

## Thiết kế cơ sở dữ liệu

Câu hỏi 19.1 Kiểu dữ liệu VARCHAR trong thiết kế dữ liệu là gì? Phát biểu nào sau đây không đúng?

A. VARCHAR luôn chiếm một lượng bộ nhớ cố định, bằng với độ dài tối đa đã khai báo.

**Giải thích:** Đây chính là đặc điểm của kiểu dữ liệu CHAR(n), không phải VARCHAR(n). VARCHAR(n) chỉ sử dụng lượng bộ nhớ vừa đủ để lưu trữ chuỗi ký tự thực tế cộng thêm một vài byte để lưu độ dài của chuỗi. Điều này giúp tiết kiệm không gian lưu trữ đáng kể so với CHAR khi độ dài chuỗi thực tế thường xuyên ngắn hơn độ dài tối đa đã khai báo. Các phát biểu B, C, và D đều đúng về kiểu dữ liệu VARCHAR.

B. VARCHAR là kiểu dữ liệu chuỗi ký tự có độ dài thay đổi.

C. VARCHAR(n) định nghĩa một chuỗi ký tự có độ dài tối đa là 'n' ký tự.

D. VARCHAR giúp tiết kiệm không gian lưu trữ so với kiểu CHAR khi độ dài chuỗi thực tế nhỏ hơn độ dài tối đa.

Câu hỏi 19.2 Các mối quan hệ giữa các bảng trong thiết kế dữ liệu thường bao gồm những kiểu nào? Phát biểu nào sau đây không đúng?

A. Mối quan hệ một-một (1-1), trong đó một bản ghi ở bảng này liên kết với tối đa một bản ghi ở bảng kia và ngược lại.

B. Mối quan hệ một-nhiều (1-N), trong đó một bản ghi ở bảng này có thể liên kết với nhiều bản ghi ở bảng kia, nhưng một bản ghi ở bảng kia chỉ có thể liên kết với một bản ghi ở bảng này.

C. Mối quan hệ nhiều-nhiều (N-N), trong đó nhiều bản ghi ở bảng này có thể liên kết với nhiều bản ghi ở bảng kia và ngược lại, thường được giải quyết bằng bảng trung gian.

D. Mối quan hệ không-một (0-1), trong đó một bản ghi ở bảng này có thể không liên kết hoặc liên kết với tối đa một bản ghi ở bảng kia. Đây là một kiểu quan hệ cơ bản thường được nhắc đến trong mọi mô hình dữ liệu.

**Giải thích:** Mối quan hệ "không-một" (0-1) thực chất là một dạng của mối quan hệ **một-một (1-1)** hoặc **một-không hoặc một (1-0 hoặc 1)**. Cách gọi phổ biến và cơ bản trong mô hình hóa dữ liệu thường tập trung vào ba kiểu chính: một-một, một-nhiều, và nhiều-nhiều. Mối quan hệ "không-một" mô tả tính tùy chọn của mối liên kết, nhưng về bản chất vẫn là một dạng của quan hệ một-một (với khả năng không có bản ghi liên kết). Nó không phải là một kiểu quan hệ cơ bản riêng biệt thường được liệt kê bên cạnh ba kiểu chính kia trong mọi mô hình dữ liệu.

## Làm việc với Postgresql

Câu hỏi 20.1 Postgresql là gì? Phát biểu nào sau đây không đúng?

A. Postgresql là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, mạnh mẽ và đáng tin cậy.

B. Postgresql là một cơ sở dữ liệu mã nguồn mở.

C. Postgresql tuân thủ chuẩn SQL và hỗ trợ đầy đủ các thuộc tính ACID cho giao dịch.

D. Postgresql hỗ trợ nhiều kiểu dữ liệu đa dạng, bao gồm JSON/JSONB, ARRAY và dữ liệu địa lý (PostGIS).

Câu hỏi 20.2 Bạn có thể sử dụng các công cụ nào để kết nối và làm việc với PostgreSQL? Phát biểu nào không đúng?

A. Dùng công cụ dòng lệnh: `psql`.

**B. Dùng giao thức ping**

**Giải thích:** Giao thức ping là một công cụ mạng cơ bản được sử dụng để kiểm tra khả năng kết nối mạng giữa hai thiết bị bằng cách gửi các gói tin ICMP echo request và lắng nghe phản hồi. Nó không phải là một giao thức hoặc công cụ được sử dụng để kết nối và làm việc trực tiếp với cơ sở dữ liệu PostgreSQL. Để kết nối và tương tác với PostgreSQL, bạn cần sử dụng các công cụ hoặc thư viện được thiết kế đặc biệt cho mục đích đó, như `psql`, `pgAdmin`, `DBeaver`, `JDBC`, `Sequelize`, v.v.

C. Dùng công cụ GUI: `pgAdmin` hoặc `DBeaver`.

D. API/Thư viện (gắn với một ngôn ngữ lập trình): `JDBC` hoặc `Sequelize`.

## Lập trình với cơ sở dữ liệu

Câu hỏi 21.1 Phát biểu nào sau đây KHÔNG đúng về Code First và Database First trong lập trình với cơ sở dữ liệu?

A. Code First là phương pháp định nghĩa mô hình dữ liệu trong mã nguồn trước, sau đó ORM sẽ tạo hoặc cập nhật schema cơ sở dữ liệu.

B. Database First là phương pháp thiết kế cơ sở dữ liệu trước, sau đó ORM sẽ sinh ra các mô hình trong mã nguồn dựa trên schema đã có.

**C. Cả Code First và Database First đều bỏ qua vai trò của ORM và cho phép lập trình viên tương tác trực tiếp với cơ sở dữ liệu bằng SQL.**

**Giải thích:** Cả Code First và Database First đều là các phương pháp làm việc với ORM (Object-Relational Mapping), chứ không hề bỏ qua vai trò của nó. ORM là cầu nối giữa các đối tượng trong mã nguồn và các bảng trong cơ sở dữ liệu.

- **Code First:** Lập trình viên định nghĩa các lớp (model) trong mã nguồn, và ORM sẽ sử dụng các định nghĩa này để tạo hoặc cập nhật schema cơ sở dữ liệu.
- **Database First:** Lập trình viên thiết kế cơ sở dữ liệu trước, sau đó ORM sẽ tạo ra các lớp (model) trong mã nguồn dựa trên schema đã có.

Cả hai phương pháp này đều tận dụng ORM để giảm thiểu việc viết SQL trực tiếp và giúp thao tác với cơ sở dữ liệu một cách hướng đối tượng hơn.

D. Code First thường phù hợp với các dự án mới, trong khi Database First thích hợp với các dự án làm việc trên cơ sở dữ liệu đã tồn tại.

Câu hỏi 21.2 Phát biểu nào sau đây KHÔNG đúng về ORM (Object-Relational Mapping)?

A. ORM là một kỹ thuật lập trình giúp ánh xạ giữa các đối tượng trong mã nguồn và các bảng trong cơ sở dữ liệu quan hệ.

**B. ORM làm tăng độ phức tạp của mã nguồn và giảm hiệu suất phát triển ứng dụng.**

**Giải thích:** Mặc dù ORM có thể mang lại một số overhead về hiệu suất trong một số trường hợp cụ thể (so với việc viết SQL tối ưu), nhưng nhìn chung, ORM **giúp giảm độ phức tạp** của mã nguồn bằng cách trừu tượng hóa việc tương tác với cơ sở dữ liệu. Nó cũng thường **tăng tốc độ phát triển** ứng dụng bằng cách cho phép lập trình viên làm việc với các đối tượng quen thuộc của ngôn ngữ lập trình thay vì phải viết và quản lý các câu lệnh SQL phức tạp. Các phát biểu A, C, và D đều đúng về ORM.

C. ORM cho phép lập trình viên thao tác với cơ sở dữ liệu bằng cú pháp của ngôn ngữ lập trình thay vì viết câu lệnh SQL trực tiếp.

D. Sequelize (Node.js) và Django ORM (Python) là những ví dụ về các ORM phổ biến

Câu hỏi 21.3 Bốn thư mục chính được tạo ra bởi lệnh sequelize init là config, models, migrations, và seeders. Phát biểu nào sau đây về mục đích của các thư mục này là KHÔNG đúng?

A. Thư mục migrations chứa các tập tin SQL script để thực hiện các thay đổi lược đồ cơ sở dữ liệu theo thời gian.

**Giải thích:** Thư mục migrations chứa các tập tin **JavaScript**, không phải SQL script, định nghĩa các thay đổi lược đồ cơ sở dữ liệu (ví dụ: tạo bảng, thêm cột, sửa đổi cột) theo thời gian. Sequelize sẽ đọc và thực thi các migration này để cập nhật schema cơ sở dữ liệu. Các thư mục config, models, và seeders đều được mô tả đúng trong các phát biểu B, C, và D.

B. Thư mục config chứa các tập tin cấu hình kết nối đến cơ sở dữ liệu cho các môi trường phát triển, kiểm thử và sản xuất.

C. Thư mục models chứa các tập tin định nghĩa cấu trúc các bảng trong cơ sở dữ liệu dưới dạng các model (lớp) JavaScript.

D. Thư mục seeders chứa các tập tin JavaScript để chèn dữ liệu mẫu ban đầu vào cơ sở dữ liệu.