# Python数据库操作 教案

周次：7

课次：2

章节内容：MySQL数据库连接

课时：2

## 单元教学目标

- 知识目标：能够识别MySQL数据库连接所需的参数（如host、user、password、port）并理解其含义   
- 技能目标：能够使用mysql-connector-python库编写代码实现MySQL数据库连接，并正确配置参数   
- 素养目标：养成规范的数据库连接操作习惯，遵守数据安全规范，确保系统性能与稳定性

## 教学重点

• 使用mysql-connector-python安装和导入模块   
• 建立数据库连接的基本步骤（如`mysql.connector.connect()`）   
• 使用cursor对象执行查询并处理结果集（如`cursor.execute()`和`cursor.fetchall()`)

## 教学难点

• 学生可能难以掌握MySQL连接库的安装与使用，例如不清楚如何通过pip安装mysql-connector-python，或误将连接字符串写成Python语法（如`mysql+mysqlconnector://user:password@host/dbname`），导致连接失败。   
• 如何正确执行SQL语句并处理结果集是难点，例如学生可能混淆`cursor.execute()`与`cursor.fetchall()`的使用场景，或未掌握参数化查询的语法（如`cursor.execute("SELECT \* FROM users WHERE id=%s", (1,))`），导致数据插入/查询不准确。

## 教学活动

新课导入【5分钟】   
教师通过讲述“医院管理系统案例”引入主题：某医院需用数据库管理患者信息，学生讨论如何用Python连接MySQL数据库。教师提问“你们知道MySQL的连接方式吗？”激发兴趣，引导学生思考数据库操作的重要性。   
  
讲授新课【10分钟】   
1. \*\*讲授法\*\*（3分钟）：   
 - 教师讲解MySQL安装步骤（Windows命令行安装），强调服务启动与端口配置。   
 - 通过案例分析法演示代码片段：`import mysql.connector`，说明依赖库安装命令。   
2. \*\*讨论法\*\*（2分钟）：   
 - 学生分组讨论“如何配置MySQL用户权限”，教师引导学生思考权限设置的必要性。   
3. \*\*练习法\*\*（3分钟）：   
 - 教师示范连接数据库代码，学生尝试运行并记录错误，教师点评修正。   
  
巩固练习【5分钟】   
1. \*\*小组合作\*\*（2分钟）：   
 - 学生分组完成“连接MySQL数据库并查询患者信息”的基础练习，教师巡视指导。   
2. \*\*角色扮演\*\*（2分钟）：   
 - 学生模拟数据库管理员角色，设计权限配置方案并展示，教师点评其合理性。   
  
归纳总结【5分钟】   
教师总结关键步骤（安装、配置、连接），学生用头脑风暴法复述连接流程，教师补充注意事项（如密码安全、连接关闭）。通过提问“如何避免连接错误？”引导学生巩固知识。

## 教学资源

</think>  
  
• \*\*教学设备和工具\*\*   
- 计算机（带Python环境）   
- MySQL数据库服务器（建议安装MySQL并配置好）   
- 屏幕投影设备（用于展示演示内容）   
- 白板或黑板（用于板书讲解）   
- 软件开发工具包（如PyCharm、VS Code）   
- 数据库管理工具（如MySQL Workbench）   
- 本地或远程MySQL数据库实例（可选）   
  
• \*\*推荐的参考资料或网站\*\*   
- 《Python数据库操作》书籍（可参考《Python程序设计基础》或《Python数据库开发实战》）   
- 官方文档：[MySQL 官方文档](https://dev.mysql.com/doc/)   
- Python官方文档：[Python数据库模块](https://docs.python.org/3/library/dbapi.html)   
- 在线教程：[Python for Data Analysis](https://www.datacamp.com/tutorials/python-for-data-analysis)   
- GitHub仓库：[mysql-connector-python](https://github.com/mysql-connector-python/mysql-connector-python)   
- 网站：[Real Python](https://realpython.com/)（提供Python数据库操作教程）   
- 课程网站：[Coursera 或 edX 上的Python数据库课程](https://www.coursera.org)（根据实际课程选择）

## 教学反思

• 教学效果方面：学生是否掌握MySQL连接的基本概念（如驱动安装、连接字符串构建）？是否能独立完成数据库连接操作？是否存在学生在实际操作中出现连接失败或语法错误？   
• 学生反馈方面：是否发现部分学生对MySQL客户端安装流程理解困难？是否在实践环节中出现因库版本兼容性问题导致的困惑？是否对数据库查询结果的展示方式（如打印输出或可视化）存在兴趣差异？   
• 教学方法优化：是否需要增加分步骤的安装教程（如依赖库安装、环境变量配置）？是否应通过对比MySQL与SQLite的连接方式，帮助学生理解数据库类型差异？   
• 实践环节设计：是否因时间限制导致学生无法完成完整数据库操作（如查询、插入、更新）？是否需要提供更详细的示例代码或分组练习任务？   
• 技术细节讲解：是否遗漏了连接池配置、异常处理等高级话题？是否应补充数据库驱动版本兼容性提示（如Python 3.10以上需使用mysql-connector-python 8.0+）？   
• 互动与反馈机制：是否通过课堂提问或小组讨论发现学生对数据库事务、索引等概念的模糊理解？是否需要增加实时答疑环节或课后巩固练习？   
• 差异化教学：是否对不同基础的学生设置不同难度的练习任务（如仅用基础语句连接数据库 vs. 完整操作流程）？是否需要补充常见错误排查指南（如权限问题、网络阻断）？   
• 教学资源补充：是否因课时限制未能提供完整的代码示例或操作指南？是否应提前准备安装包下载链接或虚拟环境配置教程？

## 教学评价

• 过程性评价   
• 课堂参与度：观察学生在教师讲解和小组讨论中的主动性和表达能力，记录其对MySQL连接知识点的掌握情况（如是否能正确使用`mysql.connector`或`pymysql`模块）。   
• 作业完成质量：批改学生提交的连接代码，评估其逻辑正确性、语法规范性及是否解决常见错误（如端口、用户名、密码配置错误）。   
• 课堂练习表现：通过小组任务或实践操作，评价学生能否独立完成数据库连接测试，如使用`mysql.connector`连接MySQL数据库并执行简单查询。   
• 问题解决能力：记录学生在遇到连接问题时的思考过程和解决方案，如是否能通过调试工具定位错误并修正。   
  
• 结果性评价   
• 项目成果：根据学生提交的完整数据库操作项目（如创建表、插入数据、查询结果展示）评分，重点考察代码是否符合规范、逻辑是否清晰、功能是否完整。   
• 代码质量评分：采用代码审查表评估代码的可读性、注释完整性、是否遵循最佳实践（如使用参数化查询防止SQL注入）。   
• 课堂测试成绩：通过随堂测试或小测验（如判断是否正确使用`connect()`方法、是否能写出连接字符串）验证学生对知识点的掌握程度。   
• 项目答辩表现：学生需展示其项目成果并解释技术实现，评价其对MySQL连接原理的理解深度及实际应用能力。   
  
• 评价标准说明   
• 过程性评价：以4级评分（优秀/良好/一般/需要改进）量化学生表现，结合课堂观察、作业批改和实践操作数据综合评定。   
• 结果性评价：以4级评分（优秀/良好/合格/需要提升）评估项目成果和代码质量，重点考核技术准确性、逻辑性及学习成果达成度。