Python数据库操作 教案

|  |  |  |  |
| --- | --- | --- | --- |
| **周次** | **课次** | **章节内容** | **课时** |
| 7 | 2 | MySQL数据库连接 | 2 |
| **单元教学目标** | | | |
| - 知识目标：掌握MySQL连接的基本命令和参数设置，理解数据库连接的协议规范及参数含义。  - 技能目标：能够使用Python代码成功连接MySQL数据库，并处理常见的连接错误，如认证失败或网络中断。  - 素养目标：具备良好的编程规范意识，能够通过代码注释和文档说明实现数据库连接逻辑，体现职业素养。 | | | |
| **教学重点** | | **教学难点** | |
| • 掌握MySQL数据库连接的基本语法和库安装方法（如`pip install mysql-connector-python`）  • 理解连接字符串的构成要素（主机名、端口、数据库名、用户名、密码）  • 能够通过Python代码实现简单查询操作（如`execute()`和`fetchall()`方法的使用） | | • 学生可能难以掌握如何正确配置MySQL数据库连接参数（如host、user、password、database），尤其是需要处理不同数据库的认证信息（如密码加密或认证方式）  • 学生可能难以理解如何通过游标执行SQL查询并处理结果集（如区分fetchall()与fetchone()的区别，或处理多行结果的逻辑） | |
| **教学活动** | | | |
| - 新课导入【5分钟】：   教师讲述一个医院管理系统案例，引入数据库连接需求。通过提问引导学生思考：“若要管理患者信息，需如何与数据库交互？”结合学生生活经验，用故事导入激发兴趣，设置学习情境。   - 讲授新课【15分钟】：   1. \*\*讲授法\*\*：   - 介绍MySQL基础概念（数据库、表、字段），强调连接的重要性。   - 通过代码示例演示安装MySQL（如使用`mysql-connector-python`库）。   2. \*\*案例分析法\*\*：   - 分组分析“学生信息管理系统”案例，完成连接代码（如`import mysql.connector`）。   3. \*\*讨论法\*\*：   - 学生讨论连接过程中可能遇到的问题（如权限不足、连接失败），教师引导归纳解决步骤。   4. \*\*头脑风暴法\*\*：   - 课堂讨论“如何确保数据库连接安全”，学生提出使用用户名、密码、主机名等参数。   5. \*\*示范法\*\*：   - 教师演示完整连接代码，分步骤讲解（如连接、查询、关闭）。   - 巩固练习【10分钟】：   1. \*\*分层练习\*\*：   - \*\*基础练习\*\*：学生独立编写简单连接代码，查询学生表数据。   - \*\*挑战练习\*\*：分组设计“查询班级成绩”功能，要求使用参数化查询防止SQL注入。   2. \*\*实践操作\*\*：   - 学生分组完成小项目：   - 第一组：连接MySQL数据库，编写查询语句。   - 第二组：设计用户界面（如使用Tkinter）展示查询结果。   3. \*\*互评机制\*\*：   - 学生互评代码逻辑正确性，教师点评关键点（如异常处理）。   - 归纳总结【5分钟】：   1. \*\*教师总结\*\*：   - 简要回顾连接步骤（安装库、配置参数、编写代码），强调安全实践（如使用参数化查询）。   2. \*\*学生复述\*\*：   - 学生用关键词卡片总结核心内容（如“连接步骤”“安全原则”），教师补充遗漏点。   3. \*\*角色扮演\*\*：   - 学生模拟“数据库管理员”角色，复述连接流程并提出优化建议（如使用数据库驱动）。   \*\*教学方法整合\*\*：  - \*\*讲授法\*\*：知识点分解与代码示范。  - \*\*案例分析法\*\*：实际项目拆解与问题讨论。  - \*\*讨论法\*\*：学生协作解决连接问题。  - \*\*头脑风暴法\*\*：安全实践方案生成。  - \*\*分层练习\*\*：差异化任务设计，兼顾基础与挑战。  - \*\*角色扮演\*\*：模拟真实场景，增强应用能力。 | | | |
| **教学资源** | | | |
| • 教学设备：计算机（安装Python环境）、投影仪、键盘鼠标、数据库管理工具（如MySQL Workbench或命令行工具）  • 推荐参考资料：   - 书籍：《Python数据库应用》（机械工业出版社）   - 在线资源：W3Schools Python数据库教程（https://www.w3school.com/xml/python\_xml.asp）   - 官方文档：Python官方文档（https://docs.python.org/3/library/sqlite3.html）   - 实践案例：MySQL数据库连接示例代码（附带注释）  • 建议工具：安装MySQL服务器（如通过`mysql-server`命令行安装）  • 学习路径：   1. 安装MySQL并创建测试数据库   2. 使用`mysql.connector`库连接数据库   3. 实现增删改查操作（示例代码：`cursor.execute("INSERT INTO table...")`）  • 教学辅助：提供MySQL数据库连接参数配置表（如host、user、password、database） | | | |
| **教学反思** | | | |
| • 教学效果方面：学生是否能够正确连接MySQL数据库并执行基本操作（如查询、插入），是否理解事务处理的原理，是否存在操作失误或理解偏差  • 学生反馈方面：学生对实际应用（如数据查询）的兴趣程度，是否对SQL语法或数据库连接细节存在困惑，是否有因安装库或配置问题导致的困难  • 教学改进建议：增加分步骤演示（如安装库、连接配置、SQL语句编写），提供更多实践练习（如模拟数据库操作），补充常见错误排查案例  • 技术细节讲解：是否需要更详细解释参数化查询的原理，或强调安全最佳实践（如避免硬编码密码）  • 互动与参与度：增加实时问答环节，针对学生常见问题（如连接失败原因）进行针对性讲解  • 差异化教学：针对不同水平学生设计不同难度的练习任务，如基础操作与复杂查询结合  • 工具与资源：提供更直观的示例代码（如使用pymysql库的简化版示例），减少配置复杂度  • 评估与反馈：通过小测验或代码提交验证学生掌握程度，及时调整教学重点  • 课后延伸：引导学生探索数据库优化（如索引使用）或跨数据库操作（如MySQL与PostgreSQL） | | | |
| **教学评价** | | | |
| • 过程性评价  • 课堂参与度：观察学生在教师讲解、小组讨论中的主动性和互动情况（如回答问题、提问、参与讨论）。  • 学习态度：评估学生是否按时完成作业、主动预习或复习相关知识（如通过课堂笔记、代码提交记录）。  • 实践操作表现：记录学生在MySQL连接任务中的操作准确性（如是否正确安装库、编写连接代码、处理异常）。  • 小组合作：通过小组任务完成情况（如分工协作、沟通效率）评估团队合作能力。   • 结果性评价  • 课堂测试：通过随堂测试（如连接参数配置、异常处理判断）评估知识掌握程度。  • 项目作业：根据学生提交的连接代码（如`mysql.connector`库的使用是否正确）评分，重点关注逻辑完整性与错误处理。  • 代码提交：通过代码评审表（如语法正确性、功能完整性）评估学生对MySQL连接的掌握深度。  • 任务完成度：评估学生是否独立完成连接任务（如写出完整的连接字符串、测试连接是否成功）。   • 评价标准与方式  • 课堂参与度：20%（观察记录表+教师评分）  • 学习态度：15%（作业提交记录+教师反馈）  • 实践操作表现：25%（代码提交+教师评审）  • 小组合作：20%（小组任务完成情况+教师评分）  • 结果性评价：30%（课堂测试+项目作业+代码提交）   • 评价工具  • 评分表：针对每个评价维度设定具体分数（如课堂参与度满分10分）  • 观察记录表：记录学生在课堂中的具体表现（如提问次数、操作准确性）  • 代码评审表：对提交代码进行技术性评分（如语法正确性、逻辑完整性） | | | |