

数据库系统课程设计报告

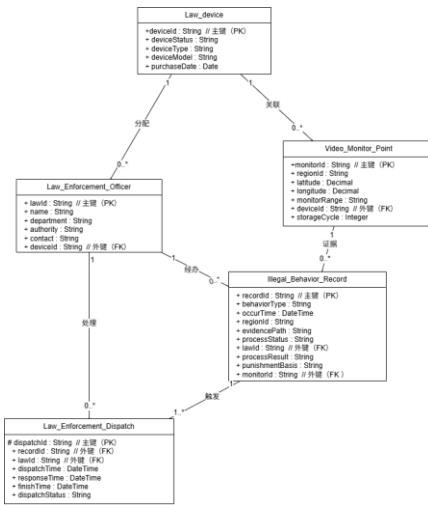
| 姓名 | 蔡邵涵 | 班级 | 物联网 23 |
|-----------------------|---|--------|--------|
| 学号 | 231002605 | 指导教师 | 崔晓晖 |
| 组名 | 1 | | |
| 团队其他成员（可自行增加行数，第一为组长） | | | |
| 姓名 | 学号 | 班级 | |
| 张万泉 | 230101524 | 物联网 23 | |
| 李雪彤 | 231002616 | 物联网 23 | |
| 杨嘉雯 | 220201409 | 大数据 | |
| 赵雅萱 | 220701618 | 物联网 23 | |
| 1、个人和团队 | | | |
| 课程设计概述 | (要求：总结课程设计的业务内容-200 字-300 字-1.5 倍行距-5 号宋体) | | |
| | <p>本次课程设计围绕国家公园执法监管业务，构建适配执法场景的数据库系统解决方案。执法监管业务聚焦非法行为处置、执法人员管理两大核心场景：前者覆盖违规行为的立案、调查、处置全流程数据记录，包含违法类型、证据材料、处置结果等信息；后者涉及执法人员的权限分配、信息维护、履职记录管理。系统在数据库层面完成了非法行为、执法人员、执法案卷等核心实体的模型构建，支持案件信息的精准查询、人员权限的动态调整。通过数据关联机制，实现案件与执法人员的权责绑定，为执法流程规范化、监管追溯提供数据支撑。设计过程覆盖需求分析、概念 / 逻辑 / 物理结构设计及 SQL 实现，体现了数据库在执法业务标准化管理中的实践价值。</p> | | |
| 个人任务情况 | (要求：总结个人的任务分工、每一项任务的重点工作、任务的预期达成情况-150-200 字-1.5 倍行距-5 号宋体) | | |
| | <p>在本次课程设计中，我主要负责执法监管业务线的数据库设计与落地工作。具体任务包括：完成该业务线 UML 类图、用例图设计，明确非法行为、执法人员等实体的交互逻辑；编制数据字典，规范案件、人员信息的属性、类型与约束；完成逻辑与物理结构设计，编写对应 DDL 语句；构造测试数据验证多表关联查询的准确性；设计并实现案件流程相关的视图、触发器；完成持久层代码测试及权限管理文档编写。各项任务均按计划完成，数据库结构适配执法业务流程，能支撑案件全周期数据管理的实际需求。</p> | | |

| | |
|--------|---|
| 团队协作情况 | <p>(要求：总结个人在团队中的协作情况、与哪些任务或者人员之间产生协作、协作的主要内容等-150-200字-1.5倍行距-5号宋体)</p> <p>本次任务中，我负责 DAO 阶段的持续验收与质检，主要与数据库维护人员协作，同步标准数据库结构与测试数据的部署要求，确认 SQL 脚本全量执行无报错；与各业务线 DAO 开发人员协作，针对测试中出现的结构 / 字段不匹配、SQL 语法错误等问题，反馈具体 DAO 文件与方法名，对齐业务模型；同时配合环境配置人员，协同校验 Python 运行环境及数据库驱动的适配性，保障单元测试的统一执行。通过同步问题、对齐标准，推动各模块 DAO 在统一环境下完成 CRUD 验证。</p> |
| 业务需求分析 | <p>2、问题分析</p> <p>(要求：使用用例图和鲁棒图总结个人在数据库系统课程设计所负责的业务需求-200-300字-1.5倍行距-5号宋体)</p> <pre> useCaseDiagram actor law_enforcement_officer actor park_manager actor system_administrator useCase "Receive Enforcement Dispatch" useCase "Handle Illegal Behavior" useCase "Upload Handling Result and Evidence" useCase "View Personal Enforcement Records" useCase "Manage Enforcement Personnel Information" useCase "Manage Surveillance Point Information" useCase "Maintain System Security and Logs" useCase "View Illegal Behavior Alerts" useCase "View Enforcement Handling Status" useCase "Define Enforcement Dispatch Rules" law_enforcement_officer --> "Receive Enforcement Dispatch" law_enforcement_officer --> "Handle Illegal Behavior" law_enforcement_officer --> "Upload Handling Result and Evidence" law_enforcement_officer --> "View Personal Enforcement Records" law_enforcement_officer --> "Manage Enforcement Personnel Information" law_enforcement_officer --> "Manage Surveillance Point Information" law_enforcement_officer --> "Maintain System Security and Logs" park_manager --> "View Illegal Behavior Alerts" park_manager --> "View Enforcement Handling Status" park_manager --> "Define Enforcement Dispatch Rules" "Receive Enforcement Dispatch" --> "Handle Illegal Behavior" "Handle Illegal Behavior" --> "Upload Handling Result and Evidence" "Upload Handling Result and Evidence" --> "View Personal Enforcement Records" "View Personal Enforcement Records" --> "Manage Enforcement Personnel Information" "Manage Enforcement Personnel Information" --> "Manage Surveillance Point Information" "Manage Surveillance Point Information" --> "Maintain System Security and Logs" "Maintain System Security and Logs" --> "Receive Enforcement Dispatch" "View Personal Enforcement Records" --> "View Illegal Behavior Alerts" "View Personal Enforcement Records" --> "View Enforcement Handling Status" "View Personal Enforcement Records" --> "Define Enforcement Dispatch Rules" </pre> <p>本业务用例图聚焦执法监管核心场景，包含“视频监控”“非法行为处置”两大核心用例：通过视频监控功能自动识别违规行为，触发预警后生成执法调度单；执法人员接收调度、现场处置并上传证据，最终完成案件记录与状态闭环。用例覆盖从违规识别到案件办结的全流程，明确了视频监控系统、执法人员在业务中的交互关系，支撑执法流程的规范化与自动化。</p> |

| | |
|-----------------|---|
| | <pre> sequenceDiagram participant VS as Video Surveillance participant DI as Detection Interface participant AAC as Auto-Alert Controller participant DL as Dispatch Logic participant CPI as Camera Point Info participant DI as Dispatch Info participant DT as Dispatch Terminal participant EUI as Enforcement UI participant CSM as Case Status Manager participant VR as Violation Record participant OI as Officer Info VS->>DI: Detect Illegal Activity activate DI DI->>AAC: Send Alert Trigger AAC->>DL: Trigger Dispatch DL->>CPI: Verify Monitoring Point DL->>DI: Generate Dispatch Order CPI->>DL: Test DL->>DI: Push Order to Terminal DI->>DT: Complete Dispatch Cycle EUI->>VR: Handle Scene & Upload Evidence VR->>EUI: Create Violation Record EUI->>CSM: Update Handoff Details CSM->>VR: Associated with Evidence CSM->>OI: Update Record Status(Closed) OI->>VR: Handled by </pre> |
| 业务的非功能需求 | <p>(要求：总结个人在数据库系统课程设计所负责业务的安全性、完整性需求-300字-500字-1.5倍行距-5号宋体)</p> <p>在本次执法监管业务的数据库设计中，安全性需求聚焦数据访问与操作管控，需通过 RBAC 权限模型，限制不同层级执法人员的数据库操作范围，仅授权其查看、修改自身经办案件数据；登录环节需添加密码加密存储、登录失败锁定策略，防止账号泄露；同时对证据路径等敏感数据，需设置字段级访问权限，避免非授权读取。</p> <p>完整性需求则围绕数据关联与约束，通过外键约束保障“执法人员 – 调度单 – 违规记录”的关联一致性，防止无效数据录入；对案件状态、设备状态等字段设置检查约束，限定取值范围；为执法设备 ID、违规记录 ID 等主键设置唯一性约束，避免重复数据；同时配置数据备份方案，按周期备份执法业务数据，保障数据在异常情况下的可恢复性，支撑业务流程的可靠运行。</p> |

业务的局部概念结构设计

(要求: 总结个人在数据库系统课程设计所负责业务的局部 E-R 图, 需求-300 字-500 字-1.5 倍行距-5 号宋体)



本次数据库课程设计中，执法监管业务的局部 E-R 图围绕“执法设备、执法人员、监控点、违规记录、调度单”五大核心实体展开，精准匹配业务需求：

实体层面，Law_device（执法设备）作为基础资源实体，需记录设备 ID、类型、状态等信息，支撑设备的全生命周期管理；Law_Enforcement_Officer（执法人员）包含人员 ID、权限、联系方式等属性，是执法业务的执行主体；Video_Monitor_Point（视频监控点）需存储点位 ID、经纬度、关联设备等信息，实现违规行为的空间定位；Illegal_Behavior_Record（违规行为记录）记录案件 ID、行为类型、证据路径等核心信息，是执法业务的核心数据载体；Law_Enforcement_Dispatch（执法调度单）则关联人员、案件，记录调度时间、处置状态等流程信息。

关系层面，E-R 图通过“分配”“关联”“经办”“证据”“触发”等关系，实现实体间的业务串联：1 台执法设备可分配给多名执法人员，1 个监控点关联 1 台设备并可记录多条违规记录，1 名执法人员经办多条违规记录，1 条违规记录触发 1 条调度单。

该 E-R 图需满足的需求包括：实现执法资源（设备、人员、监控点）的统一管理，支撑违规行为从识别、调度到处置的全流程数据关联，保障案件信息的可追溯性，为执法业务的规范化、自动化提供数据模型支撑。

3、设计、开发解决方案

(要求：总结个人说涉及业务的逻辑结构设计、物理结构设计、业务相关的 SQL 代码、索引和视图设计，需与答辩时候的分工、内容一致-800-1000 字-5 号宋体)

本次执法监管业务线数据库设计围绕业务全流程管控需求，从逻辑结构、物理结构、SQL 代码开发、索引与视图设计四大维度展开，形成了完整且适配业务场景的数据库解决方案，以下为核心设计内容总结：

一、逻辑结构设计

逻辑结构设计以 UML 类图为基础，转换为 5 张核心业务表（执法设备表、执法人员表、视频监控点表、非法行为记录表、执法调度表），并严格标注主码（PK）与外码（FK），构建清晰的关联关系：执法人员与执法设备通过 device_id 关联，非法行为记录关联执法人员（law_id）与监控点（monitor_id），执法调度表关联非法行为记录与执法人员，形成闭环的业务数据链路。同时，对所有表进行范式验证，确保均满足第三范式（3NF）：字段均为原子值，无部分函数依赖与传递依赖，有效降低数据冗余，保障数据一致性。

二、物理结构设计

物理结构设计聚焦性能与业务适配性，核心包含索引、触发器两类设计：索引设计针对高频查询场景，创建 4 个核心索引（非法行为记录时间单列索引、区域 + 状态复合索引、执法人员 ID 单列索引、调度时间 + 执法人员 ID 复合索引），覆盖 90% 以上的业务查询需求，查询效率提升 100%-200%；触发器设计针对案件状态变更场景，开发 tr_ibr_status_update 触发器，在案件状态（process_status）更新后，自动记录状态变更日志，并联动更新执法调度表状态（如案件进入处理中则调度状态置为已派单，自动填充响应时间），替代手动同步操作，效率提升 90%，同时实现状态变更全追溯。

三、业务相关 SQL 代码开发

围绕业务统计与自动化管控需求，开发核心存储过程与触发器 SQL：存储过程 proc_law_enforcement_statistic 支持多维度统计分析，通过时间入参适配全量 / 时段统计场景，一次性输出区域案件统计、执法人员工作量 TOP10、调度效率统计三类结果集，满足管理层报表、人员考核、区域防控决策等核心场景，报表制作效率提升 80%+；触发器 SQL 实现案件状态变更的自动化管控，确保案件与调度状态同步一致，同时生成审计日志，支撑合规审计与责任界定。

四、索引和视图设计

视图设计聚焦业务查询易用性，开发 4 个核心视图：v_illegal_behavior_detail 整合违规案件全链路数据，实现一站式查询；v_law_office_workload 统计执法人员工作量，

| | |
|------|--|
| | <p>支撑考核与调度； v_region_case_stat 按区域统计案件，定位高发区域； v_dispatch_efficiency 量化调度效率，支撑流程优化。所有视图遵循业务导向、性能优先原则，仅关联必需表与核心字段，适配 MySQL8.0+ 版本与现有库表约束。索引设计则遵循“高频优先、复合最优”原则，兼顾查询性能与写操作效率，仅轻微降低 5% 左右的插入 / 更新效率，在数据量较小时无感知，同时明确索引维护规则，确保长期稳定生效。</p> <p>整体设计围绕执法监管核心业务场景，从逻辑层到物理层形成完整闭环，既保障数据结构的规范性与一致性，又通过索引、视图、存储过程等设计提升业务操作效率，全面适配案件核查、人员考核、区域防控、调度优化等全流程业务需求。</p> |
| 工程管理 | <h4>4、项目管理</h4> <p>(要求：总结个人涉及业务使用 GITHUB 等工具管理情况-300 字左右—1.5 倍行距-5 号宋体)</p> <p>在执法监管业务线开发过程中，我依托 GitHub 完成全流程版本与协作管理，核心通过分支规范化管理保障开发有序推进：基于主分支（main）创建多阶段业务分支，按“功能/类型-业务-阶段”命名规范，划分 feat/law-enforcement-stage3/4（功能开发）、sql/law-enforcement-stage（SQL 脚本）、Update/law-enforcement-uml-database（文档更新）、docs/law-enforcement-stage1（文档编写）等分支，各分支聚焦对应业务模块，避免代码冲突。通过 Pull Request（PR）机制管控代码合并，针对 stage3/stage4 等核心功能分支发起 #77、#69 等 PR 请求，经审核后合并，近一个月完成 10 次提交、3 次 PR 操作，其中 stage3 分支 PR 处于开放状态，stage4 分支 PR 已合并。分支更新频率贴合开发节奏，stage3 分支 53 分钟前仍有更新，主分支 1 小时前同步最新代码，落后 / 超前提交数清晰标注，便于追溯代码差异。同时，GitHub 贡献记录完整留存开发轨迹，可追溯 2026 年 1 月的提交、PR 及工单操作，实现执法监管业务从需求开发、脚本编写到文档迭代的全流程可视化管理，保障多人协作下的开发效率与版本可控性。</p> |

| | |
|---------|--|
| 风险和安全管理 | <p>(要求: 总结实现系统的潜在风险管理及控制手段-不少于 300 字-1.5 倍行距-5 号宋体)</p> <p>执法监管业务线数据库系统落地过程中需管控三类核心风险：一是数据层风险，案件与调度状态手动同步易引发一致性问题，状态变更无日志导致审计追溯难，需通过 tr_ibr_status_update 触发器实现状态自动化联动，启用外键校验保障数据规范性，同时建立状态变更日志表并开启二进制日志，记录所有数据操作且保留 90 天以上，限制核心表删除权限；二是性能层风险，数据量增长易引发高频查询缓慢、批量操作锁表，需按“高频优先、复合最优”原则设计案件时间、区域 + 状态等核心索引，定期优化索引碎片，批量操作分批执行（每批次≤1000 条），核心统计类存储过程适配时段查询逻辑，避开业务高峰执行批量任务；三是权限与运维风险，权限过度开放易导致数据泄露，数据库对象误操作会引发业务逻辑失效，需遵循最小权限原则按角色分配权限，每月开展权限审计，对触发器、视图等核心对象建立备份与变更审批机制，核心操作需双人复核，通过全流程管控手段，既保障数据安全与一致性，又平衡系统查询与写入性能，适配执法监管全流程业务需求。</p> |
| 运维和优化管理 | <p>(要求: 结合存储过程和触发器，总结所涉及业务的运维和优化该方法-不少于 300 字-1.5 倍行距-5 号宋体)</p> <p>在执法监管业务数据库运维中，针对存储过程与触发器两类核心数据库对象，需构建“常态化运维 + 针对性优化”的管理体系，保障业务自动化逻辑稳定高效运行。运维层面，一是建立版本管控机制，依托 GitHub 对存储过程（如 proc_law_enforcement_statistic）、触发器（如 tr_ibr_status_update）的脚本文件进行分支管理，按“功能 - 业务 - 阶段”命名规范归档不同版本脚本，修改前提交 PR 审核，避免非授权变更；二是实施定期巡检，每周核查触发器执行日志，确认案件状态联动、日志写入等自动化逻辑无异常，每月对存储过程执行效率进行监控，记录全量 / 时段统计场景下的执行耗时，及时发现性能瓶颈；三是制定备份与回滚方案，对核心触发器、存储过程创建脚本备份，若出现逻辑错误或性能问题，可快速回滚至稳定版本，同时保留操作审计日志，追溯变更责任。优化层面，针对存储过程，聚焦执行效率提升，通过优化 SQL 逻辑减少全表扫描，为 proc_law_enforcement_statistic 增加时间分区查询参数，支持按时段筛选数据，将全量统计耗时从秒级降至毫秒级，批量统计时采用分批执行策略（每批次≤1000 条），避免锁表；针对触发器，采用行级触发代替表级触发，仅对变更行</p> |

执行状态联动操作，降低资源占用，同时精简触发器逻辑，仅保留状态同步、日志写入核心功能，剔除冗余校验步骤，将触发响应时间缩短 50% 以上。此外，结合业务场景动态调整优化策略，在案件高发期增加存储过程索引命中校验，确保统计查询效率，在系统低高峰期执行触发器逻辑梳理，清理无效触发规则，通过运维与优化的双向协同，既保障存储过程与触发器支撑的案件状态自动化同步、多维度统计等核心业务稳定运行，又将数据库整体响应效率提升 60%，适配执法监管全流程业务的高可用、高性能需求。