

1. 자료의 추가

테이블에 새로운 행을 추가할 때 INSERT문을 실행한다. INSERT문은 한 행을 추가하거나, 서브 쿼리를 사용하여 테이블로부터 행을 복사하는 두 가지 형식이 있다.

1.1. 일반적인 INSERT 문

한 번 실행할 때마다 한 행의 데이터가 저장된다. INSERT문은 테이블에 VALUES 이하의 값1, 값2, ...에 한 번 실행할 때마다 한 행을 저장한다.

```
INSERT INTO 테이블명  
[ (칼럼명1, 칼럼명2, ..... ) ] VALUES (값1, 값2, .....);
```

※ 기술방법

- 테이블명 : 행을 추가할 테이블명을 기술한다.
- 칼럼명1, 칼럼명2, : 추가할 테이블의 칼럼명을 기술한다.
- 값1, 값2, : 입력할 칼럼의 값을 기술한다.

🔑 INSERT문 실행시 주의사항

- 칼럼명과 입력하는 값의 수가 반드시 동일해야 한다.
- 칼럼명과 입력하는 값의 데이터타입이 반드시 동일해야 한다.
- 기본 키와 필수(NOT NULL) 칼럼은 반드시 입력해야 한다.
- 칼럼명이 생략되면 모든 칼럼의 값이 입력되어야 한다.
- 입력되지 않는 칼럼의 값은 널(NULL) 값이 저장된다.
- 입력되지 않는 칼럼에 기본 값이 선언된 칼럼은 기본 값이 저장된다.

🔑 데이터타입이 허용되는 예외사항

칼럼의 데이터타입과 입력하는 값의 데이터타입이 동일하지 않을 경우 허용되는 경우가 있다.

(1) 문자형 칼럼에 숫자형 리터럴로 입력할 때

예: 12, 88, 100,

(2) 숫자형 칼럼에 문자형 리터럴로 입력할 때

예: '12', '100', '555',

(3) 날짜형 칼럼에 문자형 리터럴로 입력할 때, 단, 오라클의 날짜 기본 형식('YYYY-MM-DD' 또는

'DD-MOM-YY')이 일치되어야 한다.

예: '98/01/01', '08/12/11', '14/03/16',

☞ INSERT문 실행 시 오류가 발생하는 경우는 다음과 같다.

- 칼럼명과 입력하는 값의 수가 동일하지 않을 때
- 칼럼명과 입력하는 데이터타입이 동일하지 않을 때, 단 예외 사항은 허용
- 기본 키와 필수(Not Null) 칼럼의 값이 입력되지 않을 때
- 기본 키가 중복된 데이터나 널(null) 값을 입력될 때
- 외부 키 칼럼의 값이 참조하는 테이블의 기본 키 값이 아닐 때
- 고유 키 제약조건이 지정된 칼럼에 중복된 데이터가 입력될 때
- 체크(check) 제약조건이 지정된 칼럼에 유효하지 않은 데이터가 입력될 때
- 입력값이 칼럼의 크기를 초과할 때이다.

예) 1. T_Course 테이블에 새로운 과목을 추가하시오. 과목코드는 'L1061'이고, 과목명이 'ERP실무', 3학점이며, 담당교수가 'P12'이고, 추가 수강료는 50000원이다.

2. T_Course 테이블에 과목코드는 'L1062'이고, 과목명이 '그룹웨어구축', 3학점이며, 담당교수가 'P13'이고, 추가 수강료는 40000원이다.

3. 학번이 'B0901'인 'L1051' 과목의 85점에 대한 성적 취득 날짜는 2010년 6월 28일이다. 이 정보를 SG_Scores 테이블에 입력하시오.

4. 'C0901'의 '박은혜' 학생의 ID_Number 칼럼에 고유 키(Unique) 제약조건(SYS_C0017694)이 중복된 데이터를 찾아 수정하여 입력하시오.

☞ 의사 컬럼

오라클 데이터베이스에서는 몇 가지 의사 컬럼(Pseudo Column)이 있으며, 의사 컬럼을 입력 데이터로 사용할 수 있다.

- SYSDATE : 현재의 날짜와 시간이 기억되어 있다.
- USER : 로그인한 오라클 사용자명이 기억되어 있다.
- ROWNUM : 테이블에 저장된 행의 번호가 기억되어 있다.

특히, SYSDATE와 USER 의사 컬럼은 언제 누가 입력하였는가를 확인할 때 유용할 수 있다.

예) 2010년 6월 28일에 취득한 'C0931' 학번의 'L1031'과목의 97점이 누락하여 추가 입력하고자

한다. 누가 언제 입력하는지의 정보를 추가하시오.

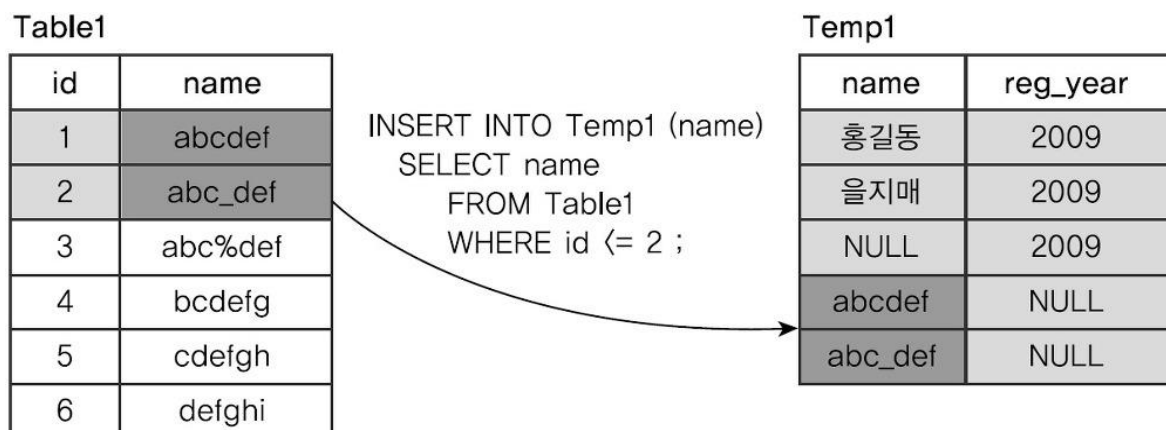
☞ INSERT문에서 칼럼명을 기술해야 하는 이유

- 테이블의 구조는 항상 변할 수 있다.
- INSERT문의 이해가 쉬어진다.
- 인접된 칼럼에 잘못 저장되는 경우를 방지할 수 있다.

1.2. 서브 쿼리가 있는 INSERT문

서브 쿼리가 있는 INSEERT문은 기존 테이블의 모든 데이터 또는 일부분을 다른 테이블에 동일하게 복사할 때 유용하게 사용된다. INSERT문에 사용된 SELECT문을 서브 쿼리라 한다.

```
INSERT INTO 테이블명  
[ (칼럼명1, 칼럼명2, ..... ) ] SELECT문;
```



☞ 서브 쿼리가 있는 INSERT문 사용시 주의사항

- INSERT문의 칼럼 수와 SELECT절의 칼럼 수가 반드시 동일해야 한다.
- INSERT문의 칼럼과 SELECT절의 칼럼의 데이터타입이 반드시 동일해야 한다.
- SELECT절에 기술된 칼럼이 INSERT문에 기술된 칼럼의 순서대로 값이 저장된다.

예) Course 테이블로부터 교수번호가 'P12'인 교수의 개설과목의 정보를 T_Course 테이블에 저장하시오.

2. UPDATE문

테이블에 저장된 각 행들의 칼럼 값을 변경할 때 실행한다.

UPDATE 테이블명

SET 칼럼명1 = expr1, 칼럼명2 = expr2, [WHERE 조건]

※ 기술 방법

- 테이블명 : 수정할 테이블명을 기술한다.
- 칼럼명1, 칼럼명2, : 수정할 테이블명의 칼럼명을 기술한다.
- expr1, expr2, : 수정할 값을 리터럴, 칼럼명, 수식, 함수 등을 기술한다.
- WHERE 조건 : 수정할 조건을 기술한다.

UPDATE문은 expr1, expr2,의 값을 왼쪽의 칼럼명에 저장한다. WHERE절이 없으면 모든 행이 변경되고, WHERE절이 있으면 조건이 참이 되는 행에 대해서만 변경된다.

기본 키 값은 변경할 수 없으며, 외부 키 컬럼은 참조하는 테이블의 기본 키 값을 만족할 때 변경할 수 있다.

예) 1. Department 테이블의 학과명(Dept_Name)을 '컴퓨터정보계열'로 수정하시오.

2. 학번이 'C0901'이고, 과목코드가 'L1042'의 성적이 85점을 105점으로 잘못 입력하였다. SG_Scores 테이블의 성적을 변경하시오.

🔗 UPDATE문에서 서브 쿼리 사용

칼럼 값을 동적으로 변경하기 위해서 UPDATE문에 서브 쿼리를 사용할 수 있다.

예) Course 테이블의 추가 수강료(Course_Fees)가 평균값보다 높은 경우 5000원을 삭감하시오.

3. DELETE문

테이블의 행을 삭제한다. 모든 행을 삭제할 수도 있고, 특정 행을 삭제할 수도 있다.

DELETE FROM 테이블명

[WHERE 조건] ;

※ 기술 방법

- 테이블명 : 삭제할 테이블명을 기술한다.
- 조건 : 삭제할 조건을 기술한다.

DELETE문에서 WHERE절을 기술하지 않으면 모든 행을 삭제하고, WHERE절을 기술하면 삭제 조건이 참이 되는 행을 삭제한다.

예) 1. T_Couse 테이블의 모든 행을 삭제하시오.

2. Cours 테이블의 과목코드(Course_ID)가 'L0012'이고, 과목명(Title)이 '문학과 여행'인 과목을 삭제하시오.

☞ 테이블의 행 삭제가 불가능한 경우

예를 들어 삭제할 행이 Professor 테이블의 외부 키로 참조되고 있을 경우에는 Department 테이블의 행을 삭제할 수 없다.

예) Department 테이블에서 '컴공'학과의 행을 삭제하시오.

☞ TRUNCATE 명령어를 이용한 행 삭제

SQL*Plus 명령어로 테이블의 모든 행을 삭제한다. 모든 행을 삭제할 때는 DELETE문보다 TRUNCATE TABLE 명령어가 더 효율적이다. 그러나 TRUNCATE TABLE 명령어로 삭제된 행들은 복수할 수가 없다.

예) Computer_Student 테이블의 모든 행을 삭제하시오.

4. MERGE문

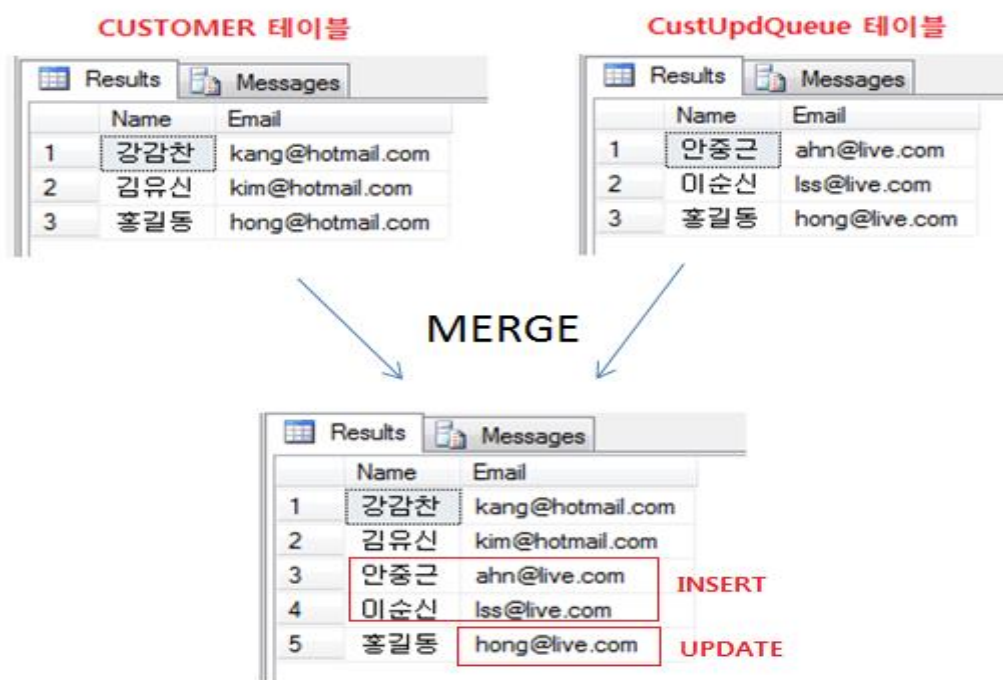
테이블 구조가 동일한 두 개 이상의 테이블이나 뷰, 서브 쿼리로부터 데이터를 비교하여 하나의 테이블에 데이터를 병합하는 것으로, 오라클 데이터베이스에서만 사용 가능한 명령문이다. MERGE 문을 사용하면 입력과 수정을 동시에 할 수 있다.

```
MERGE INTO 테이블명-A
      USING [ 테이블-B | 뷰 | 서브 쿼리 ] ON [조건]
WHEN MATCHED THEN
      UPDATE SET ...
WHEN NOT MATCHED THEN
      INSERT INTO ... VALUES ...
```

※ 기술 방법

- 테이블명1 : 대상 테이블명을 기술한다.

- USING : 원본 테이블이나 뷰 또는 서브 쿼리를 기술한다.
- 조건 : 원본 테이블과 대상 테이블에 대한 조건을 비교하여, 만족하면 UPDATE문이나 DELETE문을 실행하고, 만족하지 않으면 INSERT문을 실행한다.
- WHEN MATCHED : 조건이 만족할 경우 실행할 UPDATE문이나 DELETE문을 기술한다.
- WHEN NOT MATCHED : 조건이 만족하지 않을 경우 실행할 INSERT문을 기술한다.



예) Course_Temp 테이블에 새로 개설된 과목과 변경된 과목을 이용하여 Course 테이블에 병합 하시오.

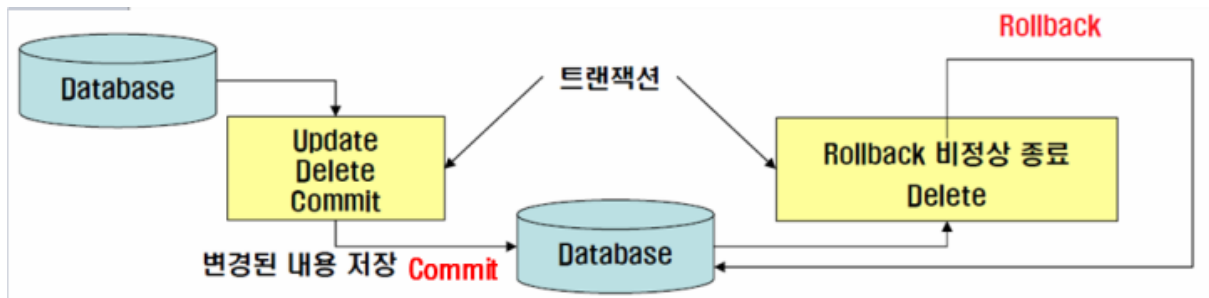
5. 트랜잭션 제어 명령

오라클 데이터베이스에서는 테이블에 대한 데이터 변경 작업이 즉각적으로 저장되지 않는다. 오라클 데이터베이스는 어떤 기간을 두고, 사용자가 일정기간 테이블에 대한 변경 작업을 할 수 있도록 한 후에, 그 변경 작업을 테이블에 저장하도록 하고 있다.

트랜잭션(Transaction)이란 테이블에 대하여 '행을 추가하거나 칼럼 값 수정, 또는 행의 삭제에 의해 변경되는 작업의 논리적 단위' 또는 '변경 작업을 할 수 있도록 사용자에게 할당된 기간'을 말한다.

테이블의 INSERT문, UPDATE문, DELETE문에 대한 데이터 변경은 직접적으로 테이블에 적용되는

것이 아니고, 변경하는 테이블의 연관된 트랜잭션 영역에 일정기간동안 임시로 저장되었다가 트랜잭션 제어문에 의해서 영구히 저장되기도 하고, 취소되기도 한다.



트랜잭션 제어문이 실행되면, 트랜잭션 영역은 초기화되고, 트랜잭션이 종료되며, 그 이후의 실행되는 명령문에 대하여 트랜잭션이 다시 시작되고, 트랜잭션 영역이 다시 설정된다.

- COMMIT문은 테이블에 대한 트랜잭션 영역의 변경 작업을 영구히 저장한다.
- ROLLBACK문은 테이블에 대한 트랜잭션 영역의 변경 작업을 취소한다.

5.1. COMMIT문

트랜잭션 영역에 있는 행의 추가, 삭제, 수정된 내용을 테이블에 영구히 저장 또는 삭제한다. COMMIT문이 실행되면, 트랜잭션은 종료되고, 트랜잭션 영역은 초기화된다.

예) Student 테이블로부터 '컴공' 학과의 학생 정보를 Computer_Student 테이블에 영구히 저장 하시오.

5.2. ROLLBACK문

트랜잭션 영역에 있는 행의 추가, 수정, 삭제에 대한 명령의 실행을 취소하여, 데이터를 원상 복구시킨다. 그러나 COMMIT된 테이블의 데이터에 대해서는 원상복구를 시킬 수 없다. ROLLBACK문이 실행되면, 트랜잭션은 종료되고, 트랜잭션 영역은 초기화된다.

```
ROLLBACK;

ROLLBACK TO 저장점명;
```

- ROLLBACK문은 트랜잭션 영역의 모든 명령을 취소한다.
- ROLLBACK TO문은 트랜잭션 영역의 부분적으로 실행된 명령문을 취소할 수 있다.
- 저장점은 SAVEPOINT문으로 정의한다.

예) Computer_Student 테이블을 잘못하여 모든 행을 삭제하였다. 트랜잭션의 명령을 취소하시오.

5.3. 저장점 지정

SAVEPOINT문은 간혹 실수할지도 모르는 테이블 데이터 변화에 대한 부분적인 작업 취소를 위해 트랜잭션의 기간을 부분적으로 할당할 때 사용하고, ROLLBACK TO문으로 부분적으로 할당받은 트랜잭션에 대하여 취소한다.

예) Temp 테이블을 생성하여 데이터를 입력한 후, 저장점을 선언하고, 부분적으로 취소한 후, 영구히 저장하시오.

5.4. 자동 COMMIT과 자동 ROLLBACK

- SQL*Plus에서 COMMIT문이나 ROLLBACK문을 실행하지 않고 정상적으로 종료하면, 트랜잭션 영역에 있는 데이터들은 그 테이블에 자동으로 COMMIT 된다.
- SQL*Plus 실행 중 정전이 발생하거나, 비정상적으로 시스템이 다운되면 자동으로 ROLLBACK 된다.