Alibaba Cloud MaxCompute

Product Introduction

Issue: 20180806

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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 content of the Alibaba Cloud website, including but not limited to works, products , images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content 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 contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
⛔	This warning information 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.
⚠	This warning information 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 restore business.
📋	This indicates warning information, supplementary instructions, and other content that the user must understand.	📋 **Note:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	📋 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
`Italics`	It is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *`[-all|-t]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 What is MaxCompute

MaxCompute is a big data processing platform that processes and stores massive batch structural data to provide effective data warehousing solutions and big data modeling. MaxCompute supports a variety of classic distributed computing models that enable you to solve massive data calculation problems while reducing business costs, and maintaining data security.

MaxCompute seamlessly integrates with DataWorks, which provides one-stop data synchronization, task development, data workflow development, data operation and maintenance, and data management for MaxCompute. For more information, see *DataWorks*.

MaxCompute mainly used to store and compute batch structured data, providing solutions for mass data warehouses and analytical modeling services for large data sets. With the continuous enrichment and improvement of social data collection methods, more and more industry data have been accumulated. As a result, many businesses have masses of data (GB-, TB-, and even PB-levels) unable to be processed by traditional software service providers.

For analyzing big data, the limited of the processing capacity of any single server forces data analysts to use distributed computing. But the distributed computing model puts forward the high requirement to the data analysts, and it is not easy to maintain. Data analysts must be aware of both the business demand and the bottom computing models to use distributed computing models. The purpose of maxcompute is to provide you with a convenient way to analyze and handle massive amounts of data, you can achieve the goal of analyzing big data without having to care about the details of distributed computing.

Note:
MaxCompute has been widely used inside Alibaba, such as for data warehouse and BI analysis of large Internet enterprises, log analysis of websites, transaction analysis of e-commerce websites, discovering user features and interests, and more.

Benefits of MaxCompute

- **Large-scale computing and storage**

  MaxCompute is suitable for storage and computing large volumes of data (up to PB-level).

- **Multiple computation models**

  MaxCompute supports data processing methods based on SQL, MapReduce, Graph, MPI iteration algorithm, and other programming models.

- **Strong data security**

  MaxCompute supports all offline business analysis of Alibaba Group with robust multi-layer sandbox protection and monitoring.
- **Low-cost**

  MaxCompute can help reduce procurement costs by 20%-30% compared with on-premises private cloud models.

Function

- **MaxCompute Tunnel**
  - Supports large volumes of historical data channels

    Tunnel provides high concurrency data upload and download services. You can use Tunnel to import TB/PB level data from various heterogeneous data sources into MaxCompute, or export data from MaxCompute. As the unified channel for MaxCompute data transmission, Tunnel provides stable and high- throughput services. Tunnel provides RESTful APIs and a Java SDK to facilitate programming.
  - Real-time and incremental data channels

    For real-time data upload scenarios, maxcompute provides a data hub service with low latency and easy-to-use, it is especially suitable for incremental data import. The datahub also supports a variety of data transfer plug-ins, such as logstroudsburg, flume, fluentd, sqoop, and so on, support Log service log at the same time Delivery logs in the service go to maxcompute, and then use dataworks for log analysis and mining.
- **Computing and analysis tasks**

  MaxCompute provides multiple computing models.
  - *SQL*: In MaxCompute, data is stored in tables. MaxCompute provides an SQL query function for the external interface. You can operate MaxCompute similarly to a traditional database software but with the ability to process PB-level data.

    Note:

    - MaxCompute SQL does not support transactions, index, or UPDATE/DELETE operations.

    - MaxCompute SQL syntax differs from Oracle and MySQL, notably, you cannot seamlessly migrate SQL statements of other databases into MaxCompute. For more information, see *SQL syntax*.
    - After you submit MaxCompute jobs, the jobs can be queued and scheduled for execution. MaxCompute SQL can complete queries at the second- to millisecond-level.
    - MaxCompute SQL can complete queries at the second- to millisecond-level. If you have database operation experience, you can quickly get familiar with the usage of MaxCompute SQL.

  - *UDF*: A user-defined function.

    MaxCompute provides numerous *built-in functions* to meet your computing needs, while also supporting the creation of custom functions.
  - MapReduce: MapReduce is a Java MapReduce programming model provided by MaxCompute and uses the Java programming interface. It simplifies the developmen t process, however, users are recommended to have a basic understanding of the concept of distribution, and relevant programming experience, before using MapReduce. Maxcompute MapReduce provide you with Java programming interface.
  - Graph: Graph in MaxCompute is a processing framework designed for iterative graph computing. Graph jobs use graphs to build models. Graphs are composed of vertices and edges. Vertices and edges contain values. After performing iterative graph editing and evolution, you can get the final result. Typical applications include *PageRank*, *SSSP algorithm*, and *K-Means algorithm*.
- **SDK**

  A convenient toolkit provided for developers. For more information, see *MaxCompute SDK*.
- **Security**

  MaxCompute provides powerful security services that fully protects user data. For more information about each function model, see *MaxCompute Security Manual*.

Subsequent steps

Now that you've learned the product advantages, features, and more about MaxCompute, you can move on to the next tutorial. In this tutorial, you will learn about the charges for maxcompute, for more information see *product pricing*.

2 History

Since the founding of the Alibaba Cloud in September 2009, the vision is the first platform for computing/sharing data. In April 2010, with the launch of the loan business of Ali Finance, ODPS was formally put into production. In 2012 a Unified Data Platform was established. In 2013, have the ability to process massive data on a large scale. From 2014 ~ In 2015, the big data platform began to mature, and in 2016 maxcomputerte2.0 was born, the vision was gradually realized.

Key milestones

- 2010.04: Named ODPS, the service is released as an operational component of Alibaba Group 's Ant Financial
- 2013.05: ODPS is released for beta testing
- 2013.07: ODPS v1.0 is released as a commercially available service. A single cluster contains 5,000 servers, with support for multi-level clusters available.
- 2016.09: Renamed to MaxCompute, v2.0 is released as a commercially available service.

3 Definition

3.1 Project

Project is the basic unit of operation in MaxCompute. It is similar to the concept of Database or Schema in traditional databases, and sets the boundary for MaxCompute multi-user isolation and access control. You can have multiple project permissions at the same time and, by granting relevant authorization, users can access the objects of another project in their own project, such as *Table*, *Resource,*, *Function* and *Instance*.

To enter a project (in this example, ‘my project’), run the Use Project command, as follows:

```
use my_project -- Enter a project named 'my_project'.
```

After running the preceding command, you can enter a project named my project and all objects in this project can be operated. Use Project is a command provided by the MaxCompute client. For more commands, see *Common Commands*.

3.2 Table

A table is the data storage unit in MaxCompute. A table is a two-dimensional data structure composed of rows and columns. Each row represents a record, and each column represents a field with the same data type. One record can contain one or more columns. The column name and data type comprise the schema of a table.

The operating objects (input, output) of various computing tasks in MaxCompute are tables. You can create a table, delete a table, and import data into a table.

Note:

Dataworks's data management module can create, collect, modify data lifecycle management for the maxcompute table, modify Table Structure and data table/resource/function rights management approval, etc, for more information, see *data management overview*.

MaxCompute v2.0 supports two types of tables: internal tables and external tables.

- For internal tables, all data is stored in MaxCompute tables, and the columns in the table can be any of the *data types* supported by MaxCompute.
- For external tables, data is not stored in MaxCompute. Instead, table data can be stored in *OSS* or *OTS* . MaxCompute only records meta information of the table. You can use MaxCompute’s external table to process unstructured data on OSS or Table Store, such

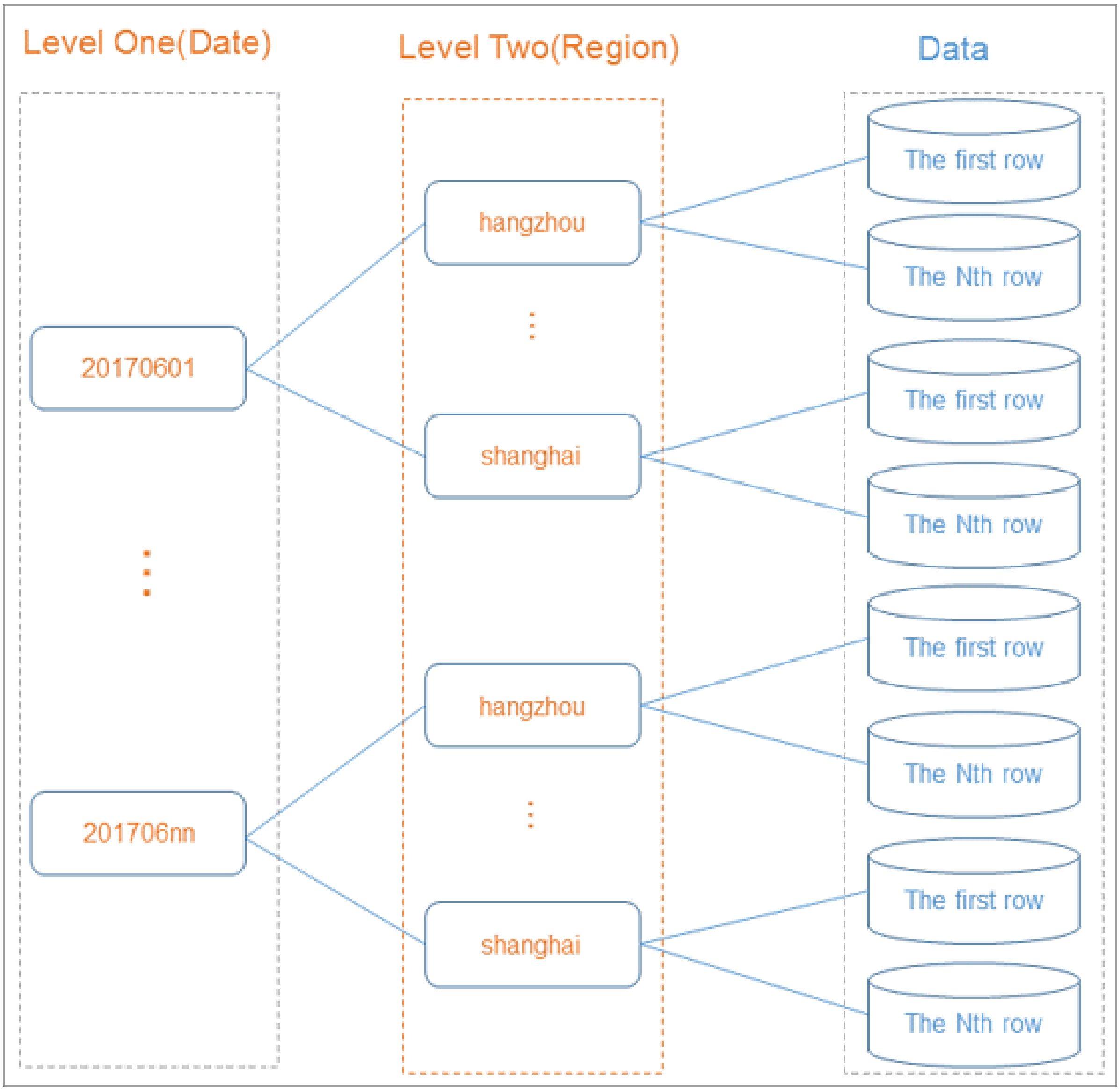
as video, audio, genetics, meteorological, and geographic information. For external table processing, see *processing unstructured data*.

3.3 Partition

To improve MaxCompute’s processing efficiency, you can specify a partition when creating a table. Specifically, several fields in the table can be specified as partition columns. A partition is comparable in terms of functionality to a directory under a file system.

In MaxCompute each value of a partition column is used as a partition. You can specify multiple fields of the table as a partition whereby they then function similarly to multi-level directories.

If the partitions to be accessed are specified when you use data, then only corresponding partitions are read and a full table scan is avoided, improving processing efficiency while reducing costs.



Partition types

MaxCompute 2.0 has expanded its support for partition types. Currently, MaxCompute supports the following partition types: TINYINT, SMALLINT, INT, BIGINT, VARCHAR, and STRING.

Note:

In MaxCompute versions earlier than 2.0, only STRING partition type is supported. Although the partition type can be specified as BIGINT, it is still handled as STRING, and only the schema of the table is indicated as a BIGINT type. An example is as follows:

```
create table parttest (a bigint) partitioned by (pt bigint);
insert into parttest partition(pt) select 1, 2 from dual;
insert into parttest partition(pt) select 1, 10 from dual;
```

```
select * from parttest where pt >= 2;
```

After the execution, the returned result is only one line, because 10 was treated as a STRING and compared with 2, meaning no result can be returned.

Restrictions

When using a partition, the following restrictions apply:

- The maximum number of partition levels for a single table is 6 levels.
- The maximum number of single table partitions is 60,000.
- The maximum number of query partitions for a query is 10,000.

For example:

```
-- create a two-level partition table with the date as the level one
partition and the region as the level two partition
create table src (key string, value bigint) partitioned by (pt string,
region string);
```

When querying, use the partition column as a filter condition in the WHERE condition filter:

```
select * from src where pt='20170601' and region='hangzhou';  -- This
example is the correct method of using WHERE conditional filter. When
MaxCompute generates a query plan, only data of the region 'hangzhou'
under the '20170601' partition is accessed.
select * from src where pt = 20170601;   -- This example is an
incorrect method of using the WHERE conditional filter. In this
 example, the effectiveness of the partition filter cannot be
guaranteed. pt is a STRING type. When the STRING type is compared with
 BIGINT type (20170601), MaxCompute converts both to DOUBLE type, and
loss of precision occurs.
```

Some SQL operations on the partitions are less efficient and may cause higher billing, for example, *using dynamic partition*.

For some MaxCompute commands, when performing operations on partitioned and non-partitioned tables, the syntax is different. For more information, see *DDL SQL* and *DML SQL*.

3.4 Data type

Basic data types

supported by MaxCompute2.0 are listed in the following table. Columns in a MaxCompute table must be any of the listed types. New types include TINYINT, SMALLINT, INT, FLOAT, VARCHAR, TIMESTAMP, and BINARY data type.

Note:

If data type such as TINYINT, SMALLINT, INT, FLOAT, VARCHAR, TIMESTAMP, or BINARY are involved when running an SQL command, the set command `set odps.sql.type.system.odps2=true;` must be added before the SQL command.

The set command and SQL command are then submitted simultaneously. If INT type is involved, and the set command is not added, the INT type is converted to BIGINT, which is 64 bits.

Type	New	Constant	Description
TINYINT	Yes	1Y，-127Y	8-bit signed integer, range -128 to 127
SMALLINT	Yes	32767S, -100S	16-bit signed integer, range -32768 to 32767
INT	Yes	1000,-15645787 (Note1)	32-bit signed integer-231 to 231 - 1
BIGINT	No	100000000000L, -1L	64-bit signed integer, range -263 + 1 to 263 - 1
FLOAT	Yes	None	32-bit binary floating point
DOUBLE	No	3.1415926 1E+7	64-bit binary floating point
DECIMAL	No	3.5BD， 99999999999.9999999BD	10-in-order exact numeric type, Plastic Part range-1036 + 1 to 1036-1, fractional portion accurate to 10-18
VARCHAR	Yes	None (Note2)	Variable-length character type, n is the length, and the range is 1 to 65535.
STRING	No	“abc”,’bcd’,”alibaba” ‘inc’ (Note3)	A single string length can be up to 8M
BINARY	Yes	None	Binary data type, a single string length can be up to 8M
DATETIME	No	DATETIME ‘2017-11-11 00:00:00’	0001-01-01 00:00:00 ~ 9999-12-31 23:59:59

Type	New	Constant	Description
			, Date type, use UTC+ 8 as the standard time system
TIMESTAMP	Yes	TIMESTAMP ‘2017-11-11 00:00:00.123456789’	It is independent of the time zone and ranges from January 1st 0000 to December 31, 9999 23.59:59.999999999 , and is accurate to nanosecond-level.
BOOLEAN	No	TRUE，FALSE	True/False, Boolean type

All data types in the preceding table can be NULL.

Note:

- NOTE 1: For INT constant, if the range of INT is exceeded, INT is converted into BIGINT; if the range of BIGINT is exceeded, it is converted into DOUBLE.

  In MaxCompute versions earlier than 2.0, all INT types in SQL script are converted to BIGINT , for example:

  ```
  create table a_bigint_table(a int);
  select cast(id as int) from mytable;
  ```

  To be compatible with earlier MaxCompute versions, MaxCompute 2.0 retains this conversion without setting odps.sql.type.system.odps2 as true, however, a warning is triggered when INT is treated as BIGINT. In this case, we recommend that you change an Int to a Bigint to avoid confusion.

- NOTE 2: VARCHAR constants can be expressed by STRING constants of implicit transformation.
- NOTE 3: STRING constants support connections, for example, `abc` `xyz` is parsed as `abcxyz`, and different parts can be written on different lines.

Complex Data Types

Complex data types that MaxCompute supports are listed in the following table.

Note:

If a complex data type is involved when you run a SQL command, the set command `set odps.sql.type.system.odps2=true;` must be added before the SQL command. The set command and SQL command are then submitted simultaneously.

Type	Definition method	Construction method
ARRAY	array< int >; array< struct< a: int, b:string >>	array(1, 2, 3); array(array(1, 2); array(3, 4))
MAP	map< string, string >; map< smallint, array< string>>	map("k1", "v1", "k2", "v2"); map (1S, array('a', 'b'), 2S, array('x ', 'y'))
STRUCT	struct< x:int, y:int>; struct< field1:bigint, field2:array< int >, field3:map< int, int>>	named_struct('x', 1, 'y', 2); named_struct('field1', 100L, ' field2', array(1, 2), 'field3', map (1, 100, 2, 200)

3.5 Lifecycle

The lifecycle of a MaxCompute table or partition is measured from the last update time. If the table or partition remains unchanged after a specified time, MaxCompute automatically recycles it. The **specified time** indicates the lifecycle.

- Lifecycle units: days, positive integers only.
- When a lifecycle is specified for a non-partition table, the lifecycle is counted from the last time the table data was modified (LastDataModifiedTime). If table data has not been changed, MaxCompute recycles the table automatically without manual operation (similar to the drop table operation).
- When a lifecycle is specified for a non-partition table, the lifecycle is counted from the last time the table data was modified (LastDataModifiedTime). If table data has not been changed, MaxCompute recycles the table automatically without manual operation (similar to the drop table operation).

Note:

Lifecycle scanning is started at a scheduled time every day, and entire partitions are scanned. If the partition remains unchanged after its lifecycle, MaxCompute automatically recycles it.

When a lifecycle is specified for a partition table, MaxCompute determines whether to recycle the partition based on its LastDataModifiedTime. Unlike non-partition tables, a partition table cannot be deleted even when all its partitions have been recycled.

- You can set the lifecycle of tables, but not of partitions. The lifecycle of a table can be specified during table creation.
- If no lifecycle is specified, the table, or partition cannot be automatically recycled by MaxCompute.

For more information on specifying or modifying lifecycles during table creation, and modifying a table's LastDataModifiedTime, see *DDL documentation*.

3.6 Resource

Resources is a concept that is unique to MaxCompute. To accomplish tasks using user-defined functions (for more information, see *UDF*), or *MapReduce*, you must use resources.

- **SQL UDF**: After writing a UDF, you must compile it as a Jar package and upload the package to MaxCompute as a resource. Then, when you run this UDF, MaxCompute automatically downloads its corresponding JAR package to obtain the written code. The JAR package is one type of MaxCompute resource.
- **MapReduce**: After writing a MapReduce program, you must compile it as a Jar package and upload the package to MaxCompute as a resource. Then, when running a MapReduce job, the MapReduce framework automatically downloads the corresponding JAR package and obtain the written code. You can upload text files and MaxCompute tables to MaxCompute as different types of resources. Then, you can read or use these resources when running UDF or MapReduce.

MaxCompute provides interfaces for you to read and use resources. For more information, see *Use Resourse Example* and *UDTF Usage* .

Note:

For more information about the resource reading capabilities of user-defined functions (*UDF*) or *MapReduce* , see *Application Restriction*.

Types of MaxCompute resources include:

- File
- Table: tables in MaxCompute

  Note:

  Currently, only BIGINT, DOUBLE, STRING, DATETIME, and BOOLEAN fields are supported in tables referenced by MapReduce.

- Jar type, which is compiled Java JAR packages
- Archive type, which is the compression type, and is determined by the resource name suffix. Supported compression types include: .zip/.tgz/.tar.gz/.tar/jar

For more information about resources, see *Add Resource* , *Drop Resource* , *List Resources* and *Describe Resource*.

3.7 Function

MaxCompute provides SQL computing capabilities. In MaxCompute SQL, you can use the *system's built-in functions* to perform common computing and counting tasks. If the built-in functions do not meet your requirements, you can use the Java programming interface provided by MaxCompute to develop user-defined functions (UDFs).

UDFs can be divided into scalar valued functions, user-defined aggregate functions (UDAFs), and user-defined tables functions (UDTFs).

After writing the code for a UDF, you must compile the code into a JAR package and upload this package to MaxCompute. Then, you can register the UDF in MaxCompute.

Note:

UDFs are used in the same way as built-in functions, in that you specify the UDF name and input relevant parameters in SQL.

For more information, see *Function introduction*.

3.8 Task

A task is the basic computing unit of MaxCompute. Computing tasks such as those involving SQL , DML and MapReduce functions are completed using tasks.

For most user-submitted tasks, such as *SQL DML statement* , *MapReduce*, etc.MaxCompute first analyzes them and then generates a task execution plan. The execution plan is composed of multiple execution stages that are dependent on each other. An execution plan consists of multiple stages with dependency links.

Currently, an execution plan can be logically viewed as a directed graph whose vertices represent the stages and whose edges represent the dependency links of the stages. MaxCompute executes each stage according to the dependencies in the graph (execution plan). A single stage comprises multiple threads, also known as workers. These workers complete the computing in this stage. Different workers in the same stage have exactly the same execution logic, but they process different data. Computational tasks are executed directly in MaxCompute instances, for example, *Status Instance* and *Kill Instance*.

For MaxCompute tasks that are not computational tasks, such as DDL statement in SQL, these tasks can only read and modify the metadata information in MaxCompute. This means that no execution plan can be analyzed and generated from the task.

Note:

Not all the requests are converted into tasks in MaxCompute, for example, the operations of *Project*, *Resource*, *UDF* and *Instance* can be completed without MaxCompute tasks.

3.9 Instance

In MaxCompute, most *tasks* are initiated in MaxCompute instances. MaxCompute instances can be in one of two phases: Running and Terminated.

The status of the running phase is ‘Running’, while the status of the Terminated phase can be ‘Success’, ‘Failed’ or ‘Canceled’. You can query or change the status using the instance ID assigned by MaxCompute. For example:

```
status <instance_id>; --View the status of a certain instance.
            kill <instance_id>; --Stop an instance and set its status
as ‘Canceled’.
            wait <instance_id>; --View the running logs of a certain
instance.
```

4 Reading guidance

For first time users

The following sections detail recommended readings for first time users, data analysts, project owner/administrators, and developers.

- *MaxCompute Summary*: Introduces MaxCompute, including its main function modules.
- *Quick Start*: Provides a step-by-step guide including how to apply for an account, how to install the client, how to create a table, how to authorize a user, how to export/import data, how to run SQL tasks, how to run UDF, and how to run Mapreduce programs.
- *Basic Introduction*: Details key terms and frequently used commands of MaxCompute.
- *Tools*: Lists all key tools used in MaxCompute

  (also, see MaxCompute *Client*).

It is recommended that you learn more about the other modules in depth, after you are familiar with the above modules.

For data analysts

If you are a data analyst, it is recommended that you read the contents of the SQL module.

- *MaxCompute SQL*: Query and analyze massive data that stored on MaxCompute. The major features are:
  - DDL statements are supported. You may manage tables and partitions by using Create, Drop, and Alter statements.
  - You can Select to select records in the table, and use the Where clause to filter return only those records that satisfy specified conditions.
  - You can associate two tables by using the Join statement.
  - You can apply Group By to some columns to aggregate them.
  - Insert overwrite/into allows you to insert results into another table.
  - You can use built-in functions and user-defined functions (UDF) to complete a variety of computations.

For developers

If you have a certain level of development experience, understanding the concept of distribution , and some data analysis may not be possible with SQL, you are advised to learn more about MaxCompute's advanced functional modules. As follows:

- *MapReduce*: Explains the MapReduce programming interface. You can use the Java API, which is provided by MapReduce, to write MapReduce program for processing data in MaxCompute.
- *Graph*: Provides a set of frameworks for iterative graph computing. Graph uses graphs for modeling. A graph consists of vertices and edges, both of which contain values. Iterations edit and evolve graphs and finally deliver the solutions.
- *Eclipse Plugin*: Facilitates users to use the Java SDK of MapReduce, UDF, and Graph.
- *Tunnel*: Facilitates users to use the Tunnel service to upload batch offline data to MaxCompute, or download batch offline data from MaxCompute.
- SDK:
  - *Java SDK*: Provides developers with Java interfaces.
  - *Python SDK*: Provides developers with Python interfaces.

Note:

MapReduce and *Graph* functions are still in public test, and if you want to use this feature, applications can be submitted through the job system. Please specify the name of your project space when applying, and we will process it within 7 working days.

For project owners/administrators

If you are a project owner or administrator, we recommend you carefully review the following chapters:

- *Security*: Explains how to grant privileges to a user, how to share resource span projects, how to set project protection, and how to grant privilege by policy, and more.
- *Billing*: Details the pricing of MaxCompute.
- Commands that only the project owner can use. For example, the **SetProject** operation in *Others* of *Common Commands*.