**Transaction Control Language**

SQL, Database

**Transaction Control Language**

SQL, Database

#### 기본 개념

#### Transaction?

- “밀접히 연관되어 분리될 수 없는 한 개 이상의 DB 조작”으로 “DB의 논리적 연산단위”
- 분할할 수 없는 최소 단위 → **All or Nothing**
- 하나의 Transaction에는 하나 이상의 SQL문 포함 → TCL의 대상이 되는 SQL문은 **DML**

### 특성

- **Atomicity** : All or Nothing(성공적 실행 or 그대로 유지)
- **Consistency** : Transaction 실행되기 전 DB 내용 잘못X → 실행 후에도 잘못X
- **Isolation** : Transaction 실행 도중 다른 Transaction 영향X
- **Durability** : Transaction 수행 성공 → DB 내용 영구적 저장

#### TCL

### 효과 (Why use?)

- 데이터 무결성 보장
- 영구적인 변경을 하기 전에 데이터 변경 사항 확인 가능
- 논리적으로 연관된 작업을 그룹 지어 처리 가능

### TCL 수행 이전의 데이터 상태

- 단지 Memory Buffer에만 영향을 받았기 때문에 데이터의 변경 이전 상태로 복구 가능.
- 현재 사용자는 SELECT문으로 결과 확인 가능, 다른 사용자는 현재 사용자가 수행한 명령 결과 확인 불가.
- 변경된 행은 Locking 설정되어 다른 사용자가 변경 불가.

### TCL을 실행하지 않아도 자동 수행 후 Transaction이 종료되는 경우

- **DDL**문 실행하면 자동 **COMMIT**. (☞ 당연한 이야기지만, DML문 → DDL문 실행하면 DML에서 COMMIT 없이 자동 COMMIT.)
- DB **정상 종료**하면 자동 **COMMIT**.
- Application **이상 종료**로 DB(Instance) 접속 단절되는 경우 자동 **ROLLBACK**.

**SQL Server에서의 3가지 Transaction 처리 방식**

- AUTO COMMIT : SQL Server의 기본 방식. DML/DDL을 실행할 때마다 **DBMS**가 **자동**으로 Transaction 제어하는 방식. (성공=Commit or 실패=Rollback 자동 수행.)
- 암시적 Transaction : Transaction **시작**은 **DBMS**가, **끝**은 **사용자가** 명시적으로 **COMMIT** / **ROLLBACK** 처리하는 방식. (Oracle과 동일)
- 명시적 Transaction : Transaction **시작, 끝** 모두 **사용자**가 명시적으로 지정하는 방식

**문법**

**COMMIT**

**"변경된 Data를 Table에 영구적으로 반영해라"**

```
DML문

COMMIT; --SQL Server는 Auto Commit이라 필요X
```

**ROLLBACK**

**"변경된 Data가 문제 있으니 변경 전 데이터로 복귀시켜라"**

```
DML문

ROLLBACK;
```

**SAVEPOINT**

**"Data 변경을 사전에 지정한 저장점까지만 ROLLBACK하라"**

- 현 시점에서 저장점(Save Point)까지 전체 Transaction의 일부만 ROLLBACK 할 경우 사용 (= 복잡한 대규모 Transaction일 때 적합!)
- 동일 이름 지정시 **가장 최근 저장점**이 유효.
- 특정 저장점에서 ROLLBACK 수행 **이후** 저장점은 **무효**.
- 저장점 없이 ROLLBACK 수행 후에는 Transaction 시작 위치로 되돌아간다. (당연한 이야기!)

```
SAVEPOINT SVPT1;

DML문
```

```
ROLLBACK TO SVPT1;


/* SQL Server
SAVE TRANSACTION SVPT1; -- TRANSACTION 대신 TRAN 대체 가능

DML문

ROLLBACK TRANSACTION SVPT1; -- TRANSACTION 대신 TRAN 대체 가능
*/
```

From:
https://ledyx.xyz/ - **Ledyx WIKI**

Permanent link:
**https://ledyx.xyz/doku.php?id=transaction_control_language&rev=1442069437**

Last update: **2026/10/24 07:03**