

## 9. 데이터 조작

컴퓨터공학부

김은경

## [학습 목표]

1. 테이블에 새로운 행을 삽입할 수 있다.
2. 테이블에 날짜 값을 갖는 행을 삽입할 수 있다.
3. 테이블에 있는 기존 행을 변경할 수 있다.
4. 테이블에 있는 기존 행을 삭제할 수 있다.
5. 데이터 삭제 시에 발생할 수 있는 오류에 대해서 설명할 수 있다

# 1. 행 삽입

## ★ 데이터 조작어(DML : Data Management Language)란?

- 이미 정의된 데이터를 검색, 삽입, 삭제, 변경하기 위해 사용되는 언어로, SELECT, INSERT, DELETE, UPDATE 명령어가 있음

## ★ 트랜잭션(Transaction)이란?

- 작업의 논리적인 단위로, 하나 이상의 DML 문장으로 구성됨
- 실제로 DB에서 한 번에 처리되는 일의 단위
  - 즉, 실제로 DB에 변화를 유발하는 일의 단위
- 실행 시 트랜잭션을 구성하는 한 문장이라도 실패하면, 다른 모든 문장의 실행이 취소됨

# 1. 행 삽입

## ★ 행 삽입 명령어

```
INSERT INTO 테이블이름 [(열 이름1 [, 열 이름2 ... ])]  
VALUES (값1 [, 값2 ...]);
```

## ★ INSERT 명령의 특징

- ① VALUES 절에 열의 값을 명시하므로 한 번에 단 한 행만 삽입할 수 있음
- ② 열 이름을 명시하지 않으면, 테이블에 정의된 열의 순서대로 값을 나열해야 함
- ③ 열 이름을 명시하면, 열 이름의 순서 및 개수와 대응하는 값의 순서 및 개수가 일치해야 함

# 1. 행 삽입

## ✓ 행 삽입 예제(1)

- dept 테이블에 부서이름이 '영업2부', 부서번호가 402, 부서위치가 '천안'인 행을 삽입하되, 열을 부서이름, 번호, 위치 순으로 삽입하시오.

```
INSERT INTO dept(dept_name, dept_id, dept_loc)
VALUES ('영업2부', 402, '천안');
SELECT *
FROM dept;
```

DEPT_ID	DEPT_NAME	DEPT_LOC
100	관리부	서울
200	연구부	대전
300	총무부	서울
400	영업부	인천
500	개발부	천안
402	영업2부	천안

6 rows selected.

# 1. 행 삽입

## ✓ 행 삽입 예제(2)

- dept 테이블에 부서번호가 404, 부서이름이 '영업4부', 부서위치가 '광주'인 행을 삽입하되, 열 이름을 나열하지 않고 삽입하시오.

```
INSERT INTO dept  
VALUES (404, '영업4부', '광주');  
SELECT *  
FROM dept;
```

DEPT_ID	DEPT_NAME	DEPT_LOC
100	관리부	서울
200	연구부	대전
300	총무부	서울
400	영업부	인천
500	개발부	천안
402	영업2부	천안
404	영업4부	광주

7 rows selected.

# 1. 행 삽입

## ✓ 널 값을 갖는 행 삽입 예제(1)

- 암시적인 방법 : 열 목록에서 널 값을 갖는 열 이름 생략하기

```
INSERT INTO dept(dept_id, dept_name)
VALUES (405, '영업5부');
```

```
SELECT *
FROM dept
WHERE dept_id = 405;
```

DEPT_ID	DEPT_NAME	DEPT_LOC
405	영업5부	

- ✓ 기본 키인 열을 제외하고, 현재 값을 알 수 없는 열의 경우 널 값을 삽입할 수 있음

# 1. 행 삽입

## ✓ 널 값을 갖는 행 삽입 예제(2)

- 명시적인 방법 : VALUES 절에 해당 열의 값을 'NULL'로 명시함  
(문자나 날짜 스트링인 경우, NULL 대신 공백 스트링(' ')을 명시해도 됨)

```
INSERT INTO dept  
VALUES (406, '영업6부', NULL);
```

```
SELECT *  
FROM dept  
WHERE dept_id = 406;
```

DEPT_ID	DEPT_NAME	DEPT_LOC
406	영업6부	

**[비교]** VALUES 절에 NULL 값 지정 생략  
INSERT INTO dept  
VALUES (407, '영업7부');



```
INSERT INTO dept  
*  
ERROR at line 1:  
ORA-00947: not enough values
```



# 1. 행 삽입

## ★ 날짜 값을 갖는 행 삽입

- 오라클에서는 반드시 `TO_DATE` 함수를 사용해서 날짜 값을 삽입해야 함
- `SYSDATE` 함수를 사용해서 오늘 날짜를 삽입할 수 있음

# 1. 행 삽입

## ✓ SYSDATE 함수를 사용한 날짜 값 삽입 예제

```
INSERT INTO emp(emp_id, emp_name, job, mgr_id,  
                hiredate, sal, bonus, dept_id)  
VALUES (7631, '박은미', '연구직', 6361, SYSDATE, 5200, NULL, 200)
```

```
SELECT *  
FROM   emp  
WHERE  emp_id = 7631;
```

EMP_ID	EMP_NAME	JOB	DEPT_ID	SAL	BONUS	MGR_ID	HIREDATE
7631	박은미	연구직	200	5200		6361	21/03/01

# 1. 행 삽입

## ✓ TO\_DATE 함수를 사용한 날짜 값 삽입 예제

```
INSERT INTO emp  
VALUES (7696, '이혜성', '사무직', 300., 2500, NULL, 6351,  
       TO_DATE('2021/02/01', 'YYYY/MM/DD') );
```

```
SELECT *  
FROM   emp  
WHERE  emp_id = 7696;
```

EMP_ID	EMP_NAME	JOB	DEPT_ID	SAL	BONUS	MGR_ID	HIREDATE
7696	이혜성	사무직	300	2500		6351	21/02/01

# 1. 행 삽입

## ★ 대체 변수를 이용한 행 삽입

- 대체 변수를 사용하여 열의 값을 사용자로부터 받아들이어서 행을 삽입할 수 있음
- 대체 변수를 사용한 하나의 INSERT 문을 여러 번 실행시켜서 서로 다른 행을 삽입할 수 있음

# 1. 행 삽입

## ✓ 대체 변수를 이용한 행 삽입 예제

```
INSERT INTO dept(dept_id, dept_name, dept_loc)
VALUES (&department, '&department_name', '&location');
```

 입력 필요

Enter value for department:

Enter value for department\_name:

Enter value for location:

old 2: VALUES (&department, '&department\_name', '&location')

new 2: VALUES (600, '교육부', '천안')

1 row created.

```
SELECT *
FROM dept
WHERE dept_id = 600;
```

DEPT_ID	DEPT_NAME	DEPT_LOC
600	교육부	천안

# 1. 행 삽입

## ★ 다른 테이블의 행을 복사한 행 삽입

- 서브쿼리를 이용해서 INSERT 문을 작성함
- VALUES 절 대신 열의 값을 반환할 서브쿼리를 기술함
- 서브쿼리가 반환한 열 수와 INSERT 절의 열 수가 반드시 일치해야 함

# 1. 행 삽입

## ✓ 서브쿼리를 이용해서 다른 테이블의 행을 복사한 행 삽입 예제

- emp 테이블에서 100번 부서에서 근무하는 모든 사원을 복사해서 foreign\_emp 테이블에 삽입하기

```
INSERT INTO foreign_emp(emp_id, emp_name, job, sal, hiredate)
(SELECT emp_id, emp_name, job, sal, hiredate
FROM emp
WHERE dept_id = 100);
```

2 rows created.

2개 행이 생성됨

```
SELECT *
FROM foreign_emp;
```

EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8211	Tom	Engineer	3200	20/03/01
8212	James	Engineer	3000	21/03/01
8213	Mary	Engineer	2900	
6311	김대정	부장	6200	18/12/17
7872	이문정	사무직	2500	21/01/12

## 2. 행 변경

### ★ 행 변경 명령어

```
UPDATE 테이블이름  
    SET 열 이름1 = 값1 [, 열 이름2 = 값2, ...]  
    [WHERE 조건];
```

### ★ 행 변경 시 주의사항

- ① 특정 행이나 열만 수정하려면 반드시 WHERE 절을 명시해야 함
- ② WHERE 절을 생략하면, 해당 테이블에 있는 모든 행을 변경하게 됨




## 2. 행 변경

### ✓ 조건을 만족하는 특정 행의 특정 열만 변경하는 예제

- new\_emp 테이블에서 사원번호가 8622인 사원의 급여를 100만큼 증가시키기

```
SELECT * FROM tab;
```

```
UPDATE new_emp  
SET      sal = sal + 100  
WHERE    emp_id = 8622;
```



EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8622	민정희	연구원	4550	20/03/01
8329	박민수	개발자	4200	20/03/01
8541	장동아		2500	21/03/01
8732	최영민		2850	21/03/01

```
SELECT * FROM new_emp  
WHERE emp_id = 8622;
```

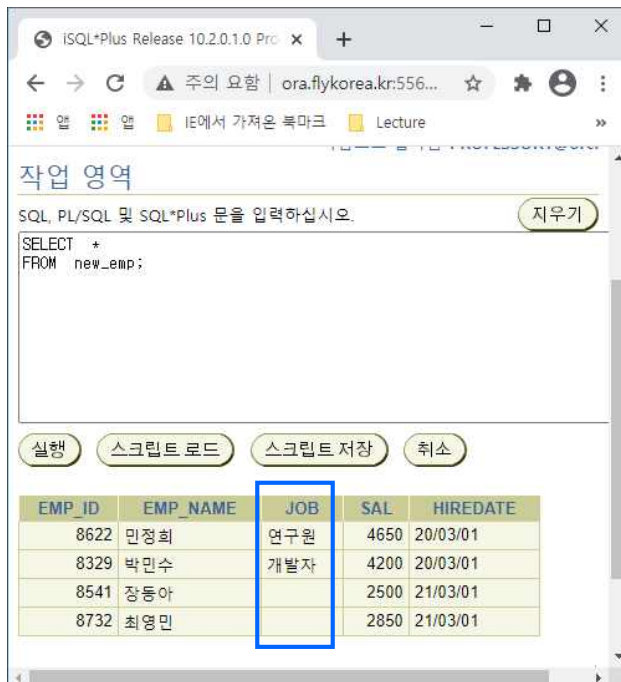
EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8622	민정희	연구원	4650	20/03/01

## 2. 행 변경

### ✓ 모든 행의 특정 열을 변경하는 예제

- new\_emp 테이블에서 업무가 정해지지 않은 모든 사원의 업무를 '사무직'으로 변경하기

```
UPDATE new_emp  
SET      job = '사무직'  
WHERE   job IS NULL ;
```

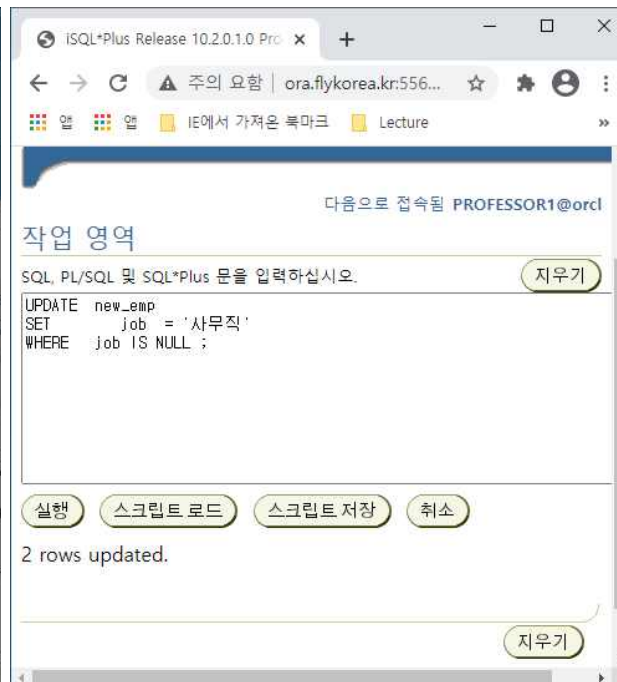


작업 영역

SQL, PL/SQL 및 SQL\*Plus 문을 입력하십시오. 지우기

실행 스크립트 로드 스크립트 저장 취소

EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8622	민정희	연구원	4650	20/03/01
8329	박민수	개발자	4200	20/03/01
8541	장동아		2500	21/03/01
8732	최영민		2850	21/03/01



작업 영역

다음으로 접속됨 PROFESSOR1@orcl

SQL, PL/SQL 및 SQL\*Plus 문을 입력하십시오. 지우기

실행 스크립트 로드 스크립트 저장 취소

2 rows updated.

지우기

EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8622	민정희	연구원	4650	20/03/01
8329	박민수	개발자	4200	20/03/01
8541	장동아	사무직	2500	21/03/01
8732	최영민	사무직	2850	21/03/01



작업 영역

SQL, PL/SQL 및 SQL\*Plus 문을 입력하십시오. 지우기

실행 스크립트 로드 스크립트 저장 취소

EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8622	민정희	연구원	4650	20/03/01
8329	박민수	개발자	4200	20/03/01
8541	장동아	사무직	2500	21/03/01
8732	최영민	사무직	2850	21/03/01

## 2. 행 변경

### ★ 다중 열 서브쿼리로 행 변경하기

- SET 절에 변경할 값을 검색하는 서브쿼리나, WHERE 절에서 조건을 비교할 값을 검색하는 서브쿼리를 기술할 수 있음

```
UPDATE 테이블이름1
    SET (열이름1, [열이름2, ...]) =
        (SELECT 열이름1, [열이름2, ...]
         FROM 테이블이름2
         WHERE 조건1)
    WHERE (열이름3, [열이름4, ...]) =
        (SELECT 열이름3, [열이름4, ...]
         FROM 테이블이름3
         WHERE 조건2);
```

### ★ 서브쿼리를 포함하는 UPDATE 명령 사용 시 주의사항

- ① SET 절의 서브쿼리가 다중 행을 반환되면 오류가 발생함
- ② WHERE 절의 서브쿼리는 다중 행을 반환해도 무관함

## 2. 행 변경

### ✓ SET 절에 서브쿼리를 사용한 행 변경 예제

- new\_emp 테이블에서 8732번 사원의 업무와 급여를 8329번 사원의 업무 및 급여와 같도록 변경하기

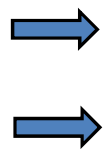
```
UPDATE new_emp
SET      (job, sal) =
          (SELECT job, sal
           FROM   new_emp
           WHERE  emp_id = 8329)
WHERE    emp_id = 8732;
```

## 2. 행 변경

### ✓ WHERE 절에 서브쿼리를 사용한 행 변경 예제

- new\_emp 테이블에서 8329번 사원과 같은 업무를 하는 모든 사원의 급여를 4500으로 변경하기

```
UPDATE new_emp
SET      sal = 4500
WHERE    job  =
        (SELECT job
         FROM  new_emp
         WHERE emp_id = 8329);
```



EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8622	민정희	연구원	4650	20/03/01
8329	박민수	개발자	4500	20/03/01
8541	장동아	사무직	2500	21/03/01
8732	최영민	개발자	4500	21/03/01

## 2. 행 변경

### ✓ 다른 테이블에 근거한 행 변경 예제

- new\_emp 테이블의 사원 가운데, emp 테이블의 7933번 사원과 같은 업무를 하는 모든 사원의 급여를 emp 테이블의 7920번 사원의 급여와 같도록 변경하기

```
UPDATE new_emp
SET      sal =
        (SELECT sal
         FROM emp
         WHERE emp_id = 7920)
WHERE    job =
        (SELECT job
         FROM emp
         WHERE emp_id = 7933);
```



EMP_ID	EMP_NAME	JOB	SAL	HIREDATE
8622	민정희	연구원	4650	20/03/01
8329	박민수	개발자	4500	20/03/01
8541	장동아	사무직	2650	21/03/01
8732	최영민	개발자	4500	21/03/01

## 2. 행 변경

### ★ 행 변경 오류

- 다른 테이블의 기본 키를 참조하는 있는 참조 키의 경우, 참조하는 테이블에 없는 값으로 변경하는 경우 "참조 무결성 제약 조건"을 위배하게 되므로 변경이 허용되지 않음

### ✓ 예제

- 100번 부서에 근무하는 모든 사원의 부서번호를 700으로 변경하기

700번 부서가 dept 테이블에 없기 때문에 오류 발생



### 3. 행 삭제

#### ★ 행 삭제 명령어

```
DELETE [FROM] 테이블이름  
[WHERE 조건];
```

#### ★ DELETE 명령 사용 시 주의사항

- ① WHERE 절을 명시하면, 그 조건을 만족하는 특정 행(들)만 삭제하게 됨
- ② WHERE 절을 생략하면, 모든 행을 삭제하게 됨



### 3. 행 삭제

#### ✓ 조건을 만족하는 행 삭제 예제

- dept 테이블에서 부서이름이 '개발부'인 모든 행을 삭제하기

```
DELETE FROM dept  
WHERE dept_name = '개발부';
```

1 row deleted.

500번 부서에 근무하는 사원이  
없기 때문에 오류가 발생하지  
않고 해당 행이 삭제됨

### 3. 행 삭제

#### ✓ 다른 테이블에 근거한 행 삭제 예제

- emp 테이블의 7720 사원과 같은 업무를 하는 모든 사원을 new\_emp 테이블에서 삭제하기

```
DELETE FROM new_emp  
WHERE job =
```

```
(SELECT job  
FROM emp  
WHERE emp_id = 7920);
```

사례 교체?

JOB
사무직

### 3. 행 삭제

#### ★ 행 삭제 오류

- 다른 테이블에서 참조 키로 사용되고 있는 열을 포함하는 행을 삭제하는 경우, "참조 무결성 제약 조건"을 위반하게 되므로 행 삭제가 허용되지 않음
- 즉, 삭제하려는 부모 레코드가 자식 레코드를 갖고 있으면, 삭제가 허용되지 않음

#### ✓ 예제

- dept 테이블에서 부서번호가 200인 부서를 삭제하기

200번 부서를 삭제하면, emp 테이블에서 200번 부서를 참조하고 있는 행에 문제가 생기므로, 참조 무결성 제약 조건을 위반하게 되어 삭제시키는 것이 불가능함





**Q & A**