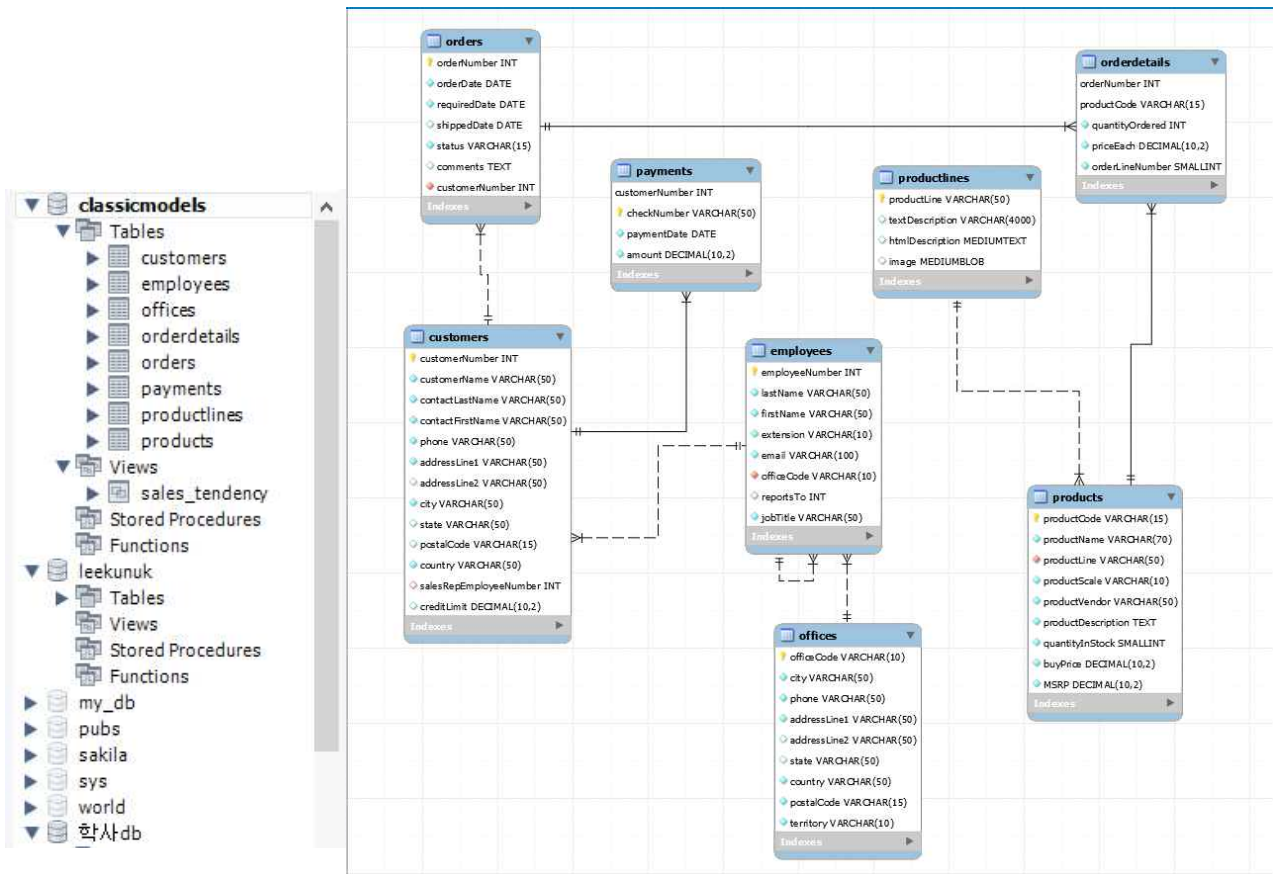


# A. MySQL 관련 문제

1.



2-a.

```
3      /*(a) 영국(UK)에 있는 도시에 근무하는 직원들의 이름(firstName + lastName)을 표시시오.*/
4      • select concat(e.firstName, e.lastName) as fullName
5      from employees e, offices o
6      where e.officeCode = o.officeCode
7      and o.country = 'UK';
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
fullName				
LarryBott				
BarryJones				

2-b.

```
/*(b) lastName이 'Patterson'인 직원이 담당하는 고객회사들의 이름(customerName)을 구하시오. */
• select c.customerName
  from customers c, employees e
 where c.salesRepEmployeeNumber = e.employeeNumber
 and e.lastName = 'Patterson';
```

customerName
Diecast Classics Inc.
Auto-Moto Classics Inc.
Marta's Replicas Co.
Gifts4AllAges.com
Online Diecast Creations Co.
FunGiftIdeas.com

Result 30

2-c.

```
/*(c) 담당 직원이 배정되지 않은 고객회사의 이름을 구하시오. */
select customerName
  from customers
 where salesRepEmployeeNumber is null;
```

customerName
Havel & Zbyszek Co
Porto Imports Co.
Asian Shopping Network, Co
Natürlich Autos
ANG Resellers
Messner Shopping Network
Franken Gifts, Co
BG&E Collectables
Schuyler Imports
Der Hund Imports
Cramer Spezialitäten, Ltd
Asian Treasures, Inc.

customers 35

## 2-d.

```
/*(d) 고객 회사들에 대해 매출 성향을 분석하려고 한다. 각 고객 회사에 대해 회사명, 주문 회수, 평균 주문 금액, 최대 주문 금액을 구하시오.*/
select c.customerName, count(*) as order_count,
       avg(p.amount) as average_amount, max(p.amount) as max_amount
from customers c, payments p
where c.customerNumber = p.customerNumber
group by c.customerName;
```

	customerName	order_count	average_amount	max_amount
▶	Atelier graphique	3	7438.120000	14571.44
	Signal Gift Stores	3	26726.993333	33347.88
	Australian Collectors, Co.	4	45146.267500	82261.22
	La Rochelle Gifts	3	38983.226667	49523.67
	Baane Mini Imports	4	26056.197500	50218.95
	Mini Gifts Distributors Ltd.	9	64909.804444	111654.40
	Blauer See Auto, Co.	4	18984.440000	33820.62
	Mini Wheels Co.	3	22236.853333	26248.78
	Land of Toys Inc.	3	35879.980000	50025.35
	Euro+ Shopping Channel	13	55056.844615	120166.58
	Volvo Model Replicas, Co	2	21840.325000	36005.71
	Danish Wholesale Imports	4	26861.625000	53959.21

Result 36 x

## 2-e.

```
/*(e) 가장 많은 주문 금액의 주문의 고객회사 명, 주문 날짜, 주문금액을 구하시오.*/
select c.customerName, o.orderDate, p.amount
from orders o, payments p, customers c
where o.customerNumber=p.customerNumber
and c.customerNumber=p.customerNumber
and p.amount = (select max(amount) from payments)
and p.paymentDate between o.orderDate and o.shippedDate;
```

	customerName	orderDate	amount
▶	Euro+ Shopping Channel	2005-03-15	120166.58

Result 34 x

### 3.

\*/

3. 2의 (d)를 View로 구현하고 View에 대한 조회의 예를 보이시오.

View의 정의와 조회 결과를 출력해서 제출하시오.

\*/

```
create view sales_tendency as
select c.customerName, count(*) as order_count,
       avg(p.amount) as average_amount, max(p.amount) as max_amount
from customers c, payments p
where c.customerNumber = p.customerNumber
group by c.customerName;
```

```
select * from sales_tendency;
```

	customerName	order_count	average_amount	max_amount
▶	Atelier graphique	3	7438.120000	14571.44
	Signal Gift Stores	3	26726.993333	33347.88
	Australian Collectors, Co.	4	45146.267500	82261.22
	La Rochelle Gifts	3	38983.226667	49523.67
	Baane Mini Imports	4	26056.197500	50218.95
	Mini Gifts Distributors Ltd.	9	64909.804444	111654.40
	Blauer See Auto, Co.	4	18984.440000	33820.62
	Mini Wheels Co.	3	22236.853333	26248.78
	Land of Toys Inc.	3	35879.980000	50025.35
	Euro+ Shopping Channel	13	55056.844615	120166.58
	Volvo Model Renting Co.	2	71840.325000	36005.71

### 4.

```
1 • insert into customers values
2   (497, 'Dankook Database', 'Lee', 'KunUk',
3     '01054845421', 'Jukjeonro', NULL,
4     'Yongin', 'Gyeonggi', '00000', 'South Korea', 1286, 1000.00);
5
6 • insert into orders values
7   (10426, '2020-09-13', '2020-09-20', null, 'In Process', NULL, 497);
8 • insert into orderdetails values
9   (10426, 'S72_3212', 10, 46.10, 12);
10 • insert into orderdetails values
11   (10426, 'S700_4002', 10, 75.20, 2);
12
13 • select c.customerName, o.orderNumber, od.productCode, od.priceEach
14   from customers c, orders o, orderdetails od
15  where c.customerName = 'Dankook Database'
16        and c.customerNumber = o.customerNumber
17        and o.orderNumber = od.orderNumber;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	customerName	ordernumber	productCode	priceEach
▶	Dankook Database	10426	S700_4002	75.20
	Dankook Database	10426	S72_3212	46.10

## B. MS Sql 문제

### 1. 고친부분을 빨간색으로 표시 했습니다.

```
/* 본 과제는 간단한 수준의 학사DB를 구성하는 내용으로
   student, course, department, instructor, course_taken으로
   구성된다. 본 과제를 통해 학생들은 테이블 간의 참조 관계를 고려한 테이블들의 생성,
   그리고 자기 참조 테이블의 경우 데이터들 간의 참조 관계를 고려한 데이터 입력을 경험한다.
   또한 기본 키, 외래 키를 포함한 주요 제약조건 constraint의 이름을
   명시하여 오류 발생 시 어떤 오류인지 확인할 수 있도록 한다.
*/

/* 데이터베이스의 생성 */
IF NOT EXISTS (select name from sys.databases where name = N'학사DB')
drop database if exists 학사DB;
CREATE DATABASE 학사DB;

/* department 테이블의 생성과 데이터 입력 */
IF NOT EXISTS (select * from information_schema.tables where table_name='department')
CREATE TABLE department (
    id CHAR(10) NOT NULL,
    name CHAR(10),
    constraint pk_department PRIMARY KEY (id)
)

INSERT INTO department (id, name) VALUES
    ('cs', '전산전공'),
    ('ss', '통계전공');

/* instructor 테이블의 생성과 데이터 입력 */
IF NOT EXISTS (select * from information_schema.tables where table_name='instructor')
CREATE TABLE instructor (
    pid CHAR(10) NOT NULL,
    name CHAR(10) NOT NULL,
    dept CHAR(10),
    constraint pk_instructor PRIMARY KEY (pid),
    constraint uniq_instructor UNIQUE (name),
    constraint fk_instructor_department foreign key(dept) references department(id)
)

INSERT INTO instructor (pid, name, dept) VALUES
    ('cs10', '구자영', 'cs'),
    ('cs11', '우진운', 'cs'),
```

```
('cs12', '유해영', 'cs'),  
('cs13', '이석균', 'cs'),  
('cs14', '조경산', 'cs'),  
('cs15', '조성제', 'cs'),  
('ss16', '이강섭', 'ss'),  
('ss17', '황형태', 'ss'),  
('ss18', '이장택', 'ss');
```

```
/* course 테이블의 생성과 데이터 입력 */
```

```
IF NOT EXISTS (select * from information_schema.tables where table_name='course')
```

```
CREATE TABLE course (  
    id CHAR(10) NOT NULL,  
    name CHAR(20),  
    instructor CHAR(10),  
    prerequisite CHAR(10),  
    PRIMARY KEY (id),  
    Constraint fk_Course_Instructor foreign key(instructor) references instructor(pid),  
    CONSTRAINT fk_Prerequisite_Course foreign key(prerequisite)  
        references course(id)  
)
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('cs111', '기초전산', 'cs13', NULL);
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('ss111', '전산통계', 'ss18', NULL);
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('cs211', '수치해석', 'cs12', 'cs111');
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('cs221', '자료구조론', 'cs11', 'cs111');
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('cs222', '시스템프로그래밍', 'cs10', 'cs111');
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('cs311', '컴퓨터 구조론', 'cs14', 'cs111');
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('cs312', '알고리즘', 'cs11', 'cs221');
```

```
INSERT INTO course (id, name, instructor, prerequisite)  
VALUES ('cs321', '프로그래밍언어론', 'cs13', 'cs221');
```

```
INSERT INTO course (id, name, instructor, prerequisite)
VALUES ('cs322', '운영체제', 'cs15', 'cs222');
```

```
INSERT INTO course (id, name, instructor, prerequisite)
VALUES ('cs411', '데이터베이스', 'cs13', 'cs321');
```

```
INSERT INTO course (id, name, instructor, prerequisite)
VALUES ('cs413', '컴퓨터네트워크', 'cs14', 'cs311');
```

```
INSERT INTO course (id, name, instructor, prerequisite)
VALUES ('ss311', '응용해석학', 'ss17', 'ss111');
```

```
INSERT INTO course (id, name, instructor, prerequisite)
VALUES ('ss312', '통계적 품질관리', 'ss16', 'ss111');
```

```
INSERT INTO course (id, name, instructor, prerequisite)
VALUES ('cs421', '소프트웨어 공학', 'cs12', 'cs312');
```

```
INSERT INTO course (id, name, instructor, prerequisite)
VALUES ('ss321', '회귀분석', 'ss18', 'ss312');
```

```
/* student 테이블의 생성과 데이터 입력 */
```

```
IF NOT EXISTS (select * from information_schema.tables where table_name='student')
```

```
CREATE TABLE student (
    id CHAR(10) NOT NULL,
    name CHAR(10) NOT NULL,
    major CHAR(10) DEFAULT NULL,
    address CHAR(30) DEFAULT '단국대학교',
    gpa FLOAT,
    bdate date,
    constraint pk_student PRIMARY KEY (id),
    constraint fk_student_department foreign key(major) references department(id)
)
```

```
INSERT INTO student (id, name, major, address, gpa, bdate) VALUES
('930405', '한나라', 'cs', '서울 마포구 원효로', 3.299999952, '1974-06-12'),
('940123', '강동희', 'ss', '서울 중구 필동', 3.625, '1975-08-07'),
('950564', '허영만', 'cs', '서울 강동구 풍납동', 1.75, '1976-12-21'),
('960157', '이동주', 'cs', '서울 서초구 잠원동', 1.799999952, '1977-10-10'),
('970734', '조용필', NULL, '서울 영등포구 영등포동', 2, '1978-07-12'),
('980115', '이미숙', 'ss', '서울 서초구 반포동', 3.75, NULL),
('980397', '조용기', NULL, '서울 서대문구 홍은동', 2.25, NULL);
```



```

/* course_taken 테이블의 생성과 데이터 입력 */
IF NOT EXISTS (select * from information_schema.tables where table_name='course_taken')
CREATE TABLE course_taken (
    no INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    sid CHAR(10),
    cid CHAR(10),
    grade FLOAT,
    year_taken INT,
    foreign key(sid) references student(id),
    CONSTRAINT fk_CourseTaken_Course foreign key(cid) references course(id)
)

```

```

INSERT INTO course_taken (sid, cid, grade, year_taken) VALUES

```

```

('930405', 'cs111', 2.0, 1993),
('930405', 'cs211', 4.0, 1996),
('930405', 'cs221', 3.0, 1996),
('930405', 'cs222', 3.0, 1996),
('930405', 'cs311', 3.0, 1997),
('930405', 'cs321', 4.0, 1997),
('930405', 'cs411', 4.0, 1998),
('940123', 'ss111', 2.0, 1994),
('940123', 'cs111', 4.0, 1997),
('940123', 'cs221', 4.0, 1997),
('940123', 'ss311', 4.0, 1997),
('940123', 'ss312', 4.0, 1998),
('940123', 'ss321', 3.0, 1998),
('950564', 'cs111', 2.0, 1995),
('950564', 'cs211', 2.0, 1996),
('950564', 'cs222', 1.0, 1997),
('950564', 'cs311', 2.0, 1998),
('950564', 'cs411', 2.0, 1999),
('960157', 'cs111', 1.0, 1996),
('960157', 'cs211', 2.0, 1997),
('970734', 'cs111', 1.0, 1997),
('970734', 'cs211', 3.0, 1998),
('970734', 'cs222', 2.0, 1998),
('980115', 'ss111', 4.0, 1998),
('980115', 'cs111', 3.0, 1998),
('980115', 'cs221', 3.0, 1998),
('980115', 'cs222', 4.0, 1998),
('980115', 'cs311', 4.0, 1999),
('980397', 'cs111', 2.0, 1998),
('980397', 'cs211', 2.0, 1999);

```

go



```
1 | select * from department;
```

	id	name
1	cs	전산전공
2	ss	통계전공

```
1 | select * from instructor;
```

	pid	name	dept
1	cs10	구자영	cs
2	cs11	우진운	cs
3	cs12	유해영	cs
4	cs13	이석균	cs
5	cs14	조경산	cs
6	cs15	조성제	cs
7	ss16	이강섭	ss
8	ss17	황형대	ss
9	ss18	이장택	ss

```
1 | select * from course;
```

	id	name	instructor	prerequisite
1	cs111	기초전산	cs13	NULL
2	cs211	수치해석	cs12	cs111
3	cs221	자료구조론	cs11	cs111
4	cs222	시스템프로그래밍	cs10	cs111
5	cs311	컴퓨터 구조론	cs14	cs111
6	cs312	알고리즘	cs11	cs221
7	cs321	프로그래밍언어론	cs13	cs221
8	cs322	운영체제	cs15	cs222
9	cs411	데이터베이스	cs13	cs321
10	cs413	컴퓨터네트워크	cs14	cs311
11	cs421	소프트웨어 공학	cs12	cs312
12	ss111	전산통계	ss18	NULL
13	ss311	응용해석학	ss17	ss111
14	ss312	통계적 품질관리	ss16	ss111
15	ss321	회귀분석	ss18	ss312

```
1 | select * from student;
```

	id	name	major	address	gpa	bdate
1	930405	한나라	cs	서울 마포구 원효로	3.299999952	1974-06-12
2	940123	강동희	ss	서울 중구 필동	3.625	1975-08-07
3	950564	허영만	cs	서울 강동구 풍납동	1.75	1976-12-21
4	960157	이동주	cs	서울 서초구 잠원동	1.799999952	1977-10-10
5	970734	조용필	NULL	서울 영등포구 영등포동	2	1978-07-12
6	980115	이미숙	ss	서울 서초구 반포동	3.75	NULL
7	980397	조용기	NULL	서울 서대문구 홍은동	2.25	NULL

```
1 | select * from course_taken;
```

	no	sid	cid	grade	year_taken
1	1	930405	cs111	2	1993
2	2	930405	cs211	4	1996
3	3	930405	cs221	3	1996
4	4	930405	cs222	3	1996
5	5	930405	cs311	3	1997
6	6	930405	cs321	4	1997
7	7	930405	cs411	4	1998
8	8	940123	ss111	2	1994
9	9	940123	cs111	4	1997
10	10	940123	cs221	4	1997
11	11	940123	ss311	4	1997
12	12	940123	ss312	4	1998
13	13	940123	ss321	3	1998
14	14	950564	cs111	2	1995
15	15	950564	cs211	2	1996
16	16	950564	cs222	1	1997
17	17	950564	cs311	2	1998
18	18	950564	cs411	2	1999
19	19	960157	cs111	1	1996
20	20	960157	cs211	2	1997
21	21	970734	cs111	1	1997
22	22	970734	cs211	3	1998
23	23	970734	cs222	2	1998
24	24	980115	ss111	4	1998
25	25	980115	cs111	3	1998
26	26	980115	cs221	3	1998
27	27	980115	cs222	4	1998
28	28	980115	cs311	4	1999
29	29	980397	cs111	2	1998
30	30	980397	cs211	2	1999

## 2-a.

```

1  /*
2  (a) 이석균 교수가 강의한 과목을 수강한 학생들의
3  이름과 수강 과목명 그리고 소속 전공 명을 구하라.
4  전공이 미정인 학생들로 포함하시오.
5  */
6
7  select s.name, s.major, d.name, c.name
8  from course c, course_taken ct, student s left join department d on d.id = s.major
9  where s.id = ct.sid
10 and c.id = ct.cid
11 and ct.cid
12 in (select id from course where instructor =
13 (select pid from instructor where name = '이석균'))
14 go

```

100 %

결과 메시지

	name	name	name
1	한나라	전산전공	기초전산
2	한나라	전산전공	프로그래밍언어론
3	한나라	전산전공	데이터베이스
4	강동희	통계전공	기초전산
5	허영만	전산전공	기초전산
6	허영만	전산전공	데이터베이스
7	이동주	전산전공	기초전산
8	이미숙	통계전공	기초전산

## 2-b.

```

1  /*
2  (b) 통계 학과에서 개설된 과목(통계학 과목)을 한 과목도
3  수강하지 않은 학생의 이름을 구하라.
4  */
5  select st.name
6  from course_taken ct, student st
7  where ct.sid = st.id
8  and ct.sid in (select sid from course_taken where cid like 'cs%')
9  and ct.sid not in (select sid from course_taken where cid like 'ss%')
10 group by st.name
11 go

```

100 %

결과 메시지

	name
1	이동주
2	조용기
3	조용필
4	한나라
5	허영만

2-c.

```
1  /*
2  (c) 기초전산과 데이터베이스를 둘 다 수강한 학생들의 이름을 구하라.
3  */
4  select s.name
5  from student s right join course_taken ct on s.id = ct.sid
6  where ct.cid in (select id from course where name in ('기초전산', '데이터베이스'))
7  group by s.name
8  having count(*) = 2;
9  go
```

100 %

결과 메시지

	name
1	한나라
2	허영만