

초보자를 위한  
**Oracle SQL Database**



## 데이터 수정



한국기술교육대학교  
온라인평생교육원

## 학습내용

- UPDATE 문
- DELETE 및 MERGE 문

## 학습목표

- UPDATE 문으로 데이터를 수정할 수 있다.
- DELETE 문으로 데이터를 삭제하고,  
MERGE 문을 사용할 수 있다.

# UPDATE 문

데이터 수정

UPDATE 문

## ◆ UPDATE 문 설명

### UPDATE 문

**UPDATE** 테이블명

**SET** 컬럼명 = 변경할 값 [, 컬럼명1 = 변경할 값] ..

[**WHERE** 조건식];

[ ] : 대괄호는 생략 가능, | : 여러 옵션 중 한 개 선택

WHERE 절에 해당하는 행이 여러 개인 경우  
모두 SET 절의 값으로 변경이 되기 때문에  
확인 후 사용할 것을 추천

데이터 수정

UPDATE 문

## ◆ UPDATE 문 예 - 1

### UPDATE 문 예 - 1

**UPDATE** "학생"

**SET** "이수학기" = 이수학기 + 1;

한 학기가 지나서 이수학기의 값을 1씩  
증가를 하게 하는 UPDATE 문

# UPDATE 문

데이터 수정

UPDATE 문

## ◆ UPDATE 문 예 - 1

The screenshot shows the Oracle SQL Developer interface. In the workspace, there is a script editor window containing the following SQL code:

```
19c remote > 교과목 >
[...]
워크시트 | 질의 작성기
[...]
UPDATE "학생"
SET "이수학기" = 이수학기 + 1;
```

Below the workspace, a script output window shows the result of the execution:

```
작업이 완료되었습니다.(0.048초)
```

A message box on the right side of the screen contains two pieces of explanatory text:

- 이수학기 + 1;에서 이수학기에는 따옴표가 없음
- 따옴표로 둘러싸면 컬럼이 아니라 문자상수가 되기 때문

At the bottom of the workspace, a message indicates 97개 행이 업데이트되었습니다.

데이터 수정

UPDATE 문

## ◆ UPDATE 문 예 - 1

### UPDATE 문 예 - 1

```
UPDATE "학생"
SET "이수학기" = 이수학기 + 1;
WHERE "학적상태" = '재학';
```

"이수학기" = 이수학기 + 1;을 하게 되면 모든 학생의 이수학기가 10이 증가

휴학생은 제외를 해야 하기 때문에 WHERE 조건이 필요하게 됨

# UPDATE 문

데이터 수정

UPDATE 문

## UPDATE 문 예 - 2

### UPDATE 문 예 - 2 상황

- 다음과 같이 군입대로 인해 “입대휴학”을 하는 학생이 있음

학번	이름	학적상태	학적변동 사항
2020136137	최현진	재학 -> 휴학	당초입학 -> 입대휴학

- 학적변동 컬럼을 갖고 있는 테이블은 “학생” 테이블
- 위 학생의 학적상태 컬럼의 값을 “휴학”으로, 학적변동 컬럼의 값을 “입대휴학”으로 변경해야 함

데이터 수정

UPDATE 문

## UPDATE 문 예 - 2

19c remote 학생

데이터 Model 제약 조건 권한 부여 통계 트리거 플래시백 증속성 세부정보 분할 영역 인덱스 SQL

학번 이름 성명 전공코드 학년 성별 학적상태 미수학과 학적변동 이메일

1 2020136137 최현진 7003 1 남 재학 당초입학 asdf34

UPDATE 문을 실행하기 전 데이터 값

```
UPDATE "학생"
SET "학적상태" = '휴학', "학적변동" = '입대휴학'
WHERE "학번" = '2020136137';
```

스크립트 출력

작업이 완료되었습니다.(0.044초)

1 행 이 (가) 업데이트되었습니다.

UPDATE 문을 실행한 후 데이터 값

19c remote 학생

데이터 Model 제약 조건 권한 부여 통계 트리거 플래시백 증속성 세부정보 분할 영역 인덱스 SQL

학번 이름 성명 전공코드 학년 성별 학적상태 미수학과 학적변동 이메일

1 2020136137 최현진 7003 1 남 휴학 입대휴학 asdf34789

# UPDATE 문

데이터 수정

UPDATE 문

## 외부와 연결된 키의 UPDATE 문

### 외부와 연결된 키의 UPDATE 문

- 테이블과 테이블들의 값이 서로 연결되어서 있는 것이 관계형 데이터베이스
- UPDATE 문으로 데이터를 수정할 때, 문법에 오류가 없다고 실행되는 것이 아님
- 데이터가 입력, 수정, 삭제 될 때에 데이터의 무결성을 유지하지 위해 RDBMS가 데이터를 확인하는 과정을 거치게 됨

데이터 수정

UPDATE 문

## 외부와 연결된 키의 UPDATE 문

### 학부코드를 10에서 11로 변경하는 UPDATE 문의 예

```
UPDATE "학부"
SET "학부코드" = 11
WHERE "학부코드" = 10;
```

# UPDATE 문

## 데이터 수정

### UPDATE 문

#### 외부와 연결된 키의 UPDATE 문

```
워크시트   질의 작성기
UPDATE "학부"
SET "학부코드" = 11
WHERE "학부코드" = 10;

스크립트 출력 x
작업이 완료되었습니다.(0.069초)

명령의 1 행에서 시작하는 중 오류 발생 -
UPDATE "학부"
SET "학부코드" = 11
WHERE "학부코드" = 10
오류 보고 -
ORA-02292: 무결성 제약조건 (STUDY01.교과목_FK)이 위배되었습니다~ 자식 레코드가 발견되었습니다.
```

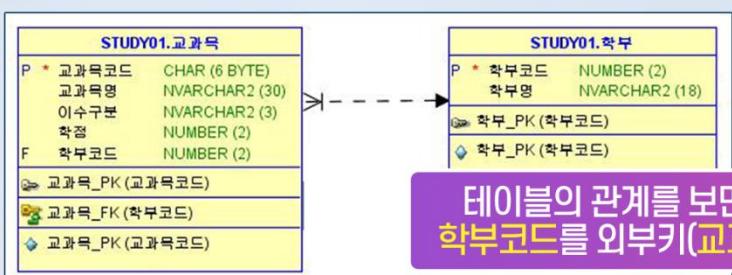
오류: “무결성 제약조건(STUDY01.교과목\_FK)이 위배되었습니다~ 자식 레코드가 발견되었습니다.”

SQL 문을 실행하면 오류가 발생

## 데이터 수정

### UPDATE 문

#### 외부와 연결된 키의 UPDATE 문



테이블의 관계를 보면 교과목테이블에서 학부테이블의 학부코드를 외부키(교과목\_FK)로 사용하고 있는 것이 보임

연관되어 있는 교과목테이블의 학부코드와 무결성이 깨지게 됨

따라서 학부코드의 값이 바뀌게 되면

학부코드를 그냥 11로 변경하게 되면, 교과목 테이블에 있는 학부코드 컬럼 값이 갖고 있는 10은 더 이상 학부 테이블에서 그 값을 찾을 수 없게 됨

학부코드 기준으로 학부테이블이 부(父)가 되고 교과목테이블이 자식(子息)의 관계가 됨

# UPDATE 문

데이터 수정

UPDATE 문

## 외부와 연결된 키의 UPDATE 문

### 무결성 제약 조건과 테이블 간 관계

- 무결성 제약 조건을 위배하지 않으려고 하면 명령문에 따라 신경 써야 할 부자간의 관계를 다음과 같이 표로 나타냄

명령문	학부테이블 (부)	교과목테이블 (자식)
INSERT		O
UPDATE	O	O
DELETE	O	

- 학부테이블을 기준으로 학부테이블과 교과목 테이블의 관계를 보면, 학부테이블이 부가되고 교과목 테이블이 자식이 됨

데이터 수정

UPDATE 문

## 외부와 연결된 키의 UPDATE 문

### 무결성 제약 조건과 테이블 간 관계

- 학부테이블에 새로운 데이터를 입력하는 것은 신경 쓸 것이 없이 넣으면 됨

```
INSERT INTO "학부" ("학부코드", "학부명")
VALUES (15, '자동차공학부');
```

- 학부테이블의 데이터를 지울 때는 교과목 테이블의 학부코드가 참조하고 있는 값을 지우면 안됨

```
DELETE FROM "학부" WHERE 학부코드 = 60;
```

# DELETE 및 MERGE 문

데이터 수정

DELETE 및 MERGE 문

## DELETE 문

### DELETE 문

**DELETE FROM** 테이블명

[**WHERE** 조건식];

[ ] : 대괄호는 생략 가능, | : 여러 옵션 중 한 개 선택.

WHERE 절에 해당하는 행이 여러 개인 경우  
모두 삭제가 되기 때문에 확인 후 사용할 것을  
추천

WHERE 절을 생략하는 경우 테이블의  
모든 행이 삭제됨

데이터 수정

DELETE 및 MERGE 문

## DELETE 문

### + DELETE 문 예 - 1

#### DELETE 문 예 - 1

**DELETE FROM** "학부" **WHERE** 학부코드 = 15;

```
워크시트  질의 작성기
DELETE FROM "학부" WHERE 학부코드 = 15;

스クリプ트 출력 x
작업이 완료되었습니다.(0.046초)

1 행 이(가) 삭제되었습니다.
```

# DELETE 및 MERGE 문

## 데이터 수정

### DELETE 및 MERGE 문

#### ◆ DELETE 문

##### + TRUNCATE TABLE 문

#### TRUNCATE TABLE 문

- 앞에서 테이블의 모든 행을 삭제하기 위해 DELETE 문에서 WHERE 절을 생략하였는데, 또 다른 방법으로 TRUNCATE TABLE 문을 사용할 수 있음

TRUNCATE TABLE 테이블명;

워크시트 | 질의 작성기

```
TRUNCATE TABLE "학생";
```

스크립트 출력 x | 작업이 완료되었습니다.(0.09초)

Table "학생"이 (가) 잘렸습니다.

## 데이터 수정

### DELETE 및 MERGE 문

#### ◆ DELETE 문

##### + DELETE 문 예 - 2

#### DELETE 문 예 - 2

- WHERE 절이 생략되었기 때문에 수강 테이블의 모든 행이 삭제됨

DELETE FROM "수강";

워크시트 | 질의 작성기

```
DELETE FROM "수강";
```

스크립트 출력 x | 작업이 완료되었습니다.(0.043초)

43개 행 이 (가) 삭제되었습니다.

# DELETE 및 MERGE 문

데이터 수정

DELETE 및 MERGE 문

## ◆ DELETE 문

### + TRUNCATE TABLE 문

#### DELETE와 TRUNCATE의 차이점

명령문	DELETE 문	TRUNCATE
속도	느리다	빠르다
AUTO COMMIT	안함	실행
데이터 용량	변화 없음	줄어듦

데이터 수정

DELETE 및 MERGE 문

## ◆ MERGE INTO 문

#### MERGE INTO 문

- MERGE INTO 문은 테이블에 데이터가 이미 존재하고 있으면 업데이트(또는 삭제)
- 데이터가 없으면 입력을 해야 할 때 사용할 수 있음

# DELETE 및 MERGE 문

데이터 수정

DELETE 및 MERGE 문

## MERGE INTO 문

### + 단일 테이블에서 MERGE 사용

#### 단일 테이블에서 MERGE 사용

- 단일 테이블 사용은 USING 뒤에 DUAL을 적어 사용

```
MERGE INTO 테이블명1 [별칭]
USING DUAL
ON (조인조건)
[WHEN MATCHED THEN
    UPDATE SET 컬럼명 = 값, 컬럼명1 = 값1
[WHERE 조건식]
    [DELETE WHERE 조건식]
| WHEN NOT MATCHED THEN
    INSERT (컬럼 목록) VALUES (값 목록)
[WHERE 조건식] ;
```

[ ] : 대괄호는 생략 가능, | : 여러 옵션 중 한 개 선택.

데이터 수정

DELETE 및 MERGE 문

## MERGE INTO 문

### + 단일 테이블에서 MERGE 사용

#### 단일 테이블에서 MERGE 사용 예

- 교수의 임용 사항이 다음과 같이 발표 되었다고 가정

교수사번	교수명	직렬	직급	학부코드
20220011	강감찬	임시교원	특임교수	70

# DELETE 및 MERGE 문

## 데이터 수정

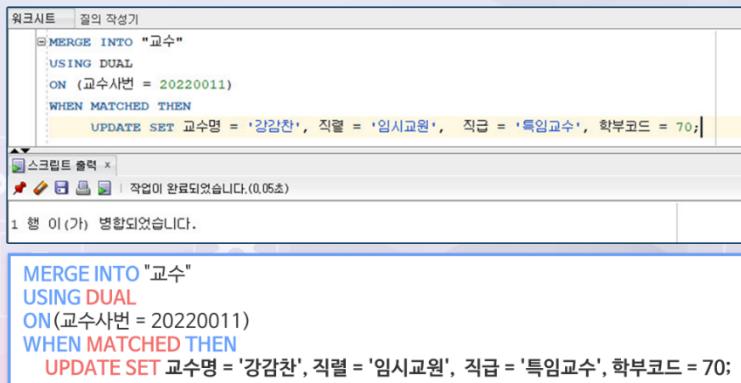
### DELETE 및 MERGE 문

#### MERGE INTO 문

##### 단일 테이블에서 MERGE 사용

###### 단일 테이블에서 MERGE 사용 예

- 기존에 있는 교수면 데이터를 수정하고, 없으면 아무것도 안 하려면 WHEN MATCHED만 작성



```

    워크시트  편집 작성기
    MERGE INTO "교수"
    USING DUAL
    ON (교수사번 = 20220011)
    WHEN MATCHED THEN
        UPDATE SET 교수명 = '강감찬', 직렬 = '임시교원', 직급 = '특임교수', 학부코드 = 70;
    
```

스크립트 출력 x | 작업이 완료되었습니다.(0.05초)

1 행 이(가) 병합되었습니다.

```

MERGE INTO "교수"
USING DUAL
ON(교수사번 = 20220011)
WHEN MATCHED THEN
    UPDATE SET 교수명 = '강감찬', 직렬 = '임시교원', 직급 = '특임교수', 학부코드 = 70;
    
```

## 데이터 수정

### DELETE 및 MERGE 문

#### MERGE INTO 문

##### 단일 테이블에서 MERGE 사용

###### 단일 테이블에서 MERGE 사용 예

- 반대로, 기존에 있으면 아무것도 안하고, 없을 때에만 입력을 하려고 하면 WHEN NOT MATCHED만 작성



```

    워크시트  편집 작성기
    MERGE INTO "교수"
    USING DUAL
    ON (교수사번 = 20220011)
    WHEN MATCHED THEN
        UPDATE SET 교수명 = '강감찬', 직렬 = '임시교원', 직급 = '특임교수', 학부코드 = 70;
    
```

스크립트 출력 x | 작업이 완료되었습니다.

1 행 이(가) 병합되었습니다.

```

MERGE INTO "교수" S
USING DUAL
ON(교수사번 = 20220011)
WHEN NOT MATCHED THEN
    INSERT (교수사번, 교수명, 직렬, 직급, 학부코드 )
    VALUES (20220011, '강감찬', '임시교원', '특임교수', 70);
    
```

# DELETE 및 MERGE 문

## 데이터 수정

### DELETE 및 MERGE 문

#### MERGE INTO 문

##### 단일 테이블에서 MERGE 사용

###### 단일 테이블에서 MERGE 사용 예

교수사번	교수명	직렬	직급	학부코드
20220011	강감찬	임시교원	특임교수	70

- 기존에 있는 교수이면 데이터를 수정하고, 없는 경우 입력을 하려고 하면 다음과 같이 MERGE 문 작성

```

MERGE INTO "교수"
USING DUAL
ON(교수사번 = 20220011)
WHEN MATCHED THEN
    UPDATE SET 교수명 = '강감찬', 직렬 = '임시교원', 직급 = '특임교수', 학부코드 = 70
WHEN NOT MATCHED THEN
    INSERT (교수사번, 교수명, 직렬, 직급, 학부코드)
    VALUES (20220011, '강감찬', '임시교원', '특임교수', 70 ) ;0
  
```

## 데이터 수정

### DELETE 및 MERGE 문

#### MERGE INTO 문

##### 동일한 구조를 갖고 있는 테이블에서 MERGE 할 경우

###### 동일한 구조를 갖고 있는 테이블에서 MERGE 할 경우

- 테이블 구조 복사는 다음과 같이 할 수 있음

```

CREATE TABLE 신규테이블명 AS
SELECT * FROM 복사할테이블명 WHERE 1=2;
  
```

# DELETE 및 MERGE 문

## 데이터 수정

### DELETE 및 MERGE 문

#### MERGE INTO 문

- 동일한 구조를 갖고 있는 테이블에서 MERGE 할 경우

동일한 구조를 갖고 있는  
테이블에서 MERGE 할 경우

- 교수 테이블과 동일한 구조를 갖는  
교수인사발령이라는 테이블 만들기

```
워크시트 | 질의 작성기
CREATE TABLE 교수인사발령 AS SELECT * FROM 교수 WHERE 1=2;
```

스크립트 출력 x | 작업이 완료되었습니다.

CREATE TABLE 교수인사발령 AS  
SELECT \* FROM 교수 WHERE 1=2;

Table 교수인사발령이 (가) 생성되었습니다.

## 데이터 수정

### DELETE 및 MERGE 문

#### MERGE INTO 문

- 동일한 구조를 갖고 있는 테이블에서 MERGE 할 경우

동일한 구조를 갖고 있는  
테이블에서 MERGE 할 경우

- 교수인사발령 테이블에 발령 데이터를 입력

교수사번	교수명	직렬	직급	학부코드
20091015	김종원	전임교원	부교수	30
20220012	이순신	전임교원	부교수	30
20092019	박홍재	전임교원	교수	60
20220013	을지문덕	전임교원	교수	40

```
워크시트 | 질의 작성기
INSERT INTO "교수인사발령" VALUES (20091015,'김종원','전임교원','부교수',30);
INSERT INTO "교수인사발령" VALUES (20220012,'이순신','전임교원','부교수',30);
INSERT INTO "교수인사발령" VALUES (20092019,'박홍재','전임교원','교수',60);
INSERT INTO "교수인사발령" VALUES (20220013,'을지문덕','전임교원','교수',40);
```

스크립트 출력 x | 작업이 완료되었습니다.(0.053초)

1 행 이 (가) 삽입되었습니다.

# DELETE 및 MERGE 문

## 데이터 수정

### DELETE 및 MERGE 문

#### MERGE INTO 문

- 동일한 구조를 갖고 있는 테이블에서 MERGE 할 경우

#### 동일한 구조를 갖고 있는 테이블에서 MERGE 할 경우

- 교수인사발령 테이블에 있는 교수이면 데이터를 수정을 하고, 없는 경우에는 입력을 하는 MERGE 문은 다음과 같음

```
워크시트 — 질의 작성기
MERGE INTO "교수" S
USING "교수인사발령" T
ON ( S.교수사번 = T.교수사번 )
WHEN MATCHED THEN
    UPDATE SET S.교수명 = T.교수명, S.직렬 = T.직렬,
               S.직급 = T.직급, S.학부코드 = T.학부코드
WHEN NOT MATCHED THEN
    INSERT ( S.교수사번, S.교수명, S.직렬, S.직급, S.학부코드 )
    VALUES ( T.교수사번, T.교수명, T.직렬, T.직급, T.학부코드 );
작업이 완료되었습니다.(0.041초)
4개 행 이(가) 병합되었습니다.
```

```
MERGE INTO "교수" S
USING "교수인사발령" T
ON ( S.교수사번 = T.교수사번 )
WHEN MATCHED THEN
    UPDATE SET S.교수명 = T.교수명, S.직렬 = T.직렬,
               S.직급 = T.직급, S.학부코드 = T.학부코드
WHEN NOT MATCHED THEN
    INSERT ( S.교수사번, S.교수명, S.직렬, S.직급, S.학부코드 )
    VALUES ( T.교수사번, T.교수명, T.직렬, T.직급, T.학부코드 );
```

# 실습하기

데이터 수정

실습하기

- 이수학기에 따라 학년을 계산하는 update문을 작성하시오.
- 힌트 : 학적상태가 '재학'인 학생들만 대상으로 해야 합니다.

이수학기가 1,2 이면 1학년, 3,4이면 2학년, 5,6이면 3학년, 7이상은 4학년입니다.

- 군입대/복학에 따른 “군휴학변동” 테이블에 다음과 같은 데이터가 들어 있을 때, 학생 테이블에 데이터를 수정하는 MERGE문을 작성하시오.

학번	성명	학적상태	학적변동
2021140029	김재민	휴학	입대휴학
2021120045	박인수	재학	제대복학
2019161095	이재준	휴학	입대휴학
2021180100	정상훈	휴학	입대휴학

- + Oracle Database 19c 버전 사용
- + 최신 버전인 Oracle Database 21c는 현재 cloud에서만 사용 가능하며, 지원 기간은 2년

## 실습단계

데이터수정 적용하기

현재 이수학기에 +1하기

대상은 학적상태가 “재학”인 학생

데이터 새로고침

이수학기에 따른 학년 계산

“군휴학변동” 테이블은 컬럼이 학번, 성명, 학적상태, 학적변동

“학생” 테이블과 “군휴학변동” 테이블의 구조가 서로 다를 때

MERGE 문을 작성하는 법은?

“군휴학변동” 테이블 생성

“군군휴학변동” 테이블에 데이터 입력

MERGE 문 작성

“학생” 테이블의 데이터를 수정하기 위해 MERGE INTO 뒤에 “학생” 적음

“학생” 테이블의 별칭을 S로 지정

테이블의 구조가 다르면 USING 뒤에 컬럼 적음



# 실습하기

## 실습단계

“군휴학변동” 테이블 별칭을 T로 지정

조인 조건을 “학생” 테이블의 학번과 “군휴학변동” 테이블의 학번이 같을 경우로 지정

“학생” 테이블 학적상태와 학적변동 데이터를 “군휴학변동” 테이블의 값으로 수정

# 실습하기

## 실습단계

데이터 삭제 및 복구

데이터 삭제

데이터 복구(ROLLBACK)

TRUNCATE TABLE로 테이블 데이터 삭제

TRUNCATE TABLE은 ROLLBACK으로 복구가 안됨

테이블 삭제 - DROP TABLE

Flashback Drop 명령 사용하기

휴지통에 있는 테이블 보기 - SHOW RECYCLEBIN;

테이블이 실제로 삭제되는 것이 아니라 휴지통에 들어가게 되는 것

테이블을 다른 이름으로 복원하기

휴지통에 삭제된 테이블 이름이 같은 것이 여러개 있을 경우 복원하기

휴지통 비우기

휴지통에서 하나 비우기

휴지통 전체 비우기

DROP TABLE 테이블명 PURGE;

<https://otsteam.tistory.com/157>