阿里云
MaxCompute

快速入门

文档版本：20190724

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
	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。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 确定。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
*## *	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。	`swich` *{stand | slave}*

目录

1 创建和查看表

本文为您演示一个完整的使用MaxCompute进行银行贷款购房人员分析的过程。您可以参考每个步骤的示例部分进行实际操作。

前提条件：您需要首先完成阿里云账号开通、MaxCompute项目购买，被添加到项目空间并被赋予建表等权限，在本地完成客户端的配置后即可操作MaxCompute。

> 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。
>
> 快速入门系列文档后续将着重介绍使用客户端配合MaxCompute Studio完成表的创建、数据的上传、加工及导出。您也可以使用DataWorks完成上述整个过程，详情参见DataWorks快速入门。

由于在MaxCompute中的操作对象（输入、输出）都是表，所以在处理数据之前，首先要创建表、分区。创建、查看或删除表的方式有以下几种，本文将为您介绍如何使用客户端创建、查看表：

- 通过客户端常用命令实现。
- 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，详情请参见创建表和删除表。

创建表

登录客户端之后，使用如下建表语句创建表。

```
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]
```

> 说明:
> 创建表的详细介绍请参见表操作。

登录MaxCompute客户端后，首先您需要确认当前是否在正确的项目中。本例中项目名称为MaxCompute_DOC，您可以使用`use MaxCompute_DOC;`命令切换到该项目（项目需要您提前创建）。

```
odps@ MaxCompute_DOC>
```

本文中，表bank_data用于存储业务数据，表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。

- 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`desc <table_name>;`命令查看表的信息。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。

```
odps@ MaxCompute_DOC>desc bank_data;

+------------------------------------------------------------------------------------
| Owner: RAM$dtplus_docs:[redacted] | Project: maxcompute_doc
| TableComment:
+------------------------------------------------------------------------------------
| CreateTime:                   2018-12-18 17:01:50
| LastDDLTime:                  2018-12-18 17:01:50
| LastModifiedTime:             2018-12-18 17:01:50
+------------------------------------------------------------------------------------
| InternalTable: YES      | Size: 0
+------------------------------------------------------------------------------------
| Native Columns:
+------------------------------------------------------------------------------------
| Field           | Type       | Label | Comment
+------------------------------------------------------------------------------------
| age             | bigint     |       | 年龄
| job             | string     |       | 工作类型
| marital         | string     |       | 婚否
| education       | string     |       | 教育程度
| default         | string     |       | 是否有信用卡
| housing         | string     |       | 房贷
| loan            | string     |       | 贷款
| contact         | string     |       | 联系途径
```

查看表信息的更多详情请参见表操作。

其他表操作

其它可选的表操作如下所示，关于表操作的更多详情请参见表操作：

- 创建分区

  本文上述示例中使用的是非分区表。

  如果您创建的是分区表，为了在分区表中使用Tunnel命令导入不同分区数据，您首先需要创建分区。命令如下。

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

  其它操作例如使用数据集成、insert等无需单独创建分区。

- 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');
  ```

- 删除表

  删除表的命令如下所示。

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

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。

2 导入数据

在您完成表格的创建后，就可以使用Tunnel命令导入数据到MaxCompute了。

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

   本文中使用的测试数据为bank.txt，主要用于记录各人员的年龄、工作、房贷等信息。请点击下载数据到您的电脑本地。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
   ```

   本文中，bank.txt本地存放路径为*D:*。

2. 创建MaxCompute表

   您需要把上面的数据导入到MaxCompute的一张表中，所以需要创建MaxCompute表，如果您已完成步骤一创建bank_data表，可跳过本步骤。

3. 执行Tunnel命令

   输入表创建成功后，可以在MaxCompute客户端输入Tunnel命令进行数据的导入，如下所示。

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

   当出现下图中OK字样，说明上传成功。

   ```
   odps@ MaxCompute_DOC>tunnel upload D:\banking.txt bank_data;
   Upload session: 20190110214[redacted]
   Start upload:D:\banking.txt
   Using \n to split records
   Upload in strict schema mode: true
   Total bytes:4841548      Split input to 1 blocks
   2019-01-10 21:49:40      scan block: '1'
   2019-01-10 21:49:40      scan block complete, blockid=1
   2019-01-10 21:49:40      upload block: '1'
   2019-01-10 21:49:43      1:0:4841548:D:\banking.txt      100%    4.6 MB  2.3 MB/s
   
   2019-01-10 21:49:43      upload block complete, blockid=1
   upload complete, average speed is 1.5 MB/s
   OK
   ```

4. 结果验证

执行成功后，您可以使用`select count(*) from bank_data;`查看表bank_data的记录数，验证是否完成所有数据上传，本文中共有41188条数据。

```
+------------+
| _c0        |
+------------+
| 41188      |
+------------+
1 records (at most 10000 supported) fetched by instance tunnel.
```

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，请参考Tunnel命令相关问题

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 等工具都可以进行数据导入到MaxCompute，详情请参见数据上传下载-工具介绍。

后续步骤

在您的数据导入到MaxCompute后，即可在MaxCompute上运行SQL来处理数据。

3 运行SQL和导出数据

在您的数据导入到MaxCompute后，即可在MaxCompute上运行SQL来处理数据。

您可以选择在MaxCompute客户端或DataWorks上运行SQL语句，本文为您演示如何使用客户端完成SQL查询、写入等过程。

MaxCompute当前支持的SQL语法：

- 支持各类运算符。
- 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。
- 支持通过内置函数和自定义函数来进行计算。
- 支持正则表达式。

 说明：

- MaxCompute SQL不支持事务、索引及Update/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。
- 关于SQL操作的详细示例，请参见SQL模块。
- 在使用方式上，MaxCompute作业提交后会有几十秒到数分钟不等的排队调度，所以适合处理跑批作业，一次作业批量处理海量数据，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优化示例。
- MaxCompute SQL的更多限制请参见SQL限制项。

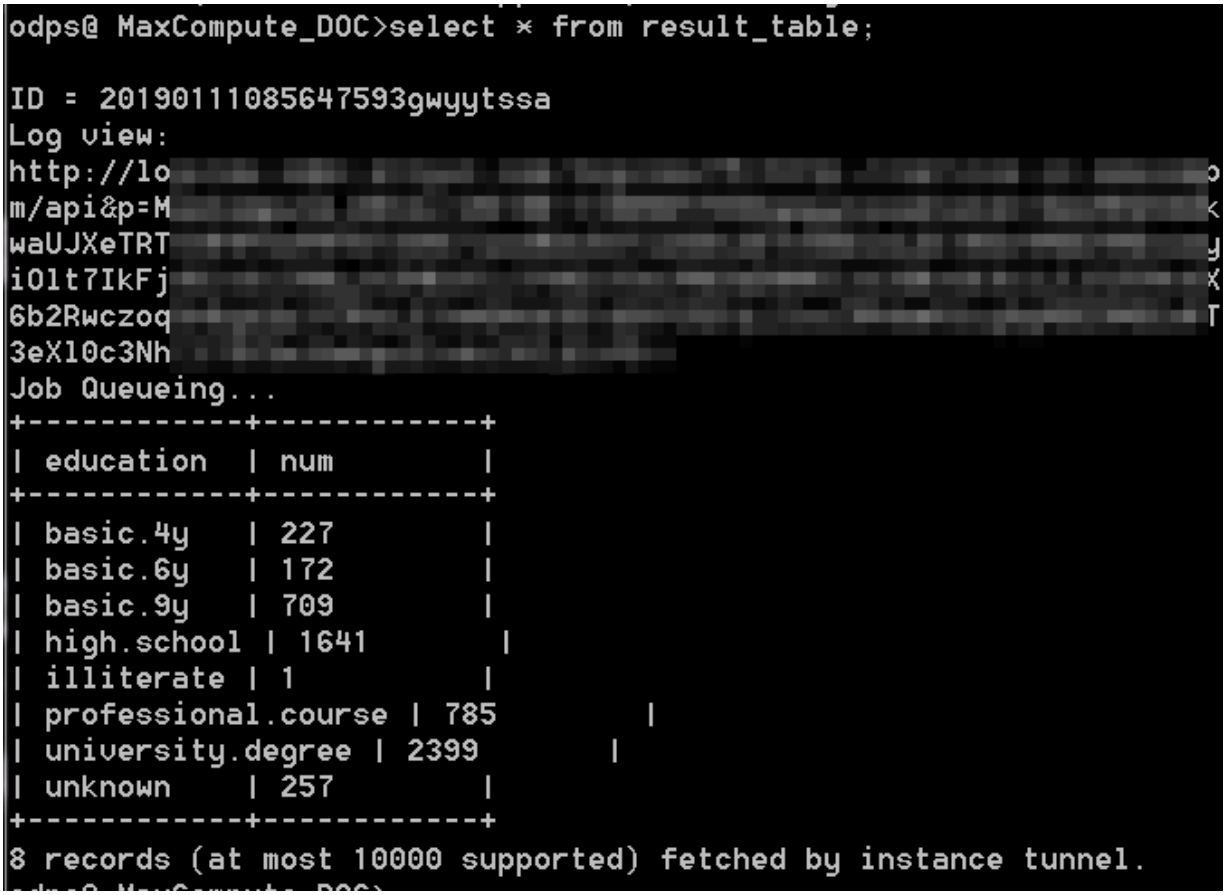
提取和分析数据

用SQL代码查询不同学历的单身人士贷款买房的数量，并将结果保存下到result_table以便分析或展现。

```
INSERT OVERWRITE TABLE result_table  --数据插入到result_table中。
SELECT education,COUNT(marital) AS num
FROM bank_data
WHERE housing = 'yes'
    AND marital = 'single'
```

```
GROUP BY education;
```

您可以使用`select * from result_table;`查看result_table表中的数据，如下图所示，可以看到当前受到各阶段教育的单身人士数量。



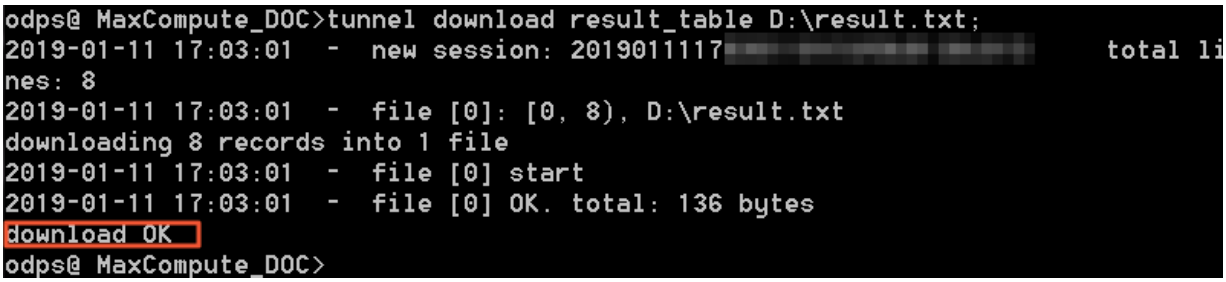
上述过程仅仅是一个最简单的数据加工举例，您在实际应用的过程中，可能需要使用多个SQL对多个表进行加工操作。推荐您使用DataWorks完成复杂的数据加工业务流程。

导出数据

您在完成SQL语句处理后，可以参考Tunnel上传下载命令将处理完的数据导出到本地。本例中，将result_table中数据导出到本地D盘。

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

导出成功后如下图所示，可以看到download OK字样。



说明:

如果您需要将数据导出到MySQL或其他数据源，推荐您使用数据集成。

4 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。

> 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>
```

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。
- 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. 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。

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

   更多创建表的语句请参见创建表。
3. 上传数据。

   您可以通过以下两种方式上传数据：

   - 使用Tunnel命令上传数据。

     ```
     tunnel upload kv.txt wc_in
     -- 上传示例数据
     ```

     kv.txt文件中的数据如下所示，您可以在本地创建kv.txt文件后再上传。

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

```
186,val_86
```

- 您也可以通过SQL语句直接插入数据，示例如下。

```
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。

   您需要先在Eclipse或MaxCompute Studio中创建一个项目工程，而后在此工程中编写MapReduce程序。本地调试通过后，将编译好的程序（Jar 包，如Word-count-1.0.jar）导出并上传至MaxCompute。详情请参见MapReduce开发插件介绍或Eclipse开发插件。

   > 说明:
   > 本文中，您直接使用Word-count-1.0.jar即可，无需自己开发。

5. 在MaxCompute客户端，添加Jar包到project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;
```

7. 在MaxCompute客户端查看结果。

```
select * from 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。

5 开发Java UDF（可选）

MaxCompute的UDF包括UDF、UDAF和UDTF三种函数。通常情况下，这三种函数被统称为UDF。

> 说明:
> 当前MaxCompute已支持Java UDF、Python UDF、UDJ、UDT，详细信息可参见Java UDF。

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>
```

通常，Java UDF的开发可以通过以下几种方式实现：

- 使用MaxCompute Studio完成Java UDF开发整个流程。
- 使用Eclipse插件开发和调试Java UDF，导出Jar包，然后通过命令或者DataWorks添加资源后再注册函数。

本文中分别提供UDF、UDAF、UDTF的代码示例，并通过提供两种开发UDF完整流程的步骤示例。UDAF、UDTF与UDF的操作步骤一致。

> 说明:
> - 关于自定义函数注册、注销、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。

UDF示例

下面将为您介绍使用MaxCompute Studio或Eclipse开发字符小写转换功能的UDF实现示例。

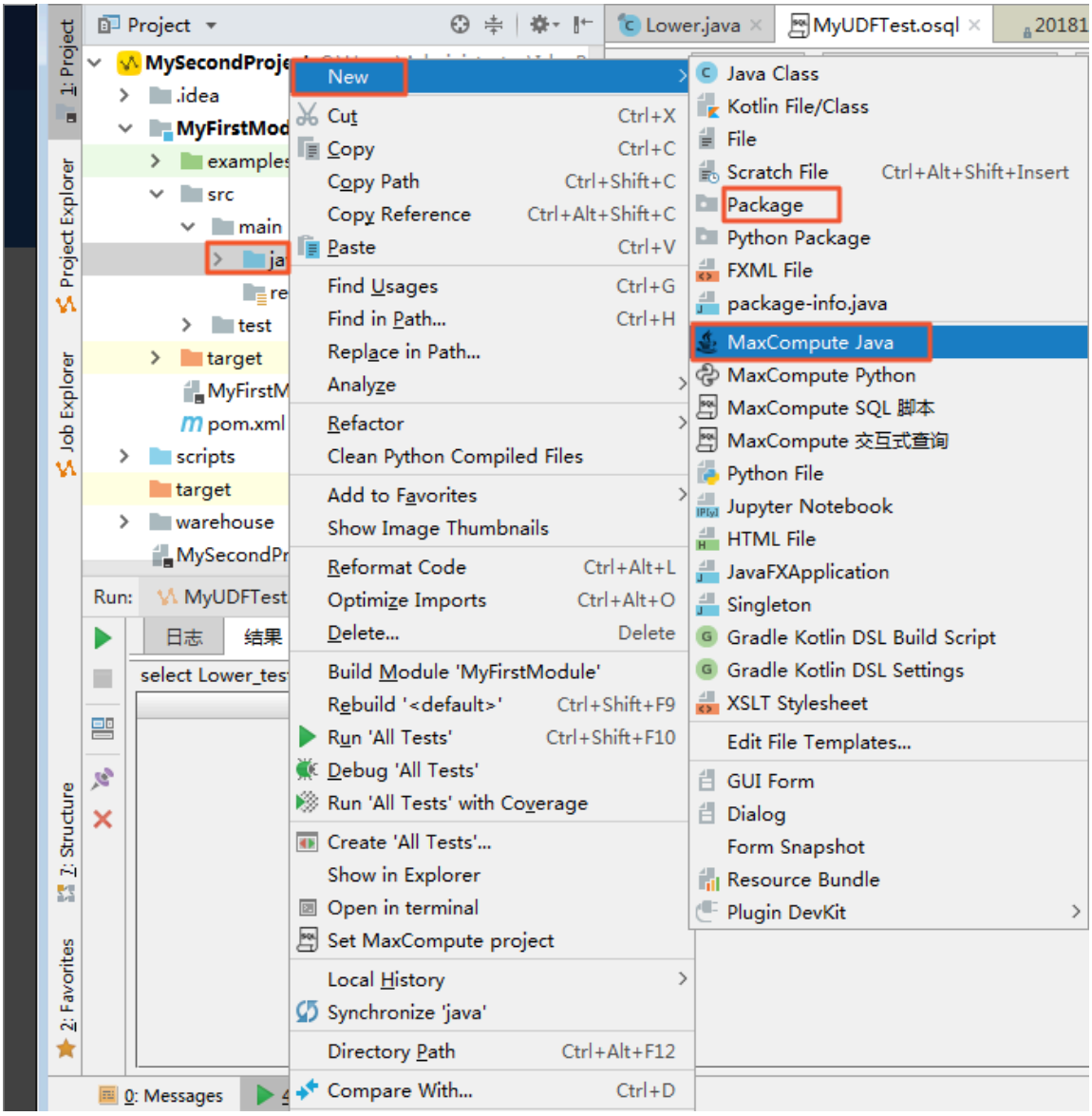
- 使用MaxCompute Studio开发

1. 准备工具环境并创建Java Module

   假设已经完成环境准备，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

   在配置好的Java Module下创建Java文件。



直接选择MaxCompute Java，然后在name一栏里输入`package名称.文件名`，Kind选择UDF。之后编辑如下代码：

```
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;
```

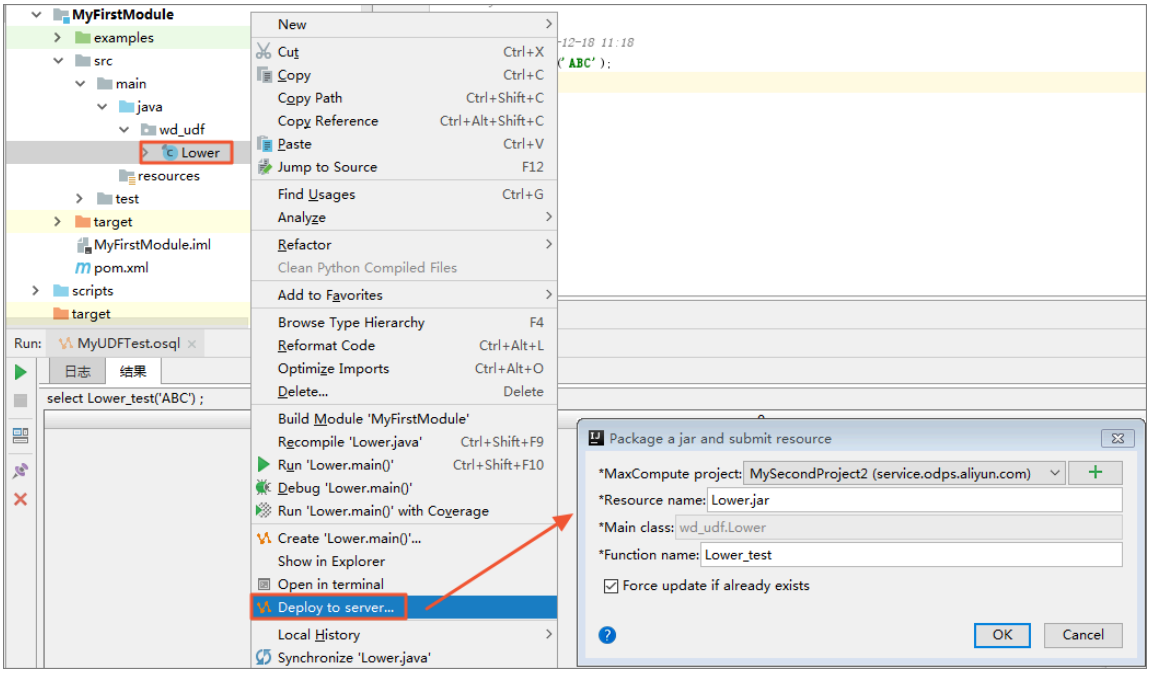
```
        }
            return s.toLowerCase();
    }
}
```

说明:

若需本地调试Java UDF，请参见开发和调试UDF。

3. 注册MaxCompute UDF

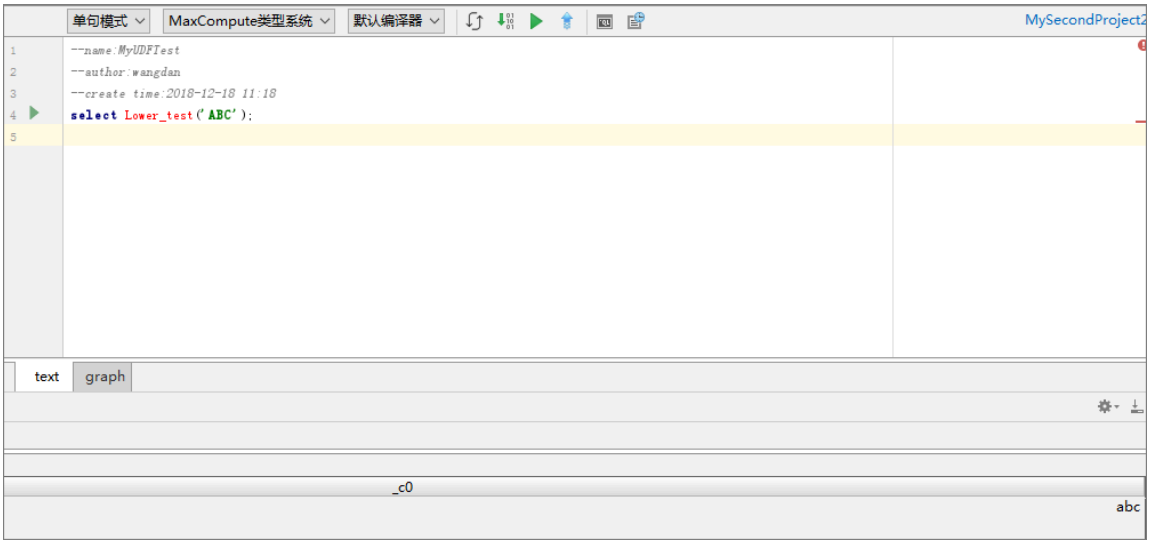
如下图所示，右键单击UDF的Java文件，选择Deploy to server，弹框里选择注册到哪个MaxCompute project，输入function name，Resource name也可以修改。



填写好后，单击OK即可。注册成功后会有提示。

4. 试用UDF

打开SQL脚本，执行代码如select Lower_test('ABC');。



说明:

Studio中编写SQL脚本请参见编写SQL脚本。

- 使用Eclipse插件开发

1. 创建工程

   请首先在Eclipse插件创建一个MaxCompute（原名ODPS）工程，详情请参见创建MaxCompute工程。

2. 编写代码

   按照MaxCompute UDF框架的规定，实现函数功能，并进行编译。

```
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();
    }
}
```

   将这个Jar包命名为my_lower.jar。

    说明:

   - 更详细的开发调试代码的介绍请参见UDF开发插件介绍。
   - SDK的使用信息请参见UDF SDK 。

3. 添加资源

   在运行UDF函数前，必须在MaxCompute中指定引用的UDF代码。您需要将编写的Java UDF编译成Jar包，并以Jar资源的形式添加至MaxCompute中。MaxCompute的UDF框架将自动加载所添加的Jar包，从而运行您自定义的UDF函数。

    说明:

   MaxCompute MapReduce也用到了资源这一特有概念，MapReduce文档中对资源的使用也有阐述。

   例如，执行以下命令在MaxCompute中添加上一步骤中编译完成的UDF函数Jar包。

```
add jar my_lower.jar;
-- 如果存在同名的资源请将这个jar包重命名
-- 并注意修改下面示例命令中相关jar包的名字
```

```
-- 又或者直接使用-f选项覆盖原有的jar资源
```

4. 注册UDF函数

在MaxCompute中添加Jar资源后，MaxCompute可以自动获取该UDF函数的代码。但在MaxCompute中使用该UDF函数前，您还需要注册该UDF函数，指定该函数名与jar资源的对应关系。

命令格式

```
CREATE FUNCTION <function_name> AS <package_to_class> USING <
resource_list>;
```

参数说明：

- function_name：UDF函数名，这个名字就是SQL中引用该函数所使用的名字。
- package_to_class：如果是Java UDF，这个名字就是从顶层包名一直到实现UDF类名的fully qualified class name。如果是python UDF，这个名字就是python脚本名.类名。并且这个名字必须使用引号。
- resource_list：UDF所用到的资源列表。
  - 此资源列表必须包括UDF代码所在的资源。
  - 如果您的代码中通过distributed cache接口读取资源文件，此列表中还要包括UDF所读取的资源文件列表。
  - 资源列表由多个资源名组成，资源名之间由逗号分隔，且资源列表必须用引号引起来。
  - 如果需要指定资源所在的project，写法为`<project_name>/resources/<resource_name>`。

例如，执行以下命令注册test_lower函数并将其与my_lower.jar资源关联。

```
CREATE FUNCTION test_lower AS 'org.alidata.odps.udf.examples.Lower
' USING 'my_lower.jar';
```

说明：

- 与资源文件一样，同名函数只能注册一次。

- 通常，您的自建函数无法覆盖系统内建函数。只有项目空间的Owner才有权利覆盖内建函数。如果您使用了覆盖内建函数的自定义函数，在SQL执行结束后，会在Summary中打印出Warning信息。

5. 在SQL中使用此函数进行验证。

```
select test_lower('A') from my_test_table;
```

UDAF示例

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的代码示例如下所示。

```
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

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创建自定义函数来满足不同的计算需求。您也可以参考更多UDF示例。

6 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。

> 说明：
> 算法详情请参见SSSP算法。

Graph作业的提交方式与MapReduce的提交方式基本相同。如果您使用Maven，可以从Maven库中搜索`odps-sdk-graph`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-graph</artifactId>
    <version>0.20.7</version>
</dependency>
```

操作步骤

1. 运行MaxCompute客户端。
2. 创建输入表和输出表 。

   ```
   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。

   本地数据的内容如下，建议您创建名为`sssp.txt`的文件后粘贴下列数据。

   ```
   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
   ```

   以空格键做两列的分隔符，执行Tunnel命令上传数据。

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

   根据Graph开发插件的介绍，本地编译、调试SSSP算法示例。本示例中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。

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

说明:

添加资源的介绍请参见资源操作。

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;
```

Jar命令用于运行MaxCompute Graph作业，用法与MapReduce作业的运行命令完全一致。

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。

7 使用临时查询快速查询SQL（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，快速书写SQL语句操作MaxCompute。

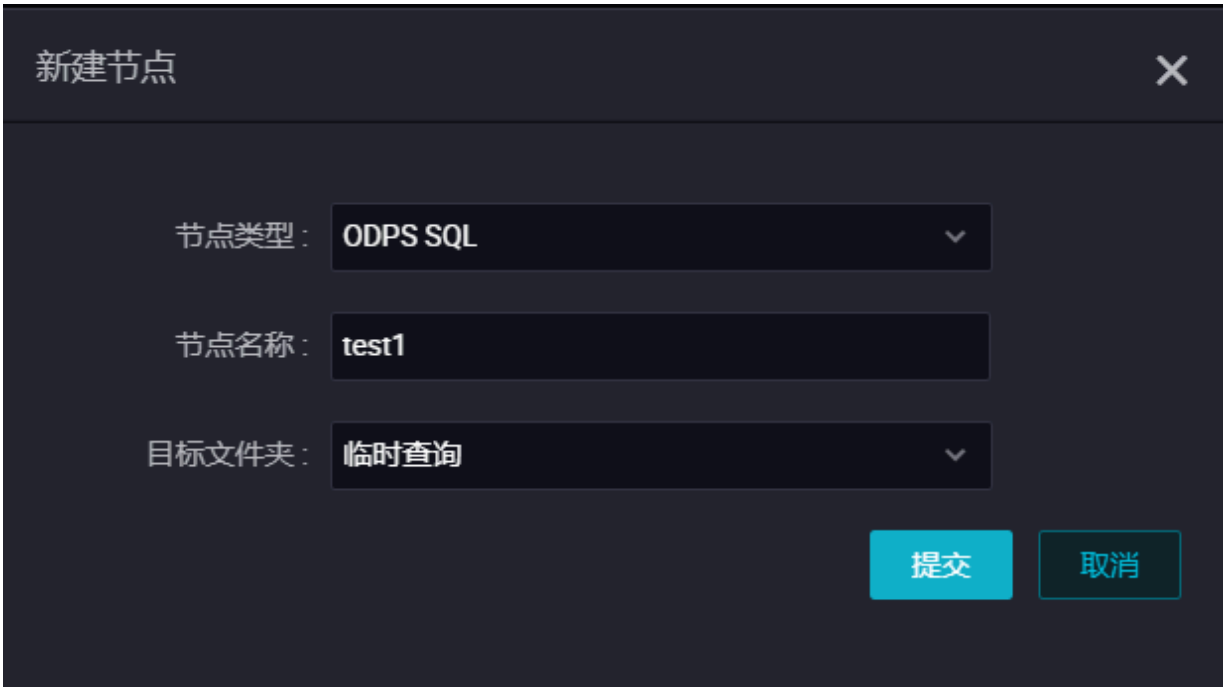
关于临时查询功能的具体信息，请参见临时查询。

进入临时查询

点击DataWorks控制台工作空间列表，选择您需要进入的项目，点击进入数据开发。

直接点击临时查询，右键临时查询，点击新建节点 > ODPS SQL。

在弹框中输入节点名称，点击提交，创建您的临时查询节点。

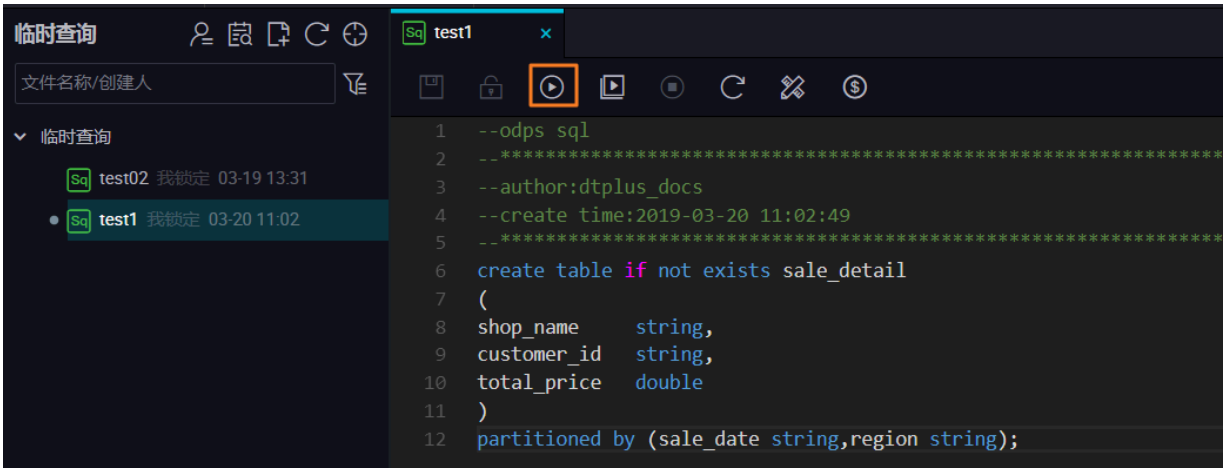


运行SQL

现在您可以在刚刚创建的临时查询节点中运行MaxCompute支持的SQL语句了，我们以运行一个DDL语句创建表为例。

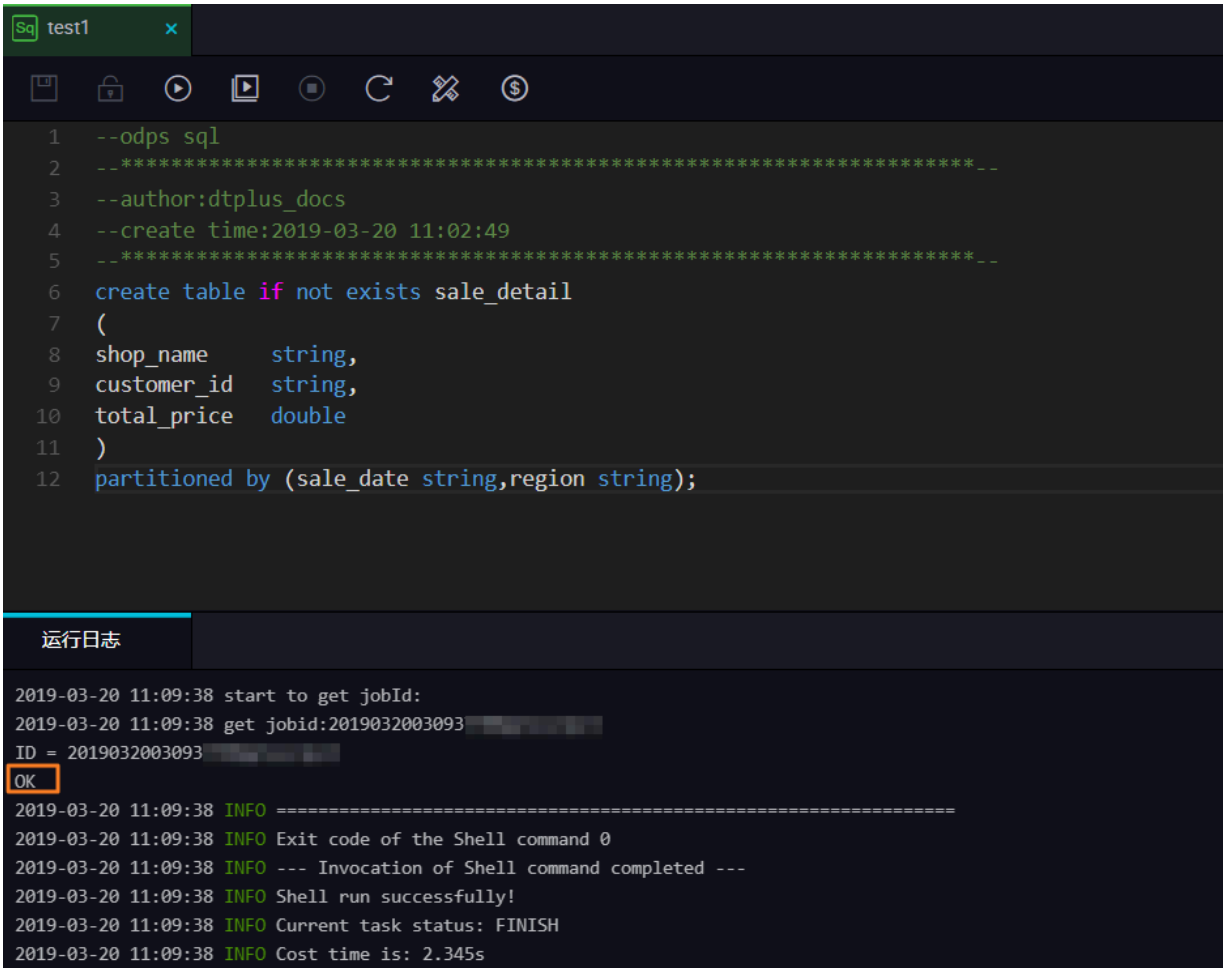
输入建表语句，点击运行即可。

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



在弹框中您可以看到本次运行的费用预估，继续点击运行。

您可以在下方的日志窗口，看到运行情况和最终结果：本次运行成功，结果为OK。



使用同样的方法，您也可以执行查询语句。