* 任务一

1.熟悉Azkaban在大数据仓库中的架构层级和使用场景，包含一些基本单元的概念熟悉，包含project、jobFlow、job等  
 2.了解Azkaban jobFlow和job的种类和依赖  
 3.本地搭建Azkaban开发环境，建立Azkaban元数据库，熟悉Azkaban元数据库每张表的用途  
 4.针对Azkaban元数据库，进行JobFlow和Job的血缘关系溯源开发，能够呈现Project--JobFlow--Job之间的血缘关系  
 5.熟悉antlr4语法解析

# 2021.6.1

学习Azkaban，复习Hive。

# 2021.6.2

学习《大数据之路：阿里巴巴大数据实践》的第9-10章，了解大数据模型的建模理论。包括阿里巴巴数据整合及管理体系，维度的设计。

# 2021.6.3

学习《大数据之路：阿里巴巴大数据实践》的第10-11章，了解如何建立维表和事实表。

# 2021.6.7

学习《大数据之路：阿里巴巴大数据实践》的日志采集理论。

开始学习Antlr，学习自定义解释器。具体：安装ANTLR插件，完成最基本的一次案例，学习如何与Java集成使用。

# 2021.6.8

学习《antlr4权威指南》第4~5章，学习基本的设计语法，和概念介绍。

利用监听器，访问器构建程序，了解词法、语法特性。

# 2021.6.9

1.在虚拟机中搭建本地的Azkaban集群环境。

2.通过阅读Azkaban的官方文档，了解Azkaban的基本使用方法，以及对project、jobFlow、job等相关概念的熟悉。

3.通过gitHub社区尝试了解Azkaban元数据表的类型和作用

# 2021.6.10

1.查询了解Azkaban元数据库每张表的用途并记录。

2. 针对Azkaban元数据库，进行了一个简单Project的运行，通过分析元数据表的数据，了解Project--JobFlow--Job之间的血缘关系并呈现。

# 2021.6.11

1.按要求设计接下来的工作安排。

2.了解YAML语法规则，尝试解析YAML语句。

# 2021.6.15

1.成功解析YAML语句，并完成JobNode的封装

2. 继续学习antlr。

# 2021.6.16

1.在昨天的基础上，设计数据结构存储jobNode，并完成对血缘关系的展示。

2. 继续学习antlr，完善解析代码。

# 2021.6.17

1.学习和熟悉HiveSQL.g4文件。

2.学习gitLab、CDP hive的使用，学习Docker。

# 2021.6.21

1.完成ods数据层表结构的Excel归档。

2.完成其他数据层中数据库的各表的统计汇总。

# 2021.6.22

修改ods表数据文件，校验建表语句的有效性。

# 2021.6.23

验证表迁移工作中要使用的各个账号。了解CDH到CDP重要的135张ods表的重构过程，并开始进行重构工作的第二步注释工作。

# 2021.6.24

因为网络原因，只在本地参考confluence上的字典文件，查询重构表的字段和表COMMENT，为后续重构做准备。

# 2021.6.25

继续完成ods表的重构任务。

# 2021.6.28

1.学习git的使用，尝试clone、pull、push以及其他操作。

2.完成了重构表的注释工作，并创建了一些表，上传了一些问题表的字段。

# 2021.6.29

完成剩下所有表的重建工作，包括无问题表的创建，和有问题表的问题上报。并将文档push至gitLab仓库。

# 2021.6.30

1.完成佳恒哥剩余的表重构工作，并push到gitLab

2.深入体验gitLab工具的用法。

# 2021.7.1

梳理Azkabn上所有Project下各个JobFlow的Job运行情况，包括JobFlow和Job的归属关系，Job的最后一次运行时间，Job的负责人和Job的调度信息。

# 2021.7.2

继续并完成梳理Azkabn上所有Project下各个JobFlow的Job运行情况。

# 2021.7.5

1.开始梳理ods\_oms库的所有表的血缘关系，血缘路径由连接 外部表->ods层任务名 -> ods层oms表名 -> dwd层任务名 -> dwd层oms表名 构成。

2.完成临时布置的任务：排查oms数据源的建表字段与ods层建表字段的差异性。

# 2021.7.6

1.完成oms相关表字段的排查

2.完成ods\_oms库的血缘关系溯源。

# 2021.7.7

1.开始进行ods\_couponcenter库表与远程数据库相关表字段的排查。

2.同时开始进行couponcenter库从远程数据库到ods层再到dwd层的血缘流向。

# 2021.7.8

1.完成ods\_couponcenter相关表字段的排查

2.完成ods\_couponcenter库的血缘关系溯源。

3.安装Python和虚拟机，尝试修复Hive不能启动的问题

# 2021.7.12

尝试完成对Azkaban页面爬虫脚本的开发。

# 2021.7.13

完成对Azkaban页面爬虫脚本，实现监控各工程下各个流执行状态。

# 2021.7.14

1.完善Azkaban页面爬虫脚本，并测试成功。

2.完成针对ETL数据源登记、管理和监控的表设计ER图。

# 2021.7.15

1.完善Azkaban页面爬虫脚本的告警输出。

2.复习flink。

# 2021.7.16

完成针对ETL数据源登记、管理和监控的表设计项目的SQL脚本，并在本地测试无误。

# 2021.7.19

1.开始写数据源检活代码。

2.梳理icc的血缘关系。

# 2021.7.20

1. 初步完成检活项目，完成mysql的对接。

2. 完成sundry血缘梳理。

# 2021.7.21

继续完善检活项目，主要完成sftp的对接。

# 2021.7.26

完成mongodb、tiflash、sqlserver、postgresql数据源的检测代码。

# 2021.7.27

1. 完成oracle、redshift数据源的检测代码，将代码重构。

2. 开始进行数据库的数据插入。

# 2021.7.28

1. 完成进行数据库的数据插入。

2. 学习DataX数据同步框架，寻找ClickHouseWriter和HiveWriter插件。找到ClickHouseWriter并在本地成功打包。

# 2021.7.29

学习DataX,并在本地搭建ClickHouse环境，准备后续测试。

# 2021.7.30

完成环境的搭建和测试dataX的使用。

# 2021.8.2

1. 完成dataX中mysql到hive的数据同步测试

2. 初步完成88节相关表状态检测报警代码

# 2021.8.3

1. 完成hive到clickhouse的数据同步测试代码，等待ck的账号密码进行最终测试

2. 基本完成88节相关表状态检测报警的脚本的基本功能。

# 2021.8.4

1. 完成88节相关表检测脚本，并已在服务器上部署好了脚本执行环境，等待DNS问题解决就可以完成最终测试部署。

2. 完成数据从mysql到hive再到clickhouse的同步测试，并完成相关文档的编写。

# 2021.8.5

完成88节相关表检测脚本，并已在服务器上部署好了脚本执行环境，等待DNS问题解决就可以完成最终测试部署。

# 2021.8.6

完成88节相关表检测脚本最终测试与上线。

# 2021.8.9

1.完成Azakaban的状态监测部署。

2. 开始梳理神策数据的血缘关系

# 2021.8.9

继续梳理神策数据的血缘关系，完成文档的撰写。

# 2021.8.10

梳理神策数据的血缘关系，完成文档的撰写。

# 2021.8.11

完成etl\_daily\_sensorst3血缘追踪。

# 2021.8.12

完成confluence上数据接入的文档撰写。

# 2021.8.13

开始进行hsql的编写，主要面向表在HDFS下的磁盘占用情况。

# 2021.8.16

完成hsql的编写，目前在ods\_sundry进行数据写入测试。

# 2021.8.17

完成表存储磁盘容量计算功能的Hql优化，并形成Excel文档。

# 2021.8.18

开始完成CDH到CDP平台的Hive表迁移，已经完成schema的准备。

# 2021.8.19

完成CDH到CDP平台的迁移表创建，并形成Excel文件。

# 2021.8.20

开始对CDH和CDP平台迁移表的数据进行校验，针对archive库的外表进行分区SQL编写，并执行分区设置操作。

# 2021.8.23

完成archive库所有外表的分区设置，并对备份表进行一次数据校验。

# 2021.8.24

1. 完成最终数据归档，并上传代码至gitlab备份，完成jira的报告。

2. 完成对azkaban监测脚本 补充超时运行的功能。

# 2021.8.25

对ods\_tracking的两张表进行创建验证，已在CDP平台创建，并验证部分。

# 2021.8.26

对ods\_tracking的ods\_ext\_csv\_unc\_sensors\_events\_cma的分区修改以及数据校验，对ods\_ext\_csv\_unc\_sensors\_events分割进行数据校验。

# 2021.8.27

完成对ods\_tracking流量历史的校验工作。

。