Project 2

Normalization and Query Processing

20191232 김도현

1. BCNF Decomposition

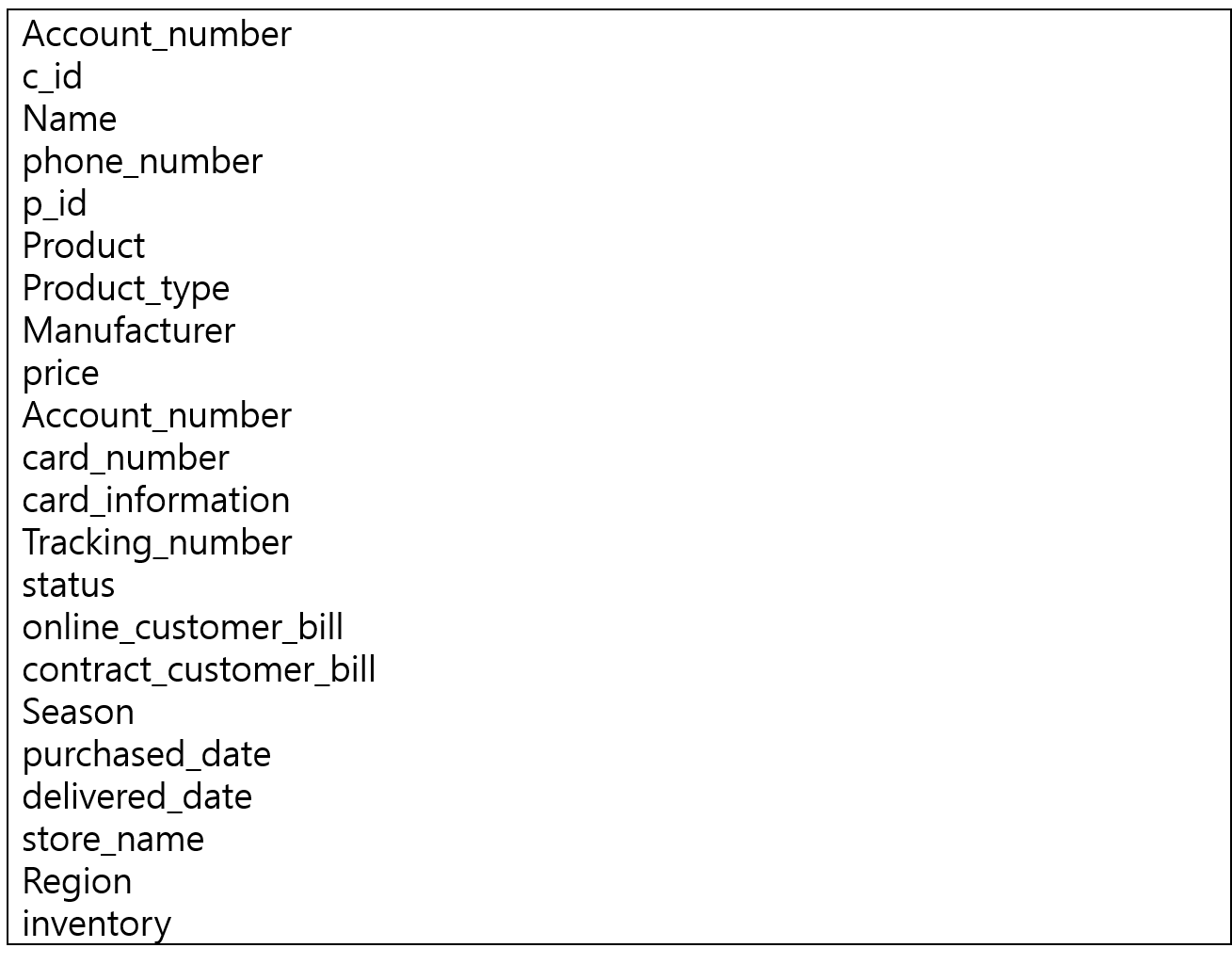


Figure 전체테이블

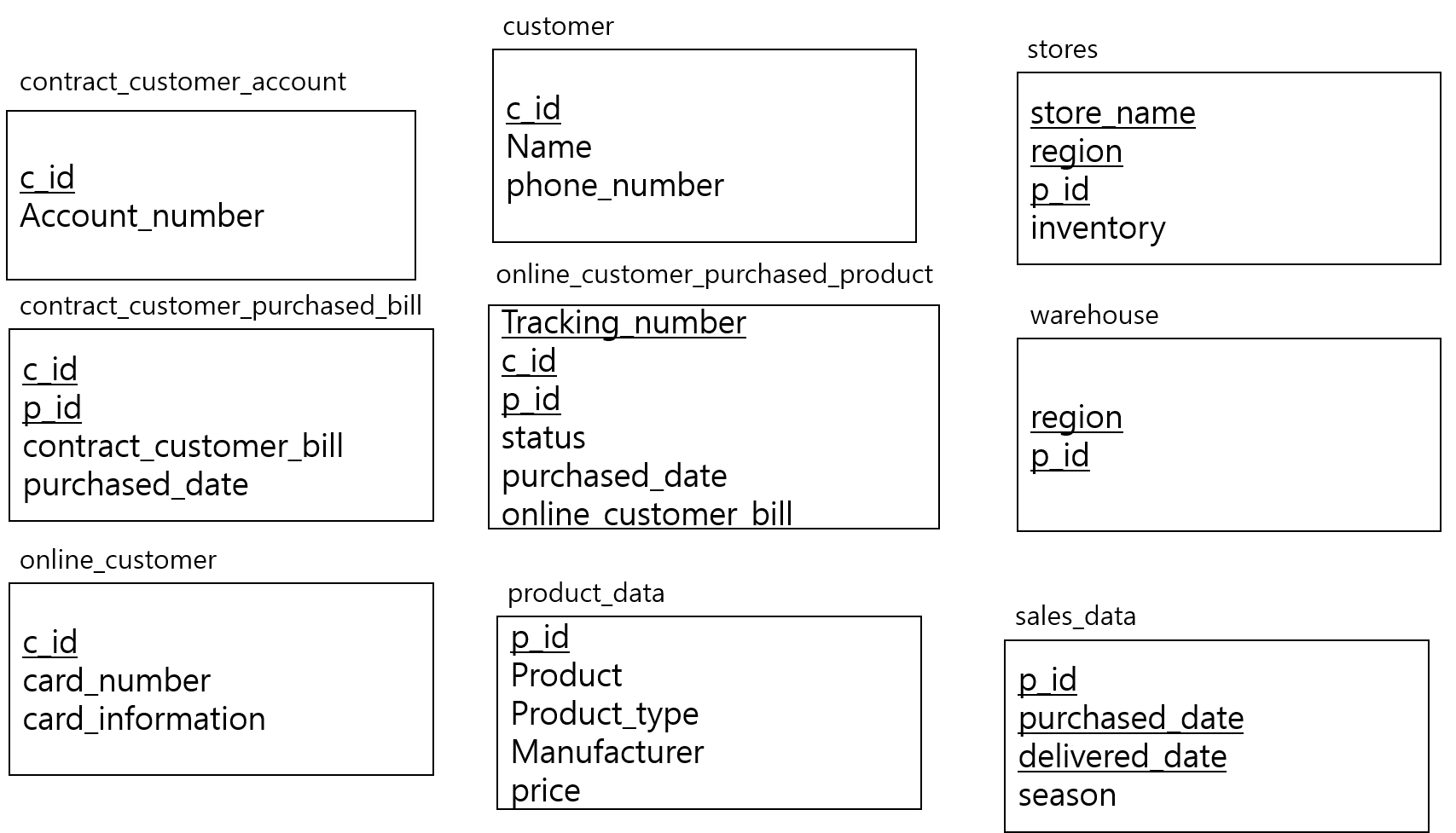


Figure 로지컬테이블

BCNF를 실행하기 위해 먼저 모든 튜플을 하나의 테이블에 있다고 가정하여 Figure 1을 만들었다. (지난 과제에서 잘못된 부분을 수정에 없는 속성을 추가하거나 잘못된 속성을 제거하였다.) 첫번째 테이블 contract\_customer\_account를 만들기 위해 c\_id → Account\_number임을 보일 수 있다. 따라서 contract\_customer\_account 테이블을 생성할 수 있고 이때 이 테이블의 primary key는 c\_id이다. 또한 전체테이블에서 β-α를 빼야한다. 이때 β는 Account\_number, α는 c\_id이므로 전체 테이블에서 Account\_number가 빠진다. 이 과정을 다음과 같은 그림으로 나타낼 수 있다.

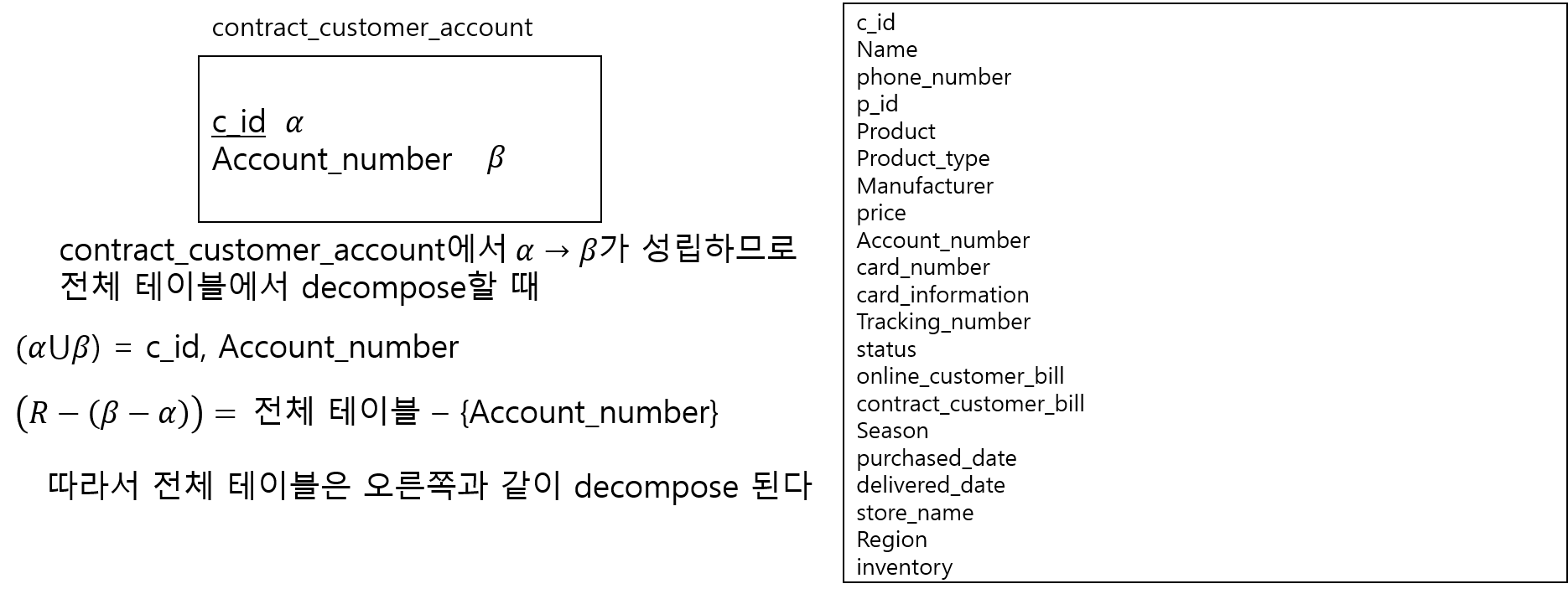


Figure BCNF 예시

마찬가지로 customer와 online\_customer에서도 c\_id가 primary key되고 β값이 각각 customer은 Name,phone\_number online\_customer는 card\_number, card\_information이므로 전체 테이블에서 β값이 빠진다. 이 과정을 반복하면 Figure2와 같은 로지컬 테이블을 만들 수 있다. 각각의 로지컬테이블에서 밑줄 친 primary key는 해당 로지컬테이블을 F라 할 때 F+에서 슈퍼키이므로 BCNF 테스트가 성립한다. 따라서 logical schema diagram은 다음과 같다.

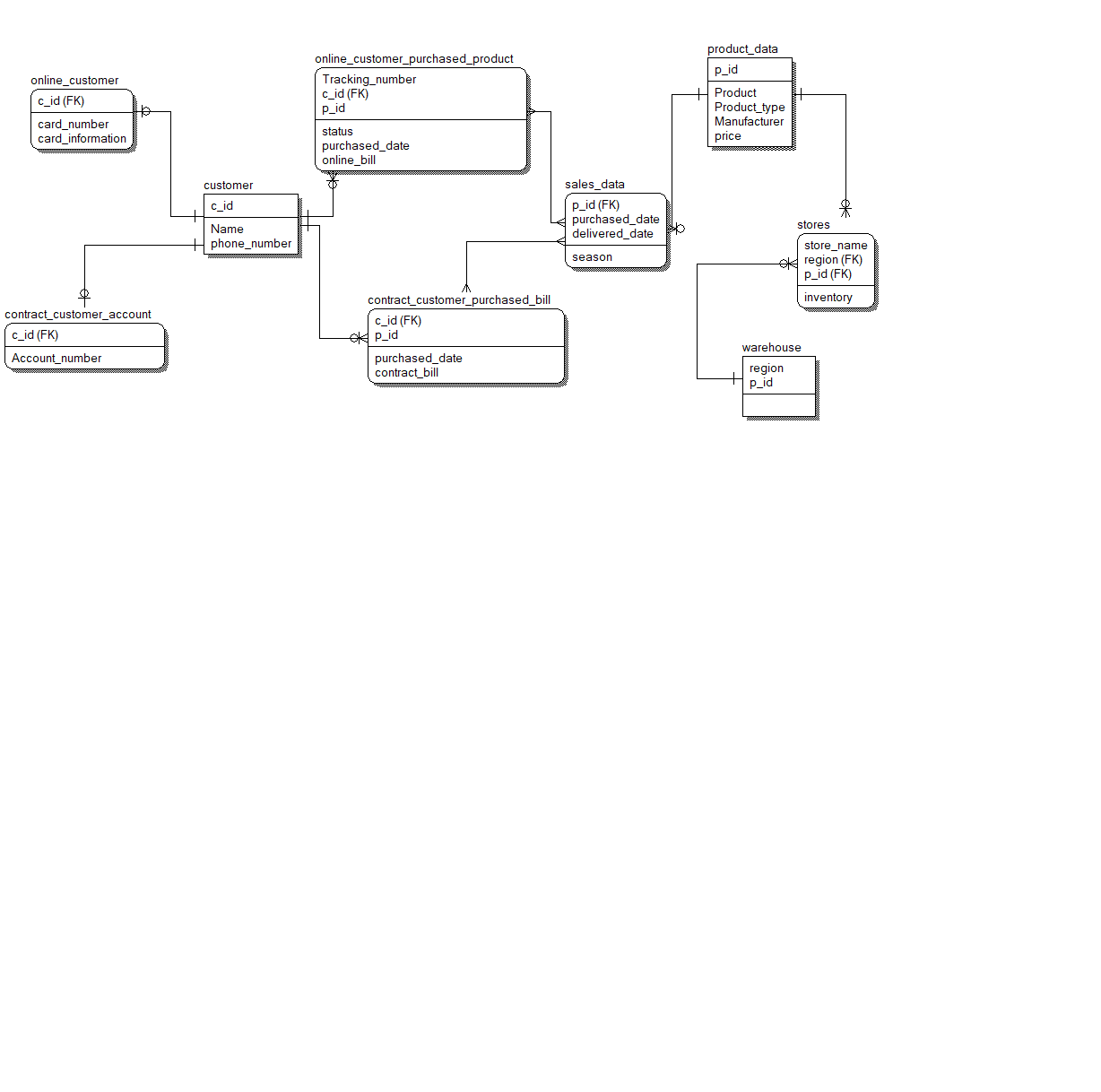


Figure logical schema diagram

1. Physical schema diagram

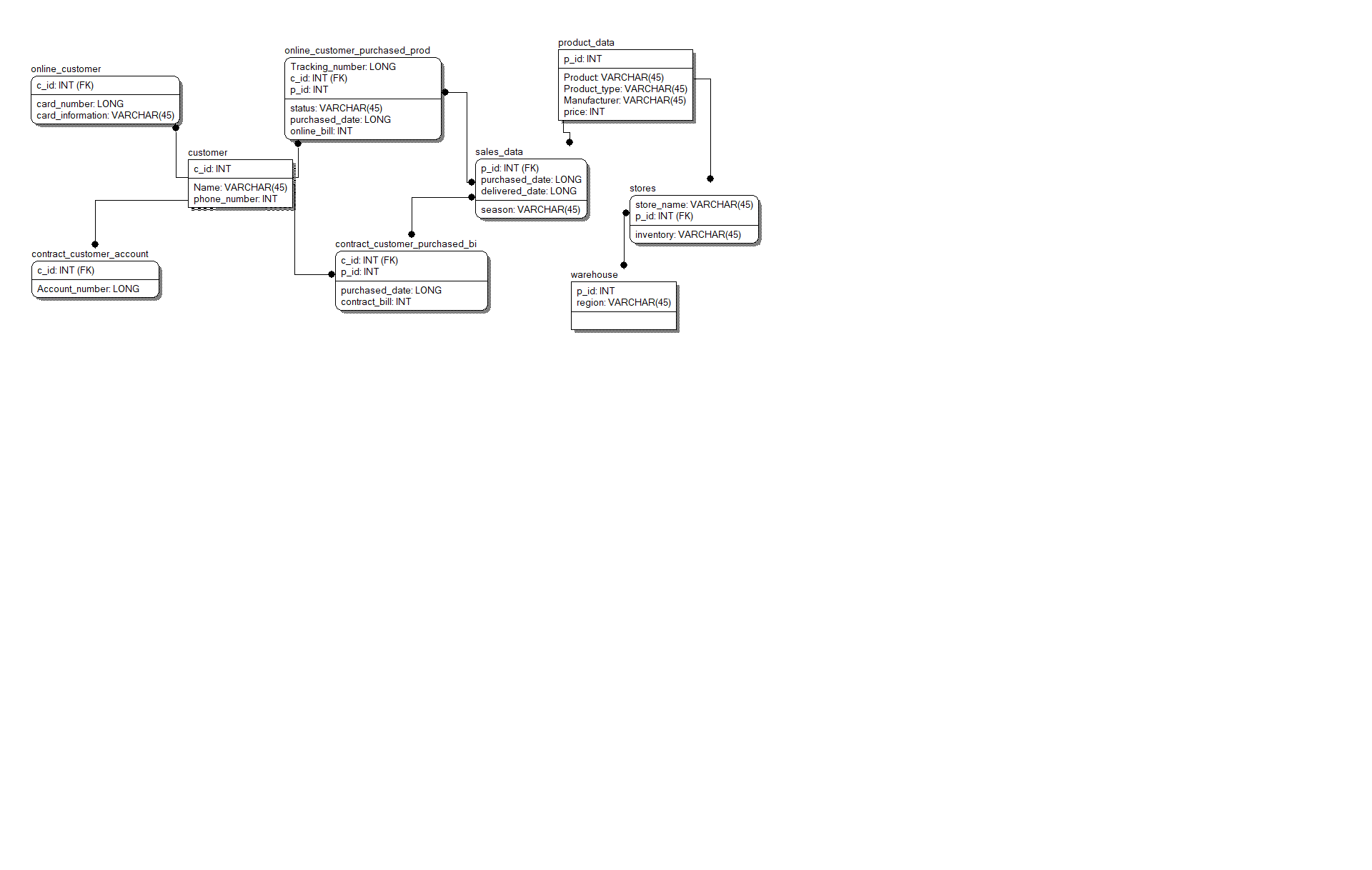


Figure Physical schema diagram

각 스키마와 속성을 정의한다.

p\_id는 제품 번호로 int형으로 정의하였다. 각 제품마다 고유의 제품 번호가 있으므로 이를 통해 제품의 특성을 확인하기 용이하며 따라서 여러 스키마에서 primary key로 사용하였다.

c\_id는 고객마다 갖는 고유의 번호로 int형으로 정의하였다. 이를 통해 해당 고객이 online 고객인지 contract 고객인지 확인할 수 있다. 또한, 각 고객의 카드 정보와 계좌 번호를 특정할 수 있어 마찬가지로 여러 스키마에서 primary key로 사용하였다.

Name은 각 고객의 이름이다. varchar(45) 형태로 선언하였으며 같은 이름을 가진 사람이 있을 수도 있으므로 primary key 역할은 하지 못한다.

Purchased\_date는 제품을 구매한 날로 연도월일 형태로 총 8자리 수이다. Int 형으로 선언해도 상관은 없으나 다른 값들에 비해 단위가 크므로 구분하기 위해 long형으로 선언하였다. 구매 날짜는 동일한 날이 존재할 수 있으므로 그 자체로는 primary key가 될 수 없지만 임의의 상품이 구매된 날과 다음에 설명할 도착하는 날을 묶어서 sales\_data에서 primary key 역할을 수행하도록 하였다.

Delivered\_date는 제품이 도착한 날로 마찬가지로 연도월일 형태로 하였으며 long형이다. 제품이 도착하는 날은 제품을 구매한 날로부터 7일 이내를 제 시간에 배달된 것으로 간주하였다. 만약 제품이 중간에 파손되어 다시 배송한 경우 구매한 날은 그대로 하되 도착일을 구매일로부터 21일 지난 것으로 간주한다.

Product는 제품 이름이고 varchar(45)로 선언하였다. Prodcut\_type은 해당 제품의 타입이고 마찬가지로 varchar(45)로 선언하였다. Manufacturer는 해당 제품의 제조사이고 varchar(45)로 선언하였다. region은 지역이고 varchar(45)로 선언하였다. Store\_name은 각 지역에 있는 매장 이름이고 varchar(45)로 선언하였다. Inventory는 매장이 가지고 있는 각 제품의 재고량이고 int로 선언하였다. 단위는 퍼센트이고 0,25,50,75,100 중에 하나의 값을 갖는다. Price는 제품의 가격이고 int형으로 선언하였다. 단위는 달러이다.

Card\_number는 각 온라인 고객이 등록한 카드번호이다. 카드번호는 (카드사 4자리)(임의의3자리)(고객고유번호 5자리)로 구성되어 있다. 총 12자리이므로 자리수가 커 long형으로 정의하였다. Card\_information은 각 카드의 카드사를 의미하고 varchar(45) 형태로 저장하였다.

Account\_number는 contract 고객이 등록한 계좌번호이다. 계좌번호는 (은행 4자리)(고객고유번호 5자리)로 구성되어 있으며 마찬가지로 long형으로 정의하였다.

Phone\_number는 각 고객의 전화번호이며 (국가번호 1자리)(지역번호 3자리)(임의의7자리)로 구성되어있다. 마찬가지로 long형으로 정의하였다.

Contract\_bill과 online\_bill은 각 고객이 구매한 제품의 영수증이다. int형으로 선언하였으며 이 값은 구매한 제품의 가격의 합이므로 달러 단위이다.

Tracking\_number는 온라인으로 주문한 제품에 붙는 배송번호이다. (고객번호5자리)(구매한 제품 번호 2자리)(임의의 2자리)로 이루어져 있으며 long형으로 선언하였다. 만약 제품이 중간에 배송되지 못하였다면 Tracking\_number를 통해 상황을 보고 받을 수 있다.

Status는 제품의 상태를 의미한다. 만약 제품이 성공적으로 전송된다면 O를 갖고 실패한다면 X를 갖는다. 따라서 varchar(45)형으로 선언하였다.

Season은 해당 제품을 구매한 계절이다. 총 4계절이 있고 varchar(45)형으로 선언하였다.

다음은 각 테이블에 대한 구체적인 정보이다.

contract\_customer\_account는 c\_id 5자리와 Account\_number 9자리로 이루어져 있다.

contract\_customer\_purchased\_bill은 c\_id 5자리, p\_id 1자리 혹은 2자리, bill 3자리 혹은 4자리, purchased\_date 8자리로 이루어져 있다.

Customer는 c\_id 5자리, 임의의 영어권 이름, phone\_number 11자리로 이루어져 있다.

online\_customer는 c\_id 5자리, card\_number 12자리, card\_information로 이루어져 있다. 카드 정보는 카드 회사를 의미하고 VISA, MASTER\_CARD, DISCOVER 중 하나이다.

online\_customer\_purchased\_product는 Tracking\_number 9자리, c\_id 5자리, p\_id 1자리 혹은 2자리, status O 혹은 X, purchased\_date 8자리, bill 3자리 혹은 4자리로 이루어져 있다.

product\_data는 p\_id 1자리 혹은 2자리, Product, Product\_type, Manufacturer, price 3자리 혹은 4자리로 이루어져 있다. Product는 Iphone 일부 기종, Galaxy S 일부 기종, Ipad 일부 기종, Galaxy Tab 일부 기종, Apple Watch, Galaxy Watch로 구성되어 있다. Product\_type은 phone, pad, watch이다. Manufacturer는 Apple과 Samsung이다. Price는 동일하게 출고가 기준이며 단위는 달러이다.

sales\_data는 p\_id 1자리 혹은 2자리, season 4가지, purchased\_date 8자리, delivered\_date 8자리이다. delivered\_date는 배송일로부터 7일 후이다. 만약 이보다 큰 delivered\_date가 있다면 이는 배송이 늦은 것으로 생각한다. 배송 중 제품이 파손된 경우 재전송이 확정될 때까지 delivered\_date를 0으로 바꾼다. 만약 재전송이 확인되었으면 delivered\_date는 purchased\_date+21일로 업데이트한다.

Stores는 store\_name 두가지, region 두가지, p\_id 1자리 혹은 2자리, inventory로 이루어져 있다. Store\_name은 Applestore, Samsungstore 중 하나이다. Region는 California, NewYork 중 하나이다. Inventory는 0, 25, 50, 75, 100 중 하나이며 단위는 퍼센트이다.

Warehouse는 지역에 있는 창고를 의미하며 region, p\_id 1자리 혹은 2자리로 이루어져 있다. 만약 Stores의 Inventory가 0이면 해당 warehouse를 찾은 다음 Inventory를 100으로 채운다.

1. Queries



Figure TYPE 1

타입 1은 위와 같다. 파손된 상품을 찾기 위해 Tracking\_number 100020205를 입력하였다. 파손된 상품의 고객번호는 10002이고 고객 이름은 robert이다. Robert의 전화번호는 12093141592이다. Type1은 서브타입을 갖기 때문에 아래에 서브타입을 실행할 것인지 나갈 것인지 확인한다.

출력되지 않았지만 입력받은 Tracking\_number에 해당하는 status를 X로, delivered\_date를 0으로 바꾼다. 이는 제품을 다시 배송할 때까지 유지된다.

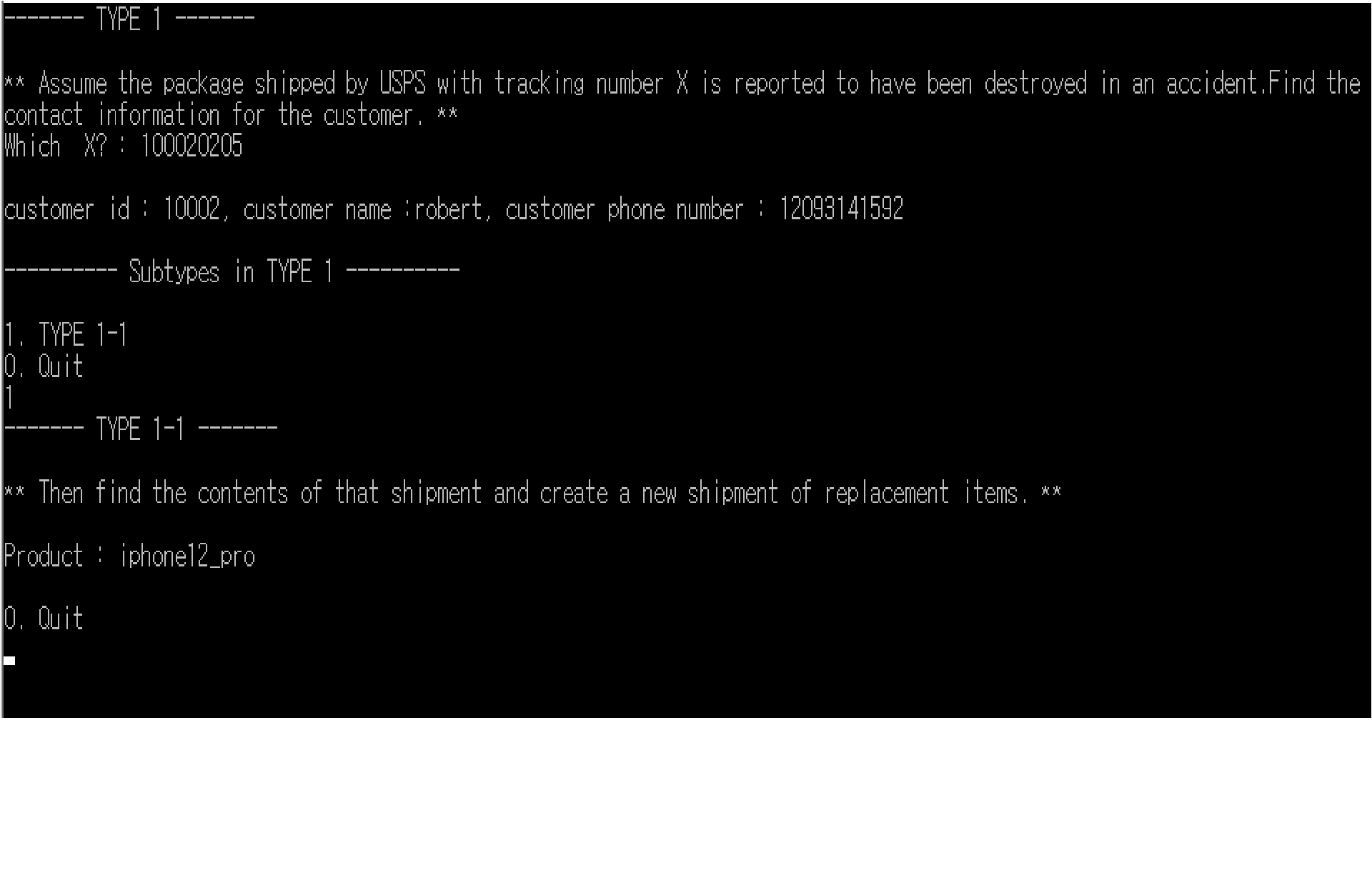


Figure TYPE 1-1

Type 1-1을 실행한 결과이다. 파손된 상품은 iphone12 pro이다. 코드에서 나타나지 않지만 Type 1-1을 실행함으로써 새 제품이 배송되었다. 따라서 status를 O로, delivered\_date를 purchased\_date+21일로 바꾼다. 더 이상 서브타입이 없으므로 0을 눌러 나간다.

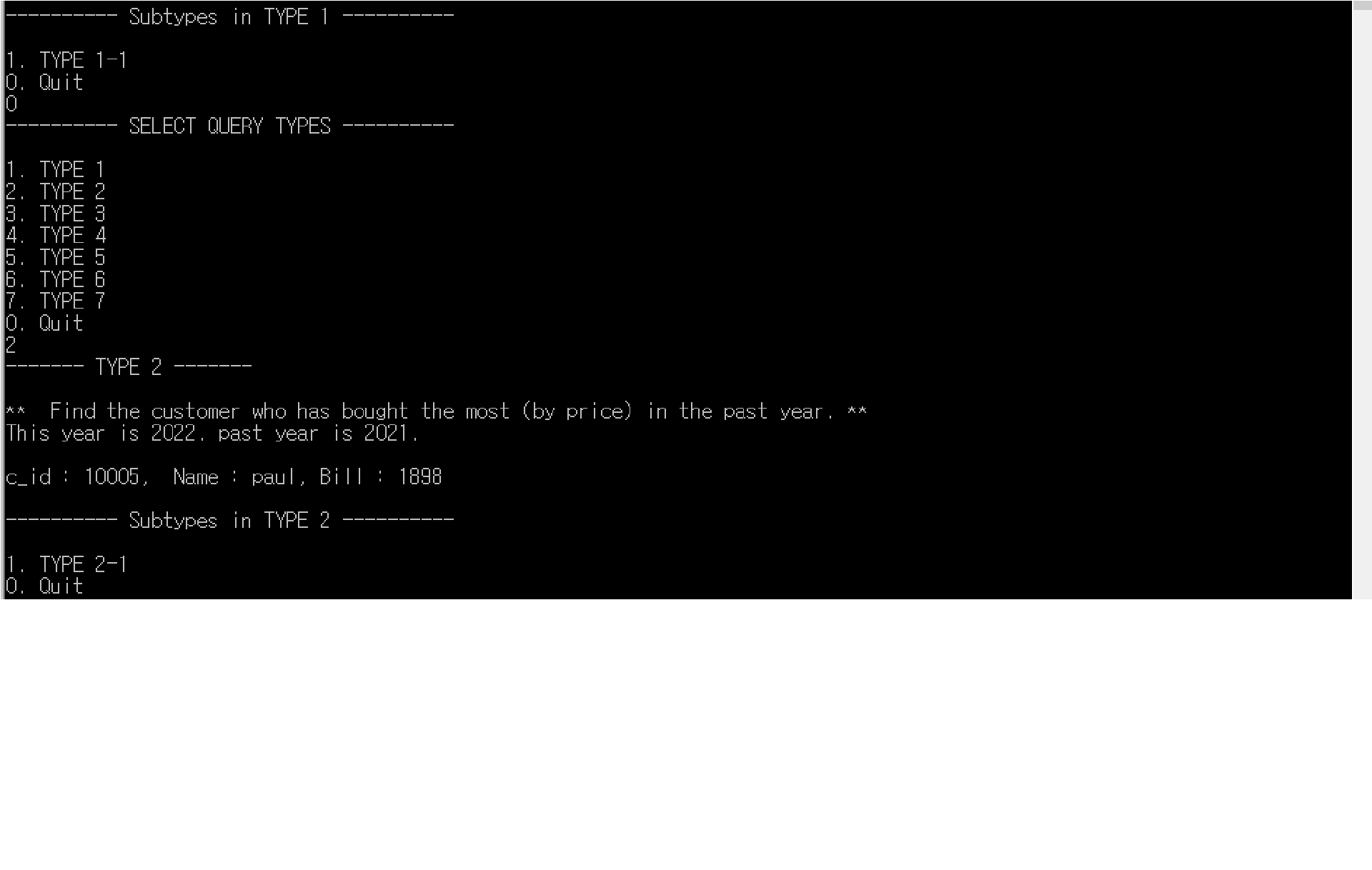


Figure TYPE 2

Type 2는 작년동안 가장 구매액이 큰 사람을 찾는 코드이다. 기준을 2022년으로 하였으며 따라서 purchased\_date가 2021년인 사람 중에서 구매액이 가장 큰 사람을 찾으면 된다. 고객번호는 10005이고 이름은 paul이며 총 1898 달러를 사용하였다. 서브타입이 있으므로 아래에 메뉴 중에 선택하면 된다.

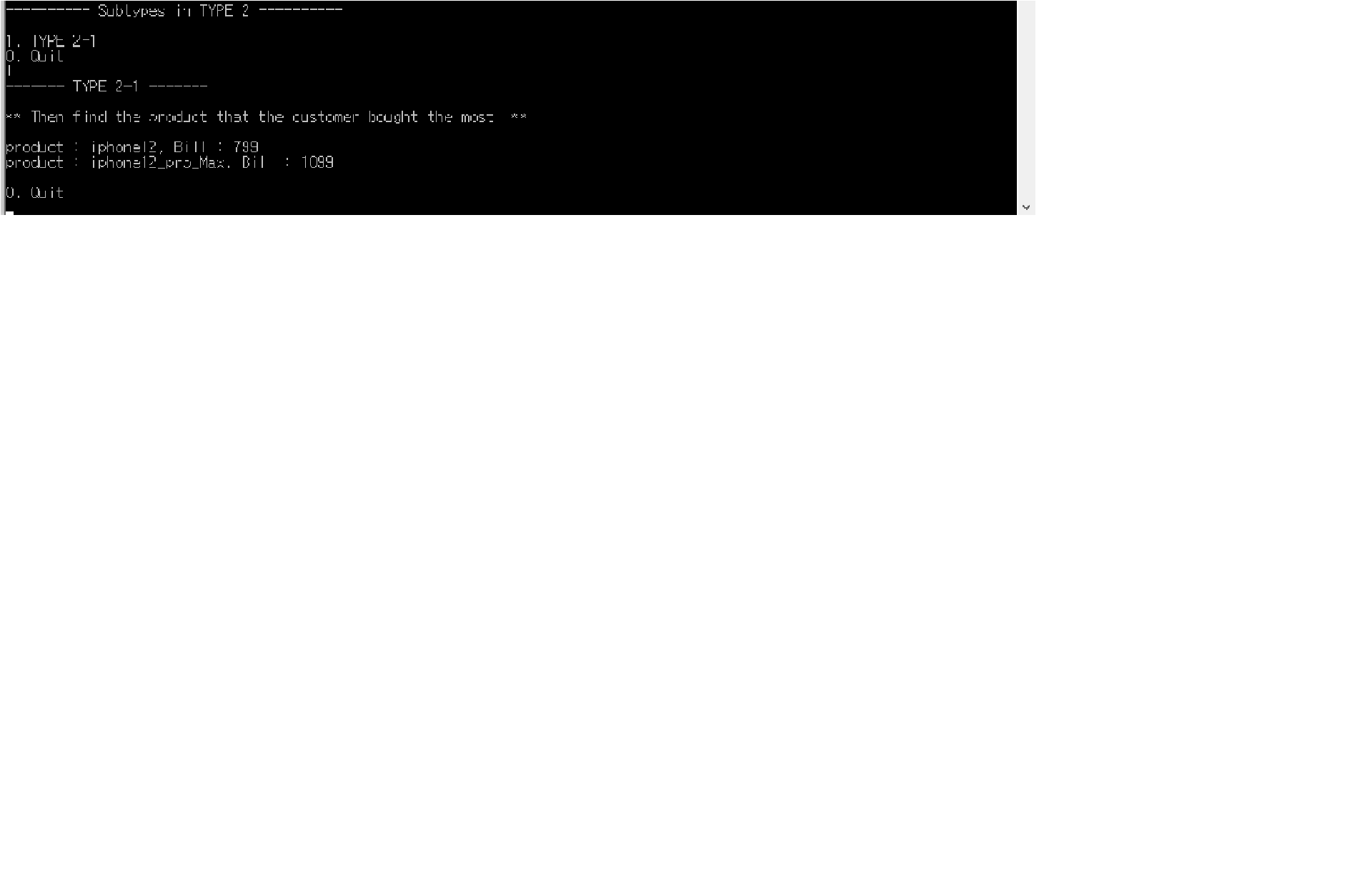


Figure TYPE 2-1

Type 2-1은 Type 2에서 구한 고객이 구매한 제품을 출력하는 코드이다. Paul이 구매한 제품은 iphone12, iphon12 pro Max이고 각각 799,1099 달러이다. 서브타입이 없으므로 종료한다.

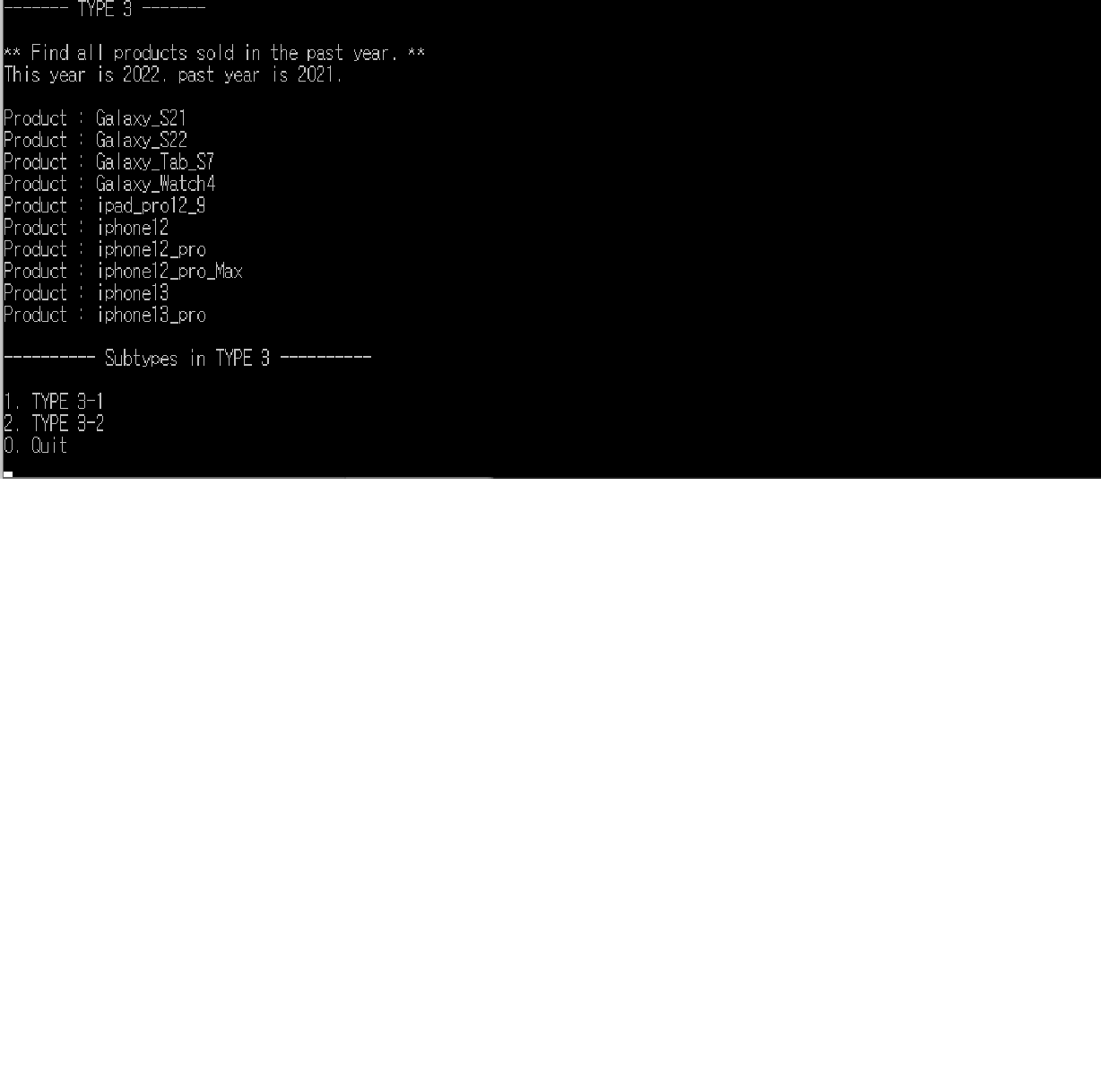


Figure TYPE 3

Type 3은 작년에 판매된 모든 상품을 출력하는 것이다. 마찬가지로 기준은 2022년이고 작년은 2021년이다. 판매된 상품은 위와 같다. 서브타입이 존재하므로 3-1, 3-2 중 선택할 수 있다.

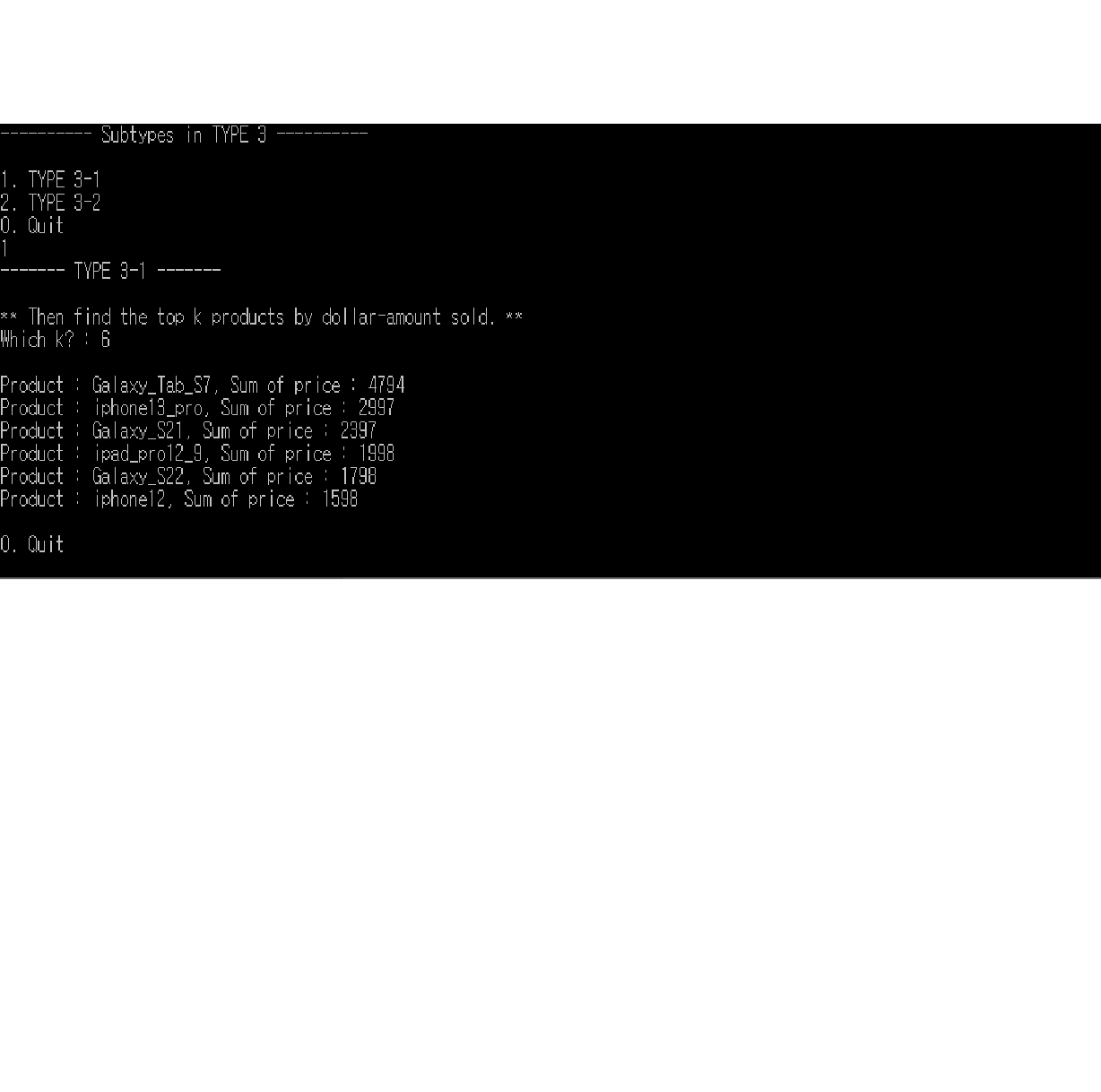


Figure TYPE 3-1

Type 3-1은 판매된 상품 중 k개 만큼을 선택해서 총 판매액을 출력하는 코드이다. K를 6으로 입력하였고 위와 같이 출력 되었다.

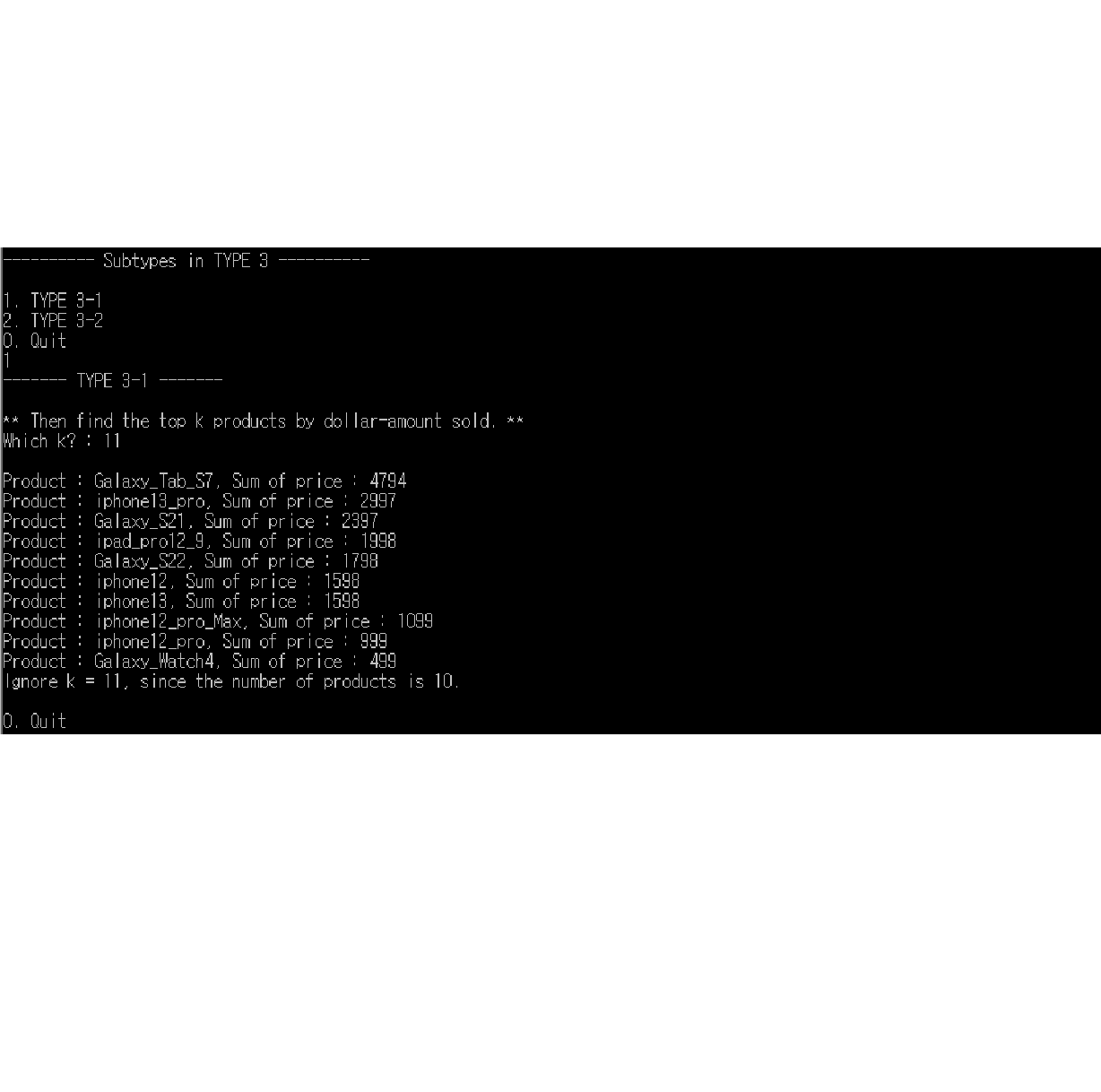


Figure TYPE 3-1 when k is over

이 테이블에서 product의 총 개수는 10이다. 만약 k가 10보다 큰 수가 들어올 경우 k 값은 무시되고 10개의 product가 출력된다.

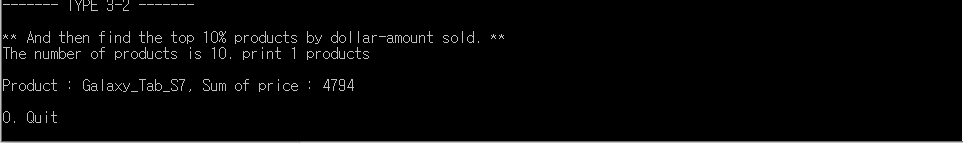


Figure TYPE 3-2

Type 3-2는 판매액이 가장 큰 10%에 해당하는 product를 출력하는 코드이다. 이때 product가 10개이므로 10%는 가장 큰 1개이다. 소수점은 무시하므로 만약 product가 19개여도 상위 10%는 1개이다.

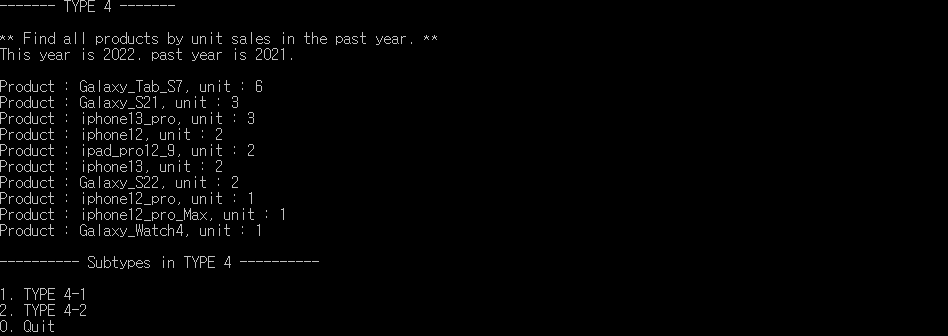


Figure TYPE 4

Type 4는 3과 유사하게 작년의 판매된 제품 중 가장 많이 판매된 제품을 출력하는 코드이다. 마찬가지로 기준은 2022년이고 작년은 2021년이다. 판매된 제품과 판매량은 위와 같다.

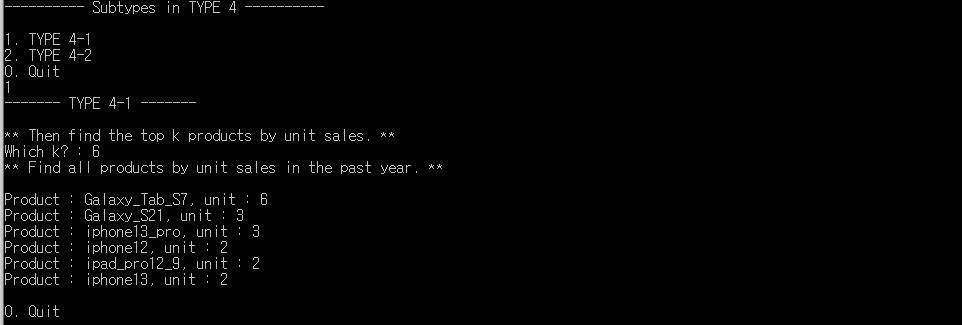


Figure TYPE 4-1

Type 4-1은 그 중에서 가장 많이 팔린 k개를 출력하는 것이다. K를 6으로 입력하였을 때 가장 많이 팔린 6개를 출력한다. Type 3-1과 마찬가지로 k가 전체 product 개수를 넘어가면 k를 무시하고 모든 product와 유닛을 출력한다.

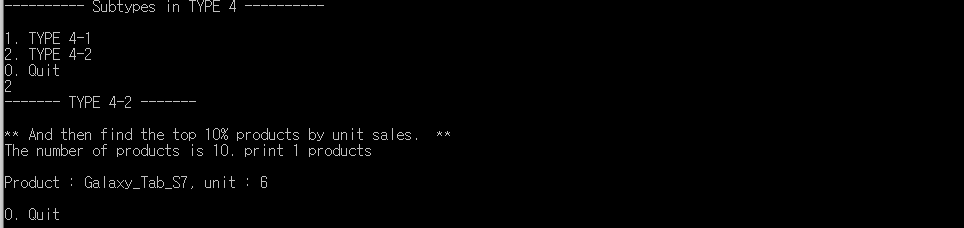


Figure TYPE 4-2

Type 4-2는 Type 3-2와 마찬가지로 상위 10%만큼의 product를 출력한다. Product의 개수가 10개이므로 상위 10%는 1개이다. 따라서 Galaxy Tab S7이 6개만큼 판매된 것을 알 수 있다.



Figure TYPE 5

Type 5는 캘리포니아 매장에서 재고가 떨어진 제품을 출력하는 코드이다. 출력한 결과 Applestore에서 iphone13 pro Max, ipad air 4 재고가 떨어졌고 Samsungstore에서는 Galaxy S22, Galaxy Tab S8, Galaxy watch 4 재고가 떨어졌다. 코드 실행과 동시에 모든 재고는 창고에서 제품을 다시 받아 재고가 100%가 된다. 따라서 Type 5를 끝내고 다시 Type 5를 실행하면 아무것도 출력되지 않는다.



Figure TYPE 6

Type 6은 제 시간에 배송되지 않는 제품을 찾는 코드이다. 제품 배송은 7일 이내를 기준으로 하였으며 구매일과 배송일의 차이가 7보다 크다면 제 시간에 배송되지 않는 것으로 볼 수 있다. 배송이 늦은 제품과 그 날짜는 위와 같다.

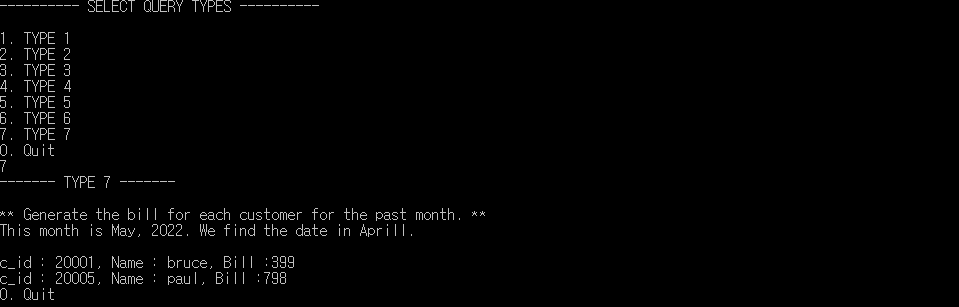


Figure TYPE 7

Type 7은 지난달에 제품을 구매한 고객에게 영수증을 출력하는 코드이다. 기준은 2022년 5월로 하였으며 따라서 지난달은 2022년 4월이다. 출력 결과 고객번호 20001 bruce가 399달러, 고객번호 20005 paul이 798달러를 지불하였다.

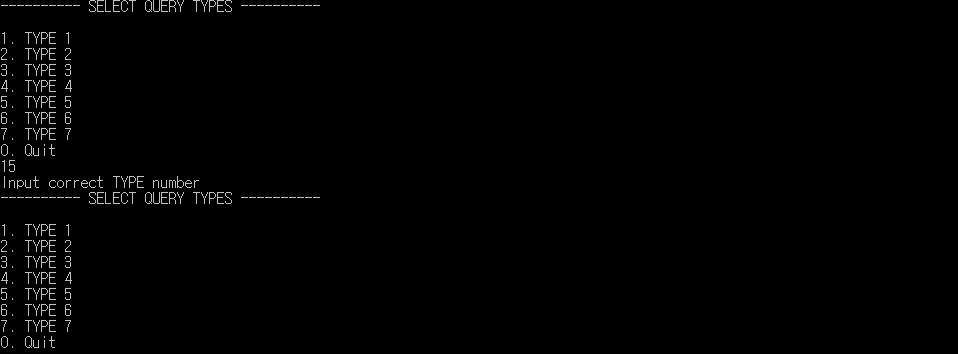


Figure wrong number

모든 메뉴에서 존재하지 않는 번호를 입력할 경우 다시 메뉴가 출력된다. 모든 메뉴는 0을 누르면 상위 메뉴로 돌아간다. 전체 메뉴에서 0을 누르면 실행이 종료된다.