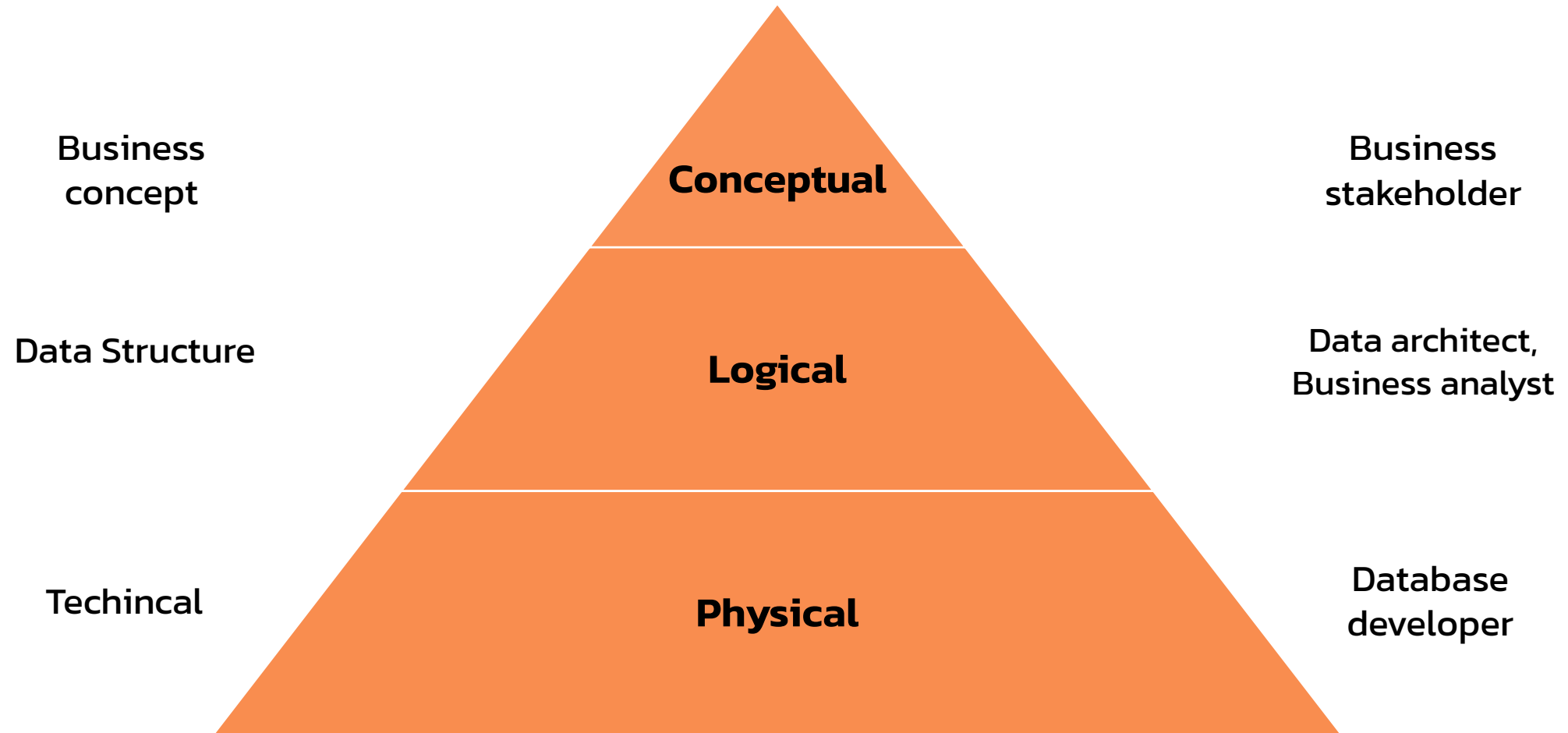


## Chapter 6 :

# การออกแบบฐานข้อมูลเบื้องต้น

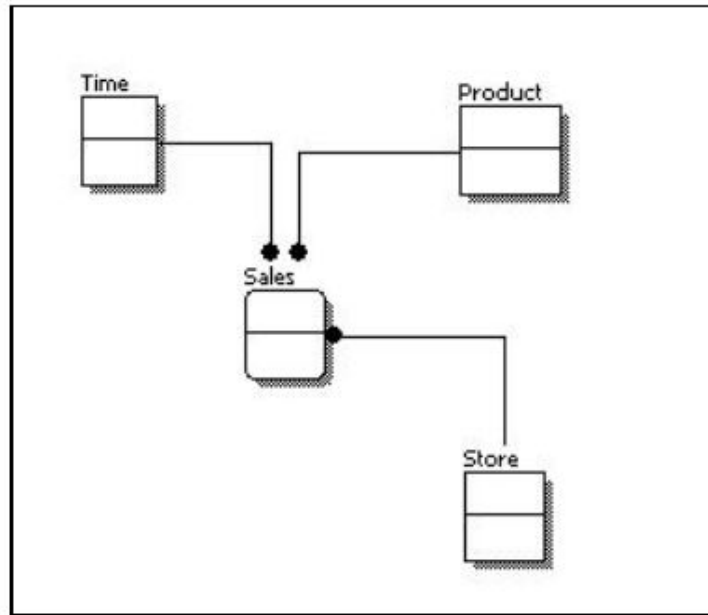
- การออกแบบฐานข้อมูล
- แบบจำลองความสัมพันธ์ระหว่างเอนทิตี (Entity-Relationship Model)
  - เอนทิตี
  - ความสัมพันธ์
  - แอททริบิว
- การจัดการข้อมูลให้อยู่ในรูปแบบบรรทัดฐาน (Normalization)
  - UNF, 1NF, 2NF, and 3NF

# การออกแบบฐานข้อมูล

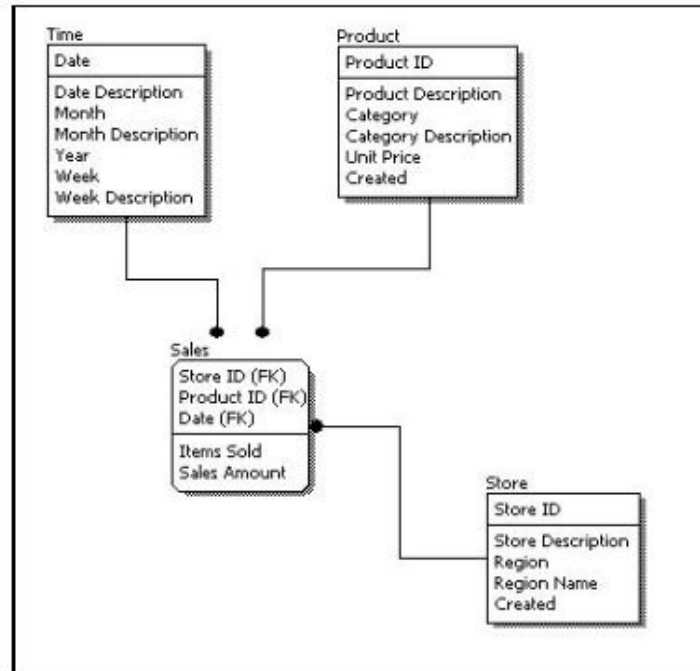


# การออกแบบฐานข้อมูล – ต่อ

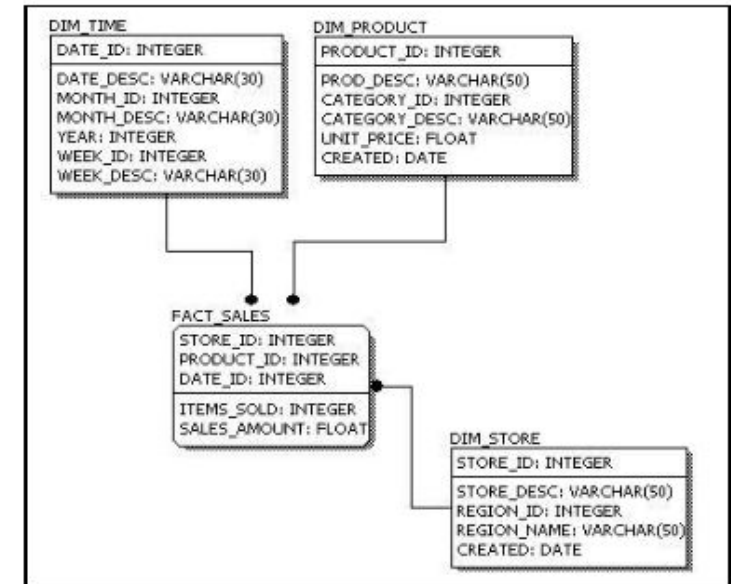
## Conceptual Model Design



## Logical Model Design



## Physical Model Design



# **แบบจำลองความสัมพันธ์ระหว่างเอนทิตี (Entity-Relationship Model)**

# เอนทิตี (Entity)

- **เอนทิตี คือ กลุ่มของวัตถุ (object)** ทั้งที่จับต้องได้ และจับต้องไม่ได้ที่มีสมบัติเหมือนกัน
  - เอนทิตีแบบจับต้องได้ (physical entity) เช่น พนักงาน สินค้า หนังสือ เป็นต้น
  - เอนทิตีแบบจับต้องไม่ได้ (conceptual entity) เช่น การขาย วิชา
- ในการเขียน ER-diagram นิยมใช้สี่เหลี่ยมผืนผ้าแทนเอนทิตี

Employee

Course

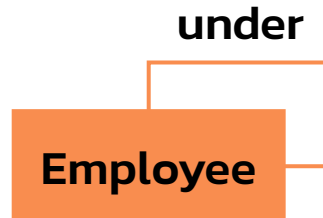
Department

# ความสัมพันธ์ (Relationship)

- ความสัมพันธ์ ใช้ระบุความสัมพันธ์ระหว่างเอนทิตี
- ตัวอย่าง
  - ความสัมพันธ์ระหว่างแผนก (Department) กับพนักงาน (Employee) คือ ประกอบด้วย (has) สามารถเขียนเป็นแผนภาพได้ดังนี้



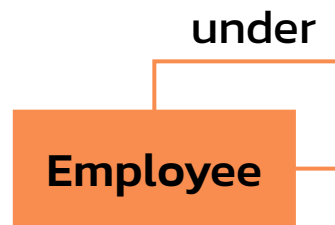
- ความสัมพันธ์ระหว่าง พนักงาน (Employee) กับ พนักงาน (Employee) ภายใต้ความสัมพันธ์ ขึ้นตรงกับ (under) สามารถเขียนเป็นแผนภาพได้ดังนี้



# ความสัมพันธ์ (Relationship) – (ต่อ)

- ดีกรีของความสัมพันธ์ คือ จำนวนเอนทิตีที่สัมพันธ์กันด้วยความสัมพันธ์นั้น

พนักงานคนหนึ่งขึ้น  
ตรงกับพนักงาน



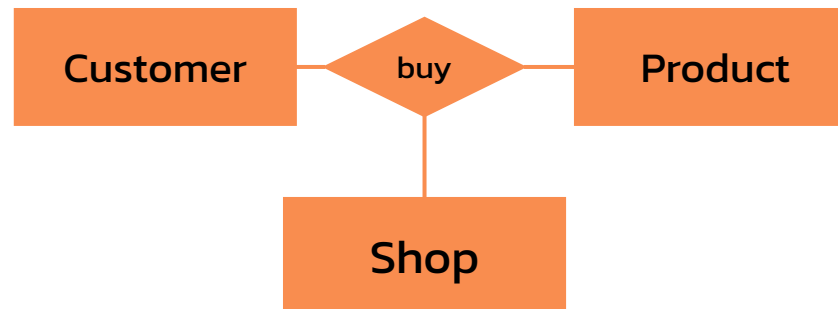
ดีกรี = 1 เรียกว่า unary

แผนก มี  
พนักงาน



ดีกรี = 2 เรียกว่า binary

ลูกค้าซื้อสินค้าหนึ่งจาก  
สาขาแห่งหนึ่ง






ดีกรี = 3 เรียกว่า ternary

# โครงสร้างข้อบังคับ (Structural Constraints)

- โครงสร้างข้อบังคับบนความสัมพันธ์ที่สำคัญคือ multiplicity
- Multiplicity คือ จำนวนความสัมพันธ์ที่เป็นไปได้จากเอนทิตีหนึ่งไปอีกเอนทิตีหนึ่ง
- การระบุ *multiplicity* ขึ้นกับตรรกะเชิงธุรกิจ (business logic)
- ความสัมพันธ์ส่วนใหญ่จะมีดีกรีเป็น 2 นั่นคือ binary relation ซึ่ง multiplicity สามารถแบ่งได้เป็น
  - one-to-one (1:1)
  - one-to-many (1:M)
  - many-to-many (M:N)



# ตัวอย่าง Multiplicity

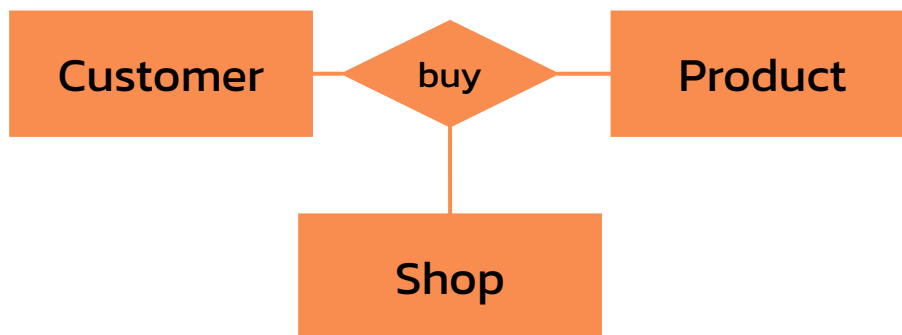
Business Logic	Multiplicity
ลูกค้า 1 คนมีบัญชีได้ 1 บัญชี และ บัญชี 1 บัญชีถือครองด้วยลูกค้า 1 คน	1:1  <pre>graph LR; Customer[Customer] -- 1 -- has -- 1 --&gt; Account[Account]</pre>
ลูกค้า 1 คนมีบัญชีได้หลายบัญชี บัญชี 1 บัญชีถือครองด้วยลูกค้า 1 คน	1:M  <pre>graph LR; Customer[Customer] -- 1 -- has -- M --&gt; Account[Account]</pre>
ลูกค้า 1 คนมีบัญชีได้หลายบัญชี บัญชี 1 บัญชีถือครองด้วยลูกค้าหลายคน	M:N  <pre>graph LR; Customer[Customer] -- M -- has -- N --&gt; Account[Account]</pre>

# แอททริบิวต์ (Attribute)

- แอททริบิวต์ (Attribute) หมายถึง ลักษณะหรือคุณสมบัติที่นำมาอธิบายเอนทิตี และ ความสัมพันธ์

- ตัวอย่าง

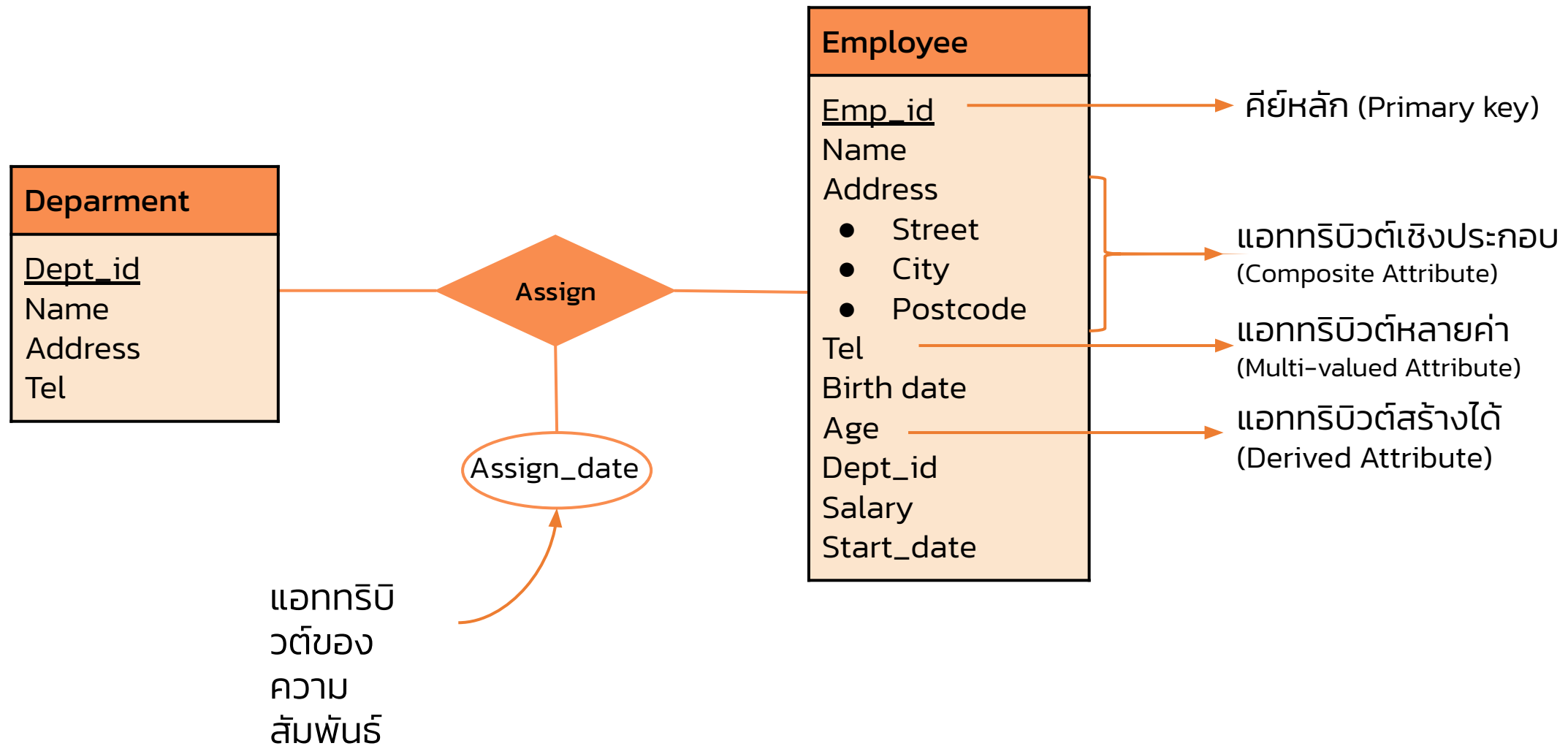
- แอททริบิวต์ สำหรับเอนทิตี Employee เช่น รหัสพนักงาน ชื่อ-นามสกุล แผนก วันบรรจุ เป็นต้น
- แอททริบิวต์ สำหรับความสัมพันธ์ 'buy' เช่น วันที่ซื้อ จำนวนที่ซื้อ ราคาต่อหน่วย เป็นต้น



## แอททริบิวต์ (Attribute) – ต่อ

- แอททริบิวต์อย่างง่าย (Simple Attribute) คือแอททริบิวต์ที่ไม่สามารถแยกข้อมูลออกเป็นข้อมูลย่อย ๆ ได้อีก
- แอททริบิวต์เชิงประกอบ (Composite Attribute) คือแอททริบิวต์ที่สามารถแยกข้อมูลออกเป็นข้อมูลย่อย ๆ ได้
- แอททริบิวต์ค่าเดียว (Single-valued Attribute) คือแอททริบิวต์ที่มีค่า 1 ค่า
- แอททริบิวต์หลายค่า (Multi-valued Attribute) คือแอททริบิวต์ที่มีค่าได้มากกว่า 1 ค่า
- แอททริบิวต์สร้างได้ (Derived Attribute) คือแอททริบิวต์ที่ค่าของแอททริบิวต์สามารถได้มาจากแอททริบิวต์อื่น

# ตัวอย่าง แอททริบิวต์



# **การจัดการข้อมูลให้อยู่ในรูปแบบ บรรทัดฐาน (Normalization)**



# ความซ้ำซ้อนของข้อมูล และการผิดในการปรับปรุงข้อมูล

- ตารางที่มีข้อมูลซ้ำซ้อน สามารถนำไปสู่การผิดพลาดในการปรับปรุงข้อมูล
- การผิดพลาดในการปรับปรุงข้อมูล แบ่งได้เป็น
  - Modification
  - Deletion
  - Insertion

# ความซ้ำซ้อนของข้อมูล และการผิดในการปรับปรุงข้อมูล - ต่อ

## Staff Branch

StaffNo	sName	Position	Salary	Branch No	bAddress
SL2	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assitanet	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Agyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

# ความซ้ำซ้อนของข้อมูล และการผิดในการปรับปรุงข้อมูล - ต่อ

## Staff Branch

StaffNo	sName	Position	Salary	Branch No	bAddress
SL2	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assitanet	12000	<b>B003</b>	<b>163 Main St, Glasgow</b>
SG14	David Ford	Supervisor	18000	<b>B003</b>	<b>163 Main St, Glasgow</b>
SA9	Mary Howe	Assistant	9000	B007	16 Agyll St, Aberdeen
SG5	Susan Brand	Manager	24000	<b>B003</b>	<b>163 Main St, Glasgow</b>
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

*Anomaly of modification*

หากต้องการแก้ไขที่อยู่ของสาขา B003 จะต้องทำอย่างไร?



# ความซ้ำซ้อนของข้อมูล และการผิดในการปรับปรุงข้อมูล - ต่อ

## Staff Branch

StaffNo	sName	Position	Salary	Branch No	bAddress
SL2	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assitanet	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Agyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

*Anomaly of deletion*

หากต้องการลบสาขา B007  
จะเกิดอะไรขึ้น?

# ความซ้ำซ้อนของข้อมูล และการผิดในการปรับปรุงข้อมูล - ต่อ

## Staff Branch

StaffNo	sName	Position	Salary	Branch No	bAddress
SL2	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assitanet	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Agyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London
				<b>B009</b>	<b>123 Main St, Glasgow</b>

*Anomaly of insertion*

หากต้องการเพิ่มสาขา B009  
โดยยังไม่มีพนักงาน จะเกิดอะไรขึ้น?

# ความซ้ำซ้อนของข้อมูล และการผิดในการปรับปรุงข้อมูล - ต่อ

## Staff Branch

StaffNo	sName	Position	Salary	BranchNo	bAddress
SL2	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assitanet	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Agyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

## Staff

StaffNo	sName	Position	Salary	BranchNo
SL2	John White	Manager	30000	B005
SG37	Ann Beech	Assitanet	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

## Branch

BranchNo	bAddress
B005	22 Deer Rd, London
B003	163 Main St, Glasgow
B007	16 Agyll St, Aberdeen

# การขึ้นต่อกัน (Functional Dependency)

- Functional dependency เป็นเครื่องมือสำคัญในการทำรูปแบบบรรทัดฐาน
- Functional dependency จะบอกถึงความสัมพันธ์ระหว่างแอททริบิวต์
- ตัวอย่างเช่น สมมติ A และ B เป็น แอททริบิวต์ของตาราง R, B **ขึ้นต่อกันกับ** A (เขียนแทนด้วย  $A \rightarrow B$ ), ถ้าแต่ละค่าของ A ในตาราง R สัมพันธ์กับค่าของ B ในตาราง R เพียงค่าเดียวเท่านั้น
  - ถ้าทราบค่าของ A แล้วจะทราบค่าของ B
  - แต่ถ้าทราบค่าของ B อาจไม่จำเป็นต้องทราบค่าของ A

# ตัวอย่าง Functional Dependency

เดือน → จำนวนวัน

รหัสไปรษณีย์ → จังหวัด

รหัสพนักงาน → ตำแหน่ง

ชื่อ, นามสกุล → ตำแหน่ง

รหัสพนักงาน → เบอร์โทรศัพท์

อาจไม่จริงในทางกลับกัน

อาจเป็นจริงในทางกลับกัน ภายใต้เงื่อนไข

พนักงานต้องลงทะเบียนเบอร์โทรศัพท์ 1 หมายเลขเท่านั้น

# ลักษณะของ Functional Dependency

- Full functional dependency คือ กรณีที่จำนวนแอททริบิวต์ทางซ้ายของ FD น้อยที่สุด ที่ยังคงระบุค่าแอททริบิวต์ทางขวาของ FD ได้

- ตัวอย่าง

ชื่อ, นามสกุล → เบอร์โทรศัพท์	Full functional dependency
ชื่อ, นามสกุล, <b>วันเดือนปีเกิด</b> → เบอร์โทรศัพท์	ไม่เป็น Full functional dependency

- Partial functional dependency คือ กรณีที่จำนวนแอททริบิวต์ที่มีใช้คีย์คู่แข่ง (candidate key) ขึ้นกับบางแอททริบิวต์ของคีย์คู่แข่ง

- ตัวอย่าง สมมติ เลขใบเสร็จ, รหัสสินค้า เป็นคีย์คู่แข่ง

เลขใบเสร็จ, รหัสสินค้า → ราคาต่อหน่วย	}	ราคาต่อหน่วยเป็น partial functional dependency ต่อเลขใบเสร็จ, รหัสสินค้า
รหัสสินค้า → ราคาต่อหน่วย		

# ลักษณะของ Functional Dependency - ต่อ

- การขึ้นต่อแบบถ่ายทอด (Transitive Dependency) ถ้า A, B, และ C เป็นแอททริบิวต์ ซึ่ง

$A \rightarrow B$  และ  $B \rightarrow C$ , แล้ว C จะขึ้นกับ A โดยการถ่ายทอดจาก B

**Staff Branch**

StaffNo	sName	Position	Salary	BranchNo	bAddress
SL2	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assitanet	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Agyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

**staffNo  $\rightarrow$  sName, position, salary, branchNo**

**branchNo  $\rightarrow$  bAddress**

**staffNo  $\rightarrow$  bAddress**

# Identifying Functional Dependencies

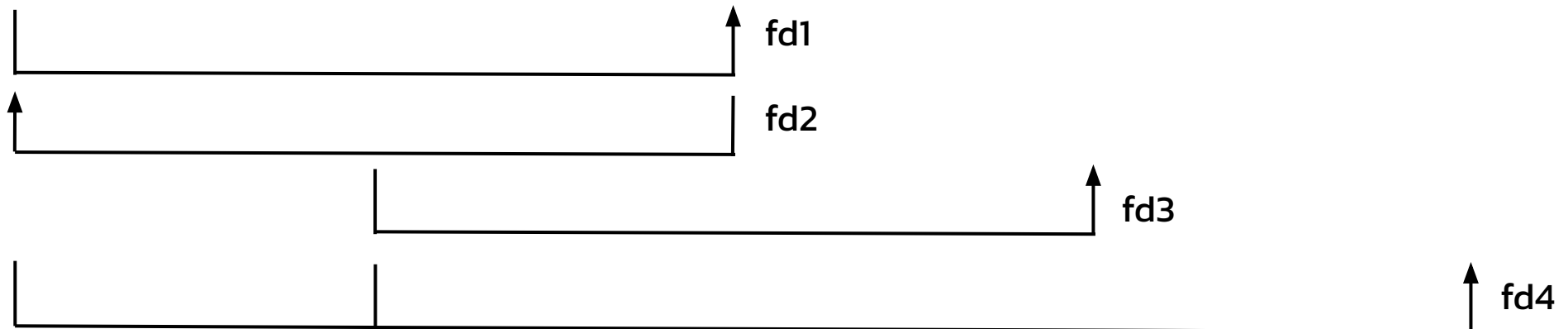
- ระบบ functional dependency ทั้งหมด ซึ่งอาจดูจากข้อมูลร่วมกับปรึกษาผู้ที่เกี่ยวข้อง



# ตัวอย่างการระบุ Functional Dependency

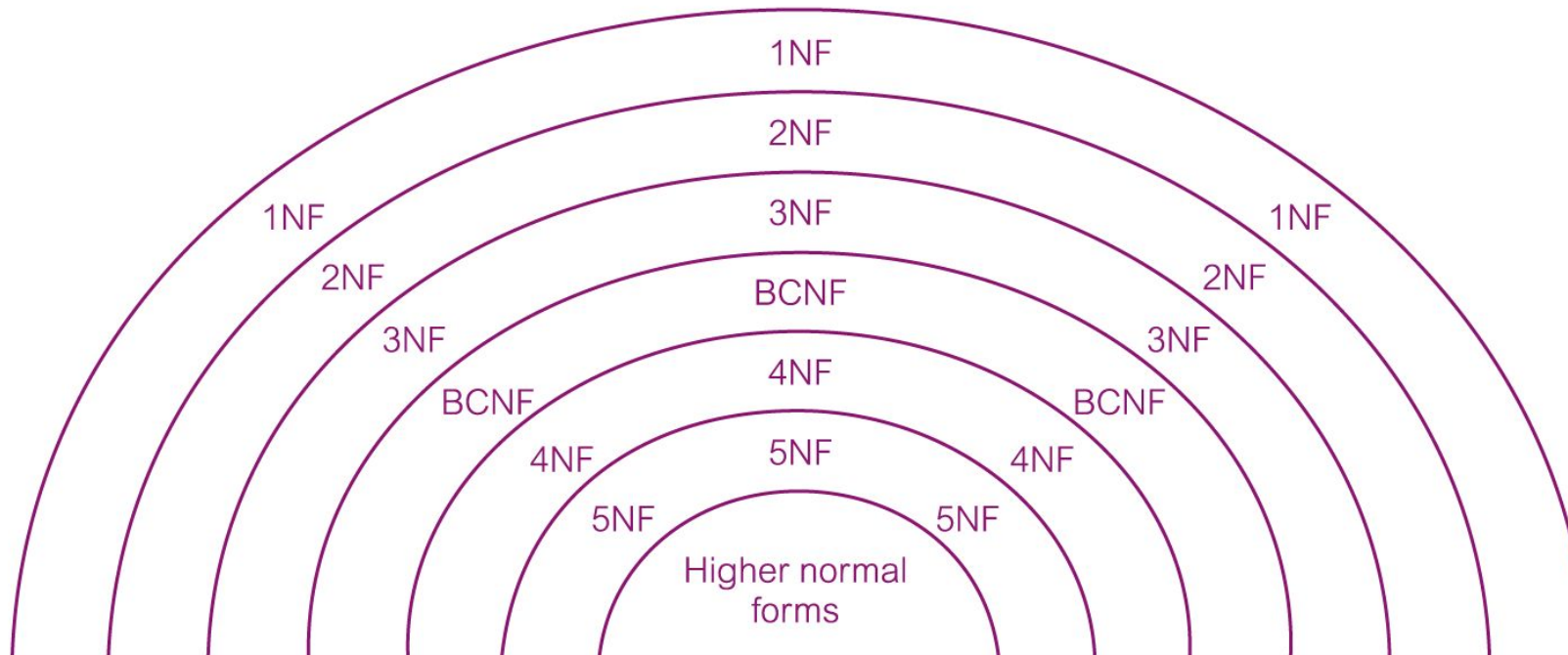
Sample Relation

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t



# ขั้นตอนจัดการข้อมูลให้อยู่ในรูปแบบมาตรฐาน (Normalization)

- ระบุ Functional dependency
- Normalization
  - UNF → 1NF → 2NF → 3NF → BCNF → 4NF → 5NF



# Unnormalized Form (UNF)

- ตารางที่อยู่ในรูป UNF เป็นการเก็บข้อมูลอย่างง่าย อาจมีการซ้ำกันของแถวทริปลิวต์ หรือมีข้อมูลหลายค่าใน 1 ช่อง

Emp_ID	First Name	Surname	Tel1	Tel2
123	Pooja	Singh	555-861-2025	192-122-1111
456	San	Zhang	555-403-1659	182-929-2929
789	John	Doe	555-808-9633	

Emp_ID	First Name	Surname	Tel
123	Pooja	Singh	555-861-2025, 192-122-1111
456	San	Zhang	555-403-1659, 182-929-2929
789	John	Doe	555-808-9633

# First Normal Form (1NF)

- ตารางจะอยู่ในรูปแบบ 1NF ถ้าแต่ละเซลล์มีค่าเพียงหนึ่งค่า

Emp_ID	First Name	Surname	Tel
123	Pooja	Singh	555-861-2025, 192-122-1111
456	San	Zhang	555-403-1659, 182-929-2929
789	John	Doe	555-808-9633

Emp_ID	First Name	Surname	Tel
123	Pooja	Singh	555-861-2025
123	Pooja	Singh	192-122-1111
456	San	Zhang	555-403-1659
456	San	Zhang	182-929-2929
789	John	Doe	555-808-9633

# Second Normal Form (2NF)

- ตารางจะอยู่ในรูป 2NF ถ้า
  - ตารางนั้นอยู่ในรูป 1NF และ
  - ทุกแอททริบิวต์ที่ไม่ใช่คีย์หลักต้องขึ้นกับคีย์หลักอย่างสมบูรณ์ (fully functionally dependent)
- ขั้นตอนแปลงจาก 1NF เป็น 2NF
  - ระบุคีย์หลัก
  - ระบุ FD
  - ถ้ามี FD แบบบางส่วน (partial dependency) กับคีย์หลัก ให้แยก FD นั้นมาสร้างเป็นตารางใหม่

# ตัวอย่างการแปลงจาก 1NF เป็น 2NF

<u>Emp_ID</u>	First Name	Surname	<u>Project_ID</u>	Project_Name
123	Pooja	Singh	P001	CRM
123	Pooja	Singh	P002	New Product
456	San	Zhang	P001	CRM
456	San	Zhang	P003	Cost Saving
789	John	Doe	P003	Cost Saving

- คีย์หลักคือ Emp\_ID , Project\_ID
- Project\_ID -> Project\_Name
- Emp\_ID -> First Name, Surname

<u>Emp_ID</u>	<u>Project_ID</u>
123	P001
123	P002
456	P001
456	P003
789	P003

<u>Emp_ID</u>	First Name	Surname
123	Pooja	Singh
456	San	Zhang
789	John	Doe

<u>Project_ID</u>	Project_Name
P001	CRM
P002	New Product
P003	Cost Saving

# Third Normal Form (3NF)

- ตารางจะอยู่ในรูป 3NF ถ้า
  - ตารางนั้นอยู่ในรูป 2NF และ
  - แอททริบิวต์ที่ไม่ใช่คีย์หลักต้องไม่ขึ้นกับคีย์หลักแบบถ่ายทอด (transitive functionally dependent)
    - ถ้า A เป็นคีย์หลัก โดย  $A \rightarrow B$  และ  $B \rightarrow C$  แล้ว  $A \rightarrow C$
- ขั้นตอนแปลงจาก 2NF เป็น 3NF
  - ถ้ามี FD แบบถ่ายทอด (transitive) กับคีย์หลัก ให้แยก FD นั้นมาสร้างเป็นตารางใหม่

# ตัวอย่างการแปลงจาก 2NF เป็น 3NF

<u>Invoice_ID</u>	<u>Item_ID</u>	Product_ID	Product_Name
210101001	1	P001	Red Pen 0.005
210101001	2	P012	Notebook
210101001	3	P033	Yellow Highlight
210101002	1	P001	Red Pen 0.005
210101003	1	P033	Yellow Highlight

- คีย์หลักคือ Invoice\_ID , Item\_ID
- Product\_ID -> Product\_Name

(Transitive FD)

- Invoice\_ID , Item\_ID -> Product\_Name

<u>Invoice_ID</u>	<u>Item_ID</u>	Product_ID
210101001	1	P001
210101001	2	P012
210101001	3	P033
210101002	1	P001
210101003	1	P033

Product_ID	Product_Name
P001	Red Pen 0.005
P012	Notebook
P033	Yellow Highlight