

ALIBABA CLOUD

Alibaba Cloud

大数据计算服务
快速入门

文档版本：20220505

 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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.通过MaxCompute客户端使用MaxCompute	05
1.1. 登录MaxCompute客户端	05
1.2. 创建表	06
1.3. 导入数据	13
1.4. 运行SQL命令并导出结果数据	14
1.5. 删除表或MaxCompute项目	17
2.常见问题	20
3.后续指引	21

1.通过MaxCompute客户端使用MaxCompute

1.1. 登录MaxCompute客户端

本文为您介绍如何启动MaxCompute客户端，启动成功后即表示已经登录MaxCompute客户端并进入MaxCompute项目。

前提条件

使用MaxCompute客户端前，请先参见准备工作章节，完成或查看以下内容：

- 准备阿里云账号
- 准备RAM用户
- 开通MaxCompute和DataWorks
- 创建MaxCompute项目
- 添加工作空间成员并设置角色
- 准备环境及安装工具

启动MaxCompute客户端

MaxCompute客户端可通过如下方式启动，您可以任选其中一种：

- 方式一：在MaxCompute客户端安装路径下的bin文件夹中，双击 `odpscmd.bat` 文件（Windows系统），即可启动MaxCompute客户端。返回如下信息，表明已成功连接MaxCompute项目。



```
odpscmd
Aliyun ODPS Command Line Tool
Version 0.35.1-public
©Copyright 2020 Alibaba Cloud Computing Co., Ltd. All rights reserved.
Connecting to http://service. aliyun.com/api, project: _dev
Project timezone: Asia/Shanghai
Connected!
odps@ _dev>
```

- 方式二：在系统的命令行执行窗口，进入MaxCompute客户端安装路径下的bin目录，执行 `odpscmd` 命令（Windows系统）或 `sh odpscmd`（Linux系统或Mac系统），即可启动MaxCompute客户端。返回如下信息，表明已成功连接MaxCompute项目。



```
D:\ aliyun\bin>odpscmd
odpscmd
Aliyun ODPS Command Line Tool
Version 0.35.1-public
©Copyright 2020 Alibaba Cloud Computing Co., Ltd. All rights reserved.
Connecting to http://service. aliyun.com/api, project: _dev
Project timezone: Asia/Shanghai
Connected!
odps@ _dev>
```

 说明 在Ubuntu执行 `sh odpscmd` 会提示报错，请您使用 `./odpscmd` 命令尝试启动。

后续步骤

登录MaxCompute客户端后，即可在MaxCompute项目内创建表，请参见[创建表](#)。

1.2. 创建表

表是MaxCompute的数据存储单元。您需要基于表进行数据处理。本文为您介绍如何创建表。

前提条件

请确认您已满足如下条件：

- 已登录MaxCompute客户端。

更多登录MaxCompute客户端操作，请参见[登录MaxCompute客户端](#)。

- 已准备好保存了待导入数据的CSV或TXT文件。

本文提供的数据文件样例如下：

- 创建非分区表的数据文件：[banking.txt](#)。
- 创建分区表的数据文件：[banking_nocreditcard.csv](#)、[banking_uncreditcard.csv](#)和[banking_yescreditcard.csv](#)。

- 如果是RAM用户创建表，请确认阿里云账号已经授予RAM用户创建表的权限（CreateTable）。

更多查看权限操作，请参见[通过MaxCompute SQL查询权限信息](#)。

更多授权操作，请参见[权限列表](#)。

注意事项

新创建表的列数据类型务必与数据文件的各列数据类型保持一致，如果二者不一致，会导致后续导入数据操作执行失败。

步骤一：设计表

您需要根据待导入的数据内容和最终需要生成的结果数据，确定需要创建的表类型、结构及各列的数据类型。

本文中提供的示例数据内容如下，展示的是银行贷款购房人员的数据。

```
--待创建非分区表的数据。
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
--待创建分区表的数据。
53,technician,married,unknown,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,yes,no,cellular,jun,thu,339,3,6,2,success,-1.7,94.05
5,-39.8,0.729,4991.6,1
39,services,married,high.school,no,no,cellular,apr,fri,185,2,999,0,nonexistent,-1.8,93.075,
-47.1,1.405,5099.1,0
```

您可以识别到数据列数、每列数据的数据类型，同时可以了解各列数据对应的含义，为每列数据定义一个列名称。假设基于示例数据，识别到的信息如下。

类型	说明
表的列数	<p>从示例数据中识别表列数信息如下：</p> <ul style="list-style-type: none">创建非分区表的示例数据包含21列，新建表的列数可设定为21列。 <div><p> 说明 如果表的列数超过示例数据的列数，不足的列将填充NULL。如果表的列数少于示例数据的列数，多余的数据将被丢弃。</p></div> <ul style="list-style-type: none">创建分区表的示例数据包含20列，新建表的列数可设定为21列，其中一列为分区列，您需要在创建表时定义分区。
	<p>从示例数据中识别表列字段信息如下：</p> <ul style="list-style-type: none">非分区表的列字段包含：<ul style="list-style-type: none">age：年龄job：工作类型marital：婚否education：教育程度credit：是否有信用卡housing：是否有房贷loan：是否有贷款contact：联系方式month：月份day_of_week：星期几duration：持续时间campaign：本次活动联系的次数pdays：与上一次联系的时间间隔previous：之前与客户联系的次数poutcome：之前市场活动的结果emp_var_rate：就业变化速率cons_price_idx：消费者物价指数

类型	<div><div>说明</div><div><ul style="list-style-type: none">cons_conf_idx: 消费者信心指数euribor3m: 欧元存款利率</div></div>
表的列字段	<div><div><ul style="list-style-type: none">nr_employed: 职工人数fixed_deposit: 是否有定期存款</div><div><div>分区表的列字段包含:</div><div><ul style="list-style-type: none">age: 年龄job: 工作类型marital: 婚否education: 教育程度housing: 是否有房贷loan: 是否有贷款contact: 联系方式month: 月份day_of_week: 星期几duration: 持续时间campaign: 本次活动联系的次数pdays: 与上一次联系的时间间隔previous: 之前与客户联系的次数poutcome: 之前市场活动的结果emp_var_rate: 就业变化速率cons_price_idx: 消费者物价指数cons_conf_idx: 消费者信心指数euribor3m: 欧元存款利率nr_employed: 职工人数fixed_deposit: 是否有定期存款credit: 是否有信用卡。该列为分区列。</div></div></div>

类型	说明
数据类型	<div>从示例数据中识别表列字段的数据类型信息如下：</div> <ul style="list-style-type: none">• age: BIGINT• job: STRING• marital: STRING• education: STRING• credit: STRING• housing: STRING• loan: STRING• contact: STRING• month: STRING• day_of_week: STRING• duration: STRING• campaign: BIGINT• pdays: DOUBLE• previous: DOUBLE• poutcome: STRING• emp_var_rate: DOUBLE• cons_price_idx: DOUBLE• cons_conf_idx: DOUBLE• euribor3m: DOUBLE• nr_employed: DOUBLE• fixed_deposit: BIGINT

步骤二：创建表

基于示例数据，创建非分区表bank_data和分区表bank_data_pt，用于存储业务数据；创建非分区表result_table1和result_table2，用于存储结果数据。更多表及分区的信息，请参见[表操作](#)和[分区和列操作](#)。创建表的操作流程如下：

1. （可选）在MaxCompute客户端，执行如下命令切换至目标MaxCompute项目。

如果您启动MaxCompute客户端后已在目标MaxCompute项目中，请忽略该步骤。

```
use doc_test_dev;
```

切换成功显示如下。

```
odps@ doc_test_dev>
```

2. 在MaxCompute客户端，创建表bank_data、bank_data_pt、result_table1和result_table2。

更多创建表语法，请参见[创建表](#)。

- 创建非分区表bank_data，命令示例如下。

```
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',
  credit       STRING comment '是否有信用卡',
  housing      STRING comment '是否有房贷',
  loan         STRING comment '是否有贷款',
  contact      STRING comment '联系方式',
  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',
  fixed_deposit BIGINT comment '是否有定期存款'
);
```

- 创建分区表bank_data_pt，并添加分区，命令示例如下。

```
create table if not exists bank_data_pt
(
  age          BIGINT comment '年龄',
  job          STRING comment '工作类型',
  marital      STRING comment '婚否',
  education    STRING comment '教育程度',
  housing      STRING comment '是否有房贷',
  loan         STRING comment '是否有贷款',
  contact      STRING comment '联系方式',
  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',
  fixed_deposit BIGINT comment '是否有定期存款'
)partitioned by (credit STRING comment '是否有信用卡');
alter table bank_data_pt add if not exists partition (credit='yes') partition (credit='no') partition (credit='unknown');
```

- 创建非分区表result_table1，命令示例如下。

```
create table if not exists result_table1
(
  education    STRING comment '教育程度',
  num          BIGINT comment '人数'
);
```

- 创建非分区表result_table2，命令示例如下。

```
create table if not exists result_table2
(
  education    STRING comment '教育程度',
  num          BIGINT comment '人数',
  credit       STRING comment '是否有信用卡'
);
```

步骤三：确认创建结果

创建表后，您需要确认表是否已创建成功，以及表的结构是否正确无误。操作流程如下：

1. 在MaxCompute客户端，执行如下命令确认表已在MaxCompute项目中。

```
show tables;
```

创建成功的表会显示在命令返回结果中，返回结果示例如下。

```
ALIYUN$****:bank_data
ALIYUN$****:bank_data_pt
ALIYUN$****:result_table1
ALIYUN$****:result_table2
```

2. 在MaxCompute客户端，执行如下命令确认表的结构正确无误。

```
--查看bank_data表结构。
desc bank_data;
--查看bank_data_pt表结构。
desc bank_data_pt;
--查看bank_data_pt的分区。
show partitions bank_data_pt;
--查看result_table1表结构。
desc result_table1;
--查看result_table2表结构。
desc result_table2;
```

以bank_data_pt为例，返回结果如下。

```
desc bank_data_pt;
```

```
--返回结果。
```

+-----+			
Owner: ALIYUN\$****	Project: doc_test_dev		
TableComment:			
+-----+			
CreateTime:	2021-05-17 14:02:21		
LastDDLTime:	2021-05-17 14:02:21		
LastModifiedTime:	2021-05-17 14:02:21		
+-----+			
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
housing	string		是否有房贷
loan	string		是否有贷款
contact	string		联系方式
month	string		月份
day_of_week	string		星期几
duration	string		持续时间
campaign	bigint		本次活动联系的次数
pdays	double		与上一次联系的时间间隔
previous	double		之前与客户联系的次数
poutcome	string		之前市场活动的结果
emp_var_rate	double		就业变化速率
cons_price_idx	double		消费者物价指数
cons_conf_idx	double		消费者信心指数
euribor3m	double		欧元存款利率
nr_employed	double		职工人数
fixed_deposit	bigint		是否有定期存款
+-----+			
Partition Columns:			
+-----+			
credit	string	是否有信用卡	
+-----+			

```
show partitions bank_data_pt;
```

```
--返回结果。
```

```
credit=no
```

```
credit=unknown
```

```
credit=yes
```

后续步骤

创建表并确认后，即可将数据文件中的数据导入表中，请参见[导入数据](#)。

1.3. 导入数据

本文为您介绍如何通过MaxCompute客户端，使用Tunnel Upload将本地数据文件中的数据导入创建好的表中。

前提条件

请确认您已满足如下条件：

- 已创建表。
更多创建表操作，请参见[创建表](#)。
- 已将CSV或TXT数据文件下载至本地。
本文提供的数据文件样例如下：
 - 导入非分区表的数据文件：[banking.txt](#)。
 - 导入分区表的数据文件：[banking_nocreditcard.csv](#)、[banking_uncreditcard.csv](#)和[banking_yescreditcard.csv](#)。

步骤一：导入数据

基于Tunnel Upload将本地数据文件内容导入MaxCompute的表中。更多Tunnel操作，请参见[Tunnel命令](#)。导入数据的操作流程如下：

1. 确认数据文件的保存路径。

数据文件的保存路径有两种选择：您可以将文件直接归档至MaxCompute客户端的 `bin` 目录中，上传路径为 `文件名.后缀名`；也可以将文件归档至其他路径下，例如D盘的test文件夹，上传路径为 `D:\test\文件名.后缀名`。

假设，本文中的示例数据文件 `banking.txt` 保存在MaxCompute客户端的 `bin` 目录中，`banking_yescreditcard.csv`、`banking_uncreditcard.csv` 和 `banking_nocreditcard.csv` 保存在D盘的test文件夹下。

2. 在MaxCompute客户端，执行Tunnel Upload命令导入数据。

命令示例如下。

```
tunnel upload banking.txt bank_data;
tunnel upload D:\test\banking_yescreditcard.csv bank_data_pt/credit="yes";
tunnel upload D:\test\banking_uncreditcard.csv bank_data_pt/credit="unknown";
tunnel upload D:\test\banking_nocreditcard.csv bank_data_pt/credit="no";
```

当出现图示OK字样，说明导入完成。

```
odps@ doc_test_dev>tunnel upload D:\test\banking_yescreditcard.csv bank_data_pt/credit="yes";
Upload session: 202105171550120131f60b242a2417
Start upload:D:\test\banking_yescreditcard.csv
Using \r\n to split records
Upload in strict schema mode: true
Total bytes:351 Split input to 1 blocks
2021-05-17 15:47:58 scan block: '1'
2021-05-17 15:47:58 scan block complete, block id: 1
2021-05-17 15:47:58 upload block: '1'
2021-05-17 15:47:59 upload block complete, block id: 1
upload complete, average speed is 351 bytes/s
OK
```

步骤二：确认导入结果

导入数据后，您需要查看导入的目标表与数据文件中的数据条数是否一致，确认所有数据均已成功导入。

本文中的示例数据文件 `banking.txt` 中有41188条数据，

`banking_yescreditcard.csv`、`banking_uncreditcard.csv` 和 `banking_nocreditcard.csv` 分别有3、8597、32588条数据。命令示例如下。

```
select count(*) as num1 from bank_data;
select count(*) as num2 from bank_data_pt where credit="yes";
select count(*) as num3 from bank_data_pt where credit="unknown";
select count(*) as num4 from bank_data_pt where credit="no";
```

返回结果如下。

```
--bank_data中的数据条数。
+-----+
| num1  |
+-----+
| 41188 |
+-----+
--bank_data_pt中credit值为yes的数据条数。
+-----+
| num2  |
+-----+
| 3     |
+-----+
--bank_data_pt中credit值为unknown的数据条数。
+-----+
| num3  |
+-----+
| 8597  |
+-----+
--bank_data_pt中credit值为no的数据条数。
+-----+
| num4  |
+-----+
| 32588 |
+-----+
```

上述命令的返回结果与示例数据文件的数据条数一致，则表明导入成功。

后续步骤

当数据导入到MaxCompute的表中后，您可以在MaxCompute客户端上运行SQL命令处理数据并导出运行结果，请参见[运行SQL命令并导出结果数据](#)。

1.4. 运行SQL命令并导出结果数据

本文介绍如何在MaxCompute客户端上运行SQL命令并通过Tunnel Download导出结果数据。

前提条件

已向MaxCompute的表中导入数据。更多导入数据操作，请参见[导入数据](#)。

背景信息

MaxCompute客户端支持DDL、DML、DQL等操作，您可以结合相应语法运行SQL命令。

MaxCompute客户端的常用SQL命令，请参见[常用命令列表](#)。

步骤一：运行SQL命令

1. 在MaxCompute客户端，基于非分区表bank_data和分区表bank_data_pt，查询各个学历下的贷款买房的单身人士数量，并将结果分别保存到result_table1和result_table2中。

命令示例如下。

```
--查询非分区表bank_data中各个学历下的贷款买房的单身人士数量并将查询结果写入result_table1。
insert overwrite table result_table1
select education, count(marital) as num
from bank_data
where housing = 'yes' and marital = 'single'
group by education;
--查询分区表bank_data_pt中各个学历下的贷款买房的单身人士数量并将查询结果写入result_table2。
set odps.sql.allow.fullscan=true;
insert overwrite table result_table2
select education, count(marital) as num, credit
from bank_data_pt
where housing = 'yes' and marital = 'single'
group by education, credit;
```

2. 查询result_table1和result_table2的写入结果。

命令示例如下。

```
select * from result_table1;
select * from result_table2;
```

返回结果如下。

```
--result_table1中的数据。
+-----+-----+
| education | num      |
+-----+-----+
| basic.4y  | 227      |
| basic.6y  | 172      |
| basic.9y  | 709      |
| high.school | 1641     |
| illiterate | 1         |
| professional.course | 785      |
| university.degree | 2399     |
| unknown   | 257      |
+-----+-----+

--result_table2中的数据。
+-----+-----+-----+
| education | num      | credit |
+-----+-----+-----+
| basic.4y  | 164      | no      |
| basic.4y  | 63       | unknown |
| basic.6y  | 104      | no      |
| basic.6y  | 68       | unknown |
| basic.9y  | 547      | no      |
| basic.9y  | 162      | unknown |
| high.school | 1469     | no      |
| high.school | 172      | unknown |
| illiterate | 1         | unknown |
| professional.course | 721      | no      |
| professional.course | 64       | unknown |
| university.degree | 2203     | no      |
| university.degree | 196      | unknown |
| unknown   | 206      | no      |
| unknown   | 51       | unknown |
+-----+-----+-----+
```

步骤二：导出结果数据

基于Tunnel [Download](#)将MaxCompute表中的数据导出到本地。更多Tunnel操作，请参见[Tunnel命令](#)。导出结果数据的操作流程如下：

1. 确认数据文件的导出路径。

数据文件的导出路径有两种选择：您可以直接将数据以文件形式导出至MaxCompute客户端的 `bin` 目录中，导出路径为 `文件名.后缀名`；也可以导出至其他路径下，例如D盘的test文件夹，导出路径为 `D:\test\文件名.后缀名`。

假设，表result_table1的数据导出至MaxCompute客户端的 `bin` 目录中，result_table2的数据导出至D盘的test文件夹下。

2. 在MaxCompute客户端，执行Tunnel Download命令导出数据。

命令示例如下。

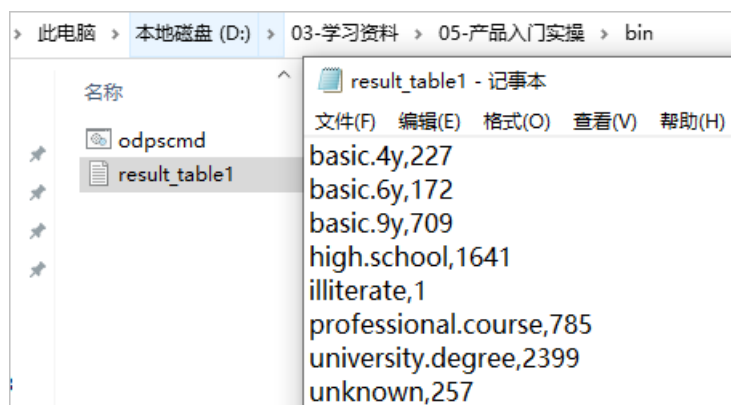
```
tunnel download result_table1 result_table1.txt;
tunnel download result_table2 D:\test\result_table2.csv;
```

当出现图示OK字样，说明导出完成。


```
odps@ doc_test_dev> tunnel download result_table2 D:\test\result_table2.csv;
2021-05-18 18:19:16 - new session: 202105181821291531f60b253e09d6 total lines: 15
2021-05-18 18:19:16 - file [0]: [0, 15), D:\test\result_table2.csv
downloading 15 records into 1 file
2021-05-18 18:19:16 - file [0] start
2021-05-18 18:19:16 - file [0] OK. total: 337 bytes
total: 337 bytes, time: 282 ms, average speed: 1,000 bytes/s
download OK
odps@ doc_test_dev>
```

3. 在导出路径下确认数据文件的存在性及完整性。

result_table1的导出结果确认如下。



result_table2的导出结果确认如下。

	A	B	C
1	basic.4y	164	no
2	basic.4y	63	unknown
3	basic.6y	104	no
4	basic.6y	68	unknown
5	basic.9y	547	no
6	basic.9y	162	unknown
7	high.scho	1469	no
8	high.scho	172	unknown
9	illiterat	1	unknown
10	professio	721	no
11	professio	64	unknown
12	universit	2203	no
13	universit	196	unknown
14	unknown	206	no
15	unknown	51	unknown

后续步骤

如果您不再需要示例数据或MaxCompute项目，可以删除数据或MaxCompute项目，以免产生不必要的资源浪费及账单费用。删除数据或MaxCompute项目的操作，请参见[删除表或MaxCompute项目](#)。

1.5. 删除表或MaxCompute项目

如果您不再需要示例数据或MaxCompute项目，可以删除数据或MaxCompute项目，以免产生不必要的资源浪费及账单费用。本文为您介绍如何删除表或MaxCompute项目。

背景信息

您可以根据实际需要判断是否要保留示例数据及MaxCompute项目：

- 如果需要保留示例数据，以供参考，存储的数据会在账户下产生存储费用，计费规则请参见[存储费用（按量计费）](#)。
- 如果不需要保留示例数据，但要保留MaxCompute项目，您可以删除表。删除表操作，请参见[删除表](#)。
- 如果不需要保留MaxCompute项目，您可以删除MaxCompute项目，删除项目的同时会删除项目中的所有表。删除MaxCompute项目操作，请参见[删除MaxCompute项目](#)。

删除表

您可以在MaxCompute客户端执行如下命令删除MaxCompute项目中的表。

```
drop table bank_data;  
drop table bank_data_pt;  
drop table result_table1;  
drop table result_table2;
```


删除表后，可执行如下命令确认表删除结果。

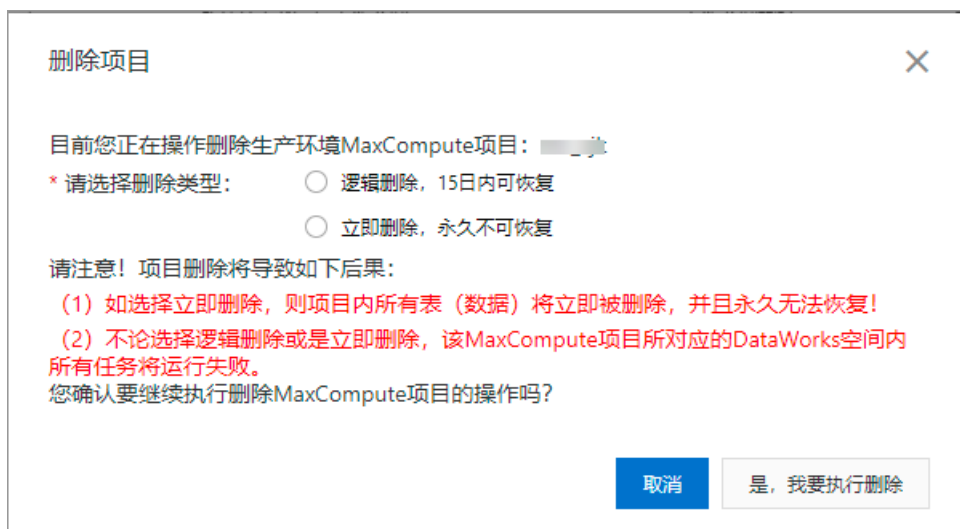
```
show tables;
```

返回结果中无表信息，表明已删除成功。

删除MaxCompute项目

您可以通过阿里云账号（主账号）或具备项目Super_Administrator角色的RAM用户账号登录MaxCompute控制台删除MaxCompute项目。操作步骤如下：

1. 登录[MaxCompute控制台](#)。
2. 在项目管理页签的目标MaxCompute项目的操作列，单击图标，选择删除。
3. 在删除项目对话框，根据实际情况选择删除类型后，单击是，我要执行删除，即可删除MaxCompute项目。



MaxCompute项目支持如下两种删除方式：

- **逻辑删除，15日内可恢复**：删除后项目不可用，15天内您可以在项目管理页签的目标MaxCompute项目的操作列单击恢复，恢复项目。超过15天会彻底删除项目且永久无法恢复。
- **立即删除，永久不可恢复**：彻底删除，永久不可恢复。此方式删除后您可以立即创建同名项目。

② 说明

- 如果MaxCompute项目已经关联对应DataWorks工作空间，删除MaxCompute项目过程中会引导您在工作空间配置界面的计算引擎区域解绑DataWorks工作空间，因此删除MaxCompute项目不会同时删除对应DataWorks工作空间。
- 创建MaxCompute时，会自动创建同名的DataWorks工作空间，单击立即删除，永久不可恢复后，如果创建同名的MaxCompute项目失败，则注意删除同名的DataWorks工作空间后再创建同名MaxCompute项目。

后续步骤

至此，您已完成MaxCompute客户端的快速入门引导操作，如需了解更多MaxCompute使用信息，请参见[后续指引](#)。

2. 常见问题

本文为您介绍执行MaxCompute快速入门工作过程中的常见问题。

快速入门常见问题如下：

- [新创建的MaxCompute项目为什么不支持数据类型自动隐式转换呢？](#)
- [在MaxCompute中创建表的方式有哪些？](#)
- [MaxCompute支持通过什么方式运行SQL语句？](#)
- [数据导入方式有哪些？](#)
- [误删除表数据如何恢复？](#)
- [不同项目的表是否可以相互引用？](#)

新创建的MaxCompute项目为什么不支持数据类型自动隐式转换呢？

当MaxCompute项目打开2.0新数据类型开关（`odps.sql.type.system.odps2=true`）时，存在隐式转换被禁用的风险，您可以关闭2.0新数据类型开关（`odps.sql.type.system.odps2=false`）。

在MaxCompute中创建表的方式有哪些？

您可以通过如下方式创建表：

- 通过MaxCompute客户端的[常用命令](#)创建表。
- 通过MaxCompute Studio可视化界面创建表，请参见[可视化管理内部表、外部表或视图](#)。
- 通过DataWorks可视化界面创建表，请参见[管理表](#)。

MaxCompute支持通过什么方式运行SQL语句？

MaxCompute支持通过DataWorks、MaxCompute客户端、SDK、JDBC等方式运行SQL语句。

数据导入方式有哪些？

除了通过MaxCompute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，更多信息请参见[数据上云工具](#)。

误删除表数据如何恢复？

MaxCompute提供数据备份与恢复功能，MaxCompute会自动备份数据的历史版本（被删除或修改前的数据）并保留一定时间，您可以对保留周期内的数据进行快速恢复，避免因误操作丢失数据。更多备份恢复信息，请参见[备份与恢复](#)。

不同项目的表是否可以相互引用？

不同项目间的表引用需要进行授权。MaxCompute提供了ACL（基于对象）、Policy、跨项目访问资源等多种授权方式，更多信息请参见[权限概述](#)。

3.后续指引

完成快速入门引导操作后，建议您阅读本文，深入了解MaxCompute并熟悉文档布局，以便在使用MaxCompute过程中快捷地获取到相关指导文档。

您可以根据想要了解的内容，单击如下对应链接获取相关指导文档。

如果您想了解...		说明	相关文档
数据迁移	上传下载数据	MaxCompute提供Tunnel批量数据通道、Streaming Tunnel流式数据写入通道和DataHub实时数据通道三种数据上传下载的通道，同时还支持基于DataHub和Tunnel提供的SDK实现上传下载数据，满足多样化场景需求。	<ul style="list-style-type: none">介绍：数据迁移实践：数据迁移最佳实践
	大规模数据迁移	MaxCompute提供MMA（MaxCompute Migration Assist）迁移工具支撑大规模数据迁移。	
开发与分析	数据类型	MaxCompute支持三种数据类型版本，为您介绍各版本支持的数据类型、适用场景及各版本间的差异，避免开发过程中出现数据类型无法解析问题。	数据类型
	常用命令	提供开发过程中的针对 表 、 资源 、 函数 、 实例 等的常用操作及相关命令信息。	常用命令列表
	开发作业	MaxCompute支持多种计算模型SQL、SQLML、PyODPS、MapReduce、Spark、Mars、Graph，满足多样化开发需求。 MaxCompute支持外部表功能，旨在提供处理除MaxCompute内部表以外的其他数据的能力。 此外，MaxCompute还支持SDK及JDBC用户接口实现海量数据开发处理。	<ul style="list-style-type: none">介绍：<ul style="list-style-type: none">SQLSQLMLPyODPSMapReduceMarsSparkGraph外部表Java SDKPython SDKJDBC实践：数据开发最佳实践
	分析作业	MaxCompute提供Logview、优化器功能，为定位问题、优化作业提供帮助。	<ul style="list-style-type: none">介绍：<ul style="list-style-type: none">查看作业运行信息优化器信息收集实践：作业优化最佳实践

如果您想了解...		说明	相关文档
访问控制	用户、角色与授权	MaxCompute为用户、角色提供ACL、Package、Label等多种访问控制方式，提供完善的安全保障机制。	<ul style="list-style-type: none">• 用户规划与管理• 角色规划与管理• 权限列表• Label权限控制
安全管理	项目、数据安全	MaxCompute提供项目及数据的安全保护机制，例如动态脱敏、数据加密等。	<ul style="list-style-type: none">• 安全参数列表• 数据保护机制• 数据加密
运维管理	资源、作业管理	MaxCompute提供元数据及MaxCompute管家工具，方便您精细化管理作业和资源。	<ul style="list-style-type: none">• Information Schema• MaxCompute管家