**Logical Database Design**

**5963 DB 설계 및 구현2**

**60221329 김예일, 60221331 김재현, 60221342 정윤휴, 60222117 이서현, 60222126 이준규**

**C. Logical Database Design**

**(1) User Requirements & Business Rules 수정**

* 주차관리시스템의 유저로 등록하려면 학생 또는 교직원이어야 하고, 학번 혹은 교번, 이메일, 이름, 휴대폰 번호들, 주소지, 성별을 입력해야 한다. 유저는 비상연락망을 포함한 휴대폰 번호를 여러 개 등록할 수 있다.
* 교직원의 경우 교직원 타입을 입력해야 한다.
* 주차장에 입차 한 모든 차는 유저가 있어야 하고, 차 번호, 차 색상, 차 모델명, 타입, 번호판 타입이 인식되고 저장된다.
* 주차요금은 구매가능 대상, 요금지불방식, 요금계산 단위, 주차요금 id, 가격, 구매고객 대상들이 저장되어야 한다. 구매고객 대상들에는 학생,교직원이 있다.
* 주차관리시스템에 등록된 유저만이 차량을 등록할 수 있고, 등록할 때는 차량인식번호, 차량등록증 번호, 운전면허 타입, 운전면허증 번호를 입력을 해야 한다.
* 주차관리시스템에 등록된 유저는 반드시 한 대 이상의 차량을 등록해야 한다. 한 차량의 소유주는 반드시 한 명이어야 한다.
* 등록되지 않은 차량은 저장하지 않는다.
* 주차관리시스템에 등록된 차는 한개의 할인권씩 구매할 수 있다. 하나의 할인권이 어떠한 차에게도 팔리지 않을 수 있다. 하나의 할인권은 여러 차에게 판매될 수 있다. 할인권을 구매하지 않은 차량이 존재할 수 있다.
* 차가 할인권을 구매할 때는 결제 수단, 결제 날짜 및 시간, 결제 가격이 저장된다.
* 교내 주차장은 주차장 번호, 이름, 주차공간면적, 전기차 충전기 여부, 주소지가 저장되어야 한다.
* 하나의 차량은 무조건 하나의 주차장에 들어가야 한다. 하나의 주차장은 여러 차량을 포함할 수 있다. 그리고 모든 차량은 주차장에 들어가야 하고 어떤 차량도 주차하지 않은 주차장이 존재할 수 있다.
* 차량이 주차장에 입출차 시 날짜 및 시간, 입출차 타입이 저장되어야 한다.
* 각 주차장에 대해 주차자리에 대한 번호, 층수, 주차가능 여부가 저장되어야 한다. 같은 주차장의 각각의 주차자리들은 층과 자리번호의 조합으로 구분될 수 있지만, 주차장은 구분할 수 없다.
* 하나의 주차장은 여러 개의 주차자리를 가질 수 있다. 하나의 주차 자리는 하나의 주차장만 가질 수 있다. 주차장에 포함되지 않은 주차자리는 없고, 주차자리가 없는 주차장은 없다.
* 각 주차관리인에 대해 주차관리인\_id, 이름, 휴대폰 번호, 연봉, 주소가 저장되어야 한다. 주차관리인은 적어도 하나 이상의 주차장을 관리해야하고, 주차장은 적어도 한 명 이상의 주차관리인에 의해 관리되어야 한다. 주차관리인이 관리해야 할 일이 발생할 때 그 날짜에 대한 정보가 저장되어야 한다.

(**2) ERD 수정**

그림, 도표, 스케치, 패턴이(가) 표시된 사진

자동 생성된 설명

**(3) Relational Model**

**[Manager]**

Manager (Mng\_ID, Mng\_Name, phone\_Num, salary, address)

**[Parkinglot]**

ParkingLot (park\_ID, name, address, charging\_station, area)

ParkingLot\_Mng (Mng\_ID, park\_ID, work\_Date)

FOREIGN KEY (Mng\_ID) REFERENCES Manager (Mng\_ID)

FOREIGN KEY (park\_ID) REFERENCES ParkingLot (park\_ID)

ParkingSlot (park\_ID, slotNum, floor, availability)

FOREIGN KEY (park\_ID) REFERENCES ParkingLot (park\_ID)

**[Owner]**

Owner (email, fName, lName, city, street, zipCode, sex)

OwnerPhone (email, phoneNum)

FOREIGN KEY (email) REFERENCES Owner (email)

Student (email, stu\_ID)

FOREIGN KEY (email) REFERENCES Owner(email)

Employee (email, emp\_ID, emp\_description)

FOREIGN KEY (email) REFERENCES Owner(email)

**[Car]**

Car (carNum, carColor, carType, plateType, carModel, date, time, enterType, VIN, registerationNum, licenseNum, licenseType ,park\_ID, email)

FOREIGN KEY (park\_ID) REFERENCES Parking\_Lot(park\_ID)

FOREIGN KEY (email) REFERENCES Owner(email)

**[Payment]**

Payment (Pay\_ID, type, unit, price)

Payment\_Car(carNum, Pay\_ID, payMean, time, date, totalPrice)

FOREIGN KEY(carNum) REFERENCES Car(carNum)

FOREIGN KEY (Pay\_ID) REFERENCES Payment(Pay\_ID)

Pay\_CustomerType (Pay\_ID, customer\_Type)

FOREIGN KEY (Pay\_ID) REFERENCES Payment(Pay\_ID)

**[정규화 과정]**

* + **1NF**

→ 모든 relation의 각 column에는 repeating group이 존재하지 않으므로 1NF에 적합하다.

* + **2NF**
    - ParkingLot\_Mng (Mng\_ID, park\_ID , work\_Date)의 경우, Mng\_ID와 park\_ID 둘 중 하나라도 없으면 workDate가 결정될 수 없다.
    - ParkingSlot (park\_ID, slotNum, floor, availability)의 경우, park\_ID와 slotNum, floor 둘 중 하나라도 없으면 availability가 결정될 수 없다.
    - OwnerPhone (email, phoneNum)의 경우, primary key를 제외한 attribute가 없기 때문에 고려하지 않아도 된다.
    - Pay\_CustomerType (Pay\_ID, customer\_Type) 의 경우, primary key를 제외한 attribute가 없기 때문에 고려하지 않아도 된다.
    - Payment\_Car(CarNum, Pay\_ID, PayMean, time, date, totalPrice)의 경우 primary key를 제외한 attribute가 없기 때문에 고려하지 않아도 된다.

→ 모든 relation의 attribute는 모두 primary key에 완전히 종속되어 있으며, 여러 개의 primary key를 가지고 있는 relation의 attribute는 모든 primary key에 완전히 종속되어 있으므로 2NF에 적합하다.

* + **3NF**

→ primary key를 제외한 attribute간의 이행적 함수 종속 (A가 B를 결정하고, B가 C를 결정하여 A가 C를 결정하는 것)이 존재하는 relation이 존재하지 않으므로 3NF에 적합하다.