教育数据中心服务指南

上海容慧信息科技有限公司

目录

1. 概述.....6
2. 技术架构.....6
3. 数据存储架构.....8
4. 数据计算框架.....8
4.1. 离线计算组件.....9
4.2. 消息中间件.....12
4.3. 实时计算组件.....13
5. 数据汇聚.....15
5.1. ELT 汇聚过程.....16
5.2. 数据梳理.....16
5.3. 异构数据接入.....17
5.4. 数据抽取方式.....17
5.5. 数据汇聚调度.....22
5.6. 数据加载.....22
5.7. 数据融合集成平台.....23
6. 数据资产管理.....24
6.1. 数据资产管理的目标.....24
6.2. 数据资产管理在数据中心架构中的位置.....25
6.3. 数据治理.....26
6.3.1. 数据预处理.....26
6.3.2. 数据清洗.....26
6.3.3. 数据加工转换.....29
6.3.4. 数据治理平台.....30
6.4. 数据标准管理.....33
6.4.1. 标准管理.....34
6.4.2. 维度标准.....34
6.5. 元数据管理.....34

6.5.1. 元数据的采集.....35
6.5.2. 元数据的管理.....37
6.5.3. 元数据的应用.....38
6.5.4. 功能模块设置.....39
6.6. 数据质量管理.....41
6.6.1. 数据质量要求.....42
6.6.2. 数据质量管理的流程.....43
6.6.3. 数据质量管理实施.....44
6.6.4. 数据质量监控.....45
6.7. 生命周期管理.....45
6.8. 数据资产地图.....46
6.9. 数据资产编目.....47
6.9.1. 分层管理.....49
6.9.2. 可视化建表.....50
6.9.3. 编目管理.....51
6.9.4. 表元数据管理.....52
6.9.5. 查看资源目录.....52
6.10. 业务数据可视化.....52
6.11. 监管数据可视化.....52
7. 数据体系设计.....53
7.1. 数据体系特征.....53
7.2. 数据体系架构.....54
8. 贴源数据层创建.....55
8.1. 贴源数据库表设计.....55
9. 统一数仓层建设.....56
9.1. 统一数仓层建设过程.....57
9.1.1. 数据域划分.....60
9.1.2. 指标设计.....61

9.1.3. 维度表设计.....61
9.1.4. 事实表设计.....62
9.1.5. 模型落地实现.....64
9.2. 基础数据层数据库.....64
9.3. 主题数据层数据库.....65
9.3.1. 学生成长主题库.....65
9.3.2. 教师发展主题库.....66
9.3.3. 教学管理主题库.....66
9.3.4. 综合管理主题库.....66
9.3.5. 学校发展主题库.....66
9.4. 主题数据层数据结构设计.....67
9.5. 主题数据层模型设计.....67
10. 标签数据层建设.....68
11. 应用数据层建设.....69
11.1. 应用数据层的概念.....70
11.2. 应用数据表设计.....70
11.3. 应用数据表实现步骤.....71
11.4. 应用数据的存储策略.....71
12. 数据开放服务（共享及交换）系统.....72
12.1. 数据服务类型.....73
12.2. 数据服务专题管理.....74
12.2.1. 专题生成.....74
12.2.2. API 服务.....75
12.2.3. 库表推送.....76
12.2.4. 离线文件.....76
12.3. 数据服务发布管理.....78
12.3.1. 服务审核.....78
12.3.2. 服务监控.....80

12.3.3. 流量管控.....81
13. 学生成长画像.....82
13.1. 学生个体数字画像.....82
13.2. 学生群体数字画像.....84

1. 概述

基于数据标准规范，提供教育数据从采集、存储、加工、分析、共享及可视化的全过程能力。针对区、校二级用户，提供独立的云上数据空间，用于存储各类数字化应用的业务数据及经标准化加工处理后的数据，并通过数据基座实现跨数据空间互联，实现教育数据在区、校之间的共享。通过可视化的呈现方式，为区、校管理平台提供基础数据、应用使用情况等数据的汇总概览。

2. 技术架构

区教育数据中心将实现数据集成、数据处理、数据服务、数据分析、数据交换等功能，服务全区学校、师生以及教育管理单位或部门。云计算、大数据技术是当前数据中心构建的主流技术，是保障数据中心高性能、高可用性的关键技术。教育数据中心构建的主要技术包括云计算、分布式数据库、大数据处理技术、分布式计算、数据资产管理、数据可视化等技术。

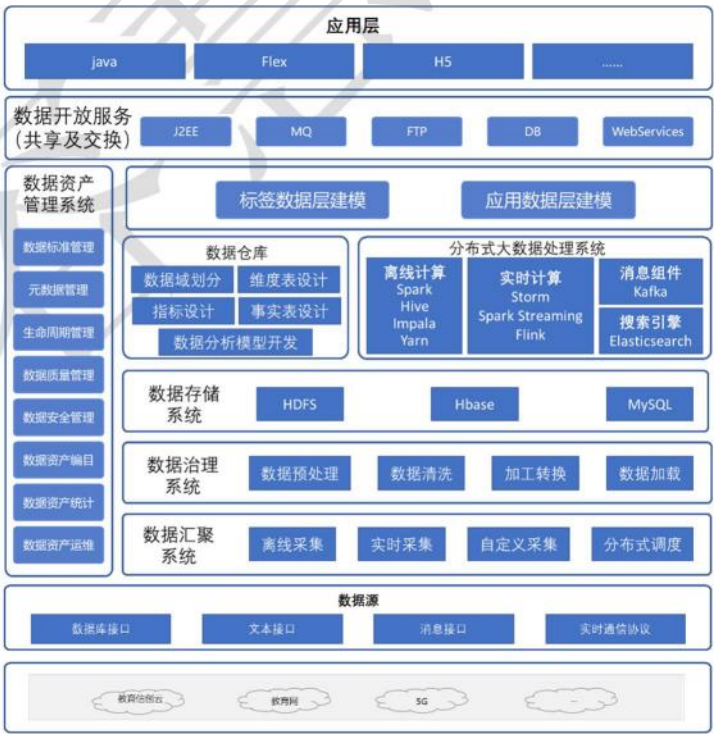

图 2-1 区教育数据中心技术架构图

云计算

数据中心将以区教育云平台为基础来构建，利用云计算技术，以降低硬件成本，提高资源利用率，同时实现数据中心的高可用性。

分布式数据存储技术

全区海量的教育业务数据中不仅仅包括结构化数据，还有大量的非结构化、半结构化数据。单台服务器已经无法满足数据存储和处理的需求，分布式存储是解决海量数据存储，并支持大数据处理的主要技术。分布式存储技术可以实现结构化、非结构、半机构化数据的分布式存储。目前大数据存储和处理技术中，以Hadoop为代表的分布式文件系统以存储和处理非结构化和半结构化数据具有优势，结构化数据的存储和处理采用关系型数据库。

分布式大数据处理系统

分布式大数据处理系统支持数据的离线批量处理、在线实时处理、消息服务等。其中融合数仓场景下，用分布式文件系统实现半结构化、非结构化数据处理；用关系型数据库处理PB级别的、高质量的结构化数据，上层数据的维度汇总以及深度分析同样采用关系型数据库，同时为应用提供丰富的SQL和事物支持能力。这样可以同时满足结构化、半结构化和非结构化数据的高效处理需求。

数据可视化技术

数据可视化技术主要旨在借助于图形化手段，清晰有效地传达与沟通信息。数据可视化满足以下需求：

（1）交互性：可视化分析是获取数据、单向表示数据、注意结果和提出后续问题的过程。后续问题可能需要向下钻取、向上钻取、筛选、引入新数据或创建数据的其他视图。如果没有交互活动，则无法通过分析解答提出的问题。通过适当的交互活动，数据可视化成为了分析师思维过程的自然延伸。

（2）多维性：数据可视化必须足够灵活以便能够说明各种问题，为了让数据以最佳可视化效果呈现，通常要综合考虑多个方面：多维性体现在对象或事件

数据的多个属性或变量，而数据可以按其每一维的值，将其分类、排序、组合和显示。

（3）可视性：数据可以用图像、曲线、二维图形、三维体和动画来显示，并可对其模式和相互关系进行可视化分析。

3. 数据存储架构

全区海量的教育业务数据中不仅仅包括结构化数据，还有大量的非结构化、半结构化数据。单台服务器已经无法满足数据存储和处理的需求，分布式存储是解决海量数据存储，并支持大数据处理的主要技术。分布式存储技术可以实现结构化、非结构、半机构化数据的分布式存储。目前大数据存储和处理技术中，以Hadoop为代表的分布式文件系统以存储和处理非结构化和半结构化数据具有优势，而关系型数据库对海量结构化数据库的存储和处理具有较高的效率。因此，关系型数据库和Hadoop混合架构是未来海量数据处理发展趋势。用关系型数据库处理PB级结构化数据存储与查询，提供完整的SQL与事务支持功能。用Hadoop处理半结构化、非结构化数据，提供灵活的自定义模型与算法开发能力，同时满足多种数据类型处理需求，并在实时查询与离线分析上都能提供较高性能。

4. 数据计算框架

数据计算主要面向开发人员、分析人员，提供离线、实时、算法的开发工具，以及任务的管理、代码发布、运维、监控、告警等一系列服务，方便使用，提升效率。在这个过程中，通过数据计算套件对大数据的存储和计算能力进行封装，通过产品化的方式让用户更容易地使用大数据。

数据计算套件主要包括三个部分，分别是离线计算、实时计算和消息中间件。

4.1.离线计算组件

大数据离线计算，就是利用大数据的技术栈（主要是Hadoop），在计算开始前准备好所有输入数据，该输入数据不会产生变化，且在解决一个问题后就要立即得到计算结果的计算模式。离线（offline）计算也可以理解为批处理（batch）计算，与其相对应的是在线（online）计算或实时ealtime）计算。

离线计算套件封装了大数据相关的技术，包括数据加工、数据分析、在线查询、即席分析等能力，同时也将任务的调度、发布、运维、监控、告警等进行整合，让开发者可以直接通过浏览器访问，不再需要安装任何服务，也不用关心底层技术的实现，只需专注于业务的开发，帮助用户快速构建数据服务，赋能业务。

1. 离线计算的特点

(1)数据量巨大，保存时间长。

(2)在大量数据上进行复杂的批量运算。

(3)数据在计算之前已经完全到位，不会发生变化。

(4)能够方便地查询计算结果。

2. 离线计算应用场景

（1）大数据离线计算主要用于数据分析、数据挖掘等领域。

（2）BI（全称为Business Intelligence，即商业智能）系统能够辅助业务经营决策。其需要综合利用数据仓库（基于关系型数据库）、联机分析处理（OLAP）工具（如各种SQL）和数据挖掘等技术。

3. 大数据离线计算涉及组件

(1) HDFS 分布式文件系统

HDFS是Hadoop上的分布式文件系统。HDFS采用主从模式，其架构主要包含NameNode，DataNode，Client三个部分：

NameNode:用于存储、生成文件系统的元数据。运行一个实例。

DataNode：用于存储实际的数据，并将自己管理的数据块信息上报给NameNode，运行多个实例。一个数据默认存储3副本，分布在3个不同的DataNode上以保证可用性。

Client：支持使用者读写HDFS，从NameNode、DataNode获取元数据或实际数据返回给使用者。可以有多个实例，和业务一起运行。

(2) MapReduce 并行计算框架

Map：映射，对一些独立元素组成的列表的每一个元素进行指定的操作。每个元素都是被独立操作的，而原始列表没有被更改。Map操作是可以高度并行的，这对高性能应用以及并行计算领域的需求非常有用。

Reduce：化简，对一个列表的元素进行适当的合并，虽然它不如Map那么并行，但是大规模的运算相对独立，所以化简函数在高度并行环境下也很有用。

适合：大规模数据集的离线批处理计算；任务分而治之，子任务相对独立。

不适合：实时的交互式计算，要求快速响应和低延迟；流式计算、实时分析；子任务之间相互依赖的迭代计算。

(3) Yarn 资源管理系统

Yarn是一个通用的资源管理模块，可为各类应用程序进行资源管理和调度。

Yarn是轻量级弹性计算平台，除了MapReduce框架，还可以支持其他框架，比如Spark、Storm等。

Yarn实现多种框架统一管理，共享集群资源，资源利用率高，运维成本低，数据共享方便。

(4) Spark

Spark是专为大规模数据处理而设计的快速通用的计算引擎。

Spark 的特点：

- 高级 API 剥离了对集群本身的关注，Spark 应用开发者可以专注于应用所要做的计算本身。
- Spark 很快，支持交互式计算和复杂算法，内存计算下，Spark 比 Hadoop 快 100 倍。
- Spark 是一个通用引擎，可用它来完成各种各样的运算，包括 SQL 查询、文本处理、机器学习等。
- 易用性，Spark 提供了 80 多个高级运算符。
- 通用性，Spark 提供了大量的库，包括 Spark Core、Spark SQL、Spark Streaming、MLlib、GraphX。 开发者可以在同一个应用程序中无缝组合使用这些库。
- 兼支持多种资源管理器，兼容 HDFS、Hbase 等分布式存储，可运行在 YARN 之中，作为 MapReduce 的有益补充。

(5) Hive

Hive是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供类SQL的查询功能。其基本原理是将HQL语句转换成MapReduce任务。

(6) Impala

Impala是一个MPP架构的大数据分析引擎，提供交互式、即时SQL查询能力。其大多数语法和Hive相同或类似，元数据也使用Hive的，因此可以直接操作Hive表。

(7) Hue

Hue（Hadoop User Experience）是一个开源的Apache Hadoop UI系统。

通过 Hue 可以：

- SQL 编辑，支持 Hive、Impala、SparkSQL 和各种关系型数据库。

- 管理 Hadoop 和 Spark 任务。
- HBase 数据的查询和修改。
- Sqoop 任务的开发和调试。
- 管理 Oozie 的工作流。

4.2.消息中间件

消息中间件是利用高效可靠的消息传递机制进行异步的数据传输，并基于数据通信进行分布式系统的集成。通过提供消息队列模型和消息传递机制，可以在分布式环境下扩展进程间的通信。开发人员不需要考虑网络协议和远程调用的问题，只需要通过各消息中间件所提供的api，就可以简单的完成消息推送，和消息接收的业务功能。

消息的生产者将消息存储到队列中，消息的消费者不一定马上消费消息，可以等到自己想要用到这个消息的时候，再从相应的队列中去获取消息。这样的设计可以很好的解决，大数据量数据传递所占用的资源，使数据传递和平台分开，不再需要分资源用于数据传输，可以将这些资源用去其他想要做的事情上。

消息中间件一般有两种传递模式：

1.点对点模式：消息生产者将消息发送到队列中，消息消费者从队列中接收消息。消息可以在队列中进行异步传输。

2.发布/订阅模式：发布订阅模式是通过一个内容节点来发布和订阅消息，这个内容节点称为主题（topic），消息发布者将消息发布到某个主题，消息订阅者订阅这个主题的消息，主题相当于一个中介。主题是的消息的发布与订阅相互独立，不需要进行基础即可保证消息的传递，发布/订阅模式在消息的一对多广播是采用。

Kafka：

Kafka是一个高吞吐、分布式、基于发布订阅的消息系统，同时支持离线和在线日志处理。利用Kafka技术可以在廉价的PC Server上搭建起大规模消息系统。目前越来越多的开源分布式处理系统如Cloudera、Apache Storm、Spark、Flink都支持与Kafka集成。

Kafka和其他组件比较，具有消息持久化、高吞吐、分布式、多客户端支持、实时等特性，适用于离线和在线的消息消费，如常规的消息收集、网站活性跟踪、聚合统计系统运营数据（监控数据）、日志收集等大量数据的互联网服务的数据收集场景。

Kafka目前已成为大数据系统在异步和分布式消息之间的最佳选择。

具体应用场景如下：

- 日志收集：用 Kafka 可以收集各种服务的 log，通过 kafka 以统一接口服务的方式开放给各种用户，例如 Hadoop、Hbase、Solr 等；
- 消息系统：解耦和生产者和消费者、缓存消息等；
- 用户活动跟踪：Kafka 经常被用来记录 web 用户或者 app 用户的各种活动，如浏览网页、搜索、点击等活动，这些活动信息被各个服务器发布到 kafka 的 topic 中，然后订阅者通过订阅这些 topic 来做实时的监控分析，或者装载到 Hadoop、数据仓库中做离线分析和挖掘；
- 运营指标：Kafka 也经常用来记录运营监控数据。包括收集各种分布式应用的数据，生产各种操作的集中反馈，比如报警和报告；
- 流式处理：比如 spark streaming 和 storm；
- 事件源。

4.3.实时计算组件

实时开发组件是对流计算能力的产品封装。实时计算具备以下三大特点：

• 实时且无界（unbounded）的数据流：实时计算面对的计算是实时的、流式的，流数据是按照时间发生的顺序被实时计算订阅和消费的。并且，由于数据产生的持续性，数据流将长久且持续地集成到实时计算系统中。

• 持续且高效的计算：实时计算是一种“事件触发”的计算模式，触发源就是无界流式数据。一旦有新的流数据进入实时计算，实时计算立刻发起并进行一次计算任务，因此整个实时计算是持续进行的高效计算。

• 流式且实时的数据集成：流数据触发一次实时计算的计算结果，可以被直接写入目的存储中，例如，将计算后的报表数据直接写入 MySQL 进行报表展示。因此，流数据的计算结果可以类似流式数据一样持续写入目的存储中。

基于Storm、SparkStreaming、ApacheFlink构建的一站式、高性能实时大数据处理能力，广泛适用于实时ETL、实时报表、监控预警、在线系统等多种场景。让用户彻底规避繁重的底层流式处理逻辑开发工作。

(1) Storm

Storm是一个分布式的、容错的实时计算系统，除了用于实时分析外，它还可以用于在线数据挖掘、机器学习、ETL、分布式远程调用等领域。Storm为分布式实时计算提供了一组通用原语。在计算时将结果以流的形式输出给用户，从而进一步提升实时性。

(2) Spark Streaming

Spark Streaming是Spark核心API的一个扩展，一个实时计算框架。具有可扩展性、高吞吐量、可容错性等特定。是一个针对实时数据进行快速流式处理的系统。

(3) ApacheFlink

Flink是一款分布式、高性能、高可用、高精确的为数据流应用而生的开源流式处理框架。Flink的核心是在数据流上提供数据分发、通信、具备容错的分布式计算。同时，Flink在流处理引擎上提供了批流融合计算能力，以及SQL表达能力。

Flink以下特点：

- 低延时，提供 ms 级时延的处理能力。
- 异步快照机制，，保证所有数据真正处理一次。
- 支持主备模式，保证无单点故障。
- 支持手动水平扩展。
- Flink 能够支持 Yarn，能够从 HDFS 和 HBase 中获取数据。
- 能够使用所有的 Hadoop 的格式化输入和输出。
- 能够使用 Hadoop 原有的 Mappers 和 Reducers，并且能与 FLink 的操作混合使用。
- 能够更快的运行 Hadoop 作业。

5. 数据汇聚

数据汇聚是数据中心建设的重要环节，其扩展能力和多协议适配能力决定数据中心的接入稳定性与扩展性。它既可以按照请求，主动进行数据采集；也可以按照通知，被动感知数据增量的变化和更新，再进行采集，并能将分散的业务数据链接成可互通共享的跨云、跨平台分布式“数据湖”。

数据融合集成平台是数据中心主体架构中负责采集汇聚各应用系统的基础数据、业务数据的工具，具备多协议和分布式数据的采集接入能力、任务编排能力、数据融合及轻度加工能力等，以支持完成多协议数据的采集和对接，实时或离线的结构化和非结构化数据入库。

5.1.ELT 汇聚过程

在数据建设过程中有ETL(Extract-Transform-Load,抽取－转换－加载）的操作，即在数据抽取过程中进行数据的加工转换，然后加载至存储中。但在大规模数据场景下，不建议采用ETL的方式，而是采用ELT(Extract-Load-Transform，抽取－加载－转换）的模式，即将数据抽取后直接加载到存储中，再通过大数据和人工智能相关技术对数据进行清洗和处理。如果用ETL的模式在传输过程中进行复杂的清洗，会因为数据体量过大和清洗逻辑的复杂性导致数据传输的效率大大降低。另一万面ETL模式在清洗过程中只提取有价值的信息进行加载，而是否付有价值是基于当前对数据的认知来判断的，由于数据价值会随有我们对数据的认知以及数据智能相关技术的发展而不断被挖掘，此ETL模式很容易出现一此有价值的数据被清洗掉，导致当某一天需要用这些数据时，又需要重新处理，甚至数据丢失午发找回。相比存储的成本，这种损失可能会更大。

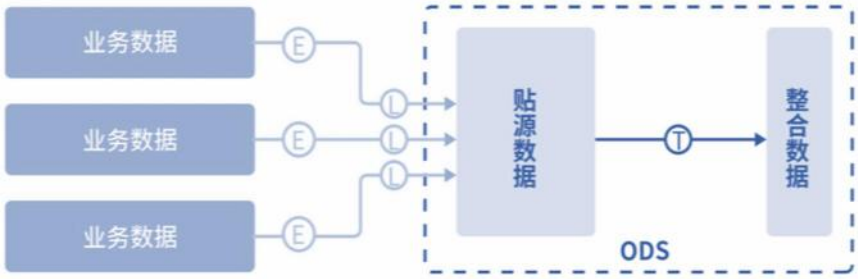

图 5-1 数据中心数据抽取转换过程

5.2.数据梳理

在数据汇聚之前需要对各应用系统的数据进行梳理，这是数据采集前的重要的环节，包括采集的标准、方法、采集的数据的属性等等一系列的数据。通过人工的梳理，分别对应用系统数据进行系统调查，根据各系统的数据字典，对各个业务应用系统进行深入分析，摸清现有系统数据库表结构，梳理出数据库表结构的关键字段有哪些，不同数据库表字段之间的关联关系，是后续数据抽取、装载、转换、汇总，建设标准数据主题库的前置条件和工作。

5.3.异构数据接入

数据源可以是已有系统存储业务数据的地方，作为数据中心的数据来源，也可以是数据应用场景，为应用场景提供结果数据存储的地方。

数据融合集成平台提供了多源异构的数据接入，并可对接主流大数据平台的能力，灵活的将数据存储到大数据平台中。支持关系型数据库、MPP数据库、CSV文件的多版本数据采集，其中关系型数据库包括Mysql、Oracle、DB2、SQL server、Postgres、Stork，MPP数据库包括Greenplum、Teryx、分布式数据库Hive以及国产MPP数据库。

5.4.数据抽取方式

按照数据结构类型的不同，教育业务数据可以分为三类：

•结构化数据：主要是关系型数据库中的数据，直接从业务系统DB抽取到贴源数据层。

• 半结构化数据：一般是纯文本数据，以各种日志数据为主，半结构化数据保留贴源数据的同时也做结构化处理，为后续使用做准备。

• 非结构化数据：主要是图片、音频、视频，一般保留在文件系统中，由于这类数据量一般比较庞大，而且没有太多挖掘分析价值，所以贴源数据层不保留原始文件，只保留对原始数据文件的描述，比如地址、名称、类型、分辨率等。

数据汇聚支持离线、实时两种抽取方式，实现数据的全量、增量同步，并支持覆盖、追加、更新三种入库策略。同时提供批量创建采集任务的快捷模式，能快速实现多表甚至整库的数据迁移，节省大量时间和人力成本。

(1) 离线采集

离线采集支持Mysql、Oracle、DB2、SQL server、Postgres、Greenplum、Teryx、Hive数据源，并支持可视化向导模式和自定义脚本模式两种任务配置方式。离线采集分为单表采集、批量采集、落地采集和自定义采集四种模式，同步方式灵活

多样。根据设置的采集策略（全量/增量）及数据入库策略（覆盖/追加），结合配置的调度策略，周期性的自动采集（也支持手动执行）。

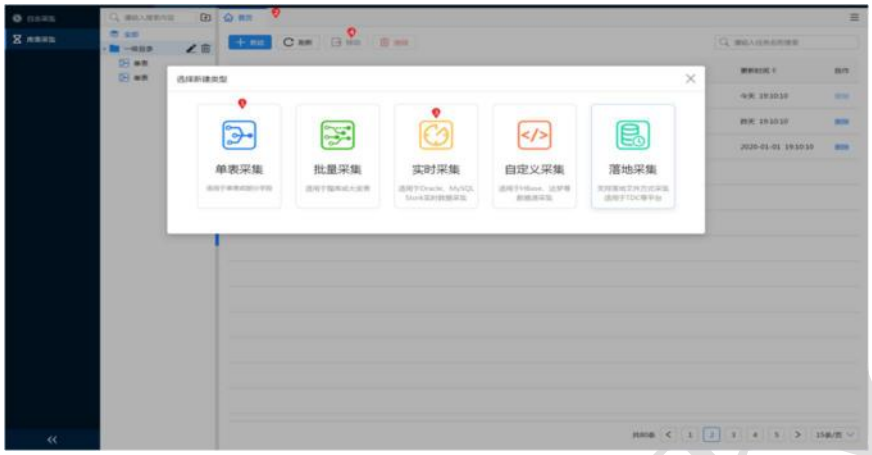

图 5-2 数据采集-库表采集

单表采集：支持对数据源进行物理表、视图字段级别抽取操作，允许抽取脏数据，提供文件类型配置项参数，可进行全量、增量的采集方式，存储数据方式支持覆盖、追加，能满足灵活的业务场景。此外，还可以查看向导式设置的单表采集任务代码，并支持转化为代码编辑模式。

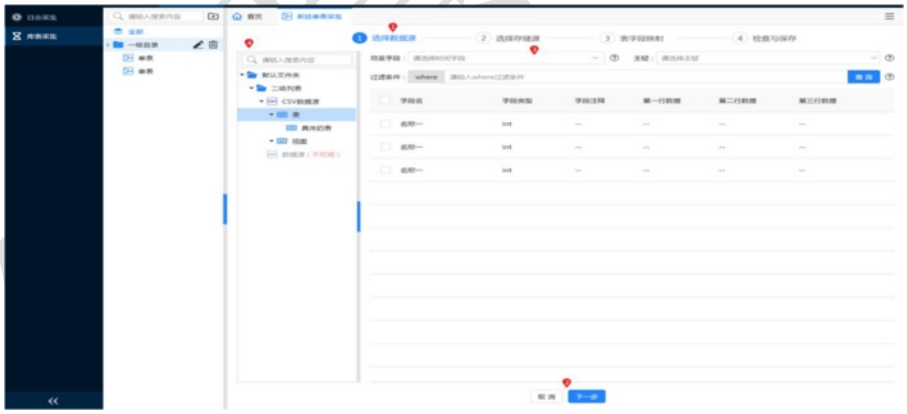

图 5-3 数据采集-库表采集-单表采集

批量采集：通过图形化的配置实现了对源库表信息的批量导入，同时提供了一键配置功能，对操作流程进行精简。针对新建目标表同名冲突情况，提供了忽略、覆盖、追加三种处理方案以及批量设置方式。支持查看代码。支持过滤空表采集的初步清洗策略。

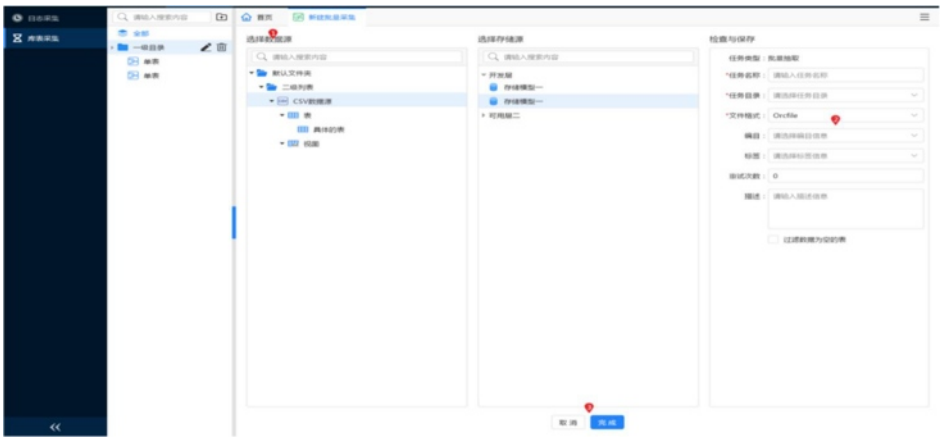

图 5-4 数据采集-库表采集-批量采集

落地采集：可视化抽取方式与原生态HDFS抽取方式的结合应用，能够满足包括TDC平台在内的大多数大数据场景下的抽取需求。

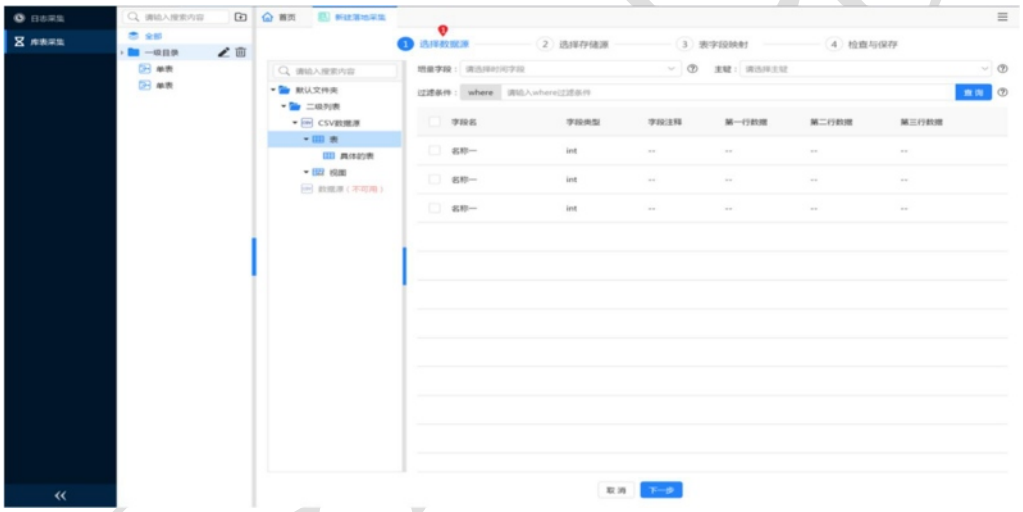

图 5-5 数据采集-库表采集-落地采集

自定义采集：提供代码化的采集环境，更专业和灵敏化配置，适用于大数据量低清洗度的数据集成，支持用户更多的特殊业务抽取场景，并且支持单表和批量抽取转化为自定义采集任务，进行代码化编辑。

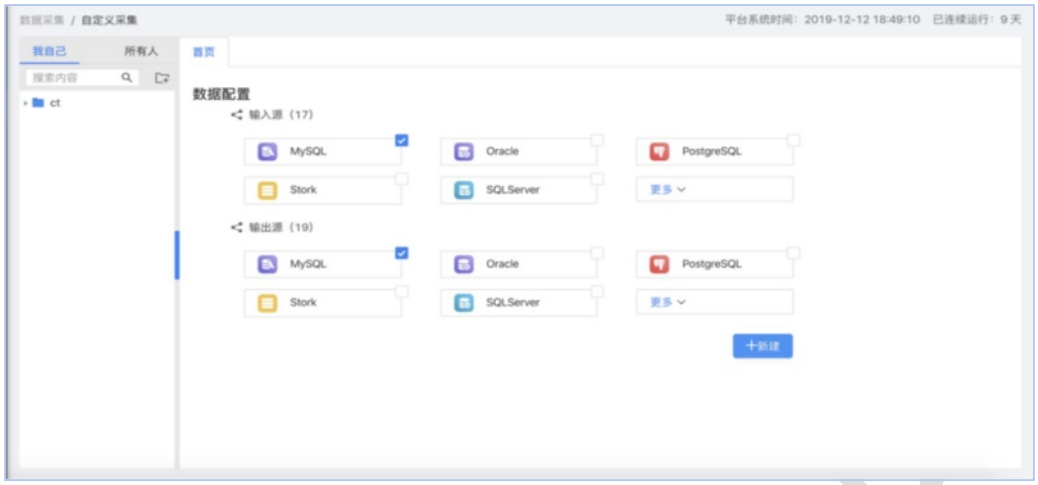

图 5-6 数据采集-库表采集-自定义采集

(2) 实时采集

实时采集是常驻运行任务，实时采集数据源支持关系型数据库和日志文件的实时采集。

- 关系型数据库

通过监控数据库归档日志的模式实现Oracle、Mysql、Postgresql实时同步数据，同时支持实时数据的落地存储，同时也可将实时采集的数据用于实时计算、分析或实时共享。实时采集支持以hive作为存储源的历史数仓和以Stork/Teryx作为存储的实时数仓建设。

图 5-7 数据采集-库表采集-实时采集

- 日志文件

通过在日志源端服务器部署一个日志采集组件，能实现对指定.csv、.txt、.log格式的日志文件实时采集，并可将数据进行离线归档至HDFS。日志组件能实现多种监控方式：

a. 监控全目录：能监控指定目录下的日志文件，对于新增的文件也能同步

b.根据目录规则监控：通过设定文件名的规则，能采集该目录下符合条件的文件，对于新增的符合条件文件，也能自动同步。

c.监控单个文件：采集指定文件的数据，对于该文件后续增加的内容也能被同步过来。

d.根据文件规则监控：能采集符合设定规则的文件，对于任务启动时符合规则的文件，后续内容有更新也会自动同步。

图 5-8 数据采集—日志采集

自定义采集：提供代码化的采集环境，集成Flume引擎，更专业和灵敏化配置，支持用户更多的特殊业务抽取场景。

5.5.数据汇聚调度

数据融合集成平台提供单次执行和周期执行两种调度策略，实现任务按分钟、小时、天、周、月的灵活调度。对于任务的运行节点能自动实现负载均衡，任务运行超时也有相应的中断处理机制确保因某些任务长期消耗系统资源而影响其他任务的正常运行。对于失败的任务，平台也提供自动重试机制，能保证因网络波动等异常导致任务失败的自动恢复。对于增量采集的离线任务，平台也提供自动补采数据的能力，保证某一周期任务失败，下一周期自动补采失败周期数据，从而确保数据的不丢失。同时平台提供丰富的任务监控及运维的能力。

5.6.数据加载

在大规模数据场景下，采用ELT(Extract-Load-Transform，抽取－加载－转换）的数据建设模式，即将数据抽取后直接加载到存储中，再通过大数据和人工智能相关技术对数据进行清洗和处理。

在完成原始应用系统数据的接入后，数据融合平台将抽取到的数据加载到存储中。

装载数据的最佳方法取决于所执行操装载作的类型以及需要装入多少数据。当目的库是关系数据库时，一般来说有两种装载方式：

（1）直接SQL语句进行insert、update、delete操作。

（2）采用批量装载方法、sqlldr等。

大多数情况下使用第一种方法，因为它们进行了日志记录并且是可恢复的。但是，批量装载操作易于使用，并且在装入大量数据时效率较高。使用哪种数据装载方法取决于业务系统的需要。

本项目数据装载的几个场景：

（1）初始装载：需要一次性对数据源中所有表进行迁移，迁移到目的库，也就是相应的数据主题库中。

（2）增量装载：根据变化需要定期将数据源中的表迁移到目的数据主题库中，对数据主题库进行更新。

（3）完全刷新：完全删除数据主题库中的一个表或者多个表，然后重新装载新的数据。

5.7.数据融合集成平台

数据融合集成平台是实现高效数据汇聚的工具，不仅支持数据库方式进行批量采集，也支持对通过api接口进行数据采集，最终实现各个来源的异构数据跨域汇聚。同时系统对所有采集任务提供统一的调度机制与监控机制，在任务异常时能够及时告警，为整个采集过程提供了可靠、可观、可控等操作保障，最终实现各个来源的异构数据跨域汇聚。

平台包括统一数据调度平台和管理后台，调度平台主要负责对于业务作业的展示、调度、日志监控、作业监控等功能，管理后台主要负责业务作业、数据输入、数据清洗计算、数据更新插入等作业的配置开发。

对于无法通过采集系统直接采集的数据，可以通过数据直报平台，以统一的数据模板上传数据。

数据融合阶段，同时支持与存放DW的数据库系统相同或不同的数据源的接入，同时考虑数据库、API与不同文件类型数据源（.txt,xls）的数据的抽取。

最后，对不一致数据进行转换统一，将业务系统数据按照数据仓库粒度进行聚合，并根据不同的业务规则、数据指标进行计算并存储在数据仓库中，以便分析使用。

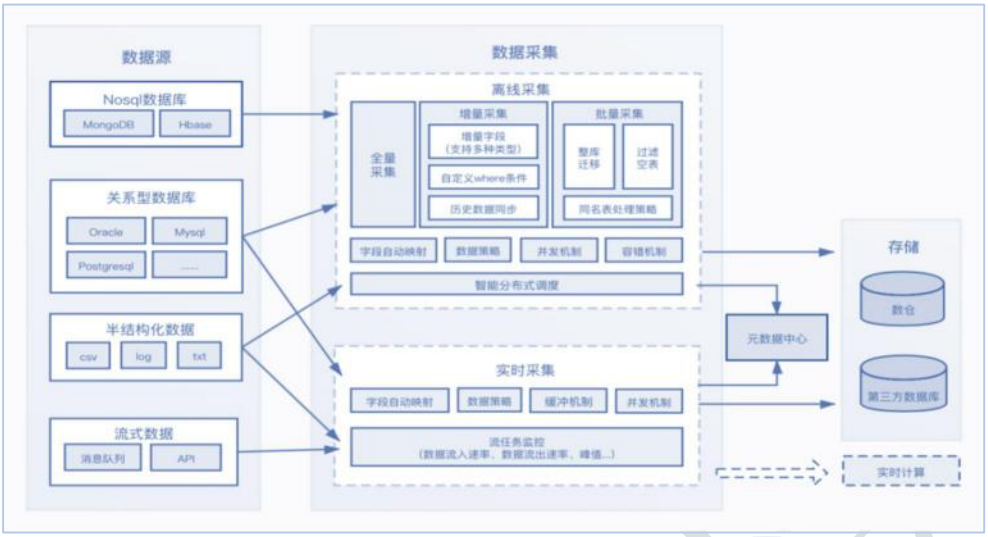

图 5-9 数据融合集成流程

6. 数据资产管理

数据资产管理也称为广义的数据治理，是对数据的全生命周期进行管理，包含数据采集、清洗、转换等传统数据集成和存储环节的工作（本方案中将这部分工作称为狭义的数据治理，简称数据治理，而将广义的数据治理称为数据资产管理。），同时还包含数据资产目录、数据标准、质量、安全、数据开发、数据价值、数据服务与应用等，整个数据生命期而开展开的业务、技术和管理活动都属于数据治理范畴。

6.1.数据资产管理的目标

数据资产管理是数据中心面向用户提供数据能力的一个窗口，数据资产中心将区域教育数据资产统一管理起来，实现数据资产的可见、可懂、可用和可运营。

可见：通过对数据资产的全面盘点，形成数据资产地图针对数据生产者、管理者、使用者等不同的角色，用数据资产目录的方式共享数据资产，用户可以快速、精确地查找到自己关心的数据资产。

可懂：通过元数据管理，完善对数据资产的描述。同时在数据资产的建设过程中，注重数据资产业务含义的提炼，将数据加工和组织成人人可懂的、无歧义的数据资产。具体来说，在数据中心之上，需要将数据资产进行标签化。标签是面向业务视角的数据组织方式。

可用：通过统一数据标准、提升数据质量和数据安全性等措施，增强数据的可信度，让数据科学家和数据分析人员没有后顾之忧，放心使用数据资产，降低因为数据不可用、不可信而带来的沟通成本和管理成本。

可运营：数据资产运营的最终目的是让数据价值越滚越大，因此数据资产运营要始终围绕资产价值来开展。通过建立一套符合数据驱动的组织管理制度流程和价值评估体系，改进数据资产建设过程，提升数据资产管理的水平，提升数据资产的价值。

6.2.数据资产管理在数据中心架构中的位置

数据资产管理在数据中心架构中处于中间位置，介于数据开发和数据应用之间，处于承上启下的重要地位（如下图所示）。数据资产管理对上支持以价值挖掘和业务赋能为导向的数据应用开发，对下依托大数据平台实现数据全生命周期的管理，并对教育数据资产的价值、质量进行评估，促进教育数据资产不断自我完善，持续向业务输出动力。

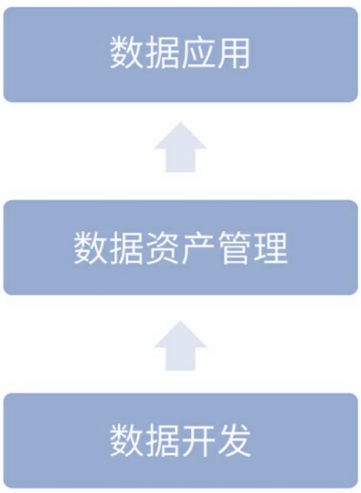

图 6-1 数据资产管理的位置

数据资产体系是通过数据开发得到的宝贵成果。通过良好的数据资产管理，一方面可以保证数据资产的质量，提升数据资产的可信度；另一方面，组织良好的数据资产，既能为各类角色的用户提供数据资产的直观视图，方便用户查看和使用，又能源源不断地输出数据资产服务能力，持续赋能业务场景。

6.3.数据治理

对汇聚加载后的各系统数据进行数据治理，并能够实现数据在不同层次的转换，形成区域教育融合数据仓库。

在大规模数据场景下，数据治理是在数据汇聚加载后对数据进行预处理，生成贴源层，然后再对贴源层数据进行进一步的数据清洗、数据加工转换和数据融合（高度汇聚），形成区域教育融合数据仓库。

6.3.1. 数据预处理

初步汇聚并加载后的数据进行预处理，尽可能保留原始业务流程数据，与业务系统基本保持一致，仅做简单整合、非结构化数据结构化处理或者增加标识数据日期描述信息，不做深度清洗加工。经过预处理的数据构成区域教育数据体系中的贴源层。

6.3.2. 数据清洗

数据清洗是指发现并纠正数据文件中可识别的错误（脏数据）的过程，包括检查数据一致性，处理无效值和缺失值等。

数据清洗是一个和业务用户反复沟通的过程，不可能在很短的时间内完成，只能不断的发现问题，可能解决问题。对于是否过滤，是否修正，由人工确定，对于过滤掉的数据要写入文本文件、Excel文件、数据库表。数据清洗需要注意的是对于每个过程规则都要认证进行验证，并由人工确定。

(1) 需要清洗的数据

主要针对以下几种脏数据进行清洗：

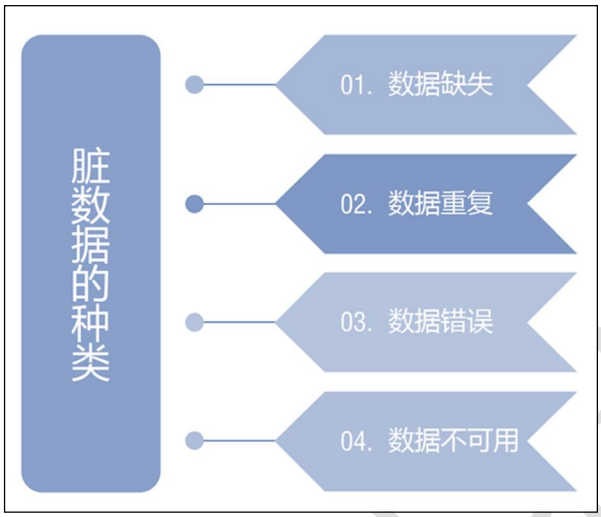

图 6-2 脏数据分类图

✓ 数据缺失

处理过程中因系统或人为导致部分记录缺失的，如一条记录里缺一些值（空值），或者两者都缺。如果有空值，为了不影响分析的准确性，则判断是否将空值纳入分析范围，或者进行补值。前者排除空值会减少分析的样本量，后者需要根据分析的计算逻辑，选择用平均数、零、或者等比例随机数等来填补。如果缺少记录部分，若业务系统中还存有这些记录，则可以通过系统再次导入解决，若业务系统内也没有上述记录，则通过手工补录或者放弃。

✓ 数据重复

相同的记录出现多条，则去掉重复记录。如出现不完全重复情况，比如两条会员记录，其余值都一样，但部分值不同，如住址不一样，则以时间属性做为新值判断依据，如无时间属性的，则通过人工判断处理。

✓ 数据错误

数据没有严格按照规范记录。比如异常值；比如格式错误，日期格式录成了字符串；比如数据不统一，有的记录叫上海，有的叫SH，有的叫shanghai。对于

异常值，可以通过区间限定来发现并排除；对于格式错误，需要从系统级别找原因；对于数据不统一，系统无法处理，这些并非真正“错误”的记录，如系统无法判断SH和shanghai是同一事物，只能通过人工干预解决，如做一张清洗规则表，给出匹配关系，第一列是原始值，第二列是清洗值，用规则表去关联原始表，用清洗值做分析结论，或通过近似值算法自动发现可能存在不统一的数据。

✓ 数据不可用

数据正确，但不可用。比如地址写成“上海市闵行虹桥火车站”，想分析“区”级别的区域时还要把“闵行”拆出来才能用。这种情况最好从源头解决，即数据治理。事后补救只能通过关键词匹配，且不一定能全部解决。

(2) 清洗方式

结构化数据、非结构化数据需要按照不同的方式进行清洗。

✓ 非结构化数据清洗

非结构数据主要为文本、XML、图片等数据，对于非结构化数据，主要通过同一时间窗口对比去重、MD5值对比去重等技术方法去重。

文本数据的清洗，主要基于自然语言处理技术，通过分词、语料标注、字典构建、关键词识别等技术，根据相应的非结构化数据特点进行数据建模，利用机器学习和数据挖掘的方法进行文件去重。

图片数据可以通过以图找图技术，进行图片去重。根据相似图像检测技术以通过提取某些表征图像内容的特征，与数据库中目标图片特征进行匹配判断。

✓ 结构化数据清洗

结构化数据清洗需遵从数据标准，根据业务规划对冗余数据进行过滤，根据不同的去重规则和方法对数据进行去重判定，去除重复冗余数据。通过定义过滤规则，使用流式SQL和表达式，按条件对数据进行重新组合和二次加工。数据清洗可以区分为冗余信息过滤、敏感信息过滤、数据去重和格式清洗等内容。通过对数据进行清洗，提高数据的使用价值。数据清洗在具体实现上可分为全景清洗、

增量清洗，根据实时性需要可以区分为实时清洗、非实时清洗。清洗过程又分为过滤、去重、检验、格转。

6.3.3. 数据加工转换

预处理后的贴源数据不一定完全满足目的库的要求，例如身份证号码，其实经常只需要抽取里面的出生年月信息。为了满足多场景的数据需求，需要对数据字段进行信息提取、计算、分组、转换等加工，将它们转换为我们需要的数据。数据的转换加工在数据处理软件引擎中进行。

1．数据加工转换类型

（1）格式修正：数据类型和字段长度；

（2）字段的解码：使得晦涩的值变得用户易于理解和有意义；

（3）计算值和导出值；

（4）单个字段的分离：姓和名，邮编和地址，等等；

（5）信息合并：从不同源系统中得到某个新的实体的过程；

（6）特征集合转化：编码的转化：ASCII码、BCD码、Unicode、Big5、GB2312等等；

（7）度量单位的转化；

（8）日期、时间格式的转化；

（9）汇总；

（10）健重构。

2. 数据加工转换的方法

（1）数据抽取

数据抽取是指保留原数据表中某些字段的部分信息，组合成一个新字段。可以是截取某一字段的部分信息——字段分裂；也可以是将某几个字段合并为一个

新字段——字段合并；还可以是将原数据表没有但其他数据表中有的字段，有效地匹配过来——字段匹配。

（2）数据计算

当需要的数据不能从数据源表字段中直接提取出来时，就需要通过计算来实现实际的数据需求。简单的数据计算通过将字段通过加、减、乘、除等简单算术运算即可实现。而较为复杂的计算需要运用函数来实现，比如平均值、总和、日期的加减、日期间隔计算等，以及更复杂的数据计算。

（3）数据分组

数据分组是根据统计研究的需要，将原始数据按照某种标准划分成不同的组别，分组后的的数据称为分组数据。数据分组的方法有单变量值分组和组距分组两种。数据分组的主要目的是观察数据的分布特征，在进行数据分组后再计算出各组中数据出现的频数，就形成了一张频数分布表。

（4）数据转换

数据转换是将数据从一种表示形式变为另一种表现形式的过程。根据数据分析的需要，有些数据需要进行适当的转换才便于利用，例如数据表的行列转换，多选题几种录入方式之间的转换等。

6.3.4. 数据治理平台

数据中心的数据治理工具可以帮助用户快速、高效地实现数据治理工作。通过治理工作流开发，用户可使用系统提供的数据治理模组，通过数据接入、质量探查、标准化、SQL加工、数据加工和数据接入不同模组的功能，从而完成简单的数据清洗工作；也可自行编写或导入Shell、Python、Python3、SQL、Kettle脚本，进一步满足自身的业务需求。

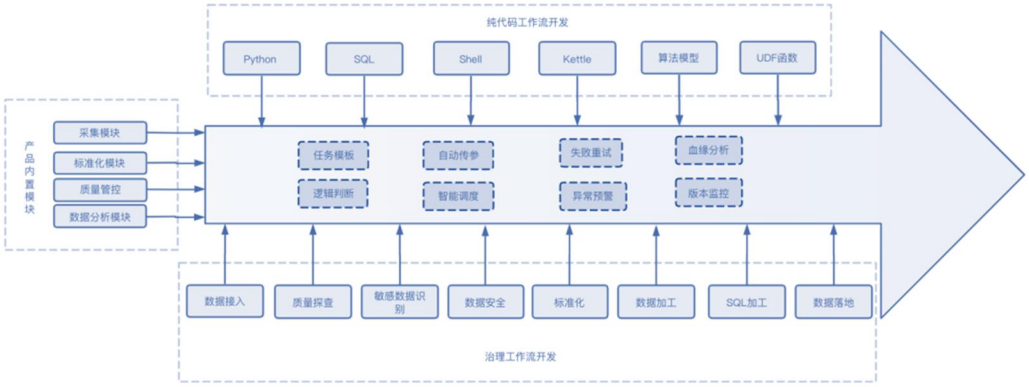

图 6-3 数据治理工作流管理

数据治理平台为治理工作流开发提供预制的治理模组，包括数据接入、质量探查、标准化、SQL、敏感数据识别、数据安全、数据加工和数据落地，数据的流向以连接治理模组的形式清晰地展示给用户，即用户通过可视化流程图的方式达到数据治理的目的，相比纯代码的工作流开发方式，治理工作流开发免去了写代码和调试代码的技术及时间成本。

支持用户查看和编辑工作流的基本信息、调度策略和参数设置详情。

1. 数据接入

数据接入模组对输入数据进行选择和基本的处理，这是治理工作流必须配置的步骤。

用户在数据接入模组选择可操作资源中的表作为处理对象，支持设置增量字段以实现增量输入，过滤重复数据，填写Where条件和筛选器过滤输入数据。支持用户查看和编辑数据接入的基本信息功能。

2. 质量探查

质量探查是对输入数据的质量检测，是开发者对数据的质量（比如脏数据比例）进行解析和判断的工具，该模组对数据不产生直接处理操作，但有过滤作用。

质量探查会对探查到的数据问题进行强弱分级，问题数据将被过滤到问题库中，探查结果可在数据中心的问题数据中查看，整体的探查结果可以生成质量报告便于二次分析。

探查规则分为基础探查和深度探查。

基础探查有：空值率探查，类型探查，长度探查，数值范围探查和字段内容探查；

深度探查有：标准规则（数据标准）、业务规则（udf函数规则）、正则表达式。

支持用户查看和编辑质量探查的基本信息功能。

3. 标准化

标准化是对数据中不规范的字段和数据进行修改，使其符合治理标准。

标准化有三个主要功能：

1. 字段名称标准化：将字段名更改为数据标准中的标准字段名。

2. 代码值标准化：将字段中的数据进行聚类，并替换为映射的标准代码值。支持 AI 自动映射或手动拖拽映射的方式映射标准代码值。

3. 添加字段备注信息：为字段添加备注信息。

为了满足更多标准化场景，允许为标准化后的字段名增加前、后缀。即可实现同一标准字段可以被多次使用且不重复。

支持用户查看和编辑标准化的基本信息功能

4. 数据加工

数据加工模组的输入可进行联表操作，有左关联，右关联，内关联和外部关联四种方式。系统提供9种基本加工组件，不同组件实现不同的字段记录处理功能：

1. 添加缺省值 ：当字段记录内容为空时填充默认值
2. 替换指定内容：替换字段记录中指定内容
3. 字段内容拼接：在开头或结尾的间隔位置添加拼接内容
4. 字段内容运算：对字段内容进行运算
5. 截取指定位置内容：截取字段记录中指定位置内容
6. 添加指定内容：在指定内容前后添加内容
7. 删减指定内容：删除字段记录中指定内容
8. 合并字段内容：多个字段合并为一个字段
9. 分裂字段内容：一个字段分裂为多个字段

支持用户查看和编辑数据加工的基本信息功能。

5. SQL 脚本

除系统提供的数据加工组件外，平台还提供IDE开发环境，支持SQL脚本的在线编辑、运行、调试以满足用户在业务和技术上对数据治理更深层的需求。

SQL加工模组支持用户查看与外部工作流同步的时间参数，支持用户查看和编辑SQL组件的基本信息和代码开发功能。

6. 数据落地

各个模组的治理工作完成后，数据落地支持对上一个模组的输出数据落地存储，支持用户查看和编辑数据落地的基本信息、字段信息和分区信息功能。

6.4.数据标准管理

实现区教育数据标准的管理，主要包括以下功能：

6.4.1. 标准管理

支持新增、修改、删除、发布、停用数据标准，支持查看所有的标准信息，包括标准名称、编码、状态、备注。

支持通过键入标准名称关键词搜索标准。

支持用户添加子集，通过填写标准名称、编码、备注新增子集。

支持添加标准集关联对象，用户可选择对象信息，添加关联对象。

6.4.2. 维度标准

支持新增、修改、删除维度类型，支持查看所有的维度类型信息，包括维度类型，维度类型代码，备注，状态。维度标准为创建数据表时某些字段的属性和取值提供统一参考标注。

支持新增、修改、删除维度，支持维度信息，包括填写，维度，维度代码，状态，备注新增维度信息的创建、修改。

6.5.元数据管理

元数据是描述数据的数据。在大数据平台中，元数据贯穿大数据平台数据流动的全过程，主要包括数据源元数据、数据加工处理过程元数据、指标层元数据、标签层元数据、服务层元数据、应用层元数据等。元数据通常分为以下类型。

- 技术元数据：库表结构、字段约束、数据模型、ETL 程序、SQL 程序等。
- 业务元数据：业务指标、业务代码、业务术语等。
- 管理元数据：数据所有者、数据质量定责、数据安全等级等。

6.5.1. 元数据的采集

元数据采集是指获取到分布在不同系统中的元数据，对元数据进行组织，然后将元数据写入数据库中的过程。

要获取到元数据，需要采取多种方式，在采集方式上，使用包括数据库直连、接口、日志文件等技术手段，对结构化数据的数据字典、非结构化数据的元数据信息、业务指标、代码、数据加工过程等元数据信息进行自动化和手动采集。

元数据采集完成后，会被组织成方便查看和分析的数据结构，通常被存储在关系型数据库中。

区域教育数字元数据包括通用、物联中心、数据中心、组织中心、消息中心、应用中心和生存期等七个部分，具体信息如下表所示。

表格 6-1 教育数据元数据结构表

编号	名称	对应的英文名称	解释	属性	大小	次序	值域	数据类型	举例
1	通用	general	该类别描述了数字基座的五大中心的一些通用信息	M	1	—			
1.1	标识	identifier	基座所属学校标号，该标号唯一	M	*10	否			
1.1.1	类型	catalog	标识规定的类型	M	1	—		字符串*(50 个字符)	
1.1.2	值	entry	遵循一种命名方案	M	1	—			
1.2	标题	title	基座名称，遵循一种命名方案	M	1	—		字符串*（100 个字符）	校名+区号+编号（需求）
1.3	覆盖范围	coverage	基座对象所涉及的时间、区域	0	*10	否			
2	物联中心	IOT Center	该类别描述物联中心的一些通用信息	M	1	—			
2.1	结构	structure	该中心的基本组织结构的描述	M	1	否			
2.2	接口	interface	该中心主要接口及描述，包括设备注册、查询、上报、获取设备状态、指令下发等	M	1	否			
2.2.1	类别	sort	接口的类别，具有唯一性	M	1	—			
2.2.2	详情	Details	指定接口中的具体参数，包括请求参数和返回参数	M	1	—			
3	组织中心	Organize Center	该类别描述组织中心的一些通用信息	M	1	—			
3.1	结构	Structure	该中心的基本组织结构的描述	M	1	—			
3.2	组织管理API	Organization Management API	包括组织的创建、修改、查询、删除、添加和移除组织中的用户等	M	1	—			
3.3	用户管理API	User Management API	对基座内各类使用人员信息进行维护管理。包括创建修改查询等用户信息	M	1	—			
4	数据中心	Data Center	该类别描述组织中心的一些通用信息	M	1	—			
4.1	结构	Structure	该中心的基本组织结构的描述	M	1	—			
4.2	数据源管理API	Data source management API	主要包括数据源的创建、查询、删除等操作	M	1	—			
4.2	数据集成	Data integratio	主要包括任务的创建、修改、查询、删除、启动等操	M	1	—			

	API	n	作						
4.3	任务监控管理API	Task monitoring management API	主要查询任务的运行信息	M	1	—			
4.4	服务集成API	Service integration API	包括API的创建、修改、查询、删除、发布等操作	M	1	—			
4.5	消息集成API	Message integration API	主要包括消息的管理和查询	M	1	—			
5	消息中心	Message Center	该类别描述组织中心的一些通用信息	M	1	—			
5.1	结构	Structure	该中心的基本组织结构的描述	M	1	—			
5.2	使用场景	Usage scenario	主要提供接口类别的服务	M	1	—			
5.3	接口详情	Interface details	主要提供接口服务的发送请求参数	M	1	—			
6	应用中心	Application Center	主要授予基座使用者应用的查询	M	1	—			
6.1	结构	Structure	该中心的基本组织结构的描述	M	1	—			
6.2	详情	Details	包括应用请求和响应的参数	M	1	—			
7	生存期	life cycle	该类别描述了基座的历史和当前状态	M	1				
7.1	版本	version	基座的版本状态	O	1				
7.2	状态	status	基座的状态或使用状况	O	1	—	测试版 试运行版 运行版 不可用		
7.3	贡献	contribute	基座的开发者	M	*30	否			
7.3.1	角色	role	开发商名称	M	*50	-			
7.3.2	日期	Date	开发日期	M	*50	-		日期	yyyy-mm-dd
7.4	使用	use	基座的使用者	M	*30	否			
7.4.1	角色	role	使用者名称		*50	-			
7.4.2	日期	Date	使用日期		*50	-		日期	yyyy-mm-dd

注1：“属性”栏内M表示必须数据元素，O表示可选数据元素。

注2：“大小”和“数据类型”栏内有星号标记的为最低峰值。

6.5.2. 元数据的管理

元数据的管理包含元数据的增删改查、变更管理、对比分析、统计分析等。

•元数据的增删改查。通过对不同的角色赋予相应的权限，实现元数据在组织范围内的信息共享。值得注意的是，对元数据的修改、删除、新增等操作，必须经过元数据管理员的审核流程。

• 元数据变更管理。对元数据的变更历史进行查询，对变更前后的版本进行比对等。

• 元数据对比分析。对相似的元数据进行比对。比如，对近似的两张表进行对比，发现它们之间的细微差异。

•元数据统计分析。用于统计各类元数据的数量，如各类数据的种类、数量、数据量等，方便用户掌握元数据的汇总信息。

6.5.3. 元数据的应用

（1）元数据浏览和检索

通过提供直观的可视化界面，让用户可以按不同类型对元数据进行浏览和检索。通过合理的权限分配，元数据浏览和检索可以大大提升信息在组织内的共享。

（2）数据血缘和影响性分析

数据血缘和影响性分析主要解决“数据之间有什么关系”的问题。

血缘分析指的是获取到数据的血缘关系，以历史事实的方式记录数据的来源、处理过程等。

影响性分析可以分析出数据的下游流向。当系统进行升级改造的时候，如果修改了数据结构、ETL程序等元数据信息，依赖数据的影响性分析，可以快速定位出元数据修改会影响到哪些下游系统，从而减少系统升级改造带来的风险。数据影响性分析和血缘分析正好相反，血缘分析指向数据的上游来源，而影响性分析指向数据的下游。

（3）数据冷热度分析

冷热度分析主要是对数据表的被使用情况进行统计，如表与ETL程序、表与分析应用、表与其他表的关系情况等，从访问频次和业务需求角度出发，进行数据冷热度分析，用图表展现表的重要性指数。

数据的冷热度分析对于用户有着巨大的价值，其典型应用场景有：如果观察到某些数据资源处于长期闲置，没有被任何用户查看，也没有任何应用去调用它的状态，用户就可以参考数据的冷热度报告，结合人工分析，对冷热度不同的数据做分层存储，以便更好地利用HDFS资源，或者评估是否对失去价值的这部分数据做下线处理，以节省数据存储空间。

6.5.4. 功能模块设置

提供企业级的元数据统一视图，提供前置库、中心库、主题库等视图，并能够清晰地分析和跟踪业务运作历史数据变化。

（1）逻辑元数据模块

逻辑数据模型包含了数据实体和实体间的关系、属性、定义、描述和范例。共享元数据模型是数据元规范、编码规范在数据中心系统层面的落地，管理数据区的模型层严格参照共享数据模型的逻辑模型进行设计。元数据模型承担起各类业务系统数据结构的校验与审计功能。

元数据建模是根据业务的需求，将业务领域内的对象进行整合建模形成业务模型，它是具有高业务价值的、可以在政府内部跨越各个部门被重复使用的数据，并且存在于多个异构的应用系统中。

支持新增、修改、发布、停用、删除元数据模型，支持查看所有元数据信息，包括模型名称，中文名，数据源，主题，层次，来源，schema，状态和更新时间。

支持用户搜索元数据模型，用户可通过键入模型名称或中文名的关键字搜索元数据。

支持用户筛选元数据模型，筛选条件包括数据源，主题，Schema，资源层次，用户可按条件筛选元数据模型。

（2）元数据采集

提供元数据全文检索功能，对元数据库中的元数据基本信息进行查询。可查询数据库表、维表、指标、过程及参与的输入输出实体信息，以及其它纳入管理

的实体基本信息，查询的信息按处理的层次及业务主题进行组织，查询功能返回实体及其所属的相关信息，帮助用户在企业级海量信息库中快速检索并准确定位信息项。

支持创建采集日志，通过填写必要信息，创建元数据采集任务，采集任务包括采集任务名，数据源，默认主题，资源层次，采集表正则，模型合并状态，自动发布状态。

支持修改、删除，支持查看所有采集任务，包括采集任务名称，采集元数据类型，创建日期，状态。

支持通过键入采集任务的关键字来搜索采集任务。

支持用户手工执行采集任务。

支持用户进行调度配置，可以通过填写调度时间进行配置，同时支持Cron表达式。

（3）元数据采集日志

系统主要对数据库的运行状态进行监控和分析，对数据库的数据抽取、清洗转换、加载入库、提供服务等过程进行全程监控，对数据采集数量、质量、频度、构成等信息进行分析挖掘，生成采集日志，动态监测预警，提醒系统管理员及时发现并处理数据异常。

支持查看所有元数据采集日志信息，包括任务名称，任务类型，执行次数，执行状态，操作，开始执行日期，结束执行日期，执行时长。

支持查看特定任务的元数据日志的详细内容。

支持搜索元数据采集日志，通过键入任务名称关键字，收缩元数据采集日志。

（4）元数据血缘维护

数据之间的链路关系就可称为数据血缘，在数据共享、数据分析过程中一个非常重要的问题就是要保障良好的数据质量，如果原数据本身就是“脏”数据，那么有可能导致共享出的数据不准确、分析出的结果和实际情况大相径庭。这不

是技术问题，而是方法和应用的问题为了更好地及时分析、查找、评估和解决数据共享和数据分析各环节的数据质量问题，保证数据质量的稳定和可靠，需要通过数据质量的管理体系，方便用户监测数据流转过程中的每个环节的数据质量，通过血缘分析实现数据融合处理的可追溯。

数据统计：按数据类型、数据目录、数据创建者、用户权限、元数据的版本号等条件或条件组合，统计符合条件的元数据的数量，方便用户全面了解元数据的分布情况及各种元数据的规模。

支持新增、修改、删除表级关系，支持查看所有的表级关系，包括关联对象类型，关联任务，源数据源，目标数据源，源数据源用户，目标数据源用户。提供表间关系和字段关系的维护，展示源表和与其相关的目标表。

数据标准检索：通过全文检索能迅速查找和关键字匹配的权限范围内的数据标准信息，为海量数据分析提供快速正确的查询处理、易使用的操作接口等。提供按照数据标准类型分类检索功能。

支持搜索表级关系，通过键入关联对象关键字搜索表级关系。

（5）血缘作业管理

支持新增、修改、删除、发布停用血缘作业任务。支持查看所有的血缘作业，包括任务名称，备注，状态。配置在进行血缘维护时需要执行的附加操作（只记录执行的sql语句，不做实际操作），字段关系维护时可以关联某个转换（执行SQL语句），表级关系维护时可以关联某个任务（一个任务可能包含多个转换）。

支持可键入任务名称关键字搜索任务。

支持筛选任务，按任务状态进行筛选，筛选状态包括发布，停用。

6.6.数据质量管理

数据质量不高将影响数据应用程度不高。低下的数据质量往往造成开发出来的系统与用户的预期大相径庭，数据质量关系建设有关分析型信息系统成败，同时数据资源是教育战略资源，合理有效的使用正确的数据能指导教育机

构做出正确的决策，提高综合治理能力。不合理的使用不正确的数据（即差的数据质量）可导致决策的失败，正可谓差之毫厘、谬以千里。

数据质量管理包含对数据的绝对质量管理、过程质量管理。绝对质量即数据的真实性、完备性、自治性是数据本身应具有的属性。过程质量即使用质量、存储质量和传输质量，数据的使用质量是指数据被正确的使用。再正确的数据，如果被错误的使用，就不可能得出正确的结论。数据的存贮质量指数据被安全的存贮在适当的介质上。所谓存贮在适当的介质上是指当需要数据的时候能及时方便的取出。数据的传输质量是指数据在传输过程中的效率和正确性。

6.6.1. 数据质量要求

高质量的教育行业数据至少有如下几项要求：

1）准确性：描述数据是否与其对应客观实体的特征一致。

举例：用户的住址是否准确；某个字段规定应该是英文字符，在其位置上是否存在乱码。

2）完整性：描述数据是否存在缺失记录或缺失字段。

举例：某个字段不能为 null 或空字符。

3）一致性：描述同一实体同一属性的值在不同的系统中是否一致。

举例：男女是否在不同的库表中都使用同一种表述。例如在A系统中，男性表述为1，女性表述为0；在B系统中，男性表述为M，女性表述为F。

4）有效性：描述数据是否满足用户定义的条件或在一定的取值范围内。

举例：年龄的值域在0~200之间。

5）唯一性：描述数据是否存在重复记录。

举例：身份证号码不能重复，学号不能重复。

6）及时性：描述数据的产生和供应是否及时。

举例：生产数据必须在凌晨2：00入库到ODS（OperationalDataStore，操作数据层）。

7）稳定性：描述数据的波动是否稳定，是否在其有效范围内。

8）连续性：描述数据的编号是否连续。

9）合理性：描述两个字段之间逻辑关系是否合理。

举例：比如毕业时间必须晚于入学时间，自然人的死亡时间必须晚于出生时间。

以上数据质量标准只是一些通用的规则，还可以根据数据的实际情况和业务要求对其进行扩展，如进行交叉表数据质量校验等。

6.6.2. 数据质量管理的流程

要提升数据质量，需要以问题数据为切入点，注重问题的分析、解决、跟踪、持续优化、知识积累，形成数据质量持续提升的闭环。数据质量的管理流程如下图所示。

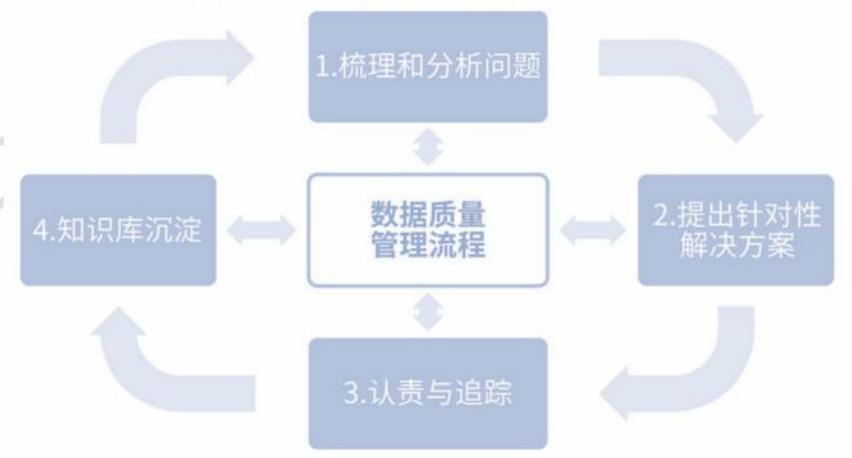

图 6-4 数据质量管理流程

首先需要梳理和分析数据质量问题，摸清数据质量的现状。在这个过程中，需要用到数据质量评估标准和评估工具，对业务数据进行全部或抽样扫描，找出不符合质量要求的数据，形成数据质量报告，提供给用户参考。

然后针对不同的质量问题选择合适的解决办法，制订详细的解决方案。如在落地到数据中心的过程中进行数据清洗，在数据录入的源头进行质量把控，对数据建模过程进行是否符合质量标准的审核，等等。

接着是问题的认责，追踪方案执行的效果，监督检查，持续优化。在这一步，要把每一个问题都落实到具体的责任人，并且形成一套最终的考核机制，督促相关的责任人持续不断地关注与提升数据质量。

最后形成数据质量问题解决方案的知识库，以供后来者参考。数据质量问题往往不是偶然出现的，许多问题都有其共性，比如由于在数据录入环节，对输入框中输入的内容没有严格限定格式，会批量造成同样的数据质量问题，把这些质量问题的解决过程沉淀下来，形成知识库，有助于提升后续数据质量问题的处理效率。

不断迭代上述步骤，形成数据质量管理的闭环。

要管理好数据质量，仅有工具支撑是远远不够的，必须要组织架构、制度流程参与进来，才能形成一套完整有效的管理流程。

6.6.3. 数据质量管理实施

实现对数据中心汇聚的数据进行质量管控，包括数据质量监测和处理的规则配置，对于数据完整性、有效性的校验，数据质量监控，对低质量的问题数据依据规则进行处理，以及定期对基础平台中的数据质量进行评估，以及生成质量评估报告，从而有效提升数据的完整性、及时性、有效性和统一性。

数据质量主要提供多种异构数据源的质量校验、通知、管理服务等功能。

1. 数据准备

支持创建表，并在表中写入样板数据。

2. 规则配置

规则配置是数据质量中最核心的部分。主要功能如下：

选择数据源。在规则配置里选择数据源。

搜索表。根据对象表名，找到对应的表。

配置监控规则。对监控规则进行配置。

配置分区表达式。分区表达式是用来对校验规则进行匹配的筛选条件。

关联调度。监控离线数据质量，将数据质量关联调度。

3. 任务查询

主要显示规则校验结果情况列表与查看，支持查看规则运行记录。

6.6.4. 数据质量监控

构建数据质量规则体系，并在数据质量规则体系的基础上，建立数据质量监控的任务管理、调度和规则执行体系，构建规则执行引擎，由系统自动对各种数据项进行审核，发现集中数据中存在的质量问题，产生数据质量问题数据库；在数据质量问题数据库基础上，构建数据质量管理应用体系，通过各种统计和分析手段对数据质量问题进行分析和汇总，并根据数据质量管理流程和制度要求，对各业务系统进行数据质量问题通报，提出数据质量整改要求，提供问题数据查询、统计等辅助功能题采集、检查、报告、总结等各步骤的跟踪，完成数据质量生命周期管理全过程。

6.7.生命周期管理

对于数据的生命周期管理，普遍的做法是，根据业务特点，分析数据使用状况，将数据分为冷数据与热数据，然后对冷热数据采取不同的管理策略。一般采用以下管理策略。

利用云对象存储的力量：将热数据保存在当前大数据集群中支撑当前的业务系统，而将冷数据备份到云对象存储上；

冷热数据分集群存储：将热数据保存在当前大数据集群中支撑当前的业务系统，并搭建专门的冷数据集群，将冷数据转存到冷集群中；（冷集群更侧重存储能力，热集群更侧重计算能力，在集群底层服务器选型上各有侧重）；

利用HDFS本身提供的分级存储的策略：可以对不同的目录，根据其数据冷热程度不同，动态配置不同的存储策略，从而存储到不同的底层存储介质上。可以使用的存储类型有 archive, disk, ssd 和 ram_disk，可以配置的存储策略有 hot, warm, cold, All_SSD, One_SSD, Lazy_Persist and Provided。

综评数据中台的数据存储采用了HDFS分布式文件系统，可以优先采用HDFS本身提供的分级存储策略。

直接删除冷数据：当前的大数据集群只保存业务需要的数据，而将业务不需要的历史数据，定期通过脚本进行删除。这种方式，因为需要删除数据，所有只有在业务方确认数据确实不需要了，这种方式很少采用。

6.8.数据资产地图

数据资产地图为用户提供多层次、多视角的数据资产图形化呈现形式。数据资产地图让用户用最直观的方式，掌握数据资产的概况，如数据总量、每日数据增量、数据资产质量的整体状况、数据资产的分类情况、数据资产的分布情况、数据资产的冷热度排名、各个业务域及系统之间的数据流动关系等。

图 6-5 区域数据资产总览

图 6-6 学校数据资产总览

6.9.数据资产编目

数据资产编目提供数字资产盘点和管理服务，可以将分散的、多数据源的数据进行统一登记、汇集、标准化等操作处理，实现对数据的分布和动态变更情况的追踪，监控和提升数据质量。

系统支持连接十余种数据库，包括关系型数据库、MPP 数据库和分布式数据库，涵盖 Mysql、Oracle、DB2、SQL server、Postgres、Greenplum、Teryx、Hive、CSV 多个版本的数据库类型。

提供数据源文件管理功能，支持对数据源的多级文件夹编目管理。

支持对数据源的管理，可对数据源进行搜索、筛选、编辑、详情查看以及数据源连接日志信息的查看。

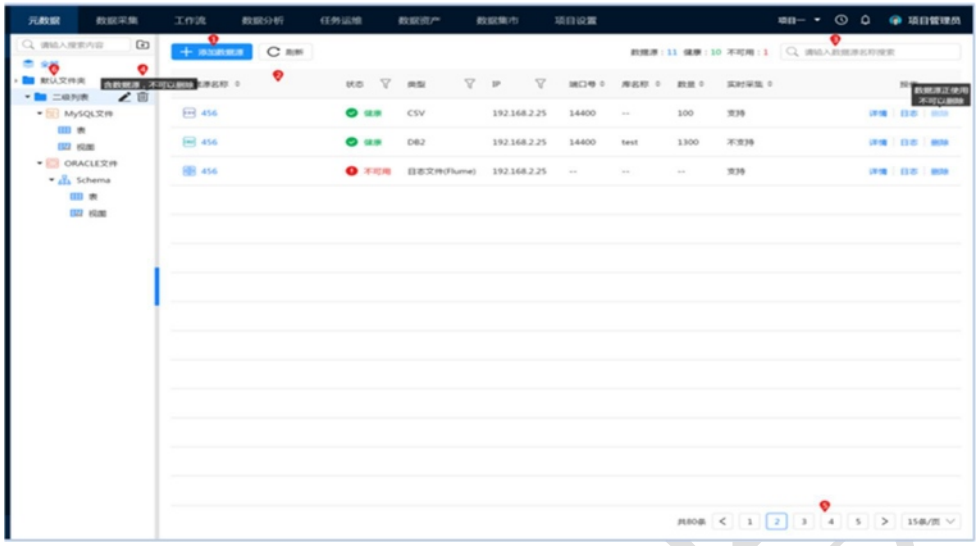

图 **错误!文档中没有指定样式的文字。**-7 数据源管理

数据源管理支持自动同步数据源的元数据信息，包括数据源自身的元数据，数据源内表和视图的元数据信息。

支持用户对每个数据源所包含的表和视图的信息进行查看，也支持对数据源中的表和视图的元数据信息的查看，包括表名称、创建时间、更新时间、描述、字段结构、历史版本元数据信息，历史版本记录元数据的历史变动信息。

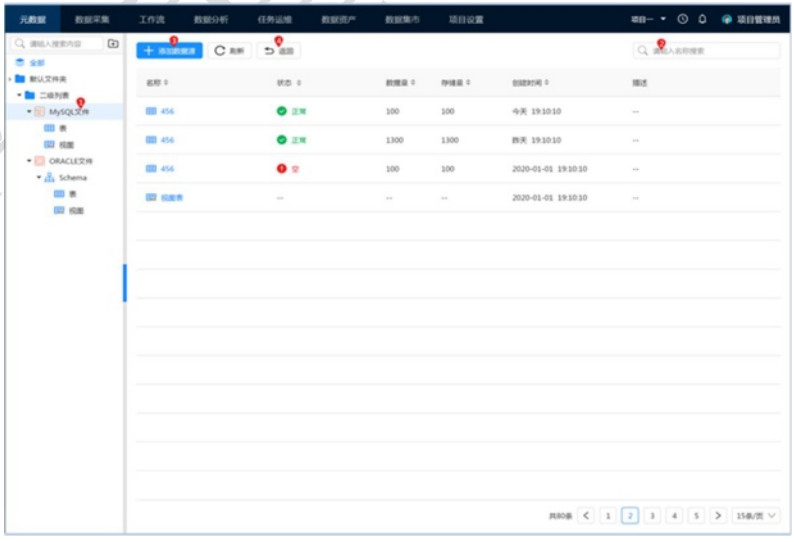

图 **错误!文档中没有指定样式的文字。**-8 数据源内数据表信息

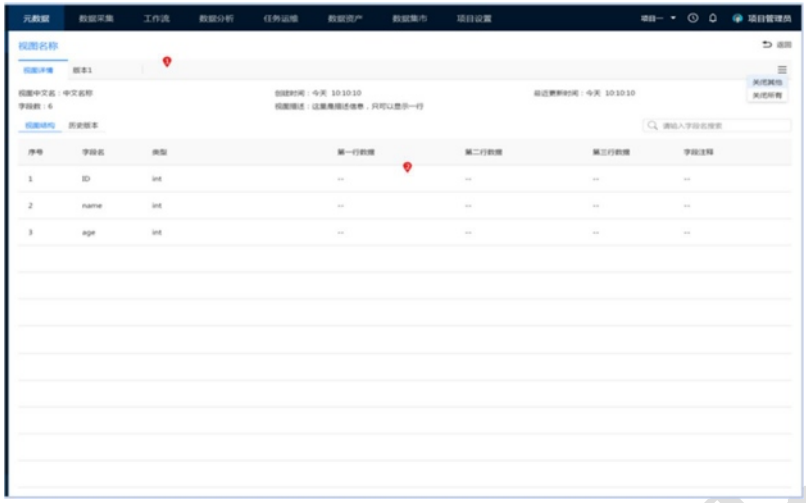

图 **错误!文档中没有指定样式的文字。**-9 数据表的元数据信息

系统提供强大的元数据开发功能，通过元数据开发可构建数据仓库分层管理模型，数据表、资产管理的编目。

6.9.1. 分层管理

数据的开发治理过程包括数据从各种业务数据库、日志数据库被同步到数仓中，再经过数据的清洗、标准化、加工治理、分析生成专题，最后沉淀为数据资产。整个数据的生命链条完整而复杂。为了使数据能够有清晰的组织形式，安全的分析开发环境,高效的查询效率，完整的血缘关系，合理的数据分层就十分有必要。

系统内置开发层、主题层和服务层三个分层，可以进行编辑修改。其中开发层为数据同步后落地的数据层，主题层为数据加工治理后以主题的形式落地的数据层，服务层是对外提供数据服务的数据层。

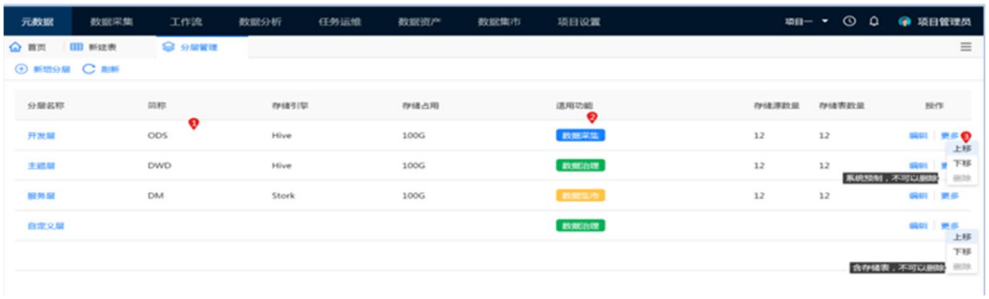

图 **错误!文档中没有指定样式的文字。**-10 分层管理

除内置的三个数据层外，支持用户根据业务需要自由增加分层，以实现更细化的数据分层管理。每个数据层可以根据软件环境信息选择单机数据仓库 stork、分布式 MPP 数据仓库 Teryx、Hadoop 大数据离线仓库 DDP、华为 FI、TDH、Cloudera CDH 作为存储引擎。

支持每个层的存储源数量、存储的数据表的数量进行统计查看。方便用户总览数据的存储情况。也支持客户编辑分层信息，修改分层的排序信息。

6.9.2. 可视化建表

可视化建表可减轻建表的技术难度，降低时间成本。帮助快速完成数据模型创建，实现规范化治理。

支持单表、批量导入的方式导入表信息、表结构进行快速建表。批量导入时可以将表所属的编目信息也导入到数仓的编目元数据中。

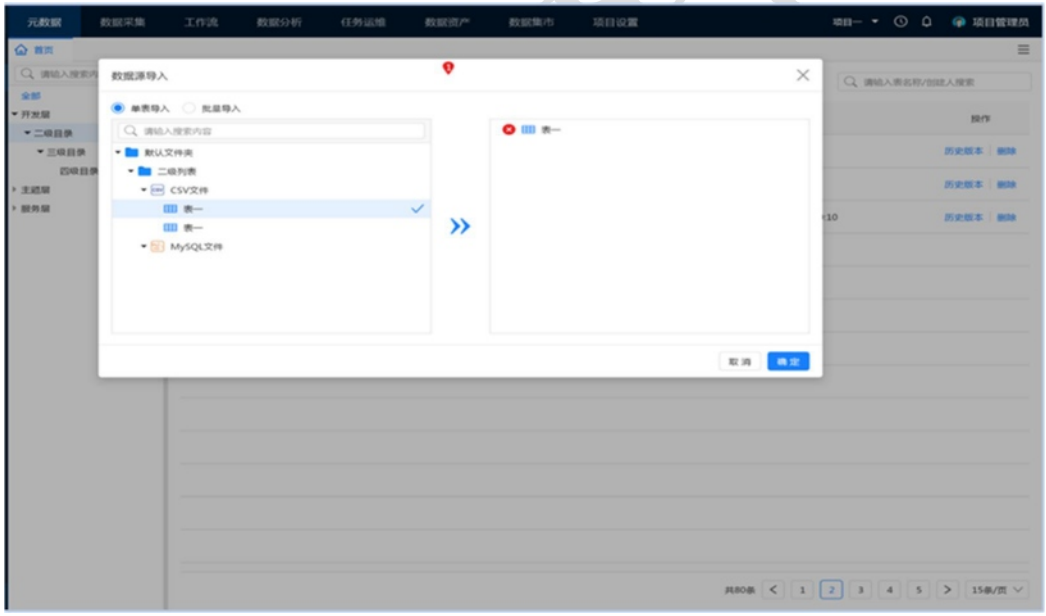

图 **错误!文档中没有指定样式的文字。**-11 导入表信息

支持标准化的方式建表，包括表名、表中文名、存储源、编目、标签、表描述、表存储的文件格式等基础信息和表结构中的字段信息。

图 **错误!文档中没有指定样式的文字。**-12 标准化建表

支持自动同步存储源中的元数据信息，可以将非可视化建表所创建的表信息从存储源数据库中同步到元数据开发管理中。也可将存储源中表的变化信息同步到元数据开发中形成新的元数据版本，原来的元数据可作为历史版本保存，支持查看表的历史版本元数据信息。

6.9.3. 编目管理

编目信息是表的数据目录，编目信息可以按照不同数据的来源、主题、用途等进行设置，帮助整理，归类表，方便后边对于表的查询和管理。

平台支持多至 5 级的编目体系，可以根据具体业务场景自由对编目的级别进行调整。

支持对于编目进行位置顺序的移动，重命名、新增、删除等操作，以满足产品使用过程中因业务场景逐渐丰富产生的对于编目的扩展性需求。

图 **错误!文档中没有指定样式的文字。**-13 编目管理

6.9.4. 表元数据管理

支持对于表的元数据按照表名、创建人名称进行搜索。可按照按照表名称、表中文名、存储源、创建人、修改时间进行排序。

支持对表信息进行删除、编辑，查看元数据详情和历史元数据信息等操作。

6.9.5. 查看资源目录

支持查看所有的资源目录信息，包括名称，数据源，数据源别名，业务主题，层次，主权人，更新时间。通过查看更新时间，用户可以知道数据源的时效。

支持搜索资源目录，通过键入模型名称或中文名关键字搜索资源目录。

支持按条件对资源目录进行筛选，筛选条件包括，按数据源，按主题，按资源层次，按所属主权人筛选。

在数据颗粒度方面，支持查看资源目录的所有字段信息，包括字段名，字段中文名，字段类型，字段长度，小数位数，是否主键，不能为空，是否隐藏，排序。

支持查看部分样板数据，用户可以通过样板数据及其包含的相关字段了解表的详细情况。

6.10. 业务数据可视化

分别从学生成长、教师发展、学校管理、教学管理、综合管理五大主题业务进行业务数据的可视化分析。

6.11. 监管数据可视化

对数据治理全过程的各个阶段的数据情况进行可视化呈现，包括数据集成情况、数据清洗情况、数据托名情况、数据入库（基础层）情况、数据汇总（主题层）情况和数据共享情况，可按时间序列查询统计。

图 6-14 监管数据可视化

7. 数据体系设计

区域教育数据体系是在全域原始数据的基础上，进行标准定义及分层建模，数据体系建设最终呈现的结果是一套完整、规范、准确的数据体系，可以方便支撑数据应用。

7.1. 数据体系特征

覆盖全域数据：数据集中建设，覆盖所有业务过程数据，业务在数据体系中总能找到需要的数据。

结构层次清晰：纵向的数据分层，横向主题域、业务过程划分，让整个层次结构清晰易理解。

数据准确一致：定义一致性指标，统一命名、统一业务含义、统一计算口径，并有专业团队负责建模，保证数据的准确一致。

性能提升：统一的规划设计，选用合理的数据模型，清晰地定义并统一规范，并且考虑使用场景，使整体性能更好。

降低成本：数据体系的建设使得数据能被业务共享，这避免了大量烟囱式的重复建设，节约了计算、存储和人力成本。

方便易用：易用的总体原则是越往后越能方便地直接使用数据、把一些复杂的处理尽可能前置，必要时做适当的冗余处理。比如在数据的使用中，可以通过维度冗余和事实冗余来提前进行相关处理，以避免使用时才计算，通过公共计算下沉、明细与汇总共存等为业务提供灵活性。

7.2.数据体系架构

数据中心的数据体系按照贴源数据、统一数仓、标签数据、应用数据的标准进行定义和划分。

贴源数据层：对各应用系统数据进行采集、汇聚，尽可能保留原始业务流程数据，与业务系统基本保持一致，仅做简单整合、非结构化数据结构化处理或者增加标识数据日期描述信息，不做深度清洗加工。

统一数仓层：又细分为基础数据层和主题数据层，与传统数据仓库功能基本一致，对全历史业务过程数据进行建模存储。对来源于多个应用系统的数据进行重新组织。业务系统是按照业务流程方便操作的方式来组织数据的，而统一数仓层从业务易理解的视角来重新组织，定义一致的指标、维度，各业务板块、业务域按照充一规范独立建设，从而形成统一规范的标准业务数据体系。

标签数据层：面向对象建模，对跨业务板块、跨数据域的特定对象数据进行整合，通过ID-Mapping把各个业务板块、各个业务过程中的同一对象的数据打通，形成对象的全域标签体系，方便深度分析、挖掘、应用。

应用数据层：应用数据层是按照业务使用的需要，组织已经加工好的数据以及一些面向业务的特定个性化指标加工，以满足最终业务应用的场景。

数据中心中数据的读取有严格的规范要求。按照规范，贴源数据层直接从业务系统或目志系统中获取数据。贴源数据层的数据只被统一数仓层使用，统一数仓层数据只被标签层和应用层使用。贴源数据层、统一数仓层只保存历史数据以及被标签层、应用层应用，不直接支撑业务，所有业务使用的数据均来源于标签层和应用层。

由于贴源数据层仅做业务数据的同步与存储，应用数据层面向特定业务组装数据，这两层没有太多的建模规范。

8. 贴源数据层创建

各应用系统数据进行汇聚后的数据形成贴源数据层。贴源数据层对各应用系统数据进行采集、汇聚，尽可能保留原始业务流程数据，与业务系统基本保持一致，仅做简单整合、非结构化数据结构化处理或者增加标识数据日期描述信息，不做深度清洗加工。贴源数据层是统一数仓层建设的数据源。贴源数据层数据不仅是业务数据库中产生的数据，跟业务相关的所有数据都应该汇聚到贴源数据层，包括业务系统数据、业务运行的日志数据或者其他方式获取的外部数据。

8.1. 贴源数据库表设计

贴源数据层中的数据表与对应的业务系统数据表原则上保持一致，数据结构上几乎不做修改，所以参考业务系统数据表结构来设计贴源数据层表结构即可，结构设计上没有太多的规范要求。考虑到业务系统数据的多样性，贴源数据层数据表的设计要遵循一定的规范。

贴源数据层数据表设计规范如下：

•贴源数据层表的命名采用前缀+业务系统表名的方式。比如，ODS_系统简称_业务系统表名，这样既可以最大限度保持与业务系统命名一致，又可以有清晰的层次，还可以区分来源。

•贴源数据层表的字段名与业务系统字段名保持一致，在ODS层不做字段命名归一。字段类型也尽可能保持一致，如果数据中心没有与业务系统对应的数据类型则用一个可以兼容的数据类型。比如，业务系统的数据类型是float，数据中心的存储系统没有float类型，则可以用double代替。

•对于一些数据量较大的业务数据表，如果采用增量同步的方式，则要同时建立增量表和全量表，增量表利用后缀标识。比如，ODS_系统简称_业务系统表名_delta，汇聚到增量表的数据通过数据加工任务合并生成全量表数据。

•对于日志、文件等半结构化数据，不仅要存储原始数据，为了方便后续的使用还要对数据做结构化处理，并存储结构化之后的数据。原始数据可以按行存储在文本类型的大字段中，然后再通过解析任务把数据解析到结构化数据表中。

通过以上建设规范，可保障所有业务数据按照一致的存储方式存储到数据中心。

9. 统一数仓层建设

经过数据治理后的数据，实现了站在业务的视角，不考虑业务系统流程，从业务完整性的角度重新组织数据，从而形成区域教育数据仓库，即数据中心的统一数仓层。统一数仓层的目标是建设一套覆盖全域、全历史的区域教育数据体系。

统一数仓层又细分为明细数据层和汇总数据层，与传统数据仓库功能基本一致，对全历史业务过程数据进行建模存储。对来源于业务系统的数据进行重新组织。业务系统是按照业务流程方便操作的方式来组织数据的，而统一数仓层从业务易理解的视角来重新组织，定义一致的指标、维度，各业务板块、业务域按照充一规范独立建设，从而形成统一规范的标准业务数据体系。

维度建模是实现统一数仓层建设目标的一种推荐建模方式，它用事实表、维度表来组织数据，这种技术已经有近30年的历史，经过大量案例考验，维度建模具备以下特点：

- 模型简单易理解： 仅有维度、事实两种类型数据，站在业务的角度组织数据。
- 性能好：维度建模使用的是可预测的标准框架，允许数据库系统和最终用户通过查询工具在数据方面生成强大的假设条件，这些数据主要在表现和性能方面起作用。
- 可扩展性好：具有非常好的可扩展性，以便容纳不可预知的新数据源和新的设计决策。可以在不改变模型粒度的情况下，很方便地增加新的分析维度和事实，不需要重载数据，也不需要为了适应新的改变而重新编码。良好的可扩展性意味着以前的所有应用都可以继续运行， 并不会产生不同的结果。
- 数据冗余：由于在构建事实表星型模式之前需要进行大量的数据预处理，因此会导致大量的数据处理工作。而且，当业务发生变化，需要重新进行维度的定义时，往往需要重新进行维度数据的预处理。而在这些预处理过程中，往往会导致大量的数据冗余。

大数据时代，数据是资产，数据应该在业务中发挥更大作用，而易理解、易用、性能好、扩展性好的模型技术能让数据更方便参与业务。随着技术的发展，存储、计算成本的降低，经常会以存储换取性能和易用性。 综上考虑，本方案推荐使用维度建模。

9.1. 统一数仓层建设过程

统一数仓层建设过程以维度建模为理论基础，构建总线矩阵，划分业务板块，定义数据域、业务过程、维度、原子指标、修饰类型、修饰词、时间周期、派生指标，进而确定维度表、事实表的模型设计。统一数仓层建设过程如下图所示。

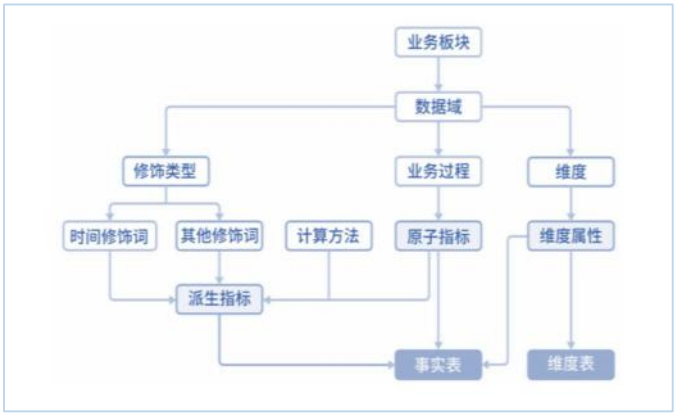

图 9-1 统一数仓层建设过程

• **业务板块：**根据业务的属性划分出的相对独立的业务板块，业务板块是一种大的划分，各业务板块中的业务重叠度极低，数据独立建设，比如学生成长、教师发展、学校发展、教育教学、综合发展等。

• **模型设计：**以建模理论为基础，基于维度建模总线架构，构建一致性的维度和事实，同时设计出一套表命名规范。

• **数据域：**数据域是统一数仓层的顶层划分，是一个较高层次的数据归类标准，是对教育业务过程进行抽象、提炼、组合的集合，面向业务分析，一个数据域对应一个宏观分析领域，比如德育域、智育域、体育域等。数据域是抽象、提炼出来的，并且不轻易变动，既能涵盖当前所有业务需求，又能在新业务进入时无影响地将其分配到已有的数据域中，只有当所有分类都不合适时才会扩展新的数据域。数据域是有效归纳、组织业务过程的方式，同时方便定位指标/度量。

• **业务过程：**业务过程是一种业务活动事件，且是业务活动过程中不可拆分的行为事件，比如典型事例填报、课题填报、教师评价过程等都是业务过程。业务过程产生度量，并且会被转换为最终的事实表中的事实。业务过程一般与事实表一一对应，也有一对多或者多对一的特殊情况，比如累计快照事实表就会把多个业务过程产生的事实在一张表中表达。

• **修饰词：**修饰词指除统计维度以外的对指标进行限定抽象的业务场景词语，修饰词隶属于一个修饰类型。比如，在日志域的访问终端类型下，有修饰词 PC、无线端。修饰类型的出现是为了方便管理、使用修饰词。

• **原子指标：**原子指标是针对某一业务事件行为的度量，是一种不可拆分的指标，具有明确业务含义，比如获奖次数。原子指标有确定的字段名称、数据类型、算法说明、所属数据域和业务过程。原子指标名称一般采用“动作+度量”方式命名，比如获奖次数、获奖等级。

• **派生指标：**派生指标可以理解为对原子指标业务统计范围的圈定。派生指标=1 个原子指标+多个修饰词+时间修饰词。

• **计算方法：**指标的数学计算方式，比如汇总、平均、最大、最小等。

• **维度表：**维度是观察事物的角度，提供某一业务过程事件所涉及的用于过滤及分类事实的描述性属性，用于描述与“谁、什么、哪里、何时、为什么、如何”（5W1H）有关的事件。维度表是统一设计的，在整个数据仓库中共享，所有数据域、业务过程都需要用到维度，都可以在公共维度表中获取相关维度属性。

• **事实表：**事实是观察事物得到的事实数据，事实涉及来自业务过程事件的度量，基本都是以数量值表示，比如一次学生身高的测量就可以理解为是一个事实，具体一个学生的身高数值就是事实信息。在确定数据域与业务过程后，就可以根据业务过程涉及的维度、度量及粒度，设计相关的事实表。事实表不跨数据域，根据需要，一个事实表可能对应同数据域下一个或者多个业务过程。事实表又分为明细事实表和汇总事实表。明细事实表记录事务层面的事实，保存的是原子数据，数据的粒度通常是每个事务一条记录，明细事实表数据被插入，数据就不再进行更改，其更新方式为增量更新。汇总事实表是把明细事实聚合形成的事实表，包括以具有规律性的、可预见的时间间隔来记录事实的周期快照事实表和以不确定的周期来记录事实的累计快照事实表。

• **粒度：**粒度是指统一数仓层数据的细化或综合程度，对各事实表行实际代表的内容给出明确的说明，用于确定某一事实表中的行表示什么。确定维度或者事实之前必须声明粒度，因为每个维度和事实都必须与定义的粒度保持一致。原子粒度是最低级别的粒度，是对业务过程最详细的刻画，原子粒度事实必须保留。

• **一致性指标定义：**指标归属到具体数据域，定义指标的含义、命名、类型、计算方法，确保指标的全局一致性。

9.1.1. 数据域划分

数据域是指面向业务或数据进行本质分析，归纳总结出来的数据集合。为保障整个体系的生命力，数据域需要抽象提炼，并且长期维护和更新，但不轻易变动。在划分数据域时，既能涵盖当前所有的业务需求，又能在新业务进入时无影响地将其插进已有的数据域中或者扩展新的数据域。具体数据域划分过程如下：

第一阶段：数据调研。

• 业务调研：确定项目要涵盖的业务领域和业务线，以及各个业务线可以细分成哪几个业务模块，各业务模块具体的业务流程是怎样的，通过跟业务专家访谈或进行资料文档收集，梳理主要业务流程、业务边界、专业术语等。

• 数据调研：调研全部数据目录信息，梳理数据流与业务过程关联关系。

第二阶段：业务分类。

• 业务过程提取：根据调研结果抽取出全部业务过程。

• 业务过程拆分：将组合型的业务过程拆分成一个个不可分割的行为事件。

• 业务过程分类：按照业务分类规则，将相似特征的业务过程分为一类，且每一个业务过程只能唯一归属于一类。

第三阶段：数据域定义。

• 业务分类确认：对业务分类结果再次确认，避免分类范围中出现业务特征明显与其他业务过程无关的情况。

• 数据域定义：根据业务分类的规律总结出划分业务范围的标准定义。

• 数据域命名：为每个数据域起一个专属名称，并附上英文全称和简称。

第四阶段：总线矩阵构建

• 关系梳理：明确每个数据域下有哪些业务过程，并梳理出业务过程与哪些维度相关。

• 矩阵构建：定义一张二维矩阵，将数据域下的业务过程与维度信息如实记录下来。

9.1.2. 指标设计

指标就是在业务运转过程中产生的度量事实，一致性指标设计是为了在使指标的命名、计算方法、业务理解达到一致，避免不同部门同一个指标的数据对不上或者对同一个指标的数据理解不一致。

一致性指标的定义为，描述原子指标、修饰词、时间周期和派生指标的含义、类型、命名、算法，被用于模型设计，是建模的基础。

一致性指标设计是事实表模型设计的来源，有了一致性的指标定义，在设计事实表模型时引用定义好的一致性指标，可达到指标的一致性与标志性。

9.1.3. 维度表设计

维度是维度建模的基础和灵魂，维度表设计的好坏直接决定了维度建模的好坏。维度表包含了事实表所记录的业务过程度量的上下文和环境，它们除了记录 5W 等信息外，通常还包含了很多描述属性字段。每个维度表都包含单一的主键列。维度设计的核心是确定维度属性，维度属性是查询约束条件

（SQLwhere 条件）、分组（SQLgroup 语句）与报表标签生成的基本来源。维度表通常有多列或者说多个属性。维度表通常比较宽，是扁平型非规范表，包含大量细粒度的文本属性。实际应用中，包含几十甚至上百个属性的维度并不少见。维度表应该尽可能包括一些有意义的文字性描述，以方便下游用户使用。维度属性尽可能丰富。维度属性设计中会有一些反规范化设计，把相关维度的属性也合并到主维度属性中，达到易用、少关联的效果。

维度表设计主要包括选择维度、确定主维表、梳理关联维表、定义维度属性等过程。

1）选择维度：维度作为维度建模的核心，在数据仓库中必须保证维度的唯一性。维度一般用于查询约束条件、分组、排序的关键属性。

2）确定主维表：主维表一般直接从业务系统同步而来，是分析事实时所需环境描述的最基础、最频繁的维度属性集合。比如学生维表从业务系统的学生基本信息表中产出。

3）梳理关联维表：数据仓库是业务源系统的数据整合，不同业务系统或者同一业务系统中的表间存在关联性。根据对业务的梳理，确定哪些表和主维表存在关联关系，并选择其中的某些表用于生成维度属性。

4）定义维度属性：从主维表或关联维表中选择维度属性或生成新的维度属性，过程中尽量生成更丰富、更通用的维度属性，并维护和描述维度属性的层次及关联关系。

9.1.4. 事实表设计

事实表是统一数仓层建设的主要产出物，统一数仓层绝大部分表都是事实表。一般来说事实表由两部分组成：一部分是由主键和外键组成的键值部分，另一部分是用来描述业务过程的事实度量。事实表的键值部分确定了事实表的粒度，事实表通过粒度和事实度量来描述业务过程。事实表的外键总是对应某个维度表的主键，实际建设过程中，为了提升事实表的易用性和性能，不仅会

存储维度主键，还会把关键的维度属性存储在实施表中。这样事实表就包含表达粒度的键值部分、事实度量及退化的维度属性。一切数据应用和分析都是围绕事实表来展开的，稳定的数据模型能大幅提高数据复用性。

事实表设计可以遵循以下步骤：

第一步：确定业务过程。

业务活动是由一个个业务过程组成的，事实表就是为了记录这些业务过程产生的事实，以便还原任意时刻的业务运转状态。所以设计事实表，第一步就是确定实施所要表达的是哪一个或者几个业务过程。在实际开发过程中，业务过程和事实表会存在多对多的关系。

第二步：定义粒度。

不管事实表对应一个还是多个业务过程，粒度必须是确定的，每个事实表都有且只能有唯一的粒度，粒度是事实表的每一行所表示的业务含义，是事实的细节级别。在实际设计过程中，粒度与主键等价，粒度更偏向业务，而主键是站在技术角度说的。定义粒度是必不可少的步骤，可避免整个事实表的业务含义模糊。

第三步：确定维度。

定义粒度之后，事实表每一行的业务含义就确定了。维度依附于粒度，是粒度的环境描述。

第四步：确定事实。

事实就是事实表度量的内容，也就是业务过程产生的事实度量的计算结果，比如成绩分数、获奖次数、身高、社会实践次数等。事实表的所有事实度量都与事实表所表达的业务过程相关，所有事实必须满足第二步所定义的粒度。

第五步：冗余维度属性。

事实表的设计要综合考虑数据来源和使用需求，在满足业务事实记录的同时也要满足使用的便利性和性能要求。为了满足业务需求，降低资源消耗，建议适当冗余维度属性数据到事实表，直接利用事实表就可以完成绝大部分业务的使用需求，这样下游使用时可减少大表关联，提升效率。

9.1.5. 模型落地实现

经过以上数据域的划分、指标的定义、维表设计、事实表设计，就完成了整个统一数仓层的设计工作。接下来要在数据开发平台结合数据平台工具，进行统一数仓层的物理层面的建设。模型结构与内容已经确定，仅仅需要代码和运维层面的实施。

落地实施的具体步骤如下：

1）按照命名规范创建表，包括维表和事实表；

2）开发生成维表和事实表的数据的逻辑代码；

3）进行代码逻辑测试，验证数据加工逻辑的正确性；

4）代码发布，加入生产调度，并配置相应的质量监控和报警机制；

5）持续任务运维监控。

9.2.基础数据层数据库

基础数据层，即 DWD（Data WareHouse Detail）数据明细层，主要是将从贴源层数据进行清洗和整合成相应的事实表。事实表作为数据仓库维度建模的核心，需要紧紧围绕着业务过程来设计。在获取教育业务系统的表结构后，进行大概的梳理，再与业务方沟通整个业务过程的流转过程，对业务的整个生命周期进行分析，明确关键的业务步骤，在能满足业务需求的前提下，尽可能设计出更通用的模型。

9.3.主题数据层数据库

基于基础数据层的相关数据，进行整合，生成高度汇聚的学生综合素质评价维度主题方向的主题域数据库。

以下以学生综合素质发展数据主题数据为例。

9.3.1. 学生成长主题库

学生成长数据模型设计依据是以上海市综合素质评价办法为指导文件。由于学生综合素质指标体系设计是一个系统的过程，且需要缜密的规划和思考，需在项目实施时进行深入探讨和完善，同时结合区、校两级数据一同构建数据主题库结构，才能使学生综合素质指标体系更加的科学。

(1) 学生成长主题数据编目

表 **错误!文档中没有指定样式的文字。**-1 学生成长主题词编目

一级指标	二级指标	基础层数据库
品德发展	爱党爱国	学生评价数据库 活动数据库 荣誉数据库
	理想信念	
	诚实守信	
	仁爱友善	
	责任义务	
	遵纪守法	
学业水平	学业水平	学业质量数据库 学生评价数据库 活动数据库 荣誉数据库
	技能掌握	
	知识运用	
	创新成果	
身心健康	健康生活方式	体质健康数据库 学生评价数据库 活动数据库 竞赛数据库 兴趣特长数据库 荣誉数据库
	体育锻炼习惯	
	身体机能	
	运动技能	
	心理素质	
艺术素养	审美感受	学生评价数据库 兴趣特长数据库 活动数据库
	艺术理解	
	艺术鉴赏	

	艺术表现	竞赛数据库 荣誉数据库
社会实践	社会生活	活动数据库 荣誉数据库 学生评价数据库
	实践活动	
	军事训练	
	生产劳动	

9.3.2. 教师发展主题库

教师发展数据仓库是针对教师专业发展设计的数据仓库，是由教师发展相关的基础汇聚层数据库构成的集合，包括对教师基础信息数据库、教师发展数据库、教学课程资源数据库、教学研究数据库等相关的数据库的高度汇总。

9.3.3. 教学管理主题库

教学管理主题库包括教师的教学准备、教学互动、教学教研、教学质量等方面的数据，数据来源可以包括教学计划、课程设计、备课数据、师生课堂行为数据、作业数据、课程资源、评价反馈、研修数据、日常教研、听评课数据、教育科研数据、考试成绩、教学质量分析等方面的数据的高度汇总。

9.3.4. 综合管理主题库

综合管理数据主题仓库是针对教育管理设计的，是由教育管理相关的数据库构成的集合，包括教务管理、教师管理、学生管理、安全管理、资产管理等相关的数据库的高度汇总。

9.3.5. 学校发展主题库

学校发展数据主题仓库是针对学校教育事业发展设计的，是由教育事业发展相关的数据库构成的集合，包括学校基本情况、班数班额情况、学生数量、学生变动情况、学生死亡数量、休退学情况、外籍学生、教职工构成、教职工专业技术职级及年龄分布、教职工学历学科情况、特殊教育专人教师情况、教师接受培训情况、校舍情况、办学条件统计数据、信息化建设情况、借读情况、外省市毕业招生情况等相关数据库的高度汇总。

9.4.主题数据层数据结构设计

区域教育数据仓库的主题数据域是统一数仓层的顶层划分，是一个较高层次的数据归类标准，是对教育业务过程进行抽象、提炼、组合的集合，面向业务分析。一个数据域对应一个宏观分析领域，比如学生德育主题、智育主题、体育主题、美育主题、劳育主题等。数据域是抽象、提炼出来的，并且不轻易变动，既能涵盖当前所有业务需求，又能在新业务进入时无影响地将其分配到已有的数据域中，只有当所有分类都不合适时才会扩展新的数据域。数据域是有效归纳、组织业务过程的方式，同时方便定位指标/度量。

9.5.主题数据层模型设计

数据模型是数据构架中重要一部分，包括概念数据模型和逻辑数据模型，是数据治理的关键、重点。理想的数据模型应该具有非冗余、稳定、一致、易用等特征。逻辑数据模型能涵盖整个系统的业务范围，以一种清晰的表达方式记录跟踪重要数据元素及其变动，并利用它们之间各种可能的限制条件和关系来表达重要的业务规则。数据模型必须在设计过程中保持统一的业务定义。为了满足将来不同的应用分析需要，逻辑数据模型的设计应该能够支持最小粒度的详细数据的存储，以支持各种可能的分析查询。同时保障逻辑数据模型能够最大程度上减少冗余，并保障结构具有足够的灵活性和扩展性。

下图为学生五育综评主题数据模型样例：

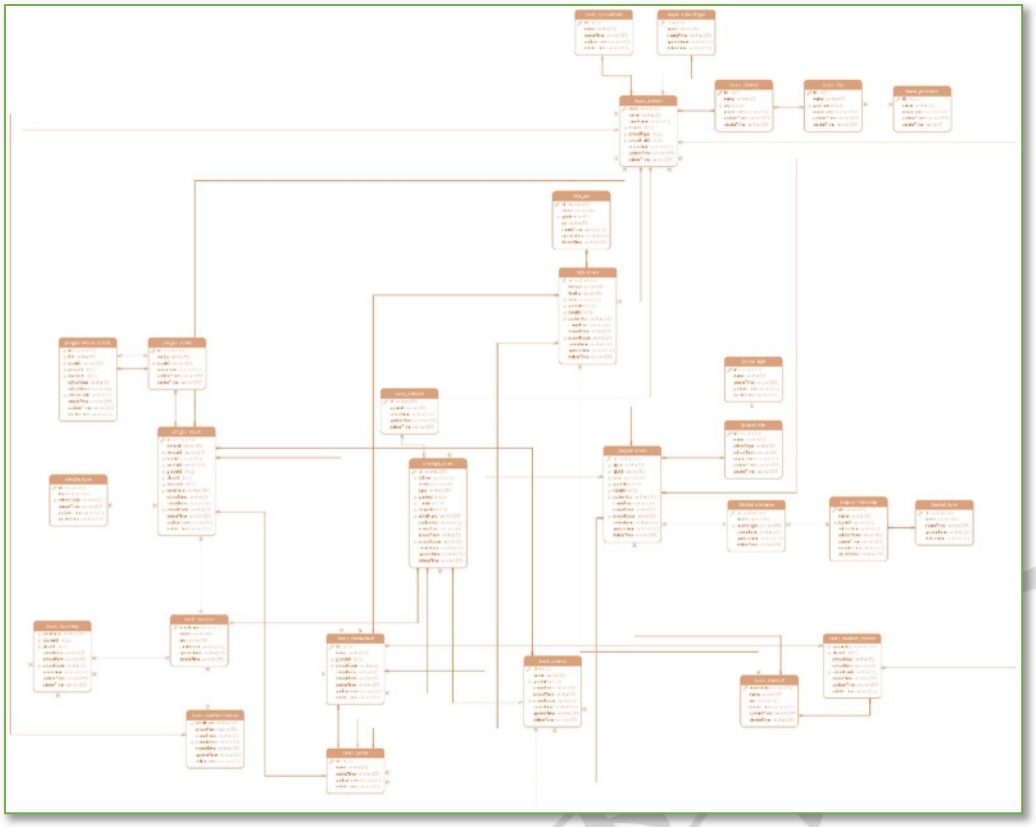

图 9-2 数据雪花模型

10. 标签数据层建设

统一数仓层是按照数仓的维度规范建模，对业务数据进行了重新组织标准化。但是同一个对象的各种信息分散在不同的数据域并且有不同的数据粒度，这导致很难了解一个学生的全面信息，要通过各种关联计算才能满足业务的需要，数据使用成本较高。而获取、分析学生的全面数据，是多个业务的共同需求，这可以通过建设标签数据层来满足。大数据的典型应用基本都是通过建立标签体系来支撑的。

标签数据层是面向对象建模，把一个对象各种标识打通归一，把跨业务板块、数据域的对象数据在同一个粒度基础上组织起来打到对象上。标签数据层建设，一方面让数据变得可阅读、易理解，方便业务使用；另一方面通过标签

类目体系将标签组织排布，以一种适用性更好的组织方式来匹配未来变化的业务场景需求。

标签归属到一个对象，标签按照产生和计算方式不同可分为属性标签、统计标签、算法标签。对象本身的性质就是属性标签。对象在业务过程事件中产生原子指标，原子指标与修饰词、计算方法可以组装出统计标签。对象在多个业务过程的规律特征通过一定的计算方法可以计算出算法标签。另外对象在特定的业务过程会与其他对象关联，关联对象的属性也可以作为标签打在主对象上。对象的属性标签、统计标签、算法标签与对象标签类目、对象标识组装起来就生成对象标签表。对象标签表没有严格的维度与事实的区分，对象标签表为标签融合表。

下图所示为标签融合表的建设过程：

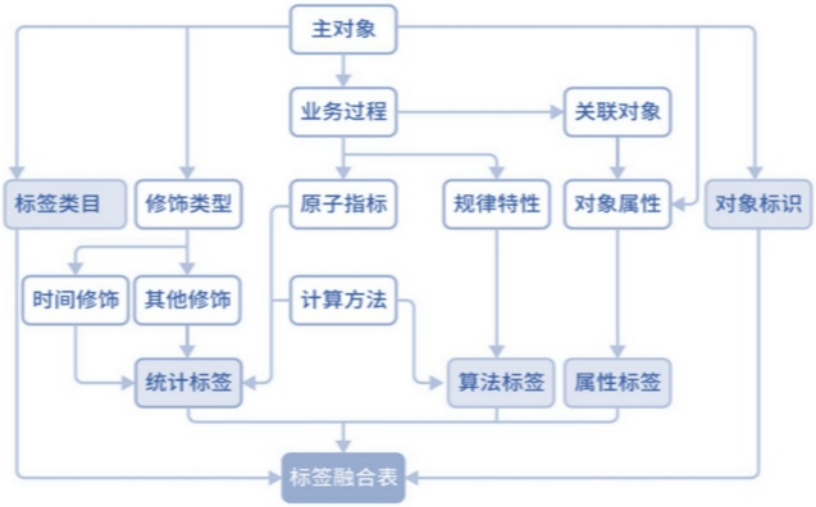

图 10-1 标签融合表建设过程

11. 应用数据层建设

统一数仓层和标签数据层数据相对稳定，然而最终用户的需求和使用方式是千变万化的，统一数仓层和标签数据层无法灵活适应各类用户的使用需求。另外最终用户使用数据也需要灵活性和高性能，而这与规范是矛盾的，因为按

规范进行建设就会把数据按照各种域、业务过程、维度、粒度等拆分，使用的时候需要各种连接，这样就无法满足对灵活、高性能的要求。为了解决规范稳定与灵活、高性能之间的矛盾，要增加应用数据层。

11.1. 应用数据层的概念

应用数据层是按照业务使用的需要，组织已经加工好的数据以及一些面向业务的特定个性化指标加工，以满足最终业务应用的场景。应用数据层一般也是采用维度建模的方法，但是为了满足业务的个性需求以及性能的要求， 会有一些反规范化的操作，所以应用数据层并没有非常规范的建设标准。

应用数据层类似于传统的数据集市，但是比数据集市更轻量化、更灵活，用于解决特定的业务问题。应用数据层整体而言是构建在统一数仓层与标签数据层之上的简单数据组装层，不像数据集市那样要为某个特定的业务独立构建，应用数据层的构建和完善是从企业级多个类似业务场景来考虑的，同时它又具备数据集市灵活响应业务需求的特点。

11.2. 应用数据表设计

应用数据层是在统一数仓层、标签数据层都已经建设好的基础上，面向特定业务需求而准备的个性数据组装层，除了特殊的业务个性标签需要单独加工外，其他尽可能复用统一数仓层和标签数据层的建设成果。

应用数据层的建设是强业务驱动的，业务部门需要参与到应用数据层的建设中来。推荐的工作方式是，业务部门的业务专家把业务需求告知数据工程师，然后在建模过程中深入沟通，这样最终形成的应用数据层的表设计才能既满足业务需求又符合整体的规范。因此应用数据层的特点就是考虑使用场景，其有以下几种结构：

1）应用场景是多维的即席分析，一般为了减少连接、提升性能，会采用大宽表的形式组织。

2）如果是特定指标的查询，可以采用 K-V 表形式组织，涉及此类表的时候需要深入了解具体的查询场景，例如是否有模糊查询，以便于选择最适合的数据结构。

3）有些场景下一次要查询多种信息，也可能会用复杂数据结构组织。

11.3. 应用数据表实现步骤

应用数据层建设步骤如下：

1）调研业务应用对数据内容、使用方式、性能的要求，需要明确业务应用需要哪些数据，数据是怎么交互的，对于请求的响应速度和吞吐量等有什么期望。

2）盘点现有统一数仓层、标签数据层数据是否满足业务数据需求，如果满足则直接跳到第 3 步；如果有个性化指标需求，统一数仓层、标签数据层数据无法满足，则进行个性化数据加工。

3）组装应用层数据。组装考虑性能和使用方式，比如应用层是多维的自由聚合分析，那就把统一数仓层、标签数据层以及个性化加工的指标组装成大宽表；如果是特定指标的查询，可以考虑组装成 K-V 结构数据。

11.4. 应用数据的存储策略

随着数据技术的发展，数据应用场景已经不限于做 BI 分析出报表，而是在更广的业务领域发挥价值。这些不同的使用场景，对数据的组织方式和底层的存储计算技术的要求是不同的，应用数据层的模型设计要考虑业务需要和技术环境。应用数据层加工的结果数据集，要根据不同的使用场景，同步到不同的存储介质，以达到业务对不同吞吐量和响应时间的需要，如下图所示。

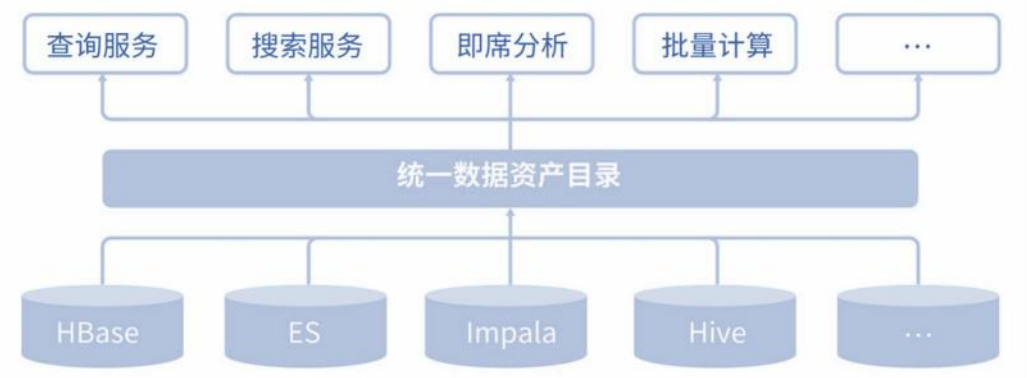

图 11-1 一套数据多套存储计算环境

比如交叉分析和特定指标查询，所有数据都是数据工程师、算法工程师在数据平台中加工而成的，一般采用分布式离线加工，加工的结果存放在分布式文件中。但是交叉分析和指标查询都需要毫秒级的快速响应，大数据存储层计算环境无法满足这样的低延迟要求，这就需要把加工好的数据同步到可以满足的环境介质中。所以交叉分析一般同步到具备高吞吐、低延迟的即席分析环境，比如 Greenplum、Impala；指标查询一般同步到 K-V 数据库，比如 HBase。这样就达到一套数据多套存储，以满足业务对于性能的要求。

12. 数据开放服务（共享及交换）系统

通过综评数据中台的数据开放服务（共享及交换）系统，为市级业务部门、学校、区教育局开放数据查询、数据分析、算法等共享服务接口和数据交换，以便他们能够更方便、更深入地使用数据。

数据开放服务系统通过对数据进行计算逻辑的封装(过滤查询、多维分析和算法推理等计算逻辑)，生成 API 服务，上层数据应用可以对接数据服务 API，让数据快速应用到业务场景中。在数据中台落地支撑业务时，数据分析师或算法工程师通过数据服务配置中台数据资产的访问 API，数据应用产品即可以方便地使用中台的数据能力，支撑业务决策和智能创新。

开放共享的数据范围包括基础层数据和主题层数据。

基础层数据包括：基础数据库、学生评价数据库、五项管理数据库、学业质量数据库、荣誉数据库、体质健康数据库、兴趣特长数据库、活动数据库、竞赛数据库等。

主题层数据包括：品德发展（品德发展与公民素养）、学业水平（修习课程与学业成绩）、身心健康（身心健康与艺术素养）、艺术素养（身心健康与艺术素养）、社会实践（创新精神与实践能力）等主题数据。

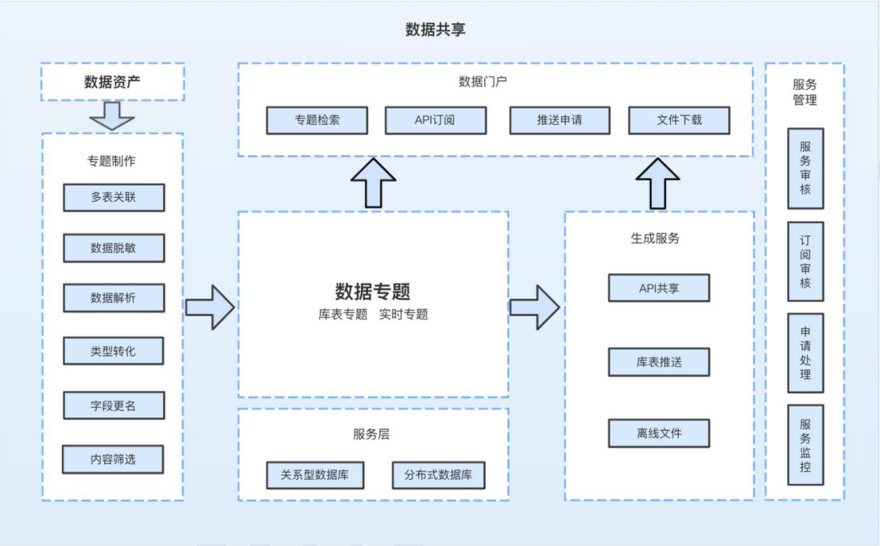

图 12-1 数据共享架构图

12.1. 数据服务类型

按照数据与计算逻辑封装方式的不同，数据服务可分为以下三类：

- **基础数据服务：**它面向的对象是基础层数据，主要面向的场景包括数据查询、多维分析等，通过自定义 SQL 的方式实现数据中台全域基础层数据的指标获取和分析。
- **标签画像服务：**它面向的对象是主题层、标签数据层和应用数据层的数据，主要面向的场景包括标签圈人、画像分析等，通过界面配置方式实现数据中台全域高度汇聚层数据跨计算、存储的统一查询分析计算，加快数据应用的开发速度。

- **算法模型服务：**它面向的对象是算法模型，主要面向的场景包括智能分析、个性化推荐和风险监控等，主要通过界面配置方式将算法模型一键部署为在线 API，支撑智能应用和业务。

12.2. 数据服务专题管理

12.2.1. 专题生成

用于生成数据专题，数据来源于数据资产中的基础层和主题层，通过各类业务规则的加工最终生成一个专题数据对外提供数据共享及交换服务。

专题制作支持多种界面化操作，例如多表关联设置，修改字段名称、修改字段类型、设置字段脱敏规则、设置字段解析规则、设置筛选器等。支持设置表信息，添加专题表元数据规则。支持设置共享信息，设置表数据权限、数据落地、数据更新策略等内容，数据权限设置为开放的专题数据可以在数据门户上对外展示，以及发布数据服务。数据专题会生成落地任务可配置各类执行策略，可一键上线至任务运维，在任务运维中查看和管理。

专题制作还支持将界面配置好的规则一键转化为 SQL 脚本，在已经形成的 sql 语句基础上再次进行更复杂的 sql 语句编辑，贴合更多的数据业务场景。

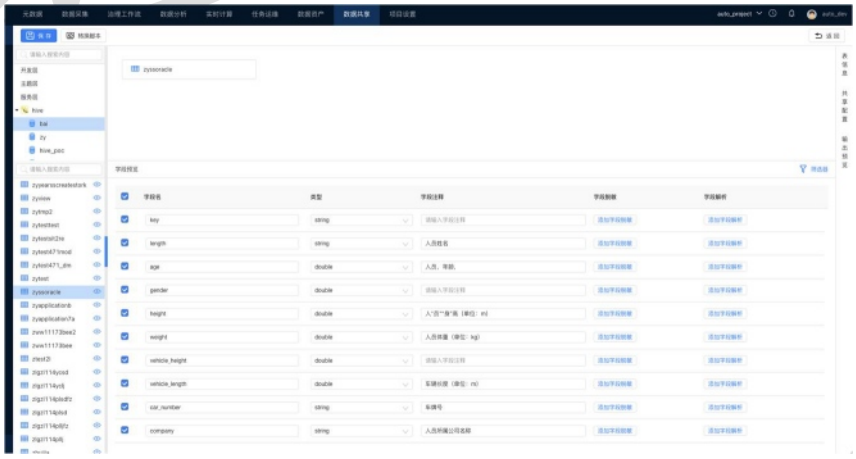

图 12-2 数据共享-专题生成

12.2.2. API 服务

数据专题支持发布 API 服务。支持设置默认查询条数查询，如果在接口调用时不指定返回条数，则按默认值返回数据条数。支持公开和受限两种数据权限，如果数据权限为公开，任意用户可以直接使用 API 服务调用数据；如果数据权限为受限，则用户需要先申请认证，获得私钥或服务令牌，才可以调用 API 服务获取数据（支持 token 和私钥两种认证方式）。

API 调用时支持指定数据返回条数，支持通过参数选取任意字段设置数据筛选条件，对返回的数据进行筛选。支持对私钥和服务令牌进行核对校验，保证数据不外泄，并且私钥和服务令牌可以设置有效期，通过调节有限期的时限达到保证数据安全性的目的。

API 调用支持通过 https 方式访问，保证数据传输的安全性。

API 服务支持开启黑/白名单，通过编辑黑/白名单中的 IP 达到控制 IP 访问服务的目的。

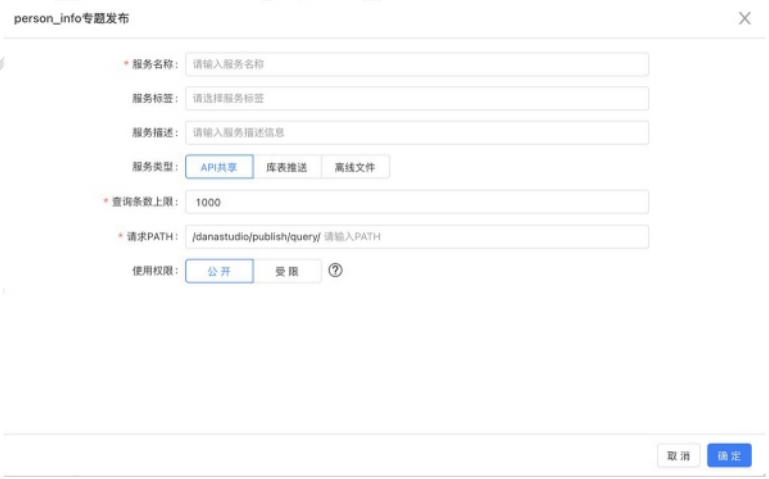

图 12-3 数据共享-发布服务-API

12.2.3. 库表推送

库表推送服务是指把专题数据推送到指定数据源的表中，支持 oracle、postgresql、mysql 类型的数据库的数据推送，支持推送至数据库中的已有表（包括分区表）和在数据库中新建一个对应的数据表进行数据推送。支持字段隐射关系设置、支持字段类型设置、支持字段名称设置。支持设置数据策略（覆盖、追加）和更新策略（全量、增量）来控制库表推送任务的执行。

库表推送服务需要预先设置一个执行策略，当服务通过审核时，库表推送服务支持以预先设置好的执行策略上线至任务运维的周期作业中进行调度。

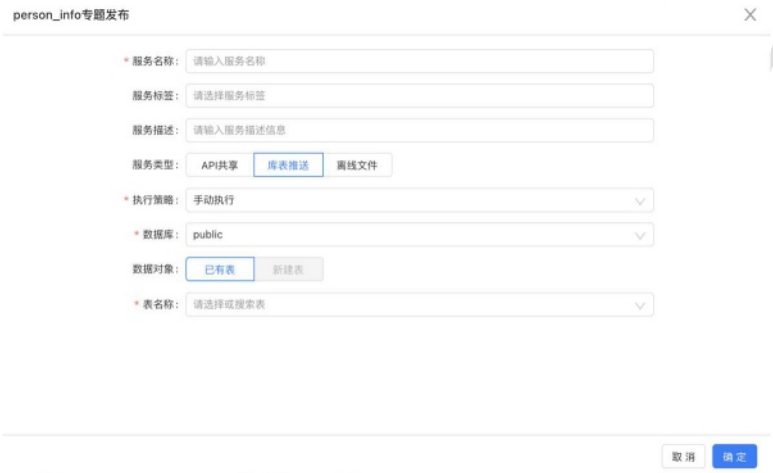

图 12-4 数据共享-发布服务-库表推送

12.2.4. 离线文件

离线文件服务是指将专题数据生成 Excel 或 CSV 格式的文件，提供给用户下载离线使用。

文件类型：

- Excel 格式的文件适用于小数据量的文件使用场景，支持更多样的场景功能。例如使用 Excel 做报表分析等。

- CSV 格式的文件适用于大数据量文件使用场景，支持大数量的数据迁移。例如不同网络之间的数据传输转移。

生成的文件支持基于专题数据字段做内容筛选，例如基于时间字段筛选去年一整年的产生的数据。支持多种条件的组合筛选，通过满足任一条件和满足所有条件定义组合筛选的与逻辑和或逻辑。支持勾选需要生成文件的字段。

图 12-5 数据共享-发布服务-离线文件

Excel 格式的文件支持设置文件加密，支持用户自己指定密码或系统随机生成密码，降低了文件内容被窃取的风险。支持对文件添加明水印（打开可见）或暗水印（文件属性，可通过系统回溯功能得出），当文件被泄露时可追踪泄露环节。

图 12-6 数据共享-服务审核-离线文件

12.3. 数据服务发布管理

12.3.1. 服务审核

服务审核界面支持对数据服务进行服务管理：详情查看和服务审核。

服务审核功能可以审核状态为未审核的服务，通过审核的数据服务变为已审核状态，意味着该服务发布成功，可在数据门户界面被用户订阅、申请和下载。未通过审核的服务标记为打回状态，并且附带评审意见，回到服务发布界面，对服务作出修改后变为未审核状态，再次进入审核流程。为了避免用户使用过程中的问题，审核通过的服务不支持编辑和修改。

审核 API 服务的时候可以对 API 服务的信息进行审核，并对 API 的权限进行调整。

图 12-7 发布管理-服务审核-API

审核库表推送服务的时候支持修改库表推送任务的执行执行策略。

图 12-8 发布管理-服务审核-库表推送

审核离线文件服务后，系统在后台生成离线文件，生成结束后界面提示生成成功，文件可在审核列表中下载查看。

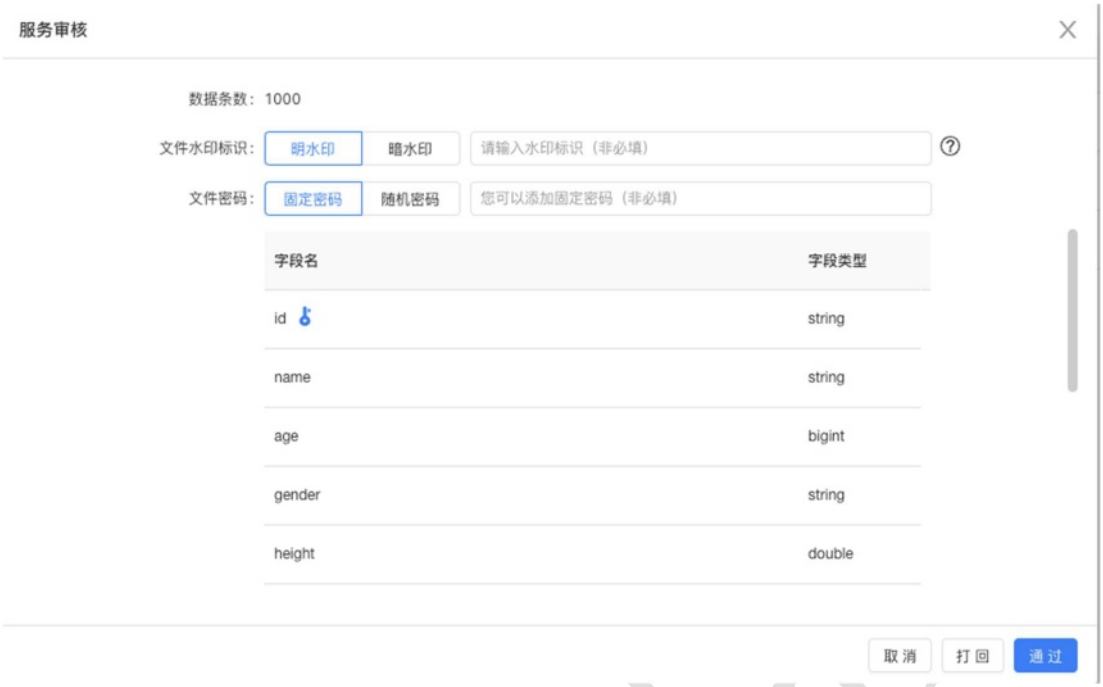

图 12-9 发布管理-服务审核-离线文件

12.3.2. 服务监控

服务监控可以监控通过审核的数据服务被使用的情况。

API 服务可以查看提供服务数量、服务的使用量，服务列表以及每个服务的具体调用信息。调用信息中可以看到调用 IP，调用时间，数据传输速率等信息。

库表推送可以查看库表推送任务执行结果汇总情况，如推送数据量和任务执行结果，具体日志信息可以到任务运维查看。

离线文件可查看离线服务的今日下载量、本周下载量、总下载量数据。

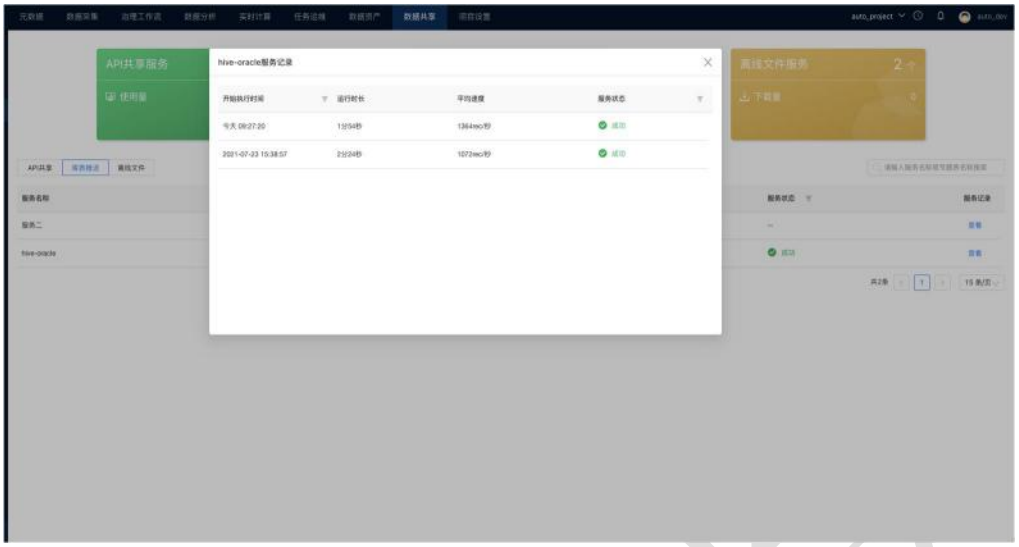

图 12-10 发布管理-服务监控

12.3.3. 流量管控

流量管控界面用于管理流量管控策略，支持流量管控策略的增加、修改、删除、服务列表管理以及服务流量监控。

流量管控策略主要针对 API 服务的流量管控。支持在单位时间内 API 级别和 IP 级别的访问次数限制的设置。支持绑定 API 服务以及对绑定的 API 服务列表进行管理。支持对绑定了流量控制策略的 API 服务访问流量进行监控，实时获取单位时间内单个 API 或单个 IP 访问次数，并对快要达到上限的服务进行预警。当 API 服务访问达到上限时，系统将阻拦访问 API 的请求，保证系统运行的安全性。

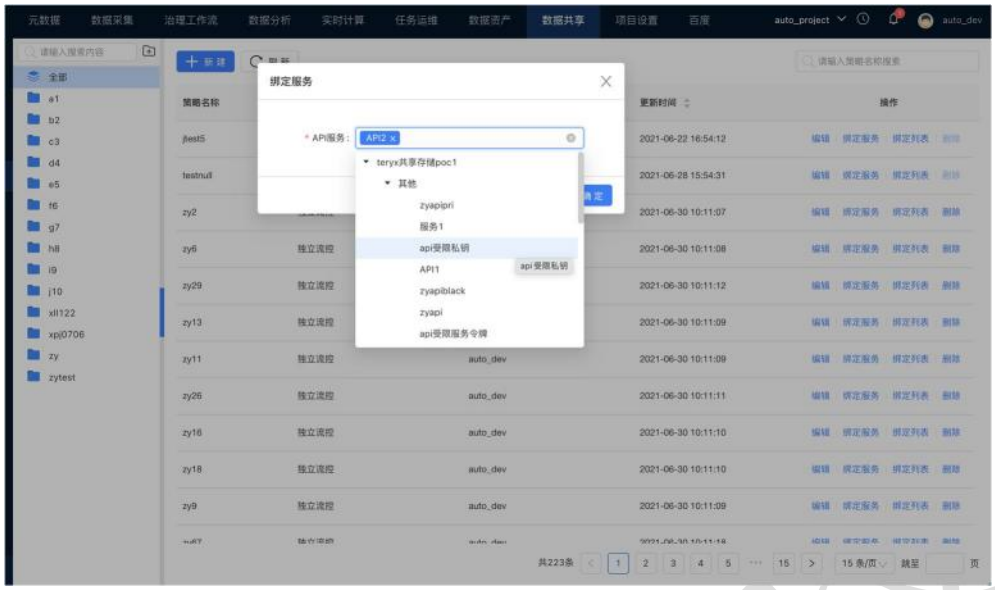

图 12-11 发布管理-流量控制

13. 学生成长画像

利用数据中心的学生成长数据以及数据中心的数据计算、数据标签及数据服务能力，通过与学生成长经历对比的方式，呈现学生品德发展、学业发展、身心健康、艺术素养、综合实践五大维度的自我成长情况，跨每个学期、每个学年、每个学段进行数据对比呈现，让学生从每一个成长经历中查看到个人进步带来的喜悦感和收获感，从而增强学习兴趣和学习自信心。同时通过精美的展现方式，也是学校教育留给学生回忆成长过程的一份礼物。对于每一个学段的教师来说，能通过学生成长时光集，更了解学生的成长历程、更懂学生的个性特长和全面发展过程，从而更有针对性给予学生适宜的教育教学。

13.1. 学生个体数字画像

学生成长画像提供小学版、初中版、高中版、职校版四大学生数字画像版本，让学生能够全方位了解自身成长历程。高中版包含该学生小学、初中和高中三个阶段的成长画像分析，职校版包含学生小学、初中和职校三个阶段的成

长画像，初中版包括学生小学和初中两个阶段的成长画像。学生个体画像以当前所处学段为主，可以呈现给学生和家长。

通过调研数据现状和实际业务需求，参考教育部义务教育质量评价指标，上海市初中、高中、职校综合素质评价规范及维度，参考国家、教育部学生综评规范要求设置学生画像维度，构建学生个体画像模型。

学生个体数字画像主要包括：

(1) 学生个体当前状态数字画像

学生基本属性以及综评各维度的当前状态，支持数据下钻进行深度分析；支持与同班、同校同年级、同区同年级的平均水平对比。

图 13-1 学生个体画像

(2) 学生当前学段成长趋势分析

呈现学生个体数字画像各维度在当前学段不同时期（可按学期、学年）的变化情况，并支持下钻深度分析，支持与同班、同校同年级、同区同年级在同一时期的平均水平对比。

(3) 学生个体生涯成长趋势分析

呈现学生个体数字画像各维度在历史各学段不同时期（可按学期、学年）的变化情况，并支持下钻深度分析，支持与同班、同校同年级、同区同年级在同一时期的平均水平对比。

13.2. 学生群体数字画像

学生群体数字画像是按照学段、学校、年级、班级等不同的群体进行画像分析、对比，并进行趋势分析。

(1) 学生群体当前状态数字画像

学生群体当前状态数字是基于学生综合素质数据，根据研究的特定对象学生群体属性信息（按学校、学段、年级、班级等），生成不同群体的学生群体数字画像。

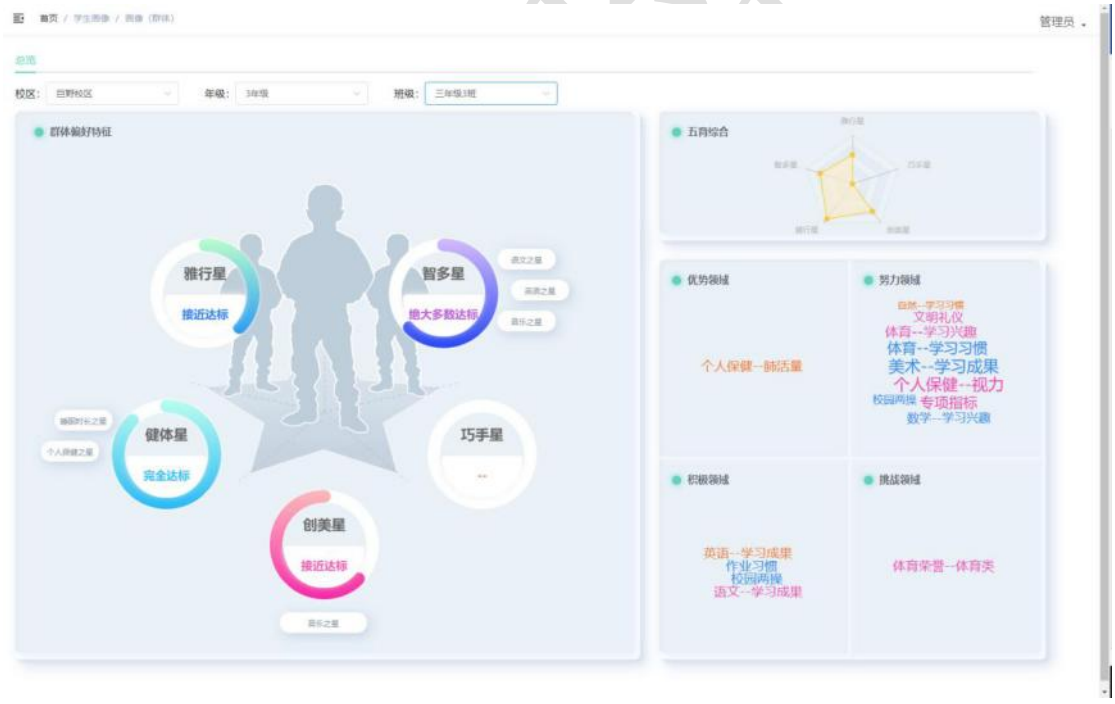

图 13-2 学生群体画像

(2) 学生群体对比分析

同一类别群体（按学校、学段、年级、班级等）各维度与平均水平对比。比如，某初级中学与本区全体初级中学的对比，某学校某年级与全区同一年级学生群体的对比等，并支持下钻深度分析。

图 13-3 学生群体对比分析

(3) 学生群体趋势分析

对一个特定群体（按学校、学段、年级、班级等）数字画像各维度不同时期（可按学期、学年）的变化情况进行分析。