



## Chapter 03

# SQL 기초

MySQL로 배우는 데이터베이스 개론과 실습

# 관계대수와 관계해석

## ❖ 관계대수

- 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어
- 릴레이션을 처리하기 위해 연산자와 연산규칙을 제공하는 언어로 피연산자가 릴레이션이고 결과도 릴레이션이다.
- 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.

연산자	기호	의미
Select (선택)	$\sigma$	조건(Predicate)을 만족하는 튜플들의 부분 집합 (수평 연산) ★
Project (추출)	$\pi$	속성들의 부분 집합, 중복은 제거됨 (수직 연산) ★
Join (조인)	$\bowtie$ ex) $R \bowtie S$	두 개의 릴레이션이 공통으로 가지고 있는 속성을 이용하여 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산 ★ 두 개의 릴레이션의 연관된 튜플들을 결합
Division (나누기)	$\div$ ex) $R \div S$	R릴레이션에서 S릴레이션의 속성 도메인 값과 일치하는 R릴레이션의 튜플들을 찾아내는 연산 ★

연산자	기호	의미
Union (합집합)	$\cup$ ex) $R \cup S$	두 개의 릴레이션의 합이 추출되고, 중복은 제거됨
Intersection (교집합)	$\cap$ ex) $R \cap S$	R릴레이션과 S릴레이션의 중복되는 값들만 추출
Difference (차집합)	$-$ ex) $R - S$	R릴레이션에서 S릴레이션에 중복되지 않는 값들만 추출
Cartesian Product (교차곱)	$\times$ ex) $R \times S$	두 릴레이션의 가능한 모든 튜플들의 집합, 차수(Degree)는 더하고, 카디널리티(Cardinality)는 곱해서 값을 구함 ★

# 관계대수와 관계해석

## ❖ 관계해석

- 관계 데이터 모델의 제안자인 Codd가 수학의 Predicate Calculus(술어 해석)에 기반을 두고 관계 데이터베이스를 위해 제안
- 관계형 데이터의 연산을 표현하는 방법으로 원하는 정보를 정의할 때는 계산수식을 사용함
- 관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특성을 지님
- 질의어로 표현한다.

구분	구성요소	기호	설명
연산자	OR 연산	$\vee$	원자식 간 "또는"이라는 관계로 연결
	AND 연산	$\wedge$	원자식 간 "그리고"라는 관계로 연결
	NOT 연산	$\neg$	원자식에 대해 부정
정량자	전칭 정량자 (Universal Quantifier)	$\forall$	모든 가능한 튜플 "For All" # All의 'A'를 뒤집은 형태 ★
	존재 정량자 (Existential Quantifier)	$\exists$	어떤 튜플 하나라도 존재 "There Exists" # Exists의 'E'를 뒤집은 형태 ★

# 관계대수와 관계해석

- ❖ 기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력면에서 동등하여 관계대수로 표현한 식은 관계해석으로 표현할 수 있다.

구분	관계대수	관계해석
특징	절차적 언어(순서 명시)	비절차적 언어(계산 수식의 유연적 사용), 프레딕트 해석(Predicate Calculus) 기반
목적	어떻게 유도하는가?(How)	무엇을 얻을 것인가?(What)
종류	순수관계 연산자, 일반집합 연산자	튜플 관계 해석, 도메인 관계 해석

# 목차

**01** SQL 학습을 위한 준비

**02** SQL 개요

**03** 데이터 조작용어 - 검색

**04** 데이터 정의어

**05** 데이터 조작용어 - 삽입, 수정, 삭제

# 학습목표

- ❖ SQL의 개념과 주요 명령어를 알아본다.
- ❖ SELECT 문을 이용하여 질의를 처리하는 방법을 알아본다.
- ❖ 집계 함수와 GROUP BY 문을 이용하여 질의를 처리하는 방법을 알아본다.
- ❖ 두 개 이상의 테이블을 조회하여 질의를 처리하는 방법을 알아본다.
- ❖ DDL로 테이블의 구조를 정의하고 변경하는 방법을 알아본다.
- ❖ DML로 데이터를 삽입, 수정, 삭제하는 방법을 알아본다.

# 01. SQL 학습을 위한 준비

1. 마당서점의 데이터
2. 마당서점 데이터베이스의 사용자들
3. MySQL과 샘플 데이터 설치



# 1. 마당서점의 데이터



그림 3-1 마당서점 현황



# 1. 마당서점의 데이터

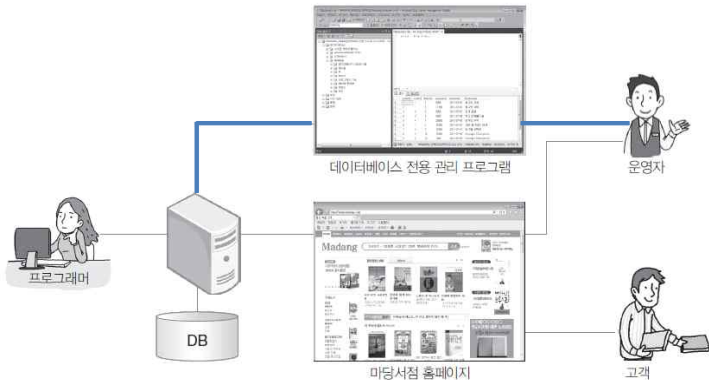


그림 3-2 마당서점 운영 시스템 환경

# 1. 마당서점의 데이터

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

Book 테이블

custid	name	address	phone
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	박세리	대한민국 대전	NULL

Customer 테이블

orderid	custid	bookid	saleprice	orderdate
1	1	1	6000	2014-07-01
2	1	3	21000	2014-07-03
3	2	5	8000	2014-07-03
4	3	6	6000	2014-07-04
5	4	7	20000	2014-07-05
6	1	2	12000	2014-07-07
7	4	8	13000	2014-07-07
8	3	10	12000	2014-07-08
9	2	10	7000	2014-07-09
10	3	8	13000	2014-07-10

Orders 테이블

# 1. 마당서점의 데이터

Book(bookid, bookname, publisher, price)

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000

Orders(orderid, custid, bookid, saleprice, orderdate)

orderid	custid	bookid	saleprice	orderdate
1	1	1	6000	2014-07-01

Customer(custid, name, address, phone)

custid	name	address	phone
1	박지성	영국 맨체스타	000-5000-0001

그림 3-6 마당서점의 데이터 구성도

\* 실무에서는 릴레이션 대신 테이블을, 속성 대신 열(column)을, 튜플 대신 행(row)이라는 용어를 더 많이 사용한다.

## 2. 마당서점 데이터베이스의 사용자들

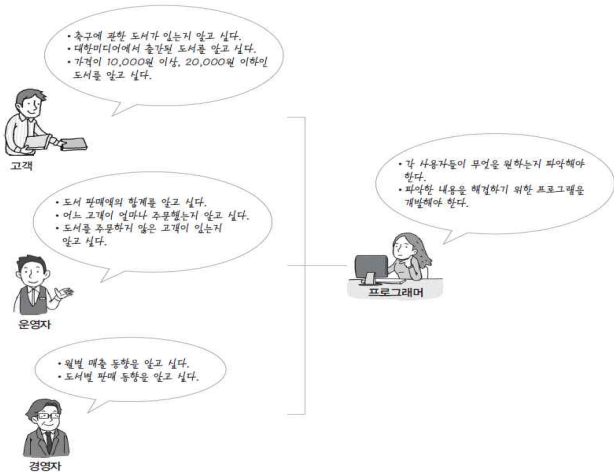


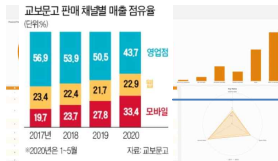
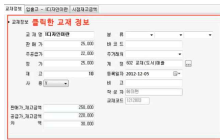
그림 3-7 사용자 그룹별로 원하는 정보

## 2. 마당서점 데이터베이스의 사용자들

## 고객

은영자

경영자



# 프로그래머

### 3. MySQL 샘플 데이터 설치

#### ❖ MySQL 설치(부록 A 참조)

- [www.mysql.com](http://www.mysql.com) 접속 – [DOWNLOADS] 클릭
- 시스템 관리자 계정 : root, 비밀번호 : root
- 설치가 완료되면 [시작]-[모든 프로그램]-[MySQL]-[MySQL Workbench] GUI 데이터베이스 관리 화면이 작동된다.

MySQL 버전	운영체제	윈도우(32비트, 64비트)	LINUX
MySQL Community Server (8.0)		○	○
MySQL Community Server (5.7) - (이전 버전)		○	○
MySQL Enterprise Edition (8.0, 상용)		고급 기능	고급 기능

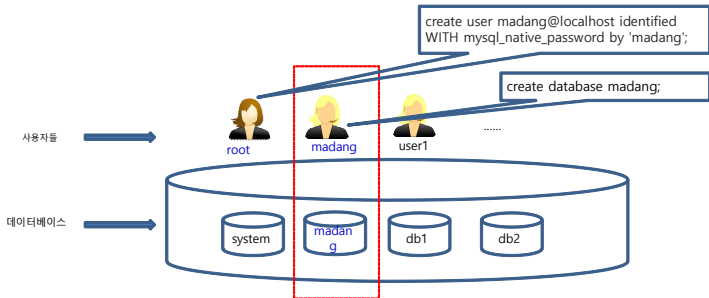
#### <여기서 잠깐> MySQL 설치 소요시간

MySQL DBMS는 설치 파일의 용량이 300MB 이상 되기 때문에 다운로드에서 설치까지 시간이 많이 소요된다.  
설치 시간을 충분히 갖는 것이 좋다.

### 3. MySQL 샘플 데이터 설치

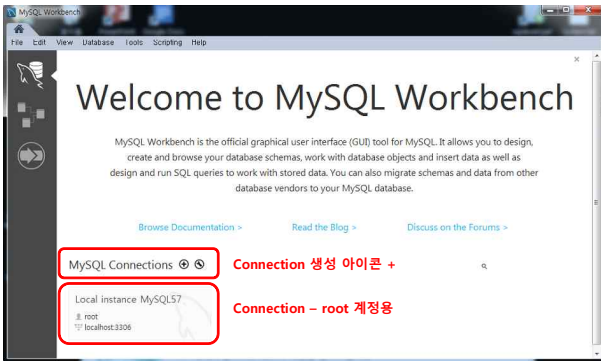
#### ❖ 실습 데이터베이스 설치(부록 B 참조)

- root 계정에서 madang 사용자 및 madang 데이터베이스 생성 (스크립트 참조 - demo\_madang.sql)
- 사용자 이름(madang), 데이터베이스 이름(madang)을 편의상 동일하게 함
- madang 사용자 계정에서 샘플 데이터베이스 설치



### 3. MySQL 샘플 데이터 설치

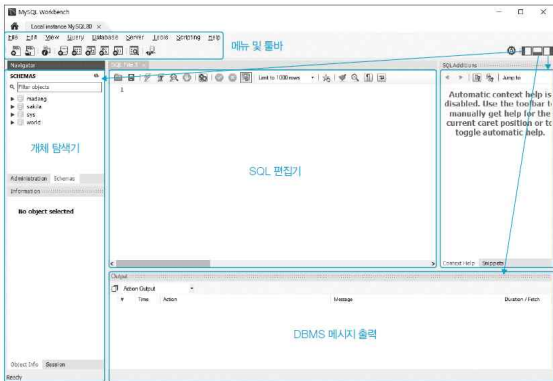
#### ❖ MySQL Workbench





### 3. MySQL 샘플 데이터 설치

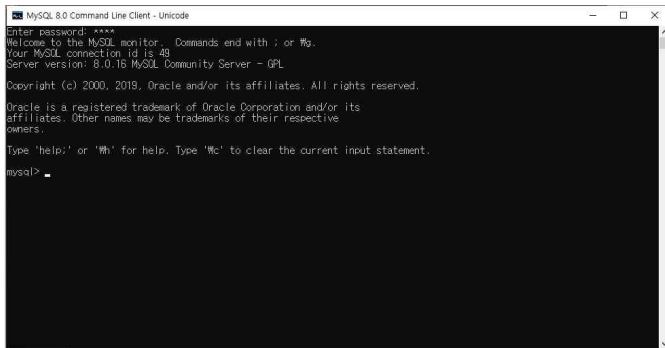
#### ❖ MySQL Workbench 시작



[그림 3-9] Workbench SQL 개발 화면

### 3. MySQL 샘플 데이터 설치

#### ❖ MySQL 명령창



```
MySQL 8.0 Command Line Client - Unicode
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 49
Server version: 8.0.16 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

그림 3-12 Command Line Client에서 MySQL 실행

### 3. MySQL 샘플 데이터 설치

#### ❖ MySQL 명령창

다음은 MySQL 명령창에서 'testdb'라는 데이터베이스를 생성하는 간략한 예제이다.

```
mysql> show databases;
mysql> create database testdb;
mysql> use testdb;
mysql> CREATE TABLE test
      (id smallint unsigned not null auto_increment,
       name varchar(20) not null);
mysql> show tables;
mysql> INSERT INTO tablename(id, name) VALUES (1, 'Sample data' );
mysql> SELECT * FROM test;
```

## 02. SQL 개요

SQL은 IBM이 SEQUEL이라는 이름으로 개발한 관계형 데이터베이스 언어이다.

이후 ANSI에 의해 관계형 데이터베이스 표준 언어로 승인되었다.

SQL은 C 같은 완전한 프로그래밍 언어가 아니라, 데이터 부속어라고 부르는데, 그 이유는 데이터베이스의 데이터와 메타 데이터(데이터 구조에 대한 데이터)를 생성하고 처리하는 문법만 갖고 있기 때문이다.

SQL은 DBMS에 직접 입력해 사용할 수 있고, 자바나 C로 작성된 클라이언트/서버 응용 프로그램이나 HTML 웹페이지 문서에 삽입하여 사용할 수도 있고, 보고서나 데이터 추출 프로그램에서도 사용할 수 있다.



# SQL 개요



그림 3-14 SQL 문을 사용해 자료를 찾는 과정

# SQL 개요

표 3-1 SQL과 일반 프로그래밍 언어의 차이점

구분	SQL	일반 프로그래밍 언어
용도	데이터베이스에서 데이터를 추출하여 문제 해결	모든 문제 해결
입출력	입력은 테이블, 출력도 테이블	모든 형태의 입출력 가능
번역	DBMS	컴파일러
사용 예	SELECT * FROM Book;	int main() {...}

# SQL 개요

## ❖ SQL 기능에 따른 분류

### ■ 데이터 정의어(DDL)

테이블이나 관계의 구조를 생성하는 데 사용하며 CREATE, ALTER, DROP 문 등이 있음

### ■ 데이터 조작어(DML)

테이블에 데이터를 검색, 삽입, 수정, 삭제하는 데 사용하며 SELECT, INSERT, DELETE, UPDATE 문 등이 있음. 여기서 SELECT 문은 특별히 질의어(query)라고 함

### ■ 데이터 제어어(DCL)

데이터의 사용 권한을 관리하는 데 사용하며 GRANT, REVOKE 문 등이 있음

# SQL 개요

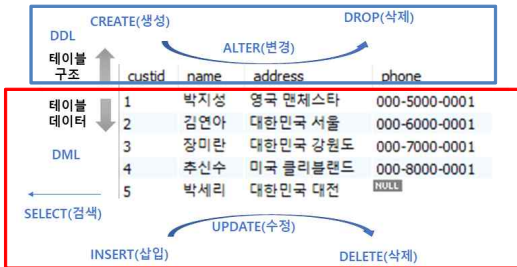


그림 3-15 데이터 정의어와 데이터 조작어의 주요 명령어



# SQL 개요

예) 김연아 고객의 전화번호를 찾으시오.

```
SELECT phone
FROM Customer
WHERE name='김연아'
```

## ① FROM Customer

custid	name	address	phone
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	박세리	대한민국 대전	NULL

## ② WHERE name='김연아'

custid	name	address	phone
2	김연아	대한민국 서울	000-6000-0001

## ③ SELECT phone

phone
000-6000-0001

그림 3-16 SQL 문의 내부적 실행 순서

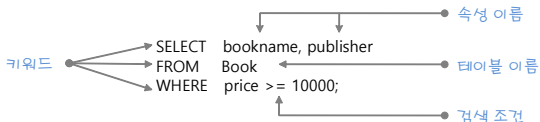
### 03. 데이터 조작어(DML) - 검색

1. SELECT 문 - 질의어
2. 집계 함수와 GROUP BY
3. 두 개 이상 테이블에서 SQL 질의



# 1. SELECT 문

## ❖ SELECT 문의 구성요소



## ❖ SELECT 문의 기본 문법

```
SELECT [ALL | DISTINCT] 속성이름(들)
FROM      테이블이름(들)
[WHERE    검색조건(들)]
[GROUP BY 속성이름]
[HAVING   검색조건(들)]
[ORDER BY 속성이름 [ASC | DESC]]
```

[ ] : 대괄호 안의 SQL 예약어들은 선택적으로 사용한다.

| : 선택 가능한 문법들 중 한 개를 사용할 수 있다.

# 1. SELECT 문

## ❖ SELECT/FROM\_서점에 어떤 도서가 있는지 알고 싶다

질의 3-1 모든 도서의 이름과 가격을 검색하시오.

```
SELECT  bookname, price
FROM    Book;
```

bookname	price
축구의 역사	7000
축구하는 여자	13000
축구의 이해	22000
골프 바이블	35000
피겨 교본	8000
역도 단계별기술	6000
야구의 추억	20000
야구를 부탁해	13000
올림픽 이야기	7500
Olympic Champions	13000

(질의 3-1 변형) 모든 도서의 가격과 이름을 검색하시오.

```
SELECT  price, bookname
FROM    Book;
```

price	bookname
7000	축구의 역사
13000	축구하는 여자
22000	축구의 이해
35000	골프 바이블
8000	피겨 교본
6000	역도 단계별기술
20000	야구의 추억
13000	야구를 부탁해
7500	올림픽 이야기
13000	Olympic Champions

# 1. SELECT 문

## ❖ SELECT/FROM\_서점에 어떤 도서가 있는지 알고 싶다

질의 3-2 모든 도서의 도서번호, 도서이름, 출판사, 가격을 검색하시오.

```
SELECT bookid, bookname, publisher, price
FROM Book;
```

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

```
SELECT *
FROM Book;
```

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

# 1. SELECT 문

## ❖ SELECT/FROM\_서점에 어떤 도서가 있는지 알고 싶다

질의 3-3 도서 테이블에 있는 모든 출판사를 검색하시오.

```
SELECT publisher
FROM Book;
```

publisher	
굿스포츠	
나무수	
대한미디어	
대한미디어	
굿스포츠	
굿스포츠	
이상미디어	
이상미디어	
삼성당	
Pearson	

SQL은 관계대수와 달리 기본적으로 중복을 제거하지 않는다.

※ 중복을 제거하고 싶으면 DISTINCT라는 키워드를 사용한다.

```
SELECT DISTINCT publisher
FROM Book;
```

publisher	
굿스포츠	
나무수	
대한미디어	
이상미디어	
삼성당	
Pearson	

# 1. SELECT 문

❖ WHERE 조건\_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

표 3-5 WHERE 절에 조건으로 사용할 수 있는 술어(predicate)

술어	연산자	예
비교	=, <>, <, <=, >, >=	price < 20000
범위	BETWEEN	price BETWEEN 10000 AND 20000
집합	IN, NOT IN	price IN (10000, 20000, 30000)
패턴	LIKE	bookname LIKE '축구의 역사'
NULL	IS NULL, IS NOT NULL	price IS NULL
복합조건	AND, OR, NOT	(price < 20000) AND (bookname LIKE '축구의 역사')

# 1. SELECT 문

❖ WHERE 조건\_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

## ■ 비교

질의 3-4 가격이 20,000원 미만인 도서를 검색하시오.

```
SELECT *  
FROM Book  
WHERE price < 20000;
```

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000



# 1. SELECT 문

❖ WHERE 조건\_가격이 10,000원 이상 20,000원 이하인 도서가 무엇인지 알고 싶다

## ■ 범위

질의 3-5 가격이 10,000원 이상 20,000원 이하인 도서를 검색하시오.

```
SELECT *  
FROM Book  
WHERE price >= 10000 AND price <= 20000;
```

bookid	bookname	publisher	price
2	축구아는 여자	나무수	13000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
10	Olympic Champions	Pearson	13000

※ BETWEEN은 논리 연산자인 AND를 사용할 수 있다.

```
SELECT *  
FROM Book  
WHERE price BETWEEN 10000 AND 20000;
```

bookid	bookname	publisher	price
2	축구아는 여자	나무수	13000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
10	Olympic Champions	Pearson	13000

# 1. SELECT 문

## ❖ WHERE 조건\_in / not in

### ■ 집합

질의 3-6 출판사가 '굿스포츠' 혹은 '대한미디어'인 도서를 검색하시오.

```
SELECT *  
FROM Book  
WHERE publisher IN ('굿스포츠', '대한미디어');
```

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000

※ 출판사가 '굿스포츠' 혹은 '대한미디어'가 아닌 도서를 검색하시오.

```
SELECT *  
FROM Book  
WHERE publisher NOT IN ('굿스포츠', '대한미디어');
```

bookid	bookname	publisher	price
2	축구하는 여자	나무수	13000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

# 1. SELECT 문

## ❖ WHERE 조건\_like

### ■ 패턴

질의 3-7 '축구의 역사'를 출간한 출판사를 검색하시오.

```
SELECT  bookname, publisher
FROM    Book
WHERE   bookname LIKE '축구의 역사';
```

bookname	publisher
축구의 역사	굿스포츠

질의 3-8 도서이름에 '축구'가 포함된 출판사를 검색하시오.

```
SELECT  bookname, publisher
FROM    Book
WHERE   bookname LIKE '%축구%';
```

bookname	publisher
축구의 역사	굿스포츠
축구하는 여자	나무수
축구의 이해	대한미디어

# 1. SELECT 문

❖ WHERE 조건\_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

질의 3-9 도서이름의 왼쪽 두 번째 위치에 '구'라는 문자열을 갖는 도서를 검색하시오.

```
SELECT *
FROM Book
WHERE bookname LIKE '_구%';
```

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000

표 3-6 와일드 문자의 종류

와일드 문자	의미	사용 예
+	문자열을 연결	'골프' + '바이블' : '골프 바이블'
%	0개 이상의 문자열과 일치	'%축구%' : 축구를 포함하는 문자열
[ ]	1개의 문자와 일치	'[0-5]%' : 0-5 사이 숫자로 시작하는 문자열
[^]	1개의 문자와 불일치	'[^0-5]%' : 0-5 사이 숫자로 시작하지 않는 문자열
_	특정 위치의 1개의 문자와 일치	'_구%' : 두 번째 위치에 '구'가 들어가는 문자열

# 1. SELECT 문

## ❖ WHERE 조건\_복합조건

### ■ 복합조건

질의 3-10 축구에 관한 도서 중 가격이 20,000원 이상인 도서를 검색하시오.

```
SELECT *  
FROM Book  
WHERE bookname LIKE '%축구%' AND price >= 20000;
```

bookid	bookname	publisher	price
3	축구의 이해	대한미디어	22000

질의 3-11 출판사가 '굿스포츠' 혹은 '대한미디어'인 도서를 검색하시오.

```
SELECT *  
FROM Book  
WHERE publisher='굿스포츠' OR publisher='대한미디어';
```

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000

# 1. SELECT 문

## ❖ ORDER BY\_도서를 이름순으로 보고 싶다

질의 3-12 도서를 이름순으로 검색하시오.

```
SELECT      *  
FROM        Book  
ORDER BY    bookname;
```

bookid	bookname	publisher	price
10	Olympic Champions	Pearson	13000
4	골프 바이블	대한미디어	35000
8	야구를 부탁해	이상미디어	13000
7	야구의 추억	이상미디어	20000
6	역도 단계별기술	굿스포츠	6000
9	올림픽 이야기	삼성당	7500
2	축구하는 여자	나무수	13000
1	축구의 역사	굿스포츠	7000
3	축구의 이해	대한미디어	22000
5	피겨 교본	굿스포츠	8000

\* SQL 문의 실행 결과 행의 순서는 DBMS에 저장된 위치에 따라 결정됨  
실행 결과를 특정 순서대로 출력하고 싶으면 ORDER BY 절을 사용한다.

# 1. SELECT 문

❖ ORDER BY\_도서를 이름순으로 보고 싶다

질의 3-13    도서를 가격순으로 검색하고, 가격이 같으면 이름순으로 검색하시오.

```
SELECT      *  
FROM        Book  
ORDER BY    price, bookname;
```

bookid	bookname	publisher	price
6	역도 단계별기술	굿스포츠	6000
1	축구의 역사	굿스포츠	7000
9	올림픽 이야기	삼성당	7500
5	피겨 교본	굿스포츠	8000
10	Olympic Champions	Pearson	13000
8	야구를 부탁해	이상미디어	13000
2	축구하는 여자	나무수	13000
7	야구의 추억	이상미디어	20000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000

# 1. SELECT 문

## ❖ ORDER BY\_도서를 이름순으로 보고 싶다

질의 3-14 도서를 가격의 내림차순으로 검색하시오. 만약 가격이 같다면 출판사의 오름차순으로 검색한다.

```
SELECT      *  
FROM        Book  
ORDER BY    price DESC, publisher ASC;
```

bookid	bookname	publisher	price
4	골프 바이블	대한미디어	35000
3	축구의 이해	대한미디어	22000
7	야구의 추억	이상미디어	20000
10	Olympic Champions	Pearson	13000
2	축구하는 여자	나무수	13000
8	야구를 부탁해	이상미디어	13000
5	피겨 교본	굿스포츠	8000
9	올림픽 이야기	삼성당	7500
1	축구의 역사	굿스포츠	7000
6	역도 단계별기술	굿스포츠	6000





## 2. 집계 함수(Aggregate Function)와 GROUP BY

### ❖ 집계 함수\_도서 판매액의 합계를 알고 싶다

질의 3-15 고객이 주문한 도서의 총 판매액을 구하시오.

```
SELECT SUM(saleprice)
FROM Orders;
```

SUM(saleprice)
118000

※ 의미 있는 열 이름을 출력하고 싶으면 속성이름의 별칭을 지칭하는 AS 키워드를 사용하여 열 이름을 부여한다.

```
SELECT SUM(saleprice) AS 총매출
FROM Orders;
```

총매출
118000

## 2. 집계 함수와 GROUP BY

### ❖ 집계 함수\_도서 판매액의 합계를 알고 싶다

질의 3-16 2번 김연아 고객이 주문한 도서의 총 판매액을 구하시오.

```
SELECT SUM(saleprice) AS 총매출
FROM Orders
WHERE custid=2;
```

총매출
-----

15000
-------

질의 3-17 고객이 주문한 도서의 총 판매액, 평균값, 최저가, 최고가를 구하시오.

```
SELECT SUM(saleprice) AS Total,
       AVG(saleprice) AS Average,
       MIN(saleprice) AS Minimum,
       MAX(saleprice) AS Maximum
FROM Orders;
```

	Total	Average	Minimum	Maximum
▶	118000	11800.0000	6000	21000

## 2. 집계 함수와 GROUP BY

### ❖ 집계 함수\_도서 판매액의 합계를 알고 싶다

질의 3-18 마당서점의 도서 판매 건수를 구하시오.

```
SELECT    COUNT(*)  
FROM      Orders;
```

	COUNT(*)
▶	10

집계함수를 사용할 때 두 가지 주의사항

1. 집계함수는 null 인 속성값은 제외하고 계산한다.
2. 집계함수는 WHERE 절에서는 사용할 수 없고,  
SELECT 절이나 HAVING 절에서만 사용할 수 있다.

## 2. 집계 함수와 GROUP BY

표 3-7 집계 함수의 종류

집계 함수	문법	사용 예
SUM	SUM([ALL   DISTINCT] 속성이름)	SUM(price)
AVG	AVG([ALL   DISTINCT] 속성이름)	AVG(price)
COUNT	COUNT([ALL   DISTINCT] 속성이름   *)	COUNT(*)
MAX	MAX([ALL   DISTINCT] 속성이름)	MAX(price)
MIN	MIN([ALL   DISTINCT] 속성이름)	MIN(price)

함수	의미	사용가능한 속성의 타입
COUNT	속성 값의 개수	모든 데이터
MAX	속성 값의 최댓값	
MIN	속성 값의 최솟값	
SUM	속성 값의 합계	숫자 데이터
AVG	속성 값의 평균	

## 2. 집계 함수와 GROUP BY

❖ **GROUP BY** 어느 고객이 얼마나 주문했는지 알고 싶다

질의 3-19 고객별로 주문한 도서의 총 수량과 총 판매액을 구하시오.

```
SELECT custid, COUNT(*) AS 도서수량, SUM(saleprice) AS 총액
FROM Orders
GROUP BY custid;
```

custid	도서수량	총액
1	3	39000
2	2	15000
3	3	31000
4	2	33000

orderid	custid	bookid	saleprice	orderdate
1	1	1	6000	2014-07-01
2	1	3	21000	2014-07-03
6	1	2	12000	2014-07-07
3	2	5	8000	2014-07-03
9	2	10	7000	2014-07-09
4	3	6	6000	2014-07-04
8	3	10	12000	2014-07-08
10	3	8	13000	2014-07-10
5	4	7	20000	2014-07-05
7	4	8	13000	2014-07-07

custid	도서수량	총액
1	3	39000
2	2	15000
3	3	31000
4	2	33000

그림 3-18 GROUP BY 질의 수행

## 2. 집계 함수와 GROUP BY

❖ **GROUP BY** 어느 고객이 얼마나 주문했는지 알고 싶다

질의 3-20 가격이 8,000원 이상인 도서를 구매한 고객에 대하여 고객별 주문 도서의 총 수량을 구하시오. 단, 두 권 이상 구매한 고객만 구한다.

```
SELECT    custid, COUNT(*) AS 도서수량
FROM      Orders
WHERE     saleprice >= 8000
GROUP BY  custid
HAVING    count(*) >= 2;
```

	custid	도서수량
▶	1	2
	4	2
	3	2

\* Having 은 group by 절의 결과 나타나는 그룹을 제한하는 역할을 한다.  
즉 그룹에 대한 조건은 WHERE 가 아닌 HAVING 키워드와 함께 작성한다.

## 2. 집계 함수와 GROUP BY

표 3-8 GROUP BY와 HAVING 절의 문법과 주의사항

문법	주의사항
GROUP BY <속성>	<p>GROUP BY로 튜플을 그룹으로 묶은 후 SELECT 절에는 GROUP BY에서 사용한 &lt;속성&gt;과 집계함수만 나올 수 있음</p> <p>▪ 맞는 예</p> <pre>SELECT      custid, SUM(saleprice) FROM        Orders GROUP BY    custid;</pre> <p>▪ 틀린 예</p> <pre>SELECT      bookid, SUM(saleprice) /* SELECT 절에 bookid 속성이 올 수 없다 */ FROM        Orders GROUP BY    custid;</pre>
HAVING <검색조건>	<p>WHERE 절과 HAVING 절이 같이 포함된 SQL 문은 검색조건이 모호해질 수 있음. HAVING 절은 ① 반드시 GROUP BY 절과 같이 작성해야 하고 ② WHERE 절보다 뒤에 나와야 함. 그리고 ③ &lt;검색조건&gt;에는 SUM, AVG, MAX, MIN, COUNT와 같은 집계함수가 와야 함.</p> <p>▪ 맞는 예</p> <pre>SELECT      custid, COUNT(*) AS 도서수량 FROM        Orders WHERE       saleprice &gt;= 8000 GROUP BY    custid HAVING      COUNT(*) &gt;= 2;</pre> <p>▪ 틀린 예</p> <pre>SELECT      custid, COUNT(*) AS 도서수량 FROM        Orders HAVING      COUNT(*) &gt;= 2 /* 순서가 틀렸다 */ WHERE       saleprice &gt;= 8000 GROUP BY    custid;</pre>

## 아래 SQL의 내부 실행 순서는 ?

```
SELECT      custid, COUNT(*) AS 도서수량
FROM        Orders -----1)
WHERE       saleprice >= 8000
GROUP BY    custid
HAVING      COUNT(*) >= 2
ORDER BY    custid;
```

orderid	custid	bookid	saleprice	orderdate
1	1	1	6000	2014-07-01
2	1	3	21000	2014-07-03
3	2	5	8000	2014-07-03
4	3	6	6000	2014-07-04
5	4	7	20000	2014-07-05
6	1	2	12000	2014-07-07
7	4	8	13000	2014-07-07
8	3	10	12000	2014-07-08
9	2	10	7000	2014-07-09
10	3	8	13000	2014-07-10



## 연습문제

### 1. 마당서점의 고객이 요구하는 다음 질문에 대해 SQL 문을 작성하시오.

- (1) 도서번호가 1인 도서의 이름
- (2) 가격이 20,000원 이상인 도서의 이름
- (3) 박지성의 총 구매액(박지성의 고객번호는 1번으로 놓고 작성)
- (4) 박지성이 구매한 도서의 수(박지성의 고객번호는 1번으로 놓고 작성)

### 2. 마당서점의 운영자와 경영자가 요구하는 다음 질문에 대해 SQL 문을 작성하시오.

- (1) 마당서점 도서의 총 개수
- (2) 마당서점에 도서를 출고하는 출판사의 총 개수
- (3) 모든 고객의 이름, 주소
- (4) 2014년 7월 4일~7월 7일 사이에 주문 받은 도서의 주문번호
- (5) 2014년 7월 4일~7월 7일 사이에 주문 받은 도서를 제외한 도서의 주문번호
- (6) 성이 '김' 씨인 고객의 이름과 주소
- (7) 성이 '김' 씨이고 이름이 '아'로 끝나는 고객의 이름과 주소

### 3. 두 개 이상 테이블에서 SQL 질의

- Customer 테이블을 Orders 테이블과 조건 없이 연결해보자.  
Customer와 Orders 테이블의 합체 결과 **튜플의 개수**는 고객이 다섯 명이고 주문이 열 개이므로 **5×10 해서 50**이 된다.

```
SELECT *
FROM Customer, Orders;
```

custid	name	address	phone	orderid	custid	bookid	saleprice	orderdate
1	박지성	영국 맨체스터	000-5000-0001	1	1	1	6000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	1	1	1	6000	2014-07-01
3	장미란	대한민국 강원도	000-7000-0001	1	1	1	6000	2014-07-01
4	추신수	미국 클리블랜드	000-8000-0001	1	1	1	6000	2014-07-01
5	박세리	대한민국 대전	000-9000-0001	1	1	1	6000	2014-07-01
1	박지성	영국 맨체스터	000-5000-0001	2	1	3	21000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	2	1	3	21000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	2	1	3	21000	2014-07-03
4	추신수	미국 클리블랜드	000-8000-0001	2	1	3	21000	2014-07-03
5	박세리	대한민국 대전	000-9000-0001	2	1	3	21000	2014-07-03
1	박지성	영국 맨체스터	000-5000-0001	3	2	5	8000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
4	추신수	미국 클리블랜드	000-8000-0001	3	2	5	8000	2014-07-03
5	박세리	대한민국 대전	000-9000-0001	3	2	5	8000	2014-07-03
1	박지성	영국 맨체스터	000-5000-0001	4	3	6	6000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	4	3	6	6000	2014-07-04
3	장미란	대한민국 강원도	000-7000-0001	4	3	6	6000	2014-07-04
4	추신수	미국 클리블랜드	000-8000-0001	4	3	6	6000	2014-07-04
... 중략 ...								
1	박지성	영국 맨체스터	000-5000-0001	10	3	8	13000	2014-07-10
2	김연아	대한민국 서울	000-6000-0001	10	3	8	13000	2014-07-10
3	장미란	대한민국 강원도	000-7000-0001	10	3	8	13000	2014-07-10
4	추신수	미국 클리블랜드	000-8000-0001	10	3	8	13000	2014-07-10
5	박세리	대한민국 대전	000-9000-0001	10	3	8	13000	2014-07-10

그림 3-16 Customer와 Orders 테이블의 합체

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개의 테이블을 합체해보자

질의 3-21 고객과 고객의 주문에 관한 데이터를 모두 보이시오.

```
SELECT *  
FROM Customer, Orders  
WHERE Customer.custid =Orders.custid;
```

custid	name	address	phone	orderid	custid	bookid	saleprice	orderdate
1	박지성	영국 맨체스타	000-5000-0001	1	1	1	6000	2014-07-01
1	박지성	영국 맨체스타	000-5000-0001	2	1	3	21000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	4	3	6	6000	2014-07-04
4	추신수	미국 클리블랜드	000-8000-0001	5	4	7	20000	2014-07-05
1	박지성	영국 맨체스타	000-5000-0001	6	1	2	12000	2014-07-07
4	추신수	미국 클리블랜드	000-8000-0001	7	4	8	13000	2014-07-07
3	장미란	대한민국 강원도	000-7000-0001	8	3	10	12000	2014-07-08
2	김연아	대한민국 서울	000-6000-0001	9	2	10	7000	2014-07-09
3	장미란	대한민국 강원도	000-7000-0001	10	3	8	13000	2014-07-10

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개의 테이블을 합체해보자

질의 3-22 고객과 고객의 주문에 관한 데이터를 고객번호 순으로 정렬하여 보이시오.

```
SELECT      *
FROM        Customer, Orders
WHERE       Customer.custid =Orders.custid
ORDER BY    Customer.custid;
```

custid	name	address	phone	orderid	custid	bookid	saleprice	orderdate
1	박지성	영국 맨체스타	000-5000-0001	6	1	2	12000	2014-07-07
1	박지성	영국 맨체스타	000-5000-0001	1	1	1	6000	2014-07-01
1	박지성	영국 맨체스타	000-5000-0001	2	1	3	21000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	9	2	10	7000	2014-07-09
3	장미란	대한민국 강원도	000-7000-0001	4	3	6	6000	2014-07-04
3	장미란	대한민국 강원도	000-7000-0001	8	3	10	12000	2014-07-08
3	장미란	대한민국 강원도	000-7000-0001	10	3	8	13000	2014-07-10
4	추신수	미국 클리블랜드	000-8000-0001	5	4	7	20000	2014-07-05
4	추신수	미국 클리블랜드	000-8000-0001	7	4	8	13000	2014-07-07

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개의 테이블을 합체해보자

질의 3-23    고객의 이름과 고객이 주문한 도서의 판매가격을 검색하시오.

```
SELECT  name, saleprice
FROM    Customer, Orders
WHERE   Customer.custid =Orders.custid;
```

name	saleprice
박지성	6000
박지성	21000
김연아	8000
장미란	6000
추신수	20000
박지성	12000
추신수	13000
장미란	12000
김연아	7000
장미란	13000

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개 이상의 테이블을 합체해보자

질의 3-24 고객별로 주문한 모든 도서의 총 판매액을 구하고, 고객별로 정렬하시오.

```
SELECT    name, SUM(saleprice)
FROM      Customer, Orders
WHERE     Customer.custid =Orders.custid
GROUP BY  Customer.name
ORDER BY  Customer.name;
```

name	SUM(saleprice)
김연아	15000
박지성	39000
장미란	31000
추신수	33000

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개 이상의 테이블을 합쳐보자



그림 3-20 마당서점 데이터 간의 연결

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개 이상의 테이블을 합체해보자

질의 3-25    고객의 이름과 고객이 주문한 도서의 이름을 구하시오.

```
SELECT    Customer.name, Book.bookname
FROM      Customer, Orders, Book
WHERE     Customer.custid =Orders.custid
AND       Orders.bookid =Book.bookid;
```

name	bookname
박지성	축구의 역사
박지성	축구의 이해
김연아	피겨 교본
장미란	역도 단계별기술
추신수	야구의 추억
박지성	축구하는 여자
추신수	야구를 부탁해
장미란	Olympic Champions
김연아	Olympic Champions
장미란	역도를 부탁해



### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개 이상의 테이블을 합체해보자

질의 3-26 가격이 20,000원인 도서를 주문한 고객의 이름과 도서의 이름을 구하시오.

```
SELECT Customer.name, Book.bookname
FROM Customer, Orders, Book
WHERE Customer.custid =Orders.custid AND Orders.bookid =Book.bookid
      AND Book.price =20000;
```

name	bookname
추신수	야구의 추억

### 3. 두 개 이상 테이블에서 SQL 질의

❖ **조인**\_2개 이상의 테이블을 합체해보자

■ **외부조인**

질의 3-27 도서를 구매하지 않은 고객을 포함하여 고객의 이름과 고객이 주문한 도서의 판매가격을 구하시오.

```
SELECT Customer.name, saleprice
FROM Customer LEFT OUTER JOIN Orders
      ON Customer.custid =Orders.custid;
```

name	saleprice
박지성	6000
박지성	21000
김연아	8000
장미란	6000
추신수	20000
박지성	12000
추신수	13000
장미란	12000
김연아	7000
장미란	13000
박세리	NULL

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 조인\_2개 이상의 테이블을 합체해보자

표 3-9 조인 문법

명령	문법	설명
일반 조인	SELECT <속성들> FROM 테이블1, 테이블2 WHERE <조인조건> AND <검색조건>	SQL 문에서는 주로 동등조인을 사용함. 두 가지 문법 중 하나를 사용할 수 있음.
	SELECT <속성들> FROM 테이블1 <b>INNER JOIN</b> 테이블2 <b>ON</b> <조인조건> WHERE <검색조건>	
외부조인	SELECT <속성들> FROM 테이블1 { <b>LEFT</b>   <b>RIGHT</b>   <b>FULL [OUTER]</b> } <b>JOIN</b> 테이블2 <b>ON</b> <조인조건> WHERE <검색조건>	외부조인은 FROM 절에 조인 종류를 적고 ON을 이용하여 조인조건을 명시함.

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 부속질의 SQL 문 내에 또 다른 SQL 문을 작성해보자

질의 3-28 가장 비싼 도서의 이름을 보이시오.

```
SELECT bookname
FROM Book
WHERE price = ( SELECT MAX(price) FROM Book );
```

bookname
골프 바이블

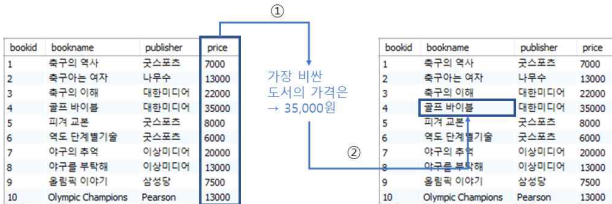


그림 3-21 부속질의의 실행 순서

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 부속질의 SQL 문 내에 또 다른 SQL 문을 작성해보자

질의 3-29 도서를 구매한 적이 있는 고객의 이름을 검색하시오.

```
SELECT name
FROM Customer
WHERE custid IN (SELECT custid FROM Orders);
```

name
박지성
김연아
장미란
추신수

질의 3-30 대한미디어에서 출판한 도서를 구매한 고객의 이름을 보이시오.

```
SELECT name
FROM Customer
WHERE custid IN (SELECT custid
                  FROM Orders
                  WHERE bookid IN (SELECT bookid
                                   FROM Book
                                   WHERE publisher='대한미디어'));
```

name
박지성

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 부속질의 SQL 문 내에 또 다른 SQL 문을 작성해보자

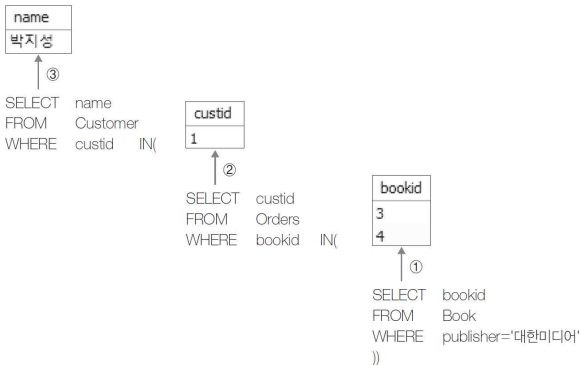


그림 3-22 3단계 부속질의의 실행 순서

### 3. 두 개 이상 테이블에서 SQL 질의

#### ❖ 부속질의 SQL 문 내에 또 다른 SQL 문을 작성해보자

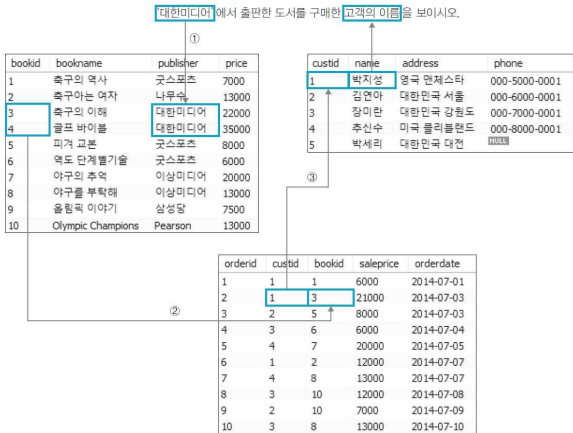


그림 3-23 3단계 부속질의의 실행 순서와 데이터 예