

# 학습목표

1. SELECT INTO
2. INSERT INTO SELECT
3. CASE ... WHEN ... END

# 학습내용

- SELECT INTO를 배울 수 있습니다.
- INSERT INTO SELECT를 배울 수 있습니다.
- CASE ... WHEN ... END를 배울 수 있습니다.

# 사전퀴즈

1. SELECT INTO(CREATE TABLE SELECT ... INTO)는 쿼리결과를 테이블로 생성하고 싶을 때 사용한다.

정답 : O

해설 : SELECT INTO(CREATE TABLE SELECT ... INTO)는 쿼리결과를 새로운 테이블을 생성해 저장한다.

2. INSERT INTO SELECT는 쿼리 결과를 기존의 테이블에 추가하고 싶을 때 사용한다.

정답 : O

해설 : INSERT INTO SELECT는 테이블의 쿼리 결과를 존재하는 테이블에(새로운 테이블 생성하지 않고) 넣고 싶을 때 사용한다.

# 수업

## 1. SELECT INTO

- 기존에 존재하지 않는 테이블이 새로 생성된다.

(일종의 뷰와 동일한 효과를 나타내지만 쿼리결과를 물리적인 새 테이블로 만든다는 점에서 차이가 있다)

- **MariaDB VS. ORACLE VS. SQL server**

**MariaDB** : CREATE TABLE 테이블명 SELECT \* FROM 테이블명;

**ORACLE** : CREATE TABLE 테이블명 **AS** SELECT \* FROM 테이블명;

**SQL server** : SELECT \* INTO 테이블명 FROM 테이블명;

### 예제 1-1

city 테이블의 내용에서 국가코드가 'KOR'인 도시를 찾아 city\_new 테이블에 넣으시오.

```
1 create table city_new
2 select *
3 from city
4 where CountryCode = 'KOR'
5 ;
6
7 select *
8 from city_new
9 ;
10
```

ID	Name	CountryCode	District	Population
2,331	Seoul	KOR	Seoul	9,981,619
2,332	Pusan	KOR	Pusan	3,804,522
2,333	Inchon	KOR	Inchon	2,559,424
2,334	Taegu	KOR	Taegu	2,548,568
2,335	Taejon	KOR	Taejon	1,425,835
2,336	Kwangju	KOR	Kwangju	1,368,341

## 2. INSERT INTO SELECT

- 쿼리결과를 기존의 테이블에 추가한다. (기존 테이블이 존재해야 함)
- INSERT INTO 테이블명1 SELECT \* FROM 테이블명2 WHERE 조건절 ...
- 두 개의 별도 쿼리(INSERT, SELECT)를 하나로 합친 것과 같음
- SELECT하는 테이블과 INSERT 하는 테이블은 동일한 구조를 가져야 함

### 예제 2-1

city 테이블의 내용에서 국가코드가 'KOR'인 도시를 찾아 이미 존재하고 있는 city\_kor 테이블에 넣으시오.

1. city 테이블을 생성할 때 사용했던 create table script 확인

\* 참고 \*

**테이블 생성할 때 사용했던 create table script 보는 방법**

**MariaDB** : SHOW CREATE TABLE 테이블명;

**ORACLE** : SELECT DBMS\_METADATA.GET\_DDL('TABLE', '테이블명') FROM dual;

2. city\_kor 테이블 생성

3. city 테이블의 내용에서 국가코드가 'KOR'인 도시를 찾아 city\_kor 테이블에 넣기

4. city\_kor 테이블 데이터 확인

```
11 show create table city; ----- 1
12
13 CREATE TABLE `city_kor` ( ----- 2
14   `ID` int(11) NOT NULL AUTO_INCREMENT,
15   `Name` char(35) NOT NULL DEFAULT '',
16   `CountryCode` char(3) NOT NULL DEFAULT '',
17   `District` char(20) NOT NULL DEFAULT '',
18   `Population` int(11) NOT NULL DEFAULT 0,
19   PRIMARY KEY (`ID`)
20 );
21
22 insert into city_kor ----- 3
23 select *
24 from city
25 where CountryCode = 'KOR'
26 ;
27
28 select * ----- 4
29 from city_kor
30 ;
31
```

city_kor (5×70)				
ID	Name	CountryCode	District	Population
2,331	Seoul	KOR	Seoul	9,981,619
2,332	Pusan	KOR	Pusan	3,804,522
2,333	Inchon	KOR	Inchon	2,559,424
2,334	Taegu	KOR	Taegu	2,548,568
2,335	Taejon	KOR	Taejon	1,425,835
2,336	Kwangju	KOR	Kwangju	1,368,341
2,337	Ulsan	KOR	Kyongsangnam	1,084,891

### 3. CASE ... WHEN ... END

- SQL의 조건문(if/switch문)에 해당

- 조건값에 따른 처리를 구분할 수 있다.

CASE WHEN 조건값1 THEN ...

```
    WHEN 조건값2 THEN ...  
    ELSE ...  
END
```

### 예제 3-1

city 테이블에서 도시명(Name)이 3자가 넘어가는 경우에 앞쪽 세 자를 대문자로 출력하고, 도시의 인구를 같이 표시하시오.

\* 참고 \*

**MariaDB에서 MID대신 SUBSTR, SUBSTRING도 사용가능**

**MID / SUBSTR / SUBSTRING (문자열, m[, n])**

문자열 중 **m** 위치에서 **n** 개의 문자 길이에 해당하는 문자를 돌려준다. n이 생략되면 마지막 문자까지이다.

**LENGTH(문자열)**

문자열의 개수를 숫자값으로 돌려준다.

```
33 select  
34     case when length(Name) >= 3 then upper(mid(Name,1,3))  
35         else upper(Name)  
36     end as upper_name, Population  
37 from city  
38 ;
```

city (2×4,079)	
upper_name	Population
KAB	1,780,000
QAN	237,500
HER	186,800
MAZ	127,800
AMS	731,200
ROT	593,321
HAA	440,900
UTR	234,323
EIN	201,843
TIL	193,238
GRO	172,701
BRE	160,398
APE	153,491

## SELECT INTO (CREATE TABLE SELECT ...) 문과 INSERT INTO SELECT ... 의 차이점?

전문가의견

CREATE TABLE SELECT ... 는 쿼리의 결과를 가지고 새로운 테이블을 생성하는 것입니다.

뷰(view)를 만드는 것과 유사하지만 뷰는 임시테이블인 점이 다릅니다.

INSERT INTO SELECT ...는 쿼리의 결과를 가지고 이미 존재하는 테이블에 넣어줍니다.

만약 존재하지 않으면 먼저 테이블을 생성해줘야 합니다.