

<데이터베이스시스템 Project 2>

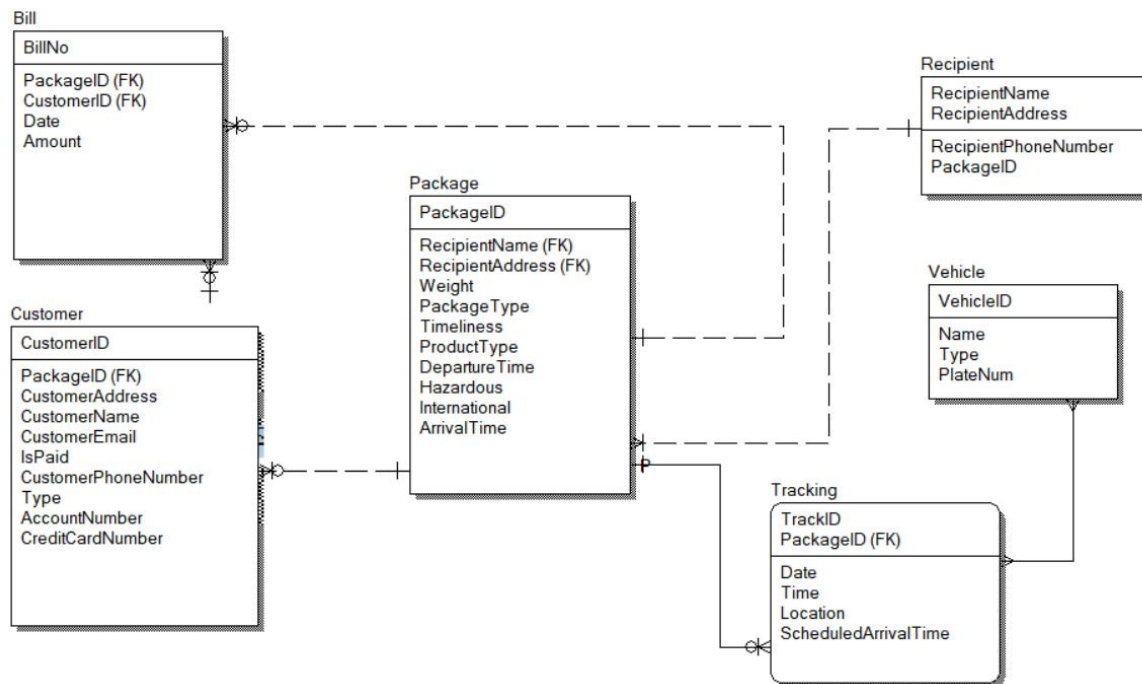
20170795 문채원

제출일자 : 2020/06/29

<목차>

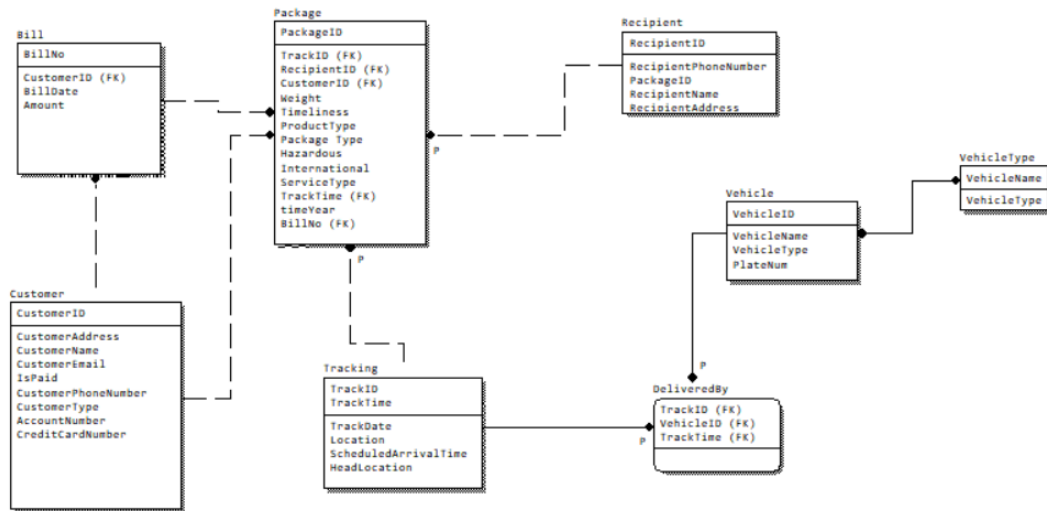
1. Project1의 ER Diagram의 차이
2. BCNF Decomposition
3. Physical Diagram 소개
 - 3-1. Customer Diagram
 - 3-2 Recipient
 - 3-3 Tracking
 - 3-4 Vehicle
 - 3-5 Package
 - 3-6 Bill
4. Create Table Using Visual Studio and ODBC
5. Dropping Table Using Visual Studio and ODBC
6. Query Explanation
 - 6-1 1 번쿼리
 - 6-1-1. 1-1 번 쿼리
 - 6-1-2. 1-2 번 쿼리
 - 6-1-3. 1-3 번 쿼리
 - 6-2 2 번 쿼리
 - 6-3 3 번 쿼리
 - 6-4 4 번 쿼리
 - 6-5 5 번 쿼리
7. 전체 코드

1. Project1의 ER Diagram의 차이



다음은 프로젝트 1에서 만든 ER diagram이다. Project 2를 수행하기 전에 몇가지 추가한 부분이 있다. 먼저, Package와 Bill 사이에 relation을 만들었다. 각각의 package에 대한 bill을 찾아내기 위한 작업이다. Package에서 timeYear이라는, Package가 보내진 year을 저장 항목을 만들어 query문 해결을 쉽게 만들었다. 또, WorkBench에서 Date, Time 과 같은 general한 form으로 key값을 줬을 때 오류가 떠 general한 이름을 가진 key값들에겐 앞에 relation이름을 붙여주었다. 가령, tracking - date이면 trackDate으로 이름을 바꾸었다.

2. BCNF Decomposition



다음 그림은 ER diagram을 BCNF 형태로 쪼갠 결과이다. 먼저, Vehicle에서 Vehicle Name과 Vehicle Type사이에 Functional Dependency가 존재한다. 가령, 쉐보레는 SUV이고 아반떼는 세단이다. 고고로는 오토바이이다. 따라서 Vehicle Name과 Vehicle Type으로 decomposition을 진행했다.

Simplified Test를 진행해보자. Vehicle(VehicleID[PK], VehicleName, VehicleType, PlateNum) relation에 VehicleName->VehicleType으로의 Functional Dependency를 갖는다.

VehicleName+ = (VehicleName, VehicleType)이고 이는 R의 superkey가 아니다. 따라서 이는 BCNF decomposition을 위배한다.

또, Tracking과 Vehicle relation은 many to many relation을 갖기 때문에 두 relation사이의 pk를 담은 relation deliveredBy를 새로 생성했다. 나머지 one to many relation에 대해서는 한쪽 relation에 다른쪽의 primary key를 집어넣는 식으로 decomposition을 수행했다.

3. Physical Diagram 소개

Physical Diagram은 Logical Diagram에서 datatype과 domain을 정하는 형식으로 진행했다. ER-diagram에서 각각의 relation을 클릭해보면 datatype을 확인해볼 수 있으나, 그림을 간결하게 보기 위해 workbench에 설정한 값을 첨부했다.

3-1. Customer Diagram

```

create table IF NOT EXISTS Customer(CustomerID varchar(20),
CustomerAddress varchar(30),
CustomerName varchar(30),
CustomerEmail varchar(40),
IsPaid numeric(1, 0),
CustomerPhoneNumber varchar(20),
CustomerType numeric(1,0),
AccountNumber varchar(20),
CreditCardNumber varchar(30),
primary key (CustomerID));

```

CustomerID는 숫자, 영문자 모두 들어갈 수 있으므로 varchar로 설정했으며 길이는 20으로 설정했다. CustomerAddress는 주소값이므로 조금더 길어질 수 있다는 생각에 30으로 설정했으며, Name도 마찬가지로 이유로 30으로 설정했다. CustomerEmail 역시 varchar, 길이 40으로 설정했다. IsPaid는 미리 지불했는지, 아니면 추가로 지불해야하는지를 나타내는데 이는 0은 지불하지 않은 상태, 1은 지불해야하는 상태를 나타낸다고 가정했다. CustomerPhoneNumber 역시 varchar로 설정했으며, CustomerType도 0과 1로 구분했다. AccountNumber, CreditCardNumber 모두 varchar로 설정했다.

3-2 Recipient

```

create table IF NOT EXISTS Recipient(
RecipientID varchar(20),
RecipientName varchar(20),
RecipientAddress varchar(30),
RecipientPhoneNumber varchar(20),
PackageID varchar(20), primary key(RecipientID)
);

```

RecipientID, Name, Address, PhoneNumber 모두 영문자 및 숫자가 들어갈 수 있으므로 varchar로 설정했다.

3-3 Tracking

```

create table IF NOT EXISTS Tracking(
TrackID varchar(20),
TrackTime varchar(10),
TrackDate varchar(10),
Location varchar(100),
ScheduledArrivalTime numeric(4,0),
HeadLocation varchar(30),
primary key(TrackID, TrackTime));

```

ScheduledArrivalTime을 제외한 모든 값을 sql 작성의 편의를 위해 varchar로 설정했다. ScheduledArrivalTime은 1920와 같은 방법으로 4글자로 들어오게 되는데, 1920은 오후 7시 20분을 나타낸다.

3-4 Vehicle

```
create table IF NOT EXISTS Vehicle(  
VehicleID varchar(20),  
VehicleName varchar(20),  
VehicleType varchar(20),  
PlateNum varchar(20),  
primary key(VehicleID));
```

Vehicle 속성들 역시 모두 varchar로 설정했다. ID, Name, Type, PlateNum 모두 영문자가 들어갈 여지가 있다.

3-5 Package

```
create table IF NOT EXISTS Package(  
PackageID varchar(20),  
TrackID varchar(20),  
TrackTime varchar(10),  
RecipientID varchar(20),  
CustomerID varchar(20),  
Weight numeric(10, 2),  
Timeliness numeric(1,0),  
ProductType varchar(20),  
PackageType varchar(10),  
Hazardous numeric(1,0),  
International numeric(1,0),  
ServiceType varchar(30),  
timeYear numeric(4,0),  
BillNo varchar(20),  
primary key (PackageID),  
foreign key (TrackID, TrackTime) references Tracking (TrackID, TrackTime) on delete set null,  
foreign key (RecipientID) references Recipient (RecipientID) on delete set null,  
foreign key (CustomerID) references Customer (CustomerID) on delete set null,  
foreign key (BillNo) references Bill (BillNo) on delete set null);
```

Package relation은 numeric으로 설정된 key들이 있고 varchar로 설정된 key들이 있다. Weight의 경우 소수점 2자리까지 허용한 10자리 숫자들로 설정했다. Timeliness는 flag변수가 0과 1로 구분되듯 한글자 숫자로 구분하게 설정했다. 위험한 상품인지를 나타내는

Hazardous와 International Delivery를 나타내는 International도 마찬가지로이다. TimeYear의 경우에는 2019, 2020과 같은식으로 4자리 숫자로 나타낼 수 있게 설정했다. 나머지의 경우 영문자가 입력될 수 있기 때문에 varchar로 설정했다.

3-6 Bill

```
create table IF NOT EXISTS Bill (  
    BillNo varchar(20),  
    CustomerID varchar(20),  
    BillDate numeric(8,0),  
    Amount numeric(10,0),  
    primary key (BillNo),  
    foreign key (CustomerID) references Customer (CustomerID));
```

BillNo와 CustomerID에는 영문자가 들어갈 수 있기 때문에 varchar로 설정했다. BillDate는 20190210처럼 년, 월, 일이 모두 들어가기 때문에 소수점없는 8자리로 설정했다. 영수증 금액합을 나타내는 Amount의 자리수는 10자리로 설정했다.

4. Create Table Using Visual Studio and ODBC

Visual Studio에 20170795_1.txt라는 파일을 만들었으며, 해당 파일에는 다음과 같은 코드가 들어가 있다.

```
1 create table IF NOT EXISTS Customer(CustomerID varchar(20), CustomerAddress varchar(30), CustomerName varchar(30), CustomerEmail varchar(40), IsPaid numeric(1, 0), CustomerPhoneNumber varchar(20), CustomerPhone varchar(20), RecipientID varchar(20), RecipientName varchar(20), RecipientAddress varchar(30), RecipientPhoneNumber varchar(20), PackageID varchar(20), primary key(RecipientID));  
2 create table IF NOT EXISTS Recipient(RecipientID varchar(20), RecipientName varchar(20), RecipientAddress varchar(30), RecipientPhoneNumber varchar(20), PackageID varchar(20), primary key(RecipientID));  
3 create table IF NOT EXISTS Tracking(TrackID varchar(20), TrackTime varchar(10), TrackDate varchar(10), Location varchar(100), ScheduledArrivalTime numeric(4,0), HeadLocation varchar(30), primary key(TrackID));  
4 create table IF NOT EXISTS Vehicle(VehicleID varchar(20), VehicleName varchar(20), VehicleType varchar(20), PlateNum varchar(20), primary key(VehicleID));  
5 create table IF NOT EXISTS VehicleType(VehicleName varchar(20), VehicleType varchar(20), primary key(VehicleName));  
6 create table IF NOT EXISTS DeliveredBy(VehicleID varchar(20), TrackID varchar(20), TrackTime varchar(10), foreign key (VehicleID) references Vehicle (VehicleID) on delete set null, foreign key (TrackID, TrackTime) references Tracking (TrackID, TrackTime) on delete set null, primary key (VehicleID, TrackID, TrackTime));  
7 create table IF NOT EXISTS Package(PackageID varchar(20), TrackID varchar(20), TrackTime varchar(10), RecipientID varchar(20), CustomerID varchar(20), Weight numeric(10, 2), Timeliness numeric(1,0), ProductID varchar(20), primary key (PackageID, TrackID, TrackTime));  
8 create table IF NOT EXISTS Bill (BillNo varchar(20), CustomerID varchar(20), BillDate numeric(8,0), Amount numeric(10,0), primary key (BillNo), foreign key (CustomerID) references Customer (CustomerID));
```

이는 SQL문으로 각각의 relation을 생성해준다.

```
FILE* fp = fopen("20170795_1.txt", "r");  
if (fp == NULL) {  
    fprintf(stdout, "FILE OPEN ERROR\n");  
    return 0;  
}  
char s[1001];  
char query0[100000];  
while (fgets(s, 1001, fp) != NULL) {  
    strncpy(query0, s, sizeof(s));  
    state = mysql_query(connection, query0);  
}
```

C파일에서는 먼저 FILE* fp로 해당 sql문이 담겨있는 파일을 연 후, 문장을 line by line으로 읽어 mysql_query를 통해 odbc에 연결해 실행시킨다.

5. Dropping Table Using Visual Studio and ODBC

```
drop table bill;
drop table vehicletype;
drop table deliveredby;
drop table package;
drop table Recipient;
drop table Tracking;
drop table vehicle;
drop table Customer;
```

Drop table을 하기 위한 SQL 문은 위와 같다. 이는 Insert했을 때와 마찬가지로 line by line으로 읽은 후 mysql_query를 통해 실행시킨다. 코드는 아래와 같다.

```
else if (n == 0) {
    FILE* dropTxt = fopen("20170795_3.txt", "r");
    char dropTable[100000];
    while (fgets(s, 1001, dropTxt) != NULL) {
        strncpy(dropTable, s, sizeof(s));
        int state2 = mysql_query(connection, dropTable);
    }
    mysql_close(connection);
    printf("HAVE A GOOD DAY :)");
    break;
}
```

drop을 한 다음에는 프로그램을 종료해야하므로 mysql_close한다.

6. Query Explanation

6-1. 1번 쿼리

1번 쿼리는 사고난 vehicle과 관련된 질문을 묻는 쿼리로, 3가지 sub 쿼리로 나뉜다.

6-1-1. 1-1번 쿼리

1-1번 쿼리는 사고가 났을 때 트럭에 패키지를 싣고 있던 고객을 찾는다. 이를 해결하기 위해 코드 및 쿼리문은 다음과 같이 설정했다.


```

if (k == 1) {
    printf("**Find all customers who had a package on the truck at the time of the crash**\n");
    char query[1000] = "Select distinct CustomerName\
from Customer, Package, DeliveredBy, Vehicle, Tracking, Recipient\
where Customer.CustomerID = Package.CustomerID and Package.TrackID = Tracking.TrackID and Package.TrackTime = Tracking.TrackTime and\
Tracking.TrackID = DeliveredBy.TrackID and Tracking.TrackTime = DeliveredBy.TrackTime\
and DeliveredBy.VehicleID = Vehicle.VehicleID and Vehicle.VehicleID = \"1721\" and Package.RecipientID = Recipient.RecipientID and Customer.CustomerAddress != Recipient.RecipientAddress";
    int statel1 = mysql_query(connection, query);
    if (statel1 == 0) {
        printf("\n---Customer Name---\n");
        sql_result = mysql_store_result(connection);
        while ((sql_row = mysql_fetch_row(sql_result)) != NULL)
        {
            printf("%s\n", sql_row[0]);
        }
        printf("\n");
        mysql_free_result(sql_result);
    }
}

```

먼저, CustomerName이 중복되지 않게 distinct option을 설정했으며 vehicle과 customer 사이에 relationship이 없기 때문에 5개의 relation을 join시켰다. 또, VehicleID를 1721로 고정해서 주었다. 해당 sql을 string에 담은 후 mysql_query로 실행시켰으며, 이를 sql_row값으로 line by line으로 프린트했다. 실행결과는 다음과 같다.

```

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

1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Which type of query? 1

----- Subtypes in TYPE I -----

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

-----
Select the query type : 1

**Find all customers who had a package on the truck at the time of the crash**

---Customer Name---
Lily Kim
Jeyme Kim
Lystarzia
Jennie Lee

```

6-1-2. 1-2번 쿼리

1-2번 쿼리는 사고가 났을 때 recipient name을 출력하게 한다. 1-1과 같은 방식으로 진행했으며 select부분에서 recipient name으로 바뀌었다. 코드 및 쿼리는 다음과 같다.

```

else if (k == 2) {
    printf("**Find all recipients who had a package on that truck at the time of the crash.**\n");
    char query[1000] = "Select distinct RecipientName\
    from Customer, Package, DeliveredBy, Vehicle, Tracking, Recipient\
    where Customer.CustomerID = Package.CustomerID and Package.TrackID = Tracking.TrackID and Package.TrackTime = Tracking.TrackTime and\
    Tracking.TrackID = DeliveredBy.TrackID and Tracking.TrackTime = DeliveredBy.TrackTime\
    and DeliveredBy.VehicleID = Vehicle.VehicleID and Vehicle.VehicleID = \"1721\" and Package.RecipientID = Recipient.RecipientID and Tracking.Location != Recipient.RecipientAddress";
    int statel1 = mysql_query(connection, query);
    if (statel1 == 0) {
        printf("---Recipient---\n");
        sql_result = mysql_store_result(connection);
        while ((sql_row = mysql_fetch_row(sql_result)) != NULL)
        {
            printf("%s\n", sql_row[0]);
        }
        printf("\n");
        mysql_free_result(sql_result);
    }
}

```

실행 결과는 다음과 같다.

```

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

1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Which type of query? 1

```

```

----- Subtypes in TYPE I -----

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

```

Select the query type : 2

```

**Find all recipients who had a package on that truck at the time of the crash.**
---Recipient---
Yewon
SangBeen
Hyejin Park
Minji Park

```

6-1-3. 1-3번 쿼리

1-3번 쿼리는 사고가 나기 전 가장 마지막으로 성공적으로 배달한 delivery를 찾는다.

```

else if (k == 3) {
    printf("**Find the last successful delivery by that truck prior to the crash**\n");
    char query[1000] = "with lastDelivery(deliveryTime) as (\n
        select max(Tracking.TrackTime)\n
        from Customer, Package, DeliveredBy, Vehicle, Tracking, Recipient\n
        where Customer.CustomerID = Package.CustomerID and Package.TrackID = Tracking.TrackID and Package.TrackTime = Tracking.TrackTime and\n
        Tracking.TrackID = DeliveredBy.TrackID and Tracking.TrackTime = DeliveredBy.TrackTime\n
        and DeliveredBy.VehicleID = Vehicle.VehicleID and Vehicle.VehicleID = \"1721\" and Package.RecipientID = Recipient.RecipientID\n
        and Recipient.RecipientAddress = Tracking.Location)\n
        select TrackID\n
        from Tracking, lastDelivery\n
        where Tracking.TrackTime = lastDelivery.deliveryTime";
    int statell = mysql_query(connection, query);
    if (statell == 0) {
        sql_result = mysql_store_result(connection);
        while ((sql_row = mysql_fetch_row(sql_result)) != NULL)
        {
            printf("Last Successful Delivery is %s\n", sql_row[0]);
        }
        printf("\n");
        mysql_free_result(sql_result);
    }
}

```

query문에서 customer, package, deliveredby, vehicle, tracking, recipient relation을 join시킨 후 where절에서 Recipient.Address = Tracking.Location이라는 옵션을 주었다. 이는, 수신자의 주소값과 tracking하고있는 주소가 같으면 배송이 이미 도착했다는 것을 의미한다. with절을 통해 max속성을 주어 가장 늦은 track time을 찾았다. 모든 시간은 오후 7시 30분이면 1930, 여섯시 십오분이면 615와 같이 나타내지기 때문에 max값을 통해 가장 마지막으로 배달에 성공한 물품을 쉽게 찾을 수 있다.

```

----- Subtypes in TYPE I -----

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

Select the query type : 3

**Find the last successful delivery by that truck prior to the crash**
Last Successful Delivery is bf45DDDr

```

결과값은 아래와 같이 trackID로 나타난다.

6-2 2번 쿼리

2번 쿼리는 특징 해에 가장 많은 소포를 보낸 고객을 찾는다. 이를 해결하기 위해 쿼리문 및 코드를 다음과 같이 짰다.

```

else if (n == 2) {
    while (1) {
        printf("***Find the customer who has shipped the most packages in the past certain year.**\n");
        char year[11];
        printf("Which year? ");          //get the year value
        scanf("%s", year);
        if (!strcmp(year, "0")) break;    //if 0, break to the main frame
        char query[1000] = "with bestBuyer(amount, customerBuyer) as(\n";
        select count(distinct packageID), Customer.CustomerName\
        from Customer, Package\
        where Customer.CustomerID = Package.CustomerID and timeYear = ";
        strcat(query, year);
        char query2[100] = " ) select max(amount), customerBuyer from bestBuyer";    //concatenate the string
        strcat(query, query2);
        int state11 = mysql_query(connection, query);
        if (state11 == 0) {                //if the connection is made
            sql_result = mysql_store_result(connection);
            bool flag = true;              //flag to let us know if the result is NULL or not
            while ((sql_row = mysql_fetch_row(sql_result)) != NULL)
            {
                if (sql_row[1] == NULL) {
                    flag = false;
                    break;
                }
                printf("<%=> has shipped the most packages that year \n\n", sql_row[1]);
            }
            if (flag == false) {
                printf("No one has shipped the packages that year\n\n");    //when there is NULL result
            }
            mysql_free_result(sql_result);
        }
    }
}

```

먼저, 어떤 해에 대한 정보를 받을지 입력을 받는다. 그리고 그 속성을 timeYear이라는 값에 넣어주기 위해 쿼리문을 slicing해 <string.h>의 strcat 속성을 통해 붙혀넣는다. query문은 간단하다. Customer와 Package를 timeYear값을 준상태에서 join하고, bestBuyer에 count(distinct PackageID) 와 customer이름 속성을 준다. distinct PackageID 는 보낸 소포의 개수 나타낸다. 그후, 다시한번 select를 통해 max(amount)를 해 가장 많이 보낸 고객을 찾는다.

```

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

1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Which type of query? 3
** Find the customer who has spent the most money on shipping in the past certain year.**
Which Year? 2019
Customer Name : Moon SangBeen

```

실행 화면은 위와 같다. 만약 해당 해에 아무도 구입한 이력이 없을 때는 아무도 구매하지 않았다는 prompt가 뜨게 만들었다.

```

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

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

Which type of query? 3
** Find the customer who has spent the most money on shipping in the past certain year.**
Which Year? 2000
No one has bought anything during the year

```

6-3 3번 쿼리

3번 쿼리는 특정 해에 소포를 보내는 데에 가장 많은 돈을 쓴 고객을 찾는다.

쿼리문 및 코드는 아래와 같다.

```

else if (n == 3) {
    while (1) {
        char year[11];
        printf("*** Find the customer who has spent the most money on shipping in the past certain year.**\n");
        printf("Which Year? ");
        scanf("%s", year);
        if (!strcmp(year, "0")) break; //loop until it is not 0
        char query[200] = "select Customer.CustomerName, max(Bill.Amount)\n";
        from Customer, Bill\
        where Customer.CustomerID = Bill.CustomerID and Bill.BillDate >= "; //BillDate>=20190101 ~ BillDate<=20191231
        strcat(query, year);
        char mid[100] = "0101 and Bill.BillDate <=";
        strcat(query, mid);
        strcat(query, year);
        char end[100] = "1231";
        strcat(query, end);
        int state11 = mysql_query(connection, query);
        if (state11 == 0) {
            sql_result = mysql_store_result(connection);
            bool flag = false;
            while ((sql_row = mysql_fetch_row(sql_result)) != NULL)
            {
                if (sql_row[1] == NULL) {
                    flag = false;
                    break;
                }
                flag = true;
                printf("Customer Name : %s\n", sql_row[0]);
            }
            if (flag == false) {
                printf("No one has bought anything during the year\n"); //when there is no customer Record
                continue;
            }
            mysql_free_result(sql_result);
        }
    }
}

```

2번과 마찬가지로 어떤 해인지 알아야하기 때문에 scanf를 통해 입력을 받았다. Bill relation에는 따로 해의 속성만을 명시하는 항목이 없기 때문이다. 그러나 BillDate가 8자리로 20190910(2019년 9월 10일)과 같은 방법으로 날짜를 나타내기 때문에 연도가 주어지면 해당연도0101(해당연도 1월 1일)보다는 같거나 크고 해당연도1231(해당연도 12월 31일)보다는 같거나 작은 값을 선택하게 만들었다

2번 쿼리와 마찬가지로 쿼리 결과값이 없을 수 있기 때문에 flag변수를 사용해서 아무런

customr가 없을 경우 아무도 사지 않았음을 나타낼 수 있게 만들었다.

실행화면은 아래와 같다.

```
----- SELECT QUERY TYPES -----
1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Which type of query? 3
** Find the customer who has spent the most money on shipping in the past certain year.**
Which Year? 2017
Customer Name : Jane

Which type of query? 3
** Find the customer who has spent the most money on shipping in the past certain year.**
Which Year? 20000
No one has bought anything during the year
```

6-4 4번 쿼리

4번 쿼리문은 예정시간에 배달물이 도착하지 않은 shipping을 고른다. 쿼리문 및 코드는 다음과 같다

```
else if (n == 4) {
    int k;
    while (1) {
        printf("Find those packages that were not delivered within the promised time\n");
        printf("IF you want to skip to main frame, press 0 : "); //give out the result until input is 0
        scanf("%d", &k);
        if (k == 0) break;
        const char* query = "select PackageID from Package where Timeliness = 0";
        int state11 = mysql_query(connection, query);
        if (state11 == 0) {
            sql_result = mysql_store_result(connection);
            while ((sql_row = mysql_fetch_row(sql_result)) != NULL) //fetch the row
            {
                printf("%s\n", sql_row[0]); //print it in the console
            }
            mysql_free_result(sql_result);
        }
    }
}
```

package에서 timeliness가 0이면 아직 도착하지 않았음을 나타낸다. 따라서 해당 속성을 이용해서 결과를 출력했다. 실행화면은 다음과 같다.


```

----- SELECT QUERY TYPES -----
1. TYPE I
2. TYPE II
3. TYPE III
4. TYPE IV
5. TYPE V
0. QUIT
Which type of query? 5
Generate the bill for each customer for the past certain month.
Customer Name : Moon SangBeen
Which year? 2020
Generating Bill..
Generation Completed
Please check 20170795_4.txt

```

20170795_4.txt의 파일은 다음과 같다.

```

Customer Name   Address   Amount
Moon SangBeen   Gyeonggi   99000
-----
Itemized Billing List
=====
Bill No | PackageID | Price | Product Type | Package Type | Service Type | Number Bought
| 1 | jydd5 | $99000 | | box | Clothing | Prepaid(reserved) | 1 |
-----

```

먼저 CustomerName, Address, Amount에 대한 bill을 출력하며 이에 대해 itemize한 bill을 출력한다.

7. 전체 코드

```
#include <stdio.h>
```

```
#include "mysql.h"
```

```
#include <string.h>
```

```
#pragma comment(lib, "libmysql.lib")
```



```

const char* host = "localhost";

const char* user = "root";

const char* pw = "Qkekwhgdk98!";

const char* db = "data";

int state;


int main(void) {

    MYSQL* connection = NULL;

    MYSQL conn;

    MYSQL_RES* sql_result;

    MYSQL_ROW sql_row;


    if (mysql_init(&conn) == NULL)

        printf("mysql_init() error!");


    connection = mysql_real_connect(&conn, host, user, pw, db, 3306, (const
char*)NULL, 0);

    if (connection == NULL)

    {

        printf("%d ERROR : %s\n", mysql_errno(&conn), mysql_error(&conn));

        return 1;

    }
}

```

```

else
{
    printf("Connection Succeeded\n"); //if connection is made successfull

    if (mysql_select_db(&conn, db))
    {
        printf("%d ERROR : %s\n", mysql_errno(&conn),
mysql_error(&conn));

        return 1;
    }

    FILE* fp = fopen("20170795_1.txt", "r"); //open txt that creates relations
    if (fp == NULL) {
        fprintf(stdout, "FILE OPEN ERROR\n");
        return 0;
    }

    char s[1001];
    char query0[100000];

    while (fgets(s, 1001, fp) != NULL) { //process query
        strncpy(query0, s, sizeof(s));
        state = mysql_query(connection, query0);
    }

    fp = fopen("20170795_2.txt", "r"); //insert values

```

```

if (fp == NULL) {

    fprintf(stdout, "FILE OPEN ERROR\n");

    return 0;

}

while (fgets(s, 1001, fp) != NULL) {

    strncpy(query0, s, sizeof(s));

    state = mysql_query(connection, query0);

}

while (1) {

    printf("\n----- SELECT QUERY TYPES ----- \n\n");

//main frame

    printf("\t1. TYPE I\n");

    printf("\t2. TYPE II\n");

    printf("\t3. TYPE III\n");

    printf("\t4. TYPE IV\n");

    printf("\t5. TYPE V\n");

    printf("\t0. QUIT\n");

    printf("Which type of query? ");

    int n;

    scanf("%d", &n);                //get the query type from input

    if (n == 1) {

        while (1) {

            printf("\n----- Subtypes in TYPE I -----

```

WnWn");//type 1 has 3 subtypes

```
printf("Wt0. QUITWn");

printf("Wt1. TYPE I-1Wn");

printf("Wt2. TYPE I-2Wn");

printf("Wt3. TYPE I-3Wn");

printf("-----Wn");

printf("Select the query type : ");

int k; scanf("%d", &k);

printf("Wn");

if (k == 0) break;

if (k == 1) {

    printf("**Find all customers who had a
package on the truck at the time of the crash**Wn");

    char query[1000] = "Select distinct
CustomerNameW
from Customer, Package, DeliveredBy, Vehicle, Tracking,
RecipientW
where Customer.CustomerID = Package.CustomerID and Package.TrackID
= Tracking.TrackID and Package.TrackTime = Tracking.TrackTime andW
Tracking.TrackID = DeliveredBy.TrackID and Tracking.TrackTime =
DeliveredBy.TrackTimeW
and DeliveredBy.VehicleID = Vehicle.VehicleID and
Vehicle.VehicleID = W"1721W"and Package.RecipientID = Recipient.RecipientID and
Customer.CustomerAddress != Recipient.RecipientAddress";

    int state11 = mysql_query(connection,
```

```

query);          //store the query in string and process it

                    if (state11 == 0) {

                        printf("\n---Customer Name---
\n");

                        sql_result =

mysql_store_result(connection);

                        while ((sql_row =
mysql_fetch_row(sql_result)) != NULL)    //fetch it line by line

                            {

                                printf("%s\n", sql_row[0]);

                                //print the result on console

                            }

                        printf("\n");

                        mysql_free_result(sql_result);

                    }

                }

                else if (k == 2) {

                    printf("**Find all recipients who had a
package on that truck at the time of the crash.**\n");

                    char query[1000] = "Select distinct
RecipientName
from Customer, Package, DeliveredBy, Vehicle,
Tracking, Recipient
where Customer.CustomerID = Package.CustomerID and
Package.TrackID = Tracking.TrackID and Package.TrackTime = Tracking.TrackTime and

```

```

        Tracking.TrackID = DeliveredBy.TrackID and
Tracking.TrackTime = DeliveredBy.TrackTime

        and DeliveredBy.VehicleID = Vehicle.VehicleID and
Vehicle.VehicleID = "1721" and Package.RecipientID = Recipient.RecipientID and
Tracking.Location != Recipient.RecipientAddress";

        int state11 = mysql_query(connection,
query);

        if (state11 == 0) {

            printf("---Recipient---\n");

            sql_result =
mysql_store_result(connection);

            while ((sql_row =
mysql_fetch_row(sql_result)) != NULL)

                {

                    printf("%s\n", sql_row[0]);

                }

            printf("\n");

            mysql_free_result(sql_result);

        }

    }

    else if (k == 3) {

        printf("***Find the last successful delivery
by that truck prior to the crash**\n");

        char query[1000] = "with
lastDelivery(deliveryTime) as ("

```

```

select max(Tracking.TrackTime)
from Customer, Package, DeliveredBy,
Vehicle, Tracking, Recipient
where Customer.CustomerID =
Package.CustomerID and Package.TrackID = Tracking.TrackID and Package.TrackTime =
Tracking.TrackTime and
Tracking.TrackID = DeliveredBy.TrackID
and Tracking.TrackTime = DeliveredBy.TrackTime
and DeliveredBy.VehicleID =
Vehicle.VehicleID and Vehicle.VehicleID = "1721" and Package.RecipientID =
Recipient.RecipientID
and Recipient.RecipientAddress =
Tracking.Location)

select TrackID
from Tracking, lastDelivery
where Tracking.TrackTime =
lastDelivery.deliveryTime";

int state11 = mysql_query(connection,
//store the query in string
if (state11 == 0) {
    sql_result =
mysql_store_result(connection);

    while ((sql_row =
mysql_fetch_row(sql_result)) != NULL)

{
    printf("Last Successful

```

```

Delivery is %s\n", sql_row[0]);

    }

    printf("\n");

    mysql_free_result(sql_result);

}

}

}

else if (n == 2) {

    while (1) {

        printf("***Find the customer who has shipped the
most packages in the past certain year.**\n");

        char year[11];

        printf("Which year? ");

        //get the year value

        scanf("%s", year);

        if (!strcmp(year, "0")) break;           //if 0,

break to the main frame

        char query[1000] = "with bestBuyer(amount,
customerBuyer) as(W

        select count(distinct packageID), Customer.CustomerNameW

        from Customer, PackageW

        where Customer.CustomerID = Package.CustomerID and timeYear = ";

        strcat(query, year);

```



```

char query2[100] = "    ) select max(amount),
customerBuyer from bestBuyer";
//concatenate the string

strcat(query, query2);

int state11 = mysql_query(connection, query);

if (state11 == 0) {
    //if the connection is made

    sql_result =
mysql_store_result(connection);

    bool flag = true;
    //flag to let us know if the result is NULL or not

    while ((sql_row =
mysql_fetch_row(sql_result)) != NULL)

    {

        if (sql_row[1] == NULL) {

            flag = false;

            break;

        }

        printf("<%s> has shipped the
most packages that year %s\n", sql_row[1]);

    }

    if (flag == false) {

        printf("No one has shipped the
packages that year\n");
//when there is NULL result

    }

    mysql_free_result(sql_result);

```

```

    }

    }

}

else if (n == 3) {

    while (1) {

        char year[11];

        printf("** Find the customer who has spent the
most money on shipping in the past certain year.**\n");

        printf("Which Year? ");

        scanf("%s", year);

        if (!strcmp(year, "0"))break;

//loop until it is not 0

        char query[200] = "select
Customer.CustomerName, max(Bill.Amount)W

        from Customer, BillW

        where Customer.CustomerID = Bill.CustomerID and Bill.BillDate
>= ";

        //BillDate>=20190101 ~ BillDate<=20191231

        strcat(query, year);

        char mid[100] = "0101 and Bill.BillDate <= ";

        strcat(query, mid);

        strcat(query, year);

        char end[100] = "1231";

        strcat(query, end);

        int state11 = mysql_query(connection, query);

```

```

        if (state11 == 0) {

            sql_result =

mysql_store_result(connection);

            bool flag = false;

            while ((sql_row =

mysql_fetch_row(sql_result)) != NULL)

                {

                    if (sql_row[1] == NULL) {

                        flag = false;

                        break;

                    }

                    flag = true;

                    printf("Customer Name : %s\n",

sql_row[0]);

                }

            if (flag == false) {

                printf("No one has bought

anything during the year\n");

                //when there is no customer Record

                continue;

            }

            mysql_free_result(sql_result);

        }

    }
}

```

```

else if (n == 4) {

    int k;

    while (1) {

        printf("Find those packages that were not
delivered within the promised time\n");

        printf("IF you want to skip to main frame, press
0 : ");

        //give out the result until input is 0

        scanf("%d", &k);

        if (k == 0) break;

        const char* query = "select PackageID from
Package where Timeliness = 0";

        int state11 = mysql_query(connection, query);

        if (state11 == 0) {

            sql_result =
mysql_store_result(connection);

            while ((sql_row =
mysql_fetch_row(sql_result)) != NULL)
                //fetch the row

            {

                printf("%s\n", sql_row[0]);

                //print it in the console

            }

            mysql_free_result(sql_result);

        }

    }
}

```

```

    }

    else if (n == 5) {

        int sss;

        while (1) {

            printf("If you want to quit, press 0 : ");

            scanf("%d", &sss);

            if (sss == 0) break;

            char year[5];

            FILE* sp = fopen("20170795_4.txt", "w");

            printf("Generate the bill for each customer for the past
certain month.\n");

            printf("Customer Name : ");

            while (getchar() != '\n');

            char tmp[200];

            fgets(tmp, 200, stdin);

            printf("Which year? ");

            fscanf(stdin, "%s", year);

            tmp[strlen(tmp) - 1] = '\0';

            printf("Generating Bill..\n");

            printf("Generation Completed\n");

            printf("Please check 20170795_4.txt\n");

            printf("\n");

            char query[1000] = "select CustomerName,

```

```
CustomerAddress, BillNo, amount, ServiceType, packageID, ProductType, PackageType,
sum(amount), count(BillNo)W
```

```
from Bill, Customer, PackageW
```

```
where Bill.CustomerID = Customer.CustomerID and
Customer.CustomerID = Package.CustomerID and CustomerName = W"";
```

```
strcat(query, tmp);
```

```
char tt[30] = "W" and timeYear = W"";
```

```
strcat(query, tt);
```

```
strcat(query, year);
```

```
char ttt[100] = "W"group by amount";
```

```
strcat(query, ttt);
```

```
int state11 = mysql_query(connection, query);
```

```
if (state11 == 0) {
```

```
    sql_result = mysql_store_result(connection);
```

```
    sql_row = mysql_fetch_row(sql_result);
```

```
    fprintf(sp, "Customer NameWt AddressWt
AmountWn");
```

```
    fprintf(sp, "%sWt%sWt%sWn", sql_row[0],
sql_row[1], sql_row[8]);
```

```
    fprintf(sp, "-----
-----Wn");
```

```
    fprintf(sp, "Itemized Billing ListWn");
```

```
    fprintf(sp,
"=====
=====Wn");
```

```

                                fprintf(sp, "Bill No | PackageID | Price | Product
Type | Package Type | Service Type | Number Bought\n");

```

```

                                fprintf(sp,
"|Wt%SWt|Wt%SWt|WtWt$%SWtWt|WtWt%SWtWt|Wt%SWt|Wt%SWt|Wt%SWt|Wt%SWt\n", sql_row[2],
sql_row[5], sql_row[3], sql_row[7], sql_row[6], sql_row[4], sql_row[9]);

```

```

                                fprintf(sp, "-----
-----\n");

```

```

                                fclose(sp);

```

```

                                mysql_free_result(sql_result);

```

```

                                }

```

```

                                }

```

```

                                }

```

```

                                else if (n == 0) {

```

```

                                FILE* dropTxt = fopen("20170795_3.txt", "r");

```

```

                                char dropTable[100000];

```

```

                                while (fgets(s, 1001, dropTxt) != NULL) {

```

```

                                        strncpy(dropTable, s, sizeof(s));

```

```

                                        int state2 = mysql_query(connection, dropTable);

```

```

                                }

```

```

                                mysql_close(connection);

```

```

                                printf("HAVE A GOOD DAY :)");

```

```

                                break;

```

```

                                }

```

```

                                }

```

```
    }  
    return 0;  
}
```