

6강

테이블 생성 및 데이터 입력



테이블 생성





➤ 데이터 정의어(DDL : Data Definition Language)

- 새로운 데이터베이스 구조를 정의하고 기존 데이터베이스 구조를 변경하는 명령어 집합
- 데이터베이스 구조를 표현하는 **데이터베이스 스키마(테이블이 모인 단위)**를 명세하기 위해 사용

DDL 명령어의 예 : **CREATE** -> 테이블 생성, **ALTER** -> 테이블 수정, **DROP** -> 테이블 삭제

➤ CREATE 형식

CREATE TABLE **[IF NOT EXISTS]** 테이블이름

(

옵션

{ 속성이름 데이터타입 [NOT **NULL** | UNIQUE | DEFAULT 기본값 | CHECK 체크조건] }

[PRIMARY KEY 속성이름(들)]

{ [FOREIGN KEY 속성이름 REFERENCES 테이블이름(속성이름)]

[ON DELETE [CASCADE | SET **NULL**] }

)



1. 테이블 생성/수정/삭제

- 작성할 테이블 구조

테이블이름	people	테이블 한글이름	사람			
한글컬럼명	Field	Type	Null	Key	Default	Extra
아이디	person_id	int				
이름	person_name	varchar(10)				
나이	age	tinyint				
생년월일	birthday	date				



1. 테이블 생성/수정/삭제 – 명령어 직접 입력

- **CREATE TABLE** - 테이블 만들기

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydatabase

Tables

Views

Stored Procedures

Functions

sakila

sys

world

더블클릭

```
1 CREATE TABLE people (  
2   person_id int,  
3   person_name varchar(10),  
4   age tinyint,  
5   birthday date  
6 );
```

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydatabase

Tables

people

Views

Stored Procedures

Functions

sakila

sys

world

```
1 • SELECT * FROM mydatabase.people;
```

Result Grid

person_id	person_name	age	birthday
-----------	-------------	-----	----------

Administration Schemas

Table: people

Columns:

Column	Type
person_id	int
person_name	varchar(10)
age	tinyint
birthday	date

Output

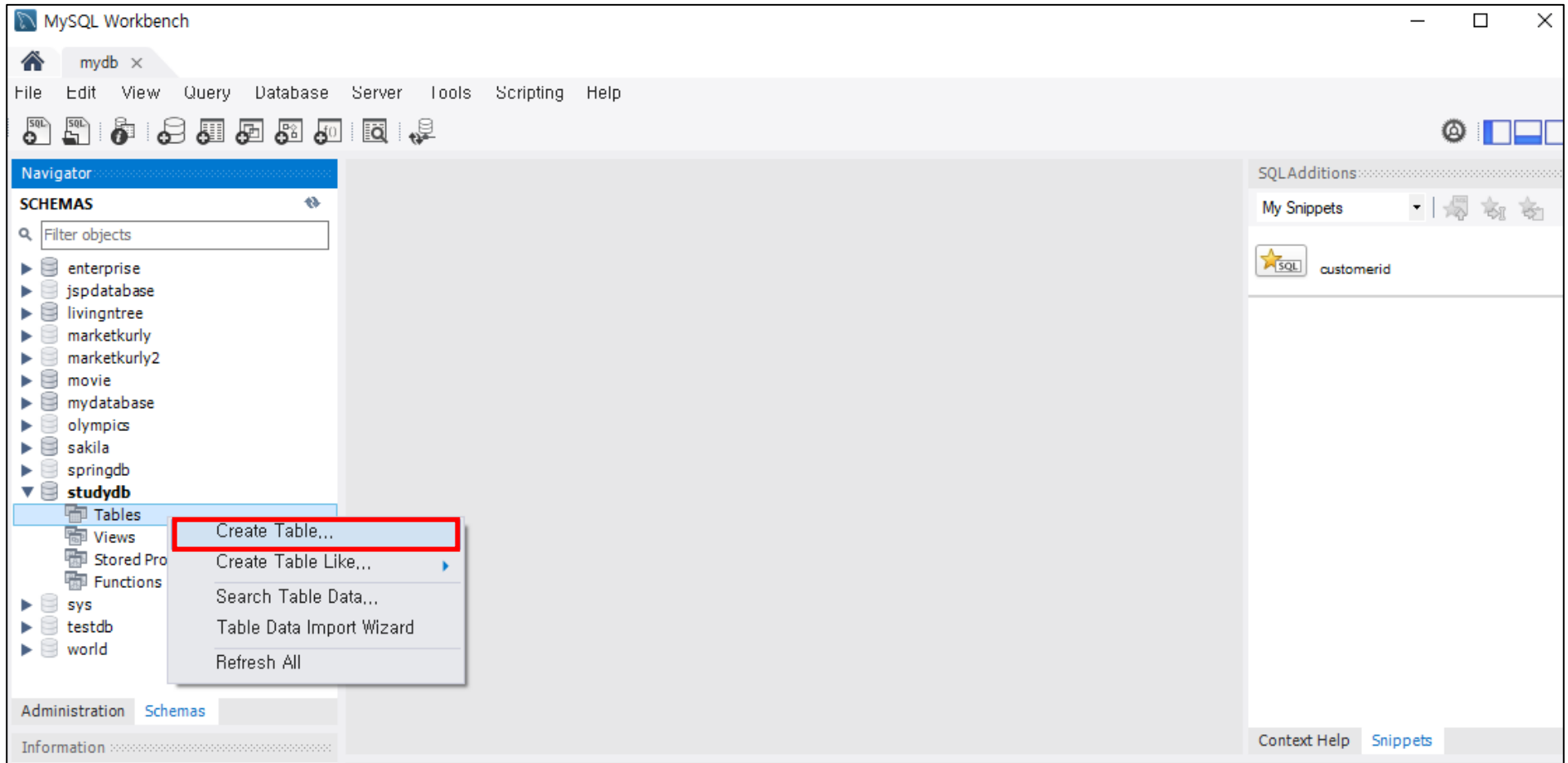
Action Output

#	Time	Action	Message	Duration / Fetch
1	19:32:15	CREATE TABLE people (person_id int, person_name varchar(10), age ...	0 row(s) affected	0.047 sec



1. 테이블 생성/수정/삭제 – 메뉴를 이용하여

1.CREATE TABLE 메뉴 클릭한다.





1. 테이블 생성/수정/삭제 – 메뉴를 이용하여

2. Table Name, Charset/Collation 아래처럼 설정 한다.

Column Name에 필드명을 직접 입력하고, Datatype도 직접 선택한다. -> Apply 클릭

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- enterprise
- jspdatabase
- livingntree
- marketkurly
- marketkurly2
- movie
- mydatabase
- olympics
- sakila
- springdb
- studydb**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
- testdb
- world

people - Table x

Table Name: people Schema: studydb

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
person_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
person_name	VARCHAR(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"/>	
age	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
birthday	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"/>	

Column Name: Data Type: Expression:

Charset/Collation: Default Charset Default Collation

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Administration Schemas Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert Context Help Snippets



1. 테이블 생성/수정/삭제 – 메뉴를 이용하여

3. Apply 버튼 클릭한다.

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm: Default Lock Type: Default

```
1 CREATE TABLE `studydb`.`people` (  
2   `person_id` INT NULL,  
3   `person_name` VARCHAR(20) NULL,  
4   `age` TINYINT NULL,  
5   `birthday` DATE NULL)  
6 ENGINE = InnoDB  
7 DEFAULT CHARACTER SET = utf8mb4;  
8
```

< >

Back

Apply

Cancel



1. 테이블 생성/수정/삭제 – 메뉴를 이용하여

4. 아래와 같이 people 테이블이 생성되어 있음을 확인 할 수 있다.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view of databases. The 'studydb' database is expanded, and the 'people' table is highlighted with a red box. In the center, the SQL editor shows the query '1 • SELECT * FROM people;' highlighted with a red box. Below the editor, the 'Result Grid' shows the columns 'person_id', 'person_name', 'age', and 'birthday'. On the right, the 'SQLAdditions' pane shows 'My Snippets' with a snippet named 'customerid'.



👉 여기서 잠깐 !!

아래 테이블 구조를 참조하여 greendb 스키마안에 <member> 테이블을 생성하시오.

테이블 구조

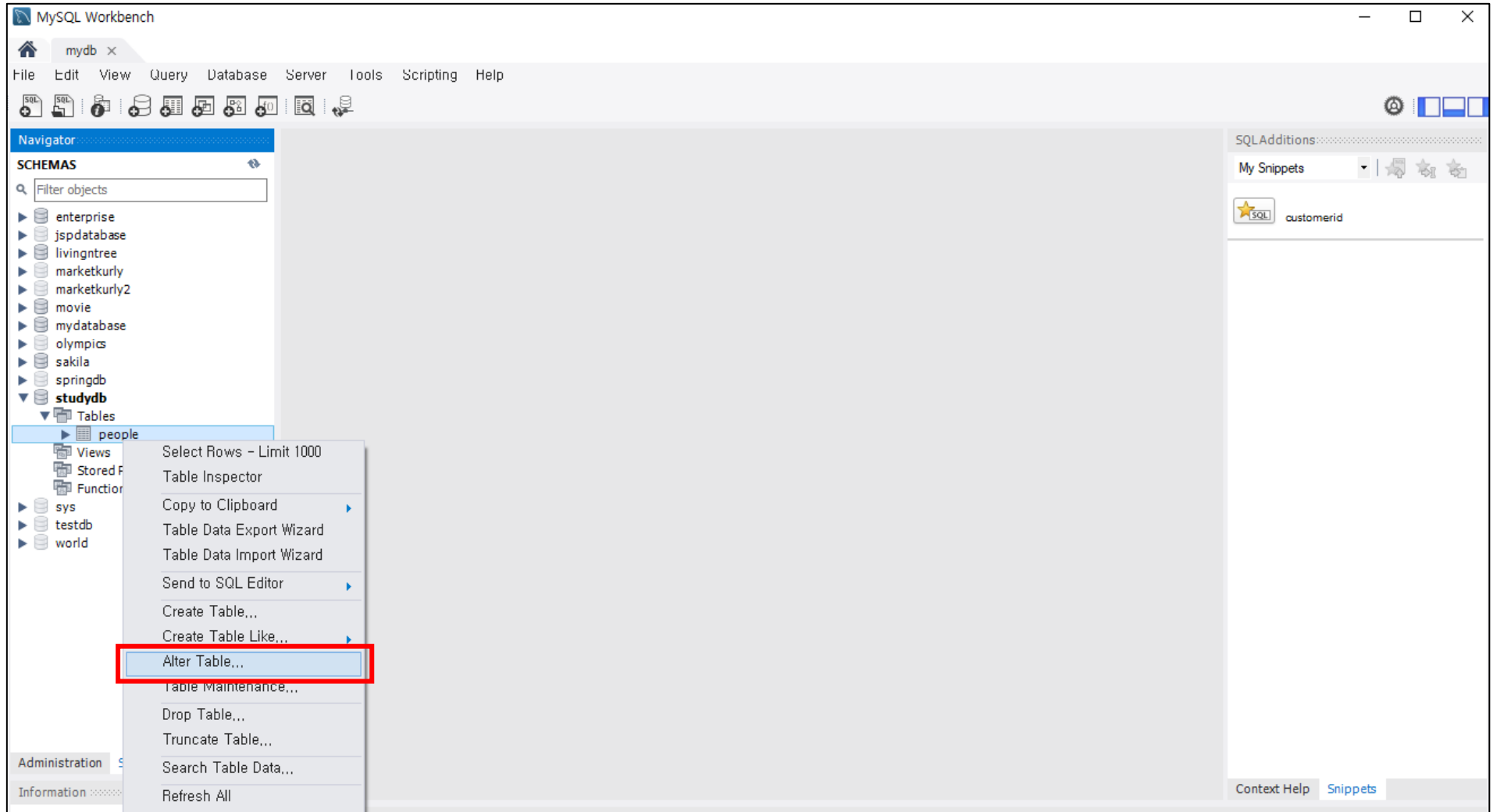
테이블이름	member	테이블 한글이름	회원			
한글컬럼명	Field	Type	Null	Key	Default	Extra
아이디	user_id	varchar(20)				
비밀번호	user_pw	varchar(20)				
이름	user_name	varchar(10)				
가입일	Reg_date	date				

테이블 수정





1. 테이블위에서 오른쪽 버튼 -> **Alter Table** 메뉴 클릭





2. 테이블위에서 오른쪽 버튼 -> **Alter Table** 메뉴 클릭 후 아래 테이블에 여러가지 속성을 수정 할 수 있다.

MySQL Workbench Schemas window showing the 'people' table structure. The table is located in the 'studydb' schema. The table structure is as follows:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
person_id	INT	<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"/>	
person_name	VARCHAR(20)	<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"/>	
age	TINYINT	<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"/>	
birthday	DATE	<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"/>	

The 'Apply' button at the bottom right is highlighted with a red box.



3. 수정이 완료된 후 **Apply** 버튼 클릭한다.

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm: Lock Type:

```
1 ALTER TABLE `studydb`.`people`  
2 CHANGE COLUMN `person_id` `person_id` INT NOT NULL ,  
3 CHANGE COLUMN `person_name` `person_name` VARCHAR(20) NOT NULL ,  
4 CHANGE COLUMN `age` `age` TINYINT NOT NULL ,  
5 CHANGE COLUMN `birthday` `birthday` DATE NOT NULL ,  
6 ADD PRIMARY KEY (`person_id`);  
7 ;  
8
```

< >

Back Apply Cancel



1. 테이블에 새로운 컬럼 추가하기

```
ALTER TABLE [테이블명] ADD COLUMN [추가할 컬럼명][추가할 컬럼 데이터형]
```

```
ALTER TABLE user_info ADD COLUMN name varchar(20) not null;
```

2. 테이블 컬럼 타입 변경하기

```
ALTER TABLE [테이블명] MODIFY COLUMN [변경할 컬럼명][변경할 컬럼 타입]
```

```
ALTER TABLE user_info MODIFY COLUMN name varchar(10) not null;
```

3. 테이블 컬럼 이름 변경하기

```
ALTER TABLE [테이블명] CHANGE COLUMN [기존 컬럼 명][변경할 컬럼 명][변경할 컬럼 타입]
```

```
ALTER TABLE user_info CHANGE COLUMN name nickname varchar(20) not null;
```

4. 테이블 컬럼 삭제하기

```
ALTER TABLE [테이블명] DROP COLUMN [삭제할 컬럼 명]
```

```
ALTER TABLE user_info DROP COLUMN nickname;
```

5. 테이블 컬럼에 인덱스 설정하기

```
ALTER TABLE [테이블명] ADD INDEX 인덱스명(인덱스 설정할 컬럼1, 인덱스 설정할 컬럼2, ...)
```

```
ALTER TABLE user_info ADD INDEX userindex(id);
```



6. 설정된 인덱스 삭제하기

```
ALTER TABLE [테이블명] DROP INDEX [인덱스명]
```

```
ALTER TABLE user_info DROP INDEX userindex;
```

7. Primary Key 설정하기

```
ALTER TABLE [테이블명] ADD PRIMARY KEY (Key 설정할 컬럼 명1, Key 설정할 컬럼 명 2, ...)
```

```
ALTER TABLE user_info ADD PRIMARY KEY (id, nickname);
```

8. Primary Key 삭제하기

```
ALTER TABLE [테이블명] DROP PRIMARY KEY
```

```
ALTER TABLE user_info DROP PRIMARY KEY;
```

9. 테이블 이름 바꾸기

```
ALTER TABLE [원본 테이블] RENAME [새로운 테이블 이름]
```

```
ALTER TABLE user_info RENAME vip_user_info;
```




1. 테이블 생성/수정/삭제

- **ALTER TABLE** - 테이블 변경
- 한번에 여러가지 수정 가능

The screenshot shows the MySQL Workbench interface with the 'people' table selected in the Navigator. The SQL editor contains the following script:

```
1  -- 테이블명 변경
2  • ALTER TABLE people RENAME TO friends,
3  -- 컬럼 자료형 변경
4  CHANGE COLUMN person_id person_id TINYINT,
5  -- 컬럼명 변경
6  CHANGE COLUMN person_name person_nickname VARCHAR(10),
7  -- 컬럼 삭제
8  DROP COLUMN birthday,
9  -- 컬럼 추가 after age => age필드 다음에 is_married 삽입하라
10 -- 맨앞에 컬럼 추가 first
11 -- 맨뒤에 컬럼 추가원하면 아무것도 적지 않는다.
12 ADD COLUMN is_married TINYINT AFTER age;
```

The 'Table: people' information panel on the left shows the current columns: person_id (int), person_name (varchar(10)), age (tinyint), and birthday (date).

The Output window at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
✓ 1	19:32:15	CREATE TABLE people (person_id int, person_name varchar(10), age tinyint, birthday ...	0 row(s) affected	0.047 sec
✓ 2	19:35:34	SELECT * FROM mydatabase.people LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✓ 3	19:36:07	select * from people LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✓ 4	20:05:35	SELECT * FROM mydatabase.people LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✓ 5	20:11:31	ALTER TABLE people RENAME TO friends, -- 컬럼 자료형 변경 CHANGE COLUMN ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec

The screenshot shows the MySQL Workbench interface with the 'friends' table selected in the Navigator. The SQL editor contains the following query:

```
1 • SELECT * FROM mydatabase.friends;
```

The Result Grid at the bottom shows the columns: person_id, person_nickname, age, and is_married.



1. 테이블 생성/수정/삭제

- **ALTER TABLE** - 테이블 변경
- 한번에 한가지만 수정

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'mydatabase' expanded, showing tables like 'friends'. The main editor window contains the SQL query: `1 alter table friends modify column person_nickname varchar(20);`. The bottom 'Output' panel shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
✓ 5	20:11:31	ALTER TABLE people RENAME TO friends, -- 컬럼 자르할 변경 CHANGE COLUMN...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
✓ 6	20:13:29	SELECT * FROM mydatabase.friends LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✗ 7	20:22:00	alter table friends change modify person_nickname varchar(20)	Error Code: 1054. Unknown column 'modify' in 'friends'	0.000 sec
✓ 8	20:22:43	alter table friends modify column person_nickname varchar(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
✓ 9	20:22:55	SELECT * FROM mydatabase.friends LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec



※ 본인이 보고싶은 테이블 선택 후 오른쪽 버튼 클릭 -> **Table Inspector** 선택 하여 컬럼 정보를 확인 할 수 있다.

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydatabase

Tables

friends

Table Inspector

Info Columns Indexes Triggers Foreign keys Partitions Grants DDL

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
person_id	tinyint		YES			select,insert,update,references
person_nickname	varchar(20)		YES	utf8mb4	utf8mb4_genera...	select,insert,update,references
age	tinyint		YES			select,insert,update,references
is_married	tinyint		YES			select,insert,update,references

Count: 4

Refresh



1. 테이블 생성/수정/삭제

- DROP TABLE - 테이블 삭제**

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'mydatabase' expanded, showing 'Tables' and 'friends'. The main editor window shows the SQL command: `DROP TABLE friends;`. The bottom 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
✓ 5	20:11:31	ALTER TABLE people RENAME TO friends, -- 컬럼 자료형 변경 CHANGE COLUMN...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
✓ 6	20:13:29	SELECT * FROM mydatabase.friends LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✗ 7	20:22:00	alter table friends change modify person_nickname varchar(20)	Error Code: 1054. Unknown column 'modify' in 'Friends'	0.000 sec
✓ 8	20:22:43	alter table friends modify column person_nickname varchar(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
✓ 9	20:22:55	SELECT * FROM mydatabase.friends LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec



👉 여기서 잠깐 !!

greendb 스키마안에 <member>테이블을 아래 조건대로 수정하시오.

<조건>

1. <member>테이블 이름을 <register>로 변경하시오
2. reg_date필드명 이전에 email 필드명을 추가하시오.
3. 추가할 email필드명의 데이터 타입은 varchar(50)으로 지정하시오.

테이블 구조

Info	Columns	Indexes	Triggers	Foreign keys	Partitions	Grants	DDL	
Column	Type	Default Value	Nullable	Character Set	Collation	Privileges		
◇ user_id	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		
◇ user_pw	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		
◇ user_name	varchar(10)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		
◇ email	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		
◇ reg_date	date		YES			select,insert,update,references		

테이블 데이터 삽입





2. INSERT INTO - 데이터 삽입

1. **INSERT INTO** 테이블이름(필드이름1, 필드이름2, 필드이름3, ...)

VALUES (데이터값1, 데이터값2, 데이터값3, ...)

2. **INSERT INTO** 테이블이름

VALUES (데이터값1, 데이터값2, 데이터값3, ...)

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'mydatabase' schema with a 'people' table. The main editor shows the following SQL query:

```
1 insert into people (person_id, person_name, age, birthday)
2 values (1, '홍길동', 21, '2000-01-31');
3
```

Below the query editor, the 'Result Grid' is displayed, showing the result of the query:

	person_id	person_name	age	birthday
▶	1	홍길동	21	2000-01-31



- 모든 컬럼에 값을 입력할 때 컬럼명들 생략 가능

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator: friends people

SCHEMAS

Filter objects

mydatabase

- Tables
 - people
- Views
- Stored Procedures
- Functions

sakila

sys

world

Limit to 1000 rows

```
1 -- 모든 컬럼에 값 넣을 때는 컬럼명들 생략 가능
2 -- 단, 반드시 필드명 순서대로 입력해야 함
3 • INSERT INTO people
4   VALUES (2, '전우치', 18, '2003-05-12');
5
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	person_id	person_name	age	birthday
▶	1	홍길동	21	2000-01-31
	2	전우치	18	2003-05-12



- 일부 컬럼에만 값을 넣기 가능 (**not null은 생략 불가**)

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator: friends x people

Limit to 1000 rows

```
1 -- 일부 컬럼에만 값 넣기 가능 (NOT NULL은 생략 불가)
2 • INSERT INTO people
3   (person_id, person_name, birthday)
4   VALUES (3, '임꺽정', '1995-11-04');
5
```

Administration Schemas

Information

Table: people

Columns:

	person_id	person_name	age	birthday
▶	1	홍길동	21	2000-01-31
	2	전우치	18	2003-05-12
	3	임꺽정	NULL	1995-11-04



- 자료형에 맞지 않는 데이터는 오류 발생

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydatabase

Tables

people

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

Table: people

Columns:

person_id int

person_name varchar(10)

age tinyint

birthday date

friends x people

Limit to 1000 rows

```
1 -- 자료형에 맞지 않는 값은 오류 발생
2 INSERT INTO people
3 (person_id, person_name, age, birthday)
4 VALUES (1, '임격정', '스물여섯', '1995-11-04');
```

Output

Action Output

#	Time	Action	Message
✓ 13	20:46:59	drop table friends	0 row(s) affected
✓ 14	20:47:14	CREATE TABLE people (person_id INT, person_name VARCHAR(10), age TINY...	0 row(s) affected
✓ 15	20:49:56	insert into people (person_id, person_name, age, birthday) values (1,'홍길동',21,'2000-...	1 row(s) affected
✓ 16	20:52:03	SELECT * FROM mydatabase.people LIMIT 0, 1000	1 row(s) returned
✓ 17	20:55:17	INSERT INTO people VALUES (2, '전우치', 18, '2003-05-12')	1 row(s) affected
✓ 18	20:55:21	SELECT * FROM mydatabase.people LIMIT 0, 1000	2 row(s) returned
✓ 19	21:00:49	INSERT INTO people (person_id, person_name, birthday) VALUES (3, '임격정', '19...	1 row(s) affected
✓ 20	21:00:50	SELECT * FROM mydatabase.people LIMIT 0, 1000	3 row(s) returned
✗ 21	21:02:15	INSERT INTO people (person_id, person_name, age, birthday) VALUES (1, '임격정...	Error Code: 1366. Incorrect integer value: '스물여섯' for column 'age' at row 1



- 여러 행을 한 번에 입력 가능

The image shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mydatabase' expanded, showing tables like 'people'. The main editor window contains an SQL query to insert data into the 'people' table. The query is as follows:

```
1  -- 여러 행을 한 번에 입력 가능
2  INSERT INTO people
3    (person_id, person_name, age, birthday)
4  VALUES
5    (4, '존 스미스', 30, '1991-03-01'),
6    (5, '루피 D. 몽키', 15, '2006-12-07'),
7    (6, '황비홍', 24, '1997-10-30');
```

Below the query editor, a 'Result Grid' window is open, displaying the results of the query. The grid has columns for 'person_id', 'person_name', 'age', and 'birthday'. The data is as follows:

	person_id	person_name	age	birthday
▶	1	홍길동	21	2000-01-31
	2	전우치	18	2003-05-12
	3	임꺽정	HULL	1995-11-04
	4	존 스미스	30	1991-03-01
	5	루피 D. 몽키	15	2006-12-07
	6	황비홍	24	1997-10-30



※ Workbench의 테이블에 데이터를 직접 입력하려면 반드시 **Primary key(기본키)**가 지정되어야 함

MySQL Workbench

mydb x

File Edit View Query Database Server Tools Scripting Help

Navigator

SQL File 4 people x

Limit to 1000 rows

1 • SELECT * FROM mydb.people;

데이터 직접입력

Result Grid

person_id	person_name	age	birthday
NULL	NULL	NULL	NULL

Administration Schemas

Result Grid Form Editor Field Types Query Stats



※ 자료입력이 완료 되면 반드시 **Apply** 버튼을 클릭해야 테이블에 데이터가 저장된다.

MySQL Workbench interface showing the 'people' table in the 'mydb' database. The table contains the following data:

person_id	person_name	age	birthday
1	홍길동	21	2000-01-31
2	전우치	18	2003-05-12

The 'Apply SQL Script to Database' dialog box is open, showing the SQL script to be applied:

```
1 INSERT INTO `mydb`.`people` (`person_id`, `person_name`, `age`, `birthday`) \
2 INSERT INTO `mydb`.`people` (`person_id`, `person_name`, `age`, `birthday`) \
3
```

The 'Apply' button in the dialog box is highlighted with a red box. A red dotted line connects this button to the 'Apply' button in the bottom toolbar.



👉 여기서 잠깐 !!

greendb 스키마안에 <register> 테이블에 아래 <출력 결과>와 같이 데이터를 삽입하시오. (단, insert into를 이용할 것)

출력 결과

	user_id	user_pw	user_name	email	reg_date
	kjb1234	1234	개나리	kjb1@naver.com	2024-05-10
	park9800	9800	진달래	park9@daum.net	2024-02-28
✎	jln3456	3456	최진솔	jln34@goole.com	2023-9-17
✱	NULL	NULL	NULL	NULL	NULL



📁 데이터 무결성이란?

데이터 무결성은 데이터베이스에 저장된 데이터의 **정확성, 일관성, 유효성**을 지키는 것이다. 보통 데이터 무결성은 제약조건으로 데이터베이스 시스템이 강제한다

📁 제약조건(constraint)이란?

제약조건(**constraint**)이란 데이터의 무결성을 지키기 위해, 데이터를 입력받을 때 실행되는 검사 규칙을 의미한다. 이러한 제약 조건은 **CREATE** 문으로 테이블을 생성할 때나 **ALTER** 문으로 컬럼을 추가할 때도 설정할 수 있다.

📁 제약조건(constraint) 종류

제약조건 명령어	단축	설명
AUTO_INCREMENT	AI	새 행 생성시 마다 자동으로 1씩 증가
PRIMARY KEY	PK	중복 입력 불가, NULL(빈 값) 불가
FOREIGN KEY	FK	외래키로서 하나의 테이블을 다른 테이블에 의존하게 만든다
UNIQUE	UQ	중복 입력 불가
NOT NULL	NN	NULL(빈 값) 입력 불가
UNSIGNED	UN	(숫자입력시) 양수만 가능
DEFAULT	Default/Expression	값 입력이 없을 시 기본값
Binary	B	데이터를 이진 문자열((010101)로 저장함
Zero Filled	ZF	열 크기보다 작은 값을 넣은 경우 0으로 채운 뒤 삽입시킴 INT(5)인 경우 모든 필드는 5번째 자리까지 0으로 채워짐 예시 : 11 = 00011, 864 = 00864 등으로 0으로 채워짐
Generated Column	G	다른 열을 기반으로 한 수식으로 생성된 값

※ default 값으로 오늘날짜, 시간을 지정할 때는 (**current_date**), (**current_time**) 반드시 괄호안에 함수 입력, now(), current_timestamp()



💣* PRIMARY KEY (기본키)

- **Primary key**는 특정 컬럼내에 데이터가 유일(=고유) 하도록 보장해주는데 사용한다.
- 관계형 데이터베이스 테이블 내에 기록을 유일(=고유)하게 식별한다.
- 테이블 내에 하나의 **primary key**만 허락한다.
- **Unique(중복 불가)와 Not Null** 제약의 결합이다.
- **Null** 값을 허락하지 않는다.
- 기본적으로 인덱스 생성 (기본키 행 기준으로 빠른 검색 가능)
- 보통 **AUTO_INCREMENT**와 함께 사용
- **Primary key**는 동시에 2개 이상 필드명을 결합하여 지정하는 것이 가능하다.
예) (A, B) 이렇게 두 개 이상의 컬럼을 묶어서 하나의 Primary key로 설정하는 → 이걸 복합 Primary Key라고 한다.

💣* FOREIGN KEY (외래키)

- **Foreign key**는 두 테이블 사이에 데이터를 연결 시켜주는 관계형 데이터베이스 테이블에 컬럼이다.
- 다른 테이블의 **primary key**를 참조 할 수 있다.
- 테이블 내에 한 개 이상의 **foreign key**가 허락된다.
- 관계형 데이터베이스 내에 중복 값과 테이블을 포함할 수 있다.
- **Null**값이 가능하다.



Primary key

CUSTOMER table

customerid	First_name	Last_name	email	phone
1	JOHN	BONHAN	john@gmail.com	020394984
2	DAVE	GROHI	dave@yahoo.com	920930938
3	ROBERT	SMITH	robert@webseo.com	987321879
4	FRANK	BLAKE	frank@iamea.com	987365463

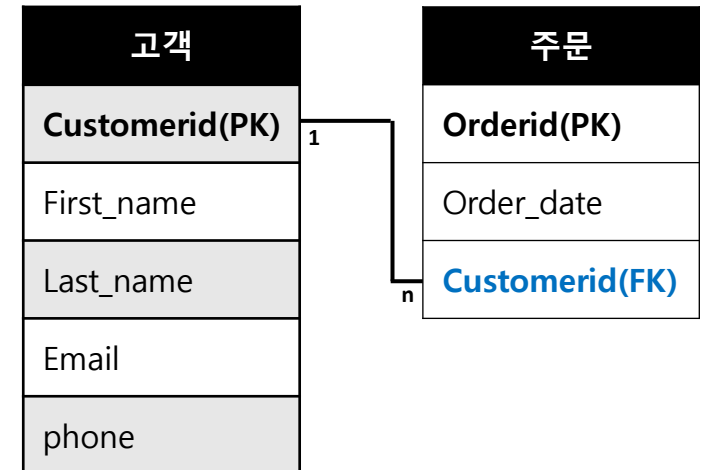
Primary key

ORDER table

orderid	Order_date	Customerid
1	2019-09-13	2
2	2019-10-13	2
3	2019-10-13	3
4	2019-09-13	3
5	2019-08-13	4

Foreign key

<관계설정>





3. 테이블 생성시 제약 넣기

- People 테이블 삭제하고 제약 넣어 다시 작성

The screenshot displays the MySQL Workbench interface. The top window shows the SQL editor with the following code:

```
1 CREATE TABLE people (  
2     person_id INT AUTO_INCREMENT PRIMARY KEY,  
3     person_name VARCHAR(10) NOT NULL,  
4     nickname VARCHAR(10) UNIQUE NOT NULL,  
5     age TINYINT UNSIGNED,  
6     is_married TINYINT DEFAULT 0  
7 );
```

The bottom window shows the 'people' table structure in the 'mydatabase' schema. The table has the following columns:

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
person_id	int		NO			select,insert,update,references
person_name	varchar(10)		NO	utf8mb4	utf8mb4_genera...	select,insert,update,references
nickname	varchar(10)		NO	utf8mb4	utf8mb4_genera...	select,insert,update,references
age	tinyint unsigned		YES			select,insert,update,references
is_married	tinyint	0	YES			select,insert,update,references



- **person_id** 필드는 **AUTO_INCREMENT** 제약조건을 가지고 있기때문에 **INSERT** 할 때 입력하지 않아도 자동으로 입력됨을 주의하자
- **is_married** 필드는 **DEFAULT** 제약조건을 값을 **0**으로 지정하였기때문 **INSERT** 하지 않으면 자동으로 **0**이 입력됨을 주의하자.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • INSERT INTO people
2   (person_name, nickname, age)
3   VALUES ('김철수', '아이언워터', 10);
```

The left sidebar shows the database schema with 'mydatabase' selected, containing tables 'people', 'sakila', 'sys', and 'world'. The bottom panel shows the 'Table: people' structure with columns: person_id, person_name, nickname, age, and is_married.

The 'Result Grid' at the bottom displays the query results:

person_id	person_name	nickname	age	is_married
1	김철수	아이언워터	10	0
NULL	NULL	NULL	NULL	NULL



- **age** 필드는 **UNSIGNED** 제약조건을 가지고 있으므로 **INSERT**에서 **-2** 값을 입력하면 아래처럼 **error**가 발생할 수 있다.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following statement:

```
1 • INSERT INTO people
2   (person_name, nickname, age)
3   VALUES ('이불가', '임파서블', -2);
```

The value **-2** in the **VALUES** clause is highlighted with a red box. A red dotted arrow points from this box to the error message in the Output window.

The Output window shows the following error message:

#	Time	Action	Message	Duration / Fetch
2	13:58:40	SELECT * FROM mydatabase.people LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
3	14:01:31	SELECT * FROM mydatabase.people LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
4	14:06:48	INSERT INTO people (person_name, nickname, age) VALUES ('이불가', '임파서블', -2);	Error Code: 1264. Out of range value for column 'age' at row 1	0.000 sec

The error message is highlighted with a red box.

On the left, the Schemas pane shows the database structure. The 'mydatabase' schema is expanded, showing tables 'people', 'sakila', 'sys', and 'world'. The 'people' table is selected, and its columns are listed: **person_id** (int AI PK), **person_name** (varchar(10)), **nickname** (varchar(10)), **age** (tinyint UN), and **is_married** (tinyint).



- **nickname** 필드는 **UNIQUE** 와 **NOT NULL** 제약조건을 가지고 있기때문에 같은 이름의 ‘아이언워터’는 입력할 수 없음을 주의하자

The screenshot shows the MySQL Workbench interface. In the center, a SQL query is being executed in the 'people' table:

```
1 • INSERT INTO people
2   (person_name, nickname, age, is_married)
3   VALUES ('박싯물', '아이언워터', NULL, 1);
4
```

The value '아이언워터' is highlighted with a red box. A red dotted arrow points from this box to the error message in the 'Action Output' pane at the bottom.

On the left, the 'SCHEMAS' pane shows the 'mydatabase' schema with a table named 'people'. The 'Table: people' section lists the columns: person_id (int AI PK), person_name (varchar(10)), nickname (varchar(10)), age (tinyint UN), and is_married (tinyint).

The 'Action Output' pane at the bottom shows the results of the SQL execution. The third row is highlighted with a red box, indicating an error:

#	Time	Action	Message	Duration / Fetch
4	14:06:48	INSERT INTO people (person_name, nickname, age) VALUES ('이불가', '임파서...	Error Code: 1264. Out of range value for column 'age' at row 1	0.000 sec
5	14:11:47	INSERT INTO people (person_name, nickname, age, is_married) VALUES ('박싯...	Error Code: 1062. Duplicate entry '아이언워터' for key 'people.nickname'	0.000 sec
6	14:14:05	INSERT INTO people (person_name, nickname, age, is_married) VALUES ('박싯...	Error Code: 1062. Duplicate entry '아이언워터' for key 'people.nickname'	0.000 sec



👉 여기서 잠깐 !!

greendb 스키마안에 <register> 테이블을 아래 조건대로 제약조건을 수정하시오.

<조건>

1. user_id는 primary key로 지정하시오.
2. user_pw는 not null, unique로 지정하시오.
3. user_name는 not null로 지정하시오.
4. reg_date는 Default 값으로 오늘날짜(current_date)함수를 지정하시오.

테이블 구조

Field	Type	Null	Key	Default	Extra
user_id	varchar(20)	NO	PRI	NULL	
user_pw	varchar(20)	NO	UNI	NULL	
user_name	varchar(10)	NO		NULL	
email	varchar(50)	YES		NULL	
reg_date	date	YES		curdate()	DEFAULT_GENERATED

6강

연습문제





1. CUSTOMER 테이블을 아래 명세표에 만족하도록 greendb 스키마 안에 작성하고, insert 문을 이용하여 다음 페이지의 데이터를 삽입하시오.

<customer 명세표>

한글컬럼명	Field	Type	Null	Key	Default	Extra
고객ID	customer_id	INT	NO	PRI		AUTO_INCREMENT
이름	user_name	VARCHAR(40)	NO			
이메일	email	VARCHAR(50)	YES		NULL	
전화번호	phone	VARCHAR(20)	YES		NULL	



<insert 데이터>

customer_id	user_name	email	phone
1	김철수	kimcs@example.com	010-1111-1111
2	박영희	parkYH@example.com	010-2222-2222
3	이민수	leems@example.com	010-3333-3333
4	최가람	choiGR@example.com	010-4444-4444
5	정하늘	jungHN@example.com	010-5555-5555
6	한별이	hanBE@example.com	010-6666-6666
7	오세훈	ohSH@example.com	010-7777-7777
8	문지혜	moonJH@example.com	010-8888-8888
9	강토리	kangTR@example.com	010-9999-9999
10	유소라	yuSR@example.com	010-1212-1212
11	윤하린	yoonHR@example.com	010-1313-1313
12	장우진	jangWJ@example.com	010-1414-1414
13	백수민	baekSM@example.com	010-1515-1515
14	홍서준	hongSJ@example.com	010-1616-1616
15	조예린	joYR@example.com	010-1717-1717



1. address 테이블을 아래 명세표에 만족하도록 greendb 스키마 안에 작성하고, insert 문을 이용하여 다음 페이지의 데이터를 삽입하시오.

<address 명세표>

한글컬럼명	Field	Type	Null	Key	Default	Extra
주소ID	address_id	INT	NO	PRI		AUTO_INCREMENT
고객ID	customer_id	INT	NO			
도시	city	VARCHAR(30)	NO			
구/군	district	VARCHAR(30)	NO			
상세주소	detail	VARCHAR(50)	YES		NULL	



<insert 데이터>

address_id	customer_id	city	district	detail
1	1	서울	강남구	역삼동 123-1
2	2	서울	서초구	서초동 88-3
3	3	서울	마포구	합정동 33-10
4	4	부산	해운대구	좌동 12-2
5	5	대구	중구	동성로 55
6	6	인천	남동구	구월동 77-8
7	7	광주	북구	신안동 19-3
8	8	대전	유성구	궁동 45-9
9	9	울산	남구	삼산동 90-4
10	10	수원	영통구	원천동 24
11	11	성남	분당구	정자동 300
12	12	고양	일산동구	마두동 22-1
13	13	용인	수지구	죽전동 101
14	14	창원	성산구	상남동 77
15	15	전주	완산구	서신동 18-5



1. category 테이블을 아래 명세표에 만족하도록 greendb 스키마 안에 작성하고, insert 문을 이용하여 다음 페이지의 데이터를 삽입하시오.

<category 명세표>

한글컬럼명	Field	Type	Null	Key	Default	Extra
카테고리ID	category_id	INT	NO	PRI		AUTO_INCREMENT
카테고리명	category_name	VARCHAR(30)	NO	UNI		

<insert 데이터>

category_id	category_name
6	간식
8	공산품
1	과일
15	기타
12	문구류
14	반려동물용품
7	생활용품
4	수산물
13	스포츠용품
3	육류
5	음료
10	의류
9	전자제품
11	주방용품
2	채소



1. product 테이블을 아래 명세표에 만족하도록 greendb 스키마 안에 작성하고, insert 문을 이용하여 다음 페이지의 데이터를 삽입하시오.

<product 명세표>

한글컬럼명	Field	Type	Null	Key	Default	Extra
상품ID	product_id	INT	NO	PRI		AUTO_INCREMENT
상품명	product_name	VARCHAR(50)	NO			
가격	price	INT	NO			
재고	stock	INT	YES		0	
카테고리ID	category_id	INT	NO			



<insert 데이터>

product_id	product_name	price	stock	category_id
1	사과 1kg	4500	120	1
2	배 1kg	5500	90	1
3	당근 1kg	2500	80	2
4	브로콜리 1송이	1800	70	2
5	돼지고기 삼겹살 1kg	14500	60	3
6	소고기 등심 500g	22000	40	3
7	고등어 1마리	3000	50	4
8	오징어 1마리	2500	70	4
9	콜라 1.5L	2100	100	5
10	사이다 1.5L	2000	100	5
11	초코파이 12개입	4500	150	6
12	감자칩 오리지널	2500	130	6
13	주방세제 1L	3500	90	7
14	휴지 30롤	14000	40	7
15	볼펜 3개세트	1500	200	12



1. orders 테이블을 아래 명세표에 만족하도록 greendb 스키마 안에 작성하고, insert 문을 이용하여 다음 페이지의 데이터를 삽입하시오.

< orders 명세표 >

한글컬럼명	Field	Type	Null	Key	Default	Extra
주문ID	order_id	INT	NO	PRI		AUTO_INCREMENT
고객ID	customer_id	INT	NO			
주문일	order_date	DATE	NO			
주문상태	status	VARCHAR(20)	YES		주문완료	



<insert 데이터>

order_id	customer_id	order_date	status
1	1	2024-01-01	주문완료
2	2	2024-01-02	배송중
3	3	2024-01-03	주문완료
4	4	2024-01-04	배송완료
5	5	2024-01-05	주문완료
6	6	2024-01-06	배송중
7	7	2024-01-07	주문완료
8	8	2024-01-08	배송완료
9	9	2024-01-09	주문완료
10	10	2024-01-10	배송중
11	11	2024-01-11	주문완료
12	12	2024-01-12	배송완료
13	13	2024-01-13	주문완료
14	14	2024-01-14	주문완료
15	15	2024-01-15	배송중



1. order_item 테이블을 아래 명세표에 만족하도록 greendb 스키마 안에 작성하고, insert 문을 이용하여 다음 페이지의 데이터를 삽입하시오.

< order_item 명세표 >

한글컬럼명	Field	Type	Null	Key	Default	Extra
주문상세ID	item_id	INT	NO	PRI		AUTO_INCREMENT
주문ID	order_id	INT	NO			
상품ID	product_id	INT	NO			
수량	quantity	INT	YES		1	



<insert 데이터>

item_id	order_id	product_id	quantity
1	1	1	2
2	2	3	1
3	3	5	3
4	4	2	1
5	5	4	2
6	6	6	1
7	7	8	4
8	8	7	2
9	9	9	1
10	10	10	3
11	11	11	2
12	12	12	5
13	13	13	1
14	14	14	2
15	15	15	1



1. shipment 테이블을 아래 명세표에 만족하도록 greendb 스키마 안에 작성하고, insert 문을 이용하여 다음 페이지의 데이터를 삽입하시오.

< shipment 명세표 >

한글컬럼명	Field	Type	Null	Key	Default	Extra
배송ID	shipment_id	INT	NO	PRI		AUTO_INCREMENT
주문ID	order_id	INT	NO			
운송장번호	tracking_number	VARCHAR(40)	YES	UNI	NULL	
배송일	shipped_date	DATE	YES		NULL	



<insert 데이터>

shipment_id	order_id	tracking_number	shipped_date
1	1	TRK-001	2024-01-02
2	2	TRK-002	2024-01-03
3	3	TRK-003	2024-01-04
4	4	TRK-004	2024-01-05
5	5	TRK-005	2024-01-06
6	6	TRK-006	2024-01-07
7	7	TRK-007	2024-01-08
8	8	TRK-008	2024-01-09
9	9	TRK-009	2024-01-10
10	10	TRK-010	2024-01-11
11	11	TRK-011	2024-01-12
12	12	TRK-012	2024-01-13
13	13	TRK-013	2024-01-14
14	14	TRK-014	2024-01-15
15	15	TRK-015	2024-01-16