

Entity- Relationship Model

Group Q&A

Yan Yan Du Shimeng Guo Fangran
Zheng Ruxu Lin Yuxin

Content

Section	Topic
1	Relational model
2	ERM theory
3	Process
4	Object orientation
5	Hybrid modeling
6	Review

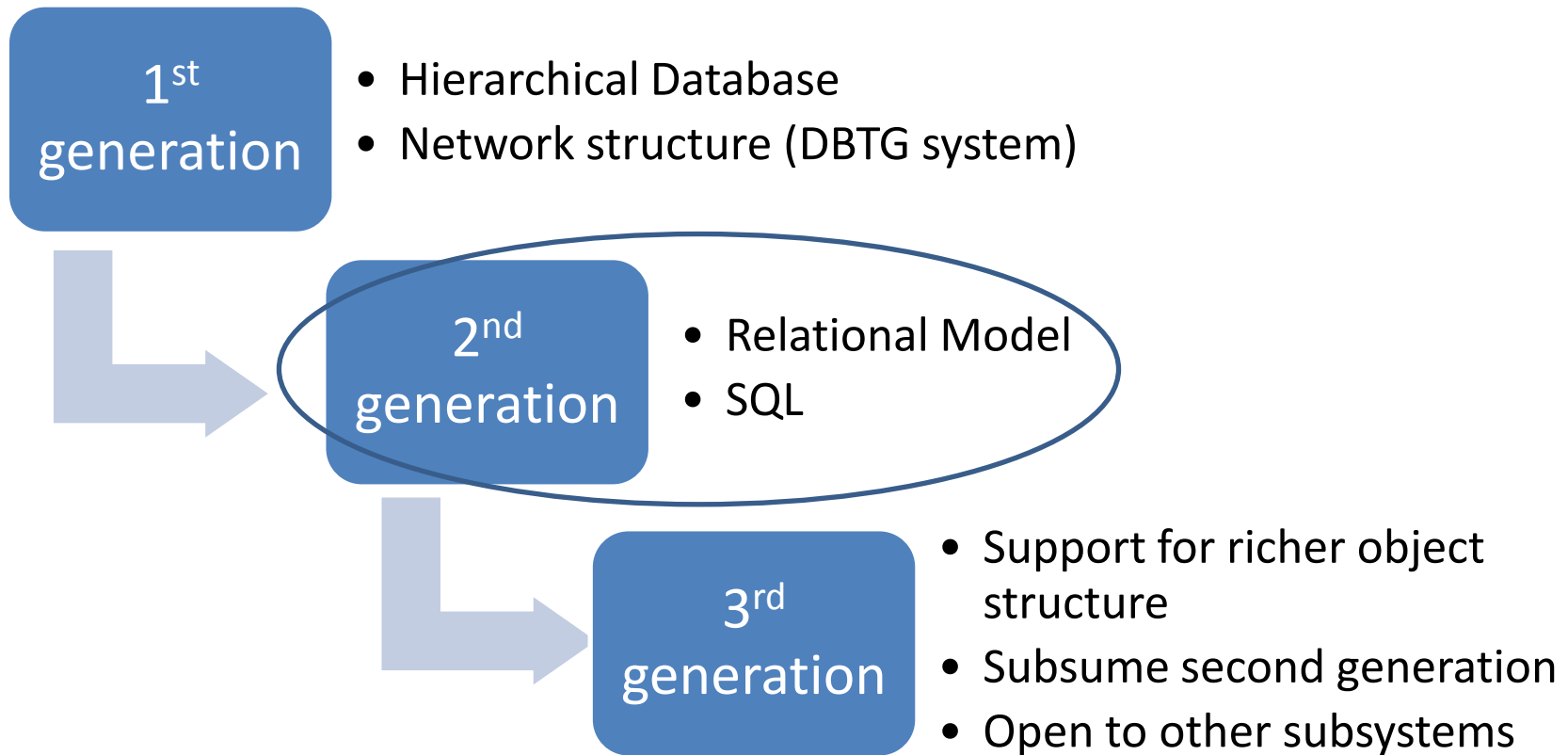
Relational Model: Review

- Definition: a subset of $D_1 \times D_2 \times \dots \times D_n$
- Terminology: attribute, tuple, etc.
- Operation Set: union, selection, etc.

Content

Section	Topic
1	Relational model
2	ERM theory
3	Process
4	Object orientation
5	Hybrid modeling
6	Review

ERM theory: Development of DBMS



ERM theory: Comparison

E.F.Codd



Peter Chen

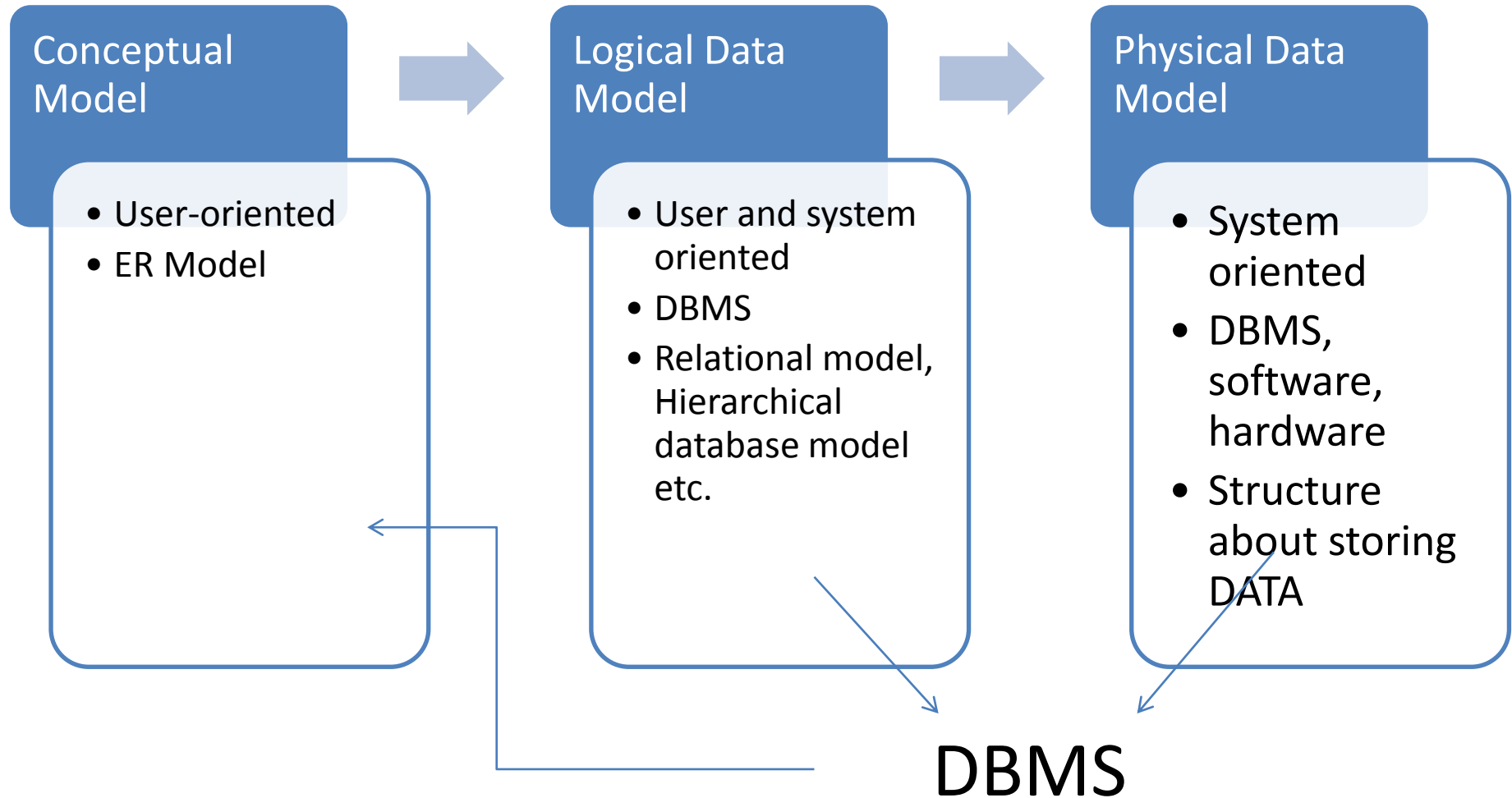


ERM theory: History

- Peter Chen (1976)
- Later improvement — Extended ER Model



ERM theory: Conceptual Model



ERM theory: Introduction

- Entity and entity set
- Relationship and relationship set
- Attribute and keys
- Mapping cardinality
- Relation, entity and relationship
- Model legend

Content

Section	Topic
1	Relational model
2	ERM theory
3	Process
4	Object orientation
5	Hybrid modeling
6	Review

E-R Design Decisions

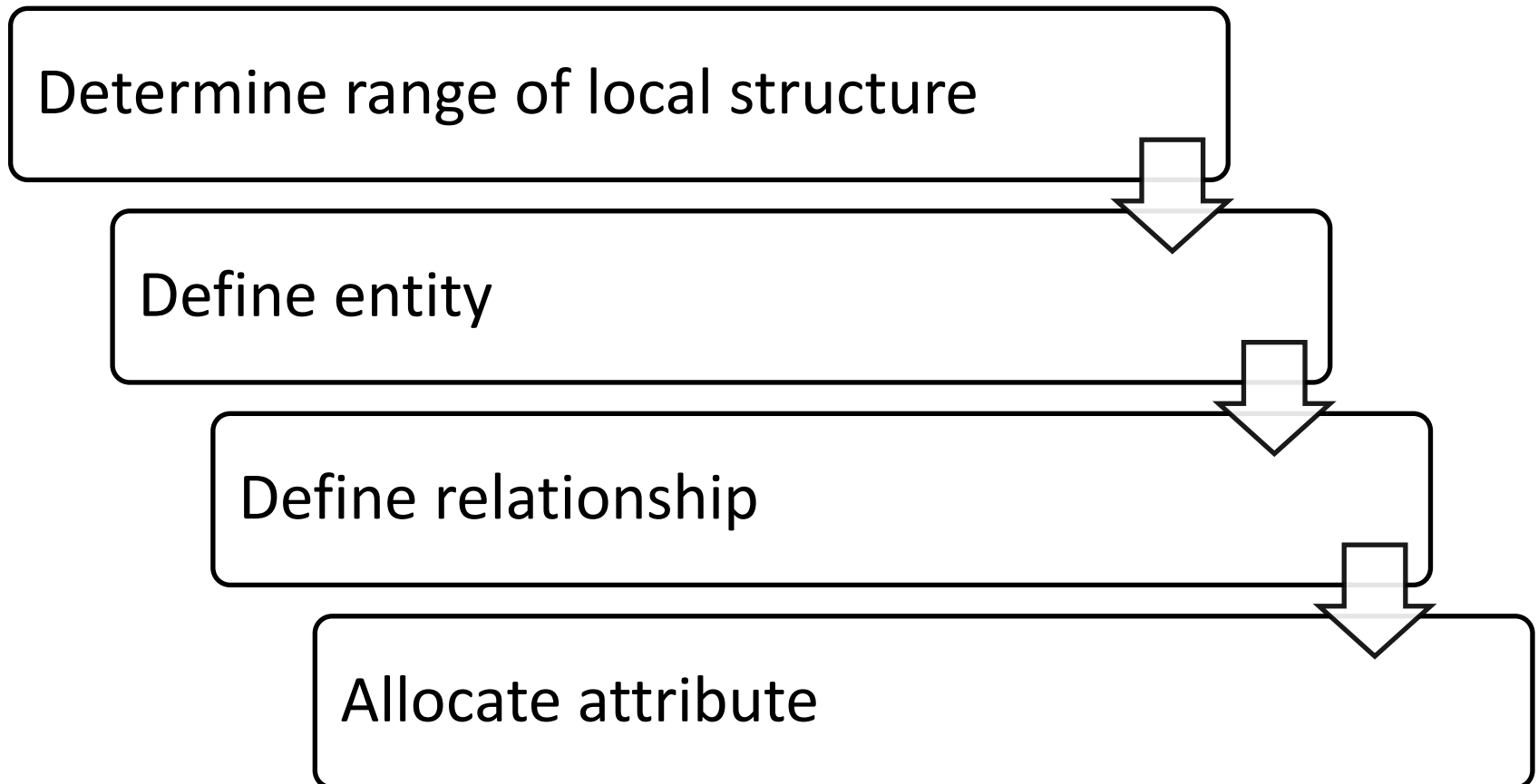
- Design philosophy
- Design strategy
- Example

Design philosophy

- Relative principle
- Consistency principle
- Simple principle

Example

Partial design of E-R Model



Example

Partial design of E-R Model

a) User management

entity:

User, Class, Avatar

relationship:

user and class n:1

user and avatar 1:1

class and avatar no relationship

attribute:

User: userid, email

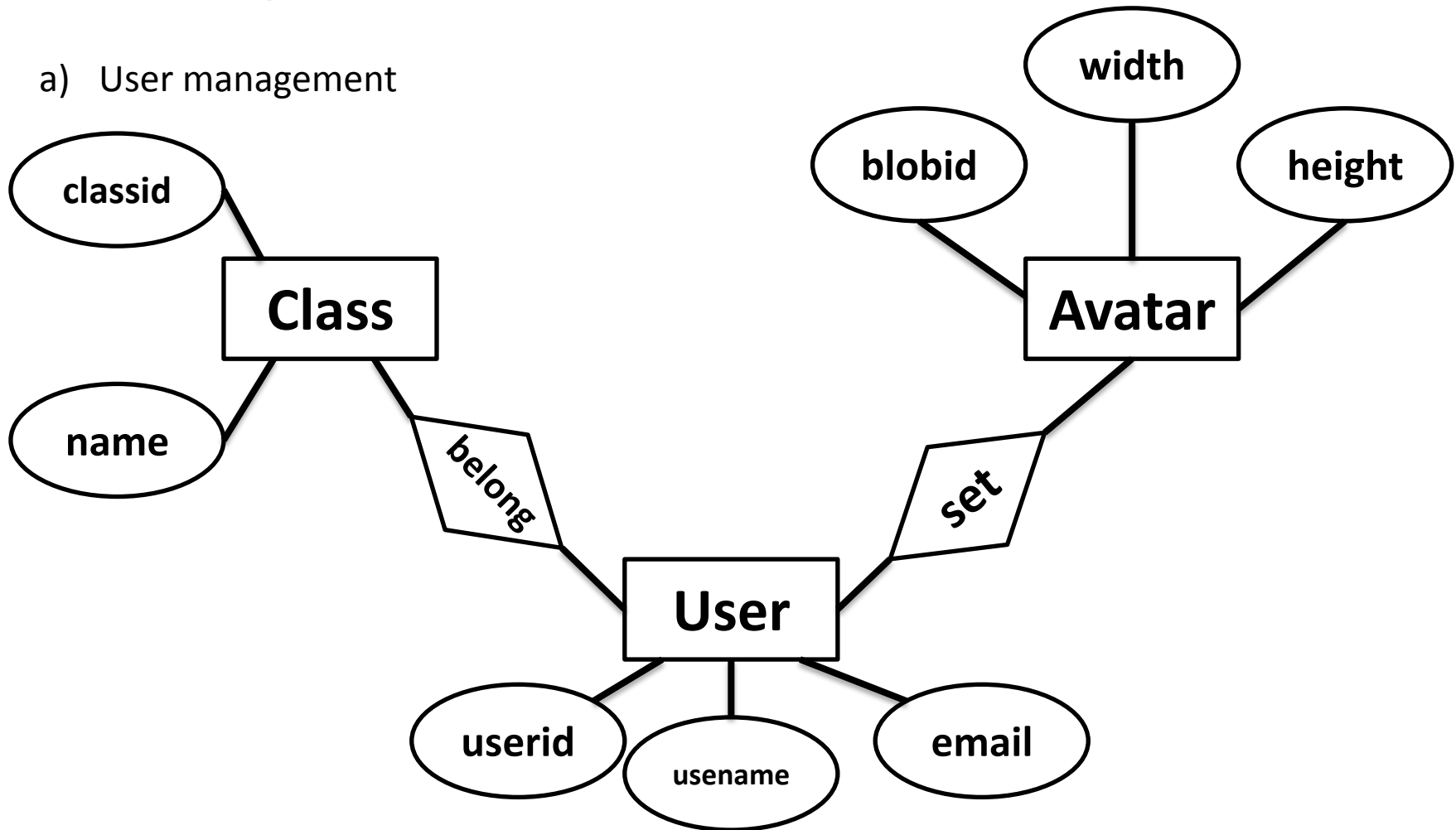
Class: class name, classid

Avatar: blobid, width, height

Example

Partial design of E-R Model

a) User management



Example

Partial design of E-R Model

b) Event management

entity:

User, event(question, answer, comment), tag

relationship:

user and event m:n

event and tag n:1

user and tag m:n

attribute:

User: userid, email

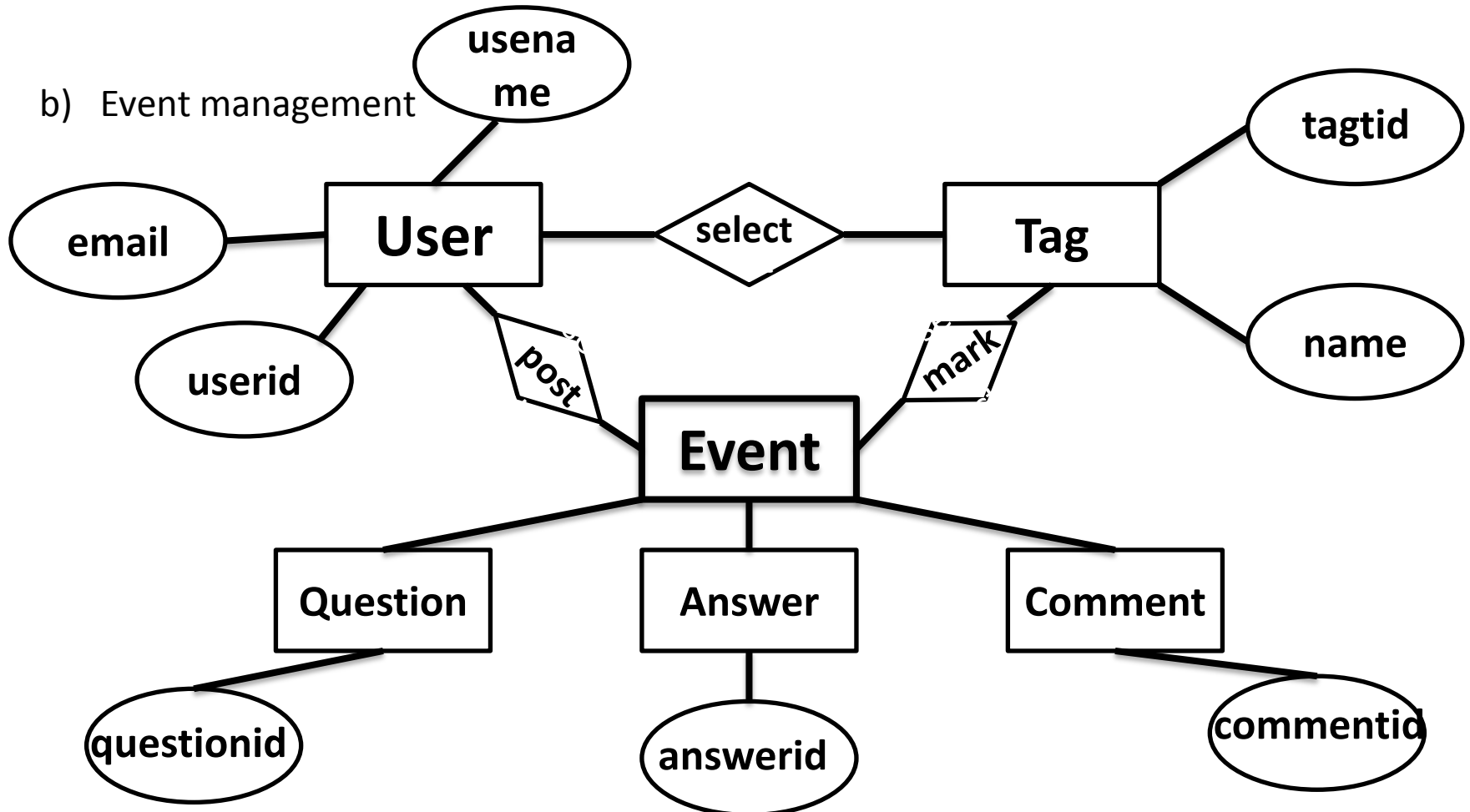
event: eventid

tag: name, tagtid

Design instance

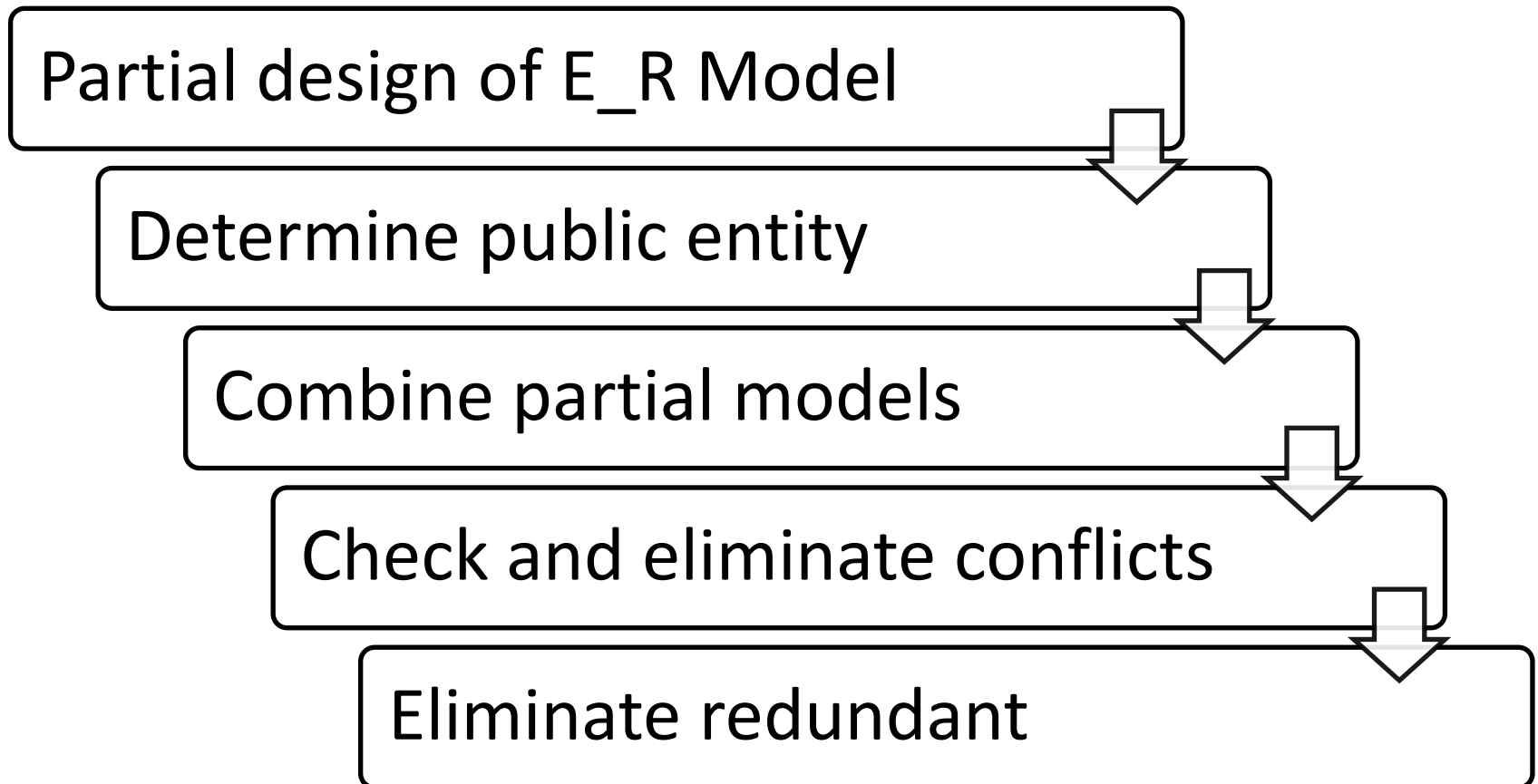
Partial design of E-R Model

b) Event management



Design instance

Global design of E-R Model



Design instance

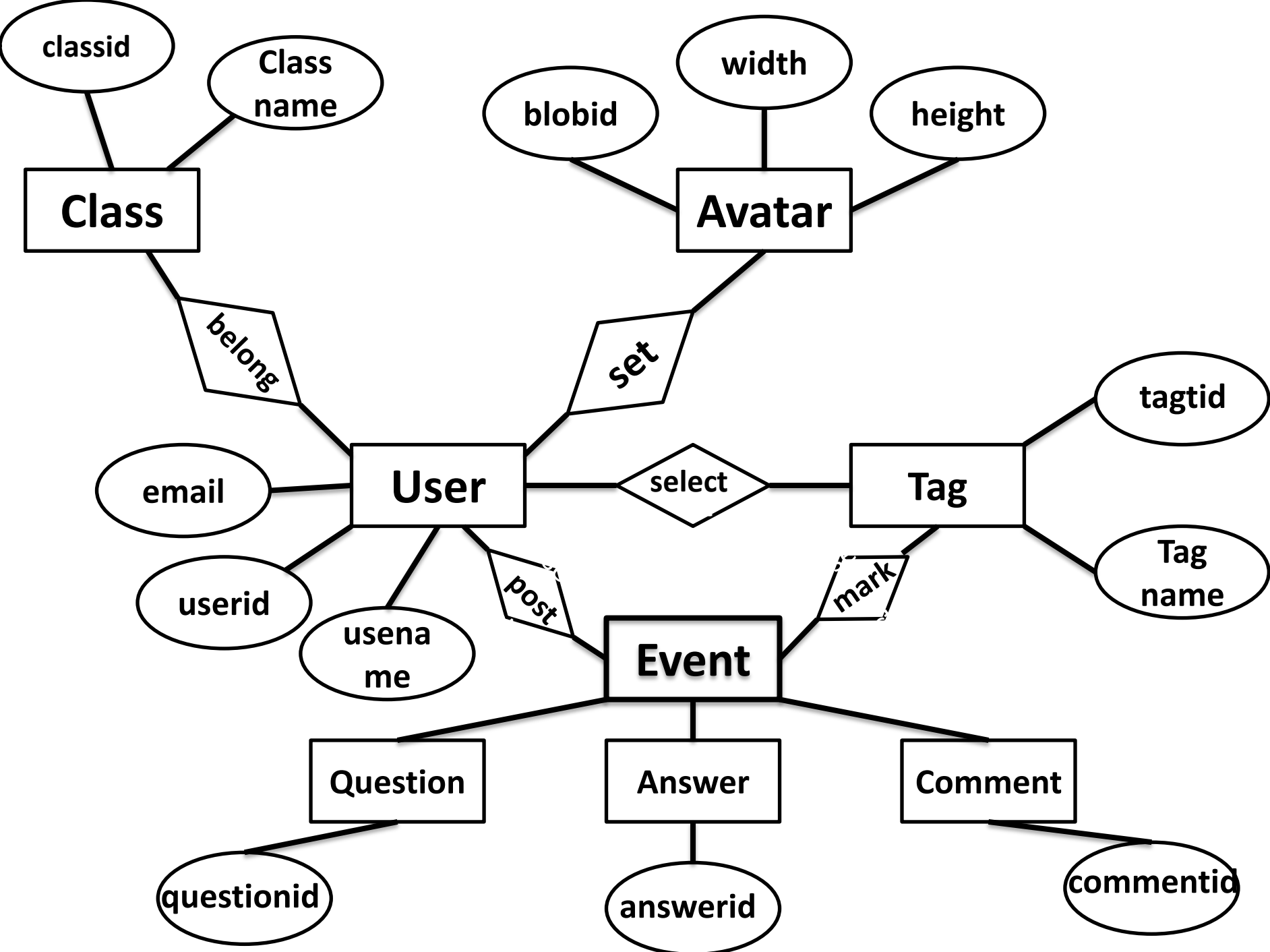
Global design of E-R Model — Eliminate conflicts

- Attribute conflict
 - a) domain conflict
 - b) unit value conflict
- Name conflict
 - a) same name, different meaning
 - b) different name, same meaning

Design instance

Global design of E-R Model — Eliminate conflicts

- structure conflict
 - a) same object, different abstract
 - b) same entity, different attribute
 - c) relationships between same entities are different in different E_R model



E-R Model to Relational Tables

Entity type → relation schema

Relationship → relation schema

Combine relation schema
with same keys

E-R Model to Relational Tables

- Entity type to relation schema

attribute of entity type -- attribute of relation

key of entity type – key of relation

name of entity type – name of relation

E-R Model to Relational Tables

- Entity type to relation schema

User (userid, username, email)

Class (classid, classname)

Avatar (blobid, width, height)

Event (eventid)

Tag (tagid, tagname)

E-R Model to Relational Tables

- Relationship to relation schema
 - a) Attribute of relation schema
 - attribute of relationship itself
 - relevant entities' attributes of relationship
 - b) Key of relation schema
 - if R is 1:1, keys of all relevant entities can be the candidate key of the relation schema;
 - if R is 1:n, key of relation schema equals key of n-side entity;
 - if R is n:m, key of relation schema is union of both entities' key

E-R Model to Relational Tables

- Relationship to relation schema

Belong	(<u>userid</u> , classid)	n:1 relationship
Set	(userid, blobid)	1:1 relationship, both can be the primary key
Post	(<u>userid</u> , <u>eventid</u>)	n:m relationship
Mark	(<u>eventid</u> , tagid)	n:1 relationship
Select	(<u>userid</u> , <u>tagid</u>)	n:m relationship

E-R Model to Relational Tables

- Combine relation schema with same keys

User (userid, username, email)

Belong (userid, classid)

Set (userid, blobid)

User (userid, username, email, classid, blobid)

Content

Section	Topic
1	Relational model
2	ERM theory
3	Process
4	Object orientation
5	Hybrid modeling
6	Review

Object orientation

- What
- Why
- How
- Object-Oriented Model

Object orientation: What

- A point of view
- Basic point
- Concepts
- Features

Object Orientation: Why

- Characteristics of OO
- 唯一性
- 分类性/抽象性
- 继承性
- 多态性

Object Orientation: How

- 几种面向对象的开发方法（略讲）
- Booch
- Coad
- OMT
- OOSE&UML

Object Orientation: OOM

- What is OOM
- 3 levels of OOM
- 对象模型
- 动态模型
- 功能模型

Content

Section	Topic
1	Relational model
2	ERM theory
3	Process
4	Object orientation
5	Hybrid modeling
6	Review

Hybrid modeling: Reasons

- **Multimedia Storage**



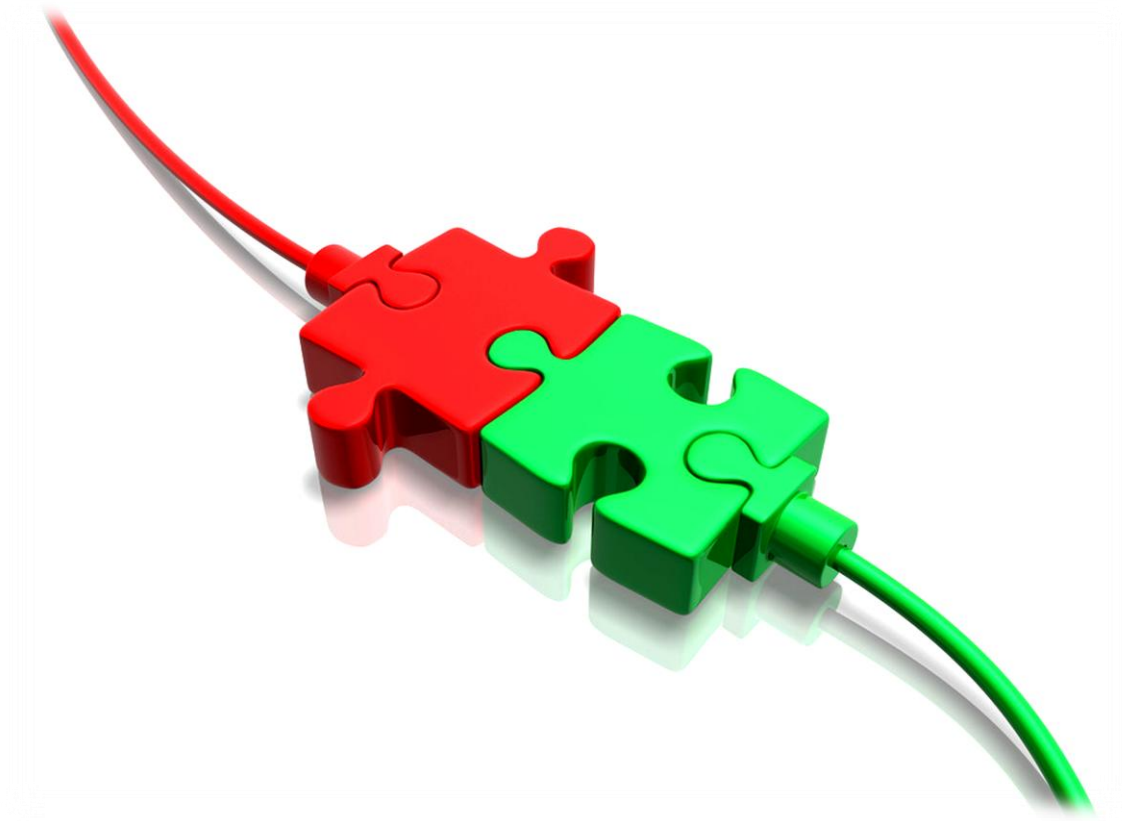
Hybrid modeling: Reasons

- **Code Compression**



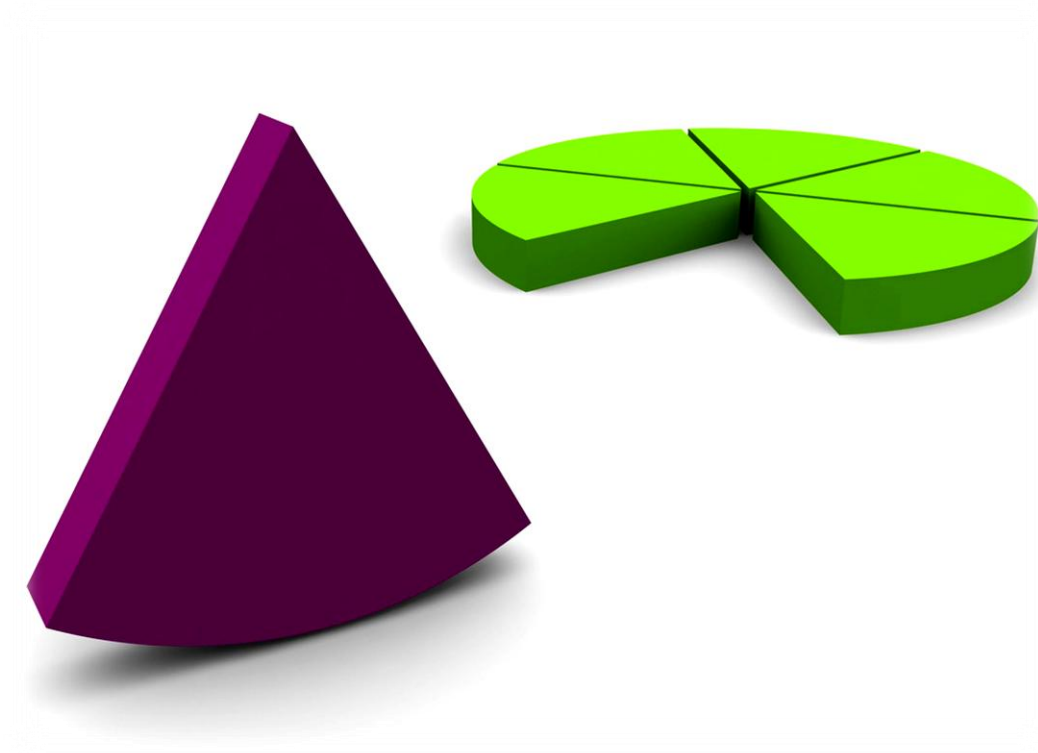
Hybrid modeling: Reasons

- **Data Relationship**



Hybrid modeling: Reasons

- **Semantic fault**



Hybrid modeling: Reasons

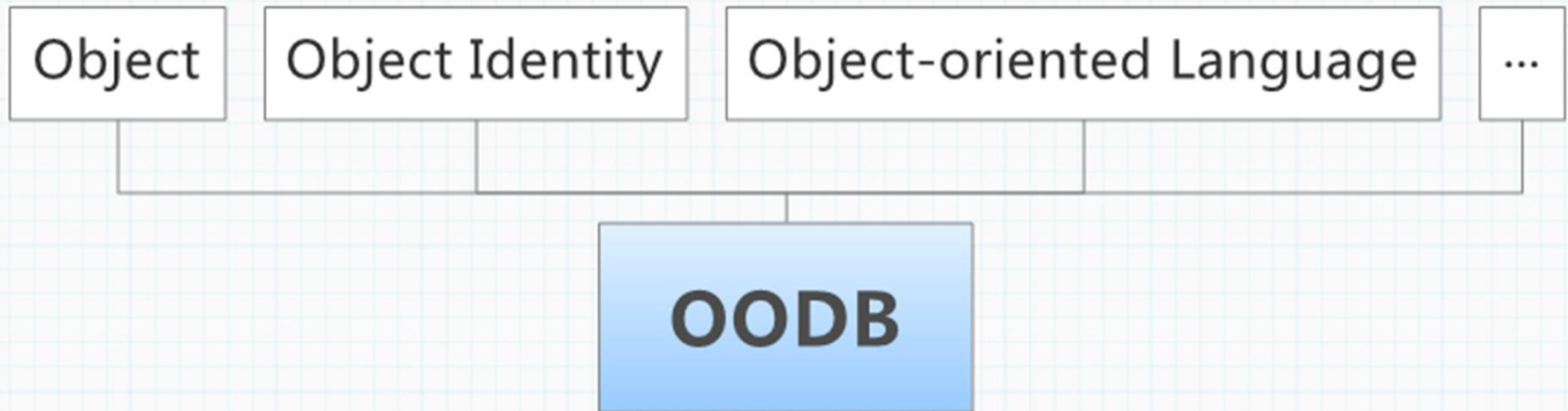
- **Automatic Detection**



Hybrid modeling: Methodology

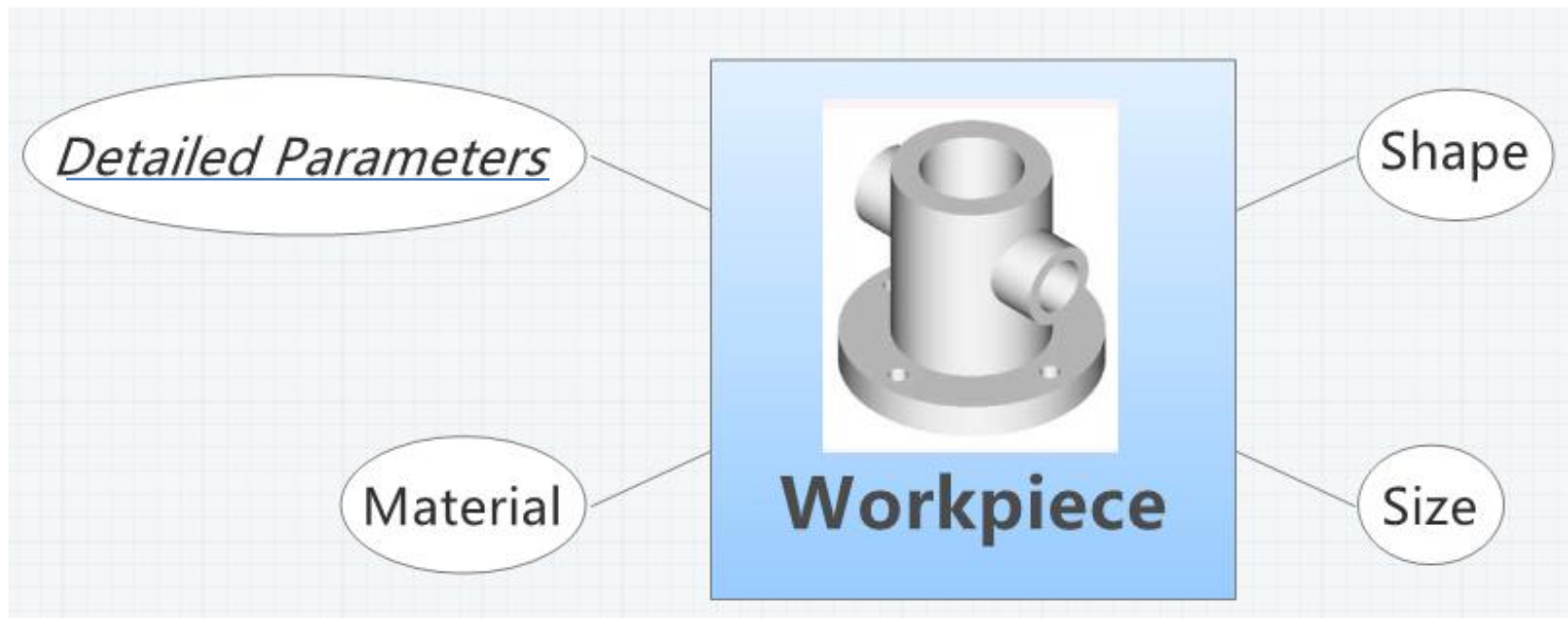
- OODB
- ORDB
- ORM

Hybrid modeling: Methodology



Hybrid modeling: Methodology

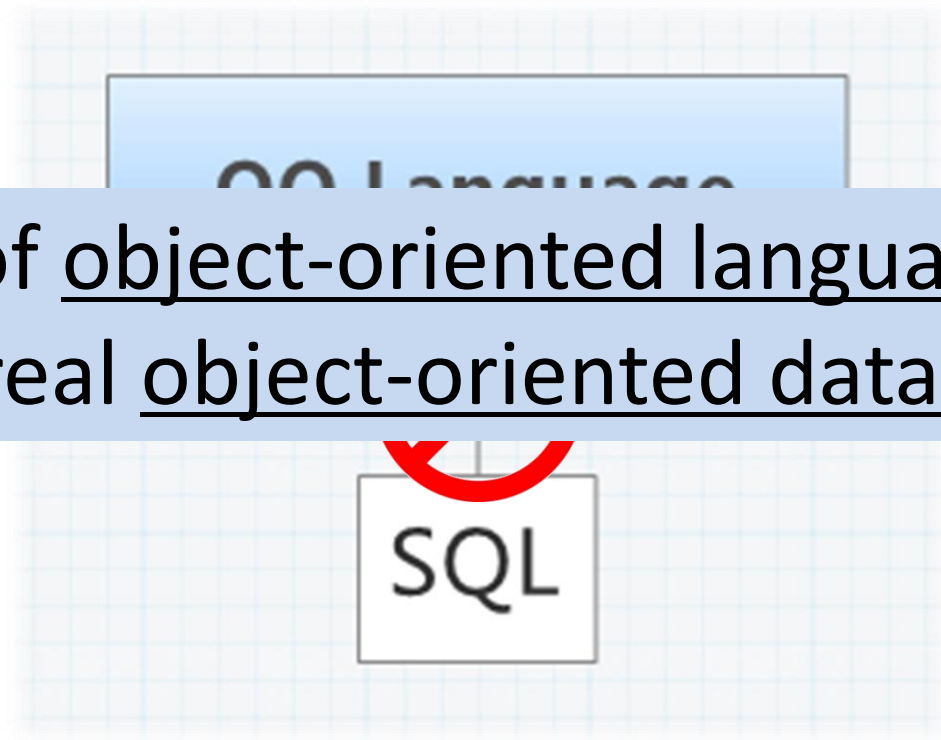
- Engineering Professions



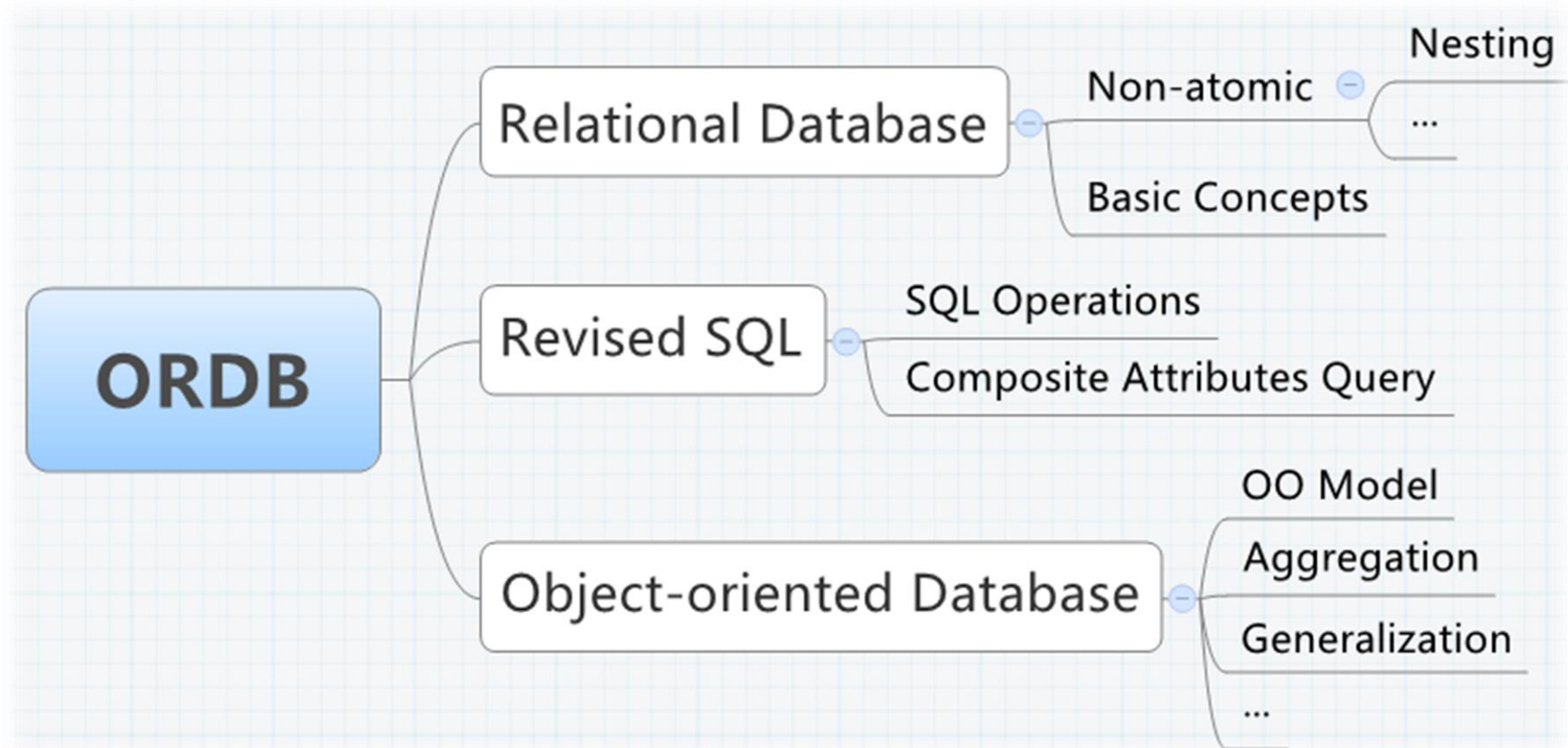
Hybrid modeling: Methodology

- Limited Modeling Language

Union of object-oriented languages rather than a real object-oriented database.



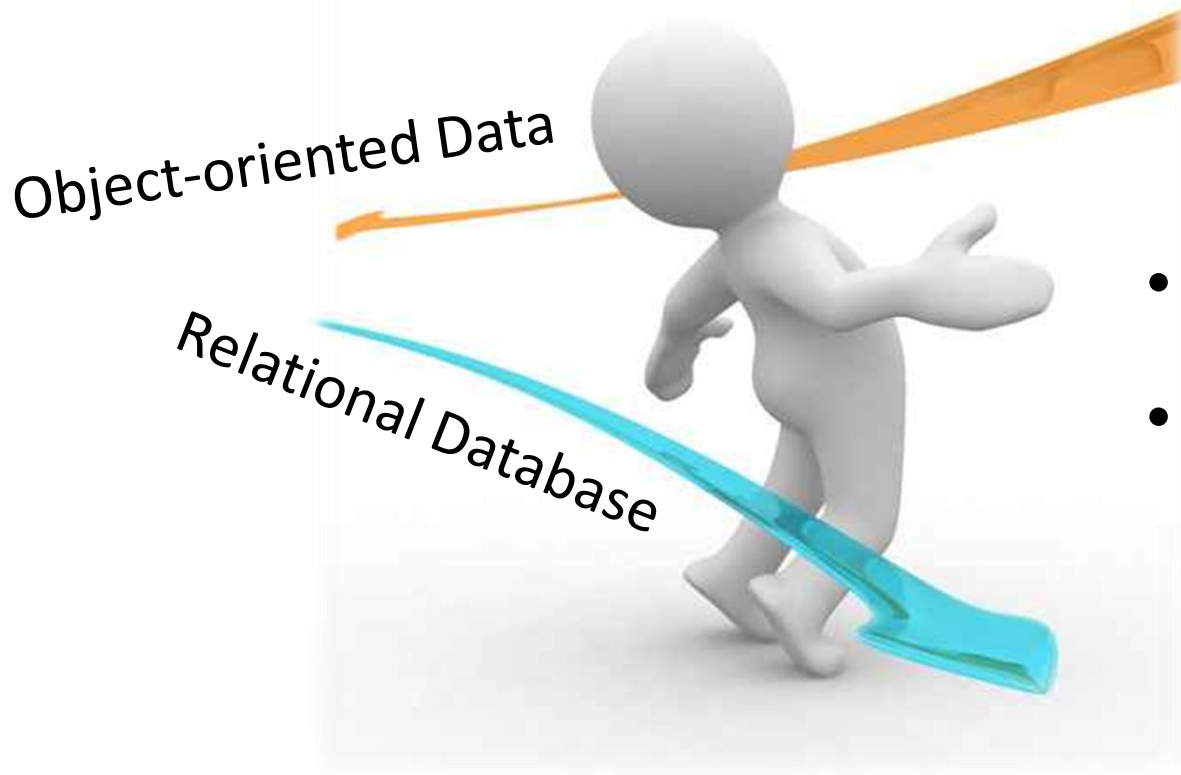
Hybrid modeling: Methodology



Hybrid modeling: Methodology

```
class qa_event_notify {  
  function process_event($event, $userid, $handle, $cookieid,  
    $params)  
  {  
    switch ($event) {  
      case 'q_post':...  
      case 'a_post':...  
      case 'c_post':...  
      ...  
      ...  
      ...  
    }  
  }  
}
```

Hybrid modeling: Methodology



ORM

- Impedance Mismatch
- Object CRUD

Hybrid modeling: Methodology

//Create a new post in the database and return its ID

```
function qa_db_post_create($type, $parentid, $userid,  
    $cookieid, $ip, $title, $content, $format, $tagstring, $notify,  
    $categoryid=null)
```

```
function qa_db_post_acount_update($questionid)
```

```
function qa_db_category_path_qcount_update($path)
```

```
function qa_db_ifcategory_qcount_update($categoryid)
```

Content

Section	Topic
1	Relational model
2	ERM theory
3	Process
4	Object orientation
5	Hybrid modeling
6	Derived design concept

Object-oriented Design: Goal

- Code compression

Object-oriented Design: Fundamental

- Object, class
- Encapsulation
- Inheritance
- Polymorphism

Object-oriented Design: Ideas

OOD addresses a bigger picture.

Ideas:

- Object oriented
- Re-usable
- Variable with minimal effort
- Extendable without change

Object-oriented Design : Why

- "Walking on water and developing software from a specification are easy if both are frozen."

- *Edward V. Berard*

Object-oriented Design : Principles

- **S** = Single Responsibility Principle
- **O** = Opened Closed Principle
- **L** = Liskov's Substitution Principle
- **I** = Interface Segregation Principle
- **D** = Dependency Inversion Principle

Slide position_6:10



SINGLE RESPONSIBILITY PRINCIPLE

Just Because You Can, Doesn't Mean You Should



OPEN CLOSED PRINCIPLE

Open Chest Surgery Is Not Needed When Putting On A Coat



LISKOV SUBSTITUTION PRINCIPLE

If It Looks Like A Duck, Quacks Like A Duck, But Needs Batteries - You Probably Have The Wrong Abstraction

Slide position_9:10



INTERFACE SEGREGATION PRINCIPLE

You Want Me To Plug This In, Where?

Slide position_10:10

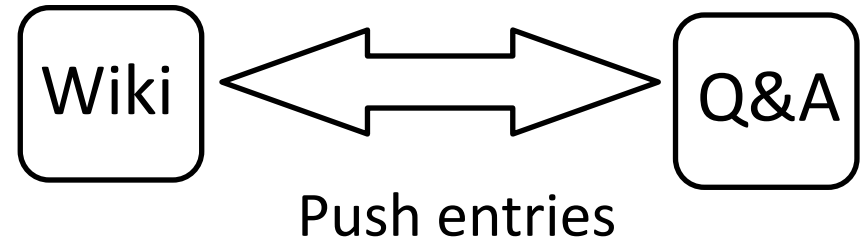


DEPENDENCY INVERSION PRINCIPLE

Would You Solder A Lamp Directly To The Electrical Wiring In A Wall?

Design by Contract

- Pre-condition
- Post-condition
- class invariant



Design patterns

- Standardized design
- OOD principles



Design patterns: Basic elements

- Pattern name
- Problem
- Solution
- Consequences

