Transaction

1. Transaction

- 데이터베이스에서 Transaction(Transaction)은 한번에 수행되어야 하는 작업 처리의 단위
- 데이터베이스에서 발생하는 여러 개의 SQL 명령문들을 하나의 논리적인 작업 단위로 처리하는데이를 Transaction
- 하나의 Transaction은 All-or-Nothing 방식으로 처리
- 여러 개의 명령어의 집합이 정상적으로 처리되면 정상 종료하도록 하고 여러 개의 명령어 중에서 하나의 명령어라도 잘못되었다면 전체를 취소
- 데이터베이스에서 작업의 단위로 Transaction이란 개념을 도입한 이유는 데이터의 일관성을 유지하면서 안정적으로 데이터를 복구시키기 위해서

1. Transaction

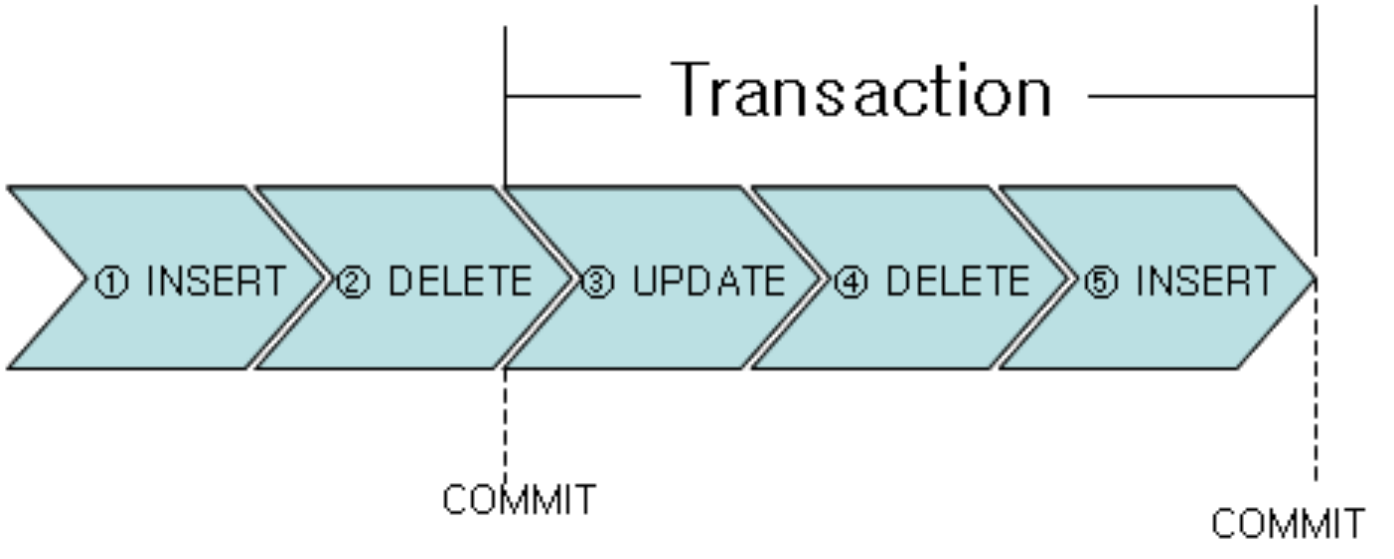
- Transaction 제어를 위한 명령어(Transaction Control Language)
  - ✓ COMMIT
  - ✓ SAVEPOINT
  - ✓ ROLLBACK

2. COMMIT과 ROLLBACK

- 데이터를 조작하는 명령어인 DML(Data Manipulation Language)은 이들이 실행됨과 동시에 Transaction이 시작
- DML 작업이 성공적으로 처리되도록 하기 위해서는 COMMIT 명령을 작업을 취소하기 위해서는 ROLLBACK 명령으로 종료
- COMMIT
  - ✓ COMMIT은 모든 작업들을 정상적으로 처리하겠다고 확정하는 명령어로 Transaction의 처리 과정을 데이터베이스에 모두 반영하기 위해서 변경된 내용을 모두 영구 저장
  - ✓ COMMIT 명령어를 수행하게 되면 하나의 Transaction 과정을 종료
- ROLLBACK
  - ✓ ROLLBACK은 작업 중 문제가 발생되어서 Transaction의 처리 과정에서 발생한 변경사항을 취소하는 명령어
  - ✓ ROLLBACK 명령어 역시 Transaction 과정을 종료
  - ✓ ROLLBACK은 Transaction으로 인한 하나의 묶음 처리가 시작되기 이전의 상태로 되돌림
  - ✓ Transaction은 여러 개의 물리적인 작업(DML 명령어)들이 모여서 이루어지는데 이러한 과정에서 하나의 물리적인 작업이라도 문제가 발생하게 되면 모든 작업을 취소해야 하므로 이들을 하나의 논리적인 작업 단위(Transaction)로 구성
  - ✓ 문제가 발생하게 되면 이 논리적인 작업 단위를 취소하면 됨

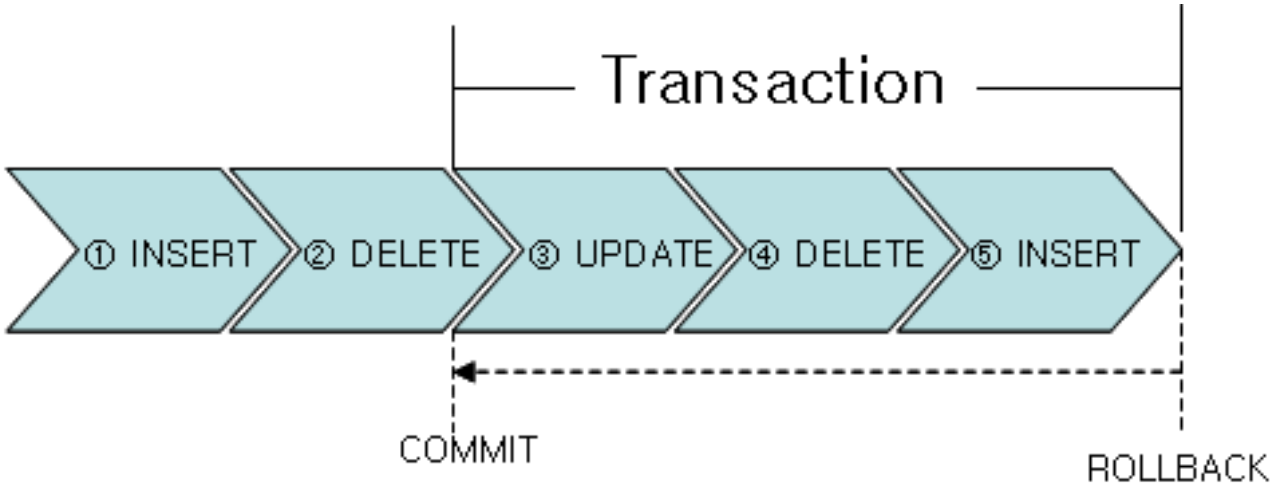
2. COMMIT과 ROLLBACK

- 아래 그림에서 UPDATE 문으로 데이터를 갱신하고(③) DELETE 문으로 데이터를 삭제하고(④) INSERT 문을 사용해 데이터를 삽입(⑤)
- 만약 이 모든 과정이 오류 없이 수행되었다면 지금까지 실행한 모든 작업(③, ④, ⑤)을 "데이터베이스에 영구 저장하라"는 명령으로 커밋을 수행



2. COMMIT과 ROLLBACK

- Rollback 명령은 마지막으로 수행한 커밋 명령까지만 정상 처리(①, ②)된 상태로 유지하고 그 이후에 수행했던 모든 DML 명령어 작업(③, ④, ⑤)들을 취소시켜 이전 상태로 원상 복귀



- Transaction은 이렇듯 All-OR-Nothing 방식으로 DML 명령어들을 처리

2. COMMIT과 ROLLBACK

- COMMIT 명령어과 ROLLBACK 명령어의 장점
  - ✓ 데이터 무결성이 보장
  - ✓ 영구적인 변경 전에 데이터의 변경 사항을 확인할 수 있음
  - ✓ 논리적으로 연관된 작업을 그룹화
- COMMIT 명령어
  - ✓ Transaction(INSERT, UPDATE, DELETE) 작업 내용을 실제 DB에 저장
  - ✓ 이전 데이터가 완전히 UPDATE
  - ✓ 모든 사용자가 변경된 데이터의 결과를 확인할 수 있음
- ROLLBACK 명령어
  - ✓ Transaction(INSERT, UPDATE, DELETE) 작업 내용을 취소
  - ✓ 이전 COMMIT한 곳 까지만 복구
- 데이터베이스 사용자가 COMMIT이나 ROLLBACK 명령어를 명시적으로 수행시키지 않더라도 다음과 같은 경우에 자동 커밋 혹은 자동 Rollback이 발생
  - ✓ SQL* PLUS가 정상 종료되었다면 자동으로 COMMIT되지만, 비정상 종료되었다면 자동으로 ROLLBACK
  - ✓ DDL과 DCL 명령문이 수행된 경우 자동으로 COMMIT
  - ✓ 정전이 발생했거나 컴퓨터 Down시(컴퓨터의 전원이 끊긴) 자동으로 ROLLBACK

실습하기

- 부서번호가 10번인 부서에 대해서만 삭제하려고 했는데 테이블 내의 모든 데이터가 삭제되어 아무런 데이터도 찾을 수 없게 되었더라도 ROLLBACK 문을 사용하여 이전 상태로 되돌릴 수 있음

```
DELETE FROM DEPT01;

SELECT *
FROM DEPT01;

ROLLBACK;

SELECT *
FROM DEPT01;
```

실습하기

❖ 원래하려고 했던 부서번호가 20번인 부서만 삭제해 봅시다.

1. 부서번호 20번 사원에 대한 정보만 삭제한 후 확인

```
DELETE FROM DEPT01
WHERE DEPTNO=20;

SELECT *
FROM DEPT01;
```

2. 데이터를 삭제한 결과를 물리적으로 영구히 저장하기 위해서 커밋을 수행

```
COMMIT;

SELECT *
FROM DEPT01;
```

3. 자동 커밋

- DDL 문에는 CREATE, ALTER, DROP, RENAME, TRUNCATE 등이 있음
- DDL문은 자동으로 커밋(AUTO COMMIT)이 발생

실습하기

❖ CREATE문에 의한 자동 커밋에 의해서 이전에 수행했던 DML 명령어가 자동 커밋

1. 부서 번호가 40번인 부서를 삭제

```
DELETE FROM DEPTO2
WHERE DEPTNO=40;
```

2. 삭제 후 부서 테이블(DEPT)과 동일한 내용을 갖는 새로운 테이블(DEPT03)을 생성

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

3. DEPT02 테이블의 부서번호가 40번인 부서를 다시 되살리기 위해서 ROLLBACK 명령문을 수행하여도 이미 수행한 CREATE 문 때문에 자동으로 커밋이 발생하였으므로 되살릴 수 없음

```
ROLLBACK;

SELECT *
FROM DEPT02;
```

실습하기

❖ TRUNCATE 문이 실패 되더라도 자동 커밋되어 이전에 수행했던 DML 명령어가 자동 커밋

1. 부서 테이블(DEPT03)에서 부서 번호가 20번인 부서를 삭제

```
DELETE FROM DEPT03
WHERE DEPTNO=20;
```

2. TRUNCATE 문을 실행시키되 테이블 명을 일부러 잘못 적어서 에러를 유도

```
TRUNCATE TABLE DEPTPPP;
```

3. 부서번호가 20번인 부서를 다시 되살리기 위해서 ROLLBACK 명령문을 수행하여도 TRUNCATE 문이 수행되면서 자동으로 커밋이 발생하였으므로 되살릴 수 없음

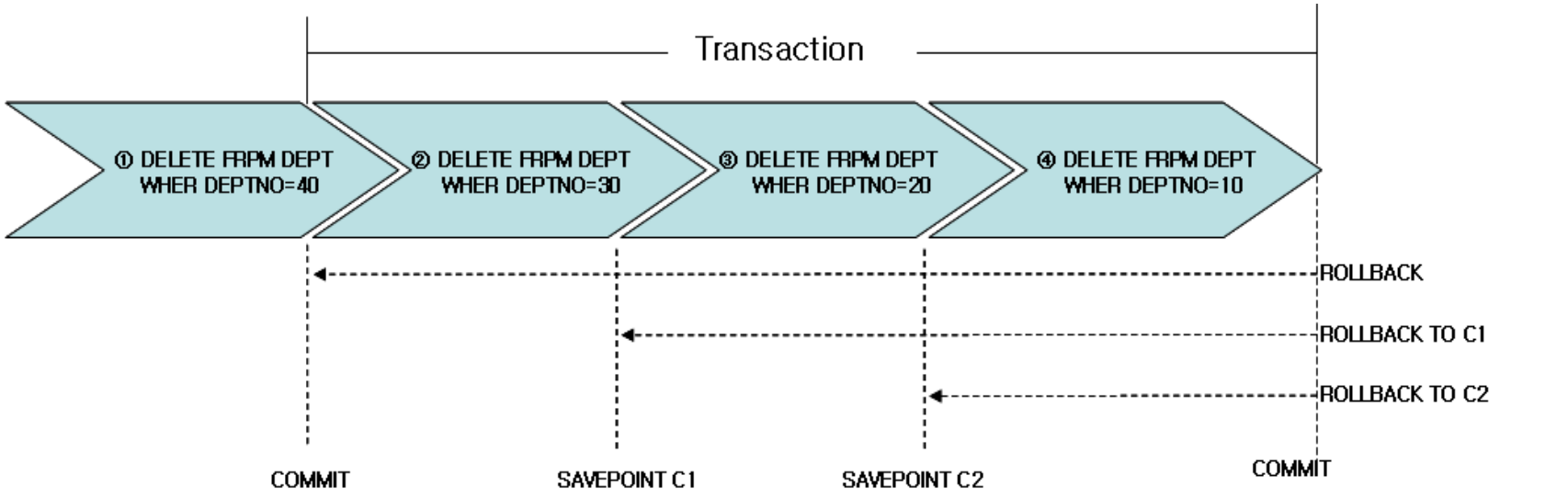
```
ROLLBACK;

SELECT *
FROM DEPT03;
```

4. SAVEPOINT

- SAVEPOINT 명령을 써서 현재의 Transaction을 작게 분할
- 저장된 SAVEPOINT는 ROLLBACK TO SAVEPOINT 문을 사용하여 표시한 곳까지 ROLLBACK할 수 있음
- 여러 개의 SQL 문의 실행을 수반하는 Transaction의 경우 사용자가 Transaction 중간 단계에서 세이브포인트를 지정할 수 있음
- Savepoint는 차후 Rollback과 함께 사용해서 현재 Transaction 내의 특정 세이브포인트까지 Rollback할 수 있음

SAVEPOINT

- COMMIT 명령이 내려진 후 다음 COMMIT 명령이 나타날 때까지가 하나의 Transaction으로 구성되므로 ②번에서 ④번까지가 하나의 Transaction
- 이렇게 트랙잭션을 구성할 때 중간 중간 SAVEPOINT 명령으로 위치를 지정해 놓으면(예를 들어 C) 하나의 Transaction 내에서도 ROLLBACK TO C(SAVEPOINT 문을 사용하여 표시한 곳)까지 ROLLBACK할 수 있음



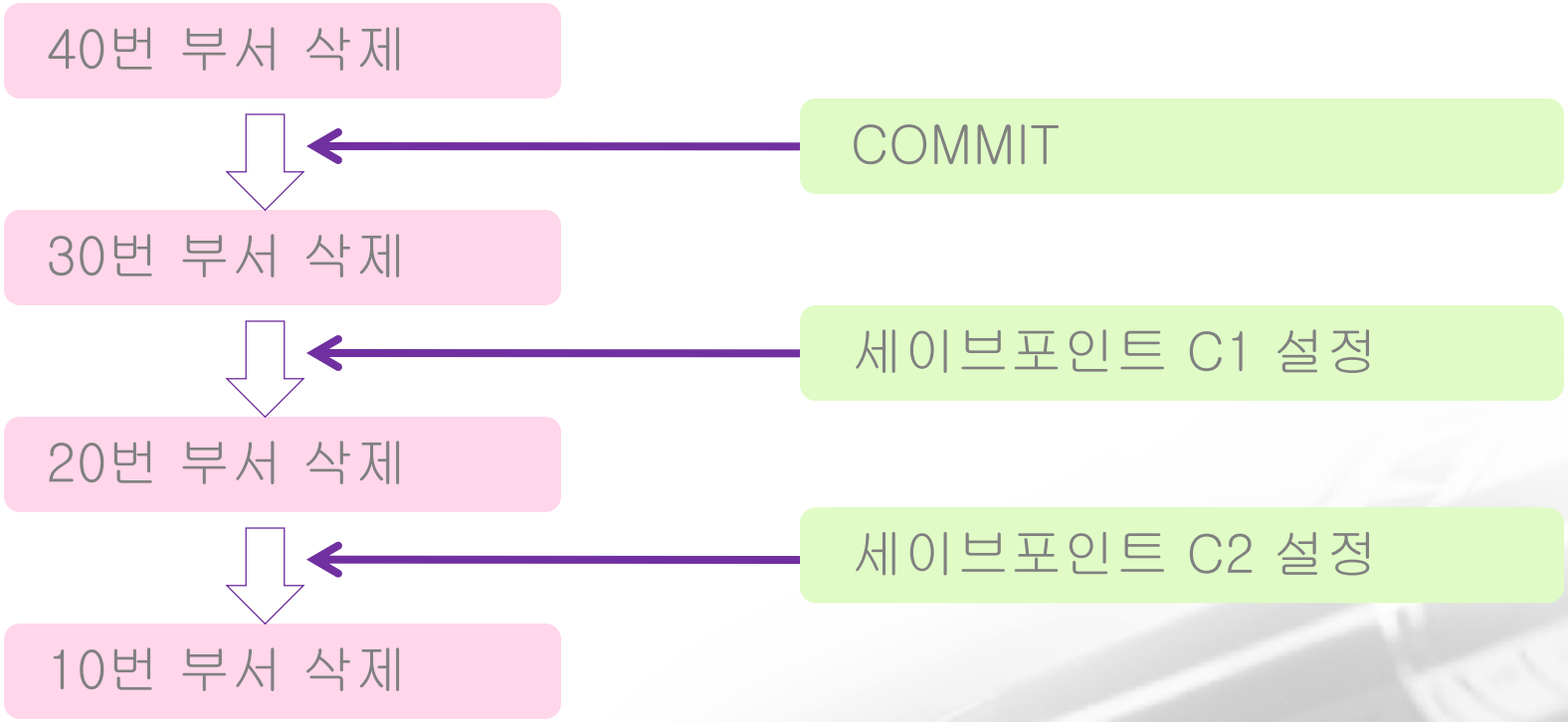
SAVEPOINT

- SAVEPOINT로 특정 위치를 지정하기 위한 형식
  SAVEPOINT *LABEL_NAME;*

- SAVEPOINT로 지정해 놓은 특정 위치로 되돌아가기 위한 형식
  ROLLBACK TO *LABEL_NAME*;

<실습하기>

❖ Transaction 중간 단계에서 세이브포인트를 지정



<실습하기>

1. 부서번호가 40번인 부서를 삭제한 후에 커밋을 수행하여 새롭게 Transaction을 시작

```
DELETE FROM DEPT01
WHERE DEPTNO=40;

COMMIT;
```

2. 이번엔 부서번호가 30번인 부서를 삭제합니다.

```
DELETE FROM DEPT01
WHERE DEPTNO=30;
```

3. 세이브포인트 C1를 설정한 후, 부서번호가 20번인 사원을 삭제합니다.

```
SAVEPOINT C1;

DELETE FROM DEPT01
WHERE DEPTNO =20;
```

4. 세이브포인트 C2를 설정한 후, 부서번호가 10번인 사원을 삭제합니다.

```
SAVEPOINT C2;

DELETE FROM DEPT01
WHERE DEPTNO =10;
```

실습하기

1. 지금 ROLLBACK 명령을 내리게 된다면 이전 COMMIT 지점으로 되돌아가므로 10, 20, 30번 부서의 삭제가 모두 취소되는데 10번 부서의 삭제 이전까지만 되돌리려면 다시 30, 20번의 부서를 삭제해야 함

2. Savepoint C2 지점으로 이동되어 10번 부서의 삭제 이전으로 되돌려진 것을 확인

```
ROLLBACK TO C2;

SELECT *
FROM DEPT01;
```

3. 마지막으로 이전 Transaction까지 Rollback 한 후의 결과 확인

```
ROLLBACK;

SELECT *
FROM DEPT01;
```