Lecture 7

Database Management System

Er. Shiva Kunwar

Lecturer, GU

Lesson 2: Data Model (6hrs)

1. Different Data Model Concepts
2. E-R Model
3. Network/Hierarchical Model
4. Relational Model
5. Entities, Relationships and Attributes
6. **E-R Diagrams**, Keys
7. **Generalization, Specialization and Aggregation**

Topics

- E-R diagram
- Reducing E-R diagram to tables
- Generalization, Specialization and Aggregation

ER Diagram Example

- In a university, a Student enrolls in Courses. A student must be assigned to at least one or more Courses. Each course is taught by a single Professor. To maintain instruction quality, a Professor can deliver only one course.

Example

- Step 1) Entity Identification
- We have three entities

- Student
- Course
- Professor

Example

- Step 2) Relationship Identification
- We have the following two relationships

- The student is assigned a course
- Professor delivers a course

Example

- Step 4) Identify Attributes
- Once, you have a list of Attributes, you need to map them to the identified entities. Ensure an attribute is to be paired with exactly one entity. If you think an attribute should belong to more than one entity, use a modifier to make it unique.

- Once the mapping is done, identify the primary Keys. If a unique key is not readily available, create one.

Example

- Step 4) Identify Attributes
- Entity Primary Key Attribute
- Student Student_ID StudentName
- Professor Employee_ID ProfessorName
- Course Course_ID CourseName

Example

- Step 4) Cardinality Identification
- For them problem statement we know that,

- A student can be assigned multiple courses
- A Professor can deliver only one course

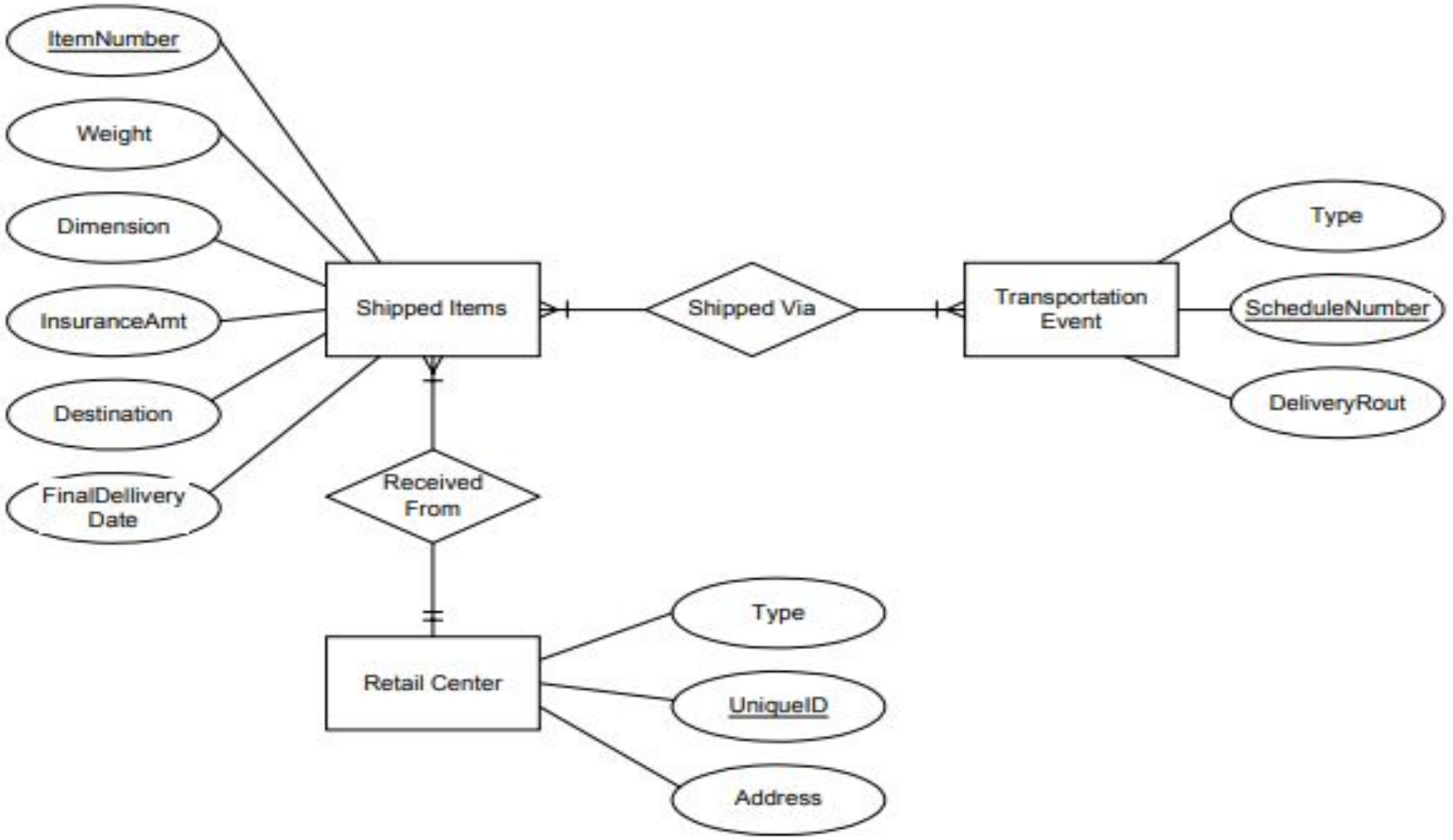
Practical

- Construct an ER diagram for a car insurance company with a registration id and an address and have name and address of its customers. Each customer own one or more cars which has its model, color and engine number. Each car is associated with zero or any number of recorded accidents.

Practical

- UPS prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system.
- Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center.
- Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute.
- Prepare an ER Diagram that captures this information.

Practical


ItemNumber
Weight
Dimension
InsuranceAmt
Destination
FinalDellivery Date
Shipped Items
Shipped Via
Transportation Event
Type
ScheduleNumber
DeliveryRout
Received From
Retail Center
Type
UniqueID
Address

Reducing ER Diagrams to Table

- We can represent the E-R database schema by a set of tables.
- Each entity sets and each relationship sets in E-R schema can be represented by their corresponding tables.
- Each attributes of entity sets, and relationship sets are map as columns of their corresponding tables.
- Similarly, constraints specified in E-R diagram such as primary key, cardinality constraints etc. are mapped to tables generated from E-R diagram.
- In fact, representing E-R schema into tables is converting E-R model of database into relational model

Reducing ER Diagrams to Table

- Consider a simple ER diagram with two entities: "Student" and "Course," and a many-to-many relationship "Enrolls In.“
- *Student: StudentID, Name, GPA*
- *Course: CourseID, Title, Department*
- *Enrolls In: EnrollmentID (PK), StudentID (FK), CourseID (FK)*
- Create Tables Student table, Course table, and enrolls table that represent many-to-many relation

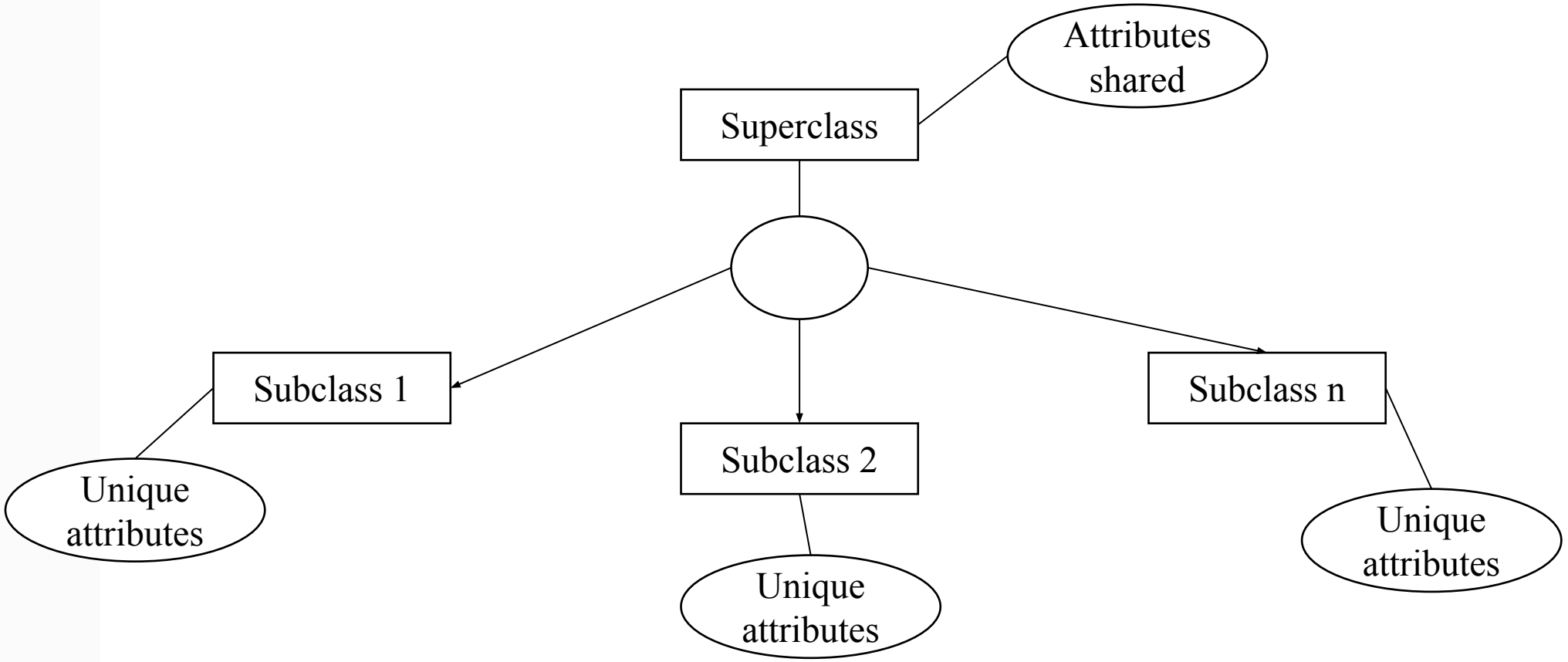
Extended ER Features

- In the pursuit of enhancing the ER model, several terms and concepts have been incorporated.
- Generalization refers to the process of abstracting common features from multiple entities into a more generalized entity, reducing redundancy and promoting a higher level of abstraction.
- Specialization, conversely, involves refining a generalized entity into more specialized entities, tailoring attributes and relationships to specific subsets of data.
- Aggregation, facilitates the modeling of relationships involving higher-level entities composed of or related to lower-level entities.

Superclass & Subclass

- **Superclass**
- A superclass is a generalized entity that contains common attributes and relationships shared by multiple specialized entities or subclasses.
- It represents the more abstract, generalized concept.
- **Subclass**
- A subclass is a specialized entity that inherits attributes and relationships from a superclass.
- It represents a more specific, specialized concept.
- Attributes are unique from other subclasses.

Superclass & Subclass



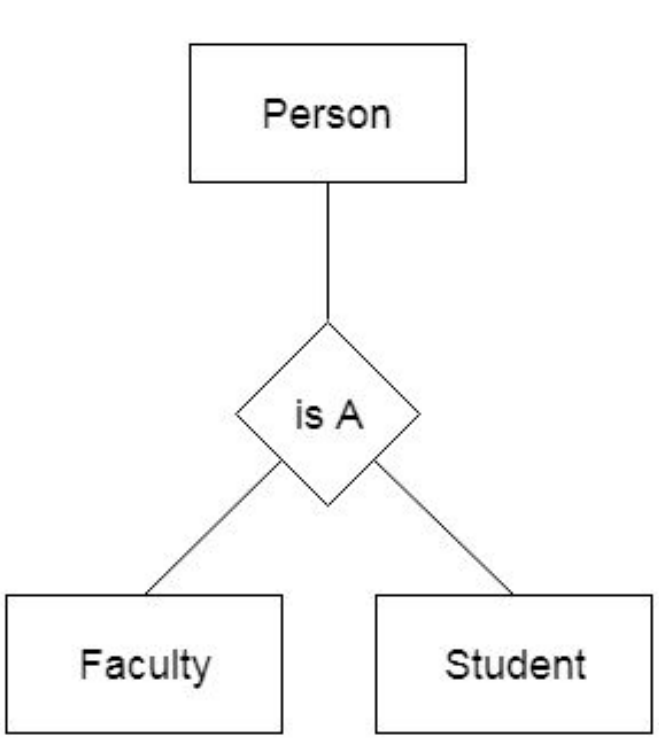
Generalization

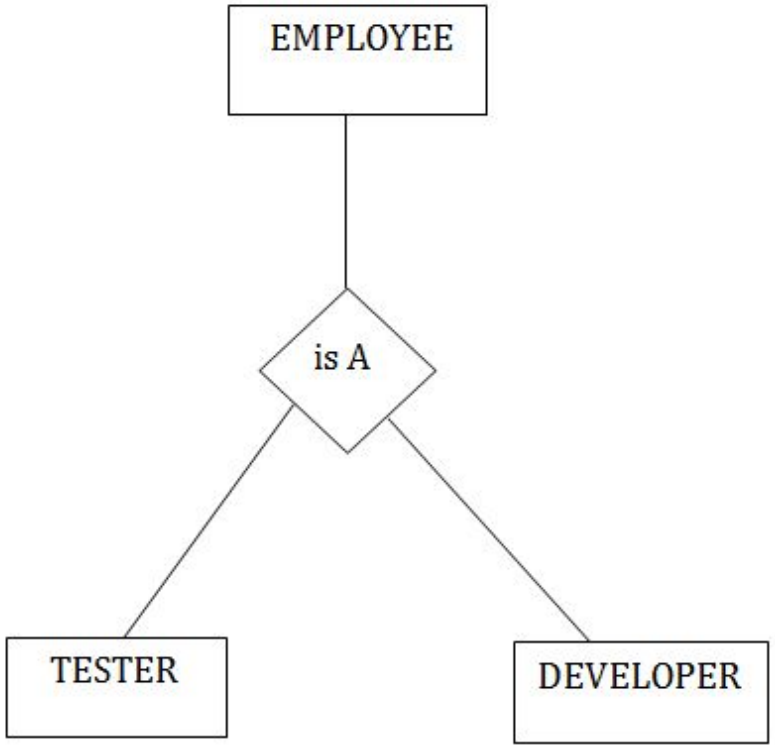
- It involves abstracting common features from multiple entities to create a more generalized entity.
- It is a bottom-up approach.
- **Characteristics**:
  - Reduces redundancy by identifying and representing commonalities.
  - Enhances simplicity by creating a higher-level, more abstract entity.
- *Consider entities like 'Car' and 'Truck.' These entities share common attributes such as 'Make,' 'Model,' and 'Year.' Generalization allows the creation of a more abstract entity called 'Vehicle,' which captures the shared characteristics of both 'Car' and 'Truck.'*

Specialization

- It involves refining a generalized entity into more specialized entities.
- It is a top-down approach.
- **Characteristics**:
- Allows for the representation of diverse entities with unique characteristics.
- Enables the modeling of specific subtypes within a broader category.
- *Building upon the 'Vehicle' entity, specialization creates more specific entities like 'Car' and 'Truck,' each inheriting the common attributes from the 'Vehicle' entity while adding attributes specific to its type.*
- This also explains the concept of Inheritance.

Generalization-Specialization





Generalization-Specialization Constraints

- **Attribute Constraints**
- Attributes can be associated with a superclass or a subclass.
- An attribute associated with a superclass is shared by all its subclasses.
- An attribute associated with a subclass is specific to that subclass.
- *If 'Person' has a 'Name' attribute, it is shared by both 'Employee' and 'Customer.' If 'Employee' has a 'Salary' attribute, it is specific to the 'Employee' subclass.*

Generalization-Specialization Constraints

- **Disjoint Constraint**
- Specifies that a superclass entity can belong to only one subclass.
- Denoted by a letter "d" next to the line connecting the superclass to the specialization symbol.
- *If 'Animal' is disjointly specialized into 'Mammal' and 'Bird,' an 'Animal' can be either a 'Mammal' or a 'Bird,' but not both.*

Generalization-Specialization Constraints

- **Overlap Constraint**
- Allows a superclass entity to belong to more than one subclass.
- Denoted by a letter "o" next to the line connecting the superclass to the specialization symbol.
- *If 'Fruit' is overlapped specialized into 'Citrus' and 'Berry,' a 'Fruit' can be both a 'Citrus' and a 'Berry.'*

Generalization-Specialization Constraints

- **Total Completeness Constraint**
- Specifies that every superclass entity must belong to at least one subclass.
- Denoted by a double line connecting the superclass to the specialization symbol.
- *If 'Vehicle' is totally specialized into 'Car' and 'Truck,' every 'Vehicle' must be either a 'Car' or a 'Truck.'*

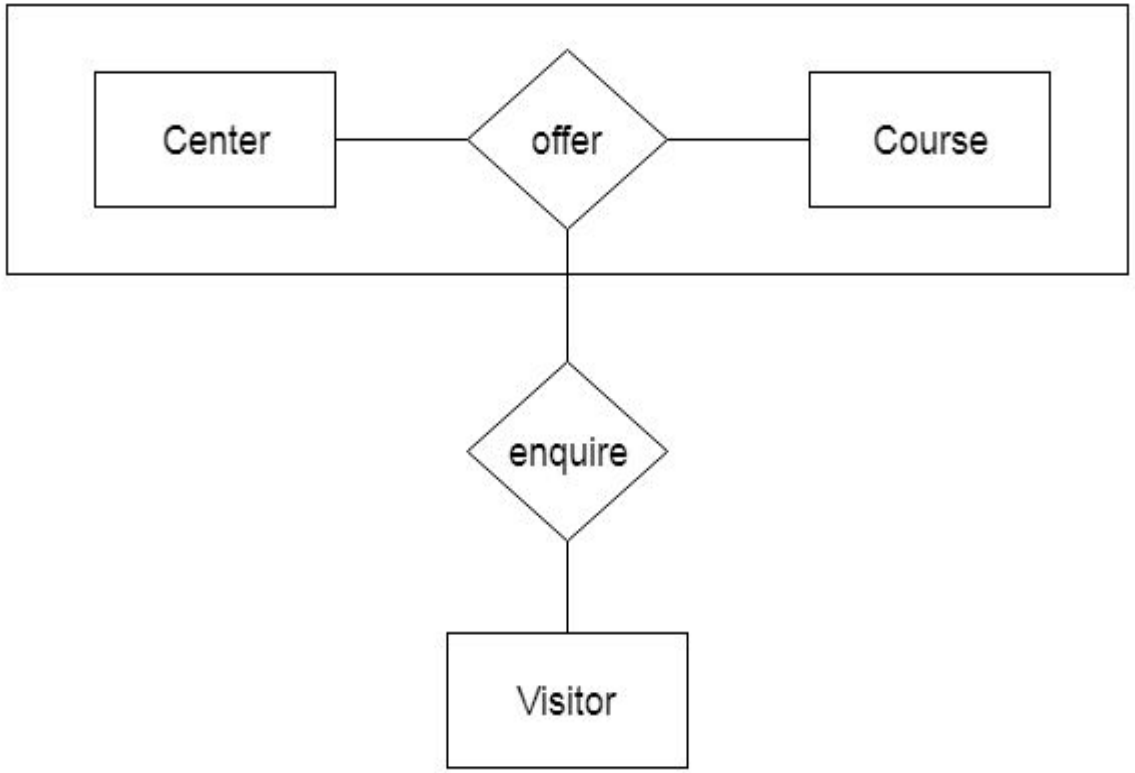
Generalization-Specialization Constraints

- **Partial Completeness Constraint**
- Allows superclass entities to exist without being members of any subclass.
- Denoted by a single line connecting the superclass to the specialization symbol.
- *If 'Person' is partially specialized into 'Employee' and 'Customer,' a 'Person' can exist without being an 'Employee' or 'Customer.'*

Aggregation

- Allows modeling of relationships involving higher-level entities composed of or related to lower-level entities.
- It helps in representing complex relationships more effectively.
- **Characteristics**:
- Useful for expressing complex relationships and hierarchies.
- Allows for the creation of composite entities representing a whole-part relationship.
- *Consider an 'Order' entity that aggregates 'Product' entities. The 'Order' entity represents a higher-level concept that consists of individual 'Product' entities.*

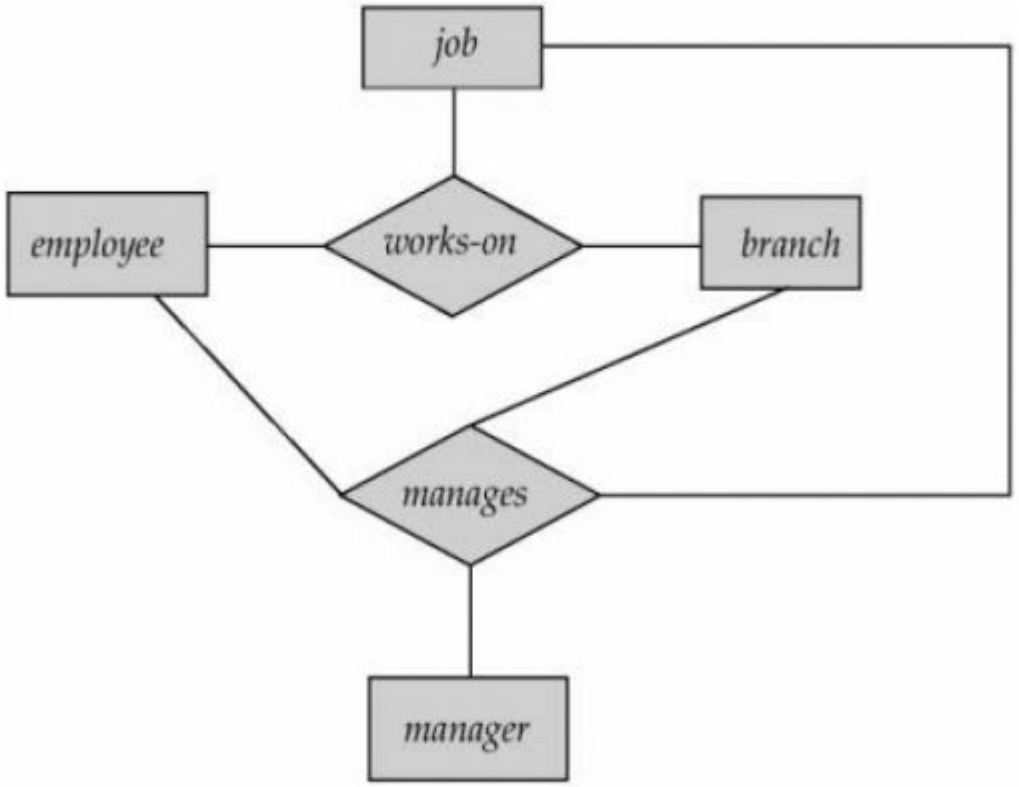
Aggregation



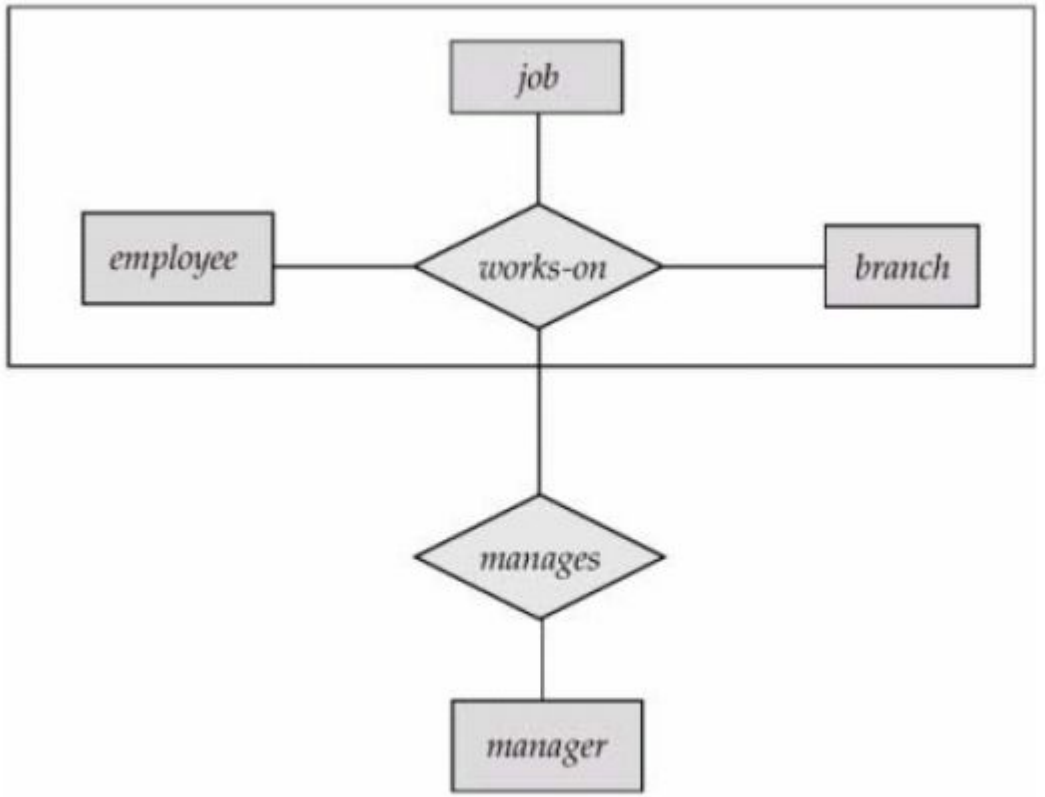
Aggregation Scenario

- Managers who manages particular job/task performed by particular employee at particular branch.

Aggregation



E-R diagram with redundant relationships.



E-R Diagram with aggregation

Practical

- Draw the ER diagram for a small database for a bookstore. The database will store information about books for sale. Each book has an ISBN, title, price and short description. Each book is published by a publisher in a certain publishing year. For each publisher, the database maintains the name, address and phone number.
- Each book is written by one or more authors. For each author, the database maintains his/her ID, name and a short introduction. Each book is stored in exactly one warehouse with a particular quantity. For each warehouse, the database maintains the warehouse name, the location and the phone number. Each book has one or more sellers, which may be either companies (corporate vendors) or individuals (individual vendors).

Practical

- For each company, the database maintains a name of the company, its address, its phone numbers (there could be more than one phone number, each with a number and a description) and its contact person. For each individual vendor, the database keeps a name, a phone number and an email address. A contact person whose company sells a book cannot be selling the same book as an individual vendor at the same time (he/she may sell other books as an individual seller).

Practical

- Identify all the entities
- -AUTHOR
- -PUBLISHER
- -BOOK
- -CUSTOMER
- -SHOPPING_BASKET
- -WAREHOUSE

Practical

- find the relations
- 1.Each book is written by a author
- 2.Each book has a publisher
- 3. Some shopping baskets may contain more than one copy of same book
- 4. The warehouse stocks several books
- 5. A customer owns several shopping basket

Practical

- Identify the key attribute
- *AUTHOR- name
- *PUBLISHER- name
- *BOOK- ISBN
- *CUSTOMER- email
- *SHOPPING_BASKET- basket_ID
- *WAREHOUSE- code

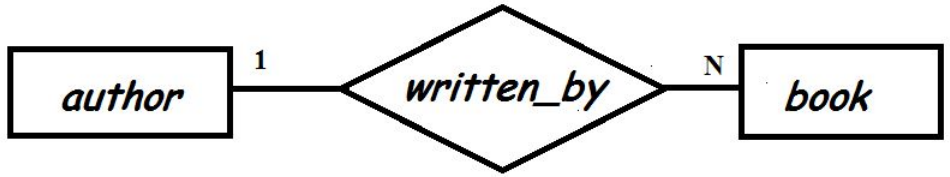
Practical

- Identify other relevant attributes
- *AUTHOR- name,address,URL
- *PUBLISHER- name,address,URL,phone
- *BOOK- ISBN,year,title,price
- *CUSTOMER- email, name,address,phone
- *SHOPPING_BASKET- basket_ID
- *WAREHOUSE- code, address,phone

Practical

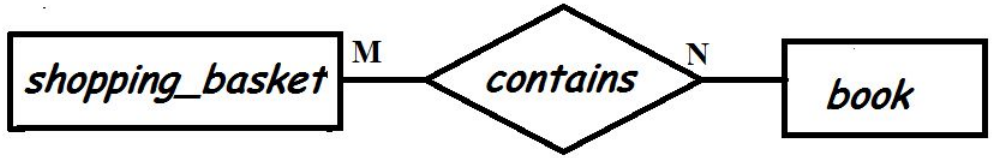
relations

1.Each book is written by a author



2.Each book has a publisher

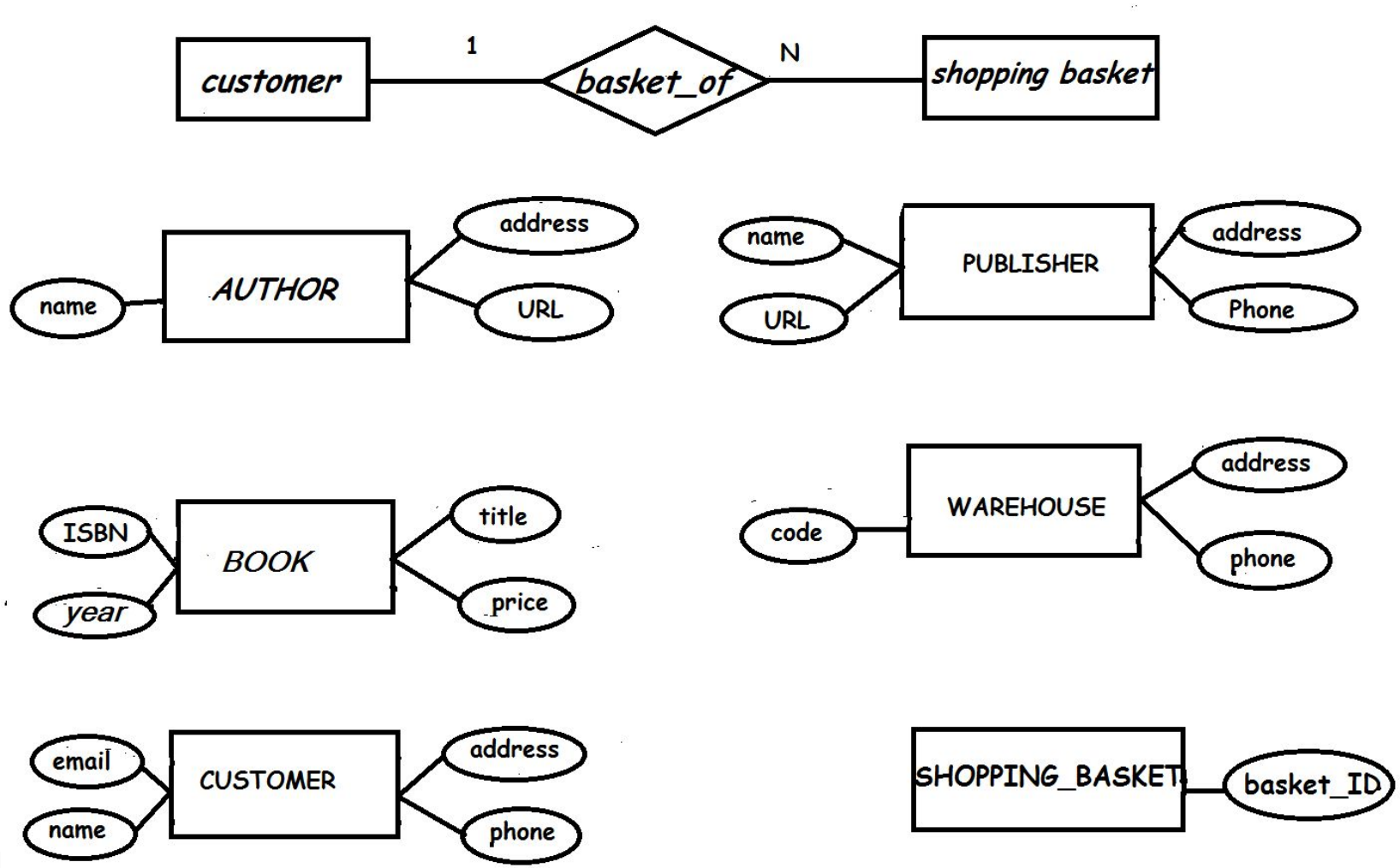
3. Some shopping baskets may contain more than one copy of same book

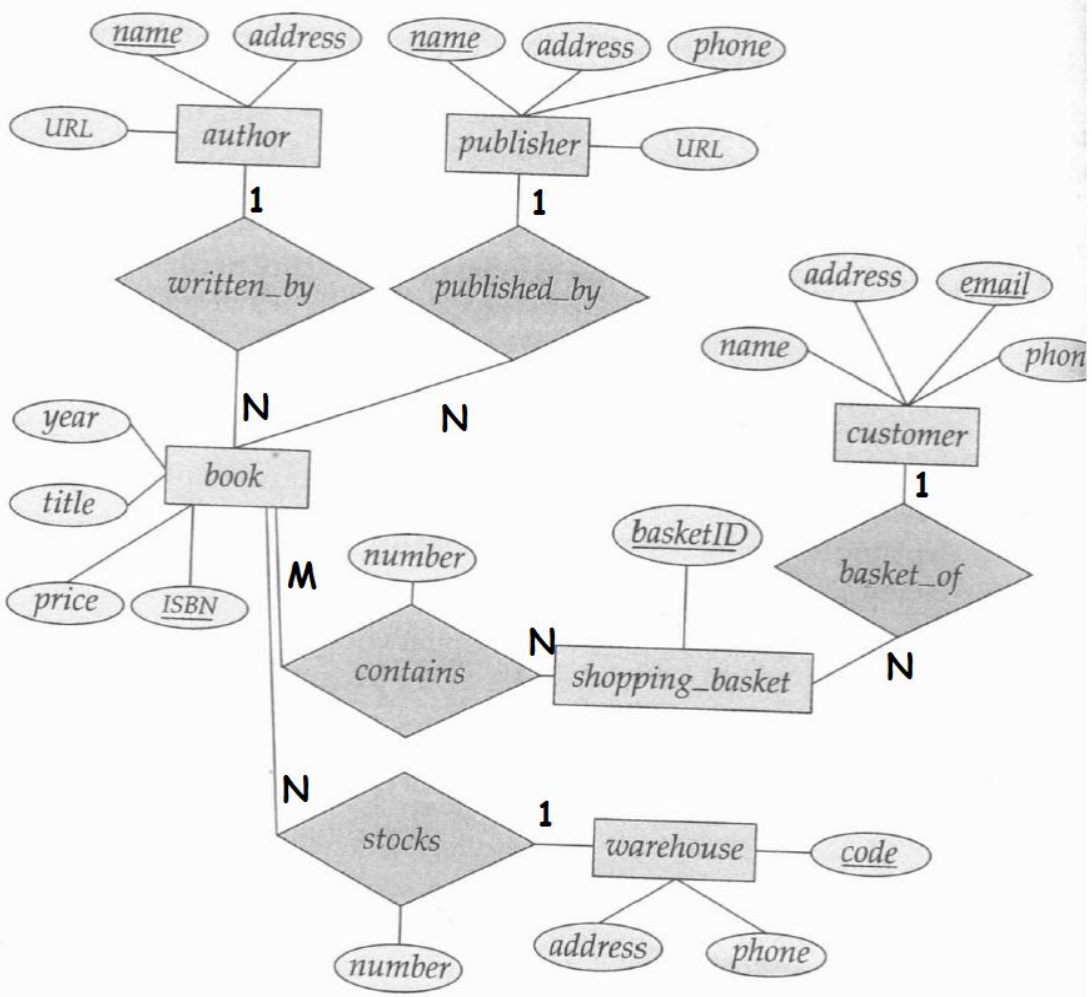


4. The warehouse stocks several books

Practical

5. A customer owns several shopping basket





E-R DIAGRAM OF ONLINE BOOKSTORE

END OF LECTURE 7

- SHIVA.KUNWAR@HOTMAIL.COM
- +977-9819123654

Google classroom code : drbzdcf

PREVIEW FOR LECTURE 8

RELATIONAL MODEL

INTRODUCTION TO RELATIONAL DATABASE

DATABASE SCHEMA AND VIEWS

RELATIONAL MODEL CONSTRAINTS