# ETL设计文档

目录

[ETL设计文档 1](#_Toc2599654)

[1、 概述 1](#_Toc2599655)

[2、 ETL系统架构设计 2](#_Toc2599656)

[3、 ETL应用架构设计 2](#_Toc2599657)

[4、 ETL作业配置 3](#_Toc2599658)

[5、 ETL数据处理方式 4](#_Toc2599659)

[5.1 ods层（文件数据到GP） 6](#_Toc2599665)

[5.2 ods至dwd 6](#_Toc2599666)

[5.3 dwd至dim\_dm 7](#_Toc2599667)

[6、 ETL监控与错误作业自动重试 7](#_Toc2599668)

[6.1 ETL监控 7](#_Toc2599670)

[6.2 错误作业重试机制 8](#_Toc2599671)

[7、 错误作业手动重跑与日常维护 8](#_Toc2599672)

[7.1 日常维护操作 8](#_Toc2599674)

[7.2 日常可能出现问题 8](#_Toc2599675)

## 概述

ETL系统的核心功能就是按照本设计说明书的架构，将数据由数据源系统加载到数据仓库中。其实现的困难在于ETL系统将面临复杂的源数据环境，包括多种多样的数据源平台、繁多的数据种类、巨大的加载数据量、错综复杂的数据关系和参差不齐的数据质量，这些都使ETL的架构和应用设计面临相当的挑战。

通过高效的ETL系统结构、层次化的应用功能划分和标准的程序模板，ETL系统和应用架构设计需要能够达到以下目标：

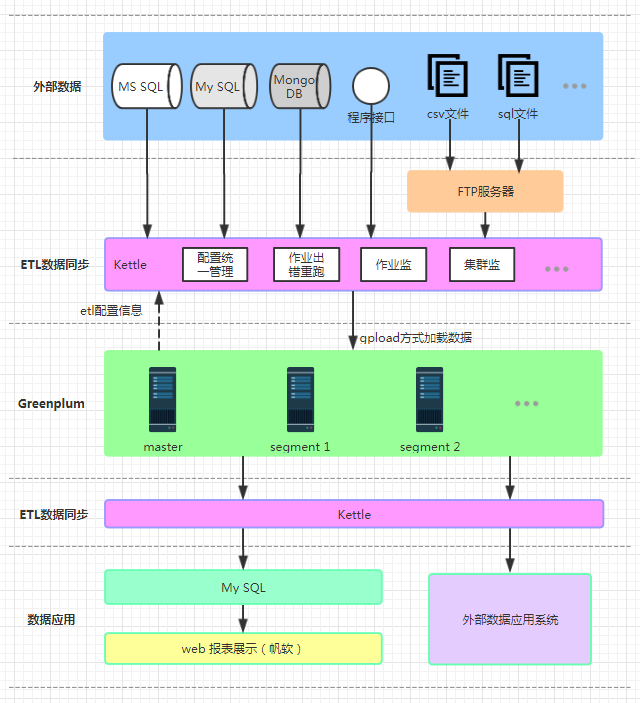
1. 支持各种数据源数据加载到数据仓库的功能；
2. 支持在规定的批处理时间窗口内能够完成数据加载工作，即需要满足日常数据加载的性能需求；
3. 支持作业依赖关系配置，保障作业的执行先后顺序；
4. 支持作业出错自动重试，降低由于短暂网络中断或者源系统升级等不稳定因素的影响，保障系统的健壮性；
5. 减少系统维护的复杂性，支持后续增加新数据或功能的开发工作。

本设计说明书包含以下部分：

1. ETL系统架构设计；
2. ETL应用架构设计；
3. ETL作业配置；
4. ETL数据处理方式；
5. ETL监控与错误作业自动重试；
6. 错误作业日常维护。

## ETL系统架构设计

系统架构设计包括ETL需要关联到的相关系统和系统模块，包括多样的数据源（包括各种数据库、文件方式和程序接口方式等）、ETL相关配置信息、数据存储系统和数据应用系统等。



## ETL应用架构设计

ETL系统部分包括以下三部分：

1. 定时任务启动

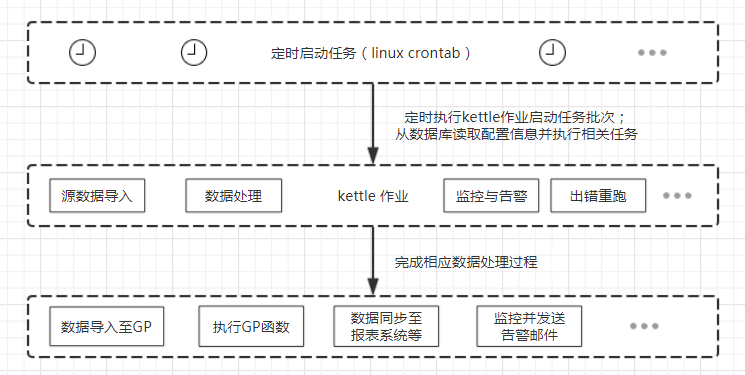
任务调度部分中，使用linux系统的crontab设置定时任务启动任务批次，启动相应的批次之后会按优先级执行前置作业已经完成的作业。

1. 任务配置

在执行作业时用到的相关信息配置在GP库的相应表中，比如数据库连接信息、源表信息、目标表信息、执行作业类型和告警邮件接收人信息等。

1. ETL功能模块

按照作业类型分多功能模块，如：数据导入至GP、执行GP函数完成相关数据汇总和数据处理、从GP将数据同步至其它系统（报表系统等）和监控告警类作业。



## ETL作业配置

所有etl作业通过读取配置表相关信息来确定需要执行哪个kettle子作业、源数据和目标表等信息，这些配置信息表存放在GP的etl模式下面。相关配置信息表说明如下：

etl.etl\_conf\_batch：保存批次的信息，每个批次会开启一个jvm来执行该批次下面的所有作业，目前基本按照每个数据源每一层处理一个单独的批次。

etl.etl\_conf\_job：保存作业信息，作业执行的kettle子作业信息，源数据信息和目标表信息，还有作业运行状态和时间，并通过batch\_id来确定在哪个批次下面执行该作业。

etl.etl\_conf\_job\_table\_field：保存作业处理的表对应字段信息（目前主要针对文件到ods层）

etl.etl\_conf\_kettle\_job：保存kettle资源库中子作业对应的目录和作业名。

etl.etl\_conf\_precondition：保存作业之间的依赖关系，作业需要等前置作业跑完才能执行

etl.etl\_receive\_mail\_address：保存需要接收告警邮件的email地址

etl.etl\_send\_mail\_account：用来设置发送邮件的邮箱的地址、密码、SMTP信息，其中密码需要通过kettle资源库下面的/public/tr\_encode\_password先加密，将密文存放到密码字段中。

## ETL数据处理方式

作业流程图如下：





### ods层（文件数据到GP）

数据源系统将数据文件上传至ftp服务器后，etl会先下载chk文件，如果chk文件正常下载完成，则下载数据文件，将数据文件中的数据通过gpload方式导入至GP；否则降低作业的优先级，先把其它作业跑完。如果出现多次等待都没有发现文件，则将作业标识为无文件错误，不再继续等待。

### ods至dwd

#### 全量数据无主键

对于ods层的全量数据表，且该表没有主键的，处理方式是将dwd原有表truncate，然后将ods最新那天的表数据insert到dwd表。

#### 全量数据有主键

对于ods层的全量数据表，如果该表是有主键的，会将ods当天数据和dwd原有数据对比（kettle对比），将有变化的数据通过gpload的merge方式更新或者insert到GP。

#### 增量数据无主键

对于ods层的增量数据表，如果该表有主键，就先将dwd表中有增量更新的记录删除，然后将ods当天增量数据insert到dwd对于的表中。

#### 增量数据有主键

对于ods层的增量数据表，如果该表没有主键，就想ods表中的数据直接insert至dwd。

### dwd至dim\_dm

#### dim数据以拉链表的形式保存至GP

直接调用GP函数处理，GP函数将dwd中更新的数据对应dim表中的有效截止时间更新为前一天日期，之后见dwd中的数据增加有效开始时间为前一天，有效截止时间9999-12-31之后插入dim表中。

#### dm数据汇总统计

直接调用GP函数处理，根据dm表的统计口径统计汇总数据直接插入dm表中

## ETL监控与错误作业自动重试



### ETL监控

1）监控作业的运气状态，作业是否正常运行、外部数据文件是否已上传、在容错范围内小量格式错误数据告警等，出现相关错误之后，通过邮件方式发送给对应负责人处理错误。

2）编写相关数据检查存储过程，监控数据异常情况，就是在作业正常完成条件下，比如出现数据量明显下降的情况下，会发出告警邮件提醒是否存在数据同步不完全的问题；比如一些平衡计算（收支问题），如果出现数据对不上，也告警提示。

3）监控Greenplum集群运行情况，如由于网络问题，会导致Greenplum集群中有segment会出现脱离集群的情况，通过监控并告警，让相关负责人恢复segment。

### 错误作业重试机制

1）由于ETL过程中涉及多数据源、多系统、跨网络的数据交互，因此会存在网络短暂中断或者源数据系统短暂升级出现系统停机等情况，因此需要调度流程有自动重试机制，保障数据源短暂不能连接，对数据同步无明显影响。每个作业批次会设置一个重试次数，如果该批次里面的作业在没有达到重试次数时，出现错就会等待一小段时间，然后重跑，直到完成作业或者到达重试次数就不再尝试。

2）另外可能存在一些数据源是对方提交文件到ftp服务器中，这种方式的如果对方接口出现问题，往往需要过较长时间才能恢复，补传数据文件，这时需要另外一个自动作业去定时扫描是否有可以运行的作业。

## 错误作业手动重跑与日常维护



### 日常维护操作

1）当出现作业执行错误时，先看日志信息，确定出错原因，等出错原因修正之后。

2）修改批次表的日期，next\_date改为要跑的日期，把last\_date改为next\_date的前一天。

3）并将作业表中的对应出错作业的状态改为正常并把update\_date改为要跑的日期前一天。

4）重新执行相应批次的启动脚本即可。例如./run\_job.sh job\_batch\_ht\_file\_to\_ods\_1。

### 日常可能出现问题

一、各数据源没有上传数据到ftp服务器，作业经过多次（目前设置为5次）等待重试之后还是没有发现ftp存在文件，则将作业的状态改为2。这时需要等文件上传完成后，按照日常维护操作的步骤重跑作业。

二、前置作业没有完成，导致作业一直没办法执行，批次在循环等待中。比如ods层一个表因为没有文件或者其它原因未成功执行，这时dwd对应的表的作业会一直在等待，后续需要用到该表的处理作业都会在等待。如果当天可以把前置作业正确跑完，则无需将后续的作业跳过，否则需要跳过后续作业的执行，这样才不会影响下一天该批次作业的执行。（目前批次中设置了如果到了晚上23点，就会跳过所有未执行的作业，只能等前置作业成功执行之后再手动重跑）

三、作业的状态正常，但是发现gpload导入数据有错误信息，需要查看这些数据是否为脏数据，如果是有效数据，比如源数据的某个字段长度变了，导致导入出错，这时候需要修改GP的对应字段的长度，重跑作业。