Soạn Audit module 3

# Phân biệt char vả varchar trong SQL:

Char: sử dụng thực thể và không gian thừa để lưu trữ

Varchar: chỉ dùng độ dài thực thể để lưu 🡺 tiết kiệm bộ nhớ hơn

# Câu 1: Cơ sở dữ liệu (database) là gì?

Là thành phần nằm cuối cùng của backend, sẽ lưu trữ trữ dữ liệu của 1 trang web hay 1 ứng dụng nào đó. Dữ liệu sẽ lưu trữ có cấu trúc , rõ ràng, có thể truy xuất:

## Ưu điểm:

* Tính chia sẽ cao, phục vụ cho nhiều người ở nhiều nơi khác nhau
* Bảo mật cao
* Giảm trùng lặp thông tin, đảm bảo tính nhất quán và toàn vẹn dữ liệu
* Thuận tiện cho việc thao tác với DL, truy cập theo nhiều cách khác nhau, từ nhiều người khác nhau và trên các ứng dụng khác nhau

## Nhược điểm:

* Tốn phí
* Phải học, phải có kiến thức về DB
* Đảm bảo tính bảo mật, chủ quyền DL: Yêu cầu lập trình viên có tính cẩn thận cao để tránh rò rỉ DL khi thiết kế DB
* Khi gặp sự cố phải đảm bảo Dl được an toàn

# Câu 2: Phân biệt các câu lệnh DML & DDL? Và DML & DDL bao gồm các câu lệnh nào?

Trong MySQL có 4 nhóm câu lệnh:

1. DDL:Data definition language: các câu lệnh được use để tạo DB, table,…thao tác với CSDL (tạo ra và thao tác chỉnh sửa hoặc xóa..các đối tượng như database, table)
2. DML:Data manipulation language: các câu lệnh thao tác với dữ liệu của bảng
3. DCL: Data control language : các câu lệnh phân quyền: GRAND/REVOKE
4. TCL: Transaction control language: nhóm lệnh quản lý các tiến trình: setAutoComit, Commit, rollback, savepoint…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **C** | **R** | **U** | **D** |
| **DDL** | **Create** |  | **Alter** | **Drop/Trancate** |
| **DML** | **Insert into** | **Select** | **Update** | **Delete** |

# Câu 3: Trình bày về mô hình thực thể mối quan hệ (ERD Entire-Relationship-Diagram)? Trong mô hình thực thể mối quan hệ có những thành phần nào?

Là sơ đồ mô hình hóa thực thể và mối quan hệ giữa chúng

Mô hình thực thể mối quan hệ (ERD - Entity-Relationship Diagram) là một công cụ hữu ích trong thiết kế cơ sở dữ liệu để mô tả cấu trúc của dữ liệu và mối quan hệ giữa các thực thể. ERD sử dụng các biểu đồ để minh họa các thực thể (entities), mối quan hệ giữa chúng và các thuộc tính (attributes) của thực thể.

Gồm có các thành phần:

* Entity (Thực thể ) và Entity set (tập thực thể)
* Relationship (mối quan hệ) và relationship set (tập quan hệ)
* Property/attributes (thuộc tính)

Các bước chuyển SRS sáng ERD:

1. Xác định thực thể
2. Xác định thuộc tính của thực thể
3. Xác định mối quan hệ 1-1, 1-n, n-n
4. Vẽ

Các hình biểu diễn trong ERD:

* HCN: thực thể
* Elip: thuộc tính
* 2 Elip lồng nhau: thuộc tính đa trị
* Elip thuộc tính có gạch chân: thuộc tính khóa chính
* Hình thoi: thể hiện mqh
* Line: đường kết nối giữa các hình

# Câu 4: Các bước chuyển đổi từ ERD -> Table?

1. Chuyển đổi thực thể thành table
2. Chuyển đổi các thuộc tính của thực thể thành các thuộc tính của table. Bao gồm xác định các thuộc tính khóa và tách các thuộc tính đa trị
3. Chuyển đổi mqh 1-1: khóa chính bên tùy chọn làm khóa ngoại bên bắt buộc
4. Chuyển đổi mqh 1-n: khóa chính bên 1 làm khóa phụ bên nhiều
5. Chuyển đổi mqh n-n: tách thành 1 bảng riêng có khóa chính là cặp khóa chính của 2 bảng còn lại, khóa này còn gọi là khóa phức hợp

# Câu 5: Các bước chuẩn hóa dữ liệu?

Bảng các thuộc tính🡺 Chuẩn 1 NF 🡺 Chuẩn 2 NF 🡺 Chuẩn 3 NF 🡺 Chuẩn Boyce codd

* Điều kiện:

1NF: Tách thuộc tính đa trị, loại bỏ các thuộc tính có thể suy ra được từ các thuộc tính khác

2NF: Đạt chuẩn 1 NF, loại bỏ các phụ thuộc bộ phận, tức là loại bỏ các thuộc tính không phải thuộc tính khóa và nó phụ thuộc vào 1 bộ phận nào đó của khóa primary key

3NF: Đạt chuẩn 2NF, Loại bỏ các phụ thuộc bắc cầu

# Câu 6: Rằng buộc (constraint) là gì? Các loại rằng buộc trong MySQL?

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.

1. Not null
2. Default
3. Unique
4. Primary key
5. Forein key
6. Check

# Câu 7: Khóa chính (primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?

* Khóa chính được sử dụng để định danh duy nhất cho mỗi bản ghi trong bảng của CSDL
* Ngoài ra còn dùng để thiết lập mqh 1-n (khóa ngoại)
* Dữ liệu khóa chính phải là duy nhất và not null
* Mỗi bảng chỉ có một khóa chính
* Khóa chính có thể được tạo ra từ 1 cột hoặc nhiều cột (khóa chính tổ hợp) của bảng

# Câu 8: Khóa ngoại (foreign key) là gì?

* Khóa ngoại là là cơ chế để tạo liên kết giữa các bảng trong cùng một CSDL. Khóa ngoại được đặt trên một cột của bảng này và tham chiếu đến khóa chính của bảng khác. Kiểu dữ liệu của khóa ngoại và khóa chính phải giống nhau

# Câu 9: Phân biệt DELETE và TRUNCATE?

1. Drop: để xóa một đối tượng toàn bộ khỏi cơ sở dữ liệu, chẳng hạn như một bảng, một cơ sở dữ liệu, hoặc một chỉ mục.Khi bạn sử dụng DROP, mọi dữ liệu và cấu trúc của đối tượng đó đều bị xóa và không thể khôi phục được.
2. Truncate: được sử dụng để xóa toàn bộ dữ liệu từ một bảng, nhưng bảng vẫn tồn tại và cấu trúc của nó không thay đổi.TRUNCATE thường nhanh hơn so với DELETE, vì nó không ghi vào log transaction và không tạo ra bản ghi transaction cho mỗi dòng được xóa.
3. được sử dụng để xóa dữ liệu từ một bảng dựa trên một điều kiện cụ thể. Delete ghi vào log transaction và tạo bản ghi transaction cho mỗi dòng bị xóa, điều này có thể ảnh hưởng đến hiệu suất so với TRUNCATE.Bạn có thể chỉ xóa một phần của dữ liệu bằng cách sử dụng điều kiện trong câu lệnh

# Câu 10: các loại JOIN? Phân biệt?

* Có 2 loại join cơ bản:

1. inner join: nối bảng lấy điểm chung của 2 bảng

2. outer join: (left, right, full):

* Left join: chung và phần bên trái
* Right join: chung và phần bên phải
* Full join: k hỗ trợ với MySQL nhưng có thể triển khai

Ngoài ra còn có cross join: bảng a 3 hàng, bảng b 3 hàng 🡺 9 hàng

# Câu 11: phân biệt JOIN và UNION?

1. UNION:

* được sử dụng để kết hợp các kết quả của hai hoặc nhiều truy vấn SELECT thành một kết quả duy nhất.
* Các cột của các SELECT phải có cùng số lượng cột và phải có kiểu dữ liệu tương ứng.
* Các bản ghi trùng lặp sẽ bị loại bỏ trong kết quả cuối cùng.

1. Join:

* JOIN được sử dụng để kết hợp các hàng từ hai hoặc nhiều bảng dựa trên một điều kiện kết nối.
* Có nhiều loại JOIN như INNER JOIN, LEFT JOIN, RIGHT JOIN, và FULL JOIN, mỗi loại đều cung cấp kết quả khác nhau dựa trên sự khớp hoặc không khớp của dữ liệu trong các bảng.
* JOIN giúp kết hợp thông tin từ nhiều bảng liên quan để tạo ra một tập dữ liệu hoàn chỉnh.

# Câu 13: Phân biệt EXIST và IN?

IN:

* In:được sử dụng ở điều kiện của câu truy vấn con, khi sử dụng IN thì câu truy vấn con chỉ được trả về một cột duy nhất. Sử dụng để so sánh một giá trị với một danh sách các giá trị cụ thể hoặc kết quả của một truy vấn con.
* Trả về TRUE/FALSE.

EXIST:

* Exist: Sử dụng để kiểm tra sự tồn tại của bất kỳ bản ghi nào trong kết quả của một truy vấn con.
* Trả về giá trị TRUE/FALSE.
* Thường được sử dụng với các truy vấn con kết hợp với các điều kiện.
* Hiệu suất hơn IN

# Câu 14: phân biệt AND và OR trong SQL?

* AND: trả về TRUE chỉ khi cả tất cả các điều kiện đều đúng
* OR: trả về TRUE khi 1 trong các điều kiện là đúng

# Câu 12: phân biệt UNION và UNION ALL?

* Union: là gộp kết quả của 2 hay nhiêu câu select (gộp và loại bỏ những kq trùng nhau)
* Union All: gộp tất cả kể cả những kq trùng

# Câu 15: Phân biệt GROUP BY và ORDER BY?

* Group by dùng để nhóm dữ liệu dựa trên cùng một giá trị trong 1 cột cụ thể.
* Order by: dùng để sắp xếp kết quả và hiển thị nó theo một thứ tự tăng dần or giảm dần.

# Câu 17: Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?

select avg(mark) as averageall from mark;

# Câu 16: Muốn nối chuỗi trong SQL thì sử dụng gì?

* Sử dụng hàm concat() hoặc + hoặc concat\_ws()

# Câu 18: Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?

select s.studentName , avg(m.mark)

from student s join mark m on s.StudentId = m.StudentId

group by s.StudentId;

# Câu 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 s.studentName, avg(m.mark) as average

from student s join mark m on s.studentId = m.studentId

group by s.studentId

having average > 5;

# Câu 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 s.studentName, c.ClassName

from student s join class c

on s.classID = c.classID;

# Câu 21: Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi?

select s.studentName, avg(m.mark) as average

from student s join mark m on s.studentId = m.studentId

group by s.studentId

order by average;

# Câu 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?

select \* from student

order by studentName desc, StudentId asc;

# câu 23: Hiển thị các sinh viên có họ là ‘Le’ và tên là ‘Hai’?

select \* from student

where studentName like 'Le%' and studentName like '%hai';

# câu 24: Hiển thị các sinh viên có tên bắt đầu là ‘C’?

select \* from student

where substring\_index(StudentName, ' ', -1) like 'c%';

# câu 25: Làm sao để có thể tăng tốc độ truy vấn? Nhược điểm là gì?

1. Hạn chế sự kết nối đến hệ quản trị CSDL để lấy dữ liệu
2. Chỉ select những trường mà bạn cần ( chiếm ít bộ nhớ máy chủ hơn)
3. Sử dụng index để tăng tốc độ câu truy vấn
4. Sử dụng EXPLAIN để phân tích câu truy vấn
5. Truy vấn trên 1 bảng sẽ nhanh hơn nhiều so với việc truy vấn 1 lúc nhiều bảng để lấy ra 1 bản record🡺 hạn chế tối thiểu, chấp nhận hi sinh bộ nhớ
6. Thay vì insert 1 lúc 1 thì nên insert 1 lúc nhiều value cho 1 câu lệnh insert
7. Việc đếm xem có bao nhiêu dữ liệu trả về rất quan trọng, nó giúp bạn so sánh được số truy vấn tính toán và số truy vấn thực tế. Nhằm tìm ra nguyên nhân và tối ưu truy vấn của bạn
8. Chỉ dùng DISTINCT khi cần: máy chủ sẽ k phải sort và loại bỏ trùng lặp
9. Chỉ dùng UNION khi cần (dùng UNION ALL)
10. Sử dụng kiểu dữ liệu chính xác cho trường
11. Giải phóng bộ nhớ sau khi sử dụng xong
12. Áp dụng các thói quen code nhanh hơn
13. [Sử dụng view và Stored Procedure thay cho các câu lệnh truy vấn phức tạp](https://jobs.hybrid-technologies.vn/blog/toi-uu-hoa-truy-van-sql/#Su_dung_view_va_Stored_Procedure_thay_cho_cac_cau_lenh_truy_van_phuc_tap)

# Câu 26: Các dạng tham số ở SP?

Có 3 dạng truyền tham số trong stored procedure: IN, OUT, and INOUT

IN: Đây là chế độ mặc định (nghĩa là nếu bạn không định nghĩa loại nào thì nó sẽ hiểu là IN).

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.

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.

# Câu 27: phân biệt stored procedure, trigger and function.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SP** | **Trigger** | **Function** |
| **Kiểu trả về** | **void** | **void** | **Có return** |
| **Sử dụng** | **Call name\_sp** | **Tự động chạy khi có sự kiện nào đó xãy ra** | **Được gọi ở câu select** |
| **Tham số truyền vào** | **Có tham số truyền vào** | **Không có tham số truyền vào** | **Có tham số truyền vào** |

* Stored procedure: là tập hợp các câu query để thực hiện 1 nhiệm vụ nhất định
* Function: giống sp nhưng có giá trị trả về
* Trigger: là một đối tượng được định nghĩa trong CSDL dùng để gắn chặt với 1 sự kiện nhất định nào đó (select,insert,…) trên 1 bảng nào đó.

# Câu 28: CSS là gì? Các cách nhúng CSS vào tài liệu html? Độ ưu tiên?

CSS:

1. CSS là viết tắt của cụm từ Cascading Style Sheets
2. CSS mô tả cách các phần tử HTML sẽ hiển thị trên màn hình, trang web, hoặc các phương tiện khác.
3. 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.
4. Các mô tả css có thể được lưu trong tệp tin CSS.

* Có 3 cách nhúng css vào html

**Inline**: mã CSS viết tại thuộc tính style của phần tử HTML

**Internal**: mã CSS trong chính văn bản HTML, nằm trong khối thẻ <style>

**External**: mã CSS ở một file riêng biệt (thường là file.css) sau đó nạp vào HTML bằng phần tử html <link> (liên kết css với html)

* Ưu tiên inline
* Cái nào dưới thì dùng
* Id # 🡺 class . 🡺 universal \* 🡺 element/tag
* Dùng ! important sau khai báo để lấy độ ưu tiên

# Câu 29: Phân biệt giữa bộ chọn class và bộ chọn id?

* Bộ chọn id và class là giống nhau, chỉ khác nhau là id là duy nhất và không được tái sử dụng nhiều lần như class, id ưu tiên hơn class

# Câu 30: Trình bày mô hình hộp (box model) trong CSS?

Box model là thuật nghữ được sử dụng cho phần không gian mà phần tử đí chiếm giữ ở trên 1 trang web

1. Marggin: khoảng cách giữa 2 border liên tiếp nhau(lấy cái lớn hơn)
2. Border: đường viền của box model
3. Padding: khoảng cách từ content 🡺 border
4. Content: nội dung của box model

# Câu 31: Trình bày thuộc tính float, clear?

* Float: là một thuộc tính được sử dụng để chuyển một phần tử sang góc trái hoặc góc bên phải của không gian bao quanh nó. Đây là một thuộc tính cơ bản và rất cần thiết khi định dạng lại bố cục trang

1. left: Di chuyển phần tử về bên trái.
2. right: Di chuyển phần tử về bên phải.
3. none: Phần tử nằm ở vị trí của nó, trạng thái mặc định.
4. inherit: Kế thừa giá trị float từ phần tử cha.

* Clear: được sử dụng để xác định xem một phần tử cụ thể có được phép xuất hiện bên cạnh phần tử đã được đặt thành float trước đó hay không.

+ Các thuộc tính của clear:

Clear - none: cho phép phần tử hiển thị lên cả hai bên trái và phải (mặc định)

1. Clear – left: Không cho phép phần tử hiển thị phía bên trái.
2. Clear – right: Không cho phép phần tử hiển thị phía bên phải.
3. Clear – both: Không cho phép phần tử hiển thị cả 2 bên trái và phải.
4. Clear – inherit: Phần tử kế thừa giá trị clear của phần tử gốc.

# Câu 32: Thuộc tính position dùng để làm gì? Phân biệt các giá trị: static, absolute, relative và fixed?

Position: để định vị vị trí hiển thị của các phần tử thẻ HTML trên 1 trang web.

* Có 5 thuộc tính position: Static, relative, absolute, fixed, inherit.

+ static: default, k chịu ảnh hưởng của các thuộc tính TRBL

+ Relative: vị trí tương đối của phần tử so với vị trí mặc định, chịu ảnh hưởng của các thuộc tính TRBL và k gây ảnh hưởng đến các pt xung quanh

+ Absolute: giá trị này sẽ giúp định vị vị trí tuyệt đối cho các thành phần con chứa trong các thành phần cha.Chịu ảnh hưởng của TRBL và gây ảnh hưởng đến các pt xung quanh nó

+ Fixed: định vị vị trí tương đối cho các thành phần theo cửa sổ trình duyệt, giúp cho các phần tử luôn cố định tại một chỗ ( khi scroll thanh trình duyệt thì vị trí phần tử sẽ không thay đổi).

+Stickey: tương tự fixed nhưng khi kéo scroll ra khỏi pt cha sẽ mất

+ Inherit: Kế thừa thuộc tính position của thành phần cha.

# Câu 33: Responsive Web Design là gì?

* Responsive Web Design (RWD): là một kiểu mẫu phong cách thiết kế về giao diện cho phép bố cục website được thể hiện chuyên nghiệp và mang tính thẩm mỹ với nội dung có thể tương thích với tất cả màn hình của các thiết bị khác nhau như: PC, laptop, tablet, mobile,…

# Câu 34: Làm thế nào để tạo 1 trang web responsive?

Meta viewport = width = device – width, Innitial scale: 1.0

@media queries

Khai báo meta viewport

viết CSS cho từng loại thiết bị : sử dụng @media

@media screen and (max-width: 1024px){

}

767.98 là breakpoint

Sử dụng các grid hoặc flex box để tạo layout có thể điều chỉnh dựa trên kích thước màn hình

Sử dụng media queries trong CSS để thay đổi kiểu dáng và cách hiển thị trang web tùy thuộc vào kích thước màn hình. Điều này giúp tối ưu hóa trải nghiệm người dùng trên cả máy tính và thiết bị di động.

Thứ 3: tạo Flexible Media: tạo những breakpoint mà tại đó bố cục websit sẽ tự thay đổi thích ứng để tạo nên giao diện responsive.

# Câu 35: Bootstrap là gì?

* Bootstrap là một framework front-end mã nguồn mở (open-source) được sáng tạo bởi Twitter. Nó cung cấp một bộ công cụ và tài nguyên thiết kế để giúp phát triển web nhanh chóng và dễ dàng. Bootstrap được xây dựng bằng HTML, CSS, và JavaScript, và nó giúp tối ưu hóa quá trình phát triển web responsive.

# Câu 36: Một số thành phần thông dụng ở Bootstrap?

* Một số thành phần thông dụng trong bootstrap: Alert, Button, Dropdown button, icon, images, card, card-panel, navbar, carousel (ảnh trượt), flext-box, pagination (phân trang),

# Câu 37: Phân biệt Static web và Dynamic web?

|  |  |
| --- | --- |
| **static** | **dynamic** |
| HTML, CSS, JS (chỉ sử dụng đê xử lý sự kiện) | HTML, CSS, JS ở BE có JAVA, PYTHON, … |
| K co DB | Có DB |
| Độ tương tác với người dùng thấp | Độ tương tác với người dùng cao |
| Nội dung ít thay đổi theo thời gian | Nội dung thay đổi thường xuyên theo thời gian |

# Câu 38: Phân biệt 2 phương thức GET/POST

|  |  |
| --- | --- |
| Get | Post |
| Kém bảo mật (vì hiển thị thông tin trên URL). | Bảo mật (không hiển thị thông tin trên URL). |
| Giới hạn số lượng ký tự hiển thị <2048 | Không giới hạn ký tự |
| Chỉ gữi được dữ liệu dạng text | Có thể gữi giá trị dạng binary (file, images, …) |
| Có thể truy cập được từ thanh address của trình duyệt. | Không thể truy cập được từ thanh address của trình duyệt. |
| Gửi bằng thẻ a, URL, form + method = get | Gửi bằng thẻ form + method = post |

Câu 39: Phân biệt request/response

* HTTP Request: một HTTP client (máy khách) gữi một HTTP request (yêu cầu) lên server (máy chủ) nhờ gữi một thông điệp có định dạng nào đó.
* HTTP Reponse: là kết quả của server phản hồi lại cho client.
* Phân biệt luôn HTTP và HTTPS:

|  |  |
| --- | --- |
| Giống nhau | Khác nhau |
| Đều là giao thức truyền tải siêu văn bản từ client đến server. | HTTPS là giao thức HTTP được tích hợp thêm chứng chỉ bảo mật SSL 🡪 nhằm mã hóa các thông tin giao tiếp nhằm tăng tính bảo mật.  Summarize: HTTPS sẽ an toàn hơn và tính bảo mật cao hơn HTTP. |

Câu 40: Phân biệt forward/redirect

|  |  |
| --- | --- |
| forward() | sendRedirect() |
| Chuyển trang có mang kèm DL | Chuyển trang không mang theo DL |
| Chỉ chuyển đến 1 trang trong 1 dự án | Có thể chuyển đến 1 trang ngoài dự án |
| Nhanh hơn | Chậm hơn do lên DB load dât lại |
| URL vẫn là trang đó | URL là trang chuyển đến |

# Câu 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?

* Vì trong quá trình biên dịch, JSP sẽ chuyển sang Servlet rồi Servelet này sẽ được biên dịch

|  |  |
| --- | --- |
| Servlet | JSP |
| JAVA , có thể nhúng HTML | HTML có thể nhúng mã JAVA |
| Mạnh mẽ về xử lý giao diện | Mạnh mẽ về xử lý nghiệp vụ logic |

# Câu 42: JSTL là gì? Giới thiệu một số thẻ bạn đã sử dụng?

* Là một thư viện được xây dựng dựa trên ngôn ngữ java. Mục đích: để hỗ trợ hiển thị trên trang JSP dễ dàng hơn.
* Các nhóm thẻ JSTL:

+ Core tag

+ Thẻ formatting tag

+ Thẻ SQL tag

+ XML tag

+ JSTL function

* Một số thẻ hay sử dụng là:

+ Example: <c: redirected url=”…” >: chuyển hướng

<c: if test=”condition” >: thẻ điều kiện

<c forEach var=”variable represent” item=”list, array,…”

varstants=” loop state” >

<c: choose> (switch)

<c: when test=”condition 1”>…</c: when> (case 1)

<c: when test=”condition 1”>…</c: when> (case 2)

…

<c: otherwise>…</c: otherwise> (default)

</c: choose>

# Câu 43: Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC?

MVC (Model – View – Controller) – MVC Pattern Design là một mô hình kiến trúc phần mềm được tạo ra với mục đích quản lí và xây dựng dự án phần mềm có hệ thống hơn, nói cụ thể thì nó là một mẫu thiết kế nhằm chia tách phần giao diện và phần code của ứng dụng để dễ dàng quản lí, bảo trì và phát triển. MVC được áp dụng trên hầu hết các ngôn ngữ lập trình hướng đối tượng hiện nay như C++, C#, Java, PHP, …

Mô hình MVC chia cấu trúc phần mềm thành 3 thành phần chính: models, view, controller.

* Model: biểu diễn dữ liệu, xử lí nghiệp vụ (xử lý logic) validate DL và kết nối DB
* View: Chưa giao diện hiển thị cho người dùng (BE)
* Controller: Điều hướng các request, respone … là cầu nối nữa model và view

Luồng thực thi MVC:

1. Người dùng tương tác với web browser để gửi các request đến server (get/post)
2. Controller tiếp nhận request và xử lí nghiệp vụ đã định nghĩa trước
3. Controller có thể sử dụng model để truy xuất đến CSDL thông qua model
4. Model thực hiện việc xử lý kết nối với CSDL và trả về kết quả, xử lý kết quả rồi truyền cho controller
5. Controller truyền cho view phù hợp
6. Controller gửi view về web browser và hiển thị lên trình duyệt

# Câu 44: Sự khác nhau giữa MVC1 và MVC2?

**MVC1:**

**1. Ưu điểm:**

*Dễ dàng và nhanh chóng để phát triển các ứng dụng web.*

**2. Nhược điểm**:

* *Việc điều khiển điều hướng là không tập trung khi mỗi page có trình logic riêng để xác định page tiếp theo. Nếu tên JSP page bị thay đổi mà được tham chiếu bởi trang khác, thì chúng ta cũng phải thay đổi tên đó trong các trang tham chiếu.*
* *Tiêu tốn thời gian. Bạn cần dành nhiều thời gian hơn để phát triển các thẻ tùy biến Custom Tag trong JSP.*
* *Khó khăn để kế thừa. Nó thực sự tốt với các ứng dụng nhỏ nhưng với các ứng dụng lớn thì chưa hẳn.*

**MVC2:**

**1.Ưu điểm:**

*– Hệ thống phân ra từng phần dễ dàng cho việc phát triển*

*– Chia thành nhiều modun nhỏ có nhiệm vụ riêng và độc lập nên nhiều người có thể làm chung cùng 1 dự án*

*-Dễ dàng cho việc bảo trì và nâng cấp*

**2. Nhược điểm:**

*– Với hệ thống nhỏ mô hình mvc sẽ cồng kềnh khiến hệ thống chạy chậm hơn.*

# Câu 45: Mô tả cách thực hiện tạo và lấy kết quả 1 câu query bất kỳ bằng JDBC?

* B1: tạo kết nối đến database
* B2: gữi câu SQL query đến database sử dụng JDBC driver tương ứng.
* B3: JDBC driver kết nối với database
* B4: thực thi câu lệnh query để lấy kết quả trả về.
* B5: gữi dữ liệu đến ứng dụng thông qua driver management
* B6: xử lý dữ liệu trả về.
* B7: giải phóng kết nối đến database.

# Câu 46: Phân biệt các loại Statement?

* Statement: dùng để tạo câu lệnh truy vấn tĩnh (có lỗ hổng về bảo mật 🡪 SQL injection)
* PrepareStatement: dùng để tạo câu lệnh truy vấn động (truyền tham số ? vào được nên ngăn chặn được cơ chế SQL injection) extend từ Statement.
* Collable Statement: dùng để gọi 1 store procedure đã được tạo ở database và sau đó trả về kết quả ở ResultSet.

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

* Transaction là một tiến trình xử lý có điểm đầu và điểm cuối được chia nhỏ thành các operation, tiến trình được thực thi một cách tuần tự và độc lập, các operation theo nguyên tắc tất cả đều thành công thì tiến trình thành công, hoặc nếu chỉ có 1 operation thất bại thì tiến trình thất bại.
* Khi tiến trình thất bại thì đồng nghĩa với việc dữ liệu được khôi phục (rollback) lại trạng thái như ban đầu.

# Câu 48: Các tính chất của Transaction?

1. Atomicity: Tất cả hoặc không có gì ( “all or nothing”)
2. Consistency: ở mọi thời điểm dữ liệu luôn luôn phải nhất quán, tức là tuân theo các ràng buộc đã được định nghĩa. Khi transaction được thực hiện, dữ liệu sau khi cập nhật cũng phải ở trạng thái nhất quán. Nếu transaction gây ra những vi phạm về ràng buộc dữ liệu, hệ thống sẽ không cho phép thực hiện tiếp và hủy bỏ toàn bộ transaction…
3. Isolation: mỗi transaction được đảm bảo thực hiện trong một ngữ cảnh riêng biệt của nó và không bị ảnh hưởng bởi các transaction khác. Khi hai transaction cùng cập nhật một dữ liệu, hệ quản trị CSDL đảm bảo chúng được thực hiện tuần tự không dẫm lên chân của nhau.
4. Durability: Khi transaction thực hiện xong (đã commit), những cập nhật trở nên cố định và dữ liệu sẽ luôn luôn là như vậy. Khi hệ thống gặp sự cố bất ngờ, trong quá trình khôi phục lại nó sẽ đảm bảo khôi phục lại dữ liệu cho những transaction đã được commit.

+ Atomicity: tất cả đều thành công 🡪 tiến trình thành công or chỉ có 1 operation thất bại 🡪 tiến trình thất bại.

+ Consistency: ở mọi thời điểm, dữ liệu luôn luôn phải nhất quán (tức là tuân theo những rằng buộc đã được định nghĩa ban đầu).

+ Isolation: mỗi transaction đảm bảo thực thi trong một ngữ cảnh riêng biệt và không bị ảnh hưởng bởi các transaction khác.

+ Durability: khi transaction thực hiện xong những cập nhật đó trở nên cố định và dữ liệu sẽ luôn luôn là như vậy. (không rollback được).

# Câu 49: Phương pháp gọi 1 SP ở JDBC? Tự làm

# Câu 50: Phân biệt Ant, Maven, Gradle? …