



## Chapter 09 MySQL 데이터베이스

# 목차

**01** 데이터베이스

**02** MySQL 서버와 phpMyAdmin

**03** 데이터베이스 생성

**04** 데이터베이스 테이블 관리

# 학습목표

- 데이터베이스에 대해 이해할 수 있다.
- MySQL 서버를 구동할 수 있다.
- phpMyAdmin을 사용할 수 있다.
- 데이터베이스를 생성할 수 있다.
- DB 테이블을 다룰 수 있다.

## 1. 데이터베이스 관리 시스템

- 데이터베이스 관리 시스템(DataBase Management System, DBMS)은 응용 프로그램이 요청한 사항을 데이터베이스와 통신하면서 처리.

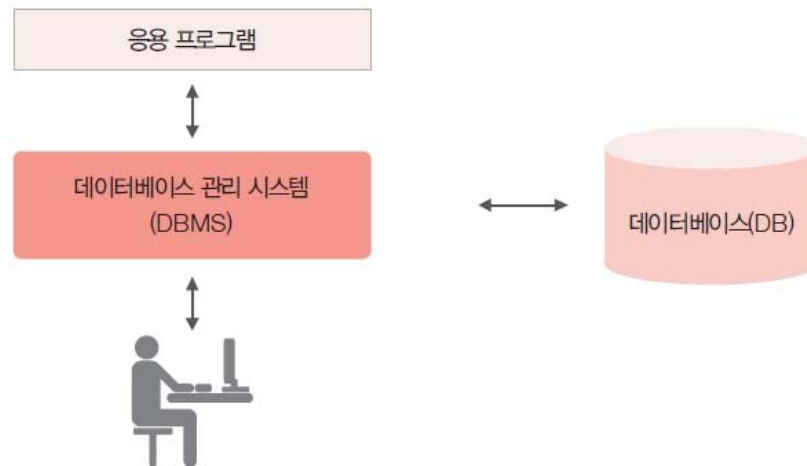


그림 9-1 DBMS의 개념

- MySQL은 DBMS의 일종이며 이외에도 MSSQL, Oracle, Infomix 등 다양한 DBMS가 있음.

- MySQL은 DBMS 중에서도 관계형 데이터베이스 관리 시스템(Relational DataBase Management System, RDBMS)에 속함.
- RDBMS는 테이블 형태로 데이터를 저장하고 이 데이터 간의 관계를 정의하는 것.
- RDBMS의 모든 데이터는 2차원 테이블에 저장된 후 검색, 수정, 삭제 가능.
- RDBMS의 세 가지 기본 구성 요소는 테이블(table), 필드(field), 레코드(record).

The diagram shows a table with 5 columns and 4 rows. A red dashed box labeled '필드(열)' (Field/Column) highlights the first two columns. A red dashed box labeled '테이블' (Table) highlights the entire table structure. A red dashed box labeled '레코드(행)' (Record/Row) highlights the first two rows.

일련번호	이름	학번	수강 과목	성적
1	황예린	201903114001	PHP	98
2	김수겸	201903114002	파이썬	76
3	이수진	201903114003	컴퓨터 구조	87
4	나여진	201903114004	알고리즘 기초	94

그림 9-2 RDBMS의 기본 구성 요소

- 테이블은 데이터베이스에서 데이터를 저장하고 관리하는 기본 단위이며 필드와 레코드로 구성. 필드는 테이블의 열(column)을, 레코드는 행(row)을 나타냄.

## 2. MySQL

- 리눅스, 유닉스, 윈도우 등 모든 플랫폼에서 사용
- PHP, 파이썬, C, C++, 자바 등의 많은 프로그래밍 언어를 지원하는 응용 프로그램 인터페이스(API)를 제공
- MySQL의 장점
  - ✓ 가장 많이 사용하는 데이터베이스. 무료로 사용할 수 있지만 상업적인 목적으로 사용하려면 라이선스 구매해야 함.
  - ✓ DBMS 중에서도 처리 속도가 상당히 빠른 편.
  - ✓ 설치 및 사용 방법이 쉬워 초보자가 쉽게 접근할 수 있음.
  - ✓ 대용량 데이터도 손쉽게 처리할 수 있음.
  - ✓ 보안성이 뛰어나.
- 최근에는 MySQL 과 거의 완벽하게 호환되면서 성능이 향상된 MariaDB가 출시. MariaDB는 MySQL을 인수한 오라클이 MySQL 라이선스를 불확실하게 만들자 이에 반발한 개발자들이 Maria 재단을 세워 MySQL의 호환성을 최대한 유지한 채 성능을 향상해서 만든 데이터베이스

### 1. MySQL 서버 구동

- MySQL 서버를 구동하기 전에 한글이 제대로 표시되도록 문자셋을 설정해야 함.
- MySQL에서 문자셋을 UTF-8로 설정하려면 C:\xampp\mysql\bin 폴더에 있는 my.ini 파일을 변경해야 함.
- 해당 폴더의 my.ini 파일을 불러와서 '## UTF 8 Settings' 아래의 다섯 줄을 수정.

```
...  
## UTF 8 Settings  
init-connect= SET NAMES utf8  
collation_server=utf8_general_ci  
character_set_server=utf8  
skip-character-set-client-handshake  
character_sets-dir="C:/xampp/mysql/share/charsets"  
...
```

그림 9-3 C:\xampp\mysql\bin 폴더의 my.ini 파일

## 02 MySQL 서버와 phpMyAdmin

- my.ini 파일을 수정하여 저장했다면 XAMPP 컨트롤 패널을 열고 MySQL의 [Start] 메뉴를 클릭.

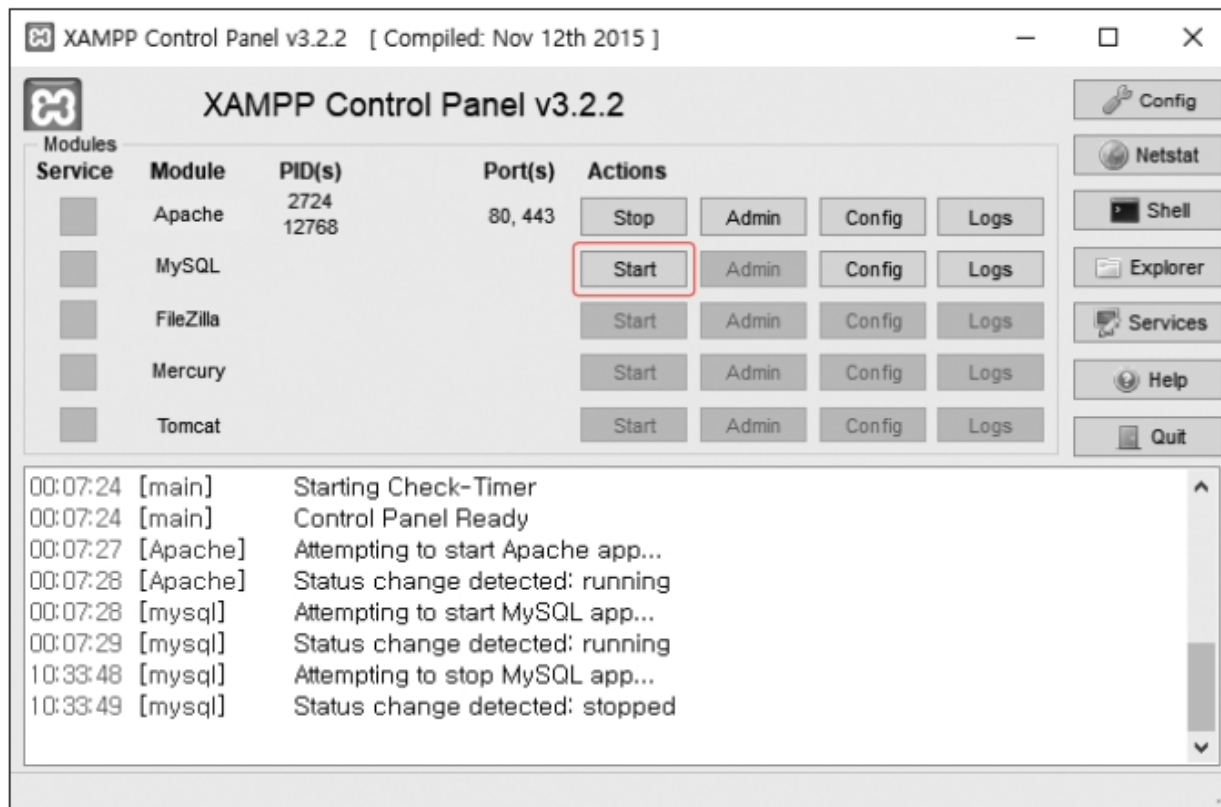


그림 9-4 MySQL 서버 프로그램 시작 전의 XAMPP 컨트롤 패널



### 2. phpMyAdmin 시작

- MySQL 서버 프로그램을 구동한 다음에는 클라이언트 측에서 사용할 MySQL 프로그램이 필요. 1장에서 XAMPP와 함께 설치된 phpMyAdmin 프로그램 사용.
- 웹 브라우저 주소 창에 `http://localhost`를 입력하여 XAMPP 접속, 화면 오른쪽 상단의 [phpMyAdmin] 메뉴 클릭.

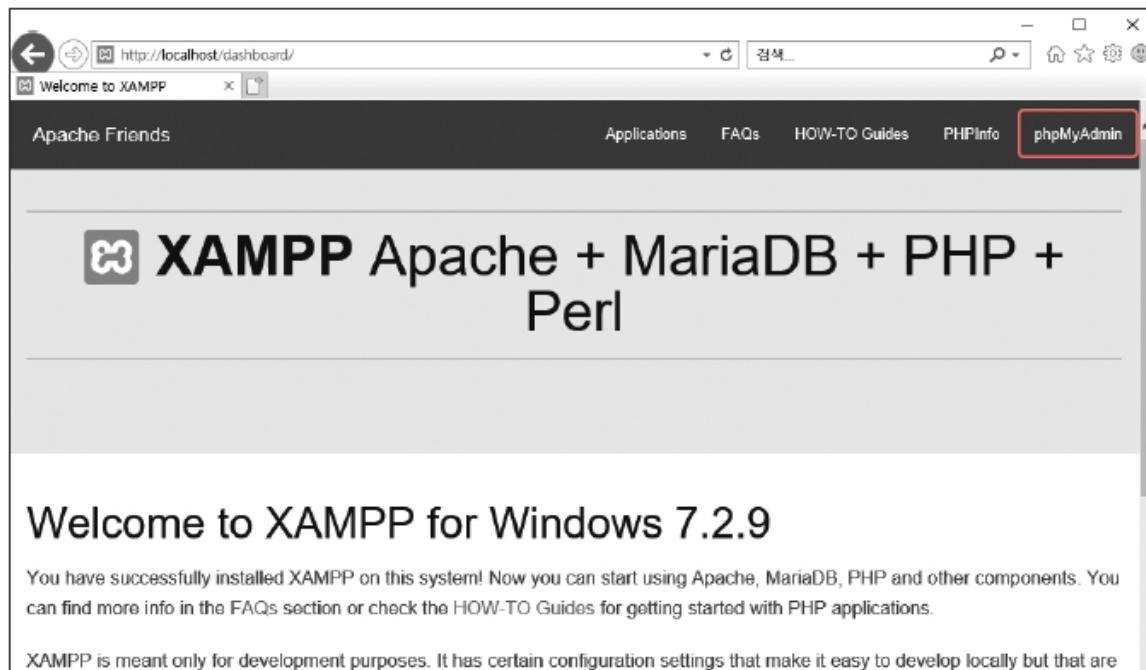


그림 9-6 XAMPP 메인 화면

## 02 MySQL 서버와 phpMyAdmin

- phpMyAdmin 페이지가 나타남. 이 화면에서 MySQL 실습을 진행.

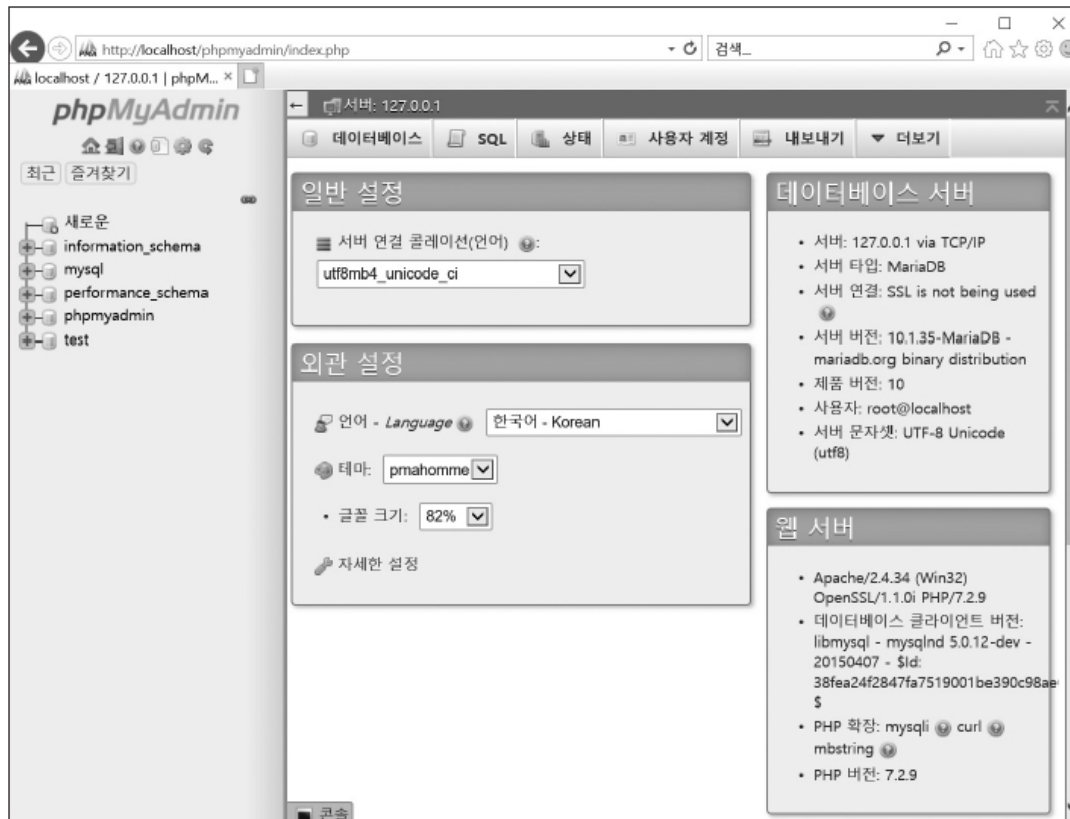


그림 9-7 phpMyAdmin 메인 화면

## 03 데이터베이스 생성

- phpMyAdmin을 이용하여 새로운 데이터베이스를 생성하기 위해 메인 화면 상단의 [데이터베이스] 메뉴를 클릭.
- '새 데이터베이스 만들기'에서 데이터베이스 이름과 데이터 정렬 방식을 다음과 같이 설정한 후 <만들기>를 클릭.
  - ✓ 데이터베이스 이름: sample
  - ✓ 데이터 정렬 방식: utf8\_general\_ci

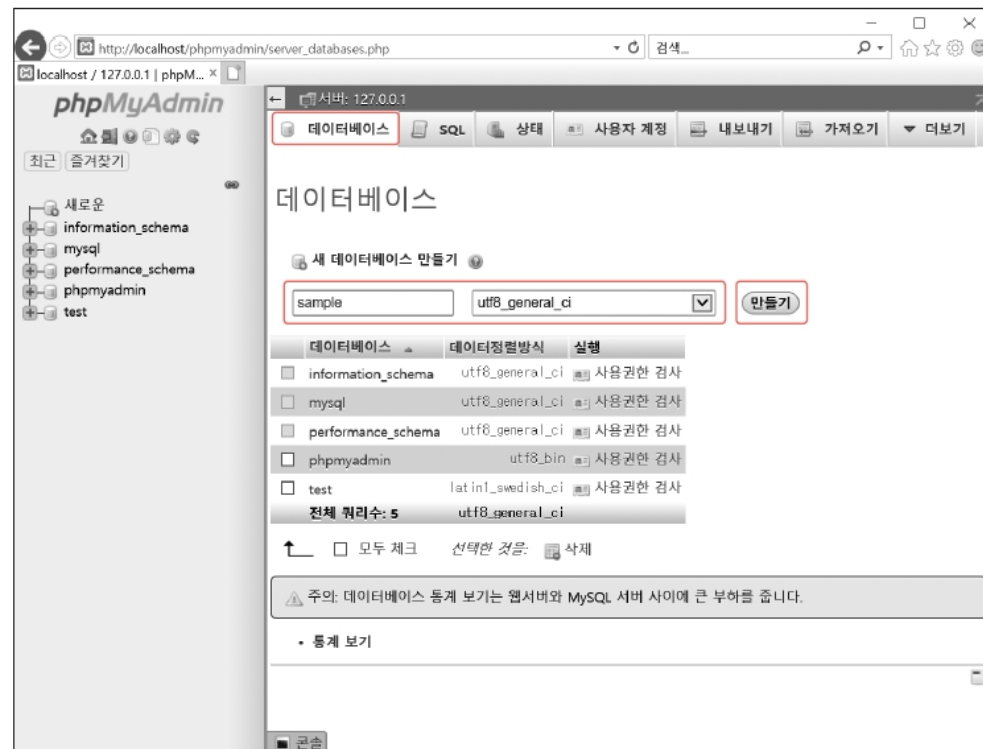


그림 9-8 새로운 데이터베이스 생성

- 화면 왼쪽 상단의 phpMyAdmin 로고를 클릭한 뒤 메인 화면의 [데이터베이스] 메뉴를 클릭하면 sample 데이터베이스가 생성된 것을 확인할 수 있음.

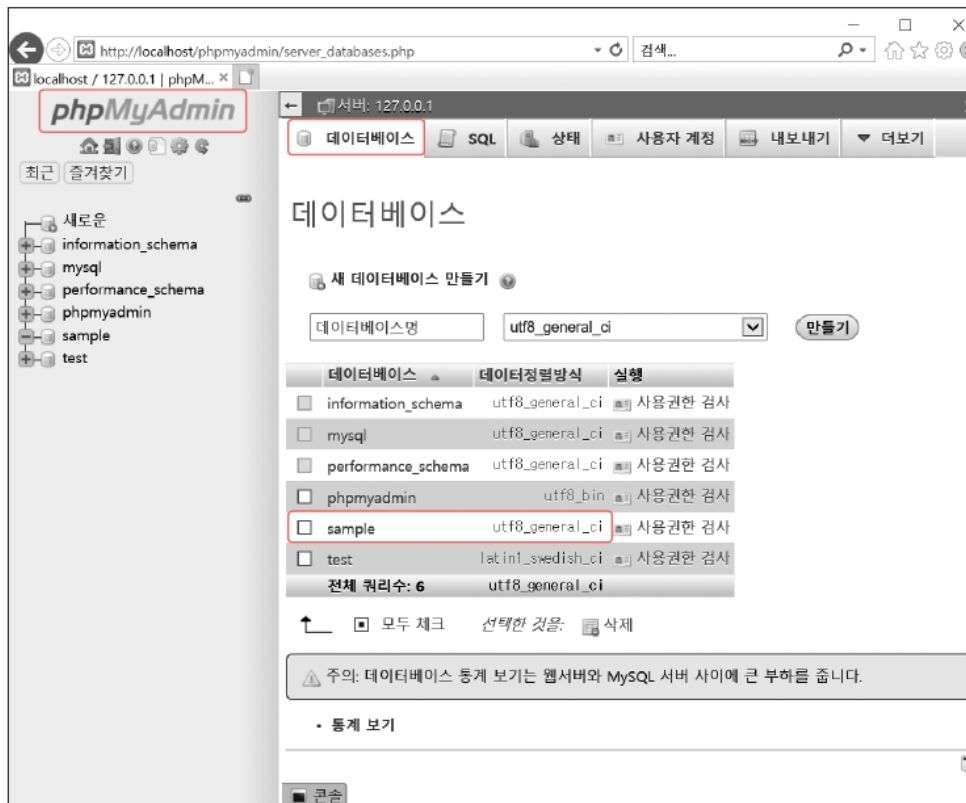


그림 9-9 sample 데이터베이스 생성

### 1. 데이터베이스 테이블 설계

- 회원 정보를 관리하기 위한 '회원 정보 테이블'을 설계. 테이블명은 mem.

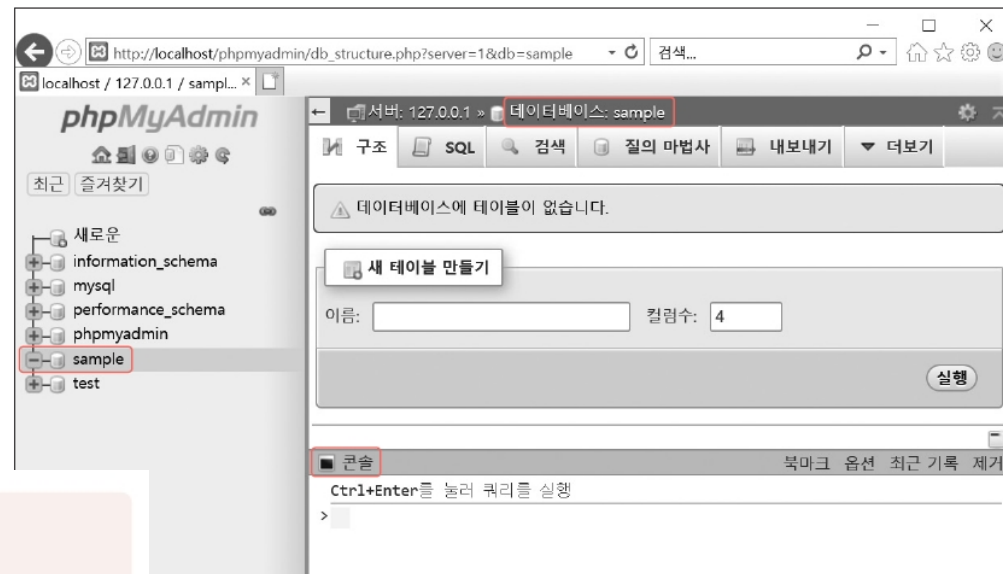
필드명	데이터형	null	추가 사항	설명
num	int	not null	auto_increment, primary key	일련번호
id	char(20)	not null		아이디
pass	char(20)	not null		비밀번호
name	char(20)	not null		이름
age	int			나이

- int - 해당 필드에 저장되는 데이터가 정수
- char(20) - 영문이나 숫자를 20자(한글은 10자)까지 저장
- null - 데이터 값이 없는 데이터형을 의미, " " 또는 ' '로 표기
- not null - 해당 필드를 비워두면 안 되고 반드시 값을 채워야 함. 회원 정보 테이블에서는 num, id, pass, name 필드가 필수 항목이므로 not null로 설정.
- auto\_increment : 레코드에 데이터를 입력할 때 값을 입력하지 않아도 0부터 1씩 값이 자동으로 증가하고 그 값이 num 필드에 저장된다는 의미.
- primary key(주 키)는 테이블의 필드 중 중복되면 안 되는 필드, 즉 레코드를 식별하는 데 사용되는 필드를 의미. DB 테이블을 설계할 때 auto\_increment가 적용된 필드가 있다면 반드시 그 필드를 주 키로 사용해야 함.

### 2. 데이터베이스 테이블 생성

```
create table 테이블명 (  
    필드명1 데이터형,  
    필드명2 데이터형,  
    필드명3 데이터형,  
    ...  
    primary key(필드명)  
);
```

```
> create table mem (  
    num int not null auto_increment,  
    id char(20) not null,  
    pass char(20) not null,  
    name char(20) not null,  
    age int,  
    primary key(num)  
);
```



phpMyAdmin 화면에서 sample 데이터베이스 선택

### 3. 데이터베이스 테이블의 구조

- desc 명령은 '테이블명'의 테이블 필드 구조를 보여줌.

desc 테이블명;

> desc mem;

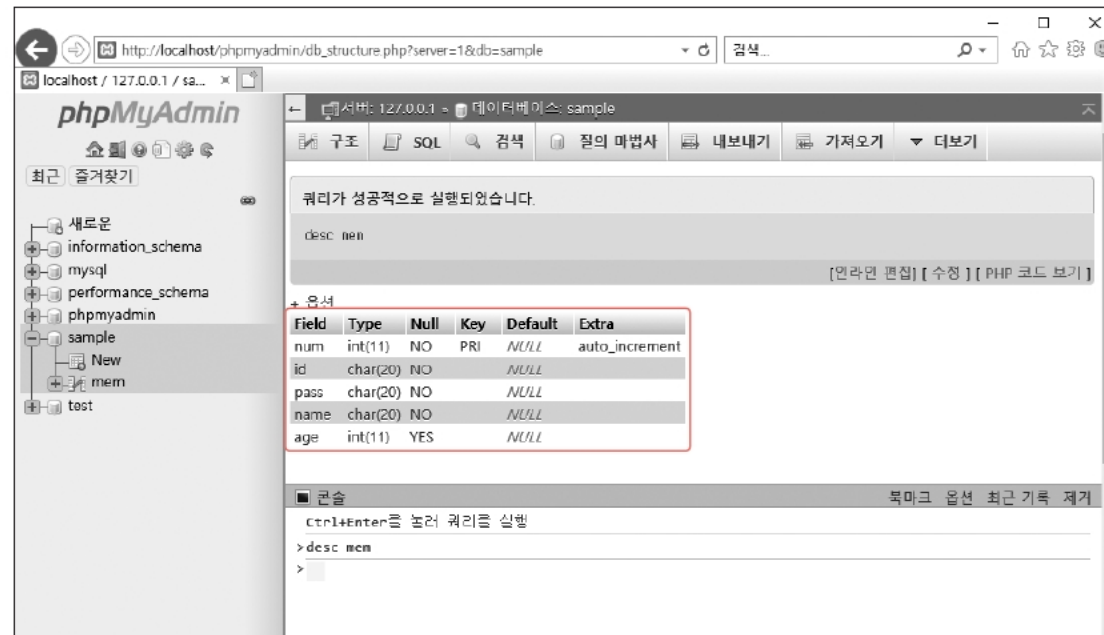


그림 9-12 mem 테이블의 구조

### 4. 데이터베이스 테이블 필드 관리

#### 4.1 새로운 필드 추가하기

- 테이블에 새로운 필드를 추가하는 명령.

```
alter table 테이블명 add 새로운_필드명 필드_데이터형 [first 또는 after 필드명];
```

```
> alter table mem add email char(30);
```

email 필드를 char(30) 데이터형으로 추가

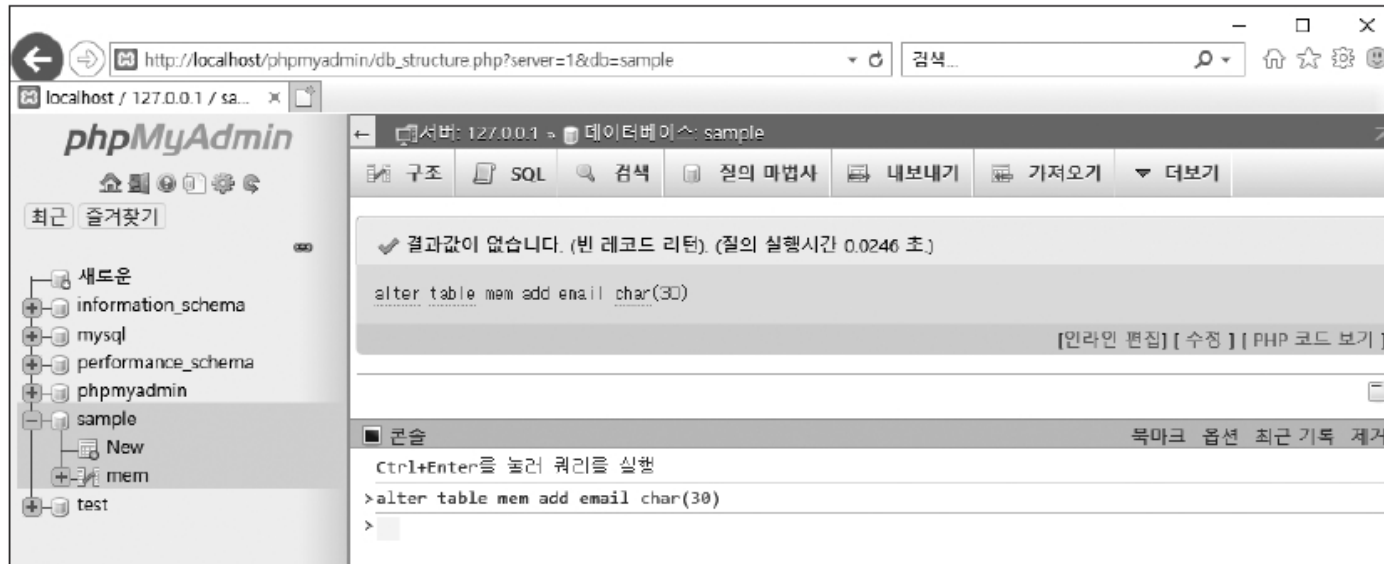


그림 9-13 mem 테이블에 email 필드 추가



- email 필드가 제대로 추가되었는지는 다음 명령으로 확인 가능.

> desc mem;

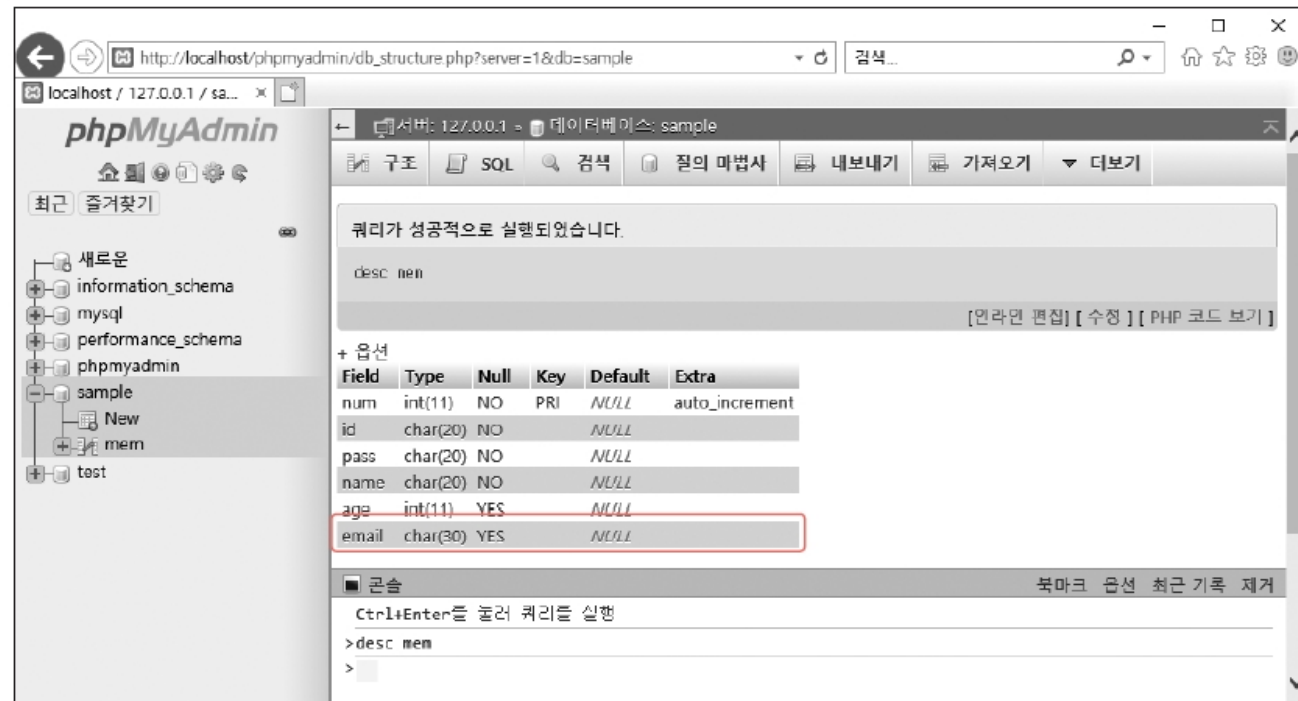


그림 9-14 mem 테이블에 추가된 email 필드 확인

### 4.2 필드 삭제하기

- 테이블에서 특정 필드를 삭제하는 명령.

```
alter table 테이블명 drop 삭제할_필드명1, 삭제할_필드명2;
```

- mem 테이블에서 email 필드를 삭제

```
> alter table mem drop email;  
> desc mem;
```

### 4.3 필드 수정하기

- 테이블의 특정 필드를 새로운 필드로 수정하는 명령.

```
alter table 테이블명 change 수정할_필드명 새로운_필드명 필드_데이터형;
```

- mem 테이블의 필드 중에서 age int를 phone char(20)으로 변경

```
> alter table mem change age phone char(20);  
> desc mem;
```

### 4.4 필드 데이터형 수정하기

- 테이블에서 특정 필드의 데이터형을 수정하는 명령.

```
alter table 테이블명 modify 필드명 수정할_데이터형;
```

- mem 테이블에서 phone 필드의 데이터형 char(20)을 int로 변경

```
> alter table mem modify phone int;  
> desc mem;
```

### 5. 데이터베이스 테이블 이름 관리

- 이미 존재하는 DB 테이블의 이름을 변경하는 명령.

```
alter table 수정할_테이블명 rename 새로운_테이블명;
```

```
> alter table mem rename mem2;
```

mem 테이블의 이름을 mem2로 변경.



그림 9-15 테이블 이름을 mem에서 mem2로 변경

### 6. 데이터베이스 테이블 삭제

- 데이터베이스의 테이블을 삭제하는 명령.

```
drop table 테이블명;
```

```
> drop table mem2;
```

mem2 테이블을 삭제하는 명령.



그림 9-16 mem2 테이블 삭제