



数据库系统原理

课程设计

李建国

lijianguo@scnu.edu.cn

授课对象

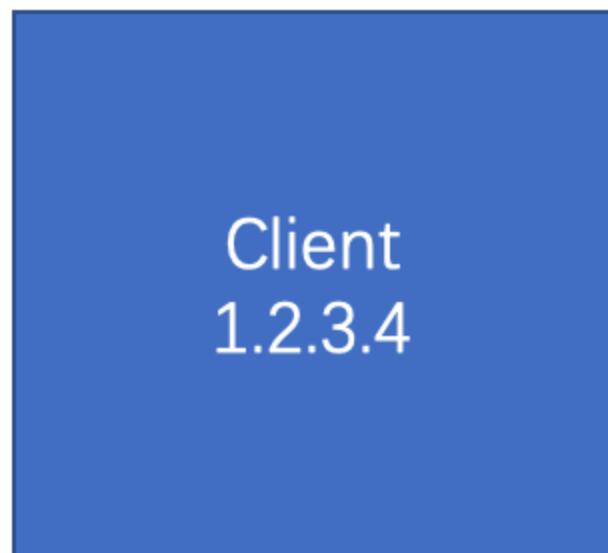
24级计科及计科（师范）（111人+41人）

按照所属班级加入学者网数据库系统课程项目

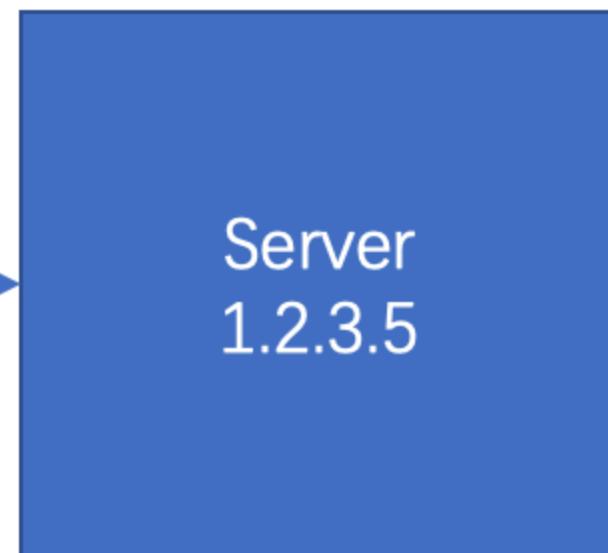
<https://www.scholat.com/course/dbproj>，加入密码：2024dbp，课程任务在学者网发布，完成后在学者网提交，如有任何疑问，可在学者网留言或者发送Email。

一、课程目标和技术要求

HOST (Windows)



VM (Linux)



Networking

1. 服务器端数据库系统软件搭建

目标： 实现数据库系统的部署，确保其具备可操作性，并根据安全策略配置访问权限。

技术要求：

- 选择并部署一个稳定且安全的Linux发行版，作为数据库系统的运行环境。该环境可以是物理服务器或虚拟化平台。
- 根据业务需求，选择并安装合适的数据库管理系统（DBMS），如MySQL、PostgreSQL、Redis等。
- 配置数据库实例，包括但不限于数据库用户账户的创建、权限的分配、存储引擎的配置以及性能参数的优化。

2. 小型数据库应用开发

目标： 开发一个客户端应用程序，用于与服务器端数据库进行交互，实现数据的访问和操作。

技术要求：

- 采用客户端-服务器（CS）或浏览器-服务器（BS）架构模式，确保客户端和服务端具有独立的IP地址和网络通信能力。
- 利用Java、Python、C#等编程语言，开发客户端应用程序，实现与数据库的连接和数据交互。
- 客户端应用程序应实现数据库的基本操作，包括数据的增删改查（CRUD）功能。

二、课程内容 - 传统（通用）数据库

1. 代表数据库

- **MySQL**: 开源的关系型数据库管理系统，适用于中小型应用。
- **PostgreSQL**: 强大的开源对象-关系型数据库系统，支持复杂查询和大规模数据处理。
- **SQL Server**: 由微软开发的商业数据库管理系统，广泛应用于企业级应用。
- **DB2**: 由IBM开发的关系型数据库管理系统，适用于大型企业应用。

2. 特征

- **通用、访问接口统一 (ODBC/JDBC):** 提供统一的数据库访问接口，便于开发者进行跨平台开发。
- **事务特性支持完整 (ACID) :** 保证数据的一致性和可靠性。
- **复杂 (安装、使用及维护) :** 安装和维护过程相对复杂，需要一定的专业知识。
- **扩展性一般:** 扩展性能不如新型数据库，但能满足一般应用需求。

3. 课程作业 – 访问和操作传统数据库（50分）

- 客户端要求语言: Java、Python、C#或其它Web开发语言
 - **Python**: 简单易学，支持多种数据库访问库，如PyODBC、SQLAlchemy。
 - **PHP**: 广泛用于Web开发，支持数据库访问库如PDO、MySQLi。
 - **C#**: 适用于Windows平台，支持ADO.NET等数据库访问技术。
 - **JavaScript (Node.js)**: 适用于前后端开发，支持数据库访问库如node-postgres、mysql。
- 服务端要求数据库: MySQL, PostgreSQL, SQLServer

通过增加多种客户端开发语言和可能的功能，可以根据自己的兴趣和项目需求选择适合的技术，实现更丰富和多样的数据库系统应用。

- **实现基本功能:**
 - 创建数据表
 - 插入数据
 - 删除数据
 - 修改数据
 - 删除数据表
 - 跨数据表操作（如JOIN操作）
- **其他要求:** 数据库内至少有5个表格，初始数据至少共100000条。

- **实现额外功能:**

- **事务支持:** 实现事务处理，保证数据操作的原子性和一致性。
- **索引支持:** 创建索引以提高查询性能。
- **多用户（多客户端）同时访问数据库:** 支持并发访问，记录和分析访问性能。
- **数据备份和恢复:** 实现数据库的备份和恢复功能，确保数据安全。
- **视图（Views）支持:** 创建视图简化复杂查询操作。
- **存储过程和触发器:** 编写和调用存储过程，设置触发器自动处理特定事件。

三、课程内容 – 新型（专用）数据库

1. 用途

- 致力于存储特别的、专门的数据: 优化资源消耗，提升访问性能。
- 部署方式:
 - 与传统通用数据库结合，提升访问性能。
 - 单独部署，针对特定应用场景。

2. 经典代表



- 键值 (KV) 数据库: Memcached、Redis
- 列族数据库: HBase
- 图数据库: Neo4j
- 文档数据库: MongoDB

SQL数据库+NoSQL数据库？



[VoltDB](#) 是一个内存中的开源OLTP SQL数据库，能够保证事务的完整性（ACID）。它是Postgres和Ingres联合创始人Mike Stonebraker领导开发的下一代开源数据库管理系统。它能在现有的廉价服务器集群上实现每秒数百万次数据处理。

迈克尔·斯通布雷克

由于对现代数据库的概念和实践作出的根本性贡献，于2015年获得计算机图灵奖



3. 特点

- **简单:** 部署、安装及使用方便。
- **功能单一和缺失:** 适用于特定场景，但功能相对简单。
- **访问接口不兼容:** 各种专用数据库的访问接口不同，不能通用。
- **扩展性表现较好:** 能够较好地扩展以应对大规模数据。
- **不一定支持事务:** 事务支持可能不完善或不支持。

4. 推荐使用的NoSQL数据库

- **SCYLLADB**: 高性能的C++重写版Cassandra，每节点每秒处理100万TPS。
- **Memcached**: 高效的分布式内存对象缓存系统。
- **Redis**: 高性能的键值数据库，支持多种数据结构。
- **VoltDB**: 内存中的开源OLTP SQL数据库，事务完整性（ACID）支持良好。
- **MongoDB**: 文档数据库，适用于大规模数据存储和查询。
- **Neo4j**: 图数据库，适用于复杂的图数据关系存储和查询。

5. 课程作业 – 访问和操作专用数据库（50分）

- **选择1个NoSQL数据库:** 搭建并编写客户端应用程序进行交互。
- **客户端开发语言:** 除了前述的C++、Java、Python、PHP、Ruby、C#、JavaScript (Node.js)外，还可以选择：
 - **Scala:** 与大数据处理框架（如Apache Spark）集成良好，支持数据库访问库如Slick。
 - **R:** 适用于数据分析，支持数据库访问包如RODBC、DBI。
- **操作要求:** 数据插入、修改、删除等，以及与该专用数据库相关的特色操作。
- **开发环境:** 与传统数据库项目要求一致。

- 可选的特色功能

- 批量操作: 批量插入、更新、删除数据，提高操作效率。
- 全文检索: 实现对数据的全文检索功能（如Elasticsearch集成）。
- 数据缓存: 实现数据缓存，减少数据库访问次数，提高系统性能。
- 分布式存储: 在多节点间分布数据，提升系统的扩展性和容错性。
- 实时数据处理: 处理和分析实时数据流，如日志、传感器数据。
- 数据压缩: 实现数据压缩功能，减少存储空间占用。
- 地理空间查询: 支持地理空间数据存储和查询，如GeoJSON。
- 复杂图查询: 对图数据库进行复杂查询操作，如路径搜索、社交网络分析。

四、作业组织和考核方式

分组完成课程设计

- **每组最多3人**: 选定组长负责统筹项目、分配任务、提交内容以及项目验收预约。

作业验收

- **预约时间:** **【课程设计预约表】**
- **系统演示**: 现场演示系统的运行情况或提前录像，并就系统功能和熟练程度进行简单问答。自备电子设备，由**组长**提前将提交内容上传至学者网，**每组只需要一个人提交**。

五、提交内容说明

- 任务分组.xlsx: 组员姓名及学号，完成的任务等。[【任务分组】](#)
- 中期报告.docx
- 课程报告.docx: 内容简明，结构合理，格式规范
- 作品演示.pptx: 系统主要功能与特色功能展示
- 源代码文件夹
- 数据文件夹

将上述内容放在文件夹**DB课程设计XXX组**并打包，由组长上传提交至学者网，每组只需提交一份。

六、作业提交和相关时间点

- 作业布置
 - 日期: 2024年9月9日前
 - 任务: 使用通用数据库的和使用NoSQL数据库的应用系统。
- 中期报告
 - 日期: 2024年10月15日前
 - 内容: 分组情况、作品初步设计、数据来源及选用的数据库。
- 作业验收
 - 日期: 学期末第17周, 即2024年12月24日