

## [Database] INSERT,UPDATE,DELETE 문 정리와 다양한 예제

노트북: [TIL-MY]

만든 날짜: 2020-07-20 오후 3:33

URL: <https://continuous-development.tistory.com/30>

# 나무늘보의 개발 블로그

홈

태그

Database

[Database] INSERT,UPDATE,DELETE  
문 정리와 다양한 예제

• by 꾸까꾸 • 2020. 7. 20. • 수정 • 삭제

## INSERT

테이블에 데이터를 넣는 작업

--구문

INSERT INTO table\_name[(column\_name],[column\_

분류 전체보기 

Python

Database 

ASP.NET

Algorithm

Deep learning

```
name]])]
VALUES (value1,[value2 ...])
```

## 다양한 INSERT 하는 방법

### 1. 모든 컬럼 이름을 명시하는 방법

```
--INSERT
INSERT INTO EMPLOYEE(EMP_ID, EMP_NO, EMP_NAME, EMAIL, PHONE, HIRE_DATE, JOB_ID, SALARY, BONUS_PCT, MARRIAGE, MGR_ID, DEPT_ID)
VALUES('900','811126-1484710','오윤하','oyuh@vcc.com','01012345678','06/01/01','J7',3000000,0,'N','176','90');

SELECT *
FROM EMPLOYEE;
```

모든 컬럼 이름 명시

### 2. 컬럼을 생략하는 방법(모든 값을 입력할 때만 가능하다)

```
INSERT INTO EMPLOYEE
VALUES('910','811126-1484710','오윤하','oyuh@vcc.com','01012345678','06/01/01','J7',3000000,0,'N','176','90');
```

컬럼 이름 생략

### 3.NULL 입력하는 방법

1)암시적 방법 - INSERT INTO 절에서 해당 컬럼 이름을 생략

```
--NULL 입력하는 방법(묵시적)
INSERT INTO EMPLOYEE(EMP_ID, EMP_NO, EMP_NAME, PHONE, HIRE_DATE, JOB_ID, SALARY, BONUS_PCT, MARRIAGE)
VALUES ('880','860412-2377610','한채연','0193382662','06/01/01','J7',3000000,0,'N');
```

암시적 방법

AWS

ETC..

공지사항

글 보실 때 주의사항

: 최근글 : 인기글

[Data  
INSEF  
PHONE, HIRE\_DATE, JO  
'01012345678','06/01

2020.07.20

[Database] DDL(데이  
터 정..

2020.07.19

[Data  
테...

2020.07.18

[Data  
Sub...

2020.07.17

[Data  
SEP...

2020.07.17

최근댓글

태그

설정, 날짜함수,

trunc 함수,

select trunc,

INTERSECT함수,

## 2) 명시적 방법 - VALUES 절에서 NULL 키워드나 " 사용

```
--NULL 입력하는 방법(명시적)  
INSERT INTO EMPLOYEE VALUES  
('840','하지연','870115-2253408','ju_ha@vcc.com',NULL,'07/06/15','J7',NULL,NULL,'N','','');;
```

명시적방법

지금 현재 위에 INSERT와 비교해 보면 칼럼과 값이 생략되어 있다.

## 4.DEFAULT로 INSERT하는 방법

DEFAULT 설정된 컬럼 이름을 생략하면 DEFAULT 값이 입력됨

```
--DEFAULT  
INSERT INTO EMPLOYEE (EMP_ID,EMP_NO,EMP_NAME,SALARY,MARRIAGE)  
VALUES ('860','810429-2165344','선애진',DEFAULT,DEFAULT);  
  
SELECT *  
FROM EMPLOYEE  
WHERE EMP_ID=860;
```

스크립트 출력 x | 결과의 결과 x

SQL | 인출된 모든 행: 1 (0.005초)

EMP_ID	EMP_NAME	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY	BONUS_PCT	MARRIAGE
860	선애진	810429-2165344	(null)	(null)	20/07/20	(null)	(null)	(null)	N

DEFAULT 값을 통해 INSERT 하는 방법

## 5. 서브 쿼리로 INSERT 하는 방법

INSERT함수,  
인스턴스,  
테이블 생성,  
trunc 사용,  
DELETE함수,  
사용법,  
CRATE TABLE,  
DDL,  
UPDATE함수,  
SUBQUERY함수,  
Subquery 함수,  
Oracle,  
서브쿼리 사용법,  
DELETE 사용법,  
SUBQUERY사용,  
MINUS함수,  
Oracle SQL,  
AWS,  
IN Operator,  
INSERT 사용법,  
SQL 테이블 생성,  
INSERT UPDATE  
DELETE,  
데이터 정의어,  
UPDATE 사용법,  
SQL

전체 방문자

92

Today : 1

Yesterday : 0

<pre> INSERT INTO EMP(     SELECT EMP_ID, EMP_NAME, DEPT_NAME     FROM EMPLOYEE     LEFT OUTER JOIN DEPARTMENT USING(DEPT_ID) );  SELECT * FROM EMP; </pre>		
<div> <div>스크립트 출력 x</div> <div>질의 결과 x</div> </div> <div> <div> <div>📌</div> <div>📄</div> <div>🔍</div> <div>❌</div> </div> <div>SQL   인출된 모든 행: 27(0,004초)</div> </div>		
EMP_ID	EMP_NAME	DEPT_NAME
1 210	감우섭	회계팀
2 208	이중기	회계팀
3 207	김술오	회계팀
4 205	임영애	본사 인사팀
5 202	권상후	본사 인사팀
6 200	고승우	본사 인사팀
7 201	박하일	해외영업1팀
8 149	성해교	해외영업1팀

※INSERT와 무결성 제약조건(O일 때만 신경 쓰면 된다.)

구분	부모	자식
INSERT	X	O
UPDATE	O	O
DELETE	O	X

```
INSERT INTO EMPLOYEE(EMP_ID,EMP_NO,EMP_NAME,DEPT_ID)
VALUES('990','810116-2154219','김태휘','45');
```

스크립트 출력 x | 질의 결과 x

작업이 완료되었습니다.(0.017초)

명령의 114 행에서 시작하는 중 오류 발생 -  
INSERT INTO EMPLOYEE(EMP\_ID,EMP\_NO,EMP\_NAME,DEPT\_ID)  
VALUES('990','810116-2154219','김태휘','45')  
오류 보고 -  
ORA-02291: integrity constraint (HR.FK\_DEPTID) violated - parent key not found

지금 같은 경우에는 무결성 제약에 걸려 오류를 발생시킨다. 부모 키에서 해당 값이 없어 오류가 난다.

이렇듯 데이터를 넣을 때는 무결성 제약 조건에 맞춰서 넣어줘야 한다.

## UPDATE문

테이블에 포함된 데이터를 수정한다.

--구문

```
UPDATE      table_nameSET
column_name=value[,column=value]
[where condition]
```

#where 절이 없으면 전체 행이 갱신된다.

### 1. 기본적인 UPDATE 구조

```

UPDATE DEPARTMENT
SET DEPT_NAME='전략기획팀'
WHERE DEPT_ID='90';

SELECT *
FROM DEPARTMENT;

```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 7(0,001초)

	DEPT_ID	DEPT_NAME	LOC_ID
1	20	회계팀	A1
2	10	본사 인사팀	A1
3	50	해외영업1팀	U1
4	60	기술지원팀	OT
5	80	해외영업2팀	A2
6	90	전략기획팀	A3
7	30	마케팅팀	A1

## 2.SET 절에 서브 쿼리가 들어가는 방법

```

UPDATE EMPLOYEE
SET (JOB_ID,SALARY) = (SELECT JOB_ID,SALARY
FROM EMPLOYEE
WHERE EMP_NAME='성해교')
WHERE EMP_NAME='심하균';

SELECT *
FROM EMPLOYEE;

```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 22(0,002초)

	EMP_ID	EMP_NAME	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY
1	100	한선기	621133-1483658	sg_ahn@vcc.com	0199949999	90/04/01	J1	9000000
2	101	강종운	621136-1006405	jh_park@vcc.com	0193334433	04/04/30	J2	5500000
3	102	최만석	861011-1940062	ms_choi@vcc.com	0198879908	95/12/30	J2	3600000
4	103	정도연	631127-2519077	sy_kang@vcc.com	0196654436	97/06/03	J4	2600000
5	104	안석규	651031-1962810	sg_han@vcc.com	0192347654	98/07/01	J3	3500000
6	107	조재형	721128-1732822	jh_jeon@vcc.com	0193325548	98/11/23	J3	3800000
7	124	정지현	641231-2269080	jih_jeon@vcc.com	01922976129	04/07/15	J7	1500000
8	141	김예수	651122-2592930	hs_kim@vcc.com	0194087600	01/03/20	J5	2100000
9	143	나승원	871024-1945881	sw_cha@vcc.com	0197243979	01/03/20	J5	2300000
10	144	김순이	741122-2515789	sm_kim@vcc.com	0192213306	99/10/20	J3	3400000
11	149	성해교	640524-2148639	hg_song@vcc.com	01992882295	03/08/16	J7	1900000

### 3.UPDATE 에 DEFAULT를 사용하는 방법.(테이블을 생성할 때 DEFAULT를 썼을 때만 가능)

```

UPDATE EMPLOYEE
SET MARRIAGE = DEFAULT
WHERE EMP_ID = '210';

SELECT *
FROM EMPLOYEE
WHERE EMP_ID='210';

```

스크립트 실행 x | 결과 x

SOL | 인출된 모든 행: 1(0.002초)

P_NAME	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY	BONUS_PCT	MARRIAGE
1	700813-1766819	manofking@vcc.com	(null)	05/07/31	J4	2500000	(null)	N

### 4.UPDATE의 WHERE 절에 서브 쿼리를 사용하는 방법

```

UPDATE EMPLOYEE
SET BONUS_PCT = 0.3
WHERE DEPT_ID = (SELECT DEPT_ID
FROM DEPARTMENT
WHERE DEPT_NAME = '해외영업2팀');

SELECT *
FROM EMPLOYEE;

```

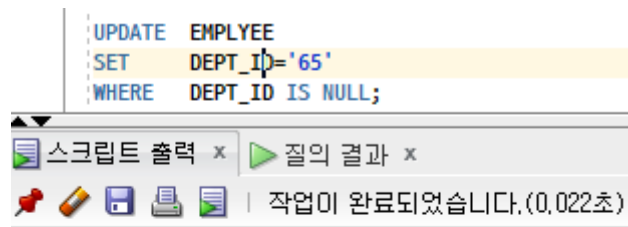
스크립트 실행 x | 결과 x

SOL | 인출된 모든 행: 22(0.002초)

EMP_ID	EMP_NO	EMAIL	PHONE	HIRE_DATE	JOB_ID	SALARY	BONUS_PCT	MARRIAGE	MGR_ID	DEF
1	621133-1483658	sg_ahn@vcc.com	0199949999	00/04/01	J1	9000000	0.2 Y	(null)	90	
2	621136-1006405	jh_park@vcc.com	0193334433	04/04/30	J2	5500000	(null) Y	100	90	
3	061011-1940062	ms_choi@vcc.com	0190879908	95/12/30	J2	3600000	(null) Y	101	90	
4	631127-2519077	sy_kang@vcc.com	0196654436	97/06/03	J4	2600000	(null) Y	104	60	
5	651031-1962810	sg_han@vcc.com	0192347654	98/07/01	J3	3500000	0.25 Y	100	60	
6	721128-1732822	jh_jo@vcc.com	0193325548	98/11/23	J3	3800000	(null) Y	104	60	
7	641231-2269080	jih_jeon@vcc.com	01922976129	04/07/15	J7	1500000	(null) N	141	50	
8	651122-2592930	hs_kim@vcc.com	0194087600	01/03/20	J5	2100000	0.1 Y	100	50	
9	871024-1945881	sw_choi@vcc.com	0197243979	01/03/20	J5	2300000	(null) Y	141	50	
10	741122-2515789	sm_kim@vcc.com	0192213306	99/10/20	J3	3400000	0.1 Y	141	50	
11	640524-2148639	hg_song@vcc.com	01992882295	03/08/16	J7	1900000	(null) N	141	50	
12	821121-1660412	ws_jeong@vcc.com	0193243388	02/07/14	J6	2090000	0.3 Y	100	80	
13	791217-2230420	jh_um@vcc.com	0194769665	04/07/21	J6	2420000	0.3 Y	174	80	

### ※UPDATE와 무결성 제약조건

데이터 무결성이 손상되는 변경 작업은 허용되지 않는다.



이미 데이터가 존재하는 상태에서 부모 키의 값을 바꾸는 것은 불가능하다. 자식 키에서 키를 잃어버리게 되기 때문이다.

## DELETE

테이블에 포함된 기존 데이터를 삭제한다. 이때 열이 아닌 행 단위의 삭제를 한다.

--DELETE 구문



DELETE FROM table\_name  
[where 조건식]

```
DELETE FROM DEPARTMENT  
WHERE LOC_ID NOT LIKE 'A%';  
  
DELETE FROM DEPARTMENT  
WHERE LOC_ID NOT LIKE 'A%';  
  
SELECT *  
FROM DEPARTMENT;  
  
DELETE FROM JOB  
WHERE JOB_ID='J2';  
  
DELETE FROM EMPLOYEE  
WHERE EMP_ID='141';
```

## 테이블 전체 데이터 삭제하는 방법

--DELETE 구문  
DELETE FROM table\_name

--TRUNCATE 구문  
TRUNCATE TABLE table\_name;

## DELETE 특징

1.TRUNCATE보다 느리다.

2.AUTO COMMIT이 되지 않아 지운 데이터를 되돌릴 수 있다.

3. 삭제 후에도 데이터 용량은 그대로이다.

# TRUNCATE 특징

1.DELETE 보다 빠르다

2.AUTO COMMIT 내장되어 있어서 지운 데이터는 다시 되돌릴 수 없다(롤백 불가능).

3.AUTO COMMIT을 하기 때문에 사용 즉시 데이터 용량이 줄어들고 인덱스 자체도 사라진다.



참조하고 있는 데이터가 있을 경우 삭제가 안된다.



## 'Database' 카테고리의 다른 글

<a href="#">[Database] INSERT,UPDATE,DELETE 문 정리와 다양한 예제</a> (0)	15:32:48
<a href="#">[Database] DDL(데이터 정의어), DML(데이터 조작어), DCL(데이터 제어어) 이란?</a> (0)	2020.07.19
<a href="#">[Database] 테이블 생성(CREATE TABLE) 및 제약조건(CONSTRAINT)</a> (0)	2020.07.18
<a href="#">[Database] Subquery(단일,다중) 사용법</a> (0)	2020.07.17
<a href="#">[Database] SEP Operator(UNION, UNION ALL, INTERSECT, MINUS) &amp; IN Operator</a> 사용법 (0)	2020.07.17

## 태그

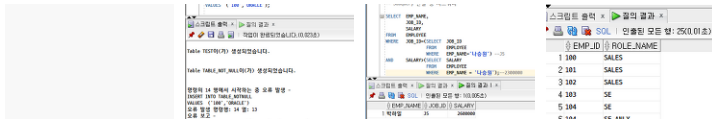
DeleteDELETE 사용법DELETE함수

INSERTINSERT UPDATE DELETE

INSERT 사용법INSERT함수Update

UPDATE 사용법UPDATE함수

## 관련글



[Database]... [Database]... [Database]... [Database]...

## 댓글 0



