

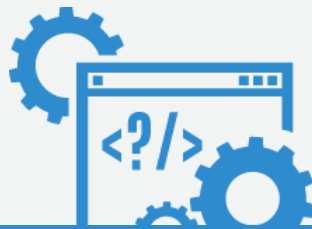
• DB구현 •

트렌젝션 관리



학습내용

- ❖ 트랜잭션의 개념과 사용
- ❖ 잠금의 개념과 사용



학습목표

- ❖ 트랜잭션의 필요성과 특성을 설명할 수 있다.
- ❖ 잠금의 개념에 대해 설명할 수 있다.
- ❖ MySQL에서 트랜잭션 연산과 잠금을 사용할 수 있다.

트랜잭션의 개념과 사용

트랜잭션 개요

➤ 트랜잭션의 필요성

- 데이터베이스에서는 다수의 읽기(Read)/쓰기(Write) 작업이 발생
- 공유된 데이터에 여러 사용자에게 의한 읽기(Read)/쓰기(Write)는 데이터의 일관성과 무결성의 훼손 문제를 발생

➤ 트랜잭션의 정의

트랜잭션이란?

하나의 논리적인 작업을 처리하기 위해 서로 연관되고 연속된 데이터베이스 명령어의 집합

- 목적 : 데이터베이스 명령어 단위가 아니 논리적인 작업 단위로 처리하는 것

➤ 트랜잭션의 계좌이체 예

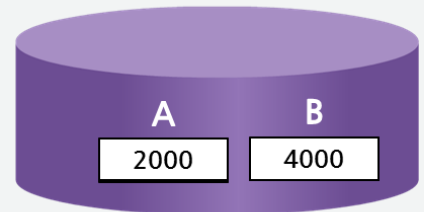
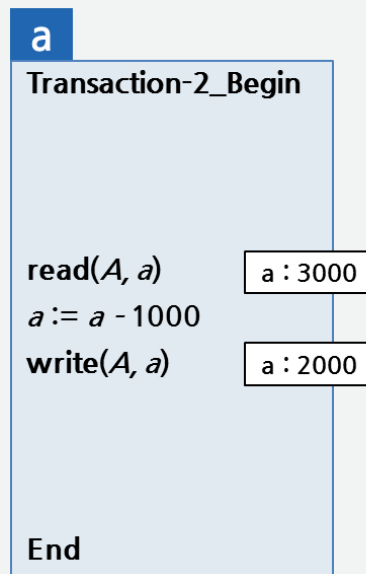
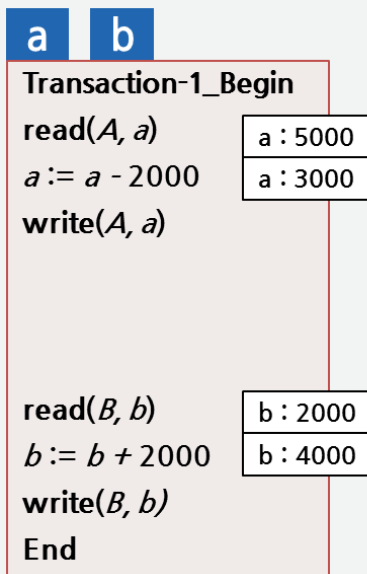
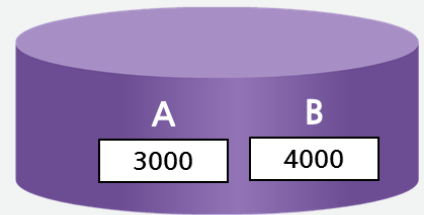
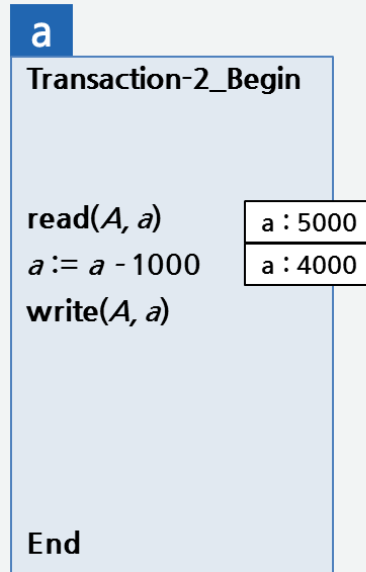
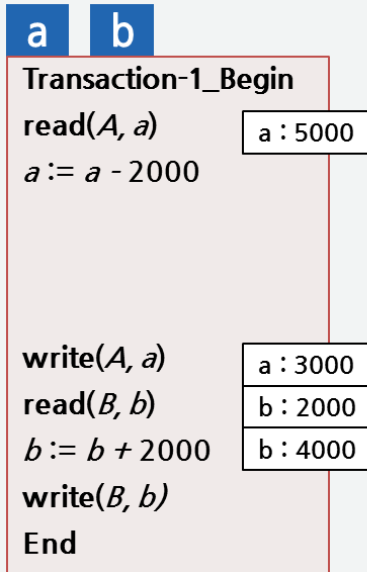
예시

- 홍길동은 A 계좌의 잔고를 조회한다.
 - 홍길동은 A 계좌에서 2000원을 인출한다.
 - 홍길동은 B 계좌의 잔고를 조회한다.
 - 홍길동은 B 계좌에 2000원을 입금한다.
-
- 허균은 A 계좌의 잔고를 조회한다.
 - 허균은 A 계좌에서 1000원을 인출한다.

트랜잭션의 개념과 사용

트랜잭션 개요

▶ 트랜잭션의 계좌이체 예



트랜잭션의 개념과 사용

트랜잭션 개요

▶ 트랜잭션의 4가지 속성(ACID)

원자성 (Atomicity)

- 하나의 트랜잭션은 분리할 수 없는 하나의 단위로, 작업이 모두 수행되거나 하나도 수행되지 않아야 함

일관성 (Consistency)

- 동시에 수행되는 트랜잭션이 없는 상태에서의 트랜잭션은 데이터베이스의 일관성을 유지시켜야 함
- 즉, 트랜잭션 수행 이전에 데이터베이스가 일관성을 지니고 있었다면, 트랜잭션 이후의 데이터베이스 상태도 일관성을 지니고 있어야 함

고립성 (Isolation)

- 다수의 트랜잭션이 동시에 수행되고 있을지라도, 하나의 트랜잭션은 다른 트랜잭션의 영향을 받지 않아야 함

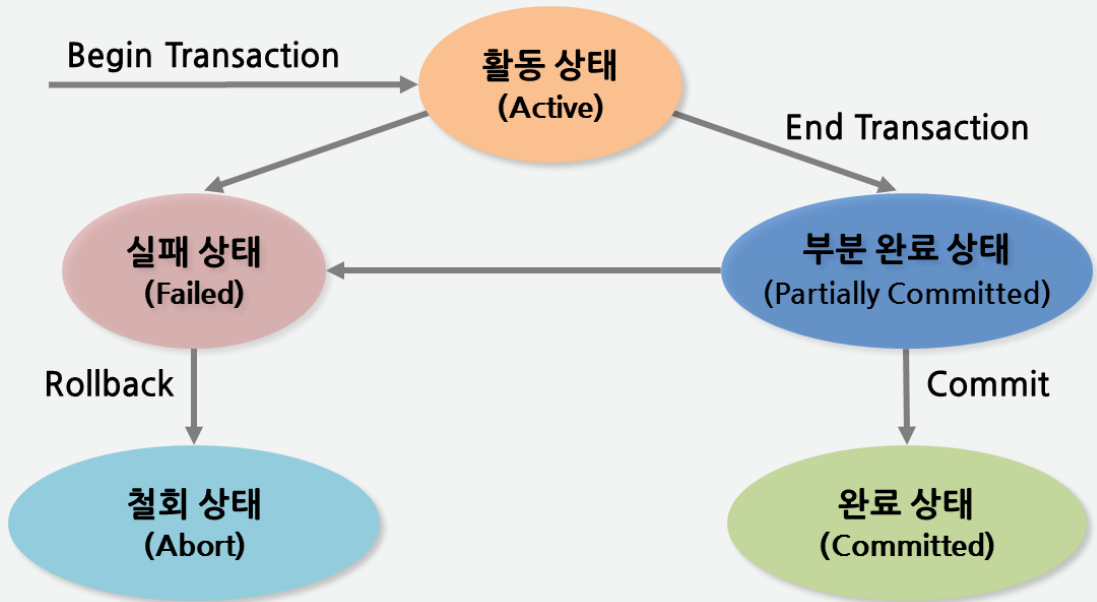
지속성 (Durability)

- 트랜잭션이 정상적으로 수행되었다면, 그 결과는 시스템 오류가 발생하더라도 영구적으로 반영되어야 함

트랜잭션의 개념과 사용

트랜잭션 상태

▶ 트랜잭션 상태 다이어그램



- 활동 상태 : 트랜잭션이 수행을 시작하여 현재 실행되는 상태
- 부분 완료 상태 : 트랜잭션에서 마지막 명령이 실행된 직후의 상태 즉, 트랜잭션의 모든 연산이 처리된 상태
- 실패 상태 : H/W, S/W, 트랜잭션 내부의 오류 등으로 인해 장애가 발생, 트랜잭션의 수행이 중단된 상태
- 완료 상태 : 트랜잭션이 성공적으로 완료된 상태를 의미, 최종 결과를 데이터베이스에 반영하고, 데이터베이스는 다시 일관된 상태로 전환
- 철회 상태 : 트랜잭션 수행이 실패하여 Rollback 연산을 실행한 상태, 수행된 트랜잭션 연산을 모두 취소하고 트랜잭션이 수행되기 전의 상태로 복귀

트랜잭션의 개념과 사용

MySQL에서 트랜잭션의 사용

➤ START(또는 BEGIN) TRANSACTION

- 새로운 트랜잭션을 실행함

➤ COMMIT

- 현재 트랜잭션에서 변경한 내역들을 영구적으로 시스템에 반영

➤ SAVEPOINT identifier

- identifier라는 이름의 저장 시점을 생성함
- ROLLBACK 시 지정된 시점에서의 복귀를 강제로 지정할 수 있음

➤ ROLLBACK [to identifier]

- 현재 트랜잭션에서 변경한 내역들을 모두 취소하여 트랜잭션을 시작하기 이전 상태로 되돌림
- identifier를 지정하여 특정 시점에서의 복귀가 가능함

➤ SET autocommit

- 현재 세션에 대한 'autocommit' 모드를 적용하거나 해제함

※ 'autocommit' 모드

: SQL 문 실행 시 변경 내역이 자동으로 완료되어 시스템에 반영되는 모드

- SET autocommit = 0; (자동 완료 비활성화)
- SET autocommit = 1; (자동 완료 활성화)

트랜잭션의 개념과 사용

MySQL에서 트랜잭션의 사용

▶ 트랜잭션 연산 사용 예제

```
1 START transaction;
2 insert into account values ('2016001', '홍길동', 50000);
3
4 SELECT * FROM account;
```

user_id	user_name	balance
2016001	홍길동	50000
NULL	NULL	NULL

트랜잭션 연산 사용 예제 화면

- 트랜잭션 연산 사용의 예제 모습입니다.

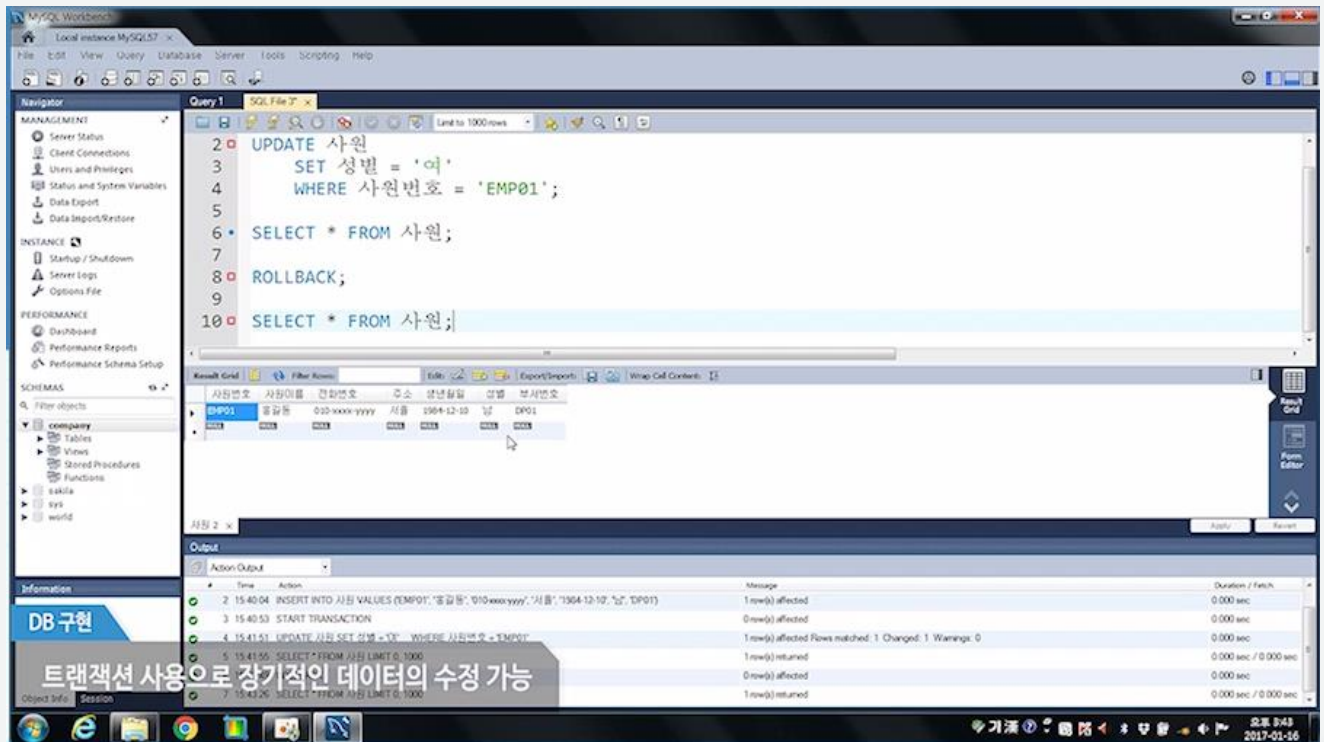
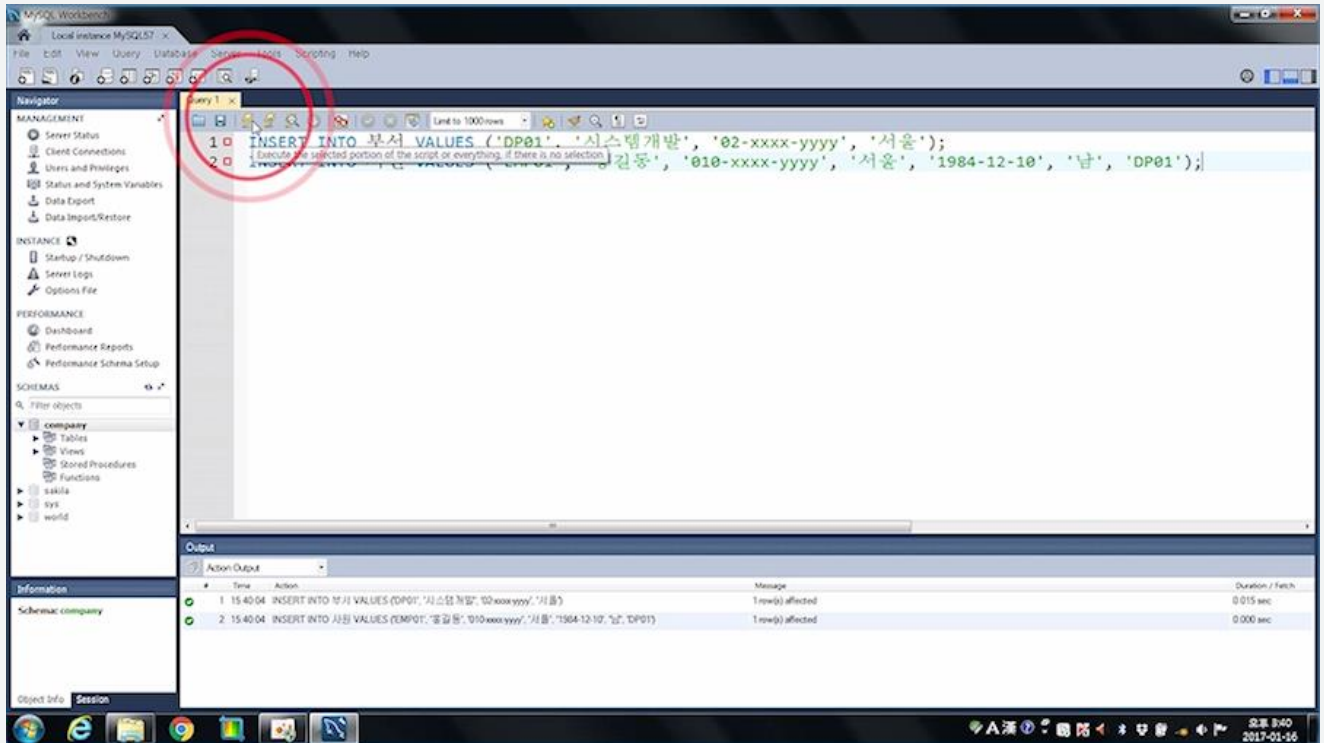
```
1 START transaction;
2 insert into account values ('2016001', '홍길동', 50000);
3
4 SELECT * FROM account;
6 ROLLBACK;
8 SELECT * FROM account;
```

user_id	user_name	balance
2016001	홍길동	50000
NULL	NULL	NULL

트랜잭션 연산 사용 예제 화면

- MySQL에서 트랜잭션에서 트랜잭션 연산을 사용할 경우의 예제화면처럼 확인할 수 있습니다.

트랜잭션의 개념과 사용



잠금의 개념과 사용

잠금의 개요와 종류

➤ 동시성 제어(Concurrency Control)의 의미

- 여러 개의 트랜잭션이 병행 수행되면서 같은 데이터에 접근하여 연산을 실행하더라도, 문제가 발생하지 않고 정확한 수행 결과를 얻을 수 있도록 트랜잭션의 수행을 제어하는 것

➤ 잠금의 개요

잠금(Lock)이란?

트랜잭션 간 테이블 접근을 조정하여, 한 트랜잭션이 다른 트랜잭션에서 접근 중인 테이블에 대한 접근을 방지하는 역할

- 잠금은 특정 자원에 대한 접근을 제한함으로써 변경잡업으로 인한 논리적인 모순이 발생하는 것을 사전에 방지

➤ 읽기 잠금(Read Lock)

- READ 잠금을 획득할 시 테이블에 대한 읽기가 가능하나 트랜잭션에서 쓰기는 불가능
- 복수의 세션에 의해 동시에 취득 가능
- LOCK 구문 사용

구문형식

```
LOCK TABLES <테이블이름> READ
```

잠금의 개념과 사용

잠금의 개요와 종류

➤ 쓰기 잠금(Write Lock)

- WRITE 잠금을 획득할 시 테이블에 대한 읽기/쓰기가 모두 가능
- 오직 하나의 세션만이 해당 테이블에 대한 WRITE 잠금을 획득 가능
- 다른 세션들은 이 잠금이 해제될 때까지 테이블에 접근 불가능
- 기다리는 세션은 테이블에 대한 읽기/쓰기가 모두 불가능
- LOCK 구문 사용

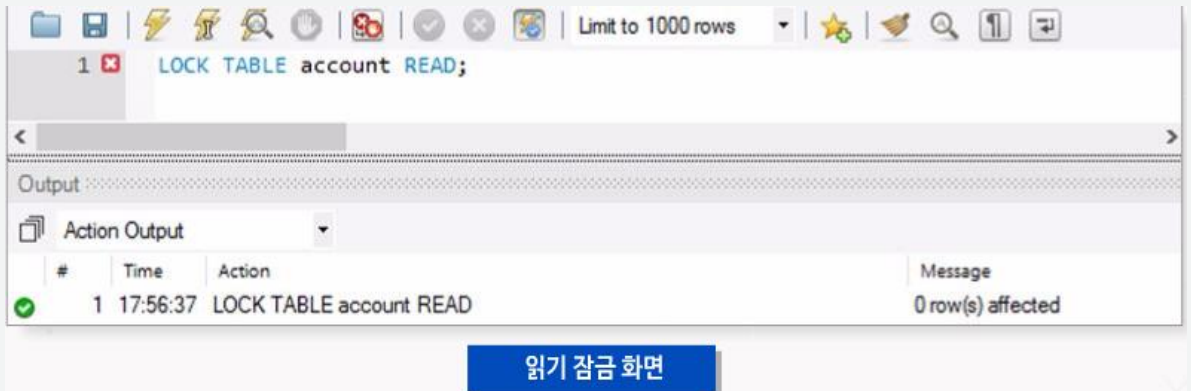
구문형식

```
LOCK TABLES <테이블이름> WRITE
```

🟡 잠금의 개념과 사용

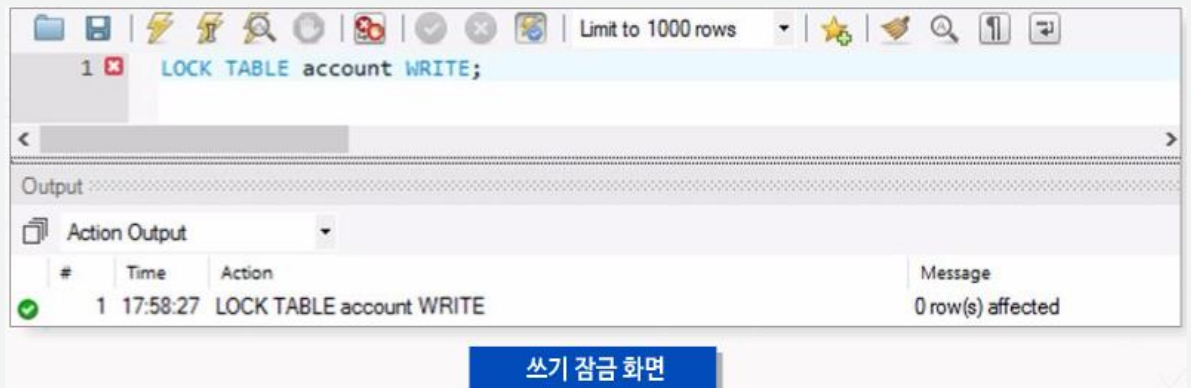
🔗 잠금의 개요와 종류

➤ 읽기 잠금(Read Lock)



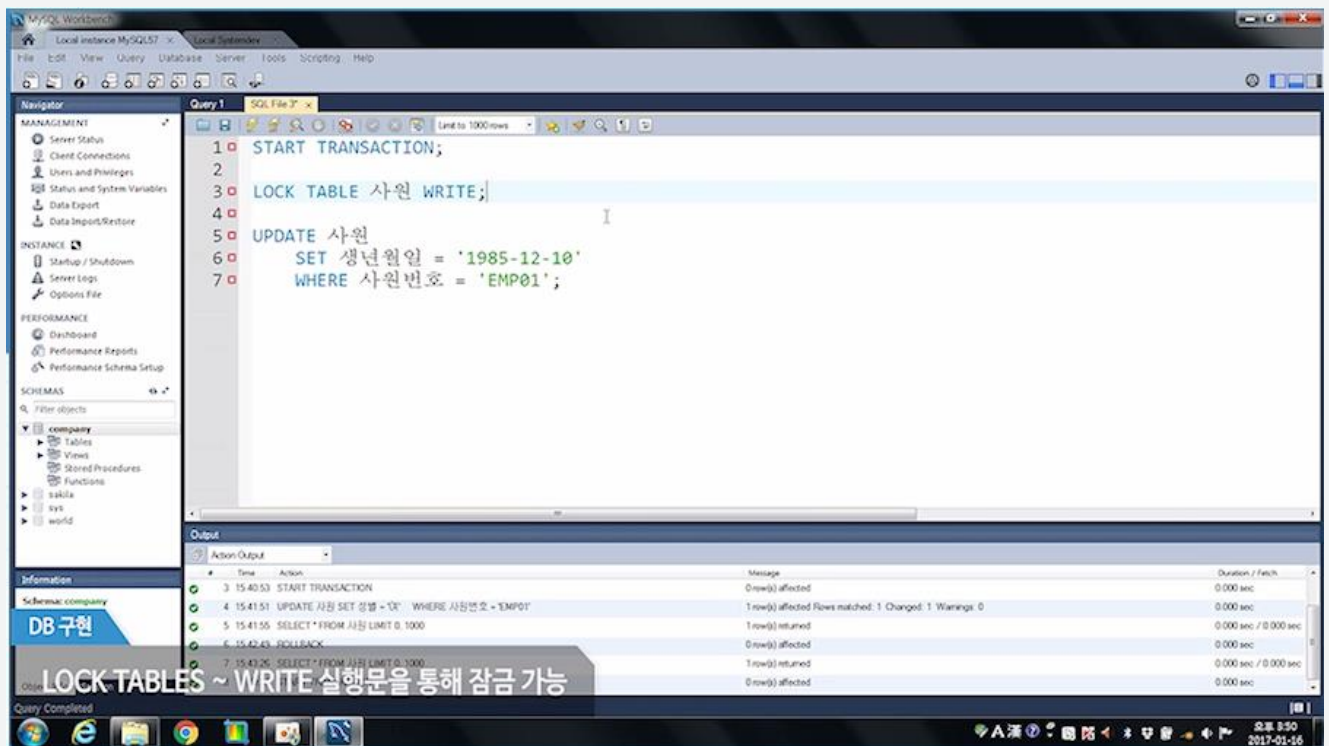
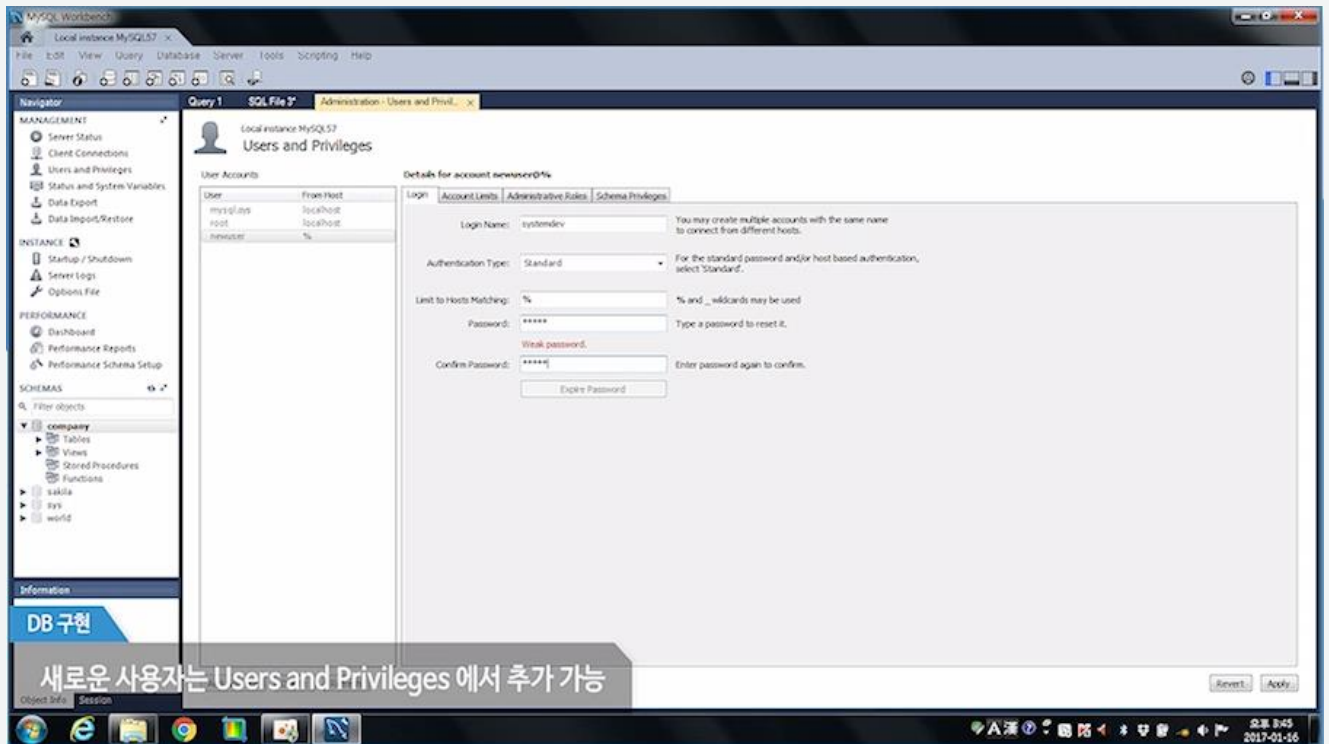
- MySQL에서 잠금은 2가지가 있습니다. 잠금의 사용에서 읽기 잠금을 할 수 있습니다.

➤ 쓰기 잠금(Write Lock)

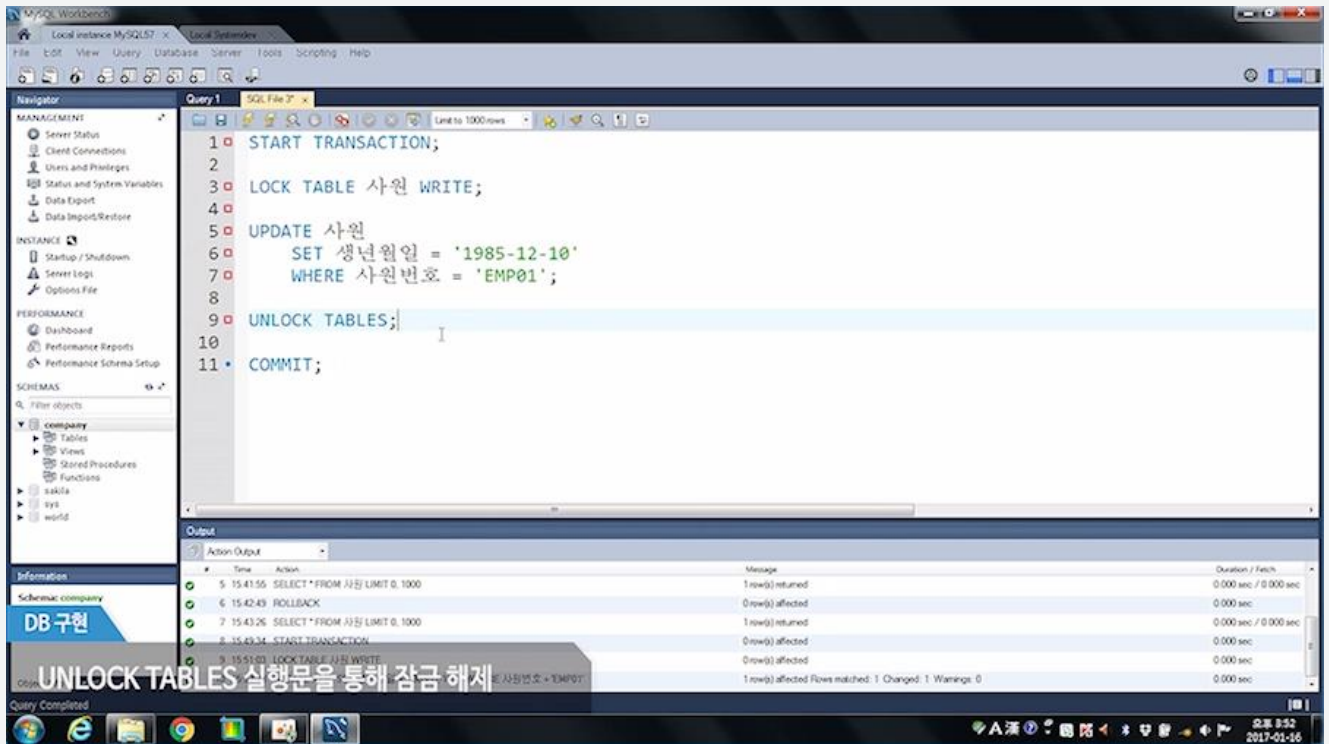


- 잠금의 사용 중 쓰기 잠금의 화면입니다.

잠금의 개념과 사용



잠금의 개념과 사용



The screenshot shows the MySQL Workbench interface. The main editor displays a SQL script for a transaction:

```
1 START TRANSACTION;  
2  
3 LOCK TABLE 사원 WRITE;  
4  
5 UPDATE 사원  
6   SET 생년월일 = '1985-12-10'  
7   WHERE 사원번호 = 'EMP01';  
8  
9 UNLOCK TABLES;  
10  
11 COMMIT;
```

The left sidebar shows the 'SCHEMAS' section with 'company' selected. The bottom status bar indicates 'Query Completed'.

The 'Output' tab at the bottom shows the execution log:

Time	Action	Message	Duration / Fetch
5 15:41:55	SELECT * FROM 사원 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6 15:42:49	ROLLBACK	0 row(s) affected	0.000 sec
7 15:43:26	SELECT * FROM 사원 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8 15:43:34	START TRANSACTION	0 row(s) affected	0.000 sec
9 15:51:03	LOCK TABLE 사원 WRITE	0 row(s) affected	0.000 sec
10 15:51:03	UPDATE 사원 SET 생년월일 = '1985-12-10' WHERE 사원번호 = 'EMP01'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec

Below the output, a text box reads: **UNLOCK TABLES 실행문을 통해 잠금 해제** (Release lock by executing UNLOCK TABLES).



핵심요약

트랜잭션의 개념과 사용

- ❖ 트랜잭션이란?
 - 하나의 논리적인 작업을 처리하기 위해 서로 연관되고 연속된 데이터베이스 명령어의 집합
- ❖ 트랜잭션의 4가지 속성(ACID)
 - 트랜잭션은 원자성, 일관성, 고립성, 지속성 특징을 만족시킴
- ❖ 트랜잭션 상태 다이어그램
 - 트랜잭션은 실행과정에서 활동, 실패, 부분 완료, 철회, 완료 상태 중의 한 상태가 됨
- ❖ MySQL에서 트랜잭션의 사용
 - SQL은 트랜잭션 관리를 위해 START TRANSACTION, END TRANSACTION, COMMIT, ROLLBACK, SAVEPOINT 명령어를 지원함



핵심요약

잠금의 개념과 사용

- ❖ 동시성 제어(Concurrency Control)의 의미
 - 여러 개의 트랜잭션이 병행 수행되면서 같은 데이터에 접근하여 연산을 실행하더라도, 문제가 발생하지 않고 정확한 수행 결과를 얻을 수 있도록 트랜잭션의 수행을 제어하는 것을 동시성 제어라고 함
- ❖ 잠금(Lock)이란?
 - 잠금은 트랜잭션 간 테이블 접근을 조정하여, 한 트랜잭션이 다른 트랜잭션에서 접근 중인 테이블에 대한 접근을 방지하는 역할
- ❖ 잠금의 종류
 - 잠금에는 읽기 잠금과 쓰기 잠금이 있음
 - 읽기 잠금 : 'LOCK TABLES <테이블이름> READ' 으로 수행함
 - 쓰기 잠금 : 'LOCK TABLES <테이블이름> WRITE'으로 수행함