MySQL Database

DML – INSERT, UPDATE, DELETE

Data Manipulation Language

- 종류
  - Add New Row(s)
    - INSERT INTO 테이블명 [(컬럼 리스트)] VALUES (값 리스트);
  - Modify Exsiting Row(s)
    - UPDATE 테이블명 SET 변경내용 [WHERE 조건];
  - Remove Existing Row(s)
    - DELETE FROM 테이블명 [WHERE 조건];

- 트랜잭션의 대상
  - 트랜잭션은 DML의 집합으로 이루어짐
  - ALL or NOTHING

INSERT

- 테이블에 새로운 튜플을 삽입할 때 사용하는 명령

- Syntax

```
INSERT INTO table_name [{column [, column ...]}]
{VALUES (value[, value...])|Subquery};
```

  - 대응하는 colum과 value는 개수와 타입이 일치해야 한다
  - 테이블 내 모든 컬럼의 내용을 삽입할 때는 column 명을 생략할 수 있지만, 이때는 CREATE TABLE 문에 기술된 컬럼 순으로 value 값들을 지정해야 한다
  - Subquery의 결과를 를 이용하여 다른 테이블의 검색 결과를 삽입할 수 있다

INSERT

- 묵시적 방법: 컬럼 이름, 순서 지정하지 않음. 테이블 생성시 정의한 순서에 따라 값 지정

```
INSERT INTO author
VALUES (1, '박경리', '토지 작가 ' );
```

- 명시적 방법: 컬럼 이름 명시적 사용. 지정되지 않은 컬럼은 NULL 자동 입력

```
INSERT INTO author( author_id, author_name )
VALUES (2, '이문열' );
```

- Subquery 이용: 타 테이블로부터 데이터 복사 (테이블은 미리 존재해야 함)

```
  INSERT INTO department_usa
    SELECT department_id, department_name
      FROM deptments
     WHEREd department_name = 'IT';
```

  - 참고로 Subquery 결과를 없는 테이블을 생성하고 데이터를 복사하고자 할 때는 CREATE TABLE AS SELECT 이용

UPDATE

- 테이블에 있는 튜플들 중에서 특정 튜플의 내용을 갱신할 때 사용하는 명령

- Syntax

```
UPDATE table_name
SET column=value[, column=value ...]
[WHERE condition];
```

  - WHERE 절이 생략된 UPDATE 문장은 해당 테이블 내의 모든 Row를 변경하므로 주의해서 사용해야 한다

UPDATE

- 조건을 만족하는 레코드를 변경
  - 10번 부서원의 금여를 100 인상 & 수수료를 0으로 변경

```
UPDATE employees
SET salary = salary + 100, commission_pct = 0
WHERE department_id=10;
```

- WHERE 절이 생략되면 모든 레코드에 적용
  - 모든 직원의 급여를 10% 인상

```
UPDATE employees SET salary = salary * 1.1
```

- Subquery를 이용한 변경
  - 담당 업무가 'Susan과 같은 사람들의 월급을 부서 최고액으로 변경

```
UPDATE employees SET salary =
    (SELECT MAX(salary) FROM (SELECT salary FROM employees) t)
WHERE job_id =
    (SELECT job_id FROM (SELECT job_id, first_name FROM employees) t2
     WHERE first_name="Susan");
```

DELETE

- 테이블에 있는 튜플들 중에서 특정 튜플을 삭제할 때 사용하는 명령

- Syntax

```
DELETE FROM table_name
[WHERE condition];
```

  - WHERE 절이 생략된 DELETE 문장은 해당 테이블 내의 모든 Row를 삭제하므로 주의해서 사용해야 한다

DELETE

- 조건을 만족하는 레코드 삭제
  - 이름이 'Susan'인 사원 삭제

```
DELETE FROM employees
WHERE first_name = 'Susan';
```

- 조건이 없으면 모든 레코드 삭제 (주의!)
  - 모든 직원 정보 삭제

```
DELETE FROM employees;
```

- FOREIGN KEY 제약이 걸려 있으면 삭제되지 않음

MySQL Database

Transaction

Transaction

- 정의
  - DB에서 하나의 작업으로 처리되는 논리적 작업 단위
  - DBMS의 Concurrency Control과 Recovery에서 중요한 역할을 수행

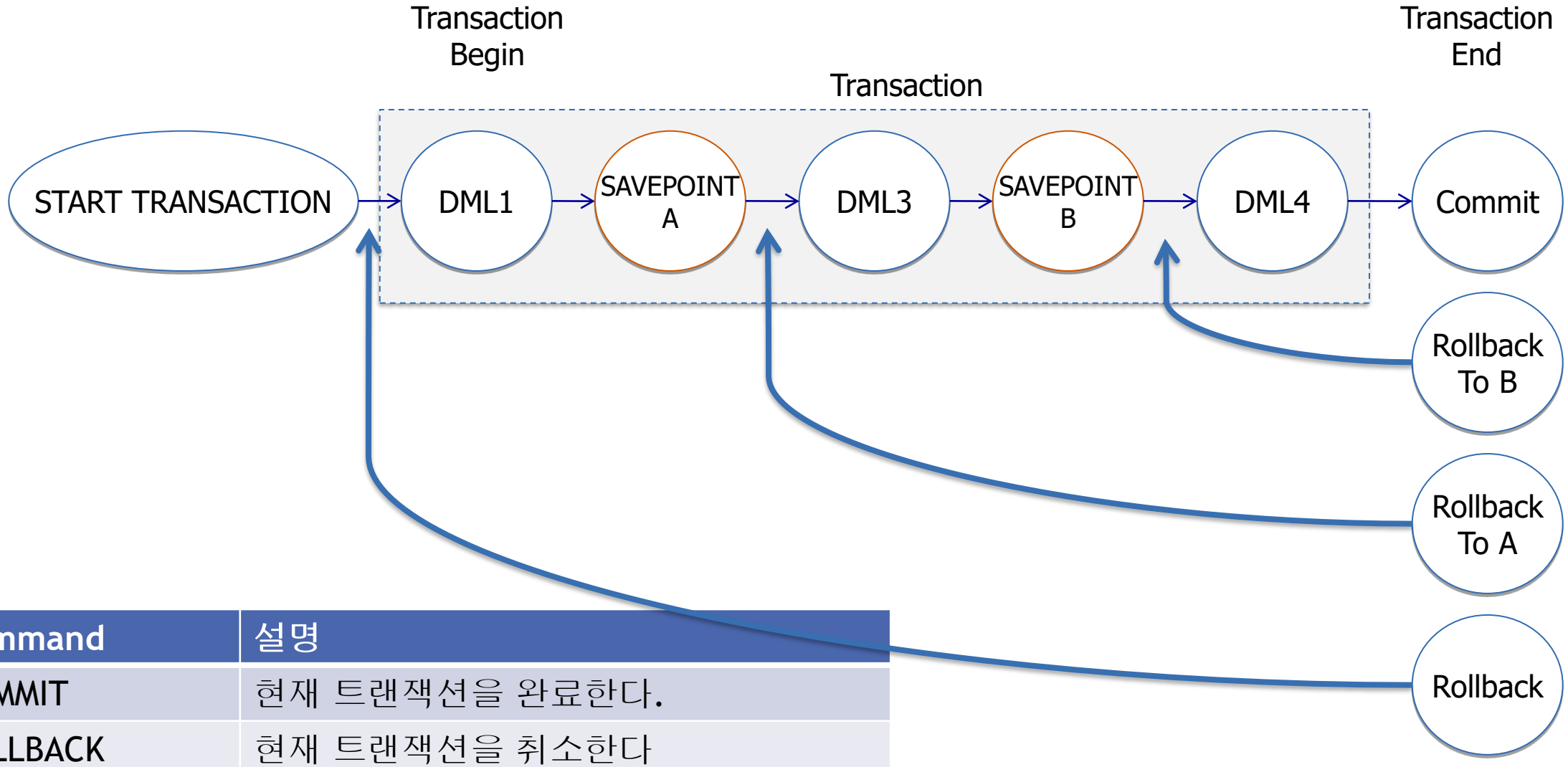
- ACID Property
  - 원자성(Atomicity): All or Nothing. 하나의 단위로 처리되어야 함 (중간 까지만 처리됨은 불가)
  - 일관성(Consistency) : 데이터베이스의 일관성(무결성)을 깨지 않아야 함
  - 격리성(Isolation): 다른 Transaction과 동시에 수행되더라도 독립적으로 영향을 받지 않아야 함
  - 영속성(Durability) : 한번 수행 완료(commit) 되면 영원히 반영되어 있어야 함(시스템 Crash 상황에서도)

Transaction in MySQL

- 구성
  - DML (INSERT, UPDATE, DELETE)의 집합
  - DDL이나 DCL은 한 문장이 트랜잭션으로 처리
- 트랜잭션의 정의
  - 시작:
    - START TRANSACTION
  - 종료:
    - COMMIT
    - ROLLBACK
- MySQL의 Transaction은 Database Engine이 InnoDB일 경우 적용 가능함
- 기본적으로 MySQL은 사용자가 입력한 문장을 자동으로 커밋(AUTO COMMIT)
  - 기본 값을 바꾸고자 할 때에는 AUTOCOMMIT 값을 변경

```
mysql> SET AUTOCOMMIT=0; -- AUTOCOMMIT OFF
mysql> SET AUTOCOMMIT=1; -- AUTOCOMMIT ON
```

Transaction Control



Command	설명
COMMIT	현재 트랜잭션을 완료한다.
ROLLBACK	현재 트랜잭션을 취소한다
SAVEPOINT x	트랜잭션의 중간에 롤백 지점을 만든다
ROLLBACK TO x	지정한 롤백 지점으로 돌아간다

State of Data

- Before Commit/Rollback
  - 현재 사용자는 DML의 결과를 볼 수 있다
  - 다른 사용자는 현재 DML 결과를 볼 수 없다 (변경 이전 버전이 보임. Read Consistency)
  - DML에 의해 변경된 모든 Row는 Lock이 걸린다. (다른 트랜잭션에서 수정 불가)
- After Commit
  - 변경이 영속적으로 DB에 반영됨. 이전 상태는 사라짐 (더 이상 Rollback 불가)
  - 변경 결과를 모든 사용자가 볼 수 있음
  - 모든 Lock이 풀림. 모든 Savepoint 사라짐
- After Rollback
  - 모든 DML의 변경이 취소됨
  - 모든 Lock이 풀림. 모든 Savepoint 사라짐