



บทที่ 2

สถาปัตยกรรมและแบบจำลองข้อมูล



สถาปัตยกรรมของฐานข้อมูล

เป็นการอธิบายถึงรูปแบบและโครงสร้างของข้อมูลภายในระบบฐานข้อมูลโดยทั่วไป ในระดับแนวความคิด โดยไม่ขึ้นอยู่กับโครงสร้างของฐานข้อมูลนั้นๆ

สถาปัตยกรรมแบบ ANSI/SPARC

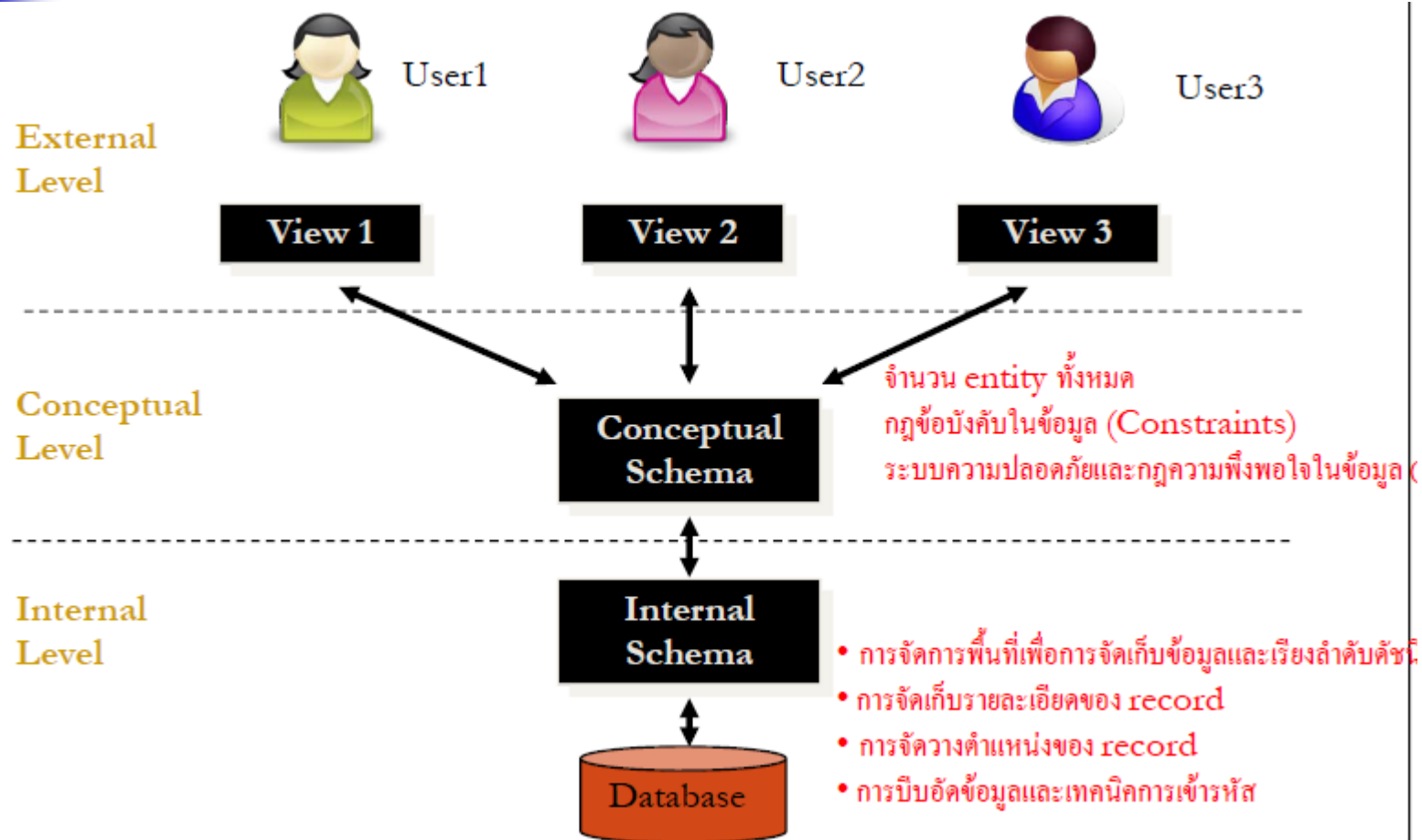
ปี 1975 (American National Standards Institute / Systems Planning And Requirements Committee) แบ่งสถาปัตยกรรมของฐานข้อมูลออกเป็น 3 ระดับคือ



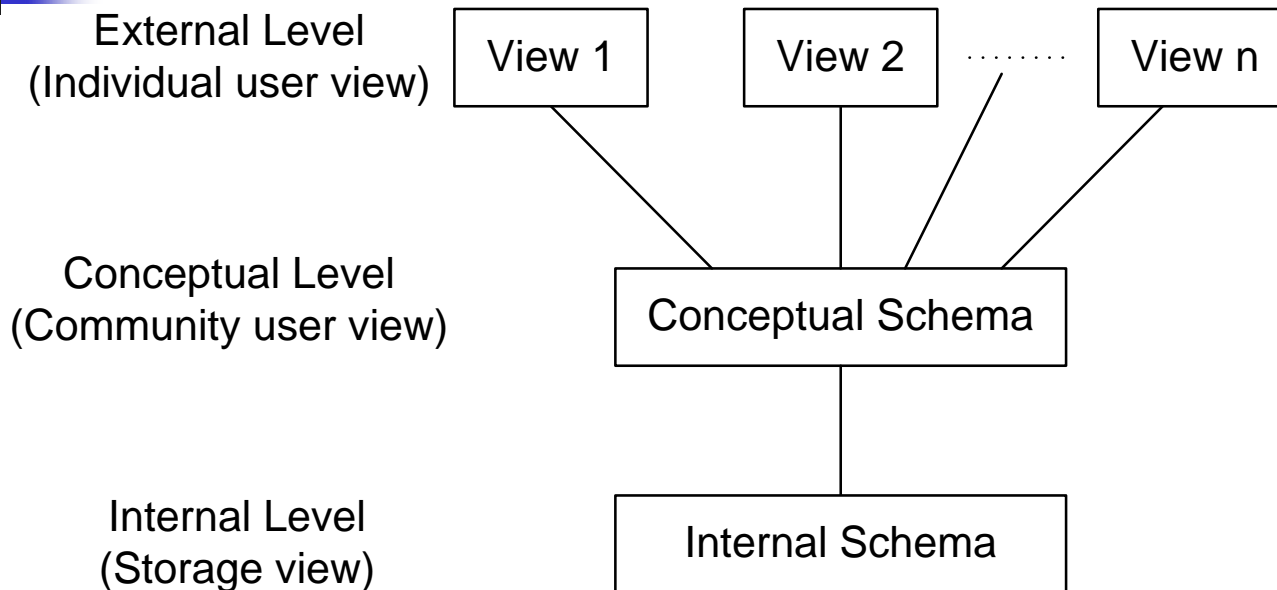
การแบ่งระดับของข้อมูล

1. **ระดับ Internal** เป็นระดับล่างสุด มีการทำงานเกี่ยวกับวิธีการจัดเก็บข้อมูล วิธีการเข้าถึงข้อมูลอย่างไร เช่น โครงสร้างแบบเรียงลำดับดัชนี, B tree, pointer
2. **ระดับ Conceptual** เป็นระดับกลาง จะเกี่ยวกับการอธิบายข้อมูลว่า ประกอบด้วยข้อมูลอะไร ข้อมูลมีความสัมพันธ์กันอย่างไร มีข้อกำหนดเงื่อนไขสำหรับ บำรุงรักษาของข้อมูล (Data Integrity) อะไรบ้าง
3. **ระดับ External** เป็นระดับบนสุด จะเกี่ยวข้องกับการสร้าง view ของข้อมูล ให้ตรงตามความต้องการใช้งานของแต่ละ User

Tree Level Architecture

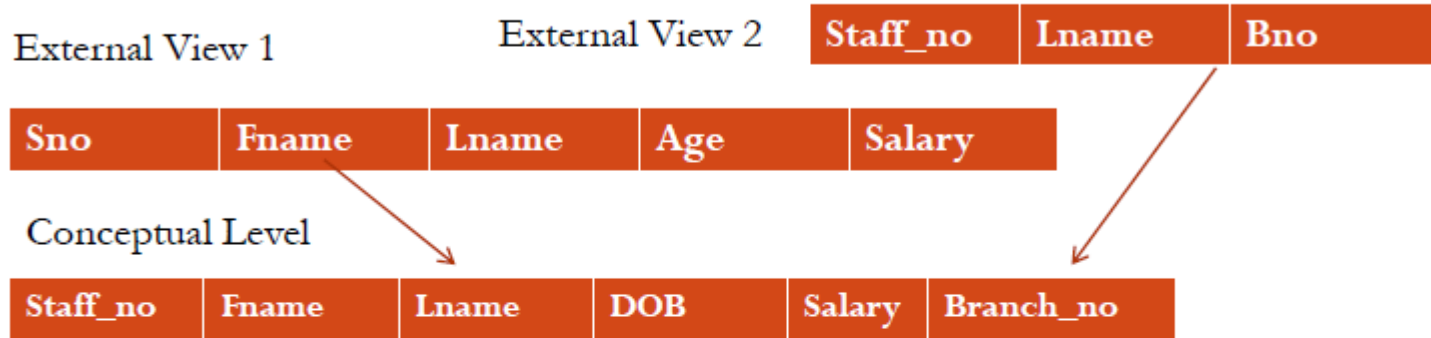


สถาปัตยกรรมแบบ ANSI/SPARC



สถาปัตยกรรมแบบ ANSI/SPARC

รูปแสดงความแตกต่างของระดับทั้งสาม



Internal Level

```
Struc STAFF {  
    int Staff_no;  
    int Branch_no;  
    char Fname [15];  
    char Lname[15];  
    struct date Date_of_birth;  
    float Salary;  
    Sturct Staff *next ; }
```



สกีมา (Schema)

- **สกีมา (Schema)** ของระบบฐานข้อมูล คือภาพโดยรวมของระบบฐานข้อมูลในเชิงตรรกะ ที่บ่งบอกให้ทราบฐานข้อมูลประกอบด้วยข้อมูลอะไรบ้าง และมีความสัมพันธ์กันอย่างไร
- **สับสกีมา (sub-schema)** มุมมองที่ผู้ใช้ฐานข้อมูลมองเห็นได้ เป็นบางส่วนของสกีมาฐานข้อมูล



ความเป็นอิสระของข้อมูล (Data Independence)

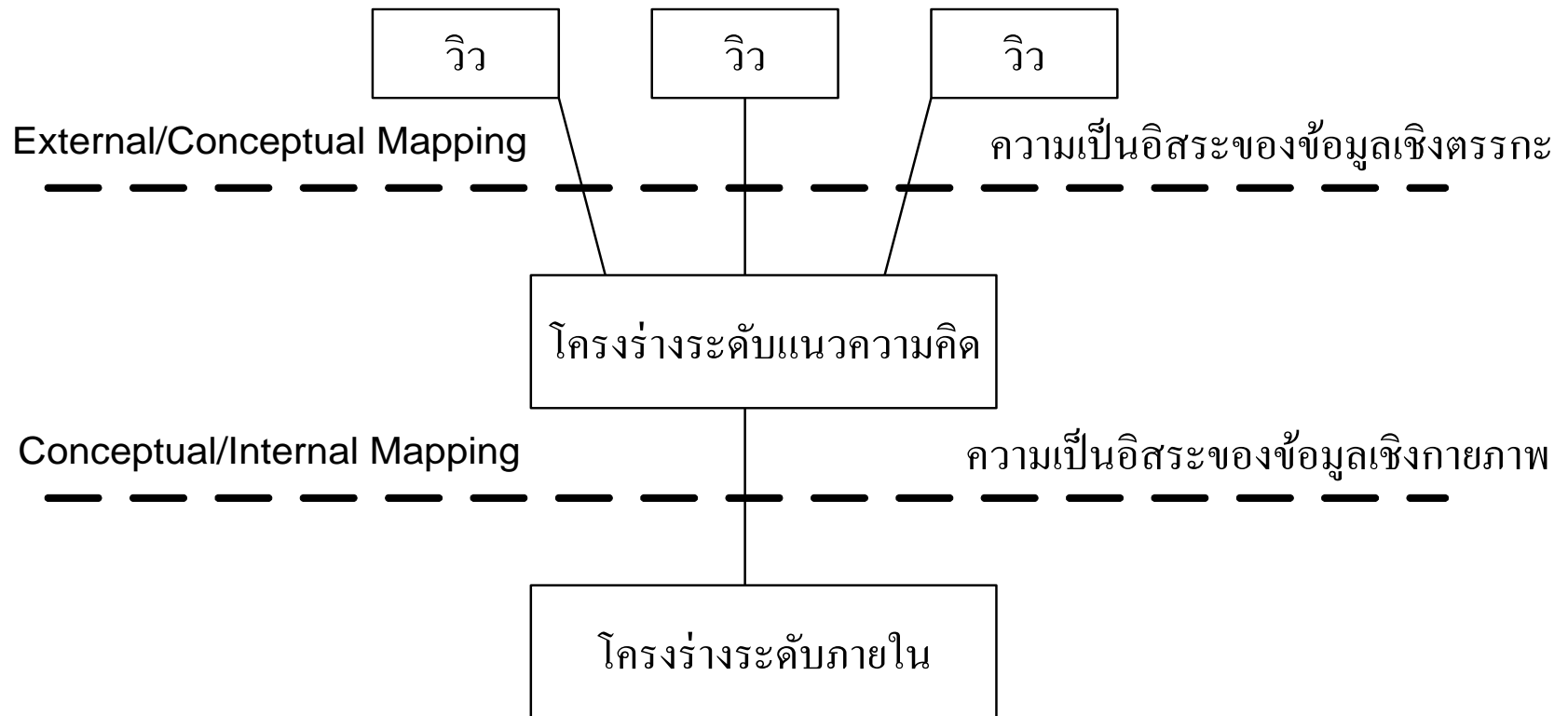
Data Independence คือความสามารถในการเปลี่ยนแปลงสถิติมาในระดับหนึ่งของฐานข้อมูล โดยไม่มีผลกระทบต่อสถิติมาในระดับสูงขึ้นไป

เป็นแนวคิดเพื่อให้โปรแกรมมีความเป็นอิสระจากการเปลี่ยนแปลงโครงสร้างข้อมูล

- ระดับ **Physical** เปลี่ยนแปลงสถิติมาทางกายภาพ จะไม่มีผลกระทบต่อสถิติมาทางตรรกะ

- ระดับ **Logical** เปลี่ยนแปลงสถิติมาทางตรรกะ จะไม่มีผลกระทบต่อโปรแกรมประยุกต์

การแปลงรูปและความเป็นอิสระของข้อมูล





ประโยชน์ของการแบ่งโครงสร้างของข้อมูลออกเป็นระดับชั้น

- ผู้ใช้งานไม่จำเป็นต้องสนใจรายละเอียดของโครงสร้างในการจัดเก็บข้อมูล
- ผู้ใช้แต่ละคนสามารถนำข้อมูลชุดเดียวกัน มาใช้งานที่แตกต่างกัน
- มีความเป็นอิสระของข้อมูลกับตัวโปรแกรม



แบบจำลองของข้อมูล (Data Model)

Data Model

เป็นเครื่องมือสำหรับใช้อธิบาย ถึงโครงสร้างและความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูล จากรูปแบบที่เป็นแนวความคิดที่เข้าใจได้ยาก ให้อยู่ในรูปแบบที่สามารถเข้าใจได้ง่ายขึ้น

แบบจำลองของข้อมูล แบ่งได้เป็น 2 ประเภทคือ

- Conceptual Model
- Implementation Model



แบบจำลองของข้อมูล (Data Model)

1. Conceptual Model

ถูกนำไปใช้ในการออกแบบฐานข้อมูล เพื่ออธิบายถึงข้อมูลและความสัมพันธ์ของข้อมูล

มักประกอบด้วย สัญลักษณ์ที่ใช้แทนตัวข้อมูล คุณสมบัติของข้อมูล และความสัมพันธ์ต่างๆ

- **Entity-Relationship Model (E-R Model)**
- **Object-oriented Model**



แบบจำลองของข้อมูล (Data Model)

2. Implementation Model

ใช้อธิบายถึงโครงสร้างข้อมูลของฐานข้อมูลแต่ละประเภท ที่ถูกคิดค้นขึ้น

- **Hierarchical Model,**
- **Network Model**
- **Relational Model**



ประเภทของความสัมพันธ์ของข้อมูล

1. ความสัมพันธ์แบบ One-to-One (1:1)
 - พนักงาน กับ แผนก (ความสัมพันธ์ : เป็นหัวหน้าแผนก)
2. ความสัมพันธ์แบบ One-to-Many (1:M)
 - พนักงาน กับ แผนก (ความสัมพันธ์ : ทำงานในแผนก)
3. ความสัมพันธ์แบบ Many-to-Many (N:M)
 - พนักงาน กับ ชมรม (ความสัมพันธ์ : อยู่ชมรม)



ประเภทของความสัมพันธ์ของข้อมูล

ฐานข้อมูลโครงการ

1. โครงการ กับ นักศึกษา
2. โครงการ กับ ที่ปรึกษาโครงการ
3. โครงการกับ รายงานความก้าวหน้า

ฐานข้อมูลสมัครงาน

1. บริษัทที่ประกาศรับสมัคร กับประกาศรับสมัครงาน
2. ประกาศรับสมัครงานกับตำแหน่งงาน
3. ผู้สมัคร กับประกาศรับสมัครงาน
4. ผู้สมัครกับความสามารถพิเศษ



ประเภทของความสัมพันธ์ของข้อมูล

ฐานข้อมูลนักศึกษา

1. นักศึกษากับสาขาวิชา
2. ภาควิชากับคณะ
3. คณะกับสถาบัน
4. รายวิชากับ ชั้นเรียนที่เปิดสอน
5. ชั้นเรียนที่เปิดสอนกับนักศึกษา
6. อาจารย์ กับภาควิชา (ทำงานอยู่ในภาควิชา)
7. อาจารย์ กับภาควิชา (เป็นหัวหน้าภาควิชา)
8. อาจารย์กับคณะ (ทำงานอยู่ในคณะ)
9. อาจารย์กับคณะ (เป็นคณบดีของคณะ)
10. อาจารย์ที่ปรึกษากับ นักศึกษา



Implementation Model

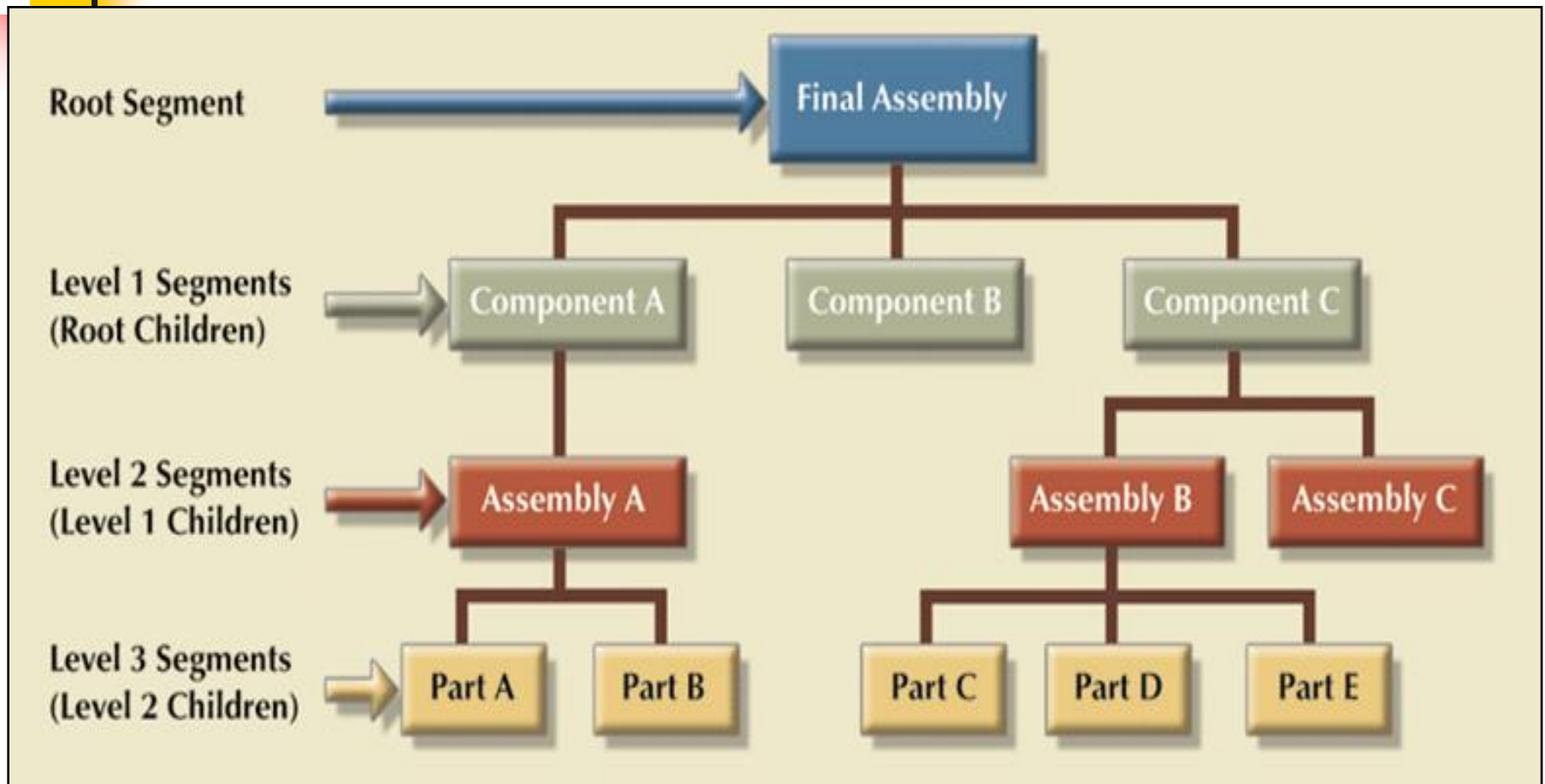
Hierarchical Model

ใช้อธิบายถึงฐานข้อมูลที่มีโครงสร้างของข้อมูลในแบบลำดับชั้น (Hierarchy) คิดค้นโดยบริษัท **North American Rockwell** เพื่อลดการซ้ำซ้อนของข้อมูล (Data Redundancy)

มีการนำข้อมูลแต่ละส่วน (Part) มาจัดเก็บเป็นกลุ่ม ที่เรียกว่า Component แล้วรวมแต่ละกลุ่มเป็นกลุ่มใหญ่ เรียกว่า Final Component

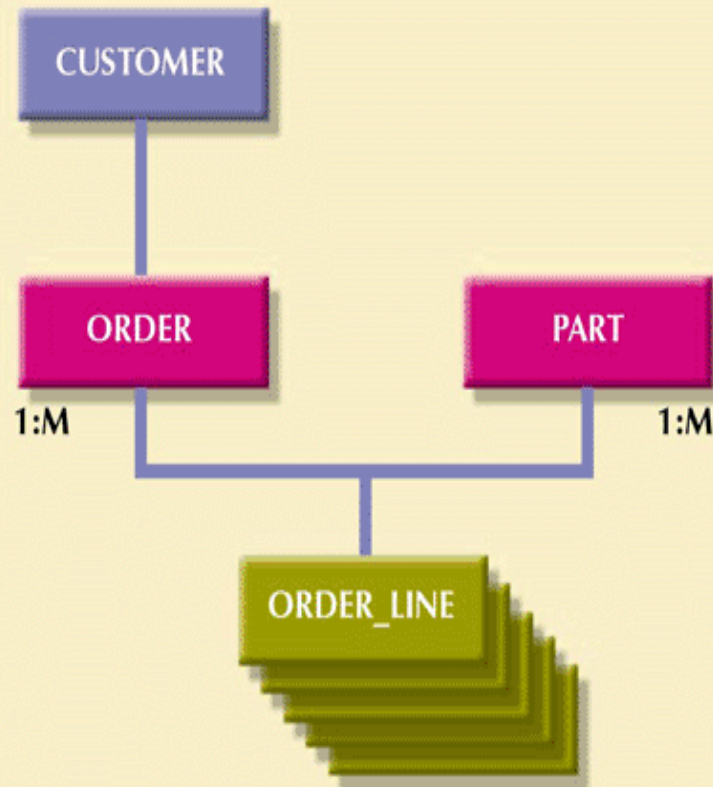
มีโครงสร้างในรูปแบบ Tree ที่เรียกว่า Upside —down Tree ต่อมา
โครงสร้างในลักษณะนี้ ได้ถูกเรียกว่าโครงสร้างแบบ **Hierarchy**

Hierarchical Model



Child with Multiple Parents

FIGURE 2.2 CHILD WITH MULTIPLE PARENTS



แบบจำลองของข้อมูล : Implementation Model



Network Model

- พัฒนามาจาก Hierarchical Model
- กำหนดให้เป็นรูปแบบของโครงสร้างข้อมูลที่เป็นมาตรฐาน
- รองรับข้อมูลที่มีความสัมพันธ์แบบ Many-to-Many

แบบจำลองของข้อมูล : Implementation Model

Network Model

ปี ค.ศ. 1970 กลุ่มผู้พัฒนาภาษา COBOL ที่ชื่อ Conference On Data System Language (CODASYL) ได้กำหนดมาตรฐานเพื่อใช้ในการสร้าง และจัดการกับข้อมูลภายในฐานข้อมูล

1. มาตรฐานในการกำหนดโครงสร้างให้กับฐานข้อมูล (มาตรฐานของกลุ่มคำสั่ง DDL)
2. มาตรฐานการเรียกใช้ข้อมูลจากฐานข้อมูลร่วมกับคำสั่งของภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรม
3. มาตรฐานในการจัดการกับข้อมูลในฐานข้อมูล (มาตรฐานของกลุ่มคำสั่ง DML)

แบบจำลองของข้อมูล : Implementation Model

Network Model

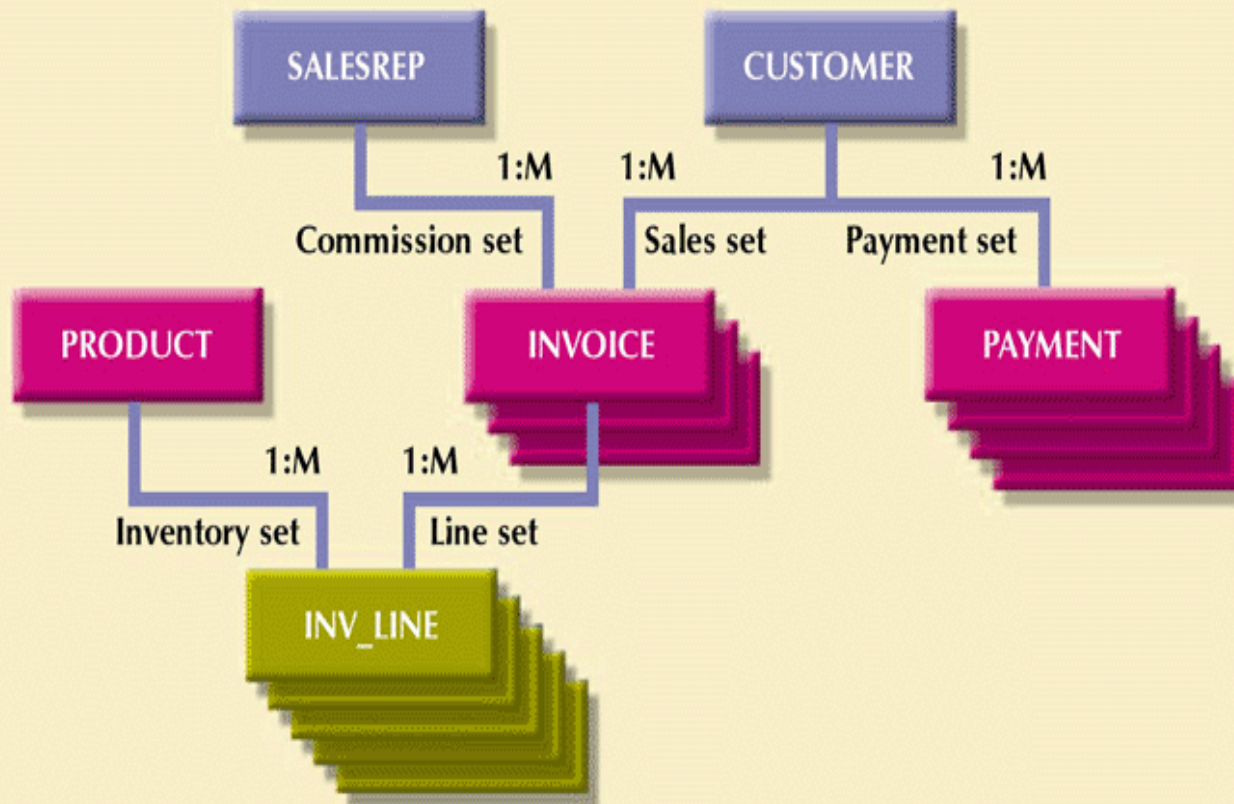
โครงสร้างข้อมูลแบบ Network ถูกกำหนดขึ้นจากความสัมพันธ์ระหว่างข้อมูลตั้งแต่ 2 Record ขึ้นไป ที่เรียกว่า Set

- Owner Record (เทียบเท่า Parent Segment)
- Member Record (เทียบเท่า Child Segment)

Member Record สามารถมีความสัมพันธ์กับ Owner Record ได้มากกว่า 1 Set (หรือ มากกว่า 1 Record)

แบบจำลองของข้อมูล : Implementation Model

FIGURE 2.3 A NETWORK DATA MODEL





บริษัท ผู้ขายสินค้า จำกัด

777/7 หมู่ 1 ต.หนองขาม อ.ศรีราชา จ.ชลบุรี 20280

โทรศัพท์ 038-288822-30 โทรสาร 038-288831

เลขประจำตัวผู้เสียภาษี x xxx xxxxx x

เอกสารออกเป็นชุด

ใบกำกับสินค้า/ใบกำกับภาษี

ต้นฉบับ

ลูกค้า บริษัท ผู้ซื้อสินค้า จำกัด
555/5 ม.8 ถ.ฉลองกรุง แขวงลำปลาทิว
เขตลาดกระบัง กทม.

เลขที่ใบกำกับ IVxxx-xxxx
วันที่ 11/09/2547
เครดิต 30 วัน ครบกำหนด 10/10/2547
เลขที่ใบส่งขาย xxxxxxx
พนักงานขาย คุณชายดี ชายเด่น
เขตการขาย กรุงเทพฯ

ลำดับที่	รหัสสินค้า	รายการสินค้า	จำนวน	ราคา/หน่วย	จำนวนเงิน
1	xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxx	30 ชุด	xxxx	xxxx
2	xxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxx	30 แผ่น	xxxx	xxxx

Remark :

รวมเป็นเงิน 75,000.00
หักส่วนลด 2% 1,500.00
จำนวนเงินหลังหักส่วนลด 73,500.00
จำนวนภาษีมูลค่าเพิ่ม 7.00% 5,145.00

(เจ็ดหมื่นแปดพันหกร้อยสี่สิบบาทถ้วน)

จำนวนเงินรวมทั้งสิ้น 78,645.00

ได้รับสินค้าตามรายการข้างบนนี้ไว้ถูกต้อง
และอยู่ในสภาพเรียบร้อยทุกประการ

ในนามบริษัท ผู้ขายสินค้าจำกัด

ผู้รับสินค้า วันที่/...../.....

ผู้รับมอบอำนาจ

แบบจำลองของข้อมูล : Implementation Model

Relational Model

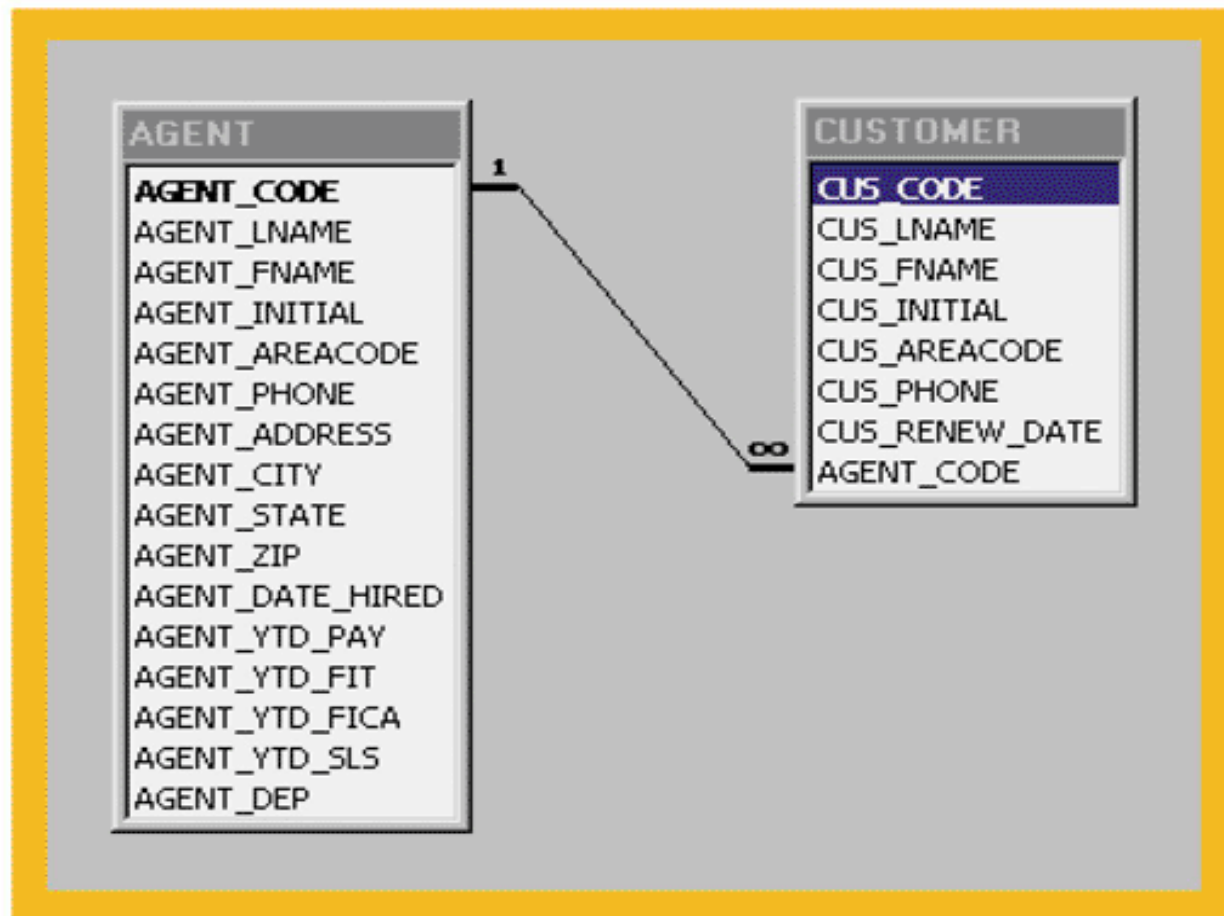
พัฒนามาจากแบบจำลองความสัมพันธ์ระหว่างข้อมูลที่มีชื่อว่า

Relational Model

ข้อมูลที่จัดเก็บในฐานข้อมูล จะถูกแยกจัดเก็บออกเป็นหน่วยย่อยๆ ที่เรียกว่า Relation หรือ Table ที่อยู่ในรูปของตารางที่ประกอบด้วยชุดของแถว และชุดของสคัมภ์

ข้อมูลที่จัดเก็บในแต่ละ Relation จะเป็นข้อมูลที่แยกเป็นเอกเทศ แต่สามารถนำมาสร้างความสัมพันธ์ร่วมกันได้ ซึ่งจะอยู่ในแนวความคิด มากกว่าโครงสร้างทางกายภาพ

A Relational Schema



Linking Relational Tables

Database name: Ch02_InsureCo

Table name: AGENT (first six attributes)

	AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
▶	501	Alby	Alex	B	713	228-1249
	502	Hahn	Leah	F	615	882-1244
	503	Okon	John	T	615	123-5589

Link through AGENT_CODE

Table name: CUSTOMER

	CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
▶	10010	Ramas	Alfred	A	615	844-2573	05-Apr-2004	502
	10011	Dunne	Leona	K	713	894-1238	16-Jun-2004	501
	10012	Smith	Kathy	W	615	894-2285	29-Jan-2005	502
	10013	Olowski	Paul	F	615	894-2180	14-Oct-2004	502
	10014	Orlando	Myron		615	222-1672	28-Dec-2004	501
	10015	O'Brian	Amy	B	713	442-3381	22-Sep-2004	503
	10016	Brown	James	G	615	297-1228	25-Mar-2004	502
	10017	Williams	George		615	290-2556	17-Jul-2004	503
	10018	Farriss	Anne	G	713	382-7185	03-Dec-2004	501
	10019	Smith	Olette	K	615	297-3809	14-Mar-2004	503



The Entity Relationship Model

- **Widely accepted and adapted graphical tool for data modeling**
- **Introduced by Chen in 1976**
- **Graphical representation of entities and their relationships in a database structure**



The Entity Relationship Model-Basic Structure

- **Entity relationship diagram (ERD)**
 - Uses graphic representations to model database components
 - Entity is mapped to a relational table
- **Entity instance is row in table**
- **Entity set is collection of like entities**
- **Connectivity labels types of relationships**
 - Diamond connected to related entities through a relationship line

Relationships: The Basic Chen ERD

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs;
each PAINTING is painted by one PAINTER



A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs;
each SKILL can be learned by many EMPLOYEEs



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE;
each STORE is managed by one EMPLOYEE

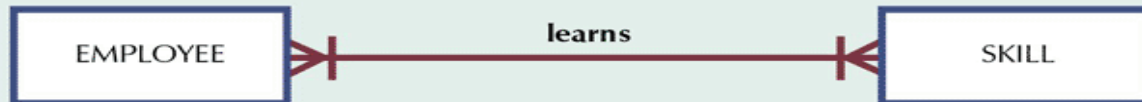


Relationships: The Basic Crow's Foot ERD

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs;
each PAINTING is painted by one PAINTER



A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs;
each SKILL can be learned by many EMPLOYEEs



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE;
each STORE is managed by one EMPLOYEE





The Object Oriented Model

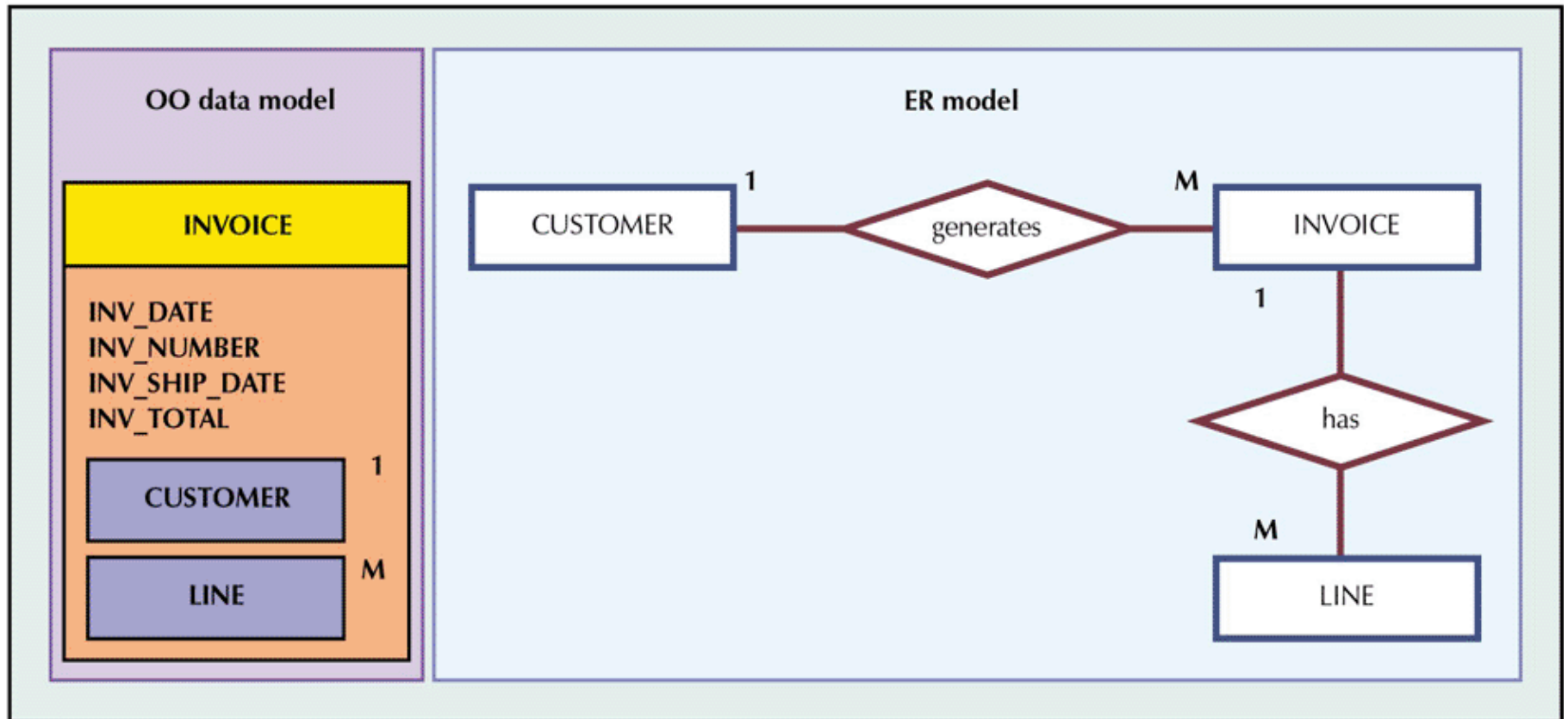
- **Object is described by its factual content**
 - **Like relational model's entity**
- **Includes information about relationships between facts within object and relationships with other objects**
 - **Unlike relational model's entity**
- **Subsequent OODM development allowed an object to also contain operations**
- **Object becomes basic building block for autonomous structures**



Object Oriented Data Model- Basic Structure

- **Object: abstraction of a real-world entity**
- **Attributes describe the properties of an object**
- **Objects that share similar characteristics are grouped in classes**
- **Classes are organized in a class hierarchy**
- **Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of classes above it**

A Comparison of the OO Model and the ER Model





Newer data model

- **Object/Relational and XML**
- **NoSQL (<https://aws.amazon.com/nosql/>)**
 - **Firebase (google)**
 - **Cassandra, Hadoop/Hbase , CouchDB (apache)**
 - **MongoDB (10gen)**
 - **Etc**
- **Cloud-bases database services**
 - **Microsoft SQL azure database**
 - **Google Cloud SQL**

NoSQL ?

ค้นหาข้อมูล

- NoSQL แบ่งเป็นกี่ประเภท อะไรบ้าง
 - ยกตัวอย่างของซอฟต์แวร์ที่เป็น NoSQL ประเภทดังกล่าว
 - ยกตัวอย่างแอปพลิเคชันที่นำ NoSQL ประเภทดังกล่าวไปใช้งาน
- ข้อดี ข้อด้อย ของ NoSQL เทียบกับ Relational Database
- ศึกษาค้นหา เรียบเรียงเนื้อหาด้วยตนเอง

Summary : The Evolution of Data Models

