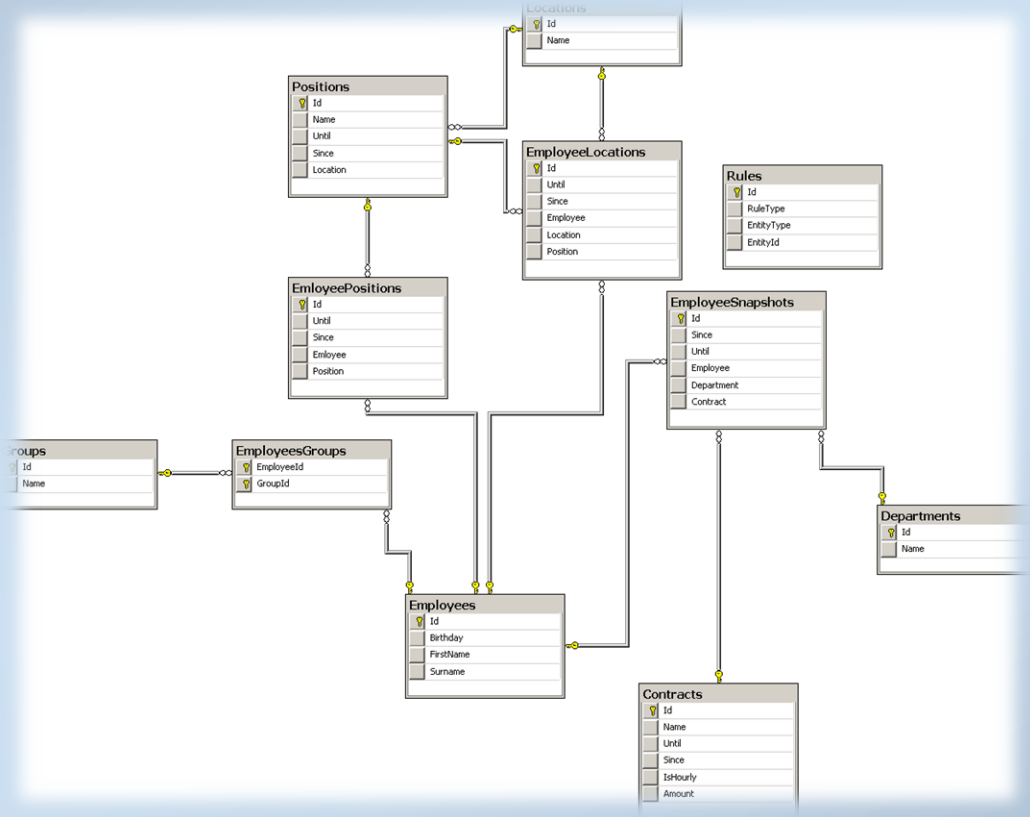
บทที่ 3 แบบจำลองของฐานข้อมูล (Database Model)

โดย อ.อภิพงศ์ ปิงยศ

รายวิชา สธ312 ระบบการจัดการฐานข้อมูลทางธุรกิจ

Overview

- แบบจำลองของข้อมูล (Data Model)
- ประเภทของความสัมพันธ์ของข้อมูล
- แบบจำลองของฐานข้อมูล (Database Model)
- Hierarchical Database Model
- Network Database Model
- Relational Database Model



บทนำ

- การนำเสนอสิ่งต่าง ๆ ที่อยู่ในรูปแบบของแนวความคิด ปกตินิยมใช้แบบจำลอง (Model)
- แบบจำลองฐานข้อมูล (Database Model) ถูกนำมาใช้ในการอธิบายรายละเอียดต่าง ๆ ของการออกแบบฐานข้อมูลและโครงสร้างของฐานข้อมูล

แบบจำลองของข้อมูล (Data Model)

- ใช้อธิบายโครงสร้างและความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูล
- สามารถอธิบายแนวความคิดที่เข้าใจยากยาก ให้สามารถเข้าใจและจับต้องได้ง่ายยิ่งขึ้น
- แบบจำลองของข้อมูลประกอบด้วย 3 ส่วนหลัก คือ
  - ข้อมูลต่าง ๆ ที่ประกอบกันเป็นฐานข้อมูล
  - กฎต่าง ๆ ที่ใช้ควบคุมความถูกต้องของข้อมูล
  - การกระทำต่าง ๆ ที่ใช้งานร่วมกับฐานข้อมูล

แบบจำลองของข้อมูล (Data Model) [cont.]

แบบจำลองข้อมูล แบ่งได้เป็น 2 ประเภท คือ

- **Conceptual Model** เป็นแบบจำลองที่นำมาใช้ออกแบบฐานข้อมูล เพื่อแสดงให้เห็นข้อมูลและความสัมพันธ์ แบบจำลองของฐานข้อมูลนิยมใช้ Entity-Relationship Model (E-R Model) และ Object-Oriented Model
- **Implementation Model** ใช้อธิบายโครงสร้างของฐานข้อมูลแต่ละประเภท เช่น Hierarchical DB Model, Network DB Model และ Relational DB Model

ประเภทของความสัมพันธ์ของข้อมูล

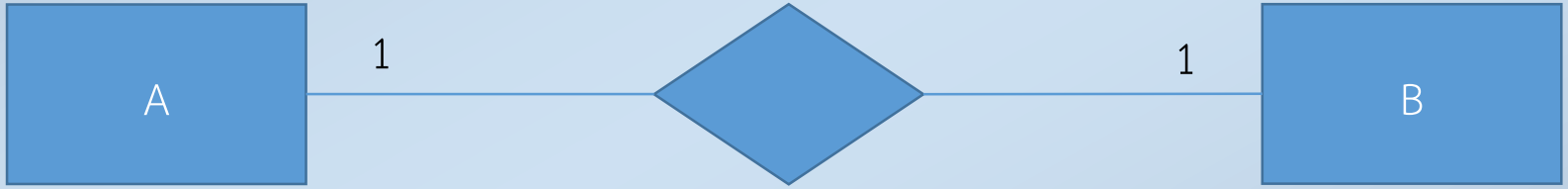
ความสัมพันธ์ของข้อมูลแบ่งออกเป็น 3 ลักษณะ ได้แก่

- One-to-One
- One-to-Many
- Many-to-Many

ประเภทของความสัมพันธ์ของข้อมูล

One-to-One

- แต่ละรายการของข้อมูล A มีความสัมพันธ์กับข้อมูล B เพียงรายการเดียว เช่น สามีสามารถมีภรรยาได้เพียงคนเดียว และภรรยาก็สามารถมีสามีได้คนเดียวเช่นกัน (ถ้าไม่นอกใจ)



ตัวอย่างที่ 1 กรณีลูกค้าสามารถมีบัญชีเงินฝากได้เพียงบัญชีเดียว และบัญชีเงินฝากสามารถมีเจ้าของบัญชีได้เพียงคนเดียว

CUSTOMER

NAME	ADDRESSS	ACC_NO
แพง พลเมืองดี	111 บางพลัด กทม.	1111111111
จิราพร สมตน	222 บางซื่อ กทม.	2222222222
สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	3333333333
กิตติ มั่นคง	444 บางบอน กทม.	4444444444
สมชาย ตั้งเจริญ	555 ลาดพร้าว กทม.	5555555555

ACCOUNT

ACC_NO	BALANCE
1111111111	5,400
2222222222	12,000
3333333333	14,000
4444444444	125,558
5555555555	100,000

ตัวอย่างที่ 1 กรณีลูกค้าสามารถมีบัญชีเงินฝากได้เพียงบัญชีเดียว และบัญชีเงินฝากสามารถมีเจ้าของบัญชีได้เพียงคนเดียว [cont.]

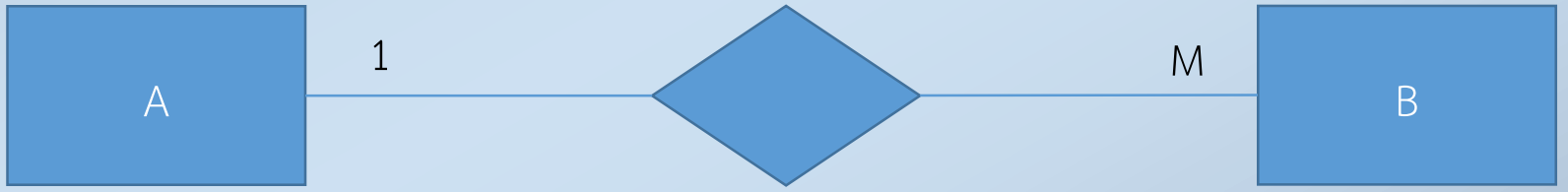
สามารถสร้างความสัมพันธ์ได้ดังนี้

NAME	ADDRESSS	ACC_NO	BALANCE
แพง พลเมืองดี	111 บางพลัด กทม.	1111111111	5,400
จิราพร สมตน	222 บางซื่อ กทม.	2222222222	12,000
สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	3333333333	14,000
กิตติ มั่นคง	444 บางบอน กทม.	4444444444	125,558
สมชาย ตั้งเจริญ	555 ลาดพร้าว กทม.	5555555555	100,000

ประเภทของความสัมพันธ์ของข้อมูล

One-to-Many

- แต่ละรายการของข้อมูล A มีความสัมพันธ์กับข้อมูล B มากกว่าหนึ่งรายการ
- เช่น ทีมฟุตบอล 1 ทีม สามารถมีผู้เล่นได้หลายคน แต่ผู้เล่นสามารถสังกัดทีมฟุตบอลได้เพียง 1 ทีมเท่านั้น (ไม่รวมทีมชาติ)



ตัวอย่างที่ 2 กรณีลูกค้าสามารถมีบัญชีเงินฝากได้มากกว่าหนึ่งบัญชี

CUSTOMER

NAME	ADDRESSS	ACC_NO
แพง พลเมืองดี	111 บางพลัด กทม.	1111111111
แพง พลเมืองดี	111 บางพลัด กทม.	1111122222
จิราพร สมตน	222 บางซื่อ กทม.	2222222222
สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	3333333333
กิตติ มั่นคง	444 บางบอน กทม.	4444444444
สมชาย ตั้งเจริญ	555 ลาดพร้าว กทม.	5555555555

ACCOUNT

ACC_NO	BALANCE
1111111111	5,400
1111122222	58,000
2222222222	12,000
3333333333	14,000
4444444444	125,558
5555555555	100,000

ตัวอย่างที่ 2 กรณีลูกค้าสามารถมีบัญชีเงินฝากได้มากกว่าหนึ่งบัญชี [cont.]

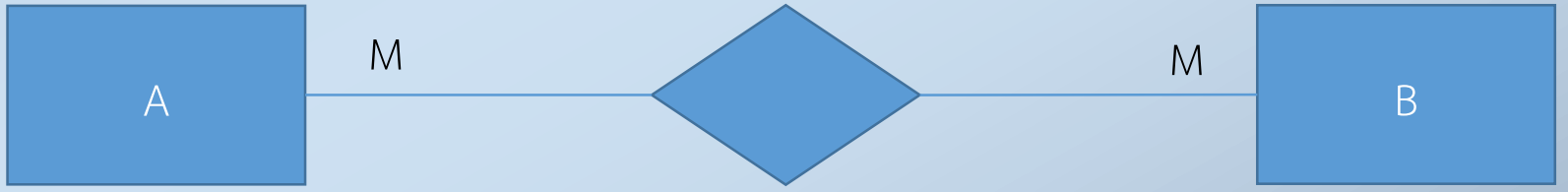
สามารถสร้างความสัมพันธ์ได้ดังนี้

NAME	ADDRESSS	ACC_NO	BALANCE
แพง พลเมืองดี	111 บางพลัด กทม.	1111111111	5,400
		1111122222	58,000
จิราพร สมตน	222 บางซื่อ กทม.	2222222222	12,000
สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	3333333333	14,000
กิตติ มั่นคง	444 บางบอน กทม.	4444444444	125,558
สมชาย ตั้งเจริญ	555 ลาดพร้าว กทม.	5555555555	100,000

ประเภทของความสัมพันธ์ของข้อมูล

Many-to-Many

- แต่ละรายการของข้อมูล A มีความสัมพันธ์กับข้อมูล B มากกว่าหนึ่งรายการ ในขณะเดียวกัน แต่ละรายการของข้อมูล B ก็มีความสัมพันธ์กับข้อมูล A มากกว่าหนึ่งรายการเช่นเดียวกัน
- เช่น นักศึกษา 1 คนสามารถลงทะเบียนได้มากกว่า 1 รายวิชา และใน 1 รายวิชาก็สามารถมีนักศึกษาลงทะเบียนได้มากกว่า 1 คนเช่นกัน



ตัวอย่างที่ 3 กรณีลูกค้าสามารถมีบัญชีเงินฝากได้มากกว่าหนึ่งบัญชี และแต่ละบัญชีสามารถมีเจ้าของบัญชีได้มากกว่าหนึ่งคน (บัญชีกลุ่ม)

CUSTOMER

NAME	ADDRESSS	ACC_NO
แพง พลเมืองดี	111 บางพลัด กทม.	1111111111
แพง พลเมืองดี	111 บางพลัด กทม.	1111122222
จิราพร สมตน	222 บางซื่อ กทม.	1111122222
สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	3333333333
กิตติ มั่นคง	444 บางบอน กทม.	4444444444
สมชาย ตั้งเจริญ	555 ลาดพร้าว กทม.	5555555555

ACCOUNT

ACC_NO	BALANCE
1111111111	5,400
1111122222	58,000
3333333333	14,000
4444444444	125,558
5555555555	100,000

ตัวอย่างที่ 3 กรณีลูกค้าสามารถมีบัญชีเงินฝากได้มากกว่าหนึ่งบัญชี และแต่ละบัญชีสามารถมีเจ้าของบัญชีได้มากกว่าหนึ่งคน (บัญชีกลุ่ม) *[cont.]*

สามารถสร้างความสัมพันธ์ตามมุมมองจากบัญชี ได้ดังนี้

ACC_NO	NAME	ADDRESSS	BALANCE
1111122222	แพง พลเมืองดี	111 บางพลัด กทม.	58,000
	จิราพร สมตน	222 บางซื่อ กทม.	
1111111111	แพง พลเมืองดี	111 บางพลัด กทม.	5,400
3333333333	สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	14,000
4444444444	กิตติ มั่นคง	444 บางบอน กทม.	125,558
5555555555	สมชาย ตั้งเจริญ	555 ลาดพร้าว กทม.	100,000

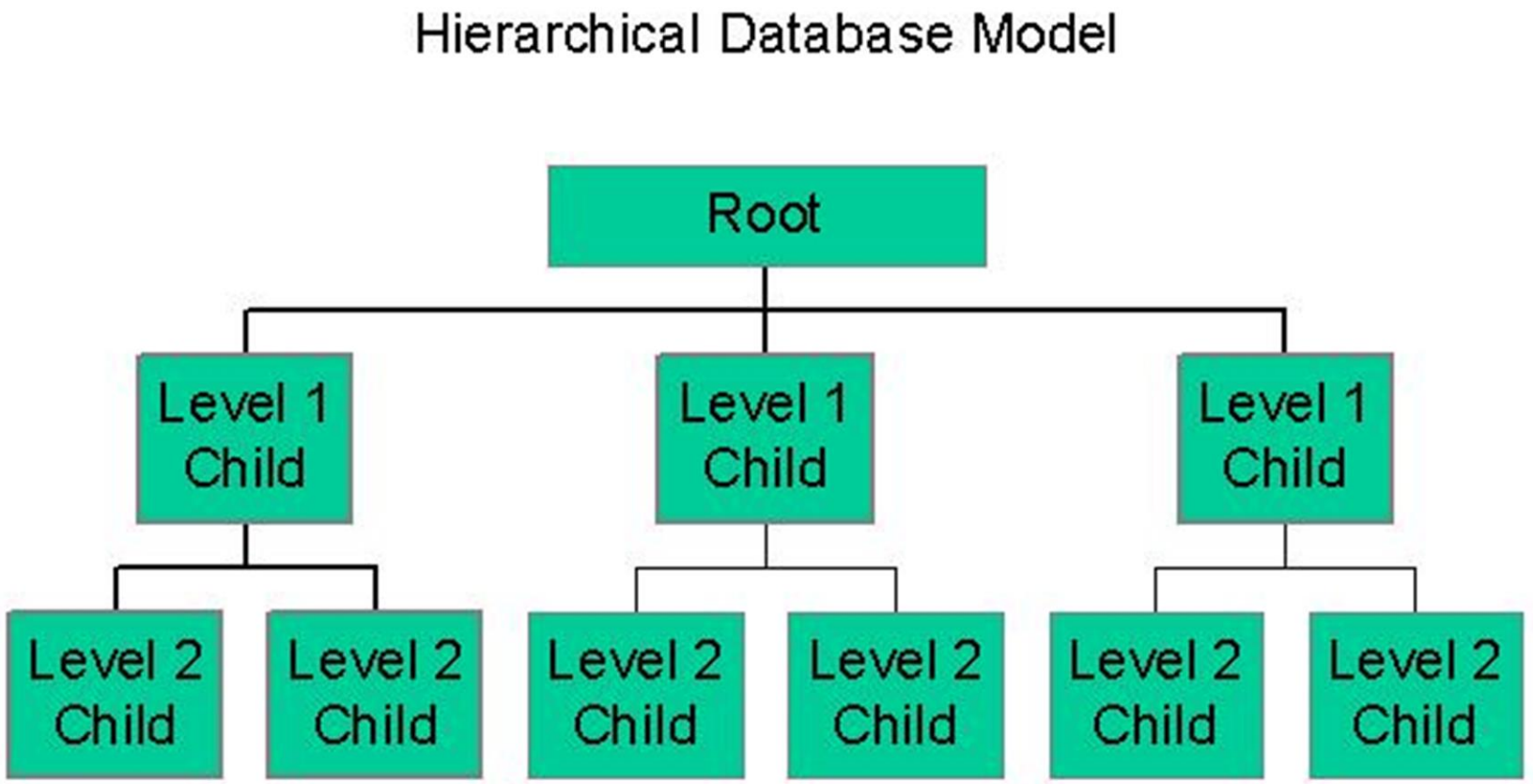
แบบจำลองของฐานข้อมูล (Database Model)

- ใช้ในการนำเสนอรายละเอียดของโครงสร้างข้อมูลของฐานข้อมูล
- มี 2 แบบ ได้แก่
  - แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model)
  - แบบจำลองฐานข้อมูลแบบเครือข่าย (Network Database Model)
  - แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

Hierarchical Database Model

- เป็นโครงสร้างข้อมูลที่อยู่ในรูปแบบลำดับชั้น (Hierarchy) หรือ Tree
- มีการใช้ Pointer เป็นตัวชี้ที่ใช้อ้างอิงตำแหน่งของข้อมูล
- เป็นโครงสร้างข้อมูลที่ใช้สร้างฐานข้อมูลของยานอวกาศอพอลโลที่นำมนุษย์คนแรกไปสู่ดวงจันทร์

Hierarchical database model



ข้อดีของ Hierarchical Database Model

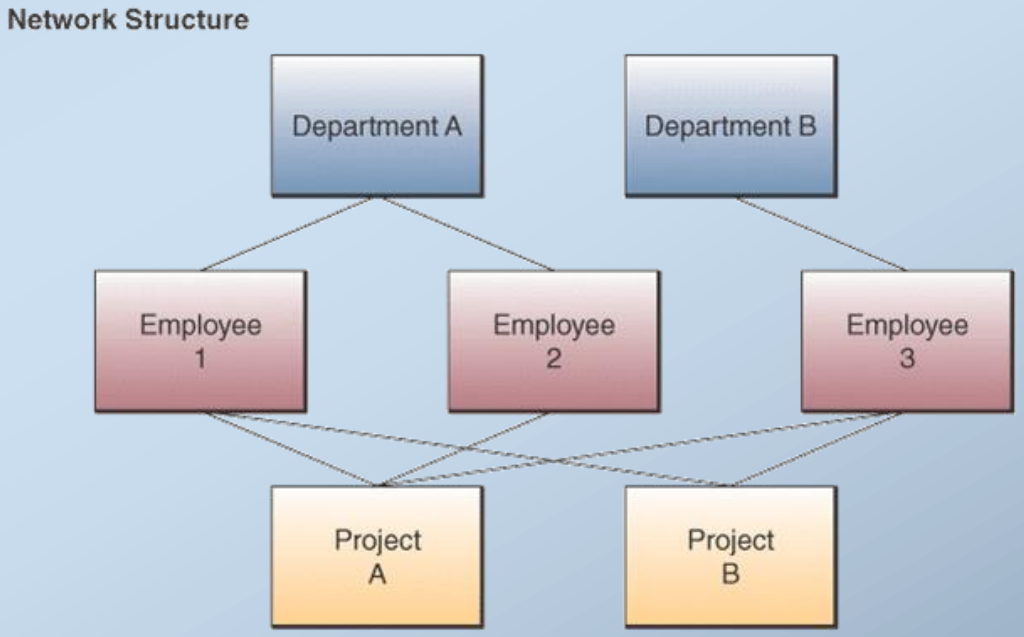
- สามารถกำหนดกฎเกณฑ์ที่ใช้ควบคุมความถูกต้องของข้อมูลได้ง่าย เพราะมีโครงสร้างแบบ Tree ซึ่ง Parent Node สามารถถ่ายทอดไปยัง Children Node ได้
- มีโครงสร้างที่เหมาะกับความสัมพันธ์ข้อมูลแบบ One-to-Many
- มีโครงสร้างที่เหมาะสมกับระบบคอมพิวเตอร์ที่มีขนาดใหญ่ เช่น ระบบ Mainframe

ข้อเสียของ Hierarchical Database Model

- ผู้ใช้ฐานข้อมูลจะต้องทราบถึงโครงสร้างของข้อมูลที่จัดเก็บอยู่ในฐานข้อมูล จึงจะสามารถเรียกใช้ข้อมูลได้ ถึงแม้จะมี DBMS ก็ตาม แต่ DBMS ในโครงสร้างแบบนี้พัฒนาได้ยาก
- ไม่สามารถรองรับข้อมูลที่มีความสัมพันธ์แบบ Many-to-Many ได้ เนื่องจาก Children Node สามารถมี Parent Node ได้เพียงโหนดเดียวเท่านั้น
- การเปลี่ยนแปลงโครงสร้างกระทำได้ยาก เพราะโครงสร้างแบบ Tree มีความยืดหยุ่นน้อย
- โครงสร้างไม่รองรับการทำงานแบบ Ad-Hoc
- ไม่มีรูปแบบที่เป็นมาตรฐาน ส่งผลให้การถ่ายโอนข้อมูลต่างองค์กรกระทำได้ยาก

Network Database Model

- พัฒนาต่อมาจาก Hierarchical Database Model โดยมุ่งหวังให้เป็นโครงสร้างข้อมูลมาตรฐาน มีความยืดหยุ่น และสามารถรองรับความสัมพันธ์แบบ Many-to-Many ได้
- จุดเด่นคือ Children Node สามารถมี Parent Node ได้มากกว่า 1 ตัว



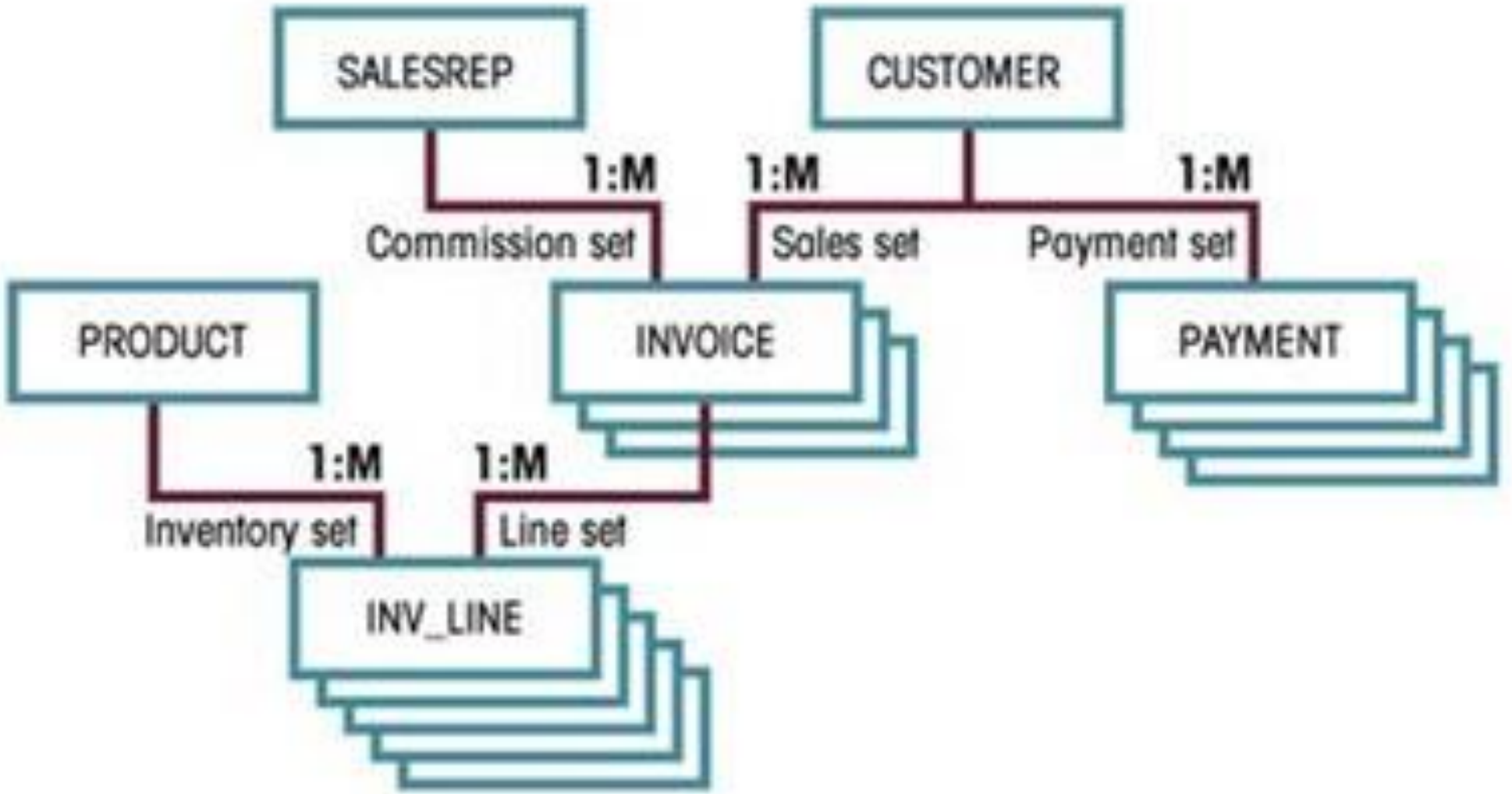


FIGURE 1.10 A NETWORK DATABASE MODEL

ข้อดีของ Network Database Model

- รองรับข้อมูลที่มีความสัมพันธ์แบบ Many-to-Many
- สามารถเข้าถึงข้อมูลใน record อื่น ๆ ได้รวดเร็วกว่า Hierarchical DB Model
- สามารถกำหนดกฎเกณฑ์ที่ใช้ควบคุมความถูกต้องของข้อมูลได้ง่าย
- โปรแกรม DBMS จะมีความเป็นอิสระจากฐานข้อมูลมากกว่าการใช้โครงสร้างแบบ Hierarchical DB Model เพราะโครงสร้างมีความซับซ้อนน้อยกว่า

ข้อเสียของ Network Database Model

- การออกแบบฐานข้อมูลจะกระทำได้ค่อนข้างยาก เพราะต้องกำหนด Set ของความสัมพันธ์ให้ครอบคลุมทุก ๆ ข้อมูล
- การเปลี่ยนแปลงโครงสร้างของข้อมูลกระทำได้ยาก เพราะต้องคำนึงถึง Set ของความสัมพันธ์ต่าง ๆ ที่กำหนดไว้

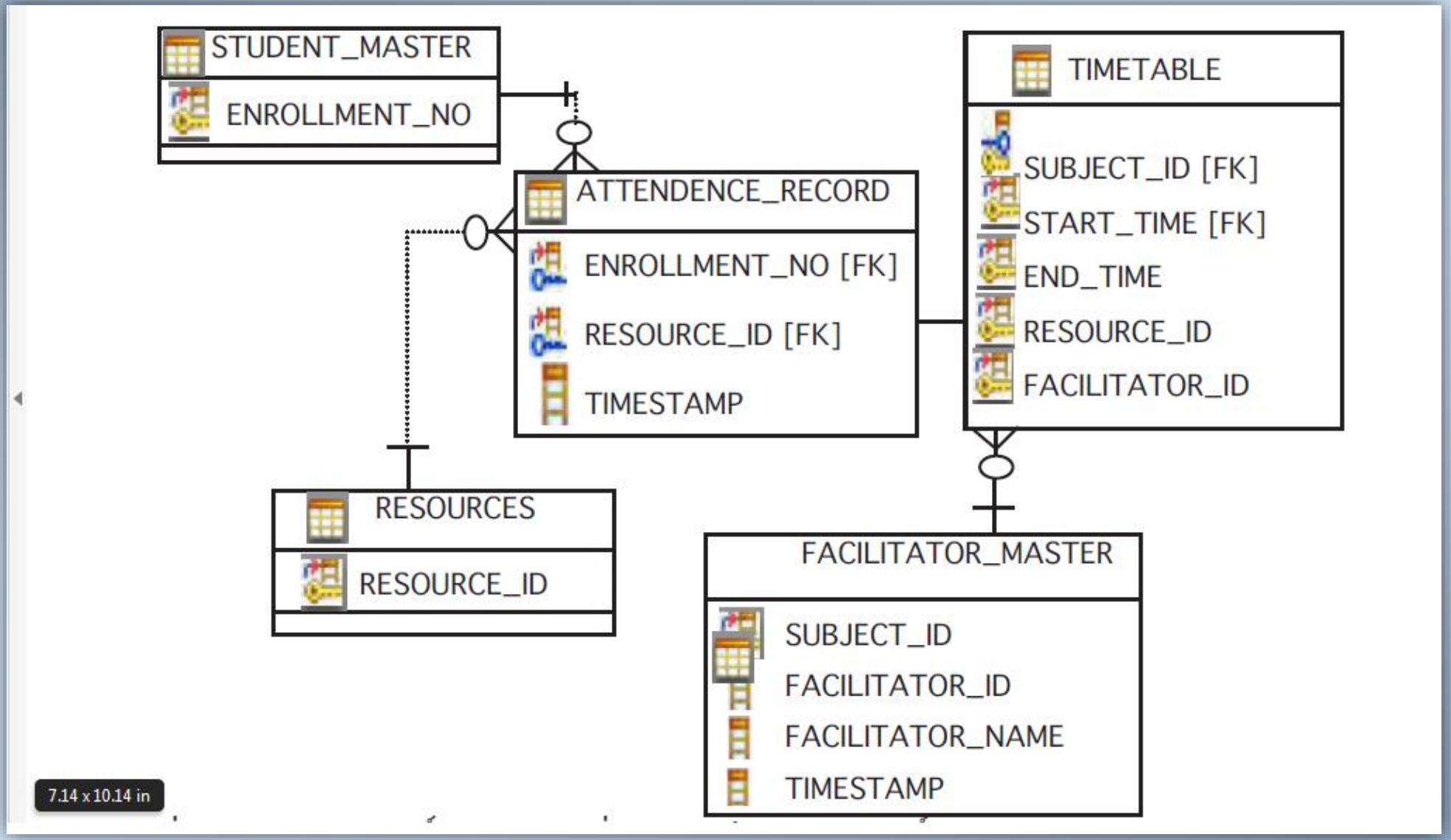
Relational Database Model

- จัดเป็นแบบจำลองที่ก่อให้เกิดการปฏิวัติระบบฐานข้อมูลขึ้น เนื่องจากเป็นโครงสร้างฐานข้อมูลที่สามารถใช้งานได้อย่างแพร่หลาย ไม่ว่าจะเป็นคอมพิวเตอร์ส่วนบุคคล ไปจนถึงคอมพิวเตอร์ที่มีประสิทธิภาพสูง
- รูปแบบของข้อมูลประเภทนี้จะมีลักษณะเป็นตาราง (Table) ที่ประกอบไปด้วยแถวและคอลัมน์ ซึ่งข้อมูลจะแยกออกจากกันเป็นเอกเทศที่มีความเป็นอิสระ แต่สามารถนำมาสร้างความสัมพันธ์ร่วมกันได้
- รายละเอียดทั้งหมดจะกล่าวถึงในบทต่อไป

Relational Database Model [cont.]

- จุดประสงค์หลักในการสร้างแบบจำลองเชิงสัมพันธ์ ประกอบไปด้วย
  - จัดเก็บข้อมูลบนโครงสร้างข้อมูลที่ง่าย (ตาราง)
  - สามารถเข้าถึงการประมวลผลแบบกลุ่ม (set-at-a-time) โดยใช้ภาษาจัดการข้อมูล (DML) ซึ่งเป็นภาษาระดับสูง
  - เป็นอิสระจากการจัดเก็บทางกายภาพ

ตัวอย่างการออกแบบฐานข้อมูลเชิงสัมพันธ์ โดยใช้ E-R Diagram



ข้อดี-ข้อเสียของ Relational Database Model

ข้อดี

- ทั้งข้อมูลและโครงสร้างมีความเป็นอิสระจากโปรแกรมที่ใช้
- มี DBMS ให้เลือกใช้งานอย่างหลากหลาย

ข้อเสีย

- ใช้ทรัพยากรจาก Hardware และ Operating System ค่อนข้างสูง เนื่องจาก DBMS ของฐานข้อมูลที่มีโครงสร้างแบบ Relational จะทำหน้าที่จัดการข้อมูลภายในฐานข้อมูลแทนผู้ใช้

สรุป

- แบบจำลองข้อมูล (Data Model) ใช้สำหรับการอธิบายโครงสร้างและความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูล จากรูปแบบแนวความคิดให้อยู่ในรูปแบบที่สามารถเข้าใจได้ง่าย
- ความสัมพันธ์ข้อมูลมี 3 แบบคือ One-to-One, One-to-Many และ Many-to-Many
- แบบจำลองฐานข้อมูล (Database Model) ใช้อธิบายและออกแบบโครงสร้างของฐานข้อมูล มี 3 แบบ คือ Hierarchy, Network และ Relational

สรุป [cont.]

- โครงสร้างข้อมูลรูปแบบต่าง ๆ สามารถสรุปได้ดังตารางนี้

	Data Independence*	Structure Independence**
File Systems	No	No
Hierarchical Database	Yes	No
Network Database	Yes	No
Relational Database	Yes	Yes

* Data Independence คือความเป็นอิสระของข้อมูลจากโปรแกรมที่ใช้

** Structure Independence คือความสามารถในการแก้ไขโครงสร้างของข้อมูลโดยไม่กระทบต่อ DBMS