CS 6360-002 Database Design Final Project
Furniture Store Database System

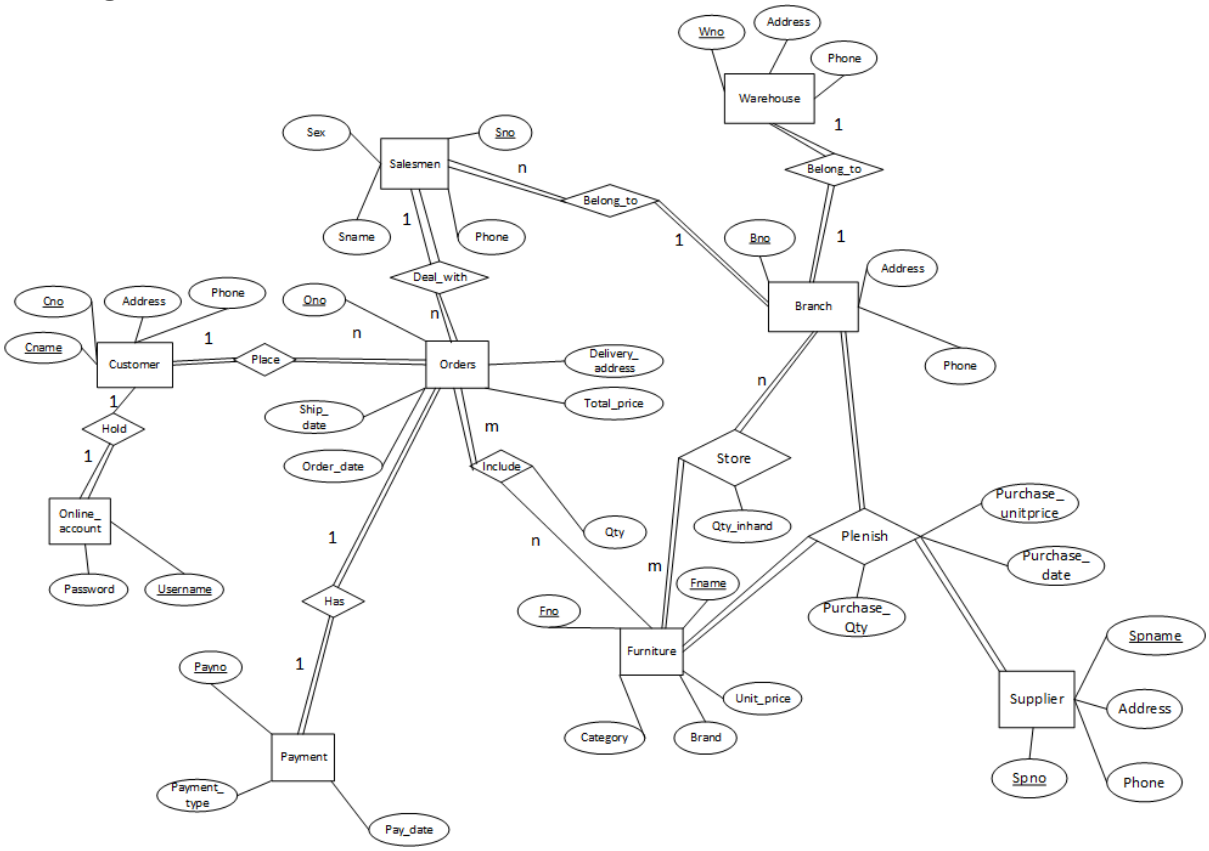
Yuzhuang Feng yxf160930
Wenqian Ji wxj160230
You Jia yxj161630

Data Requirements

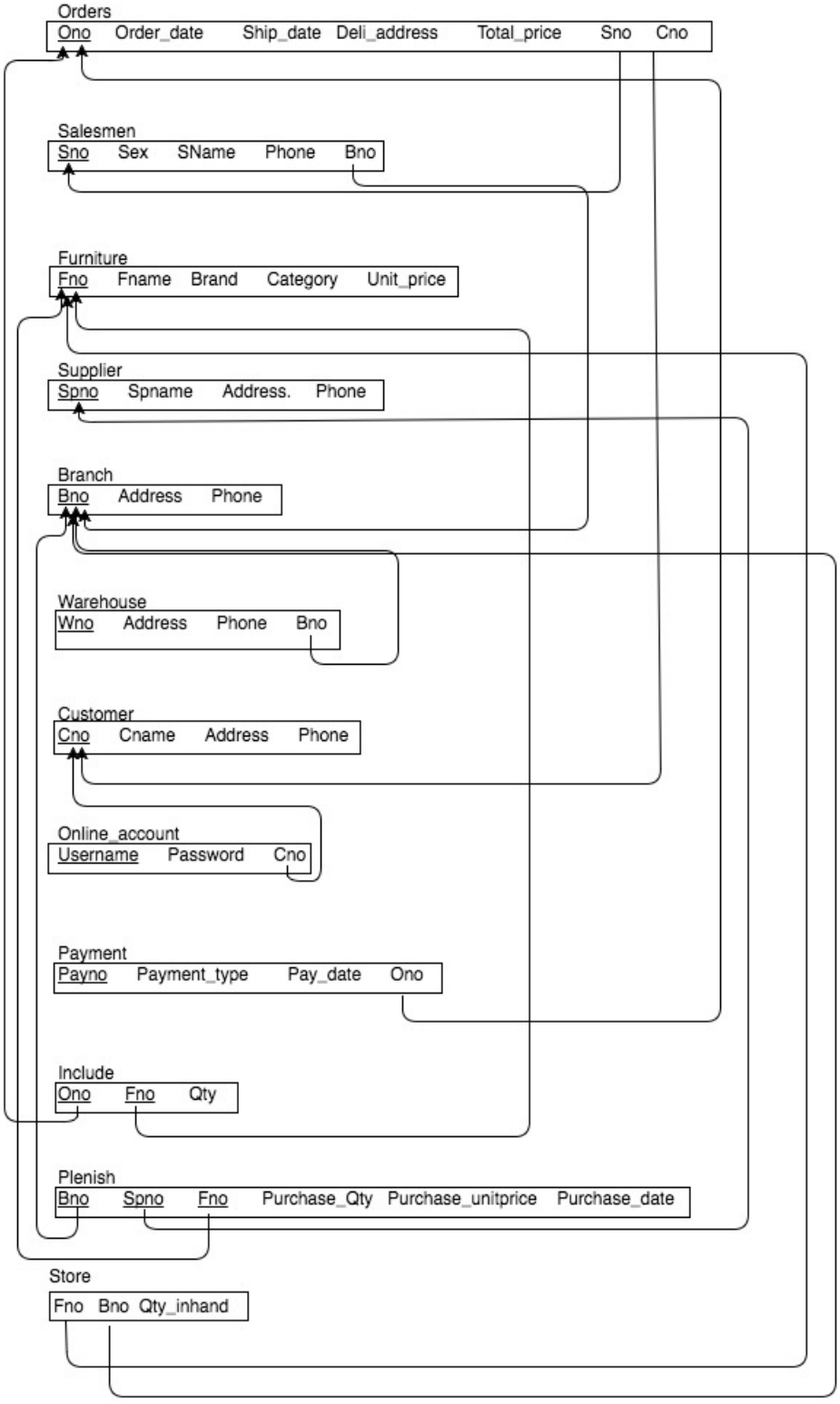
The furniture store needs to keep track of data including entities salesmen, orders, customer, furniture, supplier, branch, warehouse, payment, online-account.

1. Each Furniture can be described by a unique Fno or a unique Fname, Category, Brand, Unit_price(retail price).
2. Each Customer can be described by a unique Cno, Address, Cname, Phone.
3. Each Customer has only one Online_account which is described by a unique Username and Password.
4. Each Order can be paid by three different types named Payment_type: cash, credit and gift card. Each Payment can be described by a unique Payno, Pay_date, Payment_type.
5. Each Order can be described by a unique Ono, Order_date,Total_price, Ship_date, Deli_address. Once an order is created, it includes each furniture's Qty.
6. Each Order is dealt with only one Salesmen. Each Salesmen can be described by a unique Sno, Sex, SName and Phone.
7. Each Branch can be described by a unique Bno, Address and Phone. Each Branch has at least one Salesmen.
8. Each Branch has one Warehouse, which is described by Wno, Address and Phone.
9. Each Supplier can be described by a unique Spno, Spname, Address and Phone. When a Branch buy a particular furniture from a particular supplier, it will track Purchase_date, Purchase_Qty and Purchase_unitprice.
10. Each Branch stores many furnitures. Qty_inhand is used to describe a branch storage for a particular furniture.

ER Diagram


Wno
Address
Phone
Warehouse
1
Belong_to
Sex
Sno
Salesmen
n
Belong_to
1
Sname
Phone
1
Bno
1
Deal_with
Address
Ono
Address
Phone
Ono
n
Branch
Cname
Customer
1
Place
n
Orders
Delivery_
address
Phone
1
Ship_
date
Total_price
n
Hold
m
Store
1
Order_date
Include
Purchase_
unitprice
Online_
account
Qty
Qty_inhand
Plenish
1
n
m
Purchase_
date
Fname
Password
Username
Has
Fno
Purchase_
Qty
Furniture
Spname
Payno
1
Unit_price
Supplier
Address
Category
Brand
Payment
Spno
Phone
Payment_
type
Pay_date

Original Relational Schema



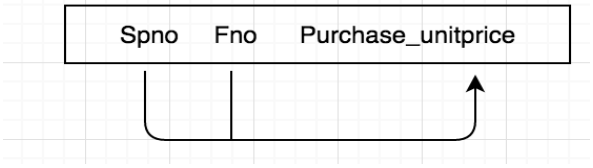
Normalization Violations

All the relations are in 1NF, 2NF. However, there exist a relation Plenish violating 3NF.

Plenish

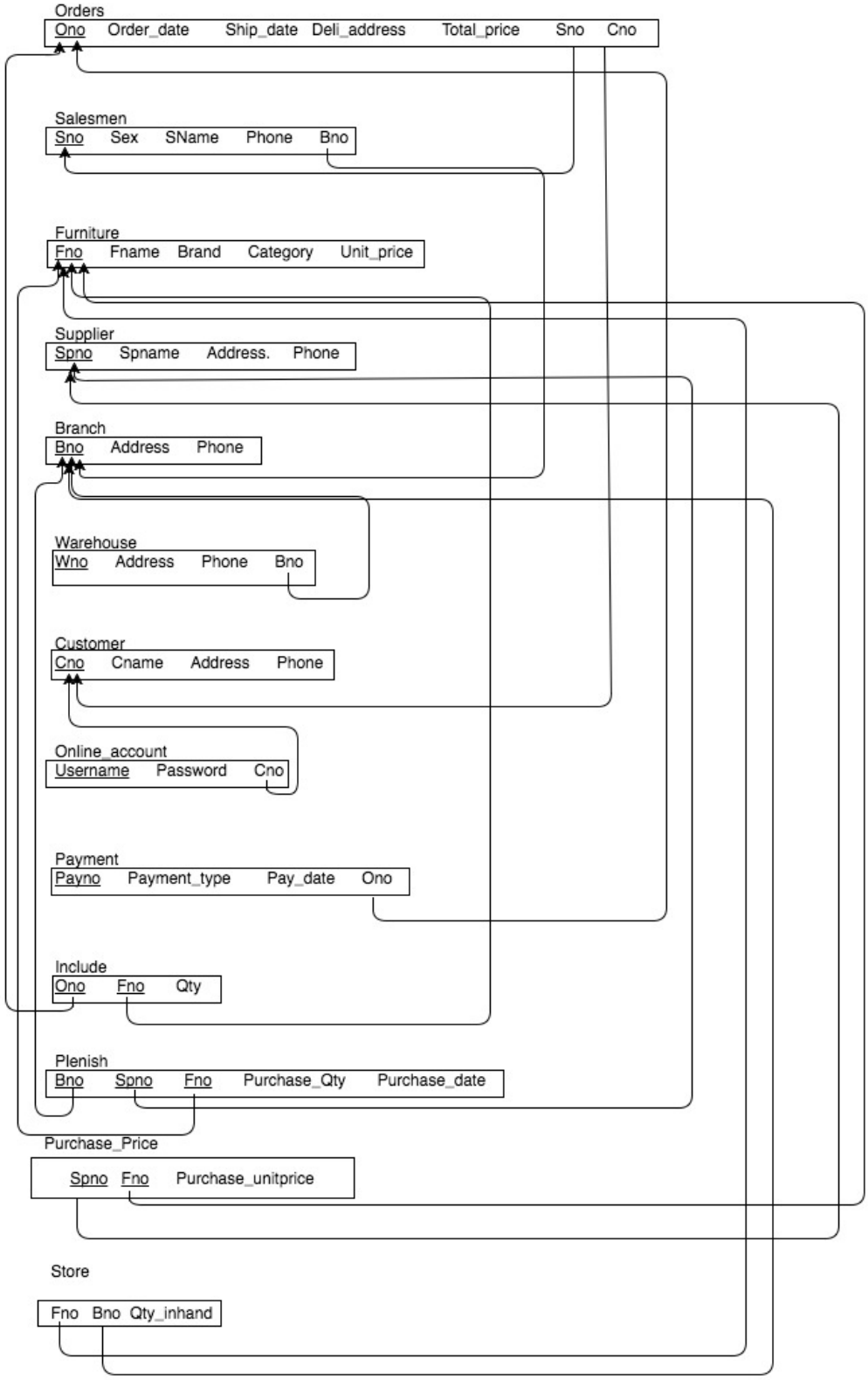
Bno	Spno	Fno	Purchase_Qty	Purchase_unitprice	Purchase_date

The relations that violate 3NF in this table are:



Supplier number and Furniture number can determine the purchase unit price for the furniture. Hence, there exists transitive dependency.

Final Normalized Relational Schema



SQL implementation of the database system

```
CREATE TABLE ORDERS (
        Ono INT(10) NOT NULL,
        Order_date DATE,
        Deli_date DATE,
        Deli_address VARCHAR(40),
        Total_price FLOAT,
        Sno INT(5),
        Cno INT(7),
        PRIMARY KEY (Ono)
);

CREATE TABLE SHIPMENT (
        Ono INT(10) NOT NULL,
        Cno INT(7),
        Ship_or_not BOOLEAN,
        PRIMARY KEY (Ono)
);

CREATE TABLE SALESMEN (
        Sno INT(5) NOT NULL,
        Sex VARCHAR(5),
        Sname VARCHAR(40),
        Phone VARCHAR(15),
        Bno INT(3),
        PRIMARY KEY (Sno)
);

CREATE TABLE FURNITURE (
        Fno INT(7) NOT NULL,
        Fname VARCHAR(50),
        Brand VARCHAR(20),
        Category VARCHAR(20),
        Unit_price FLOAT,
        PRIMARY KEY (Fno)
);

CREATE TABLE SUPPLIER (
        Spno INT(3) NOT NULL,
        Spname VARCHAR(50),
        Address VARCHAR(50),
        Phone VARCHAR(15),
        PRIMARY KEY (Spno)
);

CREATE TABLE BRANCH (
        Bno INT(3) NOT NULL,
```

```
        Address VARCHAR(50),
        Phone VARCHAR(15),
        PRIMARY KEY (Bno)
);

CREATE TABLE WAREHOUSE (
        Wno INT(3) NOT NULL,
        Address VARCHAR(50),
        Phone VARCHAR(15),
        Bno INT(3),
        PRIMARY KEY (Wno)
);

CREATE TABLE CUSTOMER(
        Cno INT(7) NOT NULL,
        Cname VARCHAR(40),
        Address VARCHAR(50),
        Phone VARCHAR(15),
        PRIMARY KEY (Cno)
);

CREATE TABLE Online_account (
        Username VARCHAR(20) NOT NULL,
        Password VARCHAR(30),
        Cno INT(7),
        PRIMARY KEY (Username)
);

CREATE TABLE PAYMENT (
        Payno INT(10) NOT NULL,
        Payment_type VARCHAR(10) DEFAULT 'Credit',
        Pay_date DATE,
        Ono INT(10),
        PRIMARY KEY (Payno)
);

CREATE TABLE INCLUDE (
        Ono INT(10) NOT NULL,
        Fno INT(7) NOT NULL,
        Qty INT,
        PRIMARY KEY (Ono, Fno)
);

CREATE TABLE PLENISH (
        Bno INT(3) NOT NULL,
        Spno INT(3) NOT NULL,
        Fno INT(7) NOT NULL,
        Purchase_Qty FLOAT,
```

```
        Purchase_date DATE,
        PRIMARY KEY (Bno, Spno, Fno)
);

CREATE TABLE PURCHASE_PRICE (
        Spno INT(3) NOT NULL,
        Fno INT(7) NOT NULL,
        Purchase_unitprice FLOAT,
        PRIMARY KEY (Spno, Fno)
);

CREATE TABLE STORE (
        Fno INT(7) NOT NULL,
        Bno INT(3) NOT NULL,
        Qty_inhand INT,
        PRIMARY KEY (Fno, Bno)
);

CREATE TABLE CUSTOMER_COUPON (
Ono INT(10),
Cno INT(7),
Number INT
);

ALTER TABLE ORDERS ADD FOREIGN KEY (Sno) REFERENCES SALESMEN(Sno)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE ORDERS ADD FOREIGN KEY (Cno) REFERENCES CUSTOMER(Cno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE SHIPMENT ADD FOREIGN KEY (Ono) REFERENCES ORDERS(Ono)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE SHIPMENT ADD FOREIGN KEY (Cno) REFERENCES CUSTOMER(Cno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE SALESMEN ADD FOREIGN KEY (Bno) REFERENCES BRANCH(Bno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE WAREHOUSE ADD FOREIGN KEY (Bno) REFERENCES BRANCH(Bno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE Online_account ADD FOREIGN KEY (Cno) REFERENCES CUSTOMER(Cno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE PAYMENT ADD FOREIGN KEY (Ono) REFERENCES ORDERS(Ono)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE INCLUDE ADD FOREIGN KEY (Ono) REFERENCES ORDERS(Ono)
        ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE INCLUDE ADD FOREIGN KEY (Fno) REFERENCES FURNITURE(Fno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE PLENISH ADD FOREIGN KEY (Bno) REFERENCES BRANCH(Bno)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE PLENISH ADD FOREIGN KEY (Spno) REFERENCES SUPPLIER(Spno)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE PLENISH ADD FOREIGN KEY (Fno) REFERENCES FURNITURE(Fno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE PURCHASE_PRICE ADD FOREIGN KEY (Spno) REFERENCES SUPPLIER(Spno)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE PURCHASE_PRICE ADD FOREIGN KEY (Fno) REFERENCES FURNITURE(Fno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE STORE ADD FOREIGN KEY (Fno) REFERENCES FURNITURE(Fno)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE STORE ADD FOREIGN KEY (Bno) REFERENCES BRANCH(Bno)
        ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE CUSTOMER_COUPON ADD FOREIGN KEY (Ono) REFERENCES ORDERS(Ono)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE CUSTOMER_COUPON ADD FOREIGN KEY (Cno) REFERENCES CUSTOMER(Cno)
        ON DELETE CASCADE ON UPDATE CASCADE;
```

PL/SQL procedures and triggers

Stored Procedure

1. Add new sno into salesmen

```
CREATE PROCEDURE
  Branch001Add (thisSNO IN SALESMEN.SNO%type) AS
BEGIN
          INSERT INTO SALESMEN(SNO,FNO) VALUES(thisSNO, '001');
END;
.
RUN;
```

2. Write a stored procedure that retrieves the list of orders have been send out at the end of the day today and that prints the order number and delivery address for each.

```
CREATE OR REPLACE PROCEDURE Ship_today
       (thisDate IN ORDERS.Ship_date%TYPE)AS
       Onumber ORDERS.Ono%TYPE;
       customerName CUSTOMER.Cname%TYPE;
       Oaddress ORDERS.Deli_address%TYPE;
       thisOrder Orders%ROWTYPE;
        CURSOR Today
        IS
        SELECT * FROM ORDERS WHERE Ship_date = thisDate;
    BEGIN
    OPEN Today;
    LOOP
         FETCH Today into thisOrder;
         EXIT WHEN (Today%NOTFOUND);
         SELECT  Ono INTO Onumber FROM ORDERS  WHERE ORDERS.Ono = thisOrder.Ono;
          sys.dbms_output.put_line (Onumber);
         SELECT  Deli_address INTO Oaddress FROM ORDERS  WHERE ORDERS.Deli_address =
         thisOrder.Deli_address;
         sys.dbms_output.put_line (Oaddress);
         SELECT Cname INTO customerName FROM CUSTOMER WHERE CUSTOMER.Cno =
         thisOrder.Cno;       sys.dbms_output.put_line (customerName);
    END LOOP;
    CLOSE Today;
END;
```

Trigger

1.After insert or update of of Deli_date on ORDER, change the shipment table based on null or not

```
create trigger Ship_or_not
after insert or update of Deli_date
on ORDERs
for each row
begin
 if :new.Deli_date is null then
   insert into SHIPMENT (Ono, Shipment_or_not)
   values (:new.Ono, false );
 elsif:new.Deli_date is not null then
   insert into SHIPMENT (Ono, Shipment_or_not)
   values (:new.Ono, true);
 end if;
end;
```

2. After the insert of Plenish.Purchase_qty, add it to Store.Qty_inhand

```
CREATE OR REPLACE TRIGGER CUS_COUPON
AFTER INSERT OR UPDATE OF Total_price ON ORDERS
FOR EACH ROW
BEGIN

CASE

WHEN :NEW.Total_price > =1000 THEN
     INSERT INTO CUSTOMER_COUPON
     VALUES (:NEW.Ono,:NEW.Cno, 3);
WHEN :NEW.Total_price  >=500  AND :NEW.Total_price  <1000 THEN
     INSERT INTO CUSTOMER_COUPON
     VALUES (:NEW.Ono,:NEW.SSN, 2);
WHEN :NEW.Total_price < 500  AND :NEW.Total_price  >=100THEN
     INSERT INTO EMPLOYEE_SALARIES
     VALUES (:NEW.Ono,:NEW.SSN, 1);
WHEN :NEW.Total_price <100 THEN
     INSERT INTO CUSTOMER_COUPON
     VALUES (:NEW.Ono,:NEW.Cno, 0);
          END CASE;
END;
```

PHP for create, read, update, delete

a) Create furniture:

When we need to a new furniture data to the database, the webpage screenshots are shown below.

After input a furniture data, click submit, the webpage show that we insert successfully.

Source Code for create are shown below.

createfurniture.html

cations/MAMP/htdocs/createfurniture.html

```
<html>
<form action="createfurniture.php" method="GET">
Number: <input type="text" name="Fno" />
Name: <input type="text" name="Fname" />
Brand: <input type="text" name="Brand" />
Category: <input type="text" name="Category" />
Unit_price: <input type="text" name="Unit_price" />
<input type="submit" />
</form>
</html>
```

createfurniture.php

cations/MAMP/htdocs/createfurniture.php

```
<html>
<head>
<title>Create</title>
</head>

<body>
<?php

$Number=$_GET ["Fno"];
$Name=$_GET ["Fname"];
$Brand=$_GET ["Brand"];
$Category=$_GET ["Category"];
$Unit_price=$_GET ["Unit_price"];

$con = mysqli_connect("127.0.0.1","root","root","furniture store",3306);

if ($con->connect_error) {
    die("Connet failed" . $con->connect_error);
}

$sql = "INSERT INTO Furniture(Fno, Fname, Brand, Category, Unit_price)
VALUES ('$Number','$Name','$Brand','$Category','$Unit_price')";

if ($con->query($sql) === TRUE) {
    echo "Insert successfully.";
} else {
    echo "Insert failed.";
}

mysqli_close($con);

?>
</body>
</html>
```

b) Read furniture

1. Input furniture number 3(screenshots are shown below)

2. **then we get information about the furniture.**

Number: 3 - Name: ARHOLMA - Brand: IKEA - Category: Sofa - Unit_price: 350

Source Code for read:

readfurniture.html

/Applications/MAMP/htdocs/readfurniture.html

```
1  <html>
2  <form action="readfurniture.php" method="GET">
3  Number: <input type="text" name="Fno" />
4  <button type="submit">Read</button>
5  </form>
6  </html>
```

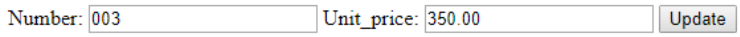
readfurniture.php

/Applications/MAMP/htdocs/readfurniture.php

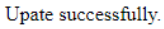
```
1  <html>
2  <head>
3  <title>Read</title>
4  </head>
5
6  <body>
7  <?php
8
9  $Number=$_GET ["Fno"];
10
11
12 $con = mysqli_connect("127.0.0.1","root","root","furniture store",3306);
13
14 if ($con->connect_error) {
15     die("Connet failed" . $con->connect_error);
16 }
17
18 $sql = "SELECT * FROM Furniture WHERE Fno='$Number'";
19
20
21
22 $result = $con->query($sql);
23
24 if ($result->num_rows > 0) {
25     while($row = $result->fetch_assoc()) {
26         echo "<br> Number: ". $row["Fno"]. " - Name: ". $row["Fname"]." - Brand: ".
27         $row["Brand"]." - Category: ". $row["Category"]." - Unit_price: ". $row["Unit_price"];
28     }
29 } else {
30     echo "No result";
31 }
32 mysqli_close($con);
33
34 ?>
35 </body>
36 </html>
```

c) Update furniture

1. When we need to update a furniture's unit price, input the price into the database.



2. The webpage shows update successfully after we click update.



Source Code for update:

updatefurniture.html

/Applications/MAMP/htdocs/updatefurniture.html

```
<html>
<form action="updatefurniture.php" method="GET">
Number: <input type="text" name="Fno" />
Unit_price: <input type="text" name="Unit_price" />
<button type="submit">Update</button>
</form>
</html>
```

updatefurniture.php

ications/MAMP/htdocs/updatefurniture.php

```
<html>
<head>
<title>Update</title>
</head>

<body>
<?php

$Number=$_GET ["Fno"];
$Unit_price=$_GET ["Unit_price"];

$con = mysqli_connect("127.0.0.1","root","root","furniture store",3306);

if ($con->connect_error) {
    die("Connet failed" . $con->connect_error);
}

$sql = "UPDATE Furniture SET Unit_price='$Unit_price'
        WHERE Fno='$Number'";

if ($con->query($sql) === TRUE) {
    echo "Upate successfully.";
} else {
    echo "Update failed.";
}


mysqli_close($con);

?>
</body>
</html>
```

d) Delete furniture

When we need to delete a furniture, the database system will show all the furniture information first, then we input the number of furniture to be deleted so that we can delete this furniture.

Number: 1 - Name: JOKKMOKK - Brand: IKEA - Category: Table - Unit_price: 123
Number: 2 - Name: GLADOM - Brand: IKEA - Category: Table - Unit_price: 29.99
Number: 3 - Name: ARHOLMA - Brand: IKEA - Category: Sofa - Unit_price: 350

Delete successfully.

Source code for deletion:

delete.php

/Applications/MAMP/htdocs/delete.php

```php
<html>
<form action="deletefurniture.php" method="GET">
Number: <input type="text" name="Fno" />
<button type="submit">Delete</button>
</form>
<?php

$con = mysqli_connect("127.0.0.1","root","root","furniture store",3306);

if ($con->connect_error) {
    die("Connet failed" . $con->connect_error);
}

$sql = "SELECT * FROM Furniture";



$result = $con->query($sql);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<br> Number: ". $row["Fno"]. " - Name: ". $row["Fname"]." - Brand: ".
        $row["Brand"]." - Category: ". $row["Category"]." - Unit_price: ". $row["Unit_price"];
    }
} else {
    echo "No result";
}
mysqli_close($con);

?>
</html>
```

deletefurniture.php

/Applications/MAMP/htdocs/deletefurniture.php

```php
<html>
<head>
<title>Delete</title>
</head>

<body>
<?php

$Number=$_GET ["Fno"];
$Unit_price=$_GET ["Unit_price"];


$con = mysqli_connect("127.0.0.1","root","root","furniture store",3306);

if ($con->connect_error) {
    die("Connet failed" . $con->connect_error);
}

$sql = "DELETE FROM Furniture WHERE Fno='$Number'";

if ($con->query($sql) === TRUE) {
    echo "Delete successfully.";
} else {
    echo "Delete failed.";
}


mysqli_close($con);

?>
</body>
</html>
```