# 自动化参数修改记录系统 - MySQL 数据库设计文档

## 📋 文档信息

|  |  |
| --- | --- |
| 项目 | 内容 |
| 数据库类型 | MySQL 8.0+ |
| 字符集 | utf8mb4 |
| 排序规则 | utf8mb4\_unicode\_ci |
| 存储引擎 | InnoDB |
| 文档版本 | 1.0 |
| 创建日期 | 2025-01-16 |
| 最后更新 | 2025-01-16 |

## 📊 数据库概览

### 数据库名称

CREATE DATABASE IF NOTEXISTS `automation\_param\_system` DEFAULTCHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci; USE `automation\_param\_system`;

### 数据表清单

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 表名 | 说明 | 记录数估算 |
| 1 | sys\_site | 运行现场信息表 | 50-100 |
| 2 | sys\_automation\_system | 自动化系统信息表 | 100-500 |
| 3 | sys\_department | 维护部门信息表 | 20-50 |
| 4 | param\_modification\_record | 参数修改记录主表 | 10,000+ |
| 5 | param\_modification\_detail | 参数修改详情表 | 50,000+ |
| 6 | param\_version\_history | 参数版本历史表 | 100,000+ |
| 7 | sys\_user | 系统用户表 | 50-200 |
| 8 | sys\_role | 角色表 | 5-10 |
| 9 | sys\_permission | 权限表 | 20-50 |
| 10 | sys\_operation\_log | 操作日志表 | 100,000+ |

## 📐 表结构设计

### 1. 运行现场信息表 (sys\_site)

**功能说明**: 存储系统部署的地理位置信息

CREATE TABLE `sys\_site` ( `site\_id` INT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '现场ID', `site\_code` VARCHAR(50) NOT NULL COMMENT '现场编码', `site\_name` VARCHAR(100) NOT NULL COMMENT '现场名称', `site\_location` VARCHAR(200) DEFAULTNULL COMMENT '详细地址', `province` VARCHAR(50) DEFAULTNULL COMMENT '省份', `city` VARCHAR(50) DEFAULTNULL COMMENT '城市', `contact\_person` VARCHAR(50) DEFAULTNULL COMMENT '联系人', `contact\_phone` VARCHAR(20) DEFAULTNULL COMMENT '联系电话', `status` TINYINT NOT NULLDEFAULT1 COMMENT '状态: 0-停用, 1-启用', `sort\_order` INTDEFAULT0 COMMENT '排序序号', `remark` VARCHAR(500) DEFAULTNULL COMMENT '备注', `created\_by` INT UNSIGNED DEFAULTNULL COMMENT '创建人ID', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', `updated\_by` INT UNSIGNED DEFAULTNULL COMMENT '更新人ID', `updated\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMPONUPDATECURRENT\_TIMESTAMP COMMENT '更新时间', `deleted` TINYINT NOT NULLDEFAULT0 COMMENT '逻辑删除: 0-未删除, 1-已删除', PRIMARY KEY (`site\_id`), UNIQUE KEY `uk\_site\_code` (`site\_code`), KEY `idx\_city` (`city`), KEY `idx\_status` (`status`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='运行现场信息表';

**示例数据**:

INSERT INTO `sys\_site` (`site\_code`, `site\_name`, `city`, `status`) VALUES ('GZ001', '广州', '广州', 1), ('ZZ001', '郑州', '郑州', 1);

### 2. 自动化系统信息表 (sys\_automation\_system)

**功能说明**: 存储自动化系统的基本信息

CREATE TABLE `sys\_automation\_system` ( `system\_id` INT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '系统ID', `system\_code` VARCHAR(50) NOT NULL COMMENT '系统编码', `system\_name` VARCHAR(200) NOT NULL COMMENT '系统名称', `site\_id` INT UNSIGNED NOT NULL COMMENT '所属现场ID', `department\_id` INT UNSIGNED DEFAULTNULL COMMENT '维护部门ID', `system\_type` VARCHAR(50) DEFAULTNULL COMMENT '系统类型', `role\_type` ENUM('主用', '备用') NOT NULLDEFAULT'主用' COMMENT '主备角色', `ip\_address` VARCHAR(50) DEFAULTNULL COMMENT 'IP地址', `port` INTDEFAULTNULL COMMENT '端口号', `version` VARCHAR(50) DEFAULTNULL COMMENT '系统版本', `vendor` VARCHAR(100) DEFAULTNULL COMMENT '厂商', `install\_date` DATEDEFAULTNULL COMMENT '安装日期', `warranty\_date` DATEDEFAULTNULL COMMENT '保修期至', `status` TINYINT NOT NULLDEFAULT1 COMMENT '状态: 0-离线, 1-在线, 2-维护中', `remark` VARCHAR(500) DEFAULTNULL COMMENT '备注', `created\_by` INT UNSIGNED DEFAULTNULL COMMENT '创建人ID', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', `updated\_by` INT UNSIGNED DEFAULTNULL COMMENT '更新人ID', `updated\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMPONUPDATECURRENT\_TIMESTAMP COMMENT '更新时间', `deleted` TINYINT NOT NULLDEFAULT0 COMMENT '逻辑删除: 0-未删除, 1-已删除', PRIMARY KEY (`system\_id`), UNIQUE KEY `uk\_system\_code` (`system\_code`), KEY `idx\_site\_id` (`site\_id`), KEY `idx\_department\_id` (`department\_id`), KEY `idx\_role\_type` (`role\_type`), KEY `idx\_status` (`status`), CONSTRAINT `fk\_system\_site` FOREIGN KEY (`site\_id`) REFERENCES `sys\_site` (`site\_id`), CONSTRAINT `fk\_system\_department` FOREIGN KEY (`department\_id`) REFERENCES `sys\_department` (`department\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='自动化系统信息表';

**示例数据**:

INSERT INTO `sys\_automation\_system` (`system\_code`, `system\_name`, `site\_id`, `role\_type`) VALUES ('SYS001', '秦雷丝', 1, '主用'), ('SYS002', '系统名称系统名称系统名称', 2, '备用');

### 3. 维护部门信息表 (sys\_department)

**功能说明**: 存储负责维护的部门信息

CREATE TABLE `sys\_department` ( `department\_id` INT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '部门ID', `department\_code` VARCHAR(50) NOT NULL COMMENT '部门编码', `department\_name` VARCHAR(100) NOT NULL COMMENT '部门名称', `parent\_id` INT UNSIGNED DEFAULTNULL COMMENT '上级部门ID', `department\_level` TINYINT DEFAULT1 COMMENT '部门层级', `leader\_name` VARCHAR(50) DEFAULTNULL COMMENT '部门负责人', `leader\_phone` VARCHAR(20) DEFAULTNULL COMMENT '负责人电话', `status` TINYINT NOT NULLDEFAULT1 COMMENT '状态: 0-停用, 1-启用', `sort\_order` INTDEFAULT0 COMMENT '排序序号', `remark` VARCHAR(500) DEFAULTNULL COMMENT '备注', `created\_by` INT UNSIGNED DEFAULTNULL COMMENT '创建人ID', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', `updated\_by` INT UNSIGNED DEFAULTNULL COMMENT '更新人ID', `updated\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMPONUPDATECURRENT\_TIMESTAMP COMMENT '更新时间', `deleted` TINYINT NOT NULLDEFAULT0 COMMENT '逻辑删除: 0-未删除, 1-已删除', PRIMARY KEY (`department\_id`), UNIQUE KEY `uk\_department\_code` (`department\_code`), KEY `idx\_parent\_id` (`parent\_id`), KEY `idx\_status` (`status`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='维护部门信息表';

**示例数据**:

INSERT INTO `sys\_department` (`department\_code`, `department\_name`) VALUES ('DEPT001', '部门名称');

### 4. 参数修改记录主表 (param\_modification\_record)

**功能说明**: 存储参数修改的汇总记录（对应界面列表显示）

CREATE TABLE `param\_modification\_record` ( `record\_id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '记录ID', `system\_id` INT UNSIGNED NOT NULL COMMENT '系统ID', `site\_id` INT UNSIGNED NOT NULL COMMENT '现场ID', `department\_id` INT UNSIGNED DEFAULTNULL COMMENT '维护部门ID', `modification\_count` INTNOT NULLDEFAULT0 COMMENT '参数修改次数', `last\_modified\_at` DATETIME DEFAULTNULL COMMENT '最后修改时间', `last\_modified\_by` INT UNSIGNED DEFAULTNULL COMMENT '最后修改人ID', `last\_modified\_by\_name` VARCHAR(50) DEFAULTNULL COMMENT '最后修改人姓名（冗余字段）', `status` TINYINT NOT NULLDEFAULT1 COMMENT '状态: 0-无效, 1-有效', `remark` VARCHAR(500) DEFAULTNULL COMMENT '备注', `created\_by` INT UNSIGNED DEFAULTNULL COMMENT '创建人ID', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', `updated\_by` INT UNSIGNED DEFAULTNULL COMMENT '更新人ID', `updated\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMPONUPDATECURRENT\_TIMESTAMP COMMENT '更新时间', `deleted` TINYINT NOT NULLDEFAULT0 COMMENT '逻辑删除: 0-未删除, 1-已删除', PRIMARY KEY (`record\_id`), KEY `idx\_system\_id` (`system\_id`), KEY `idx\_site\_id` (`site\_id`), KEY `idx\_department\_id` (`department\_id`), KEY `idx\_last\_modified\_at` (`last\_modified\_at`), KEY `idx\_created\_at` (`created\_at`), CONSTRAINT `fk\_record\_system` FOREIGN KEY (`system\_id`) REFERENCES `sys\_automation\_system` (`system\_id`), CONSTRAINT `fk\_record\_site` FOREIGN KEY (`site\_id`) REFERENCES `sys\_site` (`site\_id`), CONSTRAINT `fk\_record\_department` FOREIGN KEY (`department\_id`) REFERENCES `sys\_department` (`department\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='参数修改记录主表';

**字段说明**:

* modification\_count: 对应界面的"参数记录数"列
* last\_modified\_at: 对应界面的"更新时间"列

### 5. 参数修改详情表 (param\_modification\_detail)

**功能说明**: 存储每次参数修改的详细信息

CREATE TABLE `param\_modification\_detail` ( `detail\_id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '详情ID', `record\_id` BIGINT UNSIGNED NOT NULL COMMENT '记录ID（关联主表）', `system\_id` INT UNSIGNED NOT NULL COMMENT '系统ID（冗余字段，便于查询）', `modification\_type` VARCHAR(50) NOT NULL COMMENT '修改类型: 新增/修改/删除', `param\_category` VARCHAR(100) DEFAULTNULL COMMENT '参数分类', `param\_name` VARCHAR(200) NOT NULL COMMENT '参数名称', `param\_code` VARCHAR(100) DEFAULTNULL COMMENT '参数编码', `old\_value` TEXT DEFAULTNULL COMMENT '修改前值', `new\_value` TEXT DEFAULTNULL COMMENT '修改后值', `value\_type` VARCHAR(50) DEFAULTNULL COMMENT '值类型: string/int/float/boolean/json', `unit` VARCHAR(20) DEFAULTNULL COMMENT '单位', `modification\_reason` VARCHAR(500) DEFAULTNULL COMMENT '修改原因', `approval\_status` TINYINT DEFAULT0 COMMENT '审批状态: 0-待审批, 1-已通过, 2-已拒绝', `approver\_id` INT UNSIGNED DEFAULTNULL COMMENT '审批人ID', `approved\_at` DATETIME DEFAULTNULL COMMENT '审批时间', `effective\_status` TINYINT DEFAULT1 COMMENT '生效状态: 0-未生效, 1-已生效, 2-已失效', `effective\_at` DATETIME DEFAULTNULL COMMENT '生效时间', `modified\_by` INT UNSIGNED NOT NULL COMMENT '修改人ID', `modified\_by\_name` VARCHAR(50) DEFAULTNULL COMMENT '修改人姓名', `modified\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '修改时间', `remark` VARCHAR(1000) DEFAULTNULL COMMENT '备注', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', PRIMARY KEY (`detail\_id`), KEY `idx\_record\_id` (`record\_id`), KEY `idx\_system\_id` (`system\_id`), KEY `idx\_param\_name` (`param\_name`), KEY `idx\_modified\_at` (`modified\_at`), KEY `idx\_approval\_status` (`approval\_status`), CONSTRAINT `fk\_detail\_record` FOREIGN KEY (`record\_id`) REFERENCES `param\_modification\_record` (`record\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='参数修改详情表';

### 6. 参数版本历史表 (param\_version\_history)

**功能说明**: 存储参数的完整版本历史，支持版本对比

CREATE TABLE `param\_version\_history` ( `version\_id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '版本ID', `detail\_id` BIGINT UNSIGNED NOT NULL COMMENT '详情ID（关联详情表）', `system\_id` INT UNSIGNED NOT NULL COMMENT '系统ID', `param\_name` VARCHAR(200) NOT NULL COMMENT '参数名称', `param\_value` TEXT NOT NULL COMMENT '参数值', `version\_number` INTNOT NULL COMMENT '版本号', `is\_current` TINYINT NOT NULLDEFAULT0 COMMENT '是否当前版本: 0-否, 1-是', `change\_description` VARCHAR(500) DEFAULTNULL COMMENT '变更说明', `modified\_by` INT UNSIGNED NOT NULL COMMENT '修改人ID', `modified\_at` DATETIME NOT NULL COMMENT '修改时间', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', PRIMARY KEY (`version\_id`), KEY `idx\_detail\_id` (`detail\_id`), KEY `idx\_system\_param` (`system\_id`, `param\_name`), KEY `idx\_version\_number` (`version\_number`), KEY `idx\_is\_current` (`is\_current`), CONSTRAINT `fk\_version\_detail` FOREIGN KEY (`detail\_id`) REFERENCES `param\_modification\_detail` (`detail\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='参数版本历史表';

### 7. 系统用户表 (sys\_user)

**功能说明**: 存储系统用户信息

CREATE TABLE `sys\_user` ( `user\_id` INT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '用户ID', `username` VARCHAR(50) NOT NULL COMMENT '用户名', `password` VARCHAR(255) NOT NULL COMMENT '密码（加密存储）', `real\_name` VARCHAR(50) NOT NULL COMMENT '真实姓名', `employee\_no` VARCHAR(50) DEFAULTNULL COMMENT '工号', `email` VARCHAR(100) DEFAULTNULL COMMENT '邮箱', `phone` VARCHAR(20) DEFAULTNULL COMMENT '手机号', `department\_id` INT UNSIGNED DEFAULTNULL COMMENT '所属部门ID', `position` VARCHAR(50) DEFAULTNULL COMMENT '职位', `role\_ids` VARCHAR(200) DEFAULTNULL COMMENT '角色ID列表（逗号分隔）', `avatar` VARCHAR(500) DEFAULTNULL COMMENT '头像URL', `last\_login\_at` DATETIME DEFAULTNULL COMMENT '最后登录时间', `last\_login\_ip` VARCHAR(50) DEFAULTNULL COMMENT '最后登录IP', `login\_count` INTDEFAULT0 COMMENT '登录次数', `status` TINYINT NOT NULLDEFAULT1 COMMENT '状态: 0-禁用, 1-启用', `remark` VARCHAR(500) DEFAULTNULL COMMENT '备注', `created\_by` INT UNSIGNED DEFAULTNULL COMMENT '创建人ID', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', `updated\_by` INT UNSIGNED DEFAULTNULL COMMENT '更新人ID', `updated\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMPONUPDATECURRENT\_TIMESTAMP COMMENT '更新时间', `deleted` TINYINT NOT NULLDEFAULT0 COMMENT '逻辑删除: 0-未删除, 1-已删除', PRIMARY KEY (`user\_id`), UNIQUE KEY `uk\_username` (`username`), KEY `idx\_department\_id` (`department\_id`), KEY `idx\_status` (`status`), CONSTRAINT `fk\_user\_department` FOREIGN KEY (`department\_id`) REFERENCES `sys\_department` (`department\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='系统用户表';

### 8. 角色表 (sys\_role)

**功能说明**: 存储系统角色信息

CREATE TABLE `sys\_role` ( `role\_id` INT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '角色ID', `role\_code` VARCHAR(50) NOT NULL COMMENT '角色编码', `role\_name` VARCHAR(50) NOT NULL COMMENT '角色名称', `role\_level` TINYINT DEFAULT1 COMMENT '角色层级', `description` VARCHAR(200) DEFAULTNULL COMMENT '角色描述', `permission\_ids` TEXT DEFAULTNULL COMMENT '权限ID列表（逗号分隔）', `status` TINYINT NOT NULLDEFAULT1 COMMENT '状态: 0-停用, 1-启用', `sort\_order` INTDEFAULT0 COMMENT '排序序号', `created\_by` INT UNSIGNED DEFAULTNULL COMMENT '创建人ID', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', `updated\_by` INT UNSIGNED DEFAULTNULL COMMENT '更新人ID', `updated\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMPONUPDATECURRENT\_TIMESTAMP COMMENT '更新时间', `deleted` TINYINT NOT NULLDEFAULT0 COMMENT '逻辑删除: 0-未删除, 1-已删除', PRIMARY KEY (`role\_id`), UNIQUE KEY `uk\_role\_code` (`role\_code`), KEY `idx\_status` (`status`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='角色表';

**预置角色数据**:

INSERT INTO `sys\_role` (`role\_code`, `role\_name`, `description`) VALUES ('ADMIN', '系统管理员', '拥有所有权限'), ('OPERATOR', '系统操作员', '可以添加和修改参数'), ('VIEWER', '查看者', '仅可查看数据');

### 9. 权限表 (sys\_permission)

**功能说明**: 存储系统权限信息

CREATE TABLE `sys\_permission` ( `permission\_id` INT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '权限ID', `permission\_code` VARCHAR(100) NOT NULL COMMENT '权限编码', `permission\_name` VARCHAR(100) NOT NULL COMMENT '权限名称', `permission\_type` TINYINT NOT NULL COMMENT '权限类型: 1-菜单, 2-功能, 3-数据', `parent\_id` INT UNSIGNED DEFAULTNULL COMMENT '父权限ID', `resource\_path` VARCHAR(200) DEFAULTNULL COMMENT '资源路径', `description` VARCHAR(200) DEFAULTNULL COMMENT '权限描述', `status` TINYINT NOT NULLDEFAULT1 COMMENT '状态: 0-停用, 1-启用', `sort\_order` INTDEFAULT0 COMMENT '排序序号', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '创建时间', `updated\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMPONUPDATECURRENT\_TIMESTAMP COMMENT '更新时间', PRIMARY KEY (`permission\_id`), UNIQUE KEY `uk\_permission\_code` (`permission\_code`), KEY `idx\_parent\_id` (`parent\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='权限表';

**预置权限数据**:

INSERT INTO `sys\_permission` (`permission\_code`, `permission\_name`, `permission\_type`) VALUES ('param:view', '查看参数记录', 2), ('param:add', '添加参数记录', 2), ('param:edit', '编辑参数记录', 2), ('param:delete', '删除参数记录', 2), ('param:export', '导出参数记录', 2), ('param:approve', '审批参数修改', 2);

### 10. 操作日志表 (sys\_operation\_log)

**功能说明**: 记录所有用户操作，支持审计追溯

CREATE TABLE `sys\_operation\_log` ( `log\_id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '日志ID', `user\_id` INT UNSIGNED DEFAULTNULL COMMENT '操作人ID', `username` VARCHAR(50) DEFAULTNULL COMMENT '操作人用户名', `operation\_module` VARCHAR(50) NOT NULL COMMENT '操作模块', `operation\_type` VARCHAR(50) NOT NULL COMMENT '操作类型: 增/删/改/查/导出/登录等', `operation\_desc` VARCHAR(500) DEFAULTNULL COMMENT '操作描述', `request\_method` VARCHAR(10) DEFAULTNULL COMMENT '请求方法: GET/POST/PUT/DELETE', `request\_url` VARCHAR(500) DEFAULTNULL COMMENT '请求URL', `request\_params` TEXT DEFAULTNULL COMMENT '请求参数（JSON格式）', `response\_status` INTDEFAULTNULL COMMENT '响应状态码', `response\_message` VARCHAR(500) DEFAULTNULL COMMENT '响应消息', `ip\_address` VARCHAR(50) DEFAULTNULL COMMENT '操作IP', `browser` VARCHAR(100) DEFAULTNULL COMMENT '浏览器', `os` VARCHAR(100) DEFAULTNULL COMMENT '操作系统', `execution\_time` INTDEFAULTNULL COMMENT '执行时长（毫秒）', `error\_message` TEXT DEFAULTNULL COMMENT '错误信息', `created\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '操作时间', PRIMARY KEY (`log\_id`), KEY `idx\_user\_id` (`user\_id`), KEY `idx\_operation\_type` (`operation\_type`), KEY `idx\_created\_at` (`created\_at`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='操作日志表';

## 📊 视图设计

### 1. 参数修改记录列表视图 (v\_param\_modification\_list)

**功能说明**: 对应界面列表显示，整合多表数据

CREATEOR REPLACE VIEW `v\_param\_modification\_list` ASSELECT pmr.record\_id, ss.site\_name AS site\_name, sas.system\_name AS system\_name, sas.role\_type AS role\_type, sd.department\_name AS department\_name, pmr.modification\_count AS param\_count, pmr.last\_modified\_at AS update\_time, pmr.last\_modified\_by\_name AS modifier\_name, ss.site\_id, sas.system\_id, sd.department\_id FROM param\_modification\_record pmr LEFTJOIN sys\_automation\_system sas ON pmr.system\_id = sas.system\_id LEFTJOIN sys\_site ss ON pmr.site\_id = ss.site\_id LEFTJOIN sys\_department sd ON pmr.department\_id = sd.department\_id WHERE pmr.deleted =0AND sas.deleted =0;

**使用示例**:

-- 查询列表（对应界面显示）SELECT\*FROM v\_param\_modification\_list ORDERBY update\_time DESC LIMIT 10OFFSET10; -- 第2页，每页10条-- 按现场筛选SELECT\*FROM v\_param\_modification\_list WHERE site\_name IN ('广州', '郑州'); -- 多条件筛选SELECT\*FROM v\_param\_modification\_list WHERE site\_name ='广州'AND system\_name LIKE'%秦雷丝%'AND role\_type ='主用';

### 2. 参数修改详情视图 (v\_param\_modification\_detail)

**功能说明**: 查看参数修改的详细信息

CREATEOR REPLACE VIEW `v\_param\_modification\_detail` ASSELECT pmd.detail\_id, pmd.record\_id, sas.system\_name, ss.site\_name, pmd.modification\_type, pmd.param\_category, pmd.param\_name, pmd.old\_value, pmd.new\_value, pmd.modification\_reason, pmd.approval\_status, pmd.modified\_by\_name, pmd.modified\_at, u.real\_name AS approver\_name, pmd.approved\_at FROM param\_modification\_detail pmd LEFTJOIN param\_modification\_record pmr ON pmd.record\_id = pmr.record\_id LEFTJOIN sys\_automation\_system sas ON pmd.system\_id = sas.system\_id LEFTJOIN sys\_site ss ON pmr.site\_id = ss.site\_id LEFTJOIN sys\_user u ON pmd.approver\_id = u.user\_id;

## 🔍 索引优化建议

### 复合索引

-- param\_modification\_record 表ALTER TABLE param\_modification\_record ADD INDEX idx\_site\_system\_time (site\_id, system\_id, last\_modified\_at); -- param\_modification\_detail 表ALTER TABLE param\_modification\_detail ADD INDEX idx\_system\_param\_time (system\_id, param\_name, modified\_at); -- sys\_operation\_log 表ALTER TABLE sys\_operation\_log ADD INDEX idx\_user\_operation\_time (user\_id, operation\_type, created\_at);

### 全文索引（用于参数名称搜索）

ALTER TABLE param\_modification\_detail ADD FULLTEXT INDEX ft\_param\_name (param\_name); -- 使用示例SELECT\*FROM param\_modification\_detail WHEREMATCH(param\_name) AGAINST('速度 温度'INNATURALLANGUAGE MODE);

## 🔐 数据安全设计

### 1. 敏感数据加密

-- 密码字段使用 bcrypt 或 argon2 算法加密-- 示例（在应用层处理）:-- password: bcrypt.hash(password, 10)-- 敏感参数值加密存储ALTER TABLE param\_modification\_detail ADDCOLUMN `encrypted` TINYINT DEFAULT0 COMMENT '是否加密: 0-否, 1-是';

### 2. 审计触发器

-- 参数修改时自动更新主表计数 DELIMITER $$ CREATETRIGGER tr\_update\_modification\_count AFTER INSERTON param\_modification\_detail FOREACHROWBEGINUPDATE param\_modification\_record SET modification\_count = modification\_count +1, last\_modified\_at = NEW.modified\_at, last\_modified\_by = NEW.modified\_by, last\_modified\_by\_name = NEW.modified\_by\_name WHERE record\_id = NEW.record\_id; END$$ DELIMITER ;

### 3. 操作日志触发器

-- 记录参数修改操作日志 DELIMITER $$ CREATETRIGGER tr\_log\_param\_modification AFTER INSERTON param\_modification\_detail FOREACHROWBEGININSERT INTO sys\_operation\_log ( user\_id, username, operation\_module, operation\_type, operation\_desc, created\_at ) VALUES ( NEW.modified\_by, NEW.modified\_by\_name, 'param\_modification', NEW.modification\_type, CONCAT('修改参数: ', NEW.param\_name), NEW.modified\_at ); END$$ DELIMITER ;

## 📈 查询优化示例

### 1. 分页查询（高性能）

-- 推荐方式（使用主键范围）SELECT\*FROM v\_param\_modification\_list WHERE record\_id > ( SELECT record\_id FROM v\_param\_modification\_list ORDERBY record\_id LIMIT 1OFFSET10 ) ORDERBY record\_id LIMIT 10; -- 传统方式（性能较差）SELECT\*FROM v\_param\_modification\_list ORDERBY update\_time DESC LIMIT 10OFFSET10;

### 2. 多条件查询

-- 动态 SQL 示例（应用层实现）SELECT\*FROM v\_param\_modification\_list WHERE1=1AND (NULLIF(:site\_name, '') ISNULLOR site\_name = :site\_name) AND (NULLIF(:system\_name, '') ISNULLOR system\_name LIKE CONCAT('%', :system\_name, '%')) AND (NULLIF(:role\_type, '') ISNULLOR role\_type = :role\_type) AND (NULLIF(:department\_name, '') ISNULLOR department\_name = :department\_name) ORDERBY update\_time DESC LIMIT :limit OFFSET :offset;

### 3. 统计查询

-- 按现场统计参数修改次数SELECT ss.site\_name, COUNT(pmr.record\_id) AS record\_count, SUM(pmr.modification\_count) AS total\_modifications FROM sys\_site ss LEFTJOIN param\_modification\_record pmr ON ss.site\_id = pmr.site\_id WHERE ss.deleted =0GROUPBY ss.site\_id, ss.site\_name; -- 按系统统计SELECT sas.system\_name, sas.role\_type, COUNT(pmd.detail\_id) AS detail\_count, MAX(pmd.modified\_at) AS last\_modified\_at FROM sys\_automation\_system sas LEFTJOIN param\_modification\_detail pmd ON sas.system\_id = pmd.system\_id GROUPBY sas.system\_id, sas.system\_name, sas.role\_type;

## 🗄️ 数据归档策略

### 归档表设计

-- 创建归档表（与原表结构相同）CREATE TABLE `param\_modification\_detail\_archive` LIKE `param\_modification\_detail`; -- 添加归档时间字段ALTER TABLE `param\_modification\_detail\_archive` ADDCOLUMN `archived\_at` DATETIME NOT NULLDEFAULTCURRENT\_TIMESTAMP COMMENT '归档时间';

### 归档存储过程

DELIMITER $$ CREATEPROCEDURE sp\_archive\_old\_data(IN p\_days INT) BEGINDECLARE EXIT HANDLER FORSQLEXCEPTIONBEGINROLLBACK; SELECT'Archive failed'ASresult; END; START TRANSACTION; -- 归档 N 天前的数据INSERT INTO param\_modification\_detail\_archive SELECT\*, NOW() AS archived\_at FROM param\_modification\_detail WHERE modified\_at < DATE\_SUB(NOW(), INTERVAL p\_days DAY); -- 删除已归档数据DELETEFROM param\_modification\_detail WHERE modified\_at < DATE\_SUB(NOW(), INTERVAL p\_days DAY); COMMIT; SELECT'Archive successful'ASresult; END$$ DELIMITER ; -- 执行归档（归档1年前的数据）CALL sp\_archive\_old\_data(365);

## 💾 备份策略

### 1. 定期备份

#!/bin/bash# 每日备份脚本 BACKUP\_DIR="/backup/mysql" DB\_NAME="automation\_param\_system" DATE=$(date +%Y%m%d\_%H%M%S) mysqldump -u root -p${DB\_PASSWORD} \ --single-transaction \ --routines \ --triggers \ --events \ ${DB\_NAME} > ${BACKUP\_DIR}/${DB\_NAME}\_${DATE}.sql # 压缩备份文件 gzip ${BACKUP\_DIR}/${DB\_NAME}\_${DATE}.sql # 删除7天前的备份 find ${BACKUP\_DIR} -name "\*.sql.gz" -mtime +7 -delete

### 2. 增量备份

# 启用二进制日志# my.cnf 配置 [mysqld] log-bin=mysql-bin binlog-format=ROW expire\_logs\_days=7 # 增量备份脚本 mysqlbinlog mysql-bin.000001 > incremental\_backup.sql

## 📊 性能监控

### 慢查询日志

-- my.cnf 配置 [mysqld] slow\_query\_log =1 slow\_query\_log\_file =/var/log/mysql/slow-query.log long\_query\_time =2-- 分析慢查询SELECT\*FROM mysql.slow\_log ORDERBY query\_time DESC LIMIT 10;

### 表统计信息

-- 查看表大小SELECT table\_name, ROUND(((data\_length + index\_length) /1024/1024), 2) AS size\_mb, table\_rows FROM information\_schema.TABLES WHERE table\_schema ='automation\_param\_system'ORDERBY size\_mb DESC; -- 查看索引使用情况SELECT table\_name, index\_name, cardinality, seq\_in\_index FROM information\_schema.STATISTICS WHERE table\_schema ='automation\_param\_system'ORDERBY table\_name, index\_name;

## 🔧 维护建议

### 1. 定期优化

-- 分析表 ANALYZE TABLE param\_modification\_record; ANALYZE TABLE param\_modification\_detail; -- 优化表 OPTIMIZE TABLE param\_modification\_record; OPTIMIZE TABLE param\_modification\_detail; -- 检查表CHECKTABLE param\_modification\_record;

### 2. 数据清理

-- 清理逻辑删除的数据（定期执行）DELETEFROM sys\_site WHERE deleted =1AND updated\_at < DATE\_SUB(NOW(), INTERVAL90DAY); DELETEFROM sys\_automation\_system WHERE deleted =1AND updated\_at < DATE\_SUB(NOW(), INTERVAL90DAY);

### 3. 监控建议

* 监控表空间大小
* 监控慢查询数量
* 监控连接数
* 监控锁等待
* 定期审查索引使用率

## 📝 附录

### A. ER 图说明

sys\_site (运行现场) ↓ 1:N sys\_automation\_system (自动化系统) ↓ 1:N param\_modification\_record (参数修改记录) ↓ 1:N param\_modification\_detail (参数修改详情) ↓ 1:N param\_version\_history (版本历史) sys\_department (维护部门) ↓ 1:N sys\_user (系统用户) ↓ N:M (通过 role\_ids) sys\_role (角色) ↓ N:M (通过 permission\_ids) sys\_permission (权限)

### B. 命名规范

* 表名：小写字母+下划线，使用前缀区分模块（sys\_、param\_）
* 字段名：小写字母+下划线
* 索引名：idx\_ + 字段名
* 唯一索引：uk\_ + 字段名
* 外键：fk\_ + 表名简写 + 字段名

### C. 数据字典

详细的数据字典请参考各表结构说明。

**文档版本**: 1.0  
**创建日期**: 2025-01-16  
**维护人**: 数据库管理员  
**审核状态**: 待审核