MySQL - ROLLBACK & COMMIT

■ 데이터제어어 [DCL : Data Control Language]

데이터를 보호하고 관리하기 위한 목적으로 사용되며
무결성, 보안 및 권한 제어, 복구 등을 하기 위한 언어이다.

■ ROLLBACK & COMMIT

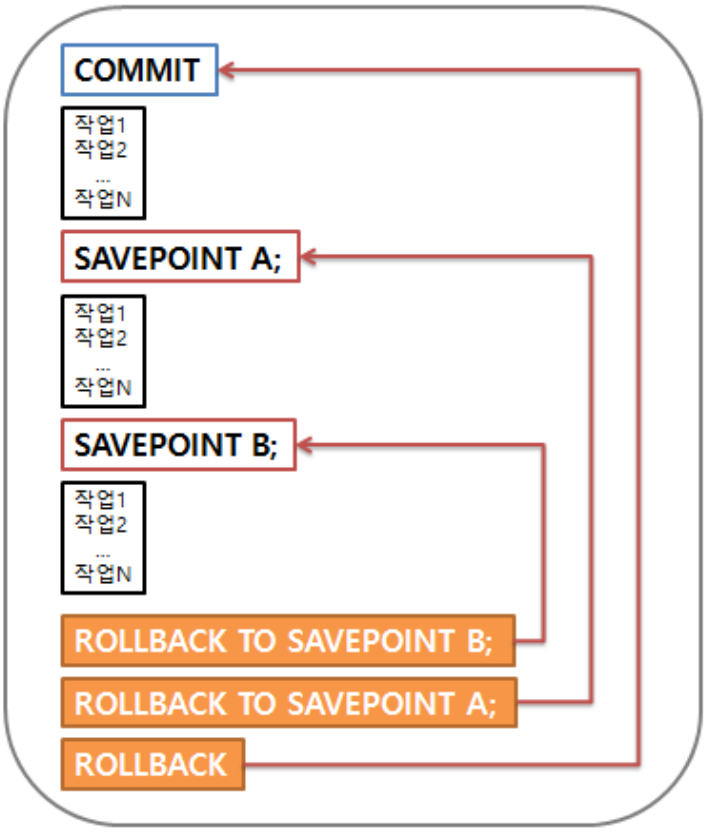
1) ROLLBACK

- 원래의 상태로 복구하기 위한 명령

2) COMMIT

- 작업 결과를 물리적 디스크로 저장하고, 조작 작업이 정상적으로 완료되었음을 관리자에게 알려줌

[작업요약]



■ 실습

 # 실습작업은 위에서부터 아래로 이어집니다. #

[필수]

ROLLBACK 을 사용하기 위해서는
MySQL의 기본 기능인 AUTOCOMMIT 기능을 비활성화 해야한다.

[작업방법]

```
> SELECT @@AUTOCOMMIT;
> SET AUTOCOMMIT=0;          [ 0 = False ,  1 = True ]
```

[실습]

```
mysql> select @@autocommit;
+--------------+
| @@autocommit |
+--------------+
|            1 |
+--------------+
1 row in set (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> select @@autocommit;
+--------------+
| @@autocommit |
+--------------+
|            0 |
+--------------+
1 row in set (0.00 sec)
```

1) ROLLBACK 실습을 위해 새로운 작업과 SAVAPOINT 를 설정하였다.

```
mysql> select * from test;
+--------+-----+---------------+----+
| name   | age | phone         | no |
+--------+-----+---------------+----+
| minki  | 25  | 0000          |  1 |
| hyewon | 23  | 010-9999-9999 |  2 |
+--------+-----+---------------+----+
2 rows in set (0.00 sec)

mysql> insert into test values('youjin', '31', '0000', 3);
Query OK, 1 row affected (0.03 sec)

mysql> savepoint A;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into test values('jinkee', '30', '0001', 4);
Query OK, 1 row affected (0.00 sec)

mysql> savepoint B;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from test;
+--------+-----+---------------+----+
| name   | age | phone         | no |
+--------+-----+---------------+----+
| minki  | 25  | 0000          |  1 |
| hyewon | 23  | 010-9999-9999 |  2 |
| youjin | 31  | 0000          |  3 |
| jinkee | 30  | 0001          |  4 |
+--------+-----+---------------+----+
4 rows in set (0.00 sec)
```

2) SAVEPOINT 를 선정한 후 작업을 더 진행하였다.

```
mysql> select * from test;
+--------+-----+---------------+----+
| name   | age | phone         | no |
+--------+-----+---------------+----+
| minki  | 25  | 0000          |  1 |
| hyewon | 23  | 010-9999-9999 |  2 |
| youjin | 31  | 0000          |  3 |
| jinkee | 30  | 0001          |  4 |
+--------+-----+---------------+----+
4 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql> update test set no=100 where name='jinkee';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from test;
+--------+-----+---------------+-----+
| name   | age | phone         | no  |
+--------+-----+---------------+-----+
| minki  | 25  | 0000          |   1 |
| hyewon | 23  | 010-9999-9999 |   2 |
| youjin | 31  | 0000          |   3 |
| jinkee | 30  | 0001          | 100 |
+--------+-----+---------------+-----+
4 rows in set (0.00 sec)
```

3) ROLLBACK 과정 [SAVEPOINT B -> SAVEPOINT A -> ROLLBACK]

```
mysql> rollback to savepoint B;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from test;
+--------+-----+---------------+----+
| name   | age | phone         | no |
+--------+-----+---------------+----+
| minki  | 25  | 0000          |  1 |
| hyewon | 23  | 010-9999-9999 |  2 |
| youjin | 31  | 0000          |  3 |
| jinkee | 30  | 0001          |  4 |
+--------+-----+---------------+----+
4 rows in set (0.00 sec)

mysql> rollback to savepoint A;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from test;
+--------+-----+---------------+----+
| name   | age | phone         | no |
+--------+-----+---------------+----+
| minki  | 25  | 0000          |  1 |
| hyewon | 23  | 010-9999-9999 |  2 |
| youjin | 31  | 0000          |  3 |
+--------+-----+---------------+----+
3 rows in set (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.06 sec)

mysql> select * from test;
+--------+-----+---------------+----+
| name   | age | phone         | no |
+--------+-----+---------------+----+
| minki  | 25  | 0000          |  1 |
| hyewon | 23  | 010-9999-9999 |  2 |
+--------+-----+---------------+----+
2 rows in set (0.00 sec)
```