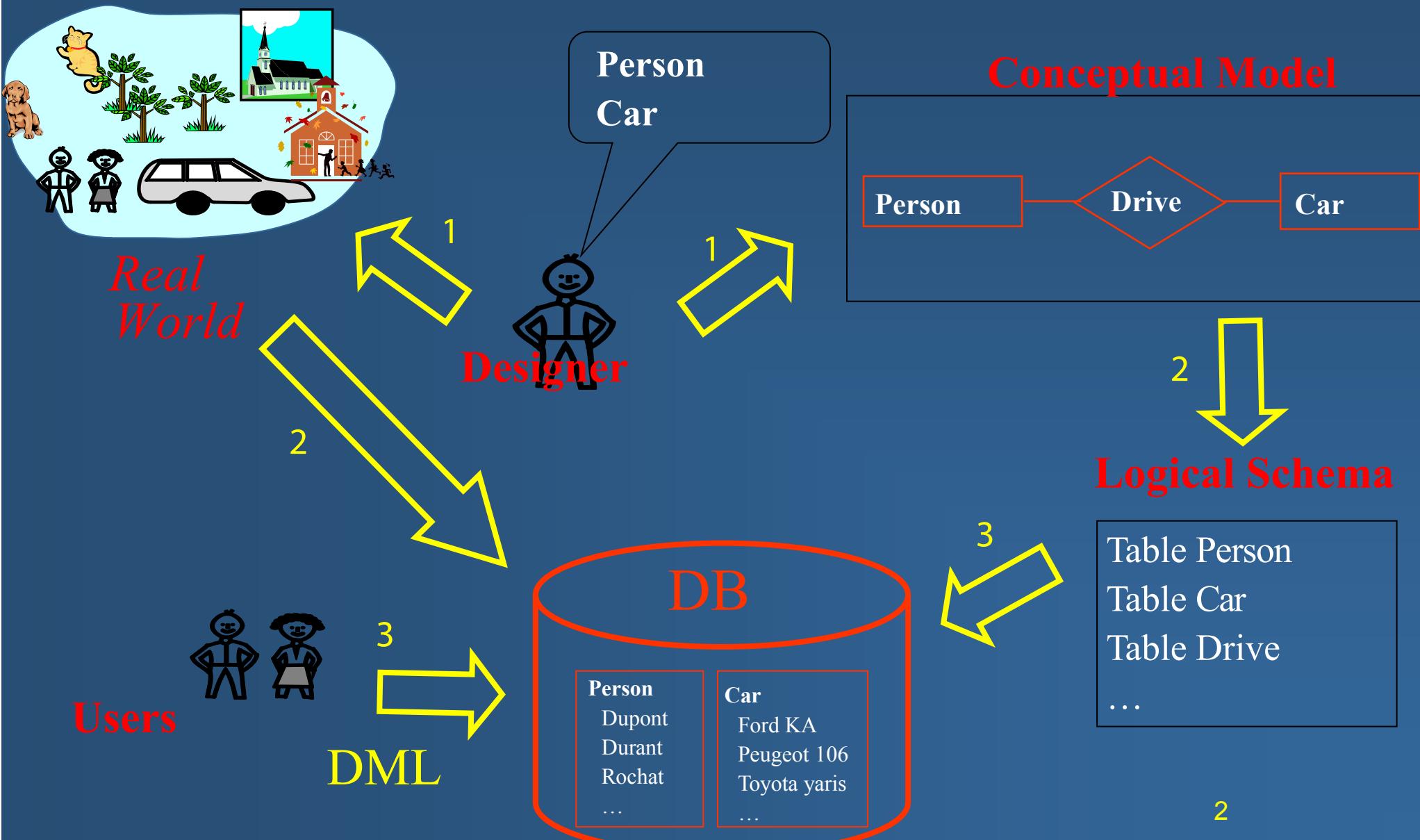


Relational Databases

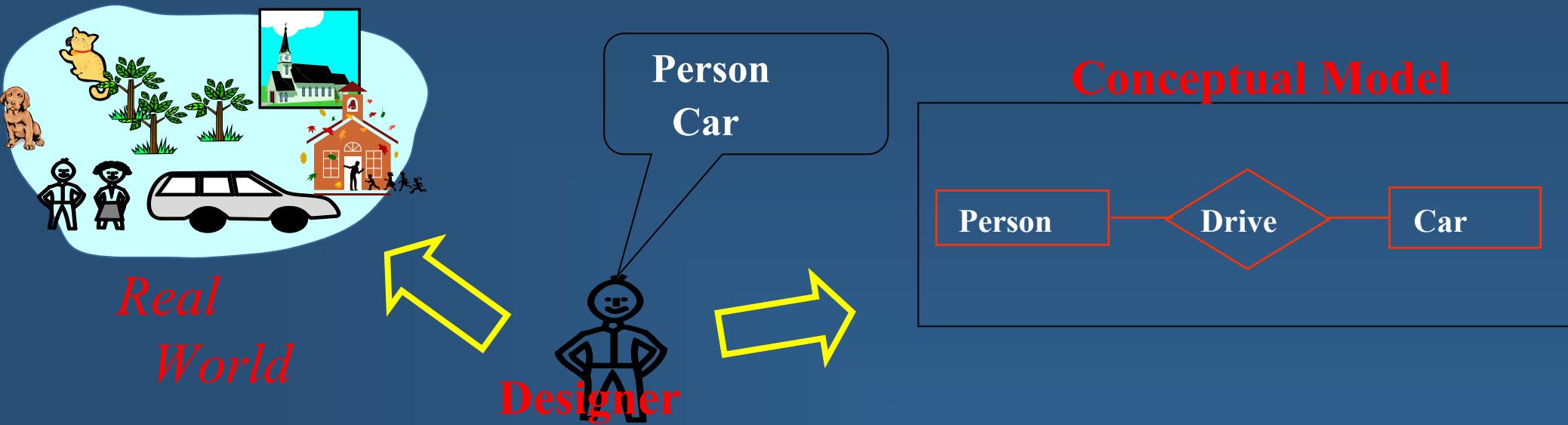
Conceptual Modeling

Part 1: General Concepts

Database Life Cycle (DBLC)

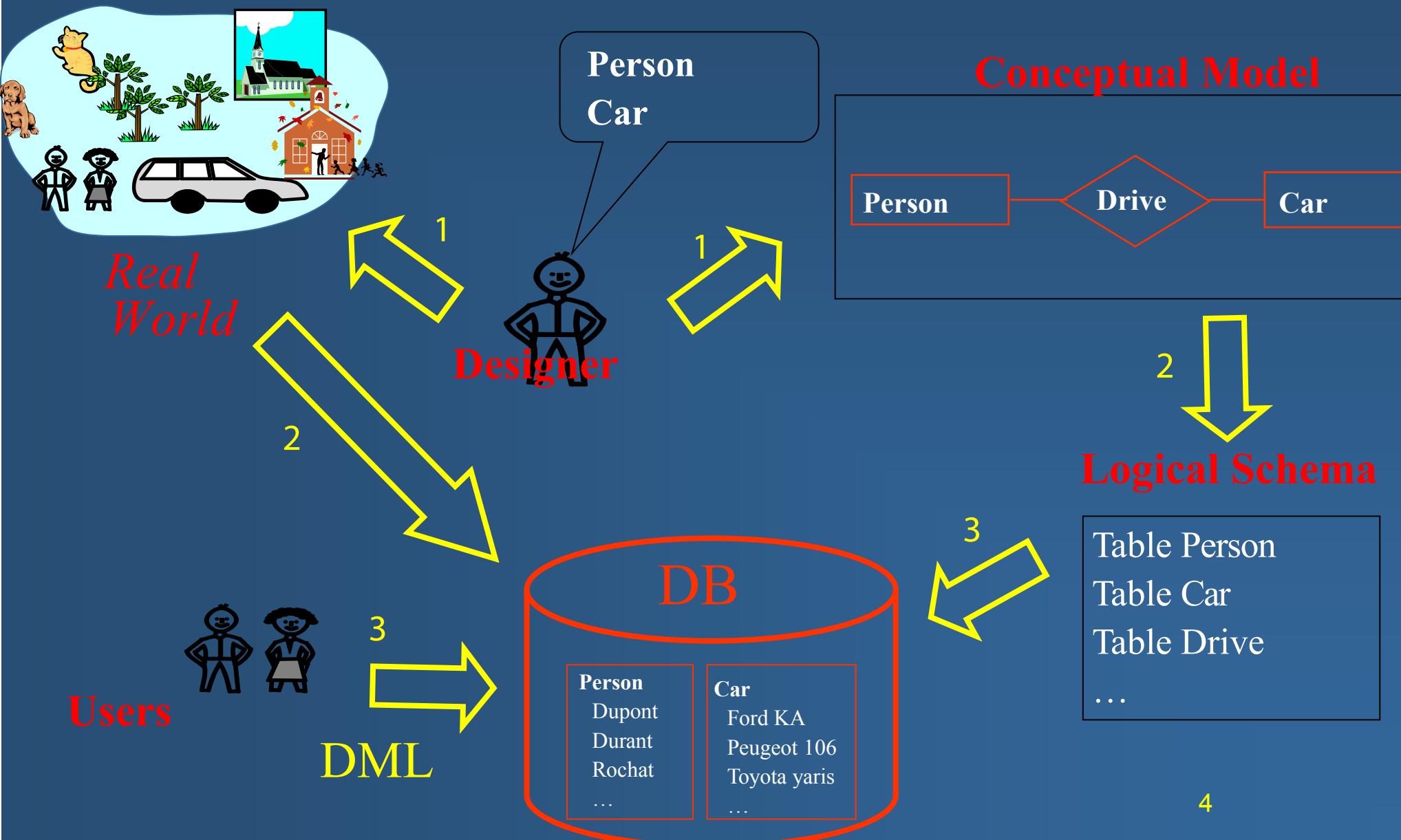


Conceptual Modeling



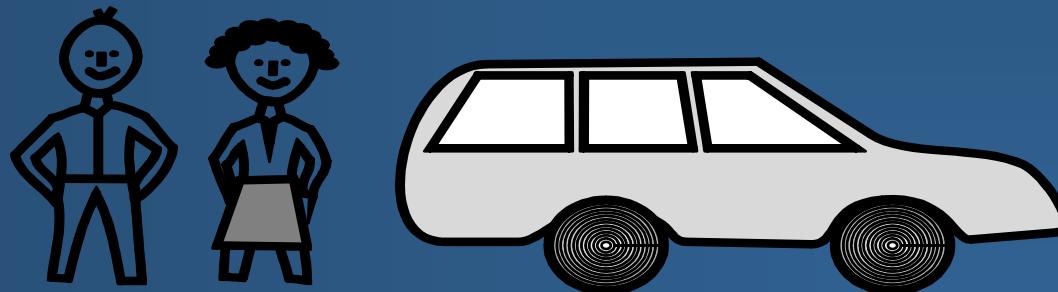
- Conceptual Modeling aims to define a **Conceptual Model** for the database

Why do we need Conceptual Modeling?



Why do we need Conceptual Modeling?

- Jean owns a Grey Honda CRV
- Arlette, his wife, is with him.



Why do we need Conceptual Modeling?

- The objective of conceptual modeling is to create a representation of reality as it is perceived by users.
- Represented objects using three basic semantic objects:
 - ◆ Entities
 - ◆ Attributes
 - ◆ Relationships

Warning !

- The quality of a database's conceptual model is a critical factor in its resulting efficiency.

Abstraction and Classification

From perceived reality
to representation:

We can classify
physical things (tangible),
or abstract intangible
things to create
classes or sets of entities.

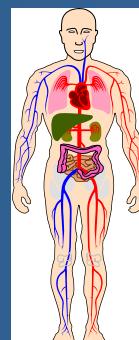


Maud Zoë Paul ... Yves



Entity Type: Person

Attributes: - name,
- age, ...



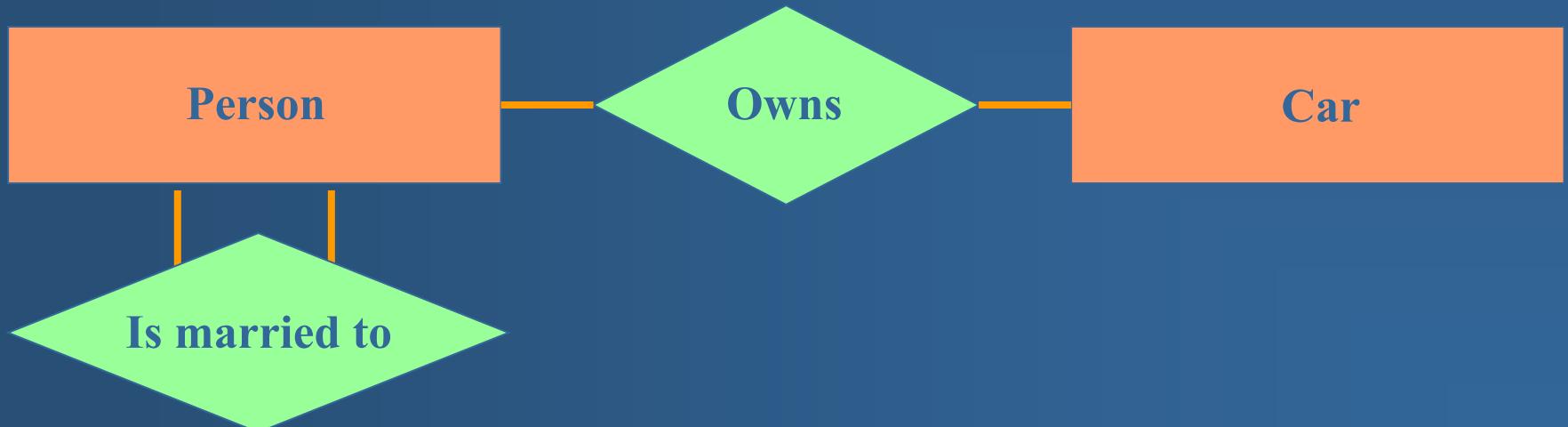
Definition of Entity-Relationship Model

A ER Model is a collection of Types:

Entities Types

Relationships Types

The database will eventually implement data or information described in the ER Model.



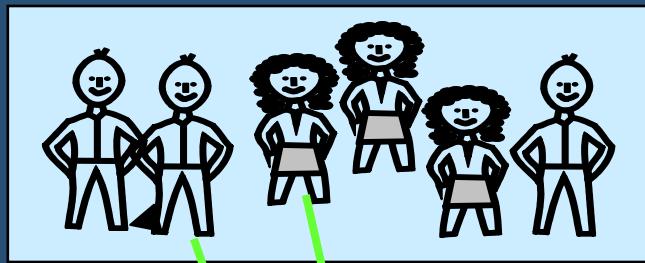
Conceptual Models

- Entity-Relationship–ER
- UML (Unified Modeling Language)
- Others (OO, OR)
 - ◆ Note: the relational model and certain other object oriented models are logical models (objective: implementation)

Some Terms

The real world

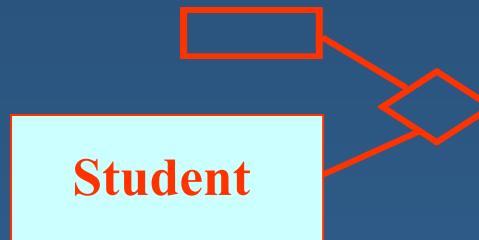
Collection of objects which are perceived as being similar



students

Abstraction / Classification

MODEL



SCHEMA



POPULATION

Database

INSTANCES or OCCURRENCES

Entity Type: Student
-first name
-last name
-date of birth
-section
-year

Relational Databases

Conceptual Modeling

Part 2: The Entity-Relationship Model

Conceptual Model

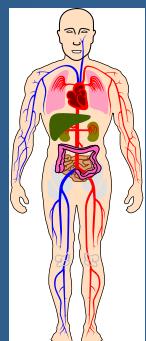
- Conceptual Models permit us to conceptually describe the data structure of our database.
- A Conceptual Model uses three basic components (which we derive from abstractions or classifications of real world things):
 - ◆ real world object <=> entity
 - ◆ real world associations <=> relationship
 - ◆ real world property <=> attribute

Entities and Entity Types

- Entity : an Entity is a “thing” that exists and can be uniquely identified.
- Entity Type (ET) : an ET is a collection of similar entities, with the same characteristics.



Aude Zoë Yves ... Paul



Person

Relationships and Relationship Types

- **Relationship:** a Relationship is a representation of a non oriented link between two or more entities (each of which plays a specific role).
- **Relationship Type (RT) :** a Relationship Type is a representation of a group of similar Relationships, each with the same characteristics.



“Buys” = < 1 person, 1 house >

The function of Relationships

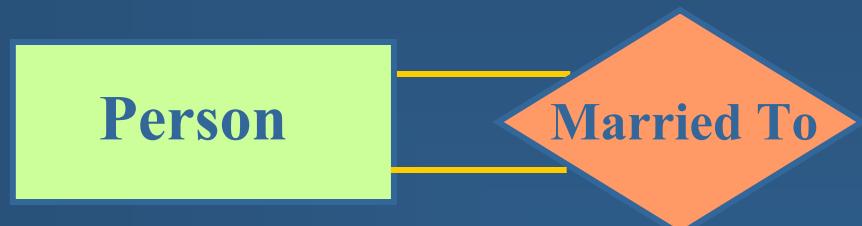
“Buying” = < 1 person, 1 house >



- In an relationship each entity plays a specific role
- Binary relationship: two roles

Cyclic Relationships

Any relationship with 2 or more roles linking the same type of entity.



“Married To” = < 1 personne, 1 personne >

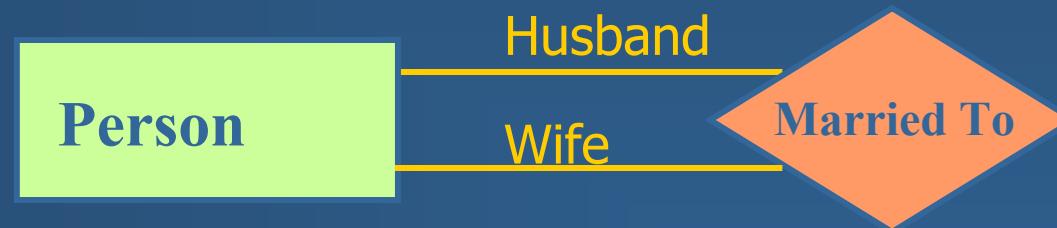
Problem: how do we know who in the couple is the husband and who is the wife?

< Dupont Dominique, Dupont Dominique >

?

Cyclical Relationships: Appointing Roles

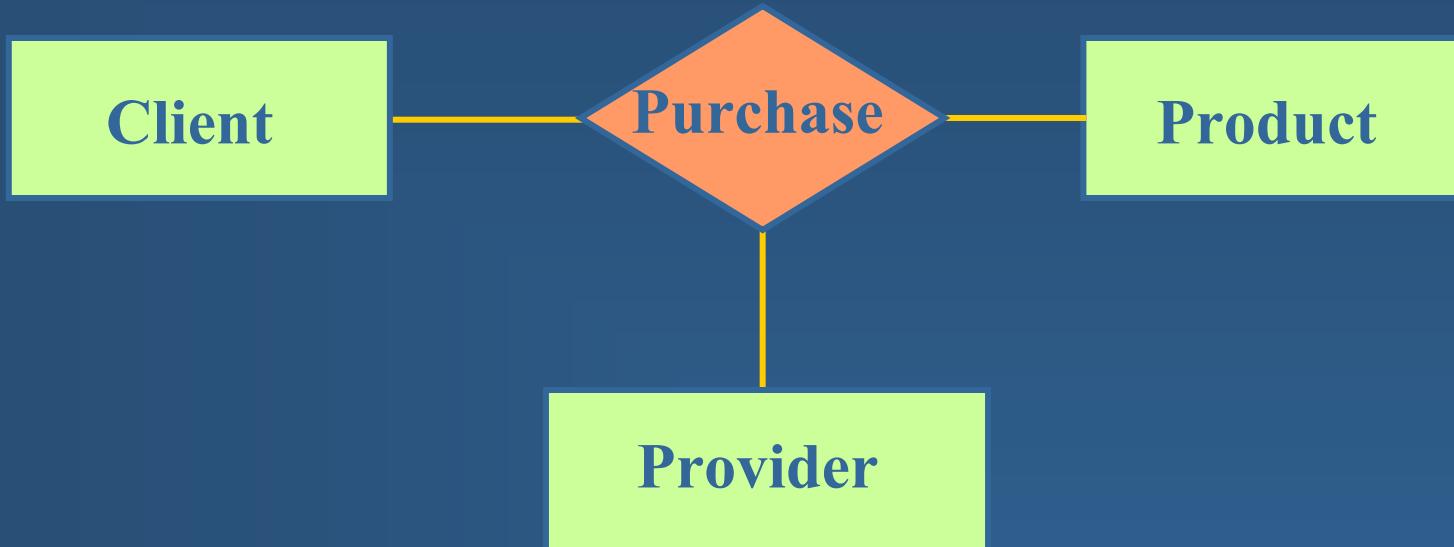
- ? Solution: specify the role of each entity to remove any ambiguity



“Married To” = < 1 person/Husband, 1 person/Wife >

< Dupont Dominique / Wife,
Dupont Dominique / Husband >

Ternary Relationships



'purchases' = < 1 client, 1 product, 1 provider >

Necessarily 3 roles

Cyclic Relationships

■ Ternary:

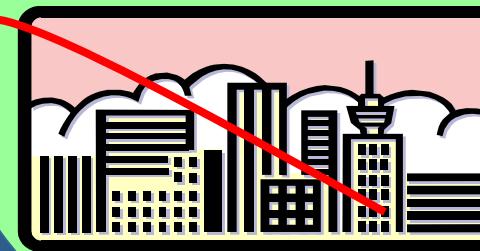
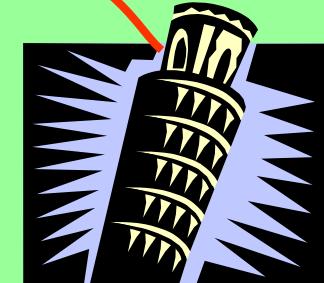
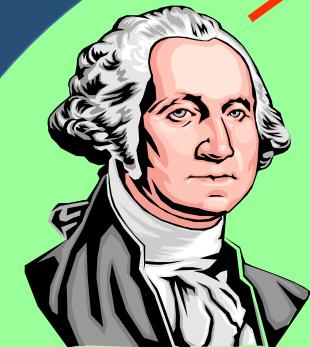


Populating an Entity Type (ET) and a Relationship Type (RT)

Person

Purchase

House



Cardinality of Roles

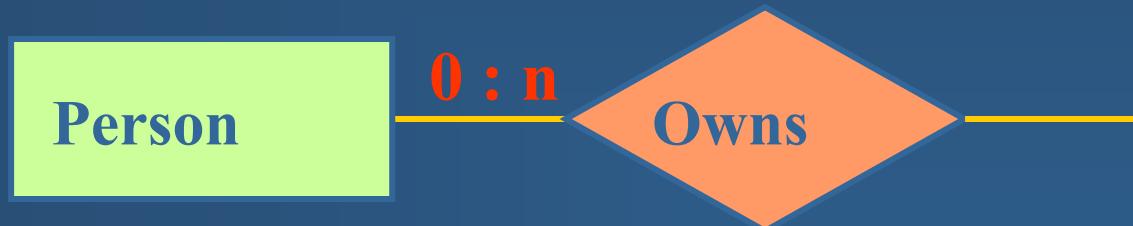


- What is the minimum number of cars a person can own?
- What is the maximum number of cars a person can own?



Cardinality Constraints

- A person might not own a car (0), own 1, 2, ... or n cars (no constraints)



- A car has one and only one owner



Cardinalities : constraints on the data

Populating a Relationship Type

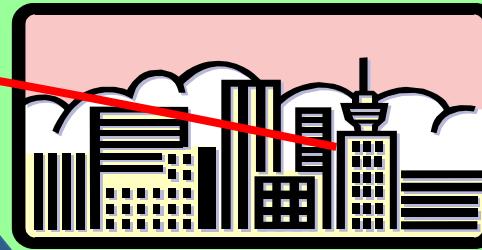
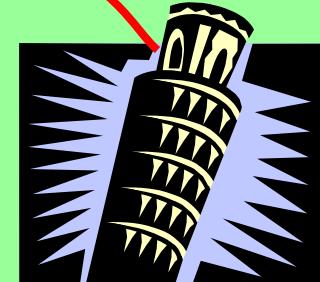
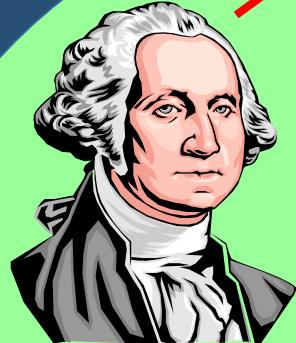
Person

0:n

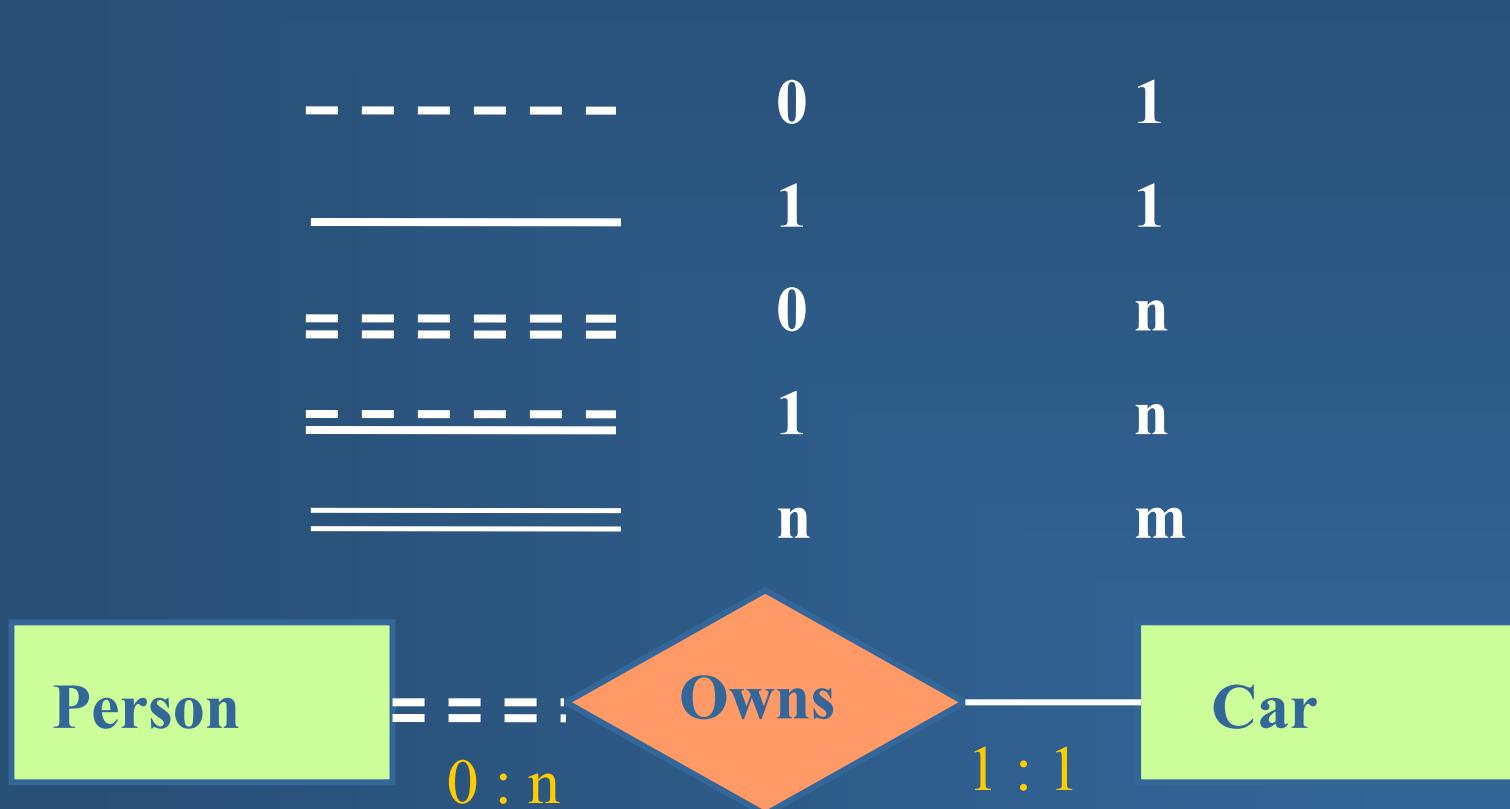
Purchase

1:1

House



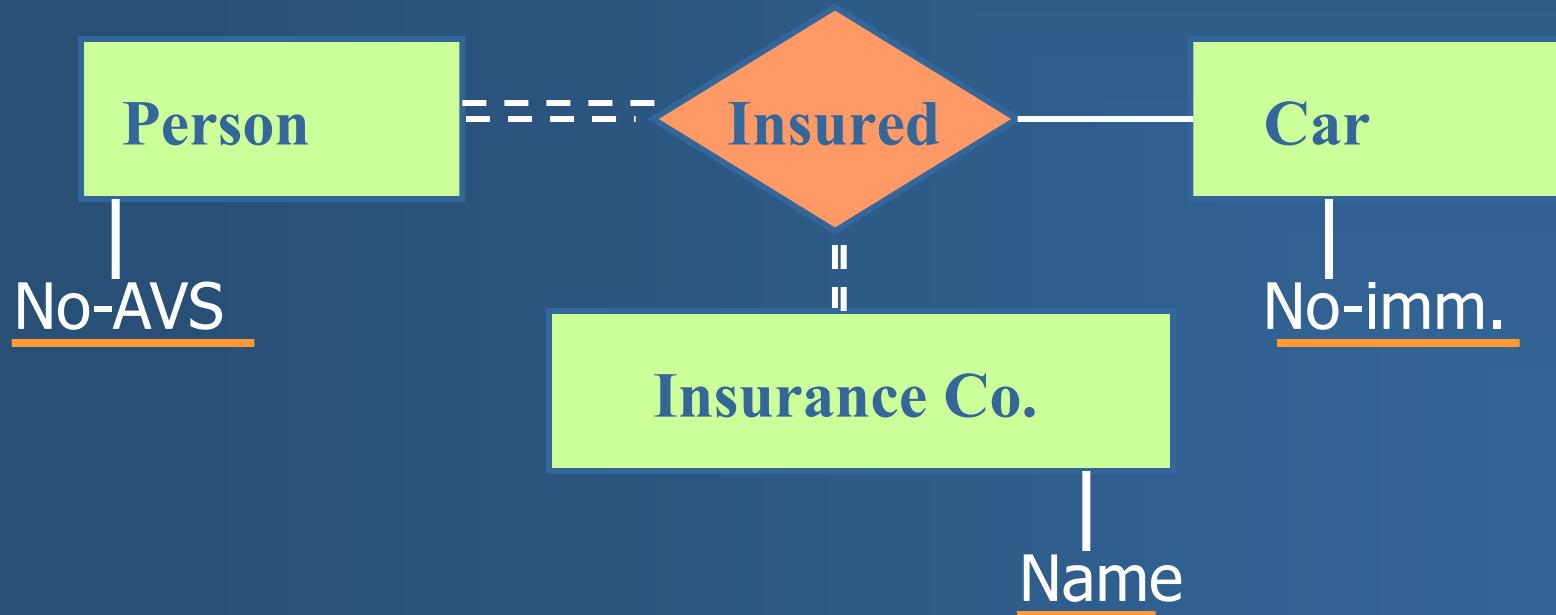
The Values and Notations of Cardinalities



Ternary Relationships: Cardinalities

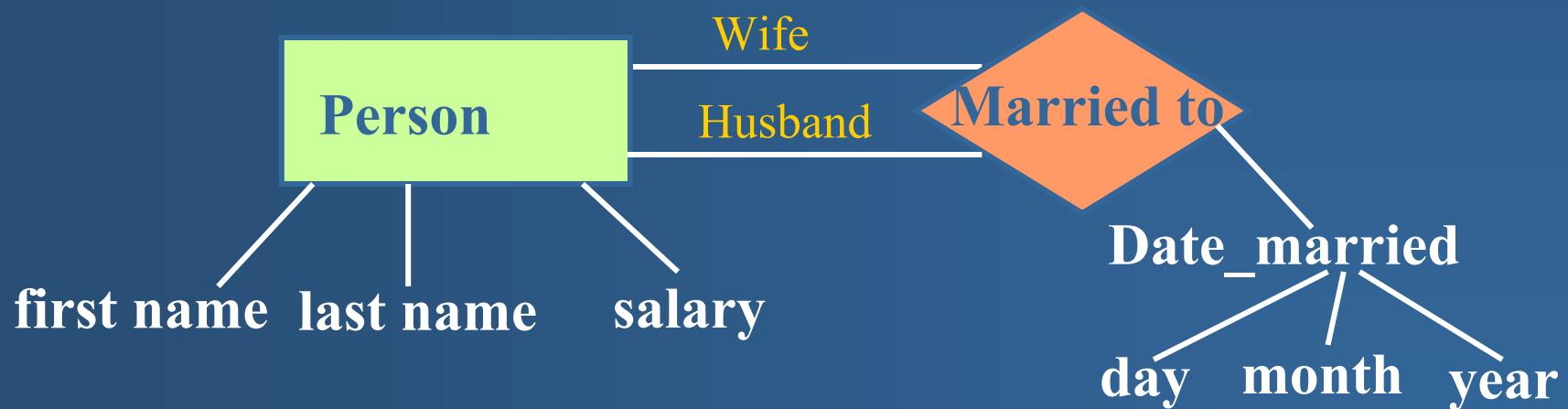
Harder to read on the Schema

The cardinalities define the minimum and maximum RT instances that can have a relationship with a given ET instance.



Attribute

- An Attribute is a property which describes a specific
 - ◆ Entity Type
 - ◆ Relationship Type
 - ◆ or another Attribute



Domaine [1,31]

Atomic Attributes

- **Atomic** : undivisible
 - ◆ Example: day, first name, last name...
- The values in the domain consist of atomic values (numbers, strings...)
 - ◆ Ex.: day - domain of values: {1, 2, ..., 31}
 - ◆ A domain predefines standards, ranges, and enumerations.

Complex Attributes

- **Complex** : decomposable into other attributes
 - ◆ Ex.: date (day, month, year),
address (road, city, postal code)
- A complex attribute does not hold its own atomic values
(no directly associated domain)
- The utility of a complex attribute comes from the attributes which comprise it.
- A component attribute may itself be a complex attribute.

Simple Value and Multi-Valued Attributes

- **Simple Value**: a single value per occurrence
(cardinality max=1)

Ex.: date of birth, AVS number

- **Multi-Valued**: multiple values per occurrence
(cardinality max>1).

Ex.: last name, phone number

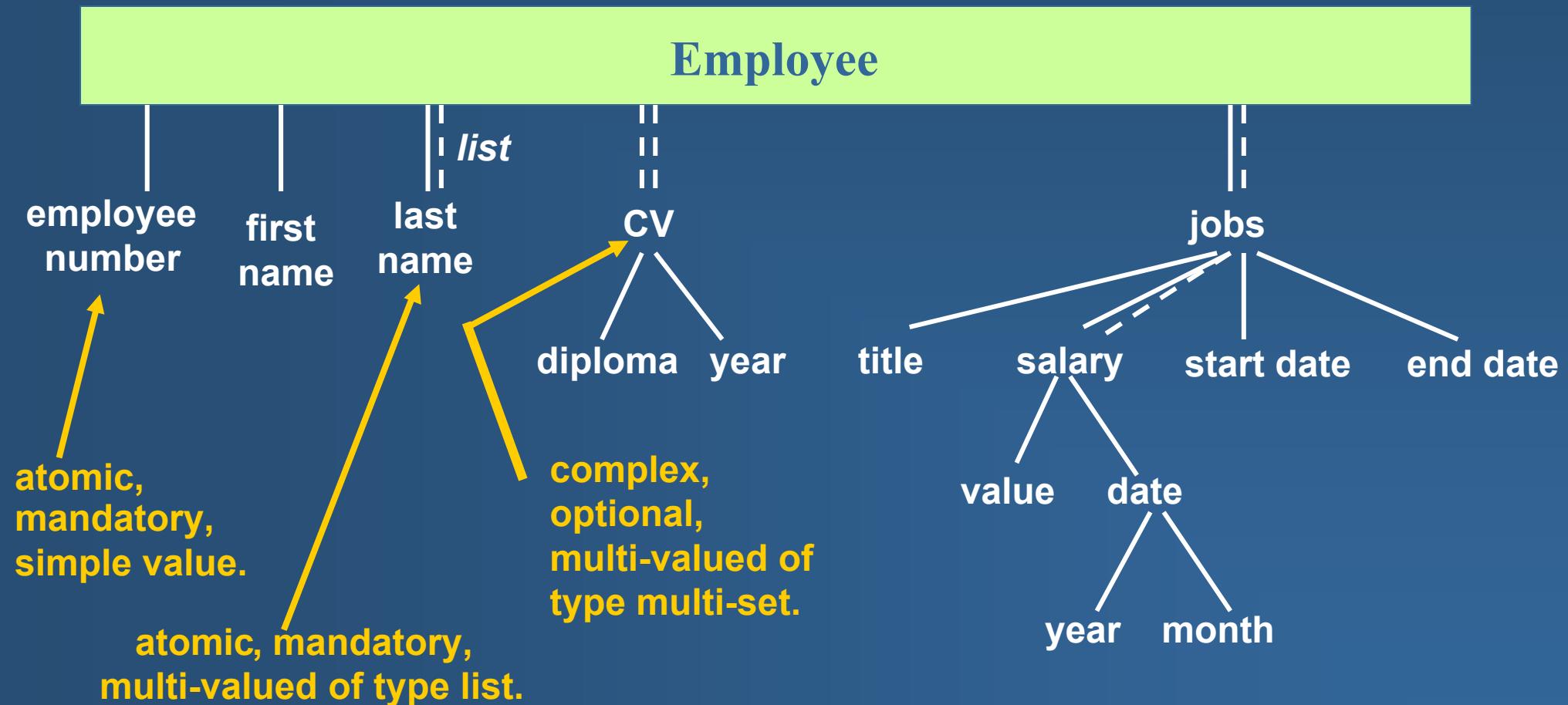
A multi-valued attribute is a set (or list, or multi-set) of values, each defined in the domain associated to attribute.

Mandatory and Optional Attributes

- **Mandatory:** has a value of at least 1.
(cardinality $\text{min}>=1$).
 - ◆ Ex.: first name, last name

- **Optional** : may have a value of 0.
(cardinality $\text{min}=0$).
 - ◆ Ex.: salary, phone number

Attributes: Examples

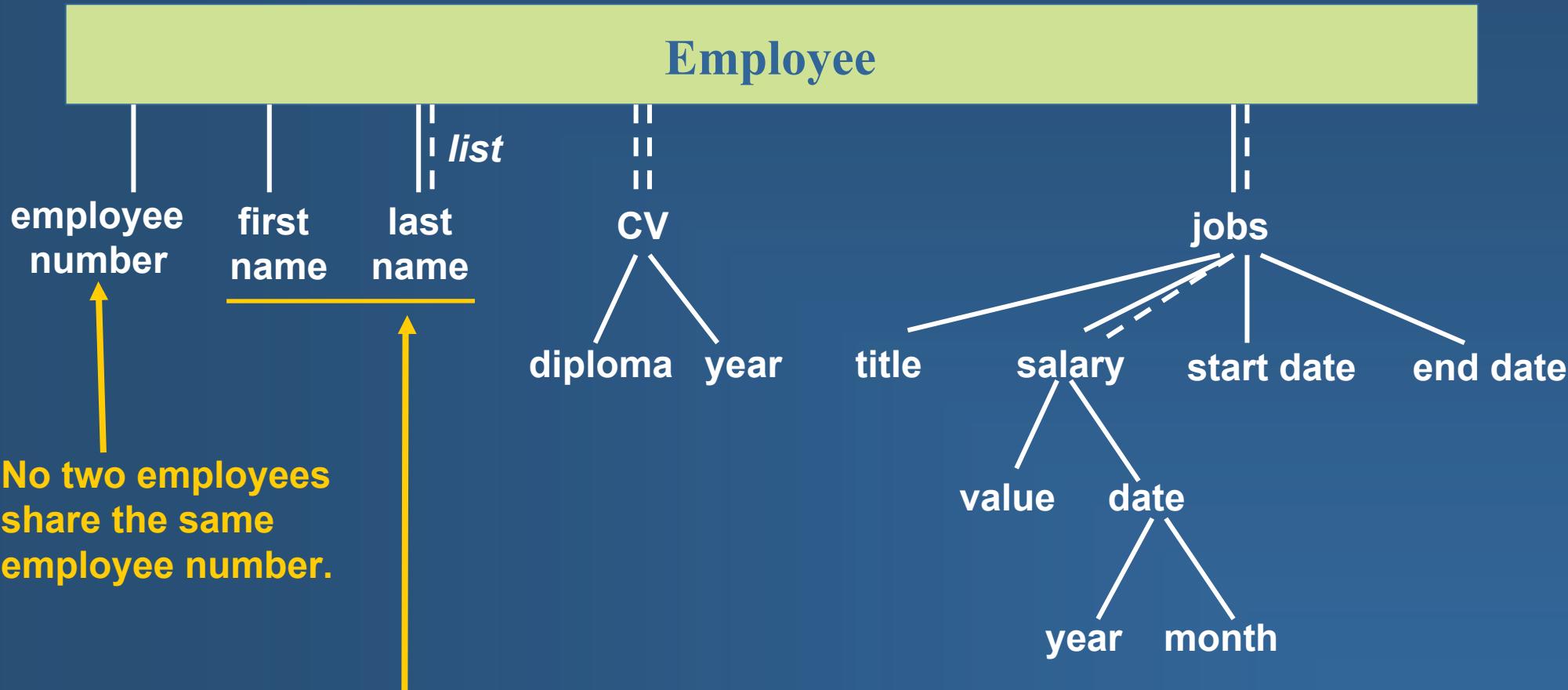


ET and RT Identifiers

- We need to be able to uniquely identity an entity or a relationship
- Identifier:
 - ◆ An Identifier is a unique set of attributes such that no two Entities or Relationships share the same attribute values.

Identifier of the ET Employee

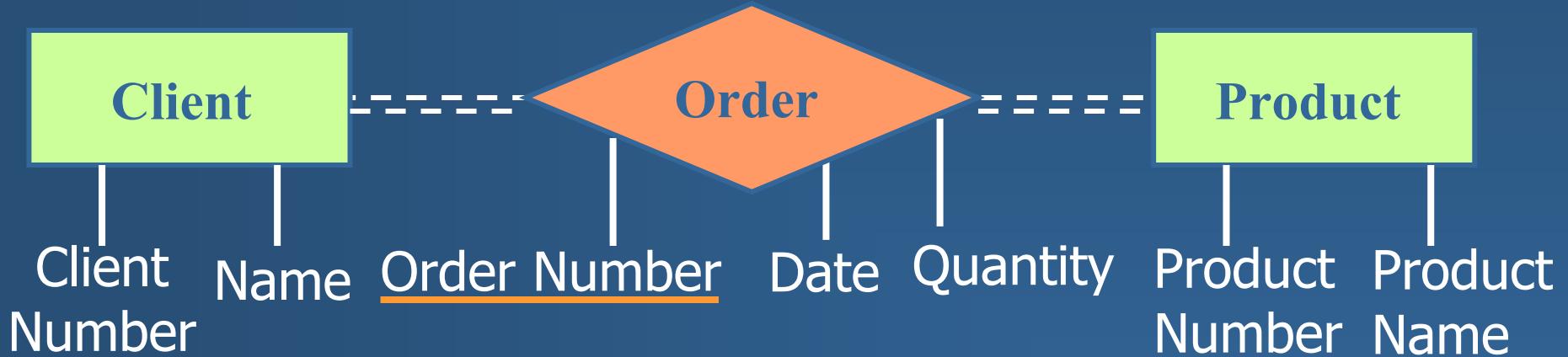
Employee has two IDs: employee number, and first + last name



No two employees share the same first and last name.

Relationship Type Identifiers

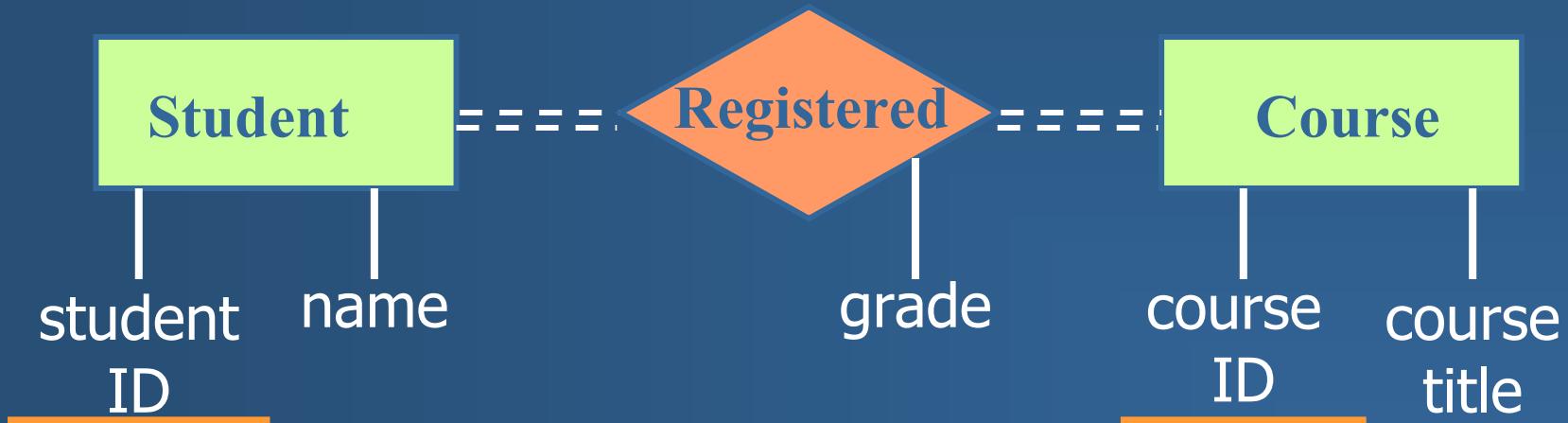
- The Identifier is an attribute of the Relationship Type it uniquely describes.



The Relationship 'Order' has a unique Identifier : **Order Number**

RT Identifiers: Multi-valued Roles

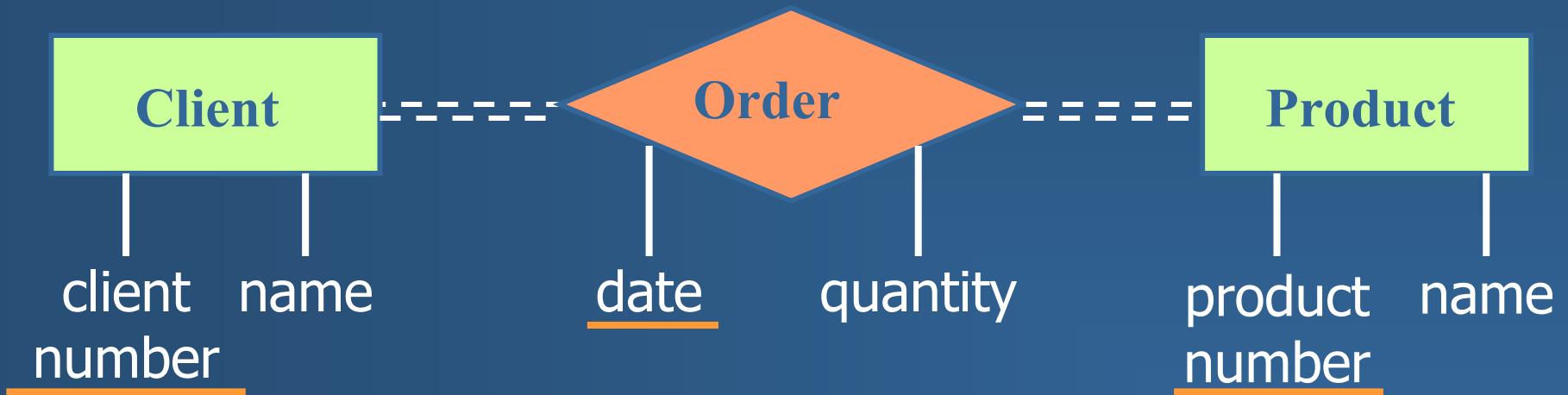
- The most common case if an RT has multi-valued roles:
RT's Identifier = the set of Identifiers of related ET



Identifier of Registered: Student.studentID + Course.courseID

RT Identifier: ET.ID + Attribute Identifier

- A client may order the same product multiple times on different dates

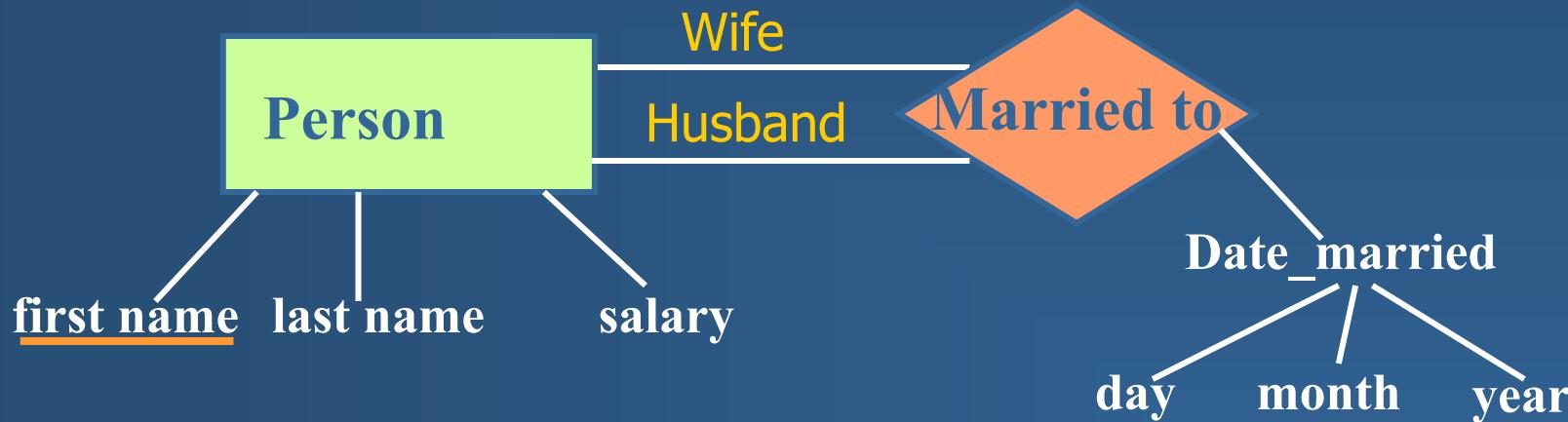


Identifier of Order:

Client.clientNumber + Product.productNumber + Order.date

Cyclic Relationship Type Identifiers

- Same procedure as for other RT



Two simple value roles:

Two Identifiers for 'Married To':

- 1) Person/Wife.firstName
- 2) Person/Husband.firstName

Weak Entity Type Identifiers

An entity which cannot be uniquely identified by its attributes alone: **Weak Entity Type**

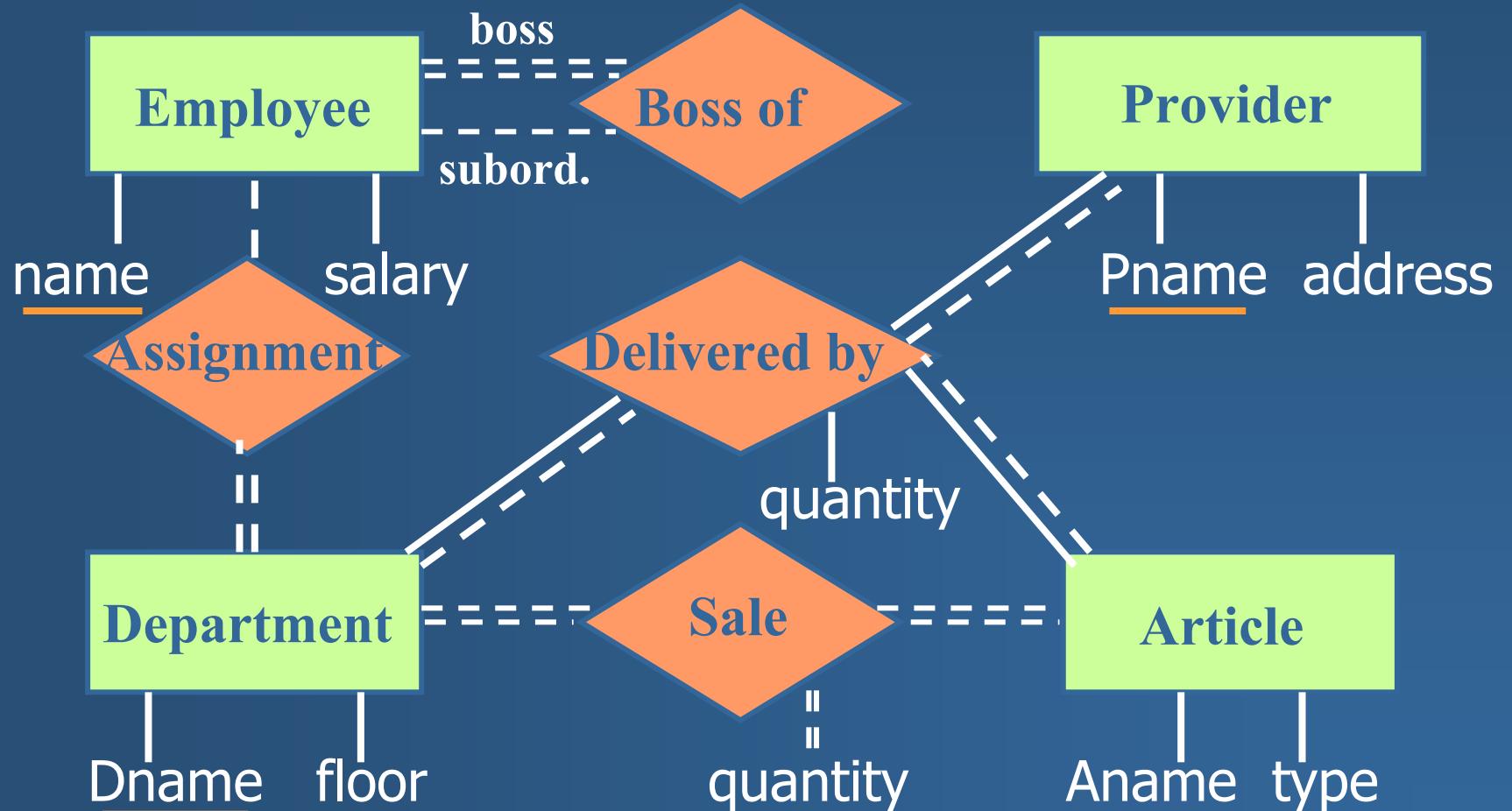


Specimen ID: Book.ISBN + Specimen.specimenNumber

Exists ID: Book.ISBN + Specimen.specimenNumber

Example of an Entity-Relationship Model

Management of a supermarket:



Exercise: Library

- A client may register at the library by paying a deposit. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A book is characterized by its library number (ID), its title, its publisher and its author(s).
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their name, address, date of loan, number of books and their titles.
- Finally, for each book, we want to know its date of purchase, and condition.

Our Entities

- A **client** may register at the **library** by paying a deposit. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A **book** is characterized by its library number (ID), its title, its **publisher(s)** and its **author(s)**.
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their name, address, date of loan, number of books and their titles.
- Finally, for each book, we want to know its date of purchase, and condition.

Our Entity Types

- Proper objects with at least one reference in our schema



Our Entities

- A **client** may register at the **library** by paying a deposit. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A **book** is characterized by its library number (ID), its title, its **publisher(s)** and its **author(s)**.
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their name, address, date of loan, number of books and their titles.
- Finally, for each book, we want to know its date of purchase, and condition.

Our Entity Types

- Proper objects with at least one reference in our schema



Author

?

Library

Publisher

Client

Our Entities

- A **client** may register at the **library** by paying a deposit. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A **book** is characterized by its library number (ID), its title, its **publisher(s)** and its **author(s)**.
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their name, address, date of loan, number of books and their titles.
- Finally, for each book, we want to know its date of purchase, and condition.

Our Entity Types

- Proper objects with at least one reference in our schema

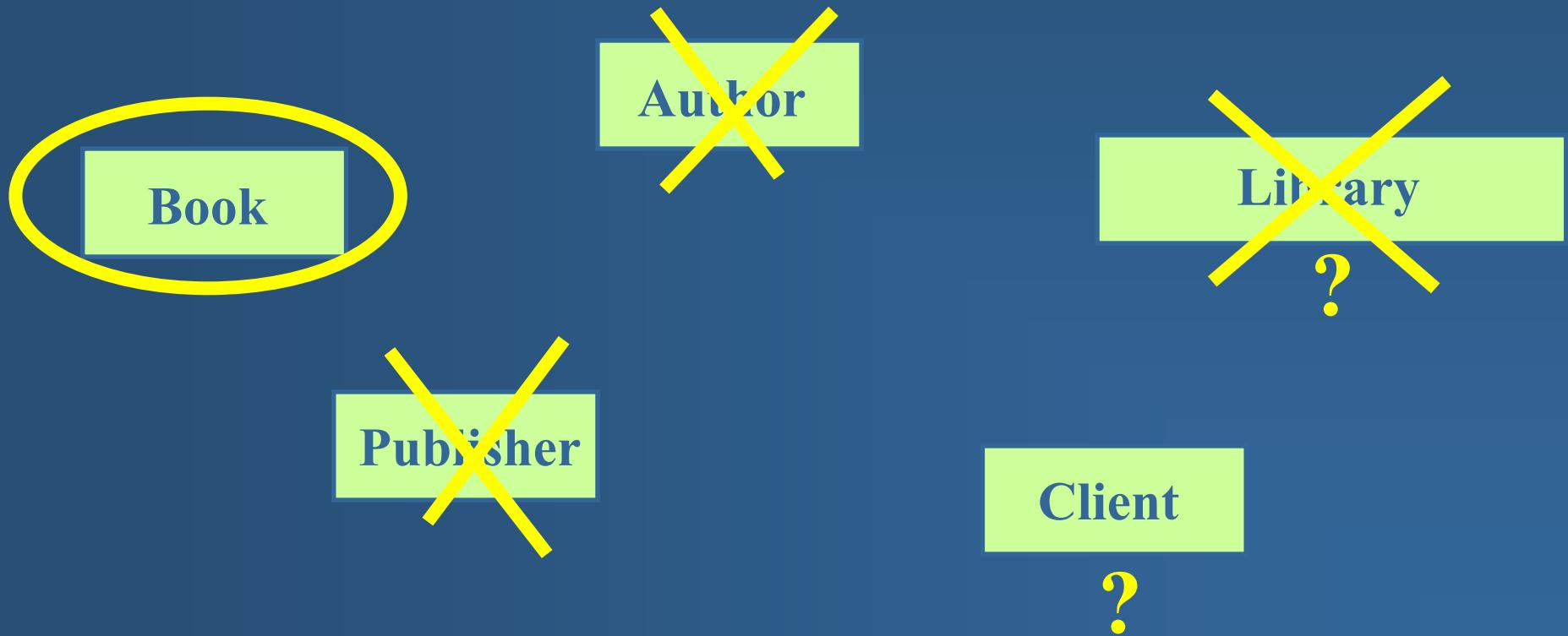


Our Entities

- A **client** may register at the **library** by paying a deposit. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A **book** is characterized by its library number (ID), its title, its **publisher(s)** and its **author(s)**.
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their name, address, date of loan, number of books and their titles.
- Finally, for each book, we want to know its date of purchase, and condition.

Our Entity Types

- Proper objects with at least one reference in our schema

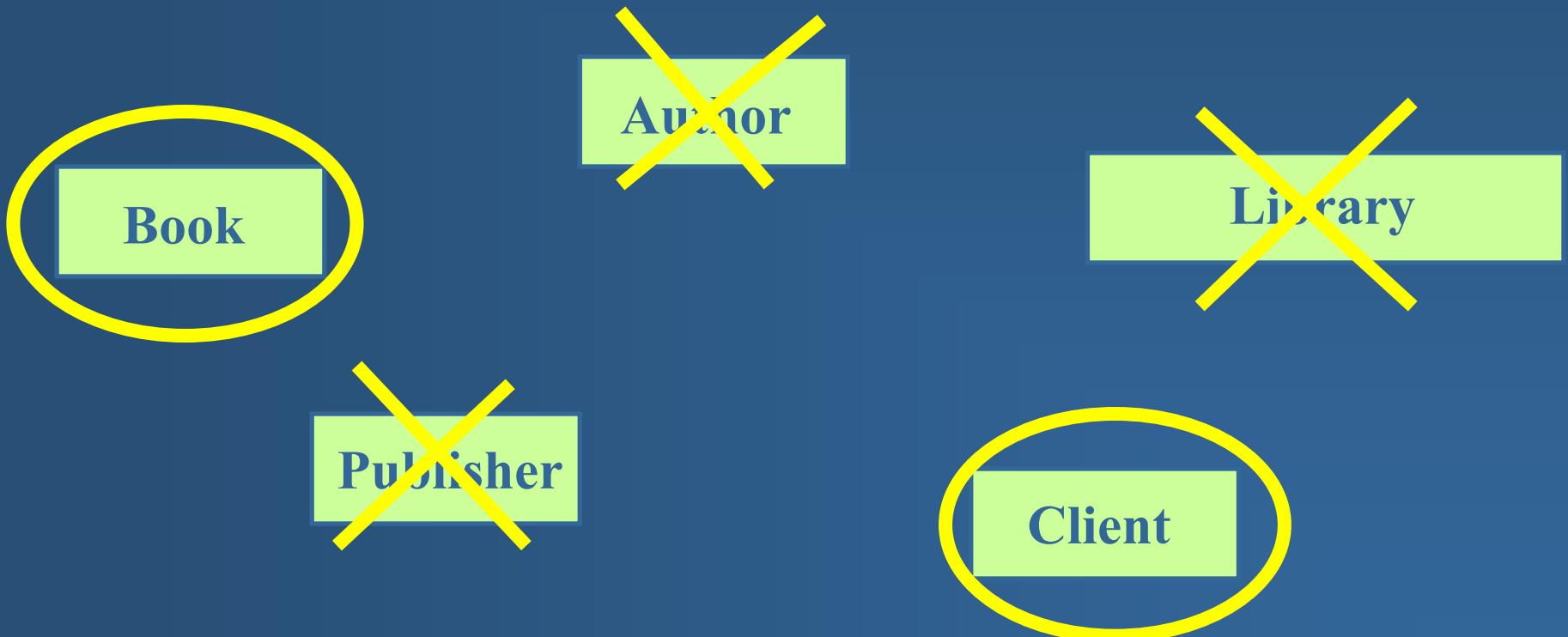


Our Entities

- A **client** may register at the **library** by paying a deposit. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A **book** is characterized by its library number (ID), its title, its **publisher(s)** and its **author(s)**.
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their name, address, date of loan, number of books and their titles.
- Finally, for each book, we want to know its date of purchase, and condition.

Our Entity Types

- Proper objects with at least one reference in our schema



Attributes of Entity Type Book

- A client may register at the library by paying a deposit. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A **Book** is characterized by its **library number** (ID), its **title**, its **publisher(s)** and its **author(s)**.
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their name, address, date of loan, number of books and their titles.
- Finally, for each book, we want to know its **date of purchase**, and **condition**.

Entity Type Book and It's Attributes



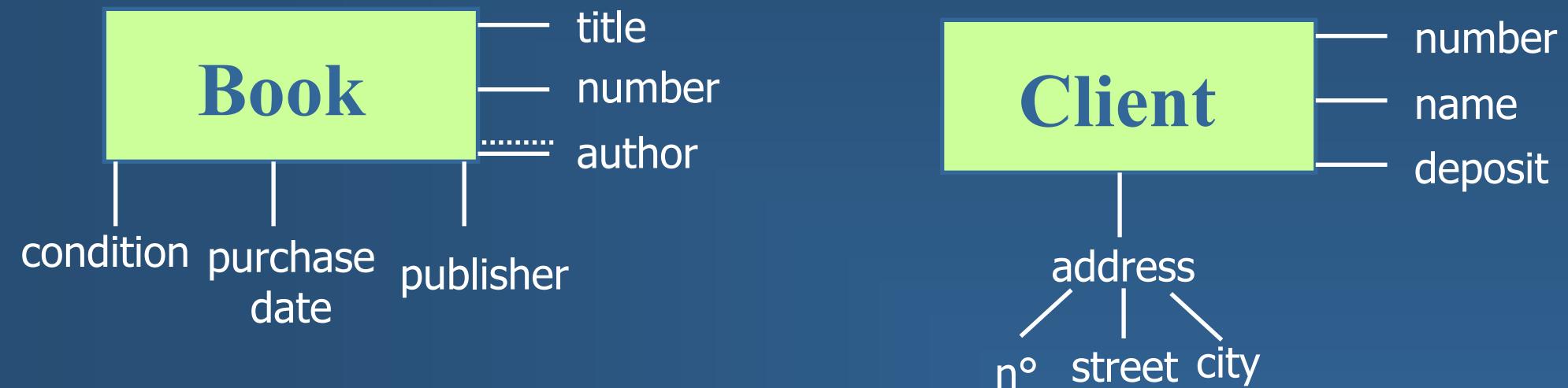
The ET Client and its attributes

- A **Client** may register at the library by paying a **deposit**. Depending on the amount of this fee, they will be able to make up to 10 loans at the same time.
- Loans last as maximum of 15 days.
- A book is characterized by its library number (ID), its title, its publisher(s) and its author(s).
- For each client, we want to be able to obtain all the books which they have borrowed (number of loans, book numbers, titles, date of loans)
- Every week, we want to publish a list of all late borrowers: their **name**, **address**, date of loan, number of books and their titles.
- Finally, for each book, we want to know its date of purchase, and condition.

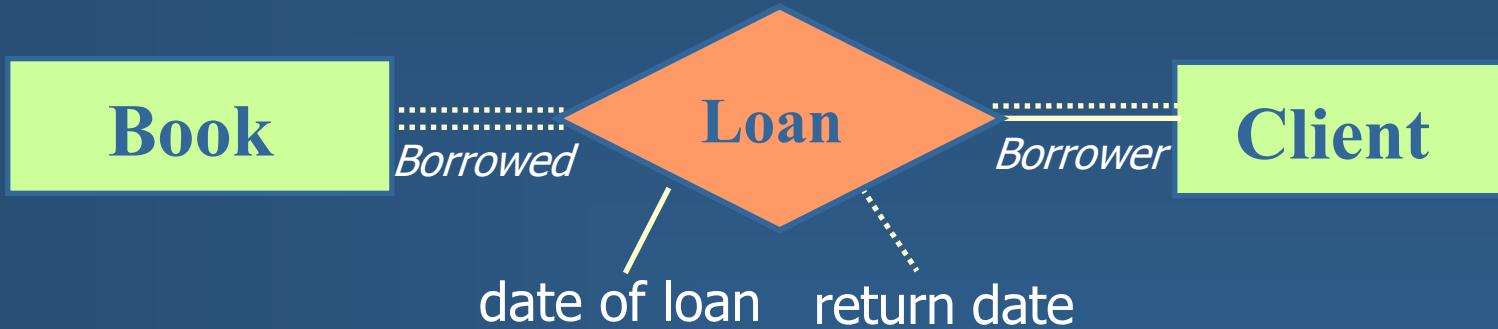
The Client Entity Type and its Attributes



The Entity Types of our Conceptual Model

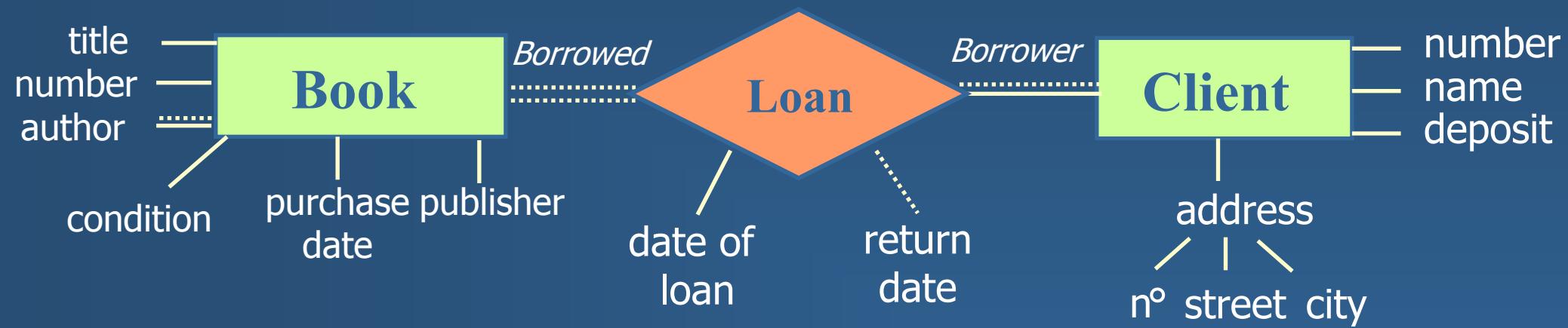


Relationship Types



- represents a nondirectional link between two entities who each play a specific role

Conceptual Model of a Library



Description of the Entity-Relationship Model

- Entity Types
- Relationship Types
- Attributes
- Types of Identifiers
- Attribute Domains

Description of an Entity Type

- name of the Entity Type;
- name(s) of Entity subTypes(s) which the Entity contains (if any);
- a free definition stating the semantic representation of the Entity Type
 - ◆ Exact characterization of the population of the Entity Type
- description of the attributes of the Entity Type;
- composite identifiers of the Entity Type, if it exists.

Description of a Relationship Type

- name of the Relationship Type
- a free semantic definition precisely describing the Relationship Type
- names of the participating Entity Types, and their respective roles.
- a cardinality for each Role.
- description of the attributes of the RT, if they exist.
- components of the identifiers of the RT, if they exist.

Description of an Attribute

- name of the attribute
- free semantic definition
- cardinalities
- if it is a simple attribute: domain of values
- if it is a complex attribute: description of the composite attributes

Domain of a Simple Attribute

- The Domain of an attribute is defined by the set of values permitted by the attribute.
- The Domain of a Simple Attribute can be
 - ◆ a domain of type: whole number, real number, string, bool, date
 - ◆ a domain of type with restriction:
Month Attribute: $[>=1, <=12]$
 - ◆ an enumerated domain:
 $\{\text{january}, \text{february}, \dots, \text{december}\}$
 - ◆ a domain defined by the user.