

ALIBABA CLOUD

阿里云

云原生数据仓库AnalyticDB
MySQL版
分析型数据库MySQL版2.0

文档版本：20210108

 阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置>网络>设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 确定 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.系统函数	05
1.1. 字符串函数	05
1.1.1. 字符串函数	05

1.系统函数

1.1. 字符串函数

1.1.1. 字符串函数

AnalyticDB for MySQL支持以下字符串函数。

- **CHR**: 返回Unicode编码对应的字符串。
- **CONCAT**: 连接字符串。
- **GROUP_CONCAT**: 通常与group by一起使用, 用于将group by产生的同一个分组中的值连接起来, 返回一个字符串结果。
- **LENGTH**: 返回字符串长度。
- **LOWER**: 将字符串转换为小写。
- **UPPER**: 将字符串转换为大写。
- **LPAD**: 左拼接字符串。
- **RPAD**: 右拼接字符串。
- **TRIM/LTRIM/RTTRIM**: 删除字符串前后所有的空格/前导空格/后置空格。
- **REPLACE**: 删除字符串中的所有指定子串。
- **REVERSE**: 将字符串逆序。
- **SPLIT**: 将字符串按分隔符进行分隔, 并返回数组。
- **POSITION**: 返回字符串中子字符串的第一次出现的起始位置。
- **CHAR**: 返回十进制数字对应的字符形成的字符串。
- **HEX**: 返回十六进制字符串。
- **INITCAP**: 将字符串首字符转换为大写。
- **INSTR**: 返回指定子串首次(或者指定次数)出现在字符串中的位置。
- **ASCII**: 返回字符或者字符串最左边字符对应的ASCII值。
- **MID**: 作用同**SUBSTR/SUBSTRING**
- **REPEAT**: 返回字符串重复多次的字符串。
- **STRPOS**: 返回字符串中子字符串的第一次出现的起始位置。
- **SUBSTR/SUBSTRING**: 返回一个从指定位置开始的指定长度的子字符串。
- **TRANSLATE**将字符表达式值中, 指定字符替换为新字符。

CHR

```
chr(n)
```

- 命令说明: 返回Unicode编码为n的字符。
- 返回值类型: VARCHAR。
- 示例:

```
SELECT Chr(65), Chr(66);
```

返回值如下：

```

+-----+-----+
|_col0 |_col1 |
+-----+-----+
| A   | B   |

```

CONCAT

```
concat(string1, ..., stringN)
```

- 命令说明：字符串连接操作，与标准SQL的连接运算符 || 功能相同。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Concat('aliyun',' ','analyticdb')
```

返回值如下：

```

SELECT Concat('aliyun',' ','analyticdb')
+-----+
|_col0          |
+-----+
| aliyun,analyticdb

```

GROUP_CONCAT

```
group_concat([distinct] 要连接的字段)
```

- 命令说明：group_concat函数通常与group by一起使用，用于将group by产生的同一个分组中的值连接起来，返回一个字符串结果。通过使用distinct可以排除重复值。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT * FROM user
```

返回值如下：

```
+-----+
| id |username|
+-----+
| 4 | lucy |
| 3 | Lucy |
| 2 | lily |
| 1 | lily |
```

```
select group_concat(distinct username)from user GROUP BY 'username';
```

返回值如下：

```
+-----+
| _col0 |
+-----+
| Lucy,Lily |
```

LENGTH

```
length(string)
```

- 命令说明：返回字符串string的长度。

 说明 MySQL中一个中文字符length为3，分析型数据库MySQL版中为1。

- 返回值类型：BIGINT。
- 示例：

```
SELECT length('aliyun')
```

返回值如下：

```
+-----+
| _col0 |
+-----+
| 6 |
```

LOWER

```
lower(string)
```

- 命令说明：将字符串string中的字母转换为小写。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT lower('Aliyun');
```

返回值如下：

```

+-----+
|_col0  |
+-----+
| aliyun

```

UPPER

```
upper(string)
```

- 命令说明：将字符串string中的字母转换为大写。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT lower('Aliyun');
```

返回值如下：

```

+-----+
|_col0  |
+-----+
| ALIYUN

```

LPAD

```
lpad(string, size, padstring)
```

- 命令说明：
 - 将字符串string左边拼接padstring直到长度达到size，并返回填充后的字符串。
 - 如果size比string长度小，则截断。size不能为负数，padstring非空。

 **说明** MySQL中一个中文字符length为3，分析型数据库MySQL版中为1。

- 返回值类型：VARCHAR。
- 示例：

```
SELECT Lpad('Aliyun',9,'#');
```

返回值如下：

```
+-----+
|_col0  |
+-----+
|###Aliyun
```

RPAD

```
rpad(string, size, padstring)
```

- 命令说明：
 - 将字符串string右边拼接padstring直到长度达到size，并返回填充后的字符串。
 - 如果size比string长度小，则截断。size不能为负数，padstring非空。

 说明 MySQL中一个中文字符length为3，分析型数据库MySQL版中为1。

- 返回值类型：VARCHAR。
- 示例：

```
SELECT rpad('Aliyun',9,'#');
```

返回值如下：

```
+-----+
|_col0  |
+-----+
|Aliyun###
```

TRIM/LTRIM/RTRIM

```
trim(string)
ltrim(string)
rtrim(string)
```

- 命令说明：
 - `trim(string)`：删除字符串string前后所有的空格。
 - `ltrim(string)`：删除字符串string所有前导空格。
 - `rtrim(string)`：删除字符串string所有后置空格。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Trim(' 135 544 '), Ltrim(' 135 544 '), Rtrim(' 135 544 ');
```

返回值如下：

```

+-----+-----+-----+
|_col0 |_col1 |_col2 |
+-----+-----+-----+
| 135 544 | 135 544 | 135 544 |

```

REPLACE

```

replace(string, search)
replace(string, search, replace)

```

- 命令说明：
 - `replace(string, search)`：删除字符串string中的所有search子串。
 - `replace(string, search, replace)`：将字符串string中所有子串search替换为replace。
- 返回值类型：VARCHAR。
- 示例：

```

SELECT REPLACE('helloworld', 'world'),
       REPLACE('helloworld', 'world', 'World!!!'),
       REPLACE('helloworld', 'notFound', 'World');

```

返回值如下：

```

+-----+-----+-----+
|_col0 |_col1 |_col2 |
+-----+-----+-----+
| hello | hello World!!! | helloworld |

```

REVERSE

```
reverse(string)
```

- 命令说明：返回string逆序后的字符串。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Reverse('123456'), Reverse(''), Reverse(Cast (NULL AS VARCHAR));
```

返回值如下：

```

+-----+-----+-----+
|_col0 |_col1 |_col2 |
+-----+-----+-----+
| 654321 | | NULL |

```

SPLIT

```
split(string, delimiter)
split_part(string, delimiter, index)
split_to_map(string, entryDelimiter, keyValueDelimiter)
```

- 命令说明：
 - `split(string, delimiter)`：将字符串string按分隔符delimiter进行分隔，并返回数组。
 - `split_part(string, delimiter, index)`：将字符串string按分隔符delimiter分隔，并返回分隔后数组下标为index的子串，index以1开头，如果大于字段数则返回null。
 - `split_to_map(string, entryDelimiter, keyValueDelimiter)`：通过entryDelimiter和keyValueDelimiter拆分字符串并返回map。entryDelimiter将字符串分解为key-value对，keyValueDelimiter将每对分隔成key、value。
- 返回值类型：VARCHAR或map<varchar, varchar>。
- 示例：

```
SELECT Split('1#2#3', '#'), Split('#1#2#3#', '#'),
       Split('123', '#');
```

返回值如下：

```
|_col0 | _col1 | _col2 |
+-----+-----+-----+
|[1, 2, 3] |[1, 2, 3, #] |[123] |
```

```
SELECT Split_part('A#B#C', '#', 2),
       Split_part('A#B#C', '#', 4);
```

返回值如下：

```
+-----+-----+
|_col0 | _col1 |
+-----+-----+
| B    | NULL  |
```

```
SELECT Split_to_map('k1:v1,k2:v2', ',', ':'),
       Split_to_map(';', ',', ':');
```

返回值如下：

```

+-----+-----+
|_col0   |_col1   |
+-----+-----+
|{k1=v1, k2=v2} |{}      |

```

POSITION

```
position(substring IN string)
```

- 命令说明：返回字符串中子字符串的第一次出现的起始位置，位置从1开始，如果未找到则返回0。
- 返回值类型：BIGINT。
- 示例：

```
SELECT position('helloworld', 'o'),
       position('helloworld', 'or'),
       position('helloworld', 'x');
```

返回值如下：

```

+-----+-----+-----+
|_col0  |_col1  |_col2  |
+-----+-----+-----+
| 5 | 7 | 0 |

```

CHAR

```
char(N,N,..)
```

- 命令说明：返回每个数字代表的字符组成的字符串。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT CHAR(97,110,97,108,121,116,105,99,100,98);
```

返回值如下：

```

+-----+
|_col0  |
+-----+
|analyticdb|

```

HEX

```
hex(bigint)
```

- 命令说明：返回16进制字符串。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Hex(10),  
       Hex(1234);
```

返回值如下：

```
+-----+-----+  
|_col0|_col1|  
+-----+-----+  
| a  |4d2 |
```

INITCAP

```
initcap(string)
```

- 命令说明：输入字符串首字符转大写。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Initcap('the soap'),  
       Initcap('the,soap'),  
       Initcap('thesoap');
```

返回值如下：

```
+-----+-----+-----+  
|_col0|_col1|_col2|  
+-----+-----+-----+  
| The Soap | The,Soap | Thesoap |
```

INSTR

```
instr(string, substring)  
instr(string, substring, position)  
instr(string, substring, position, occurrence)
```

- 命令说明：

`instr(string, substring)`：返回string中匹配substring的第一个位置信息。

`instr(string, substring, position)`：从string字符的第position位置开始搜索，返回string中匹配substring的位置信息。

`instr(string, substring, position, occurrence)`：从string字符的第position位置开始搜索第occurrence次出现的位置，返回string中匹配substring的位置信息。

- 返回值类型：BIGINT。
- 示例：

```
SELECT Instr('CORPORATE FLOOR', 'OR'),
       Instr('CORPORATE FLOOR', 'OR', 3),
       Instr('CORPORATE FLOOR', 'OR', 3, 2);
```

返回值如下：

```
+-----+-----+-----+
|_col0 |_col1 |_col2 |
+-----+-----+-----+
| 2    | 5    | 14   |
```

ASCII

`ASCII(string)`

- 命令说明：返回字符或者字符串最左边字符对应的ASCII值。
- 返回值类型：BIGINT。
- 示例：

```
SELECT Ascii('d'),
       Ascii('dx'),
       Ascii('1');
```

返回值如下：

```
+-----+-----+-----+
|_col0 |_col1 |_col2 |
+-----+-----+-----+
| 100  | 100  | 49   |
```

MID

`MID(string, start, length)`

- 命令说明：作用同`substr(string, start, length)`。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Mid('helloworld', 6, 3),  
       Mid('helloworld', -6, 2);
```

返回值如下：

```
+-----+-----+  
|_col0 |_col1 |  
+-----+-----+  
| wor  | ow   |
```

REPEAT

```
repeat(string,int)
```

- 命令说明：返回重复多次的字符串。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Repeat('a', 3),  
       Repeat('xyz', 2),  
       Repeat(Cast(NULL AS VARCHAR), 2);
```

返回值如下：

```
+-----+-----+-----+  
|_col0 |_col1 |_col2 |  
+-----+-----+-----+  
| aaa  | xyzxyz | NULL  |
```

STRPOS

```
strpos(string, substring)
```

- 命令说明：返回字符串中子字符串的第一次出现的起始位置，位置从1开始，如果未找到则返回0。
- 返回值类型：BIGINT。
- 示例：

```
SELECT Strpos('helloworld', 'o'),  
       Strpos('helloworld', 'or'),  
       Strpos('helloworld', 'x');
```

返回值如下：

```
+-----+-----+-----+
|_col0|_col1|_col2|
+-----+-----+-----+
| 5 | 7 | 0 |
```

SUBSTR/SUBSTRING

```
substr(string, start)
substr(string, start, length)
```

- 命令说明：
 - `substr(string, start)`：返回从`start`位置开始到字符串结束的子串。位置从1开始，如果`start`为负数，则起始位置从字符串的末尾开始倒数。
 - `substr(string, start, length)` 返回从`start`位置开始长度为`length`的子串，位置从1开始。如果`start`为负数，则起始位置从字符串的末尾开始倒数。

 说明 MySQL中一个中文字符的`length`为3，分析型数据库MySQL版中为1。

- 返回值类型：VARCHAR。
- 示例：

```
SELECT Substr('helloworld', 6),
       Substr('helloworld', 6, 3),
       Substr('helloworld', -6),
       Substr('helloworld', -6, 2),
       Substring('helloworld', -6, 2);
```

返回值如下：

```
+-----+-----+-----+-----+-----+
|_col0|_col1|_col2|_col3|_col4|
+-----+-----+-----+-----+-----+
| world | wor  | oworld | ow  | ow  |
```

TRANSLATE

```
translate(expr, from_string, to_string)
```

- 命令说明：将`expr`字符串中，符合`from_string`的字符，替换为`to_string`。
- 返回值类型：VARCHAR。
- 示例：

```
SELECT Translate('acbd', 'ab', 'AB'),  
       Translate('acbdaa', 'ab', 'AB'),  
       Translate('acbd', 'abc', 'A'),  
       Translate('acbd', 'abc', '');
```

返回值如下：

```
+-----+-----+-----+-----+  
|_col0 |_col1 |_col2 |_col3 |  
+-----+-----+-----+-----+  
|AcBd |AcBdAA|Ad  | d  |
```