

```
http://dev.mysql.com/doc/refman/en/using-systemd.html
Process: 11971 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited,
status=0/SUCCESS)
Main PID: 11993 (mysqld)
Status: "Server is operational"
Tasks: 649 (limit: 10000)
CGroup: /system.slice/mysql.service
        993 /usr/sbin/mysqld
Aug 11 12:49:54 linux systemd[1]: Starting MySQL Server...
Aug 11 12:49:56 linux systemd[1]: Started MySQL Server.
```

验证结论：系统级别和进程级别设置方式 2 选 1 即可，但是都需要重启 MySQL 服务生效。

4.16 mysql 5.6 升级到 8.0 失败一例处理

作者：付祥

1. 背景

当前 zabbix 系统 mysql 版本 5.6 ,数据量有 1.5T 左右,存在按天分区的一些大表,执行添加字段操作将会非常耗时,曾经遇到过给一张几百 GB 的 history 表加一个字段,使用 pt-osc 工具,跑了 2 天也没执行完;为了使用 mysql 8.0 的即时加列功能,于是决定升级到 mysql 8.0 。

2. 升级过程

因为数据量比较大,采用数据逻辑导出导入方式升级将会非常慢,不推荐,故采取 In-Place 方式升级,根据官方文档提供的升级路径,需要先从 5.6 升级到 5.7 ,然后再升级到 8.0 。

从 5.6 升级到 5.7.35 非常顺利,当从 5.7.35 升级到 8.0.25 时,升级失败,报错信息如下:

```
2021-07-20T07:33:18.138368Z 1 [ERROR] [MY-011006] [Server] Got error 197 from SE
while migrating tablespaces.
2021-07-20T07:33:18.145105Z 0 [ERROR] [MY-010020] [Server] Data Dictionary
initialization failed.
2021-07-20T07:33:18.145502Z 0 [ERROR] [MY-010119] [Server] Aborting
2021-07-20T07:33:40.435143Z 0 [System] [MY-010910] [Server]
/usr/local/mysql/bin/mysqld: Shutdown complete (mysqld 8.0.25) MySQL Community
Server - GPL.
```

单从错误信息来看，似乎是迁移表空间失败，数据字典不能初始化，导致 mysql 不能启动，这里引发了一个问题思考：为何升级到 5.7 没问题？

3. 解决过程

在错误信息不明显的情况下，降了几个 mysql 8.0 小版本升级测试，终于在 8.0.15 得到了有价值的错误提示信息：

```
2021-07-20T12:25:06.672826Z 1 [ERROR] [MY-011014] [Server] Found partially
upgraded DD. Aborting upgrade and deleting all DD tables. Start the upgrade
process again.
2021-07-20T12:25:06.773766Z 1 [Warning] [MY-012351] [InnoDB] Tablespace 7314,
name 'zabbix/#sql-ib104-715696445', file './zabbix/#sql-ib104-715696445.ibd' is
missing!
2021-07-20T12:25:06.834751Z 0 [ERROR] [MY-010020] [Server] Data Dictionary
initialization failed.
```

8.0 以前数据字典信息分布在 server 层、mysql 库下的系统表和 InnoDB 内部系统表三个地方，数据字典分散存储，DDL 没有原子性。

8.0 以后元数据信息全部存储在 InnoDB dictionary table 中，并且存储在单独的表空间 mysql.ibd 里，DDL 具有原子性。

因为数据字典的管理存储方式发生了变化，升级到 8.0 数据字典需要做迁移转换，故 5.6->5.7 没问题，5.7->8.0 由于缺失 ./zabbix/#sql-ib104-715696445.ibd 文件导致了升级失败。

当前环境 zabbix 数据库目录下并不存在 #sql-ib104-715696445.ibd 文件，只有一个 #sql- 开头的 frm 文件

```
[root@GZ-DB-6CU552YR4V zabbix]# ls -l ./#sql*
-rw-rw---- 1 mysql mysql 8808 5 9 2020 ./#sql-8427_2008.frm
[root@GZ-DB-6CU552YR4V zabbix]#
```

解析 frm 文件可以通过 mysqlfrm 、 dbsake ，本文通过 dbsake 解析：

```
./dbsake frmdump ./#sql-8427_2008.frm

-- Table structure for table `#sql-8427_2008`

-- Created with MySQL Version xxxxxx

CREATE TABLE `#sql-8427_2008` (
  `eventid` bigint(20) unsigned NOT NULL,
  `source` int(11) NOT NULL DEFAULT '0',
  `object` int(11) NOT NULL DEFAULT '0',
  `objectid` bigint(20) unsigned NOT NULL DEFAULT '0',
```

```
`clock` int(11) NOT NULL DEFAULT '0',
`value` int(11) NOT NULL DEFAULT '0',
`acknowledged` int(11) NOT NULL DEFAULT '0',
`ns` int(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`eventid`),
KEY `events_1` (`source`,`object`,`objectid`,`clock`),
KEY `events_2` (`source`,`object`,`clock`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin ROW_FORMAT=COMPRESSED;
```

这种看起来有点奇怪的文件是如何产生的呢？在表执行 ddl 操作时，有两种方式：

```
ALTER TABLE (ALGORITHM=COPY)
ALTER TABLE (ALGORITHM=INPLACE)
```

ALGORITHM=INPLACE 方式是 online ddl ,如果在操作过程中异常退出，将会产生以 #sql-ib 为前缀的孤儿中间表，并伴随着以 #sql- 为前缀的不同名 frm 文件。

对于 ALTER TABLE (ALGORITHM=COPY) 方式 ddl ，如果在操作过程中异常退出，将会产生以#sql- 为前缀的孤儿临时表，并伴随着以 #sql- 为前缀的同名 frm 文件。

要查询数据库是否存在孤儿表，可以查询数据字典 INFORMATION_SCHEMA.INNODB_SYS_TABLES

对于当前升级失败环境，根据报错信息中 Tablespace 7314 查询数据字典 INFORMATION_SCHEMA.INNODB_SYS_TABLES ，发现并不存在以 #sql 开头的表，但是在 information_schema.INNODB_SYS_DATAFILES 、 INFORMATION_SCHEMA.INNODB_SYS_TABLESPACES 中却有相关记录，也就是说数据字典记录的元数据信息之间产生了不一致：

```
root@3306 (none)> SELECT * FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES WHERE NAME LIKE
'%#sql%';
Empty set (0.01 sec)

root@3306 (none)> SELECT * FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES where SPACE=7314;
Empty set (0.00 sec)

[root@3306][(none)]> select * from information_schema.INNODB_SYS_DATAFILES where
space=7314;
+-----+-----+
| SPACE | PATH |
+-----+-----+
| 7314  | ./zabbix/#sql-ib104-715696445.ibd |
+-----+-----+
1 row in set (0.01 sec)

[root@3306][(none)]> SELECT * FROM INFORMATION_SCHEMA.INNODB_SYS_TABLESPACES WHERE
```

```
SPACE =7314;
+-----+-----+-----+-----+-----+-----+
| SPACE | NAME | FLAG | FILE_FORMAT | ROW_FORMAT |
PAGE_SIZE | ZIP_PAGE_SIZE | SPACE_TYPE | FS_BLOCK_SIZE | FILE_SIZE |
ALLOCATED_SIZE |
+-----+-----+-----+-----+-----+-----+
| 7314 | zabbix/#sql-ib104-715696445 | 41 | Barracuda | Compressed |
16384 | 8192 | Single | 0 | 0 | 0
|
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

那如何清理孤儿表呢？参考官方文档：<https://dev.mysql.com/doc/refman/5.7/en/innodb-troubleshooting-datadi>ct.html

给出的解决方案前提是 INFORMATION_SCHEMA.INNODB_SYS_TABLES 存在相关元数据信息，显然和当前环境还不一样，故通过 DROP TABLE `#mysql50##sql-ib104-715696445` 方式清理孤儿中间表失败。通过删表方式不行，那通过删库方式是否可行呢？大致步骤如下：

1. create database zabbix_new;
2. alter table zabbix.xxxxxx rename to zabbix_new.xxxxxx;
3. drop database zabbix;
4. SELECT * FROM INFORMATION_SCHEMA.INNODB_SYS_TABLESPACES WHERE SPACE =7314;

经过测试，即使删除了 zabbix 库，残留的元数据 #sql-ib104-715696445.ibd 依然存在。

数据字典存放在共享表空间 ibdata1 中，要解决此问题，只有绕过 ibdata1 文件，想到的有如下两种方案：

3.1. 逻辑导出导入升级到 mysql 8.0

常用的工具有 mysqldump 、 mydumper ，其中 mydumper 支持按表并发，能极大提升效率

3.2. 传输表空间+ In-Place 升级到 mysql 8.0

因为没有其他机器提供及磁盘空间也不充足，故采取单机多实例进行表空间传输，大致步骤如下：

1. 导出用户信息及 zabbix 元数据

```
mysqldump -uroot -p -B mysql -E -R --triggers --hex-blob --set-gtid-purged=off --
single-transaction --master-data=2 >zabbix-metadata-01.sql
mysqldump -uroot -p -B zabbix -E -R --triggers --hex-blob --no-data --set-gtid-
purged=off --single-transaction --master-data=2 >zabbix-metadata-02.sql
```

2.初始化一个 5.7 版本 3307 端口实例并启动

3.加载用户信息及 zabbix 元数据

```
mysql -h127.0.0.1 -P3307 -uroot -p < zabbix-metadata-01.sql
mysql> flush privileges;
mysql -h127.0.0.1 -P3307 -uroot -p < zabbix-metadata-02.sql
```

4.新实例 zabbix 库丢弃表空间

```
mysql -uroot -p -NBe "select concat('alter table ',TABLE_NAME,' discard tablespace;')
from information_schema.TABLES where TABLE_SCHEMA='zabbix'";
use zabbix;
set foreign_key_checks=0;
alter table xxxxxx discard tablespace;
```

5.干净的关闭原来 mysql 实例，并移动 zabbix 库下 ibd 文件到新实例对应路径，如果空间充足，最好保留原文件，通过 cp 方式

```
mv /data/3306/zabbix/*.ibd /data/3307/zabbix/
```

6. 导入表空间，这一步也比较费时，因为需要修改 ibd 文件中页的 space id 和数据字典 space id 一致等操作

```
alter table xxxxxx import tablespace;
```

导入表空间时，需要数据字典定义的 row format 和 ibd 文件中记录的 row format 一致，否则将报错：

```
ERROR 1808 (HY000): Schema mismatch (Table has ROW_TYPE_DYNAMIC row format, .ibd file
has ROW_TYPE_COMPACT row format.)
```

通过 alter table xxxxxx row_format=compact 修改行格式，第 5 步 mv 过来的 ibd 文件会被删除，导致数据丢失，故要先 mv 走 ibd 文件再修改 row format

```
mv /data/3307/zabbix/acknowledges.ibd /data/3307/zabbix/acknowledges.ibd.bak alter
table acknowledges row_format=compact;
mv /data/3307/zabbix/acknowledges.ibd.bak /data/3307/zabbix/acknowledges.ibd alter
table acknowledges import tablespace;
```

7.In-Place 方式升级新实例到 8.0