

TCL

(COMMIT,ROLLBACK,SAVEPOINT)

TCL(Transaction Control Language)

TCL이란?

트랜잭션 제어 언어로써 COMMIT과 ROLLBACK이 있음

트랜잭션이란?

한꺼번에 수행되어야 할 최소의 작업 단위를 말함

- 논리적 작업 단위 (LUW, Logical Units of Work)

하나의 트랜잭션으로 이루어진 작업들은 반드시 한꺼번에 완료되어야 하며,
그렇지 않은 경우에는 한꺼번에 취소되어야 함

ATM 기기의 업무 처리

1. 카드 삽입 -> 2. 메뉴 선택 (인출) -> 3. 금액 확인 및 인증
4. 실제 내부(계좌)에서 금액만큼 인출 -> 5. 실제 현금 인출 -> 6. 완료

TCL(Transaction Control Language)

트랜잭션 SQL

COMMIT : 트랜잭션 작업이 정상 완료 되면 변경 내용을 영구히 저장

(모든 savepoint 삭제)

SAVEPOINT <savepoint명> : 현재 트랜잭션 작업 시점에 이름을 지정함

하나의 트랜잭션 안에서 구역을 나눌수 있음

ROLLBACK : 트랜잭션 작업을 모두 취소하고 최근 commit 시점으로 이동

ROLLBACK TO <savepoint명> : 트랜잭션 작업을 취소하고 savepoint 시점으로 이동

TCL(Transaction Control Language)

트랜잭션 SQL - COMMIT 과 ROLLBACK

```
CREATE TABLE USERTBL(  
    USERNO NUMBER UNIQUE,  
    ID VARCHAR2(20) PRIMARY KEY,  
    PASSWORD CHAR(20) NOT NULL  
);
```

```
INSERT INTO USERTBL VALUES(1,'test1','password1');  
INSERT INTO USERTBL VALUES(2,'test2','password2');  
INSERT INTO USERTBL VALUES(3,'test3','password3');
```

```
COMMIT;
```

```
SELECT * FROM USERTBL;
```

```
INSERT INTO USERTBL VALUES(4,'test4','password4'); --데이터 삽입
```

```
SELECT * FROM USERTBL; -- 삽입된 데이터 확인
```

```
ROLLBACK; -- 롤백으로 취소
```

TCL(Transaction Control Language)

트랜잭션 SQL - savepoint

기존 savepoint가 없다면?

```
INSERT INTO USERTBL VALUES(4,'test4','password4');  
INSERT INTO USERTBL VALUES(5,'test5','password5');
```

```
SELECT * FROM USERTBL;  
ROLLBACK; -- 롤백시 넣었던 데이터 모두가 취소됨
```

중간에 savepoint를 추가 한다면?

```
INSERT INTO USERTBL VALUES(4,'test4','password4');  
SAVEPOINT sp1; -- savepoint 지점 설정  
INSERT INTO USERTBL VALUES(5,'test5','password5');
```

```
SELECT * FROM USERTBL;  
ROLLBACK to sp1; -- sp1 savepoint 지점으로 이동
```

```
SELECT * FROM USERTBL;  
ROLLBACK; -- 가장 마지막 COMMIT 지점으로 이동
```