

Database

◆ SQL-1

정수아

Contents

01 SQL이란?

02 데이터베이스 생성하기

03 테이블 생성하기

04 데이터 삽입하기

05 데이터 조회하기



01

SQL이란?

SQL이란?

❖ SQL(Structured Query Language)

- 관계형 데이터베이스 관리 시스템의 데이터를 관리하기 위한 언어
- 1970년대 IBM에서 최초 개발
- DBMS에 따른 SQL 명칭

DBMS	SQL 명칭
MySQL	SQL
SQL Server	Transact-SQL(T-SQL)
Oracle	PL/SQL

SQL이란?

❖ SQL 특징

- 대부분의 DBMS 제품에서 호환 가능
- 서버용, 개인용, 휴대용 장비에서 운영되는 DBMS마다 상호 호환성이 뛰어남
- 모든 DBMS의 SQL문이 완벽하게 동일하지는 않음
 - 현실적으로 완전한 통일은 어려움

SQL이란?

❖ SQL 문법 종류

종류	항목
데이터 정의어(DDL)	CREATE, ALTER, DROP 등
데이터 조작어(DML)	SELECT, INSERT, DELETE 등
데이터 제어어(DCL)	GRANT, REVOKE, COMMIT 등

자료형

❖ SQL에서 주로 사용하는 자료형

자료형	설명
INT	정수 데이터를 저장할 때 사용
CHAR(n)	고정형 문자열(0~255자)
VARCHAR(n)	가변형 문자열(1~65535자)
DATE	날짜를 표현하는 타입 1000-01-01 ~ 9999-12-31
DATETIME	날짜와 시간을 같이 표현하는 타입 1970-01-01 00:00:00 ~ 2037-01-19 03:14:07



02

데이터베이스 생성하기

데이터베이스 생성하기

❖ 데이터베이스 생성

```
CREATE DATABASE 데이터베이스이름;
```

❖ 데이터베이스 사용

```
USE 데이터베이스이름;
```

[실습] 데이터베이스 생성하기

❖ 'TestDB' 데이터베이스 생성

```
CREATE DATABASE TestDB;
```

❖ 'TestDB' 데이터베이스 사용

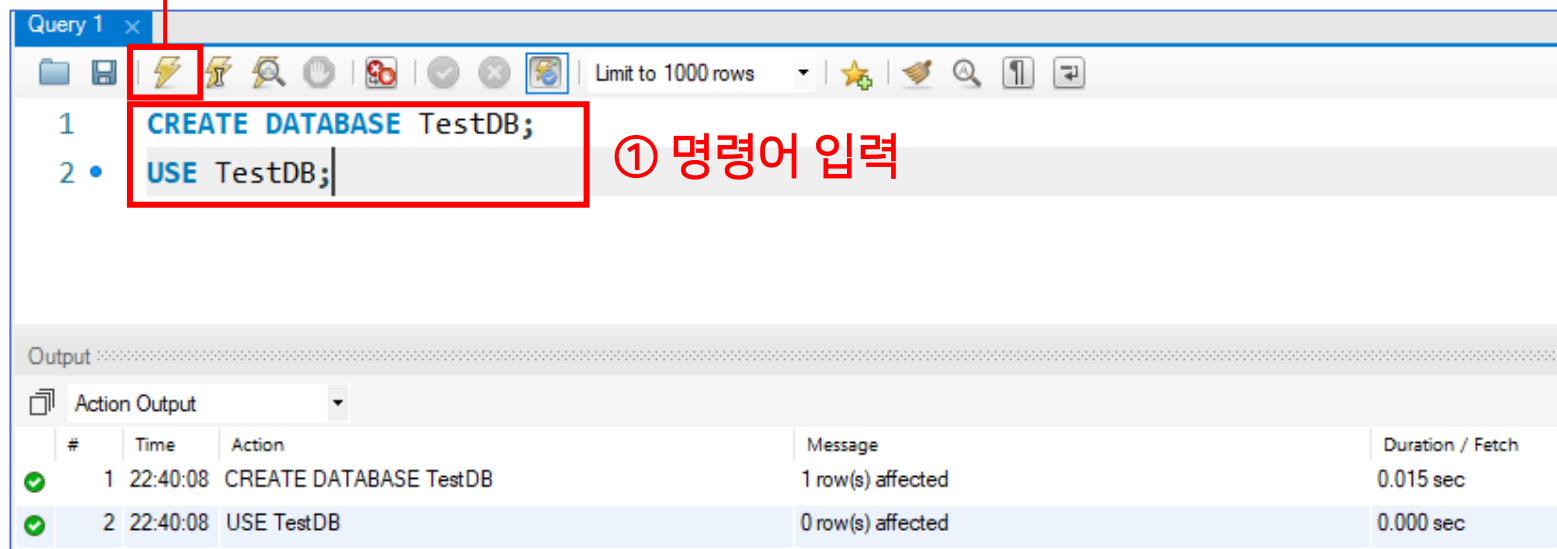
```
USE TestDB;
```

[실습] 데이터베이스 생성하기

❖ SQL 편집기에 명령어 입력 후 실행

② 명령어 실행

① 명령어 입력



Query 1 x

Limit to 1000 rows

```
1 CREATE DATABASE TestDB;
2 • USE TestDB;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	22:40:08	CREATE DATABASE TestDB	1 row(s) affected	0.015 sec
✓ 2	22:40:08	USE TestDB	0 row(s) affected	0.000 sec

데이터베이스 삭제하기

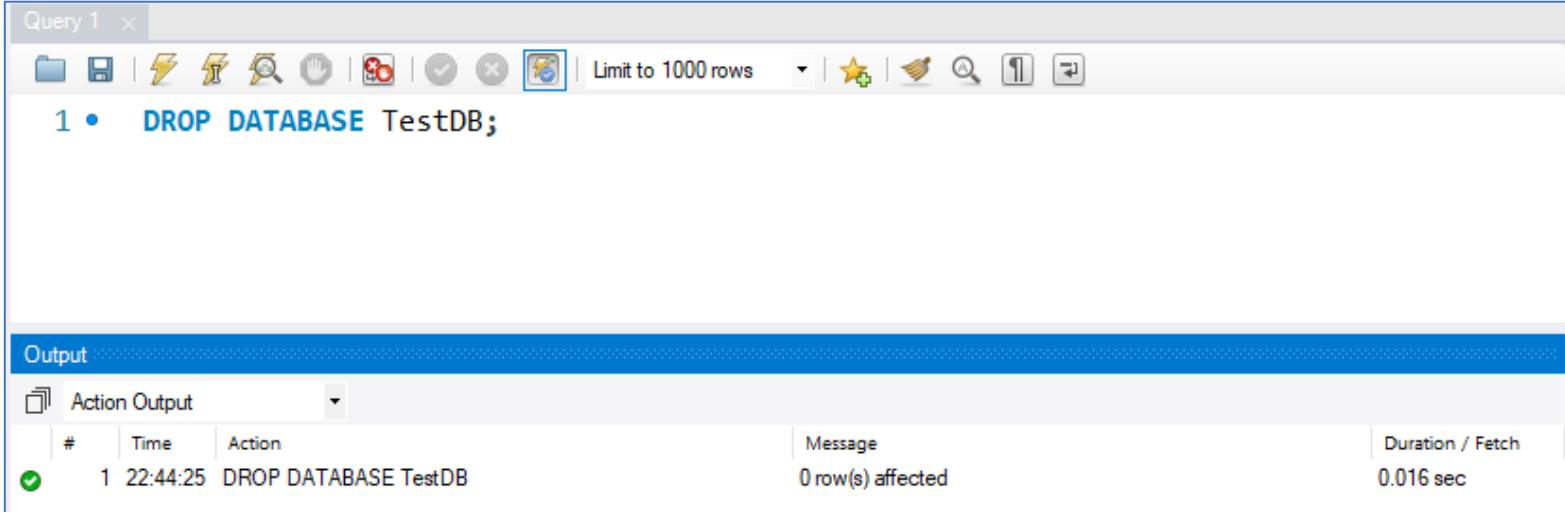
❖ 데이터베이스 삭제

```
DROP DATABASE 데이터베이스이름;
```

[실습] 데이터베이스 삭제하기

❖ 'TestDB' 데이터베이스 삭제

```
DROP DATABASE TestDB;
```



The screenshot shows a database query tool interface. The top toolbar includes icons for file operations, query execution, and viewing. The query editor displays the command: `1 • DROP DATABASE TestDB;`. The 'Output' tab is selected, showing the 'Action Output' table below.

#	Time	Action	Message	Duration / Fetch
1	22:44:25	DROP DATABASE TestDB	0 row(s) affected	0.016 sec

[실습] BookDB 데이터베이스 생성하기

❖ 'BookDB' 데이터베이스 생성

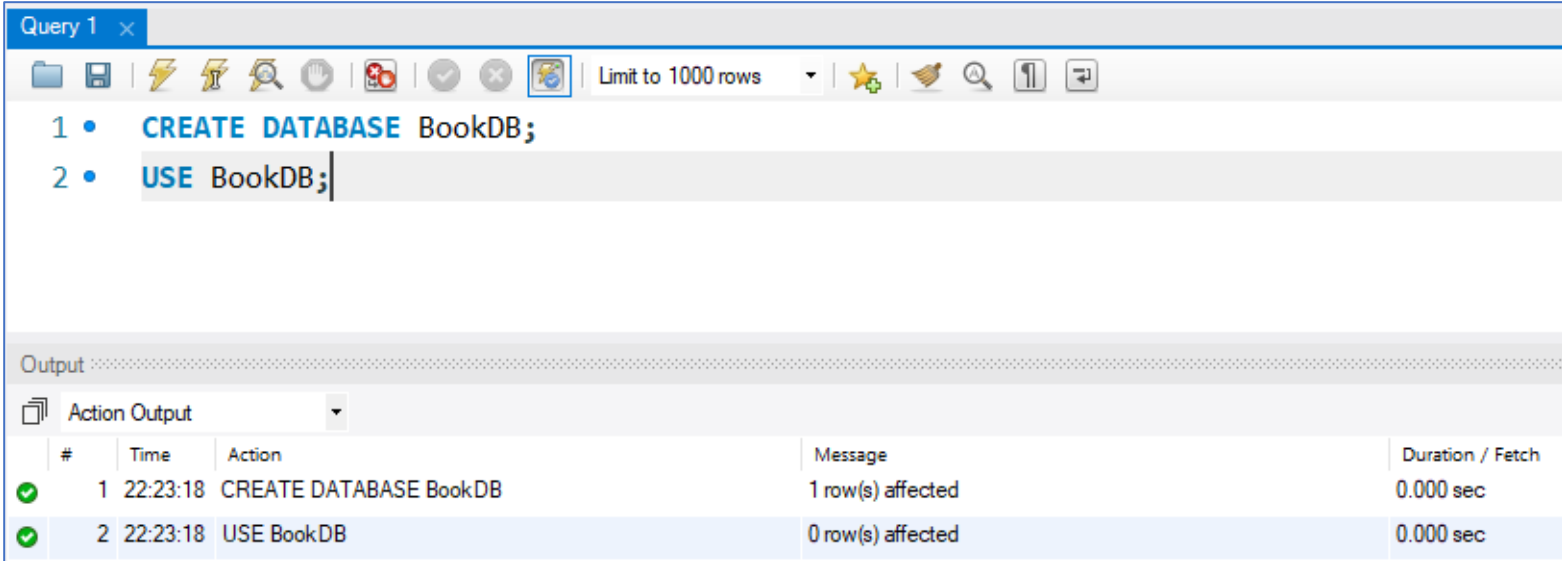
```
CREATE DATABASE BookDB;
```

❖ 'BookDB' 데이터베이스 사용

```
USE BookDB;
```

[실습] BookDB 데이터베이스 생성하기

❖ SQL 편집기에 명령어 입력 후 실행



The screenshot displays a SQL editor window titled "Query 1". The editor contains two SQL statements:

- 1 • **CREATE DATABASE** BookDB;
- 2 • **USE** BookDB;

Below the editor, the "Output" section shows the execution results in a table format:

#	Time	Action	Message	Duration / Fetch
✓ 1	22:23:18	CREATE DATABASE BookDB	1 row(s) affected	0.000 sec
✓ 2	22:23:18	USE BookDB	0 row(s) affected	0.000 sec



03

테이블 생성하기

테이블 생성하기

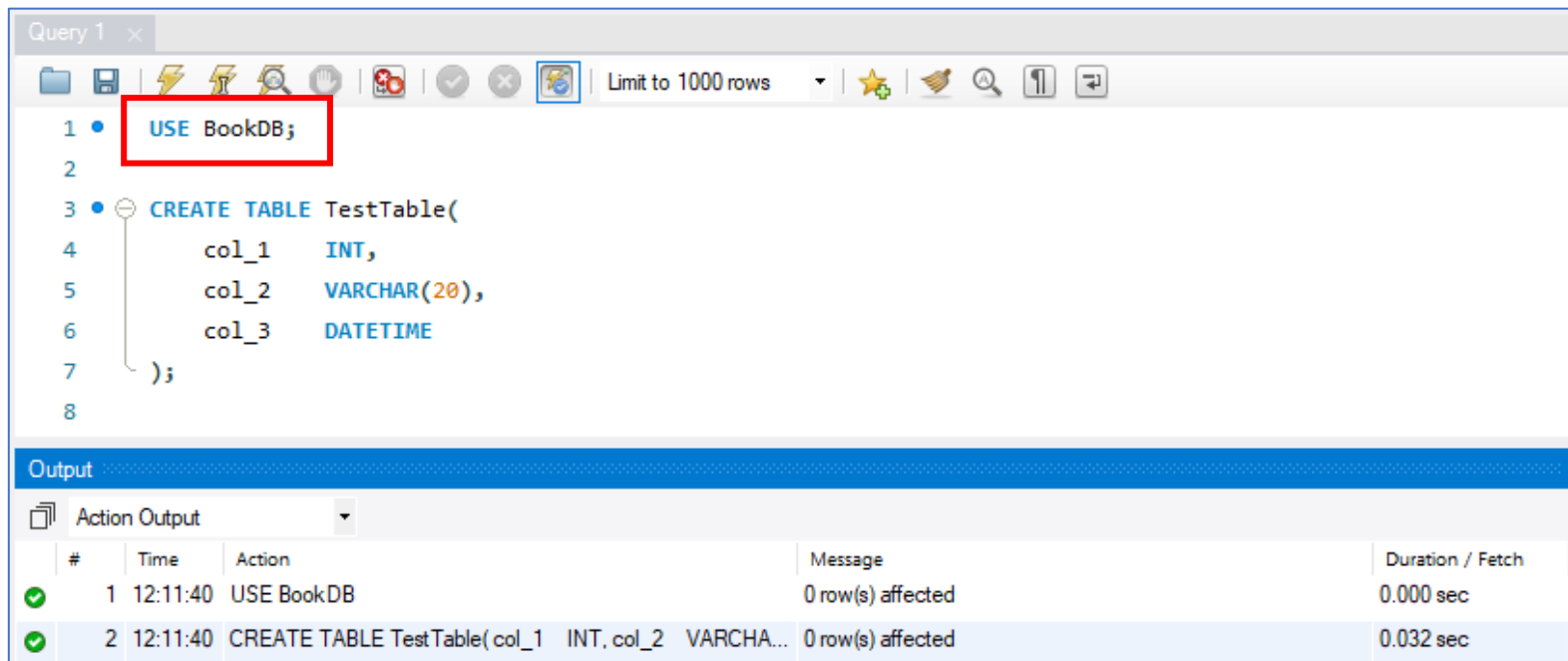
❖ 테이블 생성

```
CREATE TABLE 테이블이름(  
    속성이름1 데이터타입,  
    속성이름2 데이터타입,  
    (...생략...)  
);
```

[실습] 테이블 생성하기

❖ 'TestTable' 테이블 생성

```
CREATE TABLE TestTable(  
    col_1 INT,  
    col_2 VARCHAR(20),  
    col_3 DATETIME  
);
```



Query 1 x

Limit to 1000 rows

```
1 • USE BookDB;  
2  
3 • CREATE TABLE TestTable(  
4     col_1 INT,  
5     col_2 VARCHAR(20),  
6     col_3 DATETIME  
7 );  
8
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	12:11:40	USE BookDB	0 row(s) affected	0.000 sec
✓ 2	12:11:40	CREATE TABLE TestTable(col_1 INT, col_2 VARCHAR...	0 row(s) affected	0.032 sec

테이블 삭제하기

❖ 테이블 삭제

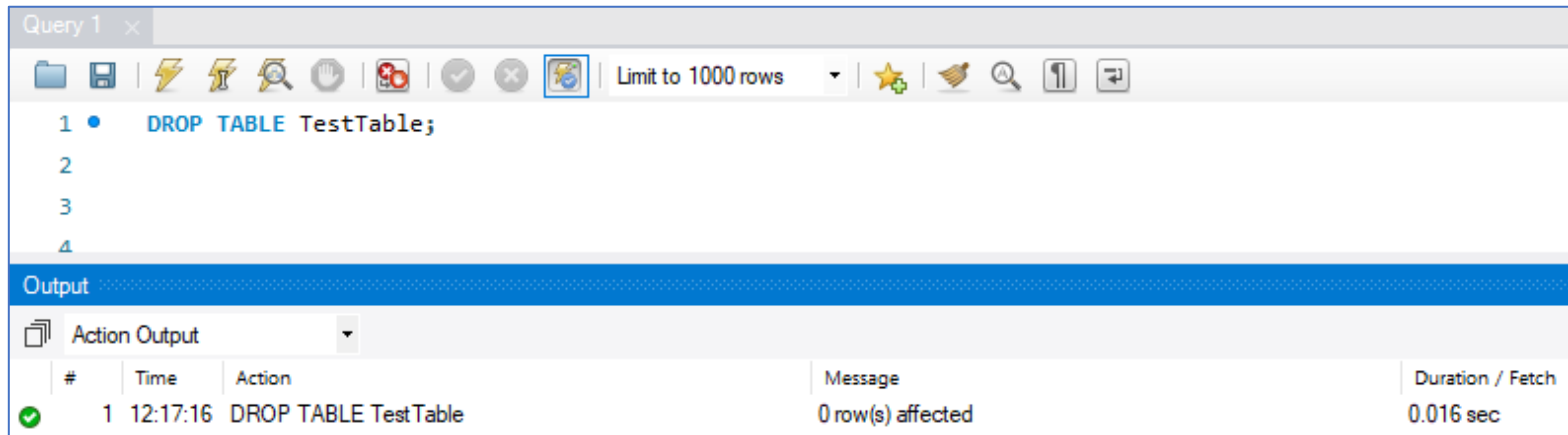
```
DROP TABLE 테이블이름;
```

- 주의할 점
 - 삭제하려는 테이블이 다른 테이블과 종속 관계이면서 부모 테이블인 경우 삭제 불가능
 - 종속 관계를 제거한 뒤 자식 테이블을 모두 삭제해야 삭제 가능

[실습] 테이블 삭제하기

❖ 'TestTable' 테이블 삭제

```
DROP TABLE TestTable;
```



The screenshot shows a SQL query editor window titled 'Query 1'. The query text is 'DROP TABLE TestTable;'. Below the query editor, the 'Output' pane is visible, showing the execution results. The output table has columns: #, Time, Action, Message, and Duration / Fetch. The first row shows a successful execution at 12:17:16, with the message '0 row(s) affected' and a duration of 0.016 sec.

#	Time	Action	Message	Duration / Fetch
1	12:17:16	DROP TABLE TestTable	0 row(s) affected	0.016 sec

[실습] Book 테이블 생성하기

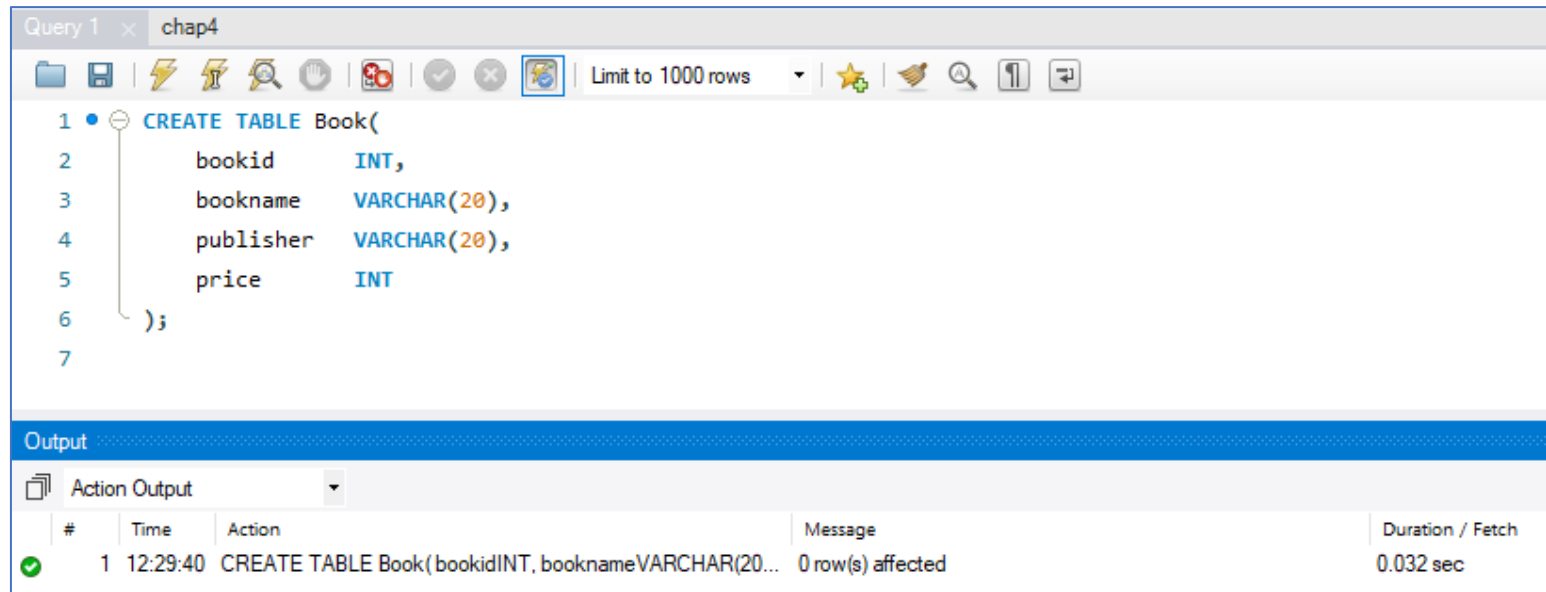
❖ 'Book' 테이블 생성

속성 이름	데이터 타입	설명
bookid	INT	정수
bookname	VARCHAR(20)	최대 20자 문자
publisher	VARCHAR(20)	최대 20자 문자
price	INT	정수

```
CREATE TABLE Book(  
    bookid      INT,  
    bookname    VARCHAR(20),  
    publisher    VARCHAR(20),  
    price       INT  
);
```

[실습] Book 테이블 생성하기

❖ 'Book' 테이블 생성 결과



The screenshot displays a database query editor window titled 'Query 1' and 'chap4'. The query text is as follows:

```
1 CREATE TABLE Book(  
2     bookid      INT,  
3     bookname    VARCHAR(20),  
4     publisher   VARCHAR(20),  
5     price       INT  
6 );  
7
```

Below the query editor, the 'Output' pane shows the 'Action Output' for the executed query:

#	Time	Action	Message	Duration / Fetch
1	12:29:40	CREATE TABLE Book(bookidINT, booknameVARCHAR(20...	0 row(s) affected	0.032 sec



04

데이터 삽입하기

데이터 삽입하기

❖ 테이블에 데이터 삽입

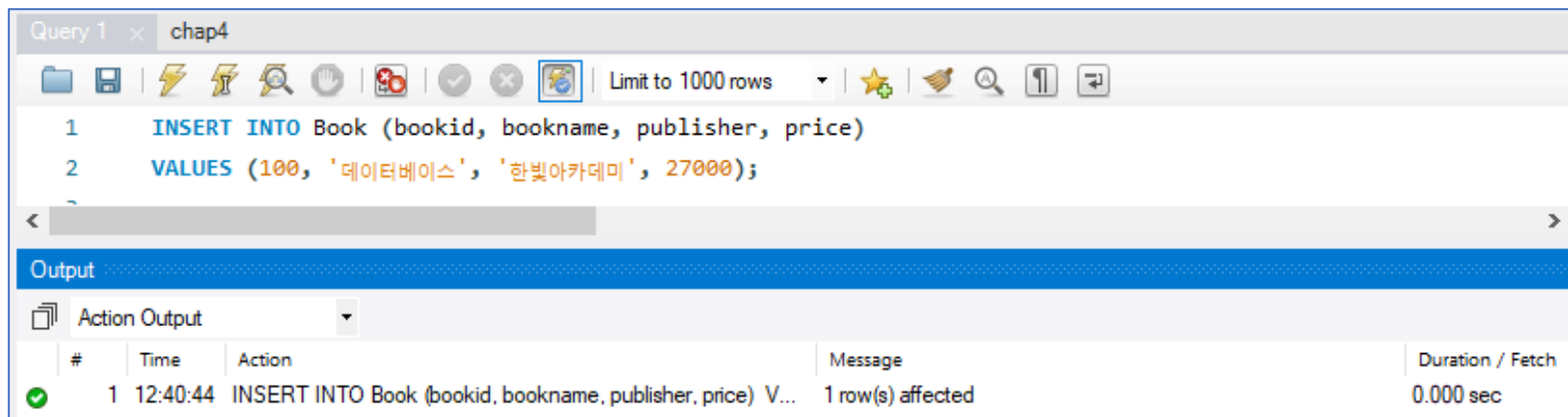
```
INSERT INTO 테이블이름 (속성1, 속성2, ...) VALUES (값1, 값2, ...);
```


[실습] 데이터 삽입하기

❖ 'Book' 테이블에 데이터 삽입

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000

```
INSERT INTO Book (bookid, bookname, publisher, price)
VALUES (100, '데이터베이스', '한빛아카데미', 27000);
```



[실습] 데이터 삽입하기

❖ 'Book' 테이블에 데이터 삽입

bookid	bookname	publisher	price
101	파이썬	한빛미디어	22000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배움터	생능출판사	30000

[실습] 데이터 삽입하기

❖ 'Book' 테이블에 데이터 삽입

```
INSERT INTO Book (bookid, bookname, publisher, price)
VALUES (101, '파이썬', '한빛아카데미', 22000);
```

```
INSERT INTO Book (bookid, bookname, publisher, price)
VALUES (102, 'JSP 프로그래밍', '생능출판사', 26000);
```

```
INSERT INTO Book (bookid, bookname, publisher, price)
VALUES (103, '자바스크립트', '길벗', 45000);
```

```
INSERT INTO Book (bookid, bookname, publisher, price)
VALUES (104, '데이터베이스 배우터', '생능출판사', 30000);
```



05

데이터 조회하기

데이터 조회하기

❖ 테이블에 저장된 전체 데이터 조회

```
SELECT * FROM 테이블이름
```

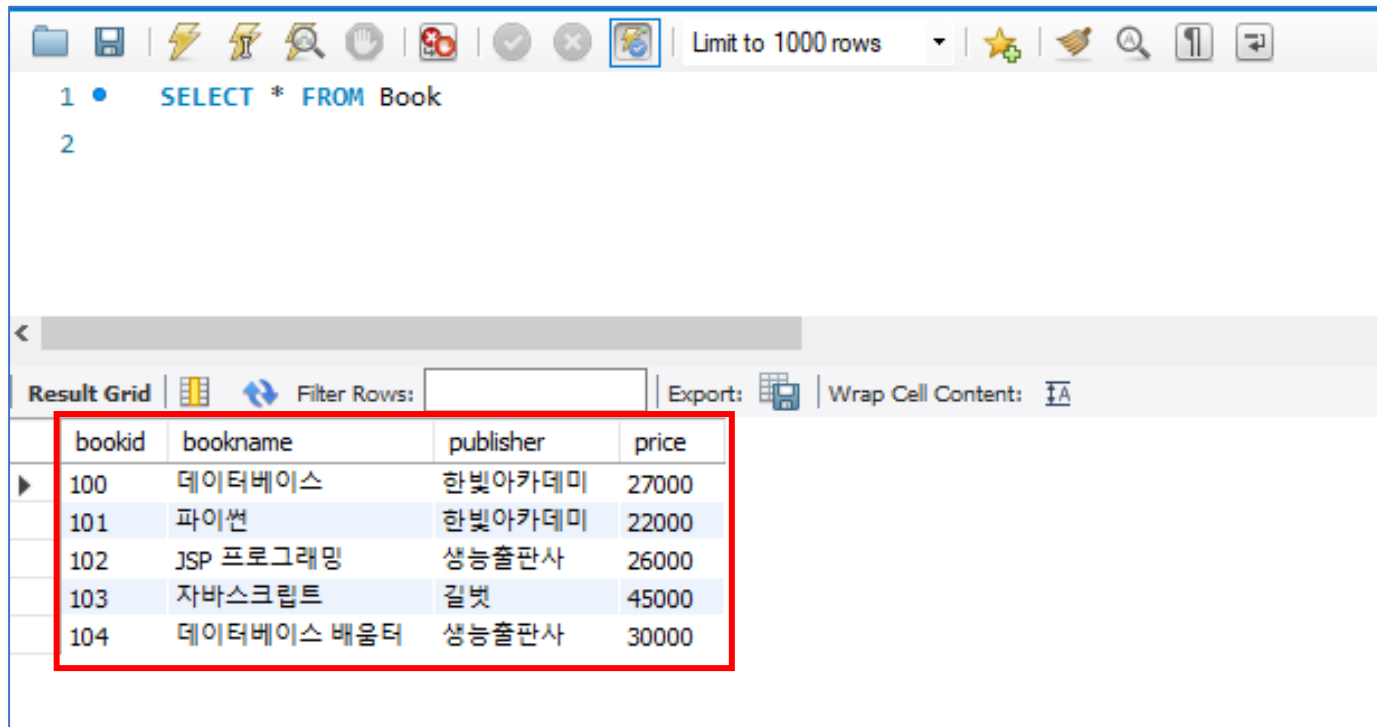
❖ 테이블에 저장된 특정 속성이름만 조회

```
SELECT 속성이름 FROM 테이블이름
```

[실습] 데이터 조회하기

❖ 'Book' 테이블에 저장된 전체 데이터 조회

```
SELECT * FROM Book
```



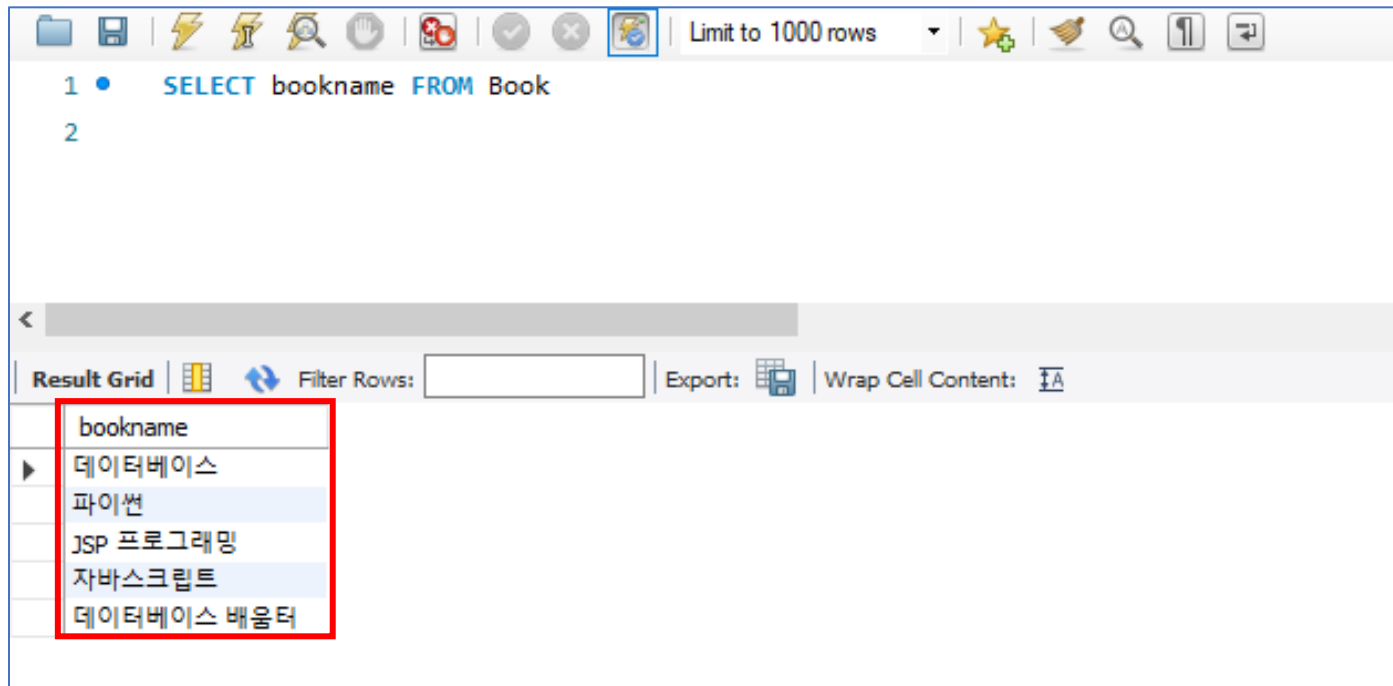
The screenshot shows a database query interface. At the top, a toolbar contains various icons for file operations, execution, and navigation. Below the toolbar, the SQL query 'SELECT * FROM Book' is entered in a text area. The interface then displays a 'Result Grid' containing the following data:

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	22000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배우터	생능출판사	30000

[실습] 데이터 조회하기

❖ 'Book' 테이블에 저장된 bookname만 조회

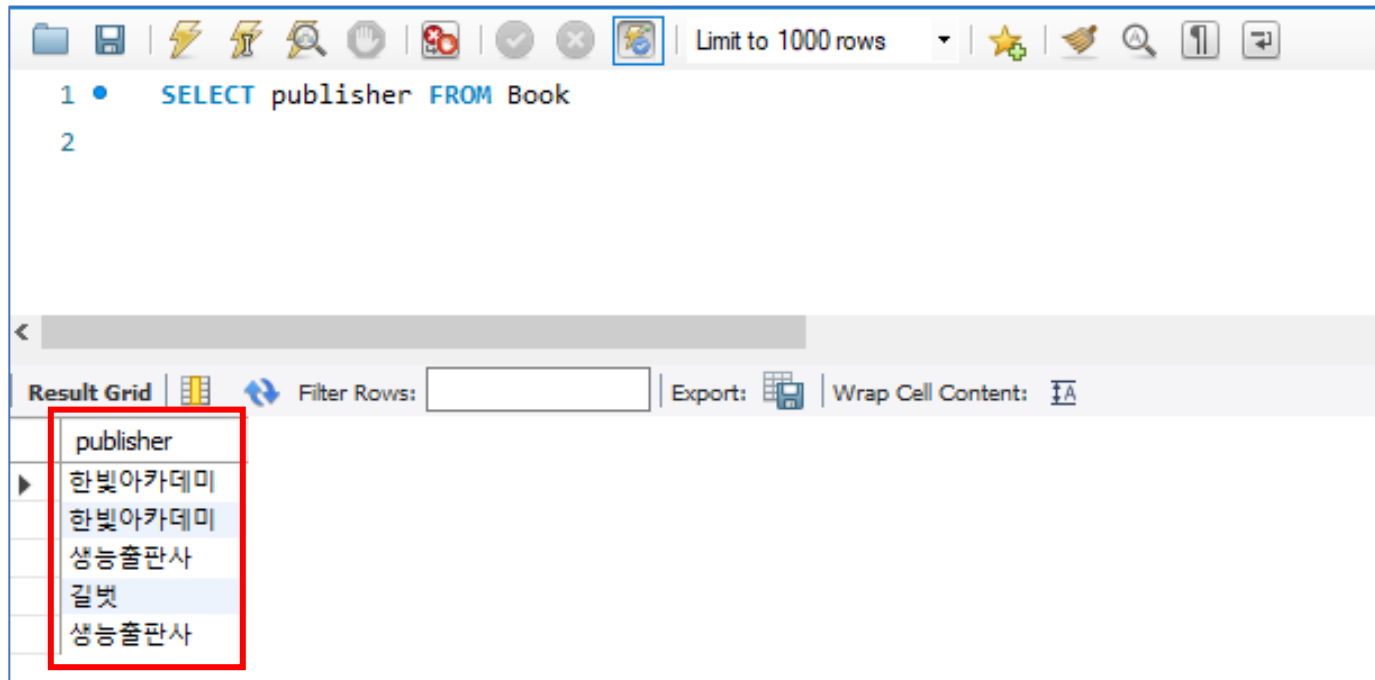
```
SELECT bookname FROM Book
```



[실습] 데이터 조회하기

❖ 'Book' 테이블에 저장된 publisher만 조회

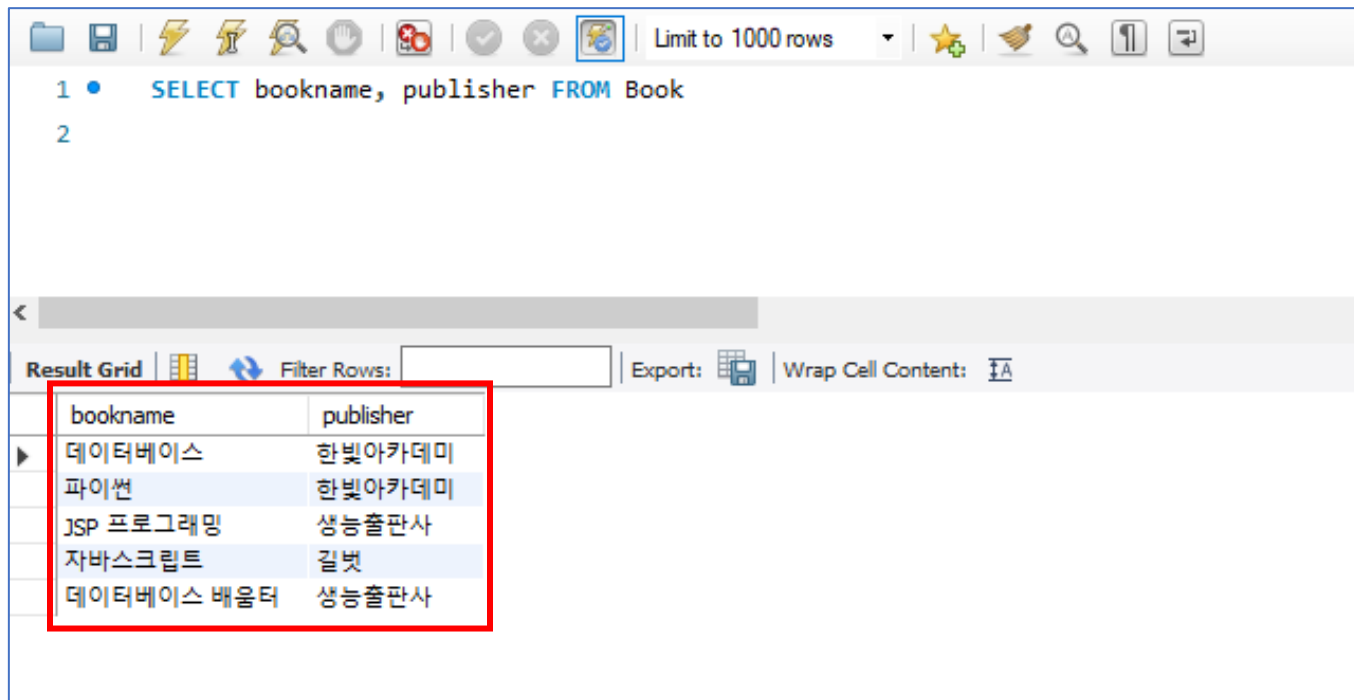
```
SELECT publisher FROM Book
```



[실습] 데이터 조회하기

❖ 'Book' 테이블에 저장된 bookname, publisher를 조회

```
SELECT bookname, publisher FROM Book
```



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query is entered in a text area:

```
1 • SELECT bookname, publisher FROM Book
2
```

Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The data grid itself is displayed below these controls. The first two columns are labeled "bookname" and "publisher". The data rows are:

bookname	publisher
데이터베이스	한빛아카데미
파이썬	한빛아카데미
JSP 프로그래밍	생능출판사
자바스크립트	길벗
데이터베이스 배우터	생능출판사

The data grid is highlighted with a red border.

Database

◆ SQL-2

정수아

Contents

01 조건으로 데이터 검색하기

02 데이터 정렬하기



01

조건으로 데이터 검색하기

조건으로 데이터 검색하기

❖ WHERE : 조건에 맞는 검색을 할 때 사용

```
SELECT 속성이름 FROM 테이블이름 WHERE 검색조건;
```

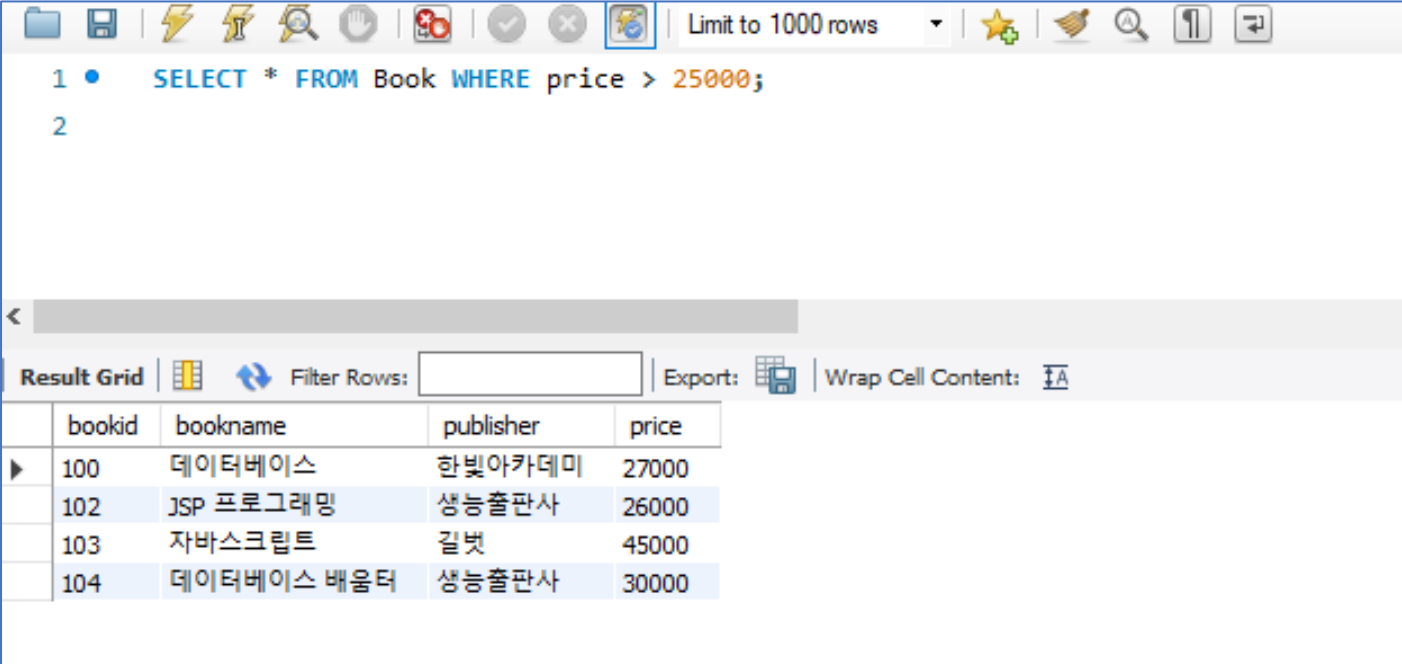
- WHERE 절의 조건

조건	연산자
비교	=, <, <=, >, >=
범위	BETWEEN
집합	IN, NOT IN
패턴	LIKE
NULL	IS NULL, IS NOT NULL
복합조건	AND, OR, NOT

[실습] 비교 연산자-1

❖ 가격이 25,000원 이상인 책 정보 검색

```
SELECT * FROM Book  
WHERE price > 25000;
```



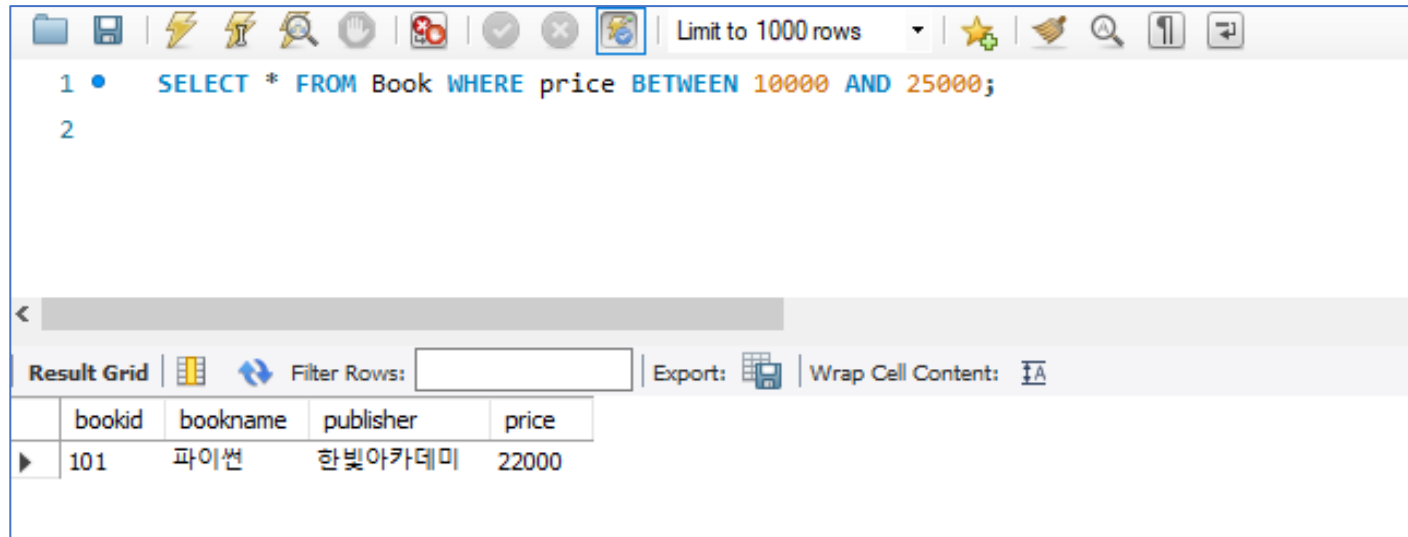
The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL query is entered in a text area: `1 • SELECT * FROM Book WHERE price > 25000;` and `2`. Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid itself is a table with the following data:

	bookid	bookname	publisher	price
▶	100	데이터베이스	한빛아카데미	27000
	102	JSP 프로그래밍	생능출판사	26000
	103	자바스크립트	길벗	45000
	104	데이터베이스 배우터	생능출판사	30000

[실습] 비교 연산자-2

❖ 가격이 10,000원 이상 25,000원 이하인 책 정보 검색

```
SELECT * FROM Book  
WHERE price BETWEEN 10000 AND 25000;
```



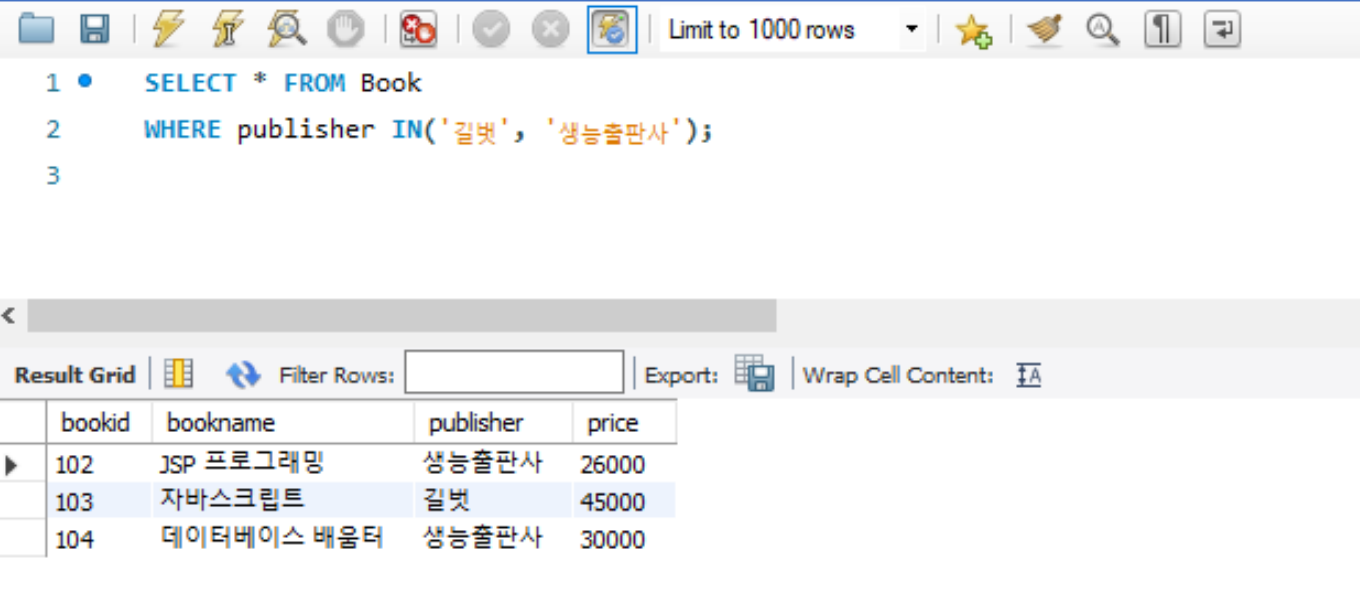
The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons for file operations, execution, and viewing. Below the toolbar, the SQL query is entered in a text area: `1 • SELECT * FROM Book WHERE price BETWEEN 10000 AND 25000;` and `2`. Below the query area, there is a horizontal scrollbar. At the bottom, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid itself is a table with the following data:

	bookid	bookname	publisher	price
▶	101	파이썬	한빛아카데미	22000

[실습] 집합-1

❖ 출판사가 '길벗' 또는 '생능출판사'인 책 정보 검색

```
SELECT * FROM Book  
WHERE publisher IN('길벗', '생능출판사');
```



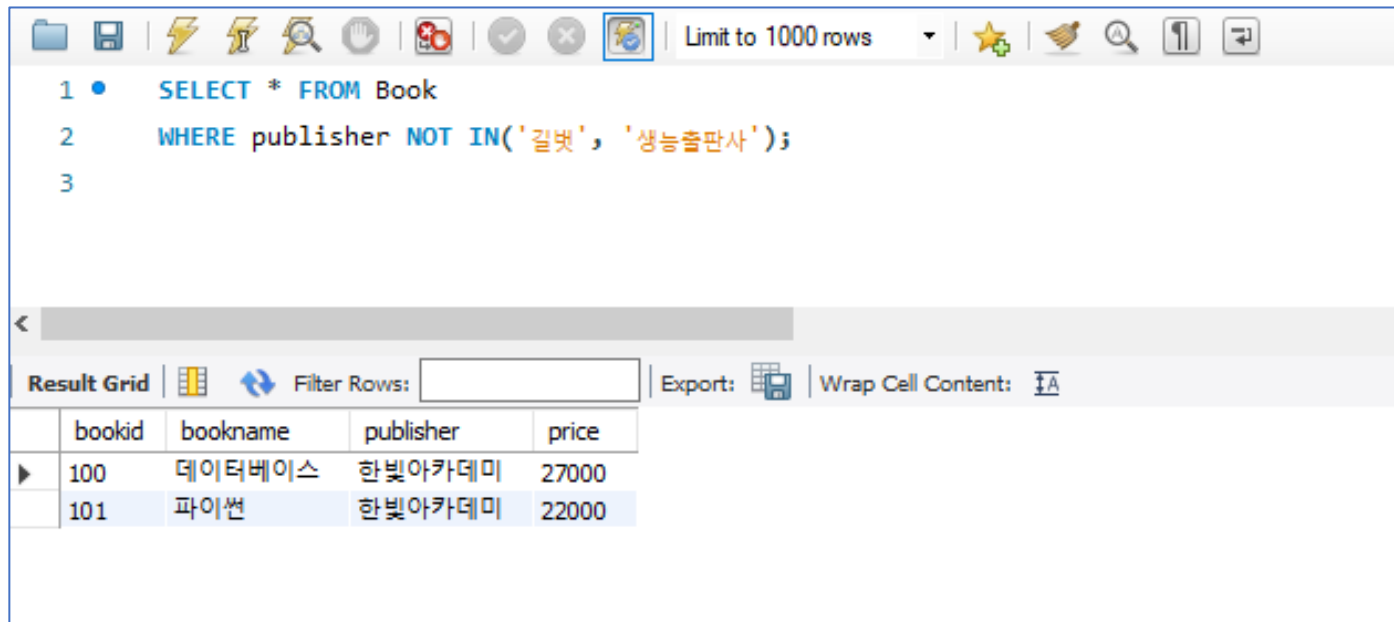
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The SQL editor contains the query: `SELECT * FROM Book WHERE publisher IN('길벗', '생능출판사');`. Below the editor, the 'Result Grid' tab is active, displaying the query results in a table. The table has four columns: bookid, bookname, publisher, and price. Three rows are visible, corresponding to the books returned by the query.

	bookid	bookname	publisher	price
▶	102	JSP 프로그래밍	생능출판사	26000
	103	자바스크립트	길벗	45000
	104	데이터베이스 배우터	생능출판사	30000

[실습] 집합-2

❖ 출판사가 '길벗' 또는 '생능출판사'가 아닌 책 정보 검색

```
SELECT * FROM Book  
WHERE publisher NOT IN('길벗', '생능출판사');
```



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT * FROM Book  
2 WHERE publisher NOT IN('길벗', '생능출판사');  
3
```

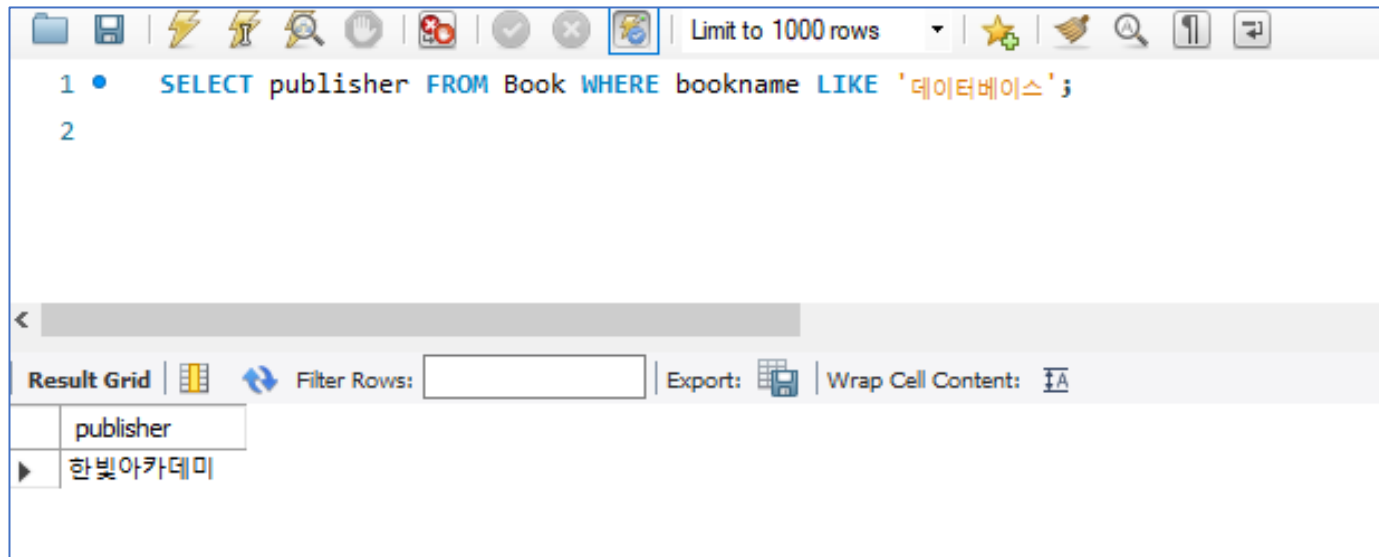
Below the query, the 'Result Grid' displays the results of the query. The grid has columns for bookid, bookname, publisher, and price. The results are as follows:

	bookid	bookname	publisher	price
▶	100	데이터베이스	한빛아카데미	27000
	101	파이썬	한빛아카데미	22000

[실습] 패턴-1

❖ '데이터베이스'를 출간한 출판사 검색

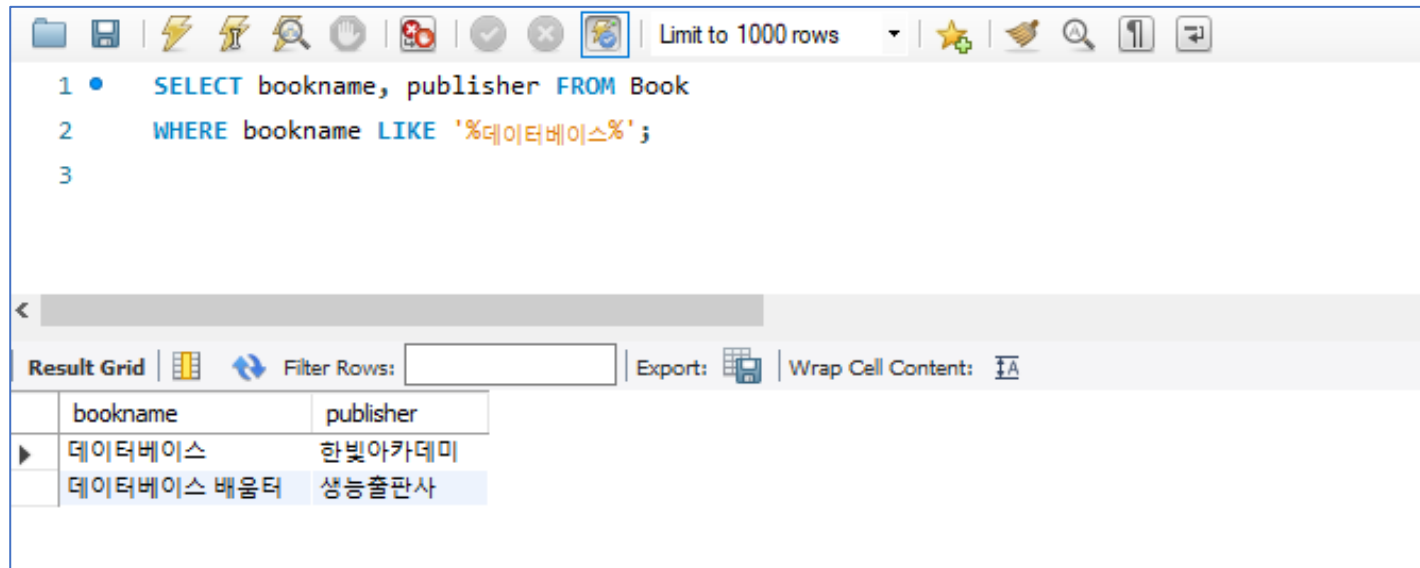
```
SELECT publisher FROM Book  
WHERE bookname LIKE '데이터베이스';
```



[실습] 패턴-2

❖ 책 이름에 '데이터베이스'가 포함된 책 이름과 출판사 검색

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



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT bookname, publisher FROM Book  
2 WHERE bookname LIKE '%데이터베이스%';  
3
```

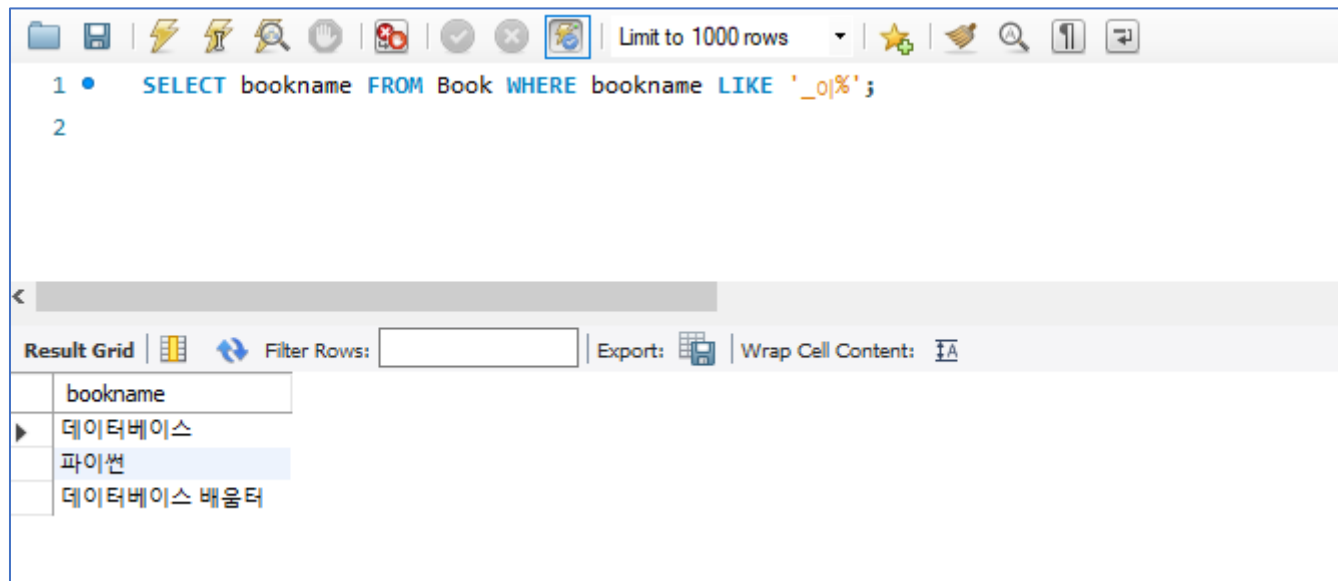
Below the query, the results are displayed in a table with columns 'bookname' and 'publisher'.

bookname	publisher
데이터베이스	한빛아카데미
데이터베이스 배우기	생능출판사

[실습] 패턴-3

❖ 왼쪽 두 번째 위치에 '이'라는 문자를 갖는 책 이름을 검색

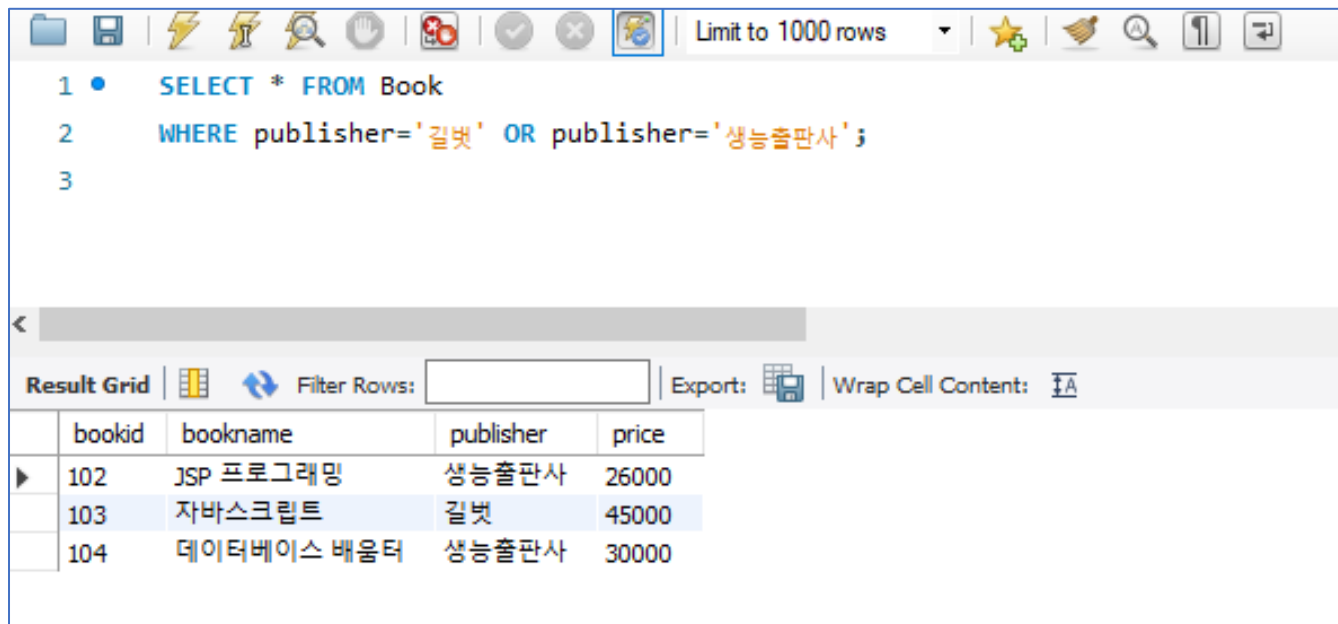
```
SELECT bookname FROM Book  
WHERE bookname LIKE '_이%';
```



[실습] 복합조건-1

❖ 출판사가 '길벗' 또는 '생능출판사'인 책 정보 검색

```
SELECT * FROM Book  
WHERE publisher='길벗' OR publisher='생능출판사';
```



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query is entered in a text area:

```
1 • SELECT * FROM Book  
2 WHERE publisher='길벗' OR publisher='생능출판사';  
3
```

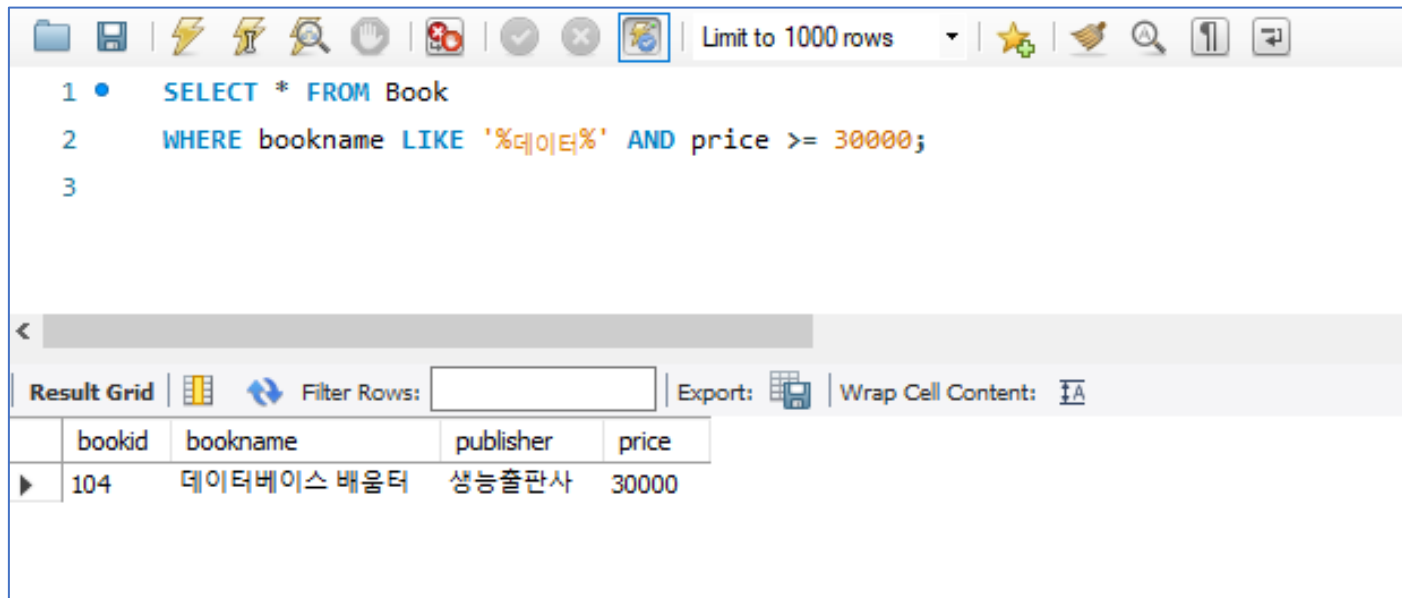
Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid displays the following data:

	bookid	bookname	publisher	price
▶	102	JSP 프로그래밍	생능출판사	26000
	103	자바스크립트	길벗	45000
	104	데이터베이스 배우터	생능출판사	30000

[실습] 복합조건-2

❖ 데이터 관련 도서 중, 가격이 30,000원 이상인 책 검색

```
SELECT * FROM Book  
WHERE bookname LIKE '%데이터%' AND price >= 30000;
```



The screenshot shows a database query tool interface. The top toolbar includes icons for file operations, execution, and settings. The query editor displays the SQL query: `SELECT * FROM Book WHERE bookname LIKE '%데이터%' AND price >= 30000;`. Below the query editor, the 'Result Grid' tab is active, showing a table with the following data:

bookid	bookname	publisher	price
104	데이터베이스 배우터	생능출판사	30000



02

데이터 정렬하기

ORDER BY

❖ ORDER BY

- SELECT 문의 가장 마지막에 ORDER BY를 추가하여 데이터를 정렬할 수 있음

```
SELECT 속성이름 FROM 테이블이름  
WHERE 검색조건  
ORDER BY 속성이름 정렬방식;
```


ORDER BY

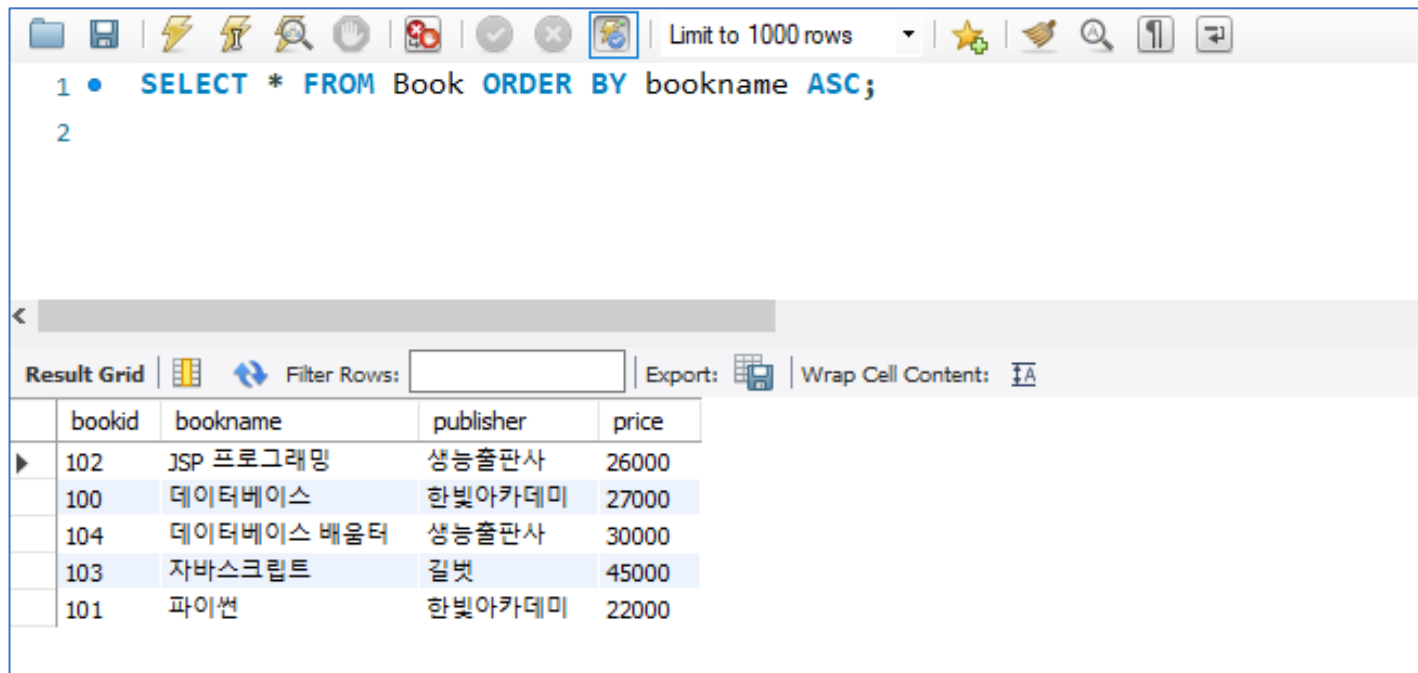
❖ 정렬 방식

	ASC(오름차순)	DESC(내림차순)
숫자	작은 값부터 정렬	큰 값부터 정렬
문자	사전 순서로 정렬	사전 반대 순서로 정렬
날짜	빠른 날짜 순서로 정렬	늦은 날짜 순서로 정렬

[실습] ORDER BY-1

❖ 도서를 이름순으로 검색(오름차순)

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



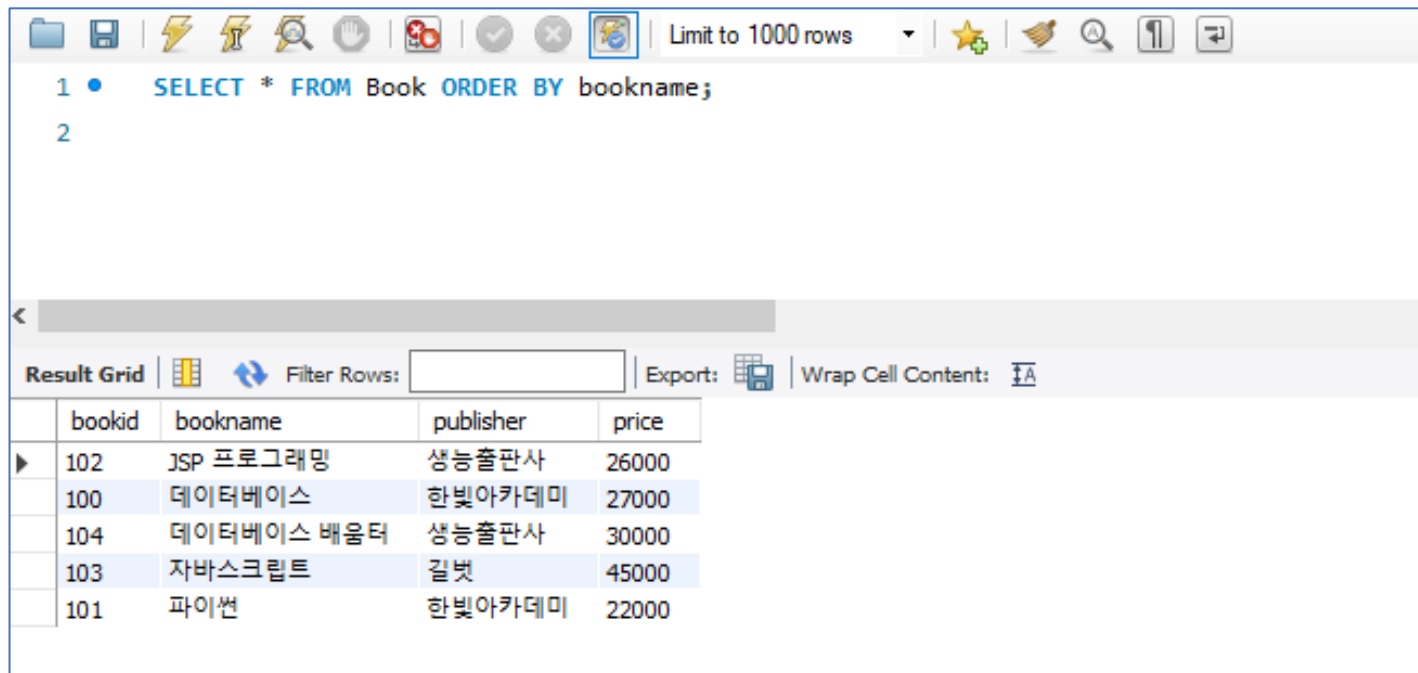
The screenshot shows a database query tool interface. The query editor at the top contains the SQL statement: `SELECT * FROM Book ORDER BY bookname ASC;`. Below the editor, the 'Result Grid' tab is active, displaying a table of book records. The table has five columns: bookid, bookname, publisher, and price. The records are sorted by bookname in ascending order. The interface also includes a toolbar with various icons and a 'Limit to 1000 rows' dropdown menu.

bookid	bookname	publisher	price
102	JSP 프로그래밍	생능출판사	26000
100	데이터베이스	한빛아카데미	27000
104	데이터베이스 배우터	생능출판사	30000
103	자바스크립트	길벗	45000
101	파이썬	한빛아카데미	22000

[실습] ORDER BY-2

❖ 도서를 이름순으로 검색(오름차순)

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



Limit to 1000 rows

1 • SELECT * FROM Book ORDER BY bookname;
2

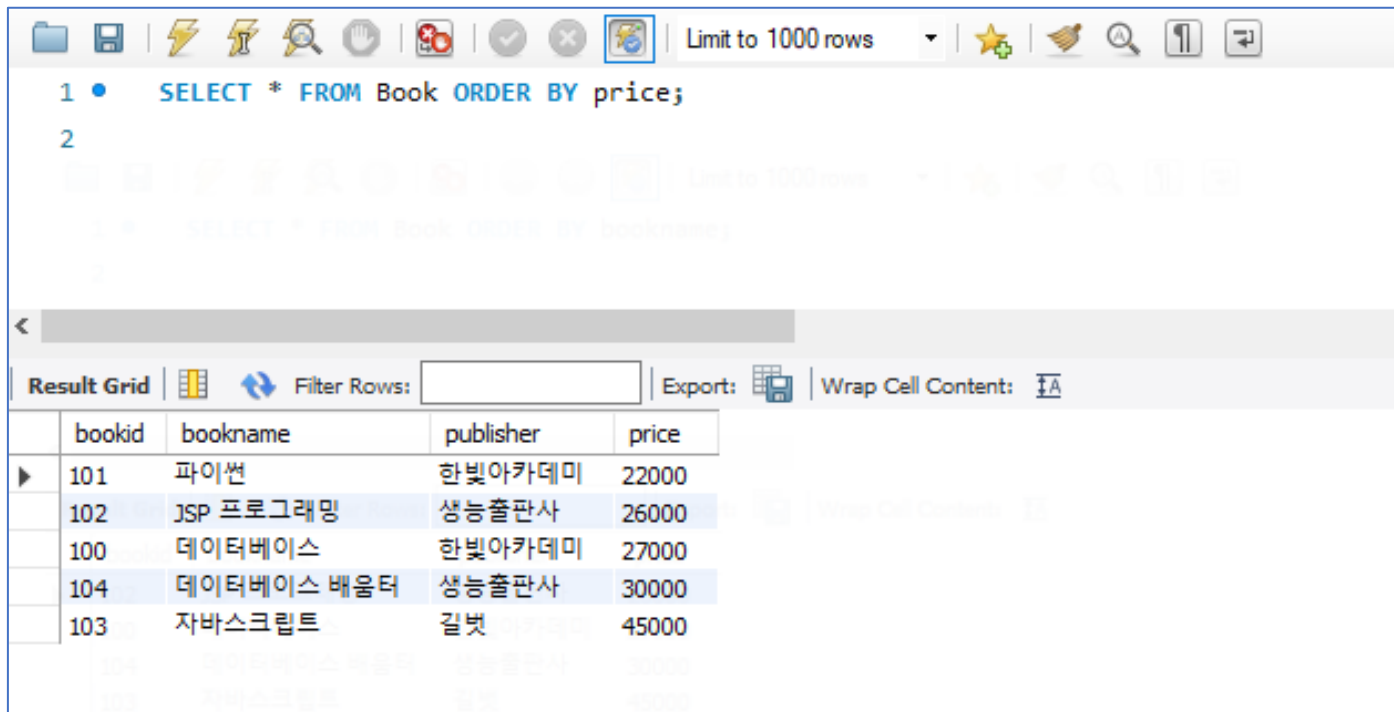
Result Grid

	bookid	bookname	publisher	price
▶	102	JSP 프로그래밍	생능출판사	26000
	100	데이터베이스	한빛아카데미	27000
	104	데이터베이스 배우터	생능출판사	30000
	103	자바스크립트	길벗	45000
	101	파이썬	한빛아카데미	22000

[실습] ORDER BY-3

❖ 도서를 가격순으로 검색(오름차순)

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



1 • SELECT * FROM Book ORDER BY price;

2

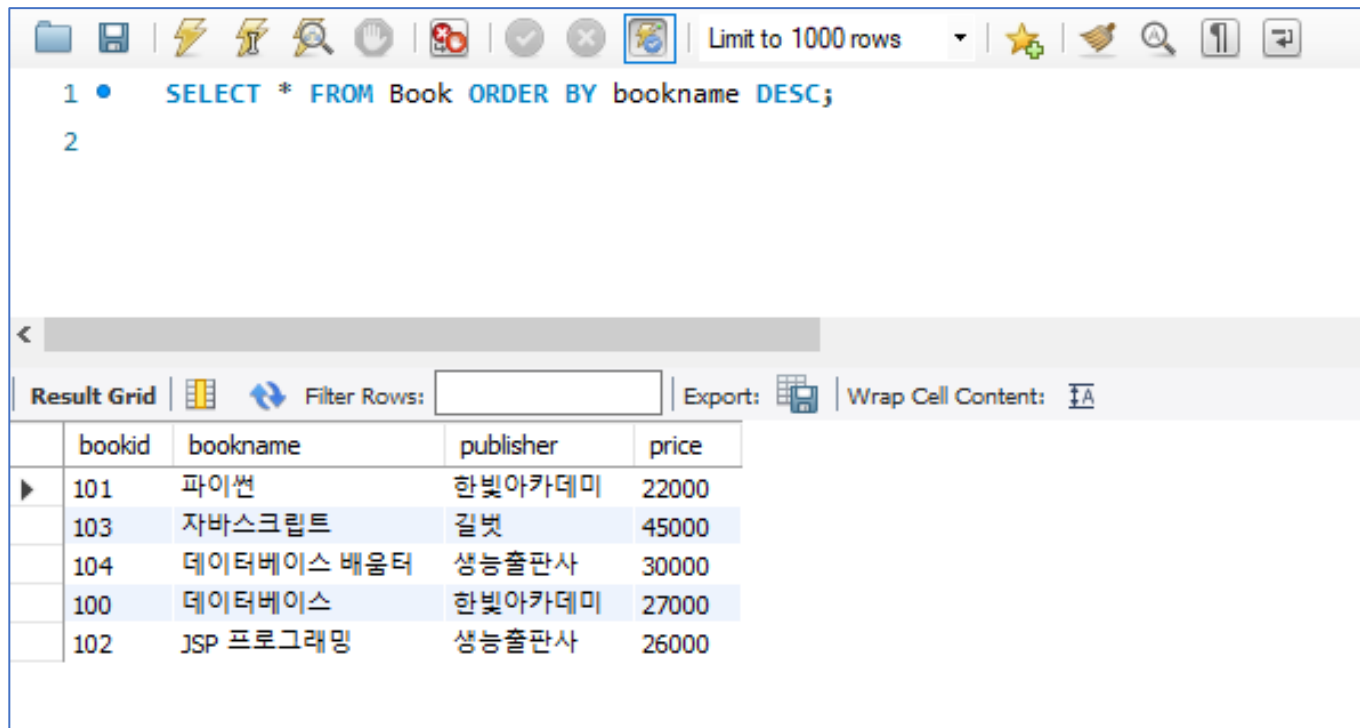
Result Grid

	bookid	bookname	publisher	price
▶	101	파이썬	한빛아카데미	22000
	102	JSP 프로그래밍	생능출판사	26000
	100	데이터베이스	한빛아카데미	27000
	104	데이터베이스 배우기	생능출판사	30000
	103	자바스크립트	길벗	45000

[실습] ORDER BY-4

❖ 도서를 이름순으로 검색(내림차순)

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



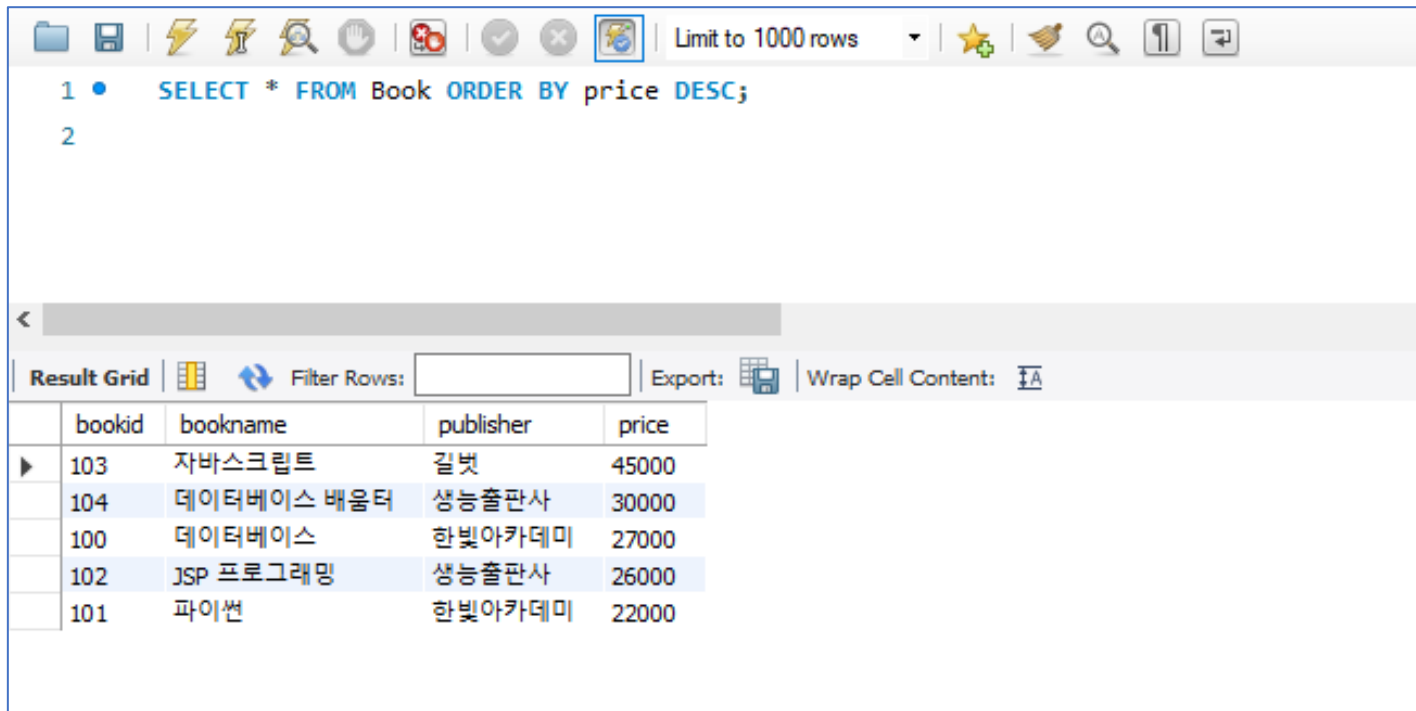
The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL query is entered in a text area: `1 • SELECT * FROM Book ORDER BY bookname DESC;`. Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid displays the following data:

	bookid	bookname	publisher	price
▶	101	파이썬	한빛아카데미	22000
	103	자바스크립트	길벗	45000
	104	데이터베이스 배우터	생능출판사	30000
	100	데이터베이스	한빛아카데미	27000
	102	JSP 프로그래밍	생능출판사	26000

[실습] ORDER BY-4

❖ 도서를 가격순으로 검색(내림차순)

```
SELECT * FROM Book ORDER BY price DESC;
```



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query is entered in a text area:

```
1 • SELECT * FROM Book ORDER BY price DESC;  
2
```

Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid displays the following data:

	bookid	bookname	publisher	price
▶	103	자바스크립트	길벗	45000
	104	데이터베이스 배우터	생능출판사	30000
	100	데이터베이스	한빛아카데미	27000
	102	JSP 프로그래밍	생능출판사	26000
	101	파이썬	한빛아카데미	22000

Database

◆ SQL-3

정수아

Contents

01 집계 함수

02 데이터 그룹화하기

03 중복 데이터 제거하기

04 데이터 수정하기

05 데이터 삭제하기

06 테이블 속성 변경하기



01

집계 함수

집계 함수

❖ 집계 함수

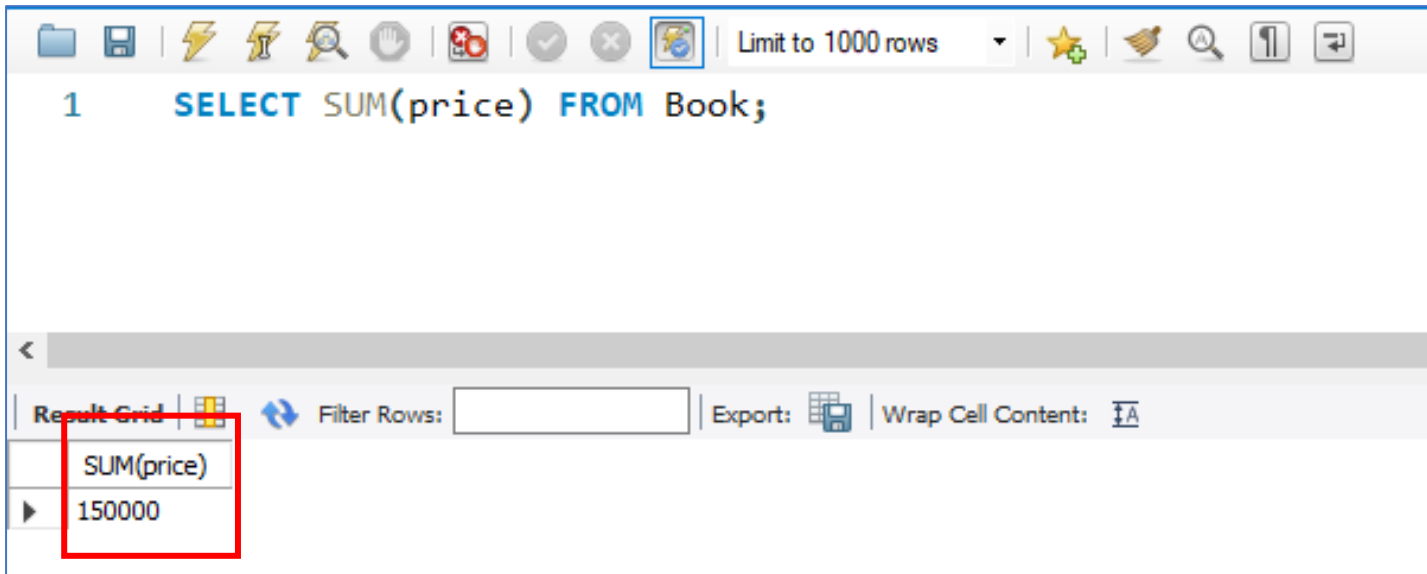
- 전체 데이터를 그룹별로 구분하여 통계적인 결과를 구하기 위해 사용하는 함수
- 집계 함수 종류

함수 종류	설명
SUM	그룹의 누적 합계를 반환
AVG	그룹의 평균을 반환
COUNT	그룹의 총 개수를 반환
MAX	그룹의 최대값을 반환
MIN	그룹의 최소값을 반환

[실습] SUM 함수

❖ 전체 도서 가격의 합 출력

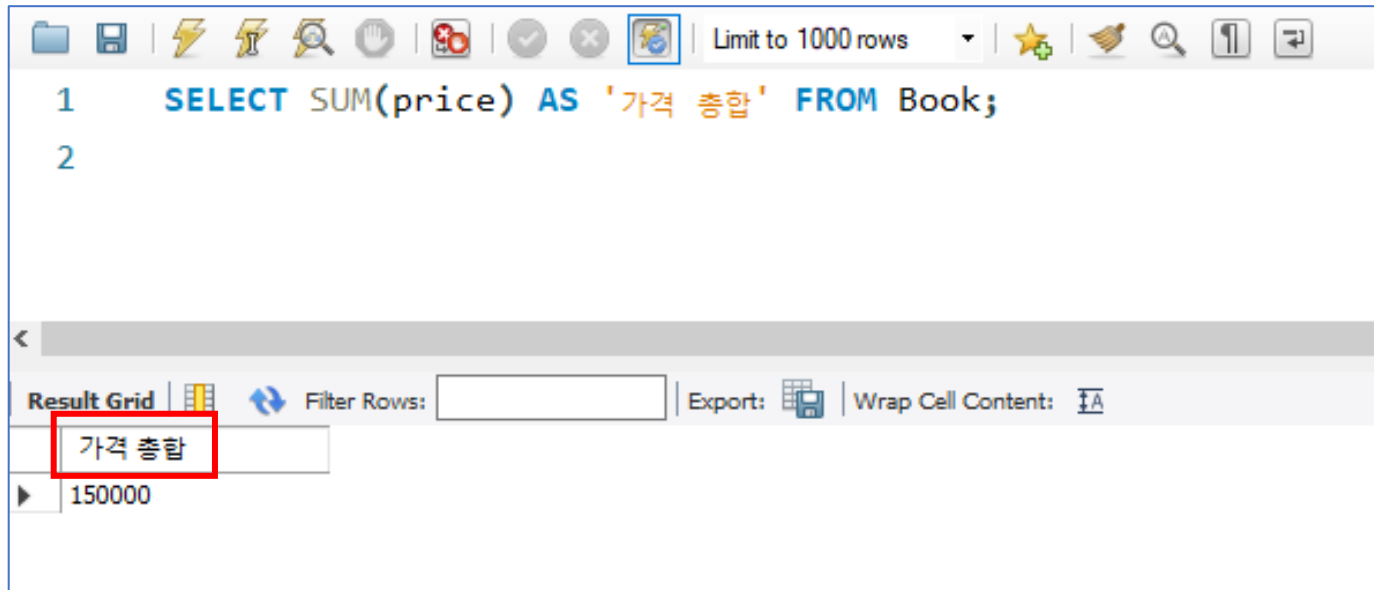
```
SELECT SUM(price) FROM Book;
```



[실습] SUM 함수

❖ 전체 도서 가격의 합 출력 - 열 제목 추가

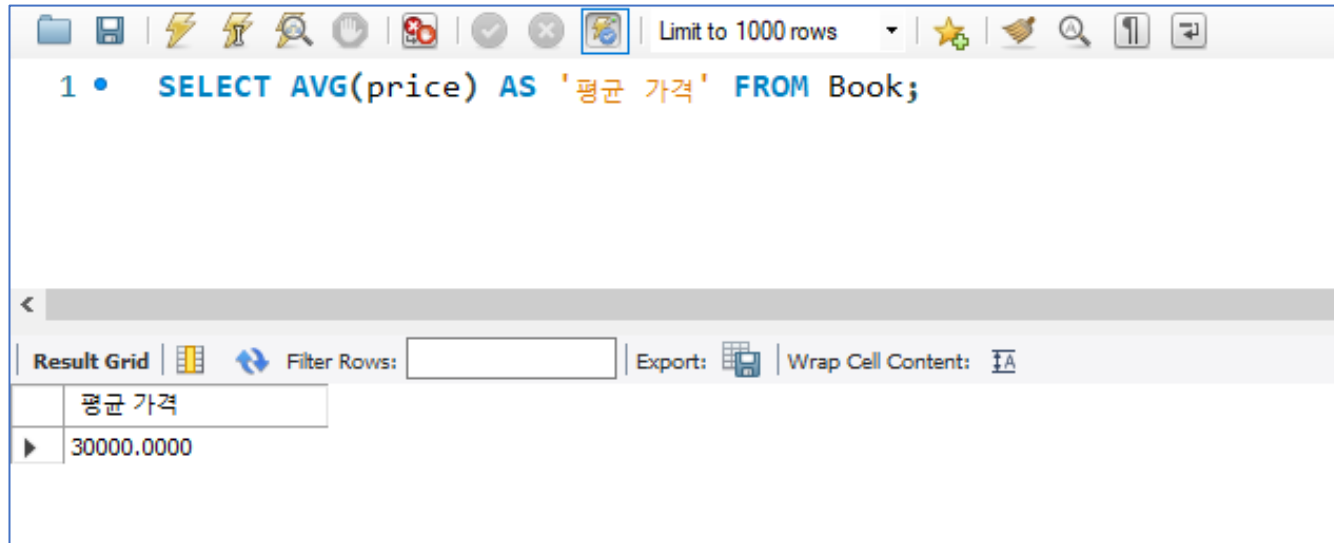
```
SELECT SUM(price) AS '가격 총합' FROM Book;
```



[실습] AVG 함수

❖ 전체 도서들의 평균 가격 출력

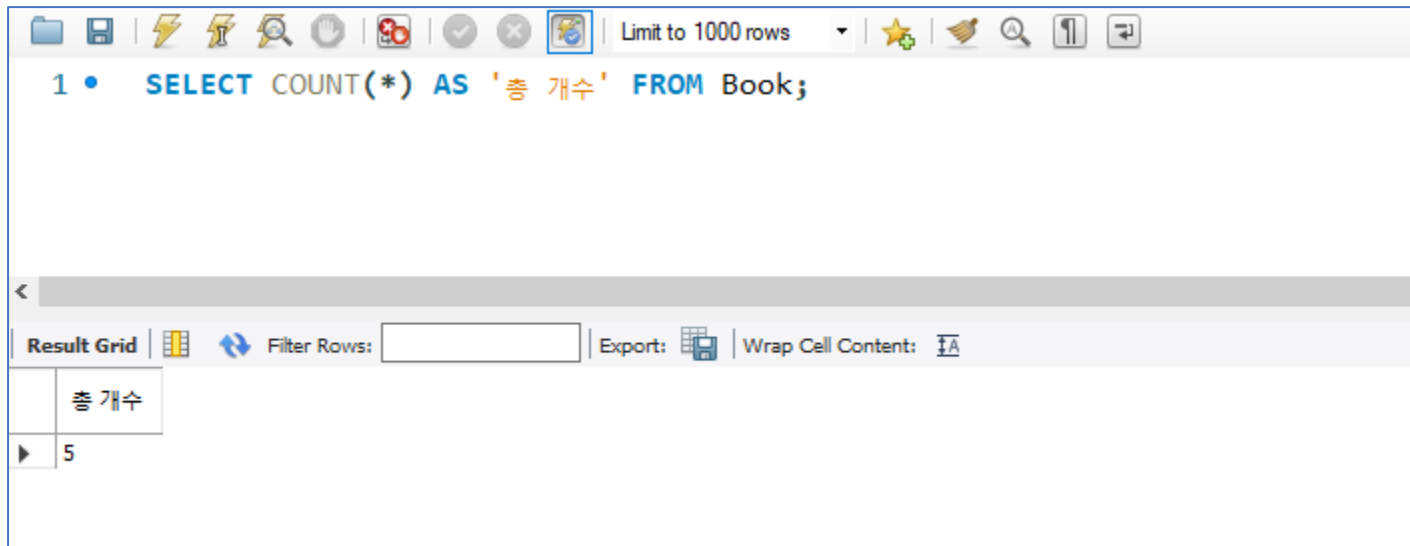
```
SELECT AVG(price) AS '평균 가격' FROM Book;
```



[실습] COUNT 함수

❖ 도서의 총 개수 출력

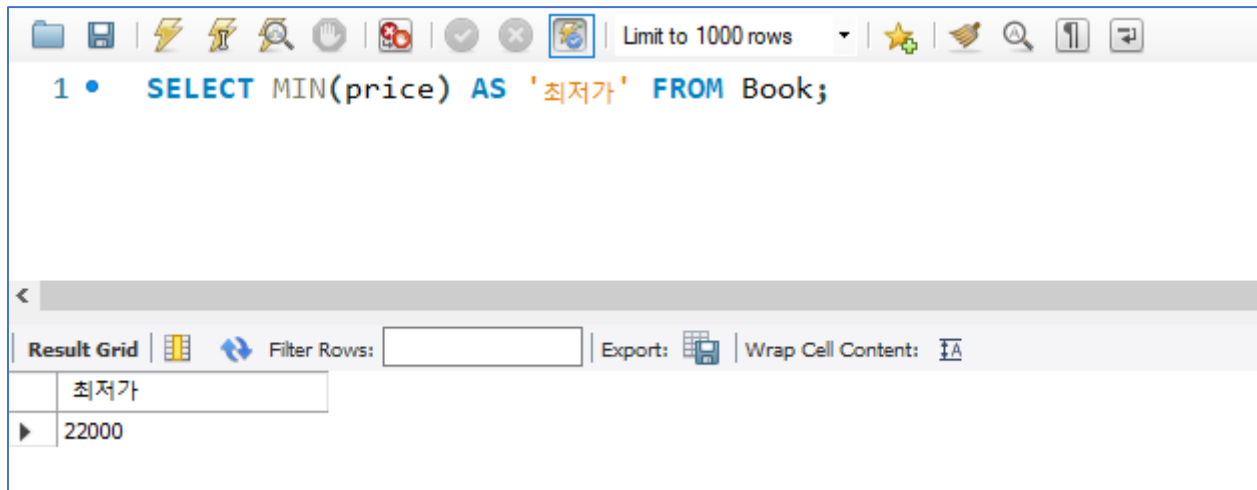
```
SELECT COUNT(*) AS '총 개수' FROM Book;
```



[실습] MIN 함수

❖ 최저가 도서 계산

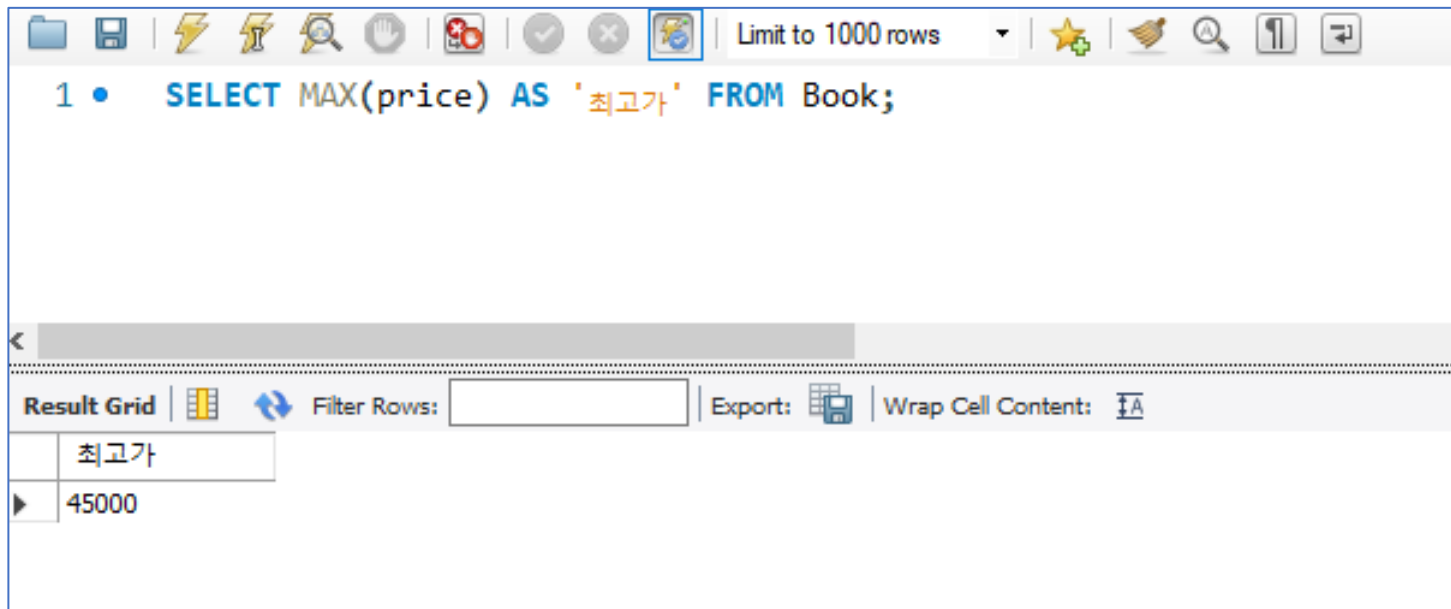
```
SELECT MIN(price) AS '최저가' FROM Book;
```



[실습] MAX 함수

❖ 최고가 도서 계산

```
SELECT MAX(price) AS '최고가' FROM Book;
```



The screenshot shows a SQL query editor interface. The query entered is `SELECT MAX(price) AS '최고가' FROM Book;`. The interface includes a toolbar with various icons and a "Limit to 1000 rows" dropdown. Below the query editor, there is a "Result Grid" section. The grid displays the result of the query, which is the maximum price of 45000, labeled as '최고가'.

최고가
45000



02

데이터 그룹화하기

데이터 그룹화하기

❖ GROUP BY

- 속성 값이 같은 것끼리 그룹으로 생성 → 데이터 중복 제거
- 주의사항
 - SELECT 절에는 GROUP BY에서 사용한 속성 및 집계함수만 가능

❖ HAVING

- 그룹화한 데이터를 필터링할 때 사용
- 주의사항
 - 반드시 GROUP BY와 함께 사용해야 함
 - WHERE 절보다 뒤에 나와야 함

데이터 그룹화하기

❖ 사용 방법

```
SELECT 속성이름 또는 집계함수 FROM 테이블이름  
WHERE 검색조건  
GROUP BY 속성이름  
HAVING 검색조건;
```

데이터 그룹화하기

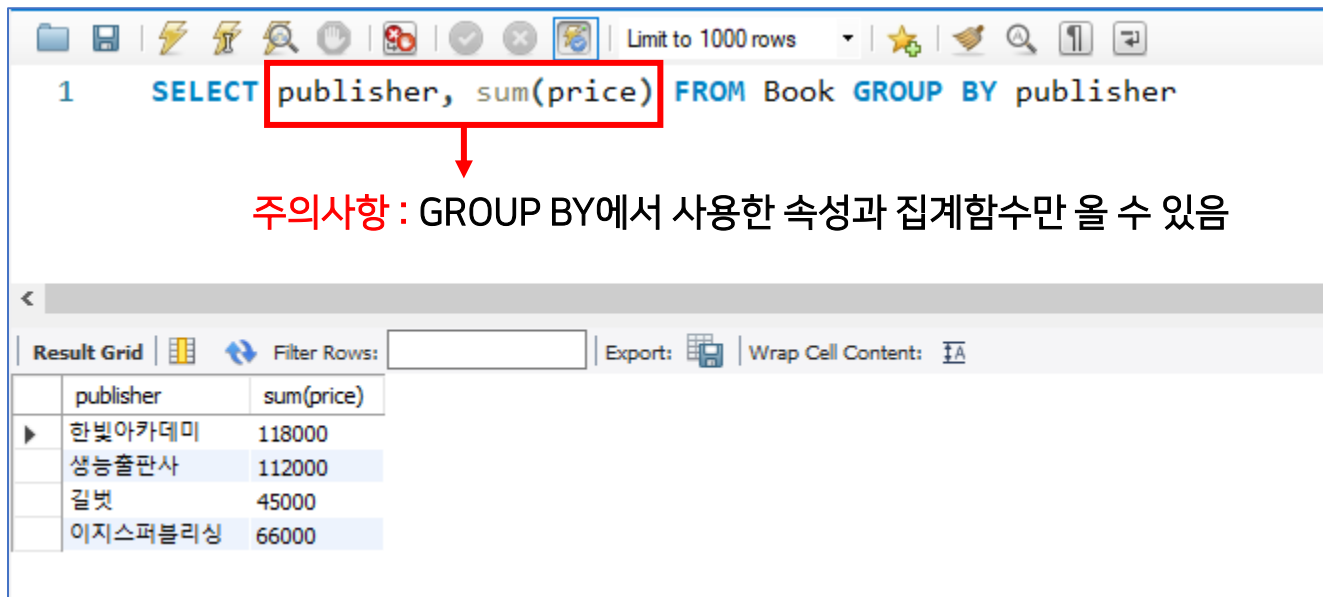
❖ 'Book' 테이블에 데이터 추가하기

bookid	bookname	publisher	price
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터분석	이지스퍼블리싱	25000
107	Chat GPT	생능출판사	29000
108	ReactJS	이지스퍼블리싱	41000
109	홈페이지 만들기 기초	한빛아카데미	32000
110	데이터 시각화	생능출판사	27000

[실습] GROUP BY

❖ 각 출판사별 도서 금액의 총합 계산

```
SELECT publisher, sum(price) FROM Book  
GROUP BY publisher
```



1 SELECT publisher, sum(price) FROM Book GROUP BY publisher

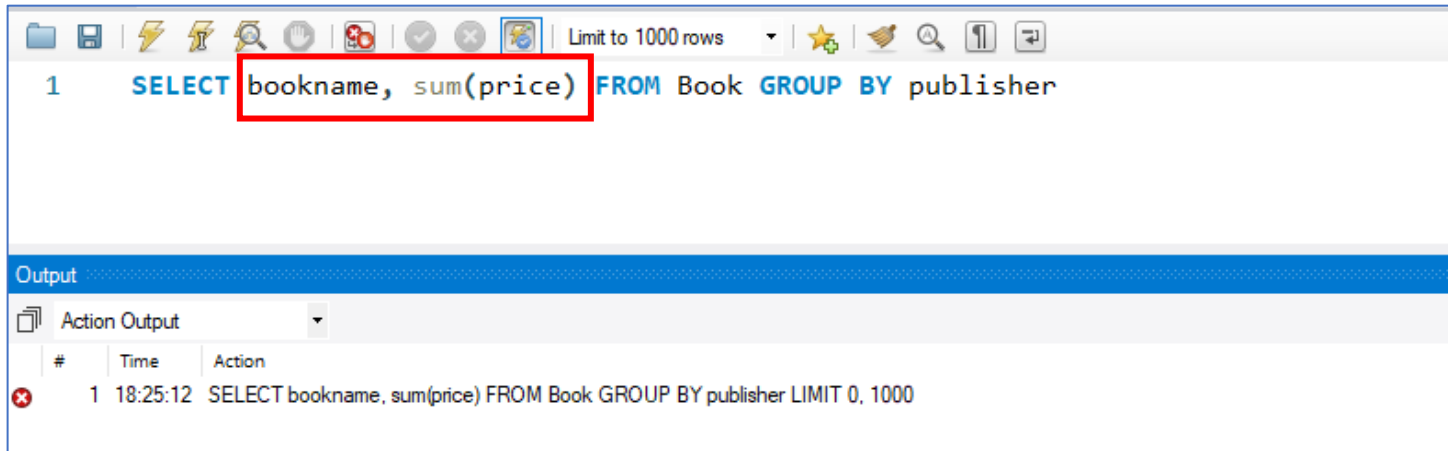
주의사항 : GROUP BY에서 사용한 속성과 집계함수만 올 수 있음

	publisher	sum(price)
▶	한빛아카데미	118000
	생능출판사	112000
	길벗	45000
	이지스퍼블리싱	66000

[실습] GROUP BY

❖ 각 출판사별 도서 금액의 총합 계산

```
SELECT bookname, sum(price) FROM Book  
GROUP BY publisher
```

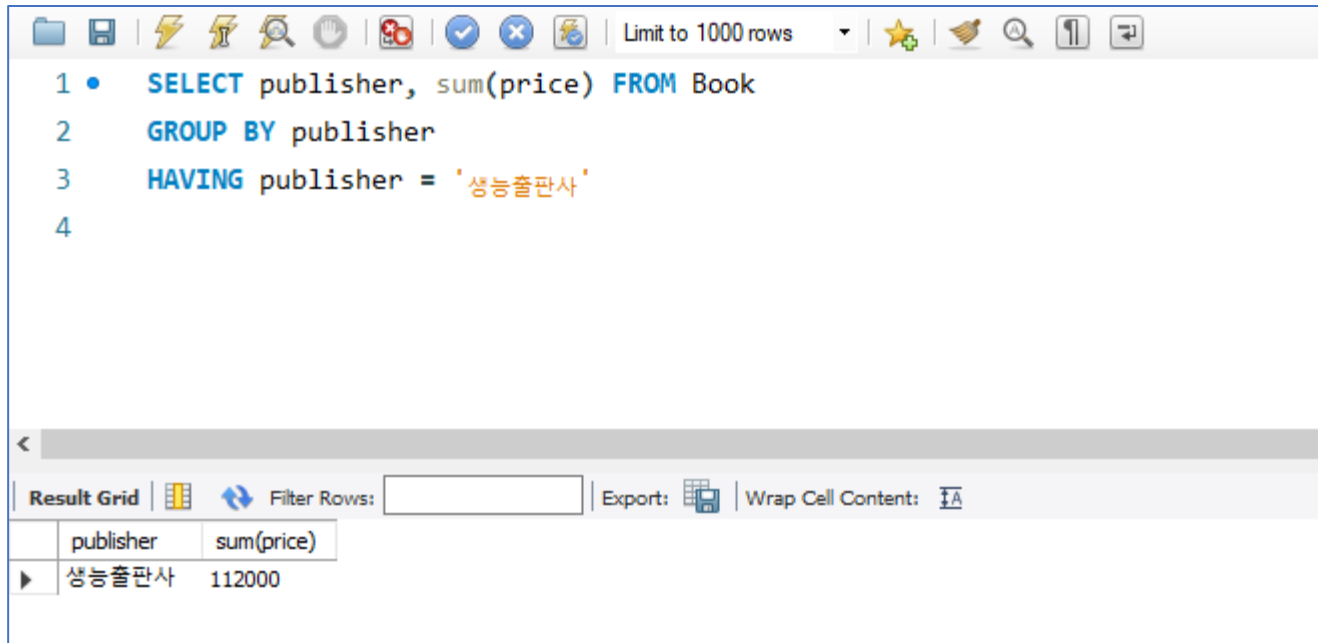


에러발생 이유 : bookname 속성을 사용하면 안됨

[실습] HAVING

- ❖ 출판사를 기준으로 그룹화 후, '생능출판사'의 데이터만 검색하여 도서 금액의 총합 계산

```
SELECT publisher, sum(price) FROM Book  
GROUP BY publisher  
HAVING publisher = '생능출판사'
```



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT publisher, sum(price) FROM Book  
2   GROUP BY publisher  
3   HAVING publisher = '생능출판사'  
4
```

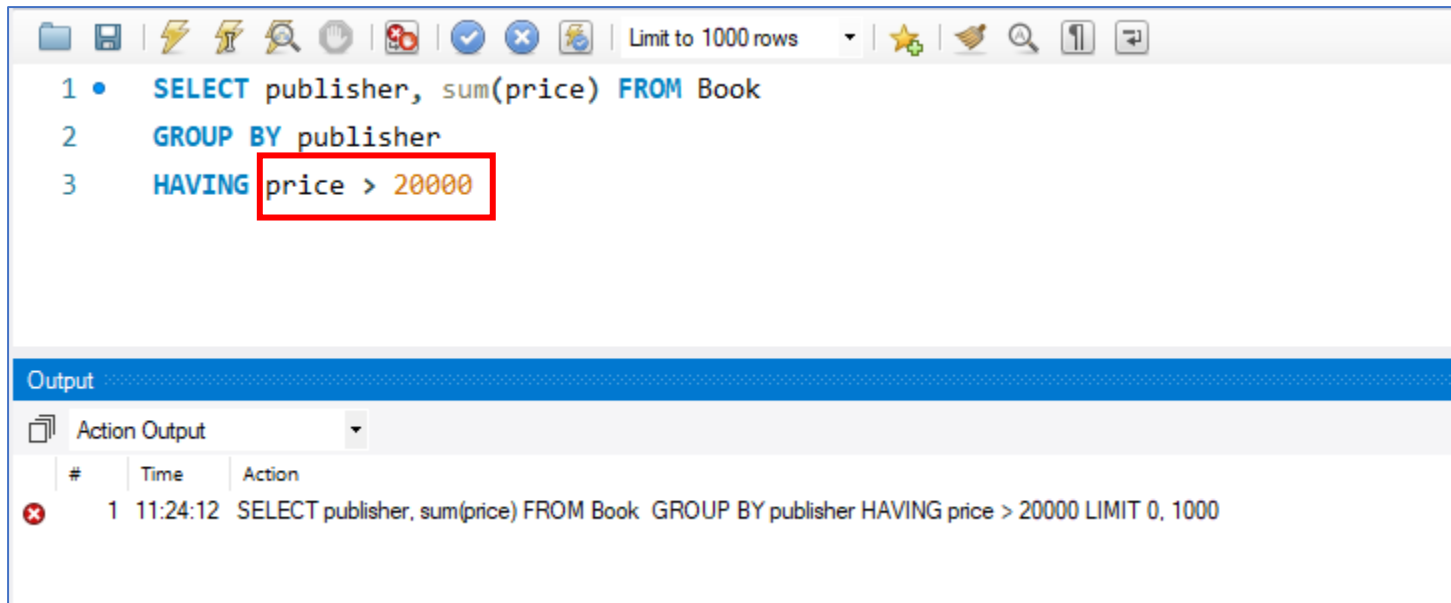
Below the query editor, the results are displayed in a table:

publisher	sum(price)
▶ 생능출판사	112000

[실습] HAVING

- ❖ 출판사를 기준으로 그룹화 후, 가격이 20,000원 이상인 도서만 검색하여 도서 금액의 총합 계산

```
SELECT publisher, sum(price) FROM Book  
GROUP BY publisher  
HAVING price > 20000
```



에러발생 이유 : 그룹화에 사용하지 않은 열을 사용하면 안됨



03

중복 데이터 제거하기

중복 데이터 제거하기

❖ DISTINCT

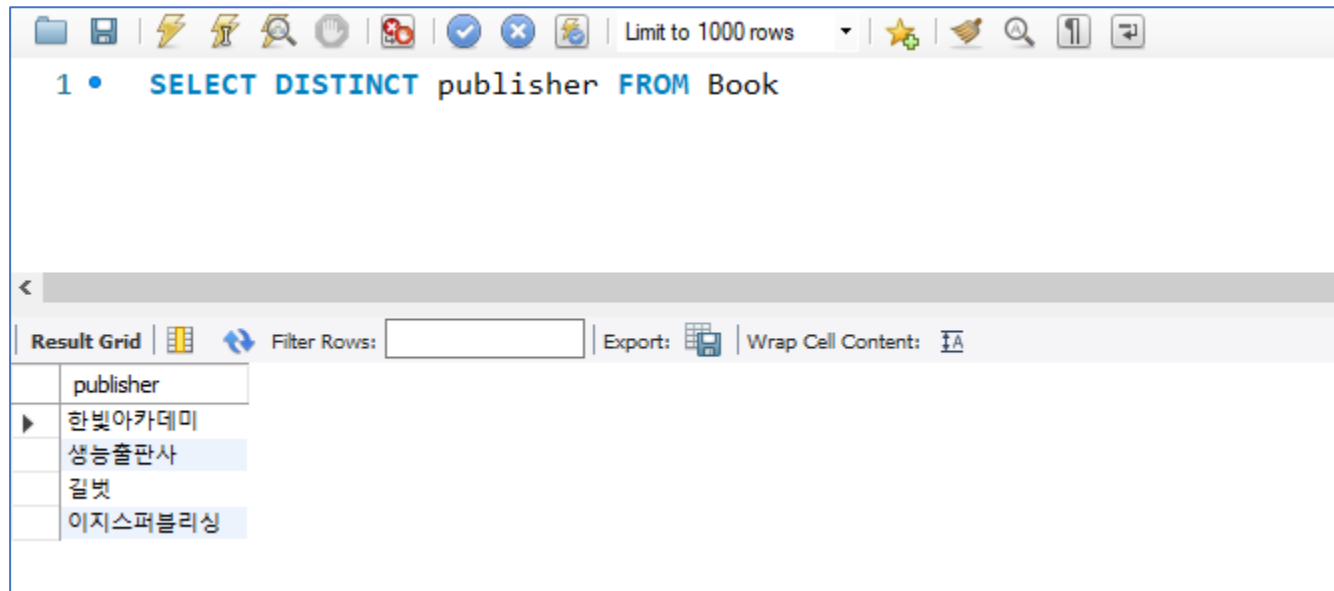
- GROUP BY를 사용하지 않고, 중복 데이터를 제거하는 방법
- 데이터의 중복만 제거할 수 있음
 - 집계나 계산은 불가
- 사용 방법

```
SELECT DISTINCT 속성이름 FROM 테이블이름
```

[실습] DISTINCT

❖ 출판사 이름의 중복을 제거하여 검색

```
SELECT DISTINCT publisher FROM Book
```





04

데이터 수정하기

데이터 수정하기

❖ UPDATE

- 테이블에 삽입된 데이터를 수정할 때 사용
- 사용 방법

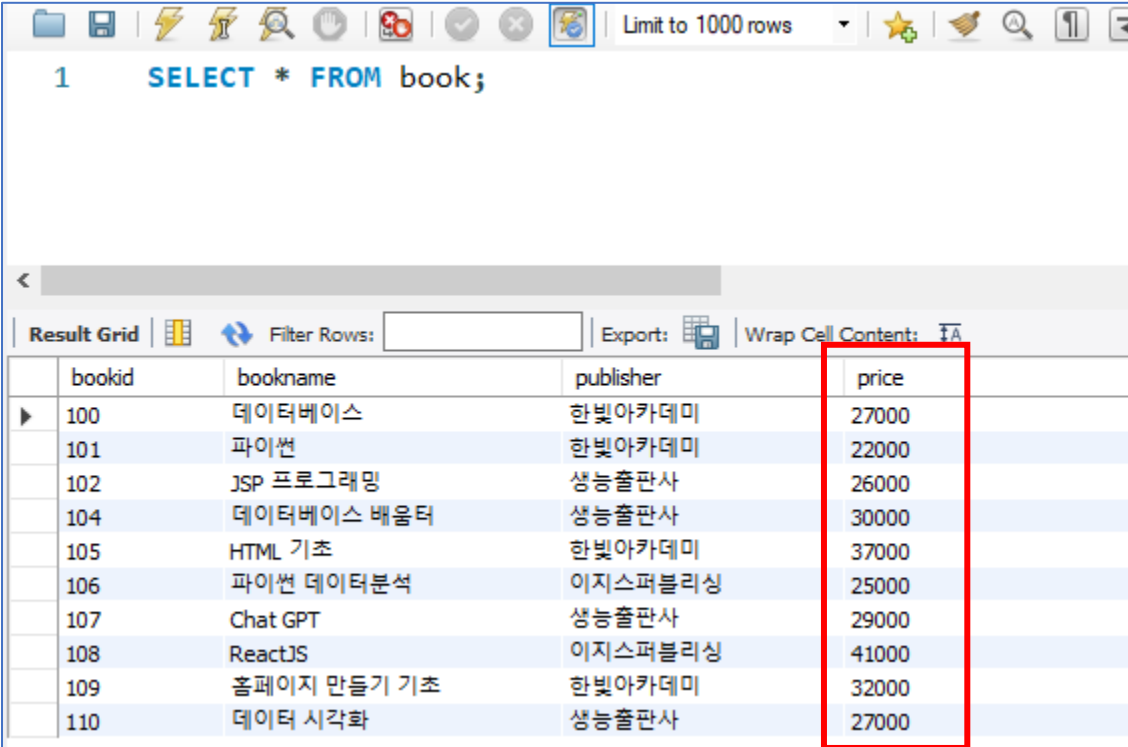
```
UPDATE 테이블이름 SET 속성이름 = 변경할값  
WHERE 속성이름 = 조건;
```

- WHERE 문 생략 가능
 - 생략 시, 테이블 전체 데이터를 수정하므로 주의할 것

[실습] UPDATE

❖ book 테이블 전체 데이터 조회

```
SELECT * FROM book;
```



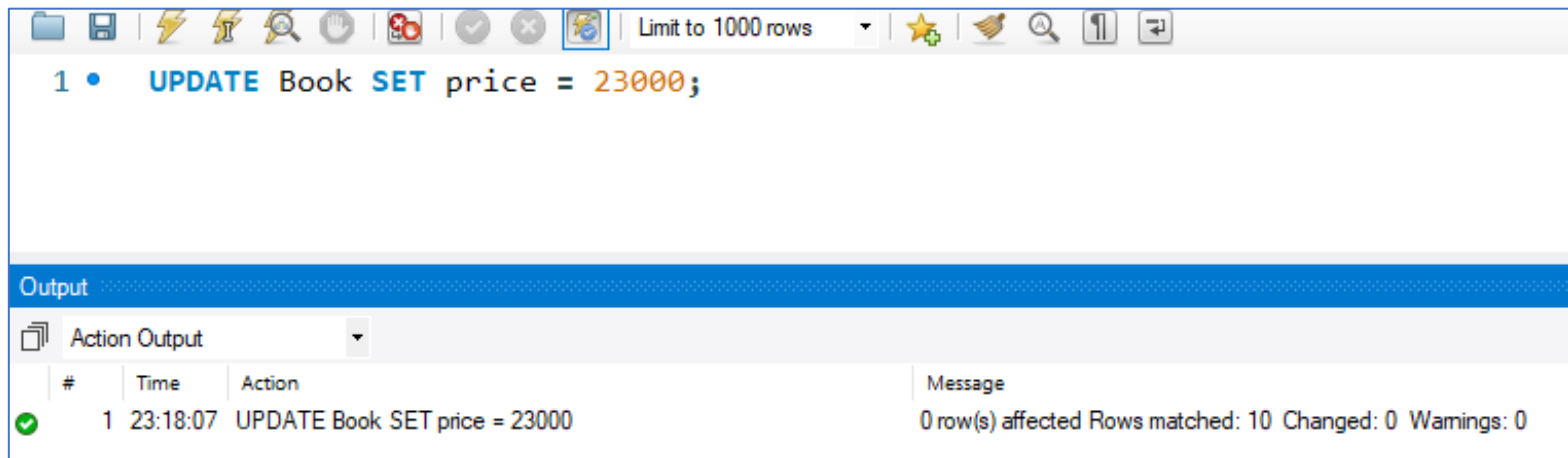
The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query "1 SELECT * FROM book;" is entered. The results are displayed in a "Result Grid" below the query editor. The grid has columns for "bookid", "bookname", "publisher", and "price". The "price" column is highlighted with a red box. The data rows are as follows:

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	22000
102	JSP 프로그래밍	생능출판사	26000
104	데이터베이스 배우터	생능출판사	30000
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터분석	이지스퍼블리싱	25000
107	Chat GPT	생능출판사	29000
108	ReactJS	이지스퍼블리싱	41000
109	홈페이지 만들기 기초	한빛아카데미	32000
110	데이터 시각화	생능출판사	27000

[실습] UPDATE

❖ 책의 가격을 23,000원으로 수정

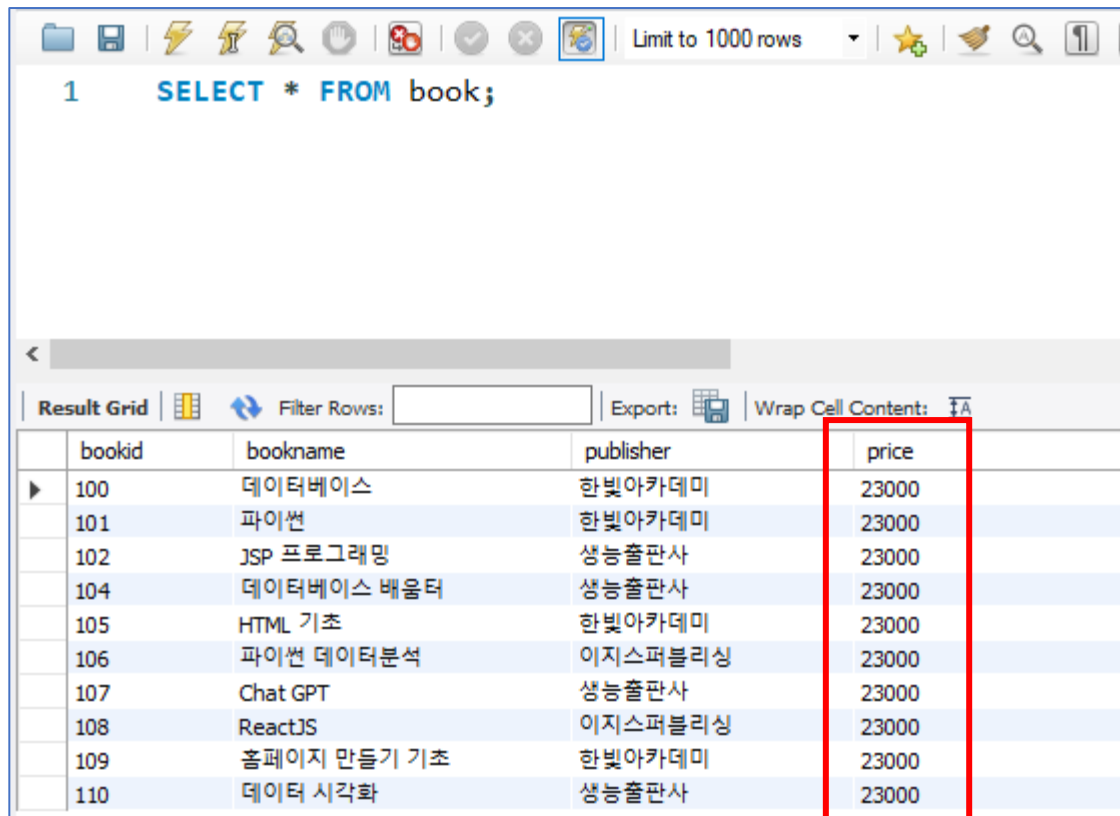
```
UPDATE Book SET price = 23000;
```



[실습] UPDATE

❖ book 테이블 전체 데이터 조회

```
SELECT * FROM book;
```



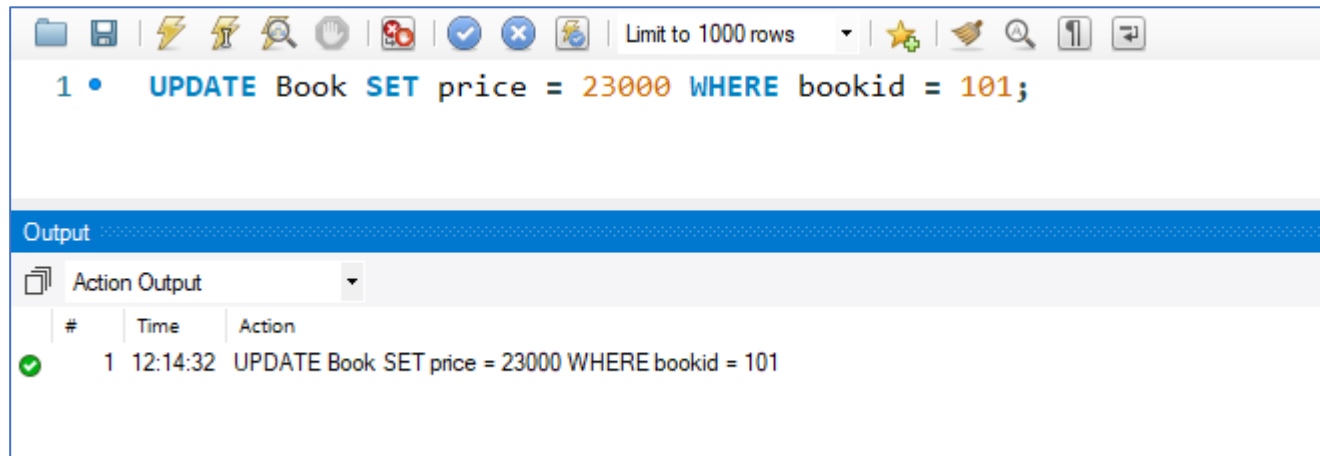
The screenshot shows a database query tool interface. The query editor at the top contains the SQL statement `1 SELECT * FROM book;`. Below the editor, the 'Result Grid' tab is active, displaying a table of 11 rows. The columns are `bookid`, `bookname`, `publisher`, and `price`. The `price` column is highlighted with a red box, indicating that all prices in the table are 23000.

	bookid	bookname	publisher	price
▶	100	데이터베이스	한빛아카데미	23000
	101	파이썬	한빛아카데미	23000
	102	JSP 프로그래밍	생능출판사	23000
	104	데이터베이스 배우터	생능출판사	23000
	105	HTML 기초	한빛아카데미	23000
	106	파이썬 데이터분석	이지스퍼블리싱	23000
	107	Chat GPT	생능출판사	23000
	108	ReactJS	이지스퍼블리싱	23000
	109	홈페이지 만들기 기초	한빛아카데미	23000
	110	데이터 시각화	생능출판사	23000

[실습] UPDATE

❖ bookid가 101인 책의 가격을 23,000원으로 수정

```
UPDATE Book SET price = 23000  
WHERE bookid = 101;
```

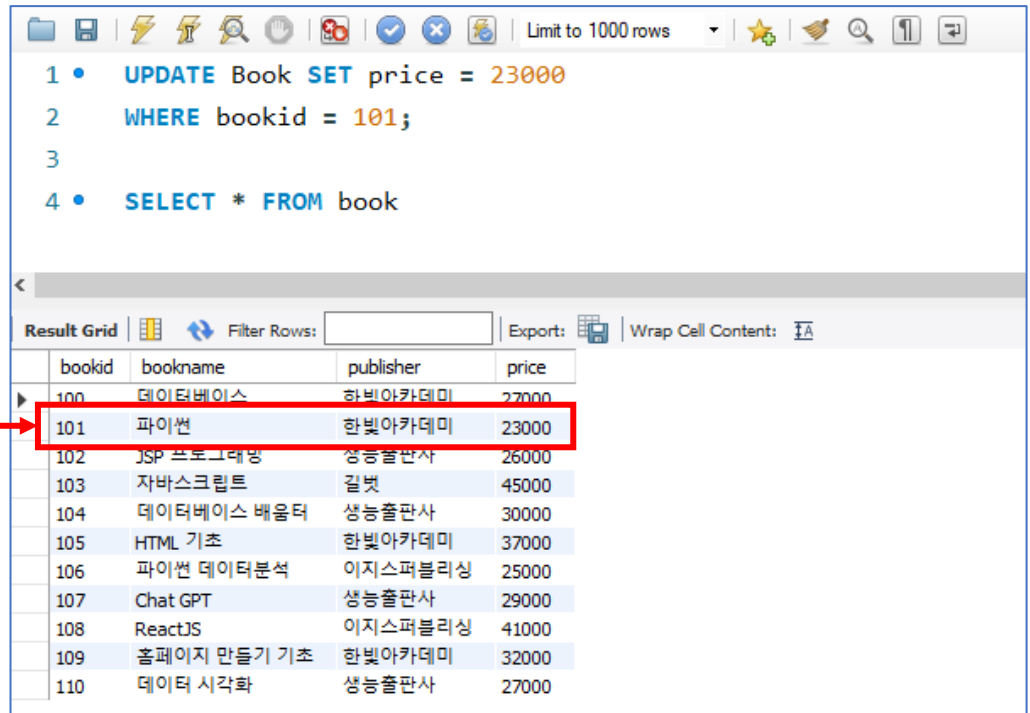


[실습] UPDATE

❖ book 테이블 전체 데이터 조회

```
SELECT * FROM book;
```

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	22000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배우기	생능출판사	30000
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터분석	이지스퍼블리싱	25000



```
1 • UPDATE Book SET price = 23000
2   WHERE bookid = 101;
3
4 • SELECT * FROM book
```

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	23000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배우기	생능출판사	30000
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터분석	이지스퍼블리싱	25000
107	Chat GPT	생능출판사	29000
108	ReactJS	이지스퍼블리싱	41000
109	홈페이지 만들기 기초	한빛아카데미	32000
110	데이터 시각화	생능출판사	27000



05

데이터 삭제하기

데이터 삭제하기

❖ DELETE

DELETE 테이블이름 WHERE 속성이름 = 조건;

- WHERE 문 생략 가능
 - 생략 시, 테이블 전체 데이터를 삭제하므로 주의할 것

[실습] DELETE

❖ '길벗' 출판사의 데이터 삭제

```
DELETE FROM Book WHERE publisher = '길벗';
```

```
SELECT * FROM book;
```

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	22000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배우기	생능출판사	30000
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터분석	이지스퍼블리싱	25000

삭제

1 SELECT * FROM book;

2

Result Grid

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	23000
102	JSP 프로그래밍	생능출판사	26000
104	데이터베이스 배우기	생능출판사	30000
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터분석	이지스퍼블리싱	25000
107	Chat GPT	생능출판사	29000
108	ReactJS	이지스퍼블리싱	41000
109	홈페이지 만들기 기초	한빛아카데미	32000
110	데이터 시각화	생능출판사	27000



06

테이블 속성 변경하기

테이블 속성 변경하기

❖ 속성 추가하기

ALTER TABLE 테이블이름 **변경옵션**

- 변경 옵션

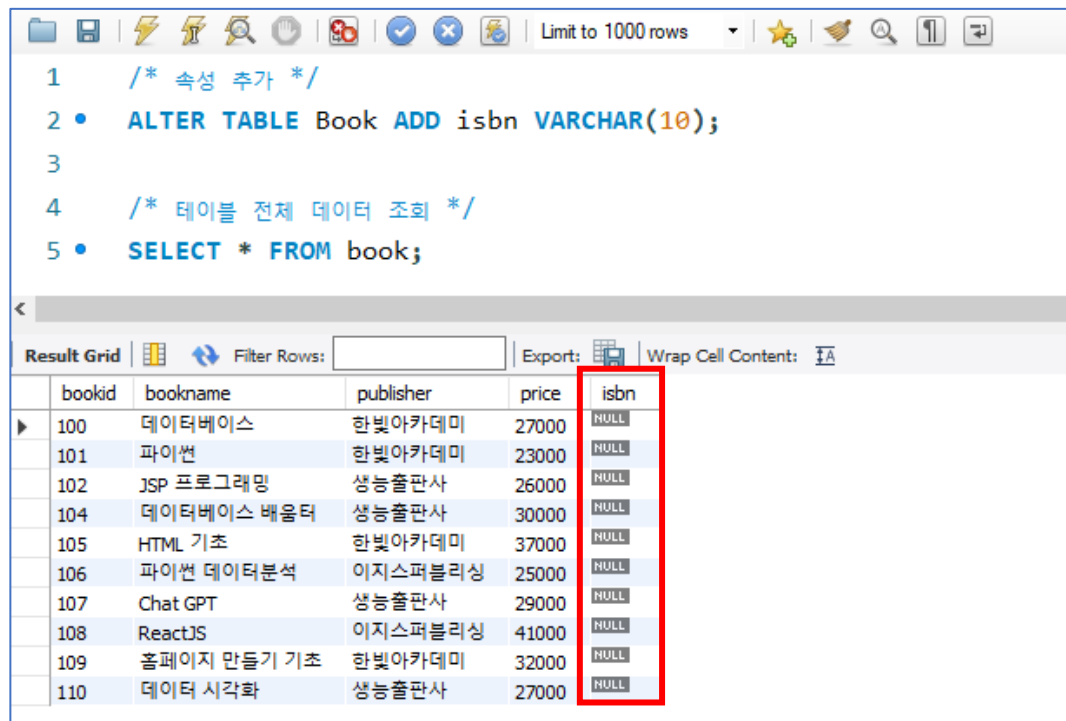
옵션	설명
ADD	속성 추가
MODIFY	속성의 데이터 타입 변경
CHANGE	속성 이름 변경
DROP	속성 삭제

[실습] ADD

❖ Book 테이블에 다음 속성 추가

- VARCHAR(10) 자료형을 가진 isbn 속성 추가

```
ALTER TABLE Book ADD isbn VARCHAR(10);
```



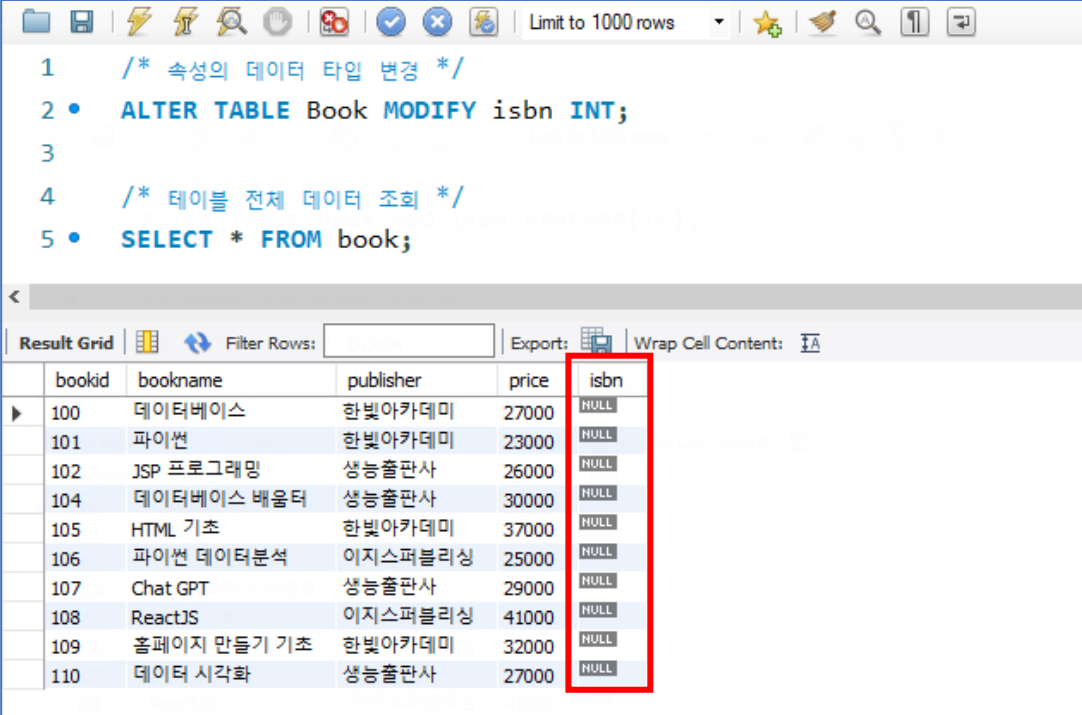
The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, execution, and settings. Below the toolbar, a SQL script is displayed with five lines: a comment, an ALTER TABLE statement to add an 'isbn' column of type VARCHAR(10) to the 'Book' table, another comment, and a SELECT statement to view all data in the 'book' table. The bottom section, titled 'Result Grid', shows the execution results. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' option. The result grid itself is a table with five columns: 'bookid', 'bookname', 'publisher', 'price', and 'isbn'. The 'isbn' column is highlighted with a red box, and all its cells contain the value 'NULL'.

bookid	bookname	publisher	price	isbn
100	데이터베이스	한빛아카데미	27000	NULL
101	파이썬	한빛아카데미	23000	NULL
102	JSP 프로그래밍	생능출판사	26000	NULL
104	데이터베이스 배우기	생능출판사	30000	NULL
105	HTML 기초	한빛아카데미	37000	NULL
106	파이썬 데이터분석	이지스퍼블리싱	25000	NULL
107	Chat GPT	생능출판사	29000	NULL
108	ReactJS	이지스퍼블리싱	41000	NULL
109	홈페이지 만들기 기초	한빛아카데미	32000	NULL
110	데이터 시각화	생능출판사	27000	NULL

[실습] MODIFY

❖ Book 테이블의 isbn 속성의 데이터 타입을 INT로 변경

```
ALTER TABLE Book MODIFY isbn INT;
```



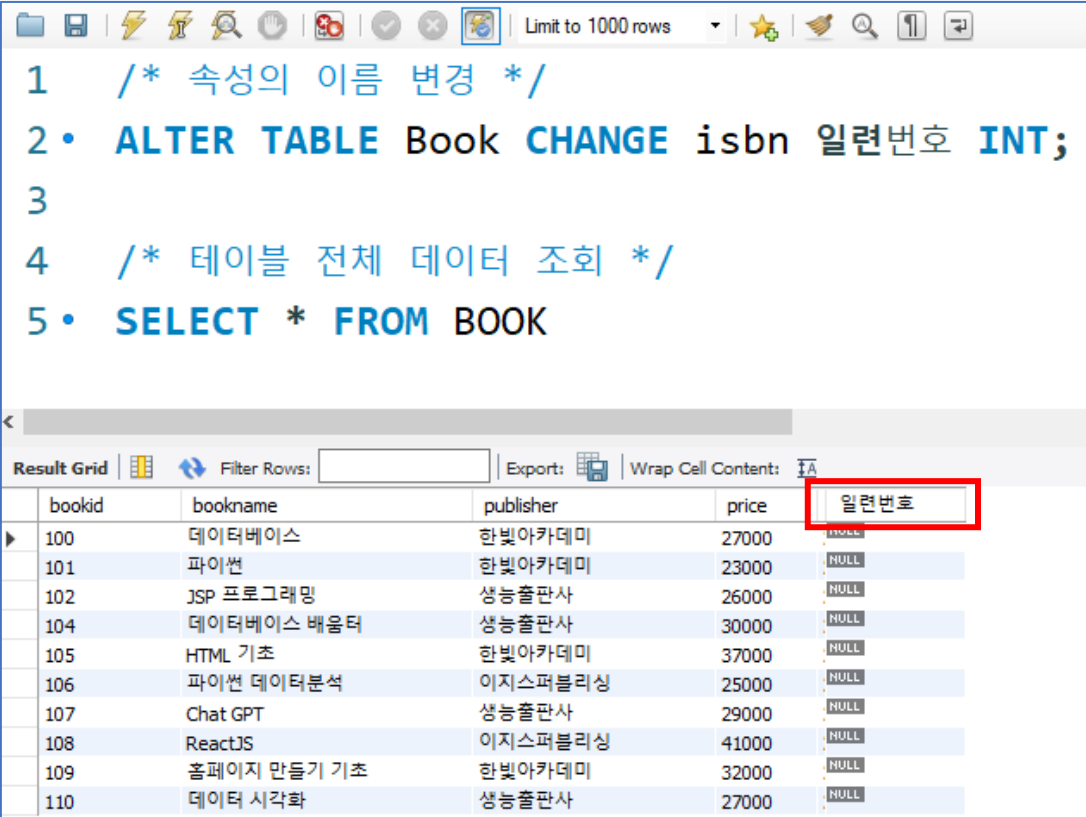
The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, search, and execution. Below the toolbar, a SQL editor contains five lines of code: a comment, an ALTER TABLE statement to change the isbn column to INT, another comment, and a SELECT statement to view the table data. Below the editor, the 'Result Grid' tab is active, displaying a table with 5 columns: bookid, bookname, publisher, price, and isbn. The isbn column is highlighted with a red box, and all its values are NULL. The table contains 11 rows of data.

bookid	bookname	publisher	price	isbn
100	데이터베이스	한빛아카데미	27000	NULL
101	파이썬	한빛아카데미	23000	NULL
102	JSP 프로그래밍	생능출판사	26000	NULL
104	데이터베이스 배우기	생능출판사	30000	NULL
105	HTML 기초	한빛아카데미	37000	NULL
106	파이썬 데이터분석	이지스퍼블리싱	25000	NULL
107	Chat GPT	생능출판사	29000	NULL
108	ReactJS	이지스퍼블리싱	41000	NULL
109	홈페이지 만들기 기초	한빛아카데미	32000	NULL
110	데이터 시각화	생능출판사	27000	NULL

[실습] CHANGE

❖ Book 테이블의 isbn 속성의 이름을 '일련번호'로 변경

```
ALTER TABLE Book CHANGE isbn 일련번호 INT;
```



The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 1000 rows' dropdown. The SQL editor contains five lines of code: a comment, an ALTER TABLE statement to change the 'isbn' column to '일련번호' (Serial Number) of type INT, another comment, and a SELECT statement to view the entire table data.

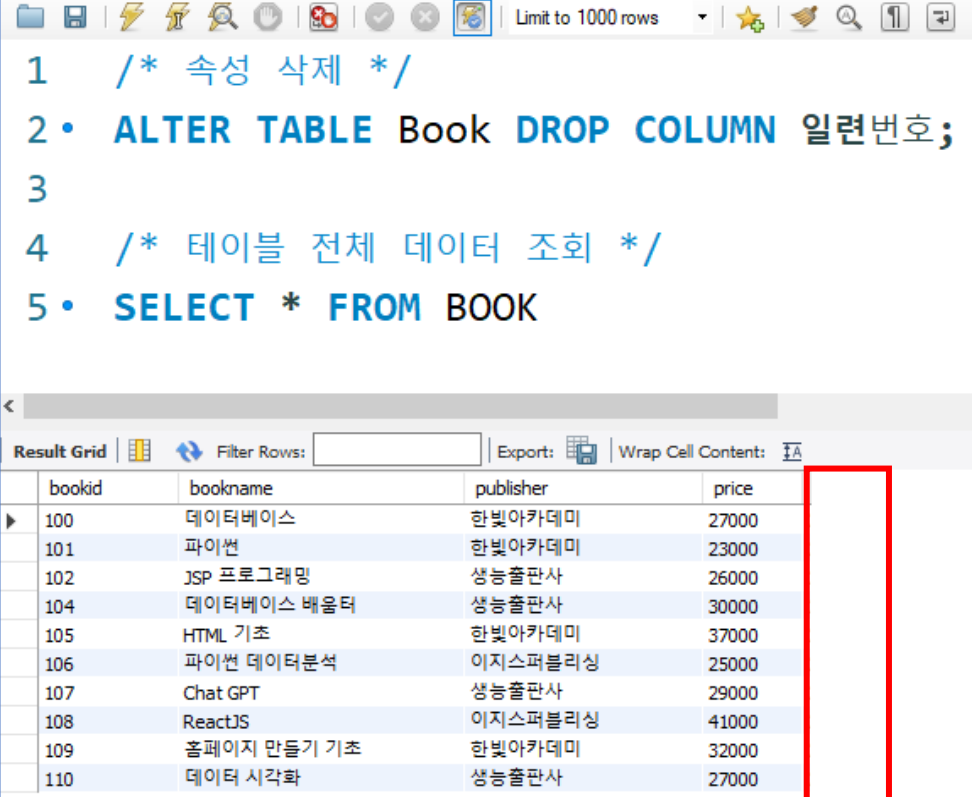
Below the SQL editor is the 'Result Grid' section. It features a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The table data is displayed in a grid with the following columns: bookid, bookname, publisher, price, and 일련번호 (highlighted with a red box). The '일련번호' column contains NULL values for all rows.

	bookid	bookname	publisher	price	일련번호
▶	100	데이터베이스	한빛아카데미	27000	NULL
	101	파이썬	한빛아카데미	23000	NULL
	102	JSP 프로그래밍	생능출판사	26000	NULL
	104	데이터베이스 배우터	생능출판사	30000	NULL
	105	HTML 기초	한빛아카데미	37000	NULL
	106	파이썬 데이터분석	이지스퍼블리싱	25000	NULL
	107	Chat GPT	생능출판사	29000	NULL
	108	ReactJS	이지스퍼블리싱	41000	NULL
	109	홈페이지 만들기 기초	한빛아카데미	32000	NULL
	110	데이터 시각화	생능출판사	27000	NULL

[실습] DROP

❖ Book 테이블의 일련번호 속성을 삭제

ALTER TABLE Book **DROP COLUMN** 일련번호;



The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 1000 rows' dropdown. The SQL editor contains five lines of code: a comment, an ALTER TABLE statement to drop the '일련번호' column, another comment, and a SELECT statement. Below the editor, the 'Result Grid' tab is active, displaying a table with 5 columns: bookid, bookname, publisher, price, and an empty column highlighted with a red box. The table contains 11 rows of data.

	bookid	bookname	publisher	price	
▶	100	데이터베이스	한빛아카데미	27000	
	101	파이썬	한빛아카데미	23000	
	102	JSP 프로그래밍	생능출판사	26000	
	104	데이터베이스 배우터	생능출판사	30000	
	105	HTML 기초	한빛아카데미	37000	
	106	파이썬 데이터분석	이지스퍼블리싱	25000	
	107	Chat GPT	생능출판사	29000	
	108	ReactJS	이지스퍼블리싱	41000	
	109	홈페이지 만들기 기초	한빛아카데미	32000	
	110	데이터 시각화	생능출판사	27000	

Database

◆ SQL-4

정수아

Contents

01 기본키

02 기타 제약조건

03 외래키

04 내장함수



01

기본키

기본키

❖ 기본키(Primary key)

- 테이블에서 특정 튜플(행)을 유일하게 구별할 수 있는 속성
- 특징
 - 기본키로 정의된 속성에는 동일한 값이 중복되어 저장될 수 없음
 - NULL 값을 가질 수 없음

튜플	id	name	age	address	course	score
	1	문종헌	24	서울	영어	A
	2	오한솔	22	부산	수학	B
	3	정국철	25	서울	음악	A
	4	박기석	27	제주	국어	C
	5	안창범	20	광주	수학	B
	6	박홍진	22	부산	컴퓨터	D
	7	공지훈	28	강원	국어	A
	8	정희성	30	제주	음악	C
	9	이봉림	34	대전	영어	D

학생 릴레이션

기본키

❖ 기본키 작성 방법

```
CREATE TABLE 테이블이름(  
    속성이름1 데이터타입 PRIMARY KEY,  
    속성이름2 데이터타입,  
    (...생략...)  
);
```

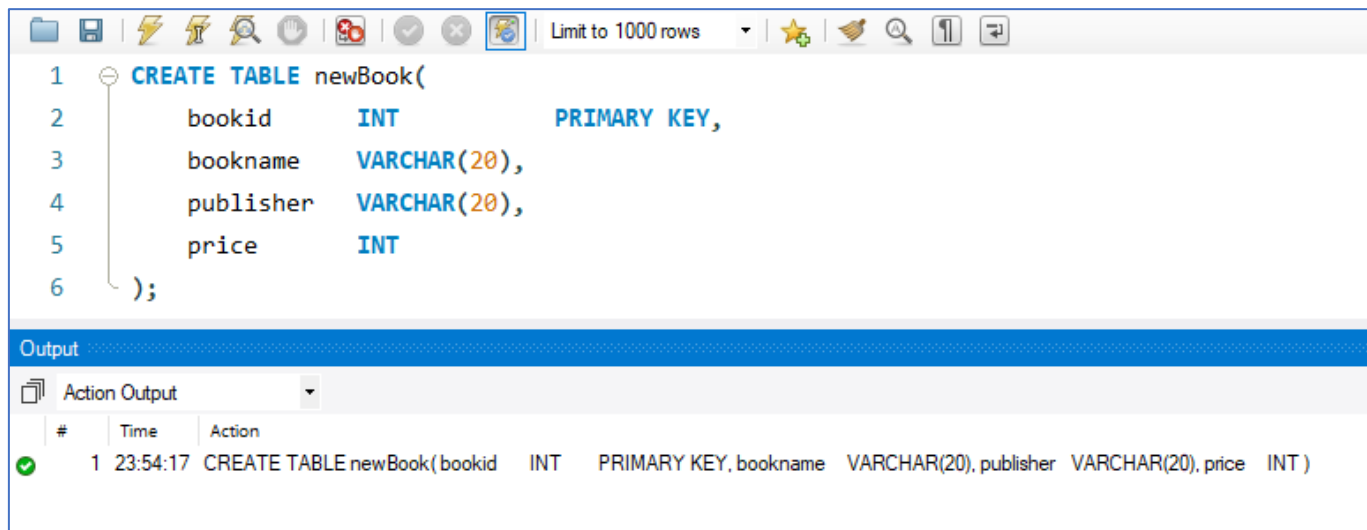
```
CREATE TABLE 테이블이름(  
    속성이름1 데이터타입,  
    속성이름2 데이터타입,  
    (...생략...),  
    PRIMARY KEY(속성이름)  
);
```


[실습] 기본키-1

❖ newBook 테이블 생성하기

- bookid를 기본키로 지정

```
CREATE TABLE newBook(  
    bookid      INT           PRIMARY KEY,  
    bookname    VARCHAR(20),  
    publisher   VARCHAR(20),  
    price       INT  
);
```

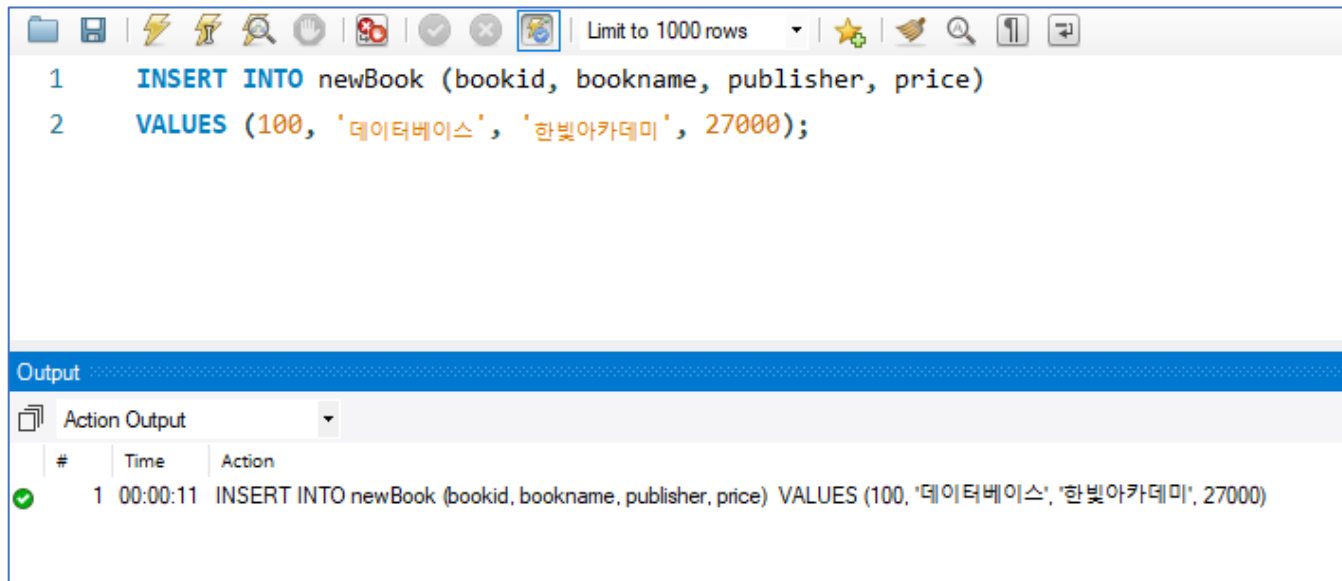


[실습] 기본키-2

❖ 데이터 삽입하기

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000

```
INSERT INTO newBook (bookid, bookname, publisher, price)
VALUES (100, '데이터베이스', '한빛아카데미', 27000);
```

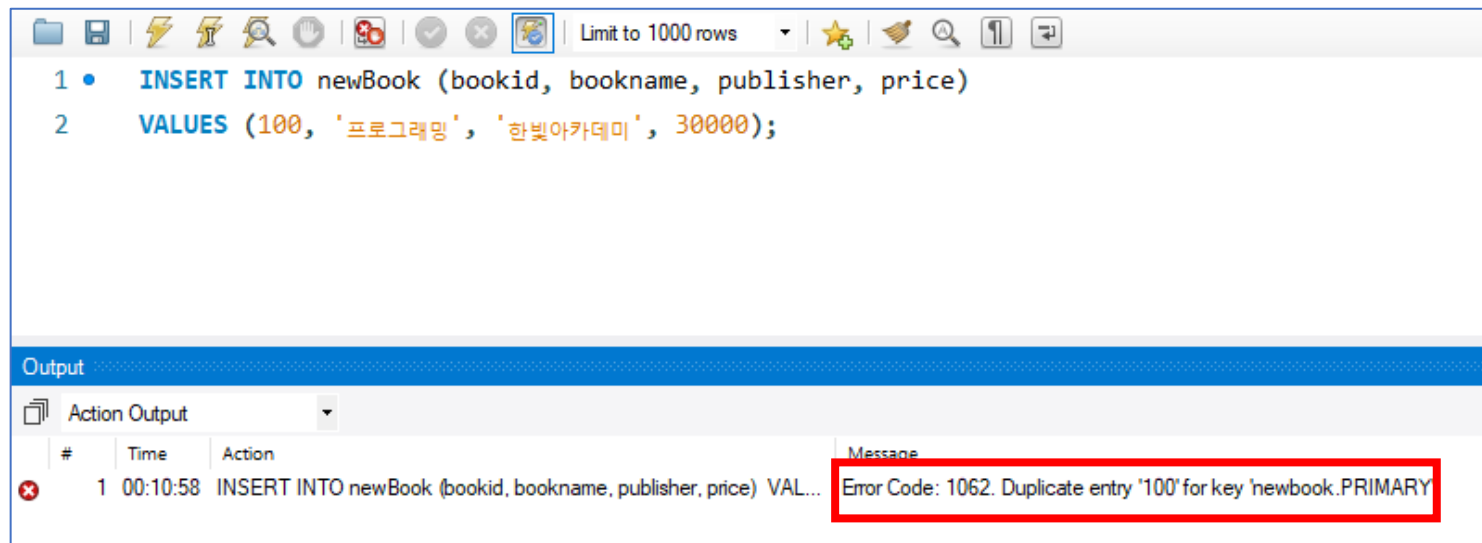


[실습] 기본키-3

❖ 데이터 삽입하기

bookid	bookname	publisher	price
100	프로그래밍	한빛아카데미	30000

```
INSERT INTO newBook (bookid, bookname, publisher, price)
VALUES (100, '프로그래밍', '한빛아카데미', 30000);
```

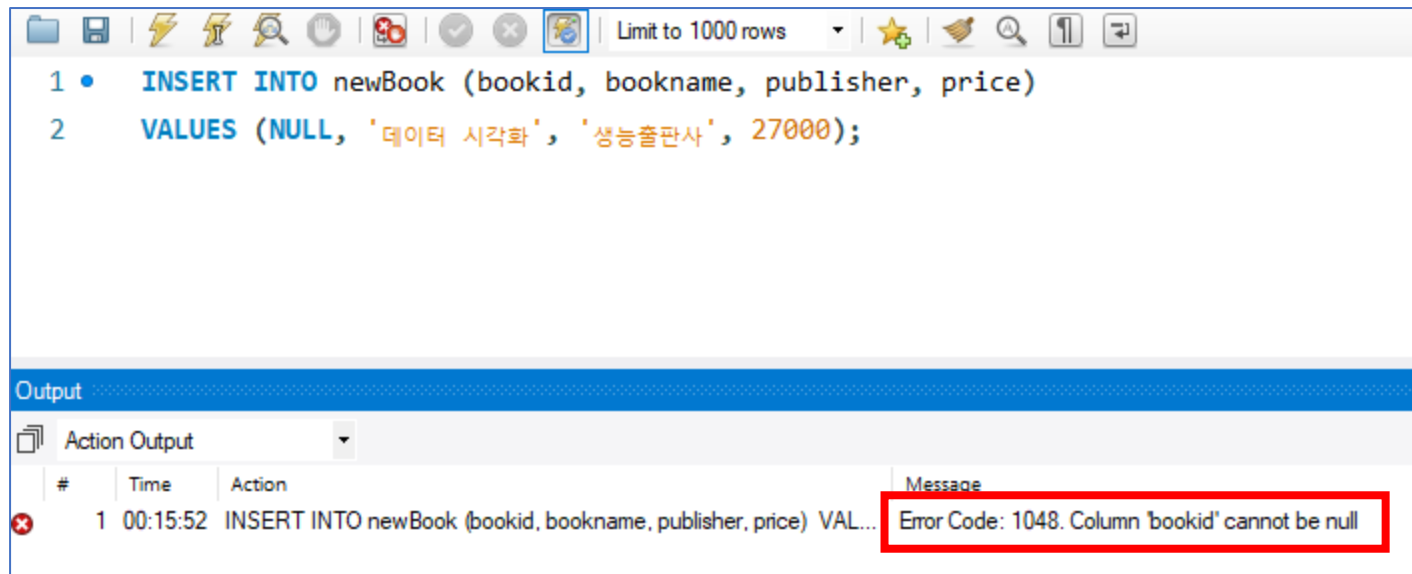


[실습] 기본키-4

❖ 데이터 삽입하기

bookid	bookname	publisher	price
	데이터 시각화	생능출판사	27000

```
INSERT INTO newBook (bookid, bookname, publisher, price)
VALUES (NULL, '데이터 시각화', '생능출판사', 27000);
```





02

기타 제약조건

NULL 제약 조건

❖ NULL

- 아직 지정되지 않은 값
- 0, ""(빈문자), ' '(공백) 등과 다른 특별한 값

❖ NOT NULL

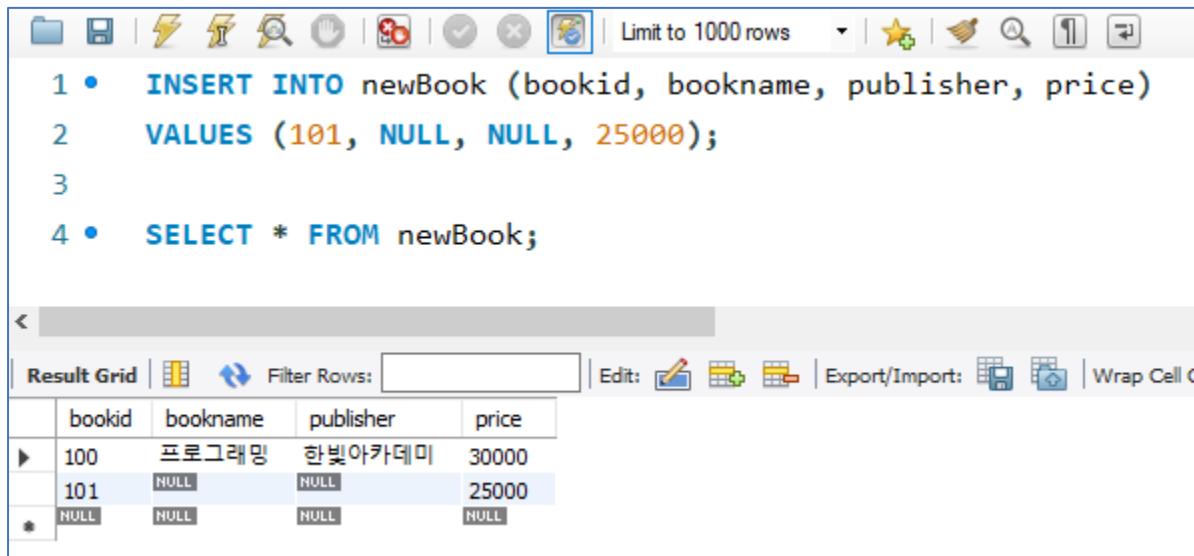
- NULL 값을 허용하지 않음

NULL 제약 조건

❖ 데이터 삽입하기

bookid	bookname	publisher	price
101			25000

```
INSERT INTO newBook (bookid, bookname, publisher, price)
VALUES (101, NULL, NULL, 25000);
```



The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • INSERT INTO newBook (bookid, bookname, publisher, price)
2   VALUES (101, NULL, NULL, 25000);
3
4 • SELECT * FROM newBook;
```

Below the editor is a 'Result Grid' section. It features a 'Filter Rows' input field and buttons for 'Edit', 'Export/Import', and 'Wrap Cell'. The grid displays the following data:

	bookid	bookname	publisher	price
▶	100	프로그래밍	한빛아카데미	30000
	101	NULL	NULL	25000
*	NULL	NULL	NULL	NULL

[실습] NOT NULL

❖ 테이블 속성 변경하기

- bookname, publisher 속성에 NOT NULL 제약 조건 추가

```
ALTER TABLE newBook MODIFY bookname VARCHAR(20) NOT NULL;
```

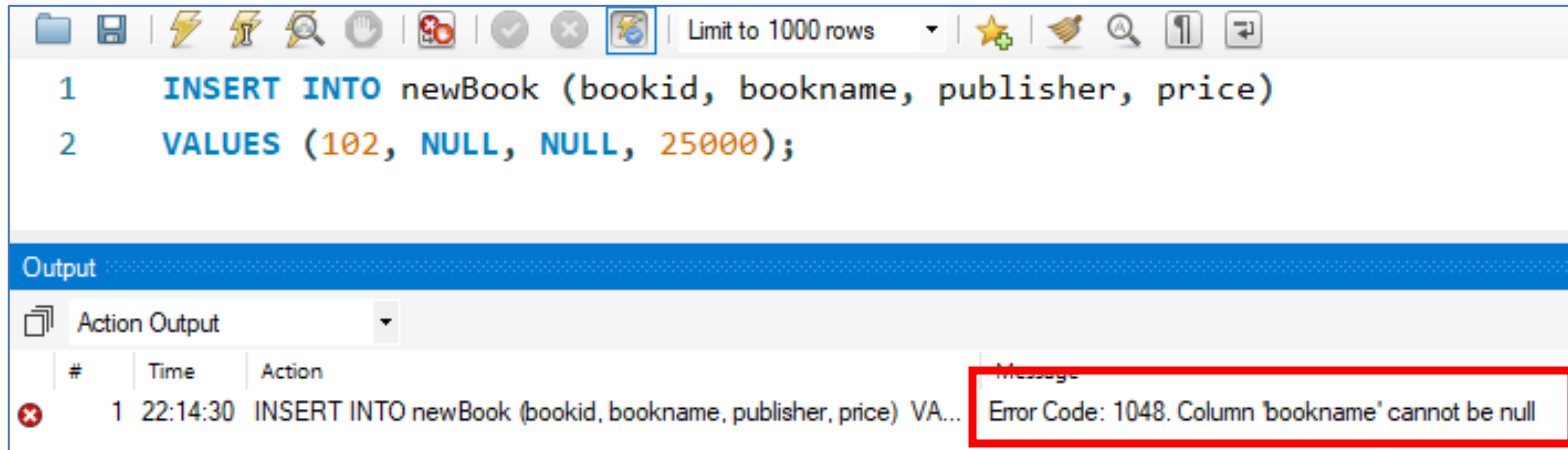
```
ALTER TABLE newBook MODIFY publisher VARCHAR(20) NOT NULL;
```


[실습] NOT NULL

❖ 데이터 삽입하기

bookid	bookname	publisher	price
102			25000

```
INSERT INTO newBook (bookid, bookname, publisher, price)
VALUES (102, NULL, NULL, 25000);
```



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains two lines of code: `1 INSERT INTO newBook (bookid, bookname, publisher, price)` and `2 VALUES (102, NULL, NULL, 25000);`. Below the editor is an 'Output' panel with a tab labeled 'Action Output'. It displays a table with columns '#', 'Time', 'Action', and 'Message'. The first row shows an error: a red 'x' icon, the number '1', the time '22:14:30', the truncated SQL statement 'INSERT INTO newBook (bookid, bookname, publisher, price) VA...', and the message 'Error Code: 1048. Column 'bookname' cannot be null'. This message is highlighted with a red rectangular box.

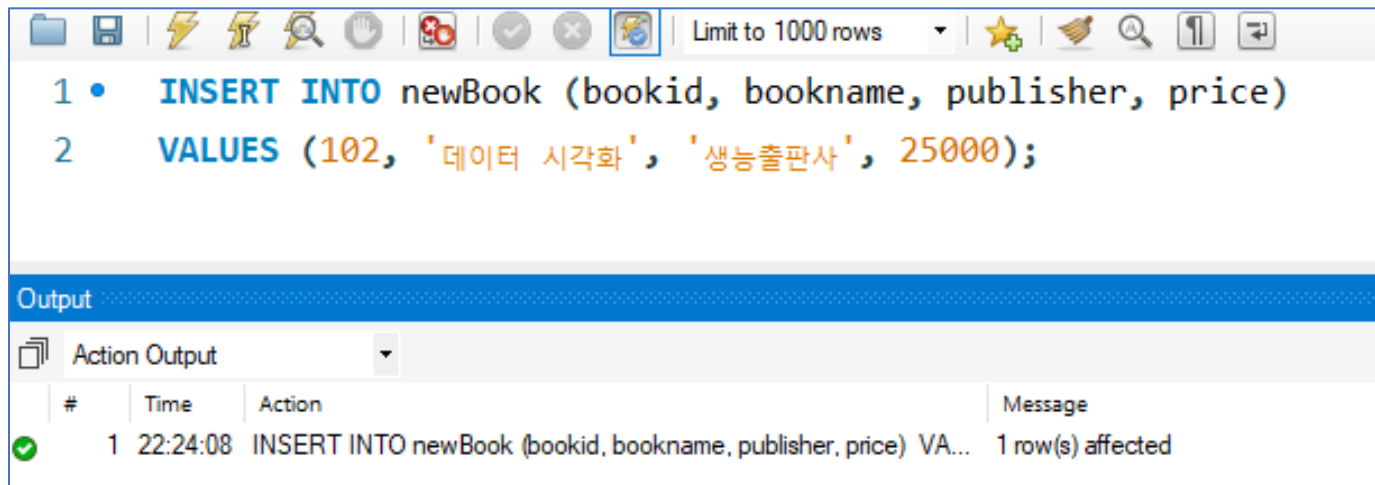
#	Time	Action	Message
1	22:14:30	INSERT INTO newBook (bookid, bookname, publisher, price) VA...	Error Code: 1048. Column 'bookname' cannot be null

[실습] NOT NULL

❖ 데이터 삽입하기

bookid	bookname	publisher	price
102	데이터 시각화	생능출판사	25000

```
INSERT INTO newBook (bookid, bookname, publisher, price)
VALUES (102, '데이터 시각화', '생능출판사', 25000);
```



DEFAULT

❖ DEFAULT

- 기본값을 지정하는 키워드
- 속성에 기본값을 지정해두지 않은 경우, 해당 속성에 NULL 값이 기본으로 저장됨
 - DEFAULT 키워드로 기본 값을 명확하게 지정할 수 있음

[실습] DEFAULT

❖ 테이블 속성 변경하기

- price 속성의 기본값을 10,000원으로 지정

```
ALTER TABLE newBook MODIFY price INT DEFAULT 10000;
```

[실습] DEFAULT

❖ 데이터 삽입하기

bookid	bookname	publisher	price
103	프로그래밍	한빛아카데미	

```
INSERT INTO newBook (bookid, bookname, publisher)
VALUES (103, '프로그래밍', '한빛아카데미');
```

The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, search, and execution, along with a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • INSERT INTO newBook (bookid, bookname, publisher)
2   VALUES (103, '프로그래밍', '한빛아카데미');
3
4 • SELECT * FROM newBook;
```

Below the editor is the 'Result Grid' section. It has a 'Filter Rows' input field and buttons for 'Edit', 'Export/Import', and 'Wrap C'. The grid displays the following data:

	bookid	bookname	publisher	price
▶	102	데이터 시각화	생능출판사	35000
	103	프로그래밍	한빛아카데미	10000
*	NULL	NULL	NULL	NULL

The value '10000' in the price column for bookid 103 is highlighted with a red rectangle.



03

외래키

newBook 테이블에 데이터 입력하기

❖ 기존 데이터를 모두 제거 후, 입력

```
DELETE FROM newBook;
```

```
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (100, '데이터베이스', '한빛아카데미', 27000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (101, '파이썬', '한빛아카데미', 22000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (102, 'JSP 프로그래밍', '생능출판사', 26000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (103, '자바스크립트', '길벗', 45000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (104, '데이터베이스 배우터', '생능출판사', 30000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (105, 'HTML 기초', '한빛아카데미', 37000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (106, '파이썬 데이터', '이지스퍼블리싱', 25000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (107, 'Chat GPT', '생능출판사', 29000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (108, 'ReactJS', '이지스퍼블리싱', 41000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (109, '홈페이지 만들기', '한빛아카데미', 32000);  
INSERT INTO newBook (bookid, bookname, publisher, price) VALUES (110, '데이터 시각화', '생능출판사', 27000);
```

외래키

❖ 외래키(Foreign key)

- 다른 릴레이션의 기본키를 참조하는 속성

<도서 테이블>

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	22000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배우터	생능출판사	30000

기본키

<주문 테이블>

orderid	bookid	member	address
p001	102	오한솔	경기
p002	107	김현우	서울
p003	103	박홍진	부산
p004	102	김현우	서울
p005	104	문종헌	대전

기본키

외래키

외래키

❖ 외래키 작성 방법

```
CREATE TABLE 테이블이름(  
    속성이름1 데이터타입 PRIMARY KEY,  
    속성이름2 데이터타입,  
    속성이름2 데이터타입,  
    FOREIGN KEY(속성이름2) REFERENCES 참조할테이블이름(속성이름)  
);
```

[실습] 외래키-1

❖ newOrders 테이블 생성하기

- bookid 속성을 외래키로 지정

```
CREATE TABLE newOrders(  
   orderid    VARCHAR(10)    PRIMARY KEY,  
    bookid    INT            NOT NULL,  
    member    VARCHAR(10)    NOT NULL,  
    address    VARCHAR(20)    NOT NULL,  
    FOREIGN KEY(bookid) REFERENCES newBook(bookid)  
);
```

[실습] 외래키-2

❖ 데이터 삽입하기

orderid	bookid	member	address
p001	102	정수아	서울

```
INSERT INTO newOrders(orderid, bookid, member, address)
VALUES ('p001', 102, '정수아', '서울');
```

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1  /* newOrders 테이블에 데이터 입력 */
2  • INSERT INTO newOrders(orderid, bookid, member, address)
3  VALUES ('p001', 102, '정수아', '서울');
4
5  /* newOrders 테이블 데이터 조회 */
6  • SELECT * FROM newOrders;
```

Below the editor, the 'Result Grid' is displayed. It shows a table with the following data:

	orderid	bookid	member	address
▶	p001	102	정수아	서울
*	NULL	NULL	NULL	NULL

[실습] 외래키-3

❖ 데이터 삽입하기

orderid	bookid	member	address
p002	120	정수아	서울

```
INSERT INTO newOrders(orderid, bookid, member, address)
VALUES ('p002', 120, '정수아', '서울');
```

1 /* newOrders 테이블에 데이터 입력 */

2 INSERT INTO newOrders(orderid, bookid, member, address)

3 VALUES ('p002', 120, '정수아', '서울');

Output

Action Output

#	Time	Action	Message
1	16:15:03	INSERT INTO new...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('bookdb'.neworders', CONSTRAINT 'neworders_ibfk_1' FOREIGN KEY (bookid) REFERENCES bookdb.books (bookid)).

에러 이유 : 외래키에 입력되는 값은 참조테이블에 입력된 값만 가능

[실습] 외래키-4

❖ 참조 테이블의 데이터 삭제하기

bookid	bookname	publisher	price
102	JSP 프로그래밍	생능출판사	26000

```
DELETE FROM newBook WHERE bookid=102;
```

The screenshot shows a database management tool interface. At the top, there's a toolbar with various icons and a dropdown menu set to 'Limit to 1000 rows'. Below the toolbar, a SQL query is entered in a text area: `1 • DELETE FROM newBook WHERE bookid=102;`. Below the query area is an 'Output' panel. The 'Output' panel has a dropdown menu set to 'Action Output'. It displays a table with columns: '#', 'Time', 'Action', and 'Message'. The first row in the table shows an error: '# 1', 'Time 11:30:39', 'Action DELETE FROM newBook ...', and 'Message Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('bookdb'.neworder...'. The 'Message' column is highlighted with a red rectangle.

에러 이유 : newOrders 테이블에서 bookid 속성을 참조하기 때문

newOrders 테이블에 데이터 입력하기

❖ 기존 데이터를 모두 제거 후, 입력

```
DELETE FROM newOrders;
```

```
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p001', 102, '오한솔', '경기');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p002', 107, '김현우', '서울');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p003', 103, '박홍진', '부산');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p004', 102, '김현우', '서울');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p005', 104, '문종헌', '대전');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p006', 105, '김현우', '서울');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p007', 103, '이봉림', '부산');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p008', 102, '정희성', '경기');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p009', 107, '오한솔', '경기');  
INSERT INTO newOrders(orderid, bookid, member, address) VALUES ('p010', 103, '김현우', '서울');
```

2개의 테이블을 이용한 조인 검색

❖ 김현우 고객이 주문한 책의 제목과 가격 검색

newBook 테이블

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	22000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배움터	생능출판사	30000
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터	이지스퍼블리싱	25000
107	Chat GPT	생능출판사	29000
108	ReactJS	이지스퍼블리싱	41000
109	홈페이지 만들기	한빛아카데미	32000
110	데이터 시각화	생능출판사	27000

newOrders 테이블

orderid	bookid	member	address
p001	102	오한솔	경기
p002	107	김현우	서울
p003	103	박홍진	부산
p004	102	김현우	서울
p005	104	문종헌	대전
p006	105	김현우	서울
p007	103	이봉림	부산
p008	102	정희성	경기
p009	107	오한솔	경기
p010	103	김현우	서울

외래키

2개의 테이블을 이용한 조인 검색

a	b	c	d

A

a	b	c	d

B

```
SELECT A.a, B.d  
FROM A, B  
WHERE 조건 AND A.b = B.b;
```


2개의 테이블을 이용한 조인 검색

❖ 김현우 고객이 주문한 책의 제목과 가격 검색

bookid	bookname	publisher	price
100	데이터베이스	한빛아카데미	27000
101	파이썬	한빛아카데미	22000
102	JSP 프로그래밍	생능출판사	26000
103	자바스크립트	길벗	45000
104	데이터베이스 배우터	생능출판사	30000
105	HTML 기초	한빛아카데미	37000
106	파이썬 데이터	이지스퍼블리싱	25000
107	Chat GPT	생능출판사	29000
108	ReactJS	이지스퍼블리싱	41000
109	홈페이지 만들기	한빛아카데미	32000
110	데이터 시각화	생능출판사	27000

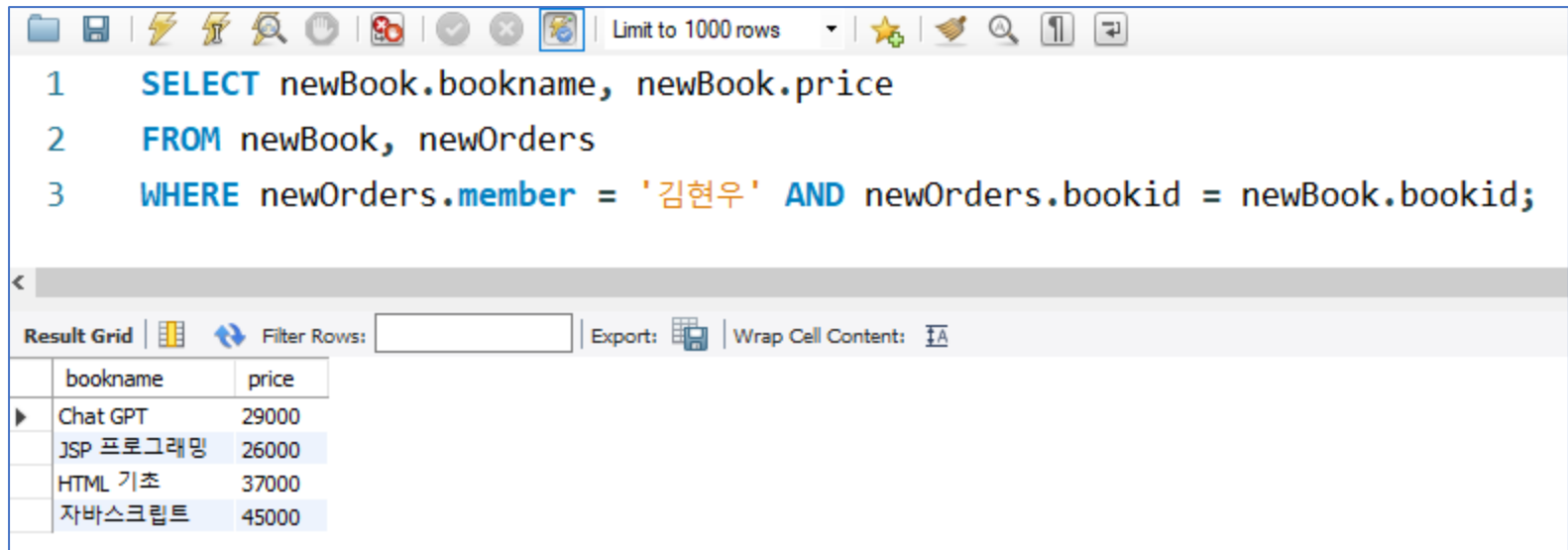
orderid	bookid	member	address
p001	102	오한솔	경기
p002	107	김현우	서울
p003	103	박홍진	부산
p004	102	김현우	서울
p005	104	문종현	대전
p006	105	김현우	서울
p007	103	이봉림	부산
p008	102	정희성	경기
p009	107	오한솔	경기
p010	103	김현우	서울



```
SELECT newBook.bookname, newBook.price
FROM newBook, newOrders
WHERE newOrders.member = '김현우'
AND
      newOrders.bookid = newBook.bookid;
```

2개의 테이블을 이용한 조인 검색

❖ 김현우 고객이 주문한 책의 제목과 가격 검색



The screenshot shows a database query tool interface. The top toolbar includes icons for file operations, a search icon, and a dropdown menu set to "Limit to 1000 rows". The SQL query is as follows:

```
1 SELECT newBook.bookname, newBook.price
2 FROM newBook, newOrders
3 WHERE newOrders.member = '김현우' AND newOrders.bookid = newBook.bookid;
```

Below the query editor, the "Result Grid" tab is active. It displays a table with two columns: "bookname" and "price". The results are as follows:

bookname	price
Chat GPT	29000
JSP 프로그래밍	26000
HTML 기초	37000
자바스크립트	45000

The interface also includes a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox.

[실습] 2개의 테이블을 이용한 조인 검색

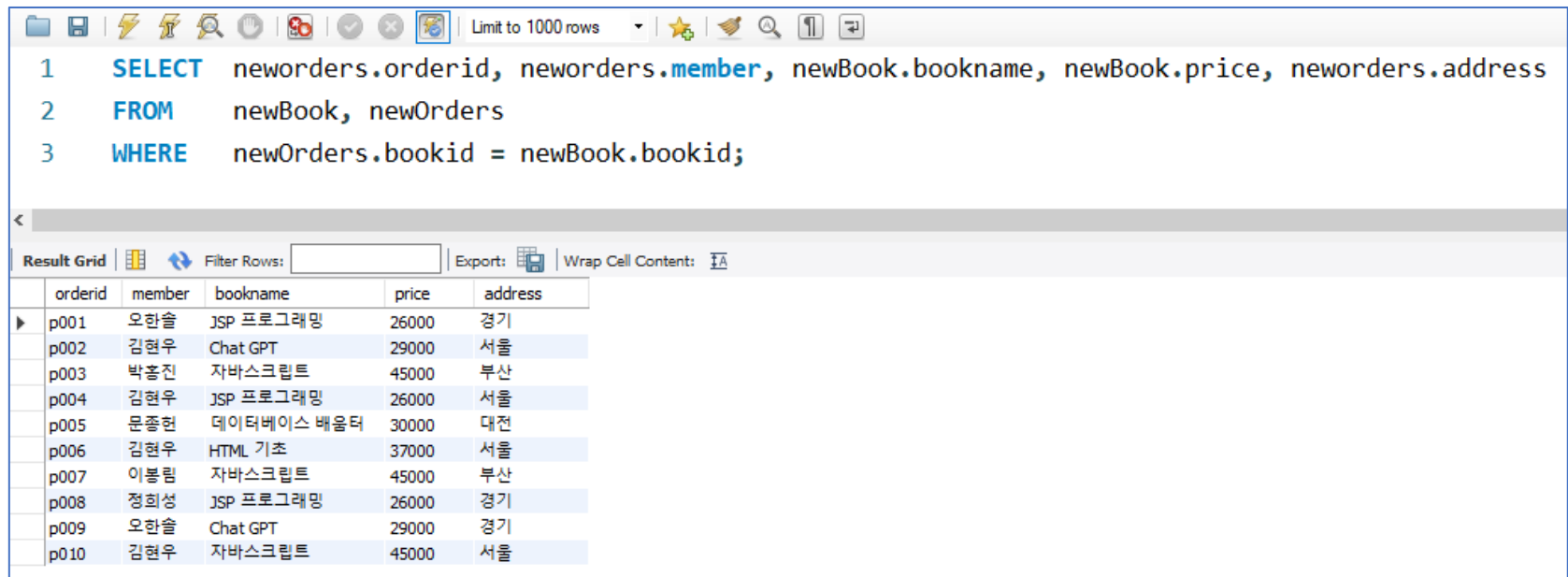
❖ 도서를 주문한 고객들의 운송장 정보 출력

orderid	member	bookname	price	address
p001	오한솔	JSP 프로그래밍	26000	경기
p002	김현우	Chat GPT	29000	서울
p003	박홍진	자바스크립트	45000	부산
p004	김현우	JSP 프로그래밍	26000	서울
p005	문종현	데이터베이스 배우터	30000	대전
p006	김현우	HTML 기초	37000	서울
p007	이봉림	자바스크립트	45000	부산
p008	정희성	JSP 프로그래밍	26000	경기
p009	오한솔	Chat GPT	29000	경기
p010	김현우	자바스크립트	45000	서울

2개의 테이블을 이용한 조인 검색

❖ 도서를 주문한 고객들의 운송장 정보 출력

```
SELECT neworders.orderid, neworders.member,  
       newBook.bookname, newBook.price, neworders.address  
FROM   newBook, newOrders  
WHERE  newOrders.bookid = newBook.bookid;
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 SELECT neworders.orderid, neworders.member, newBook.bookname, newBook.price, neworders.address  
2 FROM   newBook, newOrders  
3 WHERE  newOrders.bookid = newBook.bookid;
```

Below the query, the results are displayed in a table with the following columns: orderid, member, bookname, price, and address. The results show 10 rows of data.

orderid	member	bookname	price	address
p001	오한솔	JSP 프로그래밍	26000	경기
p002	김현우	Chat GPT	29000	서울
p003	박홍진	자바스크립트	45000	부산
p004	김현우	JSP 프로그래밍	26000	서울
p005	문종현	데이터베이스 배우기	30000	대전
p006	김현우	HTML 기초	37000	서울
p007	이봉림	자바스크립트	45000	부산
p008	정희성	JSP 프로그래밍	26000	경기
p009	오한솔	Chat GPT	29000	경기
p010	김현우	자바스크립트	45000	서울



04

내장함수

내장 함수

❖ 내장 함수

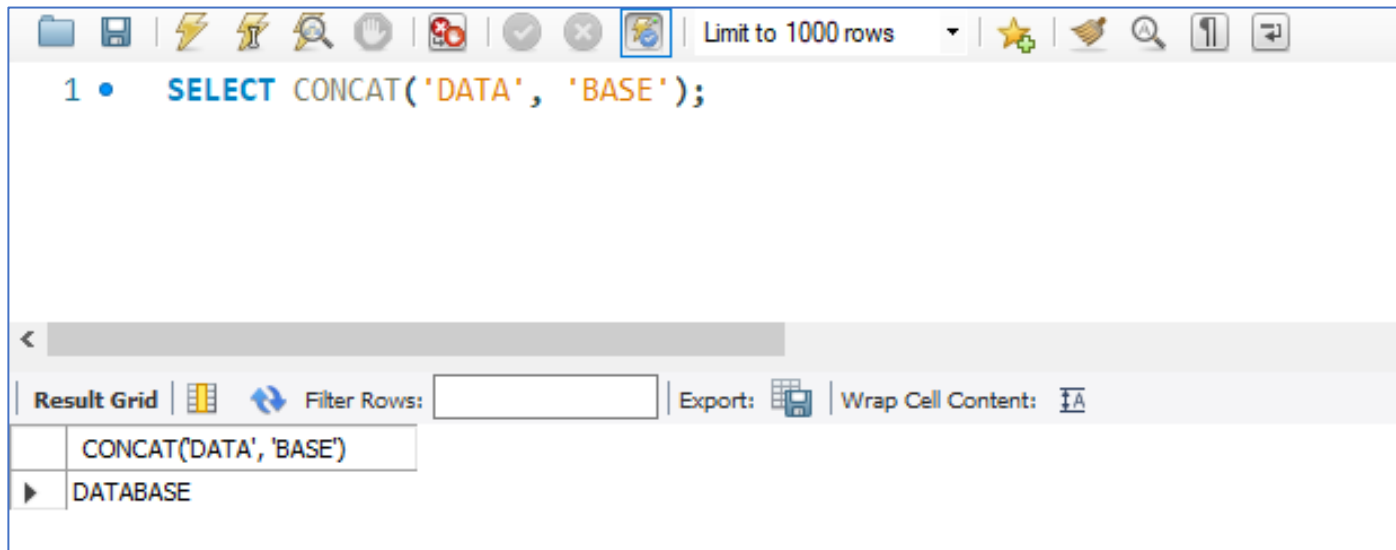
- SQL에서 제공하는 데이터를 검색할 때 유용한 함수
- 종류
 - 문자열 함수
 - 날짜 함수
 - 수학 함수

문자열 함수

❖ 문자열 연결하기

- CONCAT 함수

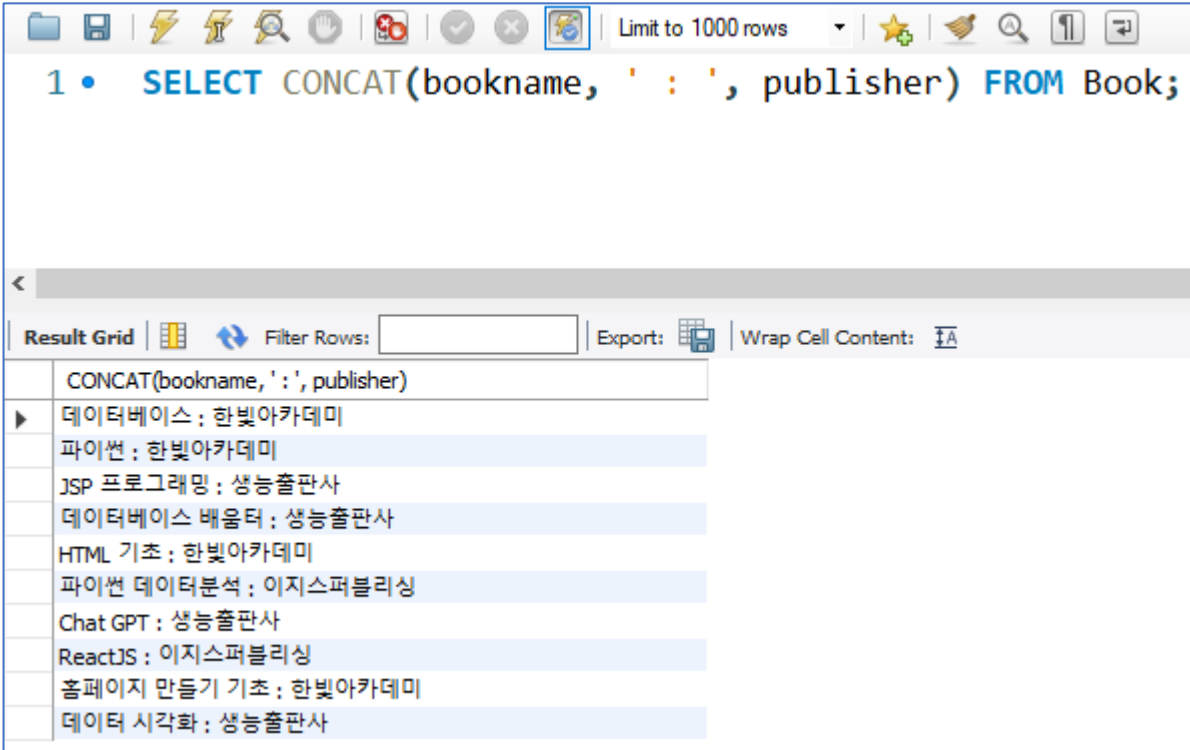
```
SELECT CONCAT('DATA', 'BASE');
```



[실습] 문자열 연결하기

❖ bookname과 publisher 연결하기

```
SELECT CONCAT(bookname, ' : ', publisher) FROM Book;
```



Limit to 1000 rows

1 • `SELECT CONCAT(bookname, ' : ', publisher) FROM Book;`

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [I A](#)

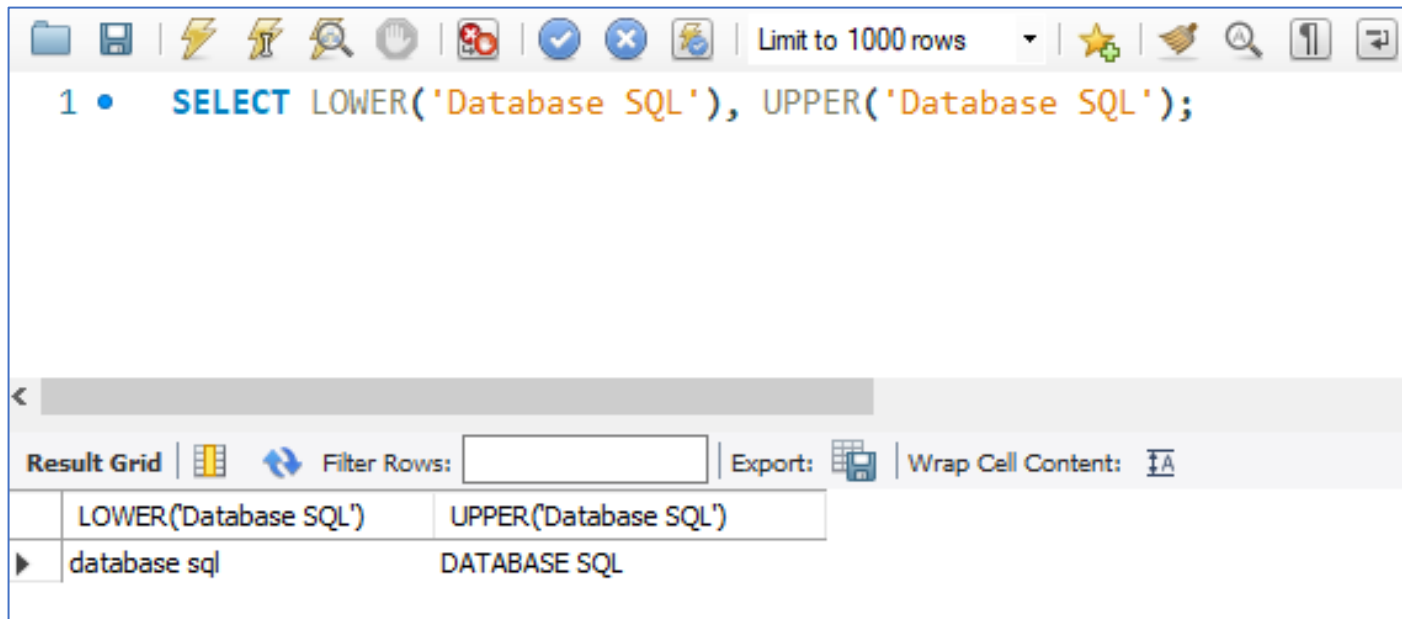
CONCAT(bookname, ' : ', publisher)
데이터베이스 : 한빛아카데미
파이썬 : 한빛아카데미
JSP 프로그래밍 : 생능출판사
데이터베이스 배우터 : 생능출판사
HTML 기초 : 한빛아카데미
파이썬 데이터분석 : 이지스퍼블리싱
Chat GPT : 생능출판사
ReactJS : 이지스퍼블리싱
홈페이지 만들기 기초 : 한빛아카데미
데이터 시각화 : 생능출판사

문자열 함수

❖ 대소문자 변경하기

- LOWER(소문자), UPPER(대문자) 함수

```
SELECT LOWER('Database SQL'), UPPER('Database SQL');
```

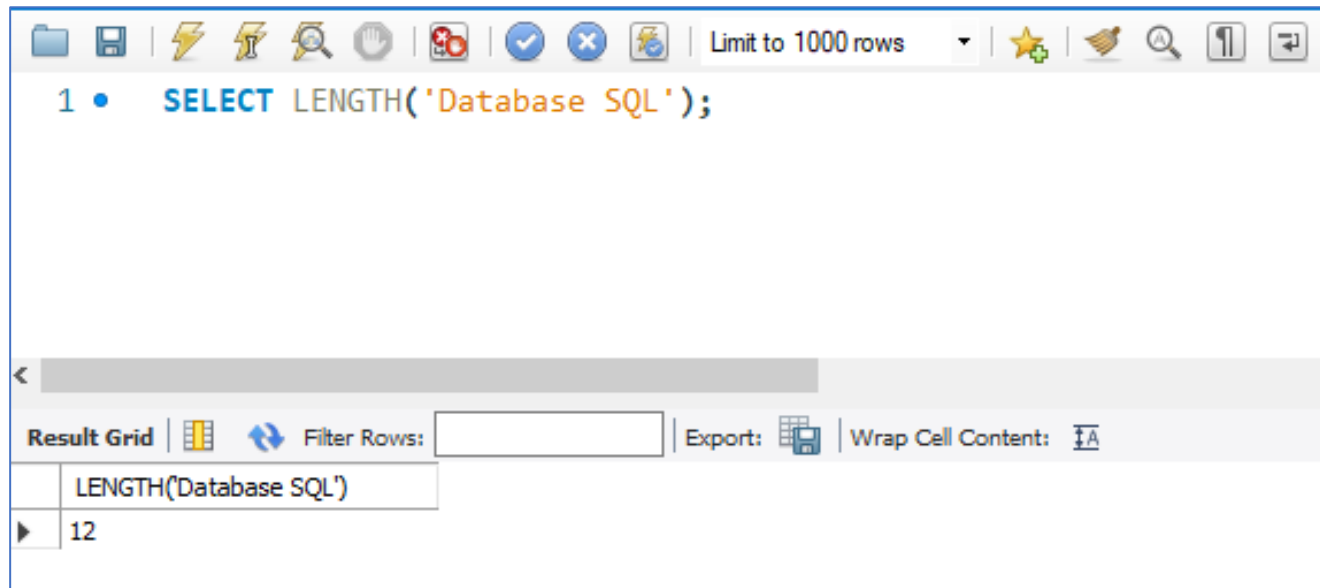


문자열 함수

❖ 영어 문자열 길이 반환하기

- LENGTH 함수

```
SELECT LENGTH('Database SQL');
```

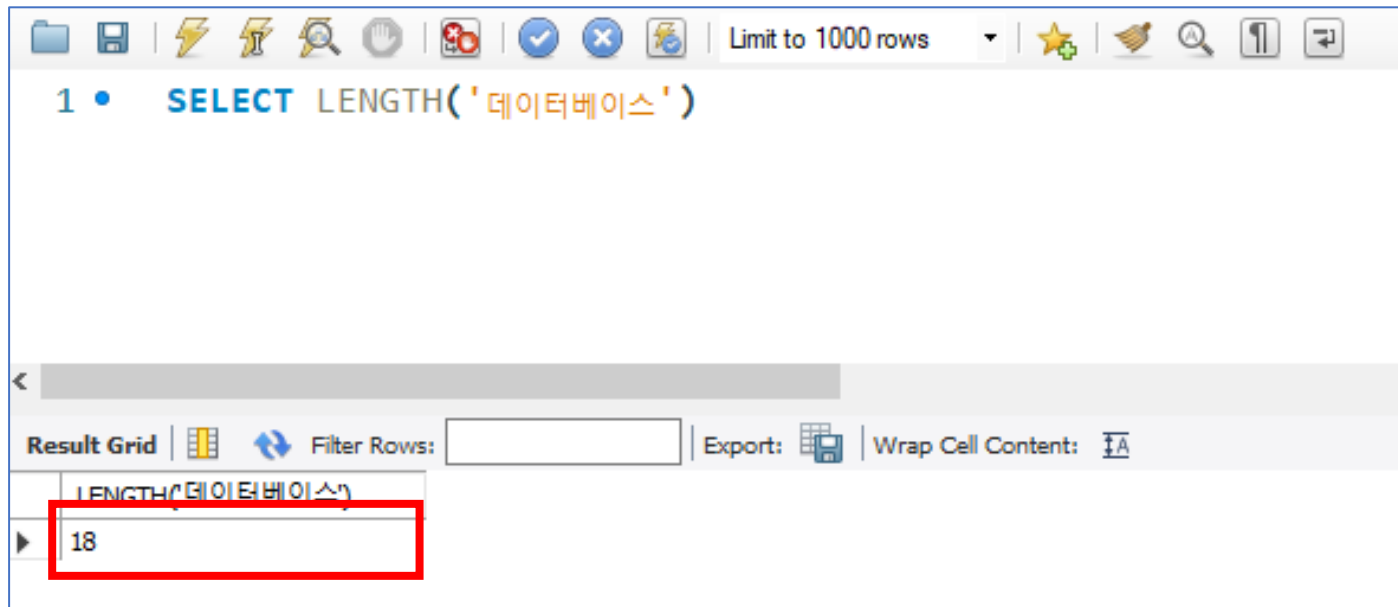


문자열 함수

❖ 한글 문자열 길이 반환하기

- LENGTH 함수

```
SELECT LENGTH('데이터베이스');
```

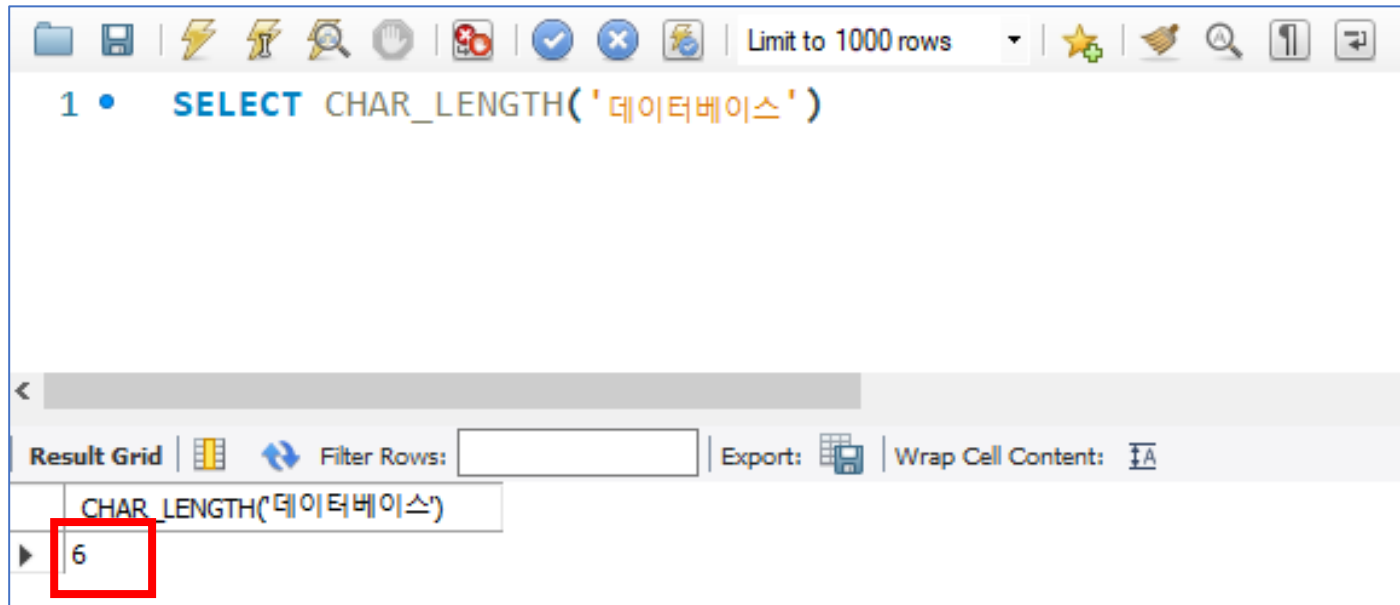


문자열 함수

❖ 정확한 문자열 길이 반환하기

- CHAR_LENGTH 함수

```
SELECT CHAR_LENGTH ('데이터베이스');
```

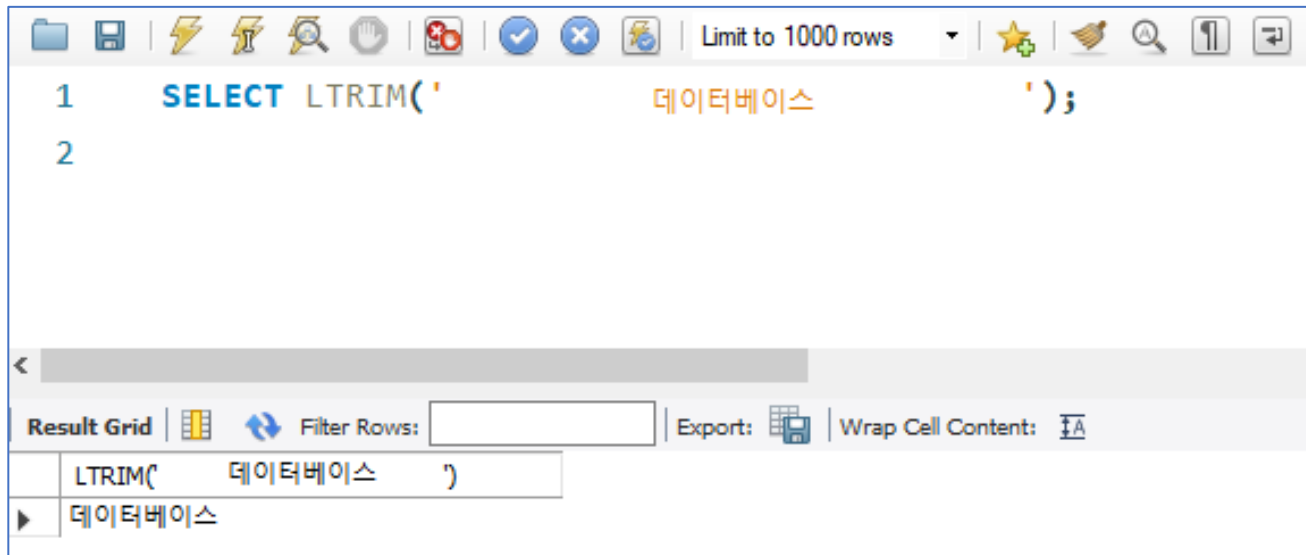


문자열 함수

❖ 문자열 왼쪽 공백 제거하기

- LTRIM 함수

```
SELECT LTRIM('    데이터베이스   ');
```

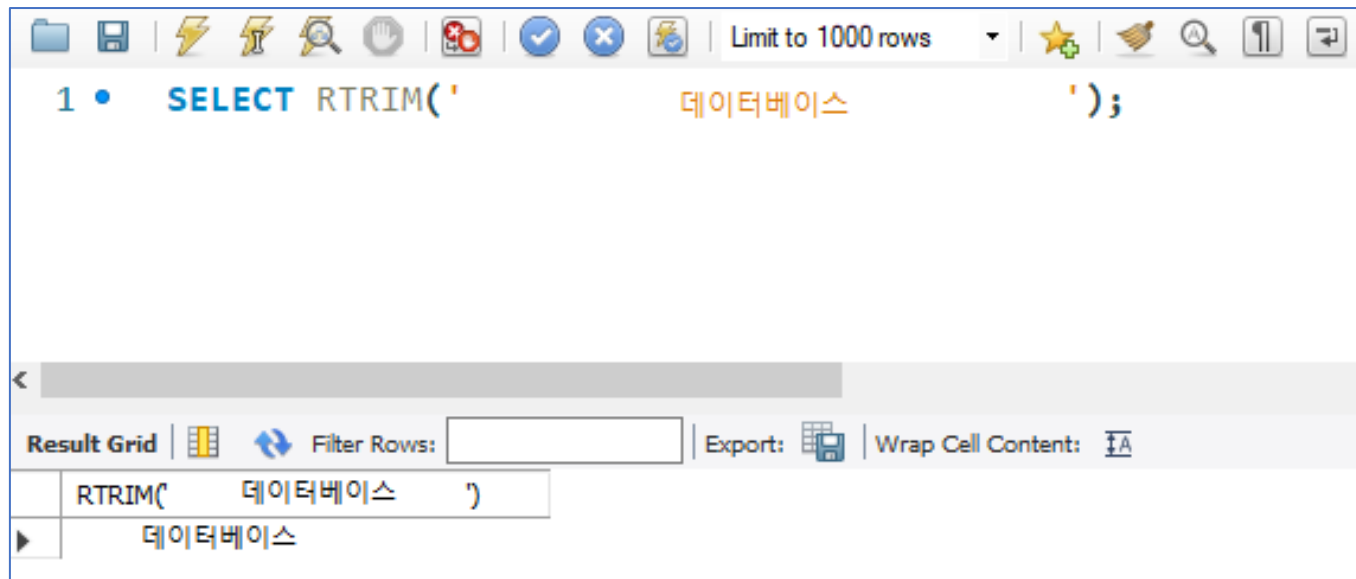


문자열 함수

❖ 문자열 오른쪽 공백 제거하기

- RTRIM 함수

```
SELECT RTRIM('   데이터베이스  ');
```

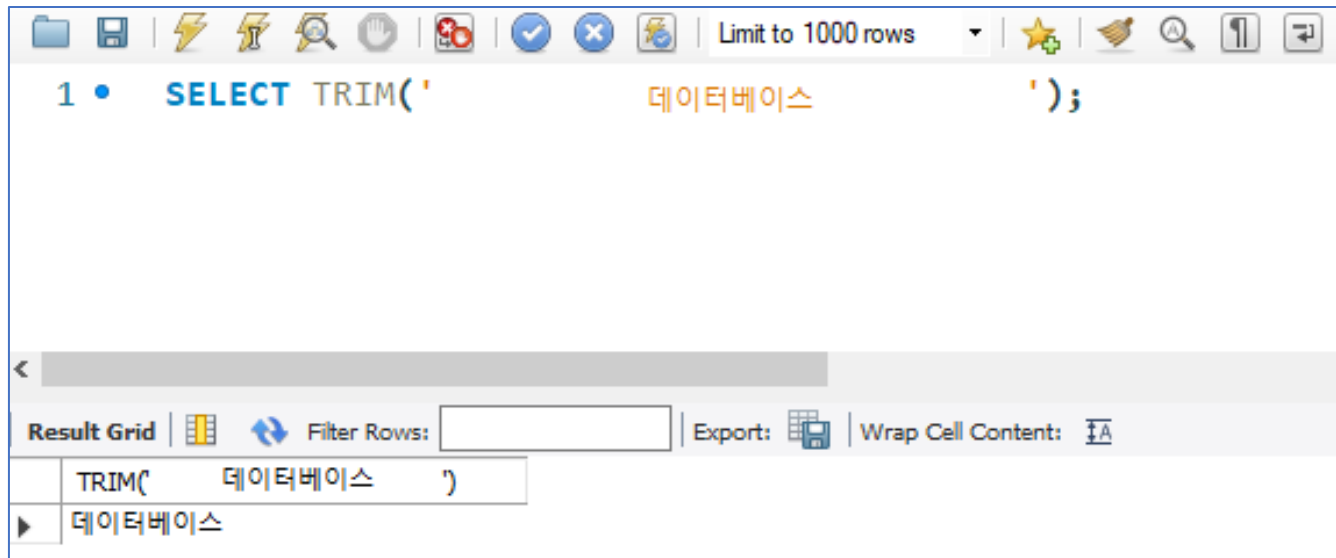


문자열 함수

❖ 문자열 양쪽 공백 제거하기

- TRIM 함수

```
SELECT TRIM('   데이터베이스  ');
```

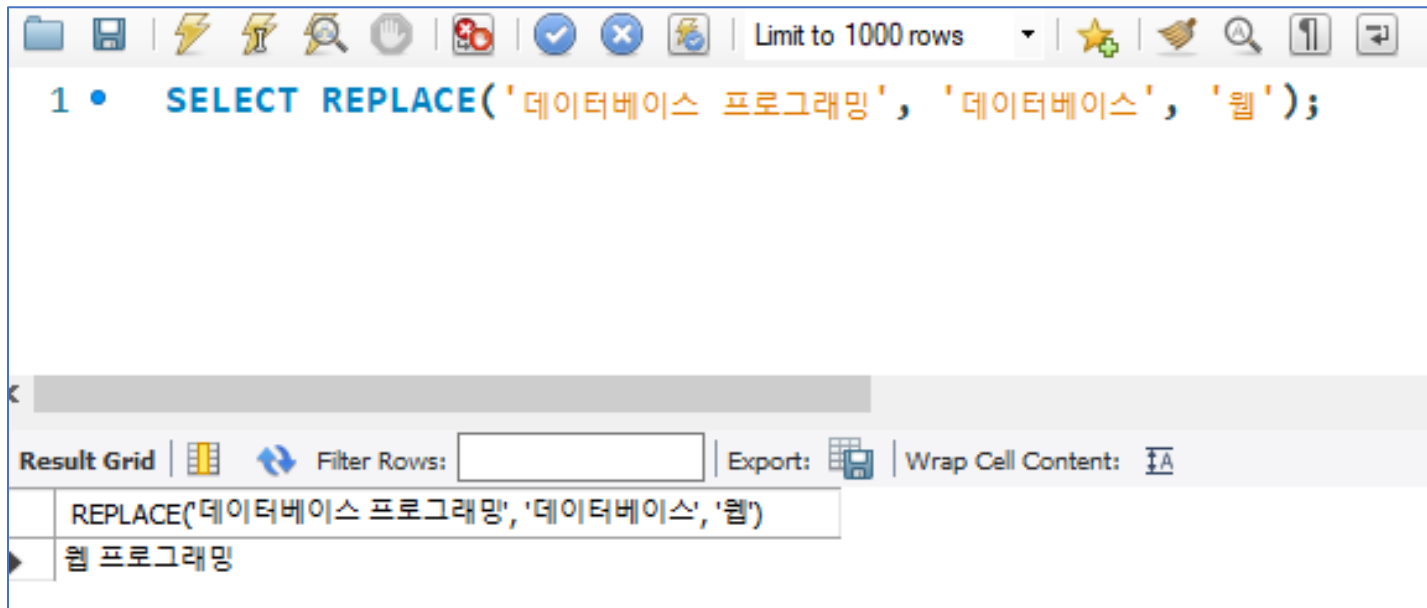


문자열 함수

❖ 문자열 변경하기

- REPLACE(열 또는 문자, 변경하려는 문자, 변경문자) 함수

```
SELECT REPLACE('데이터베이스 프로그래밍', '데이터베이스', '웹');
```

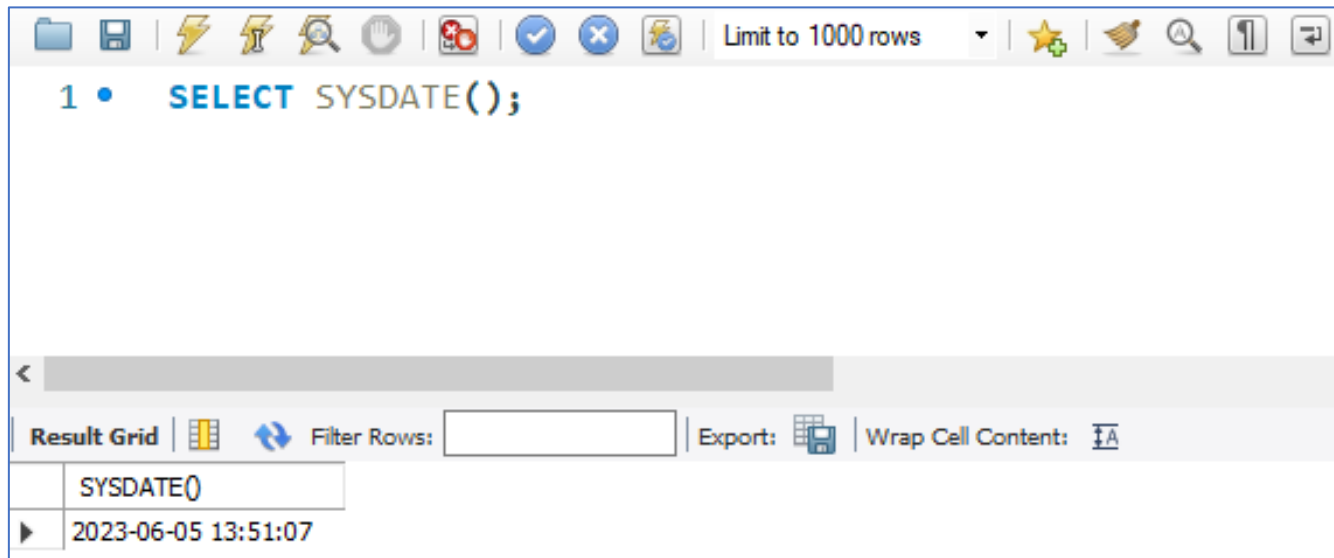


날짜 함수

❖ DBMS 시스템상의 오늘 날짜를 반환

- SYSDATE 함수

```
SELECT SYSDATE();
```

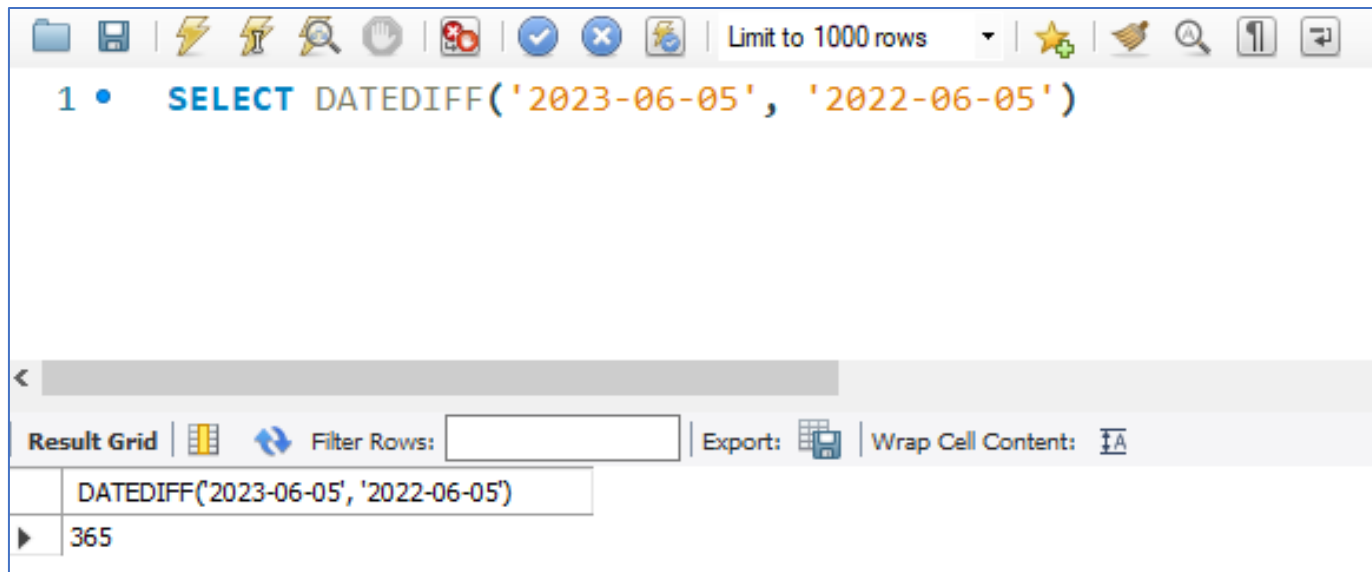


날짜 함수

❖ 날짜 차이 계산하기

- DATEDIFF(기준날짜, 비교날짜) 함수

```
SELECT DATEDIFF('2023-06-05', '2022-06-05');
```

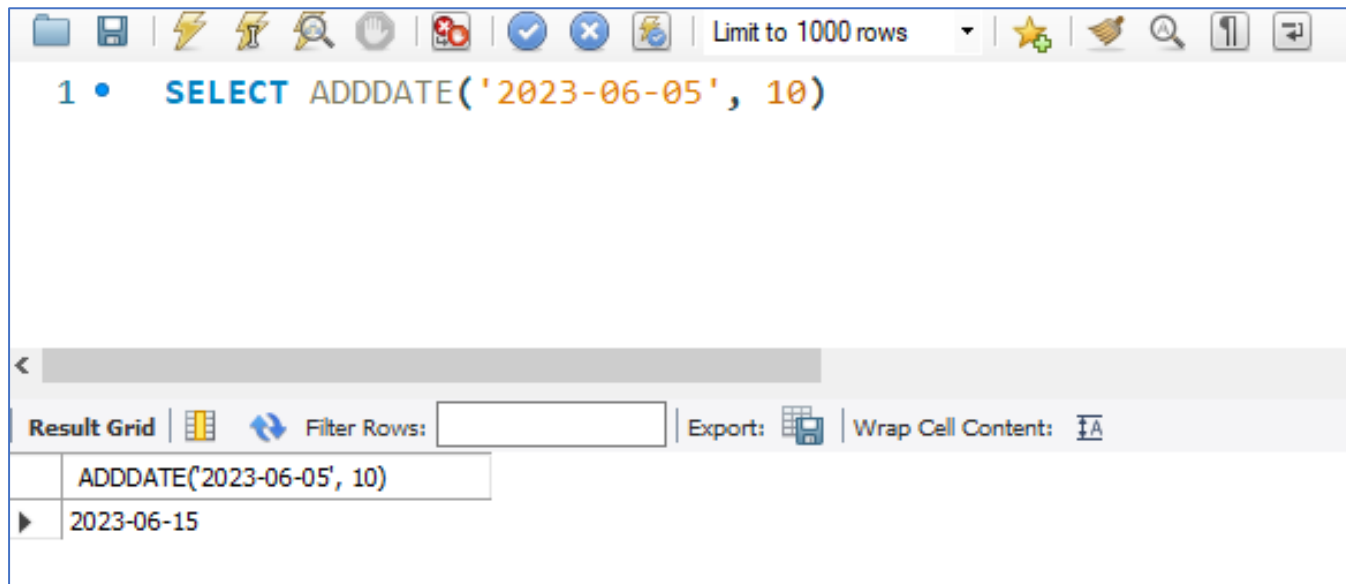


날짜 함수

❖ 날짜 더하기

- ADDDATE(기준날짜, 숫자) 함수

```
SELECT ADDDATE('2023-06-05', 10);
```

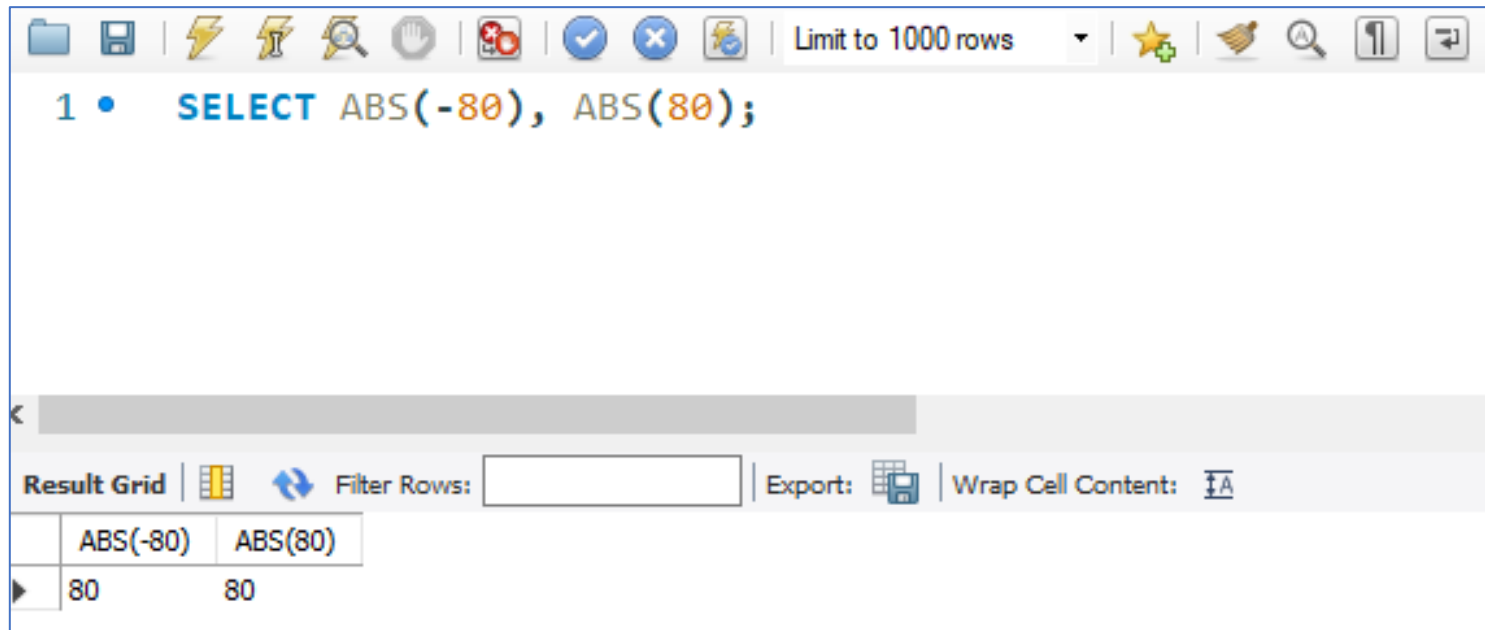


수학 함수

❖ 절대값 계산하기

- ABS(숫자) 함수

```
SELECT ABS(-80), ABS(80);
```

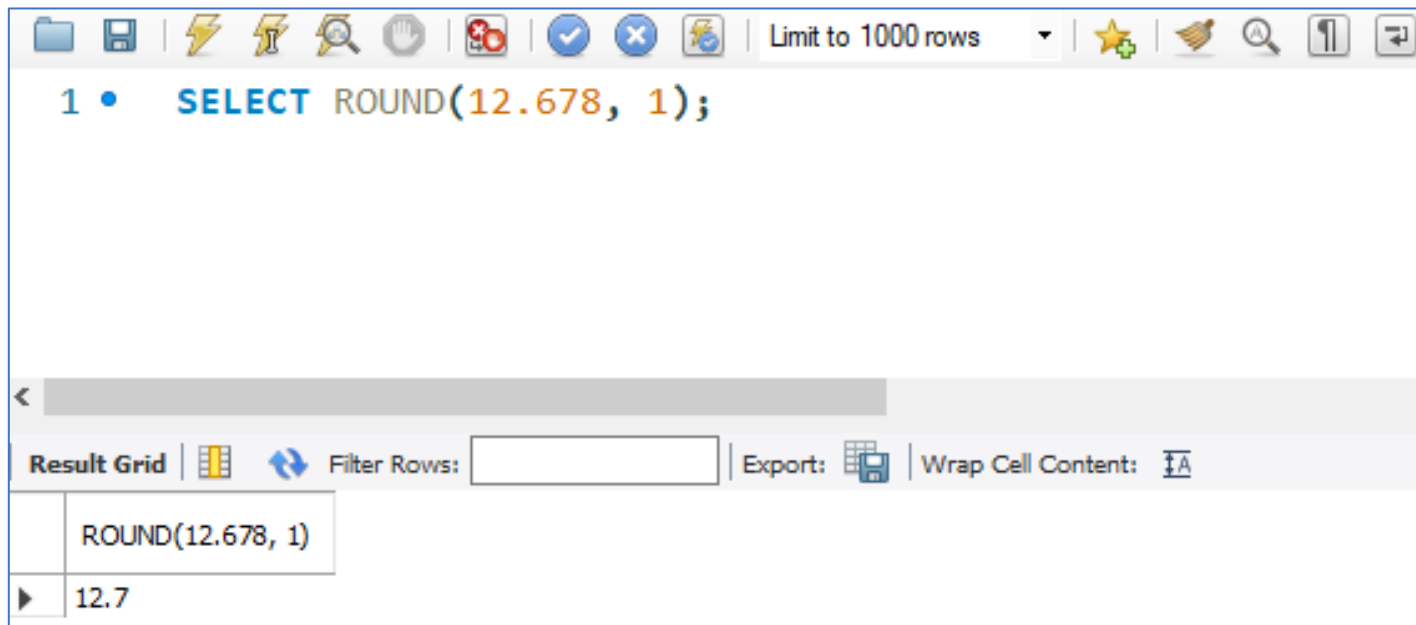


수학 함수

❖ 반올림한 값을 구하기

- ROUND(숫자, 자릿수) 함수

```
SELECT ROUND(12.678, 1);
```



THANK 😊 YOU