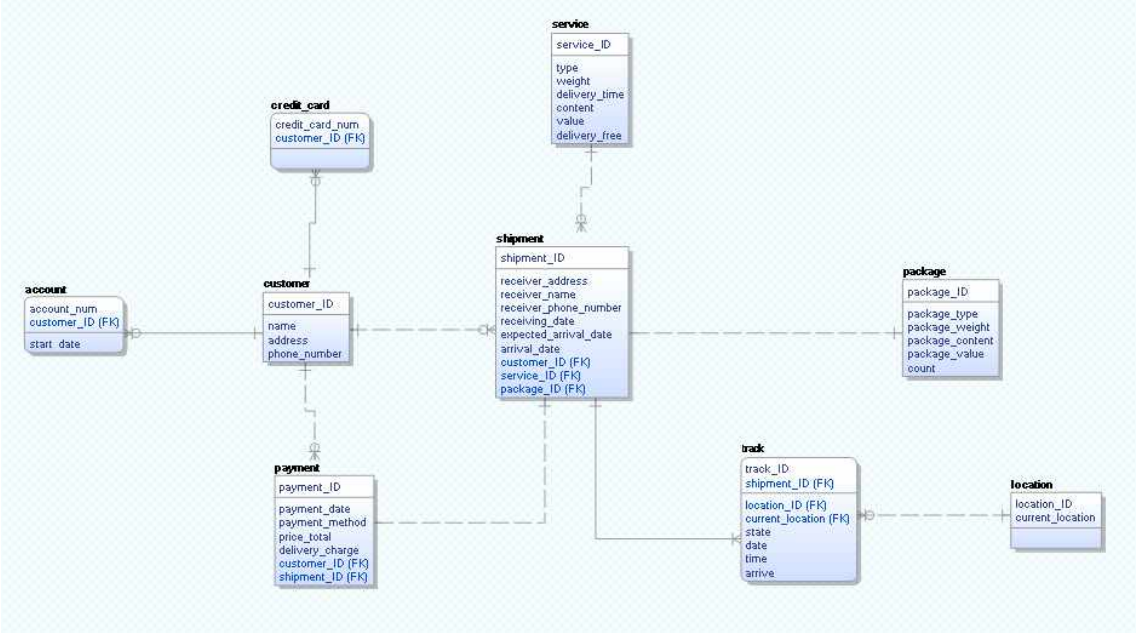
CSE4110 - Database System

Spring2023

전공 : 컴퓨터공학과
학번 : 20211569
이름 : 이승연



<Logical Schema>

1. BCNF Decomposition

모든 entity와 relation에 대해 BCNF를 만족해야 하기 때문에 project1에서 설정한 relation을 살펴보겠다. 이를 위해 각 relation의 모든 함수종속(FD) a->b가 trivial하거나 a가 super key인지 체크해야 한다. 만약 둘 중 하나도 만족하지 않는 경우 Decompotion을 해야 한다.

- shipment

shipment_ID -> receiver_address, receiver_name, receiver_phone_number, receiving_date, expected_arrival_date, arrival_date, customer_ID, service_ID, package_ID

이외에는 다른 FD가 존재하지 않는다. shipment_ID는 PK이기 때문에 BCNF Dependency를 만족한다.

- customer

customer_ID -> name, address, phone_number

이외에는 다른 FD가 존재하지 않는다. 동명이인의 경우에는 customer_ID로 식별 가능하게 만들었기 때문에 다른 요소가 customer의 속성을 결정짓지 못 한다. customer_ID는 FK이기 때문에 customer은 BCNF Dependency를 만족한다.

- package

package_ID -> package_type, package_weight, package_content, package_value, count

이외에는 다른 FD가 존재하지 않는다. package_ID는 PK이기 때문에 BCNF Dependency를 만족한다.

package_delivery_time 속성은 shipment의 arrival_date와 겹치기 때문에 삭제했다.

- tracking

track_ID, shipment_ID -> location_ID, current_location, arrival, state, tracking_date, tracking_time

이외에는 다른 FD가 존재하지 않는다. track_ID, shipment_ID는 PK이기 때문에 BCNF Dependency를 만족한다.

arrival 속성을 추가하여 추적에서 다음 운송수단을 설정하였다. 예를 들어 truck 1721 다음에 warehouse 4169로 이동한다면 current location은 truck, arrival은 warehouse 4169가 될 것이다.

- location

PK인 location_ID, current_location을 제외하면 다른 FD가 존재하지 않는다. 따라서 BCNF Dependency를 만족한다.

- service

service_ID -> type, weight, delivery_time, content, value, delivery_free

이외에는 다른 FD가 존재하지 않는다. service_ID는 PK이기 때문에 BCNF Dependency를 만족한다.

서비스에 따른 배송비를 나타내기 위해 delivery_free 속성을 추가했다.

- payment

payment_ID -> payment_date, payment_method, price_total, delivery_charge, customer_ID, shipment_ID

이외에는 다른 FD가 존재하지 않는다. payment_ID는 PK이기 때문에 BCNF Dependency를 만족한다.

- account

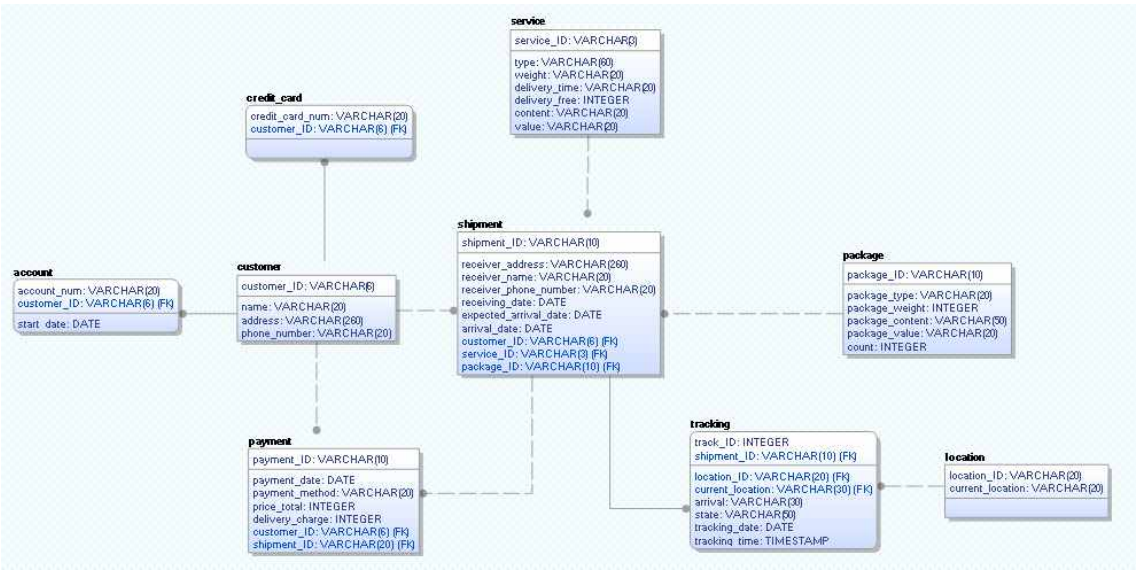
account_num, customer_ID -> start_date

이외에는 다른 FD가 존재하지 않는다. account_num, customer_ID는 PK이기 때문에 BCNF Dependency를 만족한다.

- credit_card
PK인 credit_card_num, customer_ID을 제외하면 다른 FD가 존재하지 않는다. 따라서 BCNF Dependency를 만족한다.

위와 같이 모든 relation을 검토했을 때 모든 relation이 BCNF Dependency를 만족한다고 할 수 있다. project1에서 설정을 잘 해두었기 때문에 이번 프로젝트에서 약간의 attribute 변경만 하면 됐다. 결과적으로 Decomposed Logical Shema는 이전의 Logical Schema랑 같다.

2. Physical Schema



<Physical Schema>

default 값으로 데이터 유형은 VARCHAR(20), null값 여부는 null값을 허용하지 않는 not null로 설정했다. relation마다 자세히 살펴보자.

- shipment
shipment_ID : S04-000115 형태로 VARCHAR(10)으로 설정했다. PK이므로 not null이다.
receiver_address : 택배를 보낼 주소이다. VARCHAR(260)으로 설정했다. 필수적이므로 not null로 설정했다.
receiver_name : 택배 수령인 이름이다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.
receiver_phone_number : 택배 수령인 전화번호이다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.
receiving_date : "2023-05-21" 형태로 택배를 접수한 날이다. DATE로 설정했다.

expected_arrival_date : 도착 예정일이다. 마찬가지로 DATE로 설정했다.
arrival_date : 실제 도착일이다. 마찬가지로 DATE로 설정했다.
customer_ID, service_ID, package_ID : FK이고 모두 not null로 설정했다.

- customer
customer_ID : C00115 형태로 VARCHAR(6)으로 설정했다. PK이므로 not null이다.
name : 고객의 이름이다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.
address : 고객의 주소이다. VARCHAR(260)으로 설정했다. 필수적이므로 not null로 설정했다.
phone_number : 고객의 전화번호이다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.

- package
package_ID : P51421 형태로 VARCHAR(10)으로 설정했다. PK이므로 not null이다.
package_type : 택배 유형 flat envelope, small box, large boxes을 결정한다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.
package_weight : 택배 최대 용량을 나타낸다. integer로 설정했다.
package_content : 택배 물품이다. VARCHAR(50)으로 설정했다. 필수적이므로 not null로 설정했다.
package_value : 물품 가액이다. VARCHAR(20)으로 설정했다.
count : 합배송을 고려해 몇 개의 물품을 보냈는지 알려주는 속성이다. integer로 설정했다.

- tracking
tracking_ID : 0, 1, 2, 3, 4까지 존재하며 식별자 역할을 한다. integer로 설정했다. PK이므로 not null이다.
shipment_ID : shipment에서 가져온 FK이다.
location_ID : 현재 운송수단의 ID이다. location에서 가져온 FK이다.
current_location : 현재 운송수단이다. location에서 가져온 FK이다.
arrival : 다음 목적지인 운송수단과 ID이다. VARCHAR(30)으로 설정했다.
state : 해당 track에 대한 배송 상태이다. 상품 인수, 상품 이동, 배송 출발, 배송 완료를 알려준다. VARCHAR(50)으로 설정했다. 필수적이므로 not null로 설정했다.
tracking_date : 해당 track에 대한 날짜이다. "2023-05-21"형태이다. DATE로 설정했다.
tracking_time : 해당 track에 대한 시간이다. "15:46:00"형태이다. TIME으로 설정했다.

- location
location_ID : 현재 운송수단의 ID이다. VARCHAR(20)으로 설정했다. PK이므로 not null이다.
current_location : truck, airplane, warehouse 등 현재 운송수단이다. VARCHAR(20)으로 설정했다. PK이므로 not null이다.

- service
service_ID : S01 형태로 VARCHAR(3)으로 설정했다. PK이므로 not null이다.
type : 택배 유형 flat envelope, small box, large boxes을 결정한다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.
weight : light, heavy를 결정한다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.
delivery_time : 배송 기간이다. one day, two weeks 등 서비스 유형에 따른 배송 기간을 나타낸다. VARCHAR(20)으로 설정했다. 필수적이므로 not null로 설정했다.
delivery_free : 서비스 유형에 따른 배송비이다. integer로 설정했다. 필수적이므로 not null로 설정했다.
content : hazard인지 아닌지 나타내는 속성이다. VARCHAR(20)으로 설정했다.
value : 세관 신고서가 필요한지 아닌지 나타내는 속성이다. VARCHAR(20)으로 설정했다.

- payment
payment_ID : B00000115 형태로 VARCHAR(10)으로 설정했다. PK이므로 not null이다.
payment_date : 청구된 계산서에 대한 지불이 일어난 날이다. VARCHAR(20)으로 설정했다.
price_total : 해당 계산서에 청구된 요금이다. integer로 설정했다.
delivery_charge : 추가 배송비이다. service.delivery_free에 추가적으로 청구된다. integer로 설정했다.
customer_ID, shipment_ID : FK이고 모두 not null로 설정했다.

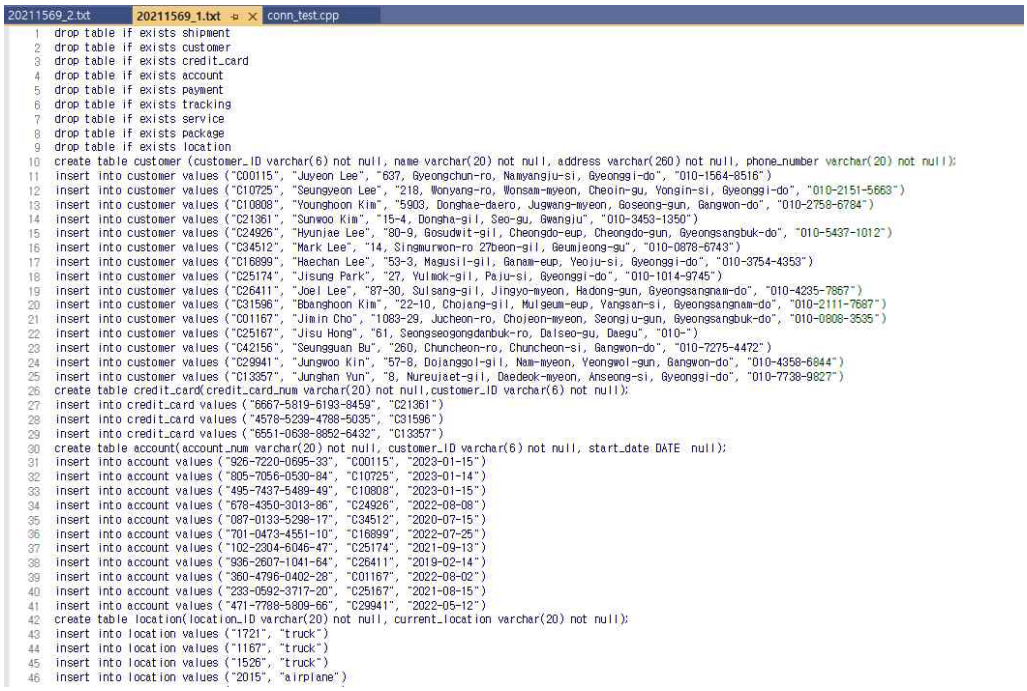
- account
account_num : 계좌번호로 926-7220-0695-33 형태이며 VARCHAR(20)으로 설정했다. PK이므로 not null이다.
customer_ID : FK이고 not null로 설정했다.
start_date : 실제 결제일이다. DATE로 설정했다.

- credit_card

credit_card_num : 카드번호로 6667-5819-6193-8459 형태이며 VARCHAR(20)으로 설정했다. PK이므로 not null이다.
customer_ID : FK이고 not null로 설정했다.

3. Query 실행

먼저 쿼리 실행을 위해 txt 파일을 만들어 CRUD를 진행하였다.



```
drop table if exists shipment
drop table if exists customer
drop table if exists credit_card
drop table if exists account
drop table if exists payment
drop table if exists tracking
drop table if exists service
drop table if exists package
drop table if exists location
create table customer (customer_ID varchar(6) not null, name varchar(20) not null, address varchar(260) not null, phone_number varchar(20) not null);
insert into customer values ("C00115", "Juyeon Lee", "637, Gyeongchun-ro, Namyangju-si, Gyeonggi-do", "010-1564-8516")
insert into customer values ("C10725", "Seungyeon Lee", "218, Wonyang-ro, Wonsam-myeon, Cheoin-gu, Yongin-si, Gyeonggi-do", "010-2151-5663")
insert into customer values ("C10808", "Younghoon Kim", "5903, Donghae-daero, Jugwang-myeon, Goseong-gun, Gangwon-do", "010-2758-6784")
insert into customer values ("C21361", "Sunwoo Kim", "15-4, Dongha-gil, Seo-gu, Gwangju", "010-3453-1350")
insert into customer values ("C24926", "Hyunjae Lee", "80-9, Gosudwit-gil, Cheongdo-eup, Cheongdo-gun, Gyeongsangbuk-do", "010-5437-1012")
insert into customer values ("C34512", "Mark Lee", "14, Singmurwon-ro 27beon-gil, Geumjeong-gu", "010-0878-6743")
insert into customer values ("C16899", "Haechan Lee", "53-3, Magusil-gil, Ganam-eup, Yeoju-si, Gyeonggi-do", "010-3754-4353")
insert into customer values ("C25174", "Jisung Park", "27, Yulmok-gil, Paju-si, Gyeonggi-do", "010-1014-9745")
insert into customer values ("C26411", "Joel Lee", "87-30, Sulsang-gil, Jingyo-myeon, Hadong-gun, Gyeongsangnam-do", "010-4235-7867")
insert into customer values ("C31596", "Bbanghoon Kim", "22-10, Chojang-gil, Mulgeum-eup, Yangsan-si, Gyeongsangnam-do", "010-2111-7687")
insert into customer values ("C01167", "Jimin Cho", "1083-29, Jucheon-ro, Chojeon-myeon, Seongju-gun, Gyeongsangbuk-do", "010-0808-3535")
insert into customer values ("C25167", "Jisu Hong", "61, Seongseogongdanbuk-ro, Dalseo-gu, Daegu", "010-")
insert into customer values ("C42156", "Seungguan Bu", "260, Chuncheon-ro, Chuncheon-si, Gangwon-do", "010-7275-4472")
insert into customer values ("C29941", "Jungwoo Kin", "57-8, Dojanggol-gil, Nam-myeon, Yeongwol-gun, Gangwon-do", "010-4358-6844")
insert into customer values ("C13357", "Junghan Yun", "8, Nureujaet-gil, Daedeok-myeon, Anseong-si, Gyeonggi-do", "010-7738-9827")
create table credit_card(credit_card_num varchar(20) not null,customer_ID varchar(6) not null);
insert into credit_card values ("6667-5819-6193-8459", "C21361")
insert into credit_card values ("4578-5239-4788-5035", "C31596")
insert into credit_card values ("6551-0638-8852-6432", "C13357")
create table account(account_num varchar(20) not null, customer_ID varchar(6) not null, start_date DATE  null);
insert into account values ("926-7220-0695-33", "C00115", "2023-01-15")
insert into account values ("805-7056-0530-84", "C10725", "2023-01-14")
insert into account values ("495-7437-5489-49", "C10808", "2023-01-15")
insert into account values ("678-4350-3013-86", "C24926", "2022-08-08")
insert into account values ("087-0133-5298-17", "C34512", "2020-07-15")
insert into account values ("701-0473-4551-10", "C16899", "2022-07-25")
insert into account values ("102-2304-6046-47", "C25174", "2021-09-13")
insert into account values ("936-2607-1041-64", "C26411", "2019-02-14")
insert into account values ("360-4796-0402-28", "C01167", "2022-08-02")
insert into account values ("233-0592-3717-20", "C25167", "2021-08-15")
insert into account values ("471-7788-5809-66", "C29941", "2022-05-12")
create table location(location_ID varchar(20) not null, current_location varchar(20) not null);
insert into location values ("1721", "truck")
insert into location values ("1167", "truck")
insert into location values ("1526", "truck")
insert into location values ("2015", "airplane")
```

20211569_1.txt는 table이 남아있다면 모두 drop하고 table을 INSERT하도록 한다.

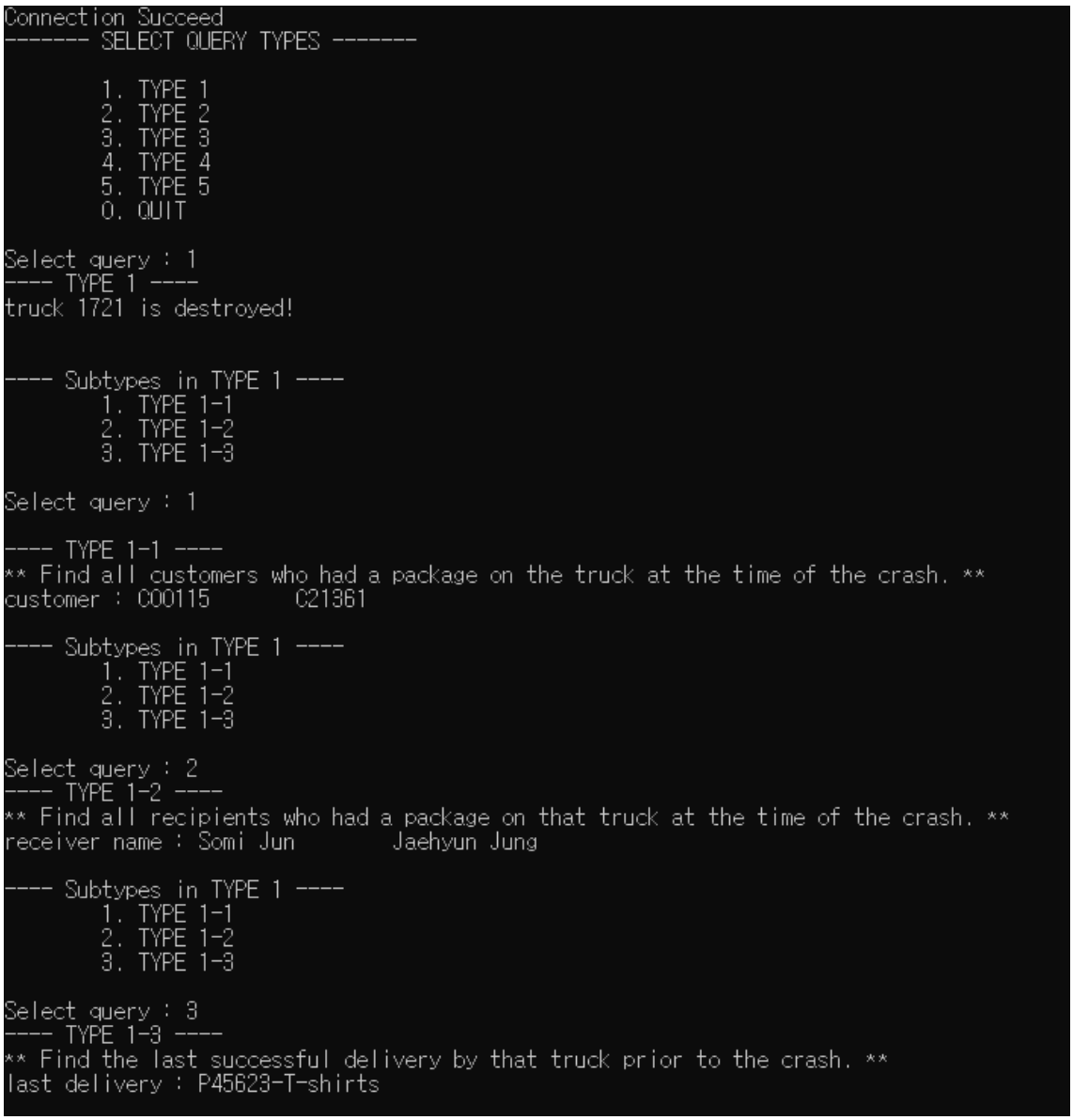
```
delete from shipment
delete from customer
delete from credit_card
delete from account
delete from payment
delete from tracking
delete from service
delete from package
delete from location
drop table shipment
drop table customer
drop table credit_card
drop table account
drop table payment
drop table tracking
drop table service
drop table package
drop table location
```

20211569_2.txt는 table을 drop하게 한다.

- TYPE 1

명세서에 나와있는 대로 truck 1721가 충돌한 것으로 가정하고 쿼리를 실행시켰다. 나는 2023-05-21에 충돌한 것으로 가정했다. 사용자에게 입력(1, 2, 3)을 받고 그에 따른 쿼리를 실행한다. 0을 누르면 이전 select 메뉴로 돌아간다.

- TYPE 1-1 : 2023-05-21에 truck 1721에 택배가 실려 있던 모든 고객의 customer_ID을 반환한다. 동명이인의 가능성 때문에 이름 대신 customer_ID를 반환했다.
- TYPE 1-2 : 2023-05-21에 truck 1721에 택배를 싣고 있던 모든 수취인의 이름을 반환한다.
- TYPE 1-3 : 트럭이 충돌 전 마지막으로 배달한 물품의 package_ID와 상품 정보를 반환한다.

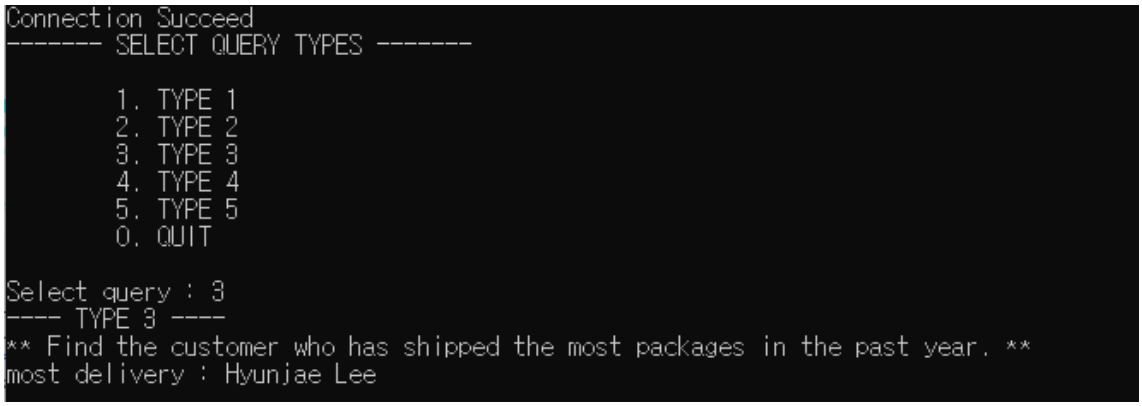


- TYPE 2

사용자에게 입력받은 년도에 가장 많은 package 발송한 고객의 이름을 반환한다. customer, shipment를 join해서 쿼리를 작성했다. 20211569_1.txt에는 2023년도만을 작성했기 때문에 input값으로 2023을 쳐야한다. 2023년에 가장 package를 많이 발송한 사람은 Bbanghoon Kim으로 적절한 결과가 나오는 것을 알 수 있다.

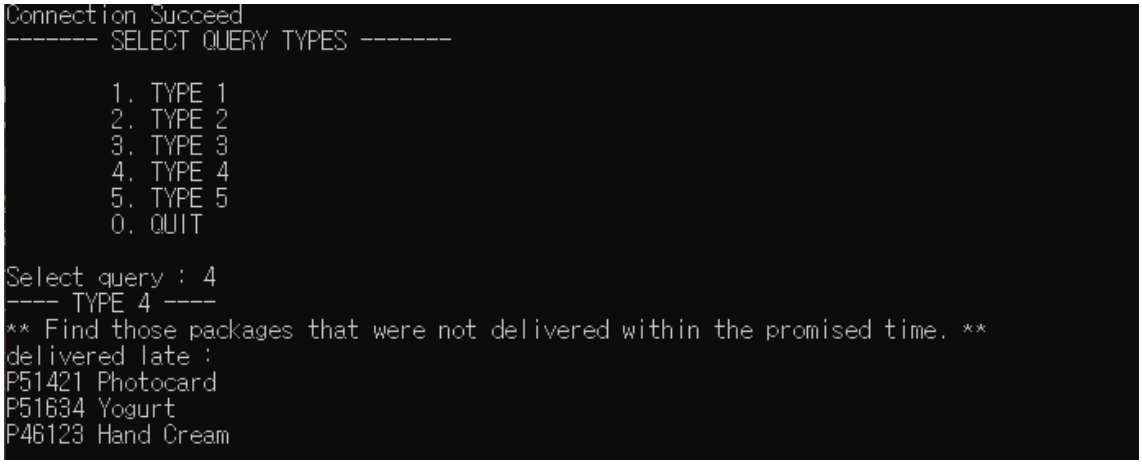
- TYPE 3

최근 1년간 가장 많은 배송비를 지출한 고객의 이름을 반환한다. 이번엔 사용자에게 년도를 입력받지 않고 현재 날짜 기준으로 쿼리를 작성했다. service의 delivery_free, payment의 delivery_charge 합으로 배송비를 계산하여 가장 높은 금액이 청구된 고객을 찾는다. Hyunjae Lee는 총 2개의 물건을 구매했고 각각 7000, 23000의 배송비로 총 30000원의 배송비를 지출했다. 가장 높은 배송비를 지출한 고객이다.



- TYPE 4

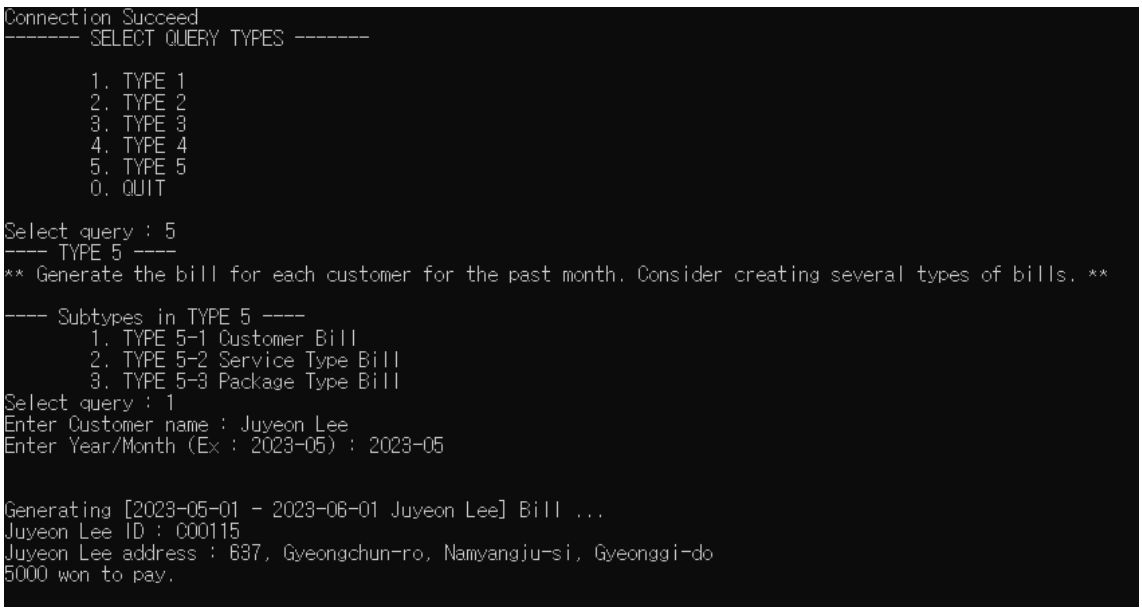
예정 배송날짜 내에 배송되지 않은 package의 ID와 content를 반환한다. shipment에서 쉽게 날짜 비교를 통해 쿼리를 작성할 수 있다. 20211569_1.txt에는 총 3개의 물품이 예상날짜보다 늦게 도착했다.



- TYPE 5

다양한 청구서를 생성한다. 총 3가지의 청구서를 생성했다. TYPE 1과 같이 사용자의 입력(1, 2, 3)에 따라 청구서를 볼 수 있도록 코드를 작성했다. TYPE 1과 마찬가지로 0을 누르면 이전 이전 select 메뉴로 돌아간다.

- TYPE 5-1 : 고객의 이름과 보고 싶은 청구서의 년도, 월을 입력하면 해당하는 달의 고객 청구서가 생성된다. payment를 이용해 적절히 쿼리를 작성할 수 있다. 예시로 Juyeon Lee의 2023-05의 청구서는 다음 사진과 같다. 또 다른 테스트를 위해 Seungyeon Lee, 2023-02 혹은 Hyunjae Lee 2023-02를 입력할 수 있다.



- TYPE 5-2 : 서비스 유형에 따른 청구서가 생성된다. 20211569_1.txt에 작성된 서비스 유형들에 대해서만 출력되므로 S02나 S04 등은 출력되지 않는다. shipment와 payment를 join해서 쿼리를 생성한다.

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

        1. TYPE 1
        2. TYPE 2
        3. TYPE 3
        4. TYPE 4
        5. TYPE 5
        0. QUIT

Select query : 5
---- TYPE 5 ----
** Generate the bill for each customer for the past month. Consider creating several types of bills. **

---- Subtypes in TYPE 5 ----
        1. TYPE 5-1 Customer Bill
        2. TYPE 5-2 Service Type Bill
        3. TYPE 5-3 Package Type Bill
Select query : 2


Generating [Service Type] Bill ...
S01 : 5000
S03 : 90000
S05 : 2000
S07 : 60000
S08 : 120000
S09 : 60000
S11 : 600000
S12 : 300000
S13 : 310000
S14 : 1900000
S15 : 10000000
```

- TYPE 5-3 : 각 발송물과 요금을 나열한 청구서가 생성된다. package를 이용해 적절히 쿼리를 작성했다.

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

        1. TYPE 1
        2. TYPE 2
        3. TYPE 3
        4. TYPE 4
        5. TYPE 5
        0. QUIT

Select query : 5
---- TYPE 5 ----
** Generate the bill for each customer for the past month. Consider creating several types of bills. **

---- Subtypes in TYPE 5 ----
        1. TYPE 5-1 Customer Bill
        2. TYPE 5-2 Service Type Bill
        3. TYPE 5-3 Package Type Bill
Select query : 3
0


Generating [Package Type] Bill ...
Ball cap : 60000
Bascketball : 600000
LG Moniter : 400000
Table : 600000
IPHONE 14 pro : 10000000
Company paper : 5000
NIKE JORDEN : 1500000
T-shirts : 250000
Hand Cream : 120000
Photocard : 90000
Yogurt : 60000
PC charger : 60000
Mouse : 300000
Love Letter : 2000
Airpot MAX : 1400000
```