Introduction to Operating Systems and SQL for Data Science

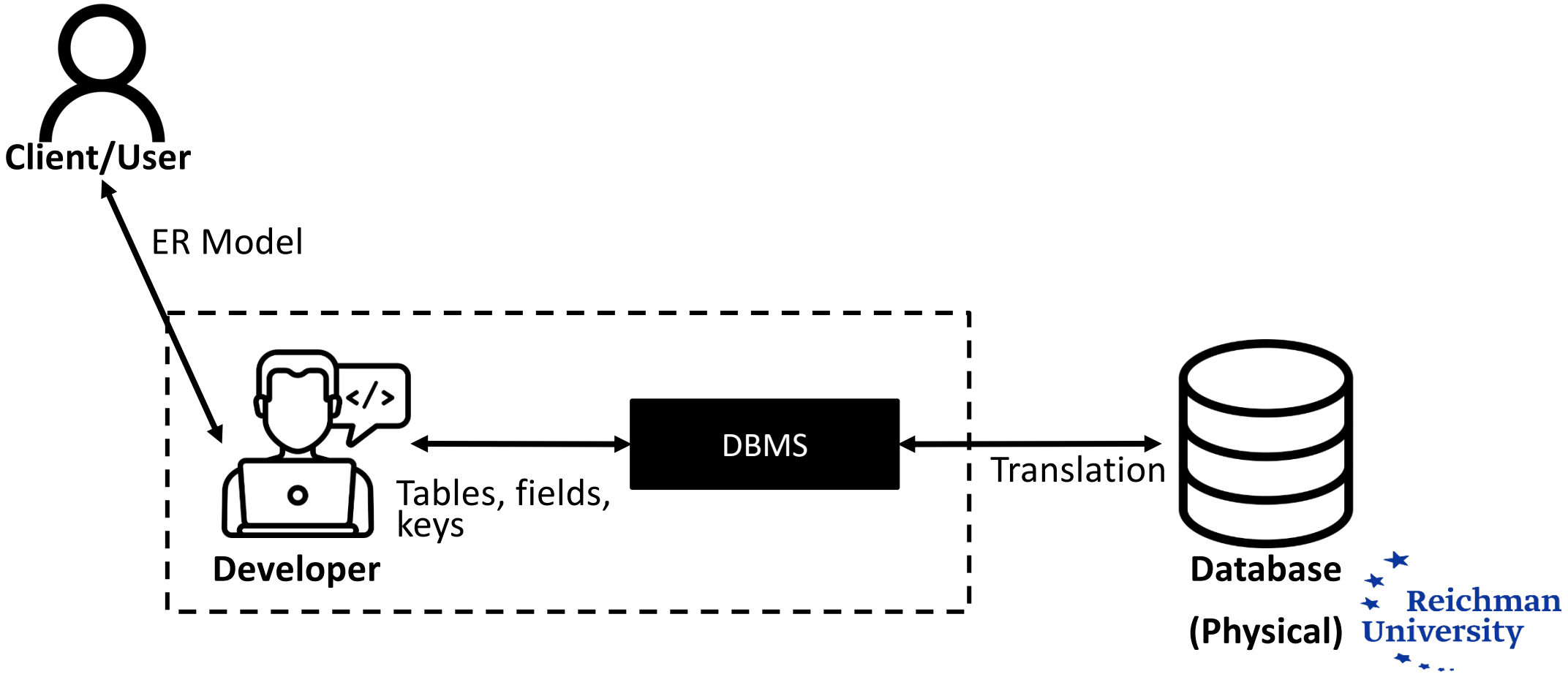
Lecture 11 – Conceptual model (ERD)

Previous lecture

- Normalization
- 1NF, 2NF, 3NF
- Functional dependencies
- Lossless-join decomposition
- BCNF

DBMS – the bridge between the logical to physical

How to build a data base to a real-world problem?



Reichman University

Reminder: design stages of databases

- Conceptual design:
  - What data should be stored in the database and the relationships between them.
  - Does not include to the implementation of the system.
- Logical design:
  - How to represent the conceptual design in the structure of a particular DBMS.
  - For example, how to represent the data and the relationships between them in RDBMS.
- Physical design
  - Translation of the logical design into a physical structure.
  - Takes into account performance constraints, storage volumes, indexes, etc.

Conceptual design

How to perform a conceptual design

- What are the goals of conceptual design?
  - **Understanding the structure** of data in reality (in the organization).
  - **Representation of the data structure** using an abstract model (No "computer" terms)
  - **Communication tools** between clients/users and developers.

- What is the product of conceptual design?
  - Conceptual model: a description of the information we want to keep.

Reichman University

Desirable features for a conceptual model

- Expressive: Allows expression of a variety of data, relationships and constraints.
- Simplicity: Easy to understand even for non-professionals (users).
- Minimalism: Includes a small number of basic terms that are simple and clear.
- Diagrammatic representation: Can be described by a simple diagram.
- Formality: Can be formally / accurately described.
- Accurate (algorithmic) mapping to a logical schema of the database.

Reichman University

Entity Relation Diagram (ERD)

Entity Relation Diagram (ERD)

- A common conceptual model.
- Developed by Peter Chen in 1976.

Peter Chen

- It is a graphical model that represents the **information system as a collection of entities and of relationships between entities**.
- The model provides a tool for designing the database based on the information template in the organization

Reichman University

Entity

An entity represents in the model a tangible or abstract object that has an existence and has a meaningful interest in a particular context.

- Examples:
  - Object (building, book, car, machine factory, item in stock).
  - Person[or any other living thing](factory worker, student, hospital patient, animal).
  - Abstract concept or idea (course, driving test, flight, employment).
  - Event (attendance report, bank account transaction, private entry into stock)
- An entity has attributes

Reichman University

Attribute

- An attribute value is defined as the attribute content at a point in time.

- Examples
  - The value of the attribute height in the object "Avi Levy" is 1.80 meters.
  - The value of the checking account feature in account number X is 2,200 NIS.

Reichman University

Types of attributes

- A simple attribute
  - Example: Student ID
- Composite attribute
  - Example: Address - including city, street, house number and zip code
- Multi-valued attribute
  - Example: Phones, grades
- Derived attribute
  - Example: The age attribute is derived from the year of birth attribute

Reichman University

Key

- An entity group key is defined as a collection of **minimal** attributes that unambiguously identify an instance of a particular entity in the entity group

- Examples
  - In the "Students" entity group: ID is the key
  - In the "Soldiers" entity Group: ID **or** identity number (military) is the key
  - In the group of entities “Grades”: student ID + course number + semester + year is the key
  - In the "Courses" entity group: course number **or** department number + course name is the key

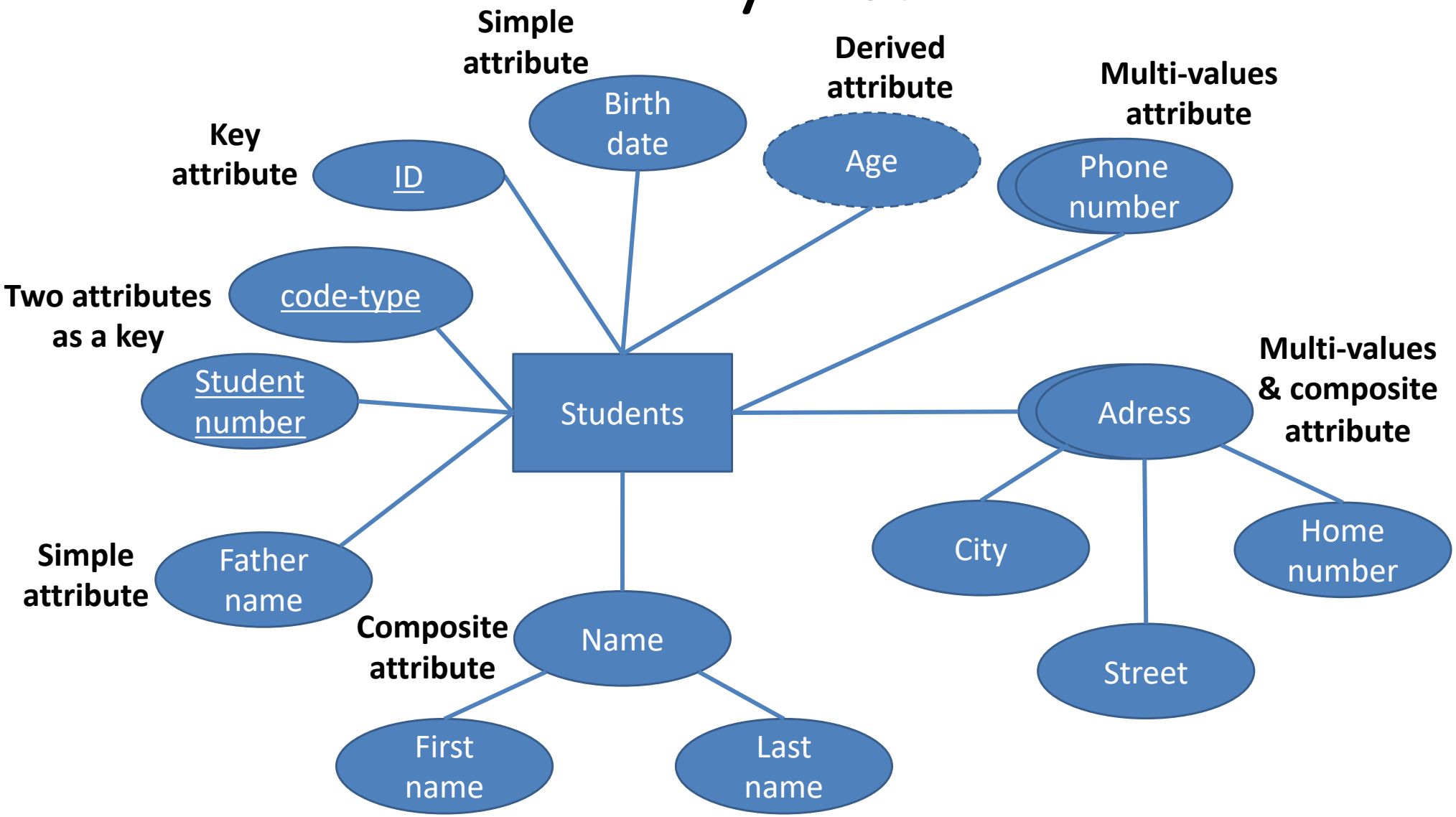
Reichman University

Is it a key?

- Is ID + student name is a key?
  - **No!** the key has to be minimal.

- Is the student name is a key?
  - **No!** because it does not unambiguously define each instance

My first ERD…



Reichman University

Connection

Connection is a meaningful relationship between entities.

Examples:

- Students **study** in courses
- Actors **participate** in movies

Reichman University

How relations looks in a diagram?

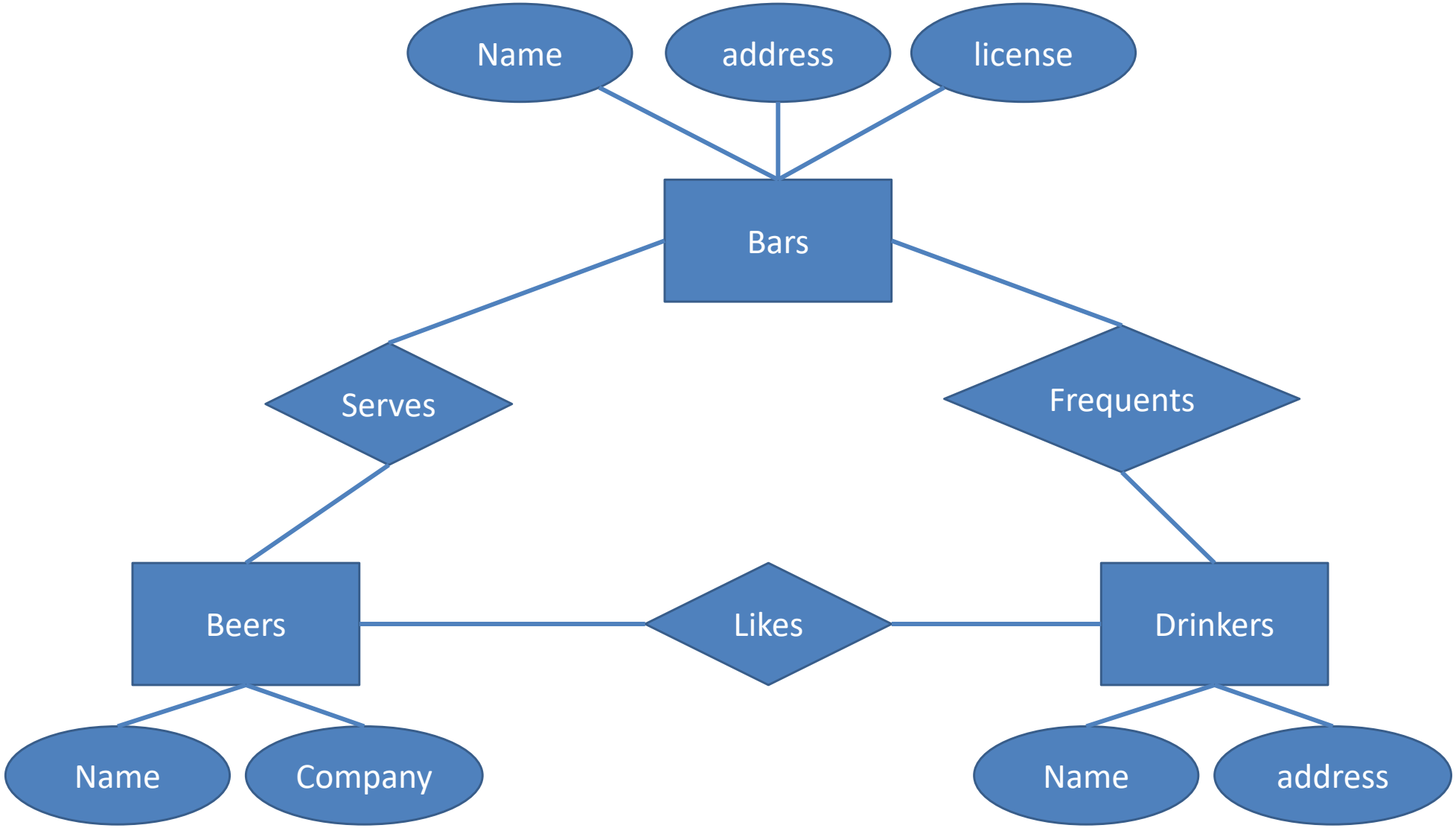
While an entity is modeled as a rectangle.

An attribute is modeled as an ellipse.

A relation is modeled as a rhombus.

Reichman University

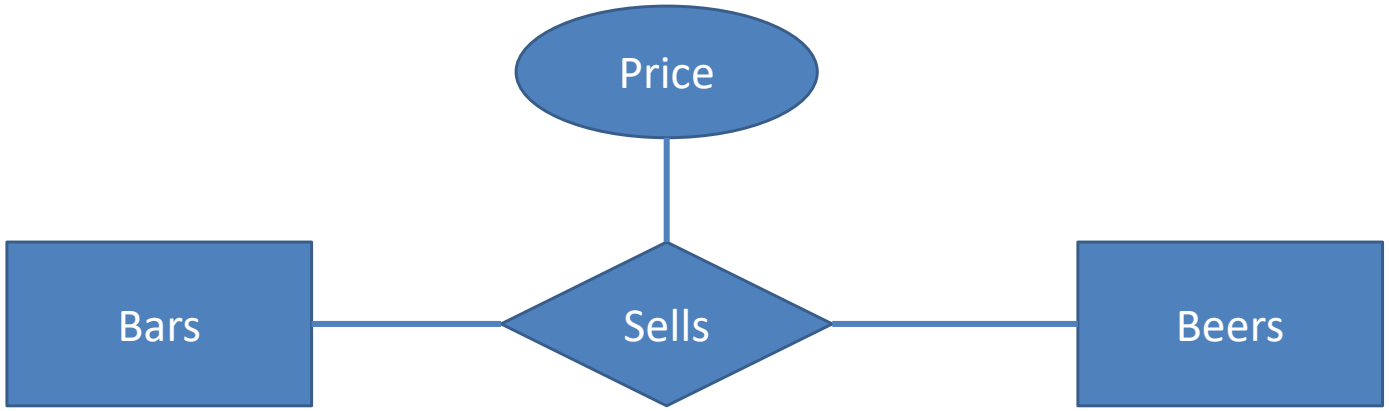
Example 2 - bars



Reichman University

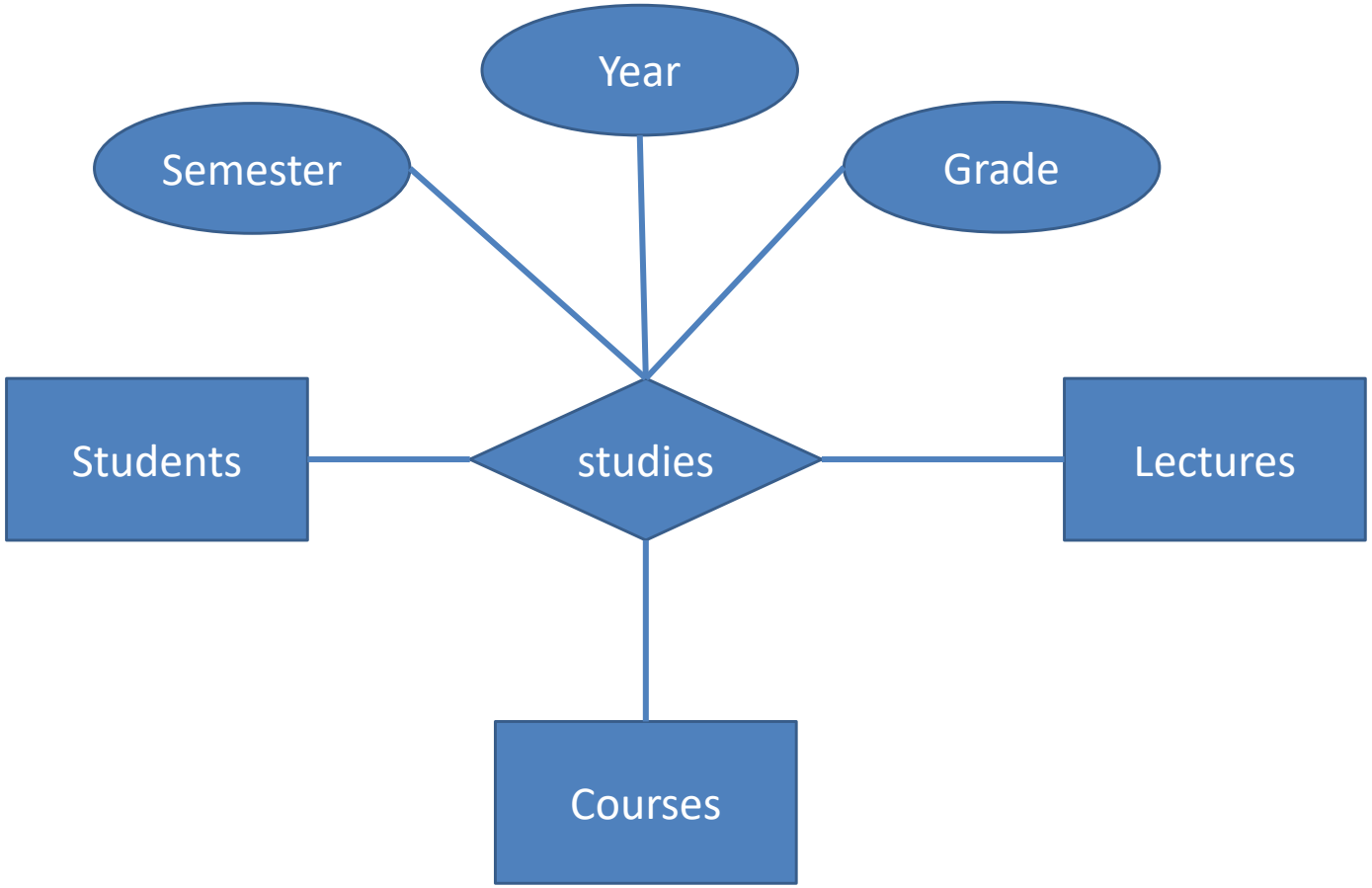
Informative connection

Sometimes we add features to the connection itself to add important information:



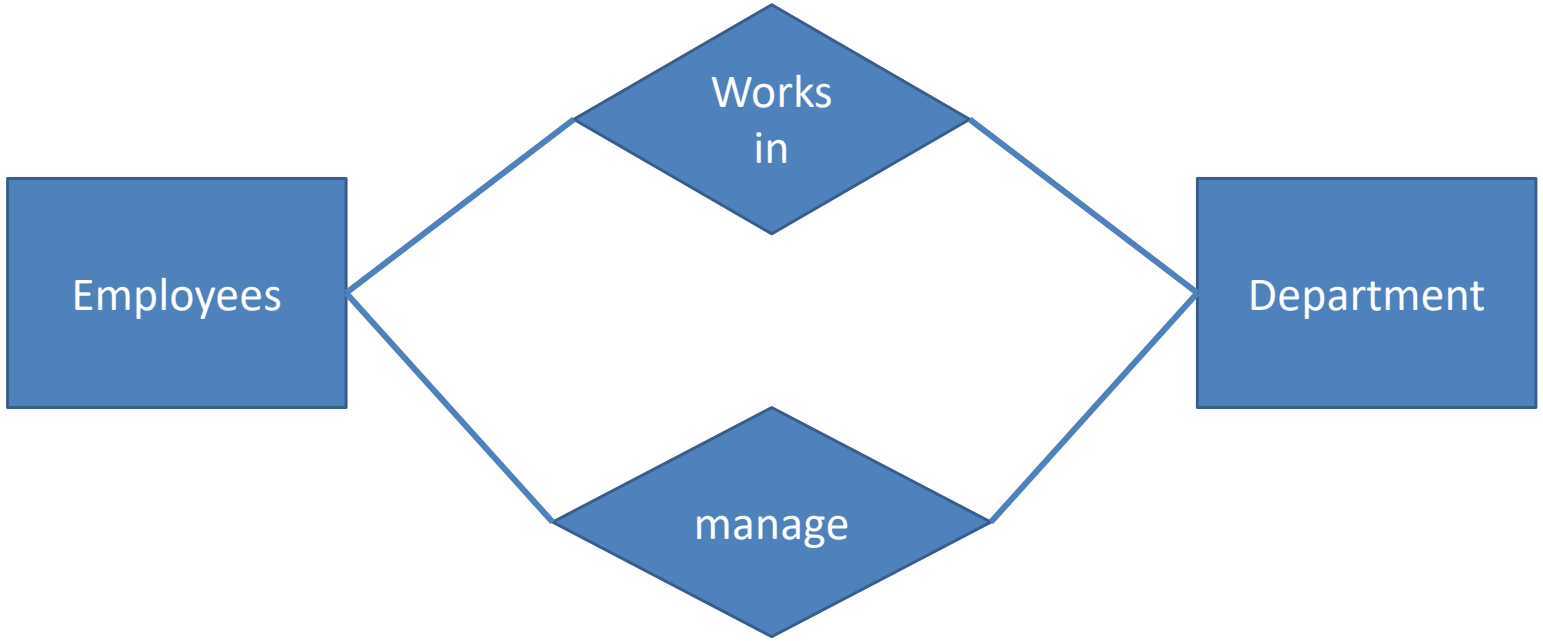
Reichman University

Example 2



Reichman University

Several connections between the same entities

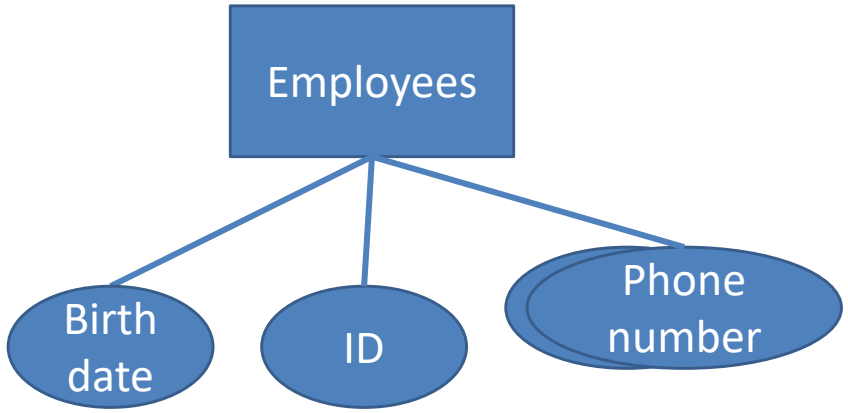


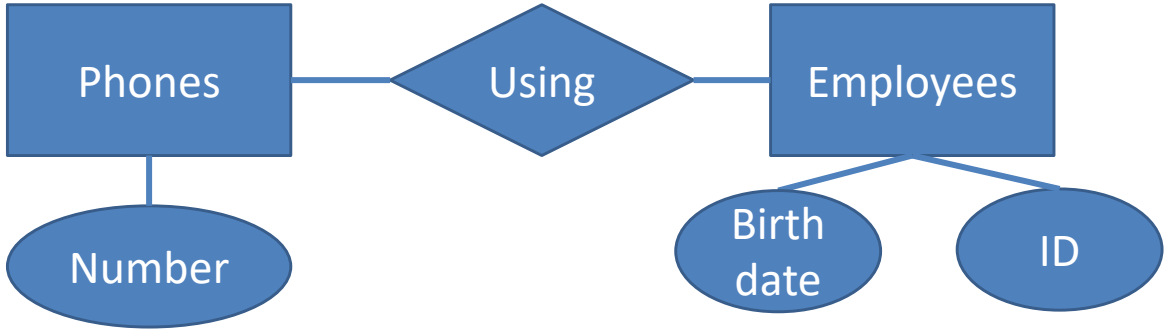
Reichman University

Connections vs. attributes

Example: for each employee there is one or more phone numbers.

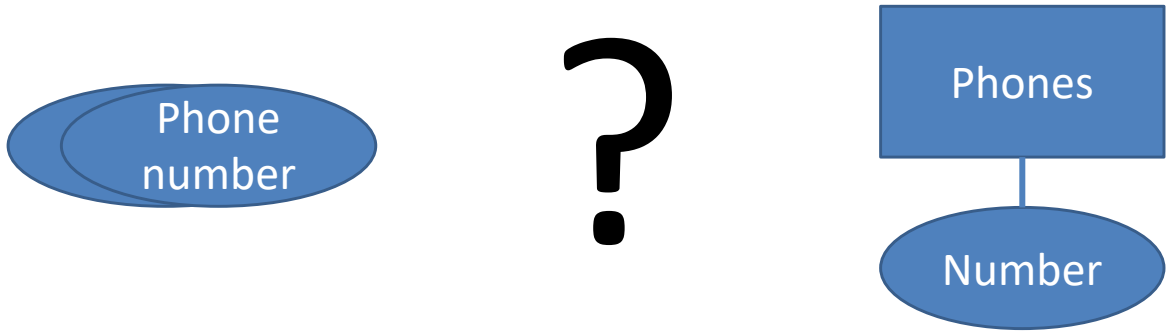
How will the ERD look?





Reichman University

When an attribute and when an entity?



Rule of thumb:

If there are some attrbitures -> then it's an entity

Otherwise -> a feature of another entity

Reichman University

Example

- In a system that deals with students in which all that is interesting to know about departments is their names.
- A department name will be defined as an attribute (of a student).
- If you are interested in more details about classes, then a class will be defined as an entity (that has at least two attributes).

Reichman University

Characterization of connections

Characterization of connections

Relationships can be characterized by:

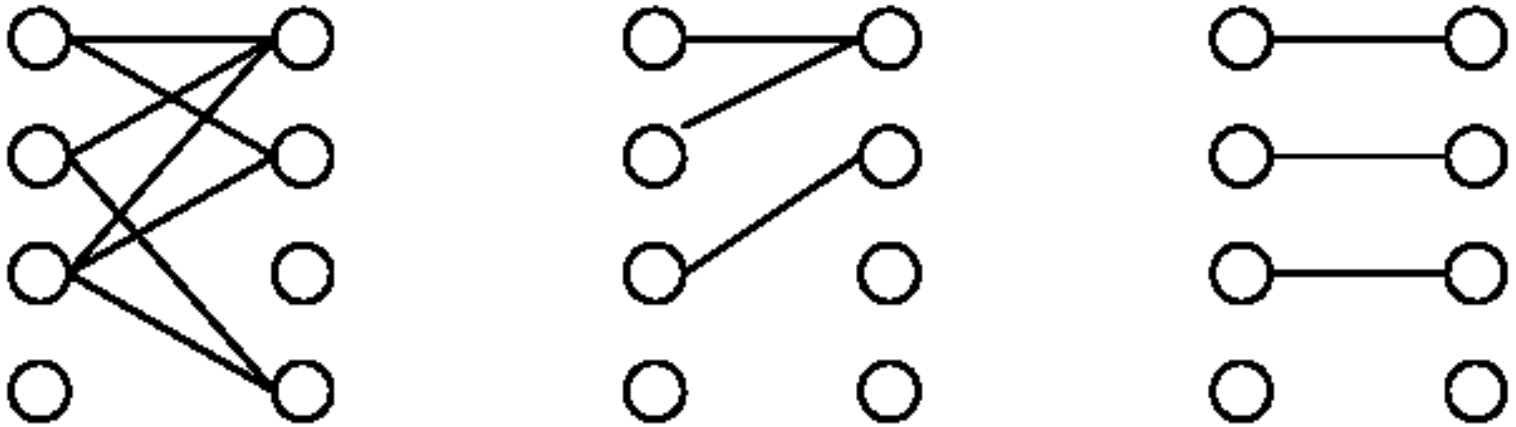
- **Connection type.**
- Connection cardinality.
- Connection degree.

Reichman University

Connection type

Examples:

- Passenger in the plane and seat in the plane.
- Student and department.
- Student and course.



Many-many Many-one One-one

Reichman University

Example to a 1-to-1 connection

A "one to one" connection exists between the "passengers" entity group and the "seat in flight" entity group.

why?

- Since each passenger has only one seat.
- Each seat has a maximum of one passenger.

Reichman University

How we represent a 1-to-1 in the diagram?

<u>Note there are arrows on both sides!</u>

Reichman University

Example to a one-to-many connection

The "One to Many" connection exists between the "Students" entity group and the "Departments" entity group.

Why? (Assuming a student can enroll in only one class)

- Because Each student belongs to only one department. (One)
- But each department has many(more than one) students. (Many)

Reichman University

How we represent a 1-to-M in the diagram?

<u>Note there is an arrow on the side of the One!</u>

Reichman University

Example to a many-to-many connection

The "Many to Many" connection exists between the "Students" entity group and the "Courses" entity group.

Why? Since-

- Each student can attend several courses.
- Each course has many(more than one) students.

Reichman University

How we represent a M-to-M in the diagram?

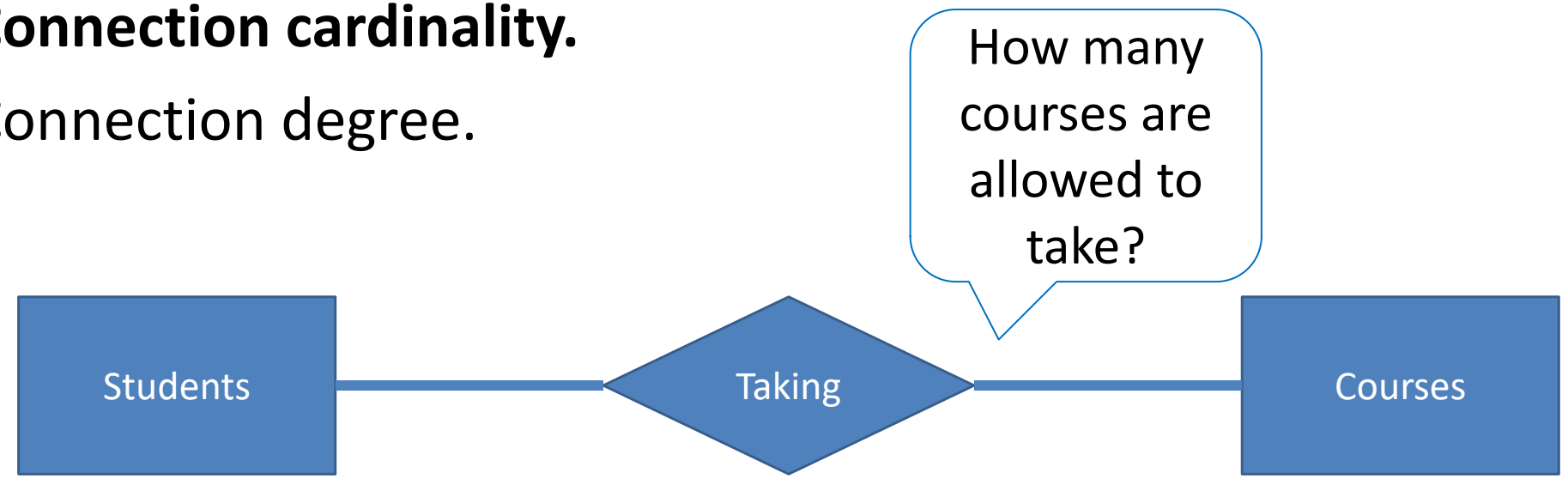
<u>Note there are no arrows on either sides!</u>

Reichman University

Characterization of connections

Relationships can be characterized by:

- Connection type.
- **Connection cardinality.**
- Connection degree.



Reichman University

Connection cardinality

Sometimes the type of connection is not enough, and more accurate details are required. For this we will use the cardinality of the connection.

Reichman University

Example a for connection cardinality

Each student can take between ? courses.

Each course must be attended by at least ? students and no more than ? students.

Reichman University

Example b for connection cardinality

Each passenger has exactly one seat.

Each seat on the flight can have between 0 and 1 passengers.

Reichman University

Example b for connection cardinality

There are those who give up the arrow when cardinality appears.

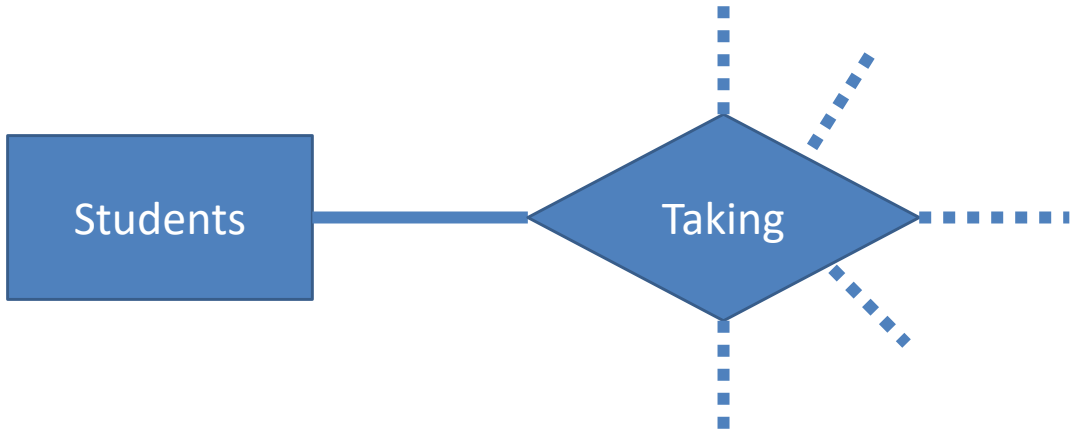
Reichman University

Characterization of connections

Relationships can be characterized by:

- Connection type.
- Connection cardinality.
- **Connection degree.**

A connection can connect several entities



Reichman University

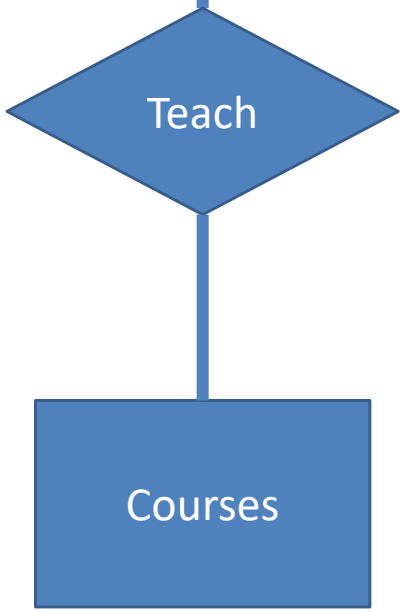
Connection degree

The degree of connection is defined as the number of entities participating in that connection.

Reichman University

Students, lecturers, and courses

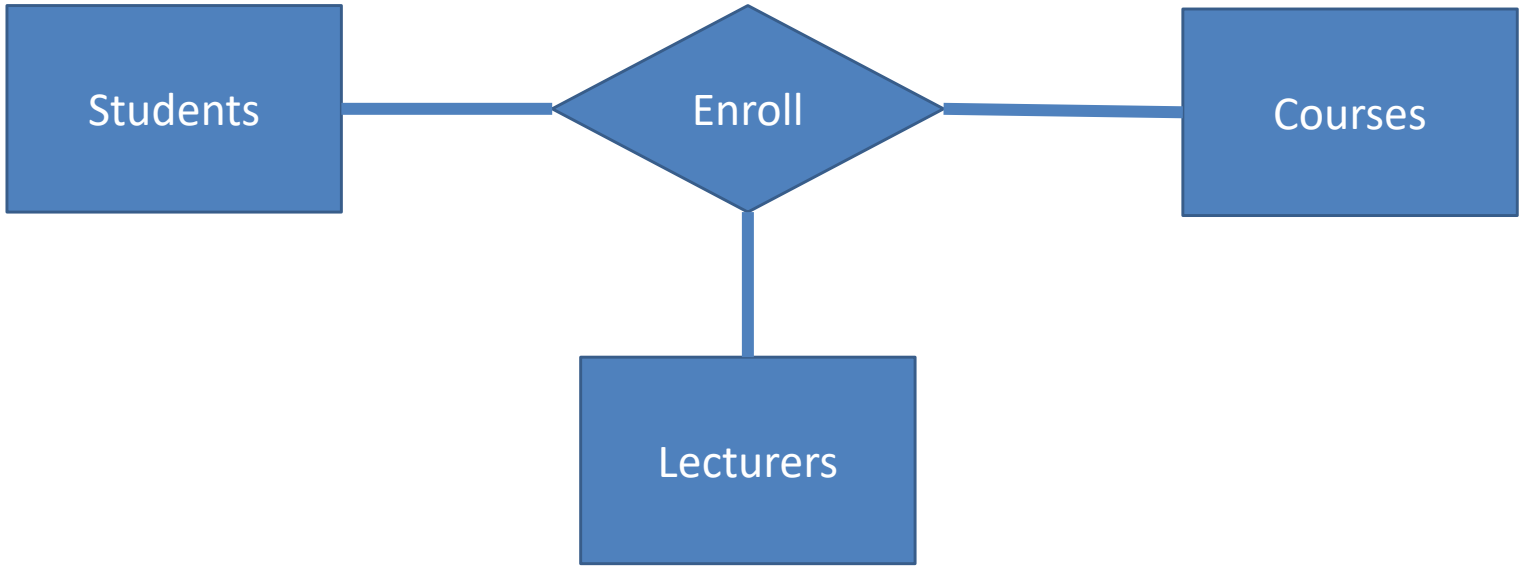
What is the problem?



Reichman University

Ternary connection

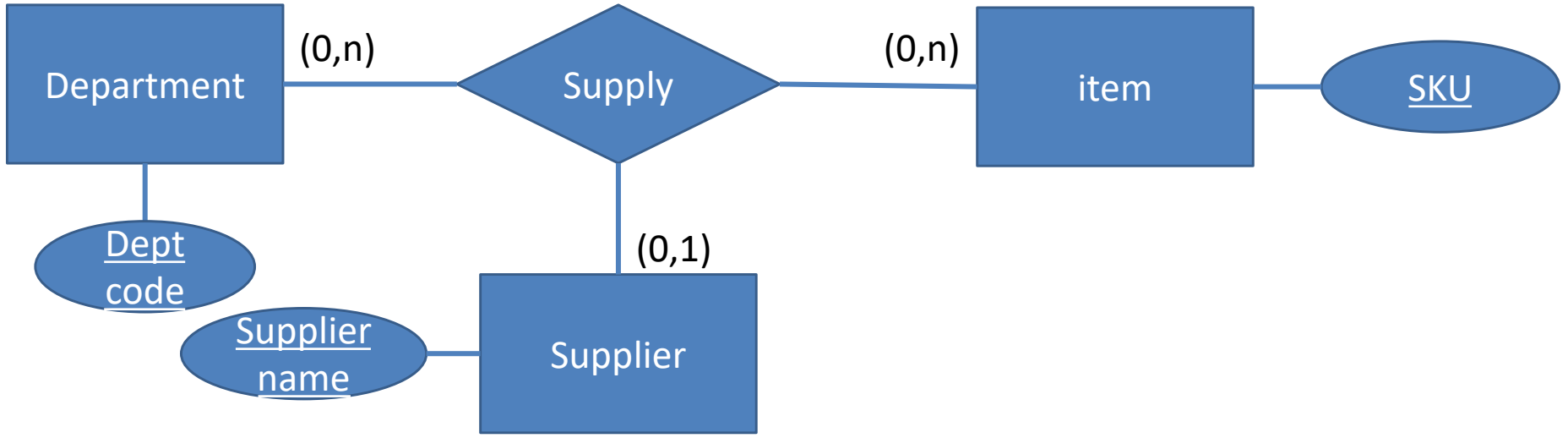
Ternary connection



- Each course may have more than one lecturer.
- Therefore, a connection between student and lecturer should be explicitly stated.

Reichman University

Ternary connection cardinality



- Examine each entity in relation to the other 2.

- In this example there is a m:n:1 relationship:
  - A particular department can get a lot of items from a particular supplier.
  - A particular supplier can supply a particular item to many departments.
  - A particular department can get a particular item from a maximum of one supplier.

Reichman University

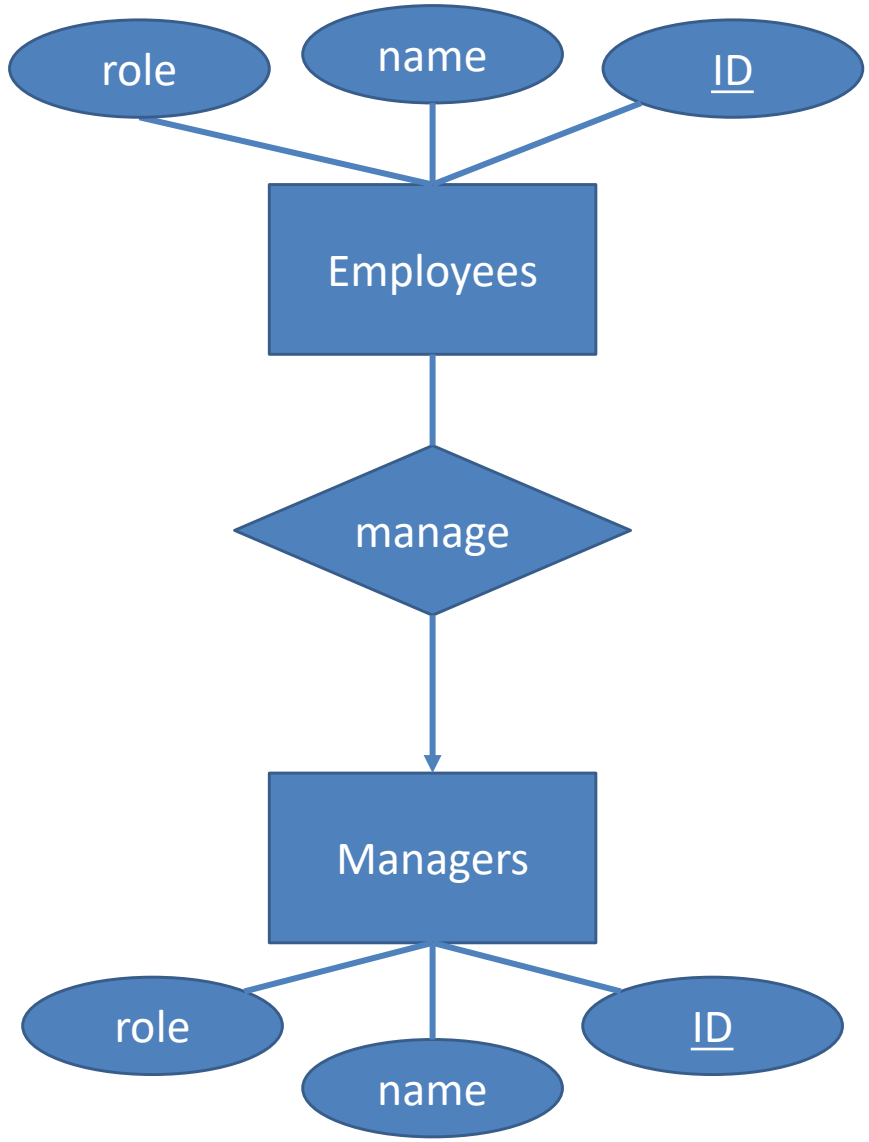
Unary connection

Unary(reflexive/recursive) connection

Sometimes it is necessary to connect one entity to another entity from the same group of entities.

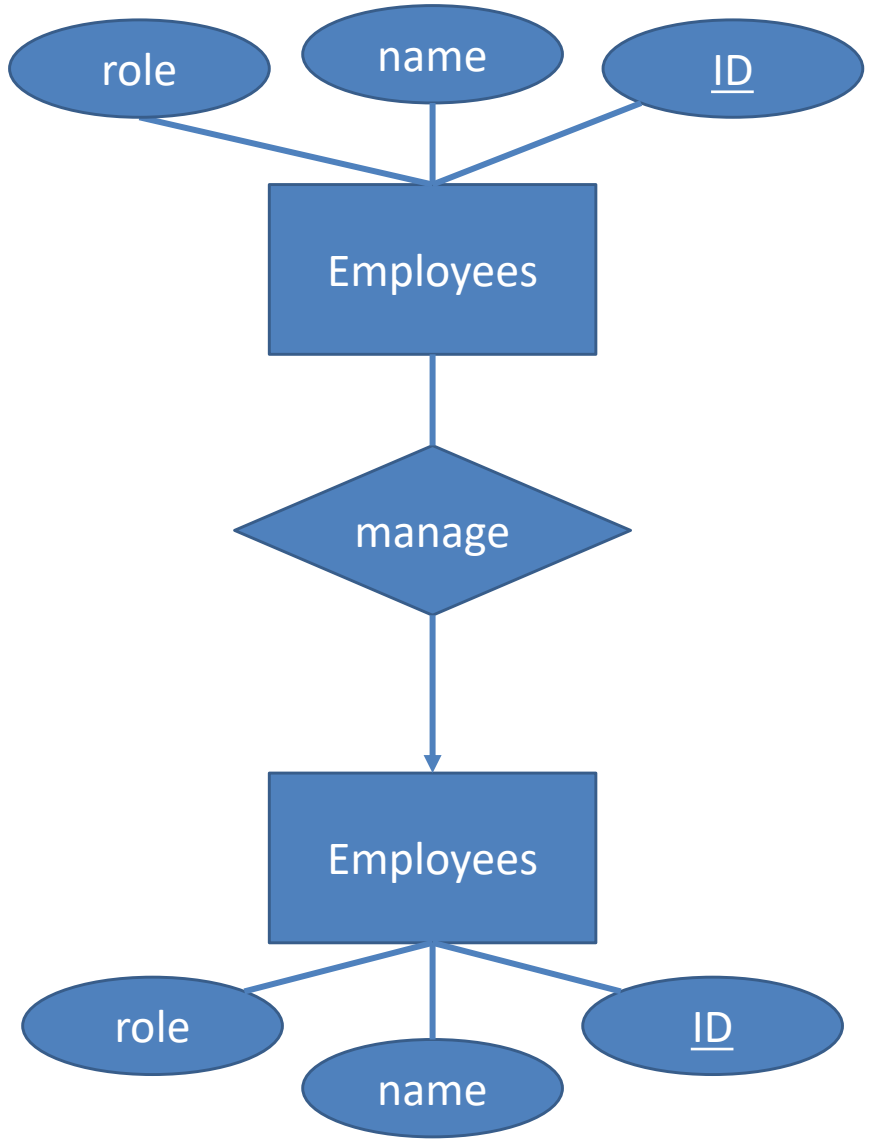
Reichman University

Unary connection - steps



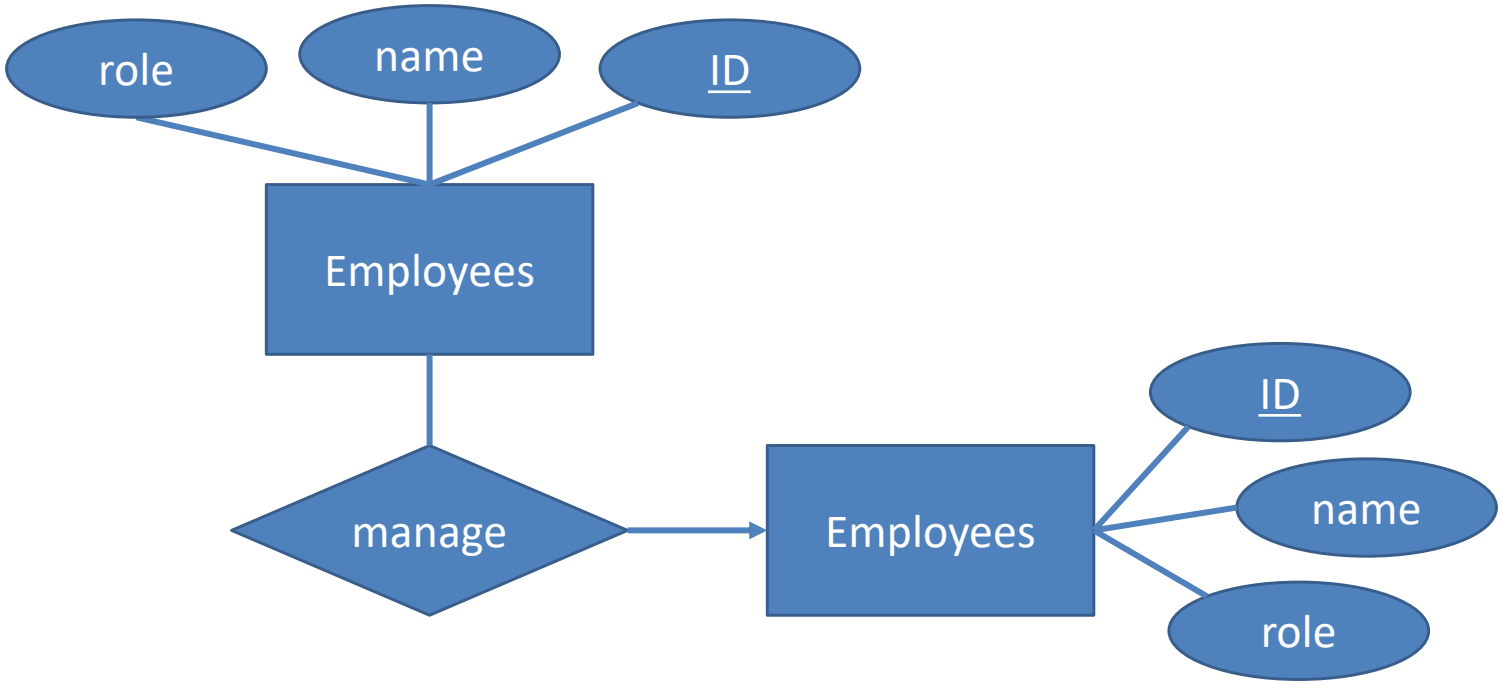
Reichman University

Unary connection - steps



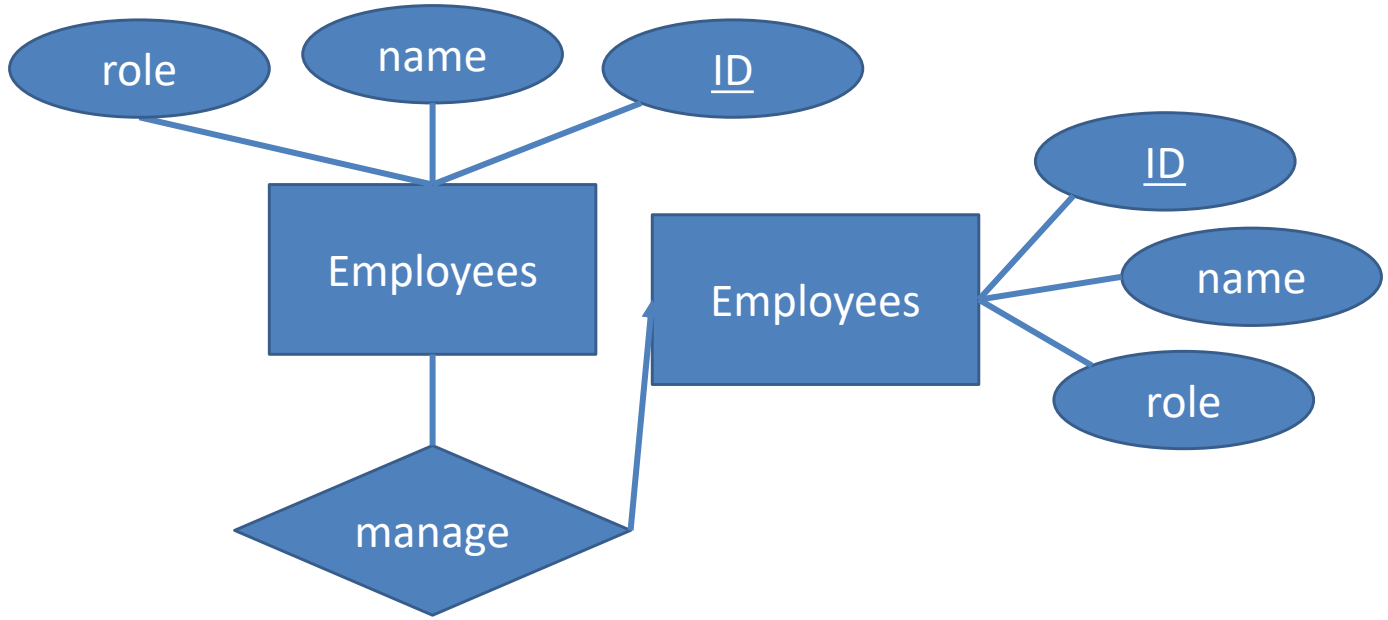
Reichman University

Unary connection - steps



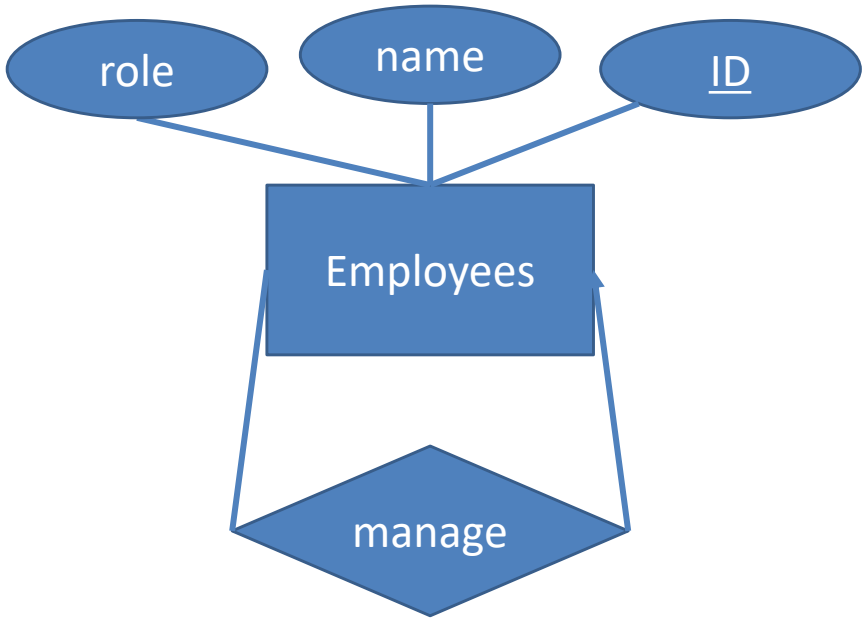
Reichman University

Unary connection - steps



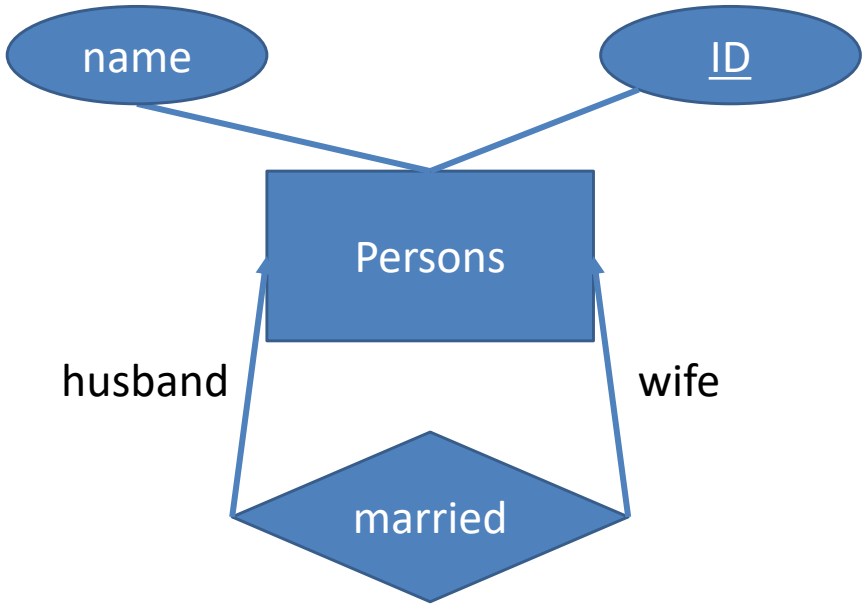
Reichman University

Unary connection - steps



Reichman University

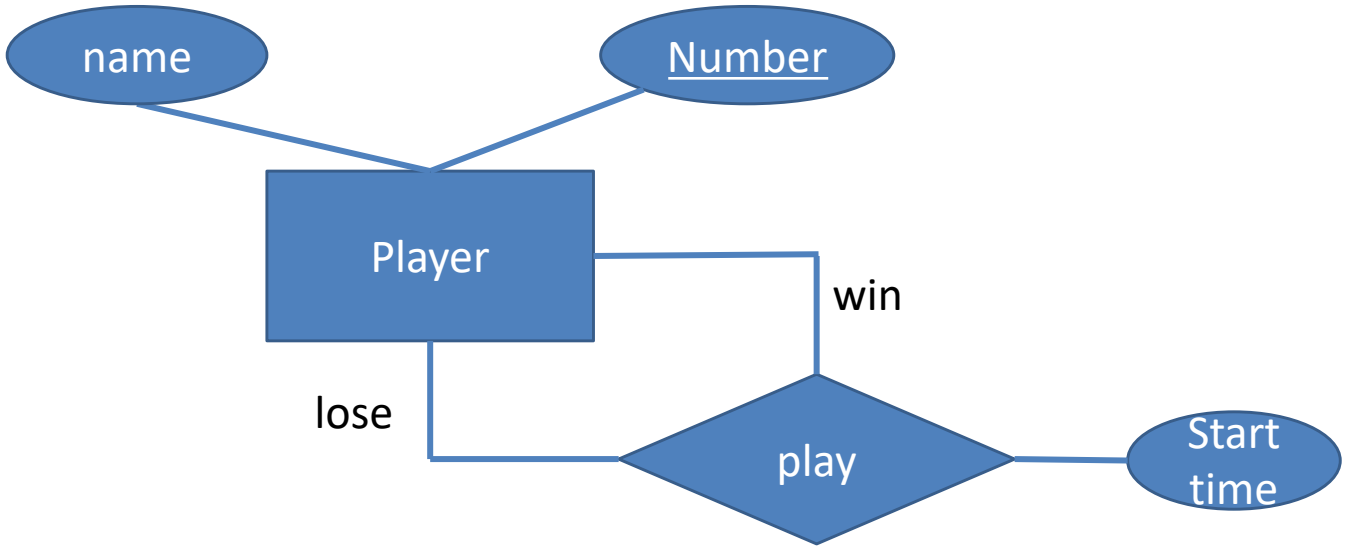
Unary connection example one-2-one



On the edges we will write the relevant role

Reichman University

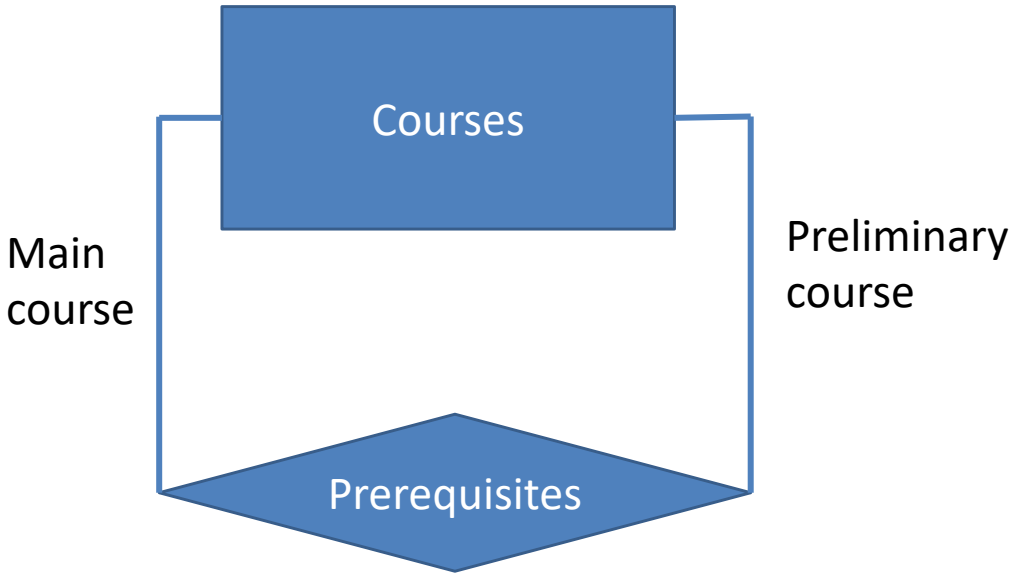
Unary connection example many-2-many



Confusing points: Each pair of players only plays once, and they have the same starting time.

Reichman University

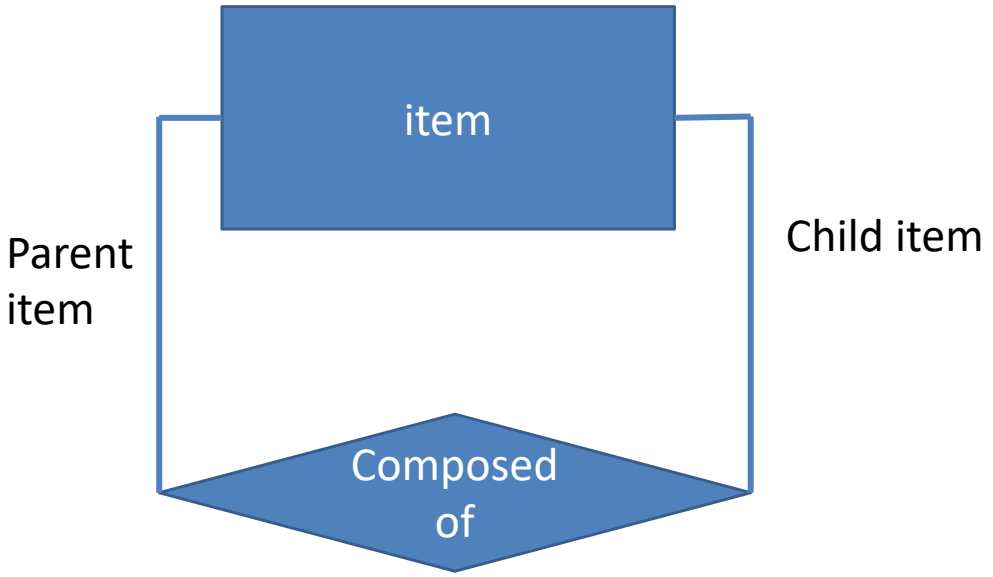
Reflexive connection example



This is a many-to-many relationship because each course can have a number of preliminary courses and each course can be a preliminary course for other courses.

Reichman University

Reflexive connection example – item tree



This is a many-to-many relationship because each item is composed of several items and each item can be used in multiple items.

Reichman University

(Interim) Summary

- Square - represents a group of entities (students / teachers ...).
- Ellipse - represents an attribute (academic year / academic degree ...).
- Rhombus - represents a connection between entities.
- Lines - linking entities to their attributes and groups of entities; to the connections in which they take part.

Reichman University

The marking standard in the diagram

- It is worth knowing that different groups use different markings to represent the same markings in the ERD.
- For example, different books - different markings.
- Different software - different markings.

- What matters is the meaning and not the form of the marking.

Reichman University

Extended ERD

- So far, we have studied the basic model of ERD.
- We will now get to know his familiar extensions.
  - Weak entity and bent entity
  - **Generalization**
  - Aggregation
  - And many more….
- In this course, we will be focusing only on generalization. (due to time limit)

Reichman University

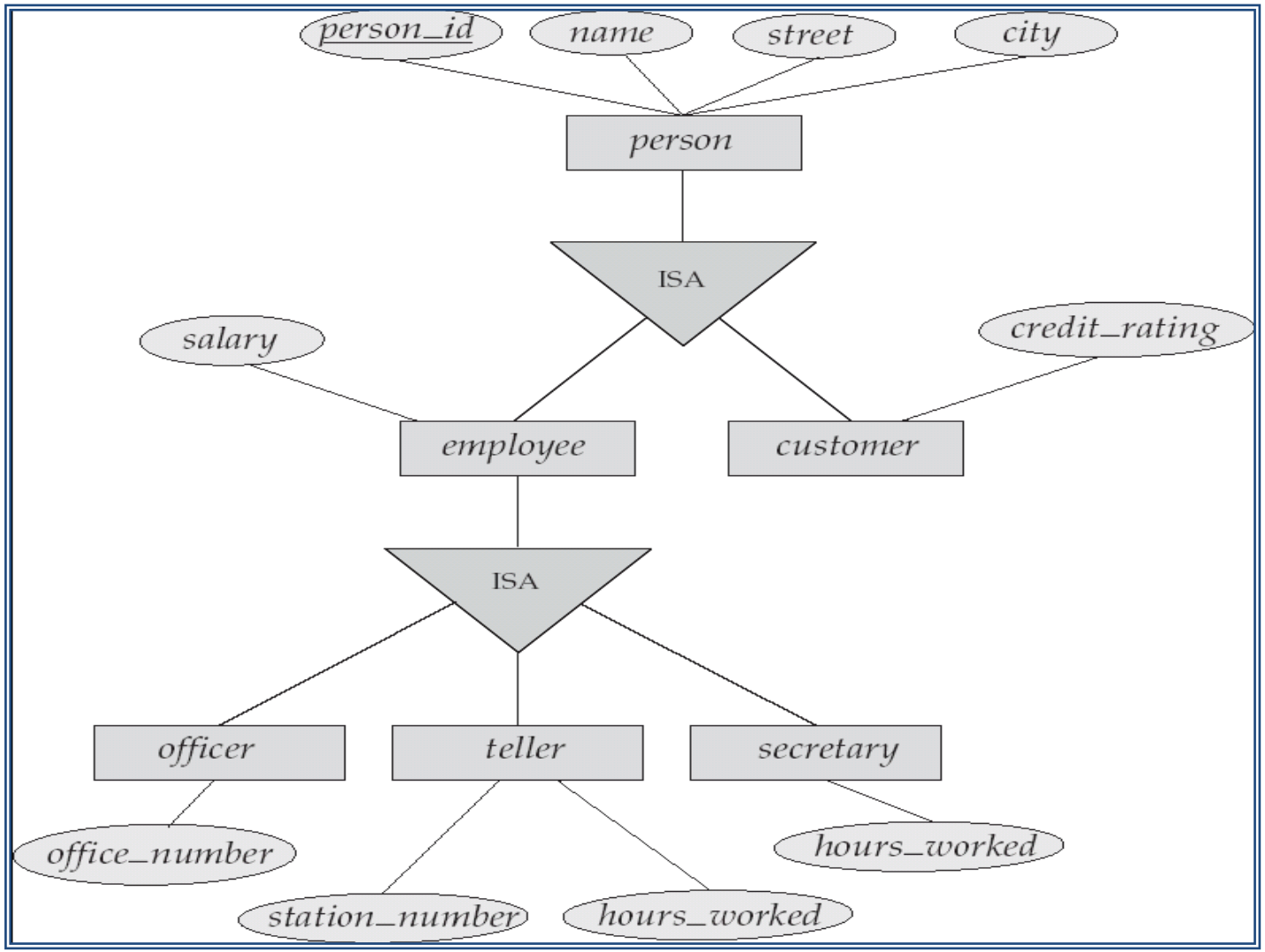
Generalization

Generalization

- Allows you to define a group of entities several groups of sub-entities that inherit the attributes from the group of super-entities.
  - The general entity group (employees) will be called a **high-level** group or a **basic** group.
  - The more specific entity groups (managers, engineers, clerks) will be called **low-level** groups or **derivative** groups.
- Example:
  - “engineers", "clerks" and "managers" are groups of sub-entities that inherit its behavior from the “employees“ entity group.

Reichman University

Generalization in ERD



Reichman University

Keys and attributes

- The sub-entity group inherits all the attributes and connection of the super group.
- The key is defined only in the super entity.
- Each sub-entity group can add its own attributes and connections.

Reichman University

Types of generalization

- Covering constraint
  - Could there be entities that are only of the parent type?

- Overlapping/Exclusive
  - Could an entity be of the type of more than one child?

Reichman University

Covering constraint

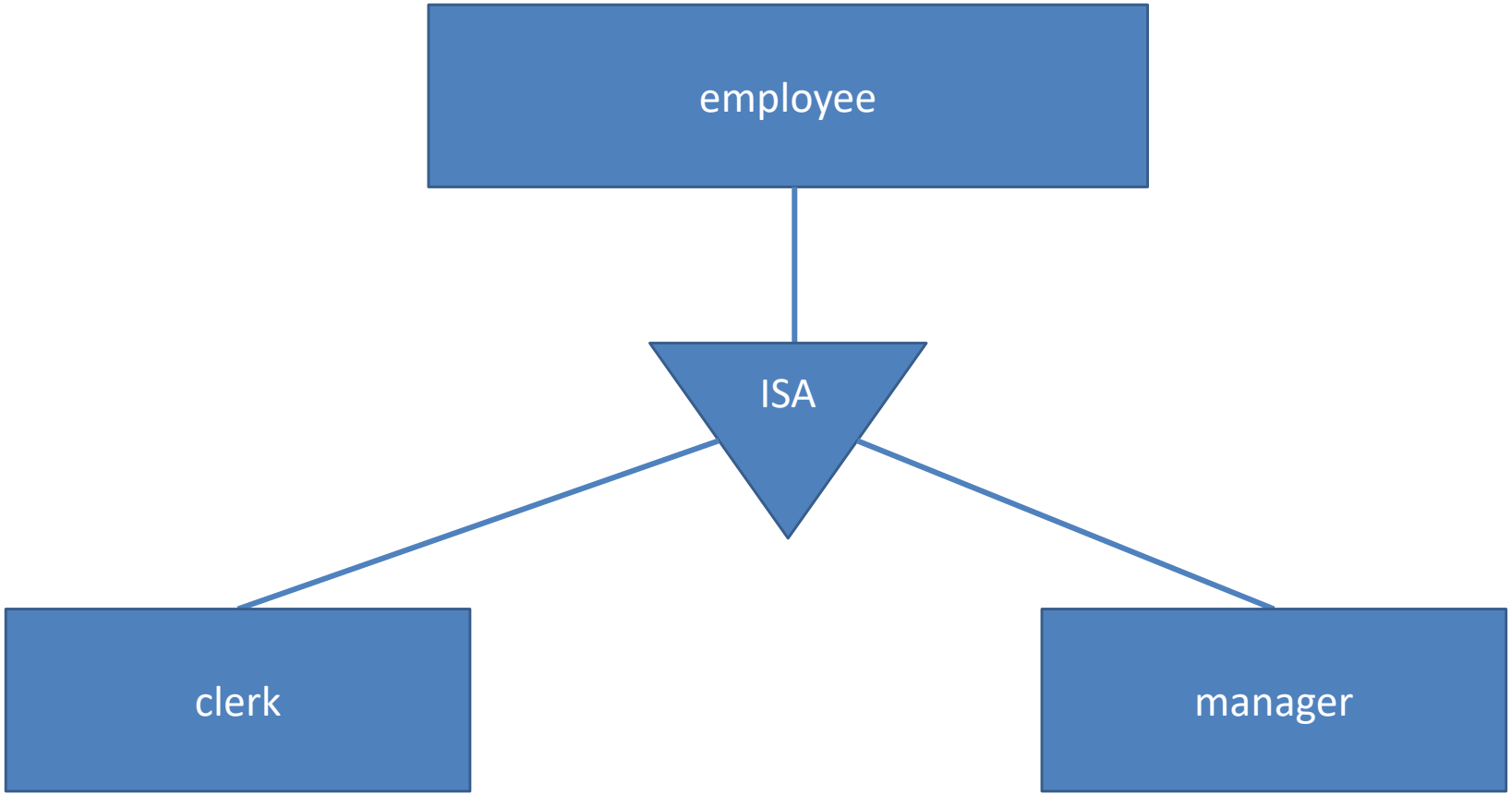
- Covering constraint defines whether the entities in the subgroups cover **all** entities in the super group.

- An example of a generalization connection that does not cover.
  - The “Employees" super group versus the "Clerk" and "Manager" subgroups.
  - An employee who is not a clerk or manager (for example an engineer).
  - In this case there may be a situation where a particular entity will belong only to the supergroup.

- An example of a generalization link that covers.
  - The super group “university employee" and the subgroups “academic faculty employee", “administrative faculty employee“.
  - Each employee must belong to one of the sub-groups.
  - To mark it we will use a double line

Reichman University

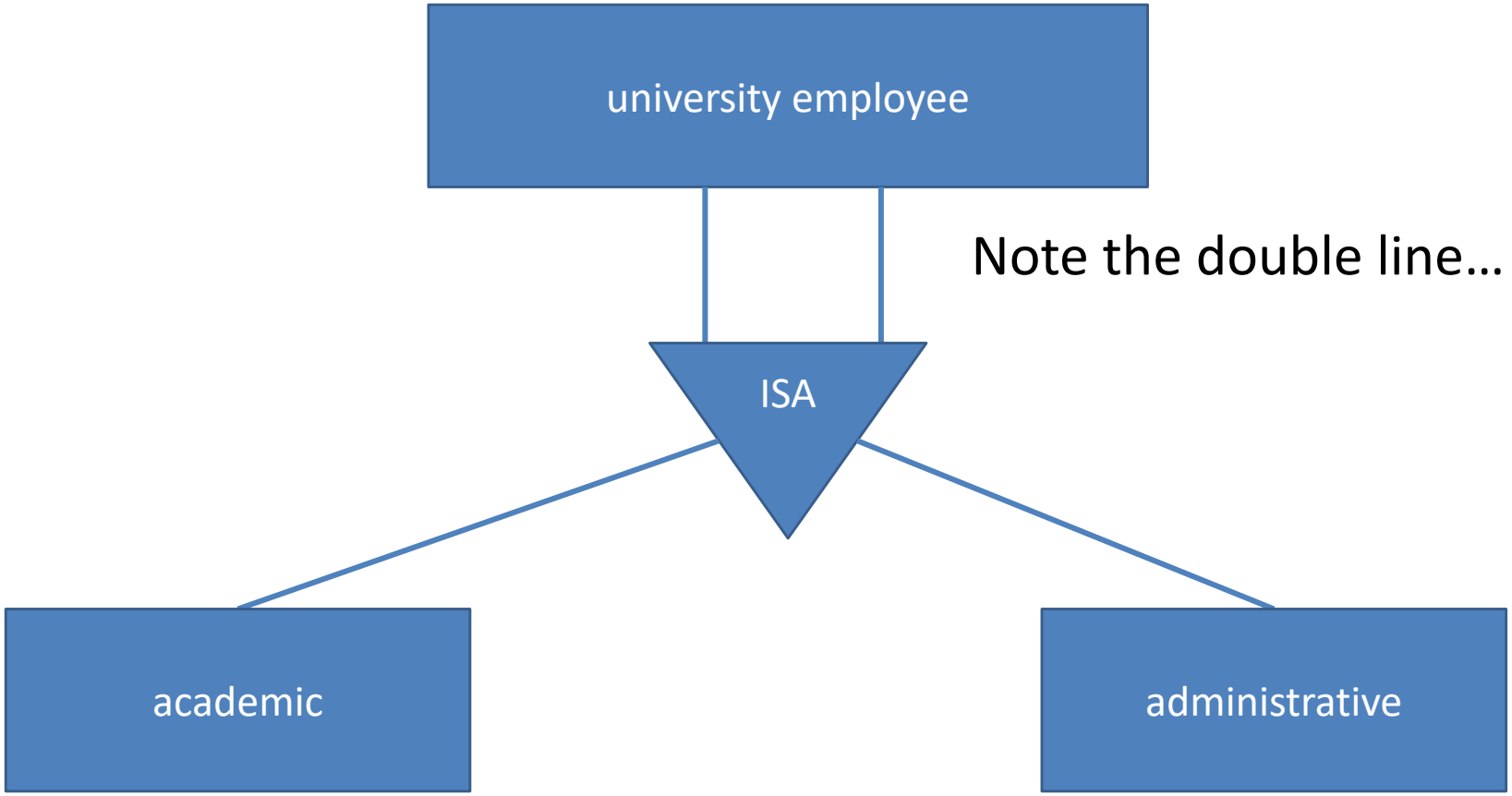
Covering constraint in ERD – not covers all



There could be an employee of a different type.
Such as, engineer for example…

Reichman University

Covering constraint in ERD – covers all



Note the double line…

There could not be a university employee of a different type.
One is either academic or administrative (or both) university employee.

Reichman University

Overlapping / Exclusive

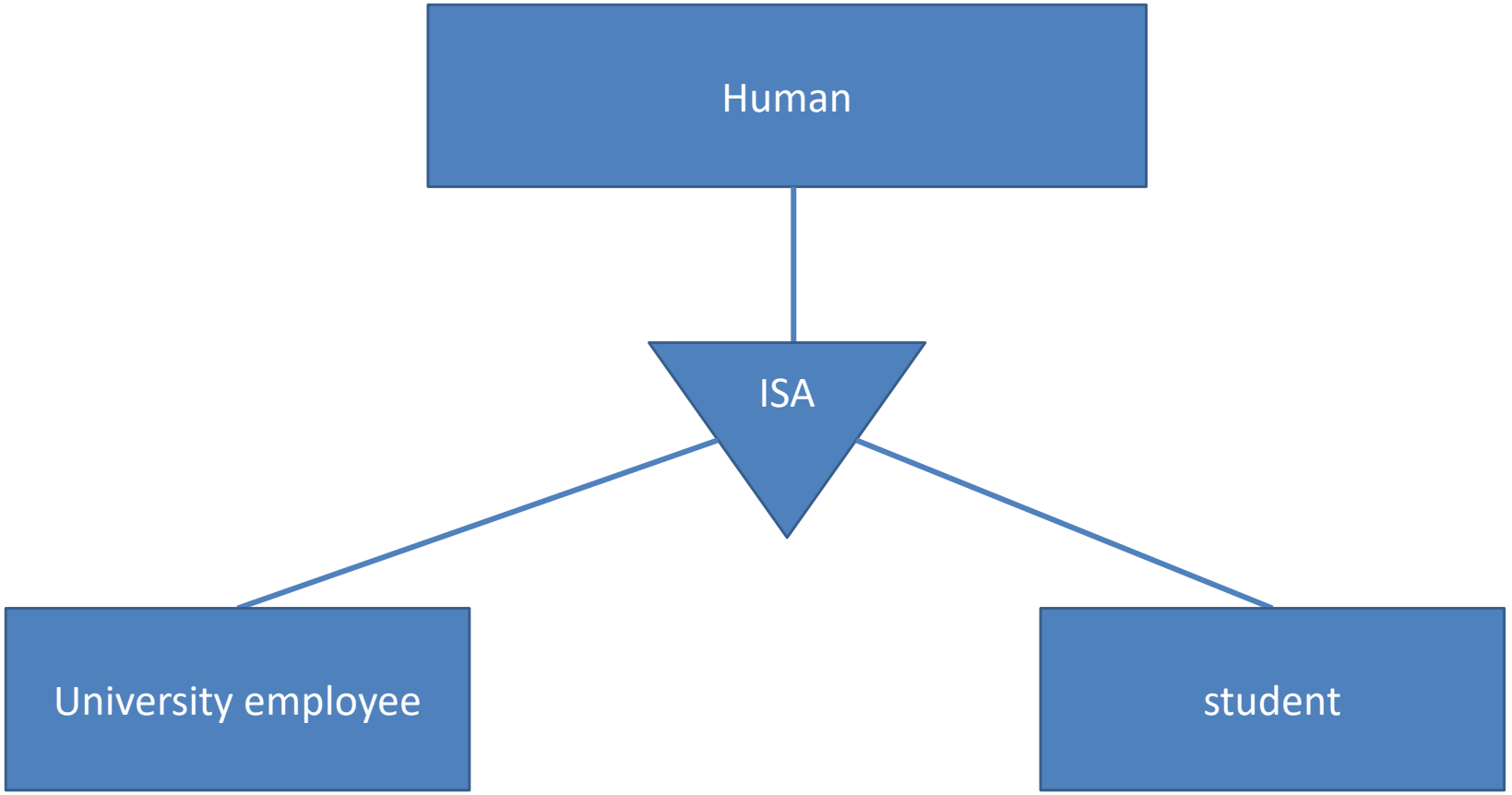
- Indicates whether a particular entity can belong to only one group or several subgroups.

- Overlapping
  - Example: the supergroup “Human" and the subgroups, "University employee" and "Student“.
  - The same "person" can be both a "university employee" and a "student".

- Exclusive
  - Example: the supergroup "Student" versus the subgroups "Undergraduate student" and "Research student“.
  - To mark it we will use the letter X.

Reichman University

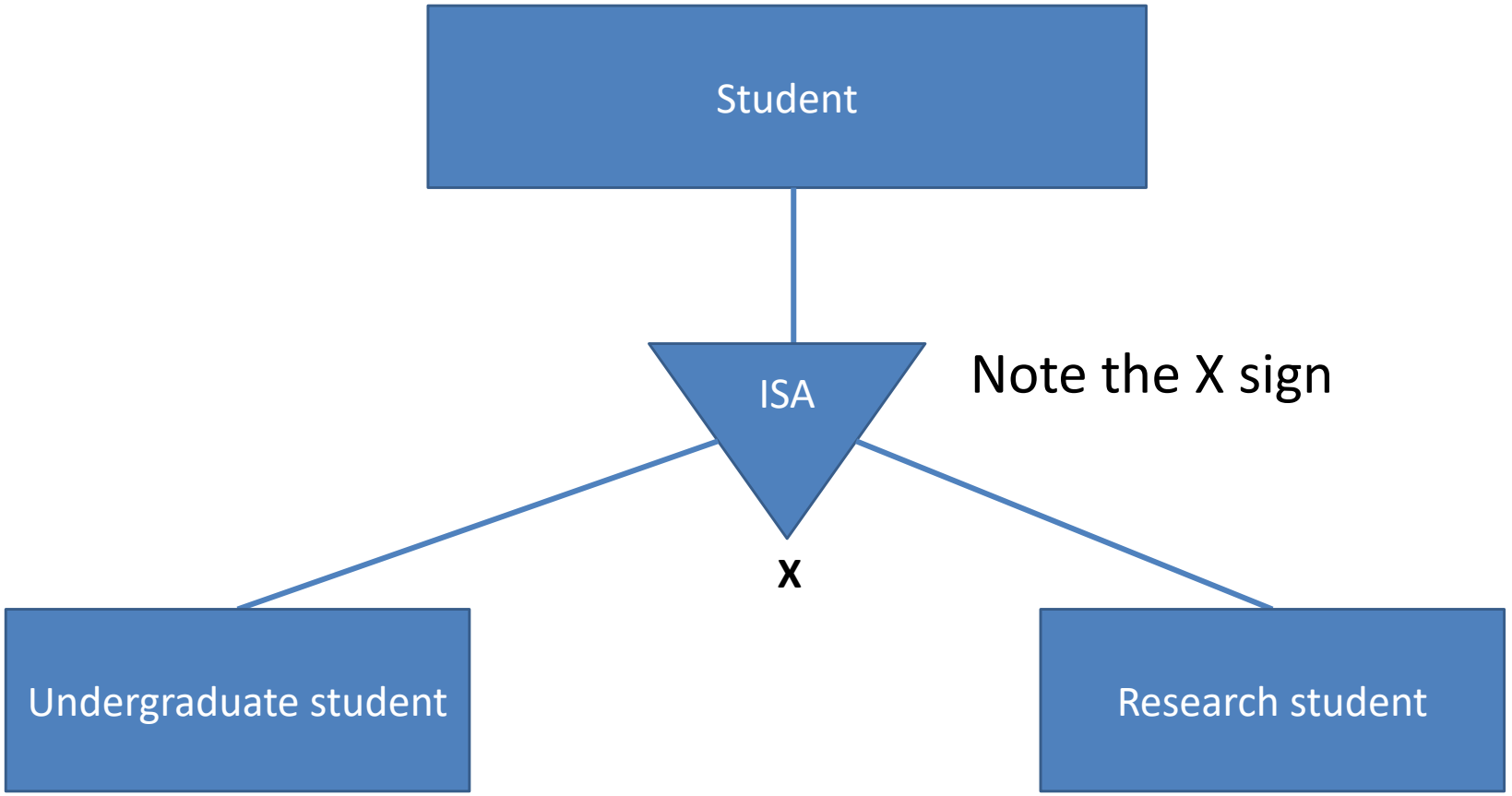
Overlapping in ERD



The human could be both university employee and student.

Reichman University

Exclusive in ERD



A student can be either an undergraduate or a research student but not both.

Reichman University

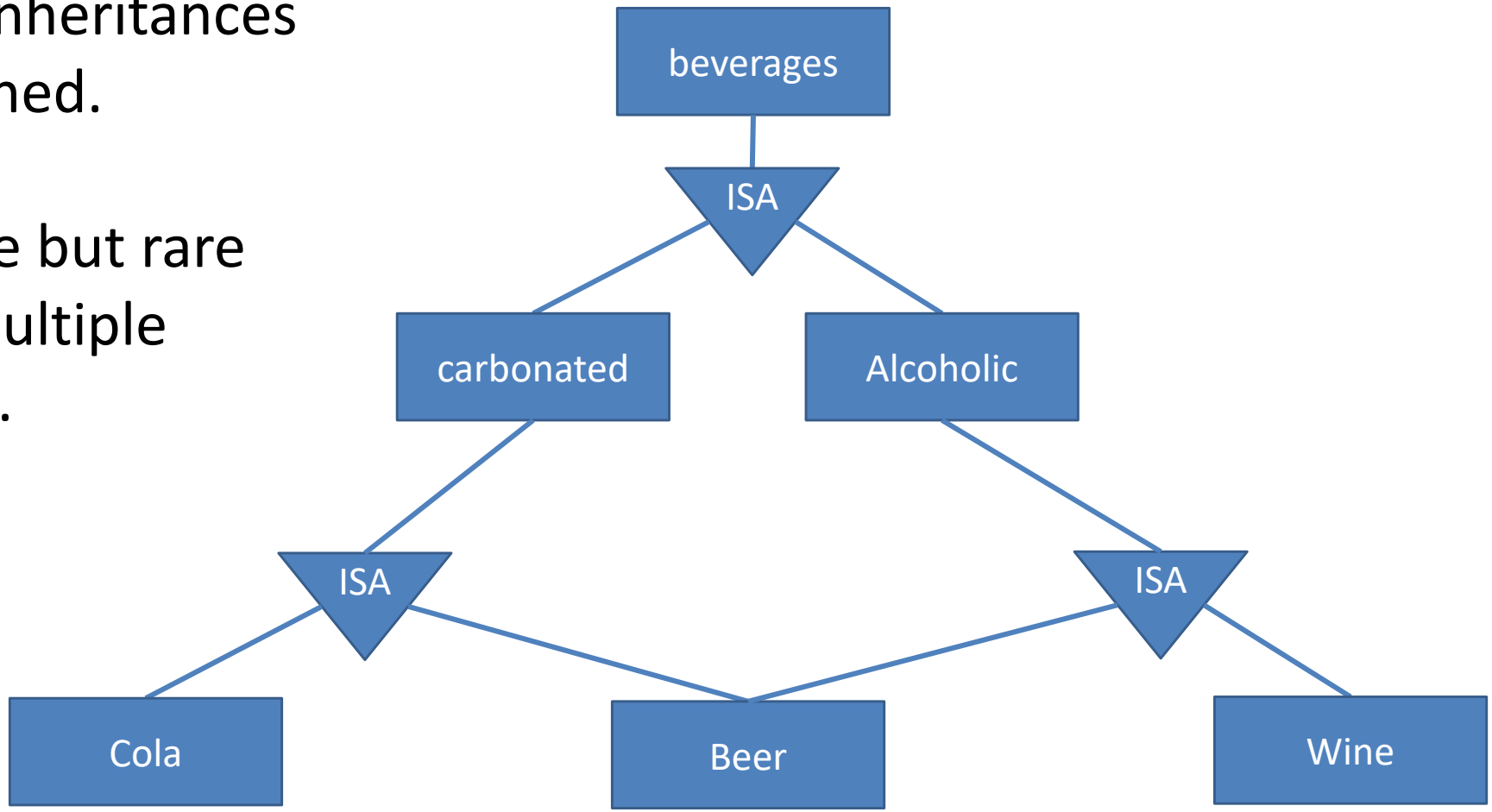
Combinations of constraints

- There could be various combinations of overlap and coverage, as follows:
  - Overlap allowed and no coverage (default)
  - Overlap is prohibited and there is no coverage.
  - Overlap is allowed and there is coverage.
  - Overlap is prohibited and there is coverage.

Reichman University

Chain of constraints

- A chain of inheritances can be defined.
- It is possible but rare to define multiple inheritance.



Reichman University