



R E P O R T

Chapter 09

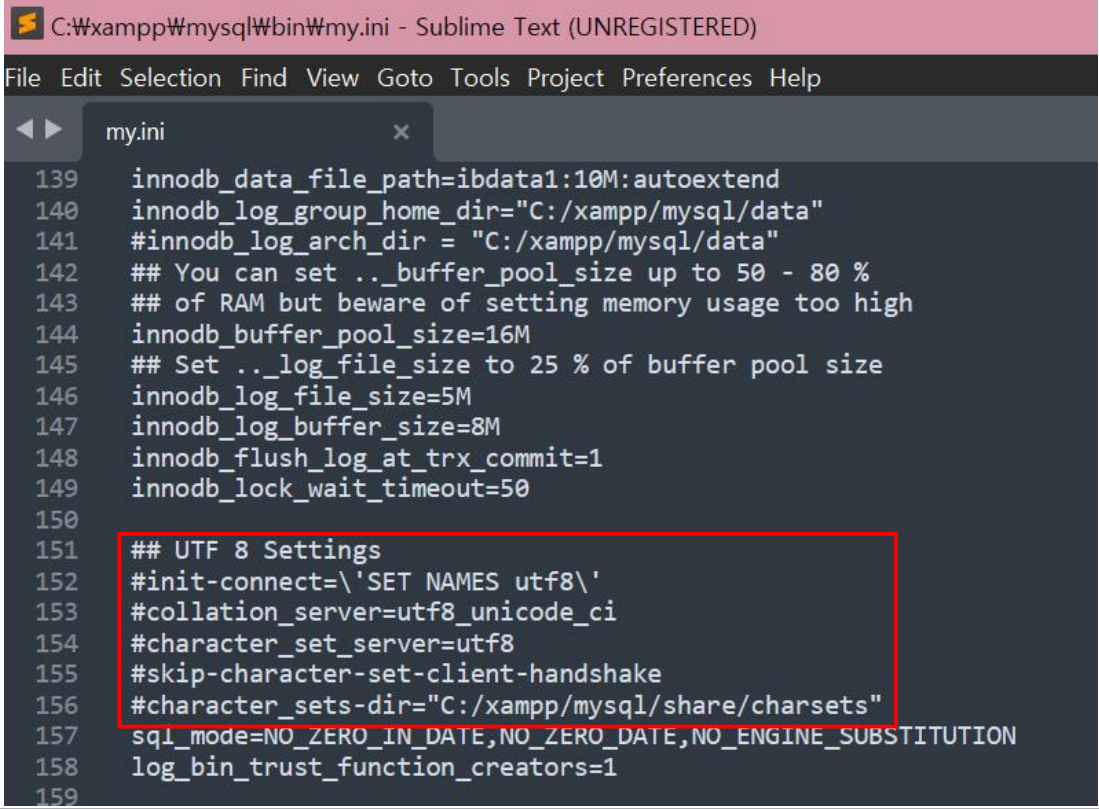
MySQL 데이터베이스

과목명	웹프로그래밍
분반	1 분반
교수	문 일 영
학번	2020136129
이름	최 수 연
제출일	2022년 5월 27일 금요일

1. MySQL 서버 구동

msvc140_codecvt_ids.dll	2021-10-23 오후 9:18	응용 프로그램 확장	20KB
my	2022-05-26 오후 2:57	구성 설정	6KB
my_print_defaults	2021-11-06 오전 6:15	응용 프로그램	3,301KB

파일 탐색기에서 C:\xampp\mysql\bin\my.ini 파일을 서버라임 텍스트로 불러옴



C:\xampp\mysql\bin\my.ini - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

my.ini

```
139 innodb_data_file_path=ibdata1:10M:autoextend
140 innodb_log_group_home_dir="C:/xampp/mysql/data"
141 #innodb_log_arch_dir = "C:/xampp/mysql/data"
142 ## You can set .._buffer_pool_size up to 50 - 80 %
143 ## of RAM but beware of setting memory usage too high
144 innodb_buffer_pool_size=16M
145 ## Set .._log_file_size to 25 % of buffer pool size
146 innodb_log_file_size=5M
147 innodb_log_buffer_size=8M
148 innodb_flush_log_at_trx_commit=1
149 innodb_lock_wait_timeout=50
150
151 ## UTF 8 Settings
152 #init-connect='SET NAMES utf8\'
153 #collation_server=utf8_unicode_ci
154 #character_set_server=utf8
155 #skip-character-set-client-handshake
156 #character_sets_dir="C:/xampp/mysql/share/charsets"
157 sql_mode=NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTION
158 log_bin_trust_function_creators=1
159
```

151~156줄 추가

2. 데이터베이스 생성

 새 데이터베이스 만들기 

▼ 만들기

	데이터베이스	데이터정렬방식	실행
<input type="checkbox"/>	information_schema	utf8_general_ci	 사용권한 검사
<input type="checkbox"/>	mysql	utf8mb4_general_ci	 사용권한 검사
<input type="checkbox"/>	performance_schema	utf8_general_ci	 사용권한 검사
<input type="checkbox"/>	phpmyadmin	utf8_bin	 사용권한 검사
<input type="checkbox"/>	sample	utf8_general_ci	 사용권한 검사
<input type="checkbox"/>	test	latin1_swedish_ci	 사용권한 검사

전체: 6

데이터베이스 이름: sample
데이터 정렬 방식: utf8_general_ci

3. 데이터베이스 테이블 생성

콘솔

Ctrl+Enter를 눌러 쿼리를 실행

>

```
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)  
);
```

콘솔 창을 열어 mem 테이블 생성 코드 입력

✓ 결과값이 없습니다. (빈 레코드 리턴). (질의 실행시간 0.0268 초.)

```
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) );
```

[인라인 편집] [수정] [PHP 코드 보기]

sample

New

mem

콘솔

북마크 옵션 최근 기록 제거

Ctrl+Enter를 눌러 쿼리를 실행

>

```
CREATE table mem ( num int not null auto_increment, id char(20) not null, pass ...
```

> |

쿼리 실행 결과 mem 테이블 생성됨

4. 데이터베이스 테이블 구조

쿼리가 성공적으로 실행되었습니다.

```
desc mem;
```

[[인리](#)

+ 옵션

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
id	char(20)	NO		NULL	
pass	char(20)	NO		NULL	
name	char(20)	NO		NULL	
age	int(11)	YES		NULL	

☐ 콘솔

Ctrl+Enter를 눌러 쿼리를 실행


```
>CREATE table mem ( num int not null auto_incremen
```

```
>desc mem;
```

```
>
```

‘desc mem;’명령을 통해 mem 테이블 구조 확인

5. 새로운 필드 추가


 콘솔

Ctrl+Enter를 눌러 쿼리를 실행

```
>CREATE table mem ( num int not null auto  
>desc mem;  
>alter table mem add email char(30);  
>
```

mem 테이블에 email 필드를 char(30) 데이터형으로 추가하는 명령

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
id	char(20)	NO		NULL	
pass	char(20)	NO		NULL	
name	char(20)	NO		NULL	
age	int(11)	YES		NULL	
email	char(30)	YES		NULL	

 콘솔

Ctrl+Enter를 눌러 쿼리를 실행

```
>CREATE table mem ( num int not null auto_increme  
>desc mem;  
>alter table mem add email char(30);  
>desc mem;
```

desc 명령을 통해 email 필드 생성 확인

6. 필드 삭제

콘솔

Ctrl+Enter를 눌러 쿼리를 실행

```
>CREATE table mem ( num int not null
>desc mem;
>alter table mem add email char(30);
>desc mem;
>alter table mem drop email;
```

mem 테이블에 email 필드를 삭제하는 명령

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
id	char(20)	NO		NULL	
pass	char(20)	NO		NULL	
name	char(20)	NO		NULL	
age	int(11)	YES		NULL	

콘솔

Ctrl+Enter를 눌러 쿼리를 실행

```
>CREATE table mem ( num int not null auto_incremer
>desc mem;
>alter table mem add email char(30);
>desc mem;
>alter table mem drop email;
>desc mem;
```

desc 명령을 통해 email 필드 삭제 확인

7. 필드 수정

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

mem 테이블에 age int를 phone char(20)으로 변경하는 명령

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
id	char(20)	NO		NULL	
pass	char(20)	NO		NULL	
name	char(20)	NO		NULL	
phone	char(20)	YES		NULL	

desc 명령을 통해 age 필드 변경 확인

8. 필드 데이터형 수정

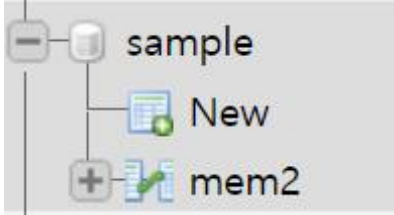
```
> alter table mem modify phone int;
```

mem 테이블에 phone 필드의 데이터 형 char(20)을 int로 변경하는 명령


Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
id	char(20)	NO		NULL	
pass	char(20)	NO		NULL	
name	char(20)	NO		NULL	
phone	int(11)	YES		NULL	

desc 명령을 통해 phone 필드 데이터형 수정 확인

9. 데이터베이스 테이블 이름 관리

<pre>> alter table mem rename mem2;</pre>	
mem 테이블의 이름을 mem2로 변경하는 명령	
	
mem2 테이블로 이름 변경됨	

10. 데이터베이스 테이블 삭제

<pre>> drop table mem2;</pre>	
mem2 테이블 삭제하는 명령	
	
mem2 테이블 삭제 확인	

연습문제 1.

<p>01. 다음은 MySQL에 관한 설명이다. 맞으면 O, 틀리면 X 표시를 하시오.</p> <p>① MySQL 데이터베이스 관리 시스템의 하나이다. (O) -> MySQL은 DBMS(DataBase Management System)의 일종이며, 이외에도 MSSQL, Oracle, Infomix 등 다양한 DBMS가 있다.</p> <p>② 주로 대형 서버 컴퓨터에 사용하고 중소 시스템이나 PC에는 잘 사용하지 않는다. (X) -> 예전에는 주로 대형 메인 프레임에 DBMS를 적용했지만 요즘은 HP, 오라클 등의 중소형 시스템과 PC에도 많이 사용되고 있다.</p> <p>③ MySQL은 테이블을 기본 구조로 하는 관계형 데이터베이스 관리 시스템에 속한다. (O) -> MySQL은 DBMS 중에서도 관계형 데이터베이스 관리 시스템(RDBMS)에 속한다.</p> <p>④ PHP로 웹 프로그래밍을 할 때 데이터베이스로 MySQL을 많이 사용한다. (O) -> MySQL은 PHP를 이용하여 웹 프로그램을 개발할 때 주로 사용하는 데이터베이스이다.</p> <p>⑤ SQL은 데이터베이스를 조회하거나 데이터를 갱신할 수 있는 데이터베이스 표준언어이다. (O) -> SQL은 데이터베이스의 데이터를 조회하거나 갱신할 수 있도록 만들어진 데이터베이스 표준언어이다. 대화 형식으로 데이터베이스에 직접 접근해서 사용하거나 PHP와 연동하여 PHP프로그램 내에서 SQL 명령을 사용할 수도 있다.</p> <p>⑥ phpMyAdmin의 문자셋은 세계 표준인 euc-kr로 설정하는 것이 바람직하다. (X) -> euc-kr은 대표적인 한글 완성형 인코딩이기 때문에, 세계 표준으로 euc-kr 방식을 사용하는 것은 바람직하지 않다. 따라서 세계 표준인 UTF-8 인코딩 방식을 사용하는 것이 바람직하다.</p> <p>⑦ MySQL은 처리 속도가 빠르고 성능이 우수한 편이지만 초보자가 익히기 어렵다. (X) -> MySQL은 DBMS 중에서도 처리 속도가 빠른 편이고, 설치 및 사용 방법이 쉬워 초보자가 쉽게 접근할 수 있다.</p>

연습문제 2.

02. MySQL 데이터베이스에서 DB 테이블을 관리하는 SQL 명령에 관한 물음에 답하시오.

① 데이터베이스를 생성하는 명령을 쓰시오.

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

② DB 테이블명이 member일 때 이 테이블의 구조를 보는 데 사용하는 명령을 쓰시오.

-> desc member;

// 테이블 필드 구조 확인 명령어: desc 테이블명;

③ member 테이블에 age 필드를 int 데이터형으로 추가하는 명령을 쓰시오.

-> alter table member add age int;

// 새 필드 추가 명령어: alter table 테이블명 add 새로운_필드명 필드_데이터형;

④ member 테이블에서 age 필드를 삭제하는 명령을 쓰시오.

-> alter table member drop age;

// 필드 삭제 명령어: alter table 테이블명 drop 삭제할_필드명;

⑤ member 테이블의 이름을 members로 변경하는 명령을 쓰시오.

-> alter table member rename members;

// 테이블명 변경 명령어: alter table 수정할_테이블명 rename 새로운_테이블명;

⑥ member 테이블의 name 필드를 char(20)으로 변경하는 명령을 쓰시오.

-> alter table member modify name char(20);

// 필드 데이터형 변경 명령어: alter table 테이블명 modify 필드명 수정할_데이터형;

연습문제 3.

03. 다음은 회사 사원의 인적 사항을 관리하기 위한 DB 테이블(테이블명: employee)의 구조이다.

필드명	데이터형	설명
num	int	일련번호
id	char(20)	아이디
name	char(10)	이름
gender	char(1)	성별(남성:M, 여성:F)
age	int	나이
department	char(20)	근무 부서
hp	char(20)	휴대전화 번호
address	char(100)	집 주소

① employee 테이블을 생성하는 명령의 빈칸을 채우시오.

```
create table employee(
    num int,
    id char(20),
    name char(10),
    gender char(1),
    age int
    department char(20),
    hp char(20),
    address char(100),
    primary key(num) );
```

② employee 테이블의 구조를 보는 데 사용하는 명령을 쓰시오.

-> desc employee;
// 테이블 필드 구조 확인 명령어: desc 테이블명;

③ employee 테이블에 email 필드를 char(30) 데이터형으로 추가하는 명령을 쓰시오.

-> alter table employee add email char(30);
// 새 필드 추가 명령어: alter table 테이블명 add 새로운_필드명 필드_데이터형;

④ employee 테이블의 이름을 member로 변경하는 명령을 쓰시오.

-> alter table employee rename member;
// 테이블명 변경 명령어: alter table 수정할_테이블명 rename 새로운_테이블명;

⑤ age 필드를 삭제하는 명령을 쓰시오.

-> alter table employee drop age;
// 필드 삭제 명령어: alter table 테이블명 drop 삭제할_필드명;

⑥ employee 테이블을 삭제하는 명령을 쓰시오.

-> drop table employee;

// 테이블 삭제 명령어: drop table 테이블명;

추가문제 1.

01. 다음은 축구 경기장의 정보를 관리하기 위한 DB 테이블 구조이다.

(테이블명 : STADIUM)

아래의 구조를 참고하여 STADIUM 테이블을 생성하는 stadium.sql

파일을 작성하여 코드와 결과화면을 보고서에 첨부하시오.

Primary key로 "stadium_id"를 사용한다.

필드명	데이터형	설명
Stadium_id	Char(3)	경기장 번호
Stadium_name	Varchar(40)	경기장 이름
Hometeam_id	Char(3)	홈팀 id
Seat_count	Int	관중석 수
Address	Varchar(60)	주소
Area_num	Varchar(3)	지역번호
tel	Varchar(10)	전화번호 (지역번호 제외)

```
CREATE TABLE STADIUM(
  Stadium_id char(3) not null comment '경기장번호',
  Stadium_name varchar(40) not null comment '경기장이름',
  Hometeam_id char(3) not null comment '홈팀id',
  Seat_count int not null comment '관중석수',
  Address varchar(60) not null comment '주소',
  Area_num varchar(3) not null comment '지역번호',
  tel varchar(10) not null comment '전화번호(지역번호제외)',
  primary key(Stadium_id)
);
```

Field	Type	Null	Key	Default	Extra
Stadium_id	char(3)	NO	PRI	NULL	
Stadium_name	varchar(40)	NO		NULL	
Hometeam_id	char(3)	NO		NULL	
Seat_count	int(11)	NO		NULL	
Address	varchar(60)	NO		NULL	
Area_num	varchar(3)	NO		NULL	
tel	varchar(10)	NO		NULL	

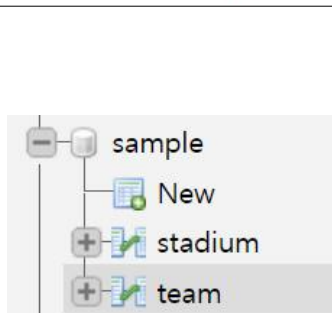
위 구조에 대한 쿼리문 작성 및 실행 후 테이블 생성 결과

추가문제 2.

02. 다음은 축구팀의 정보를 관리하기 위한 DB 테이블 구조이다.
(테이블명 : TEAM) 아래의 구조를 참고하여 TEAM 테이블을 생성하는
team.sql 파일을 작성하여 보고서에 코드와 결과화면을 첨부하시오.
Primary key로 “team_id”, Foreign key로 “stadium_id”를 사용한다.

필드명	데이터형	설명
Team_id	Char(3)	팀 일련번호
Region_name	Varchar(8)	지역명
Team_name	Varchar(40)	팀명
E_Team_name	Varchar(50)	팀 영문명
ORIG_YYYY	Char(4)	팀결성 연도
Stadium_id	Char(3)	홈구장 id
address	Varchar(80)	주소
Area_num	Varchar(3)	지역번호
tel	Varchar(10)	전화번호 (지역번호 제외)
fax	Varchar(10)	팩스번호

```
CREATE TABLE TEAM(  
    Team_id char(3) not null comment '팀일련번호',  
    Region_name varchar(8) not null comment '지역명',  
    Team_name varchar(40) not null comment '팀명',  
    E_Team_name varchar(50) not null comment '팀영문명',  
    ORIG_YYYY char(4) not null comment '팀결성연도',  
    Stadium_id char(3) not null comment '홈구장id',  
    address varchar(80) not null comment '주소',  
    Area_num varchar(3) not null comment '지역번호',  
    tel varchar(10) not null comment '전화번호(지역번호제외)',  
    fax varchar(10) not null comment '팩스번호',  
    Primary key(Team_id),  
    Foreign key(Stadium_id)  
    References stadium(Stadium_id)  
);
```



Field	Type	Null	Key	Default	Extra
Team_id	char(3)	NO	PRI	NULL	
Region_name	varchar(8)	NO		NULL	
Team_name	varchar(40)	NO		NULL	
E_Team_name	varchar(50)	NO		NULL	
ORIG_YYYY	char(4)	NO		NULL	
Stadium_id	char(3)	NO	MUL	NULL	
address	varchar(80)	NO		NULL	
Area_num	varchar(3)	NO		NULL	
tel	varchar(10)	NO		NULL	
fax	varchar(10)	NO		NULL	

위 구조에 대한 쿼리문 작성 및 실행 후 테이블 생성 결과

추가문제 3.

03. 2번 문제에서 생성한 team.sql을 수정하고자 한다.

아래의 조건에 맞는 명령어를 작성하고, 실행결과를 보고서에 첨부하시오.

- Team 테이블에 우편번호를 넣을 Zip_code 필드를 char(5) 데이터형으로 추가하시오.

```
> alter table team add Zip_code char(5);
```

- Team 테이블에 홈페이지 주소를 넣을 homepage 필드를 varchar(50) 데이터형으로 추가하시오.

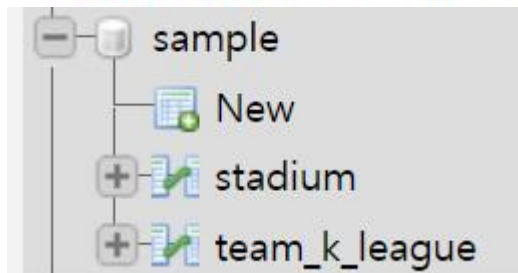
```
> alter table team add homepage varchar(50);
```

- Team 테이블에서 fax 필드를 삭제하시오.

```
> alter table team drop fax;
```

- Team 테이블의 이름을 team_k_league로 변경하시오.

```
> alter table team rename team_k_league;
```



- Team 테이블의 구조를 확인하시오.

```
> desc team_k_league;
```

Field	Type	Null	Key	Default	Extra
Team_id	char(3)	NO	PRI	NULL	
Region_name	varchar(8)	NO		NULL	
Team_name	varchar(40)	NO		NULL	
E_Team_name	varchar(50)	NO		NULL	
ORIG_YYYY	char(4)	NO		NULL	
Stadium_id	char(3)	NO	MUL	NULL	
address	varchar(80)	NO		NULL	
Area_num	varchar(3)	NO		NULL	
tel	varchar(10)	NO		NULL	
Zip_code	char(5)	YES		NULL	
homepage	varchar(50)	YES		NULL	