

Lab #4:

Data Retrieval

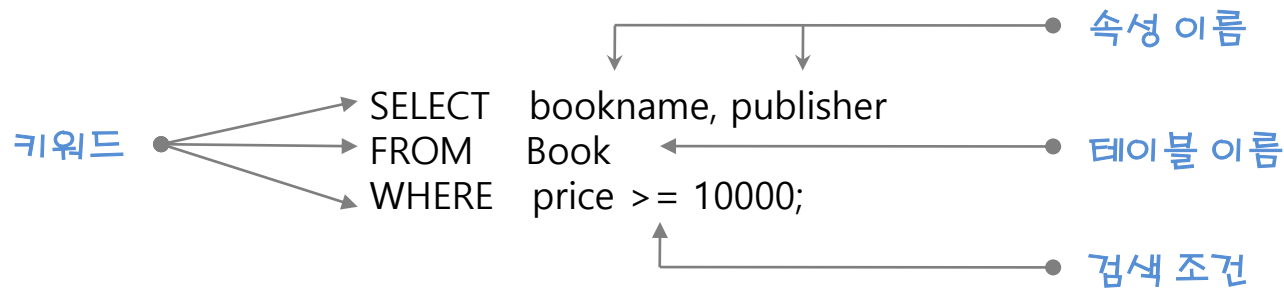
소프트웨어학부
노서영



SELECT Statement

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.

SELECT 문

Source from
MySQL로 배우는 데이터베이스 개론과 실습
한빛아카데미

❖ 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

SELECT 문

Source from
MySQL로 배우는 데이터베이스 개론과 실습
한빛아카데미

❖ 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

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

※ 중복을 제거하고 싶으면 DISTINCT라는 키워드를 사용한다.

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

publisher
굿스포츠
나무수
대한미디어
이상미디어
삼성당
Pearson

SELECT 문

❖ WHERE 조건_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

표 3-5 WHERE 절에 조건으로 사용할 수 있는 술어

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 '축구'의 역사'
NULL	IS NULL, IS NOT NULL	price IS NULL
복합조건	AND, OR, NOT	(price < 20000) AND (bookname LIKE '축구'의 역사')

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

SELECT 문

❖ WHERE 조건_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

- 범위

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

```
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

※ BETWEEN은 논리 연산자인 AND를 사용할 수 있다.

```
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

SELECT 문

❖ WHERE 조건_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

• 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

SELECT 문

❖ WHERE 조건_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

- 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

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

SELECT 문

❖ WHERE 조건_가격이 20,000원 미만인 도서가 무엇인지 알고 싶다

- 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

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	축구아는 여자	나무수	13000
1	축구의 역사	굿스포츠	7000
3	축구의 이해	대한미디어	22000
5	피겨 교본	굿스포츠	8000

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

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	축구아는 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

GROUP BY

집계 함수와 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

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

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

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)

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 절의 수행

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

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 <속성>과 집계함수만 나올 수 있음</p> <ul style="list-style-type: none"> • 맞는 예 <pre>SELECT custid, SUM(saleprice) FROM Orders GROUP BY custid;</pre> • 틀린 예 <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 <검색조건>에는 SUM, AVG, MAX, MIN, COUNT와 같은 집계함수가 와야 함.</p> <ul style="list-style-type: none"> • 맞는 예 <pre>SELECT custid, COUNT(*) AS 도서수량 FROM Orders WHERE saleprice >= 8000 GROUP BY custid HAVING COUNT(*) >= 2;</pre> • 틀린 예 <pre>SELECT custid, COUNT(*) AS 도서수량 FROM Orders HAVING COUNT(*) >= 2 /* 순서가 틀렸다 */ WHERE saleprice >= 8000 GROUP BY custid;</pre>

JOIN

(2개 이상 테이블에서 SQL 질의)

두 개 이상 테이블에서 SQL 질의

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

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

custid	name	address	phone	orderid	custid	bookid	saleprice	orderdate
1	박지성	영국 맨체스타	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	NULL	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	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	NULL	2	1	3	21000	2014-07-03
1	박지성	영국 맨체스타	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	NULL	3	2	5	8000	2014-07-03
1	박지성	영국 맨체스타	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 맨체스타	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	NULL	10	3	8	13000	2014-07-10

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

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

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

두 개 이상 테이블에서 SQL 질의

❖ 조인_2개의 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

두 개 이상 테이블에서 SQL 질의

❖ 조인_2개의 테이블을 합체해보자



그림 3-20 마당서점 데이터 간의 연결

두 개 이상 테이블에서 SQL 질의

❖ 조인_2개의 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

두 개 이상 테이블에서 SQL 질의

❖ 조인_2개의 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

두 개 이상 테이블에서 SQL 질의

❖ 조인_2개의 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

두 개 이상 테이블에서 SQL 질의

❖ 조인_2개의 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 INNER JOIN 테이블2 ON <조인조건> WHERE <검색조건>	
외부조인	SELECT <속성들> FROM 테이블1 {LEFT RIGHT FULL [OUTER]} JOIN 테이블2 ON <조인조건> WHERE <검색조건>	외부조인은 FROM 절에 조인 종류를 적고 ON을 이용하여 조인조건을 명시함.

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

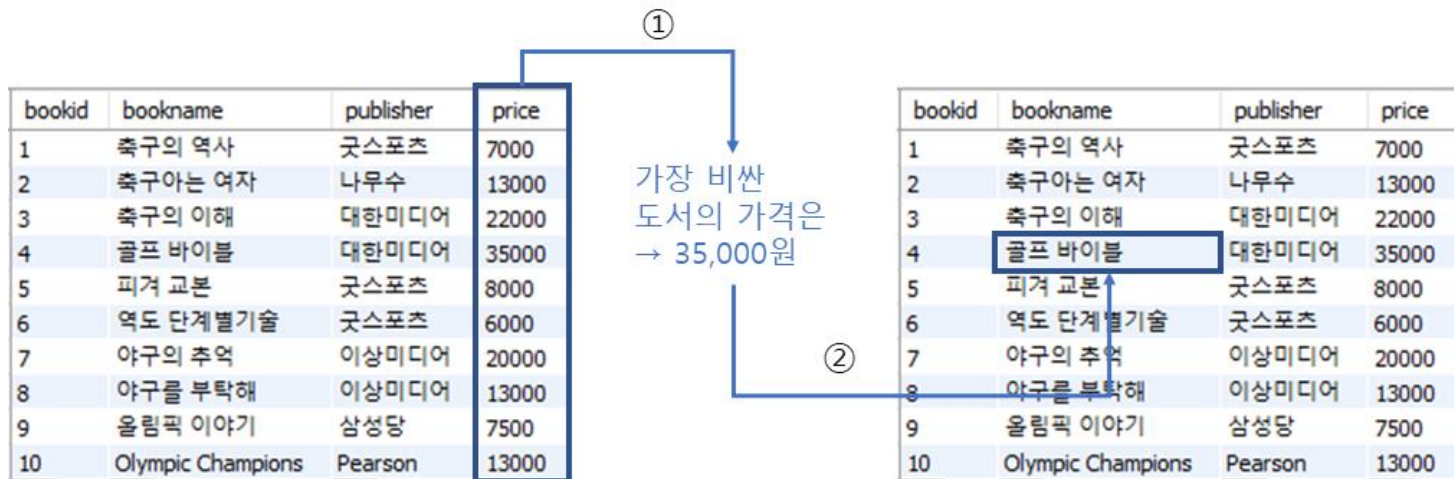


그림 3-21 부속질의의 실행 순서

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

두 개 이상 테이블에서 SQL 질의

❖ **부속질의**_SQL 문 내에 또 다른 SQL 문을 작성해보자

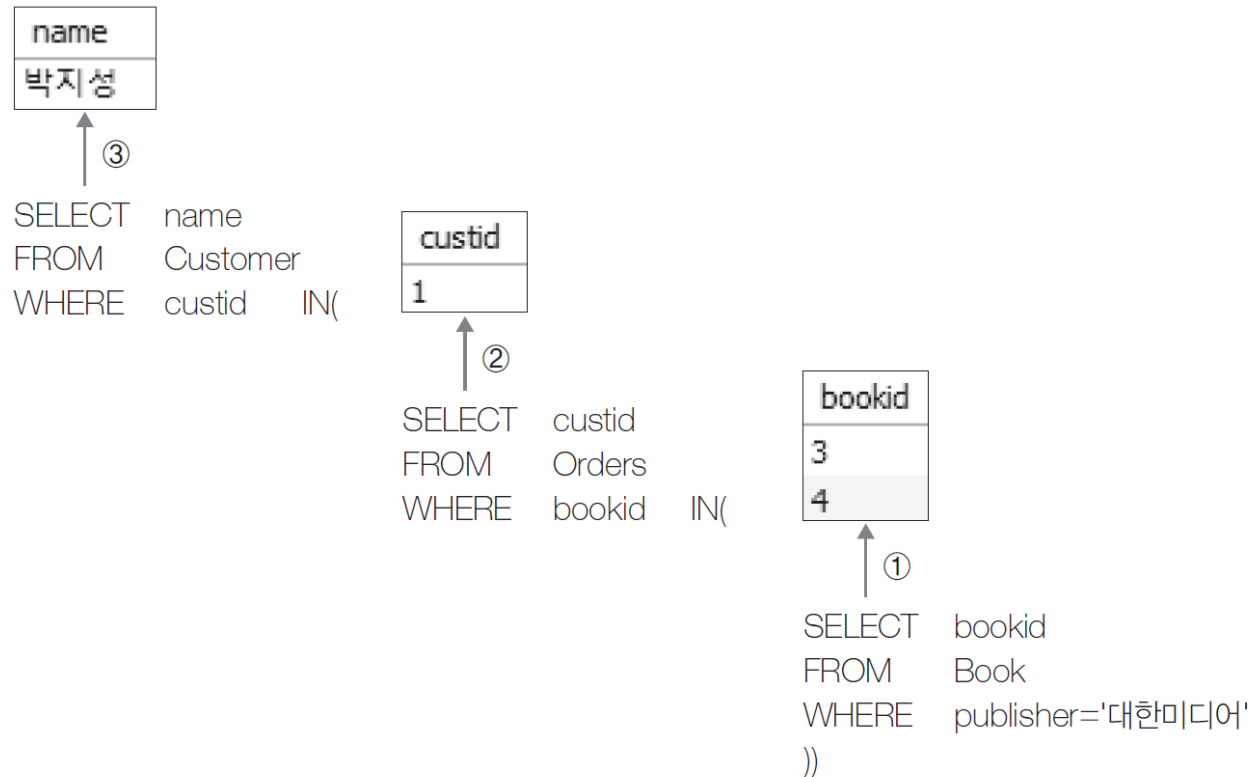
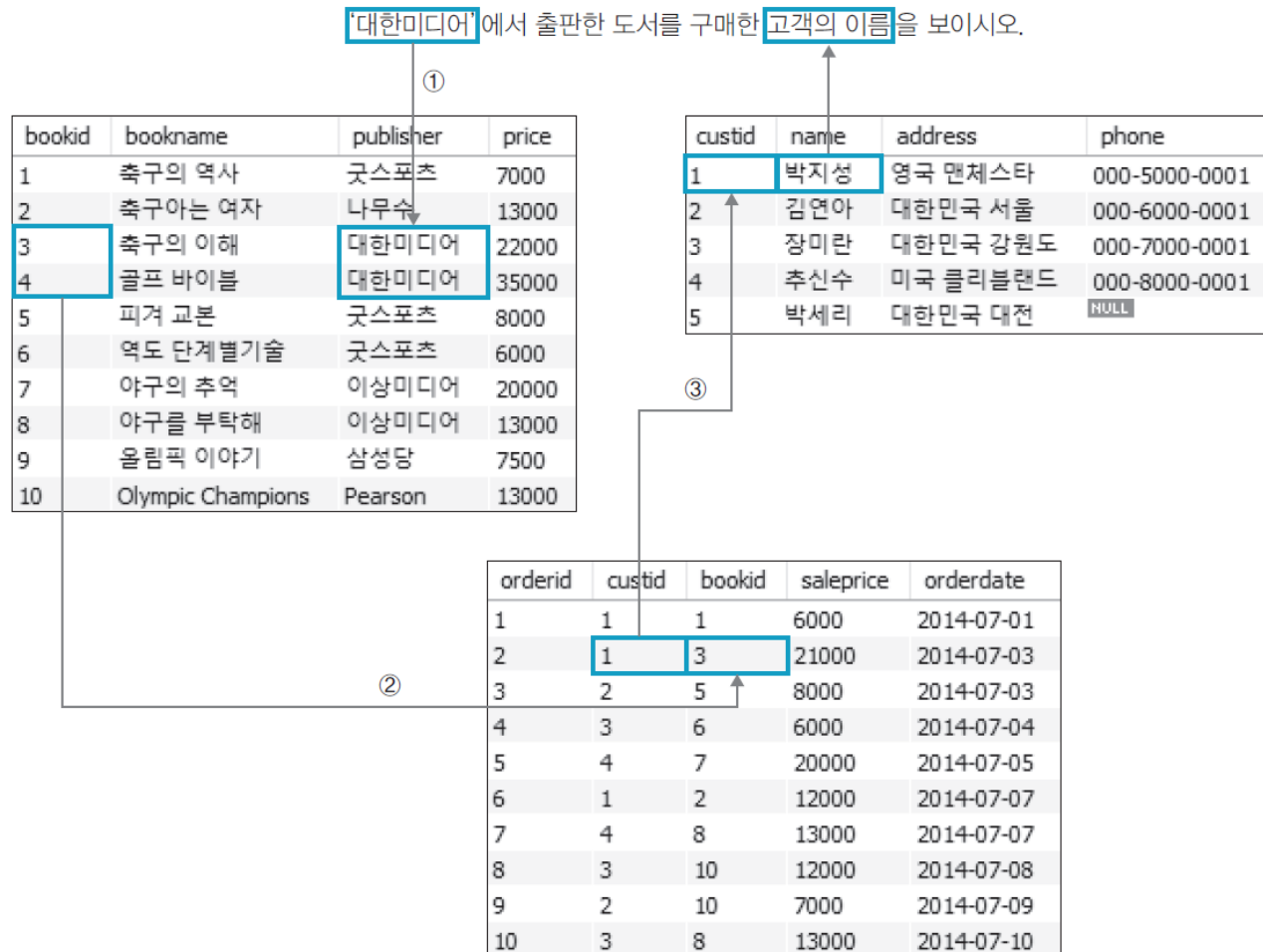


그림 3-22 3단계 부속질의의 실행 순서

두 개 이상 테이블에서 SQL 질의

❖ 부속질의 SQL 문 내에 또 다른 SQL 문을 작성해보자



두 개 이상 테이블에서 SQL 질의

❖ 부속질의(SQL 문 내에 또 다른 SQL 문을 작성해보자)

- 상관 부속질의(correlated subquery)는 상위 부속질의의 투표를 이용하여 하위 부속질을 계산함. 즉 상위 부속질의와 하위 부속질의가 독립적이지 않고 서로 관련을 맺고 있음.

질의 3-31 출판사별로 출판사의 평균 도서 가격보다 비싼 도서를 구하시오.

```
SELECT b1.bookname
FROM Book b1
WHERE b1.price > (SELECT avg(b2.price)
                  FROM Book b2
                  WHERE b2.publisher=b1.publisher);
```

bookname
골프 바이블
피겨 교본
야구의 추억

두 개 이상 테이블에서 SQL 질의

❖ 부속질의 SQL 문 내에 또 다른 SQL 문을 작성해보자

Book 테이블 : b1으로 나타냄

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

b1 테이블의
투플 t에
해당하는
출판사를
b2 테이블로
가져가서,
같은 출판사를
가진 투플들의
price 평균을
구한다.

Book 테이블 : b2로 나타냄

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

avg(b2.price)

28500

b1.price > avg(b2.price)

그림 3-24 상관 부속질의의 데이터 예

두 개 이상 테이블에서 SQL 질의

❖ 집합 연산_도서를 주문한 고객을 알고 싶다

- 합집합 UNION, 차집합 MINUS, 교집합 INTERSECT

질의 3-32 대한민국에서 거주하는 고객의 이름과 도서를 주문한 고객의 이름을 보이시오.

```
SELECT name
FROM Customer
WHERE address LIKE '대한민국%'
UNION
SELECT name
FROM Customer
WHERE custid IN (SELECT custid FROM Orders);
```

name
김연아
장미란
박세리
박지성
추신수

{고객 이름} = {대한민국에 거주하는 고객 이름} \cup {도서를 주문한 고객 이름}

두 개 이상 테이블에서 SQL 질의

- <여기서 잠깐> **MINUS, INTERSECT** 연산자
 - MySQL에는 MINUS, INTERSECT 연산자가 없음
 - 다음과 같이 표현한다.

[질의 3-32]에서 **MINUS** 연산을 수행한 “대한민국에서 거주하는 고객의 이름에서 도서를 주문한 고객의 이름 빼고 보이시오.” 질의를 NOT IN 연산자를 사용하면 다음과 같다.

```
SELECT    name
FROM      Customer
WHERE address LIKE '대한민국%' AND
        name NOT IN (SELECT name
                      FROM      Customer
                      WHERE      custid IN (SELECT custid FROM Orders));
```

[질의 3-32]에서 **INTERSECT** 연산을 수행한 “대한민국에서 거주하는 고객 중 도서를 주문한 고객의 이름 보이시오.” 질의를 IN 연산자를 사용하면 다음과 같다.

```
SELECT    name
FROM      Customer
WHERE address LIKE '대한민국%' AND
        name IN (SELECT    name
                   FROM      Customer
                   WHERE      custid IN (SELECT custid FROM Orders));
```

두 개 이상 테이블에서 SQL 질의

❖ EXISTS_주문이 있는 고객을 알고 싶다

- EXISTS는 원래 단어에서 의미하는 것과 같이 조건에 맞는 튜플이 존재하면 결과에 포함시킴. 즉 부속질의문의 어떤 행이 조건에 만족하면 참임.
- NOT EXISTS는 부속질의문의 모든 행이 조건에 만족하지 않을 때만 참임.

질의 3-33 주문이 있는 고객의 이름과 주소를 보이시오.

```
SELECT  name, address
FROM    Customer cs
WHERE   EXISTS (SELECT *
                FROM Orders od
                WHERE cs.custid =od.custid);
```

name	address
박지성	영국 맨체스타
김연아	대한민국 서울
장미란	대한민국 강원도
추신수	미국 클리블랜드

두 개 이상 테이블에서 SQL 질의

❖ EXISTS_주문이 있는 고객을 알고 싶다

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

Orders

orderid	custid	bookid	saleprice	orderdate	
1	1	1	6000	2014-07-01	true
2	1	3	21000	2014-07-03	
3	2	5	8000	2014-07-03	true
4	3	6	6000	2014-07-04	true
5	4	7	20000	2014-07-05	true
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	

①
②
③
④
⑤
+
없음
false

그림 3-25 EXISTS 상관 부속질의문 데이터 예

```

SELECT    name, address
FROM      Customer cs
WHERE     EXISTS (SELECT *
                  FROM Orders od
                  WHERE cs.custid = od.custid);
  
```