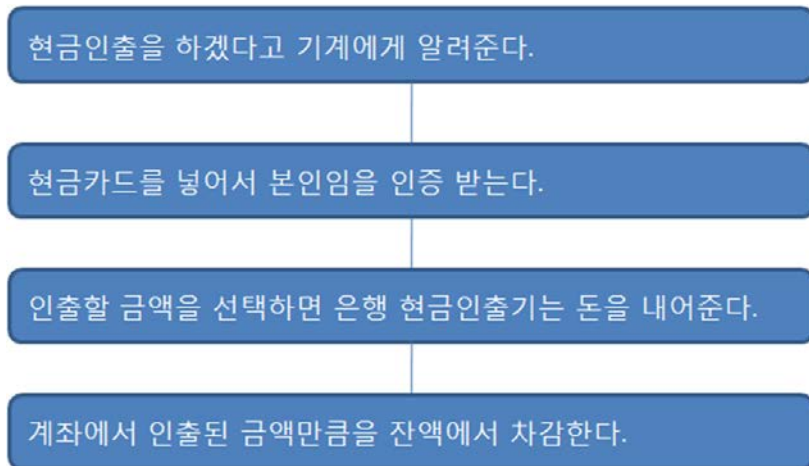


11 트랜잭션 관리

11.1. 트랜잭션

- 오라클에서 발생하는 여러 개의 SQL 명령문들을 하나의 논리적인 작업 단위로 처리하는데 이를 트랜잭션이라고 한다.
- 하나의 트랜잭션은 All-OR-Nothing 방식으로 처리된다. 즉, 여러 개의 명령어의 집합이 정상적으로 처리되면 정상 종료하도록 하고 여러 개의 명령어 중에서 하나의 명령어라도 잘못되었다면 전체를 취소해버린다.
- 데이터베이스에서 작업의 단위로 트랜잭션이란 개념을 도입한 이유는 데이터의 일관성을 유지하면서 안정적으로 데이터를 복구시키기 위해서이다.
- 예를들어 은행 현금인출기(ATM)에서 돈을 인출하는 과정으로 트랜잭션을 설명해 보면..

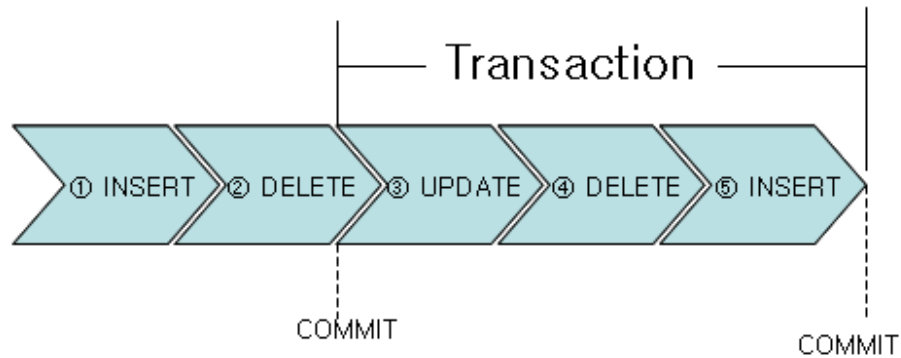


- 트랜잭션 제어를 위한 명령어(Transaction Control Language) : COMMIT, SAVEPOINT, ROLLBACK

11.2. COMMIT과 ROLLBACK

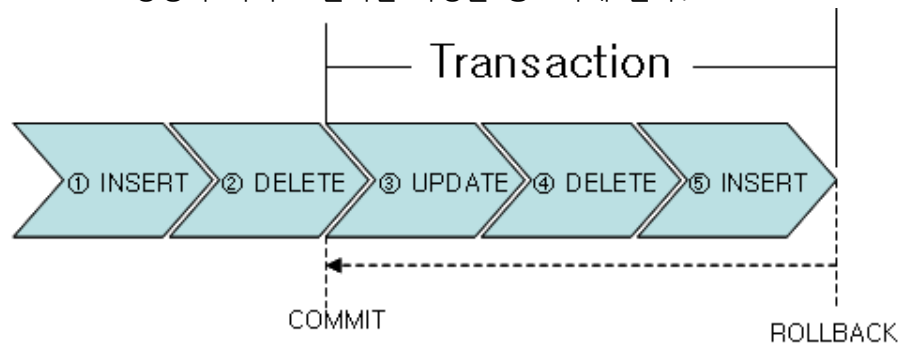
11.2.1. COMMIT과 ROLLBACK의 개념

- DML 작업이 성공적으로 처리되도록 하기 위해서는 COMMIT 명령을, 작업을 취소하기 위해서는 ROLLBACK 명령으로 종료해야 한다.
- COMMIT
 - COMMIT은 모든 작업들을 정상적으로 처리하겠다고 확정하는 명령어로 트랜잭션의 처리 과정을 데이터베이스에 모두 반영하기 위해서 변경된 내용을 모두 영구 저장한다.
 - COMMIT 명령어를 수행하게 되면 하나의 트랜잭션 과정을 종료하게 된다.



■ ROLLBACK

- ROLLBACK은 작업 중 문제가 발생되어서 트랜잭션의 처리 과정에서 발생한 변경사항을 취소하는 명령어이다. 즉, 트랜잭션으로 인한 하나의 묶음 처리가 시작되기 이전의 상태로 되돌린다.
- ROLLBACK 명령어 역시 트랜잭션 과정을 종료하게 된다.



(1) COMMIT 명령어와 ROLLBACK 명령어의 장점

- 데이터 무결성이 보장된다.
- 영구적인 변경 전에 데이터의 변경 사항을 확인할 수 있다.
- 논리적으로 연관된 작업을 그룹화할 수 있다.

(2) COMMIT 명령어

- Transaction(INSERT, UPDATE, DELETE) 작업 내용을 실제 DB에 저장한다.
- 이전 데이터가 완전히 UPDATE 된다.
- 모든 사용자가 변경된 데이터의 결과를 볼 수 있다.

(3) ROLLBACK 명령어

- Transaction 작업 내용을 취소한다.
- 이전 COMMIT한 곳 까지만 복구한다.

(4) 자동 COMMIT 명령과 자동 ROLLBACK 명령이 되는 경우

- SQL* PLUS가 정상 종료되었다면 자동으로 COMMIT되지만, 비정상 종료되었다면 자동으로 ROLLBACK 한다.
- DDL과 DCL(Data Control Language, User 관리) 명령문이 수행된 경우 자동으로 COMMIT 된다.
- 정전이 발생했거나 컴퓨터 Down시(컴퓨터의 전원이 끊긴) 자동으로 ROLLBACK 된다.

11.3. 자동 커밋

- DDL문에는 CREATE, ALTER, DROP, RENAME, TRUNCATE 등이 있다. 이러한 DDL문은 자동으로 커밋(AUTO COMMIT)을 실행한다.

[실습] CREATE문에 의한 자동 커밋

- 이전에 수행했던 DML 명령어가 CREATE문에 의해 자동 커밋되는지 확인해 본다.

```
-- CREATE문에 의한 자동 커밋
DROP TABLE DEPT02;
CREATE TABLE DEPT02
AS
SELECT * FROM DEPT;
SELECT * FROM DEPT02;

DELETE FROM DEPT02
WHERE DEPTNO=40;

CREATE TABLE DEPT03
AS
SELECT * FROM DEPT;

ROLLBACK; -- ROLLBACK 명령을 수행하였지만, DELETE된 정보가 되살아나지 않는 이유는 CREATE 명령문이 수행되면서
자동으로 커밋이 발생되었기 때문이다.

SELECT * FROM DEPT02;
```

[실습] TRUNCATE문의 실패에 의한 자동 커밋

- TRUNCATE문이 실패했더라도 이전에 수행했던 DML 명령어가 자동으로 커밋되는 것을 확인해 본다.

```
DROP TABLE DEPT03;
CREATE TABLE DEPT03
AS
SELECT * FROM DEPT;

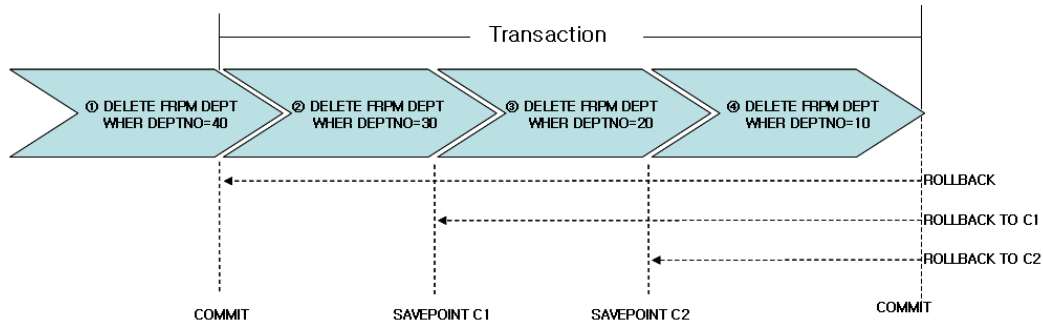
SELECT * FROM DEPT03;
DELETE FROM DEPT03
WHERE DEPTNO=20;

TRUNCATE TABLE DEPTPPP;
ROLLBACK;

SELECT * FROM DEPT03;
```

11.4. 트랜잭션을 작게 분할하는 SAVEPOINT

- SAVEPOINT 명령을 써서 현재의 트랜잭션을 작게 분할할 수 있다.
- 저장된 SAVEPOINT는 ROLLBACK TO SAVEPOINT 문을 사용하여 표시한 곳까지 ROLLBACK할 수 있다.
- 여러 개의 SQL 문의 실행을 수반하는 트랜잭션의 경우, 사용자가 트랜잭션 중간 단계에서 세이브포인트를 지정할 수 있다.
- 이 세이브포인트는 차후 롤백과 함께 사용해서 현재 트랜잭션 내의 특정 세이브포인트까지 롤백할 수 있게 됩니다.



```
-- 형식
SAVEPOINT LABEL_NAME; -- 특정 위치 지정
ROLLBACK TO LABEL_NAME; -- 특정 위치로 되돌아가기

-- 예 : 트랜잭션 중간 단계에서 세이브포인트 지정하기
DROP TABLE DEPT01;
CREATE TABLE DEPT01
AS
SELECT * FROM DEPT;

DELETE FROM DEPT01 WHERE DEPTNO=40;
COMMIT;
SELECT * FROM DEPT01;

DELETE FROM DEPT01
WHERE DEPTNO=30;

SAVEPOINT C1;
DELETE FROM DEPT01
WHERE DEPTNO=20;
SELECT * FROM DEPT01;

SAVEPOINT C2;
DELETE FROM DEPT01
WHERE DEPTNO=10;
SELECT * FROM DEPT01;

ROLLBACK TO C2;
SELECT * FROM DEPT01;

ROLLBACK TO C1;
SELECT * FROM DEPT01;

ROLLBACK;
SELECT * FROM DEPT01;
```