INFORMATION SYSTEM DEPARTMENT
INFORMATION TECHNOLOGY FACULTY – HCM
UNIVERSITY OF SCIENCE

ADVANCED DATABASE

Chapter 04

CONCEPTUAL DESIGN

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

fit@hcmus

Outline

- Recall: SDLC, DBLC & Database Design
- Conceptual Design
- Design Strategies
- Database Design Cases

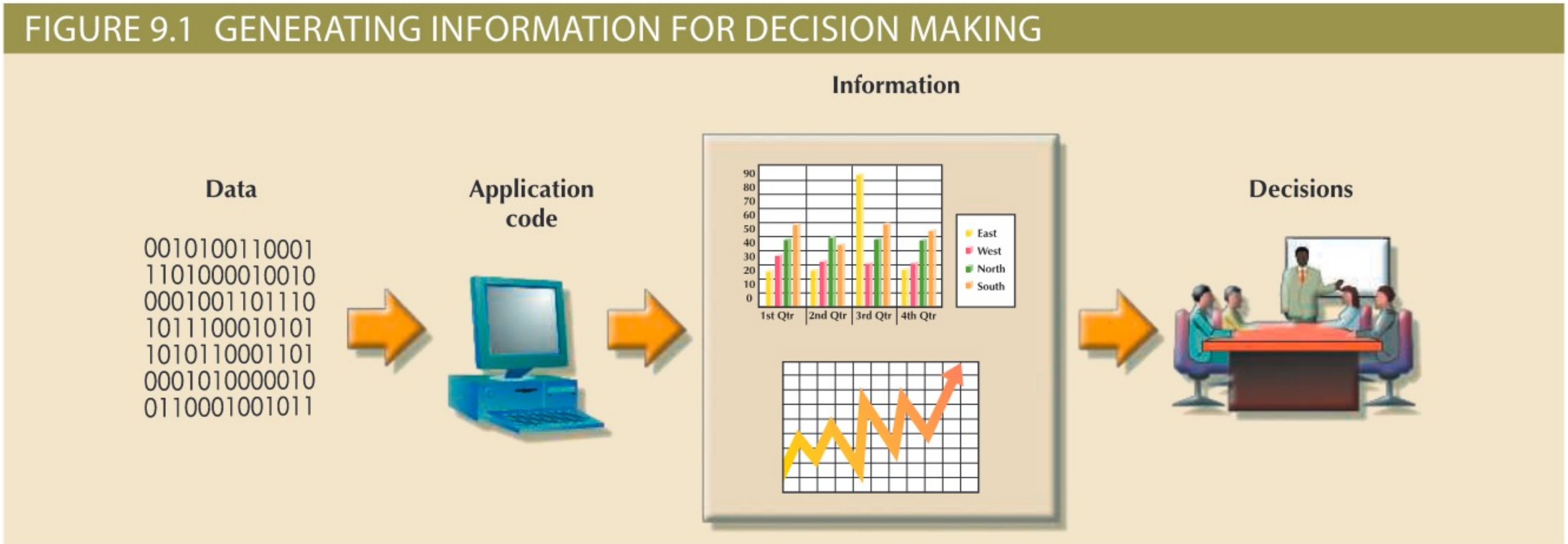
Outline

- Recall: SDLC, DBLC & Database Design
- Conceptual Design
- Design Strategies
- Database Design Cases

fit@hcmus

Information System (IS)

Data → Information → Knowledge → Insights



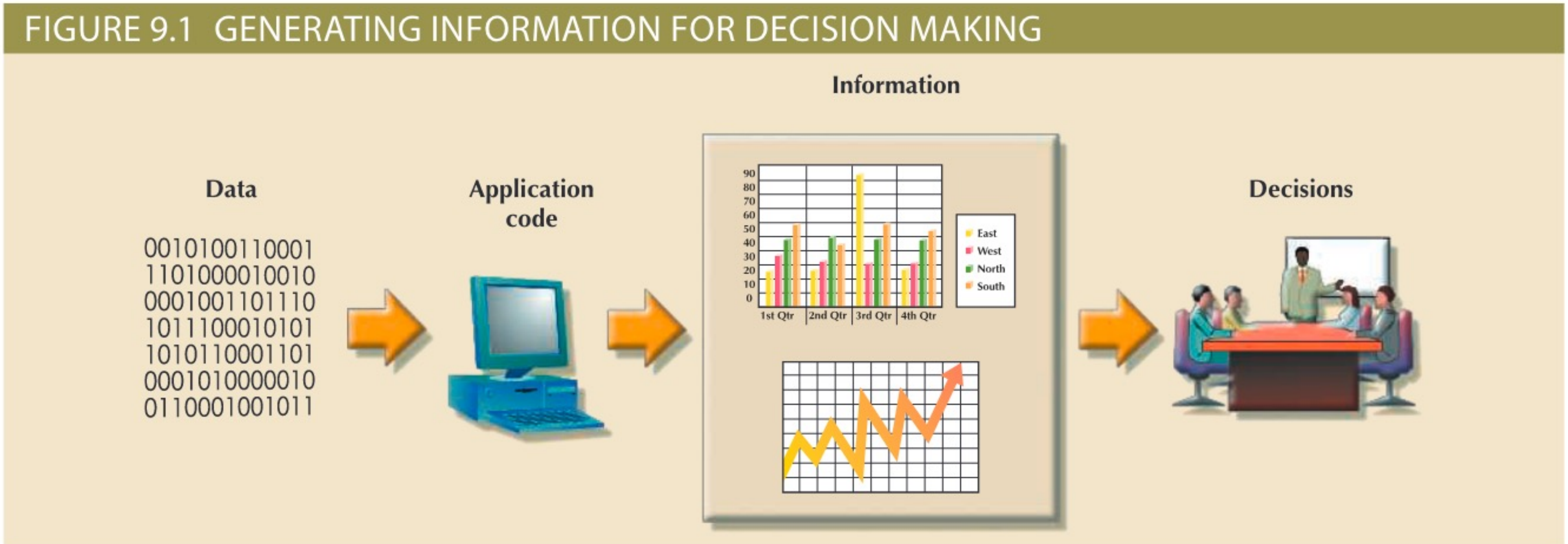
The **performance of an IS** depends on:

- **Database Design and Implementation**
- Application Design and Implementation
- Administrative Procedure

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Information System (IS)

Data → Information → Knowledge → Insights



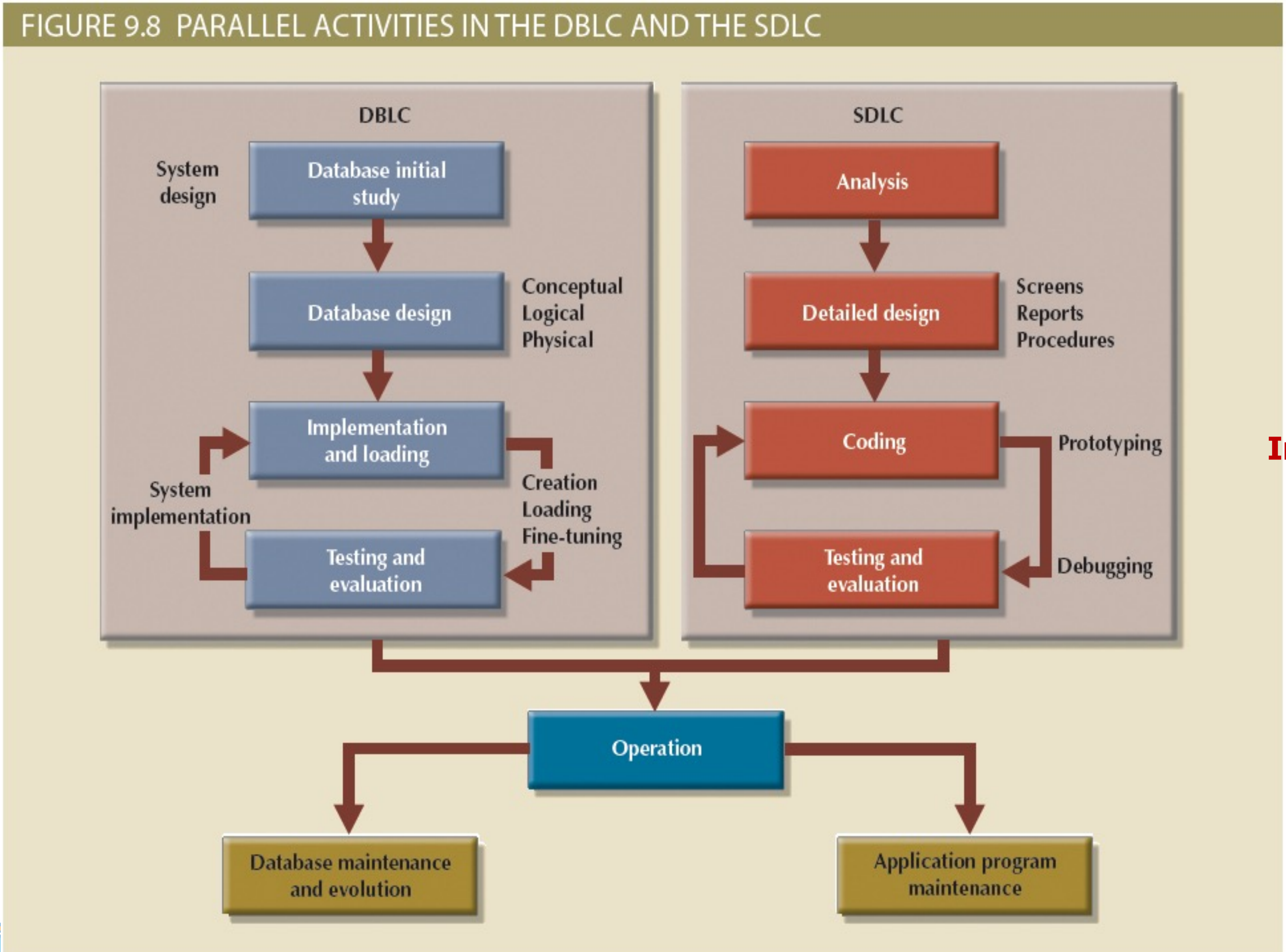
Development Process for an **IS**:

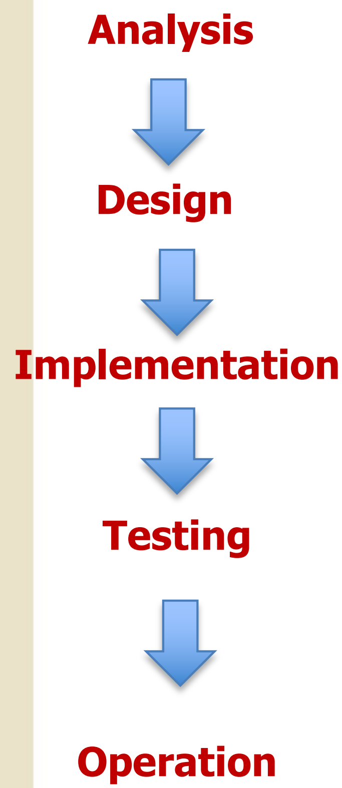
- System Development Life Cycle (**SDLC**)
- **Database Development Life Cycle (DBLC)**

(Coronel and Morris, 2015) (Hoffer et al., 2019)

cdio 4.0
fit@hcmus

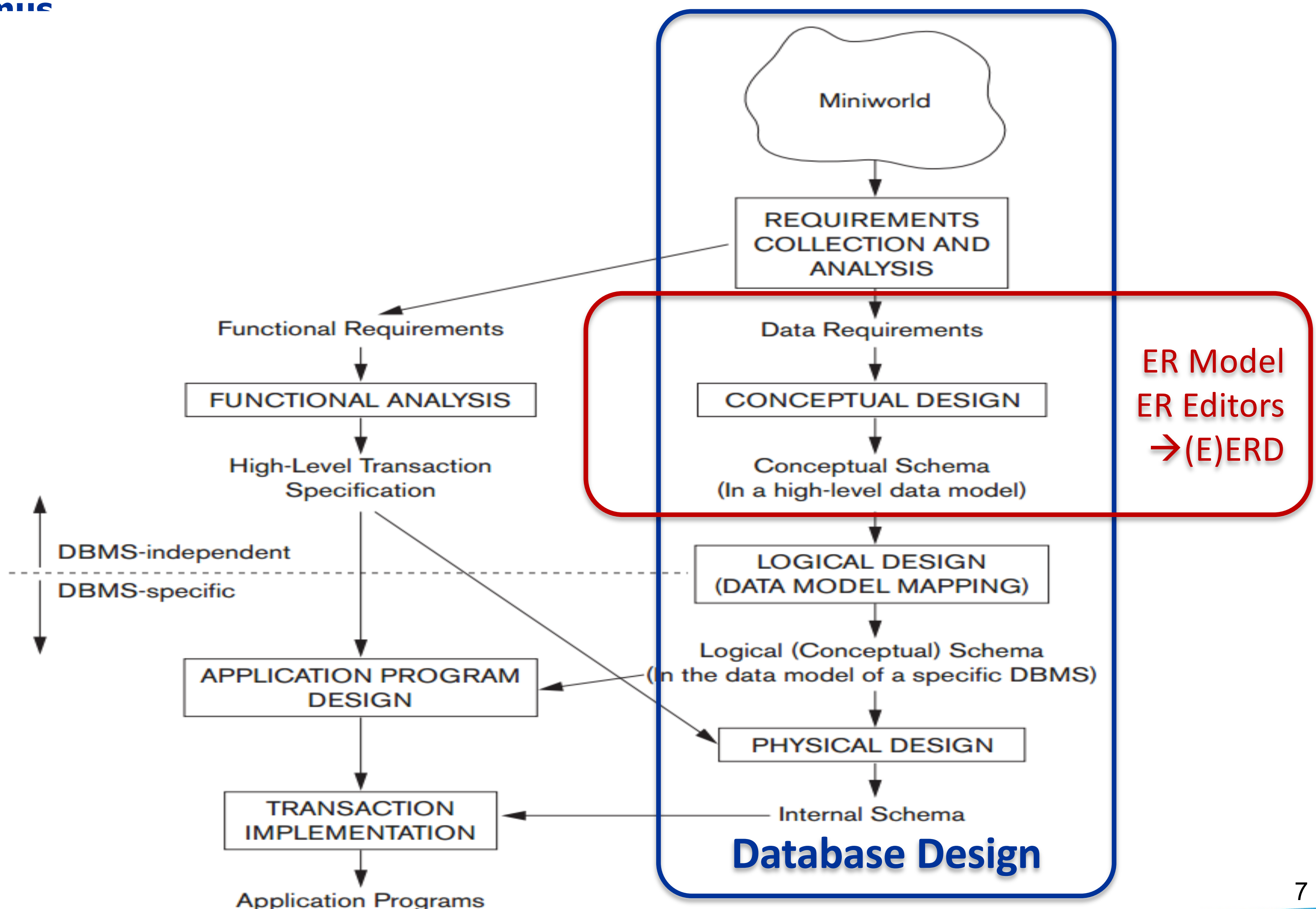
SDLC & DBLC





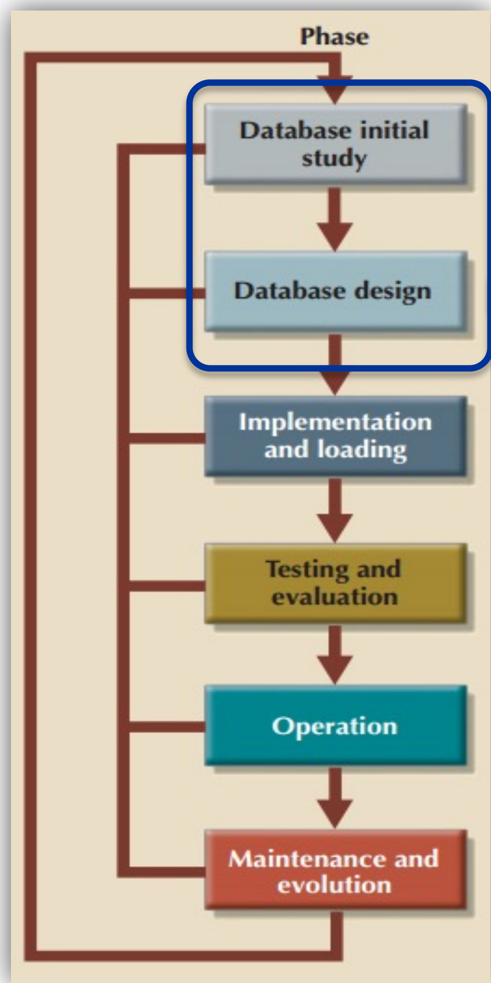
(Coronel and Morris, 2015) (Hoffer et al., 2019)

SDLC & DBLC



(Coronel and Morris, 2015) (Hoffer et al., 2019)

DBLC & Database Design



Goal: Design a data model to support company operations and objectives.

Points to keep in mind:

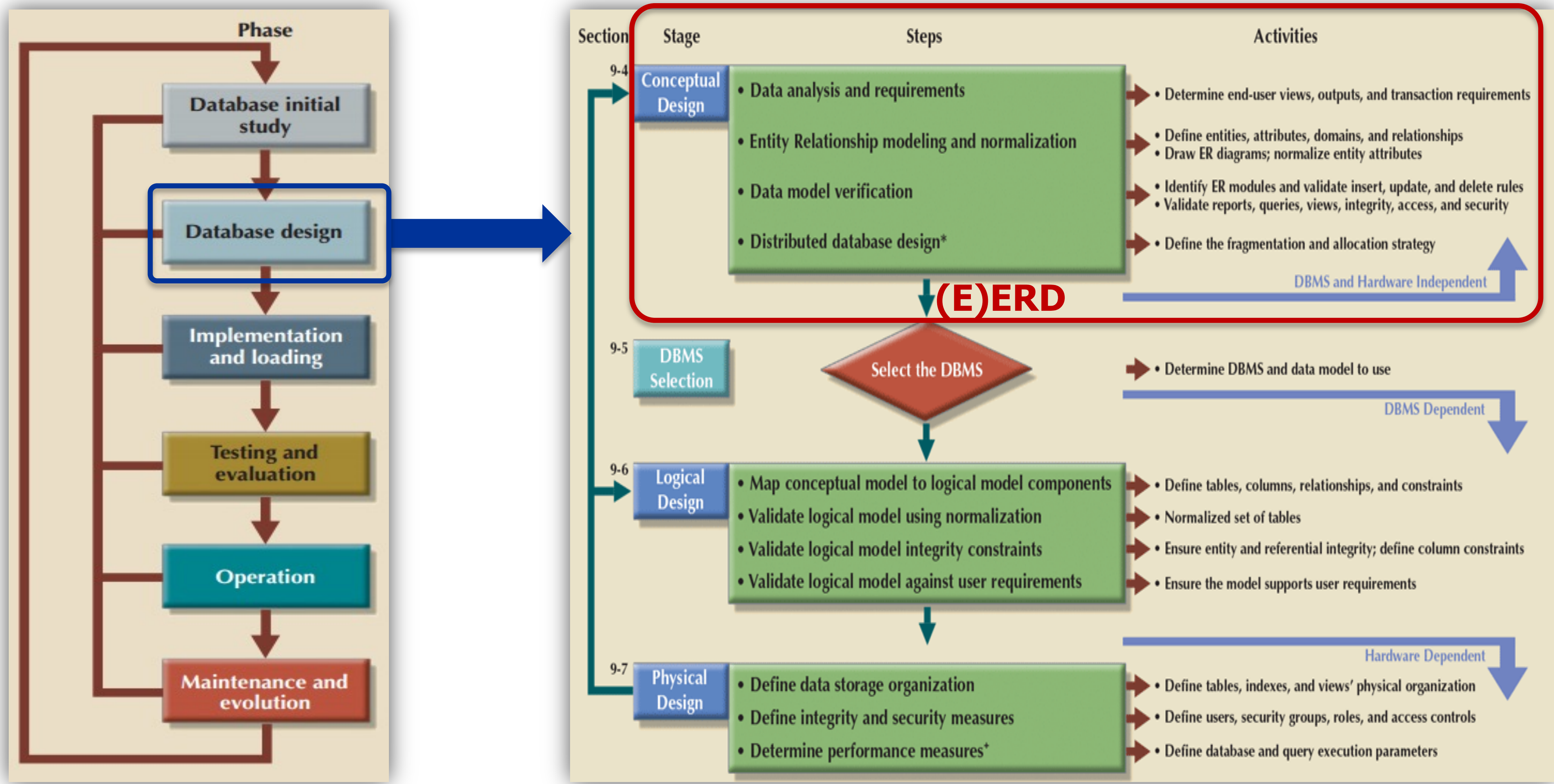
- ✓ Focus on ***data characteristics*** according to two views of the data (business and designer views)
- ✓ Data is ***only one element*** of a larger IS.
- ✓ The use of ***data across user views***.
- ✓ *Makes sure* ***final product meets requirements***

Steps:

- ✓ **Conceptual design**
- ✓ **Logical design**
  - ✓ **Selection of DBMS** (a critical step)
- ✓ **Physical design**

(Coronel and Morris, 2015) (Hoffer et al., 2019)

DBLC & Database Design



(Coronel and Morris, 2015) (Hoffer et al., 2019)

cdio 4.0
fit@hcmus

Outline

- Recall: SDLC, DBLC & Database Design
- Conceptual Design
- Design Strategies
- Database Design Cases

Conceptual Design

- **Goal:** Build a conceptual model (or an abstract database schema) to represent objects of a given problem domain (entities, attributes, relationships, and constraints)
- **Definition**: The process of constructing a data model used in an enterprise, independent of all physical considerations using the information documented in the users' requirements specification.
- **Output**: A conceptual model, which is a **source** of information for the **logical database design**.
- **Tools**: Data models (ER), ER editors (Visio, …)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

- **Criteria:**
  - Software- and hardware- independent
  - Provide a clear comprehension of the business and its functional areas
  - The collection of data becomes meaningful only when business rules are defined.
  - **Minimum Data Rule**:

All that is needed is there, and all that is there is needed.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Recall: Business rules

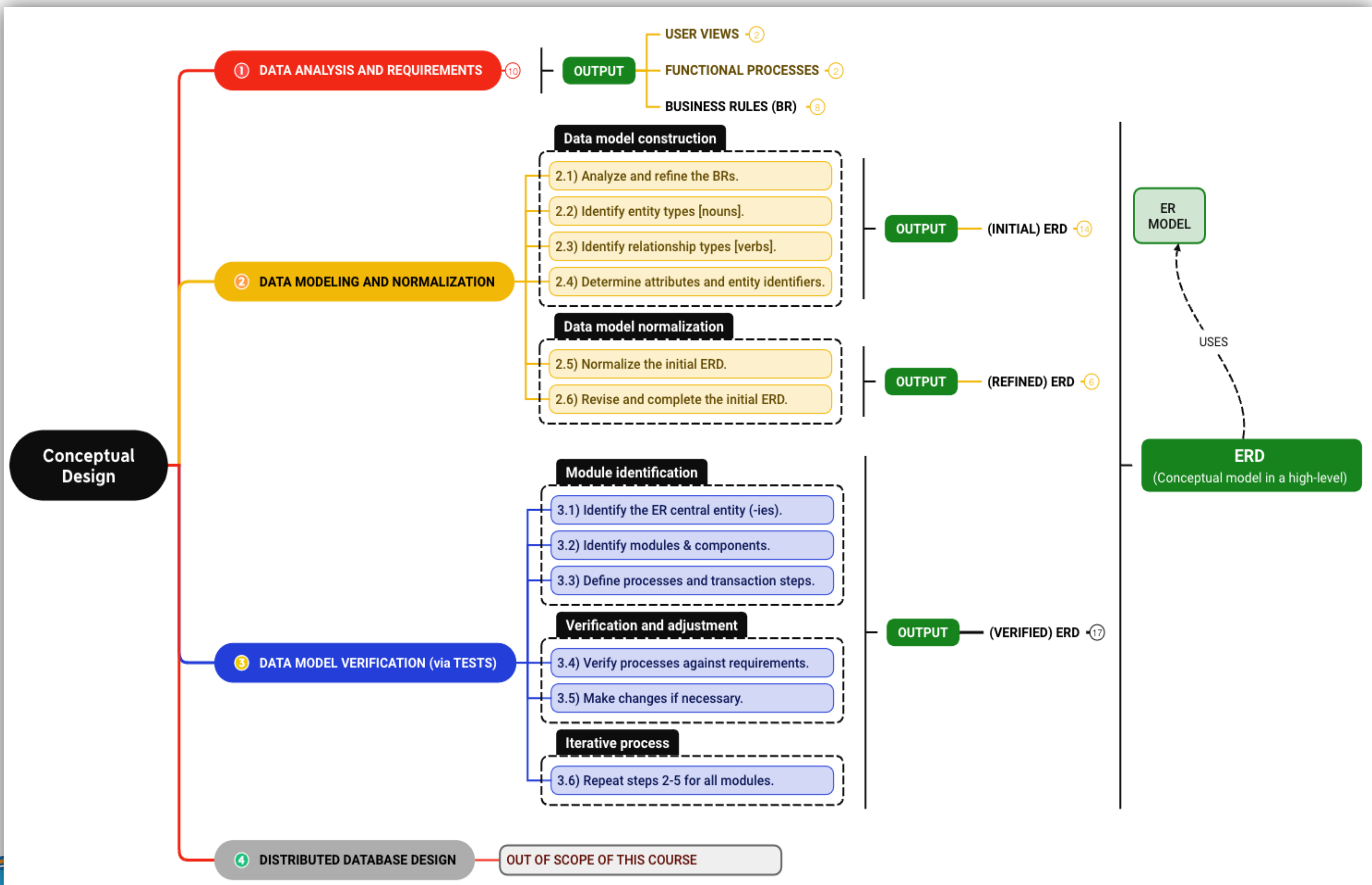
- Brief and precise **description of a policy, procedure, or principle** within a specific organization's environment
- Business rules, derived from a detailed **description of an organization's operations**

Example

- A customer may make many payments on an account.
- Each payment on an account is credited to only one customer.
- A customer may generate many invoices.
- Each invoice is generated by only one customer

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design Process



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Conceptual Design

Step 1: Data analysis and requirements

Only appropriate data element characteristics can be transformed into appropriate information.

- **Objective**: Discover characteristics of data elements.
- **Participants**: Database Designers
- **Approaches**: Utilize techniques such as interviews, questionnaires, and observations.
- **Outcome**: A **concise written documentation** of users' requirements.

STEP	ACTIVITY
1	Data analysis and requirements
2	Entity relationship modeling and normalization
3	Data model verification
4	Distributed database design

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 1: Data analysis and requirements

What should designers concentrate on?

- **Information needs**
  - *What kind of information is needed?*
  - *What output (reports and queries) must be generated?*
  - *What information does the current system generate?*
- **Information users**
  - *Who will use the information?*
  - *How is the information to be used?*
  - *What are the various end-user data views?*

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 1: Data analysis and requirements

What should designers concentrate on?

- **Information sources**
  - *Where is the information to be found?*
  - *How is the information to be extracted once it is found?*
- **Information constitution**
  - *What data elements are needed to produce the information?*
  - *What are the data attributes?*
  - *What relationships exist in the data?*
  - *What is the data volume?*
  - *How frequently is the data used?*
  - *What data transformations will be used to generate the required information?*

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 1: Data analysis and requirements

Where & How to find the required information?

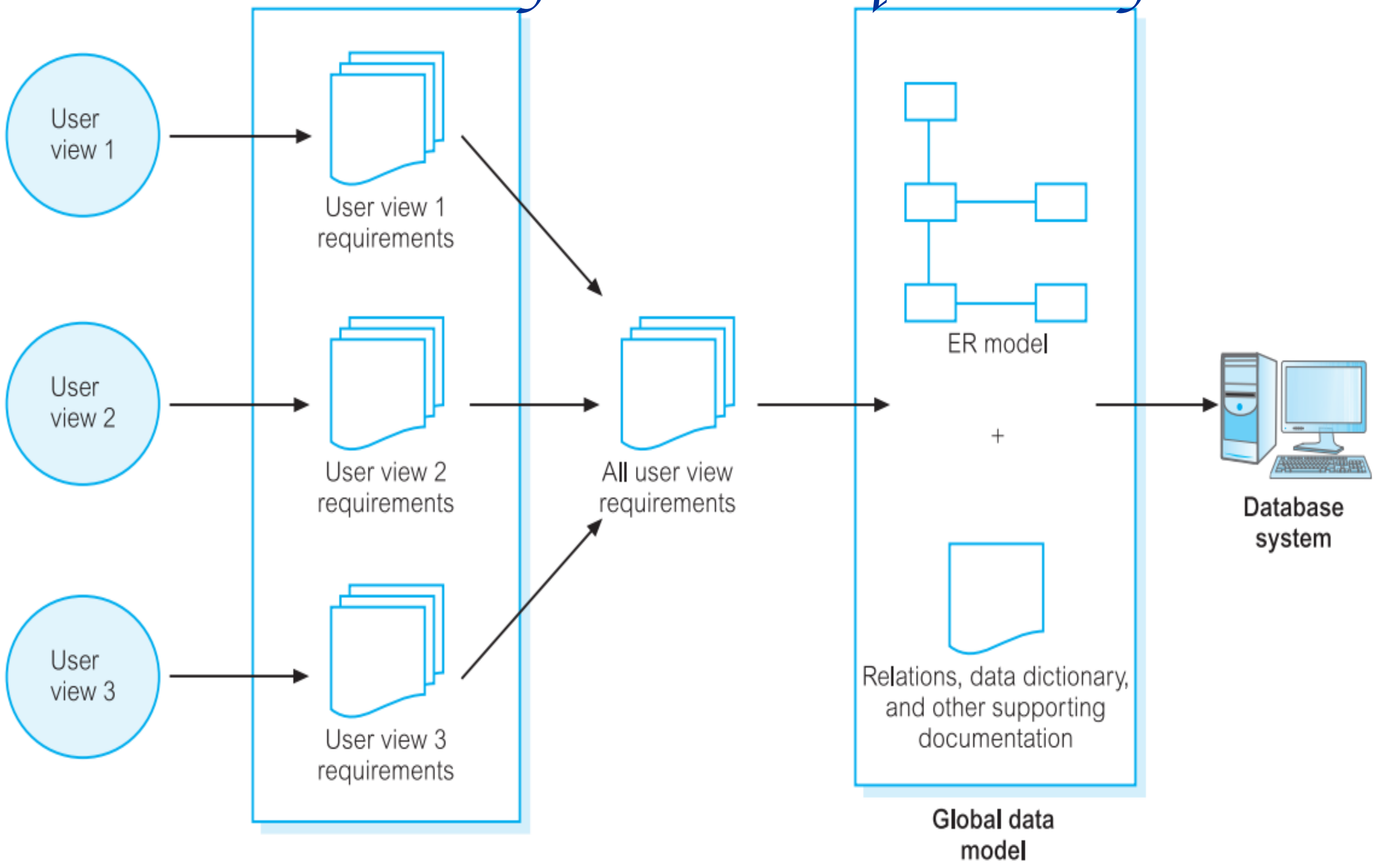
- **Develop and gather end-user data views**
  - Centralized approach: Collate requirements from different user views into a single list of requirements.
  - View integration approach: Maintain separate requirements for each user view.
  - Hybrid: Combine the two above approaches.
- Directly observe the current system
- Interface with system design groups

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 1: Data analysis and requirements

Where & How to find the required information?



Gather end-user data views: The centralized approach

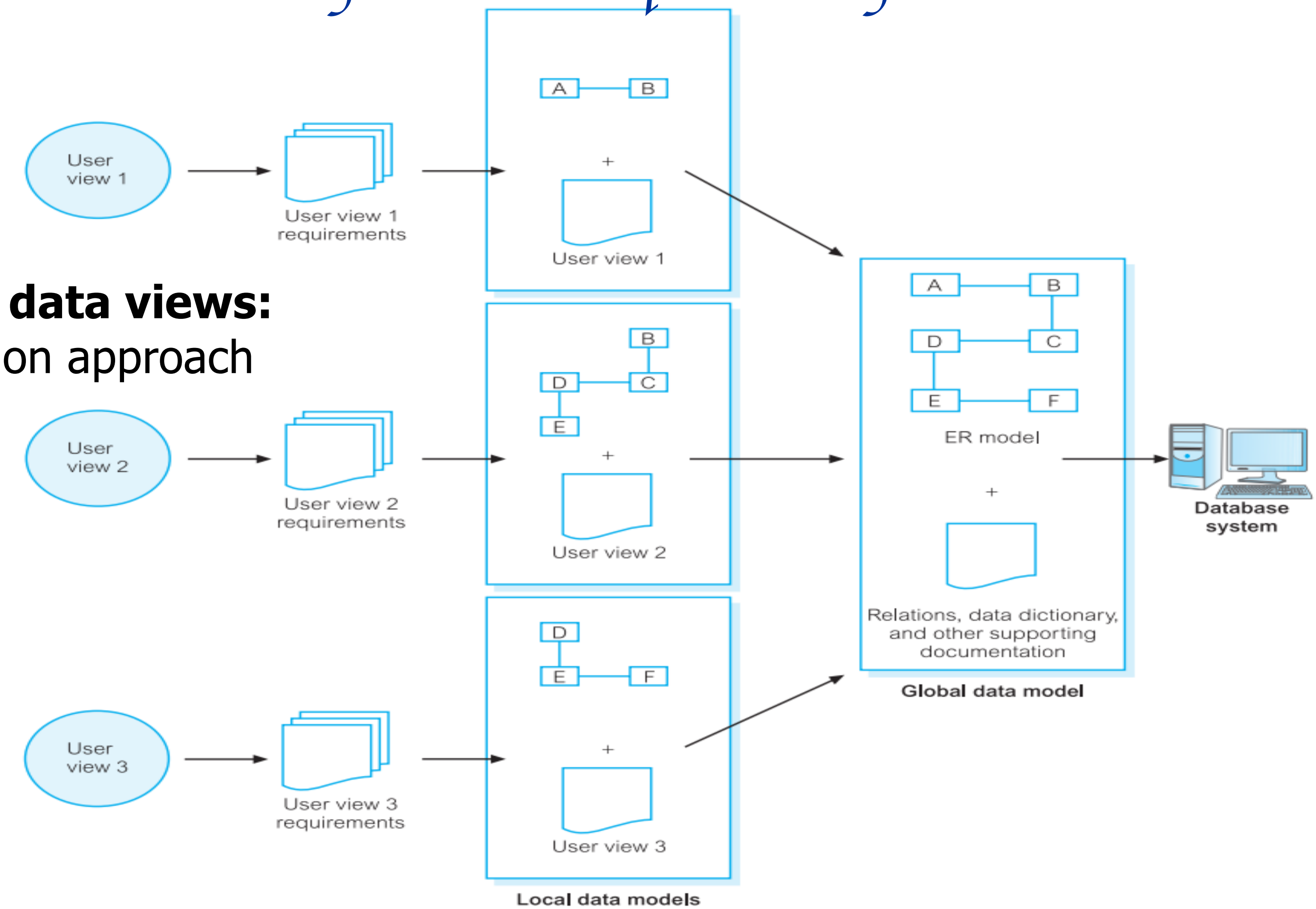
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 1: Data analysis and requirements

Where & How to find the required information?

Gather end-user data views:
The view integration approach



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Conceptual Design

Step 1: Data analysis and requirements

Where & How to find the required information?

- Develop and gather end-user data views
- **Directly observe the current system**
  - Review the existing system to identify the data and its characteristics.
  - Examine existing forms, file, etc. to discover the data type and volume.
- **Interface with system design groups**
  - Systems Analyst, DBA?

(Coronel and Morris, 2015) (Hoffer et al., 2019)

cdio 4.0
fit@hcmus

Conceptual Design

Step 1: Data analysis and requirements

Where & How to find the required information?

PURPOSE OF DOCUMENTATION	EXAMPLES OF USEFUL SOURCES
Describes problem and need for database	Internal memos, emails, and minutes of meetings Employee complaints and documents that describe the problem Social media such as blogs and tweets Performance reviews/reports
Describes the part of the enterprise affected by problem	Organizational chart, mission statement, and strategic plan of the enterprise Objectives for the part of the enterprise being studied Task/job descriptions Samples of completed manual forms and reports Samples of completed computerized forms and reports
Describes current system	Various types of flowcharts and diagrams Data dictionary Database system design Program documentation User/training manuals

Sources for examining the existing system

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 1: Data analysis and requirements

Criteria for an accurate data model: Business Rules

- **Business rules** (BR): A brief and precise description of a policy, procedure, or principle within an organization.
  - *Each student must be affiliated with only one department.*
  - *Student records must have a valid student ID and name.*
  - *A course must be managed by a department.*
- Well written BR help **defining entities, attributes, relationships, cardinalities, and constraints**.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 1: Data analysis and requirements

Criteria for an accurate data model: Business Rules

- **Sources for deriving business rules**
  - Description of operations, determined by company managers, policymakers, department managers, etc.
  - Written documents such as company procedures, standards, and operations manuals.
  - Direct interviews with end-users.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

cdio 4.0
fit@hcmus

Conceptual Design

Step 1: Data analysis and requirements

Criteria for an accurate data model: ***Business Rules***

- **Importance of BR**
  - They help standardize the company's view of data.
  - They constitute a communications tool between users and designers.
  - They allow the designer to understand the nature, role, and scope of the data.
  - They allow the designer to understand business processes.
  - They allow the designer to develop appropriate relationship participation rules and foreign key constraints.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Conceptual Design

Step 1: Data analysis and requirements

Criteria for an accurate data model: ***Business Rules***

TABLE 2-1 Characteristics of a Good Business Rule

Characteristic	Explanation
Declarative	A business rule is a statement of policy, not how policy is enforced or conducted; the rule does not describe a process or implementation but rather describes what a process validates.
Precise	With the related organization, the rule must have only one interpretation among all interested people, and its meaning must be clear.
Atomic	A business rule marks one statement, not several; no part of the rule can stand on its own as a rule (i.e., the rule is indivisible, yet sufficient).
Consistent	A business rule must be internally consistent (i.e., not containing conflicting statements) and must be consistent with (and not contradict) other rules.
Expressible	A business rule must be able to be stated in natural language, but it will be stated in a structured natural language so that there is no misinterpretation.
Distinct	Business rules are not redundant, but a business rule may refer to other rules (especially to definitions).
Business-oriented	A business rule is stated in terms businesspeople can understand, and because it is a statement of business policy, only businesspeople can modify or invalidate a rule; thus, a business rule is owned by the business.

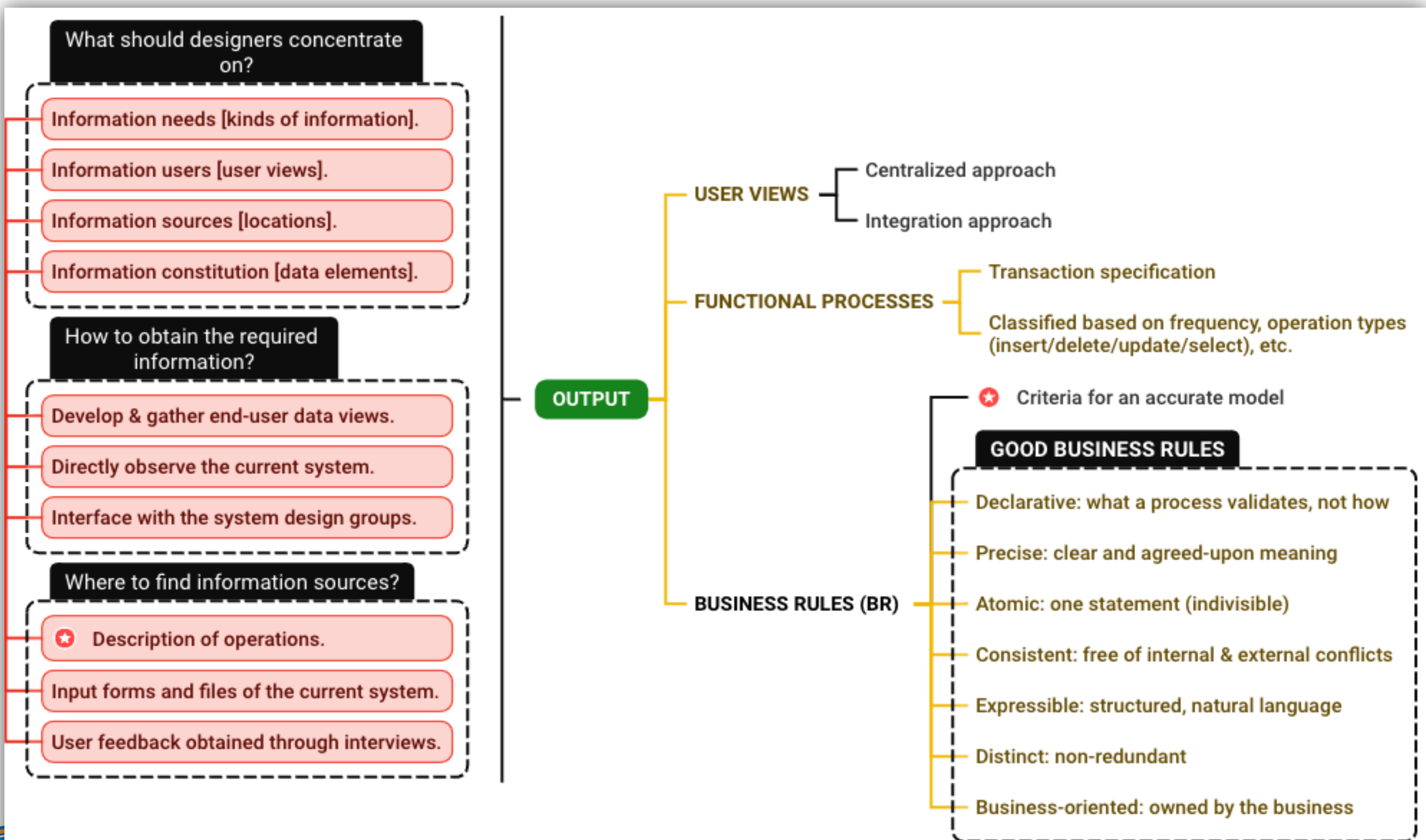
Source: Based on Gottesdiener (1999) and Plotkin (1999).

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Conceptual Design

Step 1: Data analysis and requirements



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Conceptual Design

Step 2: ER modeling and normalization

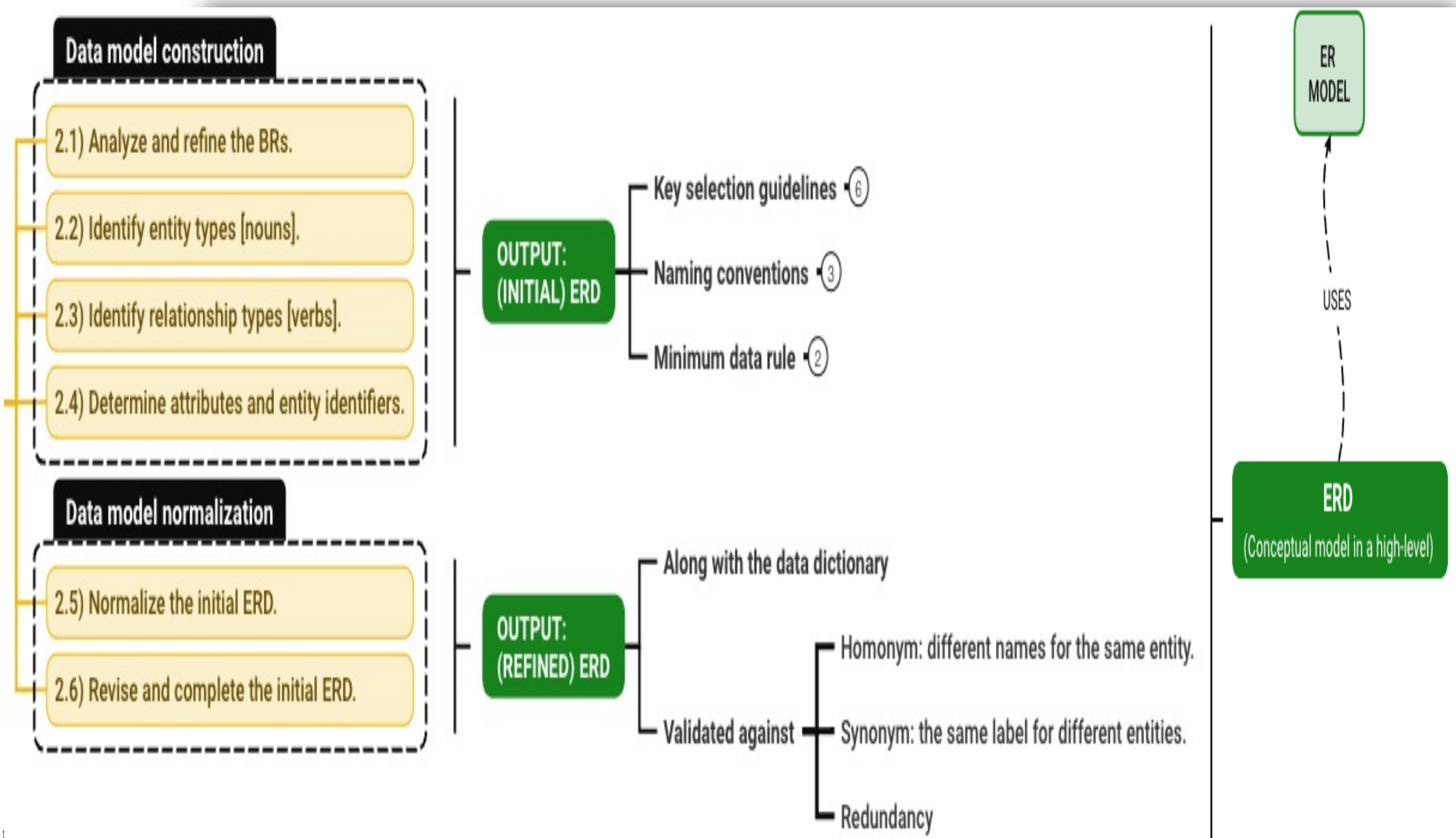
- **Objective**: All objects (entities, attributes, relations, views, and so on) are defined in a data dictionary, which is used in tandem with the normalization process to help eliminate data anomalies and redundancy problems.
- **Output**: Refined (E) ERD
- **Tools**: Data models, Graphical editors

STEP	ACTIVITY
1	Data analysis and requirements
2	Entity relationship modeling and normalization
3	Data model verification
4	Distributed database design

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

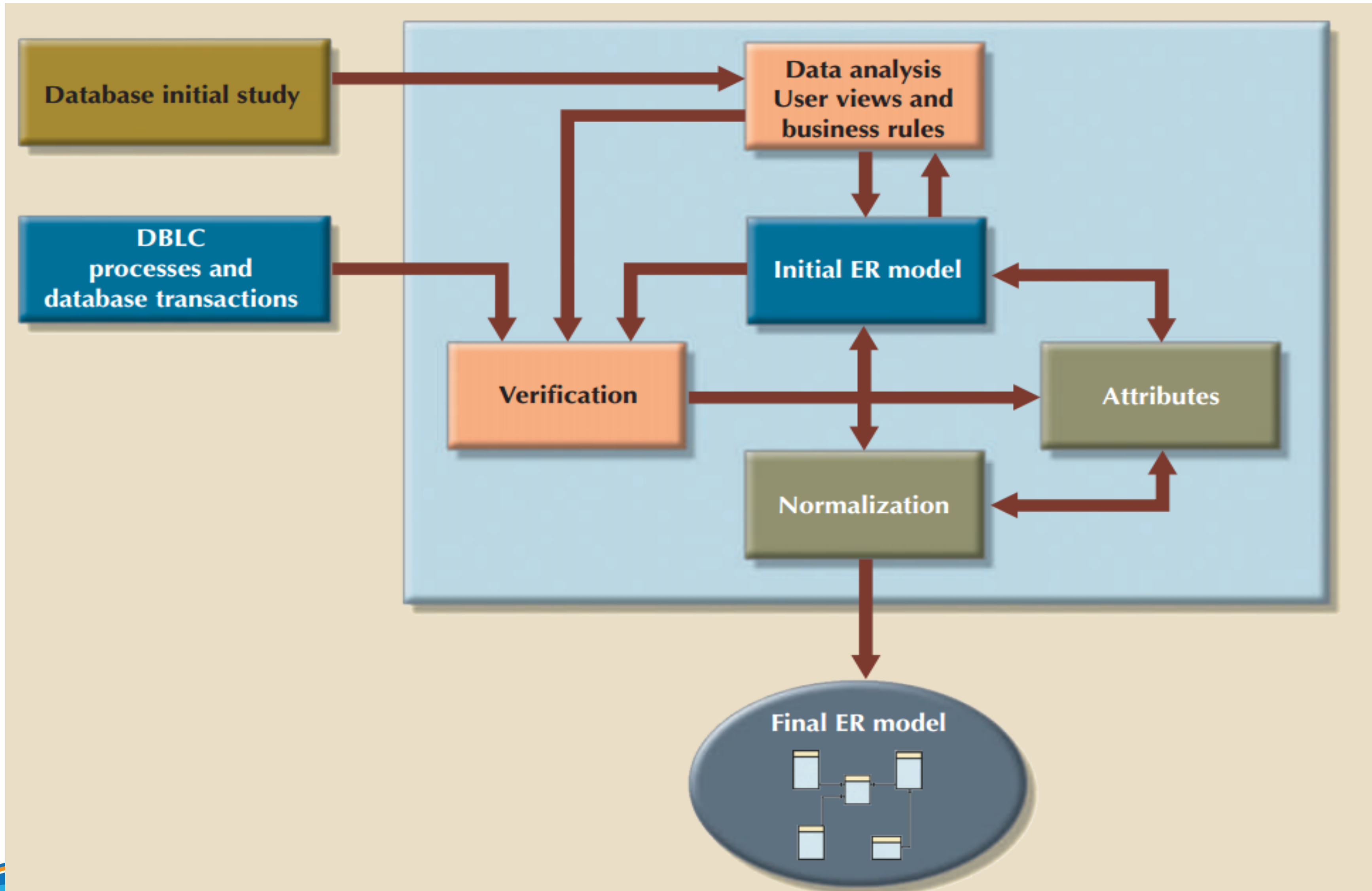
Step 2: ER modeling and normalization



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 2: ER modeling and normalization



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 2: ER modeling and normalization

- Responsibilities of a designers (**define & decide**):
  - Define entities, attributes, primary keys, and foreign keys
  - Make decisions about adding new primary key attributes
  - Make decisions about the treatment of composite and multivalued attributes
  - Make decisions about adding derived attributes to satisfy processing requirements
  - Make decisions about the placement of foreign keys in 1:1 relationships

Conceptual Design

Step 2: ER modeling and normalization

- Points to remember:
  - Avoid unnecessary ternary relationships.
  - Draw the corresponding ER diagram.
  - Normalize the entities.
  - Include all data element definitions in the data dictionary.
  - Make decisions about standard naming conventions.

Conceptual Design

Step 2: ER modeling and normalization

- **Recall: Naming Conventions**
  - Facilitates communication between parties
  - Promotes self-documentation
  - **Entity name requirements**
    - A singular noun or noun phrase
    - Specific to the organization (e.g., Customer or Client)
    - Clear and descriptive
    - Familiar to the users
    - Concise (as few words as possible)
    - Consistent

Conceptual Design

Step 2: ER modeling and normalization

- **Recall: Naming Conventions**
  - **Attribute name requirements**
    - A singular noun or noun phrase
    - Unique for clarity purposes → Follow a standard format

      [Entity type name { [Qualifier] }] Class
    - Defined using the same qualifiers and classes with other similar attributes of different entity types

Conceptual Design

Step 2: ER modeling and normalization

- **Recall: Naming Conventions**
  - **Relationship name requirements**
    - Be a verb or verb phrase
    - Represent business rules in a meaningful manner
    - Indicate the action taken (or interaction) rather than the action results or process
    - Avoid vague names (e.g., Has, Is related to, …)

Conceptual Design

Step 2: ER modeling and normalization

- Recall: Key Selection

- Key selection guidelines
  - **Unique values**
    --Uniquely identify each entity instance--
  - **Non-intelligent**
    --No embedded semantic meaning--
  - **No change over time**
    --Should be permanent and unchangeable--
  - **(Preferably) single-attribute**
    --Surrogate key could be considered to replace a composite key--
  - **(Preferably) numeric value**
    --Better for data processing at the physical level--
  - **Security-compliant**
    --No security risk or violation--

fit@hcmus

Conceptual Design

Step 3: Data model verification

- **Objective**: Verify the conceptual model against the proposed system processes to ensure they can be supported by the data model by running series of tests:
  - End-user data views
  - Required transactions (select, insert, delete, update)
  - Access rights and security
  - Business requirements and constraints
- **Output**: Verified data model

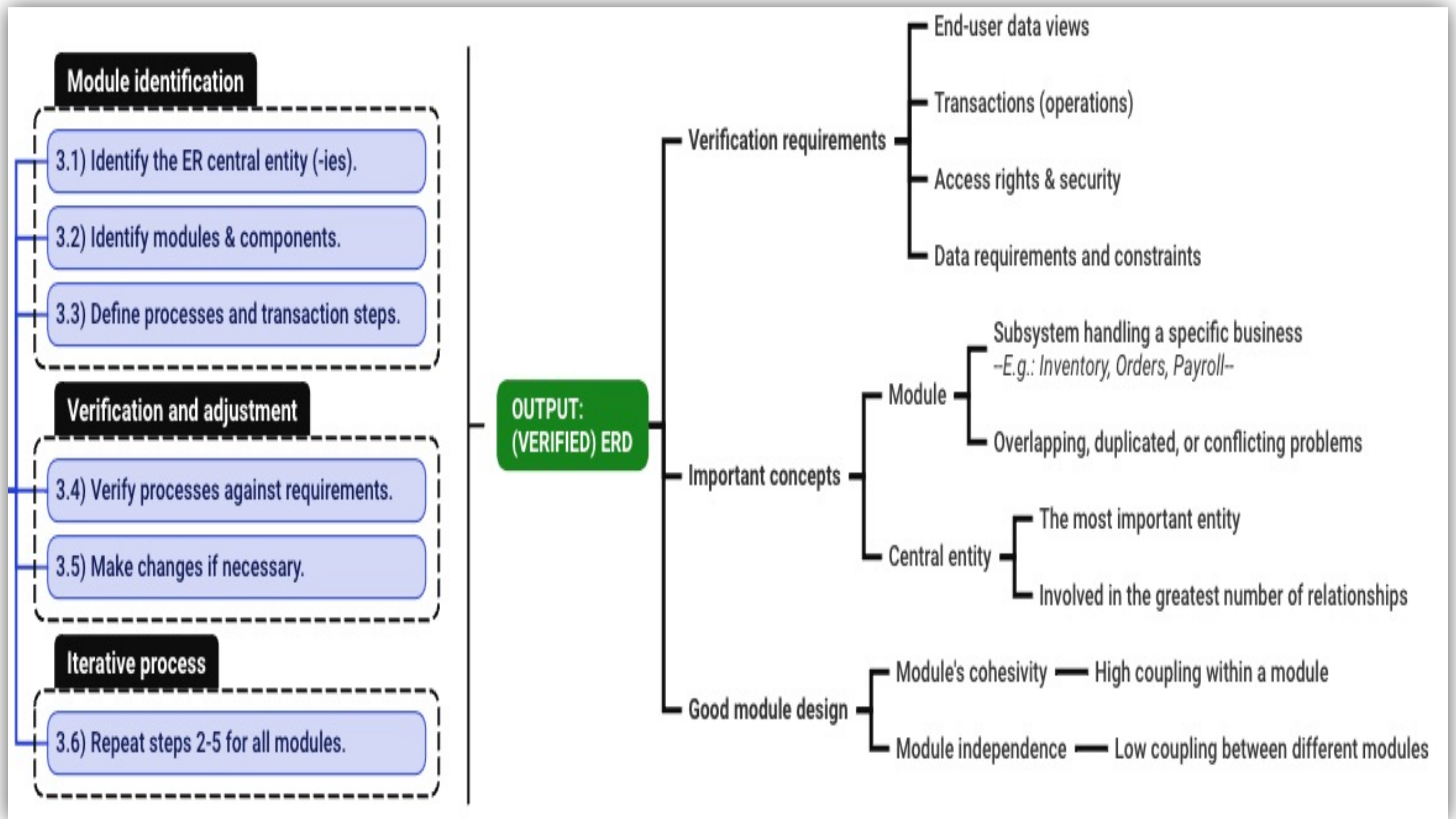
STEP	ACTIVITY
1	Data analysis and requirements
2	Entity relationship modeling and normalization
3	Data model verification
4	Distributed database design

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Conceptual Design

Step 3: Data model verification



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 3: Data model verification

- **Module**: An information system component that handles a specific function (e.g., inventory, orders, or payroll)
  - Can be delegated easily
  - Can be prototyped quickly
  - Simplify the design work
  - Speed up the development work
- **Problems**
  - Might present overlapping, duplicated, or conflicting views of the same data
  - Might not be able to support all processes in the system's modules

fit@hcmus

Conceptual Design

Step 3: Data model verification

- A **central entity**: Most important entity involved in the greatest numbers of relationships.
- A central entity belongs to the **module** (determined by its boundary and scope) that uses it most frequently.
- A **process** may be classified according to:
  - Frequently (Daily, …)
  - Operation types (CRUD)

Conceptual Design

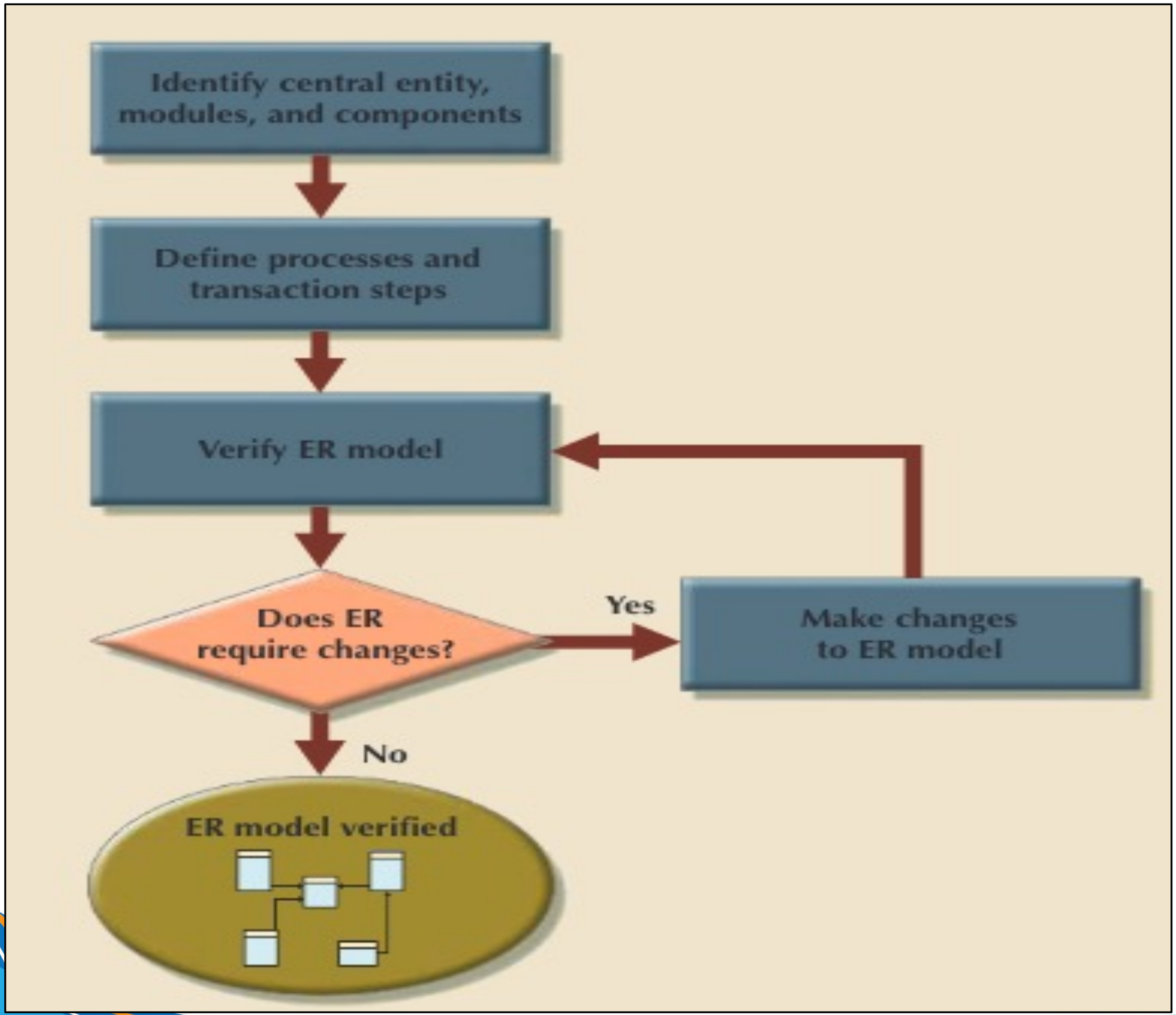
Step 3: Data model verification

- Ensure the module's cohesivity
  - The entities must be strongly related, and the module must be complete and self-sufficient
- Analyze each module's relationships with other modules to address module coupling
  - Modules must display low coupling, indicating that they are independent of other modules
- Verified each module with all identified processes
  - Each module will be verified with all identified processes to ensure its capacity to support the processes

fit@hcmus

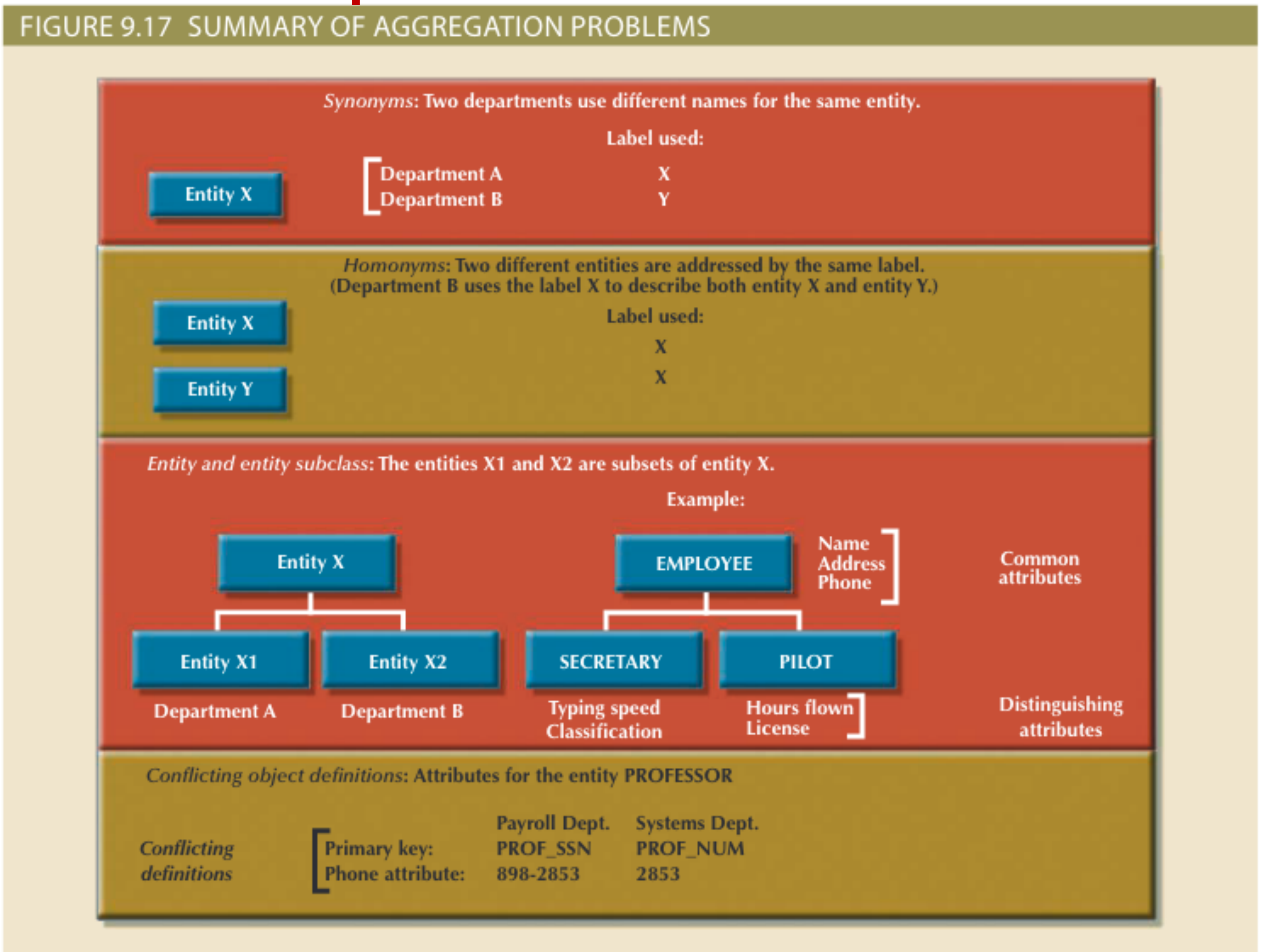
Conceptual Design

Step 3: Data model verification



cdio 4.0
fit@hcmus

Conceptual Design
Step 3: Data model verification



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Conceptual Design

Step 4: Distributed database design

- **Objective**: All objects (entities, attributes, relations, views, and so on) are defined in a data dictionary, which is used in tandem with the normalization process to help eliminate data anomalies and redundancy problems.
- **Output**: Refined (E) ERD

STEP	ACTIVITY
1	Data analysis and requirements
2	Entity relationship modeling and normalization
3	Data model verification
4	Distributed database design

Conceptual Design

Step 4: Distributed database design

- If the database data and processes will be distributed across the system, portions of a database (database fragments) may reside in several physical locations.
- A **database fragment** is a subset of a database that is stored at a given location.
- Define strategy to ensure database integrity, security, and performance

fit@hcmus

Outline

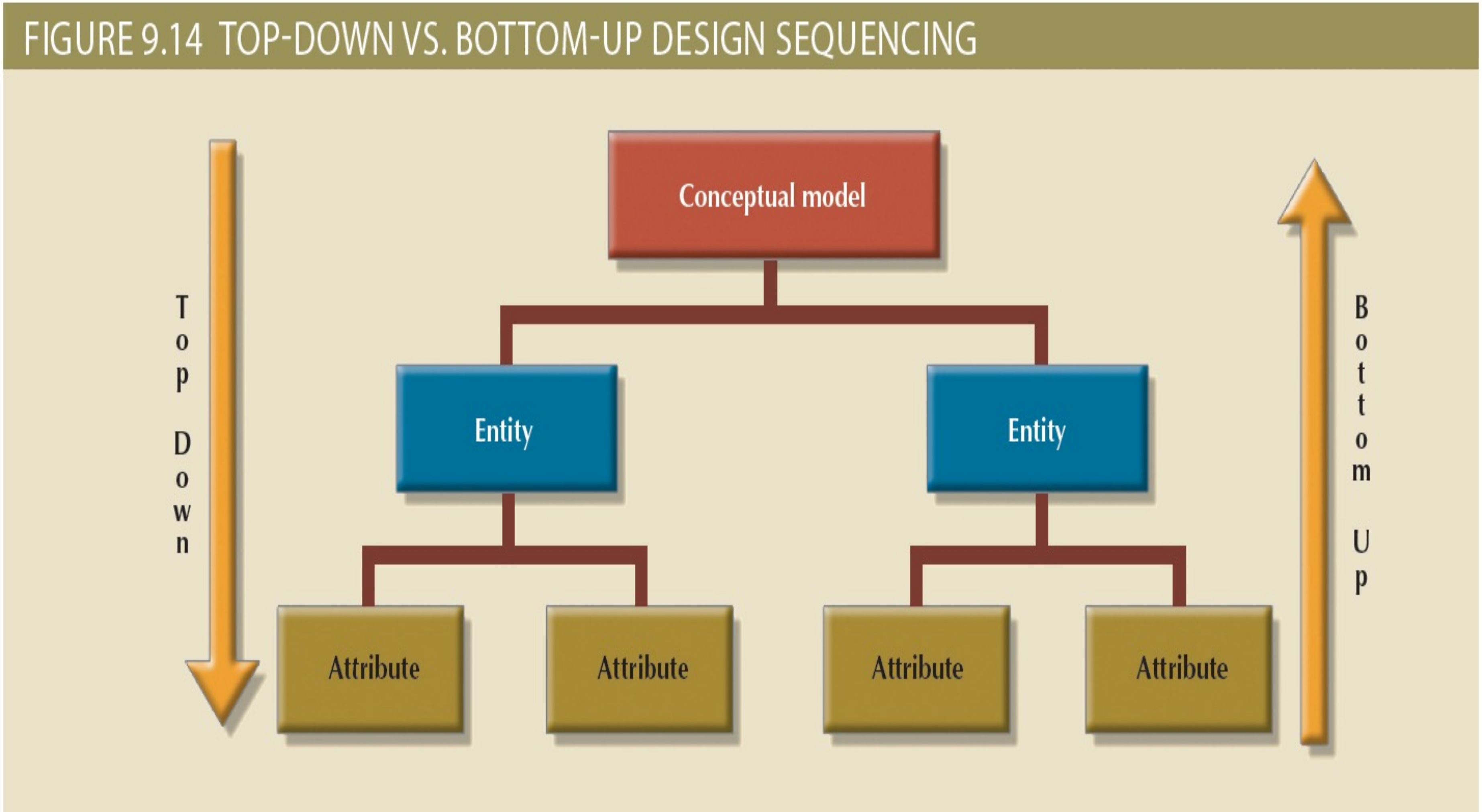
- Recall: SDLC, DBLC & Database Design
- Conceptual Design
- Design Strategies
- Database Design Cases

Design Strategies

- Top-down vs. Bottom-down Design
- Centralized vs. Decentralized Design
- Inside-Out Design
- Design decision is normally based on:
  - Size and scope of the project (or the system)
  - Company’s structure

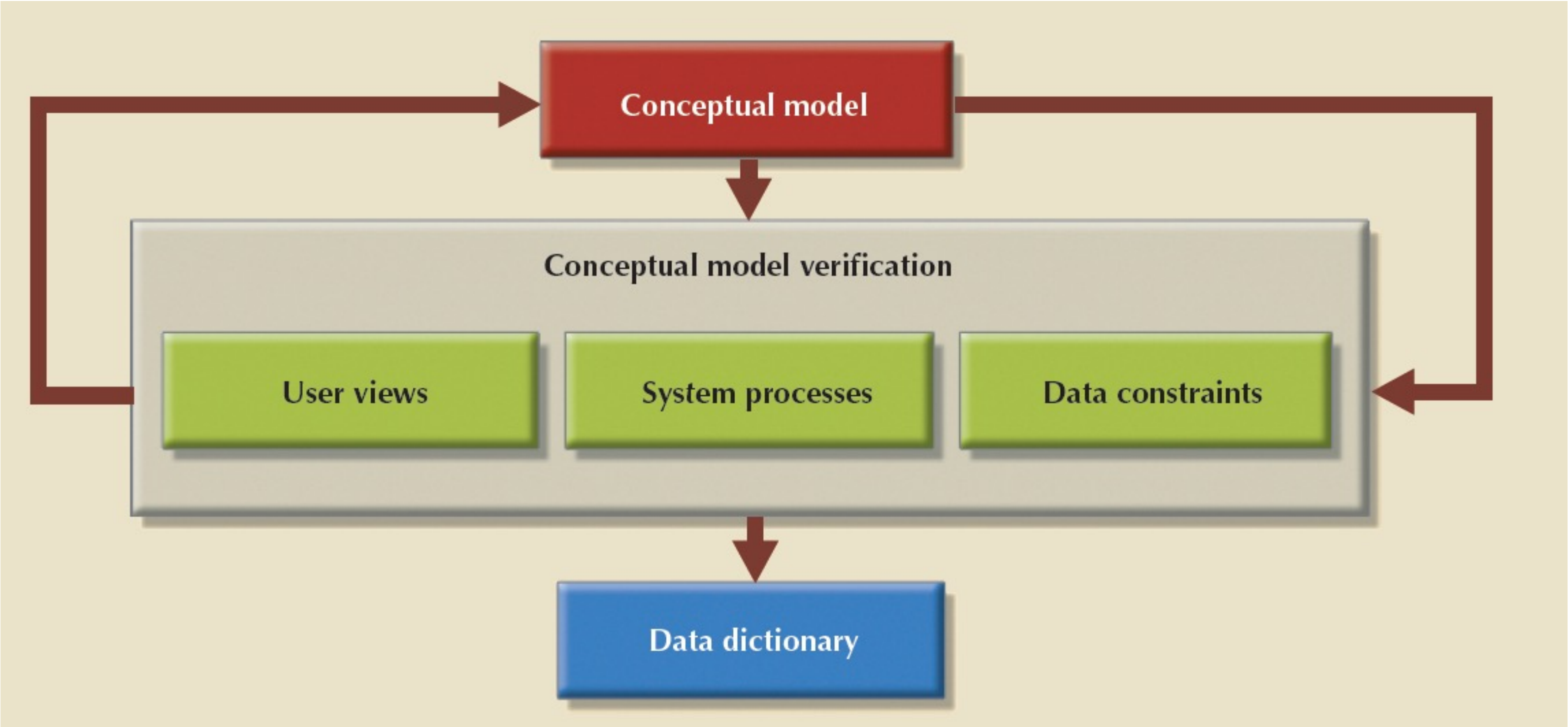
(Coronel and Morris, 2015) (Hoffer et al., 2019)

Top-down vs. Bottom-up Design



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Centralized vs. Decentralized Design

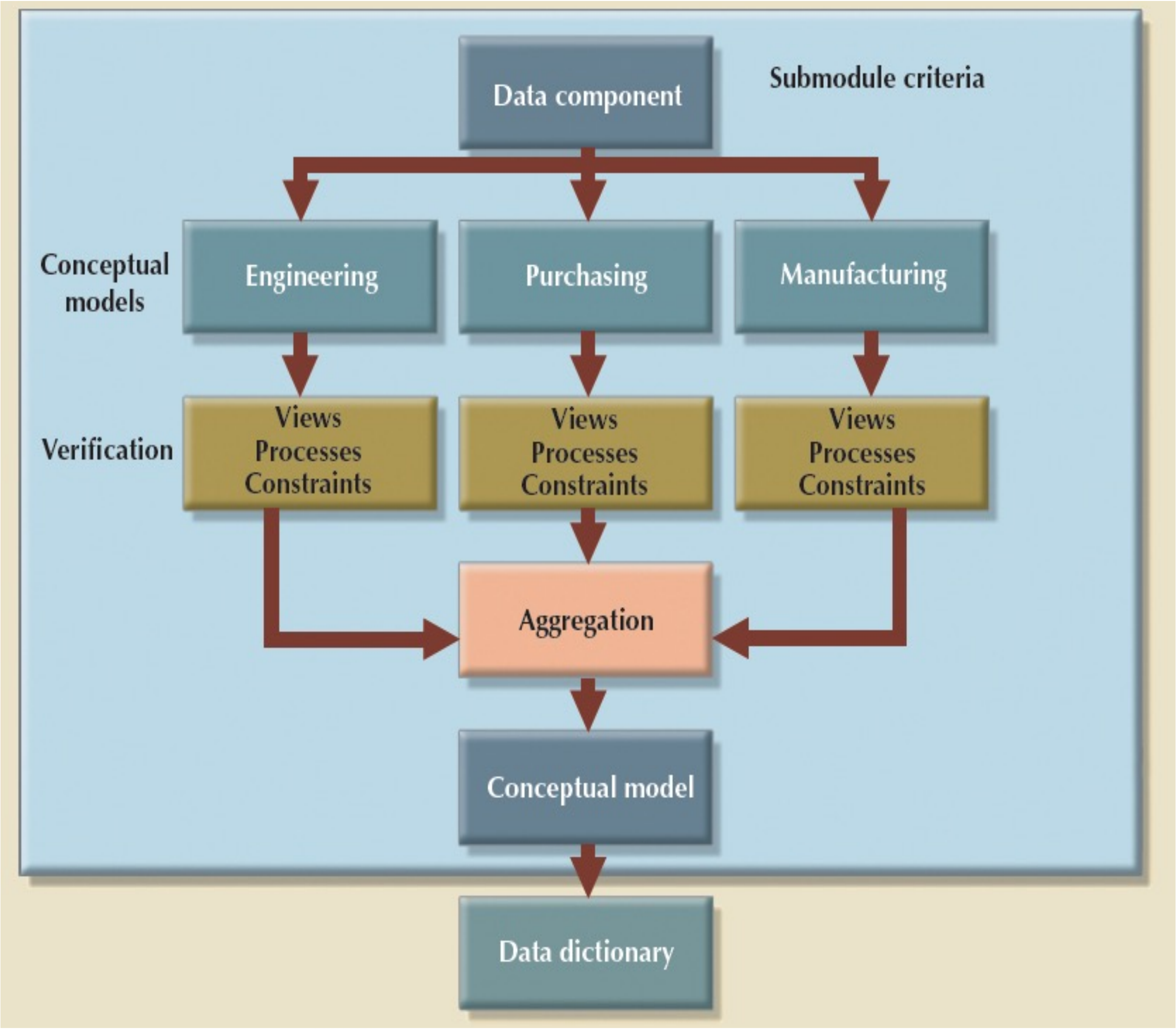


- All database design decisions are carried out centrally by a small group of people.
- Suitable for a small databases with limited operations, as a single unit or department in an organization.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Centralized vs. Decentralized Design



- A DB designer team tackles the complex database design task which is divided into sub-modules.
- The subsets have been aggregated into a larger conceptual model.
- Suitable for large and complex projects.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

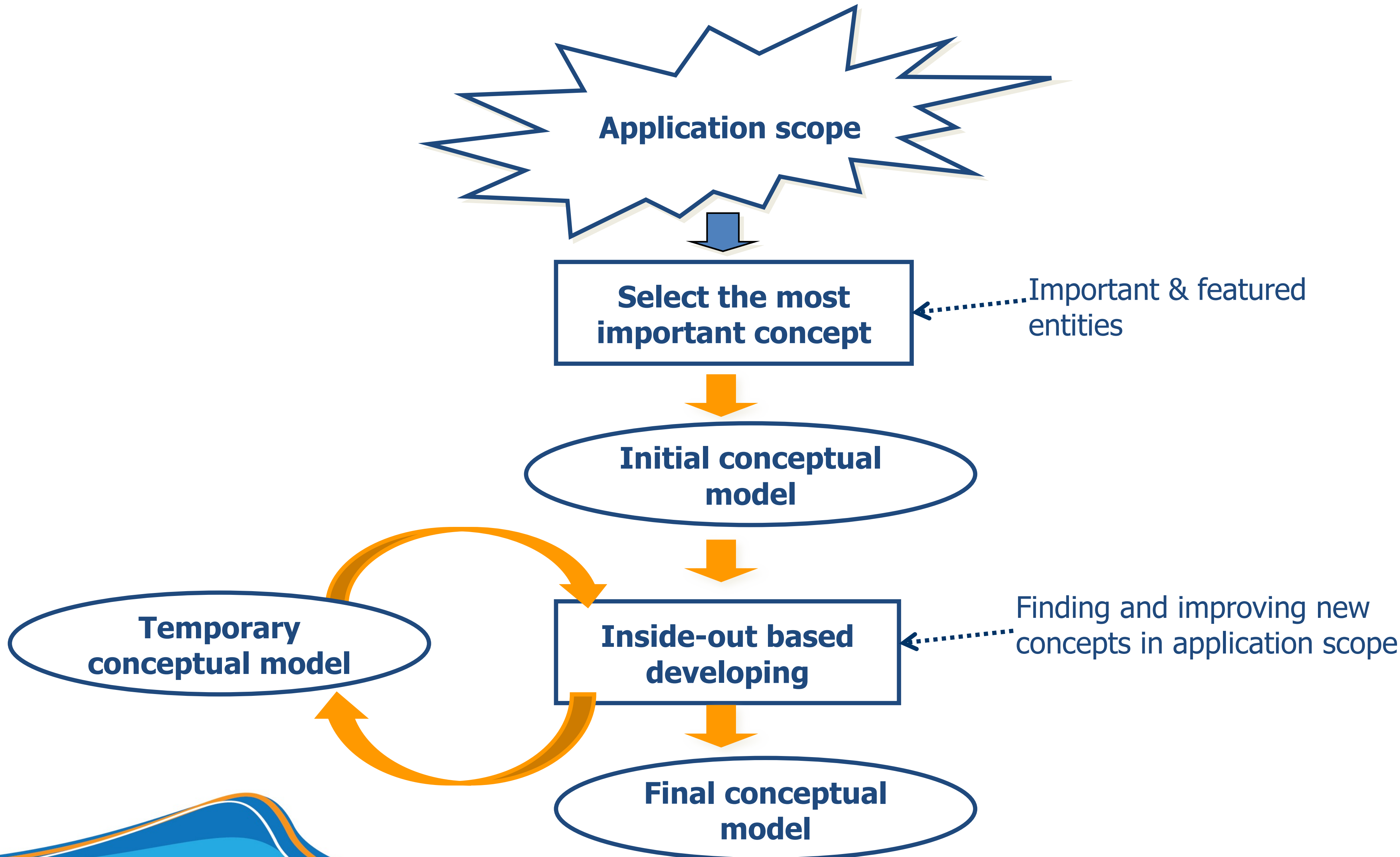
Inside-out Design

- A special case of a bottom-up strategy, which focuses on a central set of concepts that are most evident.

- This approach begins with the identification of set of major entities and then spreading out to consider other entities, relationships and attributes associated with those first identified.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Inside-out Design



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

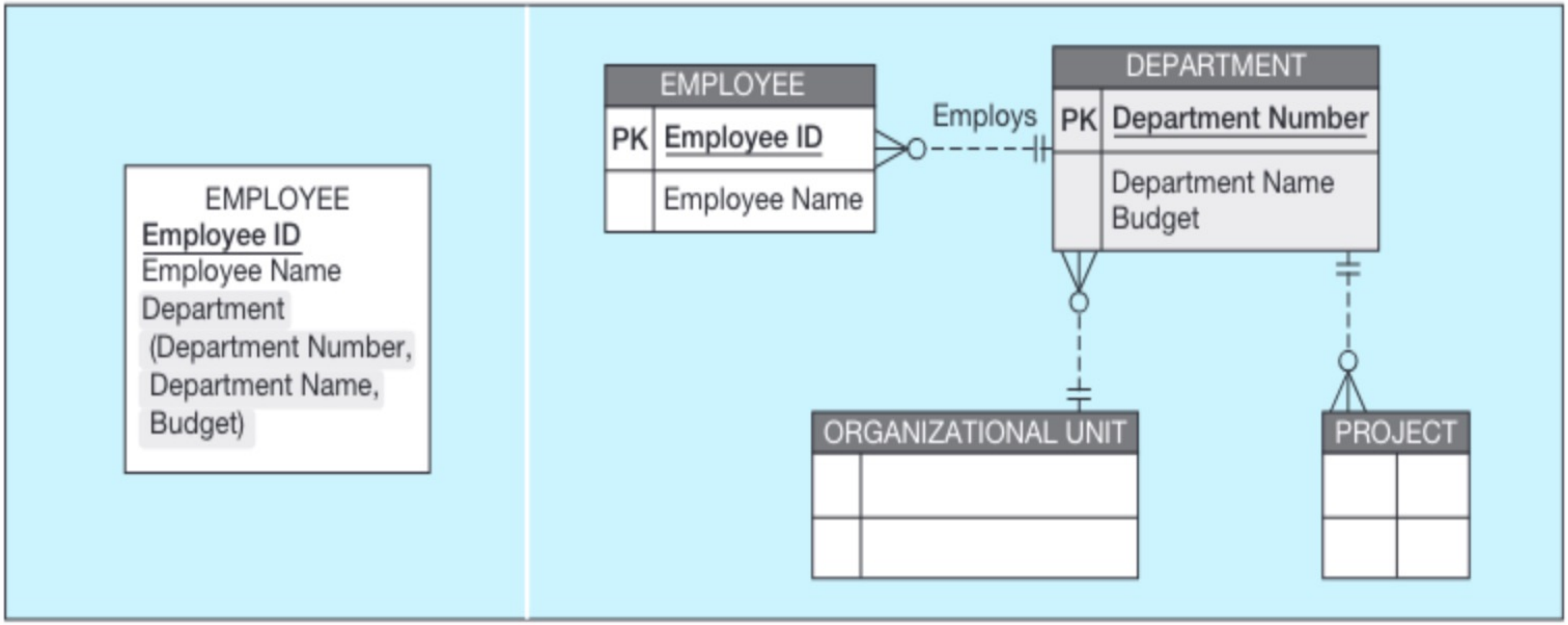
Outline

- Recall: SDLC, DBLC & Database Design
- Conceptual Design
- Design Strategies
- Database Design Cases

Design Case #1

Attributes OR Entities/Relationships?

Composite attribute vs. Entity



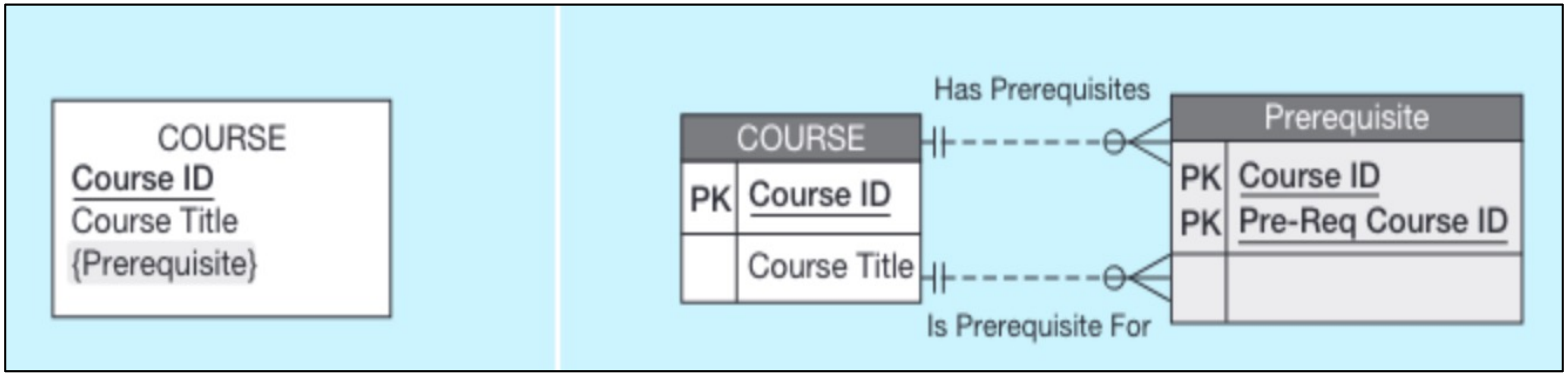
(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

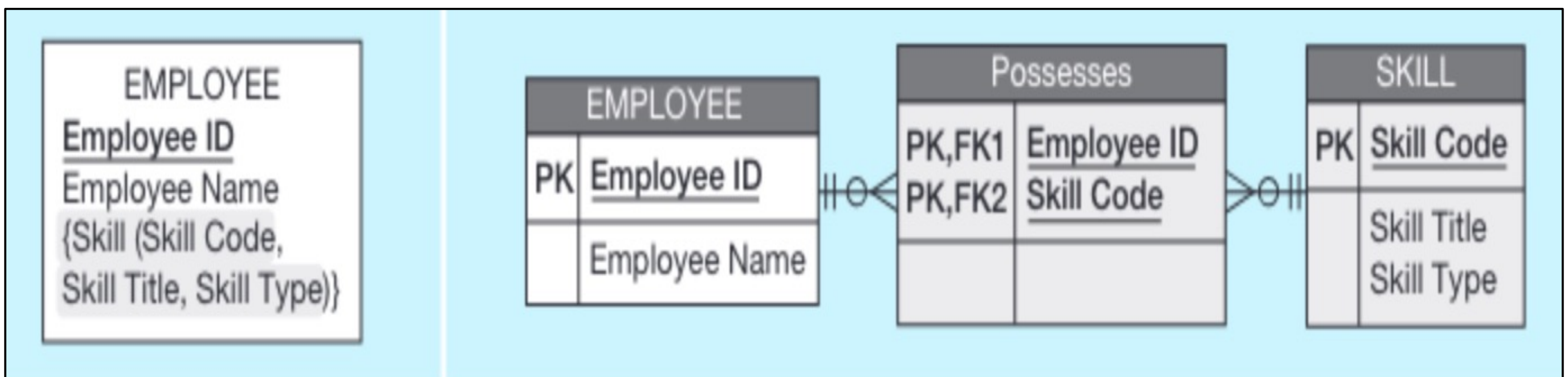
Design Case #1 (cont.)

Attributes OR Entities/Relationships?

Multi-valued attribute vs. Entity?



Multi-valued, Composite attribute vs. Entity?



(Coronel and Morris, 2015) (Hoffer et al., 2019)

Design Case #1 (cont.)

Attributes OR Entities/Relationships?

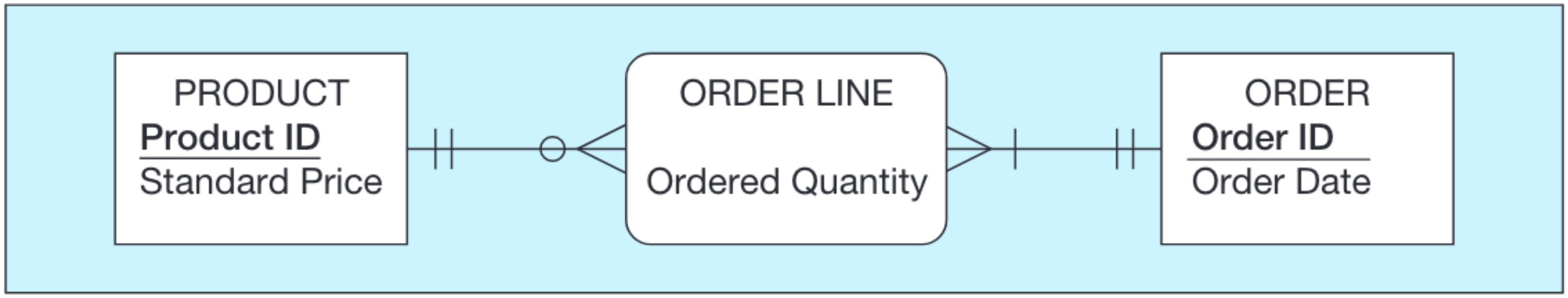
- Attributes could be modeled as entity/relationship when:
  - Have Identifiers
  - Be shared by multiple entity instances
  - Defined by a set of sub-attributes with implicit dependence, typically forming a new implicit entity.

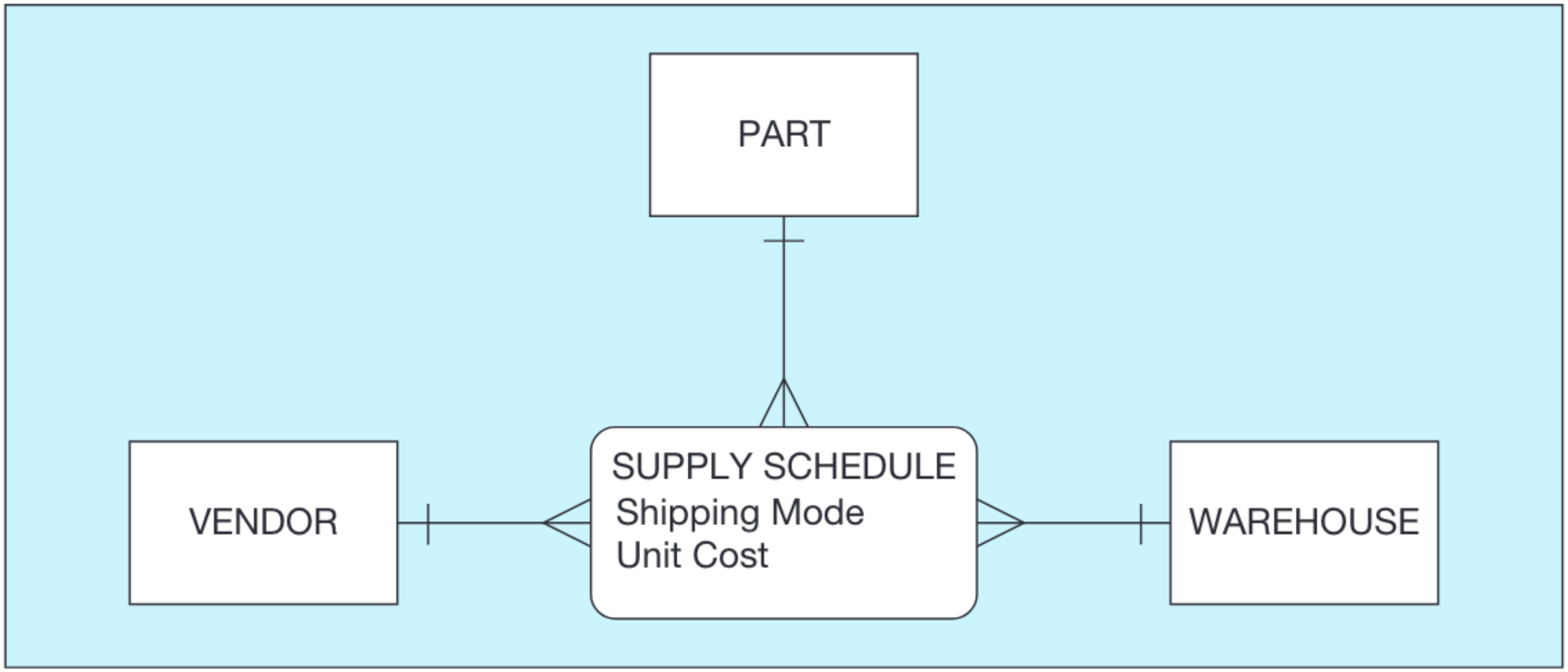
(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Design Case #2

Relationship OR Associative Entity





(Coronel and Morris, 2015) (Hoffer et al., 2019)

Design Case #2 (cont.)

Relationship OR Associative Entity

- Relationship should be converted to associative entity when:
  - Binary M-N relationships
    - Have business meaning independent of other entities
    - Have an identifier, and should also have other attributes
  - Ternary relationships
- The associative entity may participate in other relationships other than the entities of the associated relationship

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Design Case #3

Selecting Identifiers (Primary Keys)

- Identifiers (Primary Keys): Aim to guarantee entity integrity, not to “describe” the entity.
- Guidelines for selecting Identifiers
  - **Unique values** (uniquely identify each entity instance)
  - **Not intelligent** (not have embedded semantic meaning)
  - **No change over time** (be permanent and unchangeable)
  - **Preferably single-attribute** (minimum set of attributes)
  - **Preferably numeric** (for better managed)
  - **Security-compliant** (not be considered as a security risk)

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Design Case #3 (cont.)

Selecting Identifiers (Primary Keys)

- Natural Identifiers
  - Real-world identifiers to uniquely identify a real-world object (e.g., social security number, phone number).
  - Useful when there are no suitable primary keys available
- Composite Identifiers
  - A set of attributes to uniquely identify an entity instances
  - Useful when identifying:
    - Associative Entities
    - M-N Relationships
    - Weak Entities

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Design Case #3 (cont.)

Selecting Identifiers (Primary Keys)

- Surrogate Identifiers
  - Key generated by systems to simplify the identification of entity instances.
  - Have no meaning outside of the system

  ➔ **Should not be included in the conceptual model**

Identifier for the Event?

(DATE, TIME_START, ROOM) or **(DATE, TIME_END, ROOM)**

➔ **Surrogate key possible**

TABLE 5.4

DATA USED TO KEEP TRACK OF EVENTS

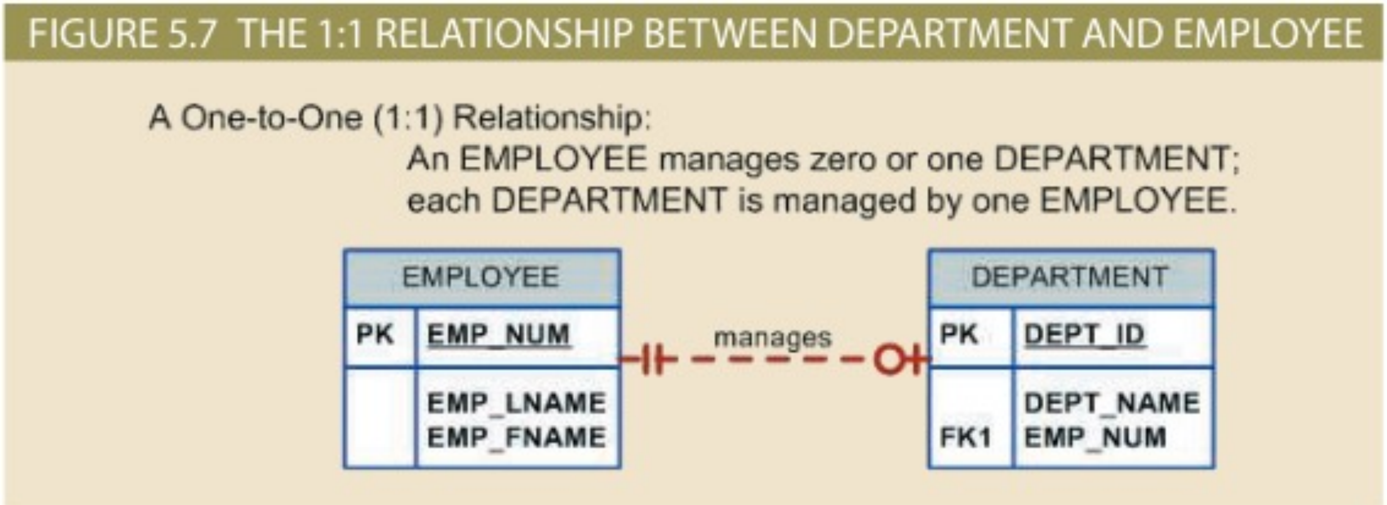
DATE	TIME_START	TIME_END	ROOM	EVENT_NAME	PARTY_OF
6/17/2016	11:00a.m.	2:00p.m.	Allure	Burton Wedding	60
6/17/2016	11:00a.m.	2:00p.m.	Bonanza	Adams Office	12
6/17/2016	3:00p.m.	5:30p.m.	Allure	Smith Family	15
6/17/2016	3:30p.m.	5:30p.m.	Bonanza	Adams Office	12
6/18/2016	1:00p.m.	3:00p.m.	Bonanza	Boy Scouts	33
6/18/2016	11:00a.m.	2:00p.m.	Allure	March of Dimes	25
6/18/2016	11:00a.m.	12:30p.m.	Bonanza	Smith Family	12

(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Design Case #4

Implementing 1-1 Relationship



SELECTION OF FOREIGN KEY IN A 1:1 RELATIONSHIP

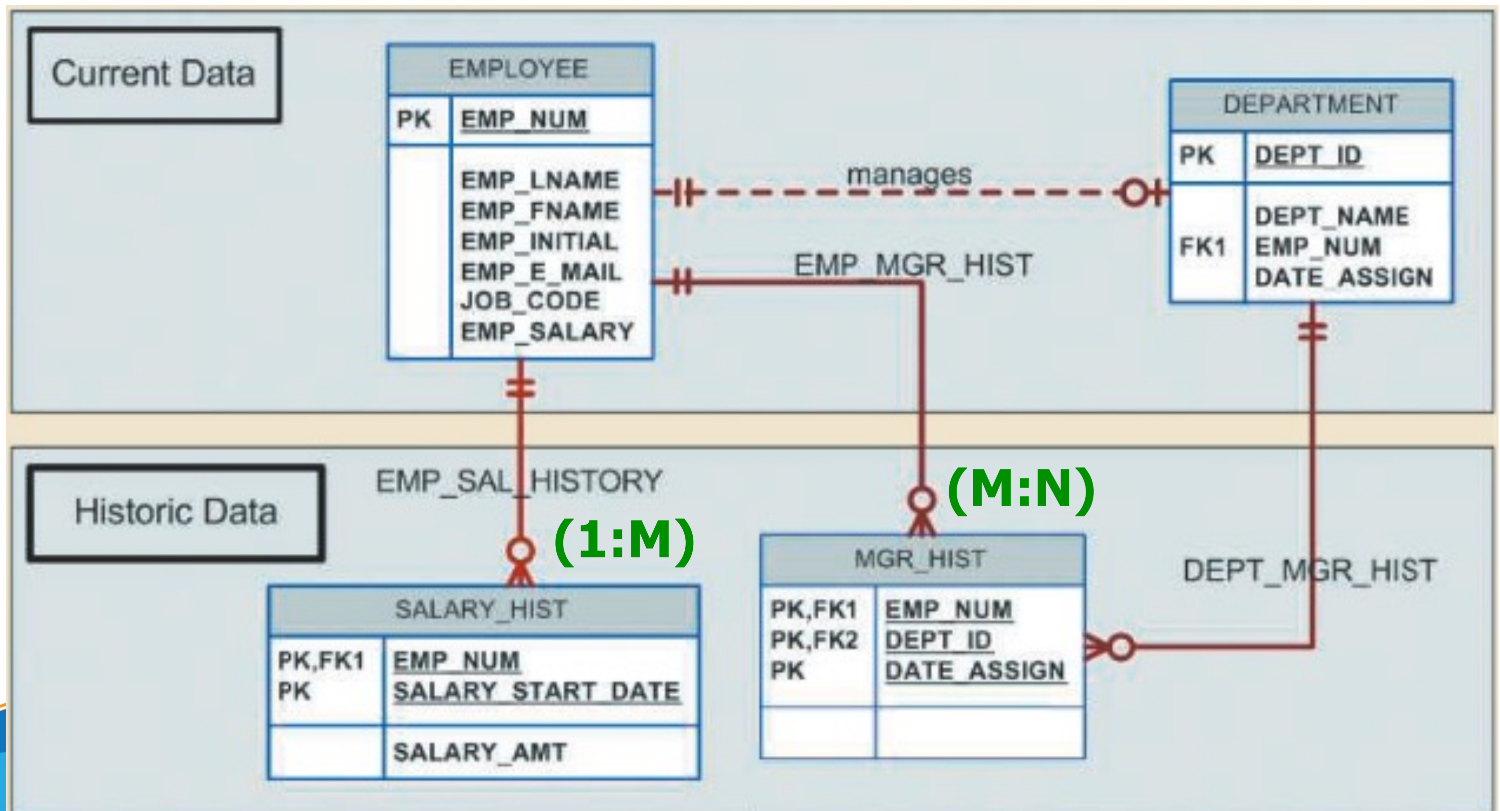
CASE	ER RELATIONSHIP CONSTRAINTS	ACTION
I (0,1)-(1,1)	One side is mandatory and the other side is optional.	Place the PK of the entity on the mandatory side in the entity on the optional side as a FK, and make the FK mandatory.
II (0,1)-(0,1)	Both sides are optional.	Select the FK that causes the fewest nulls, or place the FK in the entity in which the (relationship) role is played.
III (1,1)-(1,1)	Both sides are mandatory.	See Case II, or consider revising your model to ensure that the two entities do not belong together in a single entity.

(Coronel and Morris, 2015) (Hoffer et al., 2019)

Design Case #5

Maintaining History of Time-Variant Data

- Time-variant data refers to data whose values change over time ➔ must keep a history of the data changes.
- **Solution**: **Create a new entity in a (1:M) / (M:N) relationship with the original entity.**



(Coronel and Morris, 2015) (Hoffer et al., 2019)

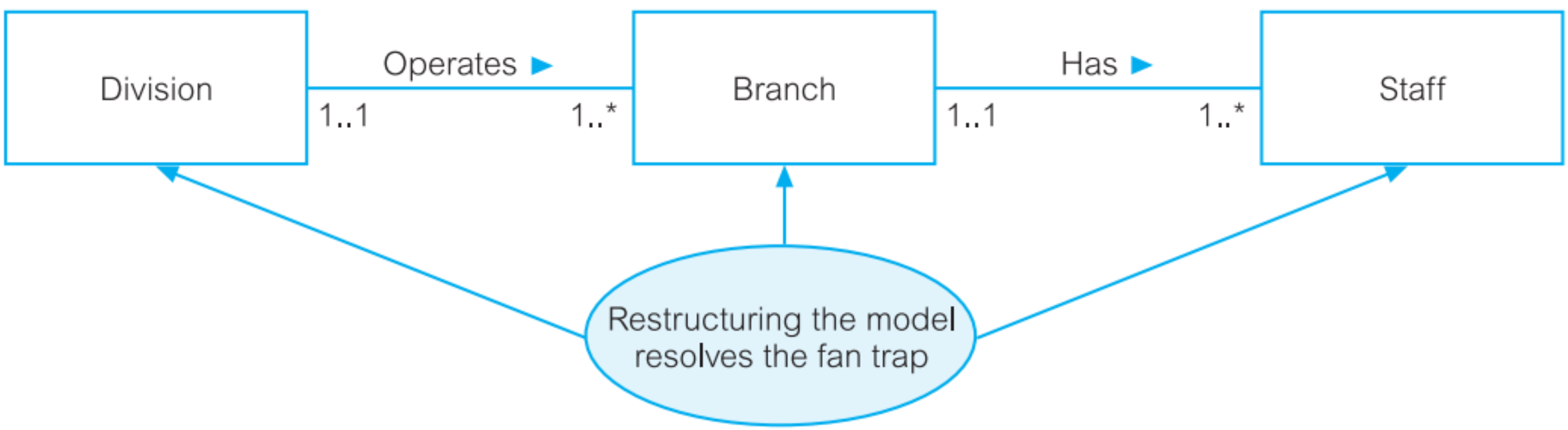
fit@hcmus

Design Case #6

Resolving Fan Traps Problem

- Fan Traps: Where a model represents a relationship between entity types, but the pathway between certain entity occurrences is ambiguous.

How to know which members of staff work at a particular branch? ➔ Restructuring models



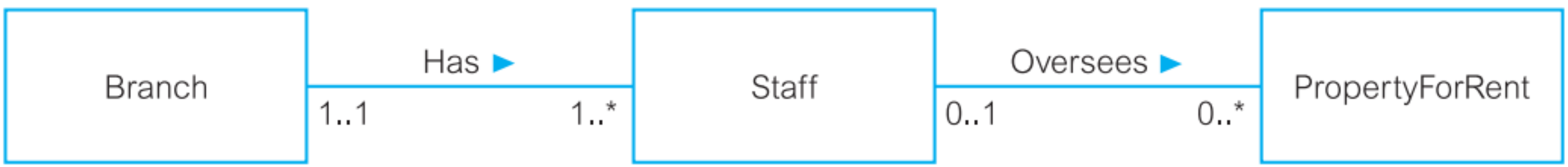
(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

Design Case #7

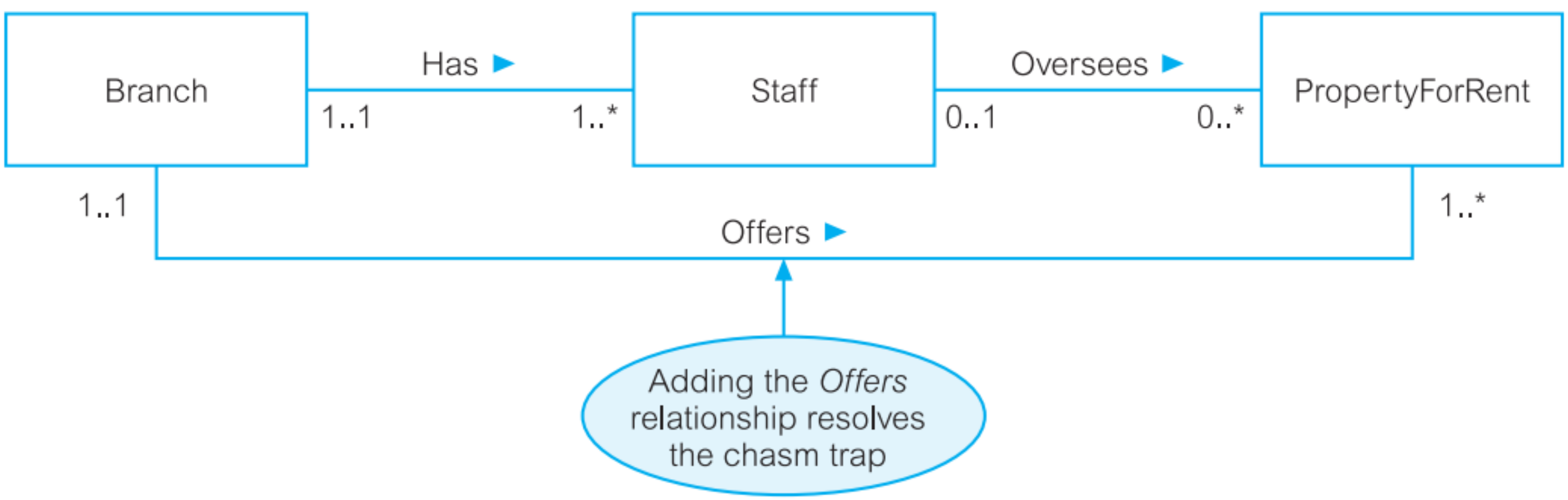
Resolving Chasm Traps Problem

- Chasm Traps: Where a model suggests the existence of a relationship between entity types, but the pathway does not exist between certain entity occurrences



At which branch is property number PA14 available?

➔ Restructuring models



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

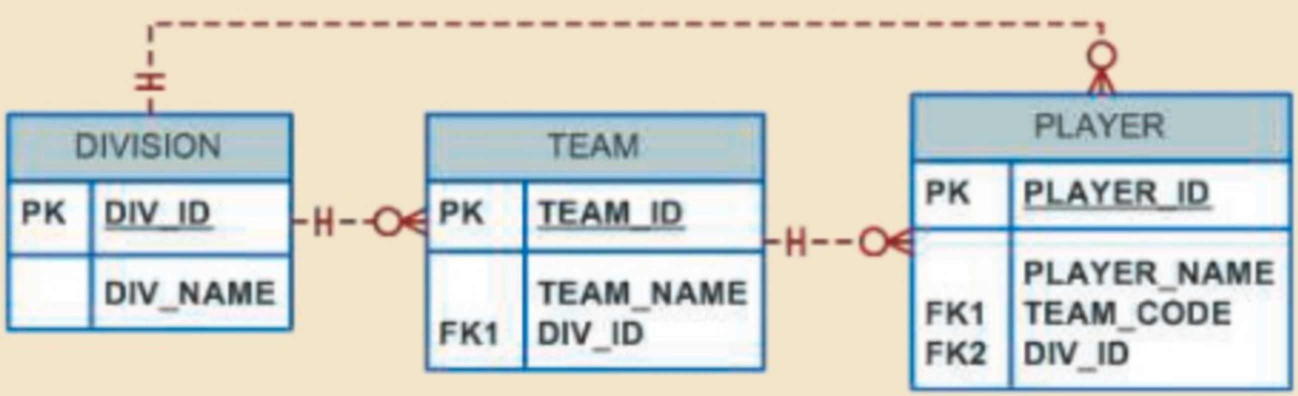
Design Case #8

Resolving Recurrency Problem

- Redundant relationships occur when there are multiple relationship paths between related entities.
- **Main concerns: Consistency** in the data model

➔ Redundant relationship **may be eliminated**?

FIGURE 5.13 A REDUNDANT RELATIONSHIP



(Coronel and Morris, 2015) (Hoffer et al., 2019)

fit@hcmus

References

(Coronel and Morris, 2015, **chapter 9**)

Database System: Design, Implementation, and Management, 12th Edition, Carlos Coronel & Stevem, Morris, 2015.

(Hoffer et al., 2019, **Part II, chapter 2**)

Modern Database Management, 13th Edition, Jeffrey A. Hoffer, V. Ramesh, Heikki Topi, 2019.

fit@hcmus

THE END