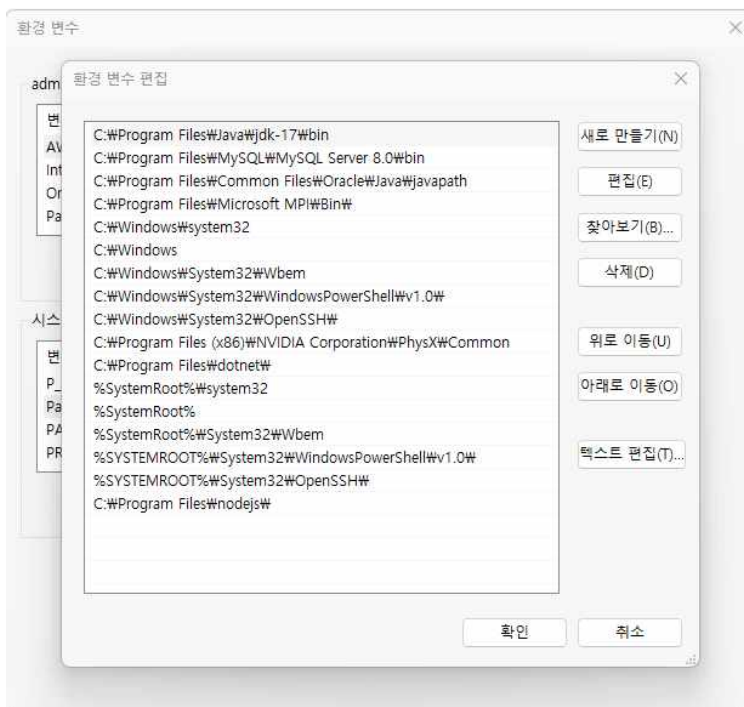


판매 데이터베이스 구축과 요구사항

판매관련 데이터베이스를 구축하기 위해 MySQL DB를 설치하였다. 아래의 내용을 수행 한 후 그 결과를 적으시오.

[문항1] MySQL DB를 설치한 후 환경변수를 설정하는 절차와 경로를 적으시오.



[문항2] root 사용자로 로그인 후 panDB를 생성하시오.

```
mysql> create database pandb;
```

[문항3] user1 사용자를 생성하고 panDB를 사용할 수 있는 권한을 설정하시오.

```
mysql> creat user user1@'%' identified by '1111';
```

```
mysql> grant all privileges on pandb.* to user1@'%' with grant option;
```

[문항4] 아래의 네 개 테이블을 분석한 후

- ① www.draw.io 를 이용하여 ERD를 그린 후 첨부하시오.
- ② 각각 테이블을 생성한 후

show create table 테이블명;

을 이용해 생성된 테이블 생성 구문을 적으시오.(모든 필드명은 소문자, snake notation 규칙을 따릅니다)

③ 외래키 constraint를 ERD를 참고하여 각 테이블에 설정하시오.

단, 1:N의 구조를 갖는 테이블에서 1 side 테이블의 레코드가 삭제되면 N side 테이블의 모든 데이터가 삭제되도록 설정하시오.

④ 샘플 데이터를 입력하시오.

Table Name : DEPT

field_name	Type	Null	PK	ETC
DEP_NO	INT		PK	
DEP_NAME	VARCHAR(10)			
LOC	VARCHAR(10)	NOT NULL,		

DATA :

10,'총무부','서울'

20,'영업부','부산'

30,'전산부','대전'

40,'자재부','광주'

Table Name : EMPLOYEE

field_name	Type	Null	PK	Default	etc
EMP_NO	INT		PK		
EMP_NAME	VARCHAR(10)	NOT NULL			
DEP_NO	INT				
position	VARCHAR(10)	NOT NULL			
SALARY	INT			0	
HIRE_DATE	DATE			now()	
JUMIN_NUM	CHAR(13)	NOT NULL			UNIQUE
PHONE_NUM	VARCHAR(15)	NOT NULL			
MGR_EMP_NO	INT				

DATA :

'홍길동', 10, '사장', 5000, '1980-01-01','7211271109410', '01099699515', NULL
 '한국남', 20, '부장', 3000, '1988-11-01','6002061841224', '01024948424', 1
 '이순신', 20, '과장', 3500,'1989-03-01','6209172010520', '01026352672', 2
 '이미라', 30, '대리', 2503,'1983-04-01','8409282070226', '01094215694', 17
 '이순라', 20, '사원', 1200,'1990-05-01','8401041483626', '01094215694', 3
 '공부만', 30, '과장', 4003, '1995-05-01', '8402121563616', '01191338328', 17
 '놀기만', 20, '과장', 2300,'1996-06-01','8011221713914', '01029463523', 2
 '채송화', 30, '대리', 1703,'1992-06-01','8105271014112', '01047111052', 17
 '무궁화', 10, '사원', 1100,'1984-08-01','8303291319116', '01025672300', 12
 '공부해', 30, '사원', 1303,'1988-11-01','8410031281312', '01027073174', 4
 '류별나', 20, '과장', 1600,'1989-12-01','8409181463545', '01071628290', 2
 '류명환', 10, '대리', 1800,'1990-10-01','8207211661117', '01042072622', 20
 '무궁화', 10, '부장', 3000,'1996-11-01','8603231183011', '01098110955', 1
 '채시라', 20, '사원', 3400,'1993-10-01','8001172065410', '01044452437', 3
 '최진실', 10, '사원', 2000,'1991-04-01','8303101932611', '01027491145', 12
 '김유신', 30, '사원', 4000,'1981-04-01','7912031009014', '01098218448', 4
 '이성계', 30, '부장', 2803,'1984-05-01', '8102261713921', '0165358075', 1
 '강감찬', 30, '사원', 1003,'1986-07-01','8203121977315', '01033583130', 4
 '임꺽정', 20, '사원', 2200,'1988-04-01','8701301040111', '01086253078', 7
 '깨똥이', 10, '과장', 4500,'1990-05-01','8811232452719', '01090084876', 13

Table Name : CUSTOMER

field_name	Type	Null	PK	Default	etc
CUS_NO	INT/자동증가		PK		
CUS_NAME	VARCHAR(10)	NOT NULL			
TEL_NUM	VARCHAR(15)				
JUMIN_NUM	CHAR(13)	NOT NULL			
EMP_NO	INT			0	

DATA :

'류민이', '123-1234', '7001131537915', 3
 '이강민', '343-1454', '6902161627914', 2

'이영희', '144-1655', '7503202636215', NULL
 '김철이', '673-1674', '7704301234567', 4
 '박류완', '123-1674', '7205211123675', 3
 '서캔디', '673-1764', '6507252534566', NULL
 '신똥이', '176-7677', '0006083648614', 7
 '도쇠돌', '123-6774', '0008041346574', 9
 '권홍이', '767-1234', '7312251234689', 13
 '김안나', '767-1677', '7510152432168', 4

Table Name : SALARY_GRADE

field_name	Type	Null	PK	Default	etc
SAL_GRADE_NO	INT		PK		
MIN_SAL	INT	NOT NULL			
MAX_SAL	INT	NOT NULL			

DATA :

1, 8001, 10000
 2, 6001, 8000
 3, 4001, 6000
 4, 2001, 4000
 5, 1000, 2000

```

create table dept(
dep_no int primary key,
dep_name varchar(10),
loc varchar(10) not null);
  
```

```

create table employee(
emp_no int primary key auto_increment,
emp_name varchar(10) not null,
dep_no int,
position varchar(10) not null,
salary int default 0,
  
```

```
hire_date date default (current_date),
jumin_num char(13) not null unique,
phone_num varchar(15) not null,
mgr_emp_no int);
```

```
create table customer(
cus_no int primary key auto_increment,
cus_name varchar(10) not null,
tel_num varchar(15),
jumin_num char(13) not null,
emp_no int);
```

```
alter table employee
add constraint fk_dep_no
foreign key(dep_no) references dept(dep_no)
on delete cascade;
```

```
alter table customer
add constraint fk_emp_no
foreign key(emp_no) references employee(emp_no)
on delete cascade;
```

```
insert into dept values(10,'총무부','서울');
insert into dept values(20,'영업부','부산');
insert into dept values(30,'전산부','대전');
insert into dept values(40,'자재부','광주');
```

```
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('홍길동', 10, '사장', 5000, '1980-01-01','7211271109410', '01099699515', NULL);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
```

```
values('한국남', 20, '부장', 3000, '1988-11-01','6002061841224', '01024948424', 1);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('이순신', 20, '과장', 3500,'1989-03-01','6209172010520', '01026352672', 2);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('이미라', 30, '대리', 2503,'1983-04-01','8409282070226', '01094215694', 17);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('이순라', 20, '사원', 1200,'1990-05-01','8401041483626', '01094215694', 3);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('공부만', 30, '과장', 4003, '1995-05-01', '8402121563616', '01191338328', 17);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('놀이만', 20, '과장', 2300,'1996-06-01','8011221713914', '01029463523', 2);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('채송화', 30, '대리', 1703,'1992-06-01','8105271014112', '01047111052', 17);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('무궁화', 10, '사원', 1100,'1984-08-01','8303291319116', '01025672300', 12);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('공부해', 30, '사원', 1303,'1988-11-01','8410031281312', '01027073174', 4);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('류별나', 20, '과장', 1600,'1989-12-01','8409181463545', '01071628290', 2);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('류명환', 10, '대리', 1800,'1990-10-01','8207211661117', '01042072622', 20);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('무궁화', 10, '부장', 3000,'1996-11-01','8603231183011', '01098110955', 1);
```

```
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('채시라', 20, '사원', 3400,'1993-10-01','8001172065410', '01044452437', 3);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('최진실', 10, '사원', 2000,'1991-04-01','8303101932611', '01027491145', 12);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('김유신', 30, '사원', 4000,'1981-04-01','7912031009014', '01098218448', 4);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('이성계', 30, '부장', 2803,'1984-05-01', '8102261713921', '0165358075', 1);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('강감찬', 30, '사원', 1003,'1986-07-01','8203121977315', '01033583130', 4);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('임꺽정', 20, '사원', 2200,'1988-04-01','8701301040111', '01086253078', 7);
insert into employee(emp_name, dep_no, position, salary, hire_date,
jumin_num,phone_num, mgr_emp_no)
values('깨똥이', 10, '과장', 4500,'1990-05-01','8811232452719', '01090084876', 13);
```

```
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'류민이', '123-1234', '7001131537915', 3);
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'이강민', '343-1454', '6902161627914', 2 );
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'이영희', '144-1655', '7503202636215', NULL );
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'김철이', '673-1674', '7704301234567', 4 );
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'박류완', '123-1674', '7205211123675', 3 );
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'서캔디', '673-1764', '6507252534566', NULL );
```

```
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'신뜰이', '176-7677', '0006083648614', 7 );
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'도쇠돌', '123-6774', '0008041346574', 9 );
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'권홍이', '767-1234', '7312251234689', 13 );
insert into customer(cus_name, tel_num, jumin_num, emp_no) values(
'김안나', '767-1677', '7510152432168', 4 );
```

```
create table salary_grade(
sal_grade_no int primary key,
min_sal int not null,
max_sal int not null);
```

```
insert into salary_grade values(1, 8001, 10000);
insert into salary_grade values(2, 6001, 8000);
insert into salary_grade values(3, 4001, 6000);
insert into salary_grade values(4, 2001, 4000);
insert into salary_grade values(5, 1000, 2000);
```


아래의 결과를 얻을 수 있도록 쿼리를 수행하고 그 쿼리문을 적고 결과를 화면캡처 후 첨부하시오.

1. 부서 테이블의 모든 정보를 보이시오.

SELECT * FROM DEPT;

2. 부서테이블의 부서ID와 위치를 출력하시오.

SELECT DEP_NO, LOC FROM DEPT;

3. 직원테이블에서 이름과 급여, 급여에 300을 더해 인센티브로 필드명을 부여해 출력하시오.

SELECT EMP_NAME, SALARY, SALARY+300 AS 인센티브 FROM EMPLOYEE;

4. 직원테이블의 이름과 매니저 번호를 출력하고 매니저 번호가 NULL이면 '없음' 으로 출력하시오.

IFNULL(필드명, NULL일경우 처리값) 함수 이용

SELECT EMP_NAME, IFNULL(MGR_EMP_NO,'없음') AS MANAGER FROM EMPLOYEE;

5. 부서 테이블의 부서명과 위치를 하나의 열에 출력하시오.(Concat()) 함수 이용
단, 부서명과 위치는 공백으로 구분하시오.(총무부 서울)

SELECT CONCAT(DEP_NO, ' ',LOC) AS 정보 FROM DEPT;

6. 부서ID가 30 인 직원의 정보를 모두 출력하시오.

SELECT * FROM EMPLOYEE WHERE DEP_NO=30;

7. 직원들의 정보를 출력하되 부서별로 같은 부서의 사람들은 가장 최근에 입사한 사람 순서로 출력하시오.

SELECT * FROM EMPLOYEE ORDER BY DEP_NO, HIRE_DATE DESC;

8. 직원테이블에서 김씨성을 갖는 사람들의 정보를 출력하시오

SELECT * FROM EMPLOYEE WHERE EMP_NAME LIKE '김%';

9. 직원테이블에서 1990년에 채용된 사람들의 정보를 출력하시오.

```
SELECT * FROM EMPLOYEE WHERE YEAR(HIRE_DATE)=1990;
```

10. 직원테이블에서 전화번호 뒷자리가 '8075'인 사람의 정보를 출력하시오.

```
SELECT * FROM EMPLOYEE WHERE RIGHT(PHONE_NUM,4)='8075';
```

11. 직원테이블에서 급여가 1200 보다 작은 사람의 정보를 출력하시오.

```
SELECT * FROM EMPLOYEE WHERE SALARY<1200;
```

12. 직원테이블에서 대리 또는 과장의 정보를 출력하시오.

```
SELECT * FROM EMPLOYEE WHERE position IN('대리','과장');
```

13. 직원테이블에서 매니저 값이 널(NULL) 값을 갖는 자료를 출력하시오.

```
SELECT * FROM EMPLOYEE WHERE MGR_EMP_NO IS NULL;
```

14. 직원테이블에서 직원ID, 이름, 직위, 급여를 출력하시오

(조건 : 급여가 2000 이상이고 부서코드가 20)

```
SELECT EMP_NO, EMP_NAME, position, SALARY FROM EMPLOYEE  
WHERE SALARY>=2000 AND DEP_NO=20;
```

15. 직원테이블에서 부서코드가 10, 20 인 사람들을 제외하고 출력하시오.

```
SELECT * FROM EMPLOYEE WHERE DEP_NO NOT IN(10,20);
```

16. 직원 테이블에서 자료를 출력하되 다음의 조건으로 출력하시오.

- 직위 필드에 '장'을 포함하는 자료를
- 이름과 직위를 묶어서 NAME이라는 필드명으로
- 이름의 길이 출력(Length 함수)
- 주민번호 중 숫자 '9'가 첫번째로 나타나는 위치를 Contains '9'? 라는 필드명으로 출력하시오. (Instr 함수)

```
SELECT CONCAT(EMP_NAME,' ',position) AS NAME, LENGTH(EMP_NAME) AS '이름길  
이',
```

```
INSTR(JUMIN_NUM, '9') AS 'Contains 9?' FROM EMPLOYEE;
```

17. 직원테이블에서 급여를 1000 으로 나눈 나머지 값 출력

```
SELECT MOD(SALARY, 1000) FROM EMPLOYEE;
```

18. 직원테이블에서 직원의 근무 년수가 30년 이상인 사람을 구하여 '30년 이상 근속자'라는 필드로 출력하시오.

```
select emp_name as '30년 근속자' , timestampdiff(year,hire_date,now()) as '근속년수'  
from employee where timestampdiff(year,hire_date,now()) >= 30;
```

19. 직원 테이블에서 직원들의 급여평균, 급여총액, 전체직원 수를 구하시오.

```
SELECT AVG(SALARY), SUM(SALARY), COUNT(EMP_NO) FROM EMPLOYEE;
```

20. 직원 테이블에서 직원들의 급여평균, 급여총액, 전체직원 수를 부서별로 구하시오.

```
SELECT DEP_NO, AVG(SALARY), SUM(SALARY), COUNT(EMP_NO)  
FROM EMPLOYEE GROUP BY DEP_NO;
```

21. 직원 테이블에서 가장 처음과 가장 마지막 입사날짜를 출력하시오.

```
SELECT MIN(HIRE_DATE), MAX(HIRE_DATE) FROM EMPLOYEE;
```

22. 직원테이블에서 부서ID 별 평균 급여를 출력하시오.

```
SELECT DEP_NO, AVG(SALARY) FROM EMPLOYEE GROUP BY DEP_NO;
```

23. 직원테이블에서 부서ID별, 직위별 급여의 합계를 구하시오.

```
SELECT d.DEP_Name, e.position, SUM(SALARY) AS '급여합계'  
FROM EMPLOYEE e  
inner join dept d  
on e.dep_no=d.dep_no  
GROUP BY DEP_name, position;
```

24. 부서ID 별 급여의 합계를 구하고 합계가 17300 이상인 자료만 출력하시오.

```
SELECT DEP_NO, SUM(SALARY) AS '급여합계' FROM EMPLOYEE  
GROUP BY DEP_NO HAVING SUM(SALARY)>=17300;
```

25. 부서ID 별 급여의 합계를 구하고 합계가 17300 이상인 자료만 출력하되 내림차순

으로 하시오.

```
SELECT DEP_NO, SUM(SALARY) AS '급여합계' FROM EMPLOYEE  
GROUP BY DEP_NO HAVING SUM(SALARY)>=17300  
ORDER BY SUM(SALARY) DESC;
```

26. 부서ID 별 평균급여의 가장 큰 값을 출력하시오.(단, subquery를 사용하시오.)

```
select MAX(평균급여) FROM  
(SELECT AVG(salary) as 평균급여 FROM employee GROUP BY dep_no) t;
```

27. 직원 중 '공부만' 보다 급여를 많이 받는 사람의 이름과 급여를 출력하시오.
(단, subquery를 사용하시오.)

```
SELECT EMP_NAME, SALARY FROM EMPLOYEE  
WHERE SALARY>(SELECT SALARY FROM EMPLOYEE WHERE EMP_NAME='공부만');
```

28. 직원 테이블에서 직원ID가 13번인 직원의 직위과 같고, 직원ID 가 17번인 직원의
급여보다 큰 직원들의 정보를 이름, 직위, 급여의 순으로 출력하는 쿼리를 작성하시오.
(단, subquery를 사용하시오.)

```
SELECT EMP_NAME, position, SALARY FROM EMPLOYEE  
WHERE position=(SELECT position FROM EMPLOYEE WHERE EMP_NO=13) AND  
SALARY>(SELECT SALARY FROM EMPLOYEE WHERE EMP_NO=17);
```

29. 직원 중 급여가 가장 적은 사람의 정보를 이름, 직위, 급여를 출력하시오.
(단, subquery를 사용하시오.)

```
SELECT EMP_NAME, position, SALARY FROM EMPLOYEE  
WHERE SALARY=(SELECT MIN(SALARY) FROM EMPLOYEE);
```

30. 각 부서의 최소 급여 중 30번 부서의 최소 급여보다 많은 타 부서의 최소급여를
부서ID, 최소급여의 순서로 출력하시오.
(단, subquery를 사용하시오.)

```
SELECT DEP_NO, MIN(SALARY) FROM EMPLOYEE GROUP BY DEP_NO  
HAVING MIN(SALARY)>  
(SELECT MIN(SALARY) FROM EMPLOYEE GROUP BY DEP_NO HAVING DEP_NO=30);
```

31. 직원테이블에서 사원의 평균급여보다 크고 과장의 평균급여보다 작은 직원이름과 급여를 출력하시오.

(단, subquery를 사용하시오.)

```
SELECT EMP_NAME, SALARY FROM EMPLOYEE WHERE  
SALARY BETWEEN  
(SELECT AVG(SALARY) FROM EMPLOYEE GROUP BY position HAVING position='사원'  
)  
AND  
(SELECT AVG(SALARY) FROM EMPLOYEE GROUP BY position HAVING position='과장'  
);
```

32. 직원 중 '놀기만'과 '류별나'의 부서ID 를 40 으로 변경하시오.

```
UPDATE EMPLOYEE SET DEP_NO=40 WHERE EMP_NAME IN ('놀기만','류별나');
```

33. 직원 중 '놀기만'의 salary와 직위를 '공부해'의 salary와 직위로 바꾸시오.

(단, subquery를 사용하시오.)

```
UPDATE EMPLOYEE  
SET SALARY=(select * from (SELECT SALARY FROM EMPLOYEE WHERE  
EMP_NAME ='공부해') as a),  
position=(select * from (SELECT position FROM EMPLOYEE  
WHERE EMP_NAME ='공부해') as b)  
WHERE EMP_NAME ='놀기만';
```

34. 다음의 내용을 부서테이블과 직원테이블에 각각 삽입하시오.

- 부서 : 50, '기획부', '제주'

```
INSERT INTO DEPT (DEP_NO, DEP_NAME, LOC) VALUES (50, '기획부', '제주');
```

- 직원 : '말뚝이', 50, '과장', 4200,'1993-01-03','8911232452719', '01990084876', 13

```
INSERT INTO EMPLOYEE (EMP_NAME, DEP_NO, position, SALARY, HIRE_DATE,  
JUMIN_NUM, PHONE_NUM, MGR_EMP_NO)  
VALUES('말뚝이', 50, '과장', 4200,'1993-01-03','8911232452719', '01990084876', 13);
```

35. 지역이 제주인 부서의 직원 데이터를 삭제하시오.

(단, subquery를 사용하시오.)

```
DELETE FROM EMPLOYEE
WHERE DEP_NO=(SELECT DEP_NO FROM DEPT WHERE DEP_NAME='제주');
```

36. 직원의 이름과 부서명, 지역을 출력하시오.

```
SELECT EMP_NAME, DEP_NAME, LOC FROM EMPLOYEE AS E JOIN DEPT AS D
ON E.DEP_NO=D.DEP_NO;
```

37. 직원이 담당자로 배정 된 고객 데이터 내에서 고객이름, 담당자 이름, 담당자 직위, 부서명을 순서대로 출력하시오.

```
SELECT CUS_NAME, EMP_NAME, position, DEP_NAME
FROM DEPT AS D JOIN EMPLOYEE AS E ON D.DEP_NO=E.DEP_NO
JOIN CUSTOMER AS C ON C.EMP_NO=E.EMP_NO;
```

38. 고객이 할당되어 있는 직원의 이름이 한번만 출력하되 내림차순으로 출력하시오.
(distinct 키워드 사용)

```
SELECT DISTINCT EMP_NAME FROM CUSTOMER AS C JOIN EMPLOYEE AS E
ON C.EMP_NO=E.EMP_NO WHERE C.EMP_NO IS NOT NULL
ORDER BY EMP_NAME DESC;
```

39. 지역별로 근무중인 직원들의 수를 지역, 직원수 필드로 출력하시오.

```
SELECT LOC, COUNT(EMP_NO) AS '직원수' FROM DEPT AS D JOIN EMPLOYEE AS E
ON D.DEP_NO=E.DEP_NO GROUP BY E.DEP_NO;
```

40. 직위별로 각 지역에 근무하는 근무자 수를 지역, 근무자 수 필드로 출력하시오.

```
SELECT position, LOC, COUNT(EMP_NO) AS '직원수' FROM DEPT AS D JOIN
EMPLOYEE AS E
ON D.DEP_NO=E.DEP_NO GROUP BY position, LOC;
```

41. 몇 년도에 몇 명이 입사했는지 영업부의 입사년도에 따른 입사자 수를 년도의 내림차순으로 구하시오.(필드명 : 입사년도, 입사자 수)

```
SELECT YEAR(HIRE_DATE) AS '입사년도', COUNT(EMP_NO) AS '입사자수' FROM
EMPLOYEE
GROUP BY YEAR(HIRE_DATE) ORDER BY YEAR(HIRE_DATE) DESC;
```

42. 담당직원이 있는 고객 중 고객이름과 고객이 방문해야할 지역을 출력하시오.
(필드명: 고객명, 방문지역)

```
SELECT CUS_NAME, LOC FROM  
DEPT AS D JOIN EMPLOYEE AS E ON D.DEP_NO=E.DEP_NO  
JOIN CUSTOMER AS C ON C.EMP_NO=E.EMP_NO;
```

43. 직원의 이름과 담당 매니저 이름, 담당 매니저 직위를 출력하시오(셀프조인)

```
SELECT A.EMP_NAME, B.EMP_NAME, B.position  
FROM EMPLOYEE A JOIN  
EMPLOYEE B ON A.MGR_EMP_NO=B.EMP_NO;
```

44. dept 테이블에

- 부서 : 1000, '인사부', '서울' 데이터를 입력한 후 Select 구문으로 전체 dept 테이블 내용을 검색한 후 RollBack 하는 과정을 쿼리로 작성하고 실행 결과를 첨부하시오.

```
mysql> start transaction;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> insert into dept values(1000,'인사부','서울');  
Query OK, 1 row affected (0.00 sec)  
  
mysql> select * from dept;  
+-----+-----+-----+  
| dep_no | dep_name | loc |  
+-----+-----+-----+  
|      10 | 총무부   | 서울 |  
|      20 | 영업부   | 부산 |  
|      30 | 전산부   | 대전 |  
|      40 | 재무부   | 광주 |  
|     1000 | 인사부   | 서울 |  
+-----+-----+-----+  
5 rows in set (0.00 sec)  
  
mysql> rollback;  
Query OK, 0 rows affected (0.00 sec)
```

45. salary_grade 테이블에 bigo 필드를 맨 뒤에 삽입하시오. (필드 타입 : varchar(255))

```
alter table salary_grade  
add column bigo varchar(255) after max_sal;
```