**BỘ CÂU HỎI PHỎNG VẤN MODULE 3**

**[BC-JAVA-JWBD] Java Web Back-end Development 2.1**

**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 (lưu lại) dưới dạng các tài liệu chữ, số, ảnh, âm thanh, hình ảnh…

=> Khi xử lý dữ liệu thì chúng ta có được thông tin (information)

- Thông tin (information) là sự thông báo, trao đổi, giải thích về một đối tượng cụ thể nào đó và được thể hiện thông qua các dạng tín hiệu như âm thanh, chữ số, chữ viết… nhằm mang lại một sự hiểu biết nào đó cho đối tượng nhận tin.

*Ví dụ: - Cô giáo giảng bài, sự truyền đạt kiến thức là thông tin*

*- Học sinh ghi chép kiến thức lại vào vở, đó là dữ liệu*

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

- Database (CSDL) là một tập hợp nhóm dữ liệu có tổ chức, là một hệ thống các thông tin có cấu trúc liên quan nhau, được lưu trữ trên các thiết bị lưu trữ, nhằm thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng.

*Ví dụ: - Banking (khách hàng, gửi tiền, rút tiền, chuyển khoản)*

*- School (class, student, teacher)*

**3. Giới thiệu một số hệ quản trị cơ sở dữ liệu hiện nay?**

MySQL, Mongo DB, Maria DB, Microsoft SQL Server, Oracle DB, IBM DB2, Microsoft Access…

**4. Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) là gì?**

*- Cung cấp nhiều chức năng, công cụ để thao tác với CSDL đó (thêm, sửa, xóa)*

***[Mở rộng thêm - CSDL Quan hệ là CSDL được xây dựng dựa trên mô hình dữ liệu quan hệ (relationalmodelofdata)]***

- RDBMS là cơ sở nền tảng cho SQL, và cho tất cả các hệ thống cơ sở dữ liệu hiện đại như MS SQL Server, IBM DB2, Oracle, MySQL và Microsoft Access.

• Dữ liệu được tổ chức trong các bảng (table)–còn được gọi là quan hệ (relation)

• Các bảng bao gồm cột (column) và dòn g(row)

• Dòng còn được gọi là bảng ghi (record) hoặc hang (tuple)

• Cột còn được gọi là thuộc tính (attribute) hoặc trường (field)

• Một bảng thường lưu trữ dữ liệu của một loại thực thể (entity) nhất định.

**5. Toàn vẹn dữ liệu (data integrity) của một cơ sở dữ liệu là gì?**

- Là cách xây dựng CSDL làm sao để dữ liệu chứa trong CSDL này phải: chính xác và đáng tin cậy, *đầy đủ, không thiếu*

**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?**

- ERD là tên gọi tắt của Entity Relationship Diagram trong đó:

+ Entity: thực thể (có thuộc tính)

+ Relationship: các mối quan hệ giữa các thực thể đó.

+ Diagram: sơ đồ

=> ERD là một sơ đồ, thể hiện các thực thể có trong database và mối quan hệ giữa chúng với nhau.

- Thành phần cơ bản của ERD:

+ Entity (Thực thể): là 1 đối tượng cụ thể hoặc trừu tượng, được mô tả bởi 1 tập các thuộc tính.

+ Relationship (Mối quan hệ): nhị nguyên, đa nguyên

+ Property/Attributes (Thuộc tính): đơn trị, đa trị

+ Entity Set (Tập thực thể)

+ Relationship Set (Tập mối quan hệ)

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

- Ràng buộc (constraint) là những quy tắc được áp dụng trên các cột dữ liệu, trên bảng. Được sử dụng để kiểm tra tính hợp lệ của dữ liệu vào, đảm bảo tính chính xác, tính toàn vẹn của dữ liệu.

|  |  |
| --- | --- |
| NOT NULL | - Thiết lập việc một cột không được phép chứa giá trị NULL. |
| UNIQUE | - Thiết lập việc dữ liệu trong một cột không được phép trùng nhau. |
| PRIMARY KEY | - Kết hợp giữa NOT NULL & UNIQUE.  (cột không được phép chứa giá trị NULL & dữ liệu không được trùng nhau) |
| FOREIGN KEY | - Xây dựng một mối liên kết giữa hai cái bảng nhằm đảm bảo tính "toàn vẹn dữ liệu" của các bảng (khi trong hai cái bảng đó, có một bảng tham chiếu đến dữ liệu của cái bảng còn lại) |
| CHECK | - Giới hạn phạm vi giá trị được nhập vào bên trong cột. |
| DEFAULT | - Thiết lập giá trị mặc định cho cột trong trường hợp chúng ta chèn một hàng mới vào trong bảng mà không gán giá trị cho cột này. |
| INDEX | - Lập chỉ mục cho các cột để cải thiện tốc độ truy xuất dữ liệu từ cơ sở dữ liệu. |

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

**- Khóa chính:** + Là một cột (hoặc nhiều cột) được sử dụng để xác định một bản ghi duy nhất trong bảng.

+ Khóa chính là UNIQUE và NOT NULL.

+ Mỗi bảng chỉ có 1 khóa chính

*(Không có 2 dòng nào trong 1 bảng trùng nhau hoàn toàn)*

**- "Khoá chính chứa một hoặc nhiều thuộc tính khoá"** có nghĩa là 1 khoá chính có thể bao gồm nhiều thuộc tính để xác định khoá chính đó.

*- Ví dụ: như mỗi người đều có một căn cước công dân khác nhau, hay mỗi bạn học viên đều có một mã học viên riêng của mình,…*

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

- Khoá ngoại của một bảng là khoá chính của bảng khác.

- Khoá ngoại được dùng để tạo liên kết giữa các bảng.

- Kiểu dữ liệu khóa ngoại, khóa chính phải giống nhau.

- Được đặt trên 1 cột của bảng này và tham chiếu đến khóa chính của bảng khác.

***(Tập dữ liệu (1,2,3,…) của bảng khóa ngoại là con <= tập dữ liệu khóa chính(1,2,3,4,…))***

*- Ví dụ: có bảng HocSinh có các trường là HOCSINHID, Ho, Ten, Ngaysinh, Dantoc, Noisinh, Diachi, GT, LopId*

*Bảng LopHoc có các trường là LopId, TenLop*

*Trong ví dụ trên LopId sẽ là khoá ngoại của bảng HocSinh vì đây là trường (cột) tham chiếu đến khoá chính của bảng LopHoc.*

**10. Trình bày về khóa duy nhất - unique? Phân biệt giữa Primary key và unique?**

*\* UNIQUE: Sử dụng để đảm bảo dữ liệu của cột là duy nhất, không trùng lặp giá trị trên cùng 1 cột.*

***Phân biệt Primary key và Unique:***

- Primary key: khoá chính duy nhất định danh cho mỗi hàng trong bảng, NOT NULL, tạo chỉ mục theo nhóm.

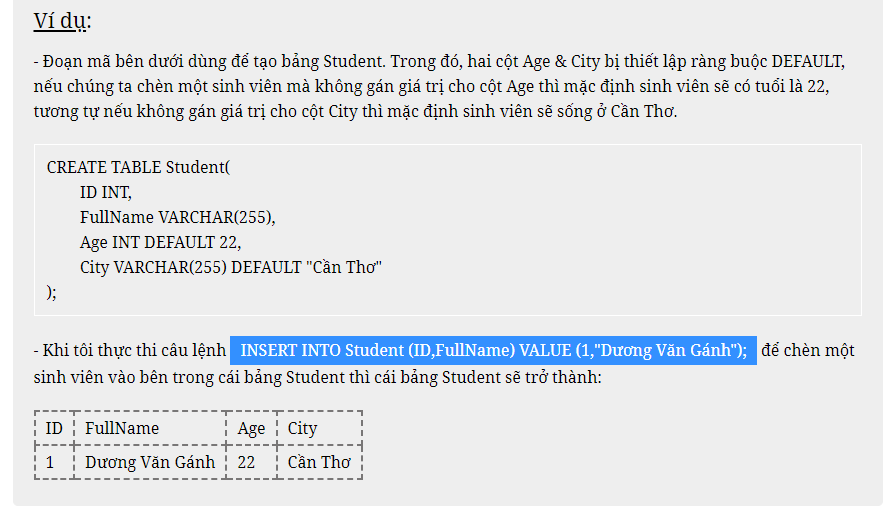
- Unique: không phải khoá chính có nhiều hơn 1 khoá, chấp nhận NOT NULL, tạo chỉ mục không phân cụm. ***(có thể sửa, thay đổi được – không nên lấy CMND, SĐT làm khóa chính => vì tuy là nó là duy nhất nhưng có thể đổi được)***

**11. Thuộc tính not null là gì?**

- Dữ liệu của cột không được nhận giá trị NULL

**12. Trình bày ràng buộc mặc định (default)? Ví dụ?**

- Gán giá trị mặc định cho cột trong trường hợp dữ liệu của cột không được nhập vào hay không được xác định.



**13. Trình bày ràng buộc kiểm tra (check)? Ví dụ?**

- Từ khóa CHECK được sử dụng để quy định điều kiện ràng buộc, xác thực cho giá trị của một cột.

- VD:

CREATE TABLE persons (

id INT NOT NULL,

last\_name VARCHAR(255) NOT NULL,

first\_name VARCHAR(255),

age INT,

CHECK (age>=18)

);

**14. Trình bày cách tự động gán và tăng giá trị trong 1 cột của MySQL?**

- Thuộc tính AUTO\_INCREMENT thường được dùng trên những cột là khóa chính & có kiểu dữ liệu là số nguyên

=> Chức năng của nó là tự động gán giá trị cho cột khi chúng ta thêm hàng mới vào bên trong bảng *(giá trị của hàng đầu tiên được thêm vào bảng là 1, giá trị này sẽ tự động tăng lên một sau mỗi hàng được thêm vào bảng)*

**15. Yêu cầu nhập vào trường tuổi của nhân viên phải từ 18-60, sử dụng ràng buộc nào? Viết ràng buộc?**

*CHECK*

*VD: CONSTRAINT Employee CHECK (Age >= 18 AND Age <= 60)*

**16. Yêu cầu trường cặp giá trị (MaSinhVien,MaMonHoc) phải duy nhất, not null, sử dụng ràng buộc nào? Viết ràng buộc?** - PRIMARY KEY

CREATE TABLE student (

MSV int,

MMH Varchar (45)

CONSTRAINT pkey\_MSV\_MMH PRIMARY KEY (MSV, MMH)

);

**17. Yêu cầu Tỉnh thành chỉ được nhập Hà Nội hoặc TP. Hồ Chí Minh, sử dụng ràng buộc nào? Viết ràng buộc? -** CHECK - OR

Ví dụ 1: CREATE TABLE thongtin (

tinhThanh VARCHAR(30),

CHECK (tinhThanh = 'Hà Nội' **OR** tinhThanh = 'TP. Hồ Chí Minh')

);

Ví dụ 2: ALTER TABLE thongtin (

ADD CHECK (tinhThanh = 'Hà Nội' **OR** tinhThanh = 'TP. Hồ Chí Minh')

);

**18. Yêu cầu trường Họ tên không được để trống sử dụng loại ràng buộc nào? Viết ràng buộc? -** *NOT NULL*

*fullname VARCHAR (45) 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*

*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?**

*SELECT studentID, studentName FROM student*

*WHERE studentName 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?** - DEFAULT

- Trong sửa bảng:

ALTER TABLE lamviec

ADD CONSTRAINT df\_ngay\_bat\_dau DEFAULT GETDATE() FOR NgayBatDau;

- Trong tạo bảng:

CREATE TABLE lamviec (

Macongviec char(10) NOT NULL,

MaNV char(10) NOT NULL,

NgayBatDau date DEFAULT GETDATE(),

ThoiHan int,

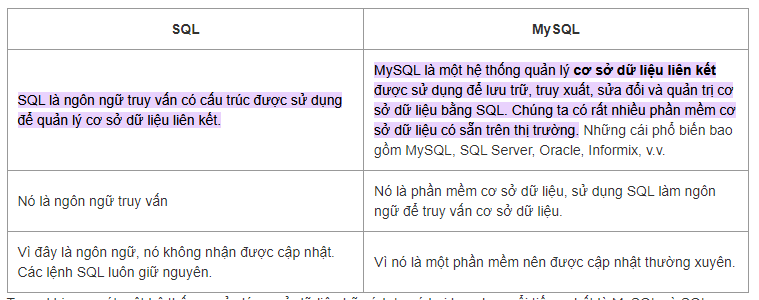
NgayHetHan date

);

**22. SQL là gì?**

- SQL là viết tắt của Structured Query Language là ngôn ngữ truy vấn cấu trúc được sử dụng để tạo mới, cập nhật và truy vấn dữ liệu từ các cơ sở dữ liệu quan hệ như MySQL, Oracle, SQL Server, PostGre,…

*[Mở rộng - MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server]*



**23. Trình bày về câu lệnh CREATE?**

- Câu lệnh CREATE dùng để tạo DATABASE, TABLE, VIEW, USER, STORE PROCEDURE.

**24. Trình bày về câu lệnh alter?**

- Thêm, xóa hoặc sửa đổi các cột trong một bảng hiện có

- Thêm và bỏ các ràng buộc khác nhau trên một bảng hiện có.

**25. Trình bày về các kiểu dữ liệu ký tự/chuỗi trong MySQL? Phân biệt giữa các loại?**

|  |  |
| --- | --- |
| CHAR | Chuỗi có độ dài cố định từ 1 đến 255 ký tự. |
| VARCHAR | Chuỗi có độ dài thay đổi từ 1 đến 255 ký tự. |
| BLOB hoặc TEXT | Lưu trữ một lượng lớn dữ liệu. Tối đa 65.535 ký tự |
| TINYBLOB hoặc TINYTEXT | Không xác định độ dài. Tối đa 255 ký tự |
| MEDIUMBLOB hoặc MEDIUMTEXT | Không xác định độ dài. Tối đa 16.777.215 ký tự |
| LONGBLOB hoặc LONGTEXT | Không xác định độ dài. Tối đa 4.294.967.295 ký tự |
| ENUM | Chỉ ra 1 danh sách các đối tượng mà trường phải nhận (có thể là Null) |

**26. Trình bày về các kiểu dữ liệu số trong MySQL? Phân biệt giữa các loại?**

|  |  |
| --- | --- |
| INT | Số nguyên có kích cỡ bình thường (độ rộng lên 11 chữ số: 0 –> 429… hoặc-214… -> 214…) |
| TINYINT | Số nguyên có kích cở rất nhỏ (độ rộng lên 4 chữ số) :(-128 -> 127 hoặc 0->255). |
| MEDIUMINT | Số nguyên có kích cỡ trung bình (độ rộng lên 9 chữ số: 0 –> 16.777.215 hoặc -838… -> 838…) |
| BIGINT | Số nguyên có kích cỡ lớn (độ rộng lên 20 chữ số) |
| FLOAT | Số thực (có dấu chấm động, phần thập phân lên tới 24 vị trí số) |
| DOUBLE | Số thực (có dấu chấm động, phần thập phân lên tới 53 vị trí số) |
| DECIMAL = NUMERIC | Mỗi chữ số thập phân chiếm 1 byte |

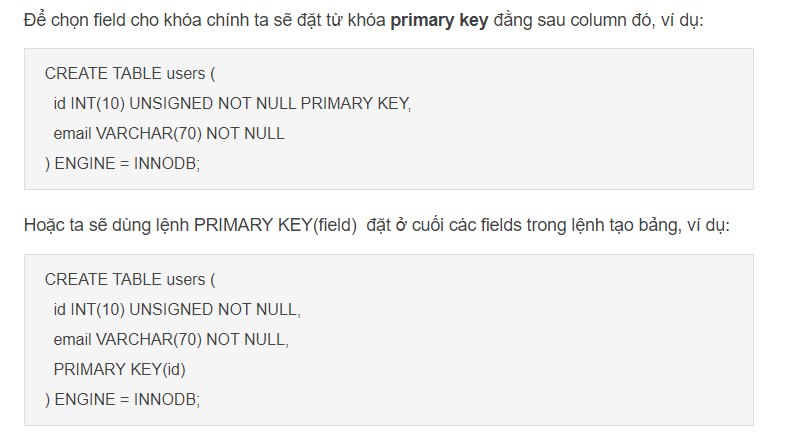
**27. Trình bày các kiểu dữ liệu ngày tháng trong MySQL? Phân biệt giữa các loại?**

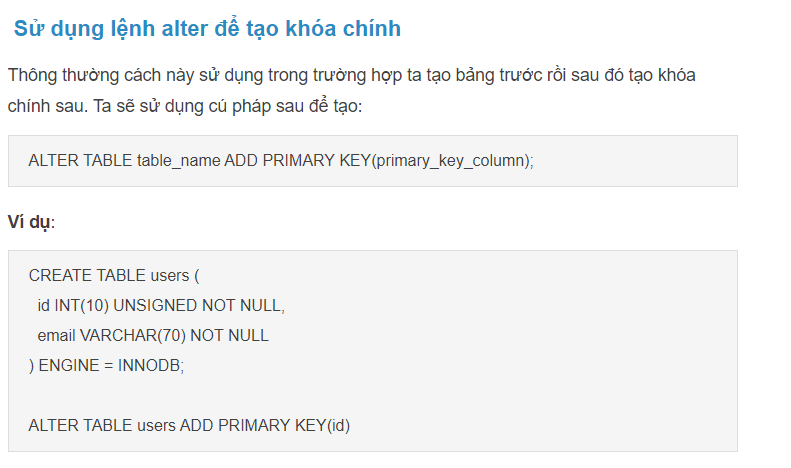
|  |  |
| --- | --- |
| *DATE* | *có định dạng YYYY-MM-DD.* |
| *DATETIME* | *có định dạng YYYY-MM-DD HH:MM:SS* |
| *TIMESTAMP* | *giống với DATETIME nhưng không có dấu gạch nối* |
| *TIME* | *định dạng HH:MM:SS* |
| *YEAR* | *định dạng 2 hoặc 4 chữ số. Độ dài mặc định là 4 (1901-2155)* |

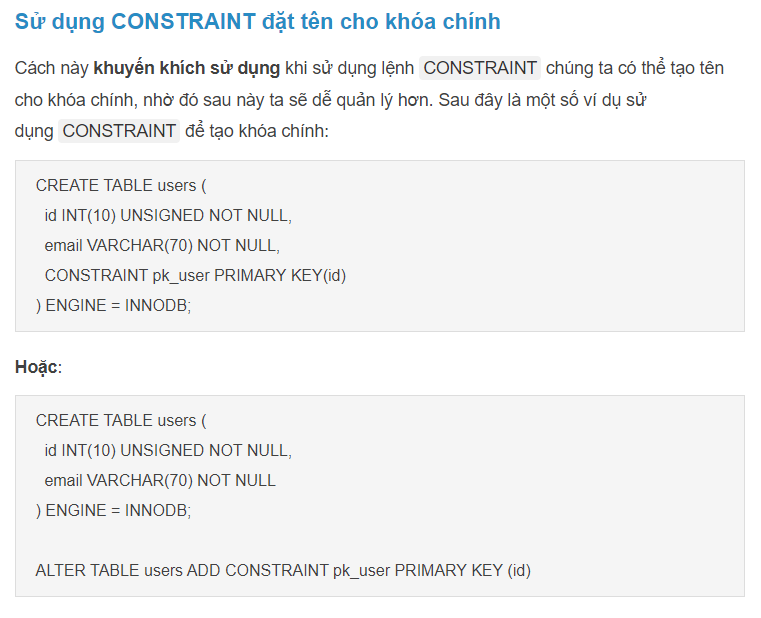
**28. Yêu cầu bổ sung thêm khóa chính vào một bảng đã được tạo sẵn thì làm thế nào?**

*ALTER TABLE table\_name*

*ADD PRIMARY KEY (COLUMN);*



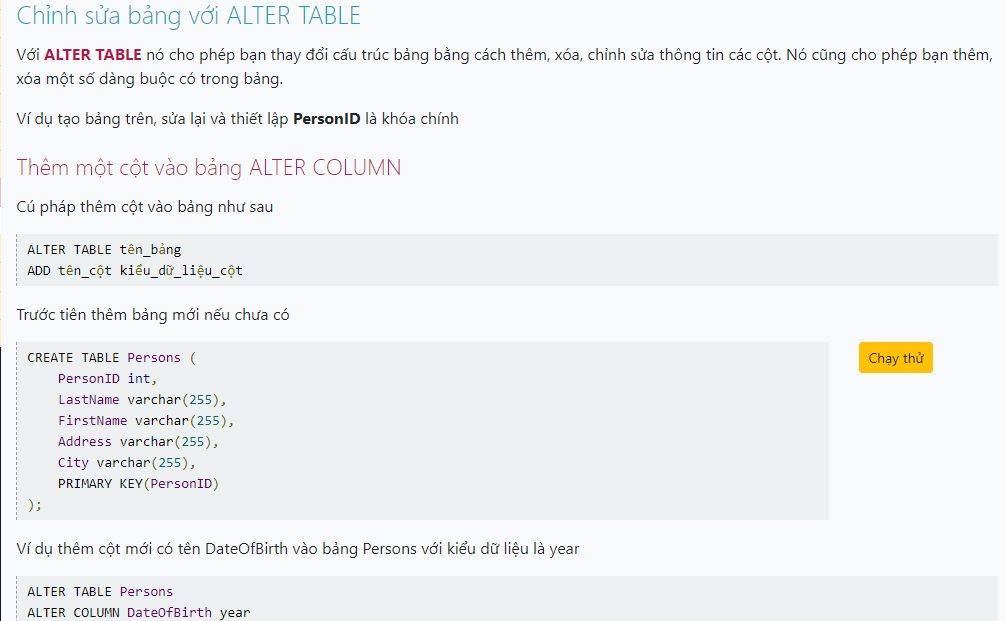




**29. Yêu cầu bổ sung thêm 1 cột vào bảng đã được tạo sẵn thì làm thế nào?**

*ALTER TABLE table\_name*

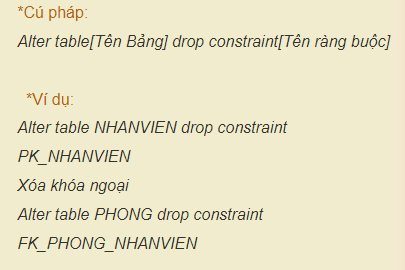
*ADD new\_column constraint;*



**30. Yêu cầu xóa một ràng buộc đã được tồn tại sẵn trong bảng thì làm thế nào?**

*ALTER TABLE table\_name*

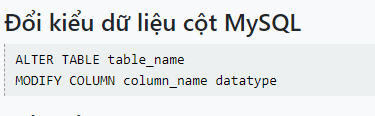
*DROP COLUMN column\_name constraint;*



**31. Yêu cầu sửa kiểu dữ liệu của 1 cột có sẵn trong bảng thì làm thế nào?**

*ALTER TABLE table\_name*

*MODIFY COLUMN name\_column constraint;*

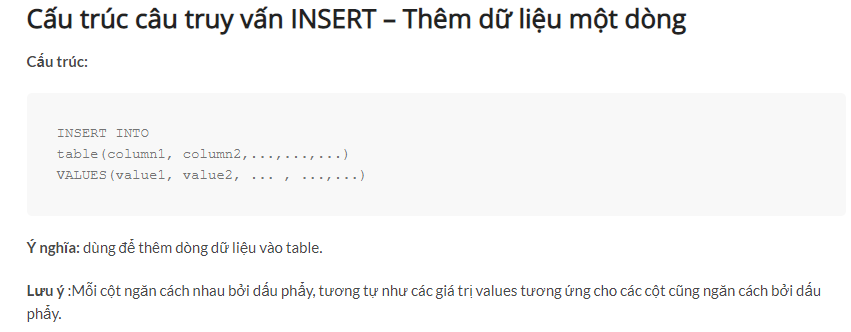


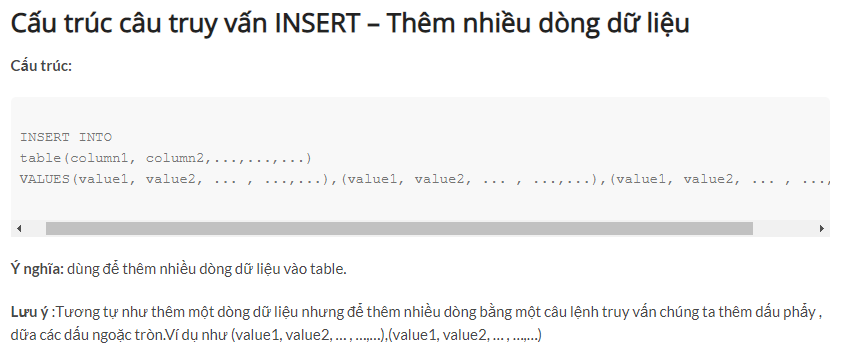
**32. Trình bày về câu lệnh insert? insert nhiều bộ giá trị?**

- Thêm dữ liệu (data) vào một bảng (table) nào đó, thì các bạn cần sử dụng câu truy vấn**INSERT**

*INSERT INTO table\_name (column1, column 2,…)*

*VALUES (values1, values2,…)*





**33. Trình bày về câu lệnh update? Phân biệt alter và update?**

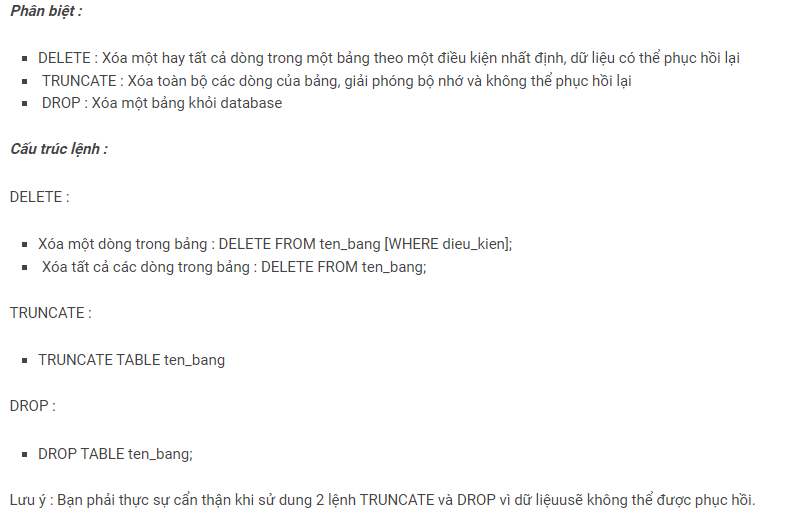




**=>** Lệnh ALTER (thêm, sửa, xóa) được sử dụng để **sửa đổi cấu trúc** của mối quan hệ trong cơ sở dữ liệu. UPDATE được sử dụng để **sửa đổi các giá trị** của các bản ghi hoặc bộ dữ liệu trong mối quan hệ

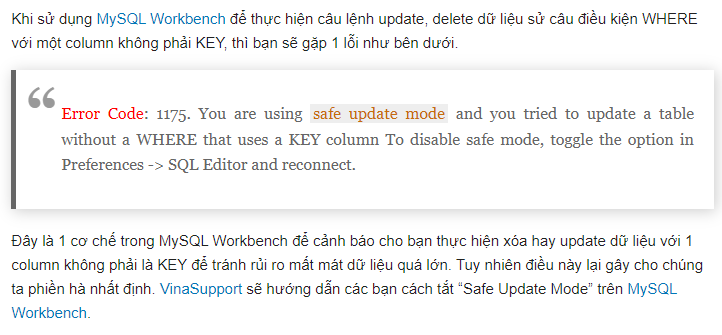
**34. Trình bày về câu lệnh delete? Phân biệt drop và delete?**

**-** Lệnh DELETE được dùng để xóa các hàng trong bảng (có thể là một hoặc nhiều hàng)

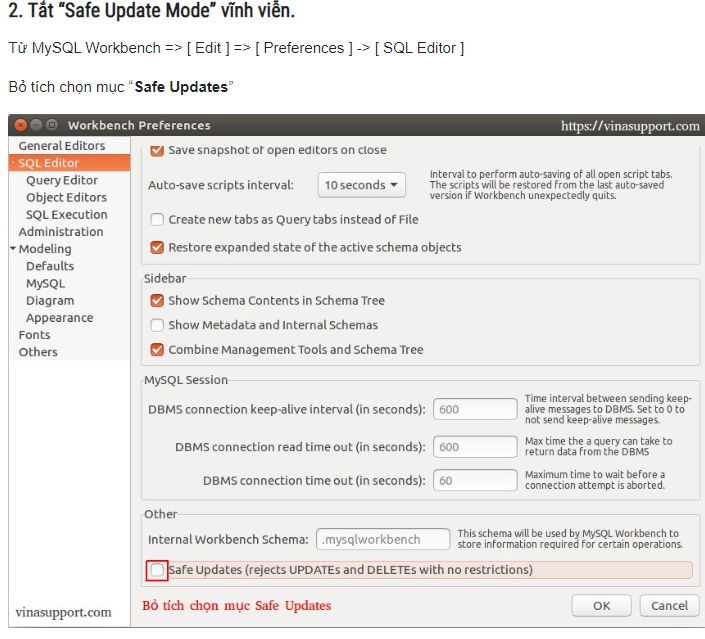


**35. Bạn có biết “safe update mode”? Nếu biết, hãy trình bày về cơ chế này**

*Cảnh báo cho bạn xoá hay update dữu liệu một cột không phải là KEY để tránh rủi ro mất mát DL.*





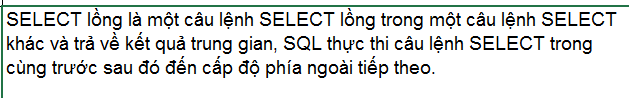


**36. Trình bày về câu lệnh select?**

- Câu lệnh SELECT trong MySQL 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. Các bảng kết quả này được gọi là tập kết quả (result-sets).

* *SELECT \* FROM table;*
* *SELECT column1, column2…FROM table;*

**37. Trình bày về select lồng? Lấy ví dụ?**



SELECT \*

FROM students

WHERE GPA > (

    SELECT AVG(GPA)

    FROM students);

**38. Trình bày về phép nối (join)?**

- Mệnh đề JOIN được sử dụng để kết hợp các hàng từ hai hoặc nhiều bảng, dựa trên một cột có liên quan giữa chúng.

- Các câu lệnh JOIN:

• INNER JOIN (hoặc JOIN): Chỉ trả về các record có mặt ở cả 2 bảng.

• LEFT JOIN: Trả về tất cả các record có mặt ở bảng bên trái,và những record tương ứng ở bảng bên phải.

• RIGHT JOIN: Trả về tất cả các record có mặt ở bảng bên phải, và những record tuơng ứng ở bảng bên trái.

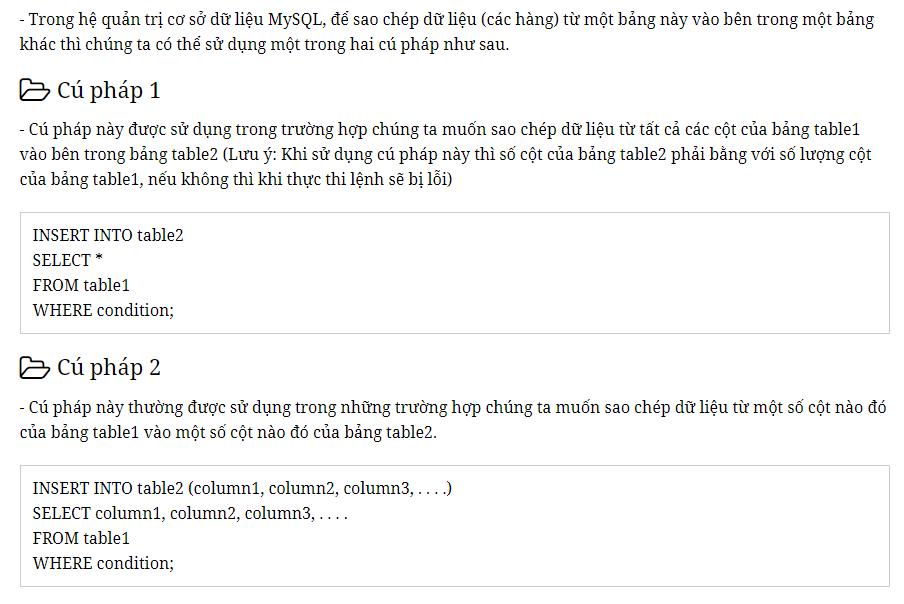
• FULL JOIN: Trả về tất cả các record ở cả hai bảng

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

Sử dụng chỉ mục (INDEX).

* *Giảm thiểu sự kết nối đến hệ quản trị CSDL.*
* *Chỉ SELECT những trường mà bạn cần.*
* *Giải phóng bộ nhớ ngay sau khi chạy xong.*

**40. Trình bày việc sao chép dữ liệu từ bảng này sang bảng khác trong MySQL?**



**41. Phân biệt giữa count(\*) và count(cột)?**

- Count(\*): đếm giá trị thẻ hàng và lấy cả giá trị null.

- Count(tên cột): đếm theo cột từ trên xuống dưới, tuy nhiên bỏ qua giá trị null và đếm tiếp.

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

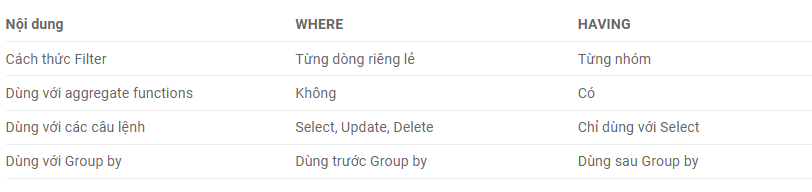
Cách 1: Dùng SELECT chọn nhiều cột từ nhiều bảng (+ WHERE).

Cách 2: Dùng các câu lệnh JOIN + ON.

Cách 3: Dùng UNION để kết hợp nhiều SELECT.

**43. Phân biệt giữa Where và Having?**

WHERE kiểm tra điều kiện trước khi tổng hợp diễn ra.

HAVING kiểm tra điều kiện sau khi tổng hợp diễn ra (thường đi với lệnh GROUP BY).

**44. Trình bày mệnh đề Order By trong câu lệnh Select?**

*Dùng kết hợp với lệnh SELECT để sắp xếp các kết quả trả về theo thứ tự tăng dần hoặc giảm dần (dựa trên giá trị của một cột nào đó)*

* *ASC: sắp xếp tăng dần.*
* *DESC: sắp xếp giảm dần.*

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

SELECT s.StudentId, s.StudentName, AVG(Mark)

FROM Student AS s

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

SELECT S.StudentId, S.StudentName, AVG(Mark)

FROM Student AS S

JOIN Mark AS M

ON S.StudentId = M.StudentId

GROUP BY S.StudentId, S.StudentName;

**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?**

SELECT s.student\_name, s.avg

FROM student AS s

WHERE s.avg >5;

Hoặc:

*SELECT S.StudentId, S.StudentName, AVG(Mark)  
FROM Student S*

*join Mark M on S.StudentId = M.StudentId  
GROUP BY S.StudentId, S.StudentName  
HAVING AVG(Mark) > 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 S.StudentId, S.StudentName, C.ClassName  
FROM Student S join Class C on S.ClassId = C.ClassID*

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

- Thêm mệnh đề ORDER BY diemthi [ASC] phía sau bảng;

**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 Hocsinh

ORDER BY Hoten DESC, age ASC;

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

**Cách 1: với 2 cột firstName và lastName**

SELECT s.firstName, s.lastName

FROM student as s

WHERE s.lastName = "Le" AND s.lastName = "Hai";

**Cách 2: với cột fullName**

SELECT \*

FROM users

WHERE SUBSTRING\_INDEX(SUBSTRING\_INDEX(name, ' ', 1), ' ', -1) = "Le"

AND SUBSTRING\_INDEX(SUBSTRING\_INDEX(name, ' ', -1), ' ', -1) = "Hai";

Ngắn hơn

SELECT \*

FROM users

WHERE SUBSTRING\_INDEX(fullname, ' ', 1) = "Le"

AND SUBSTRING\_INDEX(fullname, ' ', -1) = "Hai";

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

SELECT \* from student

WHERE StudentName LIKE ‘C%’;

**53. Union là gì? Phân biệt Union và Union All?**

*UNION: Dùng để gộp kết quả của 2 hay nhiều câu lệnh truy vấn dữ liệu lại với nhau.*

* *UNION: lấy giá trị 1 lần duy nhất(không có trường hợp giá trị trùng nhau).*
* *UNION ALL: lấy hết tất cả các giá trị.*

**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 là một cấu trúc mạnh mẽ của MySQL, ang để ang tốc độ phản hồi của các truy vấn thống thường lên gấp nhiều lần.

- Việc lập chỉ mục (index) trên các cột sẽ giúp cải thiện tốc độ truy vấn dữ liệu từ cơ sở dữ liệu.(tương tự như mục lục của cuốn sách)

- Nếu không có chỉ mục, các truy vấn phải rà soát từng ang để tìm kiếm dữ liệu đích đó.

**\* Nên tạo chỉ mục cho các cột khi:**

- Bảng chứa dữ liệu lớn.

*- Bảng có ít hoạt động update, insert*

- Cột sử dụng để tìm kiếm thường xuyên.

- Cột sử dụng để sắp xếp dữ liệu.

- Dữ liệu trong cột có sự phân biệt cao.

**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?**

- MySQL cung cấp 3 kiểu index khác nhau cho data đó là B-Tree index, Hash index và R-Tree index.

|  |  |
| --- | --- |
| **B-Tree Index** | **Hash Index** |
| - Dữ liệu index trong B-Tree được tổ chức và lưu trữ theo dạng tree, tức là có root, branch, leaf. Giá trị của các node được tổ chức tăng dần từ trái qua phải.  - B-Tree index được sử dụng trong các biểu thức so sánh dạng: =, >, >=, <, <=, BETWEEN và LIKE.  - Khi truy vấn dữ liệu thì việc tìm kiếm trong B-Tree là 1 quá trình đệ quy, bắt đầu từ root node và tìm kiếm tới branch và leaf, đến khi tìm được tất cả dữ liệu - thỏa mãn với điều kiện truy vấn thì mới dùng lại.  - B-Tree index được sử dụng cho những column trong bảng khi muốn tìm kiếm 1 giá trị nằm trong khoảng nào đó. Ví dụ: tìm kiếm những sinh viên có điểm Toán từ 5-9 | - Dữ liệu index được tổ chức theo dạng Key - Value được liên kết với nhau.  - Hash index chỉ nên sử dụng trong các biểu thức toán tử là = và . Không sử dụng cho toán từ tìm kiếm 1 khoảng giá trị như > hay <  - Không thể tối ưu hóa toán tử ORDER BY bằng việc sử dụng Hash index bởi vì nó không thể tìm kiếm được phần từ tiếp theo trong Order.  - Toàn bộ nội dung của Key được sử dụng để tìm kiếm giá trị records, khác với B-Tree một phần của node cũng có thể được sử dụng để tìm kiếm.  - Hash có tốc độ nhanh hơn kiểu Btree. |

- Chỉ mục duy nhất: **UNIQUE INDEX**

+ Để thực thi tính duy nhất của các giá trị trong một hoặc nhiều cột.

+ Có thể tạo nhiều hơn một chỉ mục duy nhất trong một bảng, điều này không thể thực hiện được với ràng buộc khóa chính.

- Một chỉ mục duy nhất, không cho phép chèn bất kỳ giá trị trùng lặp nào được chèn vào bảng, nên không thể đánh chỉ mục duy nhất lên một cột, trong đó dữ liệu của cột không đảm bảo tính duy nhất

**56. Trình bày về khung nhìn (view)? Phân biệt view và table?**

- Một khung nhìn VIEW là một bảng ảo trong cơ sở dữ liệu có nội dung được định nghĩa thông qua một câu lệnh SQL nào đó. Một VIEW bao gồm các hàng và cột giống như một bảng thực. Các trường trong một khung nhìn là các trường từ một hoặc nhiều bảng thực trong Database.

|  |  |
| --- | --- |
| **VIEW** | **TABLE** |
| - Một bảng ảo có nội dung được xác định bởi một truy vấn.  - Không được xem là một cấu trúc lưu trữ dữ liệu tồn tại trong cơ sở dữ liệu | - Bảng là nơi lưu trữ sơ bộ để lưu trữ dữ liệu và thông tin trong RDBMS.  - Bảng là một tập hợp các mục dữ liệu có liên quan và nó bao gồm các cột và hàng. |

- Điểm khác biệt giữa khung nhìn VIEW và bảng: là VIEW không được xem là một cấu trúc lưu trữ dữ liệu tồn tại trong cơ sở dữ liệu. Thực chất dữ liệu quan sát được trong VIEW được lấy từ các bảng thông qua câu lệnh truy vấn dữ liệu và được sử dụng để hạn chế truy cập cơ sở dữ liệu từ dữ liệu gốc, hoặc để ẩn dữ liệu phức tạp, lấy dữ liệu nhanh hơn.

**57. Cú pháp tạo view?**

CREATE VIEW view\_name AS

SELECT column1, column2.....

FROM table\_name

WHERE condition;

**58. Có thể tạo view từ view khác không?**

**Có.** VIEW được tạo ra bởi **câu lệnh CREATE VIEW**. Các VIEW có thể được tạo từ một bảng đơn, nhiều bảng hoặc từ VIEW khác.



**59. Trình bày về thủ tục lưu trữ (stored procedure)?**

- Store Procedure (thủ tục lưu trữ), có thể được định nghĩa là chương trình con, giống như một chương trình con được lưu trữ trong cơ sở dữ liệu.

- Là tập hợp các câu lệnh được xem như là một khối lệnh nhằm thực hiện một tác vụ cụ thể.

- Stored Procedure được tạo ra nhằm thực hiện các lệnh của mysql theo một nhóm việc cụ thể thay vì thực hiện từng thao tác (insert, update, delete).

- Cú pháp khai báo:

DELIMITER //

CREATE PROCEDURE [procedure\_name] ([param1, param2,…])

BEGIN

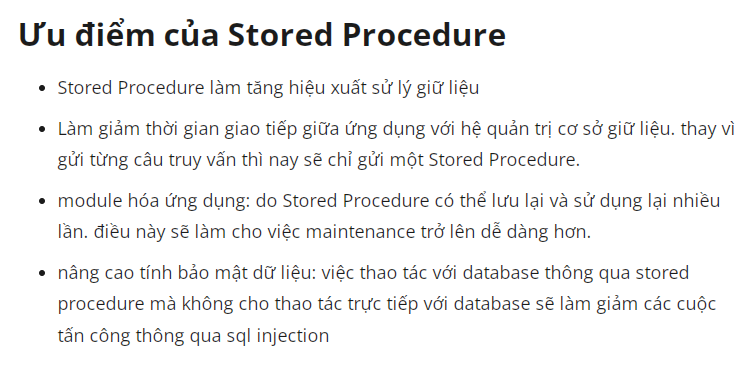
[sql\_statements]

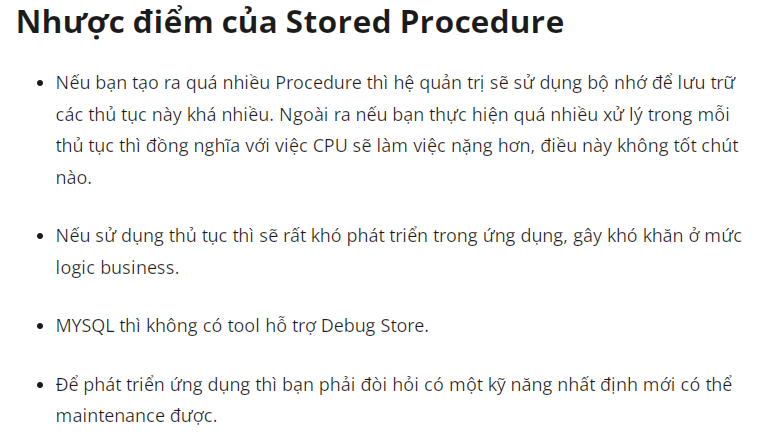
END //

DELIMITER ;

- Gọi stored procedure đã được khai báo:

CALL procedure\_name ([param1, param2,…]);





**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?**

CREATE PROCEDURE Diem\_student (

IN MSV\_input int,

OUT `message` VARCHAR(50)

)

BEGIN

SELECT MSV FROM students WHERE MSV\_input = MSV

SET message = "Sinh viên đó là “ + MSV\_input);

END //

DELIMITER ;

Call Diem\_student (2);

**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?**

CREATE PROCEDURE Dem\_student (

IN Dem int,

Diem int,

OUT `message` VARCHAR(50)

)

BEGIN

SELECT Diem = COUNT(\*) FROM students WHERE student\_id = dem\_id

SET message = " Tổng số nhân viên: “ + Dem);

END //

DELIMITER ;

Call Dem\_student (5);

**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”**

DELIMITER //

CREATE PROCEDURE sp\_example(

IN `a` INT,

OUT `message` VARCHAR(50)

)

BEGIN

IF(a > 0 )

THEN

SET message = "a la so duong";

ELSEIF (a < 0)

THEN

SET message = "a la so am";

ELSE

SET message = " a bang 0";

END IF;

END //

DELIMITER ;

**63. Css là gì? Các cách nhúng CSS vào tài liệu html?**

- CSS là viết tắt của cụm từ Cascading Style Sheets

- Là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web

- CSS mô tả cách các phần tử HTML sẽ hiển thị trên màn hình, trang web, hoặc các phương tiện khác.

**Cách nhúng CSS:**

• Khai báo trực tiếp ở thẻ HTML (Inline style)

• Khai báo trong thẻ <style> của tài liệu HTML (Internal style)

• Khai báo trong file .css riêng biệt (External style)

**64. Trình bày về các loại bộ chọn (selector) trong CSS?**

- Bộ chọn tên của các phần tử (thẻ) HTML: p {… ;}, div {… ;}

- Bộ chọn id: #name\_ID {… ;}

- Bộ chọn class: .name\_class {… ;}

- Bộ chọn thuộc tính (Sử dụng bất kì thuộc tính nào của các thành phần):

Input[type-button]{… ;}

=> Các selectors có độ ưu tiên như sau: id -> class -> tag.

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

- Bộ chọn bằng id: #tenid

- Bộ chọn bằng class: .tenclass

- Độ ưu tiên của bộ chọn bằng id > class

**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?**

- Thuộc tính margin giúp căn lề cho các phần tử html.

- Thuộc tính margin là cách dùng tắt cho các thuộc tính margin-top, margin-bottom, margin-left và margin-right.

**68. Phân biệt margin, padding và border?**

- margin: cho phép chúng ta tạo ra các khoảng cách từ thành phần được chọn ra phía bên ngoài của thành phần đó.

- padding: là khoảng trống nằm giữa nội dung và viền.

- border: được sử dụng để thiết lập đường viền trên một phần tử.

**69. Thuộc tính position dùng để làm gì? Phân biệt các giá trị: static, absolute, relative và fixed?**

- Thuộc tính position của CSS chỉ định phương thức định vị vị trí sẽ áp dụng cho một thẻ HTML.

- Các thẻ sau đó được định vị bằng cách sử dụng các thuộc tính top, bottom, left, right và chỉ làm việc khi thuộc tính position được thiết lập trước.

**Phân biệt các giá trị:**

- Static: thuộc tính mặc định của thẻ HTML, nó sẽ không bị ảnh hưởng bởi các thuộc tính top, bottom, left, right và được đinh vị theo luồng bình thường của trang.

- Absolute: được định vị tương đối với phần tử so với thằng cha gần nhất, ngoại trừ static.

- Relative: được định vị tương đối so với vị trí bình thường của bản thân nó.

- Fixed: được định vị tương đối so với chế độ xem (viewport), có nghĩa là nó luôn luôn ở cùng một vị trí ngay cả khi trang được cuộn.

**70. Phân biệt clear và float?**

- 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.

- 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.

**71. Tạo bố cục trang trong html5 với các thẻ cơ bản.**

|  |
| --- |
| **72. Bạn có biết Prototype? Trình bày?**  *Prototype là cơ chế mà các object trong javascript kế thừa các tính năng từ một object khác. Tất cả các object trong javascript đều có một prototype, và các object này kế thừa các thuộc tính (properties) cũng như phương thức (methods) từ prototype của mình.*   * *Thuộc tính: constructor,…* * *Phướng thức:* valueOf, toString() |
| **73. Responsive Web Design là gì?**  *Là kỹ thuật được sử dụng để trang web có thể hiển thị tốt trên nhiều loại thiết bị với kích thước của trình duyệt khác nhau.* |
| **74. Trình bày cú pháp RWD Media Queries?**  *Media Queries sử dụng @media để áp dụng một khối thuộc tính css chỉ khi thoả mãn một điều kiện nào đó.* |

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

- Static web (web tĩnh): **Web tĩnh tức là website cố định về nội dung, không có hoặc hạn chế về sự tương tác với website, chủ yếu sử dụng bằng mã nguồn HTML, JavaScript hoặc Flash**. Nội dung cố định sau khi được thiết kế. Thường không có trang quản trị để thay đổi nội dung, cập nhật bài viết. Không có sự tương tác ngược từ người xem vào nội dung trang.

- Dynamic web (web động): **thường có trang quản trị, để thay đổi, cập nhật nội dung trên site. Sử dụng cơ sở dữ liệu để quản lý các hạng mục trên** website (bài viết, thành viên, các menu...). Người dùng có thể tương tác dễ dàng đến nội dung site. Được ứng dụng vào rất nhiều lĩnh vực: giới thiệu sản phẩm, công ty, bán hang, tin tức…

**76. Giao thức (protocol) là gì? Kể tên một số giao thức bạn biết?**

- Protocol là một tiêu chuẩn được sử dụng để định nghĩa một phương thức trao đổi dữ liệu qua mạng máy tính, như là local area network (LAN), Internet,…

- Mỗi protocol có những phương thức định dạng dữ liệu riêng khi được gửi và cần phải làm những gì một khi nó nhận dữ liệu, các dữ liệu được nén lại hay cách kiểm tra lỗi dữ liệu**.**

- Một số giao thức: Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol over SSL/TLS (HTTPs), Internet Protocol Suite…

**77. HTTP là gì? Phân biệt HTTP và HTTPs?**

- HTTP (Hypertext Transfer Protocol = Giao thức truyền tải siêu văn bản): là giao thức ở tầng ứng dụng (Application-level) cho các hệ thống thông tin cộng tác, phân tán, siêu dữ liệu

=> Đây là giao thức truyền tải dữ liệu được sử dụng để thiết lập giao tiếp giữa máy chủ cung cấp dịch vụ web và máy khách sử dụng dịch vụ web.

|  |  |
| --- | --- |
| **HTTP** | **HTTPs** |
| - Là văn bản có cấu trúc sủ dụng các siêu liên kết giữa các node chứa văn bản. | - Sử dụng với HTTP để cung cấp các dịch vụ tương tự, kết nối bảo mật được cung cấp bởi SSL hoặc TLS  **Tham khảo:**  *(SSL là chữ viết tắt của Secure Sockets Layer (Lớp socket bảo mật). Một loại bảo mật giúp mã hóa liên lạc giữa website và trình duyệt. Công nghệ này đang lỗi thời và được thay thế hoàn toàn bởi TLS.*  *TLS là chữ viết tắt của Transport Layer Security, nó cũng giúp bảo mật thông tin truyền giống như SSL. Nhưng vì SSL không còn được phát triển nữa, nên TLS mới là thuật ngữ đúng nên dùng.)* |

**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.

### Cách thức hoạt động của mô hình TCP/IP

[IP](https://www.totolink.vn/article/74-ip-la-gi-cach-xac-dinh-dia-chi-ip-tren-may-tinh.html) (Giao thức liên mạng) cho phép các gói tin được gửi đến đích đã định sẵn, bằng cách thêm các thông tin dẫn đường vào các gói tin để các gói tin được đến đúng đích đã định sẵn ban đầu.

TCP (Giao thức truyền vận) đóng vai trò kiểm tra và đảm bảo sự an toàn cho mỗi gói tin khi đi qua mỗi trạm. Trong quá trình này, nếu giao thức TCP nhận thấy gói tin bị lỗi, một tín hiệu sẽ được truyền đi và yêu cầu hệ thống gửi lại một gói tin khác. Quá trình hoạt động này sẽ được làm rõ hơn ở chức năng của mỗi tầng trong mô hình TCP/IP.

**79. Trình bày mô hình Client/Server?**

- Mô hình mạng Client/Server là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như các máy khách, chúng làm nhiệm vụ gửi các yêu cầu đến máy chủ

=> để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.

**80. Máy chủ (Server) là gì?**

- Là một máy tính được kết nối với mạng máy tính hoặc internet, có IP tĩnh, có tính năng xử lý cao

- Người ta cài đặt các phần mềm để phục vụ cho các máy tính khác để truy cập yêu cầu cung cấp các dịch vụ và tài nguyên.

**81. Phân biệt máy chủ web và máy chủ ứng dụng?**

|  |  |
| --- | --- |
| **Máy chủ web** | **Máy chủ ứng dụng** |
| - Hữu ích phù hợp với nội dung tĩnh; tiêu thụ ít tài nguyên; sắp xếp môi trường chạy cho các ứng dụng web  - Đa luồng không được hỗ trợ; công suất thấp hơn máy chủ ứng dụng; giao thức HTTP và HTML được sử dụng | - Được trang bị cho nội dung động; sử dụng nhiều tài nguyên; sắp xếp môi trường chạy ứng dụng cho các doanh nghiệp;  - Đa luồng được hỗ trợ; công suất cao hơn máy chủ web; giao thức HTTP và RPC/RMI được sử dụng. |

**82. Tomcat là gì? Tomcat thuộc Web Server hay Application Server?**

- Tomcat là một **web server HTTP** được phát triển bởi Apache Software Foundation, *hỗ trợ mạnh cho ứng dụng Java thay vì website tĩnh.*

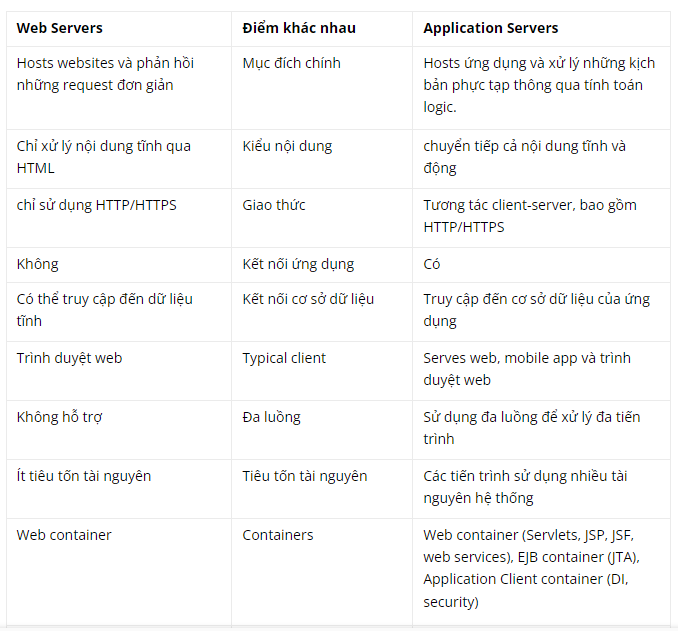
=> Do đó, nó có thể chạy nhiều bản Java chuyên biệt như Java Servlet, JavaServer Pages (JSP), Java EL, và WebSocket.

**[Mở rộng: Application server là gì?]**

Application server là từ tiếng Anh của Phần mềm máy chủ ứng dụng. Đây là một phần mềm (software engine) trong cung cấp ứng dụng phần mềm cho các máy trạm hoặc thiết bị, thông thường là qua mạng Internet sử dụng giao thức HTML.

Hiểu đơn giản, Application server có thể được coi như một khuôn khổ phần mềm, cung cấp môi trường nơi có nhiều ứng dụng có thể chạy bất kể chúng là gì.

Application server (Hệ điều hành máy chủ (OS), Phần cứng máy chủ) khác biệt với Web [server](https://wiki.tino.org/server-la-gi/) qua việc dùng rất nhiều nội dung động do máy chủ tạo ra và tích hợp chặt chẽ tới[Database server.](https://wiki.tino.org/database-server-la-gi/)



**Web Server là gì?**

Web server là một hệ thống máy tính dùng để lưu trữ, xử lý và phân phối nội dung website đến client. Theo mặc định, nó chỉ xử lý nội dung HTML tĩnh sau:

* Documents
* Images
* Videos
* Fonts

Web server chỉ chấp nhận request qua giao thức HTTP/HTTPS. Luồng xử lý request căn bản như hình bên dưới.

**Application Server là gì?**

Application server là một framework chịu trách nhiệm cung cấp nội dung động của website cho client.

Application server cung cấp cho client khả năng tương tác với hệ thống bằng những nghiệp vụ logic. Từ đó, application server sẽ chuyển đổi dữ liệu này thành nội dung động và thực thi các functions. Nội dung động có thể bao gồm:

* Kết quả phiên giao dịch
* Hỗ trợ quyết định
* Phân tích thời gian thực

Một số tác vụ chính của application server bao gồm:

* Transaction management
* Security
* Dependency injection (DI)
* Concurrency

Luồng xử lý request căn bản của application server như hình bên dưới.

**Top những server phổ biến**

* **Web Server**: LiteSpeed, Nginx, Apache, IIS
* **Application Server**: Apache Tomcat, Oracle WebLogic, JBoss

Máy chủ (Server) là một máy tính được kết nối với mạng máy tính hoặc Internet, có IP tĩnh, có năng lực xử lý cao. Trên đó người ta cài đặt các phần mềm để phục vụ cho các máy tính khác truy cập để yêu cầu cung cấp các dịch vụ và tài nguyên.

### ****2. Các loại máy chủ (Server)****

Theo **phương pháp xây dựng một hệ thống máy chủ**, máy chủ chia làm 3 loại:

- **Máy chủ vật lý riêng**(Dedicated Server): Là máy chủ chạy trên **phần cứng** và các thiết bị hỗ trợ riêng biệt gồm: [HDD](https://www.thegioididong.com/hoi-dap/hdd-la-gi-922791), [CPU](https://www.thegioididong.com/hoi-dap/cpu-la-gi-1039844), [RAM](https://www.thegioididong.com/hoi-dap/ram-lap-top-la-gi-dung-luong-bao-nhieu-la-du-1172167), [Card mạng](https://vi.wikipedia.org/wiki/Card_m%E1%BA%A1ng).

- **Máy chủ ảo (VPS)**: Là dạng máy chủ được tạo thành bằng phương pháp sử dụng **công nghệ ảo hóa** để chia tách từ một máy chủ vật lý riêng thành nhiều máy chủ ảo khác nhau. Các máy chủ ảo có tính năng tương tự như máy chủ vật lý đó và chia sẻ tài nguyên từ máy chủ.

- **Máy chủ đám mây (Cloud Server)**: Là máy chủ được kết hợp từ nhiều máy chủ vật lý khác nhau cùng với hệ thống lưu trữ [SAN](https://wiki.edu.vn/wiki/index.php?title=M%E1%BA%A1ng_l%C6%B0u_tr%E1%BB%AF).

Theo **chức năng**, máy chủ được chia thành các loại sau:

- **Database servers**(máy chủ cơ sở dữ liệu).

- **File servers** (máy chủ file, là máy chủ lưu trữ file ví dụ như [Dropbox](https://www.thegioididong.com/hoi-dap/huong-dan-su-dung-dropbox-917798), [Google Drive](https://www.thegioididong.com/hoi-dap/google-drive-la-gi-tinh-nang-va-cach-su-dung-googl-917829), [Microsoft One Drive](https://www.microsoft.com/vi-vn/microsoft-365/onedrive/online-cloud-storage)).

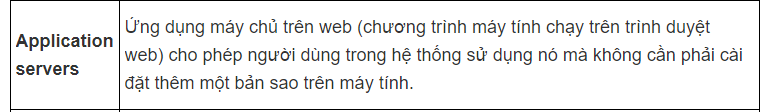
- **Mail servers** (máy chủ mail ví dụ như Gmail, Yahoo mail, Yandex, Amazon mail service).

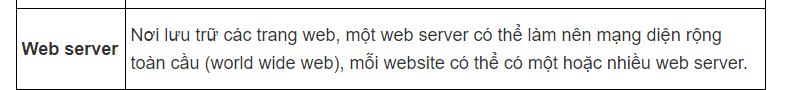
- **Print servers** (máy chủ in, thường được dùng trong mạng nhỏ của doanh nghiệp).

- **Web servers** (máy chủ web để phục vụ người dùng mua hàng như các site Amazon, Taobao, Google shopping, phục vụ người dùng đọc tin tức,...).

- **Game servers** (máy chủ trò chơi ví dụ máy chủ phục vụ game Võ Lâm, [World of Warcraft](http://vi.wikipedia.org/wiki/World_of_Warcraft), Tru tiên,...).

- **Application servers** (máy chủ ứng dụng ví dụ để chạy các phần mềm quản lý [ERP](https://vi.wikipedia.org/wiki/Ho%E1%BA%A1ch_%C4%91%E1%BB%8Bnh_t%C3%A0i_nguy%C3%AAn_doanh_nghi%E1%BB%87p), phần mềm [CRM](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_quan_h%E1%BB%87_kh%C3%A1ch_h%C3%A0ng) trong doanh nghiệp, nhưng Application Server cũng có thể được hiểu chung là máy chủ cung cấp dịch vụ web, mail, file server, database,...).





**83. Phân biệt HTTP Request và HTTP Response?**

- HTTP Request: Đây là thông tin từ client gửi đến server, yêu cầu server làm một việc gì đó (Phương thức GET và POST)

- HTTP Response: là kết quả dữ liệu mà server trả về cho client. (HTTP Status Codes - mã kết quả trả về)

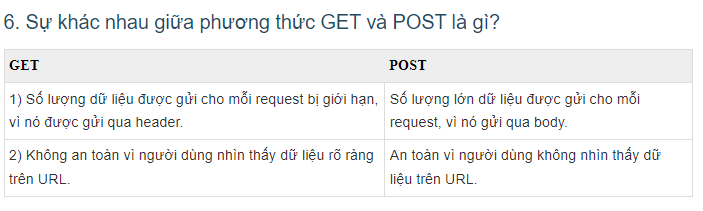
**84. Trình bày Content Type hay MIME (Multipurpose Internet Mail Extension) Type?**

Là tiêu chuẩn giúp phân loại các loại tập tin được sử dụng trên Internet, đặc biệt là dùng để định dạng cho thư điện tử.

MIME – Viết tắt của cụm từ “Multipurpose Internet Mail Extensions” có nghĩa là giao thức mở rộng thư điện tử Internet đa mục đích được mô tả trong RFC 1524 và các RFC khác như: RFC 2045, RFC 2046, RFC 2047, RFC 2231.

Loại MIME (MIME Type) là tiêu chuẩn giúp phân loại các loại tập tin được sử dụng trên Internet, đặc biệt là dùng để định dạng cho thư điện tử. Nói một cách đơn giản, lúc mới ra đời, giao thức gửi mail chỉ hỗ trợ gửi plaintext, MIME là phần mở rộng cho phép email có thể gửi/nhận các file có định dạng khác (HTML, video, nhạc, …). Hầu như mọi thư điện tử Internet được truyền qua giao thức SMTP theo định dạng MIME.

**85. Phân biệt rõ hai phương thức Get và Post?**



|  |  |
| --- | --- |
| **GET** | **POST** |
| - Có thể được cached (lưu vào bộ nhớ đệm), bookmark (dấu trang) và lưu trong lịch sử của trình duyệt.  - Bị giới hạn về chiều dài, do chiều dài của URL là có hạn.  - Không nên dùng với dữ liệu quan trọng, chỉ dùng để nhận dữ liệu. | - Không thể cached, bookmark hay lưu trong lịch sử trình duyệt.  - Không bị giới hạn về độ dài. |

**86. Trình bày lớp HTTPServlet?**

**- HttpServlet là một trong những đối tượng của Jakarta EE Servlet giúp chúng ta có thể xây dựng các ứng dụng chạy trên nền web với Java**

**- HttpServlet thì nó là một abstract class nằm trong package jakarta.servlet.http của Jakarta EE Servlet API.**

**- Nó extends từ một abstract class khác tên là GenericServlet và class GenericServlet này là đối tượng implement interface Servlet, interface chính để định nghĩa servlet của Jakarta EE Servlet API**.

*- Sử dụng đối tượng HttpServlet này, chúng ta có thể xây dựng các ứng dụng Java web để handle các request từ client với các HTTP method như GET, POST, HEAD, PUT, DELETE, TRACE và OPTIONS.*

*- Khi client send một request tới web server container, method service() trong đối tượng HttpServlet sẽ process request này.*

*- Phụ thuộc vào HTTP method mà client gửi, việc xử lý tiếp theo sẽ do các method doGet(), doPost(), doHead(), doPut(), doDelete(), doTrace() và doOptions() của đối tượng này đảm nhận.*

*- Chúng ta có thể override lại các method doGet(), doPost(), doHead(), doPut(), doDelete(), doTrace() hoặc doOptions() để process request theo ý chúng ta muốn*

**87. Trình bày Servlet Container?**

Là một phần của web server được viết bởi ngôn ngữ java, tương tác với Servlet để xử lý các web động, yêu cầu từ client.

Servlet container thực hiện các tác vụ:

- Quản lý vòng đời.

- Hỗ trợ xử lý đa luồng.

- Object Pooling

- Bảo mật…

**88. Vòng đời của Servlet?**

Là khi Servlet đó được tạo ra cho đến khi bị xoá bỏ.

Vòng đời Servlet được quản lý bởi Servlet Container, có 5 giai đoạn:

- Nạp lớp Servlet vào bộ nhớ. (load(): khi nhập url partent lên chổ locahoat ,enter cái là gửi request về LÀ LOAD() dược chạy

- Khởi tạo đối tượng.

- Gọi phương thức init() và truyền vào đối tượng ServletConfig.

- Thực thi: gọi phương thức service() và truyền đối tượng req, resp.

- Xoá bỏ: gọi phương thức destroy() để xoá bỏ.

**89. Trình bày Servlet Interface? Mô tả các phương thức init(), service(), destroy()?**

Giao diện Servlet Interface cung cấp hành vi cơ bản của Servlet và cần phải triển khai đầy đủ trước khi có thể tạo ra bất ký Servlet nào.

Mô tả các phương thức:

- Init(): khởi chạy servlet. Nó cung cấp vòng đời và được gọi lên bởi web container chỉ 1 lần duy nhất.

- Service(): cung cấp sự phẩn hồi tới yêu cầu được gửi đến.

- Destroy(): gọi một lần duy nhất và chỉ ra servlet bị huỷ.

**90. Phân biệt các loại Servlet Container: Standalone, In-process và Out-of-Servlet?**

- Standelone: là kiểu máy chủ dựa trên java điển hình, trong đó servlet container và máy chủ web được tích hợp trong một chương trình duy nhất. vd: Tomcat

- In-process: servlet được tách riêng ra khỏi máy chủ web.

- Out-of-process: máy chủ web và servlet container là các chương trình khác nhau chạy trong các tiến trình riêng biệt.

**91. Giới thiệu một số phương thức của lớp HttpServlet?**

GET, POST, HEAD, PUT, DELETE, TRACE và OPTIONS.

**92. Chỉ ra nhiệm vụ của hai phương thức:**

- public void service(ServletRequest req, ServletResponse res): chuyển tiếp yêu cầu tới phương thức service() bằng cách chuyển đổi yêu cầu và phản hồi thành những đối tượng HTTP.

- protected void service(HttpServletRequest req, HttpServletResponse res): nhận yêu cầu từ phương thức service và chuyển tiếp yêu cầu đó tới phương thức doGet hoặc doPost tuỳ thuộc vào phương thức của yêu cầu HTTP gưuỉ tới.

**93. Phân biệt doGet() và doPost()?**

- doGet(): xử lý yêu cầu GET, được sử dụng khi gửi tới một lượng nhỏ dữ liệu không nhạy cảm.

- doPost(): xử lý yêu cầu Post, được sử dụng khi gửi tới một lượng nhỏ dữ liệu nhạy cảm. VD: id đăng nhập và mật khẩu.

**94. RequestDispatcher được dùng làm gì?**

- Là một interface, thực hiện xác định một đối tượng có thể gửi yêu cầu tới bất kỳ tài nguyên nào (như HTML, JSP, Servlet) trên server.

- Phân phối các request từ client đến các nguồn tài nguyên khác nhau trên server và trả về response

**95. Phân biệt forward() của RequestDispatcher và SendRedirect()**

|  |  |
| --- | --- |
| **Forward()** | **SendRedirect()** |
| - gửi yêu cầu tương tự tới một tài nguyên khác;  - hoạt động ở phía server và hoạt động bên trong server. | - luôn gửi yêu cầu mới vì nó sử dụng thanh URL của trình duyệt;  - hoạt động ở phía client và hoạt động bên trong và bên ngoài server. |

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

- JSP (JavaServer Pages) cho phép người dùng tạo ra các trang web động.

- JSP giúp các nhà phát triển chèn java code vào các trang HTML bằng cách sử dụng các thẻ JSP đặc biệt

Đúng, vì: về bản chat thì JSP cũng chính là Servlet. Trong quá trình biên dịch JSP sẽ chuyển thành Servlet rồi Servlet này sẽ được biên dịch.

**97. So sánh giống và khác nhau giữa Jsp và Servlet?**

- Giống nhau:

+ đều nằm ở phía server

+ đều xử lý dữ liệu động

+ đều chạy web container

+ đều là những công nghệ quan trọng trong java EE.

- Khác nhau:

|  |  |
| --- | --- |
| JSP | Servlet |
| - mạnh về xử lý giao diện  - mã java nằm trong mã HTML. | - mạnh về xử lý nghiệp vụ  - mã HTML nằm trong mã java |

**98. Vòng đời của Jsp?**

Vòng đời của một JSP được tính từ khi JSP đó được tạo ra cho đến khi bị hủy bỏ.

Gồm 4 giai đoạn:

- Biên dịch: gồm 3 bước: phân tích trang JSP, chuyển trang JSP thành Servlet, biên dịch Servlet.

- Khởi tạo: gọi phương thức jspinit()

- Thực thi: gọi phương thức \_jspService()

- Xoá bỏ: gọi phương thức jspDestroy()

**99. Jstl là gì? Giới thiệu một số thẻ bạn đã sử dụng jstl?**

- JSP Standard Tag Library (JSTL) là thư viện (hỗ trợ code java trong JSP) 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.

- Các thẻ cơ bản(Core Tags), các thẻ SQL(SQL Tags), các thẻ định dạng và địa phương hoá (Formartting and Localization Tags)

**100. 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, …

MVC chia ứng dụng phần mềm ra thành 3 phần có tương tác với nhau:

• Model: thường là các class chứa thông tin đối tượng (dữ liệu), tương tác truy xuất database.

• View: là nơi nhận dữ liệu từ model, database và sắp xếp chúng chính xác nhờ yêu cầu từ controller và truyền về client.

• Controller: Đóng vai trò trung gian giữa Model và View, nó có nhiệm vụ nhận yêu cầu từ client sau đó xử lý request, load model và gửi data qua view tương ứng với nhau rồi trả kết quả về cho client.

**101.Phân biệt kiến trúc ba tầng (đa tầng hay n- tầng) với kiến trúc MVC?**

|  |  |
| --- | --- |
| **Kiến trúc ba tầng** | **MVC** |
| - tầng khách không bao giờ giao tiếp trực tiếp với tầng dữ liệu  - Trong mô hình ba tầng, tất cả giao tiếp phải đi qua tầng giữa  - kiến ​​trúc ba tầng là tuyến tính | Dạng hình tam giác:  - khung nhìn gửi các bản cập nhật cho bộ điều khiển  - bộ điều khiển cập nhật mô hình  - khung nhìn được cập nhật trực tiếp từ mô hình |
| - mọi thứ đều đi qua tầng logic (a-> b, b-> c và c-> b, b-> a) | - giao tiếp 2 trên 2, theo cách hình tam giác. (a-> b, b-> c, c-> a) |

**102. Giới thiệu một ứng dụng tổ chức kiến trúc MVC?**

*Giả sử bạn đi đến một nhà hàng. Bạn sẽ không phải vào bếp và chuẩn bị thức ăn mà bạn chắc chắn có thể làm ở nhà của mình. Thay vào đó, bạn chỉ cần đến đó và đợi người phục vụ đến.*

*Bây giờ người phục vụ đến với bạn, và bạn chỉ việc gọi món ăn. Người phục vụ không biết bạn là ai và bạn muốn gì, anh ta chỉ ghi chi tiết đơn hàng thức ăn của bạn.*

*Sau đó, người phục vụ di chuyển đến nhà bếp. Trong nhà bếp, người phục vụ không chuẩn bị thức ăn cho bạn.*

*Đầu bếp chuẩn bị thức ăn cho bạn. Người phục vụ sẽ đưa món của bạn cho anh ta cùng với số bàn của bạn.*

*Nấu thức ăn sau đó chuẩn bị cho bạn. Anh ấy sử dụng các nguyên liệu để nấu món ăn. Hãy giả sử rằng bạn đặt một bánh sandwich rau. Sau đó, anh ta cần bánh mì, cà chua, khoai tây, ớt chuông, hành tây, một chút, pho mát, v.v. mà anh ta lấy từ tủ lạnh.*

*Nấu xong cuối cùng giao đồ ăn cho người phục vụ. Bây giờ công việc của người phục vụ là chuyển thực phẩm này ra ngoài nhà bếp.*

*Bây giờ người phục vụ biết bạn đã đặt món ăn nào và chúng được phục vụ như thế nào.*

**Sau khi xem ví dụ này ta thấy được :**

View: Chính là bạn.

Controller: Là người phục vụ.

Model: Là đầu bếp.

Database: là tủ lạnh.

@Mọi người có thể xem kĩ trong phần demo ứng dụng (Quản lý danh sách khách hàng) của SLIDE 11 về bài MVC.

**103. Trình bày ưu và nhược điểm của kiến trúc MVC?**

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| - 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. | 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. |

**104 Sự giống, khác nhau và ưu nhược điểm của MVC1 & MVC2?**

|  |  |
| --- | --- |
| **MVC1** | **MVC2** |
| controller và view đều là Jsp | Controller là các Servlet còn Model là các class của Java, view là các trang JSP |
| là thế hệ đầu tiên sử dụng JSP/ JavaBeans để thực hiện mô hình MVC | là một mô hình mô tả kiến trúc MVC trên nền web do Sun phát minh |
| 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 | 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 |
| không | tốt cho các dự án lớn, có thể dễ dàng tái sử dụng, làm việc hiệu quả và logic hơn |

**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.

### (Chứa các đặc tả về Interface - kết nối ví dụ: Connection, applition, driv manager, statement và preparedstatement (là 2 interface cung cấp câu lênh thực hiện với database)

### - 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, ...

### ****Ưu điểm****

1. Giao tiếp hai chiều phải được xác nhận trong các giao dịch sử dụng API. Cũng chính vì vậy mà các thông tin rất đáng tin cậy.

2. API là công cụ mã nguồn mở, có thể kết nối mọi lúc nhờ vào Internet.

1. Hỗ trợ chức năng RESTful một cách đầy đủ.
2. Cấu hình đơn giản khi được so sánh với WCF (Window Communication Foundation). Cung cấp cấp trải nghiệm thân thiện với người dùng.

### ****Nhược điểm****

* Tốn nhiều chi phí phát triển, vận hành, chỉnh sửa.
* Đòi hỏi kiến thức chuyên sâu.
* Có thể gặp vấn đề bảo mật khi bị tấn công hệ thống.

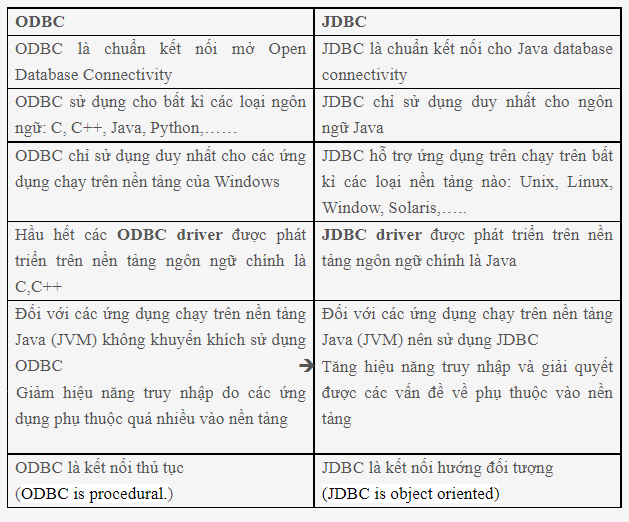
**106. JDBC là gì? Phân biệt JDBC và ODBC?**

- Java JDBC là một API được sử dụng để kết nối và thực hiện truy vấn với cơ sở dữ liệu.

- JDBC API sử dụng trình điều khiển jdbc để kết nối với cơ sở dữ liệu.

- JDBC là viết tắt của Java Database Connectivity, nó là ứng dụng mã nguồn mở cho Java, **giúp ứng dụng Java thực hiện kết nối, làm việc với CSDL.**

**- Nó cho phép ta thực hiện các thao tác truy xuất, update dữ liệu với CSDL quan hệ bằng việc sử dụng các câu lệnh SQL.**

**107. Các thành phần của JDBC API? Trình bày: DriverManager, Driver, Connection, Statement, ResultSet, SQLException?**

- DriverManager: Lớp này quản lý các Database Driver. Ánh xạ các yêu cầu kết nối từ ứng dụng Java với Data driver thích hợp bởi sử dụng giao thức kết nối phụ.

- Driver: Interface này xử lý các kết nối với Database Server. Hiếm khi, bạn tương tác trực tiếp với các đối tượng Driver này. Thay vào đó, bạn sử dụng các đối tượng DriverManager để quản lý các đối tượng kiểu này.

- Connection: Đối tượng Connection biểu diễn ngữ cảnh giao tiếp. Interface này chứa nhiều phương thức đa dạng để tạo kết nối với một Database.

- Statement: Bạn sử dụng các đối tượng được tạo từ Interface này để đệ trình các lệnh SQL tới Database. Ngoài ra, một số Interface kết thừa từ nó cung chấp nhận thêm các tham số để thực thi các thủ tục đã được lưu trữ.

- ResultSet: Các đối tượng này giữ dữ liệu được thu nhận từ một Database sau khi bạn thực thi một truy vấn SQL. Nó nóng vai trò như một Iterator để cho phép bạn vọc qua dữ liệu của nó.

- SQLException: Lớp này xử lý bất cứ lỗi nào xuất hiện trong khi làm việc với Database. **108 Trình bày Statement? Phân biệt Statement, PreparedStatement và CallableStatement?**

\* Statement được dùng để truy cập vào database tuy nhiên nó không cung cấp cơ chế truyền tham số vào câu truy vấn. Do vậy Statement sẽ hữu ích khi chúng ta sử dụng các câu truy vấn tĩnh, không chứa tham số đầu vào.

- 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 đầu vào

- 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?**

PreparedStatement Interface là một interface con của Statement. Nó được sử dụng để thực thi các truy vấn được tham số hóa.

B1: String query="INSERT INTO ten\_bang values(?,?,?)";

B2: Mở kết nối với MySQL (conn = DriverManager.getConnection(DB\_URL,USER,PASS);)

B3: Thực thi truy vấn:

PreparedStatement preparedStatement = conn.prepareStatement(query);

preparedStatement .setInt(1, 6); Thiết lập giá trị cho tham số trong dấu hỏi 1

preparedStatement .setInt(2, 1); Thiết lập giá trị cho tham số trong dấu hỏi 2

preparedStatement .executeUpdate(); Thực thi

**110. Hướng dẫn sử dụng CallableStatement?**

CallableStatement được sử dụng để gọi các stored procedures và functions trong JDBC. Một đối tượng CallableStatement có thể trả về một hoặc nhiều đối tượng ResultSet. Việc sử dụng Stored Procedure giúp nâng cao hiệu suất.

Để tạo đối tượng CallableStatement trong JDBC thì:

B1: bạn tạo một stored procedure trong MySQL.

B2: Khai báo 1 String query của stored procedure đó trong java, thay các tham số IN, OUT, INOUT bằng dấu ?

B3: Mở kết nối với MySQL (conn = DriverManager.getConnection(DB\_URL,USER,PASS);)

B4: Thực thi truy vấn:

CallableStatement callableStatement= conn.prepareCall(query);

set các tham số IN, đăng ký tham số OUT

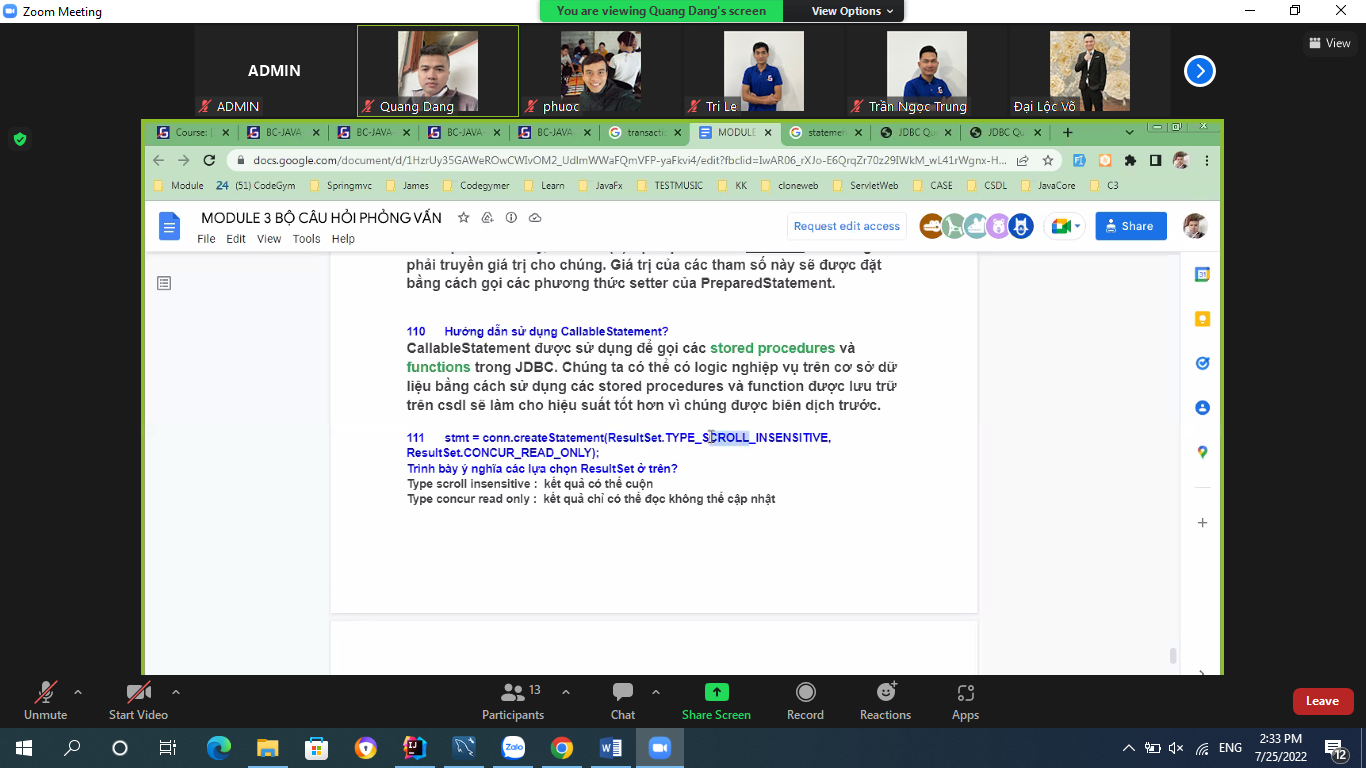
callableStatement.setInt(1, MS\_SINHVIEN);

callableStatement.registerOutParameter(2, java.sql.Types.VARCHAR);

callableStatement.execute();

**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?**

**113. Sử dụng CallableStatement trong tính năng xóa sản phẩm có id cho trước?**

**114. Gọi MySQL Stored Procedures từ JDBC như thế nào?**

- Sử dụng đối tượng CallableStatement

// đối tượng này được kế thừa từ đối tượng PreparedStatement. Nói chung cú pháp để gọi đến một thủ tục như sau: {?= call procedure\_name(param1,param2,...)}

-> Truyển giá trị dấu hỏi theo thứ tự: Sau đó lấy giá trị nó ra.

**115. Transaction là gì? Ví dụ? Mô tả 4 thuộc tính ACID của Transaction?**

\* Transaction (giao dịch) là một tiến trình xử lý, có điểm bắt đầu và điểm kết thúc, 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.

*[Tham khảo: Transaction chính là cái chữ cái T ở trong thuật ngữ OLTP (Online transaction processing). OLTP là hệ thống phục vụ các hoạt động tương tác của ứng dụng, ghi nhận dữ liệu mới liên tục, và cho phép truy xuất nhanh các dữ liệu gần đây]*

\* ACID:

+ Atomicity (bảo toàn): tất cả thành công hoặc không.

+ Consistency (nhất quán): bảo đảm tính đồng nhất của dữ liệu.

+ Isolation (độc lập): bảo đảm rằng các transaction độc lập với nhau.

+ Durability (bền bỉ): 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 được sử dụng trong các ngữ cảnh khi có nhiều các thao tác được thực thi như một khối công việc

\* **Nhược điểm của transaction:** Đôi khi một số trường hợp do phần cứng, mạng internet mà transaction hoạt động không đúng và người ta phải ghi log lại để khắc phục điều đó.

**Cú pháp để tạo Transaction:**

- SET TRANSACTION có thể được sử dụng để khởi tạo một Database Transaction. Lệnh này được sử dụng để chỉ định các đặc tính cho Transaction đó. Ví dụ, bạn có thể chỉ định một Transaction chỉ được đọc (read only) hoặc đọc viết (read write).

Cú pháp cơ bản của lệnh SET TRANSACTION trong SQL như sau:

SET TRANSACTION [ READ WRITE | READ ONLY ];

**117. Trình bày về commit()?**

- Khi một transaction hoàn chỉnh được hoàn thành thì lệnh COMMIT phải được đưa ra để mọi sự thay đổi đều được tác động đến tất cả các table liêu quan.

**-** Đây là lệnh điều khiển Transaction được sử dụng để lưu các thay đổi gọi bởi một Transaction tới cơ sở dữ liệu

**118. Trình bày về hàm rollback()?**

– Khi một transaction hoàn chỉnh được hoàn thành thì lệnh COMMIT phải được đưa ra để mọi sự thay đổi đều được tác động đến tất cả các table liên quan.

– Nếu bị lỗi, thì lệnh ROLLBACK được dùng để đưa tất cả các table liên quan với transaction về lại trạng thái trước đó.

**119. Trình bày về SavePoint trong Transaction?**

\* Đối tượng Savepoint được sử dụng để lưu trạng thái hiện tại và có thể được khôi phục sau đó về trạng thái đó trước khi thực hiện thay đổi khi có lỗi xảy ra.

\* setSavepoint(String ten\_cua\_savepoint): Định nghĩa một savepoint mới. Phương thức này cũng trả về một đối tượng Savepoint.

\* releaseSavepoint(Savepoint ten\_cua\_savepoint): Xóa một savepoint. Phương thức này nhận một đối tượng Savepoint làm tham số.

**120. Xử lý theo mẻ Batch là gì? Ví dụ?**

- 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.

Ví dụ: Một cửa hàng Thời trang bán quần áo, ngoài việc bán trực tiếp tại cửa hàng còn thực hiện bán hàng trực tuyến trên mạng thông qua các livestream trên faceboook kéo dài 2 tiếng/ngày. Cửa hàng cài đặt một hệ thống xử lý hàng loạt sẽ lọc ra những đơn hàng đáp ứng đủ các điều kiện đặt hàng trong 1 livestream để tiến hành chốt đơn.

**121. Phân biệt Transaction và Batch? Khó quá kiếm không ra!**

**Transaction**

* Được cập nhật định kỳ khi có thông tin mới, không đồng bộ theo thời gian và được chia sẻ trong các phòng ban có liên quan.
* Là dữ liệu được sử dụng bởi Master Data, liên quan đến các giao dịch của doanh nghiệp.
* Thay đổi thường xuyên.
* Có các vấn đề liên quan đến tính logic và số lượng.
* Ví dụ:
* Giá cả
* Khuyến mãi
* Phương thức thanh toán,...

**Batch**

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 database

Theo 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.

### Batch size:

Khi dùng batch xảy ra 1 vấn đề đó là nếu bạn insert 1 lần 10000 bản ghi thì sẽ tốn rất nhiều bộ nhớ, nếu số lượng bản ghi nhiều hơn có thể hệ thống sẽ bị quá tải

–> ta lại chia nhỏ ra, ví dụ mỗi lần kết nối tới database sẽ insert 100 bản ghi (con số 100 gọi là batch size) như thế vừa đảm bảo tốc độ, vừa đảm bảo bộ nhớ