Lecture 9: Conceptual Schema Design using ER model

CSX3006 DATABASE SYSTEMS

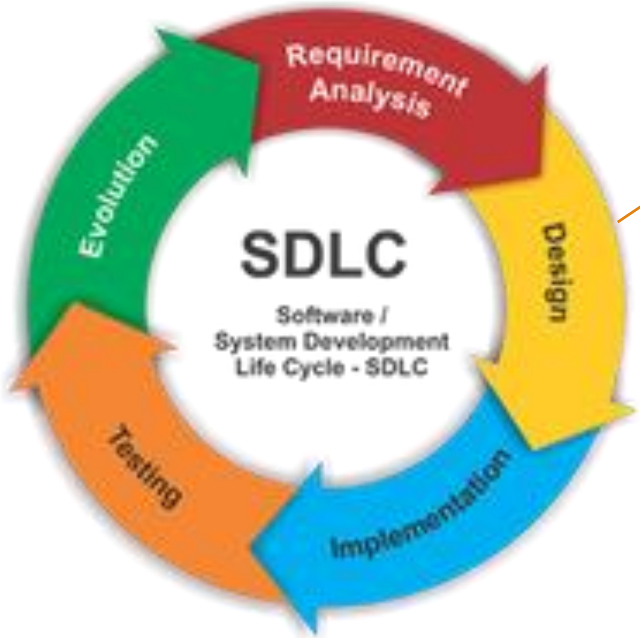
ITX3006 DATABASE MANAGEMENT SYSTEMS

Outline

- Fundamental ER Constructs: Entities, Attributes and Relationships
- Simple vs. Composite Attributes
- Single vs. Multi-valued Attributes
- Derived Attributes
- Descriptive Attributes vs. Key Attributes
- Cardinality Constraints on Relationships
- Participation Constrains
- Demonstration and Exercises on simple ER modeling

System Development Life Cycle (Software Development Process)

- Objectives:
  - Ensure that high quality systems are developed
  - Provide strong management controls over the projects
  - Maximize the productivity of staff involved in the development



Database Design takes place here!

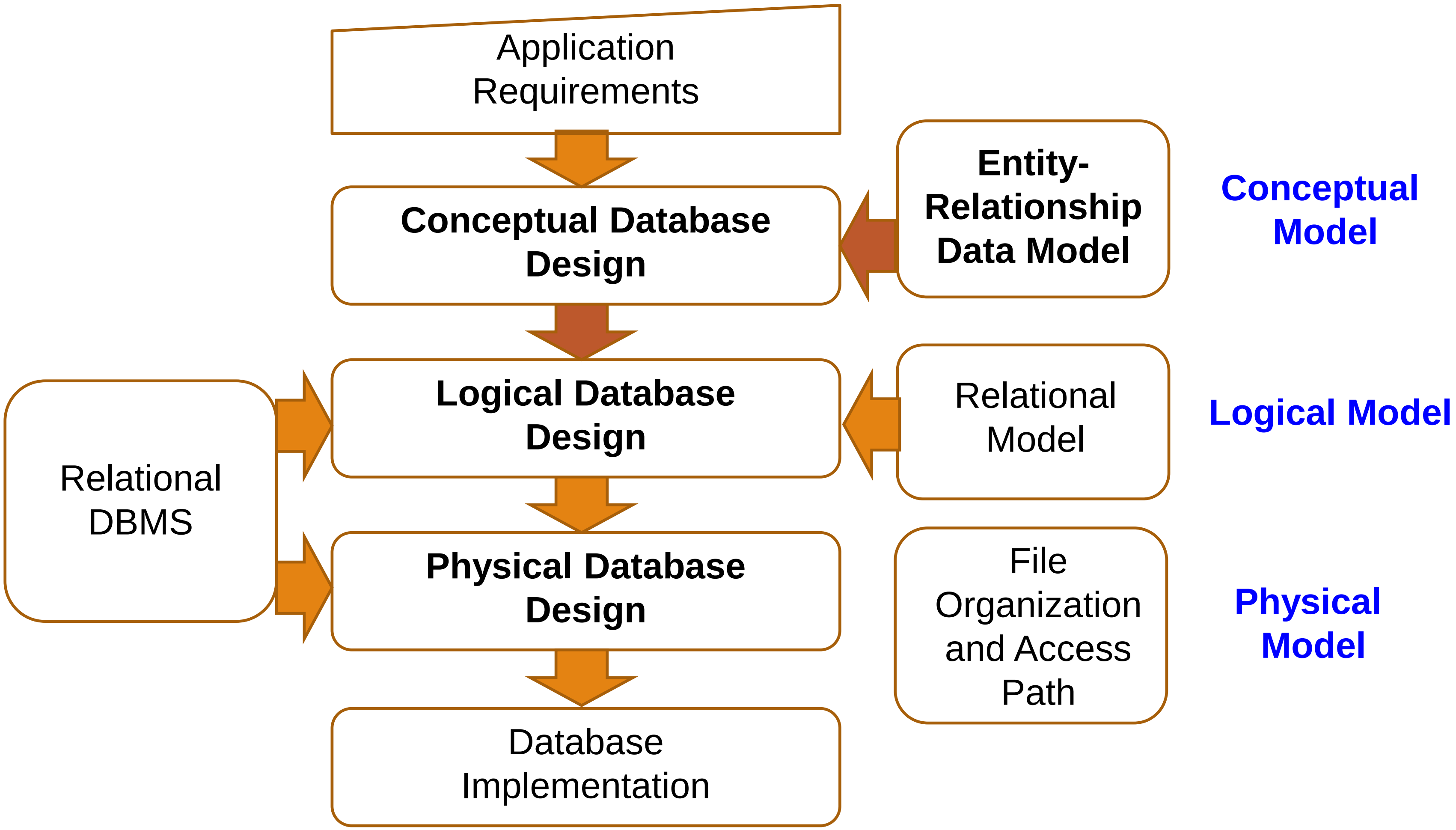
Application and DB Design

- **Database is only a part of overall system** designed to meet the all the requirements of given applications.
  - **Database is created to serve the data needs** of applications!
- **Overall goal** → Develop application programs to support real world tasks.
  - These application programs need **persistently stored data**
- **Database design is the *process of developing a database schema* for a given set of applications:** a sub-part of System Development Process
  - Database design cannot be performed without first clearly understanding the requirements of the application domain (Business Logic and Rules)

Database Design

- **Database Design is NOT simple nor straightforward; can become very complex**
  - It **requires attention to a broad set of issues.**
    - Functional requirements, non-functional requirements, performance requirements, Business Rules, different applications, different users, different goals, etc
  - **Enterprise being modelled is often big and may involve numerous inter-dependent and independent units working together, but**
    - No single person knows every aspect of the enterprise
  - Not only the designer **must learn about the application domain**, but he/she also **needs to build a 'holistic view' of data requirements and 'flow' of data from 'separate pieces and chunks' as seen by each individual units and persons**.
  - There is no single perfect design. There will always be **alternative 'good' designs**; **must be able to evaluate pros and cons of different alternative designs**
- Exception: Small applications; one person does everything (possible)

Big Picture: Design Phases



Three steps of Database Design - 1

- **Conceptual Database Design**
  - Analyze 'the problem', then define what data and information are required by the application programs and users
  - **Build an initial model or 'mini-world' of the business enterprise capturing the requirements and constraints**
  - Need **a common '*language*' to specify the design**
    - e.g.) **ER model**

Three steps of Database Design - 2

- **Logical Database Design**

  - **Transform the Conceptual Design Schema into the database model supported by the chosen DBMS** (often Relational Database)

  - Also **involves checking** the resulting relational schema **for possible data redundancy** (thus data inconsistency and modification anomalies)
    - **The process of Normalization**

  - The relational schema is input into the DBMS (now database comes to exist!)
    - Use SQL DDL (supported by the DBMS) to specify relational schema

Three steps of Database Design - 3

- **Physical Database Design**
  - Aims at improving the performance of the final system
  - Consider typical workloads and further refine the database design
    - **De-normalization**, **INDEXING**, **storage parameters specification**, etc...

Conceptual Design using ER Model

- **Entity Relationship Model**
  - The 'language' used to specify the conceptual design
    - the model depicted by graphical notation: **ER Diagram**
  - Designed to facilitate database design by allowing **specification of 'semantics' of data requirements of enterprise applications at higher abstract level**, which can be then readily transformed into lower logical level schemas
  - Other alternative: **UML Class Diagrams** (for Object Oriented Design)

ER Model - 1

- View of a real world as **Entities** and **Relationships** among Entities
  - **What are the entities in the business enterprise being modelled?**
  - **What are the relationships among these entities?**
  - **What data (attributes) about these entities and relationships should we store in the database?**
  - **What are the 'business rules' (integrity constraints) that hold?**

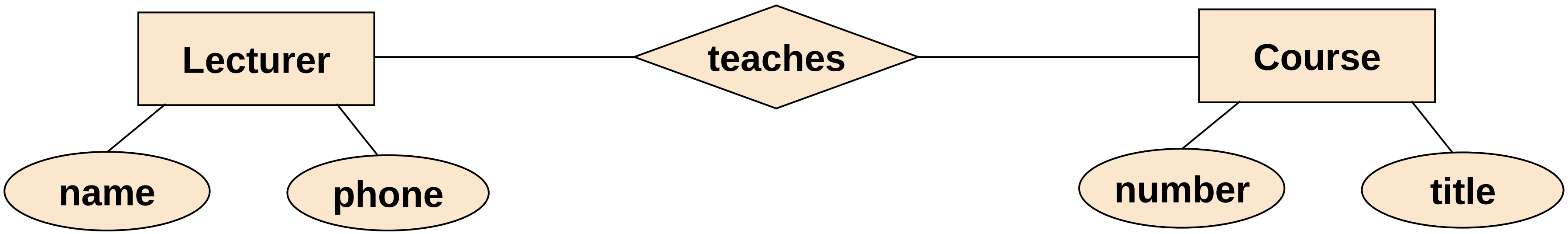
A **business rule** is a statement that defines or constrains some aspect of the business.
= *integrity constraint* in the database (to maintain valid data values and relationship in the database.)

ER Model - 2

- **ER Model helps to capture the semantics (meanings) of data that need to be maintained in the enterprise**
  - The **meanings** and **interactions** of real-world enterprises are mapped onto a conceptual schema graphically using three basic notations:
    - **Entities, Relationships and Attributes**

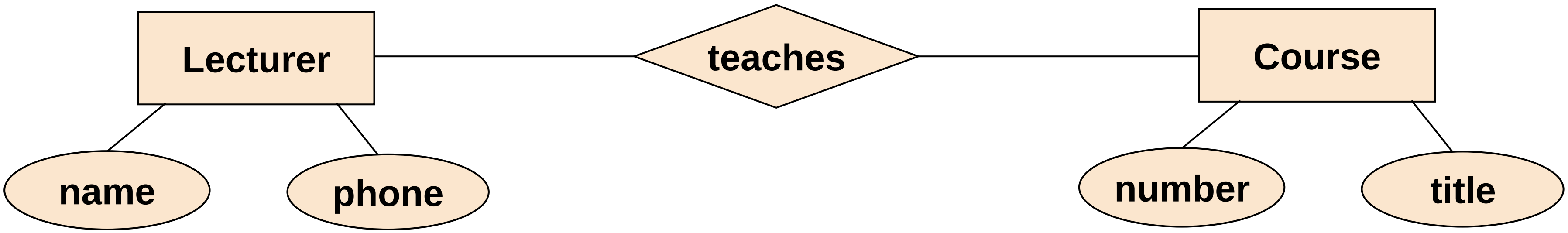
- **ER Model → Semantic data model;**
  - It more closely resembles the real-world than a relational model
    - In ER model, there is distinction between entities and relationships
    - In relational model, both are represented by relations (tables)
      - In fact, a relational database is a collection of relations (tables)

Simple ER Model Example - 1



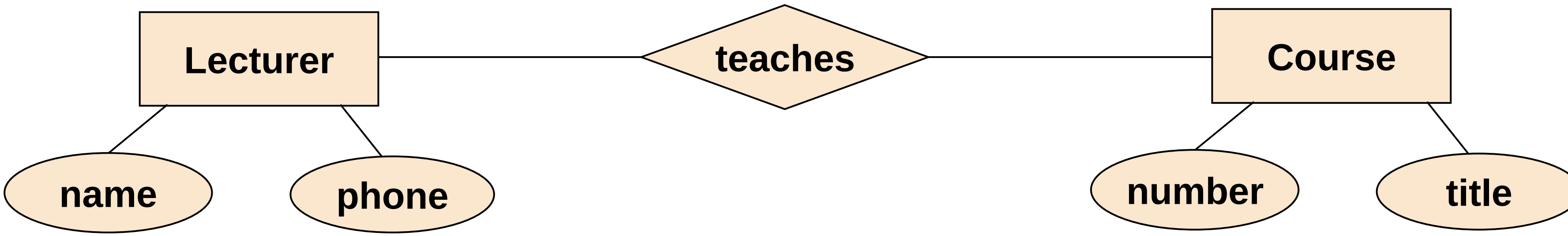
- This conceptual model is a representation of an 'enterprise' which contains Lecturers and Courses.
- It also captures the fact that Lecturer '*teaches*' Courses.
- The semantics of the enterprise is represented by the three basic notations
  - **Entities**: Lecturer, Course
  - **Relationship**: teaches
  - **Attributes**: name, phone, number, title

Simple ER Model Example - 2



- Lecturers **have** a name and a phone number
  - e.g.) Rachsuda Setthawong, 0891234567
- Courses **have** course number and title
  - e.g.) SC3423, Data Mining
- Lectures **teach** courses
  - e.g.) Rachsuda Setthawong teaches SC3423

Simple ER Model Example - 3



- ER Model also allows to capture ‘***certain***’ business rules (Integrity Constraints) pertaining to the enterprise
  - Can we have two lectures with the same name?
  - Can a lecturer have two or more phone numbers?
  - Can a lecturer teach two or more courses?
    - Or is a lecturer allowed to teach only one course?
  - Is there any upper limit to number of courses a lecture can teach?
  - Can a course be taught by more than one lecturer?

Basic ER Constructs
Entity - 1

- **An Entity (instance)**: a real-world object distinguishable from other objects
  - Can be concrete (physical); e.g) specific person, a company, a book, etc.
  - Can be abstract (conceptual); e.g) a loan, a holiday, award, etc.
- We are interested in entities in the enterprise about which data has to be maintained

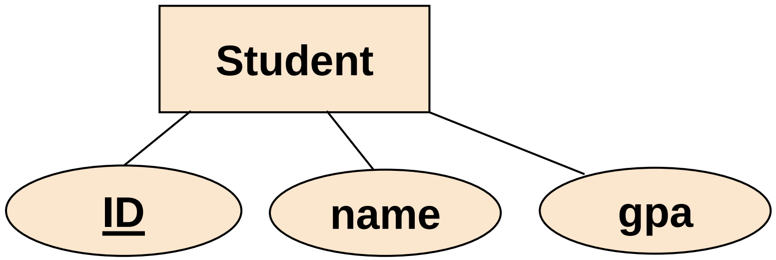
Basic ER Constructs
Entity - 2

- **An entity has a set of properties, known as attributes**
  - e.g) Lecturer has a name, and a phone number
  - **an entity is described by values of its attributes**
- **It must be possible to distinguish one entity from other entities**
  - **Entities must have unique identity** → accomplished via **a key**
    - A student entity may be uniquely identified by his/her ID number
    - A loan entity may be uniquely identified by its number and branch

Basic ER Constructs
Entity Set (or Entity Type)

- **Entity set** **(or entity type)**: A collection of 'similar' entities that share common properties or characteristics.
  - **All entities in an entity set have the same set of attributes**
    - e.g) A set of all Lecturer entities; a set of all loan entities
  - An entity in an entity set must be **distinguishable** from other entities in the set
    - Identified by the value of a **key** attribute in the entity set

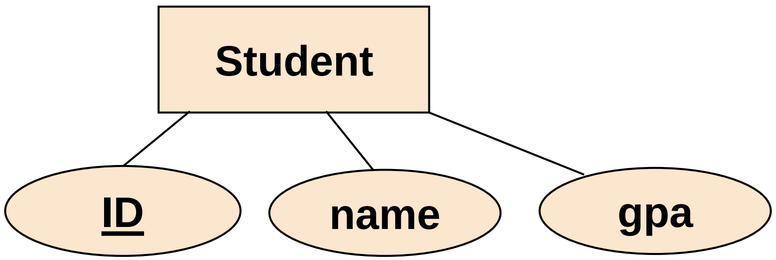
Entity Set vs Entity (Instance)



ID	name	gpa
4911234	Ryan Giggs	3.24
4923425	Wayne Rooney	2.86
4927654	Paul Scholes	2.95
5012346	Wes Brown	2.54

- **Entity Set**: student
- **Attributes**: ID, name, gpa; ID is the key
- A student entity is uniquely ***identified by the value of its ID*** *and* ***described by the values of descriptive attributes***
  - **An entity** (instance) : 4911234, Ryian Giggs, 3.24

Entity Set vs Entity (Instance) – *Cont.*



ID	name	gpa
4911234	Ryan Giggs	3.24
4923425	Wayne Rooney	2.86
4927654	Paul Scholes	2.95
5012346	Wes Brown	2.54

- Analogy
  - An entity (instance) is similar to an object in the sense of OOP
  - An entity set is similar to the class of objects
    - A class may have attributes and methods
    - Entity set only has attributes

Basic ER Constructs Attributes

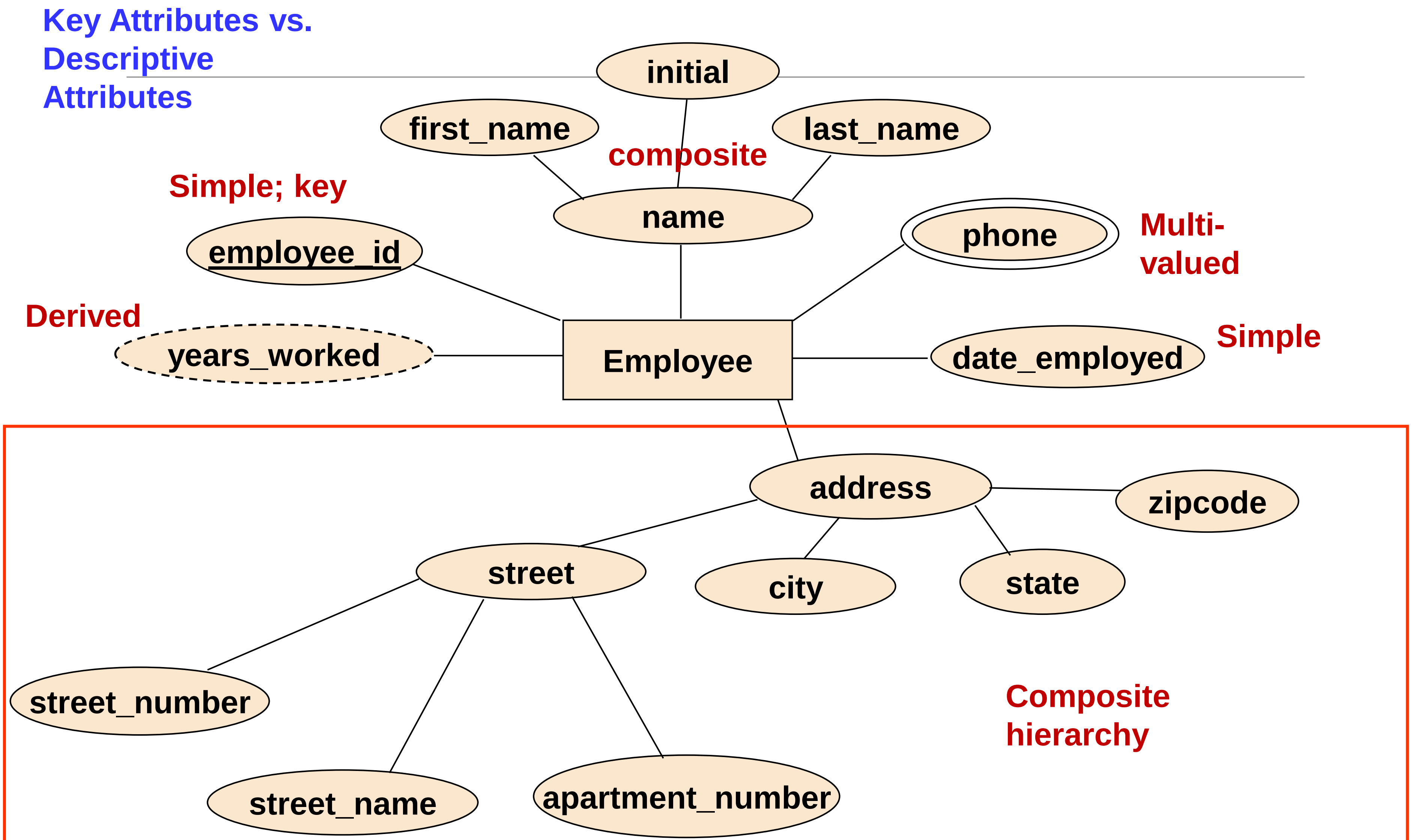
- **An Entity Set has associated Attributes** – properties of the entities in the set
  - e.g.) students have ID, name, gpa, and phone numbers
- **Attributes of an entity set characterizes the entities in the set**
  - Entities are described by the values of the attributes

- **Domain**: the set of permitted values for each attribute
  - e.g.)
    - 0.0 <= gpa <= 4.0;
    - ID must be seven digit number

Attribute Types in ER Model

- Simple vs. Composite Attribute
  - **Simple Attribute**: ID, name, gpa (**treated as Atomic value**)
  - **Composite Attribute**: name → first_name, middle_initial, last_name
- Single-valued vs. Multi-valued Attribute
  - **Single-valued attribute**: ID, gpa, name, first_name, last_name
  - **Multi-valued attribute**: A student may have multiple phone numbers
- **Derived Attribute**: values of this attribute can be 'derived' from the values of other related attributes or entities
  - e.g.) Given a birth_date attribute, we can derive the value of age

Example Different Attribute Types

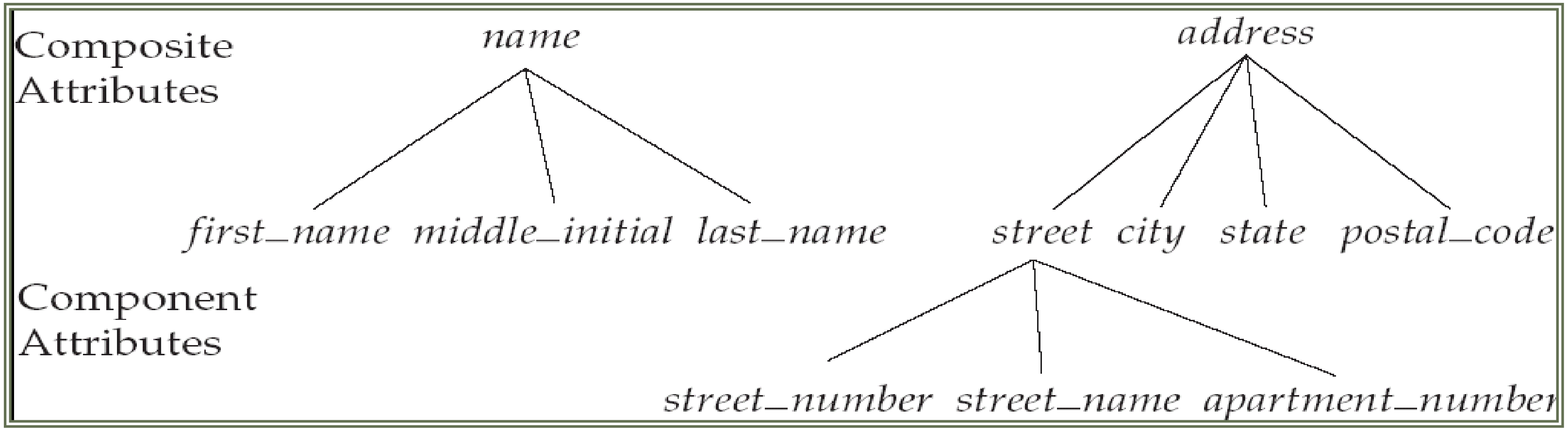


Remarks on Attributes - 1

- **Domains of attributes in *ER model*: Conceptual Domain**
  - **Domain of attributes in *logical model***: need to be specified with data types supported by a given DBMS
- **Attributes in *ER Model* can be simple, composite and/or multi-valued**
  - Remember **attributes in *relational model* must be simple atomic types**
  - **When we transform ER schema into relational schema, multi-valued and composite attribute types have to be converted to simple types**
    - We will learn how to transform later

Remarks on Attributes - 2

- We can also form 'a hierarchy' of composite attributes



- **We can also intermix multi-valued and composite attributes together**
  - e.g.) we may have multiple addresses, each of which can be a composite attribute

Remarks on Attributes - 3

- Should we represent a student's name as a 'simple' or 'composite' attribute?
  - Simple Attribute: name
  - Composite Attribute: name → first_name, middle_initial, last_name

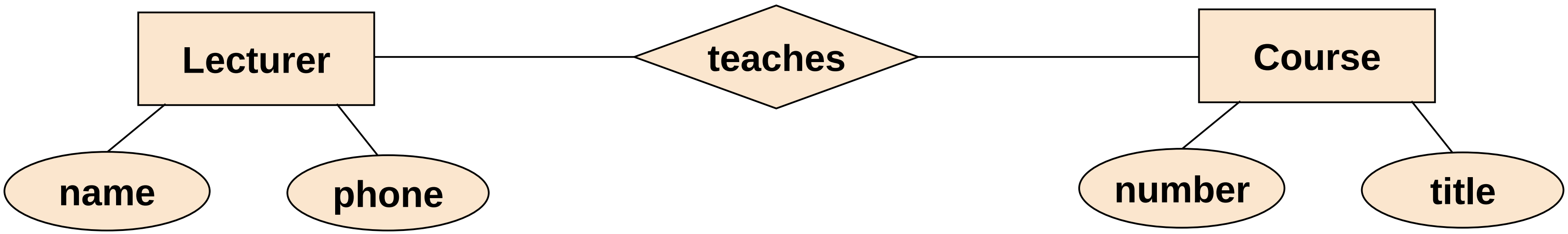
- What should be the base of your decision in a design alternatives like this?
  - Business Logic, Business Rules, Functional Requirements, Required Reports, etc...

Remarks on Attributes - 4

- How do we determine if an attribute needs to be simple or composite?
  - Is your student ID number simple or composite?

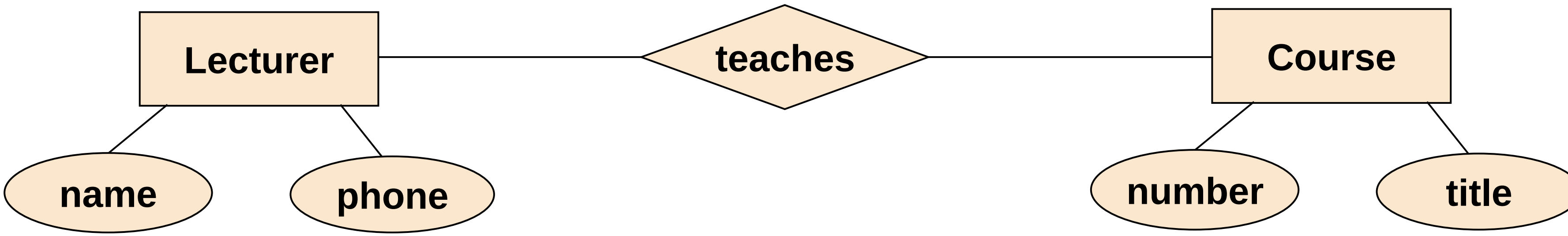
- Consider the following attribute for representing a part serial number.
  - SE-29-174232 (of course, it will be represented as char(12))
    - Is this a simple attribute or composite attribute?
- What if?
  - SE → Region Name
  - 29 → Factory Number
  - 174232 → Part number

Basic ER Constructs
Relationship



- **A Relationship (instance):** an association among two (binary) or more entities
  - e.g) **Lecturer** with the name ‘Rachsuda Setthawong’ and a phone number of 0891234567 **teaches** a **course** number CS3423 titled ‘Data Mining’

Basic ER Constructs
Relationship Set (Relationship Type)



- **Relationship set (Relationship Type)**: a set of relationships of the same type
- Example: **'teaches'** associates entities in the set 'Lecture' with entities in the set 'Course'
- **A relationship (instance) : a particular association**
  - 'Rachsuda Setthawong' teaches CS3423
    - Entities are related together through a relationship <u>via the key attributes of the entities</u> that are associated together

Formal Definitions

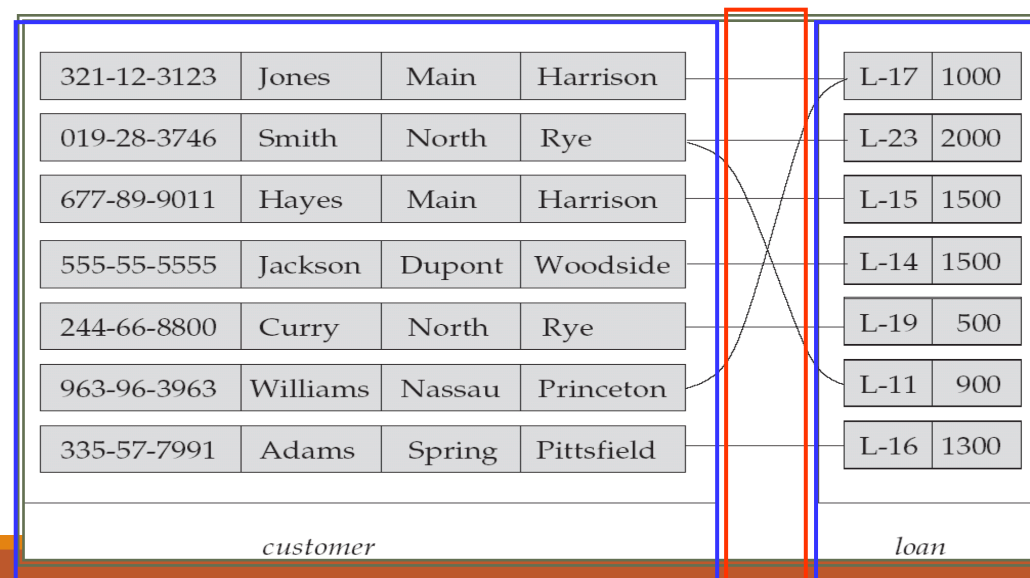
- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

  $$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

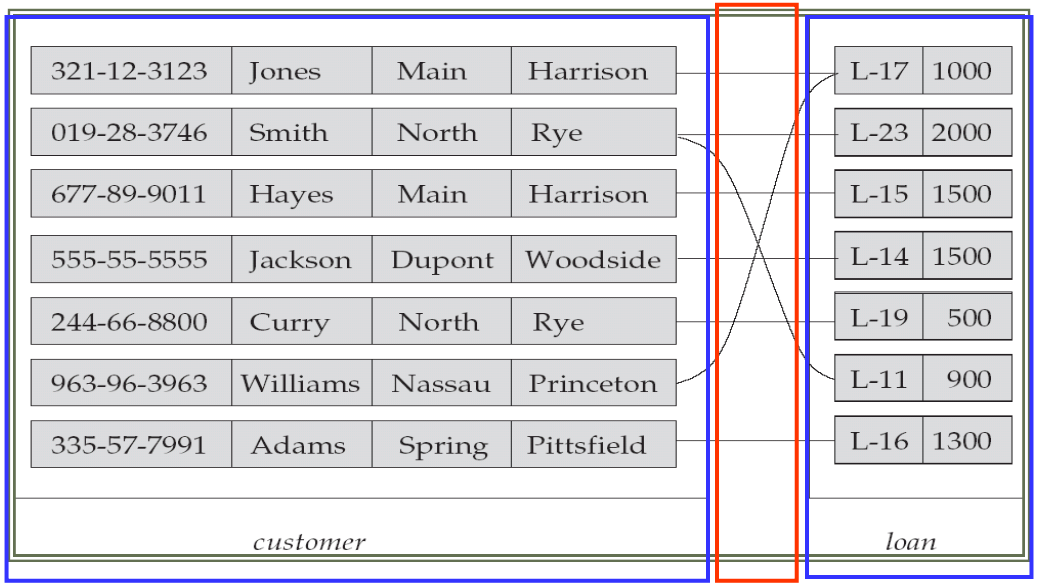
  where $(e_1, e_2, \ldots, e_n)$ is a relationship

  - A **relationship (instance)** is an association among several entities (entity instances)
  - Example:

    (Hayes, L-15) $\in$ *borrower*



Answer the Following Questions

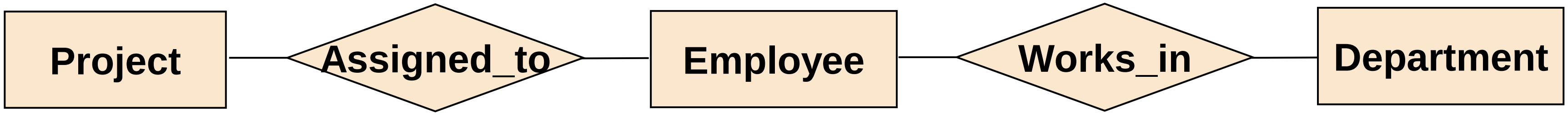


- How many entities in the ‘customer’ entity set currently?
- How many entities in the ‘loan’ entity set currently?
- How many relationships in the ‘borrower’ relationship set currently?

Terminologies are used loosely in many literatures

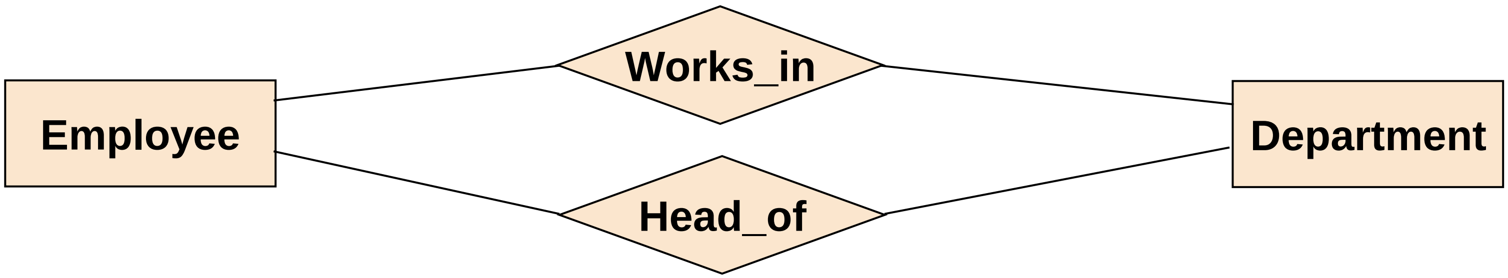
- Entity ; entity set ;
- Relationship; Relationship set

Relationship Set Examples - 1



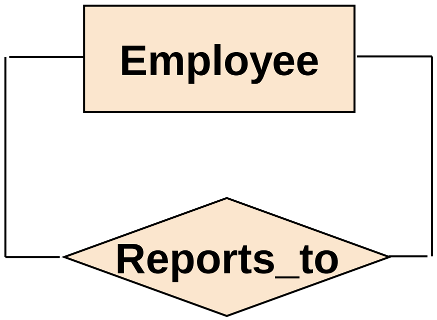
- An entity set can participate in more than one relationship set
  - ‘Employee’ participates in ‘Works_in’ and ‘Assigned_to’

Relationship Set Examples - 2



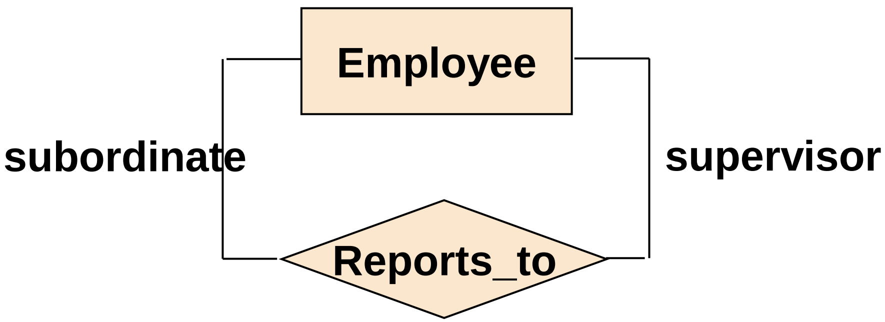
- The same two entities sets can participate in two different relationship sets
  - ‘Works_in’ and ‘head_of’ are two separate relationship sets

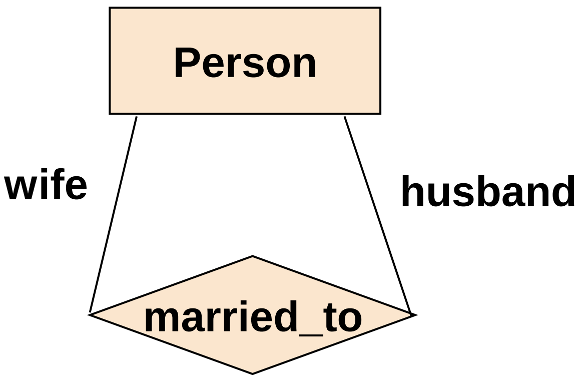
Relationship Set Examples - 3



- The same entity set can participate in a relationship set more than once.
  - **recursive relationship set**
- In such cases, we will **need to specify the entity's role in the relationship**

Role Specification - 1

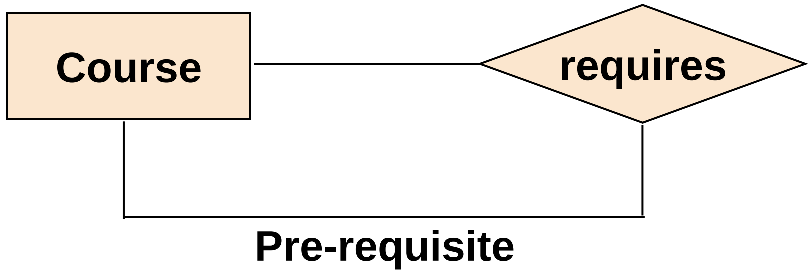




Same entity set can participate in different 'roles' in the same relationship set.

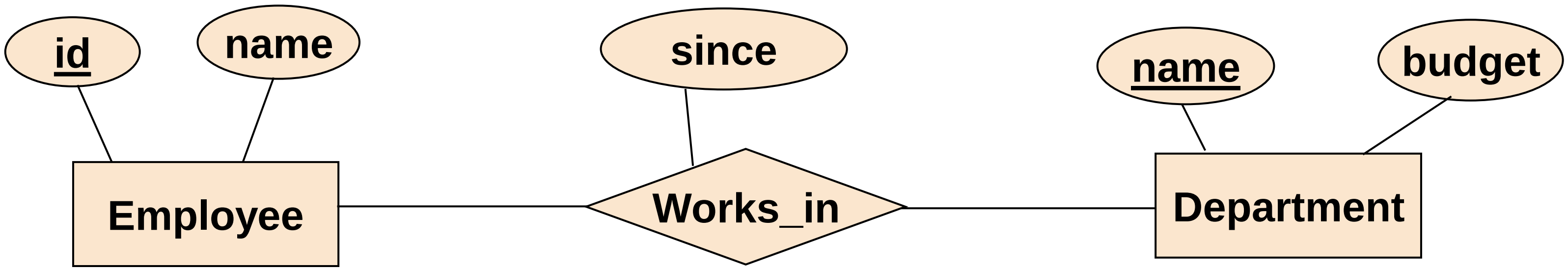
- Employee entity set participates in the relationship set 'Reports_to' as a role of supervisor and as a role of subordinate

Role Specification - 2



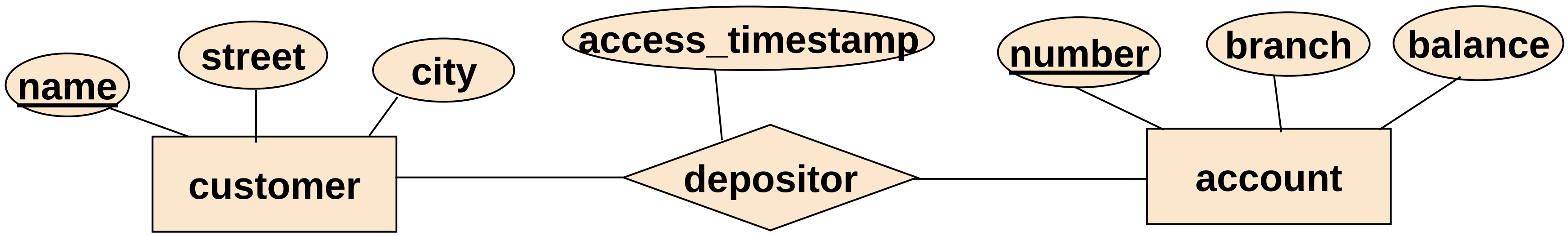
- **When the role is clear even without being specified, it can be omitted.**
- In some books, this kind of relationship set is known as
- **Unary Relationship set** (since only **one** entity set **participates**)
- **Recursive Relationship set** (since an entity set **relates to itself**)

Descriptive Attributes - 1



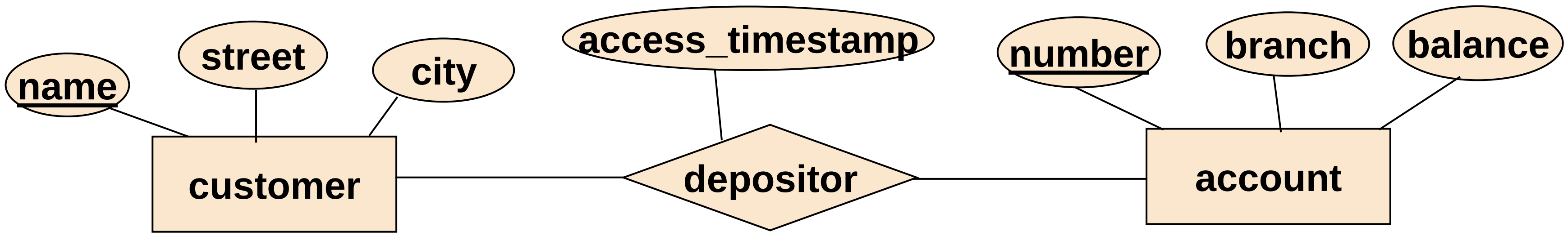
- **A relationship set may have attributes (but cannot have key attributes)**
  - e.g.) the attribute '**since**' records the date an employee started working in a department
- **A relationship instance** in a given relationship set **must be uniquely identifiable** from its participating entities, **without using any descriptive attributes** of the relationship set (NO key attributes)
  - In the above example, we **cannot** have an employee working in a department with two different since dates
    - (12345, marketing, 12/07/07), (12345, marketing, 15/02/14)

Descriptive Attributes - 2



- access_timestamp is used to keep log of when a customer accesses his account
- What can be the limitation above?

Descriptive Attributes - 3



- access_timestamp is used to keep log of when a customer accesses his account
- What can be the limitation above?
  - Can only keep a record of the **LAST time** customer **accesses** the account
    - **A relationship instance** in a given relationship set **must be uniquely identifiable** from its participating entities, **without using any descriptive attributes** of the relationship set

Degree of a Relationship Set - 1

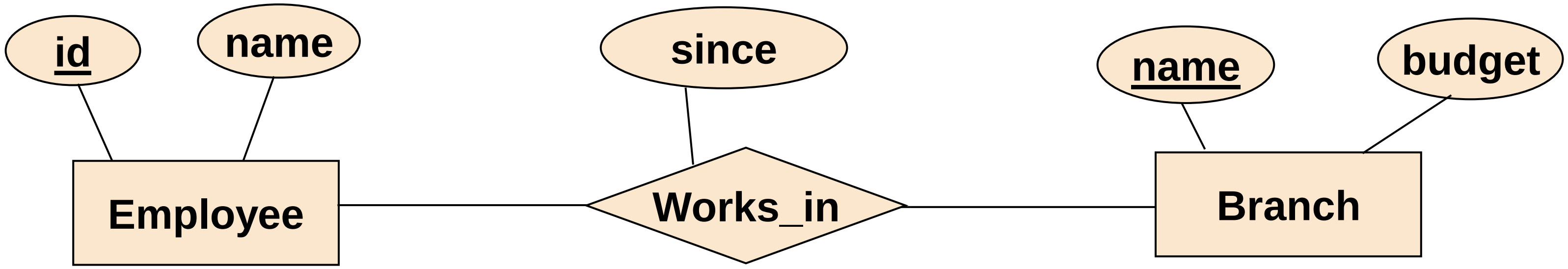
- So far, we have only seen relationship sets that associates two entity sets.
  - **Binary Relationship sets**
- A relationship set **may involve more than two** entity sets
- **Degree of a Relationship Set**
  - Number of Entity Sets that participate in a relationship set
    - **Unary** → degree of one → One Entity Set → Recursive Relationship set
    - **Binary** → **degree of two** → Two Entity Sets are associated
    - Ternary → degree of three → Three Entity Sets are related
    - Degree of N

Degree of a Relationship Set - 2

- In practice, **most relationship sets in a database system are binary;** relationships between more than two entity sets are rare.
  - There also is a standard **procedure for decomposing a higher degree relationship set into multiple binary relationship sets**

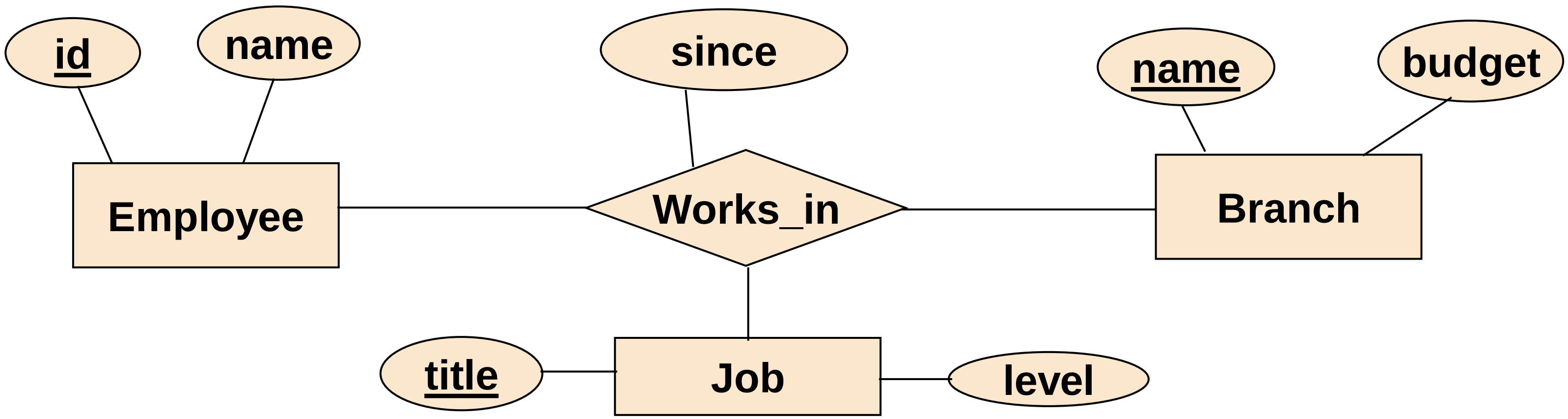
- Having said that, there are situations where higher degree (especially ternary) relationship sets can be useful and where their meaning cannot be translated into multiple binary relationship sets (more on this later)

Degree of a Relationship Set - 3



Binary Relationship set between Employee and Department

Degree of a Relationship Set - 4



Ternary Relationship set among Employee, Department and Job

- Somchai acts as a manager at the Huamark branch.
- Somchai also acts an auditor at the Silom branch
- Brown is a teller at Siam branch

Summary: Basics of ER Model - 1

- **Conceptual Database Design using ER Model**
  - **Captures the <u>semantics of data</u> in an enterprise**
  - View of the world as **Entities** and **Relationship** among Entities
  - **Allows certain Integrity Constraints (Business Rules) to be specified**
  - **Basic Constructs:**
    - **Attributes**: **Simple, Composite, Multi-valued, Derived**
      - Issues: **Domain of Attributes**, **Key Attributes vs. Descriptive Attributes**
    - **Entity and Entity Sets**
      - Uniqueness of Entity in an Entity Set
    - **Relationship and Relationship Sets**
      - **Can have only Descriptive attributes**
      - Uniqueness of Relationship instance in a Relationship Set
        - *Must not depend on descriptive attributes*
      - **Degree of a relationship set**
        - *Unary (Recursive), Binary, Ternary, N-ary*
      - **Roles of Entities in a Relationship**

Summary: Basics of ER Model - 2

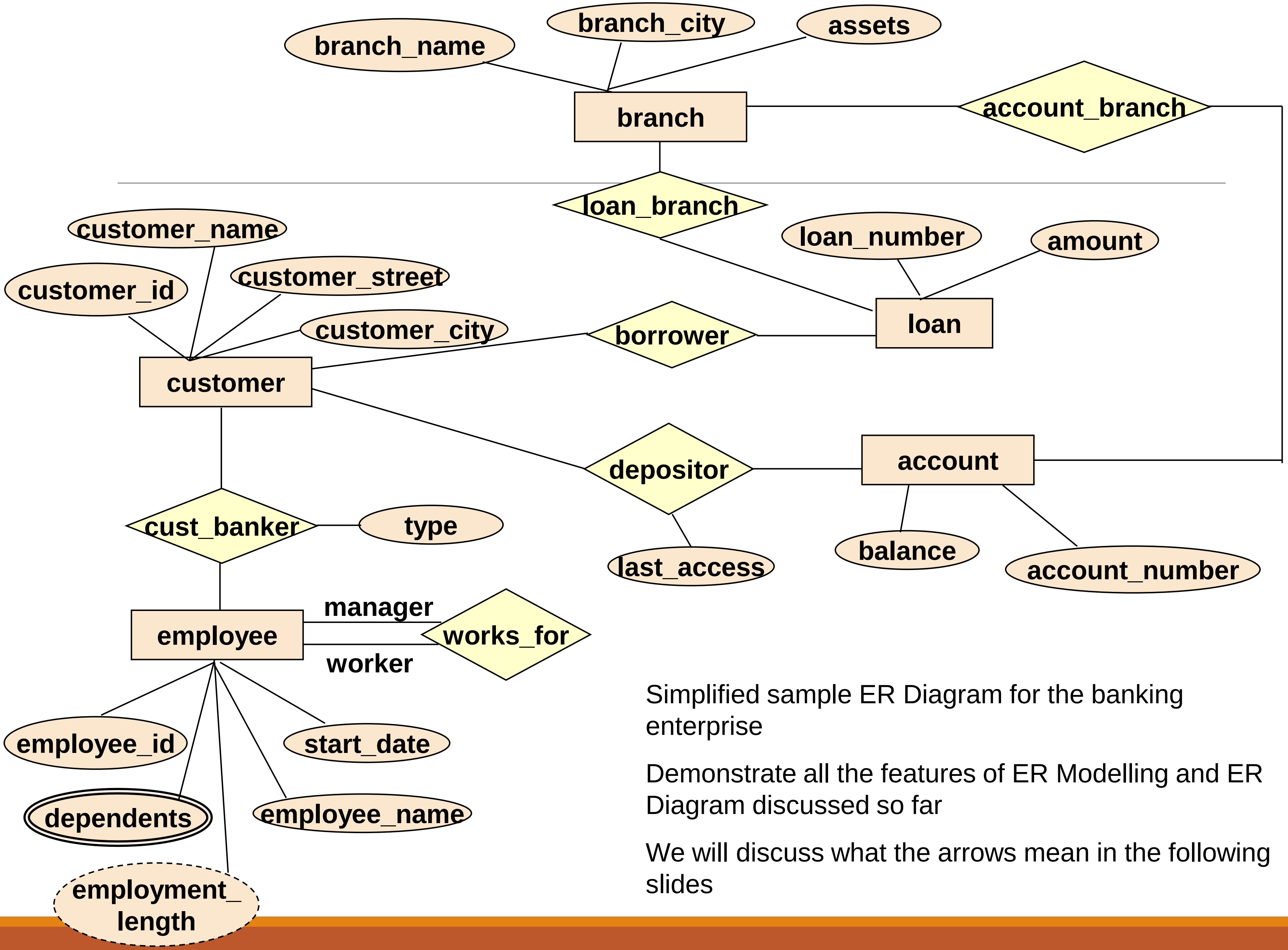
- **Rectangles** represent **entity sets**.
- **Diamonds** represent **relationship sets**.
- **Ellipses** represent **attributes**
  - **Double ellipses** represent **multivalued attributes**.
  - **Dashed (dotted) ellipses** denote **derived attributes**.
- **Lines** link the basic notational symbols
  - Attributes (Ellipses) to Entities (Rectangles) and to Relationship (Diamonds)
  - Entities (Rectangles) to Relationships (Diamonds)
  - Component attributes (Ellipses) to Composite Attributes (Ellipses)

Summary: Basics of ER Model - 3

- **Rectangles, Diamonds, and Ellipses are each labeled with a name**
  - The name is written inside the symbols
  - **Underline indicates primary key attributes in an Ellipse**
  - **The names of rectangles must be unique in the entire diagram**
  - **The names of ellipses must be unique for a single rectangle or diamond to which they are attached**
  - **Diamonds must be uniquely identified by name and their connections to the rectangles**

Workshop 9-1

- Refer to **Supplementary Material: Data Requirements for the Bank Database** (DataRequirementsBankDatabase.pdf)
  - **Create an ER model for the given scenario**



Simplified sample ER Diagram for the banking enterprise

Demonstrate all the features of ER Modelling and ER Diagram discussed so far

We will discuss what the arrows mean in the following slides

Specification of Constraints in ER

- **ER model provides constraints specification through**
  - **Keys**
    - a way **to specify how entities** within a given set of entity set **are distinguished**
    - Also **identifies a unique relationship instance** in a relationship set
  - **Mapping Cardinalities** (Cardinality Ratio)
    - A way **to specify how many entities** can be **associated with another** entity **in a relationship set**
  - **Participation Constraints**
    - A way **to specify** if **every entity** in an Entity set **needs to participate** in **at least one relationship** in a relationship set.
      - **Total vs. Partial**

Keys - 1

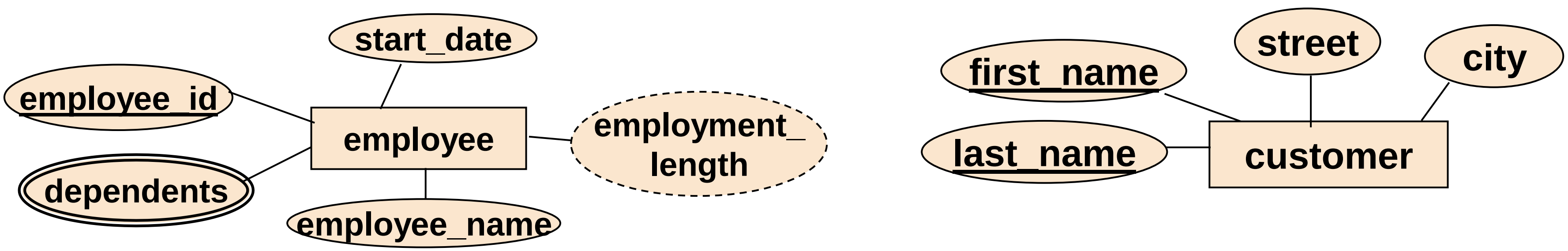
- **Conceptually, individual entities (objects) are distinct**
  - Paul and Colin are both students of CS department, but they are two different individuals
- **This individuality (uniqueness)** of entities **must be expressed in terms of their attributes** when we model the real world to be stored in a database
- **A key** allows us to identify a set of attributes that **suffice to distinguish entities** from each other.
- We already defined the notions of Keys in the context of Relational Model
  - We now define corresponding notions of keys for **ER Model**

Keys - 2

- A **super key** of an entity set is a set of one or more attributes whose values **uniquely determine** each entity.
- A **candidate key** of an entity set is a **minimal** super key
  - *Customer_id* is candidate key of *customer*
  - *account_number* is candidate key of *account*
- Although several candidate keys may exist, one of the candidate keys is **selected** to be the **primary key**.

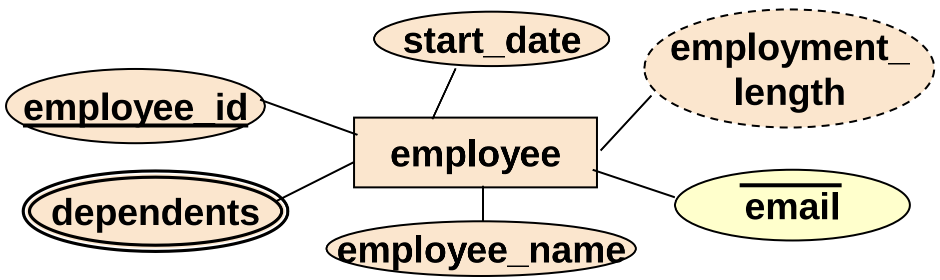
Keys - 3

- In ER diagram, **a primary key** is denoted by **underlining** the attributes which form the key



Keys - 4

- How do we represent **candidate keys** which are not selected as primary key?
  - Standard ER notation only permits the specification of primary keys
    - If there are other candidate keys, this need to be described in English



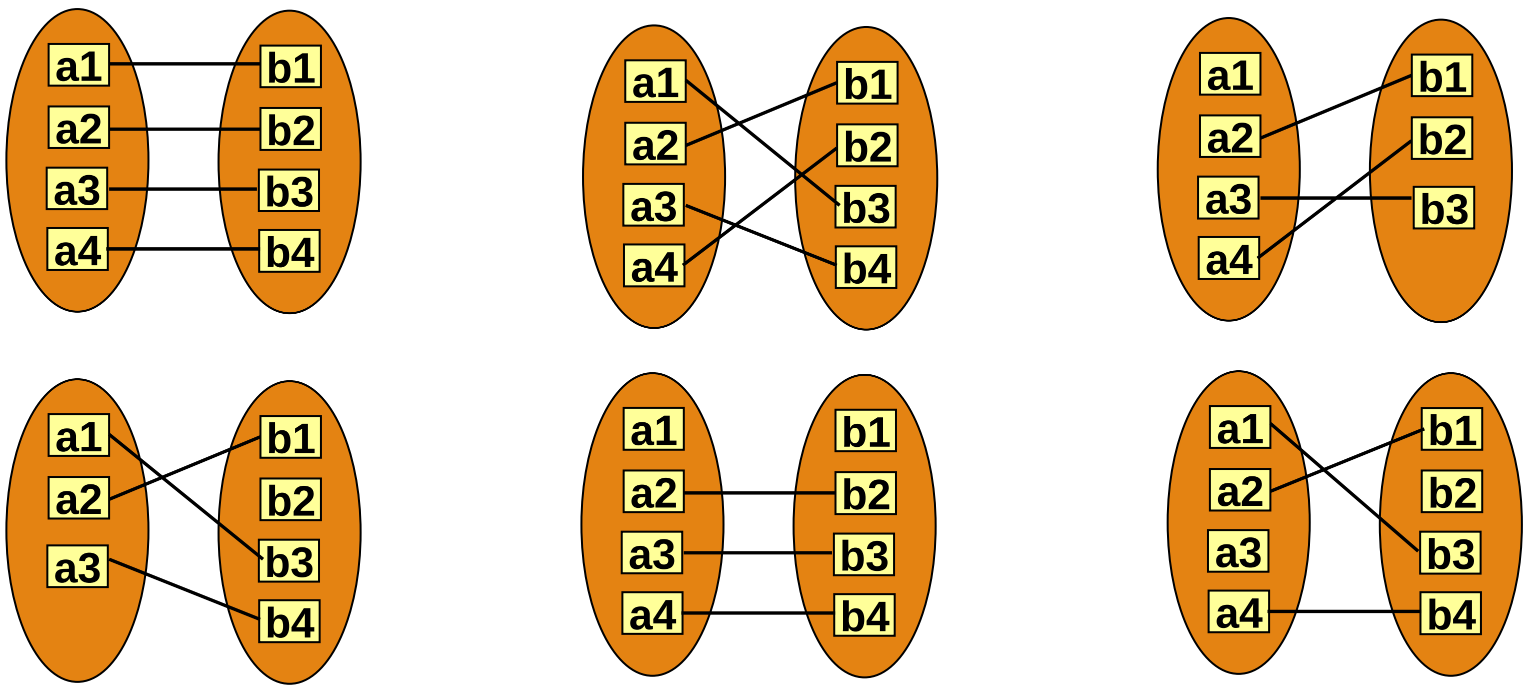
    - Some people use "**overline**" to denote other candidate keys, but this practice is not standard!

Mapping Cardinality Constraints – 1

- Express the number of entities to which another entity can be associated with via a relationship set.
  - How many times **must/may** an entity instance participate?
- For a binary relationship set the mapping cardinality must be one of the following types:
  - **One-to-One**
  - **One-to-Many**
  - **Many-to-One**
  - **Many-to-Many**

Mapping Cardinality Constraints - 2

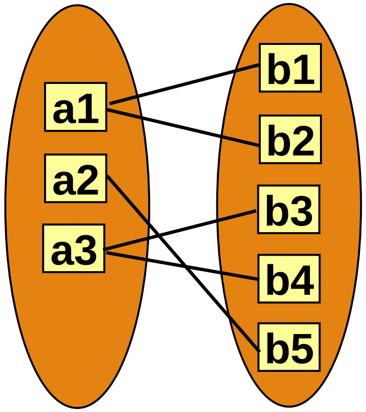
- **One-to-One**
  - An entity in the **left** set is **associated with *at most one*** entity in the **right** set; and an entity in the **right** set is **associated with *at most one*** entity in the **left** set

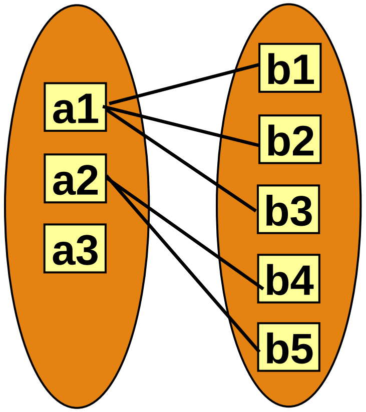


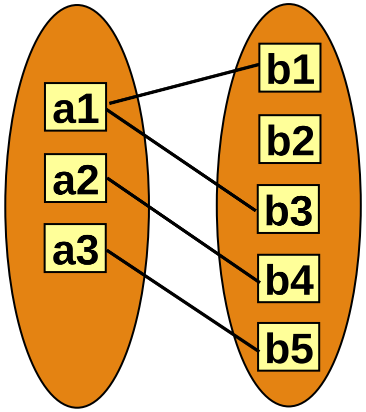
- **Note that an entity may or may not participate in a relationship**
- This is an issue of Participation Constraints

Mapping Cardinality Constraints - 3

- **One-to-Many**
  - An entity in the **left** set is **associated with** ***any number (zero or more)*** of entities in the **right** set. However, an entity in the **right** set can be **associated with** ***at most one*** entity in the **left** set.

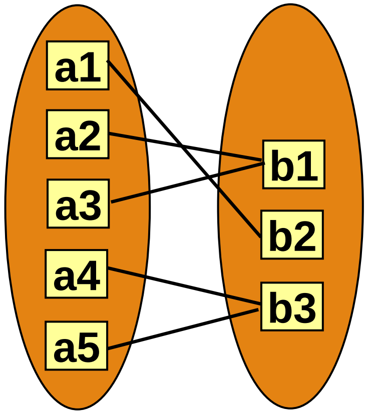


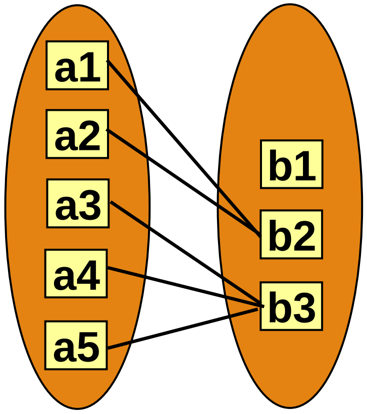


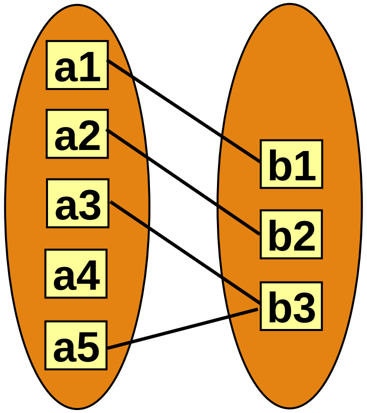


Mapping Cardinality Constraints - 4

- **Many-to-One**
  - Reverse of One-to-Many

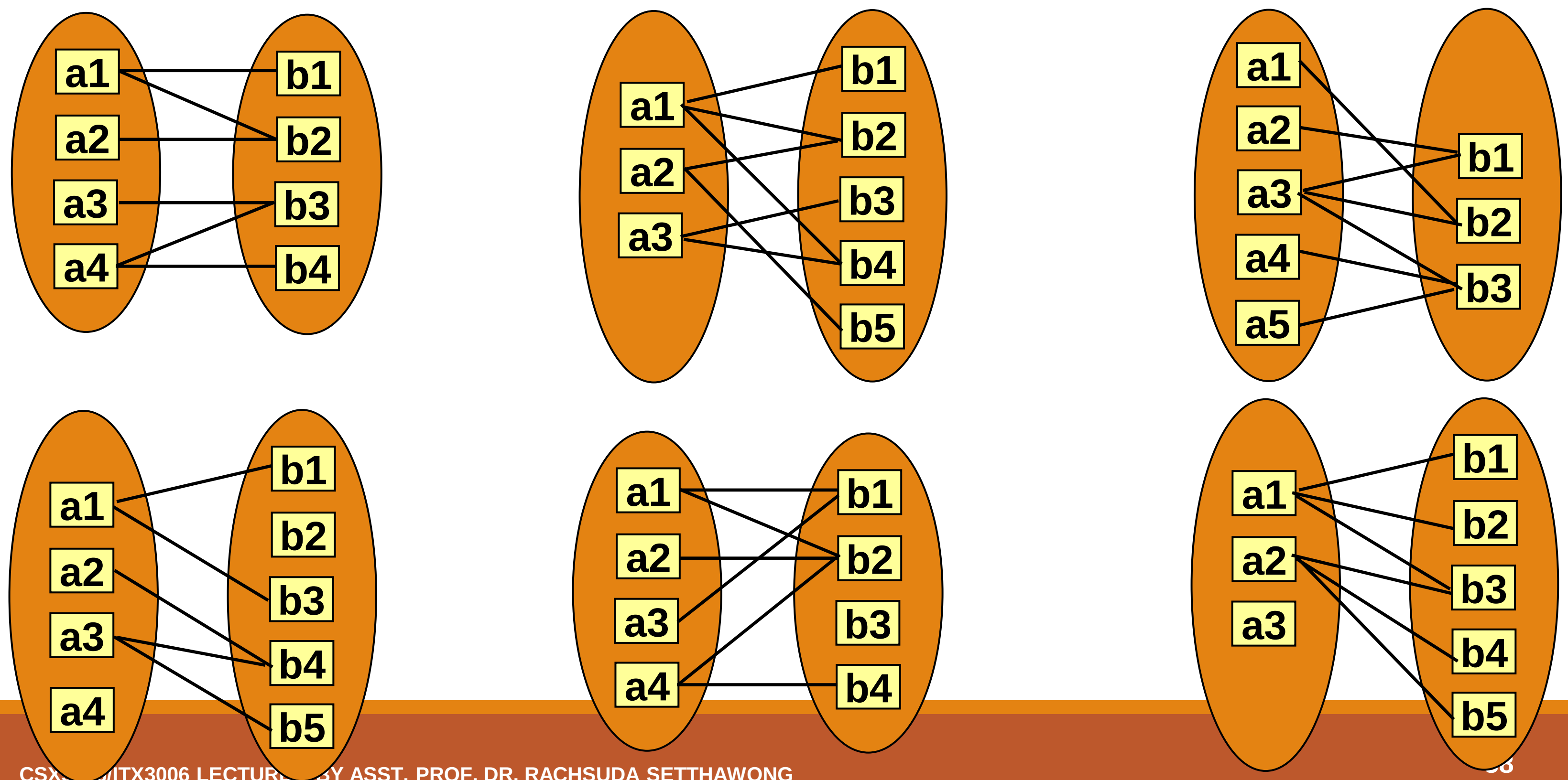




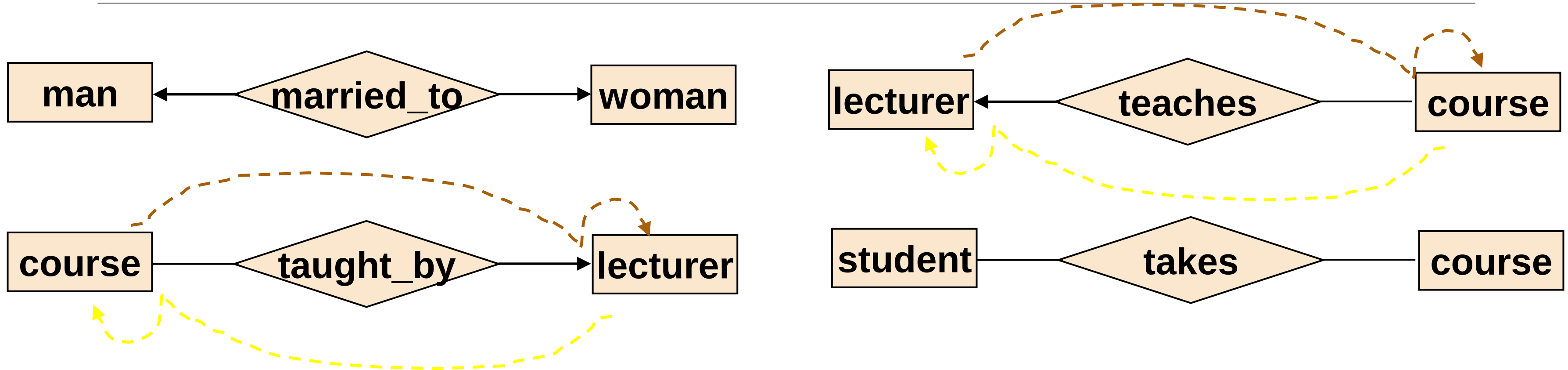


Mapping Cardinality Constraints - 5

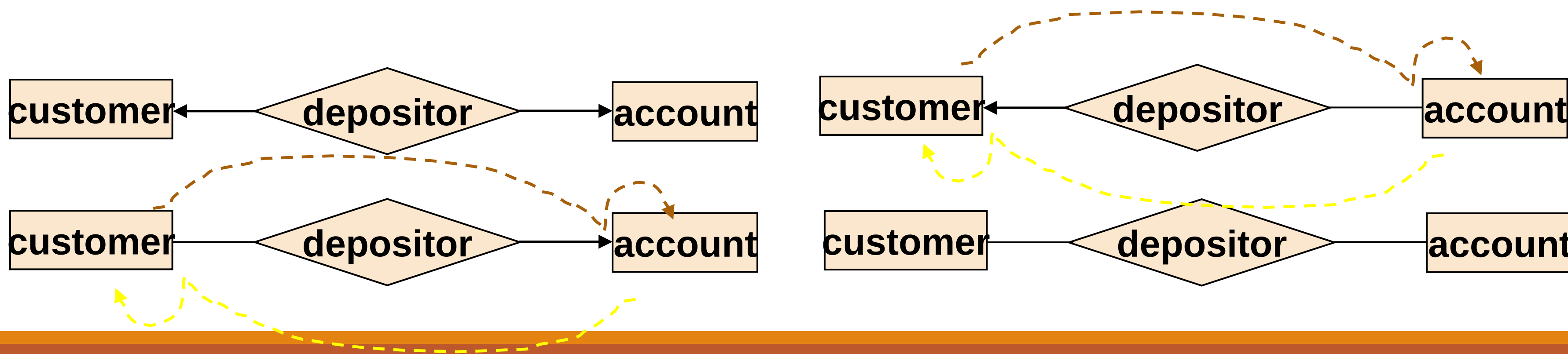
- **Many-to-Many**
  - An entity in the **left** set is **associated with** ***any number (zero or more)*** of entities in the **right** set, and an entity in the **right** set is **associated with** ***any number (zero or more)*** of entities in the **left** set.



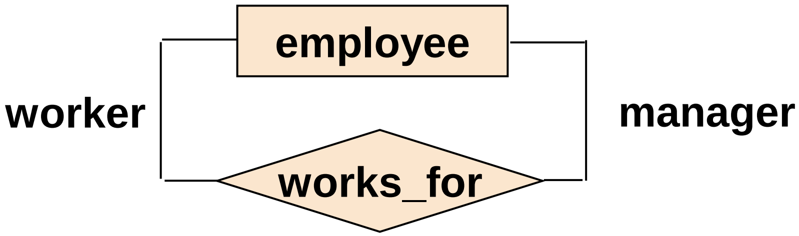
Mapping Cardinalities - 1



- **Semantics of the "arrow"** notation
  - Arrow means ***at most one***
  - It **does not guarantee the total participation** of entity set pointed to



Mapping Cardinalities - 2



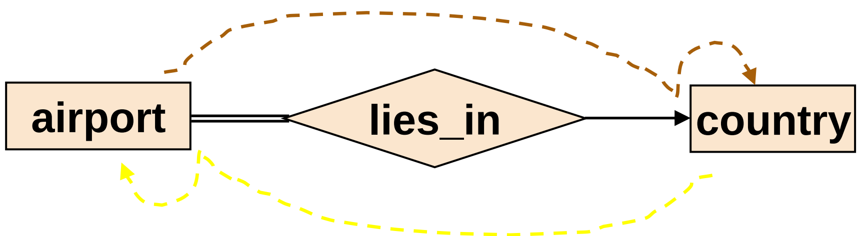
- What should be the cardinality for works_for relationship set in a typical business situation?

Participation Constraints - 1

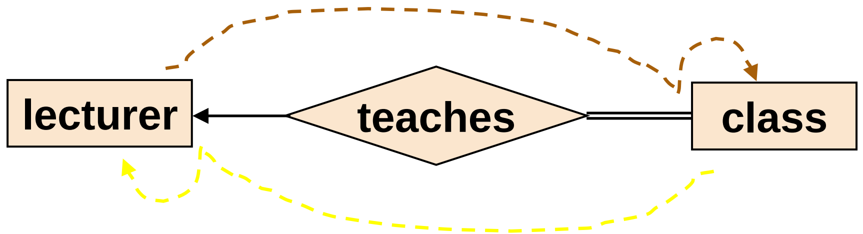
- Does every entity in an entity set needs to participate in a relationship set?
  - Yes → **Total Participation**
  - No → **Partial Participation**

Participation Constraints - 2

- **Entity Sets**: **Airport, Country**
- **Relationship Set: Lies_in**; associates an airport to a country
- **Business Rule (Constraints)**
  - An airport lies in one country; An airport cannot belong two or more countries
  - Can an airport stand on its own without belong to a country? No → Total Participation
    - An airport **must lie in** (exactly) **one** country
  - A country may have more than one airport; It may be possible that there are countries without an airport → Partial Participation
    - A country **may have zero or more** airports

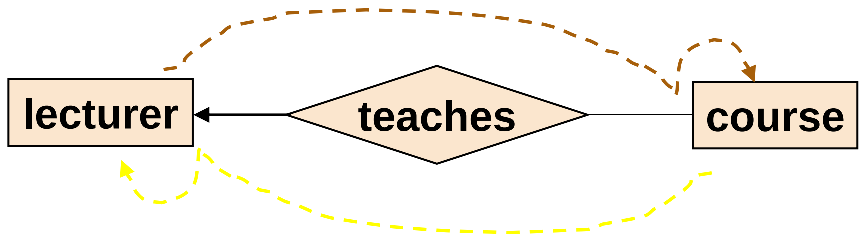


Participation Constraints - 3



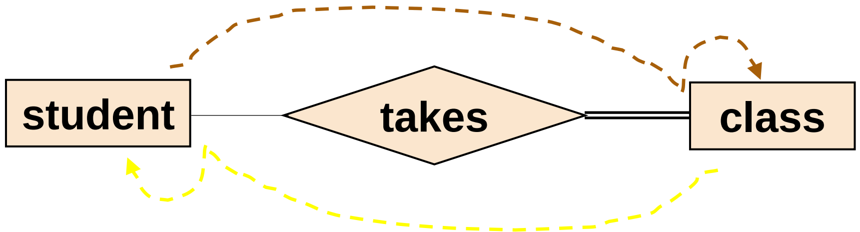
- A lecturer **may** teach zero or more classes
- A class **must** be taught by one lecturer

Participation Constraints - 4



- A lecturer **may** teach zero or more courses
- A course **may** be taught by one instructor, but not by more than one

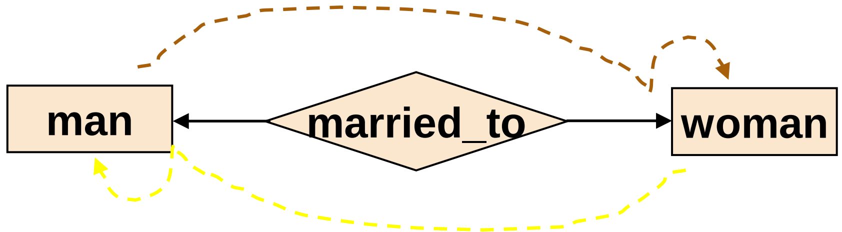
Participation Constraints - 5

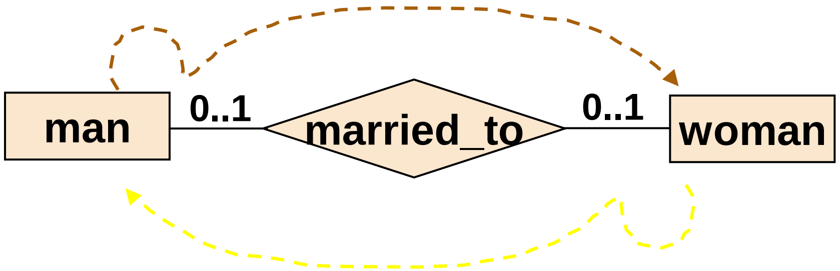


- A student **may** take zero or more class
- A class **must** be taken by at least one or more students

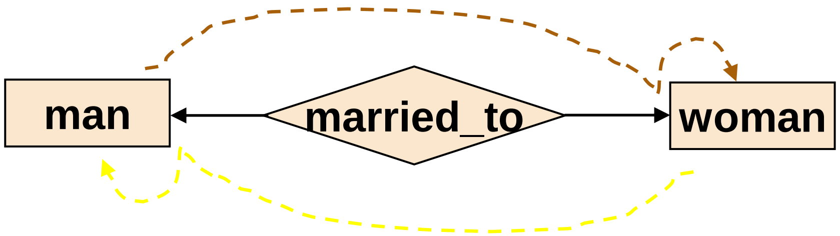
Alternative Notation: Minimum .. Maximum Cardinality - 1

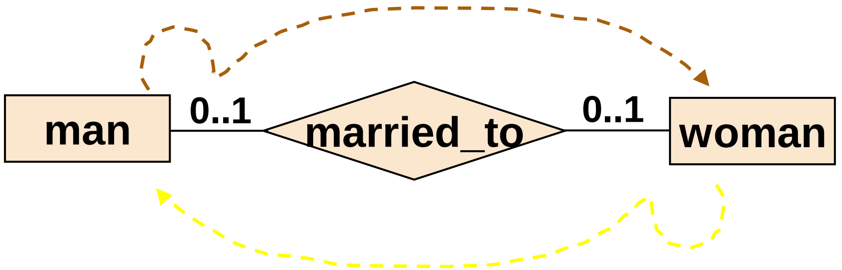
- **Cardinality Mapping** represents **maximum** number of entities
- **Participation** represents **the minimum** number of entities
- Cardinality and Participation Constraints can be combined together in an alternative notational format:
  - **Specifies the minimum and maximum cardinality**
    - **Allows more precise specification**





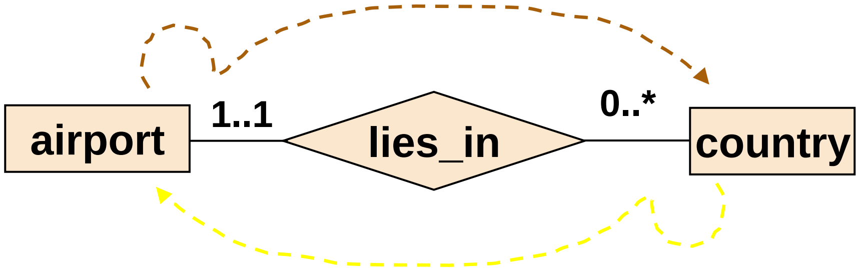
Alternative Notation: Minimum .. Maximum Cardinality - 2





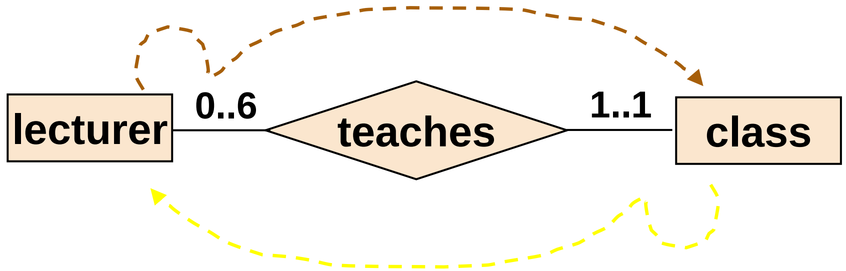
- A man entity can be associated with minimum of 0 woman entity
  - Business Logic: A man in the database may not be married
- A man entity can be associated with maximum of 1 woman entity
  - Business Logic: A man can be married to a single woman only
- A woman entity can be associated with minimum of 0 man entity
  - A woman in the database may be single and not married
- A woman entity can be associated with maximum of 1 man entity
  - A woman, if married, can only be married to a single men

Example 1



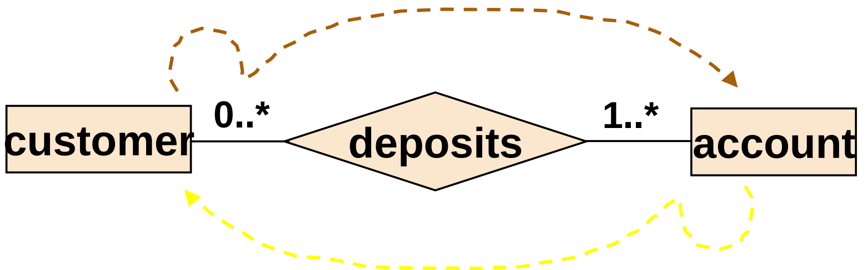
- An airport **must** lie in exactly one country
- An country **may** have one or more airports

Example 2



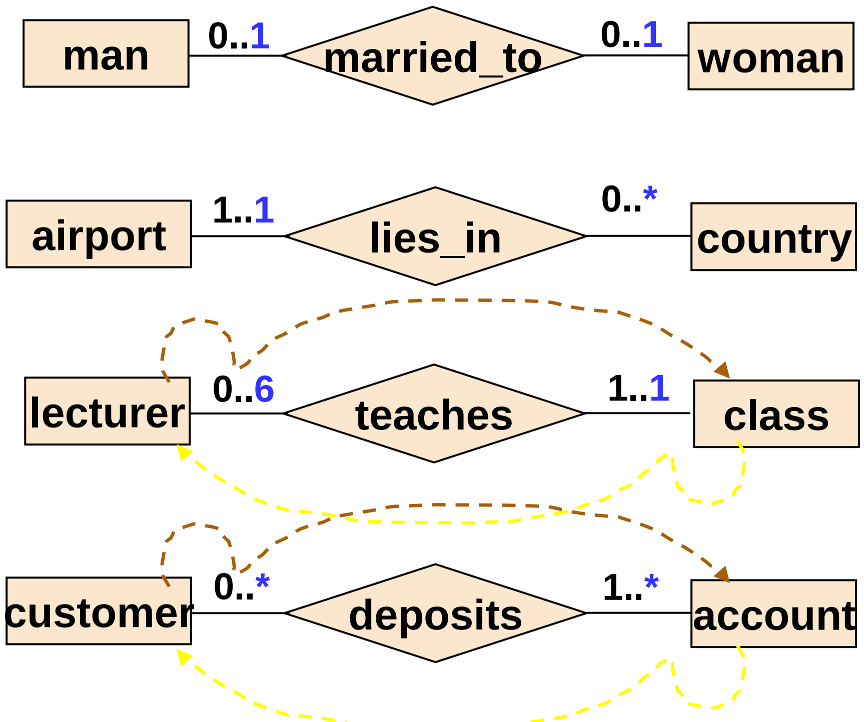
- A lecture may teach up to six classes but he/she may be freed from teaching duties when he/she is taking a research leave
- A class must be taught by a single lecturer

Example 3



- Our bank customers can have multiple accounts. We may have customers who don't have accounts (such as loan only customers)
- We allow both single-owner account and joint accounts (owned jointly by two or more customers)

Example 4



This is an One-to-One relationship

This is an Many-to-One from airport to country

This is an One-to-Many from lecturer to class

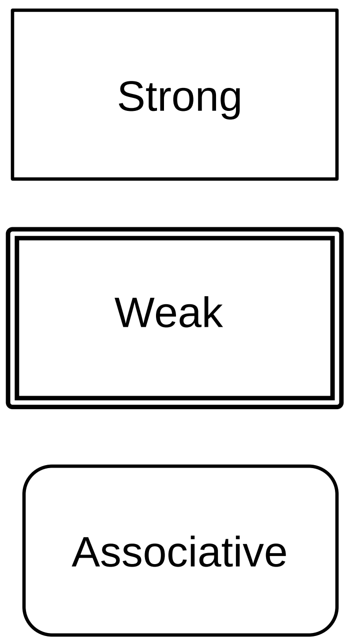
This is an Many-to-Many relationship

When we say One-to-One, Many-to-One, One-to-Many, Many-to-Many, we are only referring to maximum cardinality;

Minimum Cardinality is a more precise way of specifying participation constraints

Alternative E-R Notation

Entity Types



Attributes

Entity Name
Identifier (PK) Partial identifier (or discriminator) [Derived] { Multivalued } Composite (attr1, attr2, …, attr_n)

Student
StudentID (PK) StudentName Birthdate [age] {club} Address (Street, City, State, Postal Code)

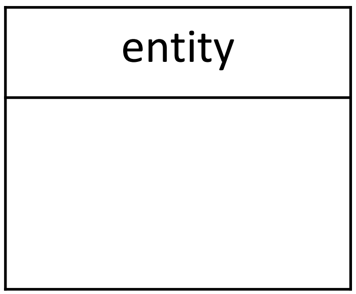
An example of composite identifier (Key): LicensePlate (LicenseNumber, Province)

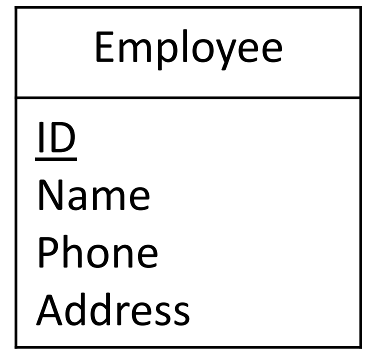
Alternative Crow's Foot Notation: Relationship Cardinality - 1

- Entities are represented as boxes,
- Relationships are represented as lines between the boxes.
- The relative cardinality of the relationship are represented as different shapes at the ends of these lines:
  - *ring* represents "zero“
  - *dash* represents "one“
  - *crow's foot* represents "many" or "infinite“

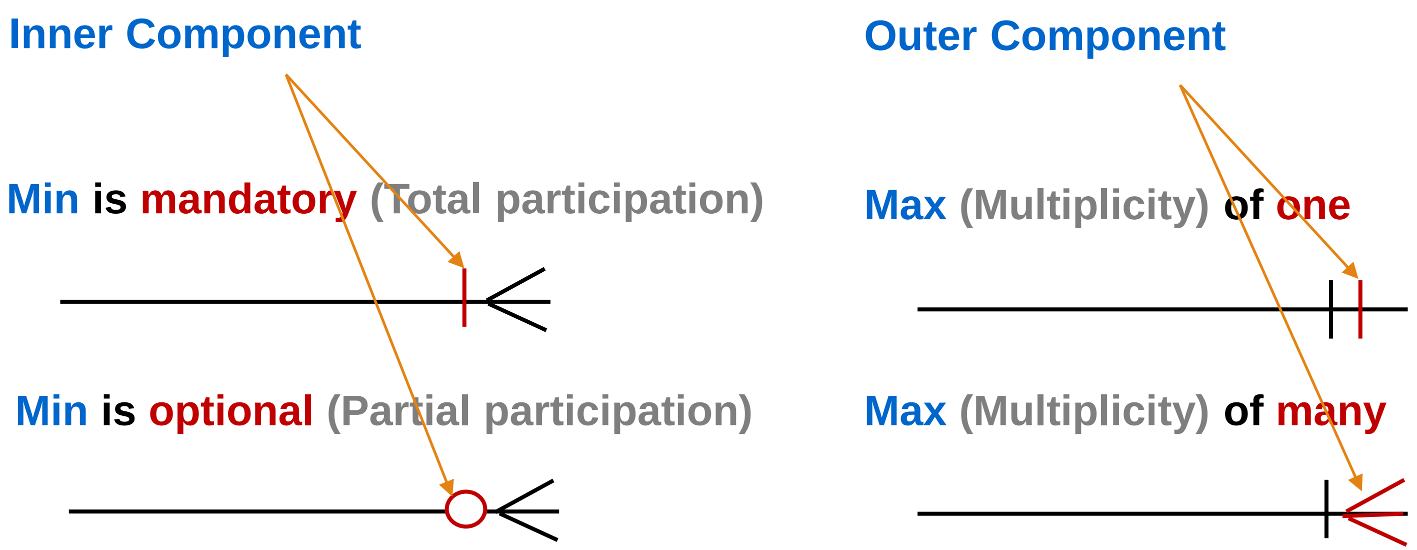
Ref: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

An Entity and Example



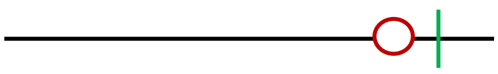


Min and Max (Multiplicity) Cardinalities in a Relationship



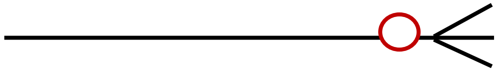
All Possible Min and Max Cardinalities in a Relationship

Optional one (zero or many)

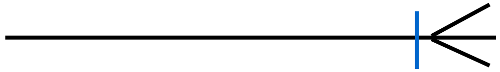


Mandatory one (one and only one)

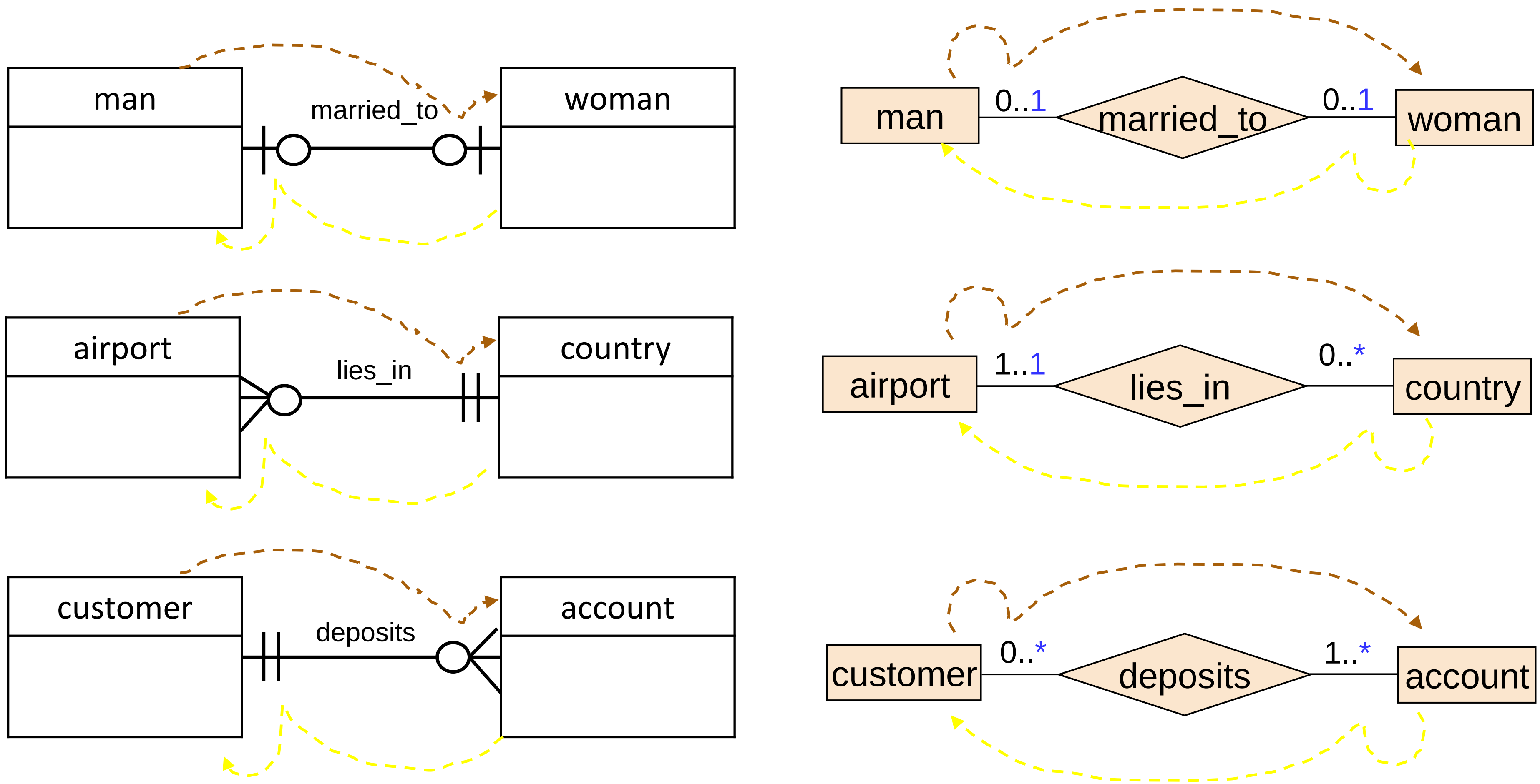
Optional many (zero or many)



Mandatory many (one or many)

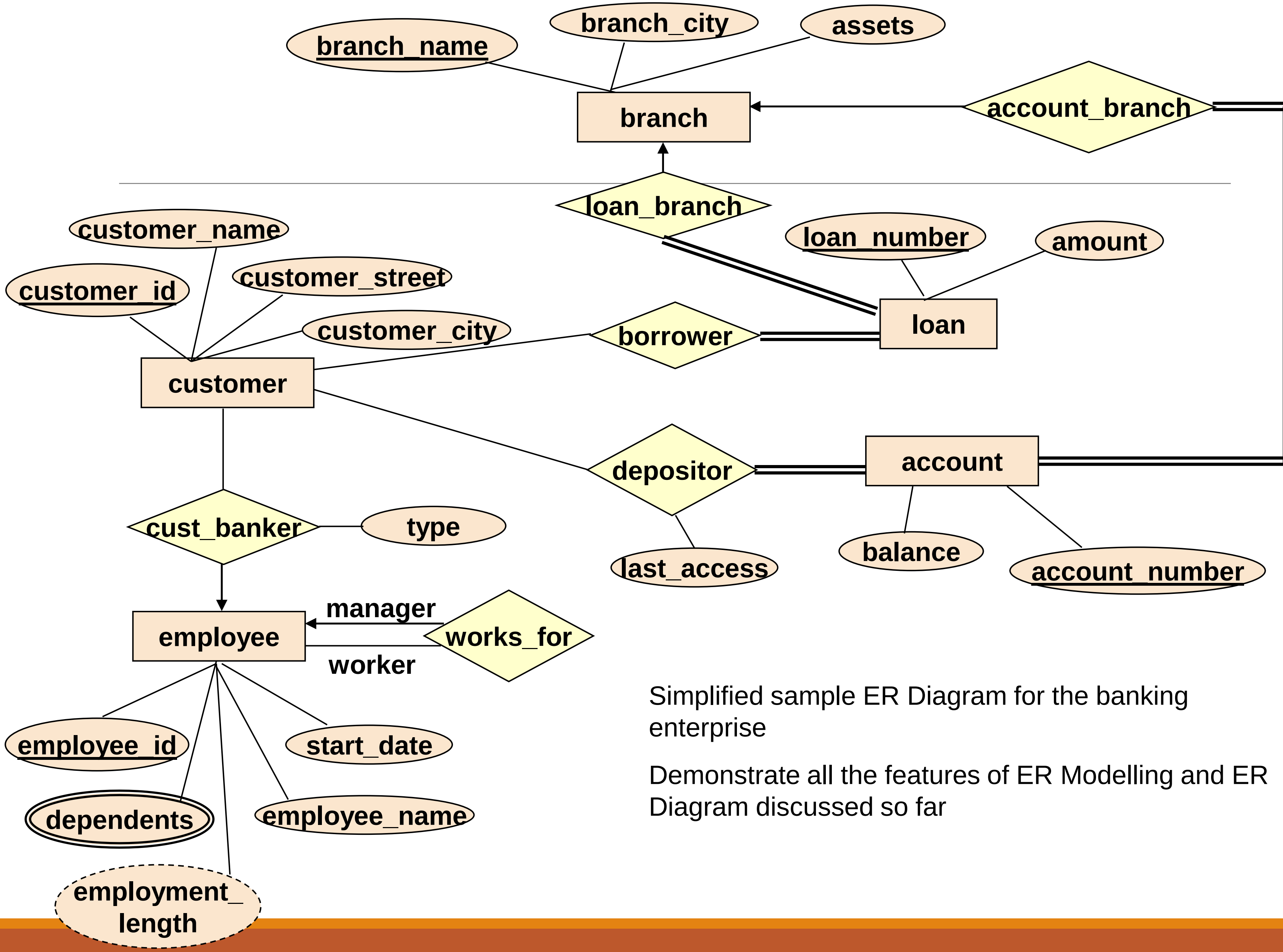


Examples of Relationships with Min and Max Cardinalities



Workshop 9-1 (Cont.)

- Refer to **Supplementary Material: Data Requirements for the Bank Database** (DataRequirementsBankDatabase.pdf)
  - **Add mapping cardinarity and participation constraints discussed to the ER model**



Simplified sample ER Diagram for the banking enterprise

Demonstrate all the features of ER Modelling and ER Diagram discussed so far

Practice 1

- Design conceptual database using ER model for the given scenario. Create an Entity-Relationship (ER) Diagram based on the following requirements of ABC LIBRARY Database. In particular, appropriately use symbols that represent Entity, Relationship, Attributes, Key Attribute, Multivalued Attribute, Composite Attribute, Derived Attribute, and Cardinality Ratio if any of them presents in the requirements.
- The library collects books and provides services to students and faculty members in ABC University. Books comprise of ISBN, Call Number, Author(s), Title, Edition, Imprint, and Subject(s). A Call Number represents a book's category, subcategory, book code, and published year. The library may keep multiple copies of the same book, and each of them has a unique Call Number (for example, using '005.74 P916**c1** 2005', '005.74 P916**c2** 2005' to represent the first and second copies of the same book). Alternatively, the computer generated number called 'Item Number' is also used in the database system to represent a copy of any book (for example, using 0000284425 and 0000284426 to represent the first and second either 'Faculty Member' or 'Student'), and Faculty. copies of the same book). The library also keeps status of any books: 'Available', 'Borrowed' and storage location: 'X library' and 'Y Library'.
- For any library members, the library keeps their ID, first name, middle name (if any), last name, status (either 'Faculty Member' or 'Student'), and Faculty.
- Faculty members and students can borrow and/or return books from the library. A librarian will keep track of all those books. In case of borrowing a book, a librarian will keep track of the borrower's ID, Item Number of a book and due date. In case of returning a book, a librarian will keep track of the borrower's ID, Item Number of a book and returned date.

An example of books' detail stored

ISBN	**0619215291**
Call Number	005.74 P916c1 2005
Author	Pratt, Phillip J.
Title	Concept of Database Management
Edition	5th ed.
Imprint	Boston, MA : Thomson/Course Technology, c2005.
Subject	Database design, Database Management
Item Number	0000284425
Status	Available
Location	X library

Practice 2

- Design conceptual database using ER model for the given scenario. In particular, appropriately use symbols that represent Entity, Relationship, Attributes, Key Attribute, Multivalued Attribute, Composite Attribute, Derived Attribute, and Cardinality Ratio if any of them presents in the requirements.
- A delivery service namely Kelly Express provides services to deliver parcels to customers. The database will be designed to manage the related data in delivering parcels.
- A customer requests to deliver one or more parcels to recipients. He has to provide his/her information as well as recipient's information as follows: first name and last name, address, (which includes address number, street, district, city, province, postcode) and phone number. Suppose that all customers are uniquely identified by their name.
- The parcel's detail must be recorded as follows: weight (in kg.), wide, length and height (in cm.). Each parcel are associated with an ID.
- There are many trucks available for the delivery services. The company keeps the ID, license plate, make (e.g., TOYOTA) model (e.g., VIGO) and year of each truck. The company also keeps information of the driver, which are employee ID, first name, last name, driver license number, and phone number.
- Only one driver will be assigned to each truck for the delivery service in each day. Suppose that every delivery is a single trip done within the same date. The delivery schedules must be recorded, which include truck ID, employee ID, arrival date and time.

Practice 3

- Design conceptual database using ER model for the given scenario. Create an Entity-Relationship (ER) Diagram based on the following requirements of Tribute Database. In particular, appropriately use symbols that represent Entity, Relationship, Attributes, Key Attribute, Multivalued Attribute, Composite Attribute, Derived Attribute, and Cardinality Ratio if any of them presents in the requirements.
- Tribute Company sells bicycles and the owner needs to create a database to manage data of his customers. The company offers several types of bicycles: road bicycle, touring bicycle, mountain bike, hybrid bicycle, and folding bicycle. Each bicycle will have a unique identifier number assigned as well as details of color, type, make, model, manufacturing year, frame, and wheel's size. The database also stores details of customers, i.e., first name, last name, date of birth, phone numbers, and address. The company owner wants to keep the record of maintenance service's data provided to his customers. In particular, the service's details are as follows: service date, bicycle id, service id, price relating to each service, and expected finishing date, e.g., changing tire and fixing rear brakes. There may be several service types requested by a customer per service. Each customer may also bring multiple bicycles to have maintenance service at the same time.