

TCL





TRANSACTION



TRANSACTION

TRANSACTION

- 한꺼번에 수행되어야 할 최소의 작업 단위
- 하나의 트랜잭션으로 이루어진 작업들은 반드시 한꺼번에 완료되어야 하며, 그렇지 않은 경우 한꺼번에 취소되어야 함

TRANSACTION – ATM 기기 업무처리

1. 카드 삽입
2. 메뉴선택(인출)
3. 금액 확인 및 인증
4. 은행 계좌에서 금액만큼 인출
5. 실제 현금 인출
6. 완료





TCL



TCL

TCL

- 트랜잭션 제어 언어
- 트랜잭션 작업을 적용, 취소 할 때 사용하는 SQL

TCL 종류

구문	설명
COMMIT	트랜잭션 작업이 정상 완료 되면 변경 내용을 저장 → 모든 SAVEPOINT 삭제
ROLLBACK	트랜잭션 작업을 모두 취소하고 최근 COMMIT 시점으로 이동
SAVEPOINT	현재 트랜잭션 작업 시점의 이름을 지정 하나의 트랜잭션 안에서 구역을 나눌 수 있음
ROLLBACK TO	트랜잭션 작업을 취소하고 SAVEPOINT 시점으로 이동



TCL

테스트를 위한 테이블 생성 및 데이터 입력

```
CREATE TABLE USERTBL(  
    USER_NO NUMBER PRIMARY KEY,  
    USER_ID VARCHAR2(20) UNIQUE,  
    USER_PW VARCHAR2(20) NOT NULL  
);
```

```
INSERT INTO USERTBL VALUES (1, 'test1', 'pass1');  
INSERT INTO USERTBL VALUES (2, 'test2', 'pass2');  
INSERT INTO USERTBL VALUES (3, 'test3', 'pass3');  
COMMIT;
```

	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3

TCL

TCL – ROLLBACK

```
INSERT INTO USERTBL VALUES (4, 'test4', 'pass4'); --데이터 추가  
SELECT * FROM USERTBL; --테이블 확인  
ROLLBACK; --트랜잭션작업 취소  
SELECT * FROM USERTBL; --테이블 확인
```

	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3
4	4	test4	pass4



	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3

TCL

TCL – COMMIT

```
INSERT INTO USERTBL VALUES (4, 'test4', 'pass4'); --데이터 추가
SELECT * FROM USERTBL; --테이블 확인
COMMIT; --트랜잭션작업 적용
ROLLBACK; --트랜잭션작업 취소
SELECT * FROM USERTBL; --테이블 확인
```

	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3
4	4	test4	pass4



	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3
4	4	test4	pass4

TCL

TCL – SAVEPOINT

```
INSERT INTO USERTBL VALUES (4, 'test4', 'pass4'); --데이터 추가
SAVEPOINT sp1 --작업 임시저장
INSERT INTO USERTBL VALUES (5, 'test5', 'pass5'); --데이터 추가
SELECT * FROM USERTBL; --테이블 확인
ROLLBACK to sp1; --sp1 시점 적용
SELECT * FROM USERTBL; --테이블 확인
```

	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3
4	4	test4	pass4
5	5	test5	pass5



	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3
4	4	test4	pass4

TCL

TCL – SAVEPOINT

SAVEPOINT 시점에서 ROLLBACK을 한다면?

```
ROLLBACK;  
SELECT * FROM USERTBL;
```

--트랜잭션작업 취소
--테이블 확인

	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3
4	4	test4	pass4



	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3

※ SAVEPOINT까지 간 것은 COMMIT이 아닌 작업 중간상태로 이동만 했을 뿐 트랜잭션작업이 종료된 것은 아님

TCL

TCL – SAVEPOINT

```
INSERT INTO USERTBL VALUES (4, 'test4', 'pass4');  
SAVEPOINT sp1  
INSERT INTO USERTBL VALUES (5, 'test5', 'pass5');  
COMMIT;  
SELECT * FROM USERTBL;  
ROLLBACK to sp1;
```

--데이터 추가
--작업 임시 저장
--데이터 추가
--트랜잭션 작업 저장
--테이블 확인
--sp1 시점 적용

COMMIT;할 때 모든 SAVEPOINT는 삭제되므로 sp1을 찾을 수 없어 에러 발생

※ 테이블에 변경사항이 발생하면
(INSERT,UPDATE,DELETE 수행) 반드시
COMMIT을 해야 함

	USER_NO	USER_ID	USER_PW
1	1	test1	pass1
2	2	test2	pass2
3	3	test3	pass3
4	4	test4	pass4
5	5	test5	pass5