

Database



Outline

1. Entity-Relationship Diagram(ERD)

- Thực thể, quan hệ, thuộc tính

2. Relational Database

- Quan hệ 1 - 1
- Quan hệ 1 - n
- Quan hệ n - n

3. Luyện tập

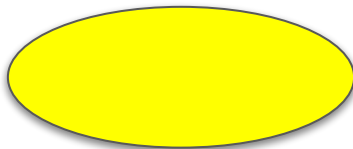
1. Entity-Relationship Diagram(ERD)

1.1. ERD là gì?

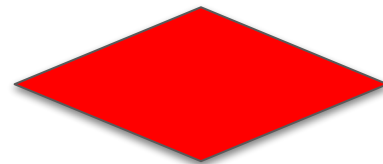
- Entity-Relationship Diagram (ERD) là một sơ đồ, thể hiện các thực thể có trong database, và mối quan hệ giữa chúng với nhau.
- Các thành phần trong lược đồ ERD



Entity
(Thực thể)



Attribute
(Thuộc tính)



Relationship
(Quan hệ)

1.2. Thực thể

- Thực thể (Entity): Là một đối tượng cần lưu trữ trong hệ thống.
- Bài toán: Xây dựng một hệ thống quản lý khóa học online, nơi người dùng có thể đăng ký các khóa học. Hãy xác định các thực thể có thể có của hệ thống?

1.2. Thực thể

Student

Teacher

Course

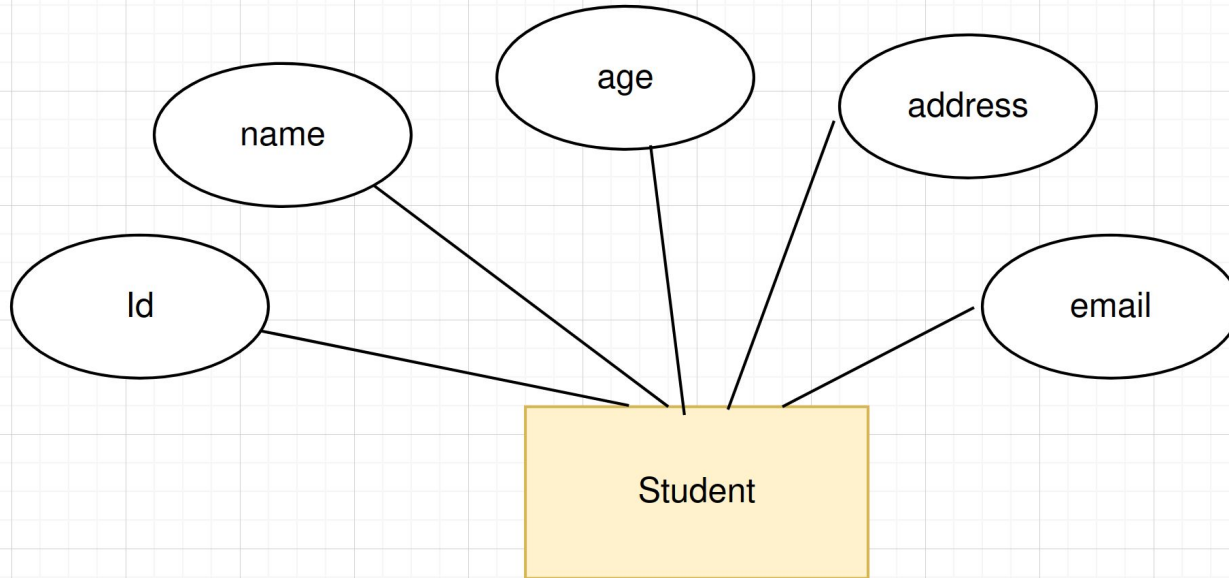
Admin

Chapter

Lesson

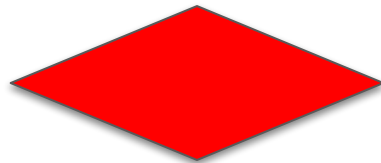
1.3. Thuộc tính

- Thuộc tính (Attribute): Là các thuộc tính của thực thể như id, tên, tuổi, ...



1.4 Mỗi quan hệ

- Mỗi quan hệ (Relationship): Là mối quan hệ giữa các đối tượng với nhau.

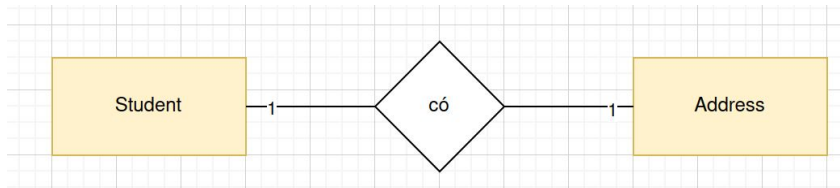


Relationship
(Quan hệ)

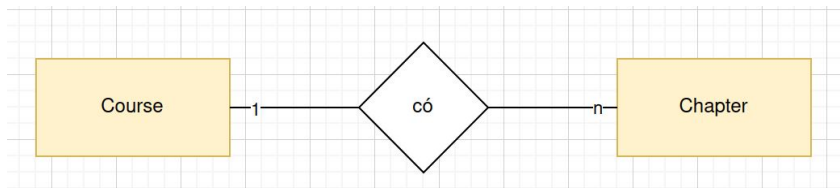
- Có ba loại mối quan hệ:
 - One-to-One: quan hệ 1-1
 - One-to-Many: quan hệ 1-nhiều
 - Many-to-Many: quan hệ nhiều-nhiều

1.4 Ví dụ về quan hệ

- **Mỗi** sinh viên chỉ có một địa chỉ



- **Mỗi** khóa học sẽ có nhiều chương



- **Một** sinh viên có thể tham gia **nhiều** khóa học, **một** khóa học có thể có **nhiều** sinh viên



2. Relational Database

- Relational Database là database có nhiều bảng (table), và giữa các bảng quan hệ với nhau bằng các khóa (key).
- 1 table chính là 1 entity (một đối tượng, hoặc một thực thể) mà database lưu trữ.
- Khóa chính (Primary Key): Là một cột có giá trị để định danh duy nhất mỗi bản ghi trong table của cơ sở dữ liệu.
- Vì các table liên kết với nhau. Nhưng để liên kết với nhau thì nó cần có điểm chung nào đó.

Foreign Key chính là điểm chung đó. Nó là key dùng để liên kết 2 tables lại với nhau.

- Khóa ngoại (Foreign Key): Là một cột dùng để xác định mối quan hệ giữa hai bảng với nhau. Giá trị của khóa ngoại thường là khóa chính của một bảng khác.

Ví dụ quan hệ 1 - 1

- **Mỗi** sinh viên chỉ có một địa chỉ

id	name	address_id
1	Nguyễn Thị C	2
2	Trần Văn A	1
3	Nguyễn Văn B	3
4	Trần Thị D	4

Student

id	address	city
1	Cầu Giấy	Hà Nội
2	Hà Đông	HCM
3	Sơn Tây	HCM
4	Ba Đình	Hà Nội

address

Ví dụ quan hệ 1 - n:

- **Mỗi** khóa học sẽ có nhiều chương

id	name	price
1	Java Cơ bản	0
2	Java nâng cao	0
3	Kafka	1999000
4	System Design	0

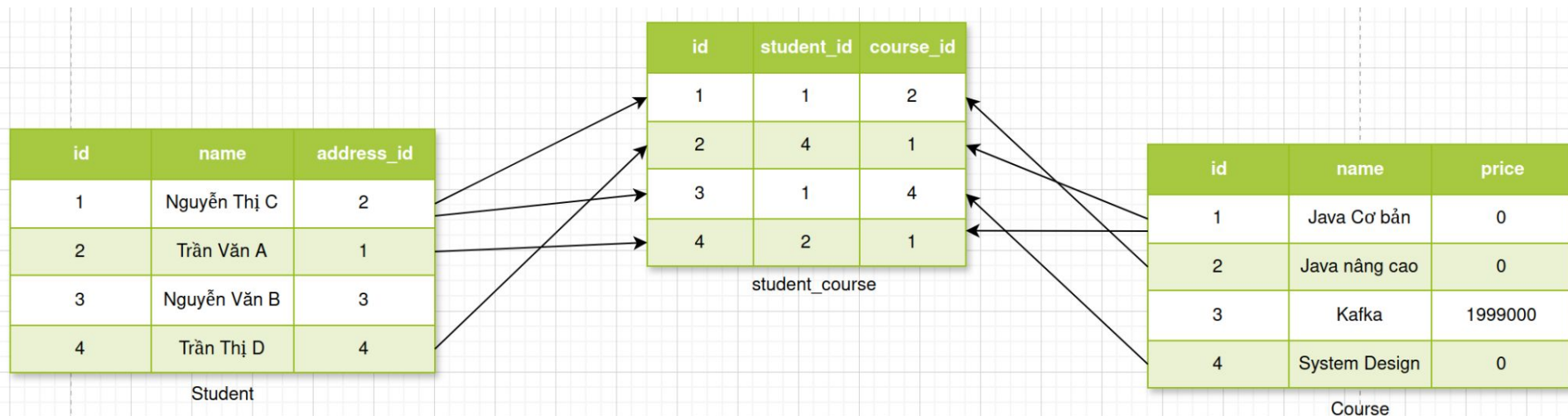
Course

id	name	course_id
1	Syntax cơ bản	1
2	Exceptions	1
3	Kakfa là gì?	3
4	SOLID	2

Chapter

Ví dụ quan hệ n - n:

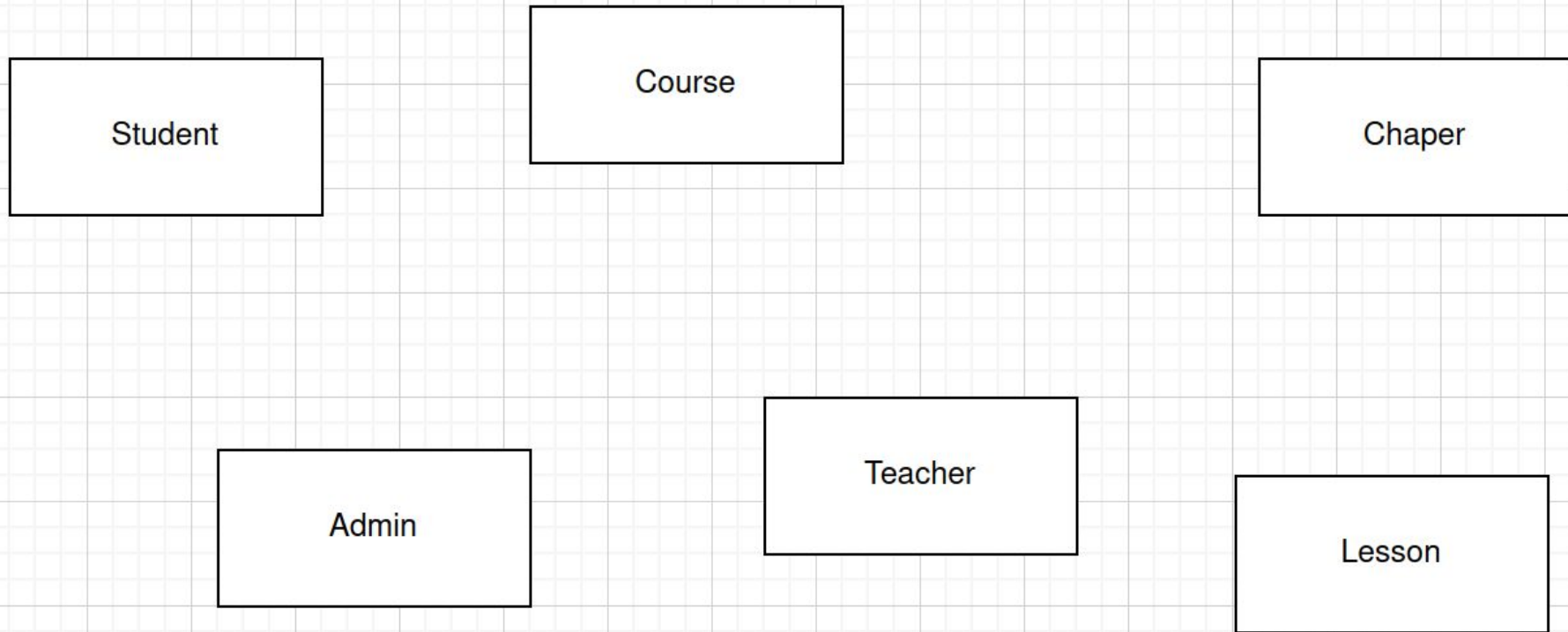
- Để biểu quan hệ nhiều - nhiều ta phải biểu diễn quan hệ thông qua một bảng trung gian (bảng liên kết). Trong đó bảng trung gian có hai cột khóa ngoại - nó có giá trị là khóa chính của hai bảng quan hệ nhiều - nhiều.
- Một** sinh viên có thể tham gia **nhiều** khóa học, **một** khóa học có thể có **nhiều** sinh viên



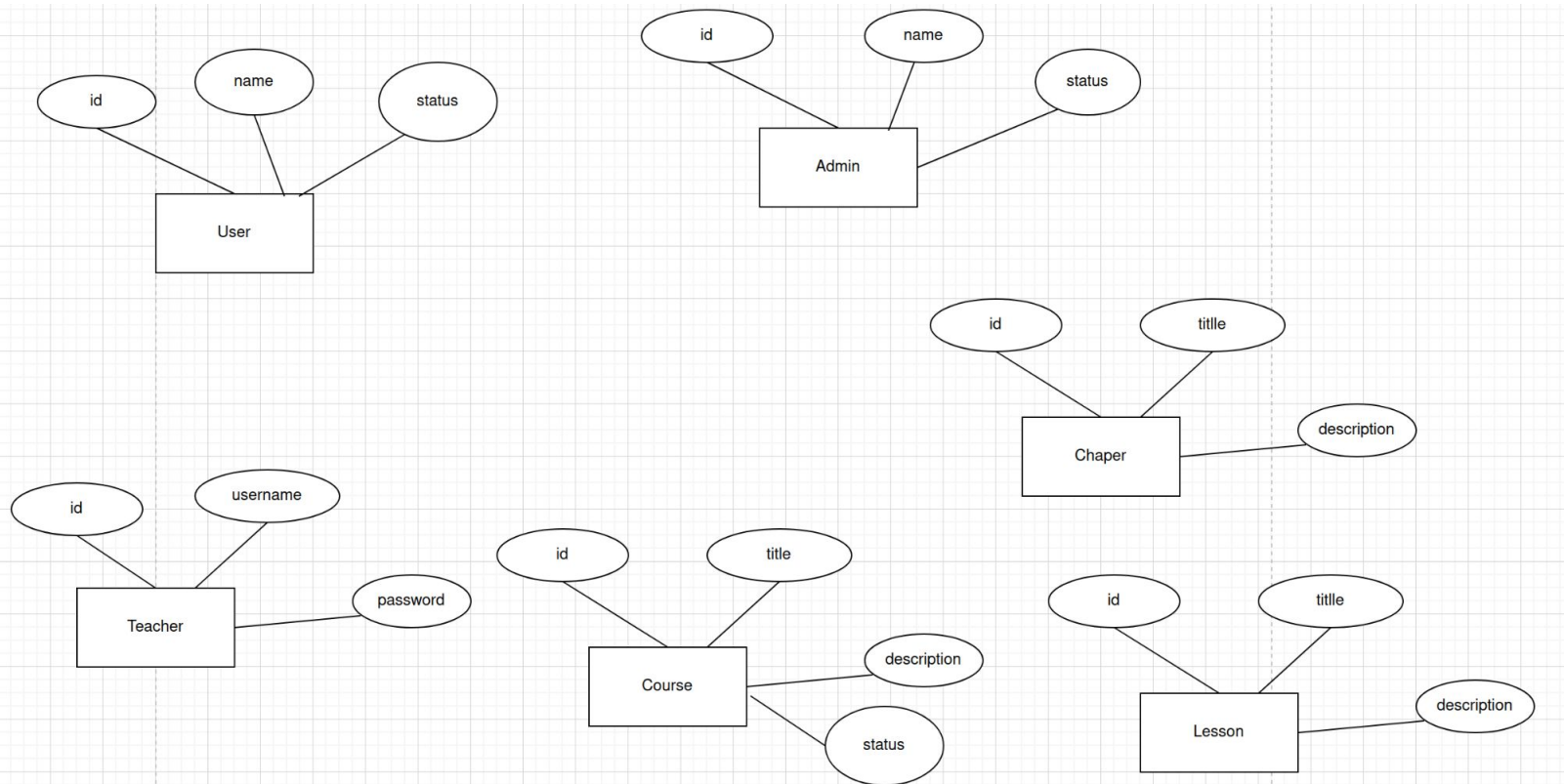
3. Practice

Bài toán: Xây dựng một hệ thống quản lý khóa học online, nơi người dùng có thể đăng ký các khóa học, đánh giá và xếp hạng, quản lý giáo viên, khóa học. Hệ thống này cũng sẽ hỗ trợ quản lý các giáo viên tham gia giảng dạy.

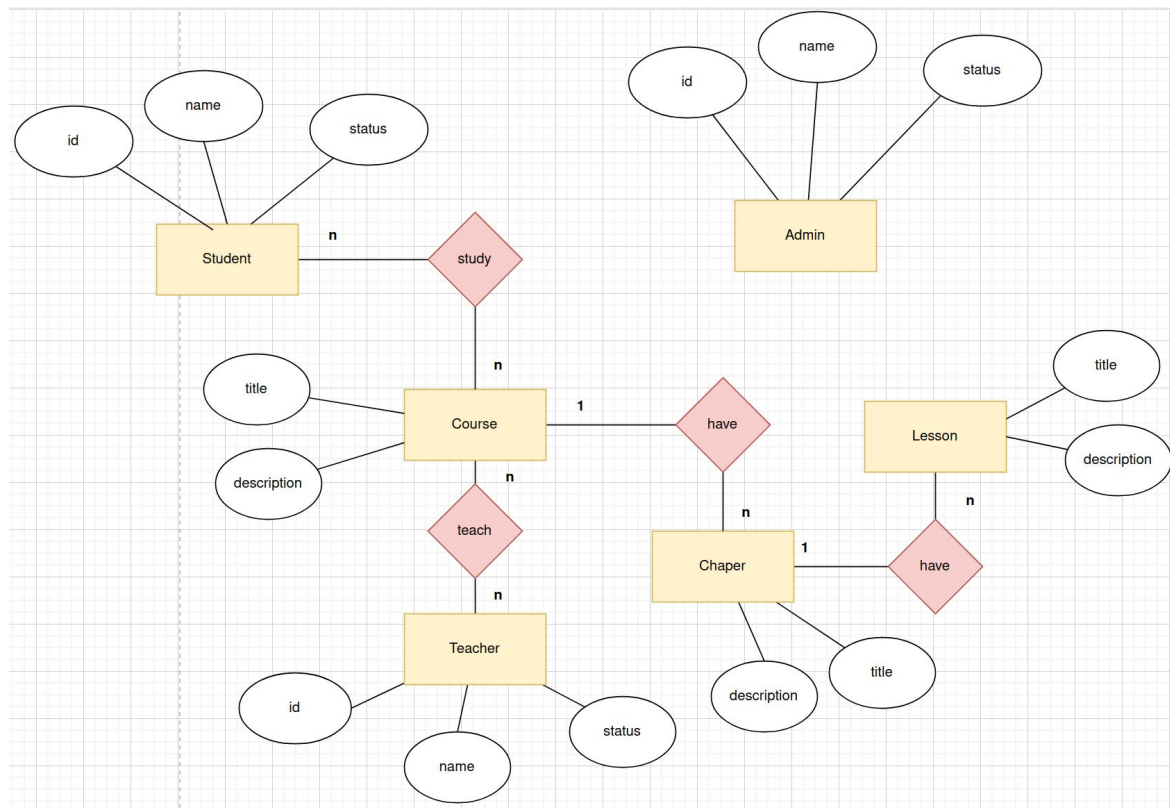
Bước 1: Xác định các thực thể



Bước 2: Thêm các thuộc tính cho thực thể



Bước 3: Xác định mối quan hệ cho các thực thể



Bước 4: Thao tác với database - Tạo bảng, thuộc tính

```
# Tạo bảng Student

create table Student

(
    STUDENT_ID int          not null primary key auto_increment,
    NAME        varchar(250) not null
);
```

Bước 4: Thao tác với database - Tạo bảng, thuộc tính

Bảng Course

```
create table course
```

```
(
```

```
    SUBJECT_ID int          not null primary key auto_increment,
```

```
    TITLE      varchar(250) not null
```

```
);
```

Bảng Chapter

```
create table chapter
```

```
(
```

```
    BOOK_ID    int          not null primary key auto_increment,
```

```
    TITLE      varchar(250) not null,
```

```
);
```

Bước 4: Thao tác với database - Tạo mối quan hệ 1 - 1, 1 - n

```
# Student ---- 1 ----- 1 ----- Card
```

```
alter table Card
```

```
add column STUDENT_ID int,
```

```
add FOREIGN KEY(STUDENT_ID) references Student(STUDENT_ID);
```

Bước 4: Thao tác với database - Tạo mối quan hệ n - n

```
# Student ---- 1 ----- N ----- Book

alter table Book

add column STUDENT_ID int,

add FOREIGN KEY(STUDENT_ID) references Student(STUDENT_ID);

# Student ---- N ----- N ----- Subject

create table Student_Subject

(

    ID int not null primary key auto_increment,

    STUDENT_ID int not null,

    SUBJECT_ID int not null,

    FOREIGN KEY(STUDENT_ID) references Student(STUDENT_ID),

    FOREIGN KEY(SUBJECT_ID) references Subject(SUBJECT_ID)

);
```

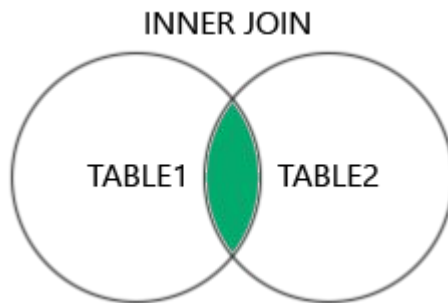
Bước 4: Thực hiện truy vấn

- Lấy danh sách tên sinh viên có trạng thái đang hoạt động.
- Lấy danh sách khóa học của sinh viên có Id = 1

⇒ Khi cần lấy thông tin ở các bảng có quan hệ với nhau, ta sử dụng câu lệnh **Join**

Câu lệnh Join

- **Câu lệnh Inner join:** Inner join là câu lệnh kết hợp hai bảng với nhau và giữ lại các bản ghi trùng khớp giữa hai bảng.



Câu lệnh Join

- **Ví dụ:** Lấy danh sách tên sinh viên và tên sách mà mỗi sinh viên đã mượn.

```
select Student.NAME, Book.NAME
```

```
from Student
```

```
inner join Book on Student.STUDENT_ID = Book.STUDENT_ID;
```

- **Giải thích:** Kết hợp các bản ghi ở cả hai bảng Student và Book vào với nhau, sao cho giá trị của cột STUDENT_ID của mỗi bản ghi ở bảng Student phải tương ứng với giá trị của cột STUDENT_ID của mỗi bản ghi ở bảng Book.



↓

	Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1	Nguyen Van A	1	Toán Cao Cấp	Hà Nội	2
2	1	Nguyen Van A	2	Tiếng Anh A11	Nghệ An	1
3	1	Nguyen Van A	4	Lập trình C++	hải Phòng	<null>
4	1	Nguyen Van A	5	Triết học	Bắc Ninh	1
5	2	Tran Van B	1	Toán Cao Cấp	Hà Nội	2
6	2	Tran Van B	2	Tiếng Anh A11	Nghệ An	1
7	2	Tran Van B	4	Lập trình C++	hải Phòng	<null>
8	2	Tran Van B	5	Triết học	Bắc Ninh	1

↓ on Student.STUDENT_ID = Book.STUDENT_ID

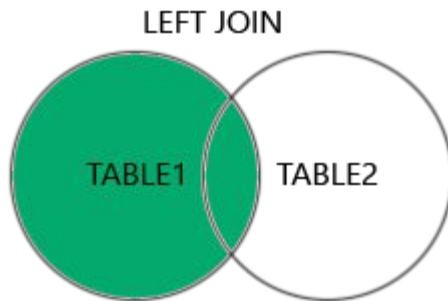
	Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1	Nguyen Van A	2	Tiếng Anh A11	Nghệ An	1
2	1	Nguyen Van A	5	Triết học	Bắc Ninh	1
3	2	Tran Van B	1	Toán Cao Cấp	Hà Nội	2

↓ select Student.NAME, Book.NAME

	Student.NAME	Book.NAME
1	Nguyen Van A	Tiếng Anh A11
2	Nguyen Van A	Triết học
3	Tran Van B	Toán Cao Cấp

Câu lệnh Join

- **Câu lệnh Left join:** Left join là câu lệnh kết hợp hai bảng với nhau và trả về các hàng từ bảng bên trái và các hàng từ bảng phải có giá trị khớp, nếu bản ghi bảng bên trái không có giá trị khớp với bảng bên phải, các cột của bảng phải sẽ có giá trị NULL.



Câu lệnh Join

- **Ví dụ:** Liệt kê tất cả các sinh viên và tên sách mà họ đã mượn (nếu có). Đối với sinh viên không mượn sách, hiển thị NULL trong cột tên sách.

```
select Student.NAME, Book.NAME
```

```
from Student
```

```
left join Book on Student.STUDENT_ID = Book.STUDENT_ID
```

Student		Book			
STUDENT_ID	NAME	BOOK_ID	NAME	PUBLISH_AT	STUDENT_ID
1	Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	Tran Van B	2	Tiếng Anh A11	Nghệ An	<null>
		3	4 Lập trình C++	hải Phòng	1



Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1 Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	1 Nguyen Van A	2	Tiếng Anh A11	Nghệ An	<null>
3	1 Nguyen Van A	4	Lập trình C++	hải Phòng	1
4	2 Tran Van B	1	Toán Cao Cấp	Hà Nội	1
5	2 Tran Van B	2	Tiếng Anh A11	Nghệ An	<null>
6	2 Tran Van B	4	Lập trình C++	hải Phòng	1



on Student.STUDENT_ID = Book.STUDENT_ID

Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1 Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	1 Nguyen Van A	4	Lập trình C++	hải Phòng	1



+ với các bản ghi còn lại trong bảng

Student (bên trái) không khớp

Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1 Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	1 Nguyen Van A	4	Lập trình C++	hải Phòng	1
3	2 Tran Van B	<null>	<null>	<null>	<null>

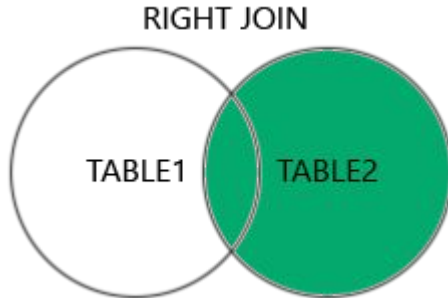


select Student.NAME, Book.NAME

Student.NAME	Book.NAME
1 Nguyen Van A	Toán Cao Cấp
2 Nguyen Van A	Lập trình C++
3 Tran Van B	<null>

Câu lệnh Join

- **Câu lệnh Right join:** Right join là câu lệnh kết hợp hai bảng với nhau và trả về các hàng từ bảng bên trái và các hàng từ bảng phải có giá trị khớp, nếu bản ghi của bảng bên phải không có giá trị khớp với bảng bên trái, các cột của bảng trái sẽ có giá trị NULL.



Câu lệnh Join

- **Ví dụ:** Liệt kê tất cả các sách và tên sinh viên mà đã mượn sách đó (nếu có). Đối với sách không được mượn, hiển thị NULL trong cột tên sinh viên.

```
select Book.NAME, Student.NAME
```

```
from Student
```

```
right join Book on Student.STUDENT_ID =
```

```
Book.STUDENT_ID
```

Student		Book			
STUDENT_ID	NAME	BOOK_ID	NAME	PUBLISH_AT	STUDENT_ID
1	1 Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	2 Tran Van B	2	Tiếng Anh A11	Nghệ An	<null>
		3	4 Lập trình C++	hải Phòng	1



Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1 Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	1 Nguyen Van A	2	Tiếng Anh A11	Nghệ An	<null>
3	1 Nguyen Van A	4	Lập trình C++	hải Phòng	1
4	2 Tran Van B	1	Toán Cao Cấp	Hà Nội	1
5	2 Tran Van B	2	Tiếng Anh A11	Nghệ An	<null>
6	2 Tran Van B	4	Lập trình C++	hải Phòng	1



on Student.STUDENT_ID = Book.STUDENT_ID

Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1 Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	1 Nguyen Van A	4	Lập trình C++	hải Phòng	1



+ với các bản ghi còn lại trong bảng

Book (bên phải) không khớp

Student.STUDENT_ID	Student.NAME	BOOK_ID	Book.NAME	PUBLISH_AT	Book.STUDENT_ID
1	1 Nguyen Van A	1	Toán Cao Cấp	Hà Nội	1
2	1 Nguyen Van A	4	Lập trình C++	hải Phòng	1
3	<null>	2	Tiếng Anh A11	Nghệ An	<null>



select Book.NAME, Student.NAME

Book.NAME	Student.NAME
1 Toán Cao Cấp	Nguyen Van A
2 Lập trình C++	Nguyen Van A
3 Tiếng Anh A11	<null>

Recap

- ER Diagram
- Database relationship
- JOIN clause

Homework

- Hãy truy vấn lấy ra các thông tin của sinh viên và số lượng môn học mà mỗi sinh viên đã học.
- Hãy lấy ra thông tin sinh viên và thông tin thẻ sinh viên tương ứng với mỗi từng sinh viên.
- Hãy lấy ra thông tin các môn học và số lượng sinh viên đã học môn học đó.

Learning Resources

- Document:
 - [SQL Tutorial - w3resource](#)
- Practices:
 - [SQLBolt](#)
 - [Practice SQL](#)
 - [SQLPad](#)

Thank you, ladies 🙏

Q&A

