

[VII] DDL, DCL, DML

<SQL문의 종류>

DML : Data Manipulation Language ; 데이터 검색, 수정 등

INSERT : 데이터베이스 객체에 데이터를 입력

DELETE : 데이터베이스 객체에 데이터를 삭제

UPDATE : 기존에 존재하는 데이터베이스 객체 안의 데이터수정

SELECT : 데이터베이스 객체로부터 데이터를 검색

DDL : Data Definition Language ; 데이터와 그 구조를 정의

CREATE : 데이터 베이스 객체 생성

DROP : 데이터 베이스 객체를 삭제

ALTER : 기존에 존재하는 데이터베이스 객체를 다시 정의

TRUNCATE : 데이터베이스 객체 내용 삭제

DCL : Data Control Language ; 데이터베이스 사용자의 권한제어

GRANT : 데이터 베이스 객체에 권한 부여

REVOKE : 이미 부여된 데이터베이스 객체의 권한을 취소

COMMIT : 트랜잭션 확정 (TCL)

ROLLBACK : 트랜잭션 취소 (TCL)

SAVEPOINT : 복귀지점 설정 (TCL)

<DDL>

CREATE TABLE 로 테이블 구조 정의

데이터형 : 데이터베이스에는 문자, 숫자, 날짜, 이미지 등과 같은 다양한 형태의 데이터가 저장됩니다. 다음은 오라클에서 제공되는 데이터 형의 종류입니다.

CHAR(size) ; 고정 길이 문자 데이터. VARCHAR2와 동일한 형태의 자료를 저장할 수 있고, 입력된 자료의 길이와는 상관없이 정해진 길이만큼 저장 영역 차지. 최소 크기는 1

VARCHAR2(size) ; Up to 4000 Bytes 가변 길이 문자 데이터. 실제 입력된 문자열의 길이만큼 저장 영역을 차지. 최대 크기는 명시해야 하며, 최소 크기는 1

NUMBER ; Internal Number Format 최고 40자리까지의 숫자를 저장할 수 있습니다. 이때 소수점이나 부호는 길이에 포함되지 않는다

NUMBER(w) ; W자리까지의 수치로 최대 38자리까지 가능하다. (38자리가 유효 숫자이다.)

NUMBER(w, d) ; W는 전체 길이, d는 소수점 이하 자릿수이다. 소수점은 자릿수에 포함되지 않는다.

DATE : BC 4712년 1월 1일~AD 4712년 12월 31일까지의 날짜

LOB : 2GB까지의 가변 길이 바이너리 데이터를 저장시킬 수 있습니다. 이미지 문서, 실행 파일을 저장할 수 있습니다

LOB(Large Object) 데이터 형은 텍스트, 그래픽 이미지, 동영상, 사운드와 같이 구조화되지 않은 대용량의 텍스트나 멀티미디어 데이터를 저장하기 위한 데이터 형입니다.

최대 4GB 까지 저장가능합니다. 오라클에서 제공되는 LOB 데이터 형은 BLOB, CLOB, NCLOB, BFILE 등이 있습니다.

BLOB는 그래픽 이미지, 동영상, 사운드와 같은 구조화되지 않은 데이터를 저장하기 위해 사용됩니다.

CLOB는 e-BOOK과 같은 대용량의 텍스트 데이터를 저장하기 위해서 사용합니다.

NCLOB은 국가별 문자셋 데이터를 저장하고, BFILE은 바이너리 데이터를 파일 형태로 저장합니다

DROP TABLE로 테이블 구조 삭제하기

- DROP TABLE *table_name*;

테이블의 모든 로우를 제거하는 TRUNCATE

- TRUNCATE TABLE *table_name*

테이블 명을 변경하는 RENAME

- RENAME *old_name* TO *new_name*

<DML>

INSERT INTO table_name (column_name, ..) VALUES (column_values, ...);

DEPT01 테이블에 칼럼 DEPTNO에는 10을 칼럼 DNAME에는 'ACCOUNTING'을, 칼럼 LOC에는 'NEW YORK'을 추가합니다. 컬럼명에 기술된 목록의 수와 VALUES 다음에 나오는 괄호에 기술한 값의 개수가 맞지 않으면 에러 납니다. 컬럼명이 잘못 되었을때나 입력할 값의 데이터 타입이 맞지 않아도 에러를 발생합니다.

```
➤ INSERT INTO DEPT01 (DEPTNO, DNAME, LOC) VALUES (10, 'ACCOUNTING','NEW YORK');
```

컬럼명을 생략한 채 테이블이 갖은 모든 컬럼에 데이터를 추가해 봅니다

INSERT INTO DEPT01 VALUES (10, 'ACCOUNTING','NEW YORK');

<NULL 값을 삽입하는 다양한 방법>

데이터를 입력하는 시점에서 해당 칼럼 값을 모르거나 확정되지 않았을 경우 NULL값을 입력해야 한다.

NULL 값 삽입은 명시적인 방법과 암시적인 방법이 있다.

명시적인 방법은 VALUES 리스트에 명시적으로 NULL을 입력한다.

암시적인 방법은 칼럼 명 리스트에 칼럼 을 생각하는 것이다. 즉, 다른 칼럼은 입력하지만 이렇게 생략한 칼럼에는 암시적으로 NULL값이 할당된다

NOT NULL 제약조건이 지정된 DEPTNO 컬럼은 널 값을 입력할 수 없다.

<서브쿼리로 데이터 삽입>

INSERT INTO 다음에 VALUES 절을 사용하는 대신에 서브 쿼리를 사용할 수 있다.
이렇게 하면 기존의 테이블에 있던 여러 행을 복사해서 다른 테이블에 삽입할 수 있다.
이때 주의할 점은 INSERT 명령문에서 지정한 컬럼의 개수나 데이터타입이
서브쿼리를 수행한 결과와 동일해야 한다는 점이다.

```
DROP TABLE DEPT02;
```

```
CREATE TABLE DEPT02 AS SELECT * FROM DEPT WHERE 1=0;
```

```
INSERT INTO DEPT02 SELECT * FROM DEPT;
```

<UPDATE>

UPDATE문은 테이블에 저장된 데이터를 수정하기 위해서 사용한다.

UPDATE 문은 기존의 행을 수정하는 것이다.

따라서 어떤 행의 데이터를 수정하는지 WHERE절을 이용하여 조건을 지정한다.

WHERE 절을 사용하지 않을 경우 테이블에 있는 모든 행이 수정된다.

정말 테이블 전체 행을 수정하려고 했던 것이 아니라면 큰 문제가 발생하므로

WHERE 절의 사용 유무를 신중히 판단하여야 한다.

<DELETE>

DELETE 문으로 부서 테이블의 모든 행을 삭제

```
DELETE FROM DEPT01;
```

부서 테이블에서 30번 부서만 삭제

```
DELETE FROM DEPT01 WHERE DEPTNO=30;
```

<테이블의 제약조건>

; DBMS에 부적합한 데이터가 테이블에 삽입되는 것을 방지하기 위해 CONSTRAINT를 사용

- 테이블에서 행이 삽입, 갱신, 삭제될 때마다 테이블에 설정된 규칙을 적용
- 다른 테이블에 종속성이 있다면 테이블의 제거를 방지

; 데이터 무결성 제약조건의 종류

- **NOT NULL** : 이 필드(컬럼, 열)은 NULL값을 포함하지 않음을 지정
- **UNIQUE** : 테이블의 모든 행에 대해 유일해야 하는 값을 가진 열 또는 열의 조합을 지정
- **PRIMARY KEY** : 유일하게 테이블의 각 행을 식별
- **FOREIGN KEY** : 열과 참조된 테이블의 열 사이의 외래키 관계를 적용하고 설정
REFERENCES 예약어 사용: 참조 무결성 제약 조건 지정
- **CHECK** : 참이어야 하는 조건을 지정
- 제약조건은 아니나 쓸 수 있는 것 : **DEFAULT**