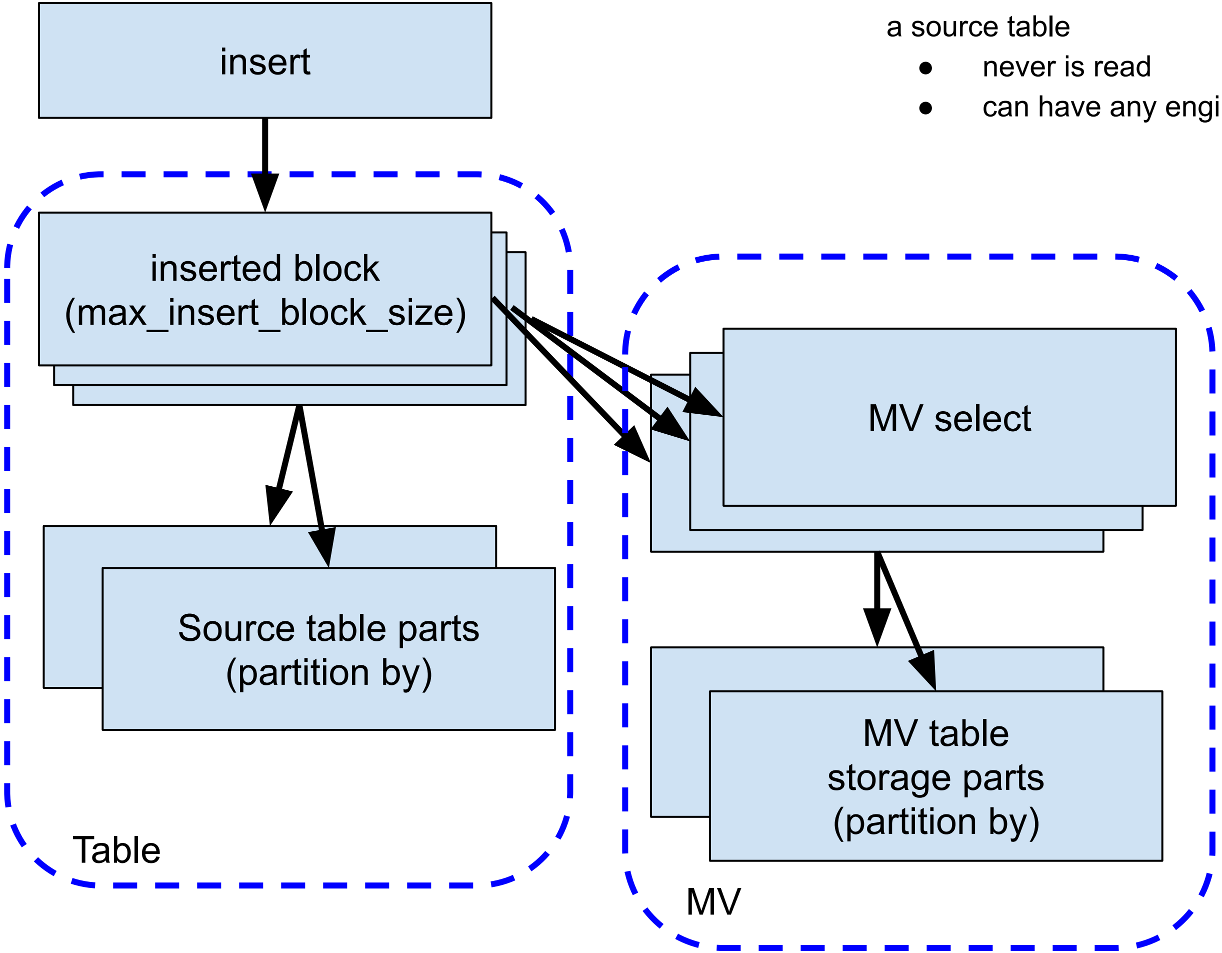
ClickHouse

Everything you should know about materialized views.

Zhuravlev, DBA, Revjet

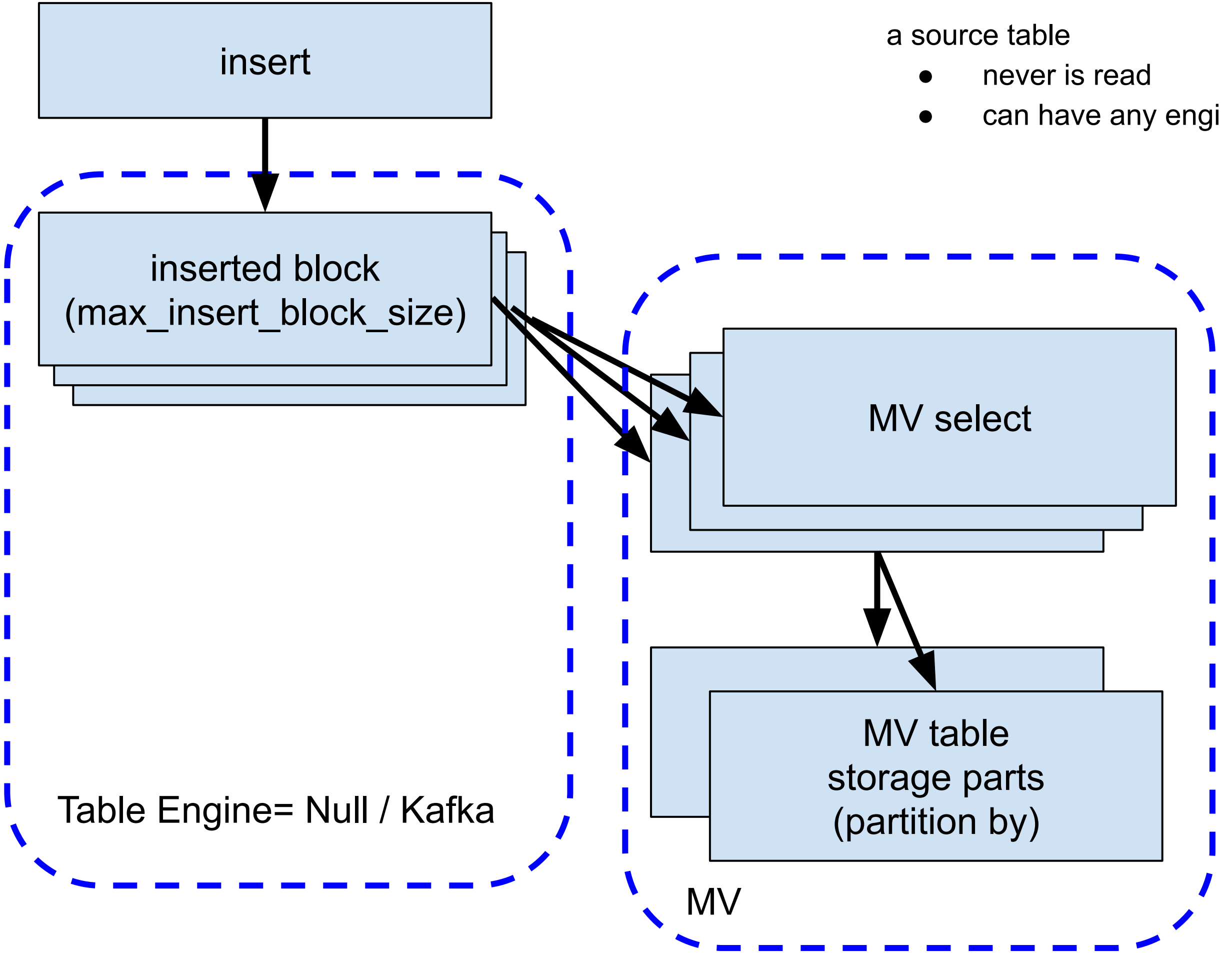
MV internally



a source table

- never is read
- can have any engine, even engine Null

MV internally



a source table

- never is read
- can have any engine, even engine Null

MV use cases

- Data pre-aggregation (Summing/Aggregating)
- Entry-point / Extract-Transform
- Duplicate data to another table with different PK (+emulation of inverse indexes)
- Kafka

MV with TO vs MV .inner

Implicit table `.inner.mv1`

```
CREATE MATERIALIZED VIEW mv1
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d)
ORDER BY (a, b, d)
AS
SELECT a, b, d, count() AS cnt
FROM source
GROUP BY a, b, d;
```

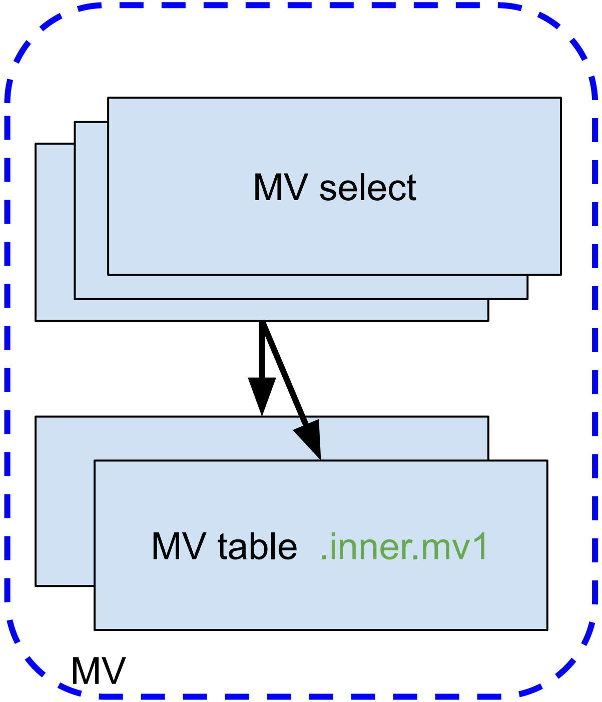
```
DESCRIBE TABLE `.inner.mv1`

┌─name─┬─type───┬─default_type─┐
│ a    │ Int64  │              │
│ b    │ Int64  │              │
│ d    │ Date   │              │
│ cnt  │ UInt64 │              │
└──────┴────────┴──────────────┘
```

Explicit table dest

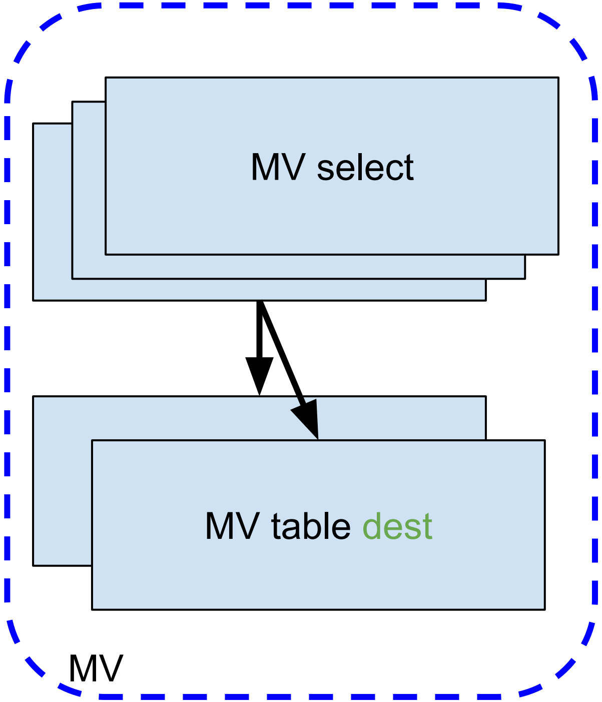
```
CREATE TABLE dest
(a Int64, b Int64, d Date, cnt UInt64)
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d)
ORDER BY (a, b, d);

CREATE MATERIALIZED VIEW mv1
TO dest
AS
SELECT a, b, d, count() AS cnt
FROM source
GROUP BY a, b, d;
```

MV with TO vs MV .inner

Implicit table `.inner.mv1`



Explicit table dest



MV with TO vs MV .inner

Implicit table `.inner.mv1`

- optimize_move_to_prewhere does not work
  should query from `.inner.mv1`
- populate does work
- drop table mv1 implicit .inner table is removed

Explicit table dest

- populate does not work (use insert)
- drop table mv1 does not remove TO table (dest)

MV with TO vs MV .inner

- insert into mv1 / insert into `.inner.mv1` / insert into dest

Populate

- I never use it

Common mistakes

- Storage table knows nothing about MV and about MV’s select !!!

Common mistakes

Incorrect PK (order by) of storage-table

Incorrect

```
CREATE MATERIALIZED VIEW mv1
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d)
ORDER BY (a, b)
AS SELECT a, b, d, count() AS cnt
FROM source
GROUP BY a, b, d;
```

Engine rules:

```
a -> a
b -> b
d -> ANY(d)
cnt -> sum(cnt)
```

Correct

```
CREATE MATERIALIZED VIEW mv1
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d)
ORDER BY (a, b, d)
AS SELECT a, b, d, count() AS cnt
FROM source
GROUP BY a, b, d;
```

Engine rules:

```
a -> a
b -> b
d -> d
cnt -> sum(cnt)
```

Common mistakes

Double grouping (incorrect)

```
CREATE MATERIALIZED VIEW mv1
ENGINE = AggregatingMergeTree
PARTITION BY toYYYYMM(hour)
ORDER BY hour
POPULATE
AS
SELECT toStartOfHour(time) hour,
       maxState(cnt_by_minute) max_by_hour,
       sumState(cnt_by_minute) sum_by_hour
FROM
(
    SELECT minute, count() AS cnt_by_minute
    FROM source
    GROUP BY minute
)
GROUP BY hour
```

```
insert into source
values (now()), (now())

result: max_by_hour = 2
```

```
insert into source
values (now());
insert into source
values (now());

result: max_by_hour = 1
```

Common mistakes

MV is not related to the source table !!!

- MV does not watch for Replacing/CollapsingMT of the source table

- MV knows nothing about truncate / alter delete / alter update / drop partition / drop table / rename of the source table

- MV can store data for different period

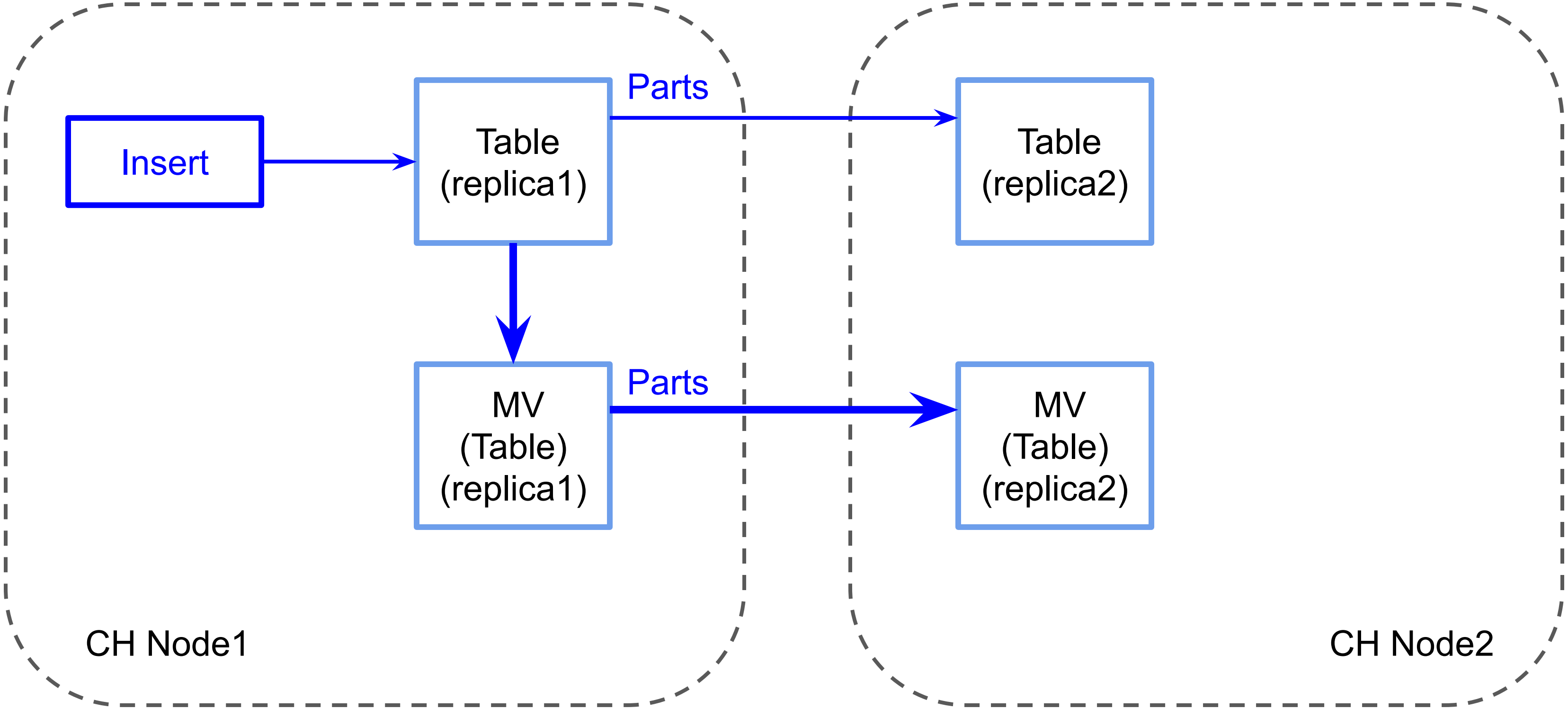
Common mistakes

MV uses column’s names !!!

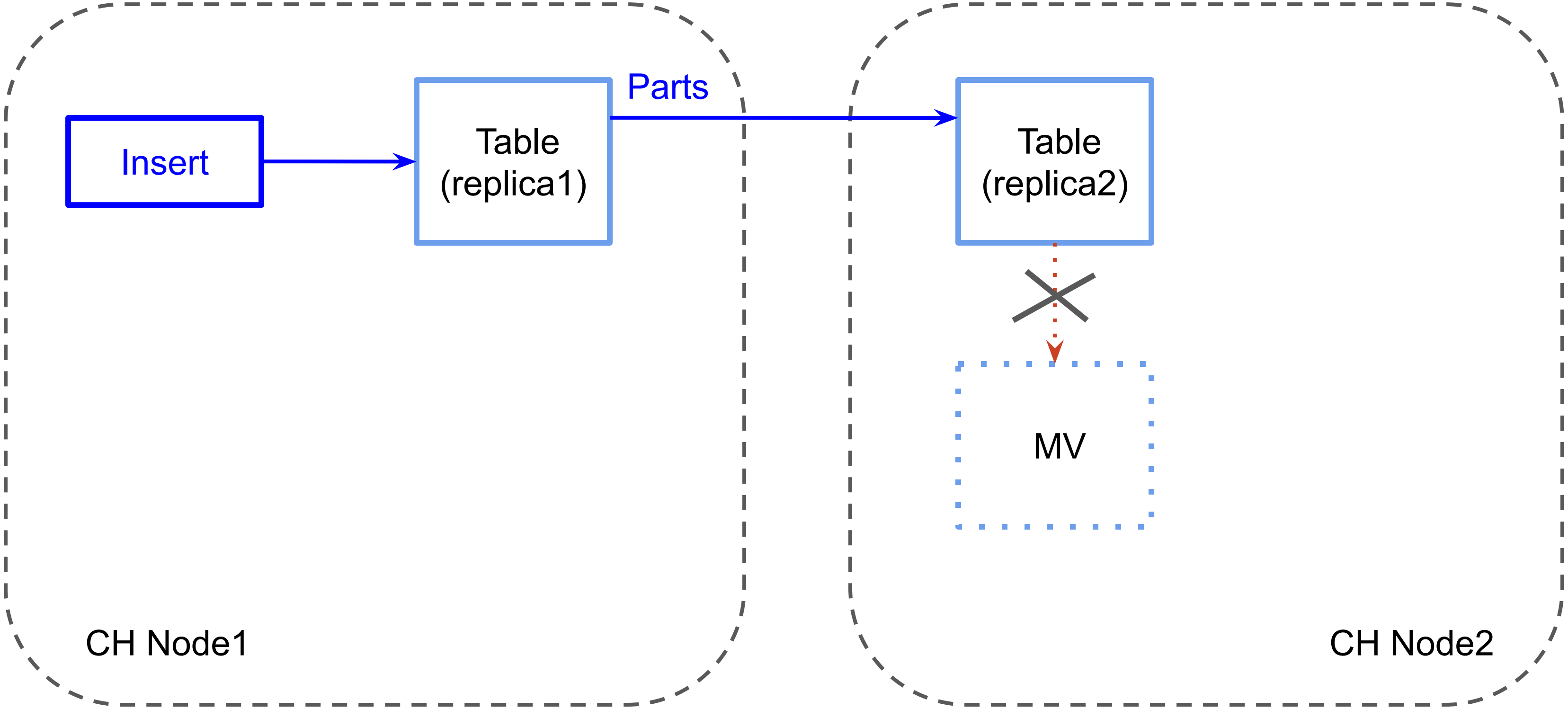
```
CREATE MATERIALIZED VIEW
mv1 (a Int64, d Date, cnt Int64)
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d)
ORDER BY (a, d)
POPULATE
AS
SELECT a, d, count()
FROM source
GROUP BY a, d;
```

```
CREATE MATERIALIZED VIEW
mv1 (a Int64, d Date, cnt Int64)
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d)
ORDER BY (a, d)
POPULATE
AS
SELECT a, d, count() as cnt
FROM source
GROUP BY a, d;
```

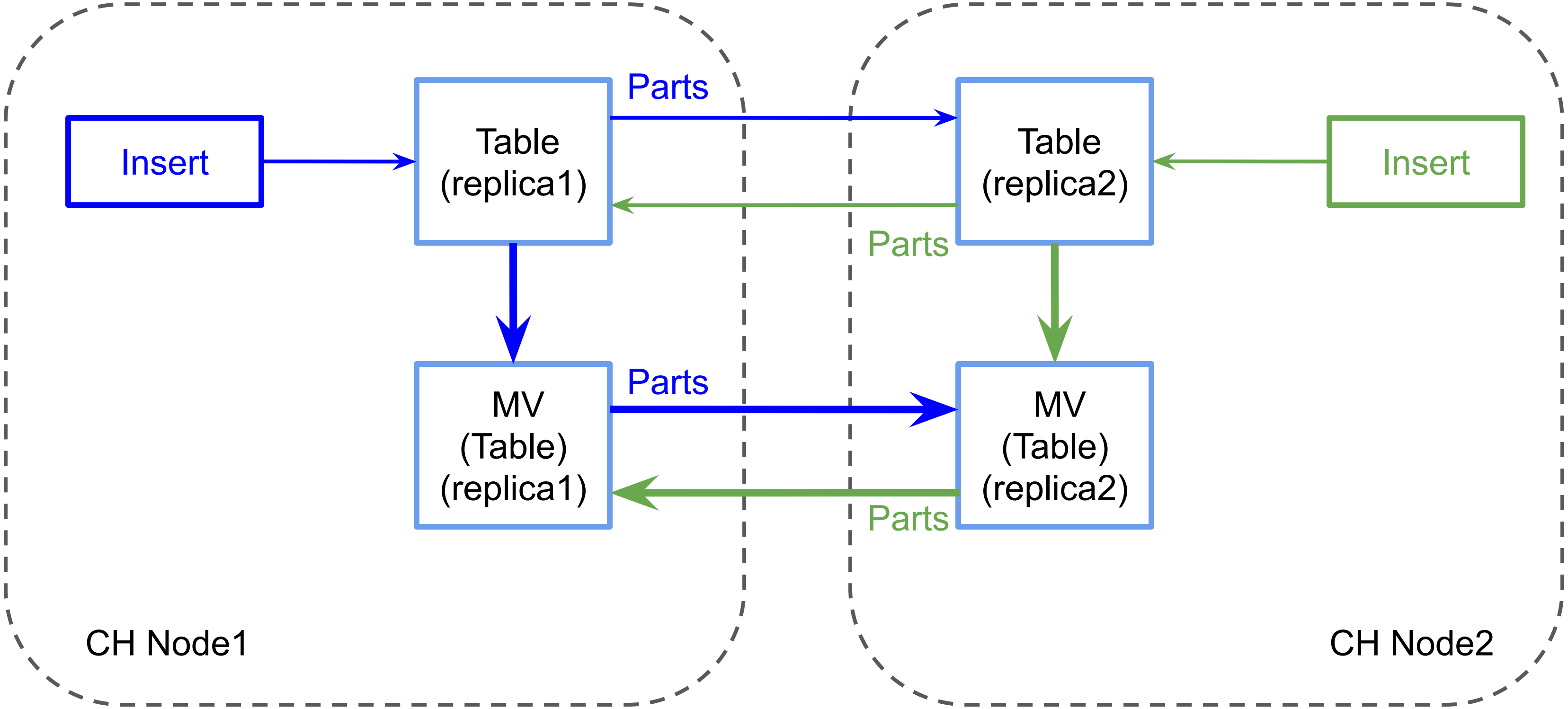
ReplicatedMT & MV



ReplicatedMT и MV

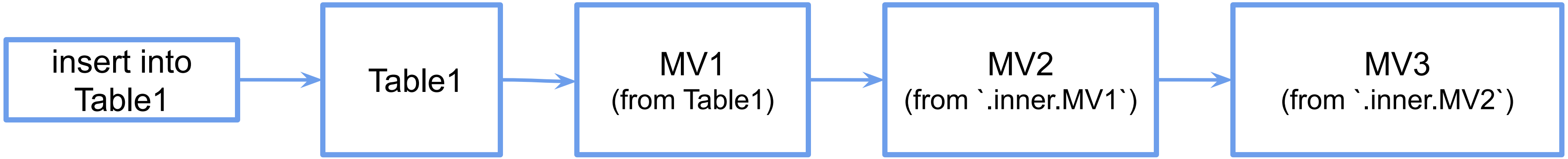


ReplicatedMT и MV



MV cascades

.inner



insert into Table1

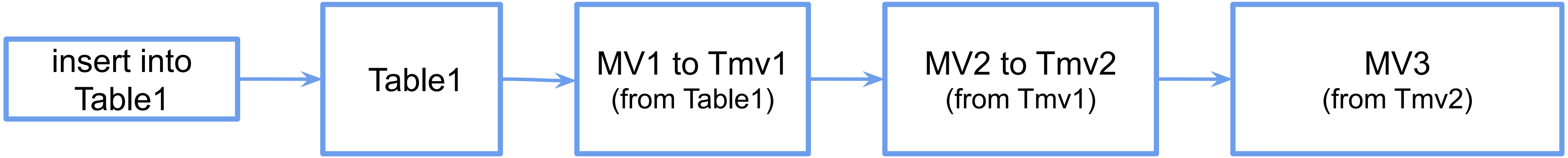
Table1

MV1 (from Table1)

MV2 (from `.inner.MV1`)

MV3 (from `.inner.MV2`)

TO dest



insert into Table1

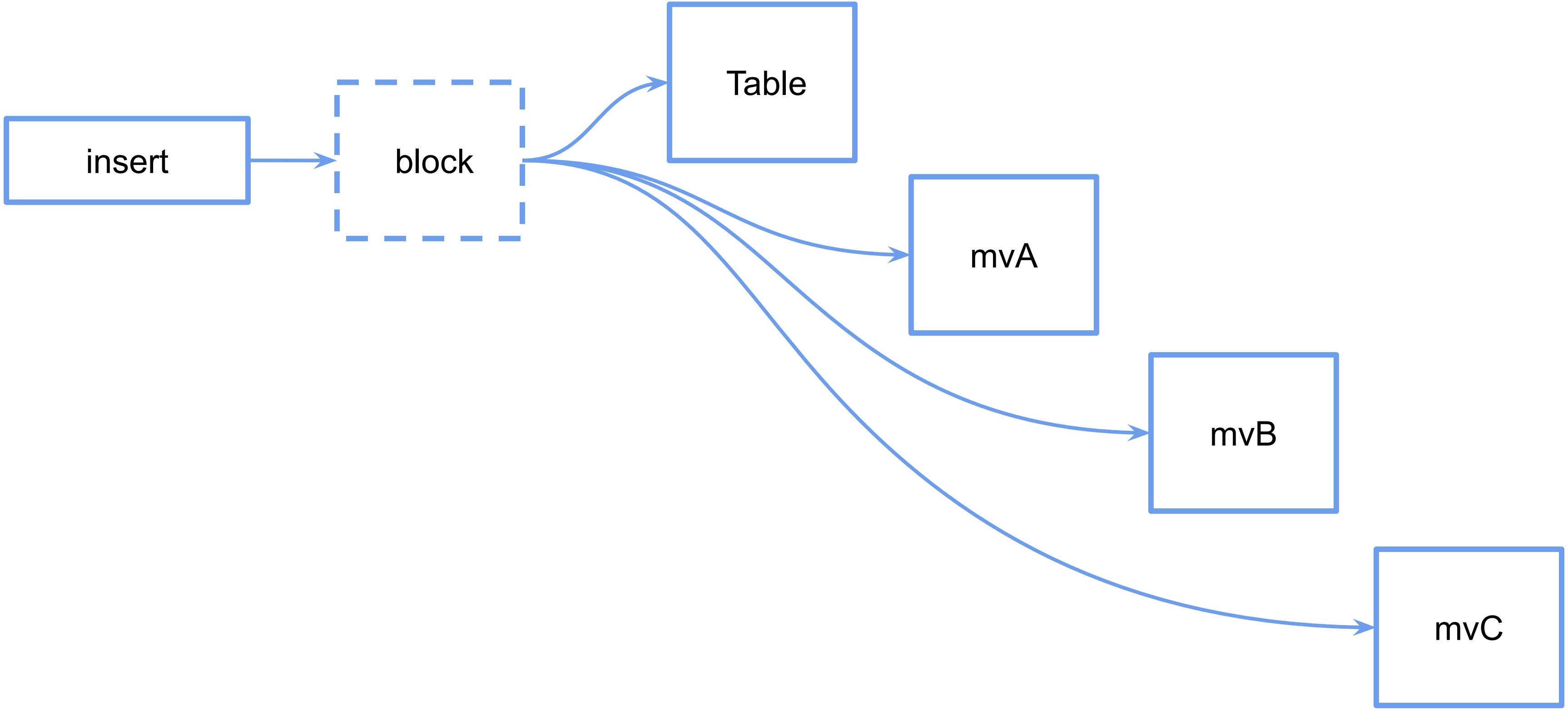
Table1

MV1 to Tmv1 (from Table1)

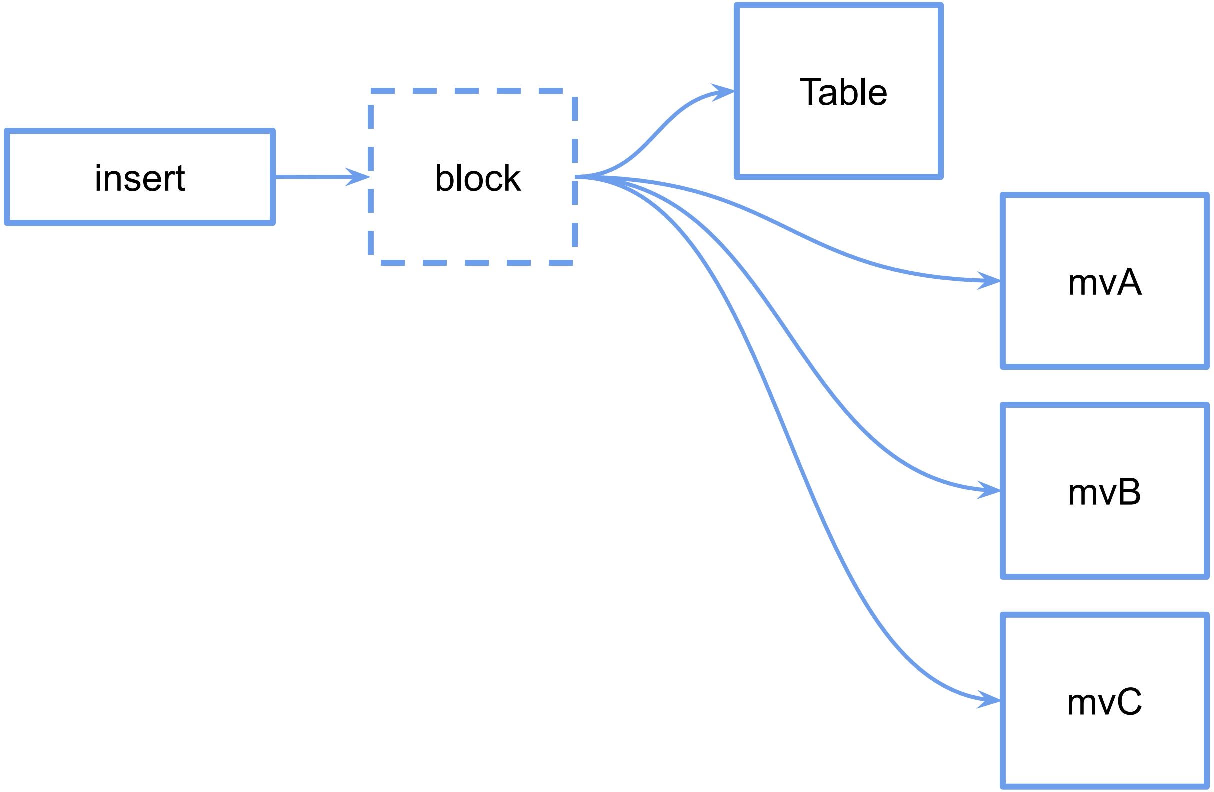
MV2 to Tmv2 (from Tmv1)

MV3 (from Tmv2)

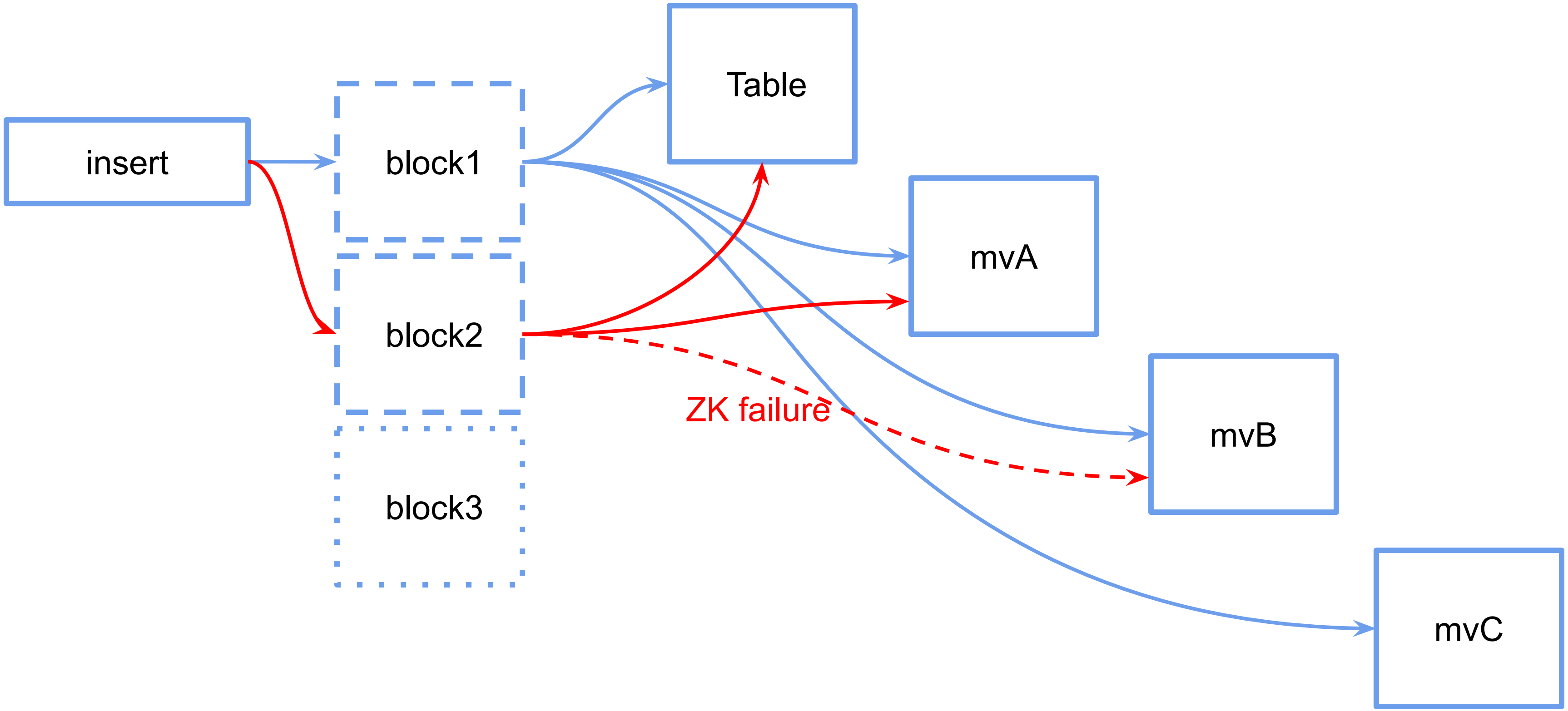
MV Insertion order



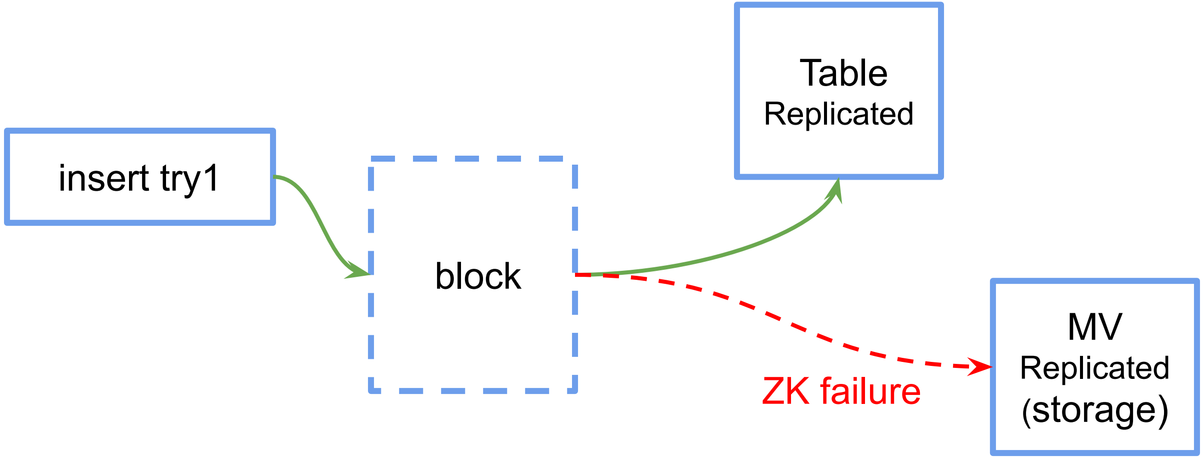
MV Insertion order & parallel_view_processing



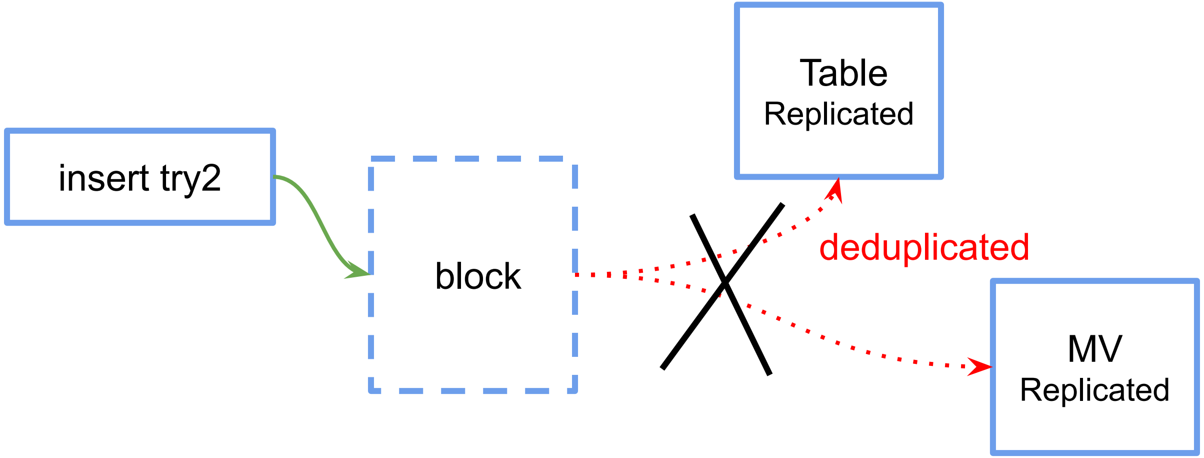
MV Insertion order



MV insert deduplication



MV insert deduplication



- setting deduplicate_blocks_in_dependent_materialized_views = 0 (default)

JOIN & MV

- Multiple blocks -> multiple JOIN

- External dictionaries и dictGet or engine=Join и joinGet

JOIN & MV

```
CREATE MATERIALIZED VIEW mv1
...
AS
SELECT ...
FROM
(
     SELECT ...
     FROM source join some_dimension on (...)
)
```

JOIN & MV

- MV for both tables JOIN (2 MV to dest + RIGHT JOIN)

```
CREATE MATERIALIZED VIEW mv1
TO dest
SELECT ...
FROM source left join some_dimension on (...)


CREATE MATERIALIZED VIEW mv2
TO dest
SELECT ...
FROM some_dimension right join source on (...)
```

MV from distributed / To distributed

- From Local* to Local
- From Distributed to Distributed
- From Local* to Distributed
- From Distributed to Local

* (Local is behind of Distributed it does not matter)

MV from distributed / To distributed

```
CREATE TABLE src (A Int64, B String) Engine=Memory;

CREATE TABLE src_log (A Int64, B String) Engine=File(JSONEachRow);

CREATE MATERIALIZED VIEW src_log_mv TO src_log AS
SELECT A, B FROM src WHERE A % 2 = 0;

INSERT INTO src SELECT number A, 'x' B FROM numbers(4);
INSERT INTO src SELECT number A, 'y' B FROM numbers(4);

SELECT count() FROM src_log
┌─count()─┐
│    4    │
└─────────┘

sudo wc -l /var/lib/clickhouse/data/default/src_log/data.JSONEachRow
4 /var/lib/clickhouse/data/default/src_log/data.JSONEachRow
```

MV from distributed / To distributed

```
SERVER2: CREATE TABLE dest (A Int64, B String)
Engine=MergeTree ...;

CREATE TABLE remote_src_log AS remote('SERVER2', default, dest)

CREATE MATERIALIZED VIEW src_log_mv1
TO remote_src_log AS
SELECT A, B
FROM src WHERE A % 2 = 0;
```

How to alter MV

Implicit table (.inner.mv1)

1. stop inserts
2. detach table mv1
3. alter table `.inner.mv1`
4. attach materialized view mv1

4a. edit .sql + attach table

Explicit table (TO dest)

1. stop inserts
2. alter table dest
3. drop table mv1
4. create materialized view mv1

How to alter MV

Implicit table (.inner.mv1)

```
DETACH TABLE mv1

ALTER TABLE `.inner.mv1`
    ADD COLUMN b Int64 AFTER a,
    MODIFY ORDER BY (a, b)

ATTACH MATERIALIZED VIEW mv1
ENGINE = SummingMergeTree
ORDER BY (a, b) AS
SELECT a, b, sum(amount) AS s
FROM source
GROUP BY a, b
```

Explicit table (TO dest)

```
ALTER TABLE dest
    ADD COLUMN b Int64 AFTER a,
    MODIFY ORDER BY (a, b)

DROP TABLE MV1

CREATE MATERIALIZED VIEW mv1
TO dest
SELECT a, b, sum(amount) AS s
FROM source
GROUP BY a, b
```

How to alter MV

```
SET allow_experimental_alter_materialized_view_structure = 1;

ALTER TABLE mv1 MODIFY QUERY
    SELECT v * 2 as v, 1 as v2
    FROM source;
```

How to fill MV for existing data (populate)

- create MV with where date >= date_in_future
  wait for date_in_future, insert all before date_in_future

```
CREATE TABLE dest(a Int64, d Date, cnt UInt64)
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d) ORDER BY (a, d);

-- create MV c where date >= in_the_future
CREATE MATERIALIZED VIEW mv1 TO dest AS
SELECT a, d, count() AS cnt
FROM source
WHERE d >= '2020-11-01'
GROUP BY a, d;

-- arrives 2020-11-01
INSERT INTO dest -- insert all for before in_the_future
SELECT a, d, count() AS cnt
FROM source
WHERE d < '2020-11-01' -- piece by piece by 1 month (or .. day)
GROUP BY a, d;
```

How to fill MV for existing data (populate)

- create MV / insert all before today , (before current partition beginning)
- wait for a new month (new partition)
- drop incomplete partition, re-compute it

```
CREATE TABLE dest(a Int64, d Date, cnt UInt64)
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d) ORDER BY (a, d);

CREATE MATERIALIZED VIEW mv1 TO dest AS
SELECT a, d, count() AS cnt
FROM source
GROUP BY a, d;

INSERT INTO dest -- insert all for before today
SELECT a, d, count() AS cnt
FROM source
WHERE d < '2020-10-01' -- piece by piece by 1 month (or .. day)
GROUP BY a, d;

-- arrives 2020-11-01
ALTER TABLE dest DROP PARTITION 202010; -- because it is inconsistent

INSERT INTO dest
SELECT a, d, count() AS cnt
FROM source
WHERE d >= '2020-10-01' and d < '2020-11-01'
GROUP BY a, d;
```

How to fill MV for existing data (populate)

- stop inserts / insert all for today
- create MV
- start inserts / insert all before today

```
CREATE TABLE dest(a Int64, d Date, cnt UInt64)
ENGINE = SummingMergeTree
PARTITION BY toYYYYMM(d) ORDER BY (a, d);

-- stop inserts
INSERT INTO dest -- insert all for today
SELECT a, d, count() AS cnt
FROM source
WHERE d >= today()
GROUP BY a, d;

CREATE MATERIALIZED VIEW mv1 TO dest AS
SELECT a, d, count() AS cnt
FROM source GROUP BY a, d;

-- start insert
INSERT INTO dest -- insert all for before today
SELECT a, d, count() AS cnt
FROM source
WHERE d < today() -- piece by piece by 1 month (or .. day)
GROUP BY a, d;
```

How to fill MV for existing data (populate)

- snapshot / alter table freeze

```
CREATE TABLE dest(a Int64, cnt UInt64)
ENGINE = SummingMergeTree ORDER BY (a);

-- stop inserts
ALTER TABLE source FREEZE;

CREATE MATERIALIZED VIEW mv1 TO dest AS
SELECT a, count() AS cnt
FROM source GROUP BY a;

-- start inserts
create table temp_snapshot as source;

-- mv .../shadow/1/data/default/source/* .../data/default/temp_snapshot/detached/
-- cd .../data/default/temp_snapshot/detached/
-- let i=1;for f in `ls -1` ; do echo $i $f;((i++)); echo "alter table temp_snapshot
attach part '$f';"|clickhouse-client ; done

INSERT INTO dest -- insert into mv storage from temp_snapshot
SELECT a, count() AS cnt
FROM temp_snapshot -- piece by piece by ‘a’
GROUP BY a;
```

cookies

Alter MV and SummingMergeTree
https://gist.github.com/den-crane/3a8d57253711e277b2a60a75b5dfeef6

How to convert not replicated MV to replicated
https://gist.github.com/den-crane/80cb95e74f046be2d8ffae58d9b04e8f

How to alter MV
how to convert MV with implicit storage .inner to explicit storage (with TO)
https://gist.github.com/den-crane/431010ca08b9e51b960e55344b1dbbe3

MV_populating_with_freeze
https://gist.github.com/den-crane/64c05be67ef8a926242011188ce96f44

Several_MV_one_internal_store
https://gist.github.com/den-crane/005633470c70877dd28c00211cd9fcfb

Populate AggregatingMergeTree through null table
https://gist.github.com/den-crane/f7382cd4f1f859ff6ac46afe7dc9925a

Rename MV in CH before 19.8.3.8
https://gist.github.com/den-crane/d9a0bceb58a7d5314ea66e8edbebc9f7