2022年宋家庚终总结

# 1 技术&业绩贡献

## 1.1 「广州科贸数据治理项目」-后端开发与实施-10月份

### 1.1.1 项目背景与目标

项目内容是根据广州科技贸易职业学院的项目标书对数据治理进行新功能的开发，以及进入学校进行项目实施。

### 1.1.2 工作内容

#### 1.1.2.1 需求开发

* 元数据采集支持DB2数据库
* 数据共享开放
  + 数据共享模块迁移
  + 数据共享api鉴权
    - ip白名单后端开发
    - 用户名密码鉴权
  + 数据共享api日志记录
  + 实体上下架
  + 数据共享-视图共享
* 数据归档
  + 数据归档-还原功能
* 数据集成
  + 自动生成值域转换规则
* 高基表数据展示应用后端开发

#### 1.1.2.2 产品优化推动

* **元数据管理**
  + 元数据管理易用性改进  
    原先的元数据管理界面对于操作太不友好，无法一眼看到可以理解的中文信息
  + 修复库迁移无法迁移到来源库的注释的问题
* **数据集成**
  + ETL流程-字段映射组件易用性改进  
    原先的字段映射组件界面操作效率太低，需要在数据库管理工具、表结构文档、数据管理系统之间来回切换才能获得对应的操作信息。
* **数据共享**
  + 共享API文档导出功能  
    配置出了API还需要手动写API文档，极为影响效率
* **元数据采集**
  + 元数据采集新增属性：字段数据量、字段示例值  
    实际清洗采集标准数据时总是会关注**字段数据量、字段示例值**的信息，需要在数据库管理工具查询对应数据是否可用后，回到系统界面来进行配置操作，极为繁琐、对操作人员要求较高：需要懂得操作数据库。
  + 修正无法采集Oracle数据库注释的问题  
    原先系统无法采集到Oracle的数据库注释，导致采集Oracle数据库的时候，需要时刻对照表结构文档来配置流程。
  + 修复数据中心在元数据重新采集后无法查看数据的问题
  + 修复元数据采集无法同时执行多个任务的问题  
    多个元数据采集任务同时执行时，只有一个能够执行成功，严重影响元数据采集的效率。
* **ETL任务**
  + 改进大量任务、大数据量采集时采集引擎宕机的问题  
    原先数据量超过百万条的数据采集就会导致内存爆满，进而导致系统宕机，无法正常提供服务
  + 修复etl任务执行失败无法再次执行的问题

#### 1.1.2.3 实施

* 部署手册编写
* 部署脚本编写
* 搭建内网穿透实现远程运维

#### 1.1.2.4 数据运维

* 数据摸排与权限申请
  + 摸排到学校共有13个业务系统，总数据量达到6千万
* 原始库采集
* 标准库采集
* 高基表数据采集

### 1.1.3 产出和成果

1. 需求开发部分，按照计划排期准时进行了提测交付
2. 产品优化推动
   1. 元数据管理、数据集成、元数据采集等模块的优化，提高了数据运维的效率，将原先需要两、三个界面切换才能获取足够信息进行数据采集配置的部分，简化为一个步骤，甚至扫一眼就能获得对应的信息；还降低了数据运维的门槛，使得非技术人员也能更加简易的使用系统进行数据运维。
   2. 元数据采集、ETL任务模块的优化解决了系统不稳定的问题，使的系统能够承担更加繁重的采集任务，优化后，采集系统可以承担数据量超过6千万而不会导致宕机的问题。
3. 实施
   1. 实施手册与脚本的编写记录了部署系统的全部操作，可以完全按照手册快速部署出可以自动恢复运行的系统。
   2. 搭建的内网穿透实现了远程运维，节省了现场运维的人力成本，按照学校要求，所有运维、实施都需要到达现场，搭建了内网穿透后只需一位运维岗人员在现场，可以有更多其他开发岗同事同时远程到学校。

## 1.2 「知彦官网」-后端开发-8月份

### 1.2.1 项目背景与目标

按照原型设计完成知彦官网的前台与后台开发

### 1.2.2 工作内容

#### 1.2.2.1 时间排期

对项目进行需求梳理、分析需求，进行项目排期。

#### 1.2.2.1 数据库设设计

按照原型设计中的首页、产品、解决方案、案例、留资表单等页面，设计相关的表结构，并进行数据库设计评审

#### 1.2.2.1 整体设计与开发

按照数据库设计，对相关代码结构进行整体设计，模块间互相协作，编写可复用、易维护的代码

### 1.2.3 产出和成果

按照排期准时完成了知彦官网的开发

# 2 专业能力

## 2.1 架构设计能力

* 优化ETL任务的执行：原先ETL执行时，碰到大数据量的任务，会导致系统宕机，优化后使用拉数据的方式按需加载数据到内存，执行同样的任务，采集万条数据时减少90%的内存占用，采集十万条数据时减少99%的内存占用
* 对需求中相似部分抽象，做出整体设计，从而实现代码复用提高开发效率；例如知彦官网中多种内容相似、且需要互相关联页面的情况，对其在存储层进行抽象，实现代码复用，提高了开发效率与代码可维护性。

## 2.2 开发能力

* 产品思维能力：根据使用的体验提出并实现操作逻辑的优化，将原来三到五个步骤的操作简化为单个操作即可完成，例如：字段映射、API文档编写、元数据搜索、表搜索等（可参考1.1.2.2章节的内容）
* 发现原先有设定代码规范，但是大多项目都没有实际遵守，实现了项目中自动化代码质量检测后，对提交的代码自动的进行质量上的过滤，保证代码的规范性，降低后续的维护成本。

## 2.3 运维能力

* 能够熟练的进行服务器部署脚本的编写，例如：广州科贸项目的部署脚本
* 快速进行bug响应，开发阶段时的bug停留时间一般不超过1天。

## 2.4 团队沟通

能够在同事遇到问题进行求助时及时相应，帮忙解决问题。

# 3 自我总结

## 3.1 自身的问题

实施科贸项目以来，发觉自己的耐心还是不够，在清洗数据、配置采集流程时没法敏锐的发现相关的数据，导致效率对比熟练的同事偏低。

## 3.2 发现的问题

* 发现公司没有没有统一管理提测版本相关信息，导致无法回溯可用的测试版本
* 发现公司没有统一的代码构建环境与应用包仓库，导致需要应用包时需要人为重新构建
* 部署项目时没有使用外部配置文件，导致改配置都需要重新打包

## 3.3 个人发展方向

个人希望在技术上有更多的探索，能够通过技术上探索提升业务上的优化，例如学习AKKA组件，提高数据治理采集数据的执行效率；探索业界的数据计算引擎flink，是否可以应用到数据治理，以提升产品的竞争力。