



数据库系统概论

An Introduction to Database System

课程实验与习题课

主讲：冯文俊



数据库系统

2

- 如何开发数据库系统？—项目开发
 - 什么是B/S C/S 架构？
 - 一个标准的软件开发流程是什么？
 - 前后端是什么？
 - AI编程入门



C/S 架构和 B/S架构

3





C/S 架构和 B/S架构

4

1. **BS架构**: B代表“Browser”（浏览器），S代表“Server”（服务器）。BS架构就是用户通过浏览器访问服务器提供的服务，比如你用浏览器打开一个网站，这个网站背后就是一个BS架构。
2. **CS架构**: C代表“Client”（客户端），S同样代表“Server”（服务器）。CS架构就是用户需要安装一个专门的客户端软件来连接服务器，比如你用QQ或微信聊天，这些都属于CS架构。



一个标准的软件开发流程

5

问题定义（背景）：

假设XX银行希望开发一个线上银行系统，希望可以将之前都需要线下办理的业务搬到线上，让这个银行的用户都可以通过手机或电脑进行账户查询、转账、支付账单等操作。

于是这个银行就找到了你所在的软件开发团队，要求你们给他开发这样一个软件。

XX银行：甲方

XX软件开发团队：乙方

软件团队的参与者：

产品经理：负责与银行的业务部门沟通，明确银行的业务需求和痛点。



一个标准的软件开发流程

6

可行性研究

- 评估项目是否可行，包括技术、经济和资源方面。技术可行性是否有技术能力开发安全、高效的线上银行系统。
- 经济可行性：开发成本是否在银行预算范围内。
- 资源可行性：是否有足够的开发人员、时间和工具支持项目

参与角色：

- 项目经理：负责制定项目计划，评估资源和时间。
- 技术架构师（比较**senior**的技术人员）：评估技术可行性，提出整体上的技术解决方案。



一个标准的软件开发流程

7

需求分析

- 确定用户需要什么功能，并将需求细化为明确的技术需求。功能需求：账户查询、转账、支付账单、交易记录查询等。
- 非功能需求：必须确保系统安全性高、响应速度快、能够处理高并发用户请求。

参与角色：

- **产品经理：**负责与客户沟通，收集用户需求。将需求与开发团队对接，去回答“做什么”的问题。



一个标准的软件开发流程

8

概要/详细设计

- 设计软件的整体结构和功能模块。比如：
 - 都有什么界面，每个界面长什么样，包括登录页面、账户查询页面。
 - 有哪些具体的实现业务逻辑，例如用户登录验证、账户查询、交易处理、转账等
 - 数据库：有哪些信息，存储用户信息、账户信息、交易记录。

参与角色：

- 产品经理：制作产品原型，定义产品功能，画ER图等



一个标准的软件开发流程

9

编码实现

- 按照详细设计，开始写代码实现功能。
 - 前端开发：写登录页面、账户查询页面。
 - 后端开发：写处理用户请求的逻辑，比如转账功能。
 - 数据库开发：创建数据库表，设置索引。

参与角色：

- 前端开发工程师：负责实现用户界面。
- 后端开发工程师：负责实现业务逻辑，负责实现数据库功能。
- 算法工程师：实现智能算法需求，例如智慧问答客服



一个标准的软件开发流程

10

软件维护

- 任务：
软件上线后进行维护，包括修复bug、优化性能、添加新功能。
 - 比如：
 - 修复用户反馈的登录失败问题。
 - 添加新功能，比如支持更多支付方式。
- 参与角色：
 - 运维工程师：负责服务器的稳定运行和监控。
 - 前后端开发工程师：负责修复bug和开发新功能。



什么是前端和后端？

11

- **前端：**这是用户可以**直接看到和与之互动的部分**。比如，打开一个网站时，你看到的页面上的文字、图片、按钮、表单，这些都是前端。前端就像一个商店的门面，吸引人并让人方便使用。
- **后端：**这是**用户看不到的幕后部分**，负责处理数据和逻辑。比如，当你在网站上点击“搜索”按钮，背后有一个系统在处理你的请求，查找数据，然后把结果传给前端显示出来。这些幕后工作的部分就是后端。



什么是前端和后端？

12

前端的细节：

前端是由一些特定的技术构建出来的，它的主要目标是让用户可以在网页或应用上看到东西、点击按钮、填写表单等。前端开发通常使用以下三种核心技术：

1. **HTML**（超文本标记语言）：负责网页的结构，比如决定标题、段落、图片放在哪里。
2. **CSS**（层叠样式表）：负责网页的样式，比如颜色、字体、布局，看起来漂亮。
3. **JavaScript**：让网页有交互功能，比如点击按钮后弹出一个窗口，或者动态改变页面内容。

后端的细节：

后端则是处理逻辑和数据的地方，它主要负责：

1. **服务器**：后端运行在服务器上。服务器就像一台强大的电脑，专门用于存储网站的数据和程序
2. **数据库**：后端会存储用户的数据，比如账号密码、订单信息等。数据库就是一个有条理的“数据仓库”。
3. **后端逻辑**：当你在前端（比如一个网页）发出请求，比如提交一个表单，后端会接收你的请求，进行处理，然后给你一个结果（比如“登录成功”或者“搜索结果”）。



什么是前端和后端？

13

- 前端框架就像一种工具箱，专门用来帮助前端开发程序员更快、更方便地制作网页上的“外观”和“互动”。
- 比如，当你打开一个网站，看到各种按钮、表单、图片，或者点击某个按钮会弹出一个窗口，这些都是“前端”的内容。而类似**Vue**、**React**都是很流行的前端框架，它帮助程序员更简单地管理网页上的这些内容。前端框架为进阶内容，大家可以按需选择是否要使用前端框架。



什么是前端和后端？

14

前端和后端和数据库如何合作？

完整流程示例（网购场景）：

- ① 你点击“购买”（前端展示网页，收集动作）
- ② 前端告诉后端：“用户要买3号商品”
- ③ 后端查询数据库：“查询3号商品还有几件？”
- ④ 数据库返回：“还剩5件”
- ⑤ 后端计算价格，告诉前端：“可以买，总价200元”
- ⑥ 前端显示支付页面给你

总的来说：前端负责展示，后端负责处理数据和业务逻辑。数据库负责存储数据



什么是前端和后端？

15

□ 语言（前端必学，但不必学的很懂，了解这三部分是做什么的就可以）

➤ HTML

学习参考：[菜鸟教程-HTML教程](#)

➤ CSS

学习参考：[菜鸟教程-CSS教程](#)

➤ JavaScript

学习参考：[菜鸟教程-JavaScript教程](#)

□ 框架（进阶内容，可选学）

➤ Vue

学习参考：[Vue官网](#)，[B站尚硅谷Vue教程](#)

➤ Element-UI

学习参考：[Element-UI官网](#)

➤ ECharts

使用 JavaScript 实现的开源可视化库，各种折线图、柱状图、路径图等酷炫的制图应有尽有（数据分析展示必备）

学习参考：[ECharts官网](#)

➤ 其他

Bootstrap、Jquery等框架可以了解下，都是以前Web开发常用框架，现在使用Vue等框架的话很少使用了



什么是前端和后端？

16

□ 语言 (必学)

➤ Python

学习参考: [菜鸟教程-Python教程](#)

➤ MySQL

学习参考: [菜鸟教程-MySQL教程](#)

□ 后端框架 (推荐Flask, 不需要学精, 能掌握部分用法就够用)

➤ Flask

学习参考: [B站知了传课Flask教程](#)

➤ Spring

学习参考: [菜鸟教程-Spring教程](#)



AI编程入门

17

使用AI辅助编程

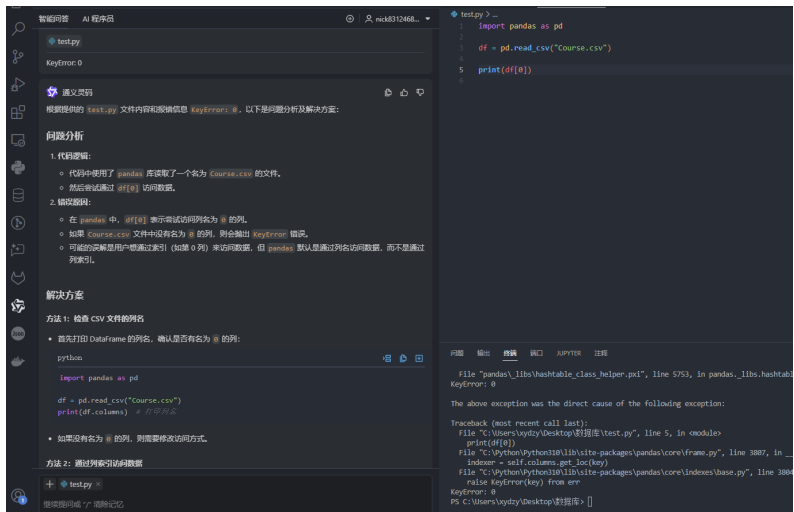
VSCode中的AI插件：通义灵码、GitHub Copilot（建议使用中科大学身份申请一个学生账号）

其他编程软件：Cursor、Trae、Windsurf

使用场景一：代码补全

```
test.py > ...
1 import pandas as pd
2
3 df = pd.read_csv('test.csv')
```

使用场景二：利用报错信息debug





AI编程入门

18

使用AI辅助编程

最常见的使用场景：输入需求AI直接写出代码：

```
test.py

请基于flask写一个后端demo

通义灵码

当然可以！下面是一个基于 Flask 的简单后端示例。这个示例将创建一个基本的 Flask 应用，包含一个路由来返回一个简单的 JSON 响应。

步骤 1: 安装 Flask

首先，确保你已经安装了 Flask。如果没有安装，可以使用 pip 进行安装：

bash

pip install Flask

步骤 2: 创建 Flask 应用

在 test.py 文件中编写以下代码：

python

from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, Flask!"

@app.route('/api/data', methods=['GET'])
def get_data():
    data = {
        "message": "This is a sample data response",
        "items": [1, 2, 3, 4, 5]
    }
```

```
test.vue

请基于vue写一个前端的注册页面

通义灵码

当然可以！下面是一个基于 Vue.js 的简单注册页面示例。这个示例将创建一个包含用户名、电子邮件、密码和确认密码输入框的注册表单，并在提交时进行简单的验证。

步骤 1: 安装 Vue CLI (如果尚未安装)

如果你还没有安装 Vue CLI，可以使用以下命令进行安装：

bash

npm install -g @vue/cli

步骤 2: 创建 Vue 项目 (如果尚未创建)

如果你还没有 Vue 项目，可以使用 Vue CLI 创建一个新的项目：

bash

vue create my-vue-app
cd my-vue-app

步骤 3: 创建注册页面组件

在 src/components 目录下创建一个新的文件 Register.vue，并添加以下代码：

vue

<template>
  <div class="register-container">
    <h2>注册</h2>
    <form @submit.prevent="register">
      <div class="form-group">
```



AI编程入门

19

- AI编程工具的出现革新了程序员的编程方式，在“聊天式编程”时代之下，程序员写代码时愈发需要重视“想清楚”和“说清楚”

使用TIP：当描述一个较复杂的需求时，使用格式化的自然语言而不是意识流语言和AI沟通：

请用Python编写代码你需要读取data.csv文件。我需要你删除所有包含空值的行，然后筛选出temperature列大于35的记录...



请用Python编写代码实现以下需求：

背景

输入：读取data.csv文件，包含列

['date', 'temperature', 'humidity']

需求

删除所有包含空值的行

筛选temperature列大于35的记录

用matplotlib绘制折线图

x轴为date列，y轴为temperature列

红线标注平均值(35)

输出：保存为high_temp.png

限制

绘图时图片中的文字必须是中文

不可以在代码中加入注释





AI编程入门

20

- 对于大的项目，将整个项目拆开成不同的模块。大模型很难一次性生成上千行无误的代码。这时应该分而治之，将复杂的问题拆分为简单问题，一步步解决。

使用例子：开发一个电商网站时，先让AI生成用户认证模块并测试，再生成商品展示模块并测试等等。而不是让AI一次性生成整个项目



AI编程入门

21

用 AI 编程软件进行前后端开发的总体开发流程（可以使用这个思路来做本次实验）

用 AI 的**总体开发流程**是：

1. **想清楚要做什么，设计系统架构。**对于这次实验，只涉及三个模块：前端、后端、数据库，
2. **让AI编程软件编写每个模块**，一般是先数据结构（数据库表结构），再前后端的顺序比较好。每个模块开始编写时最好给它一个干净的文件，外围有类似功能的文件供它参考，然后**给每个模块一个清晰的 prompt，就像跟人沟通需求一样**。就像现实世界的合作开发一样，一开始的需求思考和沟通绝对不是浪费时间，一开始定好需求，后面能省很多返工的时间。AI 用 1 分钟左右就能写出一版，看一遍，了解 AI 写的代码结构，顺便修复一些很容易看出的问题。
3. **各个模块联调。**在联调的时候发现问题，**程序员定位到问题所属的模块，可以把错误直接告诉 Cursor，让它自己去改**。如果 Cursor 改不好，程序员再自己去定位和修改问题。如果问题源于代码仓库中的已有代码，一般是需要程序员人工介入定位的，不要过度依赖 AI 编程。



其他TIPS

22

- 学校里面书本上的知识只是工具，学会“工具是什么”和学会“工具怎么用”之间存在很大的gap（这也是为什么多出去实习很重要），所以建议大家一定认真对待这次的数据库大实验！
- 由于本次实验是团队合作形式，强烈推荐大家通过这次合作机会学会如何使用github来管理、更新、维护你们团队的代码，并学会如何协同开发一个项目。熟练使用git是每一个程序员都必须会的技能，但是学校不会教你这些。
- 在大模型时代，学习新知识一定要充分利用好“大模型老师”，学会如何使用大模型学习新知识，使用大模型来写代码。AI不会淘汰程序员，而是先会淘汰不会使用AI的程序员。



现场演示

23

需求:

- 名称: 个人日志管理系统简约版
- 需求: 用户需要一个网站管理个人的日志, 记录日常事项等
- 功能: 允许个人用户创建、查看、编辑和删除日志条目。



现场演示

24

第一步：设计数据库表，数据结构

需要两张表，一张记录用户信息，一张记录日志信息

根据需求，我们设计的用户表中需要记录 **“用户id、用户名、用户密码、用户创建时间、用户邮箱”** 几个字段。

日志表中需要记录**“用户id，日志标题、日志内容、日志创建时间、日志修改时间”**，其中用户id是需要添加外键约束



现场演示

25

Prompt如下:

背景

我现在需要做一个前后端项目。名称：个人日志管理系统简约版。需求：用户需要一个网站管理个人的日志，记录日常事项等。功能：允许个人用户创建、查看、编辑和删除日志条目。

需求

- 你现在需要为我创建系统的数据库表，有两张表需要你创建
- 用户表中需要记录“用户id、用户名、用户密码、用户创建时间、用户邮箱”几个字段。
- 日志表中需要记录“用户id，日志标题、日志内容、日志创建时间、日志修改时间”，其中用户id是需要添加外键约束

约束

- 你需要使用mysql语法来创建表
- 除了我说的这些需求，你不可以做除了这个之外其他没必要的事情



现场演示

26

第二步：搭建登录模块

需求：搭建出一个最基本的flask框架。然后再在框架之上进行开发

Prompt:

背景

我现在需要做一个前后端项目。名称：个人日志管理系统简约版。需求：用户需要一个网站管理个人的日志，记录日常事项等。功能：允许个人用户创建、查看、编辑和删除日志条目。我已经完成了数据库表的创建

需求

- 你现在需要为我搭建出登录模块的前后端框架
- 根据用户输入的用户名和密码验证是否正确，若正确则跳转主页。主页先放一个空的测试页面
- 实现用户注册功能

约束

- 后端使用flask，前端使用html css javascript
- 除了我说的这些需求，你不可以做除了这个之外其他没必要的事情



现场演示

27

第三步：搭建主页

需求：搭建出主页模块。实现日志系统的主要功能

Prompt:

背景

我现在需要做一个前后端项目。名称：个人日志管理系统简约版。需求：用户需要一个网站管理个人的日志，记录日常事项等。功能：允许个人用户创建、查看、编辑和删除日志条目。我已经完成了数据库表的创建和登录模块。

需求

- 你现在需要为我搭建出主页模块的前后端框架
- 主页需要实现日志创建、查看、编辑和删除功能

约束

- 后端使用flask，前端使用html css javascript
- 除了我说的这些需求，你不可以做除了这个之外其他没必要的事情