Leeture-1 24.04.17

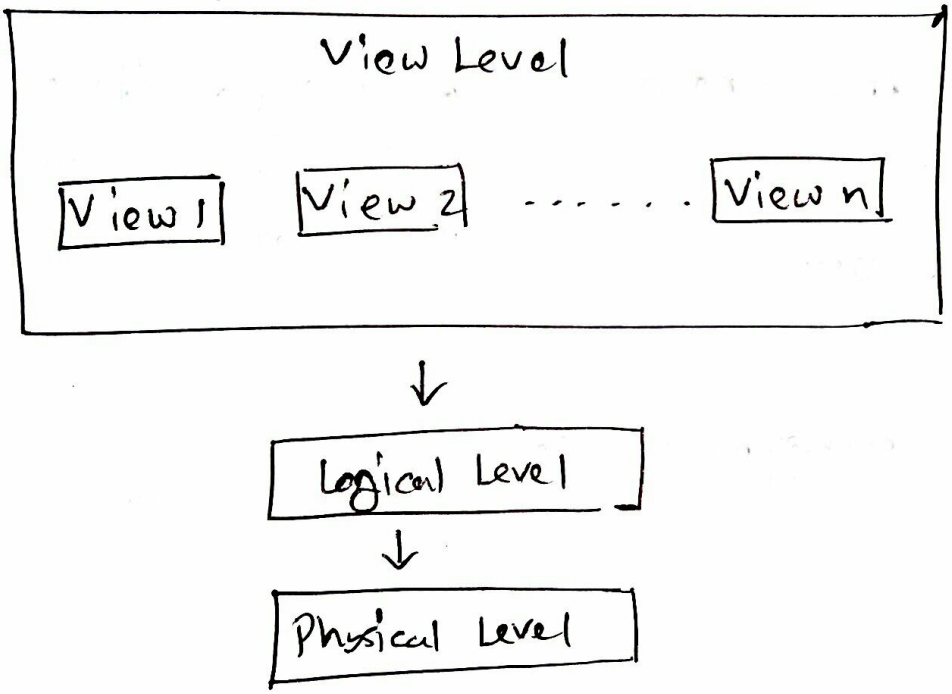
Level of Abstraction

(1). Physical

(2) Logical

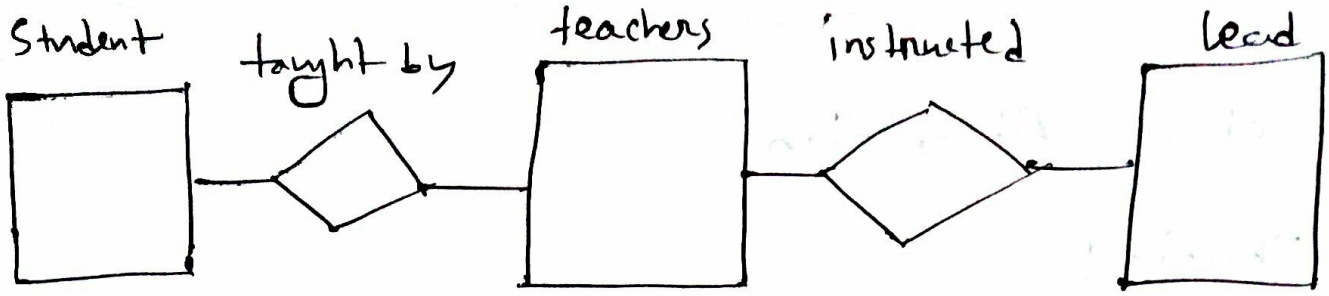
(3) View level

Architeeture of Database System



What are the level of db and defined them

Schema – describe the overall design of the DB



Sub schema – describe the different view of db

Self study

Database User

(1) Application programmer

(2) Sophisticated user

(3) Specialized user

database Administration

Function of DBA

- Storage Structure
- Schema definition
- Authorization

DBMS – DB management system

Lecture-2

তা. 05.07

Site এ slide গুলো

selection → row → σ

projection → column → π

$\sigma_{A=B \wedge D>5}(R)$

R

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

A	B	C	D
α	α	1	7
β	β	23	10

(ঘ) Select row where A=B and D > 5 from relation

$\pi_{(A,C)}(R)$ (ক) Project A, C from R

A	C
α	1
α	5
β	12
β	23

Relation R, S

R ∪ S (union এর Table গুলো)

Leeture - 3 7050 7.05.17

ER Modelling [Chap-6]

Steps : 1) To implement n project you need a IDEA First.

2) From the IDEA we need to create High level Description that is created by ER- modelling

3) In ER-Modelling speeial symbol are used to represent the requirement.

(DDL / DML) ← Two type language

IDEA → High Level Description → Relational Schema (SQL)

↓

RDBMS

⇓

Implementation

ER Diagram that visualize the ER Model with some Specific Symbol, which represent the total view of ER- model

ERD = Entity Relationship Diagram

There are 3 (three) main parts in ER Modelling

1) Entity
2) Relationship
3) Attributes

there are also some other part in ER model

After the requirement, Analysis a Storyline can be found. From there we have to identify the Entity and their relationship.

Entity:

→ Entity is always a noun
→ Entity will be that particular object in which Data will be preserve in DB
→ Each of the Entity is Distinguish from another
→ Entity will be Real world object
→ An entity is described with a set of Attributes

Examples:

People	Place	Organization
Things	Concepts	Events

Entity Set: Collection of similar entities

Entity Type: Collection of Entities that all have similar attributes

Student {ID, name, DOB, phone}

Entity Set

Entity set →

ID	name	DOB	Phone
1	A	01 Jan	019...
2	B	02 Feb	017...

Example:

There are many student in AUST. Student can enroll in different courses and receive grades.

Entity : Student :

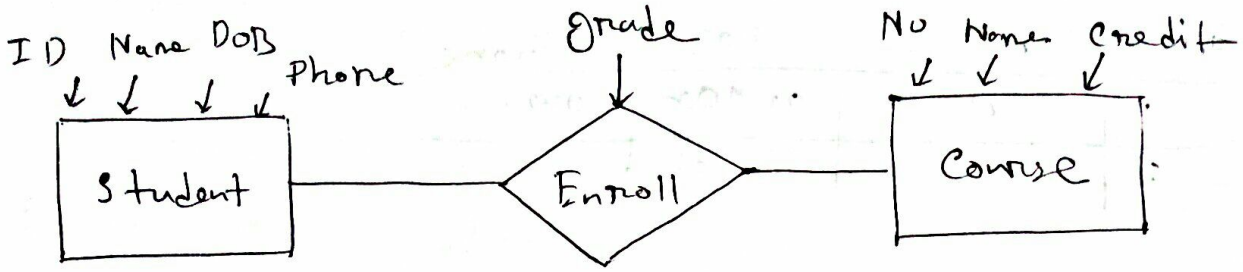
ID , name , DOB , Phone , .

Courses :

No , Name , Credit

Relationship :

- Interaction between a Entity with another Entity
- Is a verb always
- Relationship can have its own attributes
- Association among two or more attribute



Relationship Enroll

Symbols

Entity Set : Rectangle

Relationship : Diamond

Attribute : Ellipse

Attributes → The properties of Entity or Relationship

→ Properties are same for entity but value can change

→ If attributes are described in storyline use those ones, other than guess the properties and use meaningful ones.

Primary Key :

The attributes which can distinguish one Entity from another can be a primary key

- Primary key should be underlined in ERD
- More than one attribute can be primary key
- Primary key will be minimum Attribute
- Primary Key should be choose in used to form

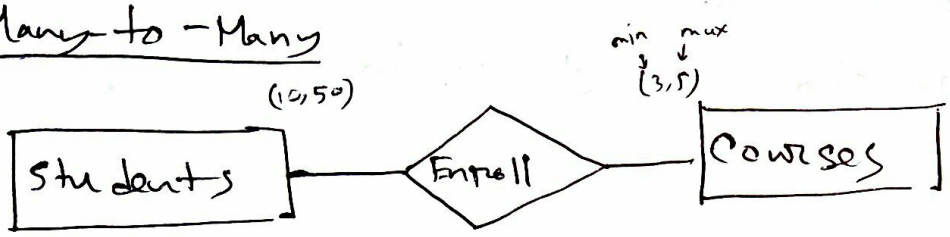
Lecture-4

8.05.17

Cardinality Constraint of relationship Cardinality Ratio:

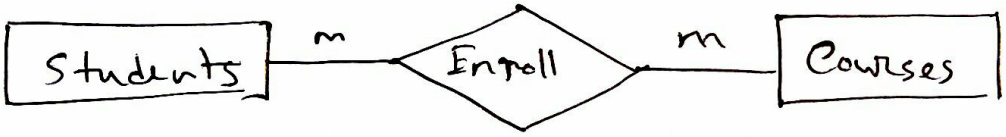
- Express the maximum number of entity can be associated with another entity via a relationship
4 types of Cardinality Constraints

1. Many-to-Many



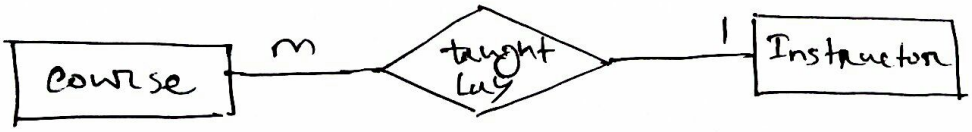
m → many
1 → one

OR



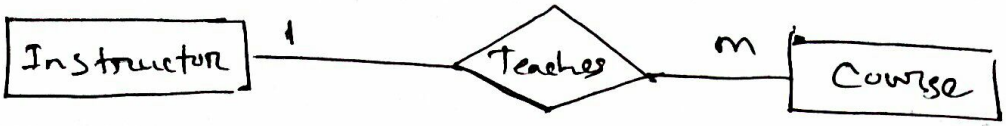
2. Many-to-One

OR

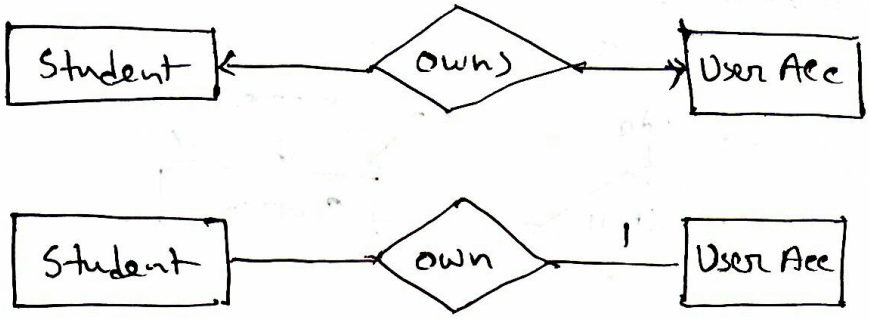


3. One-to-many

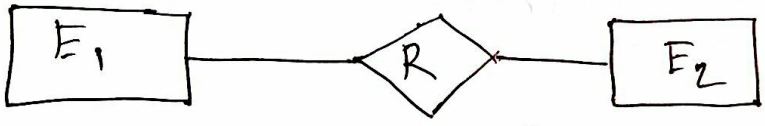
or



4. One-to-One



Cardinality Limit / Degree of Constraints



(min_1, max_1) (min_2, max_2)

Relationship	min_1, max_1	min_2, max_2
Many to many	(0, M)	(0, *)
Many to one	(0, 1)	(0, *)
One to many	(0, *)	(0, 1)
One to one	(0, 1)	(0, 1)

Example: There are many student in AUST. Student can enroll different courses and receive grades.

Noun → duplicate, one/many time, attributes

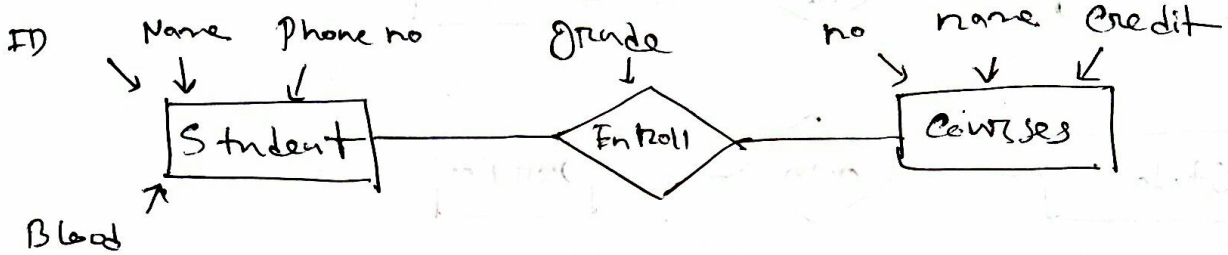
Verb → O

Attributes → declared

guess →

Student : ID, name, phone, Blood Group

Course : No, name, Credit



Quiz - 1

14.05.07

$L_1 \rightarrow (D_1, D_2, D_3)$

1st 3 থেকে slide

Lecture-5

9.05.17

Example-01 : Students take courses and each students belongs to a particular department. Students grade in different courses are stored. Each department has multiple students and a department offers multiple courses as well. A course can be offered by a single department on multiple department.

Entity : student, course, department

Relationship : takes, Belongs, offers

Primary

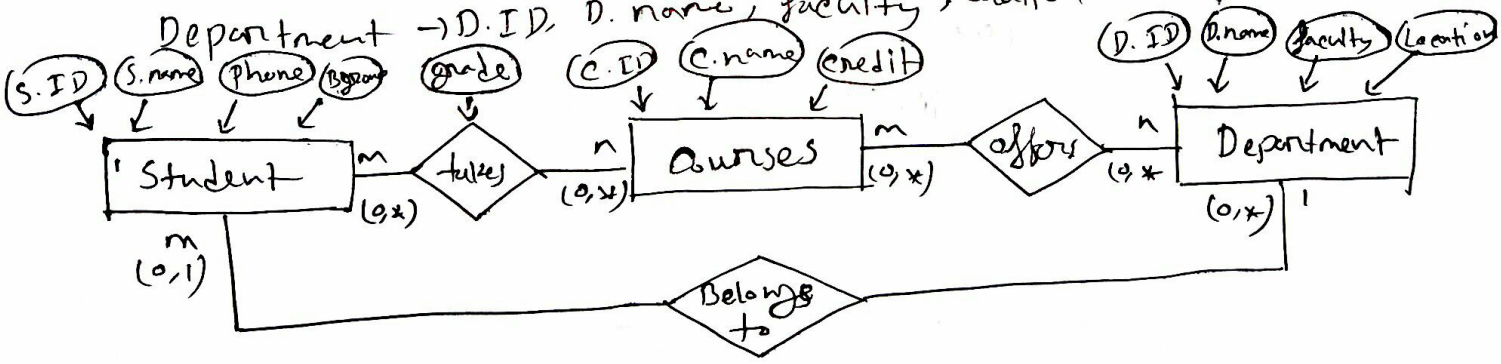
Cardinally Constraints/Ratio

Attributes :

Student → S.ID, S.name, phone, B.group

Course → C.ID, C.name, credit.

Department → D.ID, D.name, faculty, location



Example : Patients are treated in a single ward by the doctors assigned to them. Usually Each patient will be assigned a single doctor, but in rare cases they will have two. Assistants also attend to the patients and a number of these assistants are associated with each patient.

Entity : Patients, ward, doctor, assistant

Relation : treated, assigned to, attend

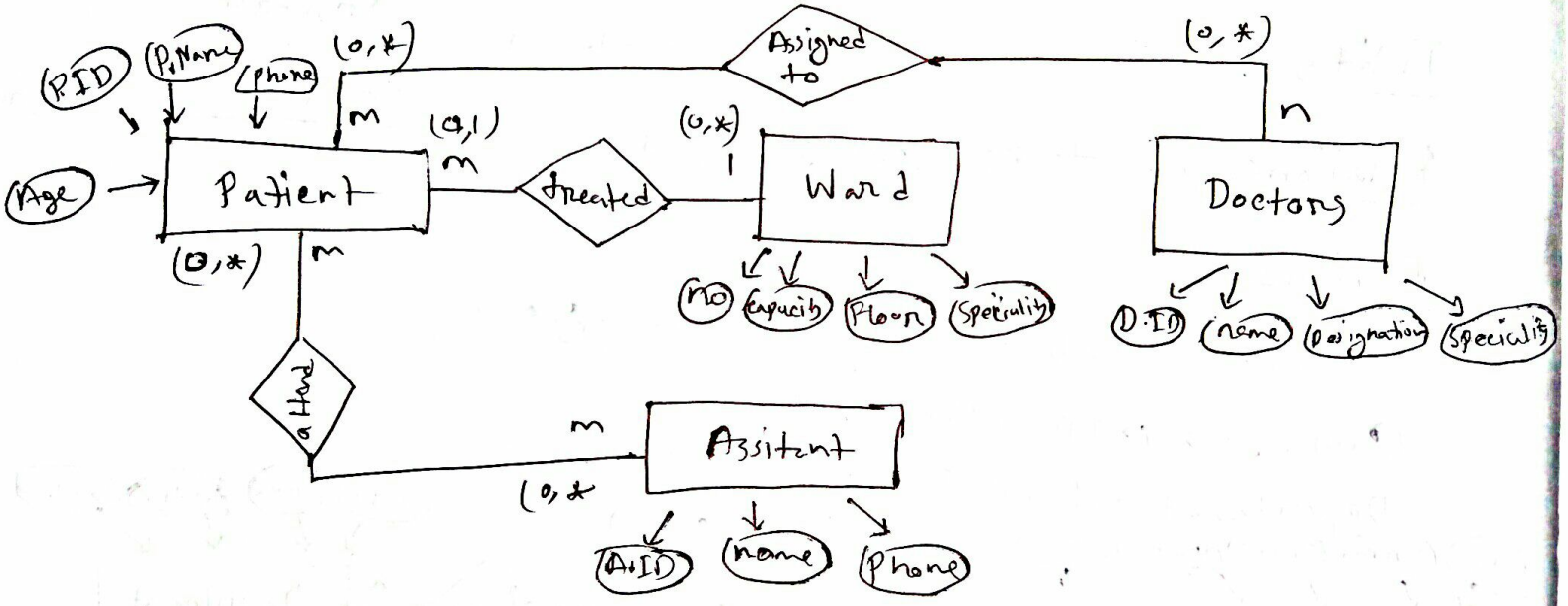
Attributes:

Patient : P.ID, name, phone, Age

Ward : No, Capacity, Floor, Speciality

Doctor : D.ID, name, Designation, Speciality

Assistant : A.ID, name, phone



Quiz-2

Date - 4 June, 2017

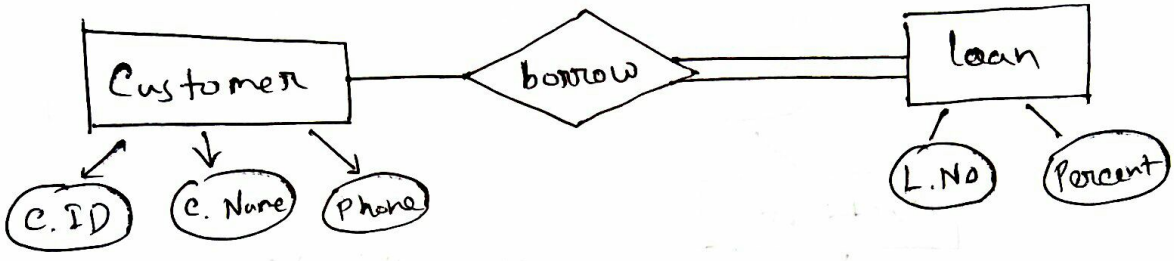
Syllabus - ERD

Lecture-6 16.05.17

Participation Constraints

1) Total Participation ═══ (double line)
2) Partial Participation ——— (single line)

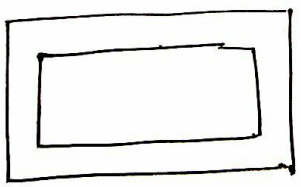
Examples Banking Example



Weak Entity Set: (1 to many ত use হয়)

Weak Entity, Identify Entity, discrimination

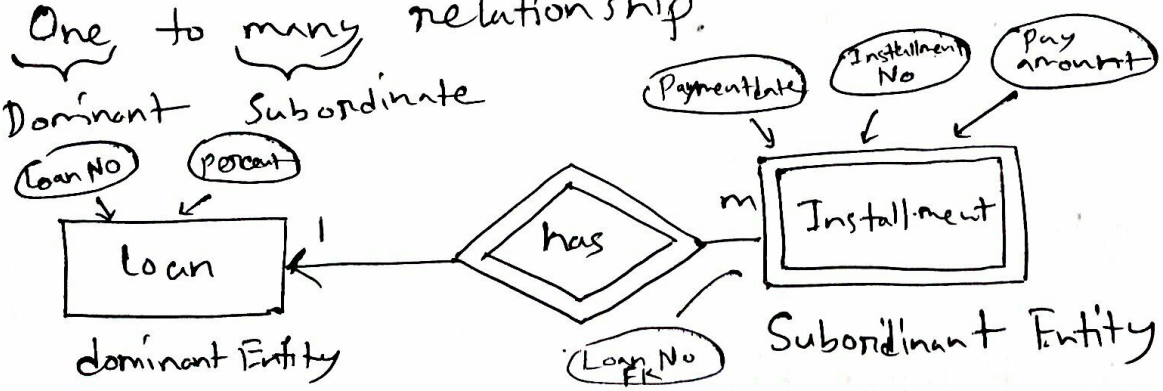
Symbol :



← Double line ~~reg~~rectangle

One to many relationship.

Dominant Subordinate



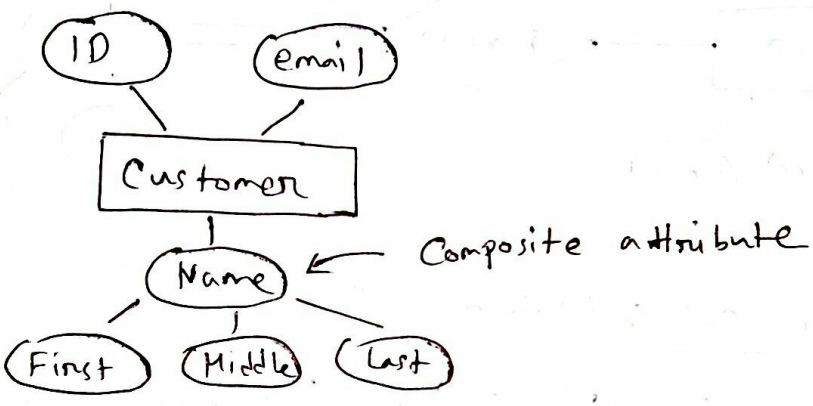
Dominant – যার Table এ Help করে সে
Subordinant entity set – (weak entity set)

No	Payment
123456	50,000

loan	Date	No	Amount
123456	16.5.17	01	10,000

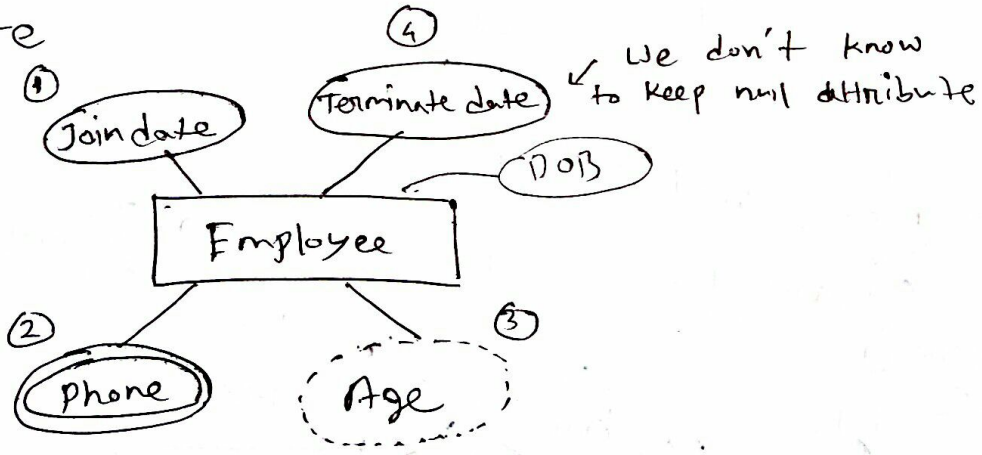
Attribute Type

Single and Composite



Other - (1) (2) (3) (4)

Single Valued , Multivalued , Derived , Null attribute

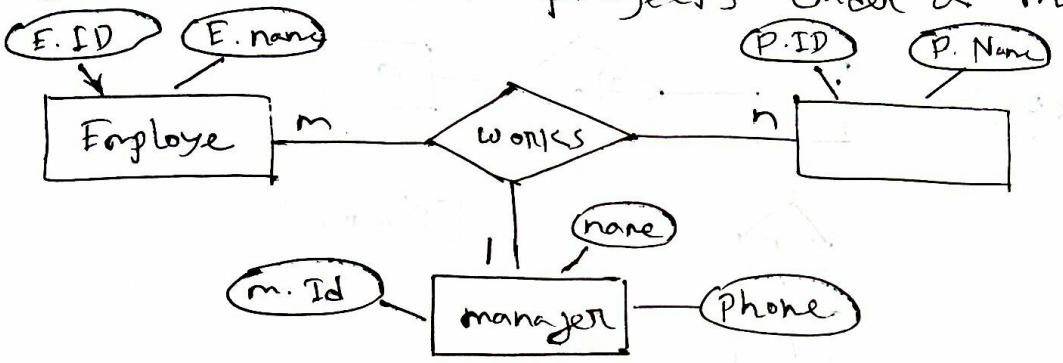


Degree Of Relationship

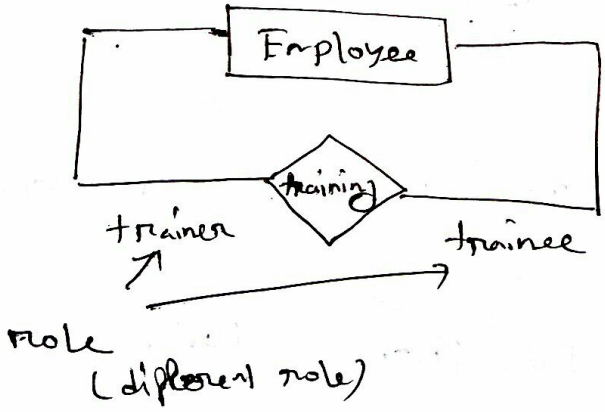
1) Binary

II) ~~Temporary~~ Ternary

Example: A Company has several projects, employes can work several projects under a manager



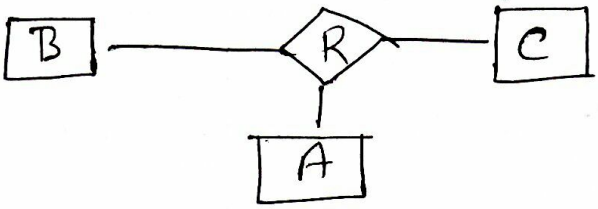
III) Unary / Recursive



IV) N-ary: More than 3 Entities

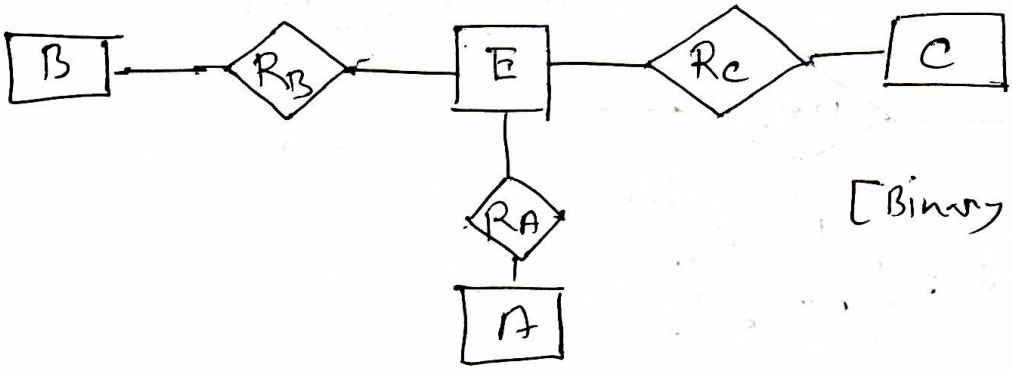
We should try to avoid Ternary and recursive cause implementation is not easy all time

Converting non Binary to Binary



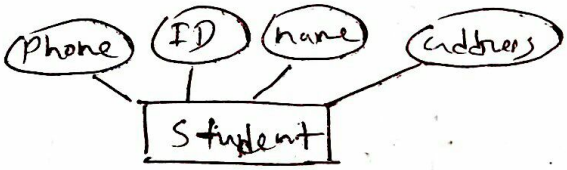
[non Binary]

⇓ Convert



[Binary]

Keys :



I) Super keys:

↓

II) Candidate key : ID, {ID, name}, {ID, phone}

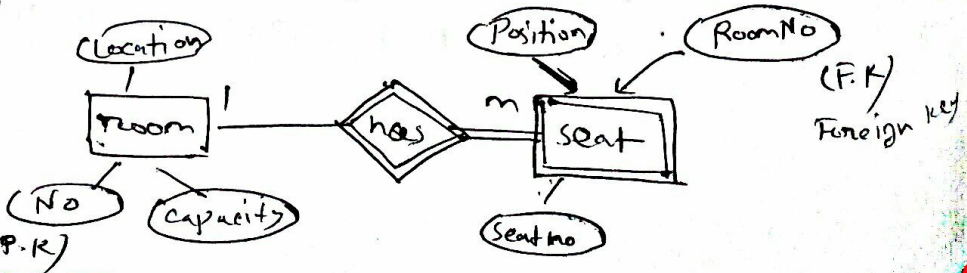
↓

III) Primary key : ID, phone (Unique key)

IV) Alternative key : Primary key ছাড়া বাকি Candidate key

V) Composite key : একাধিক Primary key নিয়ে তৈরি হয়।

VI) Foreign key:



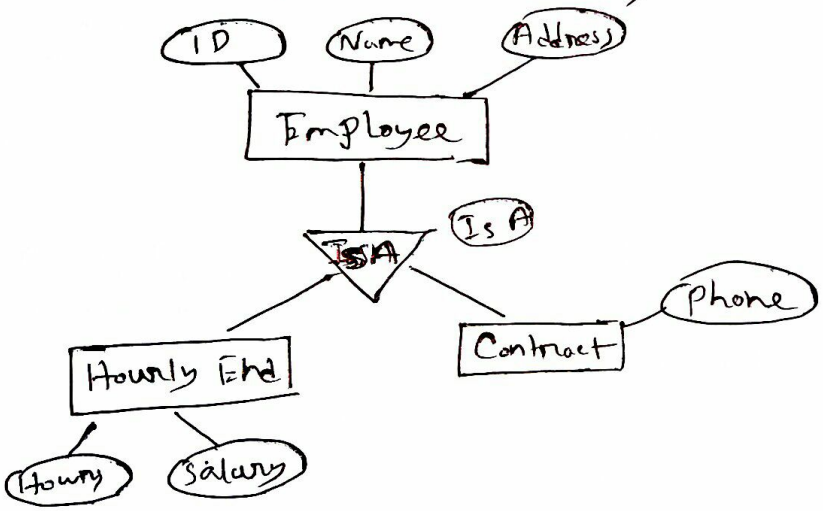
Enhanced ER Modelling

In some cases an entity Type has a number of sub groups. Relationship and attributes of superclass are inherited to subclass. Subclass can have additional attributes and relationship

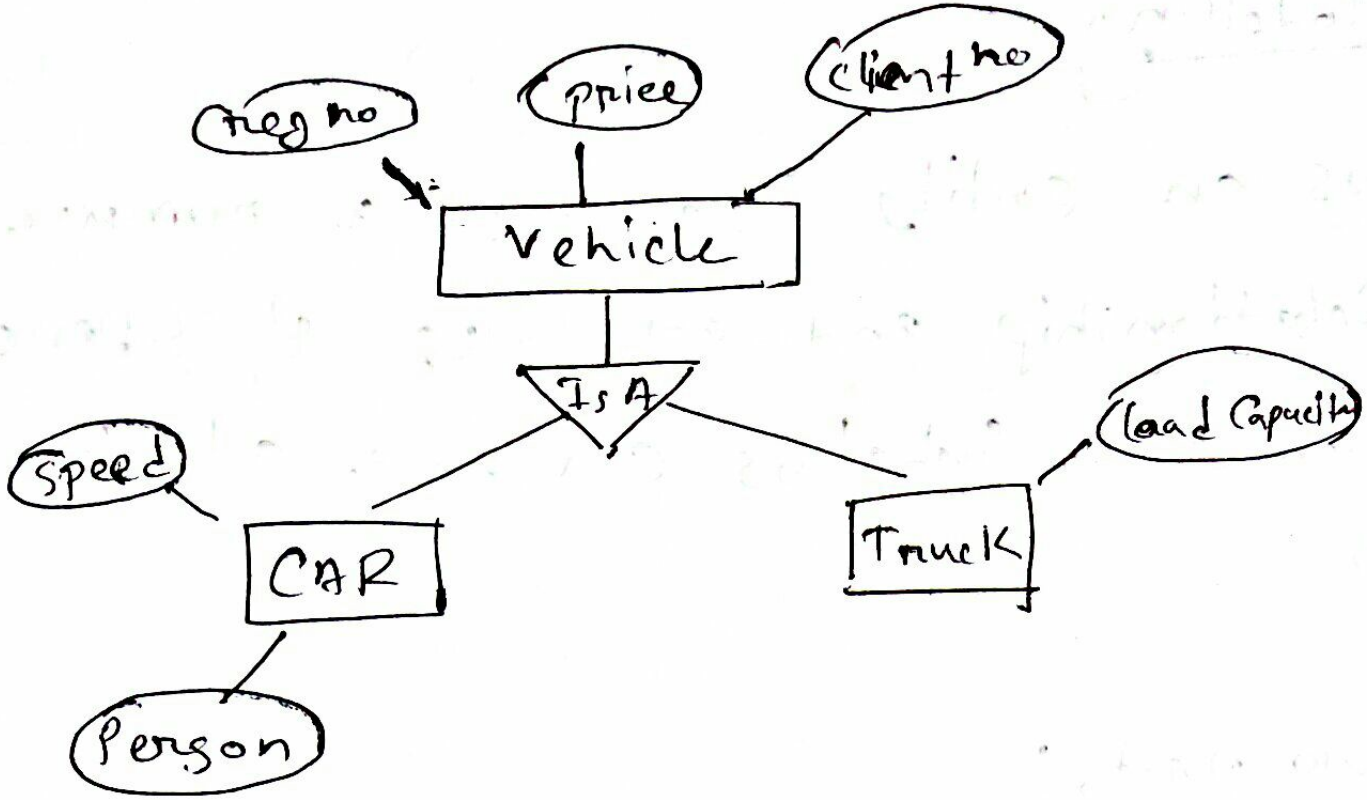
It's having two parts:

1) Specialization
2) Generalization

Specialization: Process of defining a set of sub classes of an entity type (top-down)



Generalization: Reverse process of Specialization (bottom up), identify common features of entity type and generalize them into single superclass.


reg no
price
client no
Vehicle
Is A
Speed
Load Capacity
CAR
Truck
Person

Lecture -7

16.05.17

Relational Algebra: The relational algebra is a theoritical language with operations that work on one or more relations, to define another relation without changing the original relation. It is used to manipulate relations to obtain a designed result.

Relational Algebra Operations:

Union, intersection, difference, division, renaming, selection, projection, cartesian product, join etc.

- **Unary Operation:** Operate with one relation
  F.E. Selection, projection etc.

Binary Operation: Operate on two relations.

F.E: Union, Intersection, join etc.

N-ary Operation:

- Selection operation:

  Syntax: σ predicate (R)

  σ = selection (sigma)

  predicate = Condition

  R = Relation

Symbols :

$<$ $>$ or $\neq$

$>, <, \geq, \leq, =$

$\wedge$, $\vee$, Not

$\downarrow$ $\downarrow$

AND OR

Example :

Student

Roll	Name	GPA
01	Rupa	4.5
02	Rimi	3.5
03	Shuvo	2.5
04	Sami	3.2
05	Poly	2.7
06	Keya	2.9

Employee

(3) Find the students whose gpa ≥ 3.0

$\sigma_{GPA \geq 3.0}$ (Student)

(4) Find those Employee whose age below 30 and salary 40k

=) $\sigma_{age<30 \wedge salary = 4000}$ (Employee)

(5) Find the Employee whose name is 'Luky'

=) $\sigma_{name = 'Luky'}$ (Employee)

Employee

Name	Age	Salary
Many	25	9000
Luky	40	3000
Mark	36	4500
John	42	3900

(1) Find those Employee whose salary is more than 4000.

⇒ σ salary > 4000 (Employee)

(2) Find those Employe whose age is less than 30 years.

⇒ σ age < 30 (Employee)

(3) Find those Employee whose age below 30 or salary more than 4000.

⇒ σ age < 30 ∨ Salary > 4000

Projection

Regression operation: Projection operation is used to show the desired column of a relation

Syntax

$\pi_{a_1, a_2, \ldots - a_n}(R)$

π = projection (Pie)

$a_1, \ldots - a_n$ = attributes

Staff

Name	Gender	Date of Birth	Salary
Rony	M	01/05/86	20,000
Tony	M	05/07/80	30,000
Jony	M	07/02/95	40,000
Rupa	F	03/09/96	35,000

(1) Show the names of staff relation

⇒ Π_{name} (Staff)

(2) Show the name, Date of Birth of Staff

⇒ $\Pi_{name, Date\ of\ Birth}$ (Staff)

(3) Find the staff whose salary is greater than 30000 and Gender is 'F'?

⇒ $\Pi_{name}(\sigma_{salary > 3000 \wedge gender = 'F'}(Staff))$

Lecture - 8

22.05.17

Customer

Name	Street	City	Phone	Age
Ruchi	ABC	DHAKA	123	16
Rahim	def	KHULNA	456	18
Upoma	ghi	RAJSHAHI	789	20
Smita	JKL	DHAKA	135	17

- Find the names of the customers who lives in DHAKA

→ $\Pi_{name}(\sigma_{city=DHAKA}(customer))$

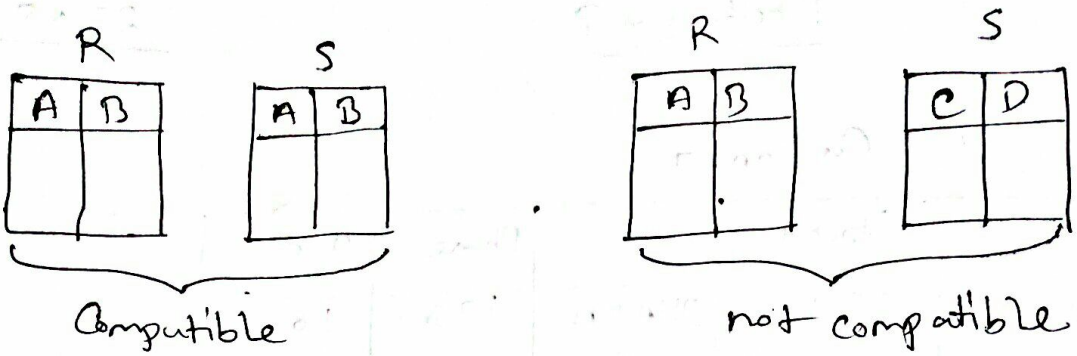
- Find the customer name whose age is more than 16 and lives dhaka where his/her phone number starts with '1'

=) $\Pi_{name}(\sigma_{age>16 \wedge city='DHAKA' \wedge phone='1\%'}(customer))$

Binary Operation Compatibility: To perform union, intersection and difference, relations should be union compatible. 2 relations are union compatible if they have same number of attributes and belong to same domain.

→ Column need to be same

→ Domains need to be same



Suppose we have 2 relations

R

A	B
α	1
α	2
β	1

S

A	B
α	2
β	3

Union Operation

R ∪ S is

A	B
α	1
α	2
β	1
β	3

Intersection Operation

R ∩ S is

A	B
α	2

Difference Operation:

R − S is

A	B
α	1
β	1

S − R is

A	B
β	3

Rename Operation:

- To rename a table : $\rho_{A}(B)$ (B → Old name, A → new name)
- To rename attribute:

$$\rho_{\text{oldname} \to \text{newname}}(R)$$

Example:

Employee (name, branch, salary)

After,

$$\rho_{\underbrace{\text{Branch, salary}}_{\text{old}} \to \underbrace{\text{location, Pay}}_{\text{New}}}(\text{Employee})$$

we will get

Employee (name, location, Pay)

Division Operation

a, common

R

A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_3	b_2

(R)

S

B
b_1
b_2

(S)

So, R/S =

A
a_1

Output

Cartesian Product Operation

R

A
a
b

S

B
1
2
3

R X S

A	B
a	1
a	2
a	3
b	1
b	2
b	3

Example : Division operation

A

S.no	P.no
S_1	P_1
S_1	P_2
S_1	P_3
S_1	P_4
S_2	P_1
S_2	P_2
S_3	P_2
S_4	P_2
S_4	P_4

B_1

P.no
P_2

B_2

P.no
P_4

i) A / B_1 ?

ii) A / B_2 ?

A / B_1 =

S.no
S_1
S_2
S_3
S_4

A / B_2 =

S.no
S_1
S_4

Lecture - 9

Borrower

Name	Loan No
KAMAL	L-17
JAMAL	L-23

Loan

Loan No	Branch	Amount
L-14	Kakrail	10000
L-23	Motijhil	25000

Find the names of Customers who have a loan at motijhil Branch ?

Borrower X Loan

Name	Borrow. Loan No	Loan Loan No	Branch	Amount
KAMAL	L-17	L-14	Kakrail	10000
KAMAL	L-17	L-23	Motijhil	15000
JAMAL	L-23	L-14	Kakrail	10000
JAMAL	L-23	L-23	Motijhil	15000

$$\Pi_{Name}\left(\sigma_{Borrow.LoanNo = Loan.LoanNo \wedge Branch = Motijhil}(Borrower \times Loan)\right)$$