



Khóa học Backend

Bài 45: Học MySQL và Sequelize (Tiết 1)



Nội dung

01

Khái niệm MySQL

...

02

Kiến thức MySQL cơ bản
(Tự nghiên cứu)

...

03

Giới thiệu project:
Tour du lịch

...

04

Cấu hình Database

...

05

Khái niệm Sequelize

...

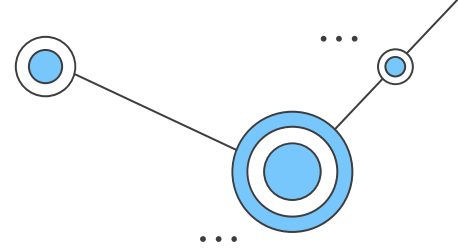


01

Khái niệm MySQL

1.1. Khái niệm

- MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ.
(RDBMS - Relational Database Management System).
- MySQL là nguồn mở và miễn phí.
- MySQL lý tưởng cho cả ứng dụng nhỏ và lớn.



1.2. So sánh MySQL với MongoDB

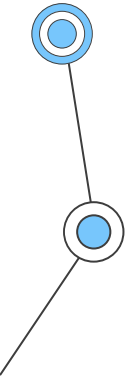
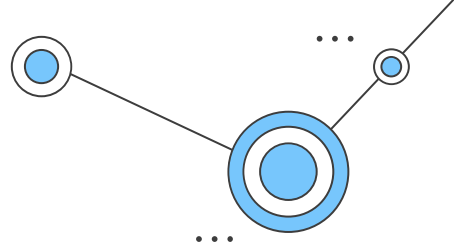
So sánh	MySQL	MongoDB
Mô hình dữ liệu	Sử dụng mô hình dữ liệu quan hệ (dữ liệu được tổ chức vào các bảng có mối quan hệ với nhau thông qua khóa ngoại).	Sử dụng mô hình dữ liệu không quan hệ, hướng tài liệu (dữ liệu được tổ chức thành các tài liệu JSON).
Thuật ngữ	Database - Table - Row - Column	Database - Collection - Document - Field
Ngôn ngữ truy vấn	Sử dụng ngôn ngữ truy vấn SQL để truy xuất và thao tác dữ liệu.	Sử dụng cú pháp truy vấn JSON-like, không giống SQL.
Dữ liệu động	Yêu cầu định nghĩa các Column cố định trước khi lưu trữ dữ liệu.	Cho phép lưu trữ dữ liệu mà không cần định nghĩa trước các Field.

02

Kiến thức MySQL cơ bản (Tự nghiên cứu)

2. Kiến thức MySQL cơ bản (**Tự nghiên cứu**)

- Link tự nghiên cứu: <https://dacavn.notion.site/Ki-n-th-c-MySQL-c-b-n-2fb5e1c25cb24f1b9c5b44b74fa1c2ee>

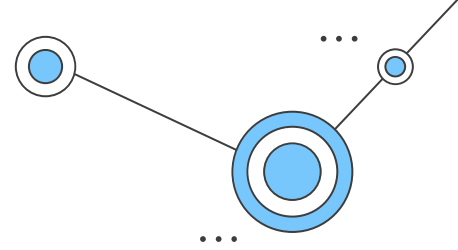


03

**Giới thiệu project:
Tour du lịch**

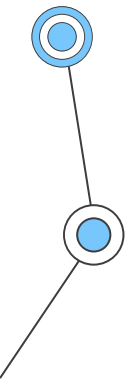
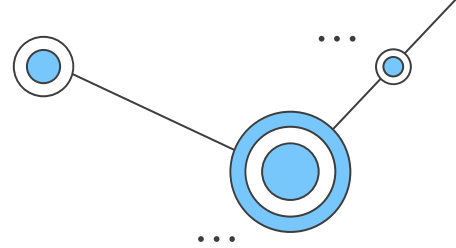
3.1. Bên admin

- Trang Tổng quan (Tự làm)
- Trang Quản lý danh mục
 - Danh sách danh mục
 - Thêm danh mục (Tự làm)
 - Sửa danh mục (Tự làm)
 - Xóa danh mục (Tự làm)
- Trang Quản lý tour
 - Danh sách tour
 - Thêm tour
 - Sửa tour (Tự làm)
 - Xóa tour (Tự làm)
- Trang Quản lý đơn hàng (Tự làm)
 - Danh sách đơn hàng
 - Xem chi tiết đơn hàng

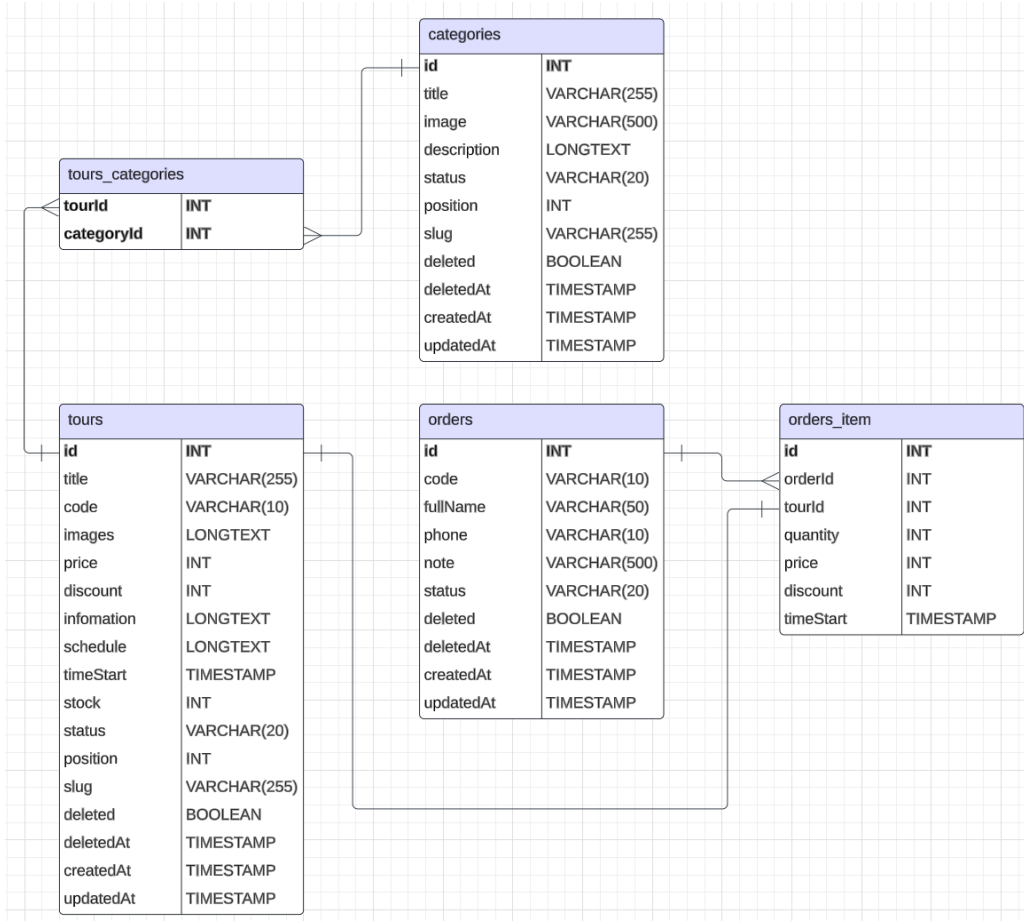


3.2. Bên client

- Trang chủ (**Tự làm**)
 - Hiển thị danh mục tour
 - Hiển thị danh sách tour
- Trang Danh sách tour theo danh mục
- Trang Chi tiết tour
- Trang Giỏ hàng + Đặt tour
 - Danh sách Tour
 - Tổng đơn hàng
 - Thông tin khách hàng
- Trang Đặt hàng thành công



3.3. Sơ đồ ERD cho project Tour du lịch

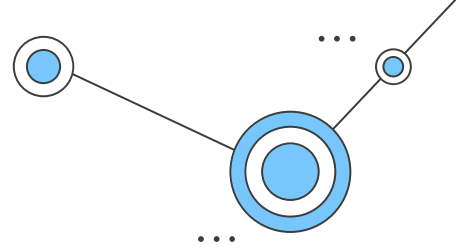


04

Cấu hình Database

4.1. Tạo Database

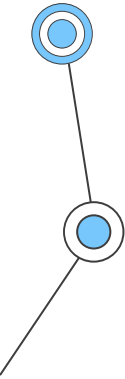
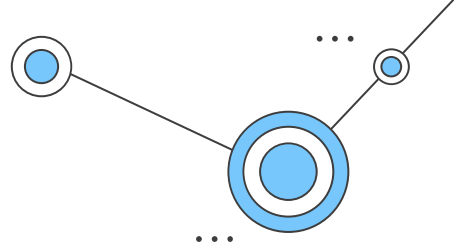
- Tải XAMPP: <https://www.apachefriends.org/download.html>



4.2. Tạo bảng

- Tạo bảng **tours**

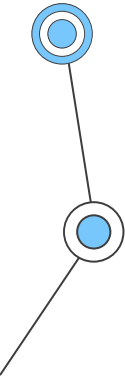
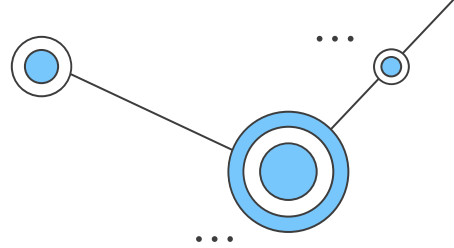
```
CREATE TABLE tours (  
  id INT NOT NULL AUTO_INCREMENT,  
  title VARCHAR(255) NOT NULL,  
  code VARCHAR(10),  
  images LONGTEXT,  
  price INT,  
  discount INT,  
  information LONGTEXT,  
  schedule LONGTEXT,  
  timeStart TIMESTAMP NULL,  
  stock INT,  
  status VARCHAR(20),  
  position INT,  
  slug VARCHAR(255) NOT NULL,  
  deleted BOOLEAN,  
  deletedAt TIMESTAMP NULL,  
  createdAt TIMESTAMP NULL,  
  updatedAt TIMESTAMP NULL,  
  PRIMARY KEY (id)  
);
```



4.2. Tạo bảng

- Tạo bảng **categories**

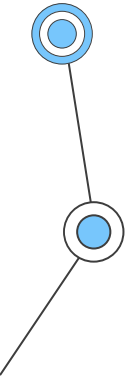
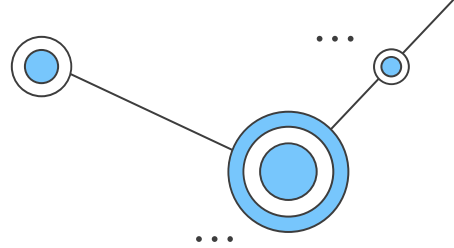
```
CREATE TABLE categories (  
  id INT NOT NULL AUTO_INCREMENT,  
  title VARCHAR(255) NOT NULL,  
  image VARCHAR(500),  
  description LONGTEXT,  
  status VARCHAR(20),  
  position INT,  
  slug VARCHAR(255) NOT NULL,  
  deleted BOOLEAN,  
  deletedAt TIMESTAMP NULL,  
  createdAt TIMESTAMP NULL,  
  updatedAt TIMESTAMP NULL,  
  PRIMARY KEY (id)  
);
```



4.2. Tạo bảng

- Tạo bảng **tours_categories**

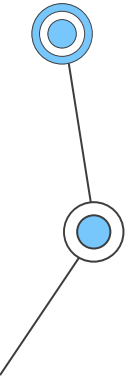
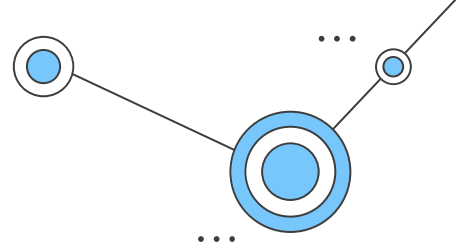
```
CREATE TABLE tours_categories (  
  tour_id INT,  
  category_id INT,  
  PRIMARY KEY (tour_id, category_id),  
  FOREIGN KEY (tour_id) REFERENCES tours(id),  
  FOREIGN KEY (category_id) REFERENCES categories(id)  
);
```



4.2. Tạo bảng

- Tạo bảng **orders**

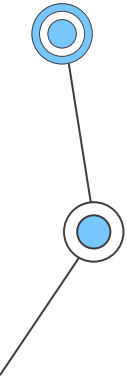
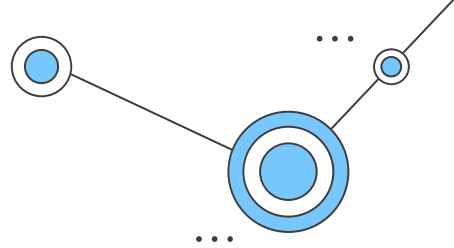
```
CREATE TABLE orders (  
  id INT NOT NULL AUTO_INCREMENT,  
  code VARCHAR(10) NOT NULL,  
  fullName VARCHAR(50) NOT NULL,  
  phone VARCHAR(10) NOT NULL,  
  note VARCHAR(500),  
  status VARCHAR(20),  
  deleted BOOLEAN,  
  deletedAt TIMESTAMP NULL,  
  createdAt TIMESTAMP NULL,  
  updatedAt TIMESTAMP NULL,  
  PRIMARY KEY (id)  
);
```



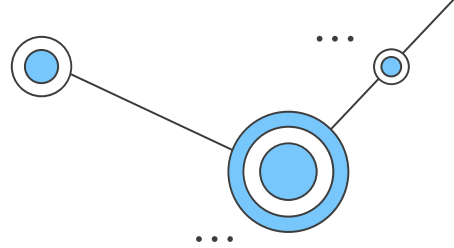
4.2. Tạo bảng

- Tạo bảng **orders_item**

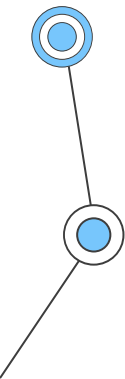
```
CREATE TABLE orders_item (  
  id INT NOT NULL AUTO_INCREMENT,  
  orderId INT NOT NULL,  
  tourId INT NOT NULL,  
  quantity INT NOT NULL,  
  price INT NOT NULL,  
  discount INT,  
  timeStart TIMESTAMP NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (orderId) REFERENCES orders(id),  
  FOREIGN KEY (tourId) REFERENCES tours(id)  
);
```



4.3. Insert data mẫu



- Data mẫu cho bảng **tours**: <https://backend.daca.vn/assets/tour/sql/tours.sql>
- Data mẫu cho bảng **categories**: <https://backend.daca.vn/assets/tour/sql/categories.sql>
- Data mẫu cho bảng **tours_categories**: https://backend.daca.vn/assets/tour/sql/tours_categories.sql



4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)

- Lấy ra danh sách tour (Chưa bị xóa, trạng thái active, trả thêm cột giá đặc biệt).

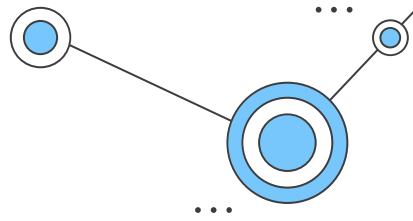
```
SELECT *, price * (1 - discount/100) AS price_special
FROM tours
WHERE
    deleted = false
    AND status = 'active';
```

4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)

- Lấy ra danh sách tour theo slug của categories (Chưa bị xóa, trạng thái active, trả thêm cột giá đặc biệt).

```
SELECT tours.*, price * (1 - discount/100) AS price_special
FROM tours
JOIN tours_categories ON tours.id = tours_categories.tour_id
JOIN categories ON tours_categories.category_id = categories.id
WHERE
  categories.slug = 'du-lich-trong-nuoc'
  AND categories.deleted = false
  AND categories.status = 'active'
  AND tours.deleted = false
  AND tours.status = 'active';
```

4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)



- Lấy ra danh sách danh mục và số lượng tour trong mỗi danh mục.

```
SELECT categories.title, COUNT(tours.id) as tour_count
FROM categories
LEFT JOIN tours_categories ON categories.id = tours_categories.category_id
LEFT JOIN tours ON tours_categories.tour_id = tours.id
GROUP BY categories.id;
```



4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)

- Lấy ra danh sách tour theo các trang 1, 2, 3...

```
-- Trang 1
SELECT *
FROM tours
WHERE deleted = false AND status = 'active'
ORDER BY createdAt DESC
LIMIT 3 OFFSET 0;
```

```
-- Trang 2
SELECT *
FROM tours
WHERE deleted = false AND status = 'active'
ORDER BY createdAt DESC
LIMIT 3 OFFSET 3;
```

```
-- Trang 3
SELECT *
FROM tours
WHERE deleted = false AND status = 'active'
ORDER BY createdAt DESC
LIMIT 3 OFFSET 6;
```

```
-- Trang 4
SELECT *
FROM tours
WHERE deleted = false AND status = 'active'
ORDER BY createdAt DESC
LIMIT 3 OFFSET 9;
```

```
-- LIMIT n OFFSET ((page_current - 1) * n)
```

4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)

- Lấy ra danh sách tour có ngày khởi hành từ 01/10/2023 đến ngày 30/10/2023.

```
SELECT *  
FROM tours  
WHERE  
  deleted = false  
  AND status = 'active'  
  AND timeStart BETWEEN '2023-10-01 00:00:00' AND '2023-10-30 23:59:59';
```


4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)

- Lấy ra danh sách tour theo slug của categories, có ngày khởi hành từ 01/01/2023 đến ngày 30/02/2023 (Chưa bị xóa, trạng thái active, trả thêm cột giá đặc biệt).

```
SELECT tours.*, price * (1 - discount/100) AS price_special
FROM tours
JOIN tours_categories ON tours.id = tours_categories.tour_id
JOIN categories ON tours_categories.category_id = categories.id
WHERE
  categories.slug = 'du-lich-trong-nuoc'
  AND categories.deleted = false
  AND categories.status = 'active'
  AND tours.deleted = false
  AND tours.status = 'active'
  AND timeStart BETWEEN '2023-01-01 00:00:00' AND '2023-02-30 23:59:59';
```

4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)

- Lấy ra danh sách tour theo slug của categories, có ngày khởi hành từ 01/05/2023 đến ngày hôm nay (Chưa bị xóa, trạng thái active, trả thêm cột giá đặc biệt).

```
SELECT tours.*, price * (1 - discount/100) AS price_special
FROM tours
JOIN tours_categories ON tours.id = tours_categories.tour_id
JOIN categories ON tours_categories.category_id = categories.id
WHERE
  categories.slug = 'du-lich-trong-nuoc'
  AND categories.deleted = false
  AND categories.status = 'active'
  AND tours.deleted = false
  AND tours.status = 'active'
  AND timeStart BETWEEN '2023-05-01 00:00:00' AND NOW();
```

4.4. Các câu truy vấn cơ bản (Tự nghiên cứu)

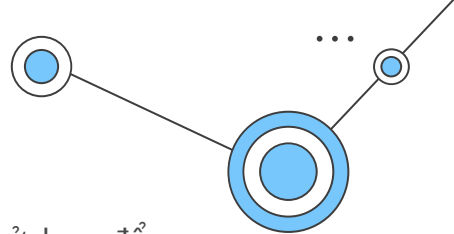
- Lấy ra danh sách tour theo slug của categories, có ngày khởi hành từ 01/05/2023 đến ngày hôm nay, sắp xếp theo giá từ thấp đến cao (Chưa bị xóa, trạng thái active, trả thêm cột giá đặc biệt).

```
SELECT tours.*, price * (1 - discount/100) AS price_special
FROM tours
JOIN tours_categories ON tours.id = tours_categories.tour_id
JOIN categories ON tours_categories.category_id = categories.id
WHERE
    categories.slug = 'du-lich-trong-nuoc'
    AND categories.deleted = false
    AND categories.status = 'active'
    AND tours.deleted = false
    AND tours.status = 'active'
    AND timeStart BETWEEN '2023-05-01 00:00:00' AND NOW()
ORDER BY price ASC;
```

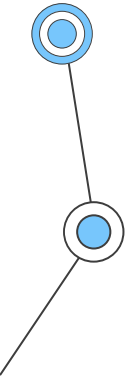
05

Khái niệm Sequelize

5.1. Khái niệm



- **Sequelize** là một thư viện ORM (Object-Relational Mapping) **dành cho Node.js** được sử dụng để **tương tác với cơ sở dữ liệu quan hệ**.
- Sequelize **cung cấp các hàm** để **truy vấn** dữ liệu **thuận tiện**, sử dụng cú pháp JavaScript thay vì truy vấn bằng các câu lệnh SQL trực tiếp.
- Giúp **bảo mật hơn** so với sử dụng câu lệnh SQL trực tiếp.



5.2. SQL Injection là gì?

- SQL Injection là một kỹ thuật tấn công phổ biến trong lĩnh vực bảo mật web, mà những người tấn công sử dụng để thực hiện các truy vấn SQL không an toàn vào cơ sở dữ liệu.
- Ví dụ: SQL Injection trong trường hợp đăng nhập vào trang admin.
 - **Trường hợp 1:** Người dùng **KHÔNG CÓ** ý định hack hệ thống.

Đăng nhập

Email

levana@gmail.com

Mật khẩu

.....

Đăng nhập

- Đoạn mã SQL để xác thực người dùng ở form đăng nhập là:

```
SELECT * FROM users WHERE email = 'levana@gmail.com' AND password = '123456';
```

- Nếu câu truy vấn tìm ra được user thì xác thực thành công.

5.2. SQL Injection là gì?

- SQL Injection là một kỹ thuật tấn công phổ biến trong lĩnh vực bảo mật web, mà những người tấn công sử dụng để thực hiện các truy vấn SQL không an toàn vào cơ sở dữ liệu.
- Ví dụ: SQL Injection trong trường hợp đăng nhập vào trang admin.
 - **Trường hợp 2:** Người dùng **CÓ** ý định hack hệ thống.

Đăng nhập

Email

Mật khẩu

Đăng nhập

- Đoạn mã SQL để xác thực người dùng ở form đăng nhập là:

```
SELECT * FROM users WHERE email = 'levana@gmail.com' OR '1'='1' -- ' AND password = '123456';
```

- Câu truy vấn luôn đúng vì **'1'='1'** luôn đúng → Vẫn truy cập được vào hệ thống.