

3장 SQL 기초(1) (Structured Query Language)

데이터베이스 연구실

[목 차]

3.1 마당 도서 판매점

3.1.1 마당 서점의 데이터 테이블

3.1.2 누가 어떤 데이터를 필요로 하는가?

3.2 SQL 기초

3.3 도서고객에게 필요한 질의

3.3.1 서점에 있는 책을 알고 싶다 - **SELECT/FROM**

3.3.2 책구에 관한 책들이 있는지 알고 싶다 - **WHERE** 조건

3.3.3 도서에 관한 내용을 좀 더 자세히 알고 싶다 - **WHERE** 조건, 비교, 논리

3.3.4 도서를 제목 순서대로 보자 - **ORDER BY**

3.4 도서 데이터베이스 운영자를 위한 질의

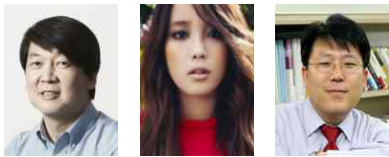
3.4.1 판매량을 집계하자 - **SQL** 집계함수 사용

3.4.2 어느 고객이 얼마나 구입했는지 알고 싶다 - **GROUP BY**

3.1 마당 도서 판매점

-마당서점 서점 모델

경영자 1명(안철수), 운영자 1명(아이유), 프로그래머 1명(박경철)



고객 5명(박지성, 김연아, 추신수, 박세리, 장미란)

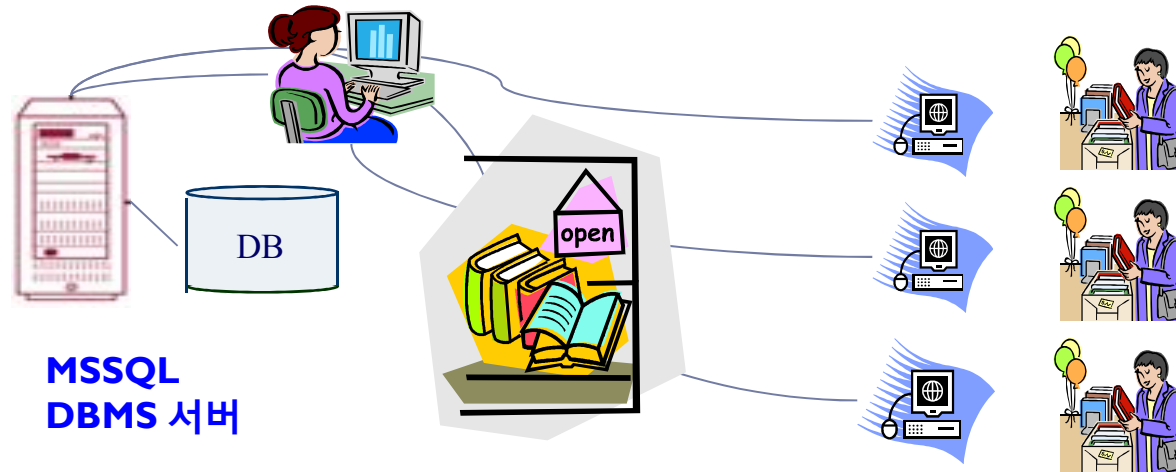


도서 10권



▶ 3 (주) 사진은 naver.com의 인물정보 검색에서 인용함.

-마당서점 데이터베이스 시스템 구조



* 운영자(점원 김병만)

- 주문의 합산
- 일별, 고객별 주문에 대한 내용 알고 싶다
- 고객별로 주문액을 알고싶다.
- 주문이 없는 사람을 알고싶다.

* 고객(박지성)의 요구사항

- 서점에 책의 제목들을 모두 알고 싶다.
- 축구에 관한 책이 있는지 알고 싶다.
- 내 주문 내역을 알고 싶다.

3.1.1 마당서점의 데이터 테이블

-마당서점 데이터

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

	custid	name	address	phone
1	1	박지성	영국 맨체스터	010-5000-0001
2	2	김연아	대한미국 서울	010-6000-0001
3	3	장미란	대한민국 강원도	010-7000-0001
4	4	추신수	미국 클리블랜드	010-8000-0001
5	5	박세리	대한민국 대전	010-9000-0001

	orderid	custid	bookid	saleprice	orderdate
1	1	1	1	6000	2011-07-01 00:00:00.000
2	2	1	3	21000	2011-07-02 00:00:00.000
3	3	2	5	7000	2011-07-03 00:00:00.000
4	4	3	6	6000	2011-07-04 00:00:00.000
5	5	4	7	19000	2011-07-05 00:00:00.000
6	6	1	3	21000	2011-07-06 00:00:00.000
7	7	4	8	12000	2011-07-07 00:00:00.000
8	8	3	9	7000	2011-07-08 00:00:00.000
9	9	2	9	7000	2011-07-09 00:00:00.000
10	10	3	9	7000	2011-07-10 00:00:00.000

3.1.2 누가 어떤 데이터를 필요로 하는가?

인터넷 서점에서 거래를 위하여 알고 싶은 내용

1. 고객(박지성)

- 서점에 책의 제목들을 모두 알고 싶다.
- 축구에 관한 책이 있는지 알고 싶다.
- 내 주문 내역을 알고 싶다.
- 중복의 제거

2. 운영자(점원 김병만)

- 주문의 합산
- 일별, 고객별 주문에 대한 내용 알고 싶다
- 고객별로 주문액을 알고싶다.
- 주문이 없는 사람을 알고싶다

3.2 SQL 배경

[표 II-1-1] SQL 문장들의 종류

명령어의 종류	명령어	설명
데이터 조작어 (DML: Data Manipulation Language)	SELECT	데이터베이스에 들어 있는 데이터를 조회하거나 검색하기 위한 명령어를 말하는 것으로 RETRIEVE 라고도 한다.
	INSERT UPDATE DELETE	데이터베이스의 테이블에 들어 있는 데이터에 변형을 가하는 종류의 명령어들을 말한다. 예를 들어 데이터를 테이블에 새로운 행을 집어넣거나, 원하지 않는 데이터를 삭제하거나 수정하는 것들의 명령어들을 DML이라고 부른다.
데이터 정의어 (DDL: Data Definition Language)	CREATE ALTER DROP RENAME	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어들로 그러한 구조를 생성하거나 변경하거나 삭제하거나 이름을 바꾸는 데이터 구조와 관련된 명령어들을 DDL이라고 부른다.
데이터 제어어 (DCL: Data Control Language)	GRANT REVOKE	데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어를 DCL이라고 부른다.
트랜잭션 제어어 (TCL: Transaction Control Language)	COMMIT ROLLBACK	논리적인 작업의 단위를 묶어서 DML에 의해 조작된 결과를 작업단위(트랜잭션) 별로 제어하는 명령어를 말한다.

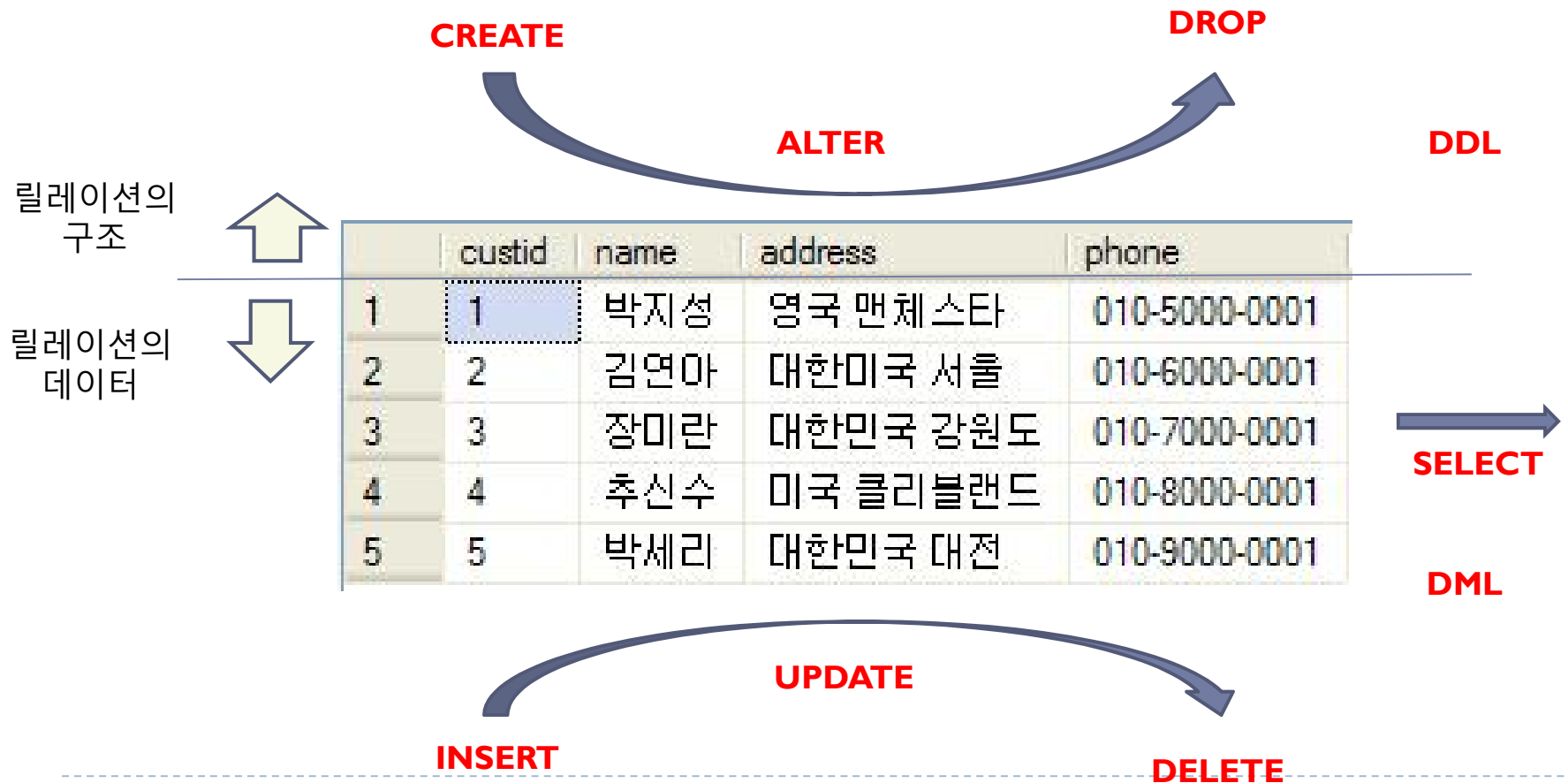
-SQL 문은 크게 2가지로 구분된다.

1. 데이터 정의 언어(DDL)로 테이블, 관계나 다른 구조를 생성하는 데 사용되는 문.

CREATE, ALTER, DROP 등

2. 데이터 조작 언어(DML)로 데이터에 대한 질의, 삽입, 수정, 그리고 삭제를 하는데 사용되는 문이다.

SELECT/FROM/WHERE, INSERT, DELETE, UPDATE



-예 : SQL문의 처리 순서

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

① FROM customer

custid	name	address	phone
1	박지성	영국 맨체스타	010-5000-0001
2	김연아	대한민국 서울	010-6000-0001
3	장미란	대한민국 강원도	010-7000-0001
4	추신수	미국 클리블랜드	010-8000-0001
5	박세리	대한민국 대전	010-9000-0001

② WHERE name='김연아';

custid	name	address	phone
2	김연아	대한민국 서울	010-6000-0001

③ SELECT phone

phone
010-6000-0001

3.3 도서고객에게 필요한 질의

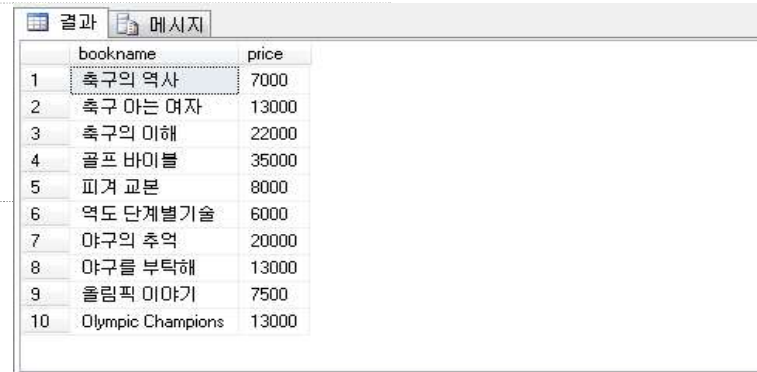
3.3.1 서점에 있는 책을 알고 싶다 - SELECT/FROM

SELECT 문

- 모든 책의 이름과 가격을 보여라.

```
SELECT bookname, price
```

```
FROM book;
```



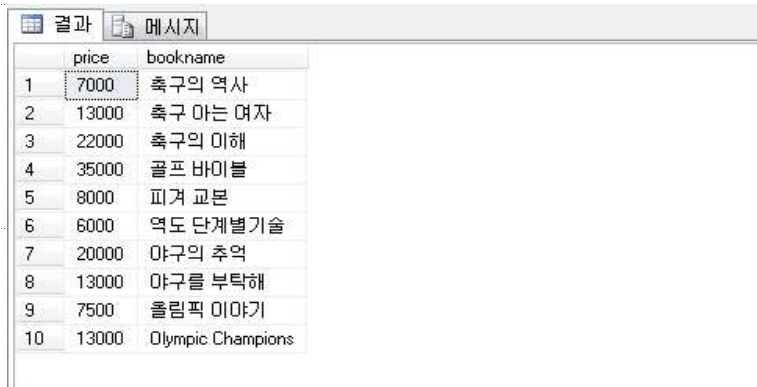
	bookname	price
1	축구의 역사	7000
2	축구 아는 여자	13000
3	축구의 이해	22000
4	골프 바이블	35000
5	피겨 교본	8000
6	역도 단계별기술	6000
7	야구의 추억	20000
8	야구를 부탁해	13000
9	올림픽 이야기	7500
10	Olympic Champions	13000

SELECT 문

- 모든 책의 가격과 이름을 보여라.

```
SELECT price, bookname
```

```
FROM book;
```



	price	bookname
1	7000	축구의 역사
2	13000	축구 아는 여자
3	22000	축구의 이해
4	35000	골프 바이블
5	8000	피겨 교본
6	6000	역도 단계별기술
7	20000	야구의 추억
8	13000	야구를 부탁해
9	7500	올림픽 이야기
10	13000	Olympic Champions

SELECT 문

- 모든 책의 번호, 이름, 출판사, 가격을 보여라.

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



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

SELECT 문

- 모든 책의 번호, 이름, 출판사, 가격을 보여라

```
SELECT *  
FROM book;
```

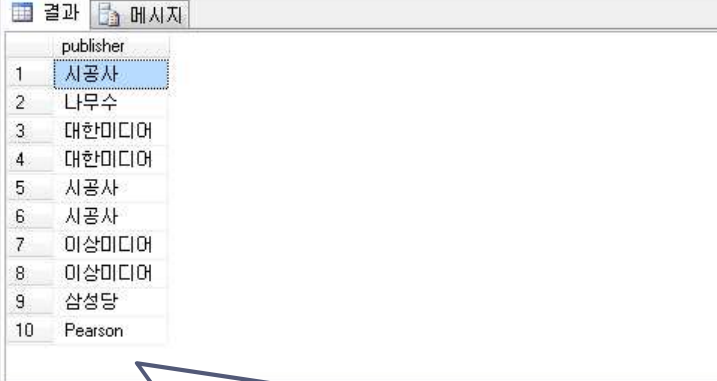


	bookid	bookname	publisher	price
1	1	축구의 역사	시공사	7000
2	2	축구 마는 여자	나무수	13000
3	3	축구의 이해	대한미디어	22000
4	4	골프 바이블	대한미디어	35000
5	5	피겨 교본	시공사	8000
6	6	역도 단계별기술	시공사	6000
7	7	야구의 추억	이상미디어	20000
8	8	야구를 부탁해	이상미디어	13000
9	9	올림픽 이야기	삼성당	7500
10	10	Olympic Champions	Pearson	13000

SELECT 문

- 출판사를 모두 보여라.

```
SELECT publisher  
FROM book;
```



A screenshot of a database query result window titled '결과' (Results) and '메시지' (Messages). The window displays a table with the following data:

	publisher
1	시공사
2	나무수
3	대한미디어
4	대한미디어
5	시공사
6	시공사
7	이상미디어
8	이상미디어
9	삼성당
10	Pearson

SQL 문은 관계대수와 달리 기본적으로 중복을 제거하지 않는다.

SELECT 문

- 출판사를 모두 보여라. 단, 중복은 제거하여라

```
SELECT DISTINCT publisher  
FROM book;
```



A screenshot of a database query result window titled '결과' (Results) and '메시지' (Messages). The window displays a table with the following data:

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

3.3.2 축구에 관한 책들이 있는지 알고 싶다 - WHERE 조건

SELECT 문

- 가격이 20,000원 미만인 책을 보여라.

```
SELECT *
```

```
FROM book
```

```
WHERE price < 20000;
```



	bookid	bookname	publisher	price
1	1	축구의 역사	시공사	7000
2	2	축구 마는 여자	나무수	13000
3	5	피겨 교본	시공사	8000
4	6	역도 단계별기술	시공사	6000
5	8	야구를 부탁해	이상미디어	13000
6	9	올림픽 이야기	삼성당	7500
7	10	Olympic Champions	Pearson	13000

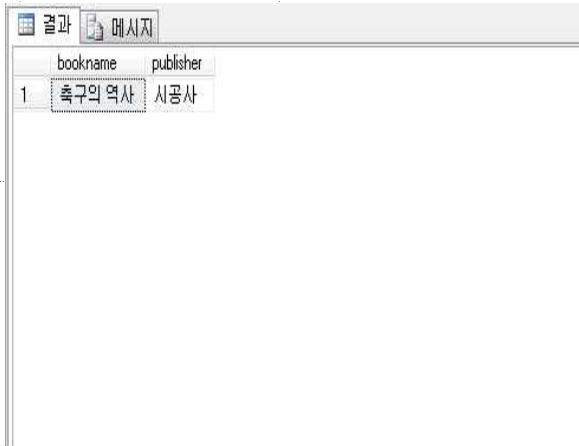
SELECT 문

- 이름이 '축구의 역사'인 책의 이름과 출판사를 보여라.

```
SELECT bookname, publisher
```

```
FROM book
```

```
WHERE bookname LIKE '축구의 역사';
```



	bookname	publisher
1	축구의 역사	시공사

SELECT 문

- '축구'가 포함된 책 제목, 출판사를 보여라.

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

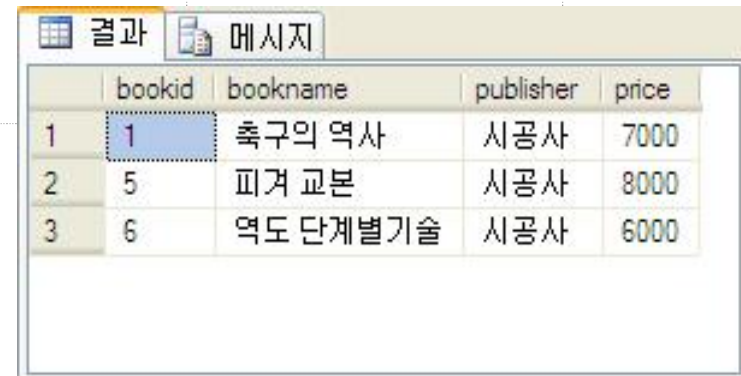


	bookname	publisher
1	축구의 역사	시공사
2	축구 아는 여자	나무수
3	축구의 이해	대한미디어

SELECT 문

-오른쪽 2번째 위치에 '공'이라는 문자를 갖는 출판사를 보여라.

```
SELECT *
FROM BOOK
WHERE publisher like '%공_';
```



	bookid	bookname	publisher	price
1	1	축구의 역사	시공사	7000
2	5	피겨 교본	시공사	8000
3	6	역도 단계별기술	시공사	6000

3.3.3 도서에 관한 내용을 좀 더 자세히 알고 싶다 - WHERE 조건, 비교, 논리

-복합 WHERE 절

SELECT 문

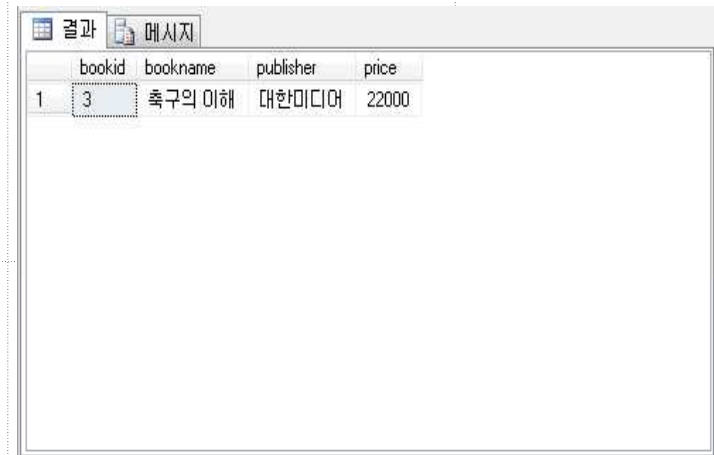
- '축구'에 관한 책 중 가격이 20,000원 이상인 책을 보여라.

```
SELECT *
```

```
FROM BOOK
```

```
WHERE bookname like '%축구%'
```

```
AND price >= 20000;
```



The screenshot shows a window titled '결과' (Result) with a sub-tab '메시지' (Message). It displays a table with the following data:

bookid	bookname	publisher	price	
1	3	축구의 이해	대한미디어	22000

SELECT 문

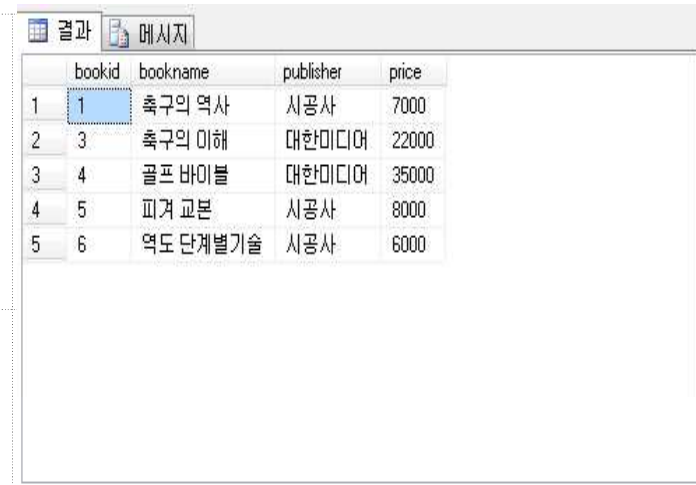
- 출판사가 '시공사' 혹은 '대한미디어'인 책을 보여라.

```
SELECT *
```

```
FROM BOOK
```

```
WHERE publisher = '시공사'
```

```
OR publisher = '대한미디어';
```



The screenshot shows a window titled '결과' (Result) with a '메시지' (Message) icon. It displays a table with the following data:

	bookid	bookname	publisher	price
1	1	축구의 역사	시공사	7000
2	3	축구의 이해	대한미디어	22000
3	4	골프 바이블	대한미디어	35000
4	5	피겨 교본	시공사	8000
5	6	역도 단계별기술	시공사	6000

-IN

SELECT 문

- 출판사가 '시공사' 혹은 '대한미디어'인 책을 보여라.

```
SELECT *
```

```
FROM BOOK
```

```
WHERE publisher IN ('시공사', '대한미디어');
```



	bookid	bookname	publisher	price
1	1	축구의 역사	시공사	7000
2	3	축구의 이해	대한미디어	22000
3	4	골프 바이블	대한미디어	35000
4	5	피겨 교본	시공사	8000
5	6	역도 단계별기술	시공사	6000

-NOT IN


SELECT 문

- 출판사가 '시공사' 혹은 '대한미디어'가 아닌 책을 보여라.

```
SELECT *
```

```
FROM BOOK
```

```
WHERE publisher NOT IN ('시공사', '대한미디어');
```



	bookid	bookname	publisher	price
1	2	축구 마는 여자	나무수	13000
2	7	야구의 추억	이상미디어	20000
3	8	야구를 부탁해	이상미디어	13000
4	9	올림픽 이야기	삼성당	7500
5	10	Olympic Champions	Pearson	13000

- BETWEEN

SELECT 문

- 가격이 10000 이상, 20000 이하인 책을 보여라.

```
SELECT *
```

```
FROM BOOK
```

```
WHERE price BETWEEN 10000 AND 20000;
```



The screenshot shows a window titled '결과' (Results) with a '메시지' (Message) icon. It displays a table with the following data:

	bookid	bookname	publisher	price
1	2	축구 마는 여자	나무수	13000
2	7	야구의 추억	이상미디어	20000
3	8	야구를 부탁해	이상미디어	13000
4	10	Olympic Champions	Pearson	13000

3.3.4 도서를 제목 순서대로 보자 - ORDER BY

SELECT 문

- 책을 제목 순으로 보여라.

```
SELECT *
```

```
FROM BOOK
```

```
ORDER BY bookname;
```

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

SELECT 문

- 책을 가격 순, 제목 순으로 보여라.

```
SELECT *
```

```
FROM BOOK
```

```
ORDER BY price, bookname;
```

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

SELECT 문

- 책을 가격의 내림차순, 출판사의 오름차순으로 보

SELECT *

FROM book

ORDER BY price DESC, publisher ASC;



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

3.4 도서 데이터베이스 운영자를 위한 질의

3.4.1 판매량을 집계하자 - SQL 집계함수 사용

SELECT 문

- 고객이 주문한 도서의 총 판매액을 보여라.

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

	(열 이름 없음)
1	119000

SELECT 문

- 고객이 주문한 도서의 총 판매액을 보여라.

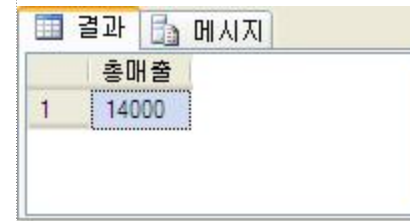
```
SELECT SUM(saleprice) as 총매출
FROM ORDERS;
```

	총매출
1	119000

AS 키워드를 사용하여 열 이름을 부여할 수 있다.

SELECT 문

- 2번 고객이 주문한 도서의 총 판매액을 보여라.



총매출	
1	14000

```
SELECT SUM(saleprice) as 총매출
FROM ORDERS
WHERE custid=2;
```

SELECT 문

- 도서의 총 판매액, 평균값, 최저가, 최고가를 보여라.



	Total	Average	Minimum	Maximum
1	119000	11900	6000	21000

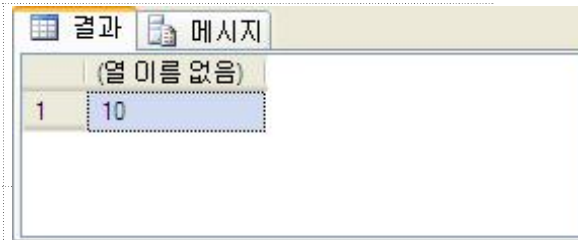
```
SELECT SUM(saleprice) as Total,
AVG(saleprice) as Average,
MIN(saleprice) as Minimum,
MAX(saleprice) as Maximum
```

▶ FROM ORDERS;

SELECT 문

- 판매 건수를 보여라.

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



The screenshot shows a window titled '결과' (Results) with a sub-tab '메시지' (Message). The window displays the result of a SQL query. The text '(열 이름 없음)' (No column names) is visible above the data. The data is presented in a table with one row containing the value '10'.

(열 이름 없음)	
1	10

3.4.2 어느 고객이 얼마나 구입했는지 알고 싶다 - GROUP BY

SELECT 문

- 고객별로 도서수량과 총액을 구하라?

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

	결과	메시지
	custid	도서수량
1	1	3
2	2	2
3	3	3
4	4	2
	총액	
		48000
		14000
		26000
		31000

orderid	custid	bookid	saleprice	orderdate
1	1	1	6000	2012-07-01
2	1	3	21000	2012-07-02
6	1	3	21000	2012-07-06
9	2	9	7000	2012-07-09
3	2	5	7000	2012-07-03
4	3	6	6000	2012-07-04
10	3	8	13000	2012-07-10
8	3	9	7000	2012-07-08
7	4	8	12000	2012-07-07
5	4	7	19000	2012-07-05

custid	도서수량	총액
1	3	48000
2	2	14000
3	3	26000
4	2	31000

SELECT 문

- 10000원 초과 도서 판매를 고객별로 도서수량을
구하여라.

```
SELECT    custid, COUNT(*) AS 도서수량
FROM      ORDERS
WHERE     saleprice > 10000
GROUP BY  custid
ORDER BY  custid;
```



The screenshot shows a window titled '결과' (Results) with a '메시지' (Message) icon. It displays a table with two columns: 'custid' and '도서수량' (Book Quantity). The table contains three rows of data:

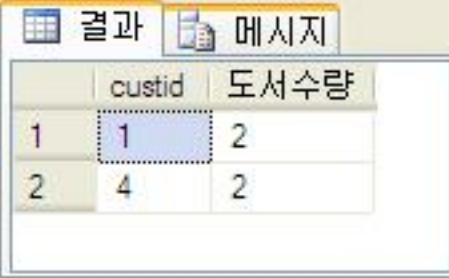
	custid	도서수량
1	1	2
2	3	1
3	4	2

-HAVING

SELECT 문

- 10000원 초과 도서 판매를 고객별로 도서수량을 구하여라. 단, 2권 이상 구매한 고객만 구하라.

```
SELECT    custid, COUNT(*) AS 도서수량
FROM      ORDERS
WHERE     saleprice > 10000
GROUP BY  custid
HAVING    count(*) >= 2
ORDER BY  custid;
```



The screenshot shows a window titled '결과' (Results) with a '메시지' (Messages) tab. The main area displays a table with two columns: 'custid' and '도서수량' (Book Count). The table contains two rows of data.

	custid	도서수량
1	1	2
2	4	2

- ⑤ SELECT custid, COUNT(*) AS 도서수량
- ① FROM ORDERS
- ② WHERE saleprice > 10000
- ③ GROUP BY custid
- ④ HAVING count(*) > 1
- ⑥ ORDER BY custid;

① FROM ORDERS

	orderid	custid	bookid	saleprice	orderdate
1	1	1	1	6000	2012-07-01
2	2	1	3	21000	2012-07-02
3	3	2	5	7000	2012-07-03
4	4	3	6	6000	2012-07-04
5	5	4	7	19000	2012-07-05
6	6	1	3	21000	2012-07-06
7	7	4	8	12000	2012-07-07
8	8	3	9	7000	2012-07-08
9	9	2	9	7000	2012-07-09
10	10	3	8	13000	2012-07-10

② WHERE saleprice > 10000

	orderid	custid	bookid	saleprice	orderdate
1	2	1	3	21000	2012-07-02
2	5	4	7	19000	2012-07-05
3	6	1	3	21000	2012-07-06
4	7	4	8	12000	2012-07-07
5	10	3	8	13000	2012-07-10

③ GROUP BY custid

	custid	(열 이름 없음)
1	1	2
2	3	1
3	4	2

④ HAVING count(*) > 1

	custid	(열 이름 없음)
1	1	2
2	4	2

⑤ SELECT custid, COUNT(*) AS 도서수량

	custid	도서수량
1	1	2
2	4	2

⑥ ORDER BY custid;

	custid	도서수량
1	1	2
2	4	2