Electronics and
Computer Science

University of Southampton

Data Modelling

COMP6226: Software Modelling Tools and Techniques for Critical Systems

Dr A. Rezazadeh (Reza)
Email: ra3@ecs.soton.ac.uk or ar4k06@soton.ac.uk

November 24

University of Southampton

Overview

- What is data modelling and what are its benefits?
- Basic concepts and constructs of a data model
- Entity Relationship Diagram for data modelling

University of Southampton

Data Modelling

- Data modelling is a technique for organising and documenting system's DATA.
- Data modelling is sometimes called database modelling because a data model is usually implemented as a database. It is sometimes called information modelling.
- Many experts consider data modelling to be the most *important* of the modelling techniques.

data more have longer lifespan than system and can share between systems

Why is data modelling considered crucial?

- Data is viewed as a shared resource to be used by many applications.
- As a result, data must be organised in a way that is flexible and adaptable to unanticipated business requirements.
- Data structures and properties are reasonably permanent
  - Certainly, a great deal more stable than the applications that use the data.
  - Often the data model of a current system is nearly identical to that of the desired system (new system).
- The process of constructing data models helps analysts and users quickly reach consensus on business terminology and rules.

University of Southampton

Why is data modelling considered crucial?

- Poor design leads to the following software flaws:
  - Poor response time and hence poor performance
  - Redundancies and difficulty in system maintenance
  - Data omissions and integrity problems
  - Security and reliability problems
  - Pressure on the programming effort to compensate for the poor design
  - Lack of clarity and inflexibility

University of Southampton

A Small test for you

- Which of the following software relies on a database?
  - Operating systems
  - Compliers
  - Expert systems
  - CAD or case tools
  - Healthcare management systems
- All of them

University of Southampton

Some Objectives of Database Design

- Efficiency and flexibility
- Reliability, security and protection
- Control of redundancy
- Consistency and Accuracy
- Ease of access and ease of change
- Data independence — immunity of application programs to structural, storage or hardware changes of the database
- Clarity and multi-user access

Approaches to Data Modelling

- Underlying Database approaches
  - **Relational** model
  - **Object-oriented** model
  - **Hierarchical** model
  - **Network** model
  - **Non-relational** models
- The relational model and the OO model are the two which are adopted by contemporary software engineering.
  - In recent years non-relational database systems gained huge popularity.

Database Design – Methodology

Steps involved in designing a relational or OO database:

1. Identify data entities or object types
2. Identify relationships
3. Eliminate unnecessary relationships
4. Develop an entity-relationship diagram (ERD) or an object- relationship diagram (ORD)
5. Prepare the database tables' specification
6. Develop and implement the database

Relational model vs the OO model

- Object Models can be Mapped to relational database using Object-relational mapping approaches.
  - ORM is a technique for converting data between object-oriented world and relational database systems.
- Entity Framework is Microsoft's primary means of interacting between .NET applications and relational databases.
  - Entity Framework is an ORM which simplifies mapping between objects in your software to the tables and columns of a relational database.

Data Modeling - Graphical Representation

- Identification of 'what' elements from the problem domain that is relevant
- There are several notations for data modeling, but the actual model is frequently called an Entity Relationship Diagram (ERD).
- An ERD depicts data in terms of the flowing graphical elements:
  - Entities
  - Attributes
  - Relationships

University of Southampton

Entities

- All systems contain data and data describes ‘things’
- A concept to abstractly represent all instances of a group of similar ‘things’ is called an entity.
- An entity is a class of persons, places, objects, events, or concepts about which we need to capture and store data.
- An entity has a set of attributes to describe it
- An entity drawn as a labeled rectangle in ERD

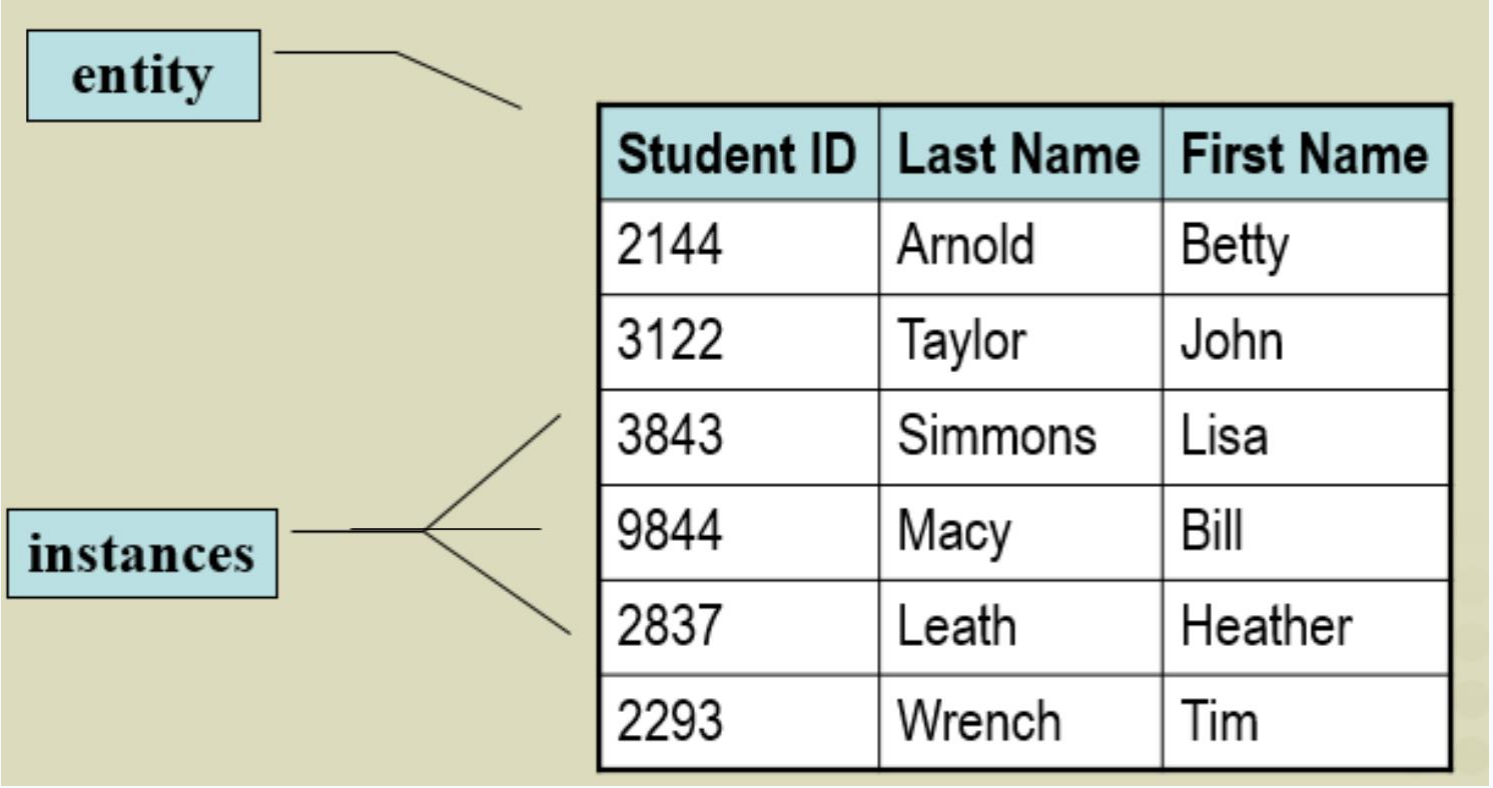
University of Southampton

Entity Examples

- External entity – (anything that produces or consumes information)
- Thing (e.g., report or display)
- Occurrence or events (e.g., phone call)
- Role (e.g., salesperson)
- Organizational unit (e.g., accounting department)
- Place (e.g., warehouse)
- Structure (e.g., File)

difference between entity and instances

Entity and instances



Student ID	Last Name	First Name
2144	Arnold	Betty
3122	Taylor	John
3843	Simmons	Lisa
9844	Macy	Bill
2837	Leath	Heather
2293	Wrench	Tim

Attributes

- The pieces of data that we want to store about each instance of a given entity are called attributes.
  - An attribute is a descriptive property or characteristic of an entity. Synonyms include element, property, and field.
- Some attributes can be logically grouped into super-attributes called compound attributes.
  - A compound attribute is one that actually consists of more primitive attributes.
  - Synonyms : concatenated attribute, composite attribute, and data structure.

University of Southampton

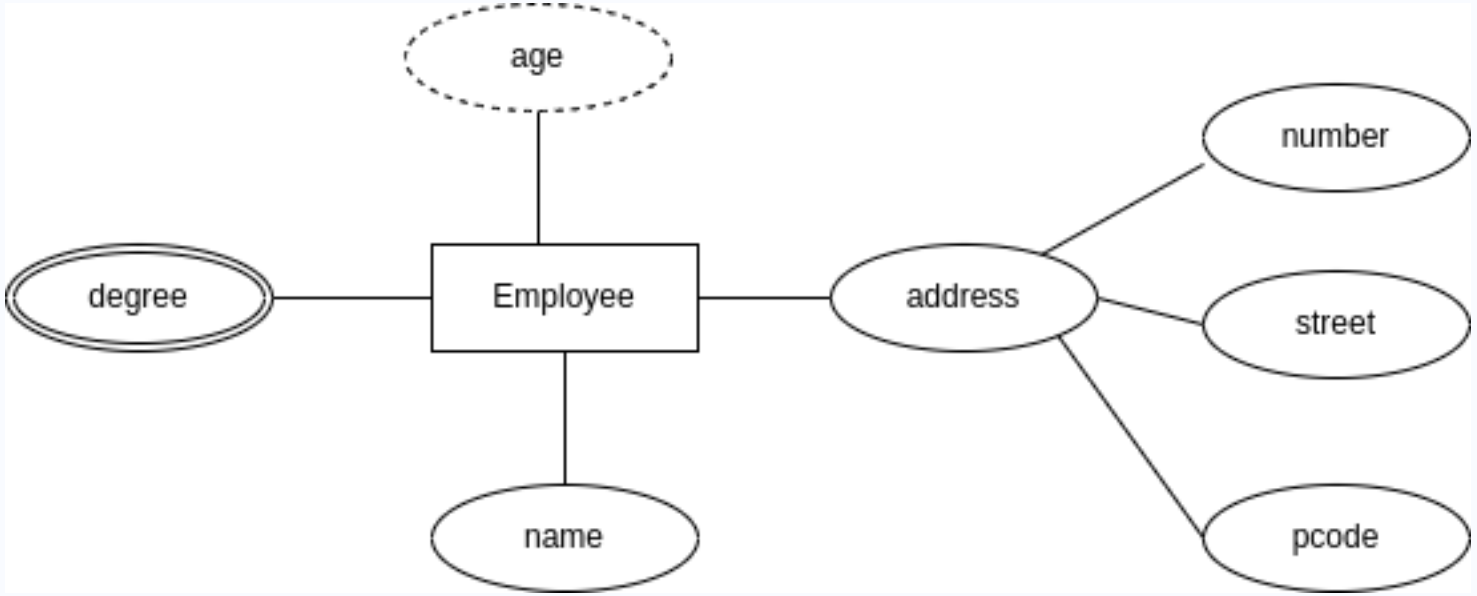
Types of attributes

- **Simple attribute**
- **Composite attribute**: consist of a hierarchy of attributes. E.g. : address {number, street, pcode}
- **Multivalued**: same instance can have different values
- **Derived**: an attribute whose value is computed from other attributes

store date of birth not age

Different types of attributes

- Address: composite,
- Degree: multivalued (Bsc, MSc, PhD)
- Age: derived (DoB)



University of Southampton

Attributes

Examples of Simple and Composite (Compound) attributes for a Student entity

STUDENT

Name
- . Last Name
- . First Name
- . Middle Initial

Address
- . Street Address
- . City
- . State or Province
- . Country
- . Postal Code

Phone Number
- . Area Code
- . Country code

Date of Birth
Gender
Ethnicity
Programme
Grade Point Average

University of Southampton

More on Attributes – Keys

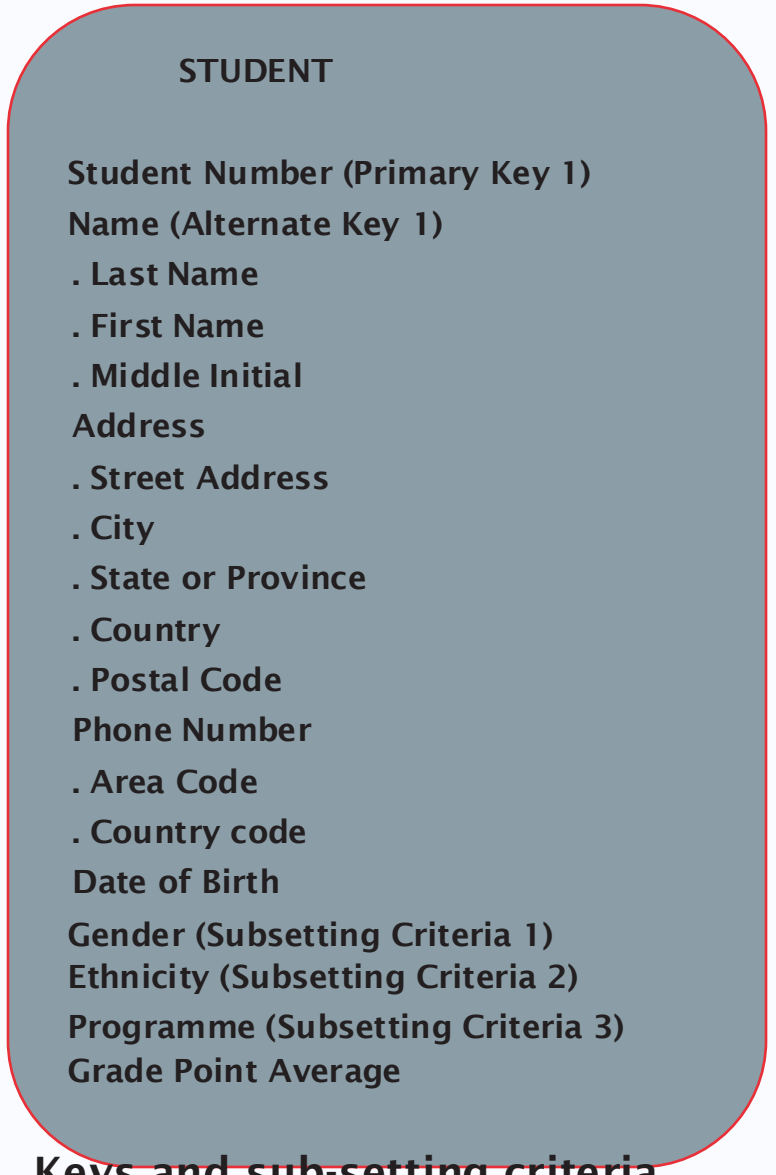
- One or more attributes must be defined as identifier - "**key**" to find an instance of the entity. (e.g. ID number of a student).
- Sometimes more than one attribute is required to uniquely identify an instance of an entity.
  - A group of attributes that uniquely identifies an instance of an entity is called a concatenated key.
  - Synonyms include composite key and compound key.

More on Keys

- Frequently, an entity may have more than one key.
- Each of these attributes is called a candidate key.
- A candidate key is a ‘candidate to become the primary identifier’ of instances of an entity. (Note: A candidate key may be a single attribute or a concatenated key.)
- A primary key is that candidate key which will most commonly be used to uniquely identify a single entity instance.
- Any candidate key that is not selected to become the primary key is called an alternate key.

Attributes - Sub-setting Criteria

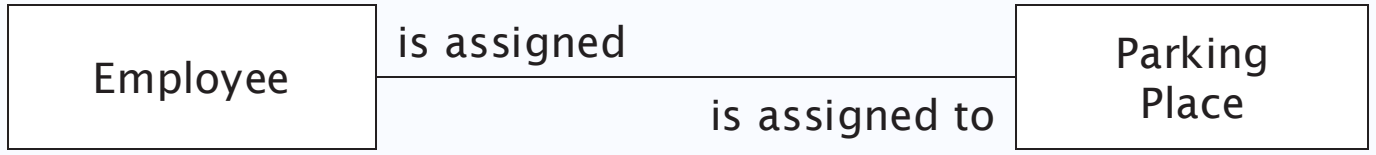
- Sometimes, it is also necessary to identify a subset of entity instances as opposed to a single instance.
  - For example, we may require a simple way to identify all male students, and all female students.
  - A sub-setting criteria is an attribute (or concatenated attribute) whose finite values divide all entity instances into useful subsets.
  - Some methods call this an inversion entry.



Keys and sub-setting criteria

Relationships

- Associations between instances of one or more entity types that is of interest
- The relationship may represent an event that links the entities, or merely a logical affinity that exists between the entities.
- Drawn as line between entities, labeled with verb phrases
- A verb phrase describes the relationship.
  - All relationships are implicitly bidirectional, meaning that they can interpreted in both directions.



Relationship - Cardinality

- Specification of the number of occurrences of one object that can be related to the number of occurrences of another object
- Possible relationships:
  - One-to-One: Each entity in the relationship will have exactly one related entity
  - One-to-Many: An entity on one side of the relationship can have many related entities, but an entity on the other side will have a maximum of one related entity
  - Many-to-Many: Entities on both sides of the relationship can have many related entities on the other side

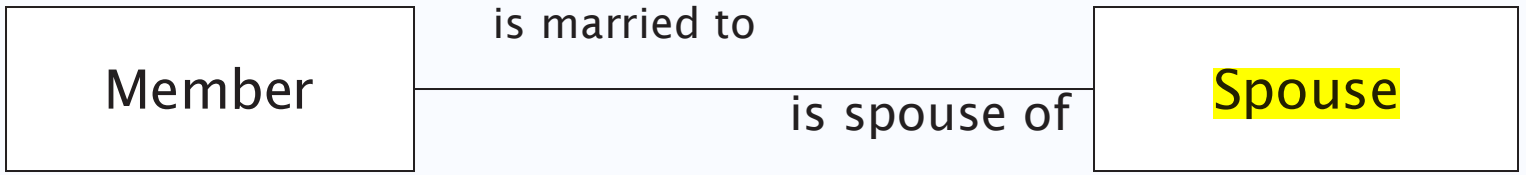
University of Southampton

Relationship – Differences with OO

- Aggregation relationship (if OO database)
- Component/Composition relationship (if OO database)
- Super-type-sub-type relationship (if OO database)

University of Southampton

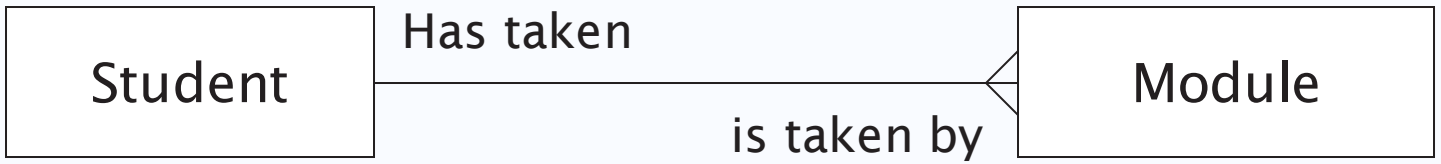
One-to-One Relationship



"Every member has a spouse"

University of Southampton

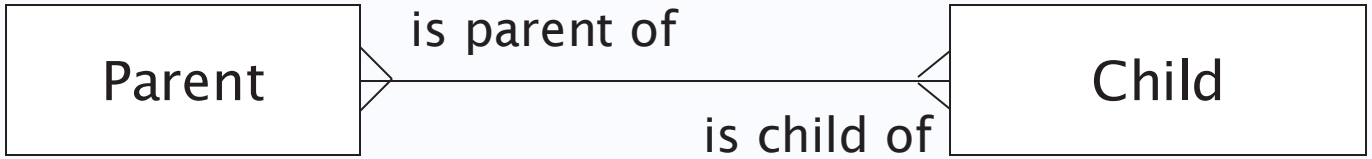
One-to-Many Relationship



"Every student has one or more Modules"

University of Southampton

Many-to-Many Relationship



"Every parent has one or more children,
and every child has one or more parent"

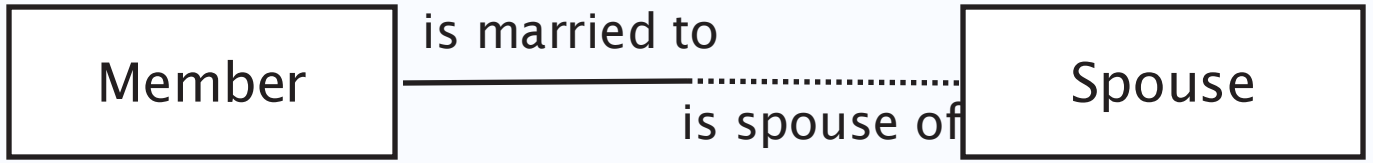
University of Southampton

Relationship - Modality

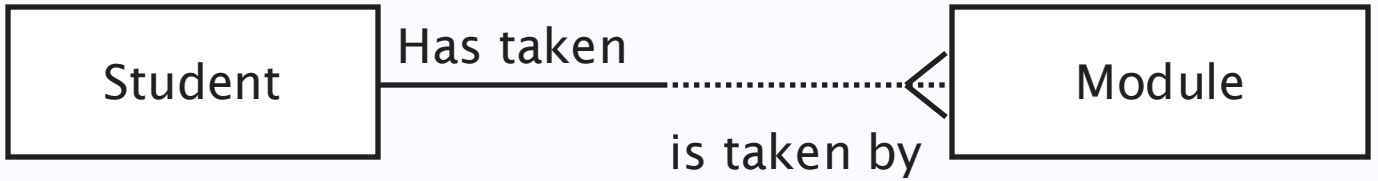
- Specifies whether the relationship is optional or mandatory
- Modality is **0** if relationship is **optional**
  - represented by dotted line in ERD
- Modality is **1** if relationship is **mandatory**
  - represented by straight line in ERD

University of Southampton

Optional relationships



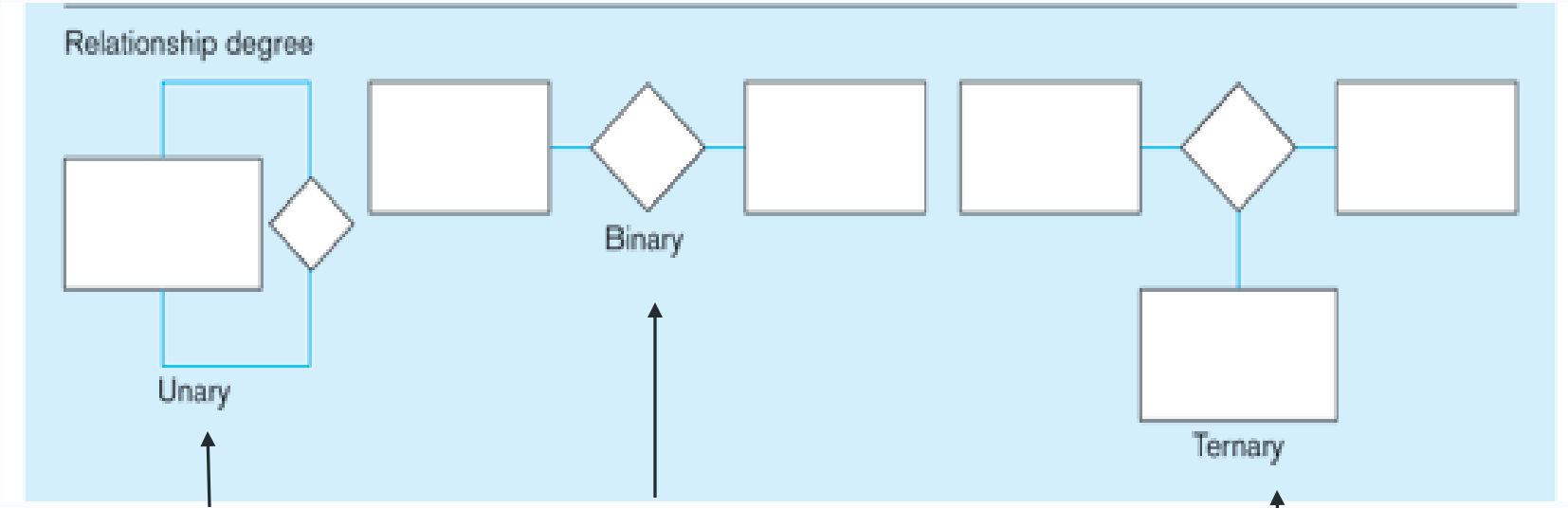
"Every member may have a spouse"



""Every student has one or more Modules"

Degree of a Relationship

- Degree of a relationship is the number of entity types that participate in it



One entity related to another of the same entity type

Entities of two different types related to each other

Entities of three different types related to each other

University of Southampton

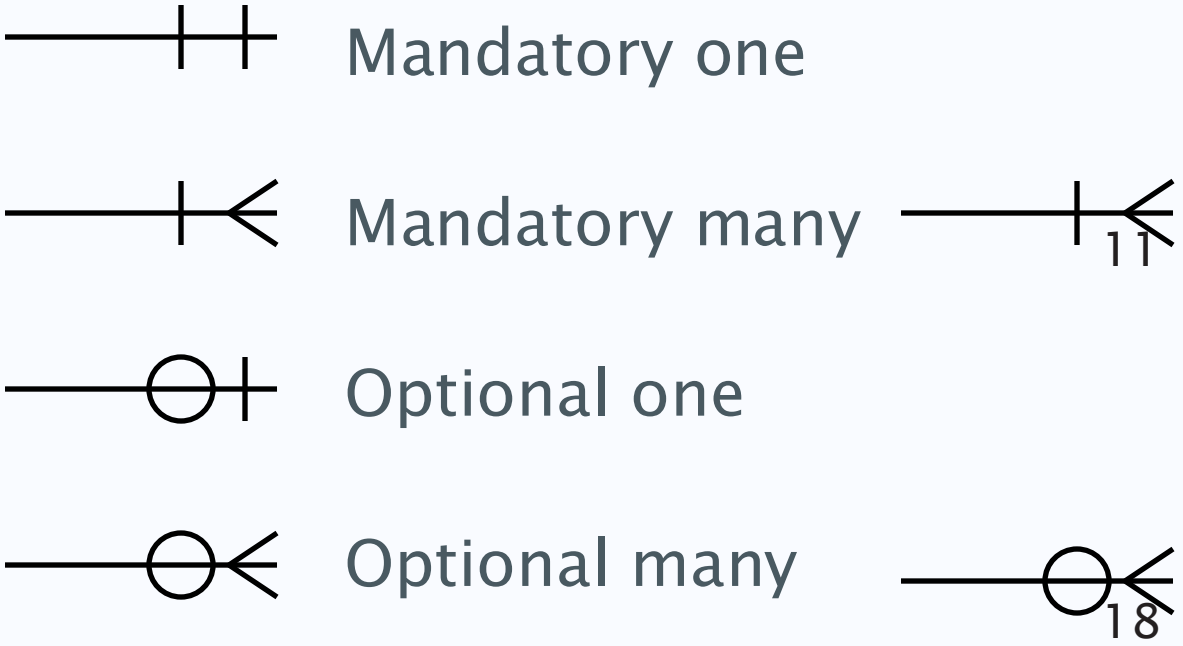
Cardinality Constraints

- Cardinality Constraints - the number of instances of one entity that can or must be associated with each instance of another entity
- Minimum Cardinality.
  - If zero, then optional
- If one or more, then mandatory
- Maximum Cardinality.
  - The maximum number

University of Southampton

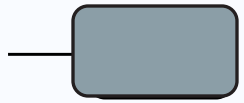
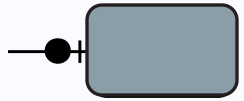
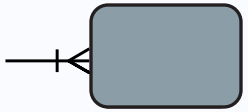
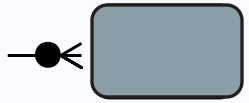
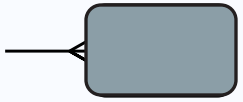
Cardinalities – Different Representation

use this table

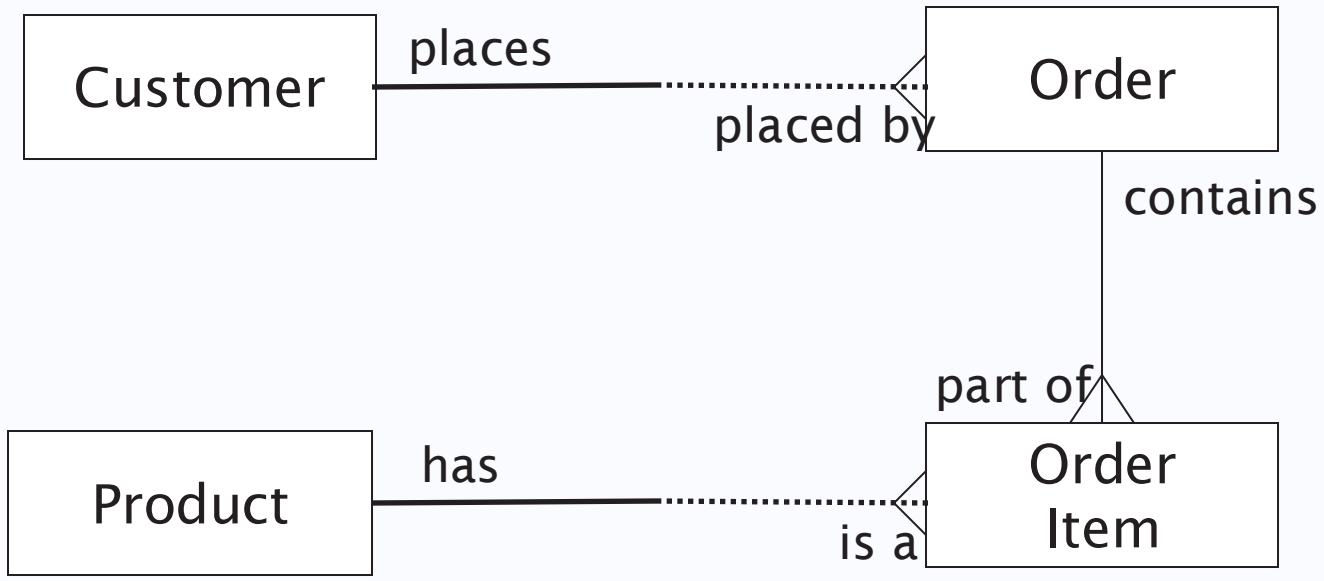


University of Southampton

Relationships – Different Representation

Cardinality Interpretation	Minimum Instances	Maximum Instances	Graphic Notation
Exactly one	1	1	
Zero or one	0	1	
One or more	1	many (> 1)	
Zero, one, or more	0	many (> 1)	
More than one	> 1	> 1	

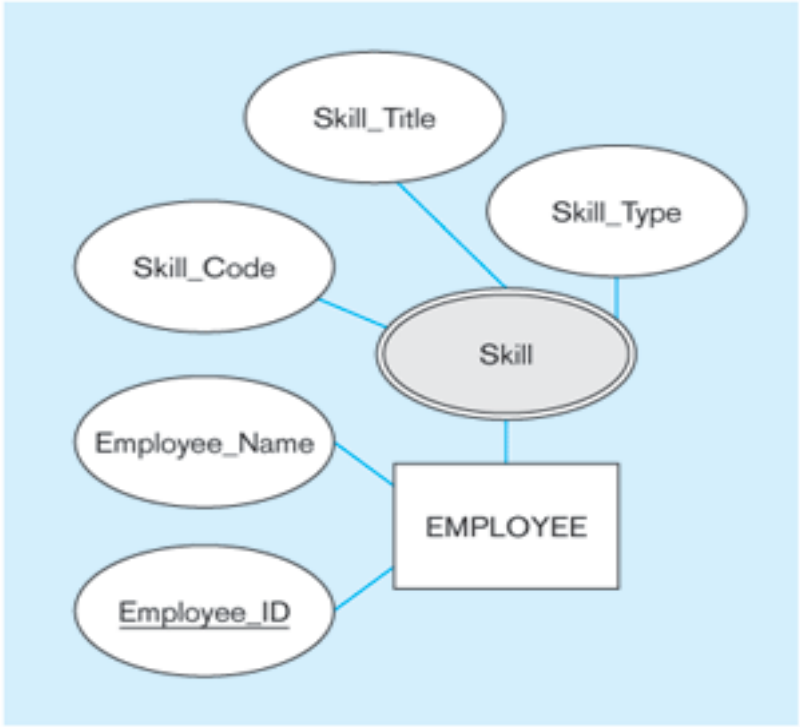
Example of Entity-Relationship Diagram



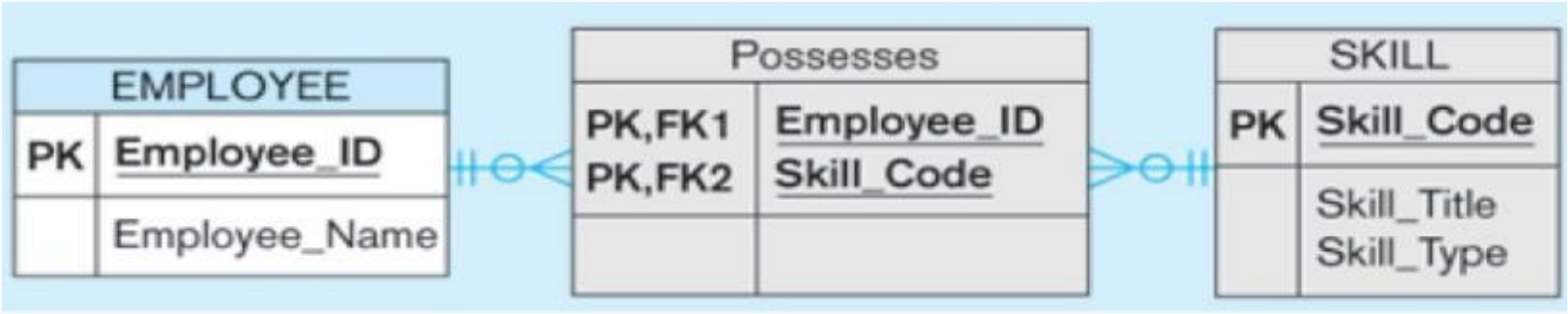
University of Southampton

Attributes or Relationship?

- Multivalued attributes can be transformed to relationships



more flexibal than the first one



affect the performance

Table Design

any problem

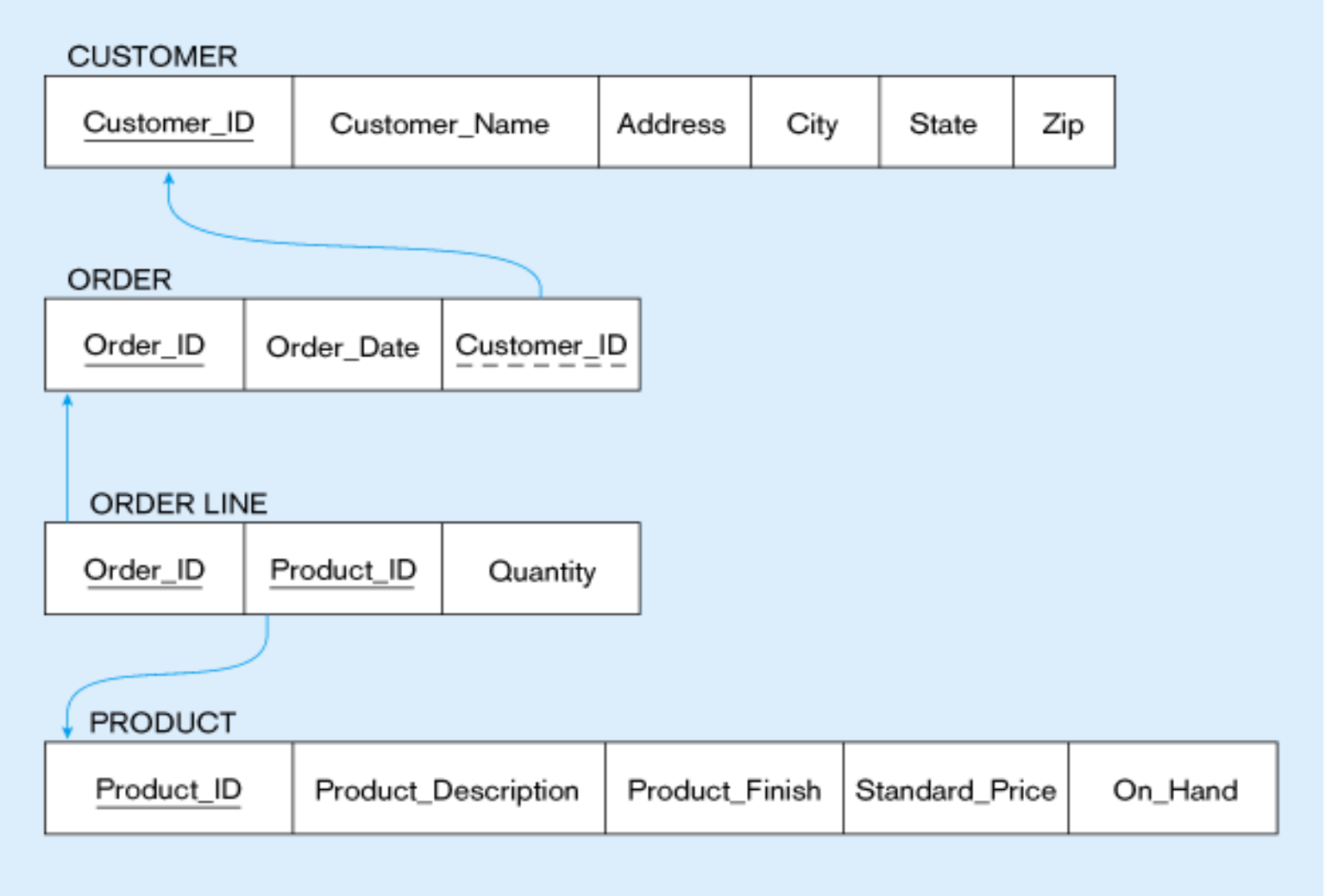
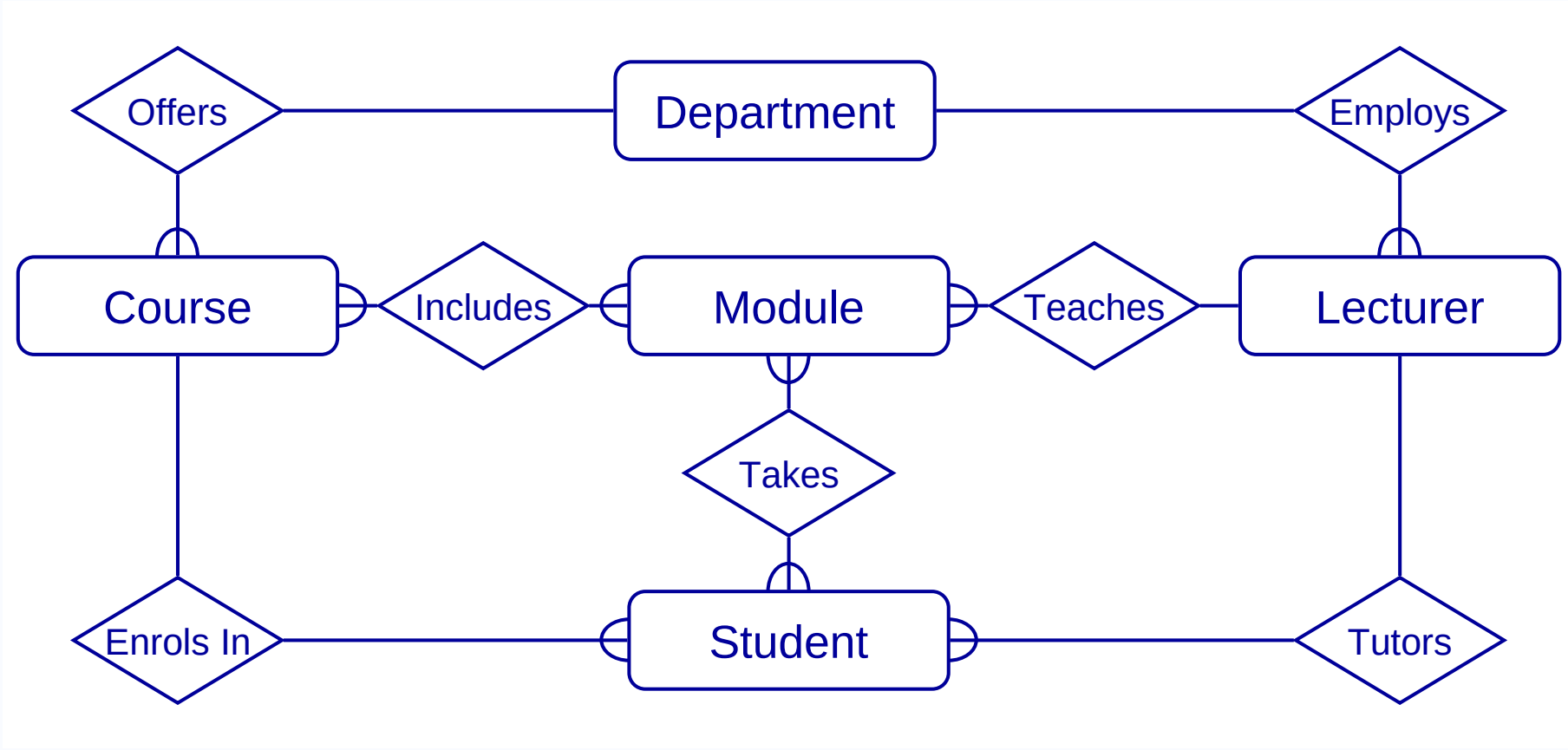


Table Design

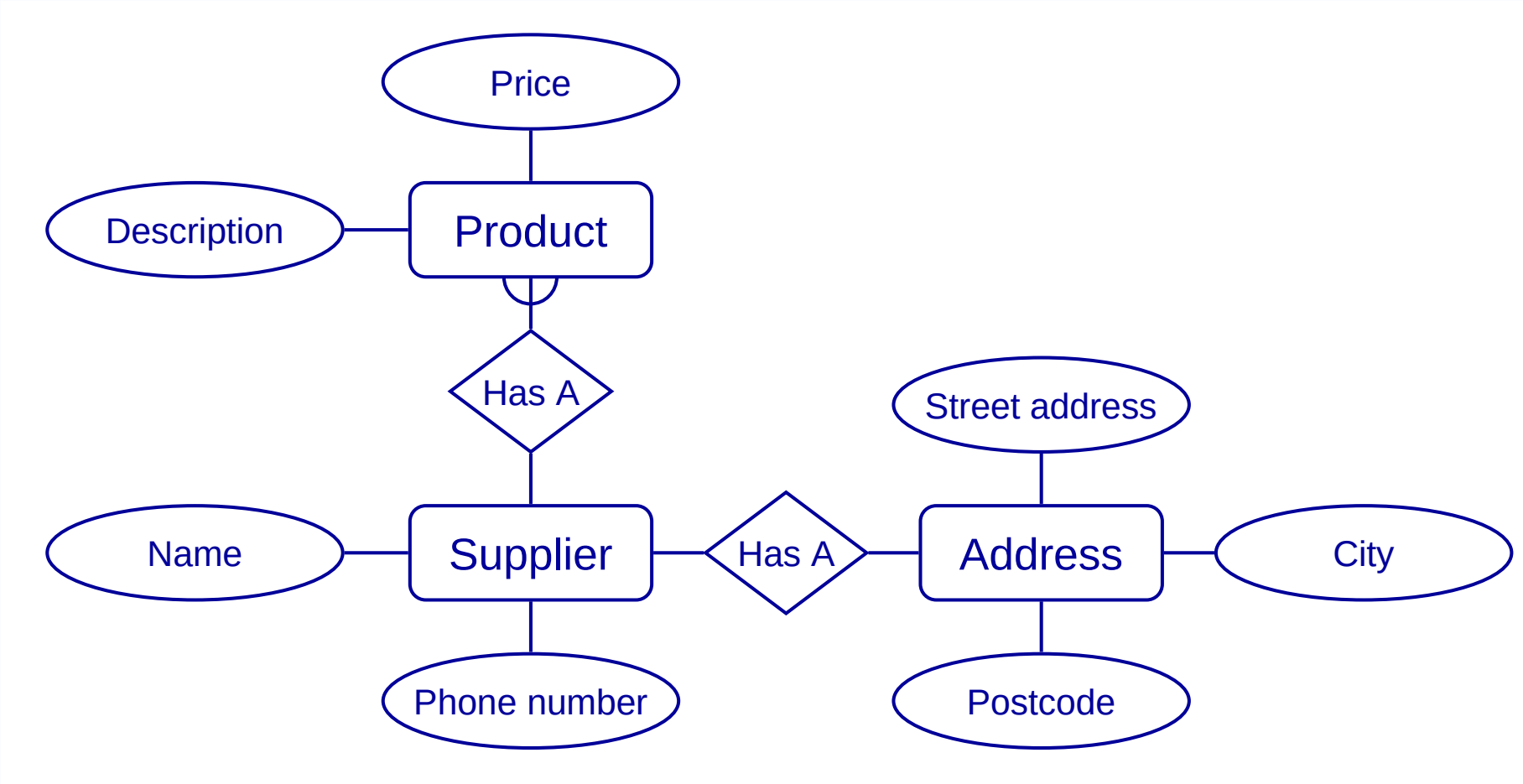
Logical Data Type	Logical Business Meaning
NUMBER	Any number, real or integer
TEXT	A string of characters, inclusive of numbers. When numbers are included in a TEXT attribute, it means we do not expect perform arithmetic or comparisons with those numbers.
MEMO	Same as TEXT but of an indeterminate size. Some business systems require the ability to attach potentially lengthy note to a give database record.
DATE	Any date in any format.
TIME	Any time in any format.
YES/NO	An attribute that can only assume one of these two values
VALUE SET	A finite set of values. In most cases, a coding scheme would be established (e.g., FR=freshman, SO=sophomore, JR=junior, SR=senior, etc.)
IMAGE	Any picture or image.

EDR Example – Student-Course (old style representation)

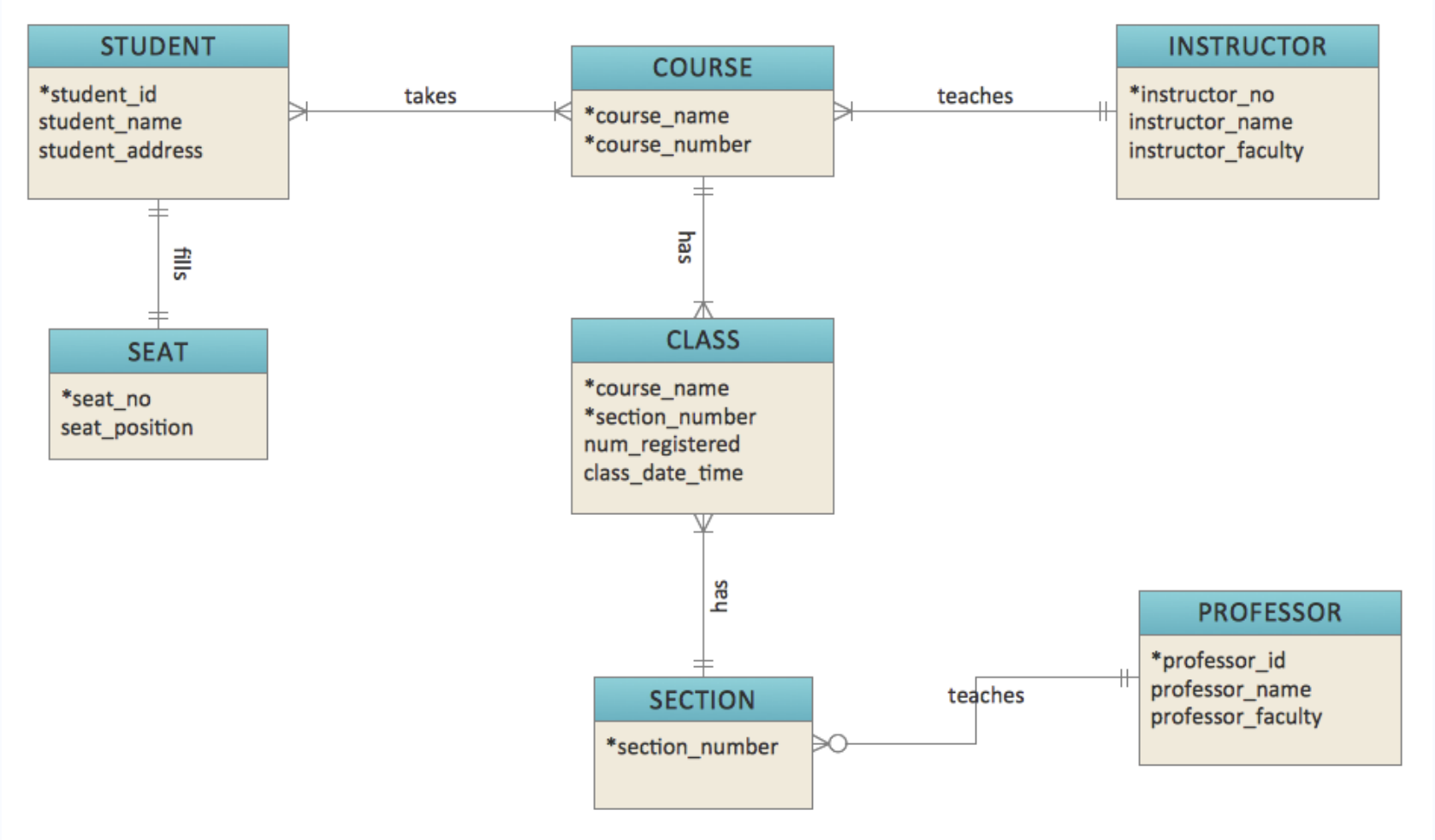
如何区别一对多和多对多的关系



EDR Example – Suppliers-Product (old style representation)

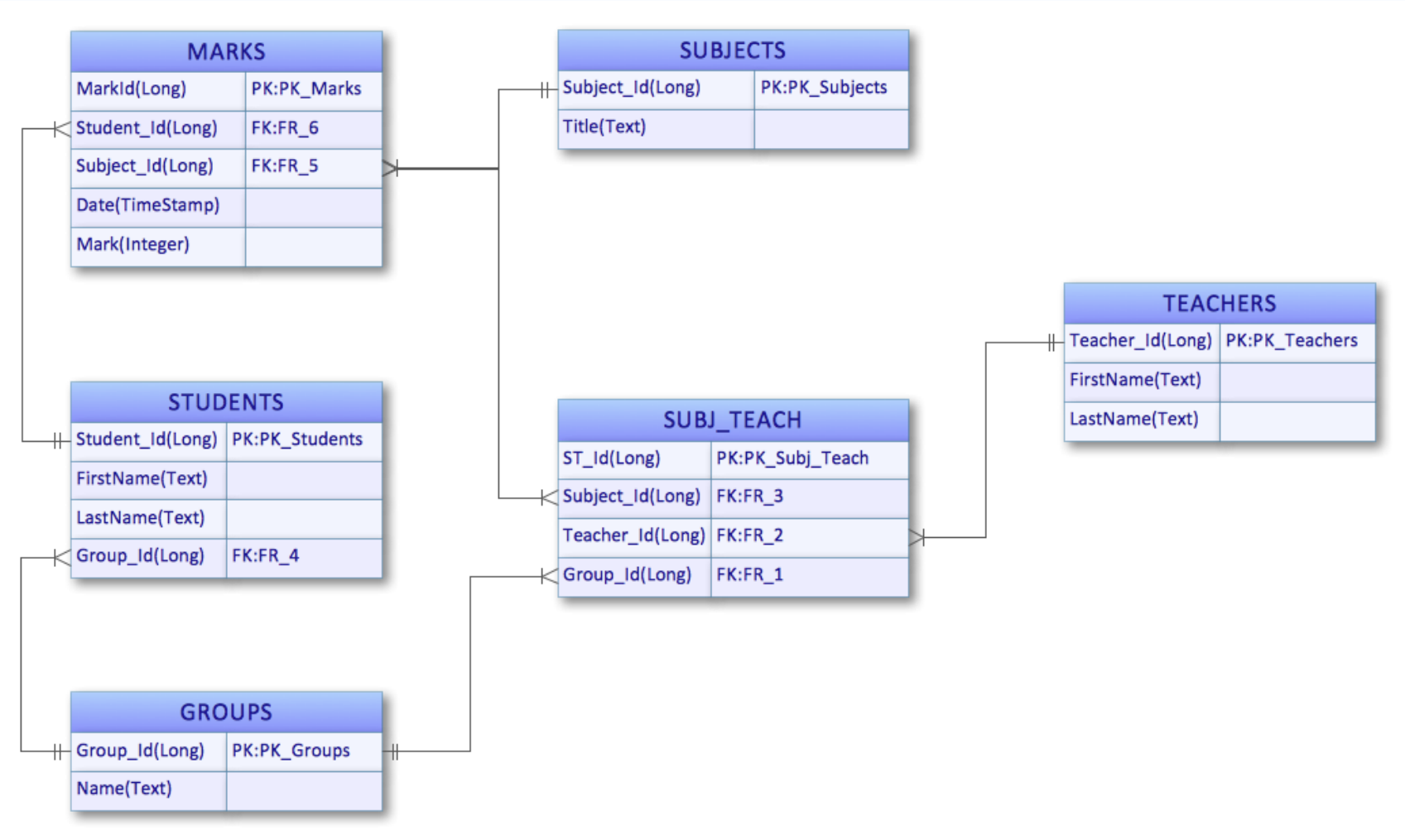


EDR Example – Student-Course (preferred representation)



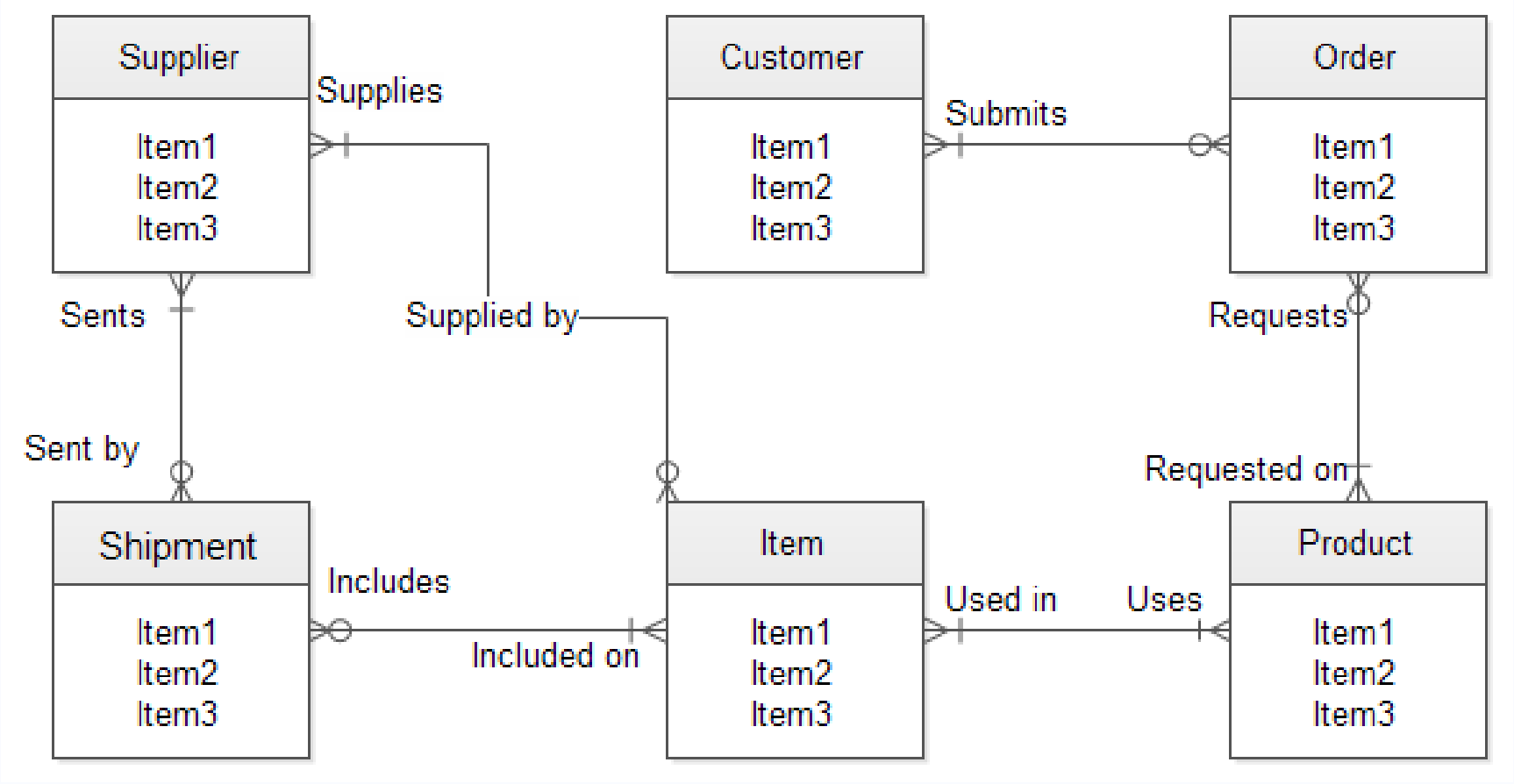
University of Southampton

ERD Example – Student-Course (preferred representation)



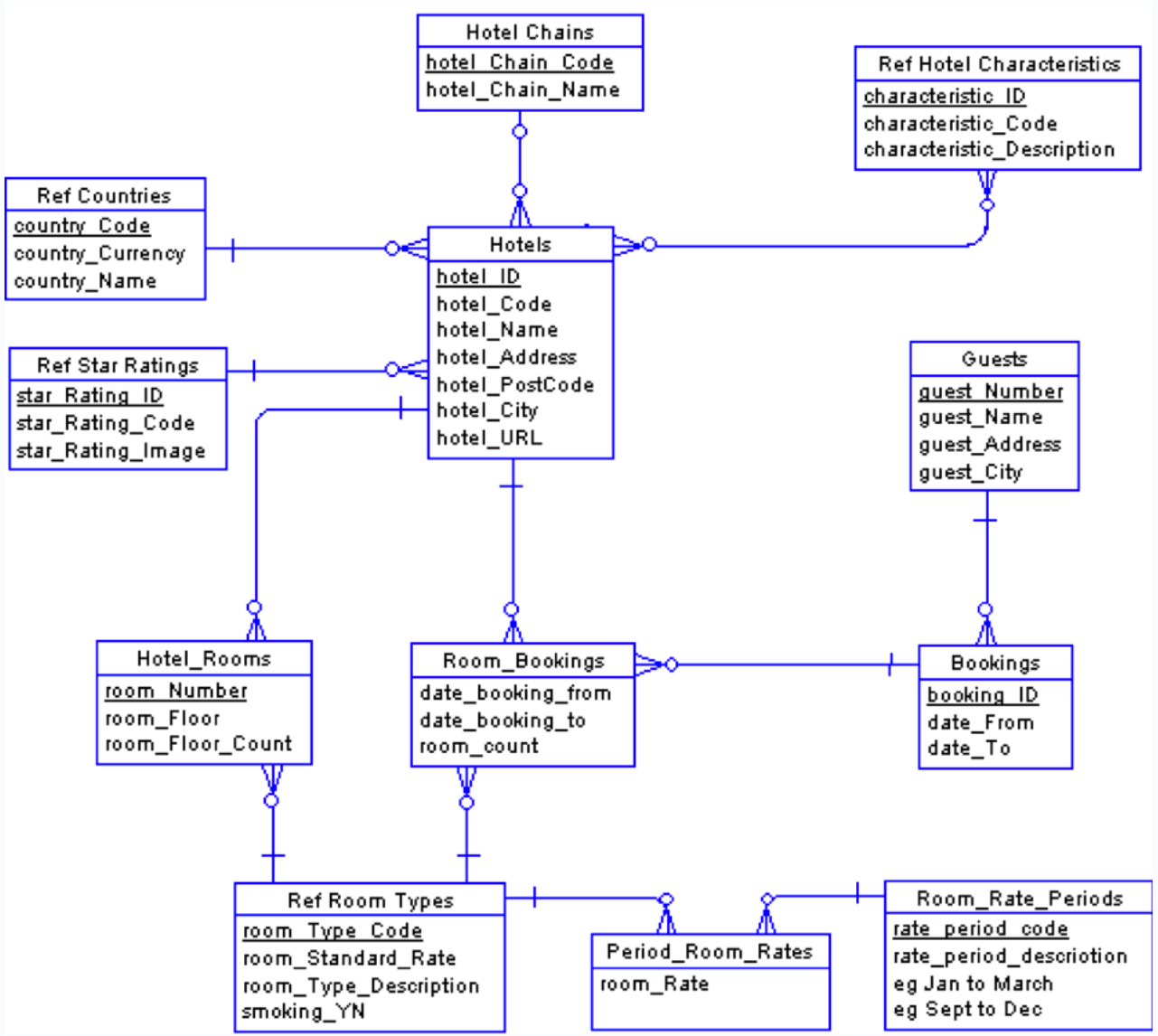
University of Southampton

EDR Example – Suppliers-Customer



University of Southampton

Conceptual Data Modelling for Hotel Reservation System



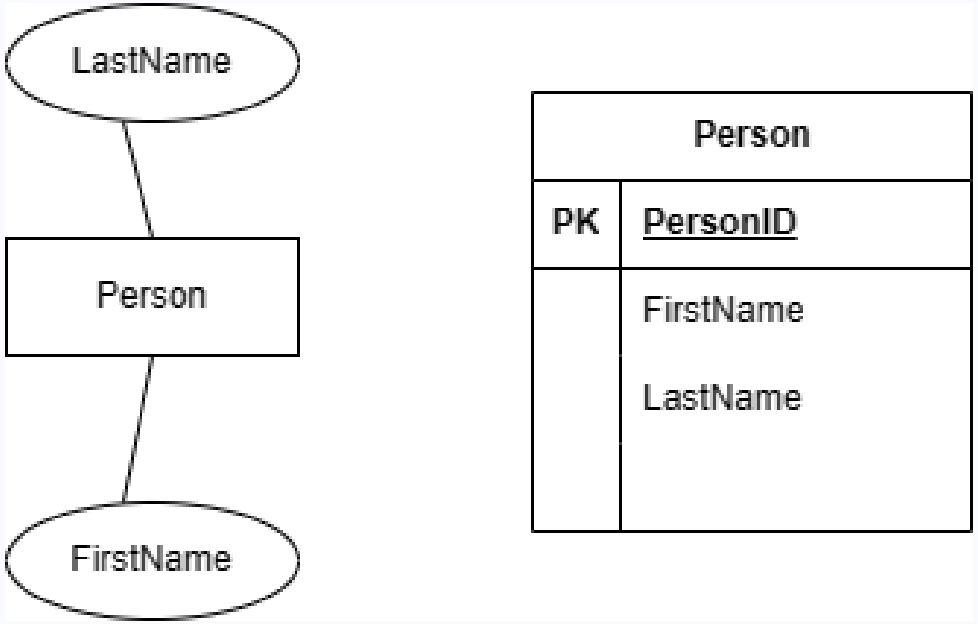
detail design should be different from the modeling design

UML versus ER

- ER
  - ER was developed earlier for relational databases
  - ER uses keys: when one or more attributes of an entity are used to uniquely identify it
  - Uses relational tables
  - Not truly platform independent
  - No representation of behaviour
- UML
- UML is object-oriented
- Do not support keys
- Not all relationships are given names or roles
- Independent of platform

From ER diagrams to relational database

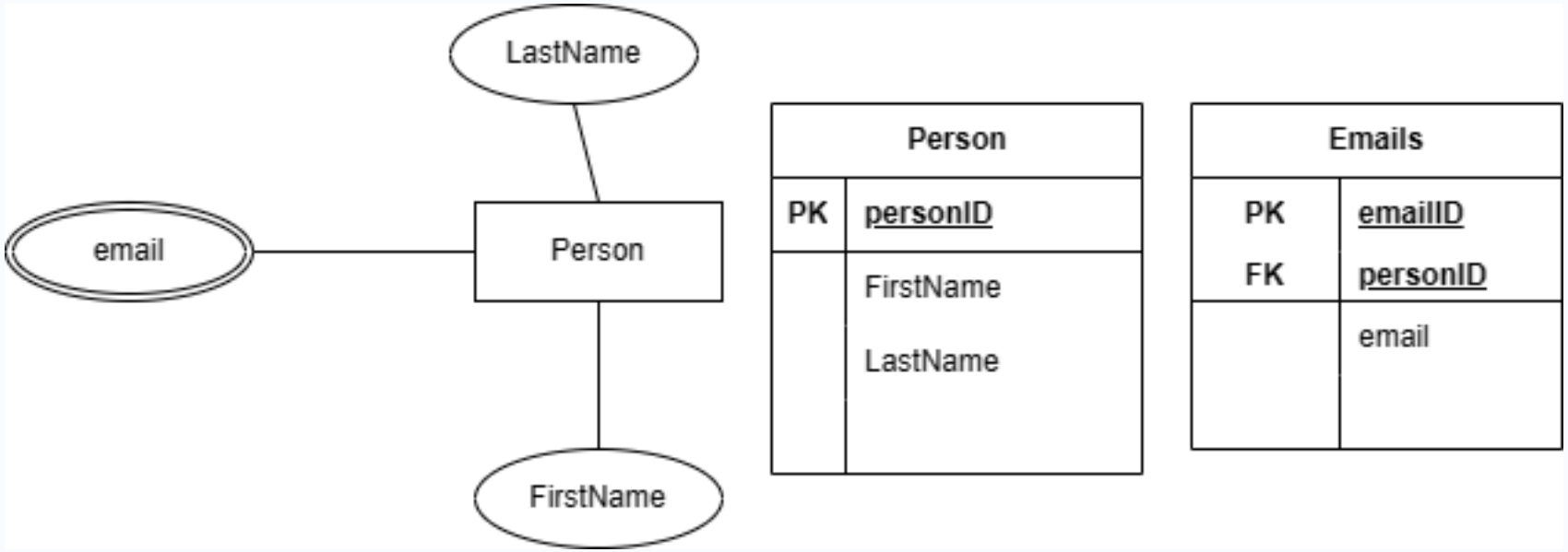
- Entities with simple attributes
- Entity is turned into a table
- Each attribute is turned into a column



From http://www.learndb.com

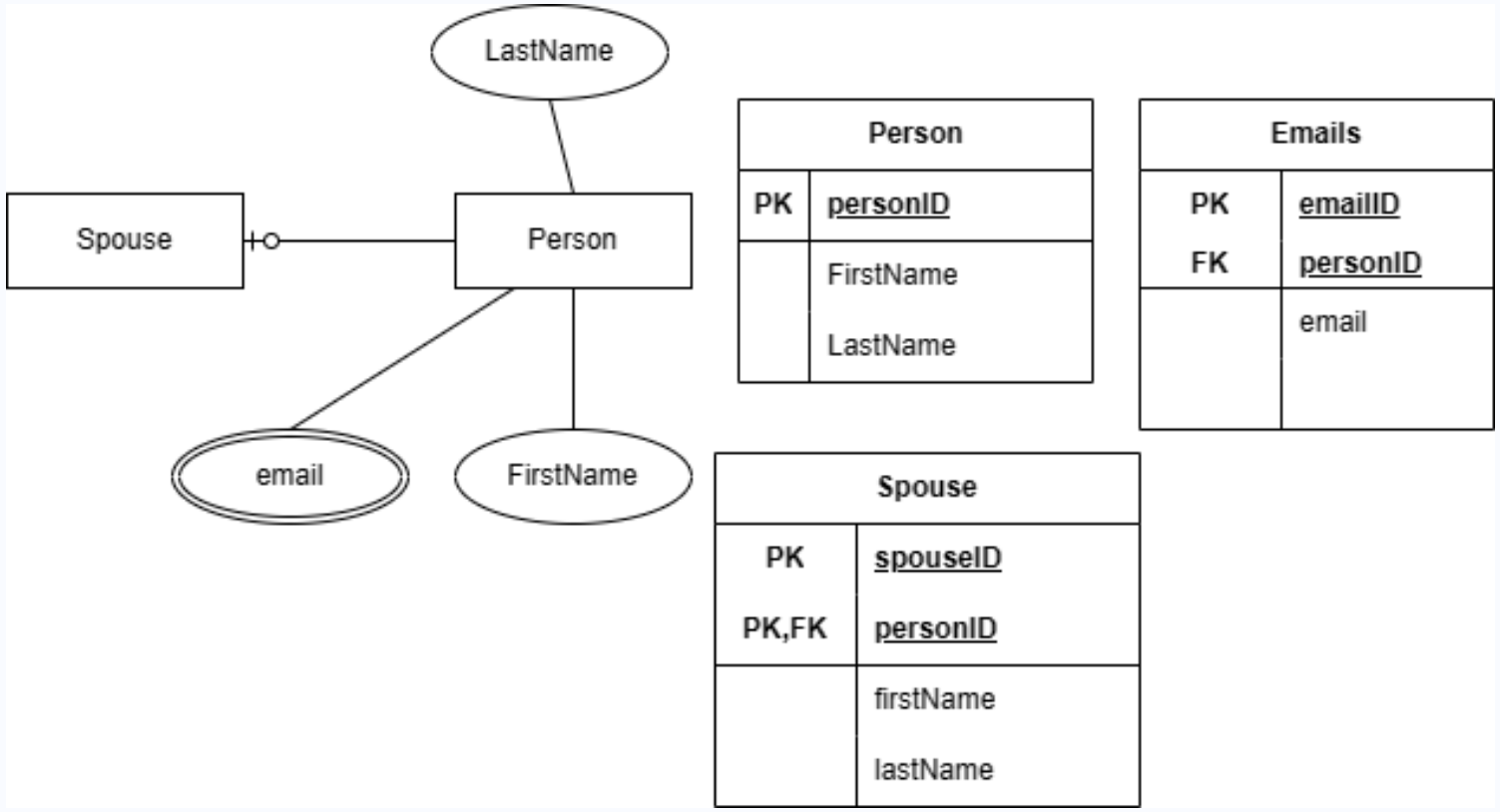
Multi-valued attributes

- Create a new table for the attribute and link it to the entity via a foreign key



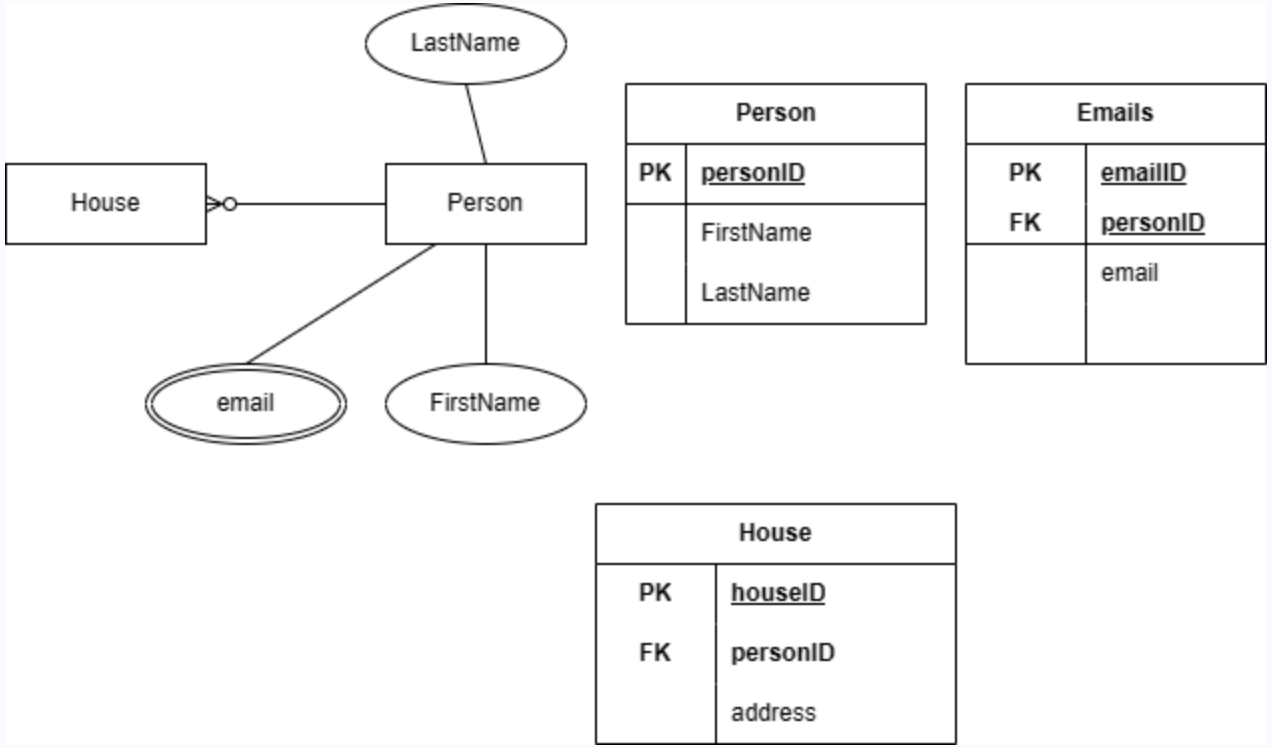
From http://www.learndb.com

Converting entities with 1:0-1 relationship



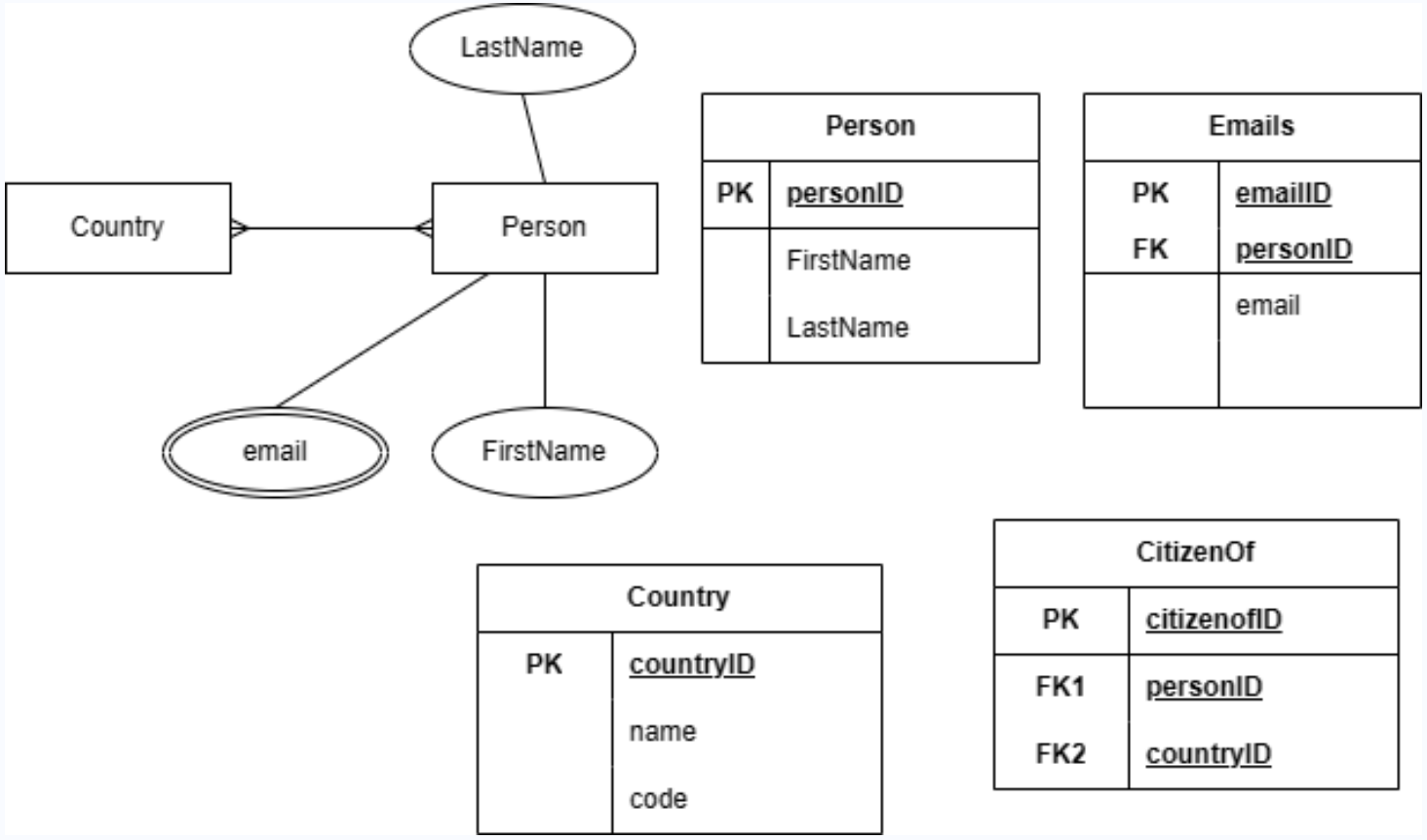
From http://www.learndb.com

Converting entities with 1:N relationships



From http://www.learndb.com

Converting entities with N:N relationships



From http://www.learndb.com

Final Points

- Data modeling should remain a value-added skill
- The demand for data modeling as a skill is dependent on two factors:
  - the need for databases, and
  - the use of relational database management system technology to implement those databases.
- Object Models can be Mapped to relational database using Object-relational mapping approaches
- ORM is a technique for converting data between object-oriented world and relational database systems.

University of Southampton

YOUR QUESTIONS