

数据仓库架构落地

尹佳俊



架构能力和演进

能力演化的不同阶段



作业工具统一/线上化, 提供可初步分析的数据。

根据线下业务流程, 进行作业系统的建设, 将业务动作线上化, 产生业务动作数据



通常是战略方向性决策、大体计划制订,辅助执行层决策。

业务专家主导, 把人工经验转换成决策算法, 依靠丰富的内部信息 + 少量外部信息, 做自动化决策; 同时建立评估体系, 评估降本增效的能力



通常是战略方向性精细到执行人员的工作计划,能够全局地平衡成本与时效。

算法专家主导,建立高度实时的数据信息平台, 汇总系统内部数据+大量系统外数据, 使用AI算法+策略混合的方式, 形成计划层、执行层、反馈层的闭环

落地关键



数据仓库构建

简单报表

数据集市

数据仓库

数据开发:

● 10人

● 30+人

● 100+人

模式:

● 支撑所有业务

● 与业务共建

● 各业务自建、自治

架构演进

小数据场景

大部分问题使用OLTP

(Online Transactional Processing)

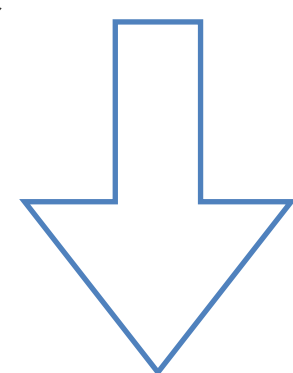
技术就能解决

核心数据全集在亿级以下，冷热数据能有效区分界限

数据潜在价值容易被忽略、不易挖掘
技术普适，适合从0开始的大数据项目，较易验证结果

小数据场景驱动决策

- 运营监控
- 产品改进
- 智能决策

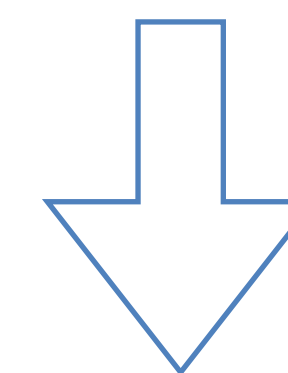


SQL

RDBMS :
MySQL

大数据平台:
Hadoop

更多数据场景



SQL

复用基础技术，降低开发门槛



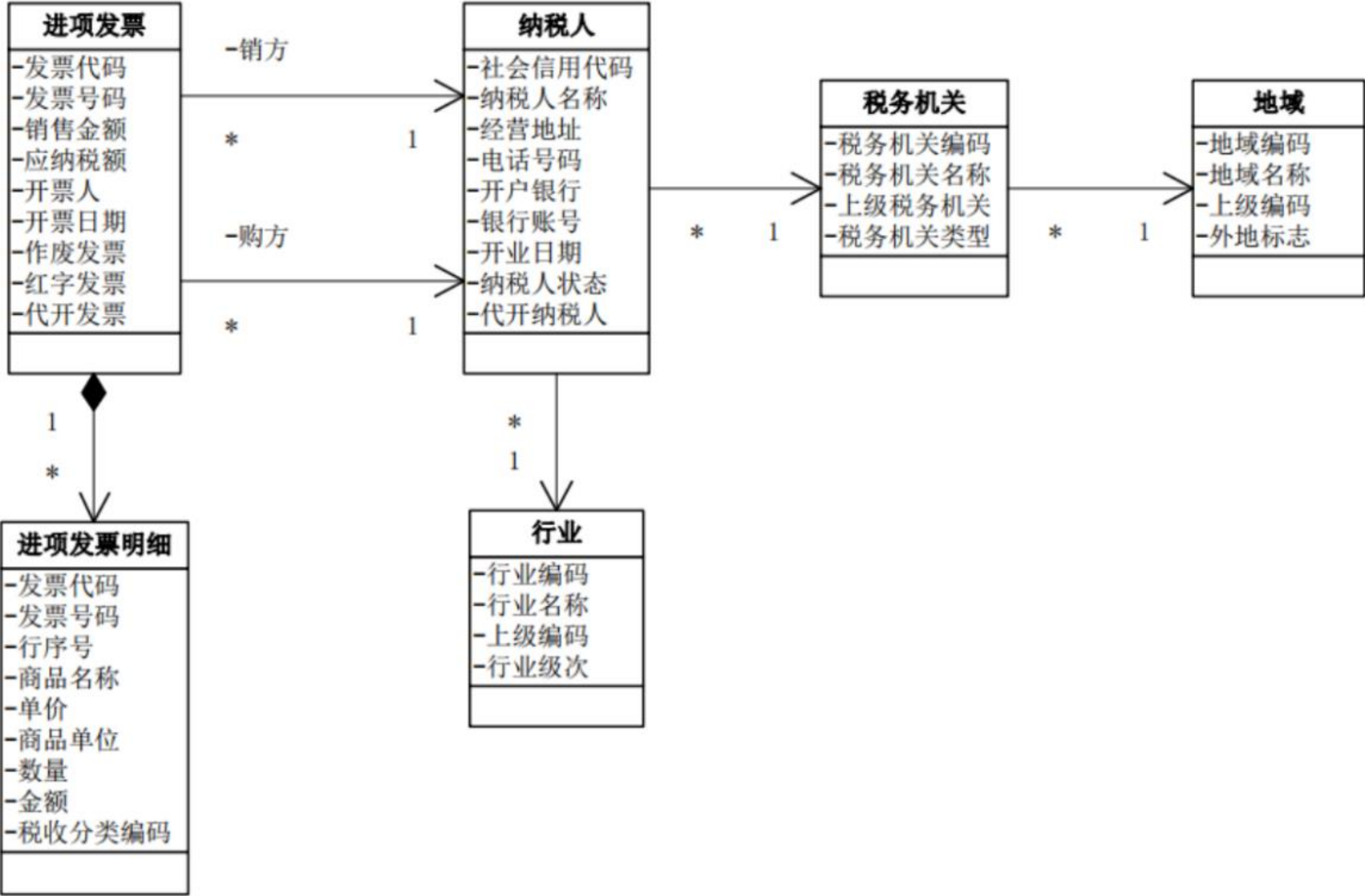
数据处理过程

数据清洗、转换、集成

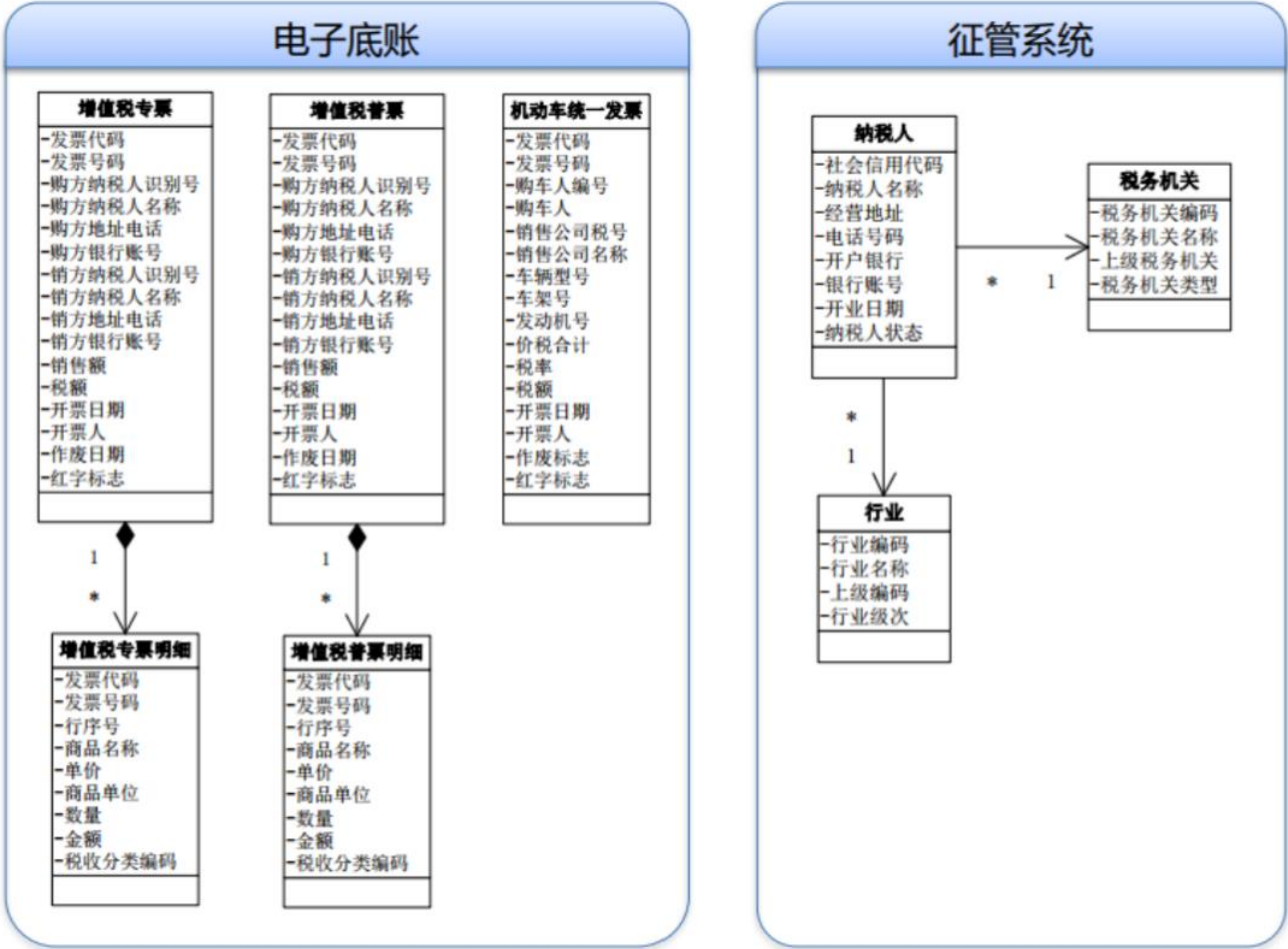
数据处理过程



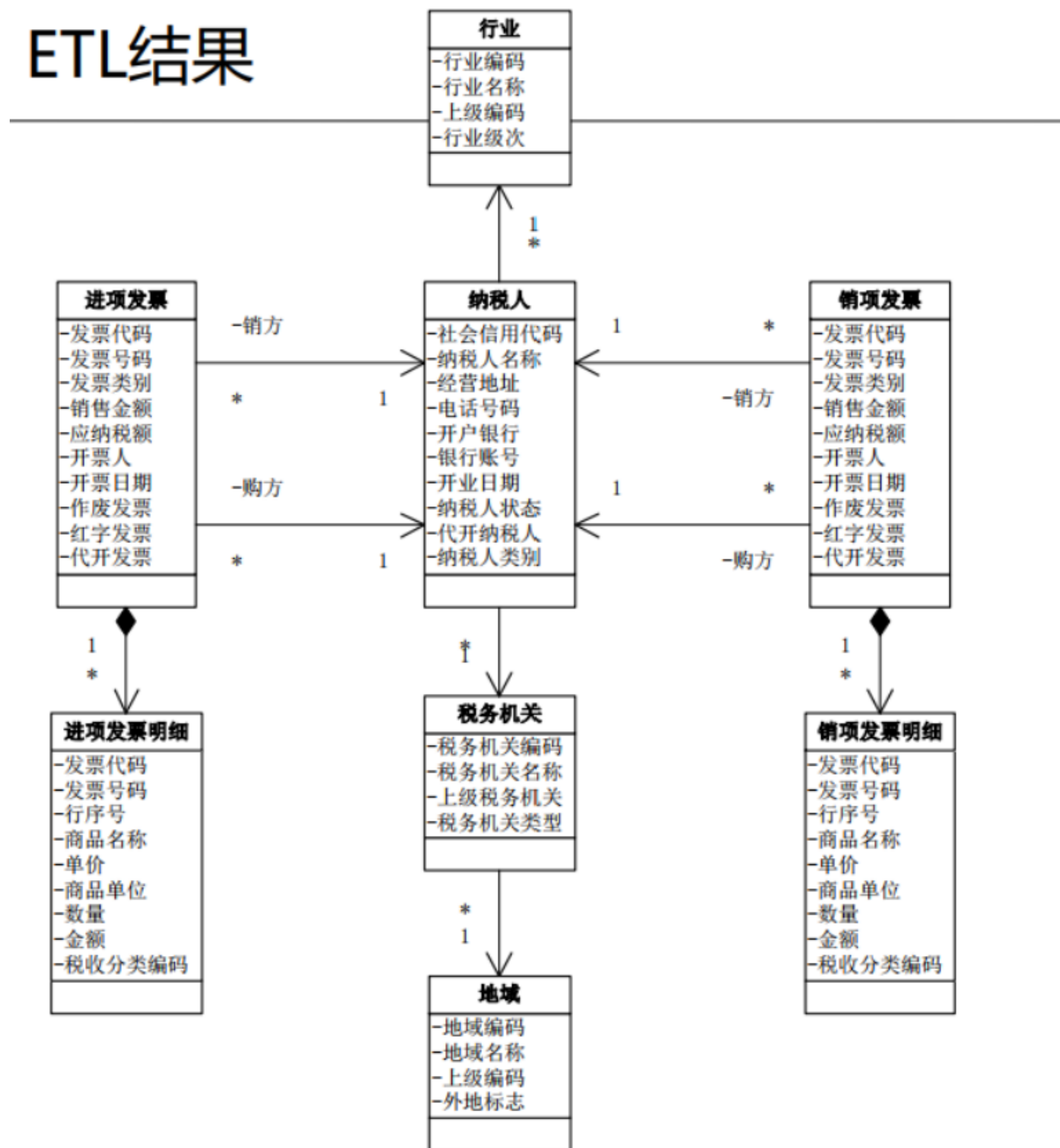
更好地整理数据：数据清洗



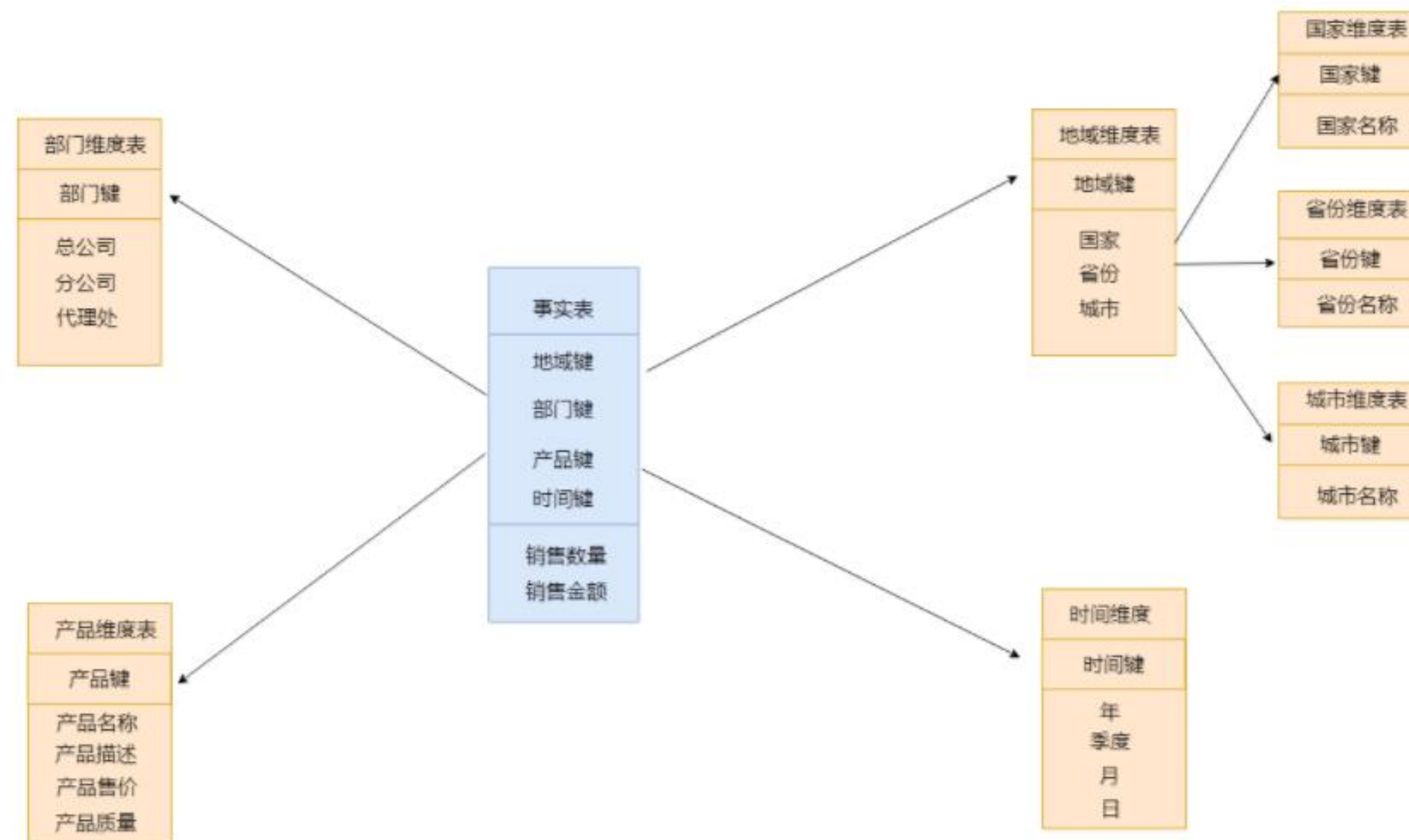
更好地整理数据：转换、集成



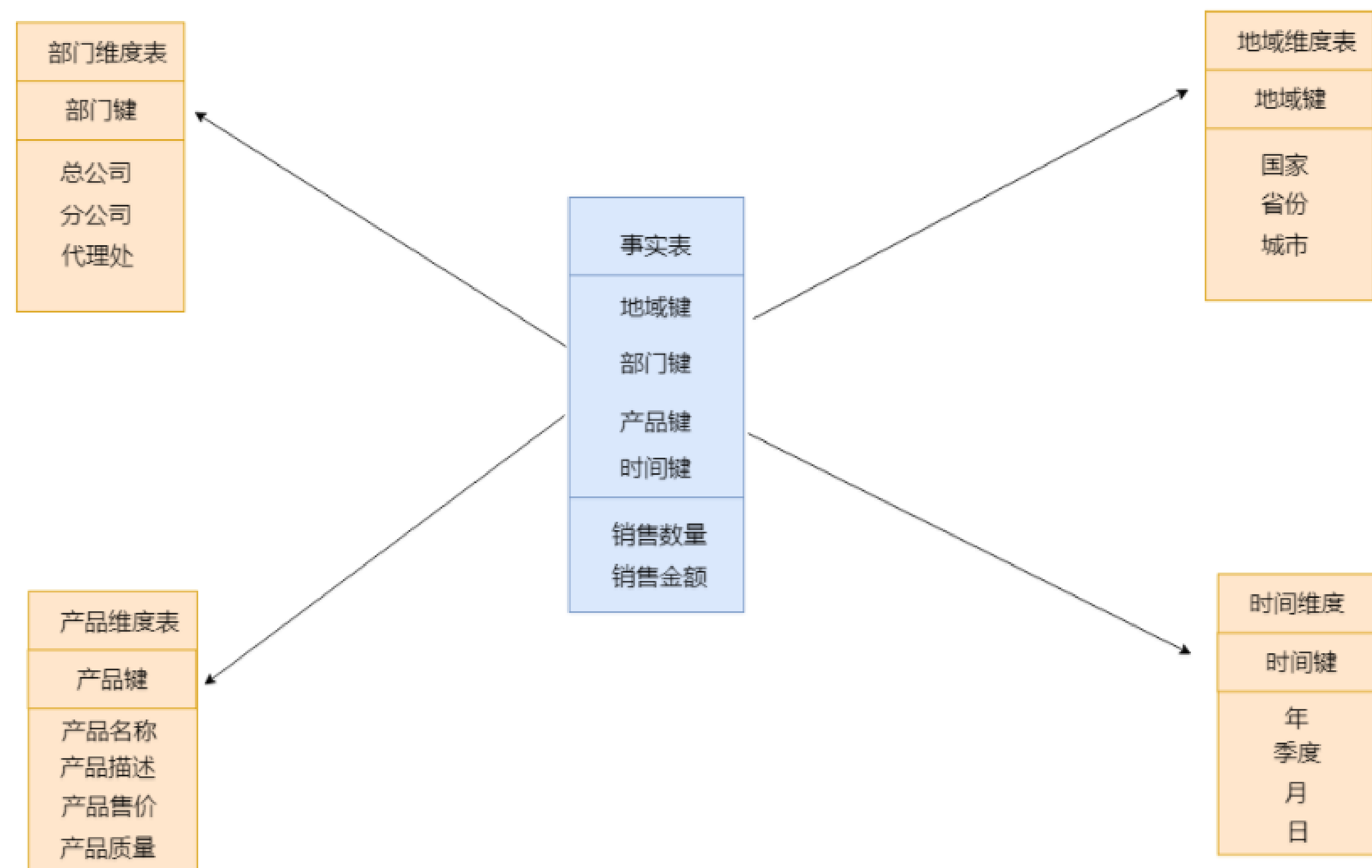
更好地整理数据：ETL结果



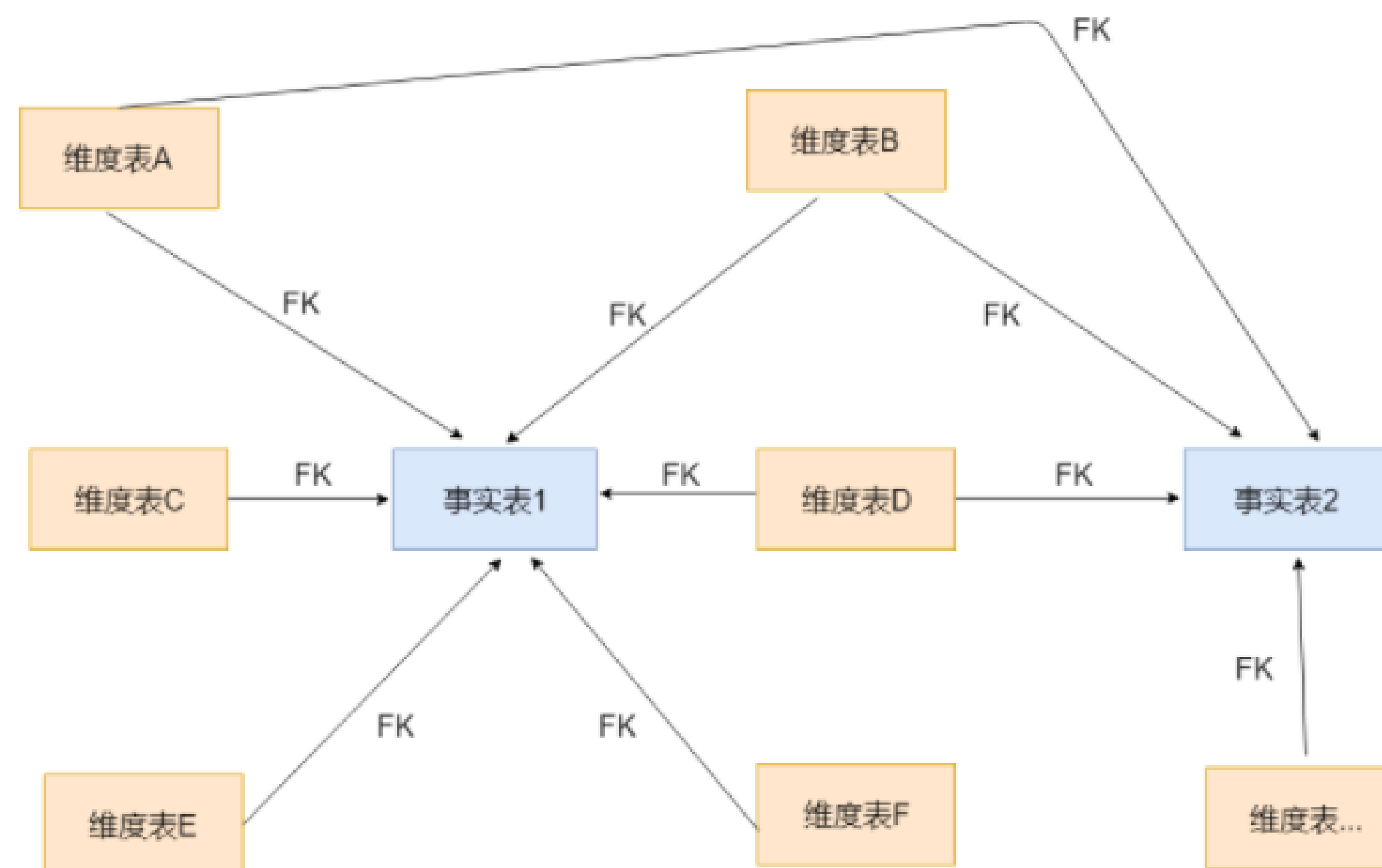
更好地整理数据：数据仓库（雪花模型）



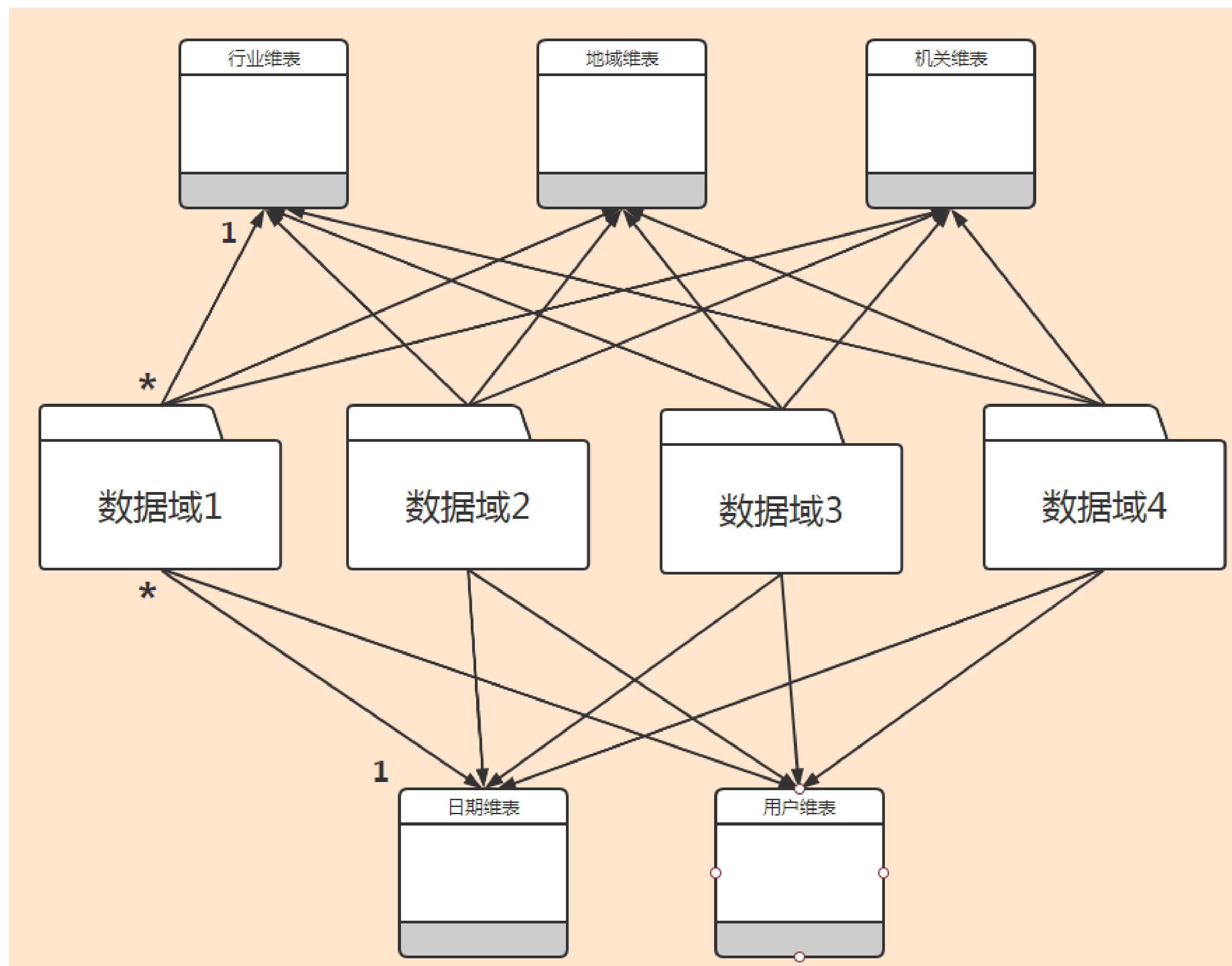
更好地整理数据：数据仓库（星形模型）



更好地整理数据：数据聚合



更好地整理数据：数据域





小数据场景解决方案

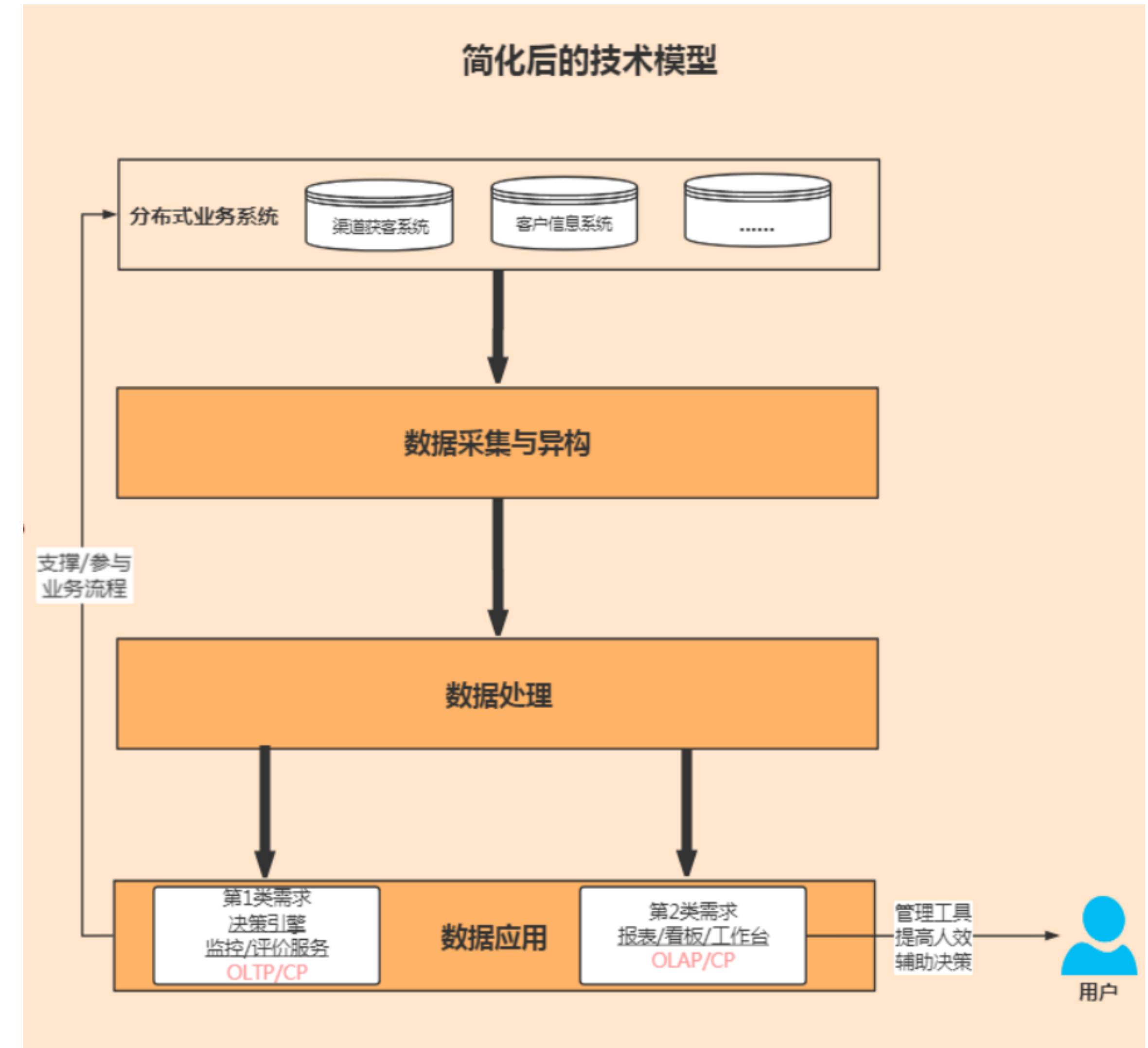
技术工具选型/架构思考

用户侧(业务人员)的基本要求

- 不能丢失数据,对脏数据有轻微容忍 (最终一致性)
- 数据要尽可能实时(准实时),否则对业务时效有影响
- 系统要高可用, 不能中断(高可用)
- 数据准确性要求高, 否则会影响业务交付(高准确)

系统侧的基本要求

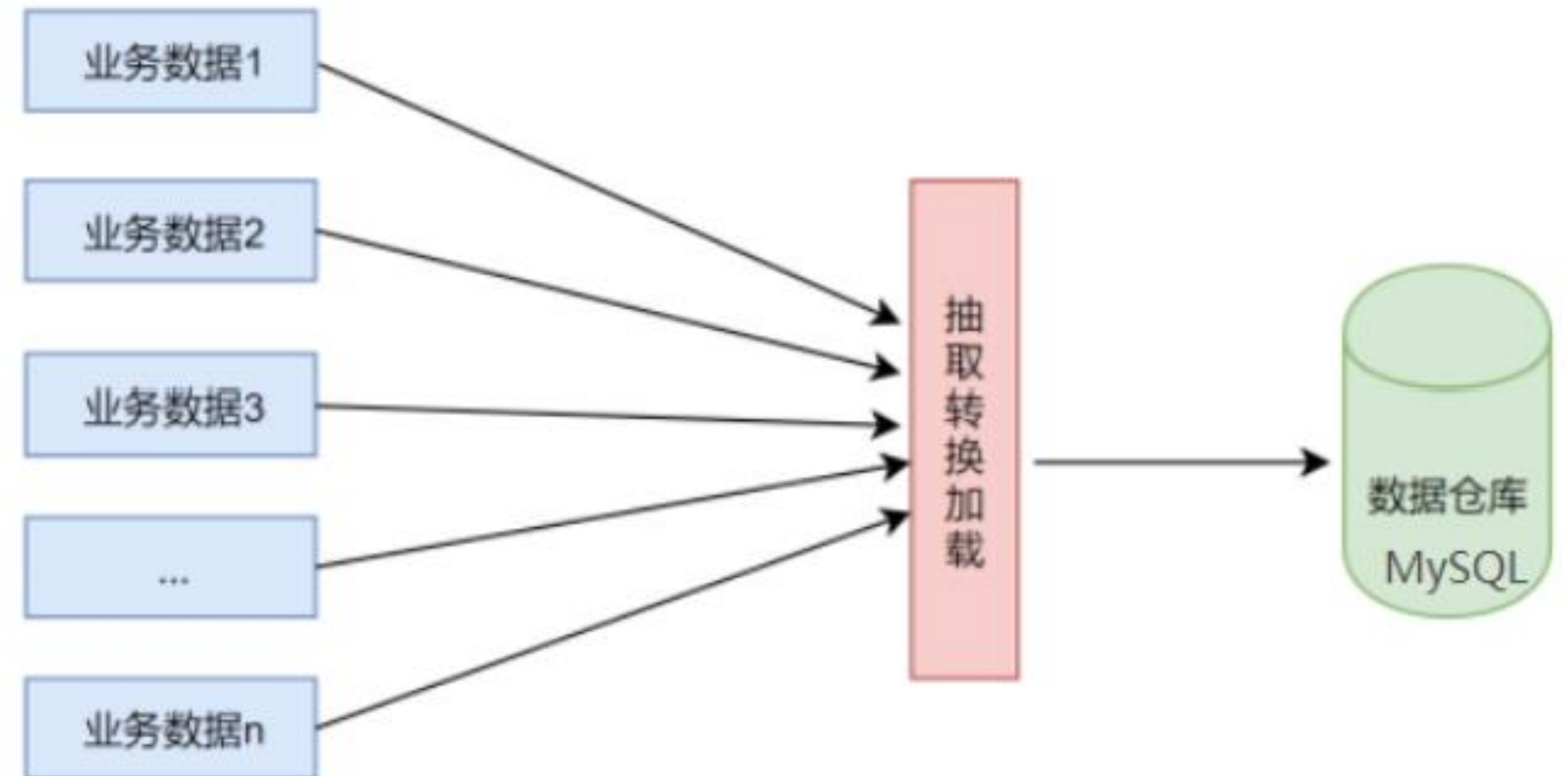
- 用于业务流程的API/数据必须保证强一致性
- 轻微容忍服务中断, 有兜底和容错能力
- 高性能、实时响应
- 不可通过扫表/调API的方式采集数据,不能影响作业系统正常工作(异步采集)



数据采集

Linux shell脚本+定时任务

前期数据较小，可通过定时任务简单快速完成数据采集，结果存储到MySQL



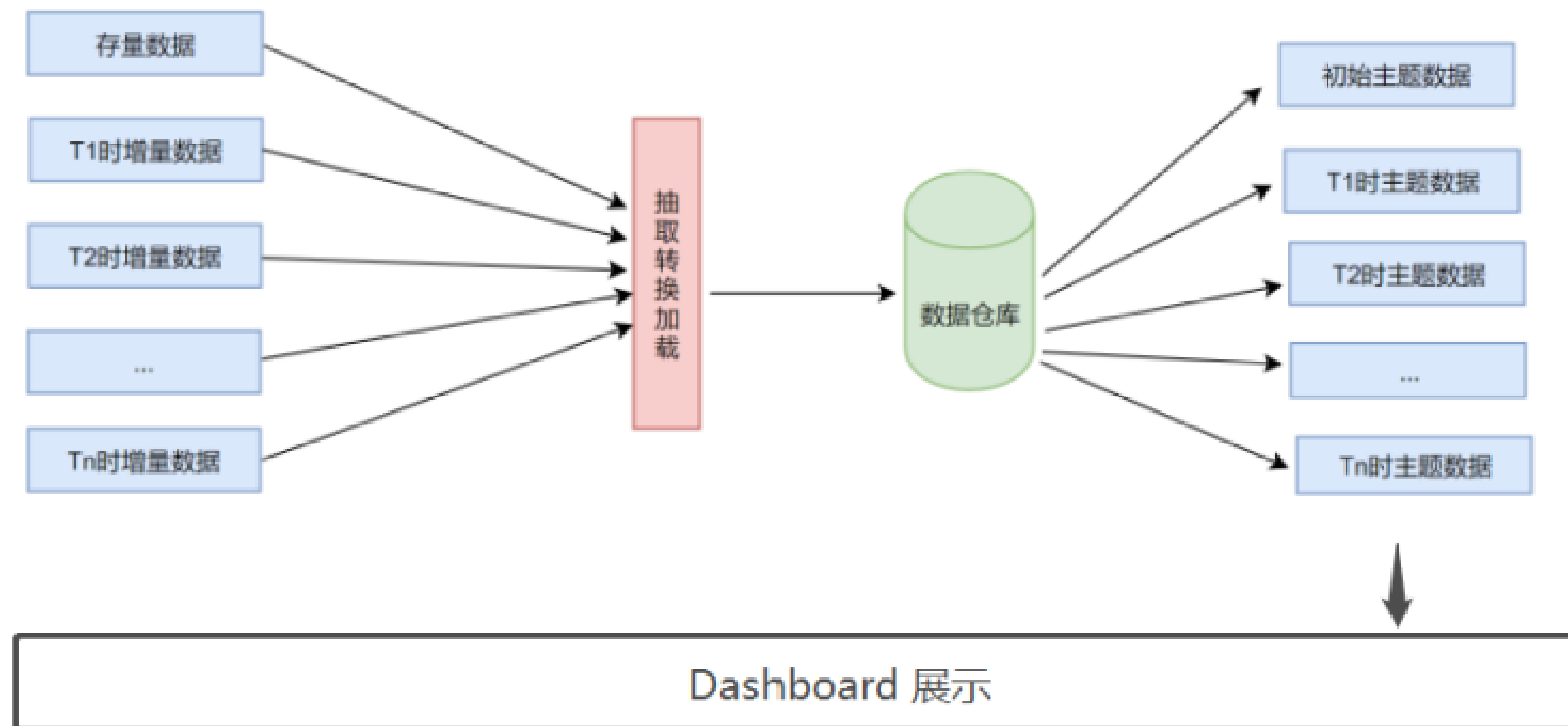
数据稳定性和时变性

时变性+稳定性

稳定性： 数据仓库中的数据只进行新增，不进行更新操作、删除操作处理。

时变性： 数据仓库的数据一般都带有时间属性，随着时间的推移而发生变化，不断地生成主题的新快照

Dashboard： 前后端分离的单体应用，如 Springboot2.*+vue



小数据场景总结

小数据量业务场景的如何用数据驱动？

- 数据与业务一定要形成闭环，让业务产生数据，
- 数据叠加策略、算法去反驱业务
- 本质是业务闭环驱动，以业务价值目标为导向
- (降本增效)
- 洞悉业务痛点与细节的业务专家至关重要
- 演进式迭代：线上化、策略驱动、数据驱动
- 可挖掘的数据价值 > 数据体量
- 准确的时效/成本的评价能力至关重要
- 专注小数据中的潜在价值,配合专家认知和策略,尽可能的去挖掘



小数据场景总结

小数据量场景的数据平台技术架构与组件选型？

- 合理抽象业务数据, 上游易加工、下游易用, 语义清晰
- 主要使用OLTP的技术进行处理
- 尽可能贴近业务去进行架构设计,让数据驱动力直接作用于业务上(如智能决策引擎)
- 发挥小数据量带来的实时性优势(技术驱动业务)
- 数据要进行合理的分层存储, 提高复用性
- 合理进行数据异构,尽量复用公司公共资源(如离线从库、数据仓库等)





大数据场景解决方案

演化式重构

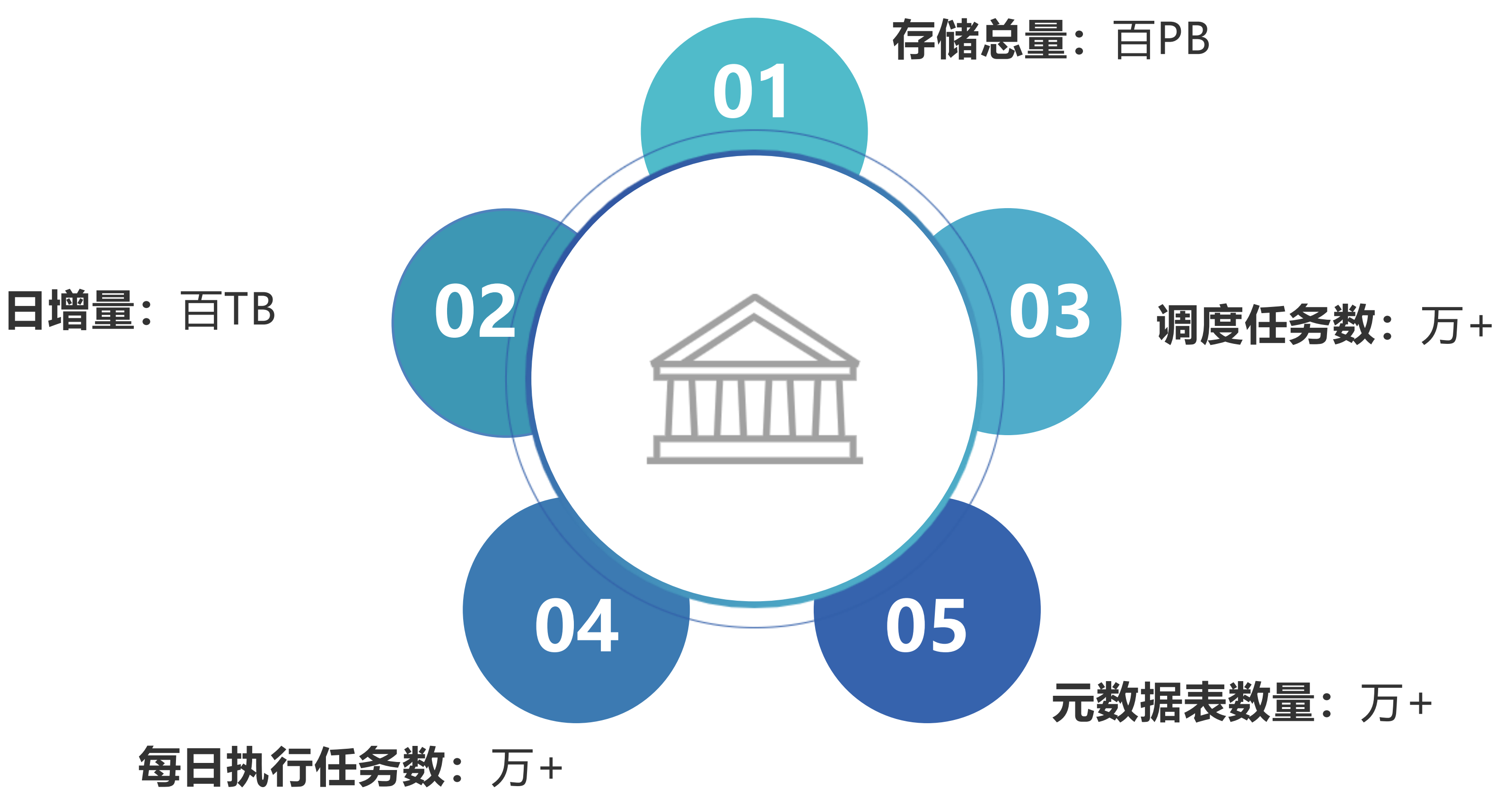
软件重做

抛弃原系统，重新搭建业务系统
很难准确识别原系统繁复的业务逻辑
不能有效替代原系统的功能

- 是在原系统的基础上一步一步地改造
- 每做一步就可以运行并验证正确性
- 每做一步都能保持原功能的一致
- 将系统改造的长周期变为短周期
- 随时都可以获得可运行的软件并发布

演化式 重构

演化式重构



全域数据仓库构建实践

面临的问题

- DB表全量同步，效率低下
- 数据孤岛，无法共享，相互孤立
- 重复计算，缺乏沉淀，资源浪费
- 数据建模意识薄弱
- 表元信息命名不规范
- 指标定义混乱

解决方案

- 制定规范 约束建表流程
- 数据同步 大表增量 拉链表设计
- 维度建模 主题域划分 轻度汇总 沉淀中间结果
- 数据地图 数据血缘关系
- 统一口径 指标字典管理
- 数据治理 报表生命周期管理

大数据体系

数据可视化/ 反馈	DA（数据应用层）										服务业务化
数据统计/ 分析/挖掘	BI报表				数据产品			业务系统		应用治理	
	渠道分析	商品分析	交易分析		智能挖掘	自助报表	精细化推送	商品系统	财务系统	指标字典	
	用户分析	订单分析	行为分析		画像档案	事件漏斗	数据地图	运营系统	客服系统	血缘关系	
	搜索推荐	竞品分析	商业分析		数据地图	监控告警	电视看板	搜索推荐	质检系统	滚动清理	
	DaaS（Data-as-a-Service）										数据服务化
数据建模 /存储	数据集市层										
	留存模型主题表		事件模型主题表		画像提取平台		实时自助框架		生命周期管理	数据质量管理	
	数据仓库层										
	用户宽表		商品宽表		交易宽表		收入宽表		广告宽表	行为宽表	
	源数据层										业务数据化
	前端埋点		后端日志		业务数据库		三方广告		竞品抓取	线下表单	
	PaaS（Platform-as-a-Service）										
数据传输 （实时/批量）	数据计算层										
	MapReduce		Spark		Storm		Flink		Kylin	Druid	
	数据存储层										
	HDFS		Hive		HBase		MySQL		TiDB	ZZRedis	
数据采集	数据传输层										
	Flume		Sqoop		Kafka		Lego		Nginx	log4j	

全域数据仓库

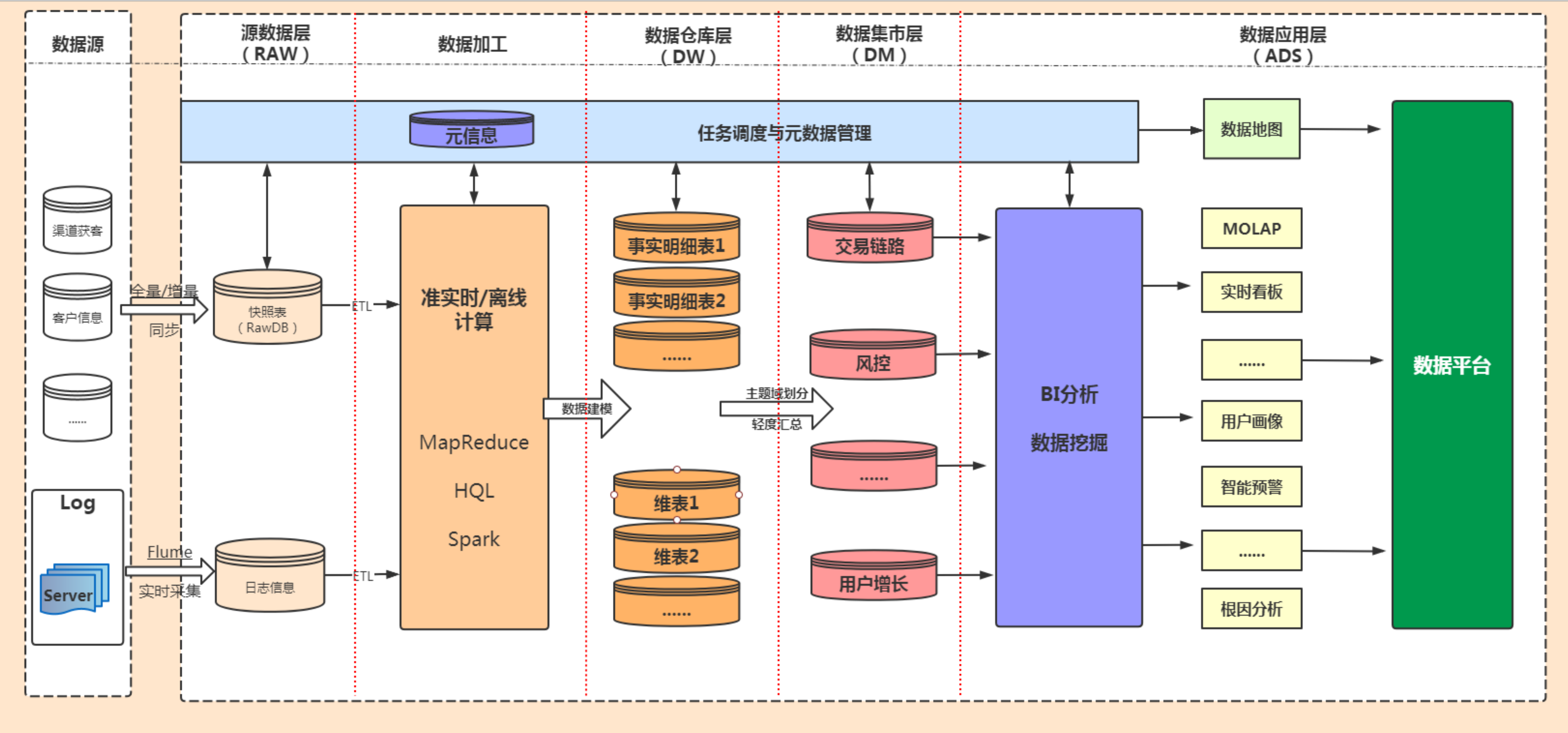


承数据启业务

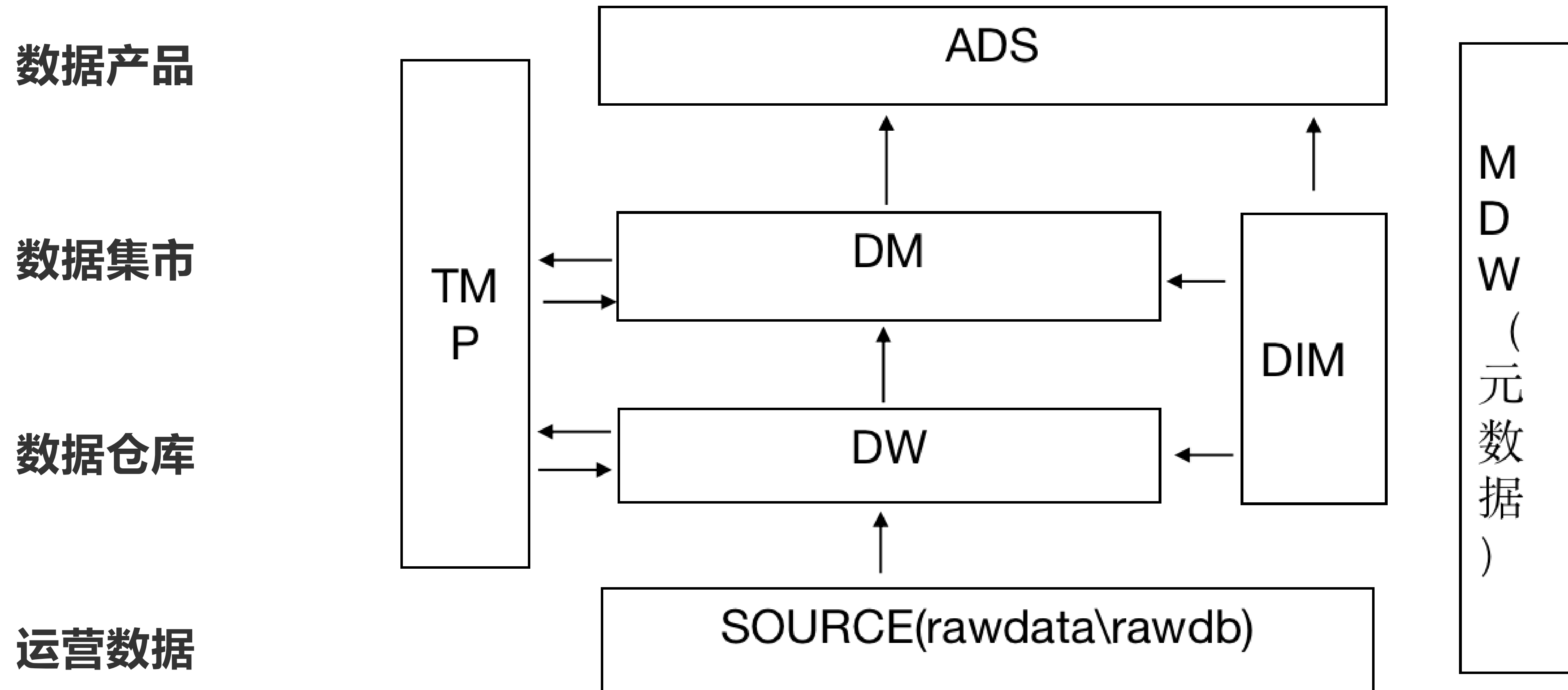
核心组件/功能、全域数据仓库、iQuery自助式可视化查询分析平台

服务业务化	数据服务层 iQueryZeppelinSCF					
资产服务化	数据集市层 留存模型主题表事件模型主题表画像提取平台实时自助框架生命周期管理数据质量管理					
数据资产化	数据仓库层 用户宽表商品宽表交易宽表收入宽表广告宽表行为宽表					
业务数据化	源数据层 前端埋点后端日志业务数据库三方广告竞品抓取线下表单					

数据处理过程：数据平台整体流程



全域数据仓库四层架构



大数据场景数据采集解决方案

Sqoop的重构与整合

主要用于在Hadoop(Hive)与传统的数据库(mysql、postgresql...)间进行数据的传递，可以将一个关系型数据库（例如：MySQL,Oracle,Postgres等）中的数据导进到Hadoop的HDFS中，也可以将HDFS的数据导进到关系型数据库中。

```
#!/bin/bash
cd /home/aisinobi/bin
connect=jdbc:oracle:thin:@
hive_dir=/user/hive/warehouse
tmp_dir=./tmp
sqoop import --hive-import --connect ${connect} --username $1 /
--password-file /user/hive/.$1 --query "$4" -m 10 --hive-database $2 /
--hive-table $3 --hive-overwrite --target-dir ${hive_dir}/$2.db/$3 /
--outdir ${tmp_dir} --bindir ${tmp_dir} -z --connect /
--null-string '\\N' --hive-delims-replacement
```

```
#!/bin/bash
sql="select * from dzdz_fpdx_hyzp t where kprq between
to_date('$1','yyyy-mm-dd') and to_date('$2 23:59:59','yyyy-mm-dd
hh24:mi:ss') and \$CONDITIONS"
map=SL=DOUBLE,SE=DOUBLE,JSHJ=DOUBLE
bash SqoopJdbc.sh DZDZ dzdz DZDZ_FPDX_HYZP "$sql" $map
```

数据仓库落地关键

- 数仓一定要层次明确、规范可落地执行，统一认知
- 面向业务建模，减少层次结构压缩流程长度

业务主体		全局						
数据主题	交易主题	√	√	√	√	√	√	√
	交易链路	√						
用户主题	活跃（DAU）	√	√	√				
	新增	√						
	注册	√						

维度		一致性维度						
数据域*业务过程		省份	城市					成交金额
		√	√	√	√	√	√	×
交易		√	√	√	√	√	√	×
		√	√	√	√	√	√	×
		√	√	√	√	√	√	√
		√	√	√	√	√	√	×
用户	激活	√	√	×	×	×	×	×
	注册	√	√	×	×	×	×	×
	活跃	√	√	×	×	×	×	×

数据仓库落地关键

- 数仓一定要层次明确、规范可落地执行，统一认知
- 面向业务建模，减少层次结构压缩流程长度

列名	类型	描述
order_id	bigint	订单id
pay_id	bigint	支付
status	int	
info_id	bigint	
cate_first_id	int	一级分类id
cate_second_id	int	二级分类id
cate_third_id	int	三级分类id
cate_first_name	string	一级分类名称
cate_second_name	string	二级分类名称
cate_third_name	string	三级分类名称
first_biz_type	int	一级业务
second_biz_type	int	二级业务
third_biz_type	int	三级业务回收
order_source	int	订单的业务来源
first_from	string	一级页面来源
sec_from	string	二级页面来源
ori_firstfrom	string	原始一级来源
ori_secfrom	string	原始二级来源

业务全量信息表

申明粒度，一行描述一个订单

普通维度：描述

退化维度：在事实表中冗余维度ID和具体维度值

普通维度：可匹配业务来源字典



数据表命名规范：*dw_mysql_order_1d*

业务规范层	数据模型层次名称	物理表命名规范	数据存储格式
运营数据	运营数据层（RAW）	raw_业务数据库表名(保持一致)_更新方式(如果增量同步加“_inc”,全量“_full”)_时间粒度	text / lzo
数据仓库	数据仓库层（DW）	日志：dw_log_业务_更新方式_时间粒度 业务数据库：dw_数据库类型（mysql hbase wtable redis）_业务_更新方式_时间粒度	parquet
	临时数据层（TMP）	tmp_数据层类型(dw dm)_业务	parquet
	维度数据层（DIM）	dim_维度类型（cate city channel group）	text / lzo / parquet
数据集市	数据主题层（DM）	dm_主题域(trade search recommend biz spam)_指标描述_更新方式_时间粒度	parquet
数据产品	应用数据层（ADS）	ads_报表描述_更新方式_时间粒度	/
元信息管理	元数据信息层（MDW）	mdw_业务_更新方式_时间粒度	/



数据资产管理

- 元数据管理
- 数据生命周期管理
- 存储、计算性能优化
- 权限管理

表信息

表字段

表信息

元数据信息

权限等级

与建表者具有相同权限等级的用户无需申请权限，其他部门用户均需要申请后才能使用

审批流程

审批流名称: flow_68000 () [查看审批流程](#)

设计者 / 建表者

所在部门

技术部

标签

血缘关系

快速筛选

数据层: 业务数据库 (biz) 快照表(rawdb) 事实明细层(dw) 集市层(dm) 字典维表层(dim)

业务类型: 全平台 风控 客服 渠道

数据主题: 用户 订单 日志 前端日志 后端日志

显示 15 项结果

表名	存储系统类型	库名
dm_trade_...tail_1d	64	hive
dm_trade_...ay_detail_1d	53	hive
dm_tr...ail_1d	38	hive
dm_trade_...detail_1d	27	hive
dm_trade_...tail_1d	24	hive

快速筛选 提示: 由于指标较多, 「标签/描述/统计规则」等补录工作仍在持续进行, 所以您的快速筛选标签尚未覆盖所有指标

计算类型: 离线 实时

数据类型: 用户 交易

业务类型: 平台 风控 渠道 具)

更多条件

全选

打标签

挂页面

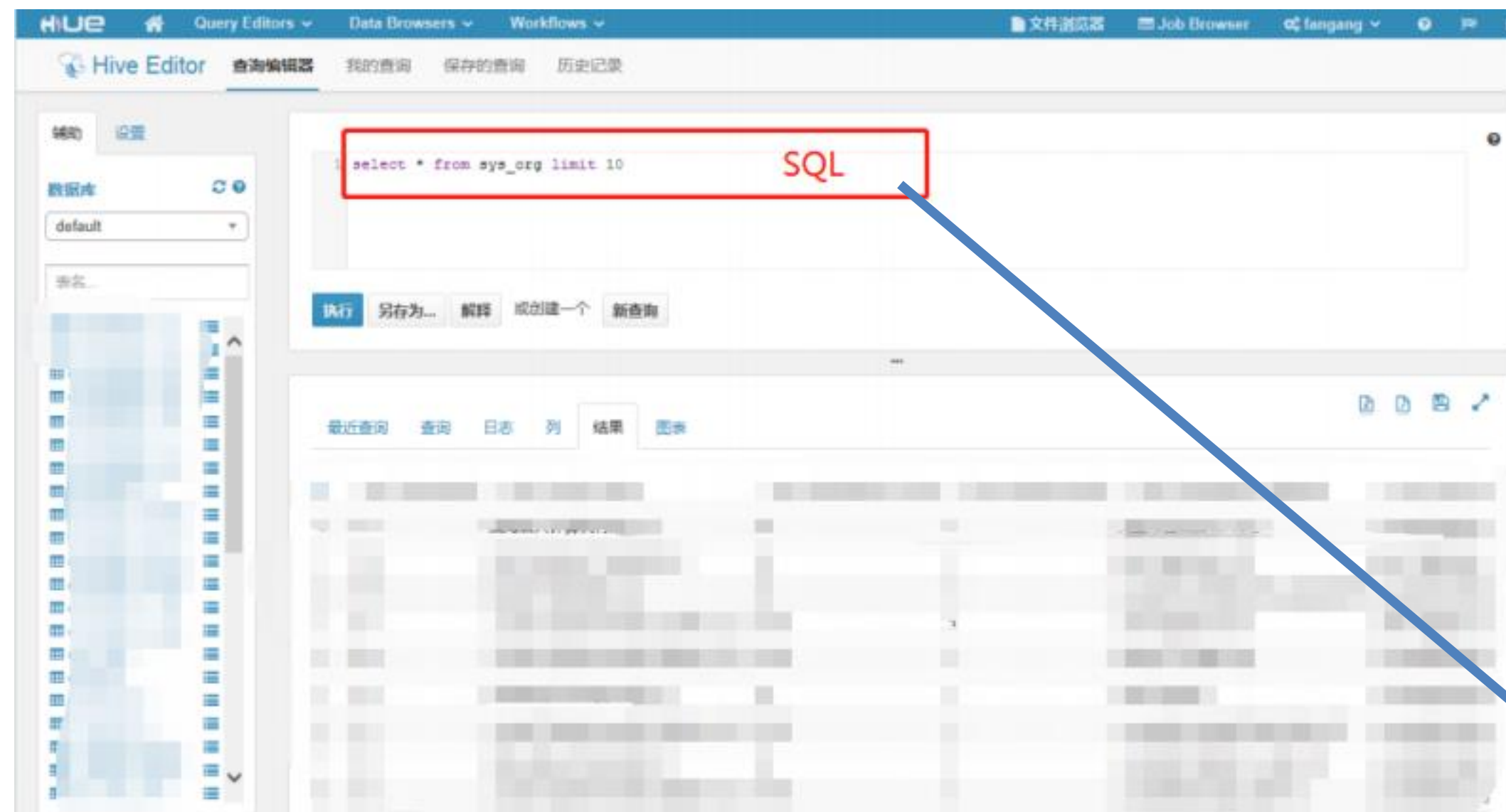
批量编辑

删除

显示 10 项结果

ID	指标名称	指标描述	统计规则
2995			
2986	新增用户数		只统计前端埋点数据
2985	DAU		包含所有终端的活跃用户数 (端内、微
2978	新注册用户		略

复用小数据场景SQL：从小数据场景迁移到大数据场景



```
object JxfoxNsr {  
  def main(args: Array[String]) {  
    val task = LogUtil.start("dw.agg.jxfoxNsr")  
    try {  
      val sc = SparkUtil.init("dw.agg.jxfoxNsr")  
      val hc = SparkUtil.getSqlContext(sc)  
      hc.udf.register("getDateKey", (dateString:String) =>  
        DateUtil.format(DateUtil.getTime(dateString),"yyyyMMdd").toLong)  
  
      val result = hc.sql("SELECT getDateKey(MX.KPRQ) DATE_KEY,MX.GF_NSRSBH GF_NSR_KEY , "+  
        " MX.GF_SWJG_DM GF_SWJG_KEY,NSR.NSRSBH XF_NSR_KEY,MX.XF_SWJG_DM XF_SWJG_KEY,MX.FP_LB,"+  
        " SUM(QD.WP_SL) WP_SL,SUM(QD.JE) JE,SUM(QD.SE) SE,COUNT(DISTINCT MX.JXFP_ID) FPFS "+  
        " FROM etl.ETL_JXFP MX INNER JOIN etl.ETL_JXFP_QD QD ON MX.JXFP_ID = QD.JXFP_ID "+  
        " INNER JOIN dw.DW_DM_NSR NSR ON MX.XF_NSRSBH = NSR.NSR_KEY"+  
        " WHERE (MX.ZFBZ='N' or MX.ZFBZ is null)"+  
        " GROUP BY getDateKey(MX.KPRQ) , MX.GF_NSRSBH, MX.GF_SWJG_DM,NSR.NSRSBH,"+  
        " MX.XF_SWJG_DM,MX.FP_LB,MX.XF_NSRMC,QD.SL")  
  
      DataFrameUtil.save(result, "dw", "dw_agg_jxfox_nsr")  
      LogUtil.end(task)  
    } catch { case ex:Exception => LogUtil.error(task, ex) }  
  }  
}
```


谢谢观看

Thanks for watching