

4.1

Normalization

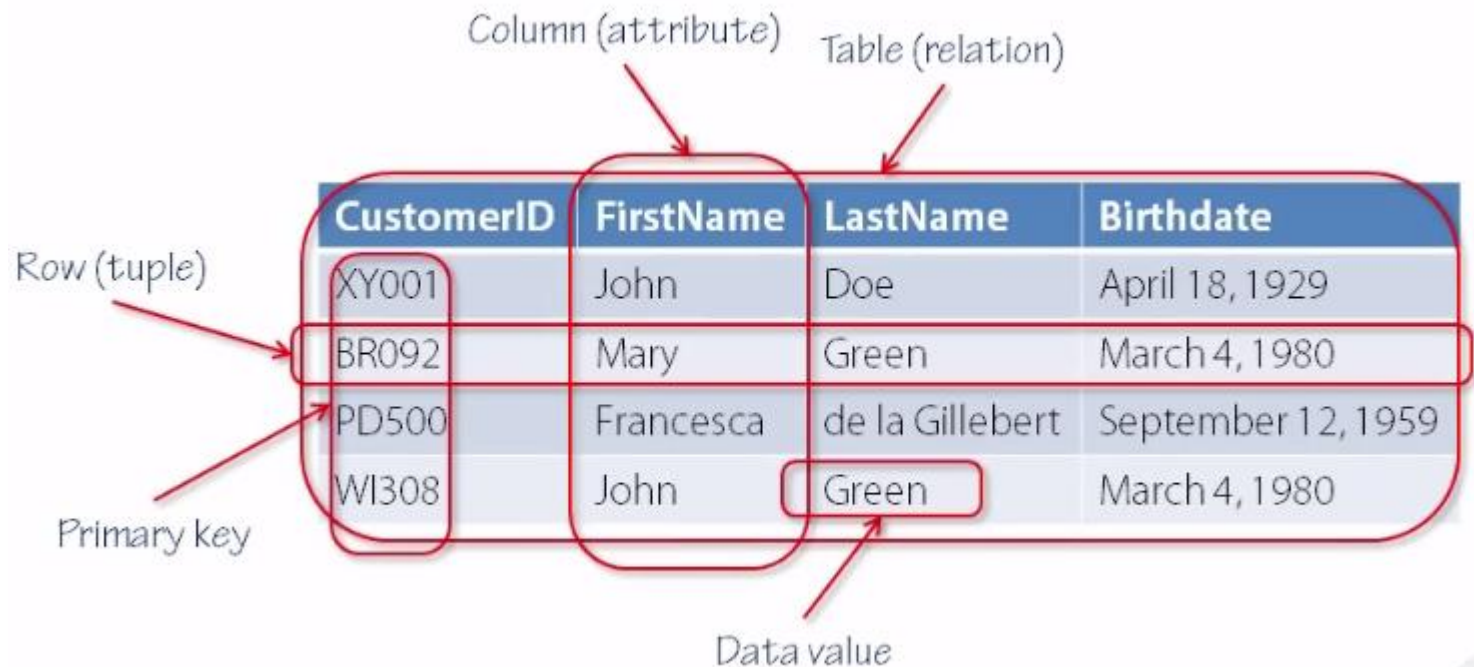
การปรับบรรทัดฐาน

DB304-DB404 การจัดการฐานข้อมูลทางธุรกิจดิจิทัล

อาจารย์เฉลิมพล ศิริกายน และ ดร.ปาณิตา ชูสรานนท์

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

- ▷ Relational database model: consists of **tables**
 - Tuple: record (or row)
 - Attribute: field (or column)
 - Relation: table of records



The Relational Database

- Relational database model: consists of tables
 - Tuple: record (or row)
 - Attribute: field (or column)
 - Relation: table of records
- To design a relational database, you must understand the entities to be stored in the database and how they relate
- Tables are independent of each other, but can be related to each other

Example:

Potential Relational Database for Coca-Cola Bottling Company of Egypt

Order Number: 34562			
Coca-Cola Bottling Company of Egypt Sample Sales Order			
Customer: Dave's Sub Shop		Date: 8/6/2008	
Quantity	Product	Price	Amount
100	Vanilla Coke	\$0.55	\$55
		Distributor Fee	\$29.50
		Order Total	\$67.95

CUSTOMER			
<i>Customer ID</i>	<i>Customer Name</i>	<i>Contact Name</i>	<i>Phone</i>
23	Dave's Sub Shop	David Logan	(555)333-4545
43	Pizza Palace	Debbie Fernandez	(555)345-5432
765	T's Fun Zone	Tom Repicci	(555)565-6655

Master Table

ORDER					
<i>Order ID</i>	<i>Order Date</i>	<i>Customer ID</i>	<i>Distributor ID</i>	<i>Distributor Fee</i>	<i>Total Due</i>
34561	7/4/2008	23	DEN8001	\$22.00	\$145.75
34562	8/6/2008	23	DEN8001	\$12.95	\$67.95
34563	6/5/2008	765	NY9001	\$29.50	\$249.50

Fact Table

ORDER LINE			
<i>Order ID</i>	<i>Line Item</i>	<i>Product ID</i>	<i>Quantity</i>
34561	1	12345AA	75
34561	2	12346BB	50
34563	5	12349EE	100

Fact or
Bridge Table

DISTRIBUTOR	
<i>Distributor ID</i>	<i>Distributor Name</i>
DEN8001	Hawkins Shipping
CHI3001	ABC Trucking

Master Table

PRODUCT		
<i>Product ID</i>	<i>Product Description</i>	<i>Price</i>
12345AA	Coca-Cola	\$0.55
12346BB	Diet Coke	\$0.55

Master Table

กระบวนการ Normalization

- กระบวนการปรับปรุงโครงสร้างข้อมูลของฐานข้อมูลที่มีความซ้ำซ้อน ให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน (Normal Form)
- วัตถุประสงค์ของ Normalization
 - ลดความซ้ำซ้อนของข้อมูล เมื่อลดความซ้ำซ้อนก็ทำให้ลดเนื้อหาที่ใช้ในการจัดเก็บข้อมูล
 - ลดปัญหาความไม่ถูกต้องของข้อมูล เมื่อข้อมูลไม่เกิดความซ้ำซ้อนทำให้การปรับปรุงข้อมูลสามารถทำได้จากแหล่งข้อมูลเพียงแหล่งเดียว
 - ลดความผิดพลาดที่อาจเกิดจากการปรับปรุงข้อมูล ซึ่งประกอบด้วย
 - Update Anomalies
 - Insert Anomalies
 - Delete Anomalies

ตัวอย่าง

รหัสพนักงาน	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา	ที่อยู่
SG21	ชูชาติ สุขศรี	ผู้จัดการ	30000	B005	เชียงใหม่
SG37	ศิริ ดวงเด่น	ผู้ช่วย	20000	B003	กรุงเทพ
SG14	ดวงใจ มีสุข	เลขานุการ	20000	B003	กรุงเทพ
SG09	อัจฉรา เขียวแก้ว	ผู้จัดการ	30000	B007	พิษณุโลก

ความผิดพลาดจากการเพิ่ม

- ถ้าต้องการเพิ่มพนักงานใหม่สาขา B005 จะต้องกรอก 1) B005 และ 2) ที่อยู่สาขา คือ เชียงใหม่
- ถ้าต้องการเพิ่มสาขา จะมีปัญหาคือ ตารางนี้มีทั้งข้อมูลพนักงานและข้อมูลสาขาอยู่รวมกัน ไม่สามารถเพิ่มเฉพาะรหัสสาขาและที่อยู่ได้ เพราะรหัสพนักงานจะมีค่าว่างไม่ได้เนื่องจากเป็น Primary Key ของตาราง ดังนั้นจะบันทึกได้ก็ต่อเมื่อมีพนักงานแล้ว

ตัวอย่าง

รหัสพนักงาน	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา	ที่อยู่
SG21	ชูชาติ สุขศรี	ผู้จัดการ	30000	B005	เชียงใหม่
SG37	ศิริ ดวงเด่น	ผู้ช่วย	20000	B003	กรุงเทพ
SG14	ดวงใจ มีสุข	เลขานุการ	20000	B003	กรุงเทพ
SG09	อัจฉรา เขียวแก้ว	ผู้จัดการ	30000	B007	พิษณุโลก

ความผิดพลาดจากการลบ

- ถ้าลบข้อมูลหนึ่งแล้วส่งผลกระทบต่อข้อมูลอื่น ที่ต้องถูกลบตาม เช่น พนักงานรหัส SG21 ลาออก ก็ลบแถวนั้นออก ดังนั้น ข้อมูลสาขา B005 ก็จะหายไปด้วย

ตัวอย่าง

รหัสพนักงาน	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา	ที่อยู่
SG21	ชูชาติ สุขศรี	ผู้จัดการ	30000	B005	เชียงใหม่
SG37	ศิริ ดวงเด่น	ผู้ช่วย	20000	B003	กรุงเทพ
SG14	ดวงใจ มีสุข	เลขานุการ	20000	B003	กรุงเทพ
SG09	อัจฉรา เขียวแก้ว	ผู้จัดการ	30000	B007	พิษณุโลก

ความผิดพลาดจากการปรับปรุง

- ในกรณีที่ต้องการเปลี่ยนแปลงข้อมูลบางตัวของสาขา เช่น เปลี่ยนที่อยู่ของ B003 ก็ต้องเปลี่ยนหลายจุด ถ้าหากมีพนักงานสังกัดสาขานี้หลายที่ก็ต้องไปตามแก้ทุกๆ จุด

ดังนั้นเราควรแยกตารางออกเป็นสองตาราง คือ 1) ตารางพนักงาน และ 2) ตารางสาขา

พนักงาน

<u>รหัสพนักงาน</u>	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา
SG21	ชูชาติ สุขศรี	ผู้จัดการ	30000	B005
SG37	ศิริ ดวงเด่น	ผู้ช่วย	20000	B003
SG14	ดวงใจ มีสุข	เลขานุการ	20000	B003
SG09	อัจฉรา เขียวแก้ว	ผู้จัดการ	30000	B007

สาขา

<u>รหัสสาขา</u>	ที่อยู่
B005	เชียงใหม่
B003	กรุงเทพ
B007	พิษณุโลก

A structural representation of a relational database

- **Key**: a field whose values identify records -> use to retrieve records


Customer Table			
CustID	CustName	CustPhone	CustAddr
33091	Jim Marasco	322-4907	121 Presidents Rd
35999	Marcia Belvadere	322-5577	32 Parentes Bl
36003	Betsy McCarthy	342-0071	9 Barrington St
36025	Rick Green	322-7299	84 Shopping Pl



Primary key: a field by which records are uniquely identified

- Each record in the table must have a unique key value

Composite key: combination of fields that serve as a primary key



Rental Table				
CustID	BookID	Date Rented	Rental Cost	Date Returned
33091	456-3X	1/15/2013	50	
35999	632-7W	1/19/2013	55	
35999	854-4E	1/19/2013	45	
36003	854-4E	1/4/2013	45	
36025	115-5G	9/3/2012	60	12/19/2012

Composite primary key

Rental Table				
CustID	BookID	Date Rented	Rental Cost	Date Returned
33091	456-3X	1/15/2013	50	
35999	632-7W	1/19/2013	55	
35999	854-4E	1/19/2013	45	
36003	854-4E	1/4/2013	45	
36025	115-5G	9/3/2012	60	12/19/2012

Part of composite key in Rental
relates to the primary key in Book

Book Table			
BookID	Title	PubID	Edition
115-5G	Marketing Research	412	1
747-8T	Investment Strategies	254	4
456-3X	IT Project Management	254	6
632-7W	Database Mgmt Concepts	254	7
854-4E	Business Communication	129	10

Primary key in Publisher relates to the foreign key in Book

Publisher Table		
PubID	Pub Name	City
129	Alpha	Chicago
254	Cengage	Boston
412	Forrest	New York

ขอบเขตเนื้อหาการทำ Normalization สำหรับวิชานี้

- 1NF - กำจัด repeating group
(กำจัดกลุ่มของข้อมูลที่มีความซ้ำซ้อน)
- 2NF - กำจัด partial dependency
(กำจัดการขึ้นต่อกันบางส่วน)
- 3NF - กำจัด transitive dependency
(กำจัดการขึ้นต่อกันของ attribute ที่ไม่ใช่ key หลัก)

First Normal Form (1NF)

- กำจัด repeating group: ทุก attribute ในแต่ละ record จะเป็น single value ไม่มีค่าของกลุ่มข้อมูลที่ซ้ำกัน (Repeating Group)
- ข้อมูลทุกแถว (Tuple) ต้องมีค่าไม่ซ้ำกัน

ตารางที่มีลักษณะข้อมูลเป็น Repeating group

รหัสนักศึกษา	ชื่อ	นามสกุล	คณะ	รหัสวิชาที่ลงทะเบียน	ชื่อวิชา	วันเดือนปีที่ลงทะเบียน
001	ยอดรัก	อบอ่อนใจ	บริหาร	124-100	การจัดการ	2558-5-5
				124-105	การเงิน	2558-5-5
				124-138	พื้นฐานคอมพิวเตอร์	2558-5-5
002	ชาตรี	สง่างาม	ศึกษาศาสตร์	124-103	การศึกษาปฐมวัย	2558-5-2
				124-138	พื้นฐานคอมพิวเตอร์	2558-5-2

แก้ไขเป็น

<u>รหัสนักศึกษา</u>	ชื่อ	นามสกุล	คณะ	รหัสวิชาที่ ลงทะเบียน	ชื่อวิชา	วันเดือนปีที่ ลงทะเบียน
001	ยอดรัก	อบอุ่มใจ	บริหาร	124-100	การจัดการ	2558-5-5
001	ยอดรัก	อบอุ่มใจ	บริหาร	124-105	การเงิน	2558-5-5
001	ยอดรัก	อบอุ่มใจ	บริหาร	124-138	พื้นฐานคอมพิวเตอร์	2558-5-5
002	ชาตรี	สง่างาม	ศึกษาศาสตร์	124-103	การศึกษาปฐมวัย	2558-5-2
002	ชาตรี	สง่างาม	ศึกษาศาสตร์	124-138	พื้นฐานคอมพิวเตอร์	2558-5-2

<u>รหัสนักศึกษา</u>	ชื่อ	นามสกุล	คณะ	<u>รหัสวิชาที่</u> <u>ลงทะเบียน</u>	ชื่อวิชา	วันเดือนปีที่ ลงทะเบียน
001	ยอดรัก	อบอุ่มใจ	บริหาร	124-100	การจัดการ	2558-5-5
001	ยอดรัก	อบอุ่มใจ	บริหาร	124-105	การเงิน	2558-5-5
001	ยอดรัก	อบอุ่มใจ	บริหาร	124-138	พื้นฐานคอมพิวเตอร์	2558-5-5
002	ชาตรี	สง่างาม	ศึกษาศาสตร์	124-103	การศึกษาปฐมวัย	2558-5-2
002	ชาตรี	สง่างาม	ศึกษาศาสตร์	124-138	พื้นฐานคอมพิวเตอร์	2558-5-2

แก้ไขเป็น

นักศึกษา

รหัสนักศึกษา	ชื่อ	นามสกุล	คณะ	สาขา
001	ยอดรัก	อบอุ่มใจ	บริหาร	คอมพิวเตอร์ธุรกิจ
002	ชาตรี	สง่างาม	ศึกษาศาสตร์	เทคโนโลยีการศึกษา

วิชาลงทะเบียน

รหัสนักศึกษา	รหัสวิชาที่ลงทะเบียน	ชื่อวิชา	วันเดือนปีที่ลงทะเบียน
001	124-100	การจัดการ	2558-5-5
001	124-105	การเงิน	2558-5-5
001	124-138	พื้นฐานคอมพิวเตอร์	2558-5-5
002	124-103	การศึกษาปฐมวัย	2558-5-2
002	124-138	พื้นฐานคอมพิวเตอร์	2558-5-2

Second Normal Form (2NF)

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

สรุปก็คือ 2NF เป็นการขจัด attribute ที่ไม่ขึ้นกับทั้งส่วนของ key หลักออกไป เพื่อให้ attribute อื่นทั้งหมดขึ้นตรงกับส่วนที่เป็น key หลักทั้งหมดเท่านั้น

ฟังก์ชันการขึ้นต่อกัน (Functional Dependency : FD)

- ค่าของ attribute หนึ่ง หรือกลุ่มของ attribute ที่ถูกนำมาใช้เป็น key ของ relation นั้น สามารถไประบุค่า attribute อื่นๆ ใน record เดียวกันของ relation นั้นได้
- เช่น attribute Y เป็นฟังก์ชันที่ขึ้นอยู่กับ attribute X เขียนแทนด้วย สัญลักษณ์

$X \rightarrow Y$

<u>EmployeeNo</u>	Name	Position
S01	ฉัตรชัย มีสมบัติ	Manager
S02	เอกชัย ใจดี	Manager Assistant
S03	ขวัญชัย ใจเพชร	Manager Assistant
S04	มานพ เกตุแก้ว	Staff
S05	ดวงกมล ทิพย์เทพ	Staff

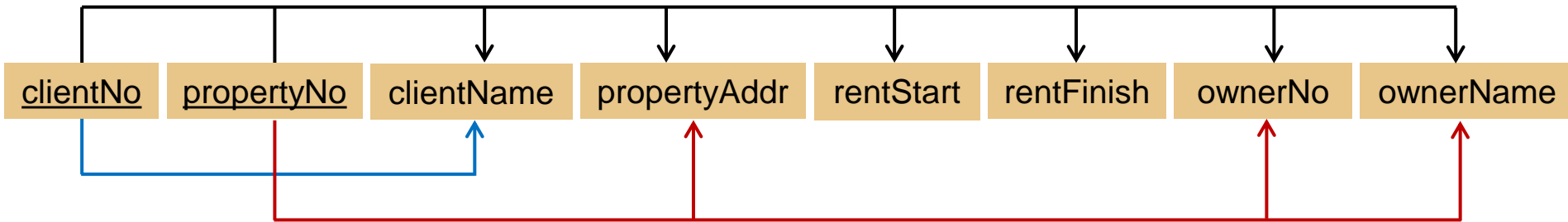
EmployeeNo \rightarrow Position

Partial Dependency

- กรณีความสัมพันธ์แบบบางส่วน สามารถเกิดขึ้นได้เมื่อ key หลักที่ประกอบไปด้วยหลายๆ key รวมกัน (หรือที่เรียกว่า Composite Key) โดยมี attribute บางส่วนของ key หลัก สามารถนำไประบุค่า attribute อื่นๆ ที่ไม่ใช่ key หลักของ relation ได้

Partial Dependency - ตัวอย่างข้อมูลลูกค้าที่เช่าบ้าน

<u>clientNo</u>	<u>propertyNo</u>	clientName	propertyAddr	rentStart	rentFinish	ownerNo	ownerName
CR76	PG04	ยงยุทธ	กรุงเทพฯ	01/07/2557	30/06/2558	CO42	ปาณิตา
CR76	PG16	ยงยุทธ	กรุงเทพฯ	01/09/2558	01/09/2559	CO45	เฉลิมพล
CR56	PG04	สิราณี	กรุงเทพฯ	15/02/2556	15/02/2557	CO42	ปาณิตา
CR56	PG36	สิราณี	กรุงเทพฯ	10/10/2557	01/12/2558	CO45	เฉลิมพล
CR56	PG16	สิราณี	กรุงเทพฯ	01/11/2559	10/08/2560	CO45	เฉลิมพล



clientNo, propertyNo → clientName, propertyAddr, rentStart, rentFinish, ownerNo, ownerName

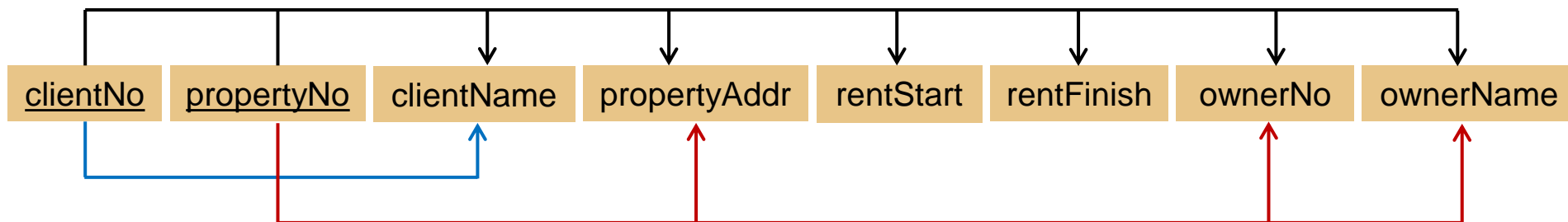
clientNo → clientName

propertyNo → propertyAddr, ownerNo, ownerName } Partial Dependency

Second Normal Form (2NF)

การแก้ปัญหา

- ต้องสร้างตารางเพิ่ม
- นำ attribute ที่มีปัญหาไปใส่ในตารางที่สร้างเพิ่ม
- กำหนด key หลักให้กับตารางที่สร้างใหม่
- Attribute ในตารางเดิมที่ถูกนำไปใส่ในตารางใหม่ ให้ตัดออกจากตารางเดิม ยกเว้นส่วนของ key หลัก ให้คงไว้ในตารางเดิม



Second Normal Form (2NF) - ตัวอย่างข้อมูลลูกค้าที่เช่าบ้าน

CLIENT

<u>clientNo</u>	clientName
CR56	สิราณี
CR76	ยงยุทธ

PROPERTY_OWNER

<u>propertyNo</u>	propertyAddr	ownerNo	ownerName
PG04	กรุงเทพฯ	CO42	ปาณิตา
PG16	กรุงเทพฯ	CO45	เฉลิมพล
PG36	กรุงเทพฯ	CO45	เฉลิมพล

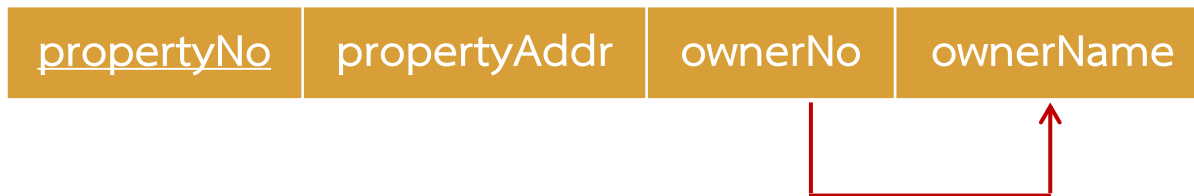
RENTAL

<u>clientNo</u>	<u>propertyNo</u>	rentStart	rentFinish
CR76	PG04	01/07/2557	30/06/2558
CR76	PG16	01/09/2558	01/09/2559
CR56	PG04	15/02/2556	15/02/2557
CR56	PG36	10/10/2557	01/12/2558
CR56	PG16	01/11/2559	10/08/2560

Third Normal Form (3NF)

- Relation นั้นจะต้องมีคุณสมบัติ 2NF
- ต้องไม่มีความสัมพันธ์ระหว่าง Non-key Attribute หรือ ต้องไม่มี Transitive Dependency

สรุป : attribute ที่ไม่ใช่ key หลัก ต้องไม่ขึ้นต่อกันเอง



Third Normal Form (3NF)

การแก้ปัญหา

- 1) สร้างตารางเพิ่ม
- 2) นำแอททริบิวต์ที่มีปัญหามาใส่ในตารางใหม่
- 3) กำหนดคีย์หลัก
- 4) แอททริบิวต์ที่ย้ายจากตารางเดิมไปใส่ในตารางใหม่ให้ตัดออกจากตารางเดิม
- 5) นำคีย์หลักในข้อ 3 ไปใส่ในตารางเดิม

Third Normal Form (3NF) - ตัวอย่างข้อมูลลูกค้าที่เช่าบ้าน

CLIENT

<u>clientNo</u>	clientName
CR56	สิราณี
CR76	ยงยุทธ

PROPERTY_FOR_RENT

<u>propertyNo</u>	propertyAddr	ownerNo
PG04	กรุงเทพฯ	CO42
PG16	กรุงเทพฯ	CO45
PG36	กรุงเทพฯ	CO45

RENTAL

<u>clientNo</u>	<u>propertyNo</u>	rentStart	rentFinish
CR76	PG04	01/07/2557	30/06/2558
CR76	PG16	01/09/2558	01/09/2559
CR56	PG04	15/02/2556	15/02/2557
CR56	PG36	10/10/2557	01/12/2558
CR56	PG16	01/11/2559	10/08/2560

OWNER

<u>ownerNo</u>	ownerName
CO42	ปาณิตา
CO45	เฉลิมพล

สรุป

- **1NF** ทุก attribute ในแต่ละแถวมีค่าของข้อมูลเพียงค่าเดียว
- **2NF** relation ต้องไม่มีความสัมพันธ์ระหว่าง attribute แบบบางส่วน (attribute ทุกตัวต้องขึ้นกับ key หลักทุกตัว ไม่ขึ้นอยู่กับตัวใดตัวหนึ่ง)
- **3NF** ทุก attribute ที่ไม่ใช่ key หลัก ไม่มีคุณสมบัติในการกำหนดค่าของ attribute อื่น

ตัวอย่างตารางที่ Partial Dependency (การขึ้นต่อกันบางส่วน)

<u>รหัสนักศึกษา</u>	<u>รหัสวิชา</u>	เกรด	ชื่อวิชา
534267001	F01	A	การเขียนโปรแกรม
534267001	F02	B	การออกแบบฐานข้อมูล
534267002	F01	D	การเขียนโปรแกรม
534267002	F02	A	การออกแบบฐานข้อมูล
534267003	F01	A	การเขียนโปรแกรม
534267003	F02	C	การออกแบบฐานข้อมูล

ตารางผลการเรียน

รหัสนักศึกษา	รหัสวิชา	เกรด	ชื่อวิชา
534267001	F01	A	การเขียนโปรแกรม
534267001	F02	B	การออกแบบฐานข้อมูล
534267002	F01	D	การเขียนโปรแกรม
534267002	F02	A	การออกแบบฐานข้อมูล
534267003	F01	A	การเขียนโปรแกรม
534267003	F02	C	การออกแบบฐานข้อมูล

ตารางนี้เมื่อทำให้อยู่
ในรูป 2 NF จะได้ 2
ตารางดังนี้



<u>รหัสพนักงาน</u>	ชื่อสกุล	รหัสแผนก	ชื่อแผนก	เงินเดือน
P001	นพเกศ แก้วใส	A001	บัญชี	25000
P002	วารุณี รวดเร็ว	F001	การเงิน	30000