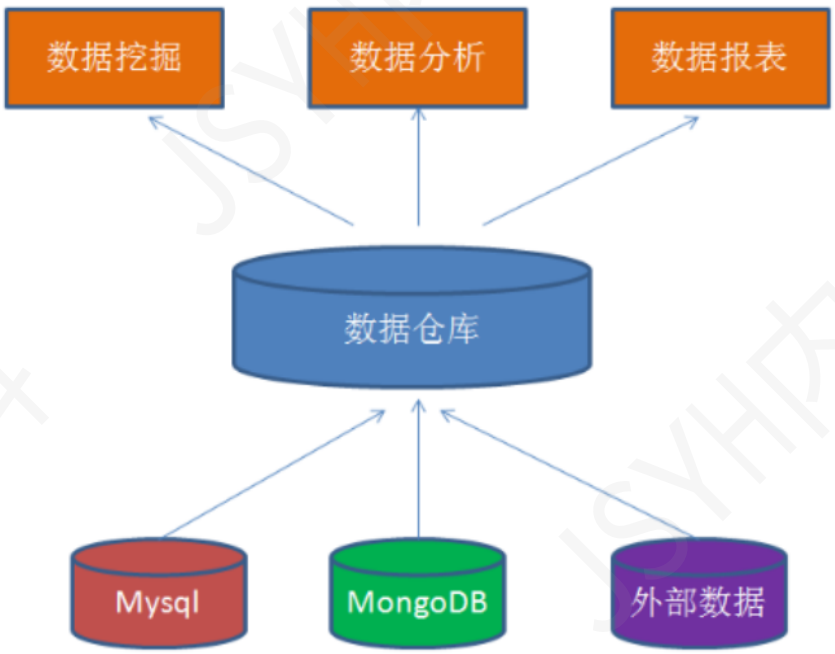
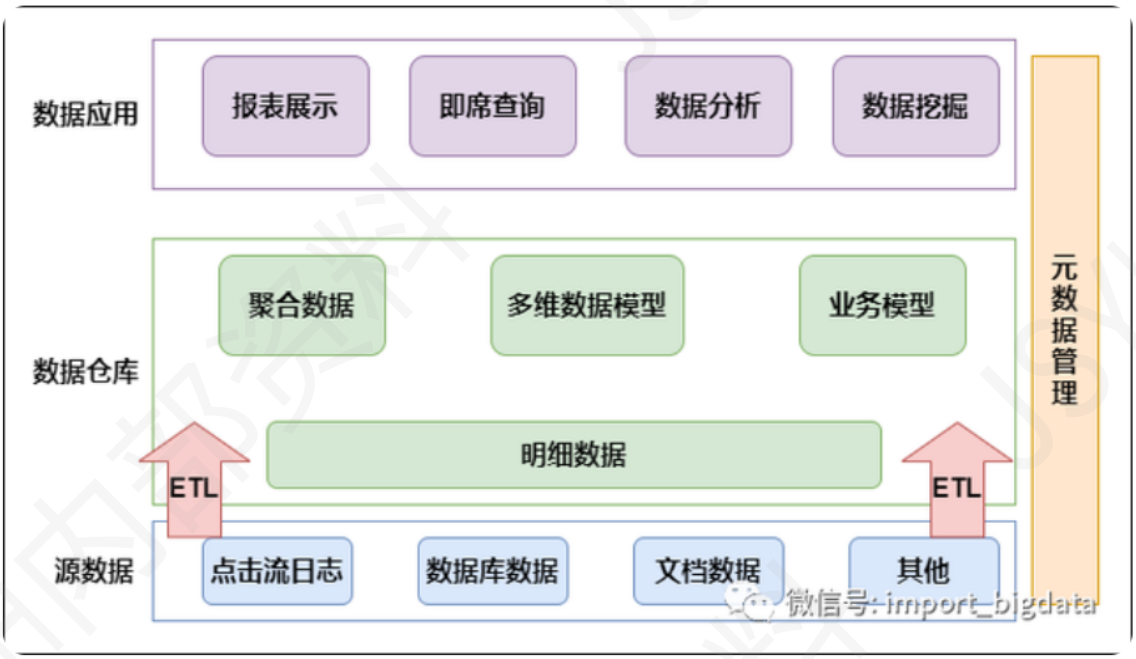
数据仓库

什么是数据仓库

- 数据仓库（Data Warehouse，简写DW）是一个数据库集合（中央存储库），存储大量的数据，做数据管理的
- 作用：主要为企业撰写分析报告与决策做支撑，对多样的业务数据进行筛选整合，为企业提供一定的BI（商业智能）能力，指导业务流程改进、监视时间、成本、质量以及控制



按照数据流入流出的过程，数据仓库架构可分为：**源数据、数据仓库、数据应用**

- 

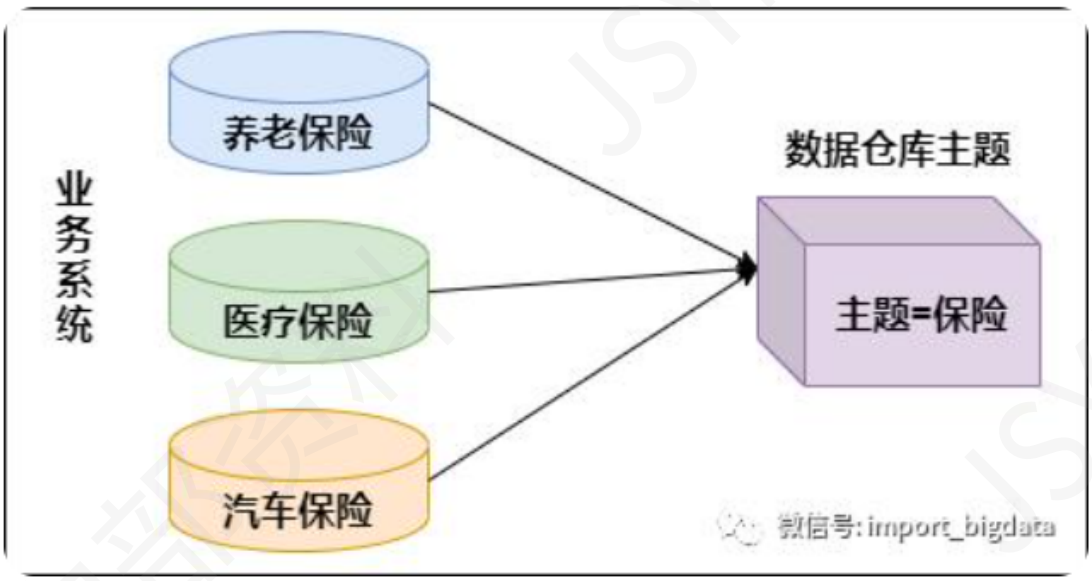
数据库分类(数据处理场景分)

1. oltp (Online transaction processing):在线/联机事务处理
   1.1 面向应用，用于业务支撑，支持增删改(insert,update,delete)，做业务支撑
   1.2 代表数据库：MySql(开源),Postgres(开源), Oracle, SQL Server, TiDB(开源), AWS Aurora, PolarDB,
2. olap (Online analytical processing):联机分析处理
   2.1 面向数据分析，侧重于决策，支持查询(select)
   2.2 开源：Hive,Impala, Presto, Spark SQL, Doris(Palo), ClickHouse, Kylin, Hawk, Druid
   2.3 数据仓库：Greenplum（商用版本，开源版本）,Teradata（商用）, Vertica（商用）

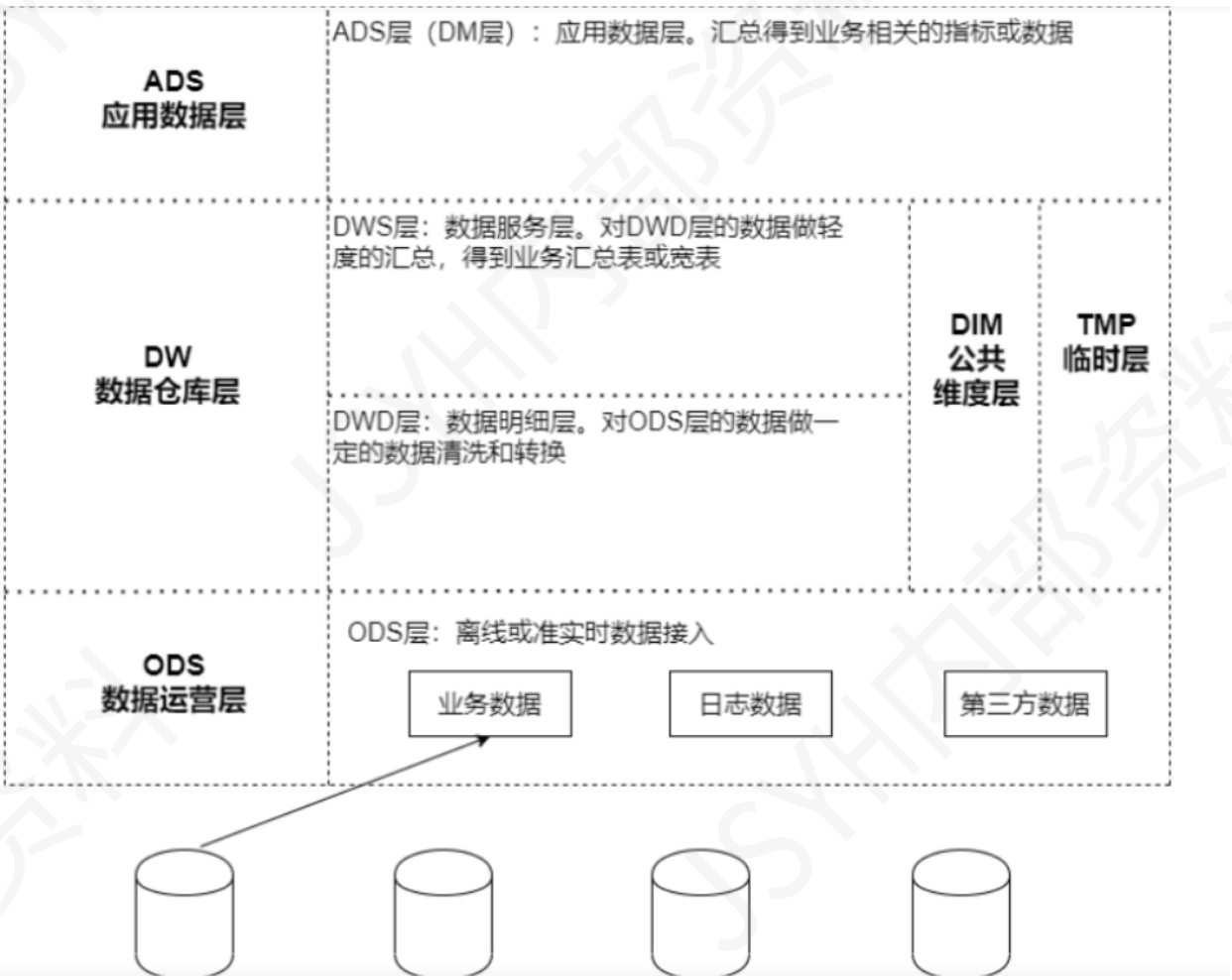
	OLTP	OLAP
操作对象	数据库	数据仓库
数据量	数据量较小	数据量大
数据模型	实体-关系（ER）	星型或雪花型
数据时效	当前数据	当前及历史数据
数据操作	支持DML、DDL	一般不支持更新和删除
操作粒度	记录级	涉及多表
性能要求	高吞吐，低延时	性能要求相对较低
操作目的	查询或改变现状	分析规律，预测趋势
业务类型	账户查询，转账等	统计报告，多维度分析

数据仓库的特点

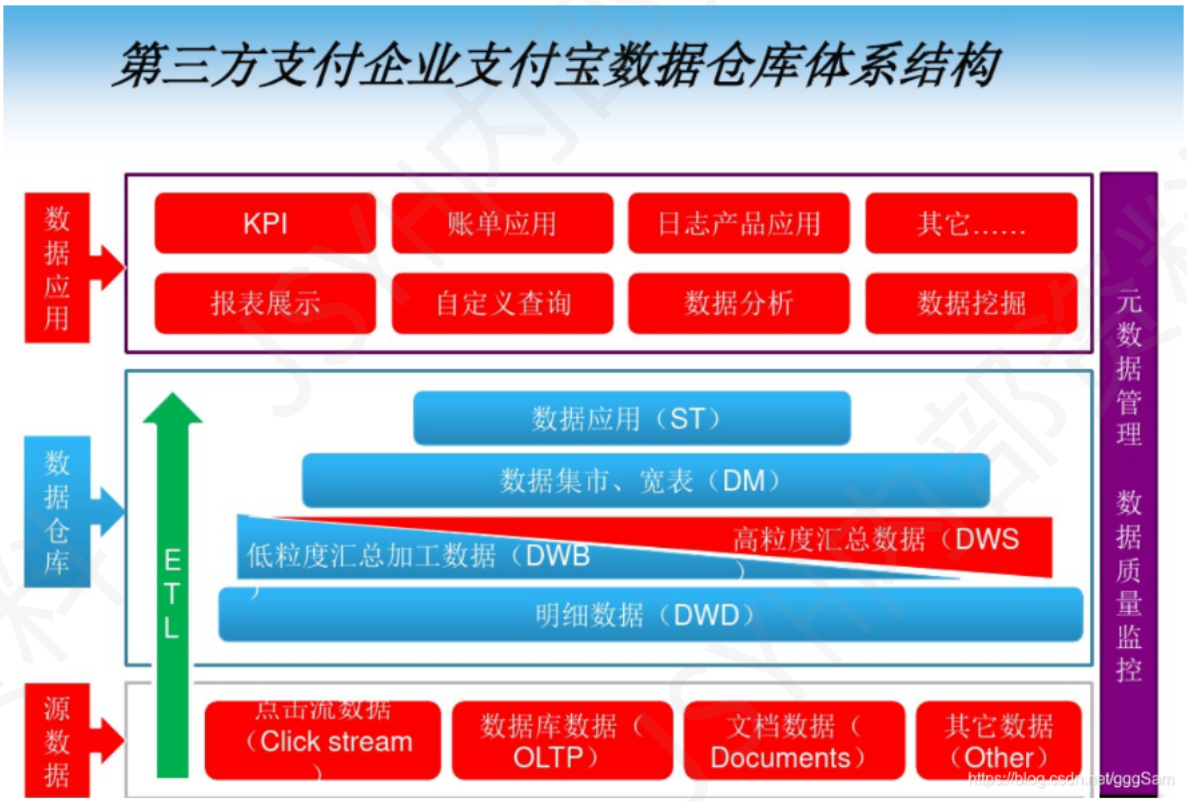
1. 集成性：
    - 数据仓库中存储的数据是**来源于多个数据源**的集成，原始数据来自不同的数据源，**存储方式各不相同。**
2. 时变性：
    - 数据仓库会**定期接收新的集成数据**，反应出最新的数据变化
3. 稳定性：
    - 数据仓库中保存的数据是**一系列历史快照，不允许被修改。**用户只能通过分析工具进行查询和分析
4. 面向主题性：
    - 不同于传统数据库对应于某一个或多个项目，数据仓库根据使用者实际需求，将不同数据源的数据在一个较高的抽象层次上做整合，所有数据都围绕某一主题来组织

- 

数据仓库的构架



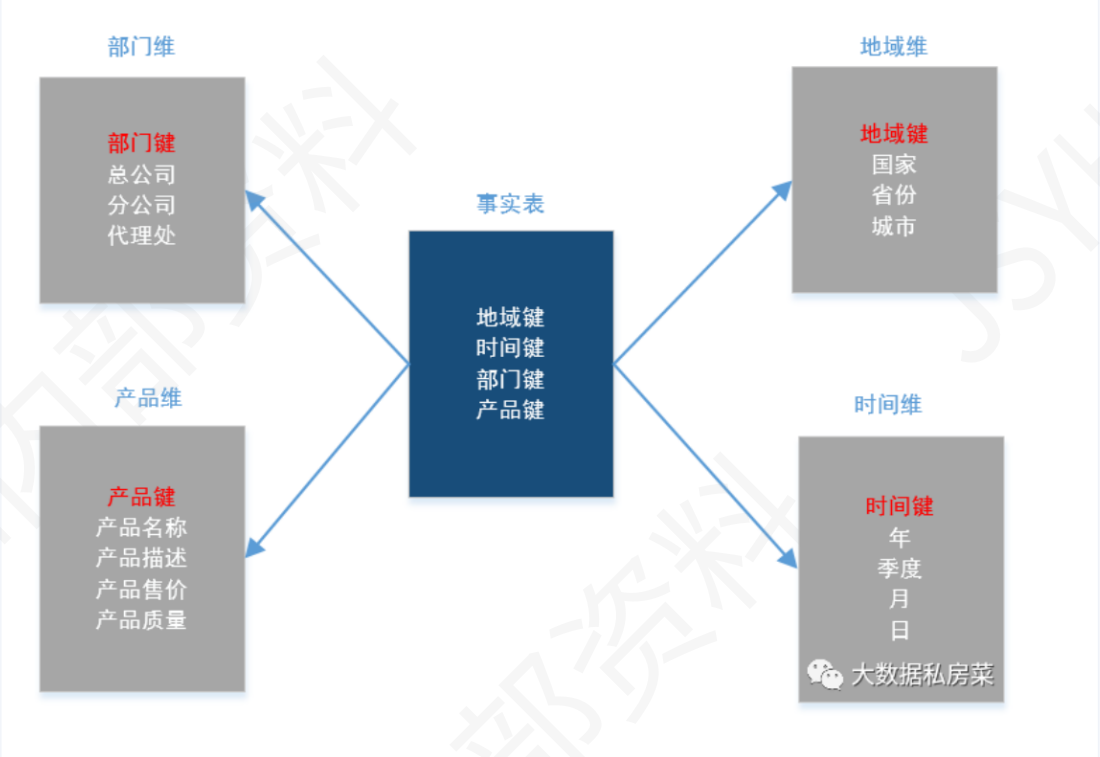
- ODS层（Operation Data Store）数据准备区：数据仓库源头系统的数据表通常会原封不动的存储一份，叫做ODS层，也叫数据操作层/贴源层/接源层。ODS层数据的主要来源是业务数据库、埋点日志、其他数据源

        - 业务数据库：可使用DataX、Sqoop等工具来抽取，每天定时抽取一次；在实时应用中，可用Canal监听MySQL的 Binlog，实时接入变更的数据。
        - 埋点日志：线上系统会打入各种日志，这些日志一般以文件的形式保存，可以用 Flume 定时抽取。
        - 其他数据源：从第三方购买的数据、或是网络爬虫抓取的数据
    - DW层（Data Warehource）数据仓库层：该层包含DWD、DWS、DIM层，由ODS层数据加工而成，主要是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。
        - DWD（Data Warehouse Detail 细节数据层），是业务层与数据仓库的隔离层。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，也即宽表化处理。
        - DWS（Data Warehouse Service 服务数据层），基于DWD的基础数据，整合汇总成分析某一个主题域的服务数据。以分析的主题为建模驱动，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。
        - DIM（公共维度层），基于维度建模理念思想，建立一致性维度
        - TMP层：临时层，存放计算过程中临时产生的数据。
    - DM（Data Marker）/ADS（Application Data Store），该层是基于DW层的数据，整合汇总成主题域的服务数据，用于提供后续的业务查询等
- 数据仓库层次的划分不是固定不变的，可以根据实际需求进行适当裁剪或者是添加。如果业务相对简单和独立，可以将DWD、DWS进行合并。 下面，以第三方支付企业支付宝数据仓库体系结构为例进行展示，如下图所示
    - 

数据仓库建模

- 目的：数仓的建模或者分层，其实都是为了更好的去组织、管理、维护数据,所以当你站在更高的维度去看的话，所有的划分都是为了更好的管理

- 引入两个定义：
  1. 事实表(fact)：事实表中的 每行数据代表一个业务事件（下单、支付、退款、评价等）。“事实”这 个术语表示的是业务事件的 度量值 （可统计次数、个数、件数、金额等），例如，订单事件中的下单金额

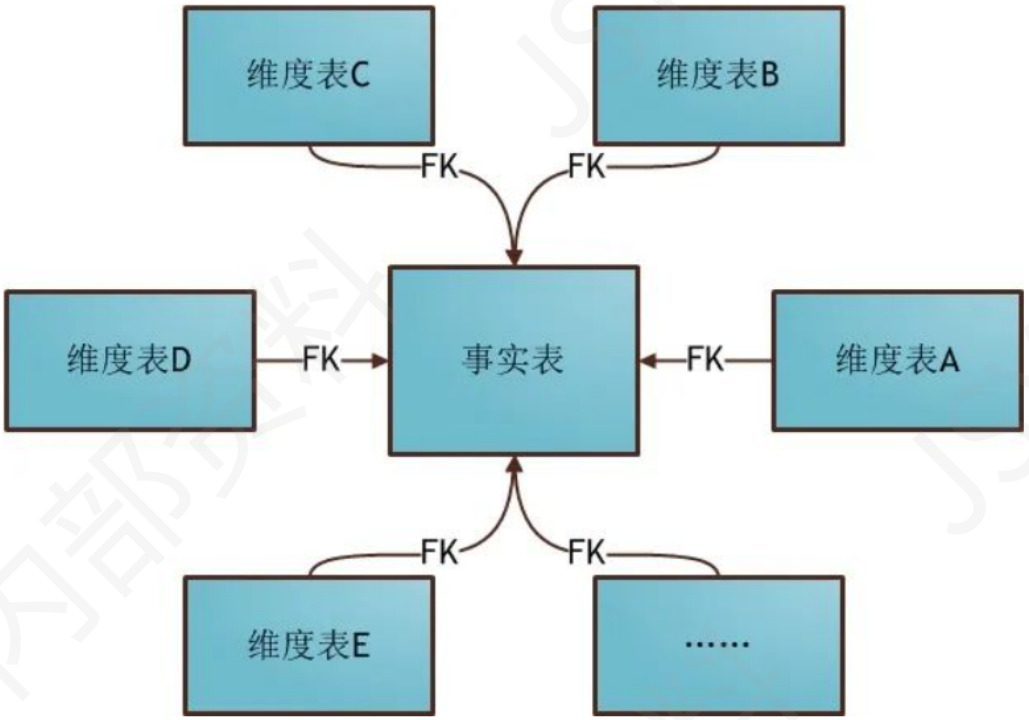


  2. 维度表(dimention)：一般是对事实的 描述信息。每一张维表对应现实世界中的一个对象或者概念
     - 如下案例：时间维度

日期 ID	day of week	day of year	季度	节假日
2020-01-01	2	1	1	元旦
2020-01-02	3	2	1	无
2020-01-03	4	3	1	无
2020-01-04	5	4	1	无
2020-01-05	6	5	1	无

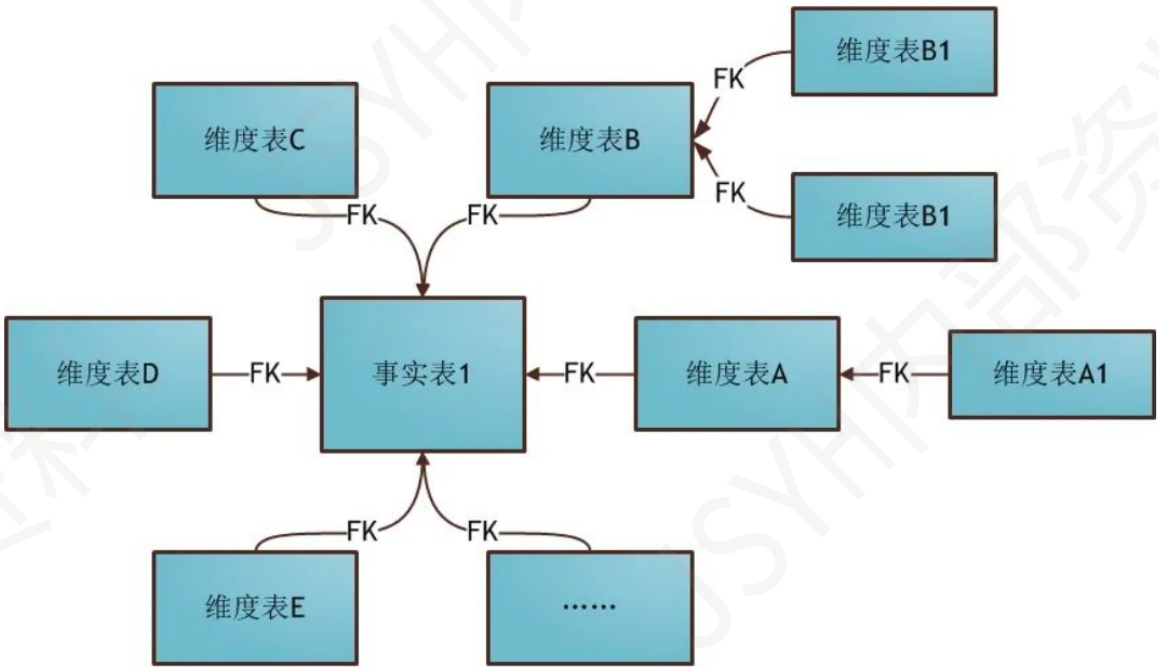
星型模型

- 核心是**一个事实表及多个非正规化描述的维度表**组成，维度表之间是没有关联的，维度表是直接关联到事实表上的，只有当维度表极大，存储空间是个问题时，才考虑雪花型维度，简而言之，最好就用星型维度即可
- **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，不存在渐变维度，所以数据有一定的冗余**，如在地域维度表中，存在国家 A 省 B 的城市 C 以及国家 A 省 B 的城市 D 两条记录，那么国家 A 和省 B 的信息分别存储了两次，即存在冗余

- 

雪花模型

- 星形模式中的维表相对雪花模式来说要大，而且不满足规范化设计。雪花模型相当于将星形模式的大维表拆分成小维表，满足了规范化设计。然而这种模式在实际应用中很少见，因为这样做会导致开发难度增大，而数据冗余问题在数据仓库里并不严重

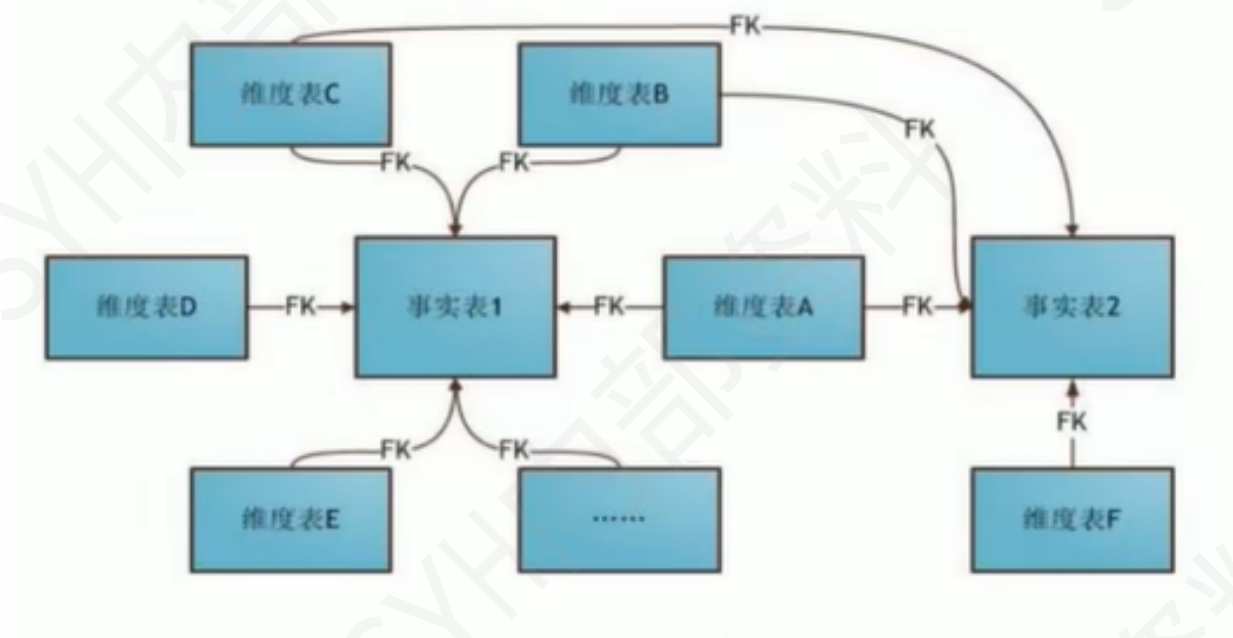
- 

星型模型 VS 雪花模型

- **主要区别在于对维度表的拆分，对于雪花模型，维度表的设计更加规范**；而**星型模型，一般采用降维的操作，利用冗余来避免模型过于复杂，提高易用性和分析效率**
- 雪花模型冗余少
- 星型模型性能好

星座模型

- 星座模式是星型模式延伸而来，星型模式是基于一张事实表的，而星座模式是基于多张事实表的，而且共享维度信息。
  前面介绍的两种维度建模方法都是多维表对应单事实表,但在很多时候维度空间内的事实表不止一个，而一个维表也可能被多个事实表用到。在业务发展后期，绝大部分维度建模都采的是星座模式。
-



模型命名规范

- 维度表的命名全部遵循了**dim _<dimension-name>**的规则，其中**<dimension-name>**用来描述维度的内容。
- 维度一般都是以**人员(who)、时间(when)、地点(where)、事件(what)**来划分 。
- 事实表的命名全部遵循了**fact_<fact-name>**的规则，其中**< fact -name>**用来描述事实的内容。
- 事实表一般都是以**多少(how much)**来划分。

处理维度表和键的管理

- 维度表分类：静态维度表和动态维度表。
  静态维度是不会变化的维度数据，如时间、日期等；
  动态维度是业务系统产生的，一段时间会产生变化。

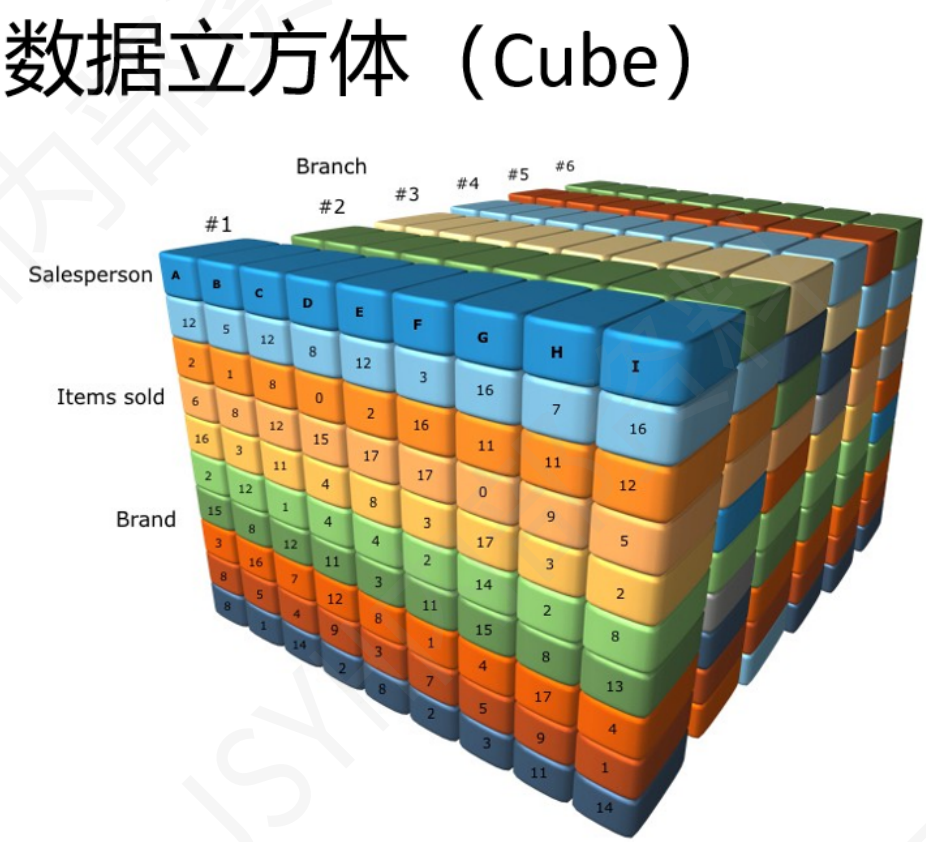
- 缓慢变化维（SCD）类型：
  类型一：对源系统的更新，也会直接更新目标的维度表。维度表总是保存当前最新的状态，如果发生变化就直接覆盖，通过 Kettle 的“ 插入/更新 ”步骤来实现。
  类型二：对源系统的更新，会往目标维度表里插入一行数据，通过不同的时间戳来维护同一条维度数据的多个版本，在任何一个给定的时间点，都可以找到一行对应的维度数据。可以按照时间追踪到维度的变化。通过 Kettle 的“ 维度查询 /更新 ”步骤来实现。
  类型三：对源系统的更新，会在目标维度里增加列，在目标维度表同一行新增的列里保存新的数据。Kettle 没有一个专用的步骤来支持，但是可以写一个作业并使用“ 表里面的列是否存在 ”步骤来判断是否要更改表结构，然后使用 SOL 脚本步骤，执行相应的DDL语句增加一个新列。

- 业务键：源系统中业务主键，用来标识唯一的一个业务实体。维度表代理键用来标识维表表里面的一行。

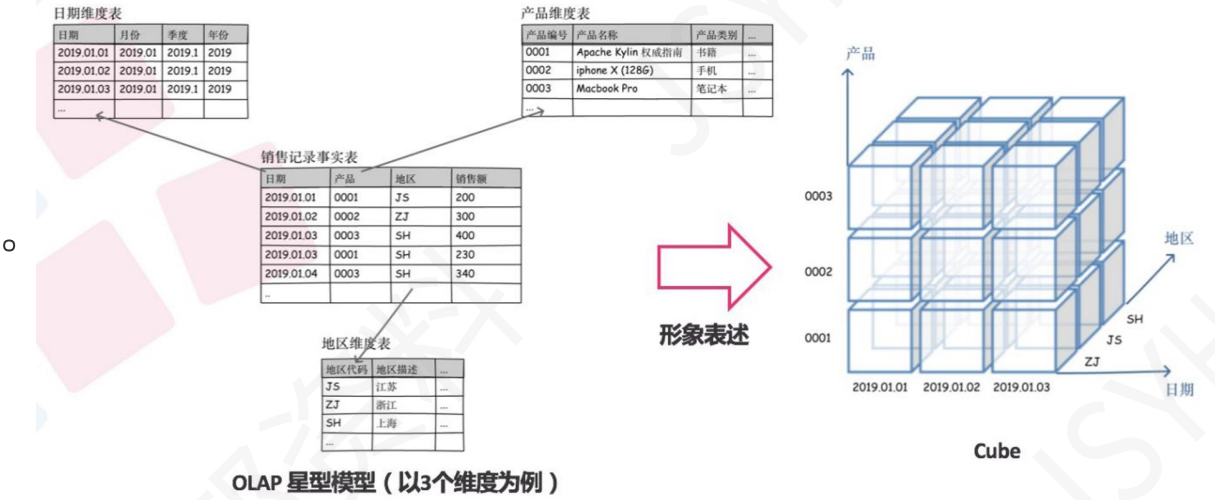
- 维度表代理键：用来标识维度表里面的一行，数据仓库最佳实践表明，原则上,维度表应该使用自动生成的无意义整型数值作为代理键。代理键的值和维度表里面的属性没有关系，代理键一般是在 ETL过程中生成。

数据立方

-



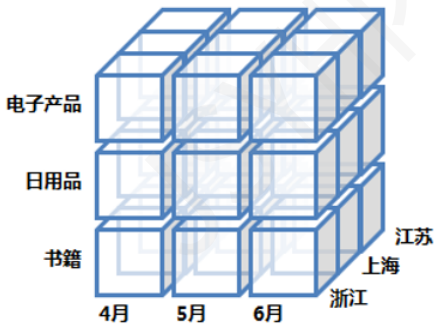
- 多维数据分析是指按照多个维度（即多个角度）对数据进行观察和分析，多维的分析操作是指通过对多维形式组织起来的数据进行切片、切块、聚合、钻取 、旋转等分析操作，以求剖析数据，使用户能够从多种维度、多个侧面 、多种数据综合度查看数据，从而深入地了解包含在数据中的信息和规律。
- 多维分析主要面向业务用户提供数据查询分析服务，由于业务人员不懂 SQL，也无法完成多表关联（有意义的查询经常是基于多表的），所以在多维分析建模阶段需要将多表转换成单表，也就是 Cube



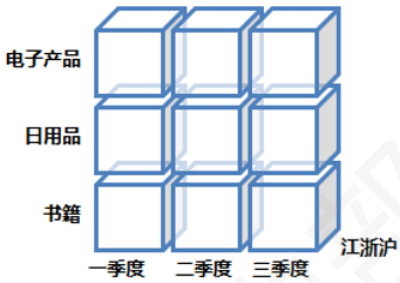
OLAP 星型模型（以3个维度为例）

OLAP的基本操作

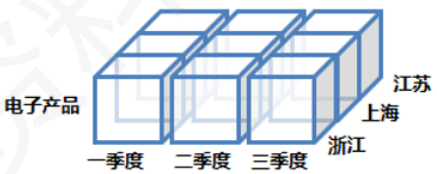
- **钻取（Drill-down）**：在维的不同层次间的变化，从上层降到下一层，或者说是将汇总数据拆分到更细节的数据，比如通过对2010年第二季度的总销售数据进行钻取来查看2010年第二季度4、5、6每个月的消费数据，如上图；当然也可以钻取浙江省来查看杭州市、宁波市、温州市.....这些城市的销售数据。
- **上卷（Roll-up）**：钻取的逆操作，即从细粒度数据向高层的聚合，如将江苏省、上海市和浙江省的销售数据进行汇总来查看江浙沪地区的销售数据，如上图
- **切片（Slice）**：选择维中特定的值进行分析，比如只选择电子产品的销售数据，或者2010年第二季度的数据。
- **切块（Dice）**：选择维中特定区间的数据或者某批特定值进行分析，比如选择2010年第一季度到2010年第二季度的销售数据，或者是电子产品和日用品的销售数据。
- **旋转（Pivot）**：即维的位置的互换，就像是二维表的行列转换，如图中通过旋转实现产品维和地域维的互换。



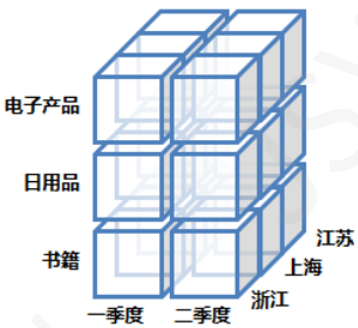
钻取(Drill-down)



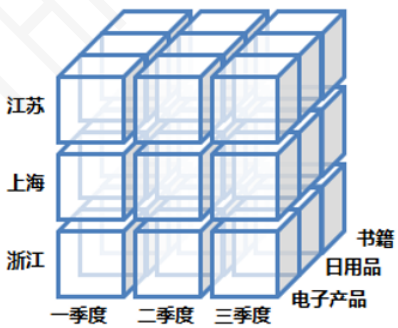
上卷(Roll-up)



切片(Slice)



切块(Dice)



旋转(Pivot)