

## แบบฝึกหัดบทที่ 2

### 1.1) อธิบายความสำคัญของ Data Modeling

การทำแบบจำลองข้อมูล (Data Modeling) คือ การสร้างแบบจำลองที่แสดงถึงโครงสร้างและความสัมพันธ์ของข้อมูลในระบบฐานข้อมูล

แบบจำลองมี 3 ประเภท ได้แก่:

1. **Conceptual Data Model (CDM):** แสดงความสัมพันธ์ของข้อมูลในระดับสูง แสดงถึงความต้องการทางธุรกิจ โดยยังไม่คำนึงถึงโครงสร้างที่ใช้จัดเก็บข้อมูล
2. **Logical Data Model (LDM):** แสดงการจัดระเบียบและโครงสร้างของข้อมูล แสดงรายละเอียดของข้อมูลในระดับที่ลึกขึ้น เช่น ชนิดข้อมูล ความสัมพันธ์ระหว่างตาราง แต่ยังไม่คำนึงถึงเทคโนโลยีที่ใช้ในการจัดเก็บข้อมูล
3. **Physical Data Model (PDM):** แสดงโครงสร้างข้อมูลในระดับที่ลึกที่สุด รวมถึงการกำหนดตาราง ชนิดข้อมูล ข้อจำกัดต่าง ๆ ที่ใช้ในการจัดเก็บข้อมูลในระบบฐานข้อมูลจริง โดยจะมีความแตกต่างกันไปตามเทคโนโลยีที่ใช้ เช่น Oracle, MySQL, PostgreSQL เป็นต้น

การทำแบบจำลองข้อมูลจะช่วยให้ผู้พัฒนาระบบสามารถเข้าใจความต้องการของธุรกิจ และออกแบบฐานข้อมูลที่ตอบสนองต่อความต้องการเหล่านั้นได้อย่างมีประสิทธิภาพและถูกต้อง นอกจากนี้ยังช่วยให้การสื่อสารระหว่างผู้พัฒนาระบบและผู้ใช้งานเป็นไปได้อย่างราบรื่น เนื่องจากมีแบบจำลองที่ชัดเจนในการอธิบายโครงสร้างและความสัมพันธ์ของข้อมูล

### 1.2) อธิบายลักษณะพื้นฐานของ Relational Data Model และความสำคัญของลักษณะเหล่านี้ที่มีต่อผู้ใช้และผู้ออกแบบฐานข้อมูล

Relational Data Model (RDM) หรือ โมเดลข้อมูลเชิงสัมพันธ์ เป็นโมเดลข้อมูลที่ใช้ในการจัดเก็บและจัดการข้อมูลในรูปแบบของสัมพันธ์ระหว่างข้อมูล (relation) หรือ ตาราง (tables)

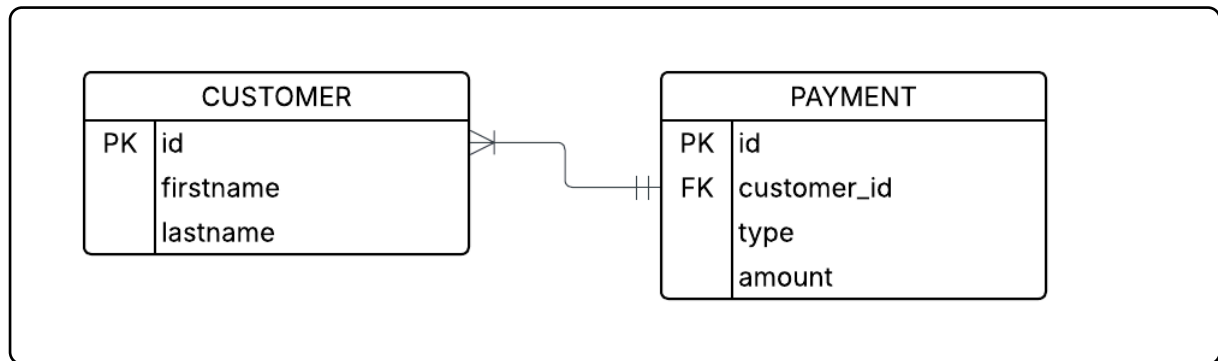
มีลักษณะพื้นฐานคือ:

- มีโครงสร้างข้อมูลที่เป็นตาราง (tables) ซึ่งประกอบด้วยแถว (rows) และคอลัมน์ (columns)
- รองรับรูปแบบความสัมพันธ์ระหว่างข้อมูลทั้งแบบ 1:1, 1:M และ M:N
- รองรับการใช้คีย์หลัก (primary key) และคีย์ต่างประเทศ (foreign key) เพื่อสร้างความสัมพันธ์ระหว่างตาราง
- มีการควบคุมความถูกต้องของข้อมูล (data integrity)
- ใช้ SQL (Structured Query Language) ในการเข้าถึงและจัดการข้อมูล

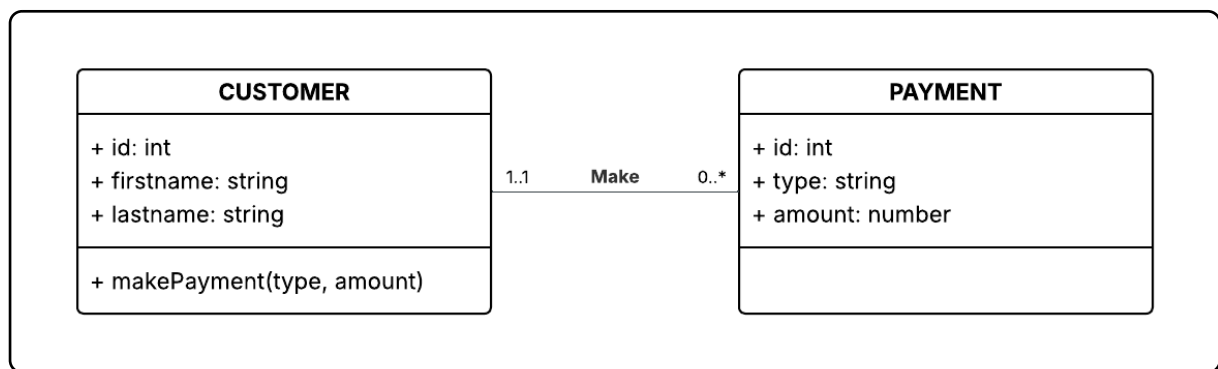
ความสำคัญต่อผู้ใช้และผู้ออกแบบฐานข้อมูล:

- ช่วยให้ผู้ใช้สามารถเข้าถึงและจัดการข้อมูลได้อย่างมีประสิทธิภาพ โดยไม่ต้องรู้รายละเอียดของการจัดเก็บข้อมูล
- ช่วยให้ผู้ออกแบบฐานข้อมูลสามารถออกแบบโครงสร้างข้อมูลที่เหมาะสมกับความต้องการของธุรกิจ และสามารถปรับเปลี่ยนโครงสร้างได้ง่ายเมื่อมีการเปลี่ยนแปลงความต้องการ
- ช่วยให้การจัดการข้อมูลมีความยืดหยุ่นและสามารถปรับขนาดได้ตามความต้องการของระบบ

**1.3) รว ER จาก “A customer can make many payments.” แต่จะการชำระเงินมาจากลูกค้าคนเดียว**



**1.4) รว OO Model จาก ข้อ 3.**



### 1.5) อธิบายความแตกต่างของ Object และคลาส ใน Object-oriented Data Model (OODM)

- **Class** คือ คำอธิบายหรือแม่แบบ (blueprint) ของวัตถุ (object) ที่มีคุณสมบัติ (attribute/property) และพฤติกรรม (behavior/method) ที่เหมือนกัน เช่น คลาส "รถยนต์" อาจมีคุณสมบัติ เช่น ยี่ห้อ, รุ่น, สี และพฤติกรรม เช่น ขับเคลื่อน, หยุด
- **Object** คือ อินสแตนซ์ (instance) ของคลาส ซึ่งเป็นวัตถุที่มีคุณสมบัติและพฤติกรรมตามที่กำหนดในคลาส โดยแต่ละวัตถุจะมีค่าของคุณสมบัติที่แตกต่างกัน เช่น วัตถุ "รถยนต์1" อาจมีคุณสมบัติ ยี่ห้อ = "Toyota", รุ่น = "Corolla", สี = "แดง"

### 1.6) อะไรคือ relationship และอธิบายชนิดของ relationship พร้อมยกตัวอย่าง

Relationship คือ ความสัมพันธ์ระหว่างเอนทิตี (entity) ในฐานข้อมูล ซึ่งแสดงถึงการเชื่อมโยงข้อมูลระหว่างตารางต่าง ๆ ในระบบฐานข้อมูล

ชนิดของ relationship มี 3 ประเภทหลัก ได้แก่:

1. **One-to-One (1:1):** ความสัมพันธ์ระหว่างเอนทิตีสองตัวที่แต่ละตัวมีความสัมพันธ์กับอีกตัวหนึ่งเพียงตัวเดียว เช่น หนึ่งบุคคลมีหนึ่งบัตรประชาชน
2. **One-to-Many (1:N):** ความสัมพันธ์ระหว่างเอนทิตีหนึ่งตัวที่มีความสัมพันธ์กับหลายตัว เช่น หนึ่งลูกค้าสามารถทำการสั่งซื้อได้หลายรายการ
3. **Many-to-Many (M:N):** ความสัมพันธ์ระหว่างเอนทิตีสองตัวที่แต่ละตัวสามารถมีความสัมพันธ์กับอีกตัวหนึ่งได้หลายตัว เช่น หลักสูตรหนึ่งสามารถมีนักเรียนหลายคนลงทะเบียน และนักเรียนหนึ่งคนสามารถลงทะเบียนในหลักสูตรหลายหลักสูตร

### 1.7) ตารางคืออะไร และมีบทบาทสำคัญอย่างไรใน relational data model

ตาราง (table) คือ โครงสร้างพื้นฐานที่ใช้ในการจัดเก็บข้อมูลใน Relational Data Model โดยข้อมูลจะถูกจัดเก็บในรูปแบบของแถว (rows) และคอลัมน์ (columns)

- **ตาราง (Tables):** ข้อมูลจะถูกจัดเก็บในรูปแบบของตาราง ซึ่งประกอบด้วยแถว (rows) และคอลัมน์ (columns) โดยแต่ละตารางจะแทนข้อมูลของเอนทิตี (entity) หนึ่ง ๆ เช่น ตารางลูกค้า, ตารางสินค้า เป็นต้น
- **แถว (Rows):** แต่ละแถวในตารางจะแทนข้อมูลของเอนทิตีนั้น ๆ หนึ่งรายการ เช่น ข้อมูลของลูกค้าแต่ละคน ข้อมูลของสินค้าแต่ละรายการ เป็นต้น
- **คอลัมน์ (Columns):** แต่ละคอลัมน์ในตารางจะแทนคุณสมบัติ (attribute) ของเอนทิตี เช่น ชื่อ, ที่อยู่, เบอร์โทรศัพท์, ราคาสินค้า, จำนวนสินค้าคงเหลือ เป็นต้น

บทบาทสำคัญของตารางใน Relational Data Model คือ:

- จัดระเบียบข้อมูลให้อยู่ในรูปแบบที่เข้าใจง่าย
- ช่วยให้สามารถเข้าถึงและจัดการข้อมูลได้อย่างมีประสิทธิภาพ
- ช่วยให้สามารถสร้างความสัมพันธ์ระหว่างข้อมูลในตารางต่าง ๆ ได้อย่างชัดเจน

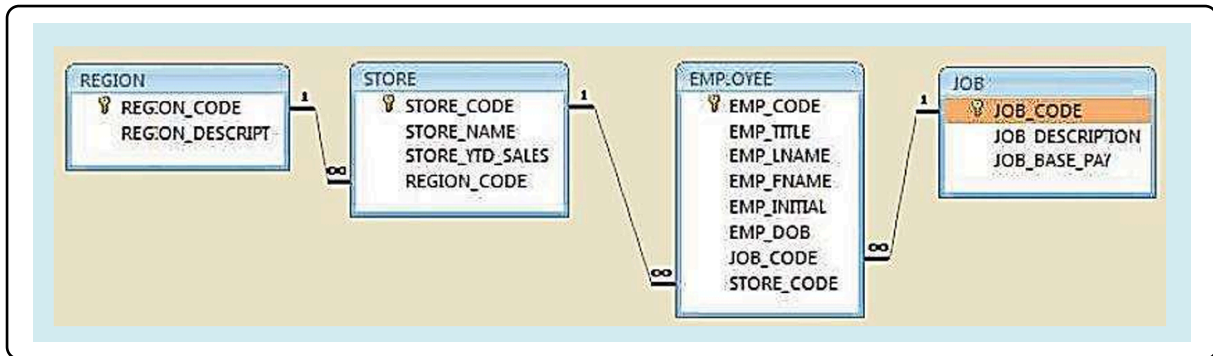
### 1.8) ER diagram คืออะไร จงยกตัวอย่างประกอบ

ER diagram (Entity-Relationship Diagram) คือ แผนภาพที่ใช้ในการแสดงความสัมพันธ์ระหว่างเอนทิตี (entity) และความสัมพันธ์ (relationship) ในฐานข้อมูล

ER diagram ประกอบด้วยองค์ประกอบหลัก 3 ประเภท ได้แก่ เอนทิตี (entity), แอตทริบิวต์ (attribute) และความสัมพันธ์ (relationship)

ตัวอย่าง:

- **เอนทิตี:** ลูกค้า (Customer), สินค้า (Product), การสั่งซื้อ (Order)
- **แอตทริบิวต์:** ชื่อลูกค้า (Customer Name), รหัสสินค้า (Product ID), วันที่สั่งซื้อ (Order Date)
- **ความสัมพันธ์:**
  - ▶ Customer 1:N Order (ลูกค้าสามารถทำการสั่งซื้อได้หลายรายการ โดยแต่ละรายการสั่งซื้อจะมีลูกค้าเพียงคนเดียว)
  - ▶ Order M:N Product (รายการสั่งซื้อสามารถมีสินค้าหลายรายการ และสินค้าหนึ่งรายการสามารถอยู่ได้ในหลายรายการสั่งซื้อ)



## 2.1) Identify each relationship type and write all of the business rules.

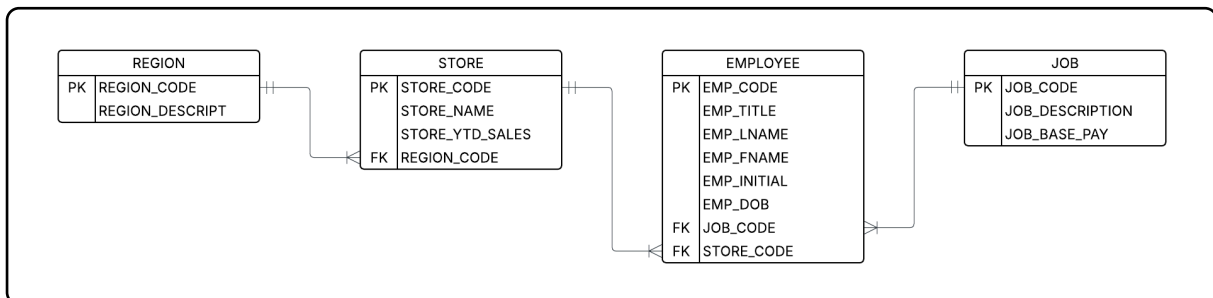
### Relationship Types:

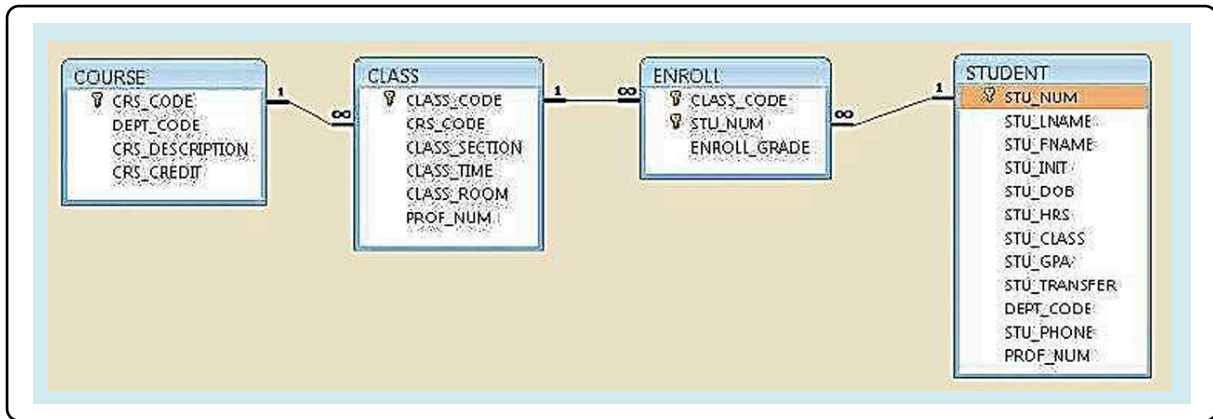
1. **REGION 1:N STORE:** One-to-Many (ภูมิภาคหนึ่งสามารถมีร้านค้าได้หลายร้าน)
2. **STORE 1:N EMPLOYEE:** One-to-Many (ร้านค้านึงสามารถจ้างพนักงานได้หลายคน)
3. **JOB 1:N EMPLOYEE:** One-to-Many (ตำแหน่งงานหนึ่งสามารถถูกมอบหมายให้พนักงานได้หลายคน)

### Business Rules:

1. ภูมิภาคหนึ่งสามารถมีร้านค้าหลายร้าน แต่แต่ละร้านจะอยู่ในภูมิภาคเดียวเท่านั้น
2. ร้านค้านึงสามารถมีพนักงานหลายคน แต่พนักงานแต่ละคนจะทำงานในร้านค้าเดียวเท่านั้น
3. พนักงานแต่ละคนจะได้รับมอบหมายตำแหน่งงานเพียงตำแหน่งเดียว แต่ตำแหน่งงานหนึ่งสามารถมีพนักงานได้หลายคน
4. แต่ละภูมิภาคจะถูกระบุโดยรหัส REGION\_CODE ที่ไม่ซ้ำกัน และมีคำอธิบาย
5. แต่ละร้านค้าจะมีรหัส STORE\_CODE ที่ไม่ซ้ำกัน มีชื่อ และบันทึกยอดขายสะสมประจำปี
6. แต่ละตำแหน่งงานจะถูกระบุโดยรหัส JOB\_CODE ที่ไม่ซ้ำกัน และมีคำอธิบายกับค่าจ้างพื้นฐาน
7. แต่ละพนักงานจะมีรหัส EMP\_CODE ที่ไม่ซ้ำกัน และมีข้อมูลประกอบด้วยตำแหน่ง ชื่อ-นามสกุล อักษรย่อ และวันเกิด

## 2.2) Create the basic Crow's Foot ERD for DealCo.





## 2.3) Identify each relationship type and write all of the business rules.

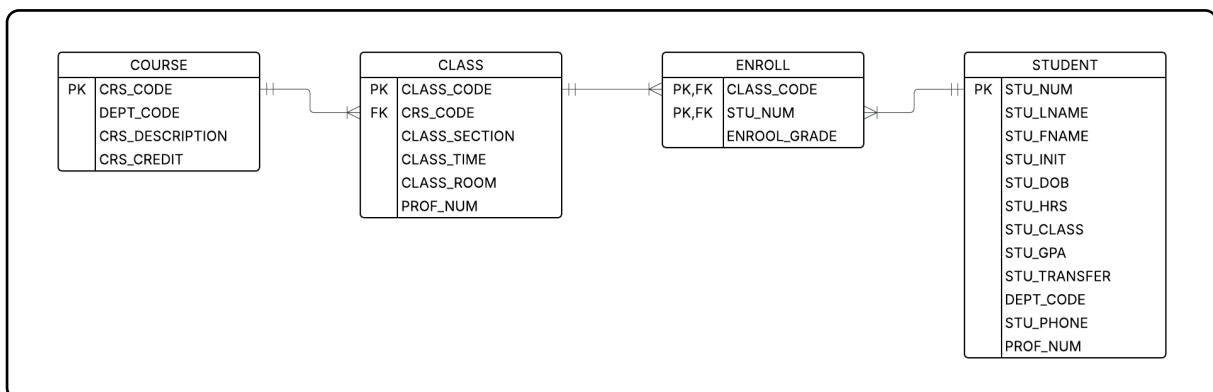
### Relationship Types:

1. **COURSE 1:N CLASS:** One-to-Many (รายวิชาหนึ่งสามารถมีคลาสเรียนหลายคลาสได้)
2. **CLASS 1:N ENROLL:** One-to-Many (คลาสเรียนหนึ่งสามารถมีนักเรียนลงทะเบียนได้หลายคน)
3. **STUDENT 1:N ENROLL:** One-to-Many (นักเรียนหนึ่งคนสามารถลงทะเบียนในหลายคลาสได้)

### Business Rules:

1. รายวิชาหนึ่งสามารถมีคลาสเรียนหลายคลาส แต่แต่ละคลาสจะต้องอิงจากรายวิชาหนึ่งรายวิชาเท่านั้น
2. คลาสเรียนหนึ่งสามารถมีนักเรียนลงทะเบียนได้หลายคน แต่แต่ละข้อมูลการลงทะเบียนจะอ้างอิงถึงคลาสเดียวเท่านั้น
3. นักเรียนหนึ่งคนสามารถลงทะเบียนในหลายคลาส แต่แต่ละข้อมูลการลงทะเบียนจะอ้างอิงถึงนักเรียนเพียงคนเดียว
4. รายวิชาทุกวิชาจะมีรหัส CRS\_CODE ที่ไม่ซ้ำกัน และจะระบุภาควิชา คำอธิบายรายวิชา และจำนวนหน่วยกิต
5. คลาสเรียนแต่ละคลาสจะมีรหัส CLASS\_CODE ที่ไม่ซ้ำกัน และมีข้อมูลเกี่ยวกับ section เวลาเรียน ห้องเรียน และรหัสอาจารย์ผู้สอน
6. ข้อมูลการลงทะเบียนจะใช้ CLASS\_CODE และ STU\_NUM เป็นคีย์รวม และจะเก็บเกรดของนักเรียนในคลาสนั้น
7. นักเรียนแต่ละคนจะมีรหัส STU\_NUM ที่ไม่ซ้ำกัน และมีข้อมูลส่วนตัวและข้อมูลทางการศึกษา เช่น ชื่อ วันเกิด GPA และเบอร์โทรศัพท์

## 2.4) Create the basic Crow's Foot ERD for Tiny College.



## 2.5) Create the UML class diagram that reflects the entities and relationships you identified in the relational diagram.

## 2.6) CREATE OO MODEL