|  |  |
| --- | --- |
| Aa |  |

|  |
| --- |
| 深圳闽星科技有限公司  大数据平台建设项目  ETL设计方案  2019年7月 |

Untitled-1目 录

[1. 文档概述 1](#_Toc16760962)

[1.1 文档目的 1](#_Toc16760963)

[1.2 项目背景 1](#_Toc16760964)

[1.3 文档读者 1](#_Toc16760965)

[1.4 参考资料 1](#_Toc16760966)

[1.5 名词与术语 1](#_Toc16760967)

[1.6 约束 2](#_Toc16760968)

[2. ETL设计概述 3](#_Toc16760969)

[2.1 ETL设计总体考虑 3](#_Toc16760970)

[2.2 ETL设计策略 3](#_Toc16760971)

[3. ETL应用框架设计 3](#_Toc16760972)

[3.1 ETL应用架构逻辑图 3](#_Toc16760973)

[3.2 ETL模式 4](#_Toc16760974)

[3.3 数据抽取（Extract） 4](#_Toc16760975)

[3.4 数据变换（Convert） 4](#_Toc16760976)

[3.5 数据转换（Transform） 4](#_Toc16760977)

[3.6 数据加载（Load） 5](#_Toc16760978)

[3.7 ETL数据质量检查 5](#_Toc16760979)

[3.8 ETL进程和进程调度 5](#_Toc16760980)

[3.9 日常数据、初始数据和历史数据ETL 5](#_Toc16760981)

[3.10 ETL中使用的日期 6](#_Toc16760982)

[4. 数据处理 6](#_Toc16760983)

[4.1 公用维度 6](#_Toc16760984)

[4.1.1 客户表 6](#_Toc16760985)

[4.1.2 仓库表 6](#_Toc16760986)

[4.2 订单主题 6](#_Toc16760987)

[4.2.1 XX维度 6](#_Toc16760988)

[4.2.2 订单头事实表 6](#_Toc16760989)

[4.2.3 订单行事实表 6](#_Toc16760990)

1. 文档概述
   1. 文档目的

该设计文档为ETL开发提供指导，着重叙述数据仓库系统的ETL系统的架构、功能和实施模板，但未包含针对数据仓库中每个具体数据表的ETL详细设计，由开发人员根据《源-目标映射》中所规定的ETL规则，按照ETL设计的应用开发流程和规范，参照模板程序，实现每一个ETL应用程序。

* 1. 项目背景

目前单表超过10亿，总体数据有8T，且最近两年数据增长了50%，预计还会持续高速增长。

SqlServer数据仓库计算时间长已经超过6小时，并且存储和算力优化空间有限，性能已经不能支撑数据处理的需要。并且数据是近两年增加了60%，未来还会快速增加，需要更高、更灵活的存储能力、更高的算力。

* 1. 文档读者
* 企业数据仓库架构师
* 数据集成分析师
* 数据集成设计师
* 数据集成开发人员
  1. 参考资料

数据仓库架构文档

* 1. 名词与术语

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **术语名称** | **术语英文名** | **术语定义** | **别名** | **关键字** |
| 数据模型 | Data Model | 数据模型是对现实世界进行抽象的工具。在信息管理中需要将现实世界的事物及其有关特征转换为信息世界的数据才能对信息进行处理与管理，这就需要依靠数据模型作为这种转换的桥梁。 | 无 | 数据模型 |
| 逻辑数据模型 | Logical Data Model | 逻辑数据模型是对概念数据模型进一步的分解和细化。逻辑数据模型是根据业务规则确定的，关于业务对象、业务对象的数据项及业务对象之间关系的基本表示。 | 无 | 逻辑数据模型 |
| 物理数据模型 | Physical Data Model | 物理数据模型是在逻辑数据模型的基础上，考虑各种具体的数据库工具、技术等实现因素而进行设计，真正实现数据在数据库中的存放。物理数据模型的内容包括确定所有的表和列，定义主、外键用于表达表之间的关系，基于用户的需求可能进行反范式化设计等内容。 | 无 | 物理数据模型、表结构 |
| ETL | Extract/Transform/Load | 数据抽取转换加载 |  | ETL |

* 1. 约束

NA

1. ETL设计概述
   1. ETL设计总体考虑

我们ETL设计过程的主要考虑因素包括：

* 系统性能——选择合适的驱动方式与方法，最快的速度。ODBC写入性能极低，全部使用Greenplum Writer；Informatica PowerCenter “UPDTRANS”组件性能较低，使用GreenPlum Writer “Merge”方式
* 数据质量——数据源有效性、数据唯一性、数据精确性
* 时间要求——在窗口期内完成全部ETL。
* 可维护性——使用Informatica PowerCenter标准组件与标准SQL，增加可维护性
* 依赖关系——EBS系统接口表结构稳定
  1. ETL设计策略
* 在ETL过程1GB以下的中间数据文件采用文本文件；大于等于1GB的中间数据使用DB做交换
* 基于性能考虑，在ETL过程中暂不对数据仓库的表建立索引，将来是否建立索引，将根据ETL处理的时间窗口、ETL系统性能和数据查询性能要求再进行考虑；
* 对于EBS源表的抽取均与源表一致进行一对一的抽取，而不要事先在对源表进行Join后抽取Join结果；后继EDW与DM库的处理基于性能考虑，大部分使用DB的Join。
* 数据仓库加载的数据量相当大，为提高运行速度，数据加载尽量不使用逐条记录Insert的方法，而是利用GreenPlum Writer进行。

1. ETL应用框架设计
   1. ETL应用架构逻辑图

ETL应用架构从功能上被划分为三个层次：

* ETL管理调度，ETL管理调度层是实现ETL的作业包和基于作业包的作业调度，以及对ETL作业运行的监控功能；
* ETL应用功能，ETL应用功能模块层次的目标是具体实现从源到目标的ETL作业程序；
* ETL控制环境，ETL控制环境主是要为ETL的作业提供公共的控制信息。

ETL管理调度及ETL应用功能构成了ETL应用，而ETL控制环境则为ETL应用的每个应用程序提供支持，应用层次上独立的功能模块都通过更上一个层次的逻辑关系联系起来，使每个模块的功能更加清晰。

* 1. ETL模式

根据模型的设计和数据源的情况，有四种数据ETL模式：

* 完全刷新（Refresh，Type 1）：数据仓库数据表中只包括最新的数据，每次加载均删除原有数据，然后完全加载最新的源数据。一般用于当前快照表；
* 镜像增量（Snapshot Append，Type 2）：源数据中的记录定期更新，但记录中包括记录时间字段，源数据中保存了数据历史的记录，ETL可以通过记录时间将增量数据从源数据抽取出来以附加的方式加载到数据仓库中，数据的历史记录也会被保留在数据仓库中。如源系统的汇率历史表；
* 事件增量（Event Append，Type 3）：每一个记录是一个新的事件，相互之间没有必然的联系，新记录不是对原有记录数值的变更，记录包括时间字段，可以通过时间字段将新增数据抽取出来加载到数据库中。如源系统的成交表；
* 镜像比较（Snapshot Delta，Type 4）：数据仓库数据具有生效日期字段以保存数据的历史信息，而源数据不保留历史并且每天都可能被更新。如源系统的客户信息表。
  1. 数据抽取（Extract）

数据抽取（EXTRACT）的主要工作有：

* 数据范围过滤，完全抽取源表所有记录或按指定日期进行增量抽取；
* 抽取字段过滤，全部抽取源表所有字段或包括过滤掉不需要的源数据字段；
* 抽取条件过滤，如过虑到指定条件的记录；
* 数据排序，如按照抽取的指定字段进行排序
* 数据抽取可以采用PULL和PUSH两种方式（目标系统主动抽数，或源系统主动推送数据）。
  1. 数据转换（Transform）

数据转换（Transform）是按照目标表的数据结构，对一个或多个源数据的字段进行翻译、匹配、聚合等操作得到目标数据的字段。

数据转换主要包括格式和字段合并与拆分、数据翻译、数据匹配、数据聚合以及其他复杂计算等。

数据转换主要的工作有：

 格式变换，如所有日期格式统一为yyyymmdd；

 赋缺省值，在数据仓库中定义取值不为空的字段在源数据对应的字段可能存在没有取值的记录，这时根据业务需要，可能有两种处理办法，一是将该记录写入到Reject文件中，由业务部门根据Reject文件检查并修补源数据，另一种是在Convert阶段直接赋一个缺省值；

 类型变换, 如将源系统的Number类型转为Number(15,4)类型等。

* 1. 数据加载（Load）

经过数据转换生成的文件的结构与数据仓库数据表的结构完全一致，可以直接通过数据加载工具，以GreenPlum Writer的方式加载到数据仓库中。数据加载工作将分为如下3步进行。

* Pre-Load，在真正进行数据加载之前需要完成的准备工作；
* Load，主要完成将文件中的数据加载到数据仓库的表中；
* Post-Load，主要完成装载后的相应的处理工作。
  1. ETL数据质量检查

ETL过程保证数据加载质量的手段就是Record Balance和Amount Balance。

* Record Balance是指数据在每个步骤的处理中进和出的记录数应保持一致(有聚合操作除外)，这样才能保证不会遗漏数据，由于本项目是采用Informatica工具实施ETL，Record Balance是由Informatica工具来保证的，同时Informatica提供了监控界面可以查看记录的平衡情况；
* Amount Balance是指数据加载完成以后对加载完成的数据从业务角度进行稽核检查，相同的业务从不同的计算角度计算出来的结果应该是一致的，当然由于不同计算方法所带来的计算误差是客观存在的，但从业务上应该有一个可以容忍的误差范围，Amount Balance的实现采用开发单独的稽核程序并统一由Informatica来调度执行。
  1. ETL进程和进程调度

进程调度的功能比较单纯，就是在规定的时刻启动程序，并记录系统运行情况和运行结果。不同数据表的更新周期不同，因此，进程调度需要能够支持日周月等多种不同的启动周期，并通过设定启动时间来确定每个任务在何时启动运行。

日常调度每天凌晨3:00；

短信调度每天上午09:00；

销售数据多跑一次，每天中午12:15;

财务数据每月1—4号凌晨4点。

* 1. 日常数据、初始数据和历史数据ETL

初始数据加载、历史数据加载和日常数据加载在设计上采用相同的ETL策略，尽量共用部分开发的工作，但根据加载的数据量、数据源和工作流程的不同，在具体实现上将会有一些不同。

* 对于日常数据加载，由于需要定期自动运行相关的程序，因此需要有支持scheduler的工具支持以满足自动运行的需要，数据源主要是业务系统针对数据仓库系统提供的接口表，程序逻辑的处理要求非常严格。
* 对于初始数据加载，由于需要在上线日之前一次性完成，因此在上线日之前业务部门必须完成相关的数据清洗，从而保证初始数据加载的顺利完成，否则，将造成上线后数据仓库的数据质量不高。与日常数据加载不同，历史数据加载可能按不同的时段需要从分散在不同的数据表中加载数据，并可能需要不同的数据转换逻辑。
* 历史数据加载(History Load)是指将历史一定时间段的数据(主要是细节数据)加载到数据仓库。
  1. ETL中使用的日期

在ETL系统中使用到多个时间概念：

* 业务日期(busdate)：ETL系统运行所处理数据的本身的业务发生日期，该条记录有字段记录了该业务的发生日期日期；creation\_date，change\_on\_dt
* 系统日期（sysdate）：指当前系统的真实日期。 w\_insert\_dt, w\_update\_dt

1. 数据处理
   1. 维度表
      1. **客户表**

客户分组，业务手工维护

数据源：

计算逻辑：

* 客户整合需提供手工表，增加逻辑判断同一客户

原数仓对应的目标表和源码：

目标表：

源码：

20200107 新增BSC的客户表customer，同时提取客户的签约主体表signbody

源表：customer，signbody

数据源：bsc系统的gc\_bsc\_common数据库

CDH数仓表：这2张表，目前都先存储在ods层中；

备注：



计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **仓库表**

逻辑仓是在WMS进行配置的，但是库位维护是在OWMS进行维护的

数据源：WMS和OWMS

计算逻辑：

* 库位，分区，物理仓都维护好上层信息
* 国家，省份/州等地理信息直接维护到仓库和物理仓上面
* 存在部分逻辑仓没有物理仓，需进一步确定现状，若存在，建议业务维护补充

原数仓对应的目标表和源码：

目标表：

源码：

* + 1. **产品表**

数据源：BSC

计算逻辑：

* 需要将产品表和产品类目表进行合并
* 产品等级，产品附加表处理方式待定，针对出关，清关业务

原数仓对应的目标表和源码：

目标表：

源码：

* + 1. **员工表**

XXXX

数据源：业务系统（OMS，WMS，OWMS是一套，TMS和TOMS是另外一套）+手工（岗位和部门）

计算逻辑：

* 将主要的岗位和部门维护到员工表上
* 员工岗位和员工部门单独存储，优先级不高
* OMS，WMS，OWMS如何合并？

原数仓对应的目标表和源码：

目标表：

源码：

* + 1. **时间表（dim\_date）**

源表：dim\_date

数据源：984的gucang\_dw库的dim\_date

CDH数仓表：dim\_date

备注：

* 模型参考984系统的dim\_date表结构
* 表字段前缀：无

计算逻辑:

* 1，直接将984系统的dim\_date表直接一次性抽取到CDH的dw层，后期无需再进行二次抽取；

原数仓对应的目标表和源码：

* 目标表：dim\_date
* 源码：无
  + 1. **国家表（TCMS）**

源表：idd\_country

数据源：谷仓tcms

CDH数仓表：dim\_tcms\_country

备注：

* 模型参考谷仓tcms系统的idd\_country表结构
* 表字段前缀：country\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_country
* 源码：谷仓tcms-dim源码
  + 1. **城市表（TCMS）**

源表：idd\_city

数据源：谷仓tcms

CDH数仓表：dim\_tcms\_city

备注：

* 模型参考谷仓tcms系统的idd\_city表结构
* 表字段前缀：city\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_city
* 源码：谷仓tcms-dim源码
  + 1. **组织表（TCMS）**

源表：sys\_organization

数据源：谷仓tcms

CDH数仓表：dim\_tcms\_organization

备注：

* 模型参考谷仓tcms系统的sys\_organization表结构
* 该表作用相当于谷仓wms系统的仓库表，由于表结构的差异性，故重新设计一份
* 表字段前缀：organization\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_organization
* 源码：谷仓tcms-dim源码
  + 1. **产品表（TCMS）**

源表：csi\_productkind，xtd\_productgroup（产品组）

数据源：谷仓tcms

CDH数仓表：dim\_tcms\_product

备注：

* 模型主要参考谷仓tcms系统的csi\_productkind表结构，同时考虑将xtd\_productgroup中的name和code等字段整合进来
* 该表作用相当于谷仓wms系统的货运方式表，由于表结构和业务的差异性，故重新设计一份
* 表字段前缀：product\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_product，dim\_productgroup
* 源码：谷仓tcms-dim源码
  + 1. **客户表（TCMS）**

源表：csi\_customer

数据源：谷仓tcms

CDH数仓表：dim\_tcms\_customer

备注：

* 模型参考谷仓tcms系统的csi\_customer表结构
* 该表作用相当于谷仓wms系统的客户表，由于表结构和业务的差异性，故重新设计一份
* 表字段前缀：customer\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_customer
* 源码：谷仓tcms-dim源码
  + 1. **客户费用类表（ALL）**

源表：xtd\_customer\_feekind（tcms），xtd\_customer\_feekind\_group（tcms），fee\_type（wms，zywms）

数据源：谷仓tcms，谷仓wms，中邮wms

CDH数仓表：(1) dim\_fee\_type; (2) dim\_tcms\_fee\_type【只针对tcms系统】

备注：

* dim\_tcms\_fee\_type模型参考谷仓tcms系统的xtd\_customer\_feekind，同时将xtd\_customer\_feekind\_group中的name和code等字段设计进去
* dim\_fee\_type模型参考谷仓wms和中邮wms中的fee\_type结构
* 表的字段前缀统一为：**ft\_**

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_fee\_type(wms,zywms)，dim\_customer\_feekind(tcms)，dim\_customer\_feekind\_group(tcms)
* 源码：谷仓tcms-dim源码，谷仓wms-dim源码，中邮wms-dim源码
  + 1. **客户问题表（TCMS）**

源表：cts\_customer\_issuekind

数据源：谷仓tcms

CDH数仓表：(1) dim\_tcms\_issue\_type【只针对tcms系统】

备注：

* 模型参考谷仓tcms系统的cts\_customer\_issuekind结构
* 表的字段前缀统一为：issue\_type\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_issuekind
* 源码：谷仓tcms-dim源码
  + 1. **服务商表（ALL）**

源表：csi\_server(tcms), service\_provider(wms,zywms)

数据源：谷仓tcms, 谷仓wms，中邮wms

CDH数仓表：(1) dim\_server; (2) dim\_tcms\_server【只针对tcms系统】

备注：

* Dim\_tcms\_server模型参考谷仓tcms系统的csi\_server结构
* Dim\_server模型参考谷仓wms系统的service\_provider结构
* 表的字段前缀统一为：server\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_server
* 源码：谷仓tcms-dim源码，谷仓wms-dim源码，中邮wms-dim源码
  + 1. **服务商渠道表（ALL）**

源表：csi\_servechannel(tcms), sp\_service\_channel(wms,zywms)

数据源：谷仓tcms, 谷仓wms，中邮wms

CDH数仓表：（1）dim\_server\_channel；（2）dim\_tcms\_server\_channel

备注：

* dim\_tcms\_server\_channel模型参考谷仓tcms系统的csi\_servechannel结构
* dim\_server\_channel模型参考谷仓wms系统的sp\_service\_channel结构
* 表的字段前缀统一为：server\_channel\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_server\_channel
* 源码：谷仓tcms-dim源码，谷仓wms-dim源码，中邮wms-dim源码
  + 1. **员工表（TCMS）**

源表：hmr\_staff

数据源：谷仓tcms

CDH数仓表：dim\_tcms\_employee

备注：

* 模型参考谷仓tcms系统的hmr\_staff表结构
* 表的字段前缀统一为：employee\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_staff
* 源码：谷仓tcms-dim源码
  + 1. **员工业务分组表（TCMS）**

源表：wkg\_group,wkg\_grouptype（分组类型）

数据源：谷仓tcms

CDH数仓表：dim\_tcms\_work\_group

备注：

* 模型参考谷仓tcms系统的wkg\_group结构
* 表的字段前缀统一为：wgp\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_business\_group
* 源码：谷仓tcms-dim源码
  + 1. **分区表（ALL）**

源表：xtd\_zone\_division(tcms), sp\_zone\_charge(wms,zywms)

数据源：谷仓tcms, 谷仓wms，中邮wms

CDH数仓表：（1）dim\_zone；（2）dim\_tcms\_zone

备注：

* Dim\_tcms\_zone模型参考谷仓tcms系统的xtd\_zone\_division结构
* dim\_zone模型参考谷仓wms系统的sp\_zone\_charge结构
* 表的字段前缀统一为：zone\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_zone\_division(tcms), dim\_zne\_charge(wms,zywms)
* 源码：谷仓tcms-dim源码, 谷仓wms-dim源码，中邮wms-dim源码
  + 1. **分区方案表（ALL）**

源表：xtd\_zone\_scheme(tcms), sp\_zone\_scheme(wms,zywms)

数据源：谷仓tcms, 谷仓wms，中邮wms

CDH数仓表：（1）dim\_zone\_scheme；（2）dim\_tcms\_zone\_scheme

备注：

* dim\_tcms\_zone\_scheme模型参考谷仓tcms系统的xtd\_zone\_scheme结构
* dim\_zone\_scheme模型参考谷仓wms系统的sp\_zone\_scheme结构
* 表的字段前缀统一为：zs\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_zone\_scheme(tcms)
* 源码：谷仓tcms-dim源码
  + 1. **币别表（TCMS）**

源表：currency

数据源：谷仓tcms

CDH数仓表：（1）dim\_tcms\_currency

备注：

* 模型参考谷仓tcms系统的currency结构
* 表的字段前缀统一为：currency\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_currency
* 源码：谷仓tcms-dim源码
  + 1. **渠道账号表（LMS）**

源表：channel\_account

数据源：谷仓lms系统的lms\_portal库

CDH数仓表：dim\_lms\_channel\_account

备注：

* 模型参考谷仓lms系统的lms\_portal库的channel\_account表结构
* 表的字段前缀统一为：ca\_
* 类型，状态等字段的val值，直接参照表结构注释

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **渠道信息表（LMS）**

源表：channel\_info

数据源：谷仓lms系统的lms\_portal库

CDH数仓表：dim\_lms\_channel\_info

备注：

* 模型参考谷仓lms系统的lms\_portal库的channel\_info表结构
* 表的字段前缀统一为：ci\_
* 类型，状态等字段的val值，直接参照表结构注释

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **渠道服务商表（LMS）**

源表：channel\_service\_provider

数据源：谷仓lms系统的lms\_portal库

CDH数仓表：dim\_lms\_channel\_service\_provider

备注：

* 模型参考谷仓lms系统的lms\_portal库的channel\_service\_provider表结构
* 表的字段前缀统一为：csp\_
* 类型，状态等字段的val值，直接参照表结构注释

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  1. 事实表
     1. **入库单表**

数据源：OWMS

计算逻辑:

* XXX
* XXX

原数仓对应的目标表和源码：

目标表：

源码：

* + 1. **箱子明细表**

源表：new：gc\_receiving\_box，gc\_receiving\_box\_detail；old：receiving\_box，receiving\_box\_detail

数据源：谷仓wms，中邮wms

计算逻辑:

* 需将旧有逻辑的上架表合并到新表中
* 合并逻辑：如果新表中没有数据的话，则从旧表进行提取，可通过单号进行对比
* 在原数仓中新旧数据通过modular\_flag识别，如modular\_flag=’new’，代表是新表数据，反之modular\_flag=’old’则为旧表数据

备注：中邮的wms数据没有采用gc开头的新表逻辑

原数仓对应的目标表和源码：

* 目标表：fact\_receiving\_box
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **签收批次表**

源表：new：gc\_receiving\_box；old：receiving\_box

数据源：谷仓wms，中邮wms

计算逻辑:

* 需将旧有逻辑的上架表合并到新表中
* 合并逻辑：如果新表中没有数据的话，则从旧表进行提取，可通过单号进行对比
* 在原数仓中新旧数据通过modular\_flag识别，如modular\_flag=’new’，代表是新表数据，反之modular\_flag=’old’则为旧表数据

备注：中邮的wms数据没有采用gc开头的新表逻辑

原数仓对应的目标表和源码：

 目标表：fact\_receiving\_box

 源码：谷仓wms-fact源码，中邮wms-fact源码

* + 1. **入库清单表**

源表：new：gc\_receiving\_detail，gc\_receiving；old：receiving\_detail

数据源：谷仓wms，中邮wms

计算逻辑:

* 需将旧有逻辑的上架表合并到新表中
* 合并逻辑：如果新表中没有数据的话，则从旧表进行提取，可通过单号进行对比
* 在原数仓中新旧数据通过modular\_flag识别，如modular\_flag=’new’，代表是新表数据，反之modular\_flag=’old’则为旧表数据

备注：中邮的wms数据没有采用gc开头的新表逻辑

原数仓对应的目标表和源码：

 目标表：fact\_receiving

 源码：谷仓wms-fact源码，中邮wms-fact源码

* + 1. **收货批次表**

源表：new：gc\_receiving\_batch；old：receiving\_detail\_batch

数据源：谷仓wms，中邮wms

计算逻辑:

* 需将旧有逻辑的上架表合并到新表中
* 合并逻辑：如果新表中没有数据的话，则从旧表进行提取，可通过单号进行对比

备注：中邮的wms数据没有采用gc开头的新表逻辑

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **上架表/上架明细表**

源表：new：gc\_putaway\_detail；old：putaway\_detail，putaway

数据源：谷仓wms，中邮wms

计算逻辑:

* 需将旧有逻辑的上架表合并到新表中
* 合并逻辑：如果新表中没有数据的话，则从旧表进行提取，可通过单号进行对比
* 在原数仓中新旧数据通过modular\_flag识别，如modular\_flag=’new’，代表是新表数据，反之modular\_flag=’old’则为旧表数据

备注：中邮的wms数据没有采用gc开头的新表逻辑

原数仓对应的目标表和源码：

* 目标表：fact\_putaway
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **操作日志表**

源表：new：gc\_receiving\_log；old：receiving\_log

数据源：谷仓wms，中邮wms

计算逻辑:

* 需将旧有逻辑的上架表合并到新表中
* 合并逻辑：如果新表中没有数据的话，则从旧表进行提取，可通过单号进行对比

备注：中邮的wms数据没有采用gc开头的新表逻辑

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **订单表（订单模块）**

源表：orders

数据源：谷仓wms，中邮wms，谷仓oms

CDH数仓表：fact\_orders

备注：

* 谷仓oms中的订单表，同其他数据不一致，可单独建事实表
* 原数仓的fact\_orders是一个宽表，包含了orders，现在需将orders单独拆分出来
* 模型参考谷仓wms系统的orders表结构（可参照前天发送的业务系统元数据）;oms系统的订单参考oms系统的orders表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_orders，fact\_oms\_orders
* 源码：谷仓wms-fact源码，中邮wms-fact源码，谷仓oms-fact源码
  + 1. **订单表-OMS（订单模块）**

源表：orders

数据源：谷仓oms，中邮oms

CDH数仓表：fact\_oms\_orders

备注：

* 2020-5-11将中邮oms中的orders已经录入到fact\_oms\_orders
* 2020-5-11之前已经将，谷仓oms的orders数据录入到fact\_oms\_orders中了

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_oms\_orders
* 源码：谷仓oms-fact源码
  + 1. **订单产品表（订单模块）**

源表：order\_product

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_order\_product

备注：

* 原数仓的fact\_orders是一个宽表，包含了order\_product，现在需将order\_product单独拆分出来
* 模型参考谷仓wms系统的order\_product表结构（可参照前天发送的业务系统元数据）

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_orders
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **订单操作时间表（订单模块）**

源表：order\_operation\_time

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_order\_operation\_time

备注：

* 原数仓的fact\_orders是一个宽表，包含了order\_operation\_time，现在需将order\_operation\_time单独拆分出来
* 模型参考谷仓wms系统的order\_operation\_time表结构（可参照前天发送的业务系统元数据）
* 2020-5-11 目前该数据源是参考owms的

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_orders
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **订单运输信息表（订单模块）**

源表：ship\_order

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_ship\_order

备注：

* 原数仓的fact\_orders是一个宽表，包含了order\_operation\_time，现在需将order\_operation\_time单独拆分出来
* 模型参考谷仓wms系统的order\_operation\_time表结构（可参照前天发送的业务系统元数据）

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_orders
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **订单地址信息表（订单模块）**

源表：order\_address\_book

数据源：谷仓wms，中邮wms，谷仓owms，中邮owms

CDH数仓表：fact\_order\_address\_book

备注：

* 模型参考谷仓wms系统的order\_address\_book表结构（可参照前天发送的业务系统元数据）

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_order\_address\_book
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **订单日志表（订单模块）**

源表：order\_log

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_order\_log

备注：

* 模型参考谷仓wms系统的order\_log表结构（可参照前天发送的业务系统元数据）

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **订单物理仓表（订单模块）**

源表：order\_physical\_relation

数据源：谷仓wms

CDH数仓表：fact\_order\_physical

备注：

* 模型参考谷仓wms系统的order\_physical\_relation表结构（可参照前天发送的业务系统元数据）

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_order\_physical
* 源码：谷仓wms-fact源码
  + 1. **订单产品物理仓表（订单模块）**

源表：order\_product\_physical\_relation

数据源：谷仓wms

CDH数仓表：fact\_order\_product\_physical

备注：

* 模型参考谷仓wms系统的order\_product\_physical\_relation表结构（可参照前天发送的业务系统元数据）

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_order\_product\_physical
* 源码：谷仓wms-fact源码
  + 1. **问题件表（订单模块）**

源表：orders

数据源：谷仓owms，中邮owms

CDH数仓表：fact\_oversea\_abnormal\_order

备注：

* 模型参考984的gucang\_dw系统的fact\_oversea\_abnormal\_order表结构（可参照gucang\_dw的pdm文件）

计算逻辑:

* 详见“谷仓owms-fact源码，中邮owms-fact源码”等源码

原数仓对应的目标表和源码：

* 目标表：fact\_oversea\_abnormal\_order
* 源码：谷仓owms-fact源码，中邮owms-fact源码
  + 1. **退件单表（订单模块）**

源表：return\_orders，after\_sales\_return\_orders，return\_order\_product，after\_sales\_return\_order\_product

数据源：谷仓wms，中邮wms，谷仓oms

CDH数仓表：fact\_return\_orders

备注：

* 模型参考984的gucang\_dw系统的fact\_return\_orders\_union表结构
* dw层的fact\_return\_orders表来源984谷仓wms-fact的fact\_after\_sales\_return\_orders和中邮wms-fact的fact\_return\_orders表，相应的对照关系参考源码
* Ods层中，各wms系统的return\_orders，after\_sales\_return\_orders，return\_order\_product，after\_sales\_return\_order\_product需录入；各oms系统的目前只有return\_orders，return\_order\_product等表；
* 字段前缀：ro\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_return\_orders\_union, fact\_return\_orders, fact\_after\_sales\_return\_orders
* 源码：谷仓oms-fact源码，中邮wms-fact源码，谷仓wms-fact源码
  + 1. **退件产品批次表（订单模块）**

源表：after\_sales\_product\_batch

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_after\_sales\_product\_batch

备注：

* 模型参考谷仓wms系统的after\_sales\_product\_batch表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **退件产品照片表（订单模块）**

源表：after\_sales\_product\_photos

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_after\_sales\_product\_photos

备注：

* 模型参考谷仓wms系统的after\_sales\_product\_photos表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **退件地址信息表（订单模块）**

源表：after\_sales\_return\_orders\_address

数据源：谷仓wms

CDH数仓表：fact\_after\_sales\_return\_orders\_address

备注：

* 模型参考谷仓wms系统的after\_sales\_return\_orders\_address表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **退件费用表（订单模块）**

源表：return\_order\_fee，bil\_income，bil\_business\_attach

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_return\_order\_fee，fact\_bil\_income，fact\_bil\_business\_attach

备注：

* 本表主要设计三个模型：

1. 针对中邮wms系统设计fact\_return\_order\_fee，模型参照谷仓wms系统的return\_order\_fee表结构；
2. 针对谷仓wms系统，中邮wms系统设计fact\_bil\_income，fact\_bil\_business\_attach，模型参考

谷仓wms系统的bil\_income，bil\_business\_attach；

* 针对第2,3的模型的设计，主要是因为谷仓wms系统在采用新的退件表after\_sale\_return\_orders后，相应的费用数据是保存在bil\_income表中，同原先的差异很大；同时由于bil\_income表如果要跟after\_sale\_return\_orders关联需要通过单号进行关联，故同时得设计fact\_bil\_business\_attach表；
* 那为何中邮的也考虑了fact\_bil\_income，fact\_bil\_business\_attach模型，主要是因为这两个表后期作为财务模块那边的收入需使用到；

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_return\_order\_fee
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **退件日志表（订单模块）**

源表：after\_sales\_return\_orders\_log

数据源：谷仓wms

CDH数仓表：fact\_after\_sales\_return\_orders\_log

备注：

* 模型参考谷仓wms系统的after\_sales\_return\_orders\_log表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **售后下架表（订单模块）**

源表：after\_sales\_picking

数据源：谷仓wms

CDH数仓表：fact\_after\_sales\_picking

备注：

* 模型参考谷仓wms系统的after\_sales\_pikcing表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **售后下架明细表（订单模块）**

源表：after\_sales\_picking\_detail

数据源：谷仓wms

CDH数仓表：fact\_after\_sales\_picking\_detail

备注：

* 模型参考谷仓wms系统的after\_sales\_pikcing\_detail表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **售后质检表（订单模块）**

源表：after\_sales\_qc

数据源：谷仓wms

CDH数仓表：fact\_after\_sales\_qc

备注：

* 模型参考谷仓wms系统的after\_sales\_qc表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **售后质检明细表（订单模块）**

源表：after\_sales\_qc\_detail

数据源：谷仓wms

CDH数仓表：fact\_after\_sales\_qc\_detail

备注：

* 模型参考谷仓wms系统的after\_sales\_qc\_detail表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **售后质检上架明细表（订单模块）**

源表：after\_sales\_putaway\_detail

数据源：谷仓wms

CDH数仓表：fact\_after\_sales\_putaway\_detail

备注：

* 模型参考谷仓wms系统的after\_sales\_putaway\_detail表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **预下架明细表（出库模块）**

源表：advance\_picking\_detail

数据源：谷仓owms，中邮owms

CDH数仓表：fact\_advance\_picking\_detail

备注：

* 模型参考谷仓owms系统的advance\_picking\_detail表结构

计算逻辑:

* 2019.11.23 将该表由原先的wms来源改为owms

原数仓对应的目标表和源码：

* 目标表：fact\_advance\_picking\_detail
* 源码：谷仓wms-fact源码
  + 1. **下架表（出库模块）**

源表：picking

数据源：谷仓wms，中邮wms，谷仓owms，中邮owms

CDH数仓表：fact\_picking

备注：

* 模型参考谷仓owms系统的picking表结构
* Fact\_picking表录入谷仓owms，中邮owms的数据
* 在ods层中4个数据源中的picking数据都需保存，以便后期调整fact\_picking表的时候，可以就近获取

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_picking
* 源码：谷仓wms-fact源码，中邮wms-fact源码，谷仓owms-fact源码，中邮owms-fact源码
  + 1. **下架明细表（出库模块）**

源表：picking\_detail

数据源：谷仓wms，中邮wms，谷仓owms，中邮owms

CDH数仓表：fact\_picking\_detail

备注：

* 模型参考谷仓owms系统的picking\_detail表结构
* Fact\_picking\_detail表录入谷仓owms，中邮owms的数据
* 在ods层中4个数据源中的picking\_detail数据都需保存，以便后期调整fact\_picking\_detail表的时候，可以就近获取

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_picking
* 源码：谷仓wms-fact源码，中邮wms-fact源码，谷仓owms-fact源码，中邮owms-fact源码
  + 1. **波次分区表（出库模块）**

源表：wellen\_area

数据源：谷仓owms，中邮owms

CDH数仓表：fact\_wellen\_area

备注：

* 模型参考谷仓owms系统的wellen\_area表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_wellen\_area
* 源码：谷仓owms-fact源码，中邮owms-fact源码
  + 1. **波次规则日志表（出库模块）**

源表：wellen\_log，new\_wellen\_log（注意：谷仓owms才有该表）

数据源：谷仓owms，中邮owms

CDH数仓表：fact\_wellen\_log

备注：

* 模型参考谷仓owms系统的wellen\_log表结构
* new\_wellen\_log表目前只存在于谷仓owms系统中，该表只需先导入数据放在ods层即可；

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_wellen\_log
* 源码：谷仓owms-fact源码，中邮owms-fact源码
  + 1. **波次规则表（出库模块）**

源表：wellen\_rule，new\_wellen\_rule（注意：谷仓owms才有该表）

数据源：谷仓owms，中邮owms

CDH数仓表：fact\_wellen\_rule

备注：

* 模型参考谷仓owms系统的wellen\_rule表结构
* new\_wellen\_rule表目前只存在于谷仓owms系统中，该表只需先导入数据放在ods层即可；

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_wellen\_rule
* 源码：谷仓owms-fact源码，中邮owms-fact源码
  + 1. **波次渠道表（出库模块）**

源表：wellen\_sc

数据源：谷仓owms，中邮owms

CDH数仓表：fact\_wellen\_sc

备注：

* 模型参考谷仓owms系统的wellen\_sc表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_wellen\_sc
* 源码：谷仓owms-fact源码，中邮owms-fact源码
  + 1. **质检表（入库模块）**

源表：quality\_control

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_quality\_control

备注：

* 模型参考中邮wms系统的quality\_control表结构
* 质检表更新主要分两种情况：

1. 谷仓wms质检表已无进行更新，故数据只需在ods层的时候一次性录入，后期无需进行；
2. 中邮wms质检表需实时进行正常更新数据

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_quality\_control
* 源码：谷仓wms-fact源码，中邮wms-fact源码
  + 1. **拣货物理仓对照表（出库模块）**

源表：picking\_physical\_relation

数据源：谷仓owms

CDH数仓表：fact\_picking\_physical

备注：

* 模型参考谷仓owms系统的picking\_physical\_relation表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **网上订单收货信息表（转运模块）**

源表：csd\_shipperconsignee

数据源：谷仓tcms中的goodcang\_toms\_web数据库

CDH数仓表：fact\_tcms\_csd\_shipperconsignee

备注：

* 模型参考谷仓toms系统的csd\_shipperconsignee表结构
* 模型字段主要取值consignee开头的字段和order\_id字段
* 字段前缀：consignee\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fat\_web\_order
* 源码：谷仓tcms-fact源码
  + 1. **运单表（转运模块）**

源表：bsn\_business

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_business

备注：

* 模型参考谷仓tcms系统的bsn\_business表结构
* 字段前缀：business\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_business\_waybill
* 源码：谷仓tcms-fact源码
  + 1. **运单轨迹表（转运模块）**

源表：bsn\_business\_operator

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_business\_operator

备注：

* 模型参考谷仓tcms系统的bsn\_business\_operator表结构
* 字段前缀：bo\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_business\_operator
* 源码：缺失
  + 1. **运单附加信息表（转运模块）**

源表：bsn\_expressexport

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_business\_expressexport

备注：

* 模型参考谷仓tcms系统的bsn\_expressexport表结构
* 字段前缀：be\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_business\_expressexport
* 源码：缺失
  + 1. **入货材积表（转运模块）**

源表：bsn\_cargovolume

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_business\_checkin\_cargovolume

备注：

* 模型参考谷仓tcms系统的bsn\_cargovolume表结构
* 字段前缀：involume\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_business\_checkin\_cargovolume
* 源码：谷仓tcms-fact源码
  + 1. **出货材积表（转运模块）**

源表：bsn\_checkout\_cargovolume

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_business\_checkout\_cargovolume

备注：

* 模型参考谷仓tcms系统的bsn\_checkout\_cargovolume表结构
* 字段前缀：outvolume\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_business\_checkout\_cargovolume
* 源码：谷仓tcms-fact源码
  + 1. **出货表（转运模块）**

源表：bsn\_departurebatch

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_business\_departurebatch

备注：

* 模型参考谷仓tcms系统的bsn\_departurebatch表结构
* 字段前缀：departbatch\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_business\_departurebatch
* 源码：缺失
  + 1. **到货表（转运模块）**

源表：bsn\_arrivalbatch

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_business\_arrivalbatch

备注：

* 模型参考谷仓tcms系统的bsn\_arrivalbatch表结构
* 字段前缀:arrivalbatch\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_business\_arrivalbatch
* 源码：缺失
  + 1. **转运问题件表（转运模块）**

源表：cts\_issue

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_issue

备注：

* 模型参考谷仓tcms系统的cts\_issue表结构
* 字段前缀：issue\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_customer\_issue
* 源码：缺失
  + 1. **异常订单表（转运模块）**

源表：order\_abnormal

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_order\_abnormal

备注：

* 模型参考谷仓tcms系统的order\_abnormal表结构
* 字段前缀：oab\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **退件订单表（转运模块）**

源表：order\_return

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_order\_ return

备注：

* 模型参考谷仓tcms系统的order\_return表结构
* 字段前缀：ro\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_order\_return
* 源码：谷仓tcms-fact源码
  + 1. **复核订单表（转运模块）**

源表：order\_review

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_order\_review

备注：

* 模型参考谷仓tcms系统的order\_review表结构
* 字段前缀：orv\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_order\_review
* 源码：谷仓tcms-fact源码
  + 1. **取消订单表（转运模块）**

源表：order\_cancel

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_order\_cancel

备注：

* 模型参考谷仓tcms系统的order\_cancel表结构
* 字段前缀：oc\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. 库存批次表**（仓储模块）**

源表：inventory\_batch

数据源：谷仓wms，中邮wms，谷仓owms

CDH数仓表：fact\_inventory\_batch

备注：

* 模型参考谷仓wms系统的fact\_inventory\_batch表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_inventory\_batch
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **批次日志表（仓储模块）**

源表：inventory\_batch\_log,inventory\_batch\_log\_wxx\_mxx

数据源：谷仓wms,谷仓wms\_ibl，中邮wms，谷仓oms(无数据)，谷仓owms

CDH数仓表：fact\_inventory\_batch

备注：

* 模型参考谷仓wms系统的 inventory\_batch\_log表结构
* 谷仓wms的inventory\_batch\_log只导一次，实时更新，只同步谷仓wms\_ibl库和中邮wms的表
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_inventory\_batch\_log
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **产品库存（仓储模块）**

源表：product\_inventory

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_product\_inventory

备注：

* 模型参考谷仓wms系统的 product\_inventory表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_product\_inventory
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **差异单表（仓储模块）**

源表：inventory\_difference

数据源：谷仓wms，谷仓owms

CDH数仓表：fact\_inventory\_difference

备注：

* 模型参考谷仓wms系统的 inventory\_difference表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_inventory\_difference

源码：谷仓wms-fact源码

* + 1. **差异单明细表（仓储模块）**

源表：inventory\_difference\_detail

数据源：谷仓wms，谷仓owms

CDH数仓表：fact\_inventory\_difference\_detail

备注：

* 模型参考谷仓wms系统的 inventory\_difference\_detail表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_inventory\_difference

源码：谷仓wms-fact源码

* + 1. **流转量表（仓储模块）**

源表：flow\_volume

数据源：谷仓wms，谷仓owms

CDH数仓表：fact\_flow\_volume

备注：

* 模型参考谷仓wms系统的 flow\_volume表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_flow\_volume

源码：谷仓wms-fact源码

* + 1. **盘点表（仓储模块）**

源表：take\_stock

数据源：谷仓wms，中邮wms，谷仓owms

CDH数仓表：fact\_take\_stock

备注：

* 模型参考谷仓wms系统的 take\_stock表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_take\_stock

源码：谷仓wms-fact源码,中邮wms-fact源码

* + 1. **盘点项（仓储模块）**

源表：take\_stock\_item

数据源：谷仓wms，中邮wms，谷仓owms

CDH数仓表：fact\_take\_stock\_item

备注：

* 模型参考谷仓wms系统的 take\_stock\_item表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_take\_stock

源码：谷仓wms-fact源码,中邮wms-fact源码

* + 1. **盘点任务分配用户对照表（仓储模块）**

源表：take\_stock\_assignment

数据源：谷仓wms，中邮wms，谷仓owms

CDH数仓表：fact\_take\_stock\_assignment

备注：

* 模型参考谷仓wms系统的 take\_stock\_assignment表结构
* 谷仓owms目前只存储在ods中

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_take\_stock
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **产品库位对照表(仓储模块)**

源表：product\_location\_map

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_product\_warehouse\_location\_map

备注：

* 模型参考谷仓wms系统的 product\_location\_map表结构
* 字段前缀：pwlcm\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_product\_wl\_map
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **结算业务表（仓储：财务模块）**

源表：bil\_business

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_bill\_business

备注：

* 模型参考谷仓wms系统的 bil\_business表结构
* 字段前缀：bb\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_bill\_income
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **结算业务附加信息表（仓储：财务模块）**

源表：bil\_business\_attach

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_bill\_business\_attach

备注：

* 模型参考谷仓wms系统的 bil\_business\_attach表结构
* 字段前缀：bba\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_bill\_income
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **应收费用表（仓储：财务模块）**

源表：bil\_income

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_bill\_income

备注：

* 模型参考谷仓wms系统的 bil\_income表结构
* 字段前缀：bi\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_bill\_income
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **应收附加信息表（仓储：财务模块）**

源表：bil\_income\_attach

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_bill\_income\_attach

备注：

* 模型参考谷仓wms系统的 bil\_income\_attach表结构
* 字段前缀：bia\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_bill\_income
* 源码：谷仓wms-fact源码,中邮wms-fact源码
  + 1. **客户账单表（仓储：财务模块）**

1:

源表：bil\_customer\_bill(wms)

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_bill\_customer\_bill

备注：

* 模型参考谷仓wms系统的 bil\_customer\_bill表结构
* 字段前缀：bcb\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_bill\_customer\_bill
* 源码：谷仓wms-fact源码,中邮wms-fact源码

2:

源表：warehouseorder(bsc)

数据源：谷仓bsc中gc\_bsc\_amc\_开头的数据库

CDH数仓表：fact\_warehouseorder

备注：

* 模型参考谷仓bsc系统的 warehouseorder表结构
* 字段前缀：wo\_
* 后期wms的客户账单等数据以bsc的为准，截止12月15号，还有1-2个客户的数据未从原客户账单表bil\_customer\_bill等表切换到bsc中，中邮的继续保留原有的业务逻辑形态
* 字段datasource\_num\_id在dw层中需用谷仓wms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_bsc\_warehouseorder
* 源码：谷仓wms-fact源码
  + 1. **客户账单明细表（仓储：财务模块）**

1:

源表：bil\_customer\_bill\_fee (wms)

数据源：谷仓wms，中邮wms

CDH数仓表：fact\_bill\_customer\_bill\_fee

备注：

* 模型参考谷仓wms系统的 bil\_customer\_bill\_fee表结构
* 字段前缀：bcbf\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_bill\_customer\_bill
* 源码：谷仓wms-fact源码,中邮wms-fact源码

2:

源表：warehouseorderdetails(bsc)

数据源：谷仓bsc中gc\_bsc\_amc\_开头的数据库

CDH数仓表：fact\_warehouseorder\_details

备注：

* 模型参考谷仓bsc系统的 warehouseorderdetails表结构
* 字段前缀：wod\_
* 后期wms的客户账单等数据以bsc的为准，截止12月15号，还有1-2个客户的数据未从原客户账单表bil\_customer\_bill等表切换到bsc中，中邮的继续保留原有的业务逻辑形态
* 字段datasource\_num\_id在dw层中需用谷仓wms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_bsc\_warehouseorderdetails
* 源码：谷仓wms-fact源码
  + 1. **客户账单流水表（仓储：财务模块）**

源表：warehousbillflow(bsc)

数据源：谷仓bsc中gc\_bsc\_amc\_开头的数据库

CDH数仓表：fact\_warehousebillflow

备注：

* 模型参考谷仓bsc系统的 warehousbillflow表结构
* 字段前缀：wod\_
* 后期wms的客户账单等数据以bsc的为准，截止12月15号，还有1-2个客户的数据未从原客户账单表bil\_customer\_bill等表切换到bsc中，中邮的继续保留原有的业务逻辑形态
* 字段datasource\_num\_id在dw层中需用谷仓wms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_bsc\_warehousebillflow
* 源码：谷仓wms-fact源码
  + 1. **仓租表（仓储：财务模块）**

1:仓租单据表

源表：wh\_inventory\_storage

数据源：谷仓wms, 中邮wms

CDH数仓表：fact\_wh\_inventory\_storage

备注：

* 模型参考谷仓wms系统的 wh\_inventory\_storage表结构
* 字段前缀：wis\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_wh\_inventory\_storage
* 源码：谷仓wms-fact源码

2：仓租快照表

源表：wh\_inventory\_storage\_snapshot

数据源：谷仓wms, 中邮wms

CDH数仓表：fact\_wh\_inventory\_storage\_snapshot

备注：

* 模型参考谷仓wms系统的 wh\_inventory\_storage\_snapshot表结构
* 字段前缀：wiss\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_wh\_inventory\_storage
* 源码：谷仓wms-fact源码

3：库存快照明细批次表

源表：wh\_inventory\_snapshot\_detail\_batch

数据源：谷仓wms, 中邮wms

CDH数仓表：fact\_wh\_inventory\_snapshot\_detail\_batch

备注：

* 模型参考谷仓wms系统的 wh\_inventory\_snapshot\_detail\_batch表结构
* 字段前缀：wisdb\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_wh\_inventory\_storage
* 源码：谷仓wms-fact源码
  + 1. **应收费用表（转运：财务模块）**

源表：bil\_income

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_bill\_income

备注：

* 模型参考谷仓tcms系统的 bil\_income表结构
* 字段前缀：bi\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_waybill\_income
* 源码：谷仓tcms-fact源码
  + 1. **应收附加信息表（转运：财务模块）**

源表：bil\_incomeattach

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_bill\_income\_attach

备注：

* 模型参考谷仓tcms系统的 bil\_incomeattach表结构
* 字段前缀：bia\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_waybill\_income
* 源码：谷仓tcms-fact源码
  + 1. **客户账单表（转运：财务模块）**

源表：bil\_shipperbill

数据源：谷仓tcms的goodcang\_settlement\_center

CDH数仓表：fact\_tcms\_bill\_shipperbill

备注：

* 模型参考谷仓tcms的goodcang\_settlement\_center库的 bil\_shipperbill表结构
* 字段bill\_status的值在bil\_shipperbill\_status表中
* 字段前缀：sb\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **客户账单明细表（转运：财务模块）**

源表：bil\_shipperbill\_income\_前缀的表

数据源：谷仓tcms的goodcang\_settlement\_center

CDH数仓表：fact\_tcms\_bill\_shipperbill\_income

备注：

* 模型参考谷仓tcms的goodcang\_settlement\_center库的 bil\_shipperbill\_income表结构
* 字段前缀：sbi\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **已出账单同账单明细对照表（转运：财务模块）**

源表：bil\_shipperbill\_mapping

数据源：谷仓tcms的goodcang\_settlement\_center

CDH数仓表：fact\_tcms\_bill\_shipperbill\_mapping

备注：

* 模型参考谷仓tcms的goodcang\_settlement\_center库的 bil\_shipperbill\_mapping表结构
* 字段前缀：sbm\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **应付费用表（转运：财务模块）**

源表：bil\_payment

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_bill\_payment

备注：

* 模型参考谷仓tcms系统的 bil\_payment表结构
* 字段前缀：bp\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_waybill\_payment
* 源码：谷仓tcms-fact源码
  + 1. **服务商账单表（转运：财务模块）**

源表：bil\_servebill

数据源：谷仓tcms的goodcang\_settlement\_center\_payment

CDH数仓表：fact\_tcms\_bill\_serverbill

备注：

* 模型参考谷仓tcms的goodcang\_settlement\_center\_payment库的 bil\_servebill表结构
* 字段前缀：sb\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **服务商账单明细表（转运：财务模块）**

源表：bil\_servebillhawb\_开头的表

数据源：谷仓tcms的goodcang\_settlement\_center\_payment

CDH数仓表：fact\_tcms\_bill\_serverbillhawb

备注：

* 模型参考谷仓tcms的goodcang\_settlement\_center\_payment库的 bil\_servebillhawb表结构
* 字段前缀：sbh\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **服务商账单明细费用表（转运：财务模块）**

源表：bil\_servebillhawb\_fee\_开头的表

数据源：谷仓tcms的goodcang\_settlement\_center\_payment

CDH数仓表：fact\_tcms\_bill\_serverbillhawb\_fee

备注：

* 模型参考谷仓tcms的库的goodcang\_settlement\_center\_payment的 bil\_servebillhawb\_fee表结构
* 字段前缀：sbhf\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **服务商账单业务订单映射表（转运：财务模块）**

源表：bil\_servebill\_business\_mapping\_开头的表

数据源：谷仓tcms的goodcang\_settlement\_center\_payment

CDH数仓表：fact\_tcms\_bill\_serverbill\_business\_mapping

备注：

* 模型参考谷仓tcms的库的goodcang\_settlement\_center\_payment的 bil\_servebill\_business\_mapping表结构
* 字段前缀：sbm\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **退件收入表（转运：财务模块）**

源表：bil\_return\_income

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_bill\_return\_income

备注：

* 模型参考谷仓tcms系统的 bil\_return\_income表结构
* 字段前缀：bri\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_waybill\_return\_income
* 源码：谷仓tcms-fact源码
  + 1. **退件收入附加信息表（转运：财务模块）**

源表：bil\_return\_incomeattach

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_bill\_return\_income\_attach

备注：

* 模型参考谷仓tcms系统的 bil\_return\_incomeattach表结构
* 字段前缀：bria\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_waybill\_return\_incomeattach
* 源码：谷仓tcms-fact源码
  + 1. **退件应付表（转运：财务模块）**

源表：bil\_return\_payment

数据源：谷仓tcms

CDH数仓表：fact\_tcms\_bill\_return\_payment

备注：

* 模型参考谷仓tcms系统的 bil\_return\_payment表结构
* 字段前缀：brp\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **交运容器明细表（仓储：交运模块）**

源表：container\_details

数据源：谷仓owms库中的ts开头的数据库

CDH数仓表：fact\_container\_details

备注：

* 模型参考谷仓owms系统ts\_au库的 container\_details表结构
* 字段前缀：cd\_
* 从ods映射到dw层后，需将data\_source\_num\_id改为‘谷仓wms’

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：
* 源码：
  + 1. **订单装箱表（仓储：订单模块）**

源表：order\_box

数据源：谷仓wms，zy\_wms

CDH数仓表：fact\_order\_box

备注：

* 模型参考谷仓wms系统的 order\_box表结构
* 字段前缀：ob\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_order\_box
* 源码：谷仓wms-fact源码
  + 1. **订单装箱明细表（仓储：订单模块）**

源表：order\_box\_detail

数据源：谷仓wms，zy\_wms

CDH数仓表：fact\_order\_box\_detail

备注：

* 模型参考谷仓wms系统的 order\_box\_detail表结构
* 字段前缀：obd\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_order\_detail
* 源码：谷仓wms-fact源码
  + 1. **历史库存表（仓储：仓储模块）**

源表：tmp\_zt136ProductInventoryHistory

数据源：谷仓wm

CDH数仓表：fact\_product\_inventory\_history

备注：

* 模型参考cdh系统的ods.gc\_wms\_inventory\_batch表结构
* 字段前缀：pih\_
* fact\_product\_inventory\_history表中，需要添加pih\_date\_key字段，该字段用于存储每日提取的日期，以便进行库存分析的时候，可以快速定位每天的库存量

计算逻辑:

* 1，每日0点30分从ods.gc\_wms\_inventory\_batch，ods.zy\_wms\_inventory\_batch将表数据全量提取到fact\_product\_inventory\_history表中，相当于每天对这两份表做一个备份的机制
* 2，历史数据从tmp\_zt136ProductInventoryHistory进行拷贝过来
* 3，fact\_product\_inventory\_history表一般保留最近1年数据，其他的数据视情况进行归档备份

原数仓对应的目标表和源码：

* 目标表：tmp\_zt136ProductInventoryHistory
* 源码：
  + 1. **订单费用日志表（仓储：订单模块）**

源表：order\_fee\_log

数据源：谷仓wm，中邮wms

CDH数仓表：fact\_order\_fee\_log

备注：

* 模型参考谷仓wms系统的order\_fee\_log表结构
* 字段前缀：ofl\_

计算逻辑:

* 在fact\_order\_fee\_log表中将ofl\_content字段转换为ofl\_logistics\_price\_grade【物流价格等级】, ofl\_zone\_schema【分区方案】, ofl\_special\_grade【特殊等级】

原数仓对应的目标表和源码：

* 目标表：
* 源码
  + 1. **客户订单表（仓储：转运模块）**

源表：csd\_order

数据源：谷仓tcms中的goodcang\_toms\_web数据库

CDH数仓表：fact\_tcms\_csd\_order

备注：

* 模型参考谷仓tcms系统的csd\_order表结构
* 字段前缀：order\_
* 字段datasource\_num\_id在dw层中需用谷仓tcms系统的编码，其他照旧

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_web\_order
* 源码：谷仓tmcs-fact源码
  + 1. **未出账订单表（仓储：转运模块）**

源表：uninvoice\_order

数据源：谷仓tcms中的goodcang\_toms数据库

CDH数仓表：fact\_tcms\_uninvoice\_order

备注：

* 模型参考谷仓tcms系统的uninvoice\_order表结构
* 字段前缀：uod\_
* 由于exception\_type，audit\_status未找到相应对照表，故在fact\_tcms\_uninvoice\_order暂时不体现这2个字段的val
* 诸如customer\_id, og\_id, serverchannel\_id等字段，暂时不需要提供相应的key字段，直接将原来的字段直接映射到fact\_tcms\_uninvoice\_order就可以。

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **LMS查询订单优化表（仓储：谷仓LMS模块）**

源表：query\_order\_info\_optimized

数据源：谷仓lms中的lms\_orderservice\_20开头的数据库

CDH数仓表：fact\_lms\_query\_order\_info\_optimized

备注：

* 模型参考谷仓lms系统的query\_order\_info\_optimized表结构
* 字段前缀：oio\_
* 分区字段：optimized\_date

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **工单订单表（仓储：谷仓工单模块）**

源表：work\_order

数据源：谷仓oms中的wos\_goodcang\_com的数据库

CDH数仓表：fact\_work\_orders

备注：

* 模型参考谷仓oms系统中wos\_goodcang\_com数据库的work\_order表结构
* 字段前缀：wo\_
* 分区字段：

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **工单问题类型表（仓储：谷仓工单模块）**

源表：feedback\_question

数据源：谷仓oms中的wos\_goodcang\_com的数据库

CDH数仓表：par\_work\_feedback\_question

备注：

* 模型参考谷仓oms系统中wos\_goodcang\_com数据库的feedback\_question表结构
* 字段前缀：
* 分区字段：

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **工单环节耗时配置表（仓储：谷仓工单模块）**

源表：segment\_time\_config

数据源：谷仓oms中的wos\_goodcang\_com的数据库

CDH数仓表：par\_work\_segment\_time\_config

备注：

* 模型参考谷仓oms系统中wos\_goodcang\_com数据库的segment\_time\_config表结构
* 字段前缀：
* 分区字段：

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：无
* 源码：无
  + 1. **物理仓库存表（仓储：仓储模块）**

源表：warehouse\_physical\_inventory

数据源：谷仓wms

CDH数仓表：fact\_warehouse\_physical\_inventory

备注：

* 模型参考谷仓wms系统的 warehouse\_physical\_inventory表结构
* 字段前缀：wpi\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_warehouse\_physical\_inventory
* 源码：谷仓wms-fact源码

1. 通用逻辑处理
   1. 值列表(par\_val\_list)

维护值列表信息，建议在DW中同时存储code和value；

若修改value，历史数据可考虑刷数的方式实现；

1. Tcms常见值列表(以下仅供参考，非全部)

（1）tcms：atd\_paymentmode

（2）tcms：atd\_cargo\_type

（3）tcms：bil\_charge\_node

（4）tcms：atd\_documentarystatus

（5）tcms：atd\_mail\_cargo\_type

（6）tcms：bsd\_operationstatus

（7）tcms：atd\_returnstatus

（8）tcms：atd\_transferstatus

（9）tcms：atd\_billing\_unittype

（10）tcms：goodcang\_toms.bsd\_servebillstatus

（11）tcms：wkg\_grouptype

* 1. 节假日（par\_holiday）

处理节假日（区分国家地区），处理周的定义（哪天为第一天）

* 1. 币种汇率（par\_currency\_rate）

原币种，原汇率，原币种金额，转化为本币，本币的金额（一直为人民币），记录到DW中

1，Tcms系统

源表：prc\_currency\_rate

数据源：谷仓tcms

CDH数仓表：par\_currency\_rate

备注：

* 模型参考CDH系统的par\_currency\_rate表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_currency\_rate
* 源码：谷仓tcms-dim源码
  1. 时间(par\_timezone)

给予配置表规则计算，处理冬令时，夏令时的时间变化

原时区，原时间，转换比率，北京时区，北京时间，记录到DW中

* 1. 员工业务分组关系（par\_tcms\_work\_groupmember）

源表：wkg\_groupmember

数据源：谷仓tcms

CDH数仓表：par\_tcms\_work\_groupmember

备注：

* 模型参考谷仓tcms系统的wkg\_groupmember表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_business\_group
* 源码：谷仓tcms-dim源码
  1. 价格套餐表（tcms：par\_tcms\_freight\_selling\_price）

源表：prc\_freight\_selling\_price

数据源：谷仓tcms

CDH数仓表：par\_tcms\_freight\_selling\_price

备注：

* 模型参考谷仓tcms系统的prc\_freight\_selling\_price表结构

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_freight\_selling\_price
* 源码：谷仓tcms-dim源码
  1. 客户价格表（tcms:par\_tcms\_customer\_price wms:par\_wms\_customer\_price）

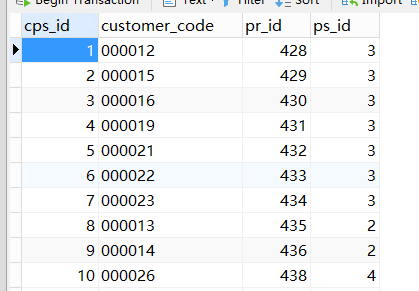
1. wms系统

源表：sp\_price\_rule(价格规则), sp\_customer\_price\_set(客户价格), sp\_price\_set(价格配置)

数据源：谷仓wms

CDH数仓表：par\_wms\_customer\_price

备注：

* 模型参考谷仓wms系统的sp\_customer\_price\_set, sp\_price\_set,sp\_price\_rule表结构
* 字段前缀：cp\_
* Sp\_customer\_price\_set表其实是sp\_price\_set和sp\_price\_rule之间互相关联的映射表，结构如下：
* 

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_customer\_price\_set
* 源码：谷仓wms-fact源码

1. Tcms系统

源表：prc\_customer\_price\_set,prc\_price\_set,prc\_price\_setgrade,prc\_product\_grade

数据源：谷仓tcms

CDH数仓表：par\_tcms\_customer\_price

备注：

* 模型参考谷仓tcms系统的prc\_customer\_price\_set,prc\_price\_set表结构, 其prc\_price\_setgrade,prc\_product\_grade等级表中的code，name
* 字段前缀：cp\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：dim\_customer\_price
* 源码：谷仓tcms-dim源码
  1. 退费费用规则表（tcms：par\_tcms\_returnfeerule）

源表：bil\_returnfeerule

数据源：谷仓tcms

CDH数仓表：par\_tcms\_returnfee\_rule

备注：

* 模型参考谷仓tcms系统的bil\_returnfeerule表结构
* 字段前缀：rfr\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_waybill\_returnfeerule
* 源码：缺失
  1. **仓租计费规则表（wms：par\_wms\_warehouse\_chargerule）**

源表：wh\_warehouse\_value

数据源：谷仓wms

CDH数仓表：par\_wms\_warehouse\_chargerule

备注：

* 模型参考谷仓wms系统的wh\_warehouse\_value表结构
* 字段前缀：wcr\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_wh\_warehouse\_chargerule
* 源码：谷仓wms-fact源码
  1. **仓租价格表（wms：par\_wms\_warehouse\_price）**

源表：wh\_warehouse\_price

数据源：谷仓wms

CDH数仓表：par\_wms\_warehouse\_price

备注：

* 模型参考谷仓wms系统的wh\_warehouse\_value表结构
* 字段前缀：wp\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_wh\_warehouse\_price
* 源码：谷仓wms-fact源码
  1. **费用折扣配置表（wms：par\_wms\_discount\_set）**

源表：sp\_discount\_sets(wms),sp\_discount\_set\_items(wms),

数据源：谷仓wms

CDH数仓表：par\_wms\_discount\_set

备注：

* 模型参考谷仓wms系统的sp\_discount\_sets和sp\_discount\_set\_items表结构
* 字段前缀：ds\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_discount\_sets, fact\_discount\_set\_items
* 源码：谷仓wms-fact源码
  1. **费用折扣表（wms：par\_wms\_discount\_fee**, **tcms：par\_tcms\_****rebate\_rate）**

源表：discount\_fee(wms), prc\_rebate\_rate (tcms)

数据源：谷仓wms, 谷仓tcms

CDH数仓表：par\_wms\_discount\_fee，par\_tcms\_rebate\_rate

备注：

* par\_wms\_discount\_fee模型参考谷仓wms系统的discount\_fee表结构，字段前缀：df\_
* par\_tcms\_rebate\_rate模型参考谷仓tcms系统的prc\_rebate\_rate表结构，字段前缀：df\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_discount\_fee(wms), ods\_customer\_rebate(tcms)
* 源码：谷仓wms-fact源码, 谷仓tcms-fact源码
  1. **客户返利规则表（wms：par\_wms\_customer\_discount\_fee）**

源表：customer\_discount\_fee(wms)

数据源：谷仓wms

CDH数仓表：par\_wms\_customer\_discount\_fee

备注：

* 模型参考谷仓wms系统的customer\_discount\_fee表结构，
* 字段前缀：cdf\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_cutomer\_discount\_fee(wms)
* 源码：谷仓wms-dim源码
  1. **客户返利关系表（wms：par\_customer\_discount\_relation）**

源表：customer\_discount\_relation(wms)

数据源：谷仓wms

CDH数仓表：par\_wms\_customer\_discount\_relation

备注：

* 模型参考谷仓wms系统的customer\_discount\_relation表结构，
* 字段前缀：cdr\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：ods\_cutomer\_discount\_relation(wms)
* 源码：谷仓wms-dim源码
  1. **杂费规则表（wms：par\_wms\_incidental\_value）**

源表：sp\_incidental\_value(wms)

数据源：谷仓wms

CDH数仓表：par\_wms\_incidental\_value

备注：

* 模型参考谷仓wms系统的sp\_incidental\_value表结构，
* 字段前缀：iv\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_incidental\_value(wms)
* 源码：谷仓wms-fact源码
  1. **杂费价格表（wms：par\_wms\_incidental\_selling\_price）**

源表：sp\_incidental\_selling\_price(wms)

数据源：谷仓wms

CDH数仓表：par\_wms\_incidental\_selling\_price

备注：

* 模型参考谷仓wms系统的sp\_incidental\_selling\_price表结构，
* 字段前缀：isp\_

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：fact\_operating\_cost(wms)
* 源码：谷仓wms-fact源码
  1. **价格规则表（wms：par\_wms\_price\_rule）**

源表：sp\_price\_rule(wms)

数据源：谷仓wms

CDH数仓表：par\_wms\_price\_rule

备注：

* 模型参考谷仓wms系统的sp\_price\_rule表结构，
* 字段前缀：pr\_

计算逻辑:

原数仓对应的目标表和源码：

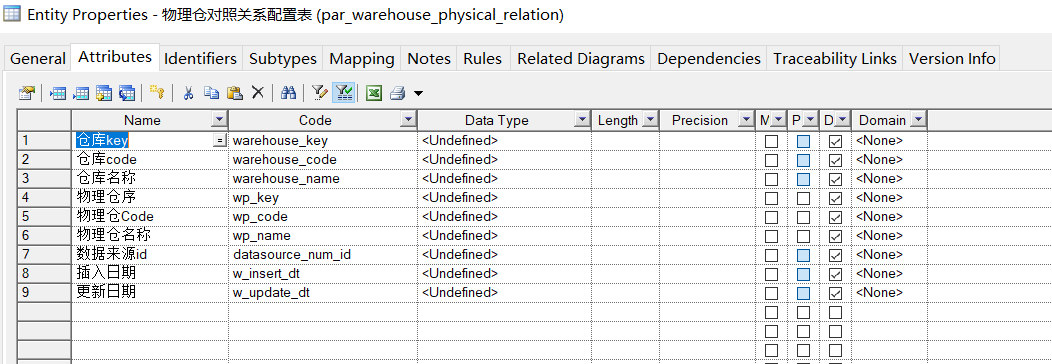
* 目标表：
* 源码：
  1. **物理仓对照关系配置表（****par\_warehouse\_physical\_relation）**

源表：tmp\_zt136\_zywmsSonwarehouse(984的gucang\_dw库)

数据源：984的gucang\_dw库

CDH数仓表：par\_warehouse\_physical\_relation

模型：



备注：

计算逻辑:

* 1，由于中邮wms在业务系统中没有物理仓概念，故需根据tmp\_zt136\_zywmsSonwarehouse表中的对照关系，在CDH中的dim\_warehouse\_physical表录入wp\_key，wp\_code等数据，这些数据参考wms系统表中的相应仓库的key值。如tmp\_zt136\_zywmsSonwarehouse表中的‘美国仓库’，则将CDH中的dim\_warehouse\_physical中对照的是wms系统（datasource\_num\_id=9004）的wp\_code=’美国仓库’的记录重新录入到dim\_warehouse\_physical表中【备注：重新录入期间，需将datasource\_num\_id改为9022，对应的wp\_key也需同步进行修改】；
* 2，在1步的完成基础上，就可以将tmp\_zt136\_zywmsSonwarehouse表中的记录录入到CDH的par\_warehouse\_physical\_relation表；

原数仓对应的目标表和源码：

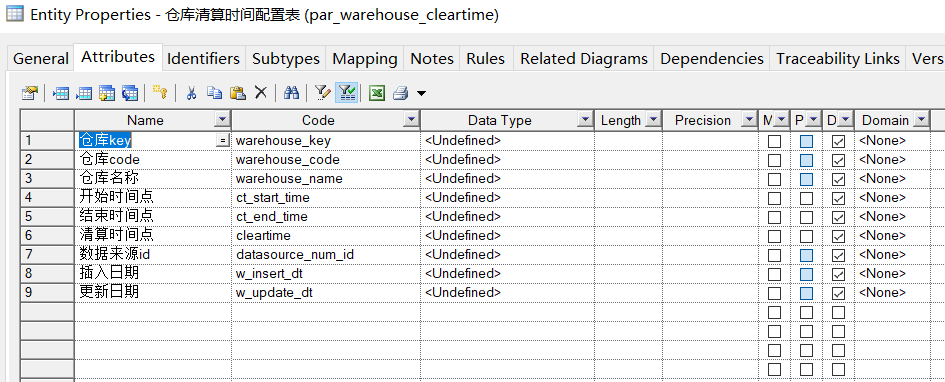
* 目标表：tmp\_zt136\_zywmsSonwarehouse
* 源码：
  1. **仓库清算时间配置表（par\_warehouse\_cleartime）**

源表：tmp\_warehousecleartime(984的gucang\_dw库)

数据源：984的gucang\_dw库

CDH数仓表：par\_warehouse\_cleartime

模型：



备注：

* 将984的gucang\_dw库的tmp\_warehousecleartime表根据CDH的par\_warehouse\_cleartime表字段进行录入

计算逻辑:

原数仓对应的目标表和源码：

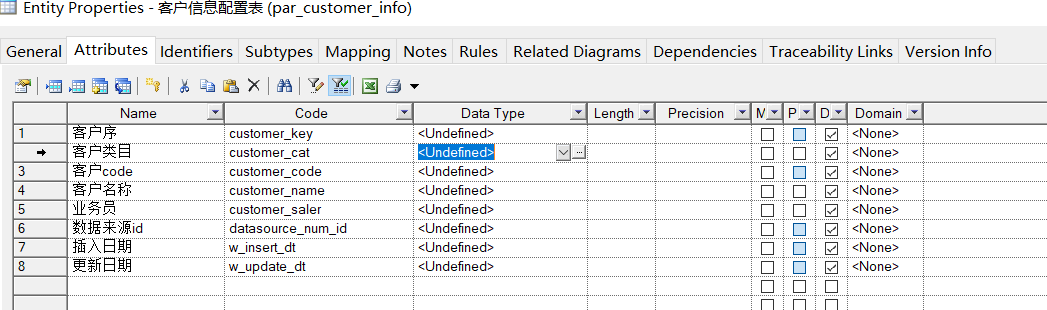
* 目标表：tmp\_warehousecleartime
* 源码：
  1. **客户信息配置表（par\_customer\_info）**

源表：tmp\_zt136\_tcmsCustomerHuizong(984的gucang\_dw库)

数据源：984的gucang\_dw库

CDH数仓表：par\_customer\_info

模型：



备注：

* 将984的gucang\_dw库的tmp\_zt136\_tcmsCustomerHuizong表根据CDH的par\_customer\_info表字段进行录入

计算逻辑:

原数仓对应的目标表和源码：

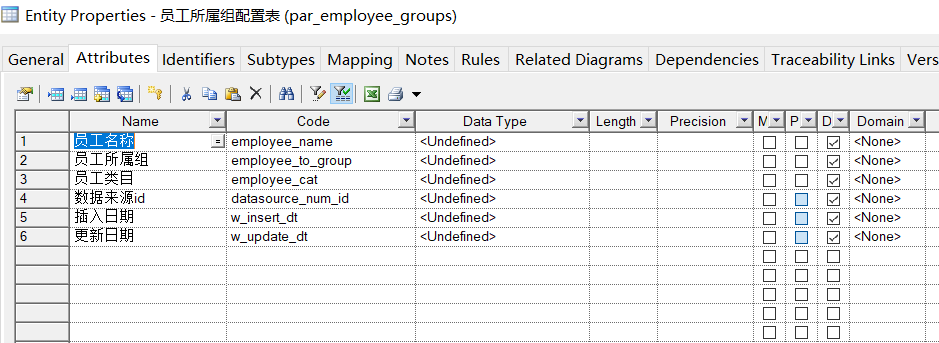
* 目标表：tmp\_zt136\_tcmsCustomerHuizong
* 源码：
  1. **员工所属组配置表（par\_employee\_groups）**

源表：tmp\_zt136\_wmsEmployeeGroup(984的gucang\_dw库)

数据源：984的gucang\_dw库

CDH数仓表：par\_employee\_groups

模型：



备注：

* 将984的gucang\_dw库的tmp\_zt136\_wmsEmployeeGroup表根据CDH的par\_employee\_groups表字段进行录入

计算逻辑:

原数仓对应的目标表和源码：

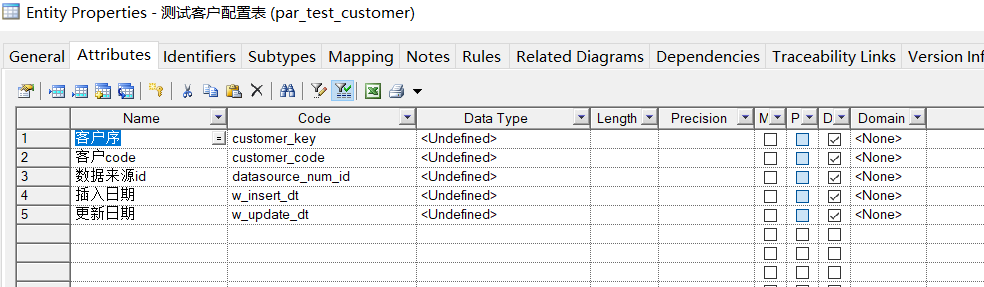
* 目标表：tmp\_zt136\_wmsEmployeeGroup
* 源码：
  1. **测试客户配置表（par\_test\_customer）**

源表：tmp\_zt136\_wmsTestCustomerList(984的gucang\_dw库)

数据源：984的gucang\_dw库

CDH数仓表：par\_test\_customer

模型：



备注：

* 将984的gucang\_dw库的tmp\_zt136\_wmsTestCustomerList表根据CDH的par\_test\_customer表字段进行录入

计算逻辑:

原数仓对应的目标表和源码：

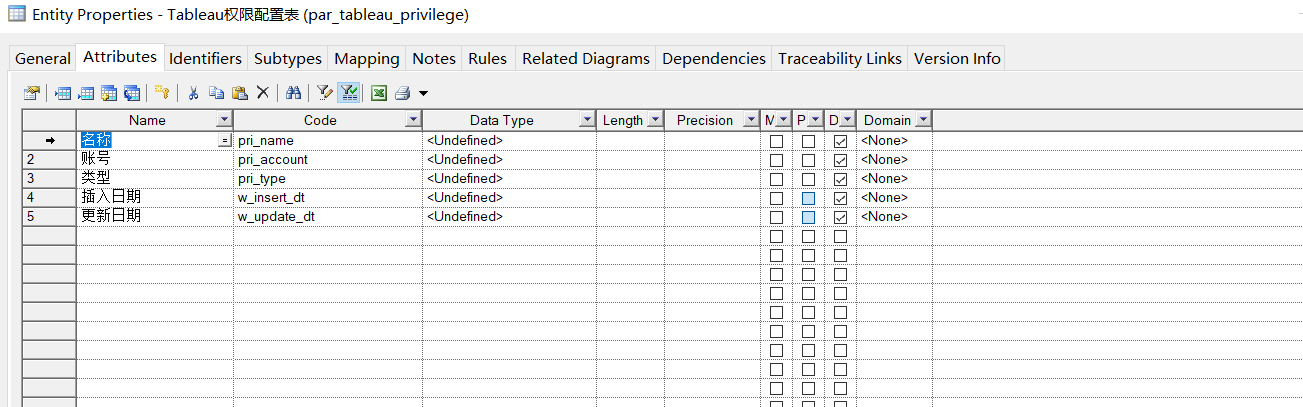
* 目标表：tmp\_zt136\_wmsTestCustomerList
* 源码：
  1. **Tableau权限配置表（par\_tableau\_privilege）**

源表：tmp\_tableau\_privilege(984的gucang\_dw库)

数据源：984的gucang\_dw库

CDH数仓表：par\_tableau\_privilege

模型：



备注：

* 将984的gucang\_dw库的tmp\_tableau\_privilege表根据CDH的par\_tableau\_privilege表字段进行录入

计算逻辑:

原数仓对应的目标表和源码：

* 目标表：tmp\_tableau\_privilege
* 源码：

1. 开发规范
   1. 存储规范
      1. 表命名规范

|  |  |
| --- | --- |
| **表类型** | **描述** |
| **聚合表(dm\_)** | 包含的是聚合后的数据 |
| **维度表(dim\_)** | 星形结构分析维度 |
| **事实表(fact\_)** | 保存可按各个维值来分析的事实数据的表。 |
| **参数表(par\_)** | 主要用于保存ETL处理过程中需要用到的数据映射关系的数据和控制ETL执行的信息，以及一些系统参数表 |
| **日志表(log\_)** | 用于记录日志的表 |
| **历史表(bak\_)** | 用于对历史数据进行备份 |
| **临时表(tmp\_)** | 用来ETL处理的数据临时存储的表 |
| **ODS表** | 用数据库缩写+原表名 |

* + 1. 字段命名规范

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **描述** | **规则** | **举例** |
| **row\_wid** | 数据仓库中唯一标识的主键 | 维度表建议序列号自增，事实表可以用组合主键 | 123 |
| **etl\_proc\_wid** | ETL编号，记录ETL跑数批次 | 一般会通过日期记录批次 | 20190821 |
| **w\_insert\_dt** | DW插入日期 | 记录数仓的插入时间 | 2019-08-01 |
| **w\_update\_dt** | DW最近更新日期 | 记录数仓的更新时间 | 2019-08-10 |
| **datasource\_num\_id** | 标识数据来源系统 | 数据来源进行编号配置 | 1,2,4,8 |
| **integration\_id** | 标识数据来源系统的主键 | 源系统的物理主键 | Mr-1898123 |
| **created\_on\_dt** | 标识数据来源对应源系统的创建时间 | 源系统的创建时间 | 2019-08-01 |
| **changed\_on\_dt** | 标识数据来源对应源系统的更新时间 | 源系统的更新时间 | 2019-08-10 |
| **ODS层字段命名** | ODS层字段名称 | ODS层字段名称参照源系统表结构 | 源：city；ODS：city |
| **DW层字段命名** | DW层字段名称 | 命名结构如下：  1，单个词表名：表名+LD层字段名称；  2，多个单词表名：表名首字母+LD层字段名称； | 1，单个单词表：  如，dim\_warehouse表，则字段名为warehouse\_code, warehouse\_city等；  2，多个单词表：  如，dim\_warehouse\_area表，则字段名为wa\_code, wa\_city等； |

* 1. 代码规范
     1. 脚本命名规范

脚本命名原则上统一使用小写字母，具体的内容为分层+目标表名，例如wms\_bil\_income。下列情况除外：

1. SHELL脚本：原则上统一用小写字母，若碰上建表语句，则相应的建表关键字用大写字母替换
2. 表脚本：数据表的关键字用大写字母，其他自定义字段采用小写字母，例如create替换为CREATE
3. 类脚本：按照常用的开发规则来进行命名，如类名采用驼峰法，方法名采用小驼峰法，关键字采用大写法等等
4. Azkaban：
5. 维度表规范(每个dw组别一个调度,包含ods层ddl建表语句和sqoop导表语句和dw层etl调度)  
   dim+"\_"+组别名称:dim\_Address  
   （2）事实表批量初始化(包含ods层ddl建表语句和sqoop导表语句)  
   fact+"\_"+init+"\_"+模块名称:fact\_init\_receiving  
   （3）事实表实时消费数据(执行实时spark代码,所有的实时job全部由一个project调度)  
   fact+"\_"+stream:fact\_stream  
   （4）事实表dw层调度(执行spark代码)  
   fact+"\_"+dw+"\_"+模块名称:fact\_dw\_receiving  
   （5）dm层调度(执行spark代码)  
   dm+"\_"+报表名称:dm\_onway

（6）dw层实时表azkaban的job命名规范:以idea中类的命名为准  
yarn中的任务名称:idea中类的相对路径  
（7）sqoop脚本azkaban的job命名规范:sqoop\_ods+数据库简称+"\_"+表名+"\_"+full  
yarn中的任务名称:sqoop\_import\_ods\_+数据库简称+"\_"+表名

* + 1. 脚本代码规范

脚本中需要有基本的注释信息方便以后维护，如参数、执行示例、脚本存放位置、日志文件地址、创建人、创建日期等信息。

　脚本注释示例：

　　　　　　###################################################################

　　　　　　#说明：将用户数据从HDFS导入MySQL

　　　　　　#参数CONNECTURL 说明：MySQL 连接地址 示例：jdbc:mysql://dajiangtai/djtdb\_test

　　　　　　#参数USERNAME 说明：MySQL 账号 示例：root

　　　　　　#参数PASSWORD 说明：MySQL 密码 示例：root

　　　　　　#执行示例： sh /home/hadoop/app/djt/sh/sqoop/sq\_ods\_d\_djt\_user\_copy.sh

　　　　　　#脚本存放地址：/home/hadoop/app/djt/sh/sqoop

　　　　　　#日志文件地址：/home/hadoop/app/djt/sh/log

　　　　　　#创建人：dajiangtai

　　　　　　#创建日期：20170401

　　　　　　#最新修改日期：20170401

　　　　　　#修改人、修改时间、修改内容、修改目的：xxxxxxxxxxxxxxxxx

　　　　　　###################################################################

* + 1. ETL调度命名规范

|  |  |
| --- | --- |
| **ETL过程** | **描述** |
| **ods\_xx\_inc** | 源层到ODS层的ETL过程,实时增量处理 |
| **ods\_xx\_full** | 源层到ODS层的ETL过程,批次全量处理 |
| **dw\_xx** | ODS层到DW层的ETL过程 |
| **dm\_xx** | DW层到DM层的ETL过程 |

1. 增量策略