

# 时序数据库 CTSDB

## InfluxQL 兼容性



腾讯云

## 【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

## 【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

# 文档目录

InfluxQL 兼容性

数据库管理

Schema 查询

数据类型

数据查询

InfluxQL 函数

# InfluxQL 兼容性

## 数据库管理

最近更新时间：2024-08-30 10:43:51

时序数据库 CTSDB 3.0版默认兼容 [开源 InfluxDB v1.7 版本](#)。本章将说明时序数据库 CTSDB 3.0版创建数据库、下载数据并将数据写入数据库，对开源 InfluxDB V1.7 的兼容情况。

### 数据库管理

全新购买的时序数据库 CTSDB 3.0版实例没有数据库，需手动创建数据库。数据库的创建与删除相关命令使用语义请参见下表。其中，✓表示兼容 InfluxDB，✗表示不兼容。更多信息，请参见 [Sample data](#)。

命令	作用	兼容性	语义
CREATE DATABASE	创建一个新的数据库	✓	<ul style="list-style-type: none"> <li>当前 CREATE DATABASE 仅支持配置如下参数，其它参数暂不支持。语义格式如下所示： <pre>CREATE DATABASE &lt;database_name&gt; [WITH [DURATION &lt;duration&gt;[默认无限制]&gt;]]</pre> <ul style="list-style-type: none"> <li>database_name: 指数据库库名。</li> <li>DURATION: 数据过期删除时间 (TTL, Time To Live), 单位: 天, 取值为非零整数, 选填参数。如果不启用数据过期, 不配置该参数, 如需启用数据过期, 该数据库中的数据达到过期时间后将被自动删除清理, 请谨慎配置! 例如 DURATION 180d, 表示该数据库中的数据达到180天即过期自动删除。</li> </ul> </li> <li>创建一个名为 test_tsdb1 的数据库, 该数据库的数据永久保留: <pre>CREATE DATABASE test_tsdb1</pre>。</li> <li>创建一个名为 test_tsdb2 的数据库, 该数据库启用数据过期, 设置数据被保留30天: <pre>CREATE DATABASE test_tsdb2 WITH DURATION 30d</pre> </li> </ul>
DROP DATABASE	删除数据库	✓	<p>执行语义如下所示:</p> <pre>DROP DATABASE &lt;database_name&gt;</pre>
DROP	删除数据库时间序列数	✗	-

SERIES	据		
DROP MEASUREMENT	删除测量值	✓	<p><b>执行语义:</b></p> <pre>DROP MEASUREMENT &lt;measurement_name&gt;</pre>
DROP SHARD	删除数据分片	✗	-
DELETE	删除数据点，可以根据时间范围和查询条件来删除一个或多个数据点	✓	<ul style="list-style-type: none"> <li><b>执行语义:</b> <pre>DELETE FROM &lt;measurement_name&gt; WHERE [&lt;tag_key&gt;='&lt;tag_value&gt;']   [&lt;time interval&gt;]</pre> </li> <li>如下以删除名为 <code>cpu_load</code> 的测量类型中时间戳在某个时间范围内的所有数据点为例，演示其执行方式。 <pre>DELETE FROM cpu_load WHERE time &gt;= '2023-05-20T00:00:00Z' AND time &lt; '2023-05-21T00:00:00Z'</pre> </li> </ul>
CREATE RETENTION POLICY	创建数据保留策略的命令	✗	-
ALTER RETENTION POLICY	修改保留策略	✗	-
DROP RETENTION POLICY	删除保留策略	✗	-

## 写入协议

InfluxDB Line Protocol 是 InfluxDB 数据库使用的一种数据格式，用于将数据以文本形式写入 InfluxDB 数据库。该协议使用逗号分隔值 (CSV) 格式，其中包含时间戳、测量标识符、标签和字段。您可以使用该协议将数据写入 InfluxDB 数据库，并从中检索和查询数据。

协议	协议说明	是否兼容	格式
----	------	------	----

<p>InfluxDB Line Protocol</p>	<p>InfluxDB Line Protocol 是一种基于文本的数据格式，用于将数据点写入 InfluxDB 数据库。每个数据点都由一个单独的行表示，行中包含时间戳和测量值名称、标签集和字段集的值。</p>	<p>✓</p>	<p>每个数据点都由一个单独的行表示，行中包含时间戳和测量值名称、标签集和字段集的值。</p> <pre data-bbox="890 286 1481 741"> &lt;measurement&gt; [, &lt;tag_key&gt;=&lt;tag_value&gt; [, &lt;tag_key&gt;= &lt;tag_value&gt;]] &lt;field_key&gt;= &lt;field_value&gt; [, &lt;field_key&gt;= &lt;field_value&gt;] [&lt;timestamp&gt;] </pre> <p>更多信息，请参见官网 <a href="#">InfluxDB line protocol reference</a>。</p>
<p>Prometheus</p>	<p>Prometheus 是一种开源的监控系统和时间序列数据库，用于收集、存储和查询各种类型的指标数据。InfluxDB 提供了一个兼容 Prometheus 的 HTTP API 端点，使得 Prometheus 可以将其数据写入 InfluxDB 中。</p>	<p>✓</p>	<p>启用 Prometheus 远程读写 API 与 InfluxDB:</p> <ol style="list-style-type: none"> <li>创建目标数据库             <p>在 InfluxDB 实例中，通过 <a href="#">库管理</a> 或参照 <a href="#">管理数据库</a>，创建一个数据库来存储从 Prometheus 发送的数据。在下面提供的示例中以 prometheus_test 举例，用作数据库名称</p> <pre data-bbox="935 1263 1481 1357"> CREATE DATABASE "prometheus_test" </pre> </li> <li>请将 URL 值添加到 Prometheus 配置文件中的以下设置：             <p>username、password：分别为实例的访问账号与密码。              influxdb_ip：指实例内网 IP 地址。              db：指数据库库名，例如 prometheus_test。</p> <pre data-bbox="935 1733 1481 2078"> remote_write: - url: "http://influxdb_ip: 8086/api/v1/prom/wri te? db=prometheus_test&amp;u </pre> </li> </ol>

```
=username&p=password
"

remote_read:
  - url:
    "http://influxdb_ip:
    8086/api/v1/prom/rea
    d?
    db=prometheus_test&u
    =username&p=password
    "
```

具体如何配置文件以及如何解析 Prometheus 指标，请参见 [Prometheus endpoints support in InfluxDB](#)。

## 下载并写入数据

下载并写入符合 InfluxDB Line Protocol 协议的时序数据，请参见 [使用数据库实例](#)。更多信息，请参见 [Download and write the data to InfluxDB](#)。

# Schema 查询

最近更新时间：2024-05-16 17:00:42

SQL	含义	兼容性	语义
SHOW DATABASES	查询所有数据库	✓	SHOW DATABASES
SHOW SERIES	查询符合指定条件的时间序列数据	✓	<ul style="list-style-type: none"> <li> <b>执行语义：</b> <pre>SHOW SERIES [FROM_clause] [WHERE &lt;tag_key&gt; &lt;operator&gt; [ '&lt;tag_value&gt;'   &lt;regular_expression&gt;]] [LIMIT_clause] [OFFSET_clause] [ORDER BY ASC/DESC]</pre> <ul style="list-style-type: none"> <li> <b>FROM_clause:</b> 必须参数，用于指定要查询的测量名称。语法为：FROM &lt;measurement_name&gt;。例如：FROM cpu_usage。 </li> <li> <b>WHERE &lt;tag_key&gt; &lt;operator&gt; [ '&lt;tag_value&gt;'   &lt;regular_expression&gt;]:</b> 可选参数，用于指定查询条件。其中，&lt;tag_key&gt;为标签名称，&lt;operator&gt;为比较运算符，&lt;tag_value&gt;或&lt;regular_expression&gt;为比较值。可以使用多个 WHERE 子句以 AND 或 OR 组合多个条件。例如：WHERE host='server1' AND region='us-west'。 </li> <li> <b>LIMIT_clause:</b> 可选参数，用于指定查询结果的最大行数。语法为：LIMIT &lt;n&gt;，其中&lt;n&gt;为最大行数。例如：LIMIT 10。 </li> <li> <b>OFFSET_clause:</b> 可选参数，用于指定查询结果的偏移量。语法为：OFFSET &lt;n&gt;，其中&lt;n&gt;为偏移量。例如：OFFSET 5。 </li> <li> <b>ORDER BY ASC/DESC:</b> 可选参数，用于指定查询结果的排序方式。语法为：ORDER BY &lt;field&gt; ASC/DESC，其中&lt;field&gt;为要排序的字段名称，ASC表示升序，DESC表示降序。例如：ORDER BY time DESC。 </li> </ul> </li> <li> <b>执行示例：</b> </li> </ul>



			<pre>SHOW SERIES FROM yottadb_partition_replicas_num_lzl where cluster_name = 'yottadb-capd-gz-dataplane6' ORDER BY ASC LIMIT 5 OFFSET 5</pre>
<p>SHOW MEASUREMENTS</p>	<p>获取当前 InfluxDB 实例中所有可用的测量 (measurement) 名称</p>	<p>✓</p>	<ul style="list-style-type: none"> <li>● <b>执行语义:</b> <pre>SHOW MEASUREMENTS [WITH MEASUREMENT &lt;operator&gt; ['&lt;measurement_name&gt;'   &lt;regular_expression&gt;]] [WHERE &lt;tag_key&gt; &lt;operator&gt; ['&lt;tag_value&gt;'   &lt;regular_expression&gt;]] [LIMIT_clause] [OFFSET_clause] [ORDER BY ASC/DESC]</pre> <ul style="list-style-type: none"> <li>○ <b>SHOW MEASUREMENTS:</b> 指定查询的目标是测量值。</li> <li>○ <b>WITH MEASUREMENT:</b> 可选参数, 用于更精细地筛选测量值。您可以使用运算符 (operator), 测量值名称或正则表达式来指定筛选条件。</li> <li>○ <b>WHERE:</b> 可选参数, 用于根据标签 (tag) 筛选测量值。您可以使用标签键 (tag key)、运算符, 标签值 (tag value) 或正则表达式来指定筛选条件。</li> <li>○ <b>LIMIT_clause:</b> 可选参数, 用于限制查询结果的数量。</li> <li>○ <b>OFFSET_clause:</b> 可选参数, 用于指定查询结果的偏移量。</li> <li>○ <b>ORDER BY ASC/DESC:</b> 可选参数, 用于指定查询结果的排序方式。ASC 表示升序排列, DESC 表示降序排列。</li> </ul> </li> <li>● 查找所有名为 sensor_data 的测量值, 其中标签 location 等于 room1, 并且标签 sensor_type 匹配正则表达式/temperature/的结果。 <pre>SHOW MEASUREMENTS WITH MEASUREMENT = 'sensor_data' WHERE location = 'room1' AND sensor_type =~ /temperature/</pre> </li> </ul>
<p>SHOW TAG KEYS</p>	<p>查询 InfluxDB 数据库中标签键 (tag key) 的查询语句</p>	<p>✓</p>	<ul style="list-style-type: none"> <li>● <b>执行语义:</b> <pre>SHOW TAG KEYS [FROM_clause] [WHERE &lt;tag_key&gt; &lt;operator&gt; ['&lt;tag_value&gt;'   &lt;regular_expression&gt;]] [LIMIT_clause] [OFFSET_clause] [ORDER BY ASC/DESC] [SIMIT_clause] [SOFFSET_clause]</pre> <ul style="list-style-type: none"> <li>○ <b>SHOW TAG KEYS:</b> 标识查询的目标是标签键。</li> </ul> </li> </ul>

			<ul style="list-style-type: none"> <li>○ FROM_clause: 可选参数, 用于指定查询的测量值。如果省略此参数, 则将查询所有测量值的标签键。</li> <li>○ WHERE: 可选参数, 用于根据标签 (tag) 筛选标签键。您可以使用标签键、运算符、标签值或正则表达式来指定筛选条件。</li> <li>○ LIMIT_clause: 可选参数, 用于限制查询结果的数量。</li> <li>○ OFFSET_clause: 可选参数, 用于指定查询结果的偏移量。</li> <li>○ ORDER BY ASC/DESC: 可选参数, 用于指定查询结果的排序方式。ASC表示升序排列, DESC表示降序排列。</li> <li>○ SIMIT_clause: 可选参数, 用于限制返回结果中每个标签键返回的标签值的数量。</li> <li>○ SOFFSET_clause: 可选参数, 用于指定返回结果中每个标签键返回的标签值的偏移量。</li> </ul> <ul style="list-style-type: none"> <li>● 执行示例: 该查询将返回包含  <pre>cluster_name = 'yottadb-capd-gz-dataplane6'</pre>                     条件的测量值的所有标签键, 并按字母顺序排列。结果集将限制为前5个标签键, 并从第2个标签键开始返回。对于每个标签键, 仅返回前两个标签值, 并从第三个标签值开始返回。  <pre>SHOW TAG KEYS WHERE cluster_name = 'yottadb-capd-gz-dataplane6' ORDER BY ASC LIMIT 5 OFFSET 1 SLIMIT 2 SOFFSET 2</pre> </li> </ul>
<p>SHOW TAG VALUES</p>	<p>查询测量值的特定标签键的所有标签值</p>	<p>✓</p>	<ul style="list-style-type: none"> <li>● 执行语义:  <pre>SHOW TAG VALUES [FROM_clause] WITH KEY [                      [&lt;operator&gt; "&lt;tag_key&gt;"                        &lt;regular_expression&gt;]   [IN ("                      &lt;tag_key1&gt;","&lt;tag_key2")]] [WHERE                      &lt;tag_key&gt; &lt;operator&gt; ['&lt;tag_value&gt;'                        &lt;regular_expression&gt;]] [LIMIT_clause]                      [OFFSET_clause] [ORDER BY ASC/DESC] [S                      IMIT_clause] [SOFFSET_clause]</pre>                     与 SHOW TAG KEYS 相比, 增加参数 WITH KEY, 指定要检索标签值的标签键名称。                 </li> <li>○ [&lt;operator&gt; "&lt;tag_key&gt;"   &lt;regular_expression&gt;]: 可选参数, 用于指定要检索标签值的标签键名称。您可以使用等于号(=)、不</li> </ul>

			<p>等于号(!=)、正则表达式或 IN 运算符来匹配标签键。如果省略此参数，则返回所有标签键的标签值。</p> <ul style="list-style-type: none"> <li>○ [IN ("<b>&lt;tag_key1&gt;</b>","<b>&lt;tag_key2&gt;</b>")]: 可选参数，用于指定要检索标签值的标签键名称列表。如果省略此参数，则返回所有标签键的标签值。</li> <li>○ [WHERE <b>&lt;tag_key&gt;</b> <b>&lt;operator&gt;</b> [<b>&lt;tag_value&gt;</b>'   <b>&lt;regular_expression&gt;</b>]]: 可选参数，用于指定过滤条件，只返回符合条件的标签值。您可以使用等于号(=)、不等于号(!=)、正则表达式、IN运算符、大于号(&gt;)、小于号(&lt;)、大于等于号(&gt;=)或小于等于号(&lt;=)来匹配标签值。如果省略此参数，则返回所有符合条件的标签值。</li> </ul> <ul style="list-style-type: none"> <li>● 执行示例： 该查询将返回三个标签键（<b>account_id</b>、<b>cluster_display_name</b> 和 <b>cluster_name</b>）的所有标签值，按升序排序，并且每个标签键返回的标签值数量限制为2，每个标签键返回的标签值偏移量为1。此外，该查询还限制了返回结果集中的行数为8，并且从第二个标签值开始返回结果。</li> </ul> <pre>SHOW TAG VALUES WITH KEY IN (account_id, cluster_display_name, cluster_name) ORDER BY ASC LIMIT 8 OFFSET 1 SLIMIT 2 SOFFSET 1</pre>
<p>SHOW FIELD KEYS</p>	<p>显示指定测量值的所有字段键。每个字段键都包括名称、数据类型和是否为标记字段的信息。</p>	<p>✓</p>	<ul style="list-style-type: none"> <li>● 执行语义：</li> </ul> <pre>SHOW FIELD KEYS [FROM &lt;measurement_name&gt;] [LIMIT_clause] [OFFSET_clause] [ORDER BY ASC/DESC] [SLIMIT_clause] [SOFFSET_clause]</pre> <p>参数含义，请参照 SHOW MEASUREMENTS，其中：</p> <ul style="list-style-type: none"> <li>○ <b>SIMIT_clause</b>: 限制每个字段键返回的值的数量。</li> <li>○ <b>SOFFSET_clause</b>: 指定每个字段键返回的值的偏移量。</li> </ul> <ul style="list-style-type: none"> <li>● 执行示例： 该查询语句将从 <b>yottadb_partition_replicas_num</b> 表中获取字段键，并按升序排序。然后，它将返回从第5行开始的4行结果。具体来说，它将返回第5行到第8行的结果，每行包含一个字段键。</li> </ul> <pre>SHOW FIELD KEYS FROM yottadb_partition_replicas_num ORDER BY ASC LIMIT 4 OFFSET 4</pre>

<p>SHOW SERIES CARDINALITY</p>	<p>显示数据库中所有时序数据的数量，并支持按测量、标签键或标签值进行筛选和分组，以及限制返回结果的数量和偏移量。</p>	<p>✓</p>	<ul style="list-style-type: none"> <li>● 执行语义：             <pre>SHOW SERIES CARDINALITY [FROM &lt;measurement_name&gt;] [where_clause] [group_by_clause] [LIMIT_clause] [OFFSET_clause] [SIMIT_clause] [SOFFSET_clause]</pre> <ul style="list-style-type: none"> <li>○ FROM &lt;measurement_name&gt;: 可选参数，指定要查询的测量名称。如果未指定，则将查询所有测量。</li> <li>○ where_clause: 可选参数，指定要筛选的标签键和标签值。格式为：                     <pre>WHERE &lt;tag_key&gt;=&lt;tag_value&gt;' [AND &lt;tag_key&gt;=&lt;tag_value&gt;']</pre> </li> <li>○ group_by_clause: 可选参数，按标签键对结果进行分组。格式为： <code>GROUP BY &lt;tag_key&gt;</code></li> <li>○ LIMIT_clause: 可选参数，限制返回的结果数。格式为： <code>LIMIT &lt;limit&gt;</code></li> <li>○ OFFSET_clause: 可选参数，从结果集的指定偏移量之后开始返回结果。 <code>OFFSET &lt;offset&gt;</code></li> <li>○ SIMIT_clause: 可选参数，限制每个分组返回的结果数。格式为： <code>SLIMIT &lt;slimit&gt;</code></li> <li>○ SIMIT_clause: 可选参数，从每个分组的指定偏移量之后开始返回结果。格式为：                     <pre>SOFFSET &lt;soffset&gt;</pre> </li> </ul> </li> <li>● 执行示例：             该示例将从 <code>cpu_usage</code> 测量中获取所有标签 <code>region</code> 为 <code>us-west</code>、<code>host</code> 为 <code>server-1</code> 的时序数据，并按标签 <code>cpu_type</code> 进行分组。它还将限制返回的结果数为10，从第 21 行开始返回结果，并限制每个分组返回的结果数为 5，从每个分组的第3行开始返回结果。             <pre>SHOW SERIES CARDINALITY FROM cpu_usage WHERE region='us-west' AND host='server-1' GROUP BY cpu_type LIMIT 10 OFFSET 20 SLIMIT 5 SOFFSET 2</pre> </li> </ul>
<p>SHOW FIELD KEY CARDINALITY</p>	<p>显示数据库中所有测量的字段键数量</p>	<p>✗</p>	<p>—</p>

SHOW MEASURE MENT CARDINALITY	显示数据库中所有测量的数量	×	—
SHOW TAG KEY CARDINALITY	显示数据库中所有测量的标签键数量	×	—
SHOW TAG VALUES CARDINALITY	显示数据库中指定标签键的标签值数量	×	
SHOW RETENTION POLICIES	显示指定数据库中的保留策略	×	
SHOW SHARDS	显示指定数据库中的所有分片	×	

# 数据类型

最近更新时间：2024-08-30 10:43:51

时序数据库 CTSDB 3.0版默认支持的数据类型，请参见下表。更多的使用示例，请参见 [数据类型](#)。

数据类型	字段	取值	描述
Float (浮点型)	Field values	64 位	用于描述时间序列数据中的数值型数据，可以用于存储测量值、传感器数据、网络流量、CPU 使用率等等。
Integer (整型)	Field values	有符号 64 位整数 (-9223372036854775808 到 9223372036854775807)	用于描述时间序列数据中的计数型数据，可以存储计数器、事件发生次数、用户访问次数等等，这些数据通常是非负整数。
String (字符型)	Measurements, tag keys, tag values, field keys, field values	长度限制 64KB。	用于描述时间序列数据中的文本型数据，可以作为Measurements、Tag Keys、Tag Values、Field Keys 和 Field Values 字段类型，可以用于存储设备名称、日志消息、状态信息等等。
Boolean (布尔型)	Field values	TRUE 或 FALSE	用于描述只有两个可能取值的数据，即"真" (True) 和"假" (False)。在数据分析和编程中，布尔型常用于表示逻辑条件、开关状态、是否满足某个条件等情况。
Timestamp (时间型)	Timestamps	Unix 纳秒时间戳。 <ul style="list-style-type: none"> <li>最小有效时间戳是：9223372036854775806或1677-09-21T00:12:43.145224194Z。</li> <li>最大有效时间戳是：9223372036854775806或2262-04-11T23:47:16.854775806Z。</li> </ul>	表示从某个特定时间点开始的时间间隔。在 InfluxDB 中，时间戳以 UTC 格式存储，并且可以精确到纳秒级别。

 **注意:**

字段值可以是浮点数、整数、字符串或布尔值。同一分片内字段值类型必须相同，不同分片之间使用不同的数据类型可能会导致数据不一致或错误，请尽可能避免使用。

# 数据查询

最近更新时间：2024-08-30 10:43:51

## Select 查询

时序数据库 CTSDB 3.0在使用 select 查询数据时，相比开源 InfluxDB，存在如下差异。各个子句的语义，请参见 [子句及表达式](#)。

- `into_clause` 子句用于将数据写入指定的测量（measurement）中，当前不支持。
- `timezone_clause` 子句用于在查询结果中将时间戳转换为指定时区的本地时间，当前已支持。
- `sql_order_by_clause` 为时序数据库 InfluxDB 版新增的子句，它允许您使用 SQL 风格的表达式对查询结果进行排序。

开源 InfluxDB	时序数据库 CTSDB 3.0版
<pre>select_stmt = "SELECT" fields from_clause [ into_clause ] [ where_clause ] [ group_by_clause ] [ order_by_clause ] [ limit_clause ] [ offset_clause ] [ slimit_clause ] [ soffset_clause ] [ timezone_clause ]</pre>	<pre>select_stmt = "SELECT" fields from_clause [ where_clause ] [ group_by_clause ] [ order_by_clause ] [ limit_clause ] [ offset_clause ] [ slimit_clause ] [ soffset_clause ] [ sql_order_by_clause ]</pre>
<p>SELECT_clause FROM ( SELECT_clause FROM ( SELECT_statement ) [...]) [...]</p>	<p>完全兼容</p>

如下查询示例，以 [使用数据库实例](#) 中的关于车辆的时序数据为例。每行代表一个车辆的信息，包括车辆的 id、所在城市、类型、速度和温度等。时间戳位于每行末尾，表示这些信息是在对应的时间戳时刻记录的。

查询数据类别	查询语句
查询全部数据	<pre>select * from car</pre>
分页获取部分数据	<pre>select * from car limit 3 offset 2</pre>
聚合函数作为复杂表达式的参数	<pre>select (max(speed) + max(temp) - min(speed)) * count(speed) / 20 % 5 as f1, floor(min(speed)) &amp; ceil(max(temp))   round(min(speed)) ^ round(max(temp)) as f2, sqrt(pow(abs(max(speed)*min(speed)), 2)) as f3 from car group by time(3s), type fill(none)</pre>



聚合计算	<pre>select count(speed), mean(speed), count(temp), percentile(temp, 50) from car group by time(3s), type fill(none) limit 1 offset 1 slimit 2 soffset 1</pre>
混合一个 Selectors 与 field, group by time intervals	<pre>select max(speed)*6, speed*10 from car where time &lt;= 2000000000ns group by time(1s)</pre>
混合一个 Selectors 与 field, group by tag	<pre>select max(speed), pow(speed, temp/100.0) from car group by type</pre>
仅 fields, group by tag	<pre>select *, temp, sqrt(pow(abs(max(speed)), 3)) FROM car group by city</pre>

## InfluxQL 子句及表达式

InfluxQL 是一种类似于 SQL 的查询语言，用于与 InfluxDB 中的数据进行交互。下表详细列出了时序数据库版 InfluxDB 相比开源 InfluxDB 在探索数据的查询语法方面的支持程度。具体如何使用，请参见 [Explore data using InfluxQL](#)。

开源 InfluxDB	时序数据库 InfluxDB 版	语义
with_tag_clause = "WITH KEY" ("=" tag_key   "!=" tag_key   "=~" regex_lit   "IN (" tag_keys ")") .	支持	<p>该查询语义通过正则表达式指定一个或多个标签键。具体信息，请参见 <a href="#">Regular expressions</a>。</p> <ul style="list-style-type: none"> <li>• <b>WITH KEY</b> 是一个关键字，用于指示后面的子句将包含一个或多个标签键和值。</li> <li>• <b>=、!=、=~和 IN</b>：这些是可用于匹配标签键和值的运算符。 <ul style="list-style-type: none"> <li>○ =、!=和=~分别表示相等、不相等和正则表达式匹配。</li> <li>○ IN 运算符用于指定标签键的值应该是给定列表中的一个。</li> </ul> </li> <li>• <b>tag_key</b>：用于匹配的标签键的名称。</li> <li>• <b>regex_lit</b>：用于匹配标签键的值。</li> <li>• <b>tag_keys</b>：用于指定要匹配的多个标签键。</li> </ul> <p><b>示例：</b><pre>SELECT * FROM http_requests_total WITH KEY (method != 'POST' AND method =~ 'PUT.*')   status_code IN ('200', '404')</pre></p>

<p>with_measurement_clause = "WITH MEASUREMENT" ( "=" measurement   "=~" regex_lit ) .</p>		<p>该语义使用 WITH MEASUREMENT 子句来筛选指定的测量值 ( measurement ) ，其中 regex_lit 指定了匹配的测量值。</p>
<p>where_clause = "WHERE" expr .</p>		<p>请参见 <a href="#">The WHERE clause</a> 。</p>
<p>var_ref = measurement .</p>		<p>表示一个变量引用，将变量与测量值进行关联。</p>
<p>user_name = identifier .</p>		<p>定义一个名为 user_name 的字段，其值为 identifier 。</p>
<p>unary_expr = "(" expr ")"   var_ref   time_lit   string_lit   int_lit   float_lit   bool_lit   duration_lit   regex_lit .</p>		<p>unary_expr 的表达式，它可以由以下几种子表达式组成：</p> <ul style="list-style-type: none"> <li>• (expr): 表示一个由括号包围的表达式，其中 expr 是另一个表达式。</li> <li>• var_ref: 表示一个变量引用，用于引用 InfluxDB 中的测量值、字段或标签。</li> <li>• time_lit: 表示一个时间字面量，用于表示时间戳。</li> <li>• string_lit: 表示一个字符串字面量，用于表示文本数据。</li> <li>• int_lit: 表示一个整数字面量，用于表示整型数据。</li> <li>• float_lit: 表示一个浮点数字面量，用于表示浮点型数据。</li> <li>• bool_lit: 表示一个布尔字面量，用于表示真或假。</li> <li>• duration_lit: 表示一个时间段字面量，用于表示持续时间。</li> <li>• regex_lit: 表示一个正则表达式字面量，用于匹配文本数据中的模式。</li> </ul>
<p>to_clause = "TO" user_name .</p>		<p>表示在查询 InfluxDB 中的数据时，将查询结果发送到指定的用户。其中，user_name 参数是指接收查询结果的用户的名称。</p>
<p>timezone_clause = tz(string_lit) .</p>	<p>支持</p>	<p>用于时序数据库中的时间戳数据，以指定时间戳的时区信息。如下示例，指定了时区信息为亚洲/上海的数据。</p> <pre>SELECT time, temperature FROM sensor_data WHERE time &gt;= '2023-07-10T00:00:00.000Z' AND time &lt; '2023-07-11T00:00:00.000Z' timezone_clause = tz('Asia/Shanghai');</pre>
<p>tag_keys = tag_key { "," tag_key } .</p>	<p>支持</p>	<p>指定要在查询结果中返回的标签键列表。</p> <ul style="list-style-type: none"> <li>• 逗号表示标签键之间的分隔符。</li> </ul>

		<ul style="list-style-type: none"> <li>如下查询语句只返回标签 tag3 的值等于 value3，并且时间戳在过去的一小时内的数据。标签键 tag1 和 tag2，只返回这两个标签键的值。</li> </ul> <pre>SELECT * FROM measurement1 WHERE tag3='value3' AND time &gt; now() - 1h TAGS tag_keys=tag1,tag2</pre>
tag_key = identifier .		<p>表示标签键由一个标识符组成。标签是存储在 InfluxDB 中的每个时间序列的元数据，由键值对组成。您可以使用标识符来命名标签键。例如：</p> <pre>measurement1,tag1=value1,tag2=value2 field1=1,field2=2</pre>
subscription_name = identifier .		<p>表示订阅名称由一个标识符组成。</p>
soffset_clause = "SOFFSET" int_lit .		<p>soffset_clause 是用于查询 InfluxDB 中的时间序列数据时指定偏移量的语法结构。它由 SOFFSET 和一个整数字面量 int_lit 组成。例如：</p> <pre>SELECT * FROM mymeasurement WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z' SOFFSET 10</pre>
slimit_clause = "SLIMIT" int_lit .		<p>slimit_clause 是用于查询 InfluxDB 中的时间序列数据时指定返回结果数量限制的语法结构。它由 SLIMIT 和一个整数字面量 int_lit 组成，只能用于 SELECT 语句，并且必须放在查询语句的末尾。例如：</p> <pre>SELECT * FROM mymeasurement WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z' SLIMIT 10</pre>
series_id = int_lit .		<p>series_id 是用于查询 InfluxDB 中的时间序列数据时指定数据系列 ID 的语法结构。它由一个整数字面量 int_lit 组成。使用 series_id，您可以查询指定 ID 的数据系列，但是只能用于 SELECT 语句，并且必须放在查询语句的末尾。例如，查询 ID 为 1234 的数据系列：</p> <pre>SELECT * FROM mymeasurement WHERE series_id = 1234</pre>
retention_policy_shard_group_duration = "SHARD DURATION" duration_lit .	不支持	-

retention_policy_replication = "REPLICATION" int_lit .	不支持	-
retention_policy_option = retention_policy_duration   retention_policy_replication   retention_policy_shard_group_duration   "DEFAULT" .	不支持	-
retention_policy_name = "NAME" identifier .	不支持	-
retention_policy_duration = "DURATION" duration_lit .	不支持	-
retention_policy = identifier .	不支持	-
query_name = identifier .	支持	<p>query_name 是用于对查询语句进行命名的语法结构。以下使用 SELECT 语句查询名为 mymeasurement 的测量中在特定时间范围内的所有数据点，并使用 GROUP BY 对结果进行分组。然后，使用 INTO 将查询结果存储到名为 mycount 的查询中。</p> <pre>SELECT COUNT(*) FROM mymeasurement WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z' GROUP BY time(1h) INTO mycount</pre>
query_id = int_lit .	支持	<p>query_id 是用于查询 InfluxDB 中的已命名查询的标识符。它由一个整数字面量 int_lit 组成。例如：</p> <pre>SELECT * FROM 1234</pre>
privilege = "ALL" [ "PRIVILEGES" ]   "READ"   "WRITE" .	不支持	-
policy_name = identifier .	不支持	-

<p>password = string_lit .</p>	<p>支持</p>	<p>password 是用于在连接 InfluxDB 时指定密码的语法结构。它由一个字符串字面量 string_lit 组成。使用 password，您可以在连接 InfluxDB 时指定密码，以便进行身份验证。例如，使用 influx 命令连接到名为 myinfluxdb.example.com 的 InfluxDB 实例，并指定用户名为 myuser，密码为 mypassword。示例如下：</p> <pre>influx -username myuser -password 'mypassword' -host myinfluxdb.example.com -port 8086</pre>
<p>order_by_clause = "ORDER BY" sort_fields . sort_fields = sort_field { "," sort_field } . sort_field = field_key [ ASC   DESC ] .</p>	<p>order_by_clause = ORDER BY time [ ASC   DESC ] sql_order_by_clause = SQLORDER BY {sql_order_expr [ ASC   DESC ]} [, ...n ] SQLLIMIT N SQLOFFSET N] sql_order_expr = {alias   expr   position}</p>	<ul style="list-style-type: none"> <li>ORDER BY 子句用于按照指定的列对查询结果进行排序。ORDER BY 子句后面必须跟一个排序字段，通常是 time，它是 InfluxDB 中所有测量的默认时间戳列。您可以选择按升序（ASC）或降序（DESC）排序。例如： <pre>SELECT * FROM mymeasurement WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z' ORDER BY time DESC</pre> </li> <li>除了 InfluxQL 的 ORDER BY 子句外，还有一个类似的 SQL 语法的 SQLORDER BY 子句。它允许您使用 SQL 风格的表达式对查询结果进行排序，并支持 LIMIT 和 OFFSET 子句。例如，以下是一个使用 SQLORDER BY 子句的示例： <pre>SELECT * FROM mymeasurement SQLORDER BY value DESC, time ASC SQLLIMIT 10 SQLOFFSET 20</pre> </li> </ul>

<p>on_cluse = "ON" db_name .</p>		<p>ON 子句用于指定查询操作的目标数据库。它通常用于跨数据库查询或在具有多个数据库的 InfluxDB 实例中执行查询。ON 子句后面必须跟一个数据库名称，例如：</p> <pre>SELECT * FROM mymeasurement ON mydatabase WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z'</pre>
<p>offset_clause = "OFFSET" int_lit .</p>	<p>支持</p>	<p>OFFSET 子句用于指定查询结果集的偏移量。它通常与 LIMIT 子句一起使用，以限制查询结果集的大小并指定从哪个偏移量开始返回结果。OFFSET 子句后面必须跟一个整数，例如：</p> <pre>SELECT * FROM mymeasurement WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05- 31T00:00:00Z' LIMIT 10 OFFSET 20</pre> <p><b>注意：</b>OFFSET 子句必须放在 LIMIT 子句之后。如果省略 OFFSET 子句，则默认偏移量为0。如果查询结果集的大小小于指定的偏移量，则返回空结果集。</p>
<p>measurements = measurement { "," measurement } .</p>		<p>measurements 是一个由多个测量名称组成的列表，用于指定查询操作的目标测量。它通常用于跨多个测量执行查询操作。measurements 语法结构由一个或多个 measurement 组成，每个 measurement 表示一个测量名称。例如，使用 SELECT 语句查询名为 measurement1 和 measurement2 的两个测量中在特定时间范围内的所有数据点，执行语句如下：</p> <pre>SELECT * FROM measurement1, measurement2 WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z'</pre>
<p>measurement_name = identifier   regex_lit .</p>	<p>measur ement_ name = identifi er</p>	<p>measurement_name 表示一个测量名称，用于标识 InfluxDB 中存储的一组数据。它是一个标识符，由一个或多个字母、数字或下划线组成，且必须以字母开头。</p>
<p>measurement = measurement_name   ( policy_name "." measurement_name )  ( db_name "." [ policy_name ] "." measurement_name ).</p>	<p>measur ement = measur ement_ name</p>	<p>measurement 表示一个测量，它是由测量名称、字段和标签组成的数据存储单位。每个测量都包含一个或多个字段和零个或多个标签，用于描述数据的特征和属性。measurement 语法结构由一个 measurement_name 组成，表示测量的名称。</p>

<p>limit_clause = "LIMIT" int_lit .</p>	<p>支持</p>	<p>limit_clause 用于限制查询结果返回的数据点数量。它由关键字 LIMIT 和一个整数值组成，该整数值表示返回的数据点数量。</p>
<p>into_clause = "INTO" ( measurement   back_ref ).</p>	<p>不支持</p>	<p>-</p>
<p>host = string_lit .</p>	<p>支持</p>	<p>host 表示一个主机名，用于标识 InfluxDB 中存储的数据所属的主机。它是一个字符串，由一系列字符组成，用于唯一标识一个主机。</p>
<p>groupby_clause = "GROUP BY" dimensions fill(fill_option).</p>		<p>GROUP BY 用于将查询结果按照指定的维度进行分组。groupby_clause 语法结构由以下部分组成：</p> <ul style="list-style-type: none"> <li>• "GROUP BY": 关键字。表示按照指定的维度进行分组。</li> <li>• dimensions: 一个或多个维度，用逗号分隔。维度可以是测量中的字段或标签名。</li> <li>• fill(fill_option): 一个可选的填充选项，用于指定如何处理缺失的数据点。</li> </ul> <p>以下示例使用 SELECT 语句查询名为 my_measurement 的测量中在特定时间范围内的所有数据点，并按照 location 标签对数据点进行分组，最后计算每组数据点的平均值。</p> <pre>SELECT mean(value) FROM my_measurement WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z' GROUP BY location</pre>
<p>from_clause = "FROM" measurements .</p>		<p>FROM用于指定查询的数据源，即测量名称。</p>
<p>fill_option = "null"   "none"   "previous"   int_lit   float_lit   "linear" .</p>	<p>fill_option = "none" .</p>	<p>fill_option 用于指定如何处理查询结果中的缺失数据点。fill_option 可以是以下三种选项之一：</p> <ul style="list-style-type: none"> <li>• fill(null): 用 null 填充缺失的数据点。</li> <li>• fill(previous): 用前一个数据点的值填充缺失的数据点。</li> <li>• fill(none): 不对缺失的数据点进行填充，直接将其忽略。</li> </ul>
<p>fields = field { "," field } .</p>	<p>支持</p>	<p>fields 用于指定需要查询的字段列表，可以是一个或多个字段，用逗号分隔。如果需要查询所有字段，可以使用通配符 "*" 。</p> <ul style="list-style-type: none"> <li>• field: 一个或多个字段名，用逗号分隔。</li> <li>• { "," field }: 可选的重复部分，用于指定多个字段名。</li> </ul>
<p>field_key = identifier .</p>		<p>field_key 用于指定测量中的字段名。identifier 为一个标识符，用于指定测量中的字段名。</p>

<p>field = expr [ alias ] .</p>		<p>field 用于指定查询结果中的一个字段。field 语法结构由以下部分组成：</p> <ul style="list-style-type: none"> <li>• expr: 一个表达式，用于计算查询结果中的值。</li> <li>• [ alias ]: 可选的别名，用于为查询结果中的字段指定一个名称。</li> </ul>
<p>expr = unary_expr { binary_op unary_expr } .</p>		<p>expr 用于表示一个表达式，可以是一个常量、一个标识符或一个由运算符和操作数组成的复杂表达式。expr 语法结构由以下部分组成：</p> <ul style="list-style-type: none"> <li>• unary_expr: 一个一元表达式，可以是一个常量、一个标识符或一个由一元运算符和操作数组成的表达式。</li> <li>• { binary_op unary_expr }: 可选的重复部分，用于指定多个二元运算符和操作数。</li> </ul>
<p>dimensions = dimension { "," dimension } .</p>		<p>dimensions 用于指定查询结果中的一个或多个维度。</p> <ul style="list-style-type: none"> <li>• dimensions 语法结构由以下部分组成：</li> <li>• dimension: 一个维度，用于指定查询结果中的一个维度。 { "," dimension }: 可选的重复部分，用于指定多个维度。</li> </ul> <p>以下示例，使用 SELECT 语句查询名为 my_measurement 的测量中在特定时间范围内的所有数据点，并计算 temperature 字段的平均值。我们还使用 GROUP BY子句指定了两个维度 location 和 device，这意味着查询结果将按照这两个维度进行分组。</p> <pre>SELECT mean(temperature) FROM my_measurement WHERE time &gt; '2023-05-01T00:00:00Z' AND time &lt; '2023-05-31T00:00:00Z' GROUP BY location, device</pre>
<p>dimension = expr .</p>		<p>dimension 用于指定查询结果中的一个维度。expr 指一个表达式，用于计算查询结果中的值。</p>
<p>db_name = identifier .</p>		<p>db_name 用于指定测量中的数据库名。identifier 为一个标识符，用于指定测量中的字段名。</p>
<p>binary_op = "+"   "-"   "*"   "/"   "%"   "&amp;"   " "   "^"   "AND"   "OR"   "="   "!="   "&lt;&gt;"   "&lt;"   &lt;="   "&gt;"   "&gt;=" .</p>		<p>表示二元运算符。这些运算符包括加号、减号、乘号、除号、取模运算、按位与、按位或、按位异或、逻辑与、逻辑或、等于、不等于、小于等于、大于等于、小于和大于。这些运算符用于操作两个值，并返回一个结果。</p>
<p>back_ref = ( policy_name ".:MEASUREMENT" )   ( db_name "." [</p>	<p>不支持</p>	<p>-</p>



policy_name ] ".:MEASUREMENT" ).		
alias = "AS" identifier .	支持	该语义用于创建一个别名。该别名由两个部分组成：AS 和 identifier。AS 是一个关键字，后面跟着的 identifier 是用于表示别名的标识符。

# InfluxQL 函数

最近更新时间：2024-08-30 10:43:51

## Functions

InfluxQL 函数聚合、选择、转换的兼容性，请参见下表。具体使用语法、示例等详情，请参见 [InfluxQL functions](#)。

类型	函数	开源 InfluxDB 版	时序数据库 CTSDB 3.0 版
Transformations	ABS()	✓	✓
	ACOS()	✓	✓
	ASIN()	✓	✓
	ATAN()	✓	✓
	ATAN2()	✓	✓
	CEIL()	✓	✓
	COS()	✓	✓
	CUMULATIVE_SUM()	✓	✗
	DERIVATIVE()	✓	✗
	DIFFERENCE()	✓	✗
	ELAPSED()	✓	✗
	EXP()	✓	✓
	FLOOR()	✓	✓
	HISTOGRAM()	✓	✗
	LN()	✓	✓
	LOG()	✓	✓
LOG2()	✓	✓	
LOG10()	✓	✓	

	MOVING_AVERAGE()	✓	✗
	NON_NEGATIVE_DERIVATIVE()	✓	✗
	NON_NEGATIVE_DIFFERENCE()	✓	✗
	POW()	✓	✓
	ROUND()	✓	✓
	SIN()	✓	✓
	SQRT()	✓	✓
	TAN()	✓	✓
Aggregations	COUNT()	✓	✓
	SUM()	✓	✓
	MEAN()	✓	✓
	PERCENTILE()	✓	✓
	DISTINCT()	✓	✗
	INTEGRAL()	✓	✗
	MEDIAN()	✓	✗
	MODE()	✓	✗
	SPREAD()	✓	✗
	STDDEV()	✓	✗
Selectors	MIN()	✓	✓
	MAX()	✓	✓
	TOP()	✓	✓
	BOTTOM()	✓	✓
	FIRST()	✓	✓* 执行示例

			<pre>SELECT FIRST("level description") FROM "h2o_feet"</pre>
	LAST()	✓	<p>✓*</p> <p><b>执行示例</b></p> <pre>SELECT LAST("level description"),"lo cation","water_le vel" FROM "h2o_feet"</pre>
	PERCENTILE()	✓	✗
	SAMPLE()	✓	✗

## 数学运算符 ( mathematical operators )

时序数据库 CTSDB 3.0版当前暂不支持数据运算法相关的函数。

函数	开源 InfluxDB	时序数据库 CTSDB 3.0版
cumulative_sum	✓	✗
derivative	✓	✗
non_negative_derivative	✓	✗
difference	✓	✗
non_negative_difference	✓	✗
elapsed	✓	✗
moving_average	✓	✗
holt_winters	✓	✗
holt_winters_with_fit	✓	✗