Hadoop的组成部分

-- HDFS (hadoop的组成部分-负责海量数据的存储)

-- NameNode（nn）:管理真实数据的元数据的（hdfs集群中的老大）

-- DataNode（dn）:主要负责对真实数据块存储（hdfs集群中的小弟）

-- SecondaryNameNode（2nn）：主要为NameNode进行一些数据备份 一般恢复数据的时候才会用到它，它也不能保证完全数据恢复。

-- YARN （hadoop的组成部分，主要负责资源调度）

-- ResourceManager（rm）:统筹管理每一台机器上的资源，并且负责接收处理客户端作业请求。

-- NodeManager（nm）:负责单独每一台机器的资源管理，实时保证和大哥ResourceManager）通信。

-- ApplicationMaster：针对每个请求job的抽象封装

-- Container：将来运行在YARN上的每一个任务都会给其分配资源， Container就是当前任务所需资源的抽象封装

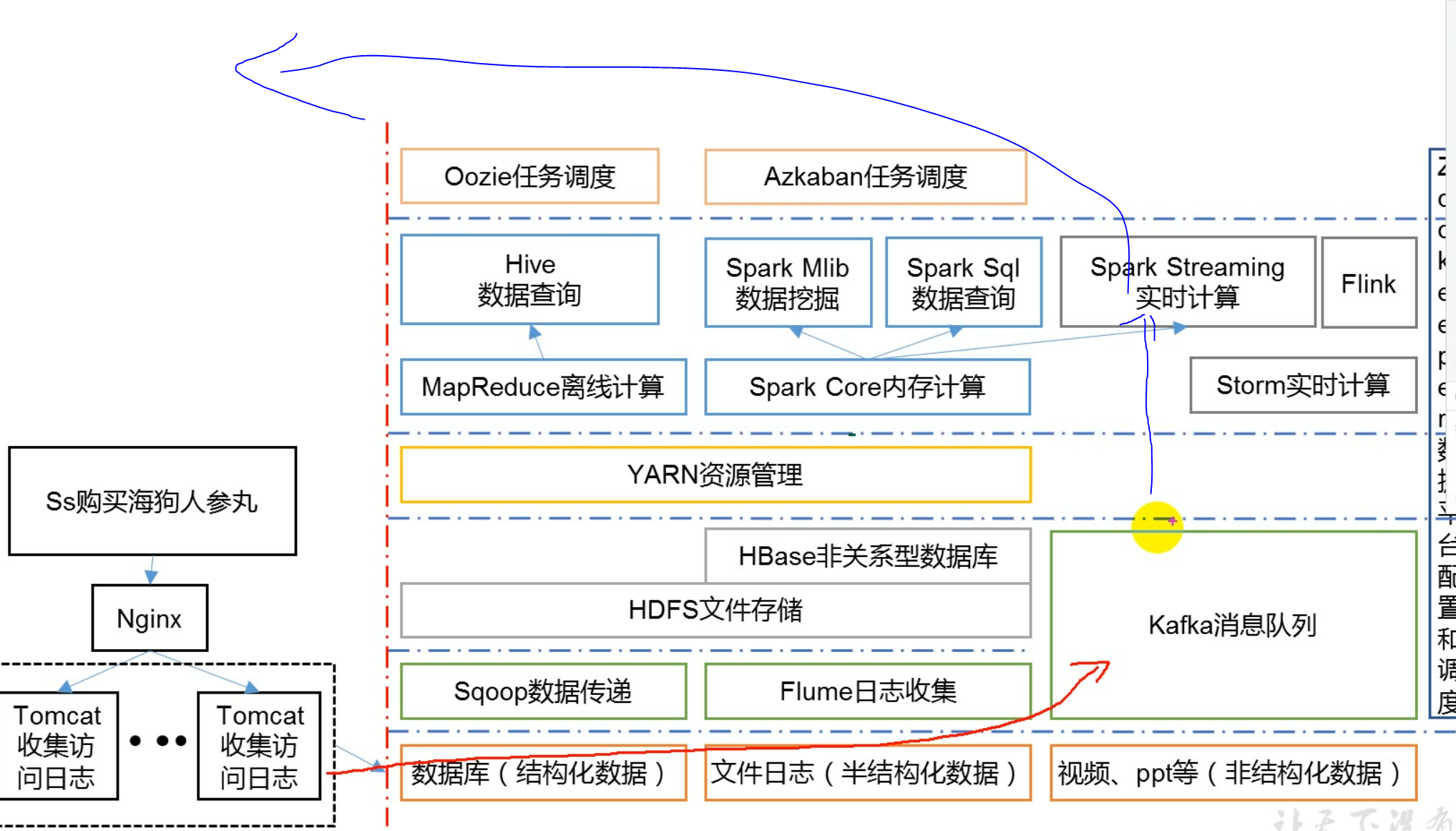
-transform

Sqoop --- database

flume ---- log data

realtime data --kafka

- storageHbase(HDFS,kafka)

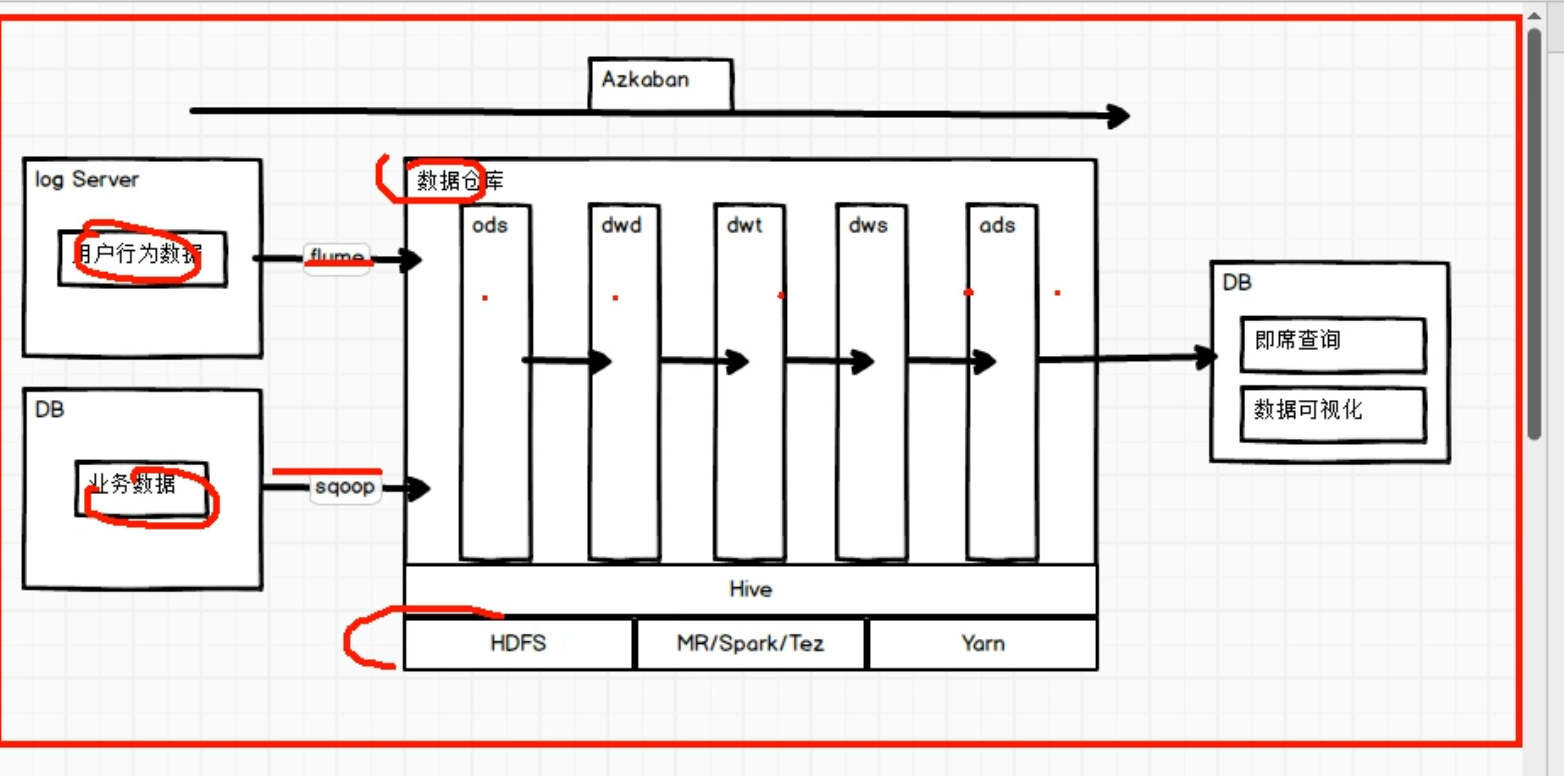


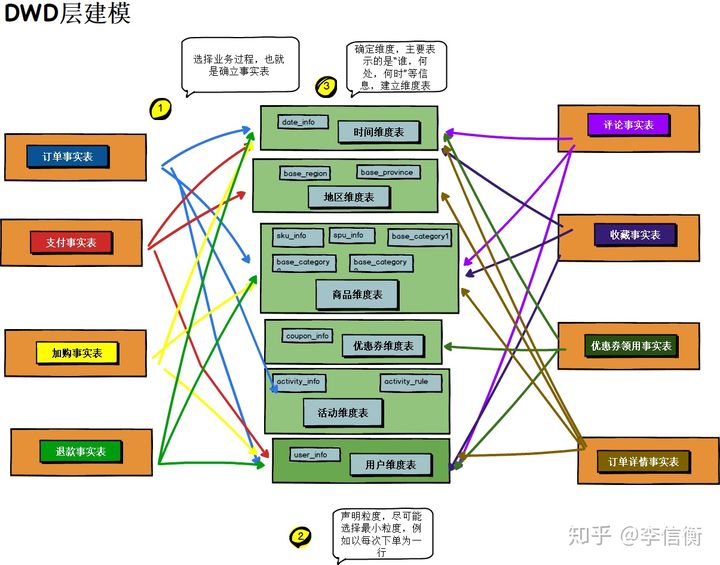
Hive no index, need to scan all before split, so high delay

Hive meta data( relation ship of forms) no storage, derby or install mysql

Hive storage in HDFs, map reduce or spark

Hive sqoop to mysql





DWD 层需构建维度模型，一般采用星型模型，呈现的状态一般为星座模型。

维度建模一般按照以下四个步骤：

选择业务过程→声明粒度→确认维度→确认事实

DWD层事实表设计原则

联系维度的外键+度量值



统计各个主题对象的当天行为，服务于DWT 层的主题宽表，以及一些业务明细数据，

应对特殊需求（例如，购买行为，统计商品复购率）

## DWS层表设计原则

通过外键获取相关的度量值，整合多个dwd事实表度量值构成新表。

## DWT层建模

以分析的主题对象为建模驱动，基于上层的应用和产品的指标需求，构建主题对象的全

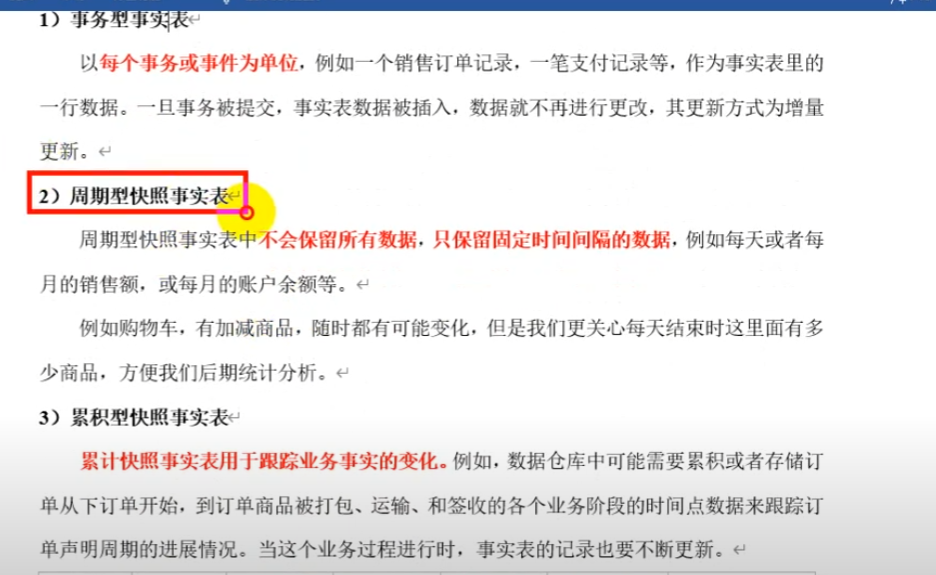
量宽表。

## DWT层事实表设计原则

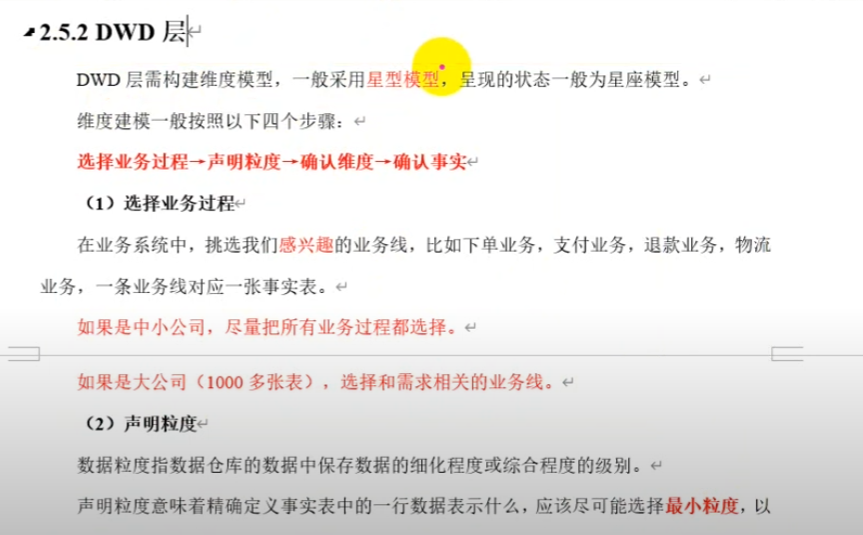
维度关联的事实表度量值+开头、结尾+累积+累积一个时间段。

比如下单

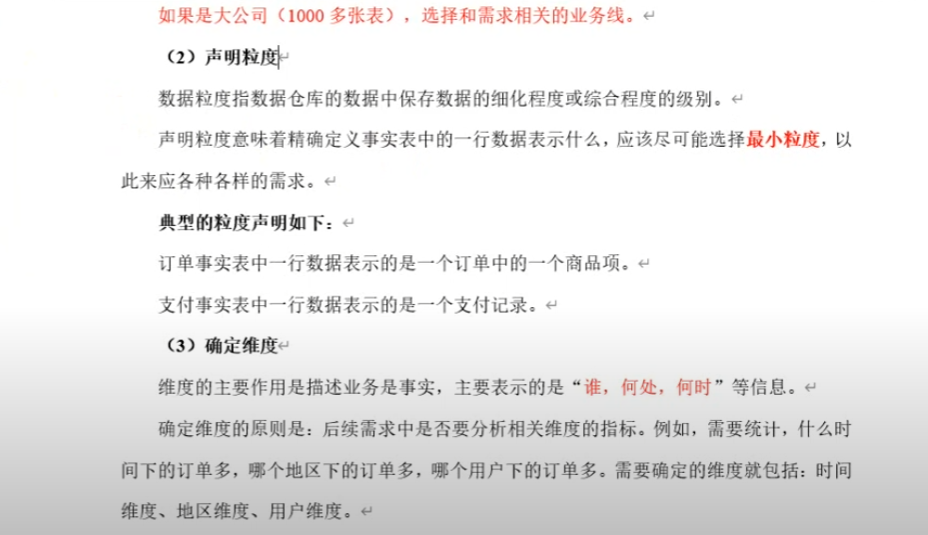




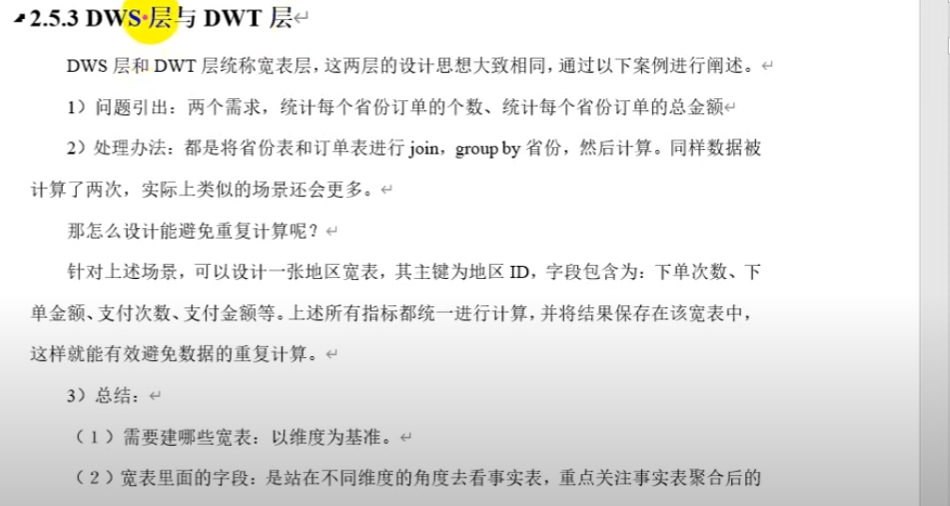
ODS:An operational **data** store (**ODS**) is a central **database** that provides a snapshot of the latest **data** from multiple transactional systems for operational



DWD: 按照内容解析日志， 声明粒度，事实表Data Warehouse Developer



粒度granularity

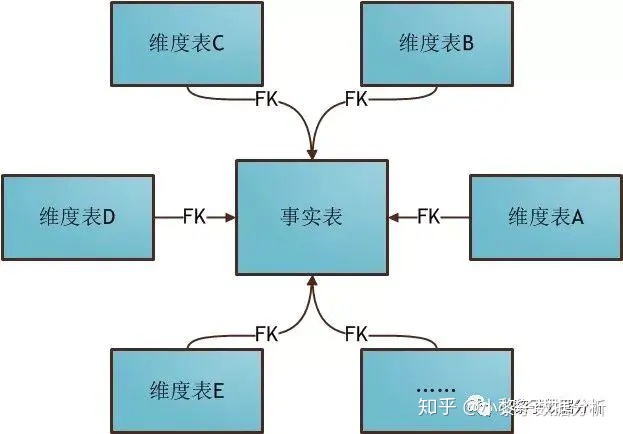


**Data** Warehouse Service 聚合值

在实际工作中多维分析的商业智能解决方案，根据事实表和维度表的关系，又可将常见的模型分为星型模型和雪花型模型。在设计逻辑型数据的模型的时候，就应考虑数据是按照星型模型还是雪花型模型进行组织。

一、星型模型 star

星型模型：是一种多维的数据关系，它由一个事实表（Fact Table）和一组维表（Dimension Table）组成。每个维表都有一个维作为主键，所有这些维的主键组合成事实表的主键。事实表的非主键属性称为事实（Fact），它们一般都是数值或其他可以进行计算的数据；如下图：



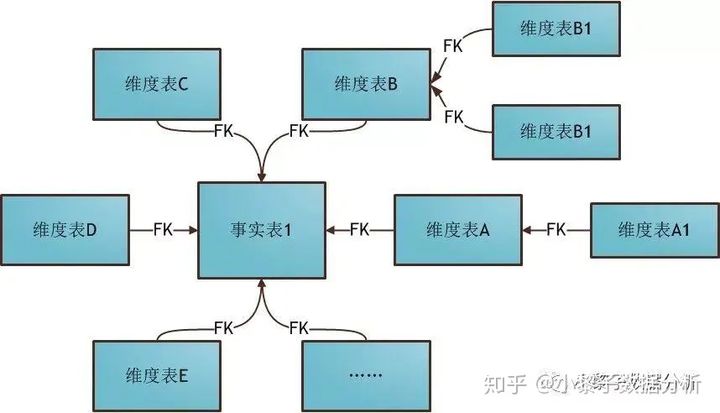
星型模型

星型架构是一种非正规化的结构，多维数据集的每一个维度都直接与事实表相连接，所以数据有一定的冗余

二、雪花型模型 snow flake

雪花型模型：当有一个或多个维表没有直接连接到事实表上，而是通过其他维表连接到事实表上时，其图解就像多个雪花连接在一起，故称雪花模型。雪花模型是对星型模型的扩展。它对星型模型的维表进一步层次化，原有的各维表可能被扩展为小的事实表，形成一些局部的 "层次 " 区域，这些被分解的表都连接到主维度表而不是事实表。

Constellation 星座



雪花型模型

通过最大限度地减少数据存储量以及联合较小的维表来改善查询性能。雪花型结构去除了数据冗余。

三、星型模型VS雪花型模型

星型模型和雪花模型的对比，可以从以下四个角度来对比。

1、查询性能角度来看

在OLTP-DW环节，由于雪花型要做多个表联接，性能会低于星型架构；但从DW-OLAP环节，由于雪花型架构更有利于度量值的聚合，因此性能要高于星型架构。

2、模型复杂度角度

星型架构更简单方便处理

3、层次结构角度

雪花型架构更加贴近OLTP系统的结构，比较符合业务逻辑，层次比较清晰。

4、存储角度

雪花型架构具有关系数据模型的所有优点，不会产生冗余数据，而相比之下星型架构会产生数据冗余。

四、总结

根据项目经验，一般建议使用星型模型。因为在实际项目中，往往最关注的是查询性能问题，至于磁盘空间一般都不是问题。当然，在维度表数据量极大，需要节省存储空间的情况下，或者是业务逻辑比较复杂、必须要体现清晰的层次概念情况下，可以使用雪花型模型。