Chapter 5

Relational Database Analysis and Design with ER Model

การวิเคราะห์และออกแบบฐานข้อมูลเชิงสัมพันธ์ด้วยแบบจำลอง ER

ผศ.ดร.เทพฤทธิ์ บัณฑิตวัฒนาวงศ์

Content

- Data modeling strategies
- Guideline for ER model improvement
- ER model integration
- Transforming ER model to conceptual schema

Data Modeling Methodology

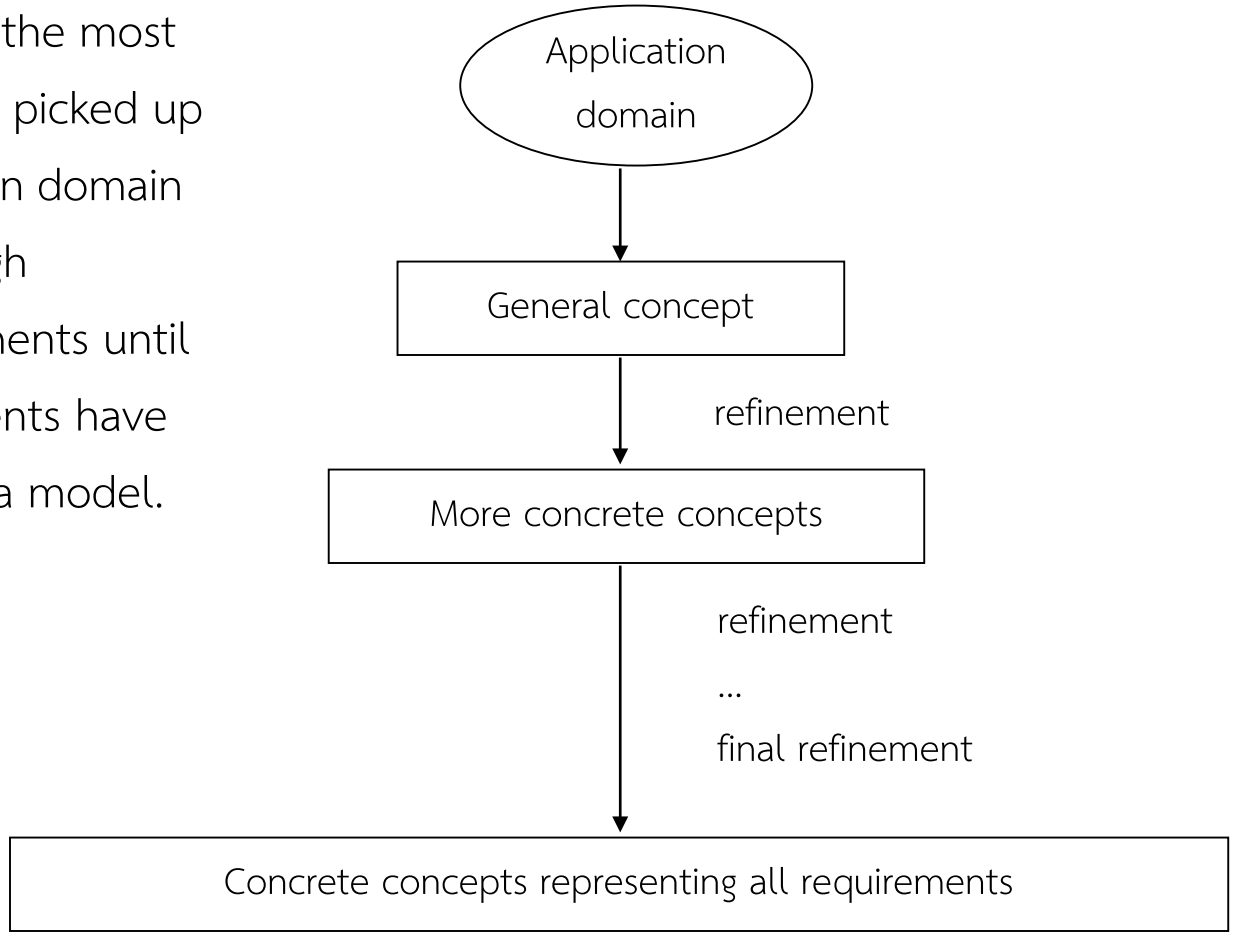
- A conceptual methodology for creating DB models through a data modeling language such as ER model.
- Consists of a set of primitives and strategy.
  - Example primitives are primitive constructs in ER model.
  - Strategy is a guideline for efficient data modeling.

Data Modeling Strategies

- Top-down strategy
- Bottom-up strategy
- Inside-out strategy
- Mixed strategy

Top-Down Strategy

With this strategy, the most general concept is picked up from an application domain and passed through successive refinements until all data requirements have been captured in a model.

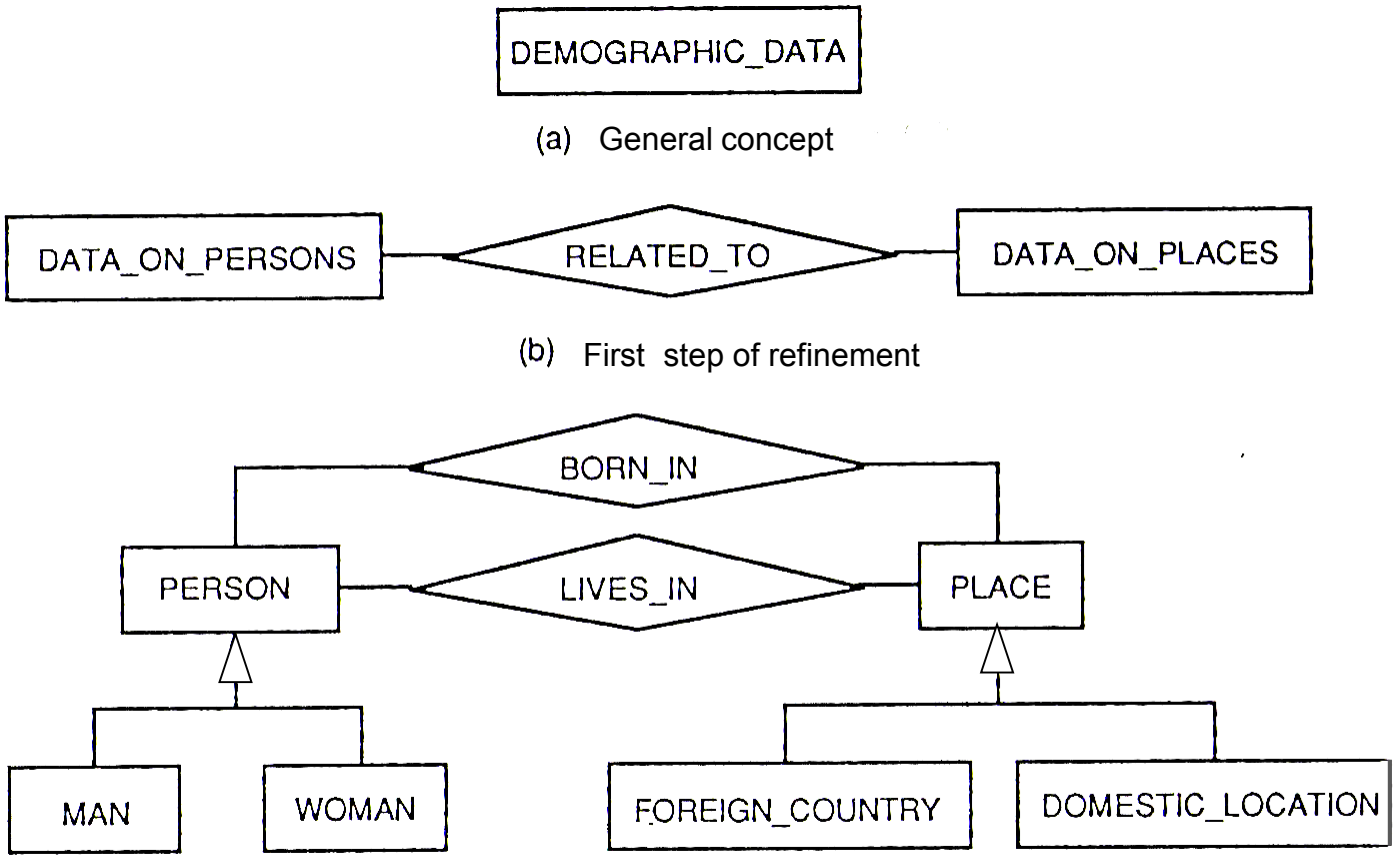


Top-Down Strategy (cont.)

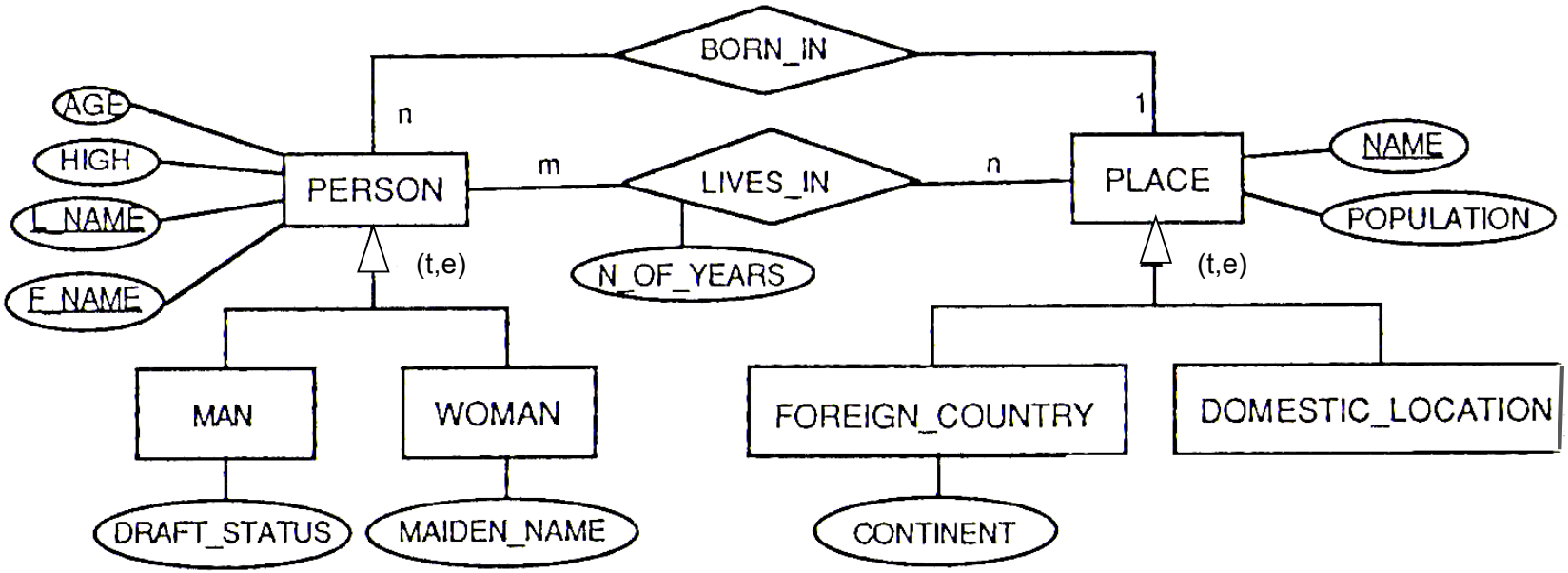
- Refinement steps:
  1. Adding aggregation relationship along with entities.
  2. Adding generalization along with more specific entities.
  3. Adding attributes, primary keys, connectivity, existence and (generalization) coverage.

Top-Down Strategy (cont.)

- Example

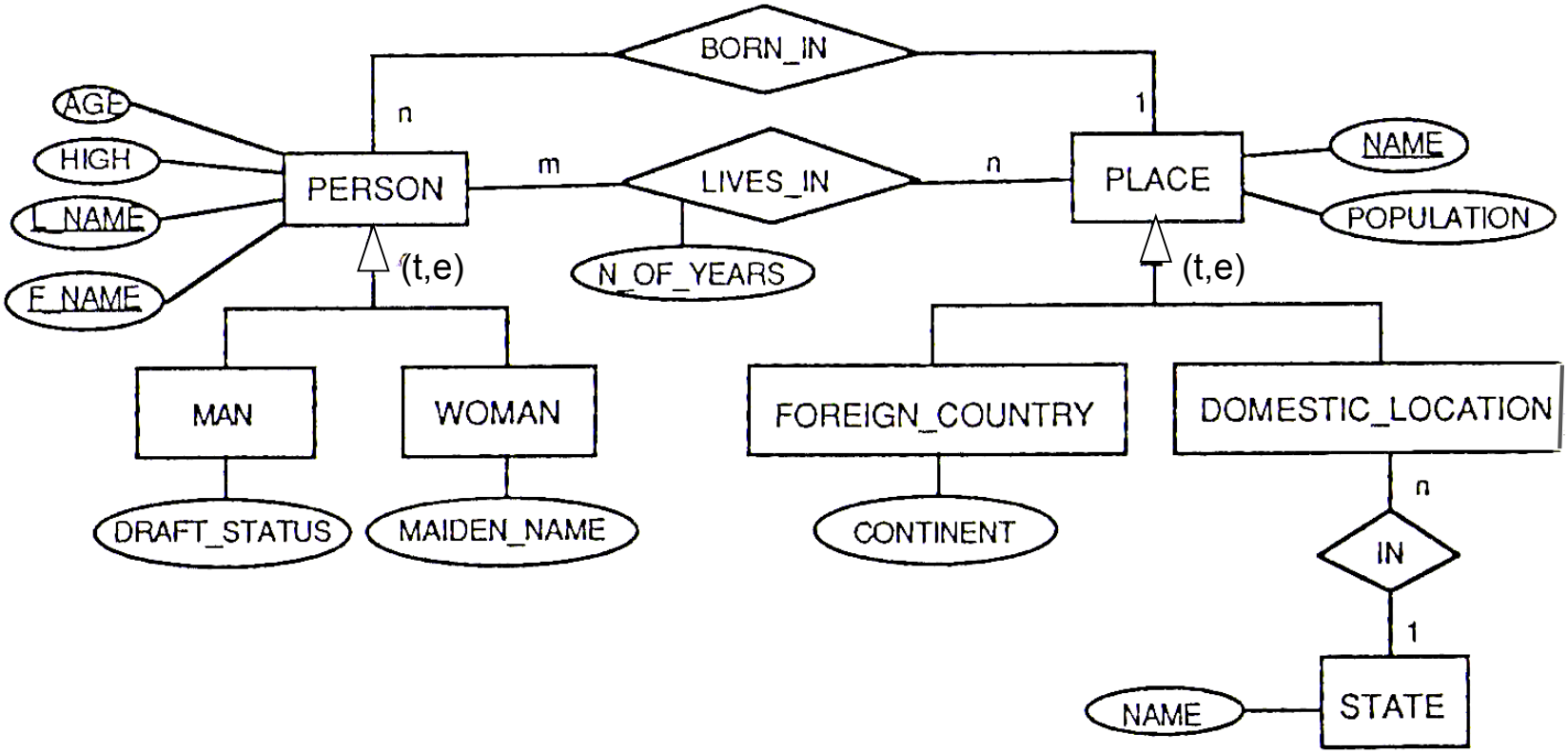


Top-Down Strategy (cont.)



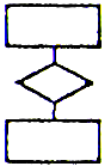
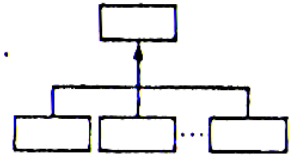
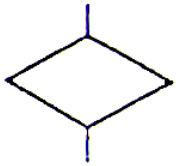
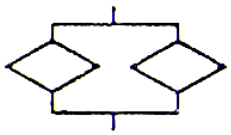
(d) Third step of refinement

Top-Down Strategy (cont.)

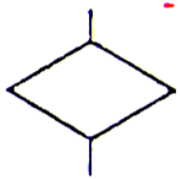
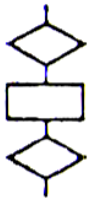
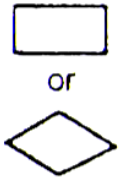
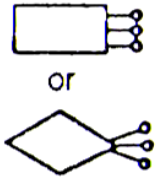
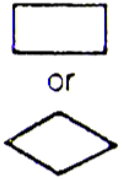
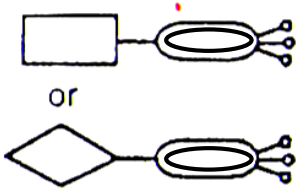
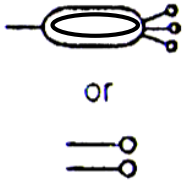


(e) After the 2nd refinement iteration

Top-Down Syntax

Syntactic rules	Starting Schema	Resulting Schema
T_1: Entity → Related entities		
T_2: Entity → Generalization (Entity → Subset)		
T_3: Entity → Uncorrelated entities		
T_4: Relationship → Parallel relationships		

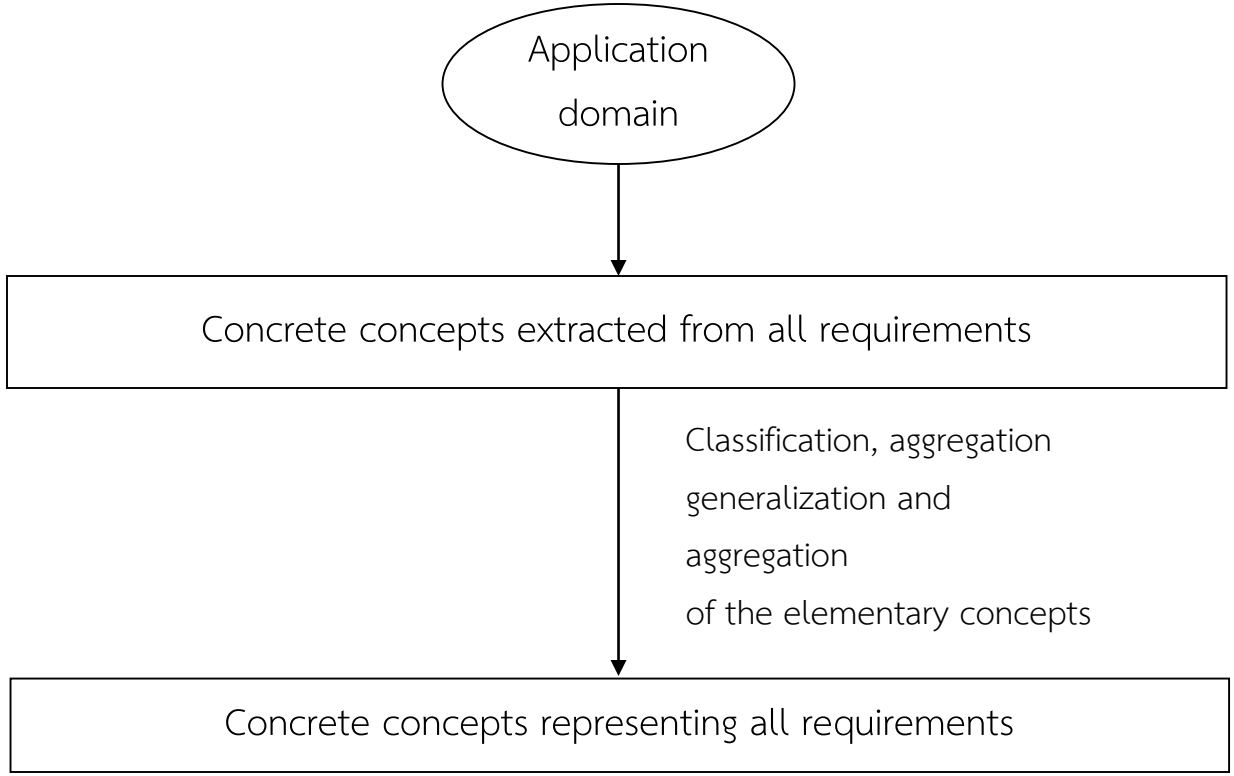
Top-Down Syntax (cont.)

T_5: Relationship → Entity with relationships		
T_6: Attribute development		
T_7: Composite attribute development		
T_8: Attribute refinement		

Bottom-Up Strategy (cont.)

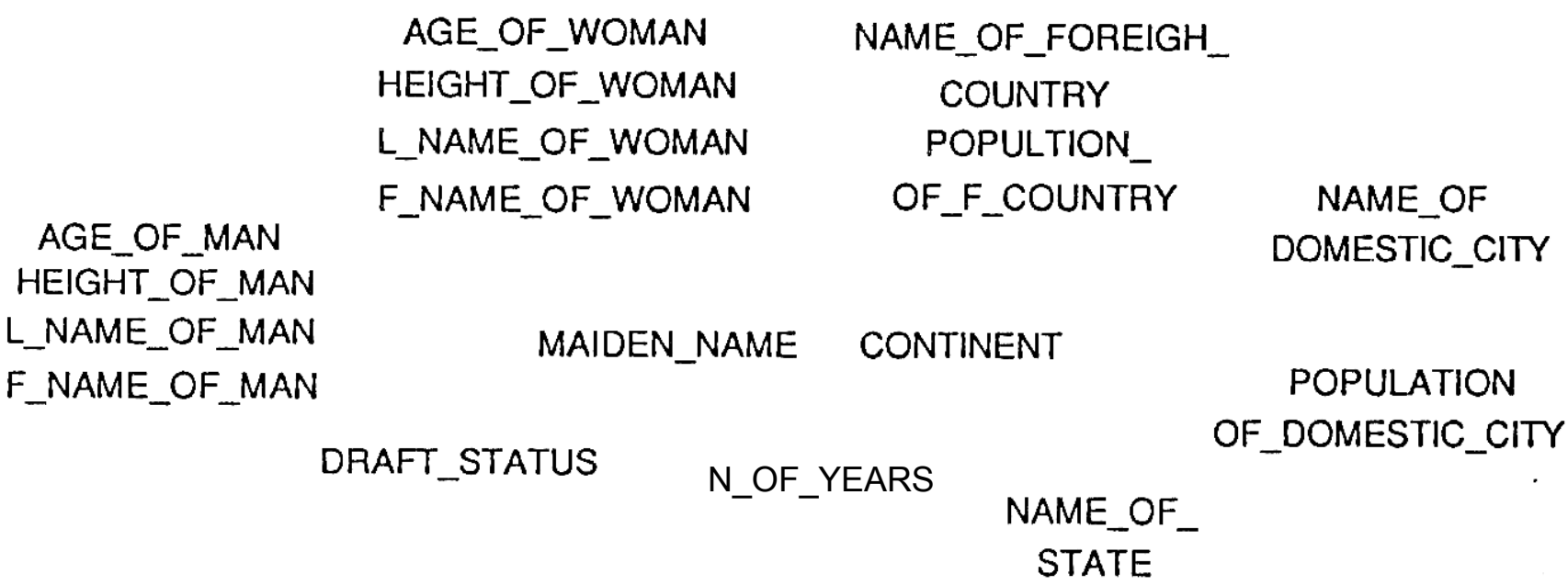
This strategy starts from elementary concepts and builds more general concepts out of them.

The requirements are decomposed, independently conceptualized, and finally merged into a global schema.



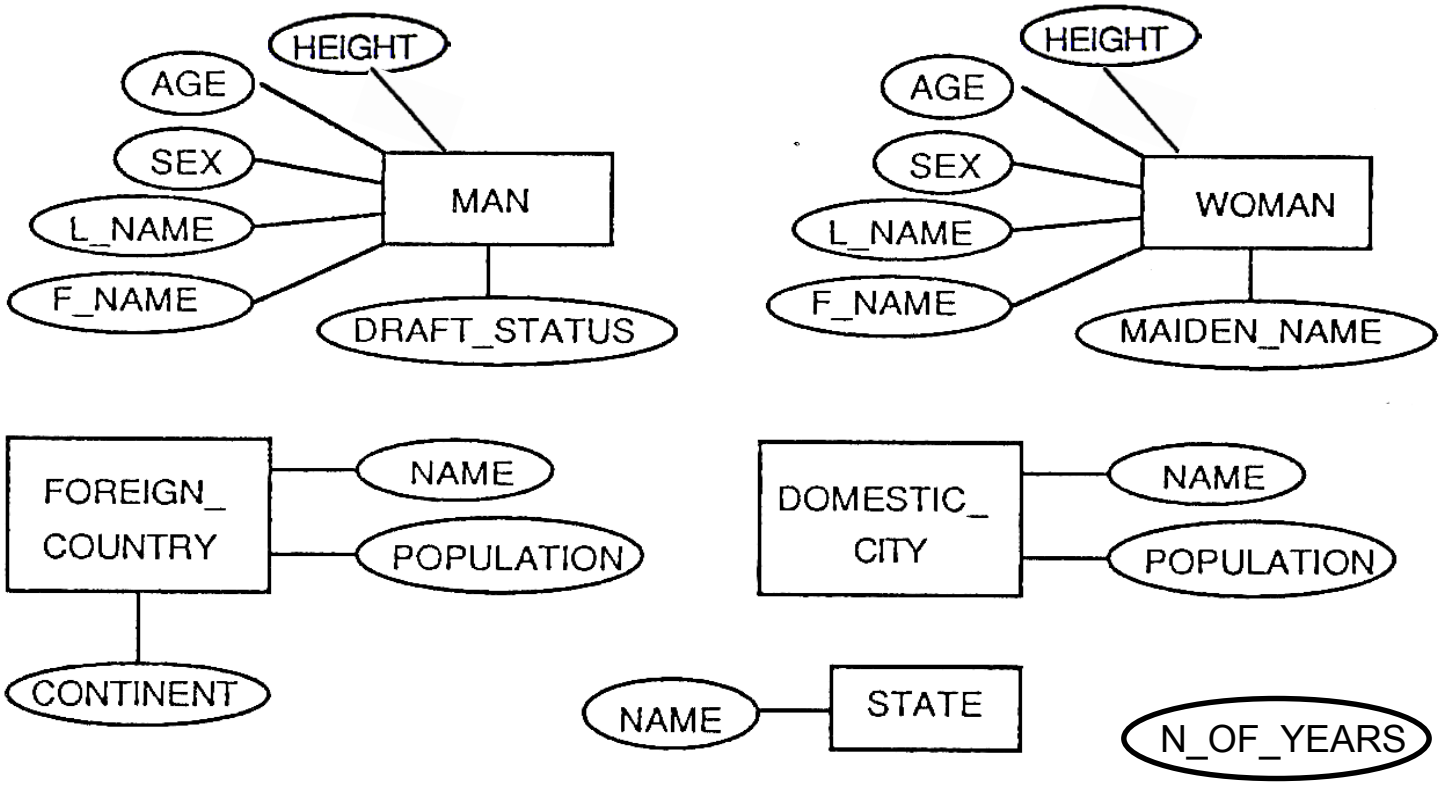
Bottom-Up Strategy (cont.)

- Example



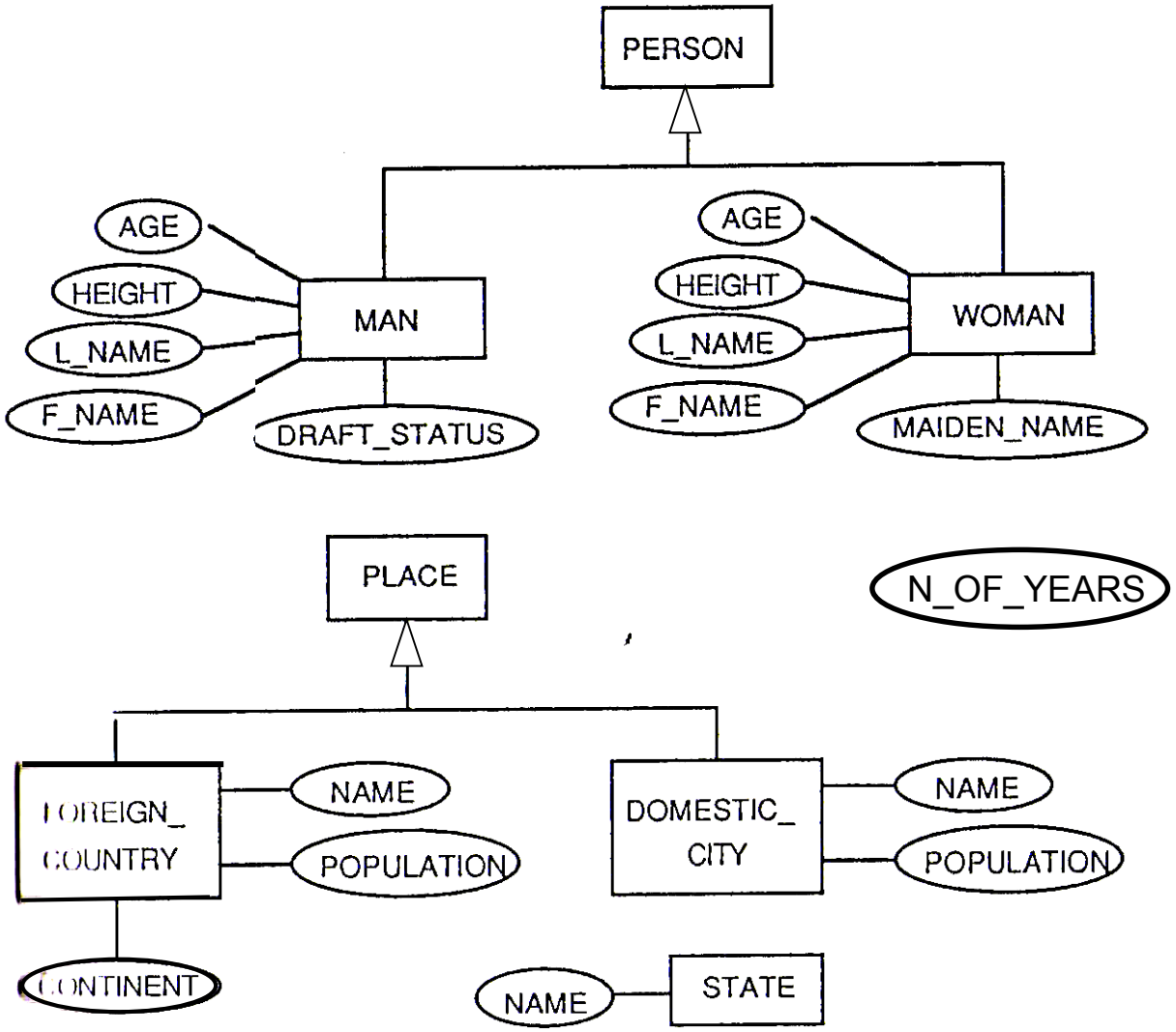
(a) Concrete concepts

Bottom-Up Strategy (cont.)



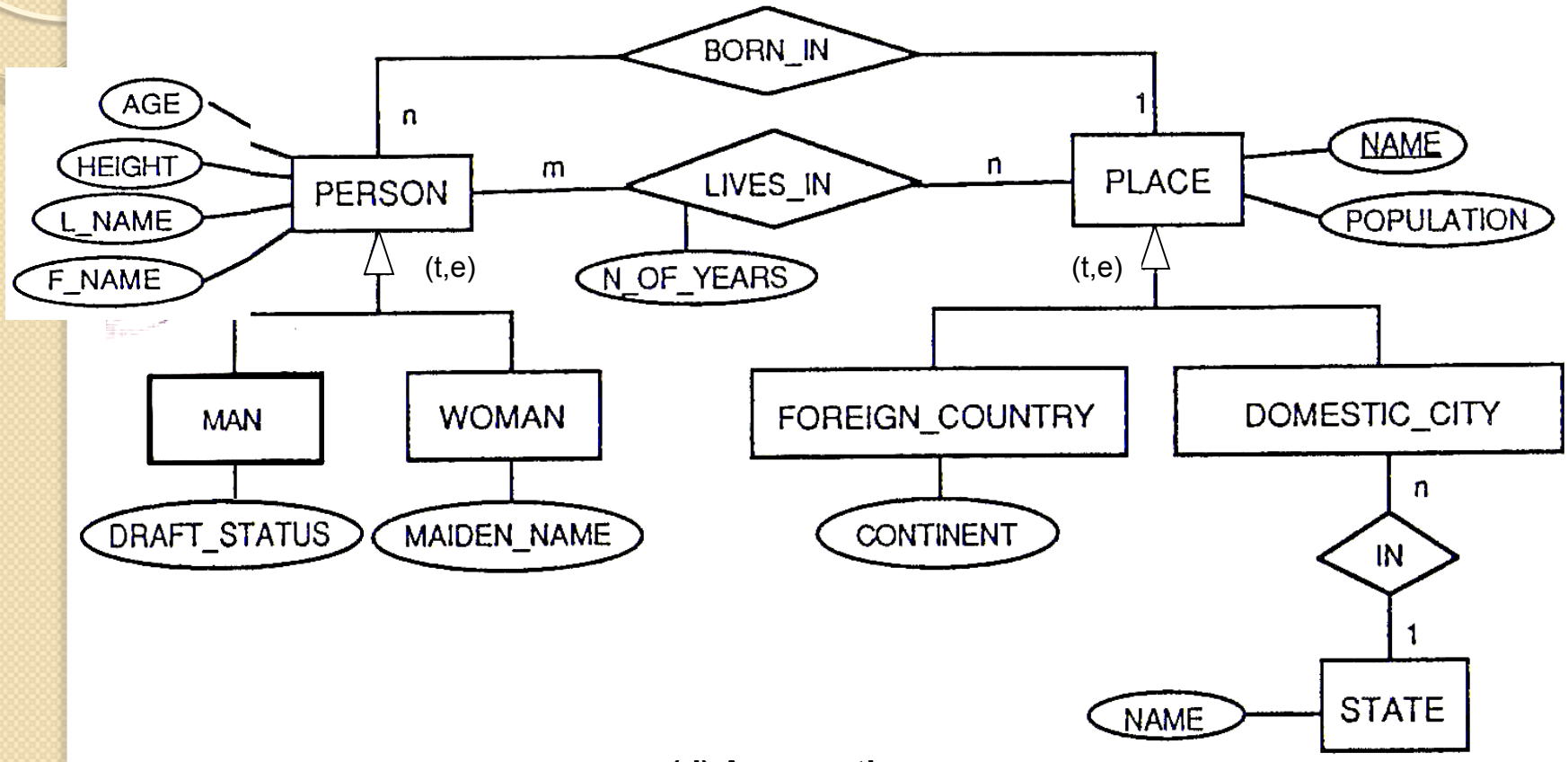
(b) Classification & Aggregation

Bottom-Up Strategy (cont.)



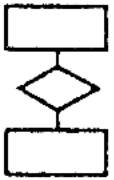
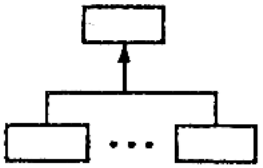
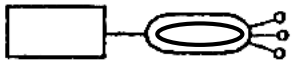
(c) Generalization

Bottom-Up Strategy (cont.)



(d) Aggregation

Bottom-Up Syntax

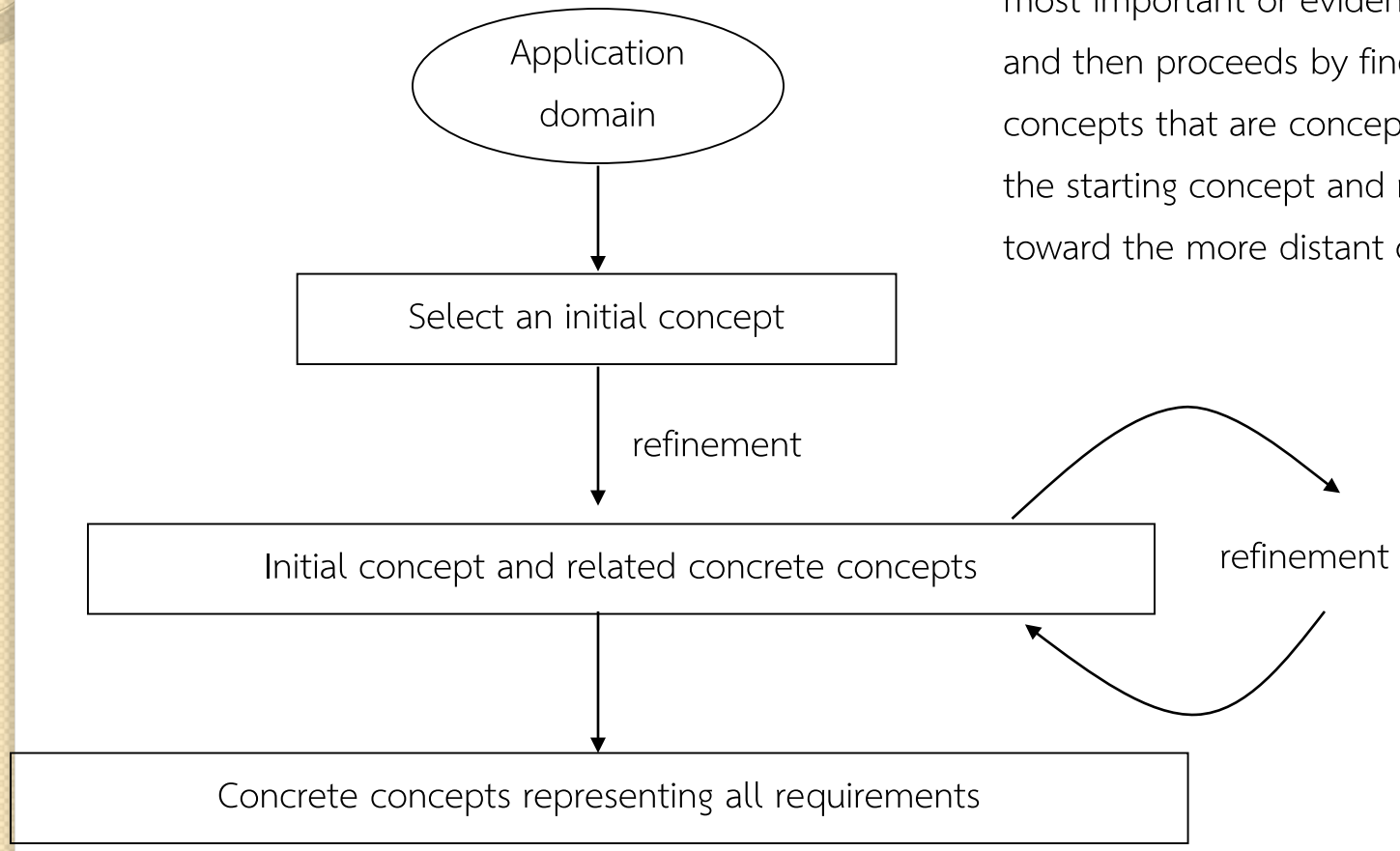
Syntactic rules	Starting Schema	Resulting Schema
B_1: Entity generation	Concrete concepts	
B_2: Relationship generation		
B_3: Generalization generation (subset generation)		
B_4: Attribute aggregation		
B_5: Composite attribute aggregation		

Inside-Out Strategy

- This strategy starts by identifying the most important or evident concepts and then proceeds by finding first the concepts that are conceptually close to the starting concept and navigating toward the more distant ones.

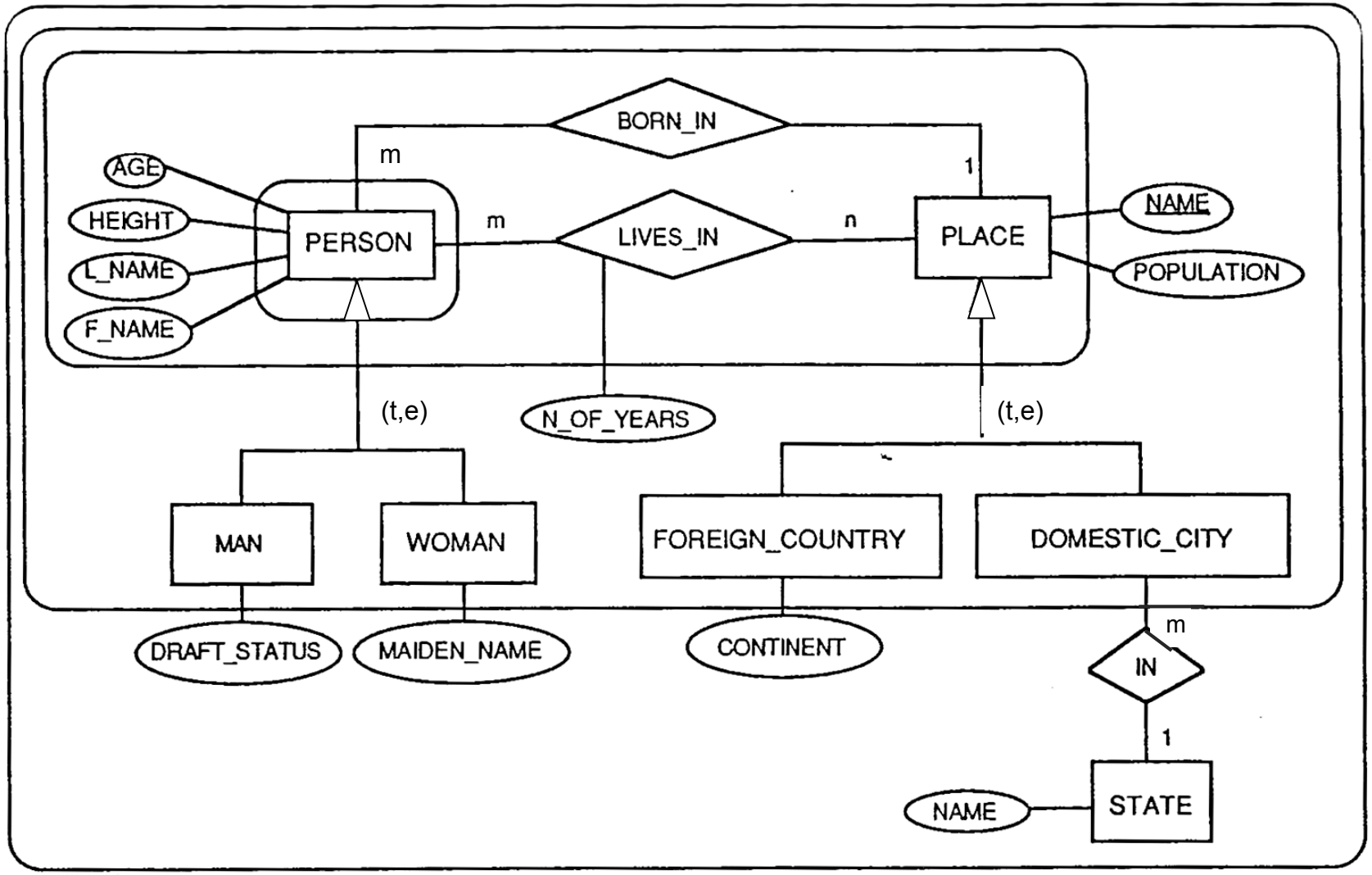
Inside-Out Strategy (cont.)

This strategy starts by identifying the most important or evident concepts and then proceeds by finding first the concepts that are conceptually close to the starting concept and navigating toward the more distant ones.



Inside-Out Strategy (cont.)

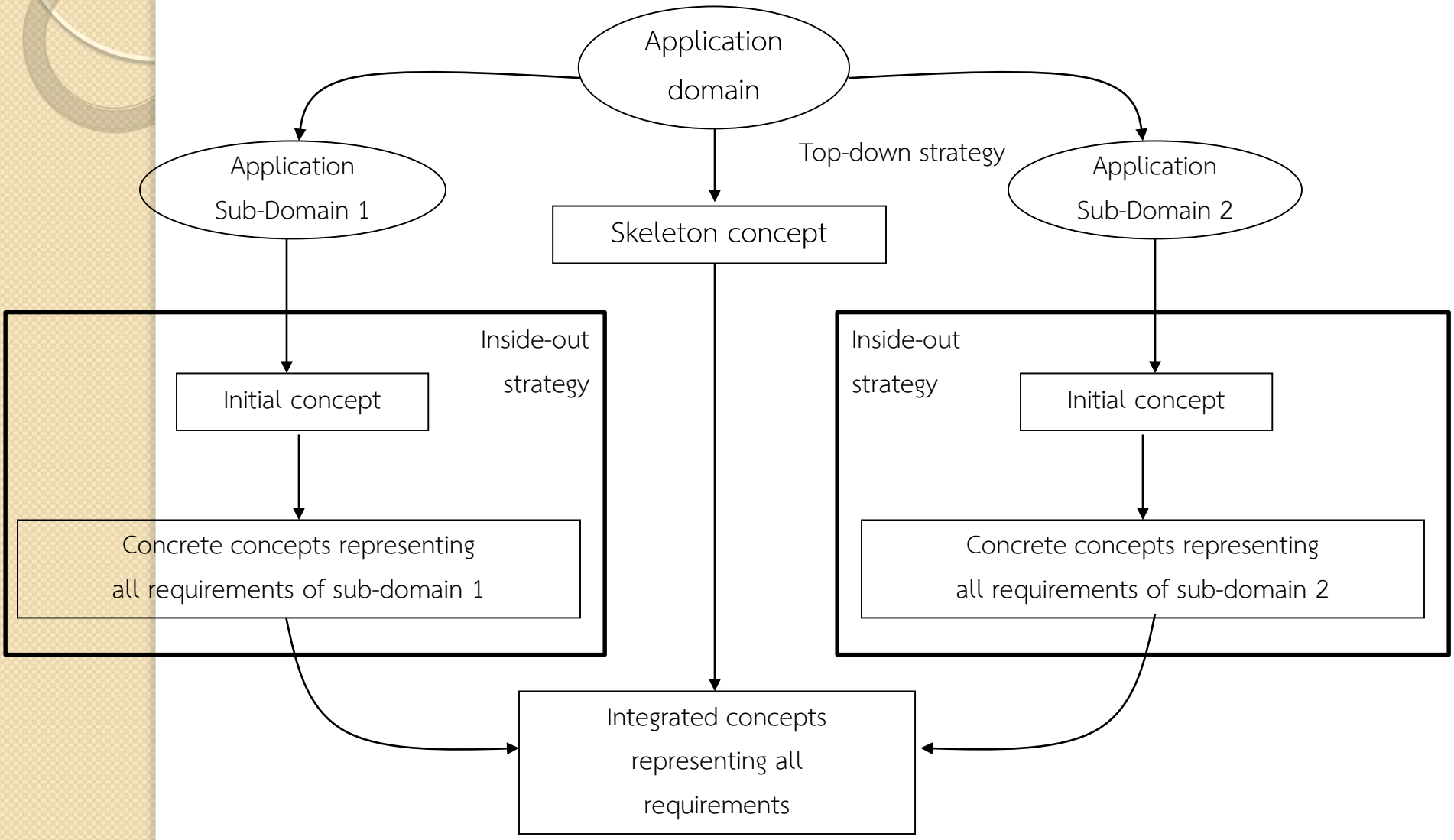
- Example



Mixed Strategy

- This strategy starts by applying the top-down strategy to partition the requirements into subsets and produces a skeleton schema, then applying the inside-out strategy to each subset.
- Each subset of requirements will be considered separately and later linked together based on the skeleton schema via bottom-up strategy.

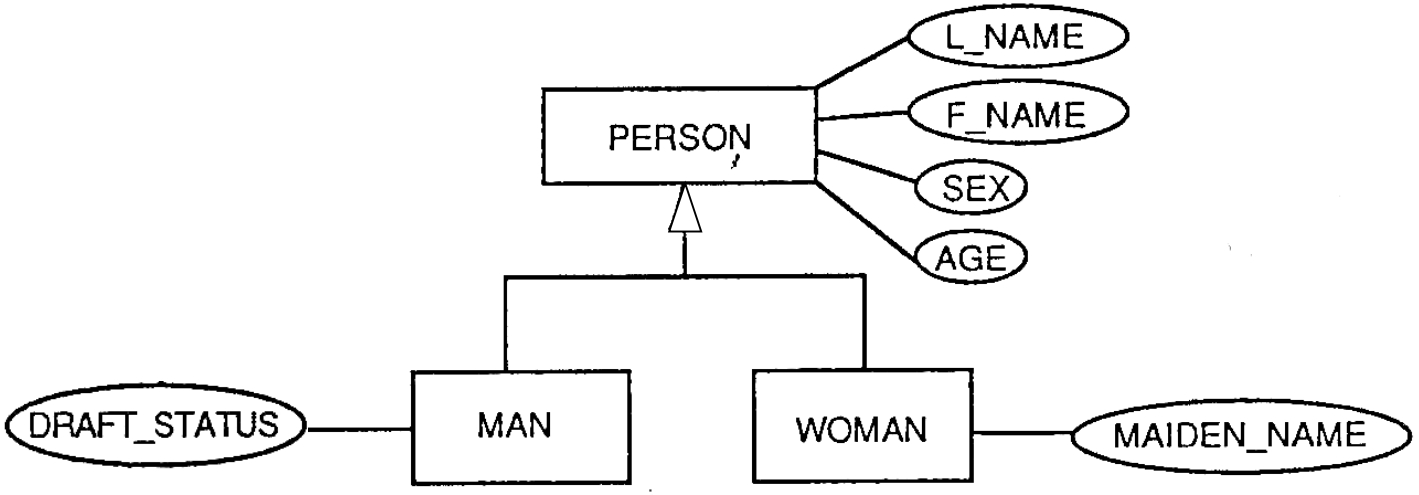
Mixed Strategy (cont.)



Mixed Strategy (cont.)

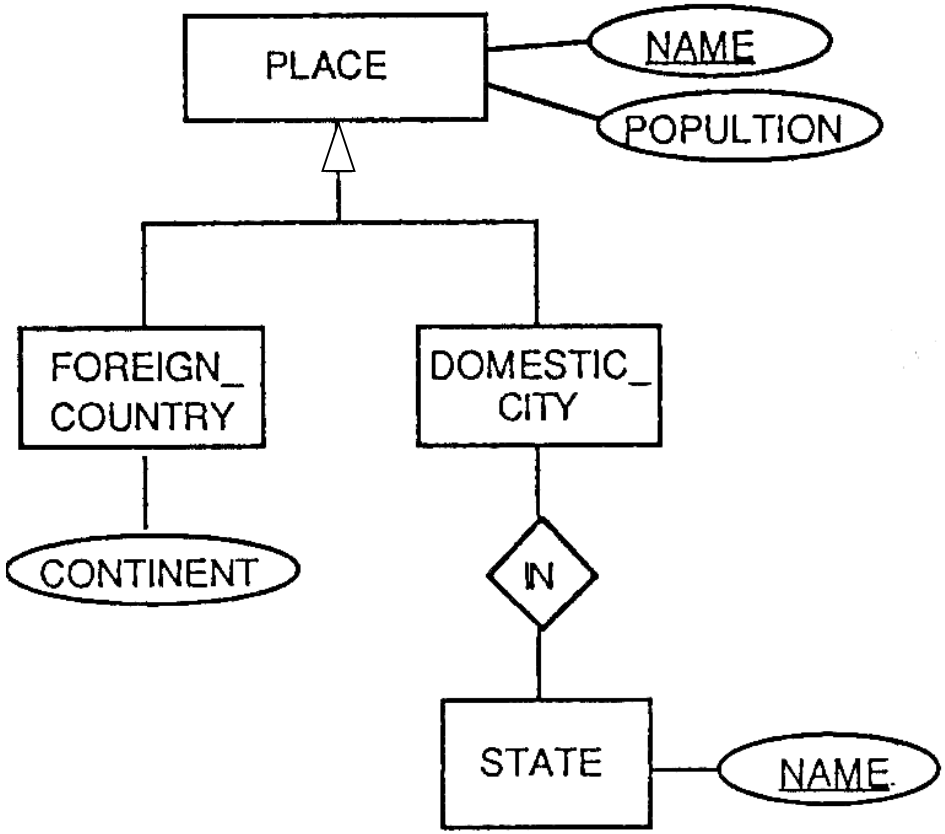
- Example

(a) Skeleton **concepts/model**



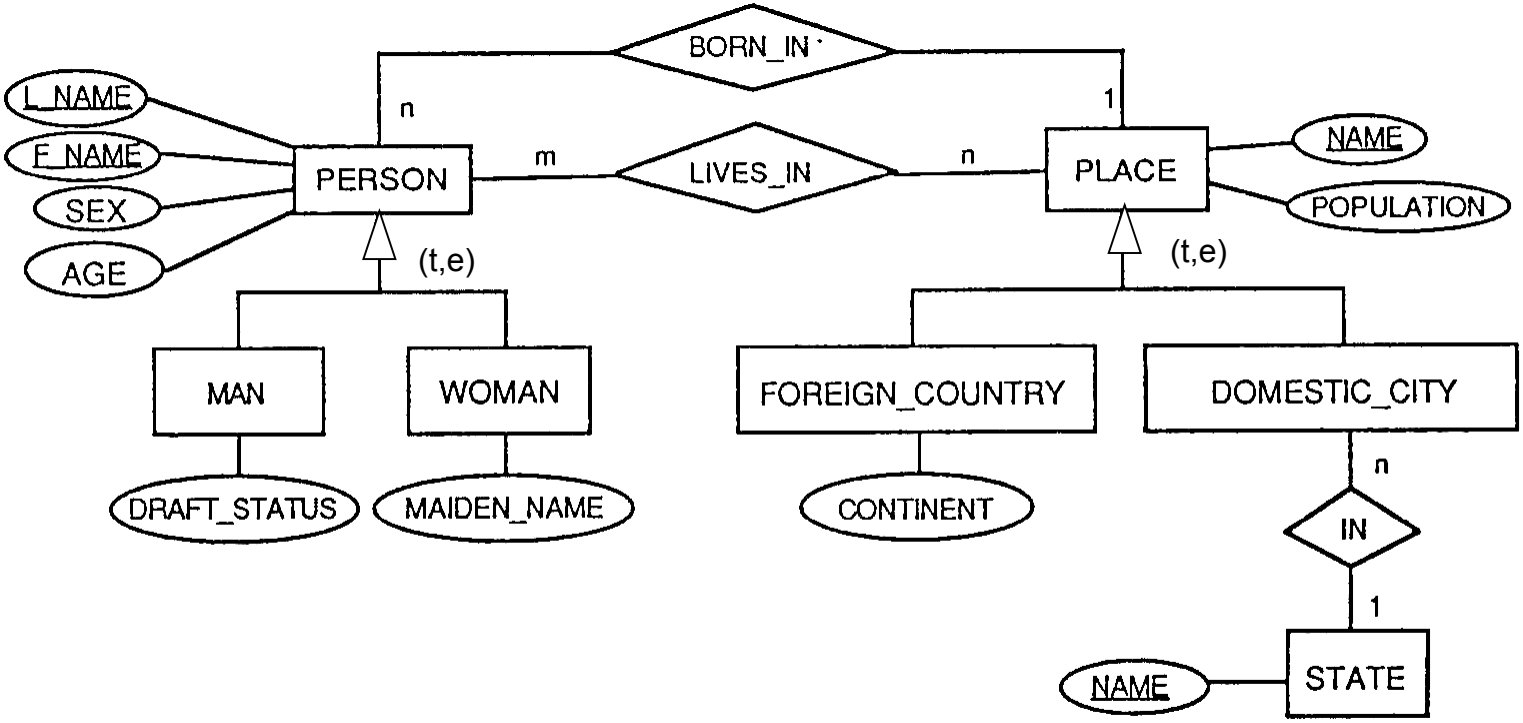
(b) **Model of the 1st sub-domain**

Mixed Strategy (cont.)



(c) **Model of the 2nd sub-domain**

Mixed Strategy (cont.)



(d) Integrated **model**

Comparison of Four Strategies

Strategy	Description	Advantages	Disadvantages
Top-down	Concepts are progressively refined	No undesired side effects	Requires a capable designer with high abstraction ability at the very beginning
Bottom-up	Concepts are built from elementary components	Ease of local design decisions No burden on initial designer	Need of restructuring after applying each bottom-up primitive

Comparison of Four Strategies (cont.)

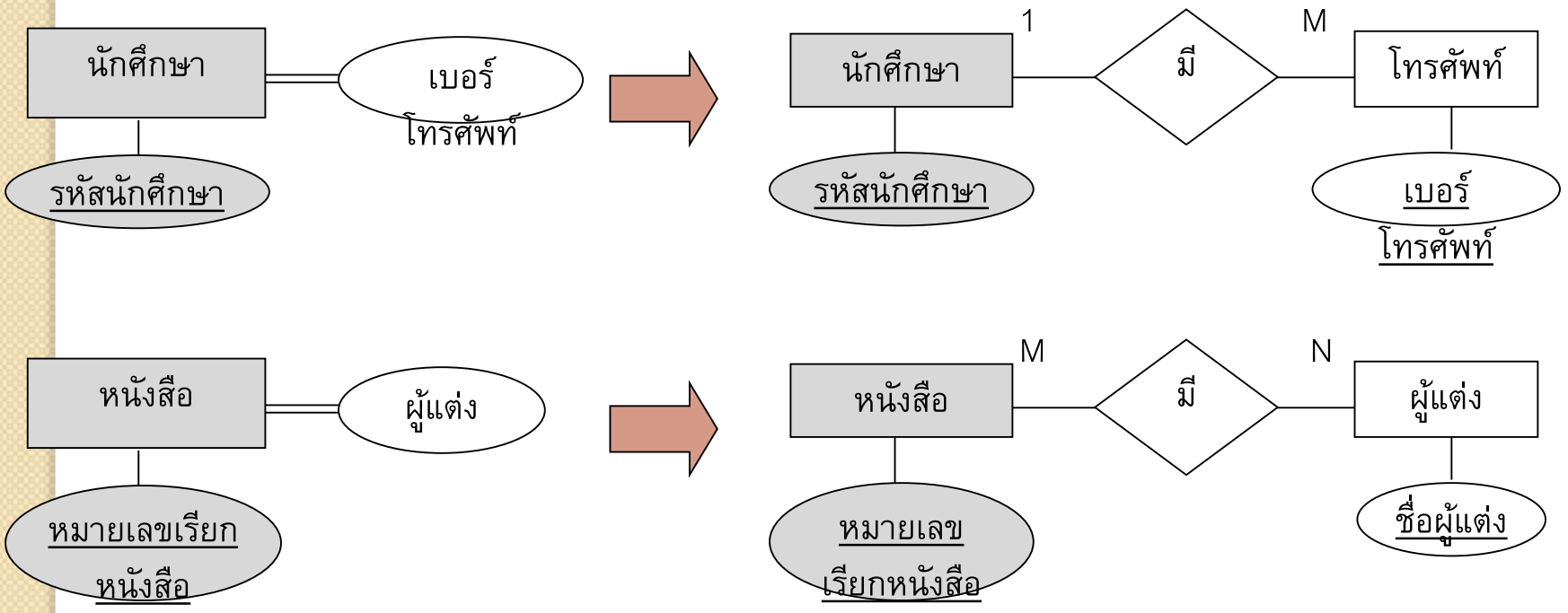
Strategy	Description	Advantages	Disadvantages
Inside-out	Concepts are built with an oil-stain approach	Ease of discovering new concepts close to previous ones	A global view of the application domain is built only at the end
Mixed	Top-down partitioning of requirements, and inside-out to derive each partial schema; bottom-up integration by using a skeleton schema	Divide-and-conquer approach	Requires critical decisions about the skeleton schema at the beginning of the design process

Guideline for ER Model Improvement

- Entity should contain descriptive information in form of nonkey attribute.
  - If an entity has only primary key without nonkey attributes and is only attached to 1-to-1 relationship, the entity should be classified as an attribute rather than an entity.

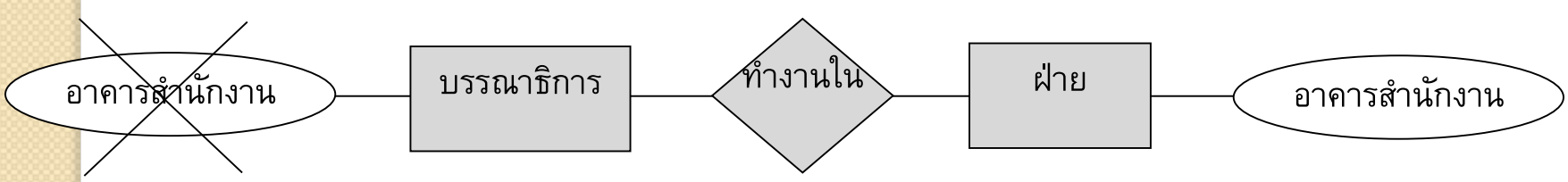
Guideline for ER Model Improvement (cont.)

- Classify multi-valued attribute as an entity along with 1-to-m or m-to-n relationship.



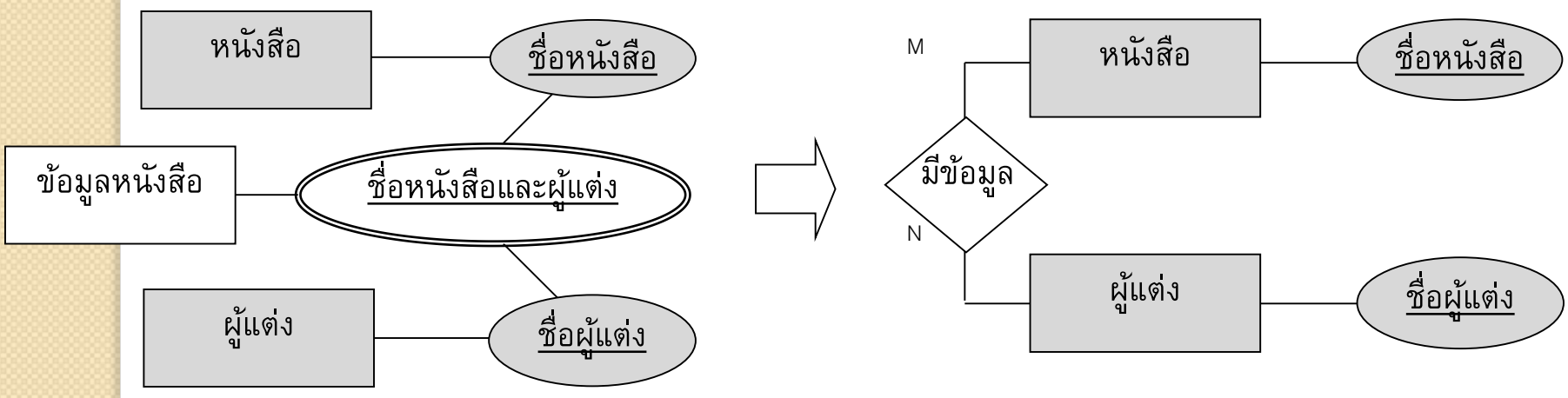
Guideline for ER Model Improvement (cont.)

- Attach attributes to the entities that the attributes describe most directly.



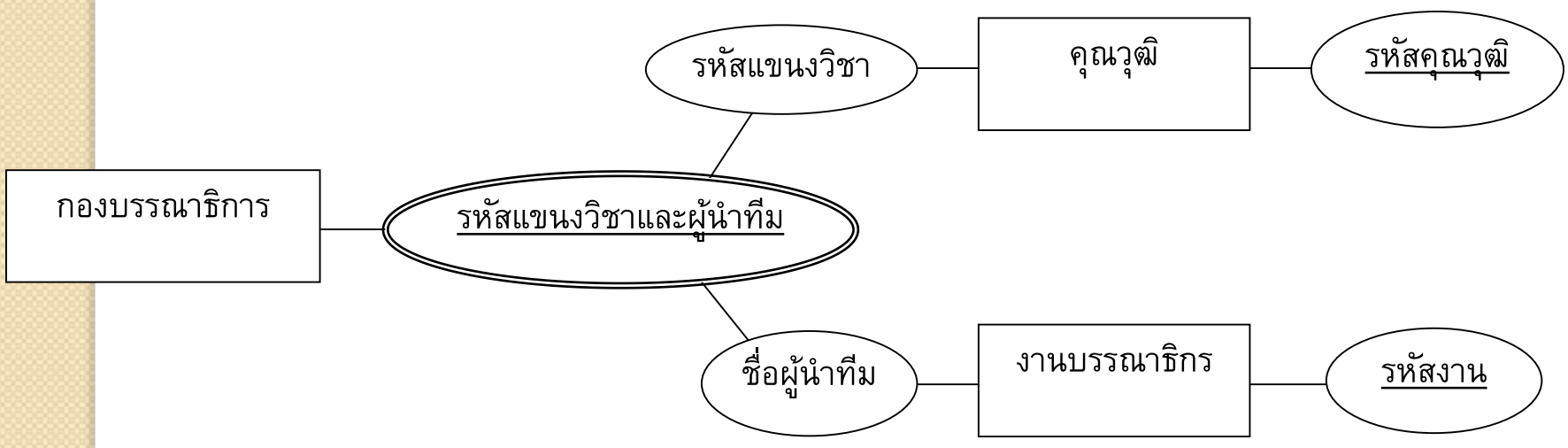
Guideline for ER Model Improvement (cont.)

- Avoid composite pk whenever possible.
  1. If a composite pk only consists of pks of other entities, then define the entity of the composite pk as a relationship.



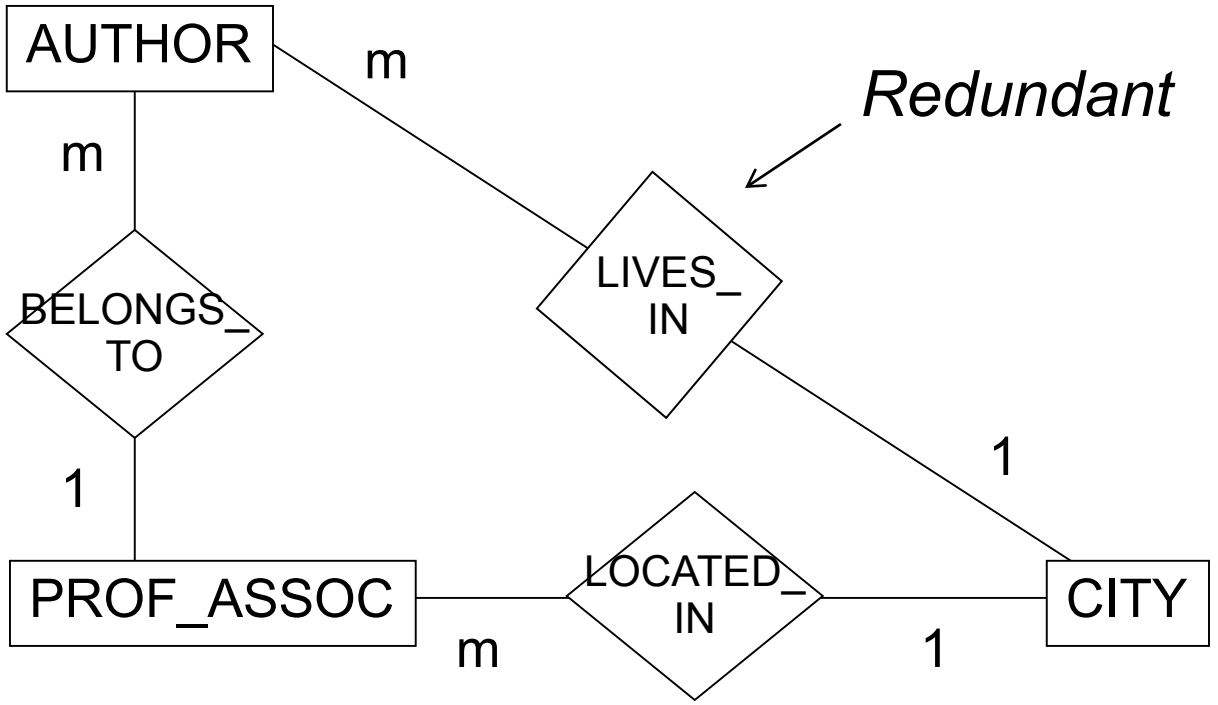
Guideline for ER Model Improvement (cont.)

2. If a composite pk consists of attributes that are not the primary keys of other entities, then take no change on the composite pk.



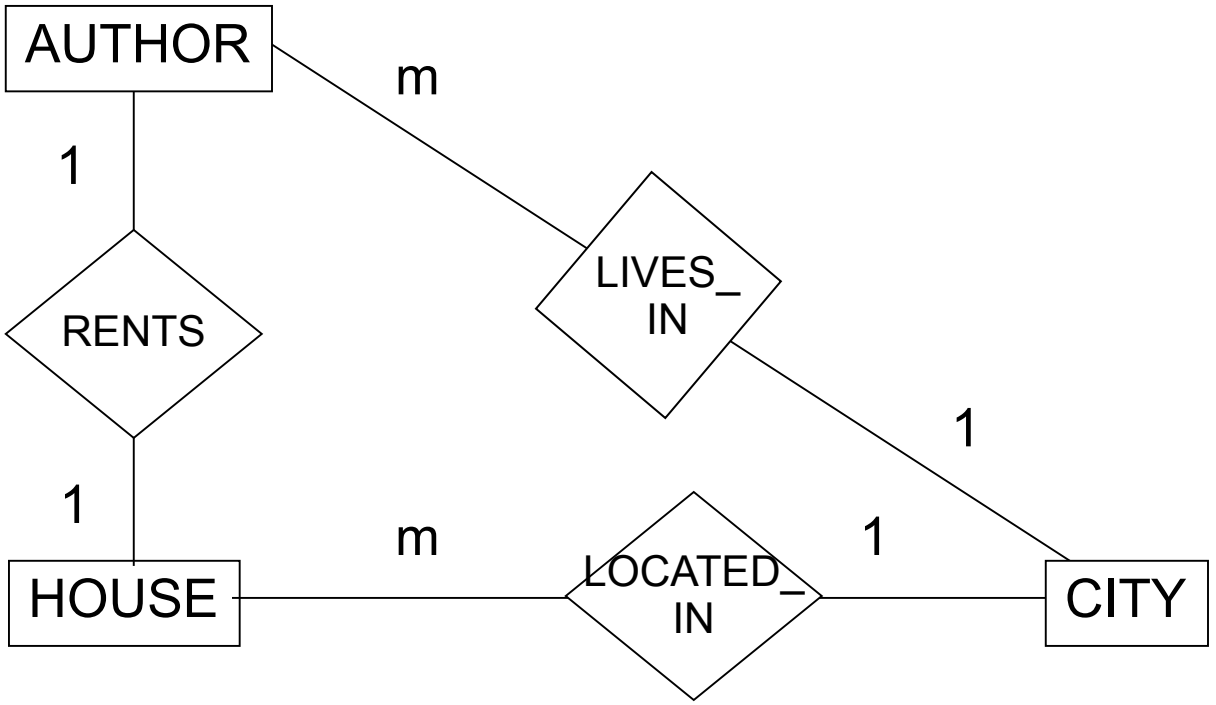
Guideline for ER Model Improvement (cont.)

- Redundant relationships should be eliminated.
- Example 1



Guideline for ER Model Improvement (cont.)

- Example 2



ER Model Integration

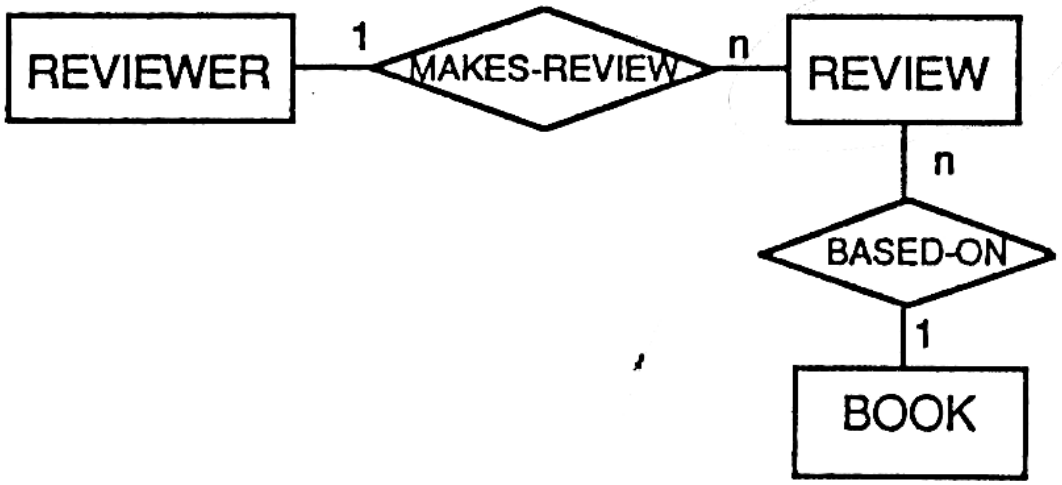
- A process of unifying partial models into a single consistent and non-redundant model.
  - Each partial model may be created by each different team.
- Involves the removal of model diversity, which appears in three forms of conflicts:
  - Naming conflict
  - Structural conflict
  - Behavioral conflict

Naming Conflicts

- Synonym
  - Different names are given for the same concept, e.g. staff VS. employee.
- Homonym
  - The same name is used to represent different concepts, e.g. COMPONENT representing computer component VS. COMPONENT representing chemical component.

Structural Conflicts

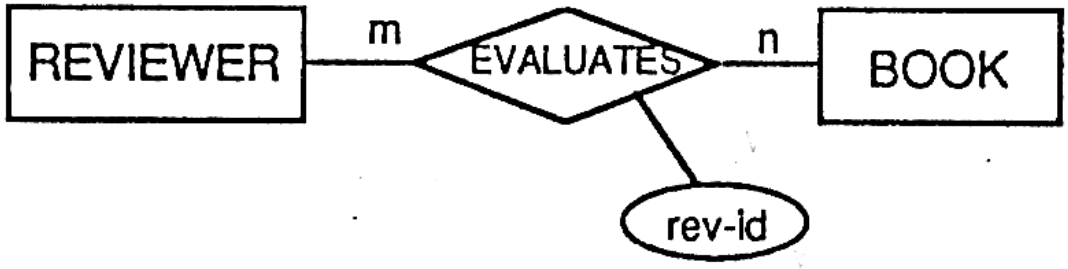
- Type conflict
  - Occurs when using different ER primitive constructs (entities, relationships, attributes) to model the same concept.
  - Examples



(a) The concept of review as an entity

Structural Conflicts (cont.)

(b) The concept of review as a relationship



(c) The concept of review as an attribute

Structural Conflicts (cont.)

- Dependency conflict
  - Occurs when different users specify different connectivity for the same relationship.
  - Example: one team specifies 1 to 1 whereas another 1 to many for the same binary relationship.
- Primary-key conflict
  - Occurs when different primary keys are assigned to the same entity.
  - Example: student_ID VERSUS citizen_ID for the same entity STUDENT.

Behavioral Conflict

- Occurs when using different integrity constraints for the same attribute.
- Example
  - One team specifies that GPA for retirement is below 3.00, whereas another team is below 2.50.

Model Integration Steps

1. Decide integration approach.
2. Resolve conflicts.
3. Merge models.
4. Eliminate redundancy.

1. Decide Integration Approach

- Binary approach
  - Two models are merged at one time.
  - Simple but probably slow.
- N-ary approach
  - N models are merged at one time.
  - Fast at a price of difficulty to conduct.

2. Resolve Conflicts

- Naming conflict is solved by renaming.
  - Synonym can be detected by scanning data dictionary (which contains definitions of names used in partial models).
  - Homonym can also be detected by scanning data dictionaries.

2. Resolve Conflicts (cont.)

- Structural conflicts
  - Type conflict is resolved by converting conflict primitive constructs to the same type.
  - Dependency conflict is resolved by choosing the connectivity with greater cardinality. If not semantically correct, rename the entities so that each connectivity involves different set of entities.
  - Key conflict is resolved by choosing the same primary key.

2. Resolve Conflicts (cont.)

- Behavioral conflict is resolved by using the weakest integrity constraint.
  - In the former example, the constraint specifying retirement GPA below 2.50 is used.

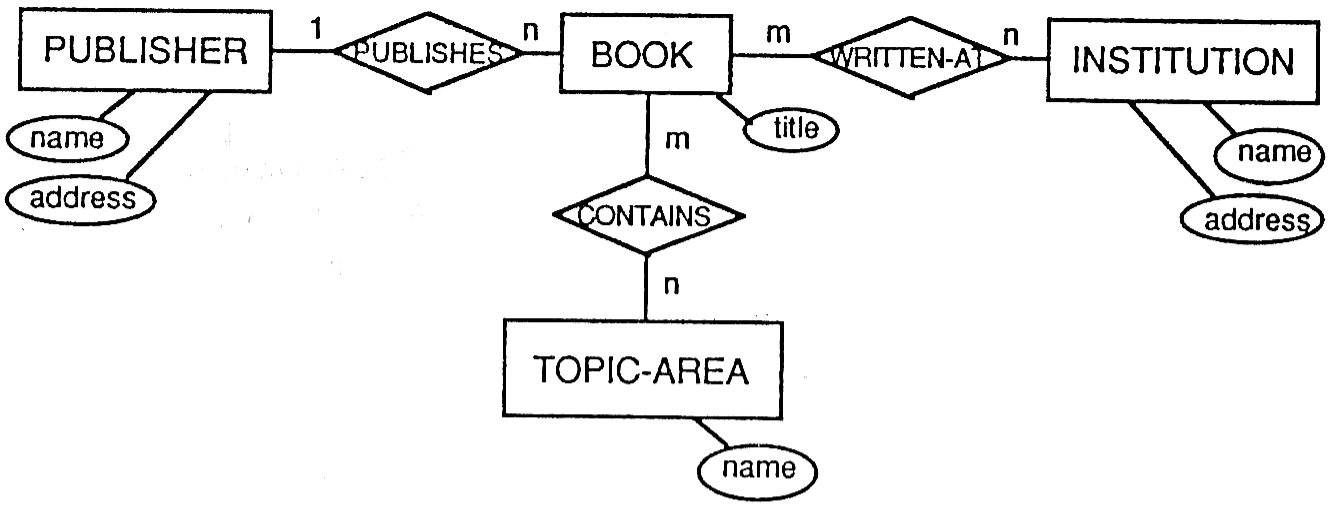
3. Merge Models

- Taking the same primitive constructs as the points of merging.
- After merging, restructuring the unified model by adding generalizations as necessary.

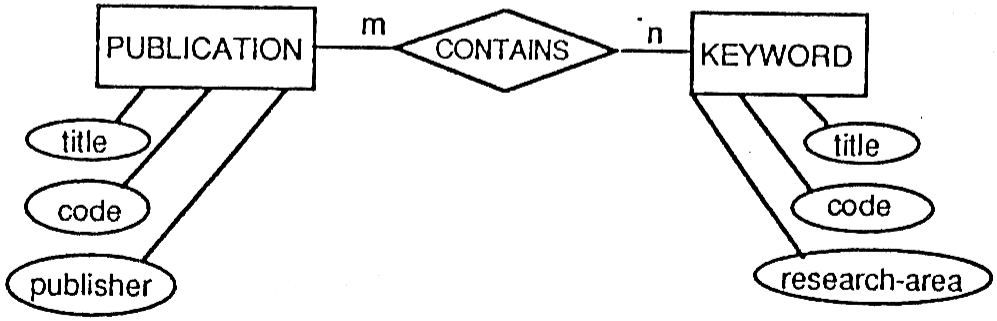
4. Eliminate Redundancy

- Remove all redundant concepts from the merged model.

A Case Study of Model Integration

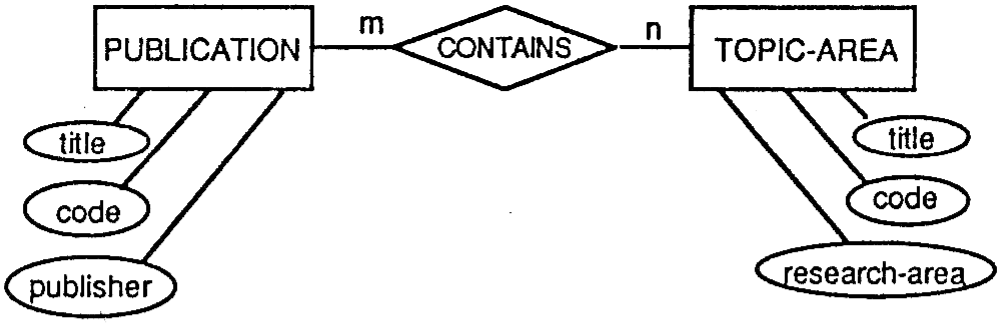


(a) Original model 1, focused on books

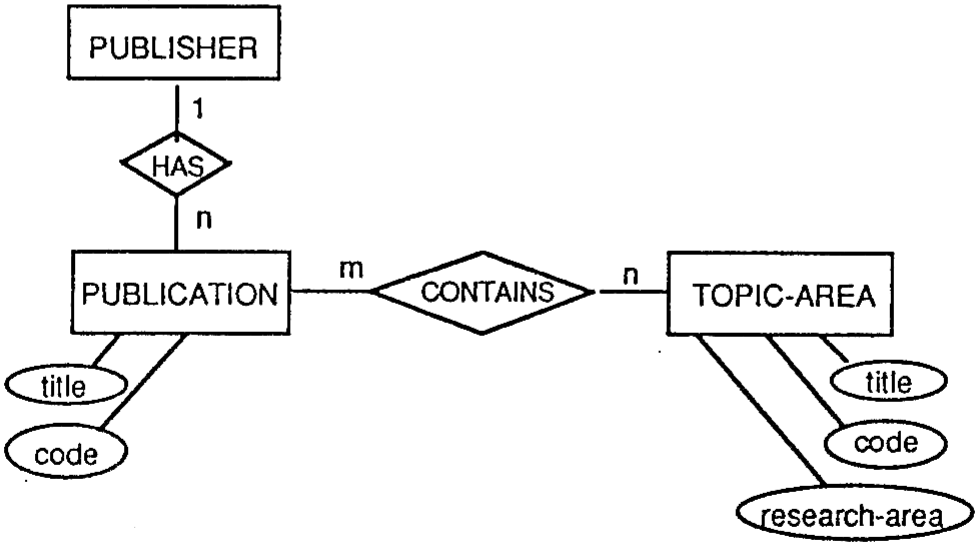


(b) Original model 2, focused on publications

A Case Study of Model Integration (cont.)

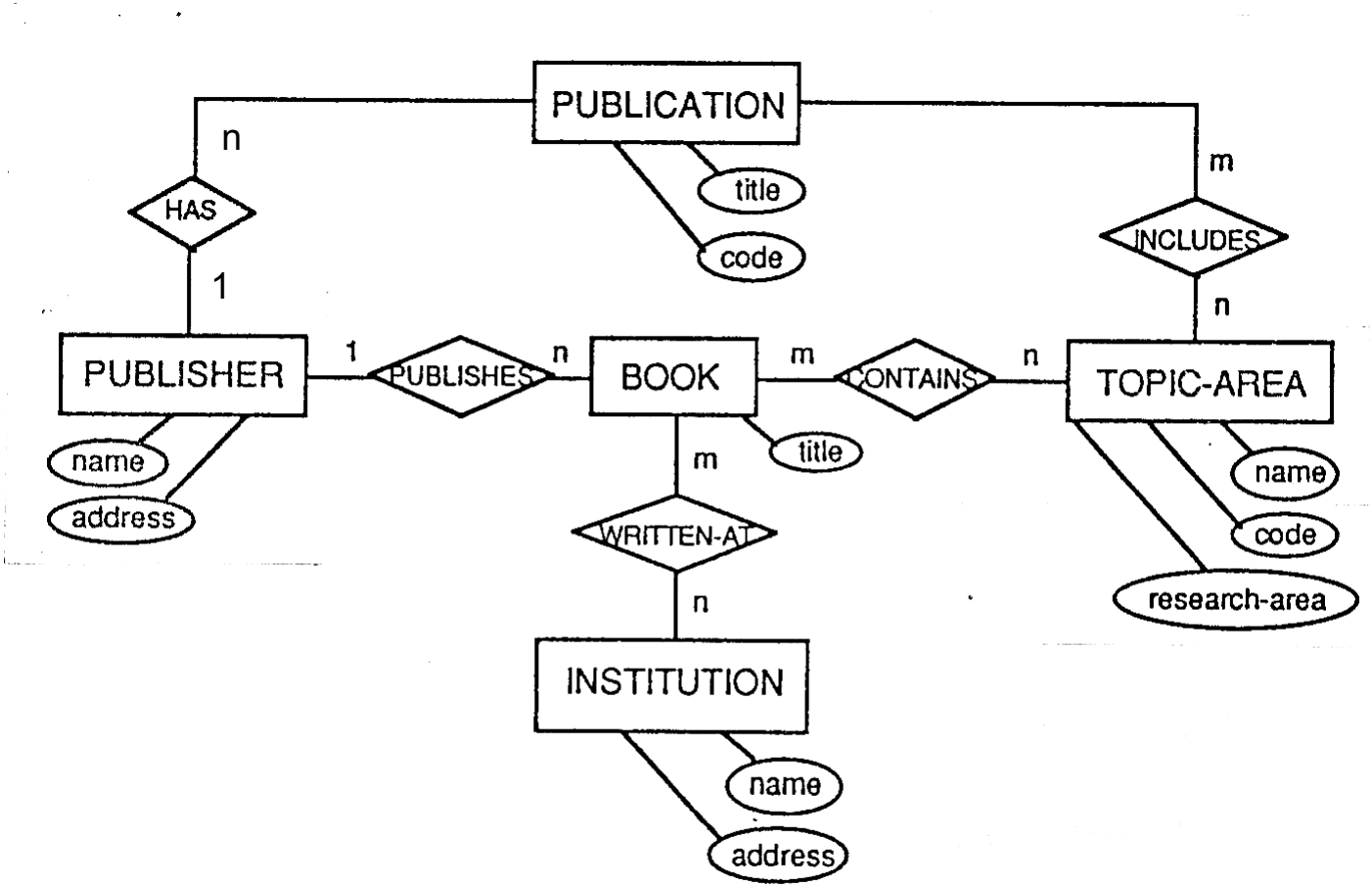


(c) Model 2.1, in which Keyword has changed to Topic-area



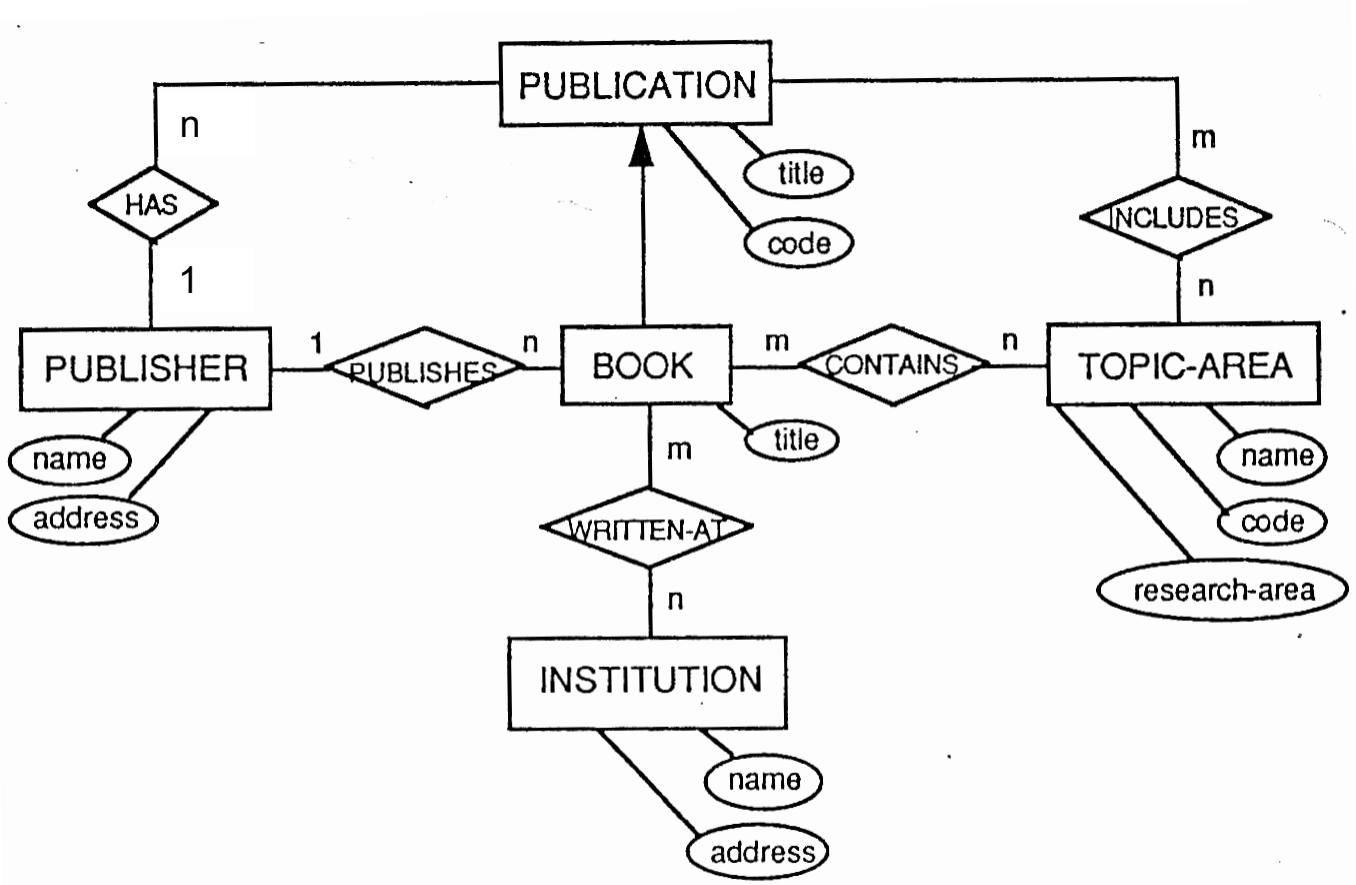
(d) Model 2.2, in which the attribute publisher has changed to an PUBLISHER entity

A Case Study of Model Integration (cont.)



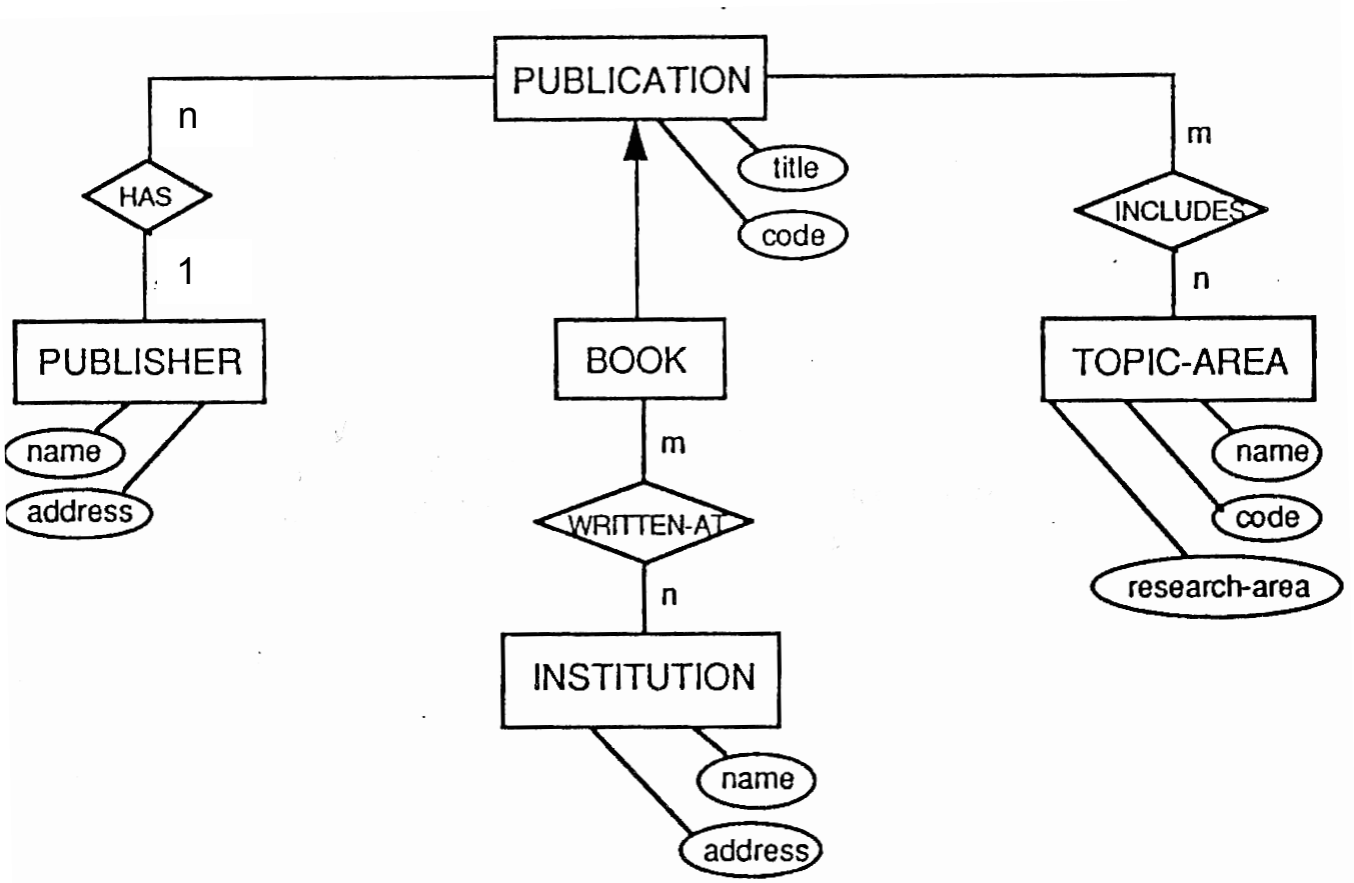
(e) Model 3, the result of merging model 1 and model 2.2

A Case Study of Model Integration (cont.)



(f) Model 3.1 : creation of a generalization relationship

A Case Study of Model Integration (cont.)



(g) Model 3.2 : eliminating of redundancy

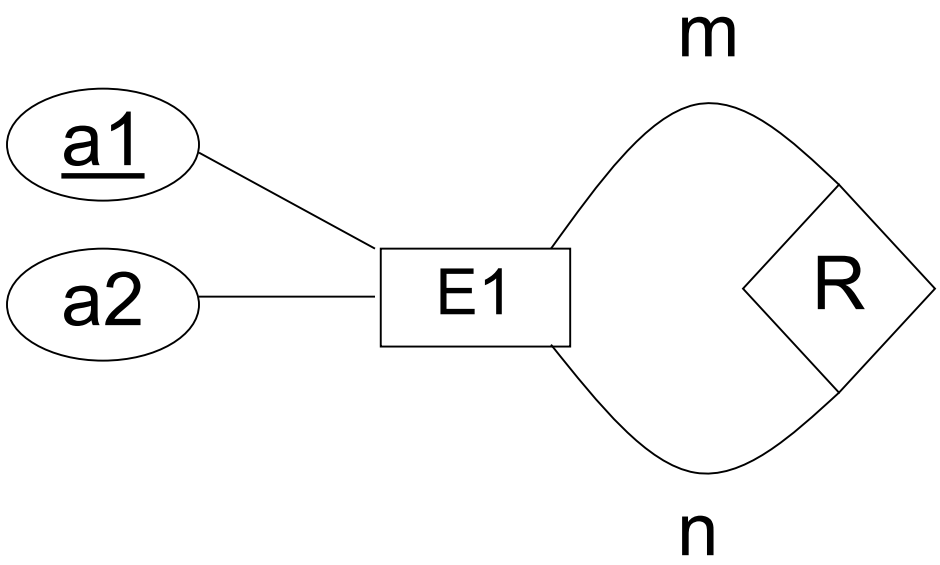
ER model-to-Conceptual Schema Transformation

- A process of transforming ER model to a set of relations.
- Achieved in three steps:
  1. Create entity relation(s)
  2. Create entity relation(s) with foreign key(s)
  3. Create relationship relation(s)

1. Entity Relations

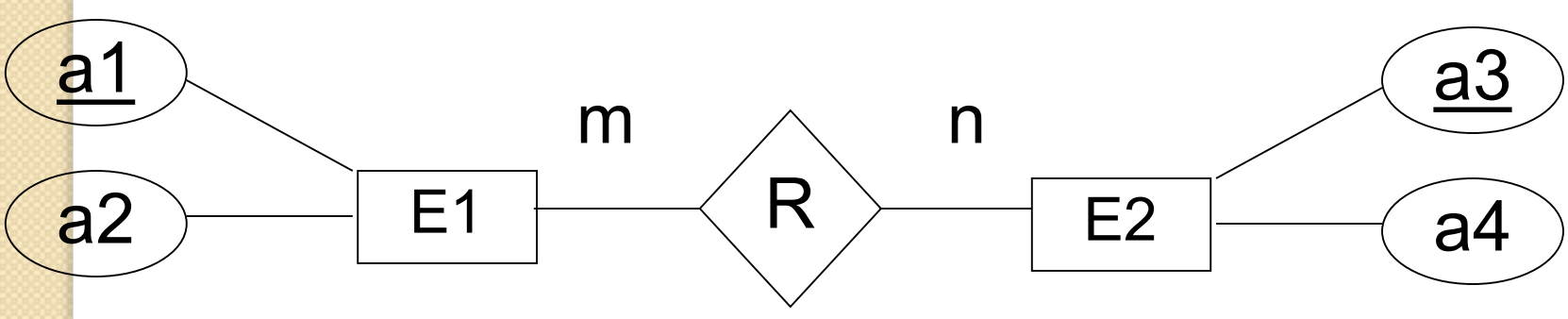
- Create a relation for each of the following entities :
  - Entity of m-to-n unary relationship
  - Entities of m-to-n binary relationship
  - Entity on the 1 side of 1-to-m binary relationship
  - Entities of ternary or n-ary relationship
  - Inherited entities of generalization

Entity of M-to-N Unary Relationship



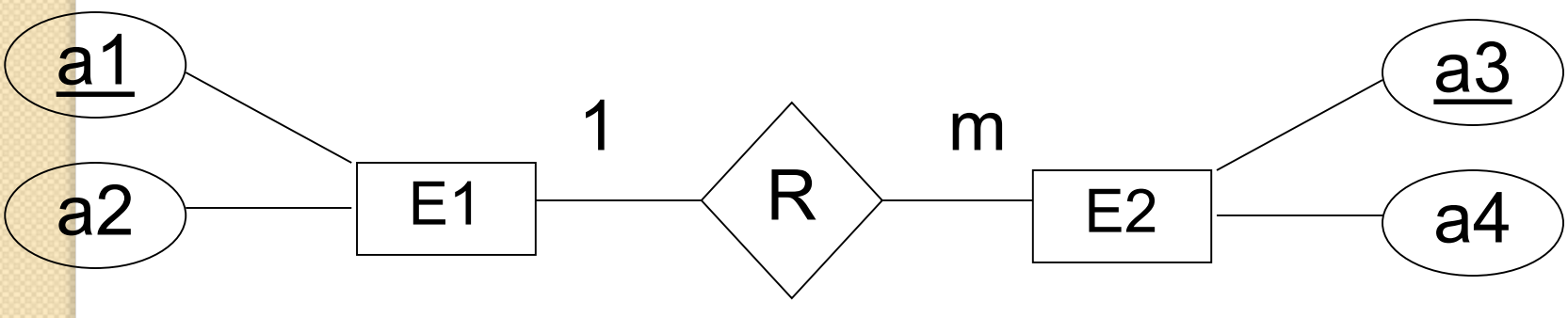
- E1(<u>a1</u>,a2)

Entities of M-to-N Binary Relationship



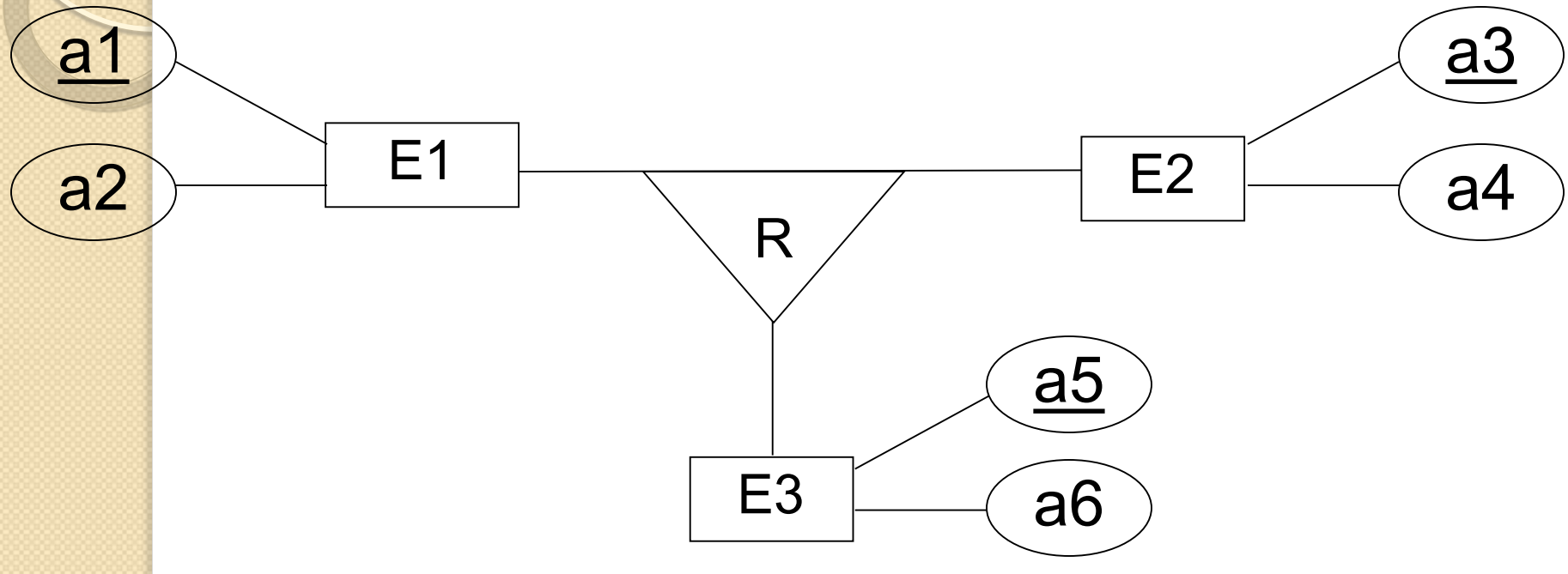
- E1(<u>a1</u>,a2), E2(<u>a3</u>,a4)

Entity on The 1 Side of 1-to-M Binary Relationship



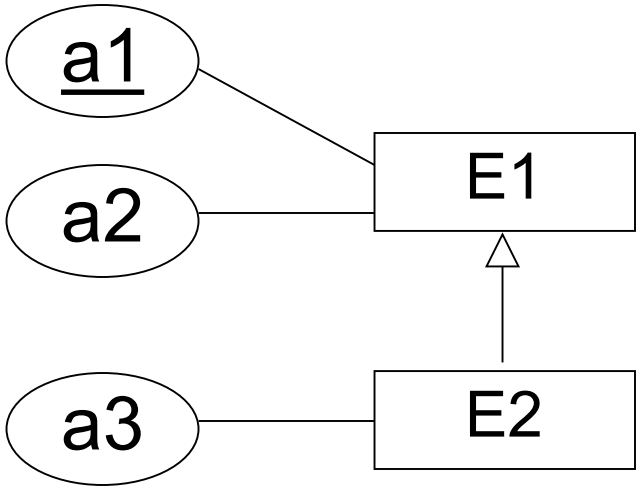
- E1(<u>a1</u>,a2)

Entities of Ternary/N-Ary Relationship



- E1(a1,a2), E2(a3,a4), E3(a5,a6)
- Remark: ignore the connectivity

Generalization

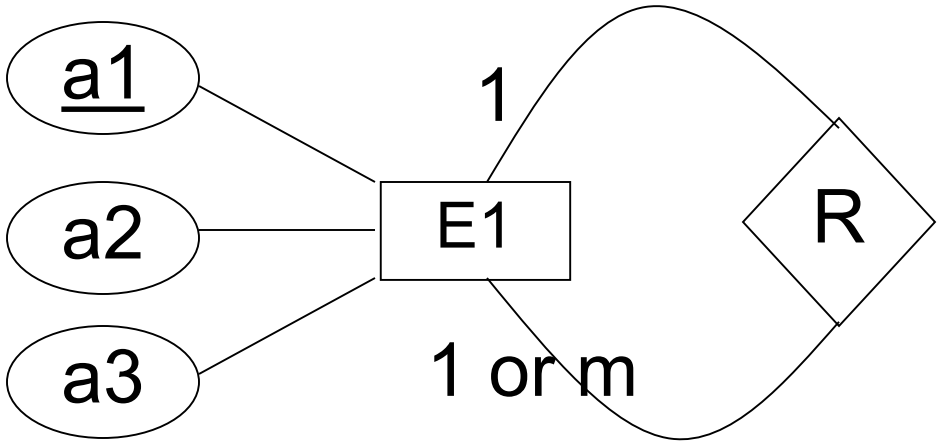


- E1(<u>a1</u>,a2)

2. Entity Relations with Foreign Keys

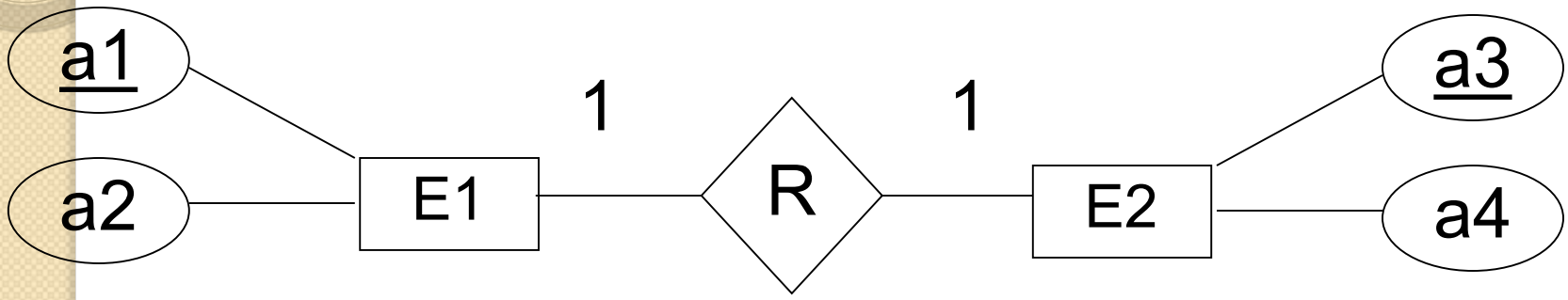
- Create a relation with foreign key for each of the following entities :
  - Entity of 1-to-1 or 1-to-m unary relationship
  - Entities of 1-to-1 binary relationship
  - Entity on the many side of 1-to-m binary relationship
  - Inheriting entities of generalization

Entity of 1-to-1 or 1-to-M Unary Relationship



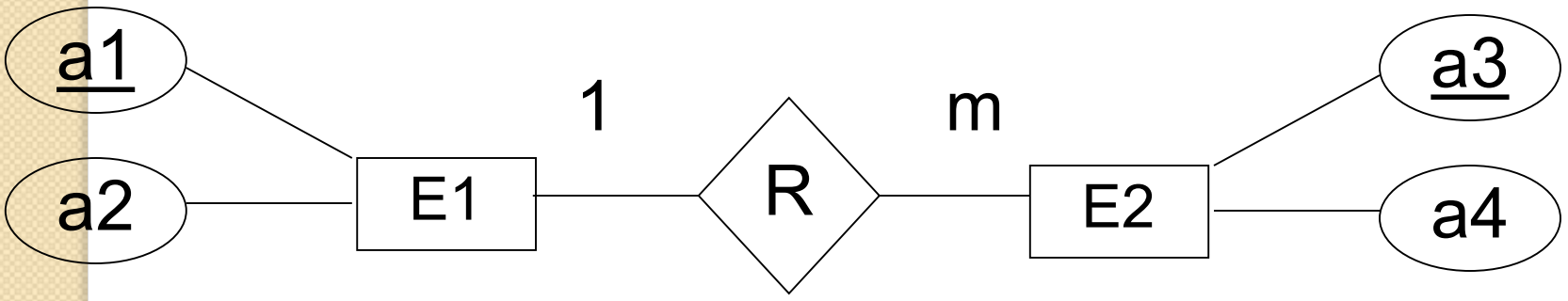
- E1(<u>a1</u>,a2,a3,**a1**)

Two Entities of 1-to-1 Binary Relationship



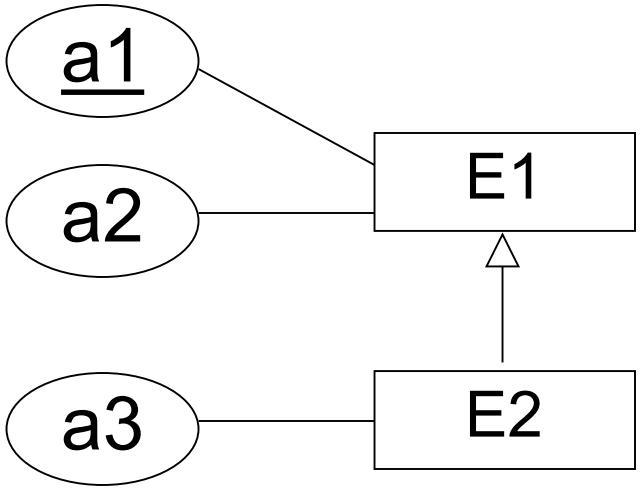
1. E1(a1,a2,**a3**), E2(a3,a4) if E1 is optional
2. E1(a1,a2), E2(a3,a4,**a1**) if E2 is optional
3. E1(a1,a2,**a3**), E2(a3,a4,**a1**) if both optional
4. Either 1 or 2 if both are mandatory

Entity on The Many Side of 1-to-M Binary Relationship



- E2(a3,a4,**a1**)

Generalization

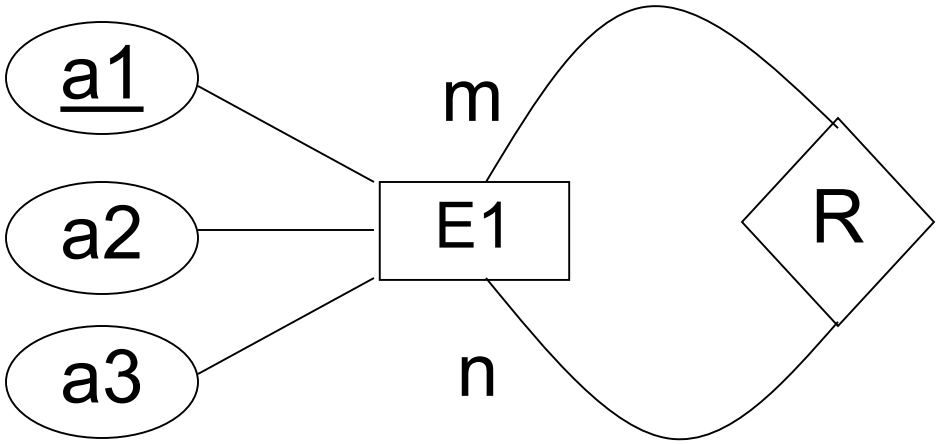


- E2(<u>a1</u>,a3)

3. Relationship Relations

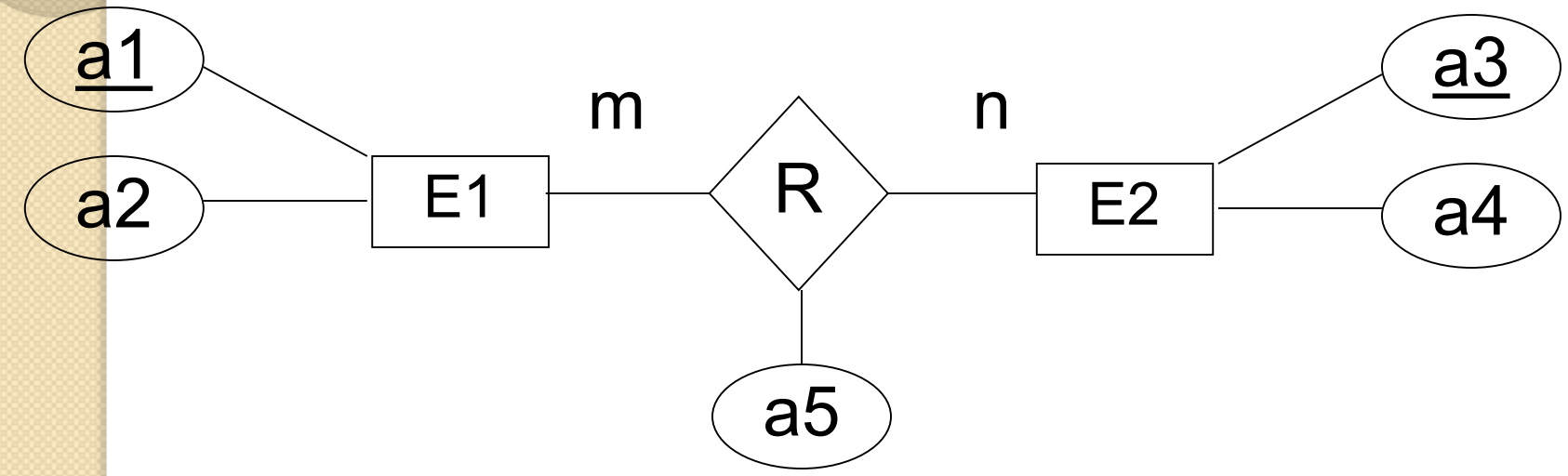
- Create a relation for each of :
  - m-to-n unary relationship
  - m-to-n binary relationship
  - Ternary or n-ary relationship

M-to-N Unary Relationship



- R(<u>a1,a1</u>)

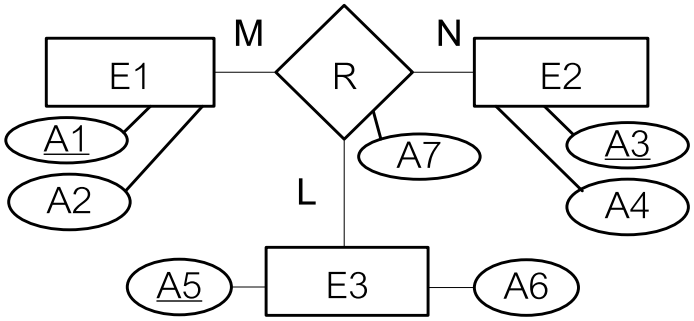
M-to-N Binary Relationship



- R(<u>a1</u>,<u>a3</u>,a5)

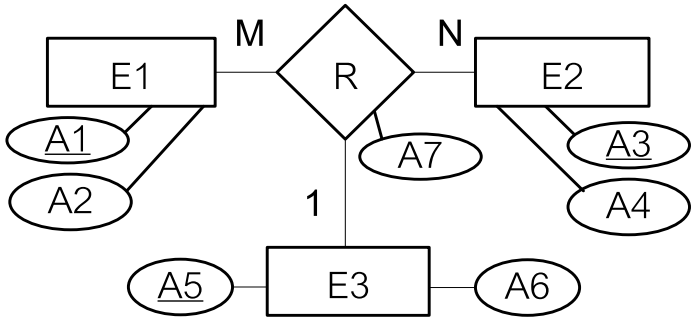
Ternary/N-ary Relationship

ภาวะการเชื่อมต่อแบบมากต่อมากต่อมาก



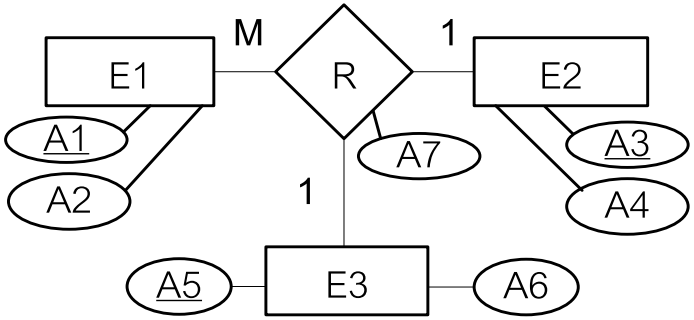
R(A1, A3, A5, A7)

ภาวะการเชื่อมต่อแบบมากต่อมากต่อหนึ่ง



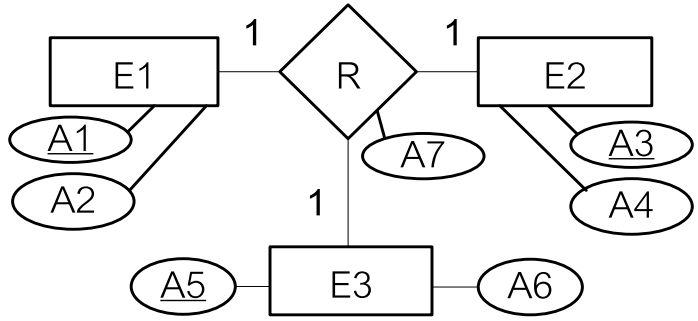
R(A1, A3, A5, A7)

ภาวะการเชื่อมต่อแบบมากต่อหนึ่งต่อหนึ่ง



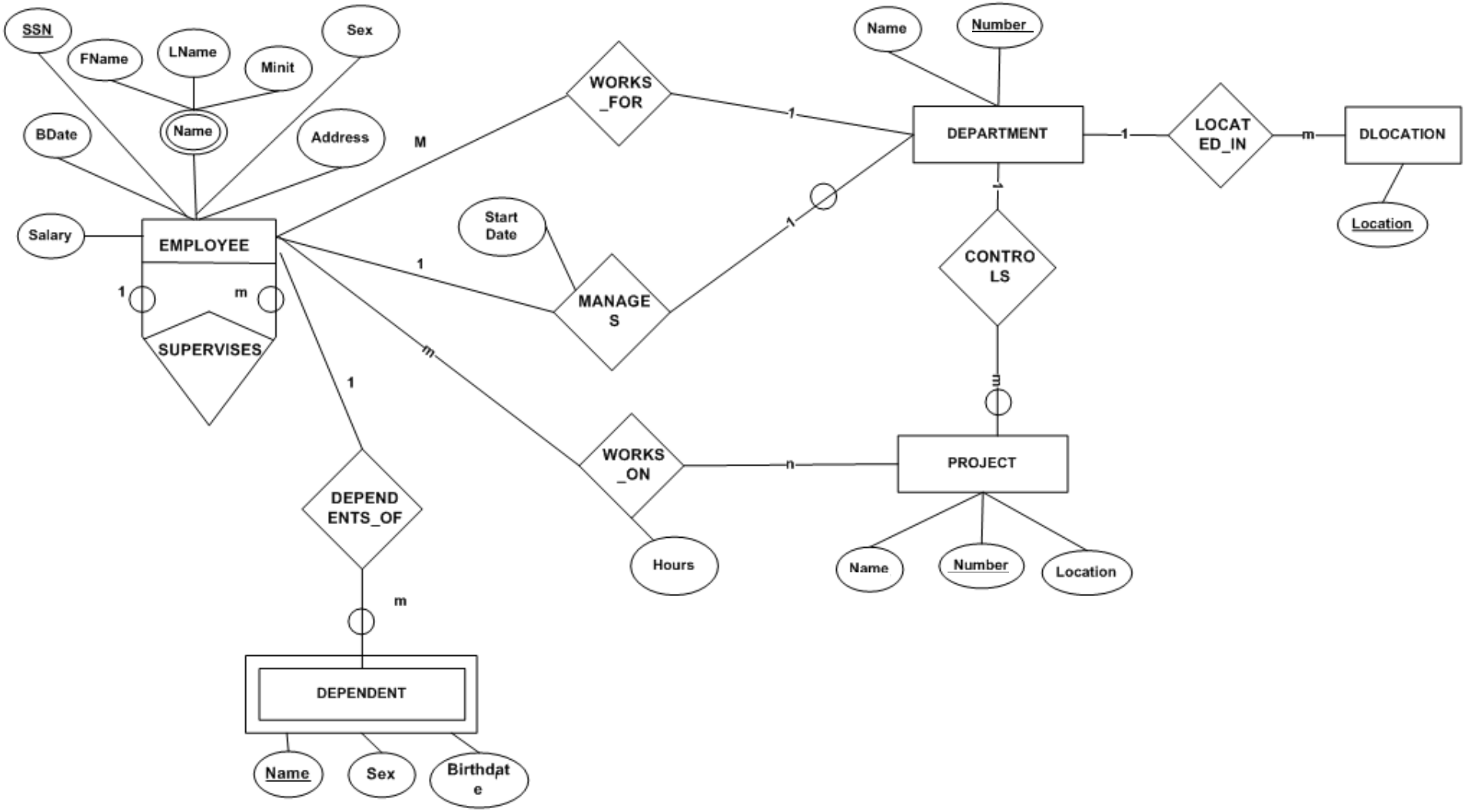
R(A1, A3, A5, A7) หรือ R(A1, A5, A3, A7)

ภาวะการเชื่อมต่อแบบหนึ่งต่อหนึ่งต่อหนึ่ง



R(A1, A3, A5, A7) หรือ R(A1, A3, A5, A7)
หรือ R(A1, A3, A5, A7)

Exercise: Transform the given ER model to relations.



Exercise: Answer

Exercises

1. อธิบายกลยุทธ์การจำลองข้อมูลทั้งสี่
2. ยกตัวอย่างแบบจำลองข้อมูลที่ไม่ถูกต้องหลักวากยสัมพันธ์ของภาษาแบบจำลองข้อมูลอีอาร์มาสามตัวอย่าง
3. เหตุใดจึงควรปรับปรุงแบบจำลองข้อมูลก่อนนำไปแปลงเป็นเค้าร่างเชิงแนวคิด