Admin Related

- Lecture recording
- Tutorial format – er diagram
- I need to leave asap after lecture
- Enrolment difficulty?

Database Design

Entity-relationship Model

(ER model)

Scenario

- A: “Welcome to the job, can you build us a database to organize all the information on a publisher and present it at the meeting?”
- B: “okay, what’ll be in the database?”
- A: “Well… authors, books, editors, printers etc. An author can write zero or more books and a book is written by one or more authors. Where a book is uniquely identified by its book-id. For each book, we also record its title, price, and availability. An editor is uniquely identified by his/her reader-id and we also record his/her name, phone number, address. The address is composed of street and suburb…”

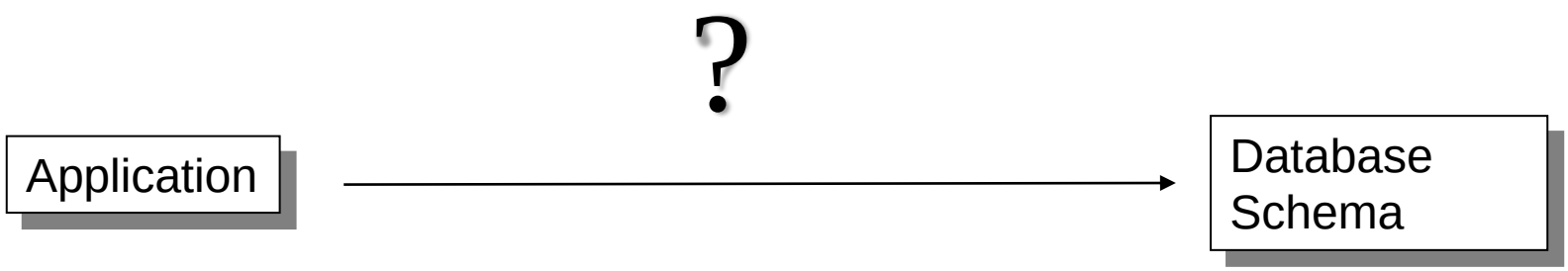
The Application

- *"An author can write zero or more books and a book is written by one or more authors. Where a book is uniquely identified by its book-id. For each book, we also record its title, price, and availability. An editor is uniquely identified by his/her reader-id and we also record his/her name, phone number, address. The address is composed of street and suburb…"*

  This is the application we need to design the design for; the application has a set of requirements.

Simplified Database Workflow

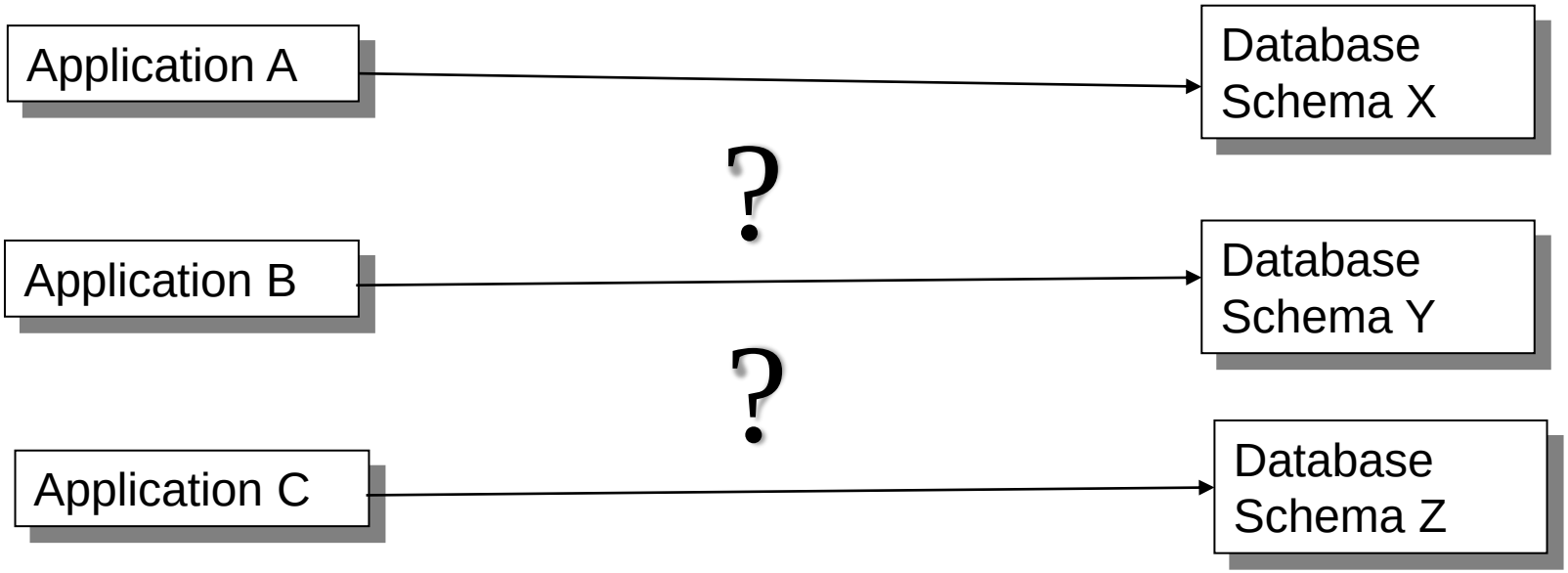
- What should we do?



- Application: has many requirements
- Database Schema: provides a logical view of data model

Simplified Database Workflow

- More Generally, how should everyone build his/her databases for an application for its given requirements?



- Database Schema: provides a logical view of data model

Solution

- Dr. Peter Chen
  https://www.csc.lsu.edu/~chen/
- His work started a new field of research and practice:
  Conceptual Modeling

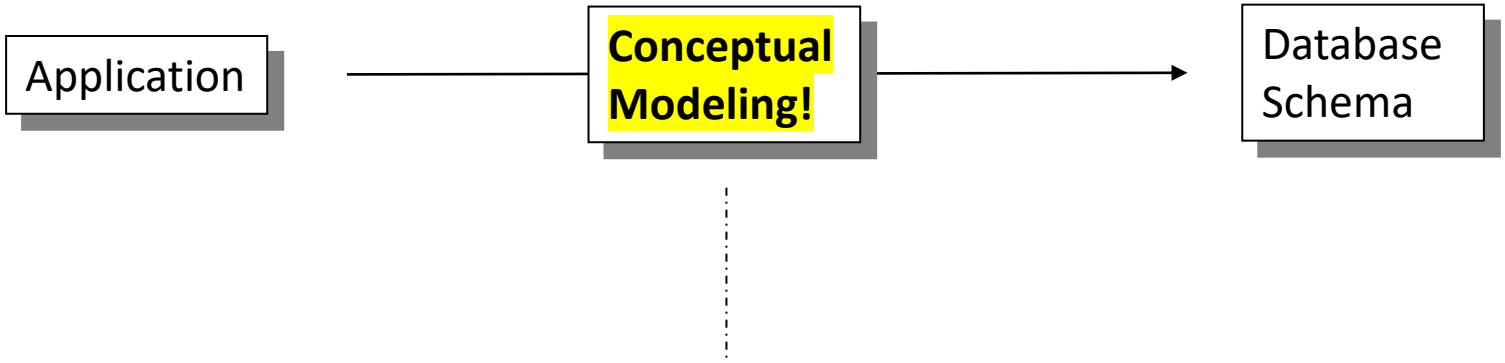
- See paper on Entity-Relationship Model in link
  https://bit.csc.lsu.edu/~chen/pdf/english.pdf

English Sentence Structure and Entity-Relationship Diagrams

PETER PIN-SHAN CHEN*

Graduate School of Management, University of California, Los Angeles, California 90024

Solution



(Chen. 1976)

- Entities
- Relationships
- Attributes

Entity-Relationship Model

- Chens ER model has two major components:
  - ***Entity***: collection of attributes describing object of interest
  - ***Relationship***: association between entities (objects)

- There's also the third components unofficially
  - ***Attribute***: data item describing a property of interest

Entity-Relationship Model

- “Entities represent things in the real world, and attributes describe properties of entities.”

Entities

- Some examples of entities

Simple Attributes

- Attributes may be **simple (atomic)**
- Each simple attribute of an entity type is associated with a **value set** (or **domain** of values), which specifies the set of values that may be assigned to that attribute for each individual entity.

  - For example:
    - Entity = Student
    - Attributes = Student number, name…
  - Student A:
    - Student number = z12345678
    - Name = Bob
    - …

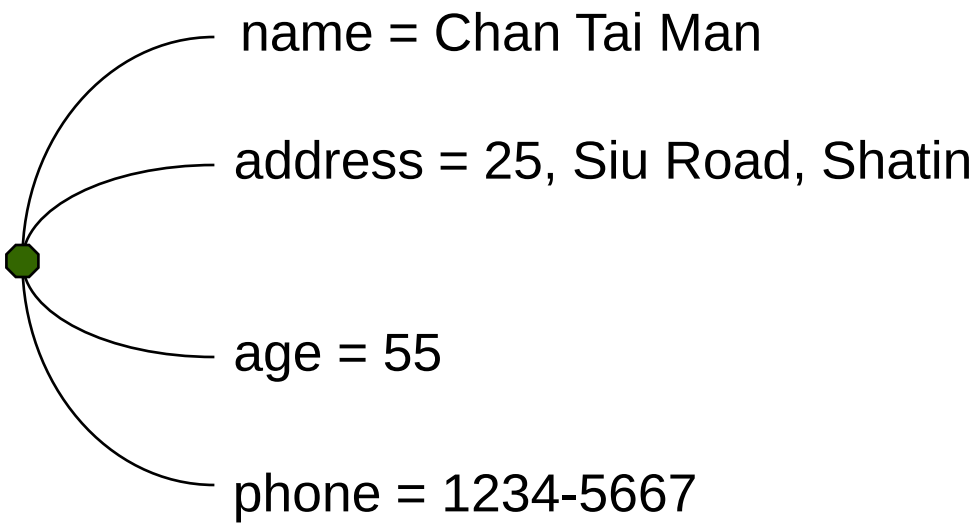
Entity-Relationship Model

- Entity-Relationship (ER) model is a popular conceptual data model.
- This model is used in the design of database applications.
- The model describes data to be stored and the constraints over the data.

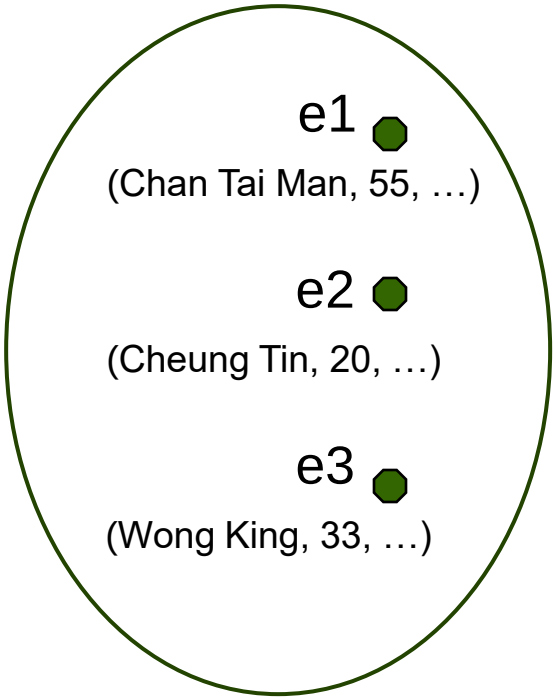
Entities and attributes

- E-R model views the real world as a collection of entities and relationships among entities.
- An entity is an object in the real world that is distinguishable from other objects.
  - Examples
    - A classroom
    - A teacher
    - The address of the teacher

- An **entity** is described using a set of **attributes** whose values are used to distinguish one entity from another of the same type.

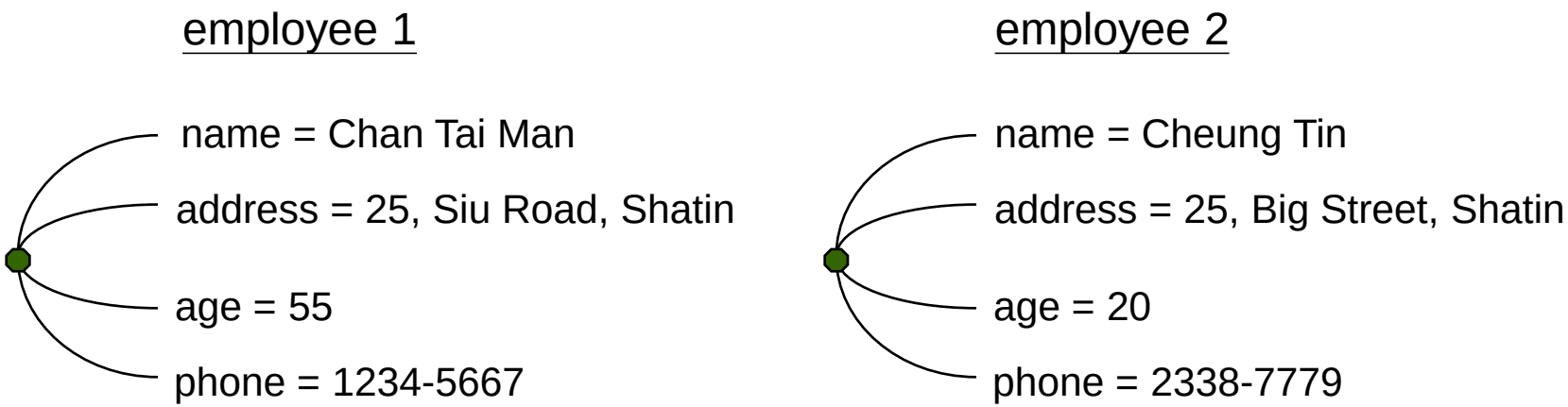


- An **entity set** is a collection of entities of the same type.



- All entities in a given entity set have the same attributes (though the values may be different).

employee = (name, address, age, phone)



- For each attribute associated with an entity set, we must identify a **domain** of possible values.
  - Example
    - The domain associated with the attribute name might be the set of 20-character strings.
    - The domain associated with the attribute age might be an integer.

Entity Instances/Facts

- A good entity schema should have good attributes that can be filled with good values.
- Car Entity Schema:
  - ((Reg. Num, State), Make, Model, Year, {Colour})
- Car instance:
  - (CS9311, NSW), Toyota, Toyota Corolla, 1999, {White, Silver})

Database Development Workflow

- Nowadays, database design process is a mature one.
    1. Analyse requirements and develop a high level model.
    2. Implement the data model as relational schema.
    3. Implement the relational schema as a database schema.

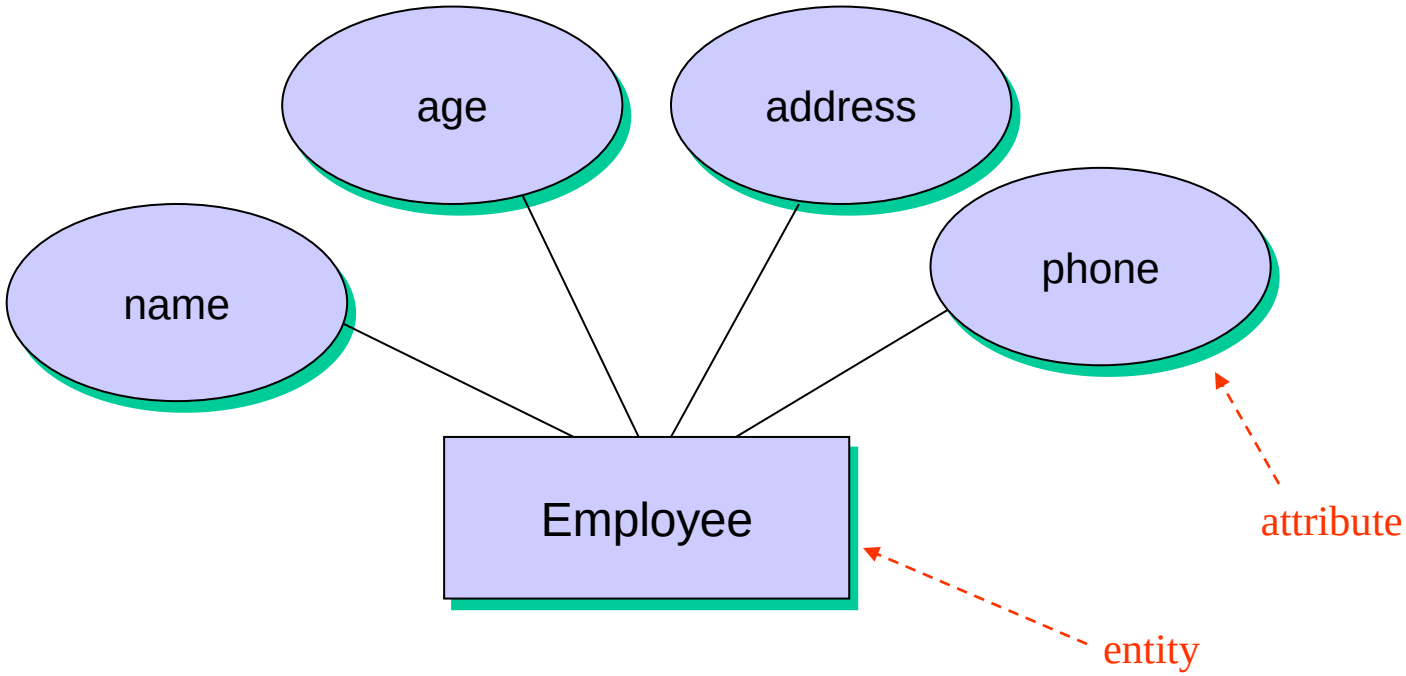
Visualizing an ER Data Model

- *At a daily meeting:*
  A: “so far we completed that high level data model that you wanted sir…”
  A: *hands a piece of paper*

- Paper reads:
  “There is one strong entity type that is a Car.
  It has a multi-labelled attribute to describe its colour.
  It has Registration with is made of Registration Number and State.
  It has a few more attributes Make, Model and Year
  …”

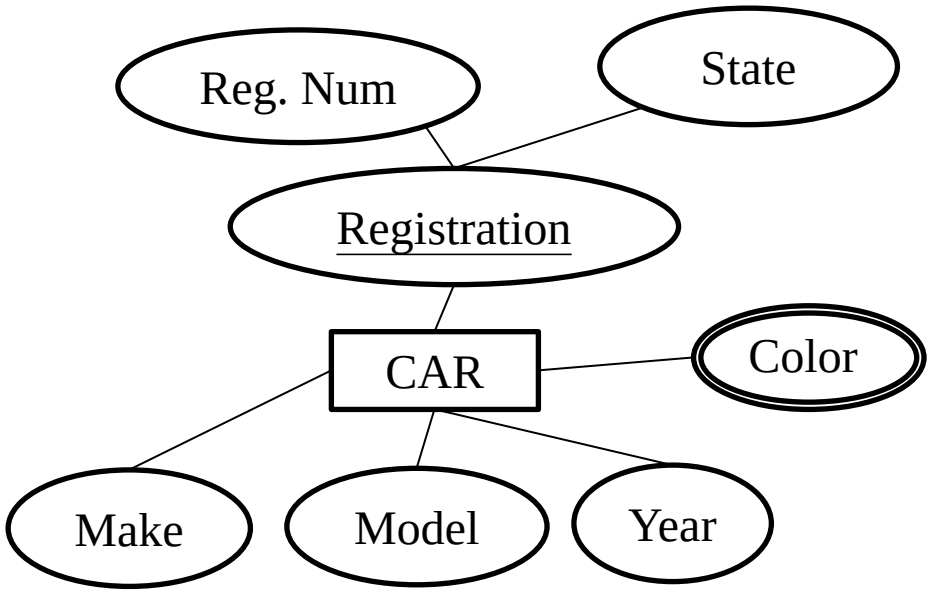
Entity-Relationship diagram (E-R diagram)

- The E-R model can be presented graphically by an E-R diagram.

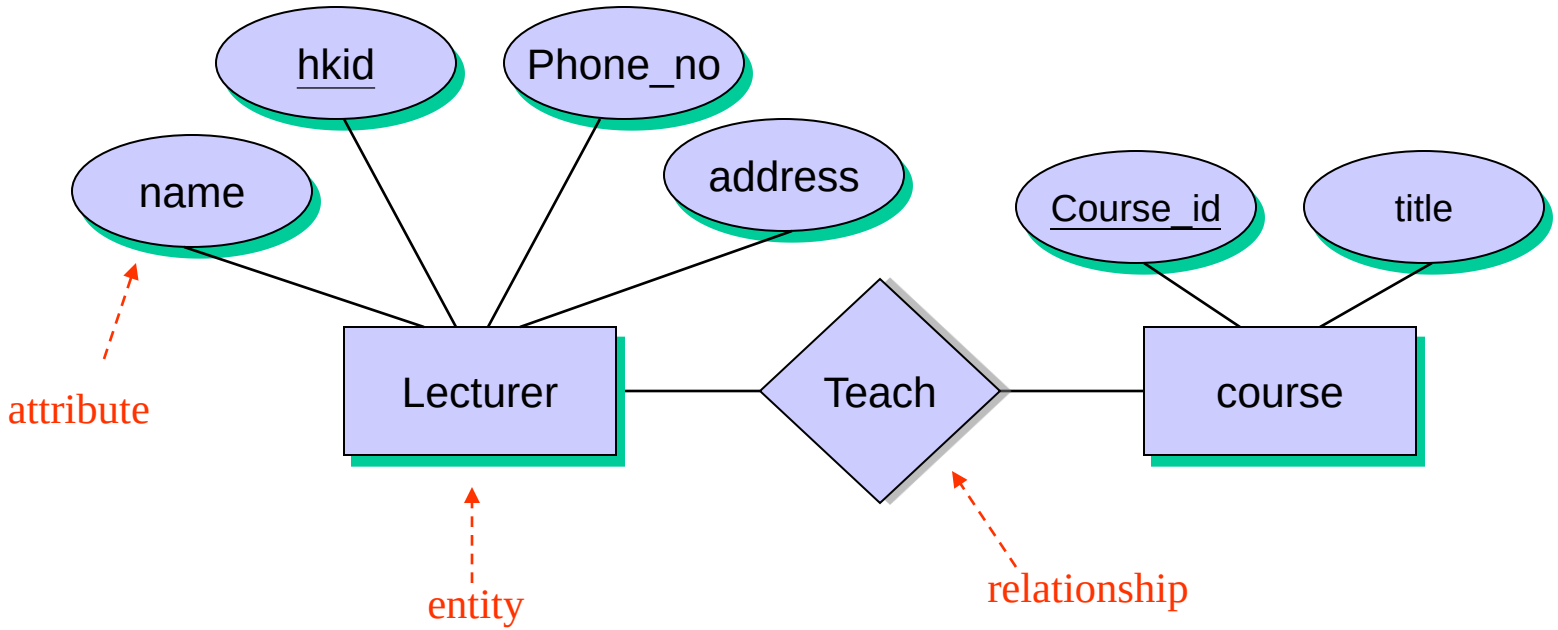


Visualizing an ER Data Model

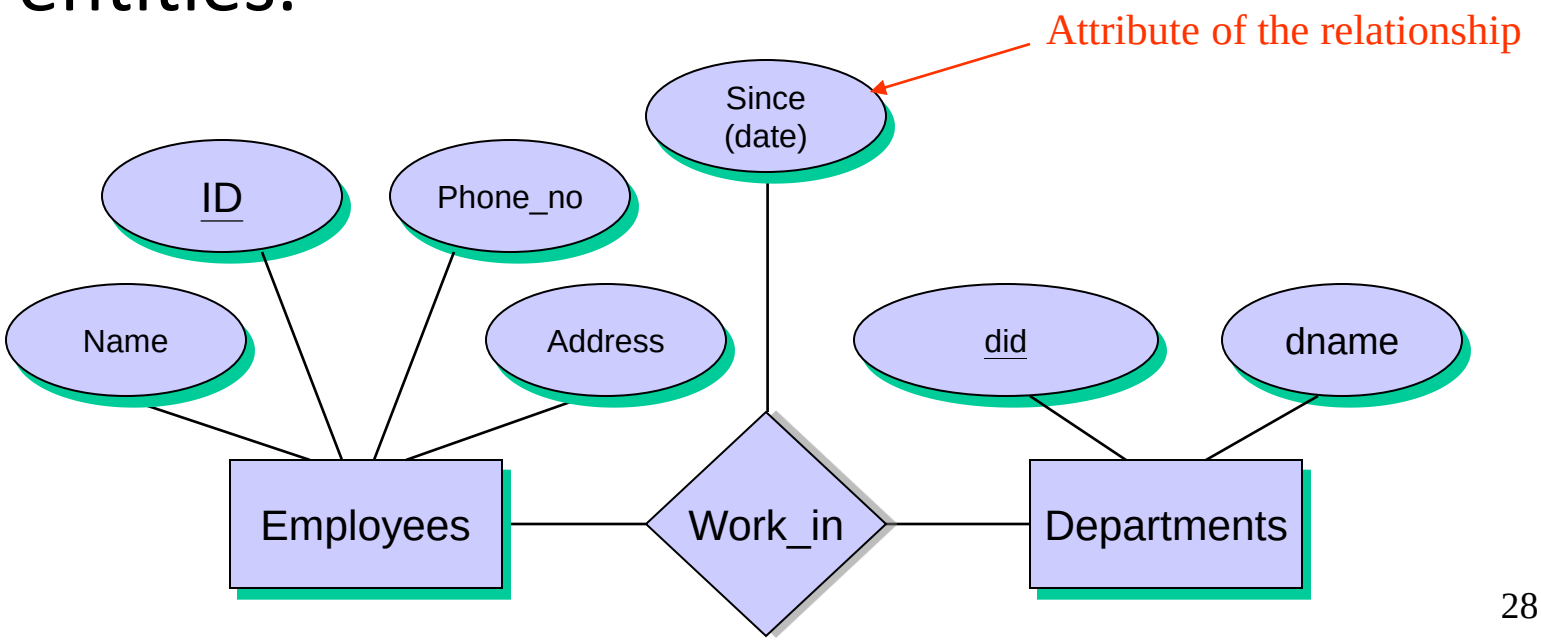
- The data model can be effectively described using the Entity-Relationship Diagrams (ERDs).



- A relationship set can also be represented by an E-R diagram.

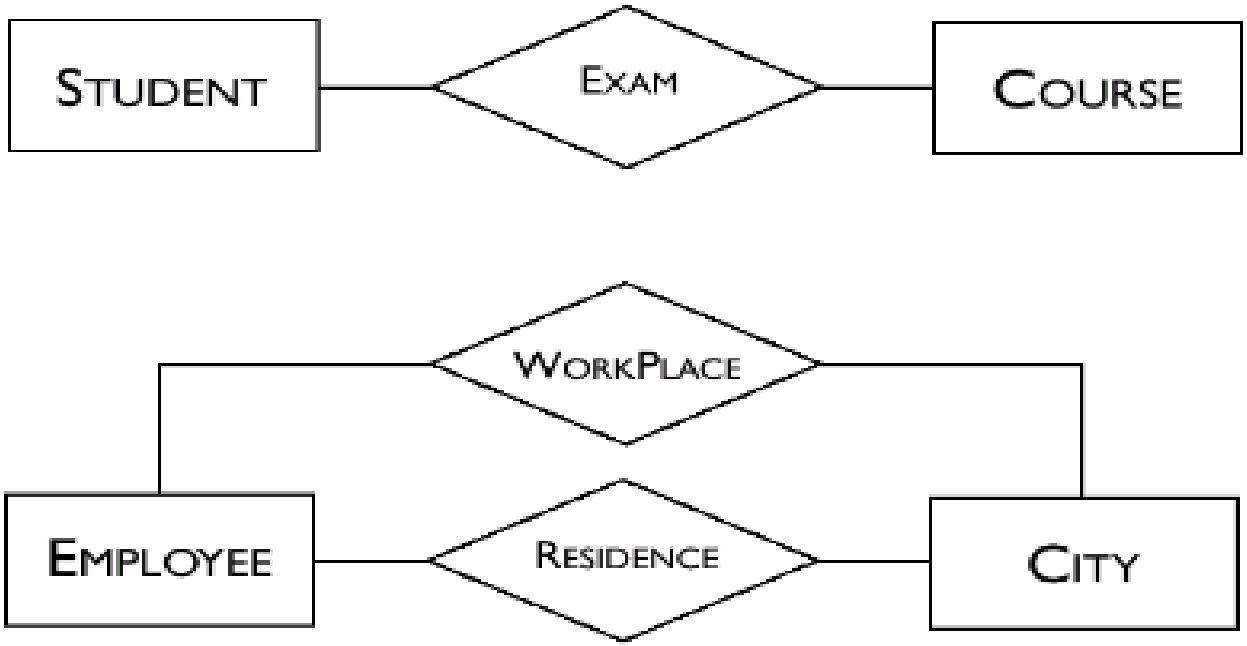


- A relationship can also have descriptive attributes.
- Descriptive attributes are used to record information about the relationship, rather than about any one of the participating entities.



Relationship

- Second Big Component of ER: They represent logical links between two or more entities.



Case in Point

- What relationships could there be if we have entities EMPLOYEE and CITY?

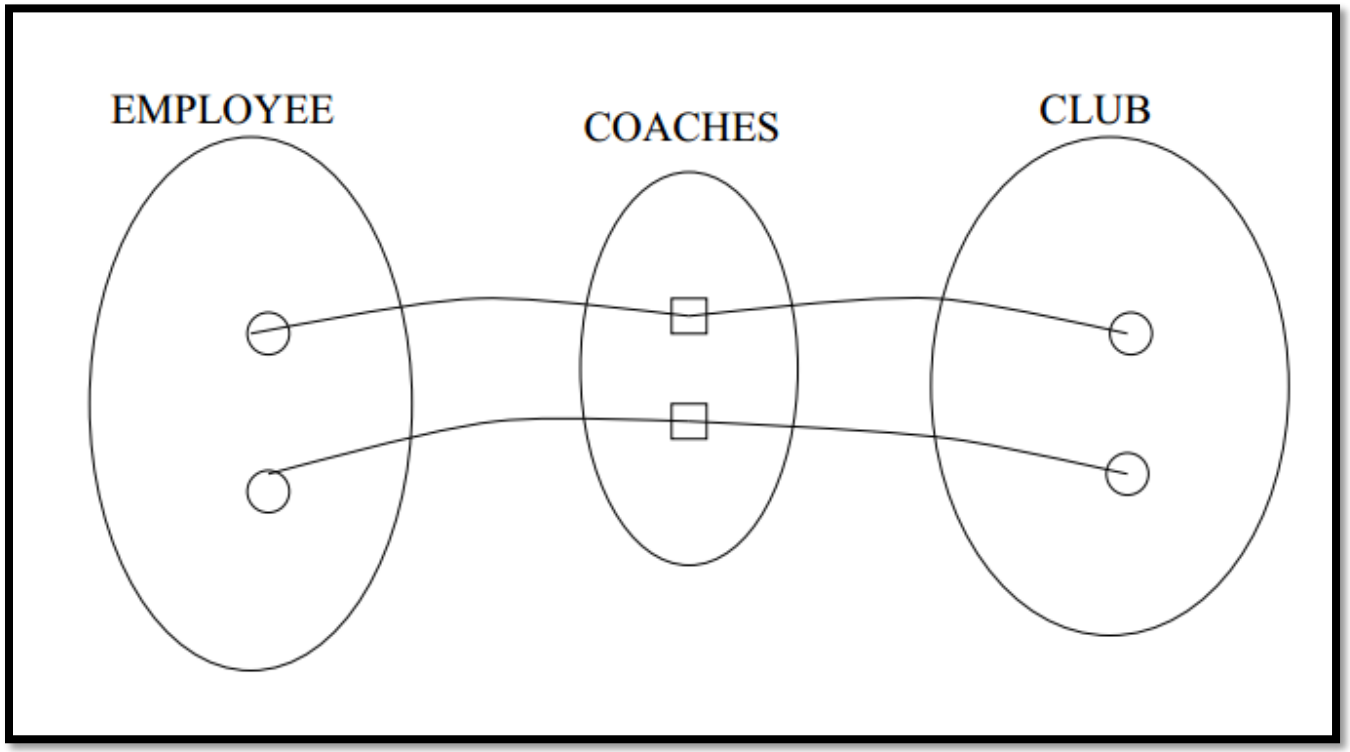
Relationships

- Relationships relates one entity type to another entity type.
- An entity type can be related to more than one other entities, and will have a different role to play for each relationship (Name each relationship to distinguish them).

An entity can also have several different relationships with another entity.

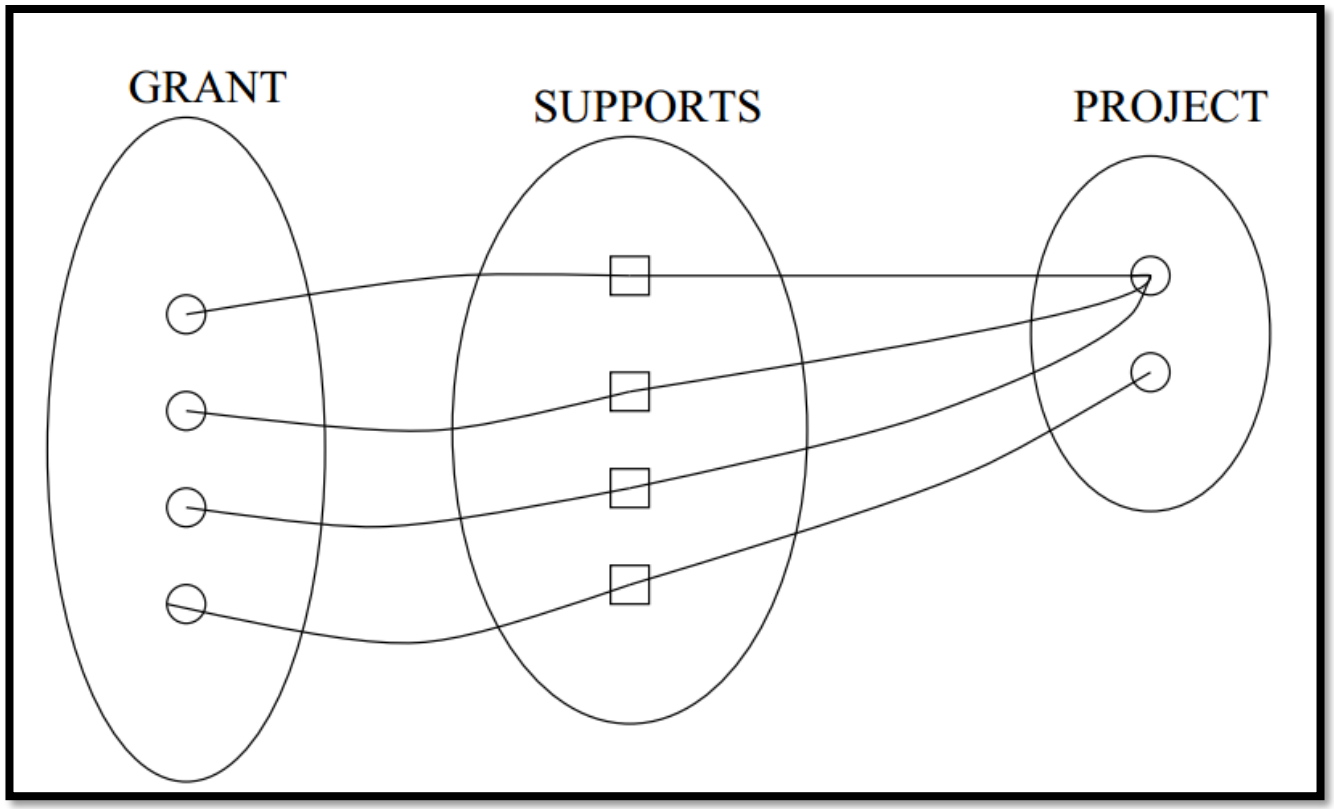
Practice

- Coaches is a relationship. It's instances describes the relation between a coach and a club.



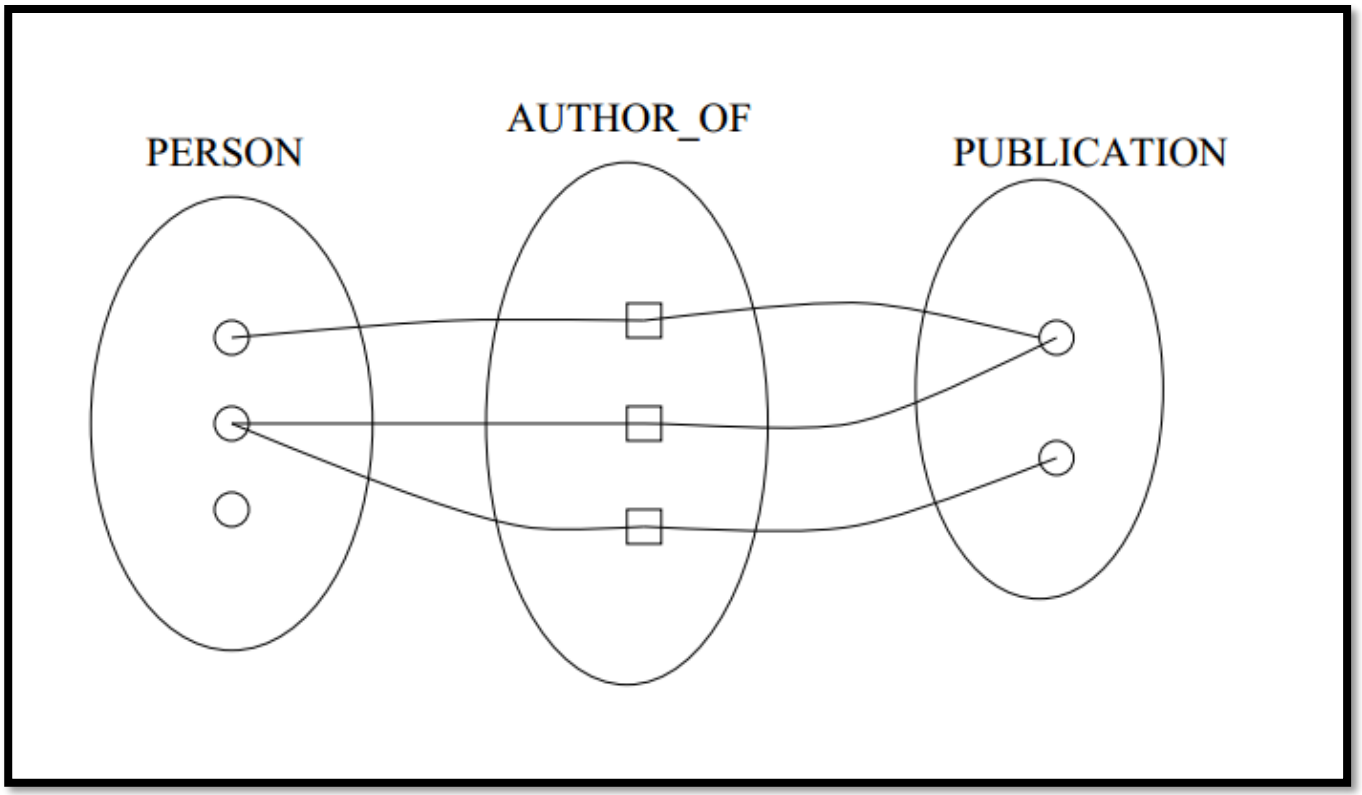
Practice

- Supports is a relationship. It's instances describe the how grants are given to projects.



Practice

- Author-of is a relationship. It's instances describe who wrote what publication.



Modelling relationships

- Relationship types usually have certain properties that limit the possible combinations of entities participating in relationship instances.

- By default, entities can be related to other entities freely.

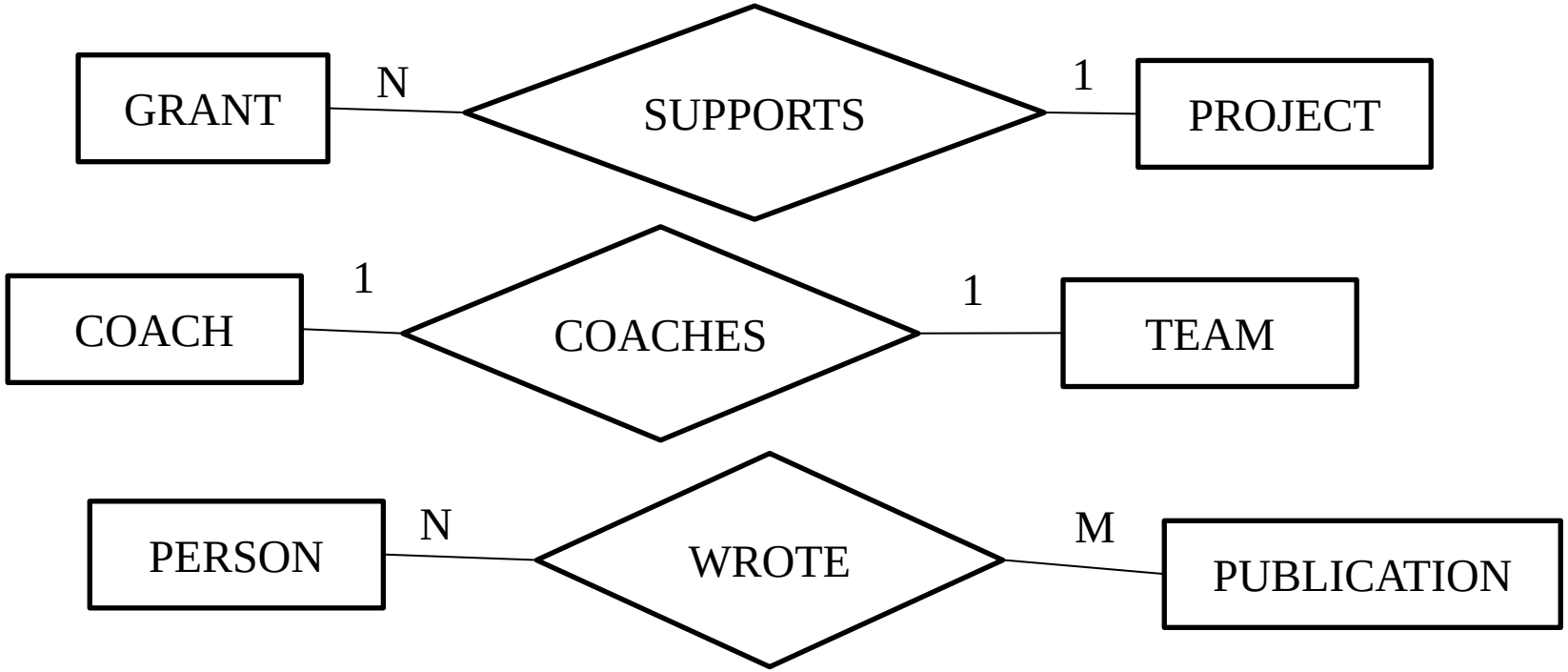
Modelling relationships

- ***Cardinality ratio***: the number of relationship instances an entity can participate in.

- There are **three** types of cardinality in relationship: is M:N (many to many), 1:N (one to many) is stricter, and 1:1 (one to one) is the strictest.

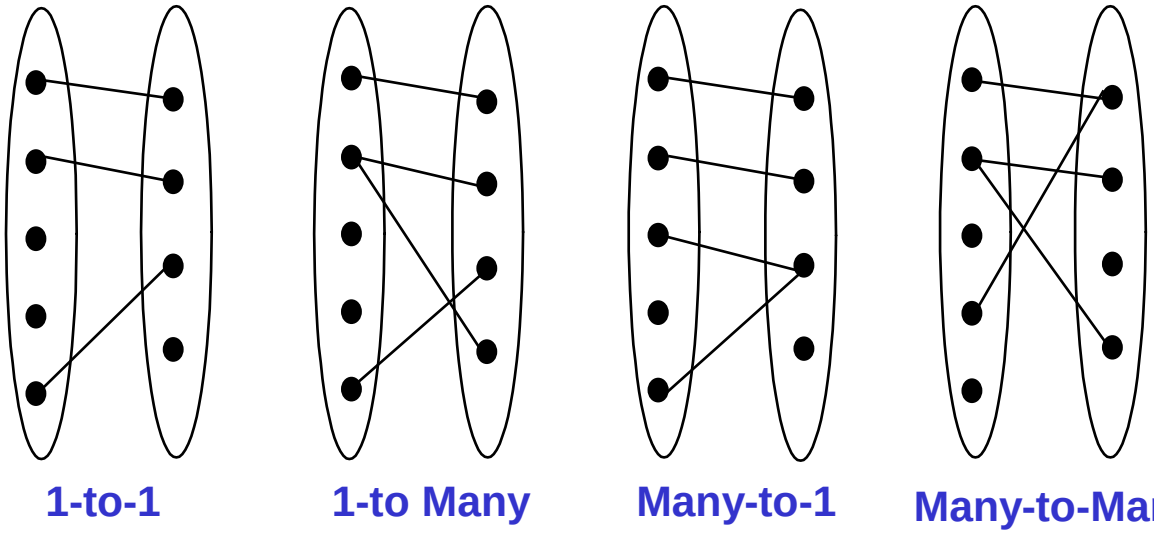
Constraints on relationship types(cont)

- We can also show this in an ERD (the notation here is just a helpful starting guideline, not to be confused with the course notation)

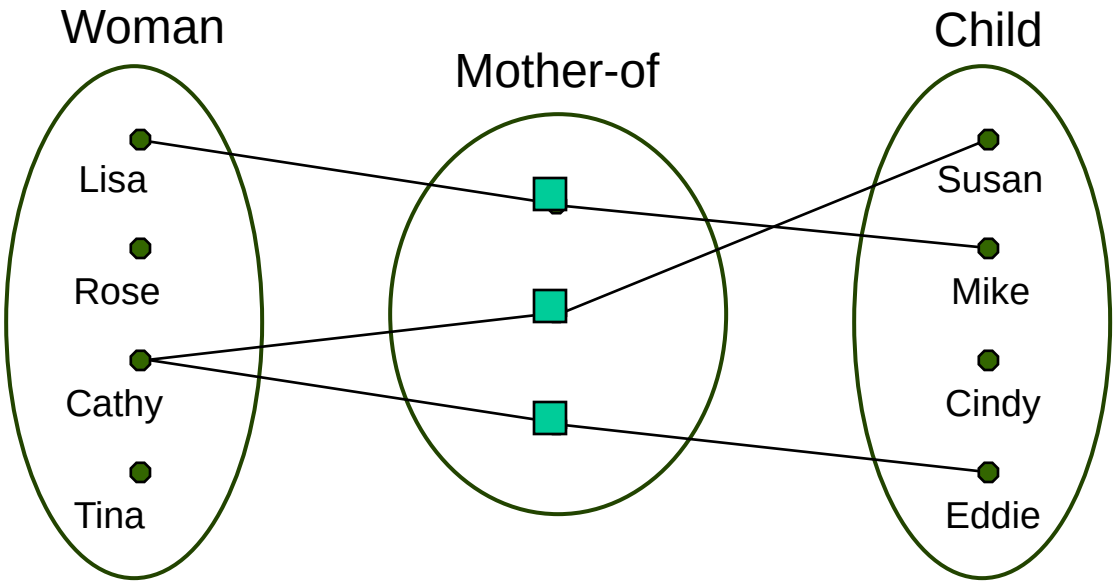


Relationship Constraints

- The mapping of the relationship can be classified into the following cases:

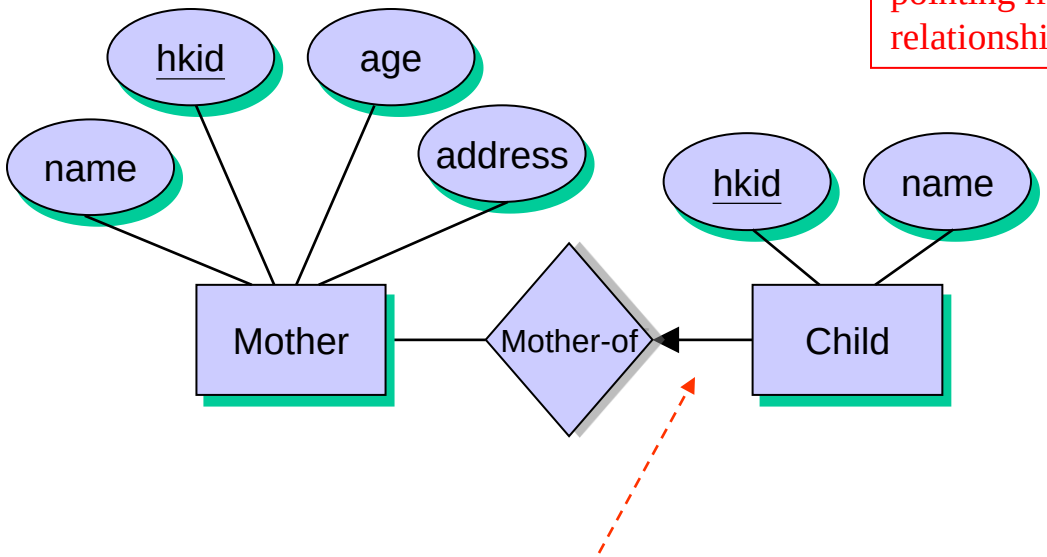


- One-to-many
  - One-to-many constraint from A to B: an entity in B can be associated with at most one entity in A.
  - Each child can appear in at most one mother-child relationship.



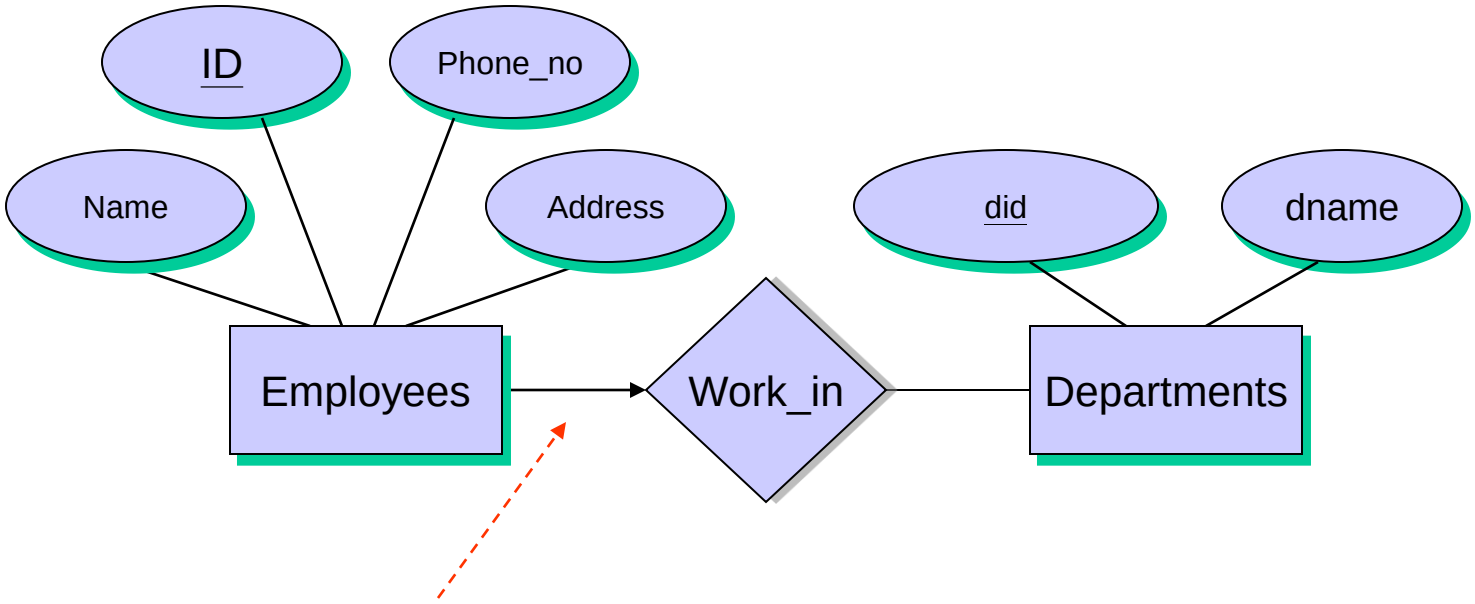
- Child has a key constraint in the mother-of relationship set.
- This restriction can be indicated by an arrow in the E-R diagram.

Note: the arrow is always pointing from an entity to a relationship.



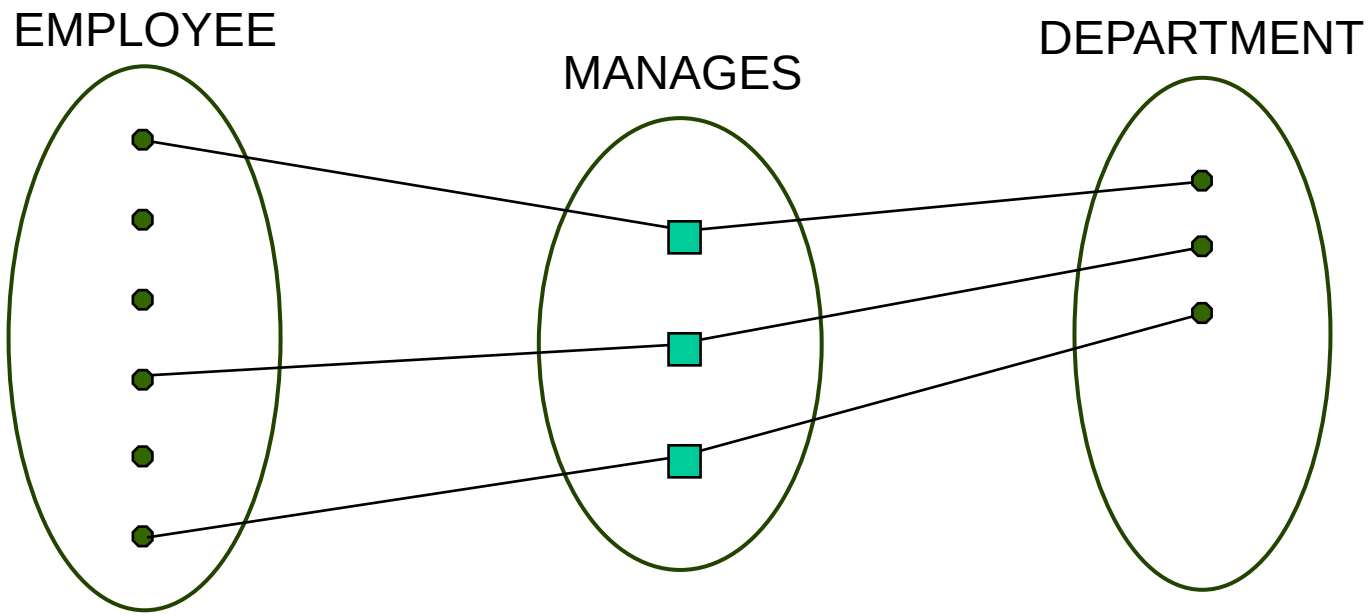
Intuitively, the arrow states that given a child entity, we can uniquely determine the mother-of relationship.

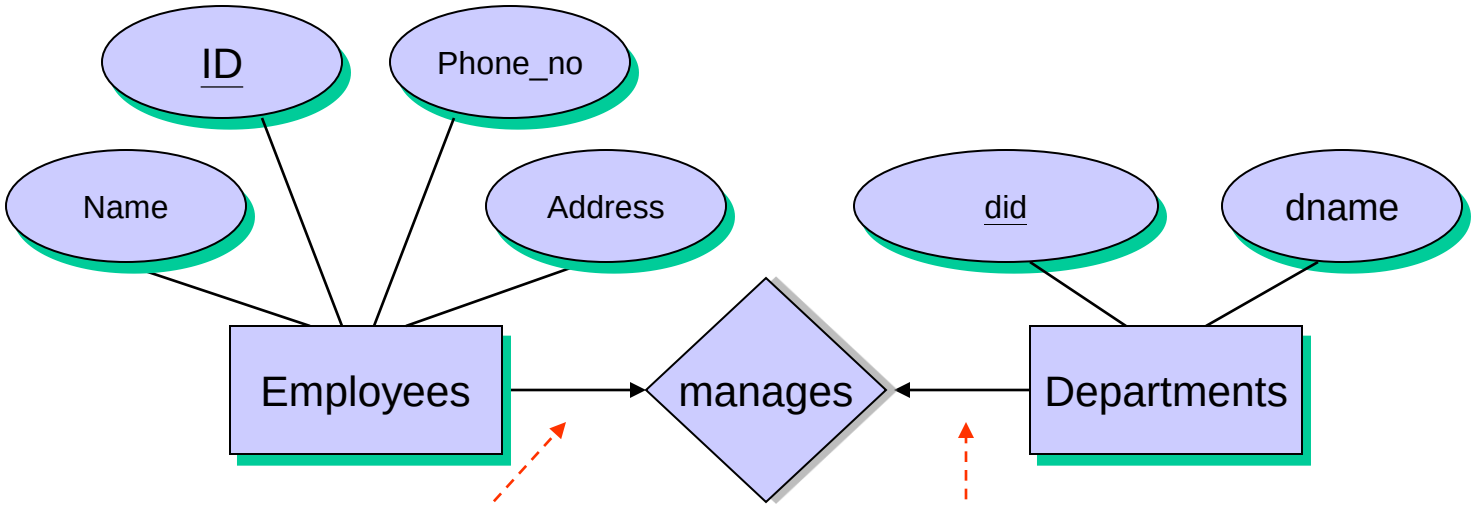
- Many-to-one
  - Similar to one-to-many



Each employee belongs to one department and a department may have many employees.

- One-to-one
  - If the relationship between A and B satisfies the one-to-one mapping constraint from A to B, then an entity in A is related to at most one entity in B, and an entity in B is related to at most one entity in A.

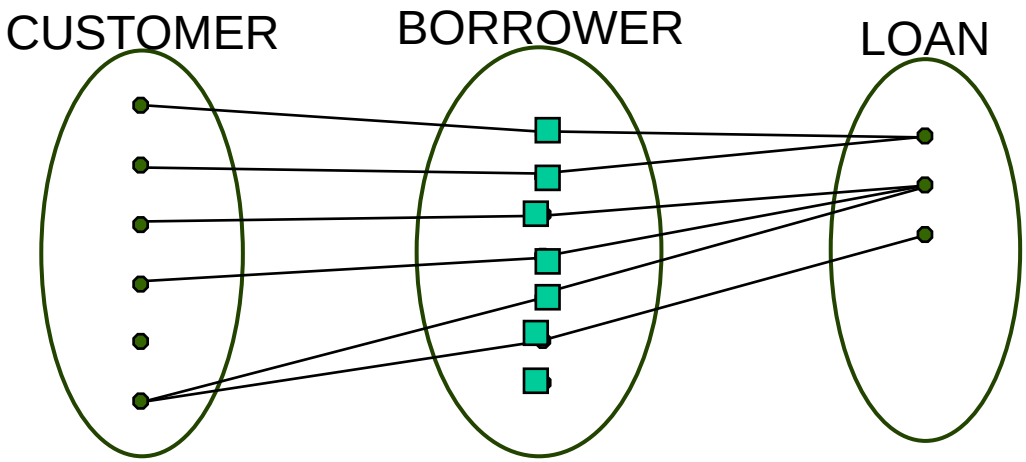


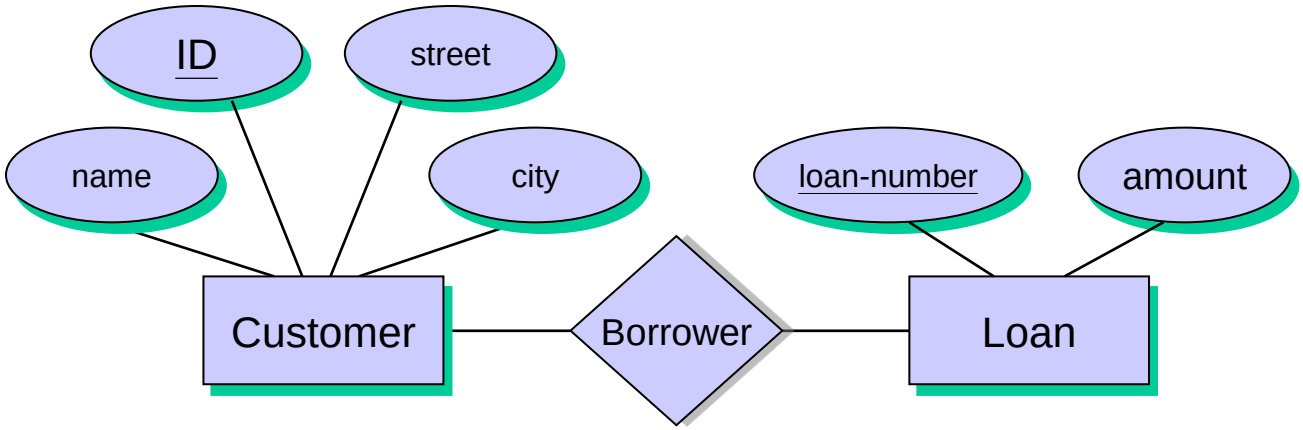


An employee can associate with at most one department via the relationship "manages".

A department can associate with at most one employee via the relationship "manages"

- Many-to-many
  - An entity in A is associated with any number of entities in B, and an entity in B is associated with any number of entities in A.
  - In fact, it means that there is no restriction in the mapping





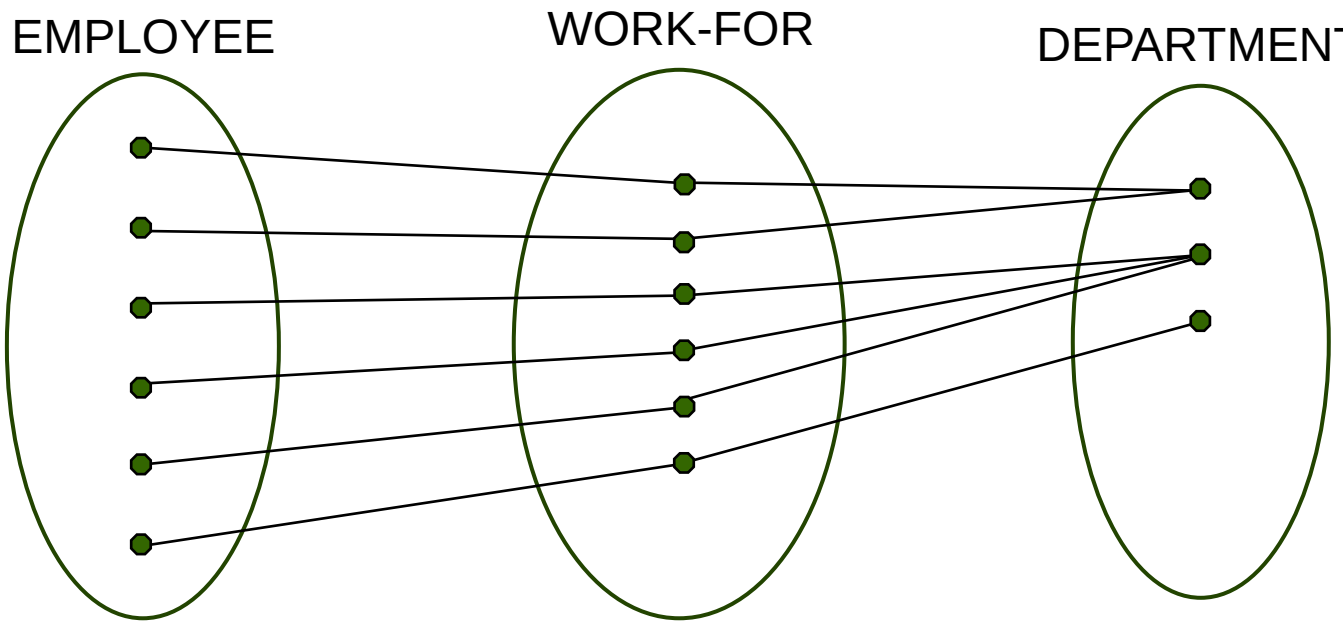
- A customer can associate with several loans (possibly 0) via Borrower
- A loan can associate with several customers (possibly 0) via Borrower

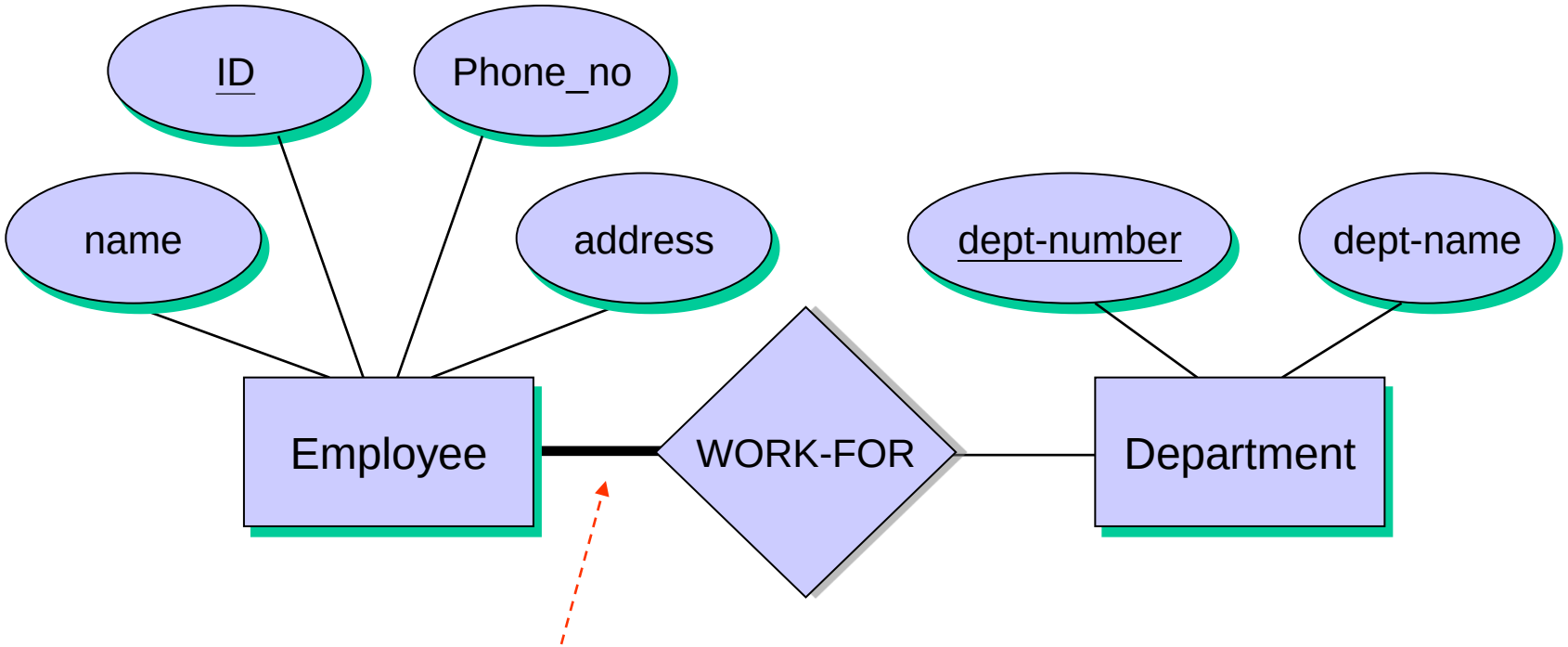
Participation constraints

- The key constraint on Manages tells us that a department has at most one manager.
- A natural question to ask is whether every department has a manager.
- Suppose every department is required to have a manager. Such a constraint is an example of participation constraint.

- In short, a participation constraint imposes some requirements on the number of times an entity participates in a relationship.
- We can classify participation in relationships as follows:
  - total - each entity in the entity set must participate in at least one relationship.
  - partial - an entity in the entity set may not participate in a relationship.

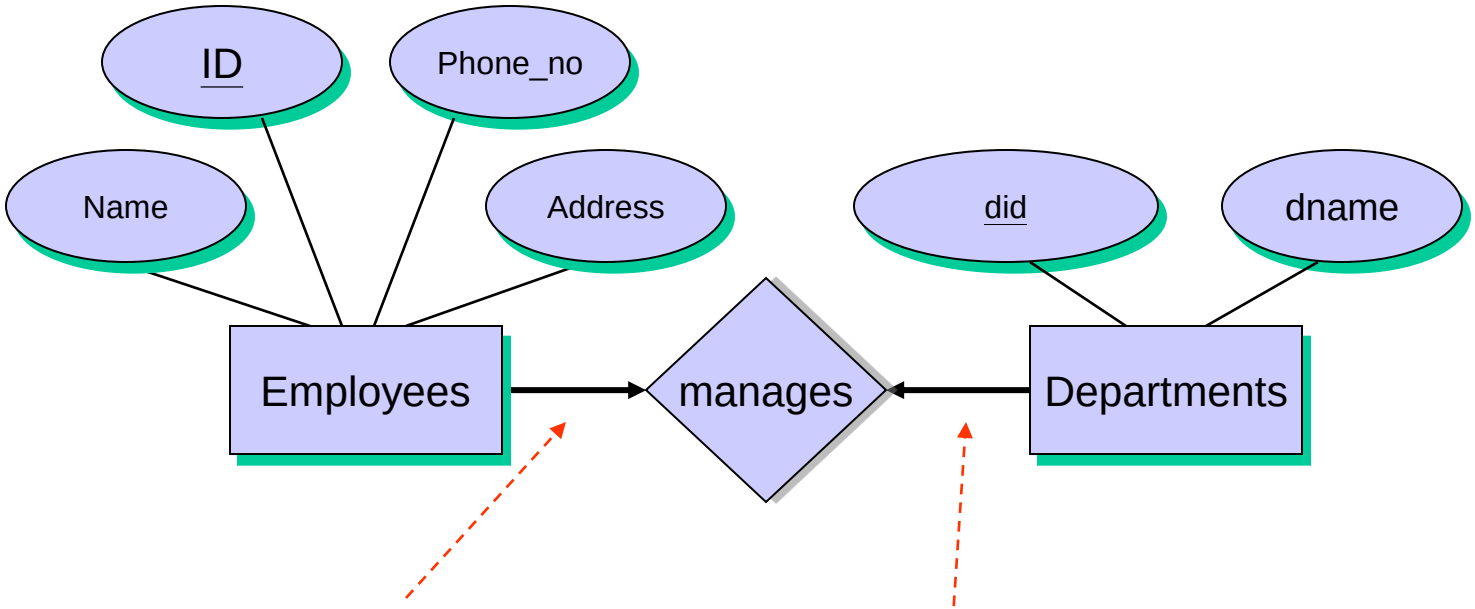
- E.g. Every employee must work for some department. The participation of EMPLOYEE in WORKS-FOR is total participation





If the participation of an entity set in a relationship set is total, the two are connected by a thick link; independently, the presence of an arrow indicates a key constraint.

In this case, the thick line means every employee has to participate in the work-for relationship.
i.e., every employee has to work for some departments (at least one department).



An employee can associate with at most one department via the relationship "manages"?

An employee can associate with at least? one department via the relationship "manages"?

A department can associate with at most one employee via the relationship "manages"?

A department can associate with at least one employee via the relationship "manages"?

Relationship properties

- Will each instance participate in this relationship?
- Relationships with a special property to be expressed.
  For example, all university students must be enrolled to university/universities.
- So far, entities don't have to participate in the relationships they are in.
  By default, participation is ***partial***

Constraints on relationship types

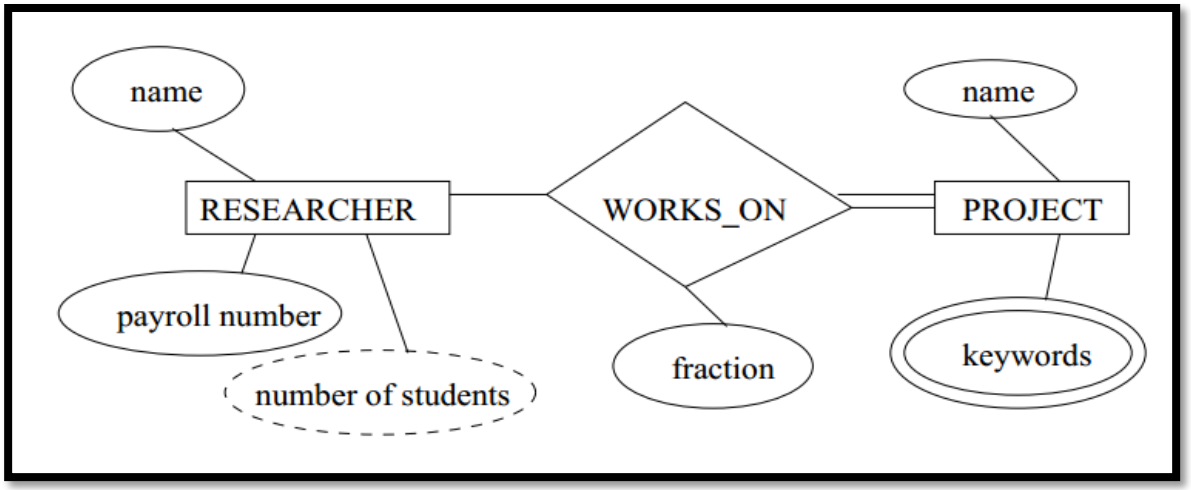
- **Total** participation: each entity instance must patriciate in at least one relationship instance.
- Example: We want this relationship to express all publications must be written by a person.

Constraints on relationship types

- e.g. every publication has an author.
- There can’t be a publication without an author
- e.g. not every person has publications.
- There can be people who don’t write publications

Attributes (Relationship)

- A researcher may work on several projects.
- The fraction of her time devoted to a particular project could be an attribute of the WORKS ON relationship type.
- Q: Why not put the attribute on either researcher or project? (on Wednesday)

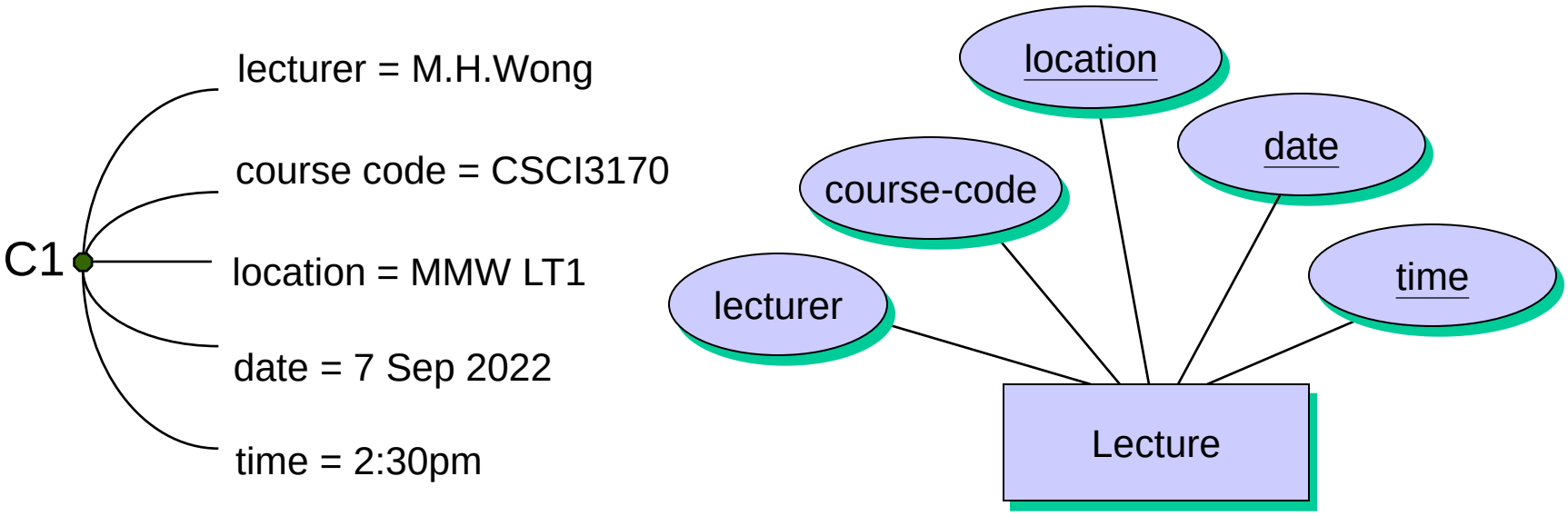


Mentioned in the Tut: Key

- A superkey is any set of attributes which can uniquely identify an entity.
- A key is a minimal set of attributes whose values uniquely identify an entity in the set.
- A key is also called a candidate key.
- There could be more than one candidate key.
- A primary key is a candidate key chosen to serve as the key for the entity set.

No proper subset of the attributes can be a superkey

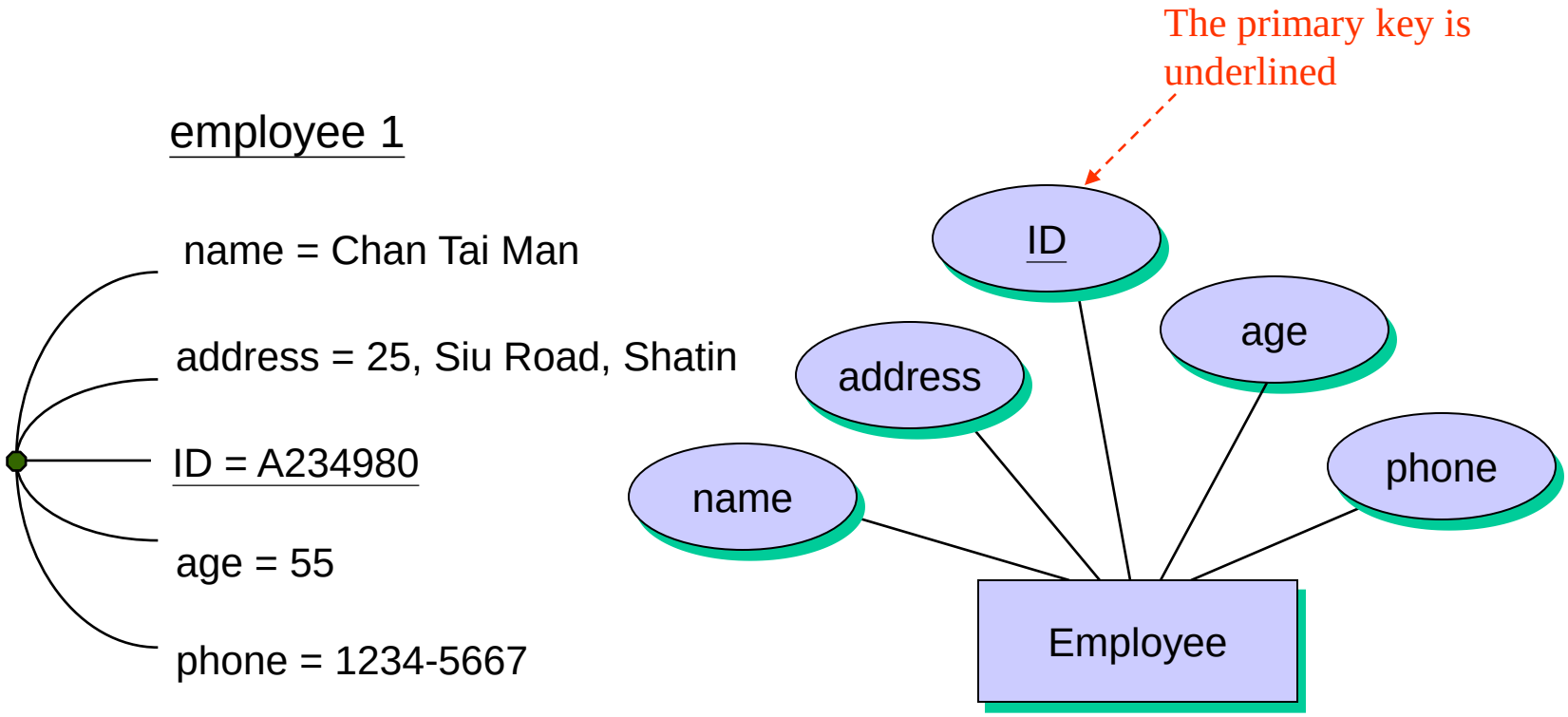
- A key may contain more than one attribute.



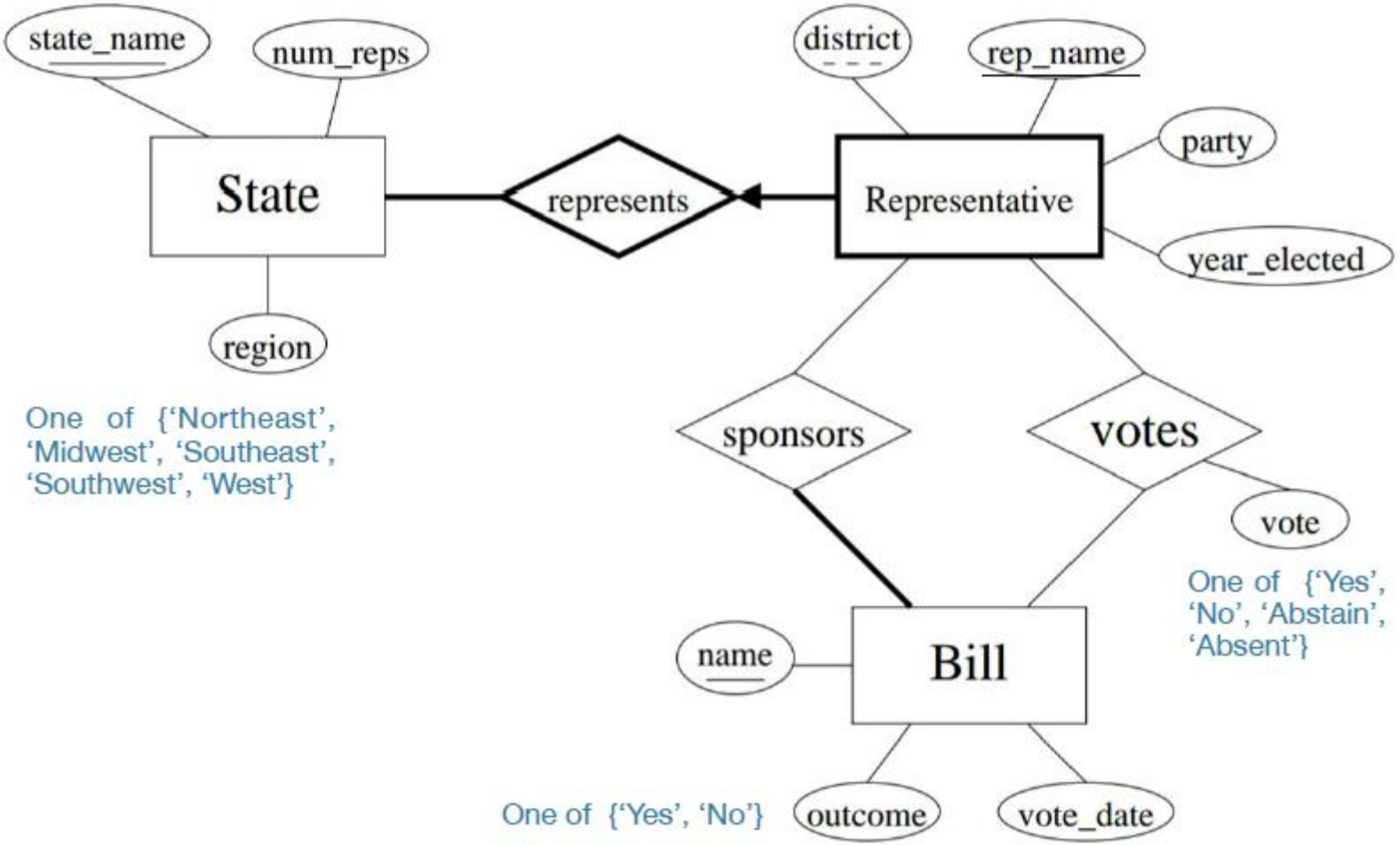
- For example, {location, date, time} is a key.
- {lecturer, location, date, time} is not a key, but it is a superkey.

- The key should depend on the real life possibility rather than on the current set of the data.
- For example, in the previous database which contains only two employees, the age can distinguish each employee. However, we may get a new employee with the same age as an existing employee.

- Usually, we need to add an extra attribute as a key.



Example



https://www.chegg.com/homework-help/questions-and-answers/following-er-diagram-chen-notation-track-information-votes-taken-us-house-representatives--q31481714

Psa, don't ask chegg for help, like AI that's also contract cheating.

Learning Outcomes

Today

- Entities and Attributes
- Relationships
  - One-to-One Relationships
  - One-to-Many Relationships
  - Many-to-Many Relationships

Wednesday

- Attributes
  - Key Attribute
  - Composite Attribute
  - Multivalued Attribute
  - Derived Attribute
- Weak Entity
- More on Relationships

Food for thought:
Conceptual Design with the E-R model

- Developing an ER diagram presents several choices, including the following:
  - Should a concept be modeled as an entity or an attribute?
  - Should a concept be modeled as an entity or a relationship?
  - What are the relationship sets and their participating entity sets? Should we use binary or ternary relationships?
  - Should we use aggregation?

Database design

- The database design process can be divided into six steps.

1. Requirement Analysis
   - What data should be stored?
   - What applications must be built?
   - What operations are most frequent and subjected to performance requirements?

Database design

2. Conceptual database design
   - High-level description of
     - ✓ Data to be stored
     - ✓ Constraints
   - Often carried out using the ER model
3. Logical Database design
   - Choose a DBMS to implement our database design.
   - Convert the conceptual database design into a database schema for the chosen DBMS.
     - Convert ER schema into a relational database schema.

Database design

4. Schema Refinement
   - Analyze the collection of relations in our relational database.
   - Identify the potential problems.
   - Refine the schema.
5. Physical database design
   - Ensure that the design meets the performance requirement.
   - Build index, clustering some tables etc.
   - Redesign parts of the database schema.

Database design

6. Application and security design
   - Write application programs.
   - Identify data that can be accessible by certain types of users.
   - Take steps to ensure that access rules are enforced.