

Alibaba Cloud DataWorks

クイックスタート

Document Version 20190527

目次

1 注意事項.....	1
2 Step 1: テーブル作成とデータのアップロード.....	2
3 ステップ 2: 業務フローの作成.....	11
4 ステップ 3: 同期タスクの作成.....	17
5 手順 4: スケジューリングと依存関係設定.....	25
6 手順 5 : O&M とログのトラブルシューティング結果の表示.....	29

1 注意事項

ここではデータ開発と運用について説明します。



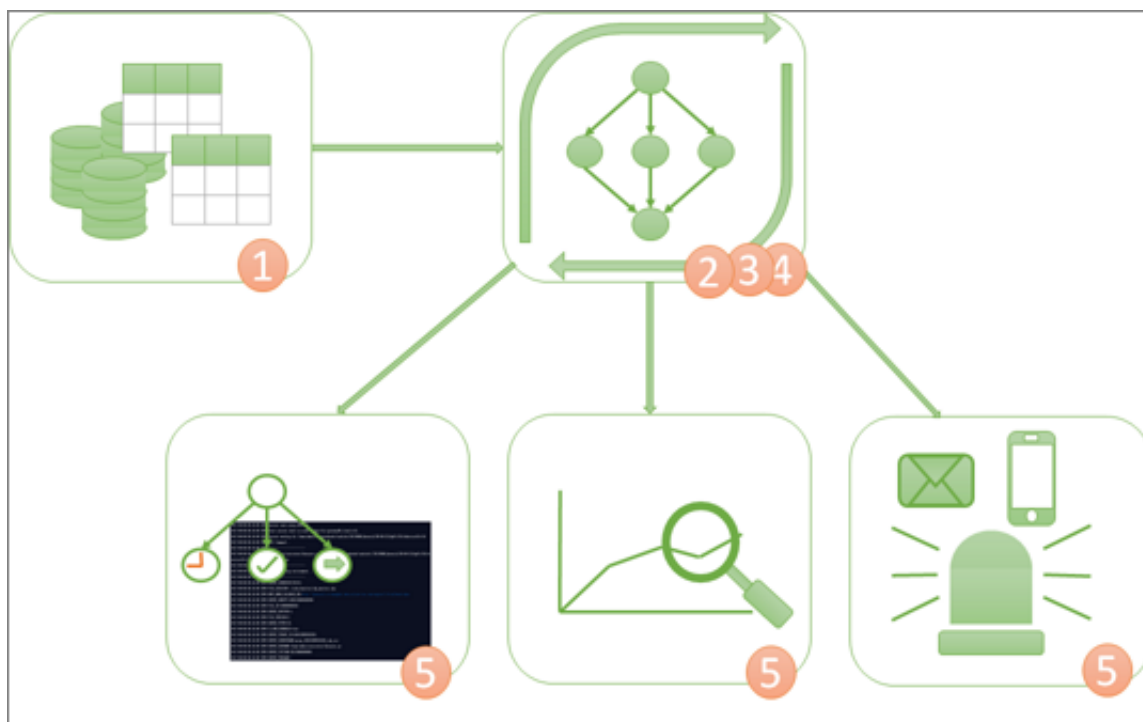
注：

DataWorks を初めて使用する場合、**[準備]** の章で記載したすべての準備(例: アカウント、プロジェクトロール、プロジェクトスペースなどの準備)が完了したことを確認し、DataWorks 管理コンソールにアクセスします。該当プロジェクトで **[ワークスペースにアクセス]** をクリックしてデータ開発を開始します。

通常、DataWorks のプロジェクトスペース上でのデータ開発と運用には以下のアクションが含まれます。

- ・ Step 1: テーブル作成とデータのアップロード
- ・ ステップ 2: 業務フローの作成
- ・ ステップ 3: 同期タスクの作成
- ・ 手順 4: スケジューリングと依存関係設定
- ・ 手順 5: O&M とログのトラブルシューティング結果の表示

全体的な手順は、以下の図をご参照ください。



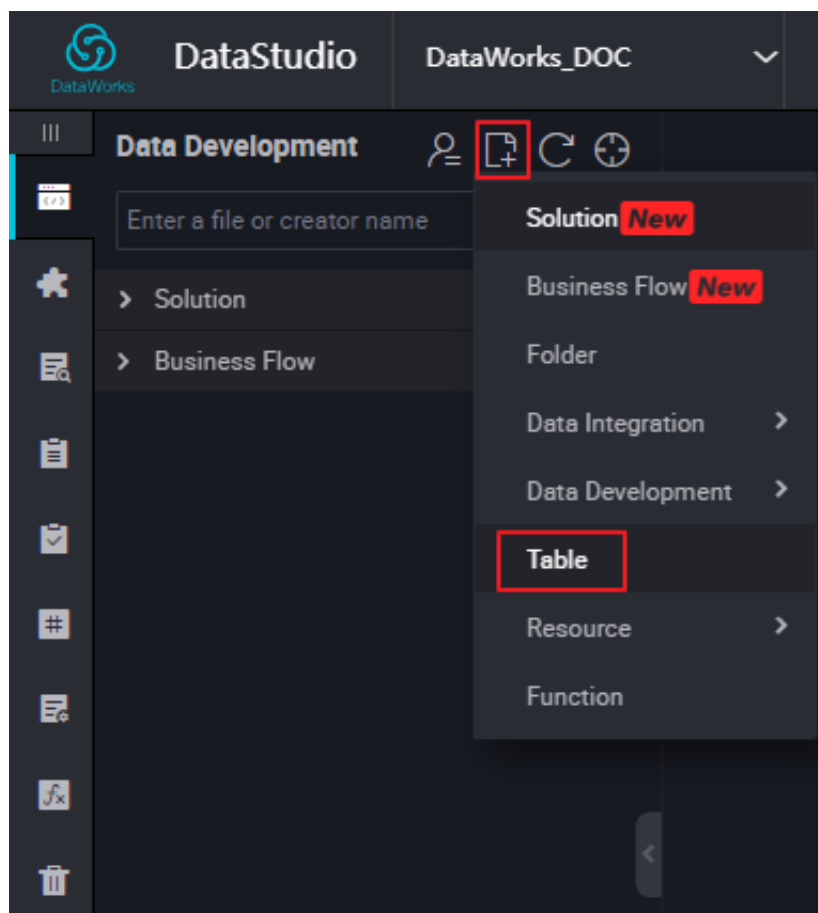
2 Step 1: テーブル作成とデータのアップロード

本ドキュメントでは、テーブル作成とデータアップロードのサンプルとして、作成済の `bank_data` と `result_table` を使用します。 `bank_data` テーブルは業務データを格納し、 `result_table` はデータ分析の結果を格納します。

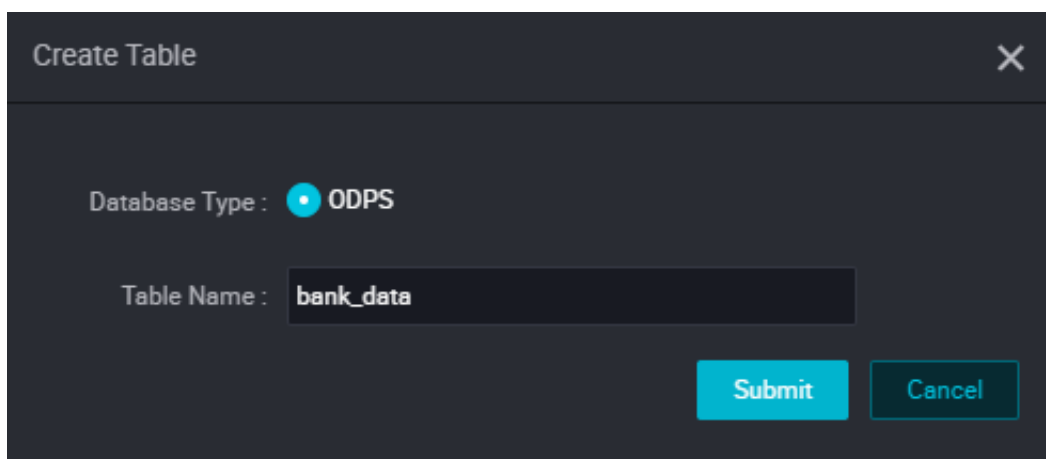
手順

`bank_data` という名前のテーブルを作成します。

1. 「[ワークスペースの作成](#)」後、該当プロジェクト内で [Enter workspace] をクリックします。
2. Data Studio (オリジナルデータ開発) ページにアクセスして、New > Table を 選択します。

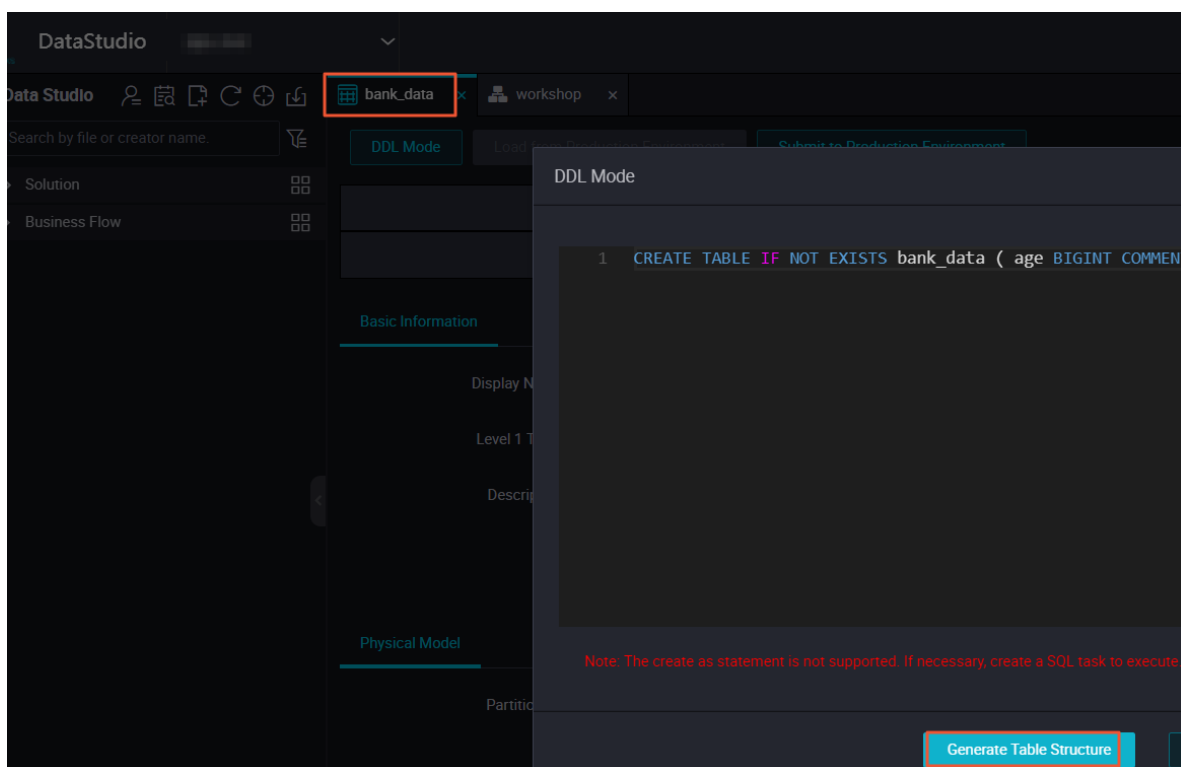


3. new table ダイアログボックスにて、テーブル名を入力します。

A dark-themed dialog box titled "Create Table" with a close button (X) in the top right corner. It contains a "Database Type" section with a radio button selected for "ODPS". Below this is a "Table Name" input field containing the text "bank_data". At the bottom right, there are two buttons: "Submit" and "Cancel".

4. [Submit] をクリックします。
5. new table ページにアクセスして DDL モードを選択します。
6. DDL schema ダイアログボックスにテーブル作成ステートメントを入力し、[build table structure] をクリックします。

テーブルを作成するための SQL 構文の詳細については、「[テーブルの作成/閲覧/削除](#)」をご参照ください。



この例で使用したテーブル作成ステートメントは以下のとおりです。

```
CREATE TABLE IF NOT EXISTS bank_data
(
  age BIGINT COMMENT 'age ',
```

```

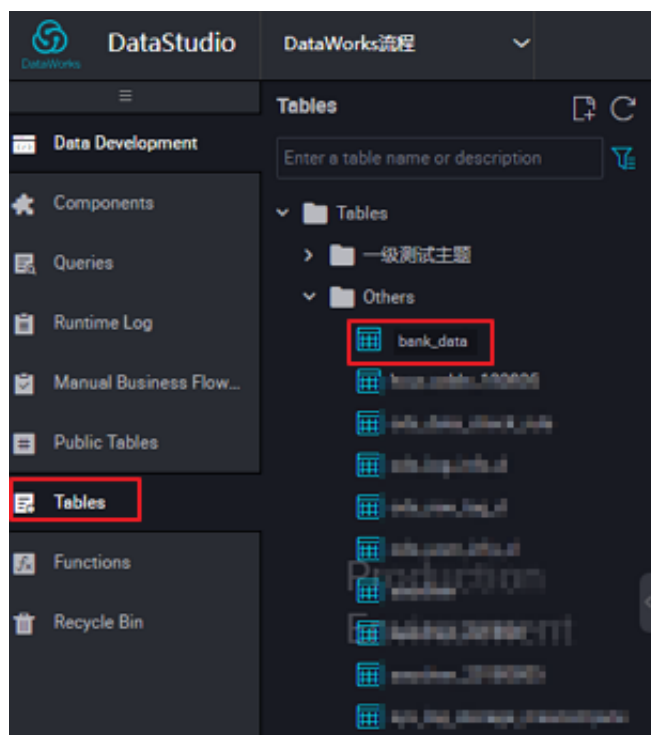
job          STRING COMMENT ' job   type ',
marital      STRING COMMENT ' marital status ',
education    STRING COMMENT ' educationa l level ',
default      STRING COMMENT ' credit card ownership ',
housing      STRING COMMENT ' mortgage ',
loan         STRING COMMENT ' loan ',
contact      STRING COMMENT ' contact informatio n ',
month        STRING COMMENT ' month ',
day_of_wee   k      STRING COMMENT ' day   of   the   week ',
duration     STRING COMMENT ' Duration ',
campaign     BIGINT COMMENT ' contact times during
the campaign ',
pdays       DOUBLE COMMENT ' time interval from the
last contact ',
previous     DOUBLE COMMENT ' previous contact times
with the customer ',
poutcome     STRING COMMENT ' marketing result ',
emp_var_ra   te     DOUBLE COMMENT ' employment change rate
',
cons_price   _idx   DOUBLE COMMENT ' consumer price index
',
cons_conf_   idx    DOUBLE COMMENT ' consumer confidence
index ',
euribor3m    DOUBLE COMMENT ' euro deposit rate ',
nr_employe   d      DOUBLE COMMENT ' number of employees ',
y            BIGINT COMMENT ' has time deposit or
not '
);

```

7. テーブル構造の生成が完了したら、テーブルの 名前を入力して、[Submit to Production Environment] をクリックします。

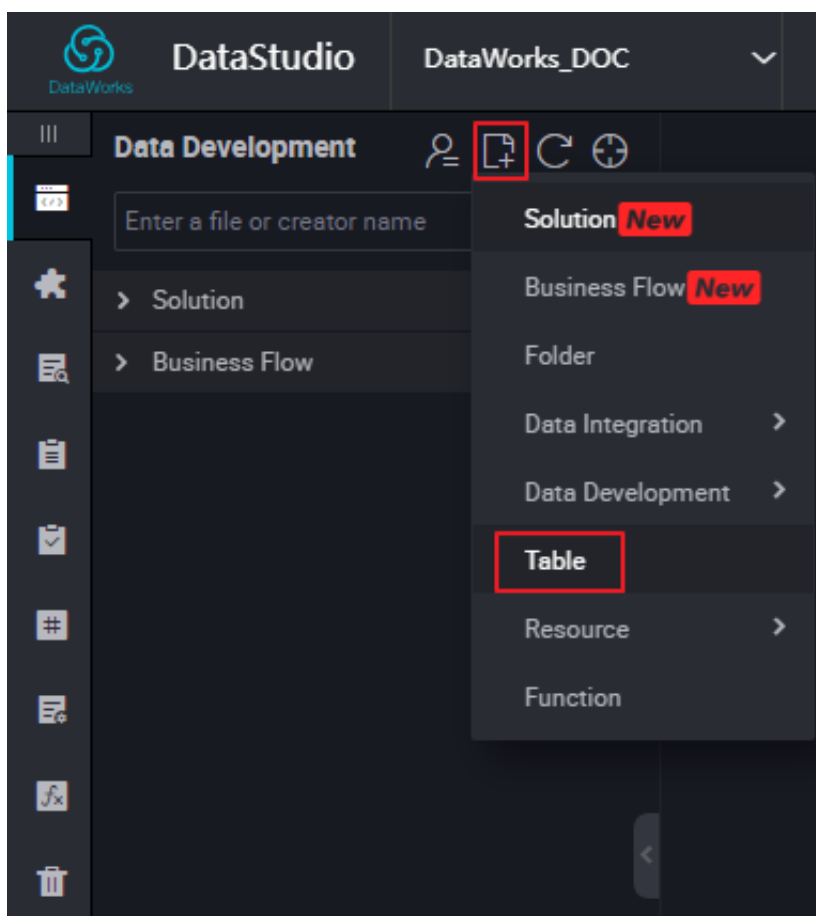
The screenshot shows the DataWorks console interface for creating a table. At the top, there's a tab labeled 'bank_data'. Below it, there are three buttons: 'DDL Mode', 'Load from Production Environment', and 'Submit to Production Environment' (which is highlighted with a red box). Under these buttons, there's a 'Table Name' field containing 'bank_data'. Below that, the 'Basics' tab is selected. In the 'Basics' section, there's a 'Table Alias' field with 'user_information'. Below that, there are two dropdown menus for 'Level 1 Topic' and 'Level 2 Topic', both set to 'Select'. To the right of these dropdowns are two buttons: 'Create Topic' and a refresh icon. At the bottom, there's a 'Description' field.

8. 左側ナビゲーションの **テーブルの管理** でテーブル名を入力することで、作成したテーブルを検索でき、テーブル情報を閲覧できます。



result_table の作成

1. DataStudio ページにアクセスしてnew > tableを選択します。



2. new table ダイアログボックスでテーブル名を入力して [Submit] をクリックします。
3. [new table] ページにアクセスして、DDL mode を選択します。
4. DDL schema ダイアログボックスで TABLE 構文を入力して、[build table structure] をクリックします。テーブル作成例を以下に示します:

```
CREATE TABLE IF NOT EXISTS result_table
(
  education STRING COMMENT 'educational level',
  num BIGINT COMMENT 'number of people'
);
```

5. 左側のナビゲーションバーで、table management 内にあるテーブル名より、作成したテーブルを検索でき、テーブル情報の閲覧が可能です。

bank_data へのローカルデータのアップロード

DataWorks は以下アクションをサポートします:

- ・ ローカルに格納されたテキストファイルをワークスペーステーブルにアップロード。
- ・ 多様なデータソースからワークスペースへ業務データをインポートするためのデータ統合。



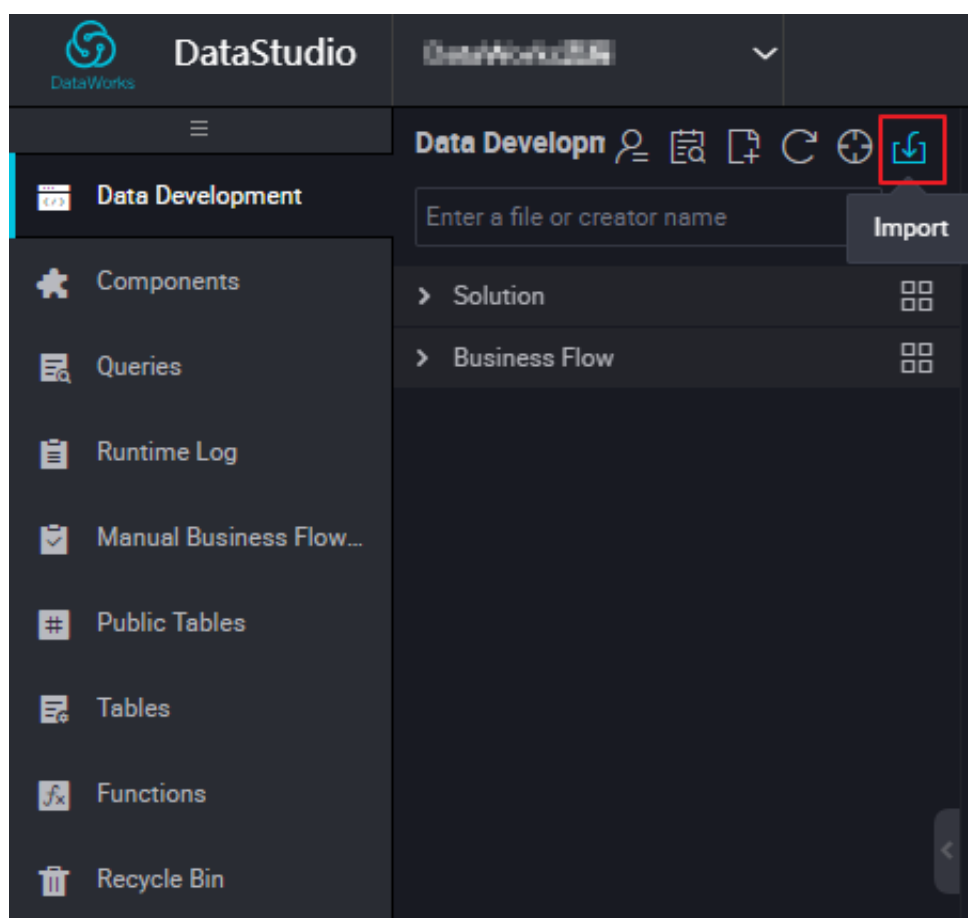
注:

ここでは、ローカルファイルはデータソースとして使用されます。ローカルテキストファイルのアップロードには、以下制限があります:

- ・ ファイルタイプ: 「.txt」と「.csv」ファイルのみがサポートされています。
- ・ ファイルサイズ: 10M 以内。
- ・ 操作オブジェクト: パーティションテーブルと非パーティションテーブルはインポート可能。中国語のパーティション値はサポート対象外。

例えば、インポートローカルファイル [banking.txt](#) を DataWorks にインポートする際の操作は以下のとおりです:

1. [Import] をクリックして、import local data を選択します。



- ローカルデータファイルを選択して、インポート情報を設定し、[Next] をクリックします。

Import Local Data

Selected Files: banking.txt Only .txt, .csv, and .log files are supported.

Separator: ☒ 逗号 ☐

Original Character: GBK

Import First Line: 1

In the first line are the following fields: ☒

由于数据量太大，只展示前100条

44	blue-collar	married	basic.4y	unknown	yes	no	cellular	aug	thu	210
53	technician	married	unknown	no	no	no	cellular	nov	fri	138
28	management	single	university.degree	no	yes	no	cellular	jun	thu	339

Next Cancel

- 名前ですべてのテーブルを検索する際は、少なくとも 2 文字以上を入力します。インポートするデータテーブルを選択します。ここでは例として bank_data を用います。

Import Local Data

Import to Table: bank

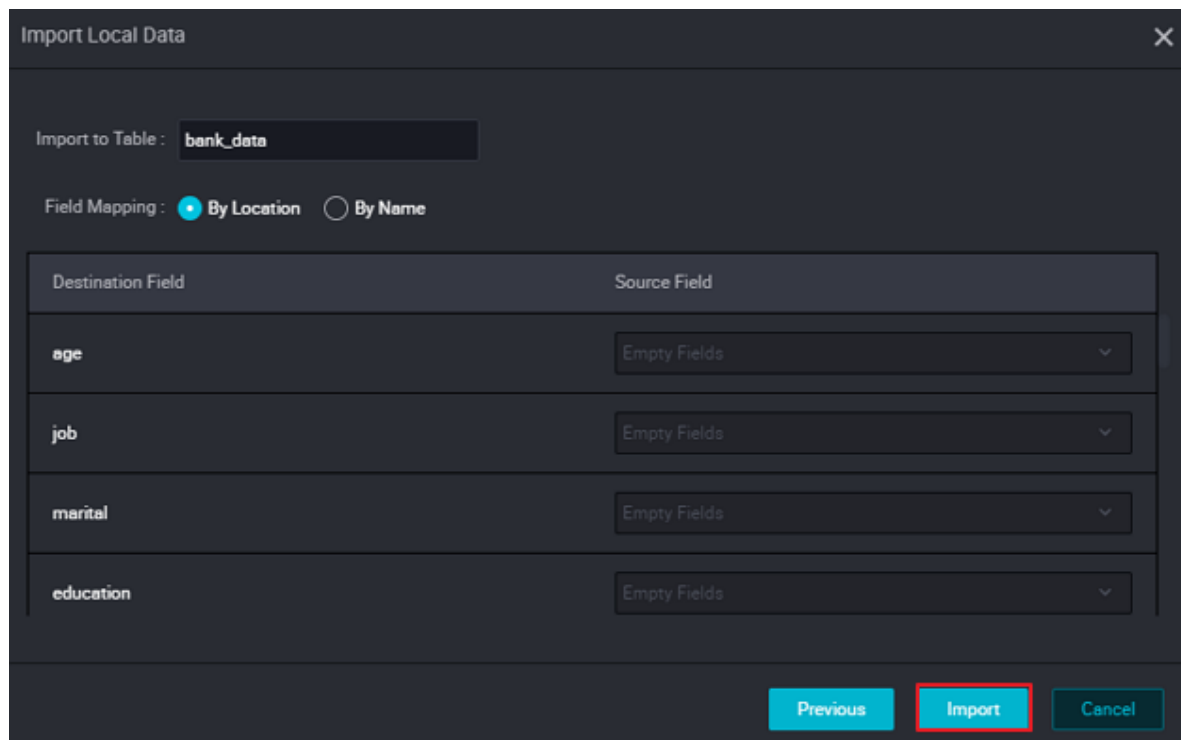
Field Mapping:

- bank_data1
- bank_data

Destination Field	Source Field
No data	

Previous Import Cancel

4. フィールド一致方法 (ここでは By Location を使用しています) を選択して、[Import] をクリックします。



Import Local Data

Import to Table : `bank_data`

Field Mapping : ☒ By Location ☐ By Name

Destination Field	Source Field
age	Empty Fields
job	Empty Fields
marital	Empty Fields
education	Empty Fields

Previous Import Cancel

ファイルをインポートしたら、システムはデータのインポートに成功した行数、または失敗した例外を返します。

その他のデータインポート方法

- データ同期タスクの作成

この方法は、異種のデータソース (Oracle、FTP、DM、HDFS、MongoDB など) から取得される RDS、MySQL、SQL Server、PostgreSQL、MaxCompute、OSS、DRDs、OSS データ等を格納する際に適用される方法です。

DataWorks でのデータ同期タスクの作成に関する詳細は、「[データ同期タスクの作成](#)」をご参照ください。

- ローカルファイルのアップロード

このファイルアップロード方法は 30M 以下の .txt と .csv ファイルに適用され、対象はパーティションテーブルと非パーティションテーブルの両方をサポートします。しかし、中国語のパーティションはサポートしません。

DataWorks ローカルファイルのアップロードについては、上述の bank_data へのローカルデータのアップロードをご参照ください。

- ・ tunnel コマンドを用いたアップロード

この方法はローカルファイルと 10M 以上のその他リソースファイルに適用されます。

「[MaxCompute クライアント](#)」によって提供された tunnel コマンドを通じてデータのアップロードとダウンロードを行い、ローカルデータファイルがパーションテーブルにアップロードされる必要がある際には、クライアントの tunnel コマンドを用いてアップロードが可能です。詳細は、「[Tunnel コマンド操作](#)」をご参照ください。

次のステップ

ここでは、テーブルの作成方法とデータアップロードの方法について説明しました。次のステップでは、より高度なデータ分析とプロジェクトスペースの計算を行うためのワークフローの作成方法について説明します。詳細は、「[ワークフローの作成](#)」をご参照ください。

3 ステップ 2: 業務フローの作成

本ドキュメントでは、業務フローにおけるノードの作成方法と依存関係の設定方法について説明します。これによりデータ分析の手順と順序の設計、表示が容易になります。まず、データ開発機能を使用したワークスペースデータの高度な分析と計算方法について簡単に説明します。

DataWorks のデータ開発機能は、業務フローで可視化されたドラッグアンドドロップをサポートしており、内部ノードの依存関係設定が可能です。データフローと依存関係は運用業務フロー形式で実施されます。現在サポートしているタスクタイプは、MaxCompute SQL、データ同期、open_mr、シェル、機械学習、仮想ノードなどです。タスクタイプ別の詳細な使用方法は、「[ノードタイプの概要](#)」をご参照ください。

前提条件

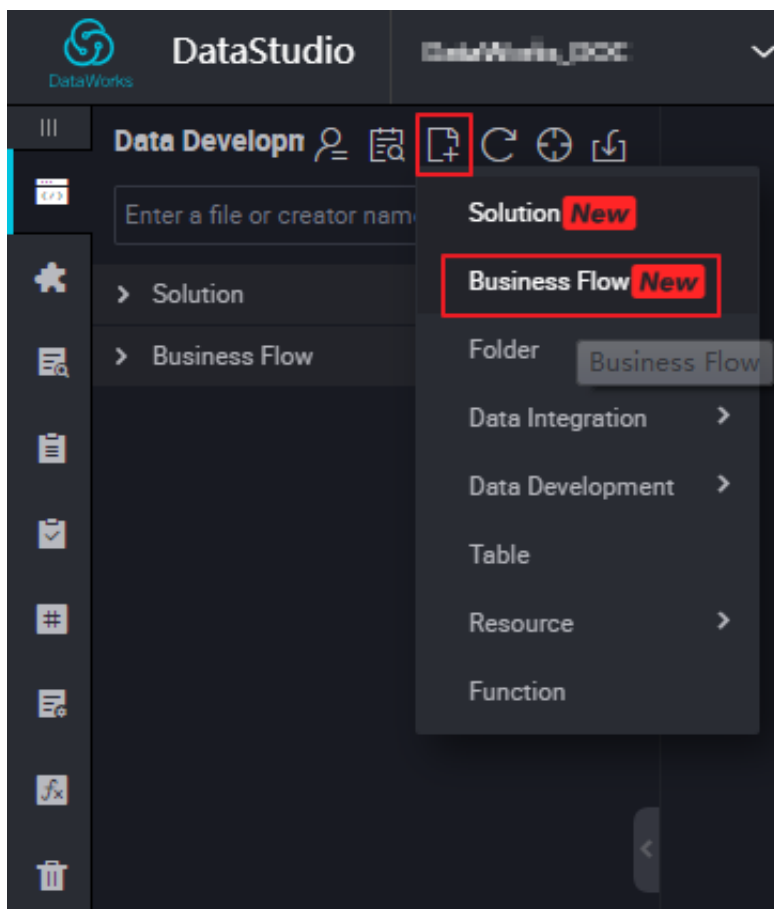
「[テーブル構築とデータのアップロード](#)」が完了したことを確認して、結果テーブルと同様に、業務データテーブル bank_data とワークスペース内のデータを準備します。

手順

業務フローの作成

1. 「[ワークスペースの作成](#)」の後に、該当プロジェクトの [Enter workspace] をクリックします。

2. DataStudio ページにアクセスして、create > business flow を選択します。



3. 業務フロー名とその説明文を入力します。

A screenshot of the 'Create Business Flow' dialog box. It has a title bar with 'Create Business Flow' and a close button. The main area contains two input fields: 'Business Name' with the value 'works' and 'Description' with the value 'quick start'. At the bottom right, there are two buttons: 'Create' (in blue) and 'Cancel' (in grey).

フローキャンバス上でのノードと依存関係の作成

ここでは、仮想ノード “start” と MaxCompute SQL ノード “insert_data” の作成方法、そして “start” に依存する “insert_data” の設定方法について説明します。

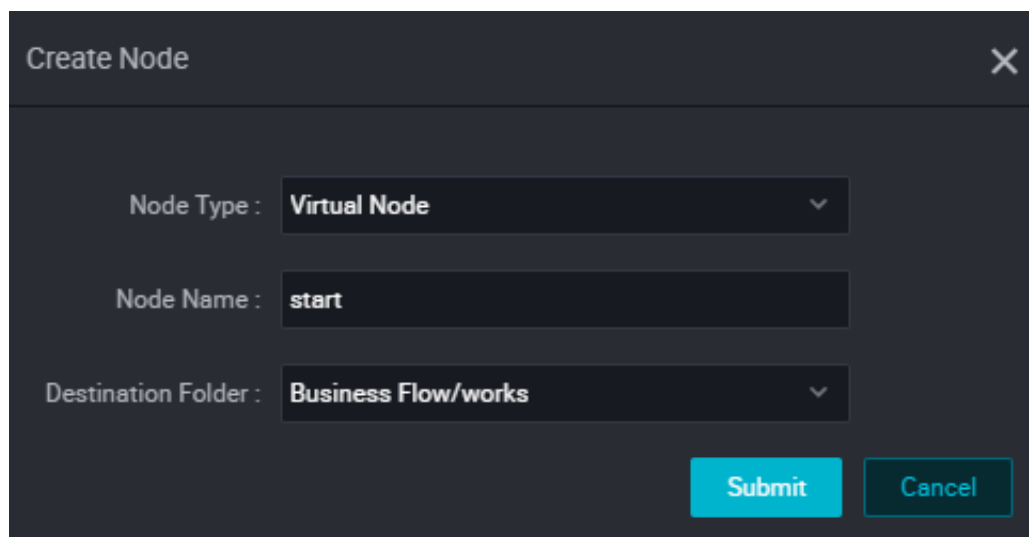


注:

- ・ 仮想ノードは制御式のノードで、業務フロー操作中にデータに影響を与えず、ダウンストリームノードの運用操作のみに使用されます。
- ・ 仮想ノードがその他のノードに依存して、運用担当者によってステータスが手動でエラーと設定された場合、未実行のダウンストリームノードがトリガーとなることはありません。これによって、運用フロー中の誤ったアップストリームデータの伝達を防ぎます。詳細は「[ノードタイプの概要](#)」内の仮想ノードセクションをご参照ください。
- ・ 業務フローでの仮想ノードのアップストリームタスクは、一般的にプロジェクトのルートノードとして設定され、プロジェクトルートノードの形式は Project name _ root となります。

フローを設計する際に、全体の業務フローを管理する仮想ノードをルートノードとして作成することを推奨します。

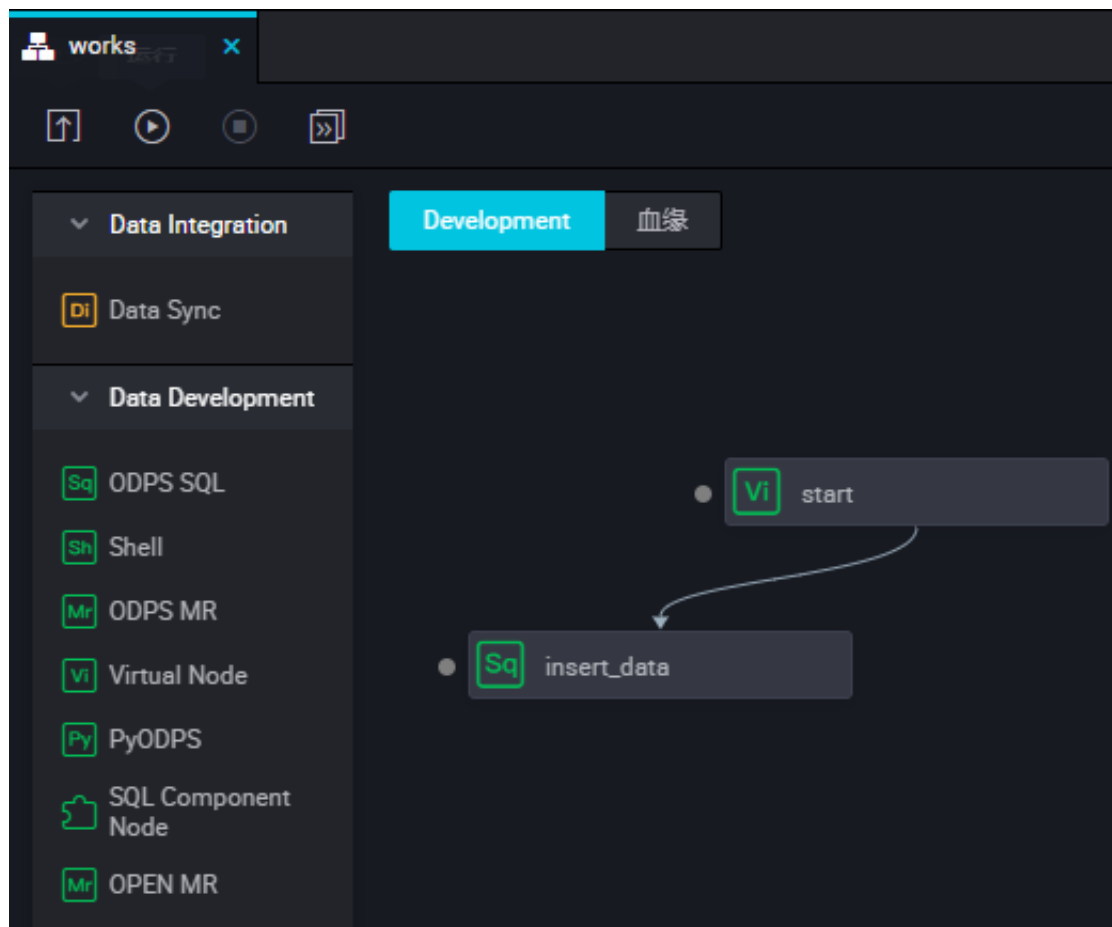
1. 仮想ノードをダブルクリックしてノード名 "start" を入力します。



The image shows a 'Create Node' dialog box with a dark background. It contains three input fields: 'Node Type' with a dropdown menu showing 'Virtual Node', 'Node Name' with the text 'start', and 'Destination Folder' with a dropdown menu showing 'Business Flow/works'. At the bottom right, there are two buttons: 'Submit' and 'Cancel'.

2. MaxCompute SQL をダブルクリックして、ノード名を "insert_data" と入力します。

3. start ノードをクリックして、start と insert_data 間に線を引き、下図に示すとおり insert_data を起点に依存関係を設定します。



MaxCompute SQL ノードでのコード編集

ここでは、MaxCompute SQL ノード insert_data での SQL コードの使用方法を説明します。学歴が異なる各個人の住宅ローン可否数の結果を問い合わせ、次に示すノードのとおり、結果を保存して分析や表示します。

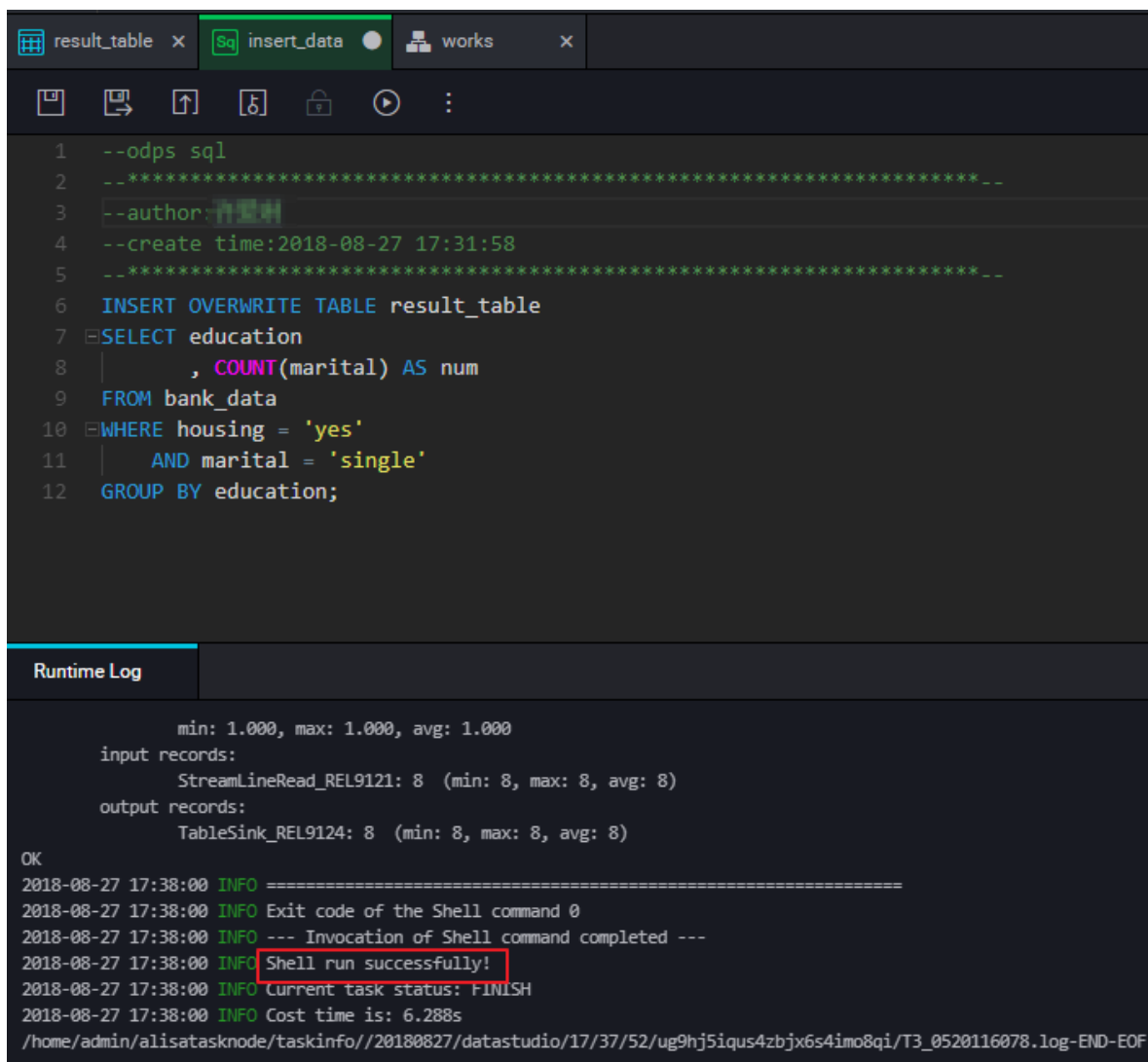
SQL 文は以下のとおりです。構文に関する詳細は、「[MaxCompute SQL](#)」をご参照ください。

```
INSERT OVERWRITE TABLE result_table -- Insert data to
result_table
SELECT education
, COUNT ( marital ) AS num
FROM bank_data
WHERE housing = ' yes '
AND marital = ' single '
GROUP BY education
```

MaxCompute SQL の実行とデバッグ

1. SQL 文の編集後、[Save] をクリックしてコードの損失を防止します。

2. [Run] をクリックして操作ログと結果を確認します。



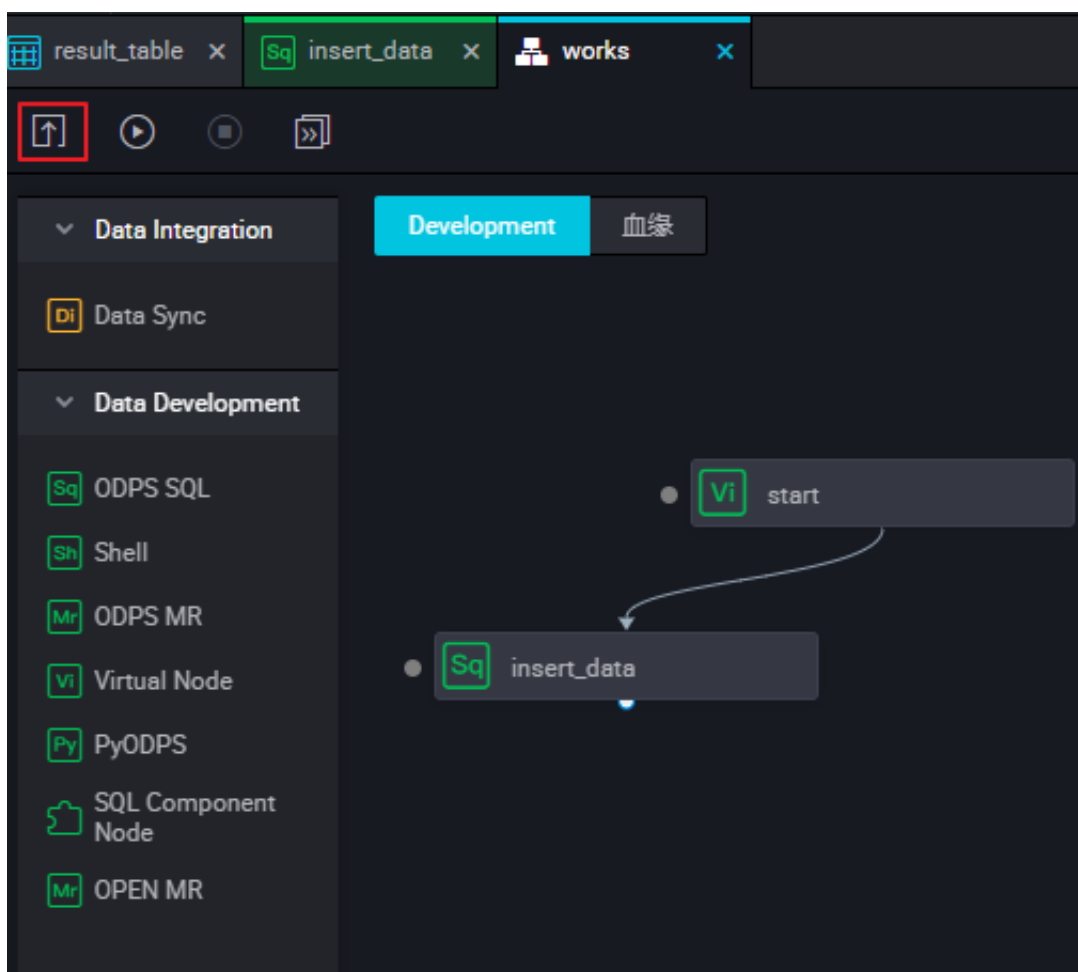
```
1  --odps sql
2  _*****_
3  --author: 竹園村
4  --create time:2018-08-27 17:31:58
5  _*****_
6  INSERT OVERWRITE TABLE result_table
7  SELECT education
8     , COUNT(marital) AS num
9  FROM bank_data
10 WHERE housing = 'yes'
11     AND marital = 'single'
12 GROUP BY education;
```

Runtime Log

```
min: 1.000, max: 1.000, avg: 1.000
input records:
  StreamLineRead_REL9121: 8 (min: 8, max: 8, avg: 8)
output records:
  TableSink_REL9124: 8 (min: 8, max: 8, avg: 8)
OK
2018-08-27 17:38:00 INFO =====
2018-08-27 17:38:00 INFO Exit code of the Shell command 0
2018-08-27 17:38:00 INFO --- Invocation of Shell command completed ---
2018-08-27 17:38:00 INFO Shell run successfully!
2018-08-27 17:38:00 INFO Current task status: FINISH
2018-08-27 17:38:00 INFO Cost time is: 6.288s
/home/admin/alisa/tasknode/taskinfo//20180827/datastudio/17/37/52/ug9hj5iqus4zbjx6s4imo8qi/T3_0520116078.log-END-EOF
```

業務フローの保存と送信

MaxCompute SQL ノード insert_data の実行とデバッグが完了したら、フローページに戻ります。全てのフローにおいて、[Save] と [Submit] をクリックします。



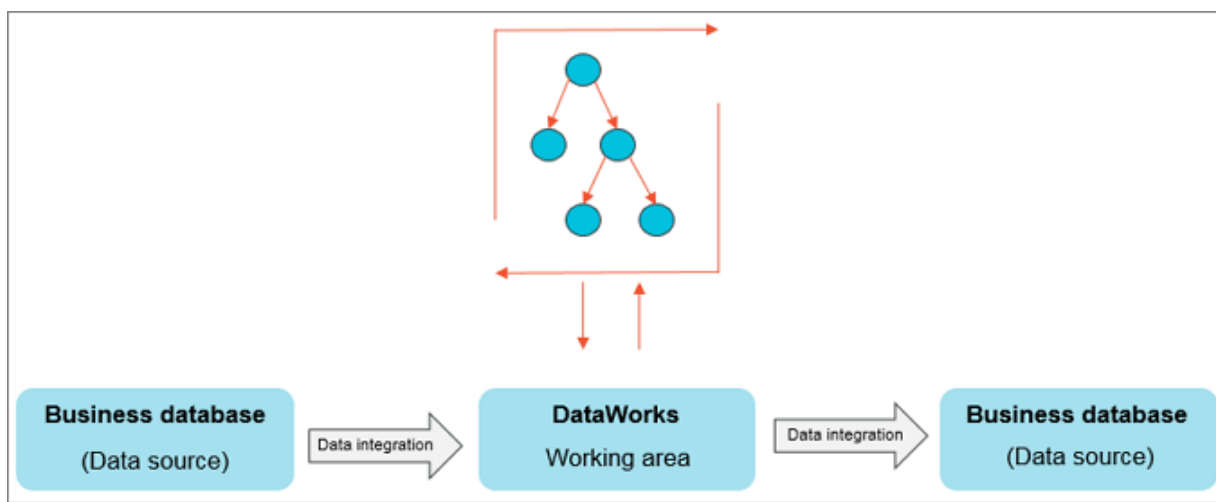
次のステップ

ここまで、業務フローの作成、保存、送信について説明しました。次のトピックでは、異種データソースへデータ出力する同期タスクの作成方法を説明します。詳細は、「[同期タスクエクスポート結果の作成](#)」をご参照ください。

4 ステップ 3: 同期タスクの作成

ここでは MySQL データソースを例に、データ統合機能を使った MaxCompute から MySQL データソースへのエクスポートの方法について説明します。

DataWorks においてデータ統合は一般的に、システムで生成した業務データを SQL タスク計算後のワークスペース内に定期的にインポートすることで使用されます。詳細情報または実行使用料の確認のために、計算結果は指定したデータソースに定期的にエクスポートされます。



データ統合機能によって、現在以下のデータソースのインポート、エクスポートが可能です: RDS、MySQL、SQL Server、PostgreSQL、MaxCompute、ApsaraDB for Memcache、DRDS、OSS、Oracle、FTP、DM、Hdfs、MongoDB 等。詳細は、『[サポートされるデータソース](#)』をご参照ください。

前提条件

- ・ ECS 上でオンプレミスのデータベースを使用している場合、ECS に [[セキュリティグループの追加](#)] を行う必要があります。
- ・ RDS や MongoDB などのデータソースを使用している場合、データソースコンソールに [[ホワイトリストの追加](#)] を行う必要があります。



注:

カスタムのリソースグループを使用して RDS データ同期タスクをスケジュールする場合、カスタムのリソースグループをホストするコンピューターの IP アドレスを RDS ホワイトリストに追加する必要があります。

手順

データソースの追加



注:

プロジェクト管理者ロールのみが新規データソースを作成でき、その他ロールメンバーはデータソースの閲覧のみ可能です。

1. プロジェクト管理者として [DataWorks 管理コンソール] にログインします。
2. [項目リスト] 下の該当するアクション項目列にある [ワークスペースに参加] を選択します。
3. 上部メニューバーにある [データの統合] をクリックします。
4. 左側のナビゲーションペインで [データソース] をクリックします。
5. 右上隅の [プロジェクトの作成] をクリックします。

The screenshot displays the 'Data Integration' console interface. On the left, a sidebar contains navigation options: Overview, Tasks, Resource Consumption, Synchronization Resource, **Data Source** (highlighted), Resource Group, and Client Data Collection. The main content area is titled 'Data Source' and includes a dropdown menu set to 'All'. Below this is a table listing data sources. The table has columns for Data Source Name, Data Source Type, Link Information, Description, Created At, and Actions. One data source is listed: 'odps_first' with type 'ODPS'. The description for this source is 'connection from odps calc engine 61155'. In the top right corner of the main area, there is a red button labeled 'Add Data Source'.

Data Source Name	Data Source Type	Link Information	Description	Created At	Actions
odps_first	ODPS	ODPS Endpoint: https://odps.aliyun.com/odps/console/odps/61155 ODPS Project Name: odps-project-61155 Access Id: odps-project-61155	connection from odps calc engine 61155	2018-08-27 13:32:26	

6. [データソース追加] ダイアログボックスの各設定項目に入力します。

Add Data Source MySQL

* Data Source Type: Has Public Network IP

* Data Source Name: clone_database

Description: add data source

* JDBC URL: jdbc:mysql://host:port/database

* Username: slapd@

* Password:

Test Connectivity: Test Connectivity

① Ensure that the database is available.
Ensure that the firewall allows the data sent from or to the database to pass by.
Ensure that the database domain name can be resolved.
Ensure that the database has been started.

Previous Finish

- ・ データソースの種類: パブリック ID アドレス
- ・ データソース名: 名称には文字、数字、アンダーラインを含む必要があり、数字またはアンダーラインで開始することはできません。例: abc_1123
- ・ データソースの記述: 80 文字以内で記述します。
- ・ JDBC URL: jdbc : mysql : // host : port / database .
- ・ ユーザー名/パスワード: データベースへの接続時に使用されるユーザ名とパスワードです。

異なるデータソース種類の設定指示については『[データソースの設定](#)』をご参照ください。

7. (任意) 関連項目の必要情報を入力したら、[接続テスト] をクリックします。

8. テスト接続が完了したら、[完了] をクリックします。



注:

対象の MySQL データベースがテーブルを含んでいるか確認します。

MySQL データベースで `odps_result` テーブルを作成します。テーブル作成時に使用されるステートメントは以下のとおりです。

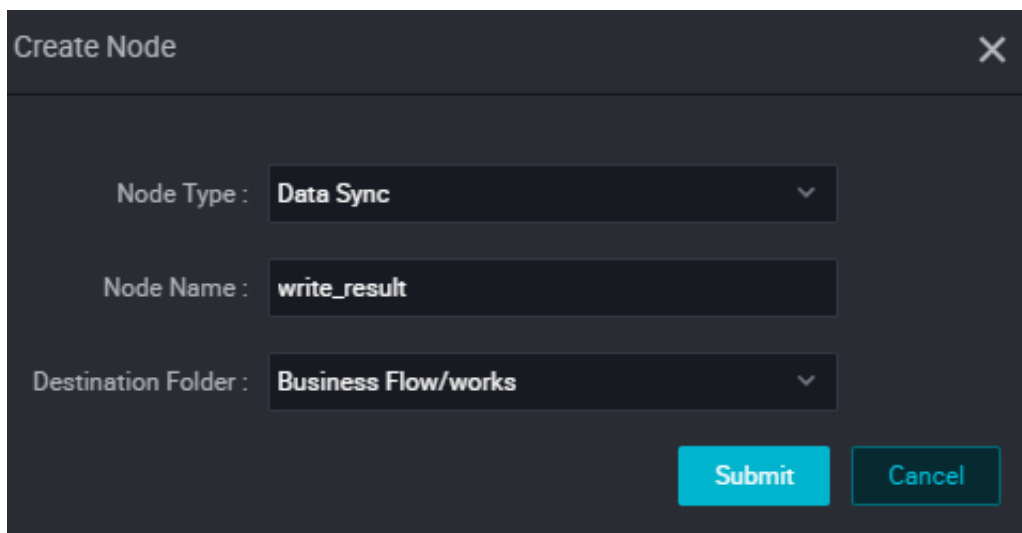
```
CREATE TABLE `ODPS_RESULT` (  
  `education` varchar(255) NULL ,  
  `num` int(10) NULL  
)
```

テーブルの作成が完了したら、`desc odps_result;` を実行してテーブルの詳細を閲覧できます。

同期ノードの作成と設定

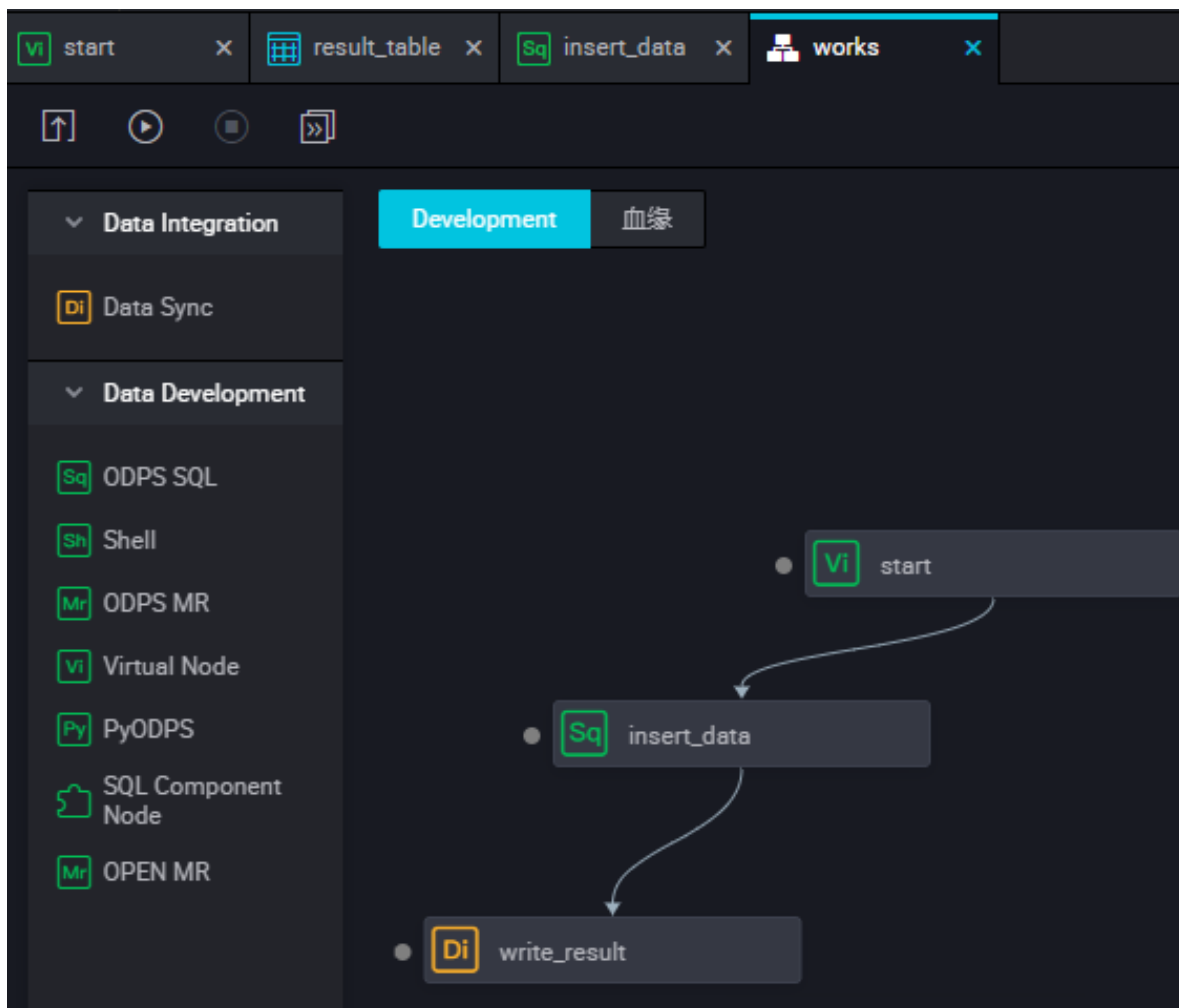
ここでは、`write_result` の同期ノードの作成と設定について、また、`result_table` からのデータを MySQL データベースに書き込む方法について説明します。詳細な手順は以下のとおりです。

1. 以下の図に示すとおり、ノードを作成します。



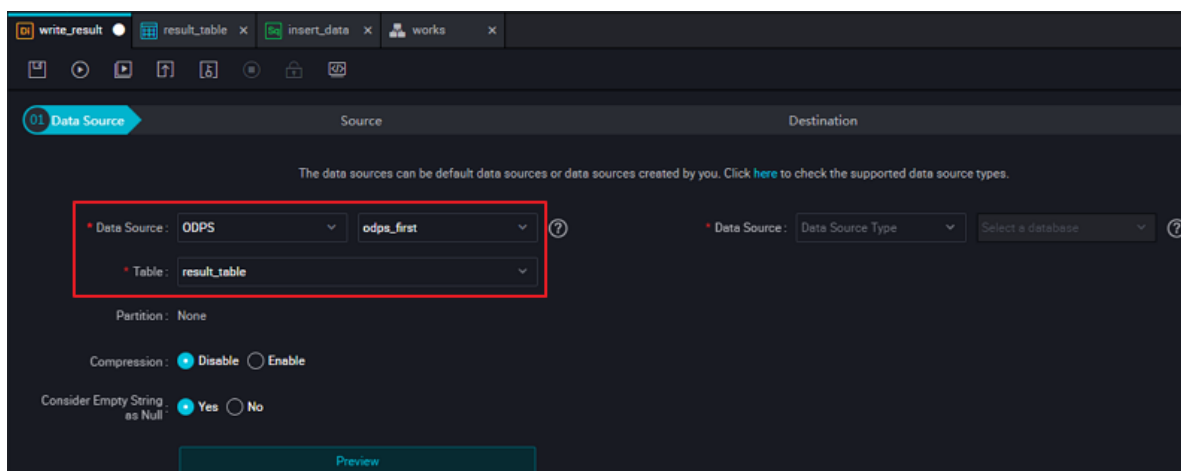
The screenshot shows a 'Create Node' dialog box. It has a title bar with 'Create Node' and a close button. The dialog contains three fields: 'Node Type' with a dropdown menu showing 'Data Sync', 'Node Name' with a text input field containing 'write_result', and 'Destination Folder' with a dropdown menu showing 'Business Flow/works'. At the bottom right, there are two buttons: 'Submit' and 'Cancel'.

2. write_result ノードが insert_data ノードに依存するような、ノード間の依存関係を設定します。



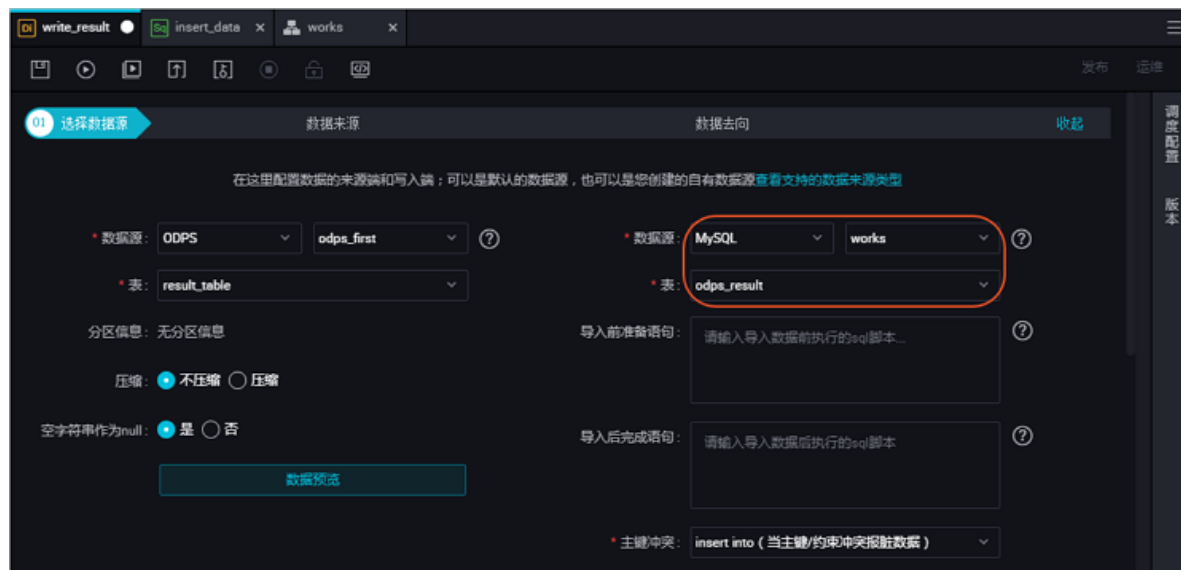
3. データソースを選択します。

MaxCompute のデータソースとソーステーブルを選択して [次へ] をクリックします。



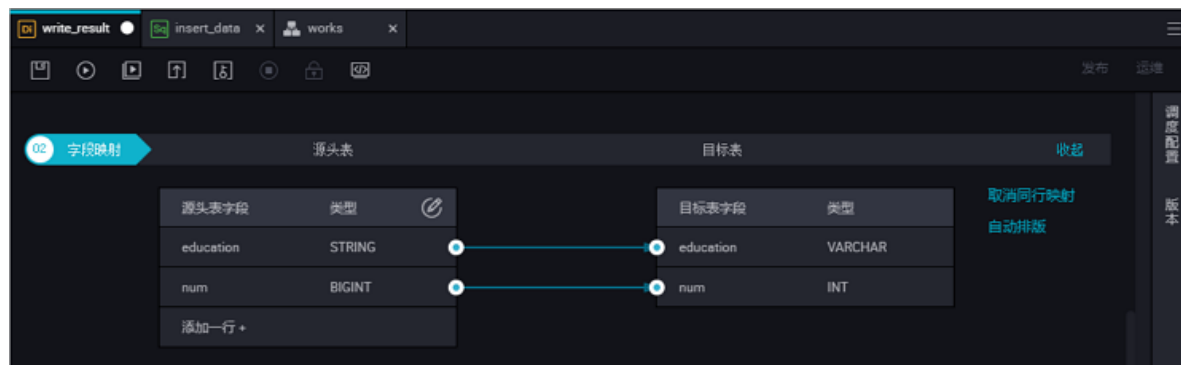
4. ターゲットを選択します。

MySQL データソースと対象テーブルの ODPS _resultを選択して [次へ] をクリックします。



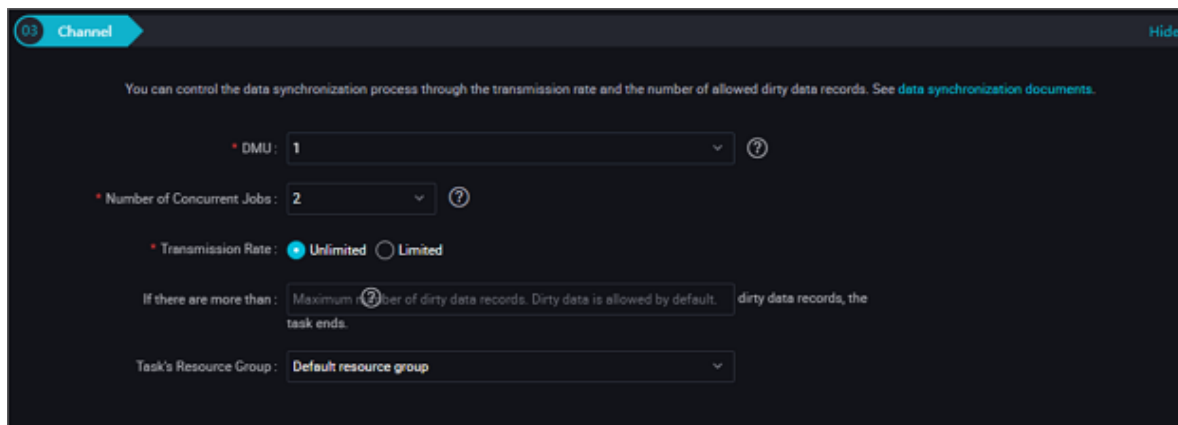
5. フィールドをマップします。

フィールド間のマッピングを選択します。フィールドのマッピング関係を設定する必要があります。左側の "ソーステーブルフィールド" は、右側の "ターゲットテーブルフィールド" と 1 対 1 で対応しています。



6. チャンネルを操作します。

[次へ] をクリックして、最大ジョブレートとダーティデータチェックルールを設定できます。



03 Channel Hide

You can control the data synchronization process through the transmission rate and the number of allowed dirty data records. See [data synchronization documents](#).

* DMU: 1 ?

* Number of Concurrent Jobs: 2 ?

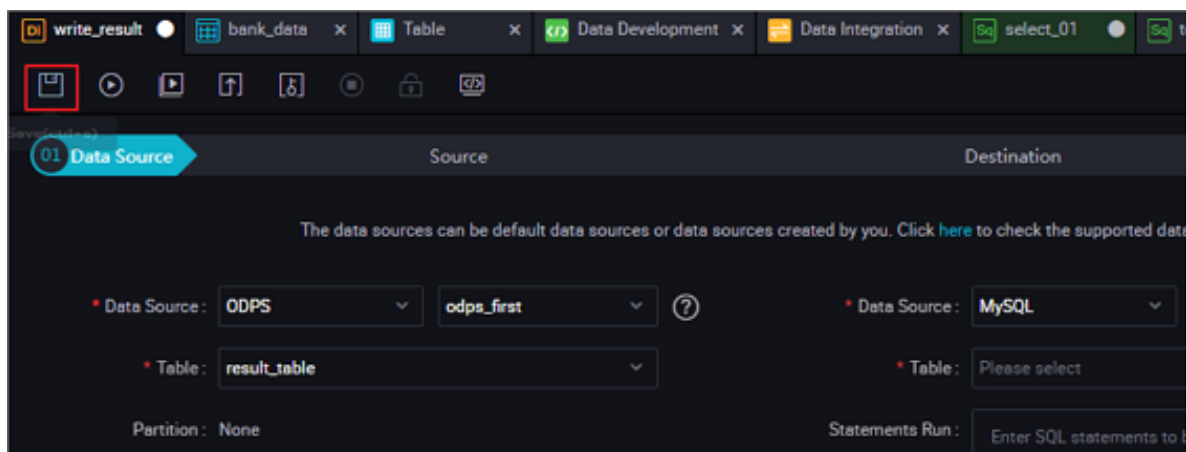
* Transmission Rate: ☒ Unlimited ☐ Limited

If there are more than: dirty data records, the task ends.

Task's Resource Group: Default resource group

7. プレビューと格納。

上記の設定が完了したら、タスク設定を閲覧するためにマウスを上下にスクロールします。設定が完了していない場合は、[保存] をクリックします。



write_result bank_data Table Data Development Data Integration select_01

01 Data Source Source Destination

The data sources can be default data sources or data sources created by you. Click [here](#) to check the supported data sources.

* Data Source: ODPS odps_first ?

* Table: result_table

Partition: None

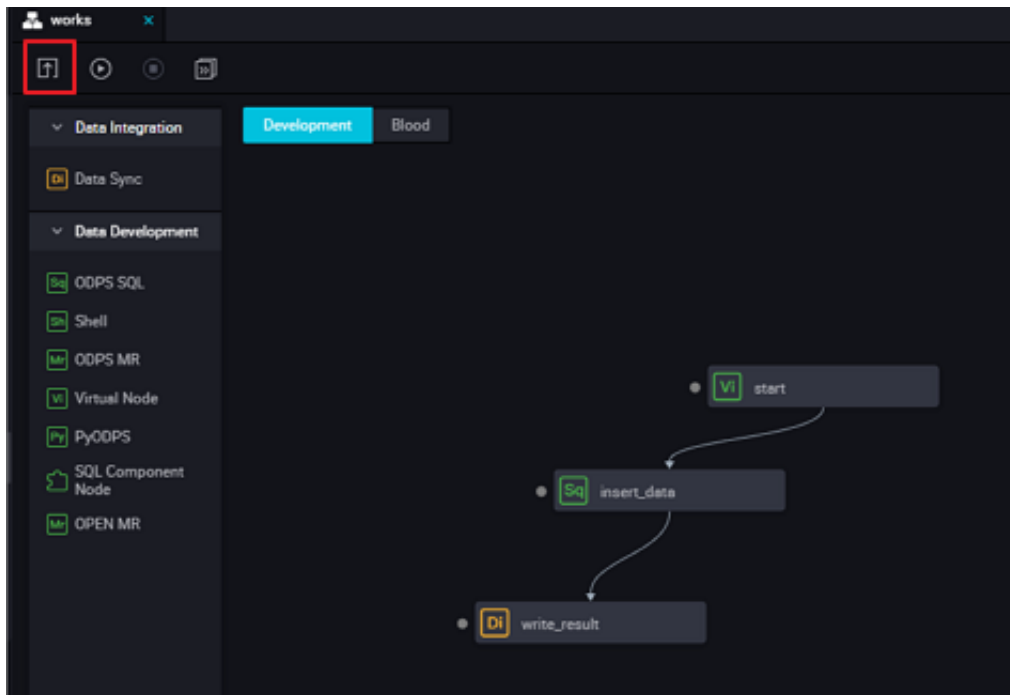
* Data Source: MySQL

* Table: Please select

Statements Run: Enter SQL statements to be executed

データ同期タスクの送信

同期タスクを保存したら、[送信] をクリックしてスケジューリングシステムにタスクを送信します。スケジューリングシステムは、設定属性に従ってジョブを2日目から自動的かつ定期的に実行します。



次のステップ

ここでは、同期タスクの作成と異なるデータソースへのデータエクスポートの方法について説明しました。続いて、同期タスクのスケジュール属性と依存関係の設定に関して説明します。詳細はタスクの『[スケジュールプロパティと依存関係の設定](#)』をご参照ください。

5 手順 4: スケジューリングと依存関係設定

本ドキュメントでは、「[同期タスクの作成](#)」で作成された "write_result" を例に取り、そのスケジュールサイクルを毎週のスケジュールとして設定し、DataWorks のスケジュール設定とタスク動作機能を紹介します。

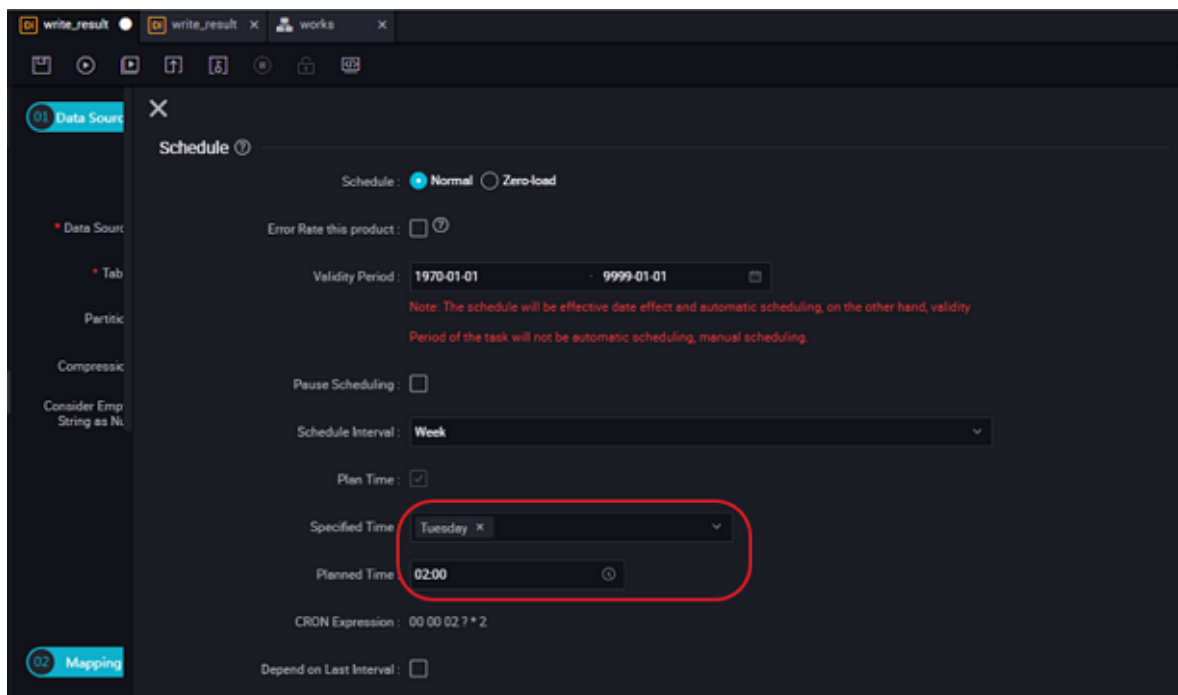
DataWorks は、DAG の関係に基づいて、毎日数千万のタスクを正確かつ適時に実行するための時間ベースまたは依存関係ベースのタスクトリガー機能を含む強力なスケジューリング機能を提供します。分、時間、日、週、月単位のスケジューリングをサポートしています。詳しくは、「[同期タスクの作成](#)」をご参照ください。

手順

同期タスクのスケジューリング属性を設定

1. 「Data Development > Task Development」ページを選択します。
2. 設定したい同期タスク (write_result) をダブルクリックします。

3. 右側の [Scheduling Configuration] をクリックして、タスクのスケジュールプロパティを設定します。

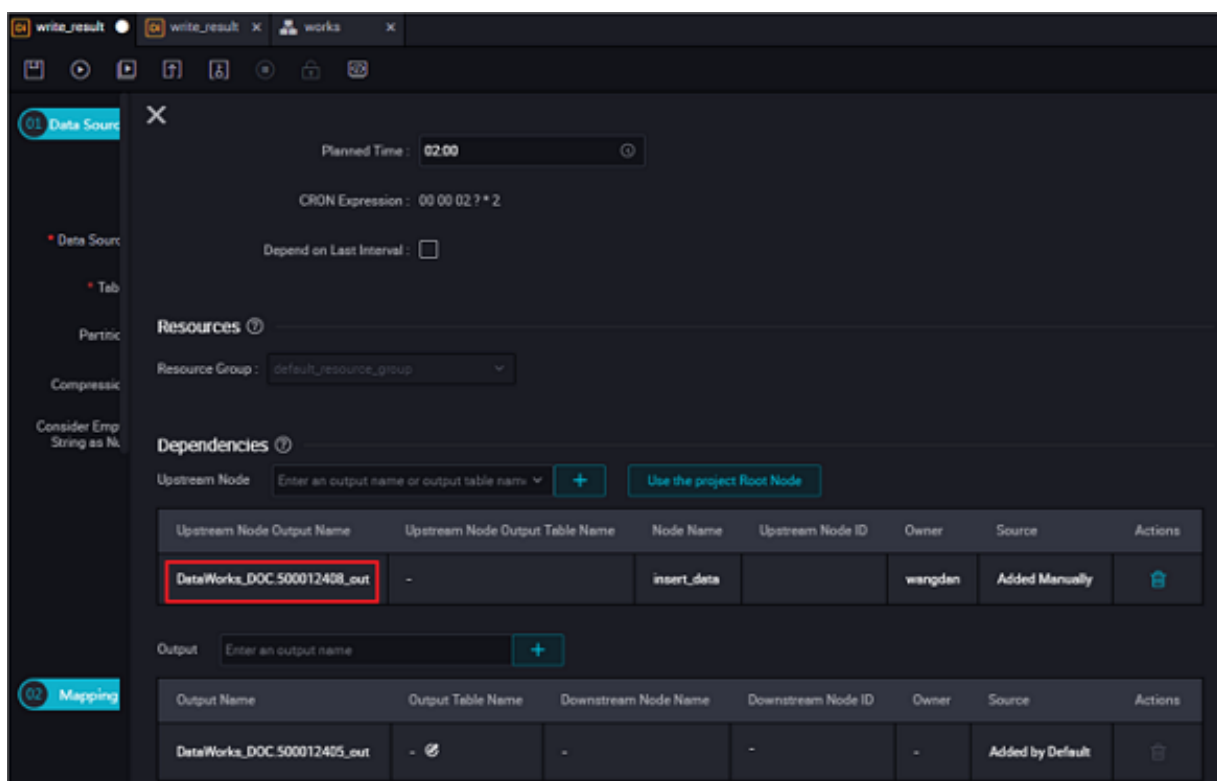


パラメーター：

- ・ スケジュール状態：このパラメーターを選択すると、タスクは一時停止します。
- ・ エラー再試行：このパラメーターを選択すると、エラー再試行が有効になります。
- ・ 開始日：タスクが有効になる日付は要件に基づいて設定できます。
- ・ スケジュール期間：タスクの動作サイクルは月、週、日、時間、分で設定できます。たとえば、タスクを毎週スケジュールすることができます。
- ・ 特定時間：特定タスクの動作時間。たとえば、毎週火曜日の 02 時 00 分にタスクを実行するように設定できます。

同期タスクの依存関係プロパティの設定

同期タスクスケジュールプロパティの設定が完了したら、その配置依存関係プロパティを設定できます。



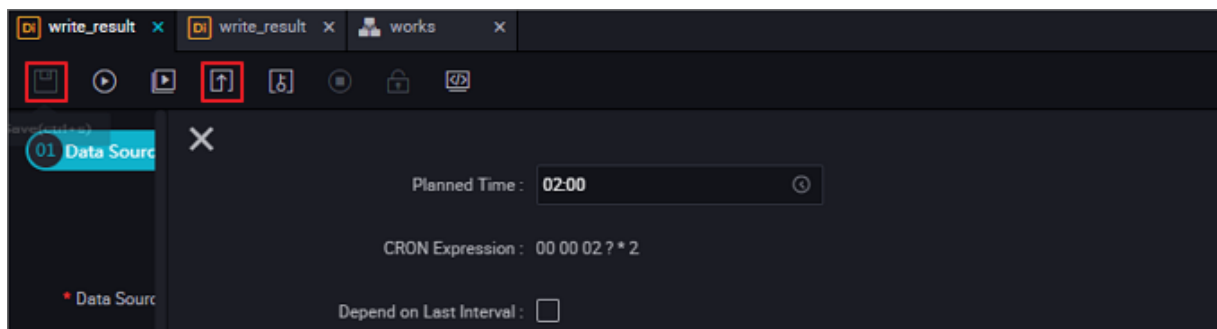
タスクに対してアップストリームの依存関係を設定できます。この場合、現在のタスクのインスタンスがスケジュールされた時間に達しても、タスクはインスタンスのアップストリームタスクが完了した後にのみ実行されます。

上の図の設定は、現在のタスクのインスタンスが、アップストリームタスク `write_result` が終了した後にのみ開始されることを示しています。アップストリームタスクに「work」を入力して、`write_result` に対しアップストリームタスクを設定できます。

アップストリームタスクが設定されていない場合は、デフォルトで現在のタスクがプロジェクトにより開始されます。その結果、現在のタスクのデフォルトのアップストリームタスクは、スケジューリングシステムの `project_start` です。デフォルトでは、`project_start` タスクは各プロジェクトのルートタスクとして作成されます。

データ同期タスクの送信

同期タスク `write_result` を保存し、[Submit] をクリックしてそれをスケジューリングシステムに送信します。



システムは、スケジューリング属性設定に従って各時点でタスクのインスタンスを自動的に生成し、タスクがスケジューリングシステムに送信された後に限り、3 日目からタスクを定期的に実行します。



注：

タスクが 23 時 30 分以降に送信された場合、スケジューリングシステムは 3 日目から自動的にインスタンスをサイクル生成し、時間どおり実行します。

次のステップ

同期タスクのスケジューリング属性と依存関係を設定する方法は、これで理解できました。次のトピックでは、送信されたタスクの定期的な O&M の実行方法とログのトラブルシューティングの結果の確認方法を説明します。詳しくは、「[定期運用とログのエラーリンク付けの確認](#)」をご参照ください。

6 手順 5 : O&M とログのトラブルシューティング結果の表示

本ドキュメントでは、タスク動作を実装する方法について説明します。

前の操作では、毎週火曜日の 02 時 00 分に同期タスクを実行するように設定しました。タスクが送信された後、次の日にスケジュールシステムで自動動作結果を表示できます。

インスタンスのスケジュールと依存関係が想定どおりに動作しているかどうかを確認するために、DataWorks では、テスト実行、データ作成、および定期実行の 3 つの起動方法が用意されています。

- ・ **テスト実行**：タスクは手動でトリガーされます。単一タスクのタイミングと動作を確認する必要がある場合は、テスト実行をお勧めします。
- ・ **データ作成**：タスクは手動でトリガーされます。この方法は、複数のタスクのタイミングと依存関係を確認する必要がある場合、またはルートタスクからデータ分析と計算を再実行する必要がある場合に適用されます。
- ・ **定期実行**：タスクは自動的にトリガーされます。送信が成功すると、スケジューリングシステムは次の日の 00 時 00 分から始まるさまざまなタイミングでタスクインスタンスを自動的に生成します。スケジューリングシステムはスケジュールされた時間に従って、各インスタンスのアップストリームインスタンスが正常に実行できているかどうかを確認します。すべてのアップストリームインスタンスがスケジュールされた時間に正常に実行されると、現在のインスタンスが自動的に実行されます。



注：

スケジューリングシステムは、手動トリガーモードと自動トリガーモードの両方に適用される規則と同じ規則に基づいてインスタンスを定期的に生成します。

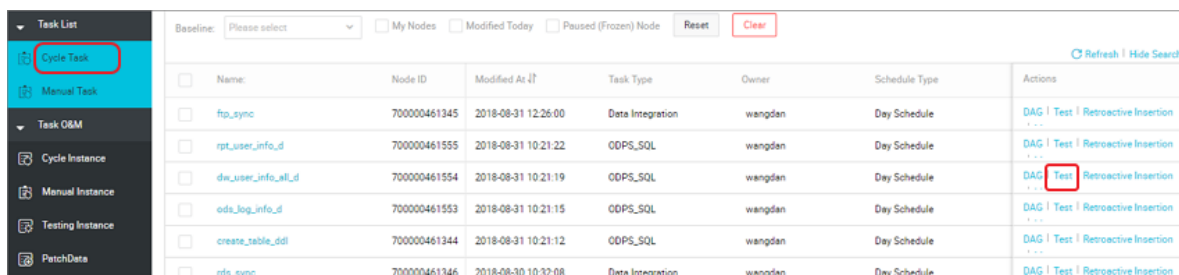
- ・ 期間は、月、週、日、時間、さらには分単位で設定できます。スケジューリングシステムは常に指定された日付または指定された時間にタスクのインスタンスを生成します。
- ・ スケジューリングシステムは、指定された日付に定期的にインスタンスを実行し、動作ログを生成します。
- ・ 指定された日付でないインスタンスは実行されず、実行条件が満たされるとき、それらの状態は直接、” 成功 ” に変更されます。そのため、実行中のログは生成されません。

動作および機能の詳細については、「[タスクの操作](#)」をご参照ください。

テスト

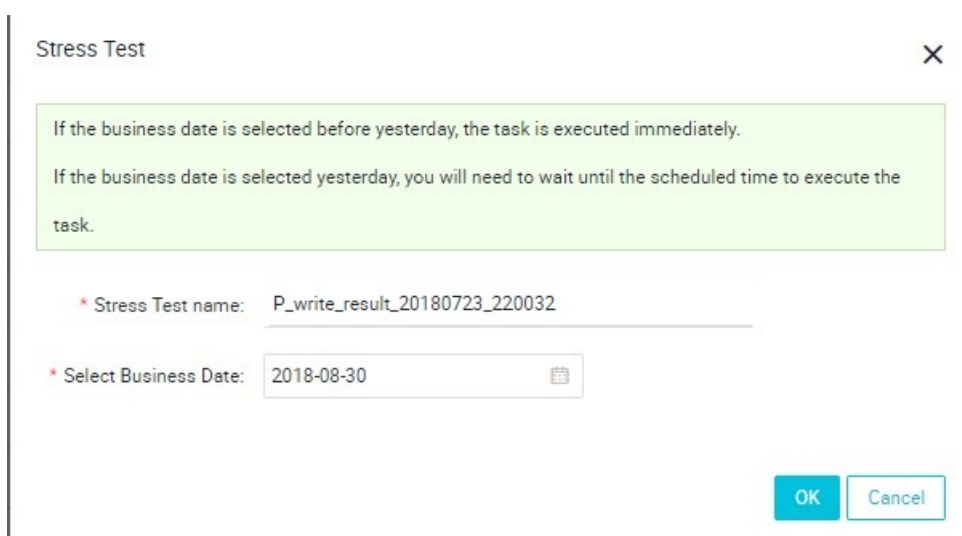
手動でテストをトリガーする

1. 「Cycle Task」 ページで、実行するタスクを見つけて [Test] をクリックします。



Name	Node ID	Modified At	Task Type	Owner	Schedule Type	Actions
ftp_sync	700000461345	2018-08-31 12:26:00	Data Integration	wangdan	Day Schedule	DAG Test Retroactive Insertion
rpt_user_info_d	700000461555	2018-08-31 10:21:22	ODPS_SQL	wangdan	Day Schedule	DAG Test Retroactive Insertion
dms_user_info_all_d	700000461554	2018-08-31 10:21:19	ODPS_SQL	wangdan	Day Schedule	DAG Test Retroactive Insertion
ads_log_info_d	700000461553	2018-08-31 10:21:15	ODPS_SQL	wangdan	Day Schedule	DAG Test Retroactive Insertion
create_table_ddl	700000461344	2018-08-31 10:21:12	ODPS_SQL	wangdan	Day Schedule	DAG Test Retroactive Insertion
rds_sync	700000461346	2018-08-30 10:32:08	Data Integration	wangdan	Day Schedule	DAG Test Retroactive Insertion

2. 営業日を入力して [OK] をクリックします。



Stress Test

If the business date is selected before yesterday, the task is executed immediately.

If the business date is selected yesterday, you will need to wait until the scheduled time to execute the task.

* Stress Test name: P_write_result_20180723_220032

* Select Business Date: 2018-08-30

OK **Cancel**

3. 「Basic information」 ページに移動して、タスク実行状態を表示します。



Task Name	ID	Duration	Status
rds_sync	ID: 700000461346	Dur: 08-31 00:12:04 ~ 00:13:38 (dur 1m34s)	Completed
ftp_sync	ID: 700000461345	Dur: 08-31 00:12:16 ~ 00:15:31 (dur 3m15s)	Completed
create_table_ddl	ID: 700000461344	Dur: 08-31 00:11:05 ~ 00:11:49 (dur 44s)	Completed
workshop_start	ID: 700000461343	Dur: 08-31 00:05:10 ~ 00:05:10 (dur 0s)	Completed
rpt_user_info_d	ID: 700000461555	Dur: Idle	Pending

Production Environment. Please be cautious.

```

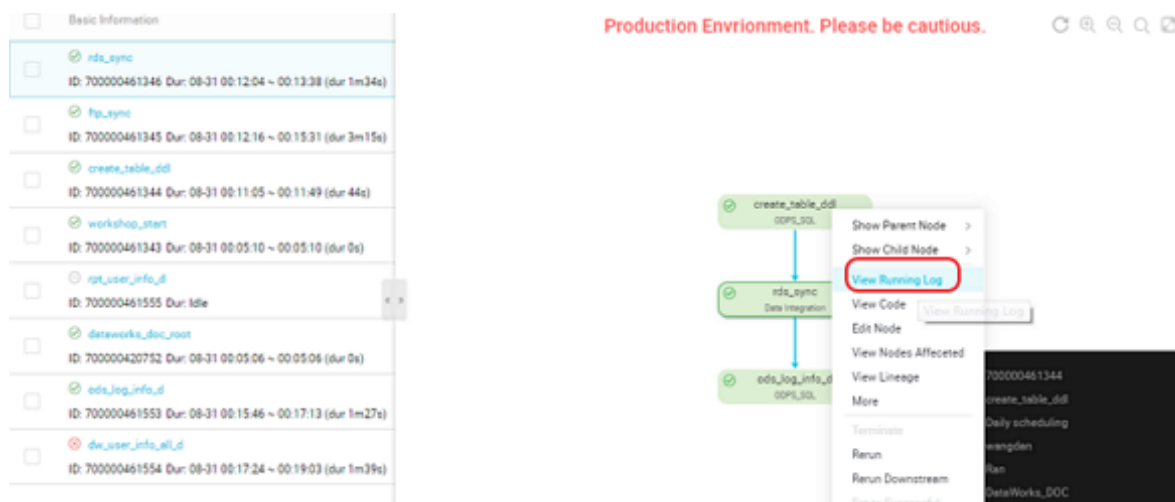
graph TD
    A[create_table_ddl  
ODPS_SQL] --> B[rds_sync  
Data Integration]
  
```

テストインスタンスの情報と動作ログの表示

「テストインスタンス」 ページで適切なタスクインスタンスを選択してをクリックすると、インスタンス DAG グラフを表示できます。

- ・ インスタンスを右クリックすると、インスタンスの依存関係の詳細を表示したり、停止、再開などの特定のアクションを実行できます。

- ・ インスタンスをダブルクリックして、タスクのプロパティ、実行ログ、動作ログ、コードなどを示すポップアップが表示されます。



注：

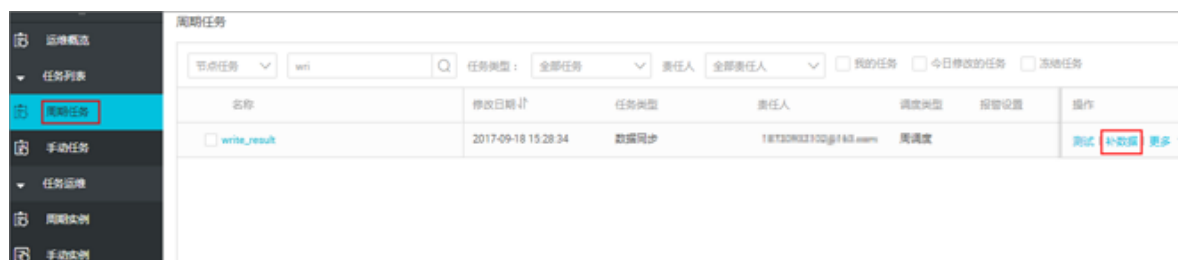
- ・ テスト実行モードでは、タスクは手動でトリガーされます。インスタンスのアップストリームの依存関係に関係なく、設定された時間に達したら、すぐにタスクが実行されます。
- ・ タスク write_result は、本ドキュメントで前述したインスタンス生成ルールに基づいて、毎週火曜日の朝に実行されるように設定されています。テストランタイムによって選択された営業日は月曜です（営業日 = 実行日 - 1）。インスタンスは実際には2時に実行されます。月曜日でない場合、インスタンスは2時の時点で成功状態に変換され、ログは生成されません。

補充データ操作

手動でデータ生成をトリガーする

複数のタスクのタイミングと依存関係を確認する必要がある場合、またはルートタスクからデータ分析計算を再実行する必要がある場合は、「O&M センター > タスクリスト > 周期タスク」ページを選択し、該当タスクの横に [replenishment data] をクリックして、一定期間にスケジュールされた複数のタスクに入力します。

1. 「O&M センター > サイクルタスク」ページを選択して、タスク名を入力します。
2. クエリ結果の後に [replenishment data] をクリックします。



3. 補充データの営業日を "to" に設定し、write_result ノードタスク] を選択して、[OK] をクリックします。

4. [to view the replenishment data results] をクリックします。

データ生成インスタンスの情報と動作ログの表示

適切なタスクインスタンスを選択することで、インスタンス DAG グラフを見ることができます。

- ・ インスタンスを右クリックすると、クリックしたインスタンスの依存関係と詳細を表示したり、停止、再開などの特定のアクションを実行できます。
- ・ インスタンスをダブルクリックすると、タスクのプロパティ、実行ログ、動作ログ、コードなどが表示されます。



注：

- ・ 「2017-09-18 15:56:30.919 [job-51109647]」 は上の図のジョブ ID です。
- ・ 上の図では、同期化されたテーブルにソースのパーティション値がないためにタスクが失敗し、読み取りエラーが発生しました。
- ・ 補充データタスクのインスタンスは日ごとに依存しています。たとえば、タスクが 2017-09-15 から 2017-09-18 まで実行され、この期間中にインスタンス番号 15 が失敗した場合は、番号 16 のインスタンスも実行されません。
- ・ タスク write_result は、毎週火曜日の朝に実行されるように設定されており、本ドキュメントで前述のインスタンス生成ルールに基づいています。補充データランタイムによって選択された営業日は月曜日です（営業日 = 実行日 - 1）。インスタンスは 02 時00 分に実行されます。月曜日ではない場合、インスタンスは午前 2 時で成功状態に変換され、ログは生成されません。

定期自動実行

定期自動実行モードでは、スケジューリングシステムはすべてのタスクスケジューリング設定に従ってタスクを自動的にトリガーします。そのため、運用ポータルは提供されていません。インスタンス情報と操作ログは次のいずれかの方法で確認できます。

- ・ 「O&M センター > サイクルインスタンス」 ページで営業日や実行日などのパラメータを選択し、write_result タスクに対応するインスタンスを検索してから、インスタンスを右クリックしてインスタンス情報と実行ログを表示します。

实例名称	状态	任务类型	责任人	定时时间	业务时间	操作
write_result	未运行	数据同步	18720933100@163.com	2017-08-29 02:00:00	2017-08-28 00:00:00	终止运行 重跑 更多
write_result	未运行	数据同步	18720933100@163.com	2017-08-28 00:00:00	2017-08-27 00:00:00	终止运行 重跑 更多
write_result	未运行	数据同步	18720933100@163.com	2017-08-27 00:00:00	2017-08-26 00:00:00	终止运行 重跑 更多
write_result	未运行	数据同步	18720933100@163.com	2017-08-26 00:00:00	2017-08-25 00:00:00	终止运行 重跑 更多
write_result	未运行	数据同步	18720933100@163.com	2017-08-25 00:00:00	2017-08-24 00:00:00	终止运行 重跑 更多

- ・ インスタンスの DAG グラフは、「サイクルインスタンス」 ページで適切なタスクインスタンスを選択してクリックすると表示できます。
- インスタンスを右クリックすると、クリックしたインスタンスの依存関係と詳細を表示したり、停止、再開などの特定のアクションを実行できます。
- インスタンスをダブルクリックすると、タスクのプロパティ、実行ログ、動作ログ、コードなどを示すポップアップが表示されます。

属性	运行日志	操作日志	代码
2017-08-28 23:30:06 持续时间: -- gateway:	当前实例, 没有产生日志信息		



注:

- アップストリームタスクが実行されていないため、タスクは実行されていません。
- タスクのインスタンスの初期状態が "Not Run" の場合、スケジュールされた時間になると、対応するインスタンスのすべてのアップストリームインスタンスが正常に実行されていることをスケジューリングシステムが確認します。
- インスタンスは、そのすべてのアップストリームインスタンスが成功し、スケジュールされた時間に達した場合にのみトリガーされます。
- "Not Run" 状態のインスタンスに対しては、対応するすべてのアップストリームインスタンスが成功し、スケジュールされた時間に達したことの確認が必要です。