

# 第7章 多层级数据架构

第一部 · 根基篇 · 第二篇 · 技术架构与基础设施

供销安选平台有一个区别于大多数互联网平台的独特特征：它需要同时服务于村、乡镇、县、市、省、国家六级行政管理体系。村级管理员需要看到本村老人的助餐领取情况，县级监管员需要看到全县各乡镇的服务覆盖情况，省级部门需要看到各地市的政策执行进度，国家级平台需要汇总全国的民生服务数据。这种多层级的数据管理需求，对平台的数据架构设计提出了独特的挑战。

本章将系统阐述平台的多层级数据架构设计，包括六级行政区域树的构建与管理、行级数据隔离的实现机制、数据从村到国家的自动聚合引擎，以及各层级监管看板的设计方案。同时，本章将从“全国一张网”的战略高度出发，阐述如何将本地的六级区域体系扩展为全国统一区域编码体系，以及如何设计跨区域数据聚合的标准接口——为“星星之火，可以燎原”的全国互联互通愿景奠定数据架构层面的基础。

## 7.1 六级行政区域树设计

### 7.1.1 区域树的数据结构

平台采用树形结构来组织六级行政区域。每个区域节点包含区域代码、区域名称、父节点代码、层级标识和全路径等核心字段。区域代码采用国家标准的行政区划代码，确保全国范围内的唯一性和可识别性。

```
CREATE TABLE t_region (  
    id BIGINT PRIMARY KEY AUTO_INCREMENT COMMENT '区域唯一标识',  
    region_code VARCHAR(12) NOT NULL UNIQUE COMMENT '行政区划代码',  
    region_name VARCHAR(100) NOT NULL COMMENT '区域名称',  
    parent_code VARCHAR(12) COMMENT '上级区划代码',  
    level TINYINT NOT NULL COMMENT '层级(1-村 2-乡 3-县 4-市 5-省 6-国家)',  
    full_path VARCHAR(300) COMMENT '全路径',  
    sort_order INT DEFAULT 0 COMMENT '排序',  
    INDEX idx_parent (parent_code),  
    INDEX idx_level (level)  
) ENGINE=InnoDB COMMENT='行政区域表';
```

区域层级的设计遵循从国家到村的六级体系。以浚县为例：国家（100000）→河南省（410000）→鹤壁市（410600）→浚县（410621）→王庄镇（410621107）→小齐村（410621107201）。每一个区域节点通过`parent\_code`字段指向其上级区域，通过`full\_path`字段记录从根节点到当前节点的完整路径。

`full\_path`字段的设计大大简化了区域树的遍历操作。当需要查询某个区域的所有下级区域时，只需要匹配`full\_path`的前缀即可，而不需要递归查询数据库。例如，查询浚县（410621）的所有下级区域，只需要`WHERE full\_path LIKE '/100000/410000/410600/410621/%'`即可一次性获取所有子节点。

### 7.1.2 区域树的缓存策略

行政区域树的数据变化频率极低——行政区划调整通常以年为单位。基于这一特性，平台将整个区域树缓存在Redis中，缓存键格式为`region:tree:{regionCode}`，缓存内容为该节点及其所有子节点的JSON序列化数据。缓存策略为永久有效，系统启动时加载完整区域树到Redis，当发生行政区划调整时通过管理后台手动刷新缓存。

对于需要频繁查询下级区域列表的场景，平台还建立了`region:children:{regionCode}`缓存，直接返回下级区域的代码列表，避免了每次查询都需要遍历整个JSON树的性能开销。

### 7.1.3 服务站点的区域绑定

每个村级服务站通过`region\_code`字段绑定到具体的行政区域。这种绑定关系使得服务站产生的所有业务数据（助餐记录、检测数据、订单信息等）都自动带有区域标识，为后续的数据聚合和行级隔离提供了基础。当需要对某个站点的归属区域进行调整时，只需要修改该站点的`region\_code`字段，历史数据保持原有区域标识不变，确保历史统计数据的连续性和准确性。

## 7.2 行级数据隔离机制

### 7.2.1 数据权限模型

多层级管理最核心的安全需求是：每一级用户只能看到自己管辖范围内的数据。县级监管员不能看到其他县的数据，乡镇级监管员只能看到本乡镇及其下属村的数据，村级管理员只能看到本村的数据。

平台采用“数据权限拦截器”来实现行级数据隔离。每个用户登录时，系统根据其角色配置的数据权限范围，确定该用户可以访问的区域代码列表。当用户进行数据查询时，数据权限拦截器自动在SQL查询条件中注入区域过滤条件，确保查询结果只包含用户权限范围内的数据。

数据权限范围有三种配置模式：第一种是“本层”——用户只能看到自己所属区域的数据，不能看到下级区域的数据；第二种是“本层及下级”——用户可以看到自己所属区域及其所有下级区域的数据，这是最常用的配置模式；第三种是“自定义”——用户可以查看指定的特定区域数据，适用于跨区域督导等特殊场景。

### 7.2.2 数据权限注入的实现

数据权限注入通过MyBatis拦截器实现。拦截器在SQL执行前，检测当前用户的数据权限配置，动态生成区域过滤条件并追加到原始SQL中。这种实现方式对业务代码完全透明——业务开发人员不需要在每个查询方法中手动添加区域过滤逻辑，只需要在业务表中包含`region\_code`字段即可自动获得行级数据隔离能力。

### 7.2.3 临时数据授权

在某些特殊场景下（如上级部门开展跨区域检查、临时借调人员需要查看其他区域数据），平台支持临时数据授权功能。管理员可以为特定用户设置临时的数据查看范围，并指定授权的有效期限。授权到期后自动失效，无需手动撤销。所有的临时授权操作都记录在审计日志中，确保临时授权的可追溯性。

## 7.3 数据聚合引擎

### 7.3.1 聚合策略

对于六级行政管理体系而言，每一级监管看板需要展示的数据指标有所不同。村级看板关注单个服务站的具体运营数据（今日助餐人数、检测人次、预警工单等），县级看板关注全县各乡镇的对比数据和服务覆盖率，省级看板关注全省各地市的排名和政策执行进度。

如果每一级看板都实时从原始业务表中查询和聚合数据，对数据库的压力将非常巨大。平台采用“实时查询+预聚合”的混合策略来解决这个问题。对于当日实时指标（如今天的助餐人数），数据量相对较小，直接从业务表实时查询并通过Redis缓存五分钟。对于历史趋势数据（如过去三十天的助餐人数变化），通过每日凌晨的定时任务进行预聚合计算，结果存储在聚合表中。

### 7.3.2 每日聚合调度

平台设计了每日聚合调度机制。每天凌晨两点（业务低谷期），聚合调度器自动执行以下流程：从最底层的村级数据开始，汇总各村当日的助餐人次、检测人次、城乡通订单数、零工撮合数、居家养老服务单数等核心指标；然后将村级数据向上汇总到乡镇级，乡镇级再向上汇

总到县级，以此类推，直到国家级。每一级的汇总结果都存储在`t\_aggregation\_day`表中，按照区域代码和统计日期进行唯一标识。

```
CREATE TABLE t_aggregation_day (  
    id BIGINT PRIMARY KEY AUTO_INCREMENT COMMENT '记录唯一标识',  
    region_code VARCHAR(12) NOT NULL COMMENT '区域代码',  
    stat_date DATE NOT NULL COMMENT '统计日期',  
    meal_total INT DEFAULT 0 COMMENT '就餐总人次',  
    meal_subsidy DECIMAL(12,2) DEFAULT 0 COMMENT '就餐补贴总额',  
    health_check_total INT DEFAULT 0 COMMENT '健康检测人次',  
    health_abnormal_count INT DEFAULT 0 COMMENT '异常指标人次',  
    disability_review_count INT DEFAULT 0 COMMENT '年审人次',  
    urban_rural_orders INT DEFAULT 0 COMMENT '城乡通订单数',  
    job_match_count INT DEFAULT 0 COMMENT '零工撮合成功数',  
    home_care_orders INT DEFAULT 0 COMMENT '居家养老服务单数',  
    alert_total INT DEFAULT 0 COMMENT '预警总数',  
    alert_handled INT DEFAULT 0 COMMENT '已处理预警数',  
    UNIQUE KEY uk_region_date (region_code, stat_date),  
    INDEX idx_stat_date (stat_date)  
) ENGINE=InnoDB COMMENT='区域每日运营汇总表';
```

这种预聚合策略使得各级监管看板的加载速度大幅提升——对于历史数据的查询，只需要从聚合表中读取预先计算好的汇总数据，而不需要扫描海量的原始业务记录。

7.3.3 聚合数据的多维度展示

聚合数据不仅用于监管看板的展示，还可以支持多维度的数据分析。例如，可以按时间维度分析某个县过去一年各月的就餐人数变化趋势，可以按区域维度对比各乡镇的服务覆盖率差异，可以按指标维度分析就餐、健康、就业等各项服务的协同发展情况。这些分析功能在数据分析与驾驶舱模块中提供。

7.4 各层级监管看板设计

7.4.1 看板的核心设计原则

各层级监管看板的设计遵循三个原则：第一，信息密度逐级递减。村级看板展示最详细的运营数据，国家级看板只展示宏观汇总指标。第二，关注焦点逐级上移。村级关注单个站点的服务质量，县级关注各乡镇的对比和排名，省级关注各地市的政策执行进度。第三，下钻能力贯穿始终。任何一级看板都支持点击某个数据节点向下钻取，查看下一级区域的详细数据。

7.4.2 各层级看板核心指标

| 层级  | 核心指标                                   | 特色功能                             |
|-----|----------------------------------------|----------------------------------|
| 村级  | 今日助餐人数、检测人次、年审完成数、城乡通订单、零工撮合数、居家养老服务单数 | 重点老人健康列表、预警工单处理、农户销售排行、两委公开统计    |
| 乡镇级 | 全镇汇总数据、环比增长率                           | 各村综合排名、地图视图标注、异常监测面板、村务公开监管      |
| 县级  | 全县汇总数据、各乡镇完成率对比                        | 资金使用看板、慢病分布热力图、农产品流通地图、零工供需分析    |
| 市级  | 全市汇总数据、各县区覆盖率排名                        | 县区综合排名、资源配置利用率、政策执行进度看板、就业分析     |
| 省级  | 全省汇总数据、各地市覆盖率                          | 地市排名大屏、全省服务密度热力图、资金监管总览、示范站点遴选   |
| 国家级 | 全国服务站总数、服务总人次、覆盖行政村数量                  | 省份排名大屏、全国服务密度热力图、城乡流通大图、政策效果评估报告 |

7.4.3 下钻交互设计

看板的下钻功能通过面包屑导航实现。用户在国家级看板上点击某个省份，页面自动切换到该省的省级看板；在省级看板上点击某个地市，切换到该市的市级看板；以此类推，直到村级看板。面包屑导航始终显示当前的层级路径，用户可以随时点击面包屑中的任意层级快速跳转。下钻过程中，时间筛选器和指标类型等筛选条件保持同步，确保用户的分析思路不会中断。

7.5 跨区域数据聚合——“全国一张网”的标准化接口

如果说前四节解决的是单个平台内部的多层级数据管理问题，那么本节要解决的是多个平台之间的跨区域数据聚合问题——这正是本书所倡导的“星星之火，可以燎原”愿景在数据架构层面的核心设计。

7.5.1 从本地六级到全国统一区域编码

第六章中，我们已经在数据库层面预留了数据联邦标识体系。本章在此基础上，进一步阐述跨区域数据聚合的完整方案。首先需要解决的是区域编码的统一问题。平台在本地使用六级

行政区域代码，当需要跨区域聚合时，这套编码体系可以直接扩展为全国统一的区域标识——因为各级区域代码本身就是国家标准行政区划代码，天然具有全国唯一性。

这意味着，浚县的区域代码“410621”在浚县本地平台和在任何一个接入全国网络的其他平台上，代表的都是同一个区域。不需要额外的编码映射或转换，各地的数据天然就能够按照统一的区域维度进行聚合和对比。这种“天然兼容”的特性，大大降低了跨区域数据聚合的复杂度。

### 7.5.2 三级联邦聚合协议

平台定义了三种层次的跨区域数据聚合协议，分别对应不同的数据敏感级别和共享需求：

第一层：脱敏汇总数据上报。各县级平台定期向上级平台（市级、省级、国家级）上报脱敏的汇总统计数据。上报的数据是聚合后的指标（如本月助餐总人次、覆盖率、补贴总额），而非原始的个人数据。这种模式保障了个人隐私和原始数据安全，同时满足了上级部门的统计和监管需求。上报接口采用标准的HTTPS+JSON格式，数据字段和统计口径在全国范围内统一规定。

第二层：有条件的数据协同。当发生跨区域的业务协同时（如A县的农产品销往B县、C县的劳动力到D县就业），相关区域平台在获得数据所有者授权后，通过标准接口进行最小必要的数据共享。共享的数据仅限于完成该业务协同所必需的字段，不包含完整的个人信息。例如，跨区域农产品流通只需要共享产品信息、农户联系方式和物流信息，不需要共享农户的健康数据或家庭信息。

第三层：联邦查询。当上级平台需要汇总多个区域的数据进行宏观分析时，采用联邦查询模式。上级平台向各下级平台发送标准化的查询请求，各下级平台在本地完成数据计算后，返回脱敏的查询结果。原始数据始终保留在各区域平台本地，不出域。联邦查询的关键在于标准化——查询请求的格式、返回结果的字段和口径、数据脱敏的规则都需要在全国范围内统一规定。

### 7.5.3 跨区域聚合接口标准

为了实现上述三级联邦聚合协议，平台定义了以下核心标准接口：

第一，汇总数据上报接口。各县级平台调用此接口向上级平台推送脱敏的汇总数据，请求体包含区域代码、统计周期、指标名称和指标值。第二，数据共享请求接口。当需要跨区域业务协同时，请求方向数据所有者发送数据共享请求，说明共享目的、所需字段和使用期限。第三，联邦查询接口。上级平台向各下级平台发送查询请求，包含查询条件、聚合维度和返回格式。

这些标准接口的定义，使得各地按照本书建设的平台在需要互联互通时，只需要按照接口规范实现对应的接口即可，不需要重新协商数据格式和通信协议。这正是标准化带来的效率提升——不是每一个平台都要与其他所有平台进行点对点的对接，而是大家按照同一套标准接入联邦网络，实现“一次对接、全网互通”。

#### 7.5.4 循序渐进的数据联邦之路

跨区域数据聚合的实现不是一蹴而就的。在第一阶段，各区域平台独立建设，优先满足本地的多层级数据管理需求，但按照本章和第六章的标准预留好联邦标识字段和标准接口。在第二阶段和第三阶段，当多个县级平台建成并运行稳定后，可以在市级或省级范围内进行互联互通试点——先实现“脱敏汇总数据上报”这一最基础的联邦层级，验证标准接口的可行性和数据质量。在第四阶段，逐步扩展到“有条件的数据协同”和“联邦查询”，最终形成完整的全国数据联邦网络。

这种循序渐进、从本地到全国的实现路径，与第三章提出的“四阶段划分”策略完全一致。每一颗扎根乡土的火种，首先要在本地扎稳根基、服务好一方百姓；当根基扎实之后，再通过标准化的通道与其他火种连接起来，形成燎原之势。

### 7.6 总结

多层级数据架构是供销安选平台区别于大多数互联网平台的核心特征之一。六级行政区域树为数据的层级化管理提供了基础框架，行级数据隔离机制确保了各级用户只能访问自己权限范围内的数据，每日聚合引擎通过预计算大幅提升了各级监管看板的响应速度。

更重要的是，这套多层级数据架构天然支持向全国范围的扩展——国家标准的行政区划代码体系为区域编码的统一提供了基础，三级联邦聚合协议为不同层级的数据共享需求提供了标准化的解决方案。各地按照本章的标准建设平台，未来就能无缝接入全国供销数据联邦网络，真正实现“全国一张网”的愿景。

#### 本章核心观点

六级行政区域树（村→国家）是平台多层级管理的数据基础，通过full\_path字段简化树遍历。行级数据隔离通过MyBatis拦截器自动注入区域过滤条件实现。每日聚合调度引擎在凌晨自动从村到国家逐级汇总核心指标。各层级监管看板遵循“信息密度逐级递减、关注焦点逐级上移、下钻能力贯穿始终”的设计原则。跨区域数据聚合采用三级联邦聚合协议——脱敏上报、条件共享、联邦查询。国家标准行政区划代码体系为全国统

一区域编码提供了天然基础，各地按照统一标准建设，即可实现“一次对接、全网互通”。