데이터베이스

<SQL DB가 존재할 때 삭제>
drop databases if exist madang;
sqldb가 존재하면 삭제한다.

drop table if exist orders;
테이블이 존재할 경우 삭제한다.

```
-- book 테이블 만들기 --
create table book
(bookid int auto_increment not null primary key,
bookname varchar(20) not null,
publisher varchar(20) not null,
price int not null
);
insert into book (bookid, bookname, publisher, price) values ('1', '축구의 역사', '굿스포츠', '7000' );
insert into book (bookname, publisher, price) values ('축구아는 여자', '나무수', '13000' );
insert into book (bookname, publisher, price) values ('축구의 이해', '대한미디어', '22000' );
insert into book (bookname, publisher, price) values ('골프 바이블', '대한미디어', '35000' );
insert into book (bookname, publisher, price) values ('피겨 교본', '굿스포츠', '8000' );
insert into book (bookname, publisher, price) values ('역도 단계별기술', '굿스포츠', '6000' );
insert into book (bookname, publisher, price) values ('야구의 추억', '이상미디어', '20000' );
insert into book (bookname, publisher, price) values ('야구를 부탁해', '이상미디어', '13000' );
insert into book (bookname, publisher, price) values ('올림픽 이야기', '삼성당', '7500' );
insert into book (bookname, publisher, price) values ('Olympic Champions', 'Pearson', '13000' );
select * from book;
-- customer 테이블 만들기 --
create table customer
(custid int auto_increment not null primary key,
name varchar(20) not null,
address varchar(20) not null,
phone varchar(13)
);
select * from customer;
insert into customer (custid, name, address, phone) values ('1', '박지성', '영국 멘체스터', '000-5000-0001' );
```

```
insert into customer (name, address , phone) values ('김연아', '대한민국 서울', '000-6000-0001' );
insert into customer (name, address , phone) values ('장미란', '대한민국 강원도', '000-7000-0001' );
insert into customer (name, address , phone) values ('추신수', '미국 클리블랜드', '000-8000-0001' );
insert into customer (name, address , phone) values ('박세리', '대한민국 대전', '' );
-- orders 테이블 만들기 --
create table orders
( orderid int auto_increment not null primary key,
bookid int not null,
custid int not null,
saleprice int not null,
orderdate date not null,
foreign key(bookid) references book(bookid),
foreign key(custid) references customer(custid)
);
select * from orders;
insert into orders (orderid, custid, bookid, saleprice, orderdate) values ('1', '1', '1', '6000', '2014-07-01');
insert into orders (custid, bookid, saleprice, orderdate) values ('1', '3', '21000', '2014-07-03');
insert into orders (custid, bookid, saleprice, orderdate) values ('2', '5', '8000', '2014-07-03');
insert into orders (custid, bookid, saleprice, orderdate) values ('3', '6', '6000', '2014-07-04');
insert into orders (custid, bookid, saleprice, orderdate) values ('4', '7', '20000', '2014-07-05');
insert into orders (custid, bookid, saleprice, orderdate) values ('1', '2', '12000', '2014-07-07');
insert into orders (custid, bookid, saleprice, orderdate) values ('4', '8', '13000', '2014-07-07');
insert into orders (custid, bookid, saleprice, orderdate) values ('3', '10', '12000', '2014-07-08');
insert into orders (custid, bookid, saleprice, orderdate) values ('2', '10', '7000', '2014-07-09');
insert into orders (custid, bookid, saleprice, orderdate) values ('3', '8', '13000', '2014-07-10');
-- 김연아 고객의 전화번호를 찾으세요 --
select name, phone from customer where name = '김연아';
-- 3. 모든 도서의 이름과 가격을 찾으세요 --
```

```
select bookname, price from book;
-- 3-1. 모든 도서의 가격과 이름을 검색하시오. --
select price, bookname from book;
-- 3-2 모든 도서의 도서 번호, 도서이름, 출판사, 가격을 검색하시오.
select * from book;
-- 3-3 도서 테이블에 있는 모든 출판사를 검색하시오.
select distinct publisher from book;
-- 3-4 가격이 20,000원 미만인 도서를 검색하시오.
select * from book where price < 20000;
-- 3-5 가격이 10,000원 이상 20,000원 미만인 도서를 검색하시오.
select * from book where price between 10000 and 20000;
-- 3-6 출판사가 '굿스포츠' 혹은 '대한미디어'인 도서를 검색하시오
select * from book where publisher in ('굿스포츠', '대한미디어');
-- * 출판사가 '굿스포츠' 혹은 '대한미디어'가 아닌 도서를 검색하시오
select * from book where publisher not in ('굿스포츠', '대한미디어');
-- 3-7 축구의 역사를 출간한 출판사를 검색하시오.
select bookname, publisher from book where bookname = '축구의 역사';
select bookname, publisher from book group by '축구의 역사';
-- 3-8 도서이름에 '축구'가 포함된 출판사를 검색하시오
select bookname, publisher from book where bookname like '%축구%';
-- 3-9 도서이름의 왼쪽 두번째 위치에 '구'라는 문자열을 갖는 도서를 검색하시오.
select bookid, bookname, publisher, price from book where bookname like '_구%';
-- 3-10 축구에 관한 도서 중 가격이 20,000원 이상인 도서를 검색하시오.
select * from book where bookname like '축구%' and price >= '20000'
-- 3-11 출판사가 '굿스포츠' 혹은 '대한미디어'인 도서를 검색하시오.
select * from book where publisher in('굿스포츠', '대한미디어');
-select * from book order by bookname asc;
-- 3-13 도서를 가격순으로 검색하고, 가격이 같으면 이름순으로 검색하시오.
select * from book order by price, bookname desc;
- 3-12 도서를 이름순으로 검색하시오
-- 3-14 도서를 가격의 내림차순으로 검색하시오. 만약 가격이 같다면 출판사의 오름차순으로 검색한다.
select * from book group by bookname order by price desc, publisher asc;
-- 3-15 고객이 주문한 도서의 총 판매액을 구하시오
select SUM(saleprice) as 총매출 from Orders;
-- 3-16 2번 김연아 고객이 주문한 도서의 총 판매액을 구하시오.
select sum(saleprice) as 총매출 from Orders where custid=2;
-- 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;
-- 3-18 마당서점의 도서 판매 건수를 구하시오.
```

```
select count(*) from orders;
-- 3-19 고객별로 주문한 도서의 총 수량과 총 판매액을 구하시오.
select custid, count(*) as 도서수량, sum(saleprice)as 총액 from orders group by custid;
-- 3-20 가격이 8,000 원 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;
-- 3-21 고객과 고객의 주문에 관한 데이터를 모두 보이시오.
select * from customer, orders where customer.custid = orders.custid
order by orderid;
-- 3-22 고객과 고객의 주문에 관한 데이터를 고객번호 순으로 정렬하여 보이시오.
select * from customer, orders where customer.custid = orders.custid order by customer.custid;
-- 3-23 고객의 이름과 고객이 주문한 도서의 판매가격을 검색하시오.
select name, saleprice from customer, orders
where customer.custid = orders.custid;
-- 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;
-- 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;
-- 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 orders.saleprice = 20000;
-- 3-27 도서를 구매하지 않은 고객을 포함하여 고객의 이름과 고객이 주문한 도서의 가격을 구하시오.
select customer.name, saleprice from customer
left outer join orders on customer.custid = orders.custid;
-- 3-28 가장 비싼 도서의 이름을 보이시오.
select bookname from book
where price = (select max(price) from book);
-- 3-29 도서를 구매한 적이 있는 고객의 이름을 검색하시오.
select name from customer where custid in (select custid from orders);
-- 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'));
```

```
-- 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);
-- 질의 3-32 대한민국에서 거주하는 고객의 이름과 도서를 주문한 고객의 이름을 보이시오.
select name from customer where address like '대한민국%'
union
select C.name from orders O inner join customer C on C.custid = O.custid;
```