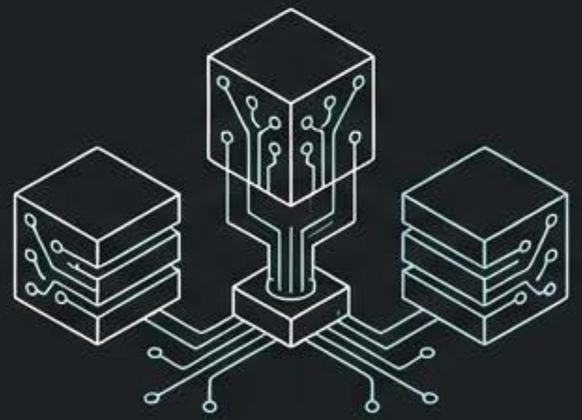
Database Fundamentals for Work

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): ต้องพิมพ์ชื่อลูกค้าซ้ำๆ
- 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)

บริบทที่เข้าใจง่าย เห็นภาพชัดเจน ครบทุกความสัมพันธ์

ลูกค้า
(Customers)

เมนู
(Products)

ออเดอร์
(Orders)

รายการสั่ง
(Order Items)

ความสัมพันธ์แบบ 1:1 (One-to-One)

Concept:

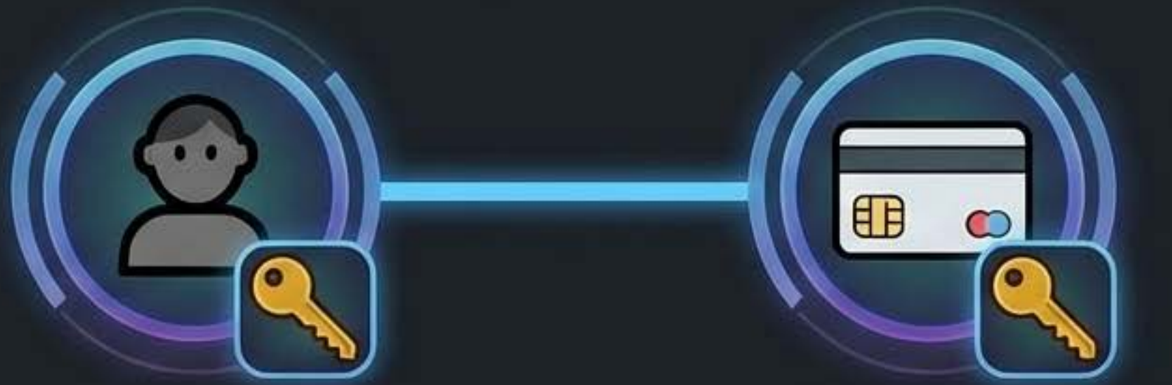
- 1 สิ่ง จับคู่กับอีก 1 สิ่งเท่านั้น (และกลับกัน)

Example:

- ลูกค้า 1 คน ↔ มี Membership Card 1 ใบ

Use Case:

- ใช้สำหรับแยกข้อมูลที่ไม่ค่อยใช้, ข้อมูลลับ, หรือข้อมูล Optional



`Technical Rule:` Foreign Key (FK) อยู่ฝั่งไหนก็ได้

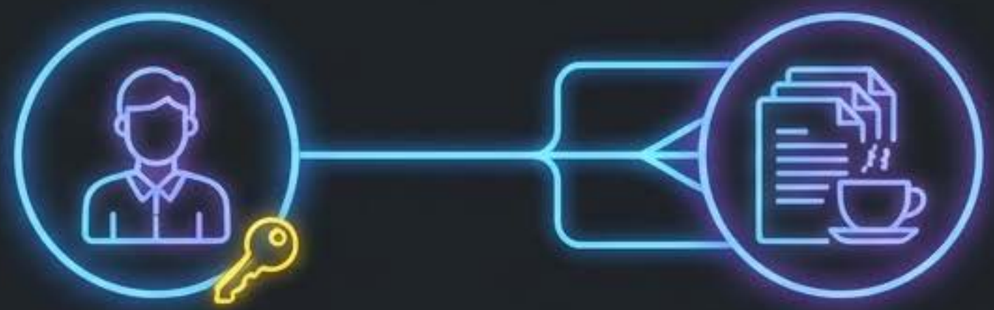
>_ Prompt

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)

>_ Prompt

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)

>_

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?

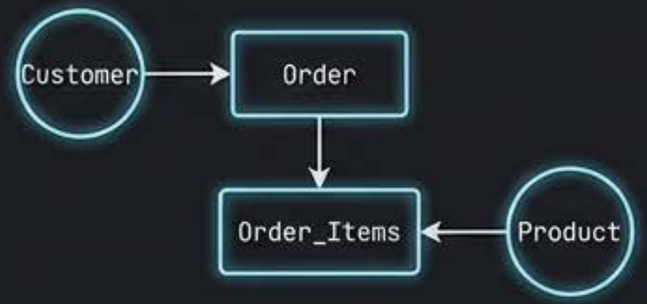
JetBrains Mono

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

Step 3: วาดความสัมพันธ์

Prompt

ลากเส้นเชื่อมโยงและกำหนด FK



SQL Basics: CRUD & SELECT

Theory

SQL (Structured Query Language) คือภาษาสำหรับจัดการฐานข้อมูล

CRUD Concept: 💡

- **C**reate (สร้าง)
- **R**ead (อ่าน) -> SELECT
- **U**pdate (แก้ไข)
- **D**elete (ลบ)

>_ SQL Terminal

```sql
-- ดูทั้งหมด
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)

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

UPDATE (แก้ไขข้อมูล)

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

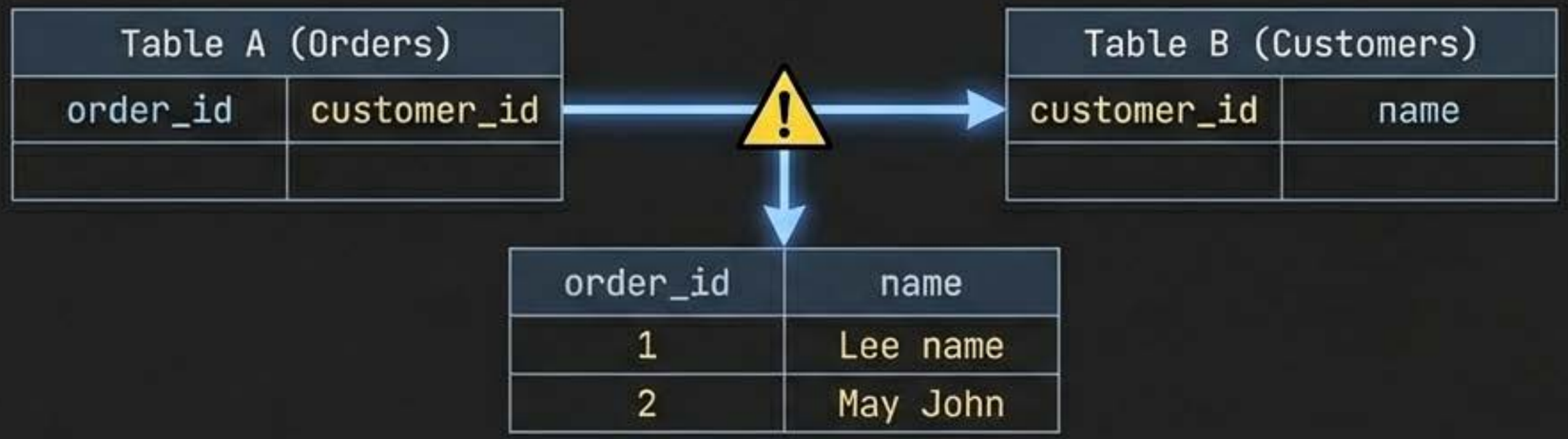
ห้ามลืม WHERE
ไม่งั้นแก้ทั้งตาราง!

DELETE (ลบข้อมูล)

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

ห้ามลืม WHERE
ไม่งั้นหายทั้งตาราง!

SQL JOIN (Connecting Tables)



>_ SQL Query

```sql
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

```sql
-- 1. สร้าง Database
CREATE DATABASE coffee_shop;

-- 2. สร้างตาราง Customers (Master Data)
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    phone VARCHAR(20)
);

-- 3. สร้างตาราง Products
CREATE TABLE products ( ... );
```

รันเลขเองอัตโนมัติ

ห้ามซ้ำ

ห้ามว่าง

Lab 2: Tables with Foreign Keys

lab2_foreign_keys.sql

```sql
-- สร้างตาราง Orders (มี FK)
CREATE TABLE orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    -- Foreign Key เชื่อมไปหาตาราง Customers
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

-- สร้างตาราง Order_Items (ตารางกลาง M:N)
-- ต้องมี FK 2 ตัว เชื่อมไปหา Orders และ Products
```

💡 **Benefit:** Foreign Key ช่วยป้องกันการบันทึกข้อมูลมั่ว (Data Integrity)

Lab 3: Working with Data

lab3_working_with_data.sql

Input (SQL)

```sql
-- เพิ่มข้อมูล
INSERT INTO customers (name) VALUES ('สมชาย'), ('สมหญิง');
INSERT INTO products (name, price) VALUES ('Americano', 50);

-- โจทย์: ดู Order พร้อมชื่อลูกค้า
SELECT o.order_id, c.name
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, สต๊อกสินค้า) 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 ...
```