**문제**



SELECT 고객회사명, CONCAT('\*\*', SUBSTR(고객회사명, 3)) AS 고객회사2,

전화번호, REPLACE(SUBSTR(전화번호, 2), ')', '-') AS 전화번호2 FROM 고객 ;



SELECT \*

, 단가\*주문수량 AS 주문금액

, truncate(주문수량\*단가\*할인율,-1) AS 할인금액

, 단가\*주문수량 - truncate(주문수량\*단가\*할인율,-1) AS 실주문금액

FROM 주문세부;

**사원 테이블에서 전체 사원의 이름, 생일, 만나이, 입사일, 입사일수, 입사한 지 500일 후의 날짜를 보이시오.**

SELECT 이름, 생일,

TIMESTAMPDIFF(YEAR, 생일, NOW()) AS 만나이, 입사일,

DATEDIFF(NOW(), 입사일) AS 입사일수,

ADDDATE(입사일, 500) AS 500일후 FROM 사원;



SELECT 담당자명, 고객회사명, 도시,

IF(도시 LIKE '%특별시' OR 도시 LIKE '%광역시', '대도시', '도시') AS 도시구분, 마일리지,

CASE WHEN 마일리지 >= 100000 THEN 'VVIP고객'

WHEN 마일리지 >=10000 THEN 'VIP고객'

ELSE '일반고객' END AS 마일리지구분 FROM 고객;



SELECT 주문번호, 고객번호, 주문일,

YEAR(주문일) AS 주문년도,

quarter(주문일) AS 주문분기,

MONTH(주문일) AS 주문월,

DAY(주문일) AS 주문일,

WEEKDAY(주문일) AS 주문요일,

CASE WEEKDAY(주문일) WHEN 0 THEN '월요일' WHEN 1 THEN '화요일' WHEN 2 THEN '수요일'

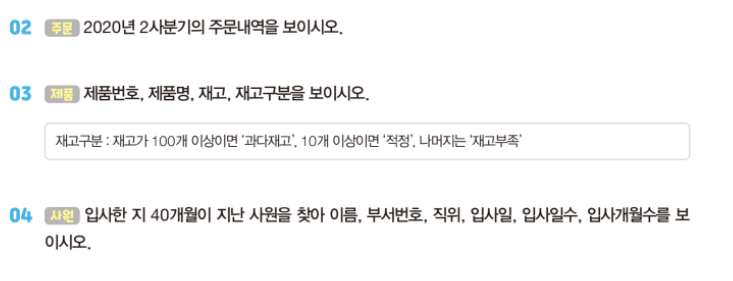
WHEN 3 THEN '목요일' WHEN 4 THEN '금요일' WHEN 5 THEN '토요일' WHEN 6 THEN '일요일' END AS 한글요일

FROM 한빛무역.주문;



SELECT \*,

DATEDIFF(발송일, 요청일) AS 지연일수 FROM 한빛무역.주문 WHERE DATEDIFF(발송일, 요청일)>=7;



1. SELECT \* FROM 한빛무역.주문 where quarter(주문일)=2 AND YEAR(주문일)=2020;
2. SELECT 제품번호, 제품명, 재고,

CASE WHEN 재고>=100 THEN '과다재고'

WHEN 재고>=10 THEN '적정' ELSE '재고부족' END AS 재고구분 FROM 한빛무역.제품;

1. SELECT 이름, 부서번호, 직위, 입사일,

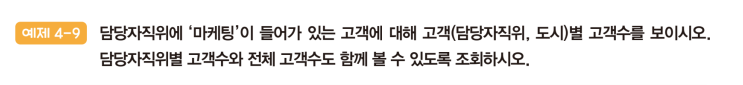
DATEDIFF(NOW(), 입사일) AS 입사일수,

TIMESTAMPDIFF(MONTH, 입사일, NOW()) AS 입사개월수

FROM 한빛무역.사원 WHERE TIMESTAMPDIFF(MONTH, 입사일, NOW())>=40;



SELECT count(\*) , count(고객번호) , count(도시) , count(지역) FROM 고객;



SELECT 담당자직위, 도시, count(\*) as 고객수

FROM 고객 where 담당자직위 like '%마케팅%' group by 담당자직위, 도시 with rollup;

문제 5-2. 이소미 사원의 사원번호, 직위, 부서번호, 부서명을 보이시오.

* **INNER JOIN**

SELECT 사원.이름, 사원.사원번호, 사원.직위, 사원.부서번호, 부서.부서명 FROM 사원, 부서 WHERE 이름 = '이소미' AND 사원.부서번호 = 부서.부서번호;

* **SELF JOIN**

SELECT 사원.이름, 사원.사원번호, 사원.직위, 사원.부서번호, 부서.부서명 FROM 사원 INNER JOIN 부서 ON 사원.부서번호 = 부서.부서번호 WHERE 이름 = '이소미';

**문제. 고객과 주문 조인. 고객회사들이 주문한 주문건수가 많은 순서대로 보이시오. 고객번호, 담당자명, 고객회사명을 보이시오.**

* **INNER JOIN**

SELECT DISTINCT 고객.고객번호, 고객.담당자명, 고객.고객회사명, COUNT(\*) AS 주문건수 FROM 고객 INNER JOIN 주문 ON 주문.고객번호 = 고객.고객번호 GROUP BY 고객.고객번호, 고객.담당자명, 고객.고객회사명 ORDER BY COUNT(\*) DESC;

* **SELF JOIN**

SELECT DISTINCT 고객.고객번호, 고객.담당자명, 고객.고객회사명, COUNT(\*) AS 주문건수 FROM 고객, 주문 WHERE 주문.고객번호 = 고객.고객번호 GROUP BY 고객.고객번호, 고객.담당자명, 고객.고객회사명 ORDER BY COUNT(\*) DESC;

**문제. 고객테이블과 마일리지 등급 조인. 고객테이블에서 담당자가 이은광인 경우의 고객번호, 고객회사명, 담당자명, 마일리지와 마일리지 등급을 보이시오.**

SELECT 고객.고객번호, 고객.고객회사명, 고객.담당자명, 고객.마일리지, 마일리지등급.등급명 FROM 고객 INNER JOIN 마일리지등급 ON 마일리지 BETWEEN 마일리지등급.하한마일리지 AND 마일리지등급.상한마일리지 WHERE 담당자명='이은광';

**Inner Join 연습 문제 5개**

1⃣ **학생 이름과 그 학생이 신청한 과목명을 모두 출력하세요.**

SELECT 이름, 수강신청.과목명 FROM 학생 INNER JOIN 수강신청 ON 학생.학번 = 수강신청.학번;

2⃣ **전공이 \*\*'컴퓨터공학'\*\*인 학생들의 이름과 수강한 과목명을 출력하세요.**

SELECT 이름, 수강신청.과목명 FROM 학생 INNER JOIN 수강신청 ON 학생.학번 = 수강신청.학번 WHERE 학생.전공='컴퓨터공학';

**3⃣ 성적이 \*\*'A+'\*\*인 학생 이름과 과목명을 출력하세요.**

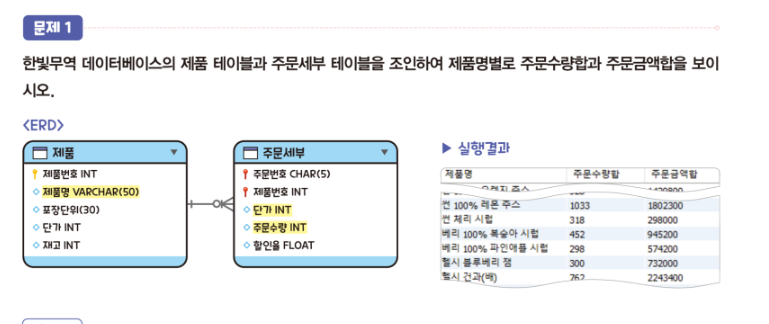
SELECT 이름, 수강신청.과목명 FROM 학생 INNER JOIN 수강신청 ON 학생.학번 = 수강신청.학번 WHERE 수강신청.성적='A+';

**4⃣ 학생 이름, 전공, 신청한 과목명, 성적을 전부 출력하고 성적순으로 내림차순 정렬하세요**.

SELECT 이름, 전공, 수강신청.과목명, 수강신청.성적 FROM 학생 INNER JOIN 수강신청 ON 학생.학번 = 수강신청.학번 ORDER BY 수강신청.성적 DESC;

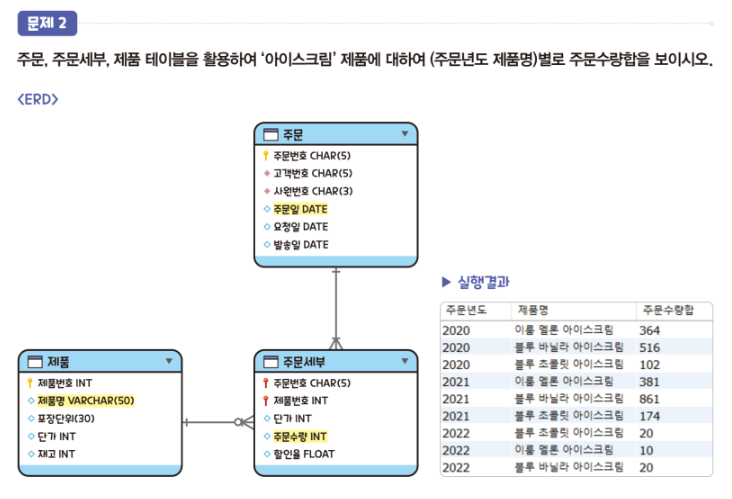
**5⃣ 학생 이름별 신청한 과목 수를 출력하세요. (이름, 과목 수)**

SELECT 이름, COUNT(\*) AS 과목수 FROM 학생 INNER JOIN 수강신청 ON 학생.학번 = 수강신청.학번 GROUP BY 이름;



SELECT 제품명, sum(주문수량) as 주문수량합, sum(주문세부.단가\*주문수량) as 주문금액합 from 제품 inner join 주문세부 on 주문세부.제품번호 = 제품.제품번호 group by 제품명;

SELECT 제품명, sum(주문수량) as 주문수량합, sum(주문세부.단가\*주문수량) as 주문금액합 from 제품, 주문세부 where 주문세부.제품번호 = 제품.제품번호 group by 제품명;



**INNER JOIN(2번)**

SELECT YEAR(주문일) AS 주문년도, 제품명, SUM(주문수량) AS 주문수량합 FROM 제품 INNER JOIN 주문세부 ON 제품.제품번호 = 주문세부.제품번호 INNER JOIN 주문 ON 주문.주문번호 = 주문세부.주문번호 WHERE 제품명 LIKE '%아이스크림' GROUP BY YEAR(주문일), 제품명 ORDER BY 1,2;

**SELF JOIN**

SELECT YEAR(주문일) AS 주문년도, 제품명, SUM(주문수량) AS 주문수량합 FROM 제품, 주문세부, 주문 WHERE 제품.제품번호 = 주문세부.제품번호 AND 주문.주문번호 = 주문세부.주문번호 AND 제품명 LIKE '%아이스크림' GROUP BY YEAR(주문일), 제품명 ORDER BY 1,2;

**문제 1: 고객의 이름과 해당 고객이 주문한 주문 ID를 함께 조회하세요.**

SELECT 이름, 주문ID FROM sample.고객 INNER JOIN 주문 ON 고객.고객ID = 주문.고객ID;

**문제 2: 주문한 고객의 이름과 주문일자, 총금액을 조회하세요.**

SELECT 이름, 주문일자, 총금액 FROM sample.주문 INNER JOIN 고객 ON 주문.고객ID = 고객.고객ID;

**문제 3: VIP 등급 고객의 이름과 그 고객의 모든 주문 ID를 조회하세요.**

SELECT 이름, 주문ID FROM sample.고객 INNER JOIN 주문 ON 고객.고객ID = 주문.고객ID WHERE 등급='VIP';

**문제 4: 총금액이 10만원 이상인 주문을 한 고객의 이름과 주문 ID, 총금액을 조회하세요.**

SELECT 이름, 주문ID, 총금액 FROM sample.고객 INNER JOIN 주문 ON 고객.고객ID = 주문.고객ID WHERE 총금액>=100000;

**문제 5: 각 고객별로 주문 건수를 조회하고, 고객 이름과 주문 건수를 함께 출력하세요. (힌트: GROUP BY 절과 COUNT() 함수를 사용하세요.)**

SELECT 이름, COUNT(\*) AS 주문건수 FROM sample.고객 INNER JOIN 주문 ON 고객.고객ID = 주문.고객ID group by 이름;

**문제**

1. 제품 테이블에 새로운 제품 ('태블릿', 450000, 1, 25)을 삽입하는 명령을 쓰시오.

INSERT INTO 제품 VALUES (NULL, ‘태블릿’, 450000, 1, 25);

1. 제품 테이블에서 제품명이 '마우스'인 제품의 가격을 30000으로 수정하는 명령을 쓰시오.

UPDATE 제품 SET 가격=30000 WHERE 제품명='마우스';

1. 제품 테이블에서 재고 수량이 10개 미만인 제품의 제품명과 재고 수량을 조회하는 명령을 쓰시오.

SELECT 제품명, 재고수량 FROM 제품.제품 WHERE 재고수량 < 10;

1. 제품 테이블에서 카테고리 ID가 NULL인 제품을 삭제하는 명령을 쓰시오.

DELETE FROM 제품 WHERE 카테고리ID IS NULL;

1. 카테고리 테이블에 새로운 카테고리 ('사무용품')을 삽입하는 명령을 쓰시오.

INSERT INTO 카테고리 VALUES (NULL, '사무용품');

1. 카테고리 테이블에서 카테고리명이 '도서'인 레코드를 삭제하는 명령을 쓰시오

DELETE FROM 카테고리 WHERE 카테고리명='도서';

JOIN 문제:

1. 제품 테이블과 카테고리 테이블을 조인하여 제품명과 해당 제품의 카테고리명을 함께 조회하세요.

SELECT 제품명, 제품.카테고리ID FROM 제품.제품 INNER JOIN 카테고리 ON 제품.카테고리ID = 카테고리.카테고리ID;

1. 가격이 10만원 이상인 제품의 제품명과 카테고리명을 조회하세요.

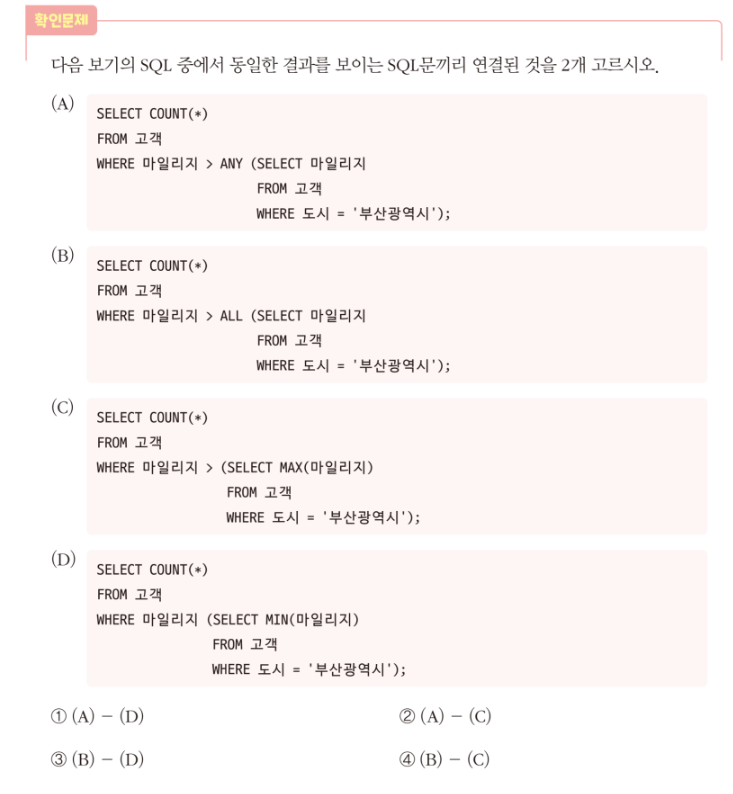
SELECT 제품명, 제품.카테고리ID FROM 제품.제품 INNER JOIN 카테고리 ON 제품.카테고리ID = 카테고리.카테고리ID WHERE 가격>=100000;

1. 카테고리명이 '컴퓨터 주변기기'인 모든 제품의 제품명과 가격을 조회하세요.

SELECT 제품명, 가격 FROM 제품.제품 INNER JOIN 카테고리 ON 제품.카테고리ID = 카테고리.카테고리ID WHERE 카테고리.카테고리명='컴퓨터 주변기기';

1. 각 카테고리별로 포함된 제품의 수를 조회하고, 카테고리명과 제품 수를 함께 출력하세요.

SELECT 카테고리명, COUNT(\*) AS 제품수 FROM 카테고리 INNER JOIN 제품 ON 제품.카테고리ID = 카테고리.카테고리ID GROUP BY 카테고리명;



문제 1: 고객의 이름과 해당 고객이 주문한 주문 ID를 함께 조회하세요.

SELECT name, orderid from customers inner join orders on customers.customerid = orders.customerid;

문제 2: 주문한 고객의 이름과 주문일자, 총 금액을 조회하세요.

SELECT name, orderdate, totalamount from customers as c inner join orders as o on c.customerid=o.CustomerID;

문제 3: 독일(Germany)에 거주하는 고객의 이름과 그 고객의 모든 주문 ID를 조회하세요.

SELECT name, orderid from customers as c inner join orders as o on c.customerid=o.CustomerID where country='Germany';

문제 4: 총 주문 금액이 1000 이상인 주문을 한 고객의 이름과 주문 ID, 총 금액을 조회하세요.

SELECT name, orderid, TotalAmount from customers as c inner join orders as o on c.customerid=o.CustomerID where TotalAmount>=1000;

문제 5: 각 고객별로 주문 건수를 조회하고, 고객 이름과 주문 건수를 함께 출력하세요. (힌트: GROUP BY 절과 COUNT() 함수를 사용하세요.)

SELECT name, count(\*) as 주문건수 from customers as c inner join orders as o on c.customerid=o.CustomerID group by name;