数据库及实现课程项目报告

DataSpider 数据库管理系统项目文档

2024–2025 春季学期

小组名称：第 4 组

组员名单：陈远洋、方昱凯、郭轩岩、王艺涵

指导教师：郑卫国

提交日期： 2025 年 6 月 21 日

项目代码、数据库脚本及详细文档已托管于 GitHub，便于团队协作和后续维护：

https://github.com/HanLoyce/project

目录

1 系统需求分析

1.1 现实需求

1.1.1 核心需求

- 快速检索：用户需要在短时间内快速检索到目标网站的所有相关信息并储存下来，包括文本、图片、链接等形式
- 精准获取：支持通过关键字或关键标志搜索特定网页信息，而非全部内容，因此需要对目标网站的信息进行筛选

1.1.2 信息类型需求

- 网站基本信息：域名、所属主体、包含网页数等
- 网页内容信息：文字、图片、所属网站等
- 文本/图片信息：内容、所属网页、来源等
- 数据源信息：文字、内容相关的数据源信息

1.2 业务流程

1.2.1 爬虫运行流程

1. 初始化阶段：导入网站 URL，设定爬取页数和时间间隔，启动爬取任务。
2. 状态显示机制：系统实时显示各个爬取任务的爬取状态，利于用户监听。
3. 异常处理机制：实时汇报爬取异常，记录爬取失败的页面信息，支持重试机制。

1.2.2 数据存储流程

1. 数据清洗：非结构化数据格式化处理，内容标准化处理。
2. 分级存储：按照网站-网页-内容架构存储，清晰易查看。
3. 数据独立：不同账户间已爬取内容相互不可见，保证账户隐私性。

1.2.3 用户使用流程

1. 参数设定：用户从系统提供的网站列表中选择目标网站, 设置检索条件，开始爬取。
2. 数据展示：系统从数据库检索匹配数据，以清晰结构分级展示结果。

3. 前端交互：支持关键词搜索相关网站、网页、数据内容，支持跳转目标网址。
4. 数据反馈：检索成功：结构化处理后保存在本地数据库中; 检索失败：提供友好提示。

1.3 功能需求

1.3.1 用户界面

1. 支持用户创建账号，账号间数据各自独立，账号内数据稳定存储。
2. 提供直观、易操作的用户界面，有充分的安全性提示。
3. 支持网站选择、内容关键字检索、直接网址跳转、数据管理等功能。
4. 显示爬取统计信息（如爬取页数、数据量等）

1.3.2 爬虫模块

1. 实现高效的数据爬取功能，实时显示任务状态。
2. 支持文本内容和图片的抓取，会存储在相关网页下。
3. 自动分类存储不同数据类型，按网站-网页-内容/图片的结构存储
4. 可配置的爬取规则和策略，如爬取网页数，爬取时间间隔

1.3.3 数据管理

1. 实现数据的 CRUD 功能（增加、删除、修改、查询）。
2. 提供后台管理界面，可查看各个用户相关爬取记录。
3. 支持导出已爬取数据，通过命令行实现导出和保存。

1.3.4 检索功能

1. 支持检索文本关键字，包含标题含有的关键字和内容中的高频关键字。
2. 设计高效的索引策略，优化检索性能，确保快速响应。
3. 支持完整查看检索结果，支持直接跳转相关网址。

1.4 数据流图

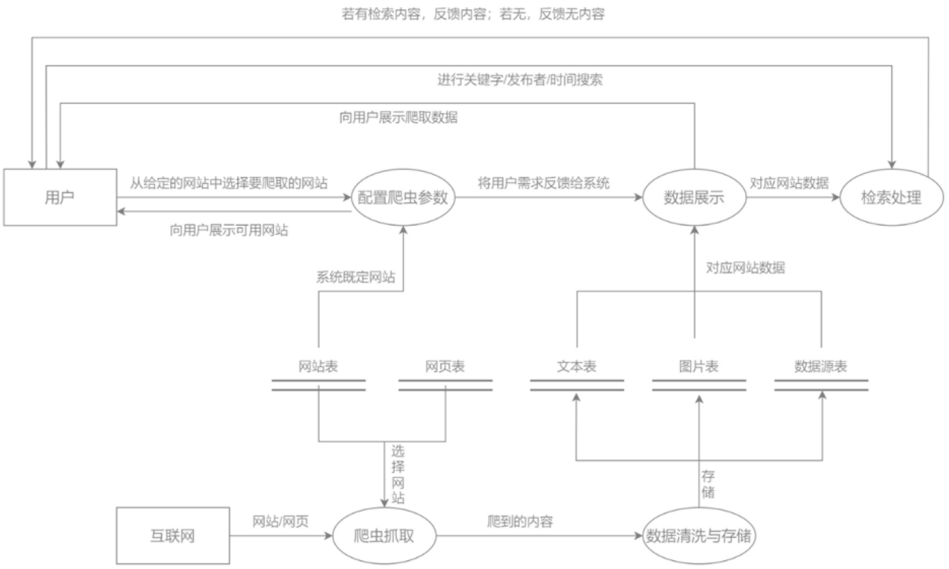

图 1: 数据流图

1.5 实体及属性

1. 网站：(ID，域名，用户归属，标题，描述，首页地址)
2. 网页：(ID，URL，抓取时间，所属网站)
3. 内容：(内容 ID，文本内容，所属网页，关键字，内容类型)
4. 图片：(URL，所属网页，图片描述，分辨率)
5. 数据源：(URL，发布者，发布时间)
6. 爬取任务：(任务 ID，用户 ID，目标网站，爬取状态，开始时间，结束时间，错误信息)

1.6 数据字典

数据名称	说明	类型	长度、取值	别名
网站序号	目标网站的序号	INT	$-2^{31} \sim 2^{31}-1$	id
用户归属	目标对象的用户归属	VARCHAR(255)	255	user
网站域名	目标网站的域名	VARCHAR(255)	255	domain
网站标题	目标网站的标题	VARCHAR(255)	255	title
网站描述	目标网站的描述	TEXT	-	description
首页地址	目标网站的首页地址	VARCHAR(500)	500	homepage
网页序号	目标网页的序号	INT	$-2^{31} \sim 2^{31}-1$	id
网页 URL	每个网页的 URL	VARCHAR(500)	500	url
抓取时间	网页的抓取时间	DATETIME	合理的时间范围	crawl_time
所属网站	网页所属的网站的域名	VARCHAR(255)	255	website
内容 ID	文本信息编号	INT	$-2^{31} \sim 2^{31}-1$	content_id
文本内容	网页的文本内容	TEXT	-	text
所属网页	文本信息所属的网页的 URL	VARCHAR(500)	500	webpage
关键字	文本信息的关键字	VARCHAR(500)	500	keywords
类型	文本信息的类型	ENUM('text','link')	在枚举中选择	type
图片 URL	唯一标识一张图片	VARCHAR(500)	500	url
所属网页	图片所属的网页的 URL	VARCHAR(500)	500	webpage
图片描述	网页中对图片的描述	VARCHAR(255)	255	description
分辨率	图片的分辨率	VARCHAR(50)	50	resolution
数据源 URL	数据来源网页 URL	VARCHAR(500)	500	data_source_url
发布者	数据源的发布者	VARCHAR(100)	100	publisher
发布时间	数据源的发布时间	DATETIME	合理的时间范围	publish_time
爬取任务序号	爬取任务的序号	INT	$-2^{31} \sim 2^{31}-1$	id
爬取状态	爬取任务的进行状态	ENUM('crawling', 'complete', 'fail')	在枚举中选择	status
开始时间	爬取任务中开始爬取的时刻	DATETIME	合理的时间范围	start_time
结束时间	爬取任务中爬取完成的时刻	DATETIME	合理的时间范围	end_time
错误信息	爬取任务异常时的错误信息提示	TEXT	-	error_msg

表 1: 数据项

数据结构名	说明	数据组成
数据源信息表	数据源的相关信息	数据源 URL 发布者信息 发布时间
网页信息表	所爬取网页的相关信息	网页序号 网页 URL 爬取时间 所属网站序号
内容信息表	所爬取内容的相关信息	内容序号 文本内容 关键字 类型 网页序号
网站信息表	所爬取网站的相关信息	网站序号 域名 标题 描述 主页 用户序号
爬取任务信息表	特定爬取任务的信息	序号 爬取状态 开始时间 结束时间 错误信息 用户序号 网站序号
图片信息表	所爬取图片的相关信息	图片 URL 图片描述 分辨率 所属网页序号

表 2: 数据结构

数据流名	说明	数据流来源	数据流去向	数据组成
CRAWL	按设定方式进行爬取操作	爬虫模块	网络数据爬取管理系统	所有实体属性
SEARCH	用户对已有网站的检索	网络数据爬取管理系统	用户	所有实体属性
VIEW	用户查看网站数据	网络数据爬取管理系统	用户	所有实体属性
DELETE	用户请求删除网站数据	用户	网络数据爬取管理系统	所有实体属性

表 3: 数据流

数据储存名	说明	编号	数据流来源	数据流去向	数据组成	数据量
网站信息表	数据库存放、记录网站详细信息	D1	爬虫模块	网络数据爬取管理系统	网站序号 域名 标题 描述 主页 用户序号	1000 个
网页信息表	数据库存放、记录网页详细信息	D2	爬虫模块	网络数据爬取管理系统	网页序号 网页 URL 爬取时间 所属网站序号	10000 个
内容信息表	数据库存放、记录网页文本信息	D3	爬虫模块	网络数据爬取管理系统	内容序号 文本内容 关键字 类型 网页序号	50000 条
图片信息表	数据库存放、记录网页图片信息	D4	爬虫模块	网络数据爬取管理系统	图片 URL 图片描述 分辨率 所属网页序号	20000 张
数据源信息表	数据库存放、记录数据源信息	D5	爬虫模块	网络数据爬取管理系统	数据源 URL 发布者信息 发布时间	20000 个
爬取任务信息表	数据库存放、记录爬取任务信息	D6	爬虫模块	网络数据爬取管理系统	序号 爬取状态 开始时间 结束时间 错误信息 用户序号 网站序号	1000 个

表 4: 数据储存

2 数据库概念模型设计

2.1 实体与联系

1. 网站 ↔ 网页 (1:N) 关联字段：Website. 域名 ↔ Webpage. 所属网站
2. 网页 ↔ 内容 (1:N) 关联字段：Webpage.URL ↔ Content. 所属网页
3. 网页 ↔ 图片 (1:N) 关联字段：Webpage.URL ↔ Image. 所属网页
4. 数据源 ↔ 内容 (1:1) 关联字段：所属 Webpage.URL ↔ Content. 所属网页
5. 数据源 ↔ 图片 (1:1) 关联字段：DataSource. 数据源 URL ↔ Image. 图片 URL

2.2 完整性约束

1. 实体完整性约束：所有主键字段 (见 3.1) 必须为 NOT NULL 且唯一。
2. 参照完整性约束：确保引用一致性，支持级联删除 (如删除网站时自动删除关联网页)。
   - Webpage. 所属网站 → Website. 域名

- DataSource_Content.URL→DataSource.URL
- DataSource_Content. 内容 ID→Content. 内容 ID
- Content. 所属网页 →Webpage.URL
- Image. 所属网页 →Webpage.URL
- Image. 图片 URL→DataSource. 数据源 URL

3. 检查完整性约束: 本项目中自定义的完整性约束，如枚举约束 (website.crawl_freq、content.type 等) 要在取值范围内。Image.resolution 这一 VARCHAR 类型需符合”宽度 × 高度” 这样的字符串格式 (如 1920×1080)。

2.3 E-R 图

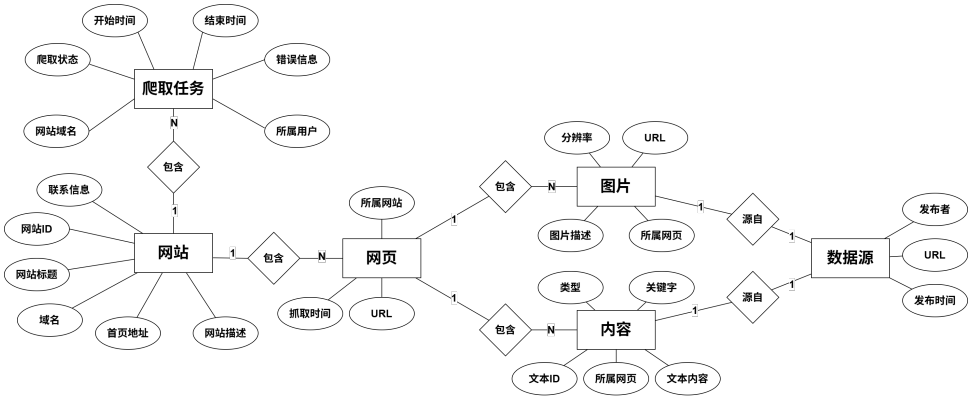

图 2: E-R 图

3 数据库设计

3.1 Website（网站）

该表用于存储被爬取的网站基本信息，字段说明如下：

id 网站编号，主键，类型为 INT

user 用户归属，类型为 VARCHAR(255)

domain 网站域名，类型为 VARCHAR(255)

title 网站标题，类型为 VARCHAR(255)

description 网站描述信息，类型为 TEXT

homepage 网站首页地址，类型为 VARCHAR(500)

3.2 Webpage（网页）

该表记录网站下的具体网页：

id 网页编号，主键，类型为 INT

url 网页地址，类型为 VARCHAR(500)

crawl_time 抓取时间，类型为 DATETIME

website 所属网站域名，类型为 VARCHAR(255)

3.3 Content（内容）

该表存储网页的文本内容：

id 内容编号，主键，类型为 INT

text 网页文本内容，类型为 TEXT

webpage 所属网页地址，类型为 VARCHAR(500)

keywords 内容关键字，类型为 VARCHAR(500)，该属性上建有索引，方便快速查询。

type 内容类型，枚举值 ('text','link')

3.4 Image（图片）

用于记录网页中提取的图片信息：

url 图片地址，主键，类型为 VARCHAR(500)

webpage 所属网页地址，类型为 VARCHAR(500)

description 图片描述信息，类型为 VARCHAR(255)

resolution 图片分辨率，类型为 VARCHAR(50)

3.5 DataSource（数据源）

该表存储来自第三方的数据源信息：

url 数据源页面地址，主键，类型为 VARCHAR(500)

publisher 发布者，类型为 VARCHAR(100)

publish_time 发布时间，类型为 DATETIME

3.6 CrawlTask（爬取任务）

用于记录每次用户提交的爬虫任务：

id 任务编号，主键，类型为 INT

user 用户 ID，类型为 VARCHAR(255)

website 目标网站，类型为 VARCHAR(255)

status 当前任务状态，枚举值 ('crawling','complete','fail')

start_time 开始时间，类型为 DATETIME

end_time 结束时间，类型为 DATETIME

error_msg 错误信息，类型为 TEXT

4 功能设计

4.1 账号管理

系统支持用户通过手动注册新账号或登录已有账号来使用平台。注册流程要求用户提供唯一的用户名、强密码，确保账户安全与真实性。已爬取的数据内容均与账号绑定，且账号间数据互不共享，保障用户隐私和数据隔离。用户可在个人中心安全退出以及切换账号。

4.2 数据爬取

用户通过输入目标网站的网址，设置爬取页数、最大超时秒数后创建爬取任务。任务提交后，系统会实时监控并反馈任务状态，包括“等待中”、“爬取中”、“已完成”及“失败”等，帮助用户掌握任务进展。后台调度系统具备负载均衡能力，保证爬取过程高效且可靠。

4.3 数据查看

系统针对已成功爬取并存储的数据，设计了多层级数据浏览机制，方便用户直观地管理和使用数据：

网站层级：展示所有爬取网站的基本信息，包括域名、最新爬取时间、总网页数等统计摘要，便于快速了解整体爬取概况。

网页层级：点击具体网站后，展示其下所有网页的 URL、标题、爬取时间以及内容摘要，支持分页与排序功能，方便用户定位感兴趣页面。

内容层级：进一步进入网页，显示网页中提取的文本内容（支持纯文本和 HTML 两种视图切换）及图片缩略图，图片支持点击查看原图或下载。所有内容均可快速跳转至对应的原始网页，确保用户访问的便捷性和数据的可追溯性。

4.4 关键词检索

系统内置全文关键词检索引擎，支持基于网页内容自动提取的关键词进行查询。用户可输入关键词，检索结果以关键词命中上下文片段的形式呈现，方便用户快速定位相关内容。所有相关链接均支持一键跳转原网页。检索性能通过高效索引结构和缓存机制保障，确保响应速度和用户体验。

4.5 数据管理

为方便用户维护数据，系统采用树形结构对爬取数据进行管理。所有删除操作均需二次确认，防止误操作导致数据丢失。

网站删除：彻底删除该网站下所有网页及其关联内容（文本、图片等）。

网页删除：删除指定网页 URL 及其所有文本和图片数据。

内容/图片删除：精细化删除选定的具体内容段或图片文件。同时，系统支持数据的批量导出、备份和恢复功能，方便用户进行离线分析及安全存档。

4.6 其他功能

数据看板：系统首页配备实时动态数据看板，展示爬取网站数量（区分活跃与失效状态）、网页总量、关键词统计、存储容量使用情况等指标，帮助用户直观了解数据变化和系统运行状况。

开发者信息：页面底部固定显示开发者联系方式和社交媒体图标（GitHub、Bilibili 等），用户点击即可跳转到相关主页，促进用户交流和反馈。

帮助中心：内置详细的用户手册，包含操作流程、常见问题解答及截图指引，帮助用户快速解决使用中遇到的问题，提高整体服务体验。

5 模块划分

5.1 数据爬取模块

该模块支持用户通过输入目标网站 URL、设置爬取深度和间隔时间等参数来创建爬取任务。系统采用多线程异步架构实现高效爬取，能够自动处理网页编码识别、动态内容渲染等复杂场景。所有爬取结果会经过结构化处理后存入数据库，包括网页文本、图片等资源。

5.2 数据查看模块

该模块以层级化方式展示已爬取的数据内容，用户可以从网站域名开始逐级下钻查看具体网页及其包含的文本和图片资源，所有网页链接都支持一键跳转至源网站；同时支持将感兴趣的内容通过命令行导出。

5.3 关键词检索模块

该模块支持通过网页内容提取的关键词查询相关网站和网页。用户可输入单个或多个关键词进行匹配，检索结果展示命中关键词的上下文片段。所有结果中的链接均支持直接跳转。

5.4 数据删除模块

该模块采用树形结构管理数据，删除操作需二次确认。网站删除：移除该域名下所有网页及关联内容；网页删除：删除选定 URL 及其文本/图片数据；内容/图片删除：删除选定的内容段和图片。

6 系统实现

6.1 开发环境与技术栈

本系统推荐在主流操作系统环境下进行开发和部署，具体包括 Windows 10 及以上版本，macOS 以及各类 Linux 发行版，确保兼容性和稳定性。编程语言选用 Python，推荐版本为 3.11 及以上，充分利用其最新特性和性能优化，同时享受丰富的第三方库支持。数据库方面，采用 MySQL 8.0 及以上版本，结合其强大的事务支持与高效的查询性能，为系统数据存储和管理提供可靠保障。

系统依赖的主要 Python 包包括：

- Django：作为核心 Web 框架，支持快速开发和部署。
- mysqlclient（或备选 pymysql）：用于与 MySQL 数据库的高效连接。
- django-environ：简化环境变量配置，方便多环境管理。
- urllib、requests：实现 HTTP 请求和网络资源访问。
- beautifulsoup4：负责 HTML 解析，结构化提取网页内容。
- url_normalize：用于统一 URL 格式，避免因不同 URL 写法导致重复爬取。
- jieba：轻量且高效的中文分词工具，支持后续关键词提取和搜索功能。

此外，开发环境建议配合使用虚拟环境（如 venv 或 conda）管理依赖，保证环境整洁且易于迁移。

6.2 核心架构图

系统采用分层架构设计，如图3所示。整体架构由用户接口层、业务逻辑层、数据持久层及爬虫模块组成。

用户接口层：基于 Django 提供的视图和模板，向用户展现交互界面，包含账号管理、任务管理、数据浏览等功能。

业务逻辑层：实现爬取任务调度、数据处理、关键词检索及权限控制等核心功能，保证系统逻辑清晰且可维护。

数据持久层：通过 Django ORM 与 MySQL 数据库交互，完成数据的存储、查询与更新。

爬虫模块：负责网络爬取和数据解析，支持多线程和异常处理，保证爬取效率和数据完整性。

该架构模块之间松耦合，便于后续功能扩展和性能优化。

图 3: 核心架构图

6.3 项目文件结构说明

本项目基于 Django 框架，采用典型的 MVC（Model-View-Controller）设计模式，目录结构规范合理，便于代码管理与维护。各主要目录和文件说明如下：

- **mysite/**：项目主配置目录，包含核心配置文件 `settings.py`（负责全局配置如数据库、应用、日志等），主路由配置文件 `urls.py`（定义全局 URL 路由规则），以及环境变量配置文件 `.env`，便于实现配置与代码分离，支持多环境灵活切换。

- **crawls/**：爬虫核心应用模块，承担整个爬取系统的业务逻辑。其中，`models.py` 定义数据库数据模型，保证数据结构规范；`views.py` 负责实现前端页面的视图逻辑；`urls.py` 作为本模块的子路由配置文件，划分模块路由；`Crawler.py`、`Downloader.py` 与 `Saver.py` 三个文件分别负责爬虫核心爬取流程、页面下载和数据保存，实现职责清晰的模块化设计；`templates/` 文件夹包含 HTML 模板，`base.html` 为主模板，定义页面的整体框架，具体页面继承并扩展于此，增强代码复用与统一风格。

- **static/**：静态资源存放目录，包括 CSS 样式文件（`css/`）和图片资源（`picture/`），支持前端界面美化和功能展示。
- **Explain.sql**：项目相关的 SQL 脚本和说明文档，包含数据库表结构创建语句、常用 SQL 操作示例及删除逻辑说明，有助于理解数据层设计及维护。
- **README.md**：项目使用说明文档，包含项目简介、安装配置步骤、功能介绍及运行指南，便于新用户快速上手。
- **manage.py**：Django 项目的管理脚本，支持项目启动、数据库迁移、应用管理等多种命令操作，是项目运行和维护的入口工具。

整体项目结构遵循模块化设计原则，各部分职责分明，既保证了系统的可扩展性和易维护性，也便于团队协作开发，提升整体开发效率和项目质量。

6.4 技术选型

本系统各技术组件的选型基于项目需求、技术成熟度和社区支持度综合考虑，具体说明如下：

Web 框架 Django Django 框架自带强大的 ORM 支持和 Admin 后台管理，具备完善的生态系统和丰富的文档资源，适合快速构建数据驱动型 Web 应用。此外，Django 内置多项安全机制，极大地降低了开发难度和安全风险。

数据库驱动 mysqlclient/pymysql mysqlclient 基于 C 语言接口，性能优越且稳定，是操作 MySQL 的主流驱动。pymysql 则为纯 Python 实现，兼容性良好，作为备用方案保障系统的稳定性与灵活性。

网页解析 BeautifulSoup4 相较于正则表达式，BeautifulSoup4 在 HTML 解析方面更加稳定和灵活，支持复杂的文档结构解析和元素定位，大幅简化爬取数据提取流程。

URL 标准化 url_normalize 该库能有效统一 URL 格式，处理多余参数、大小写等细节，避免因格式差异导致重复爬取，提高爬虫的效率和准确性。

中文分词 jieba jieba 是轻量级的开源中文分词工具，具备高效且准确的分词能力，适用于关键词提取和搜索索引构建，显著提升系统中文内容的检索效果。

6.5 数据库实现与测试验证

数据库设计严格遵循规范化原则，涵盖网站、网页、内容、图片、数据源及爬取任务等多张表结构，支持数据完整性和高效查询。数据模型通过 Django ORM 实现，确保数据库操作的安全性和一致性。

系统在开发阶段进行了全面的功能测试和压力测试，包括：

- 数据库连接稳定性测试
- 多线程爬取任务的调度与管理测试
- 数据完整性与一致性验证
- 关键词检索功能的准确性测试