

### 03. SQL 소개

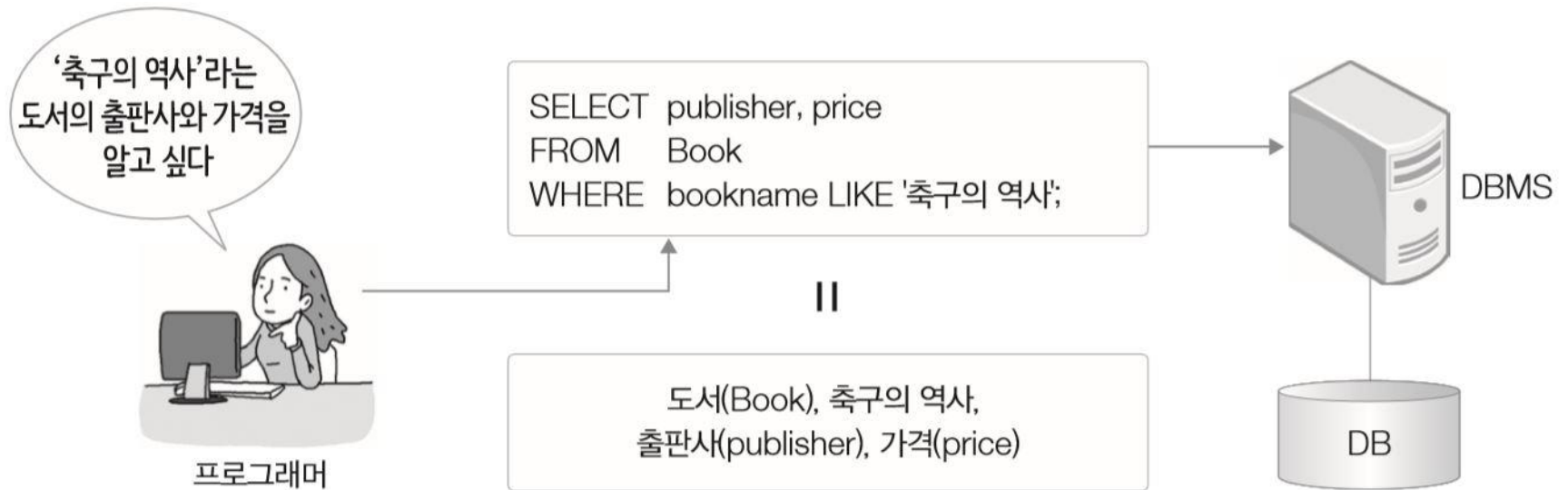


그림 3-8 SQL을 사용해 자료를 찾는 과정

## 03. SQL 소개

표 3-2 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
문법	SELECT * FROM Book;	int main() { ... }

## 03. SQL 소개

### ■ SQL 기능에 따른 분류

- 데이터 정의어(DDL)
  - 테이블이나 관계의 구조를 생성하는 데 사용
  - CREATE, ALTER, DROP 문 등이 있음.
- 데이터 조작어(DML)
  - 테이블에 데이터를 검색, 삽입, 수정, 삭제하는 데 사용
  - SELECT, INSERT, DELETE, UPDATE 문 등이 있음.
    - SELECT 문은 특별히 질의어(query)라고 함.
- 데이터 제어어(DCL)
  - 데이터의 사용 권한을 관리하는 데 사용
  - GRANT, REVOKE 문 등이 있음.

### 03. SQL 소개

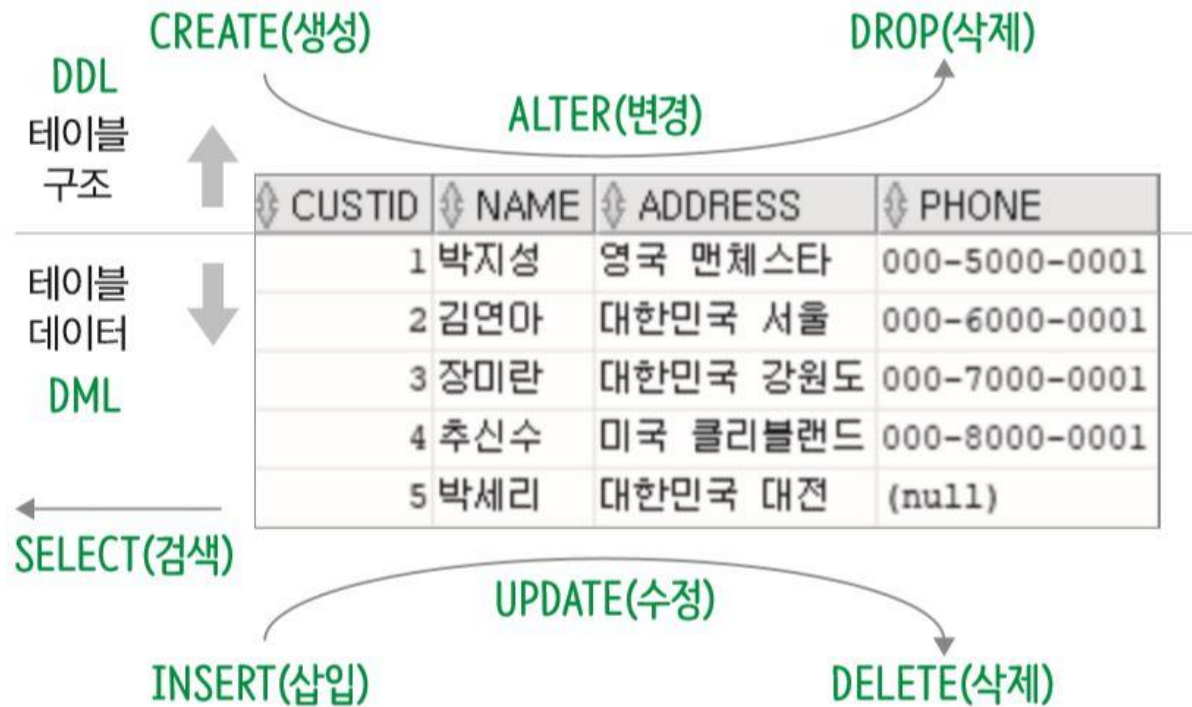


그림 3-9 데이터 정의어와 데이터 조작어의 주요 명령어

## 03. SQL 소개

예) 김연아 고객의 전화번호를 찾으시오.

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

SELECT: 질의 결과 추출되는 속성 리스트를 열거한다.

FROM: 질의에 어느 테이블이 사용되는지 열거한다.

WHERE: 질의의 조건을 작성한다.

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 3-10 SQL 문의 내부적 실행 순서

## 04. 데이터 조작용어 - 검색

---

- SELECT 문
- 집계 함수와 GROUP BY
- 두 개 이상 테이블에서 SQL 질의

## 04. 데이터 조작용어 - 검색

### ■ SELECT 문의 구성 요소

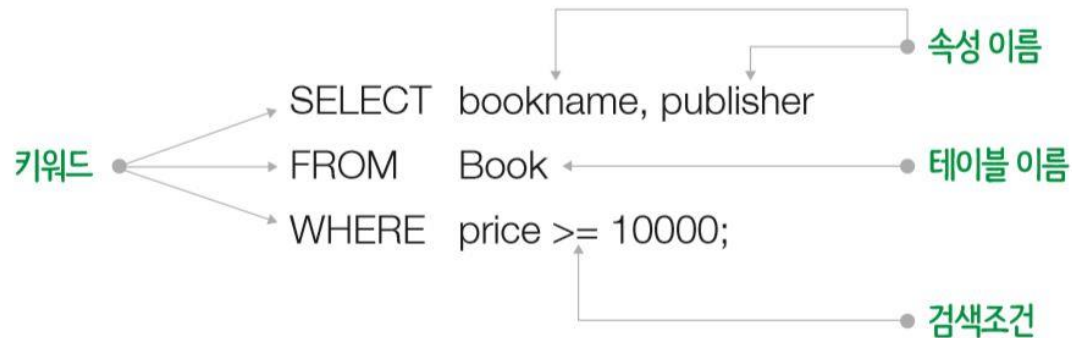


그림 3-11 SELECT 문의 예

### ■ 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.

## 04. 데이터 조작용어 - 검색

### ■ SELECT 문의 기본 문법

```
SELECT
    [ALL | DISTINCT]
    [테이블이름.] { * | 속성이름 [[ AS ] 속성이름별칭] }
[FROM
    {테이블이름 [AS 테이블이름별칭]}

    [INNER JOIN | LEFT [OUTER] JOIN | RIGHT [OUTER] JOIN
    {테이블이름 [ON 검색조건]}

    |FULL [OUTER] JOIN {테이블이름}}]
[WHERE 검색조건(들)]
[GROUP BY {속성이름, [..., n]}]
[HAVING 검색조건(들)]
[질의 UNION 질의 | 질의 UNION ALL 질의]
[ORDER BY {속성이름 [ASC | DESC], [..., n]}]
```

---

[ ]: 대괄호 안의 SQL 예약어들은 선택적으로 사용한다.

{ }: 중괄호 안의 SQL 예약어들은 필수적으로 사용한다.

|: 선택 가능한 문법 중 한 개를 사용할 수 있다.



# SELECT 문 예제

질의 3-1 모든 도서의 이름과 가격을 검색하시오

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

BOOKNAME	PRICE
축구의 역사	7000
축구마는 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 이름과 가격을 검색하시오

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

PRICE	BOOKNAME
7000	축구의 역사
13000	축구마는 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 예제

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

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

```
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 예제

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 DISTINCT 키워드를 사용

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

PUBLISHER
굿스포츠
삼성당
대한미디어
Pearson
나무수
이상미디어

# 조건 검색\_WHERE

표 3-3 WHERE 절에 조건으로 사용할 수 있는 술어

술어	연산자	사용 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')

## ■ 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	축구마는 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

# 조건 검색\_WHERE

## ■ 비교

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	축구하는 여자	나무수	13000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
10	Olympic Champions	Pearson	13000

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

BOOKID	BOOKNAME	PUBLISHER	PRICE
2	축구하는 여자	나무수	13000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
10	Olympic Champions	Pearson	13000

# 조건 검색\_WHERE

## ■ 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 검색

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

BOOKID	BOOKNAME	PUBLISHER	PRICE
2	축구마는 여자	나무수	13000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

# 조건 검색\_WHERE

## ■ 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

# 조건 검색\_WHERE

## ■ 패턴

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	축구마는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000

표 3-4 와일드 문자의 종류

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



# 조건 검색\_WHERE

## ■ 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

# 검색 결과의 정렬\_ORDER BY

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

# 검색 결과의 정렬\_ORDER BY

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

# 검색 결과의 정렬\_ORDER BY

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

```
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

# 집계 함수

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 부여

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

총매출
118000

# 집계 함수

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	6000	21000

# 집계 함수

질의 3-18 마당서점의 도서 판매 건수를 구하시오.

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

COUNT(*)
10

표 3-5 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)

# GROUP BY 검색

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
4	2	33000
3	3	31000

ORDERID	CUSTID	BOOKID	SALEPRICE	ORDERDATE
2	1	3	21000	20/07/03
6	1	2	12000	20/07/07
1	1	1	6000	20/07/01
9	2	10	7000	20/07/09
3	2	5	8000	20/07/03
4	3	6	6000	20/07/04
10	3	8	13000	20/07/10
8	3	10	12000	20/07/08
7	4	8	13000	20/07/07
5	4	7	20000	20/07/05

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

그림 3-12 GROUP BY 절의 수행



# GROUP BY 검색

## ■ HAVING 절

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

# GROUP BY 검색

## ■ GROUP BY와 HAVING 절의 문법과 주의사항

표 3-6 GROUP BY와 HAVING 절의 문법과 주의사항

GROUP BY <속성>		
주의사항	GROUP BY로 투플을 그룹으로 묶은 후 SELECT 절에는 GROUP BY에서 사용한 <속성>과 집계 함수만 나올 수 있다.	
맞는 예	SELECT	custid, SUM(saleprice)
	FROM	Orders
	GROUP BY	custid;
틀린 예	SELECT	bookid, SUM(saleprice)
	/* SELECT 절에 bookid 속성이 올 수 없다 */	
	FROM	Orders
	GROUP BY	custid;
HAVING <검색조건>		
주의사항	WHERE 절과 HAVING 절이 같이 포함된 SQL 문은 검색조건이 모호해질 수 있다. HAVING 절은 ① 반드시 GROUP BY 절과 같이 작성해야 하고 ② WHERE 절보다 뒤에 나와야 한다. 그리고 ③ <검색조건>에는 SUM, AVG, MAX, MIN, COUNT와 같은 집계 함수가 와야 한다.	
맞는 예	SELECT	custid, COUNT(*) AS 도서수량
	FROM	Orders
	WHERE	saleprice >= 8000
	GROUP BY	custid
	HAVING	count(*) >= 2;
틀린 예	SELECT	custid, COUNT(*) AS 도서수량
	FROM	Orders
	HAVING	count(*) >= 2 /* 순서가 틀렸다 */
	WHERE	saleprice >= 8000
	GROUP BY	custid;

## 05. 데이터 조작용어 – 삽입, 수정, 삭제

---

- INSERT 문
- UPDATE 문
- DELETE 문

## 5.1 INSERT 문

- INSERT 문은 테이블에 새로운 튜플을 삽입하는 명령임.

- INSERT 문의 기본 문법

```
INSERT INTO 테이블이름[(속성리스트)]  
VALUES (값리스트);
```

질의 3-44 Book 테이블에 새로운 도서 '스포츠 의학'을 삽입하시오. 스포츠 의학은 한솔의학서적에서 출간했으며 가격은 90,000원이다.

```
INSERT INTO Book(bookid, bookname, publisher, price)  
VALUES (11, '스포츠 의학', '한솔의학서적', 90000);
```

BOOKID	BOOKNAME	PUBLISHER	PRICE
11	스포츠 의학	한솔의학서적	90000
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

## 5.1 INSERT 문

질의 3-45 Book 테이블에 새로운 도서 '스포츠 의학'을 삽입하시오. 스포츠 의학은 한솔의학서적에서 출간했으며 가격은 미정이다.

```
INSERT INTO Book(bookid, bookname, publisher)
VALUES (14, '스포츠 의학', '한솔의학서적');
```

BOOKID	BOOKNAME	PUBLISHER	PRICE
11	스포츠 의학	한솔의학서적	90000
14	스포츠 의학	한솔의학서적	(null)
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

## 5.1 INSERT 문

- 대량 삽입(bulk insert)이란 한꺼번에 여러 개의 튜플을 삽입하는 방법임.

질의 3-46 수입도서 목록(Imported\_book)을 Book 테이블에 모두 삽입하시오.

```
INSERT INTO Book(bookid, bookname, price, publisher)
SELECT    bookid, bookname, price, publisher
FROM      Imported_book;
```

BOOKID	BOOKNAME	PUBLISHER	PRICE
11	스포츠 의학	한슬의학서적	90000
14	스포츠 의학	한슬의학서적	(null)
21	Zen Golf	Pearson	12000
22	Soccer Skills	Human Kinetics	15000
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

## 5.2 UPDATE 문

- UPDATE 문은 특정 속성 값을 수정하는 명령
- UPDATE 문의 기본 문법

```
UPDATE   테이블이름  
SET      속성이름 1=값 1[, 속성이름 2=값 2, ...]  
[WHERE   <검색조건>];
```

## 5.2 UPDATE 문

질의 3-47 Customer 테이블에서 고객번호가 5인 고객의 주소를 '대한민국 부산'으로 변경하시오.

```
UPDATE Customer
SET address='대한민국 부산'
WHERE custid=5;
```

**TIP** 결과를 확인하기 위해서는 'SELECT \* FROM Customer;' 명령을 실행해야 한다.

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 부산	(null)

질의 3-48 Customer 테이블에서 박세리 고객의 주소를 김연아 고객의 주소로 변경하시오.

```
UPDATE Customer
SET address=(SELECT address
              FROM Customer
              WHERE name='김연아')
WHERE name='박세리';
```

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 서울	(null)



## 5.3 DELETE 문

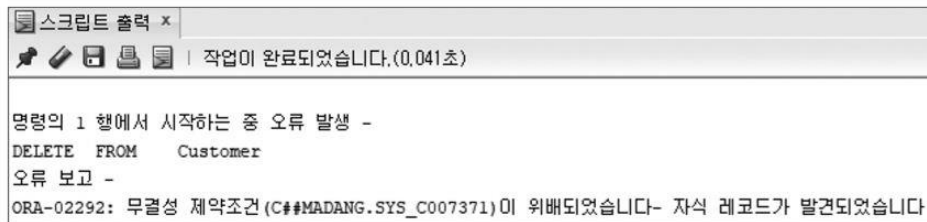
- DELETE 문은 테이블에 있는 기존 튜플을 삭제하는 명령
- DELETE 문의 기본 문법

```
DELETE FROM 테이블이름  
[WHERE 검색조건];
```

## 5.3 DELETE 문

질의 3-49 Customer 테이블에서 고객번호가 5인 고객을 삭제하시오.

```
DELETE FROM Customer;
```



질의 3-50 모든 고객을 삭제하시오.

```
DELETE FROM Customer  
WHERE custid=5;
```

```
SELECT * 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