

# 数栈

## 产品介绍

2018 年 6 月

# 简介

数栈的定位是一站式数据中式数据中台 PaaS<sup>[1]</sup>，目标是通过产品化的方式，帮助企业构建数据共享能力中心。数栈覆盖了建设数据中台过程中所需要的各种工具，完整覆盖离线计算、实时计算应用，满足开发人员从数据同步、数据分析、数据挖掘、数据质量、数据地图、数据模型、数据 API 的各层次应用，使用数栈可以解放开发人员的生产力，极大的缩短数据价值的萃取过程，提高企业提炼数据价值的能力。

数栈产品主要包括 3 大产品线条：数据开发套件、数据治理套件、数据应用引擎，如下图所示：



Figure 1 数栈产品功能模块图

每个产品的定位如下：

**开发套件：**一站式大数据开发平台，帮助企业快速完全数据中台搭建。

**数据质量：**对多种异构数据源进行质量校验，帮助企业提升数据健康度。

**数据地图：**可视化的数据资产中心，帮助企业全盘掌控数据来源去向。

**数据模型：**使企业数据规范化，标准化，模型化，帮助企业实现数据管理规范化。

**数据 API：**快速生成数据 API、统一管理 API 服务，帮助企业提高数据共享效率。

<sup>1</sup>PaaS：平台即服务（Platform as a Service)的简称。

# 为什么选择袋鼠云数栈

## 一站式数据中台 PaaS<sup>[2]</sup>

数栈覆盖了建设数据中台过程中所需要的各种工具, 完整覆盖离线计算、实时计算应用, 满足开发人员从数据同步、数据分析、数据挖掘、数据质量、数据地图、数据模型、数据 API 的各层次应用, 使用数栈可以解放开发人员的生产力, 极大的缩短数据价值的萃取过程, 提高企业提炼数据价值的能力。

### 一站式

一站式数据开发产品体系, 数据开发套件+数据治理套件+数据应用引擎, 覆盖数据采集、数据分析、数据挖掘、任务运维、数据质量、数据地图、数据模型、数据 API 开放等场景, 充分满足企业建设数据中台过程中的多样复杂需求。

### 兼容性强

经过了 10 多年的蓬勃发展, 很多企业都已经采购了商用的大数据平台, 数栈·开发套件 (IDE) 模块可以兼容企业已建设的大数据平台 (Cloudera、星环等)。

### 开箱即用

基于 WEB 的图形化操作界面, 开箱即用, 快速上手, 屏蔽底层复杂的基础组件, 大幅降低学习成本, 从入门到熟练开发仅需几天时间。

### 性价比高

最小仅需 3 个计算节点, 虚拟机可部署, 满足中小企业数据中台的建设需求, 降低企业投入成本。

## 开发套件 (DTinsightIDE)

企业搭建数据平台的最终目的是为了满足越来越多的业务诉求, 在搭建数据平台或为平台选择配套工具的过程中, 通常的思考角度是底层平台的高性能、稳定性, 而开发人员的使用便捷性可能通常会被忽略。

Figure 2 是典型的数据处理链路, 业务上急需一个统计指标来辅助决策, 但技术人员在编写指标计算的逻辑的时间通常只占 20%, 其余时间都是用在数据导入/导出、编写调度命

---

2: PaaS 是平台即服务 (Platform as a Service)的简称。

令/脚本、寻找出错的任务、开发服务接口、排查数据质量问题等等，正是这些看似不起眼的“细枝末节”导致需求的响应速度慢，因此，加快响应速度的根本就在于加快这个处理链路。

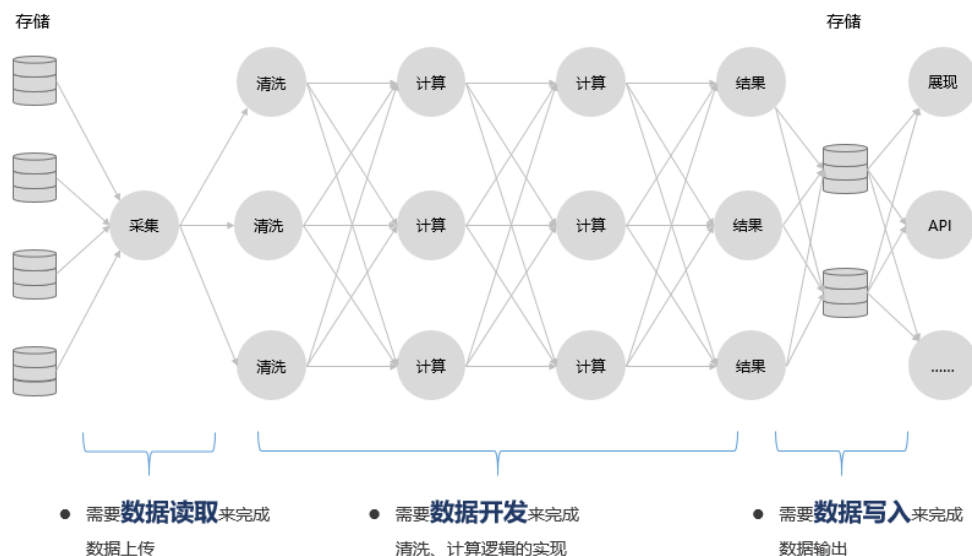


Figure 2 典型的数据处理链路

DTinsightIDE 是一款高效的大数据实时/离线任务开发、任务调度、数据管理工具，支持对大数据实时处理过程进行可视化管理与控制。帮助客户提升开发效率，快速创建实时/离线计算任务，缩短开发周期；任务管理与运维一体化，减轻繁冗的运维工作。产品的功能架构如下图所示：

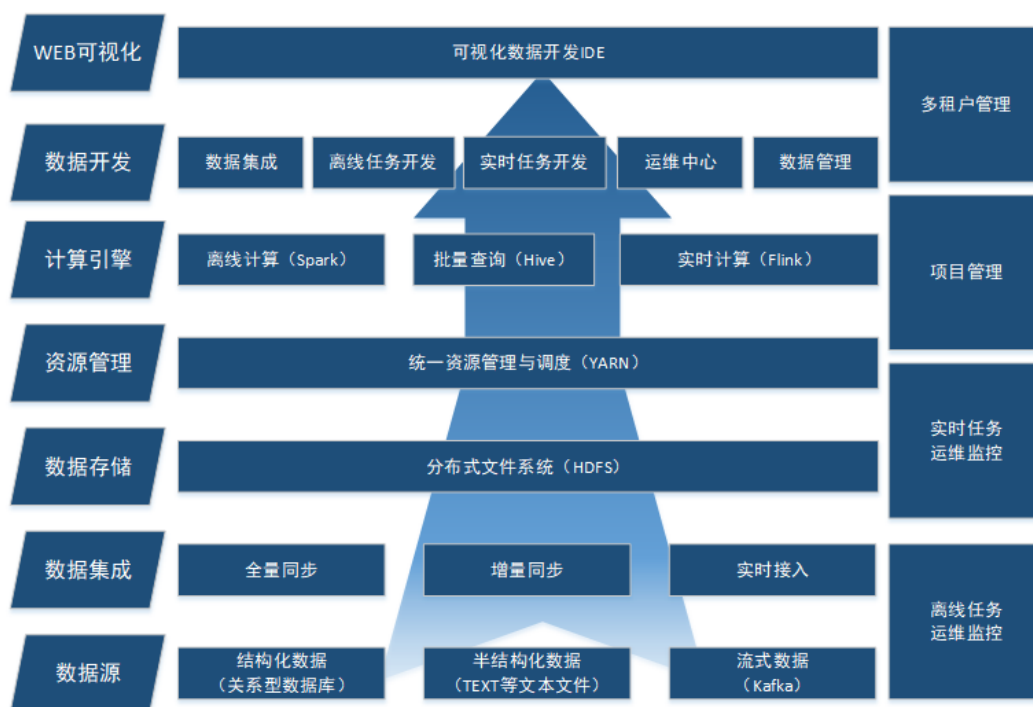


Figure 3 功能架构图

## 主要特性

- 易用

经过阿里上千位数据工程师十余年的打磨，一个平台完成数据抽取、分析、写入、运维，相比传统的手工模式，可节省 90% 的开发时间。

- 兼容

经过了 10 多年的蓬勃发展，很多企业都已经采购了商用的大数据平台，DTinsightIDE 可以兼容企业已建设的大数据平台（Cloudera、星环等），用户无需更换底层平台，即可体验到顺畅的开发体验，实现快速的数据同步、数据开发、任务运维，缩短需求的响应周期。

- 统一

同时兼容实时与离线 2 类开发工作，覆盖采集、分析、计算、运维全链路，为各类上层应用提供统一数据支撑。

## 数据同步：数据交换的管道

数据同步模块是在各个存储单元之间执行数据交换的管道。为了在 DTinsightIDE 进行大规模数据集的挖掘与计算，通常的做法是在任务执行前将数据传输至 DTinsightIDE，并在任务执行结束后将计算结果传输至外部存储单元（例如 MySQL 等应用数据库）。数据集成的作用如下图所示：

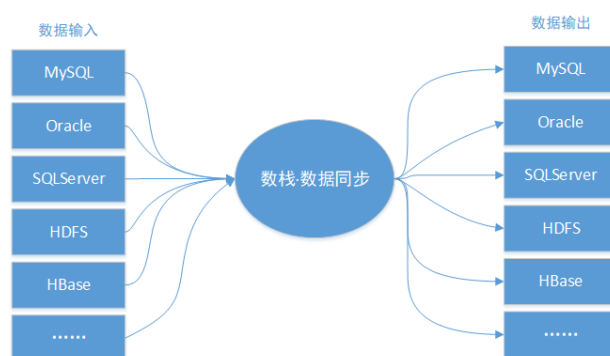


Figure 4 数据同步模块的作用

数据同步模块的特性包括以下几方面：

- 丰富的数据源支持

数据同步模块可对 MySQL、Oracle、SQLServer、HDFS、Hive、HBase、FTP、ElasticSearch、

ODPS 等数据源，支持对这些数据源进行读取或写入数据。使用时仅需配置数据源的连接信息（例如填写 Oracle 数据库的 JDBC URL、用户名、密码等信息），再配置对应的数据同步任务即可。

### ● 分布式系统架构

数据同步模块在系统架构上采用先进的分布式系统架构（FlinkX<sup>3</sup>），可实现多个节点并发读取、写入数据，可极大的提升数据同步的吞吐量，相比 Sqoop、Kettle 等开源数据同步方案，数据吞吐能力更高、配套功能更完善。

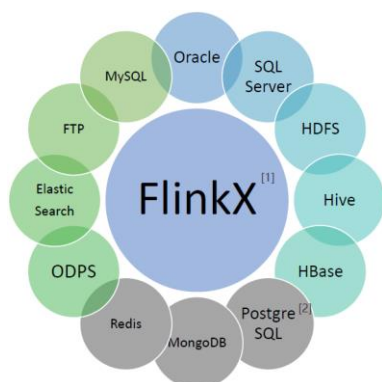


Figure 5 支持的数据源

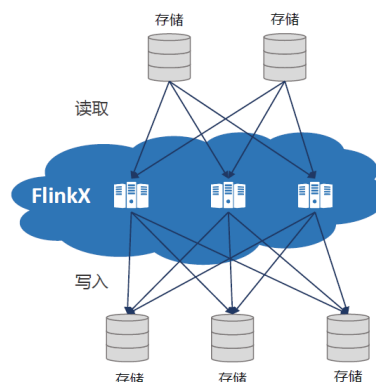


Figure 6 分布式系统架构

### ● 可视化配置

用户在使用数据同步模块时，可快速通过可视化配置的方式完成同步任务的创建与配置，主要包括同步任务选择源库源表、目标库目标表、配置字段映射、配置同步速度等。

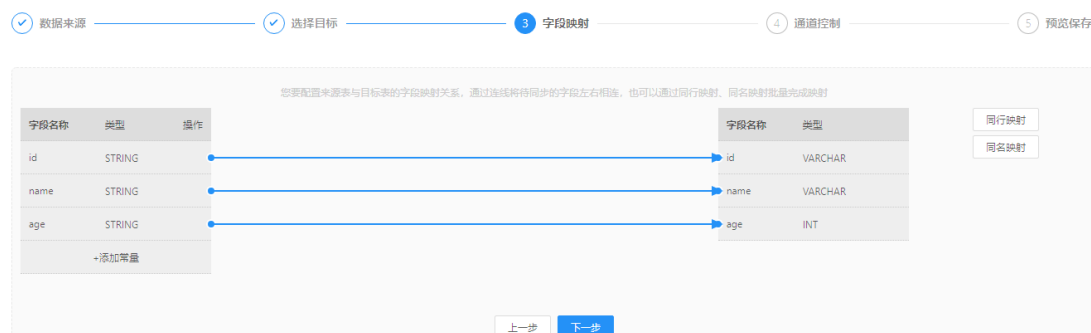


Figure 7 可视化配置-字段映射的配置

### ● 调度与依赖的配置

在实际的数据生产过程中，数据同步任务通常是数据处理链路的第一个任务和最后一个任务，分别承担“从业务系统抽取数据”和“将结果数据写出”的职责。DTinsightIDE 支持对同

<sup>3</sup> FlinkX：基于 Apache Flink 实现的分布式数据同步引擎，FlinkX 源码已由数栈研发团队贡献给开源社区，详情请参考：<https://github.com/DTStack/flinkx>

步任务配置依赖关系，约束同步任务与其他任务的执行先后顺序。

数据同步任务通常是周期执行的，每天、每周、每小时或每分钟执行一次，DTinsightIDE 支持对同步任务配置循环周期，实现同步任务的定期执行，详细的调度与依赖配置功能请参考[错误!未找到引用源。一节](#)。

### ● 全量/增量同步

从业务系统读取数据的过程中，为了最小化对业务系统的影响，通常需要进行数据的增量同步。在源数据库表中具备数据变更时间字段的情况下，DTinsightIDE 支持对关系型数据库进行增量数据同步，用户仅需输入相应的数据过滤语句即可实现。



Figure 8 增量同步的过滤语句

### ● 同步速度的控制

数据进行初始化的同步时，往往有大量历史数据需要同步至中台，需要提高数据读取的速度，当业务数据库的运行压力较大时，为了减轻数据库的压力，需要降低数据读写的速度。

DTinsightIDE 支持数据同步速度控制，通过设置同步速率上限来调整，此参数需根据硬件配置和数据量来调整，用户根据业务需求选择设定的值。



Figure 9 同步速度控制

### ● 脏数据管理

在数据同步的写入阶段可能因为数据格式转换、字段值异常等各类原因造成数据无法写入，通常的处理方案是直接丢弃无法写入的数据，但很多业务场景下这类数据其实是有意义的，并可以经过特殊处理使其变为“正常”数据。

DTinsightIDE 支持对脏数据是否需要记录进行配置，可指定脏数据的存储表名、生命周期，同时可配置当脏数据量超过一定数量或一定比例时任务置为失败，提示用户及时排查脏

数据问题。在此基础上，DTinsightIDE 同时支持脏数据的统计，包括每个同步任务的脏数据产生趋势，脏数据产生较多的任务，并自动对脏数据产生原因进行分析。

错误记录管理: ☒ 记录保存

脏数据写入hive表: 默认系统分配

\* 脏数据存储生命周期: 90天

错误记录数超过: 100 条, 任务自动结束

错误记录比例配置: 任务执行结束后, 统计错误记录占比, 大于 %时, 任务量为失败

Figure 10 脏数据的配置

## 数据开发：构建数据分析逻辑

在 Figure 2 典型的数据处理链路中，数据同步仅仅完成了数据的读取和写入，下一步就是要构建数据清洗、统计或数据挖掘的计算逻辑。DTinsightIDE 具备 3 大特性来协助用户完成这一过程：

- 广泛的任务类型支持

而企业内进行数据分析的场景多种多样，周期执行的任务、临时取数、数据挖掘任务都会同时存在，DTinsightIDE 提供 6 种不同的任务模式，分别满足不同分析场景：

会同时存在，DTinsightIDE 提供多种不同的任务模式，分别满足不同分析场景：

- SparkSQL：绝大多数任务为 SQL 任务，满足周期性数据处理场景，例如数据清洗、数据统计&分析、简单分析模型等。
- 数据同步：在不同存储单元之间进行数据交换，支持可视化配置数据源、目标选择、字段映射、同步速度控制等，详情请参考数据同步：数据交换的管道。
- HadoopMR：基于 Hadoop MapReduce 编程框架，基于 Java 语言的任务。
- PySpark：基于 PySpark API 的任务。
- 机器学习：基于 Spark MLlib API 的机器学习任务。
- 原生 Python 任务：基于原生 Python 语言的数据分析与处理过程，支持 Python2.X、Python 3.X。
- shell：支持 shell 脚本类型的任务，可用于调用外部接口、等待运行等场景。
- 深度学习：支持 TensorFlow、MXNet 框架的深度学习任务。



- 虚节点：执行批量任务管理的节点，例如对任务进行批量运行、批量停止等。
- SQL 脚本：临时数据查询、建表、删表、修改表结构等操作。

### ● 强大的调度引擎

在 Figure 2 典型的数据处理链路中，任务之间是存在依赖关系的，例如必须先完成数据采集才能进行数据清洗，两者必须串行，否则只能清洗其中一部分数据。同时每个任务都是周期运行的，例如每天凌晨 1:00 开始抽取前一天的数据，凌晨 1:30 开始进行数据清洗，这些任务每天都要运行，所以存在一个“周期”。

DTinsightIDE 提供强大的调度能力，支持按照时间、依赖关系的任务触发机制，支持各类任务按照 DAG<sup>4</sup>关系准确、准时运行，支持天、周、月、小时、分钟多种调度周期配置。用户仅需在页面进行简单配置即可。

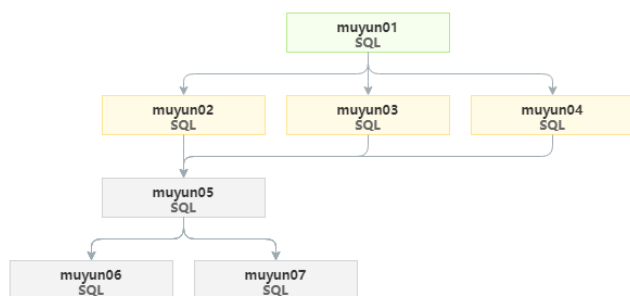


Figure 11 任务间依赖配置

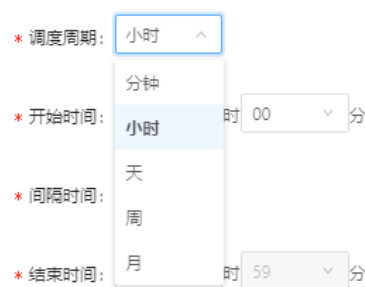


Figure 12 调度周期配置

### ● 友好的团队协作式开发平台

企业中的数据分析团队需要相互协作，每个用户既需要独立开发，又需要共享各类任务、资源等信息。

DTinsightIDE 基于 WEB IDE 的开发模式，重点面向多人协作式开发场景，通过统一任务管理、资源管理、函数管理、发布历史、任务锁等功能来提升团队的协作能力，以“修改即可见”的方式，可显著缩短开发周期。

<sup>4</sup> DAG: Directed Acyclic Graph, 有向无环图。

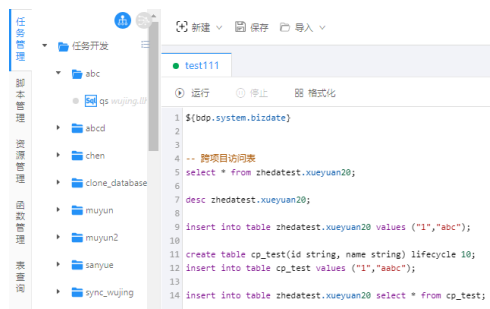


Figure 13 WEB IDE 的开发模式



Figure 14 代码版本管理

## 运维中心：保障数据正常产出

在日常使用中，开发人员除了进行数据同步和数据开发的工作之外，还需要保证平台能够正常产出数据。绝大多数的离线任务都是周期运行的，任务中的代码错误、节点运行异常等问题都会导致任务失败，因此就需要运维中心来监控每个任务的运行情况，并记录每个任务的运行日志，协助开发人员排查各种异常问题。

运维中心同时包括离线任务和实时任务的运维，本节仅介绍离线任务的运维，实时任务的开发与运维见实时任务的开发与运维。

### ● 运维总览

系统自动监控每个任务的运行状态并汇总显示，自动统计最近一段的任务运行情况，汇总易出错的任务、耗时较长的任务，协助用户排查代码质量、平台运行情况。

### ● 任务管理

发布后的任务都会显示在任务管理中，用户可以冻结/解冻操作，已冻结的任务不会再运行。同时可以进行“补数据”操作，指定此任务处理哪个时间范围内的数据。

### ● 任务实例的管理

任务的每次运行都产生一个实例，系统可监控此实例的运行状态，记录其运行日志，同时支持用户对实例进行各类操作，包括：重跑、终止、恢复调度等。



Figure 15 运维总览

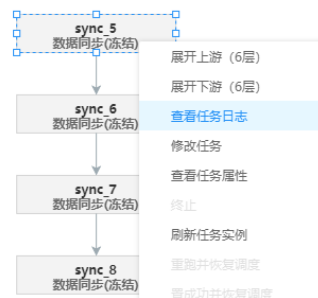


Figure 16 任务实例管理

## ● 监控告警

DTinsightIDE 支持对任务执行状态的告警，当任务因为各类原因导致超时、失败时，可触发短信、邮件或钉钉告警，及时通知任务责任人进行处理，保障每天的数据能够正常产出。

## 实时任务的开发与运维

### ● 什么是实时计算

在大数据开发领域，通常根据数据的不同性质，将任务划分为实时计算与离线计算，以温度传感器的场景举例：

假设某城市安装了大量的温度传感器，每个传感器每隔 1min 上传一次采集到的温度信息，由气象中心统一汇总，每隔 5 分钟更新一次各个地区的温度，这些数据是一直源源不断的产生的，且不会停止。实时计算就主要用于“数据源源不断的产生，而且不会停止，需要以最小的延迟获得计算结果”的场景，这种最小的延迟通常为秒级或分钟级。

为了满足这种数据量很大，而且实时性要求又非常高的场景，必须使用实时计算技术，实时计算的“数据源源不断”的特定决定了其数据处理方式与离线是截然不同的。

### ● 在 DTinsightIDE 上进行实时任务的开发

#### 开发流程

在 DTinsightIDE 上进行实时任务的开发流程非常简单，核心步骤共 3 部分：

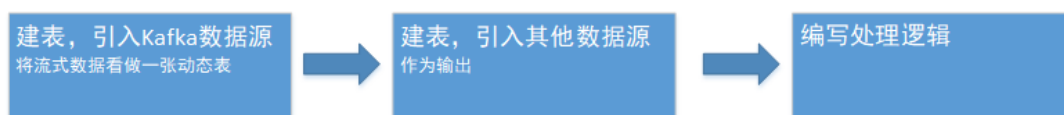


Figure 17 实时任务的开发流程

由于基于 Flink 作为计算引擎，利用 Flink 内置的 Table API，DTinsightIDE 支持对流式数据通过 SQL 代码进行处理。为了完成 Figure 17 中的整个流程，您只需要在 DTinsightIDE 的

数据开发模块中编辑 SQL 代码即可对流式数据进行处理。

### 资源与函数管理

DTinsightIDE 同时支持上传 Jar 包的形式进行流计算处理逻辑的编写，仅需在资源管理模块中上传 Jar 包，并创建相关的任务即可。

在进行 SQL 任务的开发中，DTinsight 也支持 UDF 的统一管理，通过上传资源->UDF 注册->使用 UDF 等几个步骤，用户即可快速使用 UDF，完成实时任务的开发。

### 实时任务的运维

在运维中心->实时任务运维模块中可查看实时任务实例的运行状态、查看任务运行日志，同时可以对任务状态进行切换。

DTinsight 支持对实时任务进行启动、停止操作。对于已停止的任务，可进行续跑、重跑操作，其中续跑操作可以将任务恢复至停止前的状态继续运行。

## 项目管理

不同的企业内的组织架构不尽相同，不同的业务部门、不同的数据分析主题，如果全都使用同一套开发环境的话，随着时间的推移，平台内的任务、数据表会越来越杂乱。

### ● 项目制切分

为了解决以上问题，DTinsightIDE 采用“项目”的形式进行切分，实现统一与灵活的融合。用户可以根据实际情况来灵活划分项目，例如“销售部门”和“物流部门”有不同的开发团队、不同的数据或不同的分析主题，双方可以建立不同的“项目”进行数据开发。

项目包含的内容内包含用户、计算任务、数据表、任务实例、数据源连接等信息，支持各开发小组进行不同的任务开发。

### ● 跨项目数据隔离

不同项目内的数据是相互隔离的，用户必须在审批授权的前提下才可以跨项目访问数据，保证数据安全性。

## 数据治理套件

数据治理是比较宽泛的概念，既包括管理手段也包括技术手段，DTinsight 产品提供产品化的形式，将数据治理的部分管理手段与技术手段落地，以产品化的形式帮助企业进行数据治理，主要包括以下几个模块：

**数据地图：**可视化的数据资产中心，帮助企业全盘掌控数据资产情况和数据的来源去向。

**数据模型：**使企业数据标准化，模型化，帮助企业实现数据管理规范。

**数据质量：**对过程数据和结果数据进行质量校验，帮助企业及时发现数据质量问题。

## 数据地图：数据资产的管理

数据地图的定位是可视化的数据资产中心，用户可以在数据地图模块中查看平台内的所有数据表情况，同时可以进行全方位的管理中台内的数据资产。

### ● 数据查找

汇聚平台内的所有数据表信息，方便开发人员快速定位所需数据表，支持用户根据类目、表名、所在项目、授权状态进行过滤，或直接根据表名搜索。

### ● 数据表元数据展现

用户指定某张表后，可以查看此表的基本信息，包括表名、物理存储量、生命周期、是否分区表、字段名称、字段类型、分区信息等，同时可以进行预览，直观的查看表内数据情况。

<a href="#">&lt; 返回</a> 查看表：ods_dd_staff_sign_log		<a href="#">收藏</a>	<a href="#">生成建表语句</a>
<b>基本信息</b>		<b>存储信息</b>	
所属项目	袋鼠云_工时三期报表	物理存储量	9.85KB
负责人	yida@dtstack.com	生命周期	90天
创建时间	2018-06-28 16:03:09	是否分区	是
所属类目		表结构最后变更时间	2018-06-28 17:41:57
描述	日志表，日志类型状态时间等信息，关联员工和项目	数据最后变更时间	2018-06-28 17:42:07
<b>字段信息</b> 分区信息 数据预览 血缘信息			
非分区字段 分区字段 共 14 个字段			
序号	字段名称	类型	注释
1	id	bigint	自增id
2	is_deleted	bigint	是否删除,1删除,0未删除
3	gmt_create	string	创建时间
4	gmt_modified	string	修改时间
5	date_time	string	员工领金币的时间
6	log_week_no	int	星期,星期几的意思
7	staff_id	string	员工id, 对应员工表的staff_id
8	staff_name	string	员工名, 对应员工表的staff_name

Figure 18 表的元数据信息

### ● 数据类目管理

当平台内的数据表数量逐渐增多时，有必要将数据表按照一定的类目结构组织起来，例如销售类、库存类、退货类……，方便管理的同时也加快了开发人员的寻找数据的速度，提高开发效率。当平台内的数据表越来越多时，数据类目的重要性就会日益突出。

DTinsightIDE 支持 3 层类目的管理，用户可自定义层级、名称，并将数据表指定至某个节点上，数据开发者在寻找数据时可根据数据类目快速定位。

### ● 数据审批授权

随着中台接入的数据规模逐渐变大，使用平台的用户也在逐渐增多，数据安全的问题会越来越重要，中台内的数据需要完善的管控体系，防止用户随意访问数据，降低敏感数据泄露的风险。

DTinsightIDE 支持表级数据权限的管理，当用户需要跨项目访问表时（读/写）需先经项目管理员审批授权，审批通过后才可以对表进行跨项目访问。同时，授权审批具备有效期的概念，超出有效期后自动取消授权，提升数据访问的安全程度。

### ● 生命周期管理

在平台由建设转向运营阶段的过程中，数据的生命周期管理会越来越重要，经常会有数据开发或数据分析人员临时建表并进行数据查询，日积月累，平台内的“临时数据”会越来越多，这就造成了存储空间的浪费。

DTinsightIDE 支持表的生命周期管理，用户可在建表时指定生命周期，系统定时检测每张表/分区的数据更新时间，超出时间后自动删除数据，降低临时数据造成的存储压力。

### ● 数据血缘解析

当用户配置了同步任务，并通过 SQL 任务进行多个步骤的清洗、转化处理之后最终会将结果数据输出，在整个处理链路中，数据的血缘关系就隐含在同步任务和 SQL 代码中，数据血缘表示每个统计指标是如何从原始数据得到的过程。

DTinsightIDE 自动解析同步任务和 SQL 代码，自动建立各个数据表的表级、字段级血缘关系，用户通过可直接在页面上看到每个指标的“前世今生”，便于快速排查指标问题，检查指标统计逻辑，依赖链路是否正常等。

同时，DTinsightIDE 会自动显示与此表相关的处理任务，用户可快速查看详细的源码处理逻辑。



Figure 19 表级血缘关系

## 数据模型：数据平台的规范化

随着数据中台的长期运营和开发人员的变动，中台内的数据也会变得逐渐不规范，通用性的平台工具必须依赖“人治”才能保障平台的数据健康。传统的解决办法是采用人工经验+人工约定的方式对平台内的表名做约束，实现上采用“口头约定”或“配套文档”，而数栈采用初始配置+自动检测的方式，提高数据表命名、指标命名的规范性。

### ● 应用场景

2 位用户（张三和李四）分别面对不同的分析主题，在不同的时间新建了 2 张表，如果没有任何约束，会造成表的命名很随意、字段命名随意，例如同样都是“交易金额”，张三使用 amount 表示、李四表示 sales 表示，造成定义不一致。

随着数据中台的长期运营，各种模型设计、指标定义会越来越混乱，必须采用自动化的形式进行自动约束。

张三的表，日交易额统计	李四，月交易额统计，按渠道划分																													
表名: table_sales	表名: tb_month_sale	表的命名很随意																												
<table><tr><th>字段名</th><th>备注</th></tr><tr><td>order_id</td><td>订单ID</td></tr><tr><td>user_id</td><td>用户ID</td></tr><tr><td>amount</td><td>交易金额</td></tr><tr><td>channel</td><td>交易渠道</td></tr><tr><td>payMethod</td><td>支付方式</td></tr><tr><td>time</td><td>交易时间</td></tr></table>	字段名	备注	order_id	订单ID	user_id	用户ID	amount	交易金额	channel	交易渠道	payMethod	支付方式	time	交易时间	<table><tr><th>字段名</th><th>备注</th></tr><tr><td>order_id</td><td>订单ID</td></tr><tr><td>user_id</td><td>用户ID</td></tr><tr><td>sales</td><td>交易金额</td></tr><tr><td>trans_chanel</td><td>交易渠道</td></tr><tr><td>payMethod</td><td>支付方式</td></tr><tr><td>time</td><td>交易时间</td></tr></table>	字段名	备注	order_id	订单ID	user_id	用户ID	sales	交易金额	trans_chanel	交易渠道	payMethod	支付方式	time	交易时间	业务含义相同，命名不同
字段名	备注																													
order_id	订单ID																													
user_id	用户ID																													
amount	交易金额																													
channel	交易渠道																													
payMethod	支付方式																													
time	交易时间																													
字段名	备注																													
order_id	订单ID																													
user_id	用户ID																													
sales	交易金额																													
trans_chanel	交易渠道																													
payMethod	支付方式																													
time	交易时间																													

## ● 使用方式

DTinsightIDE 的数据模型模块可通过简单的配置即可建立模型、指标约束规则，相比传统手段约束力更强，覆盖更广泛。

### 配置阶段

借鉴阿里巴巴多年的数据中台建设经验，将数据模型划分为：数仓层级、主题域、刷新频率、增量定义 4 个元素，用户可在“表名生成规则”模块配置表名由哪些元素组成。

同时支持将指标划分为原子指标、衍生指标，用户使用时需进行初始配置。

模型层级	主题域	刷新频率定义	增量定义	表名生成规则	原子指标定义	衍生指标定义	
层级编号	层级名称	层级说明	层级前缀	生命周期	是否		
731	ods	操作数据层	ods	1000	是		
733	dwd	数据仓库明细层	dwd	365	是		
735	dws	数据仓库汇总层	dws	90	是		
737	ads	数据应用层	ads	7	是		
739	tmp	临时表	tmp	7	否		
741	dim	维度表	dim	30	否		

### 检测阶段

经过初始化配置后，系统自动检测平台内的数据表、字段信息，与标准模型对比，输出检测结果，提示用户平台中有哪些不符合规范的模型和数据指标。



模型检测						
字段检测						
按字段名搜索		Q	类型:	选择类型	<input type="checkbox"/> 已忽略	
字段名称	字段描述	字段类型	所属表	最近检测时间	检测结果	操作
pt		STRING	dappa_tenant	2018-06-28 23:50:01	名称不匹配	<a href="#">修改</a>   <a href="#">忽略</a>
pt		STRING	da_snapshots_api	2018-06-28 23:50:01	名称不匹配	<a href="#">修改</a>   <a href="#">忽略</a>
id		INT	dappa_snapshot_invoke	2018-06-28 23:50:01	类型不匹配 描述不匹配	<a href="#">修改</a>   <a href="#">忽略</a>
id		INT	da_snapshots_err	2018-06-28 23:50:01	类型不匹配 描述不匹配	<a href="#">修改</a>   <a href="#">忽略</a>
user_id		INT	dappa_snapshot_invoke	2018-06-28 23:50:01	名称不匹配	<a href="#">修改</a>   <a href="#">忽略</a>
api_id		INT	da_snapshots_err	2018-06-28 23:50:01	名称不匹配	<a href="#">修改</a>   <a href="#">忽略</a>
api_id		INT	dappa_snapshot_invoke	2018-06-28 23:50:01	名称不匹配	<a href="#">修改</a>   <a href="#">忽略</a>
user_id		INT	da_snapshots	2018-06-28 23:50:01	名称不匹配	<a href="#">修改</a>   <a href="#">忽略</a>

## 数据质量：为数据正确性保驾护航

作为数据治理的一部分，数据质量的保障与提升是大数据平台的必备功能。通常含义的数据质量包括及时性、完整性、一致性、有效性、准确性，落地到具体的平台功能点，数栈的数据质量（DTInsightValid）划分为规则的配置、校验结果的查询等模块，下面详细介绍。

### ● 支持的数据源

DTInsightValid 支持对 MySQL、Oracle、SQLServer、PostgreSQL 等常用的关系型数据库进行质量校验，同时也支持 Hive、MaxCompute 等大数据存储，用户可方便的对这些数据库配置表级、字段级的质量校验规则。

### ● 规则配置

借鉴了阿里巴巴多年的数据质量监控经验，DTInsightValid 支持表级、字段级、自定义 SQL 三种形式的规则设定，满足用户在不同场景下的监控需求。

- 用户根据不同的业务场景，DTInsightValid 提供表行数、空值数、空值率、重复数、重复率等二十余种统计函数，校验方法支持固定值检测、1 天波动检测、7 天波动值变化检测、30 天波动值检测、7 天平均波动检测、30 天平均波动检测，告警阈值支持灵活的自定义。
- 校验规则的调度配置完全自定义，支持小时、天、周、月、手动触发五种模式。
- 支持邮件、短信两种告警方式。

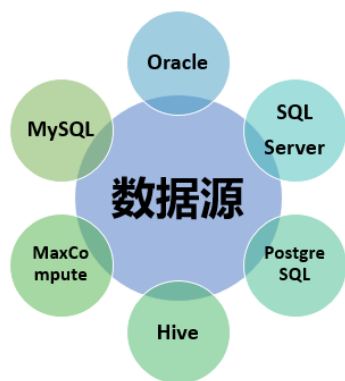


Figure 20 支持 5 种常用数据源

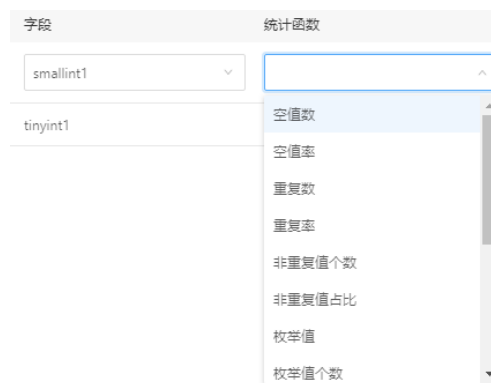


Figure 21 内置丰富的校验规则

## ● 质量报告

已配置的规则会按照调度周期运行，用户可浏览每次质量校验任务的执行结果，并提供详细的监测错误信息，便于用户进行数据问题排查。

**详细报告：**列出每条规则的校验结果，当字段波动偏离阈值时自动发出告警，便于快速定位数据异常。系统同时提供每个监控项近一个月的指标波动图，用户可通过观察指标波动是否剧烈来辅助定位问题。

**表级报告：**针对全表数据的宏观指标，例如记录数、告警数等。对最近 30 次的校验结果进行分析，统计记录数波动、平均告警值等信息。



Figure 22 字段级报告

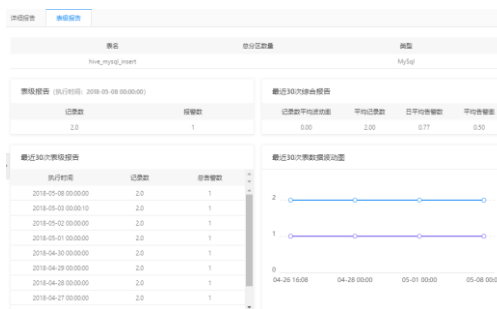


Figure 23 表级报告

## ● 远程触发

远程触发是指用户可以通过接口调用的形式，对数据质量的校验任务进行触发执行，校验通过后再启动后续的节点任务。便于用户确保数据链路的准确性，与数据开发任务形成闭环，提高整体数据质量。

DTinsightValid 支持根据已配置的校验规则来生成远程触发的地址，用户使用此链接地

址即可触发部分规则运行，并获取运行结果。

## ● 逐行校验

逐行校验是用户在做数据迁移或数据部分更新场景中的重要功能，通过逐行校验用户可以监控到数据在迁移、更新时的数据完整性、准确性。

用户仅需在页面上进行简单配置即可完成对 2 张数据表的一致性校验。DTinsightValid 支持源表、目标表的字段映射、主键选择。同时可配置对“微小差异”进行忽略，包括记录数差异、数值差异百分比、数值差异绝对值、字符大小写区分等规则，执行校验时可忽略掉相应的差异数据。

校验任务运行结束后，系统可输出校验结果，包括 2 表的总记录数差异、匹配成功的记录数，有差异记录数的明细列表等。

<input type="checkbox"/>	date1	DATE		<input type="checkbox"/>	date1	DATE
<input type="checkbox"/>	time1	TIME		<input type="checkbox"/>	time1	TIME
<input type="checkbox"/>	year1	YEAR		<input type="checkbox"/>	year1	YEAR
<input type="checkbox"/>	timestamp1	TIMESTAMP		<input type="checkbox"/>	timestamp1	TIMESTAMP
<input type="checkbox"/>	datetime1	DATETIME		<input type="checkbox"/>	datetime1	DATETIME

<input type="checkbox"/>	记录数差异，对比左右表的总记录数，差距小于		%时候，计为成功匹配
<input checked="" type="checkbox"/>	数值差异百分比，对比左右表的数值型数据时，差距百分比小于		%时候，计为成功匹配
<input checked="" type="checkbox"/>	数值差异绝对值，对比左右表的数值型数据时，差距绝对值小于		时候，计为成功匹配
<input type="checkbox"/>	数值对比忽略小数点，忽略小数点后		位
<input type="checkbox"/>	字符不区分大小写，对比左右表的字符串型数据时，不区分大小写		
<input type="checkbox"/>	空值与NULL等价，对比左右表的数据时，认为空值与NULL值是相等的		

## 数据应用引擎

数据中台内的应用层是对外提供中台的数据服务，DTinsightAPI 主要解决 API 的快速生成和

## 数据 API：统一管理对内对外数据服务

DTinsightAPI 主要解决 API 的快速生成和对外数据服务。API 管理者可利用产品化的配置工具生成各类 API 服务，监控所有 API 的调用及订购情况，让自己的数据资产价值对外输出，同时可见、可管。与此同时，对于 API 使用者，可以看见 API 市场中所有的 API，根据

需求自助选取合适的 API，开始自己的使用之旅，极大的提高了使用效率与易用性。

## ● 双视角设计

数据 API 从 API 管理者和 API 使用者双视角切入，API 管理者拥有 API 创建、所有 API 管理、API 调用情况监控、API 订购监控、授权审批等权限，API 申请者可以申请 API、管理自己的 API、查看自己 API 的使用情况，全方位为两种用户角色提供数据服务；

## ● 多数据源可视化配置

API 配置的数据源支持多种关系型数据库：MySQL、Oracle、SQLServer、PostgreSQL、Analytic DB，后续也会支持 impala 数据源。根据选取的数据源类型，数据库、以及数据表，可直观的看到该数据表下所有的数据字段，并一键配置输入参数、输出参数，快速生成 API。



基本属性 2 参数配置 3 完成

**\* 数据源配置:**

MySQL  
mysql\_xc  
da\_api

**\* 数据字段:**

字段	字段类型
<input checked="" type="checkbox"/> id	INT UNSIGNED
<input type="checkbox"/> cat_id	INT
<input checked="" type="checkbox"/> tenant_id	INT
<input type="checkbox"/> data_src_id	INT

**API参数配置**

**\* 输入参数:**


参数名称	绑定字段	字段类型	操作符	必填	说明
<input type="checkbox"/> id	id	INT UNSIGNED	=	是	

**\* 输出参数:** 返回结果分页

参数名称	绑定字段	字段类型	说明
<input type="checkbox"/> id	id	INT UNSIGNED	
<input type="checkbox"/> tenant_id	tenant_id	INT	

## ● 自定义 SQL 生成

在 API 使用过程中，用户可能会遇到需要多张表进行关联查询、以及简单聚合函数计算的功能使用场景，为了弥补模板向导模式的不足，DTinsightAPI 自定义 SQL 模式。通过写 SQL 语句实现双表关联，聚合函数计算等功能，同时可以根据 SQL 语句解析出相应的入参、出参，方便用户核对校验。



基本属性 2 参数配置 3 完成

**\* 数据源配置:**

MySQL  
mysql\_xc  
da\_api

**\* 数据字段:**

字段	字段类型
id	INT UNSIGNED
cat_id	INT
tenant_id	INT
data_src_id	INT

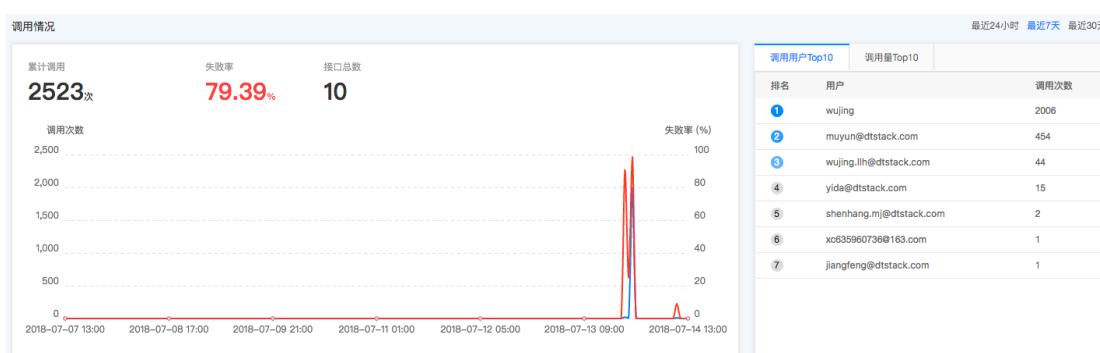
**API配置SQL语句** 格式化语句 SQL编写提示 代码 编辑参数

```
1 SELECT id, name
2 FROM da_api
3 WHERE id = $id;
```

## ● API 统一管理监控

**授权审批：**授权审批将 API 申请者申请，到 API 管理者审批流程形成闭环，对于 API 管理者，当有用户提交的 API 申请时，可对其授权审批，同时也可拒绝，取消 API 授权、禁用 API，管理每个用户的 API 使用情况。同时 API 申请目前可细化到申请周期、申请次数，方便数据 API 的价值量化。

**调用监控：**DTinsightAPI 提供每个 API 的基础信息、调用情况、订购情况查看功能，API 管理员可纵观 API 使用情况，了解 API 接口的时间段调用次数、失败率、调用 TOP 用户等，帮助用户进行 API 管理与维护。

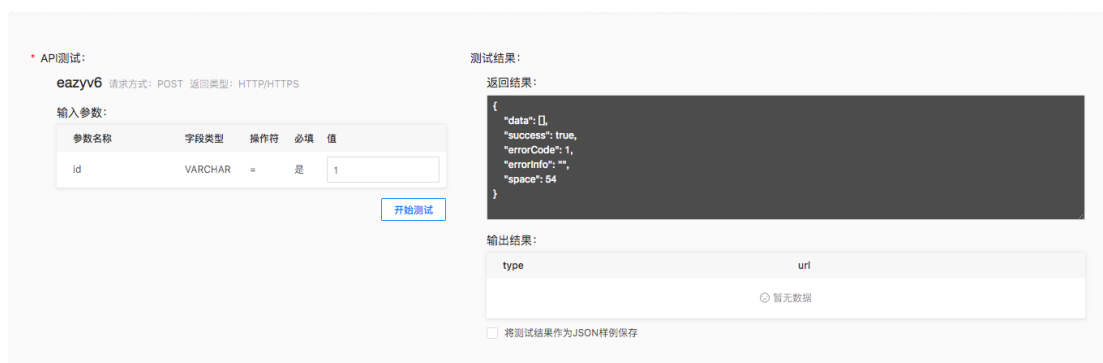


**错误查看：**在 API 调用过程中，会存在 API 调用失败的情况，目前统计 API 的 5 种错误类型：调用超时、未认证、参数错误、超出限制及其他，可以让帮助开发者直接定位到问题，提高 API 使用效率。

## ● API 测试

API 测试是为 API 生成后的发布做的一道保障程序，在生成 API 后，API 管理者需进行 API 测试来保障 API 的可用性，防止出现申请者使用不可用 API 服务的情况。

在产品操作层面，API 测试也是非常简单的可视化配置，直接填写输入参数值，就可以开始测试，最终返回 JSON 调用结果，系统会告诉用户调用成功还是失败，同时可将返回的 JSON 结果最为样例保存，作为 API 申请者的参考信息。



API 测试界面展示了配置和测试 API 的过程。配置部分包括：

- API 测试：**请求方式：POST，返回类型：HTTP/HTTPS
- 输入参数：**

参数名称	字段类型	操作符	必填	值
id	VARCHAR	=	是	1
- 开始测试：**按钮

测试结果部分显示了返回的 JSON 数据：

```
{
  "data": [],
  "success": true,
  "errorCode": 1,
  "errorMsg": "",
  "space": 54
}
```

输出结果部分显示了 type 和 url 字段，并提供了将测试结果作为 JSON 样例保存的选项。

- **数据安全保障**

数据对外提供服务，资产的安全性非常重要，DT.insight 通过多种方式保障数据安全。一是审批授权，使用 API 时需经过 API 管理者的审批，可对 API 的使用者、使用次数、使用时间做以为限制；二是对 API 的调用 URL 进行 token 加密，授权用户获得唯一 token，通过 token 验证用户真实性；三是进行 API 调用限流，对单秒的调用次数进行不超过 2000 次的限制，方式恶意调用及攻击。通过多种手段保障 API 调用的安全型。

## 应用案例

### 社保行业

- **客户：**浙江地区某县级社保单位
- **诉求：**医保异常行为识别，预测社保基金收入，指导宏观政策调整等业务诉求。
- **项目规模：**数据量：800GB+；任务数：170+；表数量：100+；
- **项目效果：**
  - **性能上：**

医保累计刷卡人数、累计刷卡金额实时获取；绝大部分指标 T+1 获取；
  - **业务上：**

发现一批医保异常的用户，监测到部分医保消费异常（金额为百万级）；

参保推荐精准化，参保率持续提升；对医保消费进行预测，已起到辅助制定医保政策的效果；

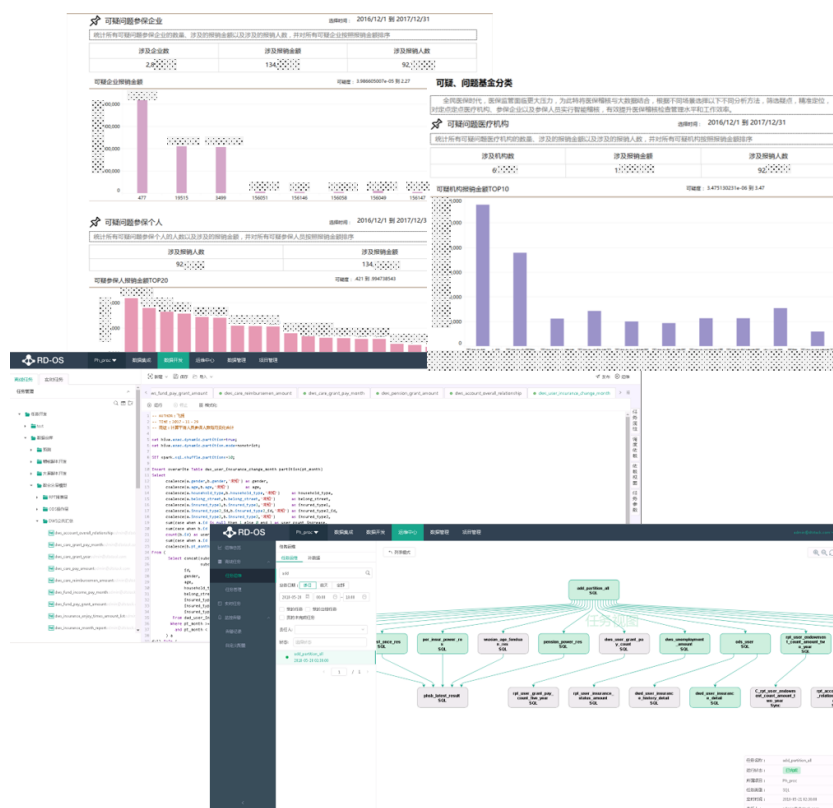


Figure 24 项目效果图，上半部分为报表，下半部分为产品使用效果

## 政府行业

- 客户：浙江某市级发改委
- 诉求：为发改委的各项报告提供数据支持，提升数据分析的效率。
- 客户现状：
  - 熟练使用 Cloudera 平台，可以熟练操作 Hadoop、Spark 等；
  - 各项数据支撑工作都是临时编写任务、脚本，没有任务、调度的概念；
  - 数据需要从 MySQL、Oracle、SQLServer 上全量/增量采集；
  - 需求较简单情况下，从数据采集、数据分析、结果数据输出最快需要 1 天时间；
- 项目规模：数据量：累计约 500GB+，月增 10GB；任务数：100+；表数量：100+；
- 项目效果：
  - 需求开发：从原来的 1 天缩短到半小时；
  - 任务运维：每天花费 3min 上巡查；

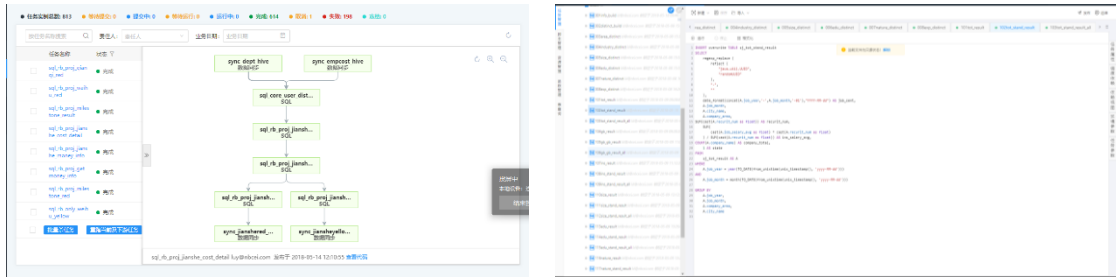


Figure 25 使用效果图