2022-1 정보시스템설계및구축 기말시험

- 답안에 학번, 성명을 기입하시오.
- 문제 순서대로 답안을 작성하시오.
- 수기로(손글씨로) 답안을 작성하지 마시오.
- 답안은 PDF 로 변환하여 제출하시오.
- 7번 문제는 작성된 ER Diagram을 화면챕쳐하여 "학번.JPG" 파일로 만들어 PDF 파일과 함께 제출하시오.
- 시험진행 시간: 11:00 ~ 11:59:59

1. 정규화(Normalization)의 목적을 한 문장으로 간략하게 설명하시오. (5점)

2. 아래 Employee 테이블의 보유 자격증(License)는 multivalue(다중값)에 해당한다. 즉, 특정 Employee가 보유하고 있는 자격증 리스트에 대한 접근보다 특정 자격증을 소유하고 있는 Employee들에 대한 접근이 빈번하게 이루어진다. 따라서, Employee 테이블은 1차 정규형 조건을 충족하지 못하고 있다. 1차 정규형 조건을 충족하기 위한 적절한 조치를 create table 문들로 제시하시오. (10점)

```
create table Employee {
    Employee_No      char(10),                  // 사원번호
    Name             varchar(20) not null,      // 사원성명
    Address          varchar(50) not null,      // 사원주소
    Hired_Date       date,                      // 입사일
    Cur_Department   char(10),                  // 현재 소속부서
    Cur_Salary       numeric(13,2)              // 금년 연봉
    License          varchar(100),              // 보유 자격증
    Primary key(Employee_No),
    Foreign key(Cur_Department) references Department on Delete Set Null
    // Department 테이블이 존재한다고 전제
}
```

3. 아래 Customer_Order 테이블에 대한 함수적 종속성(Functional Dependence)을 근간으로 BCNF(Boyce-Codd Normal Form)을 만족하는 테이블들을 생성하는 create table 문을 작성하시오. 테이블명은 해당 테이블의 역할에 맞는 것으로 각자 결정하시오. (10점)

```
create table Customer_Order {
        Customer_No             char(10),               // 고객번호
        Order_No                char(10),               // 주문번호
        Customer_Name           varchar(20),            // 고객성명
        Customer_Address        varchar(50),            // 고객주소
        Join_Date               date not null,          // 가입일자
        Membership              char(4) not null,       // 고객등급
        Order_Date              date not null,          // 주문일자
        Product_No              varchar(15),            // 제품번호
        Quantity                int not null,           // 주문수량
        Unit_Price              numeric(13,2) not null  // 제품단가
        Manufacturer_No         char(10) not null,      // 제조사번호
        Manufacturer_Name       varchar(20) not null,   // 제조사명
        Manufacturer_Address    varchar(50),            // 제조사주소
        Amount_in_Stock         int not null,           // 제품재고량
        Discount_Rate           float,                  // 할인률
        Primary key(Customer_No, Order_No, Product_No)
}
```

Customer_No → Customer_Name
Customer_No → Customer_Address
Customer_No → Join_Date
Customer_No → Membership
Order_No → Order_Date
Order_No → Discount_Rate
Product_No → Unit_Price
Product_No → Manufacturer_No
Product_No → Amount_in_Stock
Manufacturer_No → Manufacturer_Name
Manufacturer_No → Manufacturer_Address

- Primary key인 (Customer_No, Order_No, Product_No)는 Customer_Order table의 모든 column들에 대한 결정자이다.
- Foreign key 정의 시 on delete 에 대한 option(cascade, set null, set default, restrict)는 적당한 것으로 선택할 것

4. Employee, Department 테이블이 아래와 같이 생성되었다. 부서(Department)는 과, 부, 본부, 센터 등으로 계층을 이루고 있으며 각 부서마다 상위부서는 하나만 있다. 최상위 부서는 상위부서가 존재하지 않는다. 어떤 부서의 상위 부서가 삭제되었을 경우 해당 부서의 상위부서는 존재하지 않는 것으로 관리한다. 부서별 상위부서를 반영하기 위하여 아래 Department 테이블에 대한 create table 문을 다시 작성하시오. (5점)

```
create table Employee {
        Employee_No     char(10),                   // 사원번호
        Name            varchar(20) not null,       // 사원성명
        Address         varchar(50) not null,       // 사원주소
        Hired_Date      date,                       // 입사일
        Primary key(Employee_No)
}
create Department {
        Department_Code         char(10),           // 부서코드
        Department_Name         varchar(20),        // 부서명
        Budget                  numeric(13,2),      // 예산
        Office_Address          varchar(50),        // 부서주소
        Cur_Manager             char(10),           // 현 부서장
        Primary key(Department_Code),
        Foreign key(Cur_Manager) references Employee on delete set null
}
```

5. Employee, Department 테이블이 아래와 같이 생성되었다. (10점)

```
create table Employee {
        Employee_No     char(10),                   // 사원번호
        Name            varchar(20) not null,       // 사원성명
        Address         varchar(50) not null,       // 사원주소
        Hired_Date      date,                       // 입사일
        Primary key(Employee_No)
}
create Department {
        Department_Code     char(10),               // 부서코드
        Department_Name     varchar(20),            // 부서명
        Budget              numeric(13,2),          // 예산
        Office_Address      varchar(50),            // 사무실주소
        Cur_Manager         char(10),               // 현 부서장
        Primary key(Department_Code),
        Foreign key(Cur_Manager) references Employee on delete set null
}
```

사원의 급여지급 내역과 부서배정 내역을 관리하기 위하여 테이블을 추가하기 위한 create table 문을 작성하시오.

추가 생성한 테이블명은 테이블 역할에 부합하는 것으로 결정하시오. 급여지급 내역은 사원에게 지급한 금액과 지급한 일자를 관리한다. 한 사원에게 같은 날짜에 급여를 2회이상 지급하는 일은 없다. 부서배정 내역은 사원이 배정된 부서와 근무시작일자와 근무종료일자를 관리한다. 한 사원을 같은 날 2개 이상 부서에 배정하는 일은 없다. 하지만, 한 사원이 같은 부서에 2회 이상 배정될 수 있다. 즉, 사원_A를 부서_B에 배정하였다가 다른 부서_C로 옮겼다가 다시 부서_B로 배정할 수 있다. 사원이 퇴사하여 Employee 테이블에서 삭제되었을 경우 그 사원의 급여지급 내역과 부서배정 내역도 함께 삭제되어야 한다. 부서가 Department 테이블에서 삭제되었을 경우 어떤 사원이 해당 부서에 배정되었다는 사실도 함께 삭제되어야 한다.

6. 아래 테이블 내용은 이 테이블에서 등장할 수 있는 모든 경우의 사실들을 나타내는 것으로 전제한다. 아래 테이블로부터 도출되는 함수적 종속성으로 적절하지 않은 것 2개를 선택하시오. (10점)

A	B	C	D
10	3	5	1
3	8	9	5
4	6	9	6
9	2	8	5
10	3	5	1
7	3	3	1
9	2	8	5

(1) C → A
(2) A → B
(3) A → C
(4) B → D
(5) D → B
(6) (B, C) → A

7. 레스토랑에서 제공되는 디저트 관련 데이터 관리를 위한 ER Modeling을 하고자 한다. Visual Paradigm을 이용하여 작성한 ER Diagram을 화면캡쳐 기능을 이용하여 캡쳐한 후 학번.jpg 파일로 만들어 본 문제의 답으로 제출하시오. (vpp 파일을 실행하는 과정에서 오류가 발생할 수도 있어 이렇게 조치함. 한 화면에 잘 들어가도록 ER Diagram 내용을 배치할 것) (50점)

- 레스토랑은 레스토랑번호로 식별되며 레스토랑명과 수용인원, 권역, 전화번호, 개점일자, 영업시작시간, 영업종료시간이 관리된다. 휴무일은 레스토랑이 어떤 날에 휴무하는가에 대한 접근을 위하여 관리된다. (어떤 날에 휴무인 레스토랑을 검색하는 경우는 없는 것으로 전제함)
- 권역은 동, 구, 시, 도 단위로 관리되며 레스토랑의 권역은 동 단위 정보를 가져야 한다. 권역은 동이 소속된 구, 구가 소속된 시, 시가 소속된 도를 접근할 수 있도록 관리되어야 한다. 서울특별시나 부산광역시와 같이 상위 권역이 없는 시가 존재할 수 있다. 상위권역이 있는 경우에는 1개 권역만이 상위권역으로 있다.
- 레스토랑별로 1명의 소유자가 있을 수 있으며 소유자 1명은 1개 이상의 레스토랑을 소유할 수 있다. 아직 소유자가 존재하지 않은 레스토랑이 있을 수 있다. 소유자는 소유자번호로 식별되며 성명, 주소, 생년월일, 전화번호를 가진다. 소유자는 레스토랑을 소유하는 경우에만 소유자로 등록된다. 소유자가 레스토랑을 소유하기 시작한 일자와 지급한 권리금, 매입금이 관리되어야 한다. 레스토랑의 소유관련 내용은 이력정보로 관리하지 않고 현재 소유상황만을 관리한다.
- 디저트는 디저트코드로 식별된다. 디저트별로 디저트명, 디저트에 대한 세부설명, 개당 금액인 단가가 관리되어야 한다. 디저트별로 주요 구성 재료들에 대한 정보를 관리하여야 한다. 재료별로 재료코드와 재료명, 그램당 칼로리가 관리되어 디저트별 총 칼로리를 계산할 수 있도록 한다. 디저트별로 포함된 재료 각각의 함량을 그램으로 관리하여야 한다.
- 디저트는 디저트 카테고리에 속해 있어야 하며 디저트 카테고리는 디저트 카테고리 코드로 식별되고 디저트 카테고리명과 그에 대한 세부설명이 관리되어야 한다. 아직 배정된 디저트가 없는 디저트 카테고리가 존재할 수 있다.
- 어떤 레스토랑에 어떤 디저트가 어느 날, 몇 개 공급되었는가에 대한 정보를 관리하여야 한다. 아직 디저트를 공급받지 않은 레스토랑이 있을 수 있으며 아직 레스토랑에 공급되지 않은 디저트도 존재할 수 있다. 같은 날짜에 동일한 레스토랑에 동일한 디저트가 공급되는 일은 없다. 동일한 디저트가 동일한 레스토랑에 여러 번 공급될 수 있다. 어느날 어느 레스토랑에 공급된 디저트 단위에서 재고량이 관리되어야 한다.
- 디저트별로 0가지 이상의 토핑이 이루어진다. 토핑은 토핑코드로 식별되며 토핑명, 단가가 관리되어야 한다. 한 가지 토핑은 0개 이상의 디저트에서 이루어지며 한 디저트에는 0가지 이상의 토핑이 이루어진다. 토핑별 제공량은 디저트마다 다르다.
- 디저트별로 0가지 이상의 음료가 존재한다. 음료에는 음료코드와 음료명, 단가가 관리되어야 한다. 한 가지 음료는 0가지 이상의 디저트에서 제공되며 한 가지 디저트에는 0가지 이상의 음료가 제공된다. 음료별 제공량은 디저트마다 다르다.
- 각각의 레스토랑은 일반고객 부스, 어린이동반 부스, 반려동물동반 부스, 흡연부스 등 1개 이상의 다양한 부스들로 구성된다. 각 부스는 부스코드로 식별되며 부스명을 가진다. 새로운 부스가 개발되면 이를 별도로 등록할 수 있도록 하여야 한다. 레스토랑별로 각 부스별 수용인원이 관리되어야 한다. 특정일에 레스토랑에 공급된 디저트별로 해당 레스토랑의 어느 부스에서 몇 개의 매출이 발생하였는가가 관리되어야 한다. 이러한 매출은 매출번호로 식별되며 매출일자와 시간도 함께 관리되어야 한다.