

# 城市卫生管理系统

数据库系统原理课程大作业报告



学 号 2351753

姓 名 黄保翔

专 业 计算机科学与技术专业

授课老师 李文根

# 目录

一. 概述.....	4
1.1 课题背景.....	4
1.2 课题目标.....	4
二.需求分析.....	4
2.1 用户需求分析.....	4
2.2 系统功能模块.....	5
三.可行性分析.....	6
3.1 技术可行性.....	6
3.2 应用可行性.....	6
四.概念设计.....	6
4.1 实体.....	6
4.2 实体属性局部 E-R 图.....	9
4.3 全局 E-R 图 .....	13
五.逻辑设计.....	13
5.1 E-R 图向关系模型的转变.....	13
5.2 数据模型的优化及规范化设计 .....	15
5.3 表结构设计 .....	15
六、物理设计（索引设计）.....	18
6.1 分析查询条件 .....	18
6.2 分析选择条件 .....	18
6.3 联合索引设计 .....	18
6.4 索引表统计 .....	19
八.总结 .....	20

**摘要：**近年来，随着城市化进程的不断加快，城市环境卫生问题日益受到社会各界的广泛关注。城市清洁工作的复杂性和多样性不断提升，传统依靠人工记录与纸质表格的环卫管理方式已难以适应现代城市的精细化管理需求。在此背景下，构建一套高效、智能的城市卫生管理系统具有重要现实意义。

本系统以实际城市环卫业务为原型，设计并实现了一套完整的数据库模型与功能框架，旨在为城市管理者、一线环卫工人以及居民用户提供统一的数据平台。系统支持任务派发与反馈、公共设施管理、投诉处理、异常预警、数据分析与服务评价等功能模块，用户可实现如：查看工人每日任务安排、设施状态信息上报、居民在线投诉及进度跟踪、清洁服务满意度反馈等操作，有效提升了城市卫生事务的处理效率与透明度。

本报告作为系统设计的最终成果文档，内容涵盖系统背景与意义、可行性分析、E-R 模型构建、关系模式转换、关键索引设计与数据流图绘制，最终通过图示展示与文字说明相结合的方式，完整呈现本城市卫生管理系统数据库的构建过程及成果，为后续系统开发与实际部署提供了坚实的数据基础。

**关键词：**数据库；城市卫生；城市管理

# 一.概述

## 1.1 课题背景

随着城市化进程不断推进，城市人口密度不断上升，城市卫生状况对居民健康和生活质量的影响日益显著。垃圾堆积、厕所卫生、街道清洁等问题已成为城市治理的重要内容。为了有效提升城市卫生管理效率，实现科学化、精细化、智能化管理，有必要建立一套完善的城市卫生管理数据库系统。

该系统可以对城市中的环卫人员、垃圾收集站点、公厕、清扫任务、监督投诉等信息进行集中管理，为城市管理者提供实时、准确、系统化的管理依据，提升城市整体卫生水平。

## 1.2 课题目标

本项目基于数据库系统原理，设计一个“城市卫生管理系统”数据库，采用关系数据库模型，支持城市环境数据的采集、存储、管理和查询，面向环卫管理部门、环卫工人、居民投诉者等多个用户角色，提供环卫任务调度、设施管理、投诉追踪、异常预警等核心功能。

# 二.需求分析

## 2.1 用户需求分析

### 2.1.1 管理人员

管理人员通常隶属于城市管理局或城管执法机构，是城市卫生治理的组织者和指挥者。他们对城市整体卫生状况的掌握、任务的指派和投诉问题的处理具有核心职责，主要需求包括：

#### 1. 城市卫生情况总览：

系统应提供基于各区域数据的可视化展示，例如各区域清扫完成率、垃圾收集频次、异常预警分布图等，便于管理人员实时掌握卫生情况，进行科学决策。

#### 2. 任务调度与资源分配：

管理人员需根据区域任务量、设施分布及人员排班等因素，灵活分配每日的清洁任务，并调用相应的人员与工具资源，实现动态任务管理与优化。

#### 3. 投诉审核与处置安排：

开放市民监督申诉系统。系统需支持投诉的全流程管理，包括投诉内容的审核、责任人的指定、处理时限设定、跟踪反馈等，以提升居民参与城市治理的满意度和互动性。

### 2.1.2 环卫工人

环卫工人是执行城市卫生工作的基础力量。他们的任务明确性、操作简便性与上报便利性直接影响城市清洁效率。系统需满足以下需求：

#### 1. 每日任务查询：

系统提供个人任务列表，标明清洁区域、任务类型、开始时间、截止时间、工具要

求等，提升执行工作的清晰度。

### 2. 任务进度与问题上报：

支持环卫工人通过移动端实时上传任务完成情况及异常区域，如非法堆放垃圾、设施损坏等，便于后台管理及时处理

### 3. 任务调整接收：

在遇到突发事件或异常情况时，管理人员可临时更改任务或指派新任务，系统需即时向相关工人发送通知，确保任务执行连续性。

## 2.2.3 居民用户

城市卫生管理系统也应面向广大市民开放，鼓励居民参与城市卫生监督与反馈，提高城市治理透明度与响应速度。居民用户主要功能需求包括：

### 1. 在线投诉提交

居民可以通过平台上传投诉信息，包括位置、时间、问题类型、现场照片等，用于反馈垃圾堆积、厕所脏乱、公厕无纸等问题。

### 2. 投诉进度跟踪

系统应为每一位投诉人提供投诉处理的状态查询功能，包括是否立案、是否处理、处理人及反馈信息等

### 3. 服务反馈评价

在投诉处理或清洁任务完成后，用户可以进行满意度评分与意见反馈，用于城市服务质量的持续改进。

## 2.2 系统功能模块

根据上述用户角色和业务需求，城市卫生管理系统设计如下主要功能模块

### 2.2.1 环卫任务管理系统

用于管理每日的清扫任务派发、执行跟踪和结果记录。支持根据区域、设施、人员空闲程度自动生成任务计划，并支持人工调整。任务信息包含任务编号、任务区域、分配员工、所需工具、截止时间及当前状态。

### 2.2.2 公共设施管理系统

对城市中的垃圾回收站、公厕、环卫车辆、工具设备等设施进行分类管理。包括设施位置标注、状态记录、维保记录等功能。便于管理人员掌握设施分布与运行状态，并辅助决策优化。

### 2.2.3 投诉处理系统

面向市民用户，支持上传投诉信息并由管理人员进行审核、派发、处理与结果反馈。系统建立投诉处理流程状态（如：已接收、处理中、已完成、已反馈等），提升投诉处理效率和居民满意度。

### 2.2.4 异常预警系统

依托传感器数据（如垃圾桶溢满检测、公厕人流量异常、环境异味监测等）或者通过环卫工人以及居民上报机制，触发异常警报。系统支持多级预警信息发布，并对接任务调度模块，快速响应和处理。

### 2.2.5 数据分析与可视化模块（下学期实现）

将系统中积累的数据进行可视化呈现与统计分析。例如：区域任务完成率。任务频度情况，投诉地区热力图，环卫工人任务负载分析，公共设置使用频率和异常状况频率热力图。通过为管理层提供数据支撑，提升管理水平，并且进一步完善城市卫生系统。

## 三. 可行性分析

### 3.1 技术可行性

本系统采用成熟的 Web 架构设计方案，由于数据的多样性与复杂性，并且还需要同时向 3 种不同身份的用户（环卫工人，管理人员，居民）进行服务。可以采用 MySQL 或 PostgreSQL 关系型数据库，具备事务支持、查询优化能力强，适合结构化数据管理。后端框架使用 Django 具有开发效率高、社区支持广的特点，前端使用 Vue 或 React 构建数据交互页面，配合 ECharts、Leaflet 等组件展示可视化信息。

目前已有大量类似平台（如智慧城市管、数字城建）采用类似技术栈，说明方案具备高度可行性。

### 3.2 应用可行性

本系统在开发初期可以采用开源数据库和框架，部署于本地或小型云服务器中，成本可控。数据主要来源于管理部门人工录入、居民上传、IoT 设备对接，前期数据量较小，存储与处理开销低，后续可根据需求接入物联网平台和更多传感器，分阶段进行扩容与部署，使用 OpenCV 等技术处理图片上报信息，增强自动识别能力。系统具有显著的节省人力、提高效率优势，长期运营成本相对较低，具有较高投入产出比。

城市卫生管理系统可与现有智慧城市、城市大脑等平台对接，实现数据共享与互联互通，例如可以嵌入城管信息平台，实现统一调度与数据归档，可通过小程序、公众号等方式快速接入居民终端，提高用户参与度；

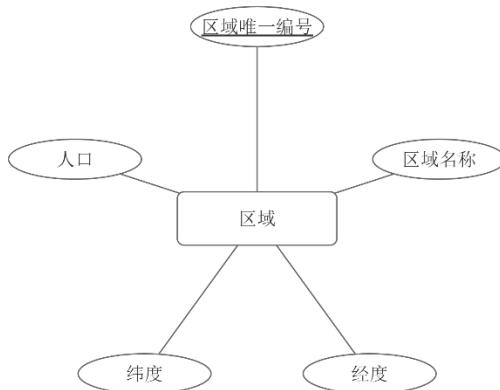
系统具有良好的扩展性与模块化特性，适应不同规模城市运行需求。从运营角度看，系统具有可持续运行的能力，便于推广和复制到更多城市。

## 四. 概念设计

### 4.1 实体

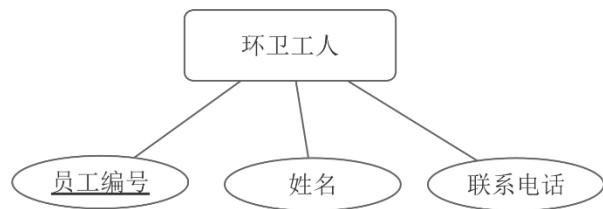
#### 4.1.1 区域设计

用于对城市管理进行区分，便于明确权责，方便管理人员对任务进行分工。每个区域有唯一编号、名称、地理坐标（经度和纬度）以及常住人口数量。区域是系统中其他实体（如环卫工人、设施、居民等）的基础划分单位。



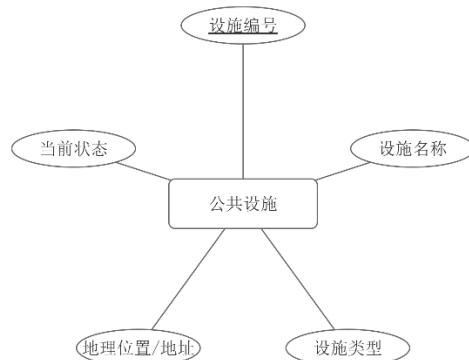
#### 4.1.2 环卫工人信息

表存储从事城市清洁工作的环卫工人信息，包括姓名、联系电话和所属区域。环卫工人负责执行清扫任务，与区域和任务表相关联。



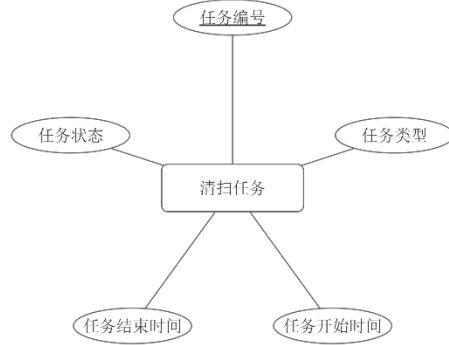
#### 4.1.3 公共设施

存储城市中涉及清洁管理的设施信息，如垃圾站、公厕等，记录设施名称、类型、地理位置、状态和所属区域。设施是清扫任务和异常预警的核心对象。



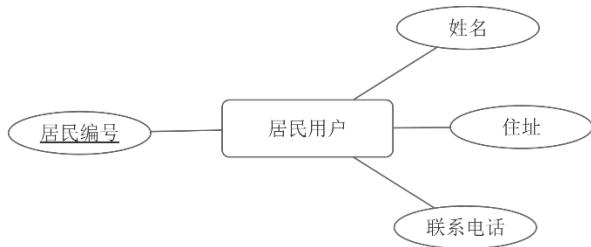
#### 4.1.4 清扫任务

存储指派给环卫工人的清洁任务信息，包括任务类型、起止时间、状态、关联设施和负责员工。任务是系统核心功能，连接设施和环卫工人。



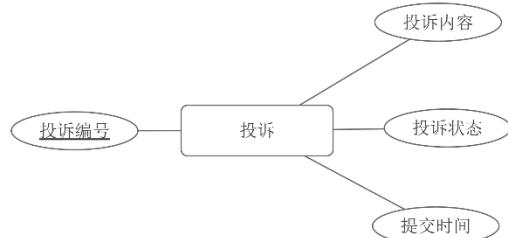
#### 4.1.5 居民用户

存储系统中注册或使用服务的居民用户信息，包括姓名、联系电话、住址和所属区域。居民可以发起投诉或提供服务反馈。



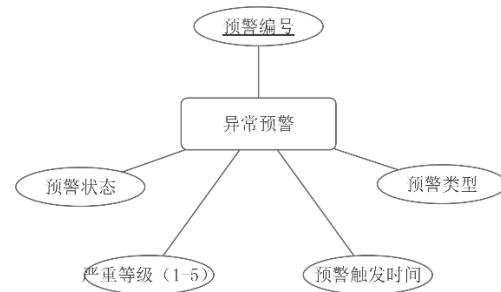
#### 4.1.6 投诉信息

存储居民对设施或服务的投诉信息，记录投诉人、被投诉设施、内容、提交时间、状态和处理反馈，关联任务以跟踪处理情况。



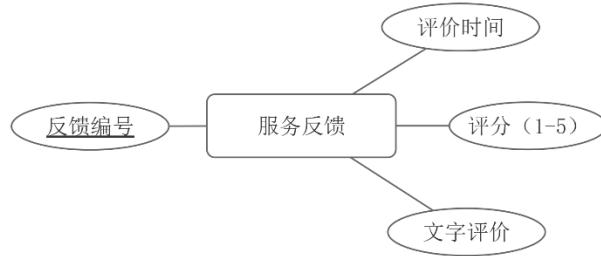
#### 4.1.7 预警信息

存储传感器或人工触发的设施异常预警信息，如垃圾溢出或设备损坏，记录预警类型（例如垃圾溢出，设备损坏）、时间、严重等级和状态（未完成，进行中，已完成）。



#### 4.1.8 服务反馈

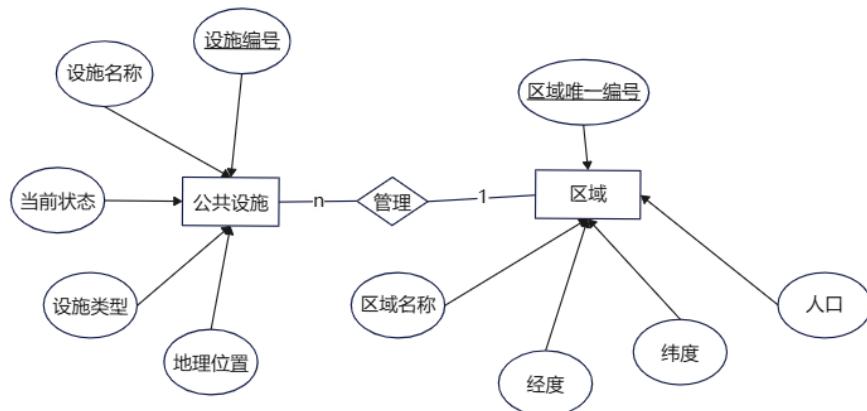
居民对清扫任务的满意度评价，记录评分和文字评价，用于评估环卫工作质量。



#### 4.2 实体属性局部 E-R 图

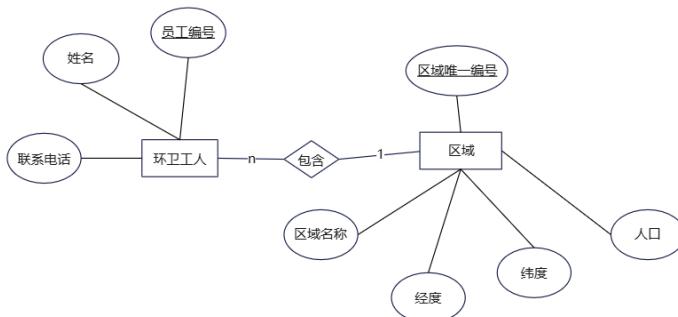
##### 4.2.1 区域 – 公共设施：管理 (1:N)

每个区域可以拥有多个公共设施，而每个公共设施只归属于一个区域。模拟正常的管理结构。



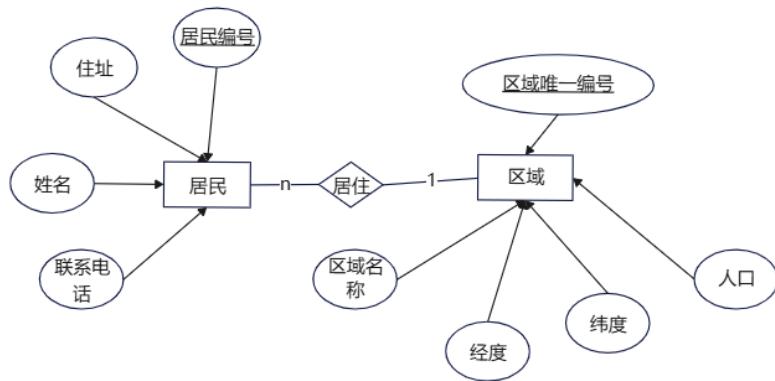
##### 4.2.2 区域 – 环卫工人：包含 (1: N)

一个区域可以包含多位环卫工人，每位工人只属于一个固定区域，区域是清洁任务调度和责任划分的基本单位，工人按区域分配任务。



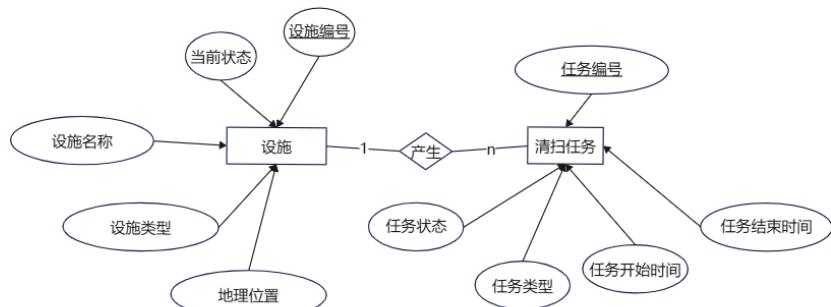
#### 4.2.3 区域 \_ 居民关系图: 居住 (1: N)

每位居民用户归属一个区域，一个区域可包含多个居民。居民的投诉、反馈等功能均依托其所属的行政区域。通过 居住于 联系建立逻辑关系，便于区域化服务统计与预警推送。



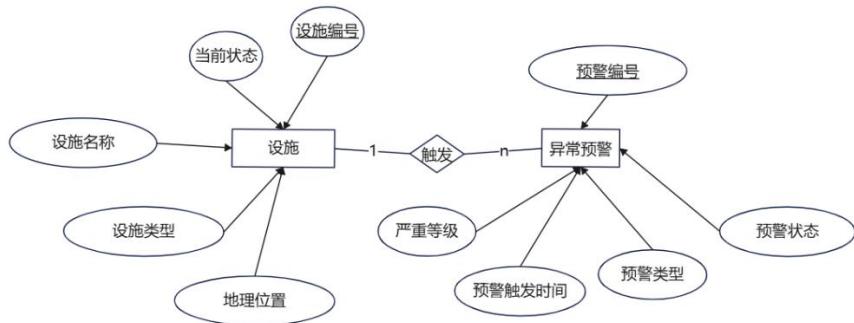
#### 4.2.4 设施 \_ 任务关系图: 产生 (1: N)

一个设施可对应多个清洁任务（如定期清扫、消毒等），每个任务服务于一个设施。任务派发以设施为单位，设施类型决定任务频率与性质。



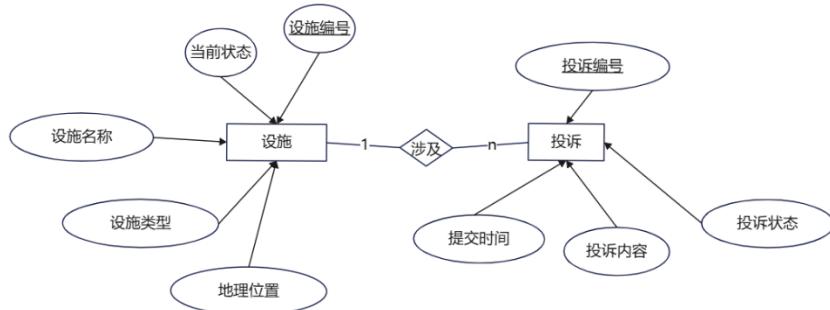
#### 4.2.5 设施 \_ 预警关系图: 触发 (1: N)

设施可能触发多条异常预警（如溢满、损坏等），每条预警对应一个设施。预警用于自动/人工监测设施运行状态，为任务派发提供依据。



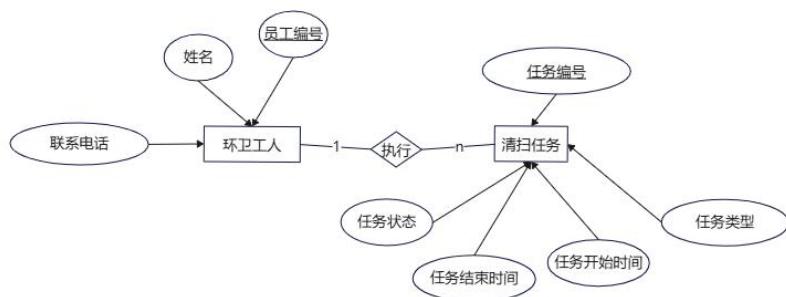
#### 4.2.6 设施 \_ 投诉关系图: 涉及 (1: N)

多个投诉可能指向同一个设施，每条投诉只能涉及一个设施。居民对设施的故障、环境问题进行投诉，便于问题定位。



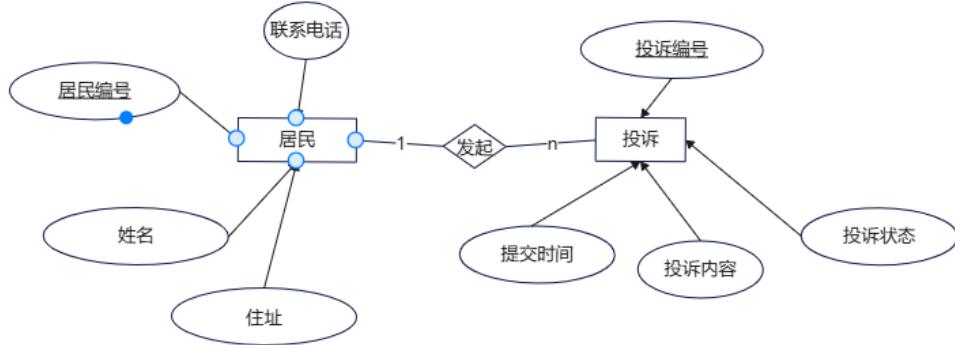
#### 4.2.7 环卫工人 \_ 清扫任务: 执行 (1: N)

一个员工可执行多个清扫任务，每个任务由一位员工负责。任务派发按人分配，记录责任归属与考核数据。



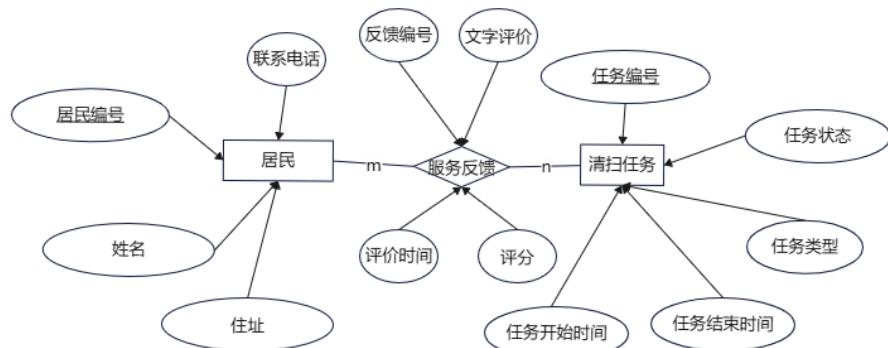
#### 4.2.8 居民 \_ 投诉反馈: 发起 (1: N)

一位居民可以提交多条投诉，每条投诉由一位居民发起。居民是投诉的第一责任方，投诉记录需保留居民标识。

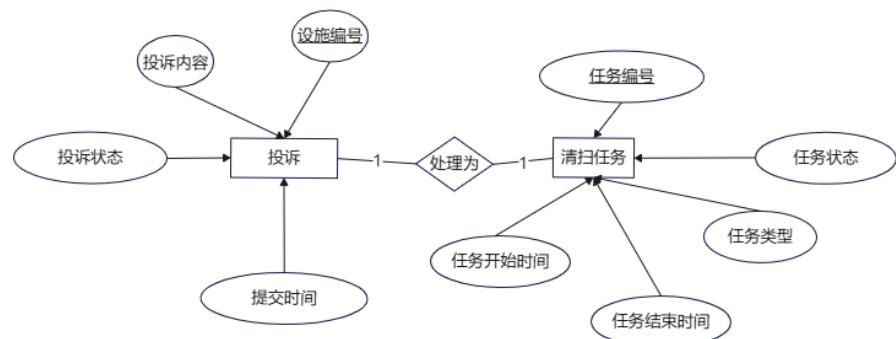


#### 4.2.9 居民 \_ 清扫任务: 服务反馈 (M: N)

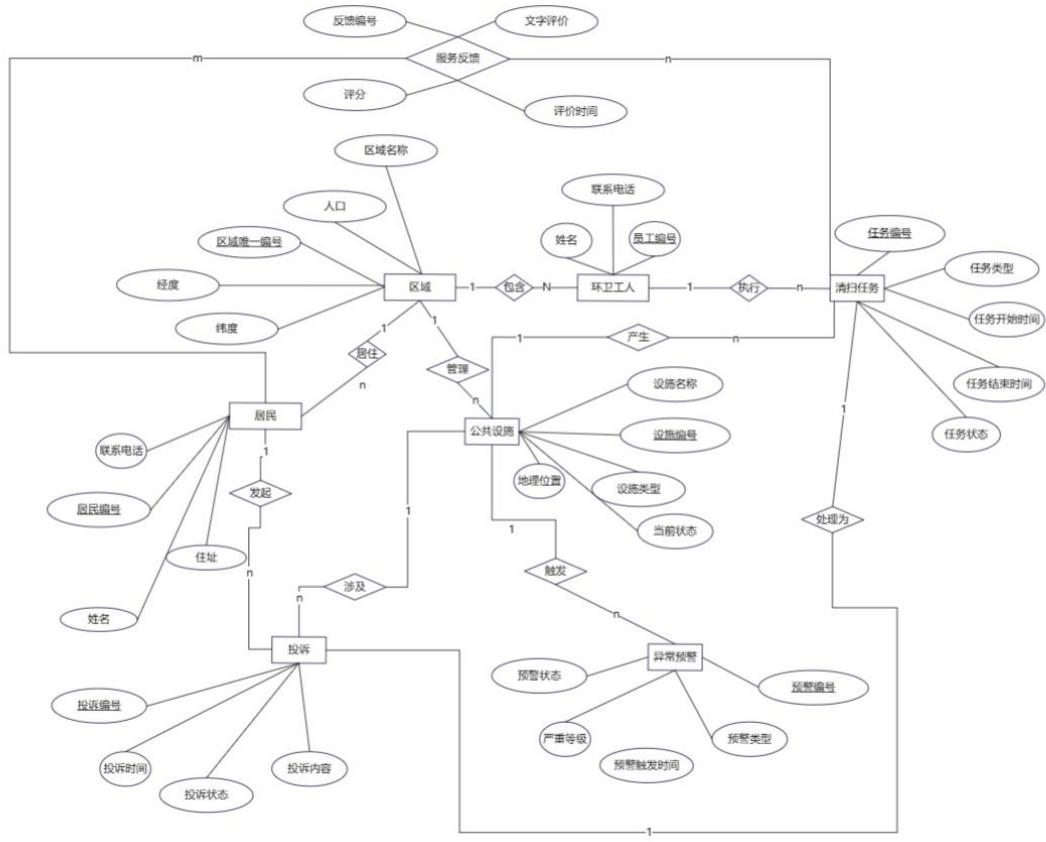
居民对清扫任务进行服务反馈，每个居民可以对多个任务进行反馈，同时，任务也可以由多个居民进行反馈，任务反馈的关系信息包括编号，文字评价，评价时间，评分。



#### 4.2.10 投诉 \_ 清理任务: 处理为 (1: 1)



### 4.3 全局 E-R 图



## 五.逻辑设计

### 5.1 E-R 图向关系模型的转变

将 ER 图中的关系转为实体，同时还要满足 3NF 范式

1:1 关系转换：在 1:1 关系中，可以选择任一端的实体，在其关系模式中添加另一端实体的主键作为外键。

1:N 关系转换：在 1:N 关系中，通常在 N 端实体的关系模式中添加 1 端实体的主键作为外键。

N:M 关系转换：对于 N:M 关系，需要创建一个新的关系模式来表示两个实体之间的联系，该关系模式包含两个实体的主键作为外键，并将这两个外键作为新关系模式的复合主键。

### 区域实体

Region(region\_id, name, longitude, latitude, population);

### 环卫工人实体

由于与区域构成 1: N 关系，添加区域主键作为属性

Employee(emp\_id, name, phone, region\_id)

其中 region\_id 为外键

## 设施实体

由于与区域构成 1: N 关系，添加区域主键作为属性

Facility(facility\_id, name, type, location, status, region\_id)、

其中 region\_id 为外键

## 任务实体

任务与环卫工人的关系是 N:1，与公共设施的关系是 N:1，因此将公共设施与环卫工人的主键添加到 task 当中，用 assign\_to 代表环卫工人的主键，表示将任务分配给了哪一个环卫工人

Task(task\_id, task\_type, start\_time, end\_time, status, facility\_id, assigned\_to)

## 居民用户实体

居民与区域形成 N:1 的关系，因此将区域的主键加入到居民的属性中。

Citizen(citizen\_id, name, phone, address, region\_id)

## 投诉实体

投诉与居民是 N:1 关系，与公共设施是 N:1 关系，与清扫任务是 1:N 关系，因此将三个主键加入到投诉的属性当中

Complaint(complaint\_id, citizen\_id, facility\_id, content, submit\_time, status, feedback, task\_id)

## 预警实体

预警与设施是 N:1 关系，因此将设施的主键加入到预警的属性当中

Alert(alert\_id, facility\_id, alert\_type, alert\_time, severity, status)

居民和清扫任务之间的关系是 M:N 联系，因此将联系本身转为关系模式，属性为两端实体居民和清扫任务的主键以及联系自身的属性

Feedback(feedback\_id, citizen\_id, task\_id, score, comments, time)

## 5.2 数据模型的优化及规范化设计

### 3NF 范式分析

所有非主属性直接依赖于主键；

不存在传递依赖（例如 region\_name 不依赖于 emp\_id）；

外键用于关联，无冗余属性重复存储；

多值属性（如评价）被建为单独实体，保持原子性。NF

分析上述内容可知，符合 3NF 范式

## 5.3 表结构设计

表 1 区域信息

字段名	字段类型	长度	允许空	主键	默认值	说明
region_id	INT	-	否	是	-	区域编号
name	VARCHAR	100	否	否	-	区域名称
longitude	DECIMAL	-	否	否	0.0	经度
latitude	DECIMAL	-	否	否	0.0	纬度
population	INT	-	否	否	0	人口数量

表 2 环卫工人

字段名	字段类型	长度	允许空	主键	默认值	说明
emp_id	INT	-	否	是	-	员工编号
name	VARCHAR	50	否	否	-	姓名
phone	VARCHAR	20	否	否	-	联系电话
region_id	INT	-	否	否	-	所属区域编号

表 3 设施信息

字段名	字段类型	长度	允许空	主键	默认值	说明
facility_id	INT	-	否	是	-	设施编号
name	VARCHAR	100	否	否	-	设施名称
type	VARCHAR	50	否	否	-	设施类型
location	VARCHAR	200	否	否	-	设施位置
status	VARCHAR	50	否	否	-	设施状态
region_id	INT	-	否	否	-	所属区域编号

表 4 任务信息

字段名	字段类型	长度	允许空	主键	默认值	说明
task_id	INT	-	否	是	-	任务编号
task_type	VARCHAR	50	否	否	-	任务类型
start_time	DATETIME	-	否	否	-	开始时间
end_time	DATETIME	-	是	否	NULL	结束时间
status	VARCHAR	50	否	否	-	任务状态
facility_id	INT	-	否	否	-	关联设施编号
assigned_to	INT	-	否	否	-	分配给的员工编号

表 5 市民信息

字段名	字段类型	长度	允许空	主键	默认值	说明
citizen_id	INT	-	否	是	-	市民编号
name	VARCHAR	50	否	否	-	姓名
phone	VARCHAR	20	否	否	-	联系电话
address	VARCHAR	200	否	否	-	地址
region_id	INT	-	否	否	-	所属区域编号

表 6 投诉信息

字段名	字段类型	长度	允许空	主键	默认值	说明
complaint_id	INT	-	否	是	-	投诉编号
citizen_id	INT	-	否	否	-	投诉人编号
facility_id	INT	-	否	否	-	关联设施编号
content	TEXT	-	否	否	-	投诉内容
submit_time	DATETIME	-	否	否	-	提交时间
status	VARCHAR	50	否	否	-	投诉状态
feedback	TEXT	-	是	否	NULL	处理反馈
task_id	INT	-	是	否	NULL	关联任务编号

表 7 告警信息

字段名	字段类型	长度	允许空	主键	默认值	说明
alert_id	INT	-	否	是	-	告警编号
facility_id	INT	-	否	否	-	关联设施编号
alert_type	VARCHAR	50	否	否	-	告警类型
alert_time	DATETIME	-	否	否	-	告警时间
severity	VARCHAR	50	否	否	-	告警严重程度
status	VARCHAR	50	否	否	-	告警状态

表 8 反馈信息

字段名	字段类型	长度	允许空	主键	默认值	说明
feedback_id	INT	-	否	是	-	反馈编号
citizen_id	INT	-	否	否	-	反馈人编号
task_id	INT	-	否	否	-	关联任务编号
score	INT	-	否	否	0	评分
comments	TEXT	-	是	否	NULL	评论

time	DATETIME	-	否	-	反馈时间
------	----------	---	---	---	------

## 六、物理设计（索引设计）

为了提升城市卫生管理系统数据库的查询性能，我们从实际业务功能出发，分析常见的查询条件与查询模式，识别出高频字段与高选择性字段，针对性地设计单列索引与联合索引。同时参考“最左前缀”原则，对多个查询字段进行联合索引优化，避免频繁回表操作，从而提高系统整体响应速度与吞吐量。

### 6.1 分析查询条件

分析在实际使用过程中哪些字段常被用于查询过滤（WHERE）、连接（JOIN）、分组（GROUP BY）或排序（ORDER BY）操作。

在城市卫生管理系统中，常见的查询条件包括：

1. 管理员按区域（region\_id）查询某区域内的工人、居民、设施
2. 工人查询自己所负责的所有任务（通过 assigned\_to）
3. 系统根据任务状态（status）筛选“未开始”、“进行中”、“已完成”任务
4. 根据设施编号（facility\_id）查询设施历史任务、异常预警或居民投诉
5. 管理员根据投诉状态（status）筛选待处理投诉列表
6. 居民评价某任务后，根据 task\_id 查看反馈记录

因此，需要对以下字段建立单列索引：region\_id、assigned\_to、facility\_id、status、task\_id 等。此外，可以结合系统未来可能扩展的功能进行预判性索引设置。例如考虑“突发卫生事件”实时预警功能，可能需要快速定位某区域内居民，因此应在 Citizen 表中的 region\_id 字段上建立索引。

### 6.2 分析选择条件

选择**选择性高**（即唯一值较多）的字段，这样的索引过滤效率更高。同时优先考虑建立唯一索引（UNIQUE）以避免数据重复。例如：

emp\_id（工人编号）、citizen\_id（居民编号）、task\_id（任务编号）均为主键字段，系统默认建立唯一索引

phone（居民电话）是具有唯一性的字段，也适合建立唯一索引

而 status（如任务状态：“进行中”、“已完成”等）值域小、选择性低，不应单独建索引，适合放入联合索引的后部。

综上，对于居民、员工等实体，优先对主键编号建立主键索引，同时对 phone 建立唯一索引，对 region\_id 建立普通索引以便按区域快速查询。

### 6.3 联合索引设计

针对一些实际场景中需要同时使用多个字段进行查询的功能设计联合索引来合并多个条件，

提高索引使用效率，并减少回表操作。

在 Task 表中，经常需要根据 assigned\_to（被分配员工）和 status（任务状态）筛选任务记录。可以设计联合索引。既可以支持单独按 assigned\_to 查询，又可以加速“查某员工未完成任务”的复合查询。

在 Complaint 表中，经常需要根据 facility\_id 和 status 查询某设施的待处理投诉。可设置联合索引。

在 Feedback 表中，需要查询某任务的评分分布，可以设置：

INDEX idx\_task\_score (task\_id, score)

上述联合索引的设计，满足了“最左匹配原则”，能够同时支持以下几类查询：

1. 查询某任务的所有反馈记录
2. 查询某任务中评分低于 3 的反馈记录
3. 查询某工人负责的所有“进行中”任务
4. 查询某设施当前状态为“故障”(即投诉未被处理)的全部投诉

## 6.4 索引表统计

表名	索引字段	类型	描述
Employee	emp_id	主键索引	环卫工人唯一标识索引
	region_id	普通索引	管理员按区域查询员工 (常用 WHERE 条件)
Task	task_id	主键索引	任务唯一标识索引
	(assigned_to, status)	联合索引	工人查询自己进行中的任务 (频繁业务查询)
Facility	facility_id	主键索引	设施唯一标识索引
	region_id	普通索引	管理员按区域查看设施 (清扫调度)
Citizen	citizen_id	主键索引	市民唯一标识 id
	region_id	普通索引	突发卫生危机时快速查询某一区域居民
Complaint	(facility_id, status)	联合索引	查询某设施未处理的投诉
	(citizen_id, status)	联合索引	查询该居民所有卫生未处理的投诉
Feedback	(task_id, score)	联合索引	管理者查看评分低的任务

## 八.总结

本项目围绕城市卫生管理系统的需求，系统性地完成了从需求分析到数据库逻辑结构与物理结构的完整设计流程，力求实现数据结构的合理性、系统功能的完整性以及未来系统扩展的可行性。

在设计初期，从实际业务出发，明确了系统中涉及的主要角色（如管理员、环卫工人、居民用户）及其对应的操作（如任务派发、设施维护、投诉处理、反馈收集等）。在此基础上，提炼出八个核心实体，包括区域、员工、设施、任务、居民、投诉、预警和反馈，作为数据库建模的基础对象。

随后，在概念设计阶段，通过 E-R 图直观地描述了实体与实体之间的联系，如区域与员工之间的“包含”关系、设施与任务之间的“产生”关系、投诉与任务之间的“处理为”关系等，明确了每一类数据的归属和依赖。

在逻辑设计阶段，将概念模型规范化转换为符合第三范式（3NF）的关系模式，合理拆分冗余字段，消除不必要的依赖关系，增强数据一致性。

在物理设计阶段，针对字段类型、主外键约束、索引结构、数据选择性等维度进行优化配置，并分析了索引数量与查询性能的权衡。设计唯一索引，主键索引，并根据属性之间的关联关系设计联合索引。最终完成整个索引的设计表。

综上所述，本次数据库设计充分结合业务背景与数据库建模原理，实现了城市卫生管理系统的整体设计。