**Audit module 3**

Họ và tên: Đặng Văn Nam. Ngày 21/01/2023

|  |  |
| --- | --- |
| 1. **Cơ sở dữ liệu (database) là gì?** | * Cơ sở dữ liệu là nơi lưu trữ dữ liệu của 1 trang web. |
| 1. **Phân biệt các câu lệnh DML & DDL? DML, DDL bao gồm các câu lệnh nào?** | * DML (Data Manipulation Language): sử dụng để thực hiện các thao tác trên bảng. Các câu lệnh : select, insert, update, delete. * DDL (Data Definition Language): sử dụng để tạo sửa đổi hoặc xóa các đối tượng trong CSDL. Các câu lệnh : create, alter, drop, |
| 1. **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?** | * Là công cụ biểu diễn cấu trúc của một cơ sở dữ liệu quan hệ. * Các thành phần : * Entity (thực thể) là một đối tượng hoặc một tập các đối tượng có thuộc tính và đặc điểm riêng biệt. Được biểu diễn bằng hình chữ nhật. * Relationship (mối quan hệ) mô tả mối liên hệ giữa các thực thể. * Attribute (thuộc tính) là một đặc điểm của thực thể, mô tả thong tin về thực thể đó. Được biểu diễn bằng các hình tròn kết nối với thực thể. * Key (khóa) : là thuộc tính hoặc một tập hợp thuộc tính của một thực thể, được sử dụng để định danh duy nhất một bản ghi. Được biểu diễn bằng dấu gạch chân dước bên trong hình tròn biểu diễn thuộc tính. |
| 1. **Các bước chuyển đổi từ ERD -> Table?** | * Xác định các thực thể (entity) và tạo các bảng tương ứng. * Xác định thuộc tính (attribute) của mỗi thực thể và tạo các cột tương ứng. * Xác định khóa chính (key) của mỗi thực thể và tạo các ràng buộc khóa trong bảng. * Xác định mối quan hệ (relationship) giữa các thực thể và tạo khóa ngoại (foreign key) trong bảng. * Xác định các ràng buộc (contraint) khác của các thuộc tính và mối quan hệ và tạo các ràng buộc tương ứng. |
| 1. **Các bước chuẩn hóa dữ liệu?** | * Xác định các thực thể (entity) trong hệ thống và các thuộc tính attribute của chúng. * Xác định mối quan hệ (reletionship) giữa các thực thể. * Xác định khóa của từng thực thể.   + - * Khóa chính (primary key) mỗi thực thể có ít nhất một khóa chính duy nhất để xác định các bản ghi riêng biệt của bảng.       * Khóa ngoại (foriegn key) được sử dụng để kết nối giữa các bảng. * Tách các thuộc tính phụ thuộc thành bảng mới. quá trình tách giúp loại bỏ dữ liệu lặp lại trong bảng. * Xác định các ràng buộc của CSDL. |
| 1. **Ràng buộc (constraint) là gì? Các loại ràng buộc trong MySQL?** | * Là quy tắc được quy định cho bảng. * Sử dụng constraint để hạn chế những dữ liệu đưa vào bảng. * Giúp cho dữ liệu chính xác, tin cậy và toàn vẹn. * Các loại ràng buộc:   + - * NOT NULL: không cho phép giá trị NULL.       * UNIQUE: mỗi giá trị là duy nhất.       * PRIMARY KEY : khóa chính (không NULL và là UNIQUE).       * FOREIGN KEY: khóa ngoại (tham chiếu sang bảng khác).       * CHECK: kiểm tra dựa vào một điều kiện.       * DEFAULT: quy định giá trị mặc định cho trường (nếu không có giá trị nào được nhập vào).       * INDEX: giúp tăng tốc độ truy vấn dữ liệu. |
| 1. **Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?** | * Xác định một bản ghi duy nhất trong bảng. giúp đảm bảo tính toàn vẹn dữ liệu bằng cách đảm bảo không có 2 hàng trong bảng có giá trị khóa giống nhau. * Nghĩa là khóa chính có thể bao gồm một hoặc nhiều thuộc tính trong bảng dữ liệu. Nếu khóa chính chỉ bao gồm một thuộc tính thì được gọi là khóa chính đơn, nếu gồm nhiều thuộc tính thì được gọi là khóa chính kết hợp. |
| 1. **Khóa ngoại (foreign key) là gì?** | * Là cơ chế tạo liên kết giữa 2 bảng trong cùng CSDL. * Khóa ngoại được đặt trên một cột của bảng này và tham chiếu đến khóa chính của bảng khác. * Kiểu dữ liệu khóa ngoại và khóa chính phải giống nhau. |
| 1. **Phân biệt DELETE và TRUNCATE?** | * Cả 2 câu lệnh được sử dụng để xóa dữ liệu khỏi bảng. * Khác nhau:   + - * DELETE: là câu lệnh dùng để xóa dữ liệu trong bản dựa trên điều kiện được chỉ định. DELETE chỉ xóa các bản ghi thỏa mãn điều where được chỉ định, và không xóa toàn bộ bảng. Ngoài ra DELETE có thể phục hồi dữ liệu sau khi xóa.       * TRUNCATE: là câu lệnh dùng để xóa toàn bộ dữ liệu trong bảng. TRUNCATE sẽ xóa toàn bộ các bản ghi của bảng, và không có điều kiện where được sử dụng. ngoài ra TRUNCATE không thể phục hồi dữ liệu sau khi xóa |
| 1. **Các loại JOIN? Phân biệt?** | * Các loại câu lệnh JOIN :   + - * INNER JOIN (hoặc JOIN) : chỉ trả về các record có mặt ở cả 2 bảng.       * LEFT JOIN: trả về tất cả các record có mặt ở bảng bên trái và những record tương ứng ở bảng bên phải.       * RIGHT JOIN: trả về tất cả record có mặt ở bảng bên phải và những record tương ứng ở bảng bên trái.       * FULL JOIN: trả về tất cả record ở cả 2 bảng. |
| 1. **Phân biệt JOIN và UNION?** | * JOIN: là câu lệnh kết hợp dữ liệu từ nhiều bảng dữ liệu trong cùng một câu truy vấn. kết qả trả về của câu lệnh JOIN là một bảng mới, chứa thông tin được kết hợp từ bảng ban đầu. * UNION: là câu lệnh được sử dụng để kết hợp các bảng dữ liệu có cấu trúc giống nhau. Kết quả trả về của câu lệnh UNION là một bảng dữ liệu mới, chứa tất cả các bảng ghi của bảng được kết hợp. |
| 1. **Phân biệt UNION và UNION ALL?** | * UNION: sử dụng để kết hợp kết qả của 2 truy vấn SELECT và loại bỏ bất kì bản ghi nào trùng lặp. * UNION ALL: nhưng không loại bỏ bất kì bản ghi trùng lặp nào. |
| 1. **Phân biệt EXIST và IN?** | * Đều được sử dụng để kiểm tra tập hợp dữ liệu trả về từ một truy vấn con có chứa một giá trị cụ thể không.   + - * EXISTS: sử dụng để kiểm tra xem có bản ghi nào trong tập kết quả trả về từ truy vấn con không.       * IN: sử dụng để so sanh một giá trị với một danh sách các giá trị được trả về từ một truy vấn con. |
| 1. **Phân biệt AND và OR?** |  |
| 1. **Phân biệt GROUP BY và ORDER BY?** | * ORDER BY: sử dụng để sắp xếp các kết qả theo một hoặc nhiều cột trong bảng dữ liệu. nó chỉ định cột sắp xếp và thứ tự sắp xếp (ASC hoặc DESC). * GROUP BY: sử dụng để nhóm các kết qả trả về thành nhóm dựa trên giá trị một hoặc nhiều cột trong bảng dữ liệu. thường dùng chung với các hàm của SQL như COUNT(), MIN(), MAX(), SUM(), AVG(). |
| 1. **Muốn nối chuỗi trong MySQL thì sử dụng gì?** | * Sử dụng phép toán CONCAT(). |
| 1. **Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?** | * SELECT students.name, AVG(grades.grade) AS avg\_grade * FROM students * INNER JOIN grades ON students.id = grades.student\_id * GROUP BY students.name; |
| 1. **Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?** | * Sử dụng where để chỉ định sinh viên cần tính điểm trung bình.   SELECT students.name, AVG(grades.grade) AS avg\_grade  FROM students  INNER JOIN grades ON students.id = grades.student\_id  WHERE students.id = 1  GROUP BY students.name; |
| 1. **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?** | * Sử dụng HAVING để lấy bảng ghi có điểm trung bình lớn hơn 5.   SELECT students.name, AVG(grades.grade) AS avg\_grade  FROM students  INNER JOIN grades ON students.id = grades.student\_id  GROUP BY students.name  HAVING AVG(grades.grade) > 5; |
| 1. **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 câu lệnh JOIN:   SELECT students.name, classes.name AS class\_name  FROM students  INNER JOIN classes ON students.class\_id = classes.id; |
| 1. **Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi?** | * SELECT \* FROM Student * ORDER BY ExamScore ASC; |
| 1. **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?** | * SELECT \* FROM Student * ORDER BY Name DESC, Age ASC; |
| 1. **Hiển thị các sinh viên có họ là ‘Le’ và tên là ‘Hai’?** | * SELECT \* FROM Student * WHERE Name = 'Hai' AND LastName = 'Le'; |
| 1. **Hiển thị các sinh viên có tên bắt đầu là ‘C’?** | * SELECT \* FROM Student * WHERE Name LIKE 'C%'; |
| 1. **Làm sao để có thể tăng tốc độ truy vấn? Nhược điểm là gì?** | * Tạo chỉ mục (index) * Sử dụng các hàm tính toán cơ bản. * Chỉ lấy các cột cần thiết thay vì lấy tất cả các cột trong bảng.   Nhược điểm:   * Tăng khối lượng dữ liệu được lưu trong bộ nhớ, làm cho việc quản lý dữ liệu phức tạp. |
| 1. **Các dạng tham số ở SP?** |  |
| 1. **Phân biệt SP, Trigger, Function?** | * Stored Procedure(sp) sử dụng để thực hiện một loạt các lệnh SQL hoặc chức năng phức tạp hơn để truy xuất và cập nhật dữ liệu trên CSDL. * Trigger: là một loại đối tượng trong CSDL được sử dụng để tự động kích hoạt một hành động nào đó khi có sự kiện xảy ra trên bảng hoặc CSDL. * Function: là một khối lệnh được chỉ định để thực hiện một chức năng cụ thể và trả về một kết quả. |
| 1. **CSS là gì? Các cách nhúng CSS vào tài liệu html? Độ ưu tiên?** | * Là viết tắt của cụm từ Cascading Style Sheets. * Để mô tả cách các phần tử HTML sẽ hiển thị ra màn hình, trang web, hoặc các phương tiện khác. * Cách nhúng:   + - * Inline style: khai báo trực tiếp ở thẻ HTML.       * Internal style sheet: khai báo trong thẻ <style> của tài liệu HTML.       * External style sheet: khai báo trong file.css riêng biệt. * Độ ưu tiên:  1. Inline style 2. External và Internal Style. 3. Kiểu mặc định của trình duyệt. |
| 1. **Phân biệt giữa bộ chọn class và bộ chọn id?** | * Tag selector: Chọn tất cả các phần tử * Id selector: chọn một phần tử với id xác định. * Class selector: chọn tất cả các phần tử xác định. * Độ ưu tiên   + - * Id -> Class -> tag |
| 1. **Trình bày mô hình hộp (box model) trong CSS?** | * Content (nội dung): là phần bên trong hộp, chứa nội dung thực sự của phần tử. * Padding (lề) là khoảng cách giữa nội dung và biên của hộp, được xác định bằng thuộc tính padding trong CSS. * Border(viền): là đường viền xung quanh hộp, được xác định bằng thuộc tính border trong CSS. * Margin (khoảng cách) là khoảng cách giữa hộp hiện tại và phần tử khác trên trang web, được xác định bằng thuộc tính Margin trong CSS |
| 1. **Trình bày thuộc tính float, clear?** | * Float sử dụng để căn lề một phần tử về phía trái hoặc phải của phần tử cha. * Clear: xác định xem phần tử đang được xếp chồng lên phần tử nào. |
| 1. **Thuộc tính position dùng để làm gì?**   **Phân biệt các giá trị: static, absolute, relative và fixed?** | * Sử dụng để xác định vị trí của một phần tử HML trên trang web. * Các giá trị của thuộc tính position:   + - * Static: giá tri mặc định của position, không bị ảnh hưởng bởi các thuộc tính top, bottom, left, right.       * Relative: được định vị tương đối so với vị trí bình thường của nó, hay nói cách khác relative sẽ định vị một thành phần so với vị trí mặc định của chính bản thân nó.       * Absolute:định vị một thành phần so với vị trí của thành phàn cha gần nhất của nó, nếu 1 phần tử được định vị absolute không có các phần tử tổ tiên được định vị, nó sử dụng phần tử và di chuyển cùng với di chuyển của trang.       * Fixed: định vị một phần so với trình duyệt, luôn luôn ở cùng 1 vị trí ngay cả khi trang được cuộn. |
| 1. **Responsive Web Design là gì?** | * Là kỹ thuật được sử dụng để trang web có thể hiển thị tốt trên nhiều loại thiết bị với kích thước của trình duyệt khác nhau. |
| 1. **Làm thế nào để tạo 1 trang web responsive?** |  |
| 1. **Bootstrap là gì?** | * Là một front-end framework miễn phí để phát triển web nhanh hơn và dễ dàng hơn. * Gồm các mẫu thiết kế dự trên HTML và CSS. * Cung cấp khả năng tạo các thiết kế responsive. |
| 1. **Một số thành phần thông dụng ở Bootstrap?** |  |
| 1. **Phân biệt Static web và Dynamic web?** | * Static web là loại website cơ bản được tạo mà không cần phải biết đến các ngôn ngữ lập trình web như java, php, hoặc thiết kế CSDL.   Được viết bằng mã HTML hoặc thêm CSS, js để thêm hiệu ứng.   * Dynamic web: web động là website có nội dung thay đổi.   Sự thay đổi có thể là tùy thời gian, người dùng, ngữ cảnh.  Thường sử dụng đến các ngôn ngữ phía server như servlet, jsp, php, python, c# |
| 1. **Phân biệt 2 phương thức GET/POST** | * Phương thức get: dùng để nhận dữ liệu, ít được sử dụng để gửi các dữ liệu nhạy cảm, và có giới hạn về độ dài cũng như kiểu dữ liệu.   It bảo mật hơn post do dữ liệu được hiển thị trên URL.   * Phương thức post: dùng để gửi các dữ liệu nhạy cảm, và không giới hạn độ dài hay các kiểu dữ liệu.   An toàn hơn get một ít do dữ liệu không bị lưu lại trên trình duyệt hay trên server web log |
| 1. **Phân biệt request/response** | * Request là yêu cầu mà client gửi đến server để nhận được các tài nguyên gửi từ server. * Reponse là phản hồi từ server trả vè cho client sau khi nhận được yêu cầu (request) từ client. |
| 1. **Phân biệt forward/redirect** | * Forward: chuyển tiếp request tới 1 trang khác trên cùng 1 server và client không biết về viêc này. * Redirect là thông báo cho client để yêu cầu họ gửi lại request tới 1 địa chỉ URL khác. |
| 1. **JSP là gì? Nói: “Về bản chất JSP cũng chính là Servlet” đúng hay sai? Giải thích?**   **So sánh giống và khác nhau giữa JSP và Servlet ?** | * Là một ngôn ngữ kịch bản phía server, cho phép người dùng tạo ra các trang web động. * Đúng, jsp được biên dịch thành một servlet khi chạy trên máy chủ web. * Giống nhau: đều cho phép tạo ra các trang web động bằng cách sử dụng mã Java. Cả hai cũng truy xuất CSDL và tương tác với các tài nguyên khác trên máy chủ web. * Khác nhau: cú pháp. Servlet yêu cầu lập trình viên phải viết mã Java để xử lý yêu cầu và phản hồi của trình duyệt web, trong khi jsp cho phép nhúng mã java vào các trang web tĩnh bằng cú pháp html , css. |
| 1. **JSTL là gì? Giới thiệu một số thẻ bạn đã sử dụng ?** | * Là bộ thư viện thẻ chuẩn được bổ sung với mục đích tối ưu lập trình trong jsp. * Các thẻ jstl cơ bản:   <c:out>  <c:set>  <c:if>  <c:foreach>  <c:url |
| 1. **Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC?** | * Là viết tắt cảu Model-view-controller * Kiến trúc gồm 3 tầng:   + - * Tầng model: biểu diễn dữ liệu và các logic nghiệp vụ.       * Tầng view: hiển thị dữ liệu và là giao diện tương tác với người dùng.       * Tầng controller: xử lý các thao tác từ người dùng, kết nối giữa model và view |
| 1. **Sự khác nhau giữa MVC1 và MVC2** | * Trong MVC1, controller và view kết hợp với nhau để xử lý yêu cầu và hiển thị dữ liệu, trong khi đó MVC2, controller và view hoạt động độc lập và chuyển đổi dữ liệu thông qa model. * Trong MVC2 controller được thiết kế để cập nhật model, MVC1 các thao tác xử lý dữ liệu được thực hiện trực tiếp trong trong jsp. |
| 1. **Mô tả cách thực hiện tạo và lấy kết quả 1 câu query bất kỳ bằng JDBC?** |  |
| 1. **Phân biệt các loại Statement ?** | * Statement: interface này cung cấp nhiều phương thức để thực thi các truy vấn cơ sở dữ liệu và trả về kết quả mà nó tạo ra. Không an toàn về bảo mật. * PreparedStatement: interface là một interface con của statement. Nó sử dụng để thực thi các truy vấn tham số hóa. Tăng hiệu suất và an toàn cho ứng dụng vì tham số đầu vào trong câu lệnh sql được xử lý an toàn hơn. * CallableStatement: interface được sử dụng để thực thi stored procedure. |
| 1. **Kỹ thuật sử dụng transaction bằng JDBC là như thế nào?** |  |
| 1. **Các tính chất của Transaction?** | * • Atomicity nghĩa là tất cả thành công hoặc không. * • Consistency bảo đảm rằng tính đồng nhất của dữ liệu. * • Isolation bảo đảm rằng Transaction này là độc lập với Transaction khác. * Durability nghĩa là khi một Transaction đã được ký thác thì nó sẽ vẫn tồn tại như thế cho dù xảy ra các lỗi, |
| 1. **Phương pháp gọi 1 SP ở JDBC?** | * Sử dụng đối tượng CallableStatement. |
| 1. **Phân biệt Ant, Maven, Gradle?** | * Ant là công cụ được sử dụng trong quá trình biên dịch phần mềm, chủ yếu để thực hiện các tác vụ tự động hóa trong quá trình phát triển phần mềm. Ant sử dụng tệp XML để mô tả quá trình biên dịch. * Maven là một công cụ quản lý dự án, sử dụng tệp pom.xml để quản lý các phụ thuộc của dự án và xây dựng các phiên bản phần mềm. Maven được sử dụng để tự động hóa quá trình biên dịch, kiểm thử và triển khai phần mềm. * Gradle cũng là một công cụ quản lý dự án, nhưng nó sử dụng cú pháp DSL (Domain Specific Language) để mô tả quá trình biên dịch. Gradle được thiết kế để hỗ trợ nhiều ngôn ngữ lập trình và các nền tảng khác nhau. |