Database Systems (ISYS1001/ISYS5008)

Lecture 3

Entity-Relationship (ER) Model

Updated: 11th August,2021

Discipline of Computing
School of Electrical Engineering, Computing and Mathematical Sciences (EECMS)

CRICOS Provide Code: 00301J

Copyright Warning

COMMONWEALTH OF AUSTRALIA
Copyright Regulation 1969

WARNING

This material has been copied and communicated to you by or on behalf of **Curtin University of Technology** pursuant to Part VB of the *Copyright Act 1968* (**the Act**)

The material in this communication may be subject to copyright under the Act. Any further copying or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice

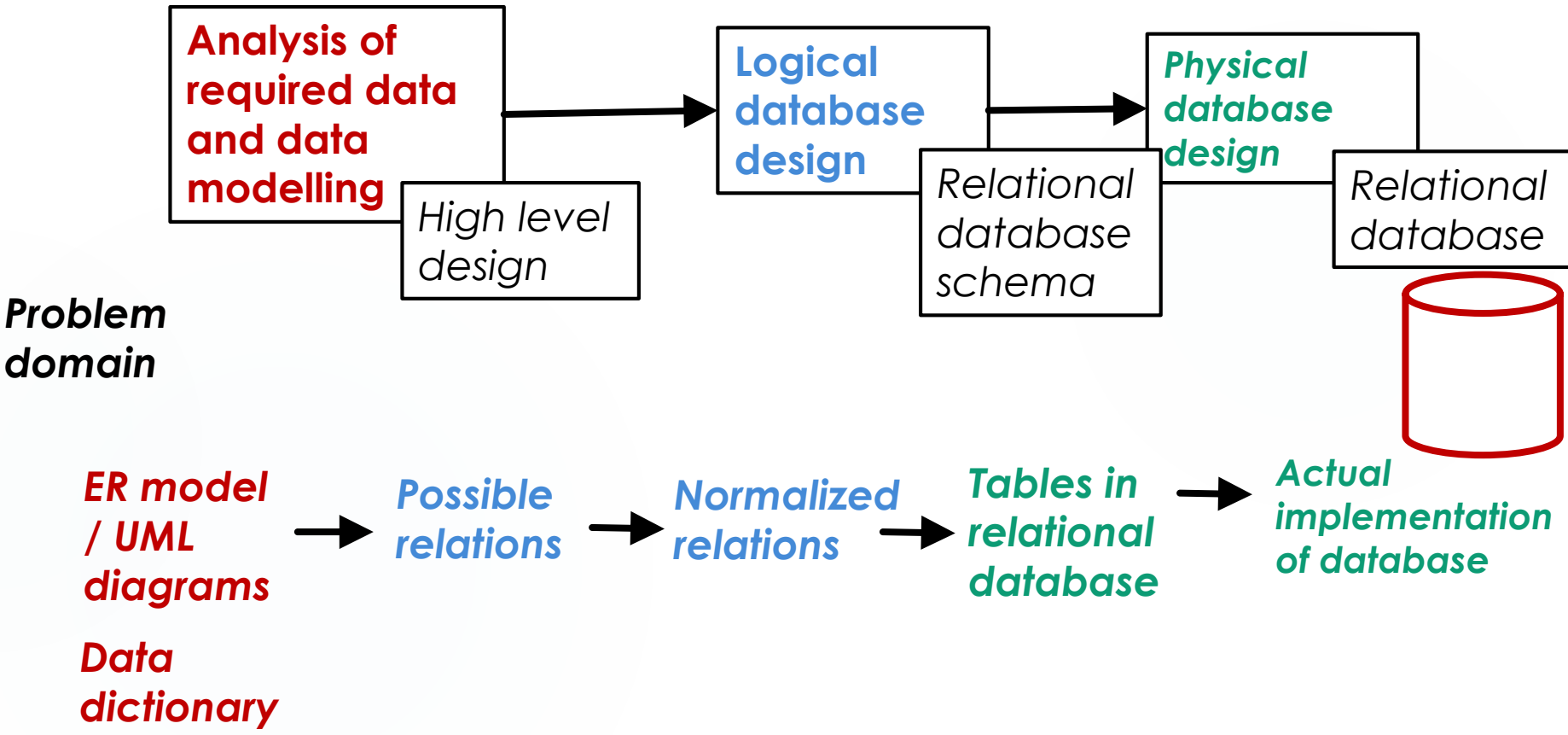
Learning outcomes

- Explain the three stage process of database modelling.
- Describe the main constructs of an ER model.
- Create an ER model to represent a real world problem, identifying entities, attributes and relationships.
- Use proper relationship types, attribute types and constraints in ER model.

Real world to database

- Databases store and manipulate data about real world or some conceptual scenarios
- Data needs to be modelled properly so that actual data items can be used for various purposes
- Though the functional requirements of a computer system may change quickly, the underlying data model may not need such changes, if properly modelled.

Real world to database



ER model (ERM)

- A model which supports representing the data needs to be stored in a database
- A graphical representation showing entities, attributes and relationships
- Introduced by Peter Chen in 1976
- ER model leads to foundation to OO design and later to UML
- There are many variations of notations evolved over the time to represent an ER model
  - Chen's notation, Crow-foot notation, IE notation …
- Some notations can represent certain aspects better than other however, basic constructs are mostly the same
- CASE tools supports some of the notations
- We will use variation of Chen's notation and then use other notations as well

Key components of a ER Model

Three core modelling components

1. **Entity**
2. **Attributes**
3. **Relationships**

Additional components enhancing the model

- E.g. Cardinality

Entity and attributes

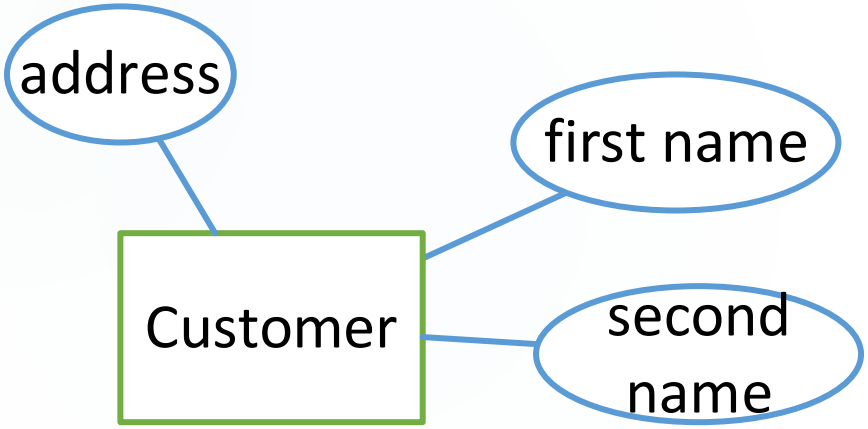
- Entity: primary objects of the problem domain
- Entity *or* entity instance *or* entity occurrence is an object in the real world with an independent existence
- Entity set *or* entity type : collection of similar entities
- Attributes : properties of (the entities of) an entity set
- Entity is characterised by its attributes

Entity and attributes

- Example:
  - *‘ Mary White’* can be an *entity instance* of the ***entity type*** called ***Student***
  - each individual student is an entity ; ***Student*** is an entity set or entity type.
  - Attributes of ***Student*** can include *Name, date of birth, address*
  - In the original ERM, attributes are **simple atomic** values, *e.g.*, numbers, dates or character strings.
- Entity and entity type terms are commonly used without distinguishing them as the meaning can be identified by the context

Entity and attributes : ER diagrams

- Entity set : rectangle (in many notations)
- Attribute : oval, with a line to the rectangle representing its entity set. (in many notations)
- Some notations do not represent attributes as a symbol and list them inside the entity rectangle



- Entity set Customer has three attributes, first name, last name and address.
- Each Customer entity has ***values*** for these three **attributes**

Key attributes

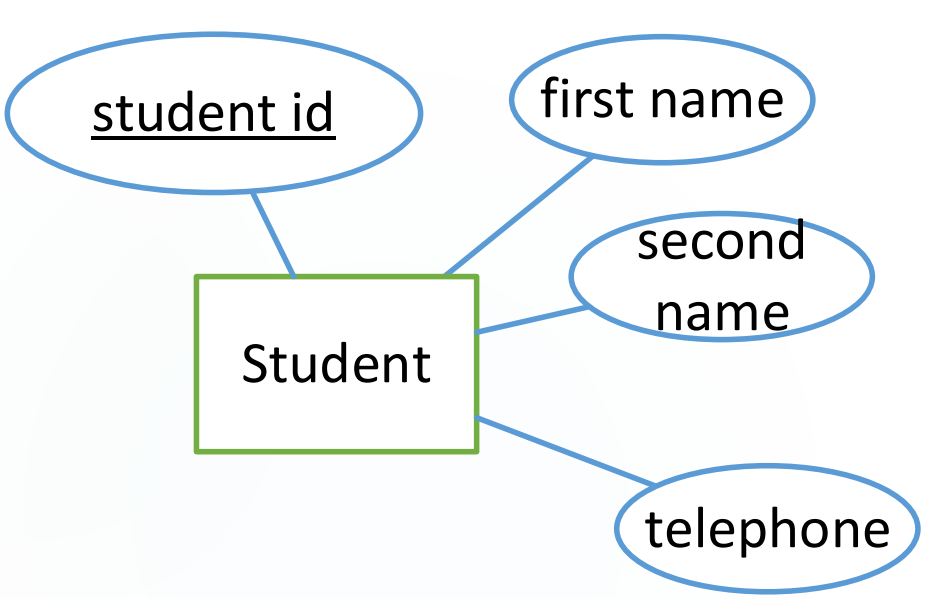
- An Entity set can have two types of attributes; identifying attributes and descriptive attributes
- Identifying attributes uniquely determine an instance of the entity type.
- Identifying attributes are called entity identifier or ***key attribute***
- A ***key*** is a set of attributes for one entity set such that no two entities in this set agree on all the attributes of the key.
  - It is allowed for two entities to agree on some, but not all, of the key attributes.
- A **Key** attribute has to be defined for each entity set.

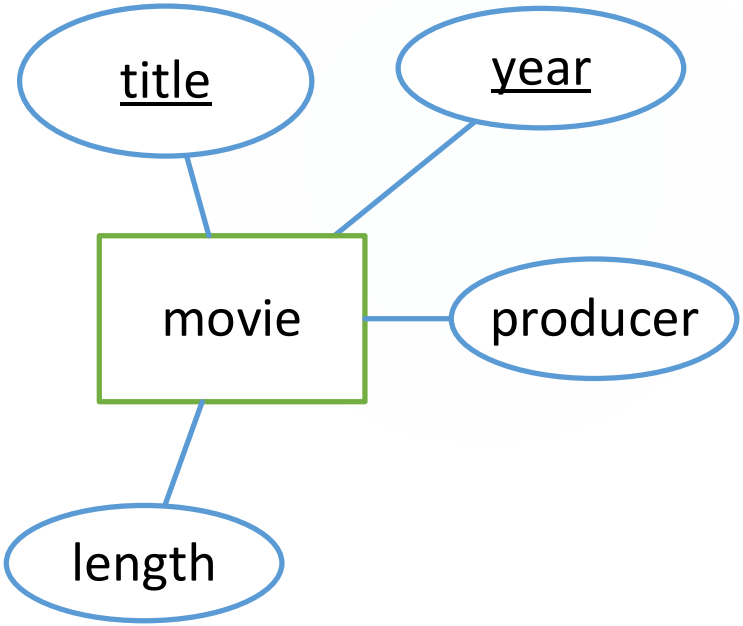
Attributes

- Descriptive attributes describes the non-unique properties of an entity set.
- Only attributes which has a meaning in the problem context are modelled
- E.g., For **Student** entity set, **student id** can be a key attribute. **name, date of birth, telephone** can be descriptive attributes

Key attributes: ER diagrams

- Key attributes are represented by underline the attribute



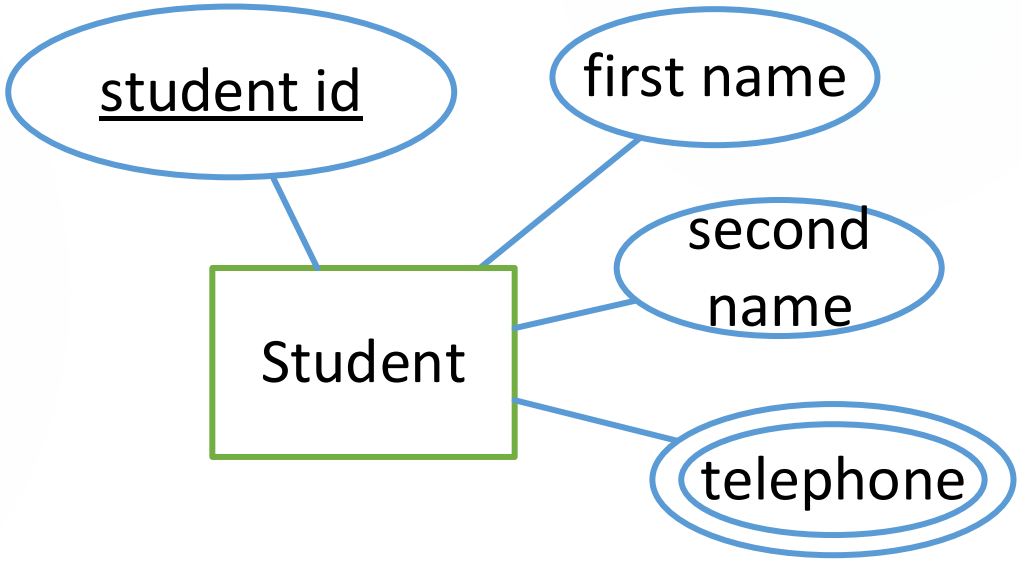


Composite key

More on attributes

- **Multi-valued attributes:** Some attributes may have multiple values instead of a single value for each instance of an entity
- E.g. : A S**tudent** can have **more than one** telephone numbers. Then attribute telephone becomes a multi-valued attribute.

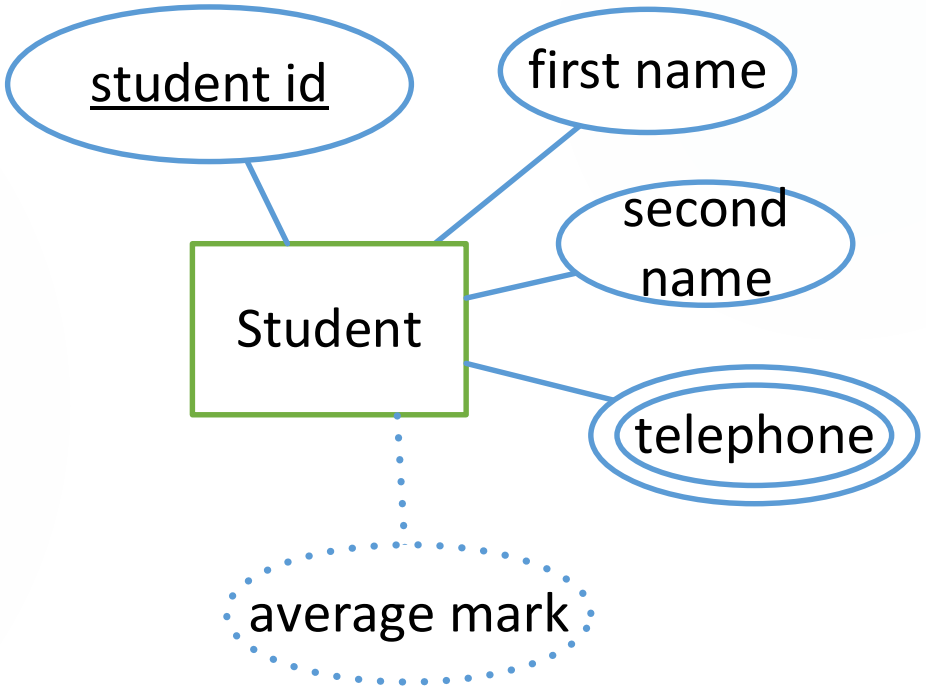
- Multi-valued attributes are represented by oval with double line



More on attributes

- **Derived attributes :** some attribute may have values which need to be derived based on other attributes
- E.g. : Average marks of a **Student** would be calculated based on other attributes. Therefore, average mark can be a derived attribute.

- Derived attributes are represented by a dashed oval

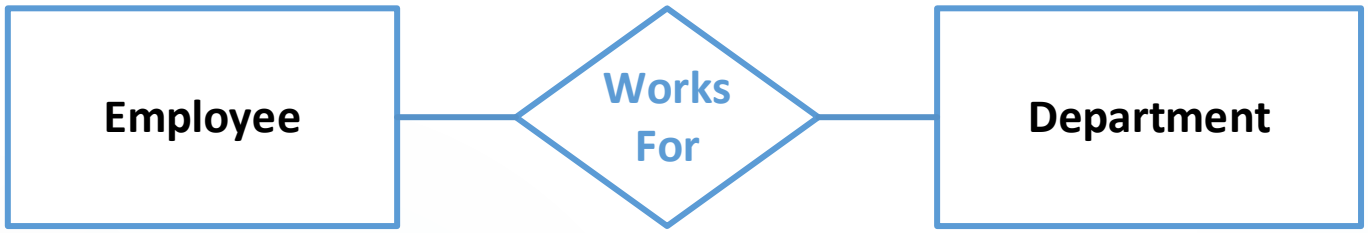


Relationships

- A relationship is a association between entity sets.
- The association should be a meaningful one.
- A relationship connects two or more entity sets.
- Relationship types:
  - **Unary relationships:** A meaningful association among a single entity set (recursive relationship)
  - **Binary relationship:** A meaningful association between two entity sets
  - **Ternary relationships** : A meaningful association between more than two entity sets.
- Some ERM notations does not support ternary relationships therefore ternary relationships should be converted to several binary relationships in such notations

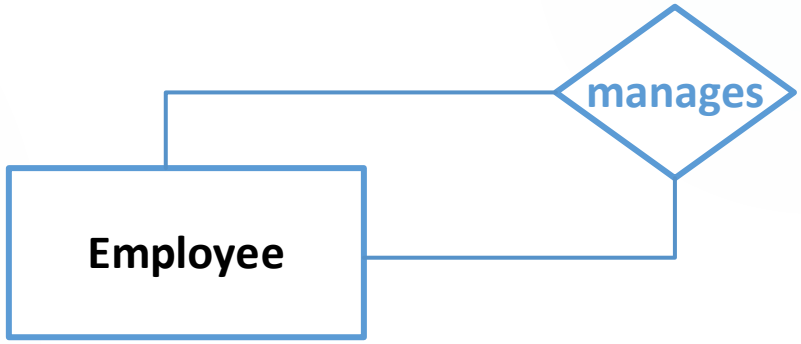
Relationships: ER diagrams

- In Chen's notation, a relationship is represented by a diamond, with lines to each of the entity sets involved.

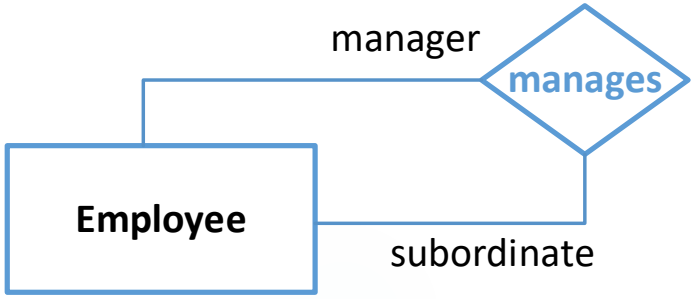


A binary relationship

A unary relationship

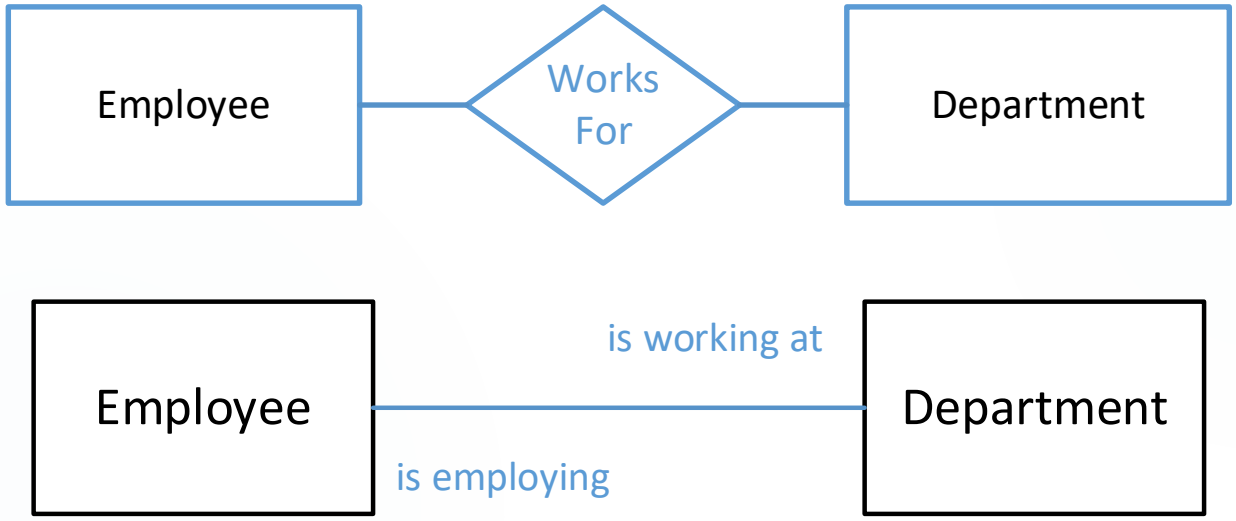


*In a unary relationship, an entity instance is having different **roles** in the relationship*



Relationships: ER diagrams

- Same information can be represented using other notations differently



Same information in different notations

More on relationships

- Relationships can show more information as constraints to provide better representation of the association between entity sets.
- **Cardinality constraint (degree of a relationship)** :
  - A constraint which specify the maximum number of relationship instances in which an entity can participate.
  - Value of cardinality is either one (1) or many (N)
  - Binary relationship can have three variations of cardinality:
    - One to one (1:1)
    - One to many (1:N)
    - Many to many (M:N)
  - 1:1:1, 1:1:N, 1:N:M, and M:N:P in ternary relationships.

Cardinality constraint

- In a *one-one* relationship, each entity of either entity set is related to at most one entity of the other set
- In a *many-many relationship*, an entity of either set can be connected to many entities of the other set
- In a *one-many relationship*, an entity of one set can connected to many entities of the other set

Cardinality constraint: examples

- many-many relationship

Consider relationship Purchases between Customers and Books entity sets

A customer can purchase many Books; A book is purchased by many customers.

- one-many relationship

Consider relationship Works between Employee and Department entity sets.

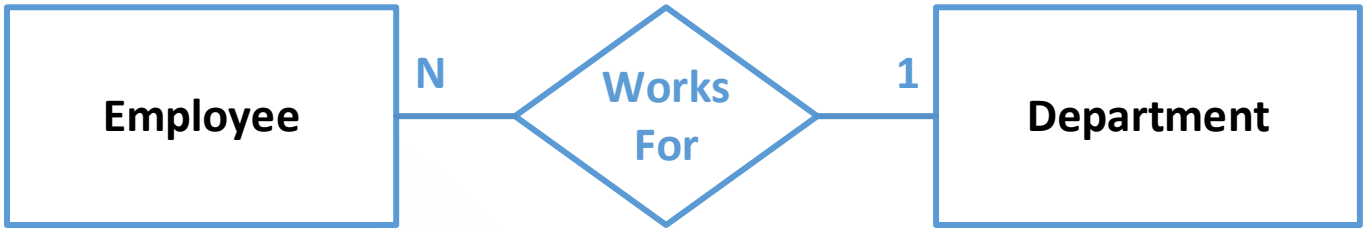
A employee can work in at most one department; A department can have many employees working in it.

More on relationships

- **Participation constraint** :
  - A constraint which specify the minimum number of entities that can be associated with another entity through a meaningful relationship
  - Participation constraint can take two values:
    - Total (or mandatory) participation
    - Partial participation
  - Total participation: If every instance of an entity type must participate in a given relationship, then that entity is having total participation in the relationship
  - Partial participation: If every instance of an entity type need not to participate in a given relationship, then that entity is having partial participation in the relationship

Relationships with cardinality and participation constraints : ER diagrams

- Cardinality in Chen's notation



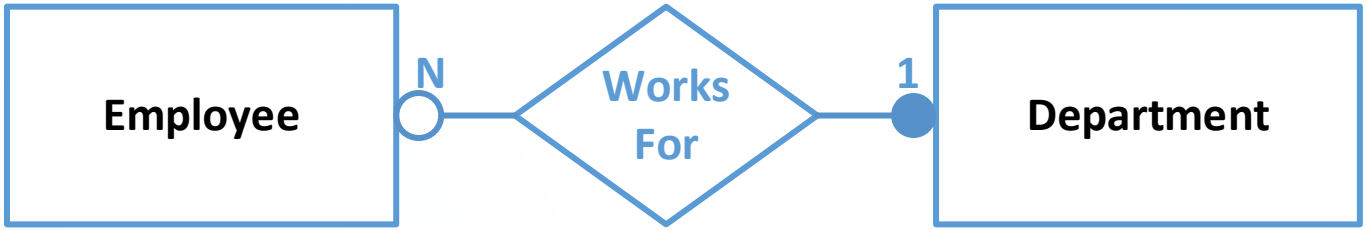
This uses a 'Look Across' notation

Constraint that a one employee works in one department is represented by placing 1 across the relationship WorksFor from Employee entity.

A employee can work in at most one department; A department can have many employees working in it.

Relationships with cardinality and participation constraints : ER diagrams

- Original Chen's notation did not have participation constraints ; added later by others.
- Cardinality and participation in Chen's notation :



One employee can exists without working for a department. (partial participation)

A department cannot exists without at least one employee.(total participation)

Participation uses 'Look Here' notation
Cardinality uses 'Look Across' notation

Example:

- One –one relationship

Cardinality

- One student works in one project ; one project is conducted by one student.

Participation

- Student can exists without project; project cannot exists without a student

Example:

- Many to many relationship

Cardinality

- A customer can purchase many Books; A book is purchased by many customers

Participation

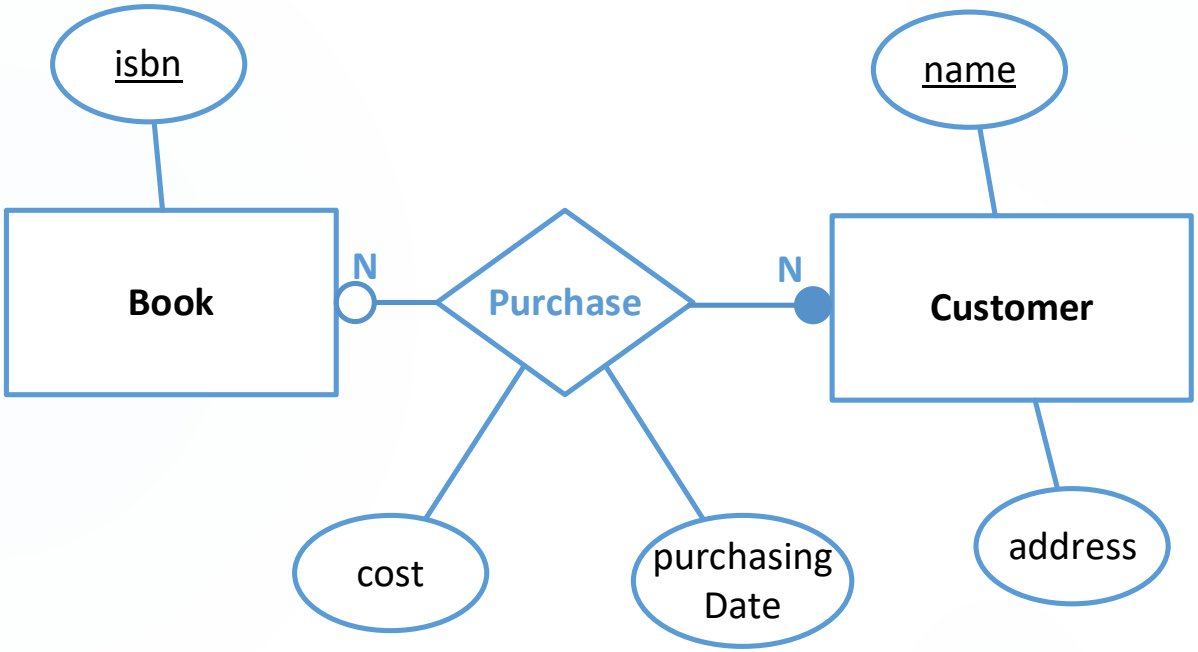
- Customer cannot exists without a book; a book can exists without a customer

Attributes on relationships

- Sometimes it is useful to attach an attribute to a relationship.
- What are the best entity to assign attributes **cost of the actual purchase of a book** and the **date a purchasing occurred**?
- Book, Customer or Purchases?

Attributes on relationships

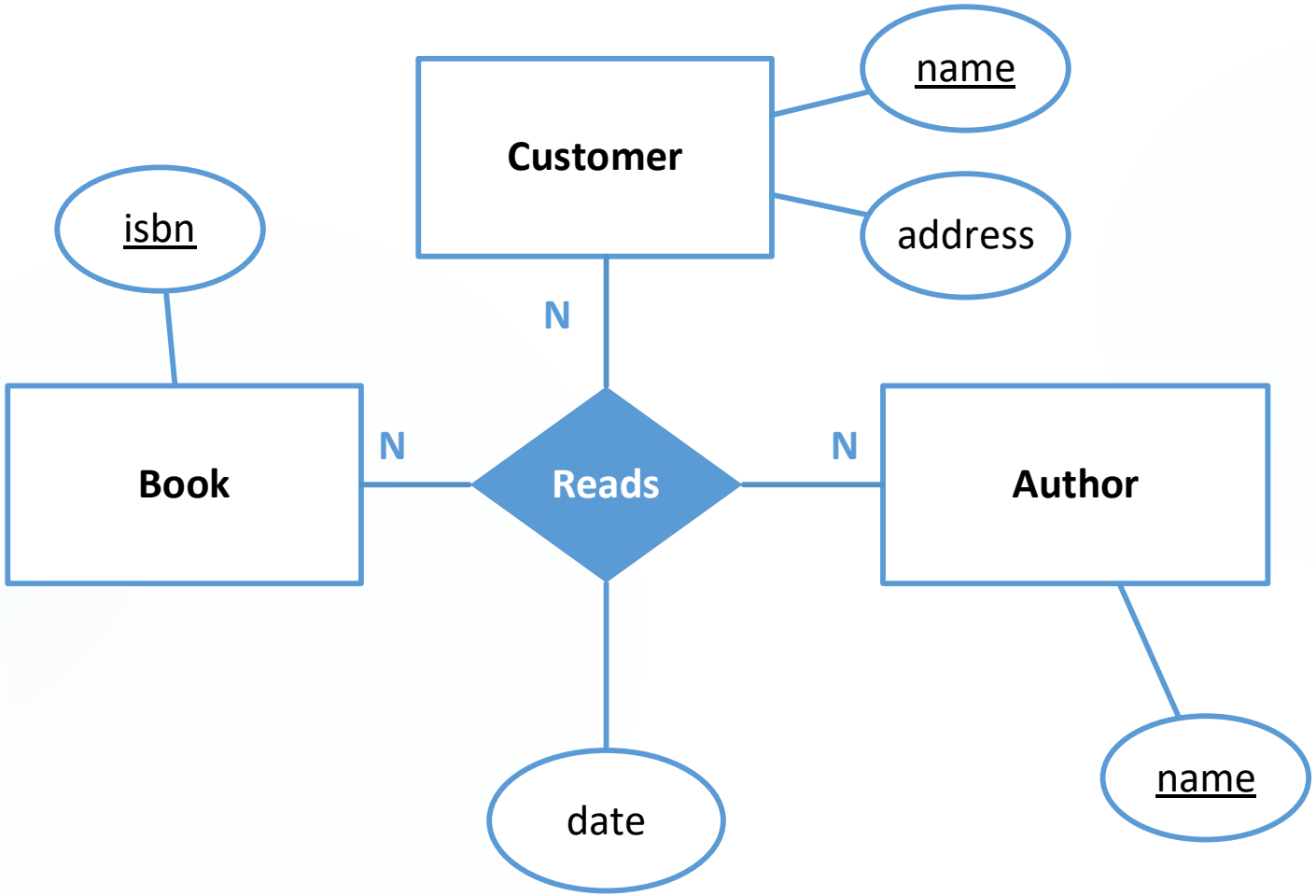
- Each purchase has an associated cost and date. These attributes are related to the purchase itself, not to any one book or customer.



Ternary relationships

- Sometimes, we need a relationship that connects more than two entity sets.
- Suppose that customers will only read certain books by certain authors.
  - Our binary relationships Writes and Purchases do not allow us to make this distinction.
  - But a 3-way relationship would.
- All entity types should occur at the same time.
- Note: Some ER modelling notations does not allow ternary relationships.

Ternary relationships



Weak entity

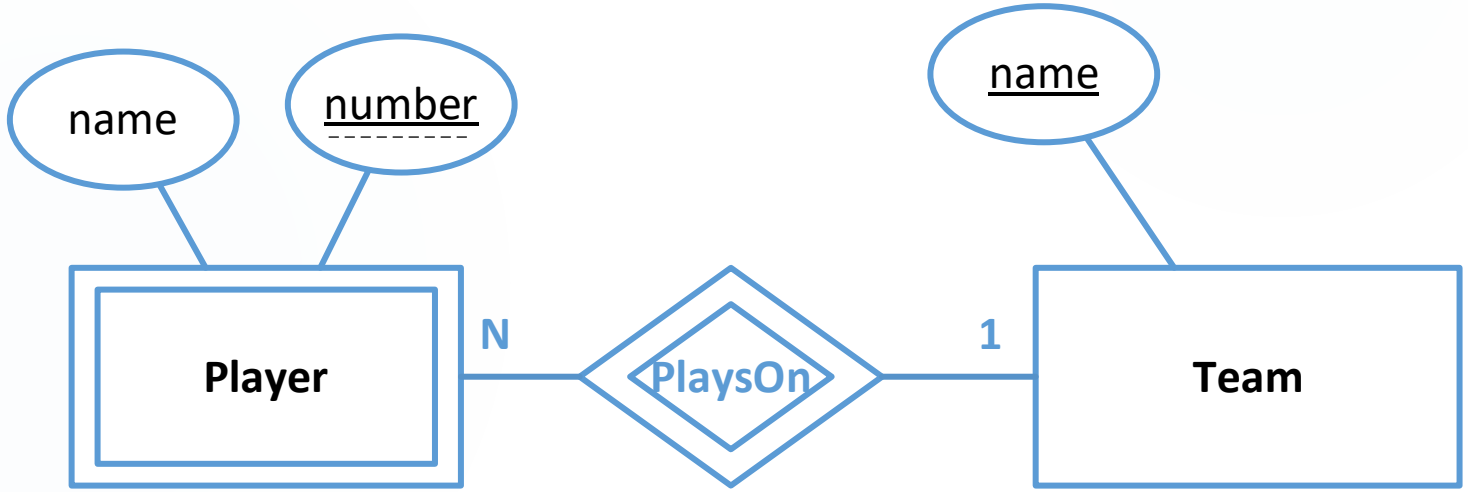
- Occasionally, entities of an entity set need “help” to identify them uniquely.
- An entity set which does not have its own unique key is a weak entity set.
- Entity set *E* is said to be *weak* if in order to identify entities of *E* uniquely, we need to follow one or more many-one relationships from *E* and include the key of the related entities from the connected entity sets.
- Therefore, key of a weak entity is always composite.
- Key of a weak entity : primary key of the owner entity set and a partial key (discriminator) of the weak entity.

Weak entities: Example

- Consider Player and Team entity sets.
- name is almost a key for football players, but there might be two with the same name.
- number is certainly not a key, since players on two teams could have the same number.
- But number, together with the team name related to the player by Plays-on should be unique.

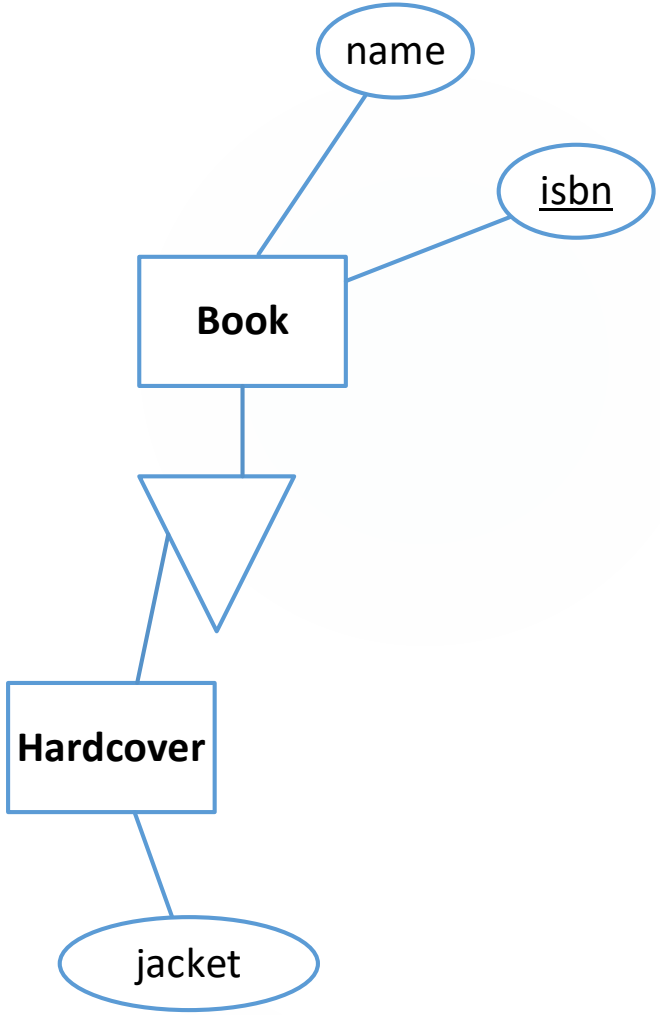
Weak entity : ER diagrams

- A weak entity type is represented by a double rectangle
- A weak relationship by a double diamond
- A discriminator is underlined with a dashed line



Generalization relationships

- *Subclass* = special case = fewer entities = more properties.
- Example: Hardcovers are a kind of Book.
  - Not every book is a hardcover, but some are.
  - Let us suppose that in addition to all the *properties* (attributes and relationships) of books, hardcovers also have the attribute **jacket**.

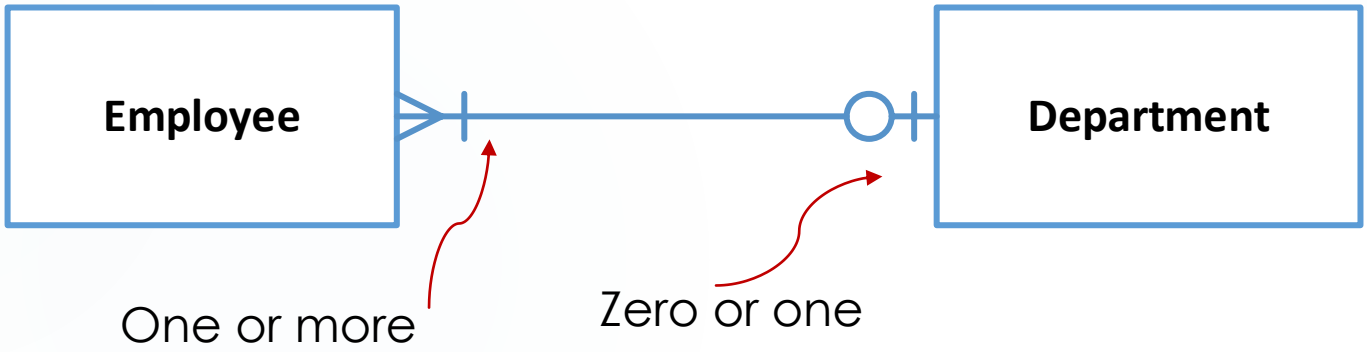


ER diagrams in other notations

- Information engineering (IE) notation is commonly used in CASE tools
- IE notation is a binary notation; Ternary relationships are decomposed to binary relationships
- Min/max and Crow-foot notations are also commonly adopted.

ER diagrams in other notations

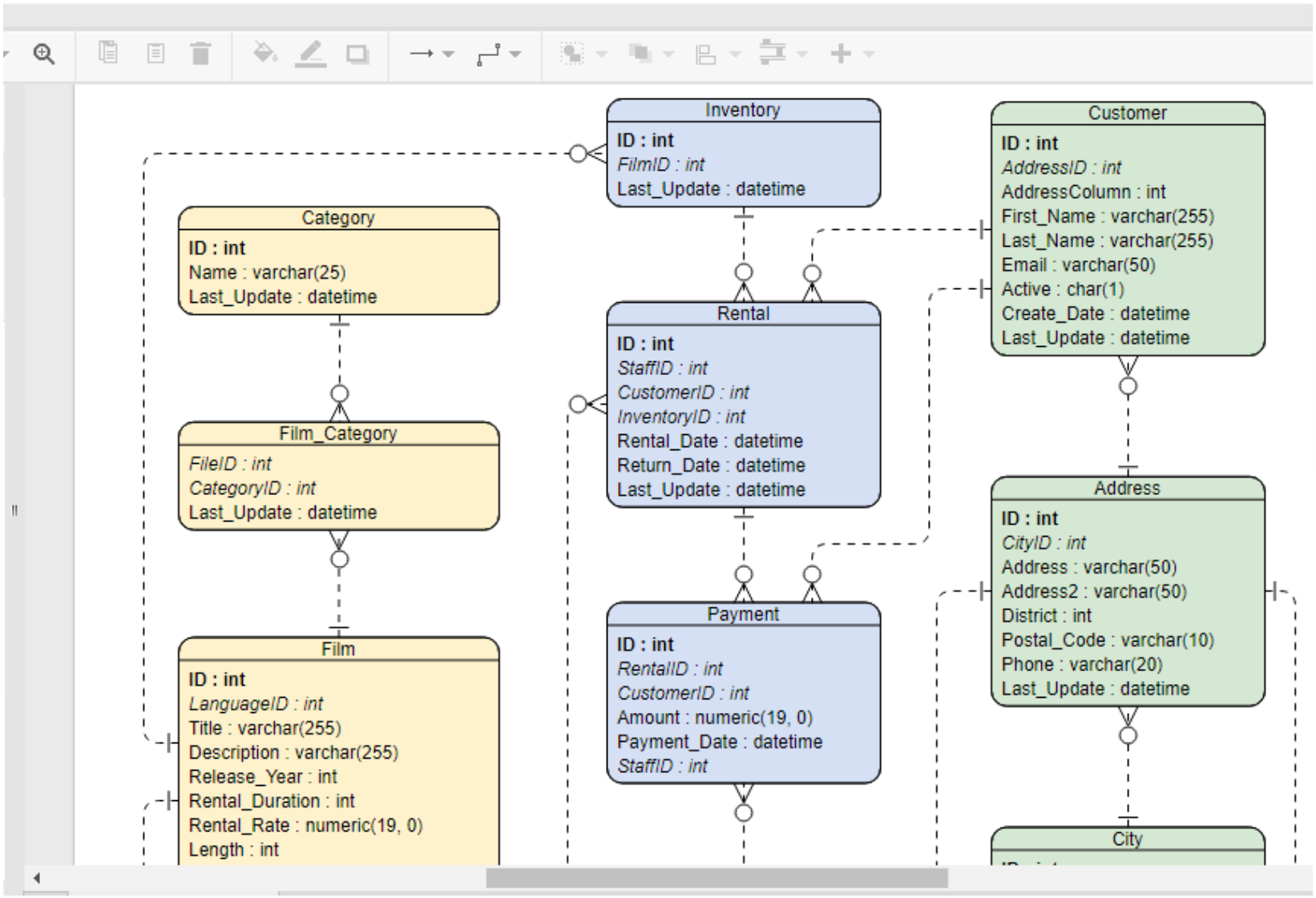
Min/Max notation



IE notation

Both notations does not use diamond for the relationship

Example :


Inventory
ID : int
FilmID : int
Last_Update : datetime
Customer
ID : int
AddressID : int
AddressColumn : int
First_Name : varchar(255)
Last_Name : varchar(255)
Email : varchar(50)
Active : char(1)
Create_Date : datetime
Last_Update : datetime
Category
ID : int
Name : varchar(25)
Last_Update : datetime
Rental
ID : int
StaffID : int
CustomerID : int
InventoryID : int
Rental_Date : datetime
Return_Date : datetime
Last_Update : datetime
Film_Category
FileID : int
CategoryID : int
Last_Update : datetime
Address
ID : int
CityID : int
Address : varchar(50)
Address2 : varchar(50)
District : int
Postal_Code : varchar(10)
Phone : varchar(20)
Last_Update : datetime
Payment
ID : int
RentalID : int
CustomerID : int
Amount : numeric(19, 0)
Payment_Date : datetime
StaffID : int
Film
ID : int
LanguageID : int
Title : varchar(255)
Description : varchar(255)
Release_Year : int
Rental_Duration : int
Rental_Rate : numeric(19, 0)
Length : int
City

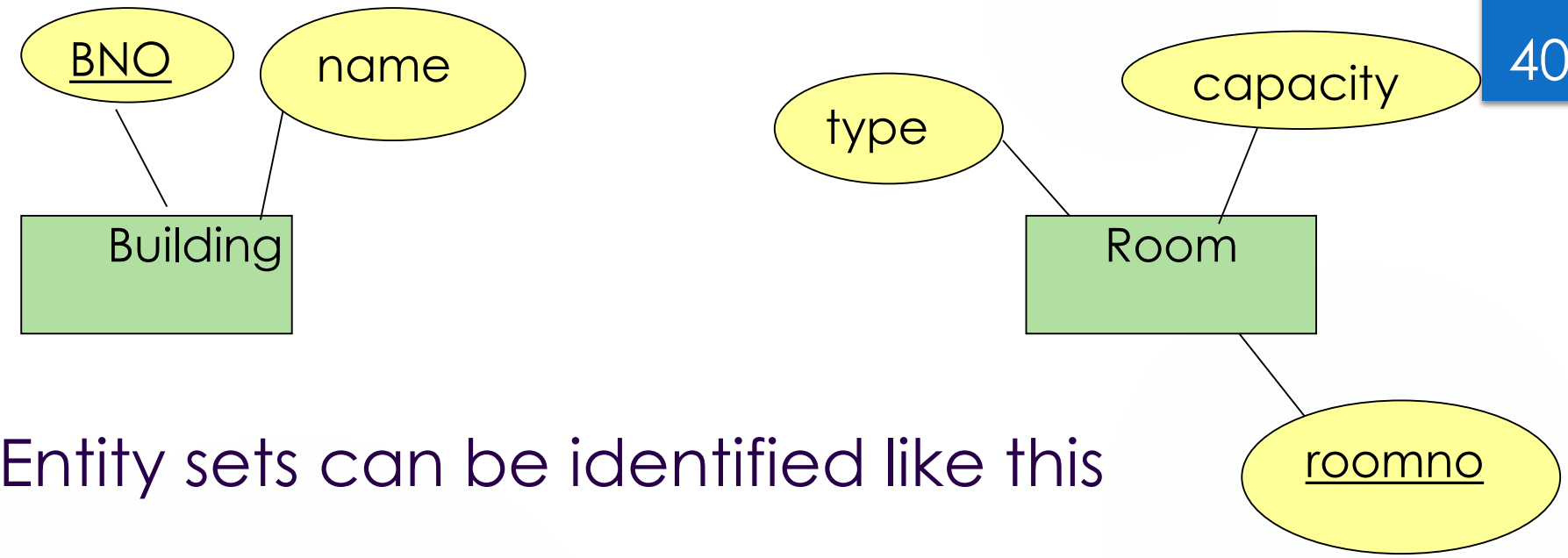
Approach to create ER diagrams

- There can be many correct models to represent a given problem domain.
- Experience and use of good modelling techniques helps to create meaningful ER diagrams
- No diagramming tool can generate ER diagram without your input.
- You have to think to identify entities, attributes and relationships.
- Domain knowledge becomes very useful.
- It is always easier to create the model in paper and transfer using diagramming tools.
- Working in groups or getting feedback from another also helps.

ER Model : Example

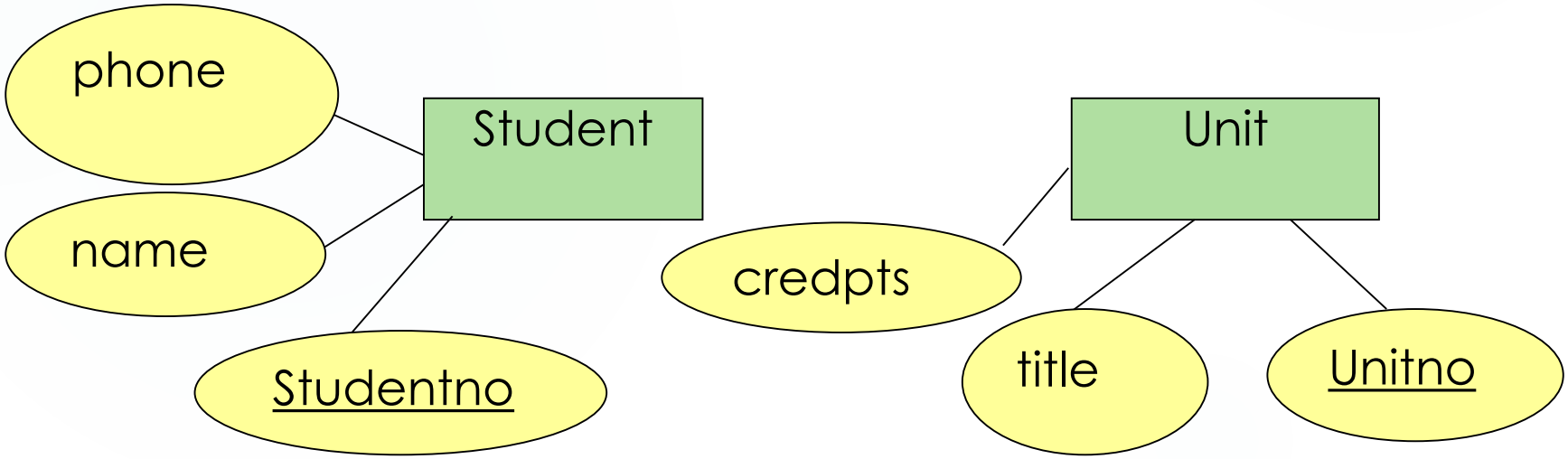
- A University database is required to store current information about students (student number, name, phone number), units (unit number, title, credit points), rooms (room number, type, capacity), and buildings (building number, name).
- Each student is identified by a unique student number, each unit by its unique unit number, each room in a building by a unique room number (but a given room number can be present in many buildings), and each building by a unique building number.
- A student may take a number of units and each unit will have many students. Each unit meets only once a week and in only one room.
- The day of the week and the time for the meeting are to be recorded.

Create the ER diagram for the above example.



Entity sets can be identified like this

Relationships have to be added now…



Summary

- Database design and implementation can be performed in three stages, analysis and data model design, logical database design and physical database design.
- Most of the time, designing a database starts with ER modelling.
- Key components of ER model are entities, attributes and relationships.
- Entities are the objects to model and the attributes describe properties of them.
- Each entity set should have an attribute as an identifier.
- Relationships connects entity sets with meaningful associations.
- Relationships can be one to one, many to many or one to many.
- Cardinality and preferences represent constraints on relationships.
- Different notations are used to draw ER models.

Happy Database systems

Next week : ER to relational schema mapping , Normalization

Practical worksheet – 4