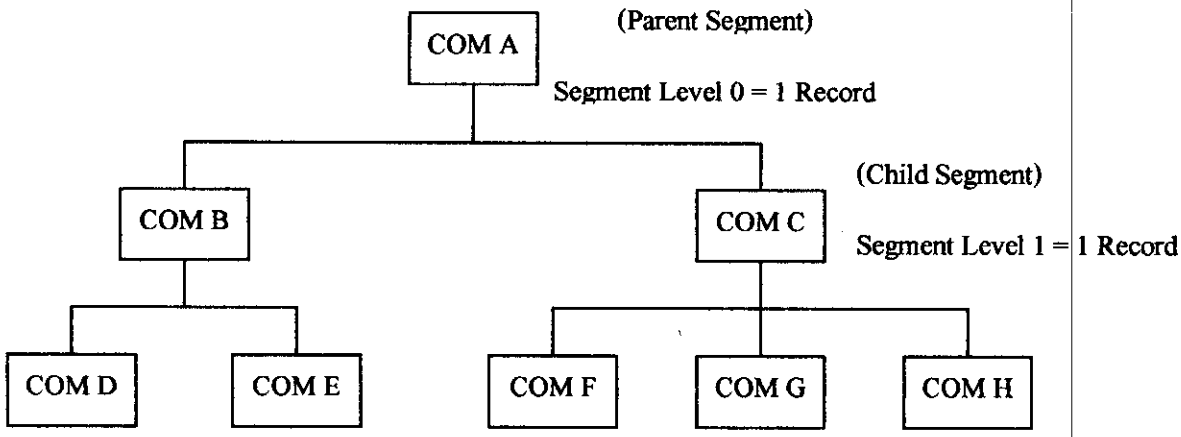
บทที่ 4
รูปแบบฐานข้อมูล
(Database Models)

Database Models แบ่งออกได้ 3 แบบ

4.1 รูปแบบฐานข้อมูลลำดับชั้น (Hierarchical Database Model)

ที่มา ของ Hierarchical Database Model

มีใช้ในครั้งแรกในปี ค.ศ.1969 โดยบริษัท Rockwell สิ่งที่เกิดขึ้น เดิมการเก็บข้อมูลจะเป็นแบบ File System ซึ่งทำให้เกิดความซับซ้อนของข้อมูล ทำให้ความถูกต้องของข้อมูลลดลง จึงมีการนำเอาแนวความคิดของ ฐานข้อมูล มาใช้ แล้วสร้าง Software ชื่อ Generalized Update Access Method (GUAM) ซึ่งเป็นการนำแนวความคิดของ ฐานข้อมูล มาใช้โดยเอาข้อมูลมาจัดเก็บในที่เดียวกัน GUAM ได้กำหนดรูปแบบ Hierarchical Structure ดังนี้



โดยความสัมพันธ์ระหว่าง Segment จากบนลงล่าง จะเป็น 1 : M

โครงสร้างของข้อมูลจะถูกวางในลักษณะของโครงสร้างลำดับชั้น (Hierarchical Structure) (เป็นลักษณะ การวาดกราฟในลักษณะของ Upside – Down)

การจัดเก็บ ใช้เทปซึ่งมีความจุข้อมูลสูงเอามาสร้างเป็น ฐานข้อมูล และใช้ DBMS ที่ชื่อว่า GUAM

Magnetic Tape

COM A	COM B	COM D	COM E	COM C	COM H	COM G	COM F

ถ้าต้องการค้นหา COM E จะต้องเริ่มค้นหาจาก COM A → COM B → COM D

ต่อมาบริษัท Rockwell และบริษัท IBM ได้ร่วมมือกันสร้าง IMS (Information Manage System) เปลี่ยนจาก เทปแม่เหล็กมาเป็น Magnetic Disk ซึ่งมีความจุมากกว่า และสามารถเข้าถึงข้อมูลแบบสุ่ม (random) ได้ซึ่ง (การเข้าถึงข้อมูล (Data Access) จะมีปัญหา คือ จะมีการค้นหาแบบ Sequential ทำให้การค้นหาข้อมูลตอนต้นเร็วกว่าการค้นหาข้อมูลตอนปลาย) IMS สร้างฐานข้อมูล ใหม่โดยได้นำเอาหลักการของ Hierarchical Database Model มาใช้ในการออกแบบและใช้กับเครื่อง Mainframe ซึ่งได้รับความนิยมมาก

จนกระทั่ง IBM ออก DB/2 ซึ่งเป็น Relational Database Model ทำให้ IMS ถูกเลิกใช้ไป

Rockwell + IBM → GUAM → IMS → DB/2

ประโยชน์ 1. แนวความคิดของ Hierarchical Model ได้กลายมาเป็นแนวความคิดในการสร้างระบบฐานข้อมูล ต่อๆ มา

2. จากการใช้แนวความคิดของ Hierarchical Model ทำให้เกิดมุมมองในระบบฐานข้อมูล ที่กว้างข้อง

ข้อดี ของ Hierarchical Database Model

1. ข้อมูลถูกเก็บไว้ในรูปฐานข้อมูล ทำให้เกิดการใช้ข้อมูลร่วมกัน (Data-Sharing) และ DBMS เป็นสิ่งที่ดูแลเรื่องความปลอดภัย (Security) ของข้อมูล
2. DBMS ทำให้เกิดความเป็นอิสระของข้อมูล (Data Independence) ทำให้ความยุ่งยากของการพัฒนาโปรแกรมและการปรับเปลี่ยน (Maintain) โปรแกรมสะดวกขึ้น
3. Hierarchical Database ใช้โครงสร้างข้อมูลแบบความสัมพันธ์ พ่อ- ลูก (Parent/Child Relationship) ซึ่งทำให้เกิดการเชื่อมโยงระหว่าง Parent Segment (ส่วนของพ่อ) กับ Child Segment (ส่วนของลูก) ทำให้เกิดความมั่นคงของข้อมูล (Data Integrity)
4. Hierarchical Database จะทำงานอย่างมีประสิทธิภาพเมื่อข้อมูลในฐานข้อมูลมีความสัมพันธ์แบบ1:M เป็นจำนวนมาก และเมื่อผู้ใช้ต้องการทำงานแบบ Transaction จำนวนมากๆ

ข้อเสีย ของ Hierarchical Database Model

1. Hierarchical Database อาจทำให้ปัญหาเรื่อง Data – Dependence น้อยลง แต่ DBMS ยังคงต้องรู้ถึงข้อมูลเชิงกายภาพ (Physical Data) ด้วยทำให้เกิดปัญหาเมื่อมีการเปลี่ยนแปลงโครงสร้างฐานข้อมูล เช่น การเปลี่ยนตำแหน่งของ Segment (ส่วน) ทำให้เกิดการเปลี่ยนแปลง Application Program ด้วย ทำให้การออกแบบ ฐานข้อมูล เป็นเรื่องยุ่งยาก
2. ข้อมูลโดยปกติแล้วไม่ได้อยู่ในรูปของ 1:M Relationship เสมอไป
3. Hierarchical Database ค่อนข้างยุ่งยากในเรื่องการจัดการ และมีปัญหาเรื่อง Flexibility (ปรับเปลี่ยน) นั่นคือเมื่อมีการเปลี่ยนแปลงมีการเกิดปัญหาซับซ้อนที่ตามมา และอาจเกิดปัญหาที่ทำให้ Segment ที่เป็นลูก ถูกลบทิ้งเมื่อเราเผลอไปลบ Segment ที่เป็น Parent (พ่อ)
4. โปรแกรมประยุกต์ (Application Programming) มักเสียเวลาและยุ่งยาก เพราะโปรแกรมเมอร์ต้องทราบวิธีการเข้าถึงข้อมูล และต้องทราบโครงสร้างของฐานข้อมูล (ต้องทราบ Hierarchical Path (เส้นทางลำดับชั้น))
5. Hierarchical Database ไม่มีแนวคิดแบบมาตรฐาน (Standard Concepts : แนวคิดแบบมาตรฐาน) หรือรูปแบบ การพัฒนาที่เป็นมาตรฐาน
6. Hierarchical Database ต้องถูกเรียกว่าเป็น "A System Created by Programmers for Programmers" ทำให้ผู้ใช้ต้อเป็นผู้ที่มีความรู้เรื่อง Hierarchical Database เป็นอย่างดีจึงจะใช้งานได้ ซึ่งกลายเป็นสิ่งยุ่งยากสำหรับผู้ใช้ที่ไม่มีประสบการณ์ (Naive User)

4.2 รูปแบบฐานข้อมูลเครือข่าย (Network Database Model)

Network Database Model ได้นำเอา Hierarchical Database Model มาปรับปรุง โดยให้ความสัมพันธ์ภายในเป็นแบบ M : N ได้ซึ่งทำให้มีความเป็น Flexibility มากขึ้น

Network Database ถูกพัฒนาขึ้นมาเพื่อ

1. ความต้องการในเรื่องของความสัมพันธ์ของข้อมูลที่ซับซ้อนขึ้น (Complex Data Relationship)
2. การปรับปรุงการทำงานของฐานข้อมูลให้ดีขึ้น
3. การพัฒนาฐานข้อมูลให้มีมาตรฐานมากขึ้น

โครงสร้างพื้นฐานของ Network Model

ที่มาของ Network Model

เพื่อให้เกิดความเป็นมาตรฐานในปี 1971 มีการประชุมของ Conference on Data Systems Language (CODASYL)

CODASYL เป็นกลุ่มผู้สร้าง Cobol Standard ซึ่งเป็นภาษาที่มีลักษณะเป็น English Like Language เช่น A = B + C → Add B To C Giving A เพื่อลดความยุ่งยากในการใช้ภาษา COBOL จึงกำหนดเป็น CODASYL COBOL และกลุ่ม CODASYL ได้นำแนวความคิดของ ฐานข้อมูล และตั้ง Database Task Group (DBTG) เพื่อทำหน้าที่กำหนดมาตรฐานของการพัฒนา ฐานข้อมูล และการจัดการข้อมูล

DBTG คือกลุ่มของบุคลากรที่ทำหน้าที่กำหนดมาตรฐานของการพัฒนา ฐานข้อมูล และการจัดการข้อมูล

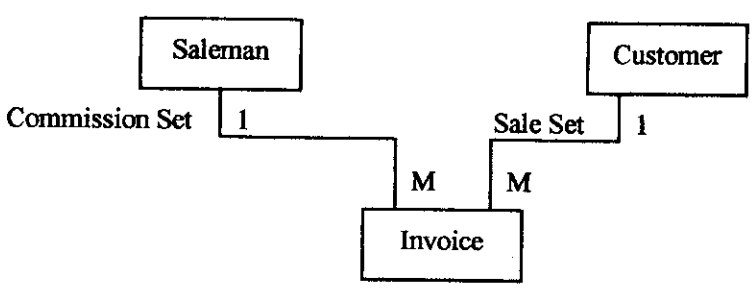
ผลสรุปของ DBTG ได้เสนอข้อกำหนดเกี่ยวกับองค์ประกอบที่สำคัญของ ฐานข้อมูล ใช้ 3 อย่างคือ

1. Network Schema Conceptual Orgranization รูปแบบความคิดของรายละเอียดทั้งหมดของ ฐานข้อมูล ที่เป็นมุมมองของ Database Administrator :DBA (บุคคลซึ่งทำหน้าที่ดูแลระบบ ฐานข้อมูล) ซึ่งได้ Schema (เค้าร่าง) นี้จะรวมรายละเอียดเกี่ยวกับชื่อของ ฐานข้อมูล ชนิดของ Record องค์ประกอบของ Record
2. Subschema (เค้าร่างย่อย) คือกำหนดส่วนย่อยของ ฐานข้อมูล ที่เป็นมุมมองของ Application Program ที่ทำให้เกิดการสร้าง Information ตามที่ต้องการ จาก ข้อมูลที่มีอยู่ใน ฐานข้อมูล นั่นคือ Application Program ที่เรียกใช้ Subschema ในการเข้าถึงข้อมูล
3. Data Management Language :DML (คำสั่งที่ใช้ในการกระทำกับข้อมูล) เพื่อใช้ในการกำหนดโครงสร้างข้อมูล และเพื่อจัดการกับ ข้อมูล

และเพื่อให้เกิดมาตรฐานขององค์ประกอบทั้ง 3 DBTG ได้กำหนดองค์ประกอบของ Data Management Language ใช้ 3 อย่างคือ

1. Data Definition Language :DDL (คำสั่งการนิยามข้อมูล) เพื่อให้ DBA ใช้กำหนด Schema
2. Subschema DDL เพื่อให้ Application Program ใช้กำหนด Subschema ที่ต้องการ
3. DML เพื่อใช้จัดการเก็บข้อมูลใน ฐานข้อมูล

จึงเกิดความเป็นมาตรฐานของการใช้ Network Database ขึ้น



4.3. ตัวแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

การที่เราต้องทราบถึงโครงสร้างฐานข้อมูลในระดับ Physical Details (รายละเอียดเชิงกายภาพ) ทำให้การออกแบบฐานข้อมูลที่ดีนั้นทำได้ยาก เพราะทำให้ผู้ใช้และนักออกแบบเพียงไม่กี่คนเท่านั้นที่สามารถใช้ ฐานข้อมูลได้อย่างมีประสิทธิภาพ (DBA และเจ้าหน้าที่ ที่ทำงานในการเก็บข้อมูลโดยตรง) และ เมื่อเข้าสู่ยุคของ Information Technology (เทคโนโลยีสารสนเทศ) ทำให้มีความต้องการระบบฐานข้อมูลที่มีประสิทธิภาพขึ้น ซึ่งทำให้การออกแบบฐานข้อมูล การจัดการและการใช้ฐานข้อมูลยังเป็นสิ่งที่ยุ่งยากขึ้นด้วย

ความต้องการ Ad Hoc Query (Ad Hoc Query คือคำสั่งที่จะใช้สั่งเรียกข้อมูลพิเศษที่ต้องการ) ในการออกแบบรายงานง่ายๆ ทำให้โปรแกรมเมอร์ต้องเขียน Code ที่ยุ่งยากสำหรับทำรายงานง่ายๆ เหล่านั้น และ Database Model ที่มีใช้อยู่นั้นถึงแม้จะได้รับ Data Independence ในระดับหนึ่ง แต่ก็ยังคงมี Structural Dependency อยู่นั่นเองที่มีผลโดยตรงต่อการทำงานของ Application Program

Relational Database Model พัฒนาขึ้นมาโดย E.F. CODD ในราวปี 1970 ซึ่งเป็นการเสนอการเปลี่ยนแปลงที่สำคัญให้กับ User และ Designer โดยการใช้รูปแบบความสัมพันธ์ที่ต่อเนื่อง ทำให้เกิด Automatic Transmission Database (การส่งข้อมูลโดยอัตโนมัติ) แทนการใช้ Standard Transmission Database (มาตรฐานในการส่งข้อมูล) ซึ่งถือว่าเป็นการปฏิบัติทาง ฐานข้อมูล อย่างแท้จริง

อย่างไรก็ตามผลงานของ CODD ได้ถูกมองว่าดีจริง แต่นำมาใช้งานจริงไม่ได้ (เป็นแค่แนวคิด) เพราะแนวความคิดของ Relational Database Model นั่นจะต้องใช้ค่าใช้จ่ายสูงในการนำมาใช้งาน เครื่องคอมพิวเตอร์ในขณะนั้นยังไม่มีประสิทธิภาพสูงพอที่จะนำมาใช้กับ Relational Database Model แต่ในปัจจุบันประสิทธิภาพของเครื่องคอมพิวเตอร์สูงขึ้นมาก ประกอบกับมี O.S. ที่ทำงานอย่างมีประสิทธิภาพเช่นกัน ทำให้มีการนำ Relational Database Model มาใช้งานได้จริง เช่น Oracle, Ingriss, Informix เป็นต้น

โครงสร้างพื้นฐานของฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

Relational Database Model ถูกสร้างขึ้นโดยอาศัย Relational Database Management Systems (RDBMS) ซึ่ง RDBMS นี้มีการทำงานพื้นฐานเช่นเดียวกับ Hierarchical และ Network DBMS แต่มีการเพิ่มการทำงานอื่นๆ ที่ทำให้การใช้และการสร้าง Relational Database Model

DBMS แต่มีการเพิ่มการทำงานอื่นๆ ที่ทำให้การใช้และการสร้าง Relational Database Model

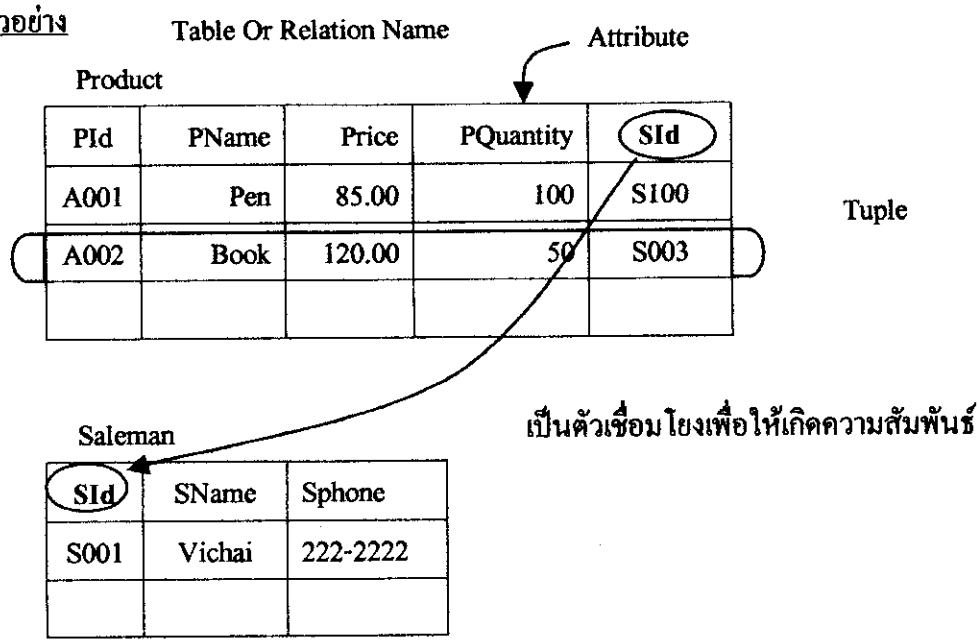
ทำได้ง่ายขึ้น โดย RDBMS จัดการกับรายละเอียดทั้งหมดที่เป็น Physical Level (ระดับกายภาพ)

ผู้ใช้จะมอง Relational Model ในรูปของ กลุ่มของ Table ซึ่ง Table แต่ละตัวจะเป็นตัวเก็บข้อมูล (Table คือที่เก็บข้อมูล ซึ่ง Table นี้บางครั้งเรียกว่า Relation) โดย Table จะถูกนำมาเชื่อมโยงกันด้วยคุณสมบัติบางอย่างที่มีอยู่ในข้อมูลที่จัดเก็บไว้ใน Table นั้น

Table ใน Relational Database Model นั้นเปรียบได้กับ File แต่สิ่งที่ต่างกันก็คือ Table เป็นโครงสร้างแบบ Logical (ผู้ใช้เห็นข้อมูลในรูปตารางซึ่งการจัดเก็บจริงๆ ไม่ได้เป็นเช่นนั้น) ทำให้เป็น Data และ Structural Independence อย่างแท้จริง

Tuple คือ ที่อยู่ของสมาชิกใน Set (Row)

<u>ตัวอย่าง</u>



ข้อดี

1. Relational Database Model เป็น Single Data Repository คือการนำเอาข้อมูลทั้งหมดมาไว้ที่เดียวกัน ซึ่งเหมือนกับ Hierarchical และ Network Database Model แต่ต่างกันตรงที่ Relational Database Structure นั้นไม่มีความจำเป็นสำหรับ User และ Designer จะต้องเรียนรู้เกี่ยวกับ Physical View ของ Data ทำให้เป็น Structural Independent ซึ่ง Hierarchical Database Network Database ไม่มี

2. Relational Database Model ง่ายต่อการออกแบบ และจัดการ เพราะผู้ใช้หรือนักออกแบบจะเห็นแต่ Logical View ของฐานข้อมูลเท่านั้น
3. ใช้ Structure Query Language :SQL (ภาษาสอบถามเชิงโครงสร้าง) ซึ่งเป็น Query ที่มีประสิทธิภาพมากในการใช้งาน (ซึ่งจัดเป็น 4thGL – Ailowt User to Specify What Must be Done Without Specifying How it must be done) โดย RDBMS จะใช้ SQL ในการแปลง User Query ให้อยู่ในรูปรหัสที่จะใช้ในการเข้าไปนำ ข้อมูลที่ต้องการออกมาจากฐานข้อมูล และ SQL ยังนำไปใช้ในการสร้างโครงสร้าง Table, ดูแล ฐานข้อมูล จัดการเรื่องการเข้าถึงข้อมูลใน Table และแปลความต้องการของผู้ใช้ให้อยู่ในรูปที่เครื่องคอมพิวเตอร์เข้าใจ

ข้อเสีย

1. ค่าใช้จ่ายสูงด้าน H.W. และ O.S. ร่วมทั้งทำงานช้า (ต้องใช้อุปกรณ์คุณภาพสูงถึงจะทำงานเร็วขึ้น)
2. การที่ใช้งานง่าย ทำให้ผู้ใช้บางกลุ่ม (Naive User) ใช้งานอย่างไม่มีรูปแบบ คือไม่มีการออกแบบที่เหมาะสม ทำให้ประสิทธิภาพการทำงานด้อยลง หรือ มีปัญหาในการใช้งาน

(การที่จะแยก Physical View กับ Logical View ออกจากกันได้ จะต้องมีทั้ง

1. Data Independent
2. Structural Independent)

ระบบสารสนเทศ (Information Systems)

ระบบฐานข้อมูลคือ แหล่งข้อเท็จจริง ซึ่งได้รับการออกแบบและสร้างขึ้นมาอย่างดี ข้อเท็จจริงที่เก็บนี้คือส่วนหนึ่งของระบบสารสนเทศ

ระบบสารสนเทศเสนอถึงการรวบรวม การจัดเก็บ และการนำมาใช้ของข้อมูล รวมถึงกระบวนการเปลี่ยนข้อมูลให้เป็นสารสนเทศ ทั้งยังจัดการข้อมูลและสารสนเทศให้มีประสิทธิภาพในการนำไปใช้ ดังนั้นระบบสารสนเทศที่สมบูรณ์จึงประกอบด้วย ฮาร์ดแวร์, ซอฟท์แวร์, บุคลากร, ระบบฐานข้อมูล, โปรแกรมประยุกต์ และขั้นตอนการทำงาน

การวิเคราะห์ระบบ (Systems Analysis) คือกระบวนการที่ซึ่งกำหนดความต้องการและขอบเขต (Extent) ของระบบสารสนเทศ กระบวนการของการสร้างระบบสารสนเทศขึ้นมาใช้งานเรียกว่า การพัฒนาระบบ (Systems Development)