Written by Thapanapong Rukkanchanunt

Structured Query Language

ทบทวนเนื้อหา

- ฐานข้อมูลเป็นแหล่งรวบรวมข้อมูล (data) ไว้อย่างเป็นระบบ
- Entity–Relationship Model หรือเรียกโดยย่อว่า ER Model เป็นแบบจำลองข้อมูลที่กำหนดรายละเอียดและข้อบังคับของข้อมูล โดยจะประกอบไปด้วยสามส่วนหลักคือ Entities, Attributes และ Relationships
- Entities คือข้อมูลหลักของวัตถุที่ต้องการเก็บข้อมูล ส่วนมากแล้วจะใช้แสดงบุคคล สถานที่ สิ่งของ หรือเหตุการณ์ที่น่าจดจำ
- Relationships แสดงความสัมพันธ์ระหว่าง Entities ตั้งแต่หนึ่ง Entities ขึ้นไป
- Attributes คือคุณลักษณะของ Entities หรือ Relationships

ภาษาที่ใช้ติดต่อกับฐานข้อมูล

- ภาษาที่ใช้ในการติดต่อกับฐานข้อมูลนั้นมีอยู่มากมาย แต่ภาษาที่ได้รับความนิยมมากที่สุดคือภาษา SQL (Structured Query Language)
- ระบบจัดการฐานข้อมูลที่มีคำว่า SQL อยู่ในชื่ออนุญาตให้ใช้ภาษา SQL ได้ เช่น MySQL, SQLite, PostgreSQL, SQL Server
- เราจะเรียนรู้ภาษา SQL ในวิชานี้

การสร้างตาราง

- ภาษา SQL สำหรับสร้างตารางในฐานข้อมูลมีรูปแบบดังนี้

```
CREATE TABLE ชื่อตาราง
(
ชื่อแอตทริบิวต์1        ประเภทของข้อมูล1,
ชื่อแอตทริบิวต์2        ประเภทของข้อมูล2,
...
);
```

ประเภทของข้อมูล

ประเภทของข้อมูล	คำอธิบาย
TEXT	มีลักษณะเป็นข้อความ ไม่สามารถนำมาคำนวณได้แม้จะพิมพ์ตัวเลข
INTEGER	เลขจำนวนเต็ม
REAL	เลขจำนวนจริงหรือทศนิยม
BLOB	เก็บข้อมูลตามที่พิมพ์จริง

ตัวอย่างการสร้างตาราง

- ถ้าต้องการสร้างตารางเพื่อเก็บข้อมูลนักศึกษา

```
CREATE TABLE students
(
student_id      INTEGER,
name            TEXT,
year            INTEGER
);
```

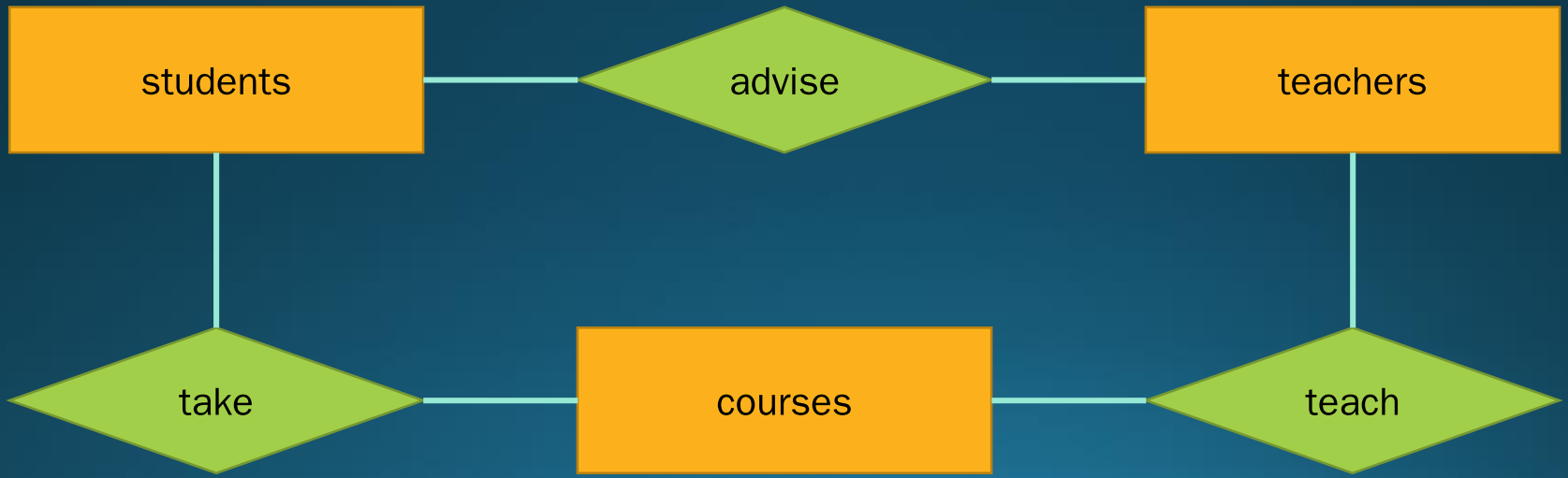
ระบุ Key

- เราสามารถระบุ Key ได้ด้วยการเติมคำว่า PRIMARY KEY ตามหลังประเภทของข้อมูล

```
CREATE TABLE students
(
student_id      INTEGER       PRIMARY KEY,
name            TEXT,
year            INTEGER
);
```

ตัวอย่างฐานข้อมูลที่จะใช้ในเทอมนี้

- ฐานข้อมูลของสถาบันการศึกษาแห่งหนึ่ง



รายละเอียดของฐานข้อมูลตัวอย่าง

- เพื่อให้ง่ายต่อการจำ เราจะเขียน Entities และ Relationships ดังนี้

students(student_id, name, year)

teachers(teacher_id, name, dept)

courses(course_id, name, credits)

advise(student_id, teacher_id)

take(student_id, course_id, term, grade)

teach(teacher_id, course_id, term)

การเพิ่มข้อมูลในตาราง

- ภาษา SQL สำหรับเพิ่มข้อมูลในตารางมีรูปแบบดังนี้

```
INSERT INTO ชื่อตาราง
VALUES (ค่าของแอตทริบิวต์1, ค่าของแอตทริบิวต์2, ...);
```

ตัวอย่างการเพิ่มข้อมูลในตาราง

- ถ้าต้องการเพิ่มชื่อนักศึกษาใหม่ที่ชื่อ Leonardo DiCaprio สำหรับปีการศึกษา 2559

```
INSERT INTO students
VALUES (590123456, 'Leonardo DiCaprio',
        2559);
```

students(student_id, name, year)
teachers(teacher_id, name, dept)
courses(course_id, name, credits)
advise(student_id, teacher_id)
take(student_id, course_id, term, grade)
teach(teacher_id, course_id, term)

การแก้ไขข้อมูลในตาราง

- ภาษา SQL สำหรับแก้ไขข้อมูลในตารางมีรูปแบบดังนี้

```
UPDATE ชื่อตาราง
SET ชื่อแอตทริบิวต์1 = ค่าใหม่1, ชื่อแอตทริบิวต์1 = ค่าใหม่2, ...
WHERE เงื่อนไข;
```

เงื่อนไข

- สำหรับเงื่อนไขในส่วนของ WHERE นั้นจะเป็นการเปรียบเทียบค่าแอตทริบิวต์ของตารางที่อยู่ในส่วน UPDATE
- เครื่องหมายที่ใช้ในการเปรียบเทียบได้แก่ =, <>, >, <, >=, <=, BETWEEN, NOT BETWEEN, LIKE, IN
- สามารถใช้คำว่า AND, OR ในการเชื่อมหลายเงื่อนไขเข้าด้วยกัน
- ถ้าไม่มีการใส่เงื่อนไข จะถือว่าเป็นการแก้ไขทุกระเบียนของตาราง

ตัวอย่างเงื่อนไขและความหมาย

เงื่อนไข	ความหมาย
name = ‘Leonardo DiCaprio’	ชื่อต้องเป็น Leonardo DiCaprio
year < 2559	ก่อนปี 2559
year BETWEEN 2557 AND 2559	ระหว่างปี 2557 และ 2559
year NOT BETWEEN 2557 AND 2559	ไม่ได้อยู่ระหว่างปี 2557 และ 2559
dept IN (‘Mathematics’, ‘Physics’)	อยู่ในภาควิชาคณิตศาสตร์หรือฟิสิกส์

เงื่อนไข LIKE

- LIKE เป็นเงื่อนไขที่ใช้ในการเปรียบเทียบข้อความด้วยการค้นหาตามรูปแบบที่กำหนด
- % แทนรูปแบบตัวอักษรใดก็ได้ตั้งแต่ 0 ตัวอักษรขึ้นไป
- _ แทนรูปแบบตัวอักษรใดก็ได้หนึ่งตัวอักษรเท่านั้น

name LIKE ‘L%’ แปลว่าชื่อที่ขึ้นต้นด้วย L

name LIKE ‘%L’ แปลว่าชื่อที่ลงท้ายด้วย L

name LIKE ‘L__’ แปลว่าชื่อที่ขึ้นต้นด้วย L และมีสามตัวอักษร

ตัวอย่างการแก้ไขข้อมูลในตาราง

- ถ้าต้องการแก้ไขชื่อจาก Leonardo DiCarprio เป็น Leonardo da Vinci

```
UPDATE students
SET name = ‘Leonardo da Vinci’
WHERE name = ‘Leonardo DiCarprio’;
```

students(student_id, name, year)
teachers(teacher_id, name, dept)
courses(course_id, name, credits)
advise(student_id, teacher_id)
take(student_id, course_id, term, grade)
teach(teacher_id, course_id, term)

การลบข้อมูลในตาราง

- ภาษา SQL สำหรับลบข้อมูลในตารางมีรูปแบบดังนี้

```
DELETE FROM ชื่อตาราง
WHERE เงื่อนไข;
```

ตัวอย่างการลบข้อมูลในตาราง

- ถ้าต้องการลบข้อมูลของวิชาที่ให้ 0 หน่วยกิจทั้งหมด

```
DELETE FROM courses
WHERE credits = 0;
```

students(student_id, name, year)

teachers(teacher_id, name, dept)

courses(course_id, name, credits)

advise(student_id, teacher_id)

take(student_id, course_id, term, grade)

teach(teacher_id, course_id, term)

การเรียกดูข้อมูลในตาราง

- ภาษา SQL สำหรับเรียกดูข้อมูลในตารางมีรูปแบบดังนี้

```
SELECT ชื่อแอตทริบิวต์1, ชื่อแอตทริบิวต์2, ...
FROM ชื่อตาราง
WHERE เงื่อนไข;
```

ดอกจันและ DISTINCT

- ในกรณีที่ต้องการดูข้อมูลทุกแอตทริบิวต์ของตาราง เราสามารถใช้เครื่องหมายดอกจัน (*)แทนการพิมพ์ชื่อแอตทริบิวต์ทุกตัว
- ในกรณีที่ต้องการดูข้อมูลค่าแอตทริบิวต์ที่ไม่ซ้ำกัน เราสามารถใช้คำว่า DINTINCT กำกับไว้ด้านหน้าชื่อแอตทริบิวต์นั้น

ตัวอย่างการเรียกดูข้อมูลในตาราง

- ถ้าต้องการทราบว่ามีนักศึกษาคนใดบ้างที่เข้ามาเรียนในปี 2557

```
SELECT name
FROM students
WHERE year = 2557;
```

students(student_id, name, year)

teachers(teacher_id, name, dept)

courses(course_id, name, credits)

advise(student_id, teacher_id)

take(student_id, course_id, term, grade)

teach(teacher_id, course_id, term)

การเรียกข้อมูลหลายตารางพร้อมกัน

- ในกรณีที่เราต้องการข้อมูลที่สัมพันธ์กันในหลายตาราง เราสามารถเพิ่มตารางที่เกี่ยวข้องไว้ที่ FROM โดยใช้เครื่องหมาย Comma คั่นระหว่างตาราง
- นอกจากนี้แล้วเราจะต้องเพิ่มเงื่อนไขในการรวมตาราง โดยการนำ Key ของสองตารางที่มีเส้นเชื่อมกันมาเปรียบเทียบด้วยเครื่องหมายเท่ากับ ซึ่งเราจะเรียกเงื่อนไขนี้ว่า Join Condition
- ถ้าหลายตารางมีชื่อแอตทริบิวต์ที่ซ้ำกัน เราจำเป็นต้องระบุชื่อตารางด้วย โดยมีรูปแบบเป็น ชื่อตาราง.ชื่อแอตทริบิวต์ เช่น students.name หรือ teachers.name

ตัวอย่างการเรียกข้อมูลหลายตาราง

- ถ้าต้องการทราบว่ามีวิชาใดบ้างที่เปิดสอนในเทอม 2/58

```
SELECT name
FROM courses, teach
WHERE term = '2/58'
    AND courses.course_id = teach.course_id;
```

students(student_id, name, year)
teachers(teacher_id, name, dept)
courses(course_id, name, credits)
advise(student_id, teacher_id)
take(student_id, course_id, term, grade)
teach(teacher_id, course_id, term)

แบบฝึกหัดที่ 1

- ต้องการทราบว่านักศึกษาที่ชื่อลงท้ายด้วยคำว่า son ได้เกรดอะไรบ้างในเทอม 1/58

students(student_id, name, year)

teachers(teacher_id, name, dept)

courses(course_id, name, credits)

advise(student_id, teacher_id)

take(student_id, course_id, term, grade)

teach(teacher_id, course_id, term)

แบบฝึกหัดที่ 2

- ต้องการทราบว่าอาจารย์ Nicolas Cage เคยสอนวิชาใดบ้าง

students(student_id, name, year)

teachers(teacher_id, name, dept)

courses(course_id, name, credits)

advise(student_id, teacher_id)

take(student_id, course_id, term, grade)

teach(teacher_id, course_id, term)

แบบฝึกหัดที่ 3

- ต้องการทราบว่ามีนักศึกษาคนใดบ้างที่เคยเรียนกับอาจารย์ Nicolas Cage

students(student_id, name, year)

teachers(teacher_id, name, dept)

courses(course_id, name, credits)

advise(student_id, teacher_id)

take(student_id, course_id, term, grade)

teach(teacher_id, course_id, term)

แบบฝึกหัดที่ 4

- ต้องการทราบว่ามีนักศึกษาคนใดบ้างที่เคยลงเรียนวิชาที่อาจารย์ที่ปรึกษาตนเองเป็นคนสอน

students(student_id, name, year)

teachers(teacher_id, name, dept)

courses(course_id, name, credits)

advise(student_id, teacher_id)

take(student_id, course_id, term, grade)

teach(teacher_id, course_id, term)