

02

## CHAPTER

# 데이터베이스 전체 운영 맛보기



# 1-2 데이터베이스 모델링과 필수 용어

## ■ 데이터베이스 모델링

- 현실 세계에서 사용되는 데이터를 MySQL에 어떻게 옮겨놓을지 결정하는 과정
- 예 : 쇼핑몰 데이터베이스(그림 2-1)

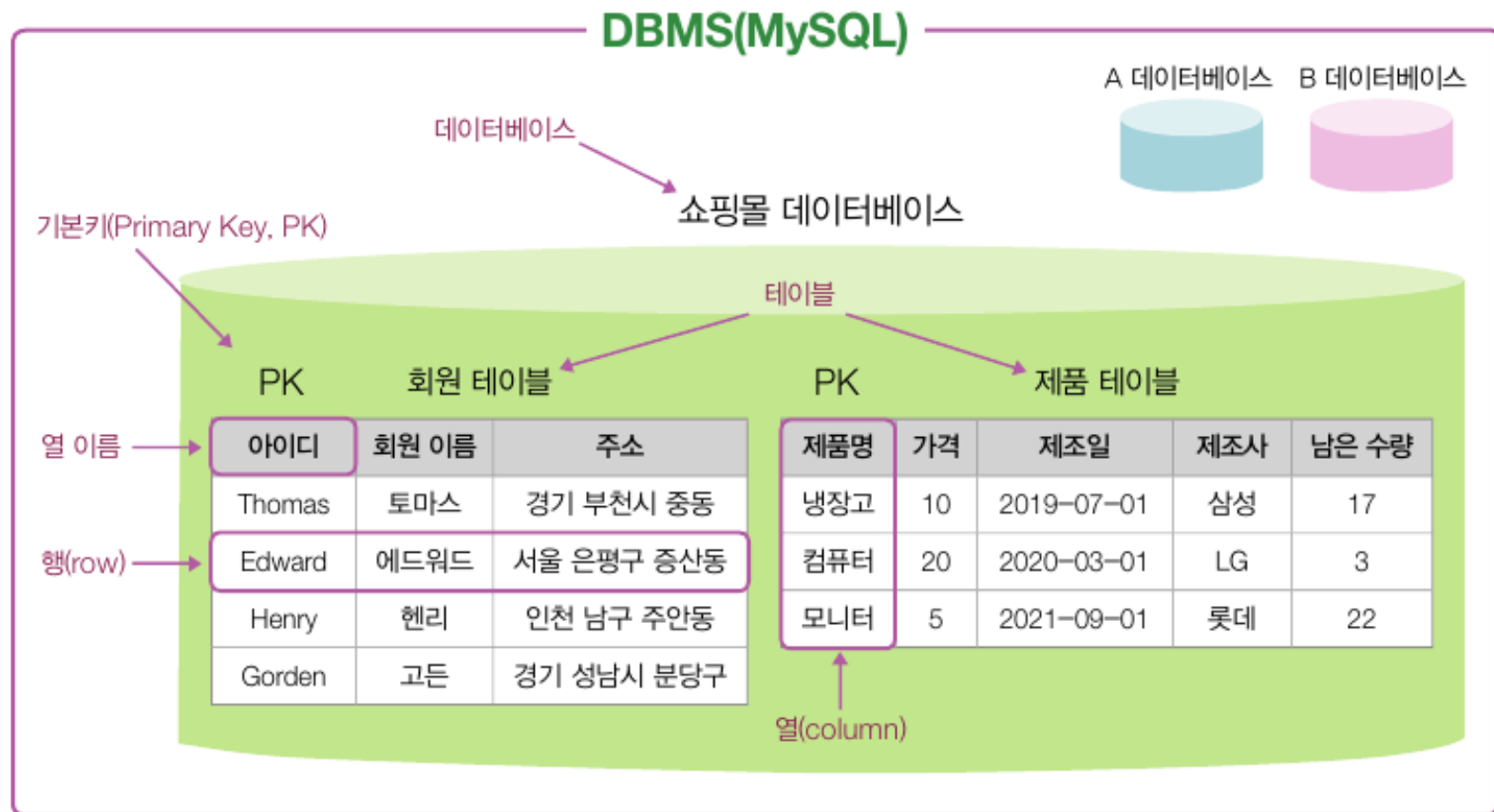


그림 2-1 테이블의 구조와 관련 용어

# 1-2 데이터베이스 모델링과 필수 용어

## ■ 데이터베이스 필수 용어

- **데이터** : 테이블에 저장된 하나하나의 단편적인 정보
- **테이블** : 데이터를 입력하기 위해 표 형태로 만든 것
- **데이터베이스** : 테이블이 저장되는 저장소로, 원통 모양으로 표현
- **DBMS** : DataBase Management System의 약자로, 데이터베이스를 관리하는 시스템 또는 소프트웨어
- **열(필드)** : 각 테이블을 열로 구성
- **열 이름** : 각 열을 구분하기 위한 이름
- **데이터 형식** : 열의 데이터 형식
- **행(레코드)** : 실질적인 데이터
- **기본키(주키)** : 각 행을 구분하는 유일한 열로, 기본키는 중복되어서도 비어 있어서도 안 됨
- **외래키** : 두 테이블의 관계를 맺어주는 키
- **SQL(구조화된 질의 언어)** : 사람과 DBMS가 소통하기 위한 말(언어)

## 2-1 데이터베이스 구축 절차 요약

### ■ 데이터베이스 구축 절차

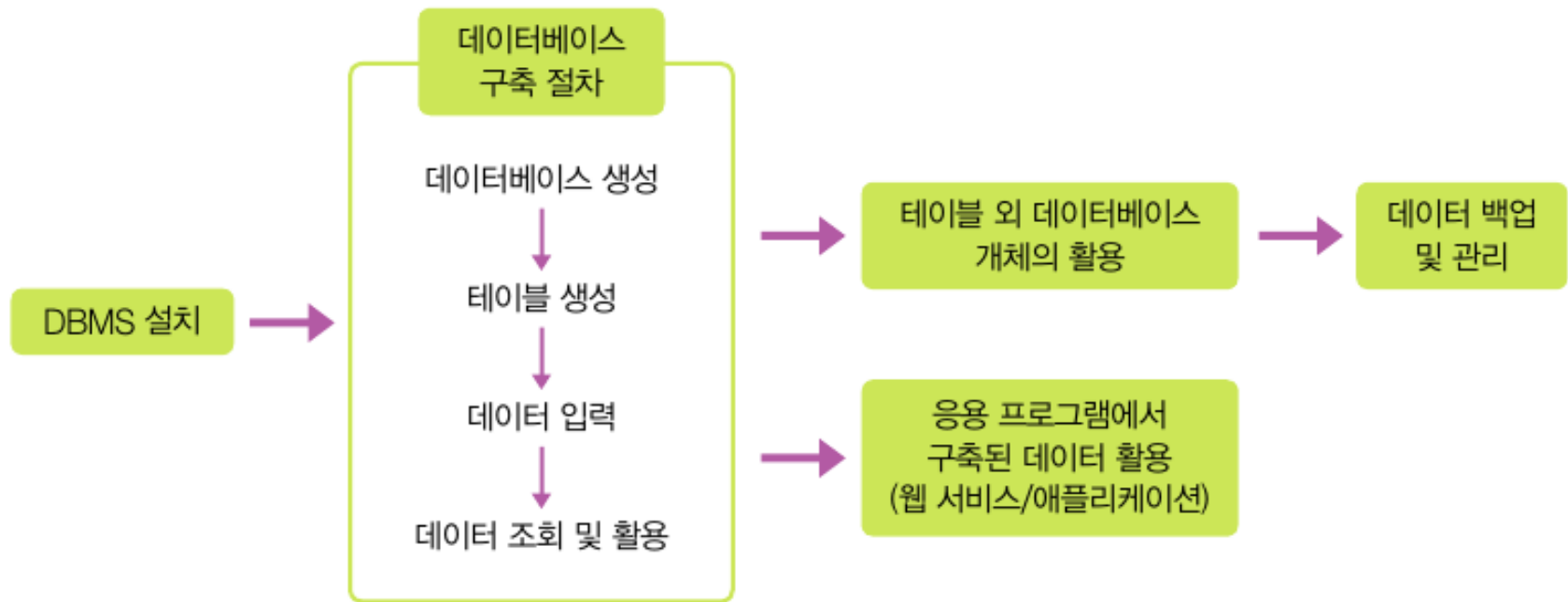


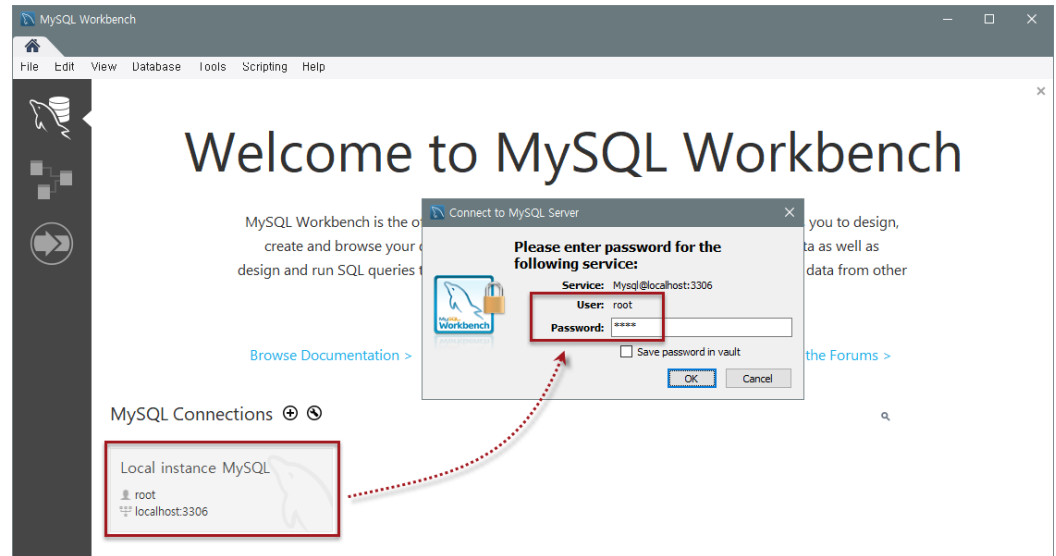
그림 2-2 데이터베이스 구축/관리 및 활용의 전반적인 절차

# [실습 2-1] 쇼핑몰 데이터베이스(shopDB) 생성하기

## 1 Workbench 실행하기

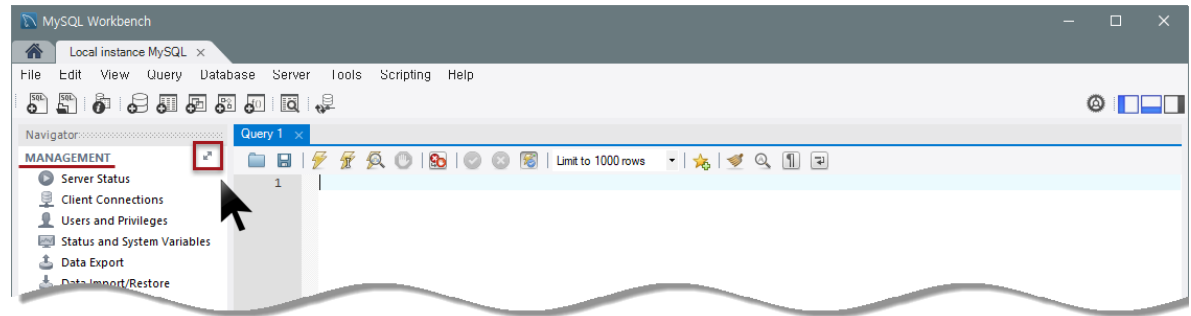
### 1-1 Workbench 실행

### 1-2 MySQL 서버에 연결

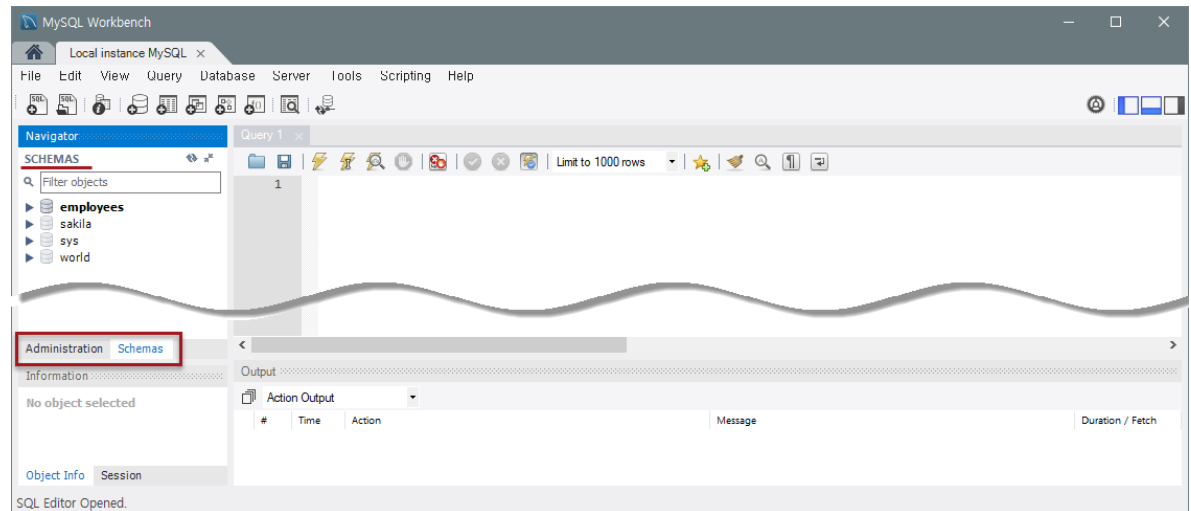


# [실습 2-1] 쇼핑몰 데이터베이스(shopDB) 생성하기

1-3 [Navigator]를 탭에 있는  
확대/축소 아이콘 클릭



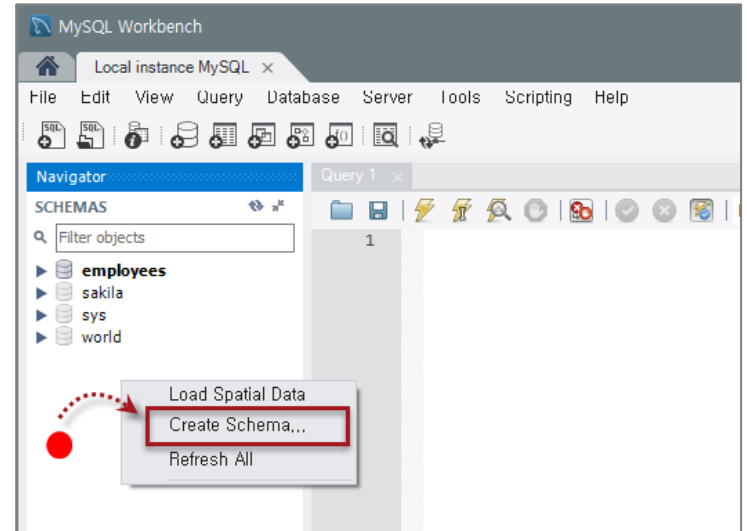
1-4 [SCHEMAS] 탭 클릭



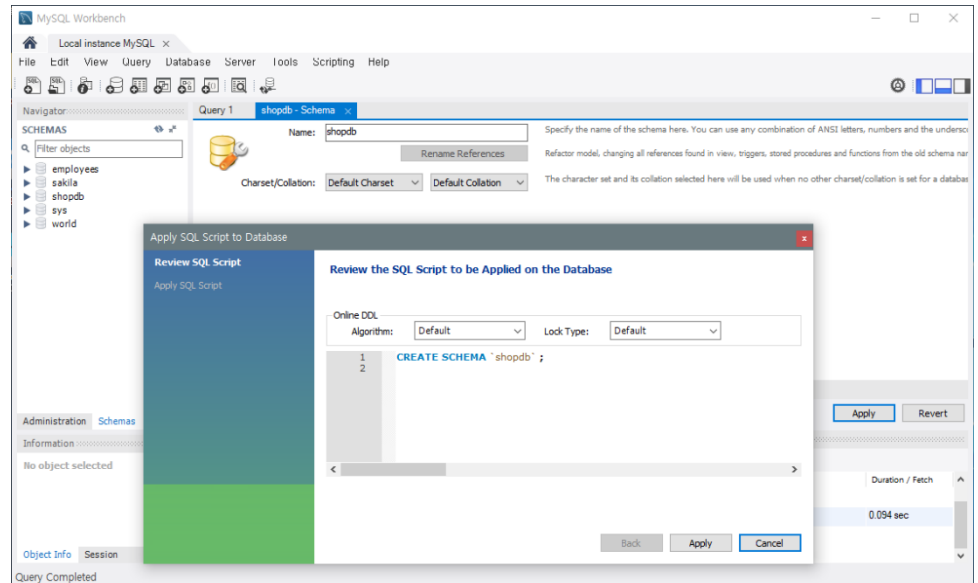
# [실습 2-1] 쇼핑몰 데이터베이스(shopDB) 생성하기

## 2 쇼핑몰 데이터베이스(ShopDB) 생성하기

### 2-1 데이터베이스 생성

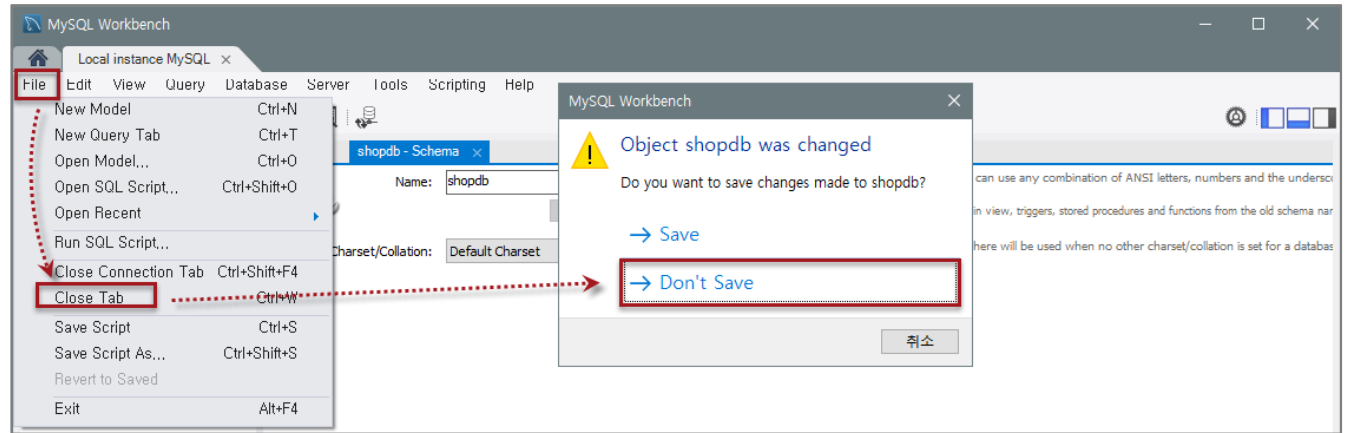


### 2-2 ShopDB 데이터베이스(스키마) 추가



# [실습 2-1] 쇼핑몰 데이터베이스(shopDB) 생성하기

## 2-3 탭 닫기





## [실습 2-2] 테이블 생성하기

### 1 개체 이름 정하기

1-1 [그림 2-1]에는 나타나 있지 않은 각 열의 영문 이름과 데이터 형식을 결정해야 함

1-2 회원 테이블(memberTBL)의 데이터 형식 지정

표 2-1 회원 테이블 정의

열 이름(한글)	영문 이름	데이터 형식	길이	NULL 허용
아이디	memberID	문자(CHAR)	8글자(영문)	×
회원 이름	memberName	문자(CHAR)	5글자(한글)	×
주소	memberAddress	문자(CHAR)	20글자(한글)	○

1-3 제품 테이블(productTBL)의 데이터 형식 지정

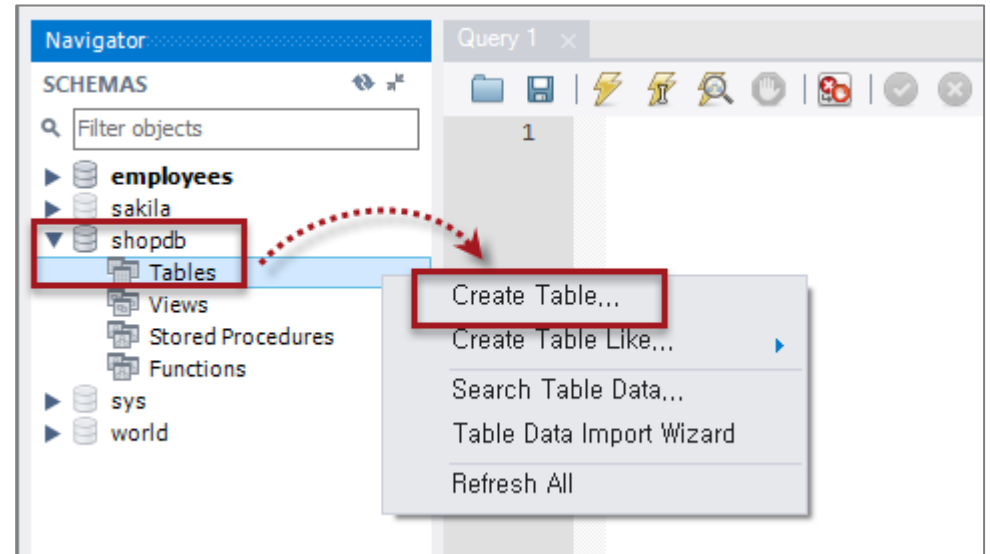
표 2-2 제품 테이블 정의

열 이름(한글)	영문 이름	데이터 형식	길이	NULL 허용
제품명	productName	문자(CHAR)	4글자(한글)	×
가격	cost	숫자(INT)	정수	×
제조일	makeDate	날짜(DATE)	날짜형	○
제조사	company	문자(CHAR)	5글자(한글)	○
남은 수량	amount	숫자(INT)	정수	×

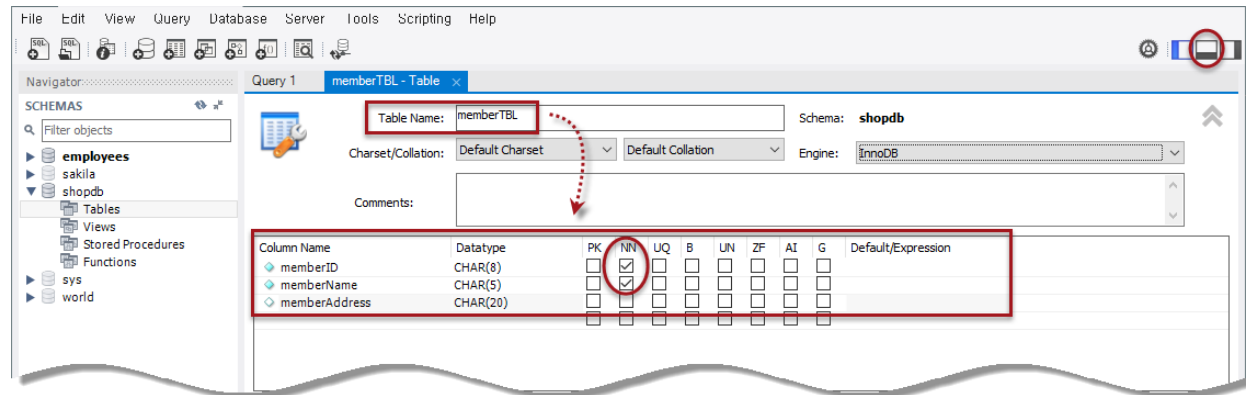
## [실습 2-2] 테이블 생성하기

## 2 회원 테이블(memberTBL) 만들기

## 2-1 [Create Table] 선택



## 2-2 회원 테이블 내용 입력

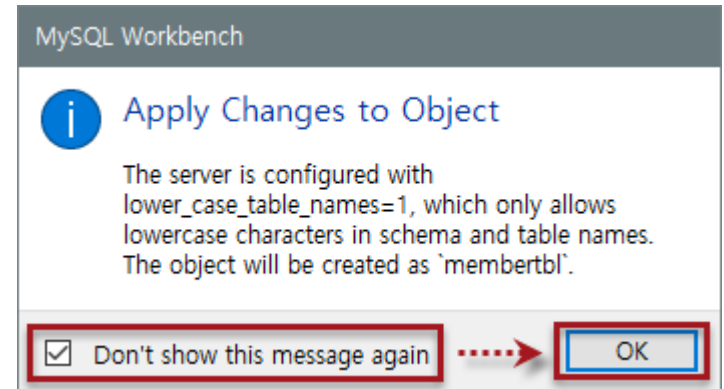


# [실습 2-2] 테이블 생성하기

## 2-3 기본키 지정

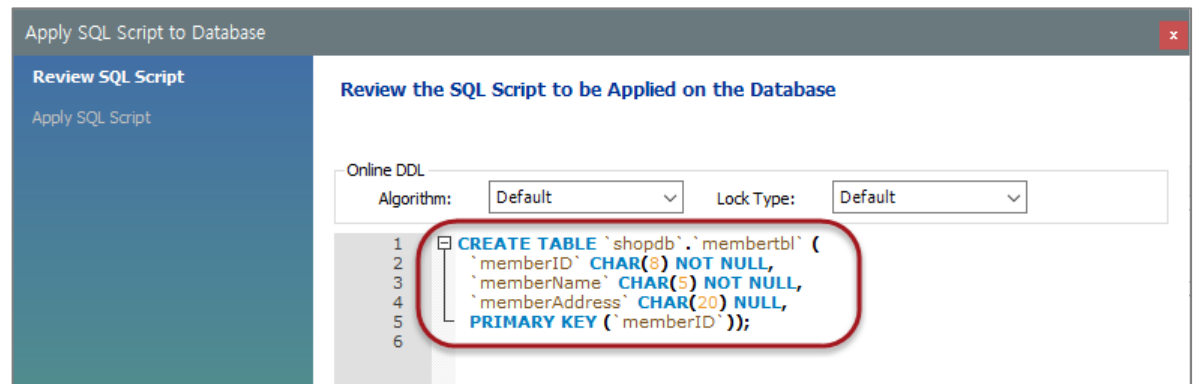
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
memberID	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
memberName	CHAR(5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
memberAddress	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

## 2-4 <Apply> 클릭



## 2-5 테이블 생성 완료

## 2-6 테이블 생성 창 닫기



## [실습 2-2] 테이블 생성하기

### 3 제품 테이블(productTBL) 만들기

#### 3-1 제품 테이블 생성

Query 1 productTBL - Table

Table Name: productTBL Schema: shopdb

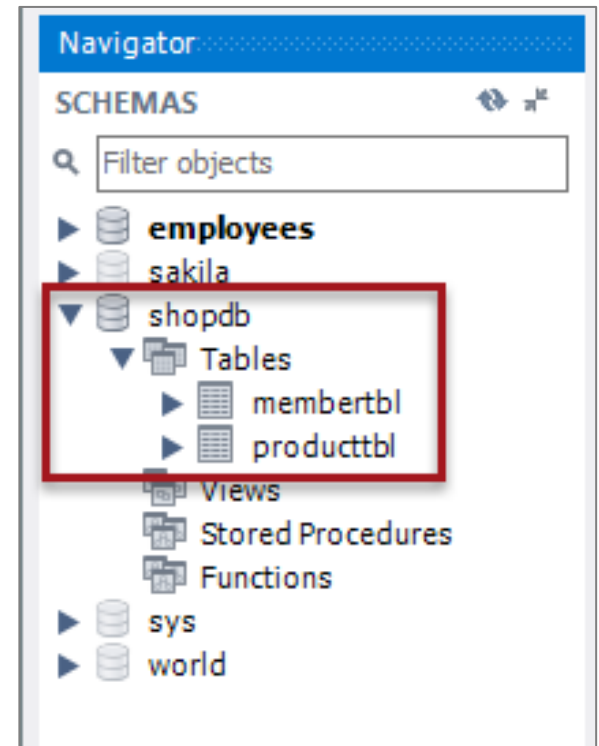
Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
productName	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cost	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
makeDate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
company	CHAR(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
amount	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

### 4 생성한 테이블 확인하기

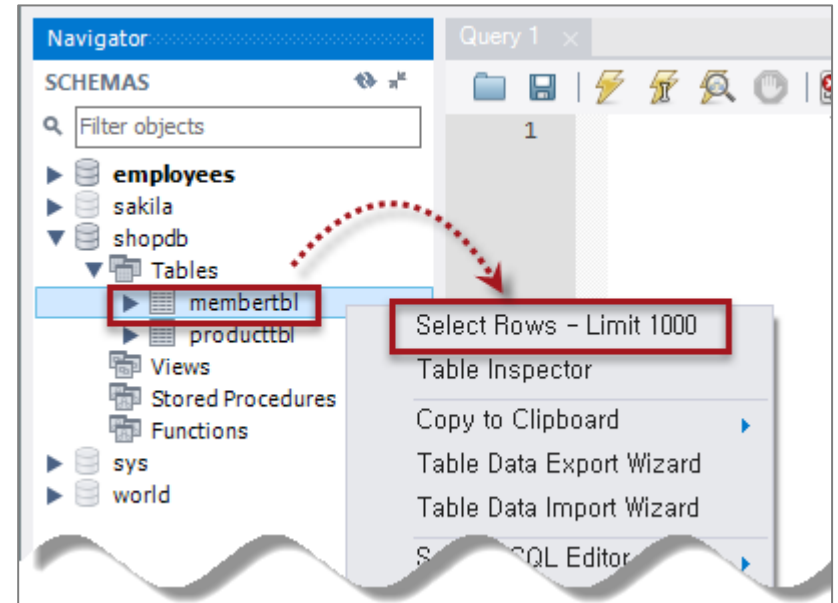
#### 4-1 테이블 2개 생성 확인



# [실습 2-3] 행 데이터 입력하기

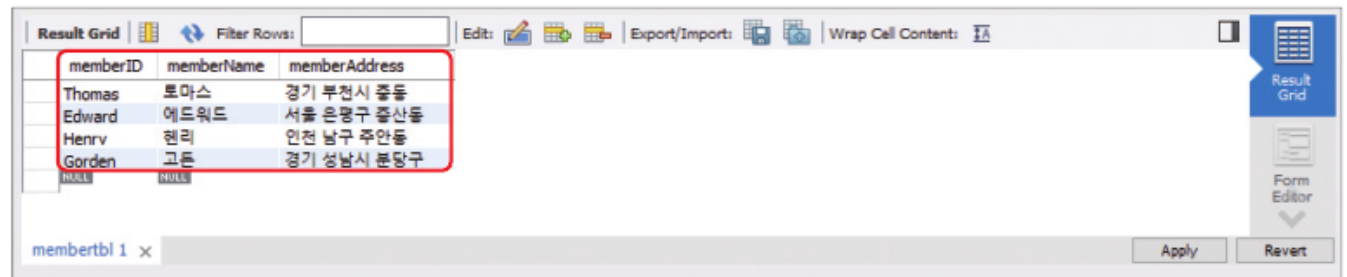
## 1 회원 테이블의 데이터 입력하기

### 1-1 [Select Rows - Limit 1000] 선택



### 1-2 데이터 입력

### 1-3 창 닫기



## [실습 2-3] 행 데이터 입력하기

### 2 제품 테이블의 데이터 입력하기

#### 2-1 데이터 입력

#### 2-2 창 닫기

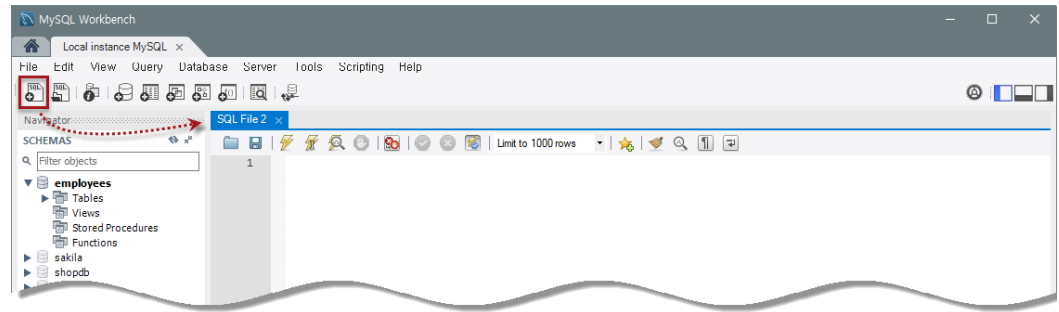
Result Grid			Filter Rows:	<input type="text"/>	Edit:				Export/Import:		
	productName	cost	makeDate	company	amount						
	냉장고	10	2019-07-01	삼성	17						
	컴퓨터	20	2020-03-01	LG	3						
	모니터	5	2021-09-01	롯데	22						
	NULL	NULL	NULL	NULL							

# [실습 2-4] SQL 문 작성하기

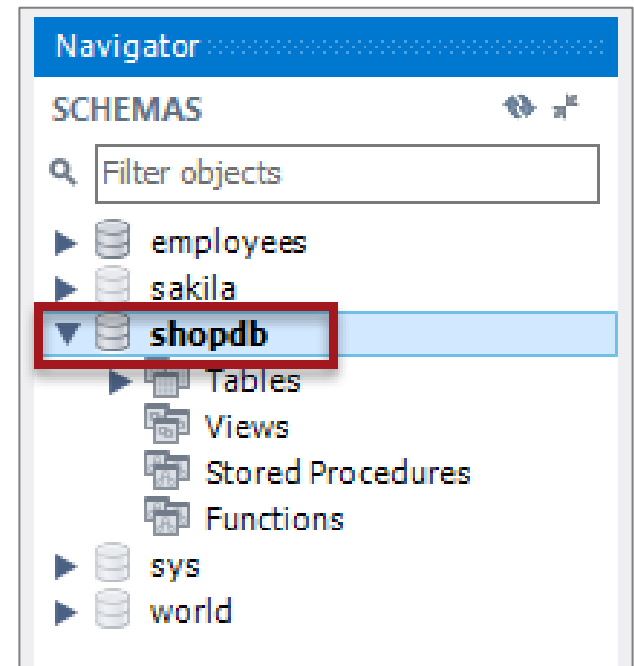
## 1 쿼리 창 열기

1-1 열려 있는 쿼리 창이 있으면 모두 닫기

1-2 새 쿼리 창 열기



1-3 사용할 데이터베이스 선택



# [실습 2-4] SQL 문 작성하기

## 2 SELECT 문 작성하기

### 2-1 회원 테이블의 모든 데이터 조회

```
SELECT * FROM memberTBL;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor window displays the query `SELECT * FROM memberTBL;`. The Results window shows the data retrieved from the `memberTBL` table in the `shopdb` schema. The data is presented in a table with columns `memberID`, `memberName`, and `memberAddress`. The results are as follows:

memberID	memberName	memberAddress
Edward	에더워드	서울 은평구 증산동
Gordon	고든	경기 성남시 분당구
Henry	헨리	인천 남구 주안동
Thomas	토마스	경기 부천시 중동



## [실습 2-4] SQL 문 작성하기

### 2-2 회원 테이블의 이름과 주소만 출력

```
SELECT memberName, memberAddress FROM memberTBL;
```

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The main editor area contains a single SQL query: `SELECT memberName, memberAddress FROM memberTBL;`, which is highlighted with a red underline. Below the editor, the 'Result Grid' tab is active, displaying a table with two columns: 'memberName' and 'memberAddress'. The table contains four rows of data. On the right side, there are buttons for 'Result Grid' and 'Form Editor'. At the bottom, a status bar indicates 'memberTBL 3' and 'Read Only'.

memberName	memberAddress
에드워드	서울 은평구 증산동
고든	경기 성남시 분당구
헨리	인천 남구 주안동
토마스	경기 부천시 중동

## [실습 2-4] SQL 문 작성하기

2-3 '토마스'에 대한 정보만 추출

```
SELECT * FROM memberTBL WHERE memberName = '토마스';
```

The screenshot shows a SQL IDE window titled "SQL File 3\* x". The query editor contains two SQL statements:

```
1  
2 • SELECT memberName, memberAddress FROM memberTBL;  
3 • SELECT * FROM memberTBL WHERE memberName = '토마스' ;
```

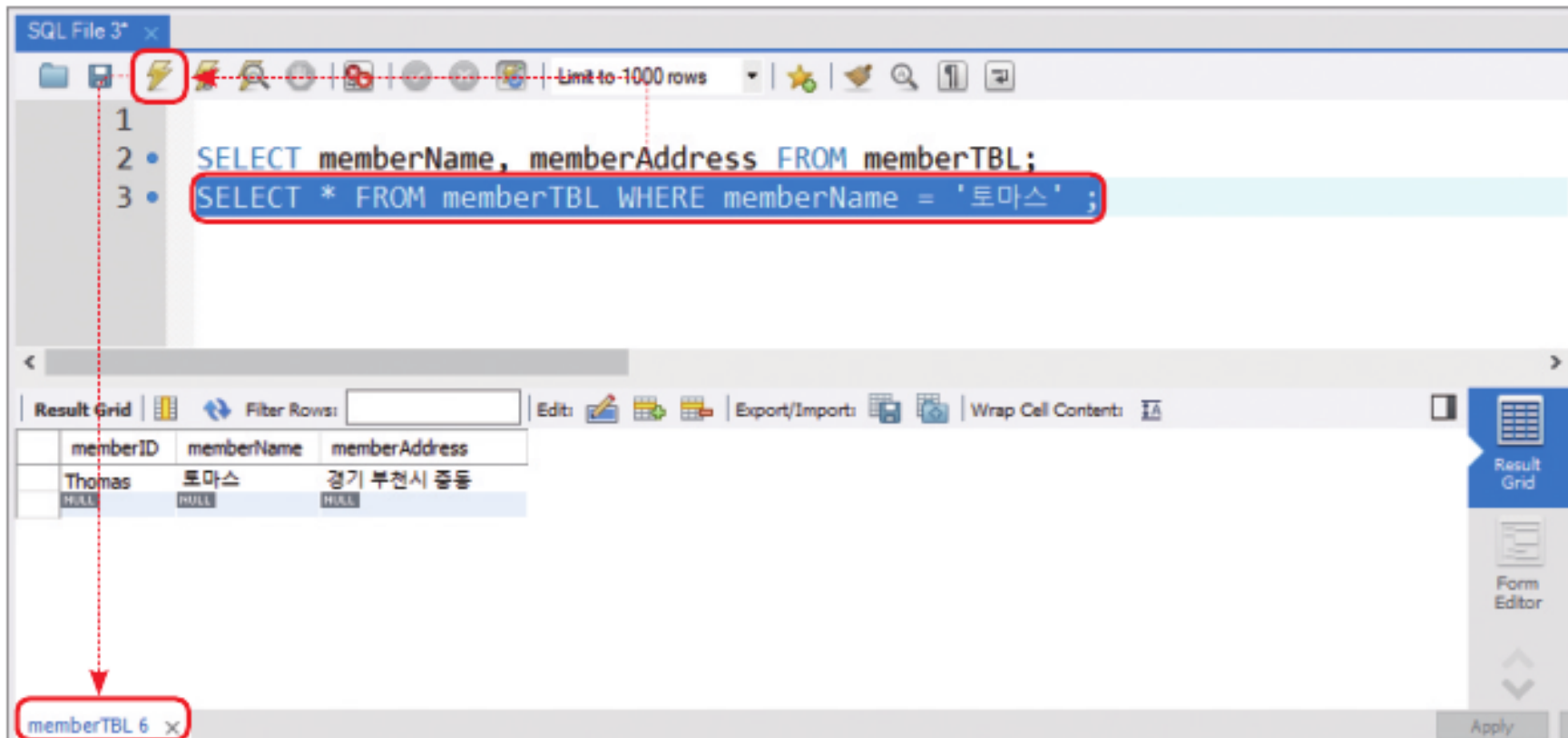
The second statement is highlighted with a red box. Below the query editor, the "Result Grid" is displayed, showing the results of the query:

memberID	memberName	memberAddress
Thomas	토마스	경기 부천시 중동
NULL	NULL	NULL

At the bottom of the IDE, the database structure is visible, showing "memberTBL 4" and "memberTBL 5 x". Red dashed arrows indicate the flow of data from the query to the result grid and from the database structure to the query.

## [실습 2-4] SQL 문 작성하기

2-4 두 번째 쿼리 부분만 실행



SQL File 3\*

Limit to 1000 rows

```
1  
2 • SELECT memberName, memberAddress FROM memberTBL;  
3 • SELECT * FROM memberTBL WHERE memberName = '토마스' ;
```

Result Grid

memberID	memberName	memberAddress
Thomas	토마스	경기 부천시 중동

memberTBL 6

## [실습 2-4] SQL 문 작성하기

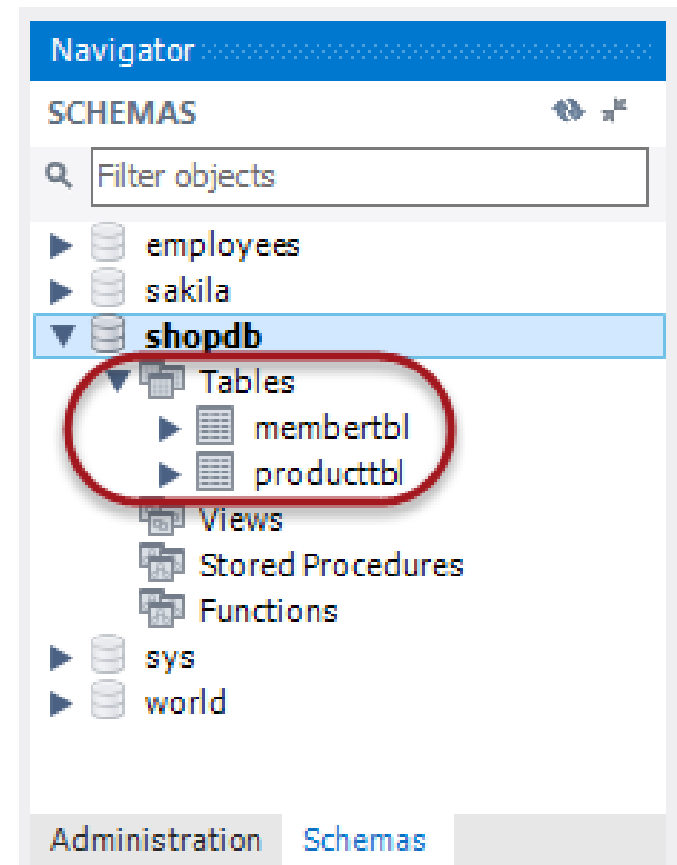
### 3 SQL 문으로 새로운 테이블 생성하기

#### 3-1 간단한 테이블을 생성하는 SQL 문 실행

```
CREATE TABLE `my testTBL` (id INT);
```

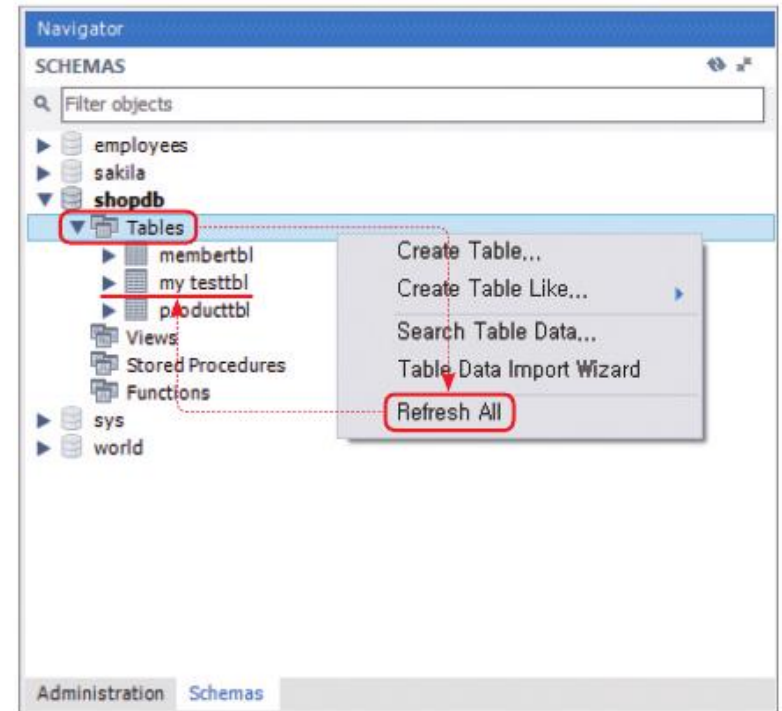
=>여기서 `은 따옴표가 아닌 백틱(backtick)이다.

#### 3-2 [Navigator]에서 방금 생성한 'my testTBL' 확인



## [실습 2-4] SQL 문 작성하기

3-3 [Navigator]에서 [Refresh All]을 선택한 후 확인



### 4 테이블 삭제하기

4-1 DROP TABLE 문을 사용하여 테이블 삭제

```
DROP TABLE `my TestTBL`;
```