Aditya Dhanaraj Kundu

Database Management System

YCS 4003

Q.1. What is Data?

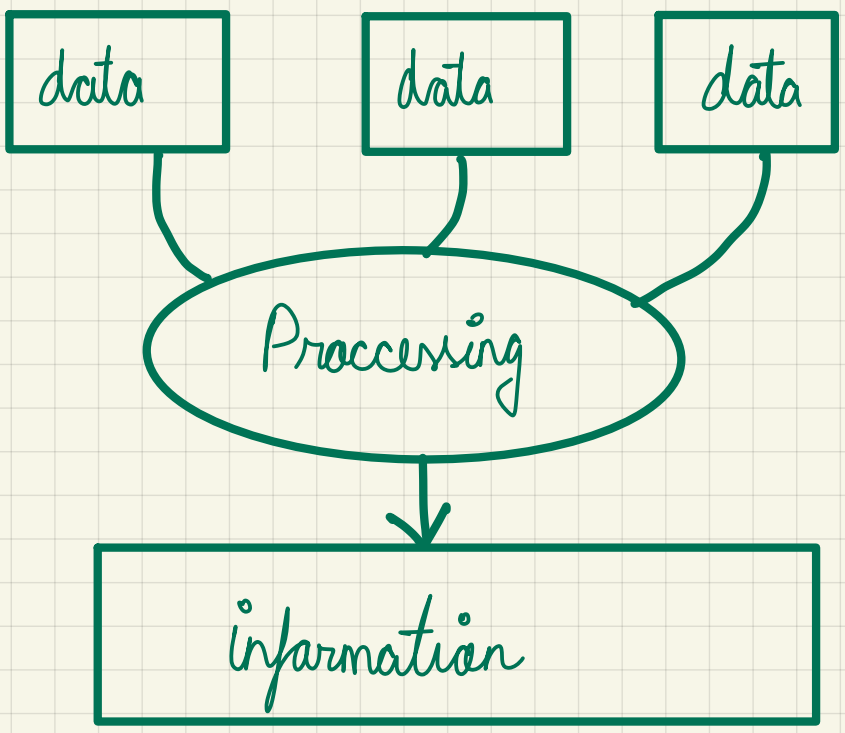
Ans: Data is raw material that can be processed for any computing machine to produced information.

Q.2. What is information?

Ans: Information is the data that has been converted into more useful or intelligent form. Example Marksheet, Report.

The information is needed for the following reasons:

- To keep the system up to date.
- To gain the knowledge about surroundings.
- To know about the rules and regulation of the society.



- Data V/s Information :-

Data	Information
1) It may or may not be meaningful.	1) It is always meaningful.
2) It is raw fact.	2) It is proccesed form of data
3) Understanding is difficult	3) Understanding is easy.
4) Phenomenal fact	4) Organised fact.

• Concepts and Overview of DBMS :-

→ insert, update, delete.

A Database is a collection of stored operations. Data used by the application system of some particular enterprise. User of the system can perform the following operations :-

a) Adding new empty file to the database.

b) Inserting data into existing file.

c) Deleting Data from existing file.

d) Removing existing file from the database.

Example of enterprises are — Manufacturing Company, Bank, University etc.

So, Company → Product Data || Bank → Customer data || University → Student data

Remember

DBMS involves 4 major components → ① Data ② Hardware ③ Software ④ Users.

• Advantages :-

- Redundancy can be controlled. In traditional non-database system each department has it's own file for handling it maintainance. data proccesing application.

Eg: The university database may have two group of users — ① Personal department ② The accounting department.

That means most of the data stored twice in files of each department.

The type of problem arise are — ⓐ The storage may be wasted.

ⓑ Files that stored data may became inconsistent due to partial updation.

- Data can be shared : Sharing means not only existing application can be shared the data in the database, But also the new application can be developed to be operate. against the same stored data.
- Standard can be enforced → Industry standard can be enforced and maintain by the Database Administrator. (DBA)
- Data integrity can be maintained : Integrity implies the correctness and accuracy of data stored in the database. Centralised control of the data enables the Database administrator to define integrity constraints.

- **Disadvantage** :-

- Cost : High initial investment in hardware, software and training overhead, for providing security recovery, concurrency control and integrity function.

- **Table creation** :-

```
CREATE TABLE <table name>  ⟶ Syntax
( field_name1 field type (width);
  field name 2 field type (width);
  field name N field type (width);
);
```

```
CREATE TABLE student
( roll_no number (5);
  name char (30);
  birth date date );
```

Table created ✓

Student

roll. no.	name	birthdate
1	Sudip	20-Feb-09
2	Charan	22-Feb-10

- **Inserting into the table** :-

INSERT INTO <table name>

VALUES (V_1, V_2, V_3, ...);

INSERT INTO Student

VALUES (1, 'Sudip', '20-Feb-09');

DESC <table name>

update table name set column 1 = value 1, column 2 = value 2 ....

Column n = value n

where [Condition];

Employee name, employee id, address, Phone no, designation, salary

09/Jan/2023

- **Database Language** :-

i) DDL → Data Definition Language

ii) DML → Data Manipulation language.

iii) DCL → Data Control Language.

DDL : A database schema is specified by a set of definition expressed by a special language called DDL. The result of compilation of DDL statement is a set of tables that is shored in special file called data dictionary or data directory. This file is consulted before data are read or modifying in the data base system.

The SQL DDL provides commands for defining relation, schemas, deleting relation, creating indices & modifying relation schemas.

Eg: CREATE (Table, View..)
ALTER (Table, view - - -)
DROP (Table, view - - -)

DML : By DML we mean the retrival of information stored in the database.

- The insertion of new information into the database.
- The deletion of information from database -
- Modification of information of the database.

A DML is a language is a language that enable users to access or manipulate data as organised by the appropriate data model.

Basically of 2 types : ① Procedural DML (what data - how to get those data) ② Non procedural DML (what data - without specifying how to get those data)

Eg: SELECT, INSERT, UPDATE, DELETE

DCL: DCL is used to control the data population. It enables users to speed the beginning & ending of transaction. It deals with the authorization, access rights, recovery and integrity. issues. The following statement are used to control the data in the data in database —

Recovery and concurrency: COMMIT, ROLLBACK

Security: GRANT, REVOKE

- DBA (DataBase Administrator):-

DBA is a team of computer professional responsible for the design & maintain of database. The main task of DBA.

① Deciding information content.

② Designing the database.

③ Backup & Recovery.

④ Authorization Check.

The example of distributed database is Banking or Railway Reservation System.

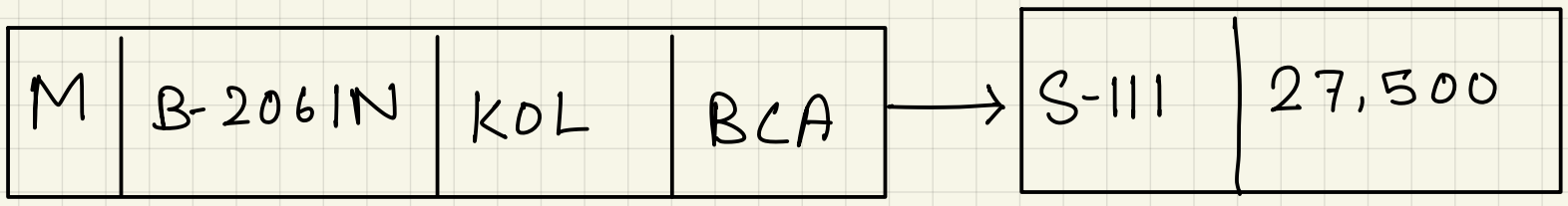
Example of centralized database is Library System.

- **DataBase Users :-**
  - DBA
  - DB designer
  - End Users
  - Tool developers
  - System Analysts and application programmers
  - Other users → DBMS system designer & implementer
  - Operation & maintainance personnel.

10/Jan/2023

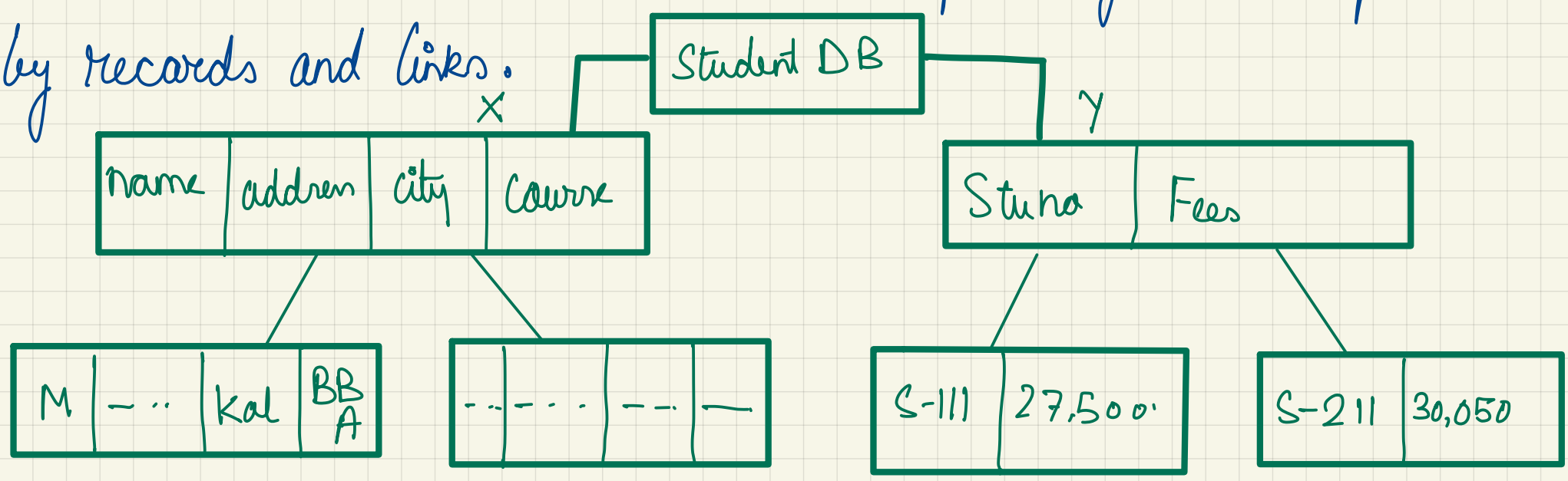
- **Network Model :-**

A network model data are represented by collection of records & relationship among data are represented by lists (pointer). The structure of network model is like that.

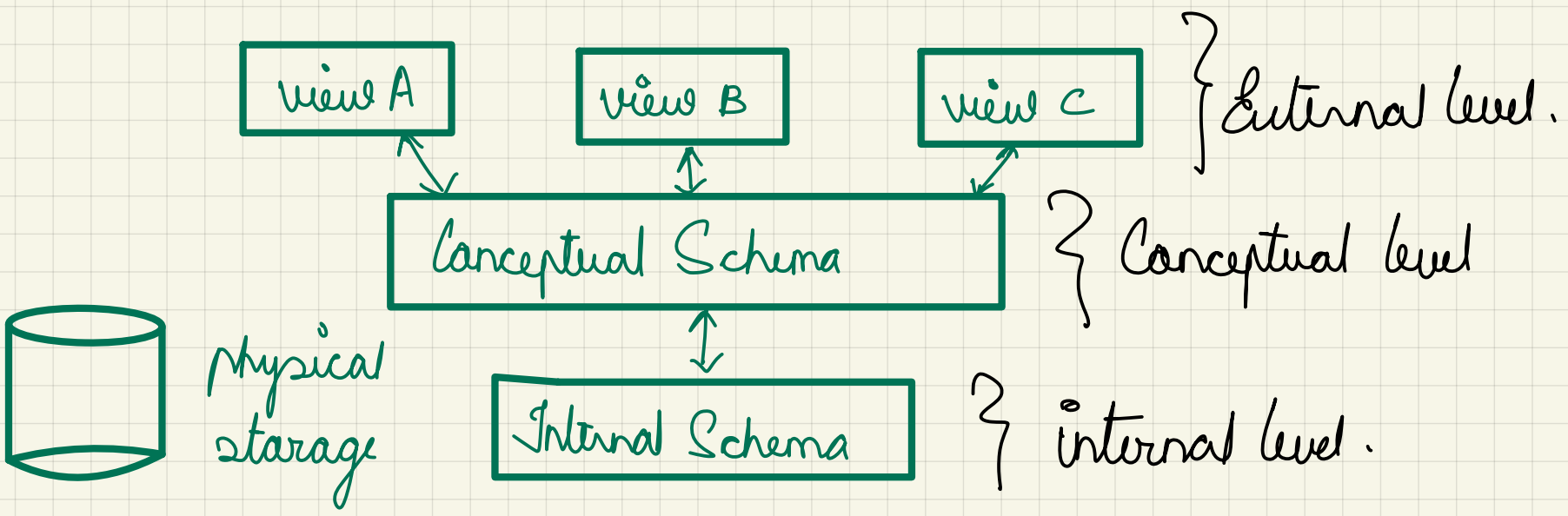


- **Hierarchial Model :-**

Hierarchial database model is one of the oldest database model. This model is based on the assumption that a tree structure is the most frequently occuring relationship. A hierarchial model is identical to the network model in the sense that data & relationship among data are represented by records and links.



SPARC Architecture / ANSI / Three Schema :-



- **Schema :** The description of database is called schema. It describes during database design and is not expected to change frequently.

- **External level :** The external level is the individual user level. The concept of data independence will help you to understand the external level. There are two types of data independence. (i) logical data independence (ii) Physical data independence.

- **Logical data independence :** The logical data independence is the ability to change the conceptual schema without changing the external schema.

- **Physical data independence :** The physical data independence is the ability is change the internal schema without changing the conceptual schema.

- **Conceptual Level :-**

The conceptual level is an illustration of the entire database content. The most commonly used conceptual model is the entity relationship model.

- **Physical level / Internal level** :-

Physical level defines the physical storage structure of the database that is how the database is actually stored. Also describes the complete details of the data storage and access paths for the database.

- **Data model** :-

Is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraint.

A data model provides a way to describe the design of a database at the physical, logic and view level. Three types of data model are —

i) Relationship model
ii) Network model
iii) Hierarchial model.

- **Relational model** :-

The relational model uses tables to represent the data and the relationship among those data each table has multiple column and each column has unique name.

Student X

Name	add	City	Course	no.
M	- - -	Kalkata	BCA	S-111
N	- - -	Delhi	B Tech	S-211
O	- - -	Mumbai	M. Tech	S-311

Y

no	Fees
S-111	27,500
S-211	30,000
S-311	40,000

16/Jan/2023

• Entity And Attributes :-

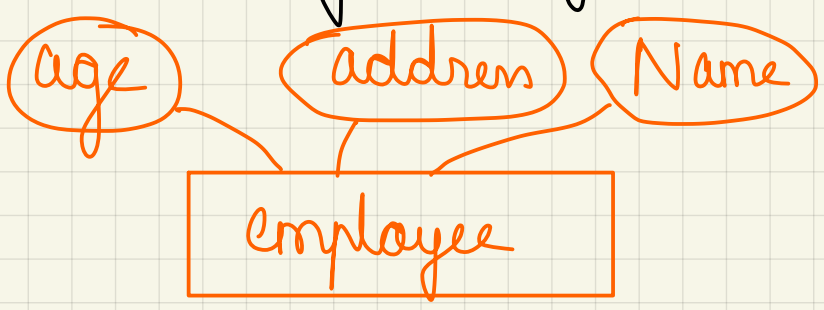
The basic object that the E-R (Entity Relationship) model represent is an entity, which is a thing in the real world with an independent existence. An entity may be an object with a physical existence (car, house etc) or it may be an object with a conceptual existence (company, university, course etc). Each entity has attributes - the particular properties that describes it. A particular entity will have a value for each of its attributes.

Eg: Employee {Name, Age, Address}

Employee — entity

Name, Age, Address — attributes

Entities are denoted by rectangle and attributes are denoted by oval.



• Different types of attributes :-

i) Simple and Composite attribute

ii) Single Valued And Multi Valued Attributes

iii) Derived attribute

iv) Null attribute

• Simple Versus Composite : Attributes which can not be divided into subparts or we can say Attributes composed of a single component is called a simple attribute

On the other hand, Comparite attribute can be divided into subparts.

Eg: Employee { Name, Age, Address }

Age → Simple attribute.

Name → F.Name, M.Name, L.Name (Comparite attribute)

entity

- **Single and Multi Valued**: Most attributes have a single value for a particular entity, such are called single valued attributes.

Eg: Age

In some cases an attribute can have set of values for the same entity, such attributes are called multi valued attributes.

Eg: Colours of a car, degree of a person

- **Derived Attribute**: The value for these type of attribute can be derived from the value of other related attributes. If you consider the Date of Birth attribute of an employee then we can get the age attribute from the DOB attribute and the current date. So the date attribute is called store attribute and age is called derived attribute

- **Null Attribute**:- A null value is used when an entity doesn't have a value for an attribute. For an example, if a particular employee has no dependance, the dependent name value will be null.

Module - ②

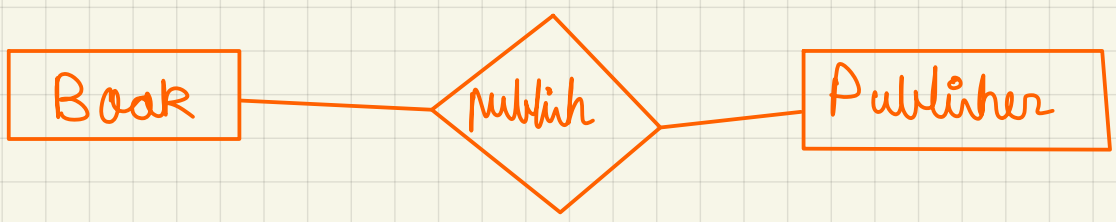
17/Jan/2023

- **Relationship :-**

"A Book (entity) is published by a publisher (entity)"

The relationship or association that exists between the entity related data item to each other in a meaningful way. This information relationship must be captured by the database model as association exists between entities. Each relationship is identified so that each name describe the relationship.

The name of the relationship are represented by active or passive verbs. It is denoted by the diamond shaped symbols.



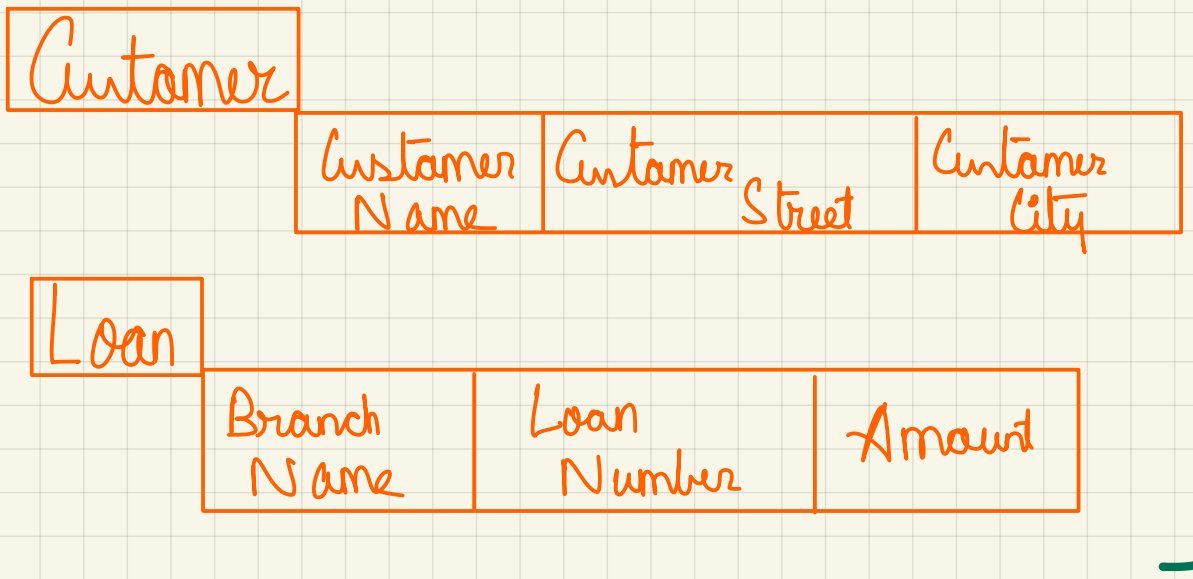
- **Entity Set :-**

An entity set is a set of entity of same type that share the same properties or attributes. The set of all person who are customer at a given bank can be defined as the entity set Cutomer. The individual entities that constitute a set are said to be the extension of the entity set. Thus all the bank Customers are the extension of the entity set Customer.

- **Relationship Set :-**

A relationship set is an association among several entities. A relationship set is a set of relationship of the same type.

Eg:



- **Mapping Constraints** :-

Cardinality ratio for a binary relationship specifies the number of relationship instantaneous that an entity can participate in.

Mainly there 4 types of Cardinality ratio exist for a binary relationship

i) One to One → 1:1

ii) One to Many → 1:N

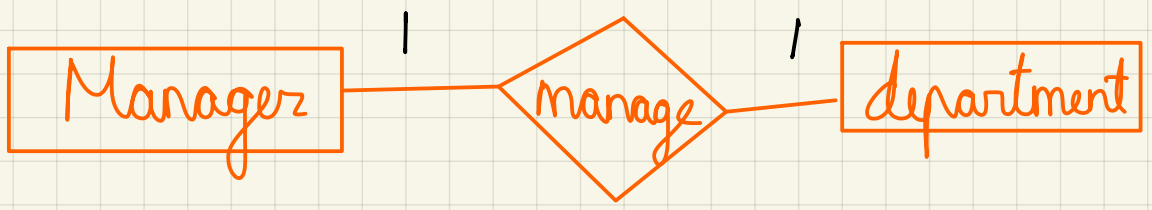
iii) Many to One → N:1

iv) Many to Many → N:N

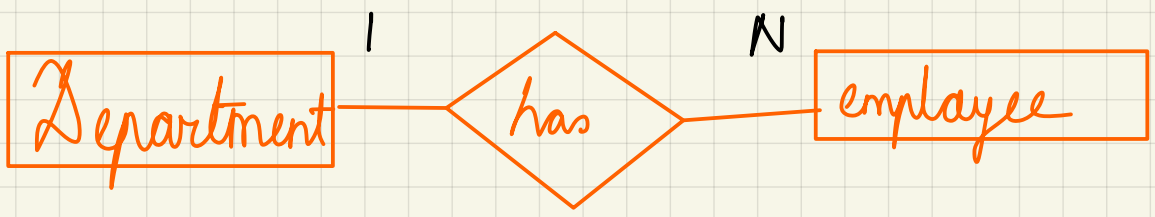
REMEMBER

Mapping cardinality or Cardinality ratio means to denote the number of entities to which another entity can be linked through a certain relation set.

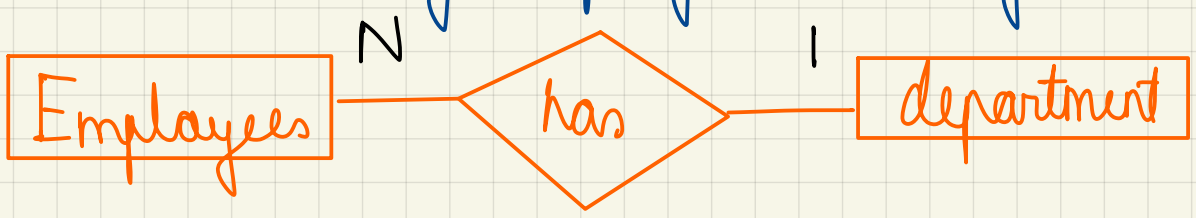
One to One : Example → Manager manage department.



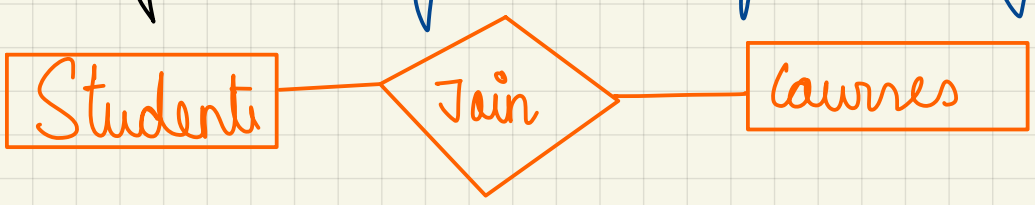
One to many : Example → A single department has many employees.



Many to One : Many employees has only one department.



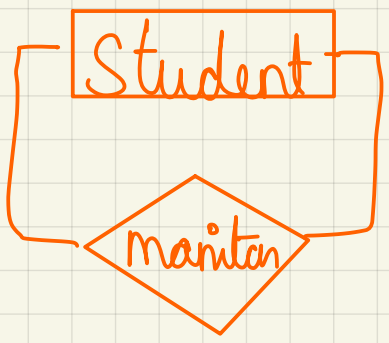
Many to many : Many students join many courses.



- **Degree of Relationship** :-

The number of participating entity in a relationship is called the degree of relationship. It may be Unary, Binary, Ternary, Quaternary

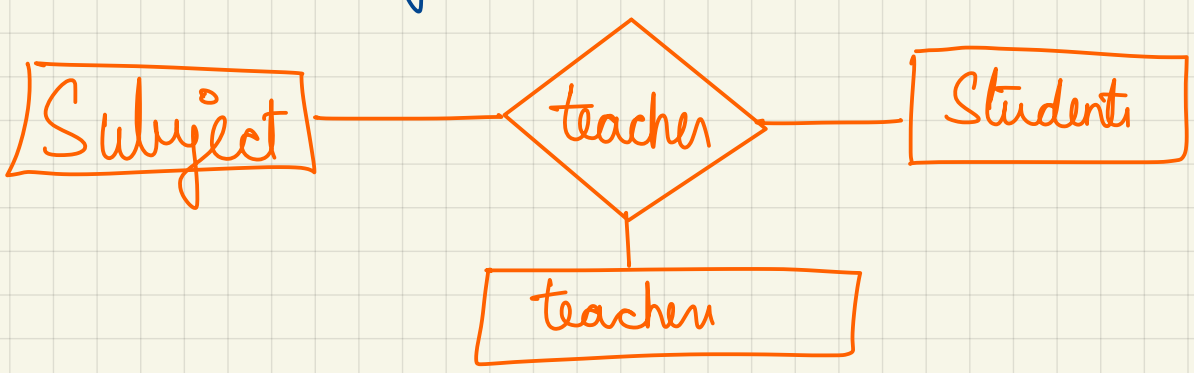
Unary Relationship : Itself when an association is maintain within an single entity. Eg: Class monitor is also a student, so there is only single entity student.



Binary Relationship : A Binary relationship is when two entities participate in a relationship. It is the most common relationship degree.

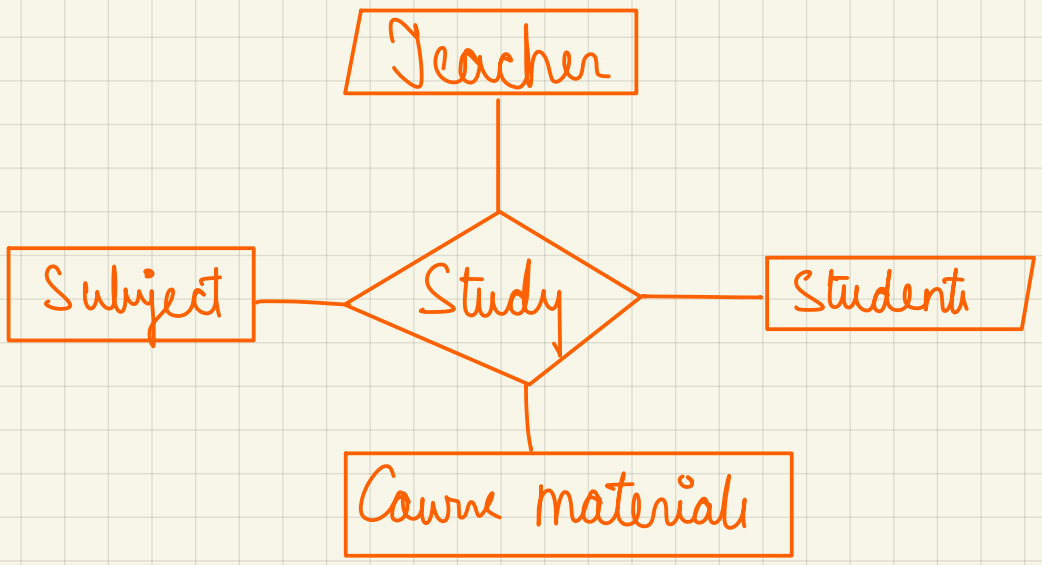
Eg: Teacher teaches Student

Ternary Relationship : It is when three entities are participating in a relationship. Eg: Teacher teaches a subject to students.



Quaternary Relationship : A quaternary relationship is when more than 3 or 4 entities participate in a relationship.

Eg: A student is studying a subject by a teacher with the help of some course material.



24/Jan/2023

Key: The value of a key attribute can be used to uniquely identify each tuple in a relation. Key is determined from the meaning of the attributes. It stores unique value. There are 4 types of keys

(I) Primary Key
(II) Super Key
(III) Foreign Key
(III) Candidate Key
(IV) Composite Key

(I) **Primary Key :** A primary key is a candidate key choosen by the DBA to identify the entities in an entity set. The primary key is used as the primary access mechanism for the entity set. Primary key values are underlined.

Eg. Teacher { name, dept, designation, parents name, specialisation };

↳ Underlined primary key.

(II) **Super Key** : Combining one or more than one attributes if we can uniquely identify a tuple from the relationship then the key is called Super key.

Eg: Student { name, roll no, class, DOB, mob no }

→ Super Key { name, roll no }

Teacher { name, dept, designation, parents name, specialization }

→ Super Key { name, dept }

(III) **Foreign Key** : Suppose in a relation R_1 there exists a primary key of R_2 then the primary key is a foreign key of R_1 referencing to R_2.

Eg. Employee { emp-id, name, age, salary, dept name, address }

Department { emp id, dept name, location }

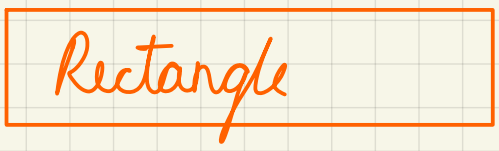
(IV) **Candidate Key** : Minimal Super key are called Candidate Key where no proper subset is a Super key.

(V) **Composite Key** : A composite key is a candidate key with more than one attribute.

- **Entity-Relationship (E-R) diagram** :-

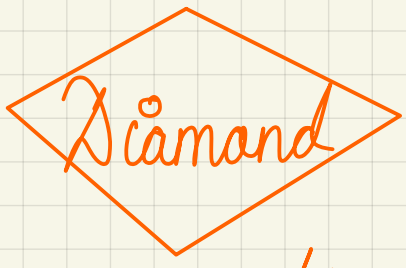
The overall logical structure of a database can be expressed graphically by an E-R diagram.

Components :



→ Represents entity set

→ Represents attributes



→ Represents relationship set.

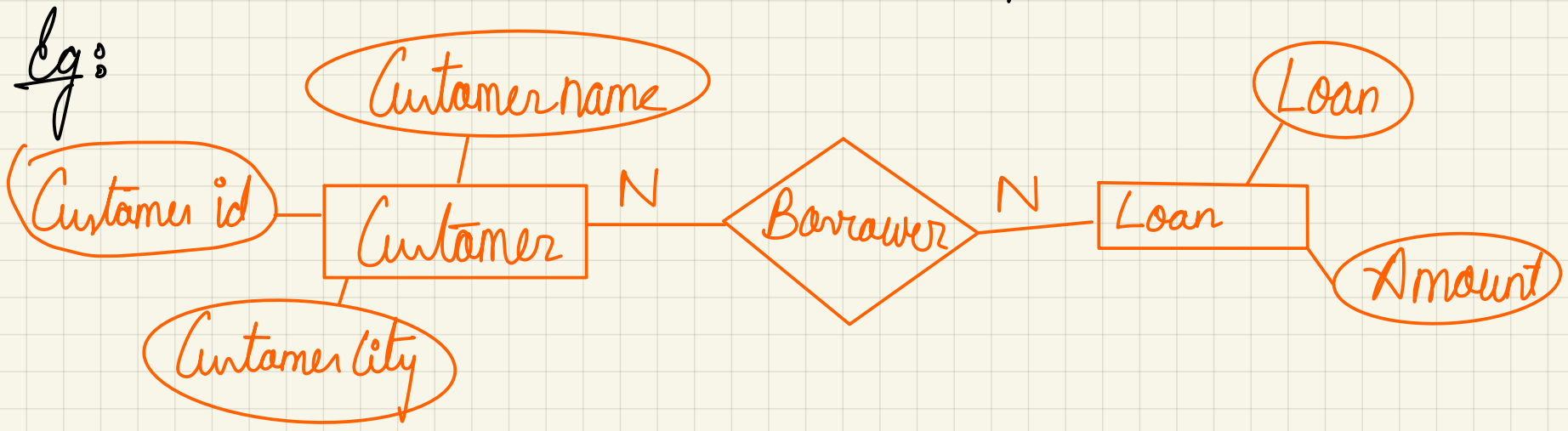
Lines → Links attributes to entity set and entity set to relationship set.

→ Represents multivalue attributes

→ Represents derived attributes

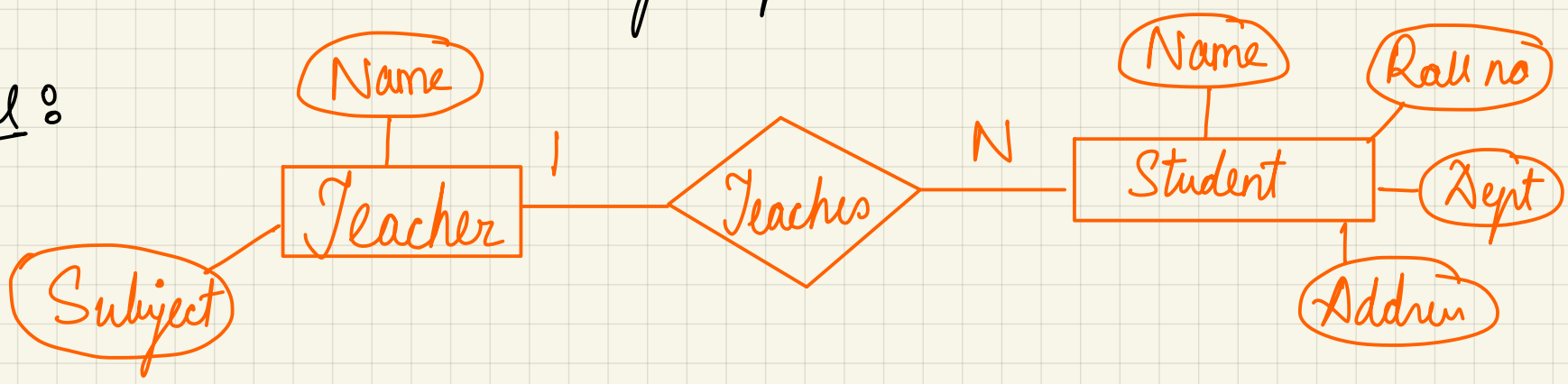
→ Represents total participation of an entity in a relationship set.

Eg:



Q. Construct an E-R diagram for teacher student and their relationships.

Ans:



Entity set : Teacher , Student

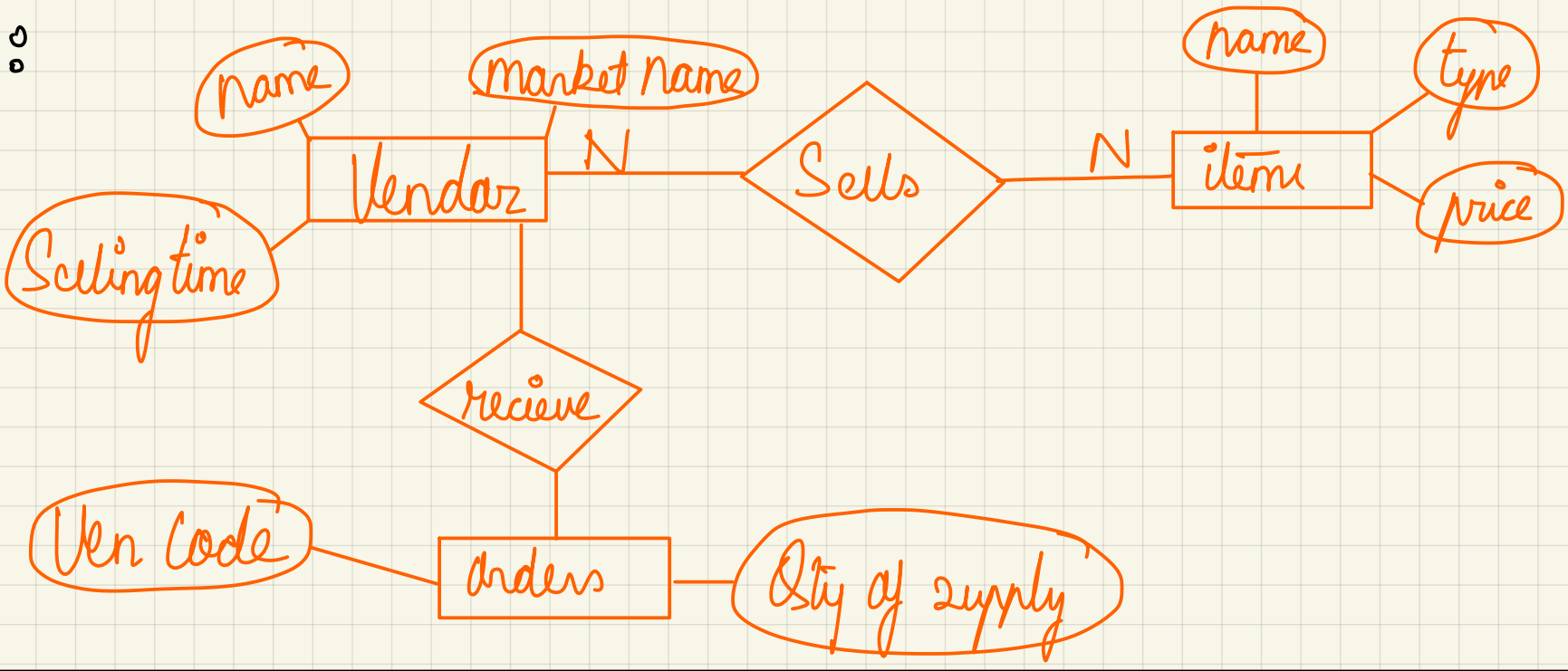
Teacher { Name, Subject }

Student {Name, Roll, dept, addrss }

Relationship : teaches

Q. Construct an E-R diagram for vendors, items and their relationship.

Ans:



31/01/2023

Consider the following set of requirements for a book club. Draw the E-R diagram.

i) The Book club has members.

ii) The Book club sells books to each member.

iii) The members place order for book which the book club fulfilled

iv) Each order contain one or more than one books

v) The Books are written by Authors

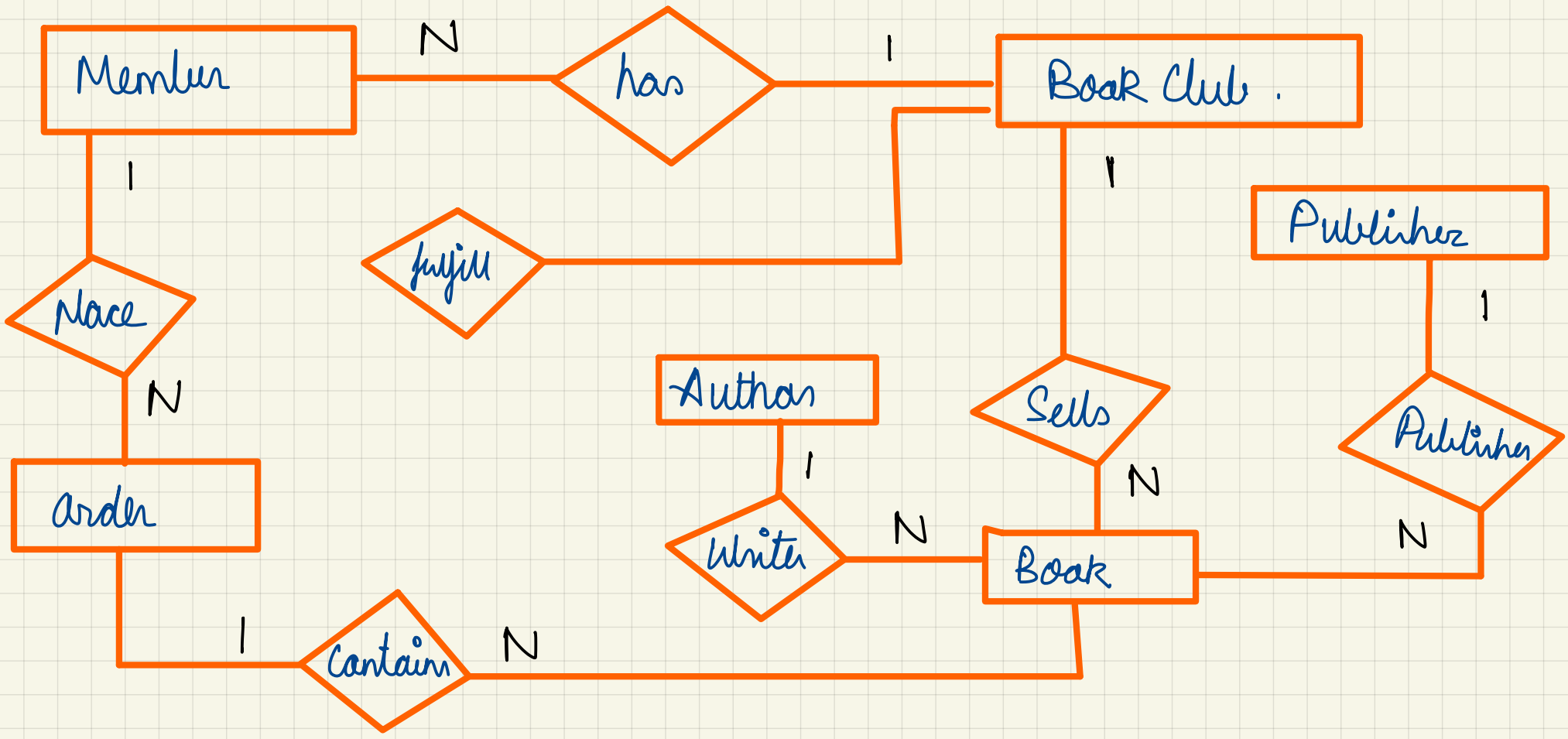
vi) The publisher publish the books.

vii) An Author can write more than one books. and a book can have many authors.

viii) The book is published by a publisher but a publisher publish many books

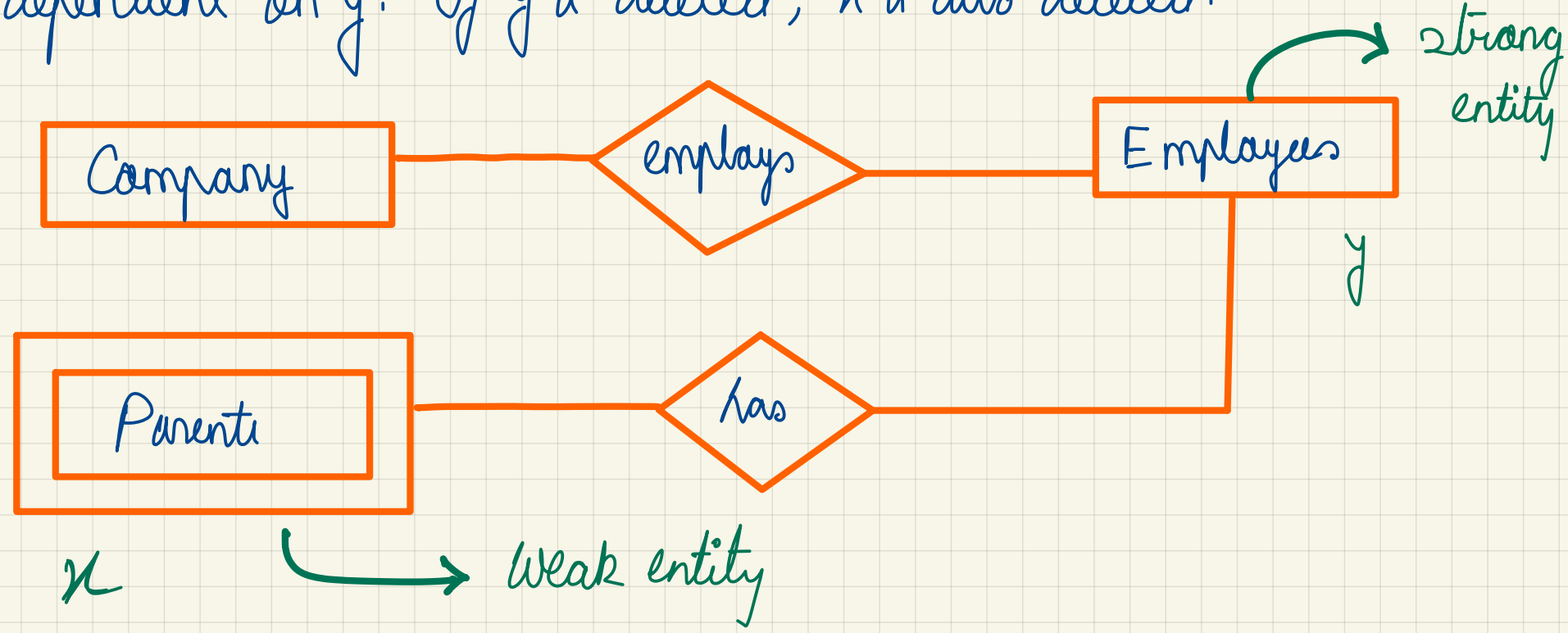
ix) A member can place more than one order and he also can choose not to place an order.

x) The Book club sells many books



- **Strong & Weak Entity:**

The entity are classified as weak entity or existence dependent and strong entity. Existence dependent means if the existence of entity x depends on the entity y. Then x is said to be existence dependent on y. If y is deleted, x is also deleted.



06/Feb/2023

Relational Algebra :-

The relational algebra is a procedural query language, it consists of a set of operation that take one or two relation as input and produce a new relation as their result.

The fundamental operations of relational algebra are :-

Unary	Binary
(i) Select Operation	(iii) Join Operation
(ii) Project Operation	(iv) Division Operation

Set of Operation :

(i) Union	(iii) Cartesian Product
(ii) Intersection	(iv) Difference

Select Operation : This operation is used to select a subset of the tuples from a relation that satisfies a selection condition (predicate) is denoted by σ. The selected operation is represented as follows.

$$\sigma < \text{Selection Condition} > (R)$$

Consider a table called book with the following attributes →

- Book id, Title, Author, Publisher, Year, Price

Query 1 : Select the tuple for all books where publishing year is 2000

$$\sigma \text{ year} = 2000 \text{ (Book)}$$

Query 2: Select the tuple for all books whose price are greater than ₹100.

σ price > 100 (Rupees)

- **Project Operation:** The select operation select the rows (tuples) that matched the selection condition from a table. The project operation selects certain columns (attribute) from a tuple while discarding others. It is an unary operation and is denoted by 'π'.

π < attribute list > (R)

Query 1: Get the columns title and author from the table Book

π title, author (Book)

Query 2: Get the columns title, author, year, price from Book table

π title, author, year, price (Book)

- **Join Operation:** It is a binary operation and denoted by the symbol '$\bowtie$'. This is used to combine and related tuples from two relation into a single tuple. The join operation is very important for any relation database with more than a single table as it enable us to process more than one table at a time. Let's consider two relation names, employee and salary

Employee → Emp no, name, city & Salary → Emp no, Salary.

Query 1: To get the salary of the employees we need to combine - employee and salary relation.

Emp_Sal ← Employee $\bowtie$ Salary

E.Empno = S.Emp no

Result ← π Name, Salary (Employee, Salary)

13/Feb/2023

i) Inner join ii) Left join iii) Right join iv) Full join or natural join

- **Innerjoin** : The inner join keyword select all rows from both the tables as long as the condition is satisfied. 'This' keyword will create the result set by combining all rows from both the tables. where the condition satisfies ie value of the common field will be the same.

Eg. Customers

Id	Name	Age	Addres	Salary
1.	Ramesh	32	Kolkata	2000
2.	Khilan	25.	Delhi	1500
3.	Kaushik	23	Kota	3000
4.	Chaitali	25.	MP	4000
5.	Hardik	27	Bhopal	3500
6.	Komal	22.	Indore	4500
7	Muffy	24	Bihar	10,000

Orders

Oid	Date	Customer id	Amount
102	10/8/23	3	3000
100	10/8/23	3	1500
101	20/11/23	2	1560
103	20/11/23	4	2060

```
Select id, Name, Amount, date
From Customers
Inner join Orders
On customers.id = Orders.customer_id
```

Output :

id	Name	Amount	Date
3	Kaushik	3000	10/8/23
3	Kaushik	1500	10/8/23
2	Khilan	1560	20/11/23
4	Chaitali	2060	20/11/23

- **Left join**: This join returns all the rows of the table on the left side of the joint and matches rows for the table on the right side of the joint. For the rows for which there is no matching row on the right side, the result set will contain null. Left join is also known as left outer join.

```
Select Student name, Student Course, Course id
from Student
left join Student Course
On Student Course.Roll no = Student.Roll_no,
```

- **Right join**: It is similar to left join, this join returns all the rows of the table on the right side of the joint and matching rows for the table on the left side of the joint. It is also known as Right Outer join.

```
Select Student name, Student Course, Course id
from Student
Right join Student Course
On Student Course.Roll no = Student.Roll_no,
```

Student

Roll no	Name	add	phone	age
1	Harsh	Delhi	xxx	18
2	Pratik	Bihar	xxx	19
3	Rianka	Siliguri	xxx	20
4	Deep	Ramnagar	xxx	18
5	Saptarshi	Kolkata	xxx	19
6	Dhanraj	Barabazar	xxx	20
7	Rohit	Balurghat	xxx	18
8	Neeraj	Nijur	xxx	19

Course

Course id	Roll no
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

Left join :

Name	Course id
Harsh	1
Pratik	2
Rianka	2
Deep	3
Saptarshi	1
Dhanraj	Null
Rohit	Null
Neeraj	Null

Right join =>

Name	Course id
Harsh	1
Pratik	2
Rianka	2
Deep	3
Saptarshi	1
Null	4
Null	5
Null	4

- **Cartesian Product** :-

Relation (C × O)

a	m
b	n
c	o

Relation O

u
v

The relation C has two columns and three tuples and relation O has one column and two tuples. So after performing the cartesian product the resultant table name 2+1=3 column and 3×2=6 tuple

a	m	u
b	n	u
c	o	u
a	m	v
b	n	u
c	o	v

- **Division Operation** :-

Branch

Branch Name	Branch City
Florida	USA
Bwrtan	Rye
Southtown	Rye
Brightan	Brooklyn

Branch City

Customer name	A/c no
William	X-204
Catherine	X-111
John	X-105
Pamela	X-228
Brandon	X-145
John	X-200

Account

Branch Name	A/c no	Balance
Florida	X-204	10,000
Burrton	X-111	15,000
South town	X-105	8,000
Florida	X-228	22,500
Brigton	X-145	6400
Burrton	X-200	10,000

- **Query 1** : We want to find all customers who have an account at all the branches located in Rye, we can obtain all the branches in Rye by the following expression.

$$r_1 \leftarrow \pi_{\text{Branch Name}} (\sigma_{\text{Branch city} = \text{Rye}} (\text{Branch}))$$

Now we can find all the customer name Branch name pairs for which the customer has an account at a branch.

$$r_2 \leftarrow \pi_{\text{customer name, Branch name}} (\text{depait} \bowtie \text{account})$$

$$\text{Acc. no} = \text{Acc no.}$$

$$\text{Result} \leftarrow r_2 \div r_1$$

- **Relational Calculus** :-

There are two types of calculus —

(I) Tuple relational Calculus (II) Domain relational calculus.

In relational calculus a query is expressed as a formula consisting of number of variables and an expression involving this variable.

In tuple relational calculus describes the desired information without giving a specific procedure for obtaining that information. The tuple relational calculus is based on specify a no of tuple variables $\{t / \text{cond}(t)\}$, where t is the tuple variable and cond(t) is a condition part and predicate part involving t.

Eg: Retrieve all the books whare price is > 200.

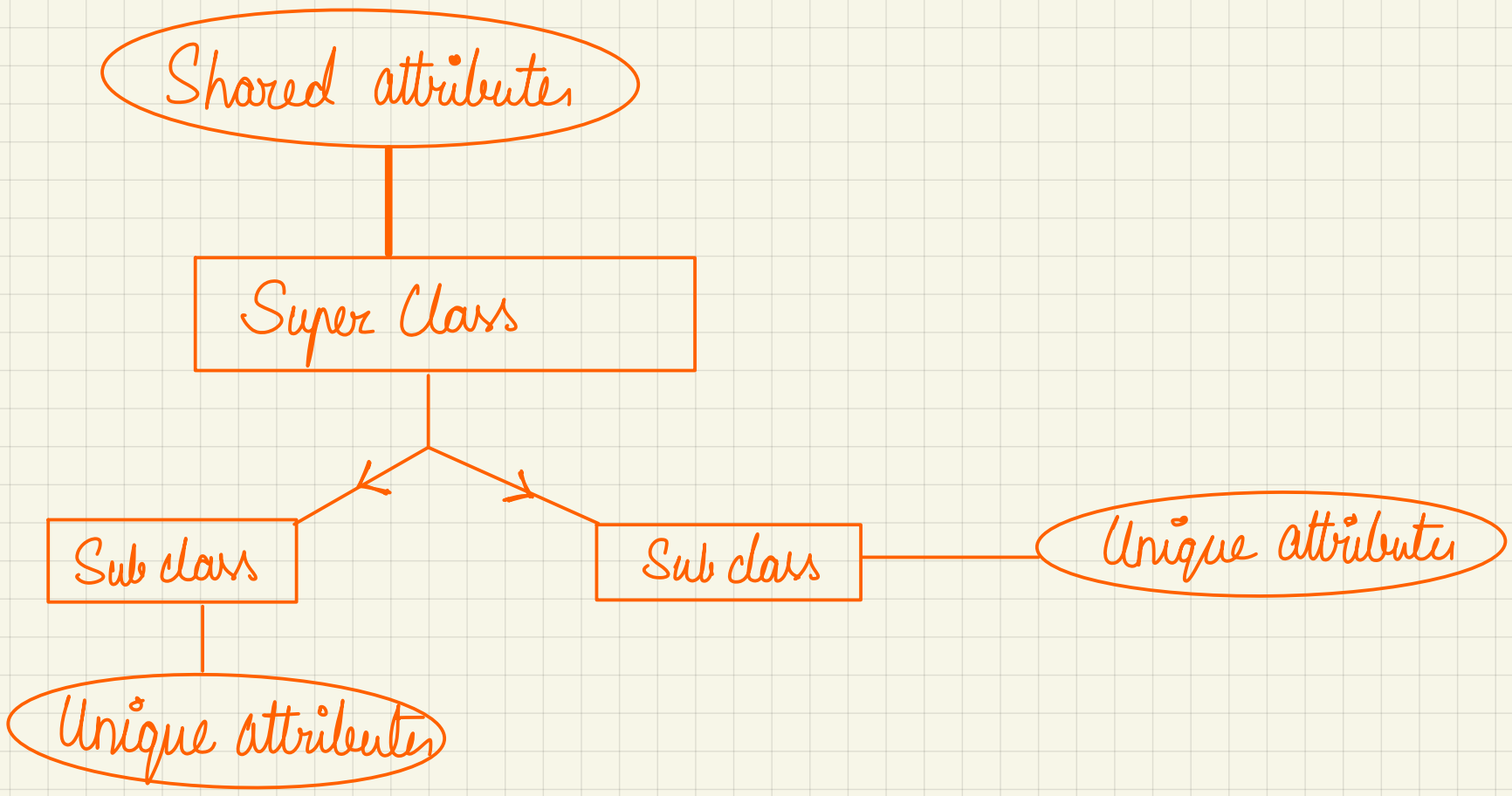
{ t / Book(t) and t.price > 200 }

- Query 2 : Retrieve the details of all books (title, author, name) which were published by Dell books and whare price > 200.

{ t.title, t.author / Book(t) and t.price > 200 }

20/Feb/2023

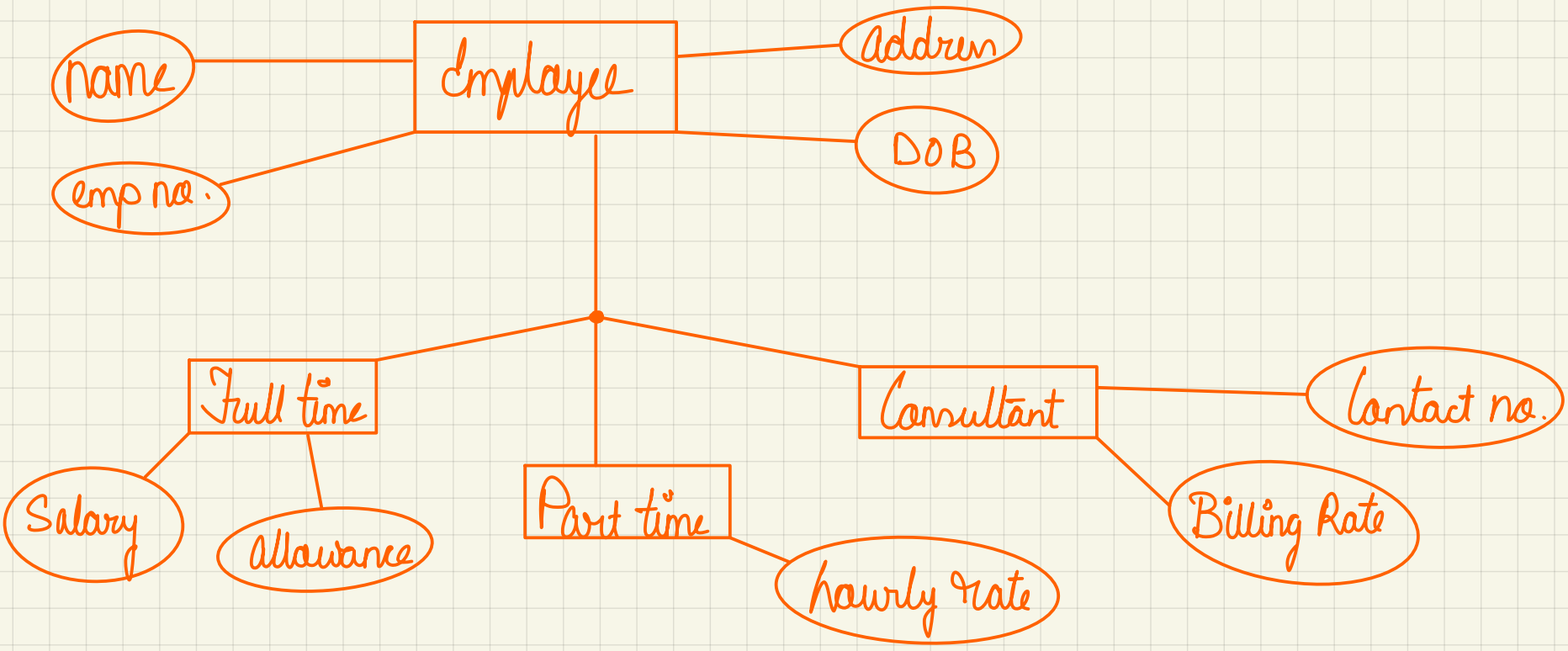
Subclass and Superclass Entity Types :



A superclass is an entity type that include distinct subclasses that required to be represented in a data model.

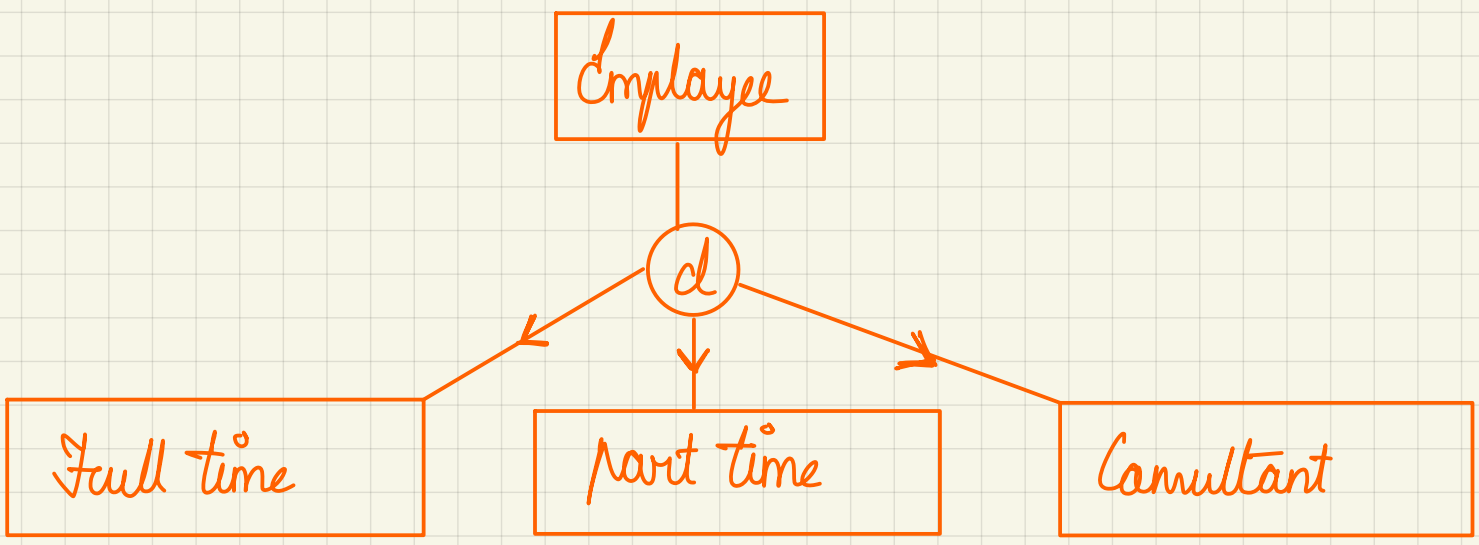
An entity may have distinct subclasses in some cases. The employee entity can have subclasses such as, full time, part time, consultant.

In a superclass subclass relationship the superclass is connected to a circle with a line. The subclasses are also connected to the circle by a line. The U shaped symbol indicate that the sub class is a subset of superclass.

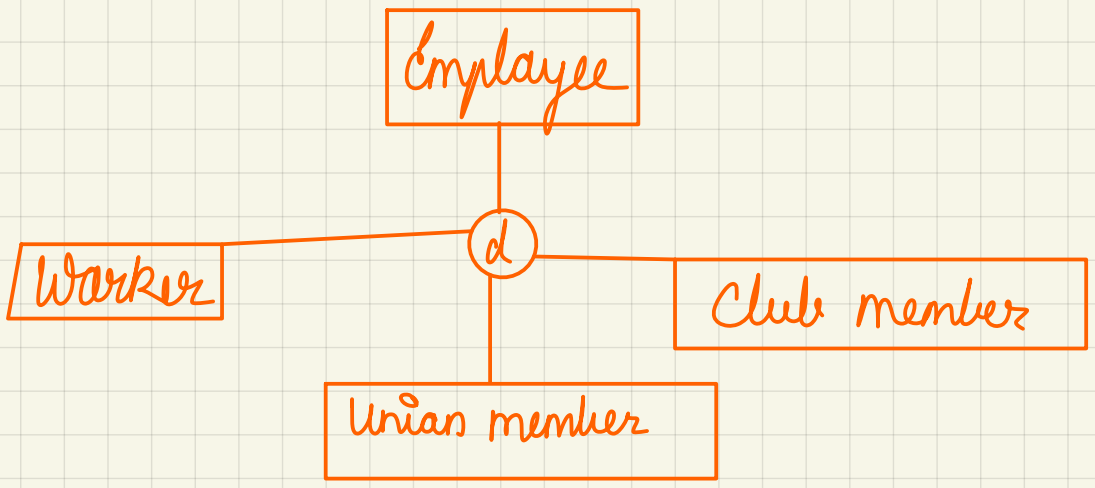


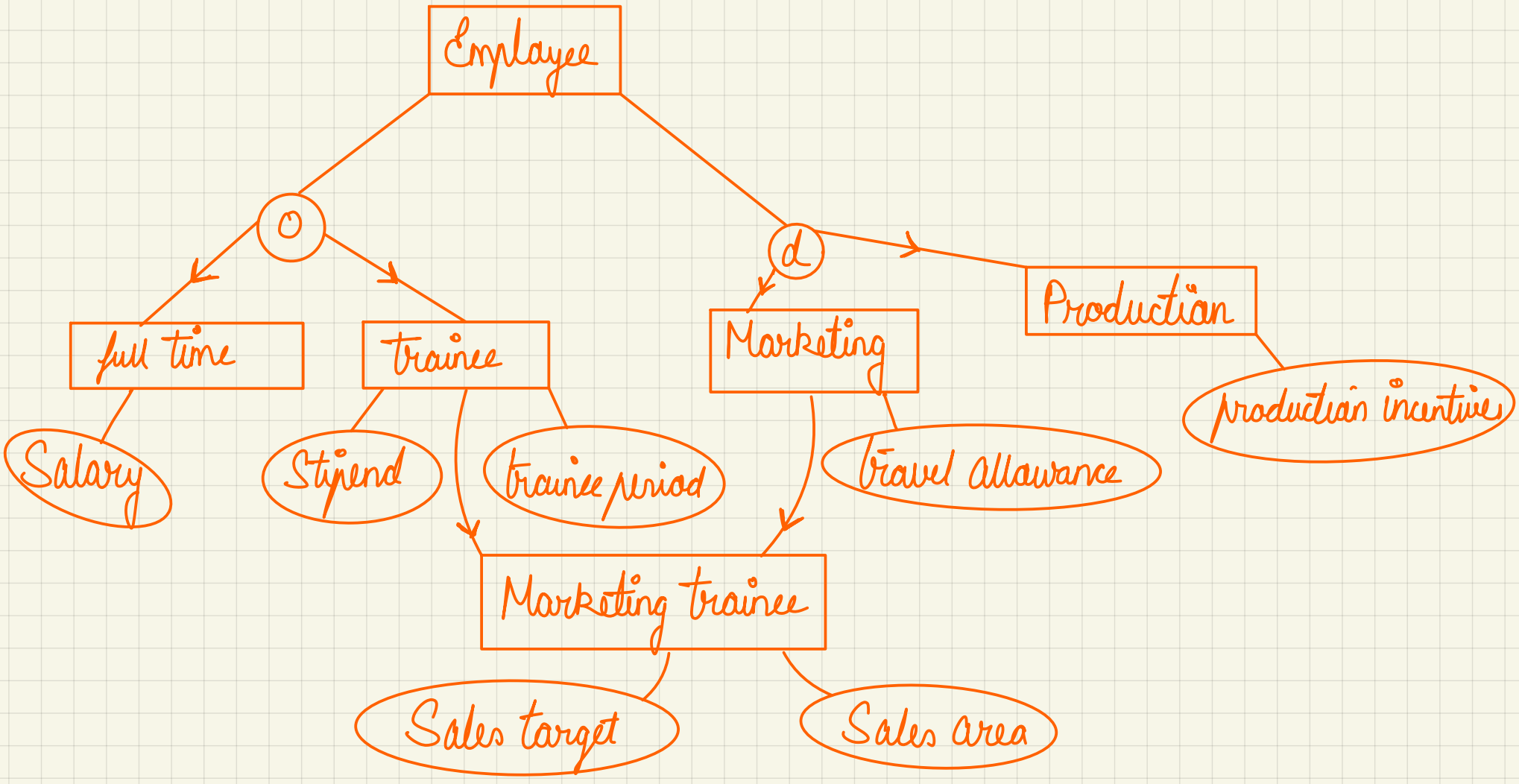
- **Specialization** : The process of designing sub grouping with an entity set is specialization we can also say that it is an process of maximizing the differences b/w members of an entity by identifying the distinguishing and unique characteristics of each member. There are two types of constraints :
  - Disjoint Constraints
  - Overlaping constraint

- **Disjoint Constraint** : The disjoint constraint specifies that if the subclass of a specialization / generalization are disjoin then the entity can be a member of atmost subclasses.



- **Overlapping Constraint** : In overlapping specialization the same entity can be a member of more than one subclasses.




Employee
o
d
full time
trainee
Marketing
Production
Salary
Stipend
trainee period
travel allowance
production incentives
Marketing trainee
Sales target
Sales area

21/Feb/2023

- **Normalization** :-

The most important logical criteria in Data base design is redundancy database consistency. Then, goal is to choose a relational design is to choose a relation that remain consistent and having minimum number of redundancy Such a relation is said to be normal form. It may so happened that even if a relation is normalized it may still posses certain undesirable properties.

- **Functional dependancy** : Given a relation R, the attribute Y of R is functionally dependent on attribute X of R, if each X value in R has associated with it precisely with one Y value in R.

$X \rightarrow Y$ (Y is functionally dependent on X)

If $t_1[X] = t_2[Y]$, then $t_1[Y] = t_2[X]$

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_3	c_2	d_3
a_3	b_3	c_2	d_4

Is $A \rightarrow C$ (Yes)

but $C \not\rightarrow A$

There are 3 types of normal form : (i) First Normal Form (1NF)
(ii) Second Normal Form (2NF)
(iii) Third Normal form (3NF)

1NF :-

A relation R, using 1NF if and only if all the underline domain contain atomic value only.

Department

D. name	D. number	D. mgrssn	D. location
Research	3	241527	Stafford
Administration	2	809137	Mianus
Headquater	1	512434	Rye.

13/March/2023

2NF (Second normal form):-

A relation R is in 2NF if it is in 1NF and every non Key attribute is fully functionally dependent on primary Key.

EMP-PROJ

SSN	P. Number	Hours	E. Name	P. name	P. location

FD1

FD2

FD3

The solution of 2NF is decomposition of a table.

EP3

P. number	P. name	P. location

EP1

SSN	P. Number	Hours

EP2

SSN	E. Name

3NF:- (Third Normal form)

A relation R is in 3NF if it is in 2NF and every non Key attribute is transitively dependent on the primary Key.

EMP - DEPT

E.name	SSN	B.date	Addres	D.number	D.name	DMGRSSN

FD1

FD2

The solution is decomposition of table. We divided the two tables into ED1 and ED2.

Consider the relation for published books (Title, Authar name, Book type, Book, list price, Authar affilation, Publisher)

Book title → Publisher, Book type

Book type → list price

Authar name → Author affilation

Q. What normal form the relation is in ?

Book

Book title	authar name	BT	LP	authar aff.	Publisher

FD1

FD2

FD3

14/March/2023

Q. Car Sale (Car, Saleman, Data of sold, commision, dicount, amount)

Data of sold → dicount, amount

Saleman → Commision

Is the relation R in 1NF, 2NF, 3NF ??

Car Sale

INF

Car	Salesman	Date of sold	Commission	discount	Amount

FD1

FD 2

Since every non Key attribute is not dependent on the primary Key. Therefore it will be INF.

Let R = (A, B, C, D)

F set of functional dependencies.

$A \to B$

$A \to C$

$BC \to D$

Prove that $A \to D$

Solution:

Applying union rule

$A \to B$, $A \to C$ $\to$ $A \to BC$

$BC \to D$ $\therefore$ $A \to D$ [Transitivity Proved]

Multi value dependency :-

CT

Course	Teacher
AI	KND
DBMS	SKD
DBMS	SC
DBMS	NC

CX

Course	Text
AI	Nihon
AI	Linger
AI	Brajno
DBMS	Karth
DBMS	Navathe.

The multivalue dependency is a generalization of functional dependency.

Although the given quotes doesn't have a single corresponding feature that is the Teacher is not functional dependency on course. Each course does have a well defined set of corresponding teachers. This is called multi value dependency. So we can say the Teacher is multi value dependency on course and similarly the text is multi value dependency on course.

Functional dependency is a special case of Multi value dependency and hence Multi Value dependency is the generalization of Functional dependency.

4NF (Fourth Normal form) :-

A normalized relation R is said to be in 4NF if and only if whenever there exists a multi value dependency in 'R' say on attribute 'B' and on attribute 'A', then all attributes of 'R' also functionally dependent on 'A'.

21/March/2023

Dependency Preservation :-

Let 'A' be a set of functional dependency on a schema 'R' and let R_1, R_2 & R_n be a decomposition of 'R'. The restriction of 'F' to 'RI' is the set FI of all functional dependancy in 'F+' that includes only attributes of 'RI'.

R(A, B, C)

$F \rightarrow \{ A \rightarrow B, B \rightarrow C \}$

$F^+ \{ A \rightarrow B, B \rightarrow C, A \rightarrow C \}$

We have a decomposition,

$R_1(A, C)$ & $R_2(A, B)$

$A \rightarrow C$ (F_1) $A \rightarrow B$ (F_2)

The set of restrictions F_1, F_2 upto F_n is the set of dependencies that can be checked efficiently.

Let, $F' = F_1 \cup F_2 \cup \ldots \ldots F_n$

In general $F' \neq F$, however even if $F' \neq F$, it may be that F^+. If the latter is true then every dependency in F is logically implied by F'. We say that decomposition having the property '$F'^+ = F^+$' is a dependency preserving decomposition.

Boyce Codd Normal Form (BCNF):-

The relational schema are in BCNF with respect to a set 'F' of functional dependency if for all Functional dependency, if 'F^+' of the form $\alpha \rightarrow \beta$, where,

α = subset of R β = subset of R

and atleast one of the following properties holds:-

(i) $\alpha \rightarrow \beta$ is a trivial functional dependency. (ie $\beta \subseteq \alpha$)

(ii) α is a Super Key of R

Comparison of 3NF and BCNF:- (3NF over BCNF!! Why??)

→ **Advantage:** There is an advantage of 3NF, in that we know that it is always possible to obtain a 3NF design without sacrificing a lossless join or dependency preservation.

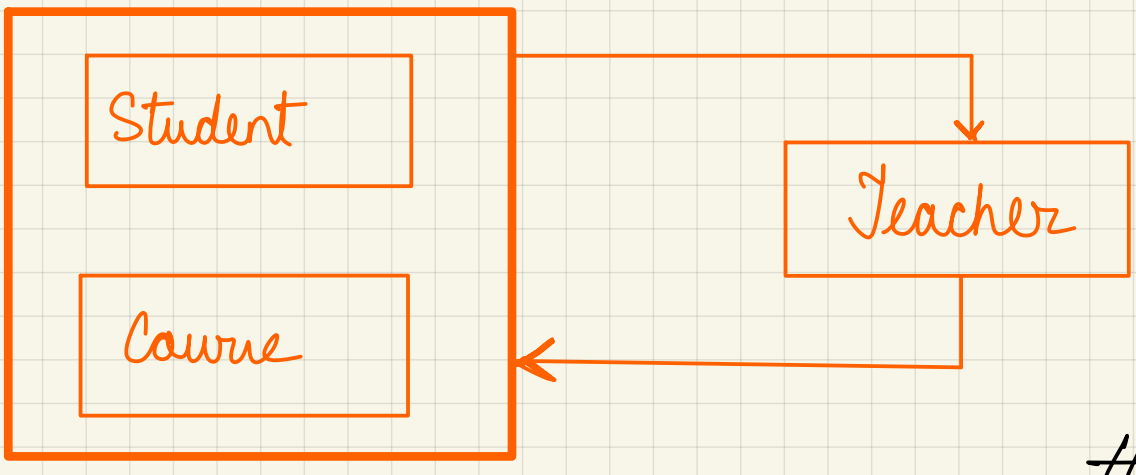
→ **Disadvantage:**

If we dont eliminate all transitive dependency we may have to use null values to represent some of the possible meaningful relationships among data items, and there is the problem of repeatation and information. To choose b/w 3NF and BCNF, it is preferable to go for 3NF.

Example of a relational schema ie in 3NF but not in BCNF,

TEACH

Student	Course	teacher
Smith	Math	White
Smith	Physics	Green.
Jones	Math	White
Jones	Physics	Brown.



Functional dependency in the relation Teach.

#(Relation) → name of the table

Lossless Join Decomposition :-

When decompose a relation into a no. of smaller relation, it is important that the decomposition is lossless let R be a relational schema and 'F' be a set of functional dependency.

Let R_1 and R_2 form a decomposition of R. There decomposition is a lossless decomposition of R if atleast one of the functional dependencies are in a 'F^+'

$R_1 \cap R_2 \rightarrow R_1$

$R_1 \cap R_2 \rightarrow R_2$

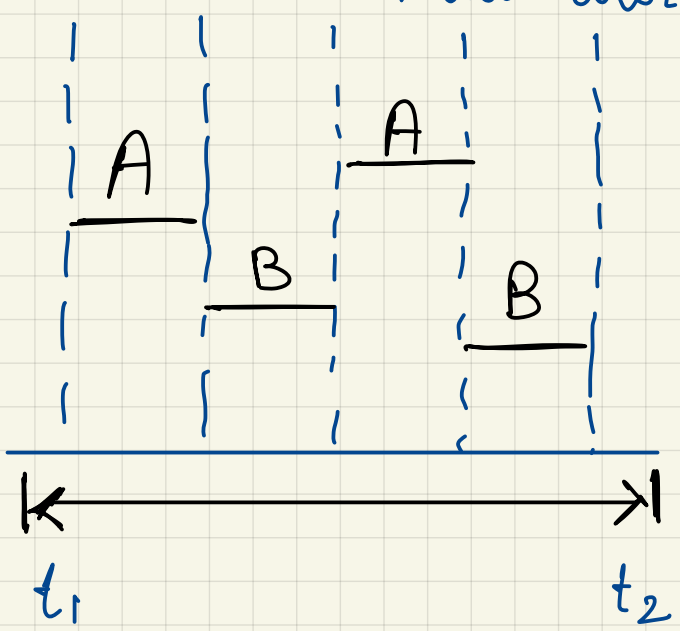
REMEMBER

A decomposition which is not lossless is called a 'Lossy decomposition

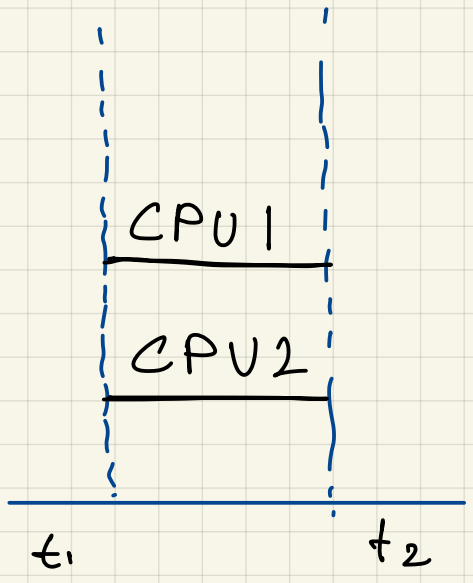
27/March/2023

Transaction :-

DBMS < Single User
DBMS < Multi user



Interleaved processing



Parallel processing

Definition: A transaction is a logical unit of database processing. It can regarded as a series of reads & writes of database update.

- **Database access operation**:-

Read (X): Reads a database item named 'X' into a program variable which is also named as X.

Write (X): Writes the value of program variable X into the database item named X.

Eg: Consider Ti as transaction that transfer ₹ 500 from account X to account Y.

Ti : read (x)
X = X − 500
Write (X)
read (Y)
Y = Y + 500
Write (Y)

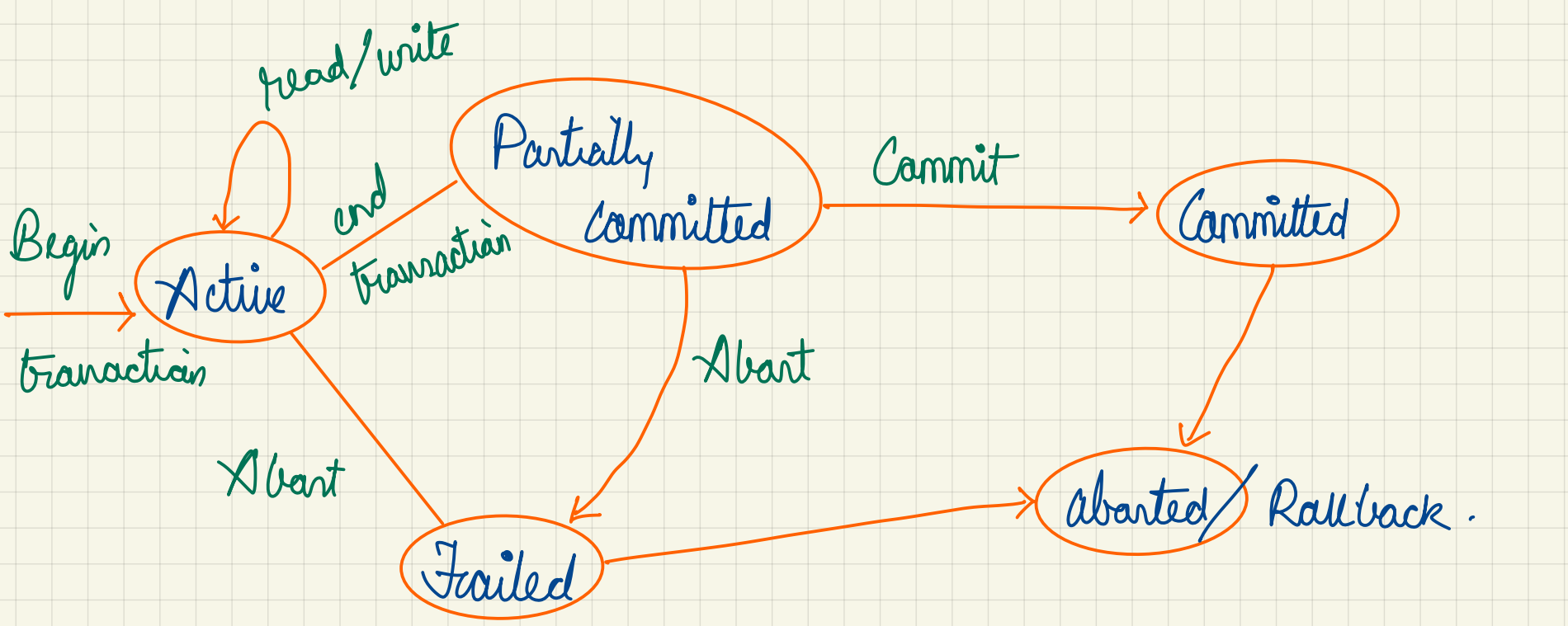
} Steps of transaction.

ACID Properties of transaction :-

A → Atomicity
C → Consistency
I → Isolation
D → Durability

- **Atomicity** : A transaction is an atomic unit of processing. It's either performs entirely or not performed at all.
- **Consistency** : The correct execution of the transaction must take the data base from one consistent state to another.
- **Isolation** : A transaction should not make it's update visible to other transaction, until it is commited. The commit stage is another stage of transaction.
- **Durability** : Once a transaction changes a database and changes are commited, these changes are must never be lost, because of subsequent failure.

State Diagram of Transaction :-



Begin transaction marks the beginning of the transaction execution

Read/write specifies the read/write operation.

End Transaction specifies that read/write operation has ended and marks the end of the transaction execution.

Commit marks successful end of the transaction, so that any changes (update) executed by the transaction can be safely committed.

Aborted / Rollback marks that the transaction has ended unsuccessfully.

A transaction must go to active state immediately after it's start execution.

When the transaction ends, it moves to partial committed state.

28/March/2023

Lost Update Problem :-

T_1

Read (x) - 80

X = X - N - 75

Write (x) - 75

read (y)

y = y + N

Write (y)

T_2

read (x) - 75

X = X + M - 79

Write (x) - 79

Initially

X = 80

N = 5

M = 4

X = 79

X = 84

T_1	T_2
80 Read (x)	
75 X = X - N	
	80 read (X)
	X = X + M 84
84 Write (x)	
60 read (y)	
65 Y = Y + N	
65 Write (y)	
	Write (X) 84

X = 80

N = 5

M = 4

Y = 60

X = 84

Y = 65.

Failures occurs when two transaction that access the same database item have their operation interleaved in a way that makes the value of some database item incorrect.

- **Types of failures** :-

There are two type of failures :

① Non-Catastrophic failure
- → Computer failure
- → System errors
- → Local errors

② Catastrophic failure
- → Disk failure
- → Physical problem

- **Computer failure** : A hardware, software or network error occurs in a computer system during transaction execution.
- **System error** : Some operation in the transaction may call it to fail such as division by zero.
- **Local errors** : Data of the transaction may not be found.
- **Disc failure** : Some disc blocks may loose their data because of a read or write malfunction or because of a disc read/write head crash.
- **System log** : To recover from failures that affect the transaction the system maintain a log to keep track of all transaction operation that affects the values of database item's. The log is kept on the disc.

– Schedule

17/April/2023

The types of entries in a log are called log records.

[Start transaction, T] → indicates that transaction T has started the execution.

[Read item, T, X] → indicates the transaction T, has read the value of database item X.

[Write item, T, X, old value – new value] → indicates that transaction T, has change the value of database item X from old value to new value.

[Commit, T] → indicates T has ended successfully.

[abort, T] → indicates T has been aborted.

- **Schedule of transaction** :

Schedule Sa : $r_1(x)$; $r_2(x)$; $w_1(x)$; $r_1(y)$; $w_2(x)$; c_2 ; $w_1(y)$; c_1 ;

read item → r
write item → w
commit n → c
abort n → a

- **Lose updation problem** :-

T_1	T_2
read (x)	
X = X – 5	
	read (x)
	X = X + M
write (x)	
	write (x)
read (Y)	← commit
abort	

S_0 : $r_1(x)$; $w_1(x)$; $r_2(x)$; $w_2(x)$; c_2 ; $r_1(y)$; a ; j

- **Schedule** : When transaction are executing concurrently in a interleaved fashion then the orders of execution of operation from the various transaction is known as scheduling.

- **Conflict schedule** : Two operations in a schedule are said to be conflict if they satisfy all three of the following condition :-
  - They belong to different transaction
  - They access same database item X
  - Account one of the operation is write (X) operation.

Sa :

$r_1(X)$
$r_2(X)$
$w_1(X)$
$r_1(Y)$
$w_2(X)$
c_2
$w_1(Y)$
c_1

$r_1(X)$ & $w_2(X)$ conflict
$r_2(X)$ & $w_1(X)$ conflict
$w_1(X)$ & $w_2(X)$ conflict

Sb :

$r_1(X)$
$w_1(X)$
$r_2(X)$
$w_2(X)$
c_2
$r_1(Y)$

$r_1(X)$ & $w_1(X)$ conflict
$w_1(X)$ & $w_2(X)$ conflict
$w_1(X)$ & $r_2(X)$ conflict

- **Recoverability** : Once a transaction T to be committed it should never be neccesary to rollback T. The schedule that meet this criteria 6 called recoverable schedule and thare don't called not recoverable.
- **To check the recoverability** : Is schedule where for each pair of transaction Ti and Tj if Tj reads a data then previously written by Ti then the commit operation of Ti preceeds the commit operation of Tj

$r_1(X)$; $r_2(X)$; $w_1(X)$; $r_1(Y)$; $w_2(X)$; c_2; $w_1(Y)$; c_1

↳ it is non recoverable, if it were,
$r_1(X)$; $w_1(X)$; $r_2(X)$ then it would have been recoverable.

18/April/2023

- **Cascading Rollbacks** :-

In a recoverable schedule no committed transaction ever needs to be rollback It is possible for a phenomenon known as cascading rollback to occur when an uncommitted transaction has to be rollback because it read an item from a transaction that failed.

- **Serializability of schedule** :-

Two transaction T_1 and T_2 are said to be serial if the

① Execute all the operation of transaction T_1 followed by all the operation of transaction T_2.

② Execute all the operation of transaction T_2 followed by all the operation of transaction T_1

(A)

T_1	T_2
read (x)	read (x) 87
X = X − N (90−3) = 87	X = X + M (87+2) = 89
write (x)	write (x)
read (y)	
Y = Y + N → 93	
write (y)	

(B)

T_1	T_2
read (x)	read (x)
X = X − N	X = X + M 92
write (x)	write (x)
read (x)	(92−3) = 89
Y = Y + N	→ 93
write (y)	

- If X = 90, Y = 90 N = 3, M = 2, when will be the value of schedule A & B ? is the two schedule to serial or not.

→ for schedule A, X = 89 , Y = 93

for schedule B, X = 89 , Y = 93

∴ schedule A & B are serial

(C)

T_1	T_2
Read (X)	
Y = X−N → (90−3) = 87	Read (X)
	X = X+M
Write (X) → 92	
	Write (X) → 92
Read (Y)	
Y = Y+N	
Write (Y) → 93	

(D)

T_1	T_2
Read (X)	Read (X)
X = X−N 87	X = X+M
Write (X) 87	Write (X)
Read (Y)	
Y = Y+M 92	
Write (Y) 92	

- If X = 90, Y = 90, N = 3, M = 2, When will be the value of schedule C and D? are they serial or not?

for schedule C X = 92, Y = 93

for schedule D X = 89, Y = 93

∴ Schedule C & D are not serial schedule.

A schedule of N transaction is serializable if it is equivalent to some serial schedule.

- Conflict equivalent :-

Two schedule are said to be conflict equivalent if the order of any two conflicting operation is the same in both schedule.

- Conflict Serializable :-

A schedule is said to be serializable, if it is conflict equivalent with some serial schedule.

For A and B schedule,

(A)

$r_1(X)$
$w_1(X)$
$r_1(Y)$
$w_1(Y)$
$r_2(X)$
$w_2(X)$

(B)

$r_1(X)$
$w_1(X)$
$r_2(X)$
$w_2(X)$
$r_1(Y)$
$w_1(Y)$

- **Testing of Conflict Serializability of a schedule :-**

Algorithm :-

Testing Conflict Serializability of schedule S

① For each transaction T_i in S create a node levelled T_i in the precedence graph.

② For each case in S where T_j executes a read (X) after a write (X) executed by T_i create and edge from $T_i \rightarrow T_j$

③ For each case in S where T_j executes a write (X) after T_i executes a read (X) create edge from $T_i \rightarrow T_j$

T_i	T_j
read (X)	write (X)

④ For each case in S where T_j executes write item (X) create an edge from $T_i \rightarrow T_j$

⑤ The schedule is serializable if and only if the precedence graph has no cyclic