Principles of Database Systems (CS213)

Lecture 6-1: Application Development

Yuxin Ma, Shiqi Yu

Department of Computer Science and Engineering
Southern University of Science and Technology

- Most contents are from slides made by Stéphane Faroult and the authors of Database System Concepts (7th Edition).
- Their original slides have been modified to adapt to the schedule of CS307 at SUSTech.

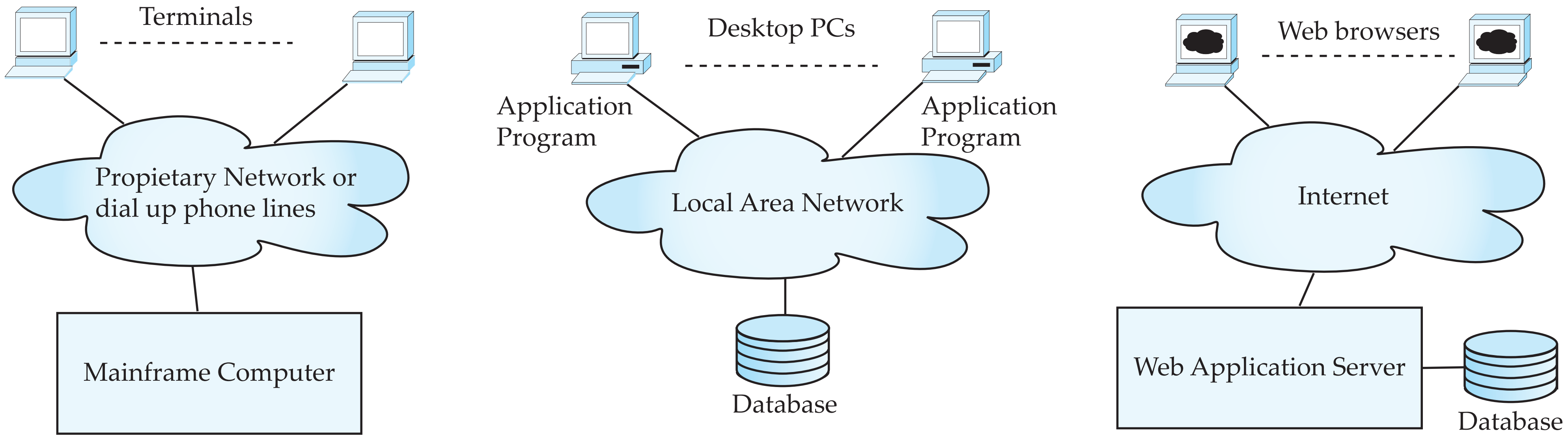
Application Development

Application Programs and User Interfaces

- Most database users do *not* use a query language like SQL
- An application program acts as the intermediary between users and the database
  - Applications split into
    - front-end
    - middle layer
    - backend
- Front-end: user interface
  - Forms
  - Graphical user interfaces
  - Many interfaces are Web-based

Application Architecture Evolution

- Three distinct era's of application architecture
  - Mainframe (1960' s and 70' s)
  - Personal computer era (1980' s)
  - Web era (mid 1990' s onwards)
  - Web and Smartphone era (2010 onwards)



(a) Mainframe Era

(b) Personal Computer Era

(c) Web era

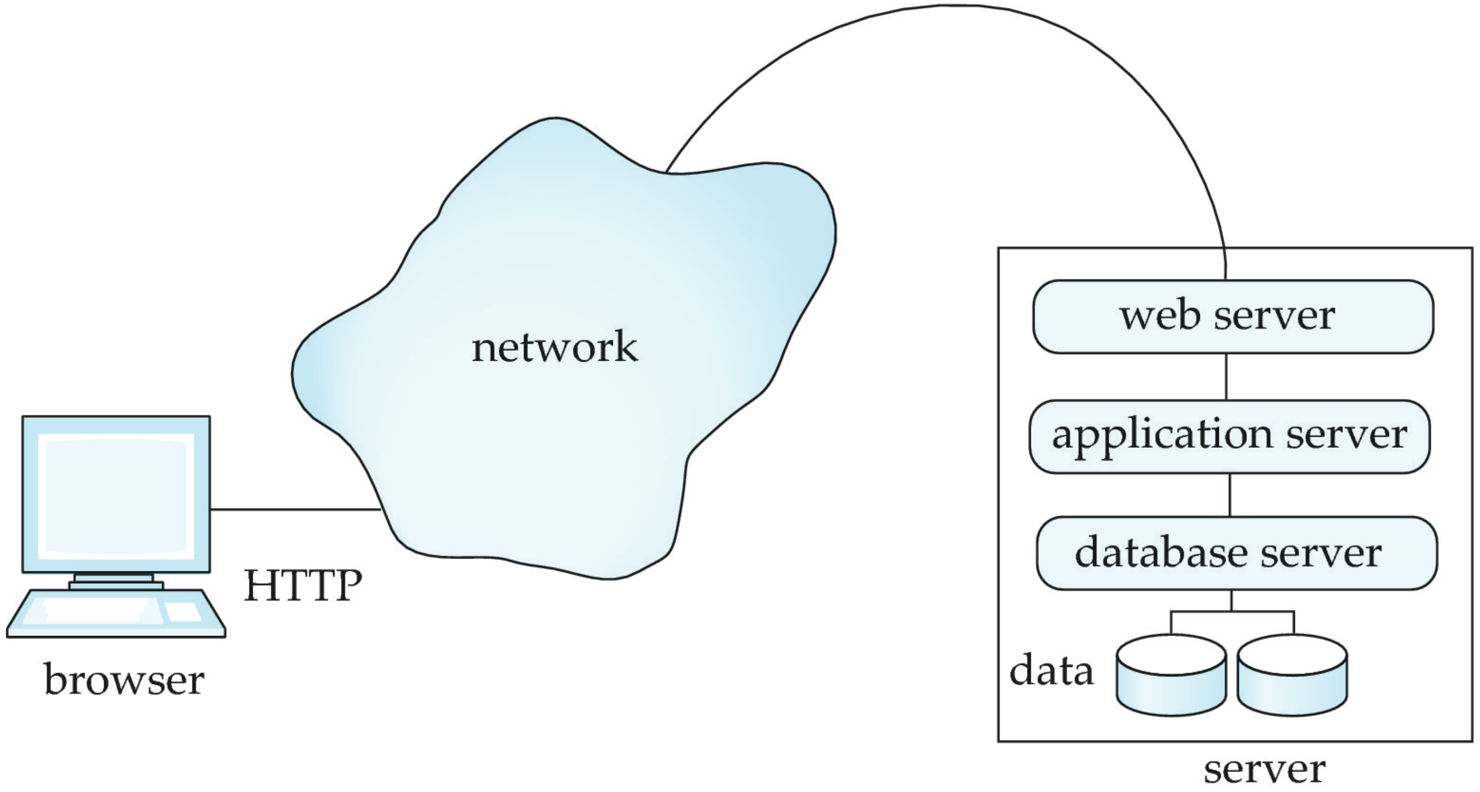
Web Interface

- Web browsers have become the de-facto standard user interface to databases
  - Enable large numbers of users to access databases from anywhere
  - Avoid the need for downloading/installing specialized code, while providing a good graphical user interface
    - Javascript, Flash and other scripting languages run in browser, but are downloaded transparently
  - Examples: banks, airline and rental car reservations, university course registration and grading, and so on.

The World Wide Web

- The Web is a distributed information system based on hypertext.
- Most Web documents are hypertext documents formatted via the HyperText Markup Language (HTML)
- HTML documents contain
  - text along with font specifications, and other formatting instructions
  - hypertext links to other documents, which can be associated with regions of the text.
  - forms, enabling users to enter data which can then be sent back to the Web server

Three-Layer Web Architecture



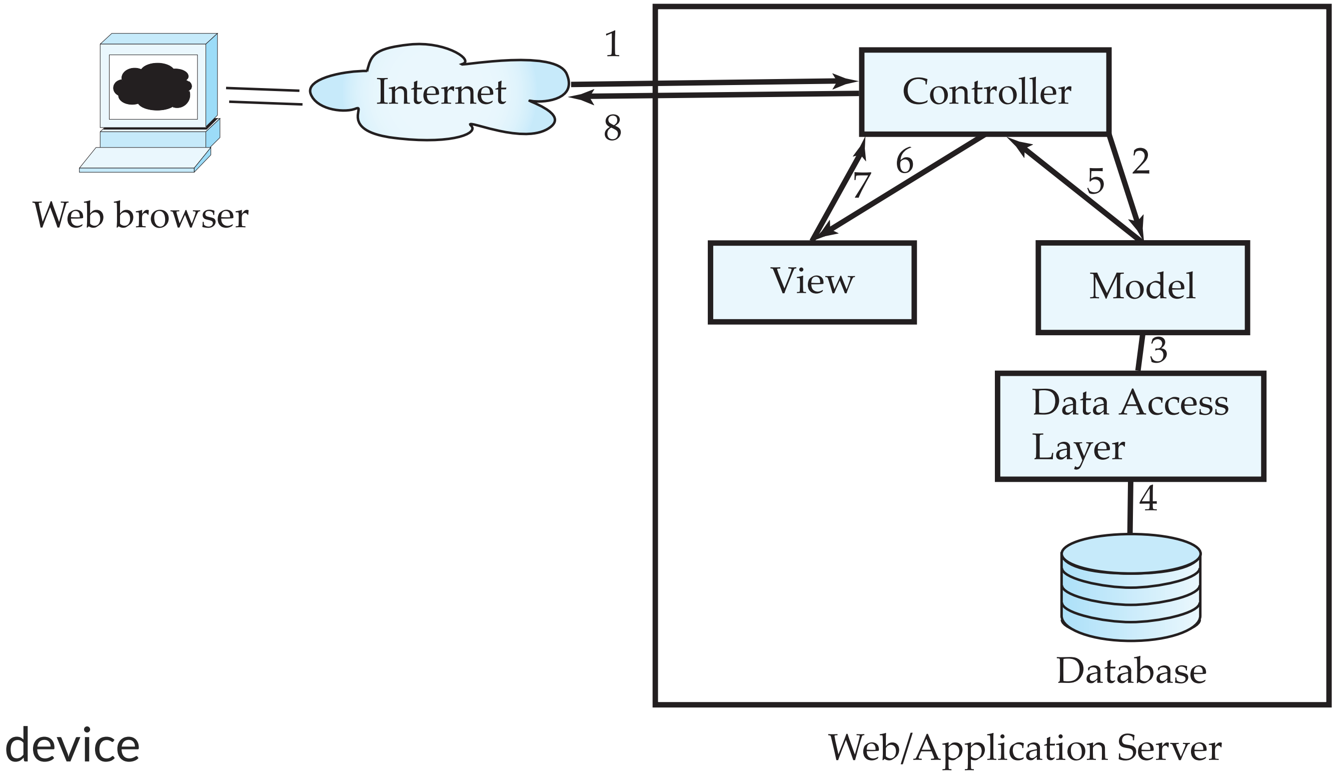
HTML and HTTP

- HTML provides formatting, hypertext link, and image display features
  - including tables, stylesheets (to alter default formatting), etc.
- HTML also provides input features
  - Select from a set of options
    - Pop-up menus, radio buttons, check lists
  - Enter values
    - Text boxes
  - Filled in input sent back to the server, to be acted upon by an executable at the server
- HyperText Transfer Protocol (HTTP) used for communication with the Web server

JavaScript

- Javascript very widely used
  - Forms basis of new generation of Web applications (called Web 2.0 applications) offering rich user interfaces
- Javascript functions can
  - Check input for validity
  - Modify the displayed Web page, by altering the underling document object model (DOM) tree representation of the displayed HTML text
    - Communicate with a Web server to fetch data and modify the current page using fetched data, without needing to reload/refresh the page
    - Forms basis of AJAX technology used widely in Web 2.0 applications
    - E.g. on selecting a country in a drop-down menu, the list of states in that country is automatically populated in a linked drop-down menu

Application Architectures



- Application layers
  - Presentation or user interface
    - model-view-controller (MVC) architecture
      - model: business logic
      - view: presentation of data, depends on display device
      - controller: receives events, executes actions, and returns a view to the user
  - business-logic layer
    - provides high level view of data and actions on data
      - often using an object data model
    - hides details of data storage schema
  - data access layer
    - interfaces between business logic layer and the underlying database
    - provides mapping from object model of business layer to relational model of database

Business Logic Layer

- Provides abstractions of entities
    - E.g., students, instructors, courses, etc
- Enforces business rules for carrying out actions
    - E.g., student can enroll in a class only if she has completed prerequsites, and has paid her tuition fees
- Supports workflows which define how a task involving multiple participants is to be carried out
    - E.g., how to process application by a student applying to a university
    - Sequence of steps to carry out task
    - Error handling
        - E.g. what to do if recommendation letters not received on time

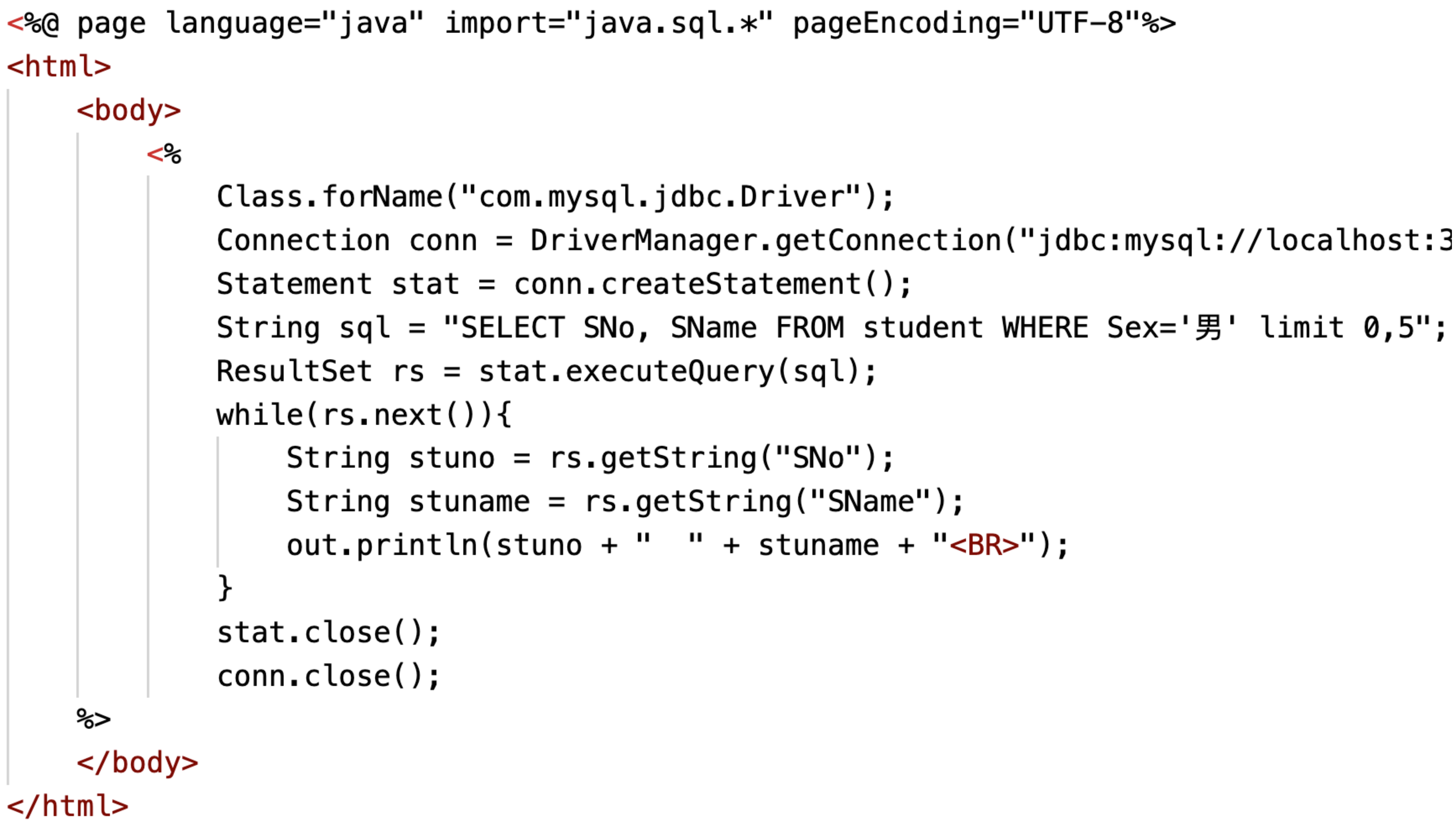
Object-Relational Mapping

- Allows application code to be written on top of object-oriented data model, while storing data in a traditional relational database
  - Alternative: implement object-oriented or object-relational database to store object model
    - Has not been commercially successful
- Schema designer has to provide a mapping between object data and relational schema
  - E.g., Java class Student mapped to relation student, with corresponding mapping of attributes
  - An object can map to multiple tuples in multiple relations
- Application opens a session, which connects to the database
- Objects can be created and saved to the database using session.save(object)
  - Mapping used to create appropriate tuples in the database
- Query can be run to retrieve objects satisfying specified predicates

Web Services

- Allow data on Web to be accessed using remote procedure call mechanism
- Two approaches are widely used
    - Representation State Transfer (REST): allows use of standard HTTP request to a URL to execute a request and return data
        - Returned data is encoded either in XML, or in JavaScript Object Notation (JSON)
    - Big Web Services:
        - Uses XML representation for sending request data, as well as for returning results
        - Standard protocol layer built on top of HTTP

Example of JSP



```
<%@ page language="java" import="java.sql.*" pageEncoding="UTF-8"%>
<html>
    <body>
        <%
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3
            Statement stat = conn.createStatement();
            String sql = "SELECT SNo, SName FROM student WHERE Sex='男' limit 0,5";
            ResultSet rs = stat.executeQuery(sql);
            while(rs.next()){
                String stuno = rs.getString("SNo");
                String stuname = rs.getString("SName");
                out.println(stuno + "   " + stuname + "<BR>");
            }
            stat.close();
            conn.close();
    %>
    </body>
</html>
```

Self Study

- Key points:
  - Techniques / libraries / APIs to connect to a database (e.g. PostgreSQL) and run SQL querys in a program
    - ODBC, JDBC?
    - SQLAlchemy? Spring Boot? Hibernate?
  - (Web) Backend Frameworks
    - Spring Boot, Flask, Django
    - NodeJS
  - Frontend Frameworks
    - React, Vue