

软件工程 · 第二次团队作业 | 概要设计阶段

智班工坊 · 班小助

数据库设计说明书（Database Design Specification）V1.0

文档编号	ZBGF-DDS-2026-003
项目名称	班小助 BanXiaoZhu —— 校园通知与事务智能体
团队名称	智班工坊
团队成员	3124004185 肖武阳（队长）；3124004175 刘旭波；3124004181 王梓瀚
数据库系统	SQLite 3（单文件部署）
编写日期	2026 年 11 月 20 日
文档版本	V1.0
密级	公开（课程作业）

1 引言

1.1 编写目的

本文档描述班小助系统的数据库设计：从概念模型（ER 图）到逻辑模型（关系模式与表结构），再到物理设计（索引、约束、存储与备份），并给出对象关系映射（ORM）方案与初始化数据。目标是让任何一位接手者只读本文档，就能建库、写数据访问代码、评估扩展方式。

1.2 范围

本期数据库包含 5 张业务表：users（用户）、notices（通知）、tasks（待办）、reminders（提醒）、kb（班级资料）。涉及的数据量级为「一个班级、百级用户、千级记录」，读写比例约为 8:2，无强并发事务要求，因此选择嵌入式 SQLite。

1.3 术语与缩写

表 1-1 术语说明

术语	说明
ER 图	实体—联系图，用实体、属性、联系描述概念模型。
关系模式	表的抽象表示，形如 表名(属性1, 属性2, …)，下划线为主键。
主键 PK	唯一标识一行记录的字段。
外键 FK	引用另一张表主键的字段，用于表达联系。
唯一约束 UQ	字段或字段组合不允许重复。
范式	关系模式的规范化程度，本文按第三范式（3NF）设计。
ORM	对象关系映射，用程序对象操作数据库表。
DDL	数据定义语言，建表、建索引的 SQL 语句。

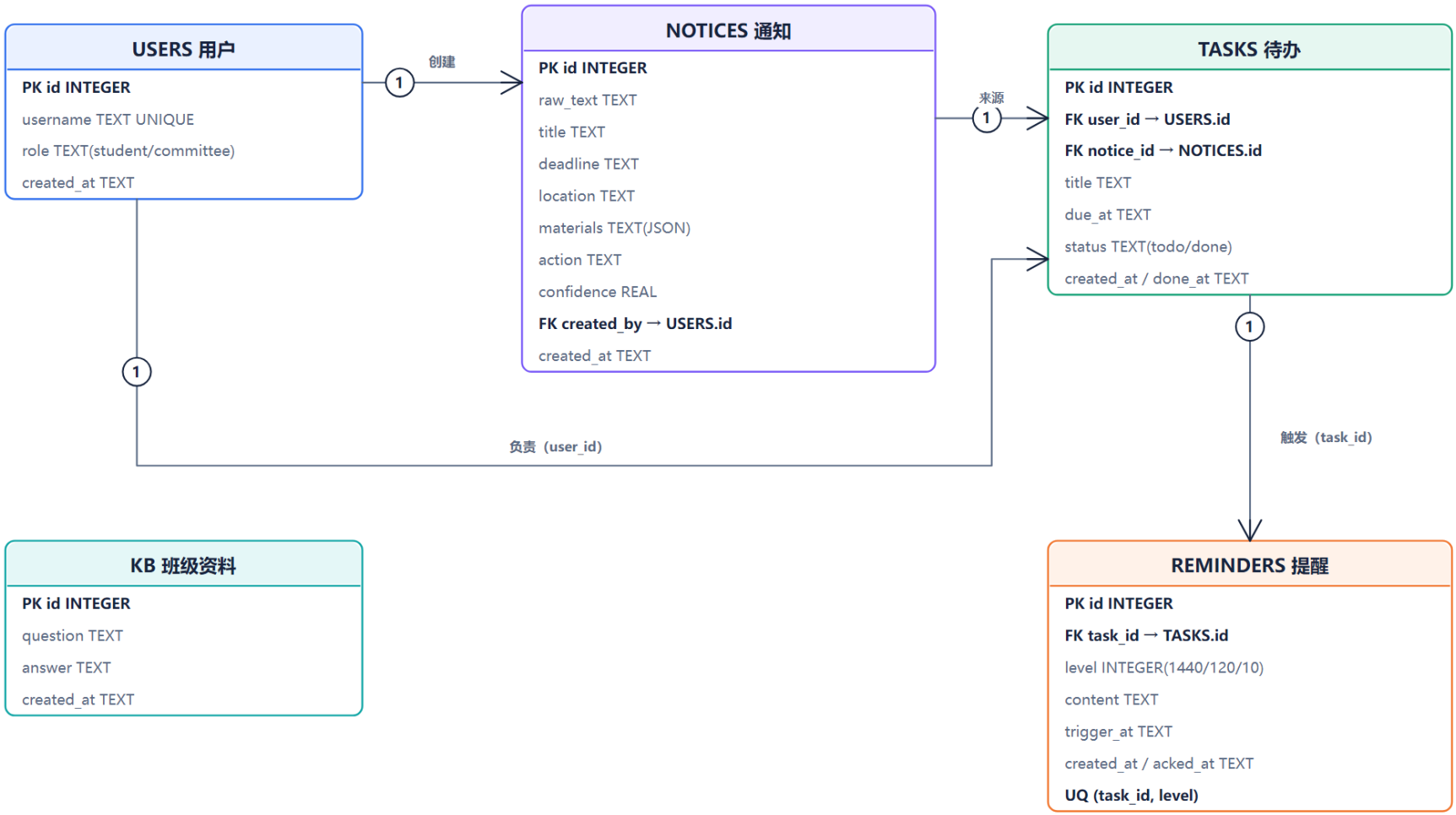
1.4 参考资料

- GB/T 8567—2006《计算机软件文档编制规范》（数据库设计说明部分）
- 《智班工坊_软件需求规格说明书》V1.1
- 《智班工坊_系统设计说明书》V1.0
- SQLite 官方文档（数据类型、约束、WAL 模式）

2 概念结构设计

2.1 实体—联系图

系统有 5 个核心实体。用户与通知之间是「创建」联系（1:N）；用户与待办之间是「负责」联系（1:N）；通知与待办之间是「来源」联系（1:N）；待办与提醒之间是「触发」联系（1:N）；班级资料为独立实体，由班委维护，不设外键。



说明：PK 主键、FK 外键、UQ 唯一约束；TASKS.status 取值为 todo / done；REMINDERS 的 (task_id, level) 唯一约束保证同一档提醒不重复入库。

KB（班级资料）由班委维护，不带外键，供答疑检索使用。

图 2-1　班小助数据库 ER 图

2.2 实体与联系说明

表 2-1 实体与联系

实体 / 联系	说明	基数	主要属性
用户 USERS	使用系统的班级成员，用角色区分同学与班委	—	昵称、角色
通知 NOTICES	一次解析所对应的原始通知与解析结果	—	原文、标题、截止时间、地点、材料、置信度
待办 TASKS	用户确认后进入个人日程的任务	—	标题、截止时间、状态
提醒 REMINDERS	待办的到点提醒流水	—	档位、内容、触发时间、确认时间
班级资料 KB	答疑用语料，一问一答	—	问题、答案
创建	班委或同学贴入通知触发解析	USERS 1 : N NOTICES	—
负责	待办归属于某个用户	USERS 1 : N TASKS	—
来源	待办由哪条通知产生（可为空，支持手工待办）	NOTICES 1 : N TASKS	—
触发	一个待办最多产生三条提醒（三档各一条）	TASKS 1 : N REMINDERS	—

2.3 概念设计要点

- **原文必须留存**：notices.raw_text 保存完整原文，所有字段证据都能回溯到这段文本，这是「可核对」设计的存储基础。
- **解析与日程解耦**：解析结果不直接生成待办，而是先落 notices，再由用户确认生成 tasks，避免误解析污染用户日程，也让「一次解析、多人加入」成为可能。
- **提醒可审计**：reminders 保存生成时间与确认时间（acked_at），既能去重，也能统计到达率。
- **资料与用户解耦**：kb 不设 created_by 外键，班级资料属于集体资产，不因班委换届而失效。

3 逻辑结构设计

3.1 关系模式

把 ER 图转换为关系模式（下划线表示主键，斜体表示外键）：

- USERS(<u>id</u>, name, role, created_at)
- NOTICES(<u>id</u>, raw_text, title, deadline, location, materials, action, confidence, model, *created_by*, created_at)
- TASKS(<u>id</u>, *user_id*, *notice_id*, title, due_at, status, created_at, done_at)
- REMINDERS(<u>id</u>, *task_id*, level, content, created_at, acked_at)
- KB(<u>id</u>, question, answer, created_at)

说明：materials 字段存储材料清单的 JSON 数组字符串（SQLite 无原生数组类型），在 ORM 层序列化/反序列化；notice_id 在 TASKS 中允许为空，以支持用户手工新增待办。

3.2 范式分析

五张表均满足第三范式（3NF）：每个非主属性完全函数依赖于主键，且不存在传递依赖。举例：若把「用户名」冗余进 tasks 表，则 tasks 会通过 user_id 传递依赖于 name，违反 3NF，也带来改名不一致的风险，因此不冗余；查询时用 JOIN 或两次查询组装。materials 作为 JSON 字符串属于「多值属性」的工程妥协：材料清单条目数不定且无需按条目查询，拆表会带来额外 JOIN 成本而无收益。

3.3 表结构详细设计

表 3-1 users（用户表）

字段	类型	约束	说明
id	INTEGER	PK, AUTOINCREMENT	用户编号
name	TEXT	NOT NULL, UNIQUE	昵称，登录标识（本期不设密码）
role	TEXT	NOT NULL, CHECK(student/committee)	角色：同学 / 班委
created_at	TEXT	NOT NULL	创建时间，ISO 8601 字符串

表 3-2 notices（通知表）

字段	类型	约束	说明
id	INTEGER	PK, AUTOINCREMENT	通知编号
raw_text	TEXT	NOT NULL	通知原文（解析证据的唯一来源）
title	TEXT		解析出的标题
deadline	TEXT		截止时间，ISO 8601（如 2026-11-08T18:00）
location	TEXT		地点
materials	TEXT		材料清单，JSON 数组字符串
action	TEXT		需要做的事，一句话
confidence	REAL	DEFAULT 0	整体置信度 0～1
model	TEXT		解析所用模型标识，便于回溯
created_by	INTEGER	FK → users.id	解析发起人
created_at	TEXT	NOT NULL	解析时间

表 3-3 tasks（待办表）

字段	类型	约束	说明
id	INTEGER	PK, AUTOINCREMENT	待办编号
user_id	INTEGER	NOT NULL, FK → users.id	归属用户
notice_id	INTEGER	FK → notices.id, 可空	来源通知（手工待办为空）
title	TEXT	NOT NULL	待办标题
due_at	TEXT		截止时间，ISO 8601；为空表示无期限
status	TEXT	NOT NULL, CHECK(todo/done)	状态：未完成 / 已完成
created_at	TEXT	NOT NULL	创建时间
done_at	TEXT		完成时间

表 3-4 reminders（提醒表）

字段	类型	约束	说明
id	INTEGER	PK, AUTOINCREMENT	提醒编号
task_id	INTEGER	NOT NULL, FK → tasks.id	所属待办
level	INTEGER	NOT NULL, CHECK(1440/120/10)	档位（分钟）：1 天 / 2 小时 / 10 分钟
content	TEXT	NOT NULL	提醒文案（含待办标题与截止时间）
created_at	TEXT	NOT NULL	生成时间
acked_at	TEXT		确认时间；为空表示未读

约束 (task_id, level) 建唯一索引，保证同一待办的同一档提醒只生成一次——这是提醒幂等性的数据库级保障，比在代码里判重更可靠。

表 3-5 kb（班级资料表）

字段	类型	约束	说明
id	INTEGER	PK, AUTOINCREMENT	资料编号
question	TEXT	NOT NULL	标准问题（检索与展示用）
answer	TEXT	NOT NULL	标准答案（答疑时的依据）
created_at	TEXT	NOT NULL	创建时间

4 物理结构设计

4.1 存储组织

表 4-1 存储组织

项目	设计	理由
存储引擎	SQLite 3，单文件 banxiaozhu.db	免安装、随应用分发，适合单班级部署
日志模式	journal_mode = WAL	读写并发更友好，崩溃恢复更快
文件位置	应用目录 /app/data（可配置）	便于整目录备份与迁移
字符集	UTF-8	中文通知原文完整保存
时间存储	TEXT（ISO 8601）	可排序、可读、跨语言一致

4.2 索引设计

表 4-2 索引设计

索引名	表 / 字段	类型	用途
idx_users_name	users(name)	唯一	登录查人，防重名
idx_notices_created	notices(created_at DESC)	普通	按时间倒序查看历史解析
idx_tasks_user	tasks(user_id, status)	普通	查某用户未完成待办（最高频查询）
idx_tasks_due	tasks(due_at)	普通	守护线程扫描到期待办
uq_reminders_task_level	reminders(task_id, level)	唯一	保证三档提醒不重复
idx_reminders_pending	reminders(task_id, acked_at)	普通	查未确认提醒
idx_kb_created	kb(created_at DESC)	普通	资料列表倒序

4.3 完整性约束

- **实体完整性**：五张表均以自增整型主键保证行唯一。
- **参照完整性**：tasks.user_id、tasks.notice_id、reminders.task_id 为外键；删除待办前先删其提醒，删除通知不级联删除待办（保留用户日程），仅在界面上标注来源已删除。
- **用户定义完整性**：role 限定 student/committee；status 限定 todo/done；level 限定 1440/120/10；confidence 限定 0～1。
- **唯一性**：name 唯一，防止同名用户混淆；(task_id, level) 唯一，保证提醒幂等。

4.4 容量与性能估算

表 4-3 容量估算

对象	估算行数	单行大小	年增长	说明
users	≤ 100	~80 B	换一届一次	一个班级规模
notices	≈ 1 000 / 年	~600 B	线性	每天 3～6 条通知
tasks	≈ 3 000 / 年	~150 B	线性	每条通知平均 2～3 人加入
reminders	≈ 9 000 / 年	~200 B	线性	每个待办 3 档
kb	≈ 200 条	~400 B	缓慢	班委维护的常见问答

按上表估算，全年数据量在 10 MB 量级，SQLite 单文件完全承载；关键查询（按用户查待办、按时间扫提醒）都有对应索引，单次查询在毫秒级。

5 对象关系映射（ORM）设计

5.1 映射策略

本期采用「轻量映射」：不引入完整 ORM 框架，而是在数据访问层用数据类（dataclass / 字典）承载一行记录，配合参数化 SQL 完成转换。这样既保留了 ORM 的表达力（以对象操作数据），又避免了框架学习成本与额外依赖。表 5-1 给出实体类与表的对应关系。

表 5-1 实体类与表映射

实体类	对应表	字段映射	说明
User	users	id → id, name → name, role → role, created → created_at	角色用枚举字符串
Notice	notices	id, raw_text → raw, title, deadline, location, materials(JSON→list), action, confidence, model, created_by → created_by, created_at	materials 需反序列化
TaskItem	tasks	id, user_id, notice_id, title, due_at, status, created_at, done_at	status 用 todo/done
Reminder	reminders	id, task_id, level, content, created_at, acked_at	level 单位分钟
KbItem	kb	id, question, answer, created_at	问答对

5.2 数据访问层方法清单

表 5-2 数据访问层方法

方法	作用	涉及表	关键约束
init_db()	建表、建索引、初始化字典数据	全部	幂等，重复执行无副作用
get_or_create_user(name, role)	按昵称取用户，不存在则创建	users	name 唯一
save_notice(...)	保存原文与解析结果，返回通知 id	notices	raw_text 必填
list_notices(limit)	按时间倒序返回解析历史	notices	走 idx_notices_created
create_task(user_id, notice_id, title, due_at)	新建待办	tasks	user_id 必填

方法	作用	涉及表	关键约束
update_task(id, title, due_at)	改标题/截止时间，并清空旧提醒	tasks, reminders	改时间需重算提醒
delete_task(id)	先删提醒再删待办	tasks, reminders	事务内完成
list_tasks(user_id)	返回该用户待办（未完成优先）	tasks	走 idx_tasks_user
mark_done(id)	置 status=done 并记 done_at	tasks	幂等
scan_reminders()	扫描到期待办并写三档提醒	tasks, reminders	INSERT OR IGNORE
pending_reminders(user_id)	返回未确认提醒	reminders, tasks	acked_at 为空
ack_reminder(id)	置 acked_at	reminders	幂等
search_kb(question, top=4)	字符重合打分检索资料	kb	分数阈值过滤
add_kb / update_kb / delete_kb	资料增删改	kb	仅班委可调

5.3 JSON 字段与枚举约定

- materials：存 JSON 数组字符串，如 ["成绩单","家庭经济困难证明"]；读出时用 json.loads，解析失败按空数组处理（不抛异常，保证页面可用）。
- role：student / committee，风险在于前端传值污染，写入前用白名单校验。
- status：todo / done，仅由 mark_done 与 update_task 改写。
- level：1440 / 120 / 10（分钟），与提醒文案「提前 1 天 / 2 小时 / 10 分钟」一一对应。

6 初始化数据与数据字典

6.1 字典数据

表 6-1 枚举字典

字典	取值	含义
role	student	普通同学：解析、待办、答疑
role	committee	班委：额外拥有资料与通知管理权限
status	todo	待办未完成
status	done	待办已完成
level	1440	提前 1 天提醒
level	120	提前 2 小时提醒
level	10	提前 10 分钟提醒

6.2 初始资料数据

首次建库时写入 5 条班级高频问答，保证答疑功能开箱可用：

表 6-2 初始班级资料

#	问题	答案要点
1	学生证丢了怎么补办？	到实验楼 A302 办公室登记，带一寸照片与 10 元工本费，约 3 个工作日可取。
2	请假流程怎么走？	先在辅导员处领请假单，填写后由班长签字，再交辅导员备案；病假需附校医院证明。
3	励志奖学金什么时候评？	每学年秋季学期评定，通常在 10 月中旬发布通知，材料交至 A302。
4	教室怎么借用？	提前 3 个工作日在教务系统提交借用申请，注明用途与人数，审批后到 A302 取钥匙。
5	四六级成绩单在哪领？	成绩单由教务处统一发放到班级，班长领取后分发；遗失可到 A302 申请补打。

7 安全性与维护设计

7.1 数据安全

- 数据库文件不对外暴露，仅由应用进程访问；不放在静态资源目录下。
- 应用不收集与业务无关的个人信息，用户以昵称标识，不存手机号与身份证号。
- 通知原文属班级内部信息，导出功能仅对班委开放。
- 备份文件命名带时间戳，避免覆盖；备份不包含模型 API Key（Key 存于配置文件）。

7.2 备份与恢复

表 7-1 备份与恢复策略

场景	策略	恢复方式
日常	每周导出一次 banxiaozhu.db 副本	替换文件后重启服务
误删待办	删除为硬删除，先确认无误再执行	从最近备份恢复，或按 notices 重新加入
文件损坏	WAL 模式下写入中断概率低	删除 -wal/-shm 后重启；失败则用备份
换机迁移	整目录拷贝（含 db 与 .env）	目标机器直接启动

7.3 演进与迁移

后续迭代若引入新字段（如 notice.source、task.priority），采用「加列 + 默认值」的兼容迁移：建表语句使用 CREATE TABLE IF NOT EXISTS，新增列通过 ALTER TABLE ADD COLUMN 完成，旧数据不受影响。若未来迁移到 PostgreSQL，字段类型映射为：INTEGER 自增 → SERIAL、TEXT → VARCHAR/TEXT、REAL → DOUBLE PRECISION，JSON 字符串可升级为 JSONB。

附录 A　建库脚本（DDL）

以下脚本为设计交付物，可直接用于建库，也是 ORM 层的实现依据：

```
PRAGMA journal_mode = WAL;
PRAGMA foreign_keys = ON;

CREATE TABLE IF NOT EXISTS users (
  id          INTEGER PRIMARY KEY AUTOINCREMENT,
  name        TEXT NOT NULL UNIQUE,
  role        TEXT NOT NULL DEFAULT 'student'
              CHECK (role IN ('student', 'committee')),
  created_at  TEXT NOT NULL
);
CREATE UNIQUE INDEX IF NOT EXISTS idx_users_name ON users(name);

CREATE TABLE IF NOT EXISTS notices (
  id          INTEGER PRIMARY KEY AUTOINCREMENT,
  raw_text    TEXT NOT NULL,
  title       TEXT,
  deadline    TEXT,
  location    TEXT,
  materials   TEXT,
  action      TEXT,
  confidence  REAL DEFAULT 0,
  model       TEXT,
  created_by  INTEGER REFERENCES users(id),
  created_at  TEXT NOT NULL
);
CREATE INDEX IF NOT EXISTS idx_notices_created ON notices(created_at DESC);

CREATE TABLE IF NOT EXISTS tasks (
  id          INTEGER PRIMARY KEY AUTOINCREMENT,
  user_id     INTEGER NOT NULL REFERENCES users(id),
  notice_id   INTEGER REFERENCES notices(id),
  title       TEXT NOT NULL,
  due_at      TEXT,
  status      TEXT NOT NULL DEFAULT 'todo' CHECK (status IN ('todo', 'done')),
  created_at  TEXT NOT NULL,
  done_at     TEXT
);
CREATE INDEX IF NOT EXISTS idx_tasks_user ON tasks(user_id, status);
CREATE INDEX IF NOT EXISTS idx_tasks_due  ON tasks(due_at);

CREATE TABLE IF NOT EXISTS reminders (
  id          INTEGER PRIMARY KEY AUTOINCREMENT,
  task_id     INTEGER NOT NULL REFERENCES tasks(id),
  level       INTEGER NOT NULL CHECK (level IN (1440, 120, 10)),
  content     TEXT NOT NULL,
  created_at  TEXT NOT NULL,
  acked_at    TEXT
);
CREATE UNIQUE INDEX IF NOT EXISTS uq_reminders_task_level ON reminders(task_id, level);
CREATE INDEX IF NOT EXISTS idx_reminders_pending ON reminders(task_id, acked_at);

CREATE TABLE IF NOT EXISTS kb (
  id          INTEGER PRIMARY KEY AUTOINCREMENT,
  question    TEXT NOT NULL,
  answer      TEXT NOT NULL,
  created_at  TEXT NOT NULL
);
CREATE INDEX IF NOT EXISTS idx_kb_created ON kb(created_at DESC);
```

附录 B　数据类型约定

表 B-1　数据类型映射

逻辑类型	SQLite 类型	约定	示例
整数	INTEGER	编号、档位分钟数	1440
短文本	TEXT	标题、地点、角色、状态	实验楼 A302 办公室
长文本	TEXT	通知原文、答案	（多行文本）
时间	TEXT	ISO 8601，本地时区省略时按 Asia/Shanghai	2026-11-08T18:00
小数	REAL	置信度 0～1，保留两位	0.95
结构化	TEXT(JSON)	材料清单等数组型字段	["成绩单"]