版本：V1.1
最后修改日期：2021/03/23
微信公众号：码上观世界

1. 数据模型设计目标

为使下游数据使用方低成本获取一致性的可靠数据服务，数据模型设计方需要达到如下目标：

- 成本：模型设计者要平衡性能和成本要素对数据模型的影响，现有海量大数据情况下，以保障业务和性能为前提，合理使用数据模型方案和存储策略，尽量消除不必要的数据复制与冗余。
- 性能：模型设计者需要兼顾模型刷新性能开销、产出时间和访问性能。
- 数据一致性及数据互通：各个数据模型或者数据表之间必须保障数据输出的一致性，相同粒度的相同数据项（指标、维度）应具有相同的字段名称和业务描述，不同算法的业务指标应显性化区分。
- 数据质量：数据公共层模型需要屏蔽上游垃圾数据源，一方面要保障数据本身的高质量，减少数据缺失、错误、异常等情况的发生；另一方面要保障其对应的业务元数据的高质量，数据有明确的业务含义，为数据提使用者供正确的指引。
- 易用：在保障以上目标的前提下，数据用户能从业务角度出发快速找到所需数据；能较快的掌握模型的适用场景和使用方法；能相对便捷获取数据。

2. 数据模型设计指导思想

数据模型设计以ER模型、维度模型和宽表模型理论为指导以及阿里巴巴数据仓库建设实践为经验参考。

2.1 ER模型

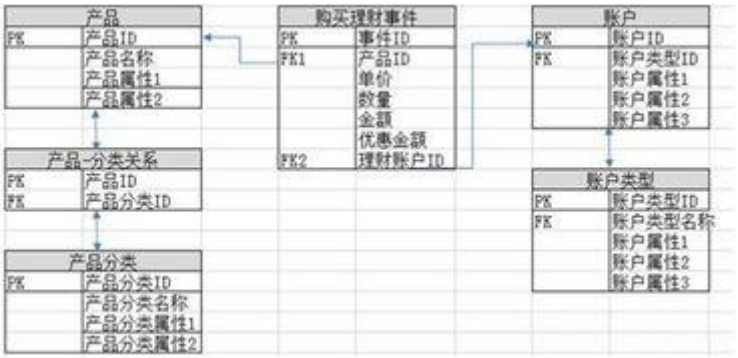
数据仓库之父Bill Inmon 提出的建模方法是从全企业的高度设计一个3NF 模型，用实体关系（ Entity Relationship, ER ）模型描述企业业务，在范式理论上符合3NF 。数据仓库中的3NF 与OLTP 系统中的3NF的区别在于，它是站在企业角度面向主题的抽象，而不是针对某个具体业务流程的实体对象关系的抽象。其具有以下几个特点：

- 需要全面了解企业业务和数据。
- 实施周期非常长。
- 对建模人员的能力要求非常高。

2.2 维度模型

数据仓库领域的Ralph Kimball 大师所倡导的，维度建模从分析决策的需求出发构建模型，为分析需求服务，因此它重点关注用户如何更快速地完成需求分析，同时具有较好的大规模复杂查询的响应性能。其典型的代表是星形模型，以及在一些特殊场景下使用的雪花模型。维度模型的维度退化即是宽表模型。

ER模型是一种范式模型，ER模型和维度模型虽然建模工具类似，比如都使用实体关系来表示，主要区别在于：
1. 着眼点不同：维度建模着眼点在产生事实的业务过程，ER模型着眼点在实体和实体的关系，ER模型的关系更为一般化。
2. ER模型的实体通常是具有业务价值的业务对象，比如商品，客户等，维度模型的维度更着重业务检索需求，如日期、地域、商品等，如下图示例。



3. 数据模型设计基本原则

- 高内聚和低耦合：软件设计方法论中的高内聚和低耦合原则同样适用于数据建模，这主要从数据业务特性和访问特性两个角度来考虑：将业务相近或者相关的数据、粒度相同数据设计为一个逻辑或者模型；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。
- 核心模型与扩展模型分离：建立核心模型与扩 展模型体系，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，必要时让核心模型与扩展模型做关联，不能让扩展字段过度侵入核心模型，破坏了核心模型的架构简洁性与可维护性。
- 公共处理逻辑下沉及单一：越是底层公用的处理逻辑更应该在数据调度依赖的底层进行封装与实现，不要让公共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。
- 成本与性能平衡：适当的数据冗余换取查询和刷新性能，不宜过度冗余与数据复制。
- 数据可回滚（数据生成支持幂等性）：处理逻辑不变，在不同时间多次运行数据结果确定不变。
- 一致性：相同的字段在不同表字段名相同，字段值相同。
- 命名清晰可理解：表命名规范需清晰、一致，表名需易于下游理解和使用。

4. 数据模型设计步骤总览

4.1 数据模型设计总体步骤

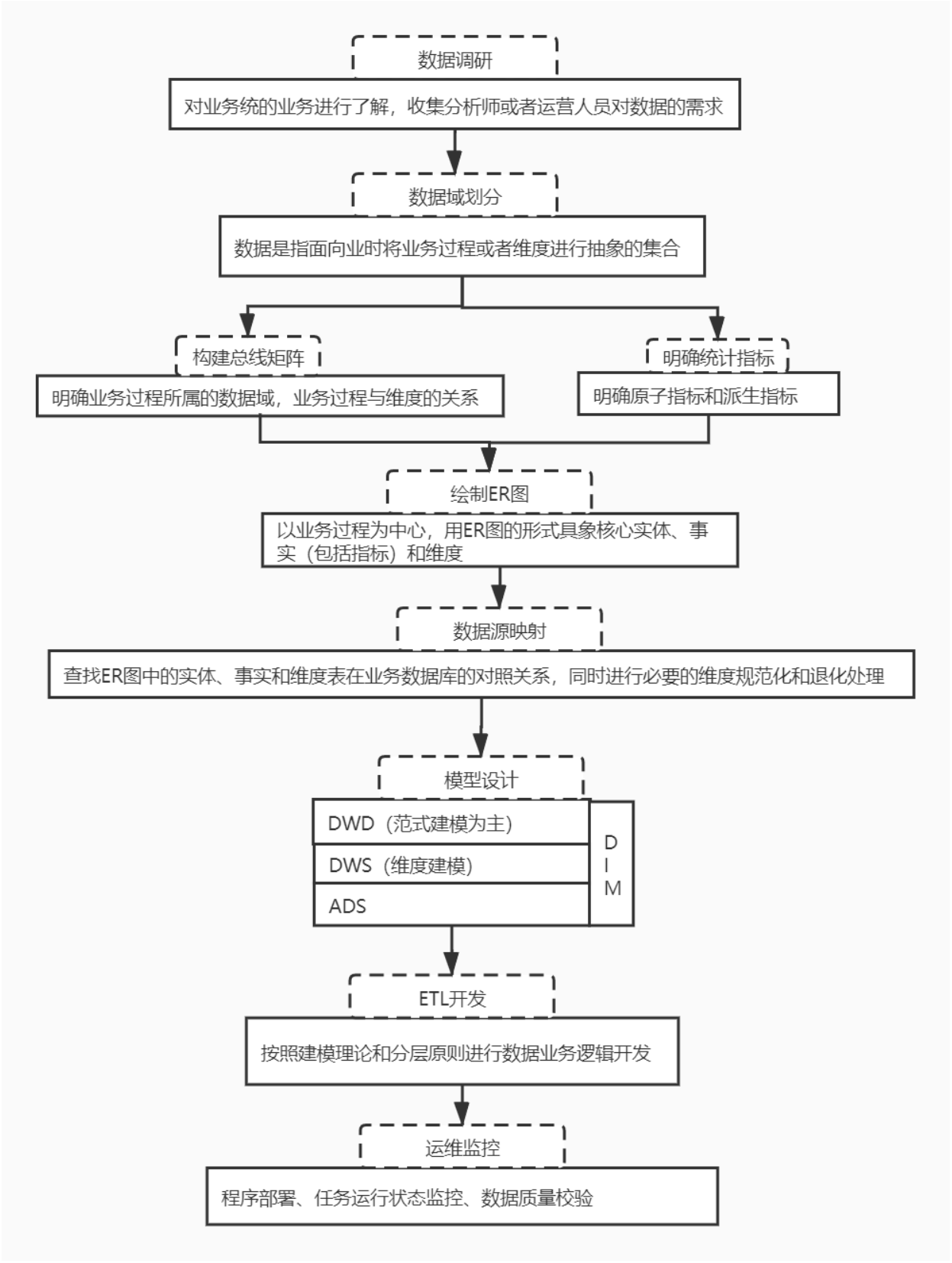
- 业务建模：生成业务模型，主要解决业务层面的分解和程序化，常用工具如流程图、时序图、活动图、数据流图、用例图等，以及常用的业务流程分析如价值链分析、客户关系分析法、供应链分析法等
- 领域建模：生成领域模型，主要是对业务模型进行抽象处理， 生成领域概念模型，这一步中会涉及到概念的分组(主题)，比如Teradata FS-LDM模型将金融业的领域概念划分成10大主题:当事人、产品、协议、事件、资产、财务、机构、地域、营销、渠道。
- 逻辑建模：生成逻辑模型，主要是将领域模型的概念实体以及实体之间的关系进行数据库层次的逻辑化。
- 物理建模：生成物理模型，主要解决逻辑模型针对不同关系数据库的物理化以及性能等一些具体的技术问题。

上述步骤是模型从抽象化到具体落地的过程，也反应了模型的不同抽象层次。每个层次内部还可以继续按照层次具体分解，比如业务模型还可以分为顶层模型，业务域，业务流程，业务环节。

4.2 数据模型设计实施流程

各步骤按照实施过程涉及到的内容见下图：

业务调研和需求分析是数据仓库建设的基石。在代码开发之前，数据架构设计主要是根据数据域对数据进行划分；按照维度建模理论，构建总线矩阵、抽象出业务过程和维度。再次，根据需求抽象整理出相关指标体系。数仓建模主要关注从业务建模领域建模到逻辑建模的过程，其中业务建模和领域建模是基础，逻辑建模是核心。本文将上述实施步骤通过流程梳理如下：



5. 数据模型设计术语名词解释

名词术语	解释
主题（域）	主题（Subject）是在较高层次上将企业信息系统中的数据进行综合、归类和分析利用的一个抽象概念，每一个主题基本对应一个宏观的分析领

	域。在逻辑意义上，它是对应企业中某一宏观分析领域所涉及的分析对象。例如“销售分析”就是一个分析领域，因此这个数据仓库应用的主题就是“销售分析”。主题域是对某个主题进行分析后确定的主题的边界。
数据域	指面向业务分析，将业务过程或者维度进行抽象的集合。其中， 业务过程可以概括为一个个不可拆分的行为事件， 在业务过程之下， 可以定义指标； 维度是指度量的环境，如买家下单事件，买家是维度。为保障整个体系的生命力， 数据域是需要抽象提炼，并且长期维护和更新的， 但不轻易变动。在划分数据域时， 既能涵盖当前所有的业务需求，又能在新业务进入时无影响地被包含进已有的数据域中和扩展新的数据域。
业务板块	业务板块定义了数据仓库的多种命名空间，是一种系统级的概念对象。当数据的业务含义存在较大差异时，您可以创建不同的业务板块，让各成员独立管理不同的业务，后续数据仓库的建设将按照业务板块进行划分。
业务过程	指企业的业务活动事件，如下单、支付、退款都是业务过程。请注意，业务过程是一个不可拆分的行为事件， 通俗地讲，业务过程就是企业活动中的事件
业务限定	统计的业务范围，用于筛选出符合业务规则的记录（类似于SQL中where后的条件，不包括时间区间）。原子指标是计算逻辑的标准化定义，业务限定则是条件限制的标准化定义。
时间周期	用来明确数据统计的时间范围或者时间点，如最近30 天、自然周、截至当日等
修饰类型	是对修饰词的一种抽象划分。修饰类型从属于某个业务域，如日志域的访问终端类型涵盖无线端、PC 端等修饰词
修饰词	指除了统计维度以外指标的业务场景限定抽象。修饰词隶属于一种修饰类型，如在日志域的访问终端类型下， 有修饰词PC 端、无线端等
统计粒度	统计分析的对象或视角，定义数据需要汇总的程度，可以理解为聚合运算时的分组条件（类似于SQL中group by的对象）。粒度是维度的一个组合，指明您的统计范围。例如，某个指标是某个卖家在某个省份的成交额，则粒度就是卖家、省份这两个维度的组合。如果您需要统计全表的数据，则粒度为全表。在指定粒度时，您需要充分考虑到业务和维度的关系。统计粒度也被称为粒度，是维度或维度组合，一般用于派生指标构建，是汇总表的唯一性识别方式。
指标	指标分为原子指标和派生指标。派生指标是以原子指标为基准，组装统计粒度、统计周期及业务限定而生成的。 原子指标是对指标统计口径、具体算法的一个抽象。根据计算逻辑复杂性，Dataphin将原子指标分为两种： 原生的原子指标：例如支付金额。 衍生原子指标：基于原子指标组合构建。例如，客单价通过支付金额除以买家数组合而来。 派生指标是业务中常用的统计指标。为保证统计指标标准、规范、无二义性地生成，OneData方法论将派生指标抽象为四部分：派生指标=原子指标+业务限定+统计周期+统计粒度。
维度	维度是度量的环境，用来反映业务的一类属性， 这类属性的集合构成一个维度， 维度也可以称为实体对象。维度属于一个数据域，如地理维度（其中包挤罔家、地区、

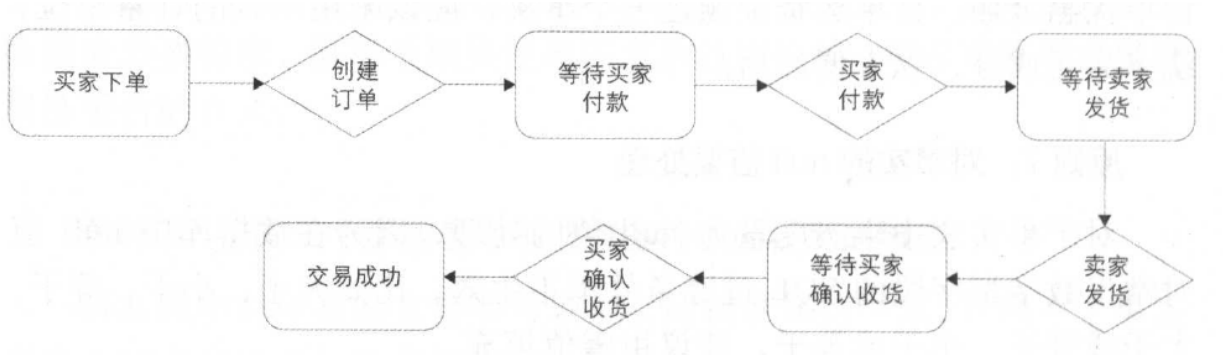
	省以及城市等级别的内容）、时间维度（其中包括年、季、月、周、日等级别的内容）
维度属性	维度属性隶属于一个维度， 如地理维度里面的国家名称、同家ID 、省份名称等都属于维度属性
派生指标	派生指标 = 一个原子指标 + 多个修饰词（可选） + 时间周期。可以理解为对原子指派生指标标业务统计范罔的圈定。如原子指标： 支付金额，最近l天海外买家支付金额则为派生指标（最近l 天为时间周期， 海外为修饰词， 买家作为维度，而不作为修饰词）

注意：
这里有两个概念：主题域和数据域，两者概念相似，都是从业务上进行领域划分，只是角度不同。主题从高层业务视角来划分，面向业务。数据域从系统数据打通的角度来划分，面向数据。数据域可能涉及到多主题域，主题域也可能涉及到多数据域。比如订单数据域会涉及到不同的业务线和主题域，在面向主题分析时候，可以针对订单分类型建立不同的主题。

6. 数据模型设计实施过程

6.1 数据调研

通过跟业务分析师或者运营人员了解数据需求，借助各种分析工具对涉及到的业务流程进行梳理和业务领域划分，形成流程图，比如这样一个示例业务过程：



这一步骤输出的文档包括业务流程图、时序图、活动图、用例图等。

6.2 数据域划分

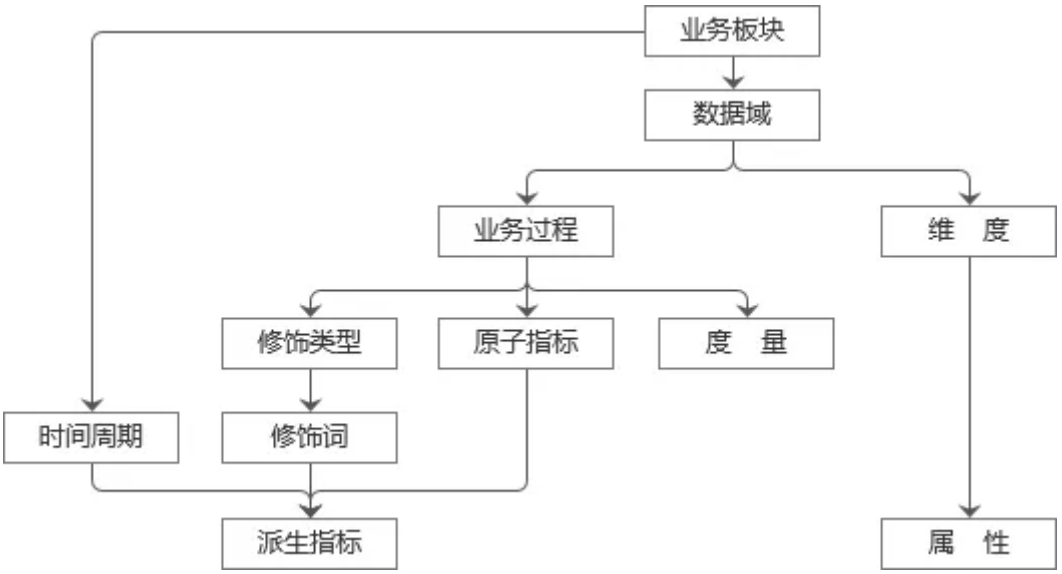
数据仓库是面向主题的应用，主要功能是将数据综合、归类并进行分析利用。数据仓库模型设计除横向的分层外，通常还需要根据业务情况纵向划分数据域。数

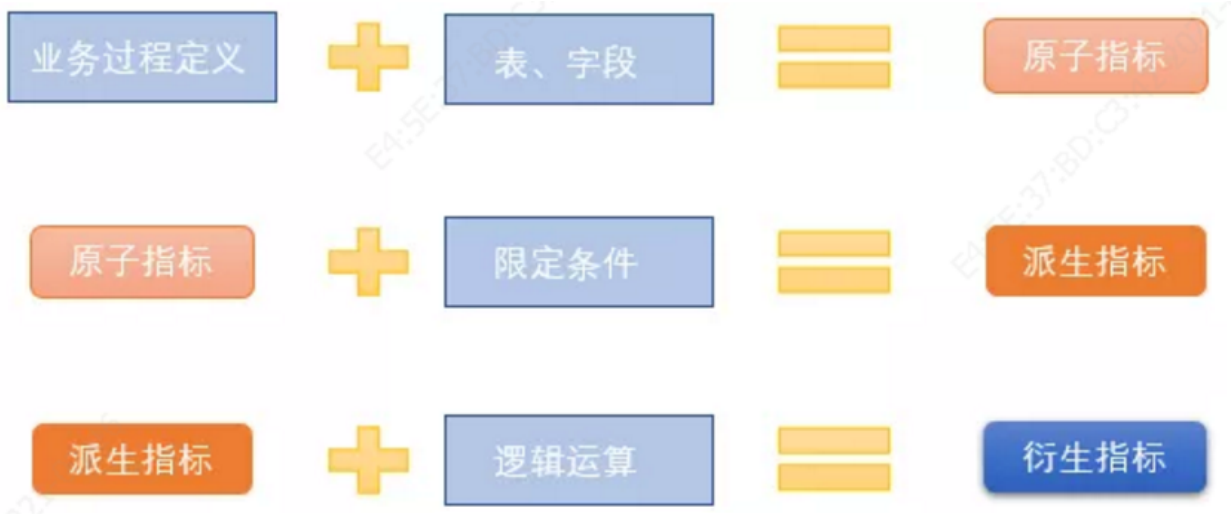
据域是联系较为紧密的数据主题的集合，是业务对象高度概括的概念层次归类，目的是便于数据的管理和应用。
通常您需要阅读各源系统的设计文档、数据字典和数据模型设计文档，研究逆向导出的物理数据模型。然后进行跨源的主题域合并，梳理出整个企业的数据域。划分数据域，需要分析各个业务模块中有哪些业务活动。数据域，可以按照用户企业的部门划分，也可以按照业务过程或者业务板块中的功能模块划分。例如，A公司电商营销业务板块可以划分为如下表所示的数据域。数据域中的每一部分，都是根据实际业务过程进行归纳、抽象得出的。

输出文档：数据域和业务过程分类关系，格式可以是类似下面的图例：

数据域	业务过程举例
会员和店铺域	注册、登录、装修、开店、关店
商品域	发布、上架、下架、重发
日志域	曝光、浏览、单击
交易域	下单、支付、发货、确认收货（交易成功）
服务域	商品收藏、拜访、培训、优惠券领用
采购域	商品采购（供应链管理）

6.3 指标规范定义





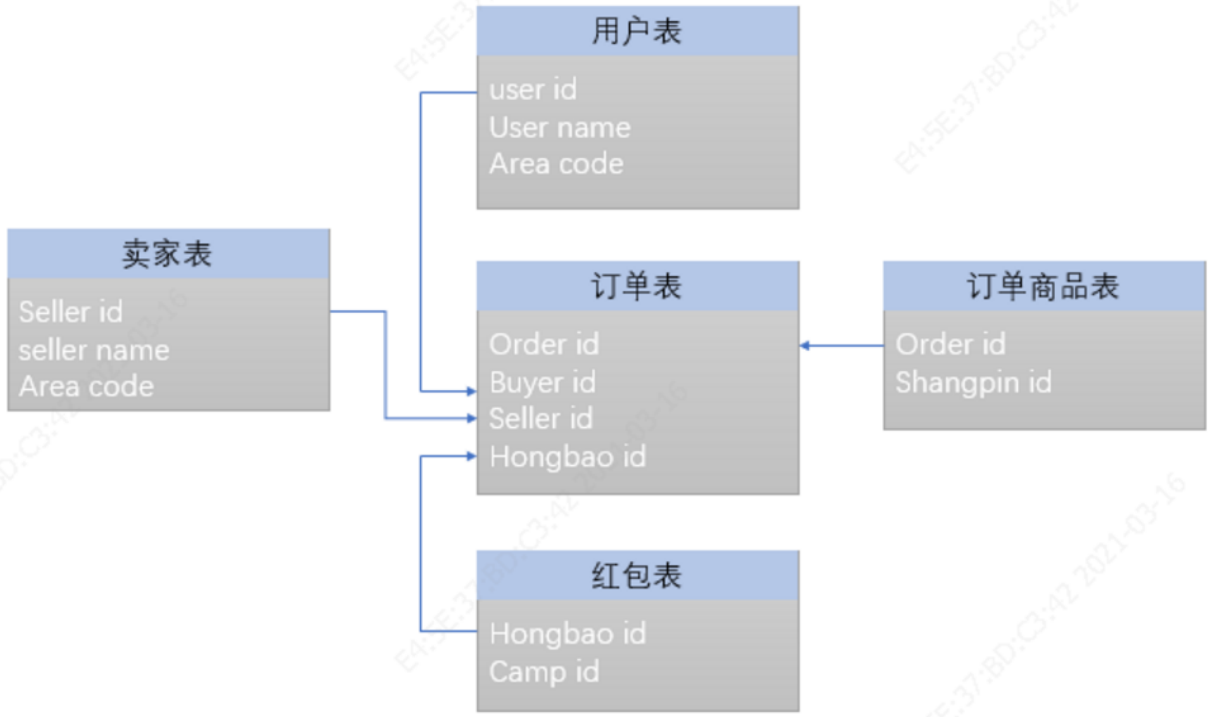
指标类型	名称命名规则	名称举例	字段命名规则	字段命名规则2
原子指标	业务流修饰+基础度量词根	如支付（业务流修饰）金额（度量）	英文（或缩写）合并，以_分隔	order_amt
派生指标	日期修饰+业务限定+聚合修饰+原子指标	近七天（统计周期）使用红包订单（业务限定）累计（聚合方式）支付金额（原子指标）	英文（或缩写）合并，以_分隔	7d_hongbao_sum_order_amt
衍生指标	日期修饰+业务限定+聚合修饰+原子指标+业务目标	近七天（统计周期）使用红包订单（业务限定）累计（聚合方式）支付金额（原子指标）占总支付金额比	英文（或缩写）合并，以_分隔	7d_hongbao_sum_order_amt_rate

指标规范定义输出文档格式为：

数据域	业务过程	指标	年月	客户类型	行政区	分公司	营业厅	服务站	其他维度
									抄表员
	抄表	完成抄表数	✓	✓	✓	✓		✓	✓
		异常抄表数	✓	✓	✓	✓		✓	✓
		计划抄表客户数	✓	✓	✓	✓		✓	✓
		计划抄表数	✓	✓	✓	✓		✓	✓
	缴费	缴费金额	✓	✓	✓	✓			收款单位
			✓	✓	✓	✓			✓
		缴费笔数	✓	✓	✓	✓	✓	✓	✓
	用气	用气量							抄表方式
			✓	✓	✓	✓			✓
销售	售气								
		进账金额明细							

6.4 ER模型梳理

这一步骤是按照自下向上快速构建数仓的方式，对业务需要的业务过程建模。细化领域模型中得出的实体及其属性、维度及其属性、业务过程及其维度和度量、指标，输出ER图文档。

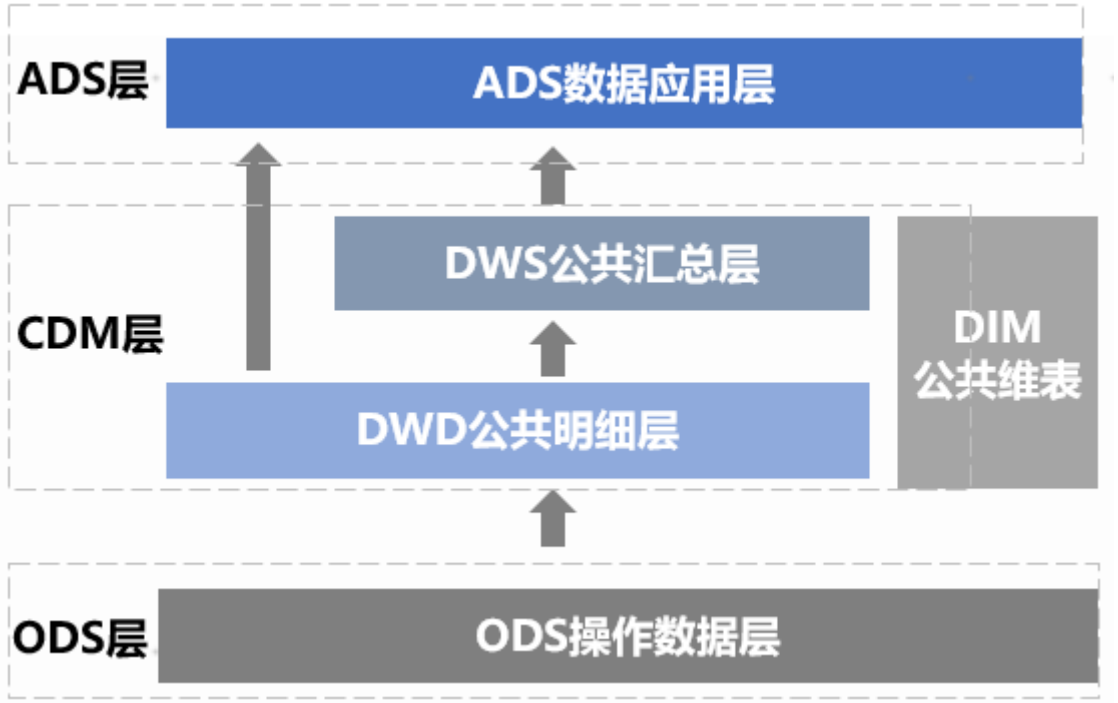


6.5 数据域映射

该步骤将ER图中每个业务过程涉及到的业务表及其数据来源维度、指标等信息列举出来,输出数据源映射文档。

数据域	业务过程	数据源	数据源类型	业务表	负责人与账号信息等	表类型	名称
现场活动	活动工单	CIS	oracle	CI_FA		事实表	现场活动表
				CI_BSEG		事实表	账单表
				CM_SGMX		事实表	事故明细
				CI_FO		事实表	活动工单
				CI_SA		维度表	服务协议
				CI_ACCT		维度表	账号表
				CI_SP		维度表	客户表
				CI_ACCT_MGMT_GR_L		维度表	服务站
				CI_CIS_DIVISION_L		维度表	分公司
				CI_SP		维度表	服务点
销售域	抄表	CIS	oracle				

6.6 数仓分层



操作数据层（ODS）：把操作系统数据几乎无处理地存放在数据仓库系统中。包括全量和增量同步的结构化数据、经过处理后的非结构化数据以及根据数据业务需求及稽核和审计要求保存历史数据、清洗数据。

- 数据公共层CDM（Common Data Model，又称通用数据模型层），包括DIM维度表、DWD和DWS，由ODS层数据加工而成。主要完成数据加工与整合，建立一致性的维度，构建可复用的面向分析和统计的明细事实表，以及汇总公共粒度的指标。
    - 公共维度层（DIM）：基于维度建模理念思想，建立整个企业的一致性维度。降低数据计算口径和算法不统一风险。

公共维度层的表通常也被称为逻辑维度表，维度和维度逻辑表通常一一对应。

    - 公共汇总粒度事实层（DWS）：以分析的主题对象作为建模驱动，基于上层的应用和产品的指标需求，构建公共粒度的汇总指标事实表，以宽表化手段物理化模型。构建命名规范、口径一致的统计指标，为上层提供公共指标，建立汇总宽表、明细事实表。

公共汇总粒度事实层的表通常也被称为汇总逻辑表，用于存放派生指标数据。

    - 明细粒度事实层（DWD）：以业务过程作为建模驱动，基于每个具体的业务过程特点，构建最细粒度的明细层事实表。可以结合企业的数据使用特点，将明细事实表的某些重要维度属性字段做适当冗余，即宽表化处理。

明细粒度事实层的表通常也被称为逻辑事实表。

- 数据应用层ADS（Application Data Service）：存放数据产品个性化的统计指标数据。根据CDM与ODS层加工生成。

7. 数据模型设计开发规范

7.1 公共规范约定

7.1.1 层次调用规范

应用层应优先调用公共层数据，必须存在中间层数据，不允许应用层跨过中间层从ODS层重复加工数据。一方面，中间层团队应该积极了解应用层数据的建设需求，将公用的数据沉淀到公共层，为其他团队提供数据服务；另一方面，应用层团队也应积极配合中间层团队进行持续的数据公共建设的改造。必须避免出现过度的引用ODS层、不合理的数据复制以及子集合冗余。

- ODS层数据不能被应用层任务引用，中间层不能有沉淀的ODS层数据，必须通过CDM层的视图访问。CDM层视图必须使用调度程序进行封装，保持视图的可维护性与可管理性。
- CDM层任务的深度不宜过大（建议不超过10层）。
- 原则上一个计算刷新任务只允许一个输出表。
- CDM汇总层应优先调用CDM明细层。在调用可累加类指标计算时，CDM汇总层尽量优先调用已经产出的粗粒度汇总层，以避免大量汇总直接从海量的明细数据层计算。
- CDM明细层累计快照事实表优先调用CDM事务型事实表，以保持数据的一致性产出。
- 避免应用层过度引用和依赖CDM层明细数据，需要针对性地建设好CDM公共汇总层。

7.1.2 数据类型规范

ODS层的数据类型应基于源系统数据类型转换。例如，源数据为MySQL时的转换规则如下。

MySQL数据类型	数仓数据类型
TINYINT	TINYINT
SMALLINT/MEDIUMINT	SMALLINT
INTEGER	INT

BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
DECIMAL	DECIMAL
CHAR/VARCHAR	VARCHAR
LONGTEXT/TEXT	STRING
DATE/TIMESTAMP/TIME/YEAR	STRING
DATETIME	DATETIME

CDM数据公共层如果是引用ODS层数据，则默认使用ODS层字段的数据类型。其衍生加工数据字段按以下标准执行：

- 金额类及其它小数点数据使用DOUBLE类型。
- 字符类数据使用STRING类型。
- ID类和整形数值使用BIGINT类型。
- 时间类型数据使用STRING类型（如果有特殊的格式要求，可以选择性使用DATETIME类型）。
- 状态使用STRING类型。

7.1.3 数据冗余

一个表做宽表冗余维度属性时，应该遵循以下建议准则：

- 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。
- 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。
- 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。
- 数据统计日期的分区字段按以下标准：按天分区：ds(YYYYMMDD)。
- 原则上不需要冗余分区字段。

7.1.4 数据拆分

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数据值、零数据值在行列二维空间上分布情况进行划分的。

- 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。
- 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。
- 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。

7.2 ODS层模型设计开发规范

ODS层存放业务系统获取的最原始的数据，是其他上层数据的源数据。业务数据系统中的数据通常为非常细节的数据，经过长时间累积，且访问频率很高，是面向应用的数据。

7.2.1 数据同步及处理规范

- 所有ODS层的表都以统计日期及时间分区表方式存储，数据成本由存储管理和策略控制。
- 如果源系统新增了字段，您需要重新配置数据集成同步作业。如果目标表的字段在源系统中不存在，数据集成自动填充NULL。
- 一个系统的源表只允许同步一次到ODS层，通过手动恢复性重跑，必须能够覆盖之前的数据，不能引入数据重复、数据不一致等情况。

7.2.2 命名规范

- 表命名规范
  - 表或字段命名尽量和业务系统保持一致，但是需要通过额外的标识来区分增量和全量表。
  - 增量数据：{层次}_{源系统库名}_{源系统表名}_di。（di表示事务型增量表）
  - 全量数据：{层次}_{源系统库名}_{源系统表名}_df。（df表示周期型全量表）
  - ODS ETL过程的临时表：{层次}_tmp_{临时表所在过程的输出表}_{从0开始的序号}。
- 字段命名规范
  - 字段默认使用源系统的字段名。
- 同步任务命名规范
  - 任务名：{源系统库名}_{源系统表名}_{di|df}。

7.2.3 数据存储及生命周期管理规范

数据表类型	存储方式	最长存储保留策略
ODS流水型全量表	按天分区	• 不可再生情况下，永久保存。 • 日志（数据量非常大，例如一天数据量大于100 GB）数据保留24个月。 • 自主设置是否保留历史月初数据。 • 自主设置是否保留特殊日期数据。
ODS镜像型全量表	按天分区	• 重要的业务表及需要保留历史的表视情况保存。 • ODS全量表的默认生命周期为2天，支持通过 `ds=max_pt(tablename)` 方式访问数据。
ODS增量表	按天分区	• 有对应全量表，最多保留最近14天分区数据。 • 无对应全量表，需要永久保留数据。
ODS ETL过程临时表	按天分区	最多保留最近7天分区。

7.2.4 数据质量规范

- 每个ODS全量表必须配置唯一性字段标识。
- 每个ODS全量表必须有注释。
- 每个ODS全量表必须监控分区空数据。
- 仅有监控要求的ODS表才需要创建数据质量监控规则。
- 建议对重要表的重要枚举类型字段进行枚举值变化及枚举值分布监控。
- 建议对ODS表的数据量及数据记录数设置周同环比监控，如果周同环比无变化，表示源系统已迁移或下线。

7.3 维度层设计开发规范

维度是维度建模的基础和灵魂。在维度建模中，将度量称为"事实"，将环境描述为"维度"，维度是用于分析事实所需要的多样环境。维度所包含的表示维度的列，称为维度属性。维度属性是查询约束条件、分组和报表标签生成的基本来源，是数据易用性的关键。维度的作用一般是查询约束、分类汇总以及排序等。

维度的设计过程就是确定维度属性的过程，如何生成维度属性，以及所生成的维度属性的优劣，决定了维度使用的方便性，成为数据仓库易用性的关键。正如Kimball所说的，数据仓库的能力直接与维度属性的质量和深度成正比。

7.3.1 维度表设计原则

- 作为维度建模的核心，在企业级数据仓库中必须保证维度的唯一性。
- 尽可能生成丰富的维度属性。
- 尽可能多地给出包括一些富有意义的文字性描述。
- 区分数值型属性和事实。数值型字段是作为事实还是维度属性，可以参考字段的一般用途。如果通常用于查询约束条件或分组统计，则是作为维度属性；如果通常用于参与度量的计算， 则是作为事实。
- 尽量沉淀出通用的维度属性。但不同来源的数据表在数仓中集成时，可能会整合或者拆分，考虑的依据是保持核心模型的相对稳定，同时兼顾性能和易用性。
- 维度缓慢变化问题可以有三种解决办法：重写维度值；插入新的维度行；添加维度列。对于选择哪种方式处理缓慢变化维，并没有一个完全正确的答案，可以根据业务需求来进行选择。
- 对长期不使用的大数据量的维表可考虑数据归档。
- 带有递归层次结构的维度可以做扁平化处理，如将商品不同层次的类目整合在一条记录。
- 对于行为维度，有两种处理方式，其中一种是将其冗余至现有的维表中，如将卖家信用等级冗余至卖家维表中；另一种是加工成单独的行为维表，如卖家主营类目。具体采用哪种方式主要参考如下两个原则：避免维度过快增长和避免稠合度过高。
- 事实表的一条记录在某维表中如果有多条记录与之对应。这类多值维度可以根据业务的表现形式
- 和统计分析需求有3种选择：降低事实表的粒度、采用多字段和用桥接表。

7.3.4 维度表设计步骤

- 选择维度或新建维度

作为维度建模的核心，在企业级数据仓库中必须保证维度的唯一性。以淘宝商品维度　　为例，有且只允许有一个维度定义。

- 确定主维表

此处的主维表一般是ODS表，直接与业务系统同步。

- 确定相关维表

数据仓库是业务源系统的数据整合，不同业务系统或者同一业务系统中的表之间存在关联性。根据对业务的梳理，确定哪些表和主维表存在关联关系，并选择其中的某些表用于生成维度属性

- 确定维度属性

主要包括两个阶段，其中第一个阶段是从主维表中选择维度属性或生成新的维度属性；第二个阶段是从相关维表中选择维度属性或生成新的维度属性。

7.3.5 维度表命名规范

命名规则：dim_{业务/pub}_{维度定义}[_{自定义命名标签}]，其中的pub与具体业务无关，各个业务部都可以共用，例如时间维度。

7.3.6 数据存储及生命周期管理规范

CDM公共维度层的表的类型为维度表，存储方式为按天分区。

模型设计者根据自身业务需求设置表的生命周期管理。您可依据3个月内的最大需要访问的跨度设置保留策略，具体计算方式如下：

- 当3个月内的最大访问跨度小于或等于4天时，建议将保留天数设为7天。
- 当3个月内的最大访问跨度小于或等于12天时，建议将保留天数设为15天。
- 当3个月内的最大访问跨度小于或等于30天时， 建议将保留天数设为33天。
- 当3个月内的最大访问跨度小于或等于90天时，建议将保留天数设为93天。
- 当3个月内的最大访问跨度小于或等于180天时， 建议将保留天数设为183天。

- 当3个月内的最大访问跨度小于或等于365天时，建议将保留天数设为368天。

7.4 明细粒度事实层（DWD）设计开发规范

明细粒度事实层以业务过程驱动建模，基于每个具体的业务过程特点，构建最细粒度的明细层事实表。您可以结合企业的数据使用特点，将明细事实表的某些重要维度属性字段做适当冗余，即宽表化处理。

公共汇总粒度事实层（DWS）和明细粒度事实层（DWD）的事实表作为数据仓库维度建模的核心，需紧绕业务过程来设计。通过获取描述业务过程的度量来描述业务过程，包括引用的维度和与业务过程有关的度量。度量通常为数值型数据，作为事实逻辑表的依据。事实逻辑表的描述信息是事实属性，事实属性中的外键字段通过对应维度进行关联。

事实表中一条记录所表达的业务细节程度被称为粒度。通常粒度可以通过两种方式来表述：一种是维度属性组合所表示的细节程度，一种是所表示的具体业务含义。
作为度量业务过程的事实，通常为整型或浮点型的十进制数值，有可加性、半可加性和不可加性三种类型：

- 可加性事实是指可以按照与事实表关联的任意维度进行汇总。
- 半可加性事实只能按照特定维度汇总，不能对所有维度汇总。例如库存可以按照地点和商品进行汇总，而按时间维度把一年中每个月的库存累加则毫无意义。
- 完全不可加性，例如比率型事实。对于不可加性的事实，可分解为可加的组件来实现聚集。

事实表相对维表通常更加细长，行增加速度也更快。维度属性可以存储到事实表中，这种存储到事实表中的维度列称为维度退化，可加快查询速度。与其他存储在维表中的维度一样，维度退化可以用来进行事实表的过滤查询、实现聚合操作等。

7.4.1 明细粒度事实层分类

明细粒度事实层通常分为三种：事务事实表、周期快照事实表和累积快照事实表。

- 事务事实表记录的事务层面的事实，用于跟踪业务过程的行为，并支持几种描述行为的事实，保存的是最原子的数据，也称为原子事实表。。事务事实表中的数据在事务事件发生后产生，数据的粒度通常是每个事务一条记录。一旦事务被提交，事实表数据被插人，数据就不能更改，其更新方式为增量更新。
- 周期快照事实表以具有规律性的、可预见的时间间隔来记录事实，如余额、库存、层级、温度等，时间间隔为每天、每月、每年等，典型的例子如库存日快照表等。周期快照事实表的日期维度通常记录时间段的终止日，记录的事实是这个时间段内一些聚集事实值或状态度量。事实表的数据一旦插人就不能更改，其更新方式为增量更新。
- 累积快照事实表被用来跟踪实体的一系列业务过程的进展情况，它通常具有多个日期字段，用于研究业务过程中的里程碑过程的时间间隔。另外，它还会有一个用于指示最后更新日期的附加日期宇段。由于事实表中许多日期在首次加载时是不知道的，而且这类事实表在数据加载完成后，可以对其数据进行更新，来补充业务状态变更时的日期信息和事实。

7.4.2 三种事实表比较：

	事务事实表	周期快照事实表	累积快照事实表
时期/时间	离散事务时间点	以有规律的、可预测的	用于时间跨度不确定的不断变化的工作流
日期维度	事务日期	快照日期	相关业务过程涉及的多个日期
粒度	每行代表实体的一个事务	每行代表某时间周期的一个实体	每行代表一个实体的生命周期
事实	事务事实	累积事实	相关业务过程事实和时间间隔事实
事实表加载	插入	插入	插入与更新
事实表更新	不更新	不更新	业务过程变更时更新

7.4.3 明细粒度事实表设计原则

明细粒度事实表设计原则如下所示：

- 通常，一个明细粒度事实表仅和一个维度关联。
- 尽可能包含所有与业务过程相关的事实 。

- 只选择与业务过程相关的事实。
- 分解不可加性事实为可加的组件。
- 在选择维度和事实之前必须先声明粒度。
- 在同一个事实表中不能有多种不同粒度的事实。
- 事实的单位要保持一致。
- 谨慎处理Null值。
- 使用退化维度提高事实表的易用性。

7.4.4 事实表设计方法

- 选择业务过程及确定事实表类型。在明确了业务需求以后，接下来需要进行详细的需求分析，对业务的整个生命周期进行分析，明确关键的业务步骤，从而选择与需求有关的业务过程。在选择了业务过程以后，相应的事实表类型也随之确定了。
- 声明粒度。应该尽量选择最细级别的原子粒度，以确保事实表的应用具有最大的灵活性。
- 确定维度。完成粒度声明以后，也就意味着确定了主键，对应的维度组合以及相关的维度字段就可以确定了，应该选择能够描述清楚业务过程所处的环境的维度信息。比如在淘宝订单付款事务事实表中，粒度为子订单，相关的维度有买家、卖家、商品、收货人信息、业务类型、订单时间等维度。
- 确定事实。事实可以通过回答“过程的度量是什么”来确定。应该选择与业务过程有关的所有事实，且事实的粒度要与所声明的事实表的粒度一致。
- 冗余维度。事实表中冗余方便下游用户使用的常用维度，以实现对事实表的过滤查询、控制聚合层次、排序数据以及定义主从关系等操作。

7.4.5 明细表命名规范

命名规则：**dwd_{业务缩写/pub}_{数据域缩写}_{业务过程缩写}[_{自定义表命名标签缩写}]_{刷新周期标识}_{单分区增量全量标识}。**

命名说明：

- pub表示数据包括多个业务的数据。

- 单分区增量全量标识：i表示增量，f表示全量。

7.4.6 数据存储及生命周期管理规范

CDM明细层的表的类型为事实表，存储方式为按天分区。

事务型事实表一般永久保存。周期快照型事实表根据业务需求设置生命周期管理。可依据3个月内的最大需要访问的跨度设置保留策略，具体计算方式如下：

- 当3个月内的最大访问跨度小于或等于4天时，建议将保留天数设为7天。
- 当3个月内的最大访问跨度小于或等于12天时，建议将保留天数设为15天。
- 当3个月内的最大访问跨度小于或等于30天时， 建议将保留天数设为33天。
- 当3个月内的最大访问跨度小于或等于90天时，建议将保留天数设为93天。
- 当3个月内的最大访问跨度小于或等于180天时， 建议将保留天数设为183天。
- 当3个月内的最大访问跨度小于或等于365天时，建议将保留天数设为368天。

7.5 公共汇总粒度事实层（DWS）设计开发规范

公共汇总粒度事实层以分析的主题对象作为建模驱动，基于上层的应用和产品的指标需求构建公共粒度的汇总指标事实表。公共汇总层的一个表通常会对应一个派生指标。

7.5.1 汇总表设计原则

- 一致性。聚集表必须提供与查询明细粒度数据一致的查询结果。从设计角度来看，确保一致性，最简单的方法是确保聚集星形模型中的维度和度量与原始模型中的维度和度量保持一致。
- 避免单一表设计。不要在同一个表中存储不同层次的聚集数据；在聚集表中有些行存放按天汇总的交易额，有些行存放按月汇总的交易额， 这

将会让使用者产生误用导致重复计算。为了避免此类问题，可以把按天与按月汇总的交易额用两列存放，但是需要在列名或者列注释上能分辨出来。

- 聚集粒度可不同。聚集并不需要保持与原始明细粒度数据一样的粒度，聚集只关心所需要查询的维度。
- 数据公用性。如果汇总的聚集会被外部用户使用，就有必要把明细数据经过汇总沉淀到聚集表中。
- 不跨数据域。数据域是在较高层次上对数据进行分类聚集的抽象。阿里巴巴以业务过程进行分类，如交易统一划到交易域下，商品的新增、修改放到商品域下。
- 区分统计周期。在表的命名上要能说明数据的统计周期，如 _ ld表示最近l 天， td 表示截至当天， nd 表示最近N 天。

7.5.2 聚集的基本步骤

- 确定聚集维度。

在原始明细模型中会存在多个描述事实的维度，如日期、商品类别、卖家等，这时候需要确定根据什么维度聚集，如果只关心商品的交易额情况，那么就可以根据商品维度聚集数据。

- 确定一致性上钻。
- 这时候要关心是按月汇总还是按天汇总，是按照商品汇总还是按照类目汇总，如果按照类目汇总，还需要关心是按照大类汇总还是小类汇总。当然，我们要做的只是了解用户需要什么，然后按照他们想要的进行聚集。
- 确定聚集事实。

在原始明细模型中可能会有多个事实的度量，比如在交易中有交易额、交易数量等，这时候要明确是按照交易额汇总还是按照成交数量汇总。

7.5.3 命名规范

命名规则：dws_{业务缩写/pub}_{数据域缩写}_{数据粒度缩写}[_{自定义表命名标签缩写}]{统计时间周期范围缩写}_{刷新周期标识}_{单分区增量全量标识}。

命名说明：

- 在默认情况下，离线计算应该包括最近一天（1d）、最近N天（nd）和历史截至当天（td）三个表。

如果nd表的字段过多，需要拆分时，只允许以一个统计周期单元作为原子拆分，即一个统计周期拆分一个表。例如，最近7天（1w）拆分一个表，不允许拆分出来的一个表存储多个统计周期。

- 对于{刷新周期标识}和{单分区增量全量标识}在汇总层不做强制要求。单分区增量全量标识：i表示增量，f表示全量。
- 对于小时表不管是按天刷新还是按小时刷新，都用_hh来表示。
- 对于分钟表不管是按天刷新还是按小时刷新，都用_mm来表示。

7.5.4 数据存储及生命周期管理规范

CDM汇总层的表的类型为事实表，存储方式为按天分区。

事务型事实表一般会永久保存。周期快照型事实表根据业务需求设置生命周期管理。您可依据3个月内的最大需要访问的跨度设置保留策略，具体计算方式如下：

- 当3个月内的最大访问跨度小于或等于4天时，建议将保留天数设为7天。
- 当3个月内的最大访问跨度小于或等于12天时，建议将保留天数设为15天。
- 当3个月内的最大访问跨度小于或等于30天时，建议将保留天数设为33天。
- 当3个月内的最大访问跨度小于或等于90天时，建议将保留天数设为93天。
- 当3个月内的最大访问跨度小于或等于180天时，建议将保留天数设为183天。
- 当3个月内的最大访问跨度小于或等于365天时，建议将保留天数设为368天。

7.6 规范文档输出列表

业务调研流程图、用例图等

领域划分图

数据域和业务过程、维度总线矩阵文档

数据域和业务过程、指标体系文档

ER模型图

维度事实表数据源映射文档

7.7 参考

《Star Schema - The Complete Reference》

《The.Data.Warehouse.Toolkit- The Definitive Guide to Dimensional Modeling》

《Building the DataHouse Third Edition》

MaxCompute设计规范文档

DataWorks设计规范文档

Dataphine设计规范文档