**CÂU HỎI AUDIT MODULE 3**

1. **Phân biệt các câu lệnh DML & DDL? DML, DDL bao gồm các câu lệnh nào?**

-DML **(Data Manipulation Language)** : là ngôn ngữ thao tác csdl

+ select : lấy các bản ghi từ một hoặc nhiều bảng

+ insert: Thêm mới bảng ghi

+delete : xóa bản ghi

+ update : sửa đổi

- DDL **(Data Definition Language):** là ngôn ngữ định nghĩa csdl

+create : tạo bảng hoặc view hoặc database

+alter : sửa đối bảng đang tồn tại - ALTER TABLE ten\_bang

ADD

+rename : đối tên dbo

+truncate: xóa bản ghi của 1 table

+ drop : xóa dbo, view,database

-

1. **Cơ sở dữ liệu (database) là gì?**

* **Cơ sở dữ liệu** (**Database**) **là**

**+** một tập hợp các **dữ liệu** có tổ chức, lưu trữ và truy cập điện tử từ hệ thống máy tính.

**+** lưu dưới dạng các hàng và cột trong một loạt các bảng và phần lớn sử dụng SQL (để ghi và truy vấn **dữ liệu**.

1. **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 mô hình biểu diễn mối quan hệ giữa các thực thẻ với nhau
* Gồm 3 thành phần
* +Entiny : thực thể
* +Relationship: quan hệ giữa các thực thể
* + Attribute : thuộc tính

1. **Các bước chuyển đổi từ ERD -> Table?**

5 bược

B1: biến đổi các thực thể thành các table

Thuộc tính khoa của thực thể là khóa chính của table

Thuộc tín bình thường là các trường trong table

B3 : biến đổi các thực thể da trị thành table mới

Tách các thực thể thành các table mới. Thuojc tính khóa ngoại vào table mới

B4 : biến đổi mối quan hệ 1-1

Chuyển thuộc tính khóa từ phía chọn sang phía bắt buộc

B4 : 1-n

Chuyển thuojc tính khóa từ 1 sang nhiều

B5: n-n

Tạo table mới chứ thuộc tính khóa của cả 2 table

1. **Các bước chuẩn hóa dữ liệu?**

Có 4 bước chuẩn hóa dl

B1: 1NF – loại bỏ thuộc tính đa trị

B2: 2NF – loại bỏ các phụ thuộc bộ phận

B3: 3NF – loại bỏ thuộc gián tiếp

B4: boyce-codd – chỉ còn các phụ thộc x->A mà x là khóa

1. **Ràng buộc (constraint) là gì? Các loại ràng buộc trong MySQL?**

* Rằng buộc constraint là những quy tác được quy định trong bảng
* Để hạn chế DL đưa vào
* Giúp DL chính xác tin cậy và toàn vẹn

-Các loại ràng buộc

+khóa chính

+ khóa ngoại

+not null

+ unique : gtr duy nhất

+default : quy định gt mạc định khi dl của 1 côt null

+check : kiểm tra dl của 1 trường dựa vào đk

+index : tăng tốc độ tìm kiếm

1. **Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?**

-1 hoặc nhiều cột dùng xác định bản ghi là duy nhất trong table

- Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là khóa chính phức hợp

1. **Khóa ngoại (foreign key) là gì?**

-là cơ chế tạo nên liên kết giữa 2 bảng

- mỗi bảng có thể có hoặc ko hoặc có nhiều

- DL của khóa ngoại phải cùng kiểu với khóa chính mà tble nó đang chiếu đến

1. **Phân biệt DELETE và TRUNCATE?**

- **DELETE** : Xóa một hay tất cả dòng trong một bảng theo một điều kiện nhất định, dữ liệu có thể phục hồi lại.

**-TRUNCATE** : Xóa toàn bộ các dòng của bảng, giải phóng bộ nhớ **và** không thể phục hồi lại.  Và đặt lại giá trị tăng tự động thành 0.

1. **Làm sao để truy vấn dữ liệu từ nhiều bảng?**

* Dùng join để truy vấn DL kết hợp từ nhiều bảng, kq trả về danh sách thỏa mãn đk

1. **Phân biệt having và where?**

|  |  |
| --- | --- |
| where | having |
| -sd để chỉ định 1 điều khiện trong tìm nạp dl từ 1 hay kết hợp nhiều bảng | - -sd để chỉ định 1 điều khiện khi hàm đã đc gộp gruop by |
| -sd trước mệnh đề group by | - sau |
| Có thể sd với slect ,update, delete | * Chi sd đc với select |
|  |  |

1. **Các loại JOIN? Phân biệt?**

-inner join : trả bản ghi phù hợp cả 2 bảng

- left join : all bảng ghi bên trái và bản ghi phù hơp bên phải

-right join : all bảng ghi bên phải và bản ghi phù hơp bên trái

- full outer join : trả về all

1. **Phân biệt AND và OR?**

* And Trả về nếu 2 đky là true
* OR trả về nếu 1 trong 2 là true

1. **Phân biệt GROUP BY và ORDER BY?**

* Group nhóm dl có số bản ghi bằng nhau thành 1 nhóm kq dựa theo gtrij 1 hay nhiều cột
* Order by : sắp xếp bản ghi theo trật từ dựa vào 1 hay nhiều cột

+ ASC : tăng dần mạc định

+ DESC : giảm gần

Vd order by name desc;

1. **Muốn nối chuỗi trong MySQL thì sử dụng gì?**

* Sử dùng CONCAT (“chin chao” , malop) as malop để nối chuỗi
* Concat ko ảnh hưởng đến DL gốc

1. **Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?**

- avg để tính trung bình của 1 cột gtri số

1. **Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?**

-

1. **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 ten, diem form sinhvien where diem > 5

1. **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 stusent.ten, class.lop from student inner join class on studen.id\_class = class.id\_class;

1. **Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi**

Order by diem ASC;

1. **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 sinhvien order by ten DESC, tuoi ASC;

1. **Hiển thị các sinh viên có họ là ‘Le’ và tên là ‘Hai’?**

Select \* from sinhvien where ten like ‘%Le’ and ten like ‘Hai%’

1. **Hiển thị các sinh viên có tên bắt đầu là ‘C’?**

Select \* from sinhvien where ten like ‘%C’

1. **Phân biệt JOIN và UNION?**

-join : qhe giữa 2 bảng phải có 1 thuộc tính chung

+ có inner join, left,right,full

+

-Union : qhe giữa 2 bảng ko cần có thuộc tính chung , sl cột phait bằng nhau

+ có union và union all

+

1. **Phân biệt UNION và UNION ALL?**

-union : kết hợp 2 bộ kq từ 2 hay nhiều bảng nhưng sluong cột phải bằng nhau, xóa các hàng trung nhau

-union all : ko xóa các hàng trung nhau

1. **Phân biệt EXIST và IN?**

- In sẽ tìm ra tất cả các bộ thỏa yêu cầu câu truy vấn con rồi mới  
so sánh với câu truy vấn mẹ  
 -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

1. **Làm sao để có thể tăng tốc độ truy vấn?**

-Chỉ lấy ra DL cần thiết

- Giới hạn kết quả trả về

-Không làm cho truy vấn phức tạp hơn mức cần thiết

-rằng buộc đkiện hợp lý

- sử dùng index

1. **Các dạng tham số ở SP?**

* Có 3 dạng

+ in : dl đầu vào

+ out: dữ liệu đầu ra

+ inout : cả ra và vào

1. **Phân biệt SP, Trigger, Function?**

-SP là thủ tục

- function là hàm

1. **CSS là gì? Các cách nhúng CSS vào tài liệu html? Độ ưu tiên?**

* Viết tắt của : cascading style sheets
* Dùng để định dạng lại các phần tử được tạo ra bởi ngôn ngữ đánh dấu (html)
* Cách nhúng :

+inline style : khai báo trực tieeos trong thẻ html (1)

+internal stylesheet : sd thẻ <style> (2) tùy vào vị trí đặt trong head

+ External Stylesheet :sd thẻ link để nhúng file css từ ngoài vào

1. **Phân biệt giữa bộ chọn class và bộ chọn id?**

-class : class có thể trùng nhau

-id : 1 page chỉ có 1 id duy nhất

1. **Trình bày mô hình hộp (box model) trong CSS?**

-1 ptu html được coi như là một chiếc hộp

-sử dùng để thiết kế bố cục

- gồm 4 phần

+ content : là nd của thẻ

+padding : k/c tư content đến border

+border : là đường viền của thẻ

+ margin : là k.c từ border này đến border khác

1. **Trình bày thuộc tính float, clear?**

-float : chuyển 1 phần tử sang góc trái hoặc phải không gian bao quanh mình,

+left : cố định bên trí

+right : phải

+none : tại vị trí nó

+ inherit : phần tử kế thừa giá trị từ float cha

* Clear: Clear ngăn chặn thành phần A chiếm vùng không gian của thành phần B

1. **"Thuộc tính position dùng để làm gì?**

* Quy định cách thức sắp xếp đặt vị trí của một thành phần trong trang web
* Static: gtri mạc định, ko dùng left right

1. **Phân biệt các giá trị: static, absolute, relative và fixed?"**

+static : gia trị mạc định ko dùng left right

+ralative : can chỉnh so với vị trí ban đầu

+absotule: căn chỉnh so với thẻ cha chưa nó ( thẻ cha cần có position : static)

+fixed : căn chỉnh so vói thanh trình duyệt ko bị ảnh hưởng bởi việc kéo thả trong thanh

+ stickey : khi scoroll vtri nào đó thì ptu đó được bám cố định trên trang

1. **Responsive Web Design là gì?**

-là thiết kế web thích ứng. Web hiển thị ko bị mất nội dung, trên nhiều loại thiết bị có kích thước khác nhau

1. **Làm thế nào để tạo 1 trang web responsive?**
2. **Bootstrap là gì**?

* Là 1 font-end frameword miễn phí phát triển web nhanh hơn
* Gồm các thiết kế dựa trên html,css
* Cung cấp khả năng tạo các thiết kế, hiển thị icon
* Chia màn hình thành 12 cột

+ưu : miễn phí, phát triển nhanh, tương tác tốt

+ nhược : vì làm framwrodk nen làm theo khuôn , dư thưa import nhưng ko dugnf hết

1. **Một số thành phần thông dụng ở Bootstrap?**

-phân tran : pagination

-nut : button

- thông báo “ modal’

-nav

1. **Phân biệt Static web và Dynamic web?**

|  |  |  |
| --- | --- | --- |
|  | Static web ( web tĩnh) | Dynamic web(động) |
| Căn bản | Nd không thay đổi cho đến khi có người thay đổi trên Url | Nd thay đổi |
| Thiết kế | Đơn giản | Phức tạp |
| Ngôn ngữ | Html, js,css.., font-end | Asp, servler, back-end |
| Thay đổi thông tin | Hiếm | Thưởng xuyên |
| Thời gian tải trang | nhanh | Châm |
| Sd csdl | Ko | có |

1. **Phân biệt 2 phương thức GET/POST**

|  |  |
| --- | --- |
| Get | Post |
| - được gọi khi nhập thông tin từ thanh url hoặc khi method= ‘get’ | - dduoc gọi khi method = ‘post’ |
| nhanh | Cham |
| dL gửi đi hiển thị trên thanhd diều khiển | DL gửi đi không hiển thị trên thanh điều khiển |
| Không nên sd để lưu trữ các DL nhạy cảm | Có thể |
| Gửi những DL nhỏ thường 2048 ký tự | Không hạn chế dung lượng |
| Chỉ gửi đi dạng văn bản | Tất cả các dạng |
| DL được lưu lại trong trang web | Không được lưu lại |
|  |  |

1. **Phân biệt request/response**

|  |  |
| --- | --- |
| Request | Response |
| -client gửi yêu cầu lên server | -là kết quả server trả về cho client |
| - thông qua lớp HttpServletRequest | -thông qua lớp HttpServletResponse |

1. **Phân biệt forward/redirect**

|  |  |
| --- | --- |
| forward | redirect |
| -chuyển hướng kèm theo DL | -chuyển hướng không gửi DL |
| -Urlpatter sẽ thay đổi theo url đích | -url không thay đổi |
| -chỉ chuyển hướng trong hệ thống web | -chuyển hướng ngoài hệ thống web |
| - tốc độ chuyển đến trang sử lý nhanh hơn | -chậm hơn |

1. **"JSP là gì? Nói: “Về bản chất JSP cũng chính là Servlet” đúng hay sai? Giải thích?**

-JSP là ngôn ngữ kịch bản phía server cho phép ng dùng tạo các trang web động

Jsp tự động biên dịch và phát triển khi có bất kì thay đổi.

-về bản chất JSP chính là servler là đùng

VÌ : vì trong quá trình biên dịch jsp chuyển thành servlet và servlet sẽ được biên dịch

1. **So sánh giống và khác nhau giữa JSP và Servlet?**

-servler là công nghệ sd để tạo web, Một API cung cấp các interface

-Giống nhau:

+ nằm phía server

+ vòng đời giống nhau

+ đều xử lý DL “ động”

-Khác

|  |  |  |
| --- | --- | --- |
|  | Servlet | JSp |
| Xly hiển thị | Yếu | Mạnh |
| Xly nghiệp vụ và điều phối | Mạnh | Yếu |

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

-JSTl là bộ thư viện chuẩn được bổ sung với mục đích tối ưu trong lập trình.

- mốt số thẻ

+ core: thẻ điều khiển

+format : định dạng đa ngôn ngữ

+xml: xử lý tai liệu xml

+sql : làm việc với sql

+foreword() : chuyển đổi request

1. **Trình bày kiến trúc MVC? Vai trò của các thành phần trong kiến trúc MVC?**

* Là kiến trúc mềm, Các thành phần tổ chức theo các tầng,theo chiều ngang, mỗi tầng đảm nhiệm 1 nhiệm vụ
* Model : chứa các class của đối tượng
* View : hiển thị DL, giao diện người dùng
* Controller : Xử lý thao tác người dùng kết nối model và view

1. **Mô tả cách thực hiện tạo và lấy kết quả 1 câu query bất kỳ bằng JDBC?**
2. **Sự giống, khác nhau và ưu nhược điểm của MVC1 & MVC2?**
3. **Kỹ thuật sử dụng transaction bằng JDBC là như thế nào?**
4. **Các tính chất của Transaction?**

- Transaction là 1 tiến trình xử lý có điểm bắt đầu và điểm kết thúc

- gồm nhiều thực thi nhỏ, trong đó mỗi phép thực thi sẽ được thực hiện tuần tự và độc lập theo nguyên tắc là tất cả thành công thì transaction mới thành công, 1 phép thực thi thất bại thì tất cả thất bại

-

- Tính nguyên tử : tất cả thành công hoặc ko

-Tính nhất quán: Đảm bảo răng tính đồng nhất của DL

- tính đọc lập : Đảm bảo transaction này độc lập với transaction khác.

- Tính bển vững : khi 1 transaction đã đc ký thác thì nó vẫn tồn tại như thế cho dù có xảy ra lỗi.

1. **Phương pháp gọi 1 SP ở JDBC?**
2. **Phân biệt Ant, Maven, Gradle?**

|  |  |  |
| --- | --- | --- |
| ant | maven | gradle |
| 1 công cụ phần mềm tự động hóa | một công cụ quản lý và hiểu dự án phần mềm | một hệ thống tự động hóa xây dựng nguồn mở |
|  | Sd xml | Không, |
|  |  | Ngắn hơn, sạch hơn tập lệnh của maven |
|  | Viết bằng java | Java, gradle, kotlin |
| Là công cụ xây dựng | Vừa là công cụ xây dựng và cung cấp các hướng dẫn cho các thực tiễn tốt nhất | ho phép cấu trúc xây dựng. hỗ trợ các bản dựng đa dự án, tăng năng suất |
| Sd tiếp cận mệnh lênh | Sd pp khai báo | một công cụ tự động hóa xây dựng nguồn mở tập trung vào tính linh hoạt và hiệu suất. |
| Ko có vòng đời | Xd vòng đời, giai đoạn, muc tiêu |  |
|  |  |  |
|  |  |  |

1. **Phân biệ SP, Trigger, Funcation?**

|  |  |  |
| --- | --- | --- |
| SP | Funcation | Trigger |
| -trả về zero , 1 hoặc nhiều giá trị | -trả về 1 giá trị duy nhất(có thể là bảng) | Về cơ bản giống thủ tục  Khác ở chỗ  Thủ tục đc sdungf khi có lời gọi đến   * Trigger tự động xảy ra |
| -có cả tham số đầu vào và đầu ra | - có hoặc ko có tham số đầu vào |  |
| Thủ tục oko gọi đc hàm | Hàm gọi đc thủ tục |  |

1. **Phân biệt Statement và PrepareStatement** ?

-Giống nhau

Cả 3 đều là interface CallbleS extends prepar extends statement

- Khác nhau

|  |  |  |
| --- | --- | --- |
| Statement | PrepareStatement | CallbleStatement |
| Tạo câu lênh sql tĩnh  Không truyenf tham số | Động, có thám số  Truyền tham số | Thường để gọi thủ tục |
| Thấp  Không bảo mật có thể bị tấn công bởi SQL injection  (sql injectin là kỹ thuật lợi dụng những lỗ hổng về câu truy vấn để chèn thêm đoạn sql làm sai lệnh câu truy vấn ban đầu) | cao | cao |