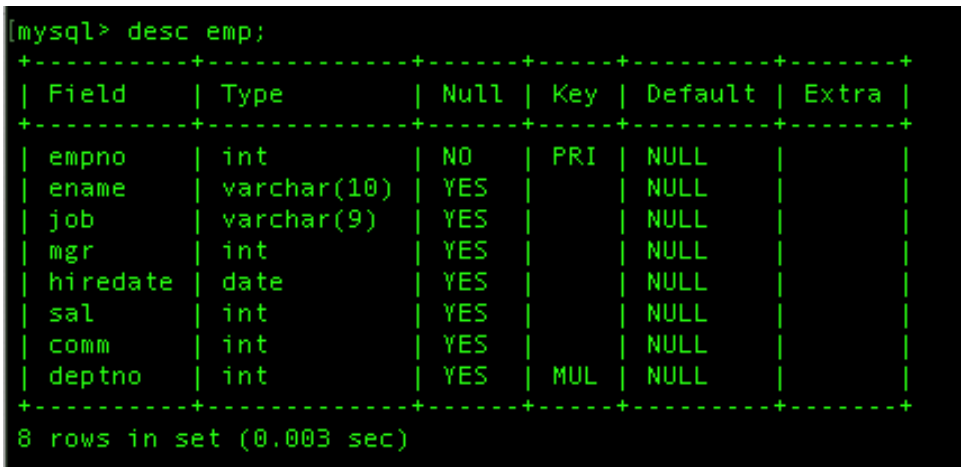
Emp, Dept 테이블로 구성된 회사 사원에 관한 데이터베이스를 만들고자 한다. 테이블을 생성하고 데이터를 입력하는 SQL 문을 작성하시오.

1) 부서(department)에 관한 Dept 테이블은 deptno(부서번호), dname(부서이름), loc(부서위치, location)으로 구성되어 있다. Dept 테이블을 생성하는 SQL 명령어는 다음과 같다. Dept 테이블을 생성해보자

2) 사원(employee)에 관한 Emp 테이블은 empno(사원번호), ename(사원이름), job(업무), mgr(직송상사번호, manager), hiredate(고용날짜), sal(월급여, salary), comm(판매수당, commission), deptno(부서번호)로 구성되어 있다. Emp의 deptno는 Dept의 deptno를 참조하는 외래키로 지정한다. Desc Emp; 명령을 사용하면 테이블의 구조를 볼 수 있다. Emp 테이블을 생성하는 SQL 문을 작성해보자

```
CREATE TABLE Emp (
        empno INT PRIMARY KEY,
        ename VARCHAR(10),
        job VARCHAR(9),
        mgr INT,
        hiredate DATE,
        sal INT
        comm INT
        deptno INT

        FOREIGN KEY (deptno) REFERENCES Dept(deptno)
        FOREIGN KEY (mgr) REFFERENCES Dept(deptno)
);
```



3) 부서에 관한 다음 네 개의 데이터를 삽입하시오

1- INSERT INTO Dept (deptno, dname, loc) VALUES(10, 'ACCOUNTING', 'NEWYORK');

2- INSERT INTO Dept (deptno, dname, loc) VALUES(20, 'RESEARCH', 'DALLAS');

3- INSERT INTO Dept (deptno, dname, loc) VALUES(30, 'SALES', CHICAGO');

4- INSERT INTO Dept (deptno, dname, loc) VALUES(40, 'OPERATIONS', 'BOSTON');

```
mysql> select * from dept;
+--------+------------+---------+
| deptno | dname      | loc     |
+--------+------------+---------+
|     10 | ACCOUNTION | NEWYORK |
|     20 | RESEARCH   | DALLAS  |
|     30 | SALES      | CHICAGO |
|     40 | OPERATIONS | BOSTON  |
+--------+------------+---------+
4 rows in set (0.001 sec)
```

4) 사원에 관한 다음 네 개의 데이터를 삽입하시오

1- INSERT INTO Emp(empno, ename, job, mgr, hiredate, sal, comm, deptno)
VALUES(7369, 'SMITH', 'CLERK', 7902, '1980-12-17 00:00:00', 800, NULL, 20);

2- INSERT INTO Emp(empno, ename, job, mgr, hiredate, sal, comm, deptno)
VALUES(7499, 'ALLEN', 'SALESMAN', 7698, '1981-02-20 00:00:00', 1600, 300, 30);

3- INSERT INTO emp(empno, ename, job, mgr, hiredate, sal, comm, deptno)
VALUES(7521, 'WARD', SALESMAN', 7698, '1981-02-22 00:00:00', 1250, 500, 30);

4- INSERT INTO emp(empno, ename, job, mgr, hiredate, sal, comm, deptno)
VALUES(7566, 'JONES', 'MANAGER', 7839, '1981-04-02 00:00:00', 2975, NULL, 20);

```
mysql> select * from emp;
+-------+-------+----------+------+------------+------+------+--------+
| empno | ename | job      | mgr  | hiredate   | sal  | comm | deptno |
+-------+-------+----------+------+------------+------+------+--------+
|  7369 | SMITH | CLERK    | 7902 | 1980-12-17 |  800 | NULL |     20 |
|  7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | WARD  | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7566 | JONES | MANAGER  | 7839 | 1981-04-02 | 2975 | NULL |     20 |
+-------+-------+----------+------+------------+------+------+--------+
4 rows in set (0.001 sec)
```

5) 사원 테이블에 다음 데이터를 삽입하려고 하니 오류가 발생하였다. 오류 메시지를 확인해보고 원인을 찾으시오

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`exam`.`emp`, CONSTRAINT `emp_ibfk_1` FOREIGN KEY (`deptno`) REFERENCES `dept` (`deptno`))

Emp 테이블의 deptno 열은 dept 테이블의 deptno 열을 외래키로 참조하고 있다.
emp.deptno에 들어가는 값은 반드시 dept 테이블에 존재해야 한다.
deptno =50은 dept 테이블에 존재하지 않기 때문에 오류가 발생한다.

6) 사원의 이름과 근무지역을 출력하는 다음 질의를 수행해 보시오

```
mysql> SELECT ename, loc FROM Emp, Dept WHERE Emp.deptno=Dept.deptno;
+-------+---------+
| ename | loc     |
+-------+---------+
| SMITH | DALLAS  |
| ALLEN | CHICAGO |
| WARD  | CHICAGO |
| JONES | DALLAS  |
+-------+---------+
```

7) 부서(Dept) 테이블의 구조를 변경하여 부서장의 이름을 저장하는 managername 속성을 추가하고자 한다. ALTER 문을 사용하여 작성해 보시오. managername 속성이 만들어졌으면 UPDATE 문을 이용하여 manager 이름을 넣어보시오

ALTER TABLE Dept ADD managername VARCHAR(14);

```
mysql> desc dept;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| deptno      | int         | NO   | PRI | NULL    |       |
| dname       | varchar(14) | YES  |     | NULL    |       |
| loc         | varchar(13) | YES  |     | NULL    |       |
| managername | varchar(14) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

₩UPDATE dept SET managername = 'Park' WHERE deptno = 10;

```
mysql> SELECT * FROM dept;
+--------+------------+---------+-------------+
| deptno | dname      | loc     | managername |
+--------+------------+---------+-------------+
|     10 | ACCOUNTION | NEWYORK | Park        |
|     20 | RESEARCH   | DALLAS  | NULL        |
|     30 | SALES      | CHICAGO | NULL        |
|     40 | OPERATIONS | BOSTON  | NULL        |
+--------+------------+---------+-------------+
4 rows in set (0.001 sec)
```

[마당서점 데이터 베이스 확장 1]

1) 고객번호 1번의 주소 변경 내역을 모두 나타내시오

SELECT *FROM Cust_addr WHERE Custid=1;

2) 고객번호 1번의 전화번호 변경 내역을 모두 나타내시오

SELECT * Changeday, Phone FROM Cust_addr WHERE Custid=1;

3) 고객번호 1번의 가입 당시 전화번호를 나타내시오 단, 가입 당시 전화번호는 주소 이력(history) 중 가장 오래된 것을 찾는다. 주소 변경 이력이 없으면 현재 주소를 반환한다.

SELECT * Phone FROM Cust_addr WHERE Custid=1 ORDER BY Changeday

4) 고객번호 1번의 '2024년01월01일' 당시 전화번호를 나타내시오 단, 주소 이력 중 changeday 속성값이 '2024년02월01일'보다 오래된 첫 번째 값을 찾는다.

SELECT Phone FROM Cust_addr WHERE Custid=1 AND Changeday <= DATE '2024-01-01 ORDER BY Changeday DESC;

[마당서점 데이터 베이스 확장 2]

1) 고객번호 1번의 cart에 저장된 도서 중 주문한 도서를 구하시오

SELECT c.Bookid FROM Cart c WHERE c.Custid=1 AND EXISTS(SELECT 1 FROM Orders o WHERE o.Custid = c.Custid AND o.Bookid = c.Bookid);

2) 고객번호 1번의 cart에 저장된 도서 중 주문하지 않는 도서를 구하시오

SELECT c.bookid FROM Cart c WHERE c.Custid =1 AND NOT EXISTS(SELECT 1 FROM

Orders o WHERE o.Custid = c.Custid AND o.Bookid = c.Bookid);

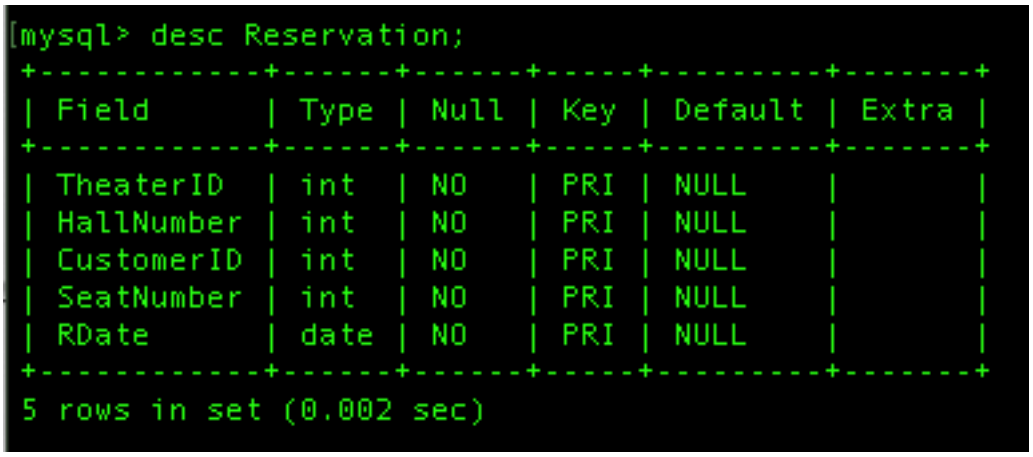
3) 고객번호 1번의 cart에 저장된 도서의 정가의 합을 구하시오
SELECT SUM(b.Price) AS Total FROM Cart c JOIN Book b ON c.Bookid = b.Bookid WHERE c.Custid=1;

[극장 데이터베이스]
다음은 지점이 세 개인 극장의 데이터베이스다. 밑줄 친 속성은 기본키다.
테이블의 구조를 만들고 데이터를 입력한 후 다음 질의에 대한 SQL 문을 작성하시오

```
mysql> desc Theater;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| TheaterID | int         | NO   | PRI | NULL    |       |
| Name      | varchar(20) | YES  |     | NULL    |       |
| Location  | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.002 sec)
```

```
mysql> desc Customer;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| CustomerID | int         | NO   | PRI | NULL    |       |
| Name       | varchar(20) | YES  |     | NULL    |       |
| Address    | varchar(50) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.002 sec)
```

```
mysql> desc Hall;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| TheaterID  | int         | NO   | PRI | NULL    |       |
| HallNumber | int         | NO   | PRI | NULL    |       |
| MovieTitle | varchar(50) | YES  |     | NULL    |       |
| Price      | int         | YES  |     | NULL    |       |
| Seats      | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.003 sec)
```



```
mysql> desc Reservation;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| TheaterID  | int  | NO   | PRI | NULL    |       |
| HallNumber | int  | NO   | PRI | NULL    |       |
| CustomerID | int  | NO   | PRI | NULL    |       |
| SeatNumber | int  | NO   | PRI | NULL    |       |
| RDate      | date | NO   | PRI | NULL    |       |
+------------+------+------+-----+---------+-------+
5 rows in set (0.002 sec)
```

단순 질의

1) 모든 극장의 이름과 위치를 나타내시오

SELECT Name, Location FROM Theater;

2) '잠실'에 있는 극장을 나타내시오

SELECT * FROM Theater WHERE Location='잠실';

3) '잠실'에 사는 고객의 이름을 오름차순으로 나타내시오

SELECT Name FROM Customer WHERE Address='잠실' ORDER BY Name;

4) 가격이 8,000원 이하인 영화의 극장번호, 상영관번호, 영화제목을 나타내시오

SELECT TheaterID, HallNumber, MovieTitle FROM Hall WHERE Price<=8000;

5) 극장 위치와 고객의 주소가 같은 고객을 나타내시오

SELECT DISTINCT c.*
FROM Customer c

집계 질의

1) 극장의 수는 몇 개인가?

SELECT COUNT(*) AS Theater FROM Theater;

2) 상영되는 영화의 평균 가격은 얼마인가?

SELECT AVG(Price) AS Price FROM Hall;

3) 2024년 9월 1일에 영화를 관람한 고객의 수는 얼마인가?

SELECT COUNT(DISTINCT CustomerID) AS View FROM Reservation WHERE RDate='2024-09-01';

부속질의와 조인

1) '대한' 극장에서 상영된 영화제목을 나타내시오

```
SSELECT MovieTitle
FROM Hall
WHERE TheaterID = (
    SELECT TheaterID FROM Theater
    WHERE Name = '대한'
);
```

2) '대한' 극장에서 영화를 본 고객의 이름을 나타내시오

```
SELECT DISTINCT c.Name
FROM Reservation r
JOIN Customer c ON r.CustomerID = c.CustomerID
WHERE r.TheaterID = (
    SELECT TheaterID FROM Theater
    WHERE Name = '대한'
);
```

3) '대한' 극장의 전체 수입을 나타내시오

```
SELECT SUM(h.Price) AS TotalRevenue
FROM Reservation r
JOIN Hall h ON r.TheaterID = h.TheaterID AND r.HallNumber = h.HallNumber
WHERE r.TheaterID = (
    SELECT TheaterID FROM Theater
    WHERE Name = '대한'
);
```

그룹 질의

1) 극장별 상영관 수를 나타내시오

```
SELECT TheaterID, COUNT(*) AS NumHalls
FROM Hall
GROUP BY TheaterID;
```

2) '잠실'에 있는 극장의 상영관을 나타내시오

```
SELECT h.*
FROM Hall h
JOIN Theater t ON h.TheaterID = t.TheaterID
WHERE t.Location = '잠실';
```

3) 2024년 9월 1일의 극장별 평균 관람 고객 수를 나타내시오

```
SELECT TheaterID, COUNT(*) * 1.0 / COUNT(DISTINCT HallNumber) AS AvgViewersPerHall
FROM Reservation
WHERE RDate = '2024-09-01'
GROUP BY TheaterID;
```

4) 2024년 9월 1일에 가장 많은 고객이 관람한 영화를 나타내시오

```
SELECT h.MovieTitle
FROM Reservation r
JOIN Hall h ON r.TheaterID = h.TheaterID AND r.HallNumber = h.HallNumber
WHERE r.RDate = '2024-09-01'
GROUP BY h.MovieTitle
ORDER BY COUNT(*) DESC
FETCH FIRST 1 ROWS ONLY;
```

DML

1) 각 테이블에 데이터를 삽입하는 INSERT 문을 하나씩 실행시켜 보시오

```
INSERT INTO Theater VALUES (1, '롯데', '잠실');
INSERT INTO Theater VALUES (2, '메가', '강남');
INSERT INTO Theater VALUES (3, '대한', '잠실');

INSERT INTO Hall VALUES (1, 1, '어려운영화', 15000, 48);
INSERT INTO Hall VALUES (3, 1, '멋진영화', 7500, 120);
INSERT INTO Hall VALUES (3, 2, '재밌는영화', 8000, 110);

INSERT INTO Customer VALUES (3, '홍길동', '강남');
INSERT INTO Customer VALUES (4, '김철수', '잠실');
INSERT INTO Customer VALUES (9, '박영희', '강남');

INSERT INTO Reservation VALUES (3, 2, 3, 15, '2024-09-01');
```

```
INSERT INTO Reservation VALUES (3, 1, 4, 16, '2024-09-01');
INSERT INTO Reservation VALUES (1, 1, 9, 48, '2024-09-01');
```

2) 영화 가격을 10%씩 인상하시오

```
UPDATE Hall
SET Price = Price * 1.1;
```