ALIBABA CLOUD

阿里云

大数据计算服务
快速入门

文档版本：20201021

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	**单击设置> 网络> 设置网络类型。**
粗体	表示按键、菜单、页面名称等UI元素。	**在结果确认页面，单击确定。**
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.快速体验MaxCompute

本文指导您基于MaxCompute提供的公开数据集，通过MaxCompute查询编辑器，快速体验MaxCompute产品，完成开通、执行SQL语句查询数据及下载查询结果到本地的操作。

前提条件

已创建阿里云账号或RAM用户，详情请参见创建阿里云账号或创建RAM用户。

背景信息

MaxCompute提供公开的数据集供您试用产品，详情请参见公开数据集。

MaxCompute提供的公开数据集数据只能用于产品测试，数据将不做周期更新，且不保障数据准确性，因此请您勿用于正式生产。

您可以在MaxCompute查询编辑器中执行各种SQL命令和授权命令，与在MaxCompute客户端（odpscmd）执行结果等效。您还可以切换到分析模式使用Web Excel强大而丰富的分析功能分析查询结果。

本文以通过MaxCompute查询编辑器查询并下载MAXCOMPUTE_PUBLIC_DATA.dwd_product_phoneno_basic_info_2020（2020年手机号归属地基本信息表）的数据为例，筛选出运营商为中国移动、省区为浙江省、城市为杭州市的数据，并下载到本地。

步骤一：开通MaxCompute服务

1. 登录阿里云官网。
2. 进入阿里云MaxCompute产品首页，单击**立即开通**。
3. 在购买页面，选择**地域**，并选中服务协议，单击**确认订单并支付**。

> **说明**
> - 购买页面默认提供的规格类型为MaxCompute**按量计费标准版**+DataWorks基础版。
> - MaxCompute的项目管理和查询编辑集成DataWorks的功能，因此需要同时开通DataWorks服务。DataWorks基础版为0元开通，如果您不使用数据集成、不执行调度任务，则不会产生费用。
> - 选择区域时，您需要考虑的最主要因素是MaxCompute与其他阿里云产品之间的关系，例如ECS所在区域，数据所在区域。

4. 在**支付**页面，单击**支付**，即可开通MaxCompute服务。

> **说明** 如果您选择的区域是首次开通MaxCompute和DataWorks服务，且没有存量项目，则开通成功后，系统会创建一个默认的项目，项目名称为主账号的UID。该项目默认仅阿里云主账号可见，RAM用户需要授权后才可见，授权详情请参见授权。

步骤二：查询并下载数据

1. 登录MaxCompute控制台后，在左上角选择MaxCompute服务开通的区域，单击**查询编辑**。

2. 在选择数据源对话框，选择数据源类型为MaxCompute，工作空间为已创建好的项目空间，单击确认。

   

   > 说明 工作空间默认为您最新创建的项目空间，如果您未自行创建过项目空间，默认选中开通MaxCompute服务时，系统为您创建的项目空间。

3. 在代码编辑区域，输入如下SQL语句，获取运营商、省区和城市字段名称，单击 图标。命令示例如下。

   ```
   DESC MAXCOMPUTE_PUBLIC_DATA.dwd_product_phoneno_basic_info_2020;
   ```

   返回结果如下。

   

   从执行结果可以得到运营商、省区和城市的字段名称分别为isp、province和city。

4. （可选）在代码编辑区域，输入如下SQL语句，查看运营商、省区和城市字段的值。命令示例如下。

   ```
   SELECT DISTINCT isp , province,city FROM MAXCOMPUTE_PUBLIC_DATA.dwd_product_phoneno_basic_info_2020;
   ```

   返回结果如下。

   

5. 在代码编辑区域，输入如下SQL语句，筛选出运营商为中国移动、省区为浙江省、城市为杭州市的数据。命令示例如下。

   ```
   SELECT * FROM MAXCOMPUTE_PUBLIC_DATA.dwd_product_phoneno_basic_info_2020 where isp='中国移动' and province='浙江' and city='杭州';
   ```

   返回结果如下。

   

6. 单击左上角的查询模式，切换为分析模式。

   

7. 在分析模式页面，单击右上角下载，即可下载查询结果至本地。下载的文件为Excel格式。

2.创建和查看表

快速入门为您演示一个使用MaxCompute对银行贷款购房人员进行分析的完整过程。您可以参考每个步骤的示例部分进行实际操作。

在MaxCompute中的创建表的方式有以下三种：

- 通过MaxCompute客户端实现，详情请参见表操作。
- 通过MaxCompute Studio实现，详情请参见可视化创建/修改/删除表。
- 通过DataWorks实现，详情请参见管理表。

本文将为您介绍如何使用MaxCompute客户端创建、查看表。

前提条件

- 已开通阿里云账号。
- 已开通MaxCompute服务。
- 已创建要使用的项目空间，详情请参见创建空间。如果要使用的项目空间已存在，请确保已被添加至此项目空间并被赋予建表等权限。
- 已安装并配置MaxCompute客户端。

> 说明
> - 如果您是第一次使用MaxCompute，在您快速入门之前，请务必完成所有的准备工作。
> - 快速入门系列文档着重介绍使用MaxCompute客户端配合MaxCompute Studio创建表、上传数据、加工数据及导出结果。您也可以使用DataWorks完成上述整个过程，详情参见DataWorks快速入门。

创建表

1. 登录客户端。

   运行客户端工具bin目录下的MaxCompute客户端（Linux系统下运行*./bin/odpscmd*，Windows下运行*./bin/odpscmd.bat*）登录。首先确认进入的项目空间名称是否正确。本例中项目空间名称为MaxCompute_DOC，如果不是该项目，您可以使用如下命令切换至该项目。

   ```
   use MaxCompute_DOC;
   ```

   切换成功如下图所示。

   

2. 创建表。使用如下建表语句创建表，详细介绍请参见表操作。

   ```
   CREATE TABLE [IF NOT EXISTS] table_name
   [(col_name data_type [COMMENT col_comment], ...)]
   [COMMENT table_comment]
   [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
   [LIFECYCLE days]
   [AS select_statement]
   ```

本文中，需要创建表bank_data和表result_table。bank_data用于存储业务数据，result_table用于存储数据分析后产生的结果。

- bank_data建表语句如下所示。

```
CREATE TABLE IF NOT EXISTS bank_data
(
 age           BIGINT COMMENT '年龄',
 job           STRING COMMENT '工作类型',
 marital       STRING COMMENT '婚否',
 education     STRING COMMENT '教育程度',
 default       STRING COMMENT '是否有信用卡',
 housing       STRING COMMENT '房贷',
 loan          STRING COMMENT '贷款',
 contact       STRING COMMENT '联系途径',
 month         STRING COMMENT '月份',
 day_of_week   STRING COMMENT '星期几',
 duration      STRING COMMENT '持续时间',
 campaign      BIGINT COMMENT '本次活动联系的次数',
 pdays         DOUBLE COMMENT '与上一次联系的时间间隔',
 previous      DOUBLE COMMENT '之前与客户联系的次数',
 poutcome      STRING COMMENT '之前市场活动的结果',
 emp_var_rate  DOUBLE COMMENT '就业变化速率',
 cons_price_idx DOUBLE COMMENT '消费者物价指数',
 cons_conf_idx DOUBLE COMMENT '消费者信心指数',
 euribor3m     DOUBLE COMMENT '欧元存款利率',
 nr_employed   DOUBLE COMMENT '职工人数',
 y             BIGINT COMMENT '是否有定期存款'
);
```

直接运行上述建表语句即可，成功后您会看到OK字样。

> 说明 如果客户端执行报错，建议您手动输入SQL语句执行，或者使用DataWorks临时查询功能运行SQL语句，详细请参考使用临时查询运行SQL语句（可选）。

- result_table建表语句如下所示。

```
CREATE TABLE IF NOT EXISTS result_table
(
 education  STRING COMMENT '教育程度',
 num        BIGINT COMMENT '人数'
);
```

查看表

当创建表成功之后，您可以通过如下命令查看表的信息。

```
desc <table_name>;
```

其中，table_name是查看表的名字。例如，您可执行命令 `desc bank_data;` 查看上述示例中bank_data表的信息。结果显示如下图所示。

查看表信息的更多信息请参见表操作。

其他表操作

- 删除表

  删除表的命令如下所示。

  ```
  DROP TABLE [IF EXISTS] table_name;
  ```

- 创建分区

  本文上述示例中使用的是非分区表。如果您需要使用分区表，可以使用如下语句创建分区。

  ```
  alter table table_name add [if not exists] partition(partition_col1 = partition_col_value1, partition_col2 = partiton_col_value2, ...);
  ```

  > 说明
  > - 如果您使用Tunnel命令导入不同分区数据，首先需要创建分区。
  > - 如果您使用数据集成、INSERT语句等方法导入分区数据则无需单独创建分区。

- 删除分区

  删除分区的命令如下所示。

  ```
  alter table table_name drop [if exists] partition(partition_col1 = partition_col_value1, partition_col2 = partiton_col_value2, ...);
  ```

  例如删除区域为hangzhou，日期为20180923的分区，语句如下所示。

  ```
  alter table user drop if exists partition(region='hangzhou',dt='20180923');
  ```

关于表操作的更多信息请参见表操作。

后续步骤

在您完成表的创建后，即可进行导入数据到MaxCompute，以便后续对数据进行进一步处理。

3.导入数据

本文为您介绍如何使用Tunnel命令导入数据到MaxCompute。

导入数据 Tunnel命令
MaxCompute提供多种数据导入导出方式，本文主要介绍在客户端上使用Tunnel命令操作进行数据导入。

Tunnel命令导入数据

1. 准备数据 。

   将测试数据下载至本地备用，假设存放路径为*D:*\。本文中使用的测试数据为banking.txt，主要用于记录各人员的年龄、工作、房贷等信息，选取其中前三条数据展示如下。

   ```
   44,blue-collar,married,basic.4y,unknown,yes,no,cellular,aug,thu,210,1,999,0,nonexistent,1.4,93.444,
   -36.1,4.963,5228.1,0
   53,technician,married,unknown,no,no,no,cellular,nov,fri,138,1,999,0,nonexistent,-0.1,93.2,-42,4.021,
   5195.8,0
   28,management,single,university.degree,no,yes,no,cellular,jun,thu,339,3,6,2,success,-1.7,94.055,-39
   .8,0.729,4991.6,1
   ```

2. （可选）创建MaxCompute表。

   如果您已完成创建和查看表示例中bank_data表的创建，请跳过本步骤，否则请参照示例创建表bank_data。

3. 执行Tunnel命令。

   登录MaxCompute客户端执行如下命令进行数据导入。

   ```
   tunnel upload D:\banking.txt bank_data;
   ```

   其中，`D:\banking.txt` 是需要上传文件的本地路径。`bank_data` 为将要导入的表名称。

   当出现下图中OK字样，说明上传成功。

4. 结果验证。

   执行成功后，您可以使用如下语句查看表bank_data的数据条数，验证是否完成所有数据上传，示例数据中共有41188条数据。

   ```
   SELECT COUNT(*) FROM bank_data;
   ```

   > **说明**
   > - 有关Tunnel命令的更多详细介绍，例如如何将数据导入分区表等，请参见Tunnel操作。
   > - 使用Tunnel上传数据如果出现问题，请参见数据上传下载。

其他导入方式

除了通过客户端导入数据，您也可以使用MaxCompute Studio、Tunnel SDK、数据集成、开源的Sqoop、Fluentd、Flume、LogStash 等工具将数据导入到MaxCompute，详情请参见数据上传下载-工具介绍。

后续步骤

当数据导入到MaxCompute后，可以在MaxCompute上运行SQL来处理数据。

4.运行SQL语句和导出数据

本文介绍如何在客户端上运行常见SQL语句和导出数据。

运行SQL语句 导出数据 Tunnel
MaxCompute支持以下方式运行SQL语句：

- 客户端
- DataWorks

背景信息

MaxCompute目前支持的SQL语法如下：

- 各类运算符。
- 通过DDL语句对表、分区以及视图进行管理。
- 通过SELECT语句查询表中的记录，通过WHERE语句过滤表中的记录。
- 通过INSERT语句插入数据、更新数据。
- 通过等值连接JOIN操作，支持两张表的关联，并支持多张小表的MapJOIN。
- 通过内置函数和自定义函数来进行计算。
- 正则表达式。

> 说明
>
> - MaxCompute SQL不支持事务、索引、UPDATE以及DELETE等操作，同时MaxCompute的SQL语法与Oracle、MySQL有一定差别，您无法将其他数据库中的SQL语句无缝迁移到MaxCompute上来，更多差异请参见与其他SQL语法的差异。
> - MaxCompute上作业提交后会有几十秒到数分钟不等的排队调度，所以MaxCompute适合一次批量处理海量数据的跑批作业，不适合直接对接需要每秒处理几千至数万笔事务的前台业务系统。作业的优化请参见SQL调优。
> - 关于SQL操作的详细示例，请参见SQL及函数。
> - MaxCompute SQL的更多限制请参见SQL使用限制项。

提取和分析数据

查询不同学历的单身人士贷款买房的数量，并将结果保存到result_table中。

1. 使用如下语句将表bank_data中不同学历单身贷款买房人士的数量保存至表result_table中。

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

2. 使用如下语句查看result_table表中的数据。

```
SELECT * FROM result_table;
```

结果如下所示。

上述过程仅仅是一个最简单的数据加工举例，您在实际应用的过程中，可能需要使用多个SQL对多个表进行加工操作。推荐您使用DataWorks完成复杂的数据加工业务流程。

导出数据

使用如下语句将表result_table中数据导出到本地D盘保存成名为*result.txt*的文件。

```
tunnel download result_table D:\result.txt;
```

其中，`result_table`为需要导出的表，*D:\result.txt*为导出后保存的路径及名称。更多Tunnel命令，请参考Tunnel命令参考。

导出成功后如下图所示，可以看到**download OK**字样。

> **说明** 如果需要将数据导出到MySQL或其他数据源，推荐您使用数据集成，详细请参见概述。

5.编写MapReduce（可选）

本文将为您介绍安装好MaxCompute客户端后，如何快速编写和运行MapReduce WordCount示例程序。

前提条件

- 编写、编译、运行MapReduce前，需要首先安装JDK 1.6或以上版本。

  > 说明 如果您使用Maven，可以从Maven库中搜索odps-sdk-mapred获取不同版本的Java SDK，相关配置信息如下。

  ```
  <dependency>
    <groupId>com.aliyun.odps</groupId>
    <artifactId>odps-sdk-mapred</artifactId>
    <version>0.26.2-public</version>
  </dependency>
  ```

- 请参见安装并配置客户端对MaxCompute客户端进行部署。更多关于MaxCompute客户端的使用，请参见MaxCompute客户端。
- 确保您购买的资源非按量计费开发者版。按量计费开发者版资源，仅支持MaxCompute SQL（支持使用UDF）、PyODPS作业，暂不支持MapReduce、Spark等其它作业。

操作步骤

1. 安装并配置好客户端后，运行bin目录下的MaxCompute客户端（Linux系统下运行*./bin/odpscmd*，Windows下运行*./bin/odpscmd.bat*），进入相应项目空间中。
2. 输入建表语句，创建输入和输出表，如下所示。

   ```
   --创建输入表wc_in。
   CREATE TABLE wc_in (key STRING, value STRING);
   --创建输出表wc_out。
   CREATE TABLE wc_out (key STRING, cnt BIGINT);
   ```

   更多创建表的语句请参见创建表。
3. 在表wc_in中插入数据。您可以通过以下两种方式插入数据：
   - 使用Tunnel命令上传数据。

     需要插入的数据如下所示，请在本地创建kv.txt文件保存数据，假设kv.txt文件本地存放路径为D:\。

     ```
     238,val_238
     186,val_86
     186,val_86
     ```

     执行如下命令，上传数据。

     ```
     Tunnel upload D:\kv.txt wc_in;
     ```

- 执行如下SQL语句直接插入数据。

```
INSERT INTO TABLE wc_in VALUES ('238',' val_238'),('186','val_86'),('186','val_86');
```

4. 开发MapReduce程序并上传MaxCompute。

   在Eclipse或MaxCompute Studio中创建一个项目工程，并在此工程中编写MapReduce程序。本地调试通过后，将编译好的程序（JAR包，例如*Word-count-1.0.jar*）导出并上传至MaxCompute。

   > **说明** 本文中，您直接使用WordCount示例中的示例代码生成*Word-count-1.0.jar*包即可，无需自己开发。

5. 在MaxCompute客户端，将JAR包添加为项目空间资源（例如，此处的JAR包名为word-count-1.0.jar）。

```
ADD JAR word-count-1.0.jar;
```

6. 在MaxCompute客户端运行Jar命令。

```
Jar -resources word-count-1.0.jar -classpath /home/resources/word-count-1.0.jar com.taobao.jingfan.WordCount wc_in wc_out;
```

   > **说明** 如果您在Java程序中使用了任何资源，请务必将此资源加入-resources参数。Jar命令的详细介绍请参见作业提交。

7. 在MaxCompute客户端查看结果。

```
SELECT * FROM wc_out;
```

6.开发Java UDF（可选）

本文为您介绍如何开发Java UDF，分别提供UDF、UDAF、UDTF的代码示例，并提供了通过两种方法开发UDF的完整流程。

背景信息

MaxCompute的UDF包括UDF、UDAF和UDTF，这三种函数被统称为UDF，更多信息请参见概述。

Java UDF的开发可以通过MaxCompute Studio实现，详情请参见 开发UDF。

> 说明
> - 自定义函数注册、注销和查看函数列表的相关命令请参见函数操作。
> - Java和MaxCompute的数据类型对应关系，请参见参数与返回值类型。
> - 如果您使用Maven实现Java UDF，可以从Maven库中搜索odps-sdk-udf获取不同版本的Java SDK。例如，使用以下配置添加指定版本的Java SDK依赖。
>
> ```
> <dependency>
>   <groupId>com.aliyun.odps</groupId>
>   <artifactId>odps-sdk-udf</artifactId>
>   <version>0.20.7</version>
> </dependency>
> ```

UDF示例

通过MaxCompute Studio开发字符小写转换功能的UDF步骤如下：

1. 准备工具环境并创建Java Module。

   您需要完成准备工作，包括安装Studio并在Studio上创建MaxCompute项目链接以及创建MaxCompute Java Module。

2. 编写代码。

   i. 在Project区域，右键单击Module的源码目录（即src > main > java），选择new > MaxCompute Java。

   

   ii. 在Create new MaxCompute java class对话框，配置Name和Kind，单击OK。

   - Name：创建的MaxCompute Java Class名称。如果还没有创建Package，在此处填写packagename.classname，会自动生成Package。
   - Kind：选择类型为UDF。支持的类型包含自定义函数（UDF、UDAF和UDTF）、MapReduce（Driver、Mapper和Reducer）和非结构化开发（StorageHandler、Extractor和Outputer）等。

iii. 创建成功后，编辑代码如下。

```
package <package名称>;
import com.aliyun.odps.udf.UDF;
public final class Lower extends UDF {
  public String evaluate(String s) {
    if (s == null) {
      return null;
    }
      return s.toLowerCase();
  }
}
```

> 说明 如果需要本地调试Java UDF，请参见开发和调试UDF。

3. 注册MaxCompute UDF。

   右键单击UDF的Java文件，选择Deploy to server...。在Package a jar and submit resource对话框中配置参数。配置完成，单击OK。

   - MaxCompute project：UDF所在的MaxCompute Project名称。
   - Resource file：选择JAR包路径。
   - Resource name：输入注册的资源名称。
   - Function name：输入注册的函数名称。

4. 试用UDF。

   打开SQL脚本，执行测试代码。例如 `SELECT Lower_test('ABC');` 。

> 说明 在MaxCompute Studio中编写SQL脚本请参见编写SQL脚本。

UDAF示例代码

UDAF的注册方式与UDF基本相同，使用方式与内建函数中的聚合函数相同。计算平均值的UDAF的代码示例如下。

```
package org.alidata.odps.udf.examples;
import com.aliyun.odps.io.LongWritable;
import com.aliyun.odps.io.Text;
import com.aliyun.odps.io.Writable;
import com.aliyun.odps.udf.Aggregator;
import com.aliyun.odps.udf.UDFException;
/**
 * project: example_project
 * table: wc_in2
 * partitions: p2=1,p1=2
 * columns: colc,colb,cola
 */
public class UDAFExample extends Aggregator {
 @Override
 public void iterate(Writable arg0, Writable[] arg1) throws UDFException {
  LongWritable result = (LongWritable) arg0;
  for (Writable item : arg1) {
   Text txt = (Text) item;
   result.set(result.get() + txt.getLength());
  }
 }
 @Override
 public void merge(Writable arg0, Writable arg1) throws UDFException {
  LongWritable result = (LongWritable) arg0;
  LongWritable partial = (LongWritable) arg1;
  result.set(result.get() + partial.get());
 }
 @Override
 public Writable newBuffer() {
  return new LongWritable(0L);
 }
 @Override
 public Writable terminate(Writable arg0) throws UDFException {
  return arg0;
 }
}
```

UDTF示例代码

UDTF的注册和使用方式与UDF相同，代码示例如下。

```
package org.alidata.odps.udtf.examples;
import com.aliyun.odps.udf.UDTF;
import com.aliyun.odps.udf.UDTFCollector;
import com.aliyun.odps.udf.annotation.Resolve;
import com.aliyun.odps.udf.UDFException;
// TODO define input and output types, e.g., "string,string->string,bigint".
@Resolve({"string,bigint->string,bigint"})
public class MyUDTF extends UDTF {
 @Override
 public void process(Object[] args) throws UDFException {
   String a = (String) args[0];
   Long b = (Long) args[1];
   for (String t: a.split("\\s+")) {
     forward(t, b);
   }
 }
}
```

MaxCompute提供多种内建函数来满足您的计算需求，同时您还可以使用DataWorks创建注册MaxCompute函数来满足不同的计算需求。UDF更多示例请参考更多UDF示例。

7.编写Graph（可选）

本文将以单源最短距离（Single Source Shortest Path，SSSP）算法为例，为您介绍如何提交Graph作业。

Graph SSSP
Graph作业的提交方式与MapReduce的提交方式基本相同。

前提条件

- 编写、编译、运行MapReduce前，需要首先安装JDK 1.6或以上版本。

  说明 如果您使用Maven，可以从Maven 库中搜索odps-sdk-mapred获取不同版本的Java SDK，相关配置信息如下所示。

  ```
  <dependency>
    <groupId>com.aliyun.odps</groupId>
    <artifactId>odps-sdk-mapred</artifactId>
    <version>0.26.2-public</version>
  </dependency>
  ```

- 请参见安装并配置客户端对MaxCompute客户端进行部署。更多关于MaxCompute客户端的使用，请参见MaxCompute客户端。

操作步骤

1. 运行MaxCompute客户端。
2. 执行如下命令创建输入表sssp_in和输出表sssp_out 。

   ```
   CREATE TABLE sssp_in (v bigint, es string);
   CREATE TABLE sssp_out (v bigint, l bigint);
   ```

   创建表的更多语句请参见表操作。
3. 上传数据至表sssp_in中。

   示例数据如下，建议您创建sssp.txt文件将数据保存至本地。假设保存在本地路径为*D:*

   ```
   1 2:2,3:1,4:4
   2 1:2,3:2,4:1
   3 1:1,2:2,5:1
   4 1:4,2:1,5:1
   5 3:1,4:1
   ```

   执行Tunnel命令上传数据至表sssp_in中， 以空格键做两列的分隔符。

   ```
   tunnel u -fd " " D:\sssp.txt sssp_in;
   ```
4. 编写SSSP示例。

本地编译、调试SSSP算法示例，假设代码被打包为名为*odps-graph-example-sssp.jar*的文件。

> 说明 仅需要将SSSP代码打包即可，不需要同时将SDK打包入*odps-graph-example-sssp.jar*中。

5. 添加Jar资源。 更多添加资源的信息，请参见资源操作。

```
add jar $LOCAL_JAR_PATH/odps-graph-example-sssp.jar;
```

6. 运行SSSP。

```
jar -libjars odps-graph-example-sssp.jar -classpath $LOCAL_JAR_PATH/odps-graph-example-sssp.jar com.aliyun.odps.graph.example.SSSP 1 sssp_in sssp_out;
```

Graph作业执行时命令行会打印作业实例ID、执行进度、结果Summary等，输出示例如下所示。

```
ID = 20130730160742915g******
2013-07-31 00:18:36    SUCCESS
Summary:
Graph Input/Output
Total input bytes=211
Total input records=5
Total output bytes=161
Total output records=5
graph_input_[bsp.sssp_in]_bytes=211
graph_input_[bsp.sssp_in]_records=5
graph_output_[bsp.sssp_out]_bytes=161
graph_output_[bsp.sssp_out]_records=5
Graph Statistics
Total edges=14
Total halted vertices=5
Total sent messages=28
Total supersteps=4
Total vertices=5
Total workers=1
Graph Timers
Average superstep time (milliseconds)=7
Load time (milliseconds)=8
Max superstep time (milliseconds) =14
Max time superstep=0
Min superstep time (milliseconds)=5
Min time superstep=2
Setup time (milliseconds)=277
Shutdown time (milliseconds)=20
Total superstep time (milliseconds)=30
Total time (milliseconds)=344
OK
```

> **说明** 如果您需要使用Graph功能，直接提交图计算作业即可。

8.使用临时查询运行SQL语句（可选）

如果您已经创建了MaxCompute项目（DataWorks工作空间），可以直接使用DataWorks临时查询功能，快速运行SQL语句操作MaxCompute。

操作步骤

进入临时查询

1. 登录DataWorks控制台。
2. 在左侧导航栏，单击**工作空间列表**。
3. 选择工作空间所在地域后，单击相应工作空间后的**进入数据开发**。
4. 创建ODPS SQL节点。
   i. 在左侧菜单栏上，单击**临时查询**。
   ii. 右键单击**临时查询**，选择**新建节点 > ODPS SQL**。

5. 填写**新建节点**对话框中的**节点名称**，单击**提交**完成节点的创建。

运行SQL

6. 在创建的临时查询节点中输入SQL语句，单击  按钮。

```
-- 创建一张分区表sale_detail
CREATE TABLE if not exists sale_detail
(
shop_name    string,
customer_id   string,
total_price   double
)
partitioned by (sale_date string,region string);
```

> **说明** 临时查询支持的SQL语句，请参见MaxCompute支持的SQL语句。

7. 您可以查看本次运行的费用预估，决定是否继续进行本次操作。单击**运行**，继续SQL语句运行。

8. 在下方的日志窗口，查看运行情况和最终结果。如果本次运行成功，结果为**OK**。