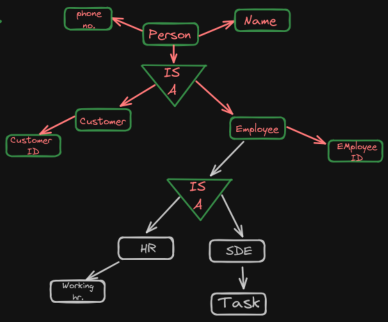
Lecture 4:-

Extended ER features

1) Specialization -> Provides 'is a' relation to avoid redundancy.

Ex.->

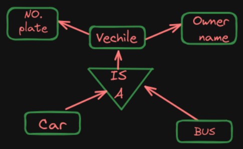


// Specialization is top down apporach
// We breaks the entity in sub entities
//Is similar to inheritance in OOP
// Establish 'IS A RELATION'
//Needed to avoid data redundancy and inconsistency
// To make DB design simpler
// differentiate the properties of sub entities

2) Generalization -> (reverse of specialization)

// It has bottom up approach

Ex.->



// Generalization is bottom up approach
// We create super entity through sub entity
// Having 'IS A RELATION'
//Needed to avoid data redundancy and inconsistency
// To make DB design simpler
// Creates super class for sub entities having some similar attributes

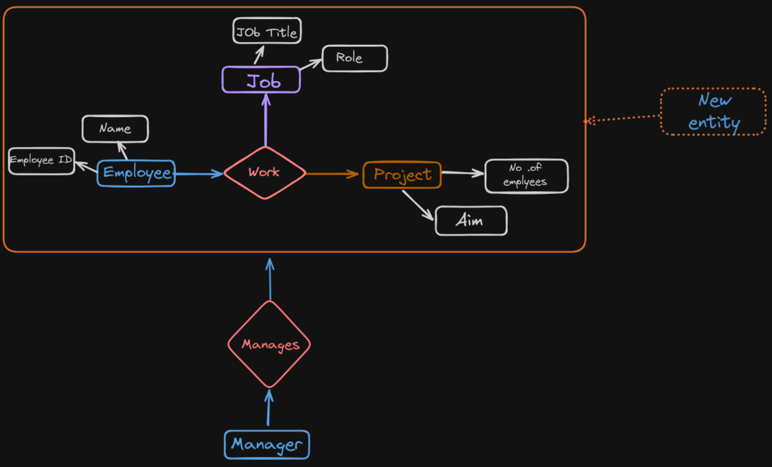
=> Attribute inheritance -> the sub entities inherit attributes of super entity

=> Paticipation inheritance -> When super entity is in relation with other entity then automatically it's sub entities are also in relation with that other entity

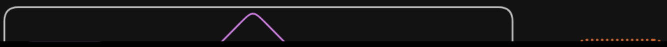
3) Aggregation -> Used to show RELATIONSHIP among relationship
We create an entity of the relation entities to relate with super entity

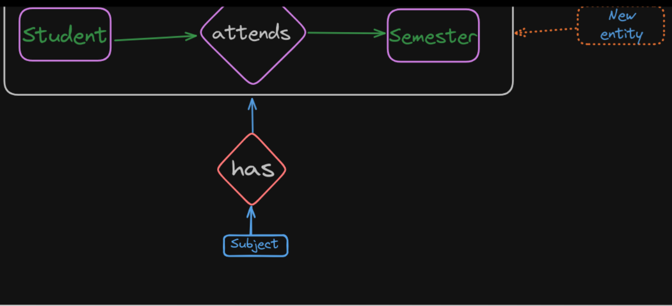
By applying abstraction

Ex. ->



Ex.2 ->





Lecture 5:-

ER - Diagrams

ER - Diagrams are used to design DB schema

Steps to make ER - Diagram

1) Identify entity sets
2) Identify attributes and their type
3) Identify relationship -> type of relationship -> Constraints
↓ Strong or weak
↓ Mapping and paticipation

=> ER - Diagram for Banking system

Possiblities :-
i) Branches
ii) Customers, having accounts and take loan
iii) Loan
iv) Accounts -> saving and current
v) Loan by some branch -> payment
vi) Banker
vii) Employees

->Assuming Entities (step1)

i) Branch
ii) Customer
iii) saving and current account (generalization -> accounts)
iv) Loan
v) Loan payment (weak entity)
vi) Employee

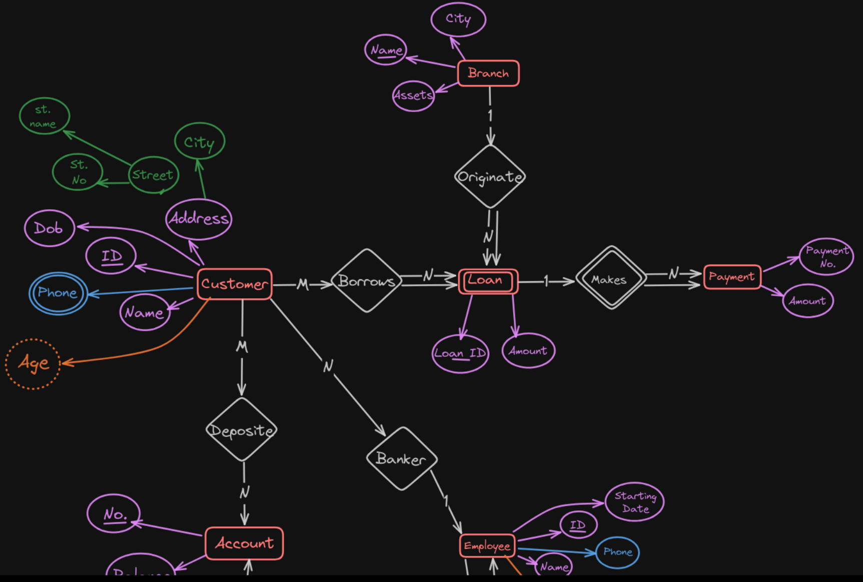
->Assuming Attributes (step2)

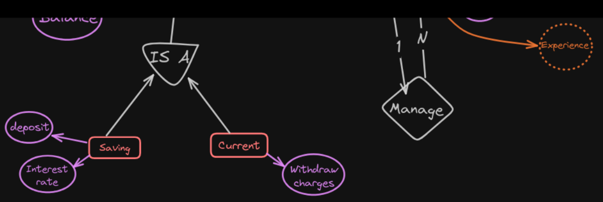
i) Branch -> id, name, assets
ii) Customer -> id, name, address (composite), dob (single valued), phone no (multi), age (derived)
iii) accounts account no. (key), balance
iv) saving acc. -> deposit, interest rate
v) current acc. -> withdraw charges
vi) Loan -> amount, id
vii) Employee -> name, id, years of experience, starting date
viii) Payment -> payment no., date

->Assuming Relationship (step3)

i) Customer - borrows - Loan (M : N)
ii) Loan - makes - payment (1 : N)
iii) Loan - originate - branch (N : 1)
iv) Customer - deposite - account (M : N)
v) Customer - banker - employee (N : 1)
vi) Employee - manage by - employee (N : 1)

ER - Diagram





HW -> E-commerce system ER Diagram

->Assuming Entities (step1)

i) Customer
ii) Delivery Branch
iii) E commenrce Company
iv) Employee
iv) Payment method -> UPI, Cash on delivery
v) Customer service

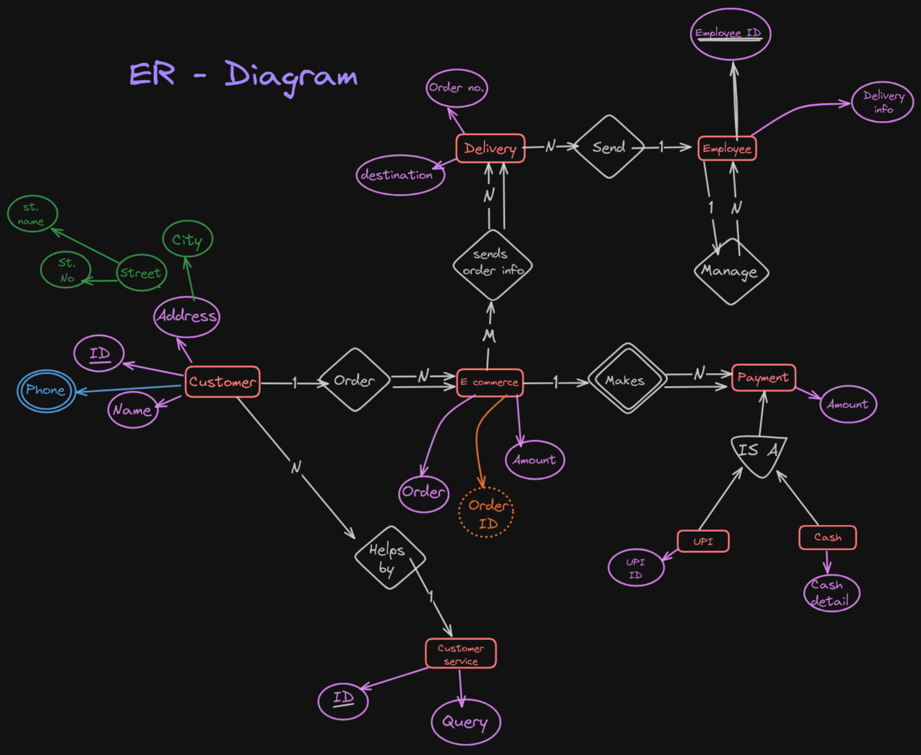
->Assuming Attributes (step2)

i) Customer -> order, current_order, id, address, phone no.
ii) Delivery branch -> destination, order no.
iii) Employee -> id, phone no., area_to_deliver
iv) Payment -> amount
v) customer service -> query, query id
vi) E-commerce (Order) -> order, order ID, order amount

->Assuming Relationship (step3)

i) Customer - orders from - E-commerce (Order) (1:N)
ii) Delivery branch - recives_from - E-commerce (Order) (M:N)
iii) Delivery branch - send_order_info - employee (N:1)v
iv) Customer - pays - payment (1:1)
v) Customer_service - help- customer (1:N)
vi) Employee manage employees (1:N)
vii) Employee - deliver to - customer (1:1)

ER - Diagram


Employee ID
Order no.
Delivery info
Delivery
N
Send
1
Employee
destination
N
1
N
sends order info
Manage
st. name
City
St. No
Street
Address
M
ID
Customer
1
Order
N
E commerce
1
Makes
N
Payment
Phone
Name
Amount
Amount
IS A
N
Order
Order ID
UPI
Cash
Helps by
UPI ID
Cash detail
1
Customer service
ID
Query