Отчет к ДЗ Репликация и удаление

1. Взять любой демонстрационный DATASET, не обязательно полный набор данных: https://clickhouse.com/docs/en/getting-started/example-datasets
2. Конвертировать таблицу в реплицируемую, используя макрос replica
3. Добавить 2 реплики
4. Отдать результаты запросов как 2 файла

    SELECT

    getMacro('replica'),

    *

    FROM remote('разделенный запятыми список реплик',system.parts)

    FORMAT JSONEachRow;

    SELECT * FROM system.replicas FORMAT JSONEachRow;

5. Добавить/выбрать колонку с типом Date в таблице, добавить TTL на таблицу «хранить последние 7 дней». Предоставить результат запроса «SHOW CREATE TABLE таблица» на проверку.

1. Создаем таблицу из тестовых датасетов и наполняем ее

```
53d58e10c49a :) CREATE TABLE uk_price_paid
(
    price UInt32,
    date Date,
    postcode1 LowCardinality(String),
    postcode2 LowCardinality(String),
    type Enum8('terraced' = 1, 'semi-detached' = 2, 'detached' = 3, 'flat' = 4, 'other' = 0),
    is_new UInt8,
    duration Enum8('freehold' = 1, 'leasehold' = 2, 'unknown' = 0),
    addr1 String,
    addr2 String,
    street LowCardinality(String),
    locality LowCardinality(String),
    town LowCardinality(String),
    district LowCardinality(String),
    county LowCardinality(String)
)
ENGINE = MergeTree
ORDER BY (postcode1, postcode2, addr1, addr2);

CREATE TABLE uk_price_paid
(
    `price` UInt32,
    `date` Date,
    `postcode1` LowCardinality(String),
    `postcode2` LowCardinality(String),
    `type` Enum8('terraced' = 1, 'semi-detached' = 2, 'detached' = 3, 'flat' = 4, 'other' = 0),
    `is_new` UInt8,
    `duration` Enum8('freehold' = 1, 'leasehold' = 2, 'unknown' = 0),
    `addr1` String,
    `addr2` String,
    `street` LowCardinality(String),
    `locality` LowCardinality(String),
    `town` LowCardinality(String),
    `district` LowCardinality(String),
    `county` LowCardinality(String)
)
ENGINE = MergeTree
ORDER BY (postcode1, postcode2, addr1, addr2)

Query id: b772afc5-4cd6-4af7-94f2-243f63c9d8ea

Ok.

0 rows in set. Elapsed: 0.010 sec.
```

```
INSERT INTO uk_price_paid
SETTINGS max_http_get_redirects = 10
WITH splitByChar(' ', postcode) AS p
SELECT
    toUInt32(price_string) AS price,
    parseDateTimeBestEffortUS(time) AS date,
    p[1] AS postcode1,
    p[2] AS postcode2,
    transform(a, ['T', 'S', 'D', 'F', 'O'], ['terraced', 'semi-detached', 'detached', 'flat', 'other']) AS type,
    b = 'Y' AS is_new,
    transform(c, ['F', 'L', 'U'], ['freehold', 'leasehold', 'unknown']) AS duration,
    addr1,
    addr2,
    street,
    locality,
    town,
    district,
    county
FROM url('http://prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com/pp-complete.csv', 'CSV', 'uuid_string String,\n    price_s
,\n    locality String,\n    town String,\n    district String,\n    county String,\n    d String,\n    e String')
SETTINGS max_http_get_redirects = 10

Query id: 4a51dc16-27c8-4c9d-a8db-67ab1ff74fbf

↑ Progress: 1.85 million rows, 325.06 MB (211.03 thousand rows/s., 37.02 MB/s.)
Press the space key to toggle the display of the progress table.
```

```
SELECT count(*)
FROM uk_price_paid

Query id: 9ed57d8f-dcae-447a-a33b-fdbf0165dd0a

   ┌─count()─┐
1. │29946800 │  -- 29.95 million
   └─────────┘

1 row in set. Elapsed: 0.003 sec.

53d58e10c49a :)
```

2. Конвертируем таблицу в реплицируемую и аттачим в нее партиции из обычной таблицы:

```
CREATE TABLE uk_price_paid_repl AS uk_price_paid
ENGINE = ReplicatedMergeTree('/clickhouse/{cluster}/{table}', '{replica}')

Query id: 5aadcee0-4812-4f33-97fd-84469700bf67

Ok.

0 rows in set. Elapsed: 0.075 sec.

53d58e10c49a :) ALTER TABLE uk_price_paid_repl ATTACH PARTITION ID 'all' FROM uk_price_paid;

ALTER TABLE uk_price_paid_repl
    (ATTACH PARTITION ID 'all' FROM uk_price_paid)

Query id: 968c29e1-7ad0-4d1d-a4da-6411e0875615

Ok.

0 rows in set. Elapsed: 0.071 sec.

53d58e10c49a :) select count(*) from uk_price_paid_repl;

SELECT count(*)
FROM uk_price_paid_repl

Query id: a7f08fa8-443f-4842-b551-9c6609cf967d

    ┌──count()─┐
1.  │ 29946800 │   -- 29.95 million
    └──────────┘

1 row in set. Elapsed: 0.003 sec.
```

3. Из той же ноды создаю таблицы на двух других нодах и проверяем что туда данные реплицировались:

```
CREATE TABLE IF NOT EXISTS uk_price_paid_repl ON CLUSTER `{cluster}`
(
    `price` UInt32,
    `date` Date,
    `postcode1` LowCardinality(String),
    `postcode2` LowCardinality(String),
    `type` Enum8('terraced' = 1, 'semi-detached' = 2, 'detached' = 3, 'flat' = 4, 'other' = 0),
    `is_new` UInt8,
    `duration` Enum8('freehold' = 1, 'leasehold' = 2, 'unknown' = 0),
    `addr1` String,
    `addr2` String,
    `street` LowCardinality(String),
    `locality` LowCardinality(String),
    `town` LowCardinality(String),
    `district` LowCardinality(String),
    `county` LowCardinality(String)
)
ENGINE = ReplicatedMergeTree('/clickhouse/{cluster}/{table}', '{replica}')
ORDER BY (postcode1, postcode2, addr1, addr2)

Query id: 122ec04b-4e8f-4ab4-aa98-dc1e8f6245fa
```

host	port	status	error	num_hosts_remaining	num_hosts_active
1. clickhouse1	9000	0		2	2

host	port	status	error	num_hosts_remaining	num_hosts_active
2. clickhouse2	9000	0		1	0
3. clickhouse3	9000	0		0	0

```
3 rows in set. Elapsed: 0.137 sec.

53d58e10c49a :)
```

```
c6866125ae81 :) show tables

SHOW TABLES

Query id: fc26d1f5-cfef-4bc7-a2f7-a43fffd0a082
```

name
1. example
2. uk_price_paid_repl

```
2 rows in set. Elapsed: 0.003 sec.

c6866125ae81 :) select COUNT(*) from uk_price_paid_repl;

SELECT COUNT(*)
FROM uk_price_paid_repl

Query id: 8316d49c-d8ac-4106-9fa8-7711c93c7e28
```

COUNT()
1. 29946800	-- 29.95 million

```
1 row in set. Elapsed: 0.003 sec.

c6866125ae81 :)
```

```
SHOW TABLES

Query id: 31e30913-52ec-4a27-88a0-74262c8e91db
```

name
1. example
2. uk_price_paid_repl

```
2 rows in set. Elapsed: 0.002 sec.

deffb0ff20ce :) select count(*) FROM uk_price_paid_repl;

SELECT count(*)
FROM uk_price_paid_repl

Query id: 14a0879e-e9e9-40b8-85e6-28d416ecb640
```

count()
1. 29946800	-- 29.95 million

```
1 row in set. Elapsed: 0.002 sec.

deffb0ff20ce :)
```

4. Файлы как результаты запроса:

output_1.json   output_2.json

5. Изменение таблицы. Добавлчем в нее TTL на последние 7 дней:

```
53d58e10c49a :) ALTER TABLE uk_price_paid_repl on cluster '{cluster}' MODIFY TTL date + INTERVAL 7 DAY;

ALTER TABLE uk_price_paid_repl ON CLUSTER `{cluster}`
    (MODIFY TTL date + toIntervalDay(7))

Query id: 10c67796-b8a9-4967-baf6-357d1fd051ca

   ┌─host────────┬─port─┬─status─┬─error─┬─num_hosts_remaining─┬─num_hosts_active─┐
1. │ clickhouse2 │ 9000 │      0 │       │                   2 │                0 │
2. │ clickhouse3 │ 9000 │      0 │       │                   1 │                0 │
3. │ clickhouse1 │ 9000 │      0 │       │                   0 │                0 │
   └─────────────┴──────┴────────┴───────┴─────────────────────┴──────────────────┘

3 rows in set. Elapsed: 8.601 sec.

53d58e10c49a :)
```

```
SHOW CREATE TABLE uk_price_paid_repl
FORMAT Native

Query id: 023b602d-c859-4266-a037-37232718be7d

The requested output format `Native` is binary and could produce side-effects when output directly into the terminal.
If you want to output it into a file, use the "INTO OUTFILE" modifier in the query or redirect the output of the shell command.
Do you want to output it anyway? [y/N] Y

       statementString◆CREATE TABLE default.uk_price_paid_repl
(
    `price` UInt32,
    `date` Date,
    `postcode1` LowCardinality(String),
    `postcode2` LowCardinality(String),
    `type` Enum8('other' = 0, 'terraced' = 1, 'semi-detached' = 2, 'detached' = 3, 'flat' = 4),
    `is_new` UInt8,
    `duration` Enum8('unknown' = 0, 'freehold' = 1, 'leasehold' = 2),
    `addr1` String,
    `addr2` String,
    `street` LowCardinality(String),
    `locality` LowCardinality(String),
    `town` LowCardinality(String),
    `district` LowCardinality(String),
    `county` LowCardinality(String)
)
ENGINE = ReplicatedMergeTree('/clickhouse/{cluster}/uk_price_paid_repl', '{replica}')
ORDER BY (postcode1, postcode2, addr1, addr2)
TTL date + toIntervalDay(7)
SETTINGS index_granularity = 8192

1 row in set. Elapsed: 0.002 sec.

c6866125ae81 :)
```

```
SHOW CREATE TABLE uk_price_paid_repl

Query id: fede78e2-80cc-4b2c-81d0-0dc3b99822df

   ┌─statement─────────────────────────────────────────────────────────────────────┐
1. │ CREATE TABLE default.uk_price_paid_repl                                         │
   │ (                                                                              │
   │     `price` UInt32,                                                            │
   │     `date` Date,                                                              │
   │     `postcode1` LowCardinality(String),                                       │
   │     `postcode2` LowCardinality(String),                                       │
   │     `type` Enum8('other' = 0, 'terraced' = 1, 'semi-detached' = 2, 'detached' = 3, 'flat' = 4),│
   │     `is_new` UInt8,                                                            │
   │     `duration` Enum8('unknown' = 0, 'freehold' = 1, 'leasehold' = 2),         │
   │     `addr1` String,                                                           │
   │     `addr2` String,                                                           │
   │     `street` LowCardinality(String),                                          │
   │     `locality` LowCardinality(String),                                        │
   │     `town` LowCardinality(String),                                            │
   │     `district` LowCardinality(String),                                        │
   │     `county` LowCardinality(String)                                           │
   │ )                                                                              │
   │ ENGINE = ReplicatedMergeTree('/clickhouse/{cluster}/uk_price_paid_repl', '{replica}')│
   │ ORDER BY (postcode1, postcode2, addr1, addr2)                                 │
   │ TTL date + toIntervalDay(7)                                                   │
   │ SETTINGS index_granularity = 8192                                             │
   └────────────────────────────────────────────────────────────────────────────────┘

1 row in set. Elapsed: 0.001 sec.

c6866125ae81 :)
```