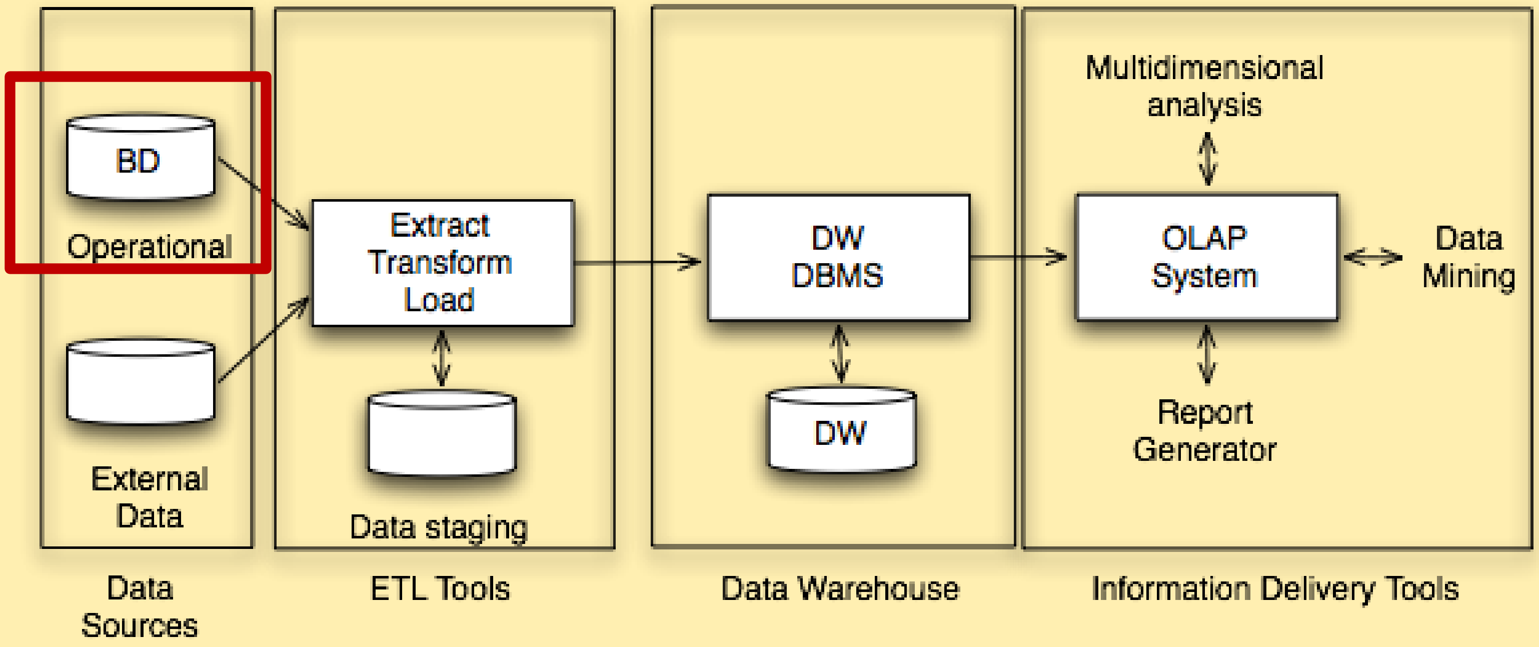
RECALLS ON OBJECT DATA MODEL

- Today, we recall notions of Information Modelling
  - **Object Data Model** (ODM)

  that are needed to understand the Operational DB in input to DW



Operational Systems

What is modeled in a DB?

- Data are organized in a **DB**.
- Data are managed by a **traditional DBMS**.
- The applications **are used to perform** structured business operational activities.

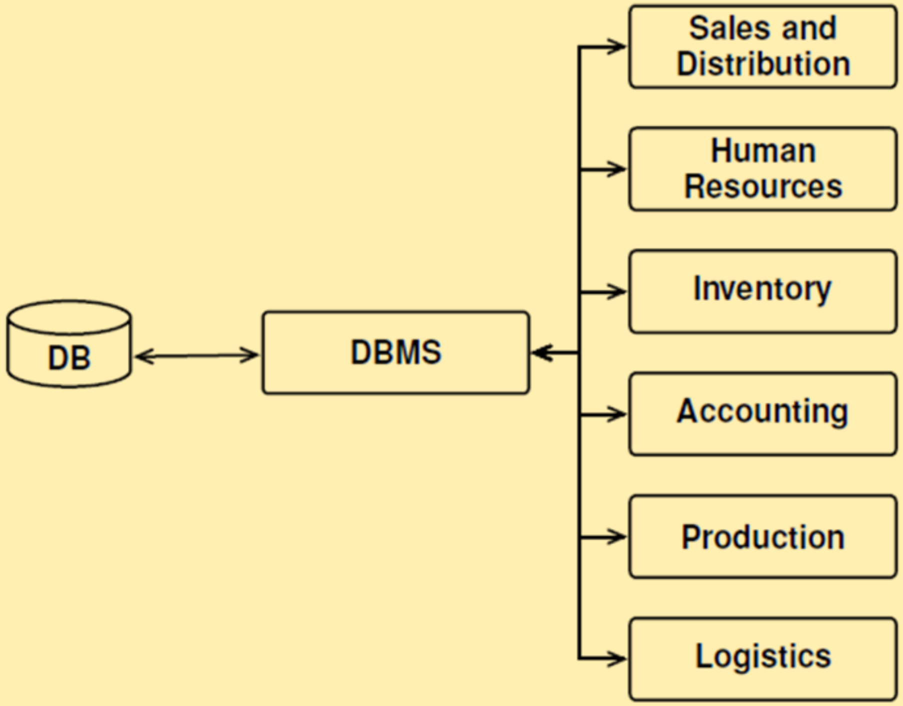


Figure 1.1: Transaction processing system

INFORMATION MODELING

- A **symbolic model** is a subjective formal representation of ideas and knowledge about some aspects of the real world (*domain of discourse*), designed to serve an explicit purpose.

INFORMATION MODELING

UNIVERSITÀ DI PISA

- A **symbolic model** is a subjective formal representation of ideas and knowledge about some aspects of the real world (*domain of discourse*), designed to serve an explicit purpose.

What is the problem?

- Conceptual data model: to analyse a problem, given user requirements
  - E.g., E-R or Entity-Relationship, **ODM or Object Data Model**

How to solve it?

- Logical model: to design a solution independently of actual DBMS
  - E.g., **Relational Data Model**

How to implement a solution?

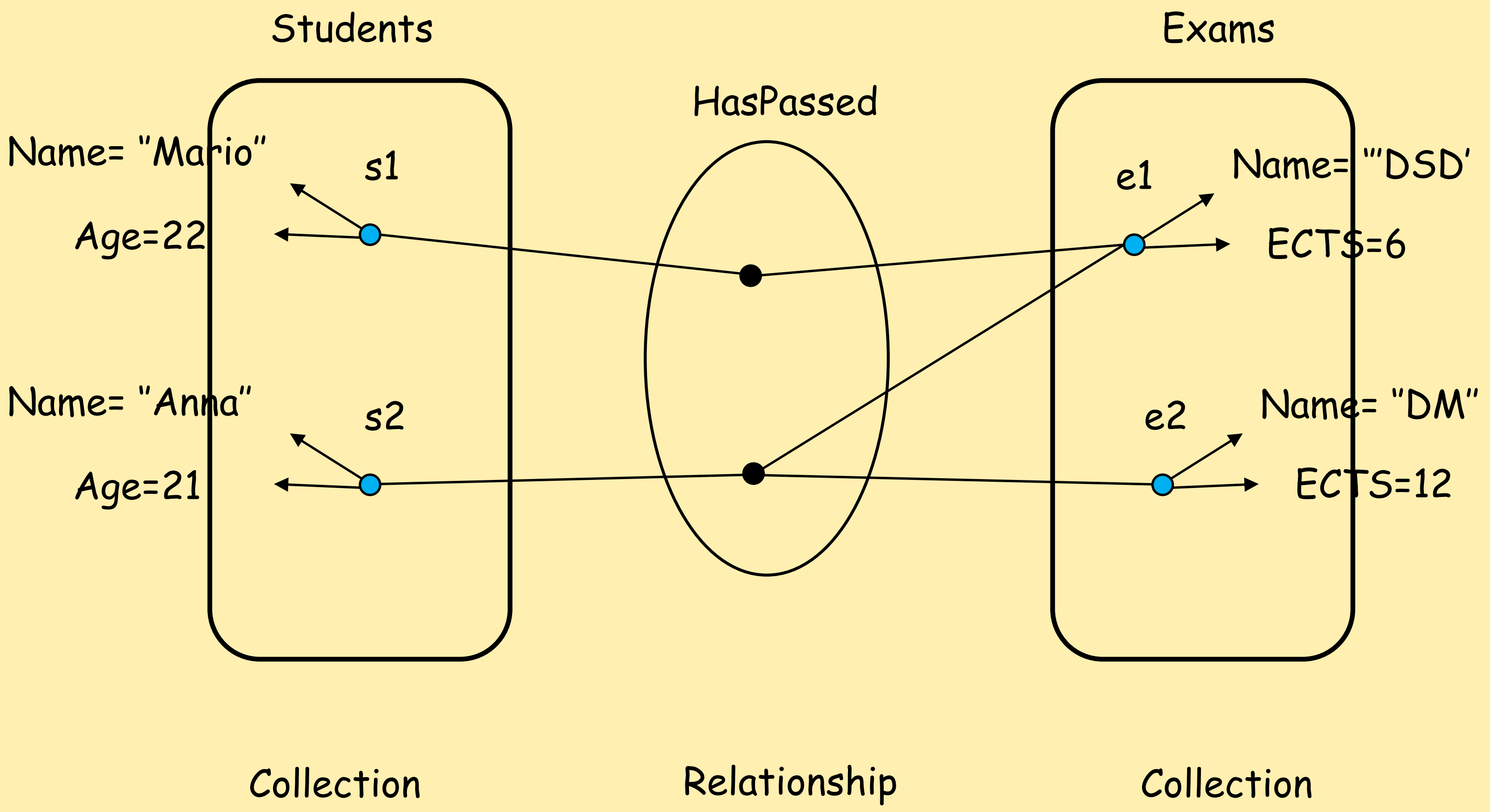
- Physical model: to realize a project on a specific DBMS

- **Operational databases**: what to model?

INFORMATION MODELING

- **Concrete knowledge**: specific facts known of the system to be modelled
  - An **entity** is anything for which certain facts should be recorded, independently from the existence of other entities
    - E.g., Mario is a Student, DS&BI is a Master Programme
  - A **property** is a fact about an entity which is not meaningful in itself, but only because it describes an entity of interest
    - E.g., Age of Mario is 22
  - A **collection** (or **class**) is a set of entities with the same properties
    - E.g., {Mario, Anna, …}
  - A **relationship** is a fact which correlates independent entities.
    - Mario *is enrolled at* DS&BI

Collections and Relationships



INFORMATION MODELING

- **Abstract knowledge**: structure of the concrete knowledge and restrictions on the admissible values
  - Property **type**
    - E.g., *Age* : int
  - Entity **type**
    - E.g., *Age*:int, Name:string
  - Collection **type**
    - E.g., { *Age*:int, Name:string }
  - Relationships have
    - **Cardinality**, one (1) or many (N), to specify how many entities of one collection may be associated with entities of another collection.
      - *Is enrolled at* has cardinality (N, 1), *HasPassed* exams has cardinality (N, N)
    - **Partecipation**, total or partial, to specify whether an entity of one collection can have entities of another collection associated to it.
      - *Is enrolled at* is total, *HasPassed* exams is partial

Data models

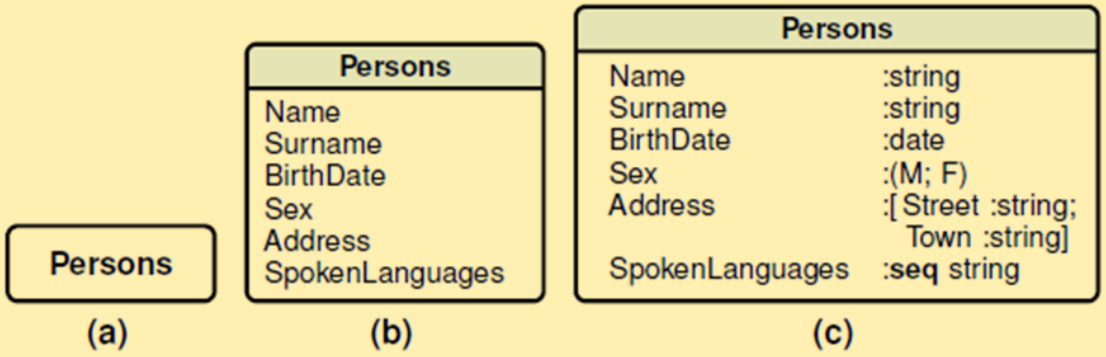
- A **symbolic model** is a subjective formal representation of ideas and knowledge about some aspects of the real world (*domain of discourse*), designed to serve an explicit purpose.

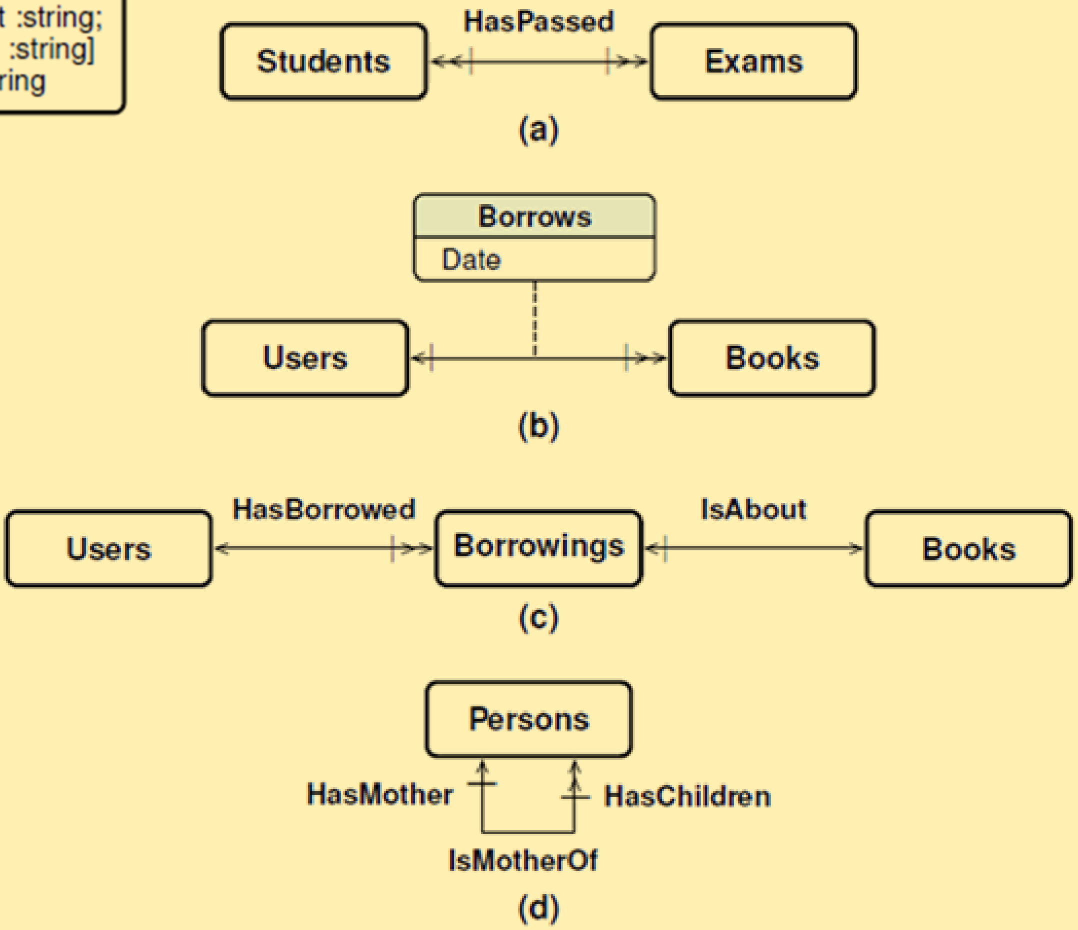
- A **data model** is a set of abstraction mechanisms to describe abstract knowledge
  - **Object data model**
    - Entity -> Object
    - Entity type -> Object type
    - Collection type -> Class
    - Relationships
    - Inheritance

Property -> (Attribute, Value)

Property types -> Attribute type

Classes and Relationships





Data models

- A **symbolic model** is a subjective formal representation of ideas and knowledge about some aspects of the real world (*domain of discourse*), designed to serve an explicit purpose.

- A **data model** is a set of abstraction mechanisms to describe abstract knowledge
  - **Object data model**
    - Entity -> Object
    - Entity type -> Object type
    - Collection type -> Class
    - Relationships
    - Inheritance

    Property -> (Attribute, Value)

    Property types -> Attribute type

- A **schema** is the abstract knowledge of a domain of discourse expressed by using a specific data model. A schema is a symbolic model.

An ODM Schema

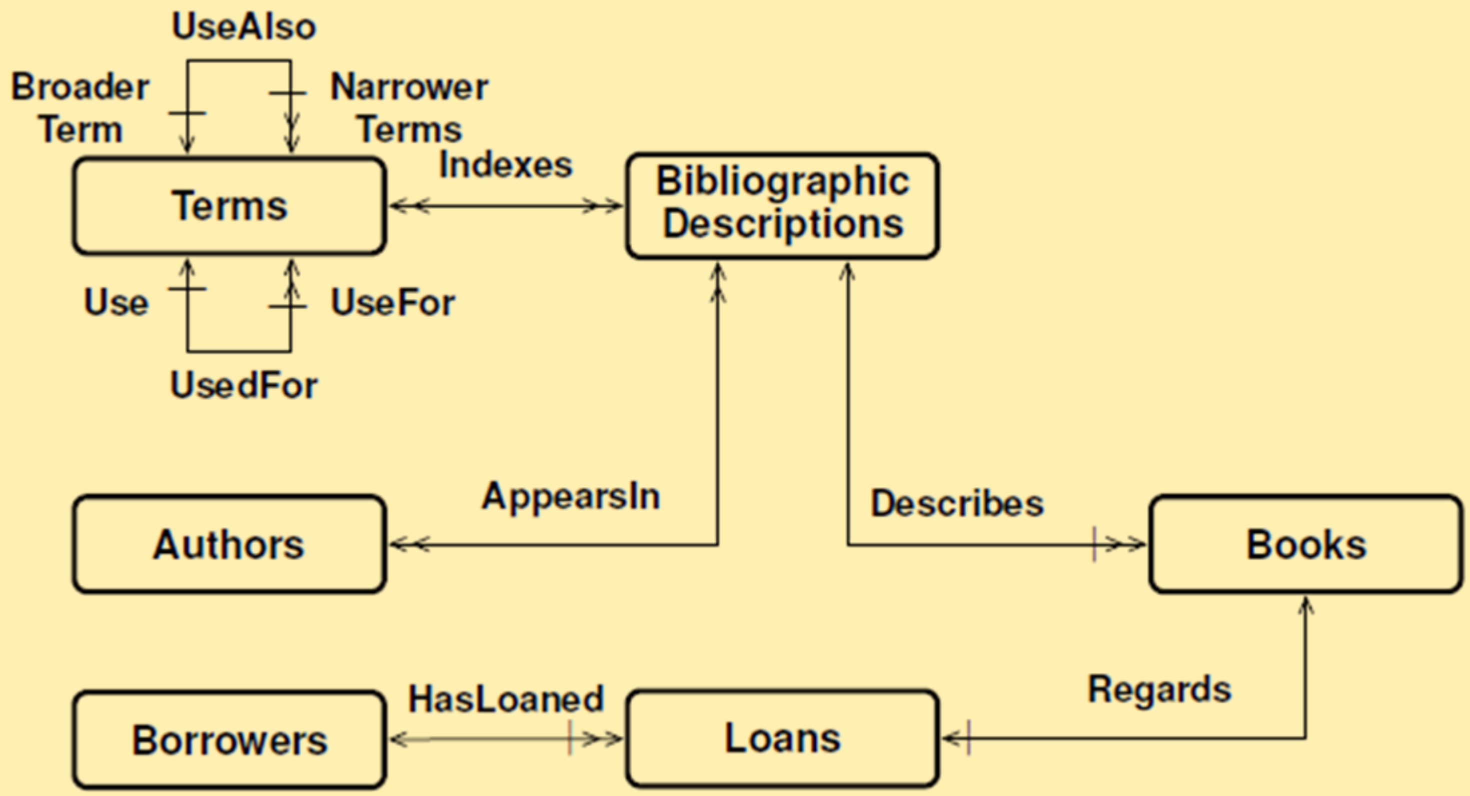
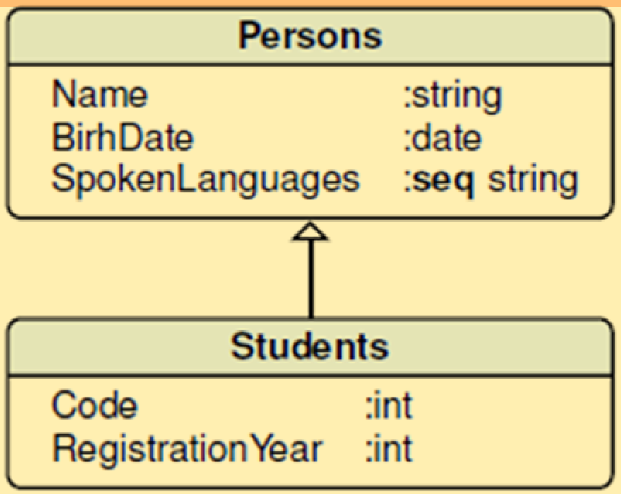
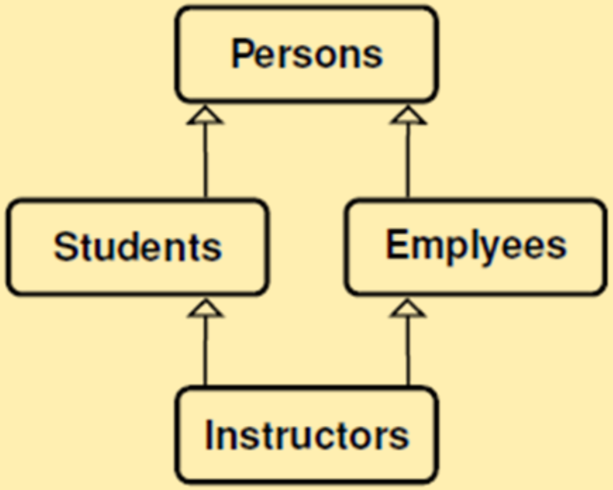
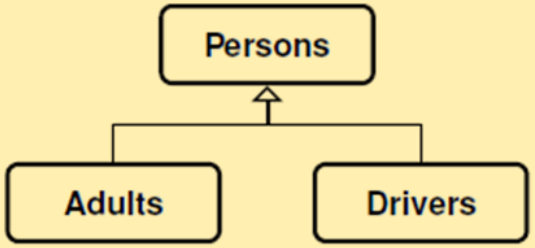


Figure 2.4: A schema for a library

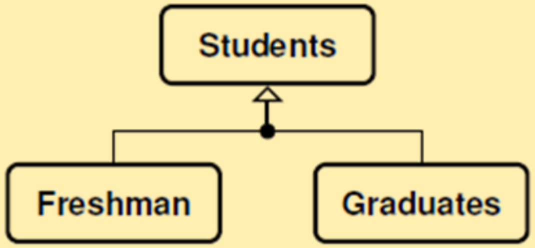
Inheritance: Super and sub-classes



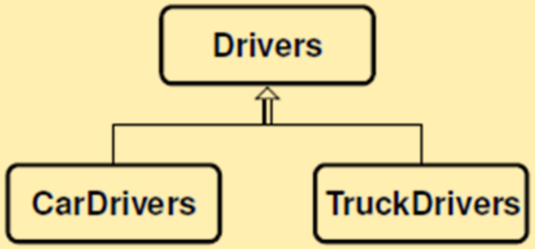




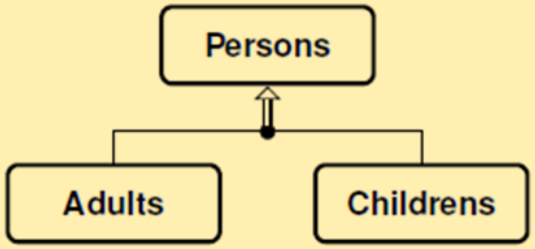
(a) *Overlapping subsets*



(b) *Non overlapping subsets*



(c) *Overlapping cover*



(d) *Non overlapping cover*

A refined ODM Schema

UNIVERSITÀ DI PISA

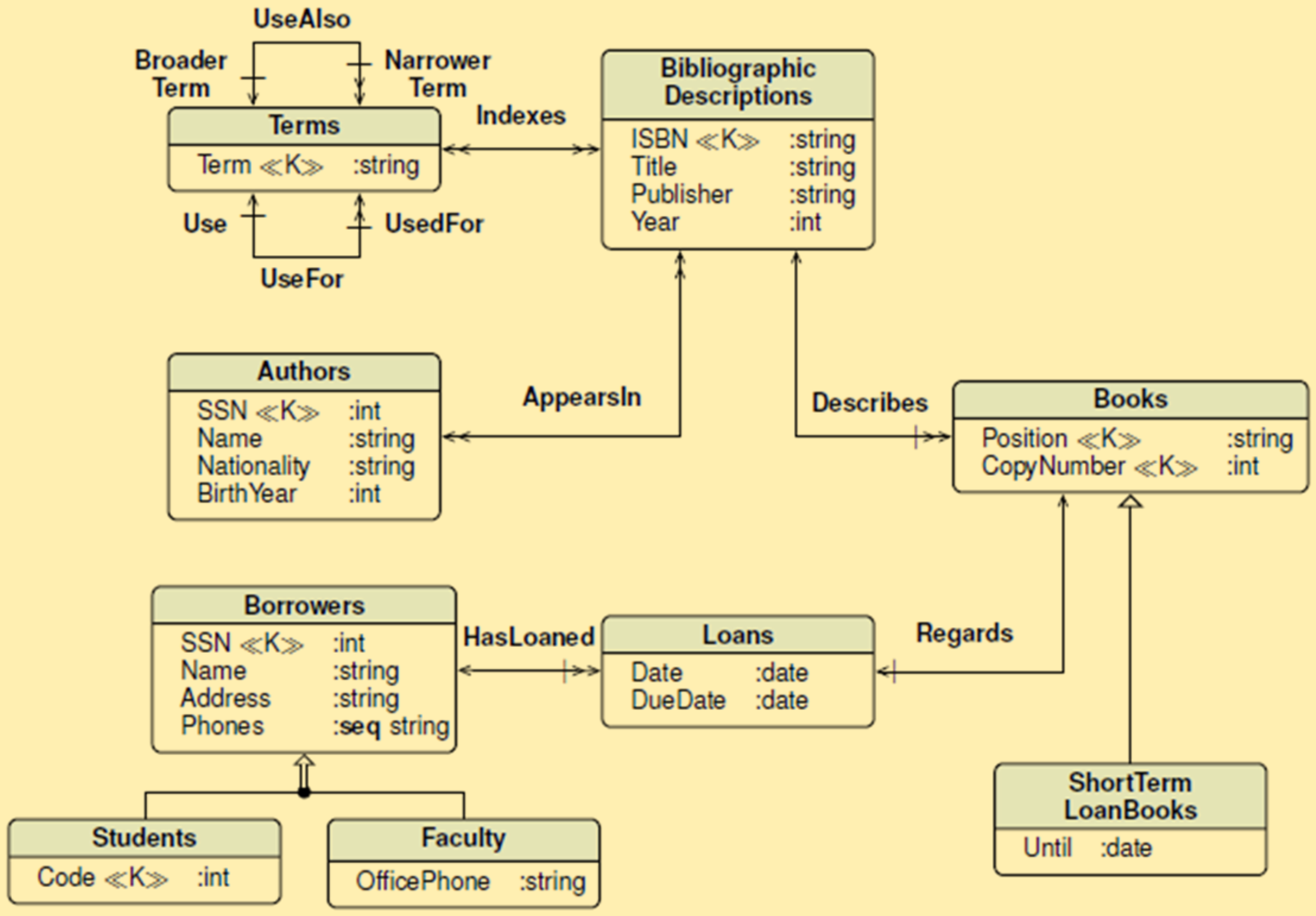
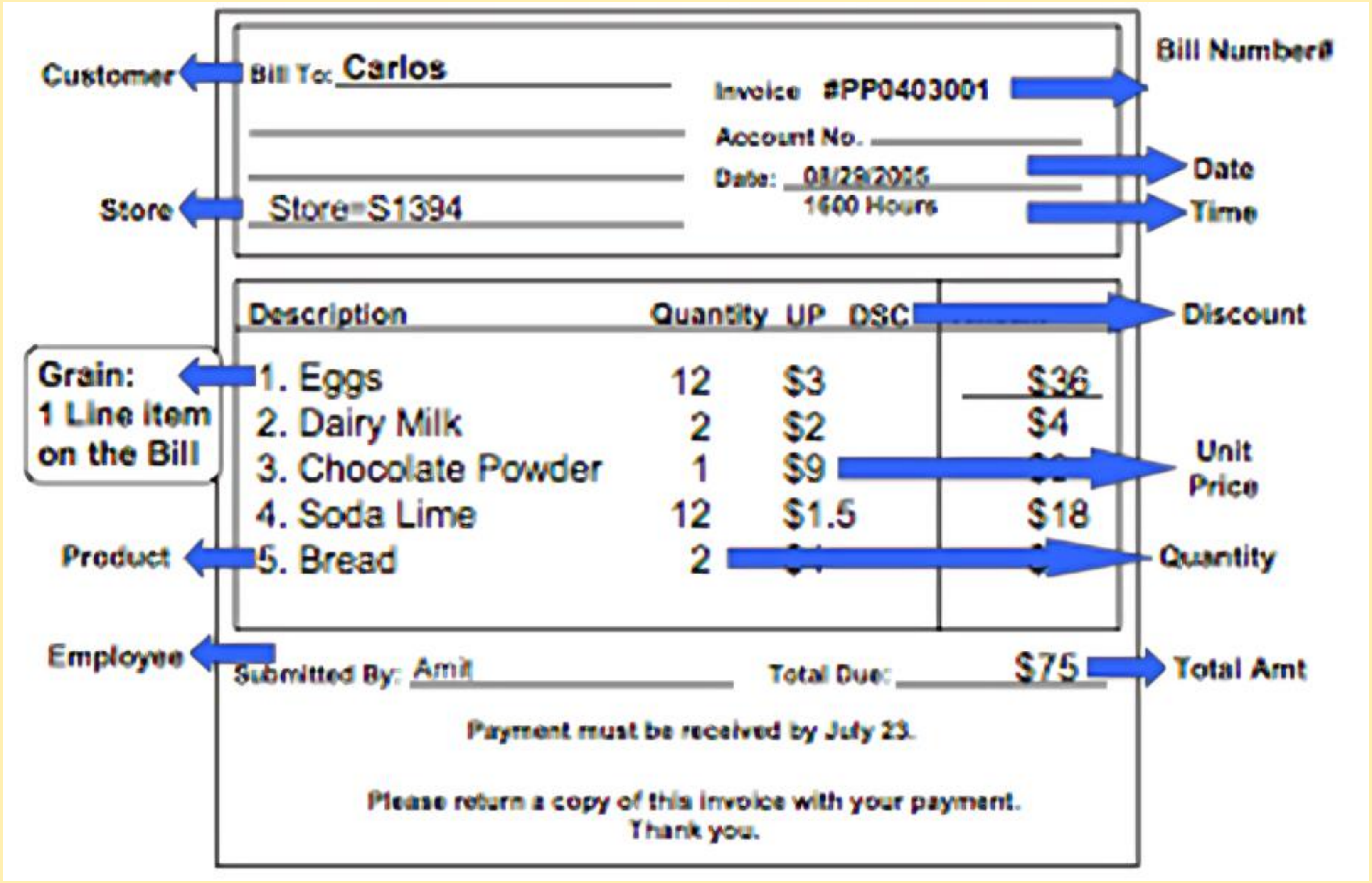


Figure 2.8: A refined schema for a library with class attributes

Product Orders


Customer
Store
Grain:
1 Line item on the Bill
Product
Employee
Bill Number#
Date
Time
Discount
Unit Price
Quantity
Total Amt

Bill To: Carlos

Invoice #PP0403001

Account No.

Date: 03/29/2016

1600 Hours

Store=S1394

Description	Quantity	UP	DSC	
1. Eggs	12	$3		$36
2. Dairy Milk	2	$2		$4
3. Chocolate Powder	1	$9		[illegible]
4. Soda Lime	12	$1.5		$18
5. Bread	2	[illegible]		[illegible]

Submitted By: Amit

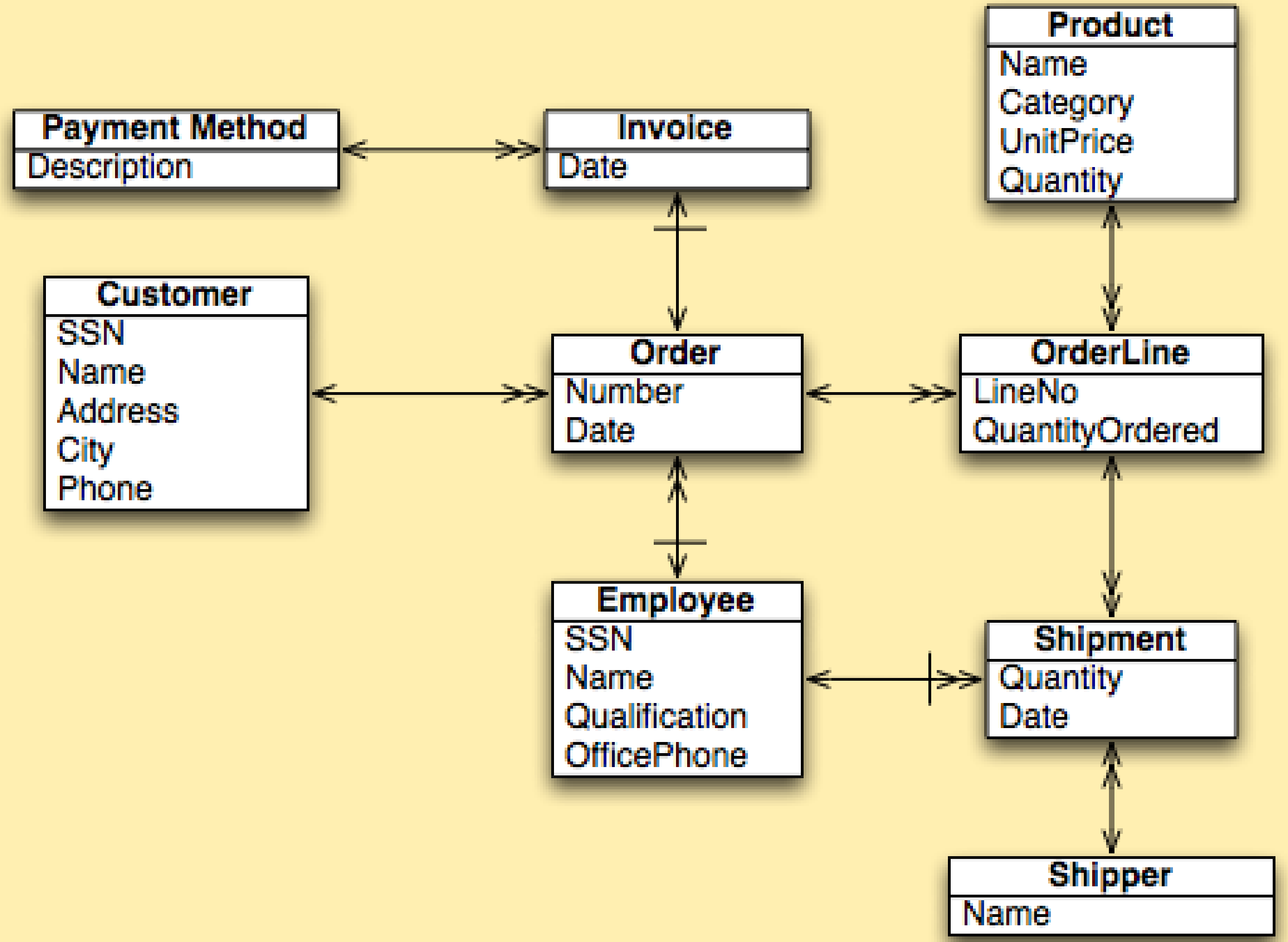
Total Due: $75

Payment must be received by July 23.

Please return a copy of this invoice with your payment.
Thank you.

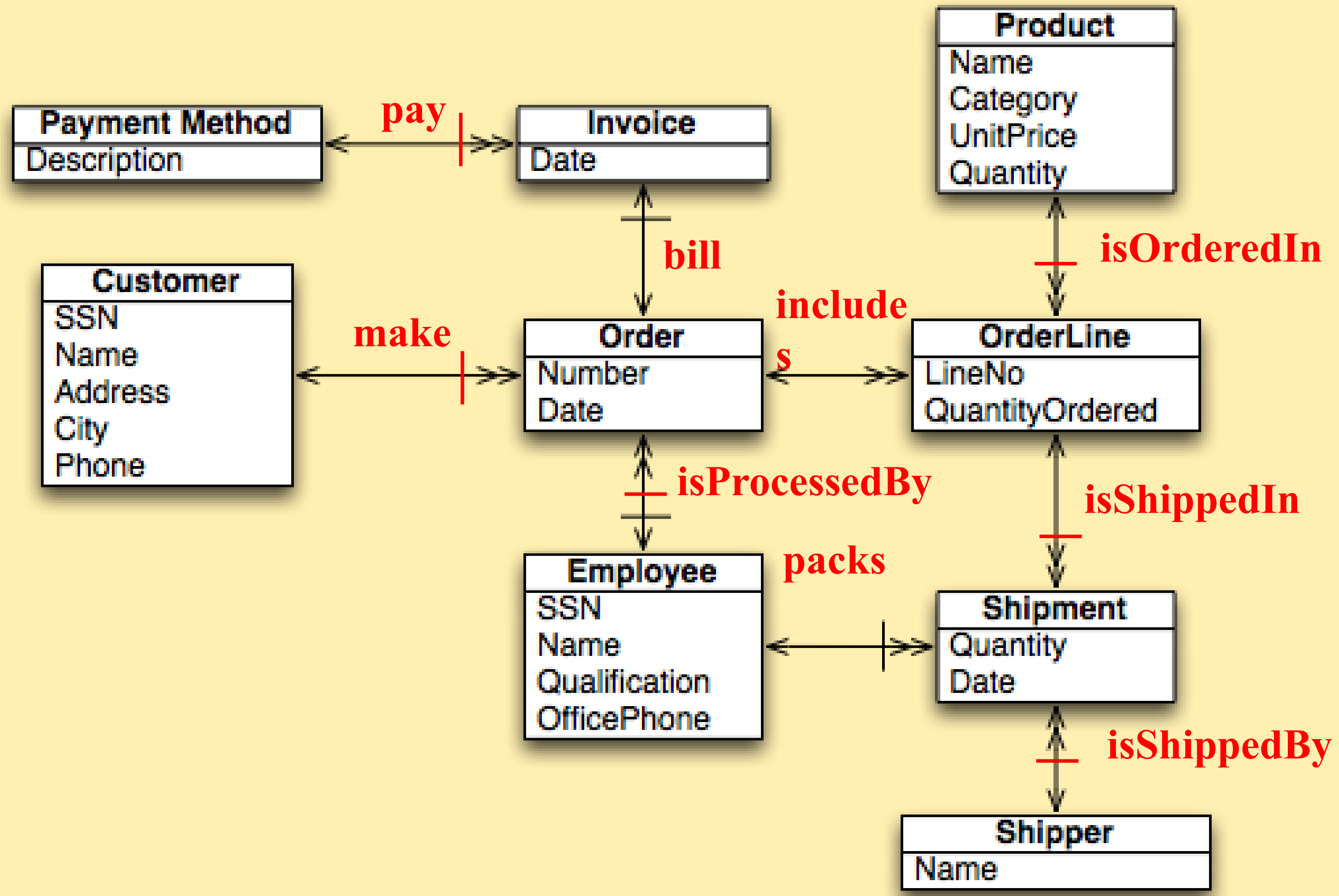
An ODM Schema for an operational DB of product orders

Exercise: assign names to relationships + check relationships



An ODM Schema for an operational DB of product orders

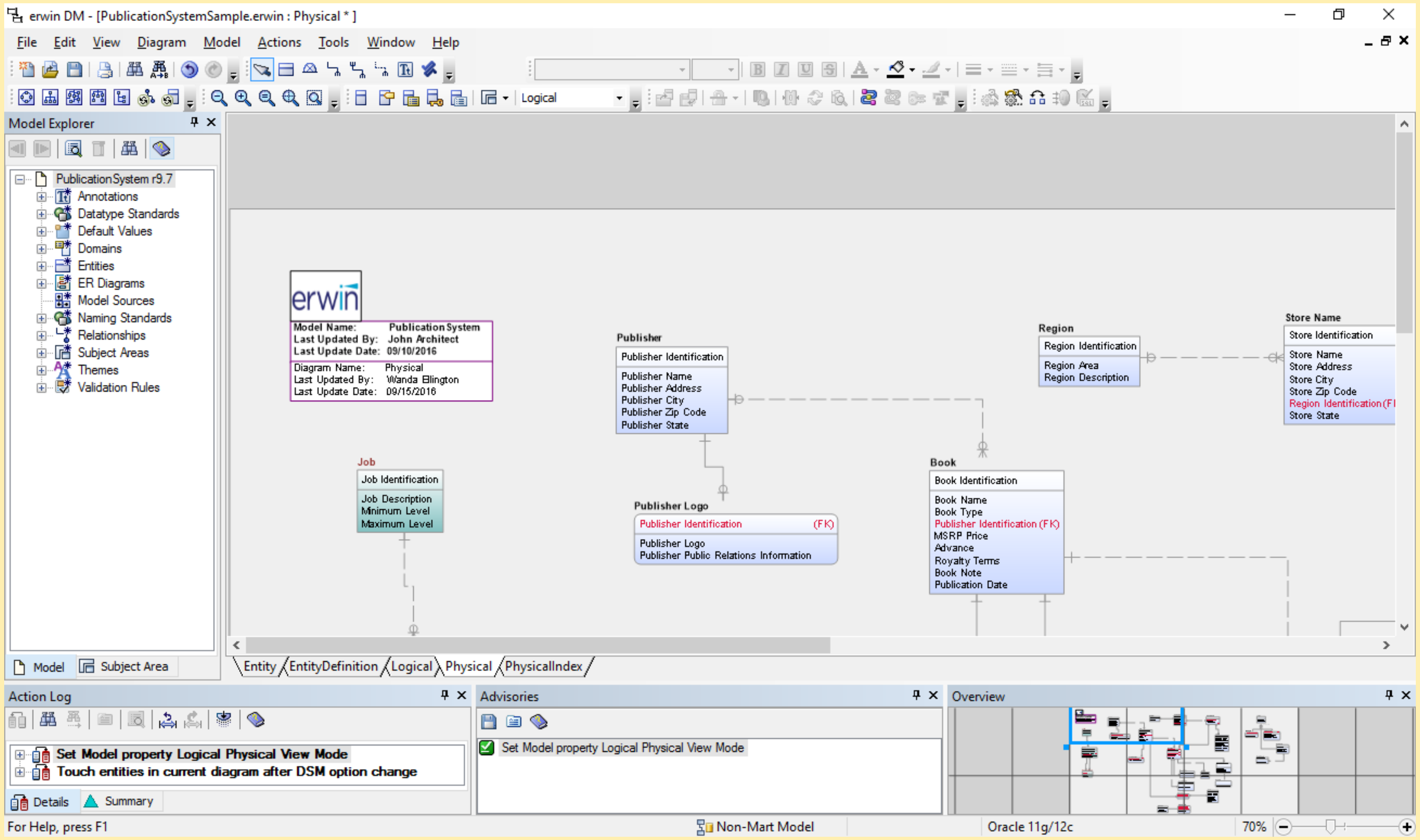
Exercise: assign names to relationships + check relationships



Data Modeling Tools:

https://dbmstools.com/categories/data-modeling-tools

Sample video



Model Name: Publication System
Last Updated By: John Architect
Last Update Date: 09/10/2016
Diagram Name: Physical
Last Updated By: Wanda Ellington
Last Update Date: 09/15/2016

Set Model property Logical Physical View Mode
Touch entities in current diagram after DSM option change

Set Model property Logical Physical View Mode