学院生活平台系统

数据库设计说明书

班级：　　学号：　　姓名：

1 引言

1.1 编写目的

本文档旨在详细说明校内综合服务平台数据库的设计方案，包括数据库的结构、表设计、功能关联等核心内容。其目的是为软件开发人员、测试人员、维护人员提供清晰的数据库设计依据，确保各模块功能有序实现，同时为后续系统升级、数据维护和扩展提供参考标准，保障数据库设计的规范性、一致性和可扩展性。

1.2 背景

a. 待开发的数据库名称为 “校内综合服务平台数据库”，使用该数据库的软件系统为 “校内综合服务平台”。

b. 该软件系统开发项目的任务提出者为校内管理部门，用户为全校师生，将安装该软件和数据库的计算站为校内信息中心服务器集群。

1.3 定义

专门术语	定义
RBAC	基于角色的访问控制（Role-Based Access Control），通过角色关联用户与权限，实现权限的灵活配与管理
OSS	对象存储服务（Object Storage Service），用于存储系统中的文件资源（如文档、图片等）
逻辑删除	通过设置删除标志（del_flag）标识数据状态，而非直接删除数据记录，保留数据追溯能力
冗余字段	为提高查询效率或满足业务需求，在表中重复存储的关联表字段（如上传者用户名快照）
幂等性	指同一操作多次执行所产生的结果与一次执行的结果相同，避免重复数据产生
JSON	JavaScript 对象表示法（JavaScript Object Notation），用于存储和交换结构化数据，此处用于储扩展配置、多选答案等非固定格式数据

1.4 参考资料

a. 本项目的经核准的计划任务书《校内综合服务平台开发项目任务书》；b. 本项目的相关文件《校内综合服务平台需求规格说明书》《校内综合服务平台总体设计方案》；c. 参考的软件开发标准《GB/T 8567-2006 计算机软件文档编制规范》《数据库设计规范 V3.0》。

2 外部设计

2.1 标识符和状态

数据库标识符	描述	状态	有效时间范围
school_service_platform_db	校内综合服务平台核心数据库，存储各功能模块业务数据	正式运行	长期有效

2.2 使用它的程序

应用程序名称	版本号	功能说明
校内综合服务平台 - Web 端	V1.0	支持师生用户进行课程资源分享、功能房预约、数字图书馆访问、失物领发布、问卷填写、通知公告查看等操作
校内综合服务平台 - 管理后台	V1.0	支持管理员进行用户管理、权限配置、数据审核、系统参数设置等管理作
校内综合服务平台 - 移动端适配版	V1.0	支持移动端用户进行核心功能操作（如资源下载、预约申请、通知查看等

2.3 约定

2.3.1 数据库版本标识约定

数据库版本采用 “主版本号。次版本号。修订号” 格式，例如 V1.0.0，其中：

主版本号：数据库结构发生重大变更（如核心表新增 / 删除）时递增；

次版本号：数据库结构发生中度变更（如字段新增 / 修改）时递增；

修订号：数据库 Bug 修复、索引优化等小范围调整时递增。

2.3.2 命名约定

对象类型	命名规则	示例
数据库	采用小写字母 + 下划线命名，前缀为 “school_service_”，后缀为功能标识	school_service_platform_db
表	采用小写字母 + 下划线命名，前缀为模块缩写，格式为 “tb_模块标识_表功能标识”	tb_course_resource（课程资源主表）、tb_facility_booking（功能房预约表）
字段	采用小写字母 + 下划线命名，遵循 “语义化” 原则，避免模糊命名	user_id（用户 ID）、create_time（创建时间）download_count（下载次数）
主键	统一命名为 “id”，数据类型统一为 bigint (20)，自增（AUTO_INCREMENT）	id bigint(20) NOT NULL AUTO_INCREMENT
外键	命名格式为 “关联表标识_id”，如关联专业表	major_id bigint (20) NOT NULL（关联

	（tb_major）则命名为“major_id”	tb_major.id）
索引	普通索引前缀为“idx_”，唯一索引前缀为“uk_”，格式为“前缀_表标识_字段组合”	idx_course_status（课程表状态索引）、uk_major_name（专业表名称唯一索引）
状态字段	统一命名为“status”，根据模块需求定义枚举值，采用 char (1) 或 tinyint (4) 类型	status char (1) DEFAULT '0'（0 正常 1 停用）
删除标志	统一命名为“del_flag”，固定枚举值（0 = 存在，2 = 删除），数据类型为 char (1)	

2.3.3 字段类型约定

数据类型	使用场景
bigint(20)	主键 ID、关联表 ID、大数值统计（如下载次数、文件大小）
varchar(n)	字符串类型数据（如名称、描述、URL 等），n 根据业务长度需求定义（如名称 255、URL512）
char(1)	状态标识、布尔型字段（如是否必填、是否解决等）
tinyint(4)	小范围枚举值（如资源类型、题型、范围类型等）
int(11)	中等数值（如排序号、容量、时长等）
datetime	时间字段（如创建时间、更新时间、预约时间等），存储格式为“YYYY-MM-DD HH:MM:SS”
text	长文本数据（如资源简介、图书摘要、正文描述等）
json	非固定格式的结构化数据（如扩展配置、多选答案等）

2.4 专门指导

2.4.1 数据格式标准

数据类型	格式要求	示例
日期时间	统一采用“YYYY-MM-DD HH:MM:SS”格式，禁止使用非标准格式（如“MM/DD/YYYY”）	2024-05-20 14:30:00
URL地址	需包含完整协议头（http:// 或 https://），确保可直接访问	https://oss.example.com/resources/202405/file.pdf
枚举	严格遵循表设计中定义的枚举范围，禁止	审核状态仅允许 0（待审）、1（已通过）、2（驳回）

值	输入无效值（如状态字段仅允许 0/1/2 等指定值）	3（已下架）
文本数据	禁止包含特殊字符（如换行符、制表符、脚本标签等），需经过前端过滤和后端校验	正文描述中不可包含<script>等危险标签
数值型 ID	仅允许输入数字，禁止包含字母、符号等非数值字符	user_id 仅允许输入整数（如 1001、2023 等）

2.4.2 数据操作流程

数据插入流程：

1.前端校验：验证字段格式、非空约束、长度限制等；

2.后端校验：二次验证数据合法性，补充默认值（如 create_time、update_time 自动填充当前时间）；

3.插入执行：执行 SQL 插入语句，返回插入结果（成功 / 失败）及主键 ID。

数据更新流程：

1.权限校验：验证操作人是否有权限更新该数据；

2.数据校验：验证更新字段的合法性（如状态流转是否符合规则）；

3.更新执行：更新业务字段，同步更新 update_by（更新者）和 update_time（更新时间）字段。

数据删除流程：

1.统一采用逻辑删除，设置 del_flag=2，不执行物理删除；

2.关联数据处理:若主表数据逻辑删除,其关联子表数据需同步设置 del_flag=2(可通过触发器或业务逻辑实现)。

数据查询流程：

1.默认查询条件包含 del_flag=0（仅查询未删除数据）；

2.关联查询时，需过滤关联表的删除标志，避免查询到无效数据。

2.4.3 参考文件

《校内综合服务平台数据操作手册》

《数据库 SQL 编写规范》

《OSS 文件上传与访问指南》

2.5 支持软件

支持软件名称	版本号	主要功能特性	技术文件来源
MySQL	8.0.x	支持 InnoDB 存储引擎、JSON 数据类型、事务 ACID 特性、索引优化；支持最大数据容量根据存储设备扩展，单表支持千万级数据存储	MySQL 官方文档（https://dev.mysql.com/doc/）
阿里云 OSS	最新	提供高可用、高并发的	阿里云 OSS 官方文档

	稳定版	文件存储服务，支持文件上传、下载、删除、访问权限控制；支持 PB 级存储容量，支持 CDN 加速访问	（https://help.aliyun.com/document_detail/31817.htm）
Redis	6.2.x	支持缓存、计数器、排行榜等功能；用于存储热点数据（如排行榜信息）、会话数据，提升系统查询性能	Redis 官方文档（https://redis.io/documentation/）
MyBatis-Plus	3.5.x	数据库访问框架，支持 CRUD 操作、条件查询、分页、逻辑删除自动处理等；简化 SQL 编写，提升开发效率	

3　结构设计

3．1 概念结构设计

3.1.1 实体及属性定义

系统核心实体包括用户、专业、课程、课程资源、功能房、预约单、图书、失物招领信息、问卷、通知公告等，各实体核心属性如下：

实体名称	核心属性
用户（User）	用户 ID、用户名、角色、所属部门 / 专业等
专业（Major）	专业 ID、专业名称、状态、负责人等
课程（Course）	课程 ID、课程名称、所属专业 ID、课程编码、状态等
课程资源（CourseResource）	资源 ID、所属专业 ID、所属课程 ID、资源名称、资源类型、URL / 外链、传者、审核状态等
楼房（Building）	楼房 ID、楼房名称、状态、备注等
功能房（Room）	房间 ID、所属楼房 ID、楼层号、房间名称、容量、设备标签、状态等
预约单（Booking）	预约 ID、房间 ID、申请人 ID、使用目的、预约时间、时长、状态等
图书（Book）	图书 ID、ISBN-13、书名、作者、出版社、出版年份、审核状态、下载次数

图书资产（BookAsset）	资产 ID、图书 ID、资产类型、文件格式、URL / 外链、文件大小等
失物招领信息（LostItem）	信息 ID、类型（丢失 / 捡到）、标题、内容、地点、发生时间、状态、是否解决等
失物招领图片（LostItemImage）	图片 ID、所属信息 ID、图片 URL、排序号等
问卷（Survey）	问卷 ID、标题、状态、可见范围、截止时间、置顶标识等
问卷题目（SurveyItem）	题目 ID、所属问卷 ID、题目内容、题型、是否必填、排序号等
问卷选项（SurveyOption）	选项 ID、所属题目 ID、选项文本、选项值、排序号等
答卷（SurveyAnswer）	答卷 ID、所属问卷 ID、提交人 ID、提交时间、最近修改时间等
答案明细（SurveyAnswerItem）	明细 ID、所属答卷 ID、所属题目 ID、文本题答案、选择题答案等
通知公告（Notice）	公告 ID、标题、内容、类型、状态、可见范围、发布者、发布时间等
公告附件（NoticeAttachment）	附件 ID、所属公告 ID、文件名、文件 URL、文件类型、文件大小等
公告阅读回执（NoticeRead）	回执 ID、所属公告 ID、用户 ID、阅读时间、是否确认等

3.1.2 实体关系定义

各实体间的核心关系如下：

1.专业与课程：一对多关系（一个专业包含多个课程，一个课程仅属于一个专业）；

2.专业与专业负责人：一对多关系（一个专业可指定多个负责人，一个用户可担任多个专业负责人）；

3.课程与课程资源：一对多关系（一个课程包含多个资源，一个资源仅属于一个课程）；

4.楼房与功能房：一对多关系（一个楼房包含多个功能房，一个功能房仅属于一个楼房）；

5.功能房与预约单：一对多关系（一个功能房可产生多个预约单，一个预约单仅关联一个功能房）；

6.预约单与使用人：一对多关系（一个预约单可包含多个使用人，一个用户可参与多个预约单）；

7.图书与图书资产：一对多关系（一本图书可对应多个资产（不同格式文件 / 外链），一个资产仅属于一本图书）；

8.图书与收藏用户：多对多关系（一本图书可被多个用户收藏，一个用户可收藏多本图书，通过收藏关系表关联）；

9.失物招领信息与图片：一对多关系（一条失物招领信息可包含多张图片，一张图片仅属于一条信息）；

10.问卷与题目：一对多关系（一个问卷包含多个题目，一个题目仅属于一个问卷）；

11.题目与选项：一对多关系（一个单选 / 多选题包含多个选项，一个选项仅属于一个题目）；

12.问卷与答卷：一对多关系（一个问卷可产生多个答卷，一个用户对一个问卷仅能提交一份答卷）；

13.答卷与答案明细：一对多关系（一份答卷包含多个题目对应的答案明细，一个答案明细仅属于一份答卷）；

14.通知公告与附件：一对多关系（一条公告可包含多个附件，一个附件仅属于一条公告）；

15.通知公告与阅读用户：多对多关系（一条公告可被多个用户阅读，一个用户可阅读多条公告，通过阅读回执表关联）。

3.1.3 用户视图

根据不同用户角色，系统提供以下核心用户视图：

1,普通用户视图：

课程资源视图：显示已审核通过的课程资源列表，包含资源名称、所属专业 / 课程、上传者、下载次数等信息；
功能房预约视图：显示可预约的功能房列表、预约状态、自身预约记录等信息；
数字图书馆视图：显示已发布的图书列表，包含书名、作者、出版社、下载次数等信息；
失物招领视图：显示已发布的失物招领信息列表，包含标题、类型、地点、发布时间等信息；
问卷填写视图：显示自己有权限填写的问卷列表及问卷题目、选项等信息；
通知公告视图：显示自己有权限查看的通知公告列表及公告内容、附件等信息。
2.专业负责人视图：
课程资源审核视图：显示本专业待审核的课程资源列表，包含资源名称、上传者、提交时间等信息；
3.图书管理员视图：
图书审核视图：显示待审核的图书列表，包含书名、作者、ISBN、上传者等信息；
4.系统管理员视图：
全量数据管理视图：包含各模块数据的查询、审核、修改、删除等操作入口，支持数据统计与导出。

3.2.1 全局逻辑结构概述

数据库采用模块化表设计，按功能模块划分为课程资源分享模块、功能房预约模块、数字图书馆模块、失物招领模块、问卷模块、通知公告模块，各模块表通过外键关联形成全局逻辑结构。核心设计原则如下：
按业务模块垂直拆分表，降低表间耦合度；
核心表（如资源表、图书表、预约表）包含完整的业务字段，关联表（如负责人映射表、收藏关系表）仅存储关联关系；
统一设计审计字段（create_by、create_time、update_by、update_time、del_flag），确保数据操作可追溯；
合理设置唯一索引，避免重复数据（如专业名称全局唯一、同专业内课程名称唯一）；
针对查询频繁的字段（如状态、创建时间、关联 ID）建立普通索引，提升查询性能。

3.2.2 核心表逻辑结构

各模块核心表的逻辑结构及关键字、关联关系如下：**一．课程资源分享模块**

1) tb_major（专业表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	专业主键 ID
major_name	varchar(128)	-	是	专业名称，全局唯一
status	char(1)	'0'	是	状态(0 正常 1 停用),停用后不可在新增课程源时选择
remark	varchar(500)	-	否	备注
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间

del_flag	char(1)	'0'	是	软删标记（0 存在 2 删除）

SQL 代码：

```
CREATE TABLE `tb_major` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '专业主键 ID',
  `major_name` varchar(128) NOT NULL COMMENT '专业名称',
  `status` char(1) DEFAULT '0' COMMENT '状态（0 正常 1 停用）',
  `remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_major_name` (`major_name`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='专业表';
```

2)tb_course（课程表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	课程主键 ID
major_id	bigint(20)	-	是	所属专业 ID（关联 tb_major.id）
course_name	varchar(255)	-	是	课程名称，同专业内唯一
course_code	varchar(64)	-	否	课程编码（可选）
status	char(1)	'0'	是	状态（0 正常 1 停用）
remark	varchar(500)	-	否	备注
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	'0'	是	软删标记（0 存在 2 删除）

SQL 代码：

```
CREATE TABLE `tb_course` (
```

```
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '课程主键 ID',
  `major_id` bigint(20) NOT NULL COMMENT '所属专业 ID',
  `course_name` varchar(255) NOT NULL COMMENT '课程名称',
  `course_code` varchar(64) DEFAULT NULL COMMENT '课程编码',
  `status` char(1) DEFAULT '0' COMMENT '状态（0 正常 1 停用）',
  `remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_course_unique` (`major_id`,`course_name`) USING BTREE,
  KEY `idx_course_major` (`major_id`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='课程表';
```

3)tb_major_lead（专业负责人映射表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID
major_id	bigint(20)	-	是	专业 ID（关联 tb_major.id）
user_id	bigint(20)	-	是	负责人用户 ID（关联 sys_user.id）
remark	varchar(500)	-	否	备注
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	'0'	是	软删标记（0 存在 2 删除）

SQL 代码：

```
CREATE TABLE `tb_major_lead` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `major_id` bigint(20) NOT NULL COMMENT '专业 ID',
  `user_id` bigint(20) NOT NULL COMMENT '负责人用户 ID',
  `remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
```

`create_time` datetime NOT NULL COMMENT '创建时间',
`update_by` varchar(64) NOT NULL COMMENT '更新者',
`update_time` datetime NOT NULL COMMENT '更新时间',
`del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
PRIMARY KEY (`id`) USING BTREE,
UNIQUE KEY `uk_major_user` (`major_id`,`user_id`) USING BTREE,
KEY `idx_lead_user` (`user_id`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='专业负责人映射表';

4)tb_course_resource（课程资源主表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	资源主键 ID
major_id	bigint(20)	-	是	专业 ID（冗余，与课程所属专业一致）
course_id	bigint(20)	-	是	课程 ID（关联 tb_course.id）
resource_name	varchar(255)	-	是	资源显示名
resource_type	tinyint(4)	-	是	资源类型（0 - 文件 1 - 外链）
file_url	varchar(512)	-	否	文件直链（资源类型为 0 时必填）
file_hash	varchar(128)	-	否	文件哈希（资源类型为 0 时必填）
file_size	bigint(20)	-	否	文件大小（字节）
link_url	varchar(512)	-	否	外链 URL（资源类型为 1 时必填）
description	text	-	是	资源简介
status	tinyint(4)	0	是	审核状态（0 待审 1 已通过 2 驳回 3 架）
audit_by	varchar(64)	-	否	审核人
audit_time	datetime	-	否	审核时间
audit_reason	varchar(512)	-	否	审核原因
publish_time	datetime	-	否	首次上架时间

download_count	int(11)	0	是	下载次数
last_download_time	datetime	-	否	最近下载时间
uploader_id	bigint(20)	-	是	上传者用户 ID
uploader_name	varchar(64)	-	是	上传者用户名快照
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	’0’	是	软删标记（0 存在 2 删除）

SQL 代码：

```
CREATE TABLE `tb_course_resource` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '资源主键 ID',
  `major_id` bigint(20) NOT NULL COMMENT '专业 ID',
  `course_id` bigint(20) NOT NULL COMMENT '课程 ID',
  `resource_name` varchar(255) NOT NULL COMMENT '资源显示名',
  `resource_type` tinyint(4) NOT NULL COMMENT '资源类型（0-文件 1-外链）',
  `file_url` varchar(512) DEFAULT NULL COMMENT '文件直链',
  `file_hash` varchar(128) DEFAULT NULL COMMENT '文件哈希',
  `file_size` bigint(20) DEFAULT NULL COMMENT '文件大小（字节）',
  `link_url` varchar(512) DEFAULT NULL COMMENT '外链 URL',
  `description` text NOT NULL COMMENT '资源简介',
  `status` tinyint(4) DEFAULT 0 COMMENT '审核状态（0 待审 1 已通过 2 驳回 3 已下架）',
  `audit_by` varchar(64) DEFAULT NULL COMMENT '审核人',
  `audit_time` datetime DEFAULT NULL COMMENT '审核时间',
  `audit_reason` varchar(512) DEFAULT NULL COMMENT '审核原因',
  `publish_time` datetime DEFAULT NULL COMMENT '首次上架时间',
  `download_count` int(11) DEFAULT 0 COMMENT '下载次数',
  `last_download_time` datetime DEFAULT NULL COMMENT '最近下载时间',
  `uploader_id` bigint(20) NOT NULL COMMENT '上传者用户 ID',
  `uploader_name` varchar(64) NOT NULL COMMENT '上传者用户名',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
```

PRIMARY KEY (`id`) USING BTREE,
UNIQUE KEY `uk_course_filehash` (`course_id`,`resource_type`,`file_hash`) USING BTREE,
UNIQUE KEY `uk_course_linkurl` (`course_id`,`resource_type`,`link_url`(191)) USING BTREE,
KEY `idx_course_status` (`course_id`,`status`) USING BTREE,
KEY `idx_status_time` (`status`,`create_time`) USING BTREE,
KEY `idx_download_count` (`download_count`) USING BTREE,
KEY `idx_uploader_id` (`uploader_id`) USING BTREE,
KEY `idx_major_course` (`major_id`,`course_id`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='课程资源主表';

5)tb_course_resource_best（最佳标记表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID
resource_id	bigint(20)	-	是	资源 ID（关联 tb_course_resource.id）
best_by	varchar(64)	-	是	标记人（用户名快照）
best_time	datetime	-	是	标记时间
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	’0’	是	软删标记（0 存在 2 删除）

SQL 代码：

CREATE TABLE `tb_course_resource_best` (
`id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
`resource_id` bigint(20) NOT NULL COMMENT '资源 ID',
`best_by` varchar(64) NOT NULL COMMENT '标记人',
`best_time` datetime NOT NULL COMMENT '标记时间',
`create_by` varchar(64) NOT NULL COMMENT '创建者',
`create_time` datetime NOT NULL COMMENT '创建时间',
`update_by` varchar(64) NOT NULL COMMENT '更新者',
`update_time` datetime NOT NULL COMMENT '更新时间',
`del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
PRIMARY KEY (`id`) USING BTREE,
UNIQUE KEY `uk_best_resource` (`resource_id`) USING BTREE) ENGINE=InnoDB DEFAULT

CHARSET=utf8mb4 COMMENT='最佳标记表';

6)tb_course_resource_log（操作审计表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID
resource_id	bigint(20)	-	是	资源 ID（关联 tb_course_resource.id）
action	varchar(32)	-	是	操作类型（CREATE/EDIT/APPROVE 等）
actor_id	bigint(20)	-	否	操作者用户 ID
actor_name	varchar(64)	-	否	操作者用户名
ip	varchar(64)	-	否	访问 IP
user_agent	varchar(255)	-	否	用户代理信息
detail	text	-	否	操作详情（JSON 格式）
result	varchar(16)	-	否	操作结果（SUCCESS/FAIL）
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	'0'	是	软删标记（0 存在 2 删除）

SQL 代码：

```
CREATE TABLE `tb_course_resource_log` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `resource_id` bigint(20) NOT NULL COMMENT '资源 ID',
  `action` varchar(32) NOT NULL COMMENT '操作类型',
  `actor_id` bigint(20) DEFAULT NULL COMMENT '操作者用户 ID',
  `actor_name` varchar(64) DEFAULT NULL COMMENT '操作者用户名',
  `ip` varchar(64) DEFAULT NULL COMMENT '访问 IP',
  `user_agent` varchar(255) DEFAULT NULL COMMENT '用户代理',
  `detail` text COMMENT '操作详情',
  `result` varchar(16) DEFAULT NULL COMMENT '操作结果',
```

`create_by` varchar(64) NOT NULL COMMENT '创建者',
`create_time` datetime NOT NULL COMMENT '创建时间',
`update_by` varchar(64) NOT NULL COMMENT '更新者',
`update_time` datetime NOT NULL COMMENT '更新时间',
`del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
PRIMARY KEY (`id`) USING BTREE,
KEY `idx_log_resource_time` (`resource_id`,`create_time`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='操作审计表';

7)tb_cr_user_score（用户积分聚合表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID
user_id	bigint(20)	-	是	用户 ID（关联 sys_user.id）
username	varchar(64)	-	是	用户名快照
major_id	bigint(20)	0	是	专业 ID（0 表示全站聚合）
total_score	int(11)	0	是	累计积分
approve_count	int(11)	0	是	审核通过次数
best_count	int(11)	0	是	被评为最佳次数
remark	varchar(255)	-	否	备注
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	’0’	是	软删标记（0 存在 2 删除）

SQL 代码：

CREATE TABLE `tb_cr_user_score` (
`id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
`user_id` bigint(20) NOT NULL COMMENT '用户 ID',
`username` varchar(64) NOT NULL COMMENT '用户名',
`major_id` bigint(20) NOT NULL DEFAULT 0 COMMENT '专业 ID（0 表示全站）',

`total_score` int(11) DEFAULT 0 COMMENT '累计积分',
`approve_count` int(11) DEFAULT 0 COMMENT '审核通过次数',
`best_count` int(11) DEFAULT 0 COMMENT '被评为最佳次数',
`remark` varchar(255) DEFAULT NULL COMMENT '备注',
`create_by` varchar(64) NOT NULL COMMENT '创建者',
`create_time` datetime NOT NULL COMMENT '创建时间',
`update_by` varchar(64) NOT NULL COMMENT '更新者',
`update_time` datetime NOT NULL COMMENT '更新时间',
`del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
PRIMARY KEY (`id`) USING BTREE,
UNIQUE KEY `uk_user_major` (`user_id`,`major_id`) USING BTREE,
KEY `idx_major_score` (`major_id`,`total_score`) USING BTREE,
KEY `idx_user` (`user_id`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='用户积分聚合表';

8)tb_cr_user_score_log（用户积分流水表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID
user_id	bigint(20)	-	是	用户 ID（资源上传者，关联 sys_user.id）
username	varchar(64)	-	是	用户名快照
major_id	bigint(20)	0	是	专业 ID（0 表示全站维度）
resource_id	bigint(20)	-	是	资源 ID（关联 tb_course_resource.id）
event_type	varchar(16)	-	是	事件类型（APPROVE/BEST）
delta	int(11)	-	是	积分变动（正数）
remark	varchar(255)	-	否	备注
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	'0'	是	软删标记（0 存在 2 删除）

SQL 代码：

```
CREATE TABLE `tb_cr_user_score_log` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `user_id` bigint(20) NOT NULL COMMENT '用户 ID',
  `username` varchar(64) NOT NULL COMMENT '用户名',
  `major_id` bigint(20) NOT NULL DEFAULT 0 COMMENT '专业 ID（0 表示全站）',
  `resource_id` bigint(20) NOT NULL COMMENT '资源 ID',
  `event_type` varchar(16) NOT NULL COMMENT '事件类型（APPROVE/BEST）',
  `delta` int(11) NOT NULL COMMENT '积分变动',
  `remark` varchar(255) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) DEFAULT '0' COMMENT '软删标记（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_once_event` (`user_id`,`resource_id`,`event_type`) USING BTREE,
  KEY `idx_user_time` (`user_id`,`create_time`) USING BTREE,
  KEY `idx_major_time` (`major_id`,`create_time`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='用户积分流水表';
```

二．功能房预约模块

1) tb_facility_building（楼房表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	楼房主键，自增生成，唯一标识每栋楼房
building_name	varchar(128)	-	是	楼房名称，全局唯一
status	char(1)	’0’	是	状态：0 - 正常，1 - 停用（停用则其下房间不可用）
remark	varchar(500)	NULL	否	备注信息
create_by	varchar(64)	-	是	记录创建者（用户名或系统标识），用于操作
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者（创建者、管理员或系统标识），操作审计
update_time	datetime	-	是	记录更新时间，信息变更时自动更新

del_flag	char(1)	’0’	是	删除标志：0 - 记录存在，2 - 记录删除（逻除）

SQL 代码：

```
CREATE TABLE `tb_facility_building` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '楼房主键',
  `building_name` varchar(128) NOT NULL COMMENT '楼房名称（唯一）',
  `status` char(1) DEFAULT '0' COMMENT '状态（0-正常 1-停用）',
  `remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_building_name` (`building_name`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='功能房-楼房表';
```

2)tb_facility_room（功能房表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	房间主键，自增生成，唯一标识每个功能房
building_id	bigint(20)	-	是	楼房 ID，逻辑外键关联 tb_facility_building.id，标识所属楼房
floor_no	int(11)	-	是	楼层号（可为负数代表地下楼层）
room_name	varchar(128)	-	是	房间名称 / 编号，同楼房同楼层内唯一
capacity	int(11)	NULL	否	房间容量（可容纳人数）
equipment_tags	varchar(255)	NULL	否	设备 / 标签（逗号分隔，如 “投影，白板”
status	char(1)	’0’	是	状态：0 - 启用，1 - 禁用
remark	varchar(500)	NULL	否	备注信息
create_by	varchar(64)	-	是	记录创建者，用于操作审计
create_time	datetime	-	是	记录创建时间

update_by	varchar(64)	-	是	记录更新者，用于操作审计
update_time	datetime	-	是	记录更新时间，信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 - 记录存在，2 - 记录删除（逻除）

SQL 代码：

```
CREATE TABLE `tb_facility_room` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '房间主键',
  `building_id` bigint(20) NOT NULL COMMENT '楼房 ID（逻辑外键：tb_facility_building.id）',
  `floor_no` int(11) NOT NULL COMMENT '楼层号',
  `room_name` varchar(128) NOT NULL COMMENT '房间名称/编号',
  `capacity` int(11) DEFAULT NULL COMMENT '容量（人数）',
  `equipment_tags` varchar(255) DEFAULT NULL COMMENT '设备/标签（逗号分隔）',
  `status` char(1) DEFAULT '0' COMMENT '状态（0-启用 1-禁用）',
  `remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_room_name` (`building_id`,`floor_no`,`room_name`) USING BTREE,
  KEY `idx_room_building_floor` (`building_id`,`floor_no`) USING BTREE,
  KEY `idx_room_status` (`status`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='
功能房-房间表';
```

3)tb_facility_booking（功能房预约表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	预约主键，自增生成，唯一标识每个预约单
room_id	bigint(20)	-	是	房间 ID，逻辑外键关联 tb_facility_room. 标识预约的房间
applicant_id	bigint(20)	-	是	申请人用户 ID，逻辑外键关联 sys_user.use
purpose	varchar(200)	-	是	预约使用目的（1~200 字）
start_time	datetime	-	是	预约开始时间（建议 30 分钟粒度）

end_time	datetime	-	是	预约结束时间（需大于开始时间且不超过全局上限）
duration_minutes	int(11)	-	是	预约时长（分钟）
status	char(1)	'0'	是	状态：0 - 待审、1 - 已批准、2 - 已驳回、3 取消、4 - 进行中、5 - 已完成
reject_reason	varchar(500)	NULL	否	驳回理由（仅状态为 “已驳回” 时必填）
create_by	varchar(64)	-	是	记录创建者，用于操作审计
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者，用于操作审计
update_time	datetime	-	是	记录更新时间，信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 - 记录存在，2 - 记录删除（逻除）

SQL 代码：

```
CREATE TABLE `tb_facility_booking` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '预约主键',
  `room_id` bigint(20) NOT NULL COMMENT '房间 ID（逻辑外键：tb_facility_room.id）',
  `applicant_id` bigint(20) NOT NULL COMMENT '申请人用户 ID（逻辑外键：sys_user.user_id）',
  `purpose` varchar(200) NOT NULL COMMENT '使用目的',
  `start_time` datetime NOT NULL COMMENT '开始时间',
  `end_time` datetime NOT NULL COMMENT '结束时间',
  `duration_minutes` int(11) NOT NULL COMMENT '时长（分钟）',
  `status` char(1) DEFAULT '0' COMMENT '状态（0 待审 1 已批准 2 已驳回 3 已取消 4 进行中 5 已完成）',
  `reject_reason` varchar(500) DEFAULT NULL COMMENT '驳回理由',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_booking_once` (`room_id`,`applicant_id`,`start_time`,`end_time`) USING BTREE,
  KEY `idx_booking_room_time` (`room_id`,`start_time`,`end_time`) USING BTREE,
  KEY `idx_booking_applicant_time` (`applicant_id`,`start_time`) USING BTREE,
  KEY `idx_booking_status` (`status`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='功能房-预约表';
```

4)tb_facility_booking_user（预约使用人关联表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键，自增生成，唯一标识每条关联记录
booking_id	bigint(20)	-	是	预约 ID，逻辑外键关联 tb_facility_booking.
user_id	bigint(20)	-	是	用户 ID，逻辑外键关联 sys_user.user_id，标 与使用的人员
is_applicant	char(1)	'0'	是	是否为申请人：1 - 是，0 - 否（申请人也需入
create_by	varchar(64)	-	是	记录创建者，用于操作审计
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者，用于操作审计
update_time	datetime	-	是	记录更新时间，信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 - 记录存在，2 - 记录删除（逻辑

SQL 代码：

```
CREATE TABLE `tb_facility_booking_user` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
  `booking_id` bigint(20) NOT NULL COMMENT '预约 ID（逻辑外键：tb_facility_booking.id）',
  `user_id` bigint(20) NOT NULL COMMENT '用户 ID（逻辑外键：sys_user.user_id）',
  `is_applicant` char(1) DEFAULT '0' COMMENT '是否申请人（1 是 0 否）',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_booking_user` (`booking_id`,`user_id`) USING BTREE,
  KEY `idx_booking_user_user` (`user_id`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='功能房-预约使用人关联表';
```

5)tb_facility_setting（功能房模块设置表）

字段名	数据类型	默认值	非空约束	字段说明

id	bigint(20)	-	是	主键（建议固定 id=1），唯一标识全置记录
audit_required	char(1)	'1'	是	是否需要审核：1 - 是，0 - 否
max_duration_minutes	int(11)	4320	是	最长使用时长（分钟，默认 72 小时）
remark	varchar(500)	NULL	否	备注信息
create_by	varchar(64)	-	是	记录创建者，用于操作审计
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者，用于操作审计
update_time	datetime	-	是	记录更新时间，信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 - 记录存在，2 - 记录删除辑删除）

SQL 代码：

```
CREATE TABLE `tb_facility_setting` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键（建议固定 1）',
  `audit_required` char(1) DEFAULT '1' COMMENT '是否需要审核（1 是 0 否）',
  `max_duration_minutes` int(11) DEFAULT 4320 COMMENT '最长使用时长（分钟）',
  `remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='功能房-全局设置表';
```

6)tb_facility_ban（申请人封禁表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键，自增生成，唯一标识每条封禁记录
user_id	bigint(20)	-	是	用户 ID，逻辑外键关联 sys_user.user_id，标封禁的申请人

reason	varchar(500)	-	是	封禁原因
status	char(1)	'0'	是	状态：0 - 生效，1 - 失效
expire_time	datetime	NULL	否	到期时间（NULL 表示永久 / 人工解封）
create_by	varchar(64)	-	是	记录创建者，用于操作审计
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者，用于操作审计
update_time	datetime	-	是	记录更新时间，信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 - 记录存在，2 - 记录删除（逻辑

SQL 代码：

```
CREATE TABLE `tb_facility_ban` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
  `user_id` bigint(20) NOT NULL COMMENT '用户 ID（逻辑外键：sys_user.user_id）',
  `reason` varchar(500) NOT NULL COMMENT '封禁原因',
  `status` char(1) DEFAULT '0' COMMENT '状态（0 生效 1 失效）',
  `expire_time` datetime DEFAULT NULL COMMENT '到期时间（NULL=永久）',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_ban_user_status` (`user_id`,`status`) USING BTREE,
  KEY `idx_ban_user` (`user_id`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='功能房-申请人封禁表';
```

三．数字图书馆模块

1) tb_library_book（数字图书馆 - 图书主体）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键 ID，自增生成
isbn13	char(13)	-	是	ISBN-13 数字字符串，全局唯一业务键
title	varchar	-	是	图书标题

author	varchar	-	是	作者
publisher	varchar	-	否	出版社
publish_year	int	-	否	出版年份
language	varchar	-	否	语言
keywords	varchar	-	否	关键词，逗号分隔
summary	text	-	否	图书摘要
cover_url	varchar	-	否	封面图片 URL
status	char(1)	'0'	是	审核状态：0 - 待审、1 - 已通过、2 - 驳 3 - 已下架
audit_by	varchar	-	否	审核人
audit_time	datetime	-	否	审核时间
audit_reason	varchar	-	否	审核原因
publish_time	datetime	-	否	首次发布时间（从待审 / 驳回转为已通过 入）
download_count	bigint	0	是	下载次数统计
last_download_time	datetime	-	否	最后下载时间
uploader_id	bigint	-	否	上传者 ID（逻辑外键关联 sys_user）
uploader_name	varchar(64)	-	否	上传者用户名（冗余）
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	'0'	是	删除标志：0 - 存在、2 - 删除（逻辑删

SQL 代码：

```
CREATE TABLE `tb_library_book` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `isbn13` char(13) NOT NULL COMMENT 'ISBN-13 数字字符串（唯一业务键）',
  `title` varchar(255) NOT NULL COMMENT '图书标题',
  `author` varchar(255) NOT NULL COMMENT '作者',
  `publisher` varchar(255) DEFAULT NULL COMMENT '出版社',
  `publish_year` int DEFAULT NULL COMMENT '出版年份',
  `language` varchar(32) DEFAULT NULL COMMENT '语言',
  `keywords` varchar(512) DEFAULT NULL COMMENT '关键词（逗号分隔）',
  `summary` text COMMENT '图书摘要',
  `cover_url` varchar(512) DEFAULT NULL COMMENT '封面图片 URL',
  `status` char(1) DEFAULT '0' COMMENT '审核状态（0 待审 1 已通过 2 驳回 3 已下架）',
  `audit_by` varchar(64) DEFAULT NULL COMMENT '审核人',
  `audit_time` datetime DEFAULT NULL COMMENT '审核时间',
  `audit_reason` varchar(512) DEFAULT NULL COMMENT '审核原因',
  `publish_time` datetime DEFAULT NULL COMMENT '首次发布时间',
  `download_count` bigint DEFAULT 0 COMMENT '下载次数统计',
  `last_download_time` datetime DEFAULT NULL COMMENT '最后下载时间',
  `uploader_id` bigint DEFAULT NULL COMMENT '上传者 ID',
  `uploader_name` varchar(64) DEFAULT NULL COMMENT '上传者用户名',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_isbn13` (`isbn13`) USING BTREE,
  KEY `idx_status_publish` (`status`,`publish_time`) USING BTREE,
  KEY `idx_title` (`title`) USING BTREE,
  KEY `idx_author` (`author`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='数字图书馆-图书主体';
```

2)tb_library_book_asset（数字图书馆 - 图书资产）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键 ID，自增生成
book_id	bigint	-	是	图书 ID，关联 tb_library_book.id（外键，删除）
asset_type	char(1)	-	是	资产类型：0 - 文件、1 - 外链

format	varchar(16)	-	否	文件格式（如 pdf/epub/mobi 等，仅 asset_type=0 时有效）
file_url	varchar(512)	-	否	文件访问 URL（仅 asset_type=0 时有效）
file_size	bigint	-	否	文件大小（单位：字节，仅 asset_type=0 时
file_hash	varchar	-	否	文件哈希值（建议 SHA-256，仅 asset_type= 有效）
oss_object_key	varchar	-	否	OSS 对象存储键（用于定位文件，仅 asset_ty 时有效）
link_url	varchar(512)	-	否	外链 URL（仅 asset_type=1 时有效）
sort	int	-	否	排序号，升序表示下载 / 展示优先级
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	'0'	是	删除标志：0 - 存在、2 - 删除（逻辑删除）

SQL 代码：

```
CREATE TABLE `tb_library_book_asset` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `book_id` bigint NOT NULL COMMENT '图书 ID（关联 tb_library_book.id）',
  `asset_type` char(1) NOT NULL COMMENT '资产类型（0 文件 1 外链）',
  `format` varchar(16) DEFAULT NULL COMMENT '文件格式（仅文件类型有效）',
  `file_url` varchar(512) DEFAULT NULL COMMENT '文件访问 URL（仅文件类型有效）',
  `file_size` bigint DEFAULT NULL COMMENT '文件大小（字节，仅文件类型有效）',
  `file_hash` varchar(64) DEFAULT NULL COMMENT '文件哈希值（SHA-256，仅文件类型有效）',
  `oss_object_key` varchar(256) DEFAULT NULL COMMENT 'OSS 对象存储键（仅文件类型有效）',
  `link_url` varchar(512) DEFAULT NULL COMMENT '外链 URL（仅外链类型有效）',
  `sort` int DEFAULT NULL COMMENT '排序号（升序优先）',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
```

PRIMARY KEY (`id`) USING BTREE,
KEY `idx_book_id` (`book_id`) USING BTREE,
UNIQUE KEY `uk_library_book_file` (`book_id`,`asset_type`,`format`,`file_hash`) USING BTREE COMMENT '文件资产幂等去重',
UNIQUE KEY `uk_library_book_link` (`book_id`,`asset_type`,`link_url`) USING BTREE COMMENT '外链资产幂等去重',
CONSTRAINT `fk_book_asset_book_id` FOREIGN KEY (`book_id`) REFERENCES `tb_library_book` (`id`) ON DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='数字图书馆-图书资产';

3)tb_library_book_favorite（数字图书馆 - 收藏关系）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键 ID，自增生成
book_id	bigint	-	是	图书 ID，关联 tb_library_book.id（外键，级除）
user_id	bigint	-	是	用户 ID（逻辑外键关联 sys_user.user_id）
create_time	datetime	-	是	收藏时间
create_by	varchar(64)	-	是	记录创建者（收藏用户）
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	’0’	是	删除标志：0 - 存在、2 - 删除（逻辑删除）

SQL 代码：

CREATE TABLE `tb_library_book_favorite` (
`id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
`book_id` bigint NOT NULL COMMENT '图书 ID（关联 tb_library_book.id）',
`user_id` bigint NOT NULL COMMENT '用户 ID（关联 sys_user.user_id）',
`create_time` datetime NOT NULL COMMENT '收藏时间',
`create_by` varchar(64) NOT NULL COMMENT '创建者',
`update_by` varchar(64) NOT NULL COMMENT '更新者',
`update_time` datetime NOT NULL COMMENT '更新时间',
`del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
PRIMARY KEY (`id`) USING BTREE,
KEY `idx_user` (`user_id`) USING BTREE,
UNIQUE KEY `uk_library_book_user` (`book_id`,`user_id`) USING BTREE COMMENT '单用户对同书唯一收藏',
CONSTRAINT `fk_favorite_book_id` FOREIGN KEY (`book_id`) REFERENCES `tb_library_book` (`id`) ON

DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='数字图书馆-收藏关系';

4)tb_library_book_download_log（数字图书馆 - 下载日志）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键 ID，自增生成
book_id	bigint	-	是	图书 ID，关联 tb_library_book.id
asset_id	bigint	-	否	资产 ID（关联 tb_library_book_asset.id，外链下为空）
user_id	bigint	-	是	用户 ID（逻辑外键关联 sys_user.user_id）
result	char(1)	'0'	是	下载结果：0 - 成功、1 - 失败
ip	varchar	-	否	下载时的 IP 地址
ua	varchar	-	否	下载时的用户代理（浏览器 / 设备信息）
referer	varchar	-	否	下载来源页面 URL
create_time	datetime	-	是	下载时间

SQL 代码：

```
CREATE TABLE `tb_library_book_download_log` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `book_id` bigint NOT NULL COMMENT '图书 ID',
  `asset_id` bigint DEFAULT NULL COMMENT '资产 ID（可空）',
  `user_id` bigint NOT NULL COMMENT '用户 ID',
  `result` char(1) DEFAULT '0' COMMENT '下载结果（0 成功 1 失败）',
  `ip` varchar(64) DEFAULT NULL COMMENT 'IP 地址',
  `ua` varchar(512) DEFAULT NULL COMMENT '用户代理',
  `referer` varchar(512) DEFAULT NULL COMMENT '来源页面',
  `create_time` datetime NOT NULL COMMENT '下载时间',
  PRIMARY KEY (`id`) USING BTREE,
  KEY `idx_book_time` (`book_id`,`create_time`) USING BTREE,
  KEY `idx_user_time` (`user_id`,`create_time`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='数字图书馆-下载日志';
```

5)tb_library_book_log（数字图书馆 - 操作日志）

字段名	数据类型	默认值	非空约束	字段说明

id	bigint	-	是	主键 ID，自增生成
book_id	bigint	-	是	图书 ID，关联 tb_library_book.id
action	varchar(32)	-	是	操作类型（如 CREATE/EDIT/APPROVE/REJECT 等
actor_id	bigint	-	否	操作人 ID（关联 sys_user.user_id）
actor_name	varchar	-	否	操作人姓名（冗余）
ip	varchar	-	否	操作时的 IP 地址
ua	varchar	-	否	操作时的用户代理（浏览器 / 设备信息）
detail	varchar	-	否	操作详情（如审核原因、修改内容等）
result	varchar	-	否	操作结果（如 SUCCESS/FAIL）
create_time	datetime	-	是	操作时间
del_flag	char(1)	'0'	是	删除标志：0 - 存在、2 - 删除（逻辑删除）

SQL 代码：

```
CREATE TABLE `tb_library_book_log` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `book_id` bigint NOT NULL COMMENT '图书 ID',
  `action` varchar(32) NOT NULL COMMENT '操作类型',
  `actor_id` bigint DEFAULT NULL COMMENT '操作人 ID',
  `actor_name` varchar(64) DEFAULT NULL COMMENT '操作人姓名',
  `ip` varchar(64) DEFAULT NULL COMMENT 'IP 地址',
  `ua` varchar(512) DEFAULT NULL COMMENT '用户代理',
  `detail` varchar(1024) DEFAULT NULL COMMENT '操作详情',
  `result` varchar(16) DEFAULT NULL COMMENT '操作结果',
  `create_time` datetime NOT NULL COMMENT '操作时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  KEY `idx_book_time` (`book_id`,`create_time`) USING BTREE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='数字图书馆-操作日志';
```

6)tb_library_librarian（数字图书馆 - 图书管理员映射）

字段名	数据类型	默认值	非空约束	字段说明

id	bigint	-	是	主键 ID，自增生成
user_id	bigint	-	是	系统用户 ID（逻辑外键关联 sys_user.user_id，一）
username	varchar	-	否	用户名（冗余快照）
nickname	varchar	-	否	昵称（冗余快照）
remark	varchar	-	否	备注
create_by	varchar(64)	-	是	记录创建者
create_time	datetime	-	是	记录创建时间
update_by	varchar(64)	-	是	记录更新者
update_time	datetime	-	是	记录更新时间
del_flag	char(1)	'0'	是	删除标志：0 - 存在、2 - 删除（逻辑删除）

SQL 代码：

```
CREATE TABLE `tb_library_librarian` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `user_id` bigint NOT NULL COMMENT '系统用户 ID（关联 sys_user.user_id）',
  `username` varchar(64) DEFAULT NULL COMMENT '用户名',
  `nickname` varchar(64) DEFAULT NULL COMMENT '昵称',
  `remark` varchar(512) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_user` (`user_id`) USING BTREE COMMENT '单用户仅一条有效映射'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='数字图书馆-图书管理员映射';
```

四．失物招领模块

（1）tb_lost_item 表字段设计详情

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键 ID，自增生成，唯一标识每条失物招领

type	varchar(10)	-	是	类型：'lost' 表示丢失，'found' 表示捡到
title	varchar(100)	-	是	标题（2~50 字）
content	text	-	是	正文描述（5~2000 字）
location	varchar(100)	NULL	否	地点（可选）
lost_time	datetime	NULL	否	发生时间（可选）
views	int	0	是	浏览量
status	char(1)	'1'	是	状态：0 = 草稿，1 = 待审核，2 = 已发布，3 回，4 = 已下架
solved_flag	char(1)	'0'	是	是否已解决：0 = 否，1 = 是（不可回退）
publish_time	datetime	NULL	否	发布时间（通过审核时写入）
reject_reason	varchar(500)	NULL	否	驳回原因（驳回动作时必填）
offline_reason	varchar(500)	NULL	否	下架原因（下架动作时必填）
contact_info	varchar(50)	NULL	否	联系信息（自由文本，如电话 / 微信等，≤50
create_by	varchar(64)	''	是	创建者（用于操作审计）
create_time	datetime	NULL	否	创建时间
update_by	varchar(64)	''	是	更新者（用于操作审计）
update_time	datetime	NULL	否	更新时间
del_flag	char(1)	'0'	是	删除标志：0 = 存在，2 = 删除（逻辑删除）

SQL 代码

```
CREATE TABLE `tb_lost_item` (
  `id`          BIGINT        NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `type`        VARCHAR(10)   NOT NULL COMMENT '类型：lost=丢失 found=捡到',
  `title`       VARCHAR(100)  NOT NULL COMMENT '标题（2~50 字）',
  `content`     TEXT          NOT NULL COMMENT '正文描述（5~2000 字）',
  `location`    VARCHAR(100)            DEFAULT NULL COMMENT '地点（可选）',
  `lost_time`   DATETIME                DEFAULT NULL COMMENT '发生时间（可选）',
  `views`       INT           NOT NULL DEFAULT 0 COMMENT '浏览量',
```

```
  `status`          CHAR(1)          NOT NULL DEFAULT '1' COMMENT '状态：0 草稿 1 待审 2 已发 3 驳回 4 下架',
  `solved_flag`     CHAR(1)          NOT NULL DEFAULT '0' COMMENT '是否已解决：0 否 1 是（不可回退）',
  `publish_time`    DATETIME                  DEFAULT NULL COMMENT '发布时间（通过审核时写入）',
  `reject_reason`   VARCHAR(500)              DEFAULT NULL COMMENT '驳回原因（必填于驳回动作）',
  `offline_reason` VARCHAR(500)               DEFAULT NULL COMMENT '下架原因（必填于下架动作）',
  `contact_info`    VARCHAR(50)               DEFAULT NULL COMMENT '联系信息（自由文本，例如电话/微信等，≤50 字）',
  `create_by`       VARCHAR(64)      NOT NULL DEFAULT '' COMMENT '创建者',
  `create_time`     DATETIME                  DEFAULT NULL COMMENT '创建时间',
  `update_by`       VARCHAR(64)      NOT NULL DEFAULT '' COMMENT '更新者',
  `update_time`     DATETIME                  DEFAULT NULL COMMENT '更新时间',
  `del_flag`        CHAR(1)          NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  KEY `idx_status_pub` (`status`, `publish_time`),
  KEY `idx_type_solved` (`type`, `solved_flag`),
  KEY `idx_title` (`title`),
  KEY `idx_creator` (`create_by`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='失物招领-主体';
```

（2）tb_lost_item_image 表字段设计详情

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键 ID，自增生成，唯一标识每张图片
item_id	bigint	-	是	条目 ID，关联 tb_lost_item.id，标识图片所属的招领信息，建立 “一对多” 关联（一个条目最多图片）
url	varchar(500)	-	是	图片 URL（存储于 OSS）
sort_no	int	0	是	排序号（0 开始），按升序排列图片顺序
create_by	varchar(64)	''	是	创建者（用于操作审计）
create_time	datetime	NULL	否	创建时间
update_by	varchar(64)	''	是	更新者（用于操作审计）
update_time	datetime	NULL	否	更新时间
del_flag	char(1)	'0'	是	删除标志：0 = 存在，2 = 删除（逻辑删除）

SQL 代码

```
CREATE TABLE `tb_lost_item_image` (
```

```
`id`          BIGINT          NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
`item_id`     BIGINT          NOT NULL COMMENT '条目 ID（tb_lost_item.id）',
`url`         VARCHAR(500) NOT NULL COMMENT '图片 URL（OSS）',
`sort_no`     INT             NOT NULL DEFAULT 0 COMMENT '排序号（0 开始）',
`create_by`   VARCHAR(64)   NOT NULL DEFAULT '' COMMENT '创建者',
`create_time`DATETIME                 DEFAULT NULL COMMENT '创建时间',
`update_by`   VARCHAR(64)   NOT NULL DEFAULT '' COMMENT '更新者',
`update_time`DATETIME                 DEFAULT NULL COMMENT '更新时间',
`del_flag`    CHAR(1)         NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
PRIMARY KEY (`id`) USING BTREE,
KEY `idx_item_sort` (`item_id`, `sort_no`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='失物招领-图片';
```

五．问卷（表单收集）模块

1) tb_survey（问卷 - 主表）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID，自增生成，唯一标识每个问卷
title	varchar(200)	-	是	问卷标题 / 表单名称，展示给用户的问卷名称
status	char(1)	'1'	是	问卷状态：0 = 草稿、1 = 发布、2 = 归档
visible_all	char(1)	'1'	是	可见范围标识：1 = 全员可见、0 = 按范围（角色门 / 岗位并集）
deadline	datetime	NULL	否	截止时间，超过后不可提交 / 修改答卷
remark	varchar(500)	NULL	否	问卷备注信息，用于补充说明问卷相关内容
pinned	char(1)	'0'	是	置顶标识：0 = 否、1 = 是
pinned_time	datetime	NULL	否	置顶时间，用于置顶排序辅助
create_by	varchar(64)	-	是	记录创建者（用户名），用于操作审计
create_time	datetime	-	是	记录创建时间，自动填充问卷创建时间
update_by	varchar(64)	-	是	记录更新者（用户名），用于操作审计
update_time	datetime	-	是	记录更新时间，问卷信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 = 记录存在、2 = 记录删除（逻辑

SQL 代码：

```
CREATE TABLE `tb_survey` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '问卷主键',
  `title` varchar(200) NOT NULL COMMENT '问卷标题/表单名称',
  `status` char(1) DEFAULT '1' COMMENT '0 草稿 1 发布 2 归档',
  `visible_all` char(1) DEFAULT '1' COMMENT '1 全员可见 0 按范围（取并集）',
  `deadline` datetime DEFAULT NULL COMMENT '截止时间（超过后不可提交/修改）',
  `remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `pinned` char(1) DEFAULT '0' COMMENT '置顶标识（0 否 1 是）',
  `pinned_time` datetime DEFAULT NULL COMMENT '置顶时间',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  KEY `idx_status_deadline` (`status`,`deadline`) COMMENT '发布中/即将到期列表与筛选',
  KEY `idx_create_by` (`create_by`) COMMENT '按创建人检索') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='问卷-主表';
```

2)tb_survey_scope（问卷 - 可见范围）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID，自增生成，唯一标识每个可见范围配置
survey_id	bigint(20)	-	是	问卷 ID，关联问卷主表（tb_survey.id），标识该范围所卷
scope_type	tinyint(4)	-	是	范围类型：0 = 角色、1 = 部门、2 = 岗位
ref_id	bigint(20)	-	是	引用 ID，对应范围类型的具体 ID（sys_role.role_id/sys_dept.dept_id/sys_post.post_
create_by	varchar(64)	-	是	记录创建者（用户名），用于操作审计
create_time	datetime	-	是	记录创建时间，自动填充范围配置创建时间
update_by	varchar(64)	-	是	记录更新者（用户名），用于操作审计
update_time	datetime	-	是	记录更新时间，范围配置变更时自动更新

del_flag	char(1)	'0'	是	删除标志：0 = 记录存在、2 = 记录删除（逻辑删除）

SQL 代码：

```
CREATE TABLE `tb_survey_scope` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
  `survey_id` bigint(20) NOT NULL COMMENT '问卷 ID（逻辑外键：tb_survey.id）',
  `scope_type` tinyint(4) NOT NULL COMMENT '0 角色 1 部门 2 岗位',
  `ref_id` bigint(20) NOT NULL COMMENT '引用 ID（角色/部门/岗位 ID）',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_scope` (`survey_id`,`scope_type`,`ref_id`) COMMENT '防重复范围项',
  KEY `idx_survey` (`survey_id`) COMMENT '按问卷加载范围') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='问卷-可见范围';
```

3)tb_survey_item（问卷 - 题目）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID，自增生成，唯一标识每个问卷题目
survey_id	bigint(20)	-	是	问卷 ID，关联问卷主表（tb_survey.id），标识所属问卷
title	varchar(300)	-	是	题目 / 提示文本，展示给用户的问题内容
type	tinyint(4)	-	是	题型：1 = 文本、2 = 单选、3 = 多选（预留 4 件、5 = 时间）
required	char(1)	'0'	是	是否必填：0 = 否、1 = 是
sort_no	int(11)	'0'	是	排序号（0 开始），按升序排列题目顺序
ext_json	json	NULL	否	扩展配置，不同题型的特殊配置（如文本框占位、最大长度等）
create_by	varchar(64)	-	是	记录创建者（用户名），用于操作审计
create_time	datetime	-	是	记录创建时间，自动填充题目创建时间

update_by	varchar(64)	-	是	记录更新者（用户名），用于操作审计
update_time	datetime	-	是	记录更新时间，题目信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 = 记录存在、2 = 记录删除（逻辑

SQL 代码：

```
CREATE TABLE `tb_survey_item` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
  `survey_id` bigint(20) NOT NULL COMMENT '问卷 ID',
  `title` varchar(300) NOT NULL COMMENT '题目/提示文本',
  `type` tinyint(4) NOT NULL COMMENT '1 文本 2 单选 3 多选（预留 4 文件/5 时间）',
  `required` char(1) DEFAULT '0' COMMENT '是否必填：0 否 1 是',
  `sort_no` int(11) DEFAULT '0' COMMENT '排序号（0 开始）',
  `ext_json` json DEFAULT NULL COMMENT '扩展配置',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  KEY `idx_survey_sort` (`survey_id`,`sort_no`) COMMENT '按序加载题目') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='问卷-题目';
```

4)tb_survey_option（问卷 - 选项，仅用于单 / 多选）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID，自增生成，唯一标识每个问卷选项
item_id	bigint(20)	-	是	题目 ID，关联问卷题目表（tb_survey_item.id）识选项所属题目，建立 “一对多” 关联
label	varchar(200)	-	是	选项展示文本（如 “男”“女”），面向用户展示容
value	varchar(200)	NULL	否	选项值（可选），默认与 label 一致，支持自定义（如数字、编码等）
sort_no	int(11)	'0'	是	排序号（0 开始），按升序排列选项顺序
create_by	varchar(64)	''	是	记录创建者（用户名或系统标识），用于操作审计

create_time	datetime	NULL	否	记录创建时间，默认自动填充选项创建时间
update_by	varchar(64)	''	是	记录更新者（创建者、管理员或系统标识），用于审计
update_time	datetime	NULL	否	记录更新时间，选项信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 = 记录存在、2 = 记录删除（逻辑删

SQL 代码：

```
CREATE TABLE `tb_survey_option` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键 ID',
  `item_id` bigint(20) NOT NULL COMMENT '题目 ID（tb_survey_item.id）',
  `label` varchar(200) NOT NULL COMMENT '选项展示文本',
  `value` varchar(200) DEFAULT NULL COMMENT '选项值（可空，默认=label）',
  `sort_no` int(11) NOT NULL DEFAULT '0' COMMENT '排序号',
  `create_by` varchar(64) NOT NULL DEFAULT '' COMMENT '创建者',
  `create_time` datetime DEFAULT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL DEFAULT '' COMMENT '更新者',
  `update_time` datetime DEFAULT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  KEY `idx_item_sort` (`item_id`,`sort_no`) COMMENT '按序加载选项') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='问卷-选项';
```

5)tb_survey_answer（问卷 - 答卷，每人一份）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID，自增生成，唯一标识每个答卷
survey_id	bigint(20)	-	是	问卷 ID，关联问卷主表（tb_survey.id），标识卷所属问卷
user_id	bigint(20)	-	是	提交人用户 ID（逻辑外键：sys_user.user_id）识答卷提交者
submit_time	datetime	NULL	否	首次提交时间（或最近提交时间）
update_time2	datetime	NULL	否	最近修改时间（区别于审计字段的 update_time
create_by	varchar(64)	-	是	记录创建者（用户名），用于操作审计

create_time	datetime	-	是	记录创建时间，自动填充答卷创建时间
update_by	varchar(64)	-	是	记录更新者（用户名），用于操作审计
update_time	datetime	-	是	记录更新时间，答卷信息变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 = 记录存在、2 = 记录删除（逻辑

SQL 代码：

```
CREATE TABLE `tb_survey_answer` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
  `survey_id` bigint(20) NOT NULL COMMENT '问卷 ID',
  `user_id` bigint(20) NOT NULL COMMENT '提交人用户 ID（逻辑外键：sys_user.user_id）',
  `submit_time` datetime DEFAULT NULL COMMENT '首次提交时间（或最近提交时间）',
  `update_time2` datetime DEFAULT NULL COMMENT '最近修改时间',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_survey_user` (`survey_id`,`user_id`) COMMENT '保证“一人一份”',
  KEY `idx_survey` (`survey_id`) COMMENT '统计/导出') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='问卷-答卷，每人一份';
```

6)tb_survey_answer_item（问卷 - 答案明细）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint(20)	-	是	主键 ID，自增生成，唯一标识每个答案明细
answer_id	bigint(20)	-	是	答卷 ID，关联答卷表（tb_survey_answer.id 标识该明细所属答卷
item_id	bigint(20)	-	是	题目 ID，关联问卷题目表（tb_survey_item. 标识该明细对应的题目
value_text	text	NULL	否	文本题的答案内容
value_option_ids	json	NULL	否	单 / 多选题的答案（JSON 数组，元素为 tb_survey_option.id）
create_by	varchar(64)	-	是	记录创建者（用户名），用于操作审计

create_time	datetime	-	是	记录创建时间，自动填充答案明细创建时间
update_by	varchar(64)	-	是	记录更新者（用户名），用于操作审计
update_time	datetime	-	是	记录更新时间，答案明细变更时自动更新
del_flag	char(1)	'0'	是	删除标志：0 = 记录存在、2 = 记录删除（逻除）

SQL 代码：

```
CREATE TABLE `tb_survey_answer_item` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
  `answer_id` bigint(20) NOT NULL COMMENT '答卷 ID（逻辑外键：tb_survey_answer.id）',
  `item_id` bigint(20) NOT NULL COMMENT '题目 ID（逻辑外键：tb_survey_item.id）',
  `value_text` text DEFAULT NULL COMMENT '文本题的答案',
  `value_option_ids` json DEFAULT NULL COMMENT '单/多选题的答案（JSON 数组，元素为 tb_survey_option.id）
',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0 存在 2 删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_answer_item` (`answer_id`,`item_id`) COMMENT '同一答卷同一题仅一条',
  KEY `idx_answer` (`answer_id`) COMMENT '按答卷加载明细') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='问卷-答案明细';
```

六．通知公告模块

1) tb_notice（通知公告 - 主体）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	公告主键
title	varchar(200)	-	是	标题
content_html	mediumtext	-	是	富文本 HTML 内容（存储已净化的 HTML）
type	tinyint	2	是	类型：1 - 通知 2 - 公告（参照字典 `sys_notice_type`）
status	tinyint	0	是	状态：0 - 草稿 1 - 已发布 2 - 撤回 3 - 已（参照字典 `sys_notice_status`）

visible_all	tinyint	1	是	是否全员可见：1 - 是 0 - 否；为 0 时生效 tb_notice_scope 限定并集
publisher_id	bigint	-	是	发布者用户 ID（逻辑外键：sys_user.user_
publish_time	datetime	NULL	否	发布时间（status=1 时应有值）
expire_time	datetime	NULL	否	到期时间（超过即视为已过期，应用层定时 / 计算）
pinned	tinyint	0	是	是否置顶：0 - 否 1 - 是
pinned_time	datetime	NULL	否	置顶时间（置顶排序辅助）
edit_count	int	0	是	编辑次数（每次保存 + 1，用于审计 / 对比）
read_count	int	0	是	阅读次数（冗余计数，接口侧聚合回写）
attachment_count	int	0	是	附件数量（冗余计数，上传 / 删除附件后更新
remark	varchar(500)	NULL	否	备注
create_by	varchar(64)	-	是	创建者（用户名或系统标识）
create_time	datetime	-	是	创建时间
update_by	varchar(64)	-	是	更新者（用户名或系统标识）
update_time	datetime	-	是	更新时间
del_flag	char(1)	'0'	是	删除标志：'0'= 存在、'2'= 删除（逻辑删除

SQL 代码：

```
CREATE TABLE `tb_notice` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '公告主键',
  `title` varchar(200) NOT NULL COMMENT '标题',
  `content_html` mediumtext NOT NULL COMMENT '富文本 HTML 内容（已净化）',
  `type` tinyint NOT NULL DEFAULT 2 COMMENT '类型（1-通知 2-公告，参照 sys_notice_type）',
  `status` tinyint NOT NULL DEFAULT 0 COMMENT '状态（0-草稿 1-已发布 2-撤回 3-已过期，参照 sys_notice_status）',
  `visible_all` tinyint NOT NULL DEFAULT 1 COMMENT '是否全员可见（1-是 0-否）',
  `publisher_id` bigint NOT NULL COMMENT '发布者用户 ID（逻辑外键 sys_user.user_id）',
  `publish_time` datetime DEFAULT NULL COMMENT '发布时间（status=1 时有效）',
  `expire_time` datetime DEFAULT NULL COMMENT '到期时间',
```

```
`pinned` tinyint NOT NULL DEFAULT 0 COMMENT '是否置顶（0-否 1-是）',
`pinned_time` datetime DEFAULT NULL COMMENT '置顶时间',
`edit_count` int NOT NULL DEFAULT 0 COMMENT '编辑次数',
`read_count` int NOT NULL DEFAULT 0 COMMENT '阅读次数（冗余）',
`attachment_count` int NOT NULL DEFAULT 0 COMMENT '附件数量（冗余）',
`remark` varchar(500) DEFAULT NULL COMMENT '备注',
`create_by` varchar(64) NOT NULL COMMENT '创建者',
`create_time` datetime NOT NULL COMMENT '创建时间',
`update_by` varchar(64) NOT NULL COMMENT '更新者',
`update_time` datetime NOT NULL COMMENT '更新时间',
`del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
PRIMARY KEY (`id`) USING BTREE,
KEY `idx_notice_status_pub` (`status`,`pinned`,`pinned_time`,`publish_time`) COMMENT '列表与置顶排序索引',
KEY `idx_notice_expire` (`expire_time`) COMMENT '到期扫描索引',
KEY `idx_notice_publisher` (`publisher_id`) COMMENT '我的发布列表索引',
KEY `idx_notice_title` (`title`) COMMENT '标题搜索索引') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COMMENT='通知公告-主体';
```

2)tb_notice_scope（通知公告 - 可见范围）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键
notice_id	bigint	-	是	公告 ID（逻辑外键：tb_notice.id）
scope_type	tinyint	-	是	范围类型：0 - 角色 1 - 部门 2 - 岗位
ref_id	bigint	-	是	引用 ID：sys_role.role_id / sys_dept.dep / sys_post.post_id
remark	varchar(255)	NULL	否	备注
create_by	varchar(64)	-	是	创建者（用户名或系统标识）
create_time	datetime	-	是	创建时间
update_by	varchar(64)	-	是	更新者（用户名或系统标识）
update_time	datetime	-	是	更新时间
del_flag	char(1)	'0'	是	删除标志：'0'= 存在、'2'= 删除（逻辑删除）

SQL 代码：

```
CREATE TABLE `tb_notice_scope` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键',
  `notice_id` bigint NOT NULL COMMENT '公告 ID（逻辑外键 tb_notice.id）',
  `scope_type` tinyint NOT NULL COMMENT '范围类型（0-角色 1-部门 2-岗位）',
  `ref_id` bigint NOT NULL COMMENT '引用 ID（角色/部门/岗位 ID）',
  `remark` varchar(255) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_scope_unique` (`notice_id`,`scope_type`,`ref_id`,`del_flag`) COMMENT '避免重复范围项',
  KEY `idx_scope_notice` (`notice_id`) COMMENT '公告关联索引',
  KEY `idx_scope_type_ref` (`scope_type`,`ref_id`) COMMENT '范围查询索引') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='通知公告-可见范围';
```

3)tb_notice_attachment（通知公告 - 附件）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键
notice_id	bigint	-	是	公告 ID（逻辑外键：tb_notice.id）
file_name	varchar(255)	-	是	文件名（展示用）
file_url	varchar(1024)	-	是	文件 URL（OSS 直链 / 授权链接）
file_type	varchar(50)	NULL	否	文件类型（扩展名或 MIME）
file_size	bigint	NULL	否	文件大小（字节）
sort	int	0	是	序号（升序显示）
remark	varchar(255)	NULL	否	备注
create_by	varchar(64)	-	是	创建者（用户名或系统标识）
create_time	datetime	-	是	创建时间
update_by	varchar(64)	-	是	更新者（用户名或系统标识）
update_time	datetime	-	是	更新时间

del_flag	char(1)	'0'	是	删除标志：'0'= 存在、'2'= 删除（逻辑删

SQL 代码：

```
CREATE TABLE `tb_notice_attachment` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键',
  `notice_id` bigint NOT NULL COMMENT '公告 ID（逻辑外键 tb_notice.id）',
  `file_name` varchar(255) NOT NULL COMMENT '文件名',
  `file_url` varchar(1024) NOT NULL COMMENT '文件 URL（OSS 链接）',
  `file_type` varchar(50) DEFAULT NULL COMMENT '文件类型（扩展名或 MIME）',
  `file_size` bigint DEFAULT NULL COMMENT '文件大小（字节）',
  `sort` int NOT NULL DEFAULT 0 COMMENT '序号（升序显示）',
  `remark` varchar(255) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  KEY `idx_attach_notice` (`notice_id`) COMMENT '公告附件关联索引') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='通知公告-附件';
```

4)tb_notice_read（通知公告 - 阅读回执）

字段名	数据类型	默认值	非空约束	字段说明
id	bigint	-	是	主键
notice_id	bigint	-	是	公告 ID（逻辑外键：tb_notice.id
user_id	bigint	-	是	用户 ID（逻辑外键：sys_user.user
ack	tinyint	0	是	是否确认（预留；后续可支持 “需确的通知）
read_time	datetime	CURRENT_TIMESTAMP	否	阅读时间
remark	varchar(255)	NULL	否	备注
create_by	varchar(64)	-	是	创建者（用户名或系统标识）
create_time	datetime	-	是	创建时间
update_by	varchar(64)	-	是	更新者（用户名或系统标识）

update_time	datetime	-	是	更新时间
del_flag	char(1)	'0'	是	删除标志：'0'= 存在、'2'= 删除（删除）

SQL 代码：

```
CREATE TABLE `tb_notice_read` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键',
  `notice_id` bigint NOT NULL COMMENT '公告 ID（逻辑外键 tb_notice.id）',
  `user_id` bigint NOT NULL COMMENT '用户 ID（逻辑外键 sys_user.user_id）',
  `ack` tinyint NOT NULL DEFAULT 0 COMMENT '是否确认（预留）',
  `read_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '阅读时间',
  `remark` varchar(255) DEFAULT NULL COMMENT '备注',
  `create_by` varchar(64) NOT NULL COMMENT '创建者',
  `create_time` datetime NOT NULL COMMENT '创建时间',
  `update_by` varchar(64) NOT NULL COMMENT '更新者',
  `update_time` datetime NOT NULL COMMENT '更新时间',
  `del_flag` char(1) NOT NULL DEFAULT '0' COMMENT '删除标志（0-存在 2-删除）',
  PRIMARY KEY (`id`) USING BTREE,
  UNIQUE KEY `uk_notice_read` (`notice_id`,`user_id`,`del_flag`) COMMENT '单用户单公告唯一回执',
  KEY `idx_read_user` (`user_id`) COMMENT '用户已读列表索引') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='通知公告-阅读回执';
```

3.2.3 数据库管理员视图

数据库管理员视图包含全量表结构、索引设计、权限配置等信息，核心管理视图如下：

表结构管理视图：显示所有表的字段定义、数据类型、约束条件、默认值等；

索引管理视图：显示所有表的索引名称、索引类型（唯一索引 / 普通索引）、索引字段、索引状态等；

权限管理视图：显示数据库用户、角色及对应的表操作权限（查询 / 插入 / 更新 / 删除）；

数据统计视图：显示各表的数据量、增长趋势、占用存储空间等信息；

操作日志视图：记录数据库的 DDL（数据定义语言）、DML（数据操作语言）操作，包含操作人、操作时间、操作内容、操作结果等。

3.3 物理结构设计

3.3.1 内存安排

索引区设计：

核心业务表的主键索引、唯一索引、高频查询索引优先加载到内存缓冲区，提升索引查询速度；

MySQL InnoDB 引擎默认使用缓冲池(Buffer Pool)缓存索引和数据,配置缓冲池大小为服务器物理内存的 50%-70%（如 16GB 内存服务器配置 8GB-12GB 缓冲池）。

缓冲区设计：

配置 MySQL 查询缓存（Query Cache），缓存高频执行的查询语句及结果（如固定条件的资源列表查询、公告列表查询）；

应用层通过 Redis 缓存热点数据，如课程资源排行榜、热门图书列表、用户会话信息等，缓存有效期根据数据更新频率设置（如排行榜缓存 5 分钟更新一次）。

3.3.2 外存设备及空间组织

外存设备：采用企业级 SSD 硬盘存储数据库文件，提升数据读写速度；OSS 存储系统文件资源（如文档、图片）与数据库文件分离存储。

空间组织：
MySQL 数据库文件按表空间划分，核心业务表（如 tb_course_resource、tb_facility_booking）单独设置表空间，便于管理和维护；
数据块划分：InnoDB 引擎默认数据块大小为 16KB，根据业务数据特征（如大文本字段较多），可调整部分表的数据块大小为 32KB；
索引区与数据区分离存储，避免索引操作与数据操作相互影响；
定期进行数据归档：对历史数据（如超过 1 年的预约记录、下载日志）进行归档存储，保留核心业务表的查询性能。
3.3.3 数据访问方式
读取方式：
单条数据查询：通过主键索引或唯一索引快速定位数据，直接读取数据记录；
批量数据查询：使用分页查询（LIMIT）避免一次性加载大量数据，通过联合索引优化查询条件，减少表扫描；
关联查询：优先使用内连接（INNER JOIN），避免外连接导致的性能损耗；关联字段确保建立索引，减少关联时的匹配开销；
热点数据读取：优先从 Redis 缓存读取，缓存未命中时查询数据库并更新缓存。
写入方式：
单条数据插入 / 更新：直接执行 SQL 语句，确保事务原子性（如资源上传时同步更新积分记录）；
批量数据插入：使用批量插入语句（INSERT INTO ... VALUES (...), (...), ...），减少 SQL 执行次数；
高并发写入：通过数据库连接池控制并发数，避免数据库连接耗尽；核心表（如操作日志表）采用分表策略（按时间分表），分散写入压力。
删除方式：
统一采用逻辑删除，仅更新 del_flag 字段，避免物理删除导致的索引碎片和数据丢失；
定期清理逻辑删除数据：对标记为删除状态（del_flag=2）且超过保留期限（如 6 个月）的数据进行物理删除，释放存储空间。
4 运用设计
4.1 数据字典设计
4.1.1 数据字典核心内容
数据字典用于记录数据库中所有数据项、表、索引、约束等对象的详细信息，核心包含以下模块：
表字典：记录每张表的表名、模块归属、表描述、创建时间、更新时间、负责人等信息；
字段字典：记录每个字段的字段名、所属表名、数据类型、长度、默认值、非空约束、字段描述、枚举值说明等信息；
索引字典：记录每个索引的索引名、所属表名、索引类型、索引字段、索引描述、创建时间等信息；
约束字典：记录每张表的主键约束、外键约束、唯一约束等，包含约束名称、约束类型、关联字段、约束描述等信息；
枚举字典：记录系统中所有状态字段的枚举值及说明（如审核状态、资源类型、问卷状态等）。
4.1.2 数据字典维护方式
数据字典与数据库结构同步更新：新增表 / 字段 / 索引时，同步更新数据字典；修改表结构时，同步更新数据字典中的对应记录；
数据字典存储在数据库中（创建专门的数据字典表），支持通过管理后台进行查询和维护；
提供数据字典导出功能，支持导出为 Excel、PDF 格式，便于文档归档和团队共享。
4.2 安全保密设计
4.2.1 访问者分类及权限控制
根据用户角色划分访问者类型，不同类型访问者拥有不同的数据库操作权限：

访问者类型	权限范围	数据访问限制
普通用户	仅拥有数据查询权限（部分模块支持数据插入，如资源上传、预约申请）	仅能查询和操作自己有权限访问的数据（如自己上传的资源、自己的预约记录）
专业负责人 / 图书管理员	拥有所属模块的数据查询、审核权限（更新数据状态）	仅能审核和管理本专业 / 图书馆模块的数据（如专业负责人仅能审核本专业的资源）
系统管理员	拥有全量数据的查询、插入、更新、删除权限	无数据访问限制，但操作需记录审计日志
数据库管理员（DBA）	拥有数据库管理权限（如索引优化、备份恢复）	仅执行管理操作，不参与业务数据的直接操作

4.2.2 访问类型控制

访问类型	控制措施
查询访问	通过角色权限过滤数据，仅返回用户有权限访问的数据；敏感字段（如联系信息）在非必要场景下脱展示
插入访问	验证数据合法性（如字段格式、长度、枚举值）；限制重复插入（通过唯一索引实现）；记录插入人插入时间
更新访问	验证操作人权限（仅允许更新自己创建的数据或拥有审核权限的数据）；记录更新前后的字段值，支数据追溯；限制非法状态流转（如已审核通过的资源不可回退为待审状态）
删除访问	仅允许逻辑删除，禁止物理删除；验证删除权限（仅管理员可删除核心数据，普通用户仅能删除自己建的数据）；记录删除人及删除时间

4.2.3 数据对象安全控制

敏感数据加密：对用户联系信息、密码等敏感数据进行加密存储（如密码使用 BCrypt 加密，联系信息使用 AES 加密）；

数据访问审计：记录所有数据操作的审计日志（操作人、操作时间、操作类型、操作内容、IP 地址），便于安全审计和问题追溯；

数据库访问控制：

限制数据库访问 IP：仅允许应用服务器和管理终端的 IP 访问数据库；

数据库账号权限最小化：应用程序使用的数据库账号仅分配必要的操作权限（如查询、插入、更新），不分配 DROP、ALTER 等高危权限；

定期更换数据库账号密码，避免密码泄露；

数据备份与恢复：

每日自动备份数据库，保留最近 30 天的备份文件；
定期进行备份恢复测试，确保备份文件可用；
核心业务数据（如资源信息、预约记录）采用异地备份，防止本地存储设备故障导致数据丢失。