

# MySQL & Maria DB

테이블 구조 생성, 변경 및 삭제하는 DDL

# 목차

- ▶ CREATE TABLE로 테이블 구조 정의하기
- ▶ ALTER TABLE로 테이블 구조 변경하기
- ▶ DROP TABLE로 테이블 구조 삭제하기
- ▶ 테이블의 모든 로우를 제거하는 TRUNCATE
- ▶ 테이블 명을 변경하는 RENAME

# CREATE TABLE로 테이블 구조 정의하기

- ▶ CREATE TABLE로 테이블 구조 정의하기
  - ▶ 지금까지는 오라클에서 학습용으로 제공해 주는 테이블을 사용하여 다양한 조회를 해 보았습니다.
  - ▶ 이번 장에서는 DDL(Data Definition Language)을 사용하여 테이블 구조 자체를 새롭게 생성, 수정, 삭제해 보겠습니다.
  - ▶ 우선 CREATE TABLE 명령어로 새로운 테이블을 생성해 보겠습니다.
  - ▶ 다음은 CREATE TABLE 문의 기본 형식입니다.

```
CREATE TABLE table_name  
(column_name, data_type expr, ...);
```

# CREATE TABLE로 테이블 구조 정의하기

- ▶ CREATE TABLE로 테이블 구조 정의하기
  - ▶ 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원이름, 급여 3개의 칼럼으로 구성된 EMP01 테이블을 생성해 봅시다.
  - ▶ CREATE TABLE 명령어로 EMP01 테이블을 새롭게 생성합니다.

```
CREATE TABLE EMP01(  
  EMPNO DECIMAL(4),  
  ENAME VARCHAR(20),  
  SAL DECIMAL(7,2));
```

	Field	Type	Null	Key	Default	Extra
1	EMPNO	decimal(4,0)	YES		<null>	
2	ENAME	varchar(20)	YES		<null>	
3	SAL	decimal(7,2)	YES		<null>	

# CREATE TABLE로 테이블 구조 정의하기

- ▶ CREATE TABLE로 테이블 구조 정의하기
  - ▶ CREATE TABLE 문에서 서브 쿼리를 사용하여 이미 존재하는 테이블과 동일한 구조와 내용을 갖는 새로운 테이블을 생성할 수 있습니다.
  - ▶ CREATE TABLE 명령어 다음에 컬럼을 일일이 정의하는 대신 AS 절을 추가하여 EMP 테이블과 동일한 내용과 구조를 갖는 EMP02 테이블을 생성해 봅시다.

```
CREATE TABLE EMP02  
AS  
SELECT * FROM EMP;
```

# CREATE TABLE로 테이블 구조 정의하기

- ▶ CREATE TABLE로 테이블 구조 정의하기
  - ▶ 기존 테이블에서 원하는 컬럼만 선택적으로 복사해서 생성할 수도 있습니다.
  - ▶ 서브 쿼리문의 SELECT 절에 \* 대신 원하는 컬럼명을 명시하면 기존 테이블에서 일부의 컬럼만 복사할 수 있습니다.

```
CREATE TABLE EMP03  
AS  
SELECT EMPNO, ENAME FROM EMP;
```

- ▶ 돌발퀴즈 1 : EMP 테이블을 복사하되 사원번호, 사원이름, 급여 컬럼으로 구성된 테이블을 생성하시오.(테이블의 이름은 EMP04 로 하시오.)

# CREATE TABLE로 테이블 구조 정의하기

- ▶ CREATE TABLE로 테이블 구조 정의하기
  - ▶ 기존 테이블에서 원하는 행만 선택적으로 복사해서 생성할 수도 있습니다.
  - ▶ 서브 쿼리문의 SELECT 문을 구성할 때 WHERE 절을 추가하여 원하는 조건을 제시하면 기존 테이블에서 일부의 행만 복사합니다.

```
CREATE TABLE EMP05  
AS  
SELECT * FROM EMP  
WHERE DEPTNO=10;
```

# CREATE TABLE로 테이블 구조 정의하기

## ▶ 테이블의 구조만 복사하기

- ▶ 서브 쿼리를 이용하여 테이블을 복사하되 데이터는 복사하지 않고 기존 테이블의 구조만 복사하는 것을 살펴봅시다.
- ▶ 테이블의 구조만 복사하는 것은 별도의 명령이 있는 것이 아닙니다. 이 역시 서브 쿼리를 이용해야 하는데 WHERE 조건 절에 항상 거짓이 되는 조건을 지정하게 되면 테이블에서 얻어질 수 있는 로우가 없게 되므로 빈 테이블이 생성되게 됩니다.

```
CREATE TABLE EMP06  
AS  
SELECT * FROM EMP WHERE 1=0;
```

- ▶ WHERE 1=0; 조건은 항상 거짓입니다. 이를 이용하여 테이블의 데이터는 가져오지 않고 구조만 복사하게 됩니다.
- ▶ 돌발퀴즈 1 : DEPT 테이블과 동일한 구조의 빈 테이블 생성하기 생성하시오.(테이블의 이름은 DEPT02로 하시오.)



# 테이블 구조 변경하는 ALTER TABLE

## ▶ 테이블 구조 변경하는 ALTER TABLE

- ▶ ALTER TABLE 명령문은 기존 테이블의 구조를 변경하기 위한 DDL 명령문입니다. 테이블에 대한 구조 변경은 컬럼의 추가, 삭제, 컬럼의 타입이나 길이를 변경할 때 사용합니다. 테이블의 구조를 변경하게 되면 기존에 저장되어 있던 데이터에 영향을 주게 됩니다.
- ▶ ALTER TABLE로 칼럼 추가, 수정, 삭제하기 위해서는 다음과 같은 명령어를 사용합니다.
  - ▶ ADD COLUMN 절을 사용하여 새로운 칼럼을 추가한다.
  - ▶ MODIFY COLUMN 절을 사용하여 기존 칼럼을 수정한다.
  - ▶ DROP COLUMN 절을 사용하여 기존 칼럼을 삭제한다.

# 테이블 구조 변경하는 ALTER TABLE

## ▶ 새로운 컬럼 추가하기

- ▶ ALTER TABLE ADD 문은 기존 테이블에 새로운 컬럼을 추가합니다.
- ▶ 새로운 컬럼은 테이블 맨 마지막에 추가되므로 자신이 원하는 위치에 만들어 넣을 수 없습니다.
- ▶ 또한 이미 이전에 추가해 놓은 로우가 존재한다면 그 로우에도 컬럼이 추가되지만, 컬럼 값은 NULL 값으로 입력됩니다.

```
ALTER TABLE table_name  
ADD (column_name, data_type expr, ...);
```

# 테이블 구조 변경하는 ALTER TABLE

- ▶ 새로운 칼럼 추가하기
  - ▶ EMP01 테이블에 문자 타입의 직급(JOB) 칼럼을 추가해 봅시다.

```
ALTER TABLE EMP01  
ADD JOB VARCHAR(9);
```

# 테이블 구조 변경하는 ALTER TABLE

- ▶ 기존 컬럼 속성 변경하기

- ▶ ALTER TABLE MODIFY 문을 다음과 같은 형식으로 사용하면 테이블에 이미 존재하는 컬럼을 변경할 수 있게 됩니다.

```
ALTER TABLE table_name  
MODIFY (column_name, data_type expr, ...);
```

- ▶ 컬럼을 변경한다는 것은 컬럼에 대해서 데이터 타입이나 크기, 기본 값들을 변경한다는 의미입니다.

# 테이블 구조 변경하는 ALTER TABLE

- ▶ 기존 칼럼 속성 변경하기

- ▶ 직급(JOB) 칼럼을 최대 30글자까지 저장할 수 있게 변경해 보도록 하자.

```
ALTER TABLE EMP01  
MODIFY JOB VARCHAR(30);
```

# 테이블 구조 변경하는 ALTER TABLE

## ▶ 기존 칼럼 삭제

- ▶ 테이블에 이미 존재하는 컬럼을 삭제해 봅시다.
- ▶ ALTER TABLE ~ DROP COLUMN 명령어로 칼럼을 삭제할 수 있습니다.

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

# 테이블 구조 변경하는 ALTER TABLE

- ▶ 기존 칼럼 삭제
  - ▶ EMP01 테이블의 직급 칼럼을 삭제해 보도록 합시다.

```
ALTER TABLE EMP01  
DROP COLUMN JOB;
```

# 테이블 구조 삭제하는 DROP TABLE

- ▶ 테이블 구조 삭제하는 DROP TABLE

- ▶ DROP TABLE문은 기존 테이블을 제거합니다.

```
DROP TABLE table_name;
```

- ▶ CREATE TABLE을 학습할 때 만들어 놓았던 EMP01 테이블을 삭제해 봅시다.

```
DROP TABLE EMP01;
```



# 테이블의 모든 로우를 제거하는 TRUNCATE

- ▶ 테이블의 모든 로우를 제거하는 TRUNCATE
  - ▶ 기존에 사용하던 테이블의 모든 로우를 제거하기 위한 명령어로 TRUNCATE가 제공됩니다.

```
TRUNCATE table_name
```

- ▶ 테이블 EMP02에 저장된 데이터를 확인하였으면 이번에는 테이블의 모든 로우를 제거해 보도록 하겠습니다.

```
TRUNCATE TABLE EMP02;
```

# 테이블 명을 변경하는 RENAME

- ▶ 테이블 명을 변경하는 RENAME

- ▶ 기존에 사용하던 테이블의 이름을 변경하기 위한 명령어로 RENAME이 제공됩니다.

```
RENAME TABLE old_name TO new_name
```

- ▶ EMP02 테이블의 이름을 TEST 란 이름으로 변경합시다.

```
RENAME TABLE EMP02 TO TEST;
```