



Database

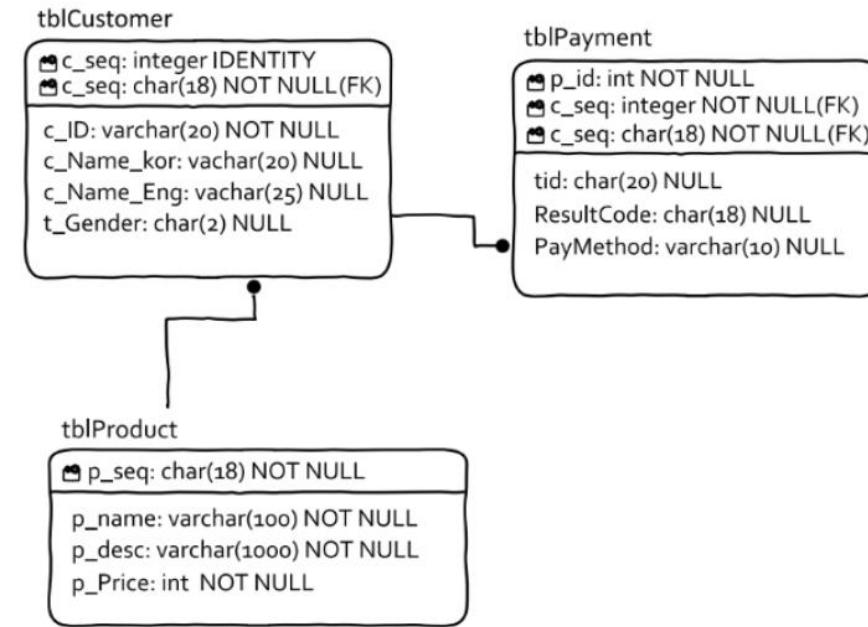
# 2장. 실전용 SQL 미리 맛보기

동아대학교 컴퓨터공학과  
양 선

# 02-1 건물을 짓기 위한 설계도: 데이터베이스 모델링 (교재 p.56)

## ◦ Database Modeling

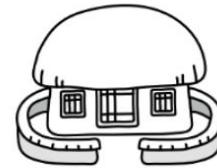
- 테이블의 구조를 미리 설계하는 개념
- 고객의 요구에 따라 특정 정보시스템의 데이터베이스를 설계
- (건축 설계도를 그리는 과정과 비슷)



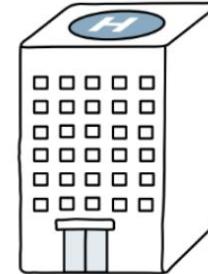
데이터베이스 모델링의 결과로  
테이블의 구조가 결정됩니다.

# 프로젝트 진행 단계

- 예를 들어 건축에서 높은 건물을 지으려면
  - 정확한 계획과 분석, 설계도 작업을 마친 후에 실제로 건물을 지어야 합니다.



한두 명이 작업 가능함



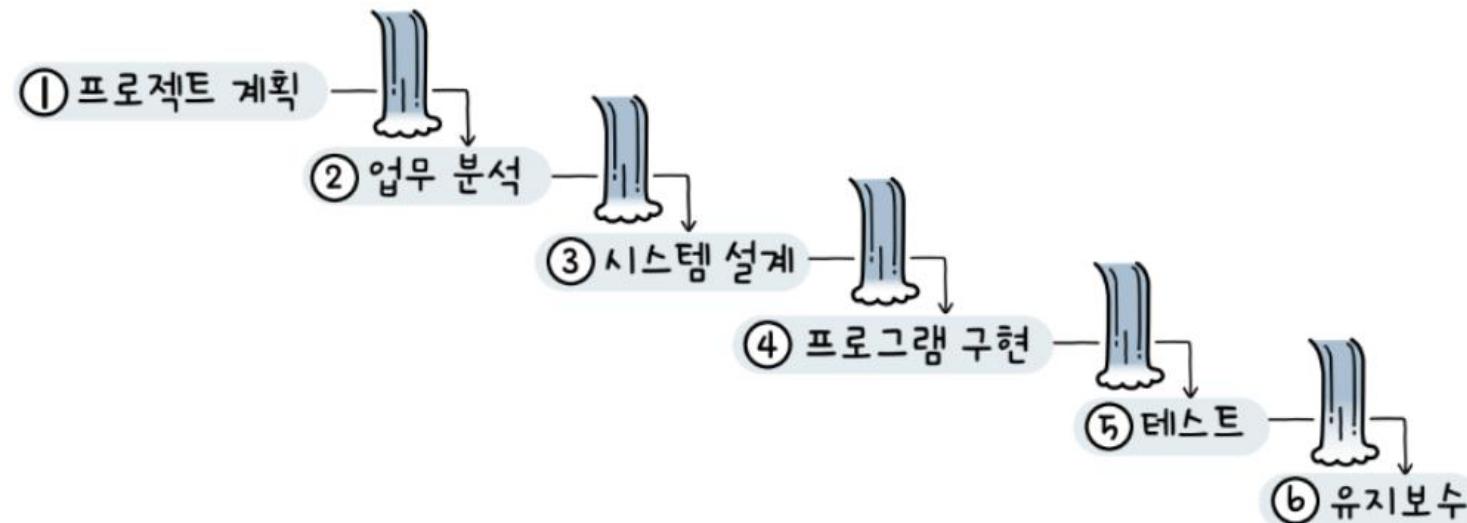
한두 명으로는 작업 불가능함

- 소프트웨어 프로젝트도 마찬가지로
  - 정확한 계획과 분석, 설계 작업을 마친 후에 구현을 시작해야 합니다.
    - 이런 절차를 연구하는 학문 분야를 '소프트웨어 공학'이라고 합니다.

# 프로젝트 진행 단계

- ▣ 폭포수 모델 waterfall model

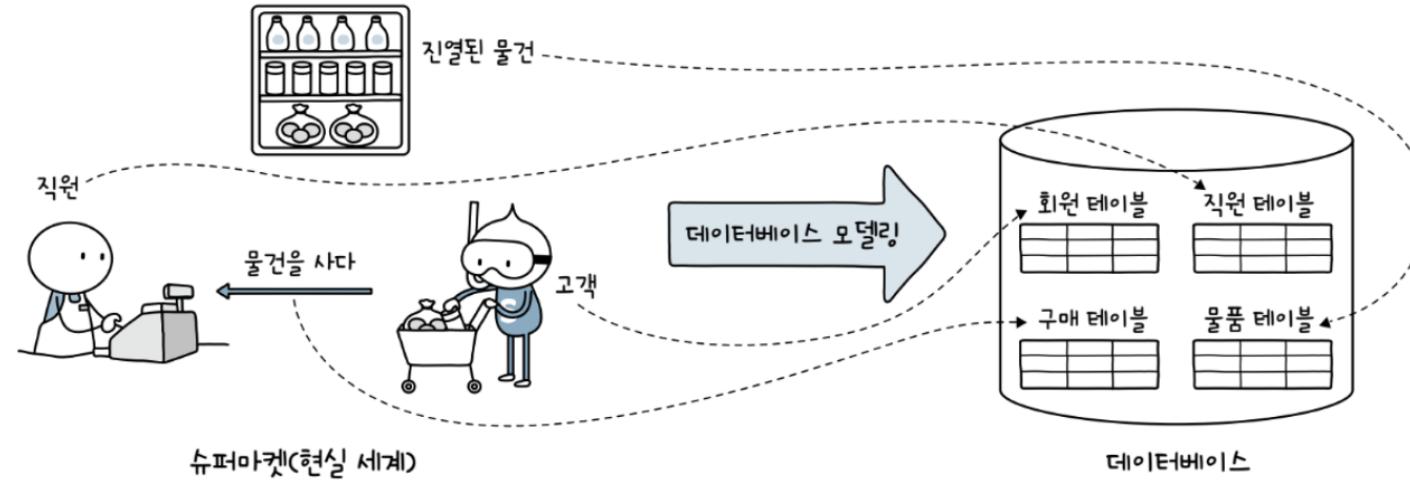
- 소프트웨어 공학에서 가장 기본적으로 언급되는 소프트웨어 개발 절차 중 하나



- ▣ 데이터베이스 모델링은 폭포수 모델에서 업무 분석과 시스템 설계 단계에 해당합니다.

# 데이터베이스 모델링

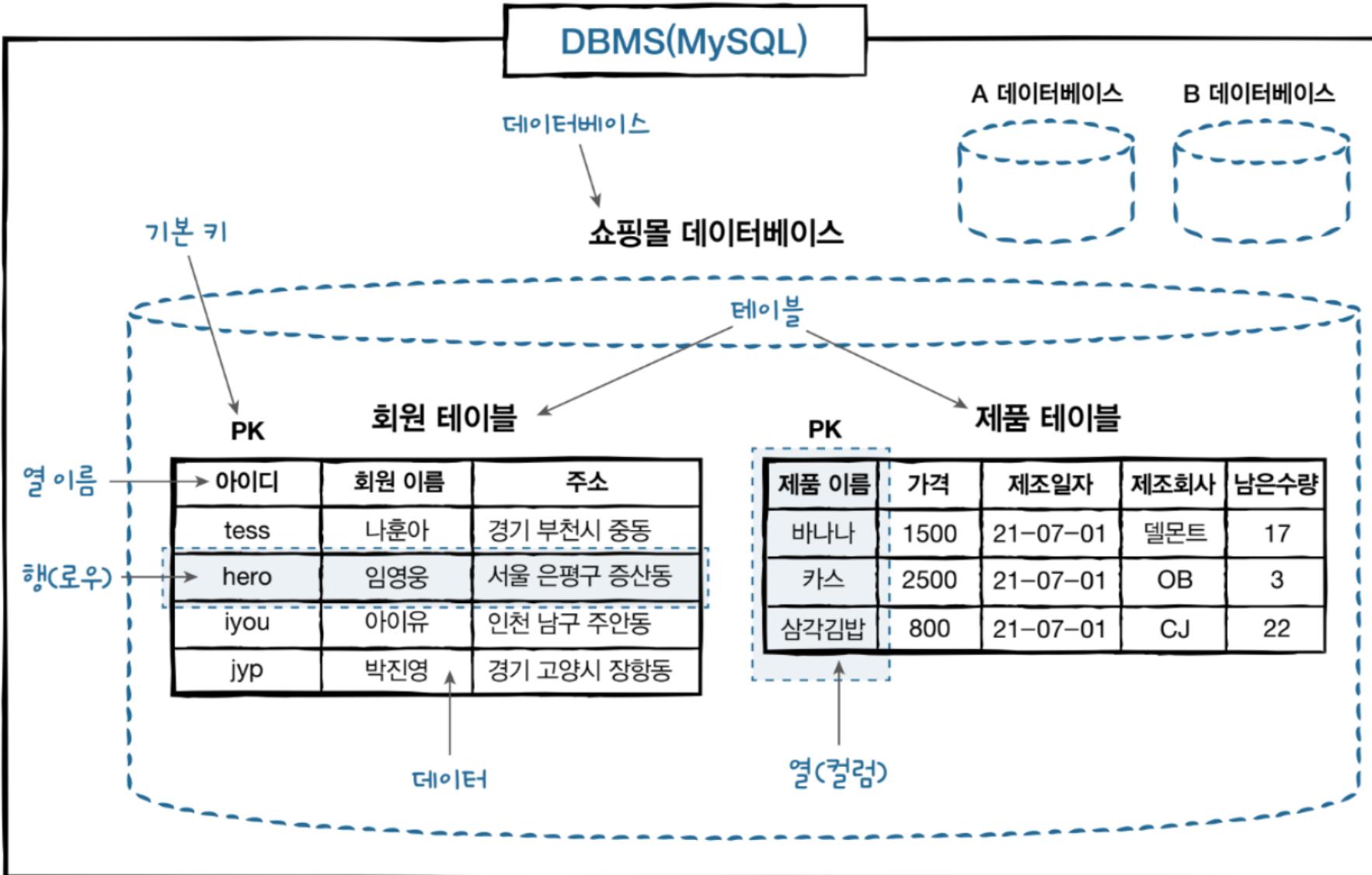
- 예를 들어 인터넷 쇼핑몰을 구현한다면
  - 현실 세계 파악부터 시작(업무 분석)한 후 설계(테이블 결정)
  - 사람, 물건 뿐만 아니라 행동도 테이블로 변환할 수 있습니다.
    - '고객이 특정 제품을 구매했다' (판매 내역), '회사가 제품을 들여왔다' (입고 내역), '고객과 상담했다' (상담 내역), '직원에게 월급을 지급했다' (인건비 지급 내역) 등



- 데이터베이스 모델링에는 정답이 없습니다.
  - 건물 설계도를 그리는 사람에 따라 다양한 결과물이 나오는 것과 같습니다.
  - 하지만 좋은 모델링과 나쁜 모델링은 분명히 존대 (다양한 학습과 실무 경험에서 우러나옴)

# 전체 데이터베이스 구성도

- 앞에서 살펴본 데이터베이스 모델링의 결과로 다음과 같은 구성이 완료되었다고 가정



## [중간 정리]

### ▶ 4가지 키워드로 끝내는 핵심 포인트

- **프로젝트**란 현실 세계를 컴퓨터 시스템으로 옮겨놓는 일련의 과정입니다.
- **폭포수 모델**은 소프트웨어 개발 단계 중 하나로, 이름 그대로 폭포가 떨어지듯 개발 단계가 진행됩니다.
- **데이터베이스 모델링**이란 현실 세계에서 사용되는 작업이나 사물들을 DBMS의 **테이블**(표 형태로 표현한 데이터베이스 개체)로 옮기기 위한 과정입니다.

# [중간 정리]

## ▶ 표로 정리하는 핵심 포인트

### 관련 중요 용어

한글 용어	영문 용어	약자(선택)	설명
데이터	data		단편적인 정보
테이블	table		데이터를 입력하기 위한 표 형태
데이터베이스	Database	DB	데이터의 저장소
데이터베이스 관리 시스템	Database Management System	DBMS	데이터베이스를 관리하는 시스템 또는 소프트웨어 (MySQL)
열(컬럼, 필드)	column, field		테이블의 세로. 테이블은 여러 개의 열로 구성됨
열 이름	column name, field name		각 열을 구분하기 위한 이름
데이터 형식	data type		열에 저장될 데이터의 형식(숫자/문자/날짜 등)
행(로우, 레코드)	row, record		테이블의 가로. 실질적인 진짜 데이터(행 데이터라고도 부름)
기본 키	Primary Key	PK	각 행을 구분하는 유일한 열
구조화된 질의 언어	Structured Query Language	SQL	사람과 DBMS가 소통하기 위한 언어

## [중간 확인문제]

1. 다음 각 설명이 의미하는 것을 관련 용어와 연결해보세요.

- ① 현실 세계를 컴퓨터 시스템으로 옮겨놓는 • 폭포수 모델  
일련의 과정을 일컫습니다.
- ② 소프트웨어 개발 절차 중 하나로 폭포가 • 프로젝트  
떨어지듯 각 단계가 진행됩니다.
- ③ 소프트웨어를 완성하는 절차를 연구하는 • 소프트웨어 공학  
분야를 통틀어서 이렇게 부릅니다.

2. 다음은 폭포수 모델의 절차입니다. 차례대로 나열해보세요.

시스템 설계, 테스트, 프로그램 구현, 프로젝트 계획, 업무 분석, 유지보수

## [중간 확인문제]

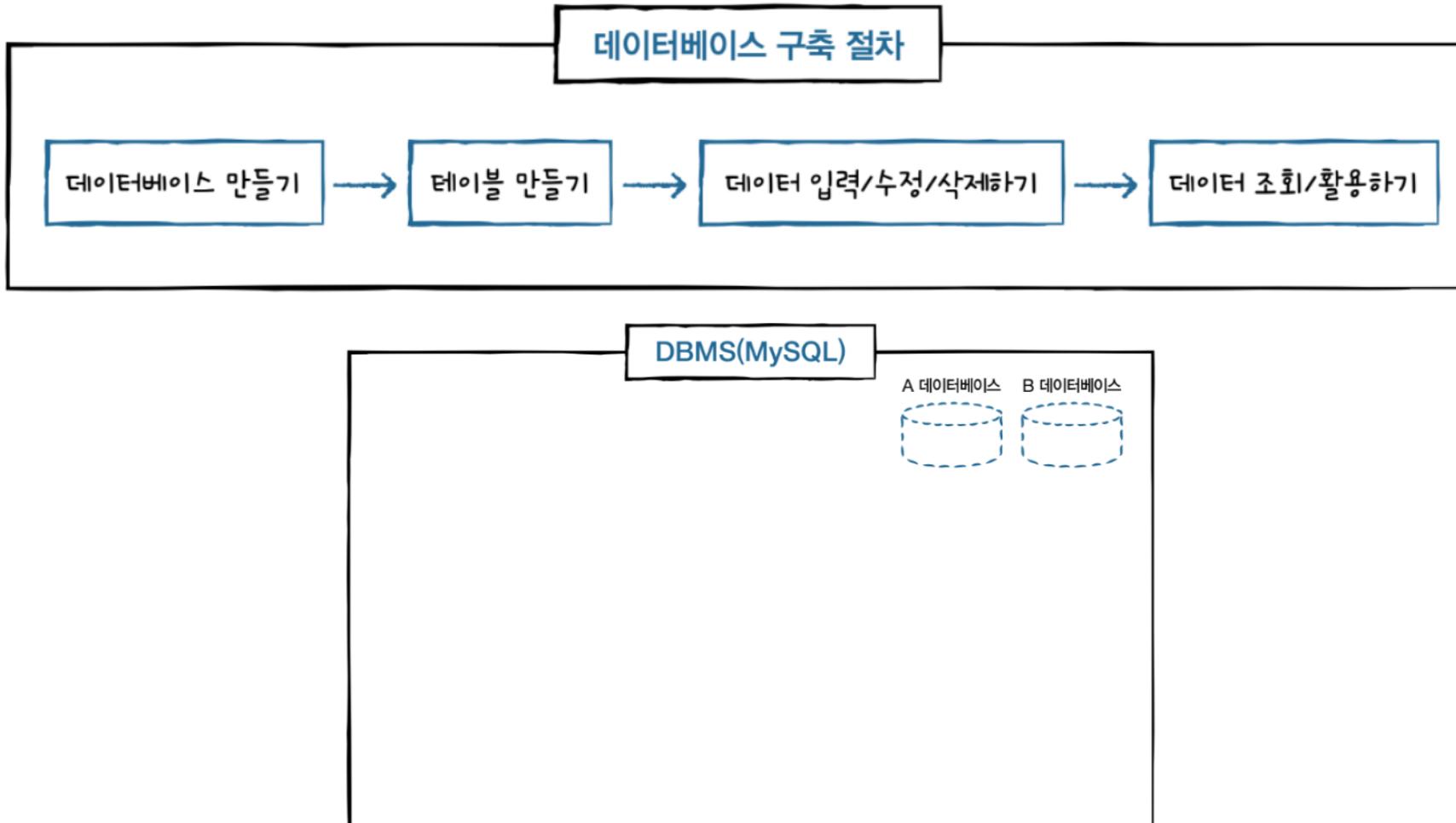
### 3. 다음은 무엇에 대한 설명인지 쓰세요.

우리가 살고 있는 세상에서 사용되는 작업이나 사물들을 DBMS의 데이터베이스 개체로 옮기기 위한 과정입니다. 또는 현실에서 쓰이는 것을 테이블로 변경하기 위한 작업입니다.

### 4. 다음 각 설명이 의미하는 것을 관련 용어와 연결해보세요.

- |   |        |
|---|--------|
| ① 회원이나 제품의 데이터를 입력하기 위해<br>표 형태로 표현한 것을 말합니다. 가로와<br>세로로 구성되어 있습니다. | • DBMS |
| ② 데이터베이스를 관리하는 시스템 또는 소<br>프트웨어를 말합니다.                              | • 행    |
| ③ 실질적인 진짜 데이터를 말합니다. 테이<br>블의 가로에 해당합니다.                            | • 테이블  |
| ④ 사람과 DBMS가 소통하기 위한 말(언어)<br>입니다.                                   | • SQL  |

## 02-2 데이터베이스 시작부터 끝까지 (교재 p.67)

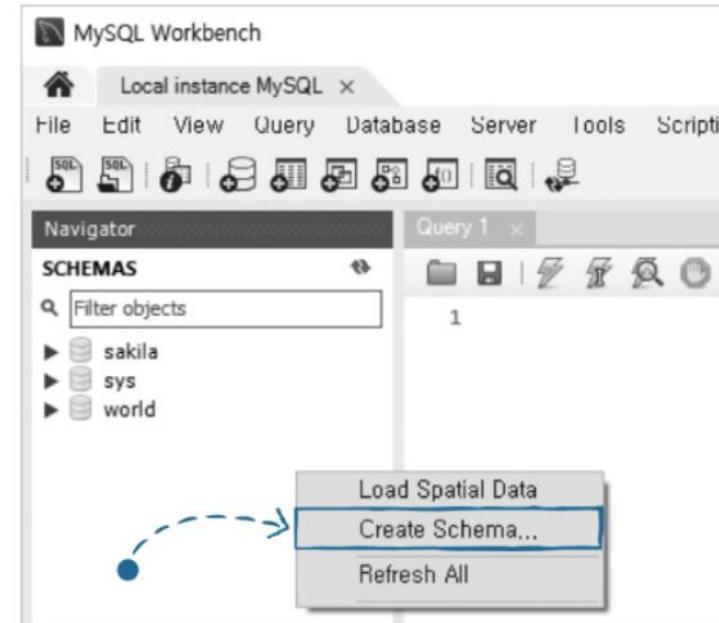


- 현재, 기본 데이터베이스들만 있습니다. 이제 쇼핑몰 데이터베이스를 만들어보겠습니다.

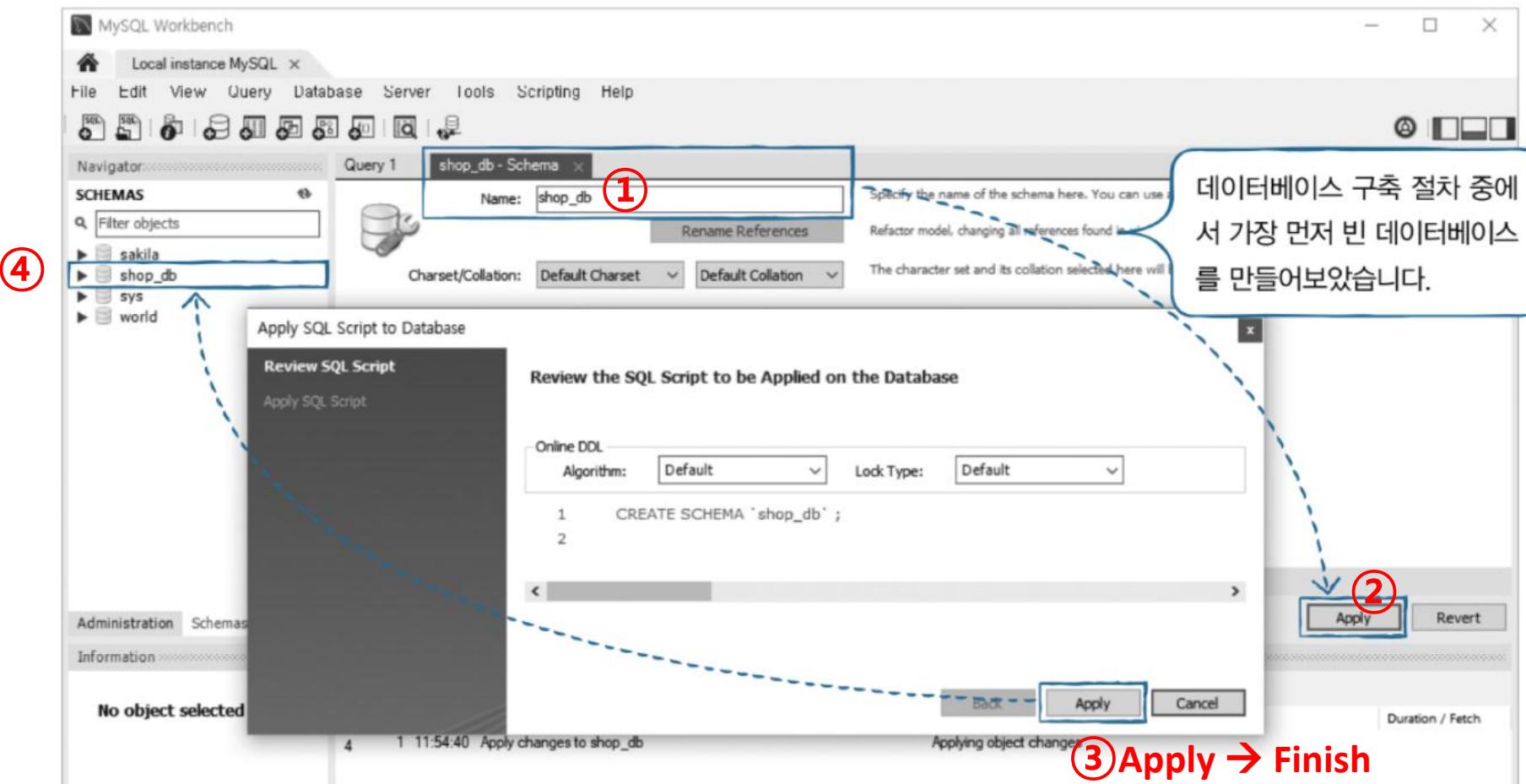
# Workbench에서 데이터베이스 생성

- MySQL에서는 스키마(schema)와 데이터베이스를 동일한 용어로 사용합니다.

[SCHEMAS] 패널의 빈 부분에서 마우스 오른쪽 버튼을 클릭한 후 [Create Schema]를 선택합니다.



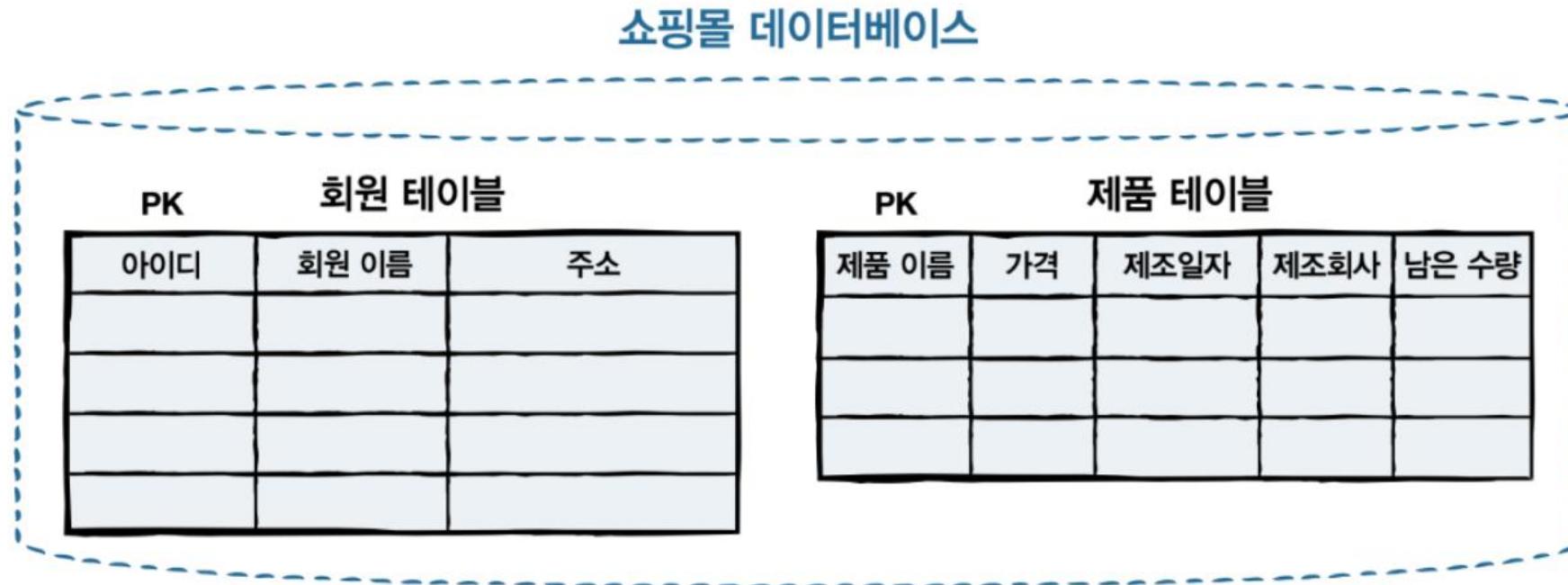
# Workbench에서 데이터베이스 생성



- MySQL에서 create database 와 create schema는 같은 의미

# 생성된 쇼핑몰 데이터베이스(shop\_db)에 테이블 생성하기

- 원래 계획했던 것처럼 새 데이터베이스에 테이블 2개를 만들어보겠습니다. (수업자료 p. 6 참고)



note

그림을 간단하게 표현하기 위해서 쇼핑몰 데이터베이스 외에 나머지는 생략했고, 무한대의 빈 줄을 3~4개로 표현했습니다.

# 테이블 설계

## 회원 테이블

열 이름(한글)	영문 이름	데이터 형식	최대 길이	널 허용 안 함(Not Null)
아이디(기본 키)	member_id	문자(CHAR)	8글자	Yes
회원 이름	member_name	문자(CHAR)	5글자	Yes
주소	member_addr	문자(CHAR)	20글자	No

## 제품 테이블

열 이름(한글)	영문 이름	데이터 형식	문자의 최대 길이	널 허용 안 함(Not Null)
제품 이름(기본 키)	product_name	문자(CHAR)	4글자	Yes
가격	cost	숫자(INT)		Yes
제조일자	make_date	날짜(DATE)		No
제조회사	company	문자(CHAR)	5글자	No
남은 수량	amount	숫자(INT)		Yes

# 테이블 생성

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 3 member - Table ×

SCHEMAS: shop\_db

Table Name: member Schema: shop\_db Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
member_id	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
member_name	CHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
member_addr	VARCHAR(45)	<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: membercol Data Type: VARCHAR(45) Charset/Collation: Default Charset Default Collation Expression: Storage: Virtual Stored Primary Key Not Null Unique Binary Unsigned Zero Fill Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Object Info Session

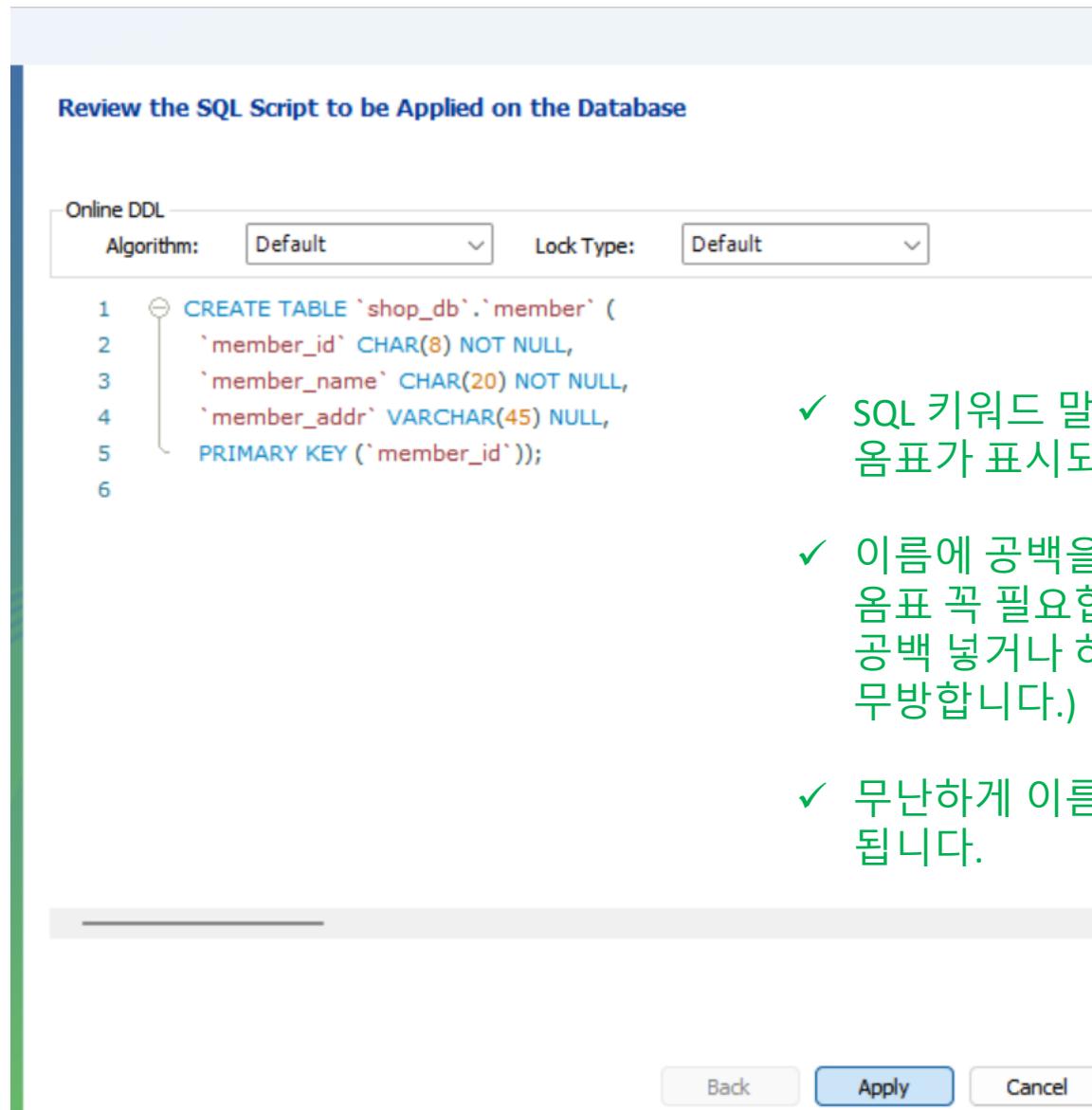
Apply Revert

혹시 화면 크기 부족해서 컬럼 입력 부분이 안보이면 클릭

고객명 길이 충분히 확보하고 주소는 길이가 다양하니 varchar (담당자가 판단)

데이터베이스 2장 16

# select문 확인 팝업창



# 방금 작업을 command line에서 한다면

- 방금 Workbench에서 테이블을 drop 한 후 command line에서 다음과 같이 작업하세요.

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mydb |  
| mysql |  
| performance_schema |  
| sakila |  
| shop_db |  
| sys |  
| world |  
+-----+  
8 rows in set (0.00 sec)
```

```
mysql> use shop_db  
Database changed  
mysql> create table member (  
-> memebr_id char(8) not null primary key,  
-> member_name char(20) not null,  
-> member_addr varchar(45)  
-> );  
Query OK, 0 rows affected (0.05 sec)
```

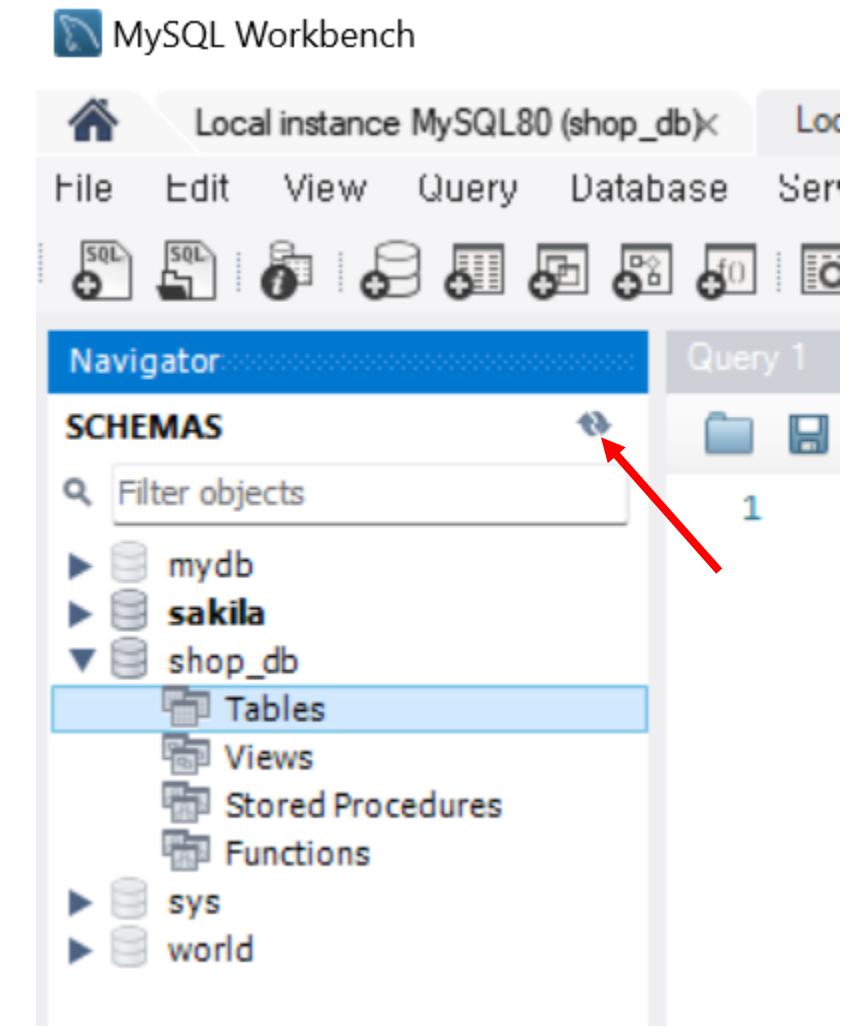
```
mysql>
```

- 회사 서버에서 직접 작업할 때는 command line에서 바로 입력하는 경우도 많이 때문에, 이 방법도 알아두세요.
- 비번 넣고 들어간 후 좌측처럼 작업하시면 됩니다.
- use shop\_db 했기 때문에 create table 할 때 shop\_db.member 아니고 그냥 member해도 됩니다. PK는 옆에 적어도 됩니다.
- create문은 아래와 같습니다. (복사 붙이기 하시면 됩니다)

```
create table member (  
memebr_id char(8) not null primary key,  
member_name char(20) not null,  
member_addr varchar(45)  
);
```

# Workbench, command line을 혼합 사용할 때

- command line에서 작업한 내용을 Workbench에서 바로 확인하려면 새로 고침 하시면 됩니다.
  - 반대로, Workbench에서 작업한 내용은 command line에서 바로 보임



# 데이터 입력하기

## 쇼핑몰 데이터베이스

PK 회원 테이블

아이디	회원 이름	주소
tess	나훈아	경기 부천시 중동
hero	임영웅	서울 은평구 증산동
iyou	아이유	인천 남구 주안동
jyp	박진영	경기 고양시 장항동

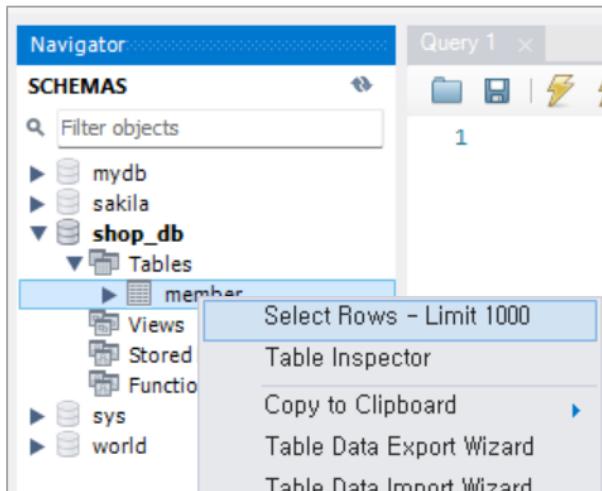
PK 제품 테이블

제품 이름	가격	제조일자	제조회사	남은 수량
바나나	1500	2021-07-01	델몬트	17
카스	2500	2022-03-01	OB	3
삼각김밥	800	2023-09-01	CJ	22



지금은 실습이라서 회원이 4명뿐이라고 가정하고 4건의 데이터만 입력합니다. 실무에서는 회원이 수십만, 수백만 명이 될 수도 있습니다.

# 테이블 안 데이터 보기



The screenshot shows the MySQL Workbench Query Editor. The 'Query 1' tab has the identifier 'member'. The SQL query `SELECT * FROM shop_db.member;` is entered in the query editor. The 'Result Grid' shows a single row with three columns: 'memebr\_id', 'member\_name', and 'member\_addr', all of which are NULL. The 'Action Output' pane shows the execution log: '# 1 00:50:23 SELECT \* FROM shop\_db.member LIMIT 0, 1000'. The 'Message' pane at the bottom right displays the text '0 row(s) returned'.

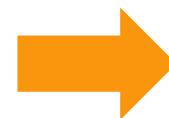
- ✓ 아직 데이터 입력 전이므로 아무것도 보이지 않음
- ✓ shop\_db가 진한글씨로 보인다는 의미는 use shop\_db 한 거나 마찬가지
- ✓ select \* from member limit 3; 이렇게 하면 3건 이하만 보겠다는 의미
- ✓ 우측 그림의 중간 null부분 클릭해서 데이터 입력 가능

# insert문 확인 팝업창

Result Grid | Filter Rows: Edit:

	member_id	member_name	member_addr
tess	나훈아	경기 부천시 중동	
hero	임영웅	서울 은평구 증산동	
iyou	아이유	인천 남구 주안동	
jyp	박진영	경기 고양시 장항동	
•	NULL	NULL	NULL

Apply



Review the SQL Script to be Applied on the Database

```
1 INSERT INTO `shop_db`.`member` (`member_id`, `member_name`, `member_addr`)
2 INSERT INTO `shop_db`.`member` (`member_id`, `member_name`, `member_addr`)
3 INSERT INTO `shop_db`.`member` (`member_id`, `member_name`, `member_addr`)
4 INSERT INTO `shop_db`.`member` (`member_id`, `member_name`, `member_addr`)
5
```

Result Grid | Filter Rows: | Edit:

	member_id	member_name	member_addr
hero	임영웅	서울 은평구 증산동	
iyou	아이유	인천 남구 주안동	
jyp	박진영	경기 고양시 장항동	
tess	나훈아	경기 부천시 중동	
•	NULL	NULL	NULL



Back Apply Cancel

- ✓ 순서 지정 (order by) 안 하면 기본적으로 PK 순으로 정렬되어서 보임

## 데이터 입력 및 수정

- 제품 테이블에도 데이터 입력

Result Grid		Filter Rows:				Edit:
	product_name	cost	make_date	company	amount	
	바나나	1500	2021-07-01	웰몬트	17	
	삼각김밥	800	2023-09-01	CJ	22	
▶	카스	2500	2022-03-01	OB	3	
*	HULL	HULL	HULL	HULL	HULL	

- 회원 테이블 및 제품 테이블에 추가 / 삭제 / 변경 자유롭게 해보세요.
    - Result창에서 수정 및 삭제 후 Apply 버튼 클릭

Result Grid | Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

	member_id	member_name	member_addr
..	carry	머라이어	영국 런던 맥자골목
	hero	임명웅	서울 은평구 증산동
	iyou	아이유	인천 남구 주안동
	jyp	박진영	경기 고양시 장항동
	tess	나훈아	경기 부천시 줄동
*	NUL	NUL	NUL

member 1

## command line 창에서 작업하는 경우

# use shop\_db 한 상태가 아니라면 테이블명 앞에 shop\_db. 붙이는 거 잊지마세요.

select \* from member where member\_id='iyou';

select member\_id, memner\_name from member;

insert into member values('gogh','빈센트 반 고흐','네덜란드 노르트브라반트 주 줄더르트');

update member

set member\_addr = '프랑스 센에우아즈 주 오베르쉬르와즈'

where member\_id='gogh';

delete from member where member\_id='gogh';

# 파일에 SQL문을 여러 개 작성했는데 그 중 일부만 실행시키려면

- ✓ 그냥 실행하면 결과 탭도 여러 개

The screenshot shows the MySQL Workbench interface. In the top editor tab titled 'SQL File 3\*', two SQL queries are listed:

```
1 • SELECT member_name, member_addr FROM member;
2 • SELECT * FROM member WHERE member_name = '아이유';
```

Below the editor, there are two separate Result Grid tabs:

- Result Grid 1:** Shows the result of the first query: member\_id, member\_name, and member\_addr for all members.
- Result Grid 2:** Shows the result of the second query: member\_id, member\_name, and member\_addr for the member named '아이유'.

Dashed arrows indicate the flow from the editor to each result grid tab.

- ✓ 필요한 부분만 드래그 후 실행해도 됨

The screenshot shows the same MySQL Workbench interface as the first one, but with a different selection. The second SQL query in the editor is highlighted with a blue selection bar:

```
1 • SELECT member_name, member_addr FROM member;
2 • SELECT * FROM member WHERE member_name = '아이유';
```

Below the editor, there is a single Result Grid tab:

- Result Grid:** Shows the result of the second query: member\_id, member\_name, and member\_addr for the member named '아이유'.

Dashed arrows indicate the flow from the editor to the single result grid tab.

# SQL 예약어와 자동 완성 기능

+ 여기서 잠깐

## SQL 예약어와 자동 완성

쿼리 창에서 SQL을 입력하면 예약어는 자동으로 파란색으로 표시됩니다. 예제로 사용한 **SELECT, FROM**은 이미 SQL에서 약속된 예약어이므로 파란색으로 표시되는 것입니다.

또한 MySQL 워크벤치는 IntelliSense 또는 AutoComplete 기능을 기능을 제공하는데, 이는 글자의 일부만 입력해도 그와 관련되는 글자들이 미리 제시되는 것을 말합니다. 다음 그림과 같이 's'만 입력해도 'select' 등이 표 현됩니다. 이 상태에서 **→** 키와 **Tab** 키를 누르면 자동으로 완성됩니다. 잘 활용하면 입력도 빨라지고 오타도 많이 줄어드는 장점이 있습니다.

자동 완성 기능을 사용하려면 MySQL 워크벤치의 [Edit] – [Preferences] 메뉴에서 [SQL Editor] – [Query Editor]의 'Automatically Start Code Completion'이 체크되어 있어야 합니다. 기본은 체크되어 있습니다.



## [중간 정리]

### ▶ 4가지 키워드로 끝내는 핵심 포인트

- **스키마**는 MySQL 안의 데이터베이스를 말합니다. 데이터베이스와 동일한 용어라고 생각하면 됩니다.
- **데이터 형식**은 문자형(CHAR), 정수형(INT) 등과 같이 열에 저장될 데이터의 형식을 말합니다.
- **예약어**는 SELECT, FROM, WHERE와 같은 기존에 약속된 SQL입니다.
- **기본 키**는 열에 지정하며, 각 행을 구분하는 유일한 값입니다.

# [중간 정리]

한글 용어	영문 용어	약자	설명
스키마	Schema		데이터베이스와 동일한 용어
문자형	Character	CHAR	문자를 입력할 데이터 형식
정수형	Integer	INT	정수를 입력할 데이터 형식
널	Null		비어 있는 값
널 허용 안함	Not Null	NN	빈 값을 허용하지 않음
자동 정렬			기본 키로 설정한 열로 자동 정렬됨
입력	INSERT		데이터를 입력하는 SQL
수정	UPDATE		데이터를 수정하는 SQL
삭제	DELETE		데이터를 삭제하는 SQL
조회	SELECT		데이터를 조회하는 SQL
조건	WHERE		SELECT 문에서 특정 조건을 조회할 때 사용하는 구문
예약어			기존에 약속된 SQL. SELECT, FROM, WHERE 등
세미콜론(:)			SQL의 끝을 표시하는 기호
인텔리센스	IntelliSense		SQL의 글자가 미리 제시되는 워크벤치의 기능

## [중간 확인문제]

1. 다음은 데이터베이스 구축 절차입니다. 차례대로 나열해보세요.

테이블 만들기, 데이터 조회하기, 데이터 입력하기, 데이터베이스 만들기

2. 다음 중에서 빙칸에 들어갈 내용으로 알맞은 것을 고르세요.

데이터베이스는 테이블을 저장하는 공간으로 라고도 부릅니다.

- ① 테이블
- ② 열 이름
- ③ 데이터 형식
- ④ 스키마
- ⑤ 기본 키

## [중간 확인문제]

3. 데이터베이스는 MySQL 워크벤치 메뉴를 이용하거나, SQL 문을 통해 구축할 수 있습니다.  
다음 각 설명이 의미하는 것을 관련 용어와 연결해보세요.

- |                         |   |          |
|-------------------------|---|----------|
| ① 데이터를 수정할 때 사용         | • | • CREATE |
| ② 데이터를 조회할 때 사용         | • | • UPDATE |
| ③ 테이블이나 데이터베이스를 만들 때 사용 | • | • DELETE |
| ④ 데이터를 삭제할 때 사용         | • | • SELECT |

## [중간 확인문제]

4. 다음 각 설명이 의미하는 것을 관련 용어와 연결해보세요.

- |                          |   |            |
|--------------------------|---|------------|
| ① 데이터 형식 중에서 소수점이 없는 정수형 | • | • CHAR     |
| ② 비어 있는 값을 허용하지 않음       | • | • INT      |
| ③ 데이터 형식 중에서 문자형         | • | • DATE     |
| ④ 데이터 형식 중에서 날짜형         | • | • NOT NULL |

5. 다음 SELECT 문에 대한 설명으로 알맞은 것을 고르세요.

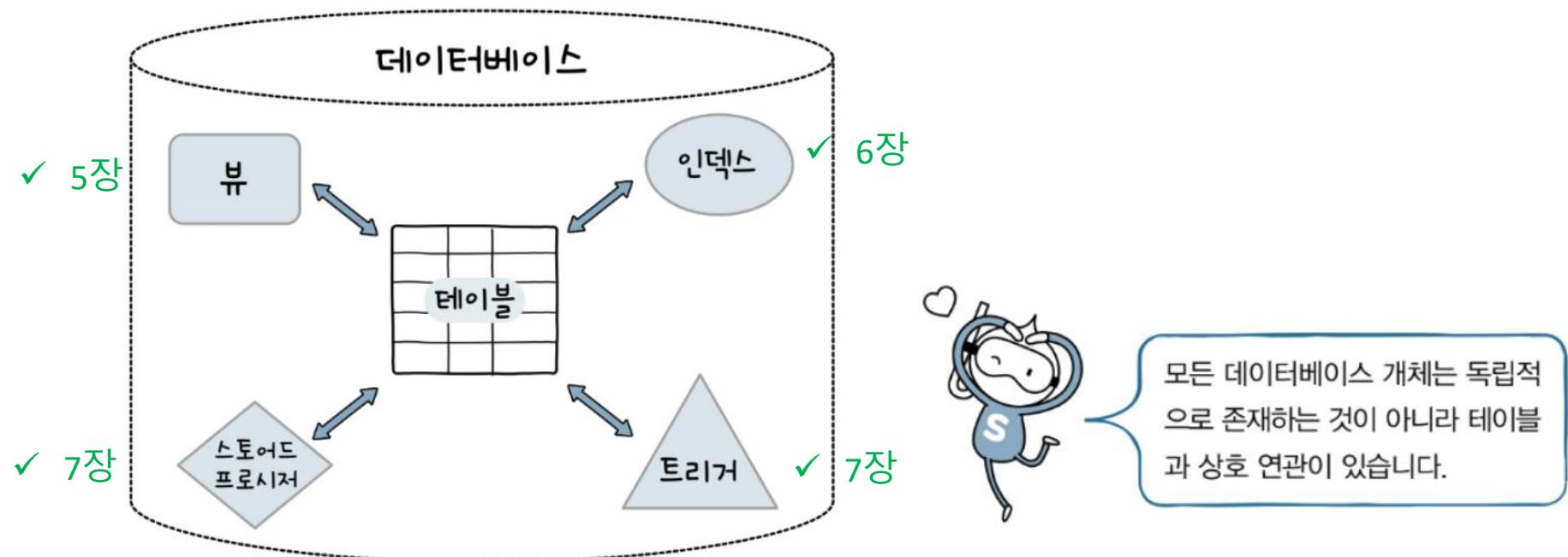
```
SELECT * FROM 테이블_이름 WHERE 열_이름 = '값' ;
```

- |                    |
|--------------------|
| ① 특정 값에 해당하는 행만 조회 |
| ② 특정 값에 해당하는 열만 조회 |
| ③ 모든 행의 모든 열을 조회   |
| ④ 모든 행의 특정 열만 조회   |

## 02-3 데이터베이스 개체 (교재 p.87)

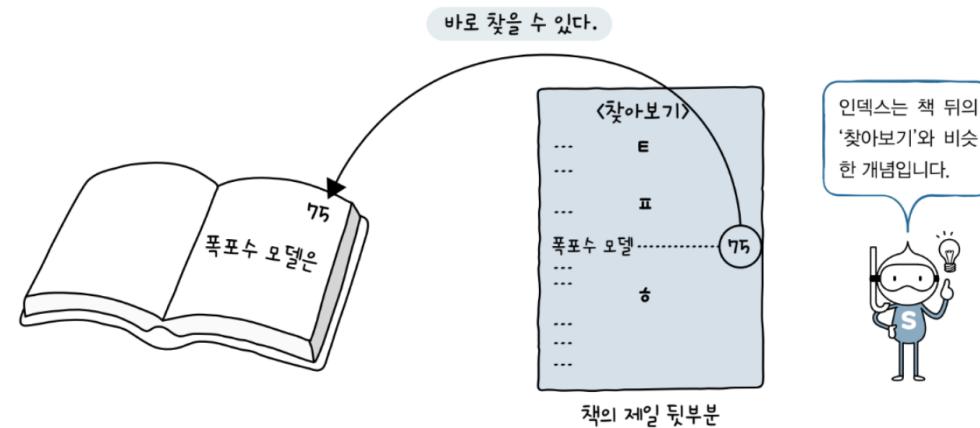
- 테이블을 데이터베이스의 핵심 개체

- 하지만 테이블 외에 인덱스, 뷰, 스토어드 프로시저, 트리거, 함수, 커서 등의 개체도 필요
- 인덱스는 조회 결과 빠르게, 뷰는 테이블 내용을 제한적으로 표현할 때,
- 스토어드 프로시저는 SQL에서 프로그래밍이 가능하게, 트리거는 잘못된 데이터 들어가는 것을 미연에 방지



# 인덱스 없는 경우

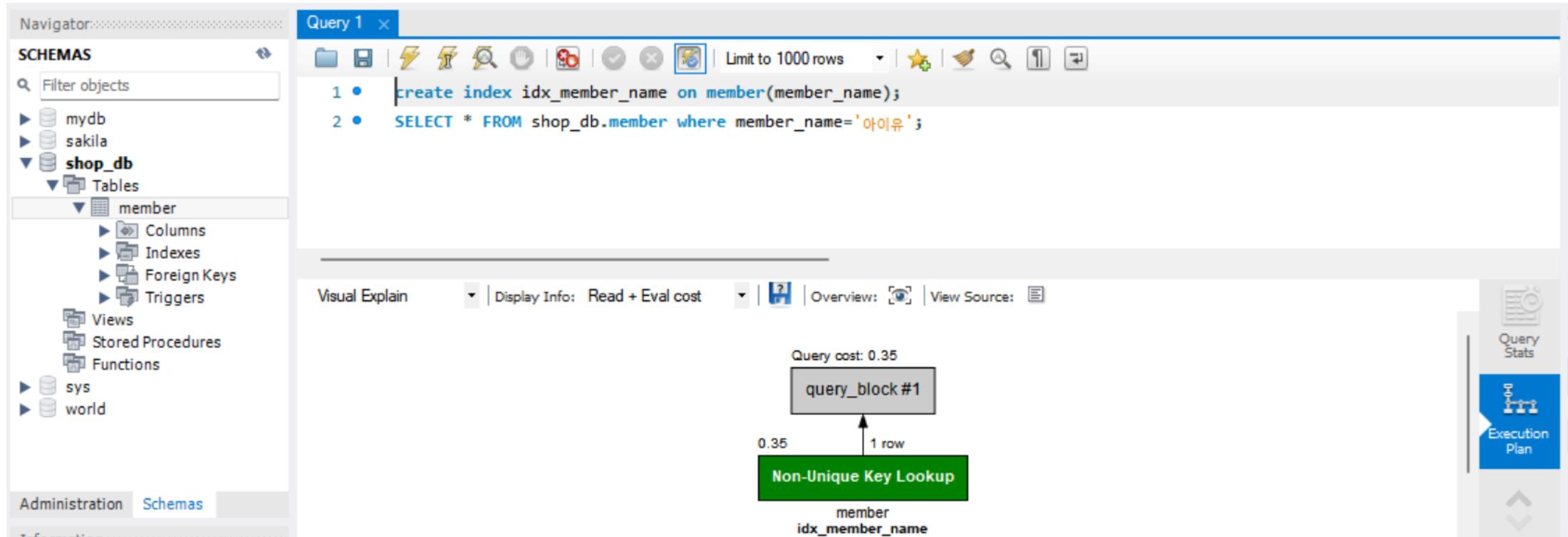
- 현재는 테이블에 데이터가 적어서 인덱스 유무에 상관없이 별로 속도 차이효과 못 느끼겠지만, 실제로 몇 억 건이 있더라도 지금 실습하는 것과 동일하게 적용됩니다.
- 회원 테이블 select \* 한 후 중간에 결과 나오는 [Result Grid] 말고 [Execution Plan] 선택해보면 Full Table Scan 한 걸 알 수 있습니다.



The screenshot shows the MySQL Workbench interface with the 'Execution Plan' tab active. The 'Visual Explain' section displays a single node labeled 'query\_block #1' with a cost of 0.35 for 1 row. The main query editor shows the SQL statement: 'SELECT \* FROM shop\_db.member where member\_name='아이유';'. The Navigator pane on the left lists databases and tables, with 'shop\_db.member' selected. The status bar at the bottom indicates 'Administration Schemas'.

# 인덱스 있는 경우

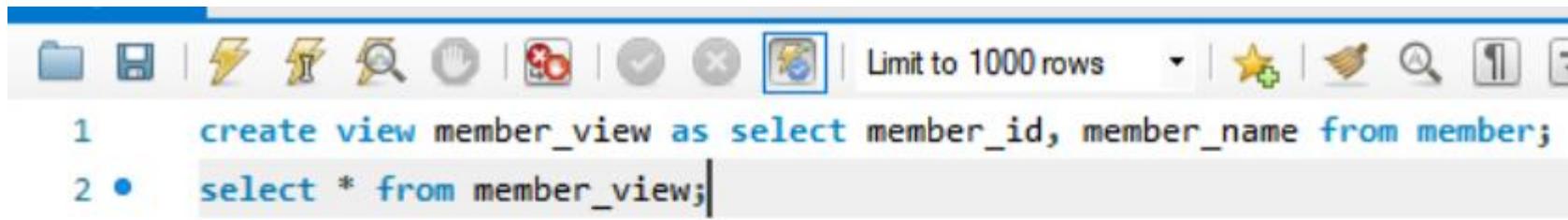
- 이번에는 create index idx\_member\_name on member(member\_name); 한 후에 select 해보세요.



- 회원 테이블에 수십억 건이 넘어도 인덱스를 통해 조회했다면 순식간에 결과가 나왔을 것입니다.
- 인덱스를 삭제하려면 drop index idx\_member\_name on member; 하시면 됩니다.

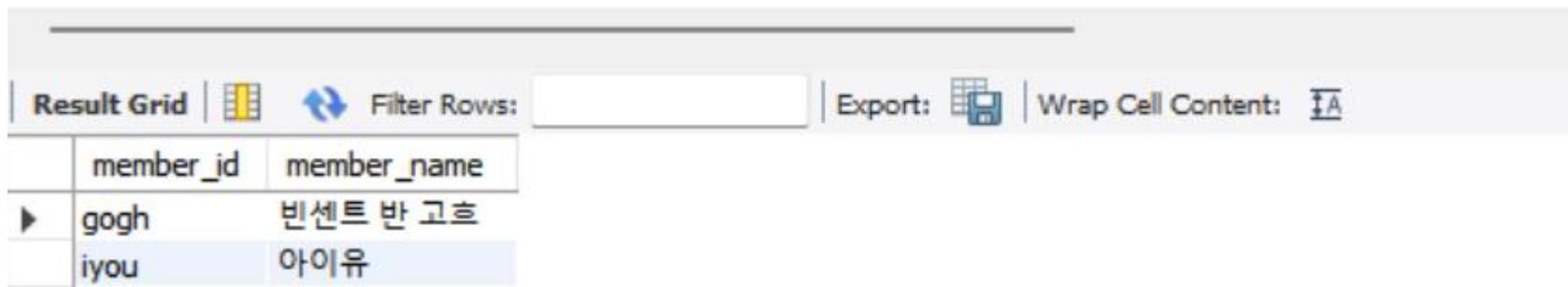
# 뷰

- 테이블의 일부만 보여주고 싶을 때
  - 예를 들어 회원 테이블에는 주소도 있지만, 주소를 뺀 다른 정보들만 보여주고 싶을 때
  - 주소를 뺀 뷰를 만들어서, 그 뷰에만 접근 권한을 주고 원래의 테이블은 접근 못하게 함



```
1 create view member_view as select member_id, member_name from member;
2 • select * from member_view;
```

- ✓ 뷰 사용은 보안에 도움이 됩니다.
- ✓ 그리고 긴 SQL문을 간략하게 만들 수도 있습니다.

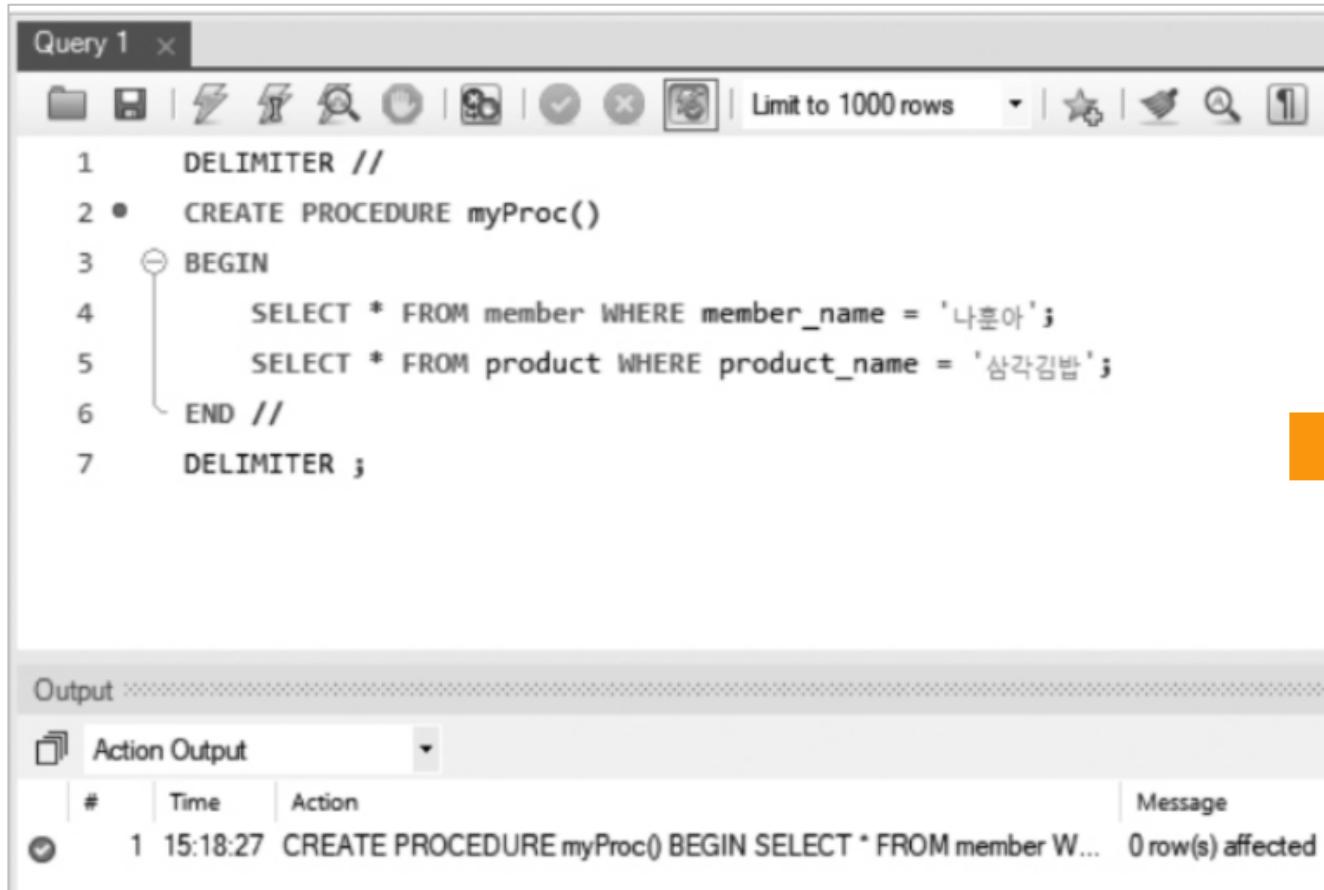


	member_id	member_name
▶	gogh	빈센트 반 고흐
	iyou	아이유

# 스토어드 프로시저

- SQL 안에서도 일반 프로그래밍 언어처럼 코딩할 수 있음

Query 1 ×



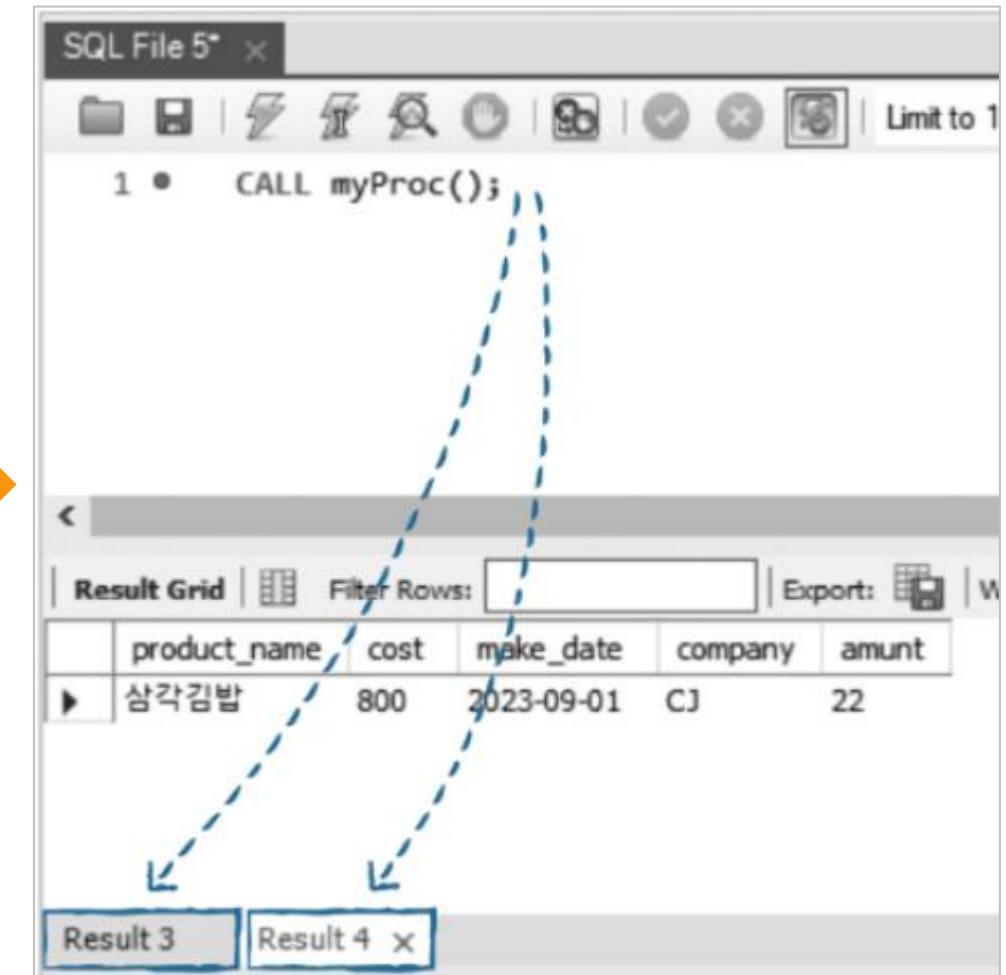
```
1  DELIMITER //
2 •  CREATE PROCEDURE myProc()
3  BEGIN
4      SELECT * FROM member WHERE member_name = '나훈아';
5      SELECT * FROM product WHERE product_name = '삼각김밥';
6  END //
7  DELIMITER ;
```

Output ::::::::::::

Action Output

#	Time	Action	Message
1	15:18:27	CREATE PROCEDURE myProc() BEGIN SELECT * FROM member W...	0 row(s) affected

SQL File 5\* ×



```
1 •  CALL myProc();
```

Result Grid | Filter Rows: | Export: | W

product_name	cost	make_date	company	amunt
삼각김밥	800	2023-09-01	CJ	22

Result 3      Result 4 X

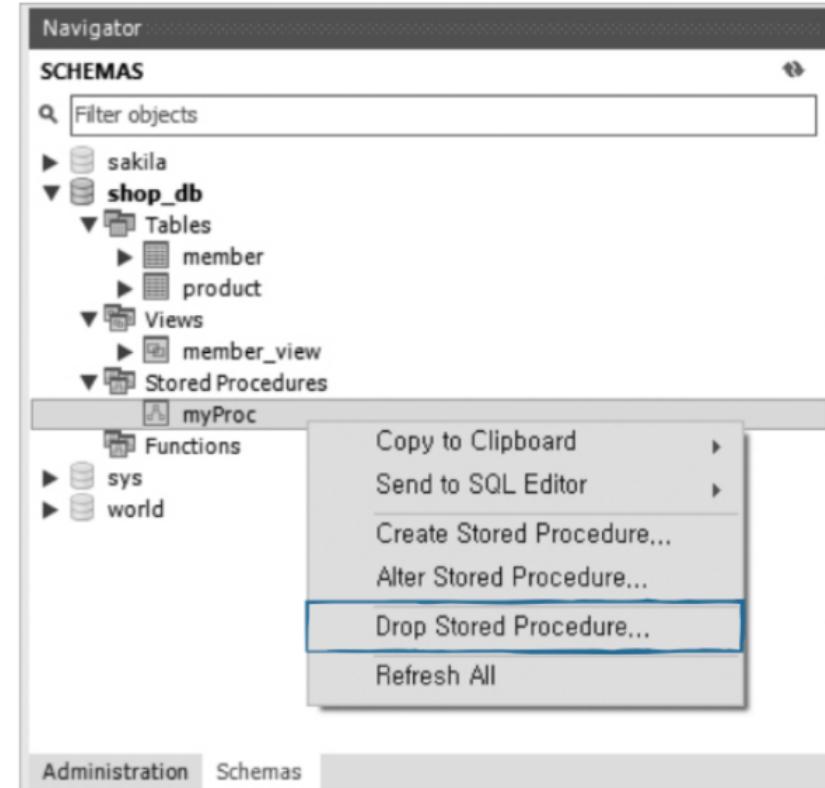
# 개체 생성 및 삭제

+ 여기서 잠깐

## CREATE 문과 DROP 문

테이블, 인덱스, 뷰, 스토어드 프로시저 등의 데이터베이스 개체를 만들기 위해서는 **CREATE 개체\_종류 개체\_이름 ~~~** 형식을 사용합니다. 반대로 데이터베이스 개체를 삭제하기 위해서는 **DROP 개체\_종류 개체\_이름** 형식을 사용합니다. 예로, 실습에서 생성한 스토어드 프로시저를 삭제하려면 **DROP PROCEDURE myProc**를 사용합니다.

MySQL 워크벤치에서 생성, 삭제하려면 먼저 [SCHEMAS] 패널의 빈 곳에서 마우스 오른쪽 버튼을 클릭하고 [Refresh All]을 선택해서 새로 고침합니다. 그리고 생성, 삭제할 데이터베이스 개체에서 마우스 오른쪽 버튼을 클릭하고 생성하려면 [Create 데이터베이스\_개체]를, 삭제하려면 [Drop 데이터베이스\_개체]를 선택합니다.



## [마무리 정리]

### ▶ 3가지 키워드로 끝내는 핵심 포인트

- **인덱스**는 책 뒤의 ‘찾아보기’와 비슷한 개념입니다. 데이터를 빠르게 찾을 수 있도록 도와줍니다.
- **뷰**는 ‘바로 가기 아이콘’과 비슷한 개념입니다. 한마디로 정의하면 ‘가상의 테이블’이라고 할 수 있습니다.
- **스토어드 프로시저**는 여러 개의 SQL을 묶어주거나, 프로그래밍 기능을 제공해줍니다.

# [마무리 정리]

한글 용어	영문 용어	설명
데이터베이스 개체	Database Object	테이블, 뷰, 인덱스, 스토어드 프로시저 등 데이터베이스 안에 저장되는 개체
실행 계획	Execution Plan	SQL을 실행할 때, 인덱스 사용 여부를 확인할 수 있는 워크벤치의 화면
전체 테이블 검색	Full Table Scan	테이블의 모든 데이터를 훑어서 원하는 데이터를 찾아내는 것을 말함. 책 전체를 찾아보는 것과 비슷함
인덱스 검색	Index Scan	인덱스를 통해서 데이터를 찾는 것을 말함. 책 뒤의 찾아보기를 사용한 것과 비슷함
구분 문자	DELIMITER	스토어드 프로시저를 묶어주는 예약어
호출	CALL	스토어드 프로시저를 호출하는 예약어
개체 생성문	CREATE	데이터베이스 개체를 생성할 때 사용하는 예약어
개체 삭제문	DROP	데이터베이스 개체를 삭제할 때 사용하는 예약어

## [마무리 확인 문제]

1. 다음 용어 중에서 데이터베이스 객체가 아닌 것을 모두 고르세요.

인덱스, 뷰, 열 이름, 스토어드 프로시저, 데이터 형식, 트리거, 함수, 커서, 기본 키

2. 다음은 인덱스에 대한 설명입니다. 거리가 먼 것을 모두 고르세요.

- ① 인덱스는 책 뒤의 ‘찾아보기’와 비슷한 개념입니다.
- ② 데이터 건수가 적어도 인덱스의 효과를 체감할 수 있습니다.
- ③ 인덱스는 테이블을 생성하면 자동으로 생성됩니다.
- ④ 인덱스를 생성하는 SQL은 CREATE INDEX 문입니다.

## [마무리 확인 문제]

**3.** 다음은 뷰에 대한 설명입니다. 거리가 먼 것을 하나 고르세요.

- ① 가상의 테이블이라고 부릅니다.
- ② 생성하면 검색 속도가 향상됩니다.
- ③ 윈도우즈의 바로 가기 아이콘과 개념이 비슷합니다.
- ④ 뷰를 만든 후에는 테이블과 동일하게 사용하면 됩니다.

**4.** 다음은 스토어드 프로시저에 대한 설명입니다. 거리가 먼 것을 모두 고르세요.

- ① 프로그래밍 기능을 제공합니다.
- ② 여러 개의 SQL을 하나로 묶을 수 있습니다.
- ③ 스토어드 프로시저를 만들 때는 DELIMITER로 묶어줍니다.
- ④ 스토어드 프로시저는 SELECT 문으로 호출합니다.
- ⑤ 스토어드 프로시저를 DELETE 문으로 삭제합니다.