**BỘ CÂU HỎI PHỎNG VẤN MODULE 2**

**ADVANCED PROGRAMMING WITH JAVA**

|  |  |
| --- | --- |
| **STT** | **Câu hỏi** |
| 1 | **Dữ liệu (data) là gì? Phân biệt hai khái niệm dữ liệu (data) và thông tin (information)?**   * Dữ liệu (data) là tập hợp các dữ kiện, ví dụ như số, chữ, phép tính, quan sát hoặc mô tả về sự vật, hiện tượng… * Hay nói cách khác, dữ liệu là thông tin đã được mã hoá trong máy tính * Khác nhau: * Thông tin là những gì đem lại hiểu biết cho con người về thế giới xung quanh và chính bản thân mình. * Ví dụ:  Thông tin về thời tiết ngày mai, thông tin về trận bóng đá sắp diễn ra,... * Dữ liệu là thông tin được ghi lên vật mang tin. * Ví dụ: con số, hình ảnh, văn bản, âm thanh,.. |
| 2 | **Cơ sở dữ liệu (database) là gì?**   * Cơ sở dữ liệu(database) nhóm dữ liệu có tổ chức. |
| 3 | **Giới thiệu một số hệ quản trị cơ sở dữ liệu hiện nay?**   * MySQL, SQL |
| 4 | **Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) là gì?**   * Hệ Quản trị CSDL(DBMS–Database Management System) là một ứng dụng máy tính được sử dụng để quản lý CSDL. |
| 5 | **Toàn vẹn dữ liệu (data integrity) của một cơ sở dữ liệu là gì?**   * Toàn vẹn dữ liệu là việc đặt ra các quy tắc trong một cơ sở dữ liệu nhằm kiểm tra các giá trị của dữ liệu trước khi được lưu trữ phải đảm bảo tính chính xác và hợp lý bên trong một cơ sở dữ liệu. |
| 6 | **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 một mô hình được sử dụng rộng rãi trong các bản thiết kế cơ sở dữ liệu ở mức khái niệm, được xây dựng dựa trên việc nhận thức thế giới thực thông qua tập các đối tượng được gọi là các thực thể và các mối quan hệ giữa các đối tượng này * Thành phần   + Thực thể và tập thực thể   + Thuộc tính |
| 7 | **Ràng buộc (constraint) là gì? Các loại ràng buộc trong MySQL?**   * Constraint là các quy tắc (rule) được quy định cho bảng. * NOTNULL: 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 |
| 8 | **Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?**   * Khóa chính là một cột (hoặc nhiều cột) được sử dụng để xác định một bảng ghi duy nhất trong bảng * Khóa chính là UNIQUE và NOT NULL * Mỗi bảng chỉ có thể có 1 khóa chính * Từ khóa PRIMARY KEY được sử dụng để định nghĩa khóa chính * Khi một khoá chính chứa nhiều thuộc tính nó còn có tên gọi là composite key |
| 9 | **Khóa ngoại (foreign key) là gì?**   * Khóa ngoại là cơ chế để 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 một bảng khác * Kiểu dữ liệu của khóa chính và khóa ngoại phải giống nhau |
| 10 | **Trình bày về khóa duy nhất - unique? Phân biệt giữa Primary key và unique?**   * Tương tự như primary key, ràng buộc unique key cũng xác định duy nhất một bộ giá trị riêng lẻ trong một mối quan hệ. Nhưng, có những khác biệt nhất định giữa chúng. * Một bảng có thể có nhiều hơn một khóa duy nhất. Ràng buộc unique key chỉ có thể chấp nhận một giá trị NULL cho một cột. * PRIMARY KEY * Không chấp nhận null * UNIQUE * Chấp nhận null |
| 11 | **Thuộc tính not null là gì?**   * Nếu bạn không muốn một cột mà có một giá trị NULL, thì bạn cần định nghĩa ràng buộc NOT NULL trên cột này, để xác định rằng bây giờ NULL là không được chấp nhận bởi cột đó. |
| 12 | **Trình bày ràng buộc mặc định (default)? Ví dụ?**   * DEFAULT: ràng buộc mặc định. * Khi nhập dữ liệu cho bảng mà cột đó không được cung cấp giá trị thì giá trị mặc định sẽ được sử dụng. * Ví dụ bạn có thể tạo ràng buộc DEFAULT cho trường Ngay GD (ngày giao dịch) của bảng bán hàng là ngày giờ hệ thống. |
| 13 | **Trình bày ràng buộc kiểm tra (check)? Ví dụ?**   * Ràng buộc kiểm tra (CHECK) dùng để giới hạn giá trị dữ liệu nhập vào trong cột. Ví dụ, cột điểm của sinh viên chỉ lưu trữ các điểm số trong đoạn [0-10]. |
| 14 | **Trình bày cách tự động gán và tăng giá trị trong 1 cột của MySQL?**   * Thuộc tính AUTO\_INCREMENT thường được dùng trên những cột là khóa chính & có kiểu dữ liệu là số nguyên. * Chức năng của nó là tự động gán giá trị cho cột khi chúng ta thêm hàng mới vào bên trong bảng (giá trị của hàng đầu tiên được thêm vào bảng là 1, giá trị này sẽ tự động tăng lên một sau mỗi hàng được thêm vào bảng) |
| 15 | **Yêu cầu nhập vào trường tuổi của nhân viên phải từ 18-60, sử dụng ràng buộc nào? Viết ràng buộc?**  CHECK  BETWEEN 18 AND 60 |
| 16 | **Yêu cầu trường cặp giá trị (MaSinhVien,MaMonHoc) phải duy nhất, not null, sử dụng ràng buộc nào? Viết ràng buộc?**   DEFAULT  UNIQUE NOT NULL |
| 17 | **Yêu cầu Tỉnh thành chỉ được nhập Hà Nội hoặc TP. Hồ Chí Minh, sử dụng ràng buộc nào? Viết ràng buộc?**    CHECK    BETWEEN "Hà Nội" AND "TP. Hồ Chí Minh" |
| 18 | **Yêu cầu trường Họ tên không được để trống sử dụng loại ràng buộc nào? Viết ràng buộc?**    DEFAULT    NOT NULL |
| 19 | **Yêu cầu trường ClassID của bảng Student phải tham chiếu từ trường ClassID của bảng Class, sử dụng ràng buộc loại nào? Viết ràng buộc?**    FOREIGN KEY    FOREIGN KEY(ClassID)    REFERNCES Class(ClassID) |
| 20 | **Yêu cầu mã sinh viên bắt đầu là chữ ‘C’, độ dài 3 ký tự, sử dụng ràng buộc nào? Viết ràng buộc?**   DEFAULT   LIKE C% |
| 21 | **Yêu cầu ngày bắt đầu làm việc phải từ ngày hiện tại trở đi, sử dụng ràng buộc nào? Viết ràng buộc?**   CHECK   CHECK(NGAY\_BAT\_DAU > GETDATE) |
| 22 | **SQL là gì?**  SQL là viết tắt của Structured Query Language, là ngôn ngữ truy vấn mang tính cấu trúc. |
| 23 | **Trình bày về câu lệnh create?**   * Trong SQL có 2 lệnh CREATE có sẵn, bao gồm:   + CREATE DATABASE   + CREATE TABLE |
| 24 | **Trình bày về câu lệnh alter?**   * Lệnh ALTER TABLE trong SQL được sử dụng để thêm, xóa hoặc sửa đổi các cột trong một bảng hiện có. * Bạn cũng nên sử dụng lệnh ALTER TABLE để thêm và bỏ các ràng buộc khác nhau trên một bảng hiện có. * ALTER TABLE name\_Table MODIFY (MODIFYS) column\_name |
| 25 | **Trình bày về các kiểu dữ liệu ký tự/chuỗi trong MySQL? Phân biệt giữa các loại?**   * Một kiểu dữ liệu cung cấp một bộ các giá trị mà từ đó một biểu thức (ví dụ như biến, hàm…) có thể lấy giá trị của nó. * Kiểu định nghĩa các toán tử có thể được thực hiện trên dữ liệu của nó, ý nghĩa của dữ liệu, và cách mà giá trị của kiểu có thể được lưu trữ. * Kiểu dữ liệu String * Hầu hết dữ liệu của bạn trong MySQL sẽ được lưu trữ sẽ ở định dạng chuỗi. Danh sách dưới đây sẽ mô tả các kiểu dữ liệu chuỗi phổ biến trong MySQL * Có 3 loại   + Kiểu dữ liệu String   + Kiểu dữ liệu ngày và giờ   + Kiểu dữ liệu số |
| 26 | **Trình bày về các kiểu dữ liệu số trong MySQL? Phân biệt giữa các loại?**   * Có 2 kiểu dữ liệu số: * Kiểu dữ liệu số nguyên * Kiểu dữ liệu số thực   **Kiểu dữ liệu số nguyên**  BIT(size): Kiểu giá trị bit. Số lượng bit trên mỗi giá trị được chỉ định rõ về kích thước. Tham số kích thước có thể giữ giá trị từ 1 đến 64. Giá trị mặc định cho kích thước là 1.  TINYINT(size): Một số nguyên rất nhỏ. Phạm vi đã ký là từ -128 đến 127. Phạm vi chưa ký là từ 0 đến 255. Tham số kích thước chỉ định chiều rộng hiển thị tối đa (là 255).  BOOL: Số 0 được coi là sai, các giá trị khác 0 được coi là đúng.  BOOLEAN: Tương đương với BOOL.  SMALLINT(size): Một số nguyên nhỏ. Dải ô đã ký là từ -32768 đến 32767. Dải ô chưa ký là từ 0 đến 65535. Tham số kích thước chỉ định chiều rộng hiển thị tối đa (là 255).  MEDIUMINT(size): Một số nguyên trung bình. Dải ô đã ký là từ -8388608 đến 8388607. Dải ô chưa ký là từ 0 đến 16777215. Tham số kích thước chỉ định chiều rộng hiển thị tối đa (là 255).  INT(size): Một số nguyên trung bình. Dải ô đã ký là từ -2147483648 đến 2147483647. Dải ô chưa ký là từ 0 đến 4294967295. Tham số kích thước chỉ định chiều rộng hiển thị tối đa (là 255).  INTEGER(size): Bằng INT(size).  BIGINT(size): Một số nguyên lớn. Dải ô đã ký là từ -9223372036854775808 đến 9223372036854775807. Dải ô chưa ký là từ 0 đến 18446744073709551615. Tham số kích thước chỉ định chiều rộng hiển thị tối đa (là 255).  **Kiểu dữ liệu số thực**  FLOAT(size, d):   * Một số dấu phẩy động. Tổng số chữ số được chỉ định về kích thước. * Số chữ số sau dấu thập phân được chỉ định trong tham số d. Cú pháp này không được chấp nhận trong MySQL và nó sẽ bị xóa trong các phiên bản MySQL trong tương lai.   FLOAT(p):   * Một số dấu phẩy động. * MySQL sử dụng giá trị p để xác định xem nên sử dụng FLOAT hay DOUBLE cho kiểu dữ liệu kết quả. * Nếu p từ 0 đến 24, kiểu dữ liệu sẽ trở thành FLOAT (). Nếu p từ 25 đến 53, kiểu dữ liệu trở thành DOUBLE ().   DOUBLE(size, d):   * Một số dấu phẩy động có kích thước thông thường. * Tổng số chữ số được chỉ định về kích thước. S * Số chữ số sau dấu thập phân được chỉ định trong tham số d.   DOUBLE PRECISION(size, d)  DECIMAL(size, d):   * Một số điểm cố định chính xác. * Tổng số chữ số được chỉ định về kích thước. * Số chữ số sau dấu thập phân được chỉ định trong tham số d. Số tối đa cho kích thước là 65. Số lớn nhất cho d là 30. * Giá trị mặc định cho kích thước là 10. Giá trị mặc định cho d là 0.   DEC(size, d): Bằng với DECIMAL(size,d). |
| 27 | **Trình bày các kiểu dữ liệu ngày tháng trong MySQL? Phân biệt giữa các loại?**  **Kiểu dữ liệu DATE**   * Dữ liệu ngày tháng. * Định dạng: YYYY-MM-DD. * Phạm vi được hỗ trợ là từ ‘1000-01-01’ đến ‘9999-12-31’.   **Kiểu dữ liệu DATETIME(fsp)**   * Ngày và giờ kết hợp. * Định dạng: YYYY-MM-DD hh: mm: ss. * Phạm vi được hỗ trợ là từ ‘1000-01-01 00:00:00’ đến ‘9999-12-31 23:59:59’. * Thêm DEFAULT và ON UPDATE trong định nghĩa cột để tự động khởi tạo và cập nhật cho ngày và giờ hiện tại.   **Kiểu dữ liệu TIMESTAMP(fsp)**   * Dấu thời gian. * Giá trị TIMESTAMP được lưu trữ dưới dạng số giây kể từ kỷ nguyên Unix (‘1970-01-01 00:00:00’ UTC). * Định dạng: YYYY-MM-DD hh: mm: ss. * Phạm vi được hỗ trợ là từ ‘1970-01-01 00:00:01’ UTC đến ‘2038-01-09 03:14:07’ UTC. * Có thể chỉ định tự động khởi tạo và cập nhật cho ngày và giờ hiện tại bằng cách sử dụng.   DEFAULT CURRENT\_TIMESTAMP và ON UPDATE CURRENT\_TIMESTAMP trong định nghĩa cột.  **Kiểu dữ liệu TIME(fsp)**   * Dữ liệu thời gian. * Định dạng: hh: mm: ss. * Phạm vi được hỗ trợ là từ ‘-838: 59: 59’ đến ‘838: 59: 59’.   **Kiểu dữ liệu YEAR**   * Một năm ở định dạng bốn chữ số. * Các giá trị được phép ở định dạng bốn chữ số: 1901 đến 2155 và 0000. * Kiểu dữ liệu trong MySQL 8.0 không hỗ trợ năm ở định dạng hai chữ số. |
| 28 | **Yêu cầu bổ sung thêm khóa chính vào một bảng đã được tạo sẵn thì làm thế nào?**  Trả lời:  ALTER TABLE table\_name  ADD PRIMARY KEY (column\_name); |
| 29 | **Yêu cầu bổ sung thêm 1 cột vào bảng đã được tạo sẵn thì làm thế nào?**  Trả lời:  ALTER TABLE table\_name  ADD new\_column\_name data\_type; |
| 30 | **Yêu cầu xóa một ràng buộc đã được tồn tại sẵn trong bảng thì làm thế nào?**  Trả lời:  Đối với ràng buộc UNIQUE:  ALTER TABLE table\_name  DROP UNIQUE column\_name;  Đối với ràng buộc PRIMARY KEY:  ALTER TABLE table\_name  DROP PRIMARY KEY;  Đối với ràng buộc FOREIGN KEY:  ALTER TABLE table\_name  DROP FOREIGN KEY column\_name;  Đối với ràng buộc CHECK:  ALTER TABLE table\_name  DROP CHECK column\_name;  Đối với ràng buộc INDEX:  ALTER TABLE table\_name  DROP INDEX; |
| 31 | **Yêu cầu sửa kiểu dữ liệu của 1 cột có sẵn trong bảng thì làm thế nào?**  Trả lời:  ALTER TABLE table\_name  MODIFY COLUMN column\_name new\_data\_type; |
| 32 | **Trình bày về câu lệnh insert? insert nhiều bộ giá trị?**  Trả lời:  Câu lệnh INSERT trong MySQL được sử dụng để chèn dữ liệu vào bảng.  Cú pháp:  INSERT INTO table\_name(column\_name\_1, column\_name\_2, ... column\_name\_n)         VALUES (value\_1, value\_2, ... value\_n);  Có thể viết đơn giản cú pháp hơn nếu chèn tất cả giá trị trong bảng:  INSERT INTO table\_name VALUES (value\_1, value\_2, ... value\_n);  Chèn nhiều bộ giá trị:  INSERT INTO table\_name(column\_name\_1, column\_name\_2, ... column\_name\_n) VALUES  (value\_1a, value\_2a, ... value\_na),  (value\_1b, value\_2b, ... value\_nb),  …  (value\_1n, value\_2n, ... value\_nn); |
| 33 | **Trình bày về câu lệnh update? Phân biệt alter và update?**  Trả lời: |
| 34 | **Trình bày về câu lệnh delete? Phân biệt drop và delete?**  Trả lời:  Câu lệnh DELETE trong MySQL được sử dụng để xóa các bản ghi hiện có từ một bảng.  Bạn có thể sử dụng mệnh đề WHERE với truy vấn DELETE để xóa các hàng đã chọn, nếu không tất cả các bản ghi sẽ bị xóa.  DELETE : Xóa một hay tất cả dòng trong một bảng theo một điều kiện nhất định, dữ liệu có thể phục hồi lại  DROP : Xóa một bảng khỏi database |
| 35 | **Bạn có biết “safe update mode”? Nếu biết, hãy trình bày về cơ chế này?**  Trả lời:  Safe update mode là 1 cơ chế trong MySQL Workbench để cảnh báo cho bạn thực hiện xóa hay update dữ liệu với 1 column không phải là KEY để tránh rủi ro mất mát dữ liệu quá lớn.    Safe update mode sinh ra nhằm hạn chế rủi ro bạn update “nhầm” các bản ghi với các field không phải là khóa. Về cơ bản thì ý tưởng của chế độ này là rất tốt; giúp hạn chế thiệt hại khi lỡ tay update. |
| 36 | **Trình bày về câu lệnh select?**  Trả lời:  Câu lệnh SELECT trong MySQL được sử dụng để lấy dữ liệu từ một bảng cơ sở dữ liệu trả về dữ liệu này dưới dạng một bảng kết quả. Các bảng kết quả này được gọi là tập kết quả (result-sets).  Cú pháp:  SELECT column1, column2, columnN FROM table\_name;  Để lấy tất cả các trường có sẵn trong bảng:  SELECT \* FROM table\_name; |
| 37 | **Trình bày về select lồng? Lấy ví dụ?**  Trả lời:  Select lồng là một truy vấn lồng bên trong 1 câu lệnh như insert, select hoặc delete.  Ví dụ: |
| 38 | **Trình bày về phép nối (join)?**  Trả lời:  Hoạt động Join trong MySQL có thể được hiểu như là việc ghim hai hoặc nhiều bảng vào trong một bảng đơn. Bạn có thể sử dụng JOIN trong các lệnh SELECT, UPDATE và DELETE để kết hợp các bảng MySQL. |
| 39 | **Làm sao để có thể tăng tốc độ truy vấn?**  Trả lời:  Sử dụng index để tăng tốc độ truy vấn.  Chỉ mục (Index) là bảng tra cứu đặc biệt mà Database Search Engine có thể sử dụng để tăng nhanh thời gian và hiệu suất thu thập dữ liệu. Hiểu đơn giản, một chỉ mục là một con trỏ tới dữ liệu trong một bảng. Một chỉ mục trong một Database là tương tự như một chỉ mục trong Mục lục của cuốn sách.  Ví dụ, nếu muốn tìm chiếu tất cả các trang trong một cuốn sách về một chủ đề nào đó, đầu tiên ta nghĩ ngay đến mục lục của nó, mà liệt kê tất cả các chương, chủ đề theo thứ tự và sau đó được tham chiếu tới một hoặc nhiều trang cụ thể. Khi đã có mục lục của cuốn sách, việc tìm kiếm trang sách đó sẽ nhanh chóng và đỡ tốn công hơn rất nhiều so với việc ta phải lật từng trang. |
| 40 | **Trình bày việc sao chép dữ liệu từ bảng này sang bảng khác trong MySQL?**  Trả lời:   * Cú pháp 1:   Cú pháp này được sử dụng trong trường hợp chúng ta muốn sao chép dữ liệu từ tất cả các cột của bảng table1 vào bên trong bảng table2 (Lưu ý: Khi sử dụng cú pháp này thì số cột của bảng table2 phải bằng với số lượng cột của bảng table1, nếu không thì khi thực thi lệnh sẽ bị lỗi)  INSERT INTO table2  SELECT \*  FROM table1  WHERE condition;     * Cú pháp 2:   Cú pháp này thường được sử dụng trong những trường hợp chúng ta muốn sao chép dữ liệu từ một số cột nào đó của bảng table1 vào một số cột nào đó của bảng table2.  INSERT INTO table2 (column1, column2, column3, . . . .)  SELECT column1, column2, column3, . . . .  FROM table1  WHERE condition; |
| 41 | **Phân biệt giữa count(\*) và count(cột)?**  Trả lời:  Hàm COUNT() được dùng để đếm số lượng mẫu tin (dữ liệu, hàng) trong bảng.   * count(\*) dùng để đếm tất cả mẫu tin trong bảng * count(cột) dùng để đếm những mẫu tin có điều kiện dựa theo cột |
| 42 | **Làm sao để truy vấn dữ liệu từ nhiều bảng?**  Trả lời:  Sử dụng select và inner join hoặc left join.  Chi tiết ví dụ: https://codegym.vn/blog/2020/05/04/thuc-hanh-truy-van-du-lieu-tu-nhieu-bang/ |
| 43 | Phân biệt giữa Where và Having?  Trả lời:  Thứ tự: where - group by - having |
| 44 | Trình bày mệnh đề Order By trong câu lệnh Select?  Trả lời:  Mệnh đề ORDER BY trong MySQL được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần hoặc giảm dần, dựa trên một hoặc nhiều cột. Một số cơ sở dữ liệu sắp xếp các kết quả truy vấn theo thứ tự tăng dần theo mặc định. Lệnh ASC được sử dụng để sắp xếp tăng dần và DESC được sử dụng để sắp xếp giảm dần.  Cú pháp  Cú pháp cơ bản của mệnh đề ORDER BY trong MySQL như sau:  SELECT column-list  FROM table\_name  [WHERE condition]  [ORDER BY column1, column2, .. columnN] [ASC | DESC]; |
| 45 | Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?  Trả lời:  Sử dụng avg()  select (avg(pointA) + avg(pointB) + avg(pointC)) / 3 as avg\_of\_avg\_points  from student; |
| 46 | Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?  Trả lời:  select id, name, (pointA + pointB + pointC) / 3 as avg\_points  from student; |
| 47 | 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?  Trả lời:  select id, name, (pointA + pointB + pointC) / 3 as avg\_points  from student  group by (id)  having avg\_points > 5; |
| 48 | Hiển thị danh sách sinh viên và lớp học tương ứng thì làm thế nào (hai bảng Student và Class có cột chung ClassID)?  Trả lời:  select \*  inner join class  from student on student.ClassID = class.ClassID  group by (student.StudentID); |
| 49 | Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi?  Trả lời:  select \* from student  group by (point) ASC; |
| 50 | 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) desc; |
| 51 | Hiển thị các sinh viên có họ là ‘Le’ và tên là ‘Hai’?   * select \* from student   where surname = ‘Le’ AND name = ‘Hai’; |
| 52 | Hiển thị các sinh viên có tên bắt đầu là ‘C’?   * select \* from student   where name LIKE 'C%'; |
| 53 | Union là gì? Phân biệt Union và Union All?   * Union loại bỏ các bản ghi trùng lặp (trong đó tất cả các cột trong kết quả là như nhau).   Cả UNION và UNION ALL đều nối kết quả của hai SQL khác nhau. Họ khác nhau trong cách xử lý các bản sao.   * UNION thực hiện DISTINCT trên tập kết quả, loại bỏ mọi hàng trùng lặp. * UNION ALL không loại bỏ các bản sao và do đó nhanh hơn UNION. |
| 54 | Trình bày về khái niệm chỉ mục (index)? Tại sao lại cần đánh chỉ mục? Nên đánh chỉ mục trong trường hợp nào?  **-Chỉ mục (Index) là bảng tra cứu đặc biệt mà Database Search Engine có thể sử dụng để tăng nhanh thời gian và hiệu suất thu thập dữ liệu. Hiểu đơn giản, một chỉ mục là một con trỏ tới dữ liệu trong một bảng**  **- Khó khăn: Khi muốn tìm kiếm dữ liệu trên bảng cần phải quét toàn bộ bảng. Điều này làm chậm tốc độ thực thi truy vấn => Việc tìm kiếm có thể được trợ giúp nhiều nếu dữ liệu được chứa trong các khối được đánh chỉ mục (index).**  **-Nên tạo chỉ mục trên các cột khi:**  **▪ Bảng chứa dữ liệu lớn**  **▪ Cột được sử dụng để tìm kiếm thường xuyên**  **▪ Cột được sử dụng để sắp xếp dữ liệu**  **▪ Dữ liệu trong cột có sự phân biệt cao** |
| 55 | Có những loại chỉ mục nào? Trình bày chỉ mục duy nhất? Nếu đánh chỉ mục duy nhất lên một cột trong đó dữ liệu của cột không đảm bảo tính duy nhất thì có được không?  - Chỉ mục phân cụm ( Clustered Indexes) và Chỉ mục không phân cụm ( Nonclustered Indexes).  -**Unique Index** là chỉ mục duy nhất, được sử dụng để tăng hiệu suất và đảm bảo tính toàn vẹn dữ liệu. Một chỉ mục duy nhất không cho phép chèn bất kỳ giá trị trùng lặp nào được chèn vào bảng.  - KHÔNG ĐƯỢC |
| 56 | Trình bày về khung nhìn (view)? Phân biệt view và table?  -Một view là một bảng ảo được tạo ra từ việc tập hợp các cột của một hoặc nhiều bảng  -View là bản ảo còn table là bản thật. Khi xóa dữ liệu trên View thì dữ liệu trên database vẫn sẽ không thay đổi. |
| 57 | Cú pháp tạo view?  Cú pháp:   CREATE VIEW view\_name AS SELECT column1, column2.....  FROM table\_name  WHERE [condition];  • Ví dụ: CREATE VIEW CUSTOMERS\_VIEW AS SELECT name, age  FROM CUSTOMERS |
| 58 | Có thể tạo view từ view khác không?   * Có thể. |
| 59 | Trình bày về thủ tục lưu trữ (stored procedure)?  - Store Procedure (thủ tục lưu trữ), có thể được định nghĩa là chương trình con, giống như một chương trình con được lưu trữ trong cơ sở dữ liệu. |
| 60 | Yêu cầu viết thủ tục lưu hiển thị họ tên sinh viên thông qua mã sinh viên là input? Chạy thủ tục với mã sinh viên là 2?   * DELIMITER *$$* * CREATE PROCEDURE getStudent * (IN ID INT(11)) * BEGIN * SELECT name FROM Student WHERE studentID = ID; * END *$$* * DELIMITER ; * Call getStudent(2); |
| 61 | Yêu cầu viết thủ tục lưu đếm số lượng sinh viên có điểm thi lớn hơn một giá trị input? Chạy thủ tục với điểm là 5?  -DELIMITER *$$*   * CREATE PROCEDURE getStudentNumber * (IN InNumber INT(11)) * BEGIN * SELECT Count(studentID) FROM Student WHERE point > InNumber; * END *$$* * DELIMITER ; * Call getStudentNumber(5); |
| 62 | Viết thủ tục dựa vào số a là input, kết quả:  +Nếu a lớn hơn 0 hiển thị “a là số dương”  +Nếu a nhỏ hơn 0 hiển thị “a là số âm”  +Nếu a  bằng 0 hiển thị “a bằng 0” |
| 63 | Css là gì? Các cách nhúng CSS vào tài liệu html?   * CSS là viết tắt của cụm từ Cascading Style Sheets * CSS mô tả cách các phần tử HTML sẽ hiển thị trên màn hình, trang web, hoặc các phương tiện khác. * CSS giúp chúng ta tiết kiệm được rất nhiều công công sức. Nó giúp chúng ta chỉnh sửa giao diện của nhiều trang web cùng một lúc. * Các mô tả css có thể được lưu trong tệp tin CSS. |
| 64 | Trình bày về các loại bộ chọn (selector) trong CSS?  -Bao gồm:   * Bộ chọn phần tử * Bộ chọn Class * Bộ chọn id |
| 65 | Phân biệt giữa bộ chọn class và bộ chọn id?  -Bộ chọn id sử dụng thuộc tính id của một phần tử HTML để chỉ định cụ thể một phần tử HTML nào đó. Một id nên là duy nhất trong một trang, do đó bộ chọn id được sử dụng khi muốn áp dụng kiểu cho một phần tử duy nhất nào đó trên trang. Để chọn một phần tử với một id cụ thể, chỉ cần viết kí hiệu # trước id (giá trị của thuộc tính id) của phần tử đó.  -Bộ chọn class (class selector) chọn các phần tử có thuộc tính class với một giá trị cụ thể.  Để chọn các phần tử với một class cụ thể, chỉ cần viết thêm dấu chấm (.) trước tên của class |
| 66 | Trình bày mô hình hộp (box model) trong CSS?  -Một phần tử HTML được coi là một Box.  Mô hình hộp (Box model) có nghĩa là một phần tử HTML sẽ bao gồm các thành phần: margin, border, padding, và content (nội dung của phần tử).  Mô hình hộp (Box model) cho phép chúng ta thêm bo viền xung quanh phần tử , và xác định khoảng cách giữa các phần tử. |
| 67 | Thuộc tính Margin dùng để làm gì?  -Lề được sử dụng để tạo không gian xung quanh các phần tử, bên ngoài bất kỳ đường viền đã xác định nào. |
| 68 | Phân biệt margin, padding và border?  -margin: Lề được sử dụng để tạo không gian xung quanh các phần tử, bên ngoài bất kỳ đường viền đã xác định nào.  -padding: được sử dụng để tạo không gian xung quanh nội dung của phần tử, bên trong bất kỳ đường viền đã xác định nào.  -border: Thuộc tính đường viền CSS cho phép bạn chỉ định kiểu, chiều rộng và màu của đường viền của phần tử. |
| 69 | Thuộc tính position dùng để làm gì? Phân biệt các giá trị: static, absolute, relative và fixed?  • Thuộc tính position của CSS quy định cách thức xếp đặt vị trí của một thành phần trong trang web, hay nói cách khác position chỉ định phương thức định vị sẽ áp dụng cho một phần tử HTML .  -static: +Khi một phần tử là static điều đó có nghĩa là phần tử này sẽ được sắp xếp đúng theo vị trí tự nhiên của nó .  + Các phần tử được định vị static không bị ảnh hưởng bởi các thuộc tính top, bottom, left, right .  + Một phần tử định vị static thì luôn luôn được định vị theo luồng (flow) bình thường của trang, tức các thành phần sẽ nằm theo thứ tự trong văn bản.   -absolute: +Định vị một thành phần so với vị trí của thành phần cha (tổ tiên) gần nhất của nó (thành phần cha phải có thuộc tính position khác static).  + Nếu một 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 .  -relative: + Một phần tử với Position: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.   + Thiết lập cho các thuộc tính định vị top, right, bottom, left cho một phần tử là relative là nguyên nhân khiến nó bị điều chỉnh ra khỏi vị trí bình thường của nó .  -fixed: + Định vị một thành phần so với trình duyệt   + Một phần tử với position:fixed được định vị tương đối so với chế độ xem (viewport), có nghĩa là nó luôn luôn ở cùng một vị trí ngay cả khi trang được cuộn  + Các thuộc tính top, right, bottom, left được sử dụng để định vị phần tử   + Một phần tử fixed không để lại một khoảng trống trong trang nơi nó thường được định vị |
| 70 | Trình bày thuộc tính float, clear?  -Thuộc tính float trong CSS có tác dụng xác định xem thành phần được chọn được hiển thị (nổi) như thể nào.  -Thuộc tính clear trong CSS có tác dụng loại bỏ các thuộc tính float (left - right) trên một thành phần. Để giải quyết vấn đề trong mục chú ý của thuộc tính float phía trên. |
| 71 | **Tạo bố cục trang trong html5 với các thẻ cơ bản?**   * Mọi người kể tên các thẻ cơ bản trong HTML nha. |
| 72 | **Bạn có biết Prototype? Trình bày?**   * Prototype là cơ chế mà các object trong javascript kế thừa các tính năng từ một object khác. Tất cả các object trong javascript đều có một prototype, và các object này kế thừa các thuộc tính (properties) cũng như phương thức (methods) từ prototype của mình. |
| 73 | **Responsive Web Design là gì?**   * Responsive Web Design (RWD) là xu hướng mới theo đó quy trình thiết kế và phát triển web sẽ đáp ứng mọi thiết bị và môi trường của người dùng theo các tiêu chí kích thước và chiều của màn hình thiết bị. |
| 74 | **Trình bày cú pháp RWD Media Queries?**   * Media Queries (Tạm dịch là Truy vấn phương tiện) là một kỹ thuật CSS, được giới thiệu trong CSS3. Cho phép kết xuất nội dung để thích ứng với các điều kiện như độ phân giải màn hình. * Với **Media Queries** ta có thể định dạng cách hiển thị trang web sao cho nó hiển thị Responsive trên nhiều kích cỡ trình duyệt khác nhau, ví dụ trên máy tính, ipad, điện thoại,... |
| 75 | **Phân biệt Static web và Dynamic web?**   * Web tĩnh đơn thuần chỉ sử dụng hoàn toàn ngôn ngữ lập trình HTML. Không lấy dữ liệu thông qua database, gần như là fix cứng, khó thay đổi, update. Website động hỗ trợ thêm các ngôn ngữ lập trình đa dạng như [ASP.NET](https://dotnet.microsoft.com/apps/aspnet), [PHP](https://php.net/)…Và cơ sở dữ liệu SQL server, [MySQL](https://www.mysql.com/)… * Đối với website tĩnh, người dùng hầu như không thể tương tác được với website. Trong 1 web động, khách hàng có được quyền trao đổi thông tin với chủ website một cách dễ dàng nhất. |
| 76 | **Giao thức (protocol) là gì? Kể tên một số giao thức bạn biết?**   * Giao thức là tập hợp các quy tắc được thiết lập nhằm xác định cách để định dạng, truyền và nhận dữ liệu sao cho các thiết bị mạng máy tính đều có thể giao tiếp với nhau, bất kể sự khác biệt về cơ sở hạ tầng, thiết kế hay các tiêu chuẩn cơ bản giữa chúng. * Một số giao thức: HTTP, HTTPS, TCP/IP, DNS, SSL/TSL, FTP. |
| 77 | **HTTP là gì? Phân biệt HTTP và HTTPs?**   * Là giao thức truyền tải siêu văn bản. * HTTP là một trong năm giao thức chuẩn về mạng Internet. * Là giao thức dùng cho WWW. * Được dùng để liên hệ thông tin giữa máy cung cấp dịch vụ (Web server) và máy sử dụng dịch vụ (web client). * **Sự khác biệt giữa HTTP và HTTPS:** * HTTPS là viết tắt của từ HyperText Transfer Protocol Secure và chính là giao thức HTTP có sử dụng thêm các chứng chỉ SSL (secure Sockets Layer). * HTTPS giúp mã hóa dữ liệu truyền tải nhằm gia bảo mật giữa Web sever đến các trình duyệt web. * Hiểu theo một cách khác, HTTPS là một phiên bản của HTTP, nhưng bảo mật, an toàn hơn. |
| 78 | **TCP/IP là gì?**   * TCP/ IP (Transmission Control Protocol/ Internet Protocol - Giao thức điều khiển truyền nhận/ Giao thức liên mạng), là một bộ giao thức trao đổi thông tin được sử dụng để truyền tải và kết nối các thiết bị trong mạng Internet. * Một mô hình TCP/IP tiêu chuẩn bao gồm 4 lớp được chồng lên nhau, bắt đầu từ tầng thấp nhất là Tầng vật lý (Physical) → Tầng mạng (Network) → Tầng giao vận (Transport) và cuối cùng là Tầng ứng dụng (Application). |
| 79 | **Trình bày mô hình Client/Server?**   * Là mô hình mạng máy tính gồm có 2 thành phần chính đó là máy khách (client) và máy chủ (server). * Server chính là nơi giúp lưu trữ tài nguyên cũng như cài đặt các chương trình dịch vụ theo đúng như yêu cầu của client. Ngược lại, Client bao gồm máy tính cũng như các loại thiết bị điện tử nói chung sẽ tiến hành gửi yêu cầu đến server. |
| 80 | **Máy chủ (Server) là gì?**   * Server chính là nơi giúp lưu trữ tài nguyên cũng như cài đặt các chương trình dịch vụ theo đúng như yêu cầu của client. |
| 81 | **Phân biệt máy chủ web và máy chủ ứng dụng?**   |  |  | | --- | --- | | **Máy chủ web (Web Server)** | **Máy chủ ứng dụng (App Server)** | | Web server hữu ích hoặc phù hợp cho nội dung tĩnh. | Trong khi application server được trang bị cho nội dung động. | | Công suất của web server thấp hơn app server. | Trong khi công suất của application server cao hơn web server. | | Web server tiêu thụ hoặc sử dụng ít tài nguyên hơn. | Trong khi application server sử dụng nhiều tài nguyên hơn. | | Trong web server, đa luồng không được hỗ trợ. | Trong application server, đa luồng được hỗ trợ. | |
| 82 | **Tomcat là gì? Tomcat thuộc Web Server hay Application Server?**   * Apache Tomcat là một web server HTTP được phát triển bởi Apache Software Foundation, hỗ trợ mạnh cho ứng dụng Java thay vì website tĩnh. |
| 83 | **Phân biệt HTTP Request và HTTP Response?**   * Một HTTP client (máy khách ) gửi một HTTP request (yêu cầu) lên server (máy chủ). Phía server nhận request của client, sau đó gửi HTTP response (phản hồi) cho client. |
| 84 | **Trình bày Content Type hay MIME (Multipurpose Internet Mail Extension) Type?**   * Multipurpose Internet Mail Extensions (MIME) là một tiêu chuẩn Internet giúp mở rộng khả năng giới hạn của email bằng cách cho phép chèn hình ảnh, âm thanh và văn bản trong một tin nhắn. |
| 85 | **Phân biệt rõ hai phương thức Get và Post?**   |  |  |  | | --- | --- | --- | | **Tiêu chí** | **Phương thức GET** | **Phương thức POST** | | **Mục đích** | Truy xuất dữ liệu, lấy dữ liệu | Cập nhật dữ liệu, đẩy dữ liệu về phía Server | | **Kết quả truy vấn** | Có thể đánh dấu (bookmark), vì dữ liệu hiển thị trên URL. | Không thể đánh dấu, do dữ liệu không hiển thị trên URL | | **Bảo mật** | Có thể nhìn thấy bởi bất kỳ ai. Dễ bị tấn công, do dữ liệu hiển thị rõ ràng trên URL | An toàn hơn GET | | **Độ dài biểu mẫu** | Bị giới hạn | Không giới hạn | |
| 86 | **Trình bày lớp HTTPServlet?**   * HttpServlet là HttpServlet thì nó là một abstract class, là một trong những đối tượng của Jakarta EE Servlet giúp chúng ta có thể xây dựng các ứng dụng chạy trên nền web với Java. * Đối tượng HTTPSevlet dùng trong các ứng dụng Java web để xử lý các request từ client với các HTTP method như GET, POST, HEAD, PUT, DELETE, TRACE và OPTIONS. * Khi client gửi một request tới web server container, method service() trong đối tượng HttpServlet sẽ process request này. Phụ thuộc vào HTTP method mà client gửi, việc xử lý tiếp theo sẽ do các method doGet(), doPost(), doHead(), doPut(), doDelete(), doTrace() và doOptions() của đối tượng này đảm nhận. |
| 87 | **Trình bày Serv**Ý tưởng cơ bản của servlet container là sử dụng Java để tự động tạo trang web ở phía máy chủ, xử lý các yêu cầu tính toán từ phía client và trả lại các kết quả theo request của client đó.   * Vậy nên, servlet container thực chất là một phần của một web server tương tác với các servlet. Và đúng như tên gọi servlet container giống như 1 cái thùng chứa, chứa tất cả các servlet ở bên trong. |
| 88 | **Vòng đời của Servlet?**   * Phương thức init() * Phương thức service() * Phương thức doGet() * Phương thức doPost() * Phương thức destroy() |
| 89 | **Trình bày Servlet Interface? Mô tả các phương thức init(), service(), destroy()?**   * **Phương thức init()** được gọi chỉ một lần để khởi tạo servlet. * **Phương thức service()** là phương thức chính để thực hiện nhiệm vụ thực tế. Bộ chứa servlet (tức là web server) gọi phương thức service() để xử lý các yêu cầu đến từ khách hàng (trình duyệt) và trả về kết quả. * **Phương thức destroy()** chỉ được gọi một lần vào cuối chu kỳ sống của một servlet. Phương thức này cho phép servlet đóng kết nối cơ sở dữ liệu, chặn các luồng chạy gầm, viết các cookie hoặc đếm số lượt truy cập và thực hiện các hoạt động dọn dẹp khac. |
| 90 | **Phân biệt các loại Servlet Container: Standalone, In-process và Out-of-Servlet?** |
| 91 | **Giới thiệu một số phương thức của lớp HttpServlet?**   * Sử dụng đối tượng HttpServlet này, chúng ta có thể xây dựng các ứng dụng Java web để handle các request từ client với các HTTP method như GET, POST, HEAD, PUT, DELETE, TRACE và OPTIONS. |
| 92 | **Chỉ ra nhiệm vụ của hai phương thức:**  **public void service(ServletRequest req, ServletResponse res)**  **protected void service(HttpServletRequest req, HttpServletResponse res)** |
| 93 | **Phân biệt doGet() và doPost()?**   |  |  |  | | --- | --- | --- | | **Tiêu chí** | **doGet()** | **doPost()** | | Lưu trữ (Cache) | Dữ liệu gửi bằng phương thức GET sẽ được lưu trữ lại trong query string và có thể được xem trong lịch sử trình duyệt. | Dữ liệu và địa chỉ URL của các request gửi bằng POST không được trình duyệt lưu lại. | | *Đánh Dấu (Bookmark)* | Đối với request gửi bằng phương thức GET người dùng có thể bookmark lại được trên trình duyệt. | Ngược lại các request gửi bằng POST sẽ không thể bookmark được. | | *Dữ Liệu* | Với phương thức GET dữ liệu gửi đi bị giới hạn sử dụng các ký tự chữ có trong [bộ ký tự ASCII](https://www.codehub.com.vn/Phan-Biet-Character-Set-va-Character-Encoding) và bị giới hạn bởi số lượng ký tự tối đa cho phép | Phương thức POST không giới hạn dung lượng dữ liệu gửi đi cũng như loại nhữ liệu (văn bản thông thường hay file nhị phân như upload tập tin hay hình ảnh, video...) | |
| 94 | **RequestDispatcher được dùng làm gì?**   * RequestDispatcher là một interface, việc **thực hiện xác định một đối tượng có thể gửi yêu cầu tới bất kỳ tài nguyên nào (như HTML, Image, JSP, Servlet) trên server**. |
| 95 | **Phân biệt forward() của RequestDispatcher và SendRedirect()**   |  |  | | --- | --- | | **RequestDispatcher (chuyển tiếp)** | **SendRedirect (chuyển hướng)** | | Khi một yêu cầu (request) của trình duyệt gửi tới một Servlet, nó có thể chuyển tiếp yêu cầu tới một trang khác (hoặc một servlet khác). | Khi một yêu cầu (request) từ phía người dùng tới một Servlet (Trang A), servlet này có thể chuyển yêu cầu này tới một trang khác (Trang B), và kết thúc nhiệm vụ của nó.  Trang được chuyển hướng tới có thể là trang trong ứng dụng của bạn, hoặc có thể là một trang bất kỳ. | | Địa chỉ trên trình duyệt của người dùng vẫn là đường dẫn của trang đầu tiên, nhưng nội dung của trang do trang được chuyển tiếp tới tạo ra. | Địa chỉ trên trình duyệt của người dùng lúc này sẽ hiển thị đường dẫn của trang B. | |
| 96 | **Jsp là gì? Nói: “Về bản chất Jsp cũng chính là Servlet” đúng hay sai? Giải thích?**   * **JSP là viết tắt của JavaServer Pages** là một công nghệ để phát triển các trang web động. JSP giúp các nhà phát triển chèn java code vào các trang HTML bằng cách sử dụng các thẻ JSP đặc biệt. * **JSP** là một kiểu Java servlet được thiết kế để tạo ra giao diện người dùng cho một ứng dụng Java web. Các nhà phát triển web viết các JSP như các tệp văn bản kết hợp mã HTML hoặc XHTML, các phần tử XML, các action và lệnh JSP. |
| 97 | **So sánh giống và khác nhau giữa Jsp và Servlet?**   * **JSP** là mã dựa trên HTML, còn **Servlet** là mã Java. * Mã của **JSP** là Java trong HTML nên viết khá dễ, còn **Servlet** là HTML trong Java nên viết mã khó hơn. * **JSP** tiếp cận MVC **và** hiển thị đầu ra, còn **Servlet** điều khiển cách tiếp cận MVC. * Bước đầu tiên của **JSP** là dịch mã Java rồi biên dịch nên **JSP** làm việc chậm hơn **so** với **Servlet**. |
| 98 | **Vòng đời của Jsp?**   * Biên dịch * Khởi tạo * Thực thi * Hủy |
| 99 | **Jstl là gì? Giới thiệu một số thẻ bạn đã sử dụng jstl?**   * JavaServer Pages Standard Tag Library (JSTL) là một tập hợp các thẻ JSP hữu ích, mà gói các tính năng lõi phổ biến tới các ứng dụng JSP. * JSTL hỗ trợ tới các tác vụ phổ biến và có tính cấu trúc, ví dụ như các tính lặp và điều kiện, các thẻ để thao tác tài liệu XML, các thẻ đa ngôn ngữ, và các thẻ SQL. Nó cũng cung cấp Framework để tích hợp các Custom Tags với các thẻ JSTL. * **Một số thẻ JSTL:** <c:out> <c:set> <c:remove> <c:catch> <c:if> <c:choose> <c:forEach> |
| 100 | **Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC?**   * MVC là viết tắt của Model - View - Controller. Là một mô hình thiết kế phần mềm thường được sử dụng để thiết kế phần mềm có giao diện người dùng. * MVC tuy là một mô hình thiết kế phần mềm nhưng nó cũng được sử dụng rộng rãi trong web, sự khác biệt được tùy chỉnh liên quan đến sự có mặt của server - client. * **Vai trò của Model:** Là một dạng mẫu dữ liệu, có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu. Controller sẽ thông qua moel để lấy các hàm, phương thức, sau đó gửi qua View * **Vai trò của View:** Là các giao diện người dùng, có nhiệm vụ tiếp nhận dữ liệu từ controller. nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống. * **Vai trò của Controller:** Là các hành vi, hành động, xử lý của hệ thống đóng vai trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client. |
| 101 | **Phân biệt kiến trúc ba tầng (đa tầng hay n- tầng) với kiến trúc MVC**?     * **Điểm giống nhau:**    + Cả hai đều là mô hình kiến trúc áp dụng trong lập trình được tách thành 3 thành phần.   + Tách biệt sự phụ thuộc giữa ngôn ngữ lập trình, môi trường phát triển, xử lý logic.   + Các thành phần ở mỗi mô hình có nhiều điểm tương đồng như Presentation Layer và View * **Điểm khác biệt:**   + Khác biệt rõ nhất là cách xử lý yêu cầu (workflow) của 2 mô hình (Tham khảo cách làm việc của 2 mô hình).   + Ở MVC, thành phần Model giữ chức năng của cả lớp Business và Data Access trong mô hình 3 Layer.   + 3 Layer thường được sử dụng trong lập trình ứng dụng, trong khi đó MVC được ưu chuộng nhiều trong lập trình website vì tính linh hoạt và không bắt buộc các DTO như 3-Layer. |
| 102 | **Giới thiệu một ứng dụng tổ chức kiến trúc MVC?**  Mô hình MVC được ứng dụng trong nhiều [ngôn ngữ lập trình](https://monamedia.co/top-10-ngon-ngu-lap-trinh-phan-mem-tot-nhat-hien-nay/) khác nhau, nhưng phổ biến nhất là ứng dụng ASP.NET MVC hay PHP MVC. |
| 103 | **Trình bày ưu và nhược điểm của kiến trúc MVC?**  **Ưu điểm**  Do được chia thành các thành phần độc lập nên Mô hình MVC giúp phát triển ứng dụng có code dễ đọc, dễ nâng cấp, bảo trì. Thể hiện tính chuyên nghiệp trong việc tạo ứng dụng.  **Nhược điểm**  Đối với dự án nhỏ việc áp dụng mô hình MVC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần. |
| 104 | Sự giống, khác nhau và ưu nhược điểm của MVC1 & MVC2?  Khong ro |
| 105 | API là gì?  **API** là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của **Application Programming Interface** – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng. |
| 106 | JDBC là gì? Phân biệt JDBC và ODBC?  **JDBC**, là viết tắt của Java Database Connectivity, là một Java API chuẩn để kết nối giữa ngôn ngữ lập trình Java và các cơ sở dữ liệu đa dạng. Sử dụng JDBC, bạn có thể thực hiện nhiều tác vụ đa dạng khi làm việc với cơ sở dữ liệu như tạo, xóa cơ sở dữ liệu; tạo và thực thi các lệnh SQL hoặc MySQL; tạo, xóa các bản ghi  khác biệt  **ODBC** là kết nối thủ tục (ODBC is procedural)  **JDBC** là kết nối hướng đối tượng ( JDBC is object oriened) |
| 107 | Các thành phần của JDBC API? Trình bày: DriverManager, Driver, Connection, Statement, ResultSet, SQLException?  **DriverManager**: dùng để quản lý danh sách các Driver(database drivers)  **Driver**: dùng để liên kết tới cơ sở dữ liệu,điều khiển các liên kết  **Connection**: biểu thị kết nối tới cơ sở dữ liệu.Dùng để tạo ra Statement,PreparedStatement và CallableStatement.  **Statement interface trong java** cung cấp các phương thức để thực thi các câu lệnh truy vấn với cơ sở dữ liệu SQL. Statement interface là một nhà máy của ResultSet, tức là nó cung cấp phương thức để tạo ra đối tượng ResultSet  **ResultSet**-biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là SELECT  **SQLException**-một lớp xử lý lỗi ngoại lệ chứa các lỗi truy cập CSDL |
| 108 | Trình bày Statement? Phân biệt Statement, PreparedStatement và CallableStatement?   * **Statement**  –  Sử dụng để thực hiện các câu truy vấn SQL tĩnh * **PreparedStatement** – Sử dụng để thực hiện các  câu truy vấn SQL động hoặc có tham số * **CallableStatement** – Sử dụng để thực thi các stored procedures (Hiểu nôm na là các lệnh định nghĩa sẵn trên database) |
| 109 | Hướng dẫn sử dụng PreparedStatement? |
| 110 | Hướng dẫn sử dụng CallableStatement? |
| 111 | stmt =  conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY); Trình bày ý nghĩa các lựa chọn ResultSet ở trên? |
| 112 | Sử dụng PreparedStatement trong tính năng edit sản phẩm? |
| 113 | Sử dụng CallableStatement trong tính năng xóa sản phẩm có id cho trước? |
| 114 | Gọi MySQL Stored Procedures từ JDBC như thế nào?  Để gọi đến một Store Procedure MySQL thì ta sử dụng đối tượng CallableStatement, đối tượng này được kế thừa từ đối tượng PreparedStatement. Nói chung cú pháp để gọi đến một thủ tục như sau:   call procedure\_name(param1,param2,...) |
| 115 | Transaction là gì? Ví dụ? Mô tả 4 thuộc tính ACID của Transaction?  Transaction (giao tác) là một tiến trình xử lý, có điểm bắt đầu và điểm kết thúc, gồm nhiều phép thực thi nhỏ, trong đó mỗi phép thực thi được thực thi một cách tuần tự và độc lập theo nguyên tắc là tất cả thành công hoặc một phép thực thi thất bại thì cả tiến trình thất bại.  Các thuộc tính ACID miêu tả rõ ràng nhất về Transaction. 4 thuộc tính này bao gồm Atomicity, Consistency, Isolation và Durability, trong đó:   * 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, … |
| 116 | Những ưu điểm khi sử dụng Transaction? Cú pháp để tạo Transaction? |
| 117 | Trình bày về commit()?    phương thức commit() sử dụng khi bạn đã thực hiện các thay đổi với cơ sở dữ liệu và muốn ký thác các thay đổi đó  cú pháp:  **commit;** |
| 118 | Trình bày về hàm rollback()?  phương thức rollback() sử dụng để xóa các thay đổi đã được thực hiện trước đó để quay về trạng thái trước khi thực hiện thay đổi khi thấy rằng có lỗi xảy ra  cú pháp:  **rollback;** |
| 119 | Trình bày về SavePoint trong Transaction?  savepoint giúp chúng ta lưu lại trạng thái của dữ liệu và các hành động của transaction tại một thời điểm cụ thể  cú pháp:  **savepoint savepoint\_name;** |
| 120 | **Xử lý theo mẻ Batch là gì? Ví dụ?**   * Xử lý hàng loạt (Batch processing) **là việc thực hiện hàng loạt công việc trong một chương trình trên máy tính mà không có sự can thiệp thủ công (không tương tác)**. |
| 121 | Phân biệt Transaction và Batch?  ko bik |