MySQL 資料庫還原(實作範例)

先在 MySQL Master 建立測試資料

```
mysql> CREATE DATABASE user;
mysql> CREATE TABLE `user_powers` (
  `user_id` varchar(50) NOT NULL COMMENT '會員ID',
  `nick_name` varchar(20) NOT NULL COMMENT '會員名稱',
  `group_id` int(11) DEFAULT NULL COMMENT '會員所屬平台',
  `updated_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP    COMMENT '異動時間',
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
;
mysql> INSERT INTO `user_powers` VALUES ('1021211','ken',119201,'2021-08-10 08:33:56'),('1232123','siang',19200,'2021-08-10 08:33:51'),('18
mysql> select * from user_powers;
+----------+-----------+----------+---------------------+
| user_id  | nick_name | group_id | updated_at          |
+----------+-----------+----------+---------------------+
| 1021211  | ken       |   119201 | 2021-08-10 08:33:56 |
| 1232123  | siang     |    19200 | 2021-08-10 08:33:51 |
| 18675543 | kitty     |    19200 | 2021-08-10 08:33:51 |
| 23122141 | kiki      |  1242000 | 2021-08-12 09:01:54 |
| 24122441 | snow      |   119201 | 2021-08-10 08:33:51 |
| 4356231  | lisa      |    19200 | 2021-08-10 08:33:56 |
| 64121211 | kevin     |   389110 | 2021-08-10 08:33:51 |
| 7622441  | ben       |   213123 | 2021-08-10 08:33:56 |
| 91311233 | alex      |   389110 | 2021-08-10 08:33:56 |
| 91675543 | kay       |   123321 | 2021-08-10 08:33:56 |
| 98212321 | sara      |   119201 | 2021-08-10 08:33:51 |
+----------+-----------+----------+---------------------+
11 rows in set (0.00 sec)
```

先 dump 出資料

- 這一步不一定要做，可以直接從 binlog 的部分重作資料

```
$ mysqldump -u root -p --routines --events --single-transaction --master-data=2 --flush-logs user > user_backup_20210817.sql

## 指定資料庫: user
## master-data=2 會紀錄此備份內容是到哪一個binlog(POS點&檔名)，方便後續復原動作。
```

檢視 dump 檔的 binlog 紀錄

```
$ grep -i "CHANGE MASTER TO MASTER_LOG_FILE" user_backup_20210817.sql
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000013', MASTER_LOG_POS=154;
```

確認備份檔是在 binlog 的哪個位置

```
mysql> show master logs;
+------------------+-----------+
| Log_name         | File_size |
+------------------+-----------+
| mysql-bin.000011 |       201 |
| mysql-bin.000012 |      3068 |
| mysql-bin.000013 |       154 |   <-目前位置
+------------------+-----------+
3 rows in set (0.00 sec)
```

補充

- 環境備份：基本上產品環境下備份都是以全庫備為主，而有些重要的表資訊可能會有備份單庫或單表的需求。

- mysqldump還原注意：備份時如果不是用指定到表的方式備份，如：`mysqldump -u root -p db_name tbl_name ...` 的話，在恢復時是沒辦法直接僅恢復單一TABLE後把資料倒回去，可採用過濾備份資料把表相關資料語法另外撈出來做復原。
- 保險操作：將備份檔和 binlog 資料在臨時庫上做復原動作，根據備份檔紀錄的binlog(POS&檔名)找後續執行的binlog內容進行復原，確認資料沒問題後再將表倒回產品庫中。避免出錯造成二次傷害的狀況!!!

備份完的資料寫入

```
mysql> INSERT INTO `user_powers` VALUES ('98112111','leo',19201,'2021-08-13 12:33:56'),('38921112','shawn',19200,'2021-08-12 10:33:51');
mysql> INSERT INTO `user_powers` VALUES ('72197123','eric',19200,'2021-08-15 01:33:51');
mysql> FLUSH LOGS;

mysql> UPDATE user_powers SET nick_name = 'siangx' WHERE user_id = 1232123;
mysql> select * from user_powers;
+----------+-----------+----------+---------------------+
| user_id  | nick_name | group_id | updated_at          |
+----------+-----------+----------+---------------------+
| 1021211  | ken       |   119201 | 2021-08-10 08:33:56 |
| 1232123  | siangx    |    19200 | 2021-08-17 02:08:05 |
| 18675543 | kitty     |    19200 | 2021-08-10 08:33:51 |
| 23122141 | kiki      |  1242000 | 2021-08-12 09:01:54 |
| 24122441 | snow      |   119201 | 2021-08-10 08:33:51 |
| 38921112 | shawn     |    19200 | 2021-08-12 10:33:51 |
| 4356231  | lisa      |    19200 | 2021-08-10 08:33:56 |
| 64121211 | kevin     |   389110 | 2021-08-10 08:33:51 |
| 72197123 | eric      |    19200 | 2021-08-15 01:33:51 |
| 7622441  | ben       |   213123 | 2021-08-10 08:33:56 |
| 91311233 | alex      |   389110 | 2021-08-10 08:33:56 |
| 91675543 | kay       |   123321 | 2021-08-10 08:33:56 |
| 98112111 | leo       |    19201 | 2021-08-13 12:33:56 |
| 98212321 | sara      |   119201 | 2021-08-10 08:33:51 |
+----------+-----------+----------+---------------------+
14 rows in set (0.00 sec)
```

(假裝)誤刪資料

```
mysql> delete from user_powers where group_id = 19200;
mysql> select * from user_powers;
+----------+-----------+----------+---------------------+
| user_id  | nick_name | group_id | updated_at          |
+----------+-----------+----------+---------------------+
| 1021211  | ken       |   119201 | 2021-08-10 08:33:56 |
| 23122141 | kiki      |  1242000 | 2021-08-12 09:01:54 |
| 24122441 | snow      |   119201 | 2021-08-10 08:33:51 |
| 64121211 | kevin     |   389110 | 2021-08-10 08:33:51 |
| 7622441  | ben       |   213123 | 2021-08-10 08:33:56 |
| 91311233 | alex      |   389110 | 2021-08-10 08:33:56 |
| 91675543 | kay       |   123321 | 2021-08-10 08:33:56 |
| 98112111 | leo       |    19201 | 2021-08-13 12:33:56 |
| 98212321 | sara      |   119201 | 2021-08-10 08:33:51 |
+----------+-----------+----------+---------------------+
9 rows in set (0.00 sec)

mysql> show master logs;
+------------------+-----------+
| Log_name         | File_size |
+------------------+-----------+
| mysql-bin.000011 |       201 |
| mysql-bin.000012 |      3068 |
| mysql-bin.000013 |       816 |
| mysql-bin.000014 |       849 |
+------------------+-----------+
4 rows in set (0.00 sec)

總結-(備份完時位置在mysql-bin.000013-(154)的地方，所以異動過後的資料紀錄於 mysql-bin.000013 & mysql-bin.000014)
```

- 備份完時位置在mysql-bin.000013-(154)的地方，所以異動過後的資料紀錄於 mysql-bin.000013 & mysql-bin.000014 之間

還原一：mysqldump 的資料寫入

- 開啟新連線，暫時關閉 binlog。
- 關閉原因：匯入備份檔時，也會紀錄binlog。而這些內容對於後續復原來說是沒作用的並且會混亂(也可以選擇先備份)。

```
mysql> SET sql_log_bin=0;
mysql> use user
mysql> source user_backup_20210817.sql

## 如此就已經回到 mysqldump 時的狀態
```

還原二：從 binlog 找錯誤指令

- 若是不確定 binlog 編號，可從 `show master log` 的最大數兩三者開始找起

```
## mysqlbinlog -v: view 檔案內容，沒有加上 iv 則只能看參數使用
## cat -n: 顯示內容且帶 line number
## grep -iw: case-insensitve & 字串要精確比對

## 沒有使用絕對路徑會找不到 binlog-n，所以要補上完整路徑

$ sudo mysqlbinlog -v /var/lib/mysql/mysql-bin.000013 | cat -n | grep -iw 'delete'
$ sudo mysqlbinlog -v /var/lib/mysql/mysql-bin.000014 | cat -n | grep -iw 'delete'
    78  ### DELETE FROM `user`.`user_powers`
    84  ### DELETE FROM `user`.`user_powers`
    90  ### DELETE FROM `user`.`user_powers`
    96  ### DELETE FROM `user`.`user_powers`
   102  ### DELETE FROM `user`.`user_powers`
$ mysqlbinlog -vv /var/lib/mysql/mysql-bin.000014 | cat -n | sed -n '55,120p';

### 找到 DELETE 後要再找到它的上一個 COMMIT，也可以這樣找
$ sudo mysqlbinlog -v /var/lib/mysql/mysql-bin.000014 | cat -n | grep -iw 'commit' | head -n 5
  ## 只要是在 line 78 之前最接近的 line number w/ commit，那就是它了

## 重要是要看最近一個 commit 下一行所帶的參數，也就是 position_id，下面是看到是 467
```

```
    55  #210817  2:08:05 server id 1  end_log_pos 467 CRC32 0xa6da553d  Xid = 6379
    56  COMMIT/*!*/;
    57  # at 467
    58  #210817  2:10:26 server id 1  end_log_pos 532 CRC32 0x52ded93f  Anonymous_GTID  last_committed=1        sequence_number=2     rbr_only=y
es
    59  /*!50718 SET TRANSACTION ISOLATION LEVEL READ COMMITTED*//*!*/;
    60  SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
    61  # at 532
    62  #210817  2:10:26 server id 1  end_log_pos 604 CRC32 0xfc237d0a  Query   thread_id=57    exec_time=0     error_code=0
    63  SET TIMESTAMP=1629166226/*!*/;
    64  BEGIN
    65  /*!*/;
    66  # at 604
    67  #210817  2:10:26 server id 1  end_log_pos 666 CRC32 0x5039b136  Table_map: `user`.`user_powers` mapped to number 435
    68  # at 666
    69  #210817  2:10:26 server id 1  end_log_pos 818 CRC32 0xf6edad08  Delete_rows: table id 435 flags: STMT_END_F
    70
    71  BINLOG '
    72  khobYRMBAAAAPgAAAJoCAAAAALMBAAAAAAEABHVzZXIAC3VzZXJfcG93ZXJzAAQPDwMRBZYAPAAA
    73  DDaxOVA=
    74  khobYSABAAAAmAAAADIDAAAAALMBAAAAAAEAAgAE//AHMTIzMjEyMwZzaWFuZ3gASwAAYRsaBfAI
    75  MTg2NzU1NDMFa2l0dHkASwAAYRI57/AIMzg5MjExMTIFc2hhd24ASwAAYRT5D/AHNDM1NjIzMQRs
    76  aXNhAEsAAGESOfTwCDcyMTk3MTIzBGVyaWMASwAAYRhu/wit7fY=
    77  '/*!*/;
    78  ### DELETE FROM `user`.`user_powers`
    79  ### WHERE
    80  ###   @1='1232123' /* VARSTRING(150) meta=150 nullable=0 is_null=0 */
    81  ###   @2='siangx' /* VARSTRING(60) meta=60 nullable=0 is_null=0 */
    82  ###   @3=19200 /* INT meta=0 nullable=1 is_null=0 */
    83  ###   @4=1629166085 /* TIMESTAMP(0) meta=0 nullable=1 is_null=0 */
    84  ### DELETE FROM `user`.`user_powers`
    85  ### WHERE
    86  ###   @1='18675543' /* VARSTRING(150) meta=150 nullable=0 is_null=0 */
```

從確定的 positionID 之前恢復

```
$ mysqlbinlog --stop-position=467 /var/lib/mysql/mysql-bin.000013 /var/lib/mysql/mysql-bin.000014 --database=user --skip-gtids | mysql -uro
```

- start-position: 備份後pos點(上面未寫 start-position，則是以最早為主)
  - 若有多個binlog檔案->則視第一個binlog檔案POS點)
  - 以這個案例，應該是要寫 start-position=154
- stop-position: 誤操作前pos點 (多個binlog檔案->則為最後的binlog檔案的POS點)
- 前面的備份還原僅針對單一資料庫做還原，不會動到其他DB。
- 透過binlog還原後續資料時，如果沒指定單一資料庫，如： --database=user，那執行到一些其他DB的異動紀錄，像是已經刪除的或是插入值等相關異動，是會噴錯的喔！！
- **binlog紀錄是包含所有DB的操作內容**

回到資料庫檢查

```
mysql> select * from user_powers;
+----------+-----------+----------+---------------------+
| user_id  | nick_name | group_id | updated_at          |
+----------+-----------+----------+---------------------+
| 1021211  | ken       |   119201 | 2021-08-10 08:33:56 |
| 1232123  | siangx    |    19200 | 2021-08-17 02:08:05 |
| 18675543 | kitty     |    19200 | 2021-08-10 08:33:51 |
| 23122141 | kiki      |  1242000 | 2021-08-12 09:01:54 |
| 24122441 | snow      |   119201 | 2021-08-10 08:33:51 |
| 38921112 | shawn     |    19200 | 2021-08-12 10:33:51 |
| 4356231  | lisa      |    19200 | 2021-08-10 08:33:56 |
| 64121211 | kevin     |   389110 | 2021-08-10 08:33:51 |
| 72197123 | eric      |    19200 | 2021-08-15 01:33:51 |
| 7622441  | ben       |   213123 | 2021-08-10 08:33:56 |
| 91311233 | alex      |   389110 | 2021-08-10 08:33:56 |
| 91675543 | kay       |   123321 | 2021-08-10 08:33:56 |
| 98112111 | leo       |    19201 | 2021-08-13 12:33:56 |
| 98212321 | sara      |   119201 | 2021-08-10 08:33:51 |
+----------+-----------+----------+---------------------+
14 rows in set (0.00 sec)
```

結束，收拾工作

```
mysql> SET sql_log_bin=1;
mysql> drop database user;
```

進階備註

- 要建立 binlog 的必要條件：在 my.cnf 紀載 `log-bin=mysql-bin`
- binlog 位置：/var/lib/mysql/

```
今天早上9點不小心砍掉哪個資料庫的資料表，可以利用下面語法來恢復
$ mysqlbinlog --stop-date="2007-03-29 8:59:59"  /var/lib/mysql/bin.000001 | mysql -u root -p
$ mysqlbinlog --stop-date="2007-03-29 8:59:59"  /var/lib/mysql/bin.000001 (--database=[database] --skip-gtids) | mysql -u root -p

如果想恢復後面9點以後sql語法，可以使用
$ mysqlbinlog --start-date="2007-03-29 9:00:00"  /var/lib/mysql/bin.000001 | mysql -u root -p
$ mysqlbinlog --start-date="2007-03-29 9:00:00"  /var/lib/mysql/bin.000001 (--database=[database] --skip-gtids) | mysql -u root -p


或者是想恢復 9點到10點之間的sql語法，則下面語法
$ mysqlbinlog --start-date="2007-03-29 9:00:00"  --stop-date="2007-03-29 10:00:00" /var/lib/mysql/bin.000001 | mysql -u root -p

其實也可以不要執行，先把sql語法輸出到 /tmp/restore.sql
mysqlbinlog --start-date="2007-03-29 9:00:00"  --stop-date="2007-03-29 10:00:00" /var/lib/mysql/bin.000001 > /tmp/restore.sql

當然，也可以指定你要輸出的 database，免得檔案很大
--database=db_name，-d db_name
--host=host_name，-h host_name
```

Reference

- https://ithelp.ithome.com.tw/articles/10264923
- https://ithelp.ithome.com.tw/articles/10265911
- https://blog.wu-boy.com/2007/03/mysql-mysqlbinlog-資料庫處理二進制日誌檔案的實用工具/