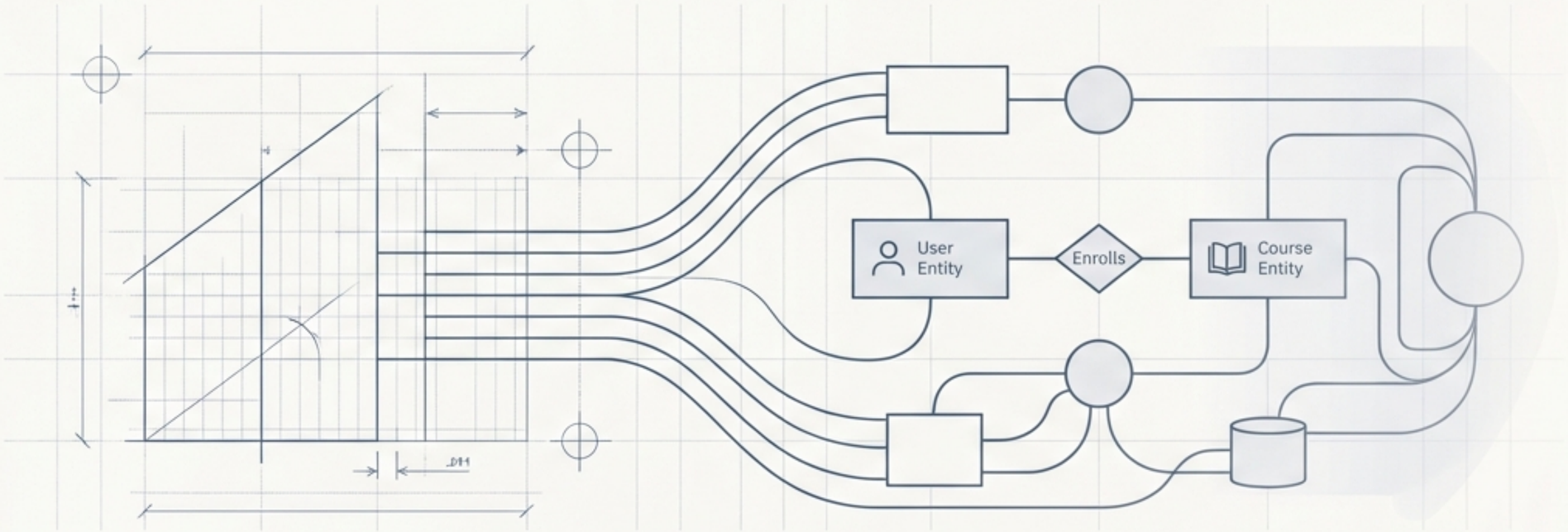
การออกแบบฐานข้อมูล: สร้างพิมพ์เขียวข้อมูลด้วย E-R Model

เปลี่ยนข้อมูลที่ซับซ้อนให้เป็นโครงสร้างที่ชัดเจนและเข้าใจง่าย

คู่มือสำหรับผู้เริ่มต้นทำความเข้าใจ Entity-Relationship Model
ตั้งแต่แนวคิดพื้นฐานสู่การสร้างแผนภาพ (ERD) ที่ใช้งานได้จริง



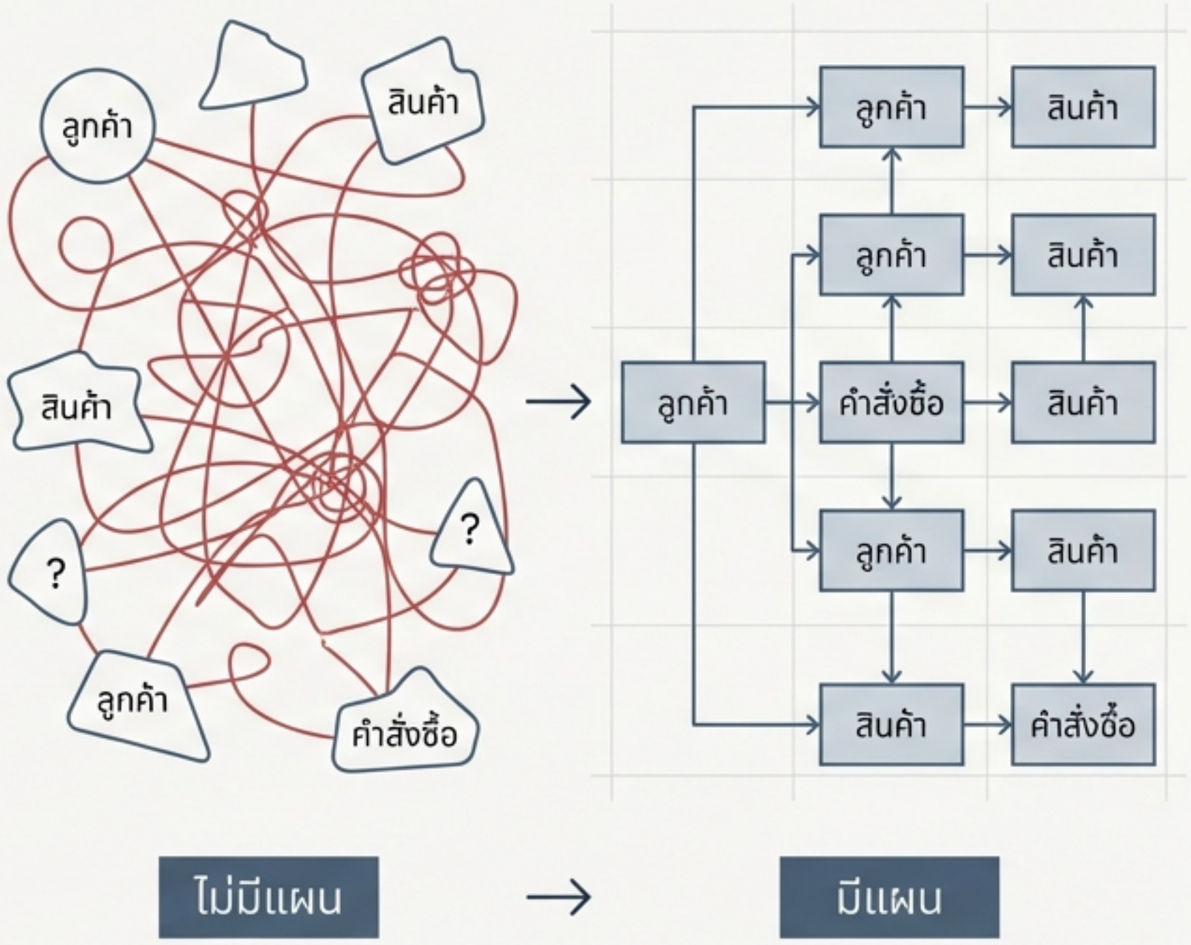
NotebookLM

ทำไมเราต้อง "ออกแบบ" ฐานข้อมูล?

เมื่อต้องจัดการกับข้อมูลจำนวนมาก เช่น ข้อมูลนักศึกษา, สินค้าในคลัง, หรือประวัติการสั่งซื้อของลูกค้า...

- เราจะจัดเก็บข้อมูลทั้งหมดอย่างไรไม่ให้ซ้ำซ้อน?
- เราจะแน่ใจได้อย่างไรว่าข้อมูลแต่ละส่วนเชื่อมโยงกันอย่างถูกต้อง?
- เราจะสร้างระบบที่ขยายขนาดในอนาคตได้อย่างไร?

หากไม่มีแผนการที่ดี ระบบข้อมูลของเราจะซับซ้อน สับสน และยากต่อการบำรุงรักษา เหมือนการสร้างบ้านโดยไม่มีพิมพ์เขียว



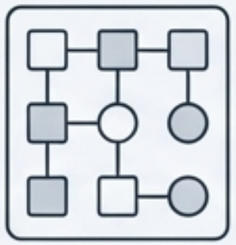
NotebookLM

เครื่องมือของสถาปนิกข้อมูล: E-R Model

E-R Model (Entity-Relationship Model)
คือ แบบจำลองที่ใช้อธิบายโครงสร้างของฐานข้อมูลในลักษณะของรูปภาพ ทำให้เราเห็นภาพรวมทั้งหมด

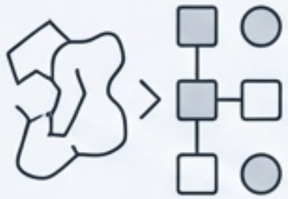
แผนภาพ ที่เราใช้วาดเรียกว่า
ERD (Entity-Relationship Diagram)

เปรียบเสมือน **"พิมพ์เขียว"** ที่ช่วยให้เราวางแผนก่อนลงมือสร้างฐานข้อมูลจริง



จัดระเบียบความคิด

ช่วยให้นักออกแบบมองเห็นภาพรวมของข้อมูลและจัดโครงสร้างได้อย่างเป็นระบบ



ลดความซับซ้อน

ทำให้เห็นส่วนประกอบหลักและความสัมพันธ์ของข้อมูลได้ชัดเจน

NotebookLM

ภาษาสากลของพิมพ์เขียวข้อมูล: สัญลักษณ์ใน ERD

ในการสร้าง ERD เราจะใช้สัญลักษณ์มาตรฐาน (ตามแบบ Chen Model) เพื่อแทนส่วนประกอบต่างๆ ของฐานข้อมูล

เอนทิตี้ (Entity Set) -
สิ่งที่เราสนใจจะเก็บข้อมูล

เอนทิตี้อ่อนแอ (Weak Entity Set)

ความสัมพันธ์ (Relationship Set) -
การเชื่อมโยงระหว่างเอนทิตี้

แอททริบิวต์ (Attribute) -
คุณสมบัติหรือรายละเอียดของเอนทิตี้

Key Attribute -
แอททริบิวต์ที่เป็นคีย์หลัก

Multi-valued Attribute -
แอททริบิวต์ที่มีได้หลายค่า

Derived Attribute -
แอททริบิวต์ที่คำนวณมาจากค่าอื่น

NotebookLM

ส่วนประกอบที่ 1: เอนทิตี้ (Entity) - "สิ่ง" ที่เราต้องการจัดเก็บ

เอนทิตี้ (Entity) คือ สิ่งต่างๆ ที่เราสนใจและต้องการเก็บข้อมูลไว้ในระบบ ซึ่งมักจะเป็นคำนาม

บุคคล
- พนักงาน
- นักศึกษา
- ลูกค้า

สถานที่
- จังหวัด
- ที่อยู่

วัตถุ
- สินค้า
- รถยนต์
- อาคาร

เหตุการณ์
- การลงทะเบียน
- การสั่งซื้อ
- ประวัติการรักษา

ใน ERD, เราใช้สัญลักษณ์ **สี่เหลี่ยมผืนผ้า** และมีชื่อเอนทิตี้อยู่ข้างใน

NotebookLM

ประเภทของเอนทิตี้: แบบแข็งแรง (Strong) และ อ่อนแอ (Weak)

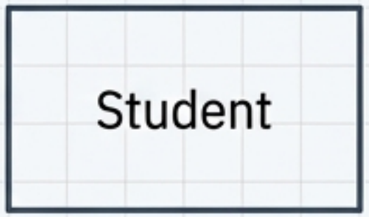
เอนทิตี้แบ่งได้เป็น 2 ชนิดหลัก:

1. เอนทิตี้ปกติ (Regular/Strong Entity)

เอนทิตี้ทั่วไปที่สามารถระบุสมาชิกแต่ละตัวได้ด้วยแอททริบิวต์ของตัวเอง (มี Primary Key ของตัวเอง)

ตัวอย่าง: เอนทิตี้ Student สามารถใช้ รหัสนักศึกษา (SID) เพื่อระบุตัวตนนักศึกษาแต่ละคนได้

สัญลักษณ์: สี่เหลี่ยมผืนผ้าเส้นเดี่ยว

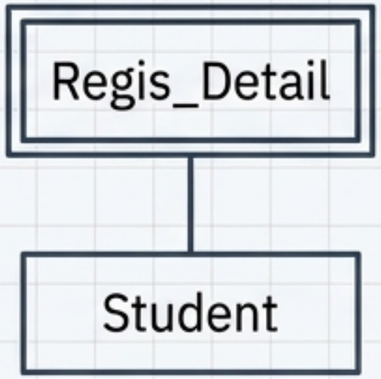


2. เอนทิตี้อ่อนแอ (Weak Entity)

เอนทิตี้ที่ต้องอาศัยแอททริบิวต์จากเอนทิตี้อื่นมาช่วยในการระบุตัวตน เพราะแอททริบิวต์ของตัวเองอย่างเดียวไม่เพียงพอ

ตัวอย่าง: เอนทิตี้ Regis_Detail (รายละเอียดการลงทะเบียน) จะไม่มีความหมายถ้าไม่รู้ว่าเป็นการลงทะเบียนของ Student คนไหน

สัญลักษณ์: สี่เหลี่ยมผืนผ้าสองชั้น



NotebookLM

ส่วนประกอบที่ 2: แอททริบิวต์ (Attribute) - "รายละเอียด" ของแต่ละสิ่ง

แอททริบิวต์ (Attribute) คือ คุณสมบัติหรือลักษณะเฉพาะของเอนทิตี้แต่ละตัว เป็นส่วนที่ใช้เก็บข้อมูลจริงๆ

ตัวอย่าง: เอนทิตี้ พนักงาน อาจมีแอททริบิวต์ได้แก่:

- รหัสพนักงาน
- ชื่อ-นามสกุล
- เพศ
- เงินเดือน

ใน ERD, เราใช้สัญลักษณ์ **วงรี** ที่มีเส้นเชื่อมต่อไปยังเอนทิตี้ของมัน

NotebookLM

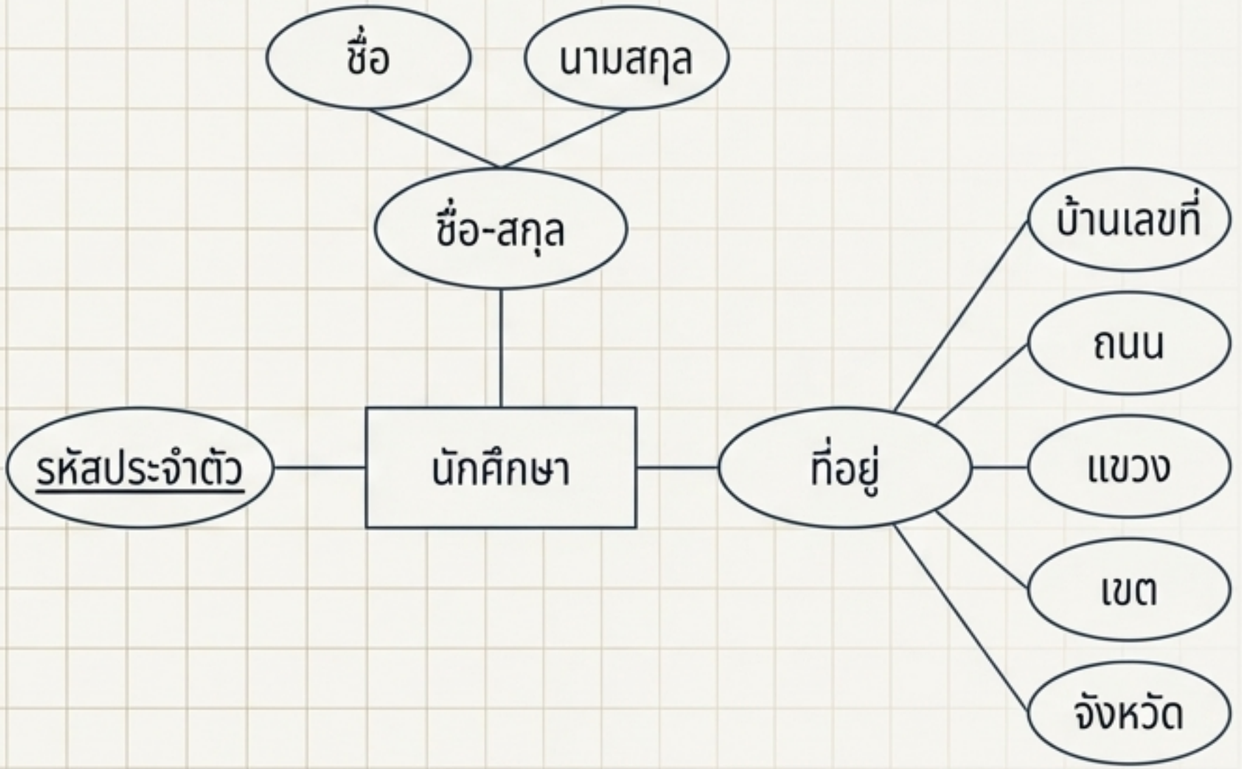
รู้จักชนิดของแอททริบิวต์ (ส่วนที่ 1)

1. Simple Attribute:
แอททริบิวต์ที่ไม่สามารถแบ่งย่อยได้อีก เช่น `เพศ', `เงินเดือน`

2. Composite Attribute:
แอททริบิวต์ที่สามารถแบ่งเป็นส่วนย่อยๆ ได้
ตัวอย่าง: `ชื่อ-สกุล` สามารถแบ่งเป็น `ชื่อ` และ `นามสกุล`

3. Key Attribute (Primary Key):
แอททริบิวต์ที่มีค่าไม่ซ้ำกันในแต่ละแถวข้อมูล
ใช้เพื่อระบุเอกลักษณ์ของเอนทิตี้นั้นๆ
ตัวอย่าง: `รหัสประจำตัวนักศึกษา`

สัญลักษณ์: มีการขีดเส้นใต้ชื่อแอททริบิวต์

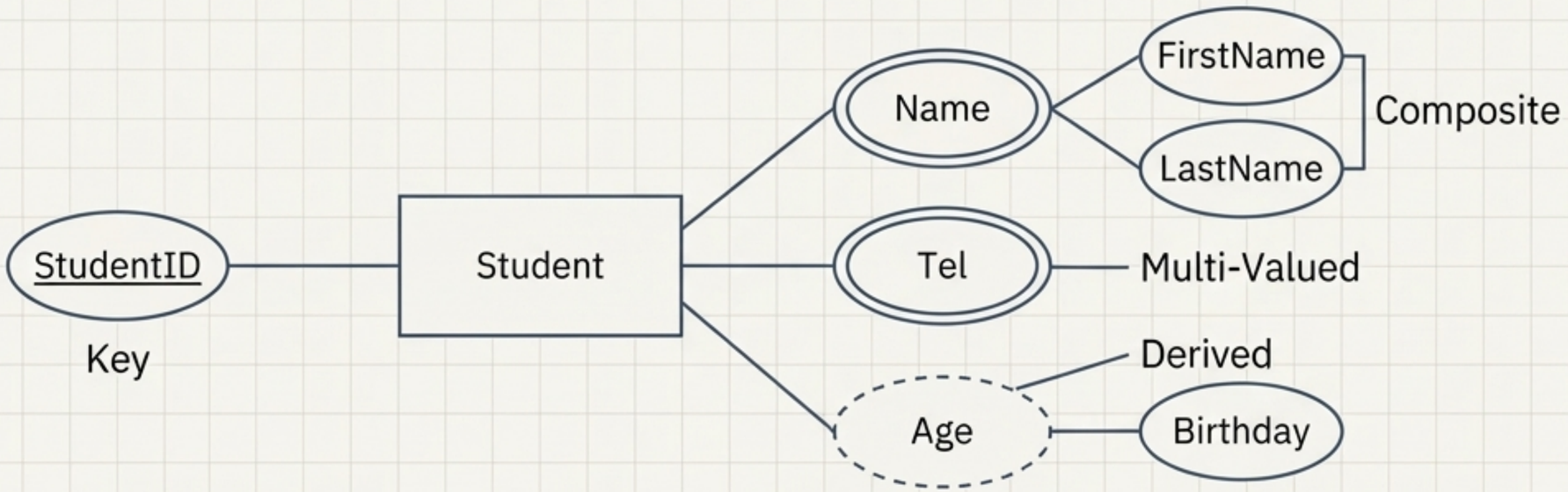


NotebookLM

รู้จักชนิดของแอททริบิวต์ (ส่วนที่ 2)

4. **Single-Valued Attribute:** แอททริบิวต์ที่เก็บข้อมูลได้เพียงค่าเดียว เช่น ‘รหัสลูกค้า’ (ลูกค้าหนึ่งคนมีรหัสได้แค่รหัสเดียว)
5. **Multi-Valued Attribute:** แอททริบิวต์ที่สามารถเก็บข้อมูลได้หลายค่า
    - *ตัวอย่าง:** ‘เบอร์โทรศัพท์’ (คนหนึ่งคนอาจมีหลายเบอร์)
    - *สัญลักษณ์:* วงรีสองชั้น
6. **Derived Attribute:** แอททริบิวต์ที่ค่าของมันได้มาจากการคำนวณจากแอททริบิวต์อื่น
    - *ตัวอย่าง:** ‘อายุ’ (สามารถคำนวณได้จาก ‘วันเกิด’)
    - *สัญลักษณ์:* วงรีเส้นประ

ภาพรวมตัวอย่าง:



NotebookLM

ส่วนประกอบที่ 3: ความสัมพันธ์ (Relationship) - ‘การเชื่อมโยง’ ระหว่างเอนทิตี้

ความสัมพันธ์ (Relationship) คือ การกระทำหรือความเกี่ยวข้องระหว่าง 2 เอนทิตี้ขึ้นไป

- ใช้เพื่อแสดงว่าเอนทิตี้ต่างๆ ในระบบของเราเกี่ยวข้องกันอย่างไร
- ใน ERD, เราใช้สัญลักษณ์ **สี่เหลี่ยมข้าวหลามตัด (Diamond)** และมีคำอธิบายความสัมพันธ์อยู่ข้างใน

ตัวอย่าง:

- **พนักงาน (Employee)** `ทำงานใน (Work_in)` **แผนก (Department)**
- **นักศึกษา (Student)** `ลงทะเบียน (Enrolls_in)` **วิชา (Subject)**

NotebookLM

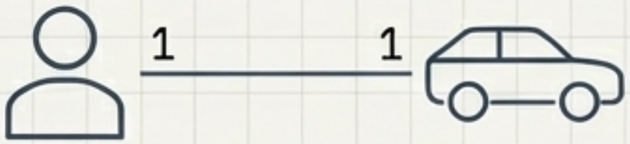
ระดับความสัมพันธ์ (Cardinality): เชื่อมโยงกันแบบไหน?

เราต้องระบุจำนวนสมาชิกของแต่ละเอนทิตี้ที่สัมพันธ์กันได้ ซึ่งมี 3 ประเภทหลัก:

1. หนึ่งต่อหนึ่ง (One-to-One / 1:1)

สมาชิกในเอนทิตี้หนึ่ง สัมพันธ์กับสมาชิกในอีกเอนทิตี้หนึ่งได้ **เพียงหนึ่งเดียว**

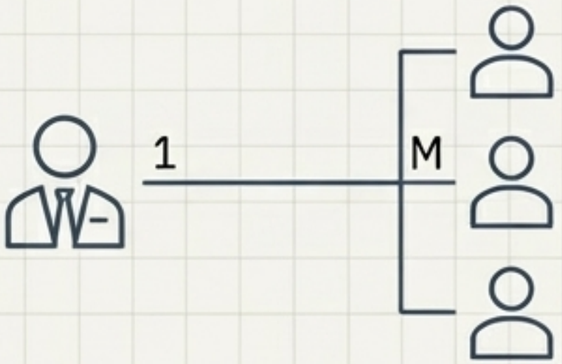
ตัวอย่าง:
พนักงาน (Employee) 1 **คน** `**ขับ** (Drives)` **รถบริษัท (Car)** **ได้** 1 **คัน**



2. หนึ่งต่อกลุ่ม (One-to-Many / 1:M)

สมาชิกในเอนทิตี้หนึ่ง สัมพันธ์กับสมาชิกในอีกเอนทิตี้หนึ่งได้ **หลายรายการ**

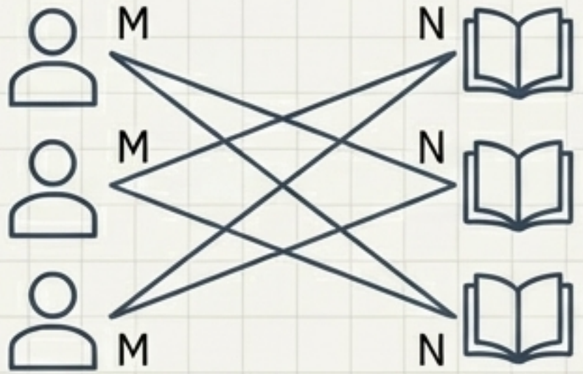
ตัวอย่าง:
ตัวแทน (Agent) 1 **คน** `**ดูแล** (Manages)` **ลูกค้า (Customer)** **ได้หลายคน**



3. กลุ่มต่อกลุ่ม (Many-to-Many / M:N)

สมาชิกในเอนทิตี้หนึ่ง **หลายรายการ** สัมพันธ์กับสมาชิกในอีกเอนทิตี้หนึ่งได้ **หลายรายการ**

ตัวอย่าง:
นักศึกษา (Student) หลายคน `**ลงทะเบียน** (Enrolls_in)` **วิชา (Subject) ได้หลายวิชา**



NotebookLM

กระบวนการสร้างพิมพ์เขียวข้อมูล (ERD) ทีละขั้นตอน

การออกแบบฐานข้อมูลด้วย E-R Model มีขั้นตอนที่เป็นระบบดังนี้:

1

ศึกษารายละเอียดของระบบ

ทำความเข้าใจการทำงาน, ขั้นตอน, และข้อมูลที่เกี่ยวข้องทั้งหมด

2

กำหนดเอนทิตี้ (Entity)

ระบุว่ามี "สิ่ง" อะไรบ้างที่เราต้องจัดเก็บข้อมูล ทั้งแบบ Strong และ Weak

3

กำหนดความสัมพันธ์ (Relationship)

หาความเชื่อมโยงระหว่างเอนทิตี้ต่างๆ และระบุประเภทความสัมพันธ์ (1:1, 1:M, M:N)

4

กำหนดคุณลักษณะ (Attributes)

ใส่รายละเอียดให้กับแต่ละเอนทิตี้ และพิจารณาชนิดของแอททริบิวต์ (เช่น Composite, Derived)

5

กำหนดคีย์หลัก (Primary Key)

เลือก Key Attribute ที่เหมาะสมสำหรับแต่ละเอนทิตี้ เพื่อให้สามารถอ้างอิงข้อมูลได้อย่างไม่ซ้ำซ้อน

NotebookLM

กรณีศึกษา: การออกแบบฐานข้อมูลสถาบันการศึกษา

โจทย์: สถาบันแห่งหนึ่งมี 3 คณะ แต่ละคณะมีหลายภาควิชา, แต่ละภาควิชามีอาจารย์หลายคน, อาจารย์หนึ่งคนสังกัดได้ภาควิชาเดียวแต่สอนได้หลายวิชา

การออกแบบตามขั้นตอน:

1. (ศึกษาโจทย์)

2. กำหนด Entities:
- คณะ (Faculty)
- ภาควิชา (Department)
- อาจารย์ (Teacher)
- วิชา (Subject)

3. กำหนด Relationships:
- Faculty `มี (Under)` Department (1:M)
- Department `มี (Under)` Teacher (1:M)
- Teacher `สอน (Teaching)` Subject (M:M)

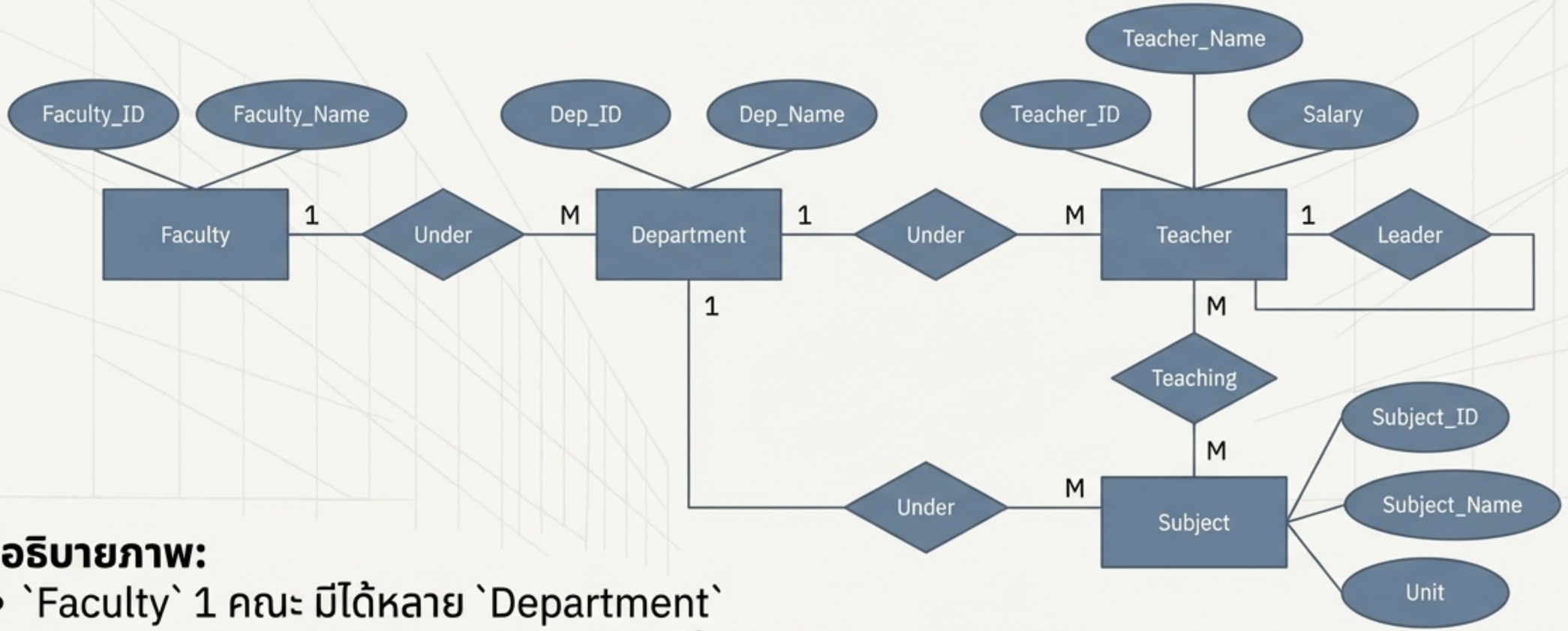
4. กำหนด Attributes:
- **Faculty:** Faculty_ID, Faculty_Name
- **Department:** Dep_ID, Dep_Name
- **Teacher:** Teacher_ID, Teacher_Name, Salary
- **Subject:** Subject_ID, Subject_Name, Unit

5. กำหนด Primary Keys:
Faculty_ID, Dep_ID, Teacher_ID, Subject_ID

NotebookLM

พิมพ์เขียวฉบับสมบูรณ์: ERD ระบบสถาบันการศึกษา

เมื่อนำส่วนประกอบทั้งหมดจากขั้นตอนการออกแบบมารวมกัน เราจะได้แผนภาพ ERD ที่แสดงโครงสร้างแ
ละความสัมพันธ์ของข้อมูลทั้งหมดในระบบ



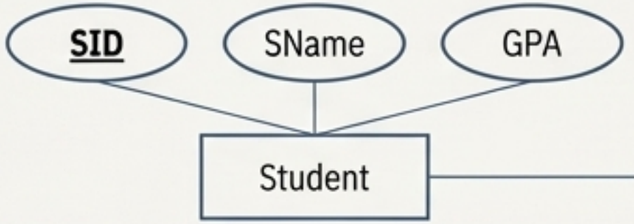
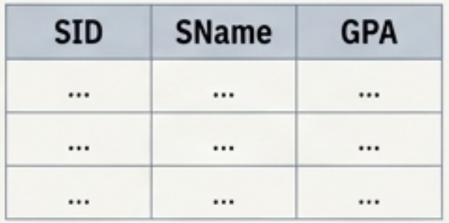
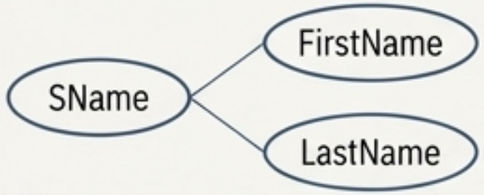
คำอธิบายภาพ:

- `Faculty` 1 คณะ มีได้หลาย `Department`
- `Department` 1 ภาควิชา มี `Teacher` ได้หลายคน
- `Teacher` และ `Subject` มีความสัมพันธ์แบบ M:M ผ่าน Relationship `Teaching`

NotebookLM

จากพิมพ์เขียวสู่โครงสร้างจริง: การแปลง ERD เป็นตารางข้อมูล

เมื่อออกแบบ ERD เสร็จสิ้น ขั้นตอนต่อไปคือการแปลงแผนภาพให้เป็นตาราง (Table) ในฐานข้อมูลจริง

ERD Components (พิมพ์เขียว)	การแปลง (Transformation)	Database Tables (โครงสร้างจริง)
 **Strong Entity** → แปลงเป็น 1 ตาราง โดยมี Attributes เป็นคอลัมน์		 เอนทิตี้ `Student` (SID, SName, GPA) → ตาราง Student (คอลัมน์ SID, SName, GPA)
 **Composite Attribute** → แอททริบิวต์ย่อยจะกลายเป็นคอลัมน์แยกกัน		`SName` (FirstName, LastName) → คอลัมน์ FirstName, LastName
Multi-valued Attribute → ต้องสร้างตารางใหม่แยกต่างหากสำหรับเก็บค่า		แอททริบิวต์ `Tel` ของ `Student` → สร้างตาราง Tel (คอลัมน์ SID, Tel)

สรุป: ERD คือเครื่องมือสำคัญที่ช่วยให้เรามั่นใจได้ว่าฐานข้อมูลที่เราสร้างขึ้นจะมีโครงสร้างที่ดี เป็นระเบียบ และพร้อมใช้งาน

NotebookLM