

New features of ClickHouse and development roadmap

What we have done
(and we are happy about it)

Done and happy about it

Custom partition key

```
CREATE TABLE ... ENGINE = MergeTree
  ORDER BY CounterID, Date
  PARTITION BY toStartOfWeek(Date)
  SETTINGS index_granularity = 8192
```

Done and happy about it

Geospatial functions:

- pointInPolygon;
- pointInEllipses;
- greatCircleDistance;

```
SELECT pointInPolygon((lat, lon),
  [(6, 0), (8, 4), (5, 8), (0, 2), ...])
```

Done and happy about it

Integration with CatBoost machine learning models

```
SELECT modelEvaluate('purchase', f1, ... fn)
```

Done and happy about it

Working with date and time intervals:

```
SELECT date + INTERVAL 1 MONTH
```

- support for time zones with fractional offset from UTC;
- extended supported time range up to year 2105.
- `timeDiff`, `add/subtractInterval` functions.

Done and happy about it

Proper support for timezones
+ toTimeZone function

Q: What is the result of this query?

```
SELECT toStartOfHour(  
    toDateTime('2017-12-12 01:02:03'),  
    'Asia/Shanghai') AS t
```

Done and happy about it

Distributed DDL queries

```
CREATE TABLE ... ON CLUSTER cluster
```

- * doesn't work with cross-over replication;
- * doesn't work properly with NAT;
- * should be enabled manually for ALTERs of Replicated tables;

Done and happy about it

Support for storing multidimensional arrays in tables.

```
CREATE TABLE t
(
    x Array(Array(String)),
    z Nested(x Array(String), ...)
)
ENGINE = MergeTree ORDER BY x
```

Done and happy about it

Extended array functions:

- arrayConcat, arraySlice;
- arrayPushFront, arrayPushBack, arrayPopFront, arrayPopBack;
- arrayReduce, arraySort, arrayReverseSort, arrayStringConcat;

Done and happy about it

Improvements in dictionaries

— introspection of dictionaries:

```
CREATE DATABASE dictionaries ENGINE = Dictionary
```

— invalidate_query for updating dictionaries;

— SYSTEM RELOAD DICTIONARY;

Done and happy about it

Sessions in HTTP interface:

```
curl 'http://host:port/?session_id=123&session_timeout=60' \  
-d 'CREATE TEMPORARY TABLE t'
```

Native support for SSL in HTTP and TCP servers.

Sending progress info in HTTP headers.

What we have done

(but even more is still to do)

Done but not quite

Querying foreign databases: MySQL, ODBC.

Through table functions mysql, odbc:

```
SELECT ...  
FROM mysql (  
    'host:port',  
    'db', 'table',  
    'user', 'password')
```

Or through CREATE TABLE ... ENGINE = MySQL(...)

Done but not quite

Improvements in SQL compatibility:

- limited support for table aliases and qualified names;
- name quoting in ANSI style;
- type aliases for compatibility;
- INSERT SELECT match columns by positions;
- type conversions in INSERT SELECT;
- NULLS FIRST/LAST for ORDER BY.

Done but not quite

ODBC driver:

- + Tableau basically works;
- but we still have many obstacles;

Done but not quite

Support for NULLs.

Almost done

Almost done

Distributed cluster copier

— for copying between different clusters
or resharding within single cluster.

Almost done

Using index for IN (subquery).

```
SELECT ...  
WHERE key IN  
(  
    SELECT ...  
)
```

Almost done

O_DIRECT for large background merges.

A bit later than right now

A bit later than right now

Support for complex DEFAULT expressions for semistructured formats
JSONEachRow, TSKV.

Support for Protobuf and Parquet formats for I/O.

A bit later than right now

Predicate push down from WHERE into subqueries.

Type conversions in UNION ALL.

Support for subqueries in VIEWS.

A bit later than right now

Support for short-circuit evaluation of expressions.

```
SELECT number != 0  
       ? intDiv(10, number)  
       : 0  
FROM system.numbers LIMIT 10
```

WITH ROLLUP and WITH CUBE modifiers for GROUP BY.

Taking advantage of primary key for ORDER BY and GROUP BY.

A bit later than right now

Allow to use custom compression algorithms
for each column.

Secondary index data structures for data skipping.

Allow to store data in different block devices on single server.

A bit later than right now

Allow to create external dictionaries through DDL query.

Don't return empty resultset in case of aggregating by empty data.

LDAP integration for managing user access rights.

A bit later than right now

Support for UPDATE/DELETE.

Resource pools; resource sharing while executing multiple concurrent queries.

Proper (SQL compatible) JOIN syntax.

"Cloud" tables.

- Web site: <https://clickhouse.yandex/>

Google groups: <https://groups.google.com/forum/#!forum/clickhouse>

Maillist: clickhouse-feedback@yandex-team.com

Telegram chat (hope it isn't banned in China):
https://telegram.me/clickhouse_en and https://telegram.me/clickhouse_ru
 (now with over 1300 members)

GitHub: <https://github.com/yandex/ClickHouse/>

Twitter: <https://twitter.com/ClickHouseDB>

China User Group: <http://clickhouse.com.cn/>

+ meetups. Moscow, Saint-Petersburg, Novosibirsk, Ekaterinburg, Minsk,
Nijmegen, Nuremberg, Delft, Bonn, Berlin, Amsterdam