

마당서점의 고객이 요구하는 다음 질문에 대해 SQL 문을 작성하시오

1) 도서번호가 1인 도서의 이름

```
SELECT bookname  
FROM Book  
WHERE bookid=1;
```

2) 가격이 20,000원 이상인 도서의 이름

```
SELECT bookname  
FROM book  
WHERE price>=20000;
```

3) 박지성의 총금액

```
SELECT SUM(saleprice)  
FROM Orders  
WHERE custid=1;
```

4) 박지성이 구매한 도서의 수

```
SELECT COUNT(*)  
FROM Orders  
WHERE custid=1;
```

5) 박지성이 구매한 도서의 출판사 수

```
SELECT COUNT(publisher)  
FROM Book, Orders  
WHERE Book.bookid = Orders.bookid AND custid=1;
```

마당서점의 운영자와 경영자가 요구하는 다음 질문에 대해 SQL 문을 작성하시오

1) 마당서점 도서의 총개수

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

2) 마당서점에 도서를 출고하는 출판사의 총개수

```
SELECT COUNT(DISTINCT publisher)
```

FROM Book;

3) 모든 고객의 이름 주소

```
SELECT name, address  
FROM Customer
```

4) 2024년 7월 4일~7월 7일 사이에 주문받은 도서의 주문번호

```
SELECT ordered  
FROM Orders  
WHERE orderdate BETWEEN '2024-07-04' AND '2024-07-07';
```

4) 2024년 7월 4일~7월 7일 사이에 주문받은 도서를 제외한 도서의 주문번호

```
SELECT ordered  
FROM Orders  
WHERE orderdate NOT BETWEEN '2024-07-04' AND '2024-07-07';
```

6) 성이 '김'씨인 고객의 이름과 주소

```
SELECT name, address  
FROM Customer  
WHERE name LIKE '김%';
```

7) 성이 '김'씨이고 이름이 '아'로 끝나는 고객의 이름과 주소

```
SELECT name, address  
FROM Customer  
WHERE name LIKE '김%아';
```

8) 주문하지 않은 고객의 이름(부속질의 사용)

```
SELECT name  
FROM Customer  
WHERE NOT EXISTS(SELECT * FROM Orders WHERE Customer.custid=Orders.custid);
```

9) 주문 금액의 총액과 주문의 평균 금액

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

10) 고객의 이름과 고객별 구매액

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

11) 고객의 이름과 고객이 구매한 도서 목록

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

다음 심화 질문에 대해 SQL 문을 작성하시오

1) 박지성이 구매한 도서의 출판사와 같은 출판사에서 도서를 구매한 고객의 이름

```
SELECT DISTINCT C2.name
FROM Customer C1, Orders O1, Book B1, Book B2, Orders O2, Customer C2
WHERE C1.name = '박지성'
AND C1.custid = O1.custid
AND O1.bookid = B1.bookid
AND B1.publisher = B2.publisher
AND B2.bookid = O2.bookid
AND O2.custid = C2.custid
AND C2.name <> '박지성';
```

2) 두 개 이상의 서로 다른 출판사에서 도서를 구매한 이름

```
SELECT C.name
FROM Customer C, Orders O, Book B
WHERE C.custid = O.custid
AND O.bookid = B.bookid
GROUP BY C.custid, C.name
HAVING COUNT(DISTINCT B.publisher) >= 2;
```

3) 전체 고객의 30% 이상이 구매한 도서

```
SELECT B.bookname
```

```

FROM Orders O, Book B
WHERE O.bookid = B.bookid
GROUP BY B.bookid, B.bookname
HAVING COUNT(DISTINCT O.custid) >= (
SELECT CEIL(COUNT(*) * 0.3)
FROM Customer
);

```

다음 질의에 대해 DDL 문과 DML 문을 작성하시오

1) 새로운 도서('스포츠 세계', '대한미디어', 10000원)이 마당서점에 입고되었다. 삽입이 안 될 경우 필요한 데이터가 있는지 더 찾아보시오

```

INSERT INTO BOOK (bookid, bookname, publisher, price)
VALUES(11, '스포츠세계', '대한미디어', 10000);

```

2) '삼성당에서 출판한 도서를 삭제하시오

```

DELETE FROM Book
WHERE publisher ='삼성당';

```

3) '이상미디어'에서 출판한 도서를 삭제하시오. 삭제가 안 되면 원인을 생각해보시오

```

DELETE FROM book
WHERE publisher ='이상미디어';

```

4) 출판사 '대한미디어'를 '대한출판사'로 이름을 바꾸시오

```

UPDATE Book
SET publisher ='대한출판사'
WHERE publisher ='대한미디어'

```

5) (테이블 생성) 출판사에 대한 정보를 저장하는 테이블 Bookcompany(name, address, begin)을 생성하고자 한다. Name은 기본키이며 VARCHAR(20), address는 VARCHAR(20), begin은 DATE 타입으로 선언하여 생성하시오

```

CREATE TABLE Bookcompany(
    name VARCHAR(20) INTEGER KEY,
    address VARCHAR(20),
    begin DATE);

```

6) (테이블 수정) Bookcompany 테이블에 인터넷 주소를 저장하는 webaddress 속성을 VARCHAR(30)으로 추가하시오

```
ALTER TABLE Bookcompany ADD webaddress VARCHAR(30);
```

7) Bookcompany 테이블에 임의의 튜플 name=한빛아카데미, address=서울시 마포구, begin=1993-01-01, webaddress=http://hanbit.co.kr을 삽입하시오

```
INSERT INTO Bookcompany(name, address, begin, webaddress)
```

```
VALUES('한빛아카데미', '서울시 마포구', '1993-01-01', 'http://hanbit.co.kr');
```

다음 EXISTS 질의의 결과를 나타내시오

```
SELECT *
```

```
FROM Customer c1
```

```
WHERE NOT EXISTS (SELECT * FROM Orders c2 WHERE c1.custid=c2.custid)
```

1) 질의의 결과는 무엇인가?

주문하지 않은 사람의 정보

2) NOT을 지우면 질의의 결과는 어떻게 되는가

주문한 사람의 정보

다음 SQL 명령문들에 대하여 DML, DDL, DCL을 구분하시오

DML: SELECT, INSERT, DELETE, UPDATE,

DDL: CREATE, ALTER, DROP

DCL: GRANT, REVOKE

릴레이션 R(A, B, C)과 S(C, D, E)에 대해 내부조인과 외부조인들에 대한 SQL 문의 결과를 작성하시오

1) SELECT * FROM R INNER JOIN S ON (R.C = S.C)

R.C = S.C 조건을 만족하는 행들만 반환

2) SELECT * FROM R LEFT OUTER JOIN S ON (R.C = S.C)

R은 모두 출력, 매칭 안되는 S는 NULL

3) SELECT * FROM R RIGHT OUTER JOIN S ON (R.C = S.C)

S는 모두 출력, 매칭 안되는 R은 NULL

4) SELECT * FROM R FULL OUTER JOIN S ON (R.C = S.C)

R과 S 양쪽에 모두 있는 행을 보여주고, 매칭 안되는 쪽은 NULL

5) SELECT * FROM M CROSS JOIN S

R행 모든 S 행 조합 출력

릴레이션 R(col1: VARCHAR(20), col2:INT)에 대해, 다음 SQL 구문을 순서대로 수행하였을 때 마지막 SQL의 수행 결과를 적으시오

10

릴레이션 R1, R2, R3에 대해 다음 SQL문을 수행하였을 때 반환되는 행의 수는 얼마인가?

9행

테이블 R1(a,b), R2(a,b)를 생성하고 데이터 삽입과 삭제를 수행하였다. 마지막 SELECT 문의 수행 결과를 적으시오. 자식 테이블에 대한 참조무결성 제약 중 ON DELETE CASCADE 조건의 의미를 확인하여 결과를 구해본다.

(2,2)