**数据仓库规范**

目录

[1. 概述 4](#_Toc5067)

[2. 数据仓库命名规范 5](#_Toc6335)

[2.1. 数据分层分类规范 6](#_Toc15180)

[2.2. ods和dwd后缀规范 7](#_Toc7708)

[2.3. 作业编写规范 7](#_Toc14899)

[2.4. 数仓规范 8](#_Toc27784)

[2.4.1. 表属性规范 9](#_Toc1324)

[2.4.2. 索引 14](#_Toc8713)

[2.4.3. 视图 15](#_Toc30211)

[2.4.4. 物化视图 15](#_Toc7576)

[2.4.5. 存储过程 15](#_Toc4800)

[2.4.6. 触发器 15](#_Toc11329)

[2.4.7. 函数 15](#_Toc32448)

[2.4.8. 数据包 16](#_Toc19564)

[2.4.9. 序列 16](#_Toc28001)

[2.4.10. 普通变量 16](#_Toc8262)

[2.4.11. 游标变量 16](#_Toc27451)

[2.4.12. 记录型变量 16](#_Toc12452)

[2.4.13. 表类型变量 16](#_Toc14846)

[2.4.14. 数据库链接 16](#_Toc27612)

[2.5. 数据类型 17](#_Toc7280)

[2.5.1. 字符型 17](#_Toc26714)

[2.5.2. 数字型 17](#_Toc6940)

[2.5.3. 日期和时间 17](#_Toc28934)

[2.5.4. 大字段 17](#_Toc20398)

[2.5.5. 唯一键 18](#_Toc20915)

[3. 数据质量监控规范 18](#_Toc29722)

[3.1. 目的 18](#_Toc19504)

[3.2. 数据抽取监控 19](#_Toc7160)

[3.3. 数据质量监控 20](#_Toc8883)

[3.3.1. 监控规则符号 20](#_Toc29114)

[3.3.2. 检验规则 21](#_Toc19118)

[3.3.3. 检验逻辑 21](#_Toc24236)

[4. BI工具Superset规范 22](#_Toc13142)

[4.1. 命名规范 22](#_Toc31709)

[4.1.1. 数据库 22](#_Toc9573)

[4.1.2. 数据集 22](#_Toc23394)

[4.1.3. 图表 23](#_Toc11740)

[4.1.4. 看板 23](#_Toc29838)

[4.2. 权限规范 23](#_Toc21448)

[5. GIT代码上传规范 24](#_Toc4887)

[5.1. 地址 24](#_Toc6500)

[5.2. 上传规范 24](#_Toc26690)

[6. 生产库相关规范 25](#_Toc2823)

[6.1. 数据权限 25](#_Toc5771)

[6.2. 数据查询 25](#_Toc29372)

# 概述

数据模型是数据管理的分析工具和交流的有力手段；同时，还能够很好地保证数据的一致性，是实现商务智能（Business Intelligence）的重要基础。因此建立、管理一个企业级的数据模型，应该遵循标准的命名和设计规范。

# 数据仓库命名规范

* 命名应该使用英文单词，避免使用拼音，不建议不应该使用拼音简写。命名不允许使用中文或者特殊字符。
* 英文单词使用用对象本身意义相对或相近的单词。选择最简单或最通用的单词。不能使用毫不相干的单词来命名。
* 当一个单词不能表达对象含义时，用词组组合，如果组合太长时，采用用简或缩写，缩写要基本能表达原单词的意义。
* 当出现对象名重名时，是不同类型对象时，加类型前缀或后缀以示区别。
* 名称一律小写，以方便不同数据库移植，以及避免程序调用问题。
* 命名的各单词之间可以使用下划线进行分隔。
* 命名不允许使用[SQL](javascript:;)保留字。
* 命名长度，表名、字段名、视图名长度应限制在30个字符内(含前缀)

## 数据分层分类规范

数据结构体系（数仓可以分为以下几层）

* \*\*ods：贴源层\*\* （ods层主要目的是为了与数据源保持一致的数据结构，便于问题的排查，ods层不对外开放，不长期持久化数据；分增量和全量两种方式进行入库。）全量数据应定时化清除其分区的TTL；
* \*\*dwd：明细层\*\* （此层为事实明细数据，主要放置细粒度的维度清洗数据，不删减数据，合理化的归并数多个数据源；对数据做初步加工，比如空值处理、数据的标准化处理、增量数据合并为全量、历史拉链表之类。）
* \*\*dws(cdm)：\*\* 公共数据层（宽维度层）（一些计算复杂或者数据来源较多，且复用度较高的业务逻辑，以模型的方式沉淀在该层。\*\*\*一般体现为大宽表和轻度汇总数据\*\*\*。\*\*\*数据维度字段来源于dim或者维度字段字典，度量字段为数值类型，需要满足可累加的条件，维度可扩展\*\*\*）（此层还未规划，后续会增加）
* \*\*pre\*\* ：数据指标层（根据快仓业务拆分出来的数仓指标层，此层根据项目、指标、统计周期做分区，计算出相应的指标数据）
* \*\*ads:\*\* 应用数据指标层（pre指标汇总层）基于快仓业务需求，在仓库中产出的数据（pre层指标汇总之后，会推送数据到相应的数据平台展示接口进行指标呈现）
* \*\*dim：\*\* 维表（缓慢变化的维度表或固定不变的字典数据放置在此层）

## ods和dwd后缀规范

ods和dwd后缀规范：

\_df 每天全量

\_di 每天增量

\_da 特殊增量表（每天的分区就是当天的数据）

\_dz 历史拉链表

## 作业编写规范

* 必须的头部注释
* ```sql

-----------------------------------------------------------

-- 运行类型 ： 日跑

-- 参数 ： d 、 pt

-- 功能描述 ： 获取小车基础信息

-- 注意 ：每天全量，每天一个快照

-- 输入表 : ods\_xxxx1,ods\_xxxx2

-- 输出表 ：dwd\_xxxx

-- 修改历史 ： 修改人 修改时间 主要改动说明

-- 1 wangziming 2021-xx-xx CREATE (modify) -----------------------------------------------------------

```

* 必要的业务说明
* 关键计算逻辑必要的备注
* 尽量避免使用 right join
* 一个业务处理过程尽量封装在一个etl任务完成，任务内部的临时中间表作用域只限于任务内部，不允许其他任务使用
* 合理使用临时表，若任务中间用到临时表（为保证补取历史数据时任务之间不相互干扰，临时表应设置分区）
* 若sql计算逻辑较为负责，可适当的使用udf函数
* 任务上线后，根据数据和业务情况，适当的评估数据质量问题，确定是否需要配置数据质量检查（后续将逐渐完善此步骤）

## 数仓规范

进行规范统一化，不同单词之间尽量用下划线分开：

* 表命名

1. 数仓的表名应在整个数据仓库中是全局唯一的
2. 命名方式：<分层>\_<应用系统>\_{模块}\_<后缀>:

如ods（ods\_qkt\_basic\_bucket\_df），

dwd（dwd\_basic\_bucket\_info\_df）;

1. dws后缀规范
2. dim层表规范

一般不加后缀，且应不考虑分区，一般都是全量更新

### 表属性规范

#### 建表规范

1. 表名和字段名均必须有完整注释；
2. 一般都会建立分区；全量数据，保留每天最新快照数据；增量数据、保留增量分区数据;
3. 默认的文件格式是textfile，建议使用orc格式进行建表（text文件在计算时不能切分）；

#### 表名

##### ODS层表名

前缀为ODS\_应用系统名（缩写）\_数据表名。

数据表名称必须以有特征含义的单词或缩写组成，中间可以用“\_”分割，例如：ODS\_FUN\_CUSTOMERINFO。

表名称不能用双引号包含，表名长度不超过30个字符。

如果ODS设计采用贴源设计，数据表名应与源系统一致。

* 系统和应用名规则如下：
  + 核心 COR
  + 报表 RPT
  + ……
  + 如有新系统，按规则添加

##### DW事实表表名

前缀为DW\_主题名（缩写）\_功能描述。

数据表名称必须以有特征含义的单词或缩写组成，中间可以用“\_”分割，例如：DW\_ORD\_DETAIL。

表名称不能用双引号包含，表名长度不超过30个字符，[时间] + 周期类型(d/w/m/q/y)+汇总类别(d-detail/s-sumary)，例如：dws\_asale\_trd\_itm\_ord\_1d（最近一天汇总事实表），此层后续会主键建立完善。

* 主题名规则如下：
  + 订单 ORD
  + ……
  + 如有新主题，按规则添加
* 数据表名规则如下：
  + 基础表 \_BA
  + 日汇总表 \_D
  + 月汇总表 \_M
  + 历史累计 \_H
  + 全量加载 \_A
  + 增量加载 \_I

##### 应用层表名

前缀为APP\_主题名（缩写）\_功能描述。

数据表名称必须以有特征含义的单词或缩写组成，中间可以用“\_”分割，例如：APP\_RPT\_ DEALER\_GOODS。

表名称不能用双引号包含，表名长度不超过30个字符。

* 主题名规则如下：
  + 报表 RPT
* 数据表名规则如下：

参考DW层表名称规范

##### DW/DM维度表表名

前缀为D\_ 。

数据表名称必须以有特征含义的单词或缩写组成，中间可以用“\_”分割，例如：D\_ACCOUNT、D\_PUB\_DATE。

表名称不能用双引号包含，表名长度不超过30个字符。

* 数据表名规则如下：
  + 日期维度 D\_PUB\_DATE
  + 城市 D\_CITY

##### 元数据表名

前缀为M\_应用名（缩写）\_功能描述。

数据表名称必须以有特征含义的单词或缩写组成，中间可以用“\_”分割，例如：M\_ETL\_TASK。

表名称不能用双引号包含，表名长度不超过30个字符。

* 应用名规则如下：
  + ETL ETL
  + 报表 RPT
  + OLAP分析 OLP
  + 源系统 SRC
  + 数据库 DB
  + 软硬件 SHW
  + ……
  + 如有新应用，按规则添加

#### 表分区名

前缀为p。

分区名必须有特定含义的单词或字串。

例如 ：tbl\_pstn\_detail 的分区p2004100101表示该分区存储 2004100101时段的数据。

#### 字段名

字段名称必须用字母开头，采用有特征含义的单词或缩写，不能用双引号包含。

#### 字段排列

每个表中的字段排列也应该遵从相应的规则进行摆放：

* 同类属性尽量靠拢摆放

例如：“协议”实体中有一组“日期”属性，包括“开户日期”、“销户日期”、“签署日期”、“起息日期”、“到期日期”等，可以排列在一起。

* 相关属性尽量靠拢摆放
* 重要的和常用的属性靠前
* 和源系统非常接近的表（特别是一对一的情况），和源系统的属性顺序一致。

#### 主键名

前缀为PK\_。主键名称应是 前缀+表名+构成的字段名。如果复合主键的构成字段较多，则只包含第一个字段。表名可以去掉前缀。

#### 外键名

前缀为FK\_。外键名称应是 前缀+ 外键表名 + 主键表名 + 外键表构成的字段名。表名可以去掉前缀。

### 索引

#### 普通索引

前缀为IDX\_。索引名称应是 前缀+表名+构成的字段名。如果复合索引的构成字段较多，则只包含第一个字段，并添加序号。表名可以去掉前缀。

#### 主键索引

前缀为IDX\_PK\_。索引名称应是 前缀+表名+构成的主键字段名，在创建表时候用using index指定主键索引属性。

#### 唯一索引

前缀为IDX\_UK\_。索引名称应是 前缀+表名+构成的字段名。

#### 外键索引

前缀为IDX\_FK\_。索引名称应是 前缀+表名+构成的外键字段名。

#### 函数索引

前缀为IDX\_func\_。索引名称应是 前缀+表名+构成的特征表达字符。

#### 簇索引

前缀为IDX\_clu\_。索引名称应是 前缀+表名+构成的簇字段。

### 视图

前缀为V\_。按业务操作命名视图。

### 物化视图

前缀为MV\_。按业务操作命名实体化视图。

### 存储过程

前缀为SP\_ 。按业务操作命名存储过程。

### 触发器

前缀为tr\_ 。触发器名应是 前缀 + 表名 + 触发器名。

### 函数

前缀为Func\_ 。按业务操作命名函数。

### 数据包

前缀为Pkg\_ 。按业务操作集合命名数据包。

### 序列

前缀为Seq\_ 。按业务属性命名。

### 普通变量

前缀为Var\_ 。 存放字符、数字、日期型变量。

### 游标变量

前缀为Cur\_ 。存放游标记录集。

### 记录型变量

前缀为Rec\_ 。 存放记录型数据。

### 表类型变量

前缀为Tab\_ 。 存放表类型数据。

### 数据库链接

前缀为dbl\_ 。 表示分布式数据库外部链接关系。

## 数据类型

### 字符型

固定长度的字串类型采用char，长度不固定的字串类型采用varchar。避免在长度不固定的情况下采用char类型。

如果在数据迁移等出现以上情况，则必须使用trim()函数截去字串后的空格。

### 数字型

数字型字段尽量采用number类型，要注意精度。

### 日期和时间

#### 系统时间

由数据库产生的系统时间首选数据库的日期型，如DATE类型。

#### 外部时间

由数据导入或外部应用程序产生的日期时间类型采用varchar类型，数据格式采用：YYYYMMDDHH24MISS。

### 大字段

如无特别需要，避免使用大字段(blob，clob，long，text，image等)。

### 唯一键

对于数字型唯一键值，尽可能用自增产生。

该监控作用于数据入库完成之后。目前主要用于场景方历史数据质量校验，校验结果记录在数据库，然后生成报表，通过邮件系统发送。 数据质量监控包括规则、函数、自定义sql语句、自定义条件；规则与规则、规则与函数之间可以组合使用，中间用and连接。下面逐个进行讲解。

# 数据质量监控规范

## 目的

监控预警可以保证数据安全入库，入库过程出现异常则会实时预警。数据安全入库之后不一定能拿来用，可能会不符合规范、数据质量不过关，这个时候就需要去监控数据质量。

目前主要两种监控，任务调度监控【XXL-JOB】，数据质量监控【数据抽取监控】；监控都都以钉钉预警的方式进行预警及时通知。

## 数据抽取监控

该监控作用于数据入库阶段，目的是为了呈现数据的入库时间、入库数量以及是否成功入库。

监控报表字段示例如下：

* 数据来源
* 约定上传时间：抽取监控表示在某时刻把历史数据上传到XX生产服务器
* 确定数据源存在时间：确定与表对应的数据文件存在的时间
* 抽取任务开始时间：与表对应的数据文件入库开始时间
* 抽取任务结束时间：与表对应的数据文件入库结束时间
* 抽取耗费时间：与表对应的数据文件入库完成所消耗的时间
* 抽取类型：分为增量和全量两种

（全量的今日抽取记录必须大于昨日抽取记录）

* 昨日抽取记录：昨日与表对应的文件的总记录数
* 今日抽取记录：今日与表对应的文件的总记录数
* 今日新增记录：新插入历史区的数据（业务主键与历史区不同）
* 今日更新记录：新插入历史区的数据（业务主键与历史区相同，非业务主键与历史区不同）
* 今日覆盖记录：新增记录数+变更记录数
* 今日全表记录：该表所有记录数

## 数据质量监控

该监控作用于数据入库完成之后。目前主要用于历史数据质量校验，校验结果记录在数据库，数据质量监控包括监控规则。

### 监控规则符号

|  |  |
| --- | --- |
| 规则码 | 规则说明 |
| PK | 判断主键重复 |
| LEN | 判断字符的长度 |
| REG | 正则匹配 |
| NCHAR | 为空字符串 |
| NULL | 为空值，包括空字符串和null（不是字符串null） |
| !NULL | 不为空值，包括空字符串和null（不是字符串null） |
| CONTAIN | 包含某字符 |
| !CONTAIN | 不包含某字符 |
| > | 大于符号 |
| >= | 大于等于 |
| < | 小于 |
| <= | 小于等于 |
| = | 等于 |
| != | 不等于 |

### 检验规则

* 完整性：主要包括实体、属性、记录和字段值缺失四个方面。
* 准确性：一个数据值与设定为准确的值之间的一致程度，或与可接受程度之间的差异。
* 合理性：主要包括格式、类型、值域和业务规则的合理有效。
* 一致性：系统之间的数据差异和相互矛盾的一致性，业务指标统一定义、数据逻辑加工结果的一致性。
* 及时性：数据仓库ETL、应用展现的及时和快速性，任务运行耗时、运行质量、依赖运行及时性。

### 检验逻辑

#### 现场项目数据

例：科力普、中储等采集过来的数据，需监控每天都正常采集。目前这部分采集同步任务是在凌晨左右，定时监控任务可设置为3点执行一次。（由于现场工作日的不一致，非工作日的报警情况可视情况忽略）

* 定时任务需进行修改，此部分数据不需要实时执行

#### 邮件报表数据

目前四个项目有报表定时邮件任务，相关统计数据的准确性需要进行监控。

* 汇总数量等指标需保证为正整数
* 单次/平均数量等指标需保证为正数
* 空闲率等计算比率的指标需保证为小于等于1的非负正数

#### 指标分析数据

相关指标的计算同步在xxljob平台设置有调度任务，对于数据每天同步的及时性和完整性情况可直接参考调度任务的执行情况。

* 具体数据的准确性由于目前指标开发的不稳定暂不能达到一个统一的规范。

# BI工具Superset规范

## 命名规范

### 数据库

数据库名称需表明数据来源、使用目的或数据类型等，便于区分。

例：**数仓中心-mysql**，如为了区分不同数据库地址，可增加“.\*”作为后缀，即**数仓中心-mysql.5**。

### 数据集

由于数据集分为物理表（Physical）和虚拟表（Virtual）两种，以下分别进行描述：

1. 物理表，即来源于数据库的具体数据表，可直接使用库表名称，例：**[ones\_work\_order\_info](http://172.31.237.5:18088/superset/explore/table/98/)**。
2. 虚拟表：通过编辑sql语句生成的数据集，为了区分数据的使用用途，可在前面加上使用项目名称或者使用模块名称；同时由于逻辑的不确定性，进行命名时需加时间后缀，例：**[通威-作业任务状态明细 09/29/2021 14:47:18](http://172.31.237.5:18088/superset/explore/table/83/)**，方便后续查看和编辑。

### 图表

由于数据以及看板的多样性，会生成许多不同或者相似的图表来满足需求，图表的名称需明确表明数据使用模块及指标开发名称，例：**项目质量评估-[全部项目产品线占比](http://172.31.237.5:18088/superset/explore/?form_data={"slice_id": 182})**。

进行图表命名的规范也方便后续在看板中添加相关图表时更加快速准确的查找。

### 看板

看板将最终呈现开发的图表，看板的使用场景多为项目或者平台功能模块，命名可参考“【项目或平台名称】功能模块名称”，例：**[【项目质量评估】版本分析](http://172.31.237.5:18088/superset/dashboard/4/)，[【通威】搬运作业单分析](http://172.31.237.5:18088/superset/dashboard/15/)**。

## 权限规范

由于superset使用频率增加和使用范围逐步扩大，需实现权限管理，以防人员操作失误和数据不规范性对图表的呈现造成一定的影响。

# GIT代码上传规范

## 地址

GitLab网页地址 <http://172.31.234.12:10080/bigdata>

Git代码上传地址 ssh://git@172.31.234.12:10022/bigdata/

## 上传规范

* 相关代码上传到相关的项目或者数据分类下的dev分支内，切记**不要直接上传到master分支内**，代码要经过审核后才能上传至master正式环境上。
* 提交到Git上的代码要做**项目、版本、数据库环境、实现功能等明显标识**，后续复用或者查找相关代码时会减少不必要的繁琐操作。
* 如更新相关代码文件，需记录出更新、优化或者修改的原因，例：

**1.升级新版本导致脚本代码需调整，应创建新的文件**；

**2.修改优化脚本需记录是否为新需求，或者之前逻辑有误需要更改取值逻辑等。**

**注：**由于目前数据平台会统计Git上的代码上传量，所以尽量要做到每天上传有更改的最新代码，也有利于代码及时存储防止丢失。

# 生产库相关规范

## 数据权限

* 如有相关查询数据需求，可**提供账号**给到用户在.5的数据库上先进行测试验证，若为新增项目需先同步给梓明回流到.5。
* 未经许可生产库数据**不可暴露给外部人员**，包含数据库、superset等使用到生产库数据的环境。

**注：**由于数据的保密性以及为了避免其他人员对数据库的操作产生失误或者其他占用资源的行为，数据库的相关权限目前由梓明进行设置。

## 数据查询

* 如用户在mysql上进行数据查询验证时，遇到数据量大、关联表多或者其他导致执行速度慢的情况，及时提醒沟通、节约时间，可在数仓hive上进行查询并导出相关结果集。