

데이터 입력/수정/삭제하고 트랜잭션 처리하기

1. INSERT 문으로 데이터 입력하기

(1) 단일 로우 입력 INSERT 문

- INSERT 한 문장을 실행하면 1개 ROW 입력

INSERT INTO 테이블 (칼럼1, 칼럼2, ...)

VALUES (값1, 값2 ...);

- 칼럼과 값은 개수와 데이터 타입이 일치해야 함
- 칼럼 목록 생략 시 테이블의 모든 컬럼에 값 입력
- NOT NULL 속성 컬럼에는 값을 반드시 입력
- 기본 키 컬럼에는 중복 값 입력 불가

1. INSERT 문으로 데이터 입력하기

(2) 다중 로우 입력 INSERT 문

- INSERT 한 문장을 실행해 여러 개의 ROW 입력

INSERT INTO 테이블 (칼럼1, 칼럼2, ...)

VALUES [ROW](값1, 값2, ...),
[ROW](값1, 값2, ...),
...;

1. INSERT 문으로 데이터 입력하기

(3) SELECT 문이 결합된 INSERT 문

- SELECT 문이 반환한 결과 값을 INSERT

INSERT INTO 테이블 (칼럼1, 칼럼2, ...)

SELECT ...

FROM ...

WHERE ...;

2. UPDATE 문으로 데이터 수정하기

(1) 단일 테이블 데이터 수정하기

- 테이블에 저장된 데이터 수정

UPDATE 테이블

SET 칼럼1 = 값1,
칼럼2 = 값2, ...

WHERE ...

ORDER BY ...

LIMIT ...;

- Safe Updates 설정 변경하기
 - MySQL Workbench는 기본으로 WHERE 절이 없는 UPDATE와 DELETE 문을 실행하면 오류가 발생하도록 설정됨
 - Edit → Preferences → SQL Editor → Safe Updates 체크 해제
 - Query → Reconnect to Server

2. UPDATE 문으로 데이터 수정하기

(2) 다중 테이블 데이터 수정하기

- 2개 이상의 테이블에서 데이터 수정

UPDATE 테이블1, 테이블2, ...

SET 칼럼1 = 값1,

칼럼2 = 값2, ...

WHERE 조건;

2. UPDATE 문으로 데이터 수정하기

(3) 입력과 수정을 동시에 처리하기

- 신규로 데이터를 입력하면서 동시에 수정하는 방법
- 조건을 확인해 이미 대상 테이블에 값이 있으면 수정, 없으면 입력

INSERT INTO 테이블명 (칼럼1, 칼럼2, ...)

VALUES 절(또는 SELECT 문)

ON DUPLICATE KEY UPDATE 칼럼 = 값1, 값2, ...;

- ON DUPLICATE KEY UPDATE
 - 기본 키 값 충돌이 발생하는 로우에서는 신규로 값을 입력하지 않고 기존에 저장된 값 변경

3. DELETE 문으로 데이터 삭제하기

(1) 단일 테이블 데이터 삭제하기

- WHERE 절 조건에 맞는 단일 테이블 데이터 삭제
 - WHERE 절이 없으면 데이터 전체 삭제
- ORDER BY 절을 명시하면 해당 순서대로 데이터 삭제
- LIMIT 절로 삭제 데이터 건수 제한

DELETE FROM 테이블

WHERE 조건

ORDER BY ...

LIMIT ...;

3. DELETE 문으로 데이터 삭제하기

(2) 다중 테이블 데이터 삭제하기

- DELETE와 FROM 사이에 명시한 별칭에 해당하는 테이블 데이터만 삭제

- 첫 번째 형태

```
DELETE 테이블별칭1, 테이블별칭2, ...
```

```
FROM 테이블1 테이블별칭1, 테이블2 테이블별칭2, ...
```

```
WHERE 조건;
```

- 두 번째 형태

```
DELETE FROM 테이블별칭1, 테이블별칭2, ...
```

```
USING 테이블1 테이블별칭1, 테이블2 테이블별칭2, ...
```

```
WHERE 조건;
```

4. 트랜잭션 처리하기

(1) 트랜잭션 처리



오류 발생 시 거래 차제를 없던 것으로 처리
입금 계좌에 입금이 확인되면 거래 성사

- COMMIT : 데이터 입력, 수정, 삭제한 후 작업을 영구적으로 테이블에 반영
- ROLLBACK : 데이터 입력, 수정, 삭제한 후 작업을 취소

4. 트랜잭션 처리하기

(2) 자동커밋 속성

- 자동커밋 : COMMIT 문을 명시적으로 실행 안 해도 입력, 수정, 삭제 데이터가 테이블에 반영됨
- 자동커밋 확인 : `SELECT @@AUTOCOMMIT;`
- 자동커밋 비활성화 : `SET autocommit = 0;`
- 자동커밋 활성화 : `SET autocommit = 1;`
- MySQL WorkBench 설정
 - Edit → Preferences → SQL Execution → New connections use auto commit mode

4. 트랜잭션 처리하기

(3) 자동커밋 비활성화 상태에서 트랜잭션 처리하기

```
-- emp_tran1 삭제
DELETE FROM emp_tran1;
-- emp_tran2 삭제
DELETE FROM emp_tran2;
-- 삭제 취소
ROLLBACK;
```

트랜잭션1

```
-- emp_tran1 삭제
DELETE FROM emp_tran1;
-- 삭제 반영
COMMIT;
-- emp_tran2 삭제
DELETE FROM emp_tran2;
-- 삭제 취소
ROLLBACK;
```

트랜잭션1

트랜잭션2

- COMMIT이나 ROLLBACK 문을 실행하면 해당 트랜잭션은 종료됨

4. 트랜잭션 처리하기

(4) 자동커밋 활성화 상태에서 수동으로 트랜잭션 처리하기

- START TRANSACTION 문 : 트랜잭션 선언, 일시적 자동커밋 비활성화

START TRANSACTION;

INSERT 문;

UPDATE 문;

DELETE 문;

...

COMMIT(또는 ROLLBACK);

4. 트랜잭션 처리하기

(4) 자동커밋 활성화 상태에서 수동으로 트랜잭션 처리하기

- SAVEPOINT 문 : 트랜잭션 안에서 특정 지점 설정
- ROLLBACK TO SAVEPOINT 식별자 : 식별자 부분까지 DML 작업 취소

SAVEPOINT 식별자;

INSERT(또는 UPDATE/DELETE) 문

ROLLBACK TO SAVEPOINT 식별자;