**BỘ CÂU HỎI PHỎNG VẤN MODULE 3**

**JAVA WEB BACK-END DEVELOPMENT**

|  |  |
| --- | --- |
| **STT** | **Câu hỏi** |
| **A** | **SQL** |
| 1 | Dữ liệu (data) là gì? Phân biệt hai khái niệm dữ liệu (data) và thông tin (information)?  - Dữ liệu (data) là các mô tả về sự vật, hiện tượng, sự kiện… được biểu diễn dưới dạng các tài liệu chữ, số, ảnh, âm thanh, hình ảnh…  - Thông tin: giá trị nhận được khi phân tích dữ liệu |
| 2 | Cơ sở dữ liệu (database) là gì?  Lưu trữ dữ liệu một cách tổ chức, có cấu trúc; lưu trữ và truy xuất dữ liệu tốt nhất có thể.  Database: MySQL – cơ sở dữ liệu dạng quan hệ, các bảng có mối liên hệ với nhau. |
| 3 | Giới thiệu một số hệ quản trị cơ sở dữ liệu hiện nay?  MySQL, MS SQL Server, Oracle, MS Access |
| 4 | Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) là gì?   * Là hệ thống phần mềm cho phép xử lý, truy xuất cơ sở dữ liệu. * Có thể nói rằng, dữ liệu trong RDBMS được lưu trữ trong các đối tượng cơ sở dữ liệu được gọi là các bảng. Bảng này về cơ bản là một bộ sưu tập các mục dữ liệu có quan hệ và nó bao gồm nhiều cột và hàng. |
| 5 | Toàn vẹn dữ liệu (data integrity) của một cơ sở dữ liệu là gì?  Tính toàn vẹn dữ liệu là dữ liệu hay thông tin không bị thay đổi, mất mát trong khi lưu trữ hay truyền tải. Nói cách khác tính toàn vẹn là tính không bị hiệu chỉnh của dữ liệu. Đây là một trong 4 khía cạnh trong an toàn điện tử: toàn vẹn, xác thực, không thoái thác, tin cậy/riêng tư, ích lợi. |
| 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 sơ đồ, thể hiện các thực thể có trong database, và mối quan hệ giữa chúng với nhau.  Thành phần: entity (thực thể), mối quan hệ.   * Mối quan hệ 1 – n: Đối với mối quan hệ 1 - nhiều chúng ta sẽ bổ sung khóa chính của bảng 1 làm khóa ngoại của nhiều bảng. * Mối quan hệ 1 – 1: Đối với mối quan hệ 1 - 1 chúng ta sẽ bổ sung khóa chính của bên này làm khóa ngoại của bảng bên kia. * Mối quan hệ n – m:   Thuộc tính đa trị: Đối với các thuộc tính đa trị của một thực thể chúng ta sẽ chuyển đổi thành một quan hệ mới (bảng mới) có khóa chính bao gồm 2 thuộc tính là thuộc tính đa trị và thuộc tính khóa của thực thể đồng thời loại bỏ thuộc tính đó khỏi thực thể cũ. |
| 7 | Ràng buộc (constraint) là gì? Các loại ràng buộc trong MySQL?  Ràng buộc (Constraint) là các quy tắc được áp dụng trên các cột dữ liệu của một  bảng. Chúng được sử dụng để kiểm tra tính hợp lệ của dữ liệu đầu vào, đảm  bảo tính chính xác, độ tin cậy và tính toàn vẹn của dữ liệu trong database.  \* Các loại ràng buộc:  +Not null:Đảm bảo dữ liệu của cột không được nhận giá trị NULL.  +DEFAULT:Cung cấp một giá trị mặc định cho một cột trong trường hợp dữ liệu  của cột không được nhập vào hay không được xác định.  +UNIQUE: dữ liệu của cột là duy nhất,không trùng lặp, truyền được null  +PRIMARY Key: Dùng để thiết lập khóa chính trên bảng, giá trị của cột làm khóa  chính phải là duy nhất, không được trùng lặp. Việc khai báo ràng buộc khóa  chính yêu cầu các cột phải NOT NULL.  +FOREIGN Key: Dùng để thiết lập khóa ngoại trên bảng, tham chiếu đến bảng  khác thông qua giá trị của cột được liên kết. Giá trị của cột được liên kết phải  là duy nhất trong bảng kia.  +CHECK: kiểm tra tính hợp lệ của dữ liệu  +INDEX: Dùng để tạo và lấy dữ liệu từ cơ sở dữ liệu một cách nhanh chóng. |
| 8 | Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?  Khóa chính là thuộc tính định danh một bản ghi, not null và unique.  Khóa chính phức hợp (composite) là khóa được tạo nên từ 2 hoặc nhiều cột. |
| 9 | Khóa ngoại (foreign key) là gì?  -Khoá ngoại là 1 ràng buộc cho 1 trường, thể hiện mối quan hệ giữa 2 bảng, 1 khoá ngoại để trỏ đến 1 bản ghi ở 1 bảng khác  -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?  -Khóa duy nhất là một tập hợp của một hoặc nhiều trường, cột của bảng xác định duy nhất một bản ghi trong bảng cơ sở dữ liệu.  -Primary key = Unique + Not Null, 1 bảng chỉ có thể có 1 khóa chính(PK) nhưng có thể có nhiều khóa duy nhất(UNi),và khóa duy nhất(UNI) có thể nhận 1 bản ghi là null còn khóa chính(PK) thì không. |
| 11 | Thuộc tính not null là gì?  Sử dụng để đảm bảo dữ liệu của cột không được nhận giá trị NULL |
| 12 | Trình bày ràng buộc mặc định (default)? Ví dụ?  • Từ khóa DEFAULT được sử dụng để quy định giá trị mặc định cho một trường  • Giá trị mặc định sẽ được sử dụng nếu không NULL được nhập vào  • Ví dụ: CREATE TABLE persons (  city varchar(255) DEFAULT 'Sandnes' ); |
| 13 | Trình bày ràng buộc kiểm tra (check)? Ví dụ?  Ràng buộc CHECK được sử dụng để quy định điều kiện ràng buộc về giá trị cho 1 cột. VD: check(age > 18) ý nghĩa : cột age có giá trị truyền vào phải lớn hơn 18. |
| 14 | Trình bày cách tự động gán và tăng giá trị trong 1 cột của MySQL?  Dùng Auto\_increment |
| 15 | Yêu cầu nhập vào trường tuổi của nhân viên phải từ 18-60, sử dụng ràng buộc nào? Viết ràng buộc?  Age Check (Age between 18 anh 60); |
| 16 | Yêu cầu trường cặp giá trị (MaSinhVien,MaMonHoc) phải duy nhất, not null, sử dụng ràng buộc nào?Viết ràng buộc?  ràng buộc:Unique,not null  PRIMARY KEY(MaSinhVien, MaMonHoc). |
| 17 | Yêu cầu Tỉnh thành chỉ được nhập Hà Nội hoặc TP. Hồ Chí Minh, sử dụng ràng buộc nào? Viết ràng buộc?  CHECK (name='Hà Nội ' OR name='TP. Hồ Chí Minh') |
| 18 | Yêu cầu trường Họ tên không được để trống sử dụng loại ràng buộc nào? Viết ràng buộc?  - Sử dụng rằng buộc NOT NULL  - Họ tên’ nvarchar(50) 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 (ClassID) references 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?  MSV varchar(3) check (MSV 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?  1, check(startDate > now())  2, StartDate DATETIME DEFAULT CURRENT\_DATE, |
| 22 | SQL là gì?  SQL là viết tắt của Structured Query Language là ngôn ngữ truy vấn có cấu trúc. Nó là một ngôn ngữ, là tập hợp các lệnh để tương tác với cơ sở dữ liệu. Dùng để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. |
| 23 | Trình bày về câu lệnh create?  Lệnh create thường được dùng để tạo CSDL và bảng trong MySQL. |
| 24 | Trình bày về câu lệnh alter?  Câu lệnh ALTER được dùng để sửa đổi cấu trúc của một bảng. VD: thêm,sửa xóa 1 hoặc nhiều cột, thêm sửa xóa rằng buộc của từng cột ..vv |
| 25 | Trình bày về các kiểu dữ liệu ký tự/chuỗi trong MySQL? Phân biệt giữa các loại?  Hình ảnh |
| 26 | Trình bày về các kiểu dữ liệu số trong MySQL? Phân biệt giữa các loại?  BIT : chỉ bao gồm 0 or 1 or NULL , INT(tinyint,smallint,bigint) : kiểu số nguyên , FLOAT(double,decimal,dec,numeric) : kiểu số thực. |
| 27 | Trình bày các kiểu dữ liệu ngày tháng trong MySQL? Phân biệt giữa các loại?  DATE:hiển thị ngày tháng  DATETIME: hiển thị cả thời gian  YEAR:hiển thị năm. |
| 28 | Yêu cầu bổ sung thêm khóa chính vào một bảng đã được tạo sẵn thì làm thế nào?  ALTER TABLE tablename  ADD PRIMARY KEY (col\_name); |
| 29 | Yêu cầu bổ sung thêm 1 cột vào bảng đã được tạo sẵn thì làm thế nào?  Alter table Ten\_bang add Ten\_cot Kieu\_du\_lieu; |
| 30 | Yêu cầu xóa một ràng buộc đã được tồn tại sẵn trong bảng thì làm thế nào?  ALTER TABLE <Table name>  Drop constraint <constraint\_name>; |
| 31 | Yêu cầu sửa kiểu dữ liệu của 1 cột có sẵn trong bảng thì làm thế nào?  alter table tên bảng  modify column cột KDL mới |
| 32 | Trình bày về câu lệnh insert? insert nhiều bộ giá trị?  câu lệnh insert: INSERT INTO table\_name (column\_list)  VALUES (value\_list);  insert nhiều giá trị: INSERT INTO table\_name (column\_list)  VALUES  (value\_list\_1),  (value\_list\_2),  ...  (value\_list\_n); |
| 33 | Trình bày về câu lệnh update? Phân biệt alter và update?  Update table\_name  Set column 1 = value 1, column 2 = value 2, column 3 = value 3, …   * ALTER: ngôn ngữ định nghĩa dữ liệu, thêm, sửa, xoá các thuộc tính của các mối quan hệ trong cơ sở dữ liệu.   UPDATE: ngôn ngữ thao tác dữ liệu sửa đổi các hàng trong bảng cơ sở dữ liệu |
| 34 | Trình bày về câu lệnh delete? Phân biệt drop và delete?  Delete from table\_name where condition;  DROP: xoá database, table… không thể phục hồi dữ liệu  DELETE: xoá các hàng trong table có thể phục hồi dữ liệu |
| 35 | Bạn có biết “safe update mode”? Nếu biết, hãy trình bày về cơ chế này?  Đây 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! |
| 36 | Trình bày về câu lệnh select?  - Câu lệnh SELECT trong SQL 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.  - Cú pháp: SELECT field1, field2, field3, ...  FROM table\_name  WHERE <dieu\_kien\_loc>  ORDER BY field\_name ASC|DESC  LIMIT start, limit  - Select có thể kết hợp với WHERE, ORDERBY, LIMIT  - Quy trình chạy của nó như sau: FROM -> WHERE -> SELECT, nghĩa là nó sẽ chọn từ bảng trước, sau đó kiểm tra điều kiện WHERE rồi mới quyết định chọn SELECT. |
| 37 | Trình bày về select lồng? Lấy ví dụ?  - Select lồng là 1 kiểu xử lý dữ liệu của T-SQL nói chung và mySQL nói riêng. Về cơ bản thì mySQL sẽ tạo một bảng ảo và sử dụng dữ liệu của bảng ảo đó để thực hiện tiếp các câu lệnh.  - Sử dụng select lồng có tốc độ xử lý chậm hơn do phải tạo 1 table ảo, từ đó cũng tốn tài nguyên lưu trữ hơn.  Vd:  Select id, name, price from product  Where price > (select avg(price) as avgPrice from product)  Order by price; |
| 38 | Trình bày về phép nối (join)?  -join==inner join: dùng để lấy giá trị giao giữa các bảng  -left join : dùng để lấy dữ liệu hoàn toàn của bảng 1( bảng ở from) và dữ liệu giao nhau(thông qua khóa ngoại) ở bảng 2( bảng ở inner)  -right join: dùng để lấy dữ liệu hoàn toàn của bảng 2(bảng ở inner) và dữ liệu giao nhau (thông qua khóa ngoại) ở bảng 1(bảng ở from).  VD:  Select name\_product, price, quantity, name\_category from product  Join category on product.id\_category = category.id\_category |
| 39 | Làm sao để có thể tăng tốc độ truy vấn?  Sử dụng select TEN\_COT from TEN\_BANG thay vì select\* from TEN\_BANG  Sử dụng Indexs và cố gắng truy vấn dữ liệu thông qua các điều kiện xác lập trên chỉ số.  Sử dụng EXPLAIN để phân tích truy vấn |
| 40 | Trình bày việc sao chép dữ liệu từ bảng này sang bảng khác trong MySQL?  \* Cú pháp 1:sao chép dữ liệu từ tất cả các cột của bảng table1 vào bên trong bảng table2  + INSERT INTO table2 SELECT \* FROM table1 WHERE condition;  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)  \* Cú pháp 2:sao chép dữ liệu từ một số cột nào đó của bảng table1 vào một số cột  nào đó của bảng table2  VD: INSERT INTO table2 (column1, column2, column3, . . . .)  SELECT column1,  column2, column3, . . . . FROM table1 WHERE condition; |
| 41 | Phân biệt giữa count(\*) và count(cột)?  Về cơ bản thì count(\*) và count(cột) có cùng chức năng là trả về số row trong table  -Count(\*): trả về số bản ghi trong một bảng (bao gồm bản ghi có giá trị null)  -Count(cột): trả về số bản ghi không null. |
| 42 | Làm sao để truy vấn dữ liệu từ nhiều bảng?   * Dùng join:   select … from table 1  Join table 2 on table 1. Primary key = table 2. Foreign key |
| 43 | Phân biệt giữa Where và Having?  HAVING: sử dụng để quy định các điều kiện trong trường hợp sử dụng các hàm SQL (không thể sử dụng câu lệnh where ví dụ như sử dụng group by)  WHERE: đưa ra điều kiện để lọc kết quả  Tóm lại:   * Having và where đều xử lý điều kiện. * Nhưng Having dùng để xử lý với hàm còn Where thì không.   Having có thể thay thế cho where nhưng where thì không. |
| 44 | Trình bày mệnh đề Order By trong câu lệnh Select?  Mệnh đề ORDER BY được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần(ASC) hoặc theo thứ tự giảm dần(DESC) trên một hoặc nhiều cột.nếu không có điều kiện thì mặc định sẽ là ASC. |
| 45 | Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?  SELECT Avg(student.studentPoint) from student |
| 46 | Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?  Sử dụng hàm avg và phân loại theo nhóm dùng ORDER BY..!  ViDu : SELECT name, avg(Student.mark) FROM Students ORDER BY name; |
| 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?  - dùng hàm avg với trường điểm để lấy điểm trung bình  - dùng where để hiển thị các bạn có điểm trung bình lớn hơn 5  Select name,avg(mark) from student  Where avg>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)?  Select student.studentName, class.className  from student  join class on student.ClassID = class.ClassID; |
| 49 | Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi?  select\*from students order by mark ; |
| 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 students ORDER BY studentName DESC |
| 51 | Hiển thị các sinh viên có họ là ‘Le’ và tên là ‘Hai’?  SELECT \* FROM students where name like "Le%Hai" |
| 52 | Hiển thị các sinh viên có tên bắt đầu là ‘C’?  Select \* From strudents where name like "C%" |
| 53 | Union là gì? Phân biệt Union và Union All?  -Toán tử UNION cho phép bạn nối kết quả của hai hoặc nhiều câu truy vấn lại với  nhau để trở thành một danh sách kết quả duy nhất.  -Phân biệt:  + UNION có tác dụng két quả hai lệnh SELECT và loại bỏ đi những dòng bị trùng  lặp chỉ để lại một dòng duy nhất  + UNION ALL có cú pháp và tác dụng tương tự như UNION, chỉ có điều là lệnh  UNION ALL sẽ giữ lại tất cả dữ liệu kể cả những dòng bị trùng lặp. |
| 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 hay index là một mô hình bảng tra cứu được Database Search Engine sử dụng để tăng tốc độ và hiệu suất truy xuất dữ liệu thông qua con trỏ.  Chỉ mục có tác dụng tăng tốc tìm kiếm, tiết kiệm tài nguyên hệ thống  Không nên sử dụng cho các bảng nhỏ  Các cột không thường được sử dụng làm điều kiện trong truy vấn  Không nên dùng chỉ mục trên các cột thường xuyên bị thay đổi  Không nên sử dụng trên các cột có chứa một số lượng lớn giá trị NULL. |
| 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?  Các kiểu index có trong SQL:  -Single-Column Index  -Unique Index(Chỉ mục duy nhất) :Các chỉ mục duy nhất không cho phép các giá trị lặp lại trong cột hoặc tổ hợp cột đánh chỉ mục. Dữ liệu lặp lại phải được cập nhật hoặc xoá trước khi chỉ mục duy nhất được tạo lập. Một giá trị rỗng trong một cột khoá phản ánh như một giá trị duy nhất.  -Composite Index: Là chỉ mục được tạo lập trên hai hoặc nhiều cột. Tạo lập một chỉ số phức hợp khi hai hoặc nhiều cột thường xuyên được tìm kiếm như một nhóm. Xác định danh sách các cột hợp thành giá trị duy nhất để tạo lập chỉ mục phức hợp. Thứ tự và số lượng các cột trong một chỉ mục tổ hợp ảnh hưởng đến sự thực thi của truy vấn.  -Implicit Index |
| 56 | Trình bày về khung nhìn (view)? Phân biệt view và table?  View cho phép chúng ta có thể tổng hợp kết quả từ nhiều bảng, tùy theo tiêu chí mong muốn.  View được coi như 1 bảng ảo có những khả năng tương tự như table thông thường, tuy nhiên là nó không hề lưu trữ dữ liệu về mặt vật lí, sẽ tự động được cập nhật khi thay đổi các table liên quan. |
| 57 | Cú pháp tạo view?  Cú pháp:  CREATE VIEW view\_name AS  SELECT column1, column2.....  FROM table\_name  WHERE [condition]; |
| 58 | Có thể tạo view từ view khác không?  Có |
| 59 | Trình bày về thủ tục lưu trữ (stored procedure)?  Có chức năng tương tự như một hàm, có tham số input, output, in-out, phần thân chứa các câu lệnh sql nào đó.  DELIMITER // CREATE PROCEDURE [procedure\_name] ([param1, param2,…])  BEGIN  [sql\_statements]  END // DELIMITER ; |
| 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 $$  create procedure hienThi (in input\_id int, out name varchar)  begin  name = (select student\_name from students where studentId = input\_id)  end $$  delimiter ; |
| 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?  select count (ten\_cot) from ten\_bang where input>5;  hoặc select ten\_cot, count(ten\_cot) from ten\_bang where input>5  group by ten\_cot; |
| 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” |
| **B** | **SQL** |
| 63 | Css là gì? Các cách nhúng CSS vào tài liệu html?  CSS (cascading style sheet).  • 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  3 cách nhúng:  + Khai báo trực tiếp ở thẻ HTML (Inline style)  + Khai báo trong thẻ <style> của tài liệu HTML (internal style sheet)  + Khai báo trong file.css riêng biệt (external style sheet) |
| 64 | Trình bày về các loại bộ chọn (selector) trong CSS?  • Id selector: 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.  • Class selector : 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:  • Tag selector: Chọn tất cả các phần tử |
| 65 | Phân biệt giữa bộ chọn class và bộ chọn id?  Về cơ bản ID và Class là giống nhau – sự khác biệt nằm ở chỗ, khi khai báo id cho một thẻ nào đó thì tên id đó không được dùng lại nữa – tức là bạn không được khai báo tên đó cho thẻ khác, trong khi đó class thì khai báo một lớp các thẻ và chúng sẽ có cùng chung thuộc tính trong CSS. |
| 66 | Trình bày mô hình hộp (box model) trong CSS?  • Mỗi phần tử HTML có thể được coi như là một chiếc hộp •  Trong CSS thuật ngữ mô hình hộp (box model) được dùng để nói về việc thiết kế bố cục (layout)  • Về cơ bản mô hình hộp CSS là một chiếc hộp bao quanh phần tử HTML như:  ✓Content: nội dung trong hộp, nơi hiển thị văn bản và hình ảnh  ✓Padding: Khoảng cách từ nội dung đến viền, chính là phần khoảng trắng xung quanh nội dung, trong suốt  ✓Border: Đường viền xung quanh nội dung và padding  ✓Margin: Phần lề bên ngoài đường viền, trong suốt |
| 67 | Thuộc tính margin dùng để làm gì?  - Tạo không gian xung quanh các phần tử  - Nằm ngoài border |
| 68 | Phân biệt margin, padding và border?  Padding: là không gian trống bao bọc phần content, nằm giữa content và border  Border: là đường viền bọc ngoài cùng của padding và content  Margin: là không gian trống bên ngoài phần border, khoảng cách từ lề đến boder |
| 69 | Thuộc tính position dùng để làm gì? Phân biệt các giá trị: static, absolute, relative và fixed?  - static: mặc định  - relative: định vị so với vị trí bình thường của nó  - absolute: định vị so với thẻ cha được định vị gần nhất. Tuy nhiên, nếu thẻ cha chưa được định vị thì nó sẽ định vị theo body và di chuyển cùng với cuộn trang  - fixed: dược đặt ở cùng 1 vị trí ngay cả khi cuộn trang |
| 70 | Trình bày thuộc tính float, clear?  - **Thuộc tính Float** sử dụng để chuyển một phần tử sang góc trái hoặc phải của không gian bao quanh nó, rất cần thiết trong việc định dạng bố cục trang.   * **left**: Cố định phần tử về bên trái. * **right:** Cố định phần tử về bên phải. * **none:** Nằm tại chính vị trí của nó (trạng thái bình thường). * **inherit** (kế thừa): Phần tử kế thừa giá trị từ float cha.   - **Clear** gần như là ngược lại với *float.* Thuộc tính *Clear* ngăn chặn thành phần A chiếm vùng không gian của thành phần B (với thành phần B là thành phần sử dụng *float).* Đôi khi không muốn float ở một số tình huống nào đó ta sẽ dùng clear để khắc chế.  Nói một cách đơn giản, *clear* sử dụng để giải quyết vấn đề trong mục **chú ý** của thuộc tính *float* phía trên.  Thuộc tính *float* có thể có một trong các giá trị sau:   * **left:** Tràn về phía bên trái. * **right:** Tràn về phía bên phải. * **none:** Cho phép tràn lên cả hai phía (mặc định). * **both:** Không cho phép tràn về bên nào * **inherit** (kế thừa): Phần tử kế thừa giá trị từ float cha. |
| 71 | Tạo bố cục trang trong html5 với các thẻ cơ bản?  bố cục trang web gồm : header, menu,left,content,right,footer |
| 72 | Bạn có biết Prototype? Trình bày?  **Prototype** là cơ chế để thực hiện mô hình OOP của ngôn ngữ lập trình [JavaScript](https://vietnix.vn/javascript-la-gi/), mà các object (đối tượng) kế thừa các tính năng từ nhau. Mỗi một object trong Javascript đều có một thuộc tính nội bộ (internal property) gọi là prototype. |
| 73 | Responsive Web Design là gì?  RWD (Thiết kế web thích ứng) là kỹ thuật được sử đụng để trang web có thể  hiển thị tốt trên nhiều loại thiết bị với kích thước và độ phân giải khác nhau |
| 74 | Trình bày cú pháp RWD Media Queries?  @media (min-width: 1025px) and (max-width: 1280px) |
| **C** | **SERVLET - JSP - JDBC** |
| 75 | 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 cơ sở dữ liệu.  • Những trang web trong website tĩnh được viết bằng mã HTML hoặc thêm CSS, JavaScript để thêm các hiệu ứng  Dyamic web  Website động là website có nội dung thay đổi.  • Sự thay đổi có thể là tùy theo thời gian, tùy theo người dùng, tùy theo ngữ cảnh  • Để tạo được website động, chúng ta thường sử dụng đến các ngôn ngữ phía server (server-side), chẳng hạn như Servlet, JSP, PHP, Python, C#...  • Một website động thường được đặt trên một máy chủ dịch vụ web (Web Server |
| 76 | Giao thức (protocol) là gì? Kể tên một số giao thức bạn biết?  Giao thức mạng protocol được định nghĩa là một bộ quy tắc và quy định để xây dựng một mạng đáng tin cậy cho một hệ thống nhúng hoặc hệ thống nhúng phức tạp. Và thực hiện giao tiếp giữa các máy tính, máy chủ, bộ định tuyến và bất kỳ thiết bị hỗ trợ mạng nào khác.  Một số giao thức em biết là :  +Giao thức điều khiển truyền (TCP)  +Giao thức gói dữ liệu người dù (UDP)  + Giao thức truyền siêu văn bản ( HTTP)  + Giao thức truyền tệp (FTP)  Mở rộng Wifi , Bluetooth , LTE cũng được coi là giao thức mạng. |
| 77 | HTTP là gì? Phân biệt HTTP và HTTPs?  HTTP là tên viết tắt của HyperText Transfer Protocol (giao thức truyền tải siêu văn bản), là một giao thức cơ bản dùng cho World Wide Web (www) để truyền tải dữ liệu dưới dạng văn bản, hình ảnh, video, âm thanh và các tập tin khác từ Web server đến các trình duyệt web và ngược lại. |
| 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. TCP/IP được phát triển để mạng được tin cậy hơn cùng với khả năng phục hồi tự động. |
| 79 | Trình bày mô hình Client/Server?  Mô hình Client Server là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó. |
| 80 | Máy chủ (Server) là gì?  Server (máy chủ) là một máy có chức năng cung cấp thông tin hoặc lưu trữ thông tin do các máy khách (Client) gửi lên hoặc tải xuống thì được gọi là Server. |
| 81 | Phân biệt máy chủ web và máy chủ ứng dụng? |
| 82 | Tomcat là gì? Tomcat thuộc Web Server hay Application Server?  Tomcat là một ứng dụng chủ (Application Server), là một phần mềm mã nguồn mở,nó có khả năng hỗ trợ mạnh cho các ứng dụng Java thay vì các website tĩnh khác. DO vậy, nó có khả năng chạy trên nhiều bản Java chuyên biệt như: Java Servlet, JavaServer Pages (JSP), Java EL và WebSocket. |
| 83 | Phân biệt HTTP Request và HTTP Response?   * HTTP Request:   Request có thể hiểu nhanh là thông tin gửi từ client lên server. Khi bạn lên trình duyệt [browser](https://freetuts.net/thu-thuat/browser-la-gi-2264t.html) gõ một địa chỉ nào đó, ví dụ bạn gõ là freetuts.net thì ngay lập tức trình duyệt sẽ dựa vào tên domain để gửi yêu cầu truy cập đến địa chỉ IP mà domain này đang trỏ tới (bạn đừng quan tâm đến domain này do ai quản lý), lúc này phía server sẽ phân tích yêu cầu và sẽ gửi luồng xử lý tới vị trí vị trí lưu trữ của mã nguồn PHP (hoặc mã nguồn bất kì) và nhiệm vụ của các mã nguồn là tiếp nhận yêu cầu, phân tích request đó và trả kết quả lại cho client.   * HTTP Response:   Reponse là dữ liệu mà server trả về cho client. Ví dụ khi bạn nhập vào địa chỉ freetuts.net thì kết quả trả về (response) chính là giao diện của website và các thông tin của header. Như vậy dữ liệu mà server trả về là những đoạn mã HTML kèm theo các thông tin của header |
| 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. Nó đã được đề xuất bởi Bell Communications vào năm 1991, và các đặc điểm kỹ thuật ban đầu được xác định trong tháng 6 năm 1992 cho RFC 1341 và 1342. |
| 85 | Phân biệt rõ hai phương thức Get và Post?  Phương thức GET và POST được sử dụng để gửi dữ liệu đến server và sự khác biệt chính giữa chúng là phương thức GET nối dữ liệu vào URI được xác định trong thuộc tính action của form. Ngược lại, phương thức POST đính kèm dữ liệu vào phần body được yêu cầu. Việc sử dụng phương thức GET là không phù hợp khi thông tin nhạy cảm cần được điền vào form. Phương thức POST hữu ích khi người dùng yêu cầu điền mật khẩu hoặc thông tin bí mật khác. |
| 86 | Trình bày lớp HTTPServlet?  Servlet là một công nghệ được sử dụng để tạo ra ứng dụng web.  Servlet là một API cung cấp các interface và lớp bao gồm các tài liệu.  Servlet là một thành phần web được triển khai trên máy chủ để tạo ra trang web động. |
| 87 | Trình bày Servlet Container?  Servlet container là một cái "thùng" để chứa rất nhiều servlet. Và servlet là một class tượng trưng cho một cặp http request và response. |
| 88 | Vòng đời của Servlet?  Servlet được khởi tạo bằng cách gọi phương thức init ().  Phương thức servlet service() được gọi để xử lý yêu cầu của khách hàng.  Servlet được hủy bằng cách gọi phương thức destroy().  Cuối cùng, servlet được thu thập bởi bộ sưu tập rác của JVM. |
| 89 | Trình bày Servlet Interface? Mô tả các phương thức init(), service(), destroy()?  Interface Servlet định nghĩa các phương thức trong vòng đời của một Servlet |
| 90 | Phân biệt các loại Servlet Container: Standalone, In-process và Out-of-Servlet?  -Standalone: là kiểu máy chủ dựa trên java điển hình, trong đó servlet container và máy chủ web là được tích hợp trong 1 chương trình duy nhất. Ví dụ: Tomcat  -In-process: servlet container được tách riêng ra khỏi máy chủ web, chạy trong không gian địa chỉ của máy chủ chính như 1 chương trình có thể cài cắm. Ví dụ: Tomcat chạy trong JBoss.  -Out-of-process: Máy chủ web và servlet container là các chương trình khác nhau được chạy trong các tiến trình riêng biệt. Để thực hiện các giao tiếp giữa chúng, máy chủ web sử dụng 1 chương trình cài cắm đặc thù cho (và thường là cung cấp bởi) servlet container . |
| 91 | Giới thiệu một số phương thức của lớp HttpServlet?  - protected void doGet  - protected void doPost |
| 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()?  doGet:cần tham số trên thanh url khi truy vấn ( chỉ dùng truy xuất lượng nhỏ dữ liệu và không nhạy cảm)  doPost:không cần tham số trên thanh url khi truy vấn (dùng khi truy xuất luọng dữ liệu lớn và nhạy cảm ) |
| 94 | RequestDispatcher được dùng làm gì?  Giúp cho trang web chuyển request từ servlet này sang một servlet khác  Để gọi RequestDispatcher Ta sử dụng phương thức include() hoặc forward()  Forward() method:  Chuyển một request từ một servlet tới resource khác trên server và chuyển hoàn toàn quyền cho servlet được chuyển tới. Trước khi forward, nó không xử lý gì về nội dung và đến servlet cuối cùng mới thực hiện điều đó (print ra màn hình)  Include() method  Để nhận kết quả trả về từ trang đích và tiếp tục xử lý. Kết quả của servlet được gọi sẽ được ghép với servlet được kích hoạt |
| 95 | Phân biệt forward() của RequestDispatcher và SendRedirect()  RequestDispatcher:dùng khi cần chuyển trang và mang theo dữ liệu  SendRedirect():dùng khi cần chuyển trang mà k cần mang theo dữ liệu |
| 96 | Jsp là gì? Nói: “Về bản chất Jsp cũng chính là Servlet” đúng hay sai? Giải thích?  Hình ảnh |
| 97 | So sánh giống và khác nhau giữa Jsp và Servlet?  • Về bản chất, JSP cũng chính là Servlet. Vì trong quá trình biên dịch, JSP sẽ được chuyển thành Servlet rồi Servlet này sẽ được biên dịch. Servlet của trang JSP có vòng đời giống như các Servlet thông thường.  Giống nhau:  • Đều nằm ở phía server.  • Đều xử lý dữ liệu “động”.  • Đều chạy với web container.  • Đều là những công nghệ quan trọng của Java EE. So sánh JSP và Servlet  Khác nhau:  • Servlet rất mạnh về xử lý và điều phối, nhưng lại rất yếu về tạo giao diện và bảo trì web.  • JSP mạnh về xử lý hiển thị nhưng lại yếu về xử lý nghiệp vụ và điều phối.  • Trong thực tế, chúng ta kết hợp sức mạnh của Servlet và JSP vào mô hình MVC thì: Servlet đóng vai trò Controller, thì JSP đóng vai trò View.  • Ở Servlet, mã HTML nằm trong mã Java. Còn ở JSP, mã Java nằm trong mã HTML |
| 98 | Vòng đời của Jsp?  - Vòng đời của JSP được định nghĩa là quá trình từ khi nó được tạo ra cho đến khi nó bị hủy.  - Vòng đời của JSP bao gồm 4 pha sau:  +Biên dịch (Compilation).  +Khởi tạo (Initialization).  +Thực thi (Execution).  +Hủy (Cleanup). |
| 99 | Jstl là gì? Giới thiệu một số thẻ bạn đã sử dụng jstl?  JSP Standard Tag Library (JSTL) là thư viện thẻ chuẩn cung cấp các thẻ để kiểm soát hành vi trang, lặp đi lặp lại và các lệnh điều khiển, các thẻ quốc tế hóa, và các thẻ SQL. <c:forEach>:Dùng để lặp trên một tập hợp.  <c:out>: |
| 100 | Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC?  • MVC là viết tắt của Model-View-Controller  • Kiến trúc MVC tổ chức các thành phần của hệ thống vào 3 tầng riêng biệt và có kết nối đến nhau:  • 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 6 |
| 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? |
| 103 | Trình bày ưu và nhược điểm của kiến trúc MVC?  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. Phải có kiến thức vững và chuẩn, để tránh bị nhầm lẫn về các mô hình.  Ngược lại, nếu bạn nắm bắt tốt thì **MVC** mang lại **ưu điểm** vô cùng lớn: Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì.. |
| 104 | Sự giống, khác nhau và ưu nhược điểm của MVC1 & MVC2?  **Sự khác, giống nhau, ưu, nhược điểm của MVC1 & MVC2.**   * Trong MVC1, controller và view đều là Jsp. MVC1 là thế hệ đầu tiên sử dụng JSP/ JavaBeans để thực hiện mô hình MVC, yêu cầu từ phía client sẽ được gửi tới trang JSP ban đầu được coi là controller xử lý, sau đó truyền lệnh thực thi đến các JavaBeans, các JavaBeans lấy dữ liệu từ database trả về cho trang JSP ban đầu theo yêu cầu phía client. * MVC2 là một mô hình mô tả kiến trúc MVC trên nền web do Sun phát minh, khác với MVC1 trong MVC2 các Controller là các Servlet còn Model là các class của Java, view là các trang JSP. Ở MVC2 ta thấy các logic phức tạp được viết trong Controller (Hard working), các servlet nhận yêu cầu từ client qua sử lý được gửi cho Model và View để tương tác trả kết quả cho client. * Ta thấy MVC2 không kế thừa từ MVC1 nhưng làm việc hiệu quả và logic hơn. * MVC2 tốt cho các dự án lớn nhưng MVC1 thì không. * Với thành được phân chia theo chức năng, MVC2 có thể dễ dàng tái sử dụng.   Cũng chính vậy mà MVC2 phức tạp hơn MVC1. |
| 105 | API là gì?  API (Application programming interface - Giao diện lập trình ứng dụng) là một tài liệu có chứa mô tả về tất cả các tính năng của một sản phẩm hoặc phần mềm. Nó đại diện cho các lớp và các giao diện (interface) mà các chương trình phần mềm có thể làm theo để giao tiếp với nhau. Một API có thể được tạo cho các ứng dụng, thư viện, hệ điều hành, ... |
| 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; …  Image |
| 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 các kết nối 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, PreparedStatement, CallableStatement: Chứa lệnh SQL gửi tới cơ sở dữ liệu để thực thi.  • 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 cơ sở dữ liệu |
| 108 | Trình bày Statement? Phân biệt Statement, PreparedStatement và CallableStatement?  -Statement được sử dụng để thực thi các câu lệnh SQL tĩnh, chúng ta không thể truyền tham số vào câu SQL trong thời gian runtime.  -Statement có hiệu năng (performance) kém hơn  PrepareStatement và CallableStatement .  -Statement thường được sử dụngtrong trường hợp câu lệnh SQL chỉ chạy 1 lần, ví dụ sử dụng để thực thi các câu SQL định nghĩa cơ sở dữ liệu – DDL (Data Denifition Language) như CREATE, ALTER, DROP…  Phân biệt:   * **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?  Prepared Statement (còn được gọi là câu lệnh được tham số hóa) chỉ đơn giản là một mẫu truy vấn SQL có chứa trình giữ chỗ thay vì các giá trị tham số thực tế.  Các trình giữ chỗ này sẽ được thay thế bằng các giá trị thực tế tại thời điểm thực hiện câu lệnh.  MySQLi hỗ trợ sử dụng trình giữ chỗ vị trí ẩn danh (?), Như được hiển thị bên dưới:  INSERT INTO persons (first\_name, last\_name, email) VALUES (?, ?, ?); |
| 110 | Hướng dẫn sử dụng CallableStatement?  CallableStatement Interface được sử dụng để thực thi Stored Procedure |
| 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?  PreparedStatement statement = connection.prepareStatement("UPDATE product SET name =?,price=? where id =?" ) |
| 113 | Sử dụng CallableStatement trong tính năng xóa sản phẩm có id cho trước? |
| 114 | Gọi MySQL Stored Procedures từ JDBC như thế nào? |
| 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  • Transaction 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?  Transaction có bốn ưu điểm tiêu chuẩn sau :  Bảo toàn : đảm bảo rằng tất cả các câu lệnh trong nhóm lệnh được thực thi thành công.  Nếu không, transaction bị hủy bỏ tại thời điểm thất bại và tất cả các thao tác trước đó được khôi phục về trạng thái cũ  Nhất quán : đảm bảo rằng cơ sở dữ liệu thay đổi chính xác các trạng thái khi một transactionđược thực thi thành công.  Độc lập : cho phép các transaction hoạt động độc lập và minh bạch với nhau.  Bền bỉ : đảm bảo rằng kết quả của một transaction được commit vẫn tồn tại trong trường hợp lỗi hệ thống. |
| 117 | Trình bày về commit()?  Phương thức commit() được sử dụng khi bạn đã thực hiện các thay đổi với cơ sở dữ liệu và muốn ký thác các thay đổi đó |
| 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 |
| 119 | Trình bày về SavePoint trong Transaction?  -SAVEPOINT là một điểm trong một transaction khi bạn có thể cuộn transaction trở lại một điểm nhất định mà không quay trở lại toàn bộ transaction.  -Cú pháp:  SAVEPOINT SAVEPOINT\_NAME;  -Lệnh này chỉ phục vụ trong việc tạo ra SAVEPOINT trong số tất cả các câu lệnh transaction. |
| 120 | Xử lý theo mẻ Batch là gì? Ví dụ?  Trong JDBC, mỗi lần bạn thực hiện insert, update, delete hoặc select, ứng dụng sẽ kết nối tới database một lần.  Giả sử, bạn cần insert 10000 bản ghi vào databaseTheo cách thông thường, chúng ta thực hiện insert 10000 lần thì sẽ phải kết nối tới database 10000 lần để thao tác với database –> tốn rất nhiều thời gian  -> Ý tưởng dùng Batch (xử lý theo mẻ), tức là chúng ta sẽ kết nối 1 lần nhưng insert 10000 bản ghi cùng lúc như vậy sẽ tiết kiệm được thời gian của 9999 kết nối tới database. |
| 121 | Phân biệt Transaction và Batch?  -Transaction 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  **-**Batch: Xử lý hàng loạt là thực hiện một loạt các chương trình (công việc) trên máy tính mà không cần sự can thiệp thủ công. |