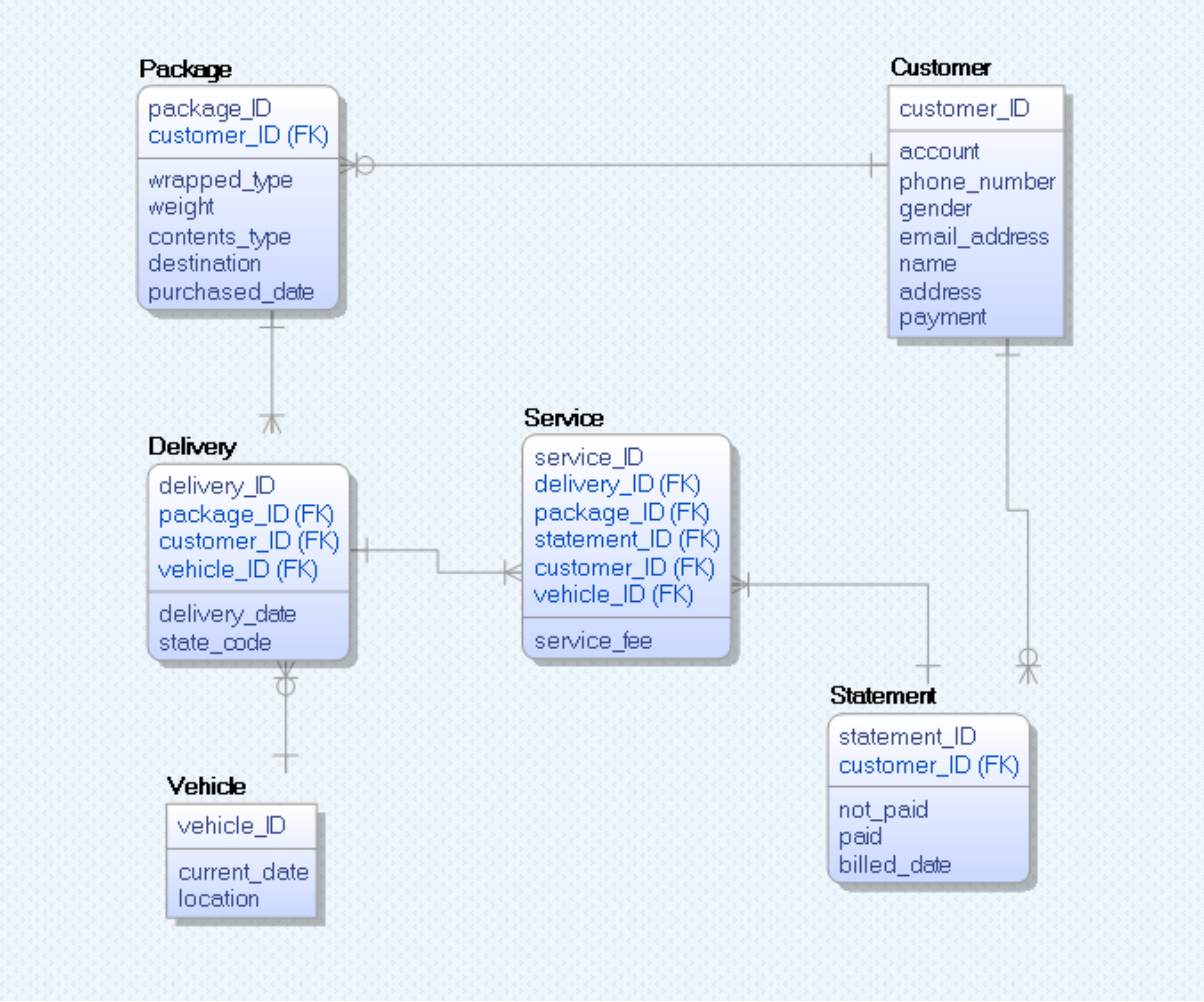
Database System Project2

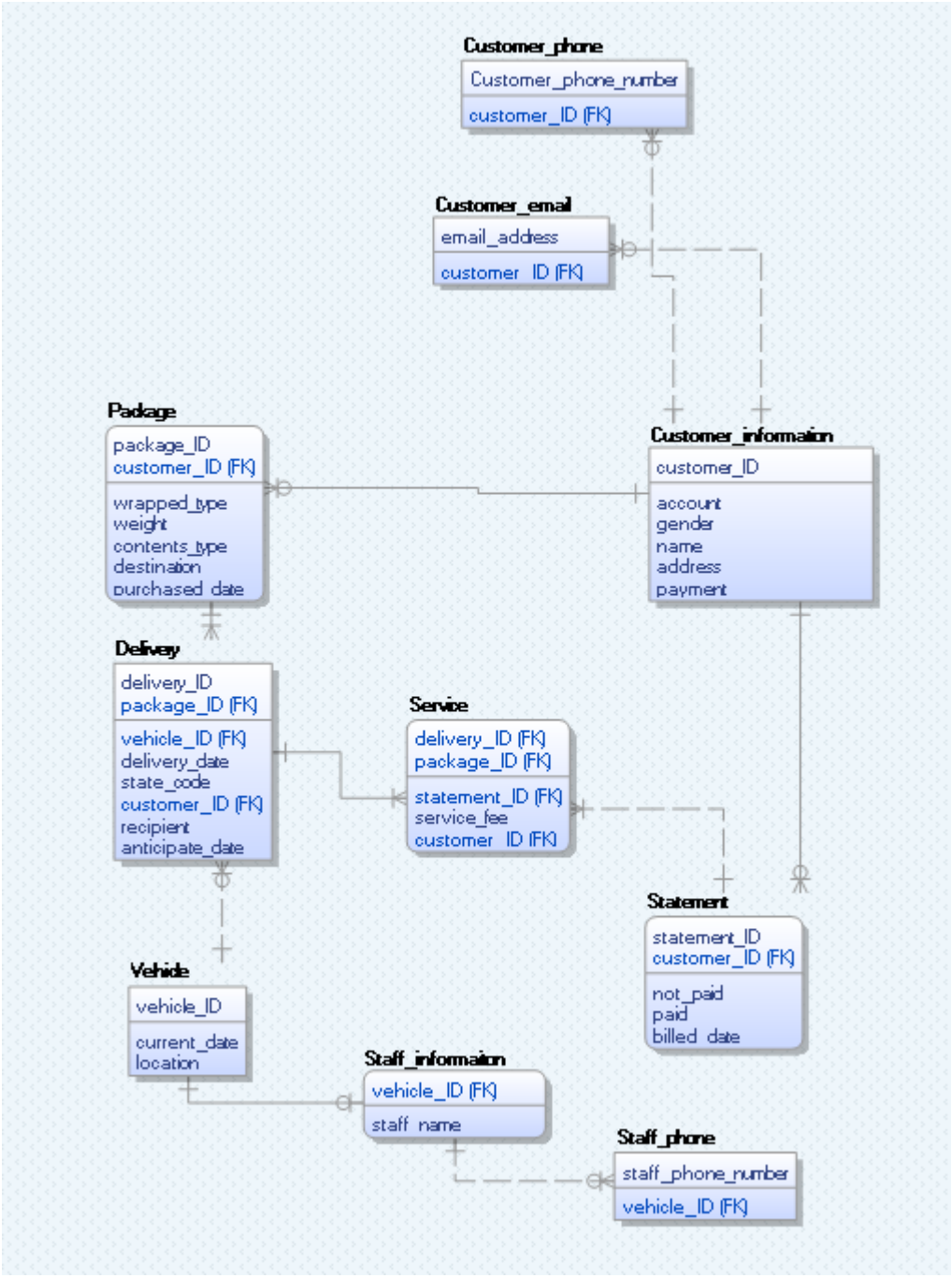
20211606 한석기

1. Project 1의 Logical Schema를 수정 및 BCNF로 Decomposition

(1) Project 1 Logical Schema



(2) Project 2 Logical Schema with BCNF



(3) Explanation

- Package entity

먼저, Package entity를 확인했다. Package_ID에 다른 key들이 모두 종속되어 있고 그 다른 key들이 customer_ID에 부분적으로 종속되어 있지 않다. 그리고 다른 key들 서로가 종속되지도 않는다. 따라서 BCNF를 만족한다.

-Customer entity

그 다음으로, Customer entity를 확인했다. Customer table에서 phone number와 email address가 다른 Key들과 종속성을 가졌다. 따라서 Customer_email, Customer_phone entity를 만들어 세 개의 entity가 BCNF를 만족하게 했다. Customer_email entity는 email_address를 Primary Key로 가지며 Customer_phone은 Customer_phone_number를 Primary Key로 가진다. 각각의 entity는 customer_ID를 가지고 있으며 Foreign Key로 연결되어 있다.

-Delivery entity

그리고 Delivery entity를 확인했다. Query에서 recipient와 예상 도착일이 필요해, delivery entity에 추가했다. delivery_ID는 vehicle_ID와는 확실히 구별된다. delivery는 하나의 package에 대한 배달 업무이며, 배송 실패, 재배송, 교환, 반품, 등의 활동에 따라 다르다. 그리고 그럴 때마다 배정되는 vehicle_ID도 다르다. delivery_ID 하나에 여러 vehicle이 배정될 수도 있다(배를 타다가 비행기를 타다가 트럭을 타는 등). 따라서 delivery_ID에 모든 속성이 종속되어 있으므로 BCNF를 만족한다.

-Vehicle entity

Vehicle entity를 확인했다. Vehicle entity에서 따로 바꿀 것은 없이 BCNF를 만족했는데 Project1에서는 못했던 Specification의 staff를 추가하라는 부분을 확인하고 추가했다. staff는 vehicle마다 한 명씩 배정되고 vehicle_ID를 그대로 사용해 구분하다. 각 스태프는 이름을 가지고 있으며 휴대폰 번호를 가지고 있다. BCNF를 만족하기 위해 staff_name과 vehicle_ID 그리고 vehicle_ID와 staff_phone_number를 구분했다.

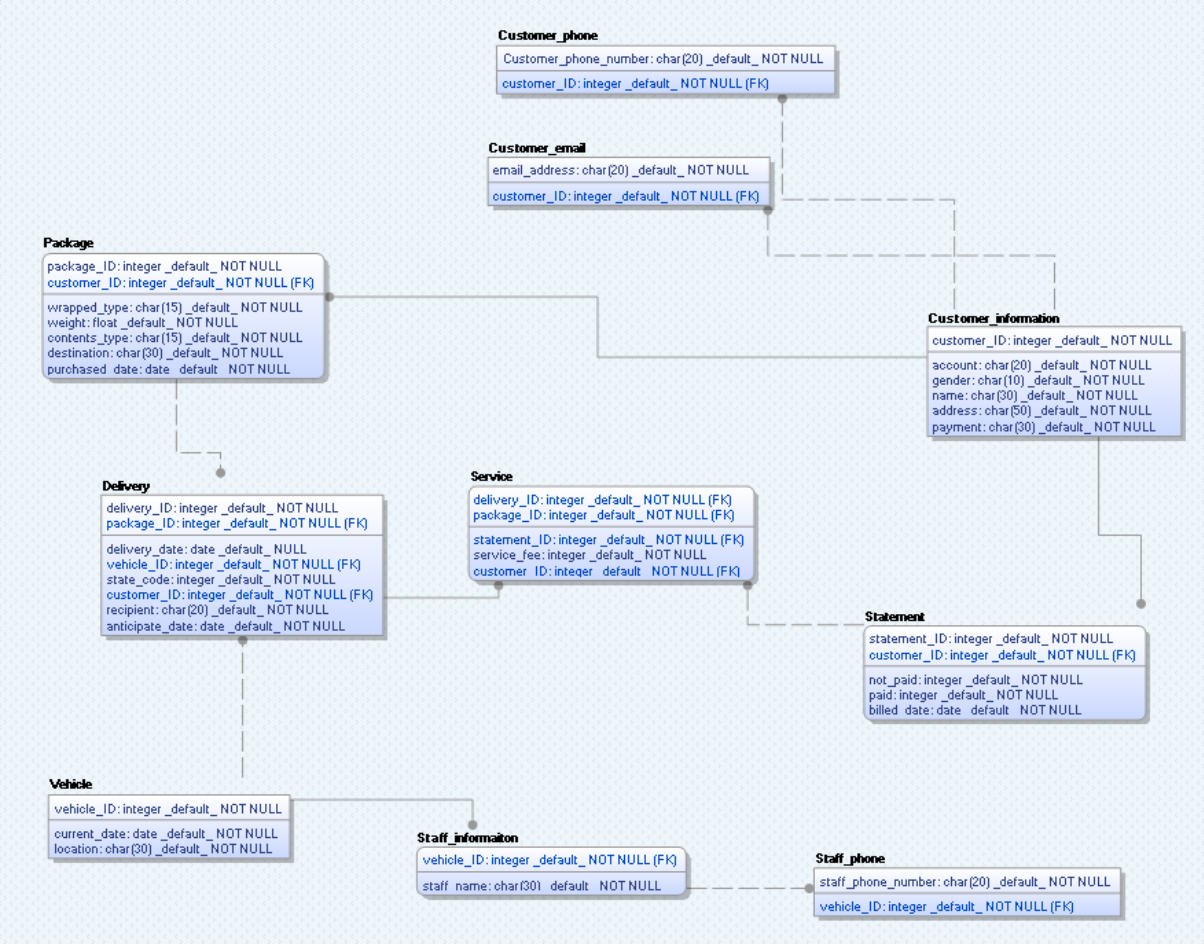
-Statement entity

statement_ID에 다른 key들이 모두 종속되어 있고 다른 key들은 서로 종속되어 있지 않기 때문에 BCNF를 만족한다.

-Service entity

service_ID와 delivery_ID가 겹쳐 service_ID를 제거했다. 그리고 vehicle_ID는 Service entity에 필요 없는 정보라 제거했다. package_ID는 여러개의 delivery를 가질 수 있으며 따라서 여러개의 service fee를 가질 수 있다. 나머지 key들 또한 delivery_ID가 결정되면 자동으로 값이 결정되므로 종속성이 있다. 그리고 해당 key들 서로 종속성이 존재하지 않아 위 entity는 BCNF를 만족한다.

2. Physical Schema diagram



위 그림은 1번에서 만든 logical schema를 physical schema로 변환한 그림이다. logical schema에서 설명한 내용과 똑같이 설계했고 data type, size, NULL여부를 표시했다.

(1) Explanation

- Package entity

*package_id: 패키지의 ID이다. 정수 id로 줄 것이라 integer이고 없을 수가 없어서 NOT NULL이다. primary key으로 활용된다.

*customer_id: 고객의 ID이다. 정수 id로 줄 것이라 integer이고 없을 수가 없어서 NOT NULL이다. 이 entity에서는 foreign key 및 primary key로 활용된다.

*wrapped_type: 포장 타입이다. big, normal, small로 줄 것이라 char타입이다. 실제 구현에서는 가변형 타입인 varchar를 사용했다. NOT NULL을 주었다.

*weight: kg단위의 package의 무게이다. 따라서 float이고 NOT NULL이다.

*contents_type: package의 배송 옵션이다. danger과 overseas가 있고 두개 옵션을 모두 주면 danger_overseas이다. 따라서 char타입으로 설정했지만 구현 시, varchar를 줬다. 그리고 None으로 설정할 수 있게 만들어 NOT NULL이다.

*destination: package의 목적지다. 주소를 입력해야 하므로 char타입으로 설정했고 구현 시, varchar를 줬다. 목적지가 없을 수 없어 NOT NULL로 설정했다.

*perchased_date: package의 구매 날짜다. 날짜이므로 date로 설정했고 NOT NULL이다.

-Customer_information entity

*customer_id: 위에 package entity에 설명되어 있고 이 entity에서 Primary key로 활용된다.

*account: customer의 계좌번호이며 char로 설정했다. 구현 시, varchar로 구현했다. 주로 결제나 환불계좌로 사용될 것이며 NOT NULL이다.

*gender: customer의 성별이다. male과 female로 줄 것이며 char로 설정했고 구현 시, varchar로 구현했다. 성별은 없을 수가 없어서 NOT NULL로 설정했다.

*name: customer의 이름이다. char로 설정했고 구현 시, varchar로 구현했다. 그리고 당연히 NOT NULL이다.

*address: customer의 주소이다. 오프라인 청구서는 이 주소로 발송될 것이며 char로 설정했고 구현 시, varchar로 구현했다. 그리고 NOT NULL이다.

*payment: customer의 결제수단이다. credit card, cell phone등이 있으며 char로 설정했고 구현 시, varchar로 구현했다. 그리고 NOT NULL이다.

-Customer_email entity

*email_address: customer의 email주소이다. 온라인 청구서는 이 주소로 발송할 것이며 char로 설정했고 구현 시, varchar로 구현했다. 또한 NOT NULL이다. 그리고 primary key로 활용되고 있다.

*customer_id: 위에서 설명한 key다. 이 entity에서는 foreign key로 활용되고 있다.

-Customer_phone entity

*customer_phone_number: customer의 휴대폰 번호이다. 입력 시 -를 입력할 수 있게 char로 설정했고 구현 시, varchar를 활용했다. 휴대번호는 배송 안내 시, 필요한 정보이므로 NOT NULL로 설정했다.

*customer_id: 위에서 설명한 key다. 이 entity에서는 foreign key로 활용되고 있다.

-Delivery entity

*delivery_id: 배송 id이다. id이므로 integer로 줬고 NOT NULL로 설정했다. 이 entity에서 primary key로 활용된다.

*package_id: 위에서 설명한 key이고 이 entity에서 foreign key와 primary key로 활용되고 있다.

*delivery_date: 배송완료된 날짜이다. 날짜이므로 date로 줬고 배송이 아직 완료되지 않았을 수 있으니까 NULL로 설정했다.

*vehicle_id: 수송 수단의 id이다. id이므로 integer로 줬고 NOT NULL로 설정했다. 그리고 이 entity에서 foreign key로 활용된다.

*state_code: 배송 당시, 수송 수단의 상태이다. 777, 555 같은 code형태로 줄 것이라 integer로 설정했다. default code는 555로 설정할 것이라 NOT NULL로 설정했다.

*customer_id: 위에서 설명한 key다. 이 entity에서는 foreign key로 활용되고 있다. 이 entity에서는 foreign키로 활용된다.

*recipient: package와 delivery의 수령인이다. 이름이므로 char로 줬고 구현 시, varchar로 설정했다. NOT NULL로 설정했다.

*anticipate_date: package와 delivery의 예상 도착일이다. 날짜이므로 date로 줬고 NOT NULL로 설정했다.

-Vehicle entity

*vehicle_id: 위에서 설명한 key고 이 entity에서 primary key로 활용된다.

*current_date: vehicle의 현재 location의 날짜이므로 date이고 NOT NULL이다.

*location: vehicle의 현재 위치이며, char로 설정했지만 구현 시, varchar를 활용했고 역시 NOT NULL이다.

-Staff_information entity

*vehicle_id: 위에서 설명한 key이고 이 entity에서 primary key와 foreign key로 활용된다.

*staff_name: vehicle의 기사의 이름이므로 char로 설정했지만, 구현 시, varchar를 활용했고 NOT NULL이다.

-Staff_phone entity

*staff_phone_number: vehicle 기사의 휴대번호이며, char로 설정했지만, 구현 시, varchar를 활용했고 NOT NULL이다.

*vehicle_id: 위에서 설명한 key이고 이 entity에서 foreign key로 활용된다.

-Statement entity

*statement_id: 명세서의 id이다. id이므로 integer로 설정했고 NOT NULL이다. 이 entity에서 primary key로 활용된다.

*customer_id: 위에서 설명한 key이며 이 entity에서 primary key와 foreign key로 활용된다.

*not_paid: 명세서에서의 아직 미납한 금액이다. 한국 돈 기준으로 했으므로 integer로 설정했고 0원으로 설정할 수 있게 NOT NULL로 설정했다.

*paid: 명세서에서의 납부한 금액이다. 한국 돈 기준으로 했으므로 integer로 설정했고 0원으로 설정할 수 있게 NOT NULL로 설정했다.

*billed_date: 명세서가 매달 나오는 날짜이다. 날짜이므로 date로 설정했고 NOT NULL이다.

-Service entity

*delivery_id: 위에서 설명한 key이고 이 entity에서 primary key와 foreign key로 활용된다.

*package_id: 위에서 설명한 key이고 이 entity에서 primary key와 foreign key로 활용된다.

*statement_id: 위에서 설명한 key이고 이 entity에서 foreign key로 활용된다.

*service_fee: delivery의 service금액을 계산해 합한 금액이다. integer로 설정했고 0원으로 하면 되므로 NOT NULL로 설정했다.

*customer_id: 위에서 설명한 key이고 이 entity에서 foreign key로 활용된다.

3. Query with C++ and mysql

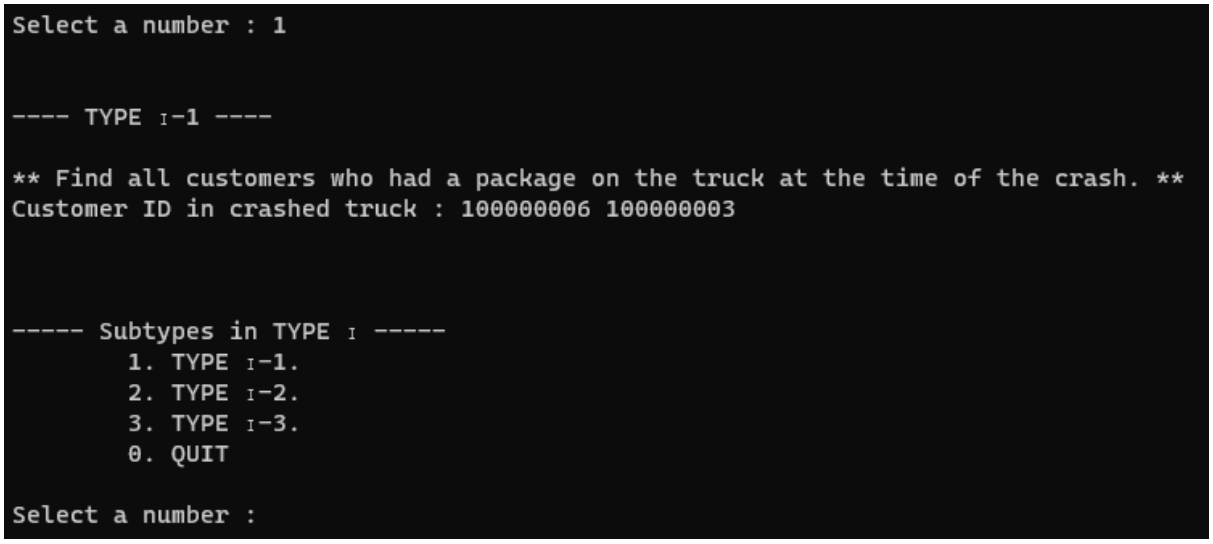
```
Connection Succeed
------- SELECT QUERY TYPES -------

        1. TYPE Ⅰ
        2. TYPE Ⅱ
        3. TYPE Ⅲ
        4. TYPE Ⅳ
        5. TYPE Ⅴ
        0. QUIT

Select a number :
```

mysql과 연결이 완료되면 connection succeed를 출력하고 TYPE를 고르라는 출력이 나온다. 그리고 1번을 입력하면 다음과 같이 나온다.

```
Select a number : 1
Assume truck X is destroyed in a crash.
Input truck X number: 300000002


----- Subtypes in TYPE Ⅰ -----
        1. TYPE Ⅰ-1.
        2. TYPE Ⅰ-2.
        3. TYPE Ⅰ-3.
        0. QUIT

Select a number :
```

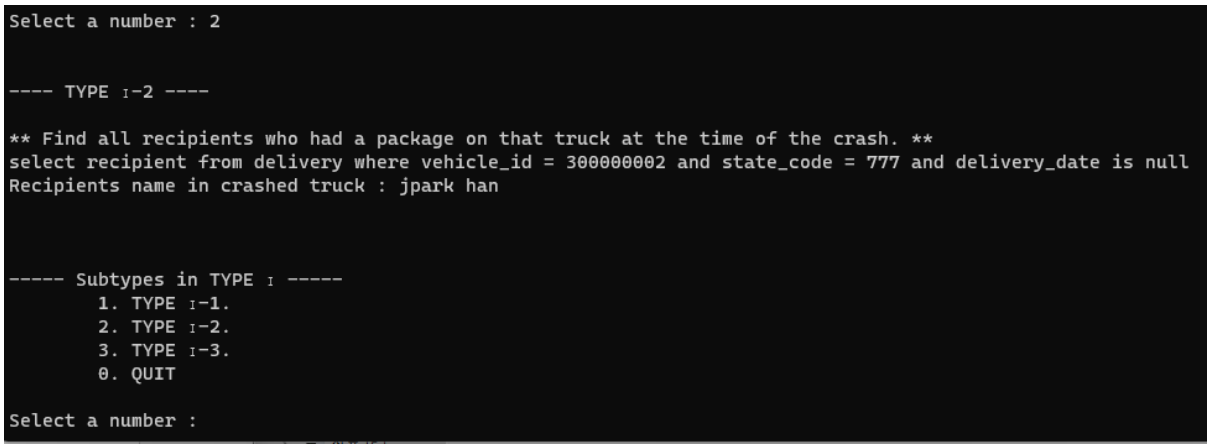
1번 쿼리의 Subquery인 1-1, 1-2, 1-3중 선택하는 것이다. 그 이후는 다음과 같다.

(1) Query 1-1



앞에서 사고난 트럭 id인 300000002를 입력했고 1-1번 쿼리를 입력했다. 그랬더니 사고난 차량에 있던 package들의 customer_id를 출력을 했다. 그리고 다시 subquery를 입력하게 form이 나왔다.

(2) Query 1-2



1-2번 쿼리를 입력했다. 사고난 차량에 있던 package들의 수령인들이 출력됐다. 그리고 다시 subquery를 입력하게 form이 나왔다.

(3) Query 1-3

```
Select a number : 3


---- TYPE Ⅰ-3 ----

** Find the last successful delivery by that truck prior to the crash. **
Last successful Delivery ID of crashed truck : 400000007



----- Subtypes in TYPE Ⅰ -----
        1. TYPE Ⅰ-1.
        2. TYPE Ⅰ-2.
        3. TYPE Ⅰ-3.
        0. QUIT

Select a number :
```

1-3번 쿼리를 실행하게 숫자를 입력했다. 사고난 차량의 사고나기 전, 마지막 성공적인 배송 id를 출력했다. 그리고 다시 subquery입력 form이 나왔다.

```
Select a number : 0
------- SELECT QUERY TYPES -------

        1. TYPE Ⅰ
        2. TYPE Ⅱ
        3. TYPE Ⅲ
        4. TYPE Ⅳ
        5. TYPE Ⅴ
        0. QUIT

Select a number :
```

subquery입력 부분에서 0을 입력했더니 subquery입력 form을 빠져나오고 query입력 form으로 다시 돌아왔다.

(4) Query 2

```
Select a number : 2


---- TYPE Ⅱ ----

** Find the customer who has shipped the most packages in certain year. **
 Which Year? : 2023
The customer ID who has shipped the most packages in 2023 : 100000001

------- SELECT QUERY TYPES -------

        1. TYPE Ⅰ
        2. TYPE Ⅱ
        3. TYPE Ⅲ
        4. TYPE Ⅳ
        5. TYPE Ⅴ
        0. QUIT

Select a number :
```

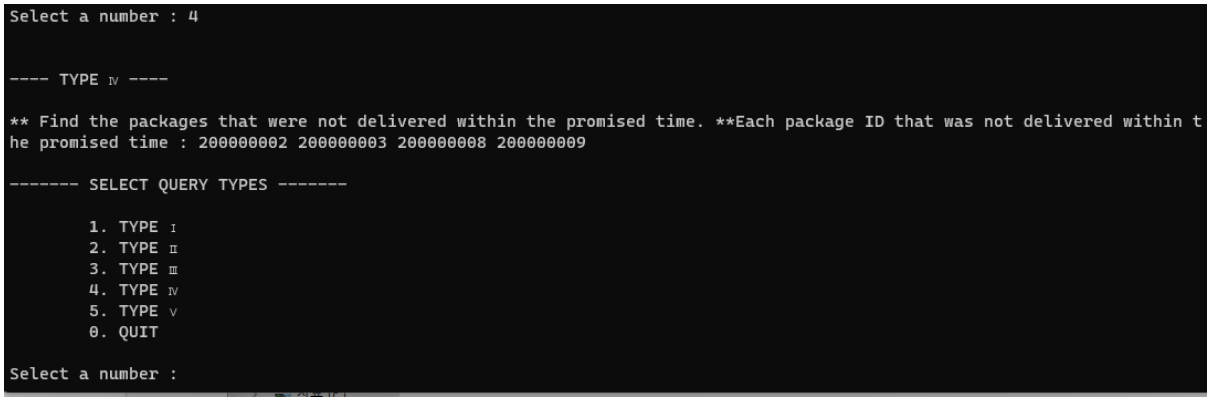
2번 쿼리를 실행하게 숫자를 입력했다. 년도를 입력 받고 해당 년도에 가장 많은 배송을 주문한 customer_id를 출력했다. 그리고 다시 query입력 form으로 돌아왔다.

(5) Query 3

```
Select a number : 3


---- TYPE Ⅲ ----

** Find the customer who has spent the most money on shipping in the past year. **
 Which Year? : 2023
The customer ID who has spent the most money on shipping in 2023 : 100000002

------- SELECT QUERY TYPES -------

        1. TYPE Ⅰ
        2. TYPE Ⅱ
        3. TYPE Ⅲ
        4. TYPE Ⅳ
        5. TYPE Ⅴ
        0. QUIT

Select a number :
```

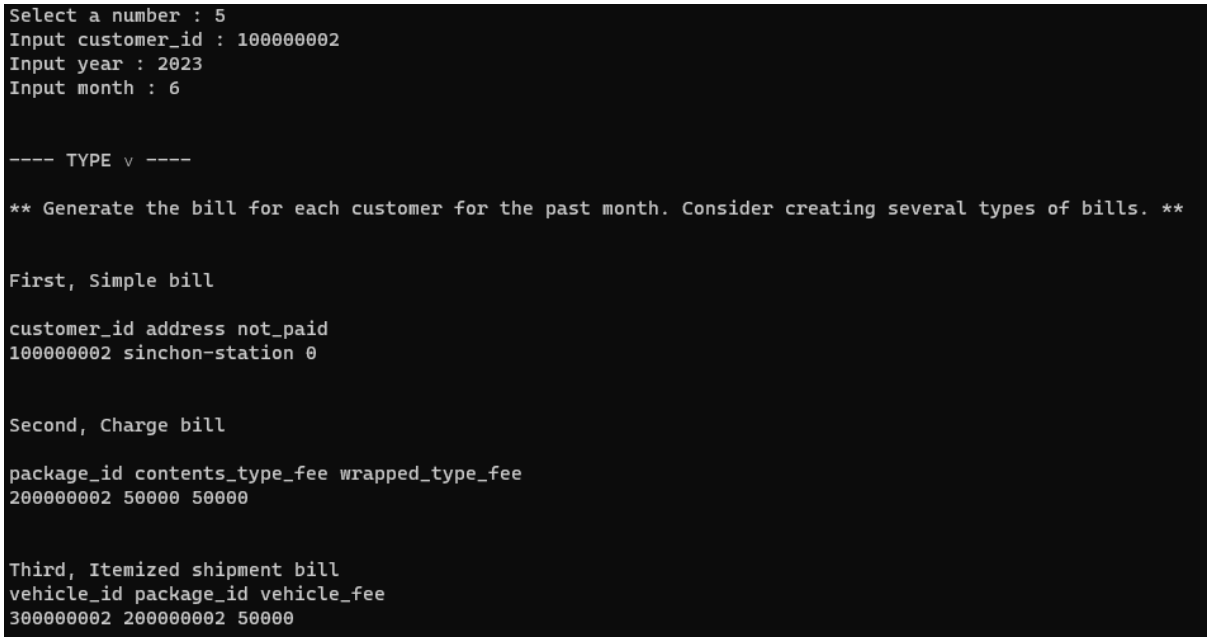
3번 쿼리를 실행하게 숫자를 입력했다. 년도를 입력 받고 해당 년도에 배송을 진행한 내역 중 가장 많은 지출을 한 customer_id를 출력했다. 그리고 다시 query입력 form으로 돌아왔다.

(6) Query 4



4번 쿼리를 실행하게 숫자를 입력했다. 약속된 시간에 도착하지 않은 delivery_id를 출력하는 쿼리인데 총 4개의 배송이 약속된 시간에 도착하지 않았다. 그리도 다시 query입력 form으로 돌아왔다.

(7) Query 5



5번 쿼리를 실행하게 숫자를 입력했다. 그리고 customer_id, year, month를 순서대로 입력했다. 2023년 6월 청구서를 총 3개 출력했다. 첫번째는 고객 ID와 주소, 미납금액을 출력했고 두번째는 해당 고객이 배송시킨 패키지의 ID와 service 요금을 각각 contents_type, wrapped_type에 따라 보여줬다. 세번째는 운송수단의 ID와 패키지의 ID 그리고 운송 요금을 각각 출력했다.

```
Input customer_id : 100000001
Input year : 2023
Input month : 6


---- TYPE Ⅴ ----

** Generate the bill for each customer for the past month. Consider creating several types of bills. **


First, Simple bill

customer_id address not_paid
100000001 suwon-station 60000


Second, Charge bill

package_id contents_type_fee wrapped_type_fee
200000001 0 10000
200000007 50000 50000


Third, Itemized shipment bill
vehicle_id package_id vehicle_fee
300000001 200000001 10000
300000002 200000007 50000
```

5번 쿼리를 다른 고객 ID를 입력해 출력한 것이다. 첫번째 청구서는 요약된 것이기에 한 줄로 출력됐고, 두번째 청구서는 각 서비스 요금을 보여준다. 세번째 청구서는 운송수단 종류에 따른 요금을 출력한다.

```
------- SELECT QUERY TYPES -------

        1. TYPE Ⅰ
        2. TYPE Ⅱ
        3. TYPE Ⅲ
        4. TYPE Ⅳ
        5. TYPE Ⅴ
        0. QUIT

Select a number : 0

E:\대학 자료\2학년 1학기\자료구조\과제\
니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

쿼리를 선택하는 창에서 0을 입력하면 프로그램이 종료된다.

4. 특이사항

(1) 요금 책정

- vehicle_id는 300000000~399999999까지이며, 짝수는 배, 비행기로 50000원이며, 홀수는 트럭으로 10000원이다.

- wrapped_type은 big이면 50000원, normal이면 30000원, small이면 10000원이다.

- contents_type은 Overseas면 100000원, Danger이면 50000원, Overseas_Danger이면 200000원이다.

- 따라서 service_fee는 wrapped_type_fee + contents_type_fee이고 전체 fee는 service_fee + vehicle_fee이다.

(2) ID 부여

- customer_id는 100000000~199999999까지다.

- package_id는 200000000~299999999까지다.

- vehicle_id는 300000000~399999999까지다.

- delivery_id는 400000000~499999999까지다.

- statement_id는 500000000~599999999까지다.

- customer_id는 고객에게 해외 관세번호처럼 알려준다고 가정한다. 그리고 해당 customer_id와 휴대번호로 인증을 통해 package_id, delivery_id를 조회 가능하다고 가정한다.

(3) 대략적인 코드 설명

먼저 mysql과의 연결을 하고 연결 성공 시, 20211606_create.txt 파일을 열어 해당 쿼리문들을 읽으며 테이블을 생성하고 값을 넣는다. 그리고 쿼리를 사용자에게 입력 받는다. 쿼리는 c++ 코드에 미리 구현이 되어있고 사용자는 미리 구현된 쿼리만을 실행할 수 있다. 모든 쿼리는 while과 switch문을 사용해 입력을 처리했고 번호 입력에 따른 쿼리문 작성은 char 배열을 이용했다. char 배열을 여러 개 선언해 base query를 미리 넣어 놓고 dynamic variable들은 run time에 입력 받은 후에 base query와 연결해 mysql에서 해당 쿼리를 실행하게 구현했다. 이때, 연결하는 함수는 sprintf함수를 사용했다. 서브쿼리던 쿼리던 쿼리를 선택하는 부분에서 0번을 입력해 quit하는 부분은 flag로 처리했다. 프로그램의 마지막에 20211606_drop.txt 파일을 열어 앞에서 만든 테이블들을 drop한다. 그리고 프로그램을 종료한다.