Q/GJQH

国金期货有限责任公司企业标准

Q/GJQH 001-2023

期货大数据平台开发规范

Development Specification for Big Data Platform of Futures

2023-09-01 发布　　2023-10-1 实施

国 金 期 货 有 限 责 任 公 司 发 布

目 录

前 言

本规范是依据维度建模方法论，参考 OneData 建设体系，以及结合期货课题来积累的数据建设经验编写而成。

本规范中主要包含数据开发的任务建立规范、数据表的统一命名规范、任务的发布流程规范、数据采集规范。

本规范由国金期货有限责任公司与杭州玳数科技有限公司联合制定。

国 金 期 货 有 限 责 任 公 司 发 布

1. 建表规范

1.1. 表设计规范

1.1.1. 表设计主要目标

- 降低存储成本:合理的表设计可以降低数据分层设计上的冗余存储，减少中间表的数据量大小。对表数据的生命周期进行正确地管理，也能够直接降低存储的数据量及存储成本。
- 降低计算成本:规范化的表设计可以帮助您优化数据的读取，从而减少计算过程中的冗余读写和计算，提升计算性能，降低计算成本。
- 降低维护复杂度:规范化的表分层设计能够直接体现业务的特点。例如，在规范化设计表的同时对数据通道中的数据采集方式进行优化，可以减少分布式系统中小文件的问题，降低表和分区维护的数量等复杂度。

1.1.2. 表设计步骤

① 确定所属项目空间，依据业务过程规划表类型，分析数据层次。
② 定义表描述，进行权限定义与 Owner 定义。
③ 依据数据量、数据集成特点定义分区表或非分区表。
④ 定义字段或分区字段。
⑤ 进行表创建、表转换。
⑥ 明确导入数据场景的相关因素(包括批量数据写入、流式数据写入、条式数据插入)。
⑦ 定义表和分区数据生命周期。

1.2. 表命名规范

表命名公式

① ODS：ods__{数据源类型}__{数据库名}__{数据库原始表名}_{同步周期}_{增量全量标识}

例如:数据源类型为 oracle 的 settleadmin 库下的 t_invstpositiondtl 表按天全量采集

命名:ods__oracle__settleadmin__t_invstpositiondtl_df

② DWD：dwd_{业务 BU/pub}_{数据域}_{业务过程}[_自定义标签缩写]_{刷新周期}{增量全量标识}

例如:投资者交易数据域下持仓查询业务的事实表按天全量刷新

命名:dwd_osc_trd_qry_investor_position_df

③ DWS：dws_{业务 BU/pub}_{数据域}[_自定义标签缩写]_{刷新周期标识}

例如: 投资者交易数据域下关于风险状态的聚合表按天增量刷新

命名:dws_osc_trd_investor_risk_di

④ ADS：ads_ {业务应用缩写}[_维度][_自定义标签缩写]_{刷新周期标识}

例如:BI 面向用户交易维度分析场景的应用表按周全量刷新

命名:ads_bi_order_user_analysis_wf

⑤ DIM：dim_{维度定义}_dic

例如:时间维表

命名:dim_pub_time_dic

例如:用户维表

命名:dim_user_dic

注:

增量全量标识

① i:表示增量

② f:表示全量

③ di:天增量

④ df:天全量

刷新周期

① d:天

② w:周

③ cm:自然月

④ y:年

⑤ nd:n 天

⑥ nw:n 周

⑦ ncm:n 个自然月

⑧ ny:n 年

自定义标签缩写:自定义标签意在完善表名的表示的含义,可以由多个标签缩写组成

ods 表命名:ods__{数据源类型}__{数据库名}__{数据库原始表名}_{增量全量标识}其中前面的下划线为 2 个,为避免数据库原始表名中出现单个下划线引起的歧义问题

大括号“{}”内为必填项，中括号“[]”为选填项

1.3. 建表语句规范

① 字段名起名规范:请参考指标命名规范

② 字段注释要完善:前人栽树后人乘凉

③ 表名描述要完善:前人栽树后人乘凉

④ 分区键数仓统一规范，时间分区键 pt；其他分区键根据实际数据业务场景进行选择。

⑤ 指定表分区生命周期:为防止存储空间的浪费,及时清理分区

⑥ 指定表存储格式：数栈中默认建表格式为 orc

例如:投资者资金对账日增量事实表的建表语句

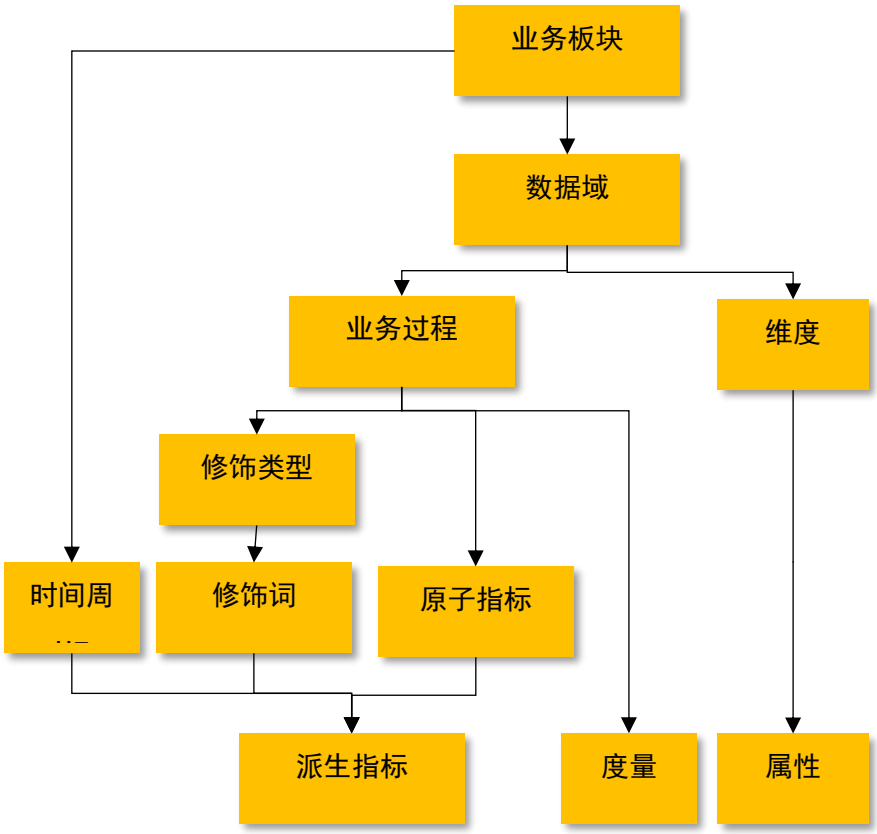
```
CREATE TABLE IF NOT EXISTS `Investor_fund_reconciliation_di`(
 TRADINGDAY                STRING           COMMENT  '交易日'
,INVESTORID                STRING           COMMENT  '投资者代码'
,T_S_INVESTOR              STRING           COMMENT  '投资者名称'
```

```
,ACCOUNTID                STRING          COMMENT  '资金账号'
,CURRENCYID               STRING          COMMENT  '币种'
,DEPARTMENTID             STRING          COMMENT  '组织架构代码'
,DEPARTMENTNAME           STRING          COMMENT  '组织架构名称'
,INVESTORTYPE             STRING          COMMENT  '投资者类型'
,OPENDEPOSIT              DOUBLE          COMMENT  '期初权益'
.....
 )COMMENT '投资者资金对账表'
PARTITIONED BY ( pt STRING COMMENT '时间天分区' )
stored as orc
lifecycle 999;
```

2. 指标规范

指标建设核心：一个指标一个定义一个计算逻辑，避免一个指标多种定义

2.1. 指标核心公式



1.数据域：是指一个或多个业务过程或者维度的集合

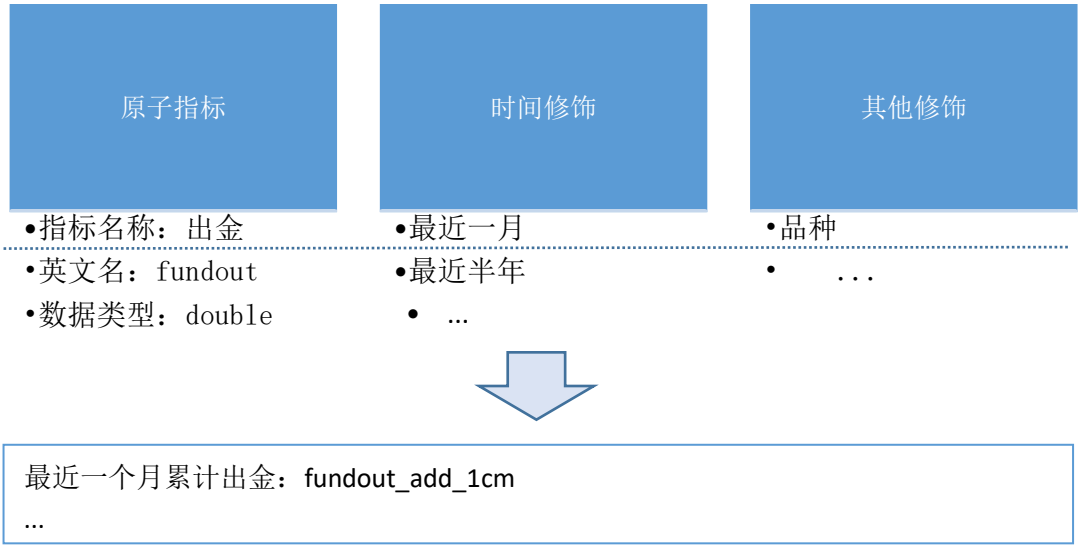
2.原子指标：基于某一业务过程下的度量。例如：出金:资金由交易账户划入银行账户的过

程。

3.派生指标=原子指标+时间修饰+其他修饰词。例如：日均可用资金

4.修饰：指针对原子指标的业务场景限定抽象。例如：最近 N 天

2.2. 指标核心公式实例



2.3. 指标命名约定

- 命名所用术语

  尽量使用英文简写，其次是英文，当指标英文名太长时，可考虑用汉语拼音首字母

- 原子指标

  英文名：动作+度量

  中文名：动作+度量

  原子指标必须挂靠在某个业务过程下。

  例如：出金，fundout

- 派生指标

  英文名：原子指标英文名+时间周期修饰词+其他修饰词（多个用下划线分隔）

  中文名：时间周期修饰词+[其他修饰词]+原子指标。

例如：最近一个月累计出金：fundout_add_1cm

2.4. 常用中文缩写命名约定

中文名	英文缩写
金额	amt
数量	cnt
比率	rate
时长	drt，如果单位是秒，则为 drt_s，小时为 drt_h
平均值	avg
排名前 N 名	topn

2.5. 常用的时间周期修饰词

中文名	英文名	中文名	英文名
最近 n 分钟	nmi	0 点开始的自然 n 分钟间隔	ncmi
最近 n 小时	nh	0 点开始点自然 n 小时间隔	nch
最近 1 天	1d	自然月	cm
最近 3 天	3d	自然季度	cq
最近 30 天	30d	周	w
最近 3 周	3w	截至当日	std
最近 3 月	3cm	0 点截至当前	tt
最近 3 年	3y	财年	fy
最近 1 小时	1h	未来 3 天	f3d

3. 数据开发任务建设规范

3.1. 目录命名规范

数据开发任务按数仓分层和场景分类的原则放到对应的文件夹下。
例如：SQL 任务对应的文件夹有 ODS,CDM,ADS，子目录根据具体场景分类。

3.2. 任务命名规范

- **SQL 类型任务**

  SQL 类型的任务大部分都是某张表的数据任务，任务命名直接以表名命名，包括 ODS,CDM,ADS 所有 SQL 表和对应任务，见名知意，有利于协作，方便跨项目依赖引用。

  例如:dwd_trd_pay_order_df 这个表对应的数据写入任务命名与表名相同命名为 dwd_trd_pay_order_df

  需要注意的是当任务调度周期改变的时候，表名和任务名都要跟随修改.

- **非 SQL 类型任务**

  命名建议采用：{任务类型}_{自定义名称}_{任务调度周期}

  任务类型：spark、mr、py、shell 等

- **hive 中数据同步到外部结构集的数据同步任务**

  命名采用：hive2{mysql/oracle}_{库名}_{同步的表名}

  例如:hive2mysql_bi_ads_biorder_user_analysis_1d

3.3. SQL 任务建设规范

3.3.1. 编写原则

① 代码行清晰、整齐，具有一定的可观赏性。
② 代码编写要充分考虑执行速度最优原则。
③ 代码行整体层次分明、结构化强。
④ 代码中应有必要的注释以增强代码的可读性。
⑤ 规范要求非强制性地约束代码开发人员的代码编写行为，在实际应用中，只要不违反常规要求，允许存在可理解的偏差。
⑥ 本规范在对日常的代码开发工作起到指导作用的同时也将得到不断的完善和补充。
⑦ 代码段中应用到的所有 SQL 关键字、保留字都需使用全大写或小写，例如 select/SELECT、from/FROM、where/WHERE、and/AND、or/OR、union/UNION、insert/INSERT、delete/DELETE、group/GROUP、having/HAVING、count/COUNT 等。不能使用大小写混合的方式，例如 Select 或 seLECT 等方式。
⑧ 代码段中应用到的除关键字、保留字之外的代码，都要求使用小写。
⑨ 四个空格为一个缩进量，所有的缩进均为一个缩进量的整数倍。
⑩ 禁止使用 select *操作，所有操作必须明确指定列名。
11 通常要求对应的括号在同一列上。

3.3.2. 关于 SQL 注释的特殊说明

① 每条 SQL 语句均应添加注释说明。
② 每条 SQL 语句的注释单独成行并置于语句前面。
③ 字段注释紧跟在字段后面。
④ 应为不易理解的分支条件表达式添加注释。
⑤ 应说明重要计算的功能。
⑥ 过长的函数实现，应将其语句按实现的功能分段加以概括性说明。
⑦ 常量及变量注释时，必须注释被保存值的含义，按需注释合法的取值范围 。

例如:

```
INSERT OVERWRITE TABLE t_trend_data PARTITION(pt=${pt})
SELECT  a.person_no
        ,b.person_sick
        ,b.person_type
        ,${pt} AS stat_date
FROM
(
        --t_flow_apply 最近一次的填写信息
        SELECT person_no
        ,max(create_time) AS max_create_time
        FROM `external`.t_flow_apply
        WHERE pt <= ${pt} --数据过滤条件
        GROUP BY person_no
) a
JOIN
(
        --查询 person_sick 在[3,5]且 is_new=1 的 t_flow_apply 数据
        SELECT person_no
        ,person_sick
        ,person_type
        ,create_time
        FROM `external`.t_flow_apply
        WHERE person_sick BETWEEN 3
        AND 5
        AND is_new = 1
) b
ON a.person_no = b.person_no
AND a.max_create_time = b.create_time
;
```

3.3.3. 表引用规范

① DWS 层任务应优先引用 DWD 和 DIM 层的表

② ADS 层的表应避免过度引用 DWD 层的表
③ ADS 层的表应避免引用 ODS 层的表
④ 公共指标应加工到 DWS 层
⑤ CDM 层的任务深度不应过大
⑥ DWD 与 DWS 层任务避免重复建设

3.3.4. 数据同步任务规范

数据采集规范

① 数据采集任务分为全量数据采集和增量数据采集，数据采集添加数据源时应查看数据源的数据库设置的时区是否为 Asia/Shanghai，以 Mysql 为例，若执行“show variables like '%time_zone%';”显示不是为 Asia/Shanghai，则需要在 jdbcURL 里添加时区设置 serverTimezone=Asia/Shanghai。
② 数据采集的方式要根据采集的数据量，业务场景，是否满足增量标识条件等综合因素进行选择，
③ 全量数据采集不需要在同步时选择增量标识，不需要添加过滤条件
④ 增量数据采集若源数据表满足增量标识的条件则选择增量标识，同时根据业务场景添加相应的过滤条件
⑤ 若是按业务数据时间进行过滤的增量同步方式，需要注意每天最后一次数据同步的时间是否符合业务场景，不符合的则需要在第二天凌晨后再进行一次数据同步任务，以保证昨天的数据不丢失，同时针对存在数据漂移情况的同步任务要做好数据的 ETL 工作
⑥ 要避免重复的数据采集
⑦ 全量同步的表必须设置生命周期，根据业务要求和历史数据分析需求来决定生命周期
⑧ 增量表暂定生命周期为永久

数据输出规范

① 数据输出同步任务应在目标数据库建立和数栈源表相同表名和对应数据类型的字段的表结构。
② 目标数据库目标表应根据相应的业务应用场景进行选择同步周期
③ 目标数据库目标表应根据业务应用场景选择对应的数据更新策略
④ 联合主键更新模式：在目标表建立联合主键进行数据更新
⑤ 全量更新模式：将目标表清空后全量同步最新应用数据进行数据更新
⑥ 按应用数据时间更新模式：按应用数据时间清空目标表一部分数据进行此部分数据的更新
⑦ 其他更新模式：按照对应的应用数据场景，针对目标表采取自定义更新策略
⑧ 数据输出同步应依赖对应的数据计算任务

3.4. ADS 应用表建设规范

1. 复用原则

输出应用表的设计应根据数据分析页面的需求进行整合,在数据量允许,分析主题,分析维度,调度周期不冲突的情况下进行优化合并.

2. 引用原则

ADS 表应优先引用 DWS 表,在没有公共建设的 DWS 表时可以引用 DWD 表,不建议直接引用 ODS 表.

3. 主题原则

ADS 是针对某一主题的分析,应避免多主题分析数据合并到一张 ADS 表中.

4. 任务发布流程规范

4.1. 发布规范共识

① 基础限制说明：任务发布要按照数栈的基础限制进行操作

② 任务发布流程：开发测试-->保存-->交叉审核-->提交

③ 交叉审核：必须由运维权限的另一名管理员（不能同时也是提交人员）审核通过，才能发布。其中重要的表任务和重要项目（超过二周）都需要在全员群里通知。

④ 变更通知提交机制：重要项目新建表、生产环境重要表变更，提交生产操作前需再三确认不会对现有生产任务产生影响，并发出信息通知其他管理员成员，由另外一名管理员审核通过才能执行，重要核心表调整需要在全员群里通知。

4.2. 发布前检查规范

① 数据验证，业务逻辑是否正确
② SQL 类型任务的编码美化
③ 检查注释是否完整
④ 验证表名和任务名是否符合规范
⑤ 自定义参数配置是否正确
⑥ 调度周期的选择是否正确
⑦ 上下游任务依赖是否正确

4.3. 发布任务变更规范

① 已发布线上任务表结构变更，先修改任务逻辑，确认无误再进行表结构变更

② 已发布任务调度周期的修改需要通知任务上下游依赖的负责人,确认无误再进行修改

③ 已发布任务计算逻辑的修改,需要在测试环境测试通过后,再修改线上任务的逻辑,通过发布前检查规范后进行发布操作。