

2. Семейство движков *MergeTree

- [Вводная часть про семейство MergeTree](#)
- [Replicated*MergeTree](#)
- [ReplacingMergeTree](#)
- [Параметры *MergeTree: PARTITION BY, ORDER BY, TTL, типы колонок, кодеки, вторичные индексы](#)

Вводная часть про семейство MergeTree

Подробная [документация ClickHouse](#).

Движок **MergeTree**, а также другие движки этого семейства (*MergeTree) — это наиболее функциональные движки таблиц ClickHouse.

Основная идея, заложенная в основу движков семейства MergeTree следующая: когда у вас есть огромное количество данных, которые должны быть вставлены в таблицу, вы должны быстро записать их по частям, а затем объединить части по некоторым правилам в фоновом режиме. Этот метод намного эффективнее, чем постоянная перезапись данных в хранилище при вставке.

Основные возможности:

- **Хранит данные, отсортированные по первичному ключу (ORDER BY).** Это позволяет создавать разреженный индекс небольшого объёма, который позволяет быстрее находить данные.
- **Позволяет оперировать партициями, если задан [ключ партиционирования](#).** ClickHouse поддерживает отдельные операции с партициями, которые работают эффективнее, чем общие операции с этим же результатом над этими же данными. Также, ClickHouse автоматически отсекает данные по партициям там, где ключ партиционирования указан в запросе. Благодаря этому запросы могут работать быстрее и потреблять меньше ресурсов.
- **Поддерживает репликацию данных.** Для этого используется семейство таблиц Replicated*MergeTree. [Подробнее про репликацию](#).

Данные хранятся по колонкам. [О колоночном хранении](#). Для того, чтобы достать несколько полей из таблицы, базе не нужно читать всю строку целиком, а только нужные колонки.

Именно поэтому не нужно использовать `SELECT * FROM table`.

Replicated*MergeTree



Для понимания этой части, важно ознакомиться с [2. Физическая архитектура аналитического CH](#)

В ClickHouse есть возможность репликации данных между нодами в шард-группе. Другими словами, таблица может синхронизировать изменения и быть одинаковой на разных серверах(нодах) ClickHouse.

Репликация работает на уровне отдельных таблиц, а не всего сервера(ноды). То есть, на сервере могут быть расположены одновременно реплицируемые и не реплицируемые таблицы.

Репликация нужна для сохранности данных и возможности работы с таблицами с разных нод.

Движок поддерживает репликацию, если у него в начале названия движка есть ключевое слово **Replicated**. Например **ReplicatedMergeTree**, **ReplicatedReplacingMergeTree**.

Replicated* движки обязательно принимают параметр пути до таблицы на репликах.

Пример движка

```
ENGINE = ReplicatedMergeTree('/tables/{shard}/analytics/shipments', '{replica}')
```

```
--
```

Replicated* таблица должна быть создана на всех нодах внутри шард-группы. Для этого нужно добавить модификатор `ON CLUSTER`. [Подробнее про ON CLUSTER](#).

Пример CREATE ON CLUSTER

```
CREATE TABLE analytics.line_items ON CLUSTER shard_group_old
...
ENGINE = ReplicatedMergeTree('/tables/{shard}/analytics/line_items', '{replica}')
ORDER BY ...
```

ReplacingMergeTree

[Из документации ClickHouse](#)

Движок отличается от MergeTree тем, что выполняет удаление дублирующихся записей с одинаковым значением ключа сортировки (секция `ORDER BY`).

Дедупликация данных производится лишь во время слияний. Слияние происходит в фоне в неизвестный момент времени, на который вы не можете ориентироваться. Некоторая часть данных может остаться необработанной.

Таким образом, **ReplacingMergeTree** подходит для фоновой чистки дублирующихся данных в целях экономии места, но не даёт гарантии отсутствия дубликатов.

В параметры движка можно указать `ver` – столбец с номером версии. В результате слияния дублей, останется строка с наибольшим полем `ver`. [Подробнее про ver](#).



Если дубли по ключу `ORDER BY` находятся в разных партициях, то они никогда не дедуплицируются.

Параметры *MergeTree: PARTITION BY, ORDER BY, TTL, типы колонок, кодеки, вторичные индексы