ALIBABA CLOUD

Alibaba Cloud

MaxCompute

Quick Start

Document Version: 20211228

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Use the MaxCompute client

1.1. Start the MaxCompute client

This topic describes how to start the MaxCompute client. After the client is started, a MaxCompute project is connected.

Prerequisites

Before you use the MaxCompute client, make sure that you have completed the following operations:

- Create an Alibaba Cloud account
- Prepare a RAM user
- Activate MaxCompute and DataWorks
- Create a MaxCompute project
- Add a workspace member and configure roles
- Prepare an environment and install required development tools

Start the MaxCompute client

You can start the MaxCompute client by using one of the following methods:

- Method 1: In the bin folder under the installation path of the MaxCompute client, double-click the `odpscmd.bat` file to start the MaxCompute client. This method is used for the Windows operating system. If the information shown in the following figure is returned, the MaxCompute project is connected.

- Method 2: In the CLI of the system, go to the bin folder under the installation path of the MaxCompute client and run the `odpscmd` command for the Windows operating system or run the `sh odpscmd` command for the macOS or Linux operating system to start the MaxCompute client. If the information shown in the following figure is returned, the MaxCompute project is connected.

What's next

After you start the MaxCompute client, you can create tables in the connected MaxCompute project. For more information about how to create a table, see Create tables.

1.2. Create tables

Table is the unit for storing data in MaxCompute. You must process data in MaxCompute based on tables. This topic describes how to create tables in MaxCompute.

Prerequisites

Make sure that the following requirements are met:

- The MaxCompute client is started.

  For more information about how to start the MaxCompute client, see Start the MaxCompute client.

- A CSV or TXT file that contains the data you want to import is prepared.

  The following sample files are used in this topic:

  - The file that is used to create a non-partitioned table: banking.txt.
  - The files that are used to create a partitioned table: banking_nocreditcard.csv, banking_uncreditcard.csv, and banking_yescreditcard.csv.

- The RAM user that you want to use to create tables is granted the CREATE TABLE permission. If you want to use your Alibaba Cloud account to create tables, ignore this requirement.

  For more information about how to check the permissions of a RAM user, see View permissions of a specified user.

  For more information about how to grant permissions to a RAM user, see Authorize users.

Usage notes

The data types of the columns in the table that you want to create must be consistent with the data types of the columns in the data files. If they are inconsistent, data cannot be imported to the table.

Step 1: Design tables

Determine the table type, table structure, and data type of each column based on the content of the data that you want to import and the expected result data.

The sample files used in this topic contain the following data of mortgagors:

```
-- Data that is used to create a non-partitioned table.
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
-- Data that is used to create a partitioned table.
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
```

You can count the number of columns, identify the data type of each column, and name each column based on the meaning of each column of data. The following table describes the information that you can obtain from the sample files.

Item	Description
Number of columns	Count the number of columns in each sample file and determine the number of columns in the table that you want to create: • The sample file that is used to create a non-partitioned table contains 21 columns. You can create a non-partitioned table that contains 21 columns based on the sample file. **Note** If the number of columns in the table exceeds the number of columns in the sample file, the additional columns in the table are filled with NULL. If the number of columns in the table is less than the number of columns in the sample file, the excessive data in the sample file is discarded. • Each of the sample files that are used to create a partitioned table contains 20 columns. You can create a partitioned table that contains 21 columns with one column as the partition key column based on the sample files. You must define partitions when you create the table.
	Identify the fields in each sample file and determine the fields in the table that you want to create: • Fields of the non-partitioned table: ◦ age: age ◦ job: job type ◦ marital: marital status ◦ education: educational level ◦ credit: have a credit card ◦ housing: mortgagors ◦ loan: have loans ◦ contact: contact information ◦ month: month

Item	Description
Fields of the table	○ day_of_week: day of the week ○ duration: duration ○ campaign: number of contacts for the current campaign ○ pdays: time elapsed after the last contact ○ previous: number of previous contacts ○ poutcome: results of previous marketing activities ○ emp_var_rate: employment variation rate ○ cons_price_idx: consumer price index ○ cons_conf_idx: consumer confidence index ○ euribor 3m: 3-month Euro Interbank Offered Rate (Euribor) ○ nr_employed: number of employees ○ fixed_deposit: have a time deposit • Fields of the partitioned table: ○ age: age ○ job: job type ○ marital: marital status ○ education: educational level ○ housing: mortgagors ○ loan: have loans ○ contact: contact information ○ month: month ○ day_of_week: day of the week ○ duration: duration ○ campaign: number of contacts for the current campaign ○ pdays: time elapsed after the last contact ○ previous: number of previous contacts ○ poutcome: results of previous marketing activities ○ emp_var_rate: employment variation rate ○ cons_price_idx: consumer price index ○ cons_conf_idx: consumer confidence index ○ euribor 3m: 3-month Euribor ○ nr_employed: number of employees ○ fixed_deposit: have a time deposit ○ credit: have a credit card (used as the partition key column)

Item	Description
Data types	Identify the data type of each column in the sample files: • 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

Step 2: Create tables

Create a non-partitioned table named bank_data and a partitioned table named bank_data_pt based on the sample files to store business data. Create non-partitioned tables named result_table1 and result_table2 to store result data. For more information about operations on tables and partitions, see Table operations and Partition and column operations. Perform the following operations to create the tables:

1. (Optional)On the MaxCompute client, run the following command to switch to the MaxCompute project that you want to use.

   If the MaxCompute project is opened when you start the MaxCompute client, ignore this step.

   ```
   use doc_test_dev;
   ```

   If the command is successfully run, the following information is returned:

   ```
   odps@ doc_test_dev>
   ```

2. On the MaxCompute client, create the bank_data, bank_data_pt, result_table1, and result_table2 tables.

   For more information about the CREATE TABLE syntax, see Create a table.

   - Create the non-partitioned table bank_data. Sample statement:

```
create table if not exists bank_data
(
 age             BIGINT comment 'age',
 job             STRING comment 'job type',
 marital         STRING comment 'marital status',
 education       STRING comment 'education level',
 credit          STRING comment 'have a credit card',
 housing         STRING comment 'mortgagors',
 loan            STRING comment 'have loans',
 contact         STRING comment 'contact information',
 month           STRING comment 'month',
 day_of_week     STRING comment 'day of the week',
 duration        STRING comment 'duration',
 campaign        BIGINT comment 'number of contacts for the current campaign',
 pdays           DOUBLE comment 'time elapsed after the last contact',
 previous        DOUBLE comment 'number of previous contacts',
 poutcome        STRING comment 'results of previous marketing activities',
 emp_var_rate    DOUBLE comment 'employment variance rate',
 cons_price_idx  DOUBLE comment 'consumer price index',
 cons_conf_idx   DOUBLE comment 'consumer confidence index',
 euribor3m       DOUBLE comment '3-month Euribor',
 nr_employed     DOUBLE comment 'number of employees',
 fixed_deposit   BIGINT comment 'have a time deposit'
);
```

- Create the partitioned table bank_data_pt and add partitions to the table. Sample statement:

```
create table if not exists bank_data_pt
(
 age             BIGINT comment 'age',
 job             STRING comment 'job type',
 marital         STRING comment 'marital status',
 education       STRING comment 'education level',
 housing         STRING comment 'mortgagors',
 loan            STRING comment 'have loans',
 contact         STRING comment 'contact information',
 month           STRING comment 'month',
 day_of_week     STRING comment 'day of the week',
 duration        STRING comment 'duration',
 campaign        BIGINT comment 'number of contacts for the current campaign',
 pdays           DOUBLE comment 'time elapsed after the last contact',
 previous        DOUBLE comment 'number of previous contacts',
 poutcome        STRING comment 'results of previous marketing activities',
 emp_var_rate    DOUBLE comment 'employment variance rate',
 cons_price_idx  DOUBLE comment 'consumer price index',
 cons_conf_idx   DOUBLE comment 'consumer confidence index',
 euribor3m       DOUBLE comment '3-month Euribor',
 nr_employed     DOUBLE comment 'number of employees',
 fixed_deposit   BIGINT comment 'have a time deposit'
)partitioned by (credit STRING comment 'have a credit card');
alter table bank_data_pt add if not exists partition (credit='yes') partition (credit
='no') partition (credit='unknown');
```

- Create the non-partitioned table result_table1. Sample statement:

```
create table if not exists result_table1
(
 education   STRING comment 'education level',
 num         BIGINT comment 'number of persons'
);
```

- Create the non-partitioned table result_table2. Sample statement:

```
create table if not exists result_table2
(
 education   STRING comment 'education level',
 num         BIGINT comment 'number of persons'
 credit      STRING comment 'have a credit card',
);
```

Step 3: Check the table creation results

Perform the following operations to check whether the tables are created and whether the table schemas are correct:

1. On the MaxCompute client, run the following command to check whether the tables that you created are in the MaxCompute project as expected:

   ```
   show tables;
   ```

   The names of the created tables are returned:

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

2. On the MaxCompute client, execute the following statements to check whether the table schemas are correct:

   ```
   -- View the schema of bank_data.
   desc bank_data;
   -- View the schema of bank_data_pt.
   desc bank_data_pt;
   -- View the partitions of bank_data_pt.
   show partitions bank_data_pt;
   -- View the schema of result_table1.
   desc result_table1;
   -- View the schema of result_table2.
   desc result_table2;
   ```

   Returned result of the desc bank_data_pt; statement:

   ```
   desc bank_data_pt;
   -- The following result is returned:
   +------------------------------------------------------------------------------------+
   | 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     |       | age                                               |
| job             | string     |       | job type                                           |
| marital         | string     |       | marital status
|
| education       | string     |       | educational level
|
| housing         | string     |       | mortgagors                                         |
| loan            | string     |       | have loans                                         |
| contact         | string     |       | contact information
|
| month           | string     |       | month                                               |
| day_of_week     | string     |       | day of the week
|
| duration        | string     |       | duration                                           |
| campaign        | bigint     |       | number of contacts for the current campaign
|
| pdays           | double     |       | time elapsed after the last contact
|
| previous        | double     |       | number of previous contacts
|
| poutcome        | string     |       | results of previous marketing activities
|
| emp_var_rate    | double     |       | employment variation rate
|
| cons_price_idx  | double     |       | consumer price index
|
| cons_conf_idx   | double     |       | consumer confidence index
|
| euribor3m       | double     |       | 3-month Euribor
|
| nr_employed     | double     |       | number of employees
|
| fixed_deposit   | bigint     |       | have a time deposit
|
+-------------------------------------------------------------------------------------------+
| Partition Columns:                                                                        |
+-------------------------------------------------------------------------------------------+
| credit          | string     | have a credit card
|
+-------------------------------------------------------------------------------------------+
```

```
show partitions bank_data_pt;
-- The following result is returned:
credit=no
credit=unknown
credit=yes
```

What to do next

After you create the tables and confirm that the table schemas are correct, you can import data in the sample files to the tables. For more information about how to import data, see Import data to tables.

1.3. Import data to tables

This topic describes how to import data from data files on your computer to MaxCompute tables by running Tunnel Upload commands on the MaxCompute client.

Prerequisites

- The tables to which you want to import data are created.

  For more information about how to create tables, see Create tables.
- The CSV or TXT data files from which you want to import data are downloaded to your computer.

  The following sample files are used in this topic:
  - The file whose data you want to import to a non-partitioned table: banking.txt
  - The files whose data you want to import to a partitioned table: banking_nocreditcard.csv, banking_uncreditcard.csv, and banking_yescreditcard.csv

Step 1: Import data

Import data from data files on your computer to MaxCompute tables by running Tunnel Upload commands. For more information about Tunnel operations, see Tunnel commands.

1. Obtain the storage paths of the data files.

   You can store the files in the `bin` directory of the MaxCompute client. In this case, you must specify a storage path in the `File name.File name extension` format in the import command. You can also store the files in another directory, such as the test folder on drive D. In this case, you must specify a storage path in the `D:\test\File name.File name extension` format in the import command.

   In this example, `banking.txt` is stored in the `bin` directory of the MaxCompute client, and `banking_yescreditcard.csv`, `banking_uncreditcard.csv`, and `banking_nocreditcard.csv` are stored in the test folder on drive D.
2. On the MaxCompute client, run the following Tunnel Upload commands to import data:

   ```
   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";
   ```

   When OK is returned, the data is imported.

```
odps@ doc_test_dev>tunnel upload D:\test\banking_yescreditcard.csv bank_data_pt/credit="yes";
Upload session: 20210517155012013lf60b242a2417
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
```

Step 2: Check the import results

After you import data to a table or partition, you must check whether the number of data records in the table or partition is consistent with that in the data file. If the numbers are inconsistent, the data is not completely imported.

In the example, `banking.txt` contains 41,188 records, `banking_yescreditcard.csv` contains 3 records, `banking_uncreditcard.csv` contains 8,597 records, and `banking_nocreditcard.csv` contains 32,588 records. Run the following commands:

```
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";
```

The following information is returned:

```
-- The number of data records in bank_data.
+------------+
| num1       |
+------------+
| 41188      |
+------------+
-- The number of data records in the partition for which the value of credit is yes in bank
_data_pt.
+------------+
| num2       |
+------------+
| 3          |
+------------+
-- The number of data records in the partition for which the value of credit is unknown in
bank_data_pt.
+------------+
| num3       |
+------------+
| 8597       |
+------------+
-- The number of data records in the partition for which the value of credit is no in bank_
data_pt.
+------------+
| num4       |
+------------+
| 32588      |
+------------+
```

The returned numbers are consistent with the numbers of data records in the sample files. This indicates that data in the sample files is completely imported to the tables.

What to do next

After the data is imported to MaxCompute tables, you can run SQL statements on the MaxCompute client to process the data and run commands to export the result data. For more information about how to execute SQL statements and export result data, see Execute SQL statements and export the result data.

1.4. Execute SQL statements and export the result data

This topic describes how to execute SQL statements on the MaxCompute client and use Tunnel Download commands to download the result data.

Prerequisites

Data is imported to a MaxCompute table. For more information about how to import data, see Import data to tables.

Context

You can execute DDL, DML, and DQL statements on the MaxCompute client based on the provided syntax.

For more information about the common SQL statements that can be executed on the MaxCompute client, see Common SQL statements.

Step 1: Execute SQL statements

1. On the MaxCompute client, query the number of single persons with home loans at each education level from the non-partitioned table bank_data and the partitioned table bank_data_pt and write the result data to the result_table1 and result_table2 tables:

```
-- Query the number of single persons with home loans at each education level from the
non-partitioned table bank_data and write the result data to 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;
-- Query the number of single persons with home loans at each education level from the
partitioned table bank_data_pt and write the result data to 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. Query the data that is written to result_table1 and result_table2:

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

The following information is returned:

```
-- Data in result_table1
+------------+------------+
| education  | num        |
+------------+------------+
| basic.4y   | 227        |
| basic.6y   | 172        |
| basic.9y   | 709        |
| high.school | 1641       |
| illiterate | 1          |
| professional.course | 785        |
| university.degree | 2399       |
| unknown    | 257        |
+------------+------------+
-- Data in result_table2
+------------+------------+------------+
| 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    |
+------------+------------+------------+
```

Step 2: Export result data

Export data from MaxCompute tables to your computer by running Tunnel Download commands. For more information about Tunnel operations, see Tunnel commands.

1. Determine an export path.

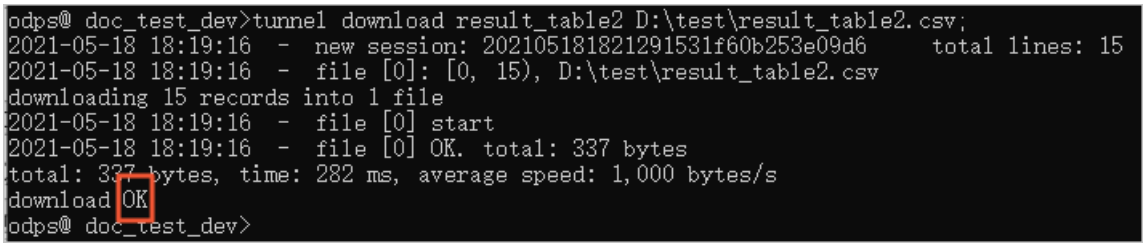
   You can export the data as a file to the `bin` directory of the MaxCompute client. In this case, you must specify an export path in the `File name.File name extension` format in the export command. You can also export the data as a file to another directory, such as the test folder on drive D. In this case, you must specify an export path in the `D:\test\File name.File name extension` format in the export command.

   In this example, the data in result_table1 is exported to the `bin` directory of the MaxCompute client, and the data in result_table2 is exported to the test folder on drive D.

2. On the MaxCompute client, run the following Tunnel Download commands to export data.

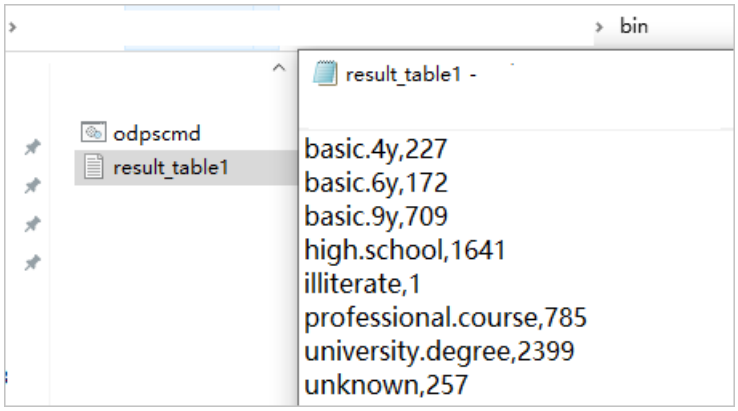
```
tunnel download result_table1 result_table1.txt;
tunnel download result_table2 D:\test\result_table2.csv;
```

If OK is returned, the data is exported.

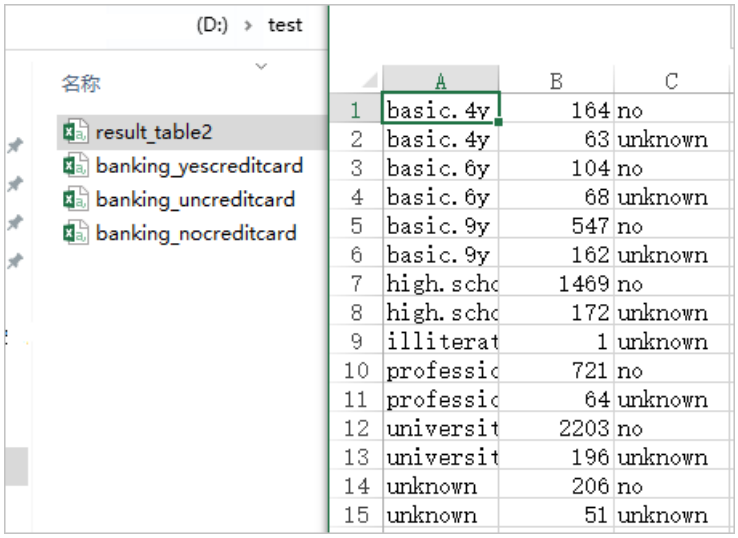


3. Check whether the data is completely exported to the export paths.

The following figure shows the data exported from result_table1.



The following figure shows the data exported from result_table2.



Additional information

If you no longer need to use the sample data or the MaxCompute project in which the sample data is used, you can delete the data or the MaxCompute project to reduce resource consumption and storage fees. For more information about how to delete data and MaxCompute projects, see Delete a table or a MaxCompute project.

1.5. Delete a table or a MaxCompute project

If you no longer need to use a set of sample data or a MaxCompute project, you can delete the data or the MaxCompute project to reduce resource consumption and storage fees. This topic describes how to delete a MaxCompute project or a table in a MaxCompute project.

Context

You can determine whether to retain a specific set of sample data and the MaxCompute project in which the sample data is used based on your business requirements.

- If you retain the sample data for reference, you are charged for the storage of the data. For more information about storage pricing, see Storage pricing (pay-as-you-go).
- If you do not want to retain the sample data but still need to use the MaxCompute project in which the sample data is used, you can delete the related tables. For more information about how to delete a table, see Delete tables.
- If you do not want to retain the MaxCompute project, you can delete the MaxCompute project. If you delete the project, all tables in the project are also deleted. For more information about how to delete a MaxCompute project, see Delete a MaxCompute project.

Delete tables

Run the following commands on the MaxCompute client to delete the specified tables from a MaxCompute project:

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

You can run the following command to check whether the tables are deleted:

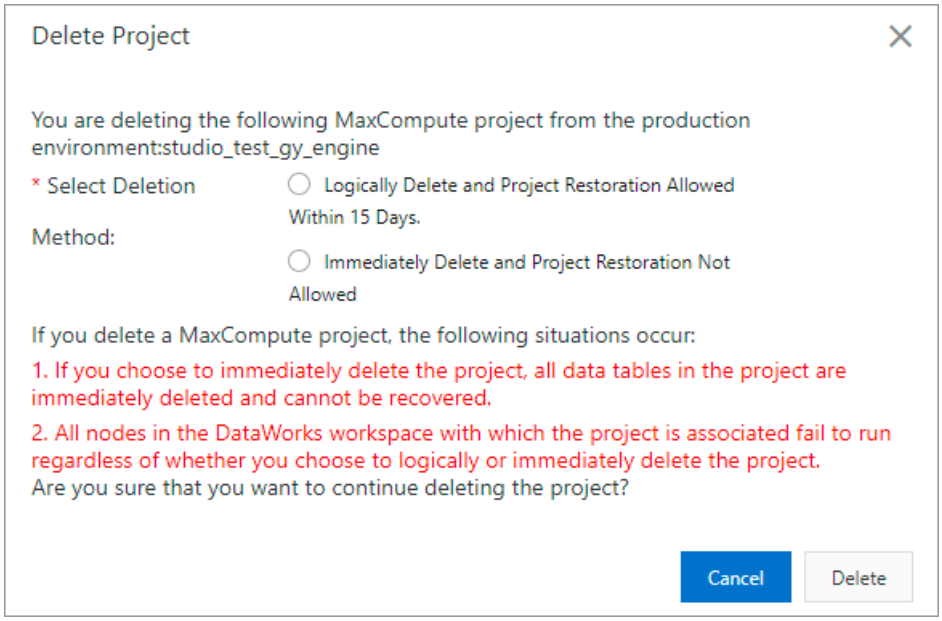
```
show tables;
```

If no table information is returned, the tables are deleted.

Delete a MaxCompute project

You can log on to the MaxCompute console by using your Alibaba Cloud account or the credentials of a RAM user that is assigned the Super_Administrator role and delete the MaxCompute project that you no longer use.

1. Log on to the MaxCompute console.
2. On the **Project management** tab, find your project, click the icon in the **Actions** column, and then select **Delete**.
3. In the **Delete Project** dialog box, select a deletion method and click **Delete**.



You can use one of the following methods to delete a MaxCompute project:

- **Logically Delete and Project Restoration Allowed Within 15 Days.**: If you delete a project by using this method, the project becomes unavailable. To restore the project, find the project and click **Restore** in the **Actions** column on the **Project management** tab within 15 days. After 15 days, the project is permanently deleted and cannot be restored.
- **Immediately Delete and Project Restoration Not Allowed**: If you delete a project by using this method, the project is permanently deleted and cannot be restored. After you delete a project by using this method, you can immediately create a project with the same name.

> **Note** When you delete a MaxCompute project that is associated with a DataWorks workspace, you are prompted to disassociate the MaxCompute project from the DataWorks workspace in the **Compute Engine Information** section of the **Configuration** page. This way, the DataWorks workspace that is associated with the MaxCompute project is retained when you delete the MaxCompute project.

What to do next

For more information about how to use MaxCompute, see Additional information.

2.Additional information

After you get a quick start on the query editor and the MaxCompute client, we recommend that you read this topic to have an overview of the features provided by MaxCompute. You can click the listed links for each feature to go to the related topics.

You can click a link for a feature to go to the related topic.

Feature		Description	References
Data migration	Data upload and download	MaxCompute provides three types of data upload and download channels: Tunnel for batch data uploads and downloads, Streaming Tunnel for streaming data writes, and DataHub for real-time data uploads and downloads. MaxCompute also allows you to upload and download data in a variety of scenarios by using the SDKs provided by DataHub and Tunnel.	• Overview: Data migration • Practices: Best practices on data migration
	Large-scale data migration	MaxCompute provides a tool named MaxCompute Migration Assist (MMA) for you to migrate large amounts of data.	
Development and analysis	Data types	MaxCompute supports three data type editions. You can read the references provided to have an understanding of the data types supported by each edition, the use scenarios of each edition, and the differences among these editions. If you use invalid data types, the system may fail to parse data during data development.	Data type editions
	Common commands	You can read the references to have an understanding of the common operations and commands related to tables, resources, functions, and instances.	Common SQL statements

analysis Feature		Description	References
	Job development	MaxCompute provides various computing models, such as SQL, SQLML, PyODPS, MapReduce, Spark, Mars, and Graph to meet diverse development requirements. MaxCompute provides the external table feature for you to process data that is not stored in the internal tables of MaxCompute. MaxCompute also provides SDKs and a Java Database Connectivity (JDBC) interface for you to develop and process large amounts of data.	• Overview: ◦ SQL ◦ PyODPS ◦ Overview ◦ Mars ◦ Spark ◦ Graph ◦ External table ◦ Java SDK ◦ Python SDK ◦ JDBC • Practices: Best practices on data development
	Job analysis	MaxCompute provides the Logview feature and an optimizer to help you identify problems and optimize jobs.	• Overview: ◦ View job information ◦ Collect information for the optimizer of MaxCompute • Practices: Best practices on job optimization
Access control	Users, roles, and authorization	MaxCompute provides multiple access control methods, such as access control list (ACL)-based authorization, access control for packages, label-based authorization, for you to manage users and roles. MaxCompute provides a comprehensive mechanism to ensure security.	• Manage users • Manage roles • Authorize users • Column-level access control
Security management	Project and data security	MaxCompute provides comprehensive mechanisms, such as dynamic data masking and data encryption, to ensure the security of projects and data.	• Project security configurations • Project data protection • Data encryption
O&M	Resource and job management	MaxCompute provides the Information Schema and MaxCompute Management tools for you to manage jobs and resources in a fine-grained manner.	• Information Schema • Use MaxCompute Management