- You cannot change Primary Key value in the Primary table, if that record has related record.
- You cannot insert a value in the Foreign Key field of the related table that does not exist in the Primary Key of the Primary Table.
- However you can enter a Null value in the Primary Key, you can specify that the records are unrelated.

→ Key Constraints :

Stud-id	Name	Age	Sem.
111	Mohan	21	1
112	Roham	20	2
113	Soham	27	3
111	Ram	25	5

Not allowed because all rows must be unique.

Key Constraint Specify that in any relation all the values of the primary key must be unique. The value of primary key must not be NULL.

Assignment

Imp Codd's 12 Rules.

(8 marks).

will not be

done in Lab

Integrity Constraints in SQL

- 1) UNIQUE
- 2) NOT NULL
- 3) PRIMARY KEY.
- 4) CHECK
- 5) DEFAULT
- 6) FOREIGN KEY .

DDL , Create Statement

Create table statement (roll no. number (3) ,
Names Varshini (20) , Sem number (1) ,
branches Varshini (20) , marks number (2) ,
Pno. number (10)) ;

* Data Definition Language (DDL)

It is used for creating and modifying the database objects such as tables , indices , views and users .

DDL Commands → used to define the structure and schema of the database
All DDL commands permanently saves all the changes in the database . Following are the DDL commands :

- 1) CREATE - To create new Table / Database .
- 2) ALTER - To alter (or modify) structure of table .
- 3) DROP - To delete a table from database .
- 4) TRUNCATE - To delete all records from the table .

SS RENAME - To Rename table.