

1	<p>Cơ sở dữ liệu (database) là gì ?</p> <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	<p>Phân biệt các câu lệnh DML & DDL? DML, DDL bao gồm các câu lệnh nào?</p> <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	<p>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> <ul style="list-style-type: none"> - ERD là một sơ đồ thể hiện các thực thể có trong database và mối quan hệ giữa chúng với nhau <p>Thành phần cơ bản của ERD</p> <p>Để xây dựng sơ đồ ERD cần phải nắm được các khái niệm sau:</p> <ul style="list-style-type: none"> • Entity (Thực thể) • Relationship (Mối quan hệ) • Property/ Attributes (Thuộc tính) • Entity Set (Tập thực thể) • Relationship Set (Tập mối quan hệ)
4	<p>Các bước chuyển đổi từ ERD -> Table?</p> <ul style="list-style-type: none"> • Biến đổi thực thể → table <ul style="list-style-type: none"> ○ Thuộc tính khoá của thực thể là khoá chính của bảng ○ Thuộc tính thường có của thực thể là thuộc tính thường của bảng • Biến đổi thuộc tính đa trị <ul style="list-style-type: none"> ○ Tách thuộc tính đa trị ra bảng mới ○ Đặt khoá ngoại vào bảng hiện tại • Biến đổi 1-1 <ul style="list-style-type: none"> ○ Đặt thuộc tính khoá ở phía tùy chọn → phía bắt buộc • Biến đổi 1-n

	<ul style="list-style-type: none"> ○ Đặt thuộc tính khoá ở 1 → n • Biến đổi n-n <ul style="list-style-type: none"> ○ Tạo ra một bảng mới có khoá chính là khoá phức hợp 2 khoá chính của 2 thực thể tham gia vào mqh → khoá composite 				
5	<p>Các bước chuẩn hóa dữ liệu?</p> <p>Có 3 dạng chuẩn hóa:</p> <p>+ Dạng chuẩn hóa 1NF: các thuộc tính phải là đơn trị, tách các thuộc tính đa trị thành bảng mới</p> <p>+ Dạng chuẩn 2NF: đạt dạng chuẩn 1NF và tất cả các thuộc tính không khoá của Q đều phụ thuộc đầy đủ vào mọi khoá</p> <p>+ Dạng chuẩn 3NF: đạt chuẩn 2NF và tất cả các thuộc tính không khoá phải phụ thuộc bắc cầu vào thuộc tính khoá</p>				
6	<p>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: <ul style="list-style-type: none"> + Not null: không cho phép giá trị null + Unique: mỗi giá trị là duy nhất + Primary key: khoá chính (not null + unique) + Foreign key: khoá 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	<p>Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?</p> <ul style="list-style-type: none"> - Khóa chính là một cột hay nhiều cột được sử dụng để xác định một bản ghi duy nhất trong bảng. 				
8	<p>Khóa ngoại (foreign key) là gì?</p> <ul style="list-style-type: none"> - Khóa ngoại là cơ chế tạo liên kết giữa 2 bảng trong cùng CSDL - Khóa ngoại được đặt trên 1 cột của bảng này và tham chiếu đến khóa chính của bảng khác - Kiểu dữ liệu khóa ngoại và khóa chính được tham chiếu đến phải giống nhau 				
9	<p>Phân biệt DELETE và TRUNCATE?</p> <table border="1"> <thead> <tr> <th>DELETE</th><th>TRUNCATE</th></tr> </thead> <tbody> <tr> <td>○ Delete có thể xoá từng dòng hoặc xoá cả</td><td>○ Truncate chỉ có thể xoá hết bảng và reset</td></tr> </tbody> </table>	DELETE	TRUNCATE	○ Delete có thể xoá từng dòng hoặc xoá cả	○ Truncate chỉ có thể xoá hết bảng và reset
DELETE	TRUNCATE				
○ Delete có thể xoá từng dòng hoặc xoá cả	○ Truncate chỉ có thể xoá hết bảng và reset				

		bảng, khi xoá thì SQL sẽ log lại từng dòng đã xoá 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 <ul style="list-style-type: none"> ○ Xoá bằng delete thì sẽ restored được ○ Chậm hơn truncate 	transaction log, vì thế khi tạo một record mới thì giá trị id sẽ bắt đầu từ 1 <ul style="list-style-type: none"> ○ Xoá bằng truncate thì không restored được ○ Xoá 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"> - Câu lệnh JOIN dùng để truy vấn dữ liệu kết hợp từ nhiều bảng có mối liên hệ với nhau - 4 loại join : <ul style="list-style-type: none"> + INNER JOIN: lấy phần chung của 2 bảng + LEFT JOIN: trả về tất cả các record của bảng bên trái và những record tương ứng với bảng bên phải + LEFT JOIN: trả về tất cả các record của bảng bên phải và những record tương ứng với bảng bên trái + FULL JOIN: tất cả các record ở 2 bảng 		
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? <ul style="list-style-type: none"> - Toán tử IN sử dụng khi muốn kiểm tra một giá trị có xuất trong tập giá trị của truy vấn con hay không? <ul style="list-style-type: none"> + Truy vấn con sẽ thực hiện đầu tiên và thực hiện 1 lần. Sau đó dùng làm điều kiện để kiểm tra biểu thức bên ngoài - Toán tử EXIST để kiểm tra xem một truy vấn con có trả về kết quả nào không, trả về true nếu có ít nhất 1 dòng và trả về false nếu không có dòng nào <ul style="list-style-type: none"> + Hàng đầu tiên từ truy vấn bên ngoài sẽ được chọn, sau đó truy vấn con mới được thực hiện. Dừng việc kiểm tra hàng đó nếu kết quả được tìm thấy. 		

	+ Truy vấn con thực hiện nhiều lần nếu truy vấn bên ngoài có nhiều record
14	Phân biệt AND và OR? - Biểu thức chứa toán tử and trả về true nếu tất cả điều kiện là đúng và trả về false nếu có ít nhất 1 điều kiện là sai - 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
15	Phân biệt GROUP BY và ORDER BY? - GROUP BY: dùng để nhóm các tập kết quả dựa theo giá trị của một cột hoặc nhiều cột - ORDER BY: dùng để sắp xếp dữ liệu theo tăng dần hoặc giảm dần theo cột chỉ định theo trật tự
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? SELECT id, name, AVG(Diem) as Diem FROM SINHVIEN GROUPBY id, name;
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? SELECT id, name, AVG(Diem) as Diem FROM SINHVIEN GROUPBY id, name; HAVING AVG(Diem) > 5;
20	Hiển thị danh sách sinh viên và lớp học tương ứng thì làm thế nào (hai bảng Student và Class có cột chung ClassID)? SELECT studentId, studentName, className FROM STUDENT s JOIN CLASS c ON s.classId = c.classId;
21	Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi? SELECT id, name, diem FROM SINHVIEN

	ORDER BY diem;
22	<p>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?</p> <pre>SELECT id, name, age FROM SINHVIEN ORDER BY name DESC, age;</pre>
23	<p>Hiển thị các sinh viên có họ là 'Le' và tên là 'Hai'?</p> <pre>SELECT id, ho, ten FROM SINHVIEN WHERE ho = "Le" and ten = "Hai"</pre>
24	<p>Hiển thị các sinh viên có tên bắt đầu là 'C'?</p> <pre>SELECT id, name FROM SINHVIEN Where name like "C%"</pre>
25	<p>Làm sao để có thể tăng tốc độ truy vấn? Nhược điểm là gì?</p> <p>Dùng INDEX để tăng tốc độ truy vấn</p> <p>Nhược điểm: thêm, xóa, sửa dữ liệu bị chậm và chiếm dụng bộ nhớ</p>
26	<p>Các dạng tham số ở SP?</p> <p>IN: lấy tham số đầu vào và không bị thay đổi bên ngoài</p> <p>OUT: tham số nhận vào giá trị null mặc dù khi ta truyền có giá trị, nếu bên trong procedure làm thay đổi biến đó thì bên ngoài cũng thay đổi</p> <p>INOUT: Kết hợp IN và OUT, nhận tham số đầu vào và nếu bên trong procedure làm thay đổi biến đó thì bên ngoài cũng thay đổi</p>
27	<p>Phân biệt SP, Trigger, Function?</p> <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

	<p>+ Thực hiện tự động khi có lệnh update, insert hoặc delete</p> <table> <tr> <th>Stored Procedure</th><th>Hàm</th></tr> <tr> <td>Được sử dụng để thực hiện một trình logic nào đó</td><td>Được sử dụng để thực hiện phép tính toán</td></tr> <tr> <td>Không bắt buộc phải có kiểu trả về</td><td>Phải có kiểu trả về</td></tr> <tr> <td>Có thể trả về 0 hoặc nhiều giá trị</td><td>Có thể trả về chỉ một giá trị</td></tr> <tr> <td>Chúng ta có thể gọi hàm từ procedure</td><td>Procedure không thể được gọi từ hàm</td></tr> <tr> <td>Procedure hỗ trợ các tham số input và output</td><td>Hàm chỉ hỗ trợ các tham số input</td></tr> <tr> <td>Xử lý ngoại lệ với khối try/catch có thể được sử dụng trong procedure</td><td>Xử lý ngoại lệ với try/catch không thể được sử dụng trong các hàm do người dùng tự định nghĩa</td></tr> </table>	Stored Procedure	Hàm	Được sử dụng để thực hiện một trình logic nào đó	Được sử dụng để thực hiện phép tính toán	Không bắt buộc phải có kiểu trả về	Phải có kiểu trả về	Có thể trả về 0 hoặc nhiều giá trị	Có thể trả về chỉ một giá trị	Chúng ta có thể gọi hàm từ procedure	Procedure không thể được gọi từ hàm	Procedure hỗ trợ các tham số input và output	Hàm chỉ hỗ trợ các tham số input	Xử lý ngoại lệ với khối try/catch có thể được sử dụng trong procedure	Xử lý ngoại lệ với try/catch không thể được sử dụng trong các hàm do người dùng tự định nghĩa
Stored Procedure	Hàm														
Được sử dụng để thực hiện một trình logic nào đó	Được sử dụng để thực hiện phép tính toán														
Không bắt buộc phải có kiểu trả về	Phải có kiểu trả về														
Có thể trả về 0 hoặc nhiều giá trị	Có thể trả về chỉ một giá trị														
Chúng ta có thể gọi hàm từ procedure	Procedure không thể được gọi từ hàm														
Procedure hỗ trợ các tham số input và output	Hàm chỉ hỗ trợ các tham số input														
Xử lý ngoại lệ với khối try/catch có thể được sử dụng trong procedure	Xử lý ngoại lệ với try/catch không thể được sử dụng trong các hàm do người dùng tự định nghĩa														
28	<p>CSS là gì? Các cách nhúng CSS vào tài liệu html? Độ ưu tiên?</p> <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 đề lại) 														
29	<p>Phân biệt giữa bộ chọn class và bộ chọn id?</p> <p>+ Bộ chọn id: chọn 1 phần tử vs id xác định, dùng dấu #</p> <p>+ Bộ chọn class: chọn 1 hoặc nhiều phần tử, dùng dấu .</p>														
30	<p>Trình bày mô hình hộp (box model) trong CSS?</p> <p>Gồm 4 phần:</p> <ul style="list-style-type: none"> - Content: nội dung của box - Padding: khoảng cách đệm từ content đến border - Border: khoảng cách đường viền của box - Margin: khoảng cách từ border của box này đến border của box khác 														
31	<p>Trình bày thuộc tính float, clear?</p> <ul style="list-style-type: none"> - Float: dùng để định vị và định dạng nội dung <p>Có thể nhận các giá trị:</p> <ul style="list-style-type: none"> + left: hiển thị ở bên trái vùng chứa nó + right: hiển thị bên phải vùng chứa nó + none: mặc định + inherit: kế thừa giá trị float của phần tử cha - Clear: dùng để chỉ định những phần tử nào có thể trôi nổi bên cạnh phần tử đã xóa và ở phía nào + none: cho phép hiển thị lên cả hai bên trái phải + left: không cho phép hiển thị bên trái + right: không cho phép hiển thị bên phải 														

	<ul style="list-style-type: none"> + both: không cho phép hiển thị hai bên + inherit: kế thừa giá trị clear của phần tử gốc
32	<p>Thuộc tính position dùng để làm gì?</p> <ul style="list-style-type: none"> - Thuộc tính position dùng để định vị vị trí hiển thị của các phần tử HTML <p>Phân biệt các giá trị: static, absolute, relative và fixed?</p> <ul style="list-style-type: none"> + static: mặc định của trang web + relative: vị trí tương đối: lấy chính bản thân làm gốc tọa độ + absolute: vị trí tuyệt đối: lấy thẳng cha bao bọc gần nó nhất có thuộc tính position làm gốc tọa độ + fixed: lấy trình duyệt làm gốc tọa độ
33	<p>Responsive Web Design là gì?</p> <p>Là kĩ thuật giúp trang web tương thích với nhiều thiết bị khác nhau mà không bị vỡ layout</p>
34	<p>Làm thế nào để tạo 1 trang web responsive?</p> <ul style="list-style-type: none"> + Dùng media queries để chỉnh sửa css cho thiết bị khác nhau + Dùng grid view: 12 cột
35	<p>Bootstrap là gì?</p> <p>Bootstrap là một thư viện front-end framework miễn phí để phát triển trang web, gồm các mẫu thiết kế dựa trên html và css và công cụ để thiết kế responsive</p>
36	<p>Một số thành phần thông dụng ở Bootstrap?</p> <p>Hệ thống grid của Bootstrap có 4 class</p> <ul style="list-style-type: none"> • xs – điện thoại • sm – máy tính bảng • md – máy tính để bàn • lg - máy tính để bàn lớn hơn <p>Tự liệt kê: ví dụ container, nav, table, form, list-group,...</p>
37	<p>Phân biệt Static web và Dynamic web?</p> <ul style="list-style-type: none"> - Static web: web tĩnh với nội dung cố định, khi muốn sửa nội dung thì cần phải sửa code. Sử dụng HTML, CSS và có thể dùng JS để tạo hiệu ứng - Dynamic web: web động với nội dung thay đổi theo tương tác của người dùng. Sử dụng HTML, CSS, JS và ngôn ngữ phía server để nội dung trang web được thay đổi phù hợp
38	Phân biệt 2 phương thức GET/POST

	<ul style="list-style-type: none"> - GET <ul style="list-style-type: none"> + 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 <ul style="list-style-type: none"> + Không hiện thị trên URL, không lưu lại lịch sử + Không giới hạn dung lượng + Không giới hạn kí tự, có thể gửi được nhị phân,... + Bảo mật
39	Phân biệt request/response <ul style="list-style-type: none"> + Request là yêu cầu từ client lên server + Response là phản hồi từ server về cho phía người dùng
40	Phân biệt forward/redirect <ul style="list-style-type: none"> + forward: chuyển tiếp: chuyển trang và truyền dữ liệu sang + redirect : chuyển hướng: chỉ chuyển trang và không truyền dữ liệu sang được
41	<p>JSP là gì? Nói: “Về bản chất JSP cũng chính là Servlet” đúng hay sai? Giải thích?</p> <ul style="list-style-type: none"> - 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. <p>So sánh giống và khác nhau giữa JSP và Servlet?</p> <p>Giống nhau</p> <ul style="list-style-type: none"> - 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. <p>Khác nhau</p> <ul style="list-style-type: none"> - Nếu như Servlet mạnh về xử lý nghiệp vụ, còn JSP mạnh về giao diện. - Ở Servlet, mã HTML nằm trong mã Java. Còn ở JSP, mã Java nằm trong mã HTML.
42	<p>JSTL là gì? Giới thiệu một số thẻ bạn đã sử dụng?</p> <ul style="list-style-type: none"> - 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:redirect: chuyển hướng

	<ul style="list-style-type: none"> - c:out - c:if: kiểm tra điều kiện - c:forEach - c:choose: when - c:url 								
43	<p>Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC?</p> <ul style="list-style-type: none"> - MVC là mô hình kiến trúc phần mềm chia thành 3 tầng + Model: xử lý nghiệp vụ + View: xử lý giao diện + tương tác người dùng + C: controller: nhận request và xử lý request, kết nối giữa view và model 								
44	<p>Sự khác nhau giữa MVC1 và MVC2</p> <p><small>Sự khác, giống nhau, ưu, nhược điểm của MVC1 & MVC2.</small></p> <ul style="list-style-type: none"> • Trong MVC1, controller và view đều là jsp. MVC1 là thể 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 xử 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. 								
45	<p>Mô tả cách thực hiện tạo và lấy kết quả 1 câu query bất kỳ bằng JDBC?</p> <p>B1: Kết nối vs database</p> <p>B2: Tạo đối tượng statement</p> <p>B3: Thực thi câu lệnh SQL</p> <ul style="list-style-type: none"> - Với câu lệnh select thì executeQuery => trả về đối tượng Resultset - Với insert, update, delete thì executeUpdate => trả về số record bị ảnh hưởng 								
46	<p>Phân biệt các loại Statement?</p> <table border="1"> <thead> <tr> <th>JDBC Statements</th><th>Mục đích sử dụng</th></tr> </thead> <tbody> <tr> <td>Statement</td><td>Hữu ích khi bạn đang sử dụng câu lệnh SQL tĩnh khi chạy. Statement không chấp nhận tham số (parameters).</td></tr> <tr> <td>PreparedStatement</td><td>PreparedStatement cho phép chỉ định tham số đầu vào khi chạy.</td></tr> <tr> <td>CallableStatement</td><td>Sử dụng khi bạn muốn truy cập stored procedures. CallableStatement cũng chấp nhận tham số đầu vào khi chạy.</td></tr> </tbody> </table>	JDBC Statements	Mục đích sử dụng	Statement	Hữu ích khi bạn đang sử dụng câu lệnh SQL tĩnh khi chạy. Statement không chấp nhận tham số (parameters).	PreparedStatement	PreparedStatement cho phép chỉ định tham số đầu vào khi chạy.	CallableStatement	Sử dụng khi bạn muốn truy cập stored procedures. CallableStatement cũng chấp nhận tham số đầu vào khi chạy.
JDBC Statements	Mục đích sử dụng								
Statement	Hữu ích khi bạn đang sử dụng câu lệnh SQL tĩnh khi chạy. Statement không chấp nhận tham số (parameters).								
PreparedStatement	PreparedStatement cho phép chỉ định tham số đầu vào khi chạy.								
CallableStatement	Sử dụng khi bạn muốn truy cập stored procedures. CallableStatement cũng chấp nhận tham số đầu vào khi chạy.								
47	<p>Kỹ thuật sử dụng transaction bằng JDBC là như thế nào?</p>								

	<p>Để thực hiện quản lý Transaction cho riêng mình, đầu tiên bạn cần tắt chế độ auto-commit bởi truyền tham số false cho phương thức setAutoCommit().</p> <p>Sau đó, khi bạn đã thực hiện các thay đổi với cơ sở dữ liệu và muốn ký thác các thay đổi đó, bạn sử dụng phương thức commit(). Hoặc bạn muốn 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, bạn sử dụng phương thức rollback().</p>
48	<p>Các tính chất của Transaction?</p> <ul style="list-style-type: none"> • 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	<p>Phương pháp gọi 1 SP ở JDBC?</p> <p>B1: Tạo CallableStatement</p> <p>+ Đối với procedure không có tham số: {call procedure_name}</p> <p>+ Đối với procedure có tham số: {call procedure_name(?, ?, ?...)} + Đối với procedure có giá trị trả về: {?= call procedure_name(?, ?, ?...)}</p>
50	<p>Phân biệt Ant, Maven, Gradle?</p> <ul style="list-style-type: none"> - 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