



DATABASE SYSTEMS ARCHITECTURE

05-041-301

DataBase Management System



หัวข้อในการเรียน

2

1. สถาปัตยกรรมระบบฐานข้อมูล
2. ความอิสระของข้อมูล
3. แบบจำลองข้อมูล (Data Model) หรือ แบบจำลองฐานข้อมูล (Database Model)
4. ประเภทของแบบจำลองฐานข้อมูล
5. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)
 - แนวคิดทั่วไปเกี่ยวกับแบบจำลองเชิงสัมพันธ์
 - กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)
 - พื้นฐานการสร้างแบบจำลองข้อมูลเชิงสัมพันธ์
 - คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์



สถาปัตยกรรมฐานข้อมูล (Database System Architecture)



สถาปัตยกรรมระบบฐานข้อมูล

4

- หมายถึง การอธิบายเกี่ยวกับโครงสร้างและส่วนประกอบหลักที่นำมาประกอบรวมกันเป็นฐานข้อมูล
- การออกแบบสถาปัตยกรรม เป็นการออกแบบจำลองเชิงนามธรรมของข้อมูล ที่ไม่ได้ลงรายละเอียดภายใน แต่จะเป็นพื้นฐานในการนำไปใช้



ประวัติ

5

□ สถาปัตยกรรมระบบฐานข้อมูล

ค.ศ. 1971

กลุ่มคณะทำงาน DBTG
(Database Task Group)
กำหนดสถาปัตยกรรม
ระบบฐานข้อมูลไว้ 2 ระดับ

1. สคีมา (Schema)
ที่เป็นมุมมองระบบ
(Systems View) และ
2. ซับสคีมา (Subschema)
ที่เป็นมุมมองผู้ใช้งาน
(User Views)

ค.ศ. 1975

ANSI /SPARC
ได้พัฒนาสถาปัตยกรรมข้อมูล
ขึ้นมาใหม่ มีลักษณะคล้ายกัน
เสนอให้แบ่งมุมมอง
ออกเป็น 3 ระดับ
แนวคิดหลัก คือ ให้ข้อมูลเป็น
อิสระต่อโปรแกรม

ปัจจุบัน

ระดับชั้น
ประกอบด้วย 3 ระดับ

1. ระดับภายใน (Internal Level)
2. ระดับแนวคิด
(Conceptual Level)
3. ระดับภายนอก
(External Level)

5



สถาปัตยกรรมระบบฐานข้อมูล

6

□ สถาปัตยกรรมฐานข้อมูล (Database System Architecture)

สถาบัน **ANSI** (American National Standard Institute) และ **SPARC** (Standard Planning and Requirement Committee) ได้พัฒนาสถาปัตยกรรมของระบบฐานข้อมูลโดยแบ่งมุมมองออกเป็น **3 ระดับ** ได้แก่

1. ระดับภายนอก (External level)
2. ระดับแนวคิด (Conceptual level)
3. ระดับภายใน (Internal level)



สถาปัตยกรรมระบบฐานข้อมูล

7

□ สถาปัตยกรรมฐานข้อมูล (Database System Architecture)

“จุดประสงค์ของสถาปัตยกรรม 3 ระดับ คือ

...เพื่อแยกมุมมองของผู้ใช้ ออกจาก ฐานข้อมูลทางกายภาพ เพื่อให้ข้อมูล
มีความเป็นอิสระจากกัน ”



สถาปัตยกรรมระบบฐานข้อมูล

8

- เหตุผลในการแยกแต่ละระดับจากกัน มีดังนี้
- ผู้ใช้แต่ละคนสามารถเข้าถึงข้อมูลเดียวกันได้ แต่อาจจะมีมุมมองในการใช้งานต่างกันได้ โดยผู้ใช้แต่ละคนสามารถเปลี่ยนแปลงมุมมองข้อมูลของตนเองได้แต่ไม่กระทบกับมุมมองของผู้ใช้อื่น
- ผู้ใช้จะไม่สามารถเข้าถึงข้อมูลในระดับกายภาพได้โดยตรง



สถาปัตยกรรมระบบฐานข้อมูล

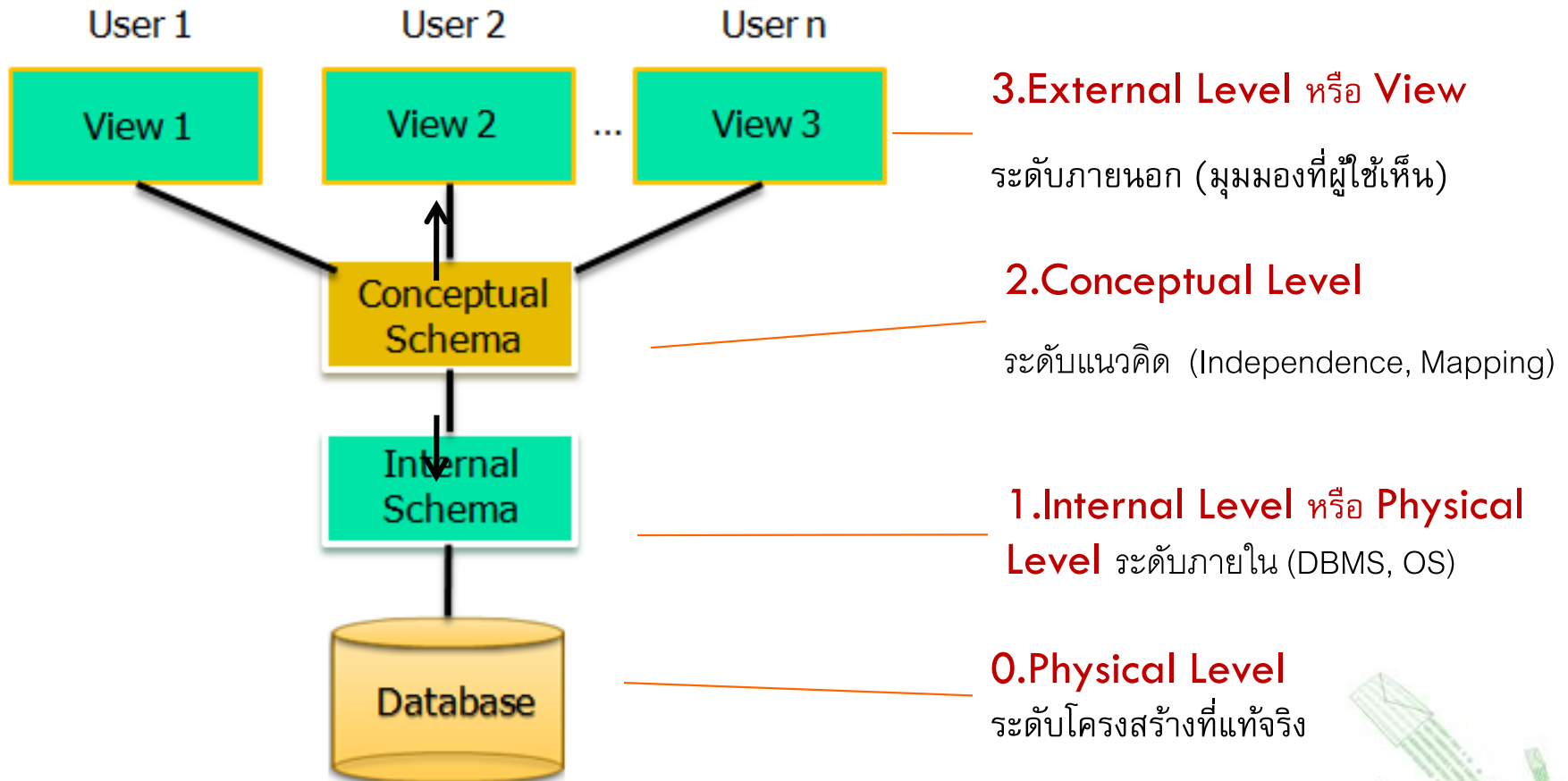
9

- ❑ เหตุผลในการแยกแต่ละระดับจากกัน (ต่อ)
- ❑ ผู้บริหารฐานข้อมูลสามารถแก้ไขโครงสร้างในการจัดเก็บฐานข้อมูล โดยไม่ส่งผลกระทบต่อมุมมองของผู้ใช้
- ❑ โครงสร้างของระดับภายในของฐานข้อมูลจะไม่ได้รับผลกระทบจากการเปลี่ยนแปลงตำแหน่งในการจัดเก็บในระดับกายภาพ เช่น การเปลี่ยนหน่วยเก็บข้อมูล (storage device) ใหม่
- ❑ ผู้บริหารฐานข้อมูล (DBA) สามารถที่จะเปลี่ยนโครงสร้างระดับแนวคิดของฐานข้อมูลโดยไม่ส่งผลกระทบกับผู้ใช้ทุกคน



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ

10



สถาปัตยกรรม 3 ระดับ ของ ANSI-SPARC

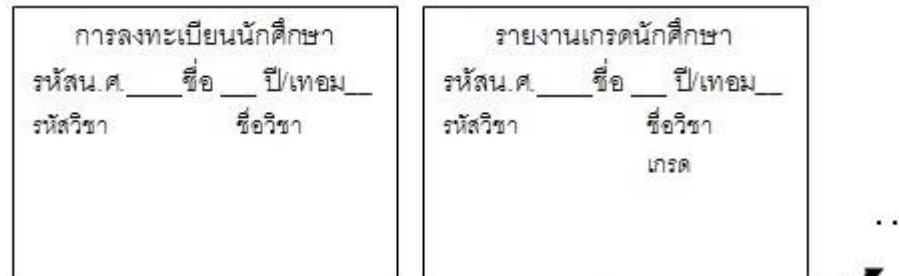


ตัวอย่าง

11

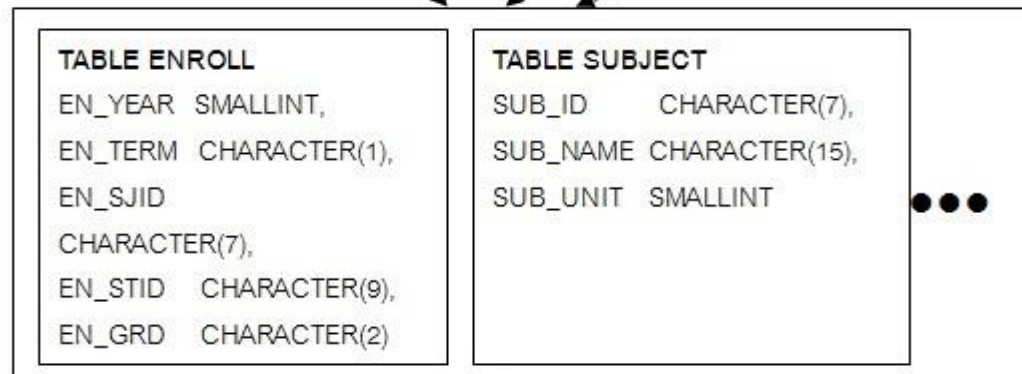
ระดับภายนอกหรือวิว

(External Level)



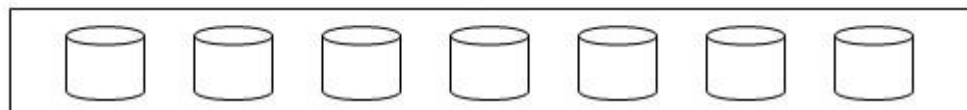
ระดับแนวคิด

(Conceptual Level)



ระดับภายใน

(Internal Level)



สถาปัตยกรรมฐานข้อมูล 3 ระดับ (ต่อ)

12

1. ระดับภายใน (Internal Level) หรือ Physical Level

- **HOW?** มีการกำหนดโครงสร้างข้อมูลเพื่อจัดเก็บข้อมูลอย่างไร
- ดีความในระดัการจัดเก็บข้อมูลจริง เป็นหน้าที่ของผู้ออกแบบอย่างแท้จริง
- ระบบการจัดการฐานข้อมูล(DBMS) และระบบปฏิบัติการ(OS) ทำงานที่ระดับนี้

```
struct person{  
    int id;  
    char name[20];  
    char address[20]  
} index id;
```



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

13

1. ระดับภายใน (Internal Level) หรือ Physical Level

- โครงสร้างภายในที่กำหนดนั้น เช่น แบบทรี (B-Tree) แบบเรียงลำดับดัชนี (Indexed Sequential) แบบ Pointer เพื่อ...?
 - ใช้ในการค้นหาข้อมูล
 - การจัดการพื้นที่เพื่อการจัดเก็บข้อมูลและการเรียงลำดับดัชนีข้อมูล
 - การบีบอัดข้อมูลและการเข้ารหัสข้อมูล



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

14

1. ระดับภายใน (Internal Level) หรือ Physical Level

- ระดับนี้จะมีโครงสร้างภายใน (Internal Schema) ที่อธิบายเกี่ยวกับโครงสร้างการจัดเก็บเชิงกายภาพของฐานข้อมูล
- โครงสร้างต่าง ๆ เหล่านี้ถูกกำหนดโดยผู้บริหารฐานข้อมูล (Database Administrator: DBA) และสามารถเปลี่ยนแปลงได้ด้วย DBA เท่านั้น

“โครงสร้างข้อมูลระดับภายใน ในมุมมองผู้ใช้ฐานข้อมูลจะไม่สามารถเห็นรายละเอียดทางกายภาพในระดับนี้ เพราะถูกซ่อนไว้ นอกจาก DBA ที่สามารถเปลี่ยนแปลงโครงสร้างข้อมูลเหล่านี้”



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

15

2. ระดับแนวคิด (Conceptual Level) หรือ โครงสร้างข้อมูลระดับตรรกะ (Logical)

- ▣ **WHAT?** เป็นระดับที่อธิบายว่ามีข้อมูลอะไรบ้างที่จะเก็บลงในฐานข้อมูล ด้วยการแสดงข้อมูลและความสัมพันธ์ระหว่างข้อมูลในระบบ
- ▣ ดีความออกมาเป็นตารางโดยนำแบบฟอร์มต่าง ๆ มารวมกัน เพื่อแสดงความต้องการของผู้ใช้ในรูปที่สมบูรณ์ อาจมีการวิเคราะห์ และออกแบบโดยผ่านขั้นตอนมากมาย ทั้ง E-R หรือ Normalization จนเสร็จสิ้น

พนักงาน (รหัส, ชื่อ, ที่อยู่) ในแบบสคีมา(Schema) หรือ

person (id, name, address) ในแบบสคีมา(Schema)



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

16

2. ระดับแนวคิด (Conceptual Level) (ต่อ)

- เป็นส่วนที่ประสานการใช้และแปลงความหมาย และทำให้เกิดความเป็นอิสระในระดับโครงสร้าง (Structure Independent)
- เป็นระดับที่เป็นโครงสร้างรวมของระบบ
- เน้นความสัมพันธ์ระหว่างข้อมูลเป็นสำคัญ หรือที่เรียกว่า **แบบจำลองข้อมูล (Data Model)** เช่น E-R หรือ Normalization
- DBA หรือโปรแกรมเมอร์เท่านั้นที่เป็นผู้จัดการและแก้ไขโครงสร้างข้อมูลได้



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

17

2. ระดับแนวคิด (Conceptual Level) (ต่อ)

■ การทำระดับแนวคิดจะเกี่ยวข้องกับ...

- การกำหนด Entity Attribute และความสัมพันธ์ของแต่ละ Entity ทั้งหมด
- กฎเกณฑ์ข้อบังคับในข้อมูล (Constraints)
- ระบบความปลอดภัยและกฎความคงสภาพในข้อมูล (Integrity)

“สำหรับการนำเสนอข้อมูลในระดับแนวคิด ข้อมูลที่แสดงจะอยู่ในรูปแบบ ตารางหรือรีเลชัน”



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

18

3. ระดับภายนอก (External Level) หรือ วิว (Views) (มุมมองที่ผู้ใช้เห็น)

- มาจากแบบฟอร์มเอกสาร ว่ามีอะไรในเอกสารบ้าง หรือจากผู้ใช้ที่แต่ละคน เป็นการรวบรวมข้อมูลอย่างง่าย ๆ จากผู้ใช้ เพื่อให้กับนักวิเคราะห์นำไปศึกษา

- ผู้ใช้คนหนึ่ง : (รหัส, ชื่อ)
- ผู้ใช้คนที่สอง : (รหัส, ที่อยู่)

- เกี่ยวข้องกับผู้ใช้งานมากที่สุด เพราะเป็นมุมมองของผู้ใช้ล้วนๆ



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

19

3. ระดับภายนอก (External Level) (ต่อ)

- โครงสร้างของข้อมูลที่ผู้ใช่มองเห็นจะแปรเปลี่ยนไปตามมุมมองของผู้ใช้
- แต่ละมุมมองจะประกอบด้วยข้อมูลในลักษณะที่แต่ละคนสนใจหรือเฉพาะที่เกี่ยวข้องกับงานของตน
- นอกจากนี้ แต่ละมุมมองอาจจะมีการนำเสนอข้อมูลเดียวกันในลักษณะที่แตกต่างกัน



สถาปัตยกรรมระบบฐานข้อมูล 3 ระดับ (ต่อ)

20

3. ระดับภายนอก (External Level) (ต่อ)

■ เช่น มุมมองรูปแบบวันที่ของผู้ใช้ 2 คน

01/11/2003
(เดือน/วัน/ปี ค.ศ.)

11 มกราคม 2546
(วัน เดือน ปี พ.ศ.)

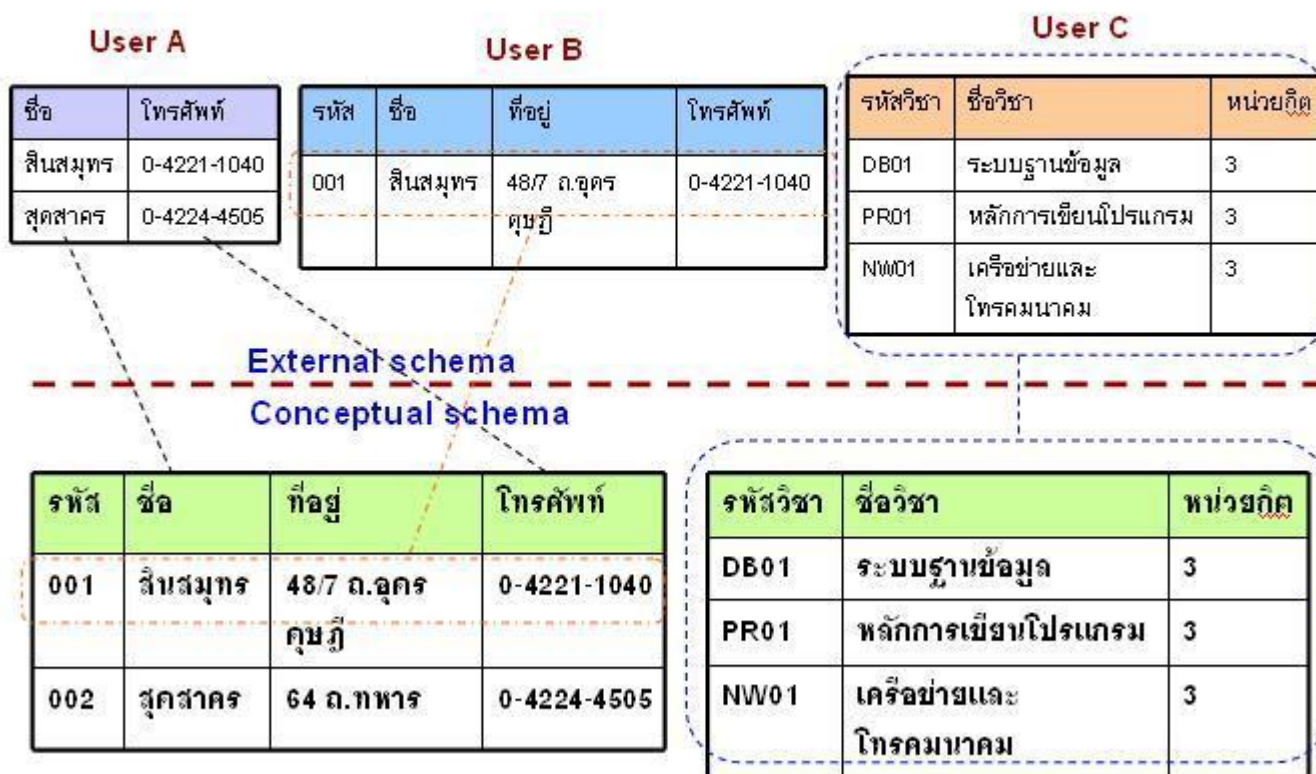
สรุป ระดับภายนอกเป็นการนำข้อมูลระดับแนวคิดมาแสดงแก่
ผู้ใช้งาน ซึ่งมุมมองการใช้งานหรือการมองเห็นข้อมูลของผู้ใช้จะไม่
เหมือนกัน



ตัวอย่าง

21

ระดับภายนอก(External level)



สถาปัตยกรรมระบบฐานข้อมูล

22

- **สรุป** วัตถุประสงค์ของการแบ่งสถาปัตยกรรมของฐานข้อมูลเป็นหลายระดับ
 - เพื่อให้ผู้ใช้แต่ละคนสามารถเข้าถึงข้อมูลชุดเดียวกัน แต่อาจมีความแตกต่างในข้อมูลที่น่าเสนอต่อผู้ใช้แต่ละคนได้
 - ผู้ใช้ไม่จำเป็นต้องปฏิบัติโดยตรงกับฐานข้อมูลในระดับภายใน ซึ่งอยู่ในชั้นระดับกายภาพ ปล่อยให้เป็นที่ของ DBMS เป็นตัวจัดการแทน
 - ผู้บริหารฐานข้อมูล (DBA) สามารถเปลี่ยนโครงสร้างการจัดเก็บข้อมูลในฐานข้อมูลได้ โดยปราศจากผลกระทบใด ๆ ต่อมุมมองของผู้ใช้งาน
 - กรณีมีการเปลี่ยนแปลงอุปกรณ์จัดเก็บข้อมูลจะไม่ส่งผลกระทบต่อโครงสร้างภายในของฐานข้อมูลใด ๆ
 - การเปลี่ยนแปลงโครงสร้างข้อมูลในระดับแนวคิดจะไม่ส่งผลกระทบต่อผู้ใช้ฐานข้อมูล



โครงสร้างฐานข้อมูล (Database schema)

23

- โครงสร้างฐานข้อมูล (Database schema) คือ รายละเอียดในภาพรวมของฐานข้อมูล โดยโครงสร้างหรือสคีมาเปรียบเสมือนพิมพ์เขียวทางเทคนิคฐานข้อมูล
- ในฐานข้อมูลมีโครงสร้างฐานข้อมูล 3 ประเภทซึ่งถูกนิยามไว้ในแต่ละระดับของสถาปัตยกรรม ได้แก่
 - 1.โครงสร้างภายนอก (External Schema บางที่เรียกว่า Subschema)
 - 2.โครงสร้างแนวคิด (Conceptual Schema)
 - 3.โครงสร้างภายใน (Internal Schema)



โครงสร้างฐานข้อมูล (Database schema) (ต่อ)

24

□ โครงสร้างฐานข้อมูล (Database schema) 3 ประเภท

- 1.โครงสร้างภายนอก (*External Schema* บางทีเรียกว่า *Subschema*) นำเสนอข้อมูลได้หลายมุมมองตามความต้องการของผู้ใช้แต่ละคน
- 2.โครงสร้างแนวคิด (*Conceptual Schema*) เกี่ยวข้องกับเอนติตี้ แอตทริบิวต์และความสัมพันธ์ร่วมกับข้อบังคับความคงสภาพของข้อมูล
- 3.โครงสร้างภายใน (*Internal Schema*) อธิบายแบบจำลองภายในของฐานข้อมูล ประกอบด้วย การนิยามการจัดเก็บรายการข้อมูล , วิธีในการนำเสนอ , ฟิลด์ข้อมูล และ คำร่างของดัชนี (Indexes) ที่ใช้

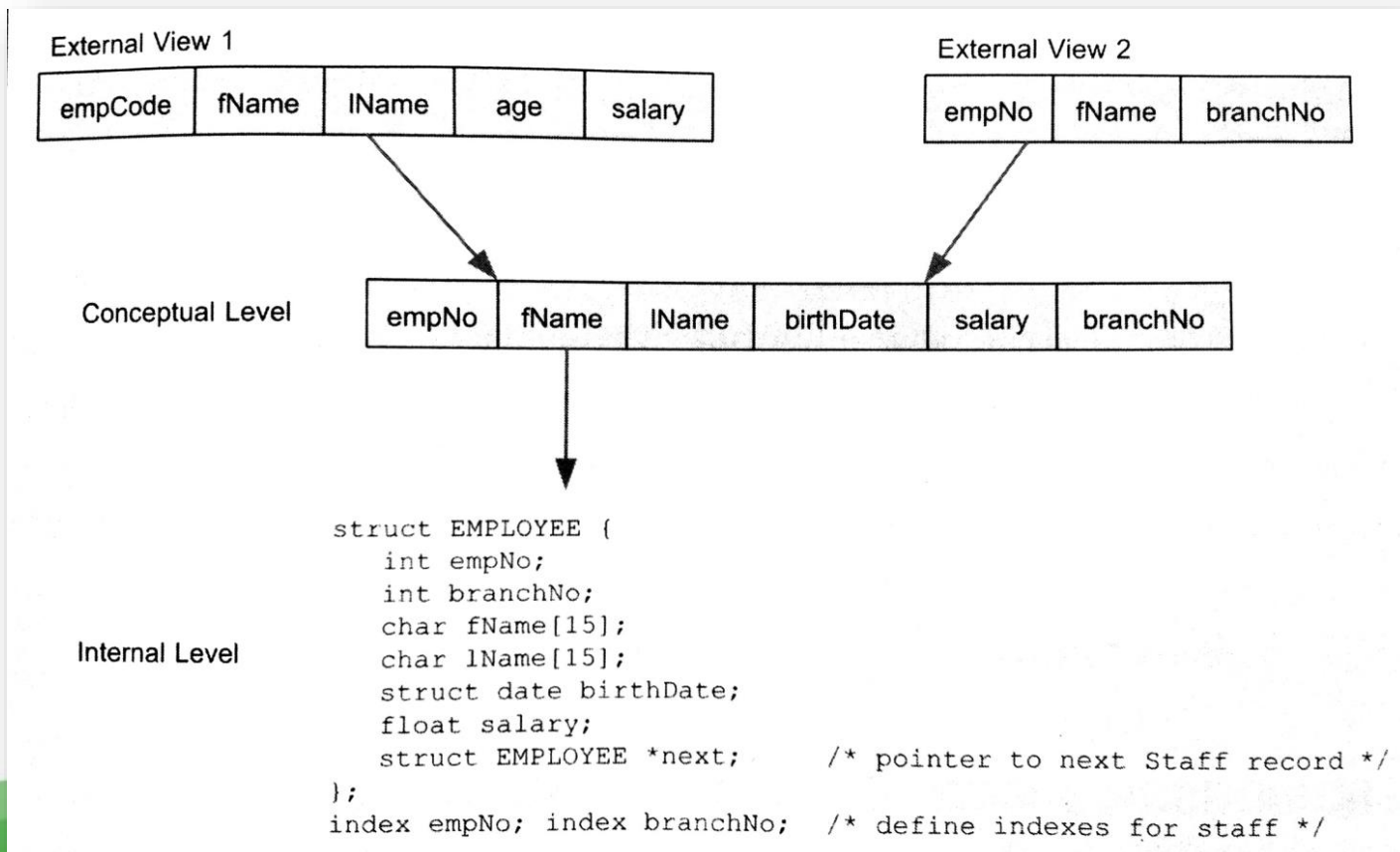
“ในฐานข้อมูลหนึ่งระบบจะมีโครงสร้างแนวคิดและโครงสร้างภายใน
อย่างละ 1 โครงสร้างต่อ 1 ฐานข้อมูล แต่โครงสร้างภายนอกสามารถมีได้หลาย
โครงสร้าง”



การแปลงความหมาย (Mapping)

25

- DBMS ทำหน้าที่ถ่ายทอดมุมมอง หรือแปลงความหมาย หรือที่เรียกว่า การทำ **Mapping** ระหว่างโครงร่าง หรือ schema ทั้ง 3 ชนิด



ความเป็นอิสระของข้อมูล (Data independence)



ความเป็นอิสระของข้อมูล (Data independence)

27

- วัตถุประสงค์ของการแบ่งสถาปัตยกรรมของระบบฐานข้อมูลออกเป็น 3 ระดับ คือ เพื่อให้เกิดความเป็นอิสระของข้อมูล (Data Independence)
- ความเป็นอิสระของข้อมูล (Data Independence) หมายความว่า ระดับที่อยู่สูงกว่าจะไม่ได้รับผลกระทบใดๆ จากความเปลี่ยนแปลงในระดับที่ต่ำกว่า



ความเป็นอิสระของข้อมูล (Data independence) (ต่อ)

28

- มีอยู่ 2 ชนิด ด้วยกัน คือ
- ความเป็นอิสระของข้อมูลทางตรรกะ
(logical data independence)
- ความเป็นอิสระของข้อมูลทางกายภาพ
(physical data independence)



ความเป็นอิสระของข้อมูล (Data independence) (ต่อ)

29

ความเป็นอิสระของข้อมูลทางตรรกะ (Logical data independence)

- หมายถึง การเปลี่ยนแปลงในโครงสร้างแนวคิดจะไม่ส่งผลกระทบต่อโครงสร้างภายนอก
- เช่น การเพิ่ม/ลบเอ็นติตี้และการเปลี่ยนแปลงแอตทริบิวต์ หรือการเปลี่ยนแปลงความสัมพันธ์ ซึ่งเป็นการกระทำบนโครงสร้างแนวคิด จะไม่ส่งผลกระทบต่อโครงสร้างภายนอก (External schema) ที่ผู้ใช้ใช้งานอยู่
- ไม่ต้องเขียนโปรแกรมใหม่
- ผู้ใช้ก็ยังคงสามารถวิวข้อมูลได้เช่นเดิม โดยอาจไม่มีความจำเป็นต้องแก้ไขโปรแกรมใด ๆ



ความเป็นอิสระของข้อมูล (Data independence) (ต่อ)

30

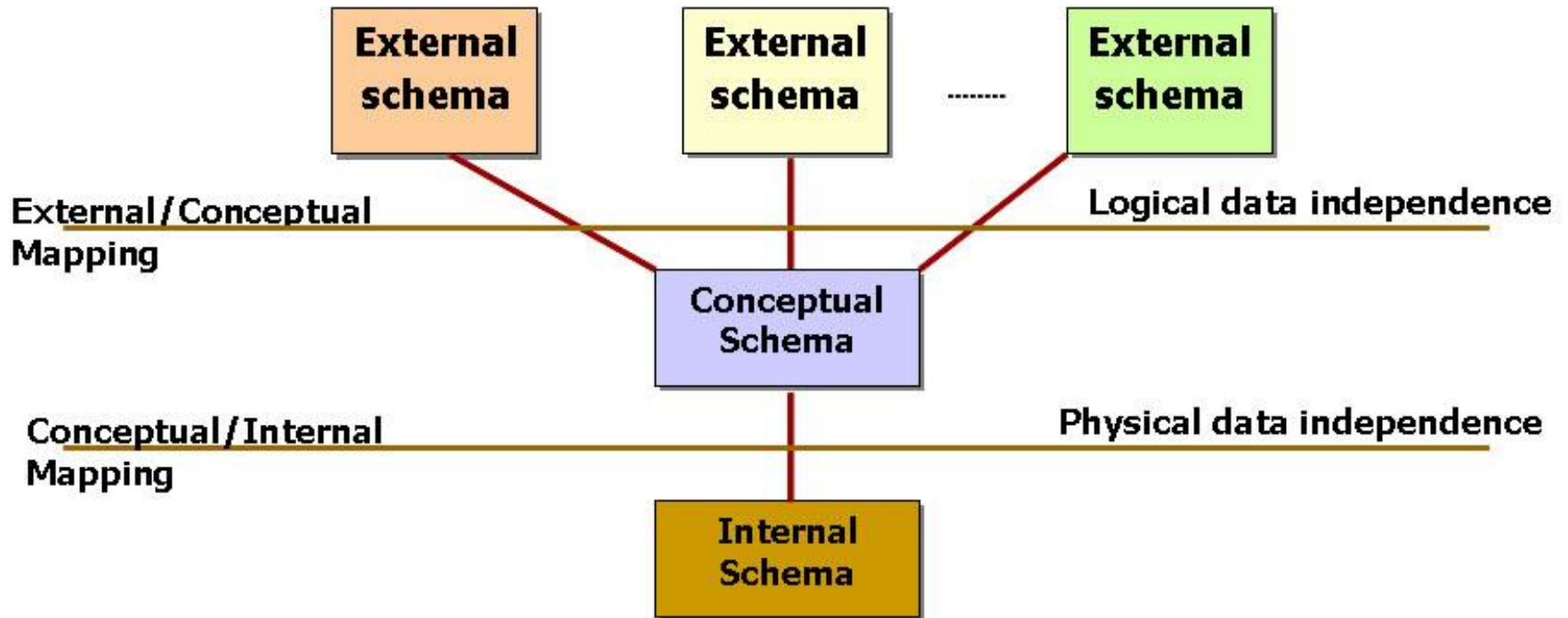
ความเป็นอิสระของข้อมูลในระดับกายภาพ (Physical data independence)

- หมายถึง การเปลี่ยนแปลงในโครงสร้างภายในไม่ส่งผลกระทบต่อโครงสร้างแนวคิดหรือโครงสร้างภายนอก
- เช่น การจัดโครงสร้างแฟ้มใหม่หรือการจัดโครงสร้างอุปกรณ์เก็บข้อมูลใหม่ เช่น การปรับปรุงดัชนี การเปลี่ยนแปลงอัลกอริทึมแฮชซึ่ง หรือการย้ายฮาร์ดดิสก์หนึ่งไปยังฮาร์ดดิสก์อีกตัวหนึ่งจะไม่ส่งผลกระทบต่อโครงสร้างแนวคิดหรือโครงสร้างภายนอก
- แต่มีผลกระทบต่อผู้ใช้งานในส่วนของการใช้งาน (แต่เป็นไปในทางที่ดีขึ้น)



ความเป็นอิสระของข้อมูล (Data independence) (ต่อ)

31



แบบจำลองข้อมูล (Data Model) หรือ แบบจำลองฐานข้อมูล (Database Model)



บทนำ

33

- **แบบจำลองข้อมูล (Data Model)** เป็นแบบจำลองนามธรรม เป็นโครงสร้างข้อมูลระดับตรรกะที่นำเสนอข้อมูลและความสัมพันธ์ระหว่างข้อมูลให้ผู้ใช้เห็นและเข้าใจได้
- เป็นแนวทางในการอธิบายแบบร่างเชิงตรรกะของข้อมูลและความสัมพันธ์ในส่วนต่าง ๆ ที่เกี่ยวข้องกัน
- เป็นแหล่งรวมของแนวคิดที่นำเสนอความเป็นจริงของวัตถุ ข้อมูล และเหตุการณ์ รวมถึงความสัมพันธ์ระหว่างข้อมูลที่มีความสอดคล้องตรงกัน

“จุดประสงค์ของแบบจำลองข้อมูล คือ การนำแนวคิดต่าง ๆ มานำเสนอให้เป็นรูปแบบจำลองขึ้นมา เพื่อใช้สื่อสารระหว่างผู้ออกแบบฐานข้อมูลกับผู้ใช้ให้เกิดความเข้าใจตรงกัน ”



ประเภทของแบบจำลองข้อมูล

34

- แบบจำลองข้อมูลแบ่งเป็น 2 ประเภท ได้แก่
 - 1. แบบจำลองเชิงแนวคิด (Conceptual Data Model)
 - 2. แบบจำลองเพื่อการนำไปใช้ (Implementation Data Model)



1. แบบจำลองเชิงแนวคิด (Conceptual Data Models)

35

- ใช้สำหรับแสดงลักษณะโดยรวมของข้อมูลทั้งหมดในระบบ โดยนำเสนอในลักษณะแผนภาพหรือไดอะแกรมที่ประกอบไปด้วยเอนิติตีต่าง ๆ และความสัมพันธ์ระหว่างเอนิติตีในระบบ
- **จุดประสงค์ของแบบจำลองเชิงแนวคิด** คือ ต้องนำเสนอให้เกิดความเข้าใจตรงกันระหว่างผู้ออกแบบและผู้ใช้งาน กล่าวคือ เมื่อเห็นแผนภาพแบบจำลองดังกล่าว ก็จะทำให้เข้าใจถึงความสัมพันธ์ของข้อมูลต่างๆ ในระบบ
- **ตัวอย่าง** เช่น แผนภาพ E-R (Entity Relationship model)



2. แบบจำลองเพื่อการนำไปใช้งาน (Implementation Data Models)

36

- เป็นแบบจำลองที่ใช้อธิบายถึงโครงสร้างข้อมูลและฐานข้อมูล ด้วยการแสดงถึงรูปแบบที่อิงกับระบบจัดการฐานข้อมูลที่ใช้

“แบบจำลองข้อมูล กับ แบบจำลองฐานข้อมูล ส่วนใหญ่มักเรียกแทนกัน”



ประเภทของแบบจำลอง ฐานข้อมูล



แบบจำลองฐานข้อมูล (Database Model)

38

1. แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model)
2. แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)
3. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)
4. แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)
5. แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)



แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical database model)

39

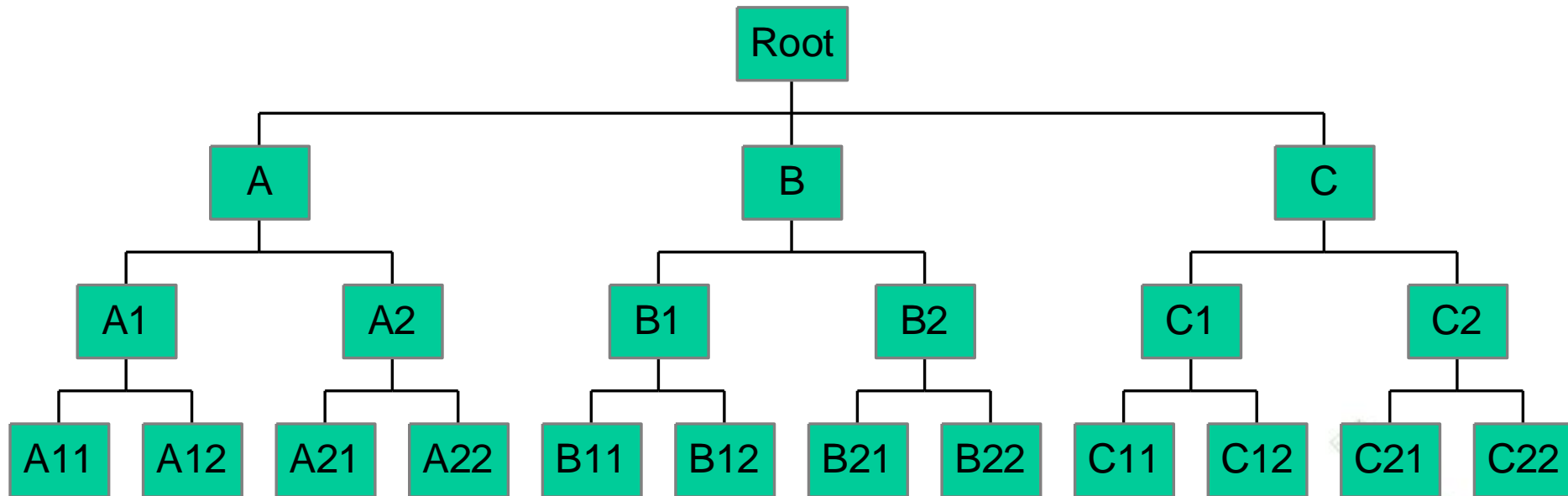
- เป็นการจัดโครงสร้างแบบบนลงล่าง (Top-down)
- มีลักษณะคล้ายโครงสร้างต้นไม้ (Tree Structure) เป็นลำดับชั้น
- ข้อมูลจะมีความสัมพันธ์แบบ one-to-many ระดับสูงสุดเรียกว่า Root
- มีความสัมพันธ์แบบ Parent / Child (พ่อ/ลูก)
- เป็นสถาปัตยกรรมฐานข้อมูลที่เก่าแก่ที่สุด ปัจจุบันไม่นิยมใช้งานแล้ว
- ยากต่อการพัฒนา Application
- การปรับปรุงโครงสร้างมีความยืดหยุ่นน้อย
- ไม่สามารถกำหนดความสัมพันธ์แบบ many-to-many



แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical database model)

40

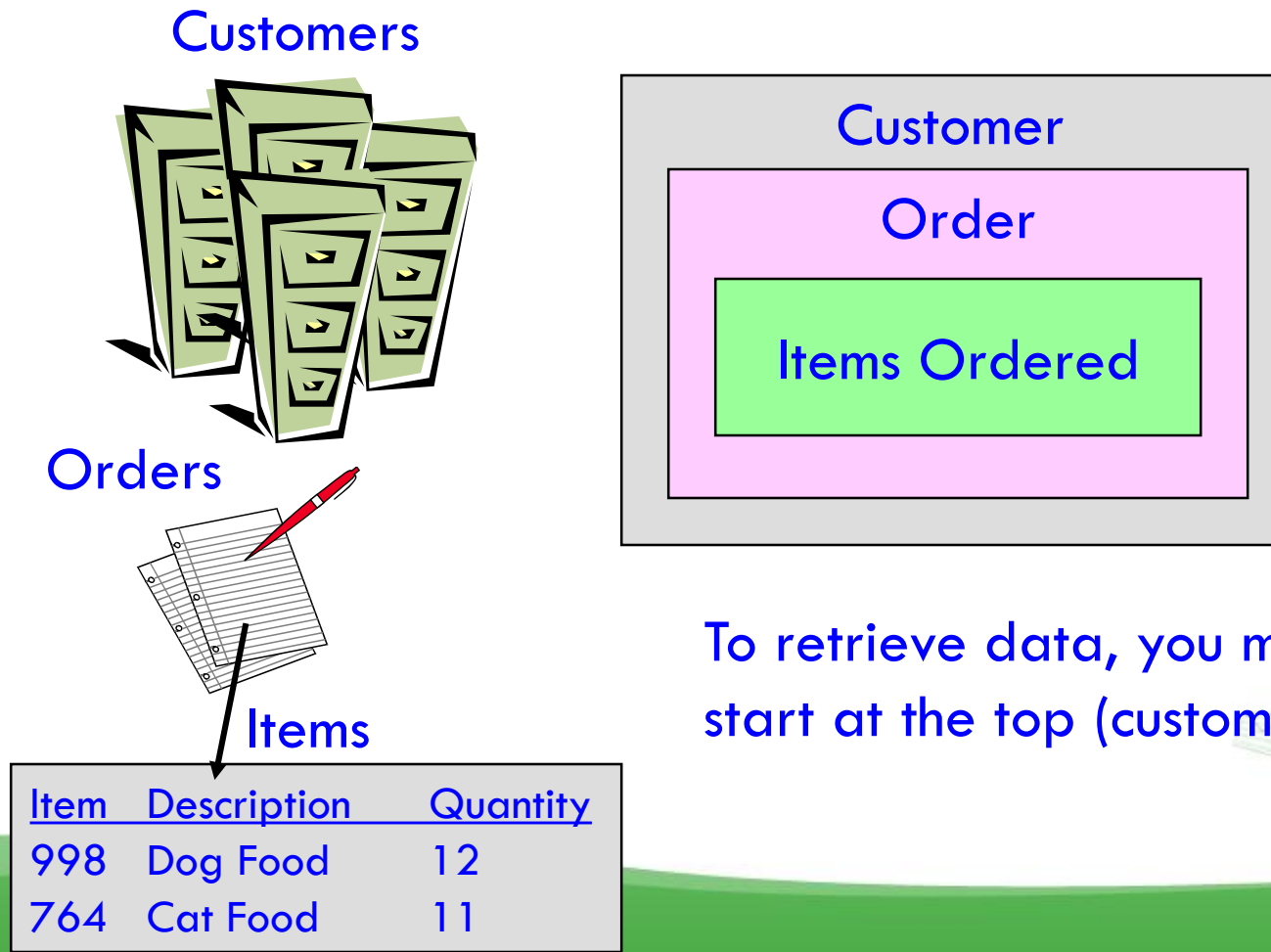
Hierarchical Data Model



แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical database model)

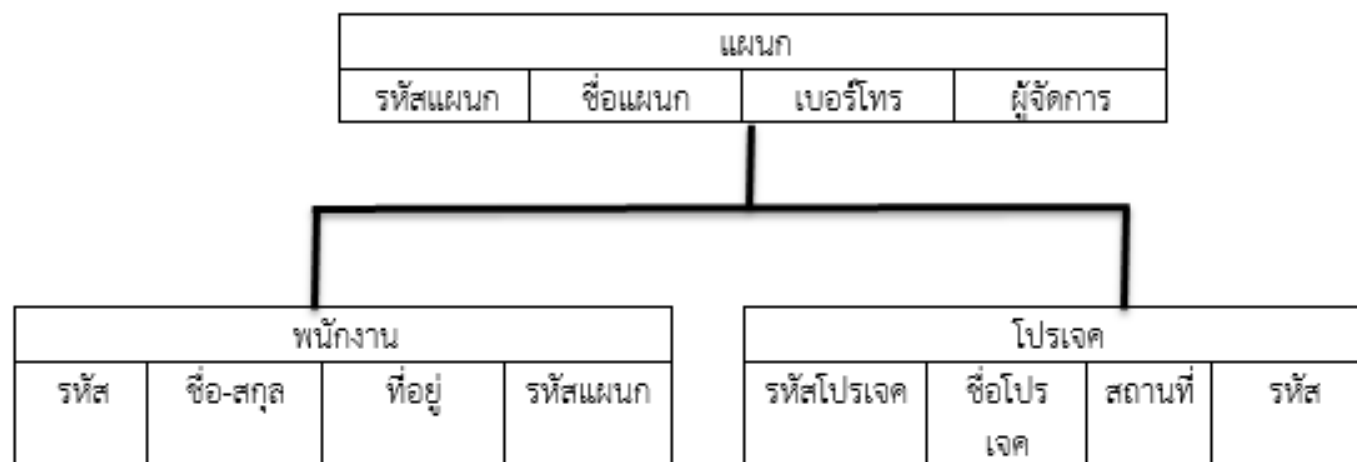
41

Hierarchical Database



To retrieve data, you must start at the top (customer).





ภาพที่ 2.3 แบบจำลองฐานข้อมูลแบบลำดับขั้น

ดัดแปลงจาก ครรชิต มาลัยวงศ์, สมลักษณ์ ละอองศรี และทัศนีย์วรรณ
ศรีประดิษฐ์ (2544, หน้า 103)



แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical database model)

43

□ ข้อดี

- มีรูปแบบโครงสร้างที่เข้าใจง่าย ซึ่งเป็นในลักษณะต้นไม้
- มีโครงสร้างที่ซับซ้อนน้อยที่สุด เหมาะกับข้อมูลที่มีความสัมพันธ์แบบ one-to-many
- ป้องกันความผิดพลาดในข้อมูลที่ดี เนื่องจากต้องอ่านข้อมูลที่เป็นต้นกำเนิดก่อนทำให้ข้อมูลมีความคงสภาพ
- เหมาะกับข้อมูลที่มีการเรียงลำดับแบบต่อเนื่อง



แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical database model)

44

ข้อเสีย

- ❑ ยากต่อการพัฒนา เพราะต้องมีความรู้ความเข้าใจถึงโครงสร้างทางกายภาพของข้อมูลที่จัดเก็บอยู่ภายในฐานข้อมูล
- ❑ มีข้อจำกัดด้านการนำไปใช้ โดยเฉพาะไม่รองรับความสัมพันธ์แบบ many-to-many
- ❑ เมื่อมีการปรับโครงสร้าง แอปพลิเคชันโปรแกรมทั้งหมดต้องเปลี่ยนแปลงตามเนื่องจากขาดอิสระในโครงสร้าง
- ❑ ในการเรียกใช้งานจำเป็นต้องผ่าน Root เสมอ ดังนั้นหากต้องการค้นหาข้อมูลซึ่งอยู่ในระดับล่าง ๆ ก็ต้องค้นหาทั้งแฟ้ม
- ❑ ไม่มีภาษาที่ใช้สำหรับการจัดการข้อมูล ใน DBMS
- ❑ ขาดมาตรฐานการรองรับที่ชัดเจน



แบบจำลองฐานข้อมูล (Database Model)

45

1. แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model)
- 2. แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)**
3. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)
4. แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)
5. แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)



แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)

46

- สร้างขึ้นมาเพื่อแก้ไขปัญหาของความสัมพันธ์แบบ M:N ที่เกิดขึ้นในแบบจำลองแบบลำดับชั้น
- ทำให้มีความสัมพันธ์ได้ทั้งแบบ one-to-one , one-to-many และ many-to-many
- สมาชิกของเซตหนึ่งๆ สามารถเป็นสมาชิกของเซตอื่นได้อีกด้วย
- โดยกลุ่มที่เรียกว่า CODASYL ซึ่งทุกEntity ที่มีความสัมพันธ์กันจะมีพอยน์เตอร์กำกับไว้
- แบบจำลองนี้ใช้ในการแก้ไขความต้องการความสัมพันธ์ของข้อมูลที่ซับซ้อนและมีประสิทธิภาพมากกว่าแบบจำลองลำดับชั้น
- แต่ยังคงขาดความเป็นมาตรฐานของฐานข้อมูล ซึ่งเป็นปัญหาต่อนักออกแบบและนักเขียนโปรแกรม เนื่องจากทำให้ใช้ร่วมกันได้ยาก

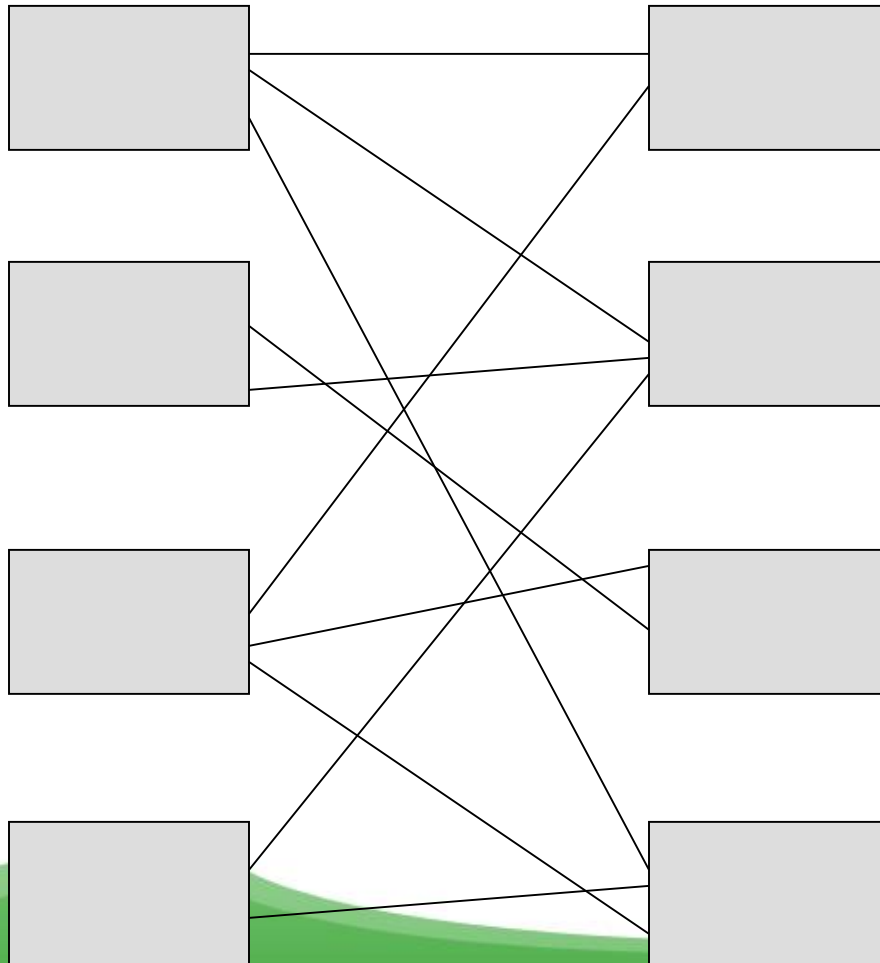


แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)

47

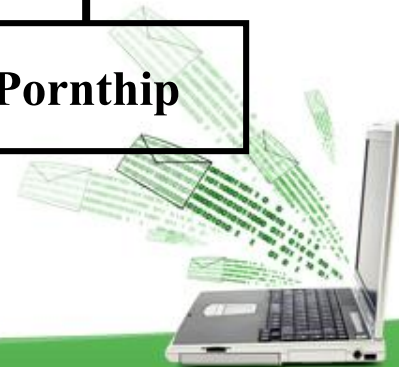
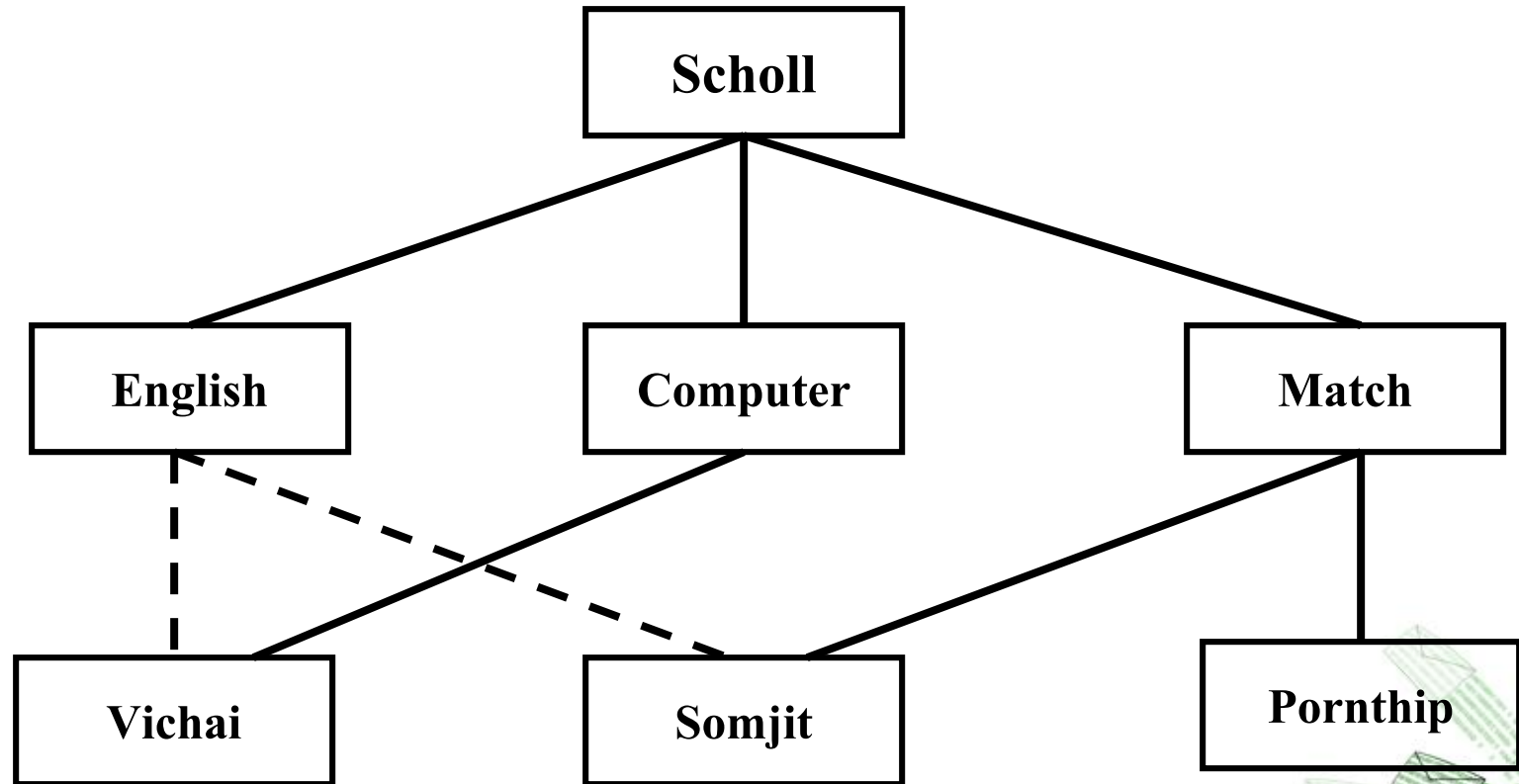
สินค้า

ลูกค้า



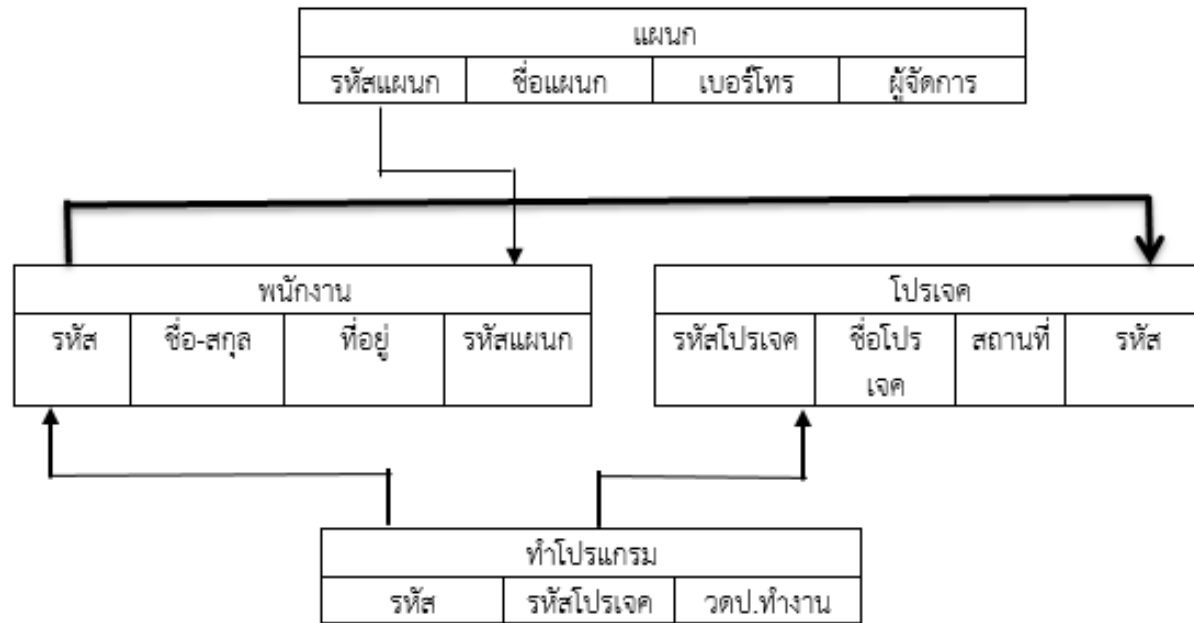
แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)

48



แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)

49



ภาพที่ 2.4 แบบจำลองฐานข้อมูลแบบเครือข่าย

ดัดแปลงจาก ครรชิต มาลัยวงศ์, สมลักษณ์ ละอองศรี และทัศนีย์วรรณ
ศรีประดิษฐ์ (2544, หน้า 110)



แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)

50

□ ข้อดี

- มีหลักการที่ง่าย ซึ่งใกล้เคียงกับแบบจำลองฐานข้อมูลลำดับชั้น
- สนับสนุนความสัมพันธ์แบบ many-to-many
- การเข้าถึงข้อมูลมีความยืดหยุ่นสูงกว่าแบบลำดับชั้นและระบบแฟ้มข้อมูล
- ความสัมพันธ์แบบ Owner/Member Relationship ทำให้ข้อมูลมีความคงสภาพที่ดี
- มีภาษานิยามข้อมูล ภาษาจัดการข้อมูลใน DBMS
- มีมาตรฐานเพื่อการนำไปปฏิบัติชัดเจน



แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)

51

❑ ข้อเสีย

- ❑ ระบบโดยรวมยังมีความซับซ้อน อีกทั้งยังมีข้อจำกัดและประสิทธิภาพ
- ❑ ยากต่อการนำไปใช้ ทั้งในด้านการพัฒนาแอปพลิเคชันและการจัดการ
- ❑ หากโครงสร้างมีการเปลี่ยนแปลง แอปพลิเคชันโปรแกรมทั้งหมดต้องเปลี่ยนตาม เนื่องจากขาดอิสระในโครงสร้าง



แบบจำลองฐานข้อมูล (Database Model)

52

1. แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model)
2. แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)
- 3. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)**
4. แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)
5. แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)



แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

53

- แบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นแบบจำลองที่นิยมใช้กันในปัจจุบัน
- เป็นผลงานของ E.F.Codd (ค.ศ. 1970)
- แบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นแบบจำลองที่มีการแสดงข้อมูลในรูปแบบของตาราง (Table) ที่ประกอบด้วย แถวหรือทูเพิล (Tuple) จำนวนหนึ่ง ซึ่งในแต่ละทูเพิล จะประกอบด้วยหลายแอตทริบิวต์ (Attributes)
- สามารถแสดงความสัมพันธ์ได้ทั้งแบบ one-to-one , one-to-many และ many-to-many
- และใช้ Key ในการอ้างอิงกับตารางอื่น (Primary key , Secondary Key)

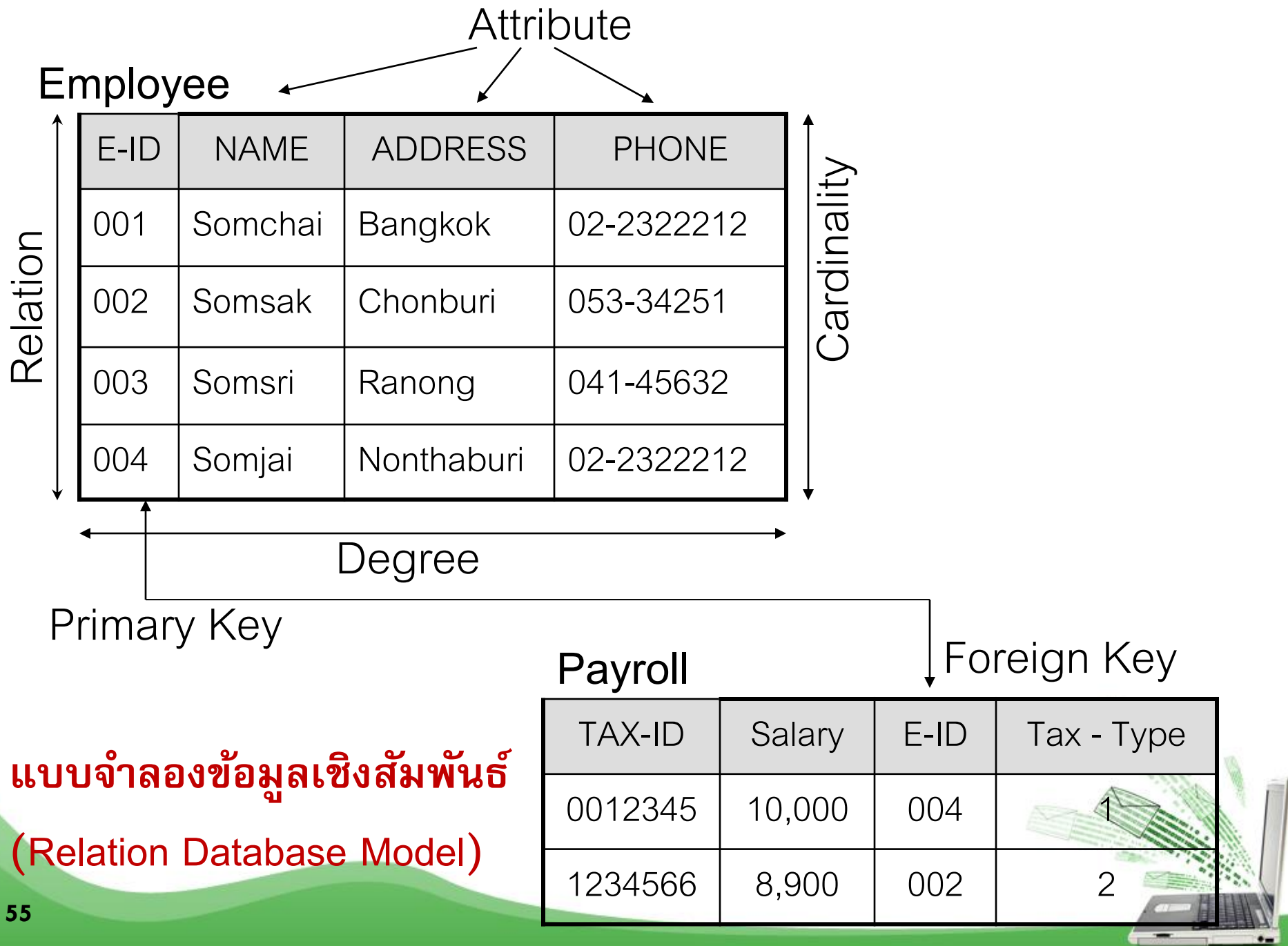


แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

54

- แบบจำลองฐานข้อมูลเชิงสัมพันธ์ จะซ่อนความซับซ้อนของระบบไว้ภายใน
- ทำให้ผู้ใช้หรือผู้ออกแบบฐานข้อมูลไม่รู้สึกถึงความยุ่งยากซับซ้อนของระบบ
- แบบจำลองฐานข้อมูลเชิงสัมพันธ์สามารถทำให้เกิดความเป็นอิสระกับข้อมูล (Data Independence) และเป็นอิสระกับโครงสร้าง (Structural Independence)
- จึงทำให้การออกแบบฐานข้อมูลเชิงสัมพันธ์สามารถทำได้ง่ายกว่าการออกแบบฐานข้อมูลแบบอื่น ๆ





ตัวอย่างแบบจำลองฐานข้อมูลเชิงสัมพันธ์

รหัสพนักงาน	ชื่อ	เงินเดือน	ตำแหน่ง	รหัสแผนก
1001	เดวิท เบคแฮม	10,000	เลขาฯ	10
3001	อลัน เชียเลอร์	17,000	พนักงานขาย	30
4001	ไมเคิล โอเวน	7,000	เสมียน	40
1002	ราฟาเอล	30,000	ผู้จัดการ	20

รหัสแผนก	ชื่อแผนก
10	บัญชี
20	บริหารทั่วไป
30	การตลาด
40	วิเคราะห์



แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

57

□ ข้อดี

- มีความเป็นอิสระในโครงสร้าง โดยหากมีการเปลี่ยนแปลงโครงสร้างตาราง จะไม่ส่งผลต่อแอปพลิเคชันโปรแกรมใช้งาน
- การนำเสนอข้อมูลในรูปแบบของตาราง ก่อให้เกิดมุมมองถึงข้อมูลที่จัดเก็บ ทำให้ง่ายต่อการออกแบบฐานข้อมูล การนำไปใช้ และการจัดการ
- การเรียกดูข้อมูล สามารถเรียกได้ด้วยชุดคำสั่ง SQL
- มีระบบความปลอดภัยที่ดี เนื่องจากโครงสร้างนี้ผู้ใช้งานจะไม่ทราบถึงกระบวนการจัดเก็บข้อมูลภายในฐานข้อมูลแท้จริงว่าเป็นอย่างไร
- DBMS ที่พัฒนาในปัจจุบันล้วนรองรับเทคโนโลยีฐานข้อมูลเชิงสัมพันธ์



แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

58

❑ ข้อเสีย

- ❑ ต้องมีการลงทุนสูงเนื่องจากต้องใช้ Hardware และ Software ที่มี
ความสามารถสูง
- ❑ แนวคิดฐานข้อมูลเชิงสัมพันธ์ในภาพรวมนั้นง่ายต่อการนำไปใช้งาน ดังนั้น
บุคลากรที่ไม่ได้รับการฝึกอบรมหรือผู้ที่มีความรู้ไม่ดีพอ ได้นำเครื่องมือไปใช้
งานที่ผิด ทำให้ระบบที่ดีต้องแย่ลง และหากไม่ได้รับการตรวจสอบ อาจทำให้
เกิดข้อมูลซ้ำซ้อนได้เช่นเดียวกับระบบแฟ้ม



แบบจำลองฐานข้อมูล (Database Model)

59

1. แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model)
2. แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)
3. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)
4. แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)
5. แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)



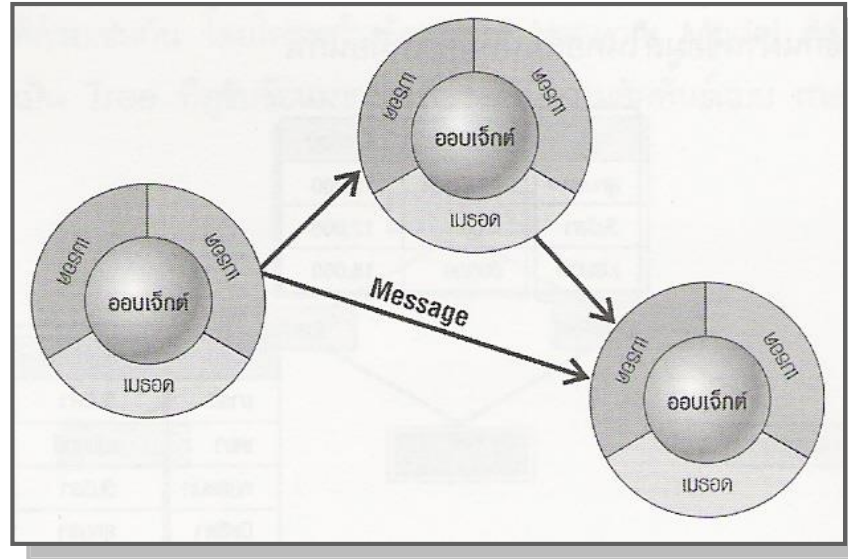
แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)

60

- เกิดจากแนวคิดของการเขียนโปรแกรมเชิงวัตถุ (OOP Object Oriented Program) โดยการมองของทุกสิ่งเป็นวัตถุ โดยแต่ละวัตถุจะเป็นแหล่งรวมของข้อมูลและการปฏิบัติงาน (Data & Procedure)
- แบบจำลองฐานข้อมูลนี้เป็นการผสมผสานกันของการสร้างแบบจำลองข้อมูลและภาษาเขียนโปรแกรมเชิงวัตถุ
- สคีมาของฐานข้อมูลเชิงวัตถุประกอบด้วย ชุดของคลาส (Class)
- โดยที่แต่ละคลาส คือ ชุดของออบเจกต์ ที่มีโครงสร้างและพฤติกรรมอย่างเดียวกัน
- โครงสร้างของออบเจกต์ถูกกำหนดโดยใช้ Property ของคลาส เช่น ลูกค้าจะมี Property ดังนี้ หมายเลขลูกค้า, ชื่อ, ที่อยู่, สถานะ



ตัวอย่างแบบจำลองฐานข้อมูลเชิงวัตถุ



เหมาะกับการจัดการข้อมูลที่ซับซ้อน (Graphic, Video, Audio)

ดูข้อมูลได้ยาก (Query) เน้นการเขียนโปรแกรมในการดูข้อมูล

เป็นเทคโนโลยีใหม่ นำไปใช้กับหน่วยงานขนาดใหญ่ ที่มีผู้เชี่ยวชาญด้านนี้



แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)

62

□ ข้อดี

- คุณสมบัติการสืบทอด Inheritance ทำให้ข้อมูลมีความคงสภาพสูง
- มีคุณสมบัติในการกลับมาใช้ใหม่
- การนำเสนอเป็นรูปแบบ Visual ทำให้อธิบายหัวข้อความหมายได้ดี



แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)

63

❑ ข้อเสีย

- ❑ ต้องพึ่งพาผู้เชี่ยวชาญโดยเฉพาะ และมีค่าใช้จ่ายระบบค่อนข้างสูง
- ❑ ยังไม่มีมาตรฐานรองรับที่ชัดเจนเมื่อเทียบกับแบบจำลองฐานข้อมูลเชิงสัมพันธ์ อีกทั้งผลิตภัณฑ์ DBMS ที่ใช้งานบนแบบจำลองฐานข้อมูลสัมพันธ์ได้พัฒนาขีดความสามารถด้วยการรวมเทคโนโลยีเชิงวัตถุเข้าไป ที่เรียกว่า Object-Relational Database
- ❑ ผู้เชี่ยวชาญส่วนใหญ่คุ้นเคยกับเทคโนโลยีเชิงฐานข้อมูลสัมพันธ์



แบบจำลองฐานข้อมูล (Database Model)

64

1. แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model)
2. แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)
3. แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)
4. แบบจำลองฐานข้อมูลเชิงวัตถุ (Object-Oriented database Model)
- 5. แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน**
(Multidimensional Database Model)



แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)

65

- แบบจำลองชนิดนี้ใช้งานกับคลังข้อมูล โดยนำเสนอข้อมูลในลักษณะไดเมนชัน (หลายมิติ) ทำให้ให้วิวข้อมูลได้ 2 ทาง เพื่อให้สามารถมองเห็นปัญหาทางธุรกิจและสร้างวิธีการแก้ไขปัญหาได้ดียิ่งขึ้น
- สามารถสร้างชนิดข้อมูลที่กำหนดเอง หรือ User defined types ได้ โดยชนิดข้อมูลที่สร้างเองนี้สามารถเก็บข้อมูลชนิดอื่นๆ ไว้ภายในได้อีก
- สามารถสร้าง Method เพื่อจัดการกับข้อมูลภายในได้อีกด้วย
- ดังนั้น ตารางจึงเก็บข้อมูลที่มีความซับซ้อนได้ตามชนิดข้อมูลที่เพิ่มขึ้นมา



แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)

66

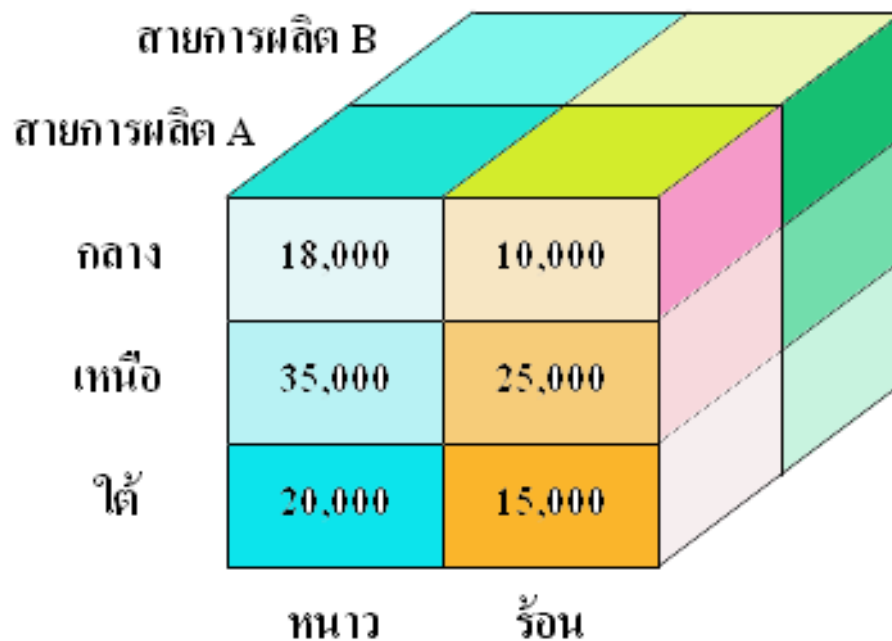
ภูมิภาค	ฤดูกาล	สายการผลิต	ยอดขาย
กลาง	หนาว	A	18,000
กลาง	หนาว	B	15,000
กลาง	ร้อน	A	10,000
กลาง	ร้อน	B	12,500
เหนือ	หนาว	A	35,000
เหนือ	หนาว	B	37,500
เหนือ	ร้อน	A	25,000
เหนือ	ร้อน	B	20,000
ใต้	หนาว	A	20,000
ใต้	หนาว	B	18,000
ใต้	ร้อน	A	15,000
ใต้	ร้อน	B	12,500



แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)

67

แสดงยอดขายในฐานข้อมูลแบบหลายมิติ



แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)

68

□ ข้อดี

- สามารถนำมาประยุกต์ใช้เพื่อวางแผนกลยุทธ์และสร้างวิธีแก้ไขปัญหาทางธุรกิจได้
- ข้อมูลที่นำเสนอสามารถนำเสนอมุมมองได้หลายมิติ



แบบจำลองฐานข้อมูลแบบมัลติไดเมนชัน (Multidimensional Database Model)

69

❑ ข้อเสีย

- ❑ ใช้เงินลงทุนสูง ทั้งฮาร์ดแวร์และซอฟต์แวร์ที่นำมาใช้เพื่อการวิเคราะห์
- ❑ คลังข้อมูลต้องได้รับการออกแบบที่ดี มิฉะนั้นอาจไม่สามารถนำมาใช้ประโยชน์
คุ้มค่า
- ❑ ผู้เชี่ยวชาญในปัจจุบันยังมีไม่มาก
- ❑ เหมาะกับธุรกิจขนาดใหญ่



แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)



แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

71

1. แนวคิดทั่วไปเกี่ยวกับแบบจำลองเชิงสัมพันธ์
2. กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)
3. พื้นฐานการสร้างแบบจำลองข้อมูลเชิงสัมพันธ์
4. คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์



แนวคิดทั่วไปเกี่ยวกับแบบจำลองเชิงสัมพันธ์

72

- วิวัฒนาการของแบบจำลองเชิงสัมพันธ์
- เดิม DBMS ส่วนใหญ่อยู่บนพื้นฐานของแบบจำลองข้อมูลเชิงลำดับชั้นและแบบจำลองข้อมูลเชิงเครือข่าย
- *E. F. Codd* เป็นผู้นำเสนอแบบจำลองข้อมูลเชิงสัมพันธ์เป็นคนแรกในปี ค.ศ. 1970
- ที่ห้องปฏิบัติการวิจัยของบริษัท IBM ในเมือง San José มลรัฐแคลิฟอร์เนีย โดยให้ชื่อระบบว่า IBM System R
- System R เป็นโครงการที่เป็นต้นกำเนิดของภาษาที่ใช้ในการจัดการฐานข้อมูลเชิงสัมพันธ์ (SQL) ที่เป็นมาตรฐานในปัจจุบัน ตลอดจนเป็นต้นแบบของระบบจัดการฐานข้อมูลเชิงพาณิชย์ เช่น IBM DB/2 และ Oracle เป็นต้น



แนวคิดทั่วไปเกี่ยวกับแบบจำลองเชิงสัมพันธ์

73

- วิวัฒนาการของแบบจำลองเชิงสัมพันธ์
- การพัฒนาแบบจำลองนี้ นำมาซึ่งการปฏิรูปครั้งใหญ่ในวงการฐานข้อมูล
- ที่เป็นไปในรูปแบบของแบบจำลองข้อมูลที่ชัดเจน สอดคล้องกับการใช้งาน และมีทฤษฎีรองรับ
- ก่อให้เกิดการศึกษา พัฒนา และใช้งานระบบจัดการฐานข้อมูลอย่างแพร่หลาย



แนวคิดทั่วไปเกี่ยวกับแบบจำลองเชิงสัมพันธ์

74

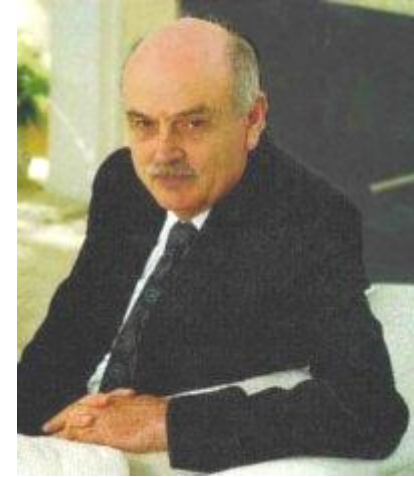
- ข้อดีของระบบจัดการฐานข้อมูลเชิงสัมพันธ์
- ระบบฐานข้อมูลเชิงสัมพันธ์ จัดเก็บข้อมูลในรูปแบบตารางและคอลัมน์ ทำให้เข้าใจง่าย
- ผู้ใช้ไม่ต้องมีความรู้เกี่ยวกับการเข้าถึงข้อมูล ก็สามารถใช้งานได้อย่างเต็มประสิทธิภาพ
- มี SQL เป็นภาษาหลัก ซึ่งเป็นภาษาที่ใช้ง่าย
- เชื่อมโยงข้อมูลได้ง่าย
- ข้อมูลมีความอิสระตามคุณสมบัติของระบบจัดการฐานข้อมูล



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

75

- ในปี ค.ศ. 1985 Edgar Frank Codd ผู้ให้กำเนิดแบบจำลองข้อมูลเชิงสัมพันธ์ได้วางกฎไว้สำหรับเป็นบรรทัดฐานของการทำงานของระบบจัดการฐานข้อมูลเชิงสัมพันธ์ต่าง ๆ เพื่อให้เป็นระบบจัดการฐานข้อมูลที่รองรับแบบจำลองเชิงสัมพันธ์โดยสมบูรณ์
- โดยกำหนดเป็นกฎ 12 ข้อ (E. F. Codd's 12 rules) ซึ่งโดยแท้จริงแล้วมี 13 ข้อดังนี้



“กฎ 12 ข้อของ E. F. Codd คือ กฎที่บัญญัติไว้สำหรับกำหนดให้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์รองรับ”



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

76

- กฎข้อที่ 0 กฎพื้นฐาน (*Foundation rule*)
- ระบบฐานข้อมูลเชิงสัมพันธ์ ต้องมีความสามารถในการจัดการฐานข้อมูลทั้งหมดโดยใช้ความสามารถเชิงสัมพันธ์ กล่าวคือ การจัดการข้อมูลใดๆ จะอ้างอิงถึงทฤษฎีเชิงสัมพันธ์เท่านั้น



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

77

- กฎข้อที่ 1 กฎสารสนเทศ (Information rule)
- สารสนเทศทั้งหมดต้องแสดงในรูปแบบของตารางที่ประกอบด้วยคอลัมน์ และแถว



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

78

- กฎข้อที่ 2 กฎการรับประกันการเข้าถึงข้อมูล (*Guaranteed access rule*)
- ข้อมูลทุกๆ ค่าที่เก็บไว้ในฐานข้อมูลจะต้องสามารถเข้าถึงได้โดยการระบุชื่อตาราง คีย์หลัก และชื่อคอลัมน์



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

79

- กฎข้อที่ 3 การรองรับค่าว่าง (*Systematic null value support*)
- ระบบจัดการฐานข้อมูลต้องรองรับค่าว่าง หรือ null value โดยค่าดังกล่าวต้องแสดงว่าข้อมูลยังคงว่างเปล่าอยู่ ต่างจากค่า 0 และมีความเป็นอิสระจากชนิดข้อมูล



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

80

- กฎข้อที่ 4 โครงสร้างของรีเลชันสามารถเรียกดูได้ (*Dynamic On-line Catalog Based on the Relational Model*)
- ทุกตาราง ทุกคอลัมน์ ที่เก็บใน DB จะต้องสามารถเรียกแสดงโครงสร้างและจัดการแก้ไขได้โดยภาษาใดๆ ที่มีโครงสร้าง และผู้ใช้งานที่ใช้งานข้อมูลจะอาศัยโครงสร้างเดียวกันนี้ในการใช้งานฐานข้อมูล



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

81

- กฎข้อที่ 5 กฎการมีภาษาที่สนับสนุนอย่างเต็มรูปแบบ
(*Comprehensive data sublanguage rule*)
- จะต้องมีการมีภาษาอย่างน้อยหนึ่งภาษาที่ออกแบบมาอย่างสมบูรณ์และสามารถจัดการฐานข้อมูลเชิงสัมพันธ์ได้เต็มความสามารถของแบบจำลอง (อาจไม่ใช่ SQL ก็ได้) โดยความสามารถที่ภาษาจะต้องรองรับได้แก่
 - 1) การนิยามข้อมูล
 - 2) การนิยามวิว
 - 3) การแก้ไขข้อมูล ได้แก่การเพิ่ม ลบ และแก้ไข
 - 4) การกำหนดกฎบูรณภาพหรือกฎเพื่อกงความถูกต้องของข้อมูล
 - 5) การกำหนดสิทธิในการเข้าใช้ฐานข้อมูล
 - 6) การจัดการธุรกรรม



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

82

- กฎข้อที่ 6 กฎการแก้ไขข้อมูลผ่านทางวิว (*View updating rule*)
- วิวทุกวิวสามารถสามารถปรับปรุงได้ผ่านระบบ
- กล่าวคือ โดยปกติวิวจะเกิดจากรีเลชัน(ตาราง)หลักในฐานข้อมูล อาจเกิดจากหลายๆ รีเลชันมารวมกัน ..เมื่อเราแก้ไขข้อมูลในวิว ข้อมูลที่สัมพันธ์กับข้อมูลในวิวนั้น ก็ต้องได้รับการแก้ไขให้ตรงกันด้วย



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

83

- กฎข้อที่ 7 มีความสามารถในการเพิ่ม ลบ และแก้ไขข้อมูล
(*High-level insertion, update, and deletion*)
- ไม่ใช่แค่การแสดงผลเท่านั้นที่ระบบจัดการฐานข้อมูลต้องรองรับ แต่ต้องรองรับการเพิ่ม การแก้ไขหรือปรับปรุง และการลบข้อมูลได้



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

84

- กฎข้อที่ 8 มีความเป็นอิสระของข้อมูลระดับกายภาพ (*Physical data independence*)
- โปรแกรมอื่นที่ใช้งานระบบจัดการฐานข้อมูลจะต้องไม่ได้รับผลกระทบกรณีมีการปรับเปลี่ยนข้อกำหนดด้านกายภาพ เช่น เมื่อมีการเปลี่ยนลำดับของไฟล์ใหม่ มีการสร้างไฟล์ดัชนีใหม่ เป็นต้น



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

85

- กฎข้อที่ 9 มีความเป็นอิสระของข้อมูลระดับตรรกะ (*Logical data independence*)
- โปรแกรมอื่นที่ใช้งานระบบจัดการฐานข้อมูลจะต้องไม่ได้รับผลกระทบกรณีมีการปรับเปลี่ยนข้อกำหนดด้านตรรกะ เช่น เมื่อมีการเปลี่ยนโครงสร้างตารางข้อมูล
- (แต่ถ้าต้องมีการจัดเก็บข้อมูลเพิ่มอีก 1 คอลัมน์ ผ่านทางหน้าจอ โปรแกรม เรายังคงต้องแก้ไขโปรแกรมประยุกต์ ให้รองรับการกรอกข้อมูลนั้น)



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

86

- กฎข้อที่ 10 มีความเป็นอิสระของบูรณภาพ (*Integrity independence*)
- ภาษาฐานข้อมูลต้องรองรับการกำหนดข้อกำหนด หรือกฎต่าง ๆ ที่บังคับให้ข้อมูลมีความถูกต้องตามข้อกำหนดทางธุรกิจหรือที่เรียกว่าบูรณภาพของข้อมูล
- ข้อกำหนดเหล่านี้ต้องถูกจัดเก็บอยู่ในระบบจัดการฐานข้อมูล และไม่สามารถที่จะละเมิดข้อกำหนดนี้ได้



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

87

- กฎข้อที่ 11 มีอิสระในการกระจาย (*Distribution independence*)
- ถึงแม้จะย้ายที่เก็บฐานข้อมูลไปไว้แบบไหน อย่างไร ผู้ใช้ต้องไม่ได้รับผลกระทบ



กฎ 12 ข้อของ E. F. Codd (E. F. Codd' 12 Rules)

88

- กฎข้อที่ 12 ไม่อนุญาตให้ภาษาในระดับต่ำกว่าเสี่ยงกฎบูรณภาพ (*Nonsubversion rule*)
- ภาษาในระดับต่ำกว่า ในที่นี้หมายถึงภาษาที่จัดการข้อมูลครั้งละข้อมูล เช่น ครั้งละระเบียน ครั้งละคอลัมน์ในระเบียน จะต้องไม่สามารถเสี่ยงกฎต่าง ๆ ที่ตั้งไว้เพื่อบูรณภาพของข้อมูลได้



พื้นฐานการสร้างแบบจำลองข้อมูลเชิงสัมพันธ์

89

1. เอนติตี้
2. แอตทริบิวต์
3. ความสัมพันธ์
4. ข้อบังคับ

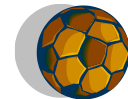


1. เอ็นทิตี (Entity)

90

- สิ่งต่าง ๆ ที่มีเอกลักษณ์สามารถชี้เฉพาะเจาะจงได้ เป็นสิ่งที่สามารถนำไปเก็บในฐานข้อมูลได้ อาจเป็นสิ่งที่เป็รูปธรรม หรือนามธรรมก็ได้ เช่น บุคคล สัตว์ สิ่งของ สถานที่ ความคิด เหตุการณ์ เป็นต้น

- ✓ **บุคคล** พนักงาน นักศึกษา นักฟุตบอล แพทย์
- ✓ **สิ่งของ** สินค้า รถยนต์ หนังสือ
- ✓ **สถานที่** สนามบิน ร้านอาหาร โกดังสินค้า
- ✓ **ความคิด** หลักสูตร แผนประกันชีวิต
- ✓ **เหตุการณ์** การแข่งขันฟุตบอล ข่าว การรักษาโรค



1. เอ็นทิตี (Entity)

91

- **เอนทิตีที่เป็นรูปธรรม** เช่น
 - เอนทิตีในมหาวิทยาลัย ได้แก่ นักศึกษา อาจารย์ อาคารเรียน ห้องเรียน ฯลฯ
 - เอนทิตีของบริษัท ได้แก่ พนักงาน สินค้า ฯลฯ
 - เอนทิตีของตู้ซ่อมรถยนต์ ได้แก่ ช่าง รถยนต์ อะไหล่ ฯลฯ เป็นต้น
- **เอนทิตีที่เป็นนามธรรม** เช่น ภาพรวมของมหาวิทยาลัย รายวิชาที่เปิดสอน งาน บริษัท เป็นต้น



2. แอททริบิวท์ (Attribute)

92

□ คือ คุณสมบัติ ลักษณะ หรือรายละเอียดของเอนทิตี

วันเดือนปีเกิด

พนักงาน

ชื่อ - นามสกุล

ที่อยู่



ส่วนสูง

ประวัติการศึกษา

น้ำหนัก

หมายเลขบัตรประจำตัวประชาชน

กรุปเลือด



2. แอททริบิวท์ (Attribute)

ร้านอาหาร



สาขา

ที่ตั้ง

เบอร์โทรศัพท์

จำนวนบริการ

ผู้จัดการร้าน

จำนวนโต๊ะ



2. แอททริบิวท์ (Attribute)

94

- **แอททริบิวท์** คือ คุณสมบัติของรีเลชั่น หรือคอลัมน์ของตารางนั้นเอง หรืออาจเทียบได้กับฟิลด์ในแฟ้มข้อมูล
- **นักศึกษา** (รหัสนักศึกษา, ชื่อ-นามสกุล, เพศ, เบอร์ติดต่อ)

รหัสนักศึกษา	ชื่อ - นามสกุล	เพศ	เบอร์ติดต่อ
4811265420	ภาณุวัฒน์ ศรีทอง	ช	066855523
4810022458	สุพรรณิ ป่อฉิม	ญ	012422115
4812212445	มีสุข ภักดีสกุล	ญ	070544551
4816445852	สุขมาลย์ ไพรีทอง	ช	065444665

แอททริบิวท์

3. ความสัมพันธ์ (Relationship)

95

□ คือ ความสัมพันธ์ระหว่างเอนิตีที่อยู่ในฐานข้อมูล

นักศึกษา กับ แผนการเรียน
ความสัมพันธ์ คือ ลงทะเบียน

ลูกค้า กับ สินค้า
ความสัมพันธ์ คือ สั่งซื้อ

อาจารย์ กับ คณะ
ความสัมพันธ์ คือ สังกัด



3. ความสัมพันธ์ (Relationship)

96

□ แบบจำลองข้อมูลจะมี **ชนิดของความสัมพันธ์อยู่ 3 ชนิด** คือ

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1:1)
2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (1:M)
3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (M:N)



3. ความสัมพันธ์ (Relationship)

97

- ความสัมพันธ์แบบ หนึ่งต่อหนึ่ง (1:1) เป็นความสัมพันธ์ที่แต่ละรายการของเอนทิตี X มีความสัมพันธ์กับข้อมูล Y เพียงหนึ่งรายการ
เช่น
 - นักศึกษามีรหัสประจำตัวเพียงหนึ่งรหัส
 - ผู้นำประเทศ – ประเทศ



3. ความสัมพันธ์ (Relationship)

98

□ ความสัมพันธ์แบบ หนึ่งต่อกลุ่ม (1:M) เป็นความสัมพันธ์ที่แต่ละรายการของเอนทิตี x มีความสัมพันธ์กับเอนทิตี y มากกว่าหนึ่งรายการ โดยแต่ละรายการของเอนทิตี y มีความสัมพันธ์กับเอนทิตี x ได้เพียงหนึ่งรายการเท่านั้น เช่น

□ แม่ – ลูก

□ อาจารย์ที่ปรึกษา – นักศึกษา

□ คณะ-โปรแกรมวิชา



3. ความสัมพันธ์ (Relationship)

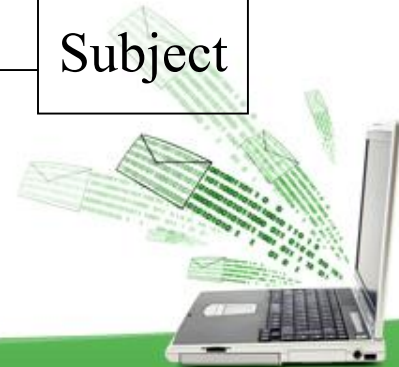
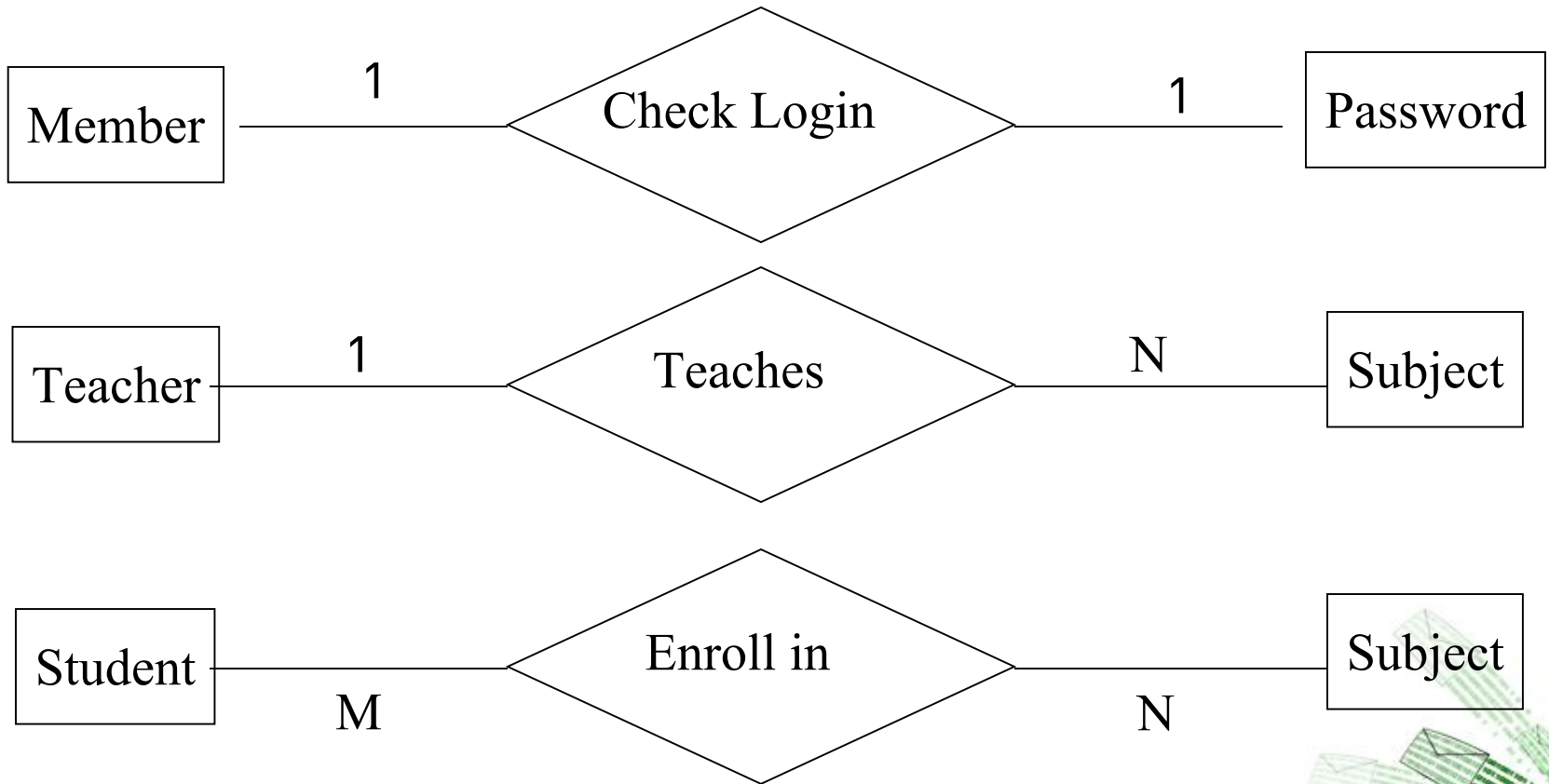
99

- ความสัมพันธ์แบบ กลุ่มต่อกลุ่ม (M:N) เป็นความสัมพันธ์ที่แต่ละรายการของเอนทิตี x มีความสัมพันธ์กับเอนทิตี y มากกว่าหนึ่งรายการ โดยแต่ละรายการของเอนทิตี y มีความสัมพันธ์กับเอนทิตี x ได้มากกว่าหนึ่งรายการ เช่น
 - นักศึกษา-รายวิชา
 - ลูกค้า – สินค้า



3. ความสัมพันธ์ (Relationship)

100



4. ข้อบังคับ (Constraints)

101

- กฎเกณฑ์เพื่อการบรรจุข้อมูล ซึ่งมีความสำคัญมาก เพราะจะช่วยให้เกิดความมั่นใจในความเป็นอันหนึ่งอันเดียวกัน เพื่อให้เกิดความสอดคล้องตรงกันของข้อมูล เช่น
 - เงินเดือนพนักงานจะต้องมีค่าระหว่าง 6,000-15,000
 - นักศึกษาสามารถลงทะเบียนเรียนได้หลายหน่วยแต่รวมแล้วต้องไม่เกิน 21 หน่วยกิต



คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์

102

- รีเลชัน (Relation)
- รีเลชัน คือ ตารางข้อมูล ที่ประกอบไปด้วยคอลัมน์/สดมภ์ (Columns) และแถว (Rows)



คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์

103

- แอทริบิวต์ (Attribute)
- แอทริบิวต์ คือ คอลัมน์ในตาราง ใช้อธิบายคุณสมบัติต่าง ๆ ของเอนทิตี



คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์

104

- โดเมน (Domain)
- คือ การกำหนดขอบเขตของค่าที่เป็นไปได้ให้กับข้อมูลในแต่ละ Attribute เพื่อป้องกันไม่ให้เกิดการป้อนข้อมูลที่เกินขอบเขตที่กำหนด
- เช่น การกำหนดให้อายุพนักงานมากกว่า 18 ปี, การกำหนดให้จำนวนเงินเดือนของพนักงานมากกว่า 0, การกำหนดให้เพศของพนักงานแต่ละคนต้องมีค่าเป็นชาย(M)หรือหญิง(F)



ตัวอย่าง โดเมน (Domain)

Domain

M,F

> 18

> 0

<u>EmpID</u>	Name	Sex	Age	Salary	DeptID
E001	Peter	M	24	55,000	D01
E002	Nicole	F	30	12,000	D02

<u>DeptID</u>	DName
D01	Sale
D02	Marketing



คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์

106

- ทูเพิล (Tuple)
- ทูเพิล คือ แถวของข้อมูลในตาราง (หรือ แถว (row) หรือ เรคคอร์ด/ระเบียน (record))
- คือชุดของข้อมูล 1 ข้อมูลของรีเลชันนั้นๆ



คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์

107

- คีย์หลัก (Primary Key)
- คีย์หลัก คือ แอทริบิวต์หรือกลุ่มของแอทริบิวต์ที่สามารถระบุ uniquely ได้
อย่างเฉพาะเจาะจง



คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์

108

- ดีกรี (Degree)
- ดีกรี คือ จำนวนแอทริบิวต์ในรีเลชัน (จำนวนคอลัมน์ในตาราง) ดีกรีอาจชี้ให้เห็นถึงความละเอียดของรายการข้อมูลก็ได้
- คาร์ดินัลลิตี (Cardinality)
- คาร์ดินัลลิตี คือ จำนวนของแถวในรีเลชัน คาร์ดินัลลิตีชี้ให้เห็นถึงจำนวนรายการข้อมูลในตารางหนึ่ง ๆ (จำนวนแถวในตาราง)



แนวคิดทั่วไปเกี่ยวกับแบบจำลองเชิงสัมพันธ์

109

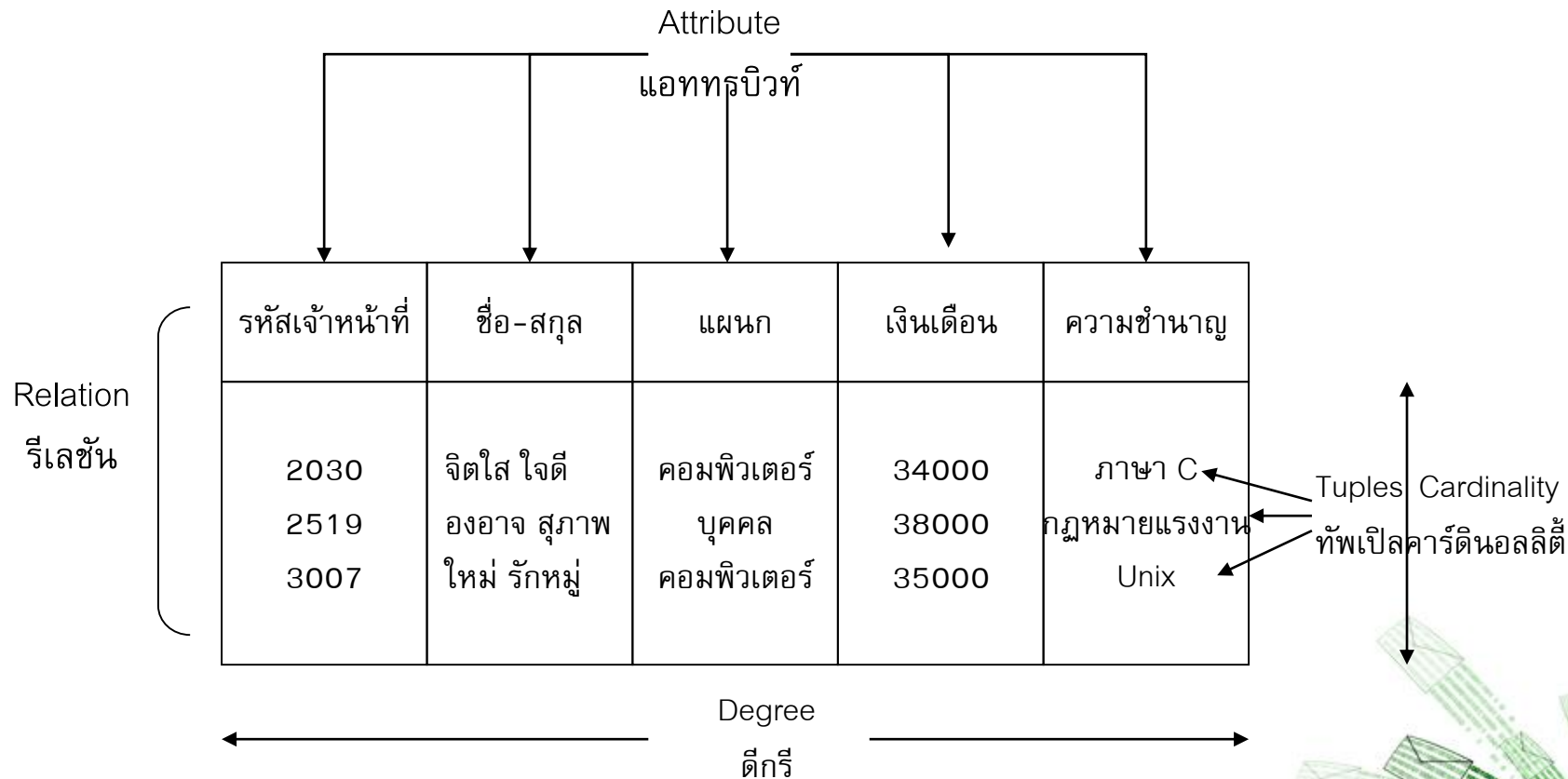
□ สรุป คำศัพท์ที่เกี่ยวข้องกับแบบจำลองเชิงสัมพันธ์

ศัพท์เฉพาะ	ศัพท์ทั่วไป
รีเลชัน (relation)	ตาราง (table)
ทูเพิล (tuple)	แถว (row) หรือ เรคคอร์ด/ระเบียน (record)
แอทริบิวต์ (attribute)	คอลัมน์/สดมภ์ (column) หรือ ฟیلด์ (field)
คาร์ดินัลลิตี (cardinality)	จำนวนแถว (number of rows)
ดีกรี (degree)	จำนวนแอทริบิวต์ (number of attributes)
คีย์หลัก (primary key)	ค่าเอกลักษณ์ (unique identifier)
โดเมน (domain)	ขอบเขตของข้อมูล (pool of legal values)



แบบจำลองฐานข้อมูลเชิงสัมพันธ์

(Relational Database Model)



องค์ประกอบของรีเลชัน