

Normalization & Data Dictionary

Database 2

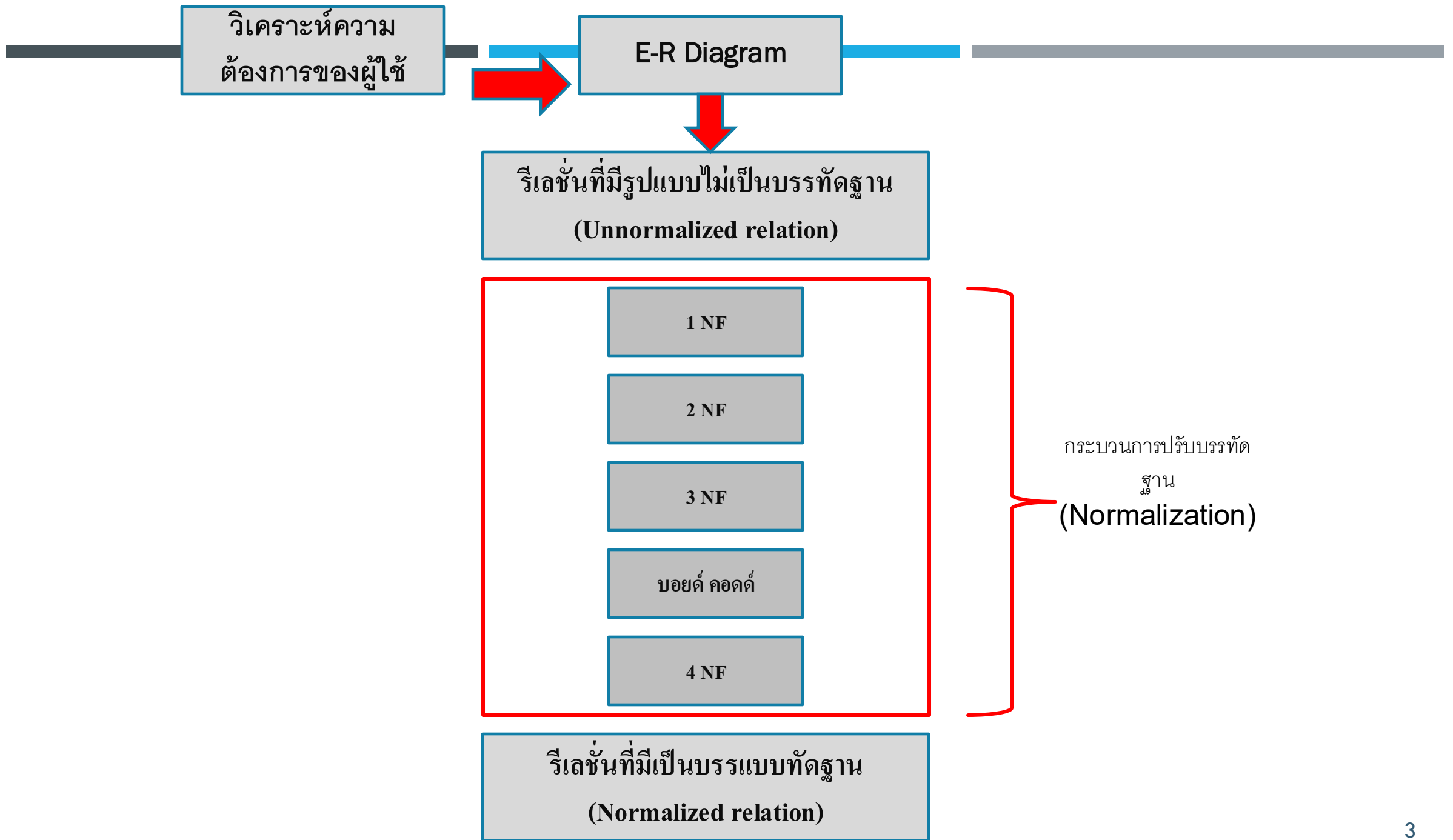


ความหมายและจุดประสงค์ของการนอร์มัลไลเซชัน

นอร์มัลไลเซชัน เป็นทฤษฎีที่ผู้ออกแบบฐานข้อมูลจะต้องนำมาใช้ในการแปลงข้อมูลที่อยู่ในรูปแบบที่ซับซ้อน ให้อยู่ในรูปแบบที่ง่ายต่อการนำไปใช้งานและก่อให้เกิดปัญหาน้อยที่สุด

ในบทนี้จะสอนกระบวนการนอร์มัลไลเซชัน ใน 3 ระดับด้วยกันคือ

- ☐ นอร์มัลไลเซชันระดับที่ 1 หรือเรียกว่า 1NF
- ☐ นอร์มัลไลเซชันระดับที่ 2 หรือเรียกว่า 2NF
- ☐ นอร์มัลไลเซชันระดับที่ 3 หรือเรียกว่า 3NF



วัตถุประสงค์ของนอร์มัลไลซ์ (Normalization)

- ❑ ลดความซ้ำซ้อนของข้อมูล เมื่อลดความซ้ำซ้อนก็ทำให้ลดเนื้อหาที่ใช้ในการจัดเก็บข้อมูล
- ❑ ลดปัญหาความไม่ถูกต้องของข้อมูล เมื่อข้อมูลไม่เกิดความซ้ำซ้อนทำให้การปรับปรุงข้อมูลสามารถทำได้จากแหล่งข้อมูลเพียงแหล่งเดียว
- ❑ ลดความผิดพลาดที่อาจเกิดจากการปรับปรุงข้อมูล (update anomalies) ซึ่งประกอบด้วย

ความซ้ำซ้อนและข้อผิดพลาดจากการปรับปรุงข้อมูล

แนวคิดหลักอันสำคัญของการออกแบบฐานข้อมูลเชิงสัมพันธ์ คือ การออกแบบให้มีการเก็บข้อมูลซ้ำซ้อนน้อยที่สุด

เพื่อประหยัดเนื้อที่ในการเก็บข้อมูลและลดปัญหาที่จะเกิดดังตัวอย่างต่อไปนี้

ตัวอย่างการออกแบบฐานข้อมูลที่ดี

Employee (พนักงาน)

รหัสพนักงาน	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา
SG21	ชูชาติ สุขศรี	ผู้จัดการ	30000	B005
SG37	ศิริ ควงเด่น	ผู้ช่วย	20000	B003
SG14	ควงใจ มีสุข	เลขานุการ	20000	B003
SG09	อจรร่า เขียวแก้ว	ผู้จัดการ	30000	B007

Branch (สาขา)

รหัสสาขา	ที่อยู่
B005	เชียงใหม่
B003	กรุงเทพ
B007	พิษณุโลก

ตัวอย่างการออกแบบฐานข้อมูลที่จะมีปัญหาของความซ้ำซ้อนตามมา

Employee_Branch (รวมรายละเอียดของพนักงานไว้ด้วยกันกับรายละเอียดของสาขา)

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

ตัวอย่างปัญหาความซ้ำซ้อนในข้อมูล รีเลชั่น EMPLOYEE_BRANCE

<u>รหัสพนักงาน</u>	<u>ชื่อ-สกุล</u>	<u>ตำแหน่ง</u>	<u>เงินเดือน</u>	<u>รหัสสาขา</u>	<u>ที่อยู่</u>
SG21	ชูชาติ สุขศรี	ผู้จัดการ	30000	B005	เชียงใหม่
SG37	ศิริ ควงเด่น	ผู้ช่วย	20000	B003	กรุงเทพ
SG14	ควงใจ มีสุข	เลขานุการ	20000	B003	กรุงเทพ
SG09	อัจฉรา เขียวแก้ว	ผู้จัดการ	30000	B007	พิษณุโลก

ความผิดพลาดจากการเพิ่ม

- ถ้าต้องการเพิ่มพนักงานใหม่ ที่อยู่สาขา B005
- จะต้องกรอก B005 และที่อยู่สาขา คือ เชียงใหม่ เพิ่มอีก

ตัวอย่างปัญหาความซ้ำซ้อนในข้อมูล รีเลชัน EMPLOYEE_BRANCE

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

ความผิดพลาดจากการเพิ่ม

- ☐ ถ้าต้องการเพิ่มสาขา จะมีปัญหาคือ ตารางนี้มีทั้งข้อมูลพนักงานและข้อมูลสาขาอยู่รวมกัน
- ☐ หากจะเพิ่มเฉพาะ รหัสสาขา และ ที่อยู่ ก็ไม่ได้เพราะ รหัสพนักงาน จะมีค่าว่างไม่ได้เพราะเป็น Primary Key ของตาราง
- ☐ ดังนั้นจะบันทึกได้ก็ต่อเมื่อมีพนักงานแล้ว

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

■ ความผิดพลาดจากการลบข้อมูล

☐ ถ้ลบข้อมูลหนึ่งแล้วส่งผลกระทบต่อข้อมูลอื่น ที่ต้องถูกลบตาม

- ☐ เช่น พนักงานรหัส SG21 ลาออก ก็ลบแฉวนั้นออก
- ☐ ข้อมูลสาขา B005 ก็จะหายไปด้วย

■ ข้อผิดพลาดจากการเปลี่ยนแปลง

☐ ในกรณีที่ต้องการเปลี่ยนแปลงข้อมูลบางตัวของสาขา

- ☐ เช่น เปลี่ยนที่อยู่ของ B003 ก็ต้องเปลี่ยนหลายที่
- ☐ ถ้าหากมีพนักงานสังกัดสาขานี้หลายที่ก็ต้องไปตามแก้ทุก ๆ ที่

- ดังนั้นเราควรแยกตาราง Employee_Branch ออกเป็นสองตาราง คือ ตารางพนักงาน และตารางสาขา

พนักงาน

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

สาขา

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

ฟังก์ชันการขึ้นต่อกัน (Functional Dependency : FD)

ถ้าให้ X และ Y เป็น Attribute ใน Relation ใดๆ แทนด้วย $R(X,Y)$

Attribute Y เป็น จะถูกเรียกว่ามีฟังก์ชันการขึ้นต่อกันกับแอททริบิวต์ X ก็ต่อเมื่อ แต่ละค่าที่ไม่ซ้ำกันของ แอททริบิวต์ X มีข้อมูลของ Y ที่เกี่ยวข้องกับ X เพียง 1 ค่า (คือ 1ต่อ1)

เขียนแทนด้วย สัญลักษณ์ $X \rightarrow Y$

ตัวอย่าง

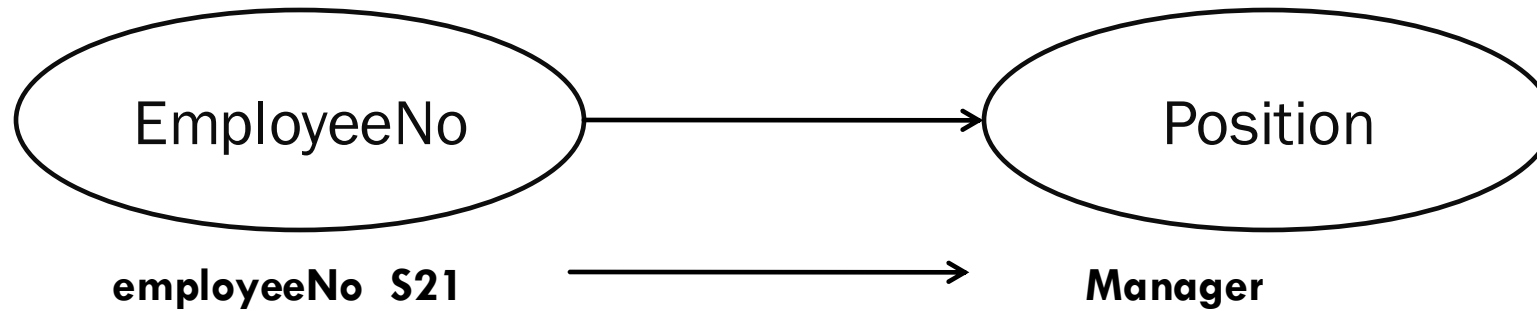
<u>EmployeeNo</u>	Name	Position
S01	ฉัตรชัย มีสมบัติ	Manager
S02	เอกชัย ใจดี	Manager Assistant
S03	มนีรัตน์ เจริญสุข	Manager
S04	ขวัญชัย ใจเพชร	Manager Assistant
S05	มานพ เกตุแก้ว	Staff
S06	ดวงกมล ทิพย์เทพ	Staff

EmployeeNo → Position

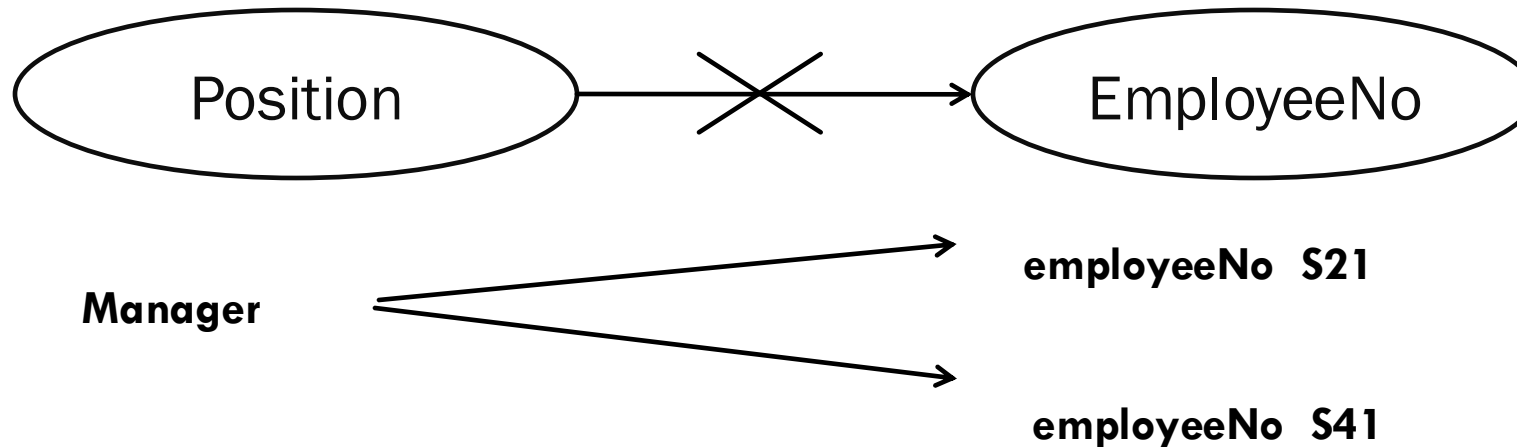
ตัวอย่าง A) EMPLOYEENO → POSITION

B) POSITION (NOT) → EMPLOYEENO

A



B



การขึ้นต่อกัน ค่า X จะต้องกำหนดค่า Y ได้ 1:1

ชนิดของฟังก์ชันการขึ้นต่อกัน (Functional Dependency :FD)

1) Complete dependencies การขึ้นต่อกันอย่างสมบูรณ์

- ☐ แอททริบิวต์ที่ไม่ใช่คีย์หลัก ขึ้นต่อ แอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่เป็นคีย์หลัก
- ☐ ตัวอย่าง ตารางที่มีแอทริบิวต์ค่าเดียวทำหน้าที่เป็นคีย์หลัก คือ หมายเลขบัตรประชาชน

หมายเลขบัตรประชาชน	ชื่อเจ้าของบัตร
3440100634931	กนกวรรณ พ่วงพงษ์
3437283420343	ชาติชาย เตชะวงศ์
2938742039485	กิงกาญ เดชาทรัพย์

หมายเลขบัตรประชาชน → ชื่อเจ้าของบัตร

ชนิดของฟังก์ชันการขึ้นต่อกัน (Functional Dependency :FD)

ตัวอย่าง ตารางที่มีแอทริบิวต์หลายตัวขึ้นกับคีย์หลักตัวเดียว

หมายเลขบัตรประชาชน	ชื่อเจ้าของบัตร	วันเกิด	วันที่ทำบัตร
3440100634931	กนกวรรณ พ่วงพงษ์	27/03/2520	28/04/2553
3437283420343	ชาติชาย เตชะวงศ์	23/06/2522	25/02/2553
2938742039485	กิงกาญ เดชาทรัพย์	21/04/2525	19/0125/52

หมายเลขบัตรประชาชน $\longrightarrow$ ชื่อเจ้าของบัตร,วันเกิด,วันที่ทำบัตร

ชนิดของฟังก์ชันการขึ้นต่อกัน (Functional Dependency :FD)

□ ตัวอย่าง ตารางที่มีแอทริบิวหลายตัวรวมกันเป็นคีย์หลัก คือ รหัสนักศึกษา รหัสวิชา

ตารางการลงทะเบียน

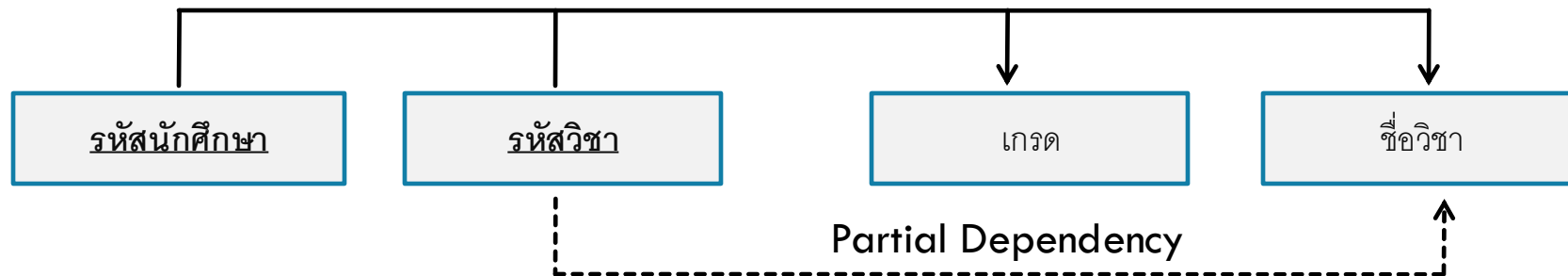
<u>รหัสนักศึกษา</u>	<u>รหัสวิชา</u>	เกรด
520014001	S001	A
520014001	S002	B
520014002	S001	C
520014002	S002	A

รหัสนักศึกษา,รหัสวิชา → เกรด

ชนิดของฟังก์ชันการขึ้นต่อกัน (Functional Dependency :FD)

2) Partial Dependency (การขึ้นต่อกันบางส่วน)

- เกิดขึ้นเมื่อคีย์หลักประกอบด้วยหลาย Attribute รวมกัน
- เมื่อแอตทริบิวต์บางส่วนของคีย์หลัก สามารถไประบุค่าแอตทริบิวต์ตัวอื่น ๆ ที่ไม่ใช่คีย์หลักของรีเลชันได้



รหัสนักศึกษา, รหัสวิชา	⇒	เกรด , ชื่อวิชา
รหัสวิชา	⇒	ชื่อวิชา

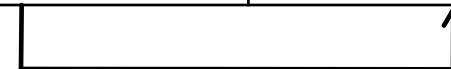
ตัวอย่างฟังก์ชันการขึ้นต่อกันแบบ PARTIAL

<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

ชนิดของฟังก์ชันการขึ้นต่อกัน (Functional Dependency : FD)

3 Transitive Dependency เกิดขึ้นเมื่อ Attribute ที่ไม่ใช่ Primary Key ไปขึ้นอยู่กับ Attribute อื่นที่ไม่ใช่ Primary Key ในรีเลชันนั้น ๆ

เลขประจำตัว	ชื่อ สกุล	ที่อยู่	ตำแหน่ง	รถประจำตำแหน่ง
01	ฉัตรชัย มีสมบัติ	กรุงเทพ	ผู้จัดการ	BMW
02	เอกชัย ใจดี	นนทบุรี	ผู้ช่วยผู้จัดการ	Honda
03	มนีรัตน์ เจริญสุข	เชียงใหม่	ผู้จัดการ	BMW
04	ขวัญชัย ใจเพชร	ราชบุรี	ผู้ช่วยผู้จัดการ	Honda

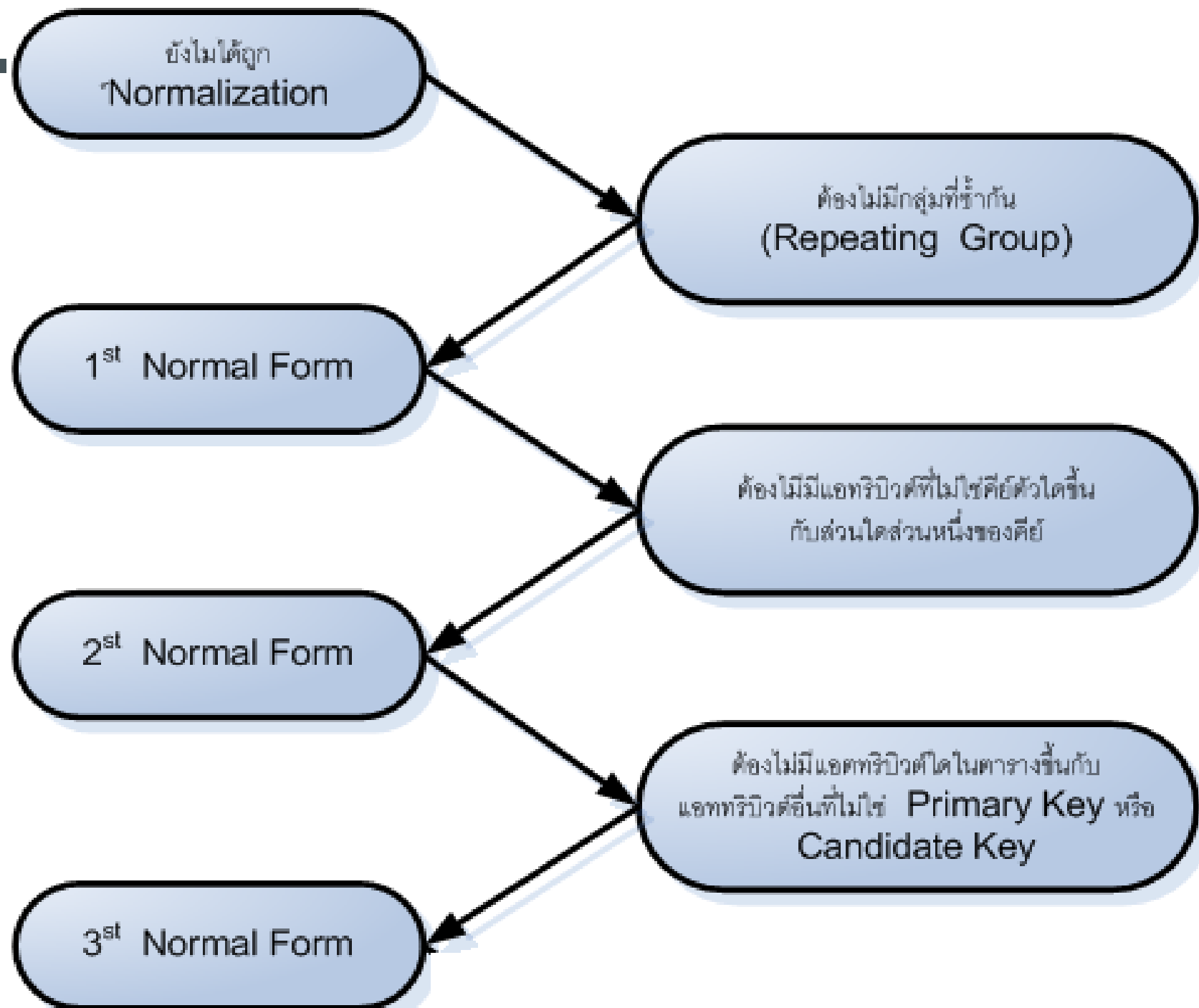


คำอธิบาย เลขประจำตัว เป็นคีย์หลัก (Primary Key) ของตาราง

เลขประจำตัว → ชื่อสกุล, ที่อยู่, ตำแหน่ง
ตำแหน่ง → รถประจำตำแหน่ง

นอร์มัลไลเซชัน (Normalization)

- Normalization คือ กระบวนการปรับปรุงโครงสร้างข้อมูลของฐานข้อมูลที่มีความซ้ำซ้อนให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน (Normal Form)
- การนอร์มัลไลเซชันมีได้ถึง 5 ระดับ ในระดับที่ 3 ก็จัดว่าเพียงพอสำหรับการออกแบบฐานข้อมูลในปัจจุบัน
 - ▣ 1NF - กำจัด repeating group
(กำจัดกลุ่มของข้อมูลที่มีความซ้ำซ้อน)
 - ▣ 2NF - กำจัด partial dependency
(กำจัดการขึ้นต่อกันบางส่วน)
 - ▣ 3NF - กำจัด transitive dependency
(กำจัดการขึ้นต่อกันของแอตทริบิวต์ที่ไม่ใช่คีย์หลัก)



First Normal Form (1NF)

- ☐ ทุก Attribute ในแต่ละ record จะเป็น single value ไม่มี ค่าของกลุ่ม ข้อมูลที่ซ้ำกัน (Repeating Group)
- ☐ ข้อมูลทุกแถว (Tuple) ต้องมีค่าไม่ซ้ำกัน

ตารางที่มีลักษณะข้อมูลเป็น Repeating group

รหัสนักศึกษา	ชื่อ	นามสกุล	รหัสวิชาที่ลงทะเบียน
001	สมชาย	สมใจนึก	204-101 204-204 204-205
002	ธีรชาย	บุญมาศ	204-102 204-204

Repeating Group

ตารางที่มีลักษณะข้อมูลเป็น Repeating group

<u>รหัสนักศึกษา</u>	<u>ชื่อ</u>	<u>นามสกุล</u>	<u>รหัสวิชาที่ลงทะเบียน</u>
001	สมชาย	สมใจนึก	204-101
			204-204
			204-205
002	ธีรชาย	บุญมาศ	204-102
			204-204

เราสามารถทำให้อยู่ในรูป 1NF ได้ดังนี้

<u>รหัสนักศึกษา</u>	<u>ชื่อ</u>	<u>นามสกุล</u>	<u>รหัสวิชาที่ลงทะเบียน</u>
001	สมชาย	สมใจนึก	204-100
001	สมชาย	สมใจนึก	204-204
001	สมชาย	สมใจนึก	204-125
002	ธีรชาย	บุญมาศ	204-102
002	ธีรชาย	บุญมาศ	204-204

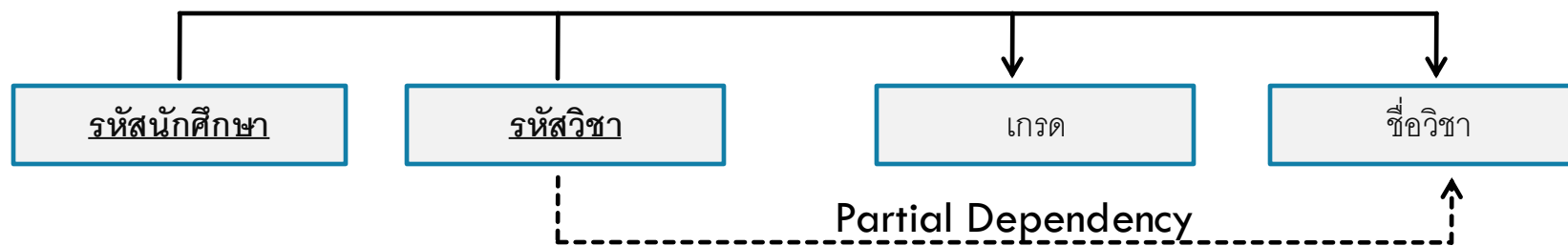
Second Normal Form (2NF)

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

สรุปก็คือ นอร์มัลไลเซชันระดับที่ 2 (Second normal form : 2NF) เป็นการจัดแอตทริบิว
ที่ไม่ขึ้นกับทั้งส่วนของคีย์หลัก ออกไป เพื่อให้แอตทริบิวอื่นทั้งหมดขึ้นตรง
กับส่วนที่เป็นคีย์หลักทั้งหมดเท่านั้น

ตัวอย่างตารางที่ PARTIAL DEPENDENCY (การขึ้นต่อกันบางส่วน)

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



รหัสนักศึกษา, รหัสวิชา → เกรด, ชื่อวิชา
 รหัสวิชา → ชื่อวิชา

Second Normal Form (2NF)

☐ วิธีขจัดปัญหา

- 1) ต้องสร้างตารางเพิ่ม
- 2) นำคอลัมย์ที่มีปัญหาไปใส่ในตารางที่สร้างเพิ่ม
- 3) กำหนดคีย์หลักให้กับตารางที่สร้างใหม่
- 4) แอททริบิวต์ใดในตารางเดิม เมื่อนำไปใส่ในตารางใหม่ให้ตัดออกจากตารางเดิม ยกเว้น ส่วนของคีย์หลัก คงไว้ในตารางเดิม

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

ตารางผลการเรียน

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

ตารางวิชา

รหัสวิชา	ชื่อวิชา
F01	การเขียนโปรแกรม
F02	การออกแบบฐานข้อมูล

ให้นักศึกษานอมนัลไลซ์ตารางนี้ให้อยู่ในรูปแบบ 2NF

ตารางผลการอบรม

<u>รหัสผู้เข้า อบรม</u>	<u>รหัสครอส อบรม</u>	ชื่อผู้เข้าอบรม	ชื่อครอสอบรม	ผลการทดสอบ
0001	TR01	นายเอ ใจดี	การซ่อมไฟฟ้า	ผ่าน
0001	TR05	นายเอ ใจดี	การซ่อมตู้เย็น	ผ่าน
0002	TR03	นางบี ใจกล้า	การทำอาหาร	ไม่ผ่าน
0002	TR09	นางบี ใจกล้า	การเลี้ยงเด็ก	ผ่าน
0003	TR01	นายรวย มีเงิน	การซ่อมไฟฟ้า	ผ่าน
0003	TR05	นายรวย มีเงิน	การซ่อมตู้เย็น	ไม่ผ่าน

Third Normal Form (3NF)

1. Relation นั้นจะต้องมีคุณสมบัติ 2NF

2. ต้องไม่มีความสัมพันธ์ระหว่าง Non-key Attribute หรือ

ไม่มี Transitive Dependency

สรุป : แอททริบิวต์ที่ไม่ใช่คีย์หลัก ต้องไม่ขึ้นต่อกันเอง

Third Normal Form (3NF)

□ วิธีขจัดปัญหา

1. สร้างตารางเพิ่ม
2. นำแอททริบิวต์ที่มีปัญหามาใส่ในตารางใหม่
3. กำหนดคีย์หลัก
4. แอททริบิวต์ที่ย้ายจากตารางเดิมไปใส่ในตารางใหม่ให้ตัดออกจากตารางเดิม
5. นำคีย์หลักในข้อ 3 ไปใส่ในตารางเดิม

ถ้าตำแหน่งงานจะต้องมีรถประจำตำแหน่ง ดังตารางด้านล่าง ถือว่า มีฟังก์ชันการขึ้นต่อกันแบบ Transitive Dependency

เลขประจำตัว	ชื่อ สกุล	ที่อยู่	ตำแหน่ง	รถประจำตำแหน่ง
01	ฉัตรชัย มีสมบัติ	กรุงเทพ	ผู้จัดการ	BMW
02	เอกชัย ใจดี	นนทบุรี	ผู้ช่วยผู้จัดการ	Honda
03	มนีรัตน์ เจริญสุข	เชียงใหม่	ผู้จัดการ	BMW
04	ขวัญชัย ใจเพชร	ราชบุรี	ผู้ช่วยผู้จัดการ	Honda

- ☐ คีย์หลักของตารางนี้คือ เลขประจำตัว
- ☐ จากตารางยังมีฟังก์ชันการขึ้นต่อกันแบบ Transitive Dependency อยู่ คือ
- ☐ ตำแหน่ง ซึ่งไม่ใช่คีย์หลักของตาราง แต่สามารถระบุค่า รถประจำตำแหน่ง ได้ คือ ถ้ารู้ตำแหน่ง ก็จะรู้รถประจำตำแหน่ง ให้แยก Table ออกมาเป็น



เลขประจำตัว	ชื่อ สกุล	ที่อยู่	รหัสตำแหน่ง
01	ฉัตรชัย มีสมบัติ	กรุงเทพ	P001
02	เอกชัย ใจดี	นนทบุรี	P002
03	มนีรัตน์ เจริญสุข	เชียงใหม่	P001
04	ขวัญชัย ใจเพชร	ราชบุรี	P002



รหัสตำแหน่ง	ตำแหน่ง	รถประจำตำแหน่ง
P001	ผู้จัดการ	BMW
P002	ผู้ช่วยผู้จัดการ	Honda

ตัวอย่าง ข้อมูลทั่วไป ก่อนทำ Normalization

P#	P_Name	E#	E_NAME	Job_Class	Chg/Hour	Hours	Total
11	ID Card	103	สมชาย ไม้ดี	Elect.Engineer	500	23.8	11900
		101	แทน งามยิ่ง	Database Designer	500	19.4	9700
		105	ชาย ศรีดี	Database Designer	500	35.7	17850
		106	แม็ก ยอดยิ่ง	Programmer	200	12.6	2520
		102	อมร ศรีดี	System Analyst	400	23.8	9520
Subtotals							51490
22	Palm	114	สุรศักดิ์ ดิงาม	Application Designer	200	24.6	4920
		118	กมล ไม้งาม	General Support	100	45.3	4530
		104	นาย ยิ่งยอด	System Analyst	250	32.4	8100
		112	ธิดา ไม้งาม	Dss Analyst	200	44.0	8800
		Subtotals					
33	Web Service	105	ชาย ศรีดี	Database Designer	500	64.7	32350
		104	นาย ยิ่งยอด	System Analyst	250	48.4	12100
		113	วิว ยิ่งดี	Application Designer	250	23.6	5900
		111	หิน งามดี	Clerical Support	200	22.0	4400
		106	แม็ก ยอดยิ่ง	Programmer	200	12.8	2560
Subtotals							57310

รูปที่ 6.2 รายงานค่าใช้จ่ายในแต่ละโครงการ

1NF

P_No (PK)	P_Name	Emp_No (PK)	Emp_name	Job_Class	Chg_Hour	Hour	Total
11	ID_CARD	103	สมชาย ไม้ดี	Elect Engineer	500	23	11,500.00
11	ID_CARD	101	แทน งามยิ่ง	Database Designer	500	19	9,500.00
11	ID_CARD	105	ชาย ศรีดี	Database Designer	500	35	17,500.00
11	ID_CARD	106	แม็ก ยอดยิ่ง	Programmer	200	12	2,400.00
11	ID_CARD	102	อมร ศรีดี	System Analyst	250	23	5,750.00
22	Palm	114	สุศักดิ์ ดีงาม	Application Designer	250	24	6,000.00
22	Palm	118	กมล ไม้งาม	General Support	100	45	4,500.00
22	Palm	104	นาย ยิ่งยอด	System Analyst	250	32	8,000.00
22	Palm	112	จิตา ไม้งาม	DSS Analyst	200	44	8,800.00
33	Web Service	105	ชาย ศรีดี	Database Designer	500	64	32,000.00
33	Web Service	104	นาย ยิ่งยอด	System Analyst	250	48	12,000.00
33	Web Service	113	วิว ยิ่งดี	Application Designer	250	23	5,750.00
33	Web Service	111	หิน งามดี	Critical Suport	200	22	4,400.00
33	Web Service	106	แม็ก ยอดยิ่ง	Programmer	200	12	2,400.00

DATA 2F

Table Project

P_No (PK)	P_Name
11	ID_CARD
22	Palm
33	Web Service

Table Employee

Emp_No (PK)	Emp_name	Job_Class	Chg_Hour
101	แพน งามยิ่ง	Database Designer	500
102	อมร ศรีดี	System Analyst	250
103	สมชาย ไม้ดี	Elect Engineer	500
104	นาย ยิ่งยอด	System Analyst	250
105	ชาย ศรีดี	Database Designer	500
106	แม็ก ยอดยิ่ง	Programmer	200
111	หิน งามดี	Critical Suport	200
112	ธิดา ไม้งาม	DSS Analyst	200
113	วิว ยิ่งดี	Application Designer	250
114	สุรศักดิ์ ดีงาม	Application Designer	250
118	กมล ไม้งาม	General Support	100

Table Work

P_No (PK)	Emp_No (PK)	Hour
11	101	19
11	102	23
11	103	23
11	105	35
11	106	12
22	104	32
22	112	44
22	114	24
22	118	45
33	104	48
33	105	64
33	106	12
33	111	22
33	113	23

Table Project	
P_No (PK)	P_Name
11	ID_CARD
22	Palm
33	Web Service

Table Employee		
Emp_No (PK)	Emp_name	Job_No (FK)
101	แท่น งามยิ่ง	J003
102	อมร ศรีดี	J001
103	สมชาย ไม้ดี	J002
104	นาย ยิ่งยอด	J001
105	ชาย ศรีดี	J003
106	แม็ก ยอดยิ่ง	J004
111	หิน งามดี	J005
112	ธิดา ไม่งาม	J008
113	วิวิ ยิ่งดี	J006
114	สุรศักดิ์ ดีงาม	J006
118	กมล ไม่งาม	J007

Table job		
Job_No (PK)	Job_Class	Chg_Hour
J001	System Analyst	250
J002	Elect Engineer	500
J003	Database Designer	500
J004	Programmer	200
J005	Critical Suport	200
J006	Application Designer	250
J007	General Support	100
J008	DSS Analyst	200

Work		
P_No (PK)	Emp_No (PK)	Hour
11	101	19
11	102	23
11	103	23
11	105	35
11	106	12
22	104	32
22	112	44
22	114	24
22	118	45
33	104	48
33	105	64
33	106	12
33	111	22
33	113	23

สรุป Normalization

1NF ทุกแอททริบิวต์ในแต่ละแถวมีค่าของข้อมูลเพียงค่าเดียว

2NF รีเลชันนั้นต้องไม่มีความสัมพันธ์ระหว่างแอททริบิวต์แบบบางส่วน (แอททริบิวต์ทุกตัวต้องขึ้นกับคีย์หลักทุกตัว ไม่ขึ้นอยู่กับตัวใดตัวหนึ่ง)

3NF ทุกแอททริบิวต์ที่ไม่ใช่คีย์หลักไม่มีคุณสมบัติในการกำหนดค่าของ แอททริบิวต์อื่น

แปลงตารางต่อไปนี้ให้อยู่ในรูป NF1-NF3

1)การลงทะเบียนเรียน

รหัสนักศึกษา	ชื่อนักศึกษา	รหัสวิชา	ชื่อวิชา	หน่วยกิต	เกรด
53001	วนิดา	AB12	บัญชี	3	A
		CD01	การเงิน	3	A
		PC09	สถิติ	3	B
53009	สุมาลี	AB12	บัญชี	3	B
		CD01	การเงิน	3	A

1NF

รหัสนักศึกษา	ชื่อนักศึกษา	รหัสวิชา	ชื่อวิชา	หน่วยกิต	เกรด
53001	วนิดา	AB12	บัญชี	3	A
53001	วนิดา	CD01	การเงิน	3	A
53001	วนิดา	PC09	สถิติ	3	B
53009	สุมาลี	AB12	บัญชี	3	B
53009	สุมาลี	CD01	การเงิน	3	A

2NF

ตารางสอบ

<u>รหัส นักศึกษา</u>	<u>ชื่อ นักศึกษา</u>	รหัสวิชา	เกรด
53001	วนิดา	AB12	A
53001	วนิดา	CD01	A
53001	วนิดา	PC09	B
53009	สุมาลี	AB12	B
53009	สุมาลี	CD01	A

ตารางวิชา

รหัสวิชา	<u>ชื่อวิชา</u>	หน่วยกิต
AB12	บัญชี	3
CD01	การเงิน	3
PC09	สถิติ	3

3NF

ตารางสอบ

<u>รหัส นักศึกษา</u>	<u>รหัสวิชา</u>	เกรด
53001	AB12	A
53001	CD01	A
53001	PC09	B
53009	AB12	B
53009	CD01	A

ตารางวิชา

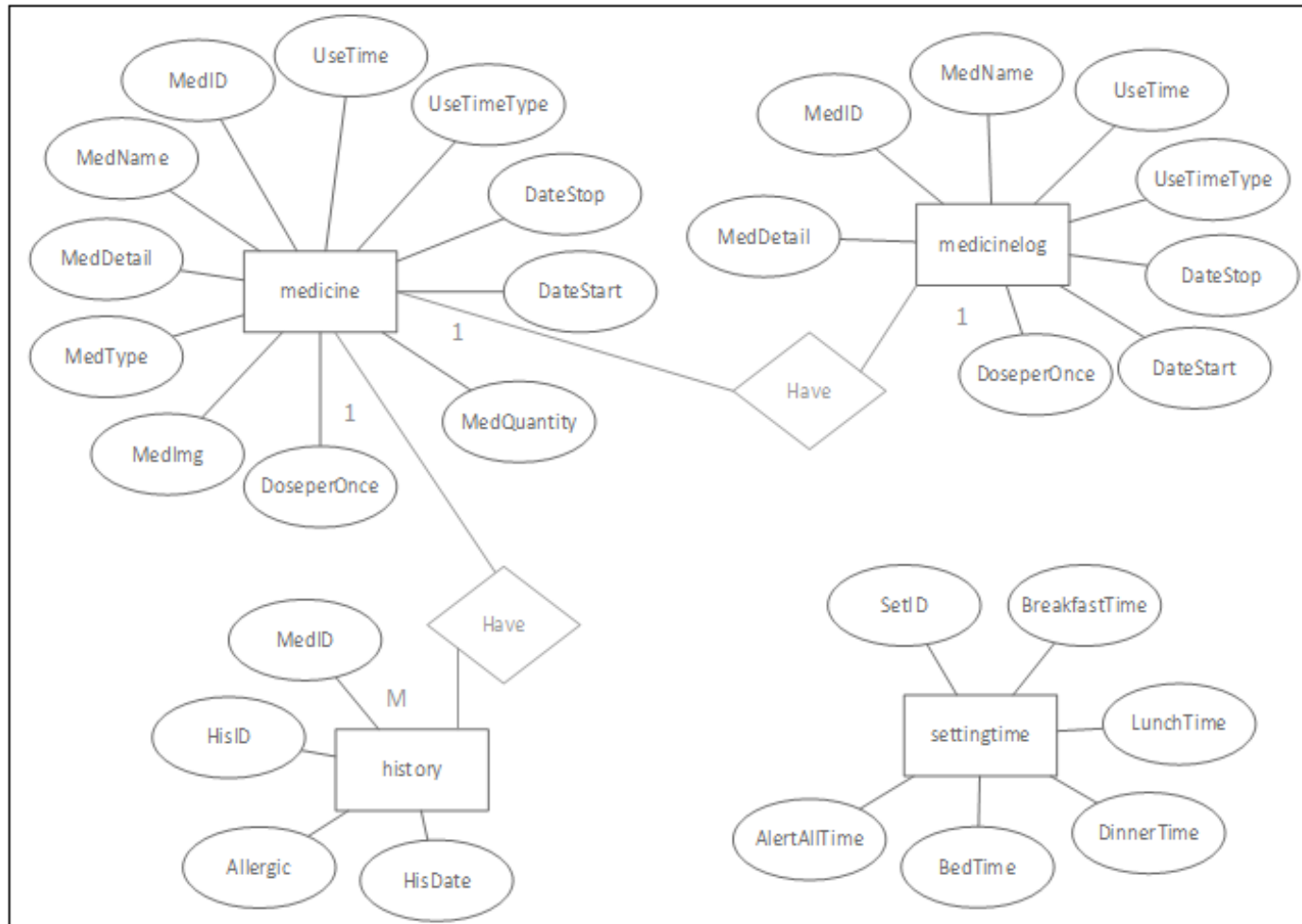
<u>รหัสวิชา</u>	<u>ชื่อวิชา</u>	หน่วยกิต
AB12	บัญชี	3
CD01	การเงิน	3
PC09	สถิติ	3

ตารางนักศึกษา

<u>รหัส นักศึกษา</u>	<u>ชื่อ นักศึกษา</u>
53001	วนิดา
53009	สุมาลี

การทำ Data Dictionary

ตัวอย่างแอปแจ้งเตือนการรับประทานยา ได้จัดทำ ER มาดังนี้



การทำ Data Dictionary ต่อ

หลังจากนั้นจะต้องอธิบายแฟ้มข้อมูลภายในระบบหรือโครงสร้างฐานข้อมูล (Data Dictionary) ดังนี้

3 แฟ้มข้อมูลภายในระบบหรือโครงสร้างฐานข้อมูล (Data Dictionary)

3.1 ตารางข้อมูลยา (medicine) เป็นตารางที่ใช้เก็บรายละเอียดเกี่ยวกับข้อมูลยาเช่น ชื่อยา ชนิด จำนวน วันที่รับประทาน

ตารางที่ 3-1 ข้อมูลยา (medicine)

ลำดับ	ชื่อฟิลด์	ชนิดและขนาดฟิลด์	คำอธิบาย	Key
1	MedID	INTEGER	รหัสลำดับยา	PK
2	MedName	TEXT(100)	ชื่อยา	
3	MedDetail	TEXT(100)	รายละเอียด ใช้เก็บคุณลักษณะการรักษา เช่น แก้ไข แก้ปวด	
4	MedType	TEXT(100)	ชนิดยา เช่น ยาเม็ด ยาน้ำ	
5	MedImg	TEXT(100)	เก็บ Path รูปภาพยา	
6	DoseperOnce	TEXT(20)	จำนวนบริโภคยาต่อครั้ง	
7	MedQuantity	TEXT(20)	จำนวนยาที่มีทั้งหมด	
8	DateStart	DATETIME	วันที่เริ่มรับประทานยา	
9	DateStop	DATETIME	วันสิ้นสุดการรับประทาน	
10	UseTimeType	TEXT(20)	ช่วงเวลาการใช้อาบน้ำ Code การใช้ (B : อาหารเช้า L : อาหารกลางวัน D : อาหารเย็น S : ก่อนนอน และ E : ใ้ตามเวลา)	
11	UseTime	INTEGER	เก็บเวลาเป็นนาที	

การทำ Data Dictionary ต่อ

3.2 ตารางข้อมูลยาสำหรับเก็บเป็นลือครายงาน (medicinelog) เป็นตารางที่ใช้เก็บลือคการใช้ยาโดยจะเก็บรายละเอียดเกี่ยวกับข้อมูลยา เช่น ชื่อยา ชนิด จำนวน วันที่รับประทาน

ตารางที่ 3-2 ข้อมูลยาสำหรับเก็บเป็นลือครายงาน (medicinelog)

ลำดับ	ชื่อฟิลด์	ชนิดและขนาดฟิลด์	คำอธิบาย	Key
1	MedID	INTEGER	รหัสลำดับยา	PK
2	MedName	TEXT(100)	ชื่อยา	
3	MedDetail	TEXT(100)	รายละเอียด ใช้เก็บคุณลักษณะการรักษา เช่น แก้ไข แก้ปวด	
4	DoseperOnce	TEXT(20)	จำนวนบริโภคยาต่อครั้ง	
5	DateStart	DATETIME	วันและเวลาเริ่มรับประทานยา	
6	DateStop	DATETIME	วันและเวลาสิ้นสุดการรับประทานยา	
7	UseTimeType	TEXT(20)	ช่วงเวลาการใช้ยาเก็บ Code การใช้ (B : อาหารเช้า L : อาหารกลางวัน D : อาหารเย็น S : ก่อนนอน E : ใช้ตามเวลา)	
8	UseTime	INTEGER	เก็บเวลาเป็นนาทื	

การทำ Data Dictionary ต่อ

3.3 ตารางประวัติการใช้ยา (history) เป็นตารางที่ใช้เก็บประวัติการใช้ยา เก็บอาการที่เกิดจากการใช้ยา เช่น อาการแพ้ยา

ตารางที่ 3-3 ข้อมูลประวัติการใช้ยา (history)

ลำดับ	ชื่อฟิลด์	ชนิดและขนาดฟิลด์	คำอธิบาย	Key
1	HisID	INTEGER	รหัสลำดับประวัติการใช้ยา	PK
2	MedID	INTEGER	รหัสลำดับยา	FK
3	Allergic	TEXT(100)	อาการที่เกิดจากการใช้ยา	
4	HisDate	DATETIME	วันเวลาที่บันทึกข้อมูล	

* ต้องทำตารางอธิบาย Data Dictionary ทั้งหมดที่แสดงใน ER

การทำ Data Dictionary ต่อ

Data Dictionary จะประกอบไปด้วยคำอธิบาย Properties หรือ Attribute ที่เกี่ยวข้องเช่น

- รหัส
- ชื่อ
- ชื่อเรียกอื่นในระบบ
- คำอธิบาย
- ชนิดของข้อมูล
- ขนาดของข้อมูล
- ค่าที่แสดงโดยอัตโนมัติ
- หมายเหตุ

เงื่อนไข

กรณีที่เป็นตัวเลขที่มีค่าความต่อเนื่อง

- ค่าสูงสุด
- ค่าต่ำสุด
- จุดทศนิยม

กรณีที่ไม่เป็นตัวเลขที่มีค่าความต่อเนื่อง

- ค่า (ไซส์)
- ความหมาย (S, M, L, XL)

ชนิดและขนาดของ Field สามารถกำหนดได้ดังนี้

Data Type	Meaning
Bit	A value of 1 or 0, a true/false value
Char, varchar, text	Any alphanumeric character
Datetime, smalldatetime	Alphanumeric data, several formats
Decimal, numeric	Numeric data that are accurate to the least significant digit; can contain a whole and decimal portion
Float, real	Floating-point values that contain an approximate decimal value
Int, smallint, tinyint	Only integer (whole digit) data
Currency, money, smallmoney	Monetary numbers accurate to four decimal places
Binary, varbinary, image	Binary strings (sound, pictures, video)
Cursor, timestamp, uniqueidentifier	A value that is always unique within a database
Autonumber	A number that is always incremented by one when a record is added to a database table

สัปดาห์หน้า (24 ต.ค. 66) สอนออนไลน์ ผ่าน Google Meet

* นิสิตต้องใช้ PC สำหรับออกแบบ UX/UI ด้วย โปรแกรม Adobe XD/เว็บไซต์ ☐ Figma