데이터베이스개론 과제02 컴퓨터소프트웨어공학과 1-YA 20202296 전채린

1) 테이블 생성한 SQL 소스

-- (hw02.sql)

use hwdb;

drop table 재고현황;

drop table 주문;

-- 테이블 생성

-- 재고현황(제품코드, 제품명, 입고량, 출고량, 현재수량, 단가)

create table 재고현황 (

제품코드 char(4) not null ,

제품명 varchar(10) ,

입고량 int ,

출고량 int ,

현재수량 int ,

단가 int ,

constraint pk\_재고현황\_제품코드 primary key(제품코드)

);

-- 주문(주문번호, 제품번호, 수량, 주문자, 주문날짜, 적립금)

create table 주문 (

주문번호 char(4) not null ,

제품번호 char(4) ,

수량 int ,

주문자 varchar(5) ,

주문날짜 varchar(20) ,

적립금 int ,

constraint pk\_주문\_주문번호 primary key(주문번호) ,

constraint fk\_주문\_제품번호 foreign key(제품번호) references 재고현황(제품코드)

);

2) 데이터 입력한 SQL 소스

-- 데이터 입력

-- 재고현황(제품코드, 제품명, 입고량, 출고량, 현재수량, 단가)

insert into 재고현황 values('a001', '삼성전자에어컨', 100, 50, 50, 1300000);

insert into 재고현황 values('a002', 'LG전자에어컨', 70, 40, 30, 1400000);

insert into 재고현황 values('a003', '캐리어에어컨', 50, 25, 25, 300000);

insert into 재고현황 values('a004', '위니아딤채에어컨', 40, 15, 25, 400000);

insert into 재고현황 values('a005', '파세코에어컨', 50, 40, 10, 700000);

-- 주문(주문번호, 제품번호, 수량, 주문자, 주문날짜, 적립금)

insert into 주문 values('b001', 'a001', 10, '전채린', '2020-10-10', 100000);

insert into 주문 values('b002', 'a003', 5, '전성용', '2020-10-10', NULL);

insert into 주문 values('b003', 'a005', 10, '이바다', '2020-10-11', 100000);

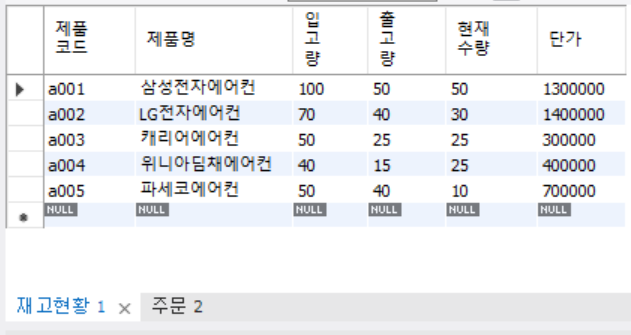
insert into 주문 values('b004', 'a001', 7, '이산', '2020-10-12', 70000);

insert into 주문 values('b005', 'a002', 20, '김보아', '2020-10-12', 200000);

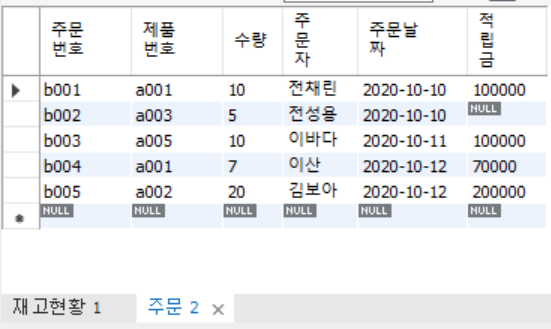
3) 데이터 조회 결과

-- 데이터 조회

select \* from 재고현황;



select \* from 주문;



4) SELECT 문제, 답안, 실행결과

**[재고현황 SELECT 문제 및 답안]**

-- 재고현황(제품코드, 제품명, 입고량, 출고량, 현재수량, 단가)

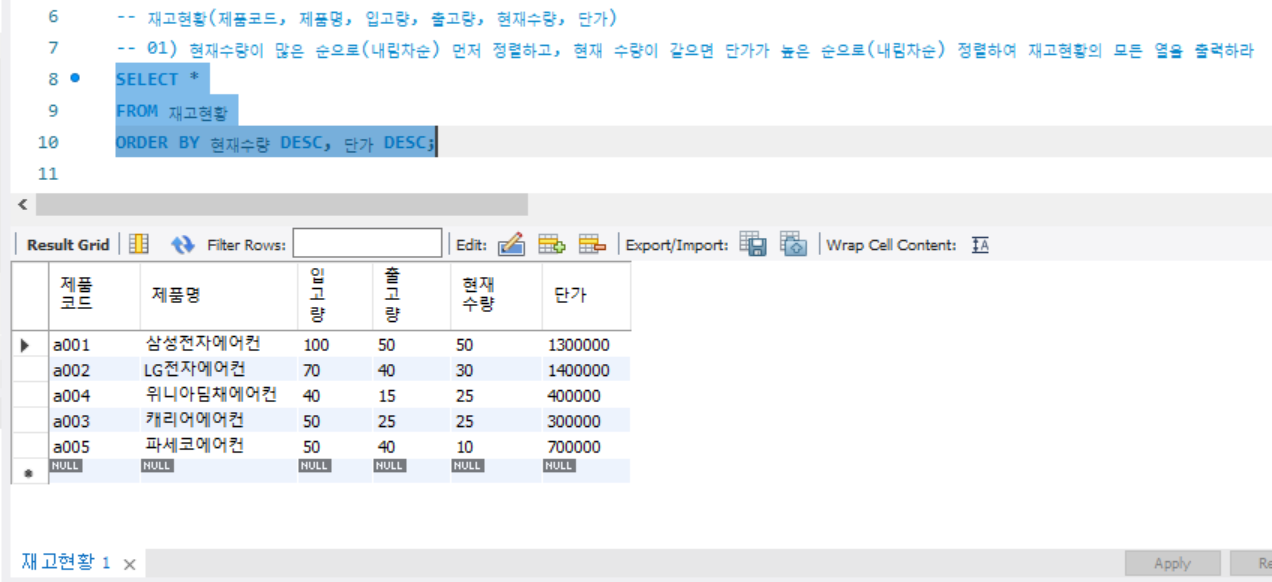
-- 01) 현재수량이 많은 순으로(내림차순) 먼저 정렬하고, 현재 수량이 같으면 단가가 높은 순으로(내림차순) 정렬하여 재고현황의 모든 열을 출력하라

SELECT \*

FROM 재고현황

ORDER BY 현재수량 DESC, 단가 DESC;

**[실행결과]**



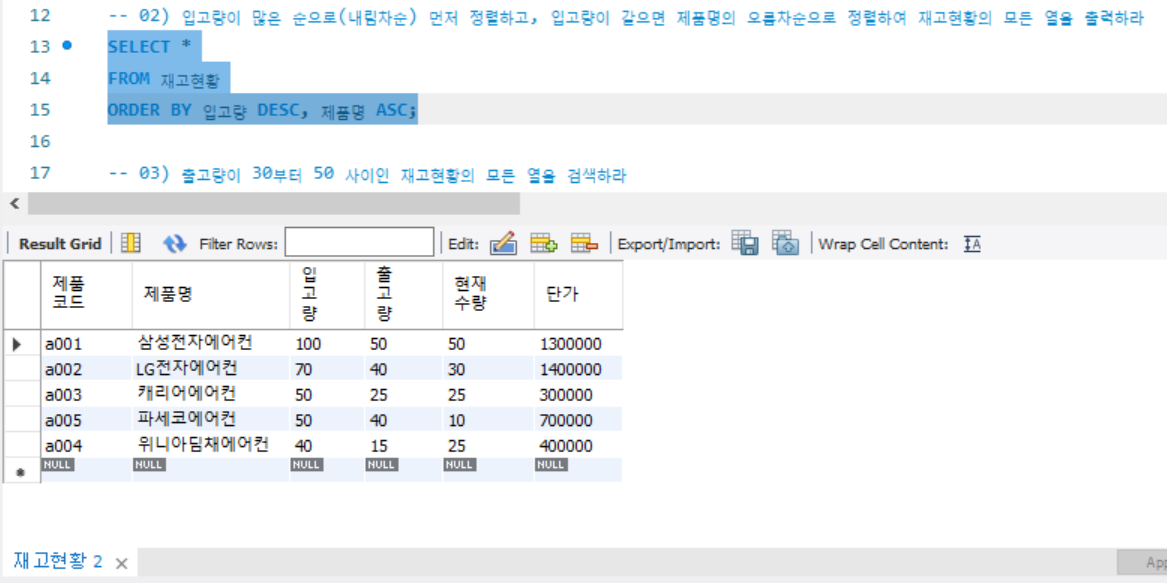
-- 02) 입고량이 많은 순으로(내림차순) 먼저 정렬하고, 입고량이 같으면 제품명의 오름차순으로 정렬하여 재고현황의 모든 열을 출력하라

SELECT \*

FROM 재고현황

ORDER BY 입고량 DESC, 제품명 ASC;

**[실행결과]**



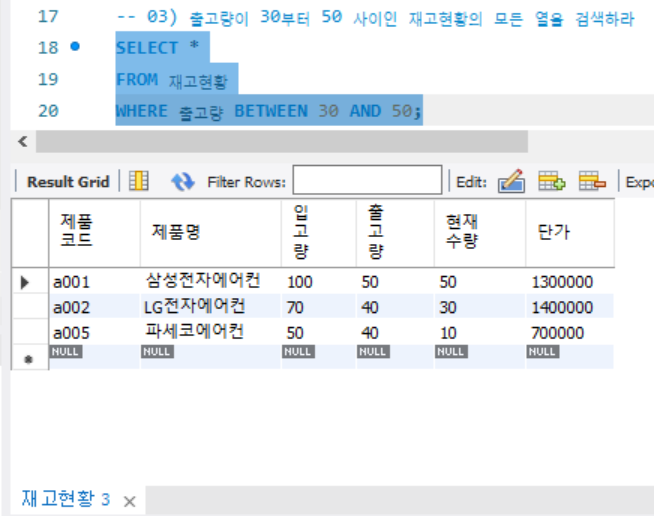
-- 03) 출고량이 30부터 50 사이인 재고현황의 모든 열을 검색하라

SELECT \*

FROM 재고현황

WHERE 출고량 BETWEEN 30 AND 50;

**[실행결과]**



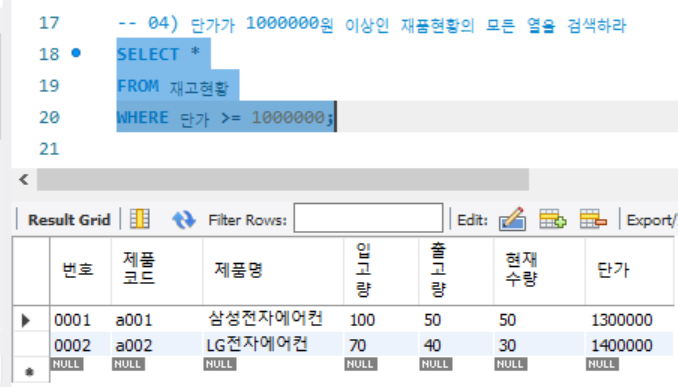
-- 04) 단가가 1000000원 이상인 재고현황의 모든 열을 검색하라

SELECT \*

FROM 재고현황

WHERE 단가 >= 1000000;

**[실행결과]**



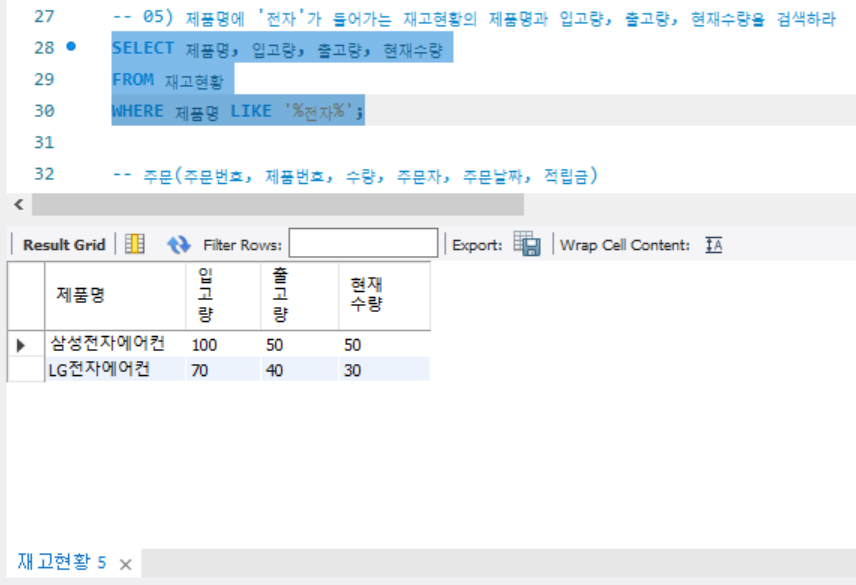
-- 05) 제품명에 '전자'가 들어가는 재고현황의 제품명과 입고량, 출고량, 현재수량을 검색하라

SELECT 제품명, 입고량, 출고량, 현재수량

FROM 재고현황

WHERE 제품명 LIKE '%전자%';

**[실행결과]**



**[주문 SELECT 문제 및 답안]**

-- 주문(주문번호, 제품번호, 수량, 주문자, 주문날짜, 적립금)

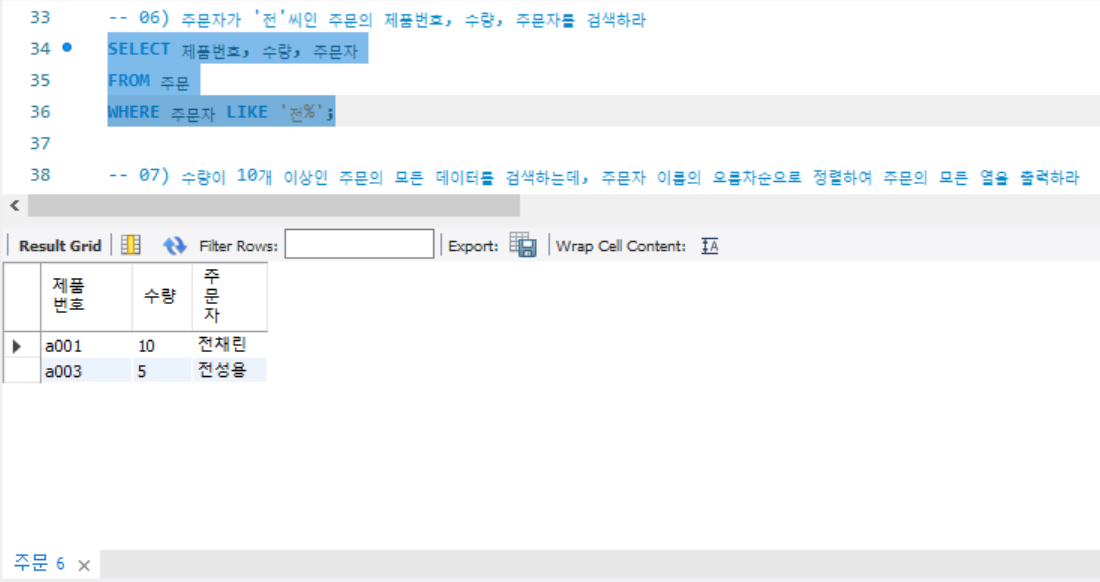
-- 06) 주문자가 '전'씨인 주문의 제품번호, 수량, 주문자를 검색하라

SELECT 제품번호, 수량, 주문자

FROM 주문

WHERE 주문자 LIKE '전%';

**[실행결과]**



-- 07) 수량이 10개 이상인 주문의 모든 데이터를 검색하는데, 주문자 이름의 오름차순으로 정렬하여 주문의 모든 열을 출력하라

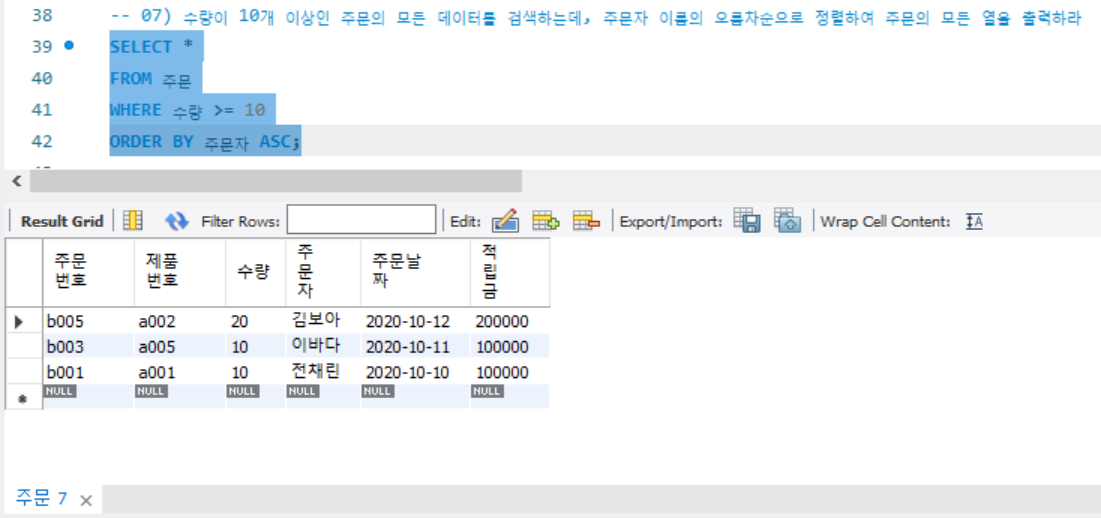
SELECT \*

FROM 주문

WHERE 수량 >= 10

ORDER BY 주문자 ASC;

**[실행결과]**



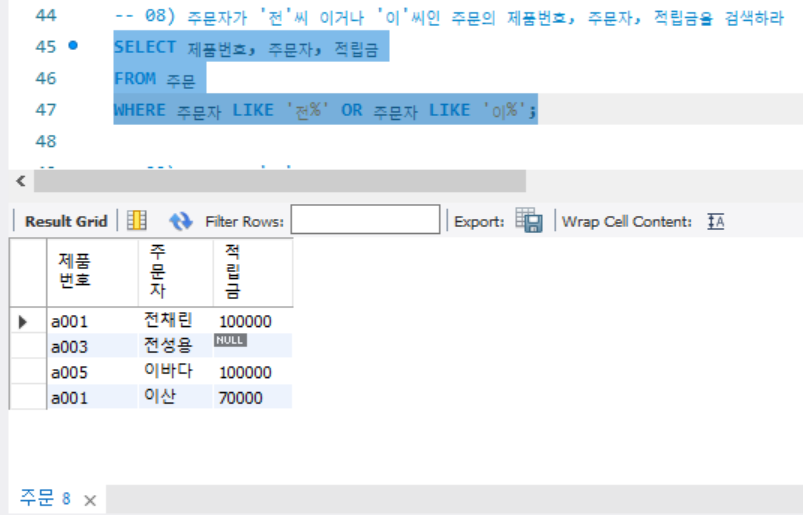
-- 08) 주문자가 '전'씨 이거나 '이'씨인 주문의 제품번호, 주문자, 적립금을 검색하라

SELECT 제품번호, 주문자, 적립금

FROM 주문

WHERE 주문자 LIKE '전%' OR 주문자 LIKE '이%';

**[실행결과]**



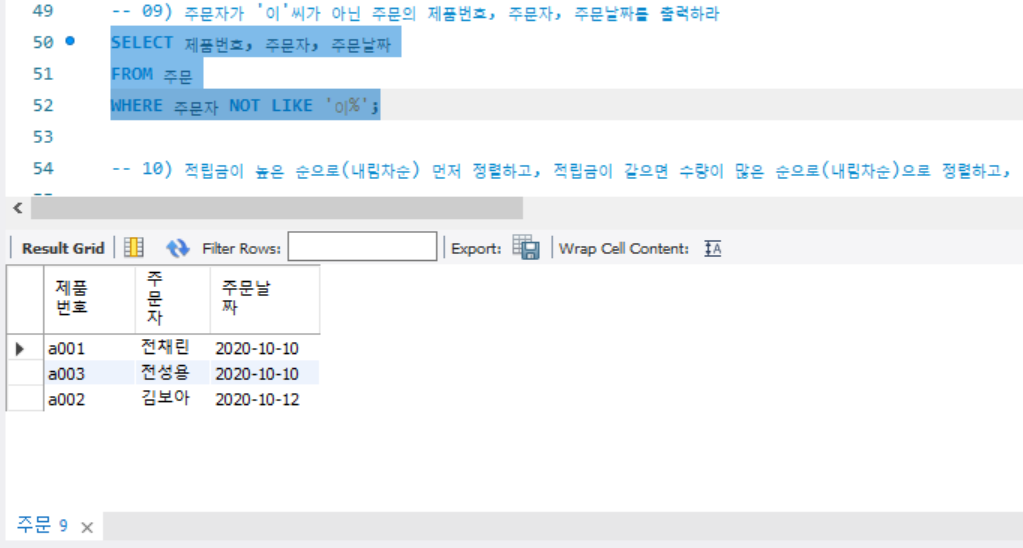
-- 09) 주문자가 '이'씨가 아닌 주문의 제품번호, 주문자, 주문날짜를 출력하라

SELECT 제품번호, 주문자, 주문날짜

FROM 주문

WHERE 주문자 NOT LIKE '이%';

**[실행결과]**



-- 10) 적립금이 높은 순으로(내림차순) 먼저 정렬하고, 적립금이 같으면 수량이 많은 순으로(내림차순)으로 정렬하고,

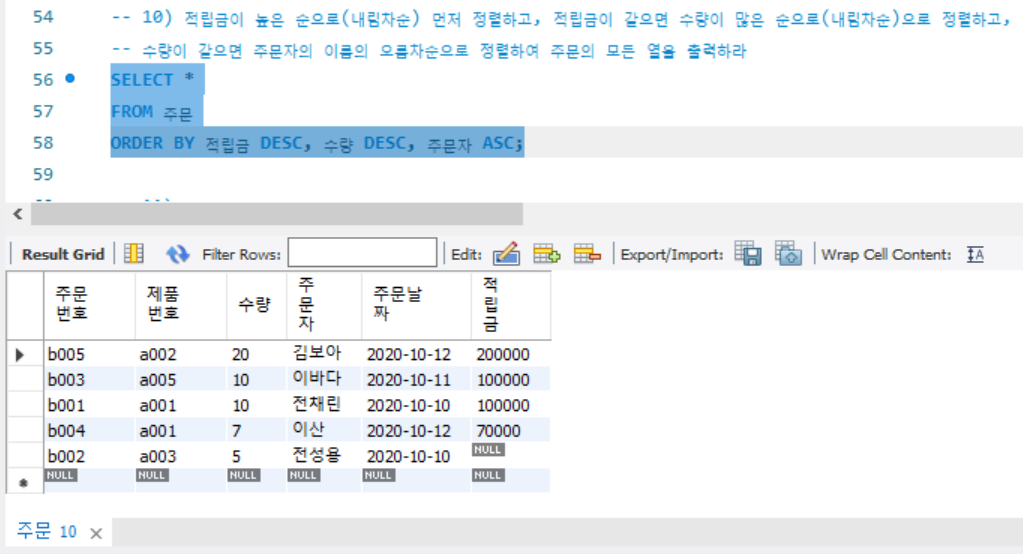
-- 수량이 같으면 주문자의 이름의 오름차순으로 정렬하여 주문의 모든 열을 출력하라

SELECT \*

FROM 주문

ORDER BY 적립금 DESC, 수량 DESC, 주문자 ASC;

**[실행결과]**



-- 11) 주문 테이블에서 적립금이 없는 주문자의 모든 열을 출력하라

SELECT \*

FROM 주문

WHERE 적립금 IS NULL;

**[실행결과]**

