
Chapter 8

트랜잭션



- 트랜잭션은 논리적인 작업의 단위입니다.
- 트랜잭션은 분리되어서는 안 될 작업의 단위입니다.
- 트랜잭션의 시작은 실행 가능한 첫 번째 SQL 문장이 실행 될 때 시작합니다.
- 트랜잭션은 COMMIT 이나 ROLLBACK 문에 의해 명시적으로 종료하거나, DDL이나 DCL 문장 실행으로 자동 커밋되어 종료될 수 있습니다.
- 사용자의 데이터베이스 종료 또는 시스템 충돌(Crash)에 의한 데이터베이스 비정상적 종료에 의해 트랜잭션이 종료되어 변경사항이 취소될 수 있습니다.

트랜잭션을 실행 가능한 SQL 문장이 제일 처음 실행될 때 시작하고, 다음의 경우에 종료합니다.

- COMMIT이나 ROLLBACK 문장이 완료될 때
- CREATE 같은 DDL 문장이 완료될 때
- DCL 문장이 완료될 때
- 사용자가 SQL Plus를 종료할 때
- 머신 실패나 시스템 충돌에 의한 데이터베이스가 비정상적으로 종료될 때

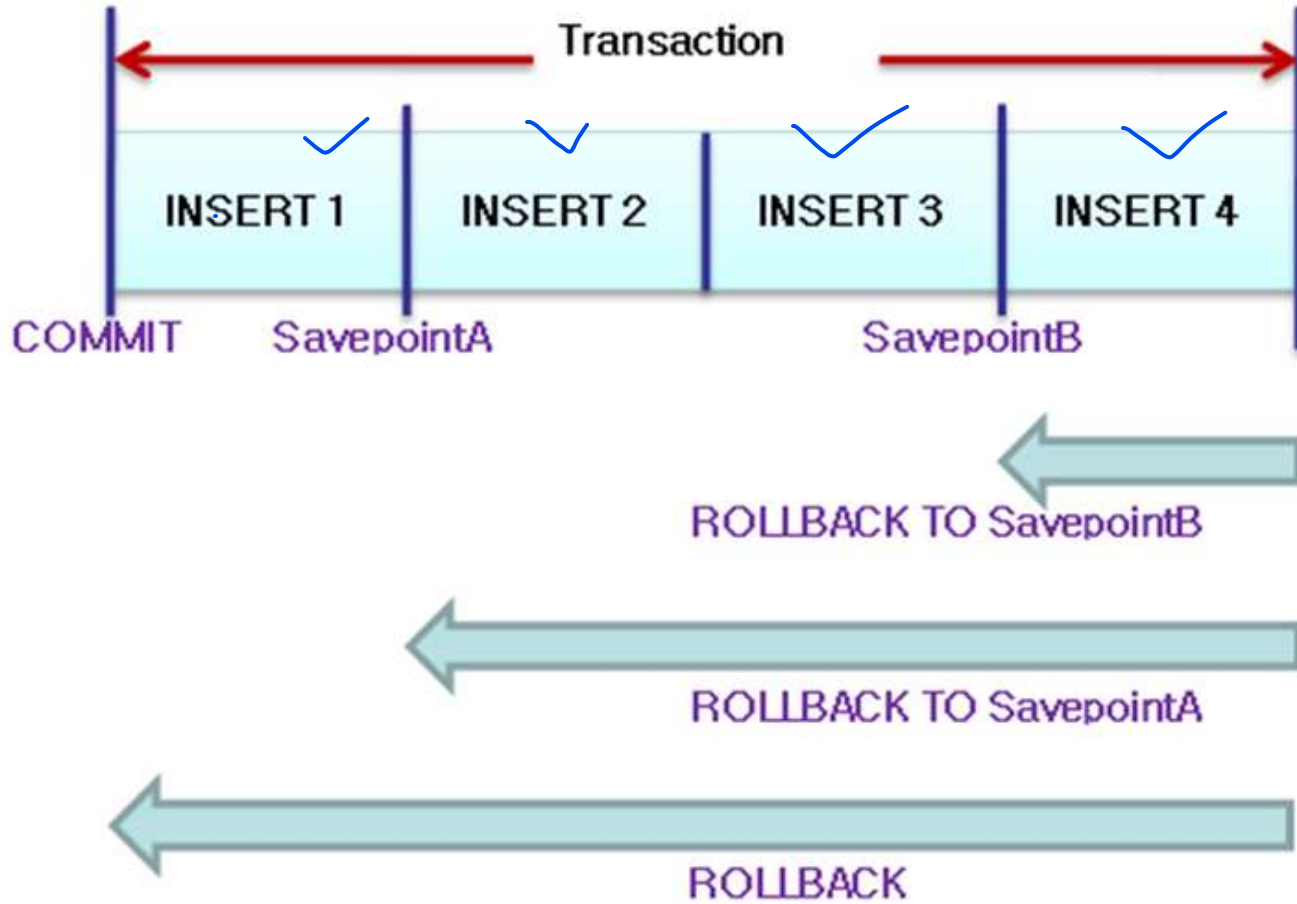
COMMIT과 ROLLBACK을 이용하면 다음 장점을 제공합니다.

- 데이터의 일관성을 제공합니다.
- 데이터를 영구적으로 변경하기 전에 데이터 변경을 미리 보게 합니다.
- 관련된 작업을 논리적으로 그룹화 합니다.
- 하나의 트랜잭션이 끝난 후에 다음의 실행 가능한 SQL 문장은 자동적으로 다음 트랜잭션을 시작할 것입니다.

DDL 문장이나 DCL 문장은 자동으로 커밋되므로 트랜잭션을 암시적으로 종료합니다. COMMIT, SAVEPOINT 그리고 ROLLBACK 문장을 사용하여 트랜잭션의 논리를 제어할 수 있습니다.

표 1. 트랜잭션 구문

문장	설명
COMMIT	모든 미결정 데이터를 영구적으로 변경함으로서 현재 트랜잭션을 종료합니다.
SAVEPOINT <i>savepoint_name</i>	현재 트랜잭션 내에 savepoint를 표시합니다. savepoint 이름은 식별자 규칙을 따라야 하며 처음 32자만 사용됩니다.
ROLLBACK [[TO [SAVEPOINT]] <i>savepoint_name</i>]	ROLLBACK은 모든 미결정 데이터 변경을 버림으로써 현재의 트랜잭션을 종료합니다. ROLLBACK TO <i>savepoint_name</i> 은 savepoint와 모든 연이은 변경을 버립니다.



9.1. 트랜잭션

자동적인 커밋은 다음의 환경에서 발생합니다.

- DDL 문장이 완료 시
- DCL 문장이 완료 시
- 명시적인 COMMIT이나 ROLLBACK 없이 SQL Plus를 정상적으로 종료 할 때(exit)

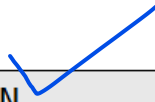
자동적인 롤백은 다음의 환경에서 발생합니다.

- ✓ SQL Plus를 비정상적으로 종료할 때
- ✓ 시스템 실패 시

트랜잭션이 시스템 실패에 의해 중단될 때 모든 트랜잭션은 자동으로 롤백됩니다. 이것은 데이터에 대해서 원하지 않는 변경을 막아 주고 테이블을 마지막으로 커밋된 시점으로 상태를 되돌립니다. SQL은 테이블의 무결성을 보호합니다.

SQL Plus에서 오토커밋(AUTOCOMMIT) 명령은 ON이나 OFF로 토글될 수 있습니다. ON이면 각각의 개별적인 DML 문장이 실행되자마자 커밋됩니다. 그러므로 변경을 롤백할 수 없습니다. OFF로 설정하면 COMMIT은 명시적으로 실행될 수 있습니다. 또한 COMMIT은 DDL 문장이 완료되거나 SQL Plus를 종료할 때 완료됩니다.

다음 구문은 오토커밋 상태를 ON 상태로 변경합니다. 이후 오토커밋 상태를 SHOW 명령으로 확인할 수 있습니다.



```
SQL> SET AUTOCOMMIT ON
SQL> SHOW AUTOCOMMIT
autocommit IMMEDIATE
```

다음 작업을 위해 오토커밋 상태를 OFF로 변경해 두세요.

```
SQL> SET AUTOCOMMIT OFF
SQL> SHOW AUTOCOMMIT
autocommit OFF
```

20번부서의 사원의 정보를 삭제합니다.

```
SQL> DELETE FROM emps WHERE department_id=20;  
2개 행 이(가) 삭제되었습니다.
```

롤백 지점을 생성합니다.

```
SQL> SAVEPOINT delete_20;
```

30번부서의 사원 정보를 삭제합니다.

```
SQL> DELETE FROM emps WHERE department_id=30;  
6개 행 이(가) 삭제되었습니다.
```

30번부서 사원의 정보를 삭제 했던 작업을 취소합니다.

```
SQL> ROLLBACK TO SAVEPOINT delete_20;  
롤백 완료.
```