离线数据仓库开发

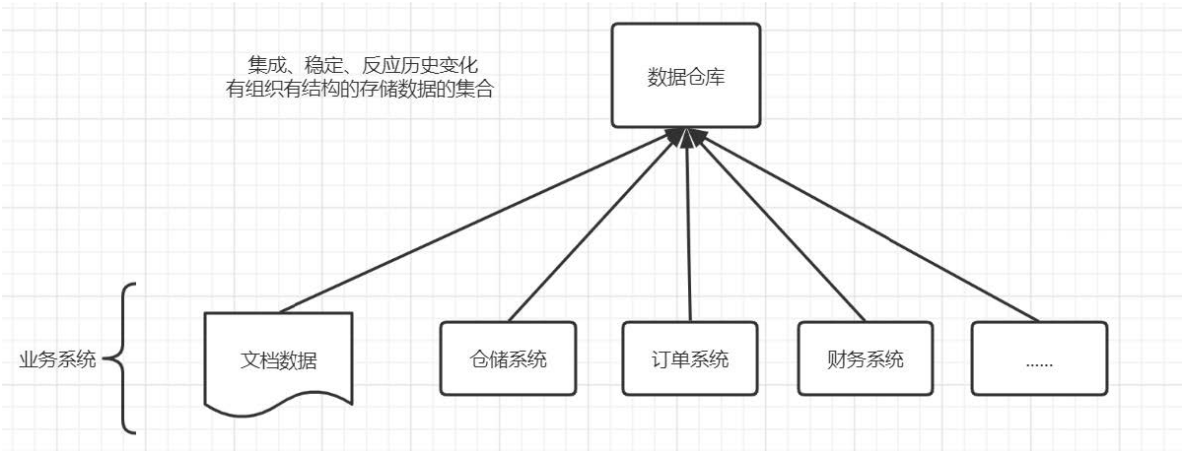
1. 数据仓库

1.1 数据库

- 引用百度百科的解释：数据库是“按照数据结构来组织、存储和管理数据的仓库”。是一个长期存储在计算机内的、有组织的可共享的、统一管理的大量数据的集合。
    - 数据库是长期储存在计算机内、有组织的、可共享的数据集合。
    - 数据库中的数据指的是以一定的数据模型组织、描述和储存在一起。
    - 具有尽可能小的冗余度、较高的数据独立性和易扩展性的特点并可在一定范围内为多个用户共享。

1.2 数据仓库

- 数据仓库，英文名称为 Data Warehouse，可简写为 DW 或 DWH。数据仓库，是为企业所有级别的决策制定过程，提供所有类型数据支持的战略集合。它是单个数据存储，出于分析性报告和决策支持目的而创建。 为需要业务智能的企业，提供指导业务流程改进、监视时间、成本、质量以及控制。数据仓库之父比尔·恩门（Bill Inmon）在1991年出版的 “Building the Data Warehouse”（《建立数据仓库》）一书中所提出的定义被广泛接受--数据仓库（Data Warehouse）是一个面向主题的（Subject Oriented）、集成的（Integrated）、相对稳定的（Non-Volatile）、反映历史变化（Time Variant）的数据集合，用于支持管理决策(Decision Making Support)。
    - 面向主题：在较高层次上将企业信息系统的数据综合归并进行分析利用的抽象的概念。每个主题基本上对应一个相应的分析领域。
    - 集成的：企业级数据，同时数据要保持一致性、完整性、有效性、精确性
    - 稳定的：从某个时间段来看是保持不变的，没有更新操作、删除操作，以查询分析为主
    - 实变：反应历史变化



1.3 数据库和数据仓库对比

- 面向业务的数据库常称作 OLTP 系统，关注增删改事务操作，面向分析的数据仓库亦称为 OLAP，关注查询分析。

功能	数据仓库	数据库
数据范围	存储历史的、完整的、反应历史变化的数据	当前状态的数据
数据变化	可添加、无删除、无变更的、反应历史变化	支持频繁的增、删、改、查操作
应用场景	面向分析、支持战略决策	面向业务交易流程
设计理论	不遵守范式、适当冗余	遵照范式、避免冗余
处理量	非频繁、批量大、高吞吐、有延迟	频繁、批量小、高并发、低延迟

1.4 数据中心

- 引用维基百科的解释：数据中心，指用于安置计算机系统及相关部件的设施，例如电信和储存系统。一般它包含冗余和备用电源，冗余数据通信连接，环境控制（例如空调、灭火器）和各种安全设备。
- 数据中心，顾名思义就是数据的中心，是处理和存储海量数据的地方，英文全称为 Data Center。用专业的名词解释，数据中心是全球协作的特定设备网络，用来在 internet 网络基础设施上传递、加速、展示、计算、存储数据信息。
- 一般来讲，数据中心主要有几大部分构成：机房、供配电系统、制冷系统、网络设备、服务器设备、存储设备、环境控制设备等。

1.5 数据平台

- 数据平台，一般叫做数据处理平台，不是一个专门被设计用来解决数据存储问题的，一个完整的数据平台包括一些关键架构设计：
    - 数据采集
    - 数据存储
    - 数据处理
    - 数据流转
    - 数据应用
- 除了提供基本的数据存储功能以外，还要提供数据采集，数据处理，数据应用等相关功能！这是数据平台和数据库或者数据仓库不同的地方！

1.6 数据湖

- 引用维基百科的解释：数据湖（英语：data Lake），是指使用大型二进制对象或文件这样的自然格式储存数据的系统[1]。它通常把所有的企业数据统一存储，既包括源系统中的原始副本，也包括转换后的数据，比如那些用于报表, 可视化, 数据分析和机器学习的数据。数据湖可以包括关系数据库的结构化数据(行与列)、半结构化的数据(CSV，日志，XML, JSON)，非结构化数据 (电子邮件、文件、PDF)和二进制数据(图像、音频、视频)。

- 数据湖是一种在系统或存储库中以自然格式存储数据的方法，它有助于以各种模式和结构形式配置数据，通常是对象块或文件。数据湖的主要思想是对企业中的所有数据进行统一存储，从原始数据（源系统数据的精确副本）转换为用于报告、可视化、分析和机器学习等各种任务的目标数据。数据湖中的数据包括结构化数据（关系数据库数据），半结构化数据（CSV、XML、JSON等），非结构化数据（电子邮件，文档，PDF）和二进制数据（图像、音频、视频），从而形成一个容纳所有形式数据的集中式数据存储。
- 数据湖从本质上来讲，是一种企业数据架构方法，物理实现上则是一个数据存储平台，用来集中化存储企业内海量的、多来源，多种类的数据，并支持对数据进行快速加工和分析。从实现方式来看，目前Hadoop是最常用的部署数据湖的技术，但并不意味着数据湖就是指Hadoop集群。为了应对不同业务需求的特点，MPP数据库 + Hadoop 集群+传统数据仓库这种“混搭”架构的数据湖也越来越多出现在企业信息化建设规划中。
- 数据湖的就是原始数据保存区. 虽然这个概念国内谈的少，但绝大部分互联网公司都已经有了。国内一般把整个HDFS叫做数据仓库（广义），即存放所有数据的地方，而国外一般叫数据湖（data lake）
- 数据湖和数据仓库的区别：

特性	数据仓库	数据湖
数据	来自业务系统，运营数据库和业务应用程序的关系数据	来自IOT设备，网站，移动应用，社交媒体，企业应用程序的非关系和关系数据
Schema	设计在数据仓库实施之前（写模式）	写入在读取数据分析时（读模式）
性价比	更快查询结果会带来较高存储成本	更快查询结果只需要较低存储成本
数据质量	可作为重要事实一句的高度监管数据	任何可以或无法进行监管的数据（原始数据）
用户	业务分析师	数据科学家，数据开发人员，业务分析师
分析	批处理报告，BI，可视化	机器学习，预测分析，数据发现和分析

1.7 数据中台

- 所谓数据中台，即实现数据的分层与水平解耦，沉淀公共的数据能力，主要包括数据模型，数据服务，数据开发三个方面的，解决企业的生产效率和团队协作的问题。
- 核心思想：OneData OneService
- 核心价值：经验沉淀 场景驱动
- 核心优势：避免重复建设，统一服务接口 沉淀通用能力，前台减负
- 团队组成：业务团队 数据团队 算法团队 工程团队

1.8 发展趋势

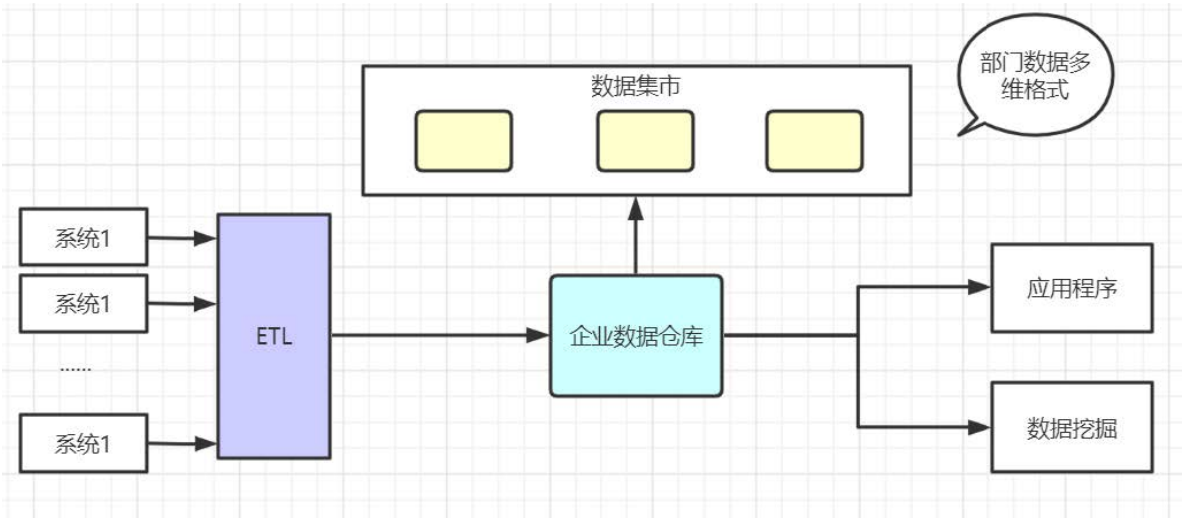
- 文件 --> 数据库 --> 数据仓库 --> 数据平台 --> 数据中台

1.9 数据仓库发展历程

- 萌芽阶段：数据仓库概念最早可追溯到 20 世纪 70 年代，MIT提出希望提供一种架构将业务处理系统和分析处理分为不同的层次
- 探索阶段：20 世纪 80 年代，建立 Technical Architecture2 规范，该明确定义了分析系统的四个组成部分：数据获取、数据访问、目录、用户服务。
- 雏形阶段：1988年，IBM第一次提出信息仓库的概念：一个结构化的环境，能支持最终用户管理其全部的业务，并支持信息技术部门保证数据质量；抽象出基本组件：数据抽取、转换、有效性验证、加载、cube开发等，基本明确了数据仓库的基本原理、框架结构，以及分析系统的主要原则。
- 确立阶段：1991年，Bill Inmon出版《Building the Data Warehouse》提出了更具体的数据仓库原则：面向主题的、集成的、包含历史的、不可更新的、面向决策支持的、面向全企业的、最明细的数据存储、快照式的数据获取。尽管有些理论目前仍有争议，但凭借此书获得数据仓库之父的殊荣。

1.10 Bill Inmon 数仓

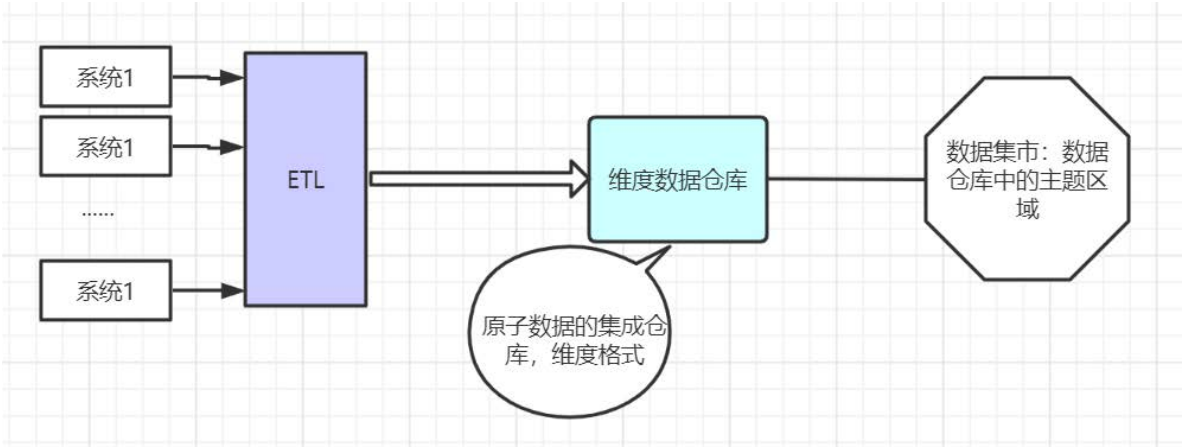
- 1991年，Bill Inmon 出版《Building the Data Warehouse》提出了更具体的数据仓库原则：
  - 数据仓库是面向主题的，集成的，包含历史的，不可更新的，面向决策支持的，面向全企业的，最明细的数据存储，数据快照式的数据获取
- Bill Inmon 凭借此书获得 “数据仓库之父” 的称号
- Bill Inmon 主张自上而下的建设企业数据仓库，认为数据仓库是一个整体的商业智能系统的一部分。一家企业只有一个数据仓库，数据集市的信息来源出自数据仓库，在数据仓库中，信息存储符合第三范式，大致架构：
  - 自上而下：分散异构的数据源 -> 数据仓库 -> 数据集市



- 操作型系统的数据和体系外数据需要经过 ETL 过程，加载到企业数据仓库中
- 企业数据仓库是企业信息化工厂的枢纽，是原子数据的集成仓库，其目的是将附加的数据存储用于各类分析型系统；在数据仓库中会对数据进行清洗，并抽取实体-关系。
- 数据集市是针对不同主题的聚集区域

1.11 Ralph Kimball 数仓

- Ralph Kimball 出版《The Data Warehouse Toolkit》，其主张自下而上的建立数据仓库，推崇建立数据集市，认为数据仓库是企业内所有数据集市的集合，信息总是被存储在多维模型当中，其思路：
  - Kimball 的模型是自下向上的，即从数据集市-> 数据仓库 -> 分散异构的数据源。



- Kimball 的模型的数据源往往是给定的若干个数据库表，数据较为稳定但是数据之间的关联关系比较复杂，需要从这些 OLTP中产生的事务型数据结构抽取出分析型数据结构。Kimball 是以最终任务为导向，将数据按照目标拆分出不同的表需求，通过 ETL 导入数据集市层
- Kimball 模型将分散异构的数据源经 ETL 转化为事实表和维度表导入数据集市，数据集市由若干个事实表和维度表组成
- 在数据集市将事实表和维度表根据分析主题组合后导入数据仓库中，用于数据分析

1.12 Inmon与Kimball 建模总结

- 两种思路和观点在实际的操作中都很难成功的完成项目交付，直至最终 Bill Inmon 提出了新的 BI 架构CIF(Corporation Information Factory)，把数据集市包含了进来。
- CIF 的核心是将数仓架构划分为不同的层次以满足不同场景的需求，比如常见的 ODS（Operational Data Store）、DW（Data Warehouse）、DM（Data Market）等，每层根据实际场景采用不同的建设方案，该思路也是目前数据仓库建设的架构指南，但自上而下还是自下而上的进行数据仓库建设，并未统一，并不是绝对的。
  - 共同点
    - 均极力推崇数据仓库，认为从OLTP到BI分析之间建立数据仓库是很有必要的；
    - 均认为数据仓库的建立需要从企业整体角度出发，迭代开发，尽量避免按部门建立独立的数据仓库；
    - 数据进入数据仓库之前，需要经过ETL整合。
  - 不同点（Inmon 理论）
    - 数仓概念：数据仓库（Data Warehouse）是一个面向主题的（Subject Oriented）、集成的（Integrated）、相对稳定的（Non-Volatile）、反映历史变化（Time Variant）的数据集合，用于支持管理决策（Decision Making Support）；
    - 自上而下按照主题建立数据仓库，如按照客户、供应商、产品等建立不同的主题，开发过程中每次增加一个主题；

        - 当建立的数据集市是跨多个主题的，需要以整合好的主题数据为基础。
    - Kimball 理论
        - 自下而上，维度建模；
        - 先按照业务主线建立最小粒度的事实表，再建立维度表，形成数据集市，通过“一致维度”能够共同看到不同数据集市的信息；

1.13 数据仓库应用前景

- 数据化运营
- 广告精准智能投放
- 用户画像，精准营销
- 数据挖掘、数据分析、人工智能、机器学习、无人驾驶
- ETC……

1.14. 基于大数据构建数仓

- 前述：
    - 1、随着我们从IT时代步入DT时代，数据从积累量也与日俱增，同时伴随着互联网的发展，越来越多的应用场景产生，传统的数据处理、存储方式已经不能满足日益增长的需求。而互联网行业相比传统行业对新生事物的接受度更高、应用场景更复杂，因此基于大数据构建的数据仓库最先在互联网行业得到了尝试。
    - 2、尽管数据仓库建模方法论是一致的，但由于所面临的行业、场景的不同，在互联网领域，基于大数据的数据仓库建设无法按照原有的项目流程、开发模式进行，更多的是需要结合新的技术体系、业务场景进行灵活的调整，以快速响应需求为导向。
- 应用场景广泛：
    - 1、传统的数仓建设周期长，需求稳定，面向CRM、BI等系统，时效性要求不高
    - 2、基于大数据的数据仓库建设要求快速响应需求，同时需求灵活、多变，对实时性有不同程度的要求，除了面向CRM、BI等传统应用外，还要响应用户画像、个性化推荐、机器学习、数据分析等各种复杂的应用场景。
- 技术栈更全面复杂：
    - 1、传统数仓建设更多的基于成熟的商业数据集成平台，比如Oracle、Teradata、Informatica等，技术体系比较成熟完善，但相对比较封闭，对实施者技术面要求也相对专业且单一，一般更多应用于银行、保险、通信等行业。
    - 2、基于大数据的数仓建设一般是基于非商业、开源的技术，常见的是基于Hadoop生态构建，涉及技术较广泛、复杂，同时相对于商业产品，稳定性、服务支撑较弱，需要自己维护更多的技术框架。
- 数仓模型设计更灵活：
    - 传统数仓有较为稳定的业务场景和相对可靠的数据质量，同时也有较为稳定的需求，对数仓的建设有较为完善的项目流程管控，数仓模型设计有严格的、稳定的建设标准。

    - 在互联网行业：行业变化快、业务灵活，同时互联网又是个靠速度存活的行业，数据源种类繁多：半结构化、非结构化、结构化数据，数据质量相对差，层次不齐
- 所以，在互联网领域，数仓模型的设计更关注灵活、快速响应和应对多变的市场环境，更加以快速解决业务、运营问题为导向，快速数据接入、快速业务接入，不存在一劳永逸。

2. 数仓构建流程

- 一个完整的关于数仓构建的流程：

```
01、需求分析
02、逻辑分析
03、ODS建模
04、数据仓库建模
05、数据源分析
06、数据获取和整合
07、应用分析
08、数据展现
09、性能调优
10、元数据管理
```

2.1 需求分析

- 需求来源/需求调研方式：客户访谈 和 调查问卷
- 对企业领导层：
    - 领导层对数据仓库的期望是什么?
    - 领导层最关心哪几个指标?
    - 领导层希望以何种方式来看这些指标?
    - 领导层希望对这些指标进行哪些方面的比较?
    - ……
- 对中间管理层：
    - 中间管理层对数据仓库的期望是什么?
    - 中间管理层希望以何种方式来看这些指标?
    - 平时领导层通常询问哪些指标?在这些指标中哪几个和此分析主题有关?
    - 中间管理层对下属的工作人员都考核哪些指标?哪几个指标与此分析主题有关?
- 对业务人员：
    - 业务人员对数据仓库的期望是什么?
    - 业务人员希望系统能提供哪些分析功能?
    - 业务人员希望以何种方式来看这些指标?
    - 业务人员希望对这些指标进行哪些方面的比较?
- 对IT技术人员：
    - 此主题所需要的数据源都取自哪些业务系统?

    - 与本主题有关的现有的业务系统的数据结构怎样?
    - IT人员对数据仓库的期望是什么?
    - IT人员在平时的工作中最关心的哪些指标?

2.2 逻辑分析

- 处理逻辑分析
    - 单一主题处理逻辑分析：从业务逻辑入手，分析各指标的组成关系
    - 多主题处理逻辑分析：综合考虑分析主题之间的逻辑关系
- 支撑数据分析
    - 单一主题支撑数据分析：单个主题分析所需要的原始支撑数据分析
    - 多主题支撑数据分析：所有主题统一考虑做需要的支撑数据分析
- 业务元数据建立
    - 使用者的业务术语所表达的数据模型、对象名和属性名;
    - 访问数据的原则和数据来源;
    - 系统所提供的分析方法及公式、报表信息。

2.3 ODS建模

- 逻辑模型:
    - 逻辑结构（完成实体的定义，各实体间的关系等）
    - 存储粒度（与源系统基本保持一致）
    - 存储周期（立即删除、过一段时间删除或者是备份到其它介质上）
- 物理模型:
    - 数据的存储结构
    - 索引策略
    - 数据存放位置（硬盘或磁带等）
    - 存储分配
    - 分区设计

2.4 数据仓库建模

- 数据仓库逻辑模型:
    - 划分粒度层次
    - 确定数据分割策略
    - 确定存储周期
    - 定义关系模式
- 数据仓库物理模型:
    - 数据的存储结构
    - 索引策略
    - 数据存放位置（硬盘或磁带等）

- 存储分配
- 分区设计

2.5 数据源分析

- 数据源范围：
  - 包括数据源逻辑范围和物理范围
- 数据源格式
  - 理解各数据源的格式，确定统一的格式，制定相应的转换规则

数据更新频率
数据量
数据质量

2.6 数据获取和整合

- 直接抽取：主要面向 业务数据库
  - ETL服务器直接连接到应用系统后台数据库中，直接抽取所需数据。
  - 采用这种抽取方式时，必须注意安全控制和抽取时间窗口两个问题。
- Web服务和数据收集工具：主要面向 网络流数据
  - 通过WEB服务获取系统需要的数据的抽取方式
- 文件收集：主要面向日志文件
  - 文件交换是指应用系统将需要抽取的业务数据保存为有格式的文本文件，然后ETL服务器通过读此文件内容来获取业务数据的数据抽取方式。
- 数据的整合：数据的 ETL：抽取，转换，装载
  - 字段映射
  - 代码转换
  - 字段拆分
  - 字段合并
  - 字段运算
  - 字段补充
  - 行列转换
  - 全部覆盖
  - 记录追加
  - 记录更新

2.7 应用分析

- 分析方法：OLAP有多种实现方法，根据存储数据的方式不同可以分为ROLAP、MOLAP、HOLAP
  - 1、ROLAP
  - 2、MOLAP

- 3、HOLAP

名称	描述	细节数据存储位置	聚合后的数据存储位置
ROLAP(Relational OLAP)	基于关系数据库的OLAP实现	关系型数据库	关系型数据库
MOLAP(Multidimensional OLAP)	基于多维数据组织的OLAP实现	数据立方体	数据立方体
HOLAP(Hybrid OLAP)	基于混合数据组织的OLAP实现	关系型数据库	数据立方体

- 预定义报表
  - 1、对单报表可以直接从数据库中取出数据进行分析展现。
  - 2、同一主题的多个报表间有较强的关联，有些数据会在多个报表中以不同方式出现。因此，可以对多个报表进行整合。
- 即席查询
  - 1、基于单个表的即席查询
  - 2、基于多个事实表关联的即席查询
- 数据挖掘
  - 根据数据功能的类型和和数据的特点选择相应的算法，在净化和转换过的数据集上进行数据挖掘。

2.8 数据展现

- 主要数据展现格式：文字不如表，表不如图，但是不能直接干掉表的存在
  - 1、汇报文案
  - 2、报表
  - 3、图形
- 最佳形式：提供自定义指标选择和条件筛查并且能下载报表明细数据的统一规范化可视化web平台。echarts

2.9 性能调优

- 主要优化目标：
  - 1、优化指标
  - 2、优化步骤
  - 3、优化系统

2.10. 元数据管理

- 在数据处理过程中，涉及到的流程比较多，因此，元数据主要有：

```
1、数据源元数据
2、ETL元数据
3、数据仓库元数据
4、数据集市元数据
5、前端展示元数据
6、数据挖掘元数据
7、其他元数据

元数据管理
1、元数据模型采用公共仓库元模型(Common Warehouse Metamodel，简称CWM) 。CWM的主要目的是在异构环境下，帮助不同的数据中心工具、平台和元数据知识库进行元数据交换。
2、CWM为数据仓库和商业智能（BI）工具之间共享元数据，制定了一整套关于语法和语义的规范。
3、元数据管理涉及到数据仓库构造、运行、维护的整个生命周期，是数据仓库构建过程中十分重要的一环。元数据以数据库存储，集中管理控制。

现在已经有了，专门用来做元数据管理的技术：Atlas
```

3. 数仓建模基本理论

3.1 建模目标

- 数据模型就是数据组织和存储方法，它强调从业务、数据存取和使用角度合理存储数据。Linux 的创始人 Torvalds 有 一段关于“什么才是 优秀程序员”的话：“烂程序员关心的是代码，好程序员关心的是数据结构和它们之间的关系”，其阐述了数据模型的重要性。有了适合业务 和基础数据存储环境的模型，那么大数据就能获得以下好处。
    - 访问性能：能够快速查询所需的数据，减少数据I/O
    - 数据成本：减少不必要的数据冗余，实现计算结果数据复用，降低大数据系统中的存储成本和计算成本
    - 使用效率：改善用户应用体验，提高使用数据的效率
    - 数据质量：改善数据统计口径的不一致性，减少数据计算错误的可能性，提供高质量的、一致的数据访问平台
- 所以，大数据的数仓建模需要通过建模的方法更好的组织、存储数据，以便在性能、成本、效率和数据质量之间找到最佳平衡点。
    - 以空间换时间 以时间换空间
        - 1、以空间换时间： Join 维度表 事实表 ===> 大宽表
        - 2、以时间换空间： 本身一张大事实表 ==> 事实表 + 多张维度表

3.2 关系型数据库范式

- 设计关系数据库时，遵从不同的规范要求，设计出合理的关系型数据库，这些不同的规范要求被称为不同的范式。
    - 关系模式范式：关系型数据库设计时，遵照一定的规范要求，目的在于降低数据的冗余性和数据的一致性，目前业界范式有

```
第一范式(1NF)：字段不可分，每个字段是原子级别的，多个字段组织在一起形成一个字段是违反第一范式的
第二范式(2NF)：有主键，非主键字段依赖主键
第三范式(3NF)：非主键字段不能相互依赖
巴斯-科德范式(BCNF)：在3NF基础上，任何非主字段不能对主键子集依赖
第四范式(4NF)：在满足3NF的基础之上，表中不能包含一个实体的两个或多个互相独立的多值因子
第五范式(5NF)完美范式：在满足4NF的基础之上，表必须可以分解为较小的表，除非那些表在逻辑上拥有与
原始表相同的主键
```

- 各种范式呈递次规范，越高的范式数据库冗余越小。有冗余的数据库未必是最好的数据库， 有时为了提高运行效率，就必须降低范式标准，适当保留冗余数据。
- 满足最低要求的范式是第一范式（1NF）。在第一范式的基础上进一步满足更多规范要求的称为第二范式（2NF），其余范式以次类推。一个数据库设计如果符合第二范式，一定也符合第一范式。如果符合第三范式，一定也符合第二范式。一般说来，数据库只需满足第三范式(3NF）就行了。总之，规范化的过程就是在数据库表设计时移除数据冗余的过程。随着规范化的进行，数据冗余越来越少，但数据库的效率也越来越低。

3.3 ER 对象关系实体模型

- 在信息系统中（CMS, ERM, OA等），将事物抽象为“实体”、“属性”、“关系”来表示数据关联和事物描述；实体：Entity，关系：Relationship；这种对数据的抽象建模通常被称为 ER 实体关系模型。
  - 实体：通常为参与到过程中的主体，客观存在的，比如商品、仓库、汽车。此实体非数据库的实体表
  - 属性：对主体的描述、修饰即为属性，比如商品的属性有商品名称、颜色、尺寸、重量、产地等
  - 关系：现实的物理事件是依附于实体的，比如商品入库事件，依附实体商品、货位，就会有“库存”的属性产生；用户购买商品，依附实体用 户、商品，就会有“购买数量”、“金额”的属性产品。
- 实体之间建立关系时，存在对照关系：

```
1:1 即1对1的关系，比如实体人、身份证，一个人有且仅有一个身份证号
1:n 即1对多的关系，比如实体学生、班级，对于某1个学生，仅属于1个班级，而在1个班级中，可以有多个学生
n:m 即多对多的关系，比如实体学生、课程，每个学生可以选修多门课程，同样每个课程也可以被多门学生选修

在日常建模过程中：所以 ER 实体关系模型也可以称作 E-R 关系图
“实体”用矩形表示
“关系”用菱形表示
“属性”用椭圆形表示

应用场景：
ER模型是数据库设计的理论基础，当前几乎所有的OLTP系统设计都采用ER模型建模的方式
Bill Inom提出的数仓理论，推荐采用ER关系模型进行建模
```

BI架构提出分层架构，数仓底层ods、dwd、dm也多采用ER关系模型就行设计

3.4 维度模型

- 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。其设计分为以下几个步骤：
    - 1、选择需要进行分析决策的业务过程。业务过程可以是单个业务事件，比如交易的支付、退款等；也可以是某个事件的状态，比如当前的账户余额等；还可以是一系列相关业务事件组成的业务流 程，具体需要看我们分析的是某些事件发生情况，还是当前状态， 或是事件流转效率。
    - 2、选择粒度。在事件分析中，我们要预判所有分析需要细分的程度，从而决定选择的粒度。粒度是维度的一个组合。
    - 3、选择维度。选择好粒度之后，就需要基于此粒度设计维表，包括维度属性，用于分析时进行分组和筛选。
    - 4、选择事实。确定分析需要衡量 的指标。Ralph Kimball 推崇数据集市的集合为数据仓库，同时也提出了对数据集市的维度建模，将数据仓库中的表划分为事实表、维度表两种类型。
    - Ralph Kimball 推崇数据集市的集合为数据仓库，同时也提出了对数据集市的维度建模，将数据仓库中
      的表划分为事实表、维度表两种类型。

3.4.1 事实表

- 在 ER 模型中抽象出了有实体、关系、属性三种类别，在现实世界中，每一个操作型事件，基本都是发生在实体之间的，伴随着这种操作事件的发生，会产生可度量的值，而这个过程就产生了一个事实表，存储了每一个可度量的事件。
- 电商场景：一次购买事件，涉及主体包括客户、商品、商家，产生的可度量值，包括商品数量、金额、件数等，所以订单明细表，就是一张事实表。
- 事实表设计原则：摘自《大数据之路：阿里巴巴大数据实践》
    - 原则1: 尽可能包含所有与业务过程相关的事实。事实表设计的目的是为了度量业务过程，事实越多，越有利于多角度多维度度量业务
    - 原则2: 只选择与业务过程相关的事实。比如下单事件，不应该存储支付金额
    - 原则3: 分解不可加性事实为可加的组件。比如订单的优惠率，应该分解为订单的原价与订单优惠金额两个事实存储在事实表中
    - 原则4: 在选择维度和事实之前必须先声明粒度。粒度是维度的组合。先确定粒度，再确定维度。粒度用于确定事实表中一行所表示业务的细节层次，决定了维度模型的扩展性，在选择维度和事实之前必须先声明粒度，且每个维度和事实必须与所定义的粒度保持一致。
    - 原则5: 在同一个事实表中不能有多种不同粒度的事实。事实表中的所有事实需要与表定义的粒度保持一致，在同一个事实表中不能有多种不同粒度的事实。
    - 原则6: 事实的单位要保持一致，对于同一个事实表中事实的单位应该保持一致。比如订单的金额、订单优化金额、订单运费金额这三个事实，应该采用一致的计量单位，统一为元或者

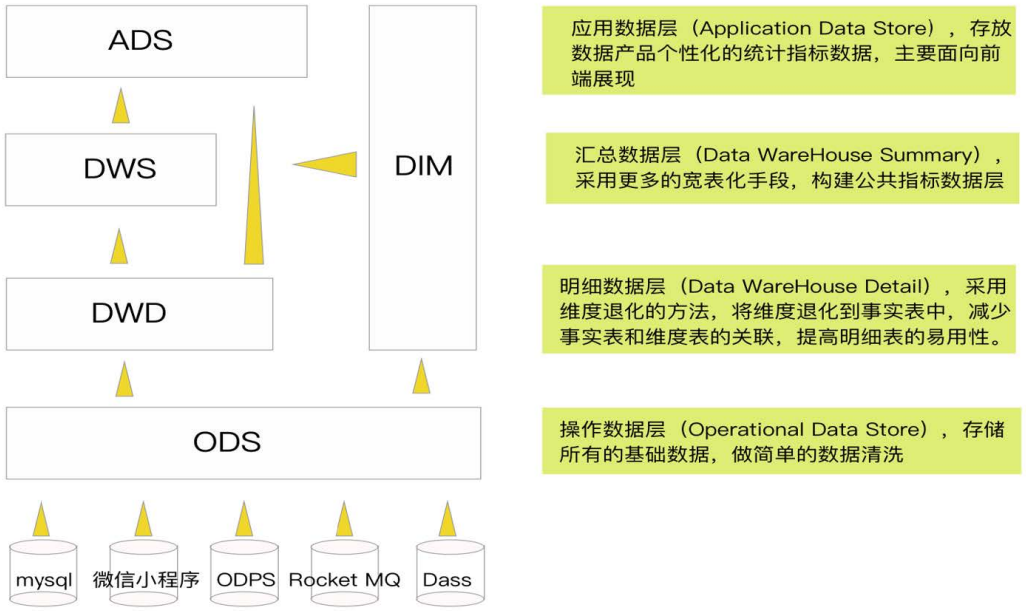
分，方便实用。

- 原则7: 对事实的 null值要处理。对于事实表中事实度量为null 值的处理，因为在数据库中null值对常用数字型字段的sql过滤条件都不生效，比如大于，小于，等于…，建议用零值填充。
- 原则8: 使用退化维度提高事实表的易用性。这样设计的主要目的是为了减少下游用户使用时关联多个表进行操作。直接通过退化维度实现事实表的操作。通过增加存储的冗余，提高计算的速度。空间置换时间的方式。

3.4.2 模型分层

- 分层通用做法：

```
ODS(Operational Data Store) 原始数据层
DWD(Data Warehouse Detail) 明细数据层
DWS(Data Warehouse Service) 汇总数据层
ADS(Application Data Store) 数据应用层
```



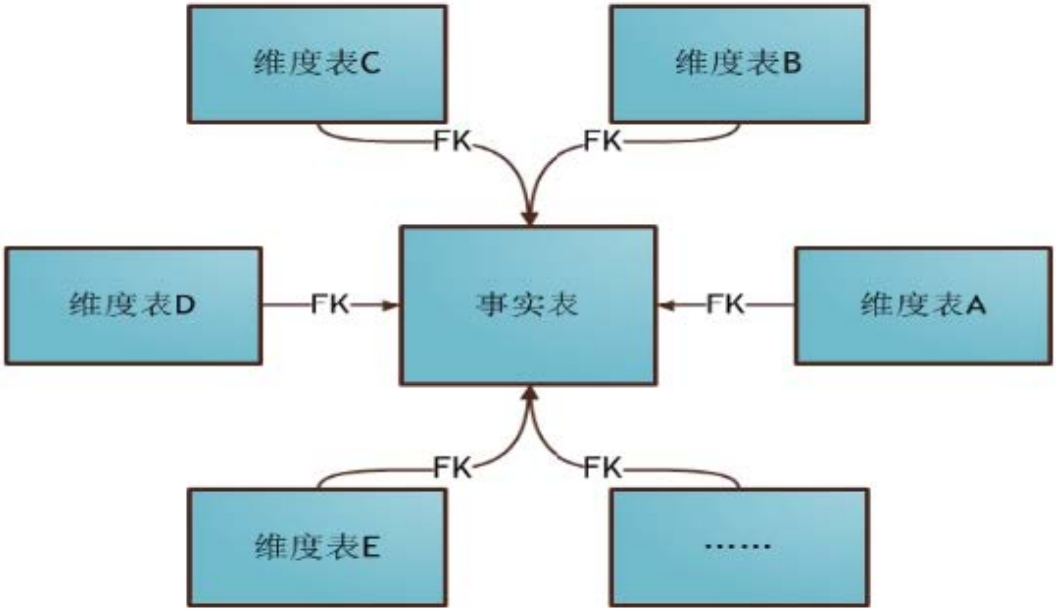
- 总体原则：
  - 1、数据库名、数据表名、字段名全部小写，单数形式，使用"_"进行分割；
  - 2、不准出现大写与复数表达形式，要见名知意；
  - 3、不能出现SQL语法中关键字，例如：table等；
  - 4、最好形成统一的，业界共识的标准和规范，比如阿里的 MySQL规范
- 关于表命名的一些企业最佳实践：
  - 1、ODS层命名为ods开头 + DWD层命名为dwd开头 + DWS层命名为dws开头 + ADS层命名为ads开头
  - 2、表名的命名中，包含分层信息，包含可选的系统标识，业务主题，粒度，维度信息等
  - 3、临时库表命名为xxxxx_tmp + 备份库表命名为xxxxx_bak

3.5 模型分类

- 在构建数据仓库的维度建模中，一般有三种模式。星型模型/雪花模型/星座模型
- 在多维分析的商业智能解决方案中，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。

3.5.1 星型模型

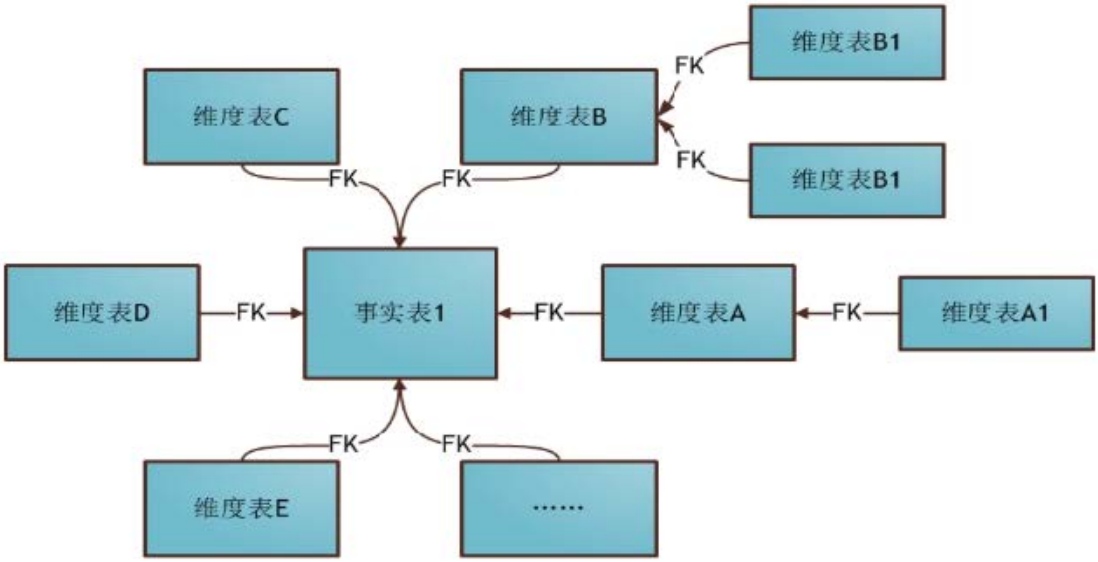
- 当所有维表都直接连接到“ 事实表”上时，整个图解就像星星一样，故将该模型称为星型模型。 星型架构是一种非正规化的结构，多维数据集的每一个维度都直接与事实表相连接，不存在渐变维度，所以数据有一定的冗余



- 星形模型的维度建模由一个事实表和一组维度表组成，有以下特点：
    - 1、维度表只和事实表关联，维度表之间没有关联；
    - 2、每个维表的主码为单列，且该主码放置在事实表中，作为两边连接的外码；
    - 3、以事实表为核心，维表围绕核心呈星形分布；

3.5.2 雪花模型

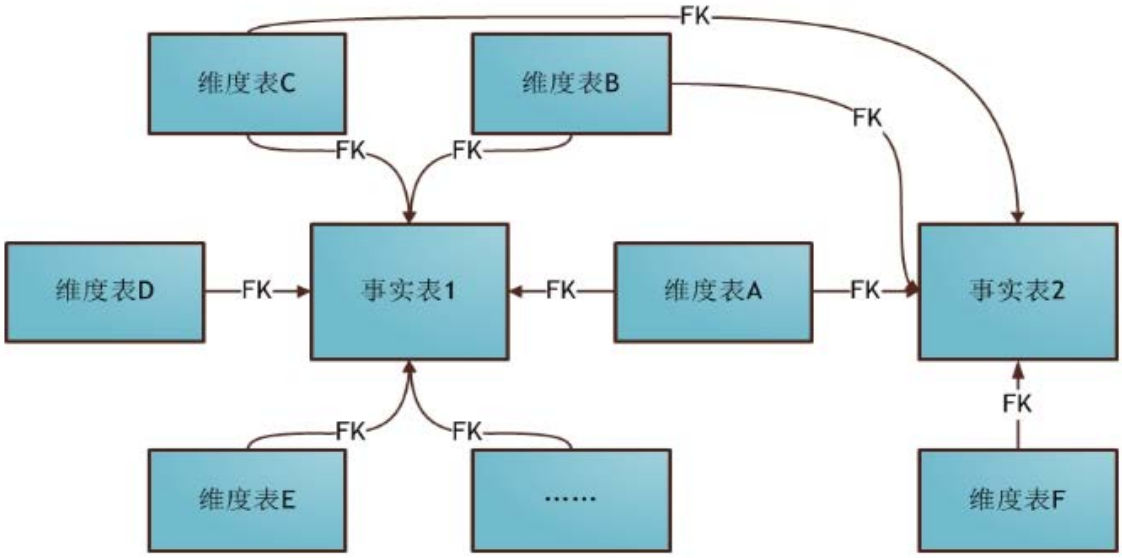
- 当一个或多个维表没有直接连接到事实表上，而是通过其他维表连接到事实表上时，其图解就像多个雪花连接在一起，故称雪花模型。雪花模型是对星型模型的扩展。它对星型模型的维表进一步层次化，原有的各维表可能被扩展为小的事实表，形成一些局部的"层次"区域，这些被分解的表都连接到主维度表而不是事实表。
- 优点：通过最大限度地减少数据存储量以及联合较小的维表来改善查询性能。雪花型结构去除了数据冗余。



- 雪花模型的特点：
  - 1、将星形模式的大维度表拆分未小维度表，满足规范化设计，但不利于开发。
  - 2、星型模型因为数据的冗余所以很多统计查询不需要做外部的连接，因此一般情况下效率比雪花型模型要高。在冗余可以接受的前提下，实际运用中星型模型使用更多，也更有效率。
  - 3、在雪花模型中，数据模型的业务层级是由一个不同维度表主键-外键的关系来代表的。而在星形模型中，所有必要的维度表在事实表中都只拥有外键。

3.5.3 星座模型

- 星座模式也是星型模式的扩展。



- 星座模型特点：
  - 1、星型模型和雪花模型都是基于多个维表对应事实表，有的时候一个维度表可能被多个事实表用到，这个时候就需要采用星座模式。

3.5.4 模型对比

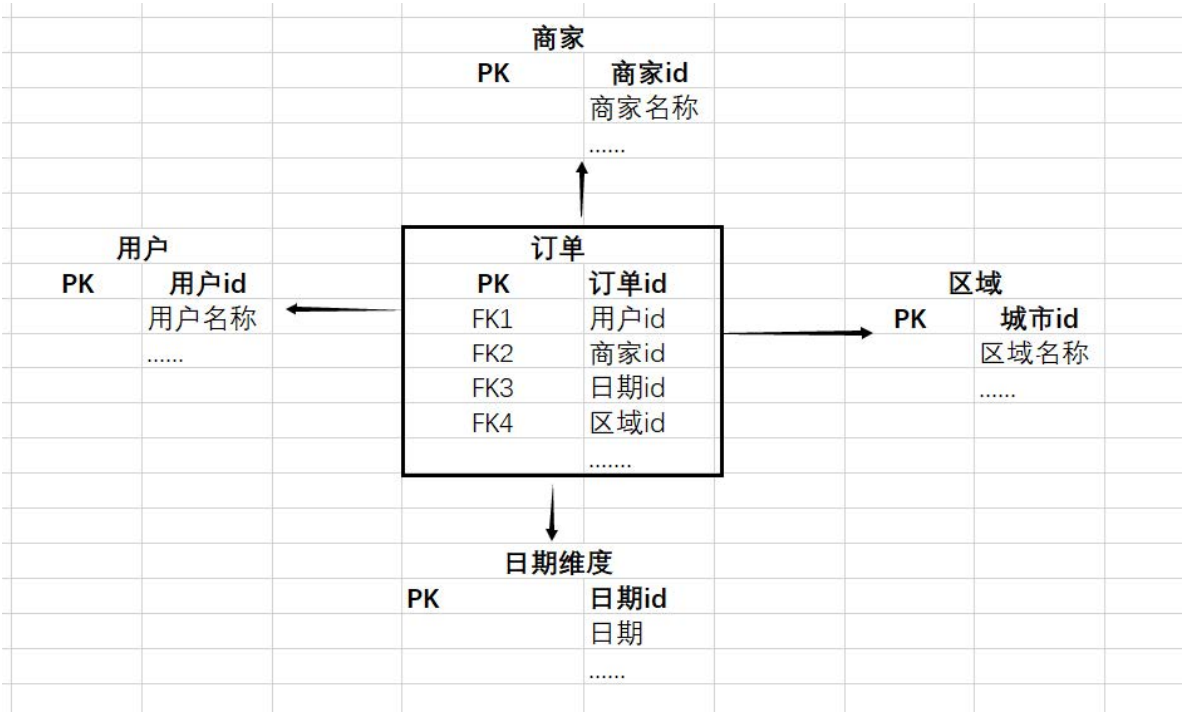
- 雪花模型是将星型模型的维度表进一步划分，使得各维度表满足规范化设计。而星座模型允许星型模型中出现多个事实表，更符合实际业务需求。 雪花模型使得维度分析更加容易。
- 综上所述可以看出：星型模型和雪花模型主要区别就是对维度表的拆分：
  - 对于雪花模型，维度表的涉及更加规范，一般符合3NF；
  - 而星型模型，一般采用降维的操作，利用冗余来避免模型过于复杂，提高易用性和分析效率
- 星型模型和雪花模型的主要区别：

```
冗余：
雪花模型符合业务逻辑设计，采用3NF设计，有效降低数据冗余；
星型模型的维度表设计不符合3NF，反规范化，维度表之间不会直接相关，牺牲部分存储空间。
性能：
雪花模型由于存在维度间的关联，采用3NF降低冗余，通常在使用过程中，需要连接更多的维度表，导致性能偏低；
星型模型反三范式，采用降维的操作将维度整合，以存储空间为代价有效降低维度表连接数，性能较雪花模型高。
ETL处理：
雪花模型符合业务ER模型设计原则，在ETL过程中相对简单，但是由于附属模型的限制，ETL任务并行化较低；
星型模型在设计维度表时反范式设计，所以在ETL过程中整合业务数据到维度表有一定难度

Note：下面的案例，对于商品再拆分出厂家、品类；用户的常住地再拆分出区域维度，改造为雪花模型。
```

3.6 建模案例

- 电商平台（比如转转），经常需要对订单进行分析，以购物订单为例，以维度建模的方式设计该模型。
- 涉及到事实表为订单表、订单明细表，维度包括商品维度、用户维度、商家维度、区域维度、时间（日期）维度
- 事实表中的维度：
  - 商品维度：商品ID、商品名称、商品种类、单价、来源地等
  - 用户维度：用户ID、姓名、性别、年龄、常住地、职业、学历等
  - 时间（日期）维度：日期ID、日期、周几、上/中/下旬、是否周末、是否假期、特殊日期等
  - 优惠券：券ID、券类别/优惠方式、优惠金额
- 订单中包含的度量：商品件数、总金额、总减免
  - 描述性属性：下单时间、支付时间、订单状态等
- 订单明细包含度量：商品ID、件数、单价、减免金额
  - 描述性属性：加入购物车时间、状态



- 根据上图的设计可以发现，其实就是一种典型的星型模型，如果区域维度，拆解成多个渐变维度，则相当于去除了冗余，则变成雪花模型。

4. 离线数仓案例--电商用户行为分析

4.1 项目背景

- 当今社会有很多的电商网站，这就存在着一些竞争关系，为了更好的设计一个网站，让一个电商网站浏览的人数更多，从而增加点击量和订阅量的，这样就需要我们对这个电商网站进行分析和数据挖掘，我们可以根据每天浏览某电商网站的人数和访客量来判断一个网站的好坏和受欢迎程度，同时也可以根据外链的跳转率和访客或会员所用的浏览器等工具的分析来进行精准的广告推广，我们也可以根据地区的点击量和访客或是会员访问的时间的分析来进行合理的商品推广，精准的推荐等操作。同时每一个电商网站，可以根据这个网站的支付订单数以及成功支付订单数来进行业务的分析，这些对于提高一个网站的点击量、浏览量、以及成功支付订单量都是必不可少的。
- 网站分析（Web Analytics）主要指的是基于网站的用户浏览行为，即指用户访问网站时的所有访问、浏览、点击行为数据。比如点击了哪一个链接，在哪个网页停留时间最多，采用了哪个搜索项、总体会话时间等。而所有这些信息都可被保存在网站日志中。通过分析这些数据，可以获知许多对网站运营至关重要的信息。采集的数据越全面，分析就能越精准。对网站的点击流数据和运营数据进行分析，以监控网站的运营状况，为网站的优化提供决策依据。网站分析系统已成为站长日常运营必不可少的工具，业界比较流行的网站分析系统主要有Google Analytics、CNZZ和百度统计等产品。
- 总之，一个电商网站就应该设计出一款产品能让用户的体验好，能让用户精准的寻找想要购买的商品，能提高用户的转化率，能提广告的转化率。

4.2 项目意义

- 网站流量统计分析，可以帮助网站管理员、运营人员、推广人员等实时获取网站流量信息，并从流量来源、网站内容、网站访客特性等多方面提供网站分析的数据依据。从而帮助提高网站流量，提升网站用户体验，让访客更多的沉淀下来变成会员或客户，通过更少的投入获取最大化的收入。
- 如下表：

网站的眼睛	网站的神经	网站的大脑
访问者来自哪里? 访问者在寻找什么? 哪些页面最受欢迎? 访问者从哪里进入? 访问者从哪里跳出?	网页布局合理吗? 网站导航清晰吗? 哪些功能存在问题 网站内容有效吗 转化路径靠谱吗?	如何分解目标? 如何分配广告预算? 如何衡量产品表现? 哪些产品需要优化? 哪些指标需要关注?

- 点击流分析的意义可分为两大方面：
    - 技术上，可以合理修改网站结构及适度分配资源，构建后台服务器群组，比如
        - 1、辅助改进网络的拓扑设计，提高性能
        - 2、在有高度相关性的节点之间安排快速有效的访问路径
        - 3、帮助企业更好地设计网站主页和安排网页内容
- 2、业务上，比如
    - 1、帮助企业改善市场营销决策，如把广告放在适当的Web页面上。
    - 2、优化页面及业务流程设计，提高流量转化率。
    - 3、帮助企业更好地根据客户的兴趣来安排内容。
    - 4、帮助企业对客户群进行细分，针对不同客户制定个性化的促销策略等
- 终极目标是：改善网站(电商、社交、电影、小说)的运营，获取更高投资回报率（ROI）

4.3 点击流数据模型

4.3.1 日志

日志的生成渠道
1、是网站的web服务器软件（apache、nginx、tomcat）所记录的web访问日志；
2、是通过在页面嵌入自定义的JS代码来获取用户的所有访问行为（比如鼠标悬停的位置，点击的页面组件等），然后通过AJAX请求到后台记录日志；这种方式所能采集的信息最全面；
3、通过在页面上埋点1像素的图片，将相关页面访问信息请求到后台记录日志；

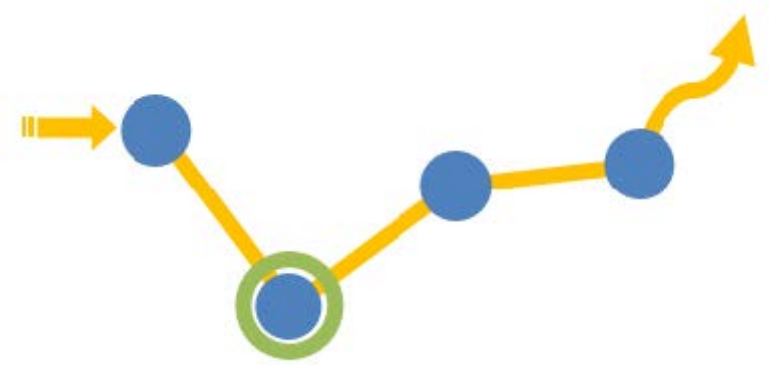
日志数据内容详述
在实际操作中，有以下几个方面的数据可以被采集：
1、访客的系统属性特征。比如所采用的操作系统、浏览器、域名和访问速度等。
2、访问特征。包括停留时间、点击的URL、所点击的“页面标签<a>”及标签的一些属性（比如业务entity<商品、电影、歌曲、小说名称>的名称）等。
3、来源特征。包括来访URL，来访IP等。
4、产品特征。包括所访问的产品编号、产品类别、产品颜色、产品价格、产品利润、产品数量和特价等级等。

日志数据案例

```
GET /log.gif?t=item.010001&m=UA-J2011-1&pin=-
&uid=1679790178&sid=1679790178|12&v=je=1$sc=24-
bit$sr=1600x900$ul=zhcn$cs=GBK$dt=xxxx$hn=item.jd.com$fl=16.0r0$os=win$br=chrome$
bv=39.0.2171.95$wb=1
437269412$xb=1449548587$yb=1456186252$zb=12$cb=4$usc=direct$ucp=-$umd=none$uct=-$
ct=1456186505411$lt=0$tad=-$sku=1326523$cid1=1316$cid2=1384$cid3=1405$brand=20583
$pinid=-&ref=&rm=1456186505411 HTTP/1.1
```

4.3.2 点击流

- 点击流这个概念更注重用户浏览网站的整个流程，网站日志中记录的用户点击就像是图上的“点”，而点击流更像是将这些“点”串起来形成的“线”。也可以把“点”认为是网站的Page，而“线”则是访问网站的Session。所以点击流数据是由网站日志中整理得到的，它可以比网站日志包含更多的信息，从而使基于点击流数据统计得到的结果更加丰富和高效。



- 点击流数据在具体操作上是由散点状的点击日志数据梳理所得，从而，点击数据在数据建模时应该存在两张模型表（pageviews和visits）：
- 用于生成点击流的原始访问日志表

时间戳	IP地址	请求URL	Referal	响应吗	流量	信息
2012-01-01 12:31:12	101.0.0.1	/a/...	somesite.com			
2012-01-01 12:31:16	201.0.0.2	/a/...	google.com			
2012-01-01 12:33:06	201.0.0.2	/a/...	google.com			
2012-01-01 15:16:39	234.0.0.3	/a/...	google.com			
2012-01-01 15:17:11	101.0.0.1	/a/...	google.com			
2012-01-01 12:39:23	201.0.0.2	/a/...	google.com			

- 页面点击流模型 pageviews 表

Session	IP地址	时间	访问页面URL	停留时长	第几步
S001	101.0.0.1	2012-01-01 12:31:12	/a/....	30	1
S001	101.0.0.1	2012-01-01 12:31:12	/a/....	310	1
S001	101.0.0.1	2012-01-01 12:31:12	/a/....	130	3
S001	101.0.0.1	2012-01-01 12:31:12	/a/....	230	2
S001	101.0.0.1	2012-01-01 12:31:12	/a/....	330	4
S001	101.0.0.1	2012-01-01 12:31:12	/a/....	430	6

- 点击流模型visits表(按session聚集的页面访问信息)

Session	起始时间	结束时间	进入页面	离开页面	IP	cookie	referal
S001	2012-01-01 12:31:12	2012-01-01 12:31:12	/a/...	/a/...	101.0.0.1	User01	somesite.com
S001	2012-01-01 12:31:12	2012-01-01 12:31:12	/a/...	/a/...	101.0.0.1	User01	somesite.com
S001	2012-01-01 12:31:12	2012-01-01 12:31:12	/a/...	/a/...	101.0.0.1	User01	somesite.com
S001	2012-01-01 12:31:12	2012-01-01 12:31:12	/a/...	/a/...	101.0.0.1	User01	somesite.com
S001	2012-01-01 12:31:12	2012-01-01 12:31:12	/a/...	/a/...	101.0.0.1	User01	somesite.com

- 这就是点击流模型。当WEB日志转化成点击流数据的时候，很多网站分析度量的计算变得简单了，这就是点击流的“魔力”所在。基于点击流数据我们可以统计出许多常见的网站分析度量指标

4.4 常见分析指标

```
PV
UV
VV
独立IP
停留时长
访问深度
```

```
转化率
跳出率
退出率
ROI
RFM
GMV
订单数
订单总金额
客单价
……
```

4.5 经典分析需求实现

4.5.1. 基础流量分析

- 比如一些非常核心的指标：PV，IP，UV，独立访客
    - 1、趋势分析：根据选定的时段，提供网站流量数据，通过流量趋势变化形态，为您分析网站访客的访问规律、网站发展状况提供参考。
    - 2、对比分析：根据选定的两个对比时段，提供网站流量在时间上的纵向对比报表，帮您发现网站发展
      状况、发展规律、流量变化率等。
    - 3、当前在线：提供当前时刻站点上的访客量，以及最近15分钟流量、来源、受访、访客变化情况等，
      方便用户及时了解当前网站流量状况。
    - 4、访问明细：提供最近7日的访客访问记录，可按每个PV或每次访问行为（访客的每次会话）显示，并可按照来源、搜索词等条件进行筛选。 通过访问明细，用户可以详细了解网站流量的累计过程，从而为用户快速找出流量变动原因提供最原始、最准确的依据。

4.5.2. 来源分析

- 1、来源分类：提供不同来源形式（直接输入、搜索引擎、其他外部链接、站内来源）、不同来源项引入流量的比例情况。通过精确的量化数据，帮助用户分析什么类型的来路产生的流量多、效果好，进而合理优化推广方案。
- 2、搜索引擎：提供各搜索引擎以及搜索引擎子产品引入流量的比例情况。从搜索引擎引入流量的的角度，帮助用户了解网站的SEO、SEM效果，从而为制定下一步SEO、SEM计划提供依据。
- 3、搜索词：提供访客通过搜索引擎进入网站所使用的搜索词，以及各搜索词引入流量的特征和分布。帮助用户了解各搜索词引入流量的质量，进而了解访客的兴趣关注点、网站与访客兴趣点的匹配度，为优化SEO方案及SEM提词方案提供详细依据。
- 4、最近7日的访客搜索记录：可按每个PV或每次访问行为（访客的每次会话）显示，并可按照访客类型、地区等条件进行筛选。为您搜索引擎优化提供最详细的原始数据。
- 5、来路域名：提供具体来路域名引入流量的分布情况，并可按“社会化媒体”、“搜索引擎”、“邮箱”等网站类型对来源域名进行分类。帮助用户了解哪类推广渠道产生的流量多、效果好，进而合理优化网站推广方案。

- 6、来路页面：提供具体来路页面引入流量的分布情况。尤其对于通过流量置换、包广告位等方式从其他网站引入流量的用户，该功能可以方便、清晰地展现广告引入的流量及效果，为优化推广方案提供依据。
- 7、来源升降榜：提供开通统计后任意两日的TOP100搜索词、来路域名引入流量的对比情况，并按照变化的剧烈程度提供排行榜。用户可通过此功能快速找到哪些来路对网站流量的影响比较大，从而及时排查相应来路问题。

4.5.3. 受访分析

- 1、受访域名：提供访客对网站中各个域名的访问情况。一般情况下，网站不同域名提供的产品、内容各有差异，通过此功能用户可以了解不同内容的受欢迎程度以及网站运营成效。
- 2、受访页面：提供访客对网站中各个页面的访问情况。 站内入口页面为访客进入网站时浏览的第一个页面，如果入口页面的跳出率较高则需要关注并优化；站内出口页面为访客访问网站的最后一个页面，对于离开率较高的页面需要关注并优化。
- 3、受访升降榜：提供开通统计后任意两日的TOP100受访页面的浏览情况对比，并按照变化的剧烈程度提供排行榜。可通过此功能验证经过改版的页面是否有流量提升或哪些页面有巨大流量波动，从而及时排查相应问题。

  4、热点图：记录访客在页面上的鼠标点击行为，通过颜色区分不同区域的点击热度；支持将一组页面设置为"关注范围"，并可按来路细分点击热度。通过访客在页面上的点击量统计，可以了解页面设计是否合理、广告位的安排能否获取更多佣金等。

  5、用户视点：提供受访页面对页面上链接的其他站内页面的输出流量，并通过输出流量的高低绘制热度图，与热点图不同的是，所有记录都是实际打开了下一页面产生了浏览次数（PV）的数据，而不仅仅是拥有鼠标点击行为。

  6、访问轨迹：提供观察焦点页面的上下游页面，了解访客从哪些途径进入页面，又流向了哪里。通过上游页面列表比较出不同流量引入渠道的效果；通过下游页面列表了解用户的浏览习惯，哪些页面元素、内容更吸引访客点击。

4.5.4. 访客分析

- 1、地区运营商：提供各地区访客、各网络运营商访客的访问情况分布。地方网站、下载站等与地域性、网络链路等结合较为紧密的网站，可以参考此功能数据，合理优化推广运营方案。
- 2、终端详情：提供网站访客所使用的浏览终端的配置情况。参考此数据进行网页设计、开发，可更好地提高网站兼容性，以达到良好的用户交互体验。
- 3、新老访客：当日访客中，历史上第一次访问该网站的访客记为当日新访客；历史上已经访问过该网站的访客记为老访客。新访客与老访客进入网站的途径和浏览行为往往存在差异。该功能可以辅助分析不同访客的行为习惯，针对不同访客优化网站，例如为制作新手导航提供数据支持等。
- 4、忠诚度：从访客一天内回访网站的次数（日访问频度）与访客上次访问网站的时间两个角度，分析访客对网站的访问粘性、忠诚度、吸引程度。由于提升网站内容的更新频率、增强用户体验与用户价值可以有更高的忠诚度，因此该功能在网站内容更新及用户体验方面提供了重要参考。
- 5、活跃度：从访客单次访问浏览网站的时间与网页数两个角度，分析访客在网站上的活跃程度。由于提升网站内容的质量与数量可以获得更高的活跃度，因此该功能是网站内容分析的关键指标之一。

4.5.5. 转化路径分析

转化定义：访客在您的网站完成了某项您期望的活动，记为一次转化，如注册或下载。
目标示例：

1、获得用户目标：在线注册、创建账号等。
2、咨询目标：咨询、留言、电话等。
3、互动目标：视频播放、加入购物车、分享等。
4、收入目标：在线订单、付款等。

转化数据的应用

1、在报告的自定义指标中勾选转化指标，实时掌握网站的推广及运营情况。
2、结合“全部来源”、“转化路径”、“页面上下游”等报告分析访问漏斗，提高转化率。
3、对“转化目标”设置价值，预估转化收益，衡量ROI。

路径分析

根据设置的特定路线，监测某一流程的完成转化情况，算出每步的转换率和流失率数据，如注册流程，购买流程等。

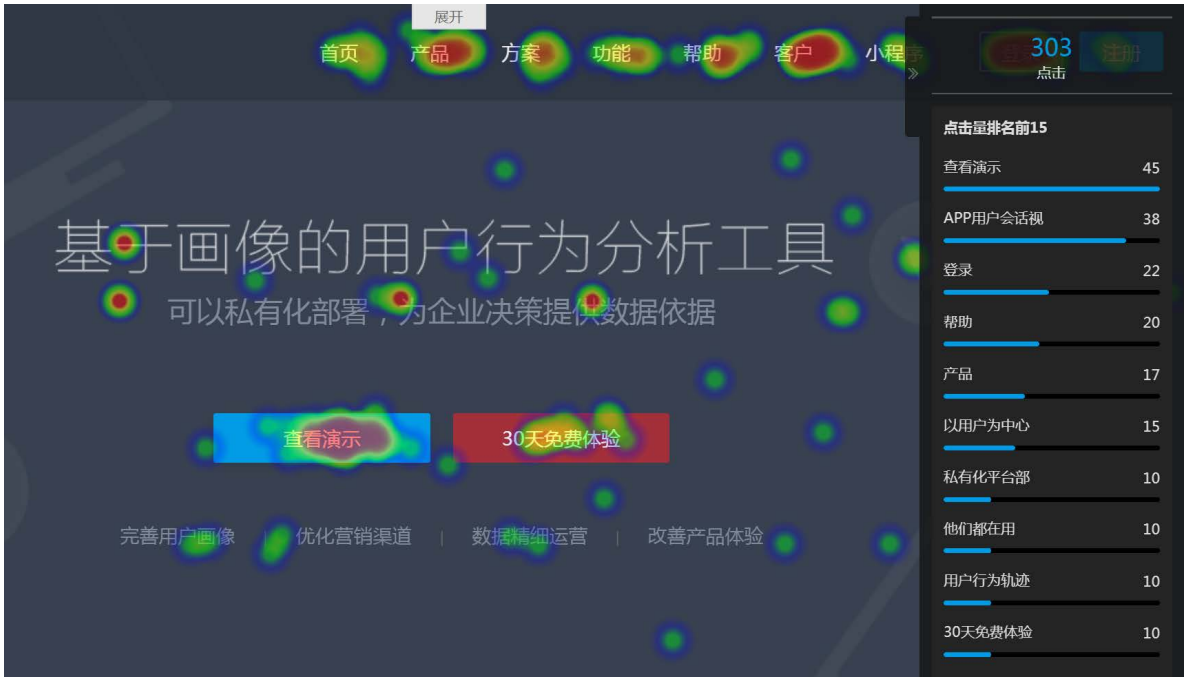
4.5.6. 用户分析

主要分析新增会员（如果按天分析，就是昨天注册的会员）、活跃会员（只要今天登陆的就是活跃会员，或者按照一定的周期去计算，只要在这个周期内都是活跃会员）以及总会员相关信息

1、访客：主要分析新增用户、活跃用户以及总用户的相关信息
2、会员：主要分析新增会员、活跃会员以及总会员的相关信息
3、会话：主要分析会话个数，会话长度和平均会话长度相关的信息
4、主要分析每天每小时的用户、会话个数以及会话长度的相关信息

4.5.7. 其他分析模块等

浏览器分析模块
地域分析模块
用户浏览深度分析模块
订单分析模块
热力图分析
投入产出分析（ROI）
RFM模型分析

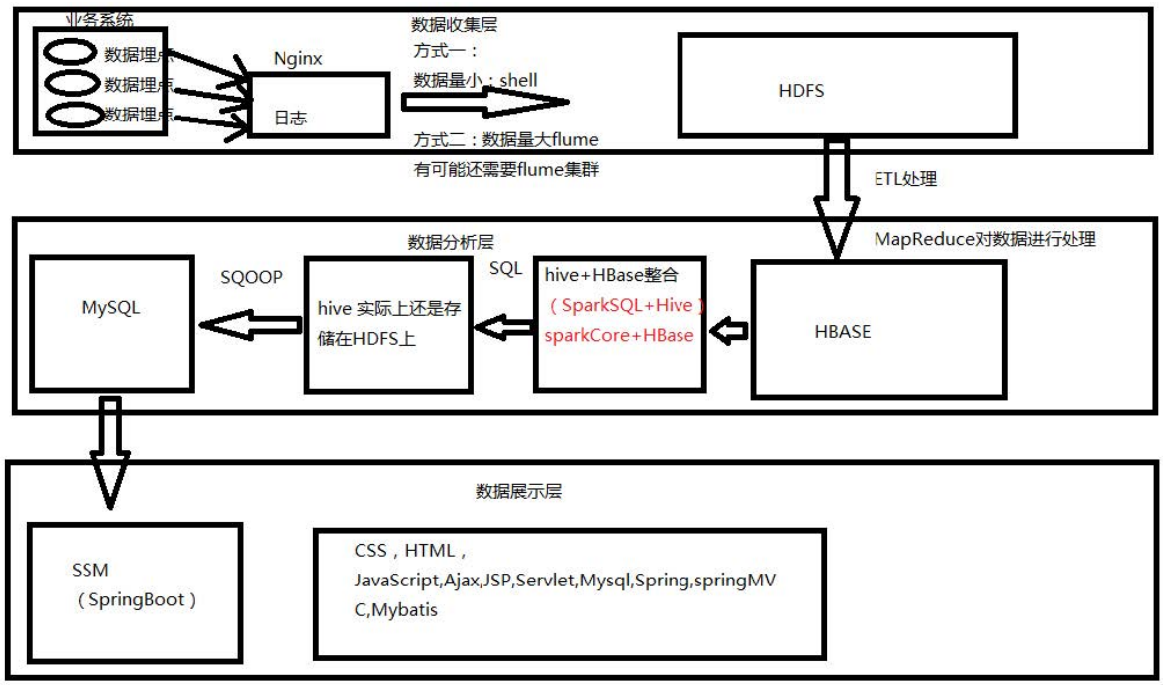


4.6. 数据展现

> “
>
> 数据展现的目的是将分析所得的数据进行可视化，以便运营决策人员能更方便地获取数据，更快更简单地理解数据

4.7. 项目技术架构

- 基本上，这一套架构几乎是离线架构的万能架构。总体上分为以下三个部分：
  - 1、数据分析层
  - 2、数据处理层
  - 3、数据展示层



4.7.1. 数据处理流程

- 该项目是一个纯粹的数据分析项目，其整体流程基本上就是依据数据的处理流程进行，依此有以下几个大的步骤：

1、数据采集

```
首先，通过页面嵌入JS代码的方式获取用户访问行为，并发送到web服务的后台记录日志
然后，将各服务器上生成的点击流日志通过实时或批量的方式汇聚到HDFS文件系统中。当然，一个综合分析系统，数据源可能不仅包含点击流数据，还有数据库中的业务数据（如用户信息、商品信息、订单信息等）及对分析有益的外部数据。
```

2、数据预处理

```
通过mapreduce程序对采集到的点击流数据进行预处理，比如清洗，格式整理，滤除脏数据等
```

3、数据入库

```
将预处理之后的数据导入到HIVE仓库中相应的库和表中
```

4、数据分析

```
项目的核心内容，即根据需求开发ETL分析语句，得出各种统计结果
```

5、数据展现

```
将分析所得数据进行可视化
```

4.7.2. 数据收集层

```
数据收集层涉及到写数据埋点，数据埋点可分为两类：
    A：前台数据埋点：
        使用JavaScript去写，问题：为什么不用JQuery写。
    B：后台数据埋点：
        使用Java去写
        Flume：收集日志（如果实时的还需要Kafka）
        MapReduce：对数据进行预处理
```

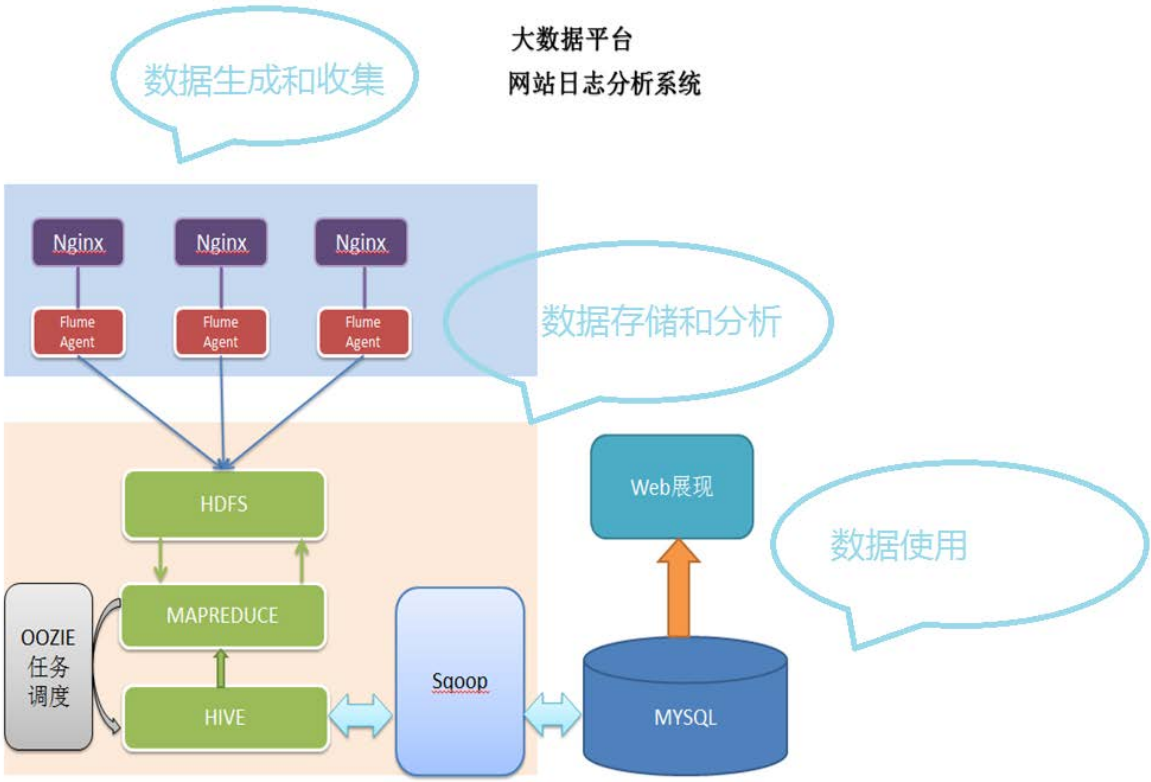
4.7.3. 数据分析层

```
MapReduce/ SparkCore
Hive/ SparkSQL
Kylin/pig/impala/drill
Oozie/azkaban/crontab
Hive + HBase(SQL)
HBase + Phoenix
```

4.7.4. 数据展示层

```
Sqoop 导入到 MySQL 或是 Oracle/PostGre (关系型数据库)/ HBase
结合三大框架:
    Spring + Sturts2 + MyBatis ==> SSM ss2m
    Spring + Sturts2 + hibernate ==> SSH ss2h
    Spring + SpringMVC + MyBatis
    SpringBoot + Spring + MyBatis等
    Echarts/Highcharts
    Tableau、Bootstrap、EasyUI、D3.js
```

4.7.5. 项目技术架构



- 其中，需要强调的是：
    - 系统的数据分析不是一次性的，而是按照一定的时间频率反复计算，因而整个处理链条中的各个环节需要按照一定的先后依赖关系紧密衔接，即涉及到大量任务单元的管理调度，所以，项目中需要添加一个任务调度模块。

5. 未来数据仓库发展

5.1. 开发调度系统

- 任务管理
- 运行调度
- 依赖处理
- 抽取推送

5.2. 数据质量

- 一致性
- 完整性
- 合理性
- 及时性

5.3. 血缘分析

- 异常分析
- 产出分析
- 依赖分析

5.4. 开放查询平台

- 数据查询
- 权限管理
- 资源管理
- Query历史管理
- Query分析

5.5. 数据图谱

表元数据
日志元数据
ETL元数据
指标口径
数据生命周期

5.6. 未来发展

数据权限管理
数据使用记录
主数据管理
工具集成
数据平台开放
生命周期管理

6. 数仓建设的体会

1、数据展现的开发和准确的数据，是能否做好仓库的基础。形式很重要。
2、只要有好的数据就可以开展一定的工作，不一定要等应用系统建设成功才开展。
3、数据模型并不是最重要的事情。分析模型的建立往往取决于分析的要求。对于大家追求的数据共享和分析
的通用模型，取决于数据源，这个工作应该在业务系统层面去实现。通用的共享视图模型可以是虚拟存在的，
比如采用数据库视图来实现。
4、数据集中很重要，没有数据的集中，就会失去动力和基础。数据有了量的积库累，一定是有文章做的，分
析模型的抽象一定是在大量数据的基础上的。对于每个业务系统，我们一定要厂家提供数据库设计文档，最好
安排资源对文档与数据库的一致性进行检查。这个工作做好了，其实可以少很多后续的协调工作。
5、完整意义上的一体化建设，会存在多次迭代和反复。数据分析的要求，会促使业务系统的升级和改造。同
时业务系统的升级，也会提供新的数据，同时引发新的数据展现要求。
6、商业智能分析工具也呈一体会趋势，目前整合趋势整体格局已定，未来就是以几大软件巨头竞争，我们没
有太多的选择。
7、早期建设，不建议做过多的数据处理，以便于核对数据的准确性。模型加工的对应性一定要可直接追溯。