

解读命令输出

目录

1

mysqldump

2

innodb_trx

课堂练习

以下关于 mysqldump 说法正确的是:

- A. --quick 的默认参数是 ON, 应该保持不变。
- B. 指定 --single-transaction 参数,会在输出文件里输出 start transaction with consistent snapshot。
- C. 同时指定 --flush-logs --single-transaction 的时候,每备份一个库,都会执行一次 flush logs。
- D. 单独备份一个库的时候, 需要指定 --routines 才可以备份出存储过程。

mysqldump 输出

```
-- MySQL dump 10.13  Distrib 8.0.40, for Linux (x86_64)
--
-- Host: 127.0.0.1    Database: x1
--
-- Server version      8.0.40-debug

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
SET @MYSQLDUMP_TEMP_LOG_BIN = @@SESSION.SQL_LOG_BIN;
SET @@SESSION.SQL_LOG_BIN= 0;
```

mysqldump 输出

```
--
-- GTID state at the beginning of the backup
--

SET @@GLOBAL.GTID_PURGED=/*!80000 '+'*/ 'f48f5f9f-a9f7-11ef-b12b-5254000d6ff4:1-25';

--
-- Temporary view structure for view `v1`
--

DROP TABLE IF EXISTS `v1`;
/*!50001 DROP VIEW IF EXISTS `v1`*/;
SET @saved_cs_client      = @@character_set_client;
/*!50503 SET character_set_client = utf8mb4 */;
/*!50001 CREATE VIEW `v1` AS SELECT
  1 AS `id`,
  1 AS `c`*/;
SET character_set_client = @saved_cs_client;

--
-- Table structure for table `xlt`
--
```


mysqldump 输出

```
DROP TABLE IF EXISTS `xlt`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `xlt` (
  `id` int NOT NULL AUTO_INCREMENT,
  `c` int DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `xlt`
--

LOCK TABLES `xlt` WRITE;
/*!40000 ALTER TABLE `xlt` DISABLE KEYS */;
INSERT INTO `xlt` VALUES (1,1),(2,2),(3,2),(4,1),(5,1),(6,1),(7,1),(8,1),(9,1),(10,2);
/*!40000 ALTER TABLE `xlt` ENABLE KEYS */;
UNLOCK TABLES;
```

mysqldump 输出

```
--
-- Final view structure for view `v1`
--

/*!50001 DROP VIEW IF EXISTS `v1`*/;
/*!50001 SET @saved_cs_client      = @@character_set_client */;
/*!50001 SET @saved_cs_results    = @@character_set_results */;
/*!50001 SET @saved_col_connection = @@collation_connection */;
/*!50001 SET character_set_client  = utf8mb4 */;
/*!50001 SET character_set_results = utf8mb4 */;
/*!50001 SET collation_connection  = utf8mb4_0900_ai_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`%` SQL SECURITY DEFINER */
/*!50001 VIEW `v1` AS select `xlt`.`id` AS `id`,`xlt`.`c` AS `c` from `xlt` where (`xlt`.`id` < 10) */;
/*!50001 SET character_set_client  = @saved_cs_client */;
/*!50001 SET character_set_results = @saved_cs_results */;
/*!50001 SET collation_connection  = @saved_col_connection */;
SET @@SESSION.SQL_LOG_BIN = @MYSQLDUMP_TEMP_LOG_BIN;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

— Dump completed on 2025-01-08 19:53:12

mysqldump 怎么解决procedure依赖的问题?

```
mysql> CREATE PROCEDURE procedure1()  
-> BEGIN  
->   -- 调用另一个存储过程 procedure2  
->   CALL procedure2(10);  
-> END //  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> CREATE PROCEDURE procedure2(IN param1 INT)  
-> BEGIN  
->   -- 存储过程的内容, 例如:  
->   SELECT CONCAT('Procedure 2 called with parameter: ', param1);  
-> END //  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>  
mysql>  
mysql> DELIMITER ;  
mysql> call procedure1();  
+-----+  
| CONCAT('Procedure 2 called with parameter: ', param1) |  
+-----+  
| Procedure 2 called with parameter: 10                  |  
+-----+  
1 row in set (0.00 sec)  
  
Query OK, 0 rows affected (0.00 sec)
```


mysqldump 怎么解决procedure依赖的问题?

```
mysql> CREATE PROCEDURE procedure1()  
-> BEGIN  
->    -- 调用另一个存储过程 procedure2  
->    CALL procedure2(10);  
-> END //  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> CREATE PROCEDURE procedure2(IN param1 INT)  
-> BEGIN  
->    -- 存储过程的内容, 例如:  
->    insert into table_not_exis values(1,1);  
-> END //  
Query OK, 0 rows affected (0.00 sec)  
  
mysql>  
mysql>  
mysql> DELIMITER ;  
mysql> call procedure1;  
ERROR 1146 (42S02): Table 'x1.table_not_exis' doesn't exist
```

目录

1 mysqldump

2 innodb_trx

表结构

```
mysql> select * from t;
```

id	c	d
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

```
5 rows in set (0.00 sec)
```

```
mysql> show create table t\G
```

```
***** 1. row *****
```

```
Table: t
```

```
Create Table: CREATE TABLE `t` (
```

```
  `id` int NOT NULL,
```

```
  `c` int DEFAULT NULL,
```

```
  `d` int DEFAULT NULL,
```

```
  PRIMARY KEY (`id`),
```

```
  KEY `c` (`c`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

```
1 row in set (0.00 sec)
```


课堂练习:以下结果是哪个语句序列产生的

```
***** 1. row *****
      trx_id: 421715323446248
      trx_state: RUNNING
      trx_started: 2025-01-11 18:05:18
      trx_requested_lock_id: NULL
      trx_wait_started: NULL
      trx_weight: 2
      trx_mysql_thread_id: 12
      trx_query: NULL
      trx_operation_state: NULL
      trx_tables_in_use: 0
      trx_tables_locked: 1
      trx_lock_structs: 2
      trx_lock_memory_bytes: 1192
      trx_rows_locked: 3
      trx_rows_modified: 0
      trx_concurrency_tickets: 0
      trx_isolation_level: REPEATABLE READ
      trx_unique_checks: 1
      trx_foreign_key_checks: 1
      trx_last_foreign_key_error: NULL
      trx_adaptive_hash_latched: 0
      trx_adaptive_hash_timeout: 0
      trx_is_read_only: 0
      trx_autocommit_non_locking: 0
      trx_schedule_weight: NULL
```

- A. begin; select * from t limit 3;
- B. begin; select * from t limit 3 for share;
- C. begin; select * from t limit 3 for update;
- D. begin; update t set c=c+1 limit 3;

事务状态

session1	session2
begin select * from t where id= 1 for share;	
	begin; update t set c=c+1 where id=1;

```
      trx_id: 421715323445240
      trx_state: RUNNING
      trx_started: 2025-01-11 19:08:26
trx_requested_lock_id: NULL
      trx_wait_started: NULL
      trx_weight: 2
      trx_mysql_thread_id: 11
      trx_query: NULL
trx_operation_state: NULL
      trx_tables_in_use: 0
      trx_tables_locked: 1
      trx_lock_structs: 2
```

```
      trx_id: 109297
      trx_state: LOCK WAIT
      trx_started: 2025-01-11 19:08:30
trx_requested_lock_id: 140240346735592:164:968:4:2:140240278841632
      trx_wait_started: 2025-01-11 19:08:30
      trx_weight: 2
      trx_mysql_thread_id: 19
      trx_query: update t set c=c+1 where id=1
trx_operation_state: starting index read
      trx_tables_in_use: 1
      trx_tables_locked: 1
      trx_lock_structs: 2
```

trx_operation_state

"adding foreign keys";	"inserting";
"calculating upper bound for table rows";	"inserting index entries";
"checking table";	"marking transaction as prepared in TC";
"cluster: purging delete marked records";	"preparing";
"committing";	"purge trx";
"counting records";	"read meta-data file";
"creating index";	"renaming table";
"creating table";	"returning various info to MySQL";
"discarding tablespace";	"rollback";
"doing SYNC commit";	"rollback of internal trx on stats tables";
"doing SYNC index";	"rollback of SQL statement";
"dropping table";	"rollback to a savepoint";
"estimating records in index range";	"secondary: purge delete marked records";
"fetching FT table rows count";	"setting auto-inc lock";
"fetching rows";	"setting table lock";
"flushing log";	"setting table lock for creating or dropping index";
"inserting table";	"sleeping before entering InnoDB";
"locking table";	"starting index read";
"purging delete marked records";	"try to drop any indexes after an aborted index creation";
"reading table";	"unlock_row";
"renaming table";	"updating or deleting";

update t set c=c+1 where id=1;

```
      trx_id: 109297
      trx_state: LOCK WAIT
      trx_started: 2025-01-11 19:08:30
trx_requested_lock_id: 140240346735592:164:968:4:2:140240278841632
      trx_wait_started: 2025-01-11 19:08:30
      trx_weight: 2
      trx_mysql_thread_id: 19
      trx_query: update t set c=c+1 where id=1
      trx_operation_state: starting index read
      trx_tables_in_use: 1
      trx_tables_locked: 1
      trx_lock_structs: 2
      trx_lock_memory_bytes: 1192
      trx_rows_locked: 1
      trx_rows_modified: 0
      trx_concurrency_tickets: 0
      trx_isolation_level: REPEATABLE READ
      trx_unique_checks: 1
      trx_foreign_key_checks: 1
trx_last_foreign_key_error: NULL
      trx_adaptive_hash_latched: 0
      trx_adaptive_hash_timeout: 0
      trx_is_read_only: 0
      trx_autocommit_non_locking: 0
      trx_schedule_weight: 1
```

```
begin;(rc)
select ... (rc)
update (rc)
set iso.. = rr
delete (rc)
```

```
start trasaction; (rr)
```

```
start trasaction with
consistent snapshot;(rc)
```

show slave status\G

```
mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for source to send event
        Master_Host: 127.0.0.1
        Master_User: root
        Master_Port: 13000
        Connect_Retry: 1
        Master_Log_File: master-bin.000001
  Read_Master_Log_Pos: 1377
        Relay_Log_File: slave-relay-bin.000002
        Relay_Log_Pos: 1595
  Relay_Master_Log_File: master-bin.000001
    Slave_IO_Running: Yes
   Slave_SQL_Running: Yes
      Replicate_Do_DB:
  Replicate_Ignore_DB:
      Replicate_Do_Table:
  Replicate_Ignore_Table:
  Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
          Last_Errno: 0
          Last_Error:
        Skip_Counter: 0
   Exec_Master_Log_Pos: 1377
        Relay_Log_Space: 1805
        Until_Condition: None
        Until_Log_File:
        Until_Log_Pos: 0
```

show slave status\G

```
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID: d2481773-d0e3-11ef-a37f-52540056a162
Master_Info_File: mysql.slave_master_info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Replica has read all relay log; waiting for more updates
Master_Retry_Count: 10
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set: d2481773-d0e3-11ef-a37f-52540056a162:1-6
Executed_Gtid_Set: d2481773-d0e3-11ef-a37f-52540056a162:1-6
Auto_Position: 1
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
Master_public_key_path:
Get_master_public_key: 0
Network_Namespace:
1 row in set, 1 warning (0.00 sec)
```


start slave

```
until_option:
  UNTIL {    {SQL_BEFORE_GTIDS | SQL_AFTER_GTIDS} = gtid_set
            | SOURCE_LOG_FILE = 'log_name', SOURCE_LOG_POS = log_pos
            | RELAY_LOG_FILE = 'log_name', RELAY_LOG_POS = log_pos
            | SQL_AFTER_MTS_GAPS  }
```

一个神命令

```
mysqladmin -uroot -h -P extended-status --relative --sleep 1 |  
grep Innodb_rows_inserted
```

Q&A

THANKS