



Ch 03. MariaDB 전체 운영 실습

이것이 MariaDB다



❖ 핵심 개념

- 데이터베이스 관련 필수 용어 파악
- 데이터베이스 생성
- 테이블 생성
- 데이터 입력
- 데이터 조회와 활용
- 인덱스, 뷰, 스토어드 프로시저, 트리거 등의 활용
- 데이터 백업과 복원
- 응용프로그램과 MariaDB의 연동



3.1 요구사항 분석과 시스템 설계 그리고 모델링

3.1.1 정보시스템 구축 절차 요약

- 분석, 설계, 구현, 시험, 유지보수의 5가지 단계
- 분석
 - 구현하고자 하는 프로젝트의 가장 첫 번째 단계
 - 시스템 분석 또는 요구사항 분석(무엇을 'What' 할 것인지)
 - 현재 우리가 무엇을 할 것인지 결정
 - 사용자의 인터뷰와 업무 조사 등을 수행
 - 프로젝트의 첫 단추를 끼우는 중요한 단계
 - 많은 시간 동안 심혈을 기울여야 함
 - 분석의 결과로서 많은 문서 작성



3.1 요구사항 분석과 시스템 설계 그리고 모델링

3.1.1 정보시스템 구축 절차 요약

■ 설계

- 시스템 설계 또는 프로그램 설계라는 용어로 부름
- 구축하고자 하는 시스템을 어떻게 'How' 할 것인지 결정
- 시스템 설계가 끝나면 가장 큰 작업이 끝난 것으로 간주
- 대부분의 프로젝트에서 이 분석과 설계의 과정이 전체 공정의 50% 이상을 차지

■ 구현

- 시스템 설계가 끝나고 나면 그 결과 문서들을 프로그래머 (또는 코더)에게 넘겨주면 프로그래머는 설계서에 나온 그대로 프로그램 작성

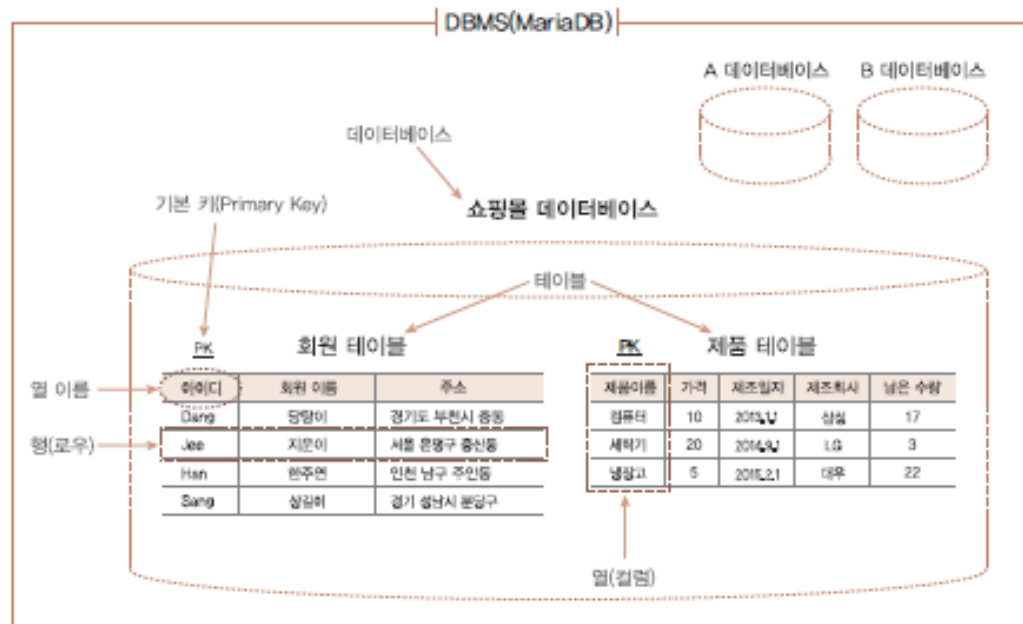


3.1 요구사항 분석과 시스템 설계 그리고 모델링

3.1.2 데이터베이스 모델링과 필수 용어

■ 데이터베이스 모델링

- 현실세계에서 사용되는 데이터를 MariaDB에 어떻게 옮겨 놓을 것인지를 결정하는 과정
- 저장할 정보는 테이블(Table)이라는 형식에 맞춰 저장



[그림 3-1] 테이블의 구조와 관련 용어



3.1 요구사항 분석과 시스템 설계 그리고 모델링

3.1.2 데이터베이스 모델링과 필수 용어

- 데이터베이스 모델링 관련 용어

- 데이터

- 단편적인 정보.

- Ex) 당탕이, 컴퓨터, 2013.1.1

- 정보는 있으나 아직 체계화되지 못한 상태

- 테이블

- 회원이나 제품의 데이터를 입력하기 위해, 표 형태로 표현

- Ex) 회원 테이블, 제품 테이블



3.1 요구사항 분석과 시스템 설계 그리고 모델링

3.1.2 데이터베이스 모델링과 필수 용어

- 데이터베이스 모델링 관련 용어
 - 데이터베이스(DB)
 - 테이블이 저장되는 저장소
 - 원통 모양으로 주로 표현
 - 각 데이터베이스는 서로 다른 고유한 이름을 가지고 있어야 함
 - DBMS
 - DataBase Management System
 - 데이터베이스를 관리하는 시스템 또는 소프트웨어



3.1 요구사항 분석과 시스템 설계 그리고 모델링

3.1.2 데이터베이스 모델링과 필수 용어

- 데이터베이스 모델링 관련 용어
 - 열(=컬럼(column)=필드(Field))
 - 테이블의 구성 요소
 - 회원 테이블의 경우에는 아이디, 회원 이름, 주소 등 3개의 열로 구성
 - 열 이름
 - 각 열을 구분하기 위한 이름
 - 열 이름은 각 테이블 내에서 중복되지 않고, 고유해야 함
 - Ex) 회원 테이블의 아이디, 회원 이름, 주소



3.1 요구사항 분석과 시스템 설계 그리고 모델링

3.1.2 데이터베이스 모델링과 필수 용어

- 데이터베이스 모델링 관련 용어
 - 데이터 형식 (Data Type)
 - 열의 데이터 형식
 - » 테이블을 생성할 때 열 이름과 함께 지정
 - Ex) 회원 테이블의 회원 이름 -문자 형식
 - 행(=로우(Row)=레코드(Record))
 - 실질적인 데이터
 - ex) Jee/지운이/서울 은평구 증산동



3.1 요구사항 분석과 시스템 설계 그리고 모델링

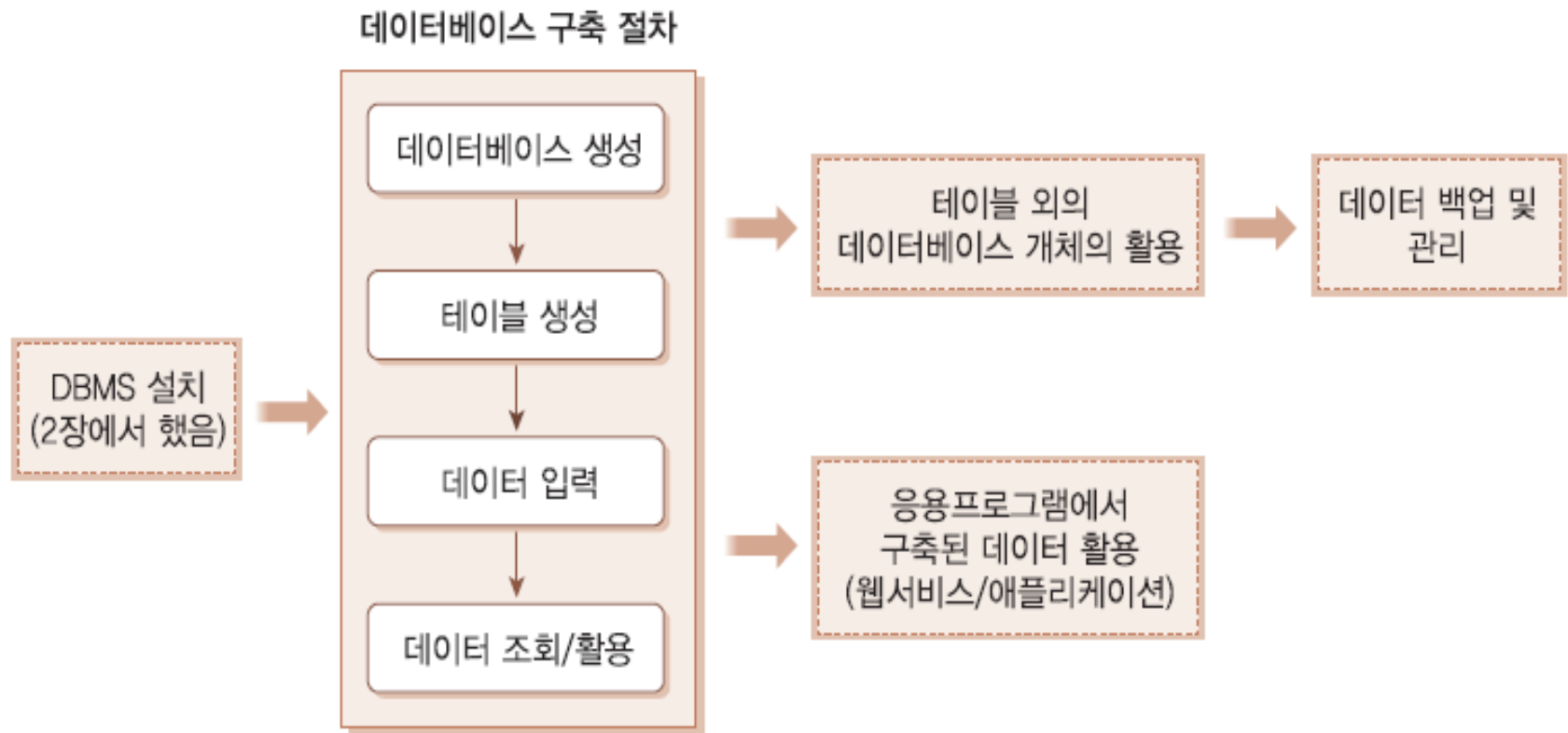
3.1.2 데이터베이스 모델링과 필수 용어

- 데이터베이스 모델링 관련 용어
 - 기본 키
 - Primary Key (주 키)
 - 각 행을 구분하는 유일한 열(Unique)
 - 중복 X, 비는 것 X (Not Null)
 - 각 테이블에는 기본 키가 하나만 지정
 - 외래 키
 - Foreign Key 필드
 - 두 테이블의 관계를 맺어주는 키
 - SQL
 - 사람과 DBMS가 소통하기 위한 말(언어)



3.2 MariaDB를 이용한 데이터베이스 구축 절차

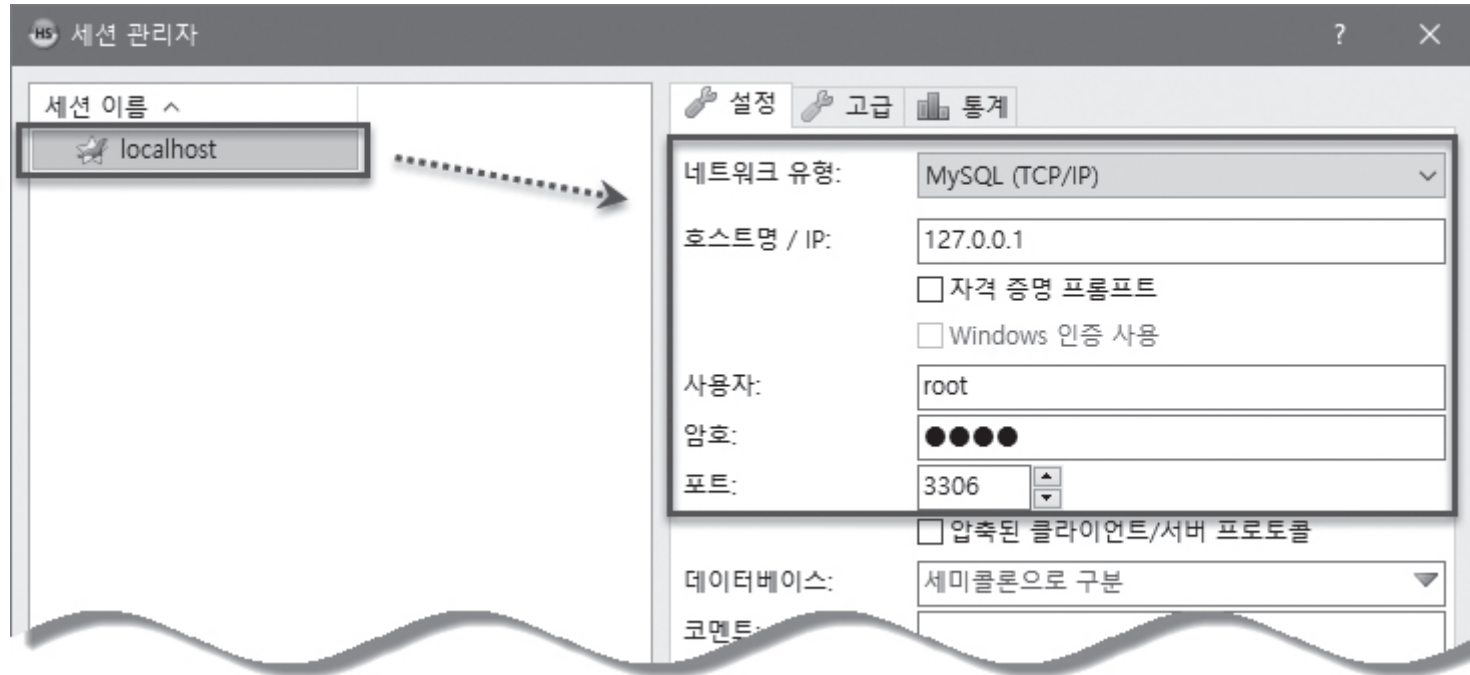
❖ 데이터베이스 구축/관리 및 활용의 전반적인 절차



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.1 데이터베이스 생성

- 쇼핑몰 (shopDB) 데이터베이스 생성 (0)
 - HeidiSQL 실행 후 MariaDB에 연결



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.1 데이터베이스 생성

- 쇼핑몰 (shopDB) 데이터베이스 생성 (0)
- HeidiSQL 실행 창의 명칭

The screenshot shows the HeidiSQL 9.4.0.5125 interface. The left sidebar displays the database structure for 'localhost', including 'employees' (197.4 MiB), 'information_schema' (176.0 KiB), 'mysql', 'performance_schema', and 'test'. The main window shows a query: 'SELECT * FROM dept_manager;'. Below the query, the 'dept_manager (4x24)' table is displayed with the following data:

dept_no	emp_no	from_date	to_date
d001	110,022	1985-01-01	1991-10-01
d001	110,039	1991-10-01	9999-01-01
d002	110,085	1985-01-01	1989-12-17
d002	110,114	1989-12-17	9999-01-01
d003	110,183	1985-01-01	1992-03-21
d003	110,228	1992-03-21	9999-01-01

Annotations on the screenshot include:

- '데이터베이스 목록 창' (Database List Window) pointing to the left sidebar.
- '쿼리 창' (Query Window) pointing to the main query area.
- '쿼리 실행 버튼' (Query Execution Button) pointing to the play button icon in the toolbar.
- '새 쿼리창 열기' (Open New Query Window) pointing to the plus icon in the toolbar.
- '쿼리 결과 창' (Query Results Window) pointing to the table displaying the results of the query.
- '결과 메시지 창' (Results Message Window) pointing to the bottom status bar.

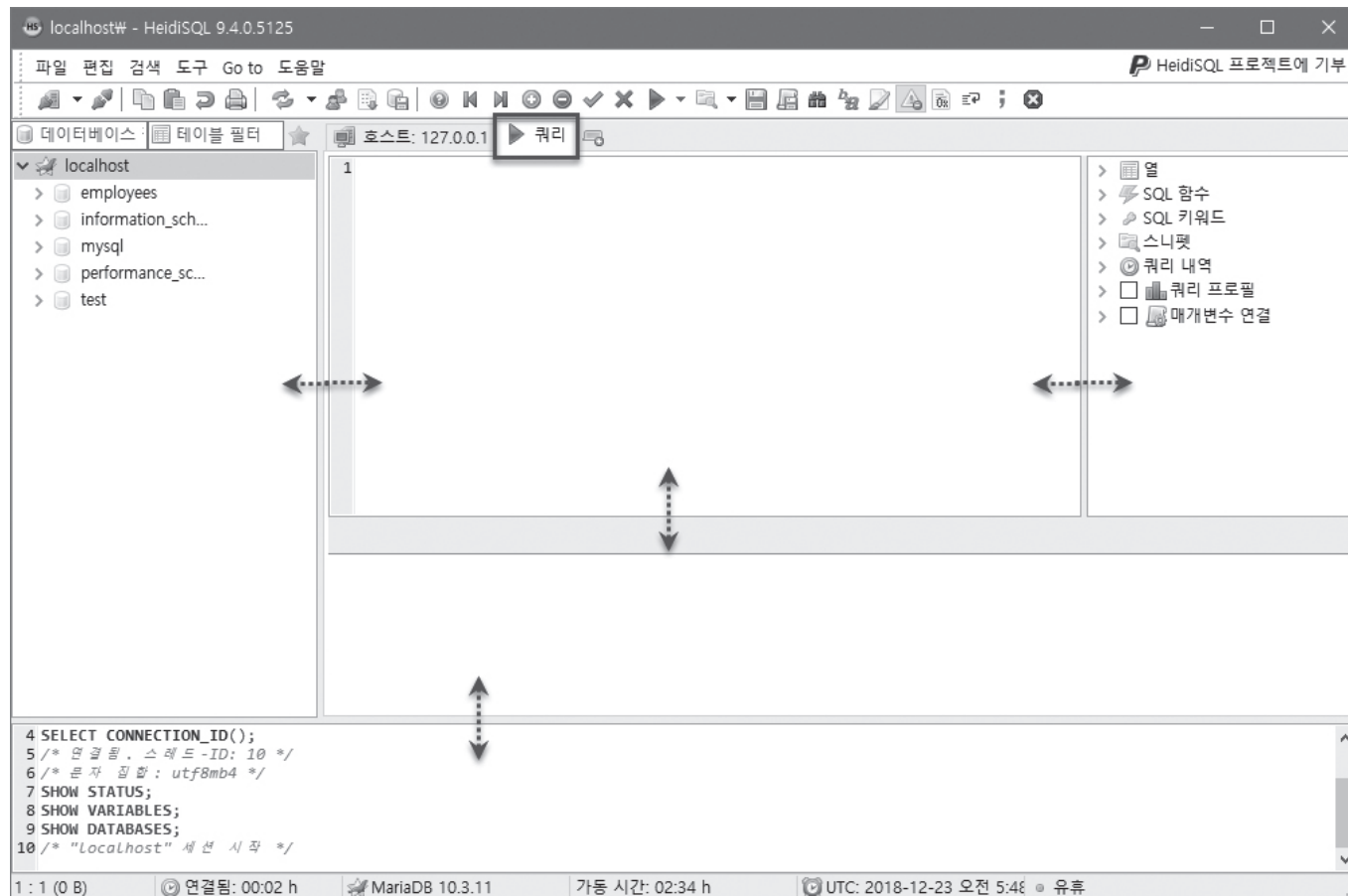
The bottom status bar shows the following information: 2:31 (36 B), 연결됨: 00:01 h, MariaDB 10.3.11, 가동 시간: 00:31 h, UTC: 2018-12-23 오전 3:45, 유틸.



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.1 데이터베이스 생성

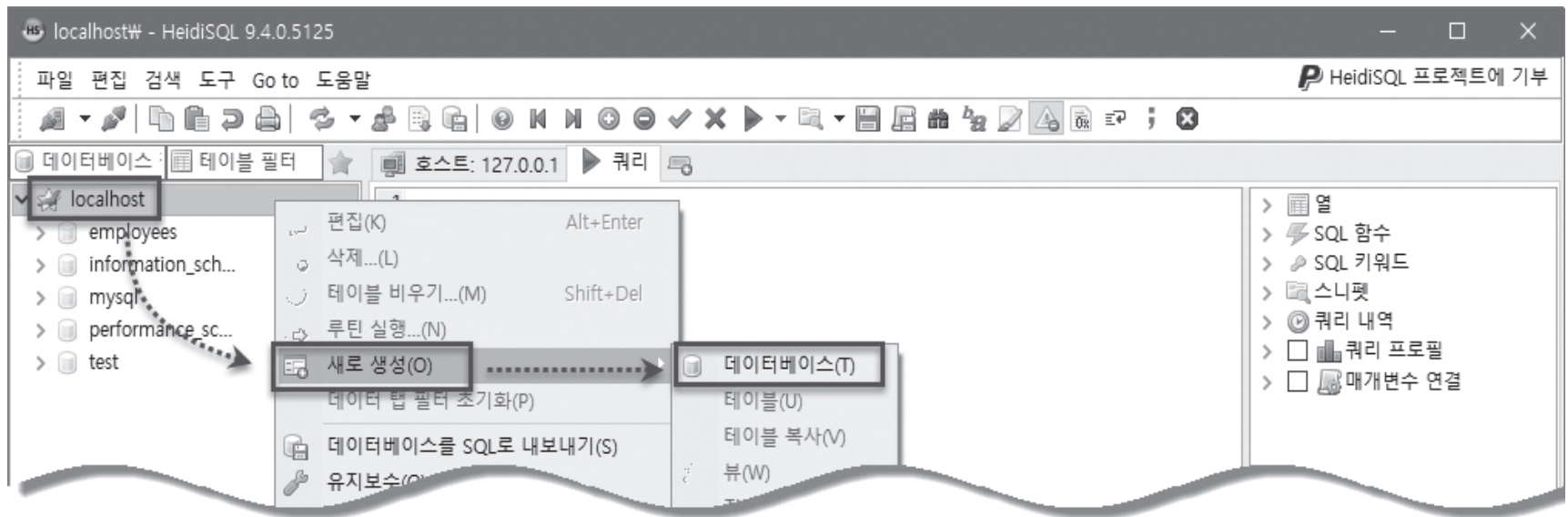
- 쇼핑몰 (shopDB) 데이터베이스 생성 (0)
 - 쿼리 창의 크기를 조금 크게 해 놓으면 편리



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.1 데이터베이스 생성

- 쇼핑몰 (shopDB) 데이터베이스 생성 (1)
 - 쇼핑몰 데이터 베이스 생성



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.1 데이터베이스 생성

- 쇼핑몰 (shopDB) 데이터베이스 생성 (1)

데이터베이스 생성...

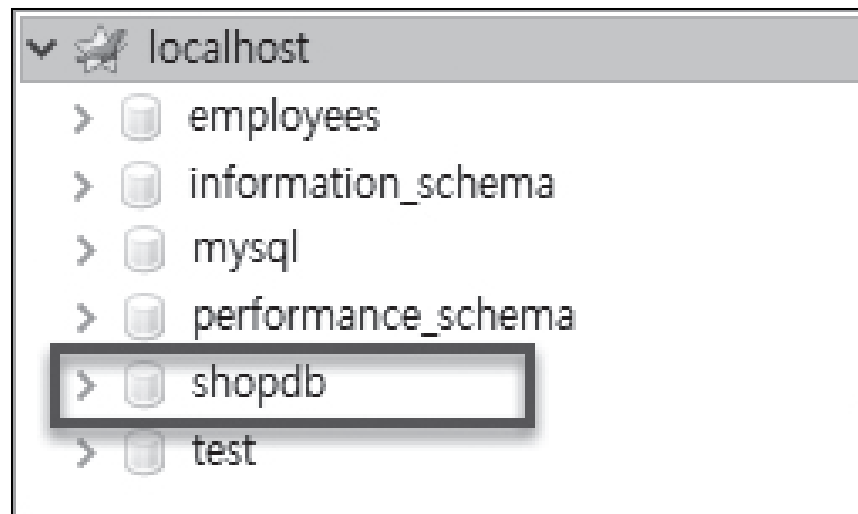
이름(N)

조합(O)

서버 기본값: utf8_general_ci

CREATE 코드:

```
CREATE DATABASE `shopdb` /*!40100 COLLATE
```



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.2 테이블 생성

■ 테이블 생성하기 (0)

- 각 열의 영문 이름 및 데이터 형식 결정
- 원래 데이터베이스 모델링(특히 물리적 모델링) 시 결정

열 이름(한글)	영문 이름	데이터 형식	길이	NULL 허용
아이디(Primary Key)	memberID	문자(CHAR)	8글자(영문)	X
회원 이름	memberName	문자(CHAR)	5글자(한글)	X
주소	memberAddress	문자(CHAR)	20글자(한글)	O

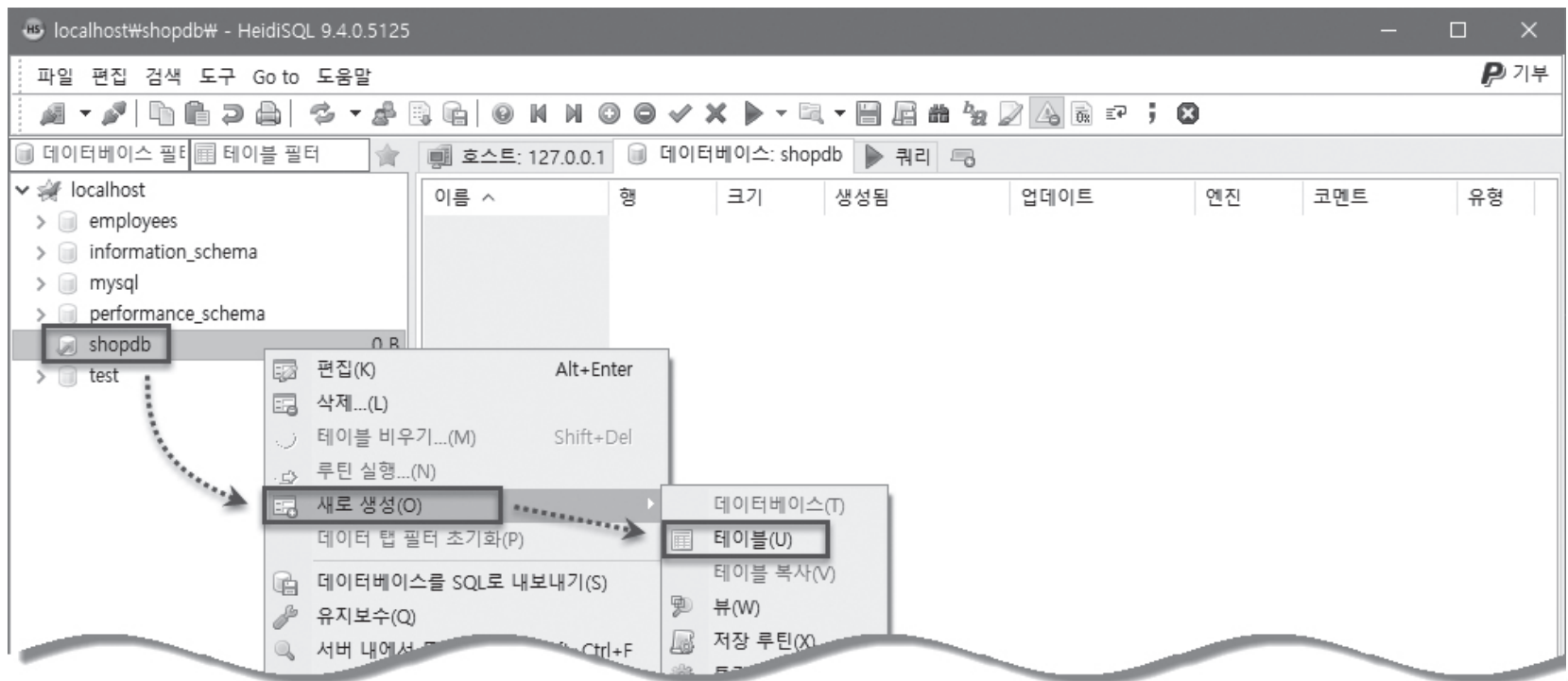
열 이름(한글)	영문 이름	데이터 형식	길이	NULL 허용
제품 이름(Primary Key)	productName	문자(CHAR)	4글자(한글)	X
가격	cost	숫자(INT)	정수	X
제조일자	makeDate	날짜(DATE)	날짜형	O
제조회사	company	문자(CHAR)	5글자(한글)	O
남은 수량	amount	숫자(INT)	정수	X



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.2 테이블 생성

- 테이블 생성하기 (1)
 - 회원 테이블 만들기



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.2 테이블 생성

■ 테이블 생성하기 (1)

• 회원 테이블 내용 입력하기

호스트: 127.0.0.1 데이터베이스: shopdb 테이블: [Untitled] 쿼리

기본 옵션 인덱스 외래 키 분할 CREATE 코드

이름: memberTBL

코멘트:

열: 추가 제거 위로 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값	코멘트
1	memberID	CHAR	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	

도움말 되돌리기 저장

열: 추가 제거 위로 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값	코멘트
1	memberID	CHAR	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
2	memberName	CHAR	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
3	memberAddress	CHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	

3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.2 테이블 생성

■ 테이블 생성하기 (1)

- 회원 테이블 내용 입력하기 - 기본 키 설정하기

The screenshot shows a table creation interface with the following table structure:

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값	코멘트
1	memberID	CHAR	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
2	memberName		5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
3	memberAddress		20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	

A context menu is open over the 'memberID' field, showing options like '복사(C) Ctrl+C', '선택한 열 복사(S)', '열 붙여넣기(T)', '열 추가(U) Ctrl+Ins', '열 제거(V) Ctrl+Del', '위로(W) Ctrl+U', '아래로(X) Ctrl+D', '새 인덱스 생성(Y)', and '인덱스에 추가(Z)'. The '새 인덱스 생성(Y)' option is selected, and a sub-menu is open showing index types: PRIMARY, KEY, UNIQUE, FULLTEXT, and SPATIAL. The 'PRIMARY' option is highlighted.

At the bottom of the interface, there are buttons for '도움말' (Help), '되돌리기' (Undo), and '저장' (Save).

3.2 MariaDB를 이용한 데이터베이스 구축 절차

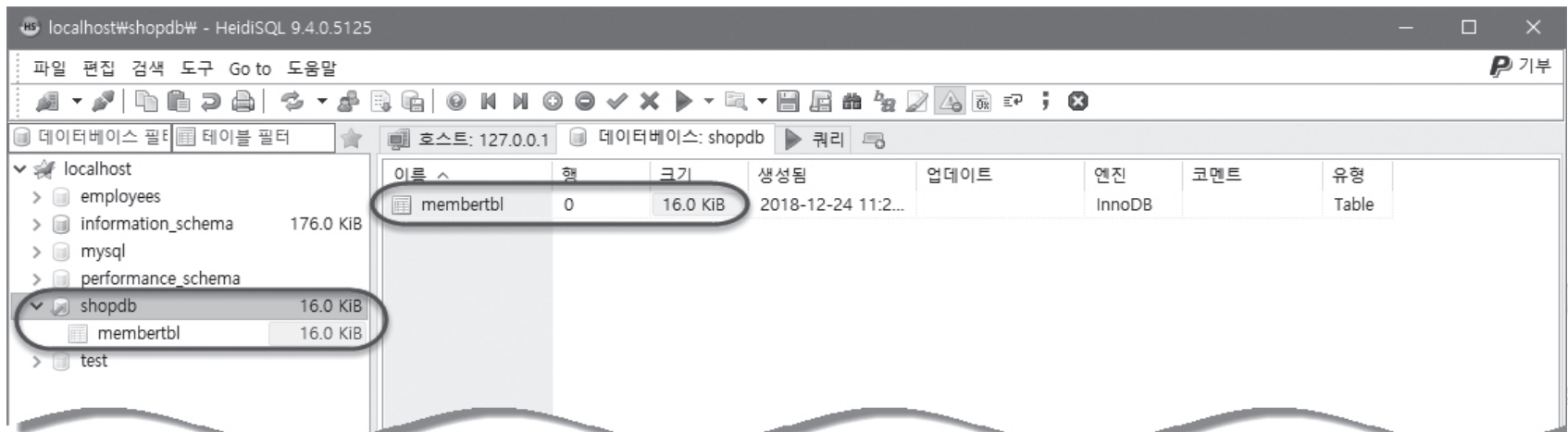
3.2.2 테이블 생성

- 테이블 생성하기 (1)
 - 회원 테이블 내용 저장과 확인

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값	코멘트
1	memberID	CHAR	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
2	memberName	CHAR	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
3	memberAddress	CHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음	

< >

도움말 되돌리기 **저장**



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.2 테이블 생성

- 테이블 생성하기 (1)
 - 제품 테이블 생성과 저장

기본 옵션 인덱스 외래 키 분할 CREATE 코드

이름: productTBL

코멘트:

열: [+] 추가 [-] 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값	코멘트
1	productName	CHAR	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
2	cost	INT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	
3	makeDate	DATE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	기본값 없음	
4	company	CHAR	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	기본값 없음	
5	amount	INT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	

< >

도움말 되돌리기 저장

3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.2 테이블 생성

- 테이블 생성하기 (1)
 - 테이블 생성 결과 확인

localhost#shopdb# - HeidiSQL 9.4.0.5125

파일 편집 검색 도구 Go to 도움말

데이터베이스 필터 테이블 필터

호스트: 127.0.0.1 데이터베이스: shopdb 테이블: productTBL 데이터 쿼리+

localhost

localhost

- employees
- information_schema 176.0 KiB
- mysql
- performance_schema
- shopdb 32.0 KiB**
 - membertbl 16.0 KiB
 - producttbl 16.0 KiB
- test

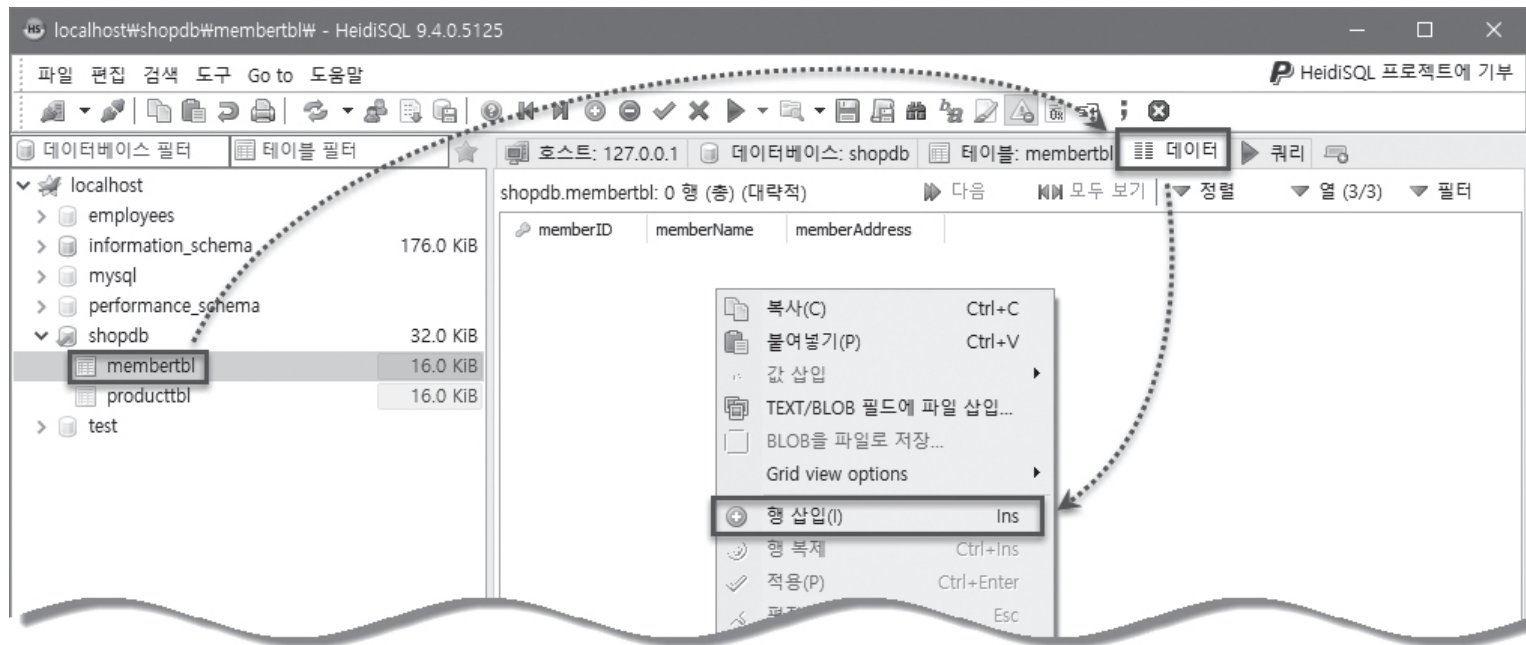
이름 ^	행	크기	생성됨	업데이트	엔진	코멘트	유형
membertbl	0	16.0 KiB	2021-11-10 10:1...		InnoDB		Table
producttbl	0	16.0 KiB	2021-11-10 10:2...		InnoDB		Table



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.3 데이터 입력

- 행 데이터 입력 (1)
 - 데이터 행 삽입 버튼 사용



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.3 데이터 입력

■ 행 데이터 입력 (1)

- 데이터는 빈 칸을 더블 클릭해 입력
- 행 데이터 삭제도 메뉴 통해 가능
- 회원 테이블, 제품 테이블 모두 데이터 입력하도록

localhost#shopdb#membertbl# - HeidiSQL 9.4.0.5125

파일 편집 검색 도구 Go to 도움말

데이터베이스 필터 테이블 필터

localhost

- employees
- information_schema 176.0 KiB
- mysql
- performance_schema
- shopdb 32.0 KiB
 - membertbl 16.0 KiB**
 - producttbl 16.0 KiB
- test

shopdb.membertbl: 4 행 (총) (대략적)

memberID	memberName	memberAddress
Dang	당탕이	경기 부천시 중동
Jee	지운이	서울 은평구 증산동
Han	한주연	인천 남구 주안동
Sang	상질이	경기 성남구 분당구

shopdb.membertbl: 4 행 (총) (대략적)

memberID	memberName	memberAddress
Dong	당탕이	경기 부천시 중동
Jee	지운이	서울 은평구 증산동
Han	한주연	인천 남구 주안동
Sang	상질이	경기 성남구 분당동

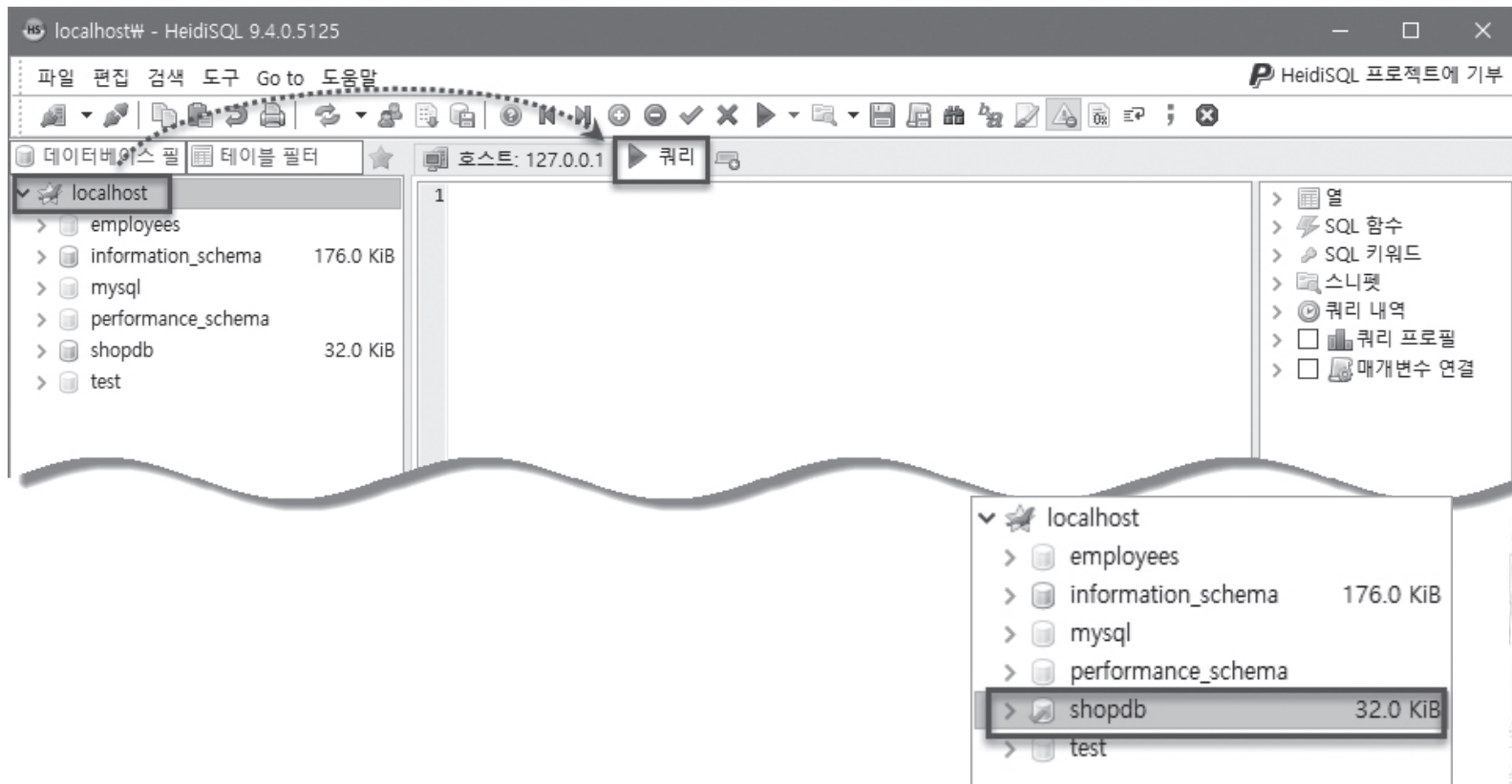
shopdb.producttbl: 3 행 (총) (대략적)

productName	cost	makeDate	company	amount
냉장고	5	2019-02-01	대우	22
세탁기	20	2018-09-01	LG	3
컴퓨터	10	2017-01-01	삼성	17

3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.4 데이터 활용

- SELECT 문 사용법 익히기 (0)
 - SQL문을 직접 입력할 수 있는 쿼리 창 열기
 - 적용되는 DB를 꼭 설정하기



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.4 데이터 활용

- SELECT 문 사용법 익히기 (1)

- 회원 테이블의 모든 데이터 조회

– SELECT 열 이름 FROM 테이블 이름 WHERE 조건

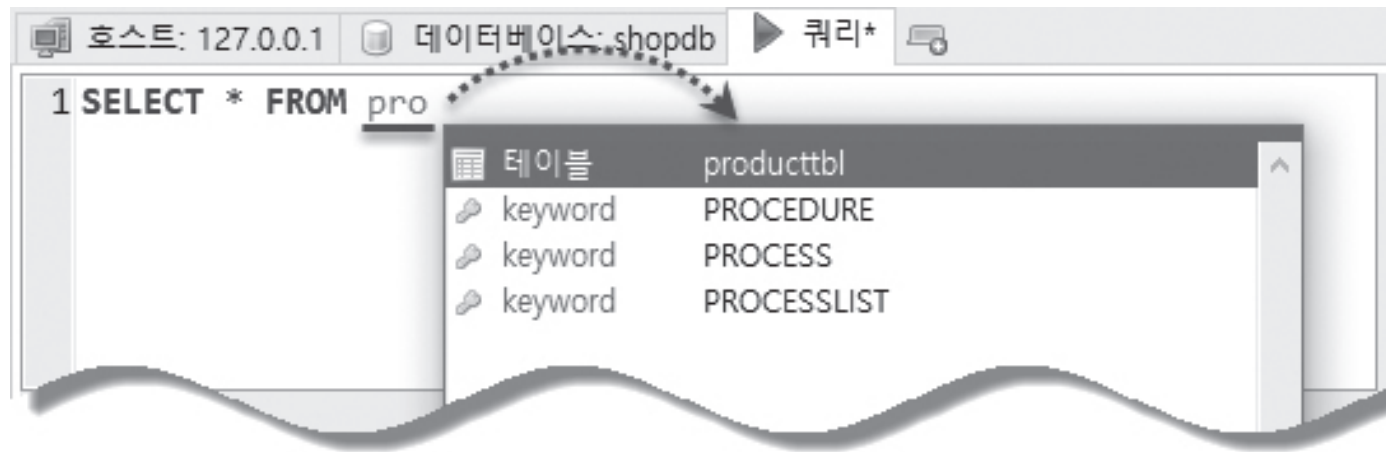
The screenshot shows the HeidiSQL interface. The left sidebar displays the database structure for 'localhost', with 'shopdb' selected. The main window shows the SQL editor with the query: `1 SELECT * FROM memberTBL;`. Below the editor, the results of the query are displayed in a table format, titled 'membertbl (3x4)'. The table has three columns: 'memberID', 'memberName', and 'memberAddress'. The data rows are: 'Dang' (당당이), 'Han' (한주연), 'Jee' (지운이), and 'Sang' (상길이). The status bar at the bottom indicates the connection is successful, showing '62 SELECT * FROM memberTBL;' and '63/* Affected rows: 0 읽은 행: 4 경고: 0 지속 시간 1 쿼리: 0.000 sec. */'.

memberID	memberName	memberAddress
Dang	당당이	경기 부천시 중동
Han	한주연	인천 남구 주안동
Jee	지운이	서울 은평구 종산동
Sang	상길이	경기 성남구 분당구

3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.4 데이터 활용

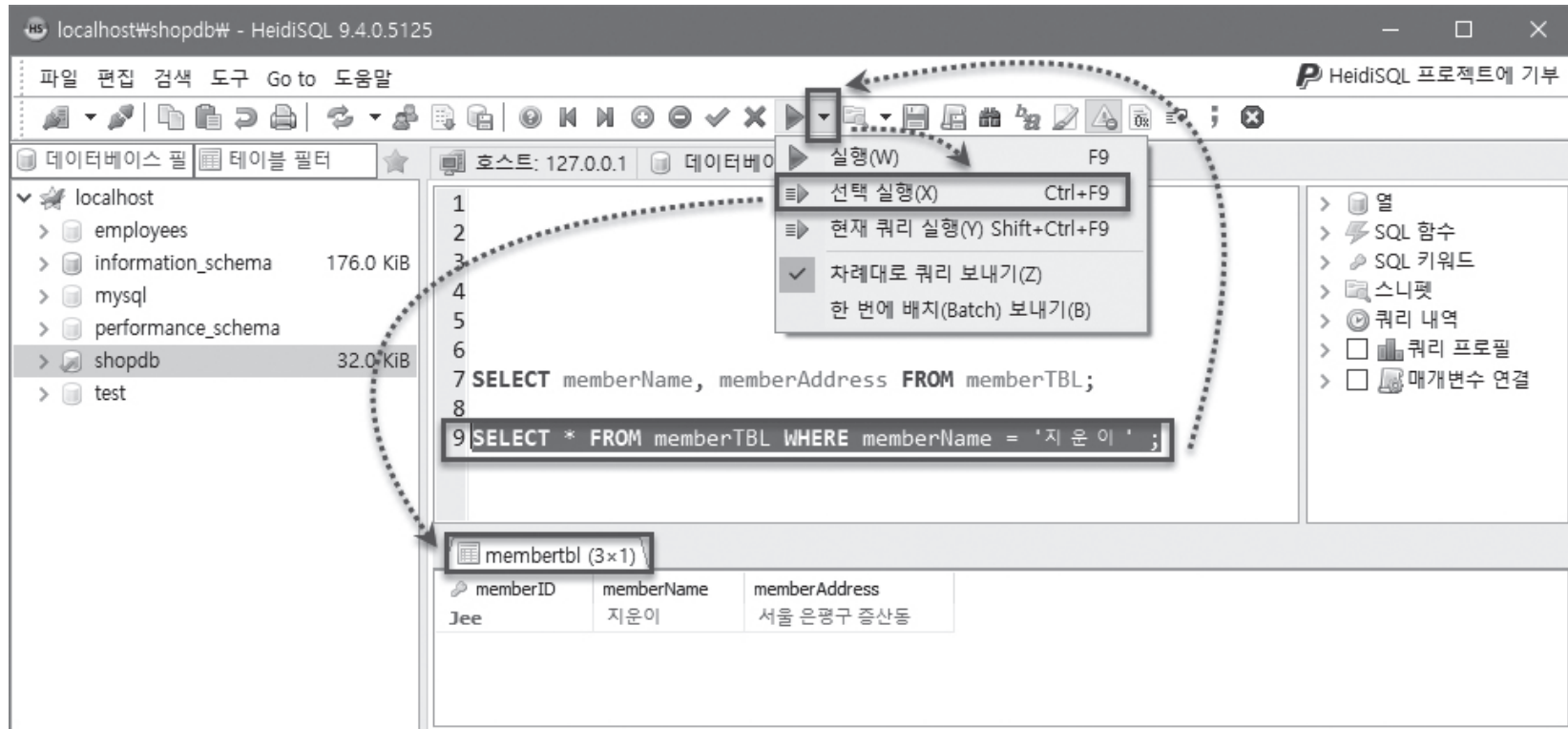
- SELECT 문 사용법 익히기 (1)
 - IntelliSense 기능
 - 예약어는 자동으로 파란색으로 표시
 - Ctrl + Space 키를 누르면 관련 글자 표시



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.4 데이터 활용

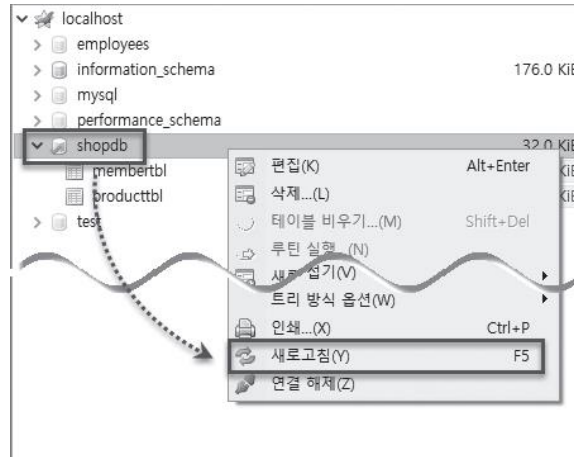
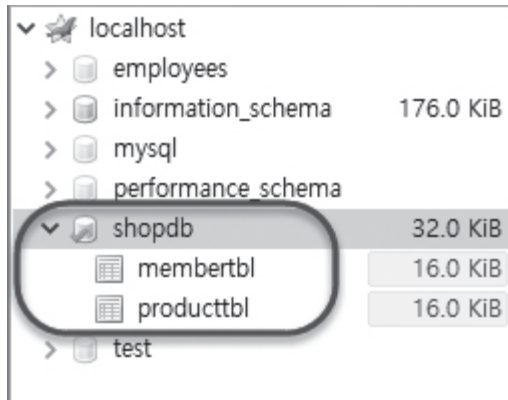
- SELECT 문 사용법 익히기 (1)
 - 선택 실행 사용 - SQL 결과 사이에 일어나는 간섭 막음



3.2 MariaDB를 이용한 데이터베이스 구축 절차

3.2.4 데이터 활용

- SELECT 문 사용법 익히기 (2)
 - SQL 문으로 테이블 생성하기
 - 공백을 처리해주는 backtick 실습
 - » `CREATE TABLE my testTBL (id INT);`
 - -- 하이픈 두 개 + space 로 주석 처리 실습
 - 새로 고침 해줘야 SQL로 만들어진 테이블 보임
 - 테이블 삭제도 같은 방식으로 실습



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.1 인덱스

■ 인덱스 (INDEX)란?

- 책의 뒤에 붙어 있는 ‘찾아보기’(색인)와 같은 개념
- 실무에서 사용되는 대용량 데이터의 경우 필요
- 데이터베이스 기능을 향상시키는 튜닝에서 중요한 요소
 - 쿼리 문에 대한 응답 속도 향상
- 데이터의 열 단위에 생성



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.1 인덱스

- 인덱스 (INDEX)의 간단한 사용 (0)
 - 인덱스를 사용하기 위해 적정량이 있는 테이블 생성
 - 샘플DB인 employee 에서 대량의 데이터 복사해 indexTBL 생성

localhost#shopdb# - HeidiSQL 9.4.0.5125

파일 편집 검색 도구 Go to 도움말 HeidiSQL 프로젝트에 기부

데이터베이스 필 테이블 필터 호스트: 127.0.0.1 데이터베이스: shopdb 쿼리*

```
1 CREATE TABLE indexTBL (first_name varchar(14), last_name varchar(16), hire_date date);
2 INSERT INTO indexTBL
3   SELECT first_name, last_name, hire_date
4   FROM employees.employees
5   LIMIT 500;
6 SELECT * FROM indexTBL;
7
```

indextbl (3x500)

first_name	last_name	hire_date
Georgi	Facello	1986-06-26
Bezael	Simmel	1985-11-21
Parto	Bamford	1986-08-28
Chirstian	Koblick	1986-12-01

```
25 CREATE TABLE indexTBL (first_name varchar(14), last_name varchar(16), hire_date date);
26 INSERT INTO indexTBL   SELECT first_name, last_name, hire_date   FROM employees.employees   LIMIT 500;
27 SELECT * FROM indexTBL;
28 /* Affected rows: 500   찾은 행: 500   경고: 0   지속 시간 3 queries: 0.047 sec. */
```

연결됨: 00:03 h MariaDB 10.3.11 가동 시간: 2 일, 00:17 h UTC: 2018-12-25 오전 3:31 유휴

3.3 테이블 외의 데이터베이스 개체의 활용

3.3.1 인덱스

- 인덱스 (INDEX)의 간단한 사용 (1)
 - 인덱스가 없을 때의 쿼리 작동
 - EXPLAIN 을 쿼리 앞에 붙여 실행될 때 어떤 방식으로 실행되는지
 - ‘실행 계획Execution Plan’의 내용을 보여줌

결과 #1 (10×1)								
id	select_type	table	type	possible_keys	key	key_len	ref	rows
1	SIMPLE	indexTBL	ALL	(NULL)	(NULL)	(NULL)	(NULL)	500
		Extra Using where						



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.1 인덱스

- 인덱스 (INDEX)의 간단한 사용 (1)

- 인덱스 생성 후 실행계획 확인

– CREATE INDEX idx_indexTBL_firstname ON
indexTBL(first_name);

결과 #1 (10×1)									
id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	indexTBL	ref	<u>idx_indexTBL_firstname</u>	idx_indexTBL_firstname	45	const	1	Using index condition



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.2 뷰

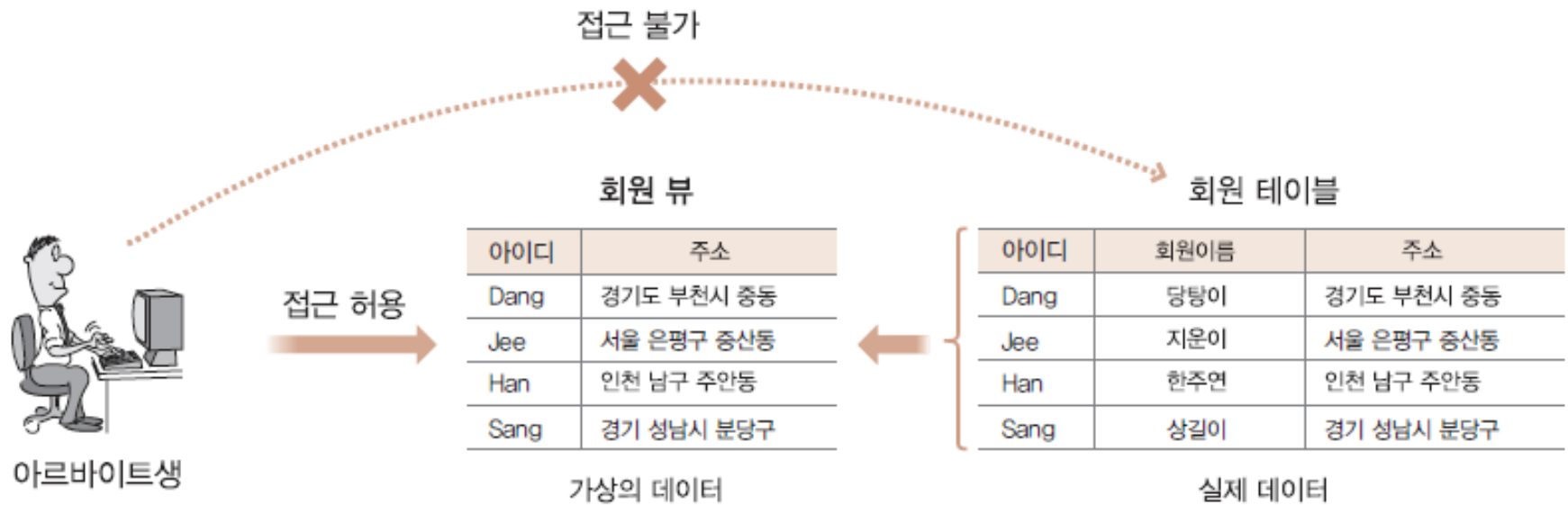
- 뷰 (View) 란?
 - 가상의 테이블
 - 실제 행 데이터를 가지고 있지 않음
 - 사용자의 입장에서는 테이블과 동일하게 보임
 - 실체는 없는 것이며, 진짜 테이블에 링크(Link)된 개념
 - 뷰를 SELECT하면 진짜 테이블의 데이터를 조회하는 것과 동일한 결과
 - 개인정보 보호와 같은 데이터 보호/보안을 위해서도 사용



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.2 뷰

■ 뷰 (View) 의 사용 예



[그림 3-34] 뷰의 사용 예



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.2 뷰

- 뷰 (View) 의 사용법 실습 (0)
 - 뷰를 만들고 내용 조회

```
CREATE VIEW uv_memberTBL
AS
    SELECT memberName, memberAddress FROM memberTBL ;
```

The screenshot shows a SQL IDE interface. The top pane contains the query: `1 SELECT * FROM uv_memberTBL ;`. A dashed arrow points from this query to the bottom pane. The bottom pane displays the result of the query as a table with 2 columns and 4 rows. The columns are labeled `memberName` and `memberAddress`. The rows contain the following data:

memberName	memberAddress
당탕이	경기 부천시 중동
한주연	인천 남구 주안동
지운이	서울 은평구 증산동
상길이	경기 성남구 분당구

On the right side of the IDE, there is a sidebar with a list of icons and labels: > 열, > SQL 함수, > SQL 키워드, > 스니펫, > 쿼리 내역.

3.3 테이블 외의 데이터베이스 개체의 활용

3.3.3 스토어드 프로시저

- 스토어드 프로시저란?
 - MariaDB에서 제공해주는 프로그래밍 기능
 - SQL 문을 하나로 묶어서 편리하게 사용하는 기능
 - 다른 프로그래밍 언어와 같은 기능을 담당하기도 함
 - 실무에서는 SQL문을 하나씩 실행하는 것 보다 스토어드 프로시저로 관리하는 경우가 많음



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.3 스토어드 프로시저

- 스토어드 프로시저 실습
 - 매번 같은 SQL 문을 연 달아서 실행하는 경우
 - 오타의 위험이나 문법 실수를 할 가능성이 높음
 - 스토어드 프로시저를 사용하지 않은 경우

```
1 SELECT * FROM memberTBL WHERE memberName = '당 탕 이';
2
3 SELECT * FROM productTBL WHERE productName = '냉 장 고';
```

membertbl (3x1) producttbl (5x1)

memberID	memberName	memberAddress
	Dang	경기 부천시 중동

3.3 테이블 외의 데이터베이스 개체의 활용

3.3.3 스토어드 프로시저

■ 스토어드 프로시저 실습

```
1 DELIMITER //
2 CREATE PROCEDURE myProc()
3 BEGIN
4     SELECT * FROM memberTBL WHERE memberName = '당 탕 이' ;
5     SELECT * FROM productTBL WHERE productName = '냉 장 고' ;
6 END //
7 DELIMITER ;
```

1 CALL myProc() ;

membertbl (3×1) producttbl (5×1)

memberID	memberName	memberAddress
Dang	당탕이	경기 부천시 중동

- > 열
- > SQL 함수
- > SQL 키워드
- > 스니펫
- > 쿼리 내역
- > 쿼리 프로필
- > 매개변수 연결



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.4 트리거

■ 트리거란?

- 테이블에 부착되어서, 테이블에 INSERT나 UPDATE 또는 DELETE 작업이 발생되면 실행되는 코드
- Ex) 회원이 탈퇴할 경우 탈퇴자 테이블에 데이터를 일부 옮겨놓고 지우는 작업



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.4 트리거

- 트리거의 용도 실습
 - 회원 가입/업데이트/삭제의 일련 작업
 - 회원 가입했던 적이 있는지 기록이 없어 증명을 요구할 경우에는 어떻게 하는가?
 - 삭제 작업 일어날 경우 백업 테이블에 지워진 데이터 기록하는 트리거 생성



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.4 트리거

- 트리거의 용도 실습
 - 백업 테이블에 지워진 결과 기록하는 트리거

```
1 DELIMITER //
```

```
2 CREATE TRIGGER trg_deletedMemberTBL  -- 트리거 이름
```

```
3     AFTER DELETE -- 삭제 후에 작동하게 지정
```

```
4     ON memberTBL -- 트리거를 부착할 테이블
```

```
5     FOR EACH ROW -- 각 행마다 적용시킴
```

```
6 BEGIN
```

```
7     -- OLD 테이블의 내용을 백업 테이블에 삽입
```

```
8     INSERT INTO deletedMemberTBL
```

```
9         VALUES (OLD.memberID, OLD.memberName, OLD.memberAddress, CURDATE() );
```

```
10 END //
```

```
11 DELIMITER ;
```



3.3 테이블 외의 데이터베이스 개체의 활용

3.3.4 트리거

- 트리거의 용도 실습
 - 삭제 이후의 데이터 확인하기

membertbl (3×3)		
memberID	memberName	memberAddress
Han	한주연	인천 남구 주안동
Jee	지운이	서울 은평구 증산동
Sang	상길이	경기 성남구 분당구

deletedmembertbl (4×1)			
memberID	memberName	memberAddress	deletedDate
Dang	당탕이	경기 부천시 중동	2018-12-25



3.4 데이터베이스 백업 및 관리

3.4.1 백업과 복원

■ 백업

- 현재의 데이터베이스를 다른 매체에 보관하는 작업
- 다른 디스크에 백업을 받아야 의미가 있음
 - 디스크가 깨지면 백업 받은 것도 다 날아감

■ 복원(Restore)

- 데이터베이스에 문제가 발생했을 때 다른 매체에 백업된 데이터를 이용해서 원상태로 돌려놓는 작업

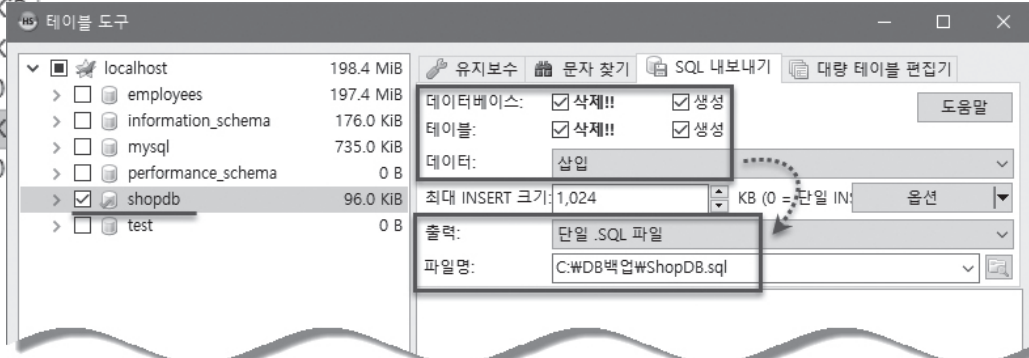
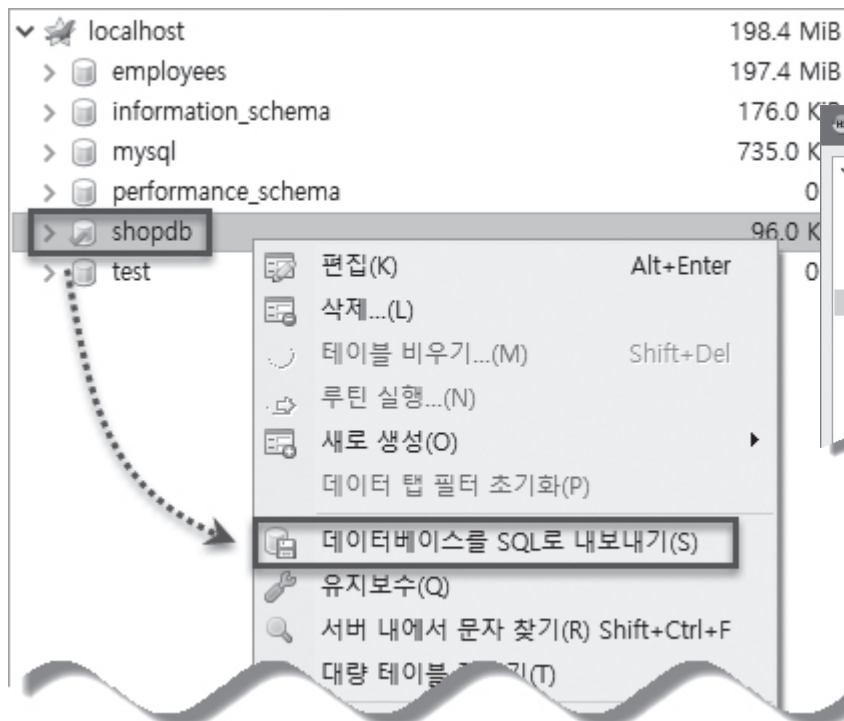


3.4 데이터베이스 백업 및 관리

3.4.1 백업과 복원

■ 백업과 복원 실습

- 실습 때는 다른 디스크나 테이프 장치 준비가 어려우므로 다른 폴더를 사용해서라도 개념 숙지



3.4 데이터베이스 백업 및 관리

3.4.1 백업과 복원

- 백업과 복원 실습
 - DB 삭제 후 불러오기 사용해 복구

The screenshot shows the HeidiSQL interface with the 'File' menu open, highlighting 'SQL 파일 불러오기...(S) Ctrl+O'. A secondary window titled '열기' (Open) is displayed, showing a search for 'DB백업' (DB Backup) and a list of files, with 'ShopDB.sql' selected. The main window shows the 'Database' tab with a list of databases on the left and the SQL editor on the right. The SQL editor contains the following code:

```
1 --  
2 -- 호스트 : 127.0.0.1  
3 -- 서버 버전 : 10.3.11-MariaDB - mariadb.org binary distribution  
4 -- 서버 OS : Win64  
5 -- HeidiSQL 버전 : 9.4.0.5125  
6 --  
7  
8 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;  
9 /*!40101 SET NAMES utf8 */;  
10 /*!50503 SET NAMES utf8mb4 */;  
11 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;  
12 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;  
13  
14  
15 -- shopdb 데이터베이스 구조 내보내기  
16 DROP DATABASE IF EXISTS `shopdb`;  
17 CREATE DATABASE IF NOT EXISTS `shopdb` /*!40100 DEFAULT CHARACTER SET utf8 */;  
18
```



Thank You !

이것이 MariaDB다