

Alibaba Cloud

MaxCompute Release Note

Document Version: 20211213

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 <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1. Announcements	05
2. Document updates	21
2.1. Updates in 2021	21
2.2. Updates in year 2020	31
2.3. Updates in year 2019	40

1. Announcements

This topic describes the updates to MaxCompute in reverse chronological order.

October 15, 2021 (UTC+08:00): Notice on the maintenance of MaxCompute Management

MaxCompute Management is maintained from 22:00 to 24:00 on October 15, 2021 (UTC+08:00). The features provided by MaxCompute Management, such as quota management and job maintenance, becomes unavailable during this period. You can use MaxCompute Management after the maintenance is completed. If you encounter urgent issues, [submit a ticket](#) to contact the MaxCompute technical support team for assistance.

October 15, 2021 (UTC+08:00): The backup and restoration feature is released for commercial use

As of October 15, 2021, you are charged for data that is backed up on MaxCompute on a pay-as-you-go basis.

Take note of the following billing rules:

- MaxCompute projects automatically back up data after data changes and generate a data version for each change. By default, all data versions are retained for one day free of charge.
- A project administrator can specify a longer retention period for backup data. Backup data that is retained for more than one day is charged based on the volume of the data. The unit price for backup data is 0.0006 USD/GB/day.

For more information about the backup and restoration feature, see [Backup and restoration](#). For more information about storage pricing, see [Storage pricing \(pay-as-you-go\)](#).

September 10, 2021 (UTC+08:00): The UPDATE, DELETE, and MERGE INTO statements are released for commercial use

As of September 10, 2021, the public preview of the **UPDATE**, **DELETE**, and **MERGE INTO** statements of MaxCompute SQL ends. After the public review ends, you are charged when you use these statements. Pay-as-you-go jobs that already use the UPDATE, DELETE, and MERGE INTO statements will incur computing fees as of that date. If you did not use the UPDATE, DELETE, or MERGE INTO statement due to concerns about service stability in the public preview phase, you can now use these statements without worrying about the stability. MaxCompute provides the same guarantees for the availability and stability of the UPDATE, DELETE, and MERGE INTO statements as it provides MaxCompute SQL.

Take note of the following billing rules:

- If the billing method of your MaxCompute project is subscription, you are not separately charged for data computations.
- If the billing method of your MaxCompute project is pay-as-you-go, you are charged for data computations. The computing fee is calculated by using the following formula: Computing fee of a single SQL statement = Amount of input data × SQL complexity × Unit price (USD 0.0438 per GB). The SQL complexity is measured based on the number of SQL keywords. Number of SQL keywords = Number of JOIN clauses + Number of GROUP BY clauses + Number of ORDER BY clauses + Number of DISTINCT clauses + Number of window functions + MAX(Number of INSERT statements|Number of UPDATE statements|Number of DELETE statements - 1, 1).

To prevent excessive costs, you can configure alerts to control costs. For more information about

monitoring and alerting, see [消费监控告警Consumption control](#).

Take note of the following points:

- If the billing method of a MaxCompute project is pay-as-you-go, MaxCompute calculates fees by using the following formula: Amount of input data scanned by using the UPDATE, DELETE, and MERGE INTO statements × SQL complexity × Unit price. When you execute an UPDATE statement or a DELETE statement on a transactional table, a Delta file is generated. If a large number of Delta files are generated, the data expands and the read performance is affected. In addition, the amount of data scanned increases gradually, which increases costs. A MaxCompute project that uses the subscription billing method occupies more resources and decreases the efficiency of subsequent job execution. You can run the compact command to merge Delta files. For more information about the syntax, see [ALTER TABLE COMPACT](#).
- MaxCompute can execute multiple `DELETE` and `UPDATE` statements at a time. Each statement uses resources and incurs fees. We recommend that you delete or update a batch of data at a time. For more information, see [UPDATE and DELETE](#). Example:

```
-- Recommended: Use the association method to update or delete all the required data at the same time.
update table1 set col1=(select value1 from table2 where table1.id = table2.id and table1.region = table2.r
egion);
-- Not recommended: Perform related operations on the data row by row.
update table1 set col1=1 where id='2021063001'and region='beijing';
update table1 set col1=2 where id='2021063002'and region='beijing';
.....
```

April 20, 2021 (UTC+08:00): The ON clause of OUTER JOIN does not filter data for new MaxCompute projects

Before the update, the filter condition specified for a single table in the ON clause filters data before a JOIN operation is performed, regardless of whether the JOIN operation is OUTER JOIN. For example, the execution results of Script 1, Script 2, and Script 3 in the following code are consistent.

```
-- Construct data.
CREATE TABLE t1 AS SELECT * FROM VALUES (1, 20180101),(2, 20180101),(2, 20180102) t (key, ds);
CREATE TABLE t2 AS SELECT * FROM VALUES (1, 20180101),(3, 20180101),(2, 20180102) t (key, ds);
-- Script 1: The filter condition is in the ON clause.
SELECT t1.*,t2.*
FROM t1
LEFT OUTER JOIN t2
ON t1.key = t2.key and t1.ds='20180101';
-- Script 2: The filter condition is in the subquery.
SELECT a.*,t2.*
FROM (SELECT * FROM t1 WHERE t1.ds='20180101') a
LEFT OUTER JOIN t2
ON a.key = t2.key;
-- Script 3: The filter condition is in the WHERE clause.
SELECT t1.*,t2.*
FROM t1
LEFT OUTER JOIN t2
ON t1.key = t2.key
WHERE t1.ds='20180101';
```

This is not a standard behavior. If values that are not 20180101 are included in the t1.ds field in Script 1, LEFT OUTER JOIN retains all the values of the left table, performs a left join on the left table t1 and the right table t2, and returns the data in the matched rows or returns NULL for rows in which no data is matched. This is the standard processing behavior. However, in Script 2 and Script 3, values that are not 20180101 in the t1.ds field are filtered out and are not displayed in the output. The execution results of Script 2 and Script 3 are inconsistent with the execution result of Script 1.

The default processing behavior in MaxCompute is changed to support the standard processing behavior. If you want to filter data from table t1, use the processing behavior of Script 2 or Script 3.

For MaxCompute projects that are created on or after April 20, 2021, the filter condition in the ON clause does not filter data before the OUTER JOIN operation is performed. For MaxCompute projects that are created before April 20, 2021, the original processing behavior is retained to prevent the impact of the update on existing jobs. For existing MaxCompute projects, we recommend that you use the new processing method. Setting method:

- Session level: Add the `set odps.task.sql.outerjoin.ppd=true;` command before an SQL statement and submit the SQL statement with the command.
- Project level: [Submit a ticket](#) to request MaxCompute technical support engineers to change the value of `odps.task.sql.outerjoin.ppd` to True for your project.

The following example helps you determine which processing behavior is used in your MaxCompute project.

```
-- Construct data based on the preceding code.
SELECT t1.*, t2.*
FROM t1 FULL OUTER JOIN t2
ON t1.key = t2.key and t1.ds='20180101' and t2.ds='20180101';
-- The following results are returned if the original processing behavior is used.
| key  | ds    | key2 | ds2  |
+-----+-----+-----+-----+
| 2    | 20180101 | NULL | NULL |
| 1    | 20180101 | 1    | 20180101 |
| NULL | NULL    | 3    | 20180101 |
-- The following results are returned if the new processing behavior is used.
| key  | ds    | key2 | ds2  |
+-----+-----+-----+-----+
| NULL | NULL    | 2    | 20180102 |
| 2    | 20180101 | NULL | NULL |
| 2    | 20180102 | NULL | NULL |
| 1    | 20180101 | 1    | 20180101 |
| NULL | NULL    | 3    | 20180101 |
```

April 20, 2021 (UTC+08:00): The GET_JSON_OBJECT function used in new projects retains the original strings instead of using escape characters to return JSON-formatted results

Before the update, the GET_JSON_OBJECT function returns JSON-formatted results by using escape characters. In this case, reserved characters in the results, such as line feeds (\n) and quotation marks ("), are presented as `'\n'` or `'\"'`. These formats are not standard.

After the update, the GET_JSON_OBJECT function retains the original strings. This ensures standard returned results and higher data processing performance. After the update, the returned results are still JSON strings. However, you no longer need to use functions such as REGEXP_REPLACE or REPLACE to replace backslashes (\).

As of April 20, 2021, the GET_JSON_OBJECT function automatically retains the original strings for the returned results. For MaxCompute projects created before April 20, 2021, the GET_JSON_OBJECT function automatically returns JSON-formatted reserved characters by using escape characters. This prevents the impact on existing jobs.

You can run the following command to determine the method that the GET_JSON_OBJECT function uses to return results in a MaxCompute project:

```
select get_json_object('{\"a\": \"\\\"1\\\"\"}', '$.a');
--Return JSON-formatted reserved characters by using escape characters.
[\"1\"]
--Return the original strings.
[\"1\"]
```

We recommend that you use the new method. You can use this method at the following levels:

- Session level: Add the `set odps.sql.udf.getjsonobj.new=true;` command before an SQL statement and submit the SQL statement with the command.
- Project level: [Submit a ticket](#) to request the MaxCompute technical support personnel to change the method that the GET_JSON_OBJECT function uses to return results in your MaxCompute project.

For more information about the GET_JSON_OBJECT function, see [String functions](#).

December 14, 2020 (UTC+08:00): Check on input parameters of some date functions in MaxCompute is changed

MaxCompute strictly checks the format of input parameters of some date functions to prevent inaccurate results or other issues caused by invalid input. This change applies to all regions from December 14, 2020 to January 30, 2021 at the International site (alibabacloud.com) where MaxCompute is available. You must evaluate the current usage of related functions in your project and prepare for the update in advance. The update involves the following functions:

- **CAST(expr AS DATE)**
 - Before the update: The CAST(expr AS DATE) function does not strictly check input strings, and characters that are not of the DATE type may be used as the input parameters. For example, execute the following query statement:

```
select cast(a as date) from values ('2020-1x-19') t(a);
```

The following information is returned:

```
+-----+
| a      |
+-----+
| 2020-01-19 |
+-----+
```


- After the update: The CAST(expr AS DATE) function strictly checks the input strings. Only the input strings whose year, month, and day fields are all represented by digits can pass the check. If one of the three fields in an input string contains letters or spaces, the CAST(expr AS DATE) function considers the input string invalid and returns NULL. For example, execute the following query statement:

```
select cast(a as date) from values ('2020-1x-19') t(a);
```

The following information is returned:

```
+-----+
| a      |
+-----+
| NULL   |
+-----+
```

- **TO_DATE**

- Before the update: The TO_DATE function does not strictly check the input strings, and characters that are not of the DATE type may be used as the input parameters. For example, execute the following query statement:

```
select to_date(a) from values ('2020-1x-19') t(a);
```

The following information is returned:

```
+-----+
| _c0    |
+-----+
| 2020-01-19 |
+-----+
```

- After the update: The TO_DATE function strictly checks input strings. Only the input strings whose year, month, and day fields are all represented by digits can pass the check. If one of the three fields in an input string contains letters or spaces, the TO_DATE function considers the input string invalid and returns NULL. For example, execute the following query statement:

```
select to_date(a) from values ('2020-1x-19') t(a);
```

The following information is returned:

```
+-----+
| _c0    |
+-----+
| NULL    |
+-----+
```

- **CAST(expr AS TIMESTAMP)**

- Before the update: The CAST(expr AS TIMESTAMP) function does not strictly check input strings, and characters that are not of the DATE type may be used as the input parameters. For example, execute the following query statement:

```
select cast(a as timestamp) from values ('2020-11-19 16:3x:00.001') t(a);
```

The following information is returned:

```
+-----+
| a      |
+-----+
| 2020-11-19 16:03:00.1 |
+-----+
```

- After the update: The CAST(expr AS TIMESTAMP) function strictly checks the input strings. Only the input strings whose year, month, and day fields are all represented by digits can pass the check. If one of the three fields in an input string contains letters or spaces, the CAST(expr AS TIMESTAMP) function considers the input string invalid and returns NULL. For example, execute the following query statement:

```
select cast(a as timestamp) from values ('2020-11-19 16:3x:00.001') t(a);
```

The following information is returned:

```
+-----+
| a      |
+-----+
| NULL   |
+-----+
```

October 13, 2020 (UTC+08:00): The SQL engine is updated for better compatibility


The following table describes the schedule for updating the SQL engine. If a change occurs, the new schedule prevails.

Sequence	Region	Date
1	India (Mumbai), Indonesia (Jakarta), and UK (London)	October 13, 2020
2	US (Virginia), UAE (Dubai), China North 2 Ali Gov, and China East 2 Finance	October 15, 2020
3	Japan (Tokyo), Australia (Sydney), US (Silicon Valley), and Malaysia (Kuala Lumpur)	October 20, 2020
4	Singapore (Singapore), China (Hong Kong), and Germany (Frankfurt)	October 22, 2020

The `URL_DECODE` and `CONV` functions in MaxCompute SQL are updated. This section describes the update details.

- `URL_DECODE`

- Before the update: If two parameters are specified for the `URL_DECODE` function, the function ignores the second parameter and decodes the value of the first parameter in UTF-8. For example, if you specify `URL_DECODE(url, "gbk")` in code, `URL_DECODE(url)` is executed.
- After the update: If two parameters are specified for the `URL_DECODE` function, the function first performs decoding based on the percent sign (%). Then, the function performs decoding based on the encoding format of the second parameter.

 **Note** The command-line interface (CLI) of the Windows operating system has the following issue: If you use the MaxCompute client (odpscmd) to run commands in the Windows operating system, the GBK decoding result of the `URL_DECODE` function may be decoded in another format.

- **CONV**
 - In a project that uses the Hive-compatible data type edition, the `CONV` function returns 0 both before and after the update if the input parameters are invalid.
 - In a project that uses the MaxCompute V1.0 or MaxCompute V2.0 data type edition:
 - Before the update: The `CONV` function returns garbled characters if the input parameters are invalid.
 - After the update: The `CONV` function returns NULL if the input parameters are invalid.
- For example, if you specify `CONV("00e04c9d034a", 2, 10)` in code, NULL is returned.

July 24, 2020 (UTC+08:00): MaxCompute provides more built-in aggregate functions

The following built-in aggregate functions are added to MaxCompute:


- `APPROX_DISTINCT(value)` : returns the approximate number of distinct input values.
- `ANY_VALUE(value)` : returns a non-deterministic value from the specified value range.
- `ARG_MAX(valueToMaximize, valueToReturn)` : finds the row where the maximum value of `valueToMaximize` resides and then returns the value of `valueToReturn` in the row.
- `ARG_MIN(valueToMinimize, valueToReturn)` : finds the row where the minimum value of `valueToMinimize` resides and then returns the value of `valueToReturn` in the row.

For more information about the built-in aggregate functions, see [Aggregate functions](#).

July 29, 2020 (UTC+08:00): The default data type edition for a new project is changed from V1.0 to V2.0

If you create a MaxCompute project in the DataWorks console, the default data type edition of the project is changed from the MaxCompute V1.0 data type edition to the MaxCompute V2.0 data type edition. This change applies to all regions from July 29, 2020 to August 6, 2020 at the International site (alibabacloud.com) where MaxCompute is available. If you have created projects by using your Alibaba Cloud account, make sure that the data type edition you select for a new project is the same as that of the existing projects. Incompatibility issues may occur if projects of different data type editions interact with each other.


MaxCompute provides the following data type editions: MaxCompute V1.0 data type edition, MaxCompute V2.0 data type edition, and Hive-compatible data type edition. These editions are different in definitions and usage. MaxCompute provides three attributes to configure data type editions. You can configure these attributes to enable an edition. For more information, see [Data type editions](#).

 **Note** This feature has no impact on the data type editions of existing MaxCompute projects. You can change the data type editions of existing MaxCompute projects. For more information, see [Change the data type edition of a project](#).

June 29, 2020 (UTC+08:00): Users can choose data type editions for new projects

This feature applies to all regions from June 29, 2020 to July 15, 2020 at the International site (alibabacloud.com) where MaxCompute is available. After the feature is available, you must select initial data type editions for new MaxCompute projects.

MaxCompute provides the following data type editions: MaxCompute V1.0 data type edition, MaxCompute V2.0 data type edition, and Hive-compatible data type edition. These editions are different in definitions and usage. MaxCompute provides three attributes to configure data type editions. You can configure these attributes to enable an edition. For more information, see [Data type editions](#).

 **Note** This feature has no impact on the data type editions of existing MaxCompute projects. You can change the data type editions of existing MaxCompute projects. For more information, see [Change the data type edition of a project](#).

March 15, 2020 (UTC+08:00): The storage price of MaxCompute is reduced

As of March 15, 2020, the storage price of MaxCompute is reduced. For more information, see [Storage pricing \(pay-as-you-go\)](#). The price is reduced based on the following rules:

- The number of pricing tiers is reduced from five to three to simplify storage billing.
- The unit price of each new tier is reduced to lower the overall storage price.

The following table describes the tiered pricing method that was used before March 15, 2020.

Volume of stored data	Tiered unit price	Fixed price
0 < Data volume ≤ 1 GB	N/A	USD 0.00 per day
1 GB < Data volume ≤ 100 GB	USD 0.0028 per GB per day	N/A
100 GB < Data volume ≤ 1 TB	USD 0.0014 per GB per day	N/A
1 TB < Data volume ≤ 10 TB	USD 0.0013 per GB per day	N/A
10 TB < Data volume ≤ 100 TB	USD 0.0011 per GB per day	N/A
Data volume > 100 TB	USD 0.0009 per GB per day	N/A

The following table describes the tiered pricing method that is used since March 15, 2020.

Volume of stored data	Tiered unit price	Fixed price
0 < Data volume ≤ 1 GB	N/A	USD 0.00 per day
1 GB < Data volume ≤ 10 TB	USD 0.0011 per GB per day	N/A
10 TB < Data volume ≤ 100 TB	USD 0.0009 per GB per day	N/A
Data volume > 100 TB	USD 0.0006 per GB per day	N/A

The storage billing method remains unchanged. For more information, see [Storage pricing \(pay-as-you-go\)](#).

- You are charged for daily stored data, including tables and resources, in MaxCompute by tier based on the data volume.
- MaxCompute records the volume of data that is stored in each project on an hourly basis and then calculates the average volume of stored data for each day. The storage fee is equal to the average volume of stored data multiplied by the unit price. MaxCompute calculates the average volume of stored data in each project in a day. Therefore, storing more data in a specific project means a lower storage fee.

For example, the daily average data volume of a project is 1 PB, and the following formula is used to calculate the daily fee based on the original tiered pricing method:

```
(100 - 1) GB × USD 0.0028 per GB per day
+ (1024 - 100) GB × USD 0.0014 per GB per day
+ (10240 - 1024) GB × USD 0.0013 per GB per day
+ (102400 - 10240) GB × USD 0.0011 per GB per day
+ (10240 × 10240 - 102400) GB × USD 0.0009 per GB per day
= USD 966.486 per day
```

The following formula is used to calculate the daily fee based on the new tiered pricing method:

```
(10240 - 1) GB × USD 0.0011 per GB per day
+ (102400 - 10240) GB × USD 0.0009 per GB per day
+ (10240 × 10240 - 102400) GB × USD 0.0006 per GB per day
= USD 661.9125 per day
```

February 24, 2020 (UTC+08:00): The SQL engine is updated for better compatibility

The following table describes the schedule for updating the SQL engine. If any change occurs, the new schedule prevails.

Sequence	Region	Date
1	Indonesia (Jakarta), UK (London), and India (Mumbai)	February 24, 2020

Sequence	Region	Date
2	UAE (Dubai), US (Virginia), China North 2 Ali Gov, and China (Hong Kong)	February 26, 2020
3	Malaysia (Kuala Lumpur), Japan (Tokyo), and Germany (Frankfurt)	March 2, 2020
4	US (Silicon Valley), Singapore (Singapore), and Australia (Sydney)	March 4, 2020

- Changes to the execution rule of the `GET_IDCARD_AGE` function
 - If the difference between the current year and the year of birth is greater than 100, NULL is returned, based on the original execution rule of the `GET_IDCARD_AGE` function. After the new rule is applied, the difference between the current year and the year of birth is returned even if the difference exceeds 100. For example, the execution result of `get_idcard_age('110101190001011009')` is NULL before the change and 120 after the change.
 - If you want to apply the original execution rule to a query statement after the change, you must find the query statement and modify this statement based on your business requirements. For example, you can add the IF function or CASE WHEN expression for processing the return result of the `GET_IDCARD_AGE` function to the query statement.

Query statement before the change	Query statement after the change
<code>GET_IDCARD_AGE(idcardno)</code>	<code>if(GET_IDCARD_AGE(idcardno) > 100, NULL, GET_IDCARD_AGE(idcardno))</code>
<code>GET_IDCARD_AGE(idcardno)</code>	<code>CASE WHEN GET_IDCARD_AGE(idcardno) > 100 THEN NULL ELSE GET_IDCARD_AGE(idcardno) END</code>

- Changes to the execution rule of the `CONCAT_WS` function
 - Before the change, if the `CONCAT_WS` function that is used in a query does not support Hive and has three or more parameters, including at least one parameter of the ARRAY type, the array items do not appear in the final result. For example, the expected result of the `concat_ws(',', array('a'), array('b', 'c'))` function is "a,b,c", but the actual result is ",,,".
 - After the change, the parameters of the STRING and ARRAY types can coexist in the `CONCAT_WS` function without Hive support enabled. For example, the return result of the `concat_ws(',', array('a'), array('b', 'c'))` function is "a,b,c".
- Changes to the execution rule of the `Like%%` function when the input value is an empty string.

Before the change, if the input value for the character matching function `LIKE` is an empty string and its pattern is `%%`, False is returned. After the change, True is returned.

```
-- Create a table and insert an empty string into the table.
create table if not exists table_test (a string) lifecycle 3;
insert into table table_test values ('');
select a like '%%' from table_test;
-- The following result is returned before the change.
```

```
+-----+
```

```
| _c0 |
```

```
+-----+
```

```
| false |
```

```
+-----+
```

```
-- The following result is returned after the change.
```

```
+-----+
```

```
| _c0 |
```

```
+-----+
```

```
| true |
```

```
+-----+
```

December 25, 2019 (UTC+08:00): MaxCompute is compatible with open source geospatial UDFs

MaxCompute is compatible with open source geospatial UDFs, which are implemented by Esri for Apache Hive. You can register open source geospatial UDFs with MaxCompute so that the functions can be called the same as open source Hive UDFs. For more information, see [Open source geospatial UDFs](#).

October 11, 2019 (UTC+08:00): New features of MaxCompute SQL are introduced

- Support for specifying the priorities of the JOIN operations or set operations, including UNION, INTERSECT, and EXCEPT by using parentheses ().

```
SELECT * FROM src JOIN (src2 JOIN src3 on xxx) ON yyy;
SELECT * FROM src UNION ALL (SELECT * FROM src2 UNION ALL SELECT * FROM src3);
```

For more information, see [JOIN](#) and [INTERSECT](#), [UNION](#), [EXCEPT](#), and [MINUS](#).

- Support for `odps.sql.orderby.position.alias` and `odps.sql.groupby.position.alias`

If the two flags are enabled, integer constants in the ORDER BY and GROUP BY clauses of the SELECT statements are processed as column IDs.

```
The columns in the src table can be sorted by keys or values.
SELECT * FROM src ORDER BY 1;
-- The preceding statement is equivalent to the following statement:
SELECT * FROM src ORDER BY key;
```

For more information, see [SELECT syntax](#).

- Support for new built-in functions

- `STRING JSON_TUPLE(String json,String key1,String key2,...)` : converts a JSON string to a tuple based on a group of keys. The `JSON_TUPLE()` function supports multi-level nesting. It can parse JSON data that contains Chinese characters or nested arrays. To parse the same JSON string multiple times, you must call the `GET_JSON_OBJECT()` function multiple times. However, the `JSON_TUPLE()` function can parse the JSON string only once after you enter multiple keys at a time. Compared with `GET_JSON_OBJECT()`, `JSON_TUPLE()` is more efficient. For more information, see [String functions](#).
- `INT EXTRACT(datepart from timestamp)` : extracts a part specified by the datepart parameter from a date value. The value of datepart can be YEAR, MONTH, or DAY. The value of timestamp is a date value of the `TIMESTAMP` type. For more information, see [Date functions](#).

- Support for the default values of columns in a table

The `DEFAULT VALUE` clause allows you to specify a default value for a column when you create a table. If you do not specify a value for the column in an `INSERT` statement, this default value is inserted. Example:

```
CREATE TABLE t (a bigint default 10, b bigint);
INSERT INTO TABLE t(b) SELECT 100;
-- The preceding statement is equivalent to the following statement:
INSERT INTO TABLE t(a, b) SELECT 10, 100;
```

- Support for a `NATURAL JOIN` operation

A `NATURAL JOIN` operation is a process in which two tables are joined based on their common columns. Common columns are columns that have the same name in both tables. MaxCompute supports `OUTER NATURAL JOIN`. You can use the `USING` clause so that the `JOIN` operation returns common columns only once. Example:

```
-- To join the src table that contains the key1, key2, a1, and a2 columns and the src2 table that contains the
key1, key2, b1, and b2 columns, you can execute the following statement:
SELECT * FROM src NATURAL JOIN src2;
-- Both the src and src2 tables contain the key1 and key2 fields. In this case, the preceding statement is equivalent
to the following statement:
SELECT src.key1 as key1, src.key2 as key2, src.a1, src.a2, src2.b1, src2.b2 FROM src INNER JOIN src2 ON src.
key1 = src2.key1 AND src.key2 = src2.key2;
```

For more information, see [JOIN](#).

- Support for simultaneous execution of the `OFFSET` and `ORDER BY LIMIT` clauses

The `OFFSET` clause can be used together with the `ORDER BY LIMIT` clause to skip the rows specified by `OFFSET`. For example, execute the following statement to sort the rows of the `src` table in ascending order by `key`, and return the 11th to 20th rows. `OFFSET 10` indicates that the first 10 rows are skipped. `LIMIT 10` indicates that a maximum of 10 rows can be returned.

```
SELECT * FROM src ORDER BY key LIMIT 10 OFFSET 10;
```

For more information, see [SELECT syntax](#).

- Support for built-in operators

- The `IS DISTINCT FROM` operator is supported. `a is distinct from b` is equivalent to `not(a <=> b)`, and `a is not distinct from b` is equivalent to `a <=> b`.
- The `||` operator is supported to concatenate strings. For example, `a || b || c` is equivalent to `concat(a, b, c)`.

For more information, see [Operators](#).

- Support for partition merging

In MaxCompute, a maximum of 60,000 partitions can be created in a table. If excessive partitions exist, you can merge partitions to archive data in a data warehouse. This reduces the number of partitions. If you trigger partition merging, MaxCompute merges multiple partitions in the same table into one partition, migrates their data to the merged partition, and then drops the previous partitions. The following syntax is used to merge partitions. For more information, see [Partition and column operations](#).

```
ALTER TABLE <tableName> MERGE [IF EXISTS] PARTITION(<predicate>) [, PARTITION(<predicate2>) ...] OVERWRITE PARTITION(<fullPartitionSpec>);
```

- Add/Drop Partitions

MaxCompute allows you to add or drop multiple partitions at a time by using statements. Syntax for the statements:

```
ALTER TABLE t ADD [IF NOT EXISTS] PARTITION (p = '1') PARTITION (p = '2');
ALTER TABLE t DROP [IF EXISTS] PARTITION (p = '1'), PARTITION (p = '2');
-- Note that no commas (,) exist between partitions in the ADD clause, whereas commas (,) are used to separate partitions in the DROP clause.
```

August 29, 2019 (UTC+08:00): A custom storage handler for an external table is used to upgrade the Outputter interface in some regions

On August 29, 2019, MaxCompute is upgraded. The upgrade may fail if you use a custom storage handler for an external table to update the Outputter interface and the column data is obtained by column name instead of array subscript.

Upgrade time: 14:00 to 23:00 on August 29, 2019 (UTC+08:00)

Regions: US (Silicon Valley) and Singapore (Singapore)

August 21, 2019 (UTC+08:00): A custom storage handler for an external table is used to upgrade the Outputter interface in some regions

On August 21, 2019, MaxCompute is upgraded. The upgrade may fail if you use a custom storage handler for an external table to upgrade the Outputter interface and the column data is obtained by column name instead of array subscript.

Upgrade time: 14:00 to 23:00 on August 21, 2019 (UTC+08:00)

Regions: Japan (Tokyo), Germany (Frankfurt), China (Hong Kong), and Australia (Sydney)

Impact: In `Outputter.output(Record record)`, the input record is generated by the last operator of Outputter. Column names are not fixed.

For example, the column name that is generated by the `some function(column_a)` expression is a temporary column name. We recommend that you use `record.get(index)` instead of `record.get(Column name)` to obtain the content of a column. To obtain column names from a table in Outputter, call `DataAttributes.getFullTableColumns()`.

If you have questions, submit a ticket.

July 24, 2019 (UTC+08:00): Spark on MaxCompute is supported

Regions: China (Hangzhou), China (Beijing), China (Shenzhen), US (Silicon Valley), China (Hong Kong), Germany (Frankfurt), Singapore (Singapore), and India (Mumbai)

March 26, 2019 (UTC+08:00): MaxCompute SQL is upgraded

- The GROUPING SETS clause can be used with both the CUBE and ROLLUP subclauses to aggregate and analyze data of multiple dimensions. For example, you must aggregate Column a, aggregate Column b, or aggregate both Column a and Column b. For more information, see [GROUPING SETS](#).
- The INTERSECT, MINUS, and EXCEPT clauses are supported. For more information, see [INTERSECT, UNION, EXCEPT, and MINUS](#).
- If MaxCompute reads files in the ORC or Parquet format in OSS by using external tables, it prunes the columns in the files to reduce I/O, save resources, and lower overall computing costs.
- Systems that run in the Java UDX framework are enhanced to support writable parameters. For more information, see [Java UDFs](#).

Optimized SQL performance

- DynamicDAG: a required mechanism for dynamic optimization. DynamicDAG delays the optimization of resource allocation or algorithm selection and triggers it at runtime to improve optimizations and reduce the probability of generating low-performance execution plans.
- ShuffleRemove optimization: optimization for shuffles. MaxCompute supports ShuffleRemove for right tables that have duplicate keys during the execution of the INNER JOIN clause.

March 1, 2019 (UTC+08:00): MaxCompute SQL jobs that process data in external tables begin to incur charges

As of March 1, 2019, all MaxCompute SQL jobs that are used to process data in OSS and Tablestore tables begin to incur charges.

Billing standard for SQL computing that involves external tables:

Fee of an SQL job that involves external tables = Amount of input data × Unit price for SQL computing that involves external tables

The unit price for SQL computing that involves external tables is USD 0.0044 per GB. All fees are charged the next day, and you will receive a bill in your account. For more information, see [Billing method](#). If you have questions, submit a ticket.

16:00 to 20:00 on January 15, 2019 (UTC+08:00): The underlying metadata warehouse of MaxCompute in the China (Hong Kong) region is optimized

The underlying metadata warehouse of MaxCompute in the China (Hong Kong) region is optimized from 16:00 to 20:00 on January 15, 2019. This optimization improves the performance and stability of MaxCompute. During the optimization, users in the China (Hong Kong) region may encounter job submission delays or job failures for about 1 minute. In the worst cases, applications may be unavailable for up to 30 minutes. We recommend that you do not submit jobs during optimization. Users in other regions are not affected. If you have questions, contact the MaxCompute team over DingTalk or submit a ticket.

December 24, 2018 (UTC+08:00): MaxCompute supports time zone configuration

The default time zone of MaxCompute projects is UTC+8. The system executes time-related built-in functions and calculates fields of the DATETIME, TIMESTAMP, and DATE types based on UTC+8. As of December 24, 2018, users can configure time zones in MaxCompute for a session or project.

- To configure a time zone for a session, execute the `set odps.sql.timezone=<timezoneid>;` statement along with a computing statement. Example:

```
set odps.sql.timezone=Asia/Tokyo;
select getdate();
-- The following result is returned:
output:
+-----+
|_c0   |
+-----+
| 2018-10-30 23:49:50 |
+-----+
```

- To configure a time zone for a project, execute the `setProject odps.sql.timezone=<timezoneid>;` statement as the project owner. After the time zone of a project is configured, the time zone is used for all time computing, and the data of existing jobs is affected. Therefore, exercise caution when you perform this operation. We recommend that you perform this operation only on new projects.

Limits and usage notes:

- SOL built-in date functions, UDFs, user-defined type (UDT), user-defined joins (UDJs), and the `SELECT TRANSFORM` statement allow you to obtain the time zone attribute of a project to configure the time zone.
- The `Asia/Shanghai` time zone is supported. If this time zone is used, daylight saving time is considered. The GMT+9 time zone is not supported.
- If the time zone in the SDK differs from that of the project, you must configure the GMT time zone to convert the data type from DATETIME to STRING.
- After the time zone is configured, differences exist between the real time and the output time of the related SQL statements that you execute in DataWorks. Between the years of 1900 and 1928, the time difference is 352 seconds. Before the year of 1900, the time difference is 9 seconds.
- MaxCompute, SDK for Java, and the related client are upgraded to ensure that data of the DATETIME type in MaxCompute is accurate across time zones. The destination versions of SDK for Java and the related client have the `-oversea` suffix. The upgrade may affect the display of data of the DATETIME type that was generated before January 1, 1928 in MaxCompute.
- If the local time zone is not UTC+8 when you upgrade MaxCompute, we recommend that you also upgrade the SDK for Java and the related client. This ensures that the SQL-based computing result and data that is transferred by using Tunnel commands after January 1, 1900 are accurate and consistent. For DATETIME data that was generated before January 1, 1900, the SQL-based computing result and data that is transferred by using Tunnel commands may differ up to 343 seconds. For DATETIME data that was generated before January 1, 1928 and was uploaded before SDK for Java and the related client are upgraded, the time in the new version is 352 seconds earlier.
- If you do not upgrade the SDK for Java or the related client to versions with the `-oversea` suffix, the SQL-based computing result may differ from data that is transferred by using Tunnel commands. For data that was generated before January 1, 1900, the time difference is 9 seconds. For data that was generated within the period from January 1, 1900 to January 1, 1928, the time difference is 352

seconds.

Note The modification of the time zone configuration in SDK for Java or on the related client does not affect the time zone configuration in DataWorks. Therefore, the time zones are different. You must evaluate how this may affect scheduled jobs in DataWorks. The time zone of a DataWorks server in the Japan (Tokyo) region is GMT+9, and that in the Singapore (Singapore) region is GMT+8.

- If you are using a third-party client that is connected to MaxCompute by using Java Database Connectivity (JDBC), you must configure the time zone on the client to ensure that the time of the client and that of the server are consistent.
- MapReduce supports time zone configuration.
- Spark on MaxCompute supports time zone configuration.
 - If jobs are submitted to the MaxCompute computing cluster, the time zone of the project is automatically obtained.
 - If jobs are submitted from spark-shell, spark-sql, or pyspark in yarn-client mode, you must configure parameters in the `spark-defaults.conf` file of the driver and add `spark.driver.extraJavaOptions=-Duser.timezone=America/Los_Angeles`. The `timezone` parameter specifies the time zone you want to use.
- Machine Learning Platform for AI (PAI) supports time zone configuration.
- Graph supports time zone configuration.

2.Document updates

2.1. Updates in 2021

This topic describes the latest updates to MaxCompute documentation in 2021. These updates provide detailed information about the new features, new syntax, and permission updates in MaxCompute to help you improve the efficiency of project development.

Updates in August 2021

Release date	Feature	Category	Description	References
2021-08-27	Data lakehouse of MaxCompute based on the Delta Lake or Apache Hudi storage mechanism	New feature	Delta Lake and Apache Hudi are commonly used storage mechanisms in data lake solutions. These storage mechanisms provide stream processing and batch processing capabilities for data lakes. MaxCompute provides a data lakehouse solution that supports Delta Lake and Apache Hudi. This solution is developed based on Alibaba Cloud services, such as Data Lake Formation (DLF), Relational Database Service (RDS), and Object Storage Service (OSS).	Data lakehouse of MaxCompute based on the Delta Lake or Apache Hudi storage mechanism
2021-08-26	CORR function	New description	The CORR function is added to mathematical functions. This function calculates the Pearson correlation coefficient for two columns of data.	Mathematical functions
2021-08-25	Guidance on planning resources and selecting resource packages	New description	A guidance is provided for you to estimate the number of resources that you need to purchase and help you select a resource package.	Plan resources and select a resource package
2021-08-20	Syntax updates for aggregate functions	Updated description	The "Syntax" section is added to the "Aggregation functions" topic.	Aggregate functions

Release date	Feature	Category	Description	References
2021-08-19	HASH, COMPRESS, DECOMPRESS, NULLIF, FORMAT_NUMBER, SHA, SHA1, SHA2, and CRC32 added to other functions	New description	<ul style="list-style-type: none"> HASH: calculates the hash value of the input parameters. COMPRESS: uses the GZIP algorithm to compress input strings. DECOMPRESS: uses the GZIP algorithm to decompress the input parameters of the BINARY type. NULLIF: returns NULL if the values of expr 1 and expr 2 are the same. Otherwise, expr1 is returned. FORMAT_NUMBER: converts a number into a string of the specified format. SHA: calculates the SHA-1 hash value of a string or a binary value and returns a hexadecimal string. SHA1: calculates the SHA-1 hash value of a string or a binary value and returns the value as a hexadecimal string. SHA2: calculates the SHA-2 family hash value of a string or a binary value. SHA-224, SHA-256, SHA-384, and SHA-512 are supported. CRC32: calculates the cyclic redundancy check (CRC) value of a string or binary value. 	Other functions
2021-08-19	TO_MILLIS function	New description	The TO_MILLIS function is added to date functions. This function converts a date into a UNIX timestamp in milliseconds.	Date functions
2021-08-18	Change of the minimum number of MaxCompute CUs	Updated description	The minimum number of MaxCompute compute units (CUs) that you must purchase is changed from 10 CUs to 50 CUs.	Billing method

Release date	Feature	Category	Description	References
2021-08-11	Materialized view supported by MaxCompute	New feature	A materialized view is a database object that stores the pre-calculation results of a time-consuming query, such as JOIN or AGGREGATE. You can directly reuse the pre-calculation results without the need to perform such time-consuming queries. This accelerates the query speed.	Materialized view operations
2021-08-11	Commercial use of the UPDATE, DELETE, and MERGE INTO statements	Updated description	As of August 10, 2021, the public preview of the UPDATE, DELETE, and MERGE INTO statements of MaxCompute SQL ends. After the public preview ends, you are charged when you use these statements. Pay-as-you-go jobs that already use the UPDATE, DELETE, and MERGE INTO statements are charged computing fees after the public preview ends. If you did not use the UPDATE, DELETE, or MERGE INTO statement due to concerns about service stability in the public preview phase, you no longer need to worry about stability when you use these statements. MaxCompute provides the same guarantees for the availability and stability of the UPDATE, DELETE, and MERGE INTO statements as the guarantee for MaxCompute SQL.	<ul style="list-style-type: none"> • UPDATE and DELETE • MERGE INTO
2021-08-10	Functions of complex data types	New feature	MaxCompute can process data of complex data types, such as MAP, ARRAY, and STRUCT, and provides a variety of functions to support diversified data processing.	Complex type functions

Release date	Feature	Category	Description	References
2021-08-04	ETL tools	New description	Extract, transform, and load (ETL) tools can be used to manage MaxCompute projects.	<ul style="list-style-type: none"> • Use Kettle to schedule MaxCompute jobs • Use Apache Airflow to schedule MaxCompute jobs • Use Azkaban to schedule MaxCompute jobs
2021-08-02	IN/NOT IN SUBQUERY and SCALAR SUBQUERY	Updated description	IN/NOT IN SUBQUERY and SCALAR SUBQUERY support multi-column subqueries, which is a feature in PostgreSQL.	Subqueries

Updates in July 2021

Release date	Feature	Category	Description	References
2021-07-16	Top scheduled SQL tasks that are managed by using a dynamic filter	New feature	If a top scheduled SQL task is found, the ID of the most recent instance on which the task runs is displayed. We recommend that you use a dynamic filter to check and optimize the related job to reduce resource consumption and improve the job running performance. For more information, see Dynamic filtering .	Use MaxCompute Management
2021-07-15	Limits on packages	Updated description	The limits on packages are added.	Use a package

Updates in June 2021

Release date	Feature	Category	Description	References
2021-06-07	Description of data import and data export by using Data Integration	Updated description	The description of data import and data export by using Data Integration is modified.	<ul style="list-style-type: none"> • Import data by using Data Integration • Export data by using Data Integration

Updates in May 2021

Release date	Feature	Category	Description	References
2021-05-31	Descriptions related to preparation items	Updated description	The preparation procedure and related descriptions are modified.	Prepare
2021-05-31	Quick start	Updated description	A quick start guide is provided for each tool.	Quick Start

Updates in March 2021

Release date	Feature	Category	Description	References
--------------	---------	----------	-------------	------------

Release date	Feature	Category	Description	References
2021-03-25	MaxCompute error codes	Updated description	New information is added to the MaxCompute error code list.	<ul style="list-style-type: none"> • Error code overview • Common errors (ODPS-00CCCCX) • SQL errors (ODPS-01CCCCX) • PL errors (ODPS-02CCCCX) • API errors (ODPS-04CCCCX) • Xlib errors (ODPS-05CCCCX) • MapReduce errors (ODPS-07CCCCX) • ODPS SDK or CLT errors (ODPS-08CCCCX) • AlgoTask errors (ODPS-12CCCCX)
2021-03-16	NOT NULL attribute for a specific column in the CREATE TABLE syntax	Updated description	The NOT NULL attribute can be configured in the CREATE TABLE syntax to specify that the values in a specific column cannot be NULL.	Table operations

Release date	Feature	Category	Description	References
2021-03-16	FROM_JSON and TO_JSON functions	New description	The FROM_JSON and TO_JSON functions are added. You can use the FROM_JSON function to convert data in the JSON format into data of a data type that is supported by MaxCompute. You can also use this function to extract information from data in the JSON format and return data of the ARRAY, MAP, or STRUCT type based on the jsonStr and schema information. You can use the TO_JSON function to convert data in the ARRAY, MAP, or STRUCT format into data in the JSON format.	String functions
2021-03-16	ACID semantics for transactional tables	New feature	New atomicity, consistency, isolation, durability (ACID) semantics for transactional tables are added based on the ACID semantics for concurrent write jobs.	ACID semantics
2021-03-12	A periodic check for scheduled tasks that are continuously run with empty output and scheduled tasks with continuous full table scans	New feature	Top scheduled SQL tasks that are continuously run with empty output and top scheduled SQL tasks with continuous full table scans can be periodically checked in MaxCompute. If these tasks are found, the ID of the most recent instance on which the task runs is displayed. We recommend that you check and optimize the related jobs to reduce resource consumption.	Use MaxCompute Management
2021-03-12	Column-level access control	Updated description	Column-level access control is supported.	Authorize users
2021-03-12	Enabling and disabling of download control based on property settings	Updated description	A property can be configured to enable or disable download control. To enable or disable download control, you must configure a property at the project level as the project owner or a user who is assigned the Super_Administrator role.	Policy-based access control and download control

Release date	Feature	Category	Description	References
2021-03-01	Data lakehouse	New feature	MaxCompute provides the lakehouse solution, which enables you to build a data management platform that combines data lakes and data warehouses. The lakehouse solution integrates the flexibility and diverse ecosystems of data lakes with the enterprise-class deployment of data warehouses.	Lakehouse of MaxCompute

Updates in February 2021

Release date	Feature	Category	Description	References
2021-02-23	Example on how to use a Python 2 UDTF to read resources from MaxCompute	New example	An example of how to use a Python 2 user-defined table-valued function (UDTF) to read resources from MaxCompute is added.	Python 2 UDTFs
2021-02-23	Example on how to use a Python 3 UDTF to read resources from MaxCompute	New example	An example of how to use a Python 3 UDTF to read resources from MaxCompute is added.	Python 3 UDTFs
2021-02-23	External files referenced by Spark jobs	New description	The description that Spark jobs reference external files is added.	Set up a Spark on MaxCompute development environment
2021-02-20	An example on how to use a UDTF with LATERAL VIEW is added.	New example	An example of how to use a UDTF with LATERAL VIEW is added.	UDTF usage
2021-02-09	Deletion of user permissions	New description	The description for the deletion of user permissions is added.	Manage users
2021-02-07	Removal of the limit that only constants can be used in VALUES	Updated description	The limit that only constants can be used in VALUES is removed.	VALUES
2021-02-07	Multipart upload of Object Storage Service (OSS) external tables	New description	The multipart upload feature of OSS is supported. This feature allows you to perform the INSERT operation to write data to OSS external tables.	Export unstructured data to OSS

Release date	Feature	Category	Description	References
2021-02-07	<code>ARRAY_INTERSECT</code> and <code>SORT_ARRAY</code> functions	New description	The <code>ARRAY_INTERSECT</code> function is added to calculate the intersection of two arrays. The <code>isAsc</code> parameter is added to the <code>SORT_ARRAY</code> function. This parameter specifies whether to sort the given arrays in ascending or descending order. The default value of this parameter is True, which indicates that given arrays are sorted in ascending order.	Other functions
2021-02-07	Filter expressions and the <code>COUNT_IF</code> function	New description	Filter expressions are supported. You can specify filter conditions for an aggregate function in a <code>SELECT</code> clause. This way, you can separately control the data range of the aggregate function in an aggregate statement. The <code>COUNT_IF</code> function is added to count the number of data records that meet the <code>IF</code> condition.	Aggregate functions
2021-02-07	Support for the DATE and TIMESTAMP data types in <code>DATEADD</code> , <code>DATE_ADD</code> , <code>DATEDIFF</code> , <code>DATEPART</code> , and <code>DATETRUNC</code> functions	New description	The date functions <code>DATEADD</code> , <code>DATE_ADD</code> , <code>DATEDIFF</code> , <code>DATEPART</code> , and <code>DATETRUNC</code> are enhanced to support the DATE and TIMESTAMP data types.	Date functions
2021-02-07	<code>BASE64</code> , <code>UNBASE64</code> , and <code>PARSE_URL</code> functions	New description	The string function <code>PARSE_URL</code> is added to parse URLs. The <code>BASE64</code> and <code>UNBASE64</code> functions are added to support conversions between binary strings and Base64-encoded strings.	String functions

Release date	Feature	Category	Description	References
2021-02-07	Disabling of header display in the MaxCompute command line to facilitate shell calls	New description	In a shell window or in the Windows command line, you may need to use the dynamic return value that is obtained by executing the <code>odpscmd -e SQL</code> statement. The variable of the shell can obtain this dynamic return value and run subsequent jobs in the shell. In this scenario, only field values are required. Other information, such as runtime information and headers, must not be returned. You can run <code>set odps.sql.select.output.format={needHeader:false,fieldDelim:''}</code> to disable the header display and export the standard output from the calculation result to the destination handle.	MaxCompute client
2021-02-07	Data export from MaxCompute to OSS by using UNLOAD	New feature	MaxCompute allows you to export data to OSS by using the UNLOAD command. This way, you can use OSS to store structured data and use other computing engines in OSS to process and analyze the data that is exported from MaxCompute.	UNLOAD
2021-02-07	SKEWJOIN HINT	New feature	If two tables that you want to join have hot key values, a long tail may occur. In this case, SKEWJOIN HINT can be used to automatically or manually extract hot key values from the two tables, separately calculate the join results of the hot key values and non-hot key values, and then join the calculated data. This way, the JOIN operation is accelerated.	SKEWJOIN HINT

Release date	Feature	Category	Description	References
2021-02-07	Reading and writing MC-Hologres data by using the JDBC driver	New feature	You can use MaxCompute external tables to access data of MC-Hologres data sources by using the Java Database Connectivity (JDBC) driver. You can create an MC-Hologres external table in MaxCompute to query the data of MC-Hologres data sources by using the JDBC driver that is provided by PostgreSQL. This method prevents redundant data storage and allows you to obtain query results at a fast speed without the need to import or export data. Reading and writing data of MC-Hologres external tables is in the public preview stage. Data computing is free of charge but the Service Level Agreement (SLA) is not guaranteed.	Hologres external tables
2021-02-01	Change records of Mars V0.6.3	New description	Change records of Mars V0.6.3 are added.	Version updates

Updates in January 2021

Release date	Feature	Category	Description	References
2021-01-21	Change in the return behavior of the <code>GET_JSON_OBJECT</code> function	Updated description	For new MaxCompute projects, the <code>GET_JSON_OBJECT</code> function retains the original strings instead of escaping JSON reserved characters when the function returns a value.	String functions
2021-01-19	Content adjustment and optimization of the SELECT syntax	Updated description	The content of the SELECT syntax is adjusted, and sample statements are optimized.	SELECT syntax
2021-01-05	Description of data migration	New description	The Data migration topic is added. The content related to data upload and download and the migration tool is moved under this topic.	Data upload and download

2.2. Updates in year 2020

This topic describes the latest updates to MaxCompute documentation in year 2020. These updates allow you to understand the new features, syntax, and permissions in MaxCompute and help you improve project development efficiency.

Updates in December 2020

Release date	Feature	Category	Description	Documenta tion
2020-12-21	STACK and GET_USER_ID functions	New descripti on	The STACK function is added to separate specified data into a specified number of columns. The GET_USER_ID function is added to obtain the ID of the current account.	Other functions
2020-12-21	DataWorks-based permission management for roles on a project	New descripti on	DataWorks is used to manage permissions of a role on a project.	Use DataWorks to manage permission s of a role on a project
2020-12-21	Information Schema service provided by default	Updated descripti on	By default, a new MaxCompute project provides the Information Schema service. You do not need to manually install the Information Schema permission package.	Overview of Information Schema
2020-12-15	Version updates of Mars	New descripti on	Version updates of Mars are provided.	Version updates
2020-12-15	Version updates of SDK for Java	New descripti on	Version updates of SDK for Java are provided.	Version updates
2020-12-15	Version updates of the JDBC driver	New descripti on	Version updates of the JDBC driver are provided.	Version updates
2020-12-11	TASK view added to the metadata view list	New descripti on	The TASK view is added to store real-time snapshots of running jobs.	Metadata views
2020-12-10	Instructions on how to add project members and configure roles	New descripti on	If you create a project by using your Alibaba Cloud account and require RAM users for data development, you can add the RAM users as project members and configure roles for these users.	Add a workspace member and configure roles
2020-12-09	AdminEvent events	New descripti on	The CreateProject, UpdateProject, and DeleteProject events are added to record project-related logs.	Audit logs

Release date	Feature	Category	Description	Documentation
2020-12-08	Adjustment and optimization of the content of common commands	Updated description	The content of common commands is adjusted, and sample commands are optimized.	Common SQL statements
2020-12-08	Adjustment and optimization of the content of INSERT statements	Updated description	The content of INSERT statements is adjusted, and statement examples are optimized.	<ul style="list-style-type: none"> • Insert or update data into a table or static partitions (INSERT INTO and INSERT OVERWRITE) • Insert or overwrite data into dynamic partitions (DYNAMIC PARTITION) • MULTI INSERT • VALUES

Updates in November 2020

Release date	Feature	Category	Description	Documentation
2020-11-30	Information of public datasets	Updated description	Information of public datasets is added to the International site (alibabacloud.com) for testing and trial use.	Overview
2020-11-30	Query editor	New feature	The MaxCompute V2.0 console provides a query editor for you to execute SQL statements and analyze data.	Query editor

Release date	Feature	Category	Description	Documentation
2020-11-27	Configuration upgrade and downgrade	Updated description	Configuration upgrade and downgrade are performed in the MaxCompute V2.0 console.	Upgrade or downgrade configurations
2020-11-26	Fields of audit logs	Updated description	Fields of audit logs are updated.	Audit logs
2020-11-20	ORDER BY without LIMIT	New description	MaxCompute supports the execution of ORDER BY without LIMIT .	SELECT syntax
2020-11-17	Instructions on how to reference third-party packages in Python user-defined functions (UDFs)	New description	Instructions on how to reference third-party packages in Python UDFs are added. These packages include NumPy packages, third-party packages that need to be compiled, and third-party packages that are dependent on dynamic-link libraries (DLLs).	Reference third-party packages in Python UDFs
2020-11-04	Optional parameters for enabling MaxCompute Query Acceleration (MCQA) on the MaxCompute JDBC driver	Updated description	Optional parameters are added for enabling MCQA on the MaxCompute JDBC driver.	Usage notes

Updates in October 2020

Release date	Feature	Category	Description	Documentation
2020-10-30	MAPJOIN hint supported by SEMI JOIN	Updated description	SEMI JOIN supports the MAPJOIN hint, which improves the performance of LEFT SEMI or ANTI JOIN and resolves data skew issues.	SEMI JOIN
2020-10-30	SORT BY without DISTRIBUTE BY	Updated description	MaxCompute supports the execution of SORT BY without DISTRIBUTE BY . This resolves data reordering issues and improves the filtering performance during the execution of SQL statements.	SELECT syntax

Release date	Feature	Category	Description	Documentation
2020-10-30	ZORDER BY clause supported by INSERT	New feature	The INSERT statement supports the ZORDER BY clause, which can co-locate rows with similar data together. This improves the filtering performance during queries and reduces storage costs.	Insert or update data into a table or static partitions (INSERT INTO and INSERT OVERWRITE)
2020-10-30	Deletion of multiple partitions at the same time by using conditional filtering	New feature	If you want to delete one or more partitions that meet a specific condition at the same time, you can use a conditional expression to delete the partitions that match the condition at the same time.	Partition and column operations
2020-10-30	GBK encoding supported for OSS external tables in the CSV or TSV format	Updated description	The <code>odps.text.option.encoding</code> property supports GBK encoding.	Access OSS data by using a built-in extractor
2020-10-30	DATETIME data type supported by time functions YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE, and SECOND	Updated description	The time functions YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE, and SECOND support the DATETIME data type.	Date functions
2020-10-30	WIDTH_BUCKET function	New feature	The WIDTH_BUCKET function is added. This function returns the ID of the bucket into which the value of a specific field falls.	Mathematical functions
2020-10-12	Commercial use of MCQA	Updated description	The MCQA feature is available for commercial use and charged.	Overview
2020-10-10	Modification of the clustering attribute of tables	New description	The description of how to modify the clustering attribute of tables is added.	Table operations

Updates in September 2020

Release date	Feature	Category	Description	Documentation
--------------	---------	----------	-------------	---------------

Release date	Feature	Category	Description	Documentation
2020-09-24	PyODPS module	New description	The description of PyODPS jobs is added.	PyODPS
2020-09-17	Description of creating a RAM user	New description	The description of creating a RAM user is added.	Prepare a RAM user
2020-09-08	Description of the LOAD command	New description	The description of the LOAD command is added.	LOAD
2020-09-03	Description of Tunnel Upload	Updated description	The description of the Tunnel Upload command is updated.	Tunnel commands
2020-09-01	Best practice to migrate data from BigQuery to MaxCompute	New practice	A best practice is added to describe how to migrate data from BigQuery to MaxCompute.	Migrate data from BigQuery to MaxCompute
2020-09-01	Best practice to migrate data from Amazon Redshift to MaxCompute	New practice	A best practice is added to describe how to migrate data from Amazon Redshift to MaxCompute.	Migrate data from Amazon Redshift to MaxCompute

Updates in August 2020

Release date	Feature	Category	Description	Documentation
2020-08-07	Best practice to optimize costs	New practice	A best practice is added to describe how to optimize the computing, storage, data upload, and data download costs.	Overview
2020-08-06	Best practice to assign the Super_Administrator role to a RAM user for a MaxCompute project	New practice	A best practice is added to describe how to assign the Super_Administrator role to a RAM user for a MaxCompute project. It also describes how to manage members and permissions by using the Super_Administrator role.	Set a RAM user as the super administrator for a MaxCompute project

Release date	Feature	Category	Description	Documentation
2020-08-05	Best practices to segment Chinese texts by using Jieba in a PyODPS node	New practice	The following best practices are added: 1. Segment Chinese texts by using Jieba, an open source segmentation tool, and write the segmented words and phrases to a new table in a PyODPS node in DataWorks. 2. Use Jieba to segment Chinese texts based on a custom dictionary referenced by a closure function.	Use a PyODPS node to segment Chinese text based on Jieba
2020-08-05	Best practice to grant access to a specific UDF only to a specified user	New practice	A best practice is added to describe how to set resources (tables or UDFs) to be accessible only to specified users.	Grant access to a specific UDF to a specified user
2020-08-05	Best practice to migrate data from Oracle to MaxCompute	New practice	A best practice is added to describe how to use the data integration feature of DataWorks to migrate data from Oracle to MaxCompute.	Best practice to migrate data from Oracle to MaxCompute
2020-08-05	Best practice to migrate data from a self-managed MySQL database on an Elastic Compute Service (ECS) instance to MaxCompute	New practice	A best practice is added to describe how to use exclusive resource groups for data integration to migrate data from a self-managed MySQL database on an ECS instance to MaxCompute.	Migrate data from a user-created MySQL database on an ECS instance to MaxCompute
2020-08-05	<code>odps.text.option.use.quote</code> property supported by SERDEPROPERTIES	New description	The description of the <code>odps.text.option.use.quote</code> property is added. This property specifies whether to recognize a double quotation mark (") as the column delimiter in a CSV file.	Access OSS data by using a built-in extractor

Updates in July 2020

Release date	Feature	Category	Description	Documentation
--------------	---------	----------	-------------	---------------

Release date	Feature	Category	Description	Documentation
2020-07-29	Best practice to migrate data from MaxCompute to Tablestore	New practice	A best practice is added to describe how to migrate data from MaxCompute to Tablestore.	Migrate data from MaxCompute to Tablestore
2020-07-29	Best practice to migrate data from MaxCompute to Object Storage Service (OSS)	New practice	A best practice is added to describe how to use the data synchronization feature of DataWorks to migrate data from MaxCompute to OSS.	Migrate data from MaxCompute to OSS
2020-07-24	Data encryption	New feature	The description of the data encryption feature is added. MaxCompute uses Key Management Service (KMS) to encrypt data for storage. This way, MaxCompute can provide static data protection to meet corporate governance and security compliance requirements.	Data encryption
2020-07-23	Aggregate functions	New description	The description of aggregate functions <code>APPROX DISTINCT</code> , <code>ANY VALUE</code> , <code>ARG_MAX</code> , and <code>ARG_MIN</code> is added.	Aggregate functions
2020-07-23	New data types supported by Python UDFs	New description	New data types are supported by Python UDFs.	<ul style="list-style-type: none"> • Python 2 UDFs • Python 3 UDFs
2020-07-23	SQL functions	New feature	The description of SQL functions that allow you to reference SQL UDFs in SQL scripts is added.	SQL functions
2020-07-23	Code-embedded UDFs	New feature	The description of code-embedded UDFs is added. Code-embedded UDFs allow you to embed Java or Python code into SQL scripts.	<ul style="list-style-type: none"> • Code-embedded UDFs • Usage examples
2020-07-20	Audit logs	New feature	The description of the features, scenarios, scope, and fields of audit logs is added.	Audit logs

Release date	Feature	Category	Description	Documentation
2020-07-15	Best practice to use Tunnel to upload log data to MaxCompute	New practice	A best practice is added to describe how to use Tunnel to upload log data to MaxCompute.	Use Tunnel to upload log data to MaxCompute
2020-07-15	Best practice to use DataHub to migrate log data to MaxCompute	New practice	A best practice is added to describe how to use DataHub to migrate log data to MaxCompute.	Use DataHub to migrate log data to MaxCompute
2020-07-15	Best practice to use the data integration feature of DataWorks to migrate log data to MaxCompute	New practice	A best practice is added to describe how to use the data integration feature of DataWorks to synchronize data collected by LogHub to MaxCompute.	Use DataWorks Data Integration to migrate log data to MaxCompute
2020-07-08	CLONE TABLE	New feature	The description of the CLONE TABLE statement is added. MaxCompute supports the CLONE TABLE statement that allows you to clone data from one table to another. This statement facilitates data migration and replication.	CLONE TABLE

Updates in June 2020

Release date	Feature	Category	Description	Documentation
2020-06-03	Tunnel OVERWRITE command	New description	The description of the Tunnel OVERWRITE command is added.	Tunnel commands
2020-06-01	Optimization of access to instances in a VPC from Spark on MaxCompute	New description and example	<p>The following content is added:</p> <ul style="list-style-type: none"> Limits on VPC whitelists and regions Examples of merged JSON text when Spark on MaxCompute is used to access different instances 	Access instances in a VPC from Spark on MaxCompute

Release date	Feature	Category	Description	Documentation
2020-06-01	Policy-based access control and download control	New example	Examples on how to use policy-based access control and permission revocation are added.	Policy-based access control and download control

Updates in January 2020

Release date	Feature	Category	Description	Documentation
2020-01-14	Improved SQL compatibility	New description	The execution rules of the <code>GET_IDCARD_AGE</code> , <code>CONCAT_WS</code> , and <code>LIKE</code> functions are modified.	<ul style="list-style-type: none"> Other functions String functions LIKE usage
2020-01-09	Parameter description	New description	The parameters in examples are described.	Project data protection

2.3. Updates in year 2019

This topic describes the latest updates to MaxCompute documentation in 2019. These updates allow you to understand the new features, syntax, and permissions in MaxCompute and help you improve project development efficiency.

Updates in December 2019

Release date	Feature	Category	Description	Documentation
2019-12-25	Open source geospatial UDFs	New feature	Open source geospatial UDFs are added. You can register open source geospatial UDFs with MaxCompute and use them as open source Hive UDFs.	Open source geospatial UDFs

Updates in November 2019

Release date	Feature	Category	Description	Documenta tion
2019-11-06	Description of whether MaxCompute functions support partition pruning	New descripti on	The description of whether MaxCompute functions support partition pruning is added.	Comparison of functions built in MaxCompute, MySQL, and Oracle

Updates in October 2019

Release date	Feature	Category	Description	Documenta tion
2019-10-09	New SQL syntax	New feature	<ul style="list-style-type: none"> The syntax to merge partitions is added. The syntax that uses a pair of parentheses () to specify the priorities of operations in JOIN and SETOP statements is added. The built-in function JSON_TUPLE is added. The date function EXTRACT is added. Two flags are added. ORDER BY LIMIT can be used with OFFSET. Default values can be specified for the columns in a table. The NATURAL JOIN statement is supported. New operators are supported. The syntax to delete partitions is added. 	<ul style="list-style-type: none"> Partition and column operations JOIN String functions Date functions SELECT syntax SELECT syntax Table operations JOIN Operators Partition and column operations

Updates in July 2019

Release date	Feature	Category	Description	Documenta tion
--------------	---------	----------	-------------	-------------------

Release date	Feature	Category	Description	Documenta tion
2019-07-12	odps-sql-reshuffle-dynamicpt added to the SET command	New command	odps-sql-reshuffle-dynamicpt is added to configure dynamic partitions. This prevents the excessive generation of small files when dynamic partitions are split.	SET operations

Updates in June 2019

Release date	Feature	Category	Description	Documenta tion
2019-06-17	Description of the VALUES statement	New description	The description of how to create data for simple business testing is added.	VALUES