# 概述

数据仓库，英文名称为Data Warehouse，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)。

1、面向主题的

数据仓库中的数据是按照一定的主题域进行组织。

主题是一个抽象的概念，是指用户使用数据仓库进行决策时所关心的重点方面，一个主题通常与多个操作型信息系统相关。而操作型数据库的数据组织面向事务处理任务，各个业务系统之间各自分离。

2、集成的

数据仓库中的数据是在对原有分散的数据库数据抽取、清理的基础上经过系统加工、汇总和整理得到的，必须消除源数据中的不一致性，以保证数据仓库内的信息是关于整个企业的一致的全局信息。面向事务处理的操作型数据库通常与某些特定的应用相关，数据库之间相互独立，并且往往是异构的。

3、相对稳定的

数据仓库的数据主要供企业决策分析之用，所涉及的数据操作主要是数据查询，一旦某个数据进入数据仓库以后，一般情况下将被长期保留，也就是数据仓库中一般有大量的查询操作，但修改和删除操作很少，通常只需要定期的加载、刷新。操作型数据库中的数据通常实时更新，数据根据需要及时发生变化。

4、反映历史变化

数据仓库中的数据通常包含历史信息，系统记录了企业从过去某一时点(如开始应用数据仓库的时点)到目前的各个阶段的信息，通过这些信息，可以对企业的发展历程和未来趋势做出定量分析和预测。而操作型数据库主要关心当前某一个时间段内的数据。

## 数据湖 vs 数据仓库



## 核心概念

### ETL

ETL，是英文Extract-Transform-Load的缩写，用来描述将数据从来源端经过抽取（extract）、转换（transform）、加载（load）至目的端的过程，是数据仓库的生命线。

抽取（Extract）主要是针对各个业务系统及不同服务器的分散数据，充分理解数据定义后，规划需要的数据源及数据定义，制定可操作的数据源，制定增量抽取和缓慢渐变的规则。

转换（transform）主要是针对数据仓库建立的模型，通过一系列的转换来实现将数据从业务模型到分析模型，通过ETL工具可视化拖拽操作可以直接使用标准的内置代码片段功能、自定义脚本、函数、存储过程以及其他的扩展方式，实现了各种复杂的转换，并且支持自动分析日志，清楚的监控数据转换的状态并优化分析模型。

装载（Load）主要是将经过转换的数据装载到数据仓库里面，可以通过直连数据库的方式来进行数据装载，可以充分体现高效性。在应用的时候可以随时调整数据抽取工作的运行方式，可以灵活的集成到其他管理系统中。

### ETL & ELT

伴随着数据仓库的发展（传送门：数据仓库的八个发展阶段），数据量从小到大，数据实时性从T+1到准实时、实时，ETL也在不断演进。

在传统数仓中，数据量小，计算逻辑相对简单，我们可以直接用ETL工具实现数据转换（T），转换之后再加载到目标库，即（Extract-Transform-Load）。但在大数据场景下，数据量越大越大，计算逻辑愈发复杂，数据清洗需放在运算能力更强的分布式计算引擎中完成，ETL也就变成了ELT（Extract-Load-Transform）。

即：Extract-Transform-Load >> Extract-Load-Transform

通常我们所说的ETL，已经泛指数据同步、数据清洗全过程，而不仅限于数据的抽取-转换-加载。

# 分类

我们可以从传统型、大数据型、云原生型、以及开源型角度来分类：

## 传统型数据仓库工具



### Oracle Exadata

### Teradata

### IBM DB2 Warehouse

### SAP BW

### Microsoft SQL Server

## 云原生数据仓库



### Snowflake

### Amazon Redshift

### Google BigQuery

### Databricks

## 大数据/开源型数据仓库



### Hive

### Hudi

### Iceberg

### Delta Lake

### ClickHouse

### StarRocks

### Trino

### Greenpulm

## 对比



## 适用场景



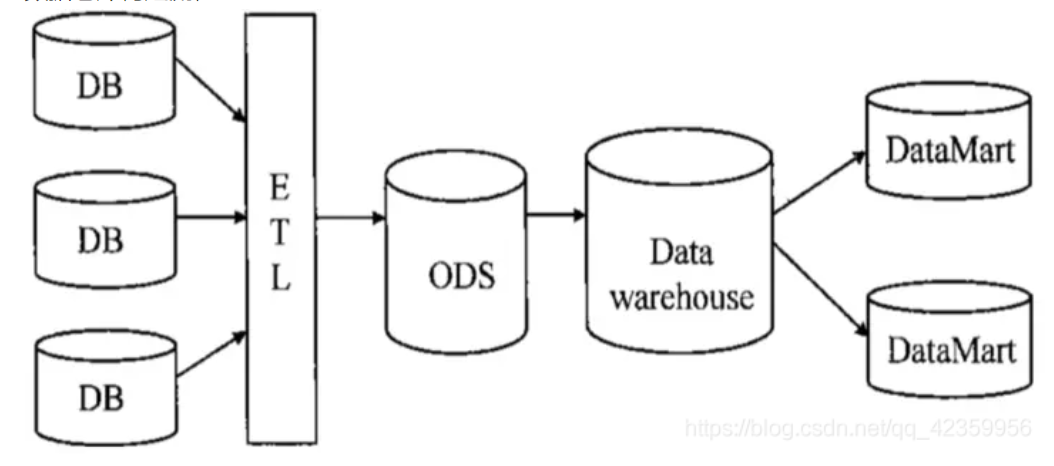
总结

数据仓库工具种类繁多，已从早期封闭架构演变为开放式、云原生、大数据友好型架构。

对于现代企业，通常选择“数据湖 + 开源数仓引擎”或“云原生数据仓库”结合的方式，实现高性能与低成本的平衡。

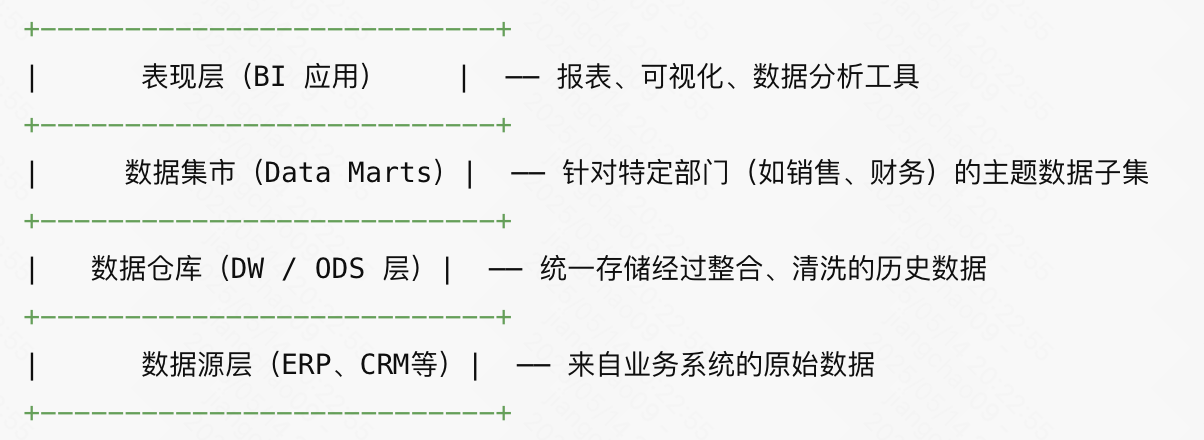
未来趋势是存算分离、实时与批处理融合、统一元数据管理。

# 架构

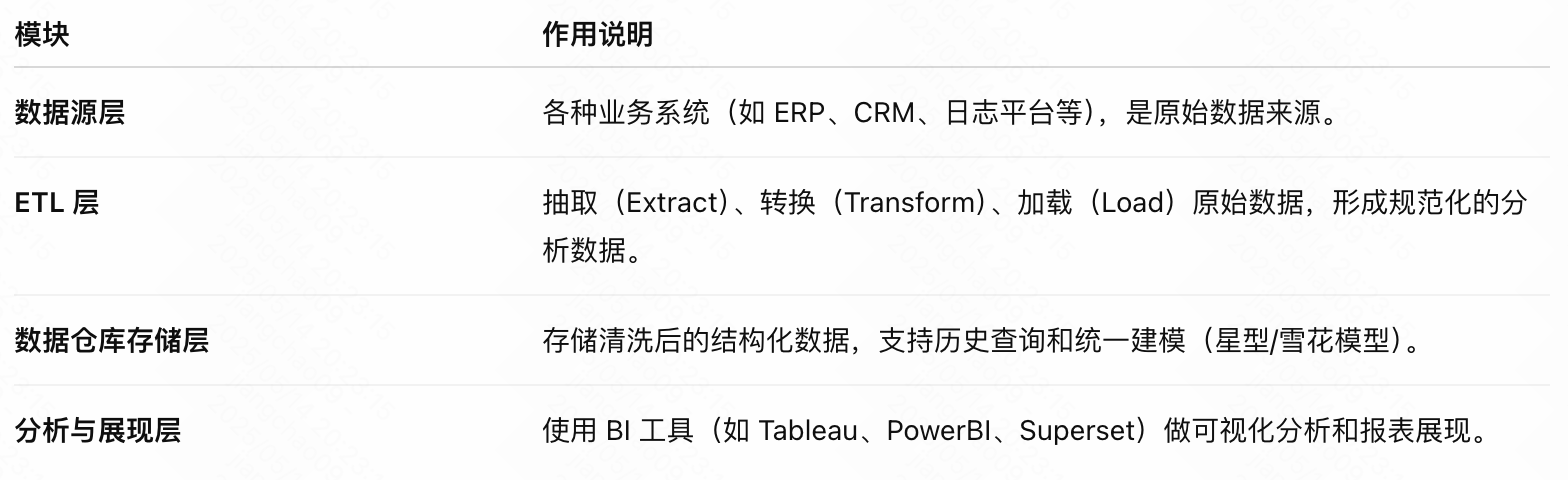


## 数据仓库的架构：经典三层结构（Inmon 模型）

数据仓库的传统架构通常分为三层（或四层）结构：



## 核心模块



### 数据源层

### ETL层

### 数据仓库存储层

### 分析和展现层

## 基本原理

### 数据建模

维度建模：包括星型模型、雪花模型，用于构建事实表（记录业务事件）和维度表（描述事件属性）。

主题域划分：以“主题”为中心组织数据（如客户、销售、订单），便于业务分析。

### ETL 流程

Extract（抽取）：从多个异构系统中提取数据，如数据库、文件、API、Kafka 等。

Transform（转换）：清洗数据（去重、规范格式）、构建主键、维度映射、业务规则处理。

Load（加载）：将数据加载到数据仓库（如 Hive、Snowflake、Redshift）中，按主题和时间组织。

### OLAP（联机分析处理）

数据仓库通常支持 OLAP 查询，具备以下特点：



## 现代数据仓库架构演化（Lakehouse架构）

随着数据量激增，传统数据仓库存在扩展性差、实时性弱、结构刚性强等问题，出现了以下现代趋势：



例如：

Snowflake、BigQuery、Redshift 采用云存算分离

Databricks Lakehouse 构建在 Delta Lake + Spark 上，融合数据湖与数仓能力

## 典型数据仓库架构图

