引言

首先分享我工作中遇到的一个小案例：

小A：我发现XXX这个历史累计指标既可以从服务端出，又可以从日志出，这2份数据源貌似有差异。

小B：是的，服务端记录的是当前还存在的，日志会把删除的也记下来。

小A：嗯，2份数据源会导致数据不一致的情况，我觉得我们应该从指标命名上区分开来。

小B：嗯，我这边是这么记得：服务端的直接从后端表获取，我直接放在维表中，且不加_std。

小A：啊？我不是这么写的呢，我是想在前面加个后缀呢。

小B：。。。

小A：。。。

案例虽小，但折射出来的问题是：每个人都有自己的一套准则，一个数仓几套准则的话，业务怎么用？新人怎么了解？指标一致性怎么保证？所以需要建设数仓标准化规范，保持队形。

另外一方面，所谓"前人留坑、后人填坑"，接盘侠一般会很痛苦，好的规范一定程度上可以降低接盘侠的痛苦。比如版本迭代规范，可以很好的追溯新指标上线情况、bug修复情况、数据回刷情况等。

数据规范是数仓体系建设的"语言"，是数据使用的说明书和翻译官，同时也是数据质量的保驾护航者。采用标准化、系统化规范让用户"看得懂"、"找得到"、"放心用"，保证指标的一致性、保证数据高质量生产。

一、模型层次调用规范

设计模型层次和制定调用规范的目的是：避免烟囱式建设，数据更多的共享，减小重复计算、保持数据一致性等。主要有以下几点：

- ods层一般只被dwd层调用，不建议跨层调用ods层
- ads层优先调用dwd和dws层数据，不建议直接抽ods层数据，优先调用dws层
- 应充分了解业务需求，尽量沉淀常用指标到dwd层、dws层、dim层
- dws层可先建轻度汇总层，避免指标直接从dwd明细层出
- dws层要尽量沉淀出常用指标，避免ads层过度调用dwd层数据
- dws层不宜过深，比如超过10层

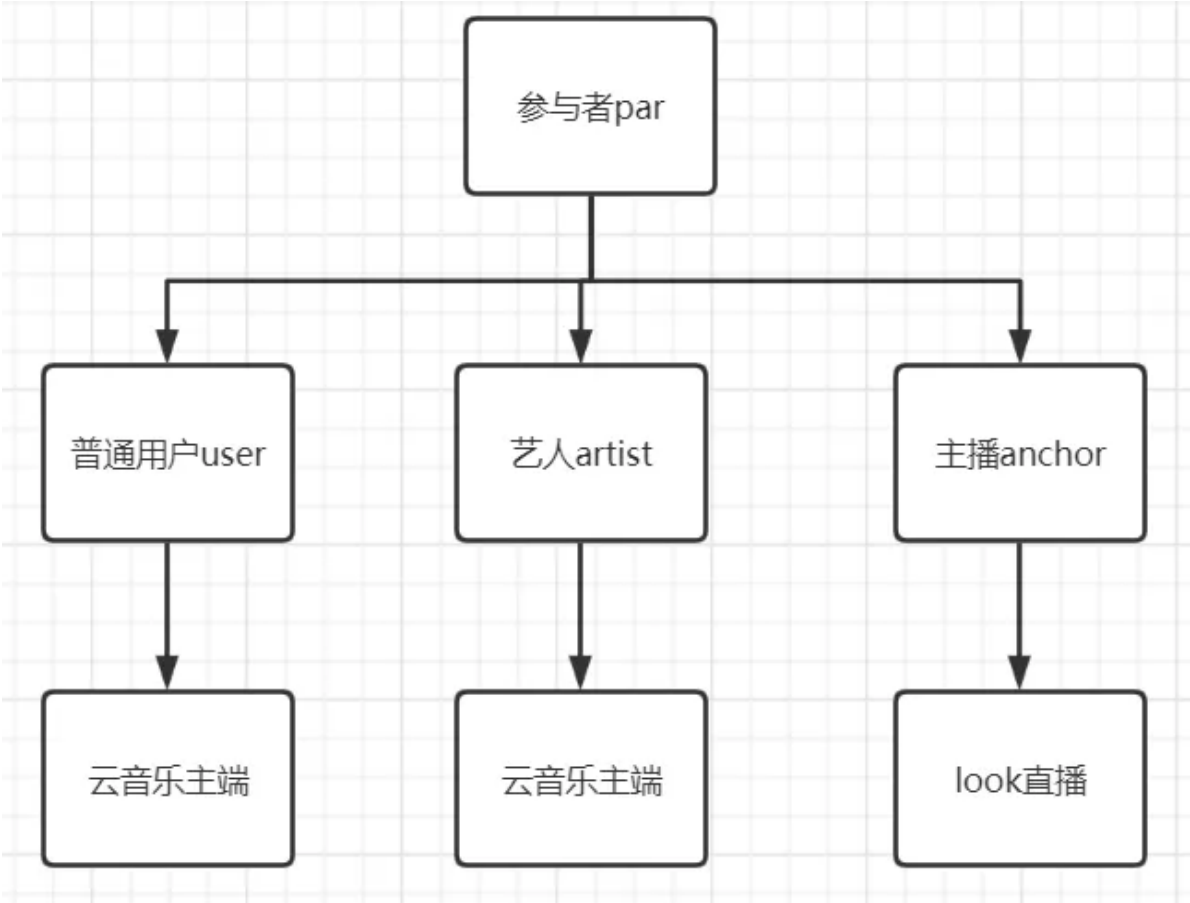
二、命名规范

字段命名一般有以下通用规范：

- 命名一律采用小写
- 由字母、下划线、数字组成，不能以下划线和数字开头
- 尽量用英文简写，其次是英文
- 部分可以汉语拼音首字母标准缩写，如中国制造zgzz
- 命名不宜过长

1、数据域命名规范

确定好数据域后，需要对其命名，比如上一篇涉及到的



此外一些常用的命名，如：

- 用户域：par
- 日志域：log
- 关系域：sns
- 广告域：adv
- 位置域：loc
- ……

2、表命名规范

可以对表进行如下的约定：

1）临时表：tmp*名字全称*对应输出正式表名*自由发挥*yyyyMMdd

2）正式表：可以采用**模型层次、数据域、表描述和分表规则**结合的方式

[层次][一级主题域][二级主题域][产品或业务板块][内容描述][分表策略]

说明：

1. ods层表：尽量保持与源系统相同的命名，不涉及主题域，在此基础上增加层次或分表策略
2. 内容描述
    - dim表内容描述一般是维度标识词
    - dwd事实表，内容描述跟业务过程有关
    - dws/ads可以增加统一关键词标识有聚合功能，比如aggr、tag等
3. 层次：就是数仓的模型层次ods/dwd/dim/dws/ads
4. 主题域：即指上面提到的数据域，根据业务情况确定是否有二级主题和产品或业务板块
5. 分表策略：即为数据的生命周期，主要有下面几种：

表策略	简称	说明
小时全量	dh	每小时分区中保留的是历史至今的全量数据
小时增量	hi	每小时分区中保留的是当前小时的增量数据
日全量	dd	每天分区中保留的是历史至今的全量数据
日增量	di	每天分区中保留的是当日的增量数据，可以是汇总数据也可以是明细数据
周全量	wd	每周的分区中保留的是历史至今的全量数据
周增量	wi	每周的分区中保留的是对应周的增量数据，可以是汇总数据也可以是明细数据
月增量	mi	每月的分区中保留的是对应月份的整个月的增量数据，可以是汇总数据也可以是明细数据
月全量	md	每月的分区中保留的是历史至今的全量数据
无时间分区	nd	死表或者不定期更新的全量表

3）表的分区命名规范

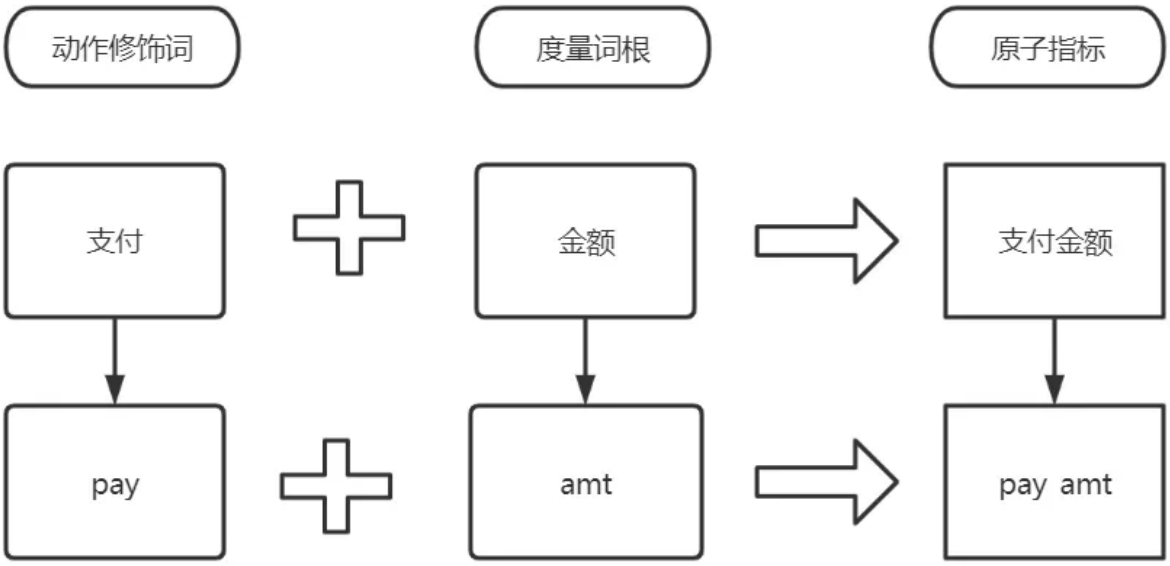
即表更新周期，确定表数据按小时、天、周、月、年更新：

分区名	分区字段	说明
小时分区	pt_d	HH
天分区	pt_d	yyyy-MM-dd
周分区	pt_w	yyyy-MM-dd，无特殊需求统一写所在周周一的日期
月分区	pt_m	yyyy-MM
年分区	pt_y	yyyy

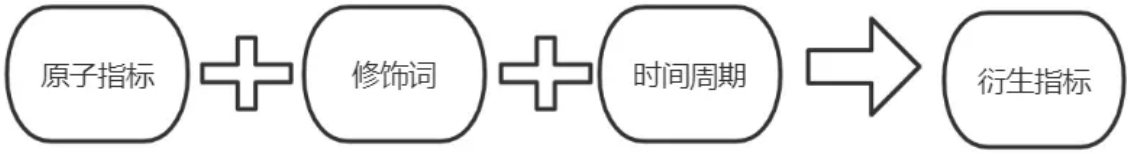
3、指标命名规范

1）原子指标：原子指标就是度量，对某一业务事件进行度量，**有明确的业务含义**，比如支付金额等。

原子指标隶属于业务过程，一般在事实表中包含，所以创建原子指标时必须选择所属的业务过程。原子命名规范可由业务修饰词 + 词根组成：



2）衍生指标：对原子指标业务统计范围的确定。由一个原子指标 + 修饰词 + 时间周期组成。



衍生指标唯一归属一个原子指标，继承原子指标的数据域。

衍生指标可以分为三类：事务型指标、存量型指标和复合型指标。按照其特性不同，有些必须新建原子指标，有些可以在其他类型原子指标的基础上增加修饰词形成衍生指标。

1. **事务型指标：**是指对业务活动进行衡量的指标，如近N天支付金额。这类指标需维护原子指标及修饰词，在此基础上创建衍生指标。
2. **存量型指标**：是指对实体对象某些状态的统计，对应的时间周期一般为“历史截止当前某个时间”。这类指标需维护原子指标及修饰词，在此基础上创建衍生指标。
3. **复合型指标：**在事务型指标和存量型指标的基础上复合而成，有以下几种：

- 比率型：比如xxxCTR、xxx满意度。这种情况下需要创建原子指标，比如创建CTR、满意度等原子指标。
- 比例型：比如xxx百分比，xxx占比。这种情况下需要创建原子指标，比如创建播放歌曲人数占比。
- 变化量型：比如xxx指标相对上N天的变化量。这种情况下不不创建原子指标，增加统计方法相关的修饰词，然后在此基础上创建衍生指标，比如上N天变化量的修饰词。
- 变化率型：比如xxx指标相对上N天的变化率。这种情况需要创建xxx变化率原子指标。
- 统计型：比如人均、次均，xxx分位数等。这种情况下不创建原子指标，增加统计方法相关的修饰词，在此基础上创建衍生指标。
- 排名型：一般为TOP_xxx_xxx。这种情况下创建原子指标，比如top_n_支付金额，在此基础上创建衍生指标。

3）修饰词的规范

为了规范指标的命名，我们可以对**词根、时间周期等**修饰词进行约定，比如：

①常用的【词根】可以做如下约定：

词根	命名规范	类型	说明
用户数量	unt	bigint	
次数	cnt	bigint	
数量	num	bigint	
金额	amt	Decimal	
首次	fst		类型根据具体度量确定
末次	lst		类型根据具体度量确定
平均	avg		类型根据具体度量确定
常用	com		类型根据具体度量确定
比率	ratio	Decimal	
时间戳	timestamp	bigint	
时间	time	string	yyyy-MM-dd HH:mm:ss
日期	date	string	yyyy-MM-dd
加密	encry	string	
描述	desc	string	
人民币	RMB	Decimal	
美元	USD	Decimal	
前N	topN		类型根据具体度量确定
升序	asc		类型根据具体度量确定
降序	desc		类型根据具体度量确定
时长	duration	Decimal	
活跃	active	string	
新/老	new/old	string	
版本	ver	string	

②常用的【时间周期】可以做如下约定：

时间周期	简称	说明
最近1天	_1d	d：day
最近3天	_3d	
最近N天	_Nd	
最近7天	_1w	w：week
最近14天	_2w	
最近30天	_1m	m：30天b：beforey：year
最近60天	_2m	
最近90天	_3m	
最近180天	_6m	
最近一年	_1y	
180天以前	_b6m	
未来7天	_p1w	p：predict；w：week
未来4周	_p1m	p：predict；m：month
自然周	_cw	c：calendarw：week；q：quarter；y：year；fy：fiance year；hfy：half fiance year
自然月	_cm	
自然季度	_cq	
自然年	_cy	
财年	_fy	
半财年	_hfy	
历史截至当日	_std	std：start to day
自然年初截至当日	_ytd	ytd：year to day
自然季度初截至当日	_qtd	qtd：quarter to day
自然月初截至当日	_mtd	mtd：month to day
自然周初截至当日	_wtd	week to day
财年年初截至当日	_ftd	fiance(year)to day
最近1小时	_1h	h：hour

时间周期	简称	说明
0点截至当前小时	_dth	day to hour
0点截至当前	_dtr	dtr：day to realtimehtr：hour to realtimemtr：minute to realtime
小时截至当前	_htr	
分钟截至当前	_mtr	
活动开始截至当日	_atd	Activity start to day
活动开始截至当前	_atr	Activity start to realtime

4）公共字段规范

根据部门业务特色，我们可以沉淀一些共有的字段命名，比如维表中的一些字段、专有词语等：

字段描述	缩写	字段类型
年龄	age	string
性别	gender	string
城市	city	string
省份	province	string
国家	country	string
用户id	user_id	string
出生日期	birth_date	string
地址	addr	string
邮箱	email	string
星座	constellation	string
昵称	nick_name	string
收入	income	Decimal
设备id	dev_id	string
操作系统	os	string
操作系统版本	os_ver	string
app版本	app_ver	string

三、开发规范

数据开发工程师在开发模型的时候，也尽量保持一致的规范，方便统一管理和运维。具体规范根据部门使用的数据平台有关，比如网易使用的是网易易数开发平台，那我们一般会有如下的约定：

1）建表规范

- 临时表采用内部表；正式表采用外部表
- 采用分库形式，按照模型层次建立对应的库名，保持库名、表层级的一致性
- 建表时增加表属性：开启impala快速查询功能、添加表创建人POPO账号
- 建表必须要有comment，说明表使用场景等
- 表字段必须要有详细的描述说明，枚举值必须要有数据字典
- 临时表后缀要添加创建日期，便于表生命周期管理
- 表的存储也有严格的规定
    - 临时表统一放在临时库对应的临时目录下，比如：公共路径/临时库/tmp/账号/表名
    - 正式表按模型层级存在统一的目录下，比如：公共路径/库名/层次/表名
- 表数据存储格式通常采用parquet的格式
- 表一般会采用snappy压缩
- 建表一般都有分区

2）作业流规范

根据网易易数开发平台的特点：数仓以作业流的形式开发。为了便于管理开发作业，我们做了如下规定：

- 一个作业流只输出一张结果表，通过配置跨流依赖来保持顺序持续任务，表之间的依赖关系通过血缘可以快速查看。
- 作业流的名称以产出表名命名
- 作业流按模型层次分组，保持模型、表、作业流一致性，比如dim表只放在dim组下。
- 非实时作业流一般采用T+1的方式调度，即今天更新作业的数据。

3）数据格式规范

我们对数据格式也做了一些约定，比如：

- 时长统一到秒，时间戳统一到毫秒
- 时间和日期格式统一采用yyyy-MM-dd HH:mm:ss和yyyy-MM-dd

这里根据具体部门的业务需求来统一约定。

4）数据字典规范

开发过程中会涉及到各种枚举值，为了方便管理，也可以统一码值，建立标准化码表。

5）维度规范

维度是维度建模**的基础和灵魂，**维度的作用是：查询约束、分类汇总、以及排序等。所以对维度统一规范非常重要，比如：

- 共享公共维度
- 维度采用一张主维表+多张次维表的形式。
- 维度主键唯一
- 维度属性的枚举值要统一
- 维度属性命名要统一

- 维度属性值格式要规范
- 维度数值型单位要统一

6）指标来源规范

- 为了保持数据一致性，同一业务含义指标的源头尽量保持同一张表，比如所有相关评论指标应该能追溯到同一个数据表。
- 同一业务指标需要涉及多个源数据时，比如日志和服务端的数据。建议尽量加以区分，比如命名上区分。

7）指标一致性建设规范

在开发指标过程中，做到需求描述、计算口径、指标口径保持一致性，同时做到指标可追溯可管理。可依托于工具来完成指标的管理。

8）迭代规范

模型上线后期会出现补数据、bug修复、增加指标等情况，为了做到版本迭代可追溯，需要在代码中增加版本迭代备注，比如：版本迭代时间、开发人员、迭代内容、版本上线时间、是否补数据、补数据的分区范围等。

四、交付标准规范

kimball指出，对数仓的基本要求是：对业务用户发布高质量、相关的、可访问的信息和分析 。

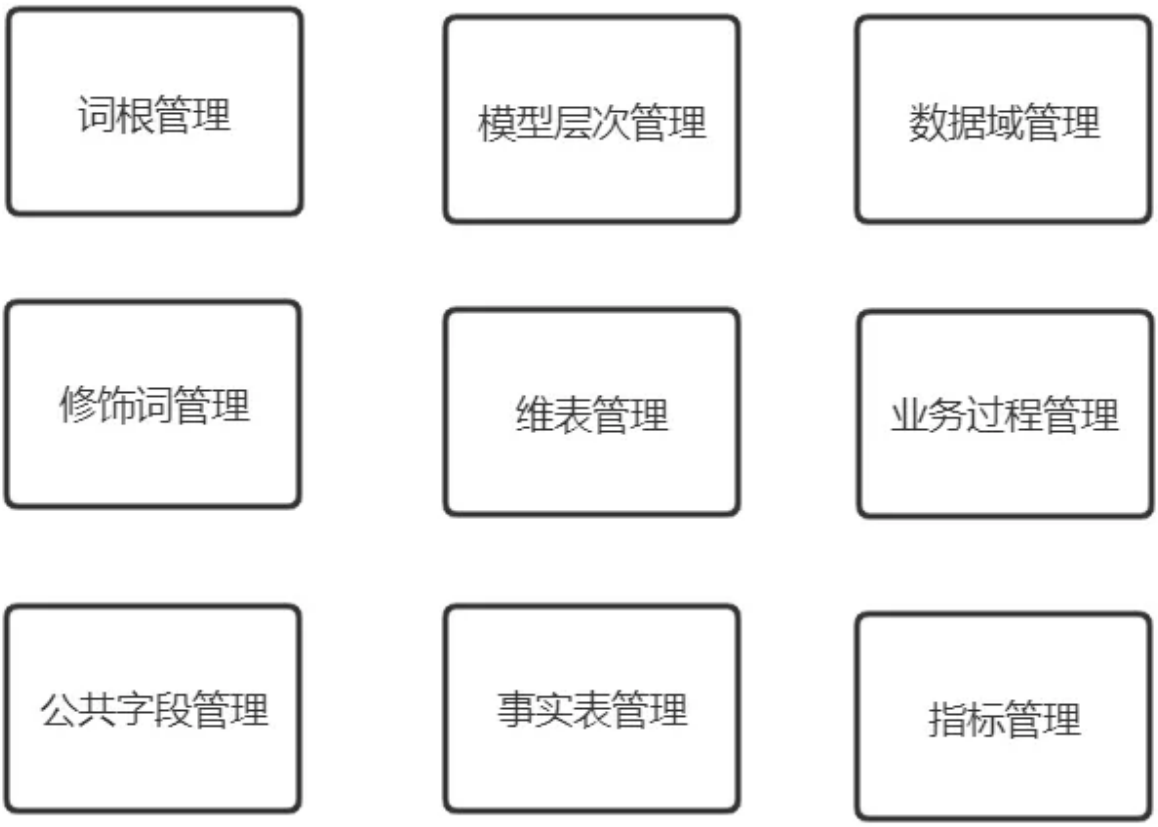
那么，数据开发完成后，并不是直接上线交付，还需要重要的一步是保证交付标准，保证数据质量和产出的及时性。实现的手段是数据质量测试、质量监控、调度监控。基本原则主要有以下：

1. **准确性**：准确性是指数据中记录的信息和数据是否准确，是否存在异常或错误的信息。
2. **完整性**：指数据的记录和信息是否完整，是否存在缺失的情况。数据的缺失主要包括记录的缺失和记录中某个字段信息的缺失，两者都造成统计结果不准确，所以说完整性是数据质量最基础的保障。
3. **一致性**：一致性一般体现在跨度很大的数据仓库体系中，比如云音乐数据仓库，内部有很多业务线数据仓库分支，对于同一份数据，必须保证一致性。例如用户id,必须保证所有业务线的用户id数据类型、长度等一致。
4. **唯一性**：对于需要保证主键唯一的产出表，必须保证主键的唯一性，比如维表、实体聚集表。
5. **及时性**：在确保数据的完整性、准确性和一致性等后，接下来就要保障数据能够及时产出，这样才能体现数据的价值。

实现这五大标准，可以采用事前质量测试、时候质量&调度监控相结合的方式。比如易数有专门的数据质量中心，支持表和字段级别。任务上线后，尽量配置下质量监控。

五、数据资产管理

好的规范需要有专业的工具来管理，单靠文档和人工肯定会出错的。比如我们使用的是网易内部数据中台产品：【指标管理系统】和【模型设计平台】。实现整个数据资产管理，它的功能如下：



- 统一词根、修饰词、公共字段管理，保证原子指标和衍生指标的一致性
- 统一指标管理，保证需求描述、计算口径、指标业务口径数据来源的一致性。
- 统一维度管理，保证了维度定义、维度值的一致性。
- 统一模型层次、维表、事实表的关联，保证模型设计的一致性
- 统一数据域、业务过程管理，保证数仓体系的一致性