阿里云
MaxCompute

产品简介

文档版本：20180806

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。
斜体	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 产品概述

大数据计算服务（MaxCompute，原名ODPS）是一种快速、完全托管的GB/TB/PB级数据仓库解决方案。MaxCompute为您提供了完善的数据导入方案以及多种经典的分布式计算模型，能够更快速的解决海量数据计算问题，有效降低企业成本，并保障数据安全。

同时，DataWorks和MaxCompute关系紧密，DataWorks为MaxCompute提供了一站式的数据同步、任务开发、数据工作流开发、数据管理和数据运维等功能，详情请参见*DataWorks*。

MaxCompute主要服务于批量结构化数据的存储和计算，可以提供海量数据仓库的解决方案以及针对大数据的分析建模服务。随着社会数据收集手段的不断丰富及完善，越来越多的行业数据被积累下来。数据规模已经增长到了传统软件行业无法承载的海量数据（百GB、TB乃至PB）级别。

在分析海量数据场景下，由于单台服务器的处理能力限制，数据分析者通常采用分布式计算模式。但分布式的计算模型对数据分析人员提出了较高的要求，且不易维护。使用分布式模型，数据分析人员不仅需要了解业务需求，同时还需要熟悉底层计算模型。MaxCompute的目的是为您提供一种便捷的分析处理海量数据的手段，您可以不必关心分布式计算细节，便可达到分析大数据的目的。

说明：

MaxCompute已经在阿里巴巴集团内部得到大规模应用，例如大型互联网企业的数据仓库和BI分析、网站的日志分析、电子商务网站的交易分析、用户特征和兴趣挖掘等。

产品优势

- 大规模计算存储

  MaxCompute适用于100GB以上规模的存储及计算需求，最大可达EB级别。
- 多种计算模型

  MaxCompute支持SQL、MapReduce、Graph等计算类型及MPI迭代类算法。
- 强数据安全

  MaxCompute已稳定支撑阿里全部离线分析业务7年以上，提供多层沙箱防护及监控。
- 低成本

  与企业自建私有云相比，MaxCompute的计算存储更高效，可以降低20%-30%的采购成本。

功能概述

- 数据通道

— 支持批量、历史数据通道

*TUNNEL*是MaxCompute为您提供的数据传输服务，提供高并发的离线数据上传下载服务。支持每天TB/PB级别的数据导入导出，特别适合于全量数据或历史数据的批量导入。Tunnel为您提供Java编程接口，并且在MaxCompute的客户端工具中，有对应的命令实现本地文件与服务数据的互通。

— 实时、增量数据通道

针对实时数据上传的场景，MaxCompute提供了延迟低、使用方便的DataHub服务，特别适用于增量数据的导入。DataHub还支持多种数据传输插件，例如Logstash、Flume、Fluentd、Sqoop等，同时支持日志服务Log Service中的投递日志到MaxCompute，进而使用DataWorks进行日志分析和挖掘。

- 计算及分析任务

MaxCompute支持多种计算模型，详情如下：

— *SQL*：MaxCompute只能以表的形式存储数据，并对外提供了SQL查询功能。您可以将MaxCompute作为传统的数据库软件操作，但其却能处理TB、PB级别的海量数据。

> 说明：
>
> - MaxCompute SQL不支持事务、索引及Update/Delete等操作。
> - 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。详情请参见与其他*SQL*语法的差异。
> - 在使用方式上，MaxCompute SQL最快可以在分钟、乃至秒级别完成查询，无法在毫秒级别返回结果。
> - MaxCompute SQL的优点是学习成本低，您不需要了解复杂的分布式计算概念。如果您具备数据库操作经验，便可快速熟悉MaxCompute SQL的使用。

— *UDF*：即用户自定义函数。

MaxCompute提供了很多内建函数来满足您的计算需求，同时您还可以通过创建自定义函数来满足不同的计算需求。

— *MapReduce*：MaxCompute MapReduce是MaxCompute提供的Java MapReduce编程模型，它可以简化开发流程，更为高效。您若使用MaxCompute MapReduce，需要对分布式计算概念有基本了解，并有相对应的编程经验。MaxCompute MapReduce为您提供 Java 编程接口。

- *Graph*：MaxCompute提供的Graph功能是一套面向迭代的图计算处理框架。图计算作业使用图进行建模，图由点（Vertex）和边（Edge）组成，点和边包含权值（Value）。通过迭代对图进行编辑、演化，最终求解出结果，典型应用：*PageRank*、单源最短距离算法 、*K*-均值聚类算法 等。

- **SDK**

  SDK是MaxCompute提供给开发者的工具包，详情请参见*SDK* 介绍。

- 安全

  MaxCompute提供了功能强大的安全服务，为您的数据安全提供保护，详情请参见 安全指南。

后续步骤

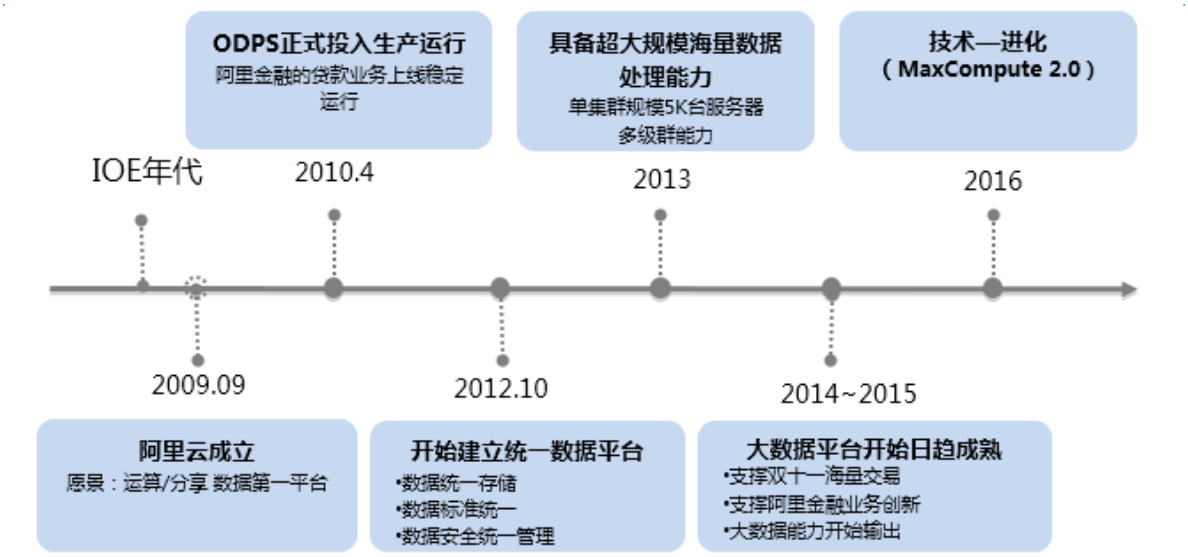
现在，您已经学习了MaxCompute的产品优势、功能特性等相关简介，您可以继续学习下一个教程。在该教程中您将了解MaxCompute的相关收费情况，详情请参见产品定价。

2 发展历程

从2009年9月阿里云成立，愿景就是做运算/分享数据的第一平台。2010年4月，伴随阿里金融的贷款业务上线，ODPS正式投入生产运行。2012 年建立统一数据平台，2013年具备超大规模海量数据处理能力，2014~2015年大数据平台开始日趋成熟，2016年MaxCompute2.0诞生，成立之初的愿景正在逐步实现。

关键性里程碑

- 2010.04 ODPS正式投入生产运行，阿里金融的贷款业务上线稳定运行。
- 2013.05 ODPS公测。
- 2013.07 ODPS正式提供商业化服务，单集群规模5K台服务器多级群能力。
- 2016.09 ODPS正式更名为MaxCompute，并推出MaxCompute2.0，实现高性能，新功能，富生态。



3 基本概念

3.1 项目空间

项目空间（Project）是MaxCompute的基本组织单元，它类似于传统数据库的Database或Schema的概念，是进行多用户隔离和访问控制的主要边界。一个用户可以同时拥有多个项目空间的权限，通过安全授权，可以在一个项目空间中访问另一个项目空间中的对象，例如表#Table#、资源#Resource#、函数#Function#和实例#Instance#。

您可以通过Use Project命令进入一个项目空间，如下所示：

```
use my_project -- 进入一个名为 my_project 的项目空间
```

运行此命令后，您会进入一个名为my_project的项目空间，您可操作该项目空间下的对象，例如表、资源、函数和实例等，不需关心操作对象所在的项目空间。Use Project是MaxCompute客户端提供的命令。在详细介绍这部分内容之前，文档会对常用的命令做简短介绍，详情请参见*MaxCompute*常用命令。

3.2 表

表是MaxCompute的数据存储单元，它在逻辑上也是由行和列组成的二维结构，每行代表一条记录，每列表示相同数据类型的一个字段，一条记录可以包含一个或多个列，各个列的名称和类型构成这张表的Schema。

MaxCompute中不同类型计算任务的操作对象（输入、输出）都是表。您可以创建表、删除表以及向表中导入数据。

说明：

DataWorks的数据管理模块可以对MaxCompute表进行新建、收藏、修改数据生命周期管理、修改表结构和数据表/资源/函数权限管理审批等操作，详情请参见数据管理概述。

MaxCompute的表格有两种类型：内部表及外部表（MaxCompute2.0版本开始支持外部表）。

- 对于内部表，所有的数据都被存储在MaxCompute中，表中的列可以是MaxCompute支持的任意一种数据类型。
- 对于外部表，MaxCompute并不真正持有数据，表格的数据可以存放在*OSS*或*OTS*中。MaxCompute仅会记录表格的Meta信息，您可以通过MaxCompute的外部表机制处

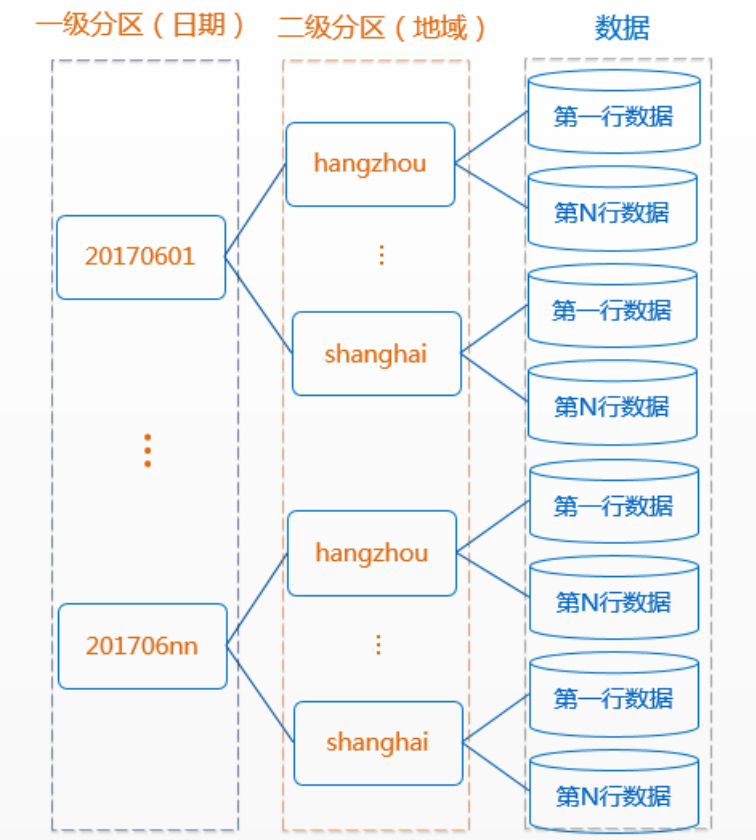
理OSS或OTS上的非结构化数据，例如视频、音频、基因、气象、地理信息等。外部表的处理请参见处理非结构化数据。

3.3 分区

分区表是指在创建表时指定分区空间，即指定表内的某几个字段作为分区列。大多数情况下，您可以将分区类比为文件系统下的目录。

MaxCompute将分区列的每个值作为一个分区（目录），您可以指定多级分区，即将表的多个字段作为表的分区，分区之间如多级目录的关系。

使用数据时，如果指定需要访问的分区名称，则只会读取相应的分区，可避免全表扫描，提高处理效率，降低费用。



分区类型

MaxCompute2.0对分区类型的支持进行了扩充，目前MaxCompute支持TINYINT、SMALLINT、INT、BIGINT、VARCHAR和STRING分区类型。

说明：

在旧版MaxCompute中，仅承诺STRING类型分区。因为历史原因，虽然可以指定分区类型为BIGINT，但是除了表的schema表示其为BIGINT外，任何其他情况实际都被处理为STRING。示例如下：

```
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;
```

执行上述语句后，返回的结果只有一行，因为10被当作字符串和2比较，所以没能返回。

分区使用限制

分区有以下使用限制：

- 单表分区层级最多6级。
- 单表分区数最多允许60000个分区。
- 一次查询最多查询分区数为10000个分区。

示例如下：

```
--创建一个二级分区表，以日期为一级分区，地域为二级分区
create table src (key string, value bigint) partitioned by (pt string,
region string);
```

查询时，Where条件过滤中使用分区列作为过滤条件：

```
select * from src where pt='20170601' and region='hangzhou'; --正确使用方式。MaxCompute在生成查询计划时只会将'20170601'分区下region为'hangzhou'二级分区的数据纳入输入中。
select * from src where pt = 20170601; -- 错误的使用方式。在这样的使用方式下，MaxCompute并不能保障分区过滤机制的有效性。pt是String类型，当String类型与Bigint（20170601）比较时，MaxCompute会将二者转换为Double类型，此时有可能会有精度损失。
```

部分对分区操作的SQL的运行效率则较低，会给您带来较高的计费，例如使用动态分区。

对于部分MaxCompute的操作命令，处理分区表和非分区表时的语法有差别，详情请参见*DDL*语句和使用动态分区 。

3.4 数据类型

基本数据类型

MaxCompute2.0支持的基本数据类型如下表所示，新增类型有TINYINT、SMALLINT、 INT、FLOAT、VARCHAR、TIMESTAMP和BINARY，MaxCompute表中的列必须是下列描述的任意一种类型，详情如下：

说明：

SQL（Create、select、insert等操作）中涉及到新数据类型（TINYINT、SMALLINT、 INT、FLOAT、VARCHAR、TIMESTAMP BINARY），需在SQL语句前加语句`set odps.sql.type.system.odps2=true;`，执行时set语句和SQL语句一起提交执行。

涉及INT类型，加上述set语句的时候是32位，不加的时候会被转换成BIGINT，是64位。

MR类型任务目前暂时不支持操作新数据类型。

类型	是否新增	常量定义	描述
TINYINT	是	1Y，-127Y	8 位有符号整形，范围 -128 到 127
SMALLINT	是	32767S， -100S	16 位有符号整形， 范围 -32768 到 32767
INT	是	1000，-15645787（注释1）	32位有符号整形，范围-231到231 - 1
BIGINT	否	100000000000L， -1L	64位有符号整形, 范围-263 + 1到263 - 1
FLOAT	是	无	32位二进制浮点型
DOUBLE	否	3.1415926 1E+7	64位二进制浮点型
DECIMAL	否	3.5BD， 99999999999.9999999BD	10 进制精确数字类型，整形部分范围-1036+1到1036-1, 小数部分精确到 10-18
VARCHAR	是	无（注释2）	变长字符类型，n为长度，取值范围 1 到 65535
STRING	否	“abc”，’bcd’，”alibaba” ‘inc’（注释3）	字符串类型，目前长度限制为 8M

类型	是否新增	常量定义	描述
BINARY	是	无	二进制数据类型，目前长度限制为 8M
DATETIME	否	DATETIME ‘2017-11-11 00:00:00’	日期时间类型，范围从0000年1月1日到9999年12月31日，精确到毫秒
TIMESTAMP	是	TIMESTAMP ‘2017-11-11 00:00:00.123456789’	与时区无关的时间戳类型，范围从0000年1月1日到9999年12月31日23.59:59.999999999, 精确到纳秒
BOOLEAN	否	TRUE，FALSE	boolean 类型, 取值 TRUE 或 FALSE

上述的各种数据类型均可为NULL。

 说明：

- 注释1：对于INT常量，如果超过INT取值范围，会转为BIGINT。如果超过BIGINT取值范围，会转为DOUBLE。

  在旧版MaxCompute中，因为历史原因，SQL脚本中的所有INT类型都被转换为BIGINT，如下所示：

  ```
  create table a_bigint_table(a int); -- 这里的int实际当作bigint处理
  select cast(id as int) from mytable; -- 这里的int实际当作bigint处理
  ```

  为了与MaxCompute原有模式兼容，MaxCompute2.0在未设定odps.sql.type.system.odps2为true的情况下，仍保留此转换，但会报告一个警告，提示INT被当作BIGINT处理了，如果您的脚本有此种情况，建议全部改写为BIGINT，避免混淆。
- 注释2：VARCHAR类型常量可通过STRING常量的隐式转换表示。
- 注释3：STRING常量支持连接，例如`abc xyz`会解析为`abcxyz`，不同部分可以写在不同行上。

复杂数据类型

MaxCompute2.0支持的复杂类型如下表所示。

说明：

SQL（ create、select、insert等操作）中涉及到这几个复杂数据类型，需在SQL语句前加语句`set odps.sql.type.system.odps2=true;`，执行时set语句和SQL语句一起提交执行。

类型	定义方法	构造方法
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 生命周期

MaxCompute表的生命周期（LIFECYCLE），指表（分区）数据从最后一次更新的时间算起，在经过指定的时间后没有变动，则此表（分区）将被MaxCompute自动回收。这个**指定的时间**就是生命周期。

- 生命授权单位：days（天），只接受正整数。
- 非分区表若指定生命周期，自最后一次数据被修改的时间（LastDataModifiedTime）开始计算，经过days天后数据仍未被改动，则此表无需您干预，将会被MaxCompute自动回收（类似drop table操作）。
- 分区表若指定生命周期，则根据各个分区的LastDataModifiedTime判断该分区是否该被回收。不同于非分区表，分区表的最后一个分区被回收后，该表不会被删除。

说明：

生命周期回收都是每天定时启动，扫描全量分区，扫到的时刻，Last modify time超过lifecycle指定的时间才回收。

假设某个分区表生命周期为1天，该分区数据最后一次被修改的时间是17号15点零分，如果18号的回收扫描在15点前扫到这个表（不到一天），就不会回收上述分区。19号回收扫描时才发现这个表的这个分区Last modify time超过lifecycle指定的时间，这时上述分区会被回收。

- 生命周期只能设定到表级别，不能再分区级设置生命周期。创建表时即可指定生命周期。
- 表若不指定生命周期，则表（分区）不会根据生命周期规则被MaxCompute自动回收。

关于建表时怎么指定/修改表生命周期、修改表LastDataModifiedTime等操作请参见*DDL*文档。

3.6 资源

资源（Resource）是MaxCompute的特有概念，如果您想使用MaxCompute的自定义函数*#UDF#*或*MapReduce*功能需要依赖资源来完成，如下所示：

- **SQL UDF**：您编写UDF后，需要将编译好的Jar包以资源的形式上传到MaxCompute。运行此UDF时，MaxCompute会自动下载这个Jar包，获取您的代码来运行UDF，无需您干预。上传Jar包的过程就是在MaxCompute上创建资源的过程，这个Jar包是MaxCompute资源的一种。
- **MapReduce**：您编写MapReduce程序后，将编译好的Jar包作为一种资源上传到MaxCompute。运行MapReduce作业时，MapReduce框架会自动下载这个Jar资源，获取您的代码。您同样可以将文本文件以及MaxCompute中的表作为不同类型的资源上传到MaxCompute，您可以在UDF及MapReduce的运行过程中读取、使用这些资源。

MaxCompute提供了读取、使用资源的接口。详情请参见资源使用示例及*UDTF* 使用说明 。

> 说明：
> MaxCompute的自定义函数*#UDF#*或*MapReduce*对资源的读取有一定的限制，详情请参见*MR*限制汇总。

MaxCompute资源包括以下几种类型：

- File类型。
- Table类型：MaxCompute中的表。

  > 说明：
  > MapReduce引用的table类型资源中，table表字段类型目前只支持BIGINT、DOUBLE、STRING、DATETIME、BOOLEAN，其他类型暂未支持。
- Jar类型：编译好的Java Jar包。
- Archive类型：通过资源名称中的后缀识别压缩类型，支持的压缩文件类型包括.zip/.tgz/.tar.gz/.tar/jar。

资源的相关操作请参见创建资源 、删除资源 、查看资源清单和查看资源信息。

3.7 函数

MaxCompute为您提供了SQL计算功能，您可以在MaxCompute SQL中使用系统的内建函数完成一定的计算和计数功能。但当内建函数无法满足要求时，您可以使用MaxCompute提供的Java编程接口开发自定义函数（User Defined Function，以下简称UDF）。

自定义函数*#UDF#*可以进一步分为标量值函数（UDF），自定义聚合函数（UDAF）和自定义表值函数（UDTF）三种类型。

您在开发完成UDF代码后，需要将代码编译成Jar包，并将此Jar包以Jar资源的形式上传到MaxCompute，最后在MaxCompute中注册此UDF。

说明：

使用UDF时，只需在SQL中指明UDF的函数名及输入参数即可，使用方式与MaxCompute提供的内建函数相同。

函数的相关操作请参见创建函数 ， 删除函数及查看函数清单。

3.8 任务

任务（Task）是MaxCompute的基本计算单元，SQL及MapReduce功能都是通过任务完成的。

对于您提交的大多数任务，特别是计算型任务，例如*SQL DML*语句、*MapReduce*等，MaxCompute会对其进行解析，得出任务的执行计划。执行计划由具有依赖关系的多个执行阶段（Stage）构成。

目前，执行计划逻辑上可以被看做一个有向图，图中的点是执行阶段，各个执行阶段的依赖关系是图的边。MaxCompute会依照图（执行计划）中的依赖关系执行各个阶段。在同一个执行阶段内，会有多个进程，也称之为Worker，共同完成该执行阶段的计算工作。同一个执行阶段的不同Worker只是处理的数据不同，执行逻辑完全相同。计算型任务在执行时，会被实例化，您可以操作这个实例（Instance）的信息，例如获取实例状态*#Status Instance#*、终止实例运行*#Kill Instance#*等。

另一方面，部分MaxCompute任务并不是计算型的任务，例如SQL中的DDL语句，这些任务本质上仅需要读取、修改MaxCompute中的元数据信息。因此，这些任务无法被解析出执行计划。

说明：

在MaxCompute中，并不是所有的请求都会被转化为任务（Task），例如项目空间*#Project#*、资源*#Resource#*、自定义函数*#UDF#*及实例*(Instance)*的操作均不需要通过MaxCompute的任务来完成。

3.9 任务实例

在MaxCompute中，部分任务*#Task#*在执行时会被实例化，以MaxCompute实例（下文简称为实例或Instance）的形式存在。实例会经历运行（Running）和结束（Terminated）两个阶段。

运行阶段的状态为Running（运行中），而结束阶段则会有Success（成功）、Failed（失败）和Canceled（被取消）三种状态。您可以根据运行任务时MaxCompute给出的实例ID进行查询、改变任务的状态等操作，示例如下：

```
status ; --查看某实例的状态
kill ; --停止某实例，将其状态设置为
Canceledwait ; --查看某实例的运行日志
```

4 导读

如果您是MaxCompute初学者

如果您是初学者，建议您从如下模块开始读起：

- 产品简介： MaxCompute产品的总体介绍以及包含的主要功能。通过阅读该章节，您会对MaxCompute有一个总体的认识。
- 快速开始：通过示例，指导您如何进行申请账号、安装客户端、创建表、授权、导入导出数据、运行SQL任务、运行UDF/Mapreduce程序等操作。
- 基本介绍：MaxCompute的基本概念及常用命令介绍。您可以进一步熟悉如何操作MaxCompute。
- 工具：在分析数据之前，您需要掌握MaxCompute常用工具的下载、配置以及使用方法。

  我们为您提供工具，您可以通过此工具对MaxCompute进行操作。

建议您熟悉以上的模块后，再有针对性地对其他模块进行深入学习。

如果您是数据分析师

如果您是数据分析师，建议您熟读SQL模块的内容。

- *SQL*：您可以查询并分析存储在MaxCompute上的大规模数据。包含的主要功能如下：
  - 支持DDL语句，您可以通过Create、Drop和Alter对表和分区进行管理。
  - 您可以通过Select选择表中的某几条记录，通过Where语句查看满足条件的记录，实现过滤功能。
  - 您可以通过等值连接Join实现两张表的关联。
  - 您可以通过对某些列Group By，实现聚合操作。
  - 您可以通过Insert overwrite/into把结果记录插入到另一张表中。
  - 您可以通过内置函数和自定义函数（UDF）来实现一系列的计算。

如果您拥有一定开发经验

如果您拥有一定的开发经验，了解分布式概念，并且某些数据分析可能无法用SQL来实现，此时推荐您学习MaxCompute更高级的功能模块。如下所示：

- *MapReduce*：MaxCompute提供的Java MapReduce编程模型。您可以使用MapReduce提供的接口（Java API）编写MapReduce程序，处理MaxCompute中的数据。

- *Graph*：一套面向迭代的图计算处理框架。使用图进行建模，图由点（Vertex）和边（Edge）组成，点和边包含权值（Value）。通过迭代对图进行编辑、演化，最终得出结果。
- *Eclipse Plugin*：方便您使用MapReduce、UDF以及Graph的Java SDK进行开发工作。
- *Tunnel*：您可以使用Tunnel服务向MaxCompute批量上传离线数据或者从MaxCompute下载离线数据。
- SDK：
  - *Java SDK*：向开发者提供Java接口。
  - *Python SDK*：向开发者提供Python接口。

> **说明：**
> 目前*MapReduce*以及*Graph*功能仍处于公测中，若您想使用这部分功能，可以通过工单系统提交申请。申请时请指明您的项目空间名称，我们会在7个工作日内处理。

如果您是项目Owner或者管理员

如果您是一个项目空间的Owner或者管理员，您需要熟知以下模块：

- 安全指南：您可以通过阅读该章节，了解如何进行给用户授权、跨项目空间的资源共享、设置项目空间的数据保护功能、policy授权等操作。
- *MaxCompute*收费指南：介绍MaxCompute的收费模式。
- 部分只有项目空间Owner才能使用的命令，例如常用命令中其他操作的**SetProject**操作。