



Moving Big Data from Snowflake to ClickHouse for Fun and Profit

Robert Hodges - Altinity Engineering
23 June 2022

Let's make some introductions

Robert Hodges

Database geek with 30+ years
on DBMS systems. Day job:
Altinity CEO

Altinity Engineering

Database geeks with centuries
of experience in DBMS and
applications



ClickHouse support and services including [Altinity.Cloud](#)
Authors of [Altinity Kubernetes Operator for ClickHouse](#)
and other open source projects

Foundations

Snowflake is a SQL Data Warehouse

Understands SQL (completely)

Runs in public cloud

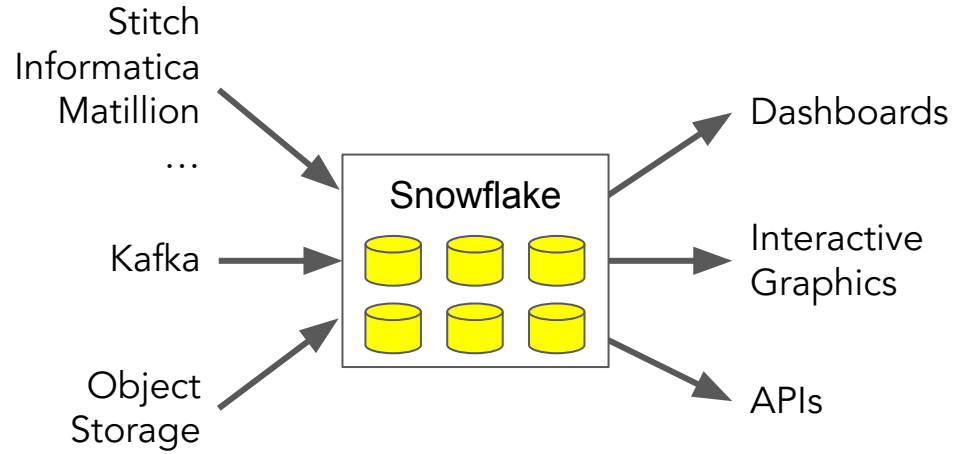
Decoupled compute/storage

Advanced cloud management

Star schemas with large table joins & complex query

Automatic performance tuning

Is proprietary



A popular engine for
general purpose
analytics

ClickHouse is a SQL Data Warehouse, too

Understands SQL

Runs anywhere Linux does

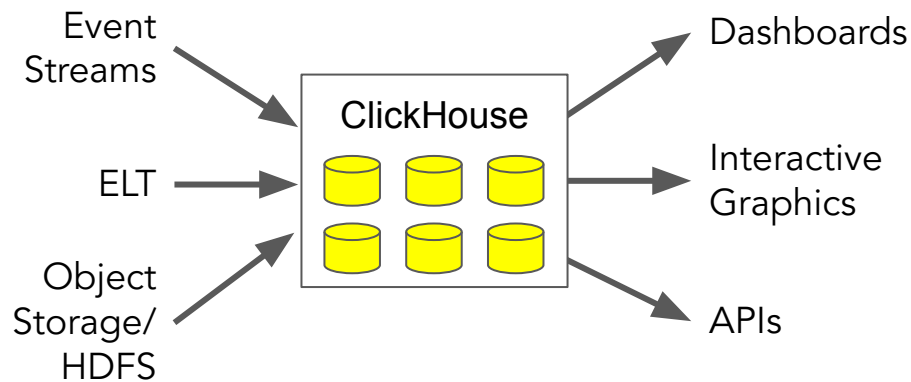
Shared nothing architecture

Extremely cost efficient

Ridiculously fast data ingest

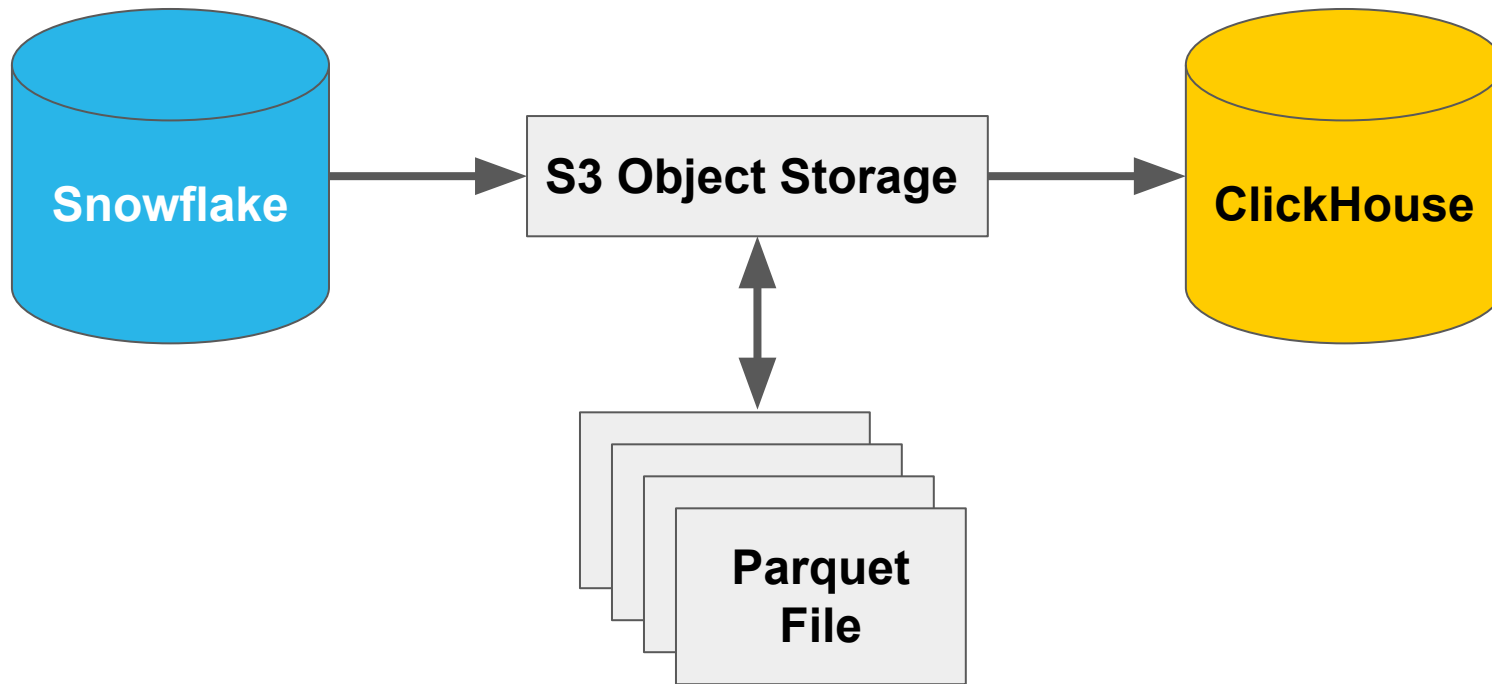
Linear scaling with fixed response

Open source (Apache 2.0)



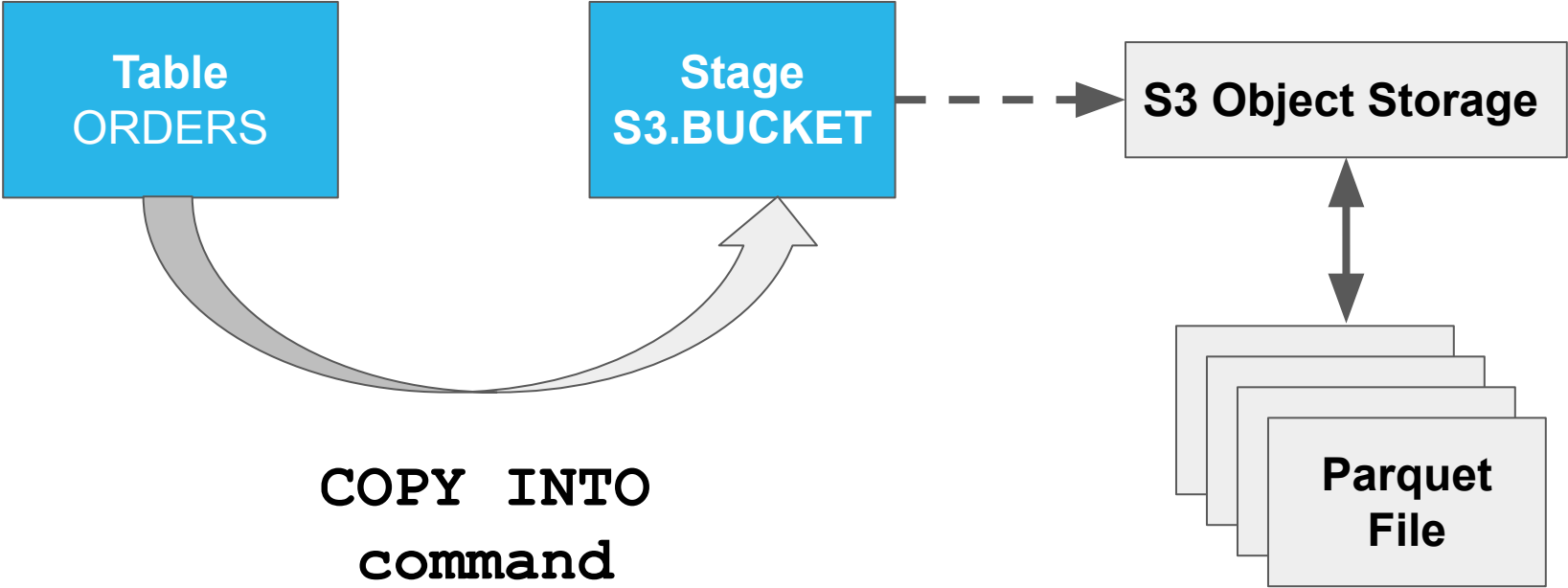
It's a popular engine for
real-time analytics

Our goal: moving data from Snowflake to ClickHouse



Extracting data from Snowflake to Parquet

Exporting from Snowflake into Parquet files on S3



Source table on Snowflake

name	type	kind	null?
O_ORDERKEY	NUMBER(38,0)	COLUMN	N
O_CUSTKEY	NUMBER(38,0)	COLUMN	N
O_ORDERSTATUS	VARCHAR(1)	COLUMN	N
O_TOTALPRICE	NUMBER(12,2)	COLUMN	N
O_ORDERDATE	DATE	COLUMN	N
O_ORDERPRIORITY	VARCHAR(15)	COLUMN	N
O_CLERK	VARCHAR(15)	COLUMN	N
O_SHIPPRIORITY	NUMBER(38,0)	COLUMN	N
O_COMMENT	VARCHAR(79)	COLUMN	N

Create a stage on SnowFlake

```
-- Create a schema to put objects in.  
CREATE SCHEMA S3  
  
-- Create a stage connected to an S3 bucket.  
CREATE OR REPLACE STAGE S3.BUCKET  
URL='s3://your-migration-bucket/snowflake'  
CREDENTIALS=  
    (aws_key_id='aws_access_key',  
     aws_secret_key='aws_secret_key')  
FILE_FORMAT = (TYPE = PARQUET);
```



**Write output using
Parquet format**

Copy from Snowflake table to stage

```
COPY INTO  
@S3.BUCKET/SNOWFLAKE_SAMPLE_DATA/TPCH_SF100/ORDERS/  
FROM SNOWFLAKE_SAMPLE_DATA.TPCH_SF100.ORDERS  
HEADER=TRUE
```



**Store column names
in Parquet**

Check your work...

```
SELECT COUNT(*)  
FROM SNOWFLAKE_SAMPLE_DATA.TPCH_SF100.ORDERS
```

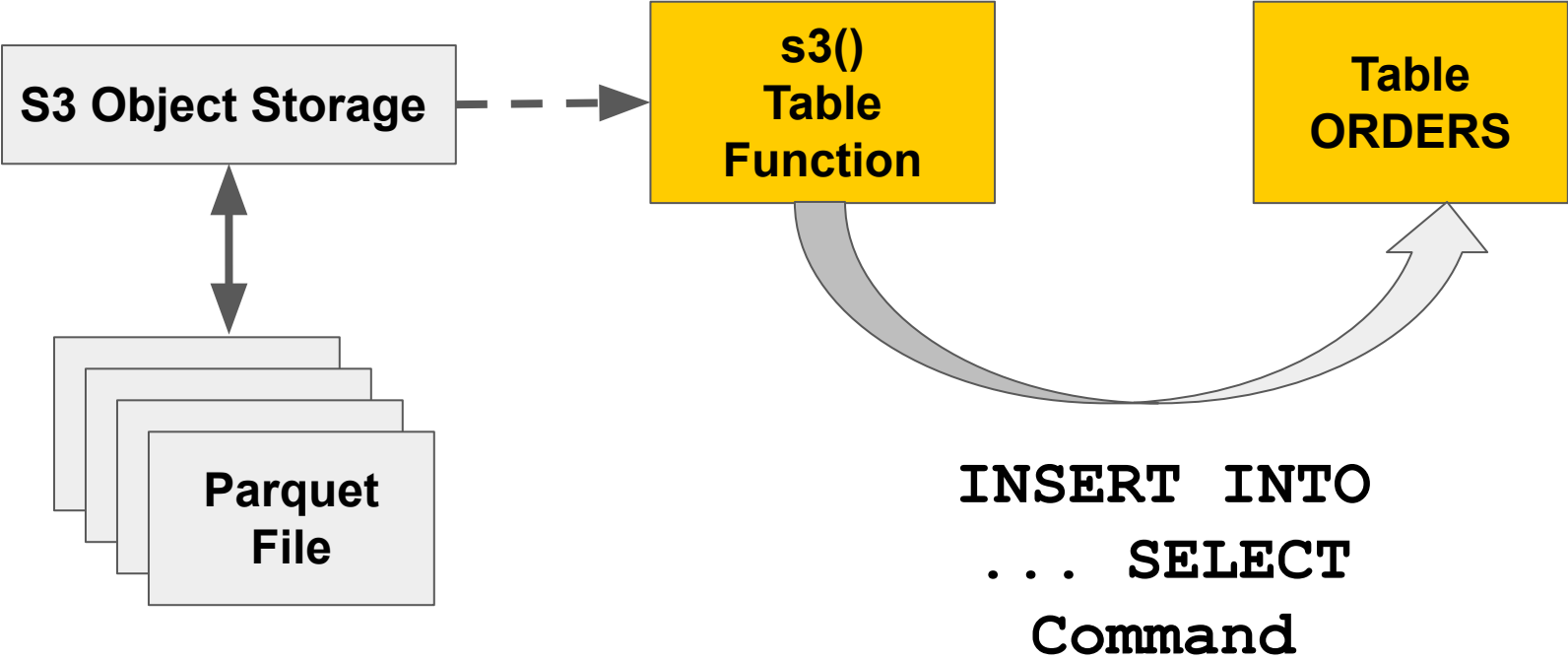
```
SELECT COUNT(*)  
FROM @S3.BUCKET/SNOWFLAKE_SAMPLE_DATA/TPCH_SF100/ORDERS/
```



Not very fast...

Loading Parquet into ClickHouse

Exporting from Snowflake into Parquet files on S3



Create a table on ClickHouse

```
CREATE TABLE IF NOT EXISTS ORDERS (  
    O_ORDERKEY Int128,  
    O_CUSTKEY Int128,  
    O_ORDERSTATUS String,  
    O_TOTALPRICE Decimal(12, 2),  
    O_ORDERDATE Date,  
    O_ORDERPRIORITY String,  
    O_CLERK String,  
    O_SHIPPRIORITY Int128,  
    O_COMMENT String  
)  
Engine=MergeTree()  
PARTITION BY tuple() ORDER BY tuple()
```

Inefficient schema!

Default compression!

**No partitioning or
sorting!**



Load data from open S3 bucket when there are no nulls...

```
-- Set to number of vCPUs.
```

```
SET max_insert_threads=16
```

```
INSERT INTO ORDERS
```

```
SELECT * FROM
```

```
s3('https://s3.us-east-1.amazonaws.com/your-migration-bucket/s  
nowflake/SNOWFLAKE_SAMPLE_DATA/TPCH_SF100/ORDERS/*.parquet',  
'aws_access_key', 'aws_secret_key', Parquet)
```

**AWS
credentials
included**

Load from S3 bucket when data have nulls

```
-- Set to number of vCPUs.  
SET max_insert_threads=16
```

```
INSERT INTO ORDERS  
SELECT * FROM  
s3('https://s3.us-east-1.amazonaws.com/your-migration-bucket/s  
nowflake/SNOWFLAKE_SAMPLE_DATA/TPCH_SF100/ORDERS/*.parquet',  
'aws_access_key', 'aws_secret_key', Parquet,  
'O_ORDERKEY Int128, O_CUSTKEY Int128, O_ORDERSTATUS String,  
O_TOTALPRICE Decimal(12, 2), O_ORDERDATE Date, O_ORDERPRIORITY  
String, O_CLERK String, O_SHIPPRIORITY Int128, O_COMMENT  
String')
```

**AWS
credentials
included**

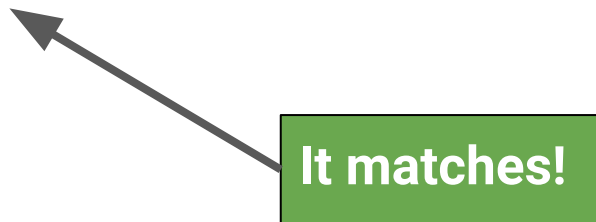
Schema required due to missing cast*

* <https://github.com/ClickHouse/ClickHouse/issues/35346>

Check your work...

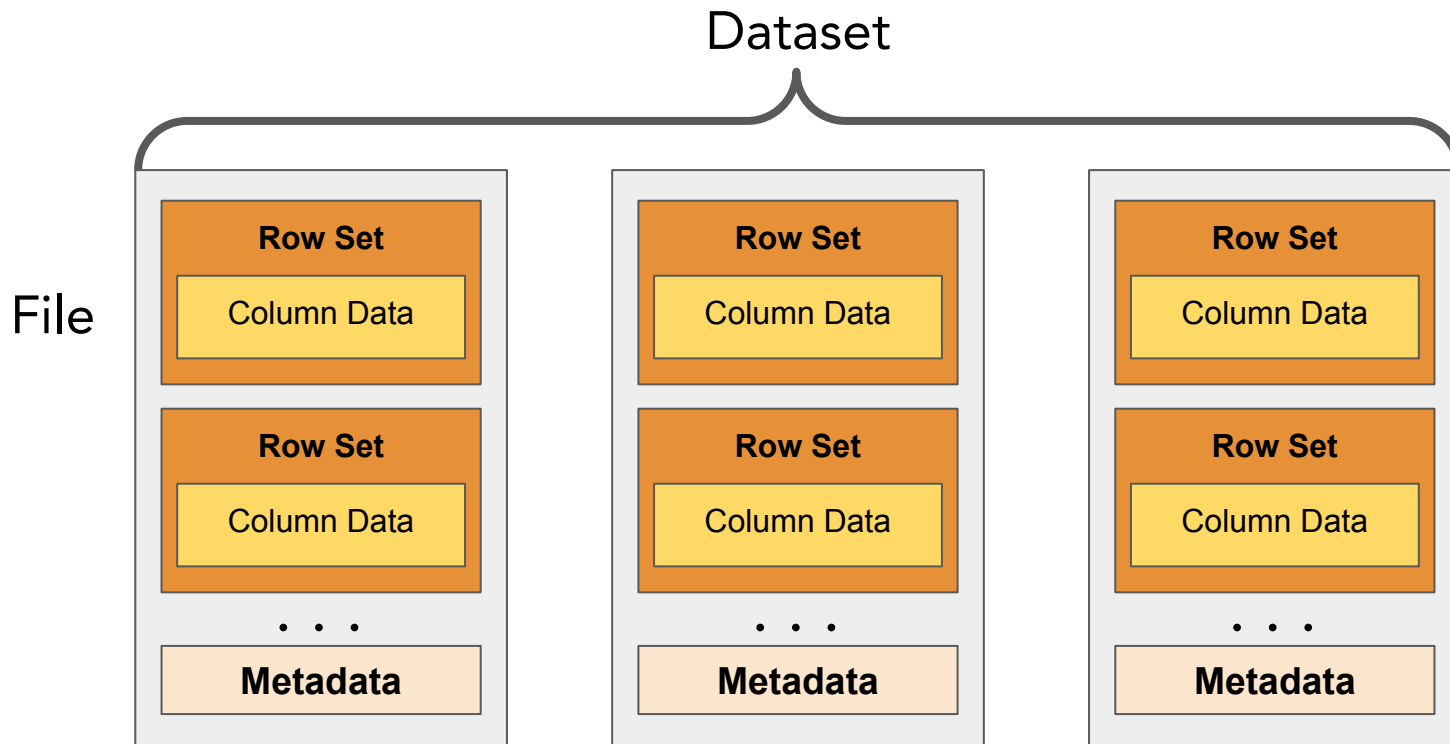
```
SELECT count() FROM ORDERS
```

count()
150000000



More about Parquet and ClickHouse

Apache Parquet file layout



Grab script to read Parquet and generate ClickHouse SQL

```
git clone \  
  https://github.com/Altinity/clickhouse-sql-examples.git  
  
cd clickhouse-sql-examples/parquet  
python3 -m venv .venv  
. .venv/bin/activate  
python3 -m pip install --upgrade pip  
pip install -r requirements.txt
```

Set environmental variables

```
# Region and keys for S3 access.  
export AWS_S3_REGION="us-east-1"  
export AWS_ACCESS_KEY_ID="aws_access_key"  
export AWS_SECRET_ACCESS_KEY="aws_secret_key"  
  
# Parquet dataset path: bucket_name/dir1/.../dirN/table_name/  
export  
S3_DATASET_PATH="your-migration-bucket/snowflake/SNOWFLAKE_SAMPLE_DATA/TPCH_SF100/ORDERS/"
```

Run script to generate CREATE TABLE and INSERT

```
$ python generate-ch-schema.py
-- Automatically generated DDL and INSERT for Parquet data
-- AWS REGION: us-east-1
-- S3 DATASET PATH:
your-migration-bucket/snowflake/SNOWFLAKE_SAMPLE_DATA/TPCH_SF1
00/ORDERS/
-- Table name: ORDERS
CREATE TABLE IF NOT EXISTS ORDERS (
  O_ORDERKEY Int128,
  O_CUSTKEY Int128,
  O_ORDERSTATUS String,
  O_TOTALPRICE Decimal(12, 2),
  ...
```

Optimizing imported table in ClickHouse

Create a better table in ClickHouse

```
CREATE TABLE IF NOT EXISTS ORDERS_OPTIMIZED (  
  O_ORDERKEY UInt64,  
  O_CUSTKEY UInt64,  
  O_ORDERSTATUS FixedString(1),  
  O_TOTALPRICE Decimal(12, 2) CODEC(ZSTD(10)),  
  O_ORDERDATE Date,  
  O_ORDERPRIORITY String,  
  O_CLERK String CODEC(ZSTD(10)),  
  O_SHIPPRIORITY UInt8,  
  O_COMMENT String CODEC(ZSTD(10))  
)  
Engine=MergeTree()  
PARTITION BY toYYYYMM(O_ORDERDATE)  
ORDER BY (O_ORDERDATE, O_CUSTKEY)
```

Smaller data types

ZSTD compression

Partition by time

Efficient ordering
and primary key

Move data into the optimized table

```
-- Set to number of vCPUs.
```

```
SET max_insert_threads=16
```

```
INSERT INTO ORDERS_OPTIMIZED SELECT * FROM ORDERS
```

```
OPTIMIZE TABLE ORDERS_OPTIMIZED FINAL
```

Check the table size before and after optimization

```
SELECT
  name, total_rows AS rows,
  formatReadableSize(total_bytes) AS size
FROM system.tables
WHERE name LIKE 'ORDERS%' ORDER BY name
```

name	rows	size
ORDERS	150000000	6.06 GiB
ORDERS_OPTIMIZED	150000000	4.37 GiB

Completely unoptimized

Within 2% of Snowflake

Conclusion and where to learn more

Moving a table from Snowflake to ClickHouse

1. Set up S3 bucket.
2. Snowflake: COPY data from Snowflake table to S3 via stage.
3. ClickHouse:
 - a. Create working table definition.
 - b. Load data from S3 using s3() function.
 - c. Optimize the schema and reload.

Where is the documentation?

Altinity Blog – [Migrating Data from Snowflaw to ClickHouse using S3 and Parquet](#)

ClickHouse SQL Examples – <https://github.com/Altinity/clickhouse-sql-examples>

Apache Parquet Documentation – <https://parquet.apache.org/docs/>

Apache Arrow Documentation – <https://arrow.apache.org/docs/>



Thank you!
Questions?

<https://altinity.com>

rhodges at altinity.com

Altinity.Cloud

Software and
Support for
ClickHouse

We're hiring!