数据仓库建设规范

# 说明

离线及实时数仓建设时，需要严格遵守本规范中：

1. 逻辑建模规范
2. 命名规范
3. 资产等级及生命周期规范
4. 元数据登记规范
5. 监控及校验规范

# 基本概念

1. 业务板块：
2. 数据域：
3. 主题： （数据域、主题域的差别？）
4. 维度：上报所得，维度是度量的环境。
5. 关联纬度：维表中纬度，通过关联进某个数仓层级。
6. 度量：上报所得，度量通常为数值型数据。
7. 总线矩阵：由一致性纬度、一致性度量为列，业务过程为行的矩阵。（？）
8. 单指标：单个度量计算所得。

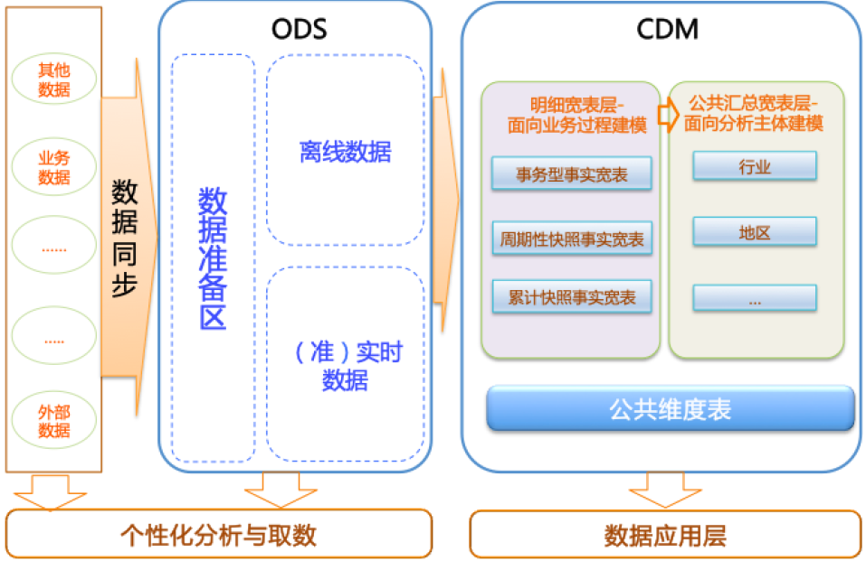
按计算类型划分：

1. 累加型：例如次数累积
2. 去重型：例如人数累积
3. 存量性：例如累计用户量。
4. 变化率型：例如同比环比。
5. 变化量型：例如次数差值。
6. 占比型。
7. 排名型：例如top100，一般依赖其他纬度进行排序。
8. 统计型：日均。
9. 衍生指标：多个指标再加工所得。例如：人均播放时长
10. 原子指标：包括单指标和衍生指标。
11. 派生指标：时间周期+修饰词+原子指标，派生指标可以理解为对原子指标业务统计范围的圈定。限制类似于SQL中**where**后的条件，包括时间条件。
12. 统计周期：例如：天、周、月。

# 整体命名规范

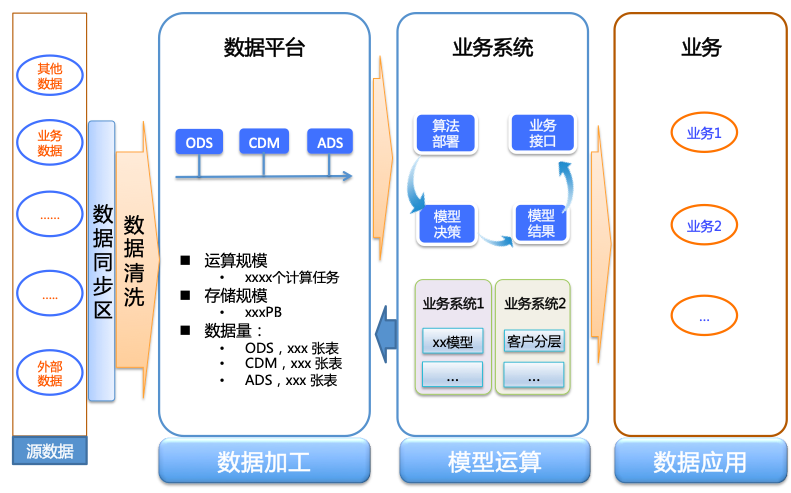
1. 全小写字母和数字，下划线分割。
2. 优先使用英文名，或英文简写名。
3. 英文名太长时，可以使用拼音首字母缩写。

# 数仓层级



1. 源数据层ODS：原始上报流水。
2. 明细层DWD：数据粒度同源数据层的的上报粒度，负责数据清洗，etl字段的逻辑维护。不做聚合操作，不做多天union操作，所有ETL逻辑均在DWD层。
3. 纬度层DIM：维表。特指外部接入，不包括流水加工后的特征表。
4. 汇总层DWS：指标产出层，需要保持口径统一，按需求做不同程度的聚合，以及维表DIM关联。可能由于运行时间问题，划分成多层的表。dws层包括特定数据系统的宽表。
5. 应用层ADS：用于报表展示。

# 数仓层级建设流程



1. 确认数据报表（例如业务产品），及数据使用方（例如推荐后台）对数据的需求。
2. 确定业务板块和数据域。
3. 确定业务过程的上报时机，梳理每个业务过程对应的纬度、度量，构建总线矩阵。
4. 确定明细层DWD设计。
5. 确定指标和衍生指标。
6. 梳理纬度对应的关联纬度。
7. 确定汇总层DWS设计。
8. 应用报表工具，或自行加工产出ADS层。

# 数仓层级调用规范

1. 原则上不允许不同的任务修改同一张表。
2. DWS层调用DWD层。
3. ADS调用DWS层。
4. 特例：如果ods过于特例化，而统计诉求单一，且长期考虑不会有新的扩展需求，可以跳过DWD或DWS。但是如果后期出现多个脚本需要访问同一个ods，则必须拓展出dwd及dws。

# 表存储类型（通用理论）

1. 全量存储：以天为单位的全量存储，以业务日期作为分区，每个分区存放截止到业务日期为止的全量业务数据。
2. 增量存储：以天为单位的增量存储，以业务日期作为分区，每个分区存放日增量的业务数据。
3. 拉链（极限）存储：拉链存储通过新增两个时间戳字段（start\_dt和end\_dt），将所有以天为粒度的变更数据都记录下来，通常分区字段也是这两个时间戳字段。

这样，下游应用可以通过限制时间戳字段来获取历史数据。例如，用户访问1月1日数据，只需限制 start\_dt<=20160101 and end\_dt>20160101。

该方法不利于数据使用者对数仓的理解，同时因为限定生效日期，产生大量分区，不利于长远的数仓维护。

极限存储底层的数据进行拉链存储，但上层通过视图操作或Hook，借助分析语句的语法树，将对极限存储之前的表查询转化成对极限存储表的查询。对于下游用户而言，极限存储表和全量存储方式是一样的。此外，为解决历史拉链分区大量产生的问题，极限存储中把日分区缩减为月份区，即以月为周期完成拉链存储。

极限存储虽然可以压缩大量的存储空间，但使用麻烦。主要难点在于全量表的维护和过滤变更频繁的维度属性。

1. 综上，通常情况下推荐使用全量存储处理缓慢变化维。在数据量巨大的情况下，建议使用拉链存储（极限存储）。

# DIM层

## DIM层设计

1. 仅包括非流水计算产生的维表。
2. 相同key的纬度需要保持一致性。
3. 如果由于历史原因，在暂时不一致的情况下，必须在规范的维度定义一个标准维度属性，不同的物理名也必须是来自标准维度属性的别名。
4. 在不同的实际物理表中，如果由于维度角色的差异，需要使用其他的名称，其名称也必须是规范的维度属性的别名。比如：视频所属帐号id，与视频分享帐号id。
5. 不同key的纬度，含义不要有交叉，避免产生同一口径，不同上报的问题。
6. 核心的维表产出时间通常有严格的要求。？

## 实时DIM层

1. 维度 DIM 层的存储选型参考：Redis/MySQL/HBase，按数据量和请求量按需选择，（构建一个统一维度服务）。
2. 为了方便不同场景下的使用，同一个实体的维度可以根据需要拆分成不同量级的维度表，存放到不同的存储中。例如，对于需要关联用户维度时，可以将全量的用户维表存储在 HBase 中，同时抽取日活用户的维度存储在 Redis 中，作为缓存，提高维度关联效率。

## 维度的组合与拆分（通用理论）

1. 组合原则
2. 将维度所描述业务相关性强的字段在一个维表实现，相关性一般指：经常需要一起查询、报表展现，比如商品基本属性和所属品牌；两个维度属性间是否存在天然的关系等。
3. 无相关性的维度可以适当考虑杂项维度，比如交易，可以构建一个交易杂项维度收集交易的特殊标记属性、业务分类等信息。也可以将杂项维度退化在事实表中处理，不过容易造成事实表相对庞大，加工处理较为复杂。
4. 所谓的行为维度是经过汇总计算的指标，在下游的应用使用时将其当维度处理。如果有需要，度量指标可以作为行为维度冗余到维度表中。
5. 垂直拆分
6. 是否需要垂直拆分。如果一个维表存在大量属性不被使用，或由于承载过多属性字段导致产出变慢，则需考虑对字段进行拆分，创建多个维表。
7. 根据维度属性的业务不相关性，将相关度不大的维度属性拆分为多个物理表存储。

比如会员表，建议拆分为核心表和扩展表。核心表相对字段较少，刷新产出时间较早，优先使用。扩展表字段较多，且可以冗余核心表部分字段，刷新产出时间较晚，适合数据分析人员使用。

1. 水平拆分
2. 是否需要水平拆分。如果记录之间有明显的界限，可以考虑拆成多个表或设计成多级分区。
3. 如果存在永远不会更新的记录，可能需要分别创建历史表与日常表。日常表用于存放当前有效的记录，保持表的数据量不会膨胀；历史表根据消亡时间插入对应分区，使用单个分区存放分区对应时间的消亡记录。
4. 冗余
5. 数据记录数较大的维度表，可以适当冗余一些子集合，以减少下游扫描数据量：

比如商品表，可以根据当天是否有行为，产出一个有活跃行为的相关维表，以减少应用的数据扫描量。

1. 可根据所属业务扫描数据范围大小的不同，进行适当子集合冗余。

## DIM层表命名规范

1. 下划线拼接各部分，全必填：

|  |  |
| --- | --- |
| 数仓分层 | 维表dim |
| 业务板块 | 视频app |
| 数据域 | 内容（维表）、用户（维表） |
| 关联key | vid（字典表的关联key为dict） |
| 自定义命名标签 | 用来区分相同的表 |
| 表分区 | 按天分区：d  按小时分区：h |
| 表分区的数据加工范围 | di - 增量表，加工范围同表分区，例如表分区为月，则该分区数据加工范围为自然月数据。  df - 统计历史到统计日的汇总数据  dl- 统计历史到统计日的汇总数据，拉链（极限）存储形式。 |

# DWD层

## DWD层设计

1. 确定涉及总线矩阵中的，哪些一致性纬度、一致性度量、业务过程。
2. 数据粒度同ODS层，不做任何汇总操作，原则上不做维表关联。
3. 底层公用的处理逻辑应该在数据调度依赖的底层进行封装与实现，不要让公用的处理逻辑暴露给应用层实现，不要让公共逻辑在多处同时存在。
4. 相同业务板块的dwd表，需要保持统一的公参列表。
5. 被ETL变更了的的纬度或度量，名称上要有区分。
6. 分解不可加度量为可加度量。
7. 减少filter条件不同产生的不同口径的表，尽量保留全表，用纬度区分口径。
8. 在上报原始数据允许的情况下，原则上只出小时表，除非有处理逻辑必须天级别处理，或者为了长时间保留，产出较少字段的天级别dwd表。
9. 实时S级DWD层必须落地存储用于校验，生命周期相对离线较短，可按需选择3到7天。

## 数据冗余（通用理论）

适当的数据冗余可换取查询和刷新性能，一个表做宽表冗余维度属性时，应该遵循以下建议准则：

1. 冗余字段与表中其它字段高频率（大于3个下游应用SQL）同时访问。
2. 冗余字段的引入不应造成其本身的刷新完成时间产生过多后延。
3. 公共层数据不允许字段重复率大于60%的相同粒度数据表冗余，可以选择在原表基础上拓宽或者在下游应用中通过JOIN方式实现。
4. 当需要从一个集合中冗余一部分记录作为另外一张表存在时，可以优先考虑子分区方式，只有以下情况才考虑冗余：
   1. 子类型表有较多（大于10）个字段，而父类型表并不存在。
   2. 子集合的过滤条件会被多次（大于5次）应用。

## 数据拆分（通用理论）

1. 数据的水平和垂直拆分是按照访问热度分布和数据表”非空或者0值”的数据值在行列二维空间上分布情况进行划分的。
2. 在物理上划分核心模型和扩展模型，将其字段进行垂直划分。核心模型包括的字段支持常用核心的业务，扩展模型包括的字段支持个性化或是少量应用的需要。
3. 将访问相关度较高的列在一个表存储，将访问相关度较低的字段分开存储。将高概率同时访问的数据放一起，将低概率同时访问的数据分开存储。
4. 将经常用到的where条件按记录行进行水平切分或者冗余；水平切分可以考虑二级分区手段，以避免多余的数据复制与冗余。
5. 将出现大量空值和零值的统计汇总表，依据其空值和零值分布状况可以做适当的水平和垂直切分，以减少存储和下游的扫描数据量。

## DWD层表命名规范

1. 下划线拼接各部分，全必填：

|  |  |
| --- | --- |
| 数仓分层 | 明细表dwd |
| 业务板块 | 参见基础概念 |
| 数据域 | 参见基础概念 |
| 主题 | 参见基础概念 |
| 自定义命名标签 | 用来区分相同的表 |
| 表分区 | 按天分区：d  按小时分区：h  实时更新：s |
| 表分区的数据加工范围 | di - 增量表，加工范围同表分区，例如表分区为d，则该分区数据加工范围为天数据。 |

# DWS层

## dws层设计

1. 基于某些维度的聚集是否经常用于数据分析设计，且纬度数量不要过多，不然没有聚合的意义，建议直接用dwd层。
2. 适当的维表的关联，方便下游使用。
3. 长周期的汇总计算，建议以天级别或小时级别来累计计算，避免周头或月头资源紧张。
4. 同数据域dws指标组保持一致，尽可能收敛产出同组指标的任务。
5. 不允许同一个dws表存储多个统计周期。
6. S级表均提供小时表及天表，处理逻辑尽量收拢到小时表。小时表主要满足时效性要求较高的分析需求，但是总存储量较大，核心表为了长时间的存储，产出天表，天表原则上不做过多的业务逻辑处理，仅基于小时表做group by。
7. 实时S级DWS需落地存储用于校验，生命周期相对离线较短，可按需选择3到7天。
8. 实时DWS的时间戳采用上传时间ftime，对齐离线。

## 周期快照型事实表（通用理论）

1. 周期快照型事实表主要用于分析状态型或者存量型事实，快照是指以预定的时间间隔来采样状态度量。

## 累计快照事实表（通用理论）

1. 累计快照事实表是基于多个业务过程联合分析从而构建的事实表，如采购单的流转环节等。
2. 累计快照事实表主要用于分析事件之间的时间间隔与周期，比如用交易的支付与发货之间的间隔，来分析发货速度，或在支付和退款环节分析支付退款率等等；同时也可以用于帮助分析一些少量的、且对刷新时间不是非常敏感的指标统计，比如，在当前事务型事实表不支持，且只有少量的统计指标时，需分析交易的关闭和发货，就可以基于累计快照事实表进行计算。

## 空值处理原则

1. 汇总类指标的空值：空值处理，填充为零。
2. 维度属性值为空：在汇总到对应维度上时，对于无法对应的统计事实，记录行会填充为null。

## dws层命名规范

1. 下划线拼接各部分，全必填：

|  |  |
| --- | --- |
| 数仓分层 | 汇总表dws |
| 业务板块/公共pub | 如果是一个业务板块就复用，否则新建 |
| 数据域 | 如果是一个原数据域就复用，否则新建 |
| 主题 | 如果是一个原主题就复用，否则新建 |
| 自定义命名标签 | 用来区分相同的表。  可与dws模型一一对应。 |
| 表分区 | 按分钟分区：min  按小时分区：h  按天分区：d  按自然周分区：w  按自然月分区：m  按自然季分区：q  按自然年分区：y |
| 表分区的数据加工范围 | tw - 滚动窗口（实时使用）  hw - 滑动窗口（实时使用）  sw - 会话窗口（实时使用）  di - 增量表，加工范围同表分区，例如表分区为月，则该分区数据加工范围为自然月数据。  df - 统计历史到统计日的汇总数据  th - 统计当天累计到统计小时的汇总数据  nd - 统计日近n天的汇总数据（滚动）  td - 统计当月累计到统计日的汇总数据  dl- 统计历史到统计日的汇总数据，拉链（极限）存储形式 |

1. 腾讯视频现有dws层表：[腾讯视频内容数仓文档](https://docs.qq.com/sheet/DZlduRnZaV0NGYWNP?tab=vzf2gv)

# ADS层

## ads层设计

1. ads层大部分由BI工具产生，不需要自己建表，表格式依赖工具的设计。
2. 字典转换由BI工具实现，下游无该功能时，才能在ads进行字典转换。
3. 工具不支持或由于日报场景时效需要自行开发ads时，格式要求如下。

## ads层命名规范

1. 下划线拼接各部分，全必填：

|  |  |
| --- | --- |
| 数仓分层 | 应用表ads |
| 业务板块/公共pub | 如果是一个业务板块就复用，否则新建 |
| 数据域 | 如果是一个原数据域就复用，否则新建 |
| 主题 | 如果是一个原主题就复用，否则新建 |
| 自定义命名标签 | 用来区分相同的表 |
| 表分区 | 按分钟分区：min  按小时分区：h  按天分区：d  按自然周分区：w  按自然月分区：m  按自然季分区：q  按自然年分区：y |
| 固定占位 | di - 增量表，加工范围同表分区，例如表分区为月，则该分区数据加工范围为自然月数据。 |

## ads层指标库表规范

1. 符合指标库规范的表，同比环比通过工具自动生成对应的波动表并计算波动值。
2. 指标表每一行只存当前维度组合下的值。
3. 库表字段固定如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段Code** | **字段名** | **字段类型** | **字段说明** |
| imp\_date/imp\_hour | 数据日期/数据时间 | bigint | 数据所属日期 |
| dim\_type | 维度组合 | string | 取值为本条记录对应的维度组合，按字段顺序，--分隔，如dim\_code\_a，或者dim\_code\_b，或者dim\_code\_a--dim\_code\_b  汇总项取值ALL |
| [dim\_code\_a] | 维度a | string | 维度code |
| [dim\_code\_b] | 维度b | string | 维度code |
| [ind\_code\_a] | 指标a | int/bigint/double | 指标code |
| [ind\_code\_b] | 指标b | int/bigint/double | 指标code |

# 宽表需求承接规范

宽表建设相当于用存储换计算，过度的宽表存储，可能会威胁底层表的存储资源，甚至影响集群稳定性，从而影响计算性能，造成本末倒置的问题。

# 库名及任务名规范

1. 每层有独立的数据库，或按业务各层合为一个库。

腾讯视频示例如下：

|  |  |
| --- | --- |
| pcg\_video\_dim | 内容dim |
| pcg\_video\_ads | 内容ads |
| pcg\_video\_dws | 内容dws |
| pcg\_video\_dwd | 内容dwd |
| pcg\_video\_ymp | 商业化ymp |
| pcg\_video\_real\_dw | 内容实时 |

1. 根据任务不同类型，确定的命名规范：

|  |  |  |
| --- | --- | --- |
| **类型** | **任务名** | **脚本名** |
| 一对一 | insert\_表名 | insert\_表名 |
| 模板一个脚本对应多个任务，一个任务对应一张结果表 | model\_结果表 | model\_结果表前缀 |
| 一个脚本对应一个任务，但有多个结果表 | multi\_结果表前缀 | multi\_结果表前缀 |
| 视图 | view\_表名 | view\_表名 |
| 出库 | export\_tdw\_表名(默认TDW出库到MYSQL)  export\_tdw\_pg \_表名(TDW出库到PG)  export\_tdw\_hbase\_表名(TDW出库到HDFS)  export\_tdw\_hdfs\_表名(TDW出库到HDFS) |  |
| 校验 | check\_表名 |  |
| 入库 | import\_表名 |  |

# 维度及指标命名规范

## 纬度及度量命名规范

1. 未进行etl的维度或度量沿用原始上报的key，部分场景会要求加前缀。
2. 需要添加前缀的特殊场景：
3. 被ETL变更了的的纬度或度量： el\_${非必选 自定义}\_key\_${非必选 单位}，前缀固定为el\_，标识进行过etl。度量可选加单位后缀。单位后缀：毫秒、秒、分钟、小时。

## 指标命名规范

1. 原子指标：一般多维度报表中，直接使用原子指标名。固定用in\_做前缀:

in\_${非必选 数据域词根}\_${非必选 自定义}\_${必选 基础指标词根}\_${非必选 日期词根}\_${非必选 单位词根}

1. 派生指标：一般用在日报结果表中，或推荐特征库中。派生指标命名包含：${必选 原子指标}\_${必选 修辞词}。 修辞词标记被限制的纬度取值，同样需要进行词根登记。
2. 指标日期词根命名规范：

|  |  |
| --- | --- |
| 最近3天 | 3d |
| 最近7天 | 1w |
| 最近14天 | 2w |
| 最近30天 | 1m |
| 最近60天 | 2m |
| 最近90天 | 3m |
| 最近180天 | 6m |
| 自然周 | w |
| 自然月 | y |
| 自然季度 | q |
| 自然年 | y |

## 字段类型

1. 数据类型不允许完全使用STRING类型，以免数据加工环节的数据质量问题无法及时暴露，必须为bigint或double的字段：
   1. 上报中的会被进行数值加工（四则运算、排序等）的纬度。
      1. 分区字段、日期字段
      2. 时间戳字段
      3. 数值类字段，例如时长。
   2. 加工出的指标。
2. dwd、dws层时长单位统一为秒（毫秒使用小数存储，保留3位）。
3. 剩余都是string类型，性能上可能有略微的差异，但是比较方便统一记忆。

# 数据资产等级

1. 资产等级的的划分，由数据本身的重要程度，以及数据链路被使用的频繁程度决定。随着业务的消亡，用户关注度的降低，资产等级会进行动态调整。
2. 数据等级定义如下：

|  |  |
| --- | --- |
| **资产等级分类** | **分类依据** |
| S | 影响新增结算、核心推荐效果 |
| A | 影响线上服务：推荐、搜索 |
| B | 影响主题数据 |
| C | 无影响 |

1. 在数据流转链路上，需要整理各个表是被哪些应用业务消费。通过给这些应用业务划分数据资产等级，结合数据的上下游依赖关系，将整个链路打上某一类资产等级的标签。
2. 根据ads的登记，来反推上报的数据等级，标注提醒开发。
3. 数据资产等级，还决定了任务调度优先级、告警级别、任务校验级别。

# 生命周期管理

1. 依据数据资产等级，结合数仓分层，以及存储用途，细分生命周期存储时间建议。

# 元数据登记

部分云组件包含元数据管理，可以在平台上登记，如果没有需要单独搭建元数据登记系统。

# 数据校验范畴（通用理论）

1. 校验分为以下四个方面：
2. 完整性

完整性是指数据的记录和信息是否完整、不缺失。数据的缺失包括数据记录的缺失（表行数异常）和记录中某字段信息的缺失（字段出现空值）。需要重点关注数据的生产环节和加工环节中表行数是否大于0、表行数波动是否正常以及字段是否出现空值或重复的情况。

1. 准确性

准确性是指数据记录中信息和数据是否准确、不存在错误或异常。例如，如果UV、PV数值小于0，则明显是错误数据。

1. 一致性

对于不同的业务流程和节点，同一份数据必须保持一致性。例如表字段的province字段中如果有浙江、ZJ两种表述，在group by province时会出现两条记录。

1. 及时性

及时性主要体现在最终ADS层的数据可以及时产出。为保证及时性，需要确保整条数据加工链路上的每个环节都可以及时产出数据。

1. 校验具体实施：
2. 监控分类：数据量、主键、离散值、汇总值、业务规则和逻辑规则。
3. 监控粒度：字段级别、表级别。
4. 监控层次：ODS和DWD层主要偏重数据的完整和一致性。DWS和ADS层数据量较小、逻辑复杂，偏重数据的准确性。不同数仓层级使用的监控规则也不一致。在数仓分层中，越靠近应用层数据越少、约束性越低，强弱选择为弱。

# 离线数据校验

1. 强校验情况下，可以阻碍后续任务不启动。
2. 问题排查记录是对数据校验的验收和改进，针对监控漏配情况进行补全。
3. 根据不同的校验目的，覆盖数据链路整个环节。
4. 上报的原始字段，包括基础校验、联动校验。

在测试时，或小版本放量时，使用大同。

针对参数空值、合法值、波动率。

1. ads分版本校验，包括各类指标横向对比

小版本放量后，连续几天观察，需要能到5%的量。通过后才能继续放量。

1. ads核心指标监控，不单独针对版本，包括各类指标横向对比。

针对全量数据异常，覆盖ads的核心纬度和核心指标。

1. 反推到dws

监控范围：

（1）ql表的自增监控。

（2）维表关联后的缺失监控。

1. 反推到dwd：

监控范围：

（1）公参清理后，使用启动表，再次监控空值、合法值、波动率。

（2）etl后纬度的空值、合法值、波动率（波动率目前误报较高，可选配置）。

1. 反推到ods：

监控范围：ATTA的行数波动。

# 实时校验

1. 为了校验和定位问题，实时S级DWD、DWS表需出库离线备份，根据这份备份数据与相应的离线数仓表及指标进行对比校验或定位问题。
2. S级实时指标和相对应的离线指标差异应满足：1）非去重类指标（如次数 PV、时长等）差异在 0.5% 以内；2）去重类指标（如人数 UV 等）差异在 1% 以内。