



链滴

ClickHouse 的 MergeTree 引擎

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1515822249896>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

背景

ClickHouse中的代表引擎是MergeTree引擎。在标准的MergeTree引擎外，还有一些特色的子引擎。这些引擎各自的特点是什么呢？

引擎列表

- [MergeTree](#)
- [AggregatingMergeTree](#)
- [CollapsingMergeTree](#)
- [ReplacingMergeTree](#)
- [SummingMergeTree](#)

AggregatingMergeTree

[AggregatingMergeTree](#)支持对AggregateFunction的中间状态进行直接Merge。

这部分不太好写，先贴两个官方实例，更多内容见文中对应的官方链接。

实例一

下面两个查询是等价的

```
SELECT uniq(UserID) FROM table
```

和

```
SELECT uniqMerge(state) FROM (SELECT uniqState(UserID) AS state FROM table GROUP BY R  
gionID)
```

实例二

创建一个快速的视图

1. 创建视图

```
CREATE MATERIALIZED VIEW test.basic  
ENGINE = AggregatingMergeTree(StartDate, (CounterID, StartDate), 8192)  
AS SELECT  
  CounterID,  
  StartDate,  
  countState() AS Counts,  
  sumState(Sign) AS Visits,  
  uniqState(UserID) AS Users  
FROM test.visits  
GROUP BY CounterID, StartDate;
```

2. 插入数据

```
INSERT INTO test.visits ...
```

数据插入数据表中时，同时也会插入视图view中。

3. 查看结果

```
SELECT
  StartDate,
  countMerge(Counts) AS Counts,
  sumMerge(Visits) AS Visits,
  uniqMerge(Users) AS Users
FROM test.basic
GROUP BY StartDate
ORDER BY StartDate;
```

实例三

来源于参考[1].

```
CREATE TABLE mt2
(
  whatever Date DEFAULT '2000-01-01',
  key String,
  value String,
  first AggregateFunction(min, DateTime),
  last AggregateFunction(max, DateTime),
  total AggregateFunction(count, UInt64)
) ENGINE = AggregatingMergeTree(whatever, (key, value), 8192)

insert into mt2 (key, value, first,last,total) select 'www.google.com','1.2.3.4',minState(toDate
ime(1498241729)),maxState(toDateTime(1498241729)), countState(cast(1 as UInt64));

insert into mt2 (key, value, first,last,total) select 'www.google.com', '1.2.3.5', minState(toDat
Time(1498242729)),maxState(toDateTime(1498242729)), countState(cast(1 as UInt64));

insert into mt2 (key, value, first,last,total) select 'www.google.com', '1.2.3.6', minState(toDat
Time(1498242729)),maxState(toDateTime(1498242829)), countState(cast(1 as UInt64));

select key, value, minMerge(first), maxMerge(last), countMerge(total) from mt2 group by k
y, value;

optimize table mt2; -- compact the table and merge multiple rows with the same (key, valu
)
```

备注

- 一般情况下用不上AggregatingMergeTree，因为直接查询也很快。

CollapsingMergeTree

Merge时支持删除数据和合并数据

官网介绍中提到的场景，做为一款统计产品，回收上来的日志有点击日志(hit logs)和变化日志(如session状态, user状态)。

实例一

来源于参考[1].

```
CREATE TABLE cmt
(
  whatever Date DEFAULT '2000-01-01',
  key String,
  value String
  sign Int8
) ENGINE = CollapsingMergeTree(whatever, (key, value), 8192, sign)
```

```
insert into cmt (key, value, sign) values ('k1', 'v1', 1)
insert into cmt (key, value, sign) values ('k1', 'v1', -1)
insert into cmt (key, value, sign) values ('k1', 'v1 update', 1)
insert into cmt (key, value, sign) values ('k2', 'just delete this one', 1)
insert into cmt (key, value, sign) values ('k2', 'just delete this one', -1)
```

```
select * from cmt FINAL;
```

| whatever | key | value | sign |
|------------|-----|-----------|------|
| 2000-01-01 | k1 | v1 update | 1 |

```
;)optimize table cmt
```

```
;) select * from cmt;
```

| whatever | key | value | sign |
|------------|-----|-----------|------|
| 2000-01-01 | k1 | v1 update | 1 |

ReplacingMergeTree

[ReplacingMergeTree](#)在merge时，对于主键相同的数据会进行去重。去重的规则是：

- 有version字段时，保留version值最大的行
- 没有version字段时，保留最后插入的行

其特点是：

- 只在merge的时候去重
- merge的时间点是不确定的，不推荐执行OPTIMIZE来强制merge，因为会带来大量的读写操作
- 这个是由非Yandex.Metrica部门来自己开发的

适用的场景为：

- 适用于移除重复数据来节省空间

- 不适用于保证唯一性(官方描述, why?)

没有merge时, 需要加FINAL关键字, 但是这样会导致查询变慢, 所以慎用。

SummingMergeTree

[SummingMergeTree](#), 在merge时, 主键相同的行, 其非主键部分的数值列会相加, 非数据列会保第一个遇到的行的值

示例

```
[(1, 100)] + [(2, 150)] -> [(1, 100), (2, 150)]
[(1, 100)] + [(1, 150)] -> [(1, 250)]
[(1, 100)] + [(1, 150), (2, 150)] -> [(1, 250), (2, 150)]
[(1, 100), (2, 150)] + [(1, -100)] -> [(2, 150)]
```

参考

1. [Redshift vs. BigQuery vs. Snowflake benchmark](https://news.ycombinator.com/item?id=1434272)(<https://news.ycombinator.com/item?id=1434272>)