8. Полезные SQL скрипты

Занимаемое место на диске таблицей

Занимаемое место на диске таблицей

```
WITH ['analytics.line_items', 'analytics.shipments'] AS required_schema_tables -- <---   .
SELECT
    concat(database, '.', table) AS schema_table,
    sum(rows) AS rows,
    formatReadableSize(sum(bytes_on_disk) as bytes_on_disk) AS size_on_disk,
    formatReadableSize(sum(data_uncompressed_bytes) as uncompressed_bytes) AS uncompressed_size,
    round(uncompressed_bytes/bytes_on_disk, 2) as comprassion_ratio
FROM system.parts
WHERE active AND schema_table in required_schema_tables
GROUP BY
    schema_table
ORDER BY bytes_on_disk DESC
;
```

В разрезе партиций

```
WITH 'analytics.line_items' AS required_schema_table -- <---   .
SELECT
    concat(database, '.', table) AS schema_table,
    partition,
    sum(rows) AS rows,
    formatReadableSize(sum(bytes_on_disk) AS bytes_on_disk) AS size_on_disk,
    formatReadableSize(sum(data_uncompressed_bytes) AS uncompressed_bytes) AS uncompressed_size,
    round(uncompressed_bytes / bytes_on_disk, 2) AS comprassion_ratio
FROM system.parts
WHERE active AND schema_table = required_schema_table
GROUP BY
    schema_table,
    partition
ORDER BY partition DESC
;
```

В разрезе колонок

```
WITH
    'analytics.line_items' AS required_schema_table, -- <---   .
    parts_columnns_cte AS
    (
        SELECT
            concat(database, '.', table) AS schema_table,
            column,
            type,
            sum(column_bytes_on_disk) AS column_bytes_on_disk,
            sum(column_data_uncompressed_bytes) AS column_data_uncompressed_bytes,
            sum(rows) as rows
        FROM system.parts_columns
        WHERE active AND (schema_table = required_schema_table)
        GROUP BY
            schema_table,
            type,
            column
    )
SELECT
    schema_table,
    column,
    type,
    rows,
    formatReadableSize(column_bytes_on_disk) as size_on_disk,
    formatReadableSize(column_data_uncompressed_bytes) as uncompressed_size,
    round(column_data_uncompressed_bytes/column_bytes_on_disk, 2) as comprassion_ratio,
    round((column_bytes_on_disk / (SELECT sum(column_bytes_on_disk) FROM parts_columnns_cte) * 100), 2) AS
percent_of_total_size
FROM parts_columnns_cte
ORDER BY column_bytes_on_disk DESC
;
```

Если engine Distributed (может не хватить прав, в таком случае обратись в dwh-support)

```
--      event.new_app, gp_rep.rep__first_shipments_rte
--      Distributed,         , :

SHOW CREATE event.new_app
--   ddl ENGINE = Distributed('shard_group_event', 'event', 'new_app__shard_20220224'),
--       - shard_group_event     event.new_app__shard_20220224

SHOW CREATE gp_rep.rep__first_shipments_rte
--   ddl ENGINE = Distributed('shard_group_data01', 'gp_rep', 'rep__first_shipments_rte_shard'),
--       - shard_group_data01     gp_rep.rep__first_shipments_rte_shard


WITH ['event.new_app__shard_20220224', ''] AS required_schema_tables -- <---   .
SELECT
    concat(database, '.', table) AS schema_table,
    sum(rows) AS rows,
    formatReadableSize(sum(bytes_on_disk) as bytes_on_disk) AS size_on_disk,
    formatReadableSize(sum(data_uncompressed_bytes) as uncompressed_bytes) AS uncompressed_size,
    round(uncompressed_bytes/bytes_on_disk, 2) as comprassion_ratio
FROM clusterAllReplicas('shard_group_event', system.parts) -- <---
WHERE active AND schema_table in required_schema_tables
GROUP BY
    schema_table
ORDER BY bytes_on_disk DESC
;

WITH ['gp_rep.rep__first_shipments_rte_shard', ''] AS required_schema_tables -- <---   .
SELECT
    concat(database, '.', table) AS schema_table,
    sum(rows) AS rows,
    formatReadableSize(sum(bytes_on_disk) as bytes_on_disk) AS size_on_disk,
    formatReadableSize(sum(data_uncompressed_bytes) as uncompressed_bytes) AS uncompressed_size,
    round(uncompressed_bytes/bytes_on_disk, 2) as comprassion_ratio
FROM clusterAllReplicas('shard_group_data01', system.parts) -- <---
WHERE active AND schema_table in required_schema_tables
GROUP BY
    schema_table
ORDER BY bytes_on_disk DESC
;
```

Изменение DDL таблицы и подготовка кода для MR в CH-схему

Изменение столбцов

Переименование столбца

```
--   ,
ALTER TABLE schema.table_name ON CLUSTER 'shard_group_old'
RENAME COLUMN old_column_name TO new_column_name;

--   MR  CH-schema
DROP TABLE IF EXISTS stage.schema__table_name ON CLUSTER 'shard_group_old';
```

Изменение типа столбца

```
--    ,     .    ,      . (  ,  ,  ).

show create schema.table_name;

CREATE TABLE sandbox_dev.schema__table_name_backup --  ON CLUSTER
[   ]
Engine = MergeTree --  Replicated
PARTITION BY [   ]
ORDER BY [  ];

--   ,
EXPLAIN SYNTAX
SELECT *
FROM schema.table_name

INSERT INTO sandbox_dev.schema__table_name_backup [ ]
SELECT
        col_1,
        ...,
        CAST(col_n AS new_type),
        ...
FROM schema.table_name;

DROP TABLE schema.table_name ON CLUSTER 'shard_group_old';

-- Stage        .
DROP TABLE IF EXISTS stage.schema__table_name ON CLUSTER 'shard_group_old';

-- merge MR

ALTER TABLE schema.table_name
REPLACE PARTITION '[ ,   ]'
FROM sandbox_dev.schema__table_name_backup;
--    ,      Insert,        .
/*
INSERT INTO schema.table_name [ ]
SELECT [ ]
FROM sandbox_dev.schema__table_name_backup;
*/

-- <> ,      ,

DROP TABLE sandbox_dev.schema__table_name_backup;
```

Удаление столбца

```
--   ,
ALTER TABLE schema.table_name ON CLUSTER 'shard_group_old'
DROP COLUMN column_name;

--   MR  CH-schema
DROP TABLE IF EXISTS stage.schema__table_name ON CLUSTER 'shard_group_old';
```

Изменение таблицы

Переименование таблицы

```
--    ,   ZooKeeper path.    ,      . (  ,  ,  ).

show create schema.old_table_name;

CREATE TABLE sandbox_dev.schema__old_table_name_backup --  ON CLUSTER
[  ]
Engine = MergeTree --  Replicated
PARTITION BY [   ]
ORDER BY [  ];

ALTER TABLE sandbox_dev.schema__old_table_name_backup
REPLACE PARTITION '[ ,   ]'
FROM schema.old_table_name;
--    ,      Insert,        .
--   ,
EXPLAIN SYNTAX
SELECT *
FROM schema.table_name
/*
INSERT INTO sandbox_dev.schema__old_table_name_backup [ ]
SELECT [ ]
FROM schema.old_table_name;
*/

DROP TABLE schema.old_table_name ON CLUSTER 'shard_group_old';

-- Stage        .
DROP TABLE IF EXISTS stage.schema__table_name ON CLUSTER 'shard_group_old';

-- merge MR

ALTER TABLE schema.new_table_name
REPLACE PARTITION '[ ,   ]'
FROM sandbox_dev.schema__old_table_name_backup;
--    ,      Insert,        .
/*
INSERT INTO schema.new_table_name [ ]
SELECT [ ]
FROM sandbox_dev.schema__old_table_name_backup;
*/

-- <> ,      ,

DROP TABLE sandbox_dev.schema__old_table_name_backup;
```

Изменение параметров таблицы (партиционирование, сортировка, движок)

```
--     ,     .    ,      . (  ,  ,  ).

show create schema.table_name;

CREATE TABLE sandbox_dev.schema__table_name_new_params --  ON CLUSTER
[  ]
Engine = MergeTree/ReplacingMergeTree --   ,   Replicated
PARTITION BY [  ]
ORDER BY [  ];

--   ,
EXPLAIN SYNTAX
SELECT *
FROM schema.table_name

INSERT INTO sandbox_dev.schema__table_name_new_params [ ]
SELECT [ ]
FROM schema.table_name;

DROP TABLE schema.table_name ON CLUSTER 'shard_group_old';

-- Stage        .
DROP TABLE IF EXISTS stage.schema__table_name ON CLUSTER 'shard_group_old';

-- merge MR

ALTER TABLE schema.table_name
REPLACE PARTITION '[ ,   ]'
FROM sandbox_dev.schema__table_name_new_params;
--    ,      Insert,        .
/*
INSERT INTO schema.table_name [ ]
SELECT [ ]
FROM sandbox_dev.schema__table_name_new_params;
*/

-- <> ,      ,

DROP TABLE sandbox_dev.schema__table_name_new_params;
```

Удаление таблицы

```
DROP TABLE schema.old_table_name ON CLUSTER 'shard_group_old';

DROP TABLE IF EXISTS stage.schema__table_name ON CLUSTER 'shard_group_old';
```

Изменение распределенных таблиц (схема gp_rep)

Дополнительные действия для Distributed таблиц

```
--  Distributed    gp_rep      3 :
-- 1.    .     ,  gp_rep  'shard_group_data01'.          (     ).
--  ,
--    ,    " "
DROP TABLE IF EXISTS gp_rep.table_name_shard ON CLUSTER 'shard_group_data01';

-- 2. -     stage   .         .

DROP TABLE IF EXISTS stage.gp_rep__table_name_shard ON CLUSTER 'shard_group_data01';

DROP TABLE IF EXISTS gp_rep.table_name ON CLUSTER 'shard_group_old';
```