

หลักการและความหมายฐานข้อมูล

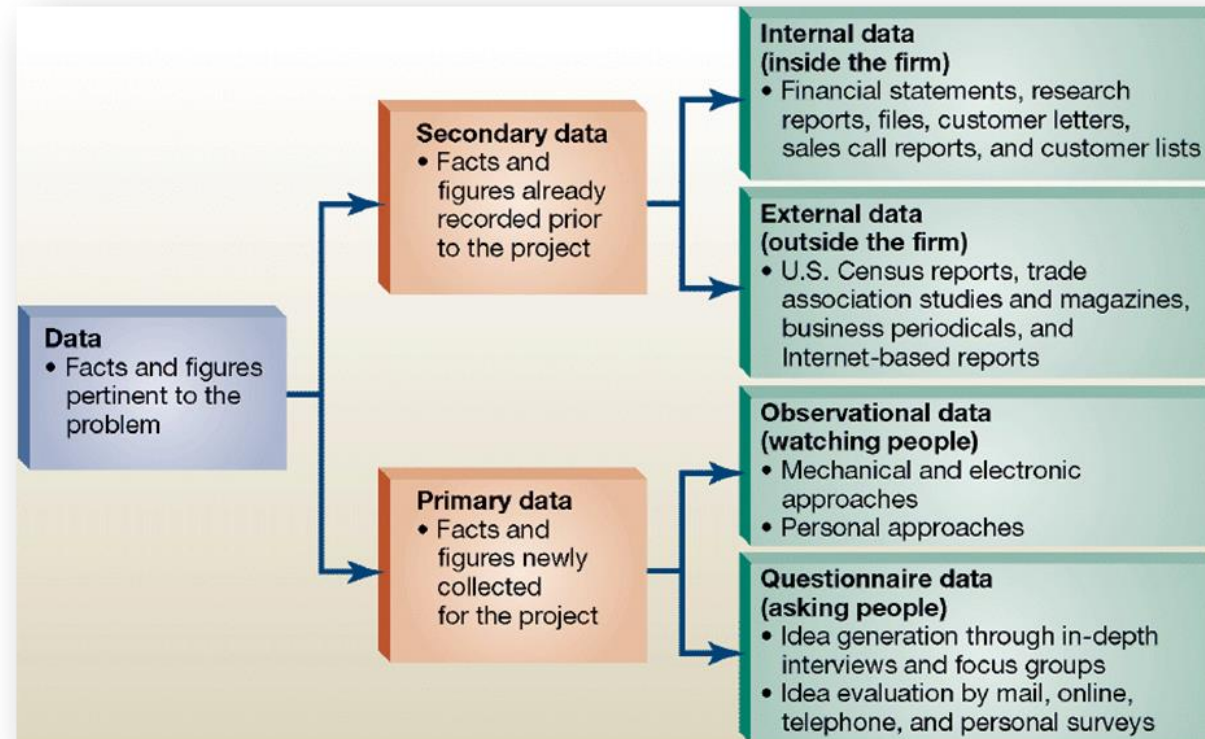
GEO3411

การออกแบบฐานข้อมูลสารสนเทศภูมิศาสตร์



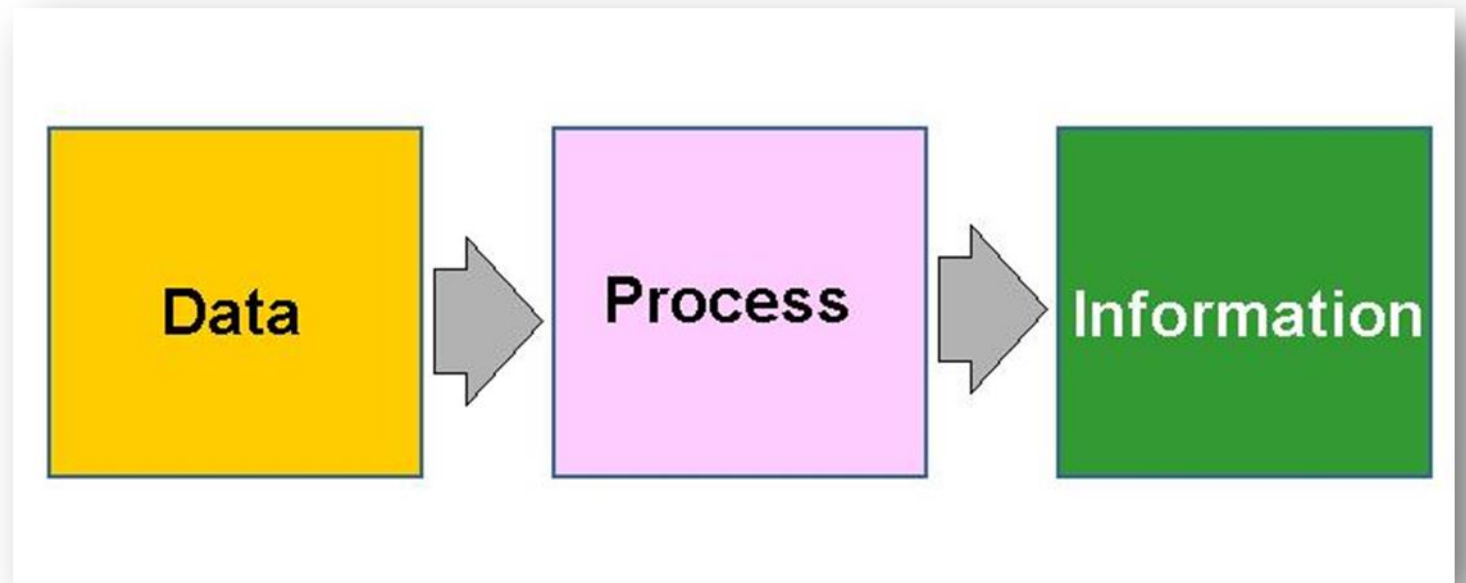
ข้อมูล (Data)

- หมายถึง ข่าวสาร เอกสาร ข้อเท็จจริงเกี่ยวกับบุคคล สิ่งของหรือเหตุการณ์ในรูปแบบของ ตัวเลข ภาพ ตัวอักษร และสัญลักษณ์ต่างๆ เช่น คะแนนสอบวิชาภาษาไทย ราคาสินค้า จำนวนนักเรียนในโรงเรียน



สารสนเทศ (Information)

- หมายถึง ข้อมูลต่างๆ ที่ได้ผ่านการเปลี่ยนแปลงหรือมีการประมวลผลหรือวิเคราะห์สรุปผลด้วยวิธีการต่าง ๆ แล้วเก็บรวบรวมไว้ เพื่อนำมาใช้ประโยชน์ตามต้องการการประมวล (Data Processing) เป็นการนำข้อมูลจากแหล่งต่างๆ ที่เก็บรวบรวมไว้มาผ่านกระบวนการต่างๆ เพื่อแปรสภาพข้อมูลให้เป็นระบบและอยู่ในรูปแบบที่ต้องการ



ฐานข้อมูล

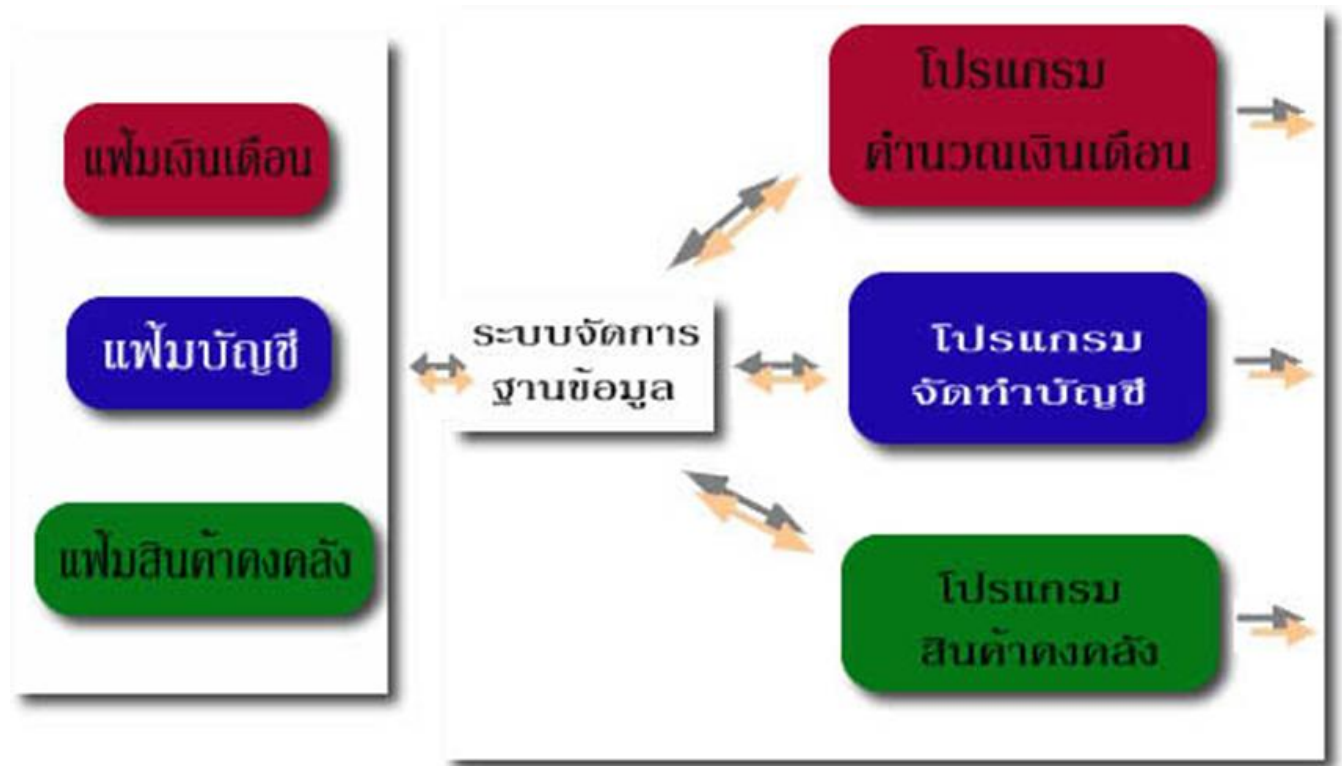


- ฐานข้อมูล คือ ชุดของข้อมูลที่มีความสัมพันธ์ที่ถูกนำมาจัดเก็บไว้ด้วยกัน เพื่อสามารถใช้ข้อมูลเหล่านั้นร่วมกันได้อย่างรวดเร็วและมีประสิทธิภาพ เช่น สมุดโทรศัพท์ หรือ การจัดเก็บข้อมูลค่าใช้จ่าย เป็นต้น



ฐานข้อมูล (Database)

- ฐานข้อมูล (Database) เป็นวิธีการที่จะเก็บข้อมูลที่มีความสัมพันธ์กันไว้ในที่เดียวกันและรวบรวมข้อมูลที่ไม่ซ้ำซ้อนและสามารถใช้ข้อมูลร่วมกันได้อย่างเป็นระบบ ให้สะดวกต่อการเรียกใช้ สามารถแก้ไขได้ง่ายสำหรับผู้ใช้งานจำนวนมาก และสามารถป้องกันไม่ให้ผู้ไม่มีสิทธิ์ใช้เข้าถึงข้อมูลได้



ระบบเพิ่มข้อมูลที่จัดทำด้วยมือ

แผนกบัญชี

การขาย

ลูกค้า

แผนกการขาย

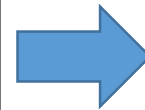
การขาย

ลูกค้า

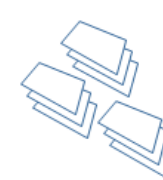
สินค้า

แผนกบุคคล

พนักงาน

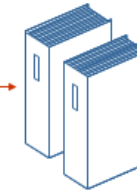


ดัชนีรายการแฟ้มเอกสาร	
ลำดับที่	รายการ
1	ToDoList ประจำวัน
2	หนังสือเข้า-ออก พอ.200 พอ.208.01 ศธ.0521.1.10
3	หนังสือขออนุมัติเดินทางไปราชการโครงการ NEC
4	ใบเสร็จรับเงิน/ใบสำคัญรับเงิน โครงการ NEC 13-14
5	แบบฟอร์มเอกสารทั่วไป โครงการ NEC 13-14
6	แบบฟอร์มอนุมัติทำงานล่วงเวลา
7	แบบฟอร์มทำงานล่วงเวลา (นักศึกษา โครงการ NEC)
8	แบบฟอร์มเอกสารทั่วไป โครงการ NEC 13-14
9	ใบบันทึกการปฏิบัติงานของทีปธิกรณา
10	แบบฟอร์ม AAR โครงการ NEC 13-14



เอกสาร

จัดเก็บ



จัดเข้าแฟ้มเอกสาร

เข้าระบบ

จัดเก็บ

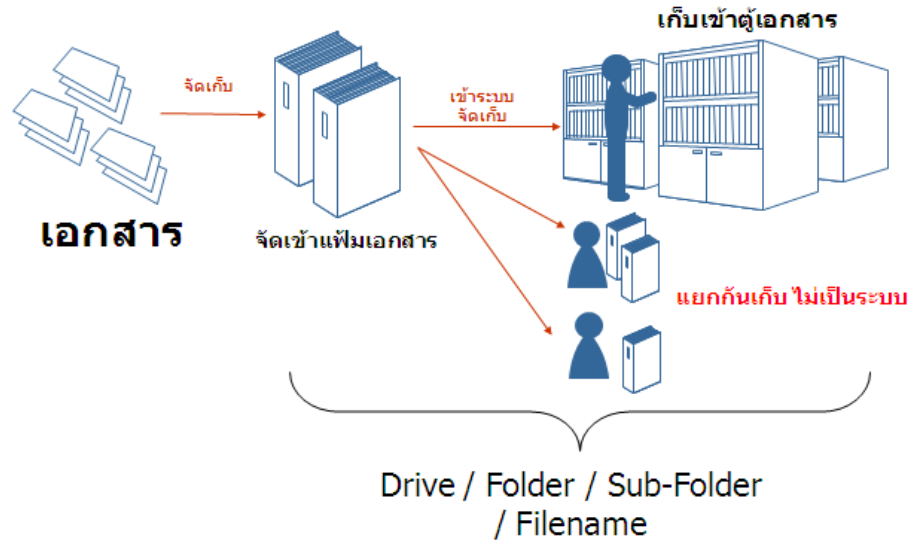
เก็บเข้าตู้เอกสาร



แยกกันเก็บ ไม่เป็นระบบ



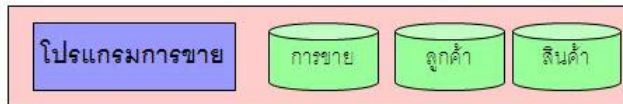
ระบบเพิ่มข้อมูลด้วยคอมพิวเตอร์



ฝ่ายบัญชี



ฝ่ายขาย



ฝ่ายบุคคล



customer	Order Deta...	Ord...	Product		
Custom	First Name	Last Name	City	Business Phone	E-Mail
	1 Sombat	SaeLim	Chicago	123555100-0	sombat_s@abc.com
	2 Somsri	Meedee	Miami	123555100-0	somsri_m@def.com
	3 Nipa	Sailom	New York	123555100-0	nipa_s@hotmail.com
	4 Thongchai	Rungkul	Denver	123555100-0	thonchai_r@aa.com
	5 Saijai	Chaiyo	Chicago		saijai_c@bb.com
	6 Prasit	Namchai	Miami	123555100-0	prasit_n@cc.com
	7 Snong	Sakul	New York	123555100-0	snong_s@dd.com
	8 Wipa	Janrueng	Denver	123555100-0	wipa_j@ee.com
	9 Somkid	Pongsawat	Chicago	123555100-0	somkid_p@ff.com
	10 Watchara	Deesakul	Miami	123555100-0	watchart@gg.com
	11 Nisa	Chaya	New York	123555100-0	nisa_c@hh.com
	12 Sarun	Lertamorn	Denver	123555100-0	sarun_l@ii.com
	13 Kalya	Sanitkul	Chicago	123555100-0	kalya_s@jj.com
	14 Arun	Wanaphan	Miami	123555100-0	arun_w@aa.com
	15 Suksan	Santi	New York		suksan_s@bb.com
	16 Boonprasit	Chudjane	Denver	123555100-0	boonprasit_c@dd.com
	17 Onuma	Komkum	Chicago	123555100-0	onuma_k@aa.com
	18 Napaporn	Yungyuen	Miami	123555100-0	nipaporn_y@ff.com
	19 Charles	Smith	New York		john_s@bb.com
	20 Nipon	Aemjai	Denver	123555100-0	nipon_a@ff.com
	21 Sataporn	Jamjai	Chicago	123555100-0	sataporn_j@bb.com
*					

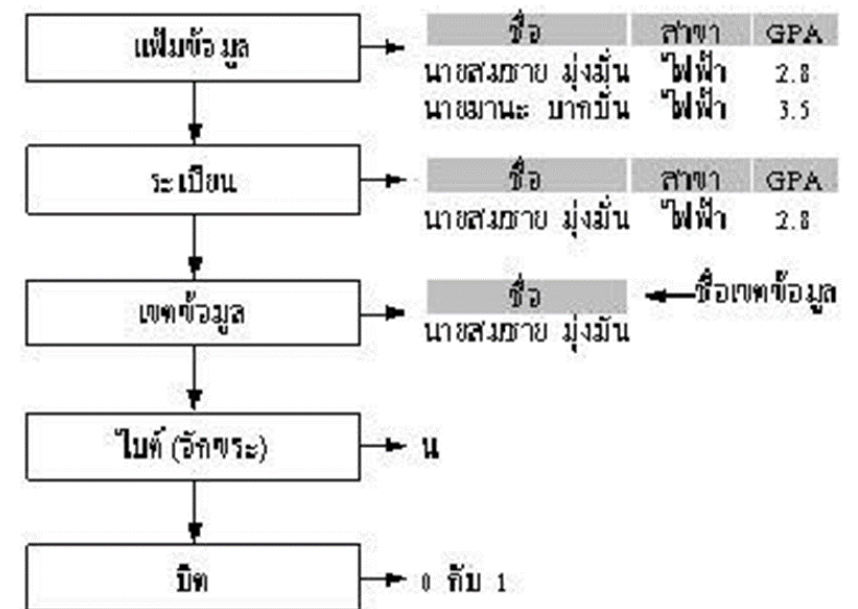
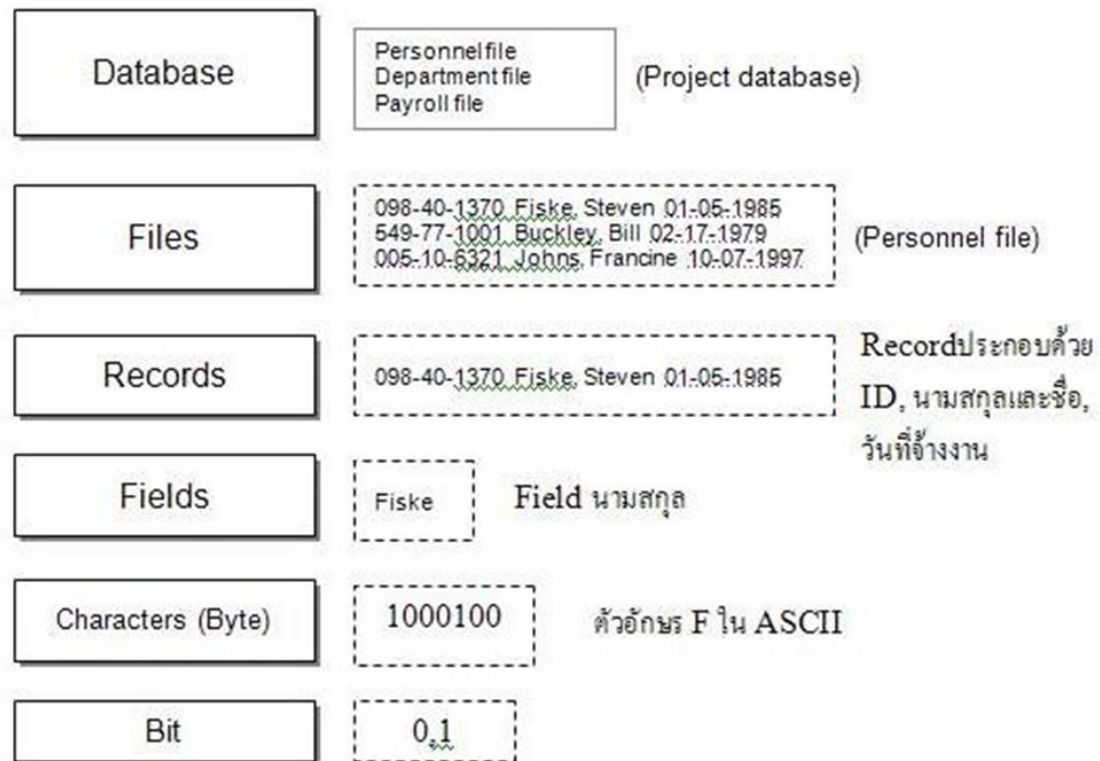
แฟ้มข้อมูล (File)

- เป็นการจัดเก็บและรวบรวมข้อมูลประเภทเดียวกันไว้ด้วยกันเพื่อให้เกิดความสะดวกในการจัดเก็บและเรียกค้น ตามวัตถุประสงค์การใช้งานของหน่วยงาน
- การจัดเก็บข้อมูลของหน่วยงานไว้แต่ละหน่วยงานหรือแต่ละแผนก เช่น งานบุคคลก็จะมีแฟ้มเกี่ยวกับบุคลากรขององค์กร งานวิชาการก็จะมีแฟ้มเกี่ยวข้องกับวิชาการที่หน่วยงานเผยแพร่หรือวิจัย เป็นต้น และจะมีโปรแกรมที่เขียนขึ้นมาเพื่อเรียกค้นข้อมูลของหน่วยงานตนเองเพื่อการประมวลผล

โครงสร้างของข้อมูลพื้นฐาน

- **บิต (Bit)** เป็นข้อมูลพื้นฐานที่เล็กที่สุดภายในแฟ้มข้อมูลเป็นฐานข้อมูล **bit** จะแทนด้วย 1 หรือ 0 ใดๆอย่างหนึ่งเพียงอย่างเดียว
- **ไบต์ (Byte)** เป็นข้อมูลขนาด 1 ตัวอักขระ ซึ่งประกอบไปด้วยข้อมูลจำนวน 7 บิต หรือ 8 บิต
- **เขตข้อมูล (Field)** เป็นข้อมูลที่เกิดจากตัวอักขระหลาย ๆ ตัวรวมกันเป็นคำที่มีความหมายในตัวเอง
- **ระเบียน (Record)** เป็นข้อมูลที่เกิดจากการรวมกันของเขตข้อมูลหลาย ๆ เขตข้อมูล ที่มีความสัมพันธ์กัน
- **แฟ้มข้อมูล (File)** เกิดจากการนำระเบียนข้อมูลประเภทเดียวกัน มารวมกัน
- **ฐานข้อมูล (Database)** เกิด จากการนำแฟ้มข้อมูลหลาย ๆ แฟ้มมารวมกันไว้ที่เดียวกัน แต่ฐานข้อมูลจะมีการเก็บคำอธิบายเกี่ยวกับโครงสร้างฐานข้อมูล ที่เรียกว่า พจนานุกรมข้อมูล (**Data Dictionary**) หรือ เบต้าดาต้า ทำหน้าที่อธิบายลักษณะข้อมูลที่เก็บอยู่ในฐานข้อมูล รวมทั้งความสัมพันธ์ของข้อมูลที่เก็บอยู่ในฐานข้อมูล พจนานุกรมข้อมูลนี้ จึงถูกเรียกใช้งานในระหว่างที่มีการประมวลผลฐานข้อมูล

โครงสร้างของข้อมูลพื้นฐาน



รูปที่ 1.4 หน่วยข้อมูลต่างๆ

ปัญหาที่เกิดขึ้นในการใช้แฟ้มข้อมูล



1. Data redundancy
2. Data inconsistency
3. Data anomaly



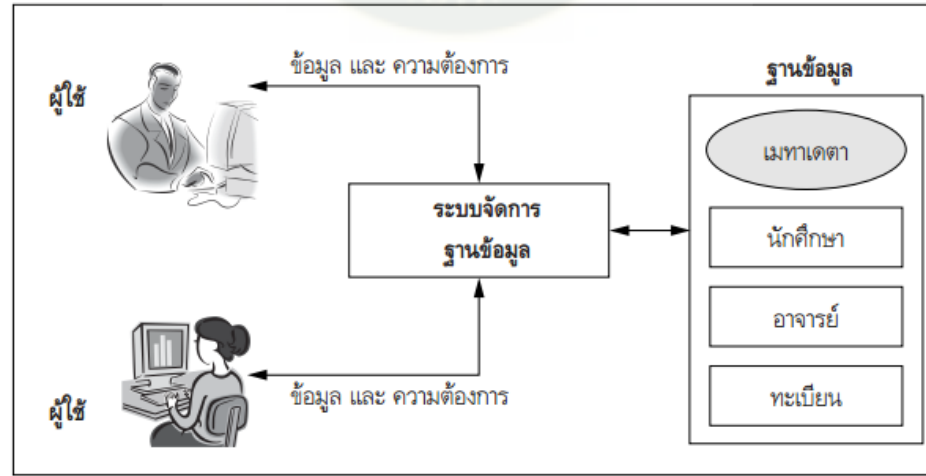
ระบบจัดการฐานข้อมูล

GEO3411

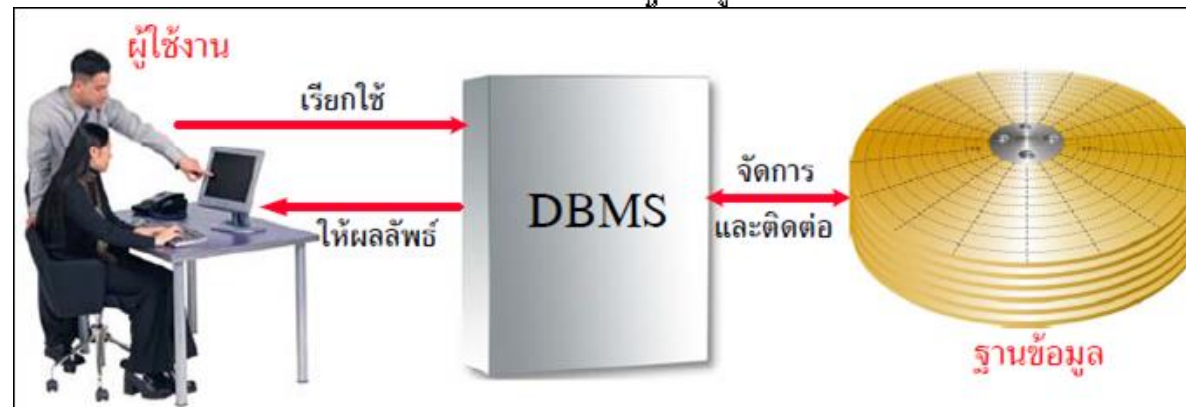
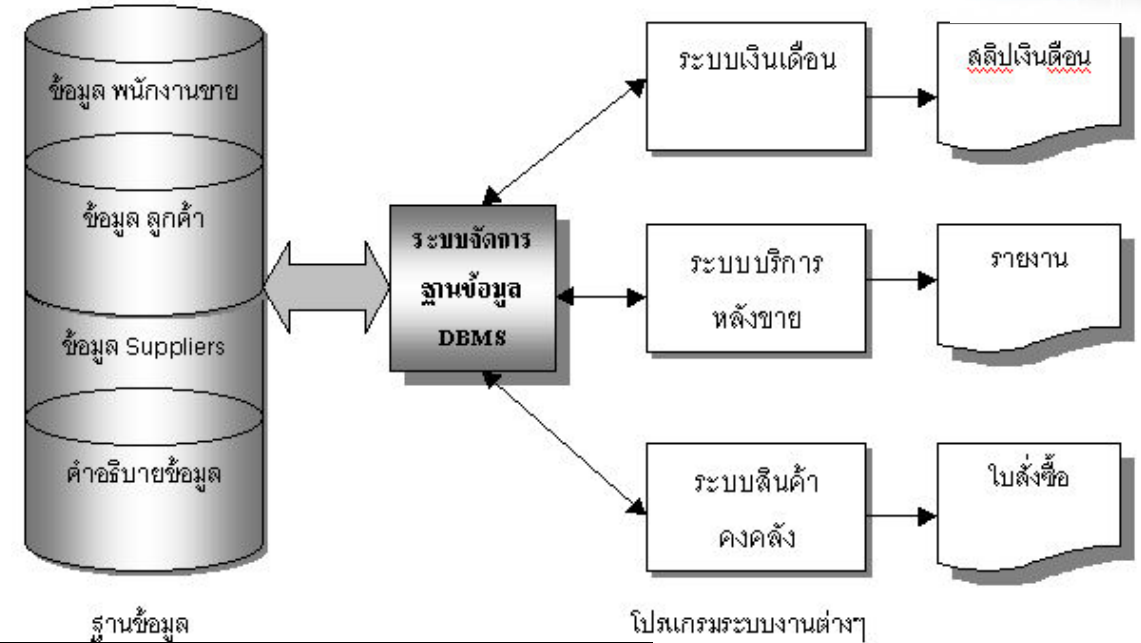
การออกแบบฐานข้อมูลสารสนเทศภูมิศาสตร์



ระบบจัดการฐานข้อมูล



รูปที่ 2.1 ความสัมพันธ์ระหว่างระบบจัดการฐานข้อมูล ผู้ใช้ และฐานข้อมูล





หน้าที่ของระบบจัดการฐานข้อมูล

- กำหนดและเก็บโครงสร้างฐานข้อมูล
- รับและเก็บข้อมูลในฐานข้อมูล
- ดูแลรักษาข้อมูล
- ติดต่อกับตัวจัดการระบบแฟ้มข้อมูล
- ควบคุมความบูรณภาพของข้อมูล (Integrity Control)
- ควบคุมความปลอดภัย (Security Control)
- การสร้างระบบสำรองและการกู้ (Backup and Recovery)
- ควบคุมภาวการณ์ใช้ข้อมูลพร้อมกันของผู้ใช้ (Concurrency Control)



ภาษาที่ใช้ในการเข้าถึงฐานข้อมูล

ภาษาคำนิยามของข้อมูล [Data Definition Language (DDL)]

เป็นภาษาที่ใช้กำหนดโครงสร้างข้อมูลหรือนิยามข้อมูลของฐานข้อมูล (Database Schema) ซึ่งโครงสร้างข้อมูลหรือนิยามข้อมูลสามารถเรียกได้อีกอย่างว่า สกีม่า (Schema)

การกำหนดโครงสร้างข้อมูลหรือนิยามข้อมูลเช่นการกำหนดว่าฐานข้อมูลที่เราสร้างมีชื่อว่าอะไร มีโครงสร้างประกอบด้วยตาราง (table) ชื่ออะไรบ้าง แต่ละตาราง (table) ประกอบด้วย เขตข้อมูลใดบ้าง เขตข้อมูล แต่ละตัวมีประเภทของข้อมูลเป็นอะไร ความกว้างของข้อมูลเท่าใด แต่ละตารางมีอินเด็กซ์ (index) ช่วยในการค้นหาข้อมูลหรือไม่ ถ้ามีจะใช้เขตข้อมูลใดบ้างเป็นคีย์ เป็นต้น

ภาษาคำนิยามของข้อมูล

Data Definition Language (DDL)



- **CREATE** คือคำสั่งสำหรับการสร้างนิยาม หรือสร้างฐานข้อมูลนั่นเอง หากเราต้องการสร้างฐานข้อมูล เราสามารถใช้รูปแบบคำสั่งได้ดังนี้

```
1 CREATE TABLE <ชื่อตารางข้อมูล>
2 (
3   <ชื่อ Field> <ชนิดข้อมูล (ขนาดข้อมูล)>,
4   <ชื่อ Field> <ชนิดข้อมูล (ขนาดข้อมูล)>
5 )
```

- **ALTER** คือคำสั่งในการแก้ไขโครงสร้างของตารางข้อมูล เช่น เราต้องการแก้ไข ชื่อ **Field** หรือแก้ไของค์ประกอบต่าง ๆ ของตารางข้อมูล มีคำสั่งโดยย่อดังนี้

```
1 ALTER TABLE <ชื่อตารางข้อมูล> DROP COLUMN <ชื่อ Field>
2 ALTER TABLE <ชื่อตารางข้อมูล> ADD COLUMN <ชื่อ Field><ชนิดข้อมูล (ขนาดข้อมูล)>
```

- **DROP** คำคำสั่งสำหรับลบตารางข้อมูล มีคำสั่งโดยย่อดังนี้

```
1 DROP TABLE <ชื่อตารางข้อมูล>
```



ภาษาที่ใช้ในการเข้าถึงฐานข้อมูล

ภาษาการจัดการฐานข้อมูล (**Data Manipulation Language (DML)**)

การจัดการข้อมูลหมายถึง การเพิ่มข้อมูลใหม่ลงในฐานข้อมูล, การลบข้อมูลที่มีอยู่ในฐานข้อมูล, การเปลี่ยนแปลงแก้ไขข้อมูลในฐานข้อมูล และการค้นคืนข้อมูลจากฐานข้อมูล



ภาษาการจัดการฐานข้อมูล

(Data Manipulation Language (DML))



- **SELECT** คือคำสั่งสำหรับสืบค้นข้อมูล หรือค้นหาข้อมูลที่จัดเก็บอยู่ในตารางข้อมูล โดยมักนิยมใช้งานรวมกับคำสั่ง **WHERE** เพื่อใช้ในการสร้างเงื่อนไขในการแสดงผลข้อมูล โดยมีคำสั่งโดยย่อดังนี้

```
1 | SELECT * FROM <ชื่อตาราง> WHERE ( <เงื่อนไข> )
```

- **INSERT** คือคำสั่งสำหรับการเพิ่มข้อมูลลงไปในตารางข้อมูล เช่นการเพิ่มข้อมูลพนักงาน, สินค้า เป็นต้น โดยมีคำสั่งโดยย่อดังนี้

```
1 | INSERT INTO <ชื่อตาราง> VALUES ( <ค่าข้อมูล>, <ค่าข้อมูล>, ... N )
```

- **UPDATE** คือคำสั่งสำหรับการปรับปรุง หรือแก้ไขข้อมูลในตารางข้อมูล โดยสามารถใช้งานรวมกับคำสั่ง **WHERE** เพื่อสร้างเงื่อนไขในการแก้ไขข้อมูล

```
1 | UPDATE <ชื่อตาราง> SET <ชื่อ Field> = <ค่าข้อมูล> WHERE ( <เงื่อนไข> )
```

- **DELETE** คือคำสั่งในการลบข้อมูลในตารางข้อมูล โดยสามารถใช้งานรวมกับคำสั่ง **WHERE** เพื่อสร้างเงื่อนไขสำหรับการลบข้อมูล

```
1 | DELETE FROM <ชื่อตาราง> WHERE ( <เงื่อนไข> )
```

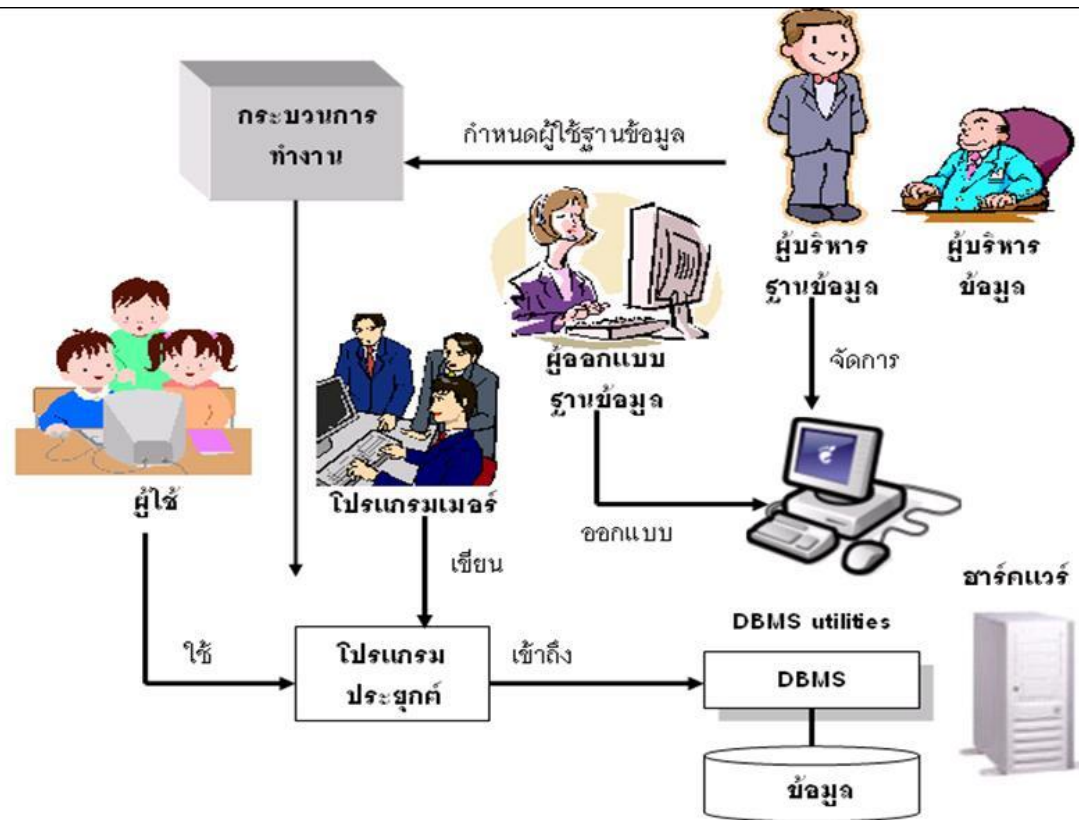
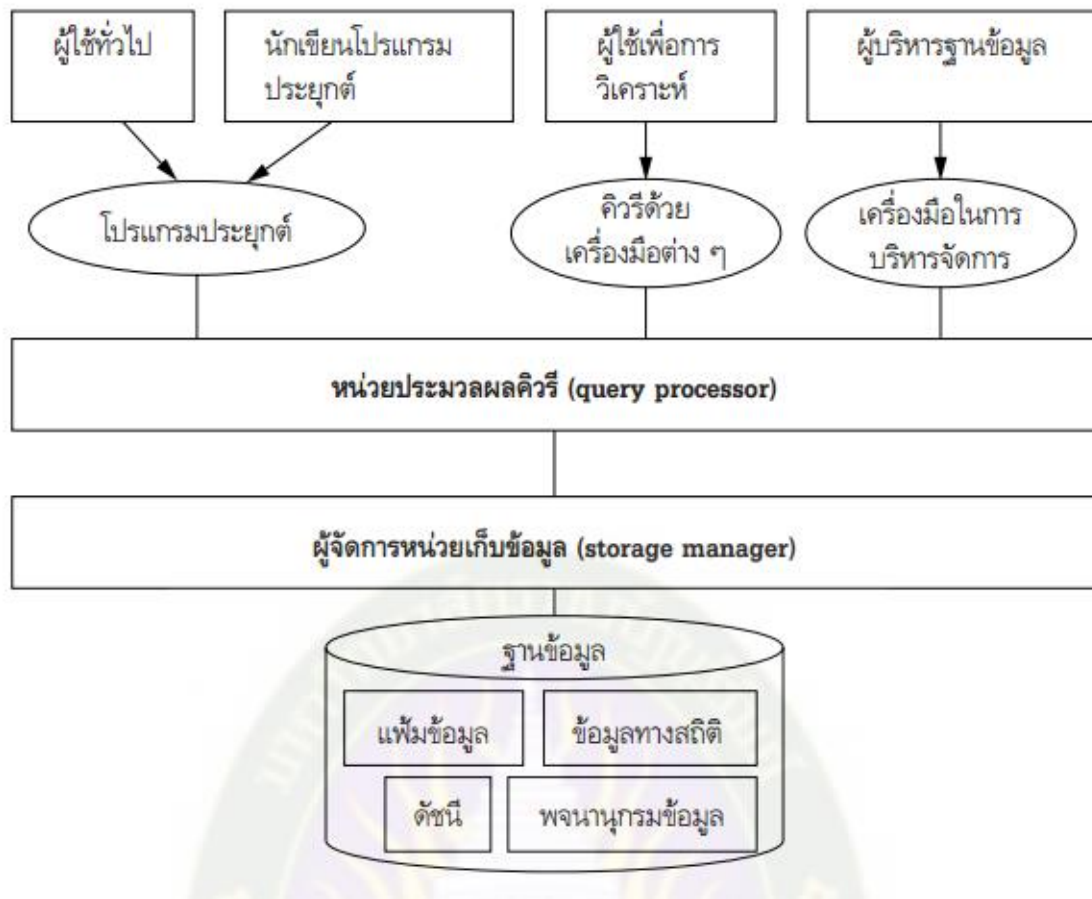
Example SQL Syntax

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

- Example
 - SELECT CustomerName,City FROM Customers;
 - UPDATE Customers
 - SET ContactName='Alfred Schmidt', City='Hamburg'
 - WHERE CustomerName='Alfreds Futterkiste';
 - DELETE FROM table_name
 - WHERE some_column=some_value;



การทำงานของระบบจัดการฐานข้อมูล





บุคคลที่เกี่ยวข้องกับระบบจัดการฐานข้อมูล

- นักเขียนโปรแกรมประยุกต์
- ผู้ใช้เพื่อการวิเคราะห์ต่างๆ
- ผู้ใช้ทั่วไป
- ผู้บริหารฐานข้อมูล



ข้อดีและข้อเสียของการใช้งานระบบฐานข้อมูล

ข้อดี

- ทำให้ข้อมูลสามารถใช้ร่วมกันได้
- สามารถควบคุมเรื่องความปลอดภัย
- การพัฒนาและบำรุงรักษา สามารถกำหนดเป็นมาตรฐาน
- ลดความซ้ำซ้อนของฐานข้อมูล
- สามารถหลีกเลี่ยงความไม่สอดคล้องของข้อมูล
- สามารถรักษาความถูกต้องของข้อมูลในระบบ
- มีความอิสระของข้อมูล
- มีทฤษฎีที่สนับสนุนการทำงานที่ชัดเจน โดยเฉพาะอย่างยิ่งในระบบฐานข้อมูลเชิงสัมพันธ์

ข้อเสีย

- ระบบมีความซับซ้อน
- มีค่าใช้จ่ายสูง
- ต้องมีซอฟต์แวร์และฮาร์ดแวร์
- หากระบบเกิดความล้มเหลว จะทำให้เกิดผลกระทบกับการทำงานขององค์กรอย่างกว้างขวาง
- การกู้คืนทำได้ยาก

Normalization

GEO3411

การออกแบบฐานข้อมูลสารสนเทศศาสตร์



1

การนอร์มัลไลเซชัน (Normalization)



- ขบวนการ **Normalization** แนวคิดนี้ถูกคิดค้นโดย E.F.Codd ซึ่งเป็นกระบวนการที่นำเค้าร่างของ **relation** มาทำให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน (**Normal Form**) เพื่อให้แน่ใจว่าการออกแบบเค้าร่างของ **relation** เป็นการออกแบบที่เหมาะสม

2

ทำไมต้องทำการนอร์มัลไลซ์?



- ลดที่ว่างที่ต้องใช้ในการเก็บข้อมูล
- ลดความผิดพลาด ความไม่ตรงกันของข้อมูลในฐานข้อมูล
- ลดการเกิดออร์มัลไลซ์ของการลบและแก้ไขข้อมูล
- เพิ่มความคงทนแก่โครงสร้างฐานข้อมูล

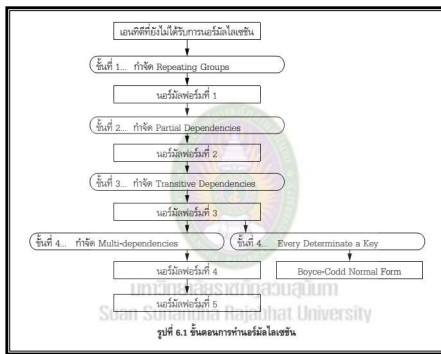
3

ระดับนอร์มัลไลเซชัน



- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
 - BCNF (Boyce/Codd Normal Form)
- 4NF (Forth Normal Form)

4



5

P#	P_Name	E#	E_Name	Job_Class	Chg/Hour	Hours	Total
11	ID Card	103	สงชาย ไม้ดี	Elect.Engineer	500	23.8	11900
		101	พวัน งานมิ่ง	Database Designer	500	19.4	9700
		105	ชาย ศุภี	Database Designer	500	35.7	17850
		106	เม็ก ยอดมิ่ง	Programmer	200	12.6	2520
		102	อวร ศุภี	System Analyst	400	23.8	9520
			Subtotals				51490
22	Palm	114	สุศักดิ์ สังาม	Application Designer	200	24.6	4920
		118	กมล ไม้งาม	General Support	100	45.3	4530
		104	นาย ย้อยอด	System Analyst	250	32.4	8100
		112	ซีกา ไม้งาม	Des Analyst	200	44.0	8800
			Subtotals				26350
33	Web Service	105	ชาย ศุภี	Database Designer	500	64.7	32350
		104	นาย ย้อยอด	System Analyst	250	48.4	12100
		113	วิธ ยิงดี	Application Designer	250	23.6	5900
		111	พิน งานดี	Clerical Support	200	22.0	4400
		106	เม็ก ยอดมิ่ง	Programmer	200	12.8	2560
			Subtotals				57310

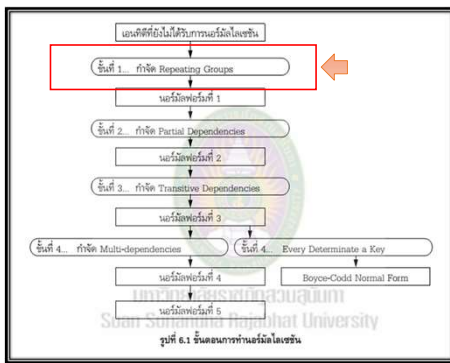
รูปที่ 6.2 รายการค่าใช้จ่ายในแต่ละโครงการ

6

ปัญหา

- เลขโครงการน่าจะมาใช้เป็นคีย์หลักคันทึ่มค่า null
- ข้อมูลมีความไม่แน่นอน เช่น SA กับ System analyst
- มีความซ้ำซ้อนของข้อมูล
 - การแก้ไข
 - การเพิ่ม
 - การลบ

7



8

First Normal Form (1NF)

- ทุก ๆ field ในแต่ละ record จะเป็น single value นั่นคือ ในตารางหนึ่ง ๆ จะไม่มี ค่าของกลุ่มข้อมูลที่ซ้ำกัน (Repeating Group) ตัวอย่างเช่น ตารางดังต่อไปนี้ (1NF)

EXPENSE	P_Name	Job	E_Name	Job_Class	Chg_Hour	Hours
11	ID Card	101	สมชาย ไม้ดี	SA	500	23.8
		101	เจ้านาย ไม้ดี	Database Designer	500	24.4
		106	ชาญ ศรีดี	Database Designer	500	35.7
		109	เจ้านาย ไม้ดี	Programmer	350	12.6
		101	ชาญ ศรีดี	System Analyst	400	23.8
22	Palm	114	สุชาติ ฝ้าย	Application Designer	250	24.6
		118	ชาญ ไม้ดี	General Support	100	45.3
		104	ชาญ ฝ้าย	System Analyst	250	35.4
		112	ชาญ ไม้ดี	DB Analyst	200	44.5
33	Web Service	105	ชาญ ศรีดี	Database Designer	500	54.7
		104	ชาญ ฝ้าย	System Analyst	250	48.4
		113	ชาญ ไม้ดี	Application Designer	250	23.6
		111	ชาญ ไม้ดี	General Support	200	25.9
		108	เจ้านาย ไม้ดี	Programmer	300	12.8

รูปที่ 6.3 ตาราง EXPENSE ที่มีค่าซ้ำกันในกลุ่มข้อมูลซ้ำกัน

EXP	P_Name	EJ	E_Name	Job_Class	Chg_Hour	Hours
11	ID Card	101	สมชาย ไม้ดี	SA	500	23.8
11	ID Card	101	เจ้านาย ไม้ดี	Database Designer	500	24.4
11	ID Card	106	ชาญ ศรีดี	Database Designer	500	35.7
11	ID Card	109	เจ้านาย ไม้ดี	Programmer	350	12.6
11	ID Card	101	ชาญ ศรีดี	System Analyst	400	23.8
22	Palm	114	สุชาติ ฝ้าย	Application Designer	250	24.6
22	Palm	118	ชาญ ไม้ดี	General Support	100	45.3
22	Palm	104	ชาญ ฝ้าย	System Analyst	250	35.4
22	Palm	112	ชาญ ไม้ดี	DB Analyst	200	44.5
33	Web Service	105	ชาญ ศรีดี	Database Designer	500	54.7
33	Web Service	104	ชาญ ฝ้าย	System Analyst	250	48.4
33	Web Service	113	ชาญ ไม้ดี	Application Designer	250	23.6
33	Web Service	111	ชาญ ไม้ดี	General Support	200	25.9
33	Web Service	108	เจ้านาย ไม้ดี	Programmer	300	12.8

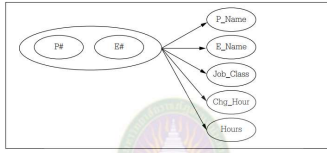
รูปที่ 6.3 ตาราง EXPENSE ที่มีค่าซ้ำกันในกลุ่มข้อมูลซ้ำกัน

9

First Normal Form (1NF)

นิยามของ 1NF

1. มีการกำหนดแอตทริบิวต์ที่เป็นคีย์
2. ต้องไม่มีวันที่ซ้ำกัน แต่ละแถวหรือคอลัมน์จะมีค่าได้เพียง 1 ค่าเท่านั้น
3. แอตทริบิวต์ทุกตัวต้องขึ้นอยู่กับคีย์หลัก ดังแสดงในรูปที่ 6.6



10

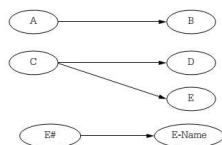
รูปที่ 6.5 แสดงตารางที่เป็น 1NF

P#	P_Name	E#	E_Name	Job_Class	Chg_Hour	Hours
11	ID Card	103	สมชาย ไม้สี	Elect Engineer	500	23
11	ID Card	101	เจน งามยิ่ง	Database Designer	500	19
11	ID Card	105	ชาน ศรีดี	Database Designer	500	35
11	ID Card	106	แม็ก ยอดยิ่ง	Programmer	200	12
11	ID Card	102	อวร ศรีดี	System Analyst	400	23
22	Palm	114	สุวิทย์ ดีงาม	Application Designer	200	24
22	Palm	118	ภาณุ ไม้งาม	General Support	100	45
22	Palm	104	นาย ยืนยง	System Analyst	250	32
22	Palm	112	ธิดา ไม้งาม	Des Analyst	200	44
33	Web Service	105	ชาน ศรีดี	Database Designer	500	64
33	Web Service	104	นาย ยืนยง	System Analyst	250	48
33	Web Service	113	วิธ ยืนยง	Application Designer	250	23
33	Web Service	111	หิน งามดี	Cherial Support	200	22
33	Web Service	106	แม็ก ยอดยิ่ง	Programmer	200	12

รูปที่ 6.5 แสดงตารางที่เป็น 1NF

11

ในการกำหนดโมเดลเซชันนั้น ผู้เขียนจะต้องเข้าใจหลักการของฟังก์ชันหนึ่งต่อหนึ่งแบบเวียน ดังนั้น การที่กล่าวหาว่า "B is functionally dependent on A if a value of A uniquely determines a value of B" หรือ B ขึ้นอยู่กับ A ถ้าทราบค่าของ A ก็จะทำให้รู้ค่าของ B ได้ สามารถเขียนแสดงเป็นภาพได้ดังแสดงในรูปที่ 6.7



ฟังก์ชันหนึ่งต่อหนึ่งแบบเวียน สามารถแสดงด้วยการใช้ลูกศร ดังนี้

$A \rightarrow B$ แสดง B เป็นฟังก์ชันหนึ่งต่อหนึ่งแบบเวียนต่อกับ A กล่าวคือ ถ้ารู้ค่า A ก็จะทำให้ทราบค่าของ B ด้วย
ทุกค่าของ A ที่มีค่าเท่ากัน จะได้ค่า B เท่ากันเสมอ

12

Partial Dependency

$P\# \rightarrow P_Name$
 $E\# \rightarrow E_Name, Job_Class, Chg_Hour$

(1NF)

รูปที่ 6.8 แสดงการขึ้นชื่อถึงบนแบบ

- ความสัมพันธ์แบบหนึ่งต่อหนึ่งได้ก็ต่อเมื่อ **relation** หนึ่ง ๆ มีคีย์หลักเป็นคีย์ผสม (composite key) นั่นคือ คีย์หลักของ **relation** นั้น ๆ ประกอบด้วย **attribute** หลาย **attribute** ความสัมพันธ์ระหว่างค่าของ **attribute** แบบบางส่วนเกิดขึ้นเมื่อ **attribute** บางส่วนของคีย์หลักสามารถระบุค่าของ **attribute** อื่น ๆ ที่ไม่ใช่คีย์หลักของ **relation** ได้ (**Non-key attribute**) จึงความสัมพันธ์แบบนี้จะทำไม่ได้ ศึกษานู้นหาในเรื่องของความซ้ำซ้อน และการปรับปรุงข้อมูล

13

รูปที่ 6.8 แสดงการขึ้นชื่อถึงบนแบบ

P#	P_Name	E#	E_Name	Job_Class	Chg_Hour	Hours
11	ID Card	103	สมชาย ไม้ดี	Elect.Engineer	600	23
11	ID Card	101	สมชาย งามยิ่ง	Database Designer	600	19
11	ID Card	105	ชาย ศรีดี	Database Designer	600	36
11	ID Card	106	เม็ก งามยิ่ง	Programmer	200	12
11	ID Card	102	สมศรี ศรีดี	System Analyst	400	23
22	Palm	114	สุวิทย์ ดีงาม	Application Designer	200	24
22	Palm	118	สมชาย ไม้งาม	General Support	100	45
22	Palm	104	นาย สมยศ	System Analyst	250	32
22	Palm	112	ธิดา ไม้งาม	Das Analyst	200	44
33	Web Service	105	ชาย ศรีดี	Database Designer	500	64
33	Web Service	104	นาย สมยศ	System Analyst	250	48
33	Web Service	113	วิภา ยิงดี	Application Designer	250	23
33	Web Service	111	สมชาย งามดี	Choral Support	200	22
33	Web Service	106	เม็ก งามยิ่ง	Programmer	200	12

รูปที่ 6.9 แสดงตารางที่เป็น 1NF

14

รูปที่ 6.1 ขั้นตอนการทำนอร์มัลไลเซชัน

15

Second Normal Form (2NF)

- ต้องเป็น First Normal Form (1NF)
- ต้องไม่มี **Partial Dependency**

P#
E#
P#, E#

↓

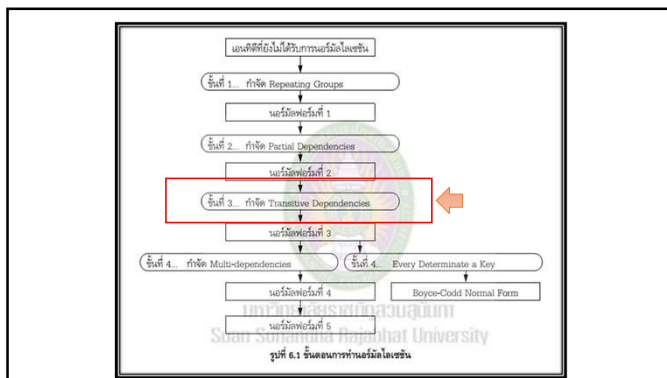
PROJECT(P# P_Name)
 EMPLOYEE(E# E_Name, Job_Class, Chg_Hour)
 WORK(P# E# Hours)

→

รูปที่ 6.9 ตัวอย่างการกำจัด Partial Dependency 2NF

(2NF)

16



17

Transitive Dependency

E#
E#
E#

↓

PROJECT(P# P_Name)
 EMPLOYEE(E# E_Name, Job_Class, Chg_Hour)
 WORK(P# E# Hours)

→

รูปที่ 6.10 ตัวอย่างการกำจัด Transitive Dependency

• attribute ที่มีคุณสมบัติเป็นคีย์หลักจะสามารถระบุค่าของทุก attribute ในแต่ละ tuples ได้ อย่างไรก็ตาม ในบาง relation อาจจะมี attribute ขึ้น (Non-key attribute) ที่สามารถระบุค่าของ attribute ขึ้น ๆ ใน tuples ได้ ลักษณะของความสัมพันธ์ในการระบุค่า attribute แบบนี้ เรียกว่า Transitive Dependency

18

Third Normal Form (3NF)

- ต้องเป็น **Second Normal Form (2NF)**
- ไม่มี **Transitive Dependency**

PROJECT(P#, P_Name)
WORK(P#, E#, Hours)
EMPLOYEE(E#, E_Name, Job_Class) (3NF)
JOB(Job_Class, Chg_Hour)

19

แนะนำท้ายบท

- วัตถุประสงค์ในการทำให้เป็นรูปแบบนอร์มัลไลซ์ก็คือ เพื่อลดปัญหาความซ้ำซ้อนของข้อมูล และลดปัญหาในเรื่องการเพิ่ม การปรับปรุง หรือลบข้อมูล โดยทั่วไปแล้วการออกแบบในระนาบความคิด ผู้ออกแบบจะพยายามวิเคราะห์ให้อยู่ในรูปแบบ **3NF** แต่ถ้ามีกรณีของปัญหาที่จำเป็นต้องทำต่อไปถึงรูปแบบของ **BCNF** หรือ **4NF** หรือ **5NF** (ซึ่งเกิดขึ้นน้อยมากในทางปฏิบัติ) แต่อย่าพยายามแต่กรณีไหนมากเกิดความจำเป็น เพราะการแต่กรณีออกมามากเกินไปนั้น จะมีผลต่อประสิทธิภาพในการทำงานของฐานข้อมูลนั้น ๆ เช่น ในการค้นหาข้อมูลจะใช้เวลามาก เป็นต้น

20

Reference

- วิเชียร เปรมชัยสวัสดิ์. (2546). ระบบฐานข้อมูล. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).
- สุเพชร จิระจกุล (2555). เรียนรู้ระบบสารสนเทศภูมิศาสตร์ด้วย โปรแกรม ArcGIS 10.1 for Desktop. นนทบุรี: บริษัท เอพี.กราวพิคดิไซน์และการพิมพ์ จำกัด.
- Ronald Briggs(2004). GIS Fundamentals
- สำราญกรมไทยสำหรับเยาวชนฯ / เล่มที่ ๒๕

<http://kanchanapisek.or.th/kp6/sub/book/book.php?book=25&chap=2&page=t25-2-infodetail08.html>

21

ระบบฐานข้อมูลภูมิสารสนเทศและ ฐานข้อมูลเชิงสัมพันธ์

GEO3411

การออกแบบฐานข้อมูลสารสนเทศภูมิศาสตร์





ระบบฐานข้อมูลภูมิสารสนเทศ

- ระบบข้อมูลภูมิสารสนเทศ
- แบบจำลอง
- ประเภทข้อมูล
- รูปแบบความสัมพันธ์
- ฐานข้อมูลเชิงสัมพันธ์
- แบบจำลอง E-R

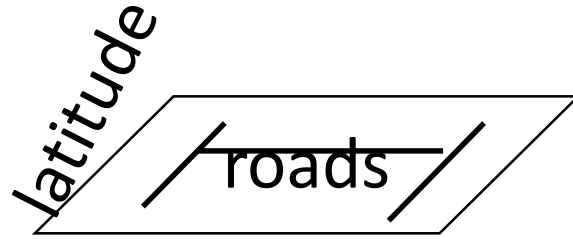


GIS Data Model

- allows the **geographic features** in **real world locations** to be digitally represented and stored in a database so that they can be abstractly presented in **map** (analog) form, and can also be worked with and **manipulated** to address some **problem**



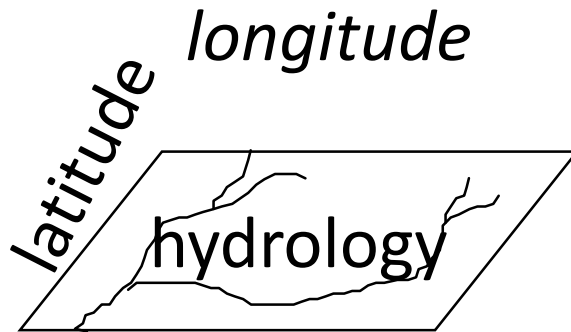
SPATIAL DATA MODELS



Here we have three layers or themes:

- roads,
- hydrology (water),
- topography (land elevation)

They can be related because precise geographic coordinates are recorded for each theme.

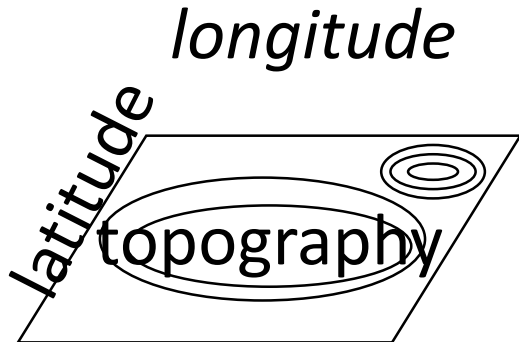


Layers are comprised of two data types

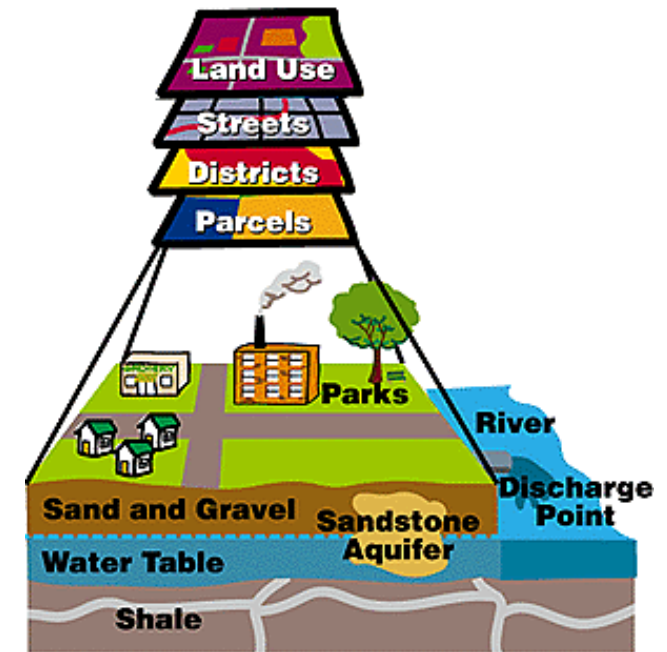
- *Spatial data* which describes location (where)
- *Attribute data* specifying what, how much, when

Layers may be represented in two ways:

- in *vector* format as points and lines
- in *raster(or image)* format as pixels



All geographic data has 4 properties:
projection, scale, accuracy and resolution





Spatial and Attribute Data

- Spatial data (*where*)
 - specifies location
 - stored in a *shape file*, *geodatabase* or similar geographic file
- Attribute (descriptive) data (*what, how much, when*)
 - specifies characteristics at that location, natural or human-created
 - stored in a data base table

GIS systems traditionally maintain spatial and attribute data separately, then “join” them for display or analysis

- for example, in ArcView, the *Attributes of ...* table is used to link a *shapefile* (spatial structure) with a *data base table* containing attribute information in order to display the attribute data spatially on a map



Representing Data with *Raster* and *Vector* Models

Raster Model

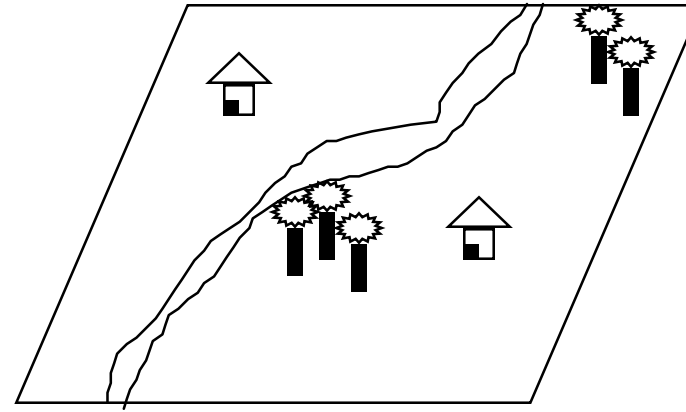
- area is covered by grid with (usually) equal-sized, square cells
- attributes are recorded by assigning each cell a single value based on the majority feature (attribute) in the cell, such as land use type.
- *Image* data is a special case of raster data in which the “attribute” is a reflectance value from the geomagnetic spectrum
 - cells in image data often called *pixels* (picture elements)
- **Vector Model**

The fundamental concept of vector GIS is that all geographic features in the real world can be represented either as:

- **points or dots (*nodes*)**: trees, poles, fire plugs, airports, cities
- **lines (*arcs*)**: streams, streets, sewers,
- **areas (*polygons*)**: land parcels, cities, counties, forest, rock type

Because representation depends on shape, ArcView refers to files containing vector data as *shapefiles*

Concept of Vector and Raster



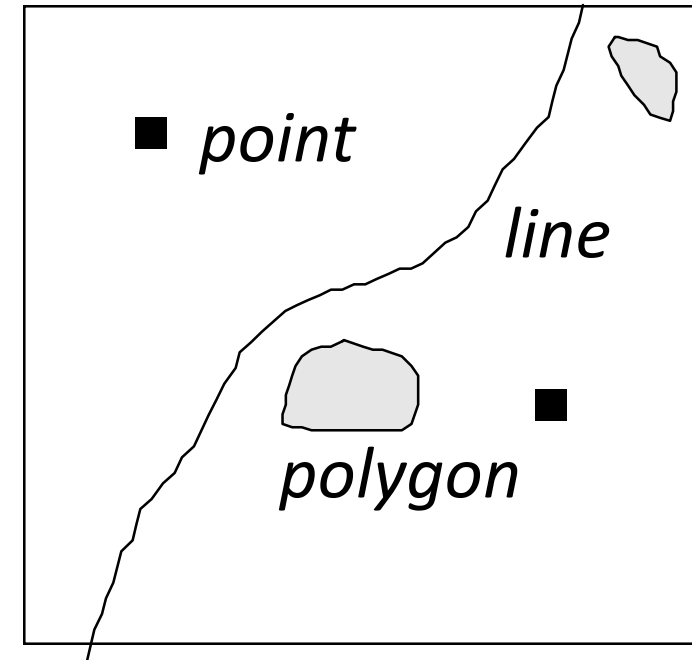
Real World



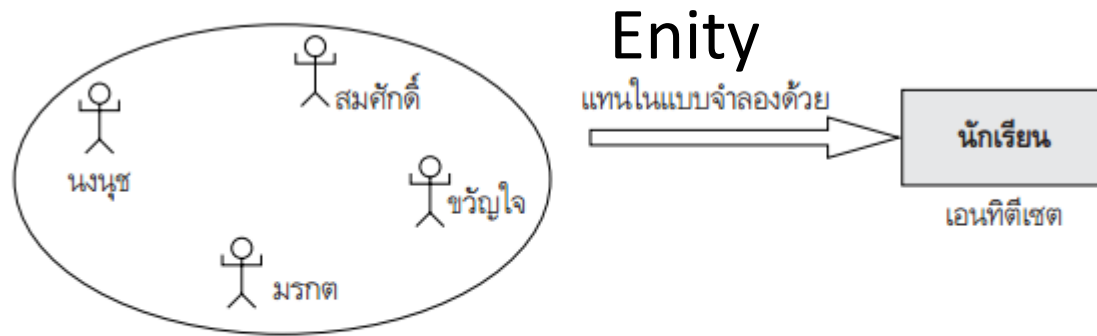
Raster Representation

	0	1	2	3	4	5	6	7	8	9
0								R	T	
1							R			T
2		H					R			
3							R			
4					R	R				
5				R						
6			R		T	T		H		
7			R		T	T				
8		R								
9		R								

Vector Representation

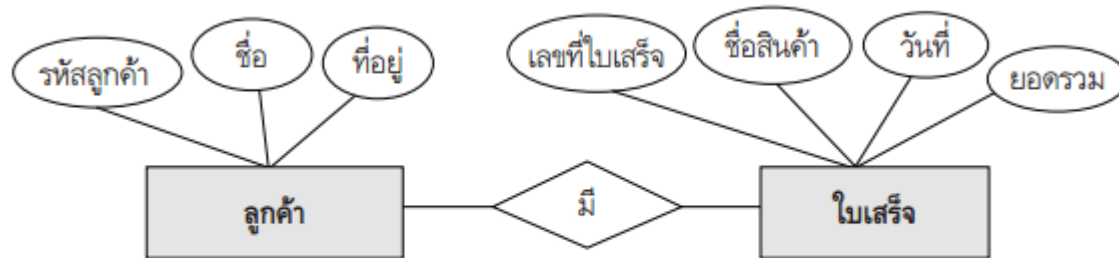


แบบจำลอง



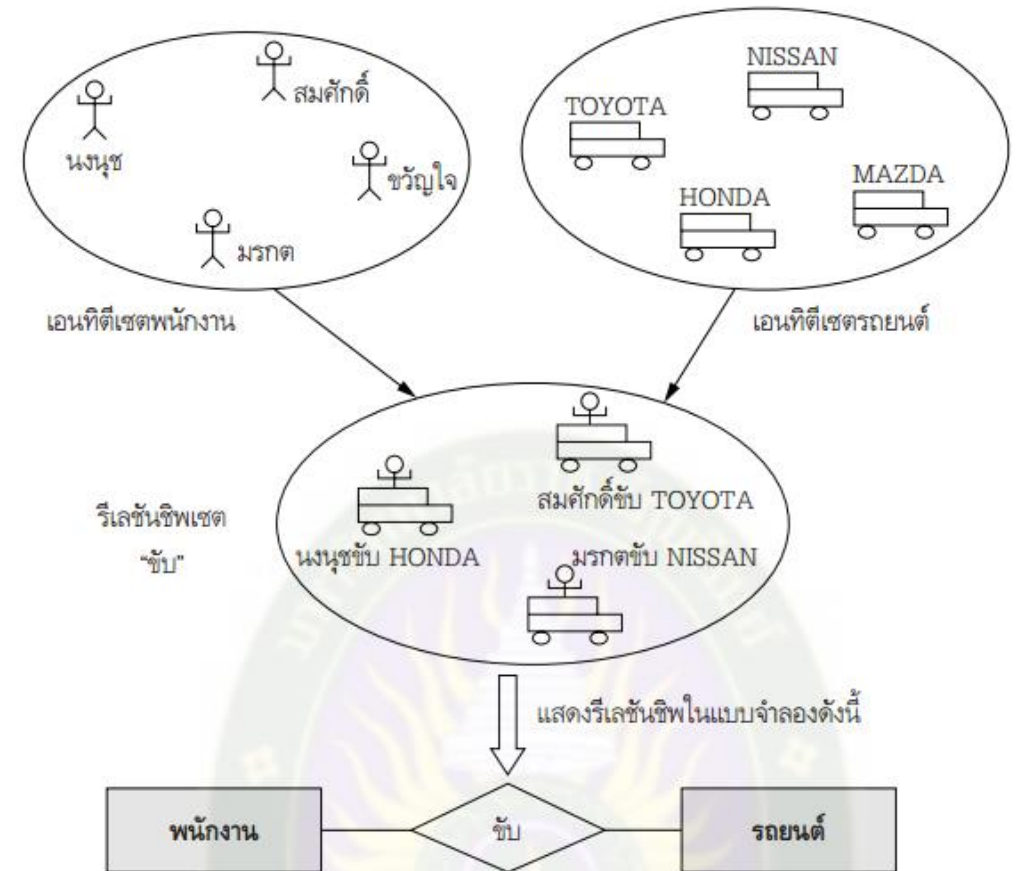
รูปที่ 3.1 การใช้แบบจำลองแทนเอนทิตีเซตนักเรียน

E-R diagram



รูปที่ 3.3 ตัวอย่างของแผนผังอี-อาร์

Relationship



รูปที่ 3.2 การใช้แบบจำลองแทนรีเลชันชิพเซต

ชนิดความสัมพันธ์



One to Many



รูปที่ 3.4 ตัวอย่างการแสดงความสัมพันธ์แบบวัน-ทู-แมนนี่

ข้อมูลลูกค้า

รหัสลูกค้า	ชื่อ	จังหวัด
001	นายแดง นอกรุงเทพ	กรุงเทพฯ
002	นางวัน มากเดือน	สุโขทัย
003	นายสมชาย จิตดีมาก	กรุงเทพฯ
004	นายบุญมี แต่น้อย	นนทบุรี

ข้อมูลใบเสร็จ

รหัสลูกค้า	เลขที่ใบเสร็จ	จำนวนเงิน
001	1234	2,000
001	1270	1,500
002	2001	200
003	1300	23,000
004	1340	800
004	1111	450

รูปที่ 3.5 ตัวอย่างข้อมูลที่มีความสัมพันธ์แบบวัน-ทู-แมนนี่



ชนิดความสัมพันธ์



รูปที่ 3.6 ตัวอย่างการแสดงความสัมพันธ์แบบแมนนี่-ทู-แมนนี่

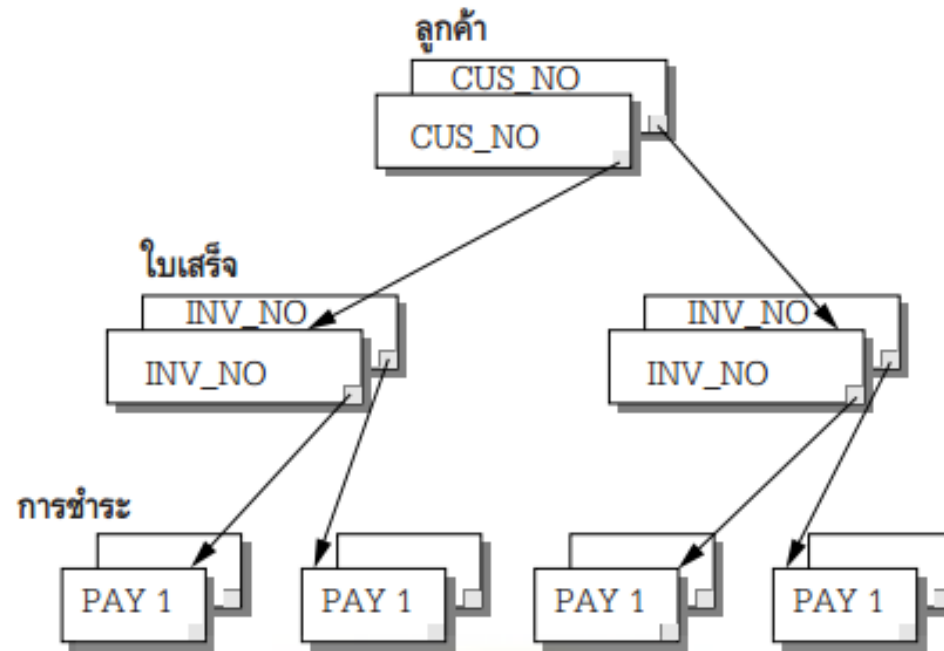
ข้อมูลอาจารย์			ข้อมูลชั้นเรียน		
รหัสอาจารย์	ชื่อ	รหัสชั้นเรียน	รหัสชั้นเรียน	ชื่อวิชา	รหัสอาจารย์
T001	ดร.วิเชียร	C001	C001	ฐานข้อมูล	T001
T001	ดร.วิเชียร	C002	C001	ฐานข้อมูล	T002
T002	ดร.วรพจน์	C001	C001	ฐานข้อมูล	T003
T002	ดร.วรพจน์	C002	C002	MIS	T001
T003	ดร.อาริต	C001	C002	MIS	T002

รูปที่ 3.7 ตัวอย่างข้อมูลที่มีความสัมพันธ์แบบแมนนี่-ทู-แมนนี่



รูปที่ 3.8 ตัวอย่างการแสดงความสัมพันธ์แบบวัน-ทู-วัน

แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Data Model)



รูปที่ 3.9 แบบจำลองฐานข้อมูลแบบลำดับชั้น



แบบจำลองเครือข่าย (Network Data Mode)

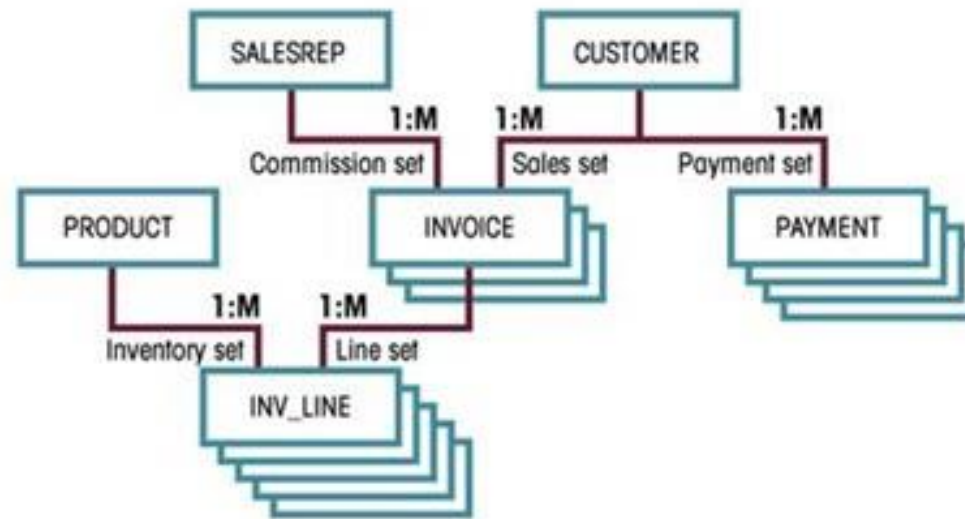
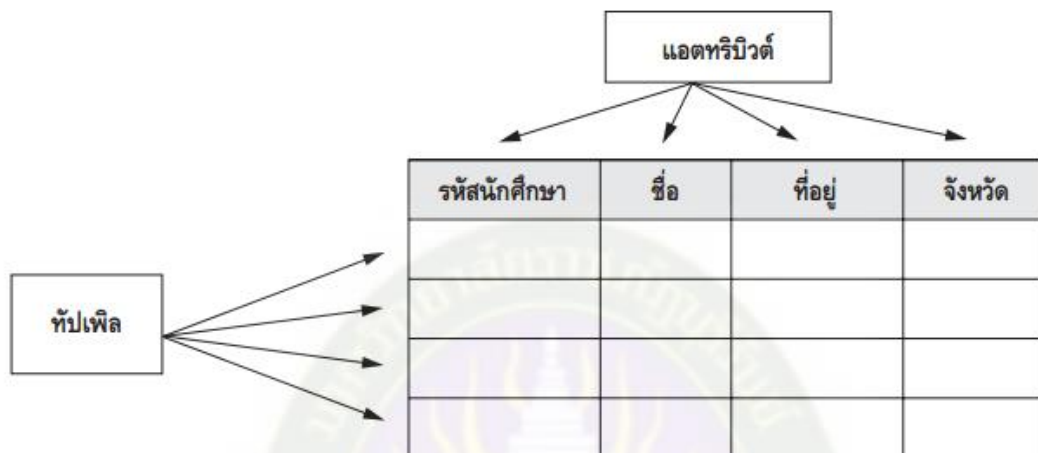


FIGURE 1.10 A NETWORK DATABASE MODEL



แบบจำลองเชิงสัมพันธ์ (Relational Data Model)



รูปที่ 3.11 แบบจำลองข้อมูลเชิงสัมพันธ์

นักศึกษา							
รหัสประจำตัว	ชื่อ	นามสกุล	ที่อยู่	เบอร์โทรศัพท์	วันเกิด	ชั้นปี	เกรดเฉลี่ยสะสม
๑๒๓	สมชาย	รักไทย	๑ หมู่ ๒ มุลเมือง ซอย ๘ จ. เชียงใหม่	๓๘๕-๕๕๕๐	๑ ม.ก. ๒๕๑๑	๔	๓.๔๔
๔๕๖	สมหญิง	รักชาติ	๓ หมู่ ๔ คลองกุ่ม บึงกุ่ม จ. กรุงเทพฯ	๓๓๔-๐๐๐๑	๑๒ มิ.ย. ๒๕๑๔	๑	๓.๘๒
๓๓๔	สมศรี	ใจดี	๕ หมู่ ๖ อ. เมืองพล จ. ขอนแก่น	NULL	๔ พ.ย. ๒๕๑๒	๓	๒.๕๕
๕๘๙	สมศักดิ์	ซื่อสัตย์	๗ หมู่ ๘ อ. หัวหิน จ. ประจวบคีรีขันธ์	๘๑๕-๔๕๐๐	๓๑ ธ.ก. ๒๕๑๔	๑	๒.๕๕

ฐานข้อมูลเชิงสัมพันธ์

GEO3411

การออกแบบฐานข้อมูลสารสนเทศภูมิศาสตร์



ศัพท์และความหมาย

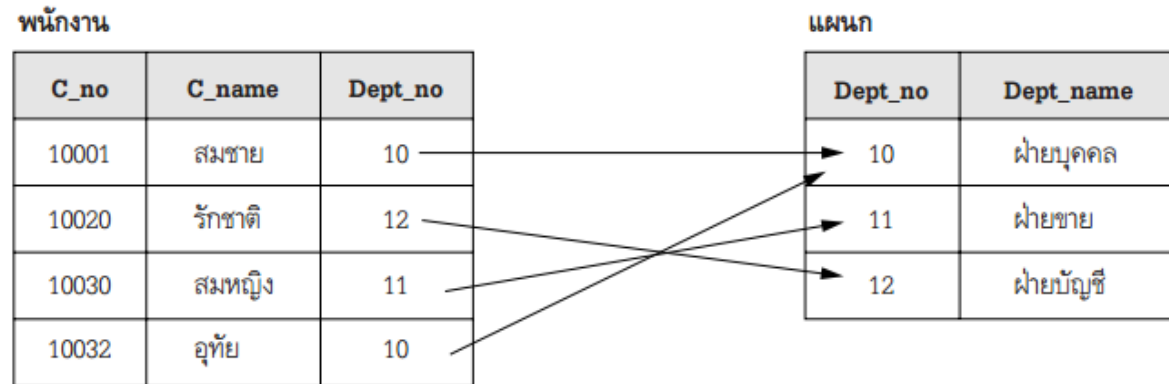


รีเลชัน(Relation)	ตาราง 2 มิติที่ประกอบด้วยแถวและคอลัมน์
แอททริบิวต์ (Attribute)	คอลัมน์ในรีเลชัน
ทัพเพิล(Tuple)	แถวในรีเลชัน
ดีกรี(Degree)	จำนวนหลักแอททริบิวต์ในรีเลชัน
คาร์ดินัลลิตี้ (Cardinality)	จำนวนแถวข้อมูลทั้งหมดที่เป็นไปได้ของแอททริบิวต์หนึ่งๆ
โดเมน(Domain)	การกำหนดขอบเขต-ประเภทของข้อมูล หรือ ค่าของข้อมูลในแอททริบิวต์

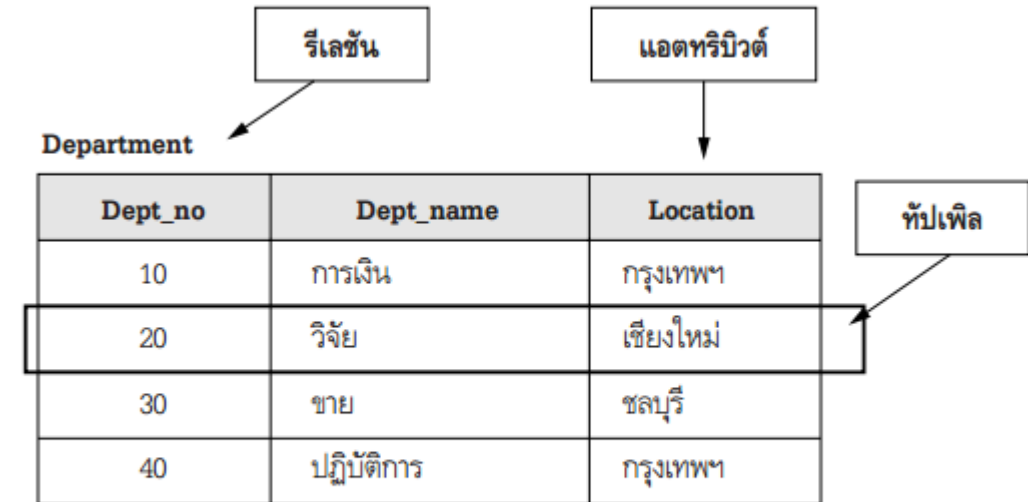
ค่าว่าง(Null Value)	ในแอททริบิวต์ไม่มีค่าของข้อมูลอยู่
คีย์หลัก(Primary Key)	แอททริบิวต์ที่สามารถเจาะจงแถวใดแถวหนึ่งได้โดยที่มีข้อมูลไม่ซ้ำกันเลย
คีย์รวม(Composite Key)	แอททริบิวต์มากกว่าหนึ่งแอททริบิวต์มารวมกันเพื่อให้ไม่มีข้อมูลซ้ำและไม่มีค่าว่าง
คีย์คู่แข่ง(Candidate Key)	เป็นคีย์ที่มีคุณสมบัติครบถ้วนจนนำมาใช้เป็นคีย์หลักแทนกันได้
คีย์นอก(Foreign Key)	แอททริบิวต์ในรีเลชันหนึ่งๆ ที่เชื่อมโยงข้อมูลระหว่างแอททริบิวต์



โครงสร้างฐานข้อมูลเชิงสัมพันธ์



รูปที่ 4.1 แสดงความสัมพันธ์ระหว่างเอนทิตีพนักงานกับแผนกโดยใช้ค่าในคอลัมน์ "Dept_no"



รูปที่ 4.2 ตัวอย่างรีเลชันในฐานข้อมูลเชิงสัมพันธ์



Attribute and Domain



รูปที่ 4.3 โดเมนของแอตทริบิวต์ Location

ตย. เช่น

กลุ่มของข้อมูลทั้งหมดที่เป็นไปได้ของแอตทริบิวต์หนึ่งๆ โดเมนสองโดเมนจะเป็นโดเมนเดียวกันถ้าสองโดเมนนั้นมีความหมายเหมือนกัน ดังนั้นแอตทริบิวต์ชื่อคนงานและแอตทริบิวต์ประเภทความชำนาญจะเป็นแอตทริบิวต์ที่มาจากโดเมนต่างกัน ถึงแม้ว่าแต่ละแอตทริบิวต์จะประกอบด้วยกลุ่มของตัวอักขระเหมือนกัน เนื่องจากโดเมนของชื่อ คนงานและโดเมนของประเภทความชำนาญจะมีความหมายต่างกัน แต่พิจารณาแอตทริบิวต์รหัสผู้ควบคุมและแอตทริบิวต์รหัสคนงาน จะเป็นแอตทริบิวต์ที่มาจากโดเมนเดียวกัน เนื่องจากโดเมนของสองแอตทริบิวต์ประกอบด้วยข้อมูลที่เป็นรหัสของคนงานเหมือนกัน

Flight

Flight_no	Dep_time	Arr_time	Origination	Destination
TG042	0900	1000	BKK	KK
TG092	1200	1350	BKK	CM
TG053	1400	1530	CM	BKK

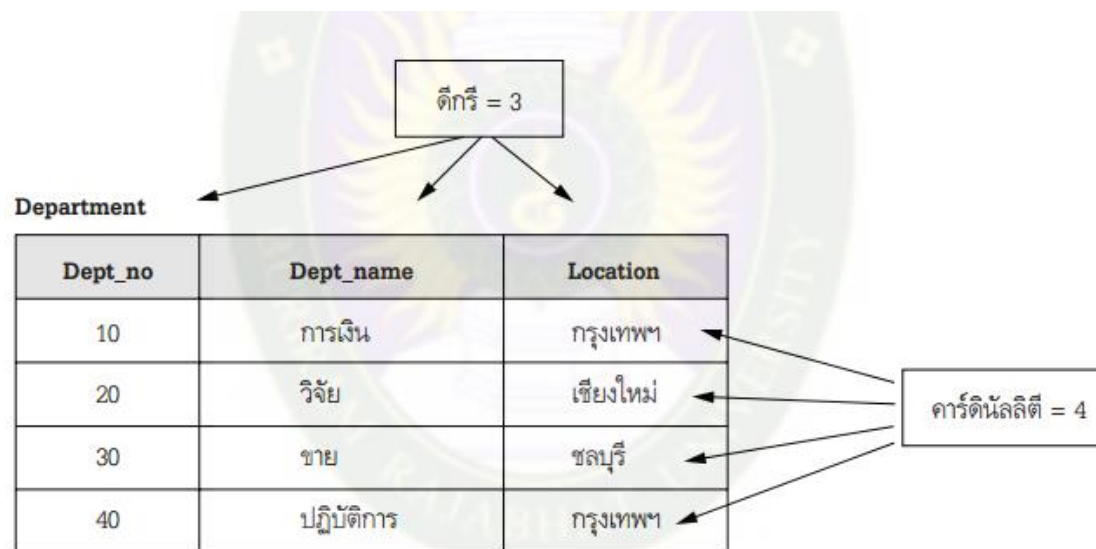
โดเมน Time

Airport

Airpt_code	City
BKK	Bangkok
KK	Khon Kaen
CM	Chiangmai

โดเมน Airport

ดีกรี(degree) และคาร์ดินัลลิตี้ (cardinality) ของรีเลชัน (Relation)



รูปที่ 4.6 ดีกรีและคาร์ดินัลลิตี้



ชนิดของคีย์ในฐานะข้อมูลเชิงสัมพันธ์

- คีย์ (key)

แอตทริบิวต์ ที่สามารถใช้ในการบ่งบอกความแตกต่างของแต่ละทUPLE ในรีเลชัน แอตทริบิวต์ที่เป็นส่วนหนึ่งของคีย์เรียกว่า คีย์แอตทริบิวต์ และคีย์แอตทริบิวต์ที่เกิดจากการนำเอาหลายแอตทริบิวต์มารวมกัน เรียกว่าคอมโพสิตคีย์ (**Composite Key**) คีย์ต่าง ๆ ที่ใช้ในฐานข้อมูลมีด้วยกันหลายชนิด



ชนิดของคีย์ในฐานข้อมูลเชิงสัมพันธ์

- **ซูเปอร์คีย์ (Superkey)**

ซูเปอร์คีย์ คือ แอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ที่เป็นไปได้ทั้งหมดและสามารถบ่งบอกความแตกต่างของแต่ละทUPLEในรีเลชันได้ ดังนั้นในหนึ่งรีเลชันสามารถมีซูเปอร์คีย์ได้หลายซูเปอร์คีย์

นักศึกษา(รหัสนักศึกษา, ชื่อ, ที่อยู่, วันเดือนปีเกิด, เลขที่บัตรประชาชน)

รีเลชันนักศึกษา มีซูเปอร์คีย์ได้หลายตัว เช่น

รหัสนักศึกษา

รหัสนักศึกษา, ชื่อ

รหัสนักศึกษา, ชื่อ, ที่อยู่

เลขที่บัตรประชาชน



ชนิดของคีย์ในฐานะข้อมูลเชิงสัมพันธ์

- แคนดิเดตคีย์ (Candidate Key)

แคนดิเดตคีย์ คือ ซูเปอร์คีย์ที่มีขนาดเล็กที่สุด ที่ยังสามารถบ่งบอกความแตกต่างของแต่ละทUPLE ในรีเลชันได้ หรือกล่าวอีกอย่างหนึ่งก็คือซูเปอร์คีย์ที่ไม่มีสับเซตของตนเองเป็นซูเปอร์คีย์นั่นเอง ในหนึ่งรีเลชัน อาจจะมีแคนดิเดตคีย์ได้หลายตัว

รหัสนักศึกษา และเลขที่บัตรประชาชน

ส่วนกลุ่มของแอตทริบิวต์ (รหัสนักศึกษา, ชื่อ) และ (รหัสนักศึกษา, ชื่อ, ที่อยู่) ไม่เป็นแคนดิเดตคีย์

เนื่องจากมีสับเซตเป็นแคนดิเดตคีย์ คือ รหัสนักศึกษาเป็นซูเปอร์คีย์นั่นเอง



ชนิดของคีย์ในฐานะข้อมูลเชิงสัมพันธ์

- คีย์หลัก (**Primary Key**)

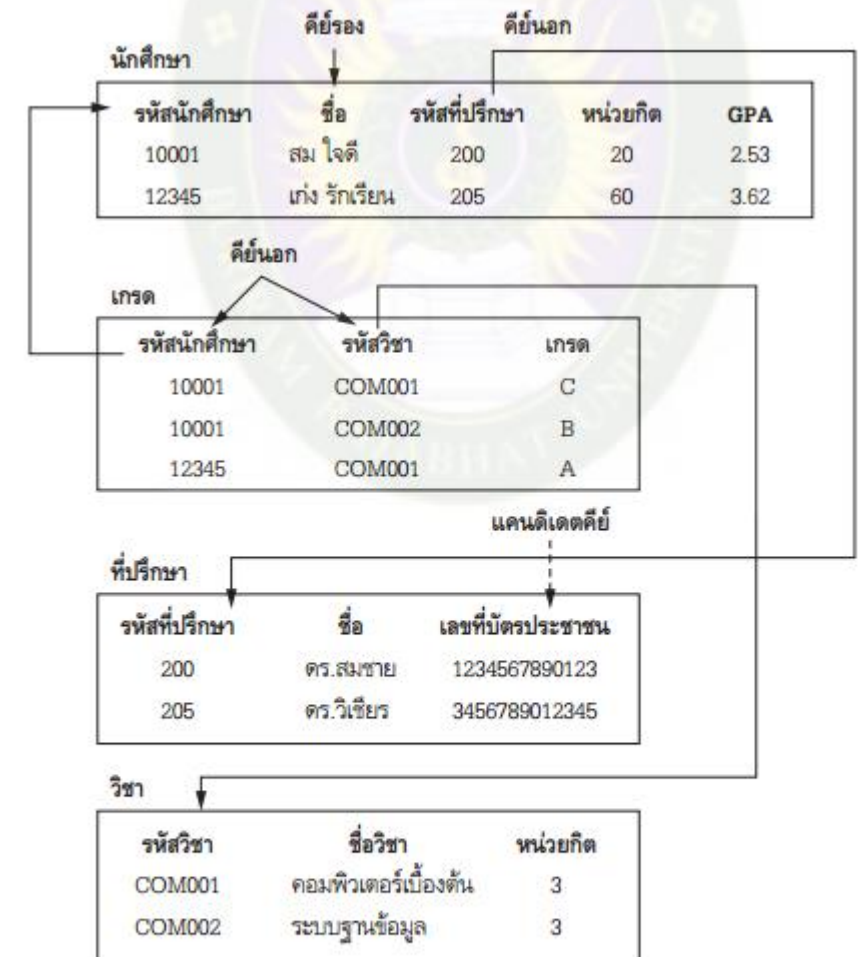
คีย์แคนดิเดตคีย์ที่ถูกเลือกเพื่อใช้บอกความแตกต่างของแต่ละทUPLEในรีเลชัน และต้องไม่มีค่าเป็นค่าว่าง (Null)

รหัสนักศึกษาและ เลขที่บัตรประชาชน

ชนิดของคีย์ในฐานข้อมูลเชิงสัมพันธ์

- คีย์นอก (foreign key)

คือ แอตทริบิวต์ที่ใช้ในการเชื่อมต่อกับรีเลชันอื่นหรือกับตัวมันเองเพื่อแสดงความสัมพันธ์ระหว่างรีเลชันโดยคีย์นอกสามารถมีค่าซ้ำกันหรือ เป็น **Null** ได้ และถ้าไม่เป็น **Null** จะเป็นค่าที่ใช้ชี้ไปที่คีย์หลักของรีเลชันที่มันมีความสัมพันธ์อยู่



รูปที่ 4.7 ตัวอย่างคีย์ต่าง ๆ ที่ใช้ในฐานข้อมูลเชิงสัมพันธ์



โอเปอเรเตอร์ฐานข้อมูลเชิงสัมพันธ์

โอเปอเรเตอร์ในการจัดการข้อมูล

INSERT
DELETE
UPDATE

โอเปอเรเตอร์พื้นฐาน

SELECTION
PROJECTION
JOIN
DIVISION
UNION
INTERSECTION
DIFFERENCE
CARTESIAN PRODUCT

รูปที่ 4.8 โอเปอเรเตอร์ในฐานข้อมูลเชิงสัมพันธ์

INSERT



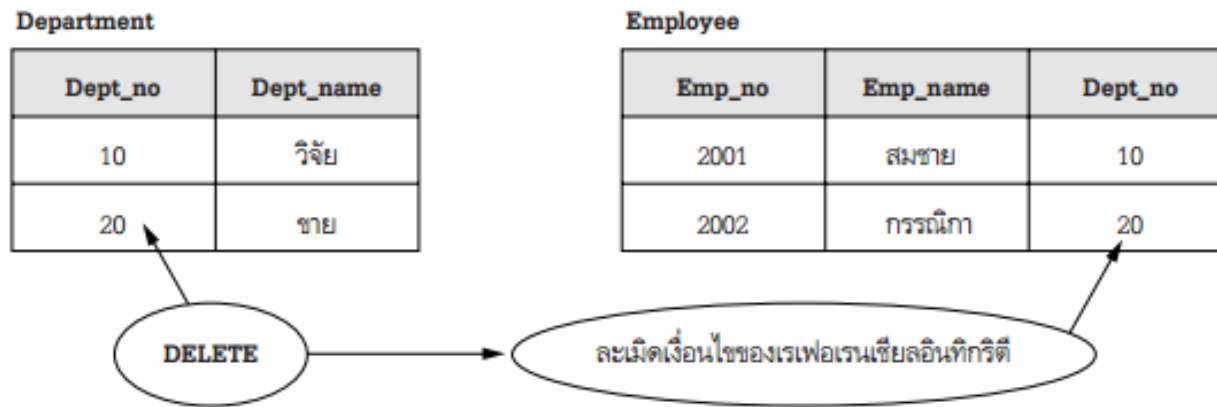
Department

Dept_no	Dept_name	Location
10	วิจัย	กรุงเทพฯ
20	ขาย	ภูเก็ต
30	การเงิน	เชียงใหม่
40	วางแผน	ลำปาง

INSERT (40, 'วางแผน', 'ลำปาง') into Department

รูปที่ 4.9 การใช้ INSERT โอเปอเรเตอร์

DELETE



รูปที่ 4.10 การใช้ DELETE โอเปอเรเตอร์



UPDATE

- เป็นโอเปอเรเตอร์ที่ใช้ในการเปลี่ยนแปลงแก้ไขข้อมูลในทปเปิลของรีเลชัน ในการใช้ **update** โอเปอเรเตอร์อาจเกิดปัญหาขึ้นได้ถ้าเป็นการแก้ไขในแอตทริบิวต์ที่เป็นหลัก หรือ คีย์นอก ทั้งนี้การใช้ **Update** โอเปอเรเตอร์เพื่อแก้ไขคีย์หลักจะมีผลเท่ากับ การใช้ **DELETE** โอเปอเรเตอร์ แล้วตามด้วย **INSERT** โอเปอเรเตอร์นั่นเอง ส่วนการใช้ **UPDATE** โอเปอเรเตอร์เพื่อแก้ไขคีย์นอกจะเป็นไปอย่างถูกต้องก็ต่อเมื่อค่าใหม่ที่ใส่ลงไปมีความสอดคล้องกับค่าที่มีอยู่ในรีเลชันที่มันไปอ้างอิง



SELECTION

$\sigma_{\langle \text{เงื่อนไข} \rangle}$ ($\langle$ ชื่อรีเลชัน $\rangle$)

เงื่อนไขที่ใช้จะเป็นนิพจน์ทางบูลีน โดยอาจจะประกอบด้วยโอเปอเรเตอร์ ดังนี้

- (1) โอเปอเรเตอร์ที่ใช้ในการเปรียบเทียบ ได้แก่ =, <>, >, >=, <, <=
- (2) บูลีนโอเปอเรเตอร์ ได้แก่ AND, OR, NOT

EMPLOYEE

Emp_no	Emp_name	Job	Manager	Salary
3001	สมชาย	ผู้จัดการ	3456	15,000
3456	มรกต	กรรมการผู้จัดการ		30,000
4000	เป็นหนึ่ง	พนักงานขาย	3001	8,500

$\sigma_{\text{Salary} > 10,000}$ (EMPLOYEE)

EMPLOYEE

Emp_no	Emp_name	Job	Manager	Salary
3001	สมชาย	ผู้จัดการ	3456	15,000
3456	มรกต	กรรมการผู้จัดการ		30,000

รูปที่ 4.11 การใช้ SELECT โอเปอเรเตอร์

PROJECTION



EMPLOYEE

Emp_no	Emp_name	Job	Manager	Salary
3001	สมชาย	ผู้จัดการ	3456	15,000
3456	มรกต	กรรมการผู้จัดการ		30,000
4000	เป็นหนึ่ง	พนักงานขาย	3001	8,500



$\pi_{\text{Job, Salary}}(\text{EMPLOYEE})$

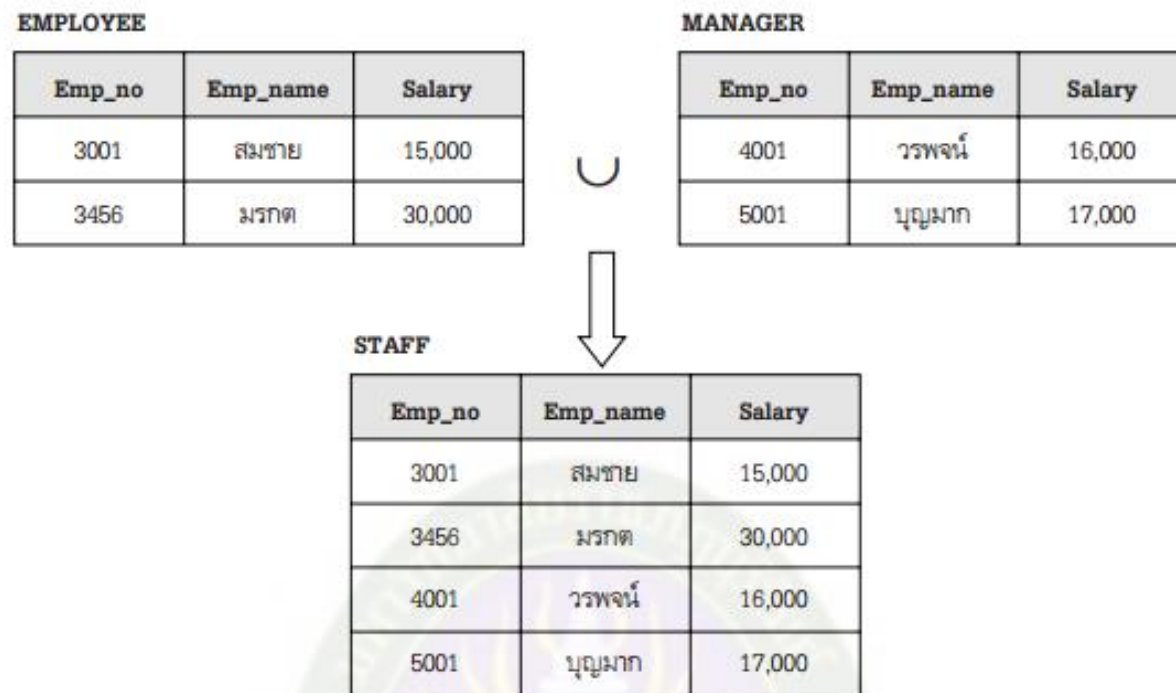


EMPLOYEE

Job	Salary
ผู้จัดการ	15,000
กรรมการผู้จัดการ	30,000
พนักงานขาย	8,500

รูปที่ 4.12 การใช้ PROJECT โอเปอเรเตอร์

UNION



รูปที่ 4.13 การใช้ UNION โอเปอเรเตอร์



INTERSECTION

EMPLOYEE

Emp_no	Emp_name	Salary
3001	สมชาย	15,000
3456	มรกต	30,000
4001	วรพจน์	16,000
5001	บุญมาก	17,000

MANAGER

Emp_no	Emp_name	Salary
4001	วรพจน์	16,000
5001	บุญมาก	17,000



MANAGER-EMPLOYEE

Emp_no	Emp_name	Salary
4001	วรพจน์	16,000
5001	บุญมาก	17,000

รูปที่ 4.14 การใช้ INTERSECTION โอเปอเรเตอร์

DIFFERENCE



EMPLOYEE

Emp_no	Emp_name	Salary
3001	สมชาย	15,000
3456	มรกต	30,000
4001	วรพจน์	16,000
5001	บุญมาก	17,000

MANAGER

Emp_no	Emp_name	Salary
4001	วรพจน์	16,000
5001	บุญมาก	17,000



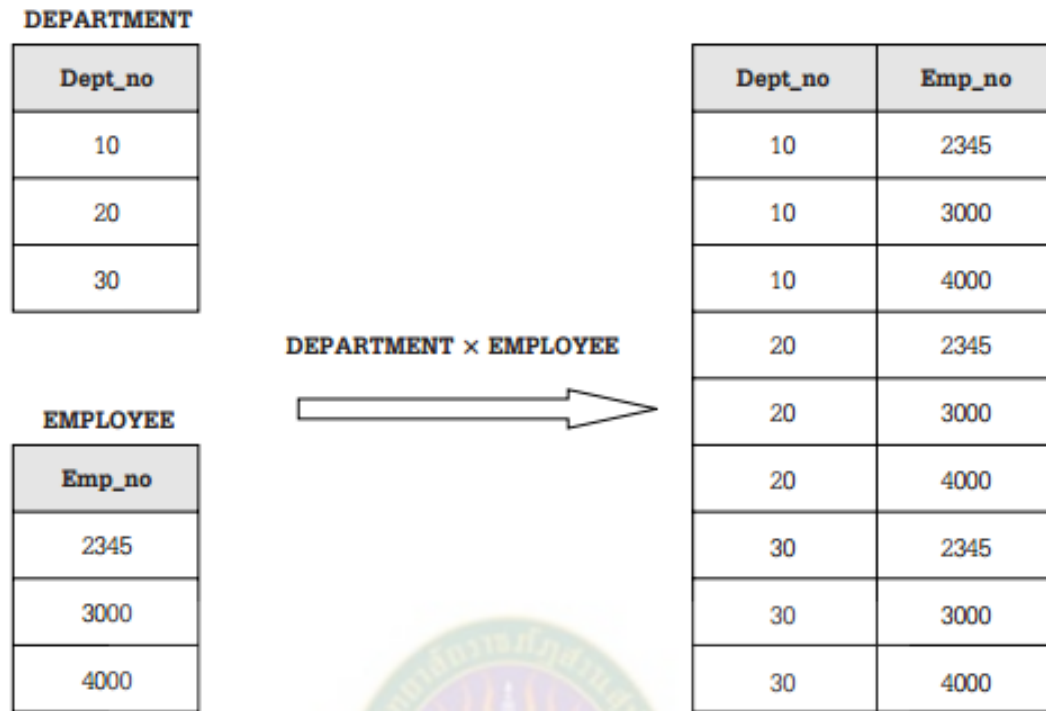
NON-MANAGER-EMPLOYEE

Emp_no	Emp_name	Salary
3001	สมชาย	15,000
3456	มรกต	30,000

รูปที่ 4.15 การใช้ DIFFERENCE โอเปอเรเตอร์



CARTESIAN PRODUCT



รูปที่ 4.16 การใช้ CARTESIAN PRODUCT โอเปอเรเตอร์

JOIN



EMPLOYEE

Emp_no	Emp_name	Emp_loc
5001	บุญเสริม	กรุงเทพฯ
5200	วรากร	เชียงใหม่
5234	สุชาติ	กรุงเทพฯ
5326	สมรักษ์	เชียงใหม่
5400	มารวย	ภูเก็ต
5555	มีชัย	เชียงใหม่

DEPARTMENT

Dept_no	Dept_loc
10	ภูเก็ต
20	กรุงเทพฯ
30	เชียงใหม่
40	สงขลา

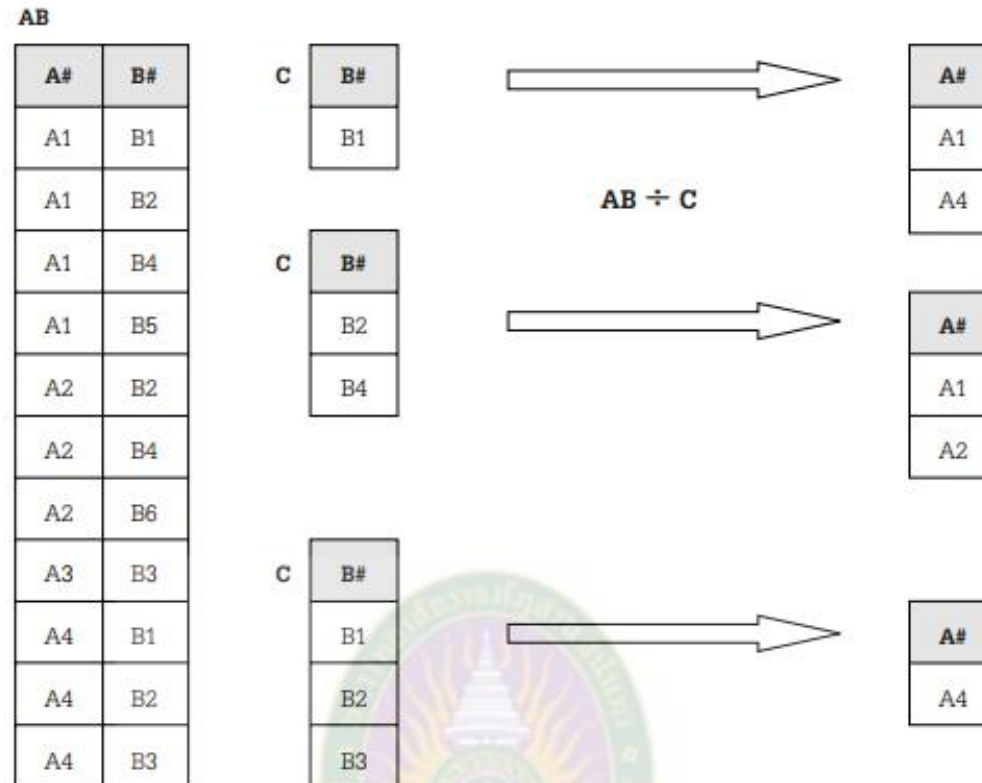
SAME-LOCATION = EMPLOYEE × (Emp_loc = Dept_loc) DEPARTMENT

SAME-LOCATION

Emp_no	Emp_name	Emp_loc	Dept_no	Dept_loc
5001	บุญเสริม	กรุงเทพฯ	20	กรุงเทพฯ
5200	วรากร	เชียงใหม่	30	เชียงใหม่
5234	สุชาติ	กรุงเทพฯ	20	กรุงเทพฯ
5326	สมรักษ์	เชียงใหม่	30	เชียงใหม่
5400	มารวย	ภูเก็ต	10	ภูเก็ต
5555	มีชัย	เชียงใหม่	30	เชียงใหม่

รูปที่ 4.17 การใช้ JOIN โอเปอเรเตอร์

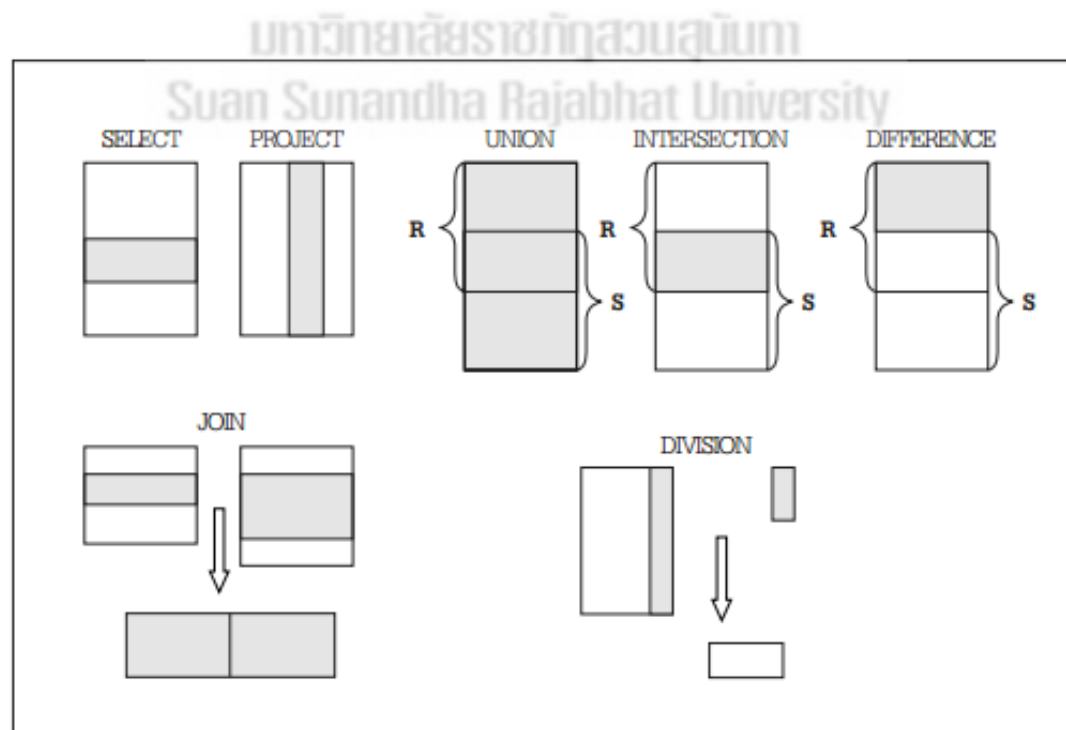
DIVISION



รูปที่ 4.18 การใช้ DIVISION โอเปอเรเตอร์



การทำงาน OPERATER สรุปรูปภาพ



รูปที่ 4.19 การทำงานของโอเปอเรเตอร์ต่าง ๆ ที่สรุปเป็นลักษณะของรูปภาพ



Reference

- วิเชียร เปรมชัยสวัสดิ์. (2546). ระบบฐานข้อมูล. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).
- สุเพชร จิรขจรกุล (2555). เรียนรู้ระบบสารสนเทศภูมิศาสตร์ด้วย โปรแกรม **ArcGIS 10.1 for Desktop**. นนทบุรี: บริษัท เอพี.กราฟิคดีไซน์และการพิมพ์ จำกัด.
- Ronald Briggs(2004). GIS Fundamentals
- สารานุกรมไทยสำหรับเยาวชนฯ / เล่มที่ ๒๕

<http://kanchanapisek.or.th/kp6/sub/book/book.php?book=25&chap=2&page=t25-2-infodetail08.html>
