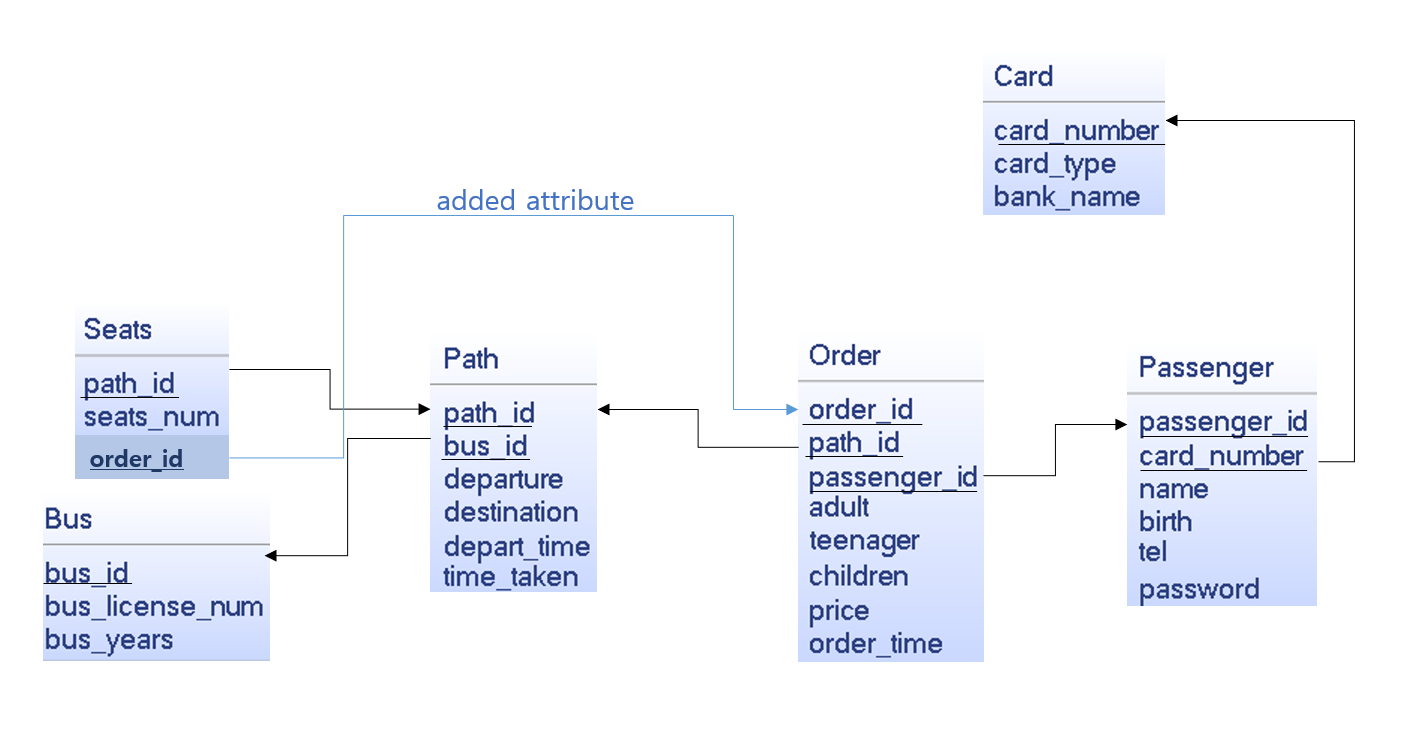
텍스트, 영수증, 스크린샷이(가) 표시된 사진

자동 생성된 설명

0 You MUST use the tables used in HW2. If you need to change your tables, you should submit the revised schema diagram of your tables.



1. Select four queries from the SQL queries in HW2. Any two queries can’t belong to the same category(I - VI). All the queries should include one or more host variables.

1) show the four queries from HW2 you selected, including the host variables.

①

set @bus\_used\_years = 5;

select \* from Bus where (date\_format(now(), '%Y') - bus\_years) > @bus\_used\_years;

②

set @P\_id = 2130;

select seats\_num from Seats where path\_id = @P\_id;

③

set @bus\_years = 2017;

select \* from path where bus\_id in (select bus\_id from bus where bus\_years >= @bus\_years);

④

set @bank\_a = "하나";

set @bank\_b = "국민";

(select \* from card where bank\_name = @bank\_a)

union

(select \* from card where bank\_name = @bank\_b);

2) write a C (or Python etc) program which can process the four SQL queries. You should include as many comments as possible to make it easier to be read.

import mysql.connector

# connecting to mysql server and database

cnx = mysql.connector.connect(user='root', password='dgu1234!', host='localhost', database='intercity\_bus\_terminal')

# getting cursor

cursor = cnx.cursor()

year = 5

# set host variable bus\_used\_years = 5

cursor.execute("set @bus\_used\_years = {}".format(year))

# Retrieve the information about buses that was made less that 5 years.

cursor.execute("select \* from Bus where (date\_format(now(), '%Y') - bus\_years) > @bus\_used\_years")

for c in cursor:

    print(c)

print()

p\_id = 2130

# set host variable P\_id = 2130

cursor.execute("set @P\_id = {}".format(p\_id))

# Retrieve the information about path\_id, departure, destination, and reserved seats\_num of the path whose path\_id is 2130.

cursor.execute("select seats\_num from Seats where path\_id = @P\_id")

seats = []

for c in cursor:

    # Because c is tuple, format is (seat\_num, ), through indexing, can get element "seat\_number". And then put in the seats list.

    a = c[0]

    seats.append(a)

for s in seats:

    print(s, end=" ")

print("\n")

bus\_years = 2017

# set host variable year = 2017

cursor.execute("set @year = {}".format(bus\_years))

# Retrieve the information of the path whose bus is made after 2017.

cursor.execute("select \* from path where bus\_id in (select bus\_id from bus where bus\_years >= @year)")

for c in cursor:

  print(c)

print()

bank\_a = "하나"

bank\_b = "국민"

# set host variable bank\_a = "하나" and bank\_b = "국민"

cursor.execute("SET @bank\_a = \"{}\"".format(bank\_a))

cursor.execute("SET @bank\_b = \"{}\"".format(bank\_b))

# Retrieve the information about cards whose bank\_name is “하나” or “국민”.

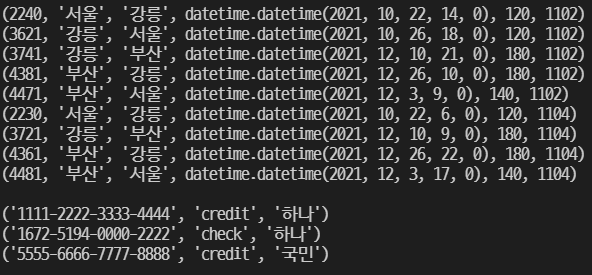
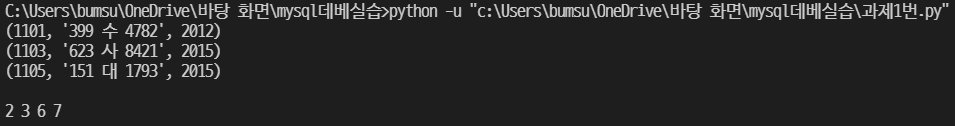
cursor.execute("(SELECT \* FROM CARD WHERE BANK\_NAME = @bank\_a) UNION (SELECT \* FROM CARD WHERE BANK\_NAME = @bank\_b)")

for c in cursor:

  print(c)

print()

3) show that your program works.



2. Implement the following.

1) define a procedure which contains an aggregate function.

delimiter //

create procedure getting\_avg\_time(

departure\_place varchar(20)

)

begin

select avg(time\_taken) from Path where departure = departure\_place;

end //

delimiter ;

2) define a function which returns a value.

delimiter //

create function getting\_avg\_time\_func(

departure\_place varchar(20)

) returns decimal(7, 2)

begin

declare result decimal(7, 2);

select avg(time\_taken) into result from Path where departure = departure\_place;

return result;

end //

delimiter ;

3) write a C (or Python etc) program which calls/uses both of the procedures/functions defined in 1)-2).

import mysql.connector

cnx = mysql.connector.connect(user='root', password='dgu1234!', host='localhost')

cursor = cnx.cursor()

cursor.execute("USE intercity\_bus\_terminal")

cursor.execute("SELECT GETTING\_AVG\_TIME\_FUNC('서울')")

for c in cursor:

  print(c)

print()

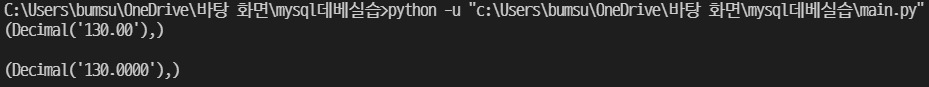
cursor.execute("CALL GETTING\_AVG\_TIME('서울')");

for c in cursor:

  print(c)

print()

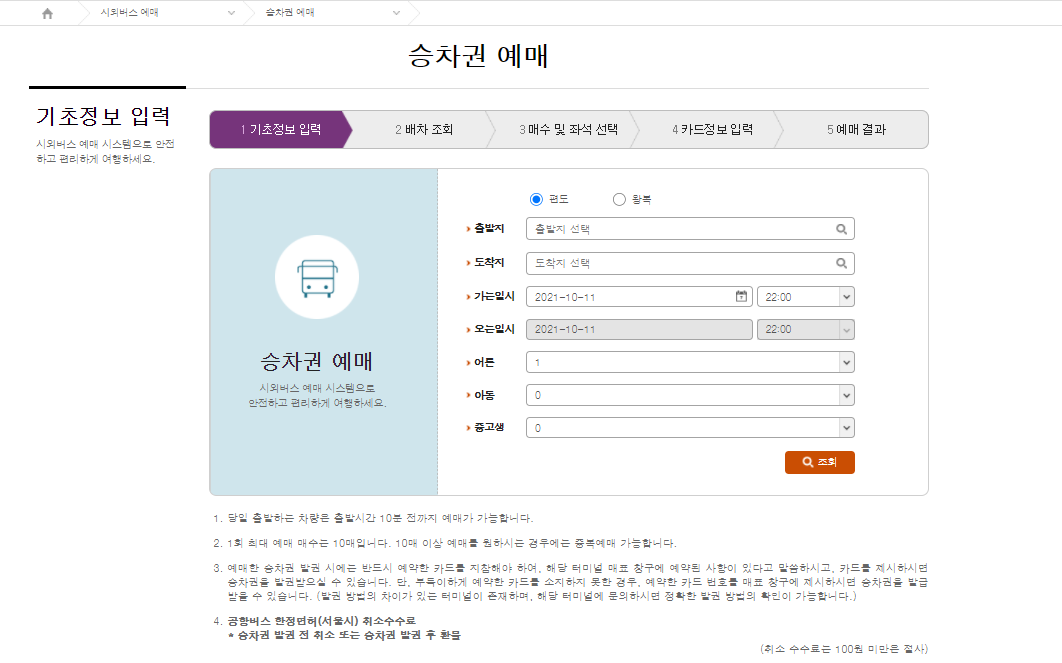
4) show that your program works.



3. The boundary of the Question 3 is flexible, and you can do as much work as you like to do. The grade will depend on the amount and quality of your work. Among the tasks you intended to do in HW 1, implement some of (if not all of, or more than) them.

For each task you will implement, provide the following information in the report.

1) describe what you are going to do with the use of web pages. Show the current web pages or you can draw ones yourself.



This page is the first page. A passenger choose departure and destination. And then he chooses when to depart. The important thing is that in our project, we only provide one-way ticket, not round-trip. And then select the number of adults, teenagers, and children.



The second page is about choosing the bus route. In this page, we show available routes satisfying the preconditions. The price is determined by the following rules.

● The price(Adult’s) is calculated by time-taken. Exactly,

(time-taken) \* 200 becomes the price. For example, when the time-taken is 120 minutes, the adult’s price is 120 \* 200 = 24000 won.

● Teenagers get 10% discount, and children get 20% discount of the adult’s price. In the above example, the teenager’s price is 24000 \* 0.9 = 21600 won, and the child’s price is 24000 \* 0.8 = 19200 won.

The passenger chooses a route in the shown routes.



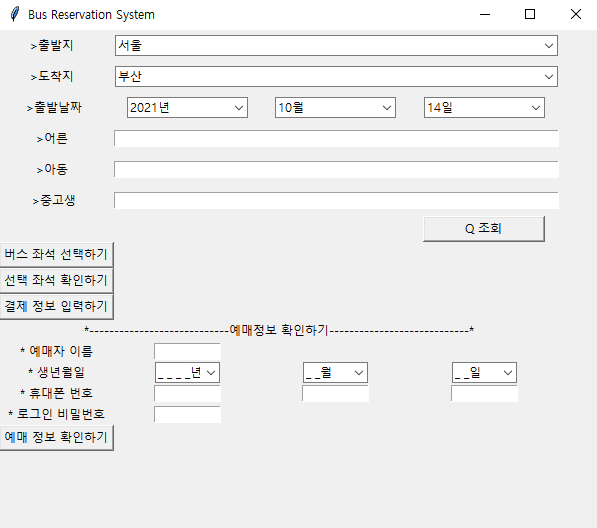
The third page is about choosing the seat numbers. The seat number that is already reserved can’t be chosen.

텍스트이(가) 표시된 사진

자동 생성된 설명

The final page is about payments and filling the information for viewing this reservation. After this process, the reservation is over, and can see the booking anytime.

2) implement a program using host language plus Mysql.

Since the program code is too long, I will attach only the execution screen. Detailed code can be found in the submitted code file.

3) show that your program is working correctly. In case you don’t use GUI(graphic user interface), make sure you provide a proof that your program is working correctly. For example, suppose the task is about showing a summary report for a given input. Then you should clearly demonstrate that your program correctly displays all necessary information when a certain input is given. There will be extra points for GUI development, though.

텍스트이(가) 표시된 사진

자동 생성된 설명

In this area, the user enters the origin and destination departure dates and number of customers by age and checks if there is a route corresponding to the input information. If there are matching paths in the database path table, the paths are displayed in the listbox. If there is no corresponding path than print nothing. Usually Intercity bus terminal has various path day by day, but in this program we have only 12 paths. If you want to check if the path finding function is working properly, please refer to the example below.

“서울 -> 부산 : 21/10/14”

“서울 -> 강릉 : 21/10/22”

“강릉 -> 서울 : 21/10/26”

“강릉 -> 부산 : 21/12/10”

“부산 -> 강릉 : 21/12/26”

“부산 -> 서울 : 21/12/03”

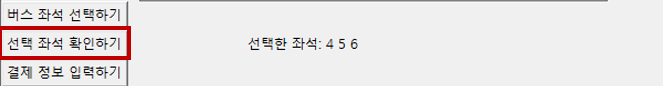
텍스트이(가) 표시된 사진

자동 생성된 설명

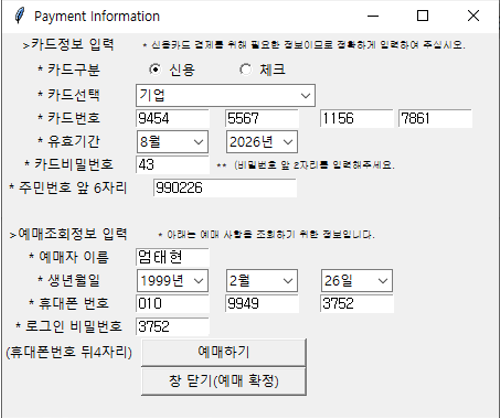
If you have selected an appropriate time from among the searched paths, you can select a seat on the bus you want through 'Select Bus Seat'. If there is a seat already selected, the seat number is displayed as 'x' and the button cannot be selected.

테이블이(가) 표시된 사진

자동 생성된 설명

After selecting the seat you want, you can check the seat you have selected one more time through "선택 좌석 확인하기".

Next, you can enter payment information through "Enter payment information Button".



If you have entered all the payment information, click the “Reservation” button. When you click the Reservation button, the “Close Window button” appears and you can confirm the payment by clicking the Close Window button. Since it is cumbersome to re-enter payment information, check the message box once more to make sure that you have entered the payment information properly. 텍스트이(가) 표시된 사진

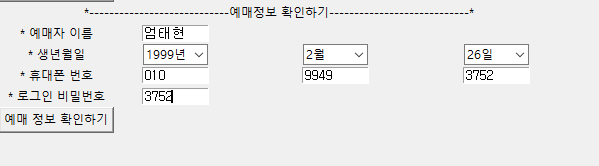
자동 생성된 설명

Even if you select yes in the close window message box, if all necessary information is not entered, the payment information is rewritten.

텍스트이(가) 표시된 사진

자동 생성된 설명

By entering payment information, all information of the reservation process is stored in the database. Also, payment information can be checked at any time by clicking Check Payment Information at the bottom of the GUI. If you enter the requested payment information carefully, you can check your reservation information at any time.



텍스트이(가) 표시된 사진

자동 생성된 설명