|  |  |
| --- | --- |
| 1 | Cơ sở dữ liệu (database) là gì?  Cơ sở dữ liệu (Database) là một tập hợp các dữ liệu có tổ chức và có liên quan với nhau, được lưu trữ trên máy tính hoặc các thiết bị lưu trữ khác. Cơ sở dữ liệu được sử dụng để lưu trữ thông tin cho một hệ thống hoặc một ứng dụng, và cho phép người dùng lấy dữ liệu theo nhiều cách khác nhau.  Các cơ sở dữ liệu có thể được tổ chức theo nhiều cách khác nhau, tuy nhiên, chúng thường được tổ chức dưới dạng các bảng và mỗi bảng có các cột và hàng. Các bảng được sử dụng để lưu trữ các loại dữ liệu khác nhau, như các khách hàng, sản phẩm, đơn hàng, hóa đơn, v.v.  Các cơ sở dữ liệu cũng cung cấp các công cụ để truy vấn và quản lý dữ liệu, bao gồm các ngôn ngữ truy vấn cơ sở dữ liệu như SQL (Structured Query Language). SQL cho phép người dùng thực hiện các truy vấn để truy xuất dữ liệu từ cơ sở dữ liệu và thực hiện các thao tác như thêm, sửa đổi hoặc xóa dữ liệu từ các bảng.  Cơ sở dữ liệu được sử dụng trong nhiều ứng dụng khác nhau, bao gồm các hệ thống quản lý khách hàng, các hệ thống đặt hàng trực tuyến, các hệ thống quản lý dữ liệu nhân sự và nhiều ứng dụng khác. |
| 2 | Phân biệt các câu lệnh DML & DDL? DML, DDL bao gồm các câu lệnh nào?  Data Definition Language (DDL - Ngôn ngữ Định nghĩa dữ liệu):  Là nhóm các câu lệnh SQL được sử dụng để định nghĩa và quản lý các đối tượng cơ sở dữ liệu như bảng, cột, khóa chính, khóa ngoại, hàm, trigger, thủ tục, v.v.  Các câu lệnh DDL sẽ thay đổi cấu trúc của cơ sở dữ liệu.  Các câu lệnh DDL bao gồm: CREATE, ALTER và DROP.  Data Manipulation Language (DML - Ngôn ngữ Xử lý dữ liệu):  Là nhóm các câu lệnh SQL được sử dụng để truy cập và thao tác dữ liệu trong các bảng.  Các câu lệnh DML thường được sử dụng để thêm, sửa đổi, xóa và truy vấn dữ liệu.  Các câu lệnh DML bao gồm: SELECT, INSERT, UPDATE, DELETE. |
| 3 | Trình bày về mô hình thực thể mối quan hệ (ERD - Entity-Relationship Diagram)? Trong mô hình thực thể mối quan hệ có những thành phần nào?  Trong mô hình thực thể quan hệ có 3 thành phần đó là các thực thể, các thuộc tính của thực thể và các mối quan hệ giữa các thực tể với nhau |
| 4 | Các bước chuyển đổi từ ERD -> Table?  B1: Biến đổi các thực thể thành table  B2: Biến đổi các thuộc tính đa trị  B3: Biến đôi các mối quan hệ 1-1  B4: Biến đổi các mối quan hệ 1-n  B5: Biến đổi các mối quan hệ n-n |
| 5 | Các bước chuẩn hóa dữ liệu?  Bước 1 - Đưa dữ liệu vào các bảng:  Tách các thông tin lưu trữ dữ liệu đang được lưu trữ trong cùng một bảng và chia chúng thành các bảng khác nhau.  Bước 2 - Xác định các khóa chính:  Chọn một trường hoặc tập hợp các trường để đại diện cho khóa chính của mỗi bảng.  Bước 3 - Thiết lập mối quan hệ giữa các bảng:  Xác định mối quan hệ giữa các bảng bằng cách sử dụng các khóa ngoại, các trường chung và các quan hệ khác.  Bước 4 - Loại bỏ các phụ thuộc phi chức năng:  Phân tích các bảng để tìm ra các phụ thuộc phi chức năng, sau đó sửa chúng bằng cách tạo ra các bảng mới hoặc thay đổi cấu trúc bảng hiện có.  Bước 5 - Loại bỏ các lặp lại dữ liệu:  Phân tích các bảng để tìm ra các lặp lại dữ liệu và sửa chúng bằng cách tạo các bảng mới hoặc thay đổi cấu trúc bảng hiện có.  Bước 6 - Tối ưu hóa cơ sở dữ liệu:  Đảm bảo rằng cơ sở dữ liệu được tối ưu hóa để đáp ứng yêu cầu hiệu suất và khả năng mở rộng của ứng dụng. |
| 6 | Ràng buộc (constraint) là gì? Các loại ràng buộc trong MySQL?  Ràng buộc là các nguyên tắc phải tuân theo để được insert vào DB.  Một số ràng buộc như not null, unique, primary key, foreign key, check, default, create index |
| 7 | Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?  Primary key là khoá định danh duy nhất của mỗi record tại table của BD  Ngoài ra PK còn dùng để thiết lập quan hệ trong DB |
| 8 | Khóa ngoại (foreign key) là gì?  Khoá ngoại là con trỏ chỉ đến khoá chính trong 1 bảng khác có liên quan. Khoá ngoại là một trong các yếu tố giúp xây dựng quan hệ trong cơ sở dữ liệu cũng như tạo ra các ràng buộc giữa các bảng với nhau. |
| 9 | Phân biệt DELETE và TRUNCATE?  DELETE: Xoá 1 record  TRUNCATE: Cắt đi phần thập phân của 1 con số |
| 10 | Các loại JOIN? Phân biệt?  INNER JOIN: Chỉ lấy các phần chung của 2 bảng  OUTTER JOIN: Lấy phần chung và phần riêng của các bảng |
| 11 | Phân biệt JOIN và UNION?  JOIN: Kết hợp 2 hay nhiều bảng lại với nhau  UNION: Gộp nhiều kết quả truy vấn từ nhiều câu lệnh SELECT lại với nhau |
| 12 | Phân biệt UNION và UNION ALL?  UNION: Loại bỏ các record trùng lặp  UNION ALL: Không loại bỏ các record trùng lặp |
| 13 | Phân biệt EXIST và IN?  "EXIST" được sử dụng để kiểm tra xem có bất kỳ bản ghi nào trong bảng con hoặc câu truy vấn con trả về dữ liệu hay không. Nếu có ít nhất một bản ghi tồn tại, truy vấn sẽ trả về giá trị true hoặc false.  "IN" được sử dụng để xác định các giá trị trong một danh sách được chỉ định. Truy vấn sẽ trả về tất cả các bản ghi có giá trị của cột được chỉ định trùng khớp với một trong các giá trị trong danh sách |
| 14 | Phân biệt AND và OR?  AND: Các điều kiện đưa ra phải đồng thời trả thoả mãn  OR: 1 trong các điều kiện đưa ra được thoả mãn |
| 15 | Phân biệt GROUP BY và ORDER BY?  GROUP BY: Nhóm các kết quả theo giá trị giống nhau tại một cột chỉ định  ORDER BY: Sắp xếp các kết quả theo thứ tự của giá trị tại một cột chỉ định |
| 16 | Muốn nối chuỗi trong MySQL thì sử dụng gì?  CONCAT |
| 17 | Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?  Dùng hàm AVG() |
| 18 | Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?  Dùng hàm AVG() sau khi GROUP BY id |
| 19 | Yêu cầu hiển thị tên sinh viên và điểm trung bình tương ứng, chỉ hiển thị các bạn có điểm trung bình lớn hơn 5 thì làm thế nào?  GROUP BY id, dùng hàm AVG() để lấy giá trị điểm trung bình, sau đó dùng câu lệnh HAVING AVG()>5 |
| 20 | Hiển thị danh sách sinh viên và lớp học tương ứng thì làm thế nào (hai bảng Student và Class có cột chung ClassID)?  Sử dụng JOIN |
| 21 | Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi?  ORDER BY cột điểm thi |
| 22 | Hiển thị danh sách sinh viên theo thứ tự giảm dần của họ tên, nếu họ tên trùng nhau thì sắp xếp theo tăng dần của tuổi?  ORDER BY họ tên DESC, tuổi |
| 23 | Hiển thị các sinh viên có họ là ‘Le’ và tên là ‘Hai’?  SELECT \* FROM table WHERE ho=Le AND ten=Hai |
| 24 | Hiển thị các sinh viên có tên bắt đầu là ‘C’?  SELECT \* FROM sinh\_vien WHERE ten likes “^C” |
| 25 | Làm sao để có thể tăng tốc độ truy vấn? Nhược điểm là gì?  Để tăng tốc độ truy vấn, bạn có thể áp dụng các kỹ thuật tối ưu hóa cơ sở dữ liệu sau:  Chỉ chọn các cột cần thiết: Tránh truy xuất tất cả các cột nếu không cần thiết. Thay vào đó, hãy chỉ truy vấn các cột mà bạn thực sự cần.  Chỉ lấy các hàng cần thiết: Sử dụng các điều kiện để giới hạn số lượng hàng được truy vấn. Điều này giúp tránh tình trạng truy vấn toàn bộ bảng dữ liệu, tốn thời gian và tài nguyên.  Thêm chỉ mục (index) vào các cột được sử dụng thường xuyên: Chỉ mục giúp cơ sở dữ liệu tìm kiếm và trả về dữ liệu nhanh hơn.  Tối ưu hóa các truy vấn: Tối ưu hóa các truy vấn SQL bằng cách sử dụng các câu lệnh phù hợp và giảm thiểu việc truy vấn dữ liệu từ nhiều bảng khác nhau.  Sử dụng bộ đệm (cache): Sử dụng bộ đệm để lưu trữ dữ liệu được truy vấn thường xuyên, giảm thiểu số lần truy vấn đến cơ sở dữ liệu.  Tuy nhiên, một số nhược điểm có thể xảy ra khi tối ưu hóa cơ sở dữ liệu:  Chi phí tối ưu hóa ban đầu: Việc tối ưu hóa cơ sở dữ liệu có thể tốn kém và đòi hỏi thời gian và tài nguyên.  Sự phức tạp: Tối ưu hóa cơ sở dữ liệu có thể làm cho cấu trúc cơ sở dữ liệu phức tạp hơn, dẫn đến khó khăn trong việc bảo trì và quản lý.  Thay đổi dữ liệu: Nếu dữ liệu thay đổi thường xuyên, việc tối ưu hóa cơ sở dữ liệu có thể không còn hiệu quả nữa và cần phải được thực hiện lại. |
| 26 | Có 3 kiểu tham số: IN, OUT, INOUT |
| 27 | Phân biệt SP, Trigger, Function?  "Stored Procedure" (SP), "Trigger" và "Function" là các đối tượng trong cơ sở dữ liệu có chức năng khác nhau. Dưới đây là mô tả của từng đối tượng:  Stored Procedure (SP):  Là một khối mã lưu trữ trong cơ sở dữ liệu và có thể được gọi để thực thi.  Có thể được sử dụng để thực hiện một loạt các hoạt động, bao gồm truy vấn cơ sở dữ liệu, thực hiện tính toán, xử lý lỗi, và thậm chí có thể trả về kết quả cho ứng dụng.  Các thao tác được thực hiện bởi một Stored Procedure được lưu trữ lại trong cơ sở dữ liệu để có thể được gọi lại nhiều lần bởi các ứng dụng hoặc các truy vấn khác.  Trigger:  Là một đoạn mã được kích hoạt tự động khi một sự kiện nhất định xảy ra trên cơ sở dữ liệu.  Có thể được sử dụng để thực hiện một loạt các hoạt động, bao gồm kiểm tra tính hợp lệ của dữ liệu, cập nhật các trường khác trong bảng, hoặc thực hiện các thao tác liên quan đến cơ sở dữ liệu khác.  Function:  Là một đoạn mã có thể được gọi để thực hiện một tác vụ cụ thể và trả về kết quả.  Thường được sử dụng để thực hiện các tính toán hoặc xử lý dữ liệu.  Có thể được sử dụng trong các truy vấn SQL như một phần của điều kiện WHERE hoặc trong các biểu thức SELECT để tính toán các giá trị mới. |
| 28 | CSS là gì? Các cách nhúng CSS vào tài liệu html? Độ ưu tiên?  Cascade Style Sheet, Có 3 cách: sử dụng code inline, sử dụng thẻ <style> tại header của file html và nhúng file css.  Cái nào gần element hơn thì được ưu tiên hơn. |
| 29 | Phân biệt giữa bộ chọn class và bộ chọn id?  Bộ chọn class:  Bộ chọn class được định nghĩa bằng cách sử dụng dấu chấm (.) trước tên class và được đặt trong các cặp dấu ngoặc vuông []. Bộ chọn class cho phép chọn tất cả các phần tử HTML có cùng tên class và áp dụng các thuộc tính CSS cho chúng.  Bộ chọn id được định nghĩa bằng cách sử dụng dấu hashtag (#) trước tên id và được đặt trong cặp dấu ngoặc nhọn {}. Bộ chọn id cho phép chọn duy nhất một phần tử HTML với tên id cụ thể và áp dụng các thuộc tính CSS cho phần tử đó. |
| 30 | Trình bày mô hình hộp (box model) trong CSS?  Trong CSS, mỗi phần tử được xem như là 1 hộp (box) có chứa các thành phần: padding, border, content, giữa các phần tử có margin |
| 31 | Trình bày thuộc tính float, clear?  Thuộc tính float:  Thuộc tính float cho phép bạn di chuyển một phần tử HTML sang trái hoặc sang phải của trang web, để tạo ra một bố cục đa cột (multicolumn layout) hoặc để tạo ra các hộp chứa (containers) để bao quanh các phần tử khác.  Các giá trị của thuộc tính float bao gồm:  left: Di chuyển phần tử sang bên trái của trang web.  right: Di chuyển phần tử sang bên phải của trang web.  none: Không di chuyển phần tử nào và phần tử sẽ hiển thị ở vị trí mặc định của nó.  inherit: Kế thừa giá trị float từ phần tử cha của nó.  Thuộc tính clear:  Khi sử dụng thuộc tính float, một vấn đề thường gặp là các phần tử khác trong trang web sẽ không "nhìn thấy" được phần tử đó và sẽ bị chồng lên nhau. Để giải quyết vấn đề này, ta có thể sử dụng thuộc tính clear.  Thuộc tính clear xác định phần tử nào sẽ không được phép nằm cạnh phần tử đang được di chuyển bằng float. Nó đặt một giới hạn cho phần tử và ngăn chúng khỏi việc chồng lên phần tử đã được float.  Các giá trị của thuộc tính clear bao gồm:  none: Không có giới hạn nào được áp dụng.  left: Chỉ định rằng phần tử không được nằm bên trái của phần tử đã được float.  right: Chỉ định rằng phần tử không được nằm bên phải của phần tử đã được float.  both: Chỉ định rằng phần tử không được nằm bên trái hoặc bên phải của phần tử đã được float.  inherit: Kế thừa giá trị clear từ phần tử cha của nó. |
| 32 | Thuộc tính position dùng để làm gì?  Thuộc tính position quy định vị trí phần tử Phân biệt các giá trị: static, absolute, relative và fixed?  Static: giá trị mặt định: phần tử sẽ được canh chỉnh so với toàn bộ cửa sổ web  Absolute: phần tử sẽ được canh chỉnh so với phần tử cha  Relative: phần tử sẽ được canh chỉnh so với phần tử static  Fixed: phần tử được canh chỉnh so với cửa sổ trình duyệt |
| 33 | Responsive Web Design là gì?  Là một kỹ thuật thiết kế web nhằm hiển tị tốt trang web trên nhiều thiết bị với kích thước màn hình khác nhau. |
| 34 | Làm thế nào để tạo 1 trang web responsive?  Sử dụng Responsive Web Design (RWD): Đây là một kỹ thuật thiết kế web giúp trang web có thể thích ứng với các thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động. Kỹ thuật này sử dụng CSS để điều chỉnh bố cục, kích thước và bố trí các phần tử trên trang web sao cho chúng phù hợp với kích thước màn hình của thiết bị sử dụng.  Sử dụng các framework responsive: Có nhiều framework responsive như Bootstrap, Foundation, Materialize, v.v. chúng cung cấp các thành phần HTML/CSS/Javascript đã được thiết kế sẵn để giúp bạn dễ dàng tạo ra trang web responsive.  Thử nghiệm trên các thiết bị khác nhau: Kiểm tra trang web của bạn trên nhiều thiết bị và kích thước màn hình khác nhau để đảm bảo rằng nó đang hoạt động tốt trên mọi thiết bị.  Sử dụng các kỹ thuật CSS để điều chỉnh bố cục và bố trí các phần tử trên trang web, ví dụ như media queries để áp dụng các phong cách khác nhau cho các thiết bị khác nhau.  Sử dụng các ảnh có kích thước thích hợp: Sử dụng các ảnh có kích thước thích hợp cho từng thiết bị, để giảm thời gian tải trang và tối ưu trải nghiệm người dùng.  Tối ưu hóa hiệu suất của trang web: Sử dụng các kỹ thuật tối ưu hóa hiệu suất, như tối ưu hóa ảnh, tối ưu hóa tệp CSS và JavaScript, sử dụng cache để giảm thời gian tải trang.  Tạo trải nghiệm người dùng tốt: Thiết kế trang web để tạo ra trải nghiệm người dùng tốt, dễ sử dụng và truy cập trên mọi thiết bị. |
| 35 | Bootstrap là gì?  Bootstrap là 1 front-end framework miễn phí bao gồm các mẫu thiết kế sẵn dựa trên html/css/js.  Bootstrap cung cấp khả năng tạo ra các thiết kế RWD một cách nhanh chóng. |
| 36 | Một số thành phần thông dụng ở Bootstrap?  Table, button, form, carousel, card, modal, |
| 37 | Phân biệt Static web và Dynamic web?  Static web là các website ít tính tương tác với người dùng, không có liên kết với CSDL, thường chỉ được viết bằng các ngôn ngữ html,css, js  Dynamic web là các website có tính tương tác với người dùng, có kết nối và xử lý cơ sở dữ liệu, ngoài việc viết bằng html,css,js cho phía font-end thì còn được viết bằng java,php cho phía back-end, và có một hệ quản trị cơ sở dữ liệu đi kèm. |
| 38 | Phân biệt 2 phương thức GET/POST  Cả 2 điều là phương thức giao tiếp giữa client-server  GET: Tốc độ cao hơn, các thông tin được truyền tải dạng chuỗi thông qua thanh url, với số lượng ký tự có hạn, tính bảo mật thấp.  POST: Tốc độ thấp hơn, thông tin được truyền tải theo dạng gói tin, thông tin được truyền tải đi không có giới hạn, tính bảo mật cao. |
| 39 | Phân biệt request/response  Request (yêu cầu) là một tín hiệu được gửi từ client đến server, yêu cầu server thực hiện một hành động nào đó và trả về kết quả. Request bao gồm các thông tin như phương thức (method) của HTTP (ví dụ: GET, POST), địa chỉ URL (Uniform Resource Locator) của tài nguyên trên server mà client muốn truy cập, thông tin về các tham số truyền vào, các header của HTTP (ví dụ: user agent, cookie) và các thông tin khác.  Response (phản hồi) là kết quả trả về từ server cho client sau khi server đã thực hiện yêu cầu của client. Response bao gồm các thông tin như status code (mã trạng thái), header của HTTP (ví dụ: content type, cache control), nội dung phản hồi (ví dụ: HTML, JSON, file). |
| 40 | Phân biệt forward/redirect  Đều dùng để chuyển hướng.  Forward: chuyển hướng kèm theo dữ liệu, tốc độ nhanh hơn, không cập nhật url, chỉ dùng để gọi url trong server hiện tại.  Redirect: chuyển hướng không kèm dữ liệu, tốc độ chậm hơn, cập nhật url, có thể gọi đến liên kết bên ngoài server hiện tại. |
| 41 | JSP là gì? Nói: “Về bản chất JSP cũng chính là Servlet” đúng hay sai? Giải thích?  Đúng, vì JSP sẽ được biên dịch thành servlet rồi mới thực thi. So sánh giống và khác nhau giữa JSP và Servlet ?  Giống: đều là ngôn ngữ dùng để thiết kế web, đều nằm ở server, xử lý dữ liệu động  JSP: Code html, mạnh về giao diện, yếu về logic nên đóng vai trò View trong mô hình MVC  Servlet: Code Java, mạnh về logic, yếu về giao diện nên đóng vai trò Controller trong mô hình MVC |
| 42 | JSTL là gì? Giới thiệu một số thẻ bạn đã sử dụng ?  Java Standard Tag Lib: bộ thư viện thẻ chuẩn Java nhằm giúp tối ưu hoá việc lập trình trong JSP.  Một số thẻ hay dùng<c:out>,<c:forEch>,<c:if>,<c:choose>,<fmt:formatDate>,<fmt:formatNumber> |
| 43 | Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC? |
| 44 | Sự khác nhau giữa MVC1 và MVC2  MVC là viết tắt của Model-View-Controller (Mô hình-View-Bộ điều khiển), là một kiến trúc phần mềm được sử dụng để phân chia các thành phần của ứng dụng web thành các phần độc lập nhau. MVC được sử dụng để tách biệt các lớp chức năng và giúp đơn giản hóa quá trình phát triển ứng dụng web. Tuy nhiên, có sự khác nhau giữa hai phiên bản của kiến trúc MVC, đó là MVC1 và MVC2.  MVC1: Kiến trúc MVC1 là phiên bản đầu tiên của MVC, được sử dụng trong các ứng dụng web ban đầu. Trong kiến trúc này, Model chứa các lớp dữ liệu, View là các giao diện người dùng, và Controller là các lớp xử lý yêu cầu của người dùng và điều hướng đến các lớp Model hoặc View tương ứng. Mô hình này thường được triển khai bằng các Servlet và JSP (Java Server Pages).  MVC2: Kiến trúc MVC2 là một phiên bản nâng cao của MVC1, được phát triển để giải quyết một số vấn đề của phiên bản trước đó. Trong kiến trúc này, Model vẫn chứa các lớp dữ liệu, View là các giao diện người dùng, và Controller là các lớp xử lý yêu cầu của người dùng. Tuy nhiên, MVC2 tách biệt hoàn toàn giữa View và Controller, bằng cách sử dụng một lớp trung gian gọi là Front Controller (bộ điều khiển phía trước), được triển khai bằng Servlet. Front Controller nhận yêu cầu của người dùng và quyết định trang View nào sẽ được hiển thị. |
| 45 | Mô tả cách thực hiện tạo và lấy kết quả 1 câu query bất kỳ bằng JDBC?  B1: Khai báo driver  B2: Tạo connection có các thông tin cần thiết như hostname, username, password, db, port  B3: Sử dụng statement để tạo các câu lệnh truy vấn  B4: Sử dụng resultset để xử lý các kết quả trả về (nếu có)  B5: Đóng các kết nối |
| 46 | Phân biệt các loại Statement ?  Giống: đều là các câu lệnh Java nhằm thực hiện truy vấn dữ liệu  Statement: dùng tạo các câu truy vấn tĩnh, bảo mật thấp, hiệu suất thấp  PrepareStatement: dùng tạo các câu truy vấn động, bảo mật cao, hiệu suất cao  CallableStatement: thường được dùng để chạy các SP |
| 47 | Kỹ thuật sử dụng transaction bằng JDBC là như thế nào?  Trong Java, kỹ thuật sử dụng transaction bằng JDBC có thể được thực hiện theo các bước sau:  Tạo kết nối đến cơ sở dữ liệu bằng JDBC.  Bật chế độ tự động commit bằng cách gọi phương thức setAutoCommit(false) trên đối tượng kết nối.  Thực hiện các hoạt động truy vấn dữ liệu bằng các đối tượng Statement hoặc PreparedStatement.  Sau khi hoàn thành các hoạt động truy vấn, gọi phương thức commit() để lưu trữ các thay đổi vào cơ sở dữ liệu. Nếu có lỗi xảy ra, sử dụng phương thức rollback() để hoàn tác các thay đổi đã được thực hiện.  Khi đã hoàn tất các hoạt động truy vấn, đóng kết nối đến cơ sở dữ liệu bằng cách gọi phương thức close() trên đối tượng kết nối. |
| 48 | Các tính chất của Transaction?  Transaction là một khái niệm quan trọng trong cơ sở dữ liệu, đại diện cho một chuỗi các hoạt động cần được thực hiện như một đơn vị duy nhất. Các tính chất quan trọng của Transaction là:  ACID: ACID là viết tắt của Atomicity, Consistency, Isolation và Durability. Đây là bốn tính chất quan trọng của transaction. Atomicity đảm bảo rằng một transaction được thực hiện hoàn toàn hay không được thực hiện một chút nào cả. Consistency đảm bảo rằng transaction sẽ đưa dữ liệu từ một trạng thái hợp lệ đến trạng thái hợp lệ khác. Isolation đảm bảo rằng một transaction không ảnh hưởng đến các transaction khác đang được thực hiện cùng một lúc. Durability đảm bảo rằng các thay đổi được thực hiện bởi transaction sẽ được lưu trữ một cách an toàn và sẽ không bị mất khi hệ thống gặp sự cố.  Consistency: Một transaction cần đảm bảo rằng dữ liệu được đưa từ trạng thái hợp lệ đến trạng thái hợp lệ khác. Điều này có nghĩa là nếu một transaction thất bại, dữ liệu sẽ được phục hồi về trạng thái trước khi transaction được thực hiện.  Isolation: Mỗi transaction cần được thực hiện độc lập với các transaction khác. Điều này đảm bảo rằng các transaction khác không ảnh hưởng đến kết quả của transaction đang được thực hiện.  Durability: Các thay đổi được thực hiện bởi transaction phải được lưu trữ một cách an toàn và không bị mất khi hệ thống gặp sự cố.  Atomicity: Một transaction phải được thực hiện hoàn toàn hay không được thực hiện một chút nào cả. Nếu một phần của transaction thất bại, toàn bộ transaction phải được hoàn tác.  Serializability: Một tập hợp các transaction được coi là serializable nếu kết quả của việc thực hiện các transaction đó có thể được đạt được bằng cách thực hiện các transaction đó một cách tuần tự.  Những tính chất này là rất quan trọng trong việc đảm bảo tính toàn vẹn của dữ liệu và độ tin cậy của hệ thống cơ sở dữ liệu. |
| 49 | Phương pháp gọi 1 SP ở JDBC?  Để gọi một stored procedure (SP) trong Java, bạn có thể sử dụng đối tượng CallableStatement của JDBC. Đối tượng CallableStatement là một đối tượng được sử dụng để thực hiện các lệnh SQL hoặc các stored procedure được lưu trữ trong cơ sở dữ liệu. |
| 50 | Phân biệt Ant, Maven, Gradle?  Ant, Maven và Gradle đều là các công cụ quản lý dự án và xây dựng phần mềm. Tuy nhiên, chúng có những khác biệt cơ bản về cách thức hoạt động và tính năng.  Ant: Apache Ant là một công cụ xây dựng và tổ chức dự án phần mềm. Ant sử dụng file XML để định nghĩa các tác vụ xây dựng, và các tác vụ này được thực thi theo thứ tự được định nghĩa trong file build.xml. Ant là một công cụ đơn giản và linh hoạt, có thể được sử dụng để xây dựng bất kỳ loại dự án phần mềm nào.  Maven: Apache Maven là một công cụ quản lý dự án và xây dựng phần mềm. Maven sử dụng file pom.xml để định nghĩa các tác vụ xây dựng và quản lý các phụ thuộc của dự án. Maven cung cấp nhiều tính năng hữu ích như quản lý thư viện, tự động tải về các phụ thuộc, tự động tạo ra các bản phát hành, và cung cấp khả năng kiểm tra phụ thuộc trùng lặp.  Gradle: Gradle là một công cụ quản lý dự án và xây dựng phần mềm tương tự Maven. Tuy nhiên, Gradle sử dụng một DSL (Domain Specific Language) để định nghĩa các tác vụ xây dựng, và nó cũng hỗ trợ các tính năng như các tác vụ xây dựng đa nhiệm, xây dựng động, và khả năng tùy chỉnh các tác vụ xây dựng. |