

Код группы: 50916kek

Подключимся к лабораторной ВМ по SSH:

```
ssh administrator@kgsu.demonstrations.com -p 9999
```

Создадим БД с названием содержащим код своей группы:

```
clickhouse-client --query "CREATE DATABASE IF NOT EXISTS 50916kek"
```

Создадим таблицу hits_v1:

```
CREATE TABLE 50916kek.hits_v1 ( WatchID UInt64, JavaEnable UInt8, Title String, GoodEvent Int16, EventTime DateTime, EventDate Date, CounterID UInt32, ClientIP UInt32, ClientIP6 FixedString(16), RegionID UInt32, UserID UInt64, CounterClass Int8, OS UInt8, UserAgent UInt8, URL String, Referer String, URLEntry String, RefererDomain String, Refresh UInt8, IsRobot UInt8, RefererCategories Array(UInt16), URLEntries Array(UInt16), URLEntryRegions Array(UInt32), ResolutionWidth UInt16, ResolutionHeight UInt16, ResolutionDepth UInt8, FlashMajor UInt8, FlashMinor UInt8, FlashMinor2 String, NetMajor UInt8, NetMinor UInt8, UserAgentMajor UInt16, UserAgentMinor FixedString(2), CookieEnable UInt8, JavascriptEnable UInt8, IsMobile UInt8, MobilePhone UInt8, MobilePhoneModel String, Params String, IPNetworkID UInt32, TrafficSourceID Int8, SearchEngineID UInt16, SearchPhrase String, AdvEngineID UInt8, IsArtificial UInt8, WindowClientWidth UInt16, WindowClientHeight UInt16, ClientTimeZone Int16, ClientEventTime DateTime, SilverlightVersion1 UInt8, SilverlightVersion2 UInt8, SilverlightVersion3 UInt32, SilverlightVersion4 UInt16, PageCharset String, CodeVersion UInt32, IsLink UInt8, IsDownload UInt8, IsNotBounce UInt8, FUniqID UInt64, HID UInt32, IsOldCounter UInt8, IsEvent UInt8, IsParameter UInt8, DontCountHits UInt8, WithHash UInt8, HitColor FixedString(1), UTCEventTime DateTime, Age UInt8, Sex UInt8, Income UInt8, Interests UInt16, Robotness UInt8, GeneralInterests Array(UInt16), RemoteIP UInt32, RemoteIP6 FixedString(16), WindowName Int32, OpenerName Int32, HistoryLength Int16, BrowserLanguage FixedString(2), BrowserCountry FixedString(2), SocialNetwork String, SocialAction String, HTTPError UInt16, SendTiming Int32, DNSTiming Int32, ConnectTiming Int32, ResponseStartTiming Int32, ResponseEndTiming Int32, FetchTiming Int32, RedirectTiming Int32, DOMInteractiveTiming Int32, DOMContentLoadedTiming Int32, DOMCompleteTiming Int32, LoadEventStartTiming Int32, LoadEventEndTiming Int32, NSToDOMContentLoadedTiming Int32, FirstPaintTiming Int32, RedirectCount Int8, SocialSourceNetworkID UInt8, SocialSourcePage String, ParamPrice Int64, ParamOrderID String, ParamCurrency FixedString(3), ParamCurrencyID UInt16, GoalsReached
```

Array(UInt32), OpenstatServiceName String, OpenstatCampaignID String, OpenstatAdID String, OpenstatSourceID String, UTMSource String, UTMMedium String, UTMCampaign String, UTMContent String, UTMTerm String, FromTag String, HasGCLID UInt8, RefererHash UInt64, URLHash UInt64, CLID UInt32, YCLID UInt64, ShareService String, ShareURL String, ShareTitle String, ParsedParams Nested(Key1 String, Key2 String, Key3 String, Key4 String, Key5 String, ValueDouble Float64), IslandID FixedString(16), RequestNum UInt32, RequestTry UInt8) ENGINE = MergeTree() PARTITION BY toYYYYMM(EventDate) ORDER BY (CounterID, EventDate, intHash32(UserID)) SAMPLE BY intHash32(UserID) SETTINGS index_granularity = 8192

Вставим в таблицу 10000 строк:

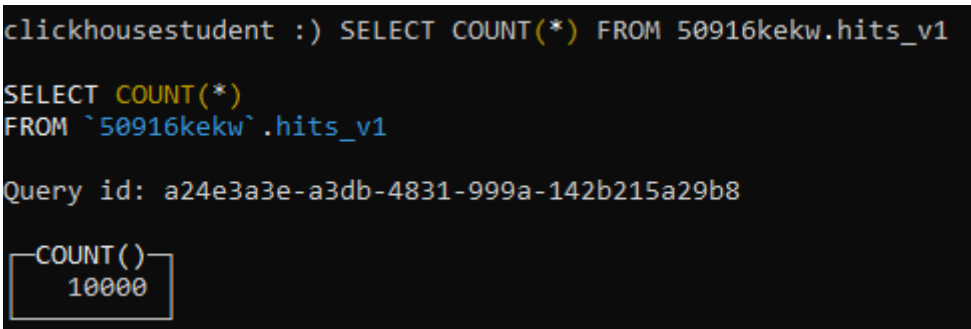
```
head --lines=10000 hits_v1.tsv | clickhouse-client --user default --password
qwerty12345 --query "INSERT INTO 50916kekw.hits_v1 FORMAT TSV" --
max_insert_block_size=100000
```

Зайдем в кликхаус:

```
clickhouse-client --user default --password qwerty12345
```

Посмотрим количество строк в таблице:

```
SELECT COUNT(*) FROM 50916kekw.hits_v1
```



The screenshot shows a terminal window with the following text:

```
clickhousestudent :) SELECT COUNT(*) FROM 50916kekw.hits_v1
SELECT COUNT(*)
FROM `50916kekw`.hits_v1
Query id: a24e3a3e-a3db-4831-999a-142b215a29b8
COUNT()
10000
```

Добавим еще 10000 строк, но уже с конца:

```
tail --lines=10000 hits_v1.tsv | clickhouse-client --user default --password
qwerty12345 --query "INSERT INTO 50916kekw.hits_v1 FORMAT TSV" --
max_insert_block_size=100000
```

Опять посмотрим количество строк в таблице:

```
clickhousestudent :) SELECT COUNT(*) FROM 50916kekw.hits_v1

SELECT COUNT(*)
FROM `50916kekw`.hits_v1

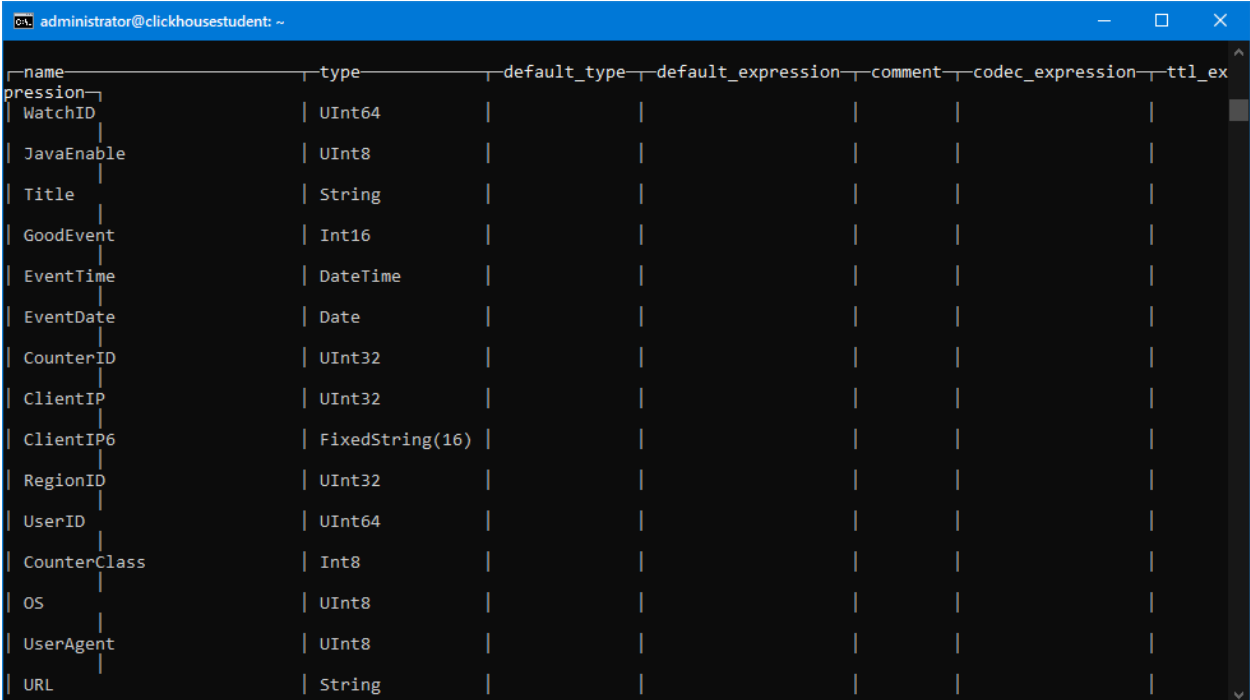
Query id: 229683a0-716d-47ae-a7d4-59da486d4730

COUNT()
20000

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

Посмотрим структуру таблицы:

DESCRIBE TABLE 50916kekw.hits_v1



name	type	default_type	default_expression	comment	codec_expression	ttl_ex
pression						
WatchID	UInt64					
JavaEnable	UInt8					
Title	String					
GoodEvent	Int16					
EventTime	DateTime					
EventDate	Date					
CounterID	UInt32					
ClientIP	UInt32					
ClientIP6	FixedString(16)					
RegionID	UInt32					
UserID	UInt64					
CounterClass	Int8					
OS	UInt8					
UserAgent	UInt8					
URL	String					

Можно создать секции, например, по CounterID, URL, EventDate, Age – так как одинаковых значений в таблице будет много.

Не подходит для создания секций, например, WatchID – так как он разный для каждой записи

Узнаем размеры таблицы:

SELECT formatReadableSize(sum(bytes)) AS size, sum(rows) AS rows FROM system.parts WHERE active and database = '50916kekw'

```
clickhousestudent :) SELECT formatReadableSize(sum(bytes)) AS size, sum(rows) AS rows FROM system.parts WHERE active and database = '50916kek'
SELECT
    formatReadableSize(sum(bytes)) AS size,
    sum(rows) AS rows
FROM system.parts
WHERE active AND (database = '50916kek')
Query id: 58185fdd-c9ab-4c1b-83e0-a9d8aa13af71
```

size	rows
3.06 MiB	20000

```
1 rows in set. Elapsed: 0.008 sec.
```

Количество строк соответствует числу загруженных – 2000 и размер равен 3.06 Мб

Загрузим ещё 10000 строк, пропустив первые 10000:

Для этого подойдет команда: `sed -n '10000, 19999p' file.txt`

```
sed -n '10000,19999p' hits_v1.tsv | clickhouse-client --user default --password
qwerty12345 --query "INSERT INTO 50916kek.hits_v1 FORMAT TSV" --
max_insert_block_size=100000
```

Проверим размер еще раз:

```
clickhousestudent :) SELECT formatReadableSize(sum(bytes)) AS size, sum(rows) AS rows FROM system.parts WHERE active and database = '50916kek'
SELECT
    formatReadableSize(sum(bytes)) AS size,
    sum(rows) AS rows
FROM system.parts
WHERE active AND (database = '50916kek')
Query id: 1e18a7eb-c1d9-4810-ab72-f27f2dbfa938
```

size	rows
4.25 MiB	30000

```
1 rows in set. Elapsed: 0.004 sec.
```

В результате добавления размер вырос на $4.25 - 3.06 = 1.19$ Мб

Сохраним эти 10000 строк на диск:

```
sed -n '10000,19999p' hits_v1.tsv > 50916kek.txt
```

Размер файла:

```
administrator@clickhousestudent:~$ wc -c 50916kek.txt
11267110 50916kek.txt
```

Переведем в мегабайты: $11267110 / 1024 / 1024 = 10,74$ Мб

В кликхаусе размер данных примерно в 9 раз меньше. Это связано с тем, что в файлах используется кодировка и на каждый символ выделяется определенное количество бит, в отличие от базы данных, где используются определенные типы данных и значение не может превышать размер этого типа.