

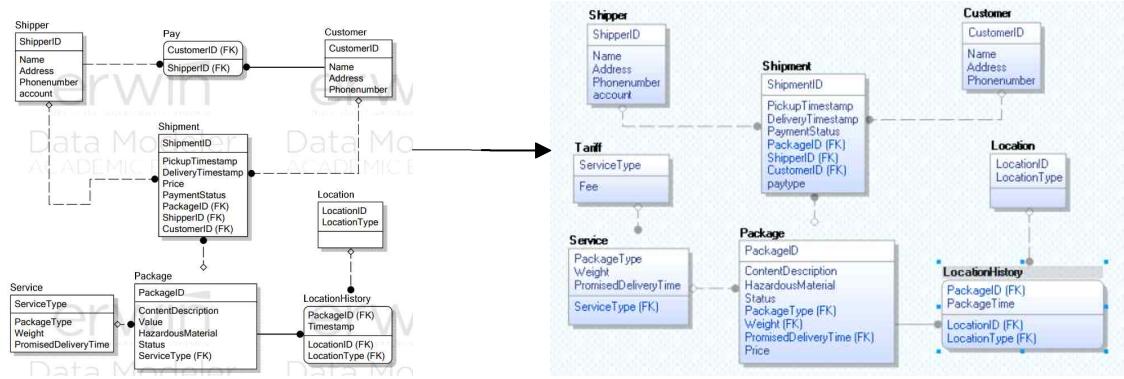
데이터베이스시스템

정성원 교수님
02반
20192136
컴퓨터공학과
이윤기

1. 프로젝트 개요

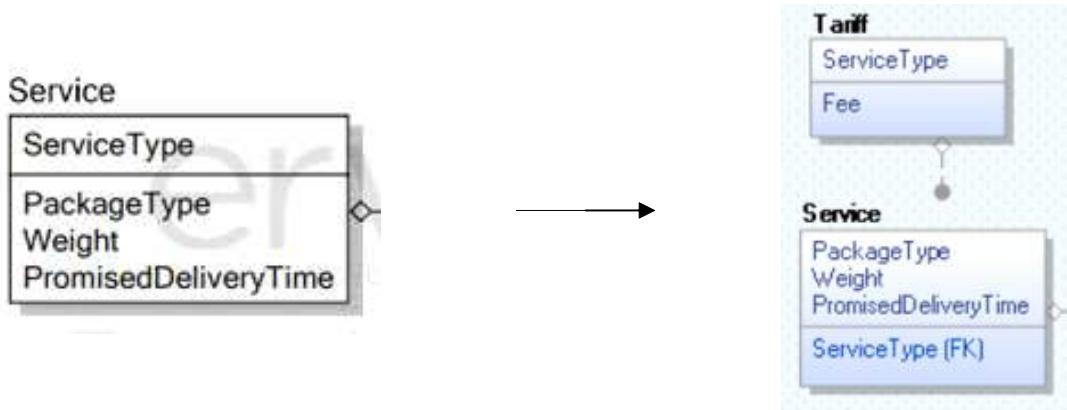
- **프로젝트 목표 :** project1에서 작성한 logical schema를 기반으로 physical schema를 생성, BCNF 개념을 이용하여 정보를 BCNF 형태로 변형, 마지막으로 MySQL을 이용하여 실제 데이터베이스를 구축하여 쿼리 비문을 수행하는 것이다.

2. 변경사항

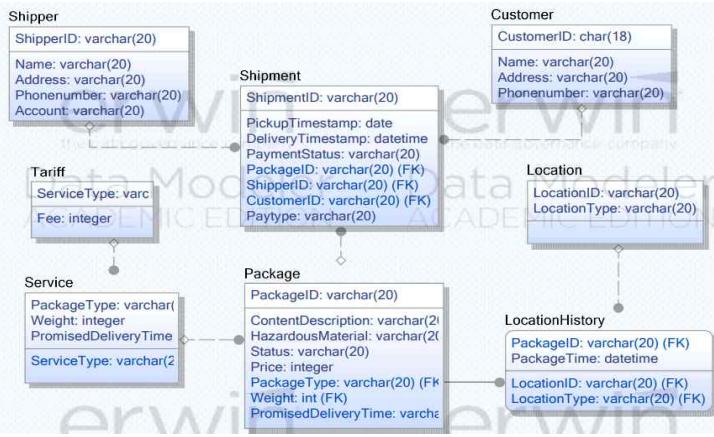


기존의 project1에서 제출한 relation schem의 경우 bcnf는 전부 만족하지만 쿼리를 구현하는데 있어 필요한 부분은 수정했다.

1. Pay의 경우 Shipper와 Customer간의 연결이기에 Shipment안에서 해결할 수 있어 삭제.
- 2. Service에서 Service type을 통해 PackageType, Weight, PromisedDeliveryTime을 전부 알 수 있도록 하였으나, 같은 type안에서도 다른 유형의 변경을 위하고, Serivcetype에 따른 가격을 추가하였더니, ServiceType-> Fee라는 dependency가 생성되어 Service를 BCNF를 위해 Service와 Tariff로 분할했다.



3. Physical Schema



4. Table Description

1. **Shipper**: 물건을 보내는 사람, sender

Primary key : CustomerDvarchar(20)

Attribute

ShipperIDDvarchar(20)

Name varchar(20)

Address varchar(20)

Phonumbervarchar(20)

Account varchar(20)

2. **Customer**: 물건을 받는 사람, recipient

Primary key : ShipperIDDvarchar(20)

Attribute

CustomerIDDvarchar(20) primary key

Name varchar(20)

Address varchar(20)

Phonumbervarchar(20)

3. **Location** : 물건을 받는 사람, recipient, 택배가 이동할 장소로서 번호와 장소 유형을 통해
식별

Primary key : LocationID, LocationType

Attribute

LocationIDvarchar(20)

LocationTypevarchar(20)

4. **Tariff**: 택배회사의 이용요금표, 서비스 유형에 따라 가격을 책정

Primary key : ShipperIDvarchar(20)

요금표:

Attribute

ServiceTypevarchar(20) primary key

A : 5000

Fee integer

B : 7000

C : 8000

D : 9000

E : 10000

5. **Service** : 무게, 배달예정시간, 포장유형에 따른 서비스 유형

Primary key : PackageType, Weight, PromisedDeliveryTime

Attribute

PackageTypevarchar(20)

서비스 유형

Type A : larger

Weight integer

Type B : flat envelop 2, small box 10 second day

PromisedDeliveryTimevarchar(20)

Type C : flat envelope 2overnight

ServiceTypevarchar(20)

Type D : small box 10 overnight

Type E : larger box 20 second day

6. **Package**: 보내는 물품, Status는 해외배송 여부,

Primary key : PackageID

Attribute

PackageIDvarchar(20) primary key

ContentDescriptionvarchar(20)

Price integer

HazardousMaterialvarchar(20)

Status varchar(20)

PackageTypevarchar(20)

Weight integer

PromisedDeliveryTimevarchar(20)

7. Shipment : 물품 발송,

Primary key : ShipmentID

Attribute

ShipmentIDvarchar(20) primary key

PickupTimestampdate

DeliveryTimestampdatetime

PaymentStatusvarchar(20)

PackageIDvarchar(20)

ShipperIDvarchar(20)

CustomerIDvarchar(20)

8. LocationHistory: 시간에 따른 물품의 위치, 물품의 위치가 바뀔 때 기록 ex) Truck 1234에서 Warehouse 1234로 옮겼을 시간마다.

Primary key : ShipmentID

Attribute

PackageIDvarchar(20)

PackageTimedatetime

LocationIDvarchar(20)

LocationTypevarchar(20)

primary key (PackageID, PackageTime)

5. ODBC

Connection을 위해 변수를 설정한다. 코드에서 host,user,pw,db를 맞게 설정하여 프로그램을 실행

다음으로 buffer에 담긴 내용을 ; 단위로 끊어서 init에 저장하여 Mysql_query를 이용해서 서버에 실행. Buffer에서 drop table까지 실행후 break. While문 실행시 table과 tuple이 생성

Type별 sql

Type I

1. connection Succeeded
---- SELECT QUERY TYPES -----
1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Select Number: 1
---- TYPE 1 ----
Truck 1721 is destroyed in a crash
---- Subtypes in Type 1----
1. TYPE I-1
2. TYPE I-2
3. TYPE I-3
Select Number(among 1, 2, 3): 1
---- TYPE I-1 ----
The Truck number 1721 is crashed
Find all customers who had a package on the truck at the time of the crash.
CustomerID : 4322

Type 1을 입력 시 서브타입을 물어보게 되고 파괴된 트럭을 1721번이라고 지정하여 출력했다. TypeI-1을 선택하는 경우 1721트럭에 있던 package의 발송자의 id가 출력되도록 해주었다.

typeI-1을 수행하기 위한 query문은 아래와 같이 작성하였다.

```
select CustomerID from Shipment S natural join locationhistory L where LocationID = "1721" and S.DeliveryTimestamp is null;
```

파괴가 된 트럭에 있던 소포의 경우 트럭이 파괴되어 물품이 도착하지 못하였기 때문에 다음과 같이 작성하였다.

2. SELECT QUERY TYPES -----
1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Select Number: 1
---- TYPE 1 ----
Truck 1721 is destroyed in a crash
---- Subtypes in Type 1----
1. TYPE I-1
2. TYPE I-2
3. TYPE I-3
Select Number(among 1, 2, 3): 2
---- TYPE I-2 ----
The Truck number 1721 is crashed
Find all recipients who had a package on that truck at the time of the crash.RecipientsID : 1237

TypeI-2을 선택하는 경우 1721트럭에 있던 package의 수신자의 id가 출력되도록 해주었다.
typeI-2을 수행하기 위한 query문은 아래와 같이 작성하였다.

```
select ShipperID from Shipment S natural join locationhistory L where LocationID = "1721" and S.DeliveryTimestamp is null;
```

typeI-1과 마찬가지로 작성해주었다.

```
3
----- SELECT QUERY TYPES -----
1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Select Number: 1
----- TYPE 1 -----
Truck 1721 is destroyed in a crash
----- Subtypes in Type I -----
1. TYPE I-1
2. TYPE I-2
3. TYPE I-3
Select Number(among 1, 2, 3): 3
----- TYPE I-3 -----
The Truck number 1721 is crashed
Find the last successful delivery by that truck prior to the crash.RecipientsID : 1237, Recipients name Luke Shaw, Recipients address : TE902
```

TypeI-3을 선택하는 경우 1721트럭에 있던 사고 전 가장 최근 완료된 수신자의 정보를 입력해주었다.

typeI-3을 수행하기 위한 query문은 아래와 같이 작성하였다.

```
select C.CustomerID,C.name,C.address from Customer C natural join Shipment S where S.DeliveryTimestamp = (select max(DeliveryTimestamp) from Shipment S natural join locationhistory L where LocationID = "1721");
```

가장 최근 성공적인 수신자라 하면, locationid가 1721인 택배로 전송된 배달 완료시점이 가장 최근인 것이기 때문에 Shipment table의 DeliveryTimestamp가 가장 큰 수신자 정보를 구하기 위해 작성하였다.

Type II

```
----- SELECT QUERY TYPES -----
1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Select Number: 2
----- TYPE II -----
Find the customer who has shipped the most packages in the past year.
Which year? : 2021
CustomerID : 1235, count : 3
```

TypeII를 선택할 경우 인자로 받은 해의 가장 많이 배송을 한 송신자의 id와 횟수가 나오도록 해주었다.

typeII를 수행하기 위한 query문은 아래와 같이 작성하였다.

```
with VIP(ShipperID, cnt) as (select ShipperID, count(PackageID) as cnt from ((Package
natural join Service) natural join Tariff) natural join Shipment S where
year(S.DeliveryTimestamp) = "%d" group by ShipperID) select ShipperID, cnt from VIP
V where V.cnt >= all(select cnt from VIP);
```

쿼리문에서 특정 연도를 지정하기 위해 ""를 사용하기 위해 %"를 사용하여 작성하였다.

Type III

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

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

Select Number: 3

---- TYPE III ----

Find the customer who has spent the most money on shipping in the past year.

Which year? : 2021

CustomerID : 1235, Total : 23000

TypeIII를 선택할 경우 인자로 받은 해의 가장 많은 배송비를 지출한 송신자의 id와 횟수가 나오도록 해주었다.

typeIII를 수행하기 위한 query문은 아래와 같이 작성하였다.

```
with VIP(ShipperID, total) as (select ShipperID, sum(Fee) as total from ((Package
natural join Service) natural join Tariff) natural join Shipment S where year(S.DeliveryTimestamp)
= "%d" group by ShipperID) select ShipperID, total from VIP V where V.total >=
all(select total from VIP);
```

쿼리문에서 특정 연도를 지정하기 위해 ""를 사용하기 위해 %"를 사용하여 작성하였다.

TypeII와 유사하지만 ServiceType에 따른 배송비가 다르기 때문에 많은 sum을 사용해서 배송비의 합이 가장 큰 개체를 구하는 방식으로 작성하였다.

TypeIV

```
----- SELECT QUERY TYPES -----
 1. TYPE I
 2. TYPE II
 3. TYPE III
 4. TYPE IV
 5. TYPE V
 0. QUIT
-----
Select Number: 4

---- TYPE IV ----
Find the packages that were not delivered within the promised time.
CustomerID : 122199, PackageID : 8232757
CustomerID : 261199, PackageID : 3818726
CustomerID : 654823, PackageID : 2912634
```

TypeIV를 선택할 경우 약속된 배송 예정일 보다 늦게 도착한 소포를 구했다.

typeIV를 수행하기 위한 query문은 아래와 같이 작성하였다.

```
select ShipmentID ,PackageID from Shipment natural join package where
((PromisedDeliveryTime = "overnight" and (day(PickupTimestamp) + 1 <
day(DeliveryTimestamp))) or (PromisedDeliveryTime = "second day" and
(day(PickupTimestamp) + 2 < day(DeliveryTimestamp))) or (PromisedDeliveryTime =
"larger" and (day(PickupTimestamp) + 3 < day(DeliveryTimestamp))));
```

ServiceType에 따른 배송예정일이 다르다 overnight은 하루 secondday는 2일 larger은 3일로 설정하였다. PickupTimestamp에 각각 하루, 이틀, 삼일보다 더한 값보다 DeliveryTimestamp가 큰 경우 더 늦게 도착한거기 때문에 위와같이 작성하였다.

Type V

```
----- SELECT QUERY TYPES -----
 1. TYPE I
 2. TYPE II
 3. TYPE III
 4. TYPE IV
 5. TYPE V
 0. QUIT
----- Select Number: 5
----- TYPE 5 -----
Generate the bill for each customer for the past month. Consider creating several types of bills.
Which year month? : 2021 3
-----Simple Bill-----
product_id: 1235CustomerID : 4322 , Customer address : R203, Total Fee : 10000
-----
-----Listing Bill-----
product_id: 1235CustomerID : 1235 , Customer address : A101, ServiceType : C, Total Fee : 8000
CustomerID : 4322 , Customer address : R203, ServiceType : E, Total Fee : 10000
-----
-----Itemizing Bill-----
product_id: 1235CustomerID : 1235 , ShipmentID : 453687, Customer address : A101, Fee : 8000
CustomerID : 4322 , ShipmentID : 623548, Customer address : R203, Fee : 10000
```

TypeV를 선택할 경우 각 타입별로 영수증을 출력하였다. 우선 인자로 연도와 달을 받아서 그해의 고객별로 영수증을 출력하였다.

typeV를 수행하기 위한 query문은 아래와 같이 작성하였다.

simple bill

```
select ShipperID, A.address, sum(Fee) from (((Package natural join Service) natural join Tariff) natural join Shipment S) natural join Shipper A where year(S.DeliveryTimestamp) = "%d" and month(S.DeliveryTimestamp) = "%d" group by ShipperID, A.address;
```

simple bill의 경우 보낸사람의 ID, 주소, 총 금액을 구해주었다.

listing bill

```
select ShipperID, A.address, servicetype, sum(Fee) from (((Package natural join Service) natural join Tariff) natural join Shipment S) natural join Shipper A where year(S.DeliveryTimestamp) = "%d" and month(S.DeliveryTimestamp) = "3" group by ShipperID, servicetype, A.address;
```

listing bill의 경우 simple과 유사하지만 serviceType에 따른 요금표를 작성하였다.

itemizing bill

```
select ShipperID, ShipmentID, A.address, Fee from (((Package natural join Service) natural join Tariff) natural join Shipment S) natural join Shipper A where year(S.DeliveryTimestamp) = "%d" and month(S.DeliveryTimestamp) = "%d";
```

itemizing bill의 경우 각 배송별로 요금을 표시해주었다.

