数据测试功能（数据沙箱）

数据测试功能，可生效在离线开发和自助分析两个子产品中，主要解决的是数据开发过程中的线上数据和开发数据隔离问题。

背景

当前线上调度任务中运行着如下的代码，代码中的表**intern_new.dim_user_info_p**被下游所使用，此时需要对该表进行逻辑修改或增加字段等操作。代码示例如下：

```
insert overwrite table intern_new.dim_user_info_p
partition (ds='2021-07-25')
select id as user_id,
name as user_name,
province,
age
from intern_new.ods_user_info
where ds='2021-07-25';
```

通常来说会有两种方式进行线上表的修改：

- 第一种，开发人员选择新建一个临时表，并复制出现有代码，先进行测试验证。当验证通过时，再将该表更新，并更新代码，然后将任务提交上线。
- 第二种，开发人员选择直接修改该表，并修改原有代码，修改完成后，将任务提交上线。

对于上述两种方法，第一种方法更可靠，能保证所有操作不影响线上数据，也不影响下游，但是会多一些额外的工作；第二种方法可以节约额外的创建临时表，以及代码反复拷贝修改的时间，但是特别容易出纰漏，影响线上数据。

为了更好解决上述问题，数开平台支持**数据测试**功能。**数据测试**功能引入了一种新的代码规则，可以实现同一份代码根据运行环境不同，自动进行部分参数替换，实现操作离线表或测试表的功能，以及操作默认HDFS文件或测试HDFS集群文件的功能。当前可生效在离线开发和自助分析两个子产品中，解决了数据开发过程中的线上数据和开发数据隔离问题。

功能介绍

针对Hive库

现将上述案例引入**数据测试**功能后，对库名intern_new进行修改为${intern_new}，代码如下：

```
insert overwrite table ${intern_new}.dim_user_info_p --修改点是intern_new 变成了 ${intern_new}
partition (ds='2021-07-25')
select id as user_id,
name as user_name,
province,
age
from intern_new.ods_user_info
where ds='2021-07-25';
```

在离线开发的**开发模式**下以及自助分析的**测试模式**下，运行上述sql，系统会自动将上述代码进行转换，并且使用测试Yarn集群的资源。自动转换后的代码如下：

```
insert overwrite table intern_new_dev.dim_user_info_p
partition (ds='2021-07-25')
select id as user_id,
name as user_name,
province,
age
from intern_new.ods_user_info
where ds='2021-07-25';
```

此处，将**${intern_new}**作为参数去匹配当前项目集群下**intern_new**库所对应的测试库**intern_new_dev**，从而实现操作测试库下的表。

如果上述代码，提交上线并设置定时调度，在线上调度、重跑、补数据运行时，系统会自动将上述代码进行转换，并且使用默认Yarn集群的资源（即项目的默认Yarn队列）。自动转换后的代码如下：

```
insert overwrite table intern_new.dim_user_info_p
partition (ds='2021-07-25')
select id as user_id,
name as user_name,
province,
age
from intern_new.ods_user_info
where ds='2021-07-25';
```

这里将**${intern_new}**作为参数去匹配当前项目集群下的普通库，即**intern_new**。

> **注意：** 要使用上述自动库替换功能，需要先确保当前项目集群下要操作的普通库已经存在对应的测试库，并且有待测试的表结构的同名表。对应测试库，可由项目的管理员进行申请，并进行授权。

针对HDFS

离线开发中，支持使用Script、MR等节点，用户可在节点中自行指定hdfs相对路径。

当开启**数据测试**功能，数据的HDFS存储路径也会根据运行环境自动转换：

- 当在离线开发的开发模式运行会自动在存储路径前补充测试HDFS集群的前缀。
- 当在离线开发的线上模式进行任务调度，或对实例进行重跑、补数据等操作时，会自动在存储路径前补充默认HDFS集群的前缀。

上述例子中，表的存储路径如下：

- 表**intern_new_dev.dim_user_info_p**的HDFS存储路径为：**hdfs://cluster1/user/intern/hive_db/intern_new.db/dim_user_info_p**。
- 表**intern_new.dim_user_info_p**的HDFS存储路径为：**hdfs://dev4/user/intern/hive_db/intern_new.db/dim_user_info_p**。

> **说明：** 对于测试表，默认location的测试HDFS集群是**cluster1**，这个是测试HDFS集群。对于离线表，默认location的默认HDFS集群是**dev4**，这个是默认HDFS集群。而两个路径的后半部分地址相同，都是**user/intern/hive_db/intern_new.db/dim_user_info_p**。

功能实现

使用**数据测试**功能需要当前平台具备生产集群和测试集群，并安装了相关服务，具体要求可联系平台的运维人员。

数据测试功能开启

目前，有3种方式来开启**数据测试**功能。

- 方式一：当申请新建项目时，可同时申请开通测试功能。此时，进行**新建项目**配置过程中还需要进行**测试功能**区块的配置。

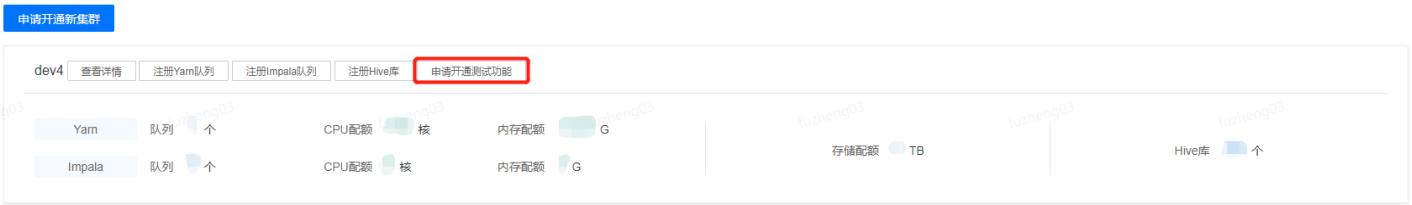
重要参数说明：

参数名称	说明
测试存储配额	是后续测试库表可存放测试数据的空间大小，取决于当前平台的测试HDFS集群的空间，具体可联系平台运维人员确定。
测试Hive库	一个新建的项目默认会按照项目名称创建一个同名的Hive离线库。当开通测试功能后，系统也会自动创建一个对应的测试库。比如项目名叫**my_dwd**，则测试Hive库就叫**my_dwd_dev**。

测试资源配置	一个新建的项目需要指定一个Yarn队列的名称，当开通了测试功能后，则会在测试集群上自动创建一个同名的Yarn队列。比如默认Yarn队列名叫**my_queue**，则测试Yarn队列也叫**my_queue**。

- 方式二：对于已有的项目，申请在新集群开通时，可同时申请开通测试功能。需要配置的内容基本和方式一相同。

- 方式三：支持在项目中，对已存在的集群申请开通测试功能。
  资源管理页面，在具备开通测试功能的集群上，单击**申请开通测试功能**按钮，在如下界面进行配置：



重要参数说明：

参数名称	**说明**
测试存储配额	是后续测试库表可存放测试数据的空间大小，取决于当前平台的测试HDFS集群的空间，具体可联系平台运维人员确定。
Hive库选择	在申请开通时，有需要选择一个已有离线库作为当前项目-集群的主库。系统会在申请通过后，自动创建一个名为**{主库}_dev**的测试库。
测试资源配置	对于已有项目，会存在一些Yarn队列。在申请开通测试功能后，系统会自动在测试Yarn集群上新建相同数量的同名Yarn队列。后续管理员可在**项目中心**的**资源管理**中，申请修改测试队列的资源。

典型场景

案例背景：当前生产环境中存在一个ods表和dim维表，dim维表由ods表加工得到。现在需要给ods表增加一个字段，因此dim维表也需要进行同步更新。在整个过程中，使用了两次数据测试功能，并使用了离线开发的**离线表新增字段**和**表克隆**功能。

场景1：基于ods表加工dim维表

1）ods表的DDL语句

假设ods表为一张用户信息表：

```
CREATE TABLE `intern_new_dev`.`ods_user_info`(
  `id` string COMMENT 'id',
  `name` string COMMENT '姓名',
  `province` string COMMENT '省份',
  `age` string COMMENT '年龄'
  )
COMMENT '用户信息表'
PARTITIONED BY (
  `ds` string COMMENT '时间分区')
ROW FORMAT SERDE
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat';
```

2）在测试库设计表

离线库为**intern_new**，测试库为**intern_new_dev**，设计的dim维表DDL如下：

```
CREATE TABLE `intern_new_dev`.`dim_user_info_p`(
  `user_id` string COMMENT 'id',
  `user_name` string COMMENT '姓名',
  `province` string COMMENT '省份',
  `age` string COMMENT '年龄'
  )
COMMENT '用户全量维表'
PARTITIONED BY (
  `ds` string COMMENT '时间分区')
ROW FORMAT SERDE
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat';
```

3）在离线开发新建任务，并拖入SQL节点

SQL节点的代码如下（其中**${azkaban.flow.1.days.ago}**是azkaban调度系统的系统参数，表示昨天）：

```
insert overwrite table ${intern_new}.dim_user_info_p
partition (ds='${azkaban.flow.1.days.ago}')
select id as user_id,
name as user_name,
province,
age
from intern_new.ods_user_info
where ds='${azkaban.flow.1.days.ago}';
```

在**开发模式**下，运行SQL节点。在**运行设置**弹框中，保持**测试模式**开关打开。运行后，数据实际插入到的是**intern_new_dev.dim_user_info_p**中。确认测试库的表的数据没有问题后，可进行下一步。

4）通过表克隆功能将测试库表克隆到离线库

通过单击**辅助功能区**的图标 ，选择**表克隆**。在**表克隆**的弹框中，设置待克隆表为**intern_new_dev.dim_user_info_p**，目标库为**intern_new**。

克隆检验通过后，则可进入下一步，系统会根据源表内容自动生成建表语句：

5）将任务提交上线并编辑调度

最后，将任务提交上线，编辑调度。在线上调度的任务，实际执行时，就会执行如下代码，将数据插入到“intern_new.dim_user_info_p”表中：

```
insert overwrite table intern_new.dim_user_info_p
partition (ds='${azkaban.flow.1.days.ago}')
select id as user_id,
name as user_name,
province,
age
from intern_new.ods_user_info
where ds='${azkaban.flow.1.days.ago}';
```

场景2：dim维表增加字段

1）ods源头表增加了性别字段

ods表新的ddl如下：

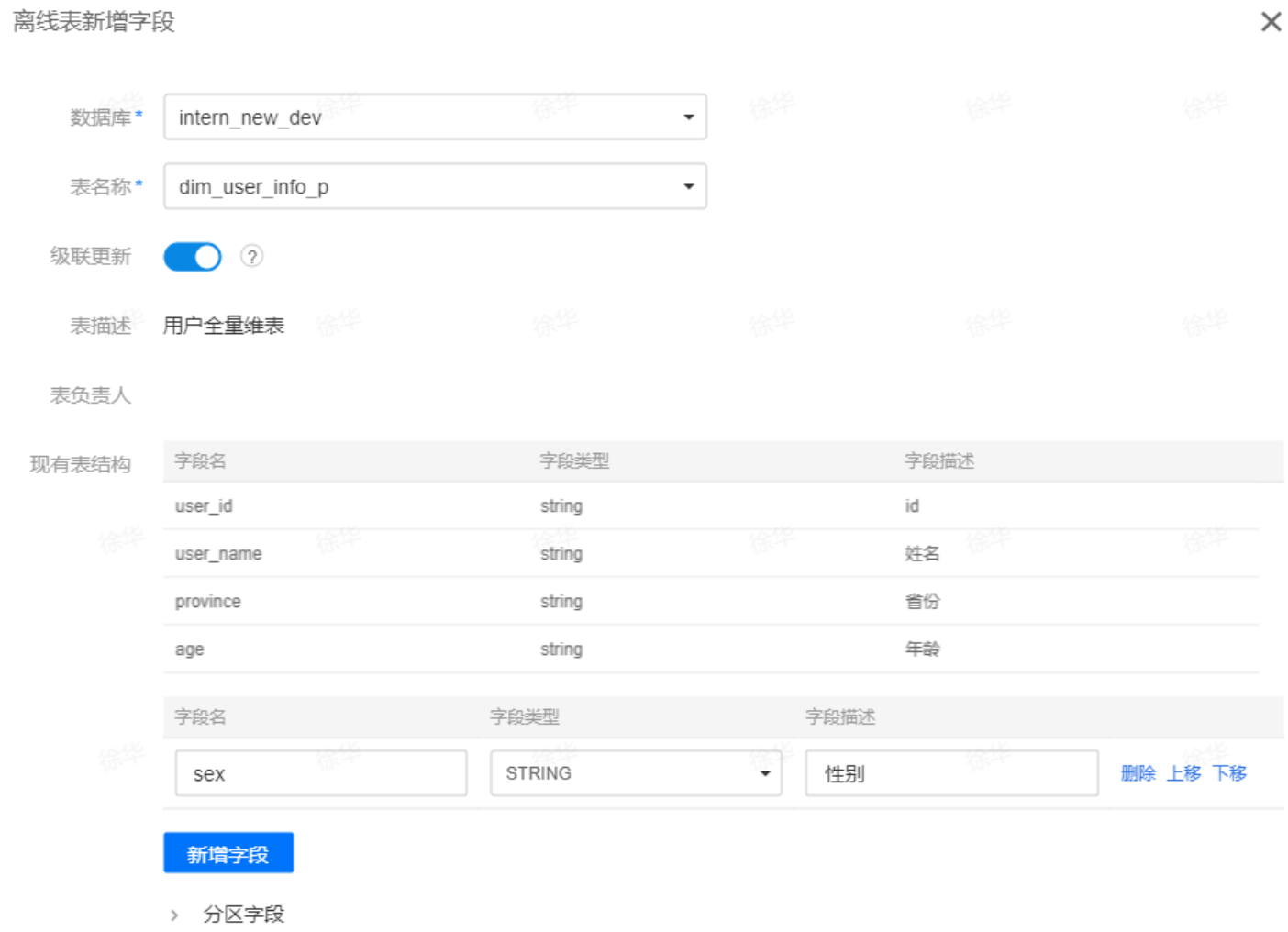
```
CREATE TABLE `intern_new_dev`.`ods_user_info`(
  `id` string COMMENT 'id',
  `name` string COMMENT '姓名',
  `province` string COMMENT '省份',
  `age` string COMMENT '年龄',
  `sex` string COMMENT '性别' --新增的字段
  )
COMMENT '用户信息表'
PARTITIONED BY (
  `ds` string COMMENT '时间分区')
ROW FORMAT SERDE
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat';
```

2）对测试dim维表表新增字段

使用离线开发的**离线表新增字段**功能，给测试维表**intern_new_dev.dim_user_info_p**增加一个性别字段。

通过单击**辅助功能区**的图标 ，选择**离线表新增字段**。

如下图，为新增性别字段：



3）修改原SQL节点代码

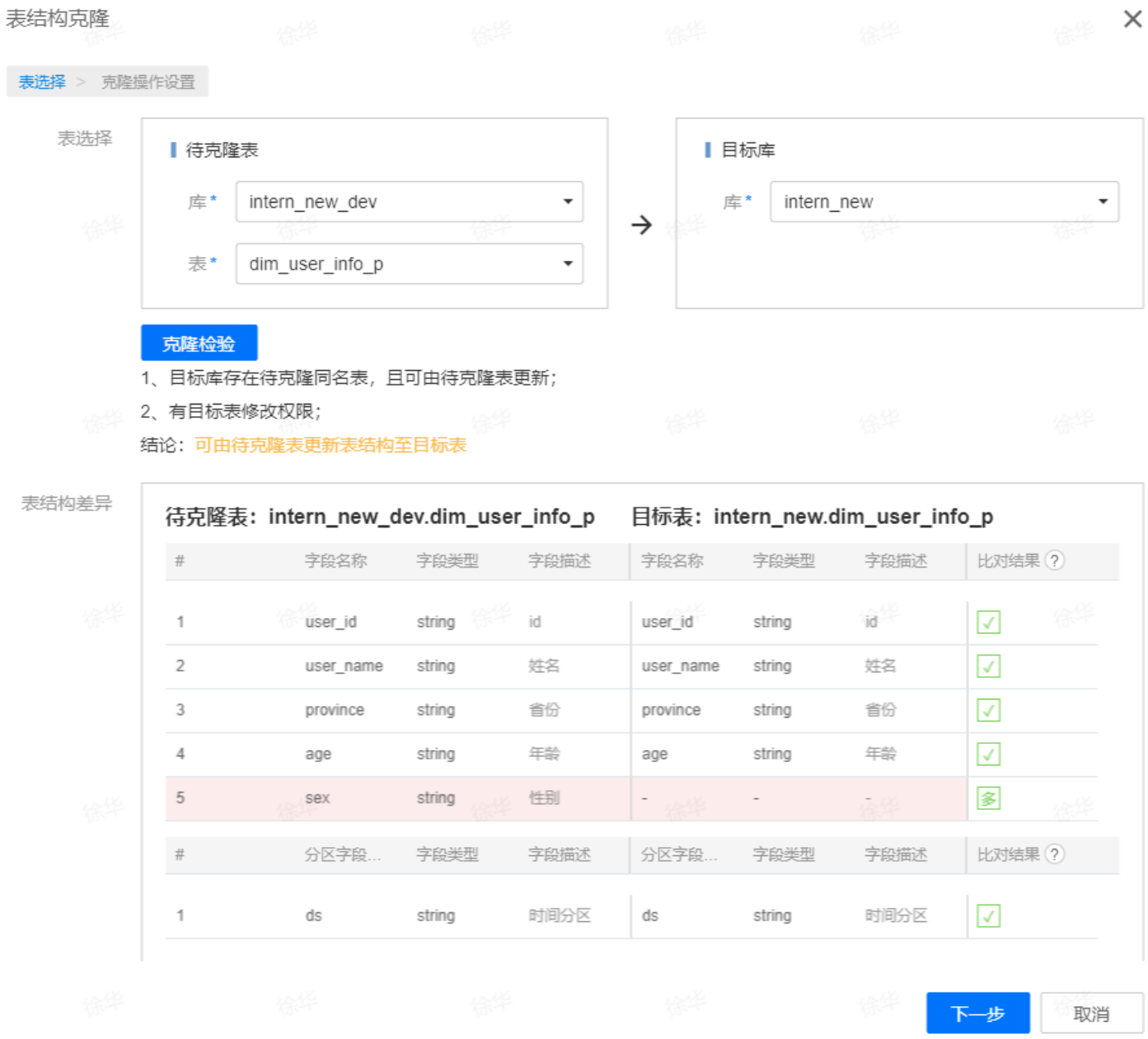
新的SQL节点代码如下：

```
insert overwrite table ${intern_new}.dim_user_info_p
partition (ds='${azkaban.flow.1.days.ago}')
select id as user_id,
name as user_name,
province,
age,
sex --为新增加的代码
from intern_new.ods_user_info
where ds='${azkaban.flow.1.days.ago}';
```

修改后，在**开发模式**下运行SQL节点。在**运行设置**弹框中，保持**测试模式**开关打开。运行后，数据实际插入到的是**intern_new_dev.dim_user_info_p**中。确认测试库的表的数据没有问题后，可进行下一步。

4）通过表克隆功能将测试库表更新到离线库

在**表克隆**的弹框中，设置**待克隆表**为**intern_new_dev.dim_user_info_p**，目标库为**intern_new**。经过**克隆检验**，发现待克隆表比目标库的表多一个字段，可由待克隆表更新表结构至目标表。



在表结构克隆界面查看建表语句，如下图所示，图中建表语句实际是对**intern_new.dim_user_info_p**表执行加字段操作。

5）将任务提交上线

最后，将任务提交上线。在线上调度的任务，实际执行时，就会执行如下代码，将数据插入到**intern_new.dim_user_info_p**表中：

```
insert overwrite table intern_new.dim_user_info_p
partition (ds='${azkaban.flow.1.days.ago}')
select id as user_id,
name as user_name,
province,
age,
sex
from intern_new.ods_user_info
where ds='${azkaban.flow.1.days.ago}';
```