

2110422

การออกแบบการจัดการฐานข้อมูล



ฐานข้อมูลเบื้องต้นและแบบจำลองเชิงแนวคิด

Introduction and Conceptual Modeling

ชนิดของฐานข้อมูล และ แอปพลิเคชันของฐานข้อมูล

- ฐานข้อมูลและระบบฐานข้อมูลในชีวิตประจำวัน
 - ตัวอย่างเช่น การจองโรงแรมและตั๋วเครื่องบิน การยืมหนังสือจากห้องสมุด ระบบการสั่งของทางไปรษณีย์ เป็นต้น
- ฐานข้อมูลตัวเลขและข้อความ
- ฐานข้อมูลมัลติมีเดีย
 - รูปภาพ คลิปวิดีโอทัศน์ หรือ ข้อความเสียง
- ฐานข้อมูลทางภูมิศาสตร์
 - แผนที่ ข้อมูลภูมิอากาศ หรือ ภาพถ่ายดาวเทียม
- คลังข้อมูล
 - สกัดข้อมูลจากข้อมูลจำนวนมากและมีเครื่องมือที่ช่วยในการวิเคราะห์ข้อมูลเพื่อใช้ในการตัดสินใจ
- ฐานข้อมูลแบบทันกาลเชิงรุก
 - ใช้ในการควบคุมกระบวนการในงานอุตสาหกรรมและการผลิต

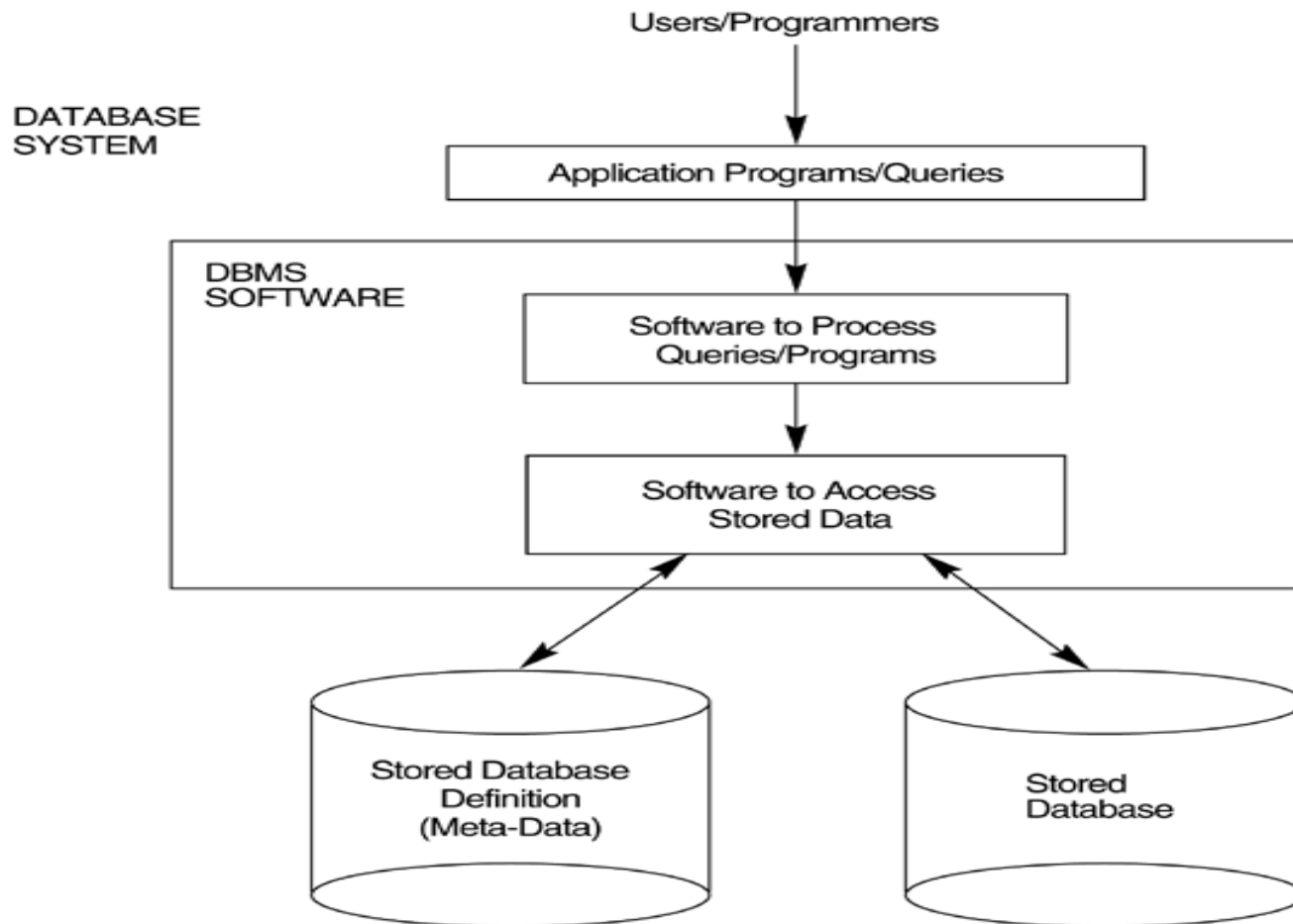
คำจำกัดความเบื้องต้น

- **ฐานข้อมูล:** กลุ่มของข้อมูลที่เกี่ยวข้องกัน
- **ข้อมูล:** ข้อเท็จจริงที่สามารถบันทึกได้และมีความหมายโดยปริยาย
- **คุณสมบัติของฐานข้อมูล:**
 - เป็นแบบจำลองของปัญหาจริง ซึ่งเรียกว่า **มินิเวิลด์**
 - ข้อมูลที่เก็บไว้ในฐานข้อมูลเดียวกันควรมีความเกี่ยวข้องกันและมีความหมายในตัวเอง
 - ได้ถูกออกแบบ สร้าง และบรรจุข้อมูลลงในฐานข้อมูลเพื่อวัตถุประสงค์อันใดอันหนึ่งที่แน่ชัด
- **มินิเวิลด์:** แบบจำลองปัญหาจริง หรือส่วนหนึ่งของปัญหาจริงที่ข้อมูลในฐานข้อมูลนั้นเกี่ยวข้อง เช่น ฐานข้อมูลการลงทะเบียนของนิสิตในมหาวิทยาลัย

คำจำกัดความเบื้องต้น (ต่อ)

- **ระบบการจัดการฐานข้อมูล (DBMS):** ซอฟต์แวร์ที่ใช้ในการสร้างและบำรุงรักษาข้อมูลและฐานข้อมูลในคอมพิวเตอร์
- **ระบบฐานข้อมูล:** ซอฟต์แวร์ระบบการจัดการฐานข้อมูลพร้อมกับข้อมูลบางครั้งอาจรวมถึงแอปพลิเคชันด้วย

รูปที่ 1.1: ระบบฐานข้อมูลเบื้องต้น



ความสามารถของระบบการจัดการฐานข้อมูล

- กำหนดคุณลักษณะของฐานข้อมูล : ตั้งแต่ชนิดของข้อมูล, โครงสร้างของข้อมูล และเงื่อนไขบังคับต่าง ๆ
- สร้างฐานข้อมูลและบรรจุข้อมูล ลงฐานข้อมูลลงในหน่วยเก็บ
- จัดดำเนินการฐานข้อมูล : การสืบค้น, การจัดทำรายงาน การเพิ่ม การลบ การแก้ไขข้อมูล
- การประมวลผลพร้อมกันและใช้ข้อมูลร่วมกัน สามารถมีผู้ใช้งานหลายคนหรือโปรแกรมหลายโปรแกรมทำงานพร้อมกันได้ – โดยที่ข้อมูลยังคงความถูกต้อง

ความสามารถของระบบการจัดการฐานข้อมูล

- คุณสมบัติอื่น ๆ:
 - การป้องกันและดูแลความปลอดภัย
 - การตรวจสอบการเข้าถึงที่ไม่ได้รับการอนุญาต
 - การจัดการด้านความมั่นคงของข้อมูลเมื่อซอฟต์แวร์หรือฮาร์ดแวร์มีปัญหา
 - การปรับปรุงและบำรุงรักษาฐานข้อมูล
 - DBMS ทำให้ระบบสามารถปรับเปลี่ยนไปตามความต้องการที่เปลี่ยนแปลงได้

ตัวอย่างของฐานข้อมูล (โดยมีแบบจำลองเชิงแนวคิดประกอบ)

- ตัวอย่างของมินิเวิลด์ : UNIVERSITY
- **เอนทิตี** ของมินิเวิลด์:
 - STUDENTs
 - COURSEs
 - SECTIONs (of COURSEs)
 - (academic) DEPARTMENTs
 - INSTRUCTORs

Note: จากมินิเวิลด์นี้ เราสามารถเขียน ENTITY-RELATIONSHIP data model ได้

ตัวอย่างของฐานข้อมูล

(โดยมีแบบจำลองเชิงแนวคิดประกอบ)

- **ความสัมพันธ์** ของมินิเวิร์ลด์:
 - SECTIONs เป็นส่วนย่อยของ COURSEs หนึ่ง ๆ
 - STUDENTs เลือก SECTIONs
 - COURSEs ต้องมี COURSEs เบื้องต้น
 - INSTRUCTORs สอน SECTIONs
 - COURSEs กำหนดโดย DEPARTMENTs
 - STUDENTs มีสังกัด DEPARTMENTs

Note: จากมินิเวิร์ลด์นี้ เราสามารถเขียน ENTITY-RELATIONSHIP data model.

รูปที่ 1.2ก:

ฐานข้อมูลสำหรับเก็บข้อมูลนักศึกษาและรายวิชา

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

รูปที่ 1.2ข:

ฐานข้อมูลสำหรับเก็บข้อมูลตอนเรียน

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

รูปที่ 1.2ค:

ฐานข้อมูลสำหรับเก็บข้อมูลรายงานผลการศึกษาและรายวิชา
พื้นฐาน

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

ลักษณะสำคัญของการใช้กลยุทธฐานข้อมูล

- ธรรมชาติในการบรรยายตนเองของระบบฐานข้อมูล:
 - Catalog ของ DBMS จะเก็บ definition หรือ description ของฐานข้อมูล
 - Definition ประกอบด้วย โครงสร้างไฟล์ ชนิด รูปแบบของข้อมูล และข้อบังคับต่างๆ
 - Description หรือเรียกว่า meta-data ดังแสดงในรูปที่ 1.1 เป็นส่วนที่ทำให้ซอฟต์แวร์ DBMS สามารถทำงานกับฐานข้อมูลอื่นๆได้

ลักษณะสำคัญของการใช้กลยุทธฐานข้อมูล (ต่อ)

- การไม่ขึ้นต่อกันระหว่างโปรแกรมและข้อมูล:
 - สามารถเปลี่ยนแปลงโครงสร้างของหน่วยเก็บข้อมูลและการจัดการต่างๆ โดยไม่จำเป็นต้องเปลี่ยนโปรแกรมการเข้าถึง DBMS
 - โครงสร้างของไฟล์ข้อมูลนั้นถูกแยกออกจากโปรแกรมการเข้าถึง (Access program) ซึ่งไม่เหมือนกระบวนการจัดการไฟล์ (File processing) แบบดั้งเดิม
 - ในกระบวนการจัดการไฟล์แบบดั้งเดิม โครงสร้างของไฟล์ข้อมูลจะถูกฝังติดกับโปรแกรมแอปพลิเคชัน

รูปที่ 1.3:

รูปแบบของหน่วยเก็บข้อมูลภายในสำหรับเก็บระเบียนนักศึกษา

<u>Data Item Name</u>	<u>Starting Position in Record</u>	<u>Length in Characters (bytes)</u>
Name	1	30
StudentNumber	31	4
Class	35	4
Major	39	4

ลักษณะสำคัญของการใช้กลยุทธฐานข้อมูล(ต่อ)

- Data Abstraction: หรือ Program-Operation

Independence

- ในระบบ Object-Oriented และ Object-Relational นั้น ผู้ใช้สามารถระบุการจัดการให้กับข้อมูลในรูปแบบของนิยามฐานข้อมูล (Database definition)
- การจัดการ (ฟังก์ชันหรือเมทอด) ประกอบด้วย 2 ส่วน
 - ส่วนติดต่อกับผู้ใช้ (หรือ Signature): ชื่อคำสั่งและพารามิเตอร์ต่างๆ
 - การปฏิบัติการ (หรือ วิธีการ) นั้นถูกระบุแยกออกมา ดังนั้นจึงสามารถทำการเปลี่ยนแปลงได้โดยไม่กระทบส่วนติดต่อผู้ใช้
- **แบบจำลองข้อมูล** ใช้ในการซ่อนรายละเอียดของหน่วยเก็บ และนำเสนอในรูปแบบของ *มุมมองระดับแนวคิด*
- DBMS จะทำการดึงรายละเอียดของหน่วยเก็บไฟล์จาก Catalog

ลักษณะสำคัญของการใช้กลยุทธ์ฐานข้อมูล(ต่อ)

- สนับสนุนการพิจารณาข้อมูลจากหลายมุมมอง : ผู้ใช้แต่ละคนสามารถพิจารณาฐานข้อมูลในมุมมอง (View) ที่แตกต่างกัน ซึ่งอธิบายเฉพาะส่วนของข้อมูล que ผู้ใช้สนใจเท่านั้น
 - มุมมอง ประกอบด้วย ข้อมูลที่อนุมานได้จากไฟล์ฐานข้อมูล แต่ไม่ได้ถูกเก็บไว้ในฐานข้อมูลอย่างชัดเจน

รูปที่ 1.4

- แสดงสองมุมมองที่ได้จากฐานข้อมูลในรูปที่ 1.2

- (a) มุมมอง STUDENT TRANSCRIPT
- (b) มุมมอง COURSE PREREQUISITES

(a)

TRANSCRIPT	StudentName	Student Transcript				
		CourseNumber	Grade	Semester	Year	SectionId
	Smith	CS1310	C	Fall	99	119
		MATH2410	B	Fall	99	112
	Brown	MATH2410	A	Fall	98	85
		CS1310	A	Fall	98	92
		CS3320	B	Spring	99	102
		CS3380	A	Fall	99	135

(b)

PREREQUISITES	CourseName	CourseNumber	Prerequisites
	Database	CS3380	CS3320
			MATH2410
	Data Structures	CS3320	CS1310

ลักษณะสำคัญของการใช้กลยุทธฐานข้อมูล (ต่อ)

- การใช้ข้อมูลร่วมกัน และ การดำเนินการโดยกลุ่มผู้ใช้:
 - ผู้ใช้หลายคนมีสิทธิ์ในการร่วมกันแก้ไขและปรับปรุงฐานข้อมูลได้
 - ต้องมีการรับรองจาก DBMS ว่าแต่ละการดำเนินการ (Transaction) จะต้องถูกปฏิบัติการ (Execute) อย่างถูกต้องจนครบสมบูรณ์หรือไม่เช่นนั้นก็ต้องไม่มีการดำเนินการใดๆเลย
 - OLTP (แอปพลิเคชันประเภทออนไลน์) เป็นส่วนสำคัญมากสำหรับแอปพลิเคชันฐานข้อมูล
 - (การดำเนินการ หมายถึงโปรแกรมหรือกระบวนการปฏิบัติการ ที่ทำการเข้าถึงฐานข้อมูลตั้งแต่หนึ่งครั้งขึ้นไป เช่น การอ่าน การแก้ไข เป็นต้น)

ผู้ใช้ฐานข้อมูล

- สามารถจัดแบ่งผู้ใช้ออกเป็น 2 ประเภทหลักๆ
 - ผู้มีบทบาทเบื้องหน้า (Actors on the Scene): ผู้ที่ใช้และควบคุมเนื้อหาของฐานข้อมูล
 - ผู้มีบทบาทเบื้องหลัง (Actor behind the Scene): ผู้ที่พัฒนาฐานข้อมูลหรือออกแบบซอฟต์แวร์ DBMS

ผู้มีบทบาทเบื้องหน้า

- **ผู้บริหารฐานข้อมูล (Database administrators : DBA):**
 - มีอำนาจในการให้สิทธิ์การเข้าถึงฐานข้อมูล
 - ประสานงานและดูแลการใช้ฐานข้อมูล
 - จัดหาทรัพยากรทางด้านซอฟต์แวร์และฮาร์ดแวร์
 - ควบคุมการใช้ฐานข้อมูลและตรวจสอบประสิทธิภาพของการจัดการ
 - แก้ไขปัญหาต่างๆ เช่น ช่องว่างของระบบความปลอดภัย และปัญหาเวลาการตอบสนองต่ำ
- **ผู้ออกแบบฐานข้อมูล (Database Designers):**
 - ระบุเนื้อหา โครงสร้าง ข้อบังคับ (Constraint) รวมถึงฟังก์ชันหรือการดำเนินการ (Transaction) ต่างๆ ที่เกี่ยวข้องกับฐานข้อมูล
 - ติดต่อและทำความเข้าใจความต้องการของผู้ใช้ปลายทาง
- **ผู้ใช้ปลายทาง (End-user):**
 - ใช้ข้อมูลในการสืบค้น (Query) และจัดทำรายงาน
 - บางส่วนอาจทำหน้าที่ในการปรับปรุงเนื้อหาของฐานข้อมูลอีกด้วย

ผู้มีบทบาทเบื้องหลัง

- **ผู้ออกแบบและพัฒนาระบบ DBMS**
 - ออกแบบและพัฒนาส່วณต่างๆ รวมทั้งส่วนติดต่อกับผู้ใช้ ในรูปแบบของชุดซอฟต์แวร์
- **ผู้พัฒนาเครื่องมือ (Tool Developers)**
 - พัฒนาเครื่องมือ หรือทูล (Tool) ซึ่งเป็นชุดซอฟต์แวร์ที่ช่วยในการออกแบบระบบฐานข้อมูลและช่วยในการพัฒนาผลงาน
- **ผู้ปฏิบัติงานและดูแลรักษา**
 - สั้่งดำเนินการ (Run) รวมถึงดูแลฮาร์ดแวร์และซอฟต์แวร์ของระบบฐานข้อมูล

ประเภทของผู้ใช้ปลายทาง

- **ผู้ใช้ชั่วคราว (Casual):** ทำการเข้าถึงฐานข้อมูลเป็นครั้งคราว
- **Naïve or Parametric:** เป็นกลุ่มผู้ใช้ปลายทางส่วนใหญ่ ซึ่งใช้ฟังก์ชันต่างๆ ที่ถูกกำหนดมาเรียบร้อยแล้วในรูปแบบของ “การดำเนินการแบบสำเร็จรูป (canned transaction)” กับระบบฐานข้อมูล เช่น พนักงานรับและจ่ายเงินในธนาคาร พนักงานสำรองที่ เป็นต้น

ประเภทของผู้ใช้ปลายทาง

- **Sophisticated** : เช่น นักวิเคราะห์ธุรกิจ นักวิทยาศาสตร์ วิศวกร รวมถึงผู้ใช้อื่นๆที่มีความคุ้นเคยกับสมรรถภาพของระบบ
- **Stand-alone** : ส่วนมากมักจะเป็นผู้ที่ดูแลฐานข้อมูลส่วนบุคคลโดยใช้ชุดแอปพลิเคชันพร้อมใช้ เช่น ผู้ใช้โปรแกรมคำนวณภาษี ซึ่งสร้างฐานข้อมูลภายในขึ้นมาเอง เป็นต้น

ข้อดีในการใช้กลยุทธ์ฐานข้อมูล

- สามารถควบคุมความซ้ำซ้อน (Redundancy) ในการจัดเก็บข้อมูล รวมถึงความยากลำบากในการพัฒนาและดูแลรักษา
 - ความซ้ำซ้อนของข้อมูลนั้นเป็นสาเหตุของปัญหามากมาย เช่น
 - จำเป็นต้องทำการปรับปรุงหรือแก้ไขข้อมูลเดียวกันหลายครั้ง
 - สิ้นเปลืองพื้นที่ในการเก็บข้อมูล
 - ไฟล์ที่เสนอข้อมูลเดียวกันอาจมีการจัดเก็บข้อมูลที่ไม่ตรงกันได้
- สามารถใช้ข้อมูลร่วมกันได้
- ป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

รูปที่ 1.5.

การเก็บข้อมูลที่ซ้ำซ้อนของ StudentName และ CourseNumber ใน

(a) กรณีข้อมูลตรงกัน (b) กรณีที่ข้อมูลไม่ตรงกัน

(a)

GRADE_REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Smith	112	MATH2410	B
	17	Smith	119	CS1310	C
	8	Brown	85	MATH2410	A
	8	Brown	92	CS1310	A
	8	Brown	102	CS3320	B
	8	Brown	135	CS3380	A

(b)

GRADE_REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Brown	112	MATH2410	B

ข้อดีในการใช้กลยุทธฐานข้อมูล(ต่อ)

- สามารถเตรียมหน่วยเก็บข้อมูลที่มั่นคง (Persistent storage) สำหรับโปรแกรม Objects (ใน Object-oriented DBMS's – บทที่ 20-22)
- จัดเตรียมโครงสร้างหน่วยเก็บข้อมูลเพื่อให้สามารถดำเนินการสืบค้น (Query) ได้อย่างมีประสิทธิภาพ
 - DBMS จะมีโครงสร้างข้อมูลพิเศษ เพื่อให้การค้นข้อมูลในดิสก์รวดเร็วยิ่งขึ้น ซึ่งทำได้โดยใช้ดัชนี (Indexes)
- สามารถทำการสำรอง (Backup) และฟื้นฟูสภาพข้อมูล (Recovery)
- มีส่วนการติดต่อผู้ใช้หลากหลายรูปแบบสำหรับผู้ใช้ในกลุ่มต่างๆกัน
 - ประกอบด้วย ภาษาสืบค้น(Query language) ภาษาโปรแกรม (Programming language) แบบฟอร์มและโค้ดคำสั่ง (Form and Command code) และ Menu-driven interface

ข้อดีในการใช้กลยุทธ์ฐานข้อมูล(ต่อ)

- สามารถนำเสนอความสัมพันธ์ที่ซับซ้อนของข้อมูลได้
- มีการกำหนดข้อบังคับในการควบคุมความคงสภาพ (Integrity constraints)
- สนับสนุนการอนุมาน (Inference) ข้อมูลใหม่ๆจากข้อเท็จจริงที่เก็บไว้ในฐานข้อมูล รวมถึงการดำเนินการ (Action) ต่างๆโดยใช้กฎเกณฑ์

สิ่งที่เกี่ยวข้องเพิ่มเติมในการใช้กลยุทธ์ฐานข้อมูล

- มีศักยภาพในการควบคุมมาตรฐาน: เป็นส่วนที่สำคัญมากของแอปพลิเคชันฐานข้อมูลที่ประสบความสำเร็จในองค์กรใหญ่ๆ โดยมาตรฐานในที่นี้ หมายถึง ชื่อรายการข้อมูล รูปแบบการแสดงผลหน้าจอ โครงสร้างรายงาน และ meta-data (นิยามของข้อมูล) เป็นต้น
- ลดเวลาในการพัฒนาแอปพลิเคชัน: เวลาที่ใช้ในการเพิ่มแอปพลิเคชันใหม่ลดลง

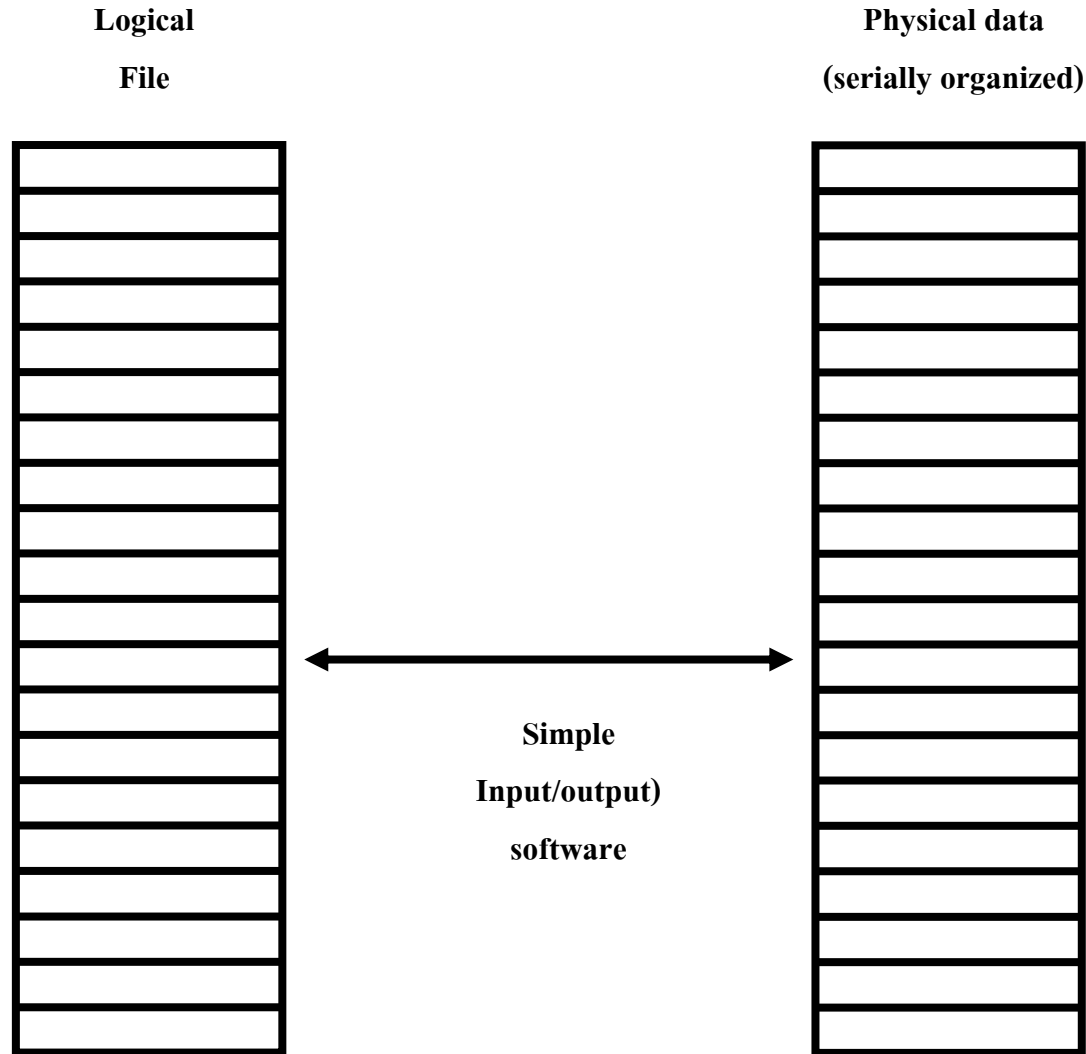
สิ่งที่เกี่ยวข้องเพิ่มเติมในการใช้กลยุทธ์ฐานข้อมูล (ต่อ)

- มีความยืดหยุ่นในการเปลี่ยนแปลงโครงสร้างข้อมูล: สามารถเปลี่ยนแปลงไปตามความต้องการที่ระบุได้
- มีการปรับข้อมูลให้ทันสมัยอยู่ตลอดเวลา: สำคัญมากสำหรับธุรกิจออนไลน์ เช่น สายการบินโรงแรม การเช่ารถ เป็นต้น
- Economies of scale: โดยการรวบรวมข้อมูลและแอปพลิเคชันระหว่างแผนก ซึ่งลดความสิ้นเปลืองจากความซ้ำซ้อนของทรัพยากรและบุคคลได้

วิวัฒนาการของ DBMS (โดย James Martin)

- Stage 1 : ไฟล์ข้อมูลพื้นฐาน – มีอิทธิพลในช่วงต้นทศวรรษที่ 60
- Stage 2 : กระบวนการเข้าถึงไฟล์ – มีอิทธิพลในช่วงปลายทศวรรษที่ 60
- Stage 3 : ระบบฐานข้อมูลในช่วงแรกเริ่ม – มีอิทธิพลในช่วงต้นทศวรรษที่ 70
- Stage 4 : ระบบฐานข้อมูลในปัจจุบัน – ตั้งแต่ปี 1970 จนถึงปัจจุบัน

ช่วงที่ 1 : ไฟล์ข้อมูลพื้นฐาน (มีอิทธิพลในช่วงต้นทศวรรษที่ 60)



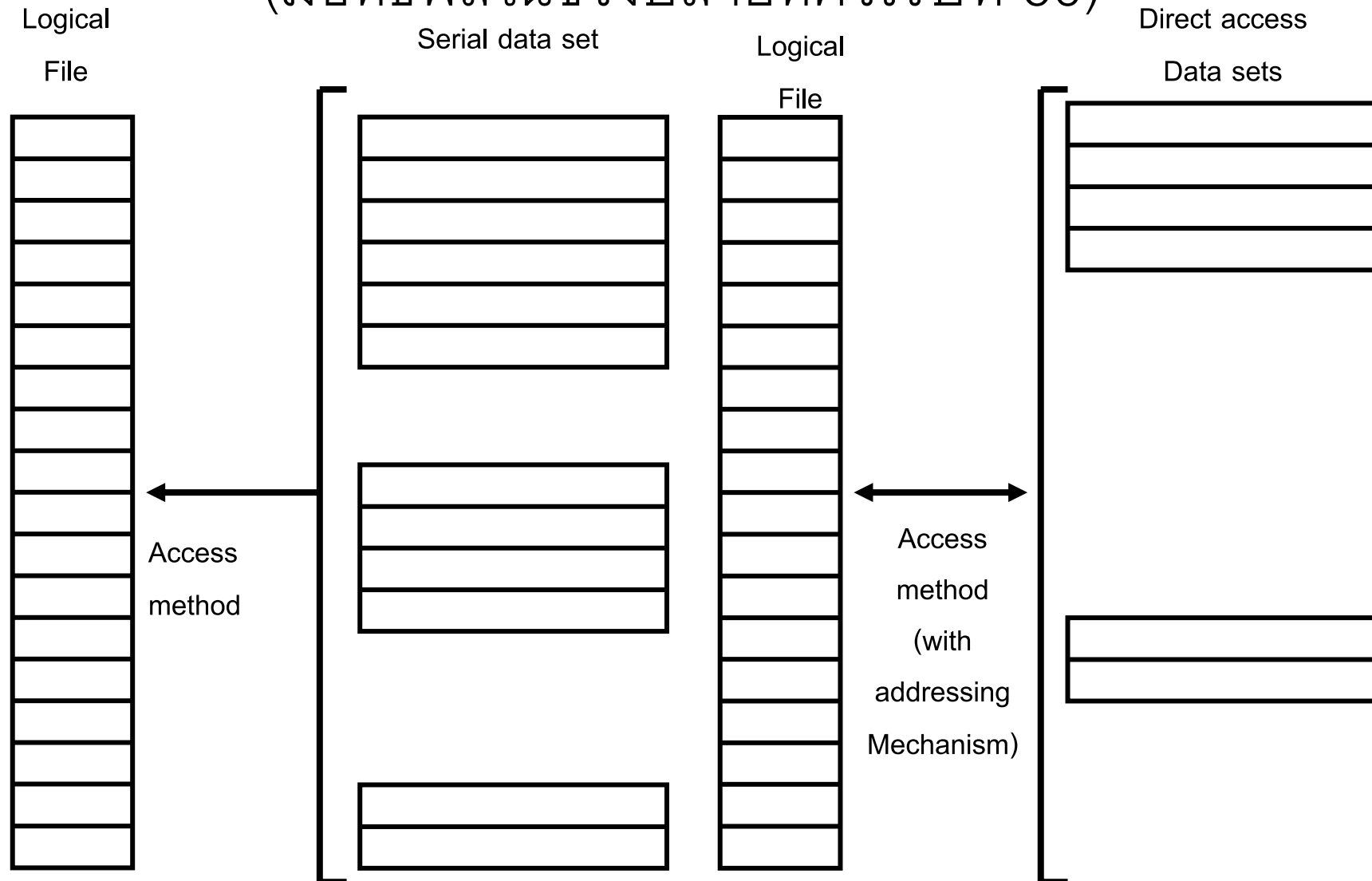
ช่วงที่ 1 : ไฟล์ข้อมูลพื้นฐาน

- ในการจัดระบบไฟล์ในรูปแบบลำดับอนุกรม
- โครงสร้างข้อมูลทางกายภาพมีลักษณะเดียวกับโครงสร้างข้อมูลทางตรรก
- ใช้กระบวนการจัดการแบบกลุ่ม (Batch-processing) โดยไม่มีการเข้าถึงข้อมูลแบบทันที (Real-time)
- มีการจัดเก็บไฟล์ที่ซ้ำซ้อน เนื่องจากข้อมูลเดิมยังคงถูกเก็บไว้
- ซอฟต์แวร์สามารถจัดการได้เพียงขั้นตอน input/output เท่านั้น

ช่วงที่ 1 : ไฟล์ข้อมูลพื้นฐาน (ต่อ)

- ผู้พัฒนาโปรแกรมแอปพลิเคชันจะทำการออกแบบรูปแบบทางกายภาพและฝังติดไว้กับโปรแกรมแอปพลิเคชัน
- ถ้ามีการเปลี่ยนแปลงโครงสร้างข้อมูลหรือส่วนควบคุมการเก็บข้อมูล (Storage device) จะต้องทำการเขียนโปรแกรมรวมทั้งแปลงภาษาและทดลองใหม่อีกด้วย
- ข้อมูลมักจะถูกออกแบบให้ทำงานอย่างมีประสิทธิภาพกับแอปพลิเคชันหนึ่งๆเท่านั้น
- เกิดการซ้ำซ้อนระหว่างไฟล์ข้อมูลมาก

ช่วงที่ 2 : กระบวนการเข้าถึงไฟล์ (มีอิทธิพลในช่วงปลายทศวรรษที่ 60)



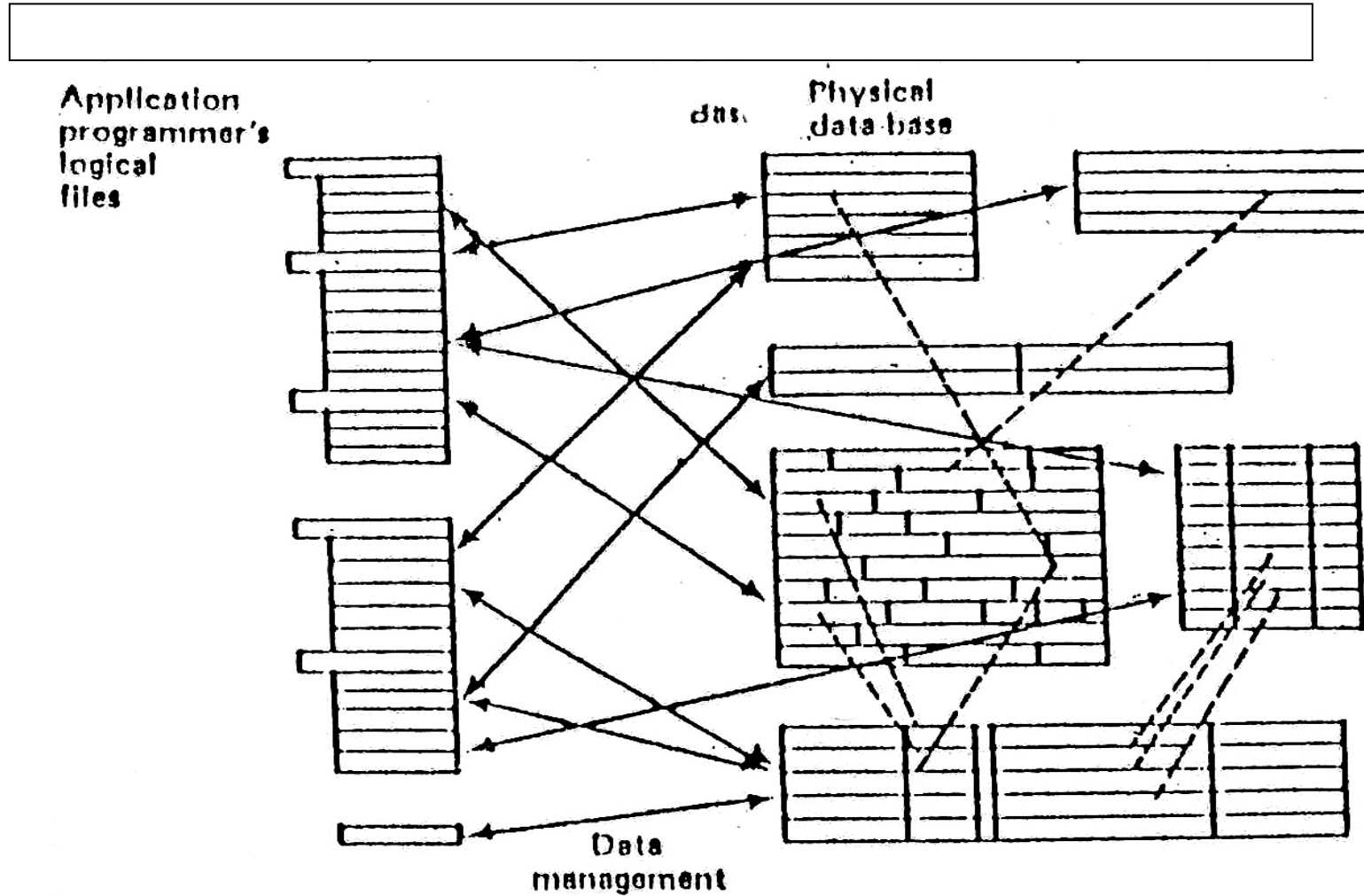
ช่วงที่ 2 : กระบวนการเข้าถึงไฟล์

- สามารถทำการบันทึกได้โดยการเข้าถึงข้อมูลแบบตามลำดับ (Serial) หรือแบบสุ่ม (Random) (ไม่ใช้การบันทึกลงในขอบเขตของข้อมูล (Field))
- ใช้กระบวนการจัดการแบบกลุ่ม (Batch) แบบอินไลน์ (In-line) หรือแบบทันที (Real-time)
- ไฟล์ทางตรรกและไฟล์ทางกายภาพนั้นแยกออกจากกัน
- เปลี่ยนแปลงหน่วยเก็บข้อมูลได้โดยไม่จำเป็นต้องทำการเปลี่ยนแปลงโปรแกรม
- โครงสร้างข้อมูล อยู่ในรูปแบบลำดับอนุกรม (Serial) ลำดับเชิงดัชนี (Index sequential) หรือ Simple direct access

ช่วงที่ 2 : กระบวนการเข้าถึงไฟล์ (ต่อ)

- ไม่มีการใช้คีย์หลายตัว (Multiple-key) ในการสืบค้นข้อมูล
- อาจมีการตรวจสอบความปลอดภัยของข้อมูล แต่นับว่ายังไม่ค่อยปลอดภัย
- ยังคงมีแนวโน้มที่จะออกแบบข้อมูลให้มีประสิทธิภาพกับแอปพลิเคชันหนึ่งๆ อยู่
- ยังคงพบความซ้ำซ้อนของข้อมูลจำนวนมากอยู่

ช่วงที่ 3 : ระบบฐานข้อมูลในช่วงแรกเริ่ม (มีอิทธิพลในช่วงต้นทศวรรษที่ 70)



ช่วงที่ 3 : ระบบฐานข้อมูลในช่วงแรกเริ่ม

- สามารถอนุมานไฟล์ทางตรรกที่แตกต่างกันได้มากมาย จากข้อมูลทางกายภาพเดียวกัน
- สามารถเข้าถึงข้อมูลเดียวกันได้จากหลายทางด้วยการกำหนดความต้องการที่ต่างกัน
- ซอฟต์แวร์มีความสามารถในการลดความซ้ำซ้อนของข้อมูล
- มีการใช้ข้อมูลร่วมกันระหว่างแอปพลิเคชันต่างๆ
- การไม่มีความซ้ำซ้อนของข้อมูล เอื้อต่อความคงสภาพของข้อมูล (Data integrity)

ช่วงที่ 3 : ระบบฐานข้อมูลในช่วงแรกเริ่ม (ต่อ)

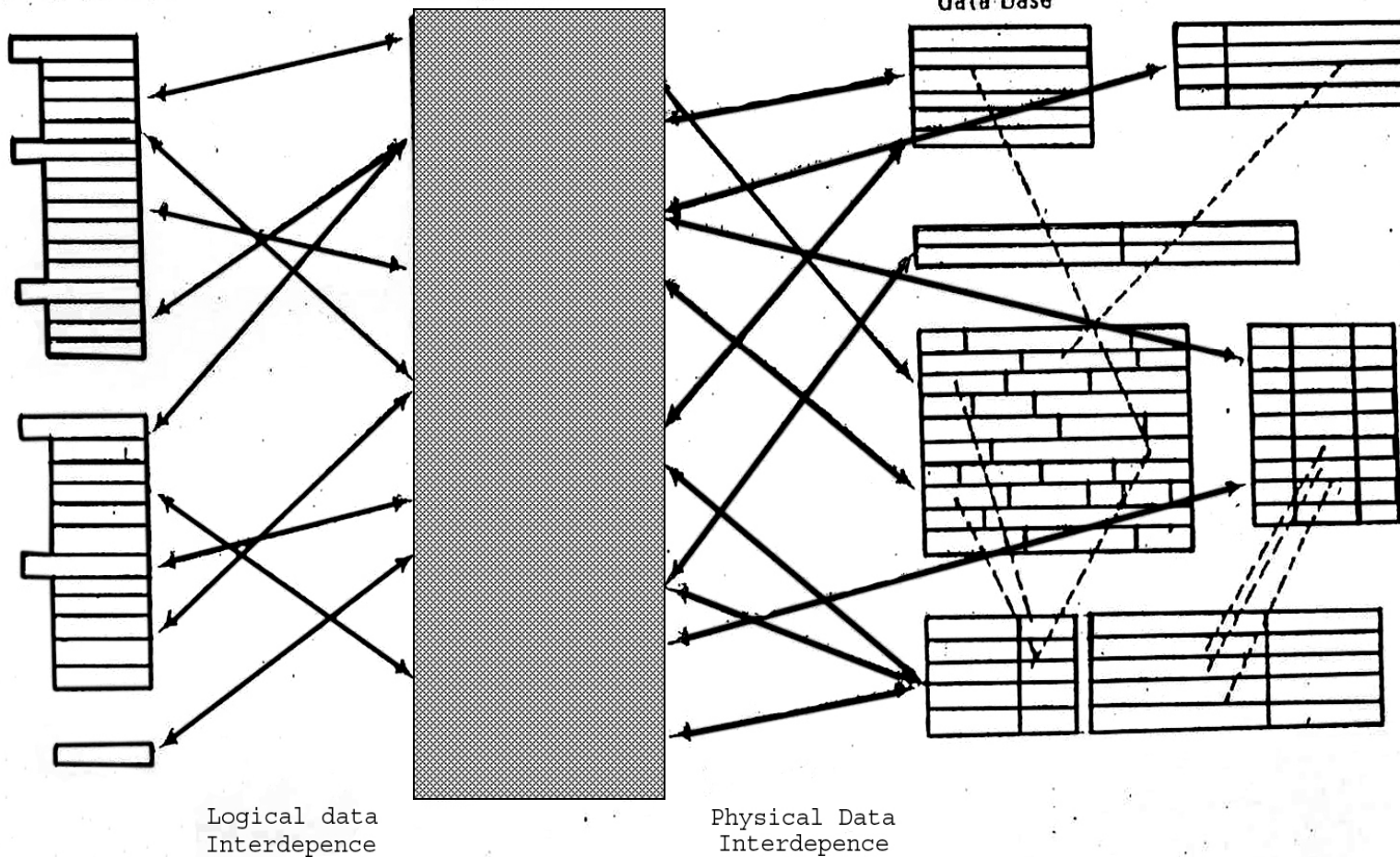
- การจัดระบบหน่วยเก็บข้อมูลทางกายภาพนั้นไม่ขึ้นกับโปรแกรมแอปพลิเคชัน
- สามารถกำหนดตำแหน่งที่อยู่ของข้อมูลในขอบเขตของข้อมูล (Filed) หรือในระดับกลุ่ม (Group level)
- สามารถใช้คีย์หลายตัว (Multiple-key) ในการสืบค้นได้
- ใช้การจัดระบบข้อมูลที่ซับซ้อนโดยไม่ทำให้โปรแกรมแอปพลิเคชันซับซ้อน

ช่วงที่ 4 : ความต้องการของระบบฐานข้อมูลในปัจจุบัน

Application programmer
logical files; External
views of the data

Global logical
data base description;
Conceptual schema

Physical
data base



ช่วงที่ 4 : ความต้องการของระบบฐานข้อมูลในปัจจุบัน

- ซอฟต์แวร์มีข้อมูลทางตรรกและทางกายภาพที่ไม่ขึ้นต่อกัน
- สามารถทำการพัฒนาฐานข้อมูลให้ดีขึ้น โดยไม่ต้องทำการดูแลรักษามาก
- มีส่วนอำนวยความสะดวกให้ผู้บริหารฐานข้อมูล ในการควบคุมข้อมูล
- มีการใช้ขั้นตอนที่มีประสิทธิภาพในการควบคุมความเป็นส่วนบุคคล ความปลอดภัย และความคงสภาพ (Integrity) ของข้อมูล

ประวัติการพัฒนาเทคโนโลยีฐานข้อมูล

- **แอปพลิเคชันฐานข้อมูลในช่วงแรก** : มีการนำเสนอโมเดลแบบแตกสาขา (Hierarchical model) และโมเดลแบบเครือข่าย (Network model) ในช่วงกลางทศวรรษที่ 70 และมีอิทธิพลในช่วงทศวรรษที่ 70 – ซึ่งยังคงพบการใช้โมเดลเหล่านี้ในการปฏิบัติการฐานข้อมูลทั่วโลกจำนวนมาก
- **ระบบที่ใช้พื้นฐานของโมเดลความสัมพันธ์** : IBM และมหาวิทยาลัยต่างๆได้ทำการวิจัยและทดลองแบบจำลองที่ได้นำเสนอในช่วงปี 1970 เป็นอย่างหนัก ซึ่งทำให้เกิดเป็น DBMS เชิงสัมพันธ์ (Relational DBMS Products) ขึ้นในช่วงทศวรรษที่ 80

ประวัติการพัฒนาเทคโนโลยีฐานข้อมูล (ต่อ)

- **Object-oriented applications:** ได้มีการนำเสนอ OODBMSs ขึ้นในช่วงปลายทศวรรษที่ 80 ถึงต้นทศวรรษที่ 90 เพื่อสนองความต้องการในการประมวลผลที่ซับซ้อนในโปรแกรมช่วยการออกแบบ (CAD) และแอปพลิเคชันอื่นๆ แต่การใช้งานยังไม่เป็นที่นิยมมากนัก
- **ข้อมูลในเว็บและแอปพลิเคชันการค้าอิเล็กทรอนิกส์:** เว็บบรรจุข้อมูลในรูปแบบของ HTML (Hypertext Markup Language) โดยมีการเชื่อมโยงระหว่างหน้า ก่อให้เกิดการพัฒนาแอปพลิเคชันใหม่ๆ ขึ้นจำนวนมาก รวมถึงการค้าอิเล็กทรอนิกส์ (Ecommerce) ซึ่งปัจจุบันใช้มาตรฐานใหม่ คือ XML (eXtended Markup Language)

การขยายความสามารถของฐานข้อมูล

- มีการเพิ่มฟังก์ชันใหม่ๆ ให้กับ DBMSs ในด้านต่างๆ ดังต่อไปนี้:
 - แอปพลิเคชันทางวิทยาศาสตร์
 - หน่วยเก็บและจัดการข้อมูลภาพ
 - การจัดการข้อมูลเสียงและภาพ
 - การค้นข้อมูล (Data Mining)
 - Spatial data management
 - การจัดการข้อมูลลำดับเวลาและประวัติข้อมูล (Time Series and Historical Data Management)

ซึ่งทั้งหมดนี้ทำให้เกิดความก้าวหน้าในการวิจัยและพัฒนาข้อมูลชนิดใหม่ๆ โครงสร้างข้อมูลที่ซับซ้อน คำสั่งและหน่วยเก็บข้อมูลใหม่ๆ รวมทั้งการเก็บดัชนี (Index schemes) ในระบบฐานข้อมูล

กรณีที่ไม่ควรใช้ DBMS

- **DBMS มีค่าใช้จ่ายที่สูง:**
 - การลงทุนเริ่มต้นทางด้านฮาร์ดแวร์สูง
 - มีค่าใช้จ่ายเพิ่มในการให้กำหนดกฎทั่วไป (Generality) ความปลอดภัย (Security) การควบคุมร่วมกัน (Concurrency control) การฟื้นฟูสภาพ (Recovery) และฟังก์ชันควบคุมความคงสภาพ (Integrity function)
- **กรณีที่ไม่จำเป็นต้องใช้ DBMS:**
 - เมื่อฐานข้อมูลและแอปพลิเคชันมีลักษณะไม่ซับซ้อน มีการระบุที่ชัดเจน และไม่ต้องการการเปลี่ยนแปลงใดๆ
 - เมื่อต้องการความทันกาลในการปรับเปลี่ยนให้ตรงความต้องการตลอดเวลา ซึ่งอาจทำไม่ได้เนื่องจากค่าใช้จ่ายที่สูงของ DBMS
 - เมื่อไม่ต้องการให้ผู้ใช้เข้าถึงข้อมูลได้หลายๆคน

กรณีที่ไม่ควรใช้ DBMS (ต่อ)

- กรณีที่ DBMS ไม่มีความสามารถเพียงพอ :
 - เมื่อระบบฐานข้อมูลไม่สามารถจัดการกับข้อมูลที่ซับซ้อน เนื่องจากข้อจำกัดของโมเดล
 - เมื่อผู้ใช้งานข้อมูลต้องการคำสั่งพิเศษที่ไม่มีใน DBMS