

Примеры работы с ClickHouse: движки таблиц, партиции и представления

Подготовка окружения

Запуск ClickHouse через Docker Compose

```
export CHVER=23.4
export CHKVER=23.4-alpine
docker compose up
```

1. ReplacingMergeTree с партиционированием по минутам

Создание таблицы с партиционированием и TTL

```
-- Создаем таблицу с движком ReplacingMergeTree
-- Партиционирование по минутам для демонстрации работы с партициями
CREATE TABLE IF NOT EXISTS web_events (
    event_id UInt64,
    user_id UInt32,
    session_id String,
    event_time DateTime,
    event_type Enum8('page_view' = 1, 'click' = 2, 'scroll' = 3, 'purchase' = 4),
    page_url String,
    country String,
    device_type String,
    revenue Nullable(Float32)
) ENGINE = ReplacingMergeTree() -- Автоматически заменяет дубликаты
PARTITION BY toYYYYMMDDhhmmss(event_time) -- Партиционирование по минутам
ORDER BY (user_id, event_id) -- Ключ сортировки определяет, что считается дубликатом
TTL event_time + INTERVAL 10000 SECOND DELETE -- Данные удаляются через 10000 секунд
SETTINGS merge_with_ttl_timeout = 300; -- Слияние с TTL каждые 5 минут
```

Вставка данных в разные партиции (разные минуты)

```
-- Вставляем данные за разные минуты для создания разных партиций
INSERT INTO web_events VALUES
-- Партиция 1: 10:00:xx
(1, 1001, 'sess_001', '2024-06-01 10:00:15', 'page_view', '/home', 'Russia', 'desktop', NULL),
(2, 1001, 'sess_001', '2024-06-01 10:00:45', 'click', '/products', 'Russia', 'desktop', NULL),

-- Партиция 2: 10:01:xx
(3, 1002, 'sess_002', '2024-06-01 10:01:20', 'page_view', '/home', 'USA', 'mobile', NULL),
(4, 1001, 'sess_001', '2024-06-01 10:01:50', 'page_view', '/products/laptop', 'Russia', 'desktop', NULL),

-- Партиция 3: 10:02:xx
(5, 1003, 'sess_003', '2024-06-01 10:02:10', 'page_view', '/about', 'Germany', 'tablet', NULL),
(6, 1001, 'sess_001', '2024-06-01 10:02:30', 'purchase', '/checkout', 'Russia', 'desktop', 1299.99),

-- Партиция 4: 10:03:xx
(7, 1004, 'sess_004', '2024-06-01 10:03:00', 'page_view', '/home', 'France', 'mobile', NULL),
(8, 1002, 'sess_002', '2024-06-01 10:03:25', 'click', '/products/phone', 'USA', 'mobile', NULL);

-- Вставляем дубликат (с тем же user_id и event_id, но другими данными)
INSERT INTO web_events VALUES
(1, 1001, 'sess_001', '2024-06-01 10:00:15', 'page_view', '/home-updated', 'Russia', 'desktop', NULL);
```

Запросы к ReplacingMergeTree

```

-- Просмотр созданных партиций
SELECT
    partition,
    name,
    rows,
    bytes_on_disk,
    modification_time
FROM system.parts
WHERE table = 'web_events'
    AND database = currentDatabase()
    AND active = 1
ORDER BY partition;

-- Остановка автоматических слияний для демонстрации дубликатов
SYSTEM STOP MERGES web_events;

-- Просмотр данных с дубликатами (слияния остановлены)
SELECT * FROM web_events
WHERE user_id = 1001 AND event_id = 1
ORDER BY event_time;

-- Запуск слияний
SYSTEM START MERGES web_events;

-- Принудительное слияние партиций для применения дедупликации
OPTIMIZE TABLE web_events;

-- Просмотр данных с FINAL - гарантированно без дубликатов
SELECT * FROM web_events FINAL
ORDER BY event_time;

```

2. Простые представления (Views)

Создание простого представления

```

-- Создаем представление для упрощения работы с данными
CREATE VIEW user_stats AS
SELECT
    user_id,
    country,
    COUNT() as total_events,
    countIf(event_type = 'page_view') as page_views,
    countIf(event_type = 'click') as clicks,
    countIf(event_type = 'purchase') as purchases,
    sum(revenue) as total_revenue
FROM web_events FINAL
GROUP BY user_id, country;

```

Использование простого представления

```
-- Запрос к представлению работает как к обычной таблице
SELECT * FROM user_stats
WHERE total_revenue > 100
ORDER BY total_revenue DESC;

-- Агрегация поверх представления
SELECT
    country,
    avg(total_events) as avg_events_per_user,
    sum(total_revenue) as country_revenue
FROM user_stats
GROUP BY country
ORDER BY country_revenue DESC;
```

3. Работа с партициями: ATTACH и REPLACE

Создание временной таблицы для демонстрации

```
-- Создаем временную таблицу с той же структурой
CREATE TABLE web_events_temp AS web_events;

-- Вставляем данные в новую партицию
INSERT INTO web_events_temp VALUES
(20, 2001, 'sess_020', '2024-06-01 10:05:30', 'page_view', '/new-page', 'Italy', 'desktop', NULL),
(21, 2002, 'sess_021', '2024-06-01 10:05:45', 'click', '/new-product', 'Italy', 'mobile', 150.00);
```

ATTACH PARTITION - присоединение партиции

```
-- Отсоединяем партицию из временной таблицы
ALTER TABLE web_events_temp DETACH PARTITION 20240601100530;

-- Проверяем, что партиция отсоединена
SELECT count() FROM web_events_temp;

-- Присоединяем партицию к основной таблице
ALTER TABLE web_events ATTACH PARTITION 20240601100530 FROM web_events_temp;

-- Проверяем, что данные появились в основной таблице
SELECT * FROM web_events
WHERE toYYYYMMDDhhmmss(event_time) = 20240601100530;
```

REPLACE PARTITION - замена партиции

```
-- Создаем новые данные для замены существующей партиции
INSERT INTO web_events_temp VALUES
(100, 1001, 'sess_new', '2024-06-01 10:00:10', 'page_view', '/replaced-home', 'Russia', 'mobile', NULL),
(101, 1001, 'sess_new', '2024-06-01 10:00:20', 'purchase', '/replaced-checkout', 'Russia', 'mobile', 999.99);

-- Смотрим текущие данные в партиции
SELECT * FROM web_events
WHERE toYYYYMMDDhhmmss(event_time) = 20240601100015;

-- Заменяем партицию данными из временной таблицы
ALTER TABLE web_events REPLACE PARTITION 20240601100015 FROM web_events_temp;

-- Проверяем замененные данные
SELECT * FROM web_events
WHERE toYYYYMMDDhhmmss(event_time) = 20240601100010;
```

Домашнее задание

Работа с ReplacingMergeTree и переносом партиций

Задание: Создать две таблицы с ReplacingMergeTree, заполнить данными и перенести партиции между таблицами.

Требования к выполнению:

1. Создание таблиц:

- Создать основную таблицу `events_main` с ReplacingMergeTree
- Создать архивную таблицу `events_archive` с аналогичной структурой
- Обе таблицы должны иметь партиционирование по минутам
- Поля: `event_id`, `user_id`, `event_time`, `event_type`, `page_url`, `country`

2. Заполнение данными:

- Вставить данные минимум в 3 разные партиции (разные минуты)
- Добавить дубликаты для демонстрации ReplacingMergeTree
- Минимум 6 записей в основной таблице

3. Перенос партиций:

- Скопировать одну партицию в архивную таблицу через `ATTACH PARTITION FROM`
- Удалить скопированную партицию из исходной таблицы через `DROP PARTITION`
- Повторить для второй партиции

4. Написать проверочные запросы:

- Запрос, показывающий партиции в обеих таблицах после переноса
- Запрос, проверяющий количество записей в каждой таблице