

故障分析 | 为什么 MySQL 8.0.13 要引入新参数 sql_require_primary_key ?

原创 王祥 爱可生开源社区 2025年01月20日 16:30 上海



作者：王祥，爱可生 DBA 团队成员，主要负责 MySQL 故障处理和性能优化。对技术执着，为客户负责。

爱可生开源社区出品，原创内容未经授权不得随意使用，转载请联系小编并注明来源。

本文约 1100 字，预计阅读需要 2 分钟。

1 问题描述

客户在测试环境给一批无主键表新增自增主键，发现个别表在新增自增主键后出现了主从数据不一致。

为什么有些表正常而有些表会主从数据不一致呢？

本次故障分析案例，将解释为什么 MySQL 8.0.13 要引入新参数 sql_require_primary_key ?

测试环境

- MySQL：8.0.25
- 架构：1 Primary – 1 Secondary

2 问题复现

根据客户描述的两种情况分别复现。

新建两张表。

```
mysql> create table t1(n int,m int);
Query OK, 0 rows affected (0.09 sec)

mysql> create table t2(n int,m int);
Query OK, 0 rows affected (0.01 sec)
```

在 t1 表中写入两行数据。

```
mysql> insert into t1 values(1,1),(2,2);
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

在 t2 表里写入两行数据，与 t1 的写入方式区别：会话一先执行后提交，会话二后执行先提交。

Session 1	Session 2
begin; insert into t2 values(1,1);	
	begin; insert into t2 values(2,2); commit;
commit;	

分别给 `t1` 、 `t2` 表新增自增主键。

```
mysql> alter table t1 add id int primary key auto_increment first;
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table t2 add id int primary key auto_increment first;
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

分别在主从实例上查询 `t1` 、 `t2` 表，看数据是否一致。

Primary

```
mysql> select * from t1;
+----+-----+-----+
| id | n     | m     |
+----+-----+-----+
| 1  |      1 |      1 |
| 2  |      2 |      2 |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from t2;
+----+-----+-----+
| id | n     | m     |
+----+-----+-----+
| 1  |      1 |      1 |
| 2  |      2 |      2 |
+----+-----+-----+
2 rows in set (0.00 sec)
```

两个表数据一致

Secondary

```
Database changed
mysql> select * from t1;
+----+-----+-----+
| id | n     | m     |
+----+-----+-----+
| 1  |      1 |      1 |
| 2  |      2 |      2 |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from t2;
+----+-----+-----+
| id | n     | m     |
+----+-----+-----+
| 1  |      2 |      2 |
| 2  |      1 |      1 |
+----+-----+-----+
2 rows in set (0.00 sec)
```

两个表数据不一致

从上面两图对比可以发现 `t1` 表主从数据是一致的，`t2` 表主从数据不一致。

问题复现！

如果是读者遇到这个问题会如何分析呢？

问题分析：

对于以上问题[官方文档](#)^[1] 相关说明：

• Adding an `auto_increment` column to a table with `ALTER TABLE` might not produce the same ordering of the rows on the replica and the source. This occurs because the order in which the rows are numbered depends on the specific storage engine used for the table and the order in which the rows were inserted. If it is important to have the same order on the source and replica, the rows must be ordered before assigning an `auto_increment` number. Assuming that you want to add an `auto_increment` column to a table `t1` that has columns `ee11` and `ee12`, the following statements produce a new table `t2` identical to `t1` but with an `auto_increment`

大致意思是说：给表增加自增主键时，主从库表中是否能产生相同的自增值，这取决于主从插入数据的顺序。

如何理解这句话中的“主从插入数据的顺序”？我们还是以上面 `t2` 表作为例子来解释。会话一先执行后提交，会话二后执行先提交。

Session 1	Session 2
begin; insert into t2 values(1,1);	
	begin; insert into t2 values(2,2); commit;
commit;	

执行完以上操作分别在主从查看 t1 表有何不同。

Primary

```
mysql> select * from t2;
+-----+-----+
| n     | m     |
+-----+-----+
|      1 |      1 |
|      2 |      2 |
+-----+-----+
2 rows in set (0.00 sec)
```

Secondary

```
mysql> select * from t2;
+-----+-----+
| n     | m     |
+-----+-----+
|      2 |      2 |
|      1 |      1 |
+-----+-----+
2 rows in set (0.00 sec)
```

从上图可以发现主库和从库 t2 两条数据的顺序是相反的，为什么会出现这种情况？

因为在 InnoDB 表中，如果表没有指定主键并且表上也无非空唯一键时，InnoDB 会使用内部 RowID 作为数据行的主键，并且 RowID 是在语句执行时已经分配，所以语句先执行的数据行分配到的 RowID 就越小，后执行的语句分配的 RowID 就越大。

上面的测试会话一语句先执行所以分配的 RowID 是较小值，会话二是后执行所以分配的 RowID 是较大值，最终主库看到的顺序是 (1,1),(2,2)。从库上的情况就刚好和主库相反，因为主库会话一先执行但并没有提交，会话二后执行先提交所以从库接收到的 binlog 顺序是会话二先执行分配的 RowID 更小，会话一后执行分配的 RowID 更大。

在 InnoDB 引擎中数据的排列是按照主键的大小顺序来排列的。最终导致在主从上查询 t2 表出现数据顺序不一致的情况。之后给 t2 添加自增主键时，自增主键的值是按照数据在表中的排列顺序赋值的，所以最终导致主从 t2 的数据不一致。

3 问题解决

上述官方手册中提供了以下语句解决，该方案需要确保操作期间无新增数据否则会导致新增数据丢失。

```
CREATE TABLE t2 LIKE t1;
ALTER TABLE t2 ADD id INT AUTO_INCREMENT PRIMARY KEY;
INSERT INTO t2 SELECT * FROM t1 ORDER BY col1, col2;
DROP TABLE t1;
RENAME TABLE t2 TO t1;
```

4 总结

如果表较大，这个方法还是比较耗时的。

最好的方法还是前期建表就建好主键，故在 MySQL 8.0.13 引入参数 `sql_require_primary_key` 可以强制要求在建表指定主键，避免该案例的问题。还可以尝试使用 SQL 质量管理平台 SQLE，可以有效保障建表合理性。

另外在 MySQL 8.0.30 引入了 GIPK(Generated Invisible Primary key) 不可见主键^[2]，可以将参数 `sql_generate_invisible_primary_key` 开启。如果表没有指定主键则会自动生成 `my_row_id` 做为表的主键。

参考资料

- [1] replication-features-auto-increment: <https://dev.mysql.com/doc/refman/8.0/en/replication-features-auto-increment.html>
- [2] gipk: <https://dev.mysql.com/doc/refman/8.0/en/create-table-gipks.html>

本文关键字：#MySQL# #主从复制# #主键自增#



推荐阅读

- 故障分析 | 如何解决由触发器导致 MySQL 内存溢出？
- 技术分享 | MySQL 主从复制过滤新增库表过滤方案
- 技术分享 | MySQL Undo 工作机制历史演变
- 技术分享 | MySQL 隐式转换必知必会
- 技术分享 | MySQL VARCHAR 最佳长度评估实践
- 故障分析 | TCP 缓存超负荷导致的 MySQL 连接中断
- 故障分析 | 一则 MySQL 从节点 hung 死问题分析

覆盖开发至生产环境的 SQL 审核和管理。



支持主流的开源、商业、国产数据库，为开发和运维提供流程自动化能力，提升上线效率，提高数据质量。



请添加小助手加入
SQLE 技术交流群

- ✨ Github : <https://github.com/actiontech/sqlc>
- 📖 文档 : <https://actiontech.github.io/sqlc-docs/>
- 🌐 官网 : <https://opensource.actionsky.com/sqlc/>

👥 微信群：请添加小助手加入 ActionOpenSource
🔗 商业支持：<https://www.actionsky.com/sql>e

微信扫描小程序码，进行在线咨询预约：



商业支持团队联系方式如下：
400-820-6580 / 13916131869 / 18930110869



MySQL 238 # 主从复制 12 # 主键自增 1

MySQL · 目录

◀ 上一篇

MySQL 核心模块揭秘 | 49 期 | 更新记录的 Undo 日志

下一篇 ▶

第 50 期：根据 EXPLAIN EXTRA 栏提示进行优化（二）