**프로젝트 마일스톤**

Final Report

# 2조

60190424 김영서

60191434 배정연

60211650 김선민

명지대학교

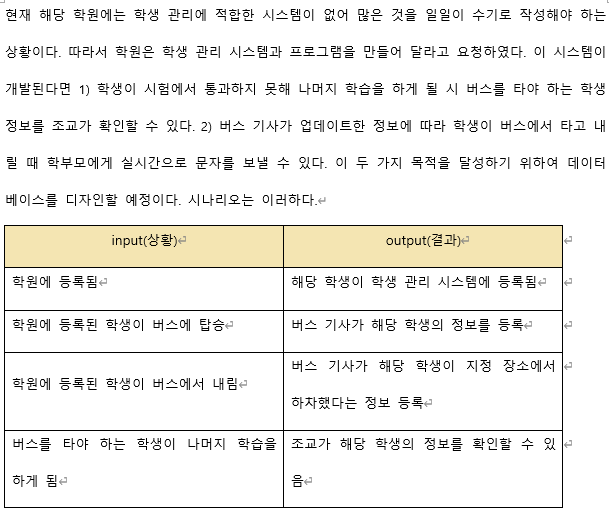
## <목차>

1. [Group members and their roles 3](#bookmark)
2. [Project organization and its purpose 3](#bookmark)
3. [User Requirements & Business Rules 3](#bookmark)
   1. [User Requirements 3](#bookmark1)
   2. [Business Rules 3](#bookmark2)
4. [Conceptual Modeling 5](#bookmark3)
   1. [Conceptual Modeling 5](#bookmark4)
5. [Logical Modeling 6](#bookmark3)
   1. Relational Modeling 6
6. Implementation in Oracle DBMS

1. Group members and their roles.

|  |  |
| --- | --- |
| 팀원명 | 담당 역할 |
| 김영서 | Project document 작성, Group meeting 참가, 최종 Document 수정 및 검토 |
| 배정연 | Project document 작성, Group meeting 참가, 최종 Document 수정 및 검토 |
| 김선민 | Project document 작성, Group meeting 참가, 최종 Document 수정 및 검토 |

2. Project organization and its purpose



3.[User Requirements & Business Rules](#bookmark5)

## 3.1.User Requirements

학원 버스 등하원 할 때 정확한 학생 관리 시스템이 부재하여 관리 프로그램을 요청하였다. 관리 프로그램은 다음과 같은 조건을 충족해야 한다.

1. 버스를 타려면 학원에 등록된 학생이어야 한다.

2. 학생이 버스를 탈 경우 정해진 시간에만 탈 수 있다.

3. 학원 등원을 할 때 버스 기사는 이동하기 전 탑승한 학생의 정보를 등록해야 한다

4. 학원에서 퇴원할 때 버스 기사는 학생이 지정 장소에서 하차한다는 정보를 등록해야 한다.

5. 버스기사가 업데이트하는 정보를 토대로 학원 행정 직원은 학생이 버스에 타거나 내릴 때 학부모에게 문자를 보낼 수 있는 시스템이어야 한다.

6. 수업을 마친 학생 중 나머지 학습을 해야 하는 학생 중에서 버스를 타야 하는 학생의 정보를 조교가 확인 가능한 시스템이여야 한다.

## 3.2.Business Rules

## 학생(Student)은 StudentID(이걸로 서로 다른 학생을 구분), 이름, 나이, 학교 이름, 성별, 전화번호를 가진다. 학부모(Parent)는 ParentID(이걸로 서로 다른 학부모를 구분), 이름, 전화번호, 주소, 메일 주소, 성별을 가진다. 그리고 모든 학생들은 부모가 있고 학원에 기록된 부모들은 모두 관련한 학생이 있다. 형제가 학원에 다닐 경우에도 부모 정보는 하나만 적혀 있다. 많은 학생들은 모두 하나의 영어 학원에 등록한다. 영어 학원 등록 날짜는 등록할 때 기록된다. 학원 데이터베이스에 들어있는 모든 학생들은 영어 학원에 등록해 있고 영어 학원은 모든 학생들의 정보가 있다. 영어 학원(English Academy)은 AcademyID(영어학원을 구성하는 데 꼭 필요함)와 이름, 주소, 메일 주소와 전화번호를 가진다. 영어 학원은 여러 명의 직원을 고용한다(고용할 때 고용날짜를 기록함). 모든 직원들은 영어 학원에 고용되어 있고 영어 학원은 모든 직원들의 정보가 있다. 직원(Employee)들은 EmployeeID, 이름 성별, 나이, 이메일 주소, 전화번호를 가진다. 직원으로는 시간제 직원(Hourly Employee)과 월급제 직원(Salaried Employee)이 구분되어 나뉘는데 시간제 직원은 시급을 갖고 월급제 직원은 월급과 주소를 가진다. 월급제 직원으로는 선생님(Teacher)과 버스기사(Driver)와 행정직원이 있다. 여러 선생님은 여러 과목을 가르치고, 가르치는 선생님이 없는 과목은 없다. 그리고 선생님이 과목을 가르치는 동안에 조교는 과목에서 시행하는 시험을 채점하여 통과하지 못한 학생들을 따로 기록하기 때문에 Failed Student I가 발생한다. 과목(Subject)은 SubjectID(이걸로 서로 다른 과목을 구분)와 이름과 시험 점수 커트라인과 선생님 이름을 가진다. 그리고 과목과 반은 서로 할당되는데 한 과목 당 여러 개의 반이 배정되고, 한 반 당 여러 개의 과목이 배정된다. 과목은 반마다 꼭 할당되어 있지 않아도 괜찮지만, 반은 과목을 무조건 배정받아야 한다. 반(Class)은 반 이름과 ClassID(이걸로 서로 다른 클래스를 구분)와 시작 시간과 끝나는 시간과 레벨로 구성된다. 그리고 하나의 반에는 한 명의 선생님만 할당되고 모든 반은 선생님을 할당 받는다. 반에 할당되지 않은 선생님이 있을 수도 있다. 한 명의 학생은 하나의 반만 가질 수 있으며 모든 학생은 반을 배정받지만 학생을 가지지 않는 반이 있을 수도 있다. 하나의 반은 한 개의 방만 사용하며 방(Room)은 아이디(이걸로 서로 다른 방을 구분), 티브이의 상태, 에어컨의 상태, 칠판의 상태, 수용 가능 인원을 속성으로 가진다. 반은 반드시 방을 사용한다. 사용되지 않는 방이 있을 수 있다.

## 버스기사 한 명당 하나의 버스를 운전한다. 버스(Bud)는 BusNumber(이걸로 서로 다른 버스를 구분)과 여러 개의 등원 시간과 여러 개의 하원 시간이 있고 좌석과 여러 개의 정류소가 있다. 버스를 타는 학생들은 타는 시간과 내리는 시간을 각각 가진다. 버스는 타야 하는 학생들을 모두 태우지만 모든 학생들이 버스에 타는 것은 아니다.

## 4. Conceptual Modeling

## 이미지 4.1. Conceptual Modeling

**5. Logical Modeling**

**5.1. Relational Modeling**

English Academy(AcademyID, Name, Address, E-mail Address, Phone Number)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 AcademyID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Parent(ParentID, Name, Phone Number, E-mail Address, Sex, Address)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 ParentID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Bus(BusNumber, Seater, DriverID)

Foreign key(DriverID) References Driver(DriverID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 BusNumber에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Bus\_AttendTime(BusNumber, AttendTime)

Foreign key(BusNumber) references Bus(BusNumber)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 BusNumber, AttendTime에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Bus\_LeaveTime(BusNumber, LeaveTime)

Foreign key(BusNumber) references Bus(BusNumber)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 BusNumber, LeaveTime에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Bus\_Spot(BusNumber, Spot)

Foreign key(BusNumber) references Bus(BusNumber)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 BusNumber, Spot에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Student(StudentID, Name, School Name, Phone Number, Sex, AcademyID, ParentID, ClassID, BusNumber, Enroll\_Date, Ride\_Time, GetOff\_Time,Age)

Foreign key(AcademyID) references English Academy(AcademyID)

Foreign key(ClassID) references Class(ClassID)

Foreign key(ParentID) references Parent(ParentID)

Foreign key(BusNumber) references Bus(BusNumber)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 StudentID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Employee(EmployeeID, Name, E-mail Address, Sex, PhoneNumber, Age, AcademyID,Hire\_Date)

Foreign key(AcademyID) references English Academy(AcademyID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 EmployeeID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Hourly Employee(EmployeeID, Hourly\_rate)

Foreign key(EmployeeID) references Employee(EmployeeID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 EmployeeID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Salaried Employee(EmployeeID, Salary, Address)

Foreign key(EmployeeID) references Employee(EmployeeID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 EmployeeID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Teacher(TeacherID)

Foreign key(TeacherID) references Salaried Employee(EmployeeID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 TeacherID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Driver(DriverID)

Foreign key(DriverID) references Salaried Employee(EmployeeID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 DriverID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Subject(SubjectID, Name, Exam Cutline)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 SubjectID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

SubjectTeacherList(SubjectID, Teacher Name)

Foreign key(SubjectID) references Subject (SubjectID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 SubjectID, Teacher Name에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Failed\_Student\_List(TeacherID, SubjectID, FailedStudentID)

Foreign key(SubjectID) references Subject(SubjectID)

Foreign key(TeacherID) references Teacher(TeacherID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 TeacherID, SubjectID, FailedStudentID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Room(RoomID, Status of TV, Status of Air conditioner, Status of Blackboard, Capacity)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 RoomID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

Class(ClassID, Level, Name, Start\_Time, End\_Time, RoomID,TeacherID)

Foreign key(TeacherID) references Teacher(TeacherID)

Foreign key(RoomID) references Room(RoomID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 ClassID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

AssignedClass(ClassID, SubjectID)

Foreign key(ClassID) references Class(ClassID)

Foreign key(SubjectID) references Subject(SubjectID)

1. 모든 attribute는 multivalued과 composite이 아님. ->1NF 달성

2. 1NF이면서 모든 non-primary key attribute가 ClassID, SubjectID에 fully functional dependent 함 -> 2NF 달성

3. 2NF이면서 모든 non-primary key attribute가 transitive functional dependency가 없음. -> 3NF 달성

**6. Implementation in Oracle DBMS**

**- 아카데미 테이블 생성**

create table Academy (

AcademyID varchar2(10) NOT NULL,

Academy\_Name varchar2(10),

Address varchar2(10),

EmailAddress varchar2(20),

PhoneNumber varchar2(10),

constraint Academy\_pk primary key (AcademyID)

);

**- 부모 테이블 생성, primary key(ParentID)로 제약 조건 걸어둠**

create table AParent (

ParentID varchar2(10) NOT NULL,

Parent\_Name varchar2(10),

Address varchar2(10),

EmailAddress varchar2(20),

PhoneNumber varchar2(20),

Sex varchar2(1),

constraint Parent\_pk primary key(ParentID)

);

**-직원 테이블 생성, primary key(EmployeeID), foreign key (AcademyID)**

**references Academy(AcademyID)로 제약 조건 걸어둠**

create table Employee (

EmployeeID varchar2(10) NOT NULL,

Employee\_Name varchar2(10),

EmailAddress varchar2(20),

PhoneNumber varchar2(20),

Sex varchar2(1),

Age varchar2(2),

AcademyID varchar2(10) NOT NULL,

Hire\_Date date,

constraint Employee\_pk primary key(EmployeeID),

constraint Employee\_fk foreign key (AcademyID) references Academy(AcademyID)

);

**-선생님 테이블 생성, primary key(TeacherID),**

**foreign key (TeacherID) references Employee(EmployeeID)로 제약 조건 걸어둠**

create table Teacher (

TeacherID varchar2(10) NOT NULL,

constraint Teacher\_pk primary key(TeacherID),

constraint Teacher\_fk foreign key (TeacherID) references Employee(EmployeeID)

);

**-운전 기사 테이블 생성, primary key(DriverID), foreign key (DriverID)**

**references Employee(EmployeeID)로 제약 조건 걸어둠**

create table Driver (

DriverID varchar2(10) NOT NULL,

constraint Driver\_pk primary key(DriverID),

constraint Driver\_fk foreign key (DriverID) references Employee(EmployeeID)

);

**-버스테이블 생성, 버스 번호는 null이 될수 없고 primary key(BusNumber), foreign key (DriverID) references Driver(DriverID)로 제약 조건 걸어둠**

create table Bus (

BusNumber varchar2(10) NOT NULL,

Seater varchar2(10),

DriverID varchar2(10),

constraint Bus\_pk primary key(BusNumber),

constraint Bus\_fk foreign key (DriverID) references Driver(DriverID)

);

**-버스 등원 시간 테이블 생성, 버스 번호, 등원 시간은 null이 될 수 없고**

**primary key(BusNumber, AttendTime)고 foreign key (BusNumber) references Bus(BusNumber)로 제약 조건 걸어둠.**

create table Bus\_AttendTime (

BusNumber varchar2(10) NOT NULL,

AttendTime varchar2(10) NOT NULL,

constraint Bus\_AttendTime\_pk primary key(BusNumber, AttendTime),

constraint Bus\_AttendTime\_fk foreign key (BusNumber) references Bus(BusNumber)

);

**-버스 하원 시간 테이블 생성, 버스 번호, 하원 시간은 null이 될 수 없고**

**primary key(BusNumber, LeaveTime)고 foreign key (BusNumber) references Bus(BusNumber)로 제약 조건 걸어둠.**

create table Bus\_LeaveTime (

BusNumber varchar2(10) NOT NULL,

LeaveTime varchar2(10) NOT NULL,

constraint Bus\_LeaveTime\_pk primary key(BusNumber, LeaveTime),

constraint Bus\_LeaveTime\_fk foreign key (BusNumber) references Bus(BusNumber)

);

* **버스 장소 테이블 생성, , 버스 번호, 정류소는 null이 될 수 없고**

**Primary key(BusNumber, Spot)이고 foreign key (BusNumber) references Bus(BusNumber)로 제약 조건 걸어둠**

create table Bus\_Spot (

BusNumber varchar2(10) NOT NULL,

Spot varchar2(10) NOT NULL,

constraint Bus\_Spot\_pk primary key(BusNumber, Spot),

constraint Bus\_Spot\_fk foreign key (BusNumber) references Bus(BusNumber)

);

**-교실 테이블 생성, 방번호는 null이 될 수 없고, primary key(RoomID)**

**임.**

create table Room (

RoomID varchar2(10) NOT NULL,

Status\_Of\_TV varchar2(10),

Status\_Of\_AirConditioner varchar2(10),

Status\_Of\_Blackboard varchar2(10),

Capacity varchar2(10),

constraint Room\_pk primary key(RoomID)

);

**-반 테이블 생성, 반 아이디는 null값이 될 수 없고, primary key(ClassID)이고**

**foreign key (RoomID) references Room(RoomID), foreign key (TeacherID) references Teacher(TeacherID)로 제약 조건 걸어둠**

create table AClass (

ClassID varchar2(10) NOT NULL,

Class\_Level varchar2(10),

Start\_Time varchar2(10),

End\_Time varchar2(10),

RoomID varchar2(10),

TeacherID varchar2(10),

constraint Class\_pk primary key(ClassID),

constraint Class\_fk1 foreign key (RoomID) references Room(RoomID),

constraint Class\_fk2 foreign key (TeacherID) references Teacher(TeacherID)

);

**-학생 테이블 생성, 학생 ID와 학생 이름,AcademyID, ParentID, ClassID는 null이 될 수 없고, primary key(StudentID)이고 foreign key (AcademyID) references Academy(AcademyID)이고 foreign key (ClassID) references AClass(ClassID)이고**

**foreign key (ParentID) references AParent(ParentID)이고 foreign key (BusNumber) references Bus(BusNumber)라고 제약 조건 걸어둠**

create table Student (

StudentID varchar2(10) NOT NULL,

Student\_Name varchar2(10) NOT NULL,

SchoolName varchar2(10),

PhoneNumber varchar2(20),

Sex varchar2(1), AcademyID varchar2(10) NOT NULL,

ParentID varchar2(10) NOT NULL,

ClassID varchar2(10) NOT NULL,

BusNumber varchar2(10),

Enroll\_Date date,

Ride\_Time varchar2(10),

GetOff\_Time varchar2(10),

Age varchar2(2), constraint Student\_pk primary key(StudentID),

constraint Student\_fk1 foreign key (AcademyID) references Academy(AcademyID),

constraint Student\_fk2 foreign key (ClassID) references AClass(ClassID), constraint Student\_fk3 foreign key (ParentID) references AParent(ParentID), constraint Student\_fk4 foreign key (BusNumber) references Bus(BusNumber) ););

**-시간제 직원 테이블 생성, 직원 ID는 NULL이 아니고, primary key(EmployeeID)이고 foreign key (EmployeeID) references Employee(EmployeeID) 라고 제약 조건 걸어둠**

create table Hourly\_Employee (

EmployeeID varchar2(10) NOT NULL,

Hourly\_Rate varchar2(10),

constraint Hourly\_Employee\_pk primary key(EmployeeID),

constraint Hourly\_Employee\_fk foreign key (EmployeeID) references

Employee(EmployeeID));

**-봉급제 직원 테이블 생성, 직원 ID는 NULL이 아니고, primary key(EmployeeID)이고 foreign key (EmployeeID) references Employee(EmployeeID) 라고 제약 조건 걸어둠**

create table Salaried\_Employee (

EmployeeID varchar2(10) NOT NULL,

Salary varchar2(10),

Address varchar2(10),

constraint Salaried\_Employee\_pk primary key(EmployeeID),

constraint Salaried\_Employee\_fk foreign key (EmployeeID) references Employee (EmployeeID)

);

* **과목 테이블 생성, SubjectID는 null이 아니고 primary key(SubjectID)이라고 제약 조건 걸어둠.**

create table Subject (

SubjectID varchar2(10) NOT NULL,

Subject\_Name varchar2(10),

Exam\_Cutline varchar2(10),

constraint Subject\_pk primary key(SubjectID)

);

* **과목담당 선생님 테이블 생성, SubjectID는 null이 아니고 primary key(SubjectID, Teacher\_Name), foreign key (SubjectID) references Subject(SubjectID)) 라고 제약 조건 걸어둠.**

create table SubjectTeacher (

SubjectID varchar2(10) NOT NULL,

Teacher\_Name varchar2(10) NOT NULL,

constraint SubjectTeacher\_pk primary key(SubjectID, Teacher\_Name),

constraint SubjectTeacher\_fk foreign key (SubjectID) references Subject(SubjectID)

);

* **낙제 학생 테이블 생성, TeacherID, SubjectID는 null이 아니고 primary key(SubjectID, TeacherID, FailedStudentID) foreign key (SubjectID) references Subject(SubjectID)) (TeacherID) references Teacher(TeacherID) 라고 제약 조건 걸어둠.**

create table Failed\_Student (

TeacherID varchar2(10) NOT NULL,

SubjectID varchar2(10) NOT NULL,

FailedStudentID varchar2(10) NOT NULL,

constraint Failed\_Student\_pk primary key(SubjectID, TeacherID, FailedStudentID),

constraint Failed\_Student\_fk1 foreign key (SubjectID) references Subject(SubjectID),

constraint Failed\_Student\_fk2 foreign key (TeacherID) references Teacher(TeacherID)

);

* **담당 클래스 생성, ClassID와 SubjectID는 null이 아니고 primary key(ClassID, SubjectID)이고 foreign key (ClassID) references AClass(ClassID)이고 foreign key (SubjectID) references Subject(SubjectID)라고 제약 조건 걸어둠**

create table AssignedClass (

ClassID varchar2(10) NOT NULL,

SubjectID varchar2(10) NOT NULL,

constraint AssignedClass\_pk primary key(ClassID, SubjectID),

constraint AssignedClass\_fk1 foreign key (ClassID) references AClass(ClassID),

constraint AssignedClass\_fk2 foreign key (SubjectID) references Subject(SubjectID)

);

insert into academy values (1, 'abc1', 'abc', 'abc1@abc.com', '123-1111');

insert into academy values (2, 'abc2', 'abc', 'abc2@abc.com', '123-2222');

insert into academy values (3, 'abc3', 'abc', 'abc3@abc.com', '123-3333');

insert into academy values (4, 'abc4', 'abc', 'abc4@abc.com', '123-4444');

insert into academy values (5, 'abc5', 'abc', 'abc5@abc.com', '123-5555');

insert into academy values (6, 'abc6', 'abc', 'abc6@abc.com', '123-6666');

insert into academy values (7, 'abc7', 'abc', 'abc7@abc.com', '123-7777');

insert into academy values (8, 'abc8', 'abc', 'abc8@abc.com', '123-8888');

insert into academy values (9, 'abc9', 'abc', 'abc9@abc.com', '123-9999');

insert into academy values (10, 'abc10', 'abc', 'abc10@abc.com', '123-1010');

insert into academy values (11, 'abc11', 'abc', 'abc11@abc.com', '123-0011');

insert into academy values (12, 'abc12', 'abc', 'abc12@abc.com', '123-1212');

insert into academy values (13, 'abc13', 'abc', 'abc13@abc.com', '123-1313');

insert into academy values (14, 'abc14', 'abc', 'abc14@abc.com', '123-1414');

insert into academy values (15, 'abc15', 'abc', 'abc15@abc.com', '123-1515');

insert into aparent values (1, 'sam', 'abc', 'sam@abc.com', '010-1212-0000', 'M');

insert into aparent values (2, 'sara', 'abc', 'sara@abc.com', '010-1111-0000', 'F');

insert into aparent values (3, 'candy', 'abc', 'candy@abc.com', '010-1212-3333', 'F');

insert into aparent values (4, 'jun', 'def', 'jun@abc.net', '010-2323-0000', 'M');

insert into aparent values (5, 'jhon', 'abc', 'jhon@abc.com', '011-2029-0000', 'M');

insert into aparent values (6, 'happy', 'def', 'happy@abc.com', '010-1212-0000', 'F');

insert into aparent values (7, 'new', 'def', 'happy@abc.com', '010-1212-0000', 'F');

insert into aparent values (8, 'coke', 'def', 'coke@abc.com', '010-1212-0000', 'F');

insert into aparent values (9, 'code', 'abc', 'code@abc.com', '010-1212-0000', 'F');

insert into aparent values (10, 'hope', 'abc', 'hope@abc.com', '010-1212-0000', 'F');

insert into aparent values (11, 'love', 'def', 'love@abc.com', '010-1212-0000', 'F');

insert into aparent values (12, 'like', 'abc', 'like@abc.com', '010-1212-0000', 'F');

insert into aparent values (13, 'seem', 'abc', 'seem@abc.com', '010-1212-0000', 'F');

insert into aparent values (14, 'daily', 'def', 'daily@abc.com', '010-1212-0000', 'F');

insert into aparent values (15, 'neo', 'def', 'neo@abc.com', '010-1212-0000', 'F');

insert into employee values (101, 'chris', 'chris@abc.com', '010-2020-2020', 'M', 28, 1, '2021-12-01');

insert into employee values (102, '홍길동', 'hong@abc.com', '010-1234-2020', 'M', 21, 1, '2020-10-01');

insert into employee values (103, '김철수', 'kcs@abc.com', '010-7262-2020', 'M', 22, 1, '2021-12-01');

insert into employee values (104, '김영희', 'kyh@abc.com', '010-2222-2020', 'F', 23, 1, '2021-12-01');

insert into employee values (105, 'mary', 'mary@abc.com', '010-2020-0283', 'F', 30, 1, '2021-12-01');

insert into employee values (106, 'tom', 'tom@abc.com', '010-8634-2020', 'M', 28, 1, '2020-10-01');

insert into employee values (107, '김영서', 'kys@abc.com', '010-9173-2020', 'F', 34, 1, '2019-10-01');

insert into employee values (108, '김선민', 'ksm@abc.com', '010-1037-2020', 'F', 34, 1, '2019-10-01');

insert into employee values (109, '배정연', 'bjy@abc.com', '010-3744-2020', 'F', 34, 1, '2019-10-01');

insert into employee values (110, '가나다', 'GND@abc.com', '010-2939-2020', 'F', 40, 1, '2019-12-01');

insert into employee values (111, '라마바', 'rmb@abc.com', '010-1212-2320', 'M', 35, 1, '2019-12-01');

insert into employee values (112, 'jun', 'jun@abc.com', '010-1212-2320', 'M', 25, 1, '2022-12-01');

insert into employee values (113, '김영수', 'kys2@abc.com', '010-1212-2320', 'M', 27, 1, '2020-12-01');

insert into employee values (114, '김영우', 'kyw@abc.com', '010-1212-2320', 'M', 30, 1, '2021-12-01');

insert into employee values (115, '유재석', 'yjs@abc.com', '010-1212-2320', 'M', 25, 1, '2021-12-01');

insert into employee values (116, '강래현', 'rmb4342@abc.com', '010-1212-2320', 'M', 35, 1, '2019-12-01');

insert into employee values (117, '장채은', 'rmb43434@abc.com', '010-1212-2320', 'M', 35, 1, '2019-12-01');

insert into employee values (118, '장나라', 'rmb434344@abc.com', '010-1212-2320', 'M', 35, 1, '2019-12-01');

insert into employee values (119, '강동원', 'rmb43364@abc.com', '010-1212-2320', 'M', 35, 1, '2019-12-01');

insert into employee values (120, '원빈', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 5, '2019-12-01');

insert into employee values (121, '소지섭', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 3, '2019-12-01');

insert into employee values (122, '김가람', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 11, '2019-12-01');

insert into employee values (123, '임성민', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 12, '2019-12-01');

insert into employee values (124, '장그래', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 15, '2019-12-01');

insert into employee values (125, '강하늘', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 10, '2019-12-01');

insert into employee values (126, '송중기', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 15, '2019-12-01');

insert into employee values (127, '김길현', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 14, '2019-12-01');

insert into employee values (128, '구하라', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 13, '2019-12-01');

insert into employee values (129, '라미란', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 12, '2019-12-01');

insert into employee values (130, '강지연', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 11, '2019-12-01');

insert into employee values (131, '고한별', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 10, '2019-12-01');

insert into employee values (132, '이한빈', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 9, '2019-12-01');

insert into employee values (133, '유채은', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 8, '2019-12-01');

insert into employee values (134, '고동현', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 7, '2019-12-01');

insert into employee values (135, '강정우', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 6, '2019-12-01');

insert into employee values (136, '고아라', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 6, '2019-12-01');

insert into employee values (137, '김태희', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 4, '2019-12-01');

insert into employee values (138, '장원빈', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 5, '2019-12-01');

insert into employee values (139, '고원빈', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 5, '2019-12-01');

insert into employee values (140, '창원빈', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 3, '2019-12-01');

insert into employee values (141, '곡원빈', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 3, '2019-12-01');

insert into employee values (142, '장아라미', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 10, '2019-12-01');

insert into employee values (143, '한아름송이', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 10, '2019-12-01');

insert into employee values (144, '홍영기', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 2, '2019-12-01');

insert into employee values (145, '홍준표', 'rmb434@abc.com', '010-1212-2320', 'M', 35, 2, '2019-12-01');

insert into salaried\_employee values (101, 2500, 'abc');

insert into salaried\_employee values (102, 2500, 'def');

insert into salaried\_employee values (103, 2500, 'abc');

insert into salaried\_employee values (104, 4000, 'abc');

insert into salaried\_employee values (105, 2500, 'def');

insert into salaried\_employee values (106, 2500, 'abc');

insert into salaried\_employee values (107, 4000, 'abc');

insert into salaried\_employee values (108, 4000, 'zzz');

insert into salaried\_employee values (109, 4000, 'zzz');

insert into salaried\_employee values (110, 2000, 'zzz');

insert into salaried\_employee values (111, 2000, 'abc');

insert into salaried\_employee values (112, 2000, 'abc');

insert into salaried\_employee values (113, 2000, 'abc');

insert into salaried\_employee values (114, 2000, 'def');

insert into salaried\_employee values (115, 2000, 'def');

insert into salaried\_employee values (116, 2000, 'abc');

insert into salaried\_employee values (117, 2000, 'abc');

insert into salaried\_employee values (118, 2000, 'abc');

insert into salaried\_employee values (119, 2000, 'abc');

insert into salaried\_employee values (120, 2000, 'abc');

insert into salaried\_employee values (121, 2000, 'abc');

insert into salaried\_employee values (122, 2000, 'abc');

insert into salaried\_employee values (123, 2000, 'abc');

insert into salaried\_employee values (124, 2000, 'abc');

insert into salaried\_employee values (125, 2000, 'abc');

insert into salaried\_employee values (126, 2000, 'abc');

insert into salaried\_employee values (127, 2000, 'abc');

insert into salaried\_employee values (128, 2000, 'abc');

insert into salaried\_employee values (129, 2000, 'abc');

insert into salaried\_employee values (130, 2000, 'abc');

insert into hourly\_employee values (131, 12);

insert into hourly\_employee values (132, 10);

insert into hourly\_employee values (133, 10);

insert into hourly\_employee values (134, 10);

insert into hourly\_employee values (135, 10);

insert into hourly\_employee values (136, 10);

insert into hourly\_employee values (137, 10);

insert into hourly\_employee values (138, 10);

insert into hourly\_employee values (139, 10);

insert into hourly\_employee values (140, 10);

insert into hourly\_employee values (141, 10);

insert into hourly\_employee values (142, 10);

insert into hourly\_employee values (143, 12);

insert into hourly\_employee values (144, 10);

insert into hourly\_employee values (145, 10);

insert into teacher values (101);

insert into teacher values (102);

insert into teacher values (103);

insert into teacher values (104);

insert into teacher values (105);

insert into teacher values (106);

insert into teacher values (107);

insert into teacher values (108);

insert into teacher values (109);

insert into teacher values (110);

insert into teacher values (111);

insert into teacher values (112);

insert into teacher values (113);

insert into teacher values (114);

insert into teacher values (115);

insert into driver values (116);

insert into driver values (117);

insert into driver values (118);

insert into driver values (119);

insert into driver values (120);

insert into driver values (121);

insert into driver values (122);

insert into driver values (123);

insert into driver values (124);

insert into driver values (125);

insert into driver values (126);

insert into driver values (127);

insert into driver values (128);

insert into driver values (129);

insert into driver values (130);

insert into bus values (1,45,116);

insert into bus values (2,30,117);

insert into bus values (3,20,118);

insert into bus values (4,20,119);

insert into bus values (5,15,120);

insert into bus values (6,15,121);

insert into bus values (7,20,122);

insert into bus values (8,20,123);

insert into bus values (9,20,124);

insert into bus values (10,20,125);

insert into bus values (11,20,126);

insert into bus values (12,22,127);

insert into bus values (13,23,128);

insert into bus values (14,24,129);

insert into bus values (15,25,130);

insert into bus\_attendtime values (2, '14:40');

insert into bus\_attendtime values (1, '15:40');

insert into bus\_attendtime values (3, '14:40');

insert into bus\_attendtime values (4, '15:40');

insert into bus\_attendtime values (5, '14:40');

insert into bus\_attendtime values (6, '15:40');

insert into bus\_attendtime values (7, '14:40');

insert into bus\_attendtime values (8, '15:40');

insert into bus\_attendtime values (2, '16:40');

insert into bus\_attendtime values (1, '17:50');

insert into bus\_attendtime values (2, '19:00');

insert into bus\_attendtime values (3, '16:40');

insert into bus\_attendtime values (4, '17:50');

insert into bus\_attendtime values (3, '19:00');

insert into bus\_attendtime values (5, '16:40');

insert into bus\_attendtime values (6, '17:50');

insert into bus\_attendtime values (5, '19:00');

insert into bus\_leavetime values (2, '17:20');

insert into bus\_leavetime values (1, '18:30');

insert into bus\_leavetime values (2, '19:50');

insert into bus\_leavetime values (2, '22:10');

insert into bus\_leavetime values (3, '17:20');

insert into bus\_leavetime values (4, '18:30');

insert into bus\_leavetime values (3, '19:50');

insert into bus\_leavetime values (3, '22:10');

insert into bus\_leavetime values (5, '17:20');

insert into bus\_leavetime values (6, '18:30');

insert into bus\_leavetime values (5, '19:50');

insert into bus\_leavetime values (5, '22:10');

insert into bus\_leavetime values (7, '17:20');

insert into bus\_leavetime values (8, '18:30');

insert into bus\_leavetime values (7, '19:50');

insert into bus\_leavetime values (7, '22:10');

insert into bus\_spot values (2, 'abc');

insert into bus\_spot values (2, 'def');

insert into bus\_spot values (1, 'abc');

insert into bus\_spot values (1, 'def');

insert into bus\_spot values (3, 'abc');

insert into bus\_spot values (3, 'def');

insert into bus\_spot values (4, 'abc');

insert into bus\_spot values (4, 'def');

insert into bus\_spot values (5, 'abc');

insert into bus\_spot values (5, 'def');

insert into bus\_spot values (6, 'abc');

insert into bus\_spot values (6, 'def');

insert into room values (1, 'broken', 'nomal', 'nomal', 25);

insert into room values (2, 'nomal', 'nomal', 'none', 20);

insert into room values (3, 'nomal', 'nomal', 'none', 20);

insert into room values (4, 'nomal', 'nomal', 'nomal', 30);

insert into room values (5, 'nomal', 'nomal', 'none', 20);

insert into room values (6, 'nomal', 'nomal', 'none', 20);

insert into room values (7, 'nomal', 'nomal', 'nomal', 25);

insert into room values (8, 'nomal', 'nomal', 'none', 20);

insert into room values (9, 'nomal', 'nomal', 'none', 20);

insert into room values (10, 'broken', 'nomal', 'none', 20);

insert into room values (11, 'nomal', 'nomal', 'none', 20);

insert into room values (12, 'nomal', 'nomal', 'none', 20);

insert into room values (13, 'nomal', 'broken', 'normal', 30);

insert into room values (14, 'nomal', 'nomal', 'none', 20);

insert into room values (15, 'nomal', 'nomal', 'none', 20);

insert into aclass values (1011, 'E-L1', '15:20', '17:20', 1, 107);

insert into aclass values (1012, 'E-L2', '15:20', '17:20', 2, 107);

insert into aclass values (1021, 'E-M1', '15:20', '17:20', 3, 107);

insert into aclass values (1022, 'E-M2', '15:20', '17:20', 5, 107);

insert into aclass values (1031, 'E-H1', '16:20', '18:20', 4, 107);

insert into aclass values (1032, 'E-H2', '16:20', '18:20', 4, 107);

insert into aclass values (2011, 'M-L1', '17:20', '19:20', 1, 108);

insert into aclass values (2021, 'M-M1', '17:20', '19:20', 2, 108);

insert into aclass values (2022, 'M-M2', '17:20', '19:20', 2, 108);

insert into aclass values (2031, 'M-H1', '18:20', '20:20', 3, 108);

insert into aclass values (3011, 'H-L1', '19:40', '21:40', 1, 109);

insert into aclass values (3012, 'H-L2', '19:40', '21:40', 1, 109);

insert into aclass values (3021, 'H-M1', '19:40', '21:40', 7, 109);

insert into aclass values (3022, 'H-M2', '19:40', '21:40', 7, 109);

insert into aclass values (3031, 'H-H1', '19:40', '21:40', 4, 109);

insert into student values (1001, 'andy', 'abc 초등', '010-2020-4932', 'F', 1, 1, 1012, null, '2021-10-01', null, null, 12);

insert into student values (1002, 'bob', 'abc 중등', '010-1212-4932', 'M', 1, 1, 1011, null, '2021-10-01', null, null, 13);

insert into student values (1003, 'coll', 'abc 초등', '010-3434-4932', 'M', 1, 2, 1021, 2, '2021-10-01', '14:50', '17:43', 13);

insert into student values (1004, 'dive', 'abc 초등', '010-4444-4932', 'F', 1, 3, 1021, 2, '2021-10-01', '14:30', '17:50', 10);

insert into student values (1005, 'elv', 'abc 중등', '010-3565-4932', 'M', 1, 4, 1032, 1, '2021-10-01', '15:50', null, 15);

insert into student values (1006, 'five', 'abc 중등', '010-3320-4932', 'F', 1, 5, 2011, 1, '2021-10-01', '15:48', '19:59', 14);

insert into student values (1007, 'girl', 'abc 초등', '010-3333-4932', 'M', 1, 5, 1022, 1, '2021-10-01', '15:48', '19:59', 9);

insert into student values (1008, 'high', 'abc 고등', '010-5656-4932', 'M', 1, 6, 3011, 2, '2021-10-01', '19:20', '22:33', 18);

insert into student values (1009, 'info', 'def 고등', '010-3423-4932', 'F', 1, 7, 3012, 2, '2021-10-01', null, '22:33', 18);

insert into student values (1010, 'jack', 'def 고등', '010-4564-4932', 'F', 1, 8, 3012, 2, '2021-10-01', '19:30', '22:23', 17);

insert into student values (1011, 'koko', 'abc 고등', '010-2435-4932', 'M', 1, 9, 3031, 2, '2021-10-01', '19:20', '22:33', 18);

insert into student values (1012, 'lim', 'def 고등', '010-7644-4932', 'F', 1, 10, 3031, 2, '2021-10-01', null, '22:33', 18);

insert into student values (1013, 'nave', 'def 고등', '010-4354-4932', 'F', 1, 11, 3021, 2, '2021-10-01', '19:30', '22:23', 17);

insert into student values (1014, 'pop', 'def 고등', '010-2794-4932', 'F', 1, 12, 3022, 2, '2021-10-01', '19:30', '22:23', 17);

insert into student values (1015, 'puppy', 'def 고등', '010-2485-4932', 'M', 1, 13, 3021, 2, '2021-10-01', '19:30', '22:23', 17);

insert into student values (1016, 'rock', 'def 고등', '010-1243-4932', 'F', 1, 14, 3021, 2, '2021-10-01', '19:30', '22:23', 17);

insert into student values (1017, 'coco', 'def 고등', '010-7578-4932', 'F', 1, 15, 3021, 2, '2021-10-01', '19:30', '22:23', 17);

insert into subject values (1, '듣기EL', 45);

insert into subject values (2, '듣기EM', 45);

insert into subject values (3, '듣기EH', 45);

insert into subject values (4, '듣기ML', 45);

insert into subject values (5, '듣기MM', 45);

insert into subject values (6, '듣기MH', 45);

insert into subject values (7, '듣기HL', 45);

insert into subject values (8, '듣기HM', 45);

insert into subject values (9, '듣기HH', 45);

insert into subject values (10, '쓰기EL', 50);

insert into subject values (11, '쓰기EM', 50);

insert into subject values (12, '쓰기EH', 50);

insert into subject values (13, '쓰기ML', 50);

insert into subject values (14, '쓰기MM', 50);

insert into subject values (15, '쓰기MH', 50);

insert into subject values (16, '쓰기HL', 50);

insert into subject values (17, '쓰기HM', 50);

insert into subject values (18, '쓰기HH', 50);

insert into subject values (19, '문법EL', 45);

insert into subject values (20, '문법EM', 45);

insert into subject values (21, '문법EH', 45);

insert into subject values (22, '문법ML', 45);

insert into subject values (23, '문법MM', 45);

insert into subject values (24, '문법MH', 45);

insert into subject values (25, '문법HL', 45);

insert into subject values (26, '문법HM', 45);

insert into subject values (27, '문법HH', 45);

insert into subjectTeacher values (1, 'chris');

insert into subjectTeacher values (2, 'chris');

insert into subjectTeacher values (3, 'chris');

insert into subjectTeacher values (4, 'mary');

insert into subjectTeacher values (5, 'mary');

insert into subjectTeacher values (6, 'mary');

insert into subjectTeacher values (7, 'tom');

insert into subjectTeacher values (8, 'tom');

insert into subjectTeacher values (9, 'tom');

insert into subjectTeacher values (10, '김영서');

insert into subjectTeacher values (11, '김영서');

insert into subjectTeacher values (12, '김영서');

insert into subjectTeacher values (19, '김영서');

insert into subjectTeacher values (20, '김영서');

insert into subjectTeacher values (21, '김영서');

insert into subjectTeacher values (13, '김선민');

insert into subjectTeacher values (14, '김선민');

insert into subjectTeacher values (15, '김선민');

insert into subjectTeacher values (22, '김선민');

insert into subjectTeacher values (23, '김선민');

insert into subjectTeacher values (24, '김선민');

insert into subjectTeacher values (16, '배정연');

insert into subjectTeacher values (17, '배정연');

insert into subjectTeacher values (18, '배정연');

insert into subjectTeacher values (25, '배정연');

insert into subjectTeacher values (26, '배정연');

insert into subjectTeacher values (27, '배정연');

insert into subjectTeacher values (16, '홍길동');

insert into subjectTeacher values (17, '김철수');

insert into subjectTeacher values (18, '김영희');

insert into failed\_student values (107, 1, 1001);

insert into failed\_student values (107, 10, 1001);

insert into failed\_student values (107, 2, 1003);

insert into failed\_student values (107, 20, 1004);

insert into failed\_student values (108, 21, 1005);

insert into failed\_student values (108, 13, 1006);

insert into failed\_student values (109, 25, 1009);

insert into failed\_student values (109, 8, 1013);

insert into failed\_student values (109, 26, 1013);

insert into failed\_student values (109, 8, 1015);

insert into failed\_student values (109, 26, 1015);

insert into failed\_student values (109, 8, 1016);

insert into failed\_student values (109, 26, 1016);

insert into failed\_student values (109, 8, 1017);

insert into failed\_student values (109, 26, 1017);

insert into assignedclass values (1011, 10);

insert into assignedclass values (1011, 19);

insert into assignedclass values (1012, 1);

insert into assignedclass values (1012, 10);

insert into assignedclass values (1021, 2);

insert into assignedclass values (1021, 20);

insert into assignedclass values (1022, 2);

insert into assignedclass values (1022, 11);

insert into assignedclass values (1031, 3);

insert into assignedclass values (1031, 21);

insert into assignedclass values (1032, 3);

insert into assignedclass values (1032, 21);

insert into assignedclass values (2011, 5);

insert into assignedclass values (2011, 22);

insert into assignedclass values (2021, 5);

insert into assignedclass values (2021, 23);

insert into assignedclass values (2022, 5);

insert into assignedclass values (2022, 14);

insert into assignedclass values (2031, 6);

insert into assignedclass values (2031, 24);

insert into assignedclass values (3011, 7);

insert into assignedclass values (3011, 25);

insert into assignedclass values (3012, 16);

insert into assignedclass values (3012, 25);

insert into assignedclass values (3021, 8);

insert into assignedclass values (3021, 26);

insert into assignedclass values (3022, 26);

insert into assignedclass values (3022, 8);

insert into assignedclass values (3031, 18);

insert into assignedclass values (3031, 27);

<유효 SQL문>

1.낙제한 학생의 부모에게 전화하기 위한 명단

select distinct student.studentid, student\_name, parent\_name, aparent.phoneNumber from student, failed\_student, aparent

where student.studentId = failed\_student.failedstudentid and student.parentid = aparent.parentid

order by student.studentid;

2.낙제한 학생 중 버스를 타는 학생에게 전달해줄 버스 시간표

select student\_name, bus\_leavetime.busnumber, leavetime from student, failed\_student, bus\_leavetime

where student.studentId = failed\_student.failedstudentid

and student.busnumber = bus\_leavetime.busnumber

and student.busnumber is not null

order by studentId;

3.각 담임 선생 별 관리하는 학생 명단

select classId, student\_name, class\_level, roomId, student.age, student.sex, schoolname from student natural join aclass

where teacherId = 107

order by classId;

select classId, student\_name, class\_level, roomId, student.age, student.sex, schoolname from student natural join aclass

where teacherId = 108

order by classId;

select classId, student\_name, class\_level, roomId, student.age, student.sex, schoolname from student natural join aclass

where teacherId = 109

order by classId;

4.담임 선생이 담당하는 학생 총 수

select teacherId, employee\_name as Teacher\_Name, count(studentId)

as total\_student from employee, student

natural join aclass where employee.employeeid = aclass.teacherid

group by teacherID, employee\_name

order by teacherId;

5. 버스기사가 관리하는 학생 명단

select driver.driverid, student.busnumber, student\_name, student.age, student.sex, schoolname

from driver, student, bus

where driver.driverid = bus.driverid and

student.busnumber = bus.busnumber

order by driver.driverid;

6. 버스기사가 관리하는 학생 총 수

select driver.driverid, bus.busnumber, count(studentid) as num\_Of\_students\_in\_charge

from driver, student, bus

where driver.driverid = bus.driverid and

student.busnumber = bus.busnumber

group by driver.driverid, bus.busnumber

order by driver.driverid;

7. 낙제생이 시험봐야 할 재시험 과목 명단

create view reexamination\_list as

select failedStudentId, subject\_name as subject\_of\_reexamination from subject left outer join Failed\_Student on subject.subjectId = failed\_student.subjectId

order by failedStudentId;

select failedStudentId, student\_name, subject\_of\_reexamination from reexamination\_list, student

where reexamination\_list.failedStudentId = student.studentId;

8. 조교가 출력해야 하는 시험지 개수 명단

select subject\_of\_reexamination, count(failedStudentId) as print from reexamination\_list

group by subject\_of\_reexamination

having count(failedStudentId)>0;

9.학생별 오늘 듣는 수업과 현재 수업듣고 있는 방을 확인하기 위한 명단

select distinct StudentID, Student\_Name, AssignedClass.ClassID, Subject\_Name, AClass.RoomID

from Student, AClass, Room, AssignedClass, Subject

where Student.ClassID = AClass.ClassID and

AClass.RoomID = Room.RoomID and

AssignedClass.ClassID = AClass.ClassID and

AssignedClass.SubjectID = Subject.SubjectID

order by StudentID;

10.학생별 시험에 통과했는지 못했는지 여부 확인

create view pass\_or\_not\_list as

select distinct studentId, student\_name, failedStudentId as Pass\_or\_not, student.classid

from student left outer join Failed\_Student on student.studentid=failed\_student.failedstudentid

order by studentId;

1) 전체 학생 조회

select studentId, student\_name, Pass\_or\_not from pass\_or\_not\_list;

2) 담임 선생별 조회

select studentId, student\_name, pass\_or\_not from pass\_or\_not\_list natural join aclass

where aclass.teacherid = 107;

select studentId, student\_name, pass\_or\_not from pass\_or\_not\_list natural join aclass

where aclass.teacherid = 108;

select studentId, student\_name, pass\_or\_not from pass\_or\_not\_list natural join aclass

where aclass.teacherid = 109;

11. 방에 고장난 티비가 있는 지 확인

select roomID as Broken\_TV\_RoomID

from room where status\_of\_tv = 'broken';

12. 방에 고장난 에어컨이 있는 지 확인

select roomID as Broken\_Aircon\_RoomID

from room where status\_of\_airconditioner = 'broken';