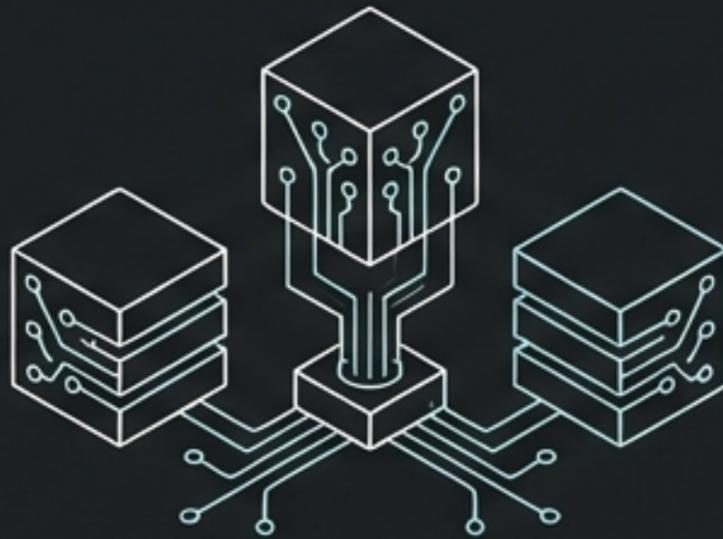


Database Fundamentals for Work

ฐานข้อมูลกับการใช้งานจริง: ออกแบบ ERD, สร้างตาราง และ SQL เบื้องต้น



Database Fundamentals for Work

ฐานข้อมูลกับการใช้งานจริง: ออกแบบ ERD, สร้างตาราง และ SQL เบื้องต้น

Prompt

โดย อาจารย์ชัยณรงค์ เหมือนรุ่ง

The Problem (Excel/Paper)



Prompt

- ข้อมูลซ้ำซ้อน (Redundancy): ต้องพิมพ์ข้อมูลซ้ำๆ
- หาข้อมูลยาก (Hard to Search): ต้อง Scroll หาเอง
- ข้อมูลขัดแย้ง (Inconsistency): ชื่อเดียวกันสะกดต่างกัน
- แก้ไขพร้อมกันไม่ได้ (No Concurrency): ไฟล์ถูกล็อก

The Solution (Database)



Prompt

- เก็บครั้งเดียว อ้างอิงด้วย ID (Reference by ID)
- ค้นหาเจอใน 1 วินาทีด้วย SQL (Fast Search)
- มีระบบตรวจสอบความถูกต้อง (Constraints)
- รองรับการใช้งานพร้อมกัน (Concurrent Access)

Case Study: ระบบร้านกาแฟ (Coffee Shop System)

บริษัทที่เข้าใจง่าย เห็นภาพชัดเจน ครบทุกความสัมพันธ์



```
>_
```

ความสัมพันธ์แบบ 1:1 (One-to-One)

Concept:

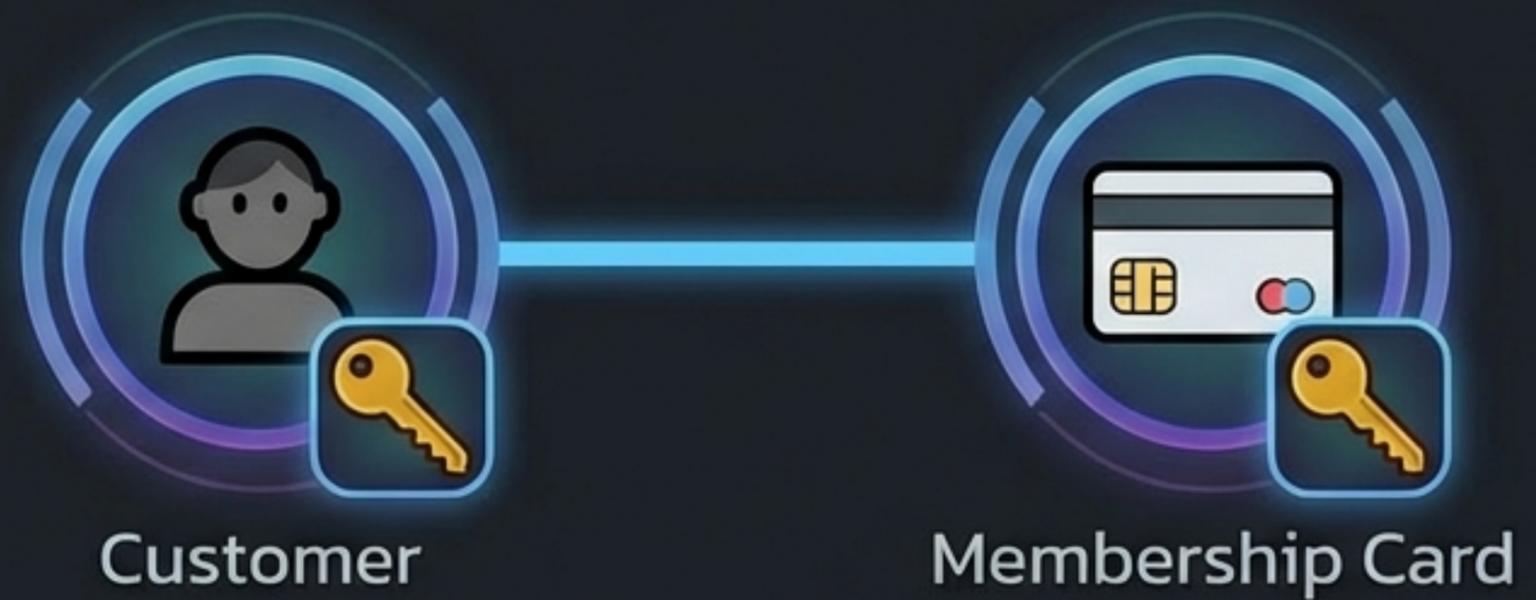
- 1 สิ่ง จับคู่กับอีก 1 สิ่งเท่านั้น (และกลับกัน)

Example:

- ลูกค้า 1 คน ↔ มี Membership Card 1 ใบ

Use Case:

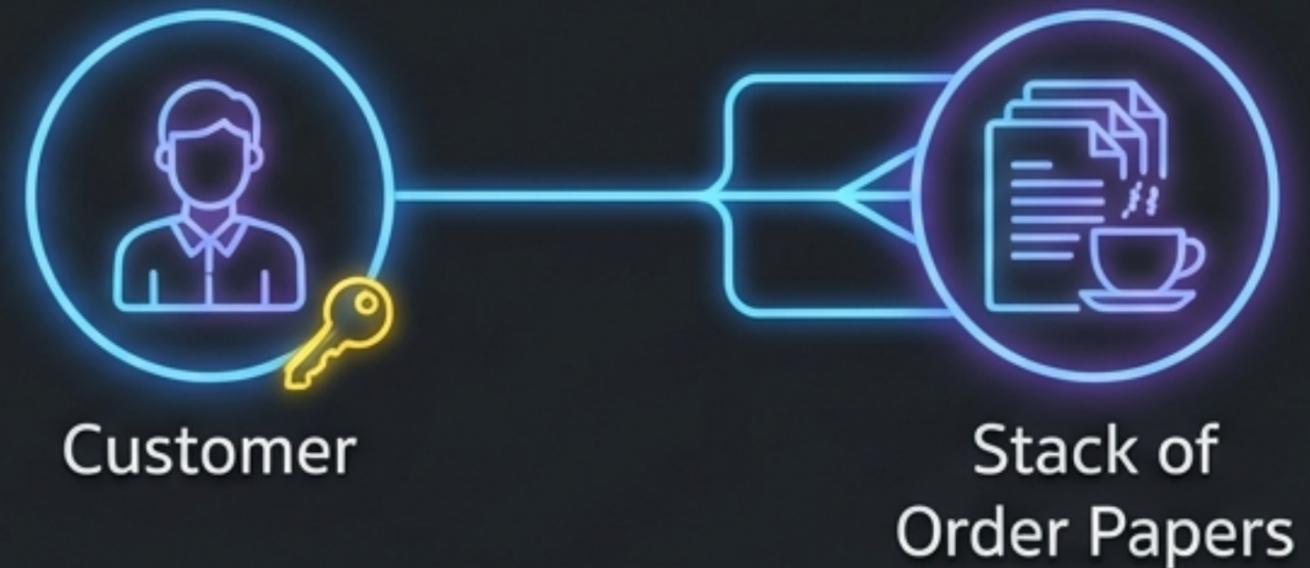
- ใช้สำหรับแยกข้อมูลที่ไม่ค่อยใช้, ข้อมูลลับ, หรือข้อมูล Optional



Technical Rule: Foreign Key (FK) อยู่ฝั่งไหนก็ได้

ความสัมพันธ์แบบ 1:M (One-to-Many)

- 🔑 **Concept:** 1 สิ่ง มีได้หลายสิ่ง แต่ปลายทางเป็นของเจ้าของเดียว
- 🔑 **Example:** ลูกค้า 1 คน (One) → สั่งได้หลาย Orders (Many)
- 🔑 **Tip:** พบบ่อยที่สุด! ถามตัวเองว่า “ใครเป็นเจ้าของ?”



****สำคัญมาก!**** Foreign Key (FK)
ต้องอยู่ฝั่ง 'Many' เสมอ
(ตาราง Orders เก็บ customer_id)

ความสัมพันธ์แบบ M:N (Many-to-Many)

Concept:

- หลายสิ่ง จับคู่กับ หลายสิ่ง

Example:

- Order 1 ใบ มีหลายเมนู ↔ เมนู 1 อย่าง ก็อยู่ในหลาย Order

Note:

- เชื่อมตรงๆ ไม่ได้! ต้องมีตารางกลาง (Intermediate Table)



ตารางกลางเก็บ FK ของทั้งสองฝั่ง

ERD Design Process (3 Steps)

Step 1: หา Entity (Nouns)

Prompt

ในระบบมี 'คำนาม' อะไรบ้างที่ต้องเก็บข้อมูล?

JetBrains Mono

Result:
Customers,
Products,
Orders



Step 2: หา Attributes (Adjectives)

Prompt

เราอยากรู้อะไรเกี่ยวกับมันบ้าง?

JetBrains Mono

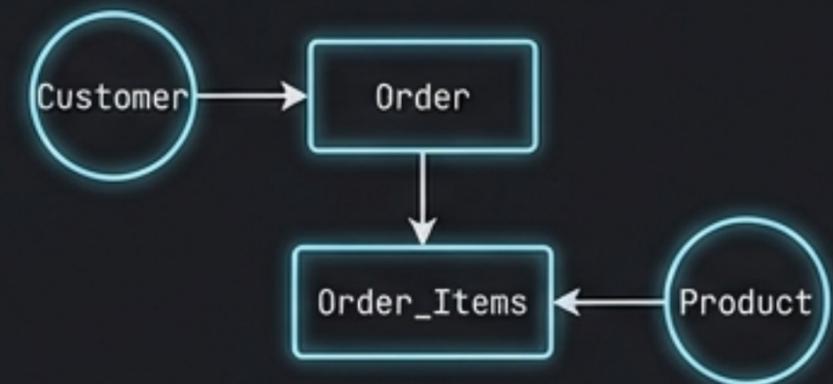
Result:
name,
price,
phone,
date + ****PK****



Step 3: วาดความสัมพันธ์

Prompt

ลากเส้นเชื่อมโยงและกำหนด FK



SQL Basics: CRUD & SELECT

Theory

SQL (Structured Query Language)
คือภาษาสำหรับการจัดการฐานข้อมูล

CRUD Concept:



- Create (สร้าง)
- Read (อ่าน) -> SELECT
- Update (แก้ไข)
- Delete (ลบ)

>_ SQL Terminal ×

-- ดูกี่ทั้งหมด

```
SELECT * FROM customers;
```

-- เลือกบางคอลัมน์

```
SELECT name, phone FROM customers;
```

-- กรองข้อมูล

```
SELECT * FROM products WHERE price < 60;
```

-- เรียงลำดับ

```
SELECT * FROM products ORDER BY price;
```

SQL Data Manipulation (Insert, Update, Delete)

INSERT (เพิ่มข้อมูลใหม่)

```
INSERT INTO customers (name, phone) VALUES ('สมชาย', '081...');
```

UPDATE (แก้ไขข้อมูล)

```
UPDATE customers SET phone = '089...'  
WHERE customer_id = 1;
```



ห้ามลืม WHERE
ไม่ขึ้นแก้ทั้งตาราง!

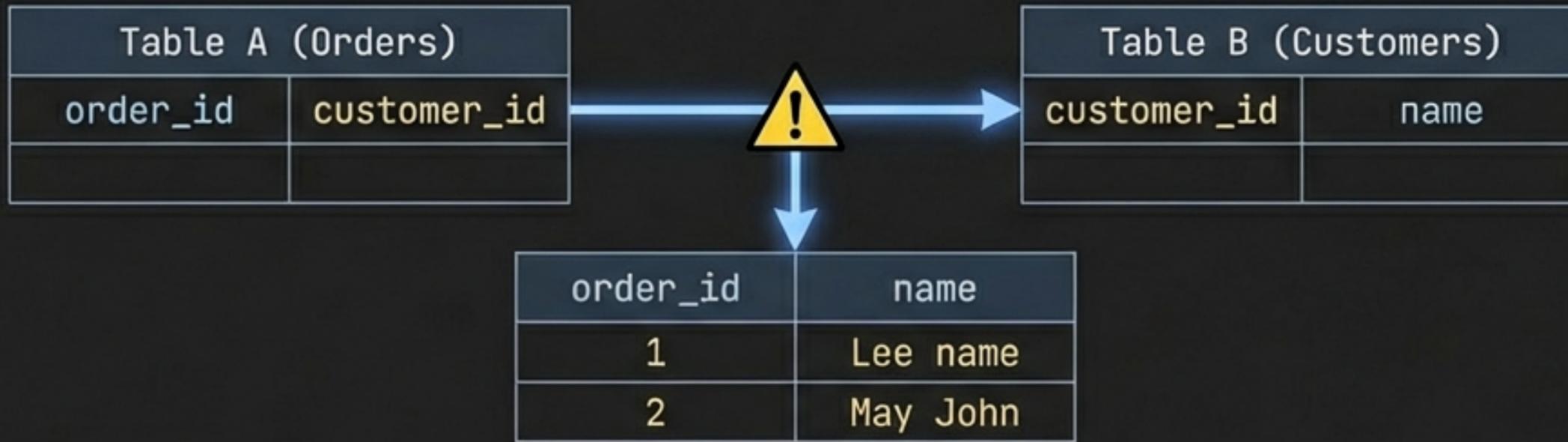
DELETE (ลบข้อมูล)

```
DELETE FROM customers WHERE customer_id = 1;
```



ห้ามลืม WHERE
ไม่ขึ้นหายทั้งตาราง!

SQL JOIN (Connecting Tables)



>_ SQL Query ×

```
SELECT orders.order_id, customers.name  
FROM orders  
JOIN customers ON orders.customer_id = customers.customer_id;
```



Concept: อยากเห็นชื่อลูกค้าในใบสั่งซื้อ (Customer Name in Orders)

Lab 1: Creating Database & Tables

lab1_create.sql

```
1  -- 1. สร้าง Database
2  CREATE DATABASE coffee_shop;
3
4  -- 2. สร้างตาราง Customers (Master Data)
5  CREATE TABLE customers (
6      customer_id INT AUTO_INCREMENT PRIMARY KEY,
7      name VARCHAR(100) NOT NULL,
8      phone VARCHAR(20)
9  );
10
11 -- 3. สร้างตาราง Products
12 CREATE TABLE products (... );
```

รับเลขเองอัตโนมัติ

ห้ามซ้ำ

ห้ามว่าง

Lab 2: Tables with Foreign Keys

lab2_foreign_keys.sql

```
1  -- สร้างตาราง Orders (มี FK)
2  CREATE TABLE orders (
3      order_id INT AUTO_INCREMENT PRIMARY KEY,
4      customer_id INT,
5      -- Foreign Key เชื่อมไปหาตาราง Customers
6      FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
7  );
8
9  -- สร้างตาราง Order_Items (ตารางกลาง M:N)
10 -- ต้องมี FK 2 ตัว เชื่อมไปหา Orders และ Products
```



Benefit: Foreign Key ช่วยป้องกันการบันทึกข้อมูลซ้ำ (Data Integrity)

Lab 3: Working with Data

```
lab3_working_with_data.sql × Input (SQL)
1  -- เพิ่มข้อมูล
2  INSERT INTO customers (name) VALUES ('สมชาย'), ('สมหญิง');
3  INSERT INTO products (name, price) VALUES ('Americano', 50);
4
5  -- โจทย์: ดู Order พร้อมชื่อลูกค้า
6  SELECT o.order_id, c.name
7  FROM orders o JOIN customers c ON o.customer_id = c.customer_id;
```

Output (Result)

order_id	name
101	สมชาย
102	สมหญิง

Challenge & Application

Challenge: ออกแบบระบบของคุณเอง

เลือก 1 หัวข้อ (ระบบลูกค้า, รายรับรายจ่าย, สต็อกสินค้า)
แล้ววาด ERD + สร้าง 2 ตาราง ✎

Modern Tools (No Server Required!)



AI Helper

ChatGPT, Claude
('ช่วยเขียน SQL ให้หน่อย')



Cloud SQL

Google Colab
+ SQLite



Google Sheets

ฟังก์ชัน QUERY / VLOOKUP



No-Code

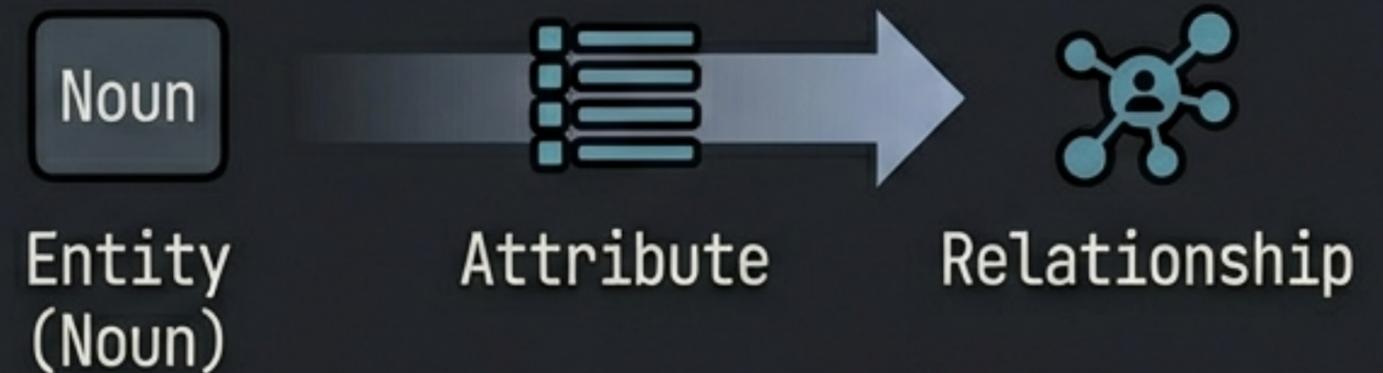
Airtable, Notion

Summary & Cheat Sheet

Relationships



Design Process



SQL Syntax Quick Ref

```
SELECT *  
FROM ...
```

```
INSERT INTO ...  
VALUES ...
```

```
UPDATE ...  
SET ...  
WHERE ...
```

```
DELETE FROM ...  
WHERE ...
```

```
JOIN ...  
ON ...
```