**DataHouseDB文档**

# 文档管理

**更改历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **状态** | **日期** | **负责人** | **更改原有** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 目录

1 文档管理 2

2 目录 3

3 数据库模块 4

3.1 存储引擎 4

3.2 查询处理器 4

3.3 事务管理 4

3.4 缓冲区管理 4

3.5 索引管理 4

3.6 元数据管理 4

3.7 用户接口 4

3.8 安全性管理 5

3.9 备份与恢复 5

3.10 监控与诊断 5

3.11 网络通信（如果适用） 5

# 数据库模块

开发一个自己的数据库系统是一个复杂且具有挑战性的任务，它涉及到多个模块和组件的设计与实现。一个典型的数据库管理系统（DBMS）通常包括以下几个核心模块：

## 存储引擎

负责数据的物理存储和检索。

- \*\*文件管理\*\*：管理数据文件、索引文件等。

- \*\*页管理\*\*：处理磁盘上的数据块（页），包括分配、回收和缓存。

- \*\*事务日志\*\*：记录所有对数据库所做的更改，以支持恢复机制。

## 查询处理器

解析用户提交的SQL语句，并生成执行计划。

- \*\*词法分析器和语法分析器\*\*：将SQL语句解析为抽象语法树（AST）。

- \*\*查询优化器\*\*：选择最有效的查询执行路径。

- \*\*执行引擎\*\*：根据优化后的执行计划执行查询。

## 事务管理

确保事务的ACID属性（原子性、一致性、隔离性和持久性）。

- \*\*锁管理器\*\*：控制并发访问，防止读写冲突。

- \*\*并发控制\*\*：实现多版本并发控制（MVCC）或其他并发控制策略。

- \*\*恢复管理\*\*：通过日志和其他手段保证数据库在故障后能够恢复到一致状态。

## 缓冲区管理

管理内存中的缓存页面，提高I/O性能。

- \*\*替换算法\*\*：决定哪些页面应该保留在缓存中，哪些应该被移出。

- \*\*脏页管理\*\*：跟踪修改过的页面，并适时地将它们写回磁盘。

## 索引管理

创建和维护索引来加速数据检索。

- \*\*B+树\*\*、\*\*哈希索引\*\*等不同类型的索引结构。

- \*\*索引创建/删除\*\*：允许用户定义索引并进行维护。

## 元数据管理

存储和管理关于数据库对象（如表、视图、索引等）的信息。

- \*\*数据字典\*\*：描述数据库结构和内容的系统表。

- \*\*模式管理\*\*：支持模式的创建、修改和删除操作。

## 用户接口

提供给用户交互的方式。

- \*\*SQL解释器\*\*：接受用户的SQL命令，并将其传递给查询处理器。

- \*\*命令行工具\*\*或\*\*图形界面\*\*：供用户直接操作数据库。

## 安全性管理

保护数据免受未授权访问。

- \*\*认证\*\*：验证用户身份。

- \*\*授权\*\*：控制用户权限，指定他们可以执行的操作。

- \*\*加密\*\*：保护敏感数据的安全性。

## 备份与恢复

定期备份数据库，并能够在需要时恢复。

- \*\*全量备份\*\*和\*\*增量备份\*\*：保存数据库的状态。

- \*\*点-in-time恢复\*\*：允许恢复到特定的时间点。

## 监控与诊断

收集运行时信息，帮助管理员了解数据库性能并解决问题。

- \*\*性能统计\*\*：跟踪关键指标，如查询延迟、吞吐量等。

- \*\*错误日志\*\*：记录任何异常情况。

## 网络通信（如果适用）

如果数据库是分布式或多节点架构，则需要网络通信模块。

- \*\*RPC框架\*\*：用于客户端和服务端之间的远程过程调用。

- \*\*消息队列\*\*：协调不同节点间的消息传递。

# DataHouseDB

客户端

ShellCommand客户端

java客户端

SQL解析

antlr4

数据存储

数据存储

索引存储

数据查询

元数据管理

表结构

字段信息

datahouse

|  |  |
| --- | --- |
| 脚本  script |  |
| 程序  bin |  |
| 配置  config |  |
| 存储目录  store | metadata  link  data  tmp |
| 备份 |  |
|  |  |