DSS | yugabyteDB

Migrating MySQL to Yugabyte using ysql_loader

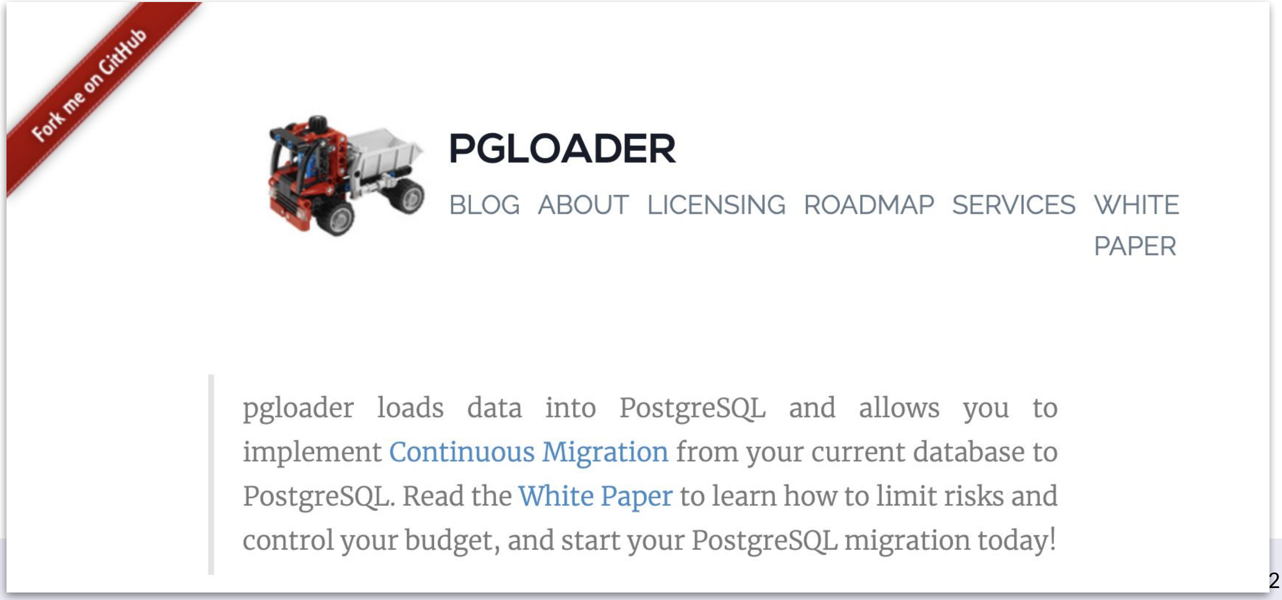
Mike Lee - Solutions Architect
Suranjan Kumar - Ecosystem Integration Engineer
Yugabyte

Agenda

- What is ysql_loader?
- Where can I Git it?
- How to use it?
    - configuration options
- How to run it!
- Yugabyte features added
- Demo

What is ysql_loader

- ysql_loader = pgloader (pgloader.io)
- Dimitri Fontaine: wrote and maintains of pgloader, Major Contributor to PostgreSQL, author of The Art of PostgreSQL
  - Open Source
  - ysql_loader forked from pgloader (Suranjan)



pgloader loads data into PostgreSQL and allows you to implement Continuous Migration from your current database to PostgreSQL. Read the White Paper to learn how to limit risks and control your budget, and start your PostgreSQL migration today!

ysql_loader

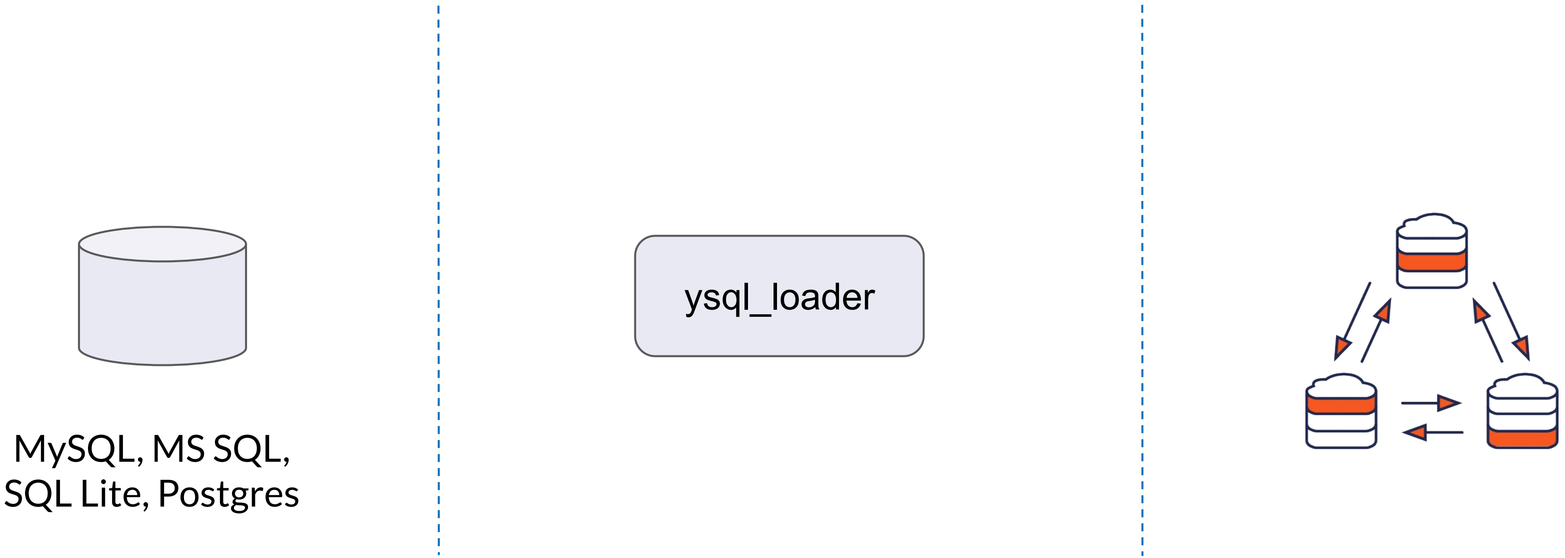
- Migrates database objects (tables, indexes, sequences) and loads data from files (CSV, Fixed, DBF, IXF) or directly connecting to source databases such as MySQL, SQL Lite, MS SQL and PostgreSQL

- Can load from flat files (CSV, Fixed Format) or directly connect and migrate entire databases to

- Follows a typical database migration workflow:
  - Create the target database on Yugabyte
  - Gather database objects from source database metadata catalog
  - Create Tables and Indexes
  - Copy table data using PostgreSQL COPY
  - Add the constraints, primary & foreign keys, and comments

- Has a rich set of options that allow you to customize ysql_loader to your exact needs.

Where to install ysql_loader?



MySQL, MS SQL,
SQL Lite, Postgres

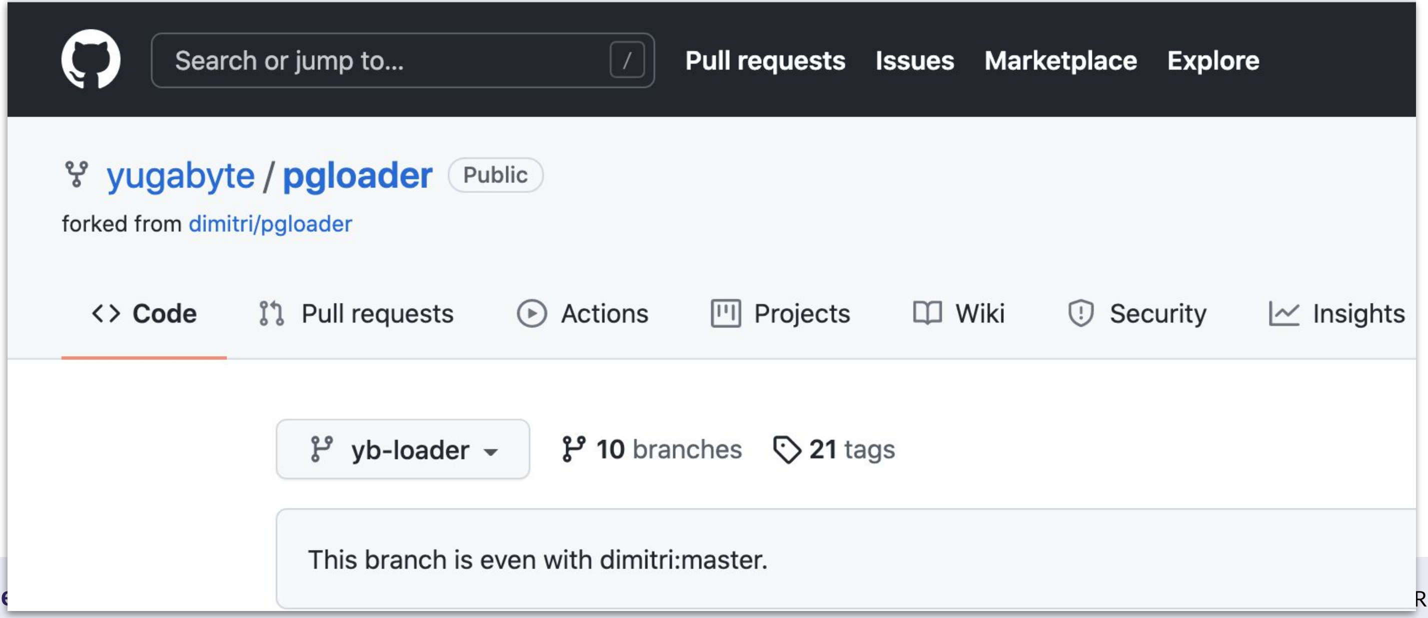
ysql_loader

For production migrations, ysql_loader should be installed on a separate server.
For this workshop, ysql_loader can be installed on the same server as MySQL

Pgloader - how to Git it...

$git clone https://github.com/yugabyte/pgloader

NOTE: install pgloader on a separate “migration server”



yugabyte / pgloader Public

forked from dimitri/pgloader

This branch is even with dimitri:master.

pgloader - build it and they will migrate….

```
$ git clone https://github.com/yugabyte/pgloader
#
# GET ALL LIBRARIES
$ apt-get install sbcl unzip libsqlite3-dev make curl gawk freetds-dev libzip-dev
$ cd /path/to/pgloader
$ make pgloader

#
# purposely renamed, but you could replace pgloader
$ sudo cp /build/bin/pgloader /usr/bin/ysql_loader
```

pgloader(Docker) - build it and they will migrate

```
$ git clone https://github.com/yugabyte/pgloader

$ cd /path/to/pgloader

$ sudo docker -t ysql-loader:v1.3 build .
```

pgloader command-file options

Configuration Sections

- **Database Source: FROM**
- **Migration Options: WITH**
- **Casting Rules: CAST**
- **Partial Migration: INCLUDING/EXCLUDING NAMES**
- Encoding Support
- Schema Transformations
- View Support

Check out: https://pgloader.readthedocs.io/en/latest/index.html

Pgloader command options - FROM

```
LOAD DATABASE
    FROM mysql://root:<password>@IP.Addr:3306/ml_migratedb
    INTO postgresql://yugabyte:<password>@IP.Addr:5433/ml_migratedb;
```

Determines the source and targets

Pre-migration checks: test the login and password and IP to make sure you can access both the source database and YugabyteDB from the wherever ysql_loader was installed.

Pgloader command options - WITH

```
LOAD DATABASE
     FROM mysql://root:<password>@IP.Ad.dre.ss:3306/ml_migratedb
     INTO postgresql://yugabyte:<password>@IP.Ad.dres.ss:5433/ml_migratedb
WITH batch rows = 1000, truncate
;
```

Default WITH options with MySQL:

- no truncate
- create tables
- include drop
- create indexes
- reset sequences
- foreign keys
- **downcase identifiers** - be mindful of tables with same name but differ by capitalization.
- **uniquify index names** - PG index names have to be unique per-schema (MySQL is per-table)

Pgloader command options - INCLUDING/EXCLUDING

```
LOAD DATABASE
     FROM mysql://root:<password>@IP.Ad.dres.ss:3306/ml_migratedb
     INTO postgresql://yugabyte:<password>@IP.Ad.dres.ss:5433/ml_migratedb
WITH batch rows = 1000, truncate
INCLUDING ONLY TABLE NAMES  MATCHING ~/ml_/, ‘orders_orig’
EXCLUDING TABLE NAMES MATCHING ~<orig>
;
```

INCLUDING: comma separated list of table names or regular expression used to limit the tables.

EXCLUDING: comma separated list of table names or regular expressions used to limit the tables, HOWEVER, This filter only affects the result of the INCLUDING filter.

Pgloader command options - CAST

```
LOAD DATABASE
     FROM mysql://root:<password>@IP.Ad.dres.ss:3306/ml_migratedb
     INTO postgresql://yugabyte:<password>@IP.Ad.dres.ss:5433/ml_migratedb
WITH batch rows = 1000, truncate
INCLUDING ONLY TABLE NAMES  MATCHING ~/ml_/, ‘orders_orig’
EXCLUDING TABLE NAMES MATCHING ~<orig>
CAST
  type tinyint to smallint drop typemod
;
```

MySQL TINYINT(1) supports values 0,1,2,3

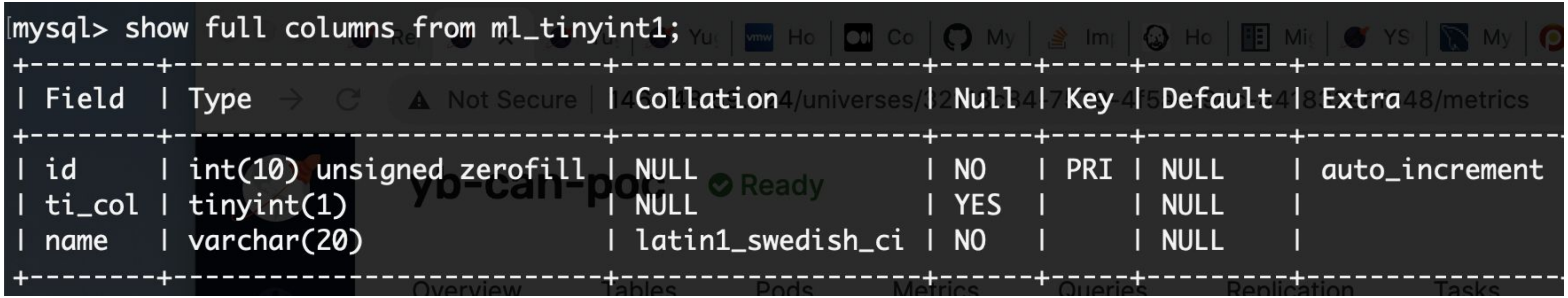
PostgreSQL has BOOLEAN, but it only supports

- TRUE, 't', 'true', 'y', 'yes', 'on', '1' or FALSE, 'f', 'false', 'n', 'no', 'off', '0'
- To retain values 2 and 3, you must convert the data type to a SMALLINT.

CAST does this automatically for you!

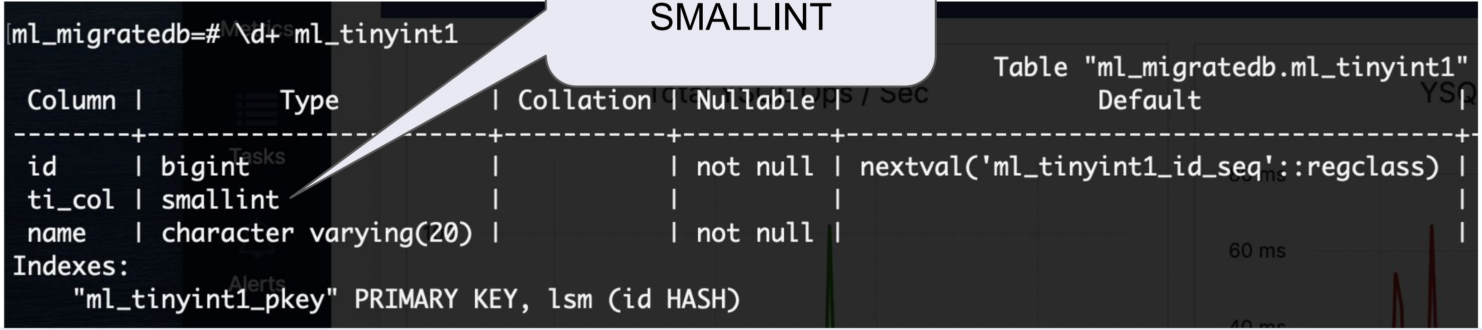
ysql_loader command options - CAST

MYSQL



```
mysql> show full columns from ml_tinyint1;
+--------+--------------------------+-------------------+------+-----+---------+----------------
| Field  | Type                     | Collation         | Null | Key | Default | Extra
+--------+--------------------------+-------------------+------+-----+---------+----------------
| id     | int(10) unsigned zerofill | NULL             | NO   | PRI | NULL    | auto_increment
| ti_col | tinyint(1)               | NULL              | YES  |     | NULL    |
| name   | varchar(20)              | latin1_swedish_ci | NO   |     | NULL    |
+--------+--------------------------+-------------------+------+-----+---------+----------------
```

YugabyteDB



```
ml_migratedb=# \d+ ml_tinyint1
                                                              Table "ml_migratedb.ml_tinyint1"
 Column |         Type          | Collation | Nullable |                 Default                 |
--------+-----------------------+-----------+----------+-----------------------------------------+
 id     | bigint                |           | not null | nextval('ml_tinyint1_id_seq'::regclass) |
 ti_col | smallint              |           |          |                                         |
 name   | character varying(20) |           | not null |                                         |
Indexes:
    "ml_tinyint1_pkey" PRIMARY KEY, lsm (id HASH)
```

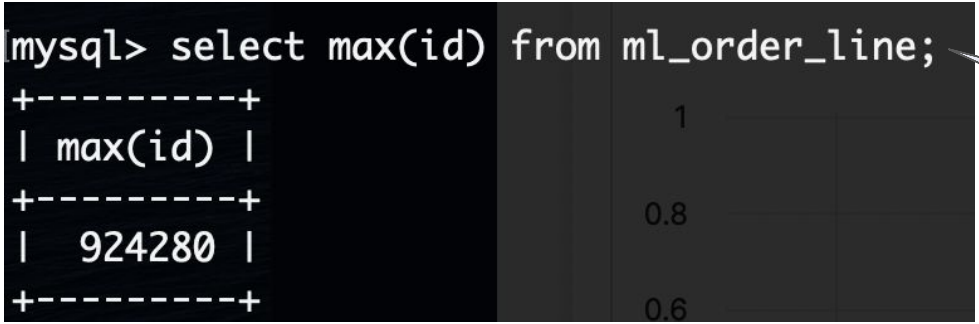
Converted to SMALLINT

Data type considerations

Data Type	MySQL	PostgreSQL
Integers(bytes)	TINYINT(1), SMALLINT(2), MEDIUMINT(3), INT(4), BIGINT(8)	SMALLINT(2), INTEGER(4), BIGINT(8)
Text	TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT	TEXT
Number	DECIMAL, NUMERIC	DECIMAL, NUMERIC
Double	DOUBLE	DOUBLE

ysql_loader command options - RESET SEQUENCES

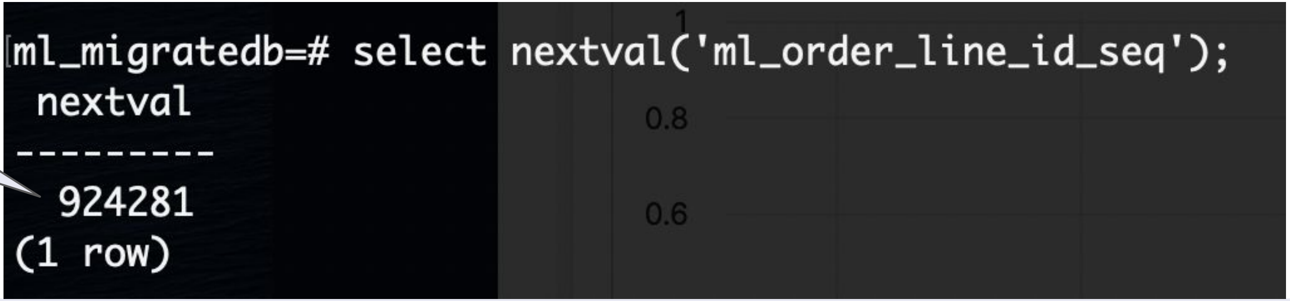
MySQL



```
[mysql> select max(id) from ml_order_line;
+----------+
| max(id) |
+----------+
|  924280 |
+----------+
```

MySQL: ml_order_line table has a sequence called **ml_order_line_id_seq**

Yugabyte



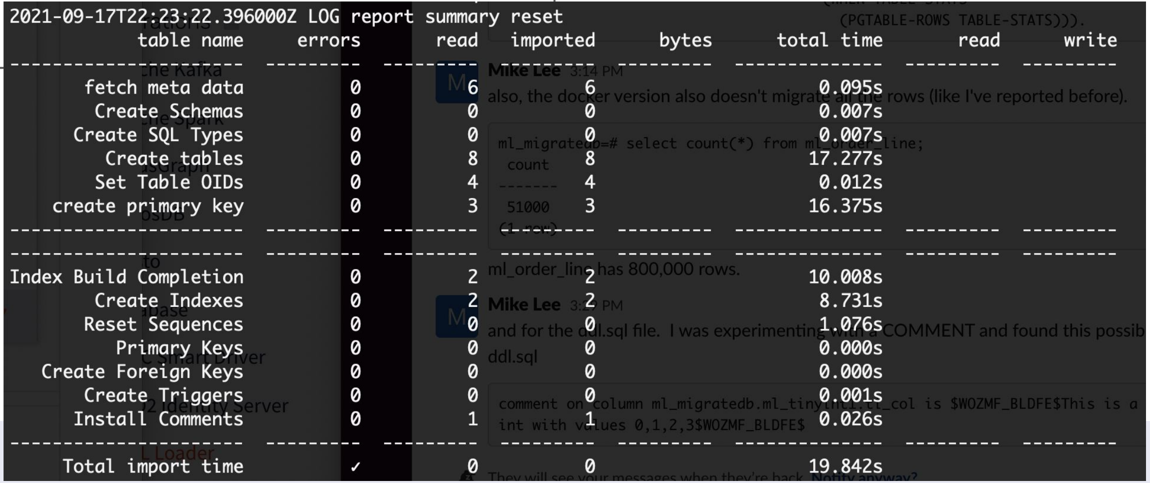
```
[ml_migratedb=# select nextval('ml_order_line_id_seq');
 nextval
---------
  924281
(1 row)
```

In ybdb: the max value is preserved and next value ready to use.

ysql_loader command options - DUMPDDL ONLY

```
LOAD DATABASE
    FROM mysql://root:P8ssw0rd2@10.142.0.2:3306/ml_migratedb
    INTO postgresql://yugabyte:yugabyte@10.204.0.5:5433/ml_migratedb
WITH dumpddl only
including only table names matching ~/ml_/, 'orders_orig'
CAST
  type tinyint to smallint drop typemod
;
```

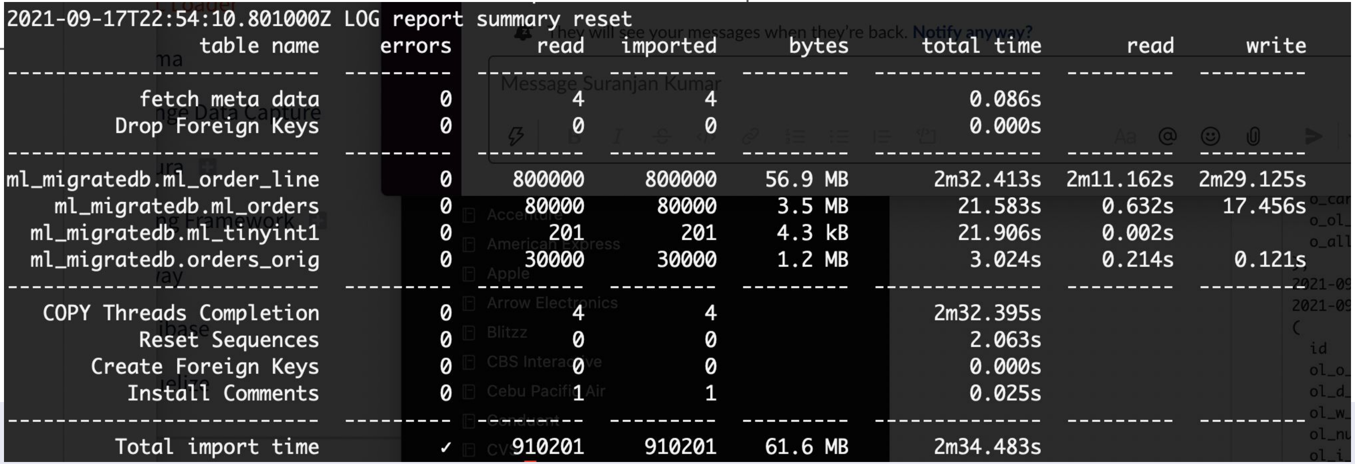
DDL extracted to file ddl.sql !



```
2021-09-17T22:23:22.396000Z LOG report summary reset
             table name     errors       read   imported      bytes      total time       read      write
-----------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
        fetch meta data          0          6          6                     0.095s
         Create Schemas          0          0          0                     0.007s
       Create SQL Types          0          0          0                     0.007s
          Create tables          0          8          8                    17.277s
         Set Table OIDs          0          4          4                     0.012s
     create primary key          0          3          3                    16.375s
-----------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
-----------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
Index Build Completion           0          2          2                    10.008s
         Create Indexes          0          2          2                     8.731s
        Reset Sequences          0          0          0                     1.076s
           Primary Keys          0          0          0                     0.000s
     Create Foreign Keys          0          0          0                     0.000s
        Create Triggers          0          0          0                     0.001s
       Install Comments          0          1          1                     0.026s
-----------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
      Total import time          ✓          0          0                    19.842s
```

ysql_loader command options - DATA ONLY

```
LOAD DATABASE
    FROM mysql://root:P8ssw0rd2@10.142.0.2:3306/ml_migratedb
    INTO postgresql://yugabyte:yugabyte@10.204.0.5:5433/ml_migratedb
WITH data only
including only table names matching ~/ml_/, 'orders_orig'
CAST
  type tinyint to smallint drop typemod
;
```



```
2021-09-17T22:54:10.801000Z LOG report summary reset
                table name     errors       read   imported      bytes      total time       read      write
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
           fetch meta data          0          4          4                     0.086s
         Drop Foreign Keys          0          0          0                     0.000s
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
ml_migratedb.ml_order_line          0     800000     800000    56.9 MB       2m32.413s  2m11.162s  2m29.125s
    ml_migratedb.ml_orders          0      80000      80000     3.5 MB         21.583s     0.632s    17.456s
  ml_migratedb.ml_tinyint1          0        201        201     4.3 kB         21.906s     0.002s
  ml_migratedb.orders_orig          0      30000      30000     1.2 MB          3.024s     0.214s     0.121s
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
   COPY Threads Completion          0          4          4                  2m32.395s
           Reset Sequences          0          0          0                     2.063s
       Create Foreign Keys          0          0          0                     0.000s
          Install Comments          0          1          1                     0.025s
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
         Total import time          ✓     910201     910201    61.6 MB       2m34.483s
```

Running pgloader

pgloader can be run by passing arguments or by passing a command-file

```
ybpgloader --help
ybpgloader [ option ... ] command-file ...
ybpgloader [ option ... ] SOURCE TARGET
 --help -h                   boolean  Show usage and exit.
 --version -V                boolean  Displays pgloader version and exit.
 --quiet -q                  boolean  Be quiet
 --verbose -v                boolean  Be verbose
 --debug -d                  boolean  Display debug level information.
 --client-min-messages       string   Filter logs seen at the console (default:
"warning")
 --log-min-messages          string   Filter logs seen in the logfile (default:
"notice")
 --summary -S                string   Filename where to copy the summary
 --root-dir -D               string   Output root directory. (default:
#P"/tmp/pgloader/")
 --upgrade-config -U         boolean  Output the command(s) corresponding
to .conf file for v2.x
 --list-encodings -E         boolean  List pgloader known encodings and exit.
 --logfile -L                string   Filename where to send the logs.
 --load-lisp-file -l         string   Read user code from files
```

```
 --dry-run                   boolean  Only check database connections, don't
load anything.
 --on-error-stop             boolean  Refrain from handling errors
properly.
 --no-ssl-cert-verification  boolean  Instruct OpenSSL to bypass
verifying certificates.
 --context -C                string   Command Context Variables
 --with                      string   Load options
 --set                       string   PostgreSQL options
 --field                     string   Source file fields specification
 --cast                      string   Specific cast rules
 --type                      string   Force input source type
 --encoding                  string   Source expected encoding
 --before                    string   SQL script to run before loading the data
 --after                     string   SQL script to run after loading the data
 --self-upgrade              string   Path to pgloader newer sources
 --regress                   boolean  Drive regression testing
```

Running pgloader

Using the command file:

```
sudo ysql_loader --verbose <cmd_file>
```

Or by running it from Docker...

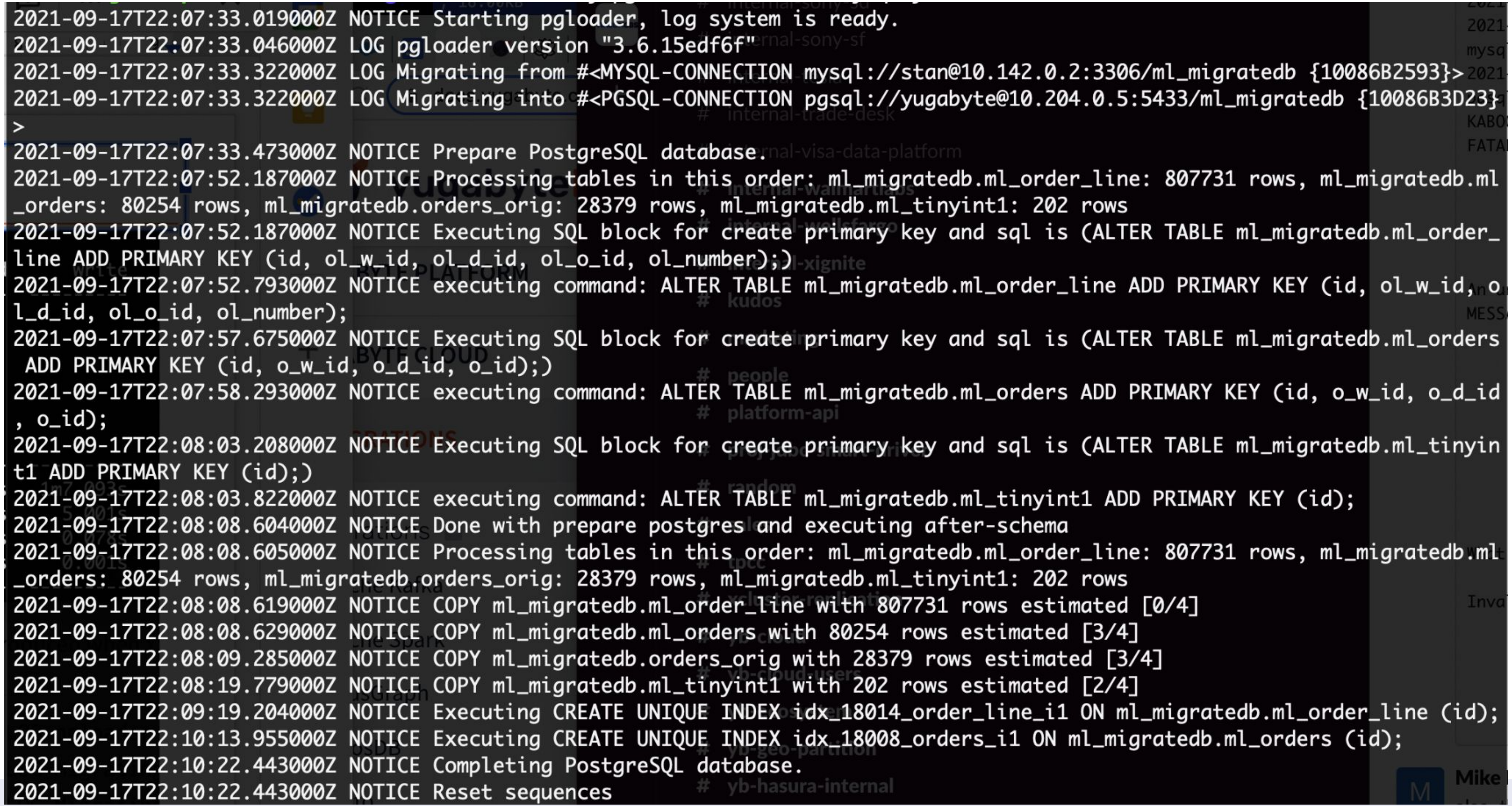
```
docker run --rm --name pgloader  \
      -v <local_dir pgloader_config_dir>:<mount_path_in_container> \
      yugabytedb/pgloader:v1.1 pgloader --verbose  \
      <mount_path_in_container>/<pgloader_config_file_in_local_dir>

example:

sudo docker run --rm --name pgloader \
-v /home/ubuntu/Mlee/PgloaderDSS:/tmp \
/bin/bash -c 'cd /tmp; pgloader --verbose \
 /tmp/schmonly.conf'
```

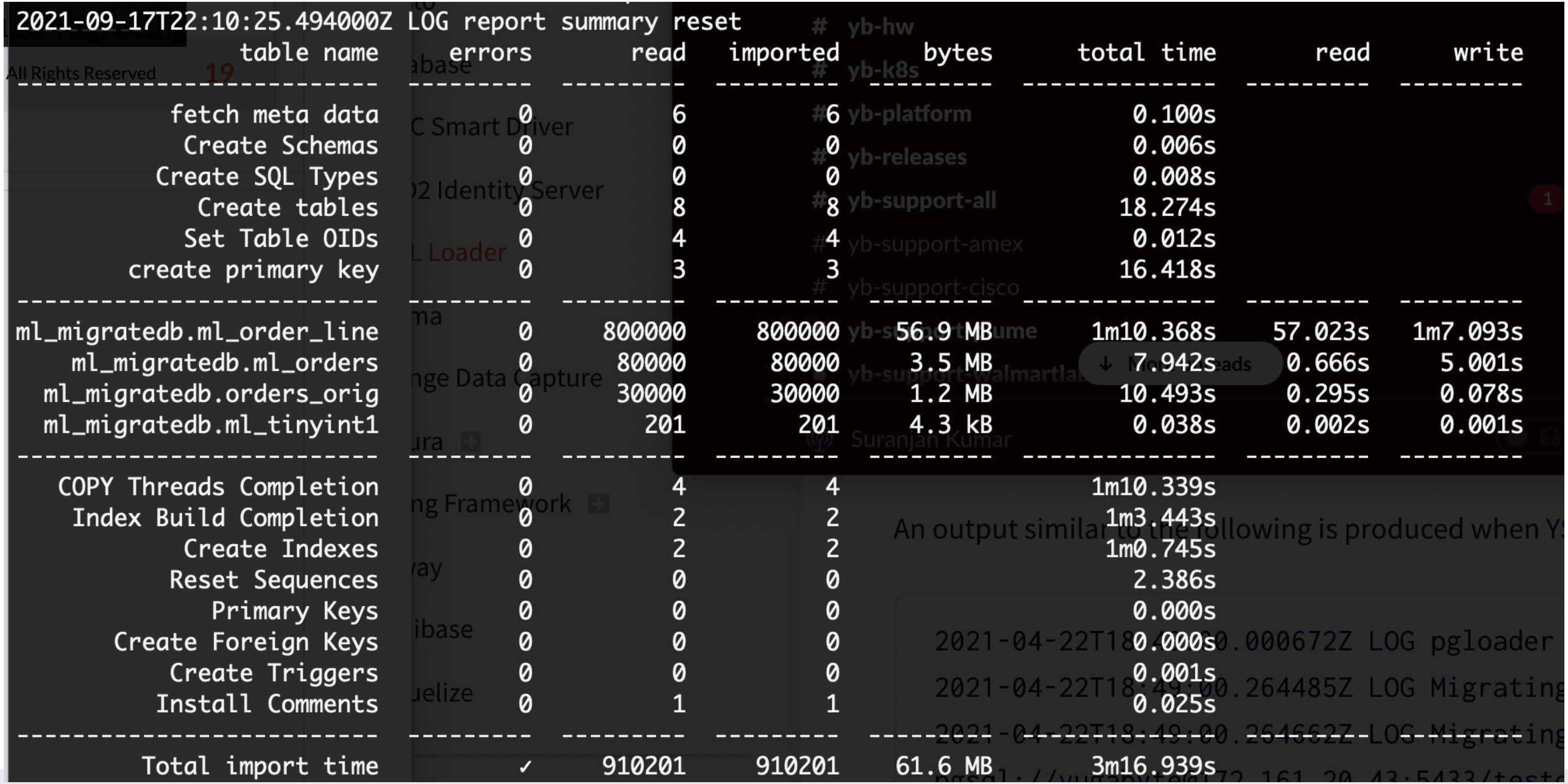
Let’s migrate some data!

Migration output using --verbose option



```
2021-09-17T22:07:33.019000Z NOTICE Starting pgloader, log system is ready.
2021-09-17T22:07:33.046000Z LOG pgloader version "3.6.15edf6f"
2021-09-17T22:07:33.322000Z LOG Migrating from #<MYSQL-CONNECTION mysql://stan@10.142.0.2:3306/ml_migratedb {10086B2593}>
2021-09-17T22:07:33.322000Z LOG Migrating into #<PGSQL-CONNECTION pgsql://yugabyte@10.204.0.5:5433/ml_migratedb {10086B3D23}
>
2021-09-17T22:07:33.473000Z NOTICE Prepare PostgreSQL database.
2021-09-17T22:07:52.187000Z NOTICE Processing tables in this order: ml_migratedb.ml_order_line: 807731 rows, ml_migratedb.ml
_orders: 80254 rows, ml_migratedb.orders_orig: 28379 rows, ml_migratedb.ml_tinyint1: 202 rows
2021-09-17T22:07:52.187000Z NOTICE Executing SQL block for create primary key and sql is (ALTER TABLE ml_migratedb.ml_order_
line ADD PRIMARY KEY (id, ol_w_id, ol_d_id, ol_o_id, ol_number);)
2021-09-17T22:07:52.793000Z NOTICE executing command: ALTER TABLE ml_migratedb.ml_order_line ADD PRIMARY KEY (id, ol_w_id, o
l_d_id, ol_o_id, ol_number);
2021-09-17T22:07:57.675000Z NOTICE Executing SQL block for create primary key and sql is (ALTER TABLE ml_migratedb.ml_orders
 ADD PRIMARY KEY (id, o_w_id, o_d_id, o_id);)
2021-09-17T22:07:58.293000Z NOTICE executing command: ALTER TABLE ml_migratedb.ml_orders ADD PRIMARY KEY (id, o_w_id, o_d_id
, o_id);
2021-09-17T22:08:03.208000Z NOTICE Executing SQL block for create primary key and sql is (ALTER TABLE ml_migratedb.ml_tinyin
t1 ADD PRIMARY KEY (id);)
2021-09-17T22:08:03.822000Z NOTICE executing command: ALTER TABLE ml_migratedb.ml_tinyint1 ADD PRIMARY KEY (id);
2021-09-17T22:08:08.604000Z NOTICE Done with prepare postgres and executing after-schema
2021-09-17T22:08:08.605000Z NOTICE Processing tables in this order: ml_migratedb.ml_order_line: 807731 rows, ml_migratedb.ml
_orders: 80254 rows, ml_migratedb.orders_orig: 28379 rows, ml_migratedb.ml_tinyint1: 202 rows
2021-09-17T22:08:08.619000Z NOTICE COPY ml_migratedb.ml_order_line with 807731 rows estimated [0/4]
2021-09-17T22:08:08.629000Z NOTICE COPY ml_migratedb.ml_orders with 80254 rows estimated [3/4]
2021-09-17T22:08:09.285000Z NOTICE COPY ml_migratedb.orders_orig with 28379 rows estimated [3/4]
2021-09-17T22:08:19.779000Z NOTICE COPY ml_migratedb.ml_tinyint1 with 202 rows estimated [2/4]
2021-09-17T22:09:19.204000Z NOTICE Executing CREATE UNIQUE INDEX idx_18014_order_line_i1 ON ml_migratedb.ml_order_line (id);
2021-09-17T22:10:13.955000Z NOTICE Executing CREATE UNIQUE INDEX idx_18008_orders_i1 ON ml_migratedb.ml_orders (id);
2021-09-17T22:10:22.443000Z NOTICE Completing PostgreSQL database.
2021-09-17T22:10:22.443000Z NOTICE Reset sequences
```

The migrated database summary



```
2021-09-17T22:10:25.494000Z LOG report summary reset
                table name     errors       read   imported      bytes      total time       read      write
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
           fetch meta data          0          6          6                     0.100s
            Create Schemas          0          0          0                     0.006s
          Create SQL Types          0          0          0                     0.008s
             Create tables          0          8          8                    18.274s
            Set Table OIDs          0          4          4                     0.012s
        create primary key          0          3          3                    16.418s
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
ml_migratedb.ml_order_line          0     800000     800000    56.9 MB       1m10.368s    57.023s    1m7.093s
    ml_migratedb.ml_orders          0      80000      80000     3.5 MB          7.942s     0.666s     5.001s
  ml_migratedb.orders_orig          0      30000      30000     1.2 MB         10.493s     0.295s     0.078s
  ml_migratedb.ml_tinyint1          0        201        201     4.3 kB          0.038s     0.002s     0.001s
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
   COPY Threads Completion          0          4          4                  1m10.339s
    Index Build Completion          0          2          2                   1m3.443s
            Create Indexes          0          2          2                   1m0.745s
           Reset Sequences          0          0          0                     2.386s
              Primary Keys          0          0          0                     0.000s
       Create Foreign Keys          0          0          0                     0.000s
           Create Triggers          0          0          0                     0.001s
          Install Comments          0          1          1                     0.025s
--------------------------  ---------  ---------  ---------  ---------  --------------  ---------  ---------
         Total import time          ✓     910201     910201    61.6 MB       3m16.939s
```

Migrating MySQL to Yugabyte best practices

The workflow is similar, but we suggest the following extra steps:

- Gather row counts from all tables being migrated
- If > 40 tables, Yugabyte Universe should set ysql_num_shards_per_tserver = 1, otherwise use the default of ysql_num_shards_per_tserver = 8.
    - Any tables and indexes with > 1M rows need to be split into 24 tablets by adding SPLIT INTO X TABLETS on CREATE TABLE statement
        - SPLIT INTO clause will specify the number of tablets to be created for the table. The hash range is then evenly split across those tablets.
        - Any tables with < 1M rows will not need to be SPLIT and will automatically be created with 1 shard/tablet

- Use the DATA ONLY option to move data after database and tables have been created

Yugabyte Branch of pgloader

- We create primary key on empty tables avoiding an expensive operation in Yugabyte
- We create indexes in single thread after table completion
- We have option to change the table schema to add YugabyteDB constructs like number of tablets, changing primary key
- We have avoided using savepoints and use Yugabyte's COPY command

Summary

- Migrate MySQL, MS SQL, SQL Lite, and Postgres databases to YugabyteDB
- Flexible command options to customize the migration
- Proven, Open Source migration tool
- Optimized to work with YugabyteDB

References/Acknowledgements

- YSQL Loader: https://docs.yugabyte.com/latest/integrations/ysql-loader/#root
- pgloader documentation: https://pgloader.readthedocs.io/en/latest/intro.html
- Yugabyte Branch of pgloader: https://github.com/yugabyte/pgloader

Thank You

Join us on Slack: yugabyte.com/slack

Star us on Github: github.com/yugabyte/yugabyte-db