2023학년도 2학기

**DB Design Project  
Final Report**

과목명:DB설계및구현2 (5963)

팀원: 60221329 김예일

60221331 김재현

60222117 이서현

60222126 이준규

60221342 정윤휴

담당 교수: 김일주 교수님

**목차**

1. Team Members & Role Description . . . P.3
2. Project Topic & Design Purpose . . . P.3
3. User Requirements & Business Rules . . . P.4
4. Conceptual Data Modeling: Enhanced E-R Diagram . . . P.5
5. Logical DataBase Design: Relational Model . . . P.6
6. Physical DataBase Design: Data Dictionary . . . P.8
7. DDL (Data Definition Language) . . . P.15
8. DML (Data Manipulation Language) . . . P.26
9. System Manual . . . P.28

**1. Team Members & Role Description**

본 팀은 명지대학교 융합소프트웨어학부 응용소프트웨어전공에 2학년으로 재학 중인 김예일, 김재현, 이서현, 이준규, 정윤휴 총 5명의 팀원으로 구성되어 있으며, 5명이 공통적으로 맡은 프로젝트 토픽 선정, E-R Diagram 설계, 10가지의 유용한 SQL Query문 작성 외 각자 맡은 역할은 다음과 같다.

김예일: Relational Model 설계, DB에 들어갈 데이터 구성 및 해당 데이터 삽입 DML 작성

김재현: Relational Model 설계, DDL (Data Definition Language) 작성, Final Report 작성

이서현, 이준규: Relational Model 설계, 각 Table과 Attribute에 대한 설명 작성

정윤휴: Data Dictionary 작성

**2. Project Topic & Design Purpose**

명지대학교 학생들로 이루어진 본 팀은 명지대학교 주차관리소의 데이터 불일치의 가능성이 높고 편리하지 못한 정보 처리 방식에 주목하였다. 현재 명지대학교의 주차 관리 시스템은 이용객 정보 관리 과정에서 대다수의 작업을 수기로 해결한다. 수기로 입력 받은 정보는 excel에 저장한다.

이러한 방식은 졸업한 학생 및 퇴사한 교직원을 관리하기 어렵고, 학생 및 교직원, 운전면허증의 진위여부를 식별하는 작업을 수기로 해결하게 되면 이용객의 정보들을 관리하는 것에 어려움을 겪을 수 있을 뿐만 아니라, 작업 처리 속도가 느려질 수 있다. 따라서 DataBase System을 활용하여 이용객 정보들을 관리해, 업무 처리 속도와 효율, 데이터의 유지/보수성을 향상시키는 것이 이용객 정보 관리에 DataBase System을 도입시키려는 목적이다.

타 학교 주차 공간 대여 서비스는 DB를 사용하여 효율적이고 정확하게 이용객의 정보를 입력 받고, 관리한다. 따라서 현재 명지대학교 주차 관리 시스템은 개선이 필요하고, 타 학교 및 타 주차 시스템이 원활하게 운영되고 있는 것을 보아 합리적이고 가능한 토픽이라고 볼 수 있다.

따라서 비즈니스 목표를 ‘명지대학교 주차관리소의 효율적이고 원활한 이용객 정보 및 주차 공간 관리’로 설정하였다.

프로젝트 토픽과 관련되어 있는 산업은 주차장 운영업이다.

**3. User Requirements & Business Rules**

**< User Requirements >**

1. 주차관리시스템의 사용자로 등록을 하려면 반드시 학생 또는 교직원 중 한 곳에 속해야 하고, 학번 혹은 교번, 이메일, 이름, 휴대폰 번호(들), 주소지, 성별을 입력해야 한다. 주소지는 사는 도시, 도로명, 우편번호로 이루어져 있다. 사용자는 비상연락망을 포함한 휴대폰 번호를 여러 개 등록할 수 있다.
2. 교직원의 경우 교직원의 분류에 대한 설명 (예시: 교수, 교직원)를 입력해야 한다.
3. 주차장에 입차 한 모든 차는 사용자가 있어야 하고, 차의 번호, 색상, 모델명, 차종, 번호판의 종류가 인식되고 저장된다.
4. 주차요금은 구매가능 대상, 요금지불방식, 요금계산 단위, 주차요금 ID값, 가격, 구매고객 대상(들)이 저장되어야 한다. 구매가능 대상에는 학생, 교직원이 있다.
5. 주차관리시스템에 등록된 사용자만이 차량을 등록할 수 있고, 등록할 때는 차량인식번호, 차량등록증 번호, 운전면허의 종류, 운전면허증 번호를 입력해야 한다.
6. 주차관리시스템에 등록된 유저는 반드시 한 대 이상의 차량을 등록해야 한다. 한 차량의 소유주는 반드시 한 명이어야 한다.
7. 등록되지 않은 차량의 정보는 저장하지 않는다.
8. 주차관리시스템에 등록된 차는 주차권을 한 개씩 구매할 수 있고, 개의 주차권이 어떠한 차에게도 팔리지 않을 수 있다. 한 개의 주차권은 여러 대의 차에게 판매될 수 있고, 주차권을 구매하지 않은 차량이 존재할 수 있다.
9. 차가 주차권을 구매할 때는 결제 수단, 결제 날짜 및 시간, 총 결제 가격이 저장된다.
10. 교내 주차장은 주차장 번호, 이름, 주차공간면적 (m2), 전기차 충전기기 존재 여부, 주차장의 주소지가 저장되어야 한다.
11. 한 대의 차량은 무조건 하나의 주차장에 들어가야 한다. 하나의 주차장은 여러 대의 차량을 수용할 수 있다. 모든 차량은 주차장에 들어가야 하고 어떠한 차량도 주차하지 않은 주차장이 존재할 수 있다.
12. 차량이 주차장에 입차하거나 출차할 때 날짜 및 시간, 입차, 출차 중 한 가지의 정보가 저장되어야 한다.
13. 각 주차장에 대해 주차자리에 대한 번호, 층수, 주차가능 여부가 저장되어야 한다. 같은 주차장의 각각의 주차자리들은 층과 자리번호의 조합으로 구분될 수 있지만, 각 주차자리는 주차장 정보 없이 독립적으로 구분이 불가능 하다.
14. 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.
15. 각 주차관리인에 대해 주차관리인 ID, 이름, 휴대폰 번호, 연봉, 주소가 저장되어야 한다. 주차관리인은 적어도 하나 이상의 주차장을 관리해야 하고, 주차장은 적어도 한 명 이상의 주차관리인에 의해 관리되어야 한다. 또한 주차관리인이 주차장을 관리하는 일이 발생했을 때 관리한 그 날짜에 대한 정보가 저장되어야 한다.

**< Business Rules >**

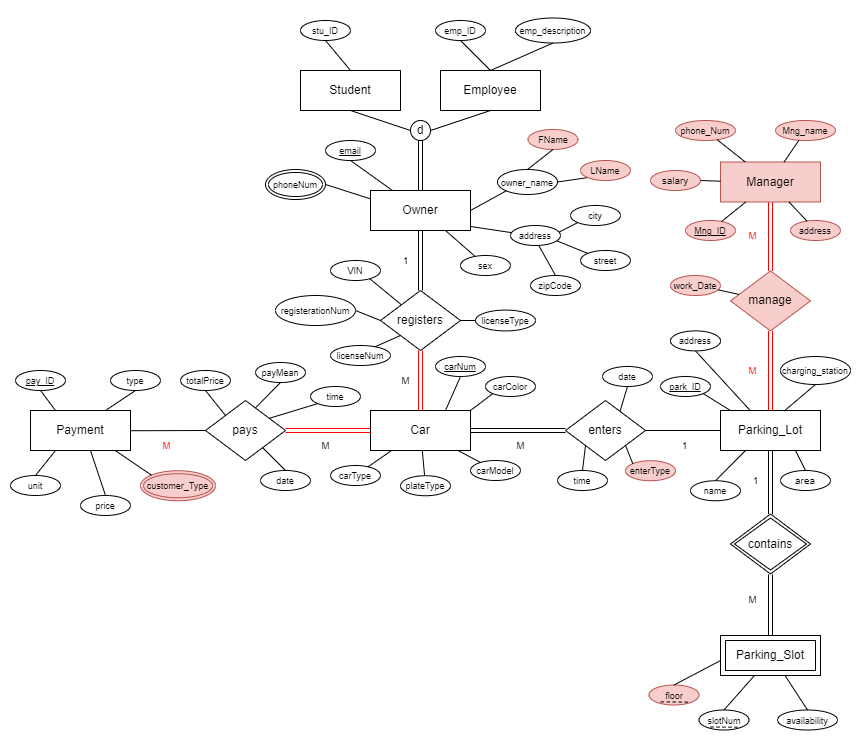
주차 이용권에는 일일권과 정기권이 있고 이는 학생 및 교직원만 구매할 수 있다.

일일권은 학번 및 사번, 이름, 휴대폰 번호, 차량 번호를 수기로 받아 학번 및 사번을 확인한 후 판매한다.

정기권은 학번 및 사번, 이름, 휴대폰 번호, 차량 번호, 학생증, 운전면허증, 차량 등록증을 수기로 받아 학번 및 사번, 차량등록증, 운전면허증 등을 확인한 후 1개월이나, 학기 단위로 판매한다.

**4. Conceptual Data Modeling: Enhanced E-R Diagram**

**< Enhanced E-R Diagram >**



**5. Logical DataBase Design: Relational Model**

**< Relational Model >**

Manager (Mng\_ID, Mng\_Name, phone\_Num, salary, address)  
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 (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 (carNum, carColor, carType, plateType, carModel, enterDate, enterTime, enterType, VIN, registerationNum, licenseNum, licenseType ,park\_ID, email)  
FOREIGN KEY (park\_ID) REFERENCES ParkingLot (park\_ID)  
FOREIGN KEY (email) REFERENCES Owner (email)  
Payment (Pay\_ID, type, unit, price)  
Payment\_Car (carNum, Pay\_ID, payMean, payTime, payDate, 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)

**< Normalization Process >**

**1NF**→ 모든 relation들의 각 column에는 repeating group이 존재하지 않으므로 1NF에 적합하다.

**2NF**

relation들이 2NF를 만족하는 지 확인하기 위해서는 Partial Functional Dependency가 있는 지 확인하면 되기 때문에, Composite Primary Key를 가지고 있는 relation들만을 검증해보면 된다.

1. 이 relation들은 우선 1NF를 만족한다.

2. ParkingLot\_Mng (Mng\_ID, park\_ID , work\_Date)의 경우, Mng\_ID와 park\_ID 둘 중 하나라도 없으면 workDate가 결정될 수 없다.

3. ParkingSlot (park\_ID, slotNum, floor, availability)의 경우, park\_ID와 slotNum, floor 셋 중 하나라도 없으면 availability가 결정될 수 없다.

4. OwnerPhone (email, phoneNum)의 경우, primary key를 제외한 attribute가 없기 때문에 고려하지 않아도 된다.

5. Pay\_CustomerType (Pay\_ID, customer\_Type)의 경우, primary key를 제외한 attribute가 없기 때문에 고려하지 않아도 된다.

6. 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**

→ 이 relation들은 우선 2NF를 만족하고, primary key를 제외한 attribute간의 이행적 함수 종속 (A가 B를 결정하고, B가 C를 결정하여 A가 C를 결정하는 것)이 존재하는 relation이 존재하지 않으므로 3NF에 적합하다.

**6. Physical DataBase Design: Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| Mng\_ID (PK) | 관리자의 ID | Number(4) | Not null |
| Mng\_Name | 관리자의 이름 | Varcher2(20) | Not null |
| Phone\_Num | 휴대전화 번호 | Varchar2(20) | Not null |
| Salary | 연봉 | Number(10) | Not null |
| Address | 주소 | Varchar2(50) | Not null |

Manager

* 휴대전화 번호는 010-####-####의 형태를 취한다.
* 관리자의 이름은 성과 이름을 포함한다.
* 연봉의 단위는 원화이다.
* 주소는 한국 도로명 주소이다.

ParkingLot

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| Park\_ID (PK) | 주차장의 ID | Number(3) | Not null |
| Name | 주차장의 이름 | Varchar2(20) | Not null |
| Address | 주소 | Varchar2(100) | Not null |
| Charging\_Station | 전기차 충전소 여부 | Varchar2(1) | Not null |
| Area | 주차장 넓이 | Number(5) | Not null |

* 주소는 한국 도로명 주소이다.
* 주차장 넓이의 단위는 ㎡이다.
* 주차장 안에 전기차 충전소가 있다면 Charging\_Station이 ‘Y’, 아니면 ‘N’이다.

ParkingLot\_Mng

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| Mng\_ID (PK) (FK) | 관리자의 ID | Number(4) | Not null |
| Park\_ID (PK) (FK) | 주차장의 ID | Number(3) | Not null |
| Work\_Date | 관리자의 근무날짜 | Date | Not null |

* Mng\_ID는 Manager의 Mng\_ID를 참조한다
* Park\_ID는 ParkingLot의 Park\_ID를 참조한다.
* Work date는 관리자가 근무한 날짜이다.

ParkingSlot

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| Park\_ID (PK) (FK) | 주차장의 ID | Number(3) | Not null |
| Slot\_num (PK) | 주차공간 번호 | Varchar2(5) | Not null |
| Floor (PK) | 층 수 | number (2) | Not null |
| Availability | 공간 사용 가능 여부 | Varchar2(1) | Not null |

* Mng\_ID는 Manager의 Mng\_ID를 참조한다
* Park\_ID는 ParkingLot의 Park\_ID를 참조한다.
* Slot은 자동차 하나의 주차공간을 말한다.
* Availability는 주차공간의 주차 가능 여부로 주차가 가능하면 ‘Y’, 불가능하면 ‘N’이다.

Owner

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| email (PK) | 소유자의 E-mail | Varchar2(50) | Not null |
| FName | 이름 | Varchar2(50) | Not null |
| LName | 성 | Varchar2(20) | Not null |
| City | 도시 | Varchar2(20) | Not null |
| Street | 도로 | Varchar2(20) | Not null |
| ZipCode | 우편번호 | Varchar2(20 | Not null |
| Sex | 성별 | Varchar2(10) | Not null |

* Sex가 여성이면 ‘Female’, 남성이면 ‘Male’이다.

OwnerPhone

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| email (PK) (FK) | 소유자의 E-mail | varchar2(50) | Not null |
| PhoneNum (PK) | 전화번호 | varchar2(50) | Not null |

* Email은 Owner의 email을 참조한다.
* PhoneNum은 010-####-####의 형태를 갖추어야 한다.

Student

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| email (PK) (FK) | 소유자의 E-mail | varchar2(50) | Not null |
| Stu\_ID | 학생번호 | number(8) | Not null |

* Email은 Owner의 email을 참조한다.
* 모든 Stu\_ID는 유일하다. 즉, 같은 Stu\_ID는 존재할 수 없다.

Employee

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| email (PK) (FK) | 소유자의 E-mail | varchar2(50) | Not null |
| emp\_ID | 직원 번호 | varchar2(8) | Not null |
| Emp\_description | 설명 | varchar2(200) |  |

* Email은 Owner의 email을 참조한다.
* 모든 Emp\_ID는 유일하다. 즉, 같은 Emp\_ID는 존재할 수 없다.

Car

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| CarNum (PK) | 차 번호 | varchar2(50) | Not null |
| email (FK) | 소유자의 E-mail | varchar2(50) | Not null |
| CarColor | 차량 색깔 | varchar2(10) |  |
| CarType | 차종 | varchar2(10) |  |
| PlateType | 번호판 종류 | varchar2(10) |  |
| CarModel | 차 모델 | varchar2(20) |  |
| VIN | 차대번호 | varchar2(20) | Not null |
| RegistrationNum | 차량등록번호 | number(20) | Not null |
| licenseNum | 면허번호 | number(20) | Not null |
| licenseType | 면허종류 | varchar2(10) | Not null |

* Email은 Owner의 email을 참조한다.
* 차종은 경차, 준중형차, 중형차, 대형차가 존재하며 특대형차는 입차할 수 없다.

CarEnterInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| CarNum (PK) (FK) | 차 번호 | varchar2(50) | Not null |
| Park\_ID (PK) (FK) | 주차장의 ID | Number (3) | Not null |
| EnterDate | 날짜 | Date | Not null |
| EnterTime | 시간 | varchar2 (10) | Not null |
| EnterType | 입/출차 유형 | varchar2 (5) | Not null |

* CarNum은 Car의 CarNum을 참조한다.
* Park\_ID는 ParkingLot의 Park\_ID를 참조한다.
* EnterType에서 입차는 ‘enter’ 출차는 ‘out’이다.

Payment

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| Pay\_ID (PK) | 정기권 ID | Number (10) | Not null |
| Type | 정기권 유형 | Varchar2 (10) | Not null |
| unit | 결제 단위 | Varchar2 (10) | Not null |
| Price | 정기권 가격 | number (10) | Not null |

* Unit은 일일권 혹은 정기권의 기간이다.
* Price의 단위는 원화이다.

Payment\_Car

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| Pay\_ID (PK) (FK) | 정기권 ID | Number (10) | Not null |
| CarNum (PK) (FK) | 차 번호 | Varchar2 (50) | Not null |
| PayMean | 결재 수단 | Varchar2 (10) | Not null |
| PayTime | 결재 시간 | Varchar2 (10) | Not null |
| PayDate | 결재일 | Date | Not null |
| TotalPrice | 정기권 가격 | number (10) | Not null |

* CarNum은 Car의 CarNum을 참조한다.
* Pay\_ID 은 Payment의 Pay\_ID을 참조한다.
* TotalPrice의 단위는 원화이다.

Payment\_CustomerType

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Description | Data Type | Constraint |
| Pay\_ID (PK) (FK) | 정기권 ID | Number (10) | Not null |
| Customer\_Type (PK) | 구매자 유형 | Varchar2 (10) | Not null |

* Pay\_ID 은 Payment의 Pay\_ID을 참조한다.
* Customer\_Type은 학생일 때 ‘Student’, 교직원일 때 ‘Employee’이다.

**< Detailed Descriptions of Each Table >**

* Manager table은 parkingLot을 관리하는 관리자를 저장하는 table이다.  
  Mng\_ID는 관리자의 ID로, 최대 4자리 숫자로 이루어져 있고, , null값을 허용하지 않으며, Primary Key이다.  
  Mng\_Name은 관리자의 이름을 의미하며, 최대 20자리의 문자열로 이루어져 있으며, null값을 허용하지 않는다.  
  Phone\_Num은 관리자의 휴대폰번호를 의미하며, 최대 20자리의 문자열로 이루어져 있고, null값을 허용하지 않는다.  
  salary는 관리자의 연봉을 의미하며, 최대 10자리의 숫자로 이루어져 있고, null값을 허용하지 않는다.  
  address는 관리자의 집 주소를 의미하며, 최대 50자리의 문자열로 이루어져 있으며, null값을 허용하지 않는다.
* ParkingLot table은 명지대학교의 주차장에 대한 정보를 저장하는 table이다.  
  park\_ID는 최대 3자리의 숫자로 이루어져 있고, null값을 허용하지 않으며, Primary key이다.  
  name은 주차장의 이름으로 최대 20자리의 문자열로 이루어져 있고, null값을 허용하지 않는다.  
  address는 주차장의 주소를 의미하고, 최대 100자리의 문자열로 이루어져 있으며, null값을 허용하지 않는다.  
  charging\_station은 전기차 충전기의 존재 여부를 나타내며 1자리의 문자열로 ‘Y’ 혹은 ‘N’ 값이 들어갈 수 있으며, null값을 허용하지 않는다.  
  Area는 주차장의 면적을 의미하고, 최대 5자리의 숫자로 이루어져 있으며, null값을 허용하지 않는다.
* ParkingSlot은 각 주차장의 주차자리에 대한 정보를 저장하는 table이다.  
  park\_ID는 주차장의 ID로, 최대 3자리의 숫자로 이루어져 있으며, null값을 허용하지 않는다.  
  ParkingSlot은 ParkingLot의 weak entity type이므로, park\_ID는 ParkingLot으로부터 참조한 foreign key이며, 동시에 primary key이다.  
  slotNum은 각 주차자리에 대한 번호로, 최대 5자리의 문자열로 이루어져 있으며, null값을 허용하지 않고, primary key이다.  
  floor은 각 주차자리에 대한 층을 나타내는 정보로, 최대 2자리의 숫자로 이루어져 있다.  
  floor은 null값을 허용하지 않고, primary key이다.
* ParkingLot\_Mng는 어떠한 관리자가 어떠한 주차장을 관리하는 지를 저장하는 table이다.  
  mng\_ID는 관리자의 ID로, 최대 4자리의 숫자로 이루어져 있으며, null값을 허용하지 않는다.  
  mng\_ID는 Manager table로부터 참조한 foreign key이며, 동시에 primary key이다.  
  park\_ID는 주차장의 ID로, 최대 3자리의 숫자로 이루어져 있으며, null값을 허용하지 않는다.  
  park\_ID는 ParkingLot으로부터 참조한 foreign key이며, 동시에 primary key이다.  
  work\_date는 관리자가 근무한 날짜로 date 형식으로 이루어져 있으며, null값을 허용하지 않는다.
* Owner Table은 차에 대한 소유자 즉, 이용자의 정보를 저장하는 Table이다.  
  email은 Primary key로, 최대 50자리의 문자열로 구성되어 있으며, null값을 허용하지 않는다.  
  fName은 이용자의 성을 제외한 이름으로, 최대 50자리의 문자열로 구성되어 있으며, null값을 허용하지 않는다.  
  lName은 이용자의 성으로, 최대 20자리의 문자열로 구성되어 있으며, null값을 허용하지 않는다.  
  city는 이용자의 주소 중 city명을 의미하며, 최대 20자리의 문자열로 구성되어 있고, null값을 허용하지 않는다.  
  street는 이용자의 주소 중 street명을 의미하며, 최대 20자리의 문자열로 구성되어 있고, null값을 허용하지 않는다.  
  zipCode는 이용자의 주소 중 우편번호를 의미하며, 최대 20자리의 문자열로 구성되어 있고, null값을 허용하지 않는다.  
  sex는 이용자의 성별로, 최대 20자리의 문자열로 구성되어 있으며, null값을 허용하지 않는다.
* OwnerPhone table은 Owner table에 multi-valued attribute였던 phoneNum에 대한 정보를 저장하는 table이다.  
  email은 Owner table로부터 참조해온 foreign key로 동시에 primary key이며, 최대 50자리의 문자열로 구성되어 있으며, null값을 허용하지 않는다.  
  phoneNum은 이용자의 휴대폰 번호로, 최대 50자리의 문자열로 구성되어 있으며, null값을 허용하지 않는다.   
  email과 phoneNum\_ID를 결합하여 Composite Primary Key를 형성한다.
* Student table은 Owner의 subtype table이었고, Owner중에 학생인 사람들의 정보를 저장하는 table이다.  
  email은 Owner table로 부터 참조해온 foreign key이며, 동시에 primary key이다.  
  email은 최대 50자리의 문자열로 구성될 수 있으며, null값을 허용하지 않는다.  
  stu\_ID는 학번을 의미하며, 최대 20자리의 문자열로 구성될 수 있으며, null값을 허용하지 않는다.  
  또한 stu\_ID의 모든 값은 서로 달라야 하므로 UNIQUE KEY 제약조건을 설정했다.
* Owner table은 Owner의 subtype table이었고, Owner중에 교직원인 사람들의 정보를 저장하는 table이다.  
  email은 Owner table로 부터 참조해온 foreign key이며, 동시에 primary key이다.  
  email은 최대 50자리의 문자열로 구성될 수 있으며, null값을 허용하지 않는다.  
  emp\_ID는 교번을 의미하며, 최대 20자리의 문자열로 구성될 수 있으며, null값을 허용하지 않는다.  
  description은 교직원의 업무 및 타입에 대한 정보로, 최대 200자리로 구성될 수 있다.  
  또한 emp\_ID의 모든 값은 서로 달라야 하므로 UNIQUE KEY 제약조건을 설정했다.
* Car table은 차량에 관련된 정보를 저장하는 table이다.  
  carNum은 차량 번호를 나타내는 최대 50자리 문자열 필드이고, null값을 허용하지 않는다.  
  carColor, carType, plateType은 차량의 색상, 유형, 번호판 유형을 나타내는 최대 10자리 문자열 필드이고, null값을 허용한다.  
  VIN, , carModel은 차량의 VIN, 자동차 모델을 나타내는 최대 20자리 문자열 필드이고, null값을 허용하지 않는다.  
  registrationNum, licenseNum은 등록번호, 면허번호를 나타내는 최대 20자리 숫자형 필드이고, null값을 허용하지 않는다.  
  licenseType은 면허 유형을 나타내는 최대 10자리 문자열 필드이고, null값을 허용하지 않는다.  
  park\_ID는 차량이 주차된 주차장의 고유한 ID를 나타내는 최대 3자리 숫자형 필드이고, null값을 허용하지 않는다.  
  carNum으로 Primary Key를 형성했다.  
  park\_ID는 ParkingLot의 park\_ID를 참조하는 Foreign Key이다.  
  email은 Employee의 email을 참조하는 Foreign Key이다.
* CarEnterInfo은 Car의 table에 존재하는 attribute 중, 입출차에 관련된 정보만 따로 partitioning한 table을 의미한다.  
  이 시스템은 입,출차에 관련된 정보만 조회하는 경우가 많아, 필요 없는 정보가 조회되어 따로 partitioning이 필요했고,  
  또한 중복된 정보가 입,출차 정보에 따라 반복적으로 적히는 비 효율적인 부분이 존재해 partitioning을 진행했다.  
  이 table은 Car로부터 enterType,enterTime,enterDate,carNum을 가져와 저장한다.  
  carNum은 차량 번호를 나타내는 최대 50자리 문자열 필드이고, null값을 허용하지 않는다.  
  enterDate는 차량의 입차 날짜를 나타내는 날짜형 필드이고, null값을 허용하지 않는다.  
  enterTime은 차량의 입차 시간을 나타내는 최대 10자리 문자열 필드이고, null값을 허용하지 않는다.  
  park\_ID는 차량이 주차된 주차장의 고유한 ID를 나타내는 최대 3자리 숫자형 필드이고, null값을 허용하지 않는다.  
  carNum으로 Primary Key를 형성했다.  
  park\_ID는 ParkingLot의 park\_ID를 참조하는 Foreign Key이다.
* Payment table은 결제와 관련된 정보를 저장하는 table이다.  
  Pay\_ID는 결제의 고유 ID를 나타내는 최대 10자리 숫자형 필드이고, null값을 허용하지 않는다.  
  type은 결제 유형을 나타내는 최대 10자리 문자열 필드이고, null값을 허용하지 않는다.  
  unit은 결제 단위을 나타내는 최대 10자리 문자열 필드이고, null값을 허용하지 않는다.  
  price는 결제 금액을 나타내는 최대 10자리 숫자형 필드이고, null값을 허용하지 않는다.  
  Pay\_ID로 Primary Key를 형성한다.
* Payment\_Car table은 차량 관련 결제 정보를 저장하는 table이다.  
  carNum은 차량 번호를 나타내는 최대 50자리 문자열 필드이고, null값을 허용하지 않는다.  
  Pay\_ID는 결제의 고유 ID를 나타내는 최대 10자리 숫자형 필드이고, null값을 허용하지 않는다.  
  payMean은 결제 수단을 나타내는 최대 10자리 문자열 필드이고, null값을 허용하지 않는다.  
  payTime은 결제 시간을 나타내는 최대 10자리 문자열 필드이고, null값을 허용하지 않는다.  
  payDate는 결제 날짜를 나타내는 날짜형식 필드이고, null값을 허용하지 않는다.  
  totalPrice는 해당 결제의 총 금액을 나타내는 최대 10자리 숫자형 필드이고, null값을 허용하지 않는다.  
  carNum과 Pay\_ID를 결합하여 Composite Primary Key를 형성한다.  
  carNum은 Car table의 carNum을 참조하는 Foreign Key이다.  
  Pay\_ID는 Payment table의 Pay\_ID를 참조하는 Foreign Key이다.
* Payment\_Customtype table은 Payment table의 multi-valued attribute였던 고객 유형과 관련된 결제 정보를 저장하는 table이다.  
  Pay\_ID는 결제의 고유 ID를 나타내는 최대 10자리 숫자형 필드이고, null값을 허용하지 않는다.  
  customer\_type은 고객 유형을 나타내는 최대 10자리 문자열 필드이고, null값을 허용하지 않는다.  
  customer\_type과 Pay\_ID를 결합하여 Composite Primary Key를 형성한다.  
  Pay\_ID는 Payment table의 Pay\_ID를 참조하는 Foreign Key이다.

**7. DDL (Data Definition Language)**

**< DDL >**

CREATE TABLE Manager (  
Mng\_ID number(4) not null,  
Mng\_Name varchar2(20) not null,  
phone\_Num varchar2(20) not null,  
salary number(10) not null,  
address varchar2(50) not null,  
CONSTRAINT mng\_pk PRIMARY KEY (Mng\_ID)  
);

CREATE TABLE ParkingLot (  
park\_ID number(3) not null,  
name varchar2(20) not null,  
address varchar2(100) not null,  
charging\_station varchar2(1) not null,  
area number(5) not null,  
CONSTRAINT ParkingLot\_pk PRIMARY KEY (park\_ID)  
);

CREATE TABLE ParkingSlot (  
park\_ID number(3) not null,  
slotNum varchar2(5) not null,  
floor number(2) not null,  
availability varchar2(1) not null,  
CONSTRAINT ParkingSlot\_pk PRIMARY KEY (park\_ID, slotNum, floor),  
CONSTRAINT ParkingSlot\_fk FOREIGN KEY (park\_ID) references ParkingLot (park\_ID)  
);

CREATE TABLE ParkingLot\_Mng (  
mng\_ID number(4) not null,  
park\_ID number(3) not null,  
work\_Date date not null,  
CONSTRAINT ParkingLot\_Mng\_pk PRIMARY KEY (mng\_ID,park\_ID),  
CONSTRAINT mng\_ID\_fk FOREIGN KEY(mng\_ID) references Manager (mng\_ID),  
CONSTRAINT park\_ID\_fk FOREIGN KEY(park\_ID) references ParkingLot (park\_ID)  
);

CREATE TABLE Owner (  
email varchar2(50) not null,  
fName varchar2(50) not null,  
lName varchar2(20) not null,  
city varchar2(20) not null,  
street varchar2(20) not null,  
zipCode varchar2(20) not null,  
sex varchar2(10) not null,  
CONSTRAINT Owner\_pk PRIMARY KEY (email)  
);

CREATE TABLE OwnerPhone (  
email varchar2(50) not null,  
phoneNum varchar2(50) not null,  
CONSTRAINT OwnerPhone\_pk PRIMARY KEY (email, phoneNum),  
CONSTRAINT OwnerPhone\_fk FOREIGN KEY (email) references Owner (email)  
);

CREATE TABLE Student (  
email varchar2(50) not null,  
stu\_ID number(8) not null,  
CONSTRAINT Student\_pk PRIMARY KEY (email),  
CONSTRAINT Student\_fk FOREIGN KEY (email) references Owner(email),  
CONSTRAINT Student\_uk UNIQUE (stu\_ID)  
);

CREATE TABLE Employee (  
email varchar2(50) not null,  
emp\_ID number(8) not null,  
emp\_description varchar2(200),  
CONSTRAINT Employee\_pk PRIMARY KEY (email),  
CONSTRAINT Employee\_fk FOREIGN KEY (email) references Owner(email),  
CONSTRAINT Employee\_uk UNIQUE (emp\_ID)  
);

CREATE TABLE Car (  
carNum varchar2(50) not null,  
carColor varchar2(10),  
carType varchar2(10),  
plateType varchar2(10),  
carModel varchar2(20),  
VIN varchar2(20) not null,  
registrationNum number(20) not null,  
licenseNum number(20) not null,  
licenseType varchar2(10) not null,  
email varchar2(50) not null,  
CONSTRAINT Car\_pk2 PRIMARY KEY (carNum),  
CONSTRAINT Car\_email\_fk FOREIGN KEY (email) references Owner(email)  
);

CREATE TABLE CarEnterInfo (  
carNum varchar2(50) not null,  
enterDate date not null,  
enterTime varchar2(10) not null,  
enterType varchar2(5),  
park\_ID number(3) not null,  
CONSTRAINT Car\_pk PRIMARY KEY (carNum),  
CONSTRAINT Car\_park\_ID\_fk FOREIGN KEY (park\_ID) references ParkingLot (park\_ID)  
);

CREATE TABLE Payment (  
Pay\_ID number(10) not null,  
type varchar2(10) not null,  
unit varchar2(10) not null,  
price number(10) not null,  
CONSTRAINT Payment\_pk PRIMARY KEY (Pay\_ID)  
);

CREATE TABLE Payment\_Car (  
carNum varchar2(50) not null,  
Pay\_ID number(10) not null,  
payMean varchar2(10) not null,  
payTime varchar2(10) not null,  
payDate date not null,  
totalPrice number(10) not null,  
CONSTRAINT Payment\_Car\_pk PRIMARY KEY (carNum, Pay\_ID),  
CONSTRAINT CarNum\_fk FOREIGN KEY (carNum) references Car (carNum),  
CONSTRAINT Pay\_ID\_fk FOREIGN KEY (Pay\_ID) references Payment (pay\_ID)  
);

CREATE TABLE Payment\_CustomerType (  
Pay\_ID number(10) not null,  
customer\_Type varchar2(10) not null,  
CONSTRAINT Payment\_CustomerType\_pk PRIMARY KEY (customer\_Type, pay\_ID),  
CONSTRAINT Payment\_CustomerType\_fk FOREIGN KEY (Pay\_ID) references Payment (pay\_ID)  
);

**< Data를 채우기 위한 INSERT문 >**

Insert into OWNER values ('seojae12@gmail.com','Jaemin','Seo','Seoul','Moraenae-ro 430','03728','Male');  
Insert into OWNER values ('leese23@gmail.com','Sejin','Lee','Seoul','Tongil-ro 332','03634','Male'); Insert into OWNER values ('sodam11@gmail.com','Sodam','Lee','Seoul','Ganhoda-ro 10','03617','Female');  
Insert into OWNER values ('kimmmin123@gmail.com','Minhyuk','Kim','Incheon','Gyesan-ro 165','21082','Male');  
Insert into OWNER values ('choijh99@gmail.com','Jaehun','Choi','Incheon','Gyeyangsan-ro 17','21042','Male');  
Insert into OWNER values ('hyunaaa42@gmail.com','Hyunah','Park','Incheon','Bongohdae-ro 441','21102','Female');  
Insert into OWNER values ('kimdaeho98@gmail.com','Daeho','Kim','Seoul','Sinchon-ro 1','03785','Male');  
Insert into OWNER values ('parkju1@gmail.com','Juwon','Park','Seoul','Yeonhee-ro 6','03784','Male');  
Insert into OWNER values ('jinha123@gmail.com','Jinha','Shin','Seoul','Sadang-ro 2','07027','Male');  
Insert into OWNER values ('chochoi33@gmail.com','Yujung','Choi','Seoul','Gangnam-ro 7','22785','Female');  
Insert into OWNER values ('heecheol03@gmail.com','Heecheol','Kim','Suwon','Hwasan-ro 8','18361','Male');  
Insert into OWNER values ('parkms02@gmail.com','Minseo','Park','Suwon','Maegok-ro 110','16396','Female');  
Insert into OWNER values ('choijunnn@gmail.com','Jun','Choi','Seoul','Sungbuk-ro 191','02878','Male');  
Insert into OWNER values ('ohhhhyun99@gmail.com','hyungyu','Oh','Seoul','Ohmok-ro 36','08029','Male');  
Insert into OWNER values ('leeeesang@gmail.com','Sangmin','Lee','Seoul','Singil-ro 108','07393','Male');  
Insert into OWNER values ('dodokim12@gmail.com','Doyeon','Kim','Seoul','Bogwang-ro 40','04415','Female');  
Insert into OWNER values ('seeeunlee@gmail.com','Seeun','Lee','Bucheon','Gyungin-ro 396','14692','Female');  
Insert into OWNER values ('yeonnnha123@gmail.com','Haeyon','Moon','Gimpo','Gamjeong-ro 5','10105','Female');  
Insert into OWNER values ('kimsh04@gmail.com','Sohee','Kim','Seoul','Yichon-ro 34','04374','Female');  
Insert into OWNER values ('thesunho0312@gmail.com','Sunho','Kim','Seoul','Baekbum-ro 250','04355','Male');  
Insert into OWNER values ('sonnyho812@gmail.com','Hojun','Son','Anyang','Gyungsudae-ro 511','14120','Male');  
Insert into OWNER values ('limchaemin123@gmail.com','Chaemin','Lim','Bucheon','Sosa-ro 828','14412','Male');  
Insert into OWNER values ('jungsangbin041@gmail.com','Sangbin','Jung','Gimpo','Jungbong-ro 18','10102','Male');  
Insert into OWNER values ('leeeesb@gmail.com','Subin','Lee','Seoul','Jangmun-ro 27','04392','Male');   
Insert into OWNER values ('limmmmsun12@gmail.com','Sunyung','Lim','Yongin','gongse-ro 76','17086','Female');  
Insert into OWNER values ('minsiuu42@gmail.com','Minsu','Kim','Yongin','gugal-ro 45','116971','Male');  
Insert into OWNER values ('minalee23@gmail.com','Minah','Lee','Gimpo','Junggu-ro 93','10098','Female');  
Insert into OWNER values ('choieunhi036@gmail.com','Eunji','Choi','Incheon','Gyungmyungda-ro 1116','21056','Female');  
Insert into OWNER values ('parkdahye123@gmail.com','Dahye','Park','Suwon','Suil-ro 13','16205','Female');  
Insert into OWNER values ('ahhhhyun52@gmail.com','Ahhyun','Hwang','Seoul','Moraenae-ro 430','03728','Female');

insert into OWNERPHONE values ('ahhhhyun52@gmail.com', '010-2115-4332');  
insert into OWNERPHONE values ('limchaemin123@gmail.com', '010-8862-4312');  
insert into OWNERPHONE values ('kimdaeho98@gmail.com', '010-3612-4222');  
insert into OWNERPHONE values ('parkdahye123@gmail.com', '010-4872-5665');  
insert into OWNERPHONE values ('dodokim12@gmail.com', '010-4213-7885');  
insert into OWNERPHONE values ('choieunhi036@gmail.com', '010-4223-5887');  
insert into OWNERPHONE values ('yeonnnha123@gmail.com', '010-3665-9875');  
insert into OWNERPHONE values ('heecheol03@gmail.com', '010-7542-2875');  
insert into OWNERPHONE values ('sonnyho812@gmail.com', '010-1221-3542');  
insert into OWNERPHONE values ('hyunaaa42@gmail.com', '010-2378-3824');  
insert into OWNERPHONE values ('choijh99@gmail.com', '010-3618-9654');  
insert into OWNERPHONE values ('seojae12@gmail.com', '010-9218-1877');  
insert into OWNERPHONE values ('jinha123@gmail.com', '010-9981-7535');  
insert into OWNERPHONE values ('choijunnn@gmail.com', '010-7642-2996');  
insert into OWNERPHONE values ('parkju1@gmail.com', '010-7642-2996');  
insert into OWNERPHONE values ('minalee23@gmail.com', '010-8972-1335');  
insert into OWNERPHONE values ('kimmmin123@gmail.com', '010-4427-7885');  
insert into OWNERPHONE values ('parkms02@gmail.com', '010-3515-9654');  
insert into OWNERPHONE values ('minsiuu42@gmail.com', '010-5210-2335');  
insert into OWNERPHONE values ('jungsangbin041@gmail.com', '010-4334-7995');  
insert into OWNERPHONE values ('leeeesang@gmail.com', '010-5123-1340');  
insert into OWNERPHONE values ('seeeunlee@gmail.com', '010-5322-7299');  
insert into OWNERPHONE values ('leese23@gmail.com', '010-9315-9816');  
insert into OWNERPHONE values ('sodam11@gmail.com', '010-7672-2302');  
insert into OWNERPHONE values ('kimsh04@gmail.com', '010-9681-7671');  
insert into OWNERPHONE values ('leeeesb@gmail.com', '010-6635-4213');  
insert into OWNERPHONE values ('thesunho0312@gmail.com', '010-8735-1175');  
insert into OWNERPHONE values ('limmmmsun12@gmail.com', '010-1167-4225');  
insert into OWNERPHONE values ('chochoi33@gmail.com', '010-1549-3477');  
insert into OWNERPHONE values ('ohhhhyun99@gmail.com', '010-9888-2024');  
insert into OWNERPHONE values ('chochoi33@gmail.com', '010-9436-3477');  
insert into OWNERPHONE values ('limmmmsun12@gmail.com', '010-2725-4225');  
insert into OWNERPHONE values ('minsiuu42@gmail.com', '010-2337-2335');  
insert into OWNERPHONE values ('parkms02@gmail.com', '010-5135-9654');  
insert into OWNERPHONE values ('seeeunlee@gmail.com', '010-3315-7299');

insert into STUDENT values ('ahhhhyun52@gmail.com', 60200123);  
insert into STUDENT values ('limchaemin123@gmail.com', 60221396);  
insert into STUDENT values ('kimdaeho98@gmail.com', 60211286);  
insert into STUDENT values ('parkdahye123@gmail.com', 60221122);  
insert into STUDENT values ('dodokim12@gmail.com', 60181356);  
insert into STUDENT values ('choieunhi036@gmail.com', 60231445);  
insert into STUDENT values ('yeonnnha123@gmail.com', 60231236);  
insert into STUDENT values ('heecheol03@gmail.com', 60210456);  
insert into STUDENT values ('hyunaaa42@gmail.com', 60194205);  
insert into STUDENT values ('sonnyho812@gmail.com', 60172315);  
insert into STUDENT values ('choijh99@gmail.com', 60181220);  
insert into STUDENT values ('seojae12@gmail.com', 60222355);  
insert into STUDENT values ('jinha123@gmail.com', 60211212);  
insert into STUDENT values ('parkju1@gmail.com', 60220142);  
insert into STUDENT values ('choijunnn@gmail.com', 60192331);

insert into EMPLOYEE values ('minalee23@gmail.com', 10180123, 'Professor');  
insert into EMPLOYEE values ('kimmmin123@gmail.com', 20141233, 'Administration Staff');  
insert into EMPLOYEE values ('parkms02@gmail.com', 30180523, 'Counselor');  
insert into EMPLOYEE values ('minsiuu42@gmail.com', 10140235, 'Professor');  
insert into EMPLOYEE values ('jungsangbin041@gmail.com', 10070356, 'Professor');  
insert into EMPLOYEE values ('leeeesang@gmail.com',20181205, 'Administration Staff');  
insert into EMPLOYEE values ('seeeunlee@gmail.com', 20221356, 'Administration Staff');  
insert into EMPLOYEE values ('leese23@gmail.com', 20142322, 'Administration Staff');  
insert into EMPLOYEE values ('sodam11@gmail.com', 10102023, 'Professor');  
insert into EMPLOYEE values ('kimsh04@gmail.com', 10051286, 'Professor');  
insert into EMPLOYEE values ('leeeesb@gmail.com', 10162356, 'Professor');  
insert into EMPLOYEE values ('thesunho0312@gmail.com', 30124056, 'Counselor');  
insert into EMPLOYEE values ('limmmmsun12@gmail.com', 20081705, 'Administration Staff');  
insert into EMPLOYEE values ('chochoi33@gmail.com', 20190233, 'Administration Staff');  
insert into EMPLOYEE values ('ohhhhyun99@gmail.com', 30201078, 'Counselor');

insert into CAR values ('123가4565', 'Black', 'Sedan', 'Standard', 'Hyundai Sonata', 'KNACJ811BCT023581', 101340606198, 133825386832, '1st Class', 'ahhhhyun52@gmail.com');  
insert into CAR values ('212러7352', 'Black', 'SUV', 'Standard', 'Hyundai Tucson', 'DXDTF245LUE220358', 485103235231, 239133466512, '1st Class', 'yeonnnha123@gmail.com');  
insert into CAR values ('112주1264', 'Gray', 'Sedan', 'Corporate', 'Hyundai Avante', 'BZTGP551HSS867396', 585922151000,139612966472, '2nd Class', 'choijh99@gmail.com');  
insert into CAR values ('455마8710', 'White', 'Sedan', 'Electric', 'Tesla Model S', 'KMYXS562UJB463290', 183411241226,118281850731, '1st Class', 'seojae12@gmail.com');  
insert into CAR values ('527오2401', 'White', 'SUV', 'Standard', 'Hyundai Santa Fe', 'TVPYD375NZC357764', 963179110740, 132837260079, '1st Class', 'jungsangbin041@gmail.com');  
insert into CAR values ('312가1130', 'Black', 'Sedan', 'Standard', 'Kia K5', 'TZZWC974WXB871380', 491773695765, 116285375742, '2nd Class', 'seeeunlee@gmail.com');  
insert into CAR values ('422고9882', 'Gray', 'Sedan', 'Standard', 'Toyota Camry', 'ACNEA391EGD031329', 835246007760, 237098418916, '2nd Class', 'sodam11@gmail.com');  
insert into CAR values ('522저5913', 'Blue', 'SUV', 'Standard', 'Kia Seltos', 'JEWND101OIH296544', 895028853893, 112771902507, '1st Class', 'kimsh04@gmail.com');  
insert into CAR values ('629어7412', 'Black', 'Subcompact', 'Standard', 'Kia Morning', 'AJUCL374WVY122623', 942777513040,238104415425, '1st Class', 'leeeesb@gmail.com');  
insert into CAR values ('223부4801', 'Red', 'Sedan', 'Corporate', 'BMW 5 Series', 'GKOZD309INH484367', 276762407574, 238120627575, '1st Class', 'thesunho0312@gmail.com');  
insert into CAR values ('443우6064', 'Blue', 'SUV', 'Electric', 'Hyundai Ioniq 5', 'XHMVD017FDI108788', 320985319549,111218411627, '2nd Class', 'chochoi33@gmail.com');  
insert into CAR values ('121사7462', 'white', 'SUV', 'Standard', 'Hyundai Tucson', 'IBURU600YZB779624', 928929706657, 138741420626,'2nd Class', 'ohhhhyun99@gmail.com');  
insert into CAR values ('327다3071', 'white', 'Subcompact', 'Standard', 'Kia Ray', 'OKZUK661LVY341796', 273418927248, 111989873574,'1st Class', 'kimdaeho98@gmail.com');  
insert into CAR values ('521라2104', 'Black', 'Sedan', 'Corporate', 'Kia K5', 'DGZSN883YEU115567', 959269465141, 235170897136, '1st Class', 'dodokim12@gmail.com');  
insert into CAR values ('338조7482', 'Gray', 'Subcompact', 'Standard', 'Chevrolet Spark', 'ALIPR423GVH422958', 442287533687,137490947577, '1st Class', 'choieunhi036@gmail.com');  
insert into CAR values ('573두8741', 'Blue', 'Sedan', 'Standard', 'Audi A4', 'ZMXGF984LIU212796', 146008359656, 135019029338,'2nd Class', 'leeeesang@gmail.com');

insert into MANAGER values ('1001', 'Kim Juhwan', '01048329888', 160, 'Annam-ro 558');  
insert into MANAGER values ('1002', 'Kim Mingyu', '01032664552', 170, 'Sungbuk-ro 18');  
insert into MANAGER values ('1003', 'Lee Jiyong', '01036844455', 165, 'Samsung-ro 151');  
insert into MANAGER values ('1004', 'Lee Ganghyun', '01021174335', 170, 'Gajwa-ro 303');  
insert into MANAGER values ('1005', 'Hong Wonjin', '01077348855', 190, 'Galhyun-ro 164');  
insert into MANAGER values ('1006', 'Choi Suhyuk', '01045204553', 150, 'Tongil-ro 780');  
insert into MANAGER values ('1007', 'Shin Jaehun', '01044257844', 150, 'Bulgwang-ro 90');  
insert into MANAGER values ('1008', 'Park Jimin', '01099413224', 180, 'Yeonseo-ro 28');  
insert into MANAGER values ('1009', 'Kim Shinwoo', '01088654255', 185, 'Jeongsan-ro 209');  
insert into MANAGER values ('1010', 'Choi Sangmin', '01037234405', 200, 'Majang-ro 199');  
insert into MANAGER values ('1011', 'Jeong Jinsoo', '01022436872', 160, 'Dongil-ro 93');  
insert into MANAGER values ('1012', 'Kim Sangho', '01034584120', 165, 'Hancheon-ro 667');  
insert into MANAGER values ('1013', 'Lee Jihun', '01066232368', 170, 'Maebongsan-ro 80');  
insert into MANAGER values ('1014', 'Kim Junho', '01056654213', 160, 'Mapodae-ro 143');  
insert into MANAGER values ('1015', 'Park Sungmin', '01037072140', 150, 'Jibong-ro 25');

insert into PARKINGLOT values (101, 'MCC\_S10', 'Seoul', 'Y', 1213);  
insert into PARKINGLOT values (102, 'Library\_S9', 'Seoul', 'Y', 713);  
insert into PARKINGLOT values (103, 'Administration\_S5', 'Seoul', 'N', 604);  
insert into PARKINGLOT values (104, 'Dormitory\_S8', 'Seoul', 'Y', 910);  
insert into PARKINGLOT values (105, 'Mirae\_S3', 'Seoul', 'N', 442);  
insert into PARKINGLOT values (106, 'International\_S4', 'Seoul', 'N', 554);  
insert into PARKINGLOT values (107, 'Student\_S', 'Seoul', 'N', 621);  
insert into PARKINGLOT values(108, 'Main\_S1', 'Seoul', 'Y', 562);  
insert into PARKINGLOT values (109, 'Design\_Y12', 'Yongin', 'Y', 548);  
insert into PARKINGLOT values (110, 'Gym\_Y7', 'Yongin', 'N', 626);  
insert into PARKINGLOT values (111, 'Construction\_Y20', 'Yongin', 'N', 746);  
insert into PARKINGLOT values (112, 'Chapel\_Y22', 'Yongin', 'Y', 884);  
insert into PARKINGLOT values (113, 'Experiment\_Y18', 'Yongin', 'N', 967);  
insert into PARKINGLOT values (114, 'Sports\_Y6', 'Yongin', 'Y', 1102);  
insert into PARKINGLOT values (115, 'Student\_Y', 'Yongin', 'N', 431);

insert into PARKINGLOT\_MNG values ('1001', '101', '23/12/01');  
insert into PARKINGLOT\_MNG values ('1002', '103', '23/12/02');  
insert into PARKINGLOT\_MNG values ('1003', '102', '23/12/03');  
insert into PARKINGLOT\_MNG values ('1001', '106', '23/12/04');  
insert into PARKINGLOT\_MNG values ('1004', '103', '23/12/05');  
insert into PARKINGLOT\_MNG values ('1005', '102', '23/12/06');  
insert into PARKINGLOT\_MNG values ('1006', '105', '23/12/07');  
insert into PARKINGLOT\_MNG values ('1002', '101', '23/12/08');  
insert into PARKINGLOT\_MNG values ('1008', '108', '23/12/09');  
insert into PARKINGLOT\_MNG values ('1009', '104', '23/12/10');  
insert into PARKINGLOT\_MNG values ('1003', '101', '23/12/11');  
insert into PARKINGLOT\_MNG values ('1010', '107', '23/12/12');  
insert into PARKINGLOT\_MNG values ('1011', '114', '23/12/13');  
insert into PARKINGLOT\_MNG values ('1004', '104', '23/12/14');  
insert into PARKINGLOT\_MNG values ('1012', '101', '23/12/15');

insert into CARENTERINFO values ('212러7352', '2023/12/11', '12:30', 'out', 101);  
insert into CARENTERINFO values ('112주1264', '2023-12-11', '13:15', 'out', 102);  
insert into CARENTERINFO values ('455마8710', '2023-12-12', '12:00', 'out', 103);  
insert into CARENTERINFO values('527오2401', '2023-12-13', '15:30', 'out', 104);  
insert into CARENTERINFO values('312가1130', '2023-12-13', '16:45', 'out', 105);  
insert into CARENTERINFO values ('422고9882', '2023-12-14', '17:20', 'out', 106);  
insert into CARENTERINFO values ('522저5913', '2023-12-14', '18:10', 'out', 107);  
insert into CARENTERINFO values ('629어7412', '2023-12-15', '07:00', 'out', 108);  
insert into CARENTERINFO values('223부4801', '2023-12-16', '08:15', 'out', 109);  
insert into CARENTERINFO values ('443우6064', '2023-12-16', '09:30', 'out', 110);  
insert into CARENTERINFO values('121사7462', '2023-12-17', '10:45', 'out', 111);  
insert into CARENTERINFO values ('327다3071', '2023-12-17', '11:30', 'out', 112);  
insert into CARENTERINFO values ('521라2104', '2023-12-18', '12:15', 'enter', 113);   
insert into CARENTERINFO values ('338조7482', '2023-12-18', '13:00', 'enter', 114);  
insert into CARENTERINFO values ('573두8741', '2023-12-18', '14:30', 'enter', 115);

insert into PARKINGSLOT values (101, 'A101', 1, 'Y');

insert into PARKINGSLOT values (101, 'B202', 2, 'Y');  
insert into PARKINGSLOT values (101, 'C103', 1, 'N');  
insert into PARKINGSLOT values (101, 'A104', 1, 'Y');  
insert into PARKINGSLOT values (101, 'D205', 2, 'N');  
insert into PARKINGSLOT values (102, 'B106', 1, 'Y');  
insert into PARKINGSLOT values (102, 'A107', 1, 'N');  
insert into PARKINGSLOT values (102, 'B208', 2, 'N');  
insert into PARKINGSLOT values (102, 'A109', 1, 'Y');  
insert into PARKINGSLOT values (102, 'D110', 1, 'N');  
insert into PARKINGSLOT values (103, 'A210', 2, 'Y');  
insert into PARKINGSLOT values (103, 'C110', 1, 'N');  
insert into PARKINGSLOT values (103, 'B110', 1, 'N');  
insert into PARKINGSLOT values (103, 'A205', 2, 'Y');  
insert into PARKINGSLOT values (103, 'A110', 1, 'Y');

insert into PAYMENT values (1, '일일권', '30 min', 2000);  
insert into PAYMENT values (2, '일일권', '1 hour', 3000);  
insert into PAYMENT values (3, '일일권', '1 day', 5000);  
insert into PAYMENT values (4, '정기권', '5 days', 10000);  
insert into PAYMENT values (5, '정기권', '7 days', 15000);  
insert into PAYMENT values (6, '정기권', '14 days', 20000);  
insert into PAYMENT values (7, '정기권', '30 days', 30000);  
insert into PAYMENT values (8, '정기권', '50 days', 40000);  
insert into PAYMENT values (9, '정기권', '60 days', 60000);  
insert into PAYMENT values (10, '정기권', '90 days', 90000);  
insert into PAYMENT values (11, '정기권', '150 days', 150000);  
insert into PAYMENT values (12, '정기권', '180 days', 180000);  
insert into PAYMENT values (13, '정기권', '240 days', 240000);  
insert into PAYMENT values (14, '정기권', '300 days', 300000);  
insert into PAYMENT values (15, '정기권', '365 days', 360000);

insert into PAYMENT\_CAR values ('212러7352', 1, 'Card', '12:50', '2023-12-11', 2000);  
insert into PAYMENT\_CAR values ('112주1264', 2, 'Card', '14:00', '2023-12-11', 3000);  
insert into PAYMENT\_CAR values ('455마8710', 3, 'Card', '16:20', '2023-12-12', 5000);  
insert into PAYMENT\_CAR values ('527오2401', 3, 'Cash', '17:15', '2023-12-13', 5000);  
insert into PAYMENT\_CAR values ('312가1130', 3, 'Cash', '21:00', '2023-12-13', 5000);  
insert into PAYMENT\_CAR values ('422고9882', 7, 'Card', '19:30', '2023-12-14', 30000);  
insert into PAYMENT\_CAR values ('522저5913', 7, 'Cash', '20:45', '2023-12-14', 30000);  
insert into PAYMENT\_CAR values ('629어7412', 1, 'Card', '07:30', '2023-12-15', 2000);  
insert into PAYMENT\_CAR values ('223부4801', 3, 'Cash', '22:30', '2023-12-16', 5000);  
insert into PAYMENT\_CAR values ('443우6064', 2, 'Card', '10:30', '2023-12-16', 3000);  
insert into PAYMENT\_CAR values ('121사7462', 3, 'Card', '18:15', '2023-12-17', 5000);  
insert into PAYMENT\_CAR values ('327다3071', 1, 'Card', '12:00', '2023-12-17', 2000);  
insert into PAYMENT\_CAR values ('521라2104', 15, 'Card', '14:40', '2023-12-18', 360000);  
insert into PAYMENT\_CAR values ('338조7482', 10, 'Card', '15:55', '2023-12-18', 90000);  
insert into PAYMENT\_CAR values ('573두8741', 7, 'Cash', '16:45','2023-12-18', 30000);

insert into PAYMENT\_CUSTOMERTYPE values (1, 'Student');  
insert into PAYMENT\_CUSTOMERTYPE values (1, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (2, 'Student');  
insert into PAYMENT\_CUSTOMERTYPE values (2, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (3, 'Student');  
insert into PAYMENT\_CUSTOMERTYPE values (3, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (4, 'Student');  
insert into PAYMENT\_CUSTOMERTYPE values (5, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (6, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (7, 'Student');  
insert into PAYMENT\_CUSTOMERTYPE values (7, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (8, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (9, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (10, 'Student');  
insert into PAYMENT\_CUSTOMERTYPE values (11, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (12, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (13, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (14, 'Employee');  
insert into PAYMENT\_CUSTOMERTYPE values (15, 'Student');  
insert into PAYMENT\_CUSTOMERTYPE values (15, 'Employee');

**8. DML (Data Manipulation Language)**

아래에 SQL DML문들은 유용한 정보들을 보여줄 수 있는 Query문들이다.

1. 하루동안 얼마나 많은 차가 입차 및 출차 했는지 구하는 Query문  
   SELECT COUNT(carNum) AS CntOfEnter, carNum, enterdate AS enterCarNumber FROM CarEnterInfo WHERE entertype = 'enter' AND TO\_DATE(enterdate, 'YY/MM/DD') = TO\_DATE('찾고 싶은 날짜', 'YY/MM/DD') group by carNum, enterdate;  
   ⇒ 해당 Query문은 주차장의 복잡도를 계산해 앞으로 주차장을 운영하는데 있어서 더욱 효율적인 운영을 위해 설계되었다.
2. 차를 보유한 학생의 수를 계산하는 Query문  
   select count(stu\_ID) As dirverStudentNum from student natural join car;  
   ⇒ 명지대학교 주차관리소가 얼마나 많은 학생들에게 활용되는지 그 수치를 보다 편하고 빠르게 분석할 수 있도록 설계되었다.
3. 전기차 충전소가 있는 주차장의 주차 가능 여부를 구하는 Query문  
   select park\_id, slotnum as available\_eleccharge\_slot ,availability from parkingSlot where park\_id in (select park\_ID from parkingLot where charging\_station = 'Y');  
   ⇒ 전기차를 이용하는 학생 및 교직원들을 위해 전기차 충전소가 있는 주차장의 주차 가능 여부를 보다 빠르고 정확하게 제공할 수 있도록 설계되었다.
4. 최근 한 달 간 주차장 관리자마다의 근무 일 수를 구하는 Query문  
   select MANAGER.MNG\_ID, Count(WORK\_DATE) as CntOfWorkD from MANAGER, PARKINGLOT\_MNG where MANAGER.MNG\_ID = PARKINGLOT\_MNG.MNG\_ID group by MANAGER.MNG\_ID order by MANAGER.MNG\_ID;  
   ⇒ 주차장 관리자들의 근무 횟수를 계산하여 관리자들의 근무 강도의 조절이 필요한지 확인하기 위해 설계되었다.
5. 특정 날짜에 정산된 자동차 중 특정 자동차 색깔에 맞는 자동차를 찾는 Query문  
   select CARNUM from PAYMENT\_CAR where PAYDATE > '찾고자 하는 날짜' and CarNum in (select CARNUM from CAR where CARCOLOR = '찾고자 하는 차 색상');  
   ⇒ 범죄 수사 등에 쓰일 수 있도록 특정한 조건을 가진 차를 쉽게 찾아내기 위해 설계되었다.
6. 구매한 정기권의 개수를 구하는 Query문  
   SELECT Type, Price, (totalPrice / Price) AS Quantity FROM Payment NATURAL JOIN Payment\_Car;  
   ⇒ 이용자 정기권을 얼마나 구매했는지 빠르게 확인할 수 있도록 설계하였다.
7. CarNum이 Student인지 Employee인지 구분하는 Query문  
   (SELECT count(Email) AS Student, null AS Employee FROM Student NATURAL JOIN Car Where CarNum = '찾으려는 차 번호') UNION (SELECT null AS Student, count(Email) AS Employee FROM Employee NATURAL JOIN Car Where CarNum = '찾으려는 차 번호');  
   ⇒ 학생이 교직원으로 등록되어 있거나, 교직원이 학생으로 등록되어 있는 등의 전산 오류가 발생했을 때, 데이터를 빠르게 조회하고 대처할 수 있도록 설계하였다.
8. 남은 주차 공간을 구하는 Query문  
   SELECT Park\_ID, floor, count(Slotnum) as NumOfAvail FROM ParkingSlot where Availability = 'Y' GROUP BY Park\_ID, floor;  
   ⇒ 수용할 수 있는 차의 수를 한 번에 보고 관리자가 대처할 수 있도록 설계하였다.
9. 들어온 차량의 경차의 수를 Query문  
   SELECT COUNT(\*) AS num\_entered\_compact\_car FROM carEnterInfo CEI JOIN Car ON CEI.carNum = Car.carNum WHERE Car.carType = 'Compact' AND CEI.enterType = 'enter';  
   ⇒ 향후 경차 주차자리를 더 확보해 놓을 지 결정을 보다 쉽게 내리기 위해 설계되었다
10. 어떤 주차장에 전기차가 가장 많이 입출차하는 지 확인하는 Query문  
    SELECT PL.park\_ID, PL.name AS parking\_lot\_name, COUNT(\*) AS num\_entered\_electric\_car FROM ParkingLot PL JOIN CarEnterInfo CEI ON PL.park\_ID = CEI.park\_ID JOIN Car ON CEI.carNum = Car.carNum WHERE Car.carType = 'Electric' AND CEI.enterType = 'enter' GROUP BY PL.park\_ID, PL.name ORDER BY num\_entered\_electric\_car;  
    ⇒ 전기차가 많이 입출차하는 주차장에 향후 전기차 충전기를 얼마나 구매할 지 결정을 내리기 위해 설계하였다.

**9. System Manual**

본 레포트에 첨부된 DDL문들을 순서대로 복사하여 테스트해야 하는 것과, Query문을 실행시킬 때 조건에 들어가는 문자열 자료형 데이터의 대소문자에 주의해야 하는 것 이외의 특별한 안내 사항은 없다.