



Unit 5

Methodologies for Conceptual Design



本單元目的

- ★ 提出 refinement primitives 用來將一初始之 schema 轉為結果所需之 schema.
- ★ Schema transformation-
 - 以漸進修改方式找 schema
- ★ Primitive transformation
 - 用難以再簡化的動作單元做轉換工作



Outline

☼ Primitives for Conceptual Design

- Top-Down Primitives
- Bottom-up Primitives

☼ Strategies for Schema Design

- Top-down Strategy
- Bottom-up Strategy
- Inside-out Strategy
- The Mixed Strategy

☼ Criteria for Choosing among Concepts

☼ Inputs, Outputs, and Activities of Conceptual Design



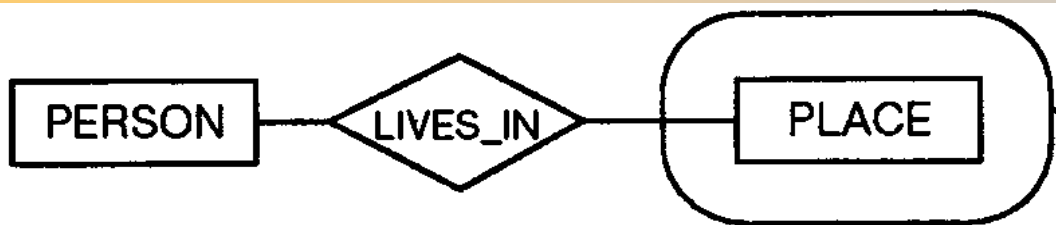
Schema Transformation



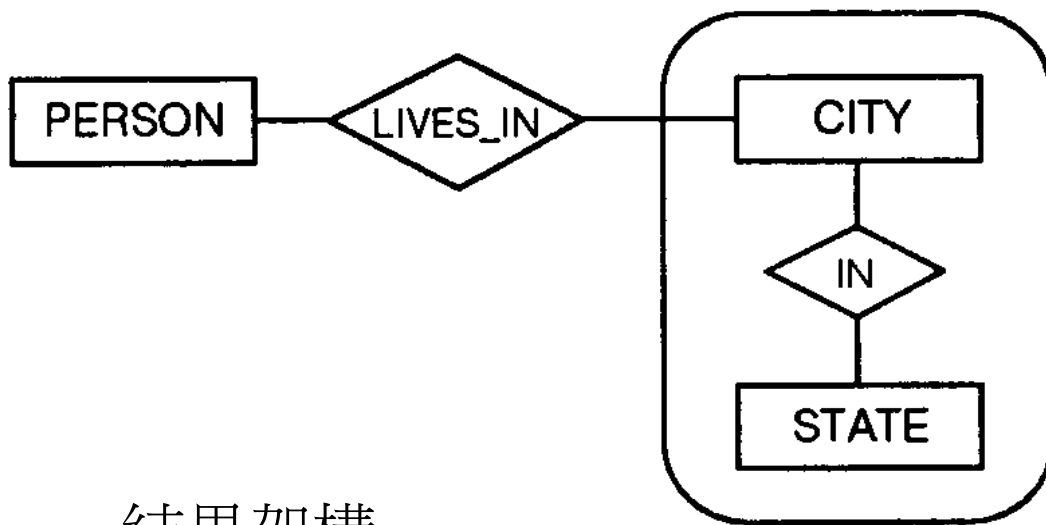
- ★ From starting schema transforming into resulting schema.
- ★ 在 starting schema 中使用的名稱, 在觀念上要對應至 resulting schema.
 - 例: PLACE 轉為 CITY, IN, STATE
- ★ 在 resulting schema 中的邏輯關係 (包括 entities, relationship...) 必須由 starting schema 繼承而來.
 - 例: PERSON 和 PLACE 之間的關係保持到 resulting schema.



An Example of Schema Transformation



(a) Starting schema
初始架構



結果架構

(b) Resulting schema

轉換










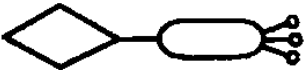



Top-Down Primitives



Primitive	Starting Schema	Resulting Schema
T_1 : Entity \rightarrow Related entities		
T_2 : Entity \rightarrow Generalization (Entity \rightarrow Subset)		
T_3 : Entity \rightarrow Uncorrelated entities		
T_4 : Relationship \rightarrow Parallel relationships		

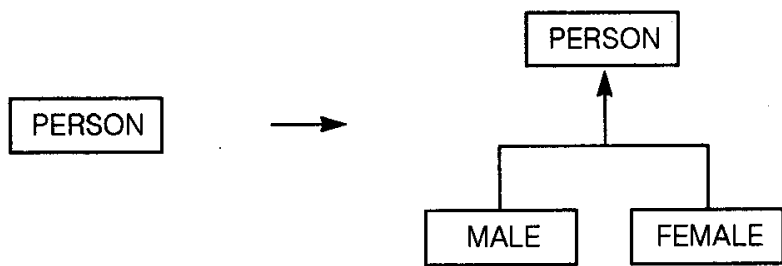


Top-Down Primitives

Primitive	Starting Schema	Resulting Schema
T ₅ : Relationship → Entity with relationships		
T ₆ : Attribute development	 or 	 or 
T ₇ : Composite attribute development	 or 	 or 
T ₈ : Attribute refinement		 or 



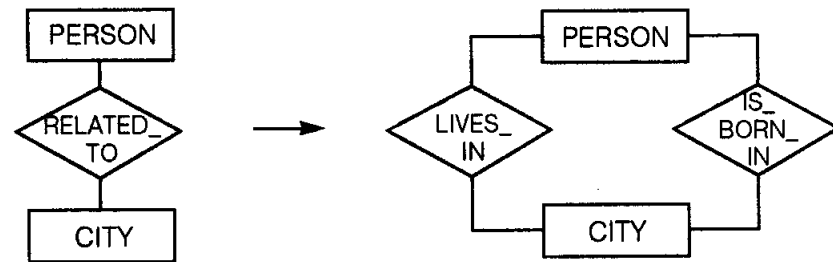
Examples of Top-Down Primitives



(a) Application of primitive T_2



(b) Application of primitive T_3

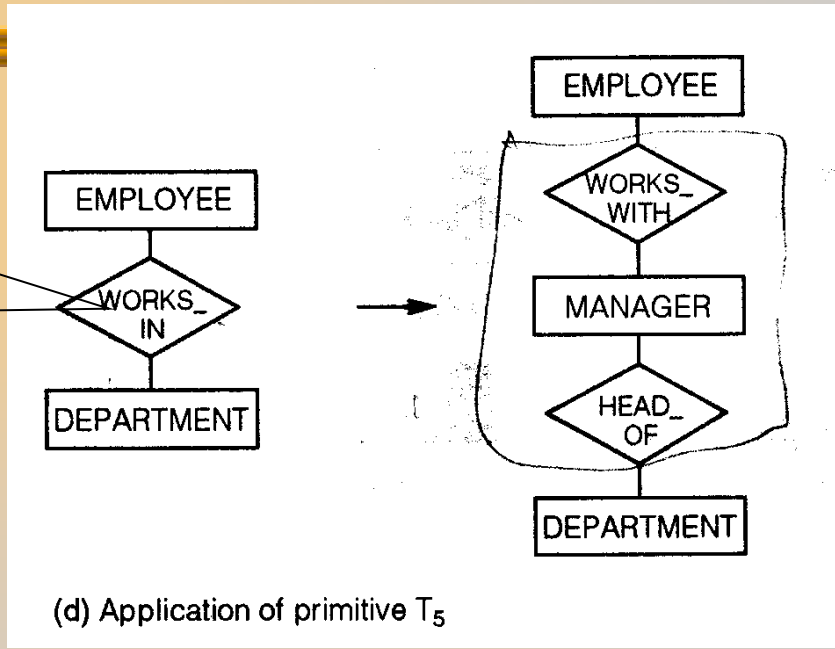


(c) Application of primitive T_4



Example of Top-Down Primitives

MANAGER 原來藏在 DEPARTMENT 裡



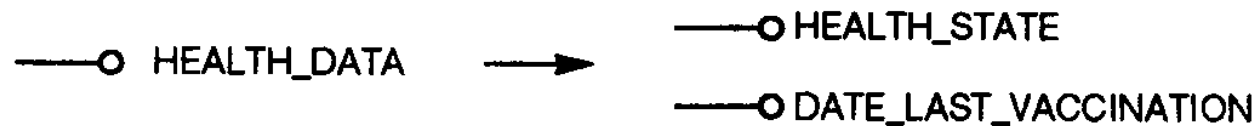
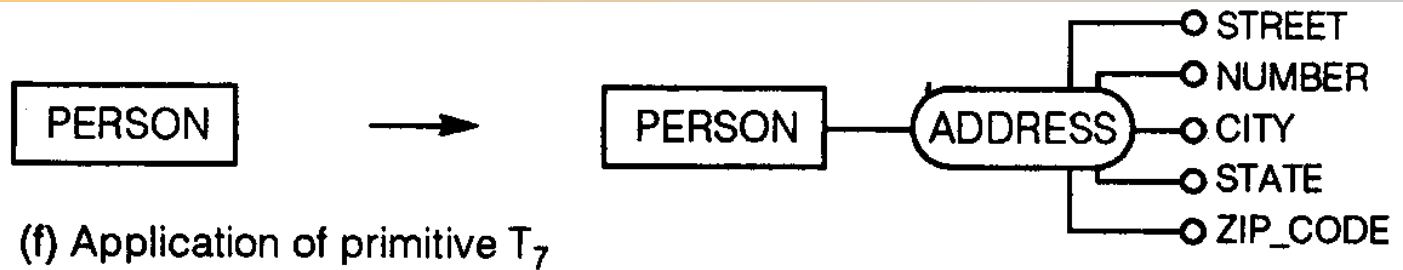
(d) Application of primitive T_5



(e) Application of primitive T_6



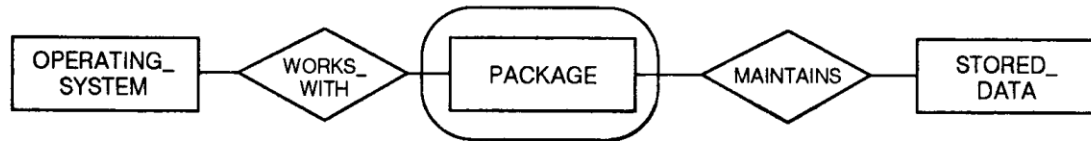
Example of Top-Down Primitives



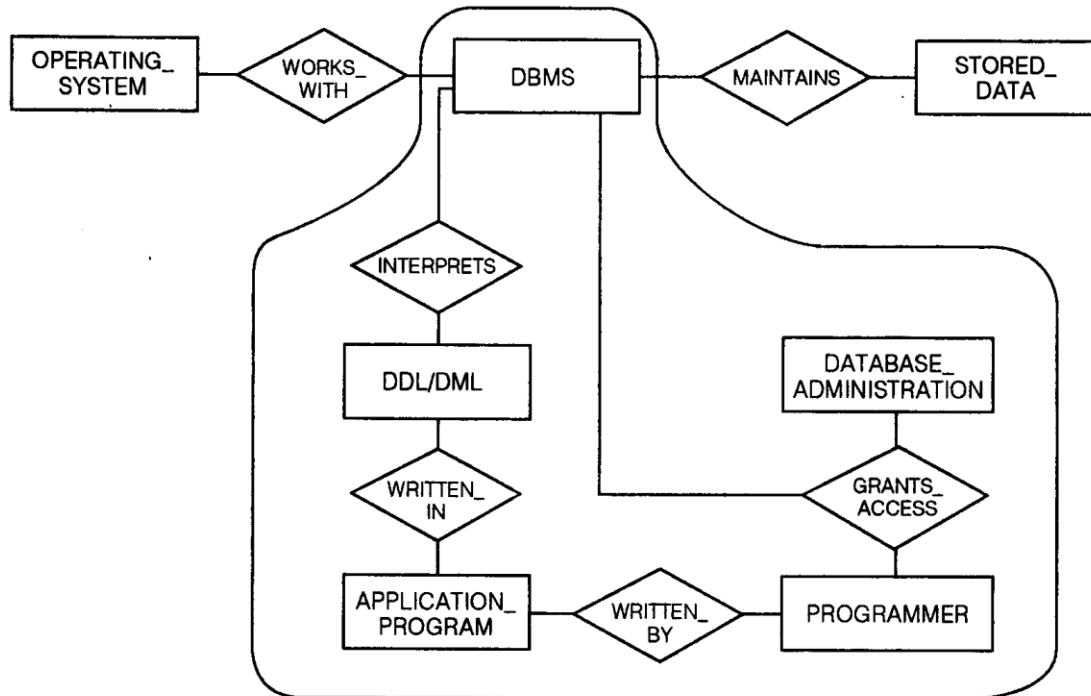
(g) Applications of primitive T_8



A Complex Top-Down Schema Transformation






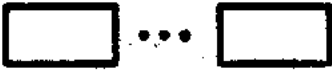
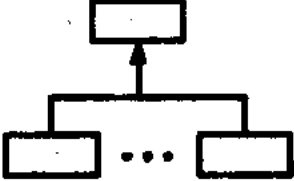




(a) Starting schema



(b) Resulting schema



Bottom-up primitives

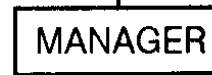
Primitive	Starting Schema	Resulting Schema
B₁: Entity generation		
B₂: Relationship generation		
B₃: Generalization generation (subset generation)		
B₄: Attribute aggregation		
B₅: Composite attribute aggregation		



Example of Bottom-Up Primitives



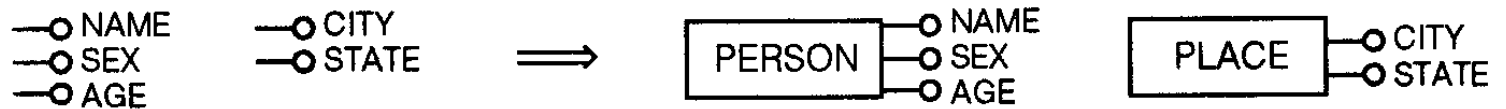
(a) Application of primitive B_2



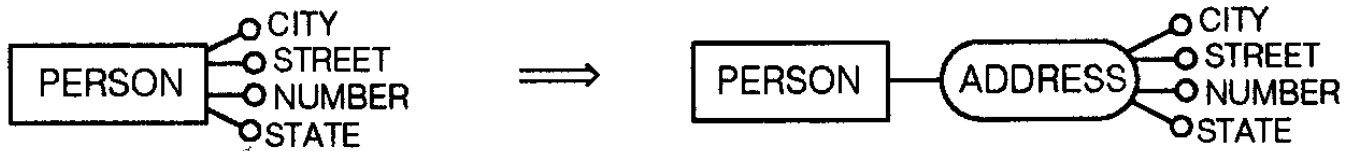
(b) Application of primitive B_3



Example of Bottom-Up Primitives



(c) Application of primitive B₄



(d) Application of primitive B₅



Strategies for Schema Design



- ★ Top-down Strategy
 - 適用於高度結構化的公司
- ★ Bottom-up Strategy
 - 適用在較非正式或結構較鬆散的公司組織
- ★ Inside-out Strategy
- ★ The Mixed Strategy
 - 較具彈性
 - 適合在多人分工設計的大型系統



Top-Down Strategy

- ★ 在一開始就把全部的概念抓住, 表現在 starting schema, 然後慢慢分解其細節.
- ★ 整個 application domain 在各分析階段皆保持相同.
- ★ 適用在一開始即能全盤掌握系統需求者, 對大系統而言幾乎不可能.

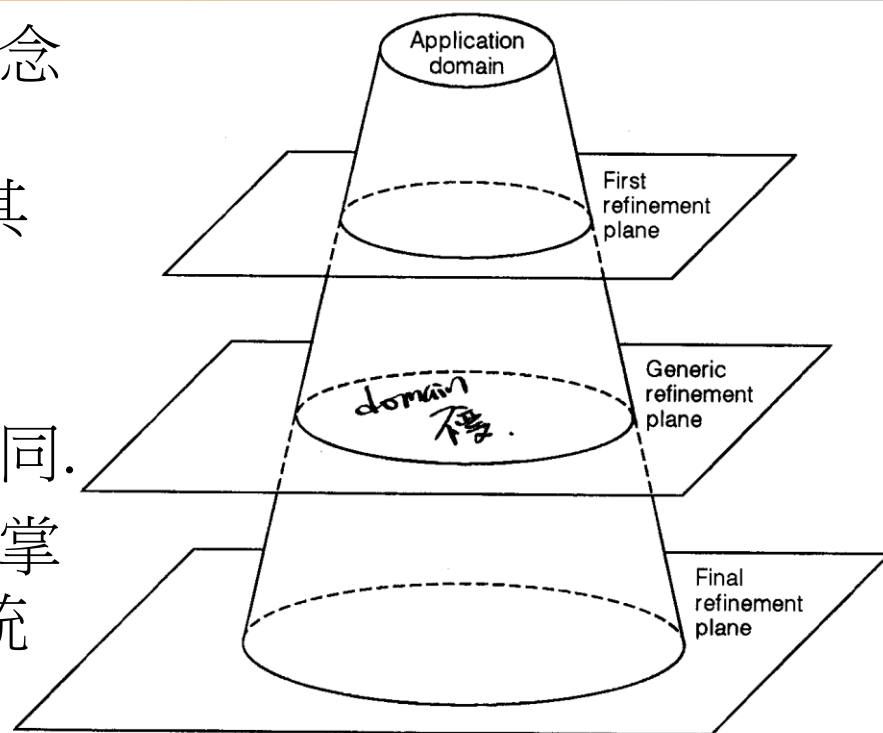


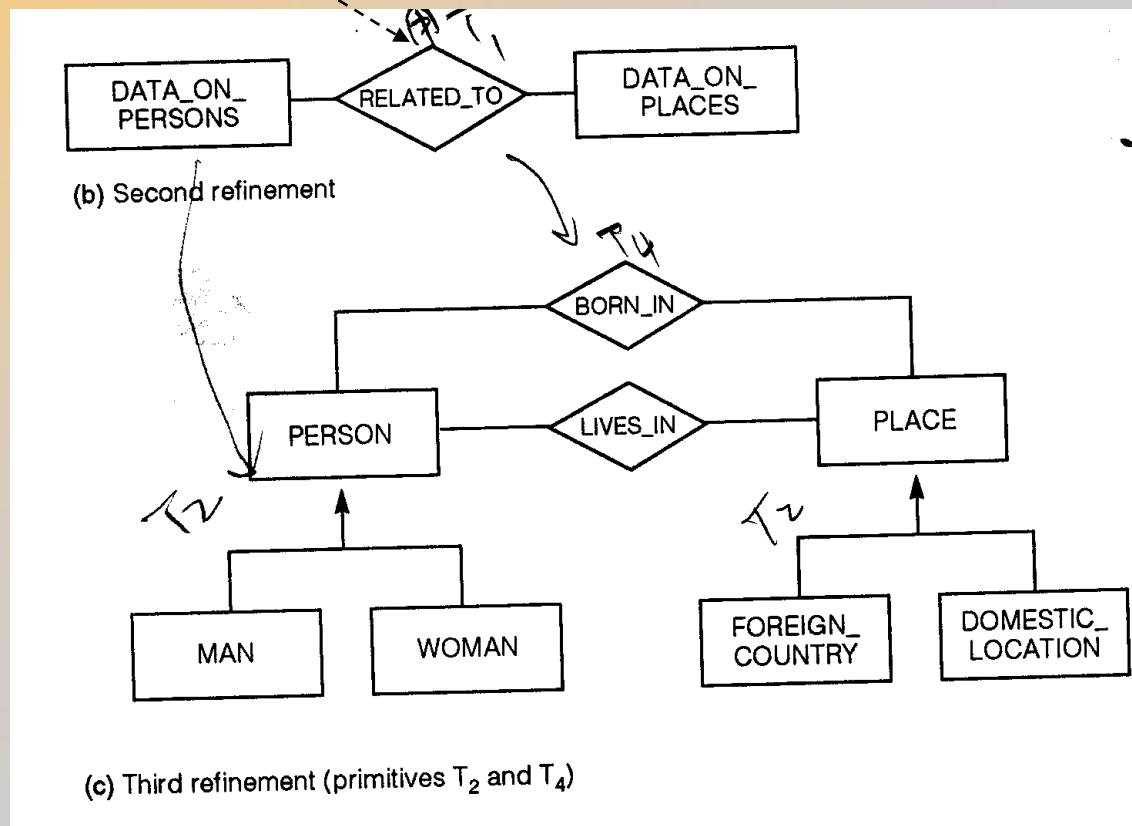
Figure 3.10 The top-down strategy



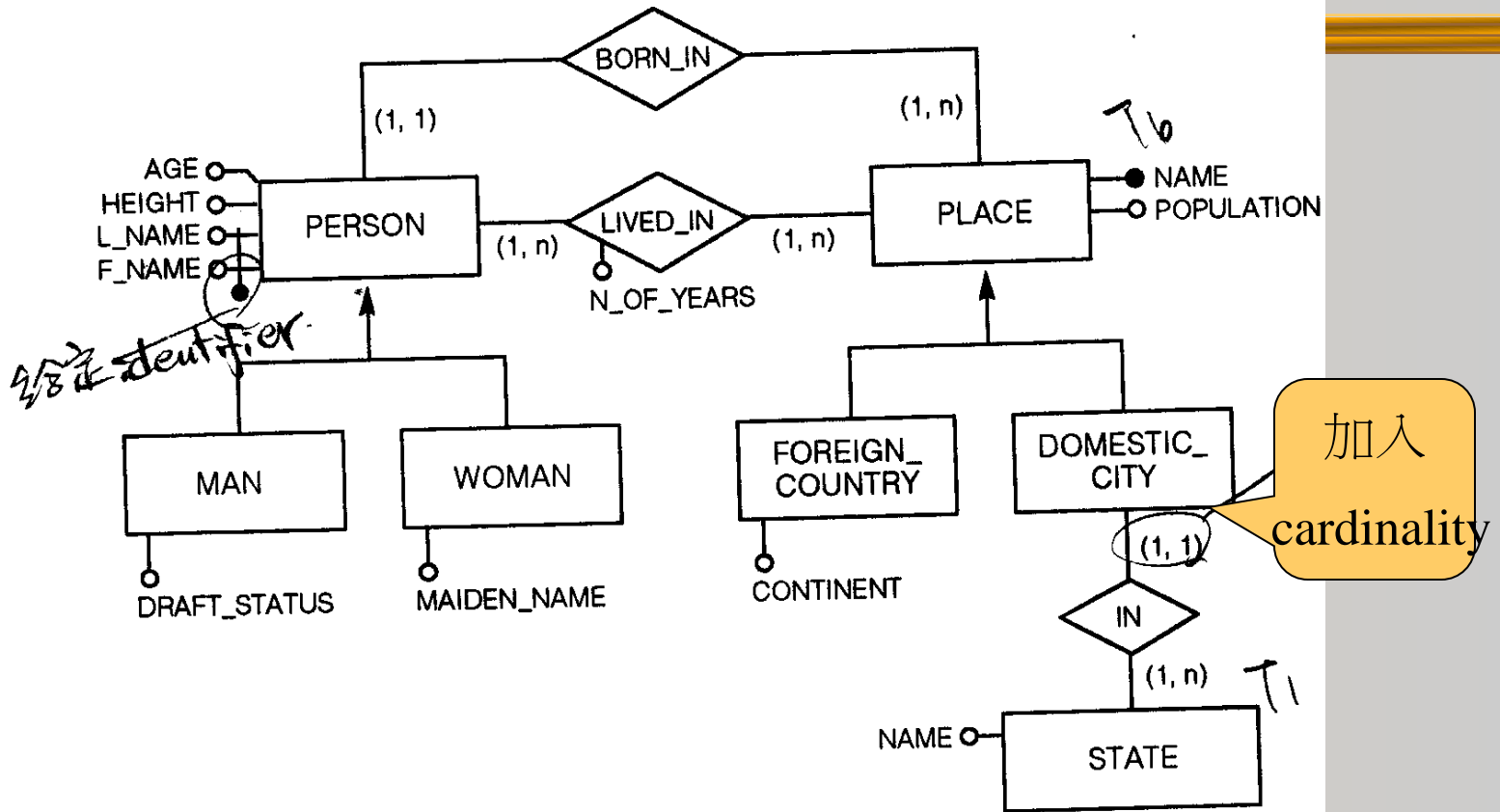
Using Top-Down Strategy

DEMOGRAPHIC_
DATA

(a) First refinement



Using Top-Down Strategy

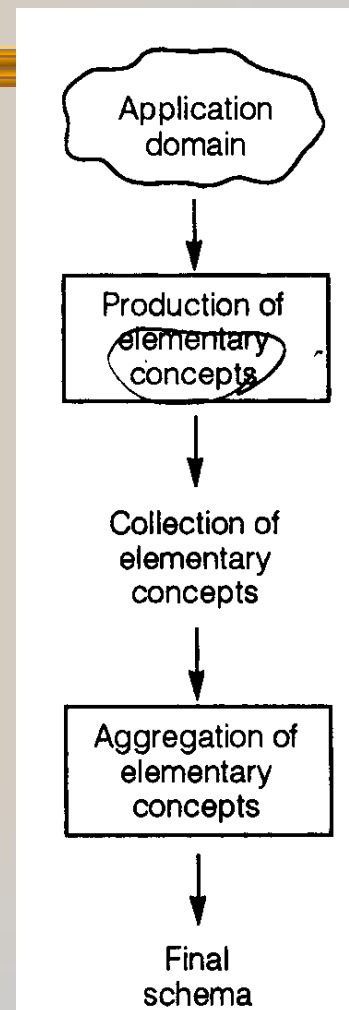


(d) Final schema (primitives T₁ and T₆)



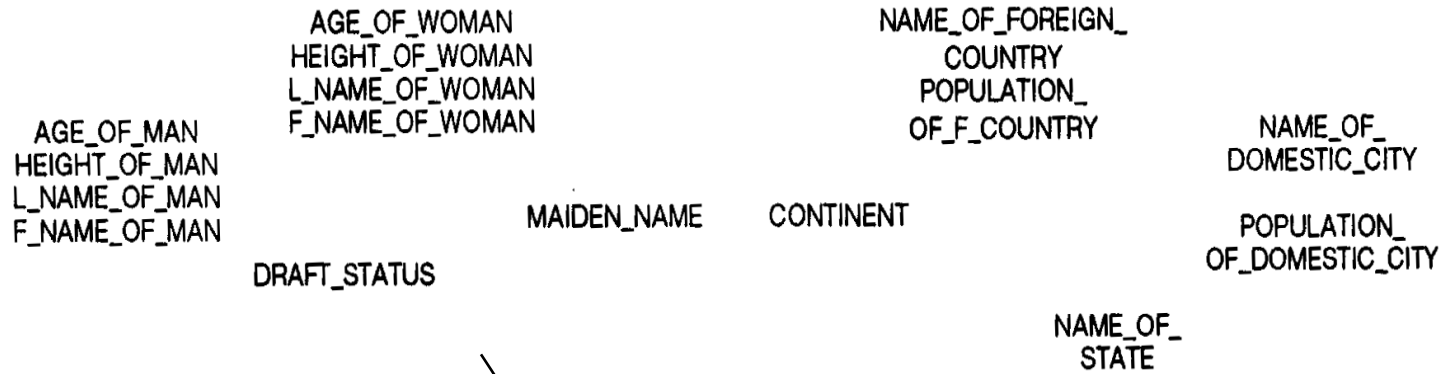
Bottom-Up Strategy

- ★ 從 application domain 中先找出基本概念之元素(如所有的 attributes), 然後逐漸整合為其他較複雜的概念 (例如建立 entities), 然後形成 final schema.
- ★ 優點---把各抽象概念逐一加入, 一一擊破.
- ★ 缺點---對於已存在的概念片段要打散重組時相當難.



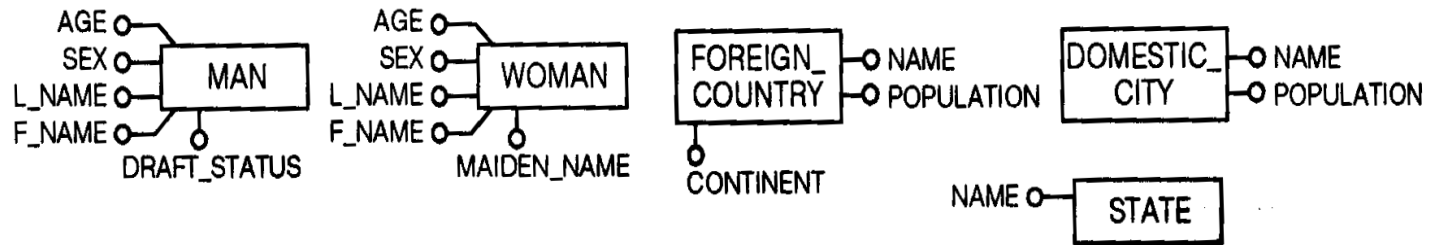


Using Bottom-Up Strategy



(a) First schema

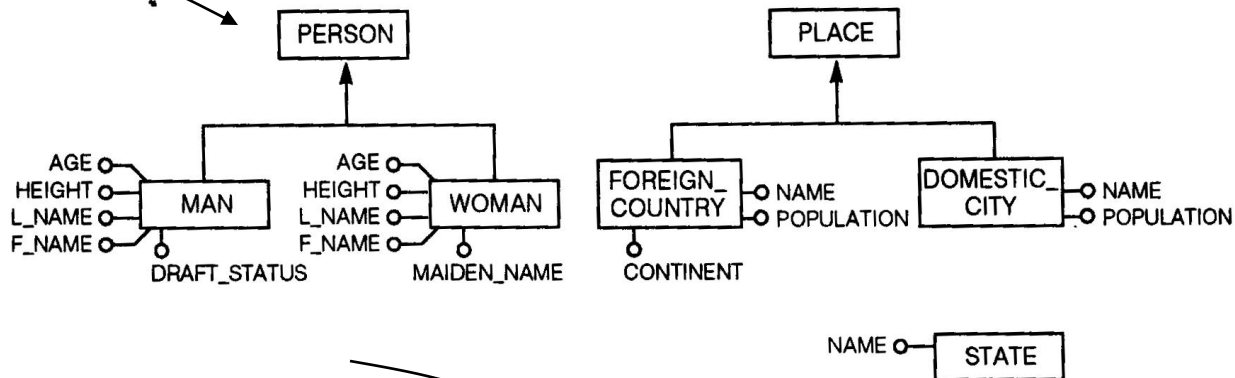
B4



(b) Second schema

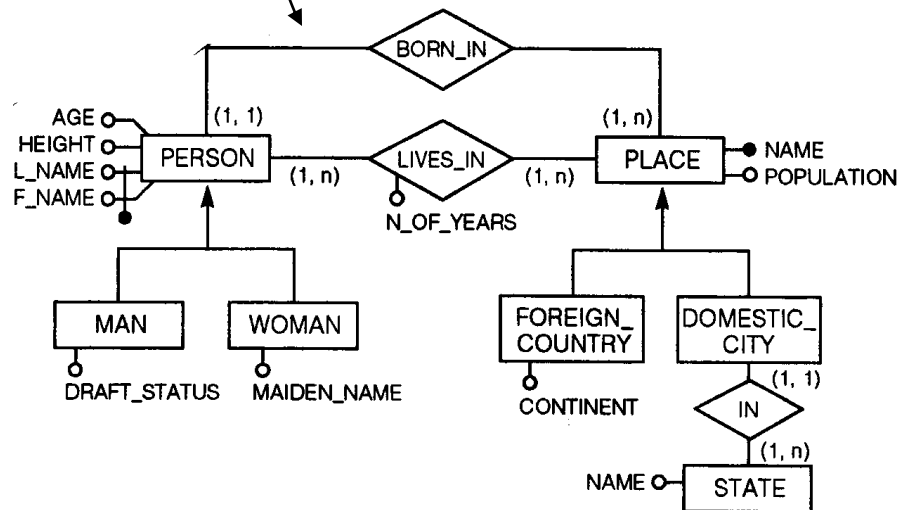
Using Bottom-Up Strategy

B3



(c) Third schema

B2



(d) Final schema (primitive B₂)



Exercise 2

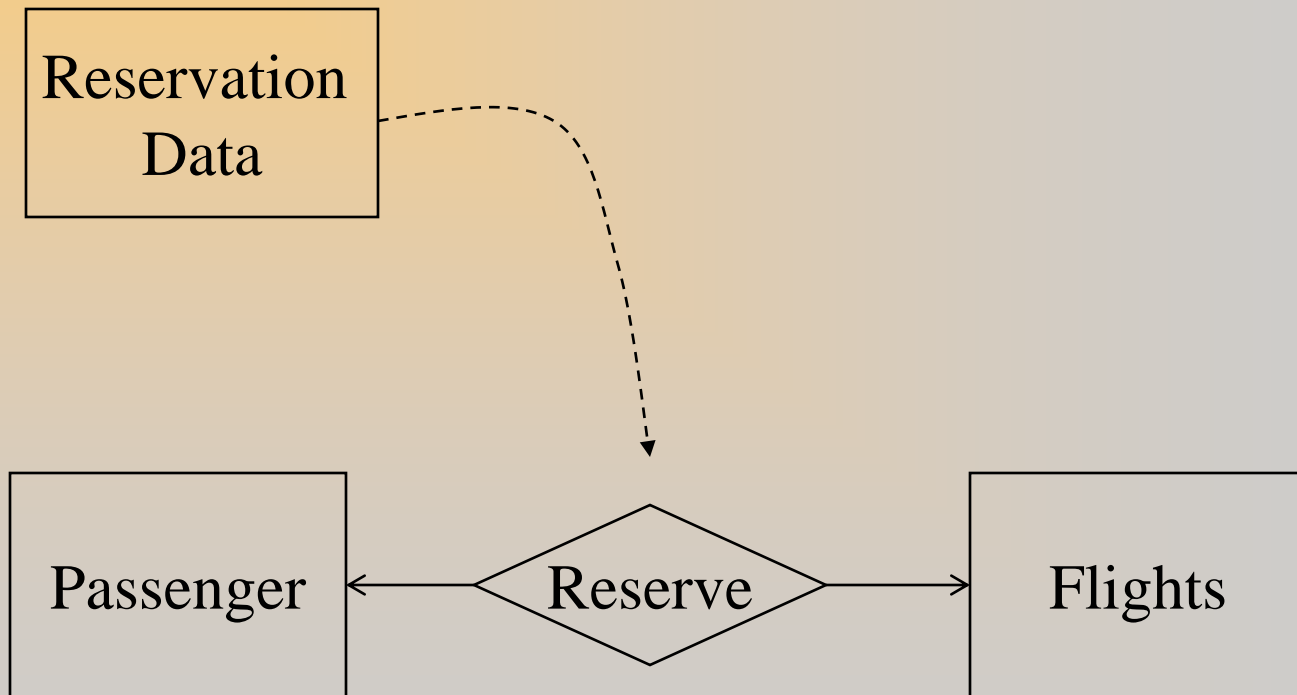
Study the following data requirements for a reservation database and produce a conceptual schema for this application domain. You may need to make certain assumptions about the application requirements; make reasonable assumptions as you proceed.

The reservation database stores data about flights and passenger reservations. For each flight, we know the departure and arrival airports, dates, and times. Assume that flights connect with just one departure airport and one arrival airport, without intermediate stops. For each passenger we know the name, sex, and telephone number; we also know the seat and smoking preference. Each passenger can hold multiple reservations.



Exercise 2

Using Top-down Strategy





Exercise 2

Using Bottom-up Strategy

Flights

Reservation

Departures airports

Arrival airports

Dates

Times

:

:

Passenger

Name

Sex

Telephone number

Seats

Smoking

:

:



Inside-Out Strategy

- ★ 在 application 中選定一個最重要或最明顯的概念當作核心部分.
- ★ 逐一加入其相關概念, 形成 resulting schema.

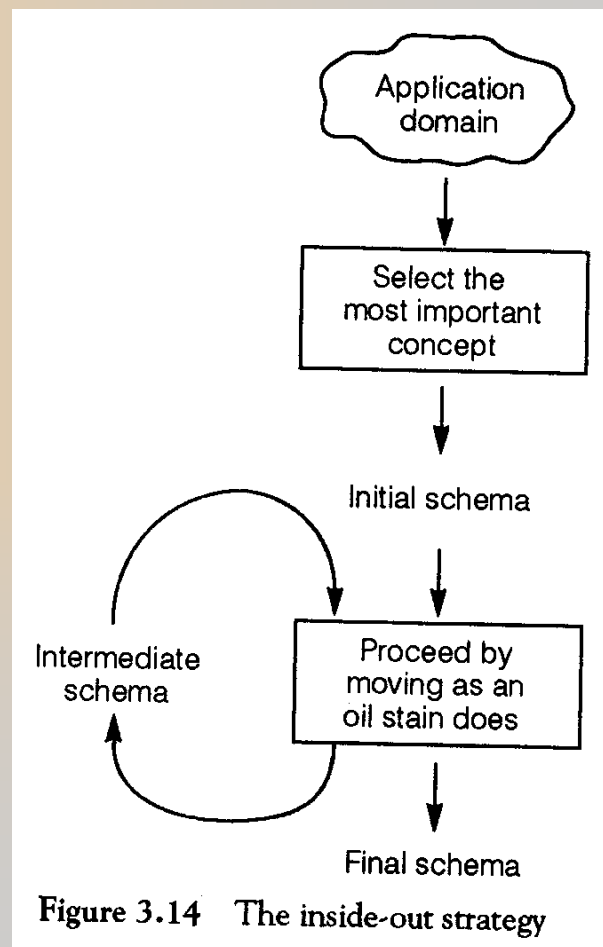
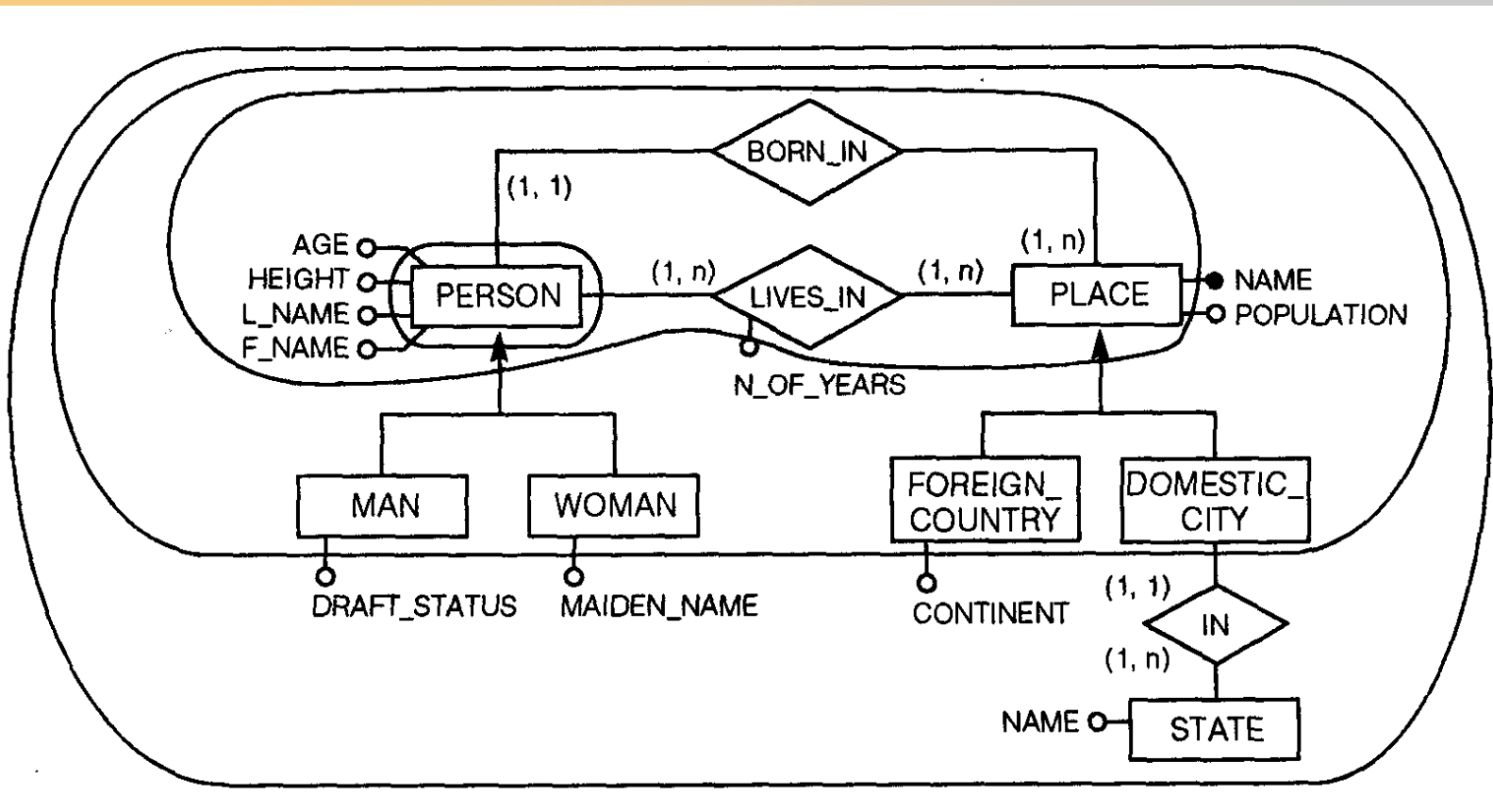


Figure 3.14 The inside-out strategy



Using Inside-Out Strategy





Mixed Strategy

- ★ 把 application domain 先分成數個子集合, 先把各子集合之概念關聯畫出一個主架構 (skeleton schema), 然後再整合其他的子集合畫出的 schema 形成 final schema.
- ★ 適合在多人分工設計的大型系統

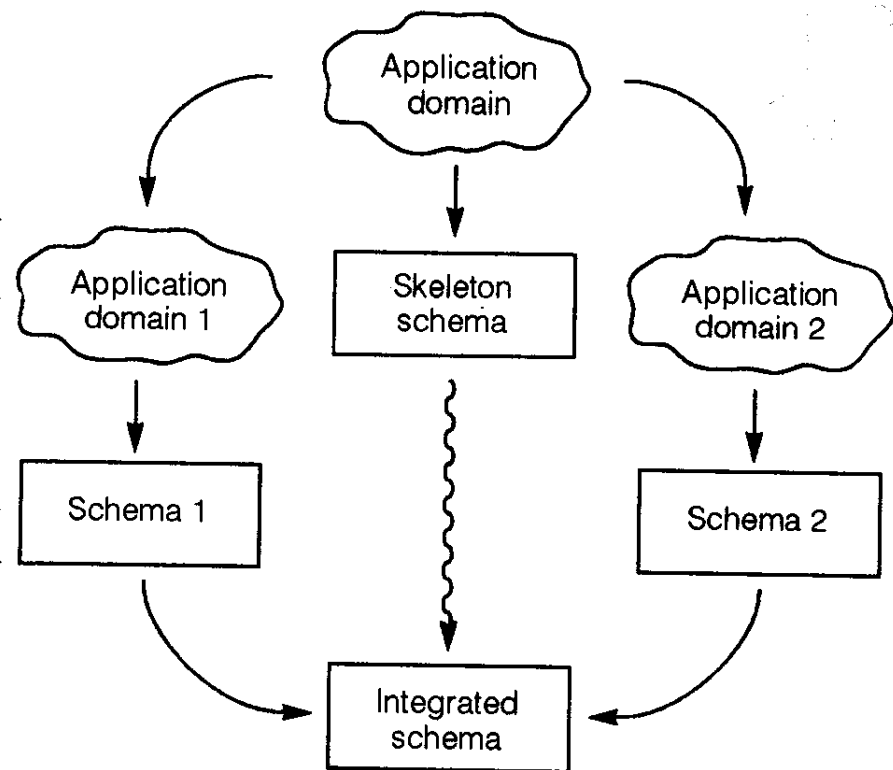
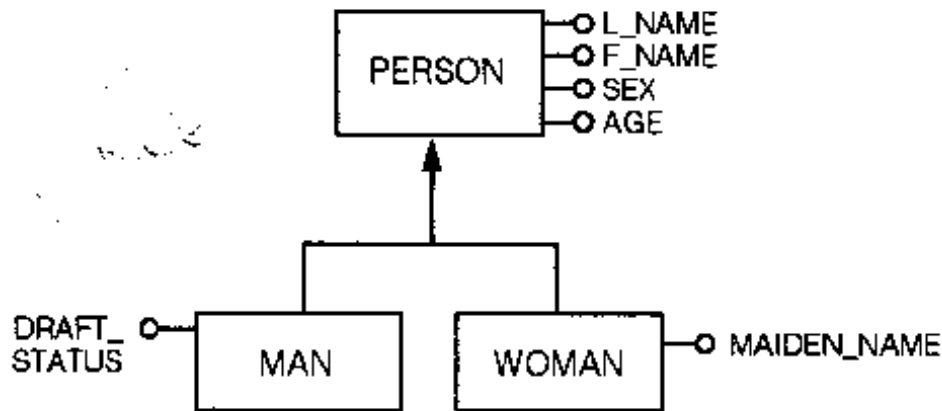


Figure 3.16 The mixed strategy

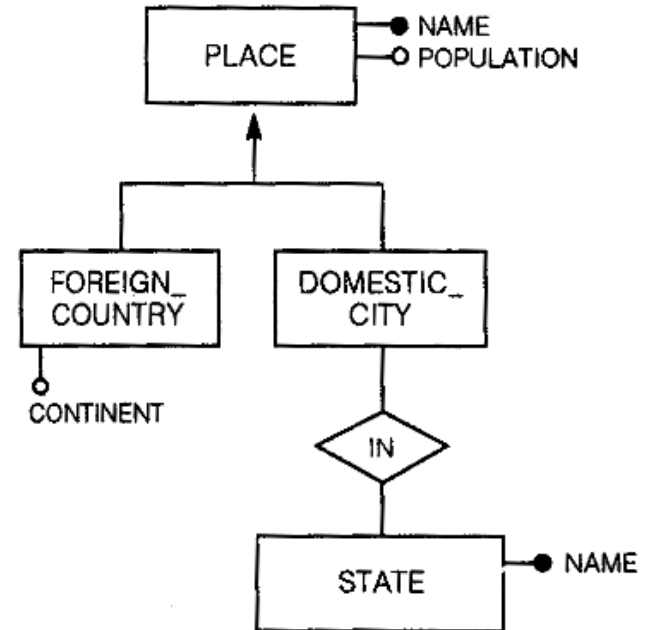
Using Mixed Strategy



(a) Skeleton schema 分為 PERSON 和 PLACE 兩個子集合



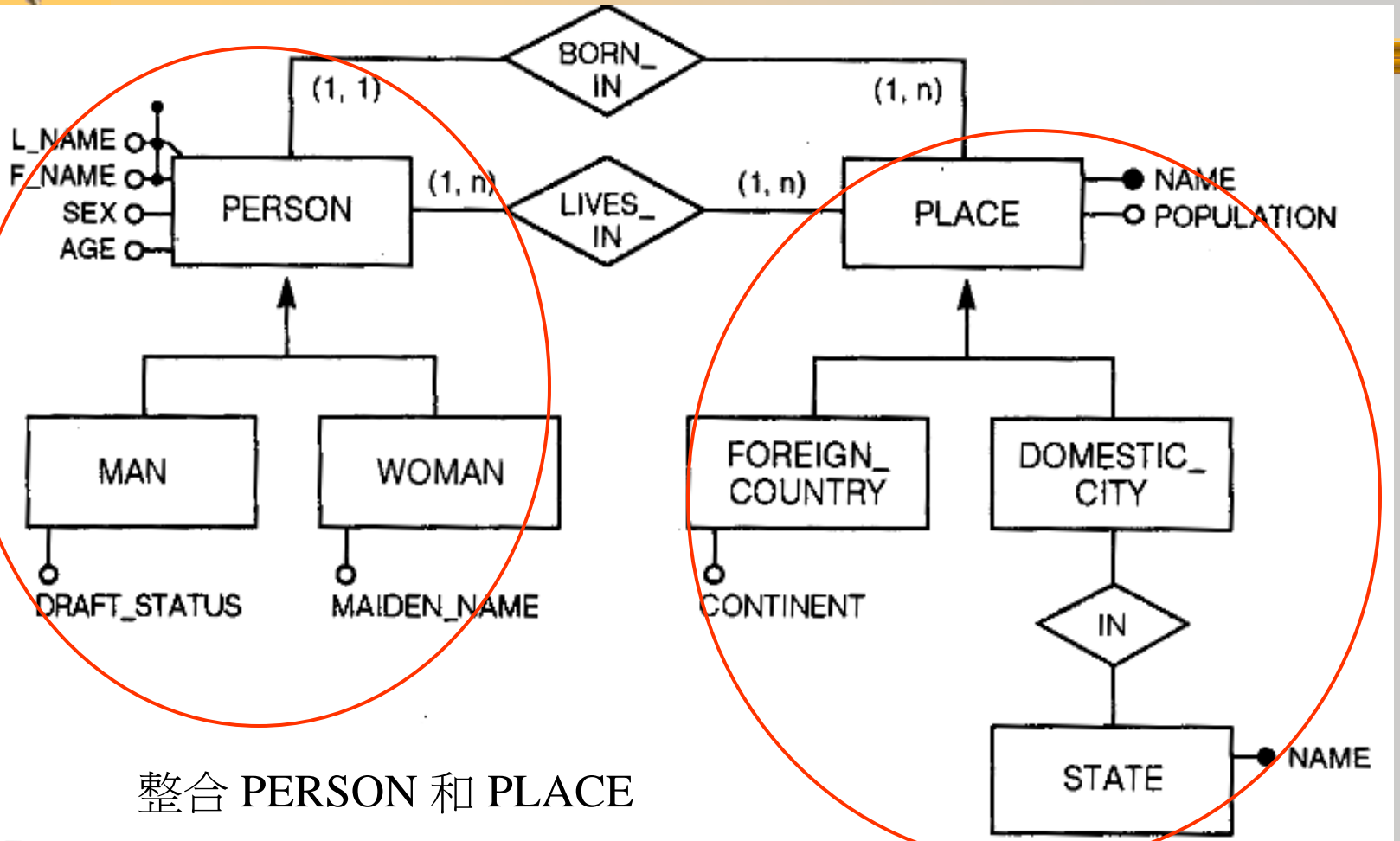
(b) PERSON schema



(c) PLACE schema



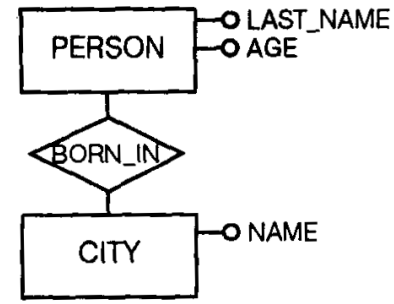
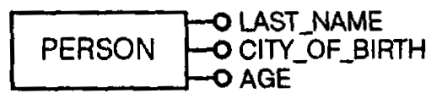
Using Mixed Strategy



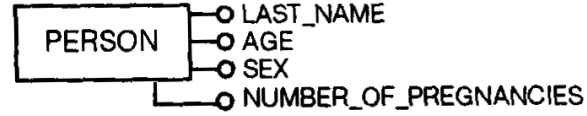
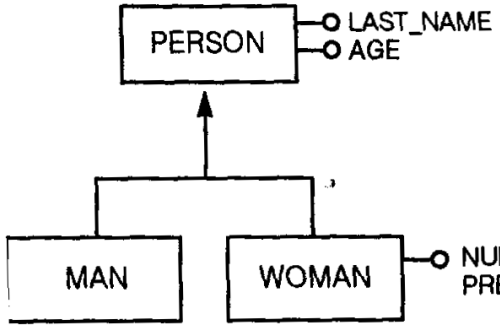
整合 PERSON 和 PLACE



Different schema for representing the same reality



(a) First example



(b) Second example



Criteria for Choosing Among Concepts

★ Entity v.s. simple attribute

- 如果要放入該 schema 的 object 有很多特性 (例: attributes, relationships, generalization) 與 object 相連, 則適合定為 entity.

★ Generalization v.s. attribute

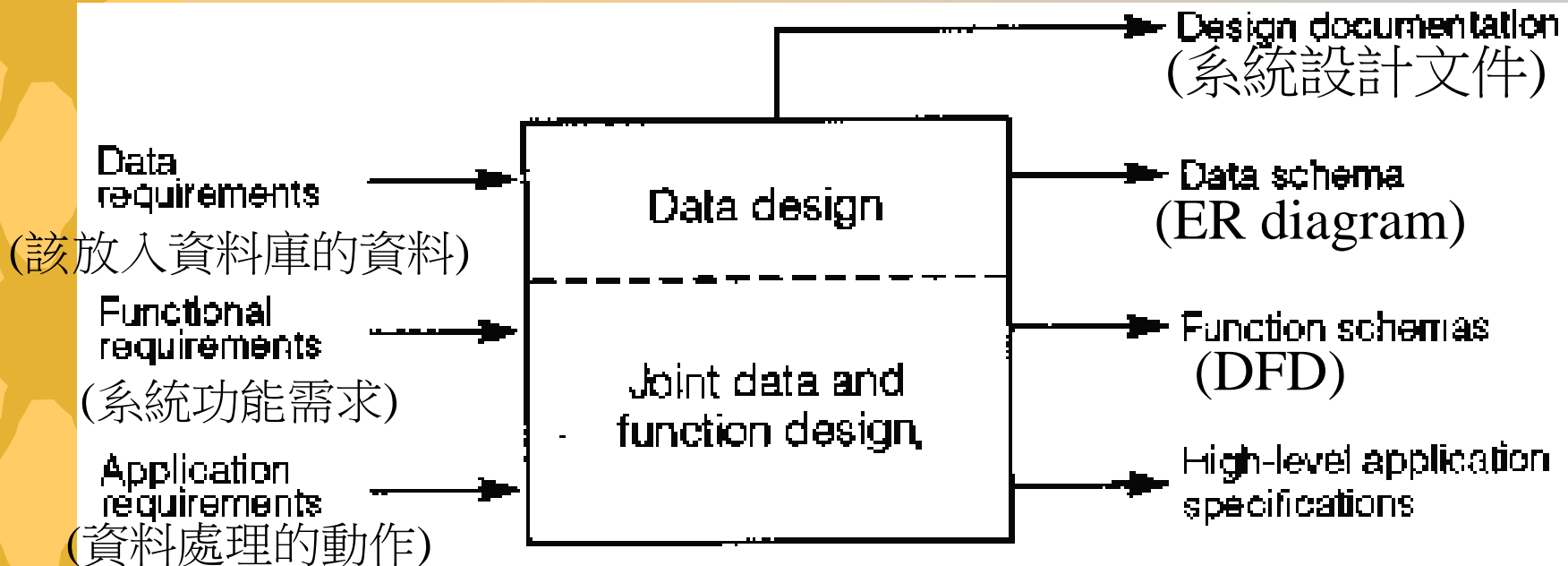
- 如果下一層分出去, 會有某些 attribute 描述該 object 或其他 relationship 與該 object 相連, 則適合分出 subset entity.

★ Composite attribute v.s. set of simple attribute

- 當一組 attributes 可以放在一起, 給一個名稱時, 則適合定為 composite attribute.



Inputs and Outputs of Conceptual Design





Activities for Conceptual DB Design

- ★ Requirement analysis
 - 把需求更結構化, 去掉重複或不清楚的概念.
- ★ Initial conceptualization
 - 挑出一初步結構
- ★ Incremental conceptual design
 - 逐步修改結構, 為概念資料庫設計的主要動作.
- ★ Integration (整合)
- ★ Reconstructing (重建)



Exercise 1



1. Consider the following requirements that refer to the organization of a course. The requirements are written with an inside-out style: they start by describing the participants in a course and proceed to describe all other relevant aspects.
 - For every participant in a course, store first name, last name, birth date, and sex.
 - Indicate whether each participant is married and the number of children.
 - Represent also the cities where they reside and the cities where they were born, with the states.
 - Store other information concerning the course they attended, number of classes, date, topics), and the teacher (or teachers) who taught them (first name, last name, affiliation).
 - For teachers affiliated with universities, indicate their university affiliation and their field of specialization.

Produce a conceptual schema, using the inside-out strategy. Then redesign the schema, using the top-down and mixed strategies.



Exercise 2

2. Study the following data requirements for a reservation database and produce a conceptual schema for this application domain, using these strategies:

Top-down Strategy

Bottom-up Strategy

Inside-out Strategy

The Mixed Strategy

You may need to make certain assumptions about the application requirements; make reasonable assumptions as you proceed.

The reservation database stores data about flights and passenger reservations. For each flight, we know the departure and arrival airports, dates, and times. Assume that flights connect with just one departure airport and one arrival airport, without intermediate stops. For each passenger we know the name, sex, and telephone number; we also know the seat and smoking preference. Each passenger can hold multiple reservations.



Exercise 3

3. Study the following data requirements for a hospital database and produce a conceptual schema for this application domain, using these strategies:

Top-down Strategy

Bottom-up Strategy

Inside-out Strategy

The Mixed Strategy

You may need to make certain assumptions about the application requirements; make reasonable assumptions as you proceed.

The Hospital database stores data about patients, their admission and discharge from hospital's departments, and their treatments. For each patient, we know the name, address, sex, social security number, and insurance code (if existing). For each department, we know the department's name, its location, the name of the doctor who heads it, the number of beds occupied. Each patient goes through multiple treatments during hospitalization; for each treatment, we store its name, duration, and the possible reactions to it that patient may have.