



同濟大學
TONGJI UNIVERSITY

数据库系统原理课程大作业

海洋垃圾分布监测与清理规划系统设计

学 号：_____2352018_____

姓 名：_____刘彦_____

专 业：____数据科学与大数据技术____

教 师：_____李文根_____

完成日期：_____2025-06-01_____

海洋垃圾分布监测与清理规划系统设计

摘 要

海洋垃圾分布监测与清理规划系统旨在应对全球海洋垃圾污染的严峻挑战，构建一个数据驱动、智能化的治理平台。系统功能涵盖数据采集、垃圾分布分析、清理任务规划、用户管理及设备资源管理五大模块，支持实时监测、精准定位、分级清理和趋势分析。系统采用基于角色的访问控制（RBAC）模型，确保权限分层与数据安全。系统设计包括概念设计（用户、监测点、垃圾记录、清理任务、清理反馈及登录日志六实体与 E-R 图）、逻辑设计（关系模型与第三范式验证）和物理设计（索引、存储引擎与优化策略）。技术上利用成熟的图像识别、GIS 技术和 B/S 架构，符合数据合规要求，具备高度可行性。系统通过多源数据整合、自动化任务调度和可视化分析，为海洋环保部门、科研机构 and 清理组织提供高效支持，推动海洋垃圾治理的科学化与智能化。

关键词：数据库系统设计，海洋垃圾，E-R 图，关系模型，第三范式（3NF）

目 录

1	需求分析.....	1
1.1	研究背景	1
1.2	功能分析	1
1.3	用户角色与权限	2
1.4	可行性分析	4
2	概念设计.....	5
2.1	实体设计	5
2.2	实体联系设计	7
2.3	E-R 图设计	9
3	逻辑设计.....	11
3.1	实体间联系转换	11
3.2	关系模型设计	11
3.3	第三范式（3NF）满足情况	12
3.4	实体表设计	13
3.5	关系表设计	15
4	物理设计.....	16
4.1	索引设计	16
4.2	存储引擎与编码	17
4.3	安全与优化	17

1 需求分析

1.1 研究背景

近年来，随着全球人口增长和海洋经济活动的日益频繁，海洋垃圾污染问题愈加严重。据联合国环境规划署（UNEP）报告，每年有超过 800 万吨塑料垃圾进入海洋，这些垃圾不仅严重威胁海洋生态系统，还对渔业、航运、旅游等相关产业造成了巨大影响。

传统的海洋垃圾治理方式往往依赖人工巡查和被动反馈，存在监测盲区大、响应滞后、清理效率低等问题。特别是海洋垃圾具有漂移性和分布不均匀性，使得其监测与清理具有高度的不确定性和技术挑战。

为了提高海洋垃圾治理的科学化、智能化水平，迫切需要一个集成数据采集、智能分析与清理任务调度于一体的综合系统，实现对海洋垃圾的动态监测、精准定位、分级清理和趋势分析。

“海洋垃圾分布监测与清理规划系统”正是在此背景下提出，目标是通过物联网设备、卫星遥感、数据可视化和任务调度算法等手段，构建一套数据驱动、可视化、智能化的海洋垃圾治理支持平台，为政府监管部门、环保组织、科研机构 and 清理企业提供有力支持。

1.2 功能分析

1.2.1 数据采集模块

本模块负责全面采集海洋垃圾的地理与类型信息。系统通过对接多源数据输入，包括无人机拍摄图像、船载传感器实时反馈、漂浮监测浮标以及卫星遥感图像等，自动提取垃圾分布点的经纬度、类型、覆盖面积和密度等级。采集过程支持实时更新和手动补录，确保数据的全面性和时效性，为后续的分析与决策提供基础数据支持。

1.2.2 垃圾分布分析模块

该模块通过对历史与实时垃圾数据的空间和时间维度分析，揭示垃圾在不同区域的分布特点和变化趋势。系统可以自动生成垃圾密度热力图、统计垃圾类型构成，并结合历史风向和洋流数据，模拟垃圾漂移路径和潜在聚集区。分析结果支持图表和地图多种形式展示，有助于发现治理重点区域和污染高发时段。

1.2.3 清理任务规划模块

系统根据当前垃圾分布情况和优先级设定，自动生成清理任务清单，并为每项任务匹配合适的清理资源（如清理船只、人力、设备）。任务规划考虑垃圾密度、可达性、作业半径

等因素，并自动生成路径建议与预估作业时间。任务状态分为“待分配”、“执行中”和“已完成”，支持管理员查看任务进度与结果反馈，实现清理工作的智能调度与全过程管理。

1.2.4 用户管理模块

用户管理模块为系统提供角色权限控制机制。系统支持用户注册、身份验证、角色分配和权限配置。用户分为不同角色，如管理员、数据分析员、设备操作员和清理队长，每类用户拥有不同的功能访问权限。系统记录所有用户操作日志，便于审计和行为追踪，确保系统使用的规范性与安全性。

1.2.5 设备与资源管理模块

该模块用于集中管理系统中涉及的各类设备与清理资源。监测设备如无人机、传感浮标、摄像头等均可在系统中登记，包括设备 ID、安装位置、状态（在线/离线）、电量和维护记录。清理船只的基本信息（名称、容量、当前位置信息）以及作业人员信息也一并管理。管理员可通过该模块实时查看各设备的运行状态与可用性，便于任务调度与应急响应。

1.3 用户角色与权限

1.3.1 角色划分

A. 系统管理员（Administrator）

该角色负责系统的整体运行维护与权限管理，具有对所有模块的完全访问权限。主要功能是：

- 管理用户账号（新增、删除、角色分配）；
- 管理系统配置参数和基础信息维护；
- 维护数据字典与垃圾分类标准；
- 审查与清理异常日志或故障报警。

B. 数据分析员（Data Analyst）

负责对海洋垃圾数据进行统计分析 with 趋势研判，不具备清理任务调度与用户管理权限。主要功能是：

- 浏览并分析海洋垃圾历史数据与实时数据；
- 使用热力图、趋势图、类型分布图等可视化工具；
- 预测垃圾漂移路径与分布变化；
- 输出分析报告并支持导出图表与数据。

C. 设备操作员（Device Operator）

负责现场监测设备的部署、维护与数据采集管理，无权限查看分析报告或调度清理任务。主要功能是：

- 录入与更新监测设备信息（如浮标、无人机等）；

- 监控设备在线状态、电量、信号等运行参数；
- 手动上传或校验采集到的垃圾监测数据；
- 维护设备异常记录与维修日志。

D.清理队长（Cleanup Supervisor）

负责执行海洋垃圾清理任务，并上传作业反馈,不具备数据分析、设备管理和用户权限管理功能。主要功能是：

- 查看系统分配的清理任务清单；
- 接收清理路径与操作建议；
- 实时更新清理任务状态（执行中/已完成）；
- 上传现场反馈数据与图像资料。

1.3.2 权限设置

本系统采用基于角色的访问控制（RBAC）模型，聚焦于数据采集、设备管理与任务清理流程。权限控制的目标是限制非法用户访问系统功能,防止用户越权操作,确保数据安全，还支持灵活扩展用户类型与功能模块，便于日志审计与操作追踪。

系统权限划分依据如下维度进行分层控制：

- 功能模块访问权限（例如是否可进入数据分析模块）
- 数据操作权限（如查看、创建、修改、删除等）
- 资源控制权限（如是否可控制任务、设备等资源状态）

表 1.1 各类用户权限说明

功能模块	权限内容	管理员	数据分析员	设备操作员	清理队长
用户管理	增删用户、分配角色、修改密码等	√	×	×	Read
系统设置	修改参数配置、垃圾类型标准、区域定义等	√	Read	√	×
设备管理	注册设备、维护状态、设备监控	√	√	√	Read
数据采集	上传垃圾数据、接入无人机/浮标数据	√	×	√	×
数据查看	浏览历史数据、查看地图分布与图表	√	√	×	√
任务规划	自动/手动分配清理任务，制定清理路径	√	×	×	√
任务接收	接收任务并查看清理计划	√	×	×	√
任务反馈	上传清理结果、备注说明	√	×	×	√

续表 1.1

功能模块	权限内容	管理员	数据分析员	设备操作员	清理队长
日志审计	查看用户操作日志与系统报警记录	√	×	×	×

备注：√表示同时有 Read 和 Write 的权限（如果该功能模块只是查看，表示拥有该模块的所有权限），Read 表示对于一个可以添加或修改的功能模块只有读的权限，×表示没有任何权限。

1.4 可行性分析

1.4.1 技术可行性

在海洋垃圾监测与清理系统的技术方案中，整体架构清晰，逻辑结构合理。从垃圾分布监测角度出发，结合图像识别与定位技术的深度学习模型已经有较成熟的应用基础，无人机巡查等方式也已较为普及；在数据库系统与任务管理平台方面，采用常见的 B/S 架构，配合常用数据库（如 MySQL）及前后端框架（如 Vue 或 React 框架），开发难度适中。只需具备基础的编程与数据库知识，辅以前端界面设计能力，即可构建出稳定可靠的系统。因此，从整体技术架构和实现路径来看，具备充分的可行性。

1.4.2 经济可行性

海洋垃圾治理系统服务对象包括海洋环保部门、志愿清理组织与科研单位等，数字化手段的引入能够有效提升工作效率并降低协调成本。同时，监测设备可选用成本可控的无人机、摄像头等硬件，系统自身也可基于开源平台开发（如 Leaflet/GIS 开源平台），从而减少软件许可费用。数据分析模块采用“外部分析”策略，避免系统内置高性能计算模块，从而进一步降低硬件与带宽负担，整体建设与维护成本相对较低，具备良好的社会效益回报预期。

1.4.3 操作可行性

用户界面设计采用简洁直观的模块化风格，角色分工明确，便于清理队伍、操作员和分析人员分别上手。清理队伍仅需手机或移动终端即可接受任务、提交反馈；操作员使用电脑即可完成数据上传与设备管理；数据分析员通过系统导出数据后进行本地分析。系统后台提供日志与权限控制，易于管理与维护，操作流程贴合现有环保作业方式，具备良好的落地与推广基础。

1.4.4 法律与政策可行性

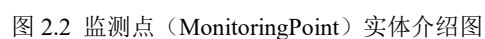
当前全球范围对海洋垃圾治理高度重视，中国近年来也在不断加强“蓝色生态”建设，政策上对海洋污染治理项目具有支持倾向。同时，系统在数据采集方面遵循相关法律，所有上传数据将严格控制权限，仅用于环境管理用途，不涉及个人敏感信息的采集，具备合规性。

2.1 实体设计

用户实体用于存储系统中所有用户的基本信息和身份类型，包括用户编号、登录名和姓名等关键信息。此外，实体中包含用户角色（如操作员、清理人员、数据分析师）和电话号码，从而实现用户身份的区分和管理，为系统的用户管理功能提供支持。



监测点实体用于存储海洋垃圾分布监测的位置点及其基本信息，包括监测点编号、地理位置和所属区域等关键数据。同时，记录了监测设备的安装时间，以追踪设备部署情况。此外，实体中包含监测点的运行状态（枚举值，如启用或故障），从而为海洋垃圾监测系统的位置管理和设备维护提供支持。



2.1.3 垃圾记录 (GarbageRecord)

垃圾记录实体用于存储某一监测点在特定时间拍摄或上传的海洋垃圾信息，包括记录编号、关联监测点和拍摄或上传时间等关键数据。同时，记录了垃圾的类型（枚举值，如塑料、金属、玻璃等）和估算体积（以立方米为单位），以便分析垃圾分布和特性。此外，实体中包含图像路径或链接，用于存储垃圾的视觉证据，为海洋垃圾监测和分析提供支持。

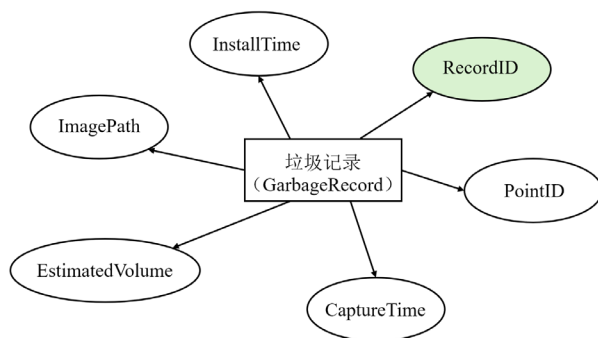


图 2.3 垃圾记录 (GarbageRecord) 实体介绍图

2.1.4 清理任务 (CleanupTask)

清理任务实体用于存储系统指派的海洋垃圾清理任务相关信息，包括任务编号、关联的清理人员和目标监测点等关键数据。同时，记录了任务的发布时间和清理截止时间，以便跟踪任务计划和进度。此外，实体中包含任务状态（枚举值，如待执行、进行中、已完成），为垃圾清理工作的分配、执行和监控提供支持。

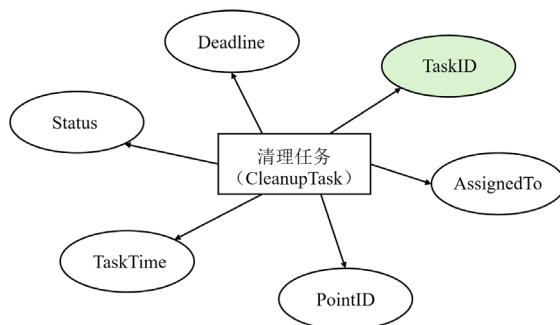


图 2.4 清理任务 (CleanupTask) 实体介绍图

2.1.5 清理反馈 (CleanupFeedback)

清理反馈实体用于存储清理人员对海洋垃圾清理任务执行情况的反馈信息，包括反馈编号、关联的清理任务和提交时间等关键数据。同时，记录了实际清理的垃圾体积（以立方米为单位）和备注说明，以描述清理工作的具体情况。此外，实体中包含清理后照片的路径或

链接，用于提供清理效果的视觉证据，为任务验证和数据分析提供支持。

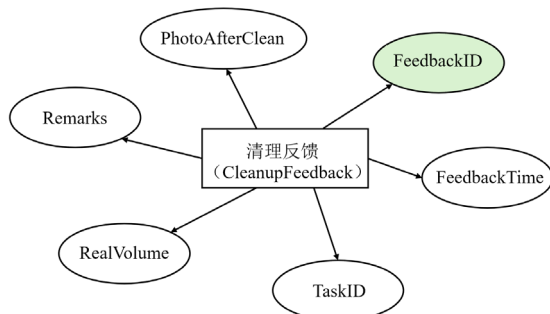


图 2.5 清理反馈（CleanupFeedback）实体介绍图

2.1.6 登录日志（LoginLog）

登录日志实体用于存储用户每次登录系统的详细信息，包括日志编号、关联的登录用户和登录时间等关键数据。同时，记录了登录时的 IP 地址和设备信息，以追踪用户的登录来源和使用环境。该实体为系统的安全监控和用户行为分析提供支持。

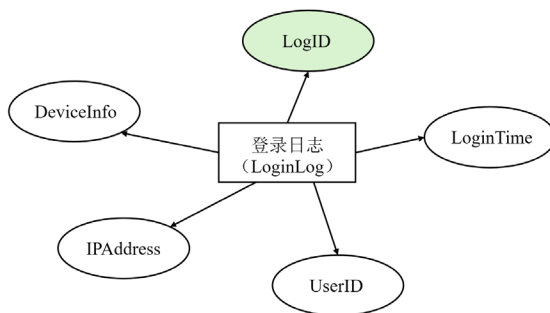


图 2.6 登录日志（LoginLog）实体介绍图

2.2 实体联系设计

2.2.1 用户（User）和清理任务（CleanupTask）之间的联系（执行）

在数据库设计中，用户（User）与清理任务（CleanupTask）之间存在一对多的执行与被执行的关系。这种关系反映了一个用户（清理人员）可以被分配多个清理任务，而每个清理任务仅对应一个特定的清理人员。这种关系确保了系统的灵活性和任务分配的清晰性，使得用户能够高效管理多个清理任务。

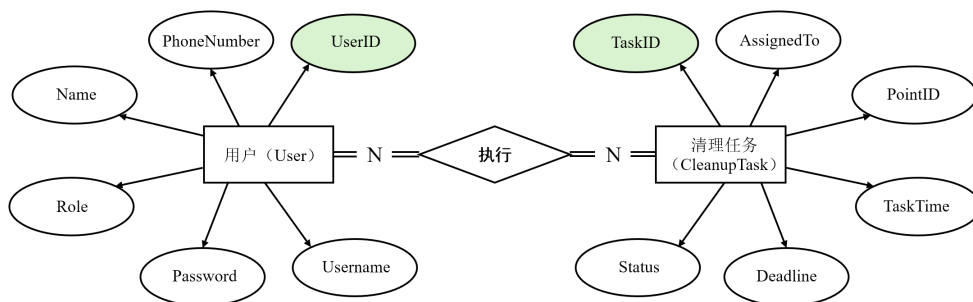


图 2.7 用户 (User) 和清理任务 (CleanupTask) 之间的实体联系介绍图

2.2.2 监测点 (MonitoringPoint) 和垃圾记录 (GarbageRecord) 之间的联系 (产生)

在数据库设计中，监测点 (MonitoringPoint) 与垃圾记录 (GarbageRecord) 之间存在一对多的联系。这种关系反映了一个监测点可以在多个时间节点生成多条垃圾记录，而每条垃圾记录仅对应一个特定的监测点。这种关系确保了系统的精确性和数据追溯能力，使得监测点能够持续记录海洋垃圾的分布情况。

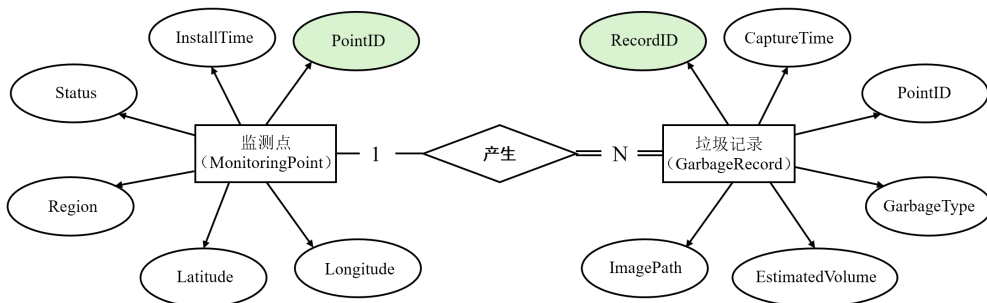


图 2.8 监测点 (MonitoringPoint) 和垃圾记录 (GarbageRecord) 之间的实体联系介绍图

2.2.3 清理任务 (CleanupTask) 和监测点 (MonitoringPoint) 之间的联系 (定位)

在数据库设计中，清理任务 (CleanupTask) 与监测点 (MonitoringPoint) 之间存在多对一的联系。这种关系反映了多个清理任务可能针对同一个监测点的垃圾分布进行处理，而每个清理任务仅指向一个特定的监测点。这种关系确保了系统的任务分配精准性和监测点垃圾清理的集中管理。

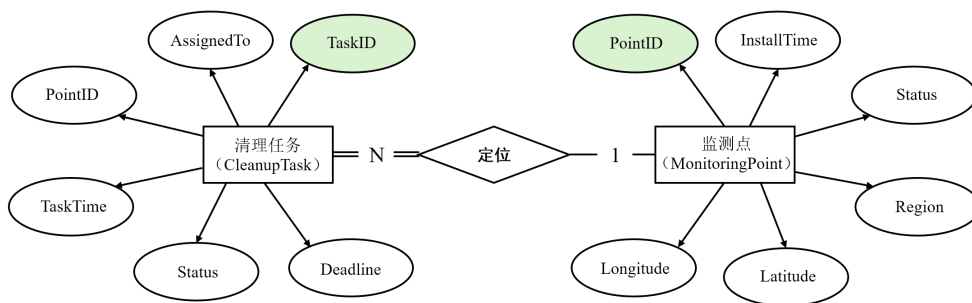


图 2.9 清理任务（CleanupTask）和监测点（MonitoringPoint）之间的实体联系介绍图

2.2.4 清理任务（CleanupTask）清理反馈（CleanupFeedback）之间的联系（有无）

在数据库设计中，清理任务（CleanupTask）与清理反馈（CleanupFeedback）之间存在一对一（或一对零/一）的联系。这种关系反映了每个清理任务在完成后对应一个清理反馈，以记录任务的执行情况，而某些未执行的任务可能尚未生成反馈。这种关系确保了系统对任务完成情况的准确跟踪和验证。

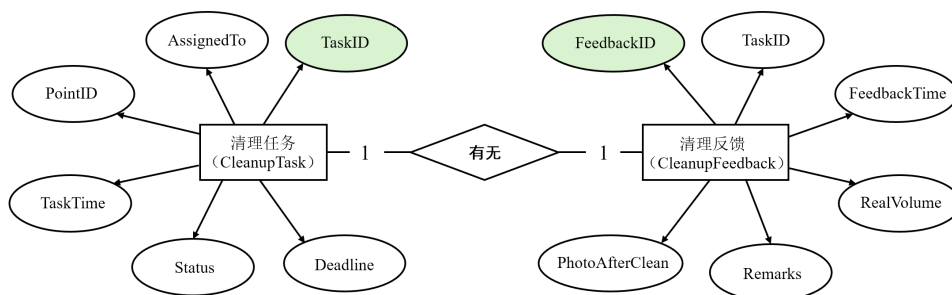


图 2.10 清理任务（CleanupTask）清理反馈（CleanupFeedback）之间的实体联系介绍图

2.2.5 用户（User）和登录日志（LoginLog）之间的联系（登录）

在数据库设计中，用户（User）与登录日志（LoginLog）之间存在一对多的联系。这种关系反映了实际应用中的真实情况，即一个用户可以多次登录系统，每次登录生成一条独立的日志记录。这种关系确保了系统能够准确跟踪用户的登录行为和历史。

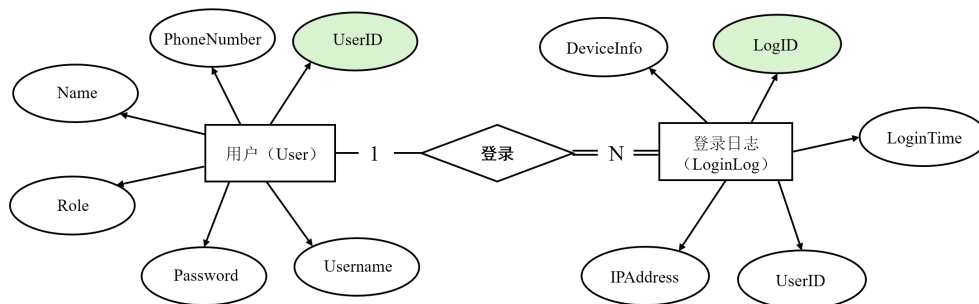


图 2.11 用户（User）和登录日志（LoginLog）之间的实体联系介绍图

2.3 E-R 图设计

在本系统的 E-R 图设计中，共包含六个主要实体：用户（User）、监测点（MonitoringPoint）、垃圾记录（GarbageRecord）、清理任务（CleanupTask）、清理反馈（CleanupFeedback）与登录日志（LoginLog）。各实体间建立了清晰的联系逻辑：用户可以执行多个清理任务，每个清理任务定位于一个具体的监测点；监测点定期生成垃圾记录，部分垃圾记录触发清理任务；

每个清理任务可对应一条清理反馈，反馈内容记录任务执行情况；同时，用户的每次登录行为都会记录在登录日志中。这一设计完整覆盖了系统中“监测—记录—派发—执行—反馈—审计”的数据闭环，结构清晰、逻辑严谨，为后续数据库逻辑结构设计与系统开发奠定坚实基础。

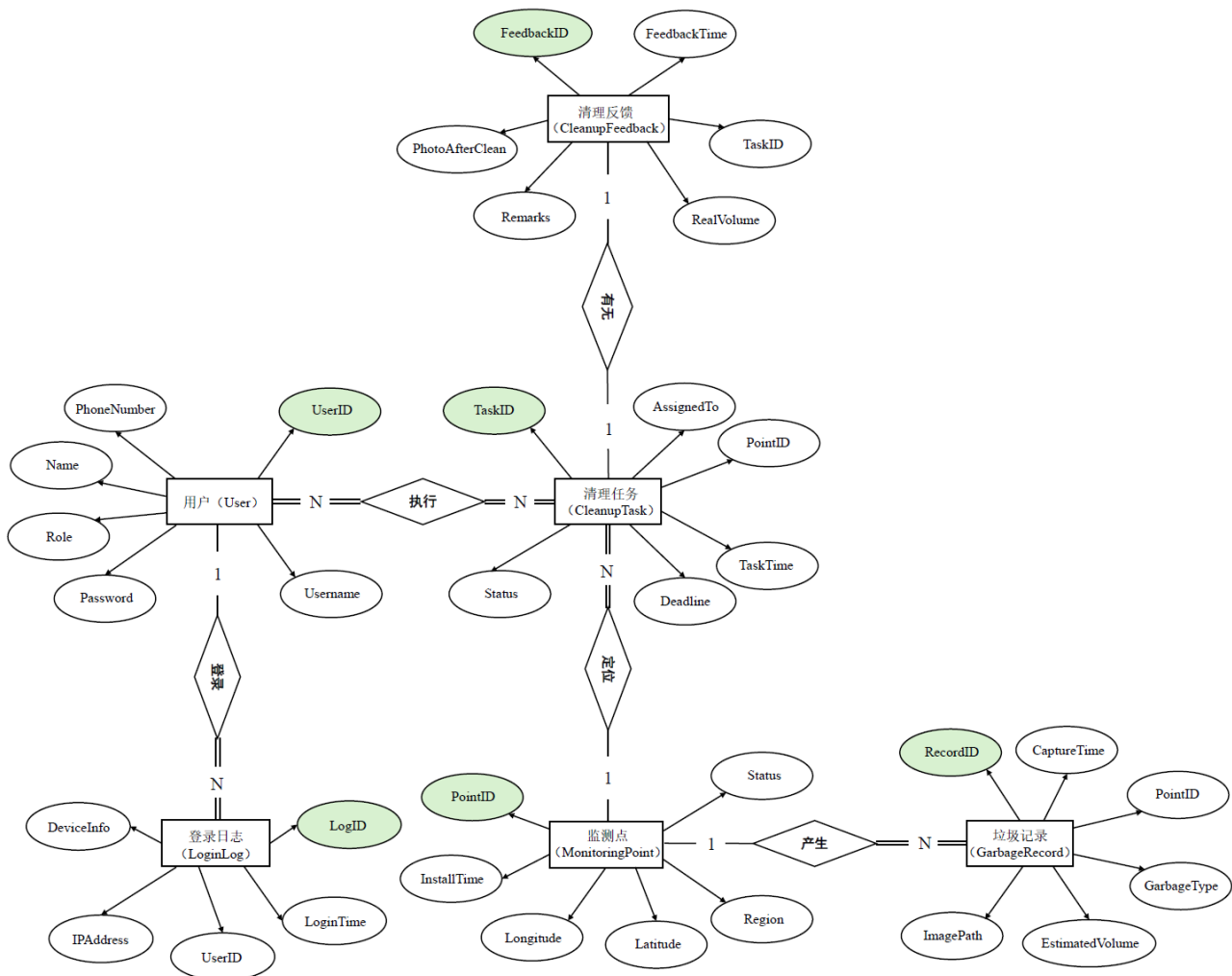


图 2.12 海洋垃圾分布监测与清理规划系统 E-R 图

3 逻辑设计

3.1 实体间联系转换

实体间联系转换可以分为弱实体处理、1 对 1 关系、1 对 n 关系和 n 对 n 关系。

对于**弱实体处理**，表 CleanupFeedback（清理反馈）依赖于 CleanupTask，其主键 FeedbackID 无法独立存在，需通过 TaskID 建立联系。处理方式是将强实体 CleanupTask 的主键 TaskID 作为外键加入 CleanupFeedback 表中，并与 FeedbackID 共同表征反馈记录。

当前模型中未出现严格的**1 对 1 关系**。若未来存在如“用户-个人档案”等 1 对 1 情况，可在任一方加入对方主键作为外键，并引入关系属性。

对于**1 对 n 关系**：

- User 与 LoginLog：一个用户对应多个登录记录（1:n），在 LoginLog 表中添加 UserID 作为外键。
- MonitoringPoint 与 GarbageRecord：一个监测点可关联多个垃圾记录，GarbageRecord 中含 PointID 外键。
- MonitoringPoint 与 CleanupTask：一个监测点可分配多个清理任务，CleanupTask 中含 PointID 外键。
- User 与 CleanupTask：一个用户可被指派多个任务，CleanupTask 中含 AssignedTo 外键。
- CleanupTask 与 CleanupFeedback：一个任务对应一个反馈（特殊 1 对 1 关系，建模上当作 1 对 n），CleanupFeedback 中含 TaskID 外键。

对于**n 对 n 关系**，GarbageRecord 与 CleanupTask，若一条垃圾记录可关联多个清理任务，反之亦然。使用关联表 GarbageTaskLink 解决：包含 RecordID 和 TaskID 两个外键，同时定义独立主键 LinkID，若后续有此联系的属性，也可加入此表中。

3.2 关系模型设计

根据上面的转换，E-R 图可以转换为如下关系模型：

- User(UserID, Username, Password, Role, Name, PhoneNumber)
- MonitoringPoint(PointID, Longitude, Latitude, Region, InstallTime, Status)
- GarbageRecord(RecordID, PointID, CaptureTime, GarbageType, EstimatedVolume, ImagePath)
- CleanupTask(TaskID, AssignedTo, PointID, TaskTime, Deadline, Status)
- CleanupFeedback(FeedbackID, TaskID, FeedbackTime, RealVolume, Remarks,

PhotoAfterClean)

根据前面实体及其联系描述,我们可以为多对多或需要单独管理属性的一些联系设计独立的联系关系表(关系型数据库中的中间表)。系统中大多数关系(如一对多、一对一)已通过外键直接实现,无需额外建表。仅垃圾记录与清理任务的逻辑联系需要中间表,可以带来显式溯源功能。

垃圾记录-清理任务联系关系表 (GarbageTaskLink):

● GarbageTaskLink(LinkID, RecordID, TaskID)

若未来有一条垃圾记录触发多个清理任务(如跨区域分派),此表为多对多关系提供结构支持。

在系统设计中,垃圾记录 (GarbageRecord) 与清理任务 (CleanupTask) 之间存在一对多的隐式逻辑联系,体现为业务上的因果关系。这种关系反映了实际应用中的真实情况,即某些垃圾记录可能触发系统生成一个或多个清理任务(例如根据垃圾量或类型分派任务),但每个清理任务通常关联特定的垃圾记录数据。这种联系无需在数据库中显式建模,而是通过程序逻辑实现绑定,确保系统能够根据垃圾监测结果灵活、高效地生成和管理清理任务。

3.3 第三范式 (3NF) 满足情况

关系数据库设计中,第三范式 (3NF) 的目标是消除传递依赖,确保数据结构简洁、冗余最小。判断数据库设计是否满足第三范式 (3NF),主要是通过三个层次逐步验证字段之间的依赖关系:

首先,数据库表必须满足第一范式 (1NF),即每个字段都是原子值,不可再拆分。例如,在“用户 (User)”表中,PhoneNumber 字段表示单一号码而非多个号码拼接,因此符合 1NF 要求。

其次,表需要满足第二范式 (2NF),这意味着在 1NF 的基础上,表中的所有非主属性必须完全依赖于主键,而不是部分依赖。部分依赖通常出现在组合主键的关系中。当一个非主属性只依赖于主键的一部分时,就违反了第二范式。本系统的各张表普遍采用单一主键(如 UserID、TaskID),避免了组合主键的复杂性,因此不存在部分依赖的问题。比如在“清理任务 (CleanupTask)”表中,AssignedTo、TaskTime 等字段都依赖于唯一标识 TaskID,并未对其主键的一部分产生依赖,因此符合 2NF。

第三范式 (3NF) 是在第一范式和第二范式的基础上提出的更高规范,要求关系模型中每一个非主属性都必须直接依赖于主键,而不能依赖于主键的一部分(即部分依赖),也不能通过其他非主属性间接依赖主键(即传递依赖)。例如,若某表中存在“部门编号→部门名称→员工地址”的依赖链,则员工地址传递依赖于部门编号。本系统中,例如“用户 (User)”表中的 PhoneNumber 和 Name 都是直接由 UserID 决定的,之间没有层级依赖,因此不存在传递依赖。

综合而言，本系统中的主要数据表设计清晰，所有属性都是原子值所有非主属性完全函数依赖主键，不存在非主属性对主键的传递依赖，当前数据库设计符合良好范式结构，满足 3NF 要求，具有高度的数据独立性与一致性。在下文实体表设计中还会逐一说明。

3.4 实体表设计

表 3.1 用户（User）实体表设计

属性名	类型	默认值	描述	约束
UserID	VARCHAR(10) PRIMARY KEY		用户唯一标识	主键、非空
Username	VARCHAR(50)		登录名	非空、唯一
Password	VARCHAR(20)		登录密码	非空
Role	ENUM	操作员	枚举值，用户角色 (如操作员、清理人员、数据分析员)	非空
Name	VARCHAR(50)		姓名	非空
PhoneNumber	CHAR(11)		电话，使用正则表达式检验输入	符合 1 开头 11 位数字的格式，非空

所有非主属性完全依赖 UserID，没有属性传递依赖于主键，满足 3NF。

表 3.2 监测点（MonitoringPoint）实体表设计

属性名	类型	默认值	描述	约束
PointID	VARCHAR(10) PRIMARY KEY		监测点唯一标识	主键、非空
Longitude	DECIMAL(9,6)	0	经度	非空
Latitude	DECIMAL(9,6)	0	纬度	非空
Region	VARCHAR(100)		所属海域或区域名	非空
InstallTime	DATETIME	DATE(current_timestamp)	监测设备安装时间	非空
Status	ENUM	启用	枚举值，状态（启用、故障、维修等）	非空

所有非主属性完全依赖 PointID，没有属性传递依赖于主键，满足 3NF。

表 3.3 垃圾记录（GarbageRecord）实体表设计

属性名	类型	默认值	描述	约束
RecordID	VARCHAR(10) PRIMARY KEY		记录唯一标识	主键、非空、自增
PointID	VARCHAR(10) FOREIGN KEY		对应监测点	外键、非空

续表 3.3

属性名	类型	默认值	描述	约束
CaptureTime	DATETIME	DATE(current_timestamp)	拍摄上传时间	非空
GarbageType	ENUM		枚举值，垃圾类型（塑料、金属、玻璃、其他）	非空
EstimatedVolume	FLOAT		估算体积（立方米）	
ImagePath	VARCHAR(255)		图像路径或链接	

所有非主属性完全依赖 RecordID，外键 PointID 不引起传递依赖,不存在“某字段依赖另一个非主字段再依赖主键”的情况，满足 3NF。

表 3.4 清理任务（CleanupTask）实体表设计

属性名	类型	默认值	描述	约束
TaskID	VARCHAR(10) PRIMARY KEY		任务唯一标识	主键、非空
PointID	VARCHAR(10) FOERIGN KRY		清理目标点	外键、非空
AssignedTo	VARCHAR(10) FOERIGN KRY		清理人员 ID（UserID）	外键、非空
TaskTime	DATETIME	DATE(current_timestamp)	任务发布时间	非空
Deadline	DATETIME	DATE(current_timestamp)	清理截止时间	
Status	ENUM		枚举值，任务状态（待执行、进行中、已完成）	非空

所有字段都是对任务本身的描述，完全依赖于 TaskID，满足 3NF。

表 3.5 清理反馈（CleanupFeedback）实体表设计

属性名	类型	默认值	描述	约束
FeedbackID	VARCHAR(10) PRIMARY KEY		反馈唯一标识	主键、非空
TaskID	VARCHAR(10) FOERIGN KRY		对应清理任务	外键、非空
FeedbackTime	DATETIME	DATE(current_timestamp)	提交时间	非空

续表 3.5

属性名	类型	默认值	描述	约束
RealVolume	FLOAT		实际清理的体积	非空
Remarks	TEXT	无	备注说明	
PhotoAfterClean	VARCHAR(255)		清理后照片路径	

即使 FeedbackID 与 TaskID 可能 1 对 1, 也没有属性依赖 TaskID 而不依赖 FeedbackID, 没有属性传递依赖于主键, 满足 3NF。

表 3.6 登录日志 (LoginLog) 实体表设计

属性名	类型	默认值	描述	约束
LogID	VARCHAR(10) PRIMARY KEY		日志唯一标识	主键、非空、自增
UserID	VARCHAR(10) FOERIGN KRY		登录用户	外键、非空
LoginTime	DATETIME	DATE(current_timestamp)	登录时间	非空
IPAddress	VARCHAR(45)	127.1	登录 IP	非空, 符合 IP 基本格式
DeviceInfo	VARCHAR(255)	PC	登录设备信息	

每条日志是独立的登录事件, 所有字段依赖 LogID, IP 和设备信息与时间无直接依赖, 未形成传递依赖, 满足 3NF。

3.5 关系表设计

表 3.7 垃圾记录-清理任务联系 (GarbageTaskLink) 关系表设计

属性名	类型	默认值	描述	约束
LinkID	VARCHAR(10) PRIMARY KEY		每条关联的唯一标识	主键、非空、自增
RecordID	VARCHAR(10) FOERIGN KRY		垃圾记录 ID	非空、唯一
TaskID	VARCHAR(10) FOERIGN KRY		清理任务 ID	非空

所有字段原子, 完全依赖主键 LinkID, 无传递依赖, 主键设计清晰, 满足 3NF。

建立此表有两种思路, 第一是使用 LinkID 作为主键, 也是比较推荐的方法, 同时也可以使用(RecordID, TaskID)作为联合主键。

4 物理设计

在完成关系模型设计并满足第三范式（3NF）后，接下来的物理设计阶段主要关注如何将逻辑模型高效映射到数据库管理系统中。物理设计旨在提高性能、可扩展性和可维护性，涉及表结构的索引策略、数据类型选择、存储优化等。

4.1 索引设计

在海洋垃圾监测系统设计中，总体原则要求每个表默认设置主键索引（PRIMARY KEY），并建议为外键字段（如 PointID、UserID）添加索引以优化 JOIN 操作，同时对查询频繁的字段（如 Username、Task.Status）单独建立索引。此外，组合常用的过滤条件（如 UserID 和 LoginTime）可创建联合索引，以提升查询效率，但需避免重复冗余索引以降低维护成本。

表 4.1 各表索引设计

表名	字段	索引类型	用途描述
User	Username	UNIQUE INDEX	用户登录识别
	UserID	PRIMARY KEY	主键
MonitoringPoint	PointID	PRIMARY KEY	主键
	(Latitude, Longitude)	COMPOSITE INDEX	地理位置范围查询
GarbageRecord	RecordID	PRIMARY KEY	主键
	PointID	INDEX	查询某监测点的垃圾记录
	CaptureTime	INDEX	时间范围查询（如查询近一个月）
	(PointID, CaptureTime)	COMPOSITE INDEX	查询某点一段时间的垃圾数据
CleanupTask	TaskID	PRIMARY KEY	主键
	AssignedTo	INDEX	查询某用户的任务
	PointID	INDEX	查询监测点的任务
	Status	INDEX	按任务状态分类
	(AssignedTo, Status)	COMPOSITE INDEX	查询某用户当前状态任务
CleanupFeedback	FeedbackID	PRIMARY KEY	主键
	TaskID	INDEX	查询任务的反馈

续表 4.1

表名	字段	索引类型	用途描述
LoginLog	LogID	PRIMARY KEY	主键
	UserID	INDEX	用户审计
	LoginTime	INDEX	最近登录记录排序
	(UserID, LoginTime)	COMPOSITE INDEX	查询某用户的登录历史，按时间倒序
GarbageTaskLink	LinkID	PRIMARY KEY	主键
	RecordID	INDEX	查找某条垃圾记录是否被指派
	TaskID	INDEX	查找某任务关联了哪些垃圾
	(RecordID, TaskID)	COMPOSITE INDEX	防止重复分配

4.2 存储引擎与编码

海洋垃圾监测系统的存储引擎与编码建议采用 MySQL、InnoDB 或 PostgreSQL 存储引擎以支持事务和外键功能，字符编码使用 UTF8MB4 以兼容 Emoji 和多语言环境，同时实施每日快照和关键表（如 LoginLog）日志归档的自动备份策略，以确保数据安全和可恢复性。

所有 ENUM 类型可根据数据库选择具体实现（如 MySQL 支持 ENUM，PostgreSQL 可用 CHECK+TEXT 或 ENUM 类型）。

4.3 安全与优化

在海洋垃圾监测系统中，为避免数据库因大量图像数据而产生性能瓶颈，系统采用对象存储服务（OSS）或内容分发网络（CDN）存储所有外部图像资源，如垃圾检测图、清理反馈照片等，仅在数据库中保留图片的访问 URL。这种设计既节省了数据库存储空间，又提高了图片访问速度，特别适合海量图像上传与展示的业务需求。

同时，系统对 LoginLog 登录日志表采取定期归档与清理策略，以控制表规模、防止性能下降。具体做法示例：保留近 6 个月的数据，历史数据通过自动任务导出为 CSV 或存入归档表后从主表中删除。此外，可采用按月分表方式优化日志查询效率，提升数据管理灵活性和系统响应速度。

为了提升查询效率与维护便利性，系统在数据库中预设了常用视图与存储过程。例如，定义用户任务视图（UserTaskView）整合了用户信息与其被分配的清理任务详情，方便管理端快速查阅人员任务分布；定义存储过程（如 AssignCleanupTask）封装任务指派逻辑，通过后端调用统一接口自动插入任务，避免业务逻辑在多个模块重复实现，提高一致性。