

数据库设计

技术栈

- 结构化元数据：MariaDB
- 文档内容块：MongoDB

设计原则

- 结构/关系 → MariaDB
- 大文本/JSON/高频更新 → MongoDB
- 一切皆 Node (统一节点抽象)

一、整体模型

代码块

```
1 Project
2   └─ Node (application/page/function)
3     ├─ Document (仅 function 有)
4     ├─ Dictionary
5     └─ Permission
```

职责拆分：

存储	负责内容
MariaDB	项目、节点树、文档元数据、权限、词典
MongoDB	文档 block 内容

二、MariaDB 表设计 (核心)

1. 项目表 project

建表语句

代码块

```
1 CREATE TABLE project (
2     id BIGINT PRIMARY KEY AUTO_INCREMENT,
3     name VARCHAR(128) NOT NULL UNIQUE,
4     description VARCHAR(400),
5
6     creator BIGINT NOT NULL,
7     created_at DATETIME NOT NULL,
8
9     editor BIGINT NOT NULL,
10    edited_at DATETIME NOT NULL
11 );
```

字段说明

字段	职责
id	项目唯一ID
name	项目名称（唯一约束）
description	项目描述
creator	创建人
created_at	创建时间
editor	最近编辑人
edited_at	最近编辑时间

2. 节点表 project_node (⭐最核心)

承载 application / page / function 全部结构

建表语句

代码块

```
1 CREATE TABLE project_node (
2     id BIGINT PRIMARY KEY AUTO_INCREMENT,
```

```
3
4     project_id BIGINT NOT NULL,
5     parent_id BIGINT DEFAULT NULL,
6
7     node_type VARCHAR(32) NOT NULL,
8     name VARCHAR(255) NOT NULL,
9     description TEXT,
10
11    path VARCHAR(1024) NOT NULL,
12    sort INT DEFAULT 0,
13
14    mode TINYINT DEFAULT 0,
15    status TINYINT DEFAULT 1,
16
17    creator BIGINT,
18    created_at DATETIME,
19    editor BIGINT,
20    edited_at DATETIME,
21
22    INDEX idx_project(project_id),
23    INDEX idx_parent(parent_id),
24    INDEX idx_path(path)
25 );
```

字段说明

字段	职责
id	节点ID
project_id	所属项目
parent_id	父节点
node_type	类型 (application/ page/ function)
name	节点名称
description	节点描述
path	树路径， 如 /1/3/8 (支持 子树查询)
sort	同级排序
mode	0编辑模式 1开发 模式
status	是否删除/有效
creator/ created_at	创建信息
editor/ edited_at	最近编辑信息

3. 节点类型约束表 node_type

建表语句

代码块

```
1 CREATE TABLE node_type (
2     code VARCHAR(32) PRIMARY KEY,
3     name VARCHAR(64),
4     parent_allow VARCHAR(255)
5 );
```

示例数据

code	parent_allow
application	root
page	application
function	page

职责

- 限制节点层级合法性
- 防止非法树结构

4. 功能设计文档表 function_document

仅 function 节点拥有

建表语句

代码块

```
1 CREATE TABLE function_document (
2     id BIGINT PRIMARY KEY AUTO_INCREMENT,
3     function_node_id BIGINT UNIQUE NOT NULL,
4
5     creator BIGINT,
6     created_at DATETIME,
7     editor BIGINT,
8     edited_at DATETIME
9 );
```

字段说明

字段	职责
id	文档ID
function_node_id	绑定功能节点 (1:1)
creator	创建人
created_at	创建时间
editor	编辑人
edited_at	编辑时间

5. 项目词典表 dictionary

建表语句

代码块

```
1 CREATE TABLE dictionary (
2     id BIGINT PRIMARY KEY AUTO_INCREMENT,
3     project_id BIGINT NOT NULL,
4
5     term VARCHAR(255) NOT NULL,
6     definition TEXT NOT NULL,
7
8     created_at DATETIME,
9
10    UNIQUE KEY uk_project_term(project_id, term)
11 );
```

字段说明

字段	职责
term	术语
definition	定义
UNIQUE	同项目不可重复

三、MongoDB 设计（文档内容）

1. 文档块集合 document_block

代码块

```
1 {
2     _id: ObjectId,
3     document_id: 1001,
4     type: "text | list | table | plugin",
5     content: {...},
```

```
6     order: 1,  
7     updated_at: ISODate()  
8 }
```

字段说明

字段	职责
document_id	关联 MariaDB 文档
type	block 类型
content	JSON 内容
order	排序
updated_at	更新时间

四、读写流程说明

创建功能节点

代码块

```
1 创建 node(function)  
2 → 自动创建 function_document
```

查询文档

代码块

```
1 MariaDB 查询 function_document  
2 → Mongo 查询 document_block  
3 → 拼装返回
```

增量更新

~~代码块~~ 增加更新变更 block

2 不覆盖全文

优势：

- 高性能
 - 减少写放大
 - 支持大文档
-

五、索引建议

MariaDB:

- project_id
- parent_id
- path
- function_node_id unique

MongoDB:

代码块

```
1 document_id + order  
2 document_id
```

六、总结

本设计实现：

- ✓ 节点统一抽象
 - ✓ 树结构高效查询
 - ✓ 文档增量更新
 - ✓ 大文本分离
 - ✓ 百万级扩展
 - ✓ 支持 MCP 高速读取
-

