

# MySQL 迁移 LightDB 手册

## 1 简介

ltloader 工具可以实现 mysql 到 LightDB 的迁移，支持迁移表结构，字段名，索引和外键约束，这里介绍使用 ltloader 工具来迁移 MySQL 数据到 LightDB 的步骤及注意事项。

## 2 具体步骤

### 2.1 配置迁移文件

示例环境如下：

mysql: 地址 192.168.184.136; 源库名: test ; 用户名:root 密码: 12345678

lightdb: 地址 localhost ; 目标库名称:test (需要预先建好) ; 用户名:lightdb

下面为示例的配置文件 test.loader，内容如下（可修改文件中 from mysql 和 into postgresql 后对应的连接信息使用）：

```
load database
  from mysql://root:12345678@192.168.184.136/test
  into postgresql://lightdb@localhost/test
WITH include drop, create tables, create indexes, reset sequences, foreign keys
```

```
CAST type int with extra auto_increment to serial drop typemod,
```

```
type int to int drop typemod,
```

```
type tinyint with extra auto_increment to serial drop typemod,
```

```
type smallint with extra auto_increment to serial drop typemod,
```

```
type mediumint with extra auto_increment to serial drop typemod,
```

```
type bigint with extra auto_increment to bigserial drop typemod,
```

```
type tinyint when (= 1 precision) to boolean using tinyint-to-boolean,
```

```
type bit when (= 1 precision) to boolean drop typemod using bits-to-boolean,
```

```
type bit to bit drop typemod using bits-to-hex-bitstring,
```

```
type tinyint when unsigned to smallint drop typemod,
```

```
type smallint when unsigned to integer drop typemod,
```

```
type mediumint when unsigned to integer drop typemod,
```

```
type integer when unsigned to bigint drop typemod,
```

```
type tinyint to smallint drop typemod,
```

```
type smallint to smallint drop typemod,
```

```
type mediumint to integer drop typemod,
```

```
type integer to integer drop typemod,
```

```
type bigint to bigint drop typemod,

type float to numeric,
type double to numeric,
type numeric to numeric keep typemod,
type decimal to decimal keep typemod

BEFORE LOAD DO
$$ select 1; $$;
```

## 2.2 MySQL8.0 注意点 (非 8.0 不用修改)

ltloader 不支持 MySQL8.0 中较新的 SHA2 身份验证模式，需修改 my.cnf 添加如下配置重启：

```
default-authentication-plugin=mysql_native_password
```

## 2.3 执行迁移命令

执行以下命令（上述迁移文件，因为配置的 lightdb 用户访问只能本地访问，需要在迁移的 lightdb 主机执行），test.load 为配置文件名，然后就会根据配置的策略迁移表结构和数据，ltloader 命令在安装 lightDB 后可直接使用。

```
ltloader test.load
```

命令运行结果如下：

```
2021-11-04T19:47:16.133000+08:00 LOG Migrating from #<MYSQL-CONNECTION
mysql://root@192.168.184.136:3306/test {10063AC833}>
2021-11-04T19:47:16.133000+08:00 LOG Migrating into #<PGSQL-CONNECTION
pgsql://lightdb@localhost:5432/test {10063ADA53}>
2021-11-04T19:47:16.691000+08:00 LOG report summary reset
table name errors rows bytes total time
-----
before load 0 1 0.010s
fetch meta data 0 38 0.073s
Create Schemas 0 0 0.002s
Create SQL Types 0 0 0.004s
Create tables 0 26 0.045s
Set Table OIDs 0 13 0.003s
-----
test_load.t1 0 0 0.018s
test_load.t10 0 1 0.0 kB 0.012s
test_load.t12 0 0 0.019s
test_load.t11 0 0 0.012s
test_load.t2 0 0 0.031s
test_load.t13 0 0 0.012s
test_load.t4 0 0 0.037s
test_load.t3 0 0 0.015s
test_load.t6 0 0 0.041s
test_load.t5 0 0 0.011s
test_load.t8 0 1 0.0 kB 0.030s
test_load.t7 0 0 0.017s
test_load.t9 0 2 0.0 kB 0.010s
-----
```

COPY Threads Completion	0	4		0.059s
Create Indexes	0	25		0.111s
Index Build Completion	0	25		0.083s
Reset Sequences	0	0		0.015s
Primary Keys	0	13		0.014s
Create Foreign Keys	0	0		0.000s
Create Triggers	0	0		0.001s
Set Search Path	0	1		0.001s
Install Comments	0	0		0.000s
-----				
Total import time	√	4	0.1 kB	0.284s

## 2.4 检测 rum 索引

少数情况下，如 varchar 字段上的 btree 索引在迁移到 LightDB 后会转为 rum 索引，需要手动转换。如下方式可以查找是否有 rum 索引并修改。rum 索引类似于 GIN 索引，主要用于全文检索，在存储上和 ES 更接近，相关度查询时性能比 GIN 索引快 1 倍（注：其缺点是索引大 2 倍，索引速度慢 1/3）。

可通过如下方式修改为其他索引，先查询使用 rum 索引的表，sql 如下：

```
select * from pg_catalog.pg_indexes where indexdef like '%rum (%)';
```

执行结果如下：

```
test=# select * from pg_catalog.pg_indexes where indexdef like '%rum (%)';
schemaname | tablename | indexname | tablespace | indexdef
-----+-----+-----+-----+-----
public | test_rum | idx_60507_uid | | CREATE INDEX idx_60507_uid ON public.test_rum
USING
rum
(name)
(1 row)
```

复制 indexdef 列的内容，并将 rum 修改为 btree(这里也可以是其他需要的索引类型)，修改后的示例内容如下：

```
CREATE INDEX idx_60507_uid ON public.test_rum USING btree (name)
```

然后删除 rum 索引

```
drop index idx_60507_uid;
```

最后执行保存的 sql，创建对应索引

```
CREATE INDEX idx_60507_uid ON public.test_rum USING btree (name)
```

## 3 注意点及常见问题

### 3.1 mysql8.0 最新认证方式不支持

报错信息如下：

```
2018-04-17T23:56:47.055000Z NOTICE Starting pgloader, log system is ready.
2018-04-17T23:56:47.146000Z LOG Migrating from #<MYSQL-CONNECTION
mysql://archiveit@wbgrp-svc413:3306/archiveit3 {10097067C3}>
2018-04-17T23:56:47.154000Z LOG Migrating into #<PGSQL-CONNECTION
pgsql://archiveit@UNIX:5432/archiveit3 {1009941FF3}>
KABOOM!
FATAL error: Failed to connect to mysql at "wbgrp-svc413" (port 3306) as user "archiveit": Condition
```

```
QMYND:MYSQL-UNSUPPORTED-AUTHENTICATION was signalled.
An unhandled error condition has been signalled:
Failed to connect to mysql at "wbgrp-svc413" (port 3306) as user "archiveit": Condition
QMYND:MYSQL-UNSUPPORTED-AUTHENTICATION was signalled.
```

解决办法: my.cnf 添加如下配置:

```
default-authentication-plugin=mysql_native_password
```

### 3.2 rum 索引不支持 unique

报错信息如下:

```
2021-11-04T19:47:16.636000+08:00 ERROR PostgreSQL Database error
0A000: access method "rum" does not support unique indexes
QUERY: CREATE UNIQUE INDEX idx_60455_uid ON test.tab_testtext USING
rum(uid);
```

解决方法: 复制报错中的 sql, 修改类型为 btree, 然后在 ltsql 中执行。

```
CREATE UNIQUE INDEX idx_60455_uid ON test.tab_testtext USING btree(uid);
```

### 3.3 迁移后的模式为与数据库同名的模式(schema) , 能否迁移到其他 schema?

配置文件增加 ALTER SCHEMA, 示例修改如下:

```
load database
  from mysql://root:12345678@192.168.184.136/test
  into postgresql://lightdb@localhost/test
WITH include drop, create tables, create indexes, reset sequences, foreign keys
```

```
CAST type int with extra auto_increment to serial drop typemod,
type int to int drop typemod,
type tinyint with extra auto_increment to serial drop typemod,
type smallint with extra auto_increment to serial drop typemod,
type mediumint with extra auto_increment to serial drop typemod,
type bigint with extra auto_increment to bigserial drop typemod,

type tinyint when (= 1 precision) to boolean using tinyint-to-boolean,

type bit when (= 1 precision) to boolean drop typemod using bits-to-boolean,
type bit to bit drop typemod using bits-to-hex-bitstring,

type tinyint when unsigned to smallint drop typemod,
type smallint when unsigned to integer drop typemod,
type mediumint when unsigned to integer drop typemod,
type integer when unsigned to bigint drop typemod,

type tinyint to smallint drop typemod,
type smallint to smallint drop typemod,
type mediumint to integer drop typemod,
type integer to integer drop typemod,
```

```
type bigint to bigint drop typemod,  
  
type float to numeric,  
type double to numeric,  
type numeric to numeric keep typemod,  
type decimal to decimal keep typemod  
  
ALTER SCHEMA 'test' RENAME TO 'public'  
  
BEFORE LOAD DO  
  $$ select 1; $$;
```

### 3.4 Datetime 转换后带时区，如果需要不带时区。

要不带时区，在配置文件里添加如下转换：

```
type datetime to timestamp
```