# 数据仓库概述

经过多年IT的建设，信息对于XXX的日常管理已经日益重要，并逐渐成为重要的信息资产，信息资产的管理已经成为日常管理中一个非常重要的环节。如何管理和利用好XXX内部纷繁的数据也越来越成为信息管理的一项重要工作。

在过去相当一段时间内，XXX业务系统的构建主要围绕着业务的数据展开，应用的构建多是自下而上构建，主要以满足某个部门的业务功能为主，我们称之为业务处理的时代。这样的构建方式造成了一个个分立的应用，分立的应用导致了一个个的静态竖井。由于数据从属于应用，缺乏XXX全局的单一视图，形成了一个个信息孤岛，分立的系统之间缺乏沟通，同样数据的孤岛导致只能获得片面的信息，而不是全局的单一视图。存储这些信息的载体可能是各种异构或同构的关系型数据库，也有可能是XML、EXCEL等文件。因此，构建新一代的一体化平台提上了日程并最终促成全域数据的管理方式，目的是覆盖XXX各个环节的关键业务数据，完善元数据管理，形成全局的数据字典、业务数据规范和统一的业务指标含义，能够灵活的获取XXX业务数据的单一视图（需要保证数据的一致性、完整性、准确性和及时性）。数据的交换和共享主要发生在上下级组织机构之间或同级的不同部门之间。最终，这些数据可以为部队分析、决策支持（多维分析、即席查询、数据挖掘）等应用提供更及时、准确、有效的支持。

数据仓库的目标是实现跨系统数据共享，解决信息孤岛，提升数据质量，辅助决策分析，提供统一的数据服务。同时，数据仓库的构建也面临着各种挑战，比如信息整合在技术上的复杂度、信息整合的管理成本、数据资源的获取、信息整合的实施周期以及整合项目的风险等。

# 全域数据库总体架构

**全域数据库总体架构**

全域数据库总体的层次，最下面是基础架构层，主要包括支撑这一架构运行的主机系统、存储备份系统、网络系统等内容。从下往上看，再上面是数据源层，既包括各个业务的关系型数据源、内容管理数据源也包括半结构化数据源比如XML、EXCEL等，也包括各个总队、支队的业务数据源。

数据源层之上是“交换服务体系”，主要包括信息服务总线和服务总线两部分。信息服务总线主要实现数据层的信息整合和数据转换，而服务总线主要实现应用层的信息交换和整合。信息服务总线主要依托联邦、复制、清洗、转换等技术实现，其主要包括信息整合服务和清洗转换加载服务两部分。通过信息服务总线的信息整合服务（数据联邦、复制），可以透明、实时的访问分布在总队和支队的各个业务系统中的各种同构、异构数据（前提是拥有足够的权限）。信息整合服务在整个XXX层面保证了数据的完整性和及时性。信息服务主要使用两种技术来完成这一功能：联邦和复制。通过联邦功能可以把关系数据、半结构化数据（如 Excel文件、XML 文件、Web 搜索引擎、MQ 查询和内容源）组成一个逻辑数据库，对这些数据源中的表可以像操作本地数据库表一样进行操作，而不必关心我们操作的这些数据底层是什么数据源，物理在什么位置。而针对大数据量的数据访问或高并发的访问，通常将源数据增量实时复制到本地，复制的实现是基于对源数据库的日志进行捕获，获取增量数据，并基于消息的机制将其复制到目的数据库，复制的过程中可以实现数据的合并、拆分、转换等操作。

信息服务总线主要完成数据的分析、清洗（标准化）、转换、加载等工作。数据清洗，主要是去除冗余数据，将零散字段合并成全局记录，并解决重叠和矛盾的数据，然后通过添加关系和层次结构完善丰富信息。首先面临的挑战就是如何更有效的识别现有的业务系统，包括业务系统使用的分类方法、层次结构、数据分布、数据字典等。如果数据字典不完整或缺失，就要通过方法找出其数据的存储结构以及各个表之间的主外键关联、各表之间的转换关系等，同样，数据的分布情况同样可以使用分析功能来完成。在对现有数据足够了解的基础上（完成了数据的分析），接下来就要制定数据的清洗规则以及转换规则，其中，清洗规则又分为两种情况，一种清洗规则是明确的，另一种清洗规则是模糊的，比如不同系统中存储的地址信息，“南京市定淮门大街9号”和“江苏省南京市下关区定淮门大街9号”实际上是一个地址，但计算机会当成两个地址来处理。概率匹配功能和动态权重策略可以匹配创建高质量、准确的数据，并在整个数据域中一致地识别核心业务信息，如人名、位置、和时间。

数据清洗、转换、加载服务对保障数据的准确性和一致性非常重要。在不同的系统中，对同一业务会使用不同的分类方法，同样，数据的类别和层次结构也会不同。需要通过数据清洗、转换、加载层实现对这些信息格式的转换，匹配成通用的信息格式和分类方法，以便提供整个XXX业务层面聚合的业务视图。实际证明一体化平台（一期）的全域数据梳理中，手工统计可以完成这项工作但不够好，不够直观和没有扩展延续能力。数据清洗、转换、加载工作对未来数据的使用非常重要，即使有工具帮助，工作量依然很大。虽然，开始的时候，这项工作看起来费时费力，但从长远来看，它使得基于这些数据的业务流程和统一数据视图实现自动化，并减少了人为干预不准确或不一致数据的努力，从而节省了大量成本。XXX层面的单一视图一经建立，其维护必将是一个持续进行的过程。

数据的管理通常需要一个管理组织来对冲突或缺失的数据进行决策，组织会通常需要各个业务部门的人参与，而不仅仅局限于通技处或信息中心的人。XXX单一数据视图的维护，很多业务部门都做的不够好，时间一久，很多业务部门就变得厌倦，数据清洗转换的工作没有坚持下去。一旦数据的准确性出现问题，业务系统的全局共享就无法再从中获益。

交换服务体系中的服务总线主要基于流程服务、传输服务、交换服务等实现。通过使用总线，可以支持各种协议以及数据格式的数据交互。通过搭建一个基于标准的、开发的、易于集成的、总线方式的服务总线，通过今后对现有系统的逐步升级改造，系统之间以一种成为“服务”的接口方式统一通过总线方式进行交互，通过对服务的管理，系统之间交互的信息格式的差异、传输协议的差异、采用技术的差异、物理位置的不同等等这些问题都由这个总线来进行屏蔽。进一步通过流程管理，将模块和系统之间的服务按照业务流程的需要进行编排，做到了“随需而变”。

数据存储区包括ODS、数据仓库/数据集市、共享数据库、特征库、模型库等，主要提供各种数据的存储服务。其中，逻辑视图中ODS部分存放了整个XXX单位全局级的明细数据，而数据仓库数据集市中存储了不同级别的汇总数据。特征库主要存放各种数据分群特征、业务分类特征等业务信息，模型库存放构建的各种业务模型信息等。

基础服务层主要包括“应用服务器”，“服务总线”，“工作流引擎”，“消息中间件”，“OLAP引擎”，“数据挖掘引擎”，“事件驱动”，“规则引擎”，“协同工作”和“空间地理数据引擎”。

应用层包括各种应用，其中多维分析、即席查询、报表统计、图形展现等。

右边的信息治理层主要是为了保证数据的完整性、一致性、准确性、及时性，保证历史数据正确归档并在需要的时候可以和现有数据一起被联合访问，提供数据库安全、审计、监控和合规服务，从而防止内部人员偷窃，防范欺诈作假，保护数据隐私，强制执行安全规范，强制满足合规的要求，防止外部攻击对数据的破坏。

而元数据管理会贯穿数据业务层面、业务系统、信息整合服务总线、ETL层、数据存储区、信息服务层、展现层等各个层面，当数据口径出现问题时，能够提供数据在各个层面的正向/逆向追踪功能。元数据的管理涉及业务元数据和技术元数据两种。

# 数据仓库架构

**数据仓库总体架构**

**多维分析**

从数据的全方位了解现状，管理人员往往希望从不同的角度来审视业务数值，比如从时间、地域、类别、功能来看同一类数据的总和。每一个分析的角度可以叫做一个维，因此，把多角度分析方式称为多维分析。以前，每一个分析的角度需要制作一张报表。由此产生了在线多维分析功能，根据用户常用的多种分析角度，事先计算好一些辅助结构，以便在查询时能尽快抽取到所要的记录，并快速地从一维转变到另一维，将不同角度的信息以数字、直方图、饼图、曲线等等方式展现在您面前。

**即席查询**

可以将数据进行查询分组，进行资源的管理，可以设置查询优先级，可以自动控制，调度复杂查询和进行跟踪分析查询。可以按照以下重要方法进行使用，主动和动态地控制 数据库的查询流程，为不同大小的查询定义不同的查询类别，从而改善查询之间的系统资源共享，避免较小的查询被较大的查询阻塞等。

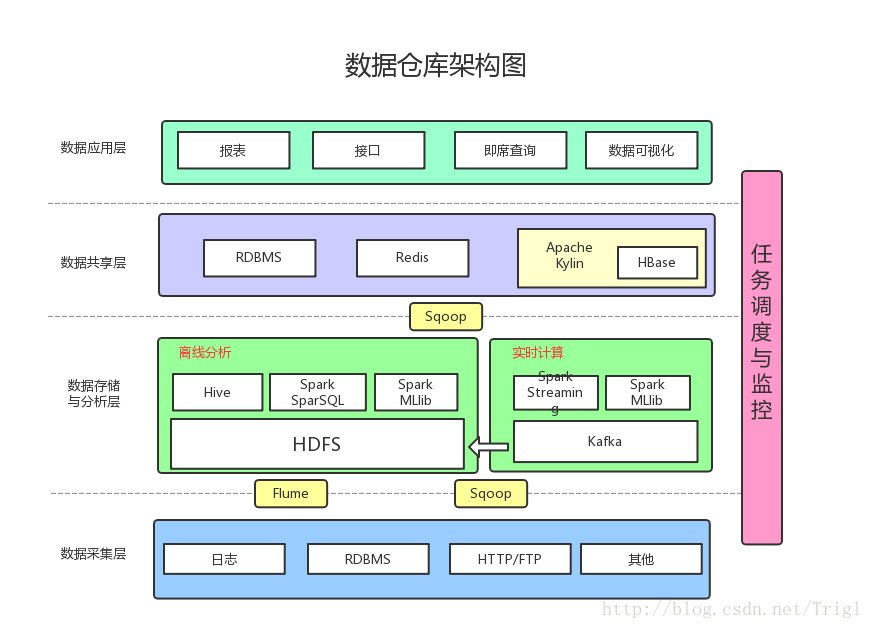
**数据挖掘**

数据挖掘正如在矿井中可以开采出珍贵的矿石，在数据仓库的数据里也常常可以开采出业务人员意想不到的信息。它比多维分析更进一步。例如，如果管理人员要求比较各个区域某类出入境特征数量在过去一年的情况，可以从多维分析中找答案。但是，如果管理人员要问为何一个地区的出入境特征情况突然变得特别好或是不好，或者问该出入境特征在另一地区将会怎么样，这时数据挖掘知识可以作出解答。

**数据仓库层**

数据仓库用于抽取、整合、分布、存储有用的信息，数据信息往往分布在不同的部门和下级单位，管理者要综观全局、运筹帷幄，必须能迅速地找到能反映真实情况的数据，这些数据也许是当前的现实数据，也可能是过去的历史数据。因此，有必要把各个区域的数据集合起来，去其糟粕、取其精华，将真实的、对决策有用的数据保留下来，随时准备管理人员使用。因此，数据仓库不仅仅是个数据的储存仓库，更重要的是它提供了丰富的工具来清洗、转换和从各地提取数据，使得放在仓库里的数据有条有理，易于使用。

# 数据仓库技术架构：



**数据采集**

数据采集层的任务就是把数据从各种数据源中采集和存储到数据存储上，期间有可能会做一些ETL操作。采用Flume收集日志，采用Sqoop将RDBMS以及NoSQL中的数据同步到HDFS上。

数据源种类可以有多种：

* 日志：所占份额最大，存储在备份服务器上
* 业务数据库：如Mysql、Oracle
* 来自HTTP/FTP的数据：合作伙伴提供的接口
* 其他数据源：如Excel等需要手工录入的数据

**数据存储与分析**

HDFS是大数据环境下数据仓库/数据平台最完美的数据存储解决方案。

离线数据分析与计算，也就是对实时性要求不高的部分，Hive是不错的选择。

使用Hadoop框架自然而然也提供了MapReduce接口，如果真的很乐意开发Java，或者对SQL不熟，那么也可以使用MapReduce来做分析与计算。

Spark性能比MapReduce好很多，同时使用SparkSQL操作Hive。

消息系统：可以加入Kafka防止数据丢失

实时计算：实时计算使用Spark Streaming消费Kafka中收集的日志数据，实时计算结果大多保存在Redis中

机器学习：使用了Spark MLlib提供的机器学习算法

**数据共享**

前面使用Hive、MR、Spark、SparkSQL分析和计算的结果，还是在HDFS上，但大多业务和应用不可能直接从HDFS上获取数据，那么就需要一个数据共享的地方，使得各业务和产品能方便的获取数据。

这里的数据共享，其实指的是前面数据分析与计算后的结果存放的地方，其实就是关系型数据库和NOSQL数据库。

多维分析OLAP：使用Kylin作为OLAP引擎

**数据应用**

报表：报表所使用的数据，一般也是已经统计汇总好的，存放于数据共享层。

接口：接口的数据都是直接查询数据共享层即可得到。

即席查询：即席查询通常是现有的报表和数据共享层的数据并不能满足需求，需要从数据存储层直接查询。一般都是通过直接操作SQL得到。

数据可视化：提供可视化前端页面，方便运营等非开发人员直接查询