Dữ liệu: tài liệu dạng thô: văn bản, sự vật, hiện tượng, chữ, số, hình ảnh

Thông tin: giá trị nhần được khi phân tích dữ liệu

Cơ sở dữ liệu: lưu trữ dữ liệu 1 cách có tổ chức, có cấu trúc; lưu trữ và truy xuất dữ liệu tốt nhất có thể

Database: MySQL - cơ sở dữ liệu dạng quan hệ; các bảng có mối liên hệ với nhau

DBMS - Database Management System

RDBMS - Relationship Database Management System: MySQL

Primary Key: khóa chính của bảng, 1 bảng chỉ có 1 khóa chính duy nhất, không trùng lặp và không null

Foreign Key: khóa ngoại, thường là khóa chính của 1 bảng khác có liên kết đến bagnr hiện

Thao tác với CSDL

- 1 bảng student: id (PK), name, age, gender, address, pointMath, pointLit, pointEng

- CRUD: Create - Read - Update - Delete

- Create 1 record:

insert into student(`name`, `age`, `gender`, `address`, `pointMath`, `pointLit`, `pointEng`)

value ('trí', 23, 'nam', 'HN', 10, 10, 10);

- Update: yêu cầu update như thế nào

update student set name = 'hùng' where id = 1;

- Delete: yêu cầu delete như thế nào

delete from student where pointMath > 5;

delete from student where ((pointMath + pointLit + pointEng)/3) > 5;

- Read: yêu cầu read như thế nào

select \* from student where student.age = 20;

select address from student group by address;

- 1-1: 2 bảng có quan hệ 1-1

- 1-N: 1 bản ghi của bảng này có thể tham chiếu đến nhiều bản ghi của bảng kia

- n-n: 1 bản ghi của bảng này có thể tham chiếu đến nhiều bản ghi của bảng kia và ngược lại

- Join:

+ Inner join: phần chung giao nhau giữa 2 bảng có tham chiếu đến nhau

+ Left join: phần chung giao nhau và bảng bên trái => giá trị tương ứng bảng bên phải là null

+ Right join: phần chung giao nhau và bảng bên phải => giá trị tương ứng bảng bên phải là null

+ Full outer join: lấy các giá trị ở 1 trong 2 bảng được tham chiếu

Các thao tác ngoài thao tác truy xuất CSDL cơ bản sẽ làm giảm đôi chút hiệu suất của hệ thống

Index: là 1 chỉ mục, dùng để tăng tốc độ truy xuất trong 1 bảng dữ liệu

+ Khóa chính: not null + unique

+ Unique

View: là 1 góc nhìn, 1 bảng ảo được sử dụng để truy xuất đến những cột mong muốn trong bảng cơ sở; đảm bảo tính bảo mật của dữ liệu

Stored Procedure: là 1 thủ tục ở trong SQL

+ khi mà sử dụng 1 câu truy vấn SQL để gọi về DB lấy dữ liệu, thì khi đó DB nó mới tiến hành compile câu truy vấn đó để lấy dữ liệu => tốn thời gian

+ để tạo sẵn các câu truy vấn SQL mà minh mong muốn => DB sẽ compile sẵn các câu SQL nằm trong DB, gọi đến procedure nó sẽ thực hiện ngay mà không compile => tăng tốc độ truy vấn

+ procedure sẽ làm tăng bộ nhớ của DB => cần nơi lưu trữ các thủ tục đã được compile

+ có thể có tham số: IN, OUT, INOUT

+ IN: tham số đầu vào; điền khi gọi procedure

+ OUT: tham số trả ra; kết quả nằm trong out

+ INOUT: vừa là đầu vào, vừa là đầu ra

Các hàm thông dụng trong SQL:

+ avg(column): tính trung bình các giá trị trong 1 cột

+ sum(column): tính tổng các giá trị trong 1 cột

+ max(column): tìm giá trị lớn nhất trong cột

+ min(column): tìm giá trị nhỏ nhất trong cột

+ count(column): đếm số lần xuất hiện trong cột

+ now(column): trả về thời gian hiện tại với KDL là ngày tháng năm time

+ ucase(column): chuyển tất cả chuỗi thành chữ hoa

+ lcase(column): chuyển tất cả chuỗi thành chữ thường

+ length(column): trả về số lượng ký tự của chuỗi

+ Mệnh đề group by: gom nhóm các giá trị giống nhau

+ Mệnh đề order by: dùng để sắp xếp giá trị, mặc định tăng dần, giảm dần thêm DESC

+ Phân biệt having và where: cả 2 đều là mệnh đề đặt điều kiện

+ having là điều kiên đi cùng vs group by, thường là biểu thức

+ where là mệnh đề đi vs select

+ round(giá trị, số chữ số thập phân cần lấy): làm tròn số thập phân

+ and: và, kết hợp 2 điều kiện, đúng khi cả 2 điều kiện cùng đúng

+ or: hoặc, kết hợp 2 điều kiện, đúng khi 1 trong 2 điều kiện đúng

+ between: khoảng giữa, đúng khi giá trị nằm trong khoảng giới hạn

+ not: không giống

+ like: giống; có thể đi kèm %, dạng giống như regex

+ exist: khi 2 bảng có quan hệ, kiểm tra xem bản ghi của bảng này có nằm trong liên kết của bảng kia hay không

+ truncate: xóa toàn bộ bản ghi đang có trong 1 bảng: truncate table <table\_name>

+ concat: nối giá trị của 2 cột trong 1 bảng

Website tĩnh và website động: Static web và dynamic web

+ Web tĩnh hạn chế các hoạt động thay đổi dữ liệu trên trang web đó, chỉ sử dụng thuần HTML, CSS và 1 phần JS

+ Web động thì sẽ có sự tương tác với người dùng nhiều hơn, sử dụng dữ liệu, thay đổi dữ liệu nếu có

HTTP: Hyper Text Transfer Protocol - Giao thức truyền tải siêu văn bản: được sử dụng trong các ứng dụng Web (www - world wide web)

HTTP/HTTPs: s - security: bảo mật

Request: yêu cầu được truyền đi từ phía client lên server để yêu cầu 1 lời hồi đáp từ server

Response: hồi đáp của server đối với client

doGet/doPost: sẽ được quyết định sử dụng bởi request

get: thường được sử dụng để trả ra dữ liệu dưới dạng các thông tin, không cần bảo mật, phần dữ liệu sẽ nằm trên thanh URL

post: dùng cho các yêu cầu bảo mật hơn, phần dữ liệu được gửi đi ở client sẽ nằm trong phần header của request

Vòng đời của 1 Servlet:

+ init() - khởi tạo

+ service() - dịch vụ mà servlet đảm nhiệm cung cấp

+ destroy()

ServletContainer: nới chứa các servlet của 1 chương trình web.

forward()

sendDirect()

Cả 2 đều được sử dụng để điều hướng

- JSTL: JSP Standard Tags Library: các thư viện thẻ tags để hỗ trợ thao tác Java trong jsp dễ dang hơn

- Định danh biến trong JSP:

+ Cú pháp: ${biến được gọi}

**\* Các thẻ Core Tags**

- c:out : hiển thị 1 giá trị được đặt trong value của thẻ

- c:set : đặt giá trị cho 1 biến được định danh trong thẻ set

+ scope: vùng truy cập để trỏ đến thẻ set

+ var: tên biến để gán giá trị

+ value: giá trị của biến đặt trong var

- c:remove: xóa bỏ giá trị của 1 biến đã được định sẵn

+ scope: vùng truy cập để trỏ đến thẻ set

+ var: tên biến sẽ xóa giá trị

- c:catch: bắt lỗi trực tiếp trong jsp

+var: đặt tên định danh cho lỗi tìm thấy

- c:if : câu lệnh điều kiện trong jsp; nếu đúng thì làm cái gì đó

+ test: 1 biến chứa biểu thức điều kiện để xác định giá trị cho CLĐK

\* c:choose - c:when - c:otherwise: gần tương tự với switch-case trong Java

- c:choose: đánh dấu phần bắt đầu biểu thức lựa chọn theo điều kiện được đặt trong when

- c:when: khi điều kiện trong đó thỏa mãn thì làm gì đó

- c:otherwise: mặc định thực hiện nếu các điều kiện trong when không tồn tại hoặc đều sai

- c:forTokens: nhận 1 chuỗi giá trị và hiển thị theo định dạng được đặt trong delims

+ items: chuỗi giá trị được định danh

+ delims: định dạng để cắt list theo yêu câu tương ứng

+ var: giá trị để hiển thị

- c:url: tạo 1 đường dẫn trong jsp

+ value: giá trị của đường dẫn đó

+ var: tên biến chưa giá trị

- c:param: thêm các param nếu cần cho url được tạo ở trên

+ name: tên của param

+ value: giá trị của param

- c:redirect: điều hướng sang 1 url mới được đặt trong url của redirect (điều hướng ngay)

+ url: đường dẫn để điều hướng

- c:forEach: tương tự như vòng lặp forEach của java

+ items: list giá trị được gửi từ servlet

+ var: biến để chứa các giá trị được gán lần lượt trong forEach

MÔ HÌNH MVC

Mô hình MVC: kiến trúc dùng để thiết kế các dự án, theo 1 mô hình chuẩn, đảm bảo các tính chất của ngôn ngữ lập trình, cũng như giúp dễ dàng nâng câp hoặc bảo trì hệ thống.

M - Model: vùng chứa các đối tượng và xử lý logic, thao tác với Database

V - View: nơi hiển thị giao diện, xử lý các phần hiển thị của dự án

C - Controller: điều hướng các request, response đúng theo yêu cầu để phần model xử lý

Kiến trúc phân tầng:

+ Tầng giao diện

+ Tầng xử lý logic

+ Tầng cơ sở hạ tầng

…

- Dự án code:

+ Package Controller: tương ứng phần Controller của MVC

+ Package Model: tương ứng phần Model của MVC

+ Package Service: thuộc M

+ webapp: tương ứng phần view của MVC

+ Package Config: package chứa các file cài đặt

+ Package Repository: thuộc M

-------------------------------------------------

Driver: kết nối với database

DriverManager: quản lý các driver dùng để kết nối với database

Connection: chứa các phương thức đa dạng để thao tác với database

Statement: gồm 3 statement chính

+ statement: dùng để sử dụng với các truy vấn tĩnh (k chưa tham số)

+ prepare statement: dùng để thao tác với các truy vấn động (chưa các tham số ?, có phương thức set để thêm các tham số vào câu truy vấn)

+ callable statement: dùng để gọi các stored produce đã khởi tạo trước đó trong mySQL

ResultSet: tập hợp các kết quả được trả về sau khi thực hiện câu truy vấn

SQLException: các lỗi khi kết nối Java vs database