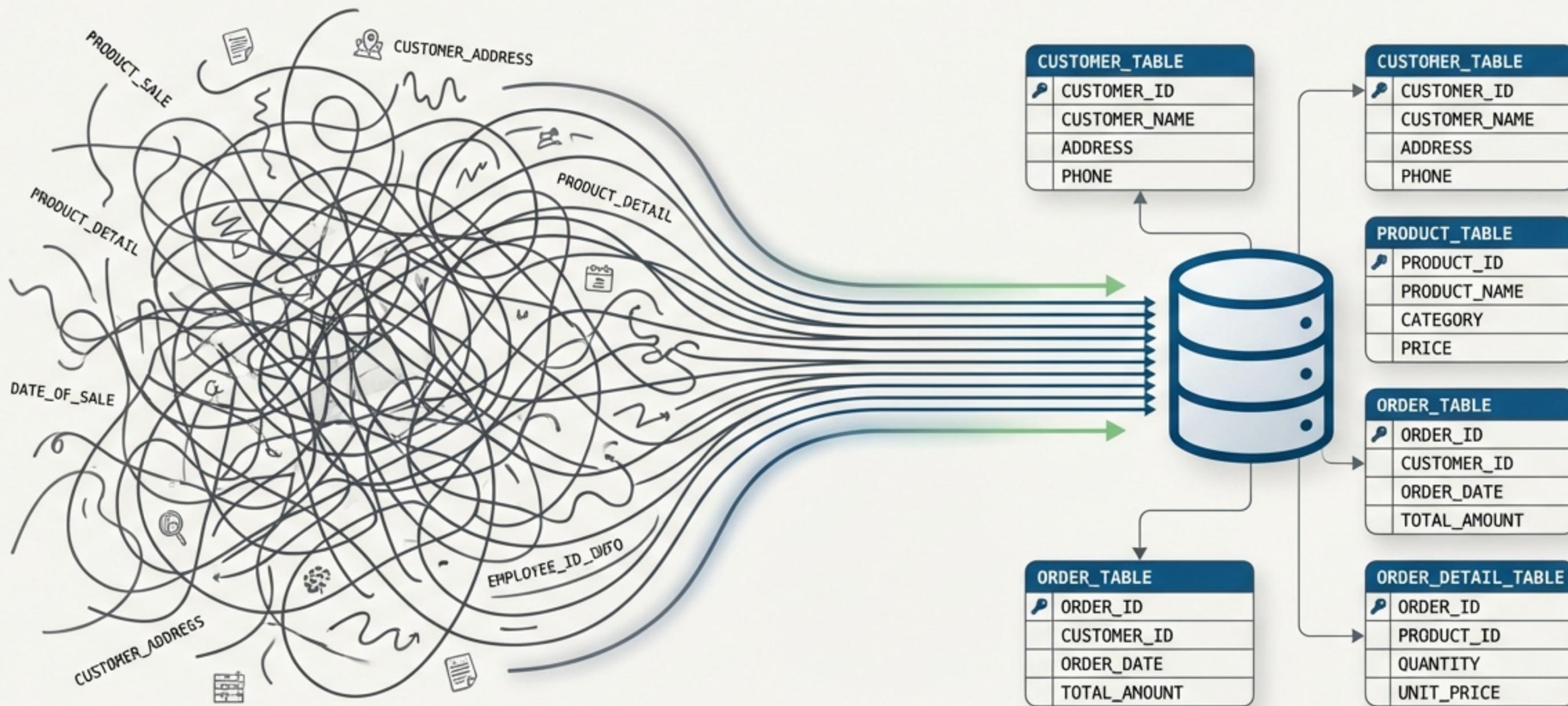


จัดการข้อมูลให้อยู่หมวด: ความลับของ Normalization

จากข้อมูลที่ยุ่งเหยิงสู่ฐานข้อมูลที่มีประสิทธิภาพ



สภาพของข้อมูลที่ยังไม่เป็นระเบียบ

ลองพิจารณาตารางที่เก็บข้อมูลพนักงานและสาขารวมกัน จะเห็นได้ว่ามีข้อมูลซ้ำซ้อนเกิดขึ้นมากมาย

EMPLOYEE_BRANCH

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

ผลกระทบที่ร้ายแรง: ปัญหาความผิดพลาดจากการปรับปรุงข้อมูล

ความซ้ำซ้อนของข้อมูลไม่ได้เป็นเพียงแค่การสิ้นเปลืองพื้นที่ แต่ยังนำไปสู่ปัญหาใหญ่ 3 ประการที่เรียกว่า 'Update Anomalies'



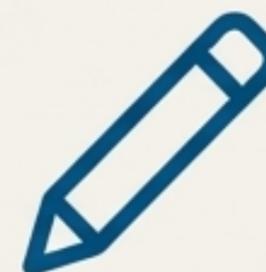
Insertion Anomaly Insertion Anomaly (ความผิดพลาดจากการเพิ่ม)

ไม่สามารถเพิ่มข้อมูลใหม่ได้ หากข้อมูลบางส่วนยังไม่ครบถ้วน



Deletion Anomaly Deletion Anomaly (ความผิดพลาดจากการลบ)

การลบข้อมูลหนึ่ง อาจทำให้ข้อมูลสำคัญอีกส่วนหนึ่งหายไปโดยไม่ตั้งใจ



Modification Anomaly (ข้อผิดพลาดจากการเปลี่ยนแปลง)

ต้องแก้ไขข้อมูลเดียวกันในหลายๆ ที่ ซึ่งเสี่ยงต่อความผิดพลาดและความไม่สอดคล้องกัน

เจาะลึกปัญหา: ตัวอย่างความผิดพลาดที่เกิดขึ้นจริง

EMPLOYEE_BRANCH

รหัสพนักงาน	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา	ที่อยู่
SG21	ชชาติ สุทธิ	ผู้จัดการ	30000	B005	เชียงใหม่
SG37	ศรี ดวงเด่น	ผู้ช่วย	20000	B003	กรุงเทพฯ
SG14	ดวงใจ มีสุข	เลขาฯ	20000	B003	กรุงเทพฯ
SG09	สิทธรา เขียวแก้ว	ผู้จัดการ	30000	B007	พิจิตร

1. Insertion Anomaly

จะเกิดอะไรขึ้นถ้าเราต้องการเพิ่มสาขาใหม่ (B002, ขอนแก่น) แต่ยังไม่มีพนักงาน?

ทำไม่ได้! เพราะ **รหัสพนักงาน** เป็น Primary Key ซึ่งจะมีค่าว่าง (NULL) ไม่ได้ การออกแบบนี้ บังคับให้เราต้องมีพนักงานก่อนจึงจะเพิ่มสาขาได้

EMPLOYEE_BRANCH

รหัสพนักงาน	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา	ที่อยู่
NULL	NULL	NULL	NULL	B002	ขอนแก่น

2. Deletion Anomaly

จะเกิดอะไรขึ้นถ้าพนักงาน SG21 (ชชาติ สุทธิ) ลาออก?

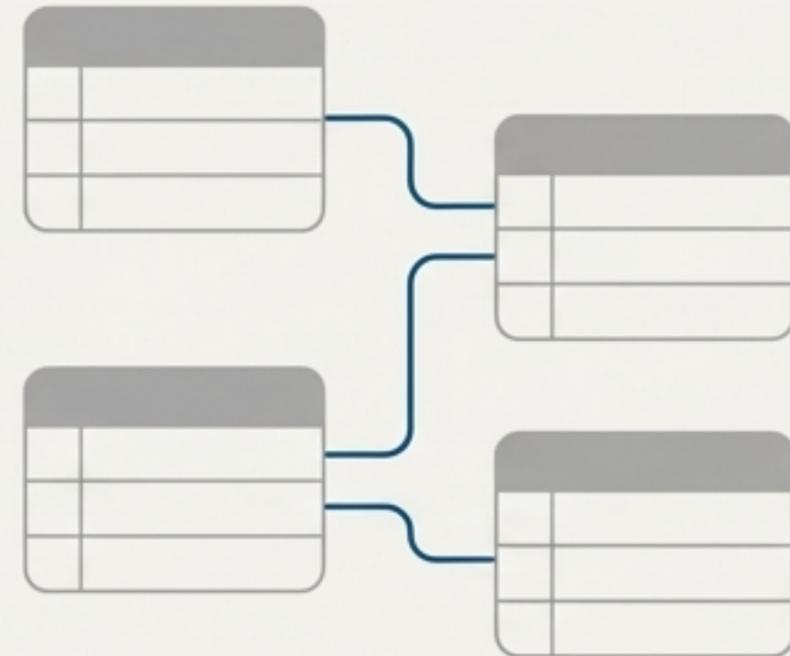
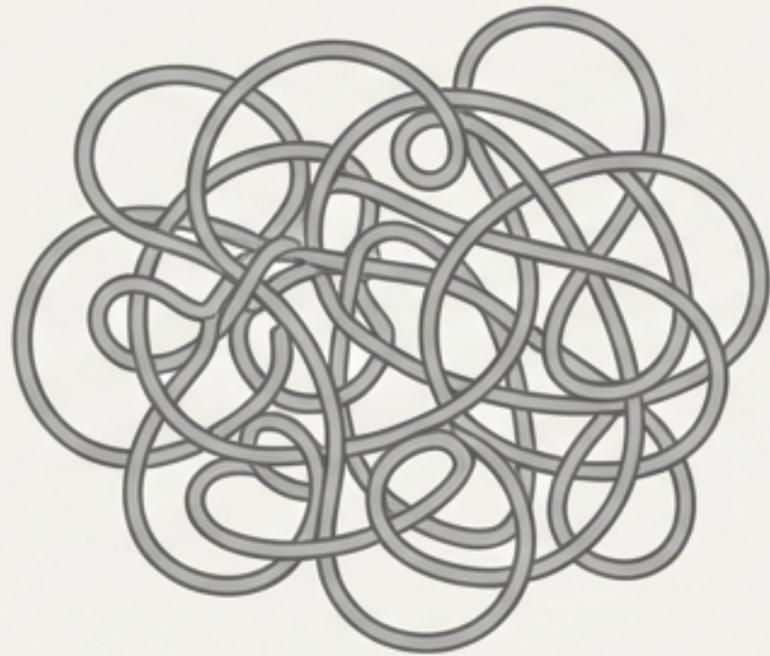
เมื่อเราลบแถวของพนักงาน SG21 ข้อมูลของสาขา B005 (เชียงใหม่) ก็จะหายไปจากฐานข้อมูลทั้งหมด!

EMPLOYEE_BRANCH

รหัสพนักงาน	ชื่อ-สกุล	ตำแหน่ง	เงินเดือน	รหัสสาขา	ที่อยู่
SG21	ชชาติ สุทธิ	ผู้จัดการ	30000	B005	เชียงใหม่
SG21	ชชาติ สุทธิ	ผู้จัดการ	30000	B005	เชียงใหม่
SG37	ศรี ดวงเด่น	ผู้ช่วย	20000	B003	กรุงเทพฯ
SG14	ดวงใจ มีสุข	เลขาฯ	20000	B003	กรุงเทพฯ
SG09	สิทธรา เขียวแก้ว	ผู้จัดการ	30000	B007	พิจิตร

ข้อมูลสาขา B005 หายไป!

ทางออกคือ Normalization



Definition:

Normalization คือกระบวนการเชิงทฤษฎีที่ใช้ในการออกแบบและปรับโครงสร้างฐานข้อมูล เพื่อลดความซ้ำซ้อนและขจัดปัญหา Update Anomalies

วัตถุประสงค์หลัก

- ลดความซ้ำซ้อนของข้อมูล (Reduce Data Redundancy): ประหยัดพื้นที่จัดเก็บและทำให้ข้อมูลเป็นระเบียบ
- ลดปัญหาความไม่ถูกต้องของข้อมูล (Reduce Data Inconsistency): ทำให้การปรับปรุงข้อมูลทำได้จากแหล่งเดียว
- ลดความผิดพลาดจากการปรับปรุงข้อมูล (Eliminate Update Anomalies): แก้ปัญหาการเพิ่ม, ลบ, และแก้ไขข้อมูลที่ผิดพลาด

เครื่องมือสำคัญ: ทำความรู้จัก Functional Dependency (FD)

FD คือความสัมพันธ์ที่ระบุว่าค่าของ Attribute หนึ่ง (หรือกลุ่มของ Attribute) สามารถใช้กำหนดค่าของ Attribute อื่นได้อย่างเฉพาะเจาะจง

Notation: $X \rightarrow Y$

อ่านว่า: 'X determines Y' หรือ 'Y is functionally dependent on X'

รหัสพนักงาน \rightarrow ชื่อ-สกุล

คำอธิบาย: ถ้ารู้ 'รหัสพนักงาน' เราจะสามารถระบุ 'ชื่อ-สกุล' ของพนักงานคนนั้นได้เพียงค่าเดียวเสมอ



EmployeeNo	Name	Position
S01	ฉัตรชัย มีสมบัติ	Manager
S02	เอกชัย ใจดี	Manager Assistant
S03	มณีนีรัตน์ เจริญสุข	Manager
S04	ขวัญชัย ใจเพชร	Manager Assistant
S05	มานพ เกตุแก้ว	Staff
S06	ดวงกมล ทิพย์เทว	Staff

ประเภทของ Dependencies ที่ต้องรู้



1. Complete Dependency (การขึ้นต่อกันอย่างสมบูรณ์)

Attribute ที่ไม่ใช่คีย์หลักขึ้นอยู่กับ Attribute ที่ไม่ใช่คีย์หลักขึ้นอยู่กับ Primary Key **ทั้งหมด** (A non-key attribute depends on the *entire* primary key). นี่คือเป้าหมายที่เราต้องการ (This is our goal).

(`รหัสนักศึกษา`, `รหัสวิชา`) → `เกรด`



2. Partial Dependency (การขึ้นต่อกันบางส่วน)

Attribute ที่ไม่ใช่คีย์หลักขึ้นอยู่กับ **เพียงบางส่วน** ของ Composite Primary Key (A non-key attribute depends on only *part* of a composite primary key). **เป็นปัญหาที่ต้องแก้ไขใน 2NF**

`รหัสวิชา` → `ชื่อวิชา` (ในตารางที่มี PK เป็น `รหัสนักศึกษา`, `รหัสวิชา`)



3. Transitive Dependency (การขึ้นต่อกันของแอตทริบิวต์ที่ไม่ใช่คีย์หลัก)

Attribute ที่ไม่ใช่คีย์หลักขึ้นอยู่กับ Attribute อื่นที่ไม่ใช่คีย์หลักเช่นกัน (A non-key attribute depends on another non-key attribute). **เป็นปัญหาที่ต้องแก้ไขใน 3NF**

`รหัสนักงาน` → `รหัสแผนก` → `ชื่อแผนก`

ขั้นที่ 1: สู่ First Normal Form (1NF)

กฎของ 1NF คือ 'กำจัด Repeating Groups'

ทุก Attribute ในแต่ละแถวจะต้องมีค่าเดียว (atomic value) ไม่สามารถเก็บข้อมูลเป็นกลุ่มซ้ำซ้อนในช่องเดียวได้

ตารางที่ยังไม่เป็น 1NF			
รหัสนักศึกษา	ชื่อ	นามสกุล	รหัสวิชาที่ลงทะเบียน
001	สมชาย	สมใจนึก	204-101 204-204 204-205
002	ธีรชาย	บุญมาศ	204-102 204-204

การแปลงข้อมูลให้อยู่ในรูปแบบ 1NF

ก่อน (Before)

รหัส นักศึกษา	ชื่อ	นามสกุล	รหัสวิชาที่ ลงทะเบียน
001	สมชาย	สมใจนึก	204-101 204-204 204-205
002	ธีรชาย	บุญมาศ	204-102 204-204

แยกแถวเพื่อ
กำจัด Repeating
Group



หลัง (After)

รหัส นักศึกษา	ชื่อ	นามสกุล	รหัสวิชาที่ ลงทะเบียน
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



อยู่ในรูปแบบ
1NF แล้ว

ขั้นที่ 2: สู่ Second Normal Form (2NF)

1. ต้องเป็น 1NF มาก่อน
2. ต้อง**ไม่มี Partial Dependency**

ทุก Attribute ที่ไม่ใช่คีย์หลัก จะต้องขึ้นอยู่กับ Primary Key **ทั้งหมดทุกส่วน** ไม่ใช่แค่บางส่วน
ปัญหานี้จะเกิดกับตารางที่มี Composite Primary Key เท่านั้น



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

ตาราง 1NF ที่มี Partial Dependency

Primary Key: **PK = (`รหัสนักศึกษา`, `รหัสวิชา`)**

การจัด Partial Dependency เพื่อให้เป็น 2NF

เราแก้ไขปัญหานี้โดยการแยกตารางออกเป็นสองตาราง

Before

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

Partial Dependency!

ตาราง 1NF ที่มี Partial Dependency

Primary Key: PK = (`รหัสนักศึกษา`, `รหัสวิชา`)

After



✓ อยู่ในรูปแบบ 2NF แล้ว

Table 1: ตารางผลการเรียน (Grades Table)

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

Primary Key: PK = (`รหัสนักศึกษา`, `รหัสวิชา`)

✓ อยู่ในรูปแบบ 2NF แล้ว

Table 2: ตารางวิชา (Courses Table)

รหัสวิชา	ชื่อวิชา
F01	การเขียนโปรแกรม
F02	การออกแบบฐานข้อมูล

Primary Key: PK = `รหัสวิชา`

ขั้นที่ 3: สู่ Third Normal Form (3NF)

1. ต้องเป็น 2NF มาก่อน
2. ต้อง**ไม่มี Transitive Dependency**
ห้ามมี Attribute ที่ไม่ใช่คีย์หลักไปขึ้นอยู่กับ Attribute อื่นที่ไม่ใช่คีย์หลักเช่นกัน

ตาราง 2NF ที่มี Transitive Dependency

Primary Key: PK = `รหัสพนักงาน`

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

การขจัด Transitive Dependency เพื่อให้เป็น 3NF

เช่นเดียวกับ 2NF เราแก้ไขโดยการแยกตารางที่มีปัญหาออกไป

Before

ตาราง 2NF ที่มี Transitive Dependency
Primary Key: PK = `รหัสพนักงาน`

`รหัสพนักงาน`	`ชื่อสกุล`	`รหัสแผนก`	`ชื่อแผนก`	`เงินเดือน`
P001	นพเกศ แก้วใส	A001	บัญชี	25000
P002	วารุณี รวดเร็ว	F001	การเงิน	30000



After

Table 1: ตารางพนักงาน (Employee Table) ✓ อยู่ในรูปแบบ 3NF แล้ว

`รหัสพนักงาน`	`ชื่อสกุล`	`เงินเดือน`	`รหัสแผนก` Foreign Key
P001	นพเกศ แก้วใส	25000	A001
P002	วารุณี รวดเร็ว	30000	F001

Primary Key: PK = `รหัสพนักงาน`

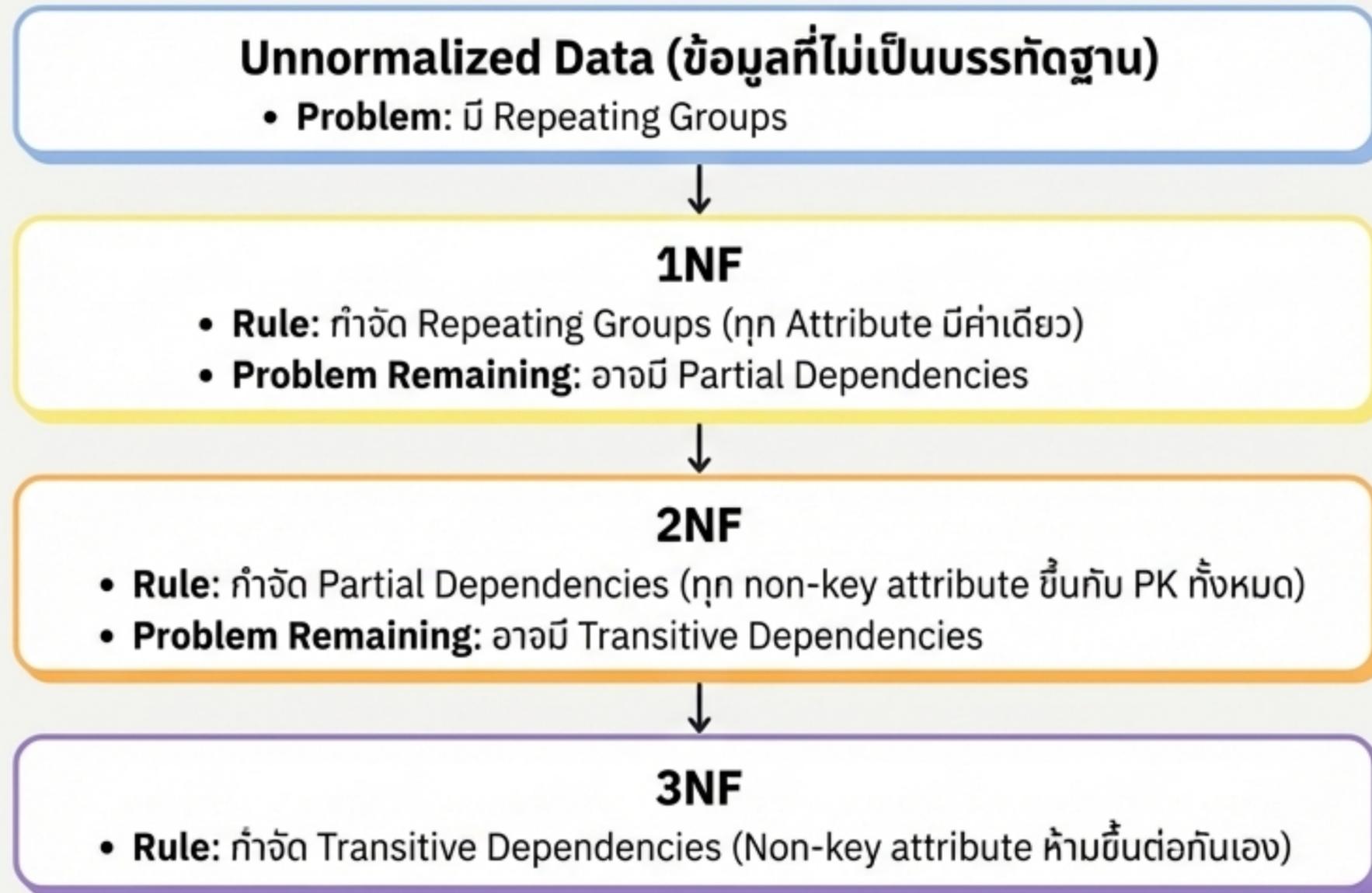
✓ อยู่ในรูปแบบ 3NF แล้ว

Table 2: ตารางแผนก (Department Table)

`รหัสแผนก`	`ชื่อแผนก`
A001	บัญชี
F001	การเงิน

Primary Key: PK = `รหัสแผนก`

สรุปเส้นทางสู่ 3NF: จากความยุ่งเหยิงสู่ความเป็นระเบียบ



การทำ Normalization ถึงระดับ 3NF ถือว่าเพียงพอสำหรับการออกแบบฐานข้อมูลส่วนใหญ่ ทำให้ได้ฐานข้อมูลที่มีประสิทธิภาพ ลดความซ้ำซ้อน และรักษาความถูกต้องของข้อมูล

ถึงตาคุณแล้ว: ลองนำไปใช้

จากความรู้ทั้งหมดที่ได้เรียนมา ลองปรับโครงสร้างตาราง 'การสั่งซื้อสินค้า' ต่อไปนี้ให้อยู่ในรูปแบบ 3NF

การสั่งซื้อสินค้า (Purchase Order)							
เลขที่ใบสั่งซื้อ	วันที่ซื้อ	รหัสลูกค้า	ชื่อผู้สั่ง	รหัสสินค้า	ชื่อสินค้า	จำนวนที่ซื้อ	ราคาต่อหน่วย
OR001	02/09/2552	C001	เดวิด	AB12	ตู้เย็น	4	4000
OR001	02/09/2552	C001	เดวิด	CD01	พัดลม	3	2000
OR001	02/09/2552	C001	เดวิด	PC09	แอร์	4	6000
OR002	02/09/2552	C005	ไมเคิล	TP01	เตาอบ	3	3000
OR002	02/09/2552	C005	ไมเคิล	CD01	พัดลม	2	2000

คุณจะสร้างตารางใหม่อะไรบ้าง และแต่ละตารางมี Attribute ใด?