



ปีการศึกษา 2564

Database System

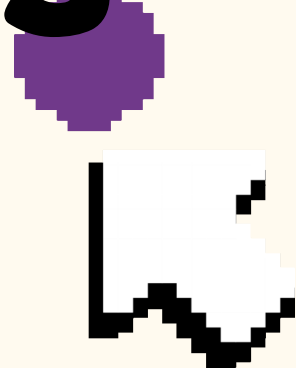
คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์



The background features abstract, wavy shapes in a soft pink color on the left side and a muted teal color on the bottom right corner, set against a light cream background.

Producer of this video

Table of Contents



01 ER Diagram

02 Normalization

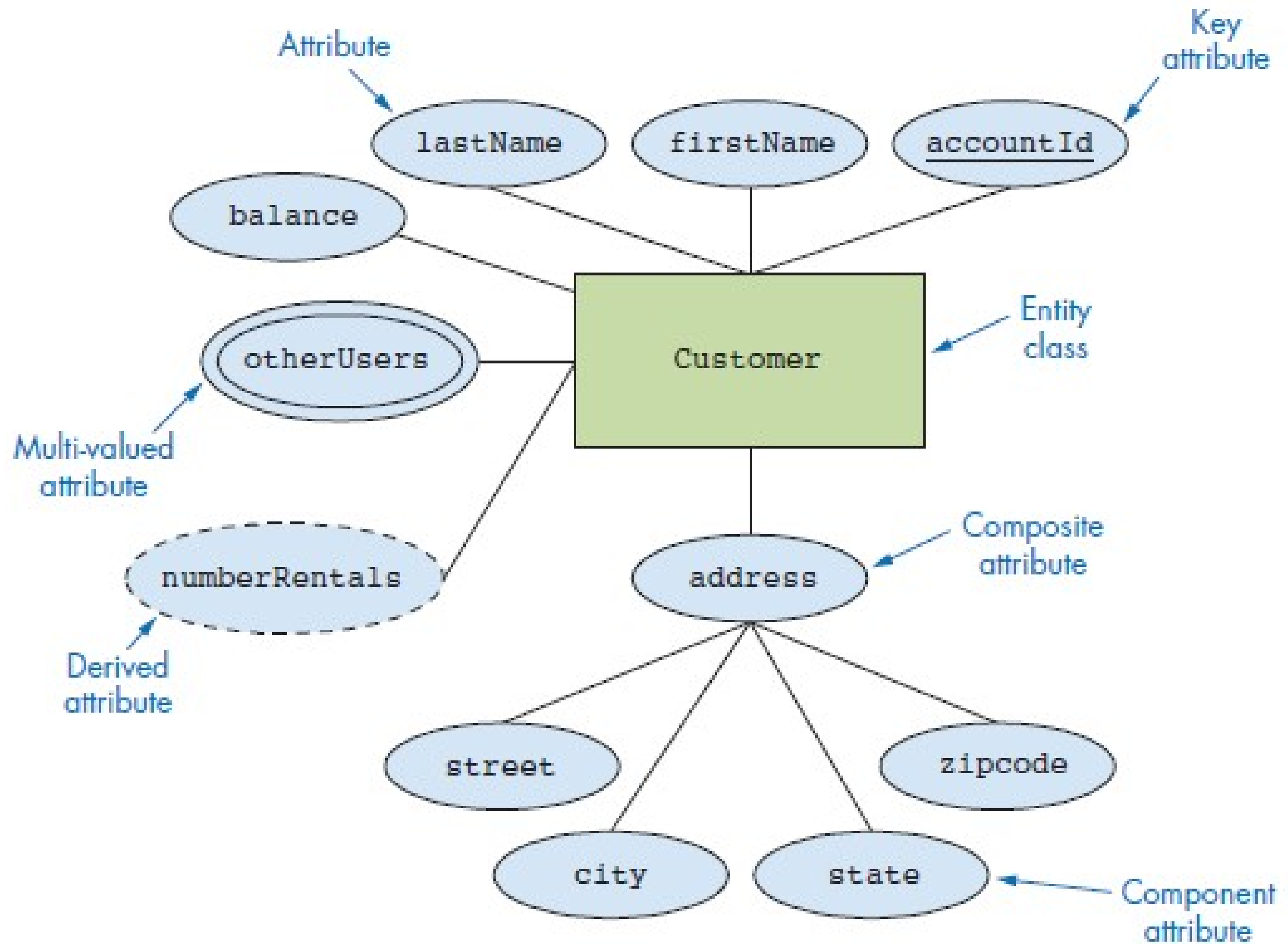
03 SQL

ER Diagram

Entity Relationship Diagram



Example





Vocabulary

Entity

Attribute

Employee ID	Name	Department
1001	Jijo	Administration
1002	Binish	HR
1003	Meera	Software

Tuple

ENTITY

เป็นสิ่งที่ต่างๆ ที่ผู้ใช้งานฐานข้อมูลสนใจ ซึ่งอาจเป็นรูปธรรม มองเห็นได้



Entity

ต้องอยู่ในฐานข้อมูลเสมอ



Weak Entity

ขึ้นกับ Entity อื่น

KEY

(กลุ่ม) Attribute ที่สามารถแยกความแตกต่างกันได้

Simple Key

Super Key

Primary Key

Composite Key

หรือ Combine Key

Secondary Key

หรือ alternate Key

Candidate Key

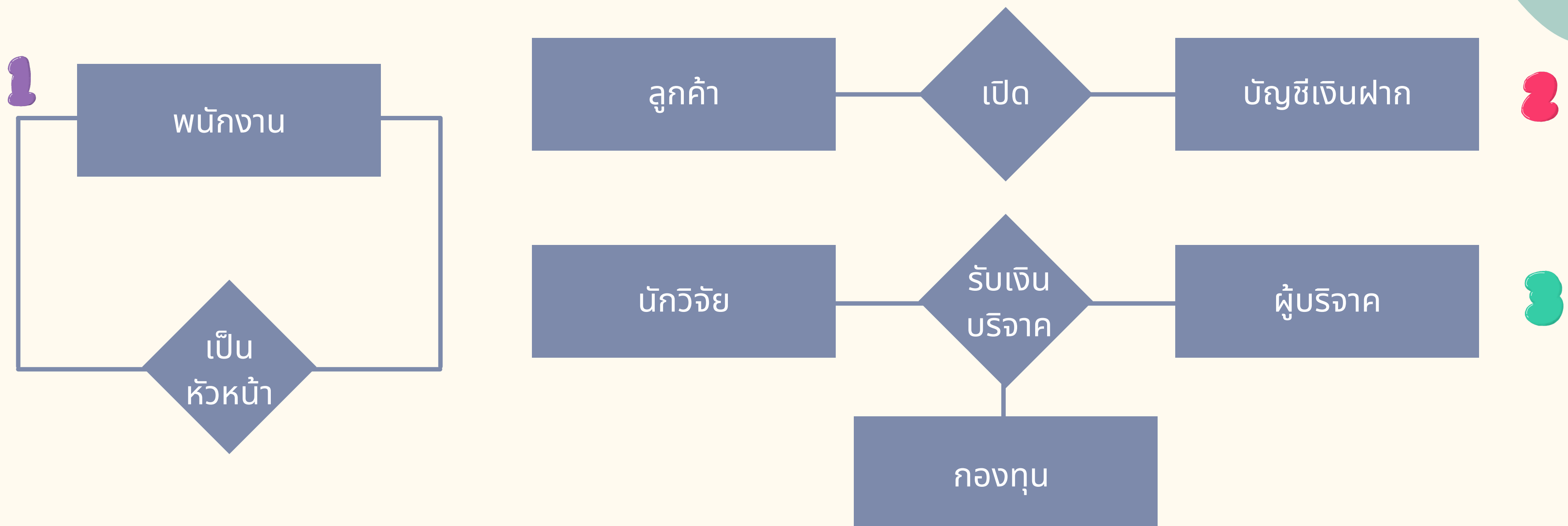
Foreign Key



ใช้ใน Normalization

RELATIONSHIP

DEGREE จำนวนของ Entity ที่
เกี่ยวข้องกับความสัมพันธ์



RELATIONSHIP CONSTRAINT

ข้อบังคับ cardinalities ของความสัมพันธ์ระหว่าง Entity :one to one



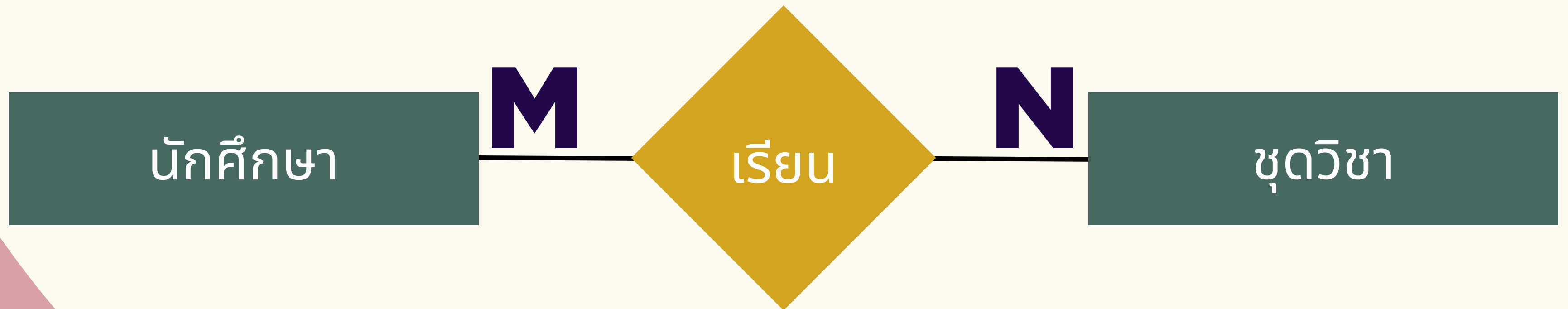
RELATIONSHIP CONSTRAINT

ข้อบังคับ cardinalities ของความสัมพันธ์ระหว่าง Entity : one to many



RELATIONSHIP CONSTRAINT

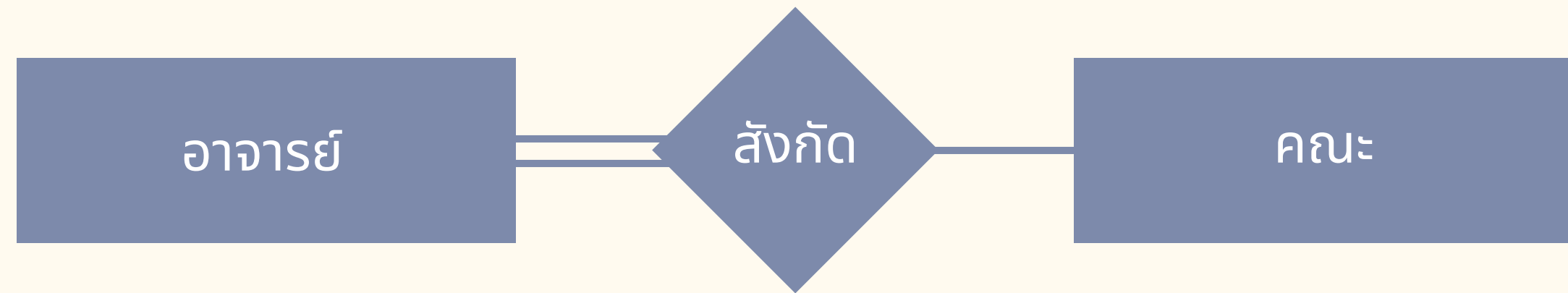
ข้อบังคับคาร์ดินัลลิตีของความสัมพันธ์ระหว่าง Entity : many to many



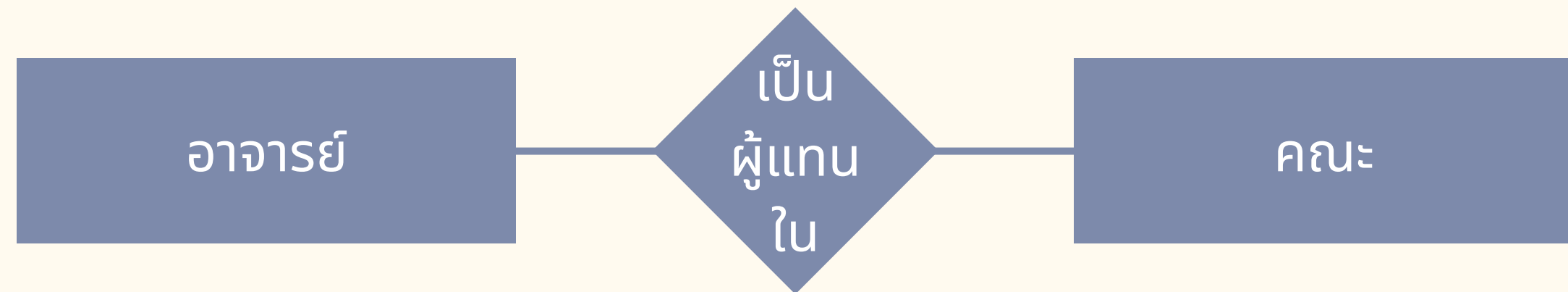
RELATIONSHIP CONSTRAINT

ข้อบังคับการมีส่วนร่วมของความสัมพันธ์ระหว่าง Entity : many to many

ทั้งหมด:



บางส่วน :

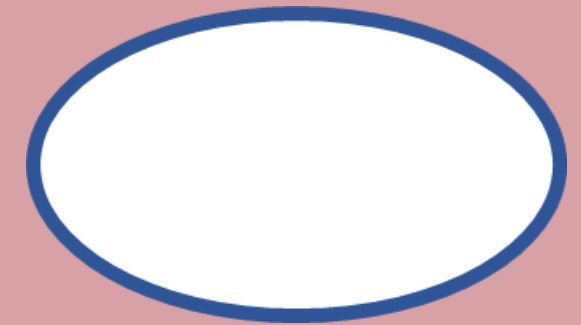


ATTRIBUTE

คุณลักษณะที่บ่งบอกถึง Entity

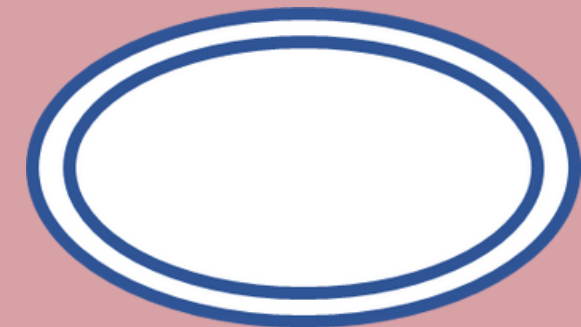
Simple Attribute

ไม่สามารถแบ่งได้
เช่น อายุ เพศ



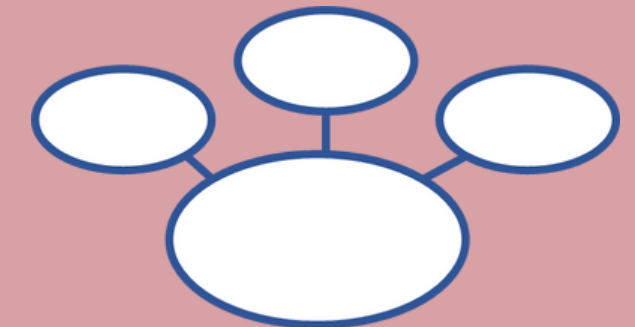
Multi-valued Attribute

เกิดการจากข้อมูลประเภทเดียวกันที่มีหลายค่า
เช่น เบอร์โทร



Composite Attribute

สามารถแบ่งได้
เช่น ที่อยู่



Derived Attribute

ผ่านการคำนวณ
เช่น อายุ



EXAMPLE

การสร้าง ER และแปลงให้เป็น
Relational Database Model



Normalization

1NF - 3NF , BCNF



Why?

Insertion Anomalies

เมื่อเพิ่มข้อมูล แต่ว่าไม่สามารถเพิ่มได้ครบทุก Attribute ทำให้เกิดปัญหา

Deletion Anomalies

เมื่อลบข้อมูลที่ใดแล้ว อาจส่งผลให้ข้อมูลที่อื่นหายไปด้วย

Modification Anomalies

เมื่อจะเปลี่ยนหรือแก้ไขข้อมูล อาจส่งผลให้ต้องทำหลายที่

<u>Student_ID</u>	Student_name	<u>Subject_ID</u>	Subject_name
1001	Aimer	9001	Math
1001	Aimer	9002	English
1001	Aimer	9003	Physic
1002	Haruko	9001	Math
1002	Haruko	9002	English

Example



1. Designed to represent each tuple.

เพื่อให้ดูข้อมูลได้ง่ายตาม Entity

2. Don't have 3 Anomalies.

เพื่อลดปัญหาและการใช้พื้นที่โดยไม่จำเป็น

3. Want "null value" to a minimum.

เพื่อลดภาระของตาราง เพราะว่าทุก Attribute จะต้องมามีค่า

4. Don't have Spurious Tuple.

เพื่อให้ข้อมูลถูกต้อง + ไม่สูญหาย



Good Relation

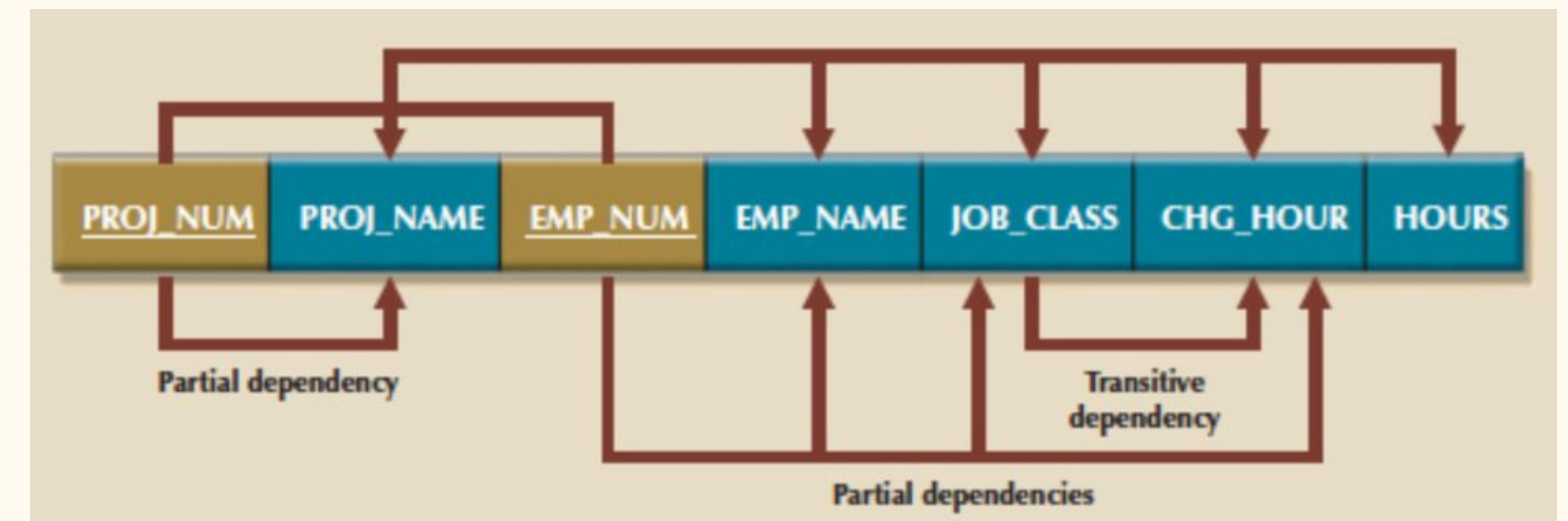
FD : Fuctional Dependencies

เป็น (กลุ่ม) Attribute ที่สามารถบ่งบอกถึง Attribute อื่นได้ในตาราง

$X \longrightarrow Y$: ค่า X บ่งบอกถึงค่า Y

there are 3 types

1. Fully Dependency
2. Partial Dependency
3. Transitive Dependency



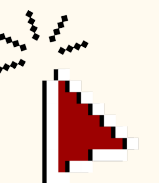
How?

1NF

- ในแต่ละ Column ต้องมีค่าเพียง 1 ค่า
- ทุกๆ Attribute ในตาราง ต้องถูกระบุได้คีย์หลัก

2NF

- ต้องเป็น 1st Normal Form
- ทุกคีย์ที่ไม่ใช่คีย์หลัก ต้องขึ้นกับคีย์หลัก (ไม่มี Partial Functional Dependency)



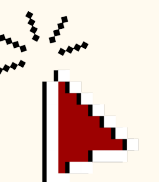
How?

3NF

- ต้องเป็น 2nd Normal Form
- ไม่มี Attribute ใดขึ้นกับ Attribute อื่นๆที่ไม่ใช่คีย์หลัก (ไม่มี Transitive Dependency)

BCNF

- ต้องเป็น 3rd Normal Form
- ไม่มี Attribute อื่นใน relation ที่สามารถระบุค่าของ Attribute ที่เป็นคีย์หลักหรือส่วนใดส่วนหนึ่งของคีย์หลักในกรณีเป็นคีย์ผสม (ไม่มี Candidate Key)



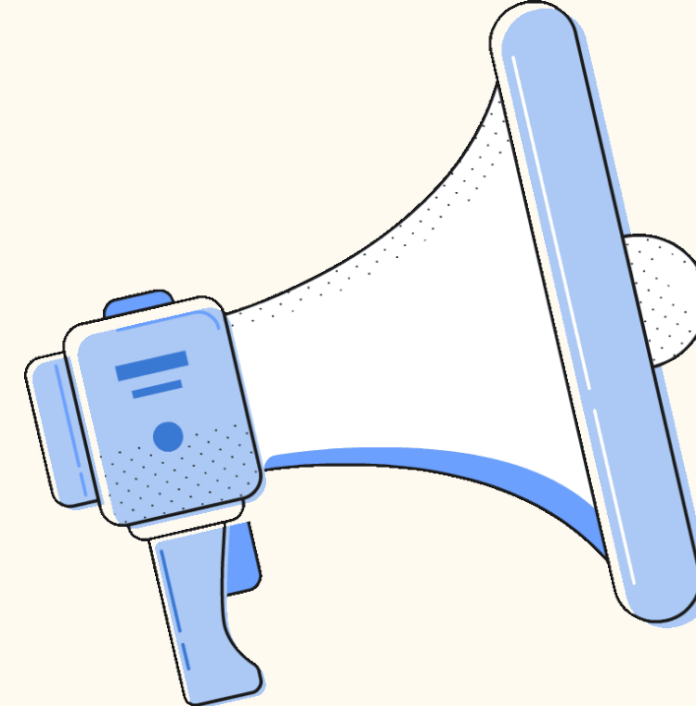
SQL

Structured Query Language





Structure



SELECT

ชื่อ Attribute

FROM

ชื่อตาราง

WHERE

เงื่อนไขเลือกข้อมูล

ORDER BY

ชื่อ Attribute ASC/DESC

GROUP BY

ชื่อ Attribute

HAVING

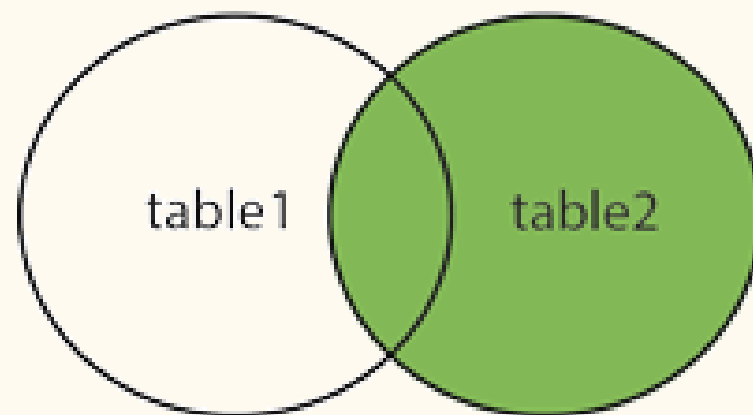
เงื่อนไขดูข้อมูล

Different Types of SQL JOINS

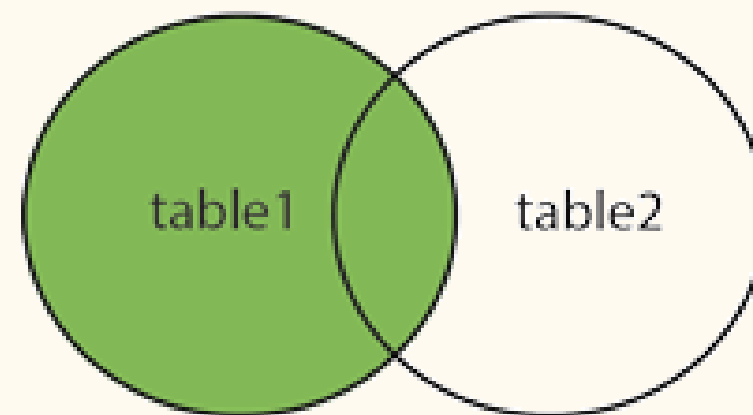
Here are the different types of the JOINS in SQL:

- (INNER) JOIN : Returns records that have matching values in both tables
- LEFT (OUTER) JOIN : Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN : Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN : Returns all records when there is a match in either left or right table

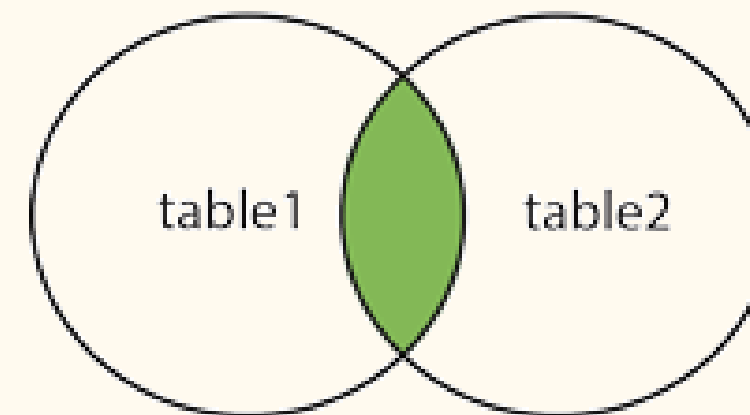
RIGHT JOIN



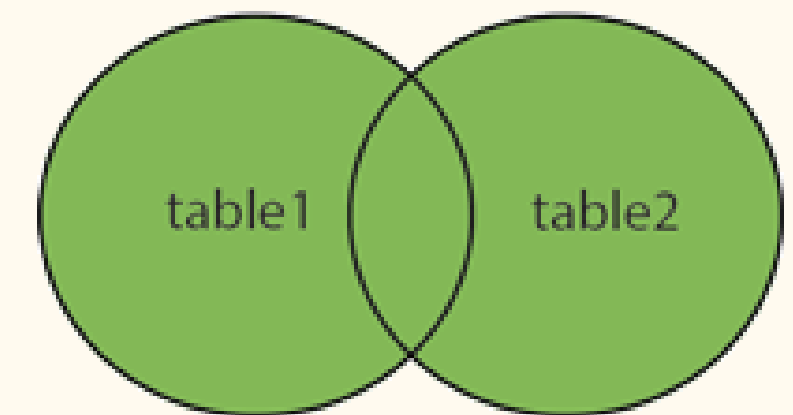
LEFT JOIN



INNER JOIN



FULL OUTER JOIN





Thank you!