

DML (DATA MANIQUULATION LANGUAGE):데이터 조작 언어

- 데이터에 값을 삽입/수정/삭제 하는 구문.

[SELECT 문]

- 5 : SELECT 컬럼명 AS 별칭, 계산식, 함수식
- 1 : FROM 참조할 테이블명
- + JOIN
- 2 : WHERE 컬럼명 | 함수식 비교연산자 비교값
- 3 : GROUP BY 그룹을 묶을 컬럼명
- 4 : HAVING 그룹함수식 비교연산자 비교값
- 6 : ORDER BY 컬럼명 | 별칭 | 컬럼순번 정렬방식 [NULLS FIRST | LAST];

[INSERT 문]

INSERT INTO 테이블명 VALUES(데이터, 데이터, ...)

INSERT INTO 테이블명(컬럼명, 컬럼명, 컬럼명,...)
VALUES (데이터1, 데이터2, 데이터3, ...)

UPDATE : 기존 내용을 수정 , 추가

- 영화/게임같은 것들을 리마스터 (기존을 보장)

UPGRADE : 등급 상승 :아예 새로운 걸 만든다

- 이전의 것을 기반으로 새로운 것

[UPDATE 문]asfh
UPDATE 테이블명 SET 컬럼명 = 바꿀값 [WHERE 컬럼명 비교연산자 비교값];
어떤 테이블을 바꾸겠다 컬럼을 무슨 값으로

DML : DATE MANIPULATION LANGUAGE (데이터 조작 언어)

- 크게 4 종료로 나뉜다
- INSERT (삽입 / 추가)
- UPDATE (수정 / 갱신)
- MERGE (병합)
- DELETE (삭제)

COMMIT ; : 커밋 전 까지 수행된 구문들을 저장한다.

ROLLBACK; : 마지막으로 커밋된 상태로 변경된다 마지막 커밋 이후 수행된 구문들은 삭제된다.

INSERT

- 테이블에 한 행을 추가하는 구문
- 방법은 두 가지 있다 '모든 값을 추가' , ' 특정 값만 추가'

INSERT INTO 테이블면 VALUES (데이터 1, 데이터 2)

```
INSERT INTO EMPLOYEE2  
VALUES (900, '김현기', '980918-1234567', 'USER@GMAIL.COM', '01099161042', 'D1', 'J7', 'S3', 4300000, 0.2, 200, SYSDATE, NULL, 'N');
```

EMPLOYEE2테이블에 모든 값을 지정해 각각 넣는다.

- 이때 전화번호 구문에서 ''(홀따옴표)로 묶지않을 경우 010에서 0이 제거된다.
- 단점은 모든 VALUE에 값을 일일이 지정해줘야된다 (장점이 될 수도 있다)

INSERT INTO 테이블명(컬럼명1, 컬럼명2, 컬럼명3....)
VALUE(데이터1, 데이터2,데이터3,)

```
INSERT INTO EMPLOYEE2 (EMP_ID,EMP_NAME,EMP_NO,EMAIL,PHONE,DEPT_CODE,JOB_CODE,SAL_LEVEL,SALARY)
VALUES(900, '장채현', '901123-2345678', 'jang_ch@kh.or.kr', '01012341234',
      'D1', 'J7', 'S3', 4300000)
```

- 테이블명을 지정한 다음 괄호 안에 특정 컬럼만 지정한 후 INSERT한다
 - 선택되지 않은 컬럼은 자동으로 NULL 값이 들어간다.

INSERT 시 VALUES 대신 서브쿼리를 사용가능하다

```
INSERT INTO EMP_01 (
  SELECT EMP_ID,EMP_NAME,DEPT_TITLE
  FROM EMPLOYEE
  LEFT JOIN DEPARTMENT ON (DEPT_CODE=DEPT_ID));
```

EMP_01 테이블에
EMPLOYEE테이블에서 조회한 EMP_ID , EMP_NAME, DEPT_TITLE을 가져와 대입한다
이때 이 값을 제외한 값들은 NULL로 대입된다.

UPDATE :테이블에 작성되어 있는 값을 수정 하는 구문

[작성법]
UPDATE 테이블명 SET 컬럼명 = 바꿀값 [WHERE 컬럼명 비교연산자 비교값];

```
UPDATE DEPARTMENT2 SET
DEPT_TITLE = '전략기획팀'
WHERE DEPT_ID = 'D9';
```

- DEPARTMENT2 테이블에서 부서코드가 D9인 DEPT_TITLE의 컬럼을 '전략 기획팀'으로

```
UPDATE EMPLOYEE2
SET BONUS = 0.1
WHERE BONUS IS NULL;
```

- EMPLOYEE2 테이블에서 BONUS가 NULL인 사람들로 BONUS의 컬럼 값을 0.1로 변경한다.
-

조건절이 없을 경우 모든 행의 컬럼 값이 바뀌기 때문에 **주의해야** 한다.
여러 컬럼을 변경하고 싶을 경우 ',' 으로 구별하면 된다.

```
UPDATE DEPARTMENT2
SET
DEPT_ID = 'D0',
DEPT_TITLE='전략기획 2팀'
WHERE DEPT_ID='D9' AND DEPT_TITLE='전략기획팀';
```

- 부서코드가 D9 이면서 부서명이 전략기획팀인 컬럼들을
- D0 , 전략기획 2팀으로 변경한다.

UPDATE에서도 서브 쿼리가 사용가능하다.

```
UPDATE EMPLOYEE2
SET
SALARY=(SELECT SALARY FROM EMPLOYEE2 WHERE EMP_NAME='유재식'),
BONUS=(SELECT BONUS FROM EMPLOYEE2 WHERE EMP_NAME='유재식')
WHERE EMP_NAME ='방명수';
```

EMPLOYEE테이블에서 EMP_NAME이 박명수인 사람의 SALARY 와 BONUS를 바꾼다
이때

- 월급은 EMPLOYEE2 테이블에서 이름이 유재식인 사람의 월급을
- 보너스는 EMPLOYEE2테이블에서 이름이 유재식인 사람의 보너스 값을

대입한다.

다중행 다중열 서브쿼리를 이용한 UPDATE문

```
UPDATE EMPLOYEE2
SET (SALARY, BONUS) = (SELECT SALARY, BONUS
                        FROM EMPLOYEE2
                        WHERE EMP_NAME = '유재식')
WHERE EMP_NAME IN ('노응철', '전형돈', '정중하', '하동운');
```

- EMPLOYEE2 테이블에서 이름이 맞는 사람들의 SALARY와BONUS를
- EMPLOYEE2 테이블에서 이름이 유재식인 사람의 월급 보너스 값으로 UPDATE한다.

MERGE : 병합

- 구조가 같은 두개의 테이블을 하나로 합친다
- 테이블에서 지정하는 조건으 ㅣ 값이 존재하면 수정(UPDATE)
- 조건의 값이 없으면 삽입(INSERT)가 된다
- 순수 노가다

DELETE :테이블에 행을 삭제하는 구문
SELECT와 작성법이 동일하다.

[작성법]

DELTE FROM 테이블명 WHERE 조건설정

**** 만약 WHERE 조건을 설정하지 않으면 모든 행이 다 삭제됨 ****

```
-- EMPLOYEE2 테이블에서 이름이 '장채현'인 사원 정보 삭제
DELETE FROM EMPLOYEE2 WHERE EMP_NAME='장채현';
```

사용시 주의

TRUNCATE (D D L)

- 데이터의 전체 행을 삭제 하는 DDL
- DELETE 보다 삭제하는 속도가 빠르다
- ROLLBACK으로 복구 할 수가 없다.