

1.Cơ sở dữ liệu là gì?	<ul style="list-style-type: none"> - Cơ sở dữ liệu là một hệ thống thông tin có cấu trúc được lưu trữ trên các hệ quản trị cơ sở dữ liệu nhằm thỏa mãn nhu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau - Cơ sở dữ liệu quan hệ là CSDL được xây dựng trên mô hình dữ liệu quan hệ - Hệ quản trị cơ sở dữ liệu là phần mềm dùng để tạo lập, lưu trữ và khai thác và thao tác trên CSDL - SQL : Structured Query Language
2. Phân biệt các câu lệnh DML & DDL?	<ul style="list-style-type: none"> - DML: Data manipulation language: ngôn ngữ thao tác dữ liệu + Insert, Update, Delete, Select - DDL: Data define language: ngôn ngữ định nghĩa dữ liệu + Create, Alter, Drop, Rename
3.Trình bày về mô hình thực thể mối quan hệ (ERD - Entity-Relationship Diagram)? Trong mô hình thực thể mối quan hệ có những thành phần nào?	<p>ERD là một sơ đồ, thể hiện các thực thể có trong database, và mối quan hệ giữa chúng với nhau.</p> <p>Các thành phần có trong ERD:</p> <ul style="list-style-type: none"> - Thực thể - Thuộc tính - Mối quan hệ giữa các thực thể
4. Các bước chuyển đổi từ ERD -> Table?	<p>B1: Biến đổi thực thể thành table theo nguyên tắc:</p> <ul style="list-style-type: none"> -Thuộc tính khóa của thực thể là khóa chính của table - Các thuộc tính thông thường của thực thể là thuộc tính của table <p>B2: Biến đổi thuộc tính đa trị</p> <ul style="list-style-type: none"> - Tách thuộc tính đa trị thành bảng mới, xác định mối quan hệ giữa bảng mới và bảng hiện tại

	<p>B3: Biến đổi quan hệ 1:1</p> <ul style="list-style-type: none"> - Đặt thuộc tính khóa ở phía tùy chọn sang bắt buộc <p>B4: Biến đổi mối quan hệ 1-n:</p> <ul style="list-style-type: none"> - Đặt khóa chính bên 1 là khóa ngoại bên n <p>B5: Biến đổi mối quan hệ n-n:</p> <ul style="list-style-type: none"> - Tách thành bảng mới, có khóa chính là tập hợp 1 khóa của 2 thực thể tham gia vào mối quan hệ => gọi là khóa phức hợp
<p>5. Các bước chuẩn hóa dữ liệu?</p>	<p>+ Dạng chuẩn 1NF:</p> <ul style="list-style-type: none"> - Các thuộc tính của bảng phải là nguyên tố - Giá trị của các thuộc tính trên các hàng phải là đơn trị, không chứa nhóm lặp - Không có một thuộc tính nào có giá trị có thể tính toán được từ một thuộc tính khác <p>+ Dạng chuẩn 2NF:</p> <p>Quy tắc chuẩn hóa từ chuẩn 1NF thành 2NF:</p> <p>Bước 1: Loại bỏ các thuộc tính không khóa phụ thuộc vào một bộ phận khóa chính và tách ra thành một bảng riêng, khóa chính của bảng là bộ phận của khóa mà chúng phụ thuộc vào.</p> <p>Bước 2: Các thuộc tính còn lại lập thành một quan hệ, khóa chính của nó là khóa chính ban đầu.</p> <p>+ Dạng chuẩn 3NF:</p> <p>Điều kiện:</p> <p>Phải đạt chuẩn 2NF</p> <p>Mọi thuộc tính không khóa phụ thuộc bắc cầu vào thuộc tính khóa (nghĩa là tất cả các thuộc tính không khóa phải được suy ra trực tiếp từ thuộc tính khóa)</p> <p>Quy tắc chuẩn hóa từ 2NF thành 3NF:</p> <p>Bước 1: Loại bỏ các thuộc tính phụ thuộc bắc cầu ra khỏi quan hệ và tách chúng thành quan hệ riêng có khóa chính là thuộc tính bắc cầu.</p> <p>Bước 2: Các thuộc tính còn lại lập thành một quan hệ có khóa chính là khóa ban đầu.</p>
<p>6. Ràng buộc (constraint) là gì? Các loại ràng buộc trong MySQL?</p>	<ul style="list-style-type: none"> - Constraint là các quy tắc quy định cho bảng, hạn chế dữ liệu có thể đưa vào trong bảng giúp cho dữ liệu chính xác, tin cậy và toàn vẹn. - Các loại ràng buộc: <p>+ Not null: không cho phép giá trị null</p> <p>+ Unique: mỗi giá trị là duy nhất</p>

	+ Primary key: khóa chính (not null + unique) + Foreign key: khóa ngoại dùng để tham chiếu đến bảng khác + Check: kiểm tra dựa vào điều kiện + Default: quy định giá trị mặc định cho trường nếu không có giá trị nhập vào + Index: giúp tăng tốc độ truy vấn	
7.Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?	<ul style="list-style-type: none"> Khóa chính (Primary Key, hay ràng buộc khóa chính) được sử dụng để định danh duy nhất mỗi bản ghi (dòng) trong bảng của cơ sở dữ liệu. <p>Khóa chính có thể tạo ra dựa trên một cột hoặc nhiều cột (được gọi là khoá chính tổ hợp) của table.</p>	
8. Khóa ngoại (foreign key) là gì?	Khoá ngoại (foreign key - viết ngắn gọn là FK) là cơ chế để tạo liên kết giữa các bảng trong cùng một cơ sở dữ liệu.	
9.Phân biệt DELETE và TRUNCATE?	Cú pháp: Delete: DELETE FROM table_name; Truncate: TRUNCATE TABLE table_name;	
	DELETE	TRUNCATE

	<ul style="list-style-type: none"> - Delete có thể xóa từng dòng hoặc xóa cả bảng, khi xóa thì SQL sẽ log lại từng dòng đã xóa vào transaction log, vì thế khi tạo 1 record mới thì id sẽ có giá trị n+1 với n là giá trị của record cuối cùng được tạo - Xóa bằng delete thì sẽ restored được - Chậm hơn truncate 	<ul style="list-style-type: none"> - Truncate chỉ có thể xóa hết bảng và reset transaction log, vì thế khi tạo một record mới thì giá trị id sẽ bắt đầu từ 1 - Xóa bằng truncate thì không restored được - Xóa bằng truncate sẽ nhanh hơn vì nó tiêu tốn ít bộ nhớ hơn
10. Các loại JOIN? Phân biệt?	<ul style="list-style-type: none"> + Inner join: Lấy phần tử chung giữa 2 bảng + Outer join: <ul style="list-style-type: none"> - Left join: Lấy phần chung và toàn bộ bảng bên trái - Right join: Lấy phần chung và toàn bộ bảng bên phải - Full join lấy hết 	
11. Phân biệt JOIN và UNION?	<ul style="list-style-type: none"> + JOIN dùng để kết hợp 2 hay nhiều bảng có mối quan hệ với nhau + UNION dùng để kết hợp kết quả của hai hay nhiều câu truy vấn lại với nhau 	
12. Phân biệt UNION và UNION ALL?	<ul style="list-style-type: none"> + UNION & UNION ALL đều dùng để kết hợp kết quả của hai hay nhiều câu truy vấn lại với nhau. Trong đó: UNION loại bỏ các record trùng lặp, còn UNION ALL giữ lại tất cả bản ghi 	
13. Phân biệt EXIST và IN?	<p>Còn về IN và EXIST thì cách sử dụng không hề khác nhau. Điểm khác biệt là về performance.</p> <p>Exist nhanh hơn</p> <p>In sẽ tìm ra tất cả các bộ thỏa yêu cầu câu truy vấn con rồi mới so sánh với câu truy vấn mẹ</p> <p>Exists sẽ quét từng bộ ở câu truy vấn con và so sánh trực tiếp với yêu cầu của mẹ, nếu không thỏa là loại trừ ngay</p>	
14.	<ul style="list-style-type: none"> - Biểu thức chứa toán tử and trả về true nếu tất cả điều kiện là đúng và 	

Phân biệt AND và OR?	<p>trả về false nếu có ít nhất 1 điều kiện là sai</p> <p>- Biểu thức chứa toán tử or trả về true nếu có ít nhất 1 điều kiện là đúng, và trả về false nếu không có điều kiện nào đúng</p>
15. Phân biệt GROUP BY và ORDER BY?	<p>- GROUP BY: dùng để nhóm các tập kết quả dựa theo giá trị của một cột hoặc nhiều cột</p> <p>- ORDER BY: dùng để sắp xếp dữ liệu theo tăng dần hoặc giảm dần theo cột chỉ định theo trật tự Mệnh đề ORDER BY được dùng để lọc bản ghi trong bộ kết quả trả về của <u>SQL Server</u>.</p> <p>ASC - tùy chọn. Lọc kết quả theo thứ tự tăng dần của biểu thức (mặc định nếu không chỉ định rõ).</p> <p>DESC - tùy chọn. Lọc kết quả theo thứ tự giảm dần của biểu thức.</p>
16. Muốn nối chuỗi trong MySQL thì sử dụng gì?	Dùng hàm concat(str1, str2)
17. Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?	SELECT AVG(Diem) FROM SINHVIEN
18	Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?
	<pre>SELECT id, name, AVG(Diem) as Diem FROM SINHVIEN GROUPBY id, name;</pre>
19	Yêu cầu hiển thị tên sinh viên và điểm trung bình tương ứng, chỉ hiển thị các bạn có điểm trung bình lớn hơn 5 thì làm thế nào?
	<pre>SELECT id, name, AVG(Diem) as Diem FROM SINHVIEN GROUPBY id, name; HAVING AVG(Diem) > 5;</pre>

20	Hiển thị danh sách sinh viên và lớp học tương ứng thì làm thế nào (hai bảng Student và Class có cột chung ClassID)?
	<code>SELECT st.name_student , cls.class_name FROM student st JOIN class cls ON cls.class_id = st.class_id;</code>
21	Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi?
	<code>SELECT * FROM student ORDER BY mark;</code>
22	Hiển thị danh sách sinh viên theo thứ tự giảm dần của họ tên, nếu họ tên trùng nhau thì sắp xếp theo tăng dần của tuổi?
	<code>SELECT * FROM student ORDER BY name_student DESC, age;</code>
23	Hiển thị các sinh viên có họ là 'Le' và tên là 'Hai'?
	<code>SELECT * FROM student WHERE student_name LIKE 'Le%' AND student_name LIKE '%Hai';</code>
24	Hiển thị các sinh viên có tên bắt đầu là 'C'?
	<code>SELECT * FROM student HAVING (REVERSE(SUBSTRING(REVERSE(student_name), 1, INSTR(REVERSE(student_name,' ')) LIKE 'C%';</code>
25	Làm sao để có thể tăng tốc độ truy vấn? Nhược điểm là gì?
	Dùng INDEX để tăng tốc độ truy vấn Nhược điểm: thêm, xóa, sửa dữ liệu bị chậm và chiếm dụng bộ nhớ
26	Các dạng tham số ở SP?
	<p>IN: Đây là chế độ mặc định (nghĩa là nếu bạn không định nghĩa loại nào thì nó sẽ hiểu là IN). Khi bạn sử dụng mức này thì nó sẽ được bảo vệ an toàn, có nghĩa là sẽ không bị thay đổi nếu như trong Procedure có tác động đến</p> <p>OUT: Chế độ này nếu như trong Procedure có tác động thay đổi thì nó sẽ thay đổi theo. Nhưng có điều đặc biệt là dù trước khi truyền vào mà bạn gán giá trị cho biến đó thì vẫn sẽ không nhận được vì mặc định nó luôn hiểu giá trị truyền vào là NULL.</p> <p>INOUT: Đây là sự kết hợp giữa IN và OUT. Nghĩa là có thể gán giá trị trước và có thể bị thay đổi nếu trong Procedure có tác động tới</p>
27	Phân biệt SP, Trigger, Function?

	<ul style="list-style-type: none"> - Store Procedure: <ul style="list-style-type: none"> + Có các tham số IN, OUT, INOUT + Có thể 1 hoặc nhiều kết quả select + Có thể gọi thủ tục hoặc hàm khác + Thực thi khi được gọi - Function: <ul style="list-style-type: none"> + Chỉ nhận các giá trị tham số vào + Trả về 1 giá trị duy nhất qua return + Chỉ có thể gọi hàm khác + Thực thi khi được gọi - Trigger: <ul style="list-style-type: none"> + Không có các tham số đầu vào + Không có giá trị trả về + Không thể gọi trigger trong 1 trigger khác
28	CSS là gì? Các cách nhúng CSS vào tài liệu html? Độ ưu tiên?
	<ul style="list-style-type: none"> - CSS là Cascading style sheet: mô tả cách các thẻ html hiển thị trên trình duyệt - Các cách nhúng <ul style="list-style-type: none"> + Inline style + trong cặp thẻ <style>: Internal style + link file .css: External style - Ưu tiên: <ol style="list-style-type: none"> 1. Inline style 2. Internal và External tùy thuộc vào khai báo, cái nào khai báo sau thì độ ưu tiên cao hơn (ghi đè lại)
29	Phân biệt giữa bộ chọn class và bộ chọn id?
	<ul style="list-style-type: none"> + Bộ chọn id: chọn 1 phần tử vs id xác định, dùng dấu # + Bộ chọn class: chọn 1 hoặc nhiều phần tử, dùng dấu .
30	Trình bày mô hình hộp (box model) trong CSS?
	<ul style="list-style-type: none"> - Box model là hộp bao quanh 1 thẻ html - Có 4 thành phần:

	<ul style="list-style-type: none"> + content: nội dung + padding: khoảng cách từ content => border + border: độ dày đường viền + margin: khoảng cách từ border của thẻ đến border của thẻ khác
31	Trình bày thuộc tính float, clear?
	<p>- Float:</p> <p>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.</p> <p>Theo mặc định, tất cả các phần tử HTML là <i>không float</i>.</p> <p>Thuộc tính <i>Float</i> có thể có một trong các giá trị sau:</p> <ul style="list-style-type: none"> • 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. <p>- Clear:</p> <p>Clear gần như là ngược lại với <i>float</i>. Thuộc tính <i>Clear</i> ngăn chặn thành phần A chiếm vùng không gian của thành phần B (với thành phần B là thành phần sử dụng <i>float</i>). Đôi khi không muốn float ở một số tình huống nào đó ta sẽ dùng clear để khắc chế.</p> <p>Nói một cách đơn giản, <i>clear</i> sử dụng để giải quyết vấn đề trong mục chú ý của thuộc tính <i>float</i> phía trên.</p> <p>Thuộc tính <i>float</i> có thể có một trong các giá trị sau:</p> <ul style="list-style-type: none"> • 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. <p>Cách phổ biến nhất để sử dụng thuộc tính <i>Clear</i> là dùng sau khi bạn đã sử dụng thuộc tính <i>Float</i> trên một phần tử. Nếu một phần tử được float sang bên trái, thì bạn nên clear bên trái. Phần tử <i>Float</i> của bạn sẽ tiếp tục nổi, nhưng phần tử bị xóa sẽ xuất hiện bên dưới nó trên trang web.</p>

32	Thuộc tính position dùng để làm gì? Phân biệt các giá trị: static, absolute, relative và fixed?
	<p>- Position là thuộc tính của CSS để xác định cách thức hiện thị vị trí của thẻ HTML</p> <p>+ static: mặc định của trang web</p> <p>+ relative: vị trí tương đối: lấy chính bản thân làm gốc tọa độ</p> <p>+ absolute: vị trí tuyệt đối: lấy thẳng cha bao bọc gần nó nhất có thuộc tính position làm gốc tọa độ</p> <p>+ fixed: lấy trình duyệt làm gốc tọa độ</p>
33	Responsive Web Design là gì?
	<p>Là kỹ thuật giúp trang web tương thích với nhiều thiết bị khác nhau mà không bị vỡ layout</p>

34	Làm thế nào để tạo 1 trang web responsive?
	<ol style="list-style-type: none"> Khai báo meta viewport <ul style="list-style-type: none"> meta viewport nghĩa là một thẻ thiết lập cho trình duyệt hiển thị tương ứng với kích thước màn hình. dựa vào chiều rộng của thiết bị (device-width) và không cho phép người dùng phóng to (thiết lập initial-scale với giá trị cố định là 1). Đây là thẻ mình khuyến khích bạn sử dụng cho toàn bộ các dự án Responsive. Sử dụng grid view (để chia dạng lưới 12 cột) + media querise: <ul style="list-style-type: none"> Media Queries được xây dựng như một cách mở rộng các đối tượng. Nó giúp chúng ta đặt các kiểu style khác nhau cho đối tượng trên các trình duyệt và tỷ lệ khung hình khác nhau.
35	Bootstrap là gì?
	<p>Bootstrap là một front-end framework miễn phí để phát triển web nhanh hơn và dễ dàng hơn.</p> <p>Bootstrap bao gồm các mẫu thiết kế dựa trên html và css.</p> <p>Bootstrap cung cấp khả năng tạo các thiết kế responsive.</p>
36	Một số thành phần thông dụng ở Bootstrap?
	<ul style="list-style-type: none"> Navbar: thường dùng để làm header Dropdown menu: thường dùng để làm menu tùy chọn Carousel: dùng để hiển thị dạng slide show

	<ul style="list-style-type: none">- Card: thường dùng cho chức năng danh sách.- Container/container-fluid: dùng để thiết kế phần thân body chứa thông tin- ...	
37	Phân biệt Static web và Dynamic web?	
	WEB TĨNH	
Về chức năng	Không thể thay đổi hay chỉnh sửa phần nội dung khi đã đăng lên website.	Có thể quản lý và chỉnh sửa nội dung bên trong website
Về ngôn ngữ lập trình	Sử dụng duy nhất ngôn ngữ HTML.	Đa dạng ngôn ngữ như ASP.NET, PHP kết hợp với cơ sở dữ liệu SQL
Về khả năng tương tác với người dùng	Người dùng không thể thực hiện các thao tác trên website như trò chuyện hay gửi bất kỳ phản hồi nào lên nó.	Khách hàng có thể thoải mái trao đổi thông tin với chủ sở hữu website khác bằng những thao tác cực kỳ đơn giản.
Về việc ứng dụng	Thường được dùng cho những trang web nhỏ, có phần nội dung nhất định, ít bị thay đổi.	Được ứng dụng trong các website phục vụ cho lĩnh vực tin tức, blog, doanh nghiệp hay các website sàn thương mại điện tử... Nói chung, ứng dụng cao hơn web tĩnh.
Về chi phí bảo trì - nâng cấp	<p>Chỉ cần bỏ vốn đầu tư một lần mà sử dụng được lâu dài.</p> <p>Web tĩnh có giao diện thân thiện với các công cụ tìm kiếm hơn web động.</p> <p>Tuy nhiên, việc bảo trì - nâng cấp và quản lý nội dung đối với <u>web tĩnh</u> tương đối phức tạp, khó thực hiện.</p>	<p>Tuy chi phí đầu tư ban đầu cho <u>web động</u> khá lớn, nhưng quá trình cùng đơn giản và dễ thực hiện.</p> <p>Thích hợp đối với các doanh nghiệp lớn</p>
38	Phân biệt 2 phương thức GET/POST	

- GET
 - + Hiện thị trên URL, lưu lại lịch sử trong trình duyệt
 - + Giới hạn dung lượng
 - + Gửi được các kí tự ASCII
 - + Ít bảo mật
- POST
 - + Không hiện thị trên URL, không lưu lại lịch sử
 - + Không giới hạn dung lượng
 - + Không giới

39	Phân biệt request/response
----	----------------------------

- + Request là yêu cầu từ client lên server
- + Response là phản hồi từ server về cho phía người dùng

40	Phân biệt forward/redirect
----	----------------------------

FORWARD	REDIRECT
Forward là chuyển tiếp từ webserver sang trang khác kèm theo dữ liệu	Redirect là chuyển hướng sang trang khác(không kèm theo dữ liệu)
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). 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.	Khi một yêu cầu (request) từ phía người dùng tới một Servlet (Trang A), Servlet chuyển yêu cầu này tới một trang khác (Trang B), và kết thúc nhiệm vụ. Khi người dùng chuyển hướng tới có thể là trang trong ứng dụng của bạn, hoặc có thể là trang ngoài. Địa chỉ trên trình duyệt của người dùng lúc này sẽ hiển thị đường dẫn của trang B.

41	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 ?
----	---

JSP là viết tắt của **JavaServer Pages** là một công nghệ để phát triển các trang web động. JSP giúp các nhà phát triển

	SERVLET	JSP
GIỐNG	Đều là công nghệ để phát triển các trang web động	
KHÁC	Được xây dựng dựa trên ngôn ngữ java	Được xây dựng dựa trên html.

	Viết mã Servlet khó hơn JSP vì đó là HTML trong Java.	JSP rất dễ viết mã vì đó là Java trong HTML.
	Trong mô hình MVC, Servlet sẽ xử lý phần controller.	JSP xử lý phần view.
		JSP chậm hơn Servlet do bước đầu tiên JSP thực hiện dịch JSP sang mã Java và biên dịch.
	Servlet có thể ghi đè phương thức service (),	JSP không thể ghi đè phương thức service

42

JSTL là gì? Giới thiệu một số thẻ bạn đã sử dụng ?

JSP Standard Tag Library (JSTL) là một bộ thư viện thẻ được xây dựng dựa trên ngôn ngữ Java.

Vì nếu viết như hiện tại, file JSP chứa cùng lúc 2 thành phần là HTML và Java, sẽ gây khó khăn cho việc phát triển => JSTL ra đời để hỗ trợ và khắc phục hạn chế này.

Một số thẻ JSTL thông dụng:

Thẻ	Mô tả
<c:out>	Để viết một cái gì đó trong trang JSP, bạn có thể sử dụng EL cũng có thẻ này
<c:import>	Giống với <jsp:include> hoặc chỉ thị include (include directive)
<c:redirect>	Chuyển hướng (redirect) yêu cầu tới một nguồn dữ liệu khác.
<c:set>	Sét đặt giá trị biến cho bởi phạm vi.
<c:remove>	Loại bỏ biến ra khỏi phạm vi đã cho.
<c:catch>	Bắt ngoại lệ và gói vào một đối tượng.
<c:if>	Điều kiện logic đơn giản, sử dụng với EL và bạn có thể sử dụng nó để xử lý các trường hợp ngoại lệ
<c:choose>	Tag có điều kiện đơn giản mà thiết lập một bối cảnh cho các trường hợp loại trừ lẫn nhau có điều kiện bằng <c:when> và <c:otherwise>
<c:when>	Thẻ con của <c:choose>, khi một điều kiện tại when là đúng.
<c:otherwise>	Thẻ con của <c:choose>, khi tất cả các điều kiện <c:when> là sai.
<c:forEach>	Dùng để lặp trên một tập hợp.

<c:forTokens>	dùng để lặp trên chuỗi (tokens) được phân cách bởi một dấu phân cách.
<c:param>	Được sử dụng với <c:import> để truyền các tham số.
<c:url>	Để tạo một URL với các tham số tùy chọn (optional query string parameters).

43	Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC?
Mô hình MVC là kiến trúc phần mềm phân thành 3 tầng: <ul style="list-style-type: none"> - MODEL: là nơi xử lý nghiệp vụ, biểu diễn dữ liệu và tương tác với database (nếu có). - VIEW: là nơi hiển thị dữ liệu và là nơi tương tác với người dùng - CONTROLLER: là nơi tiếp nhận các request, xác định thành phần nào sẽ nhận nhiệm vụ xử lý yêu cầu nào và điều khiển luồng đi của chương trình. 	

44	Sự khác nhau giữa MVC1 và MVC2
MVC1	MVC2
1. Trong MVC model 1: chỉ có 2 thành phần chính là trang JSP và file Bean, trong đó JSP đảm nhiệm cả View và Controller, file java Bean đảm nhiệm phần model. Cấu trúc: <ol style="list-style-type: none"> 1. Ví dụ về login, Khi A gửi request login từ form 2. Một trang jsp sẽ nhận được request 3. Trang Jsp sẽ sử dụng tài nguyên trong bean 4. Khi nhận được kết quả sẽ điều hướng đến trang JSP khác tùy theo kết quả nhận được 5. Hiển thị kết quả nhận được trong trong JSP tương ứng 	1. Trong MVC model 2 thì có thêm sự xuất hiện của servlet thay thế JSP trở thành controller. Cấu trúc: <ol style="list-style-type: none"> 1. A gửi request từ máy client đến server, controller sẽ nhận được request. 2. Servlet nhận dữ liệu từ request, gọi các tài nguyên từ Bean(Model). 3. Bean sẽ xử lí dữ liệu, gửi trả kết quả cho servlet 4. Servlet nhận kết quả trả về từ Model, sẽ điều hướng về view tương ứng với kết quả nhận được 5. Hiển thị trên view và trả lại máy client
45	Mô tả cách thực hiện tạo và lấy kết quả 1 câu query bất kỳ bằng JDBC?
1. Nạp driver 2. Định nghĩa connection URL 3. Kết nối cơ sở dữ liệu bằng đối tượng connection đã tạo (=DriverManager.getConnection(link database, u	

4. Tạo đối tượng statement
5. Thi hành câu truy vấn, nếu là câu SELECT (executeQuery) thì dùng ResultSet để hứng kết quả
6. Xử lý kết quả
7. Đóng kết nối

46 Phân biệt các loại Statement ?

Statement: dùng truy vấn câu lệnh tĩnh, không bảo mật, có thể bị tấn công Injection, hiệu suất thấp

PreparedStatement: Dùng để tạo câu lệnh truy vấn động, truyền tham số thông qua (?), hiệu suất cao

Callable Statement: Thường dùng để gọi thủ tục, hiệu suất cao

47 Kỹ thuật sử dụng transaction bằng JDBC là như thế nào?

Là tạo ra một tiến trình xử lý có điểm đầu và điểm cuối, được chia ra thành nhiều hoạt động. Khi một tiến trình được thực hiện thì các hoạt động sẽ được thực hiện một cách tuần tự và độc lập.

48 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, ...

49. Phương pháp gọi 1 SP ở JDBC?

B1: Tạo CallableStatement

- + Đối với procedure không có tham số: {call procedure_name}
- + Đối với procedure có tham số: {call procedure_name(?, ?, ?...)}
- + Đối với procedure có giá trị trả về: {?= call procedure_name(?, ?, ?...)}

50. Phân biệt Ant, Maven, Gradle?

- Giữa Gradle vs Maven có một số điểm khác biệt cơ bản trong cách tiếp cận build một dự án. Gradle dựa vào đồ thị phụ thuộc task. Trong khi đó Maven thì lại dựa trên mô hình cố định và tuyến tính của các giai đoạn
- Ant không có quản lý phụ thuộc còn maven vs gradle thì có