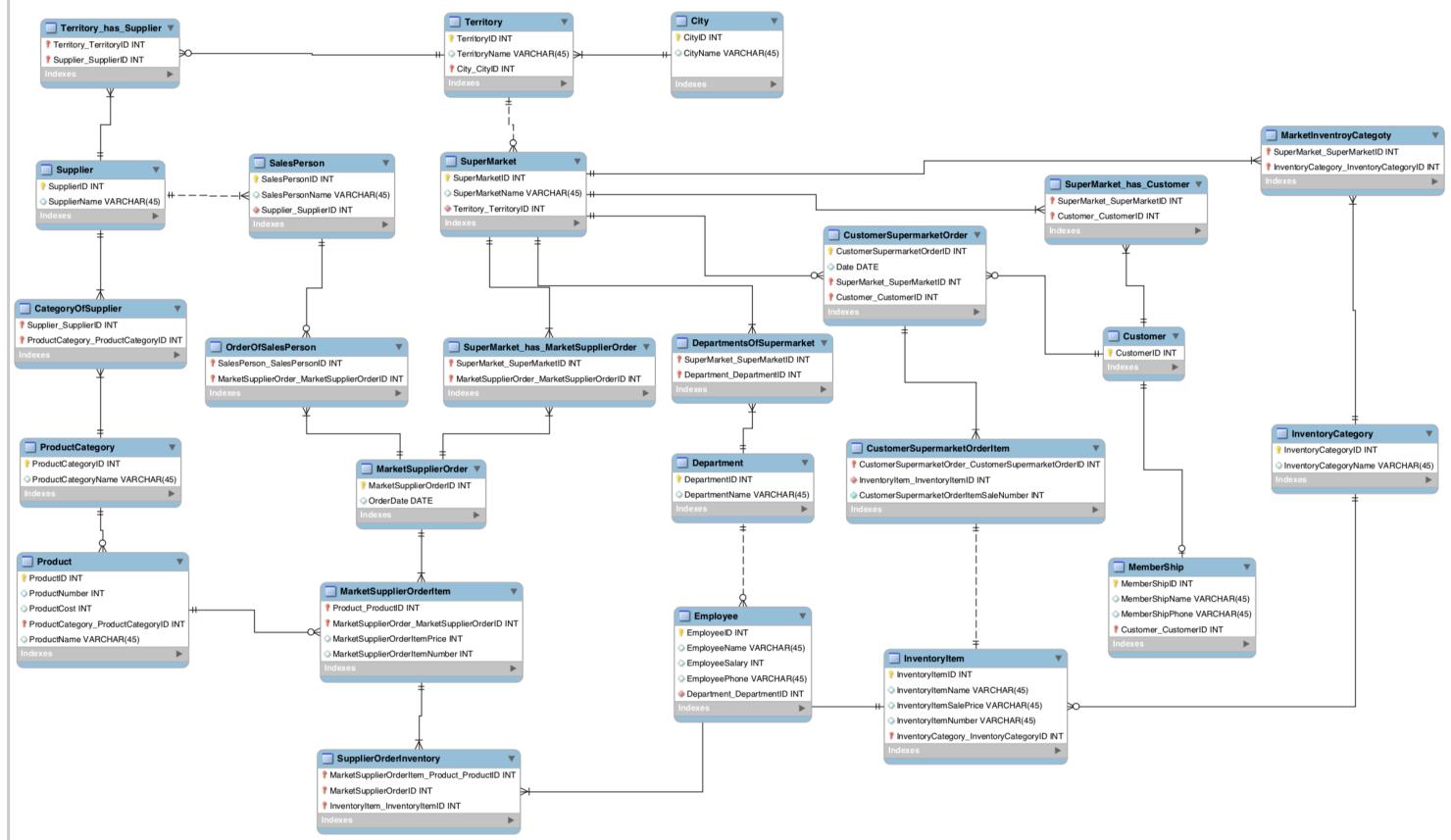


Supermarket Management System

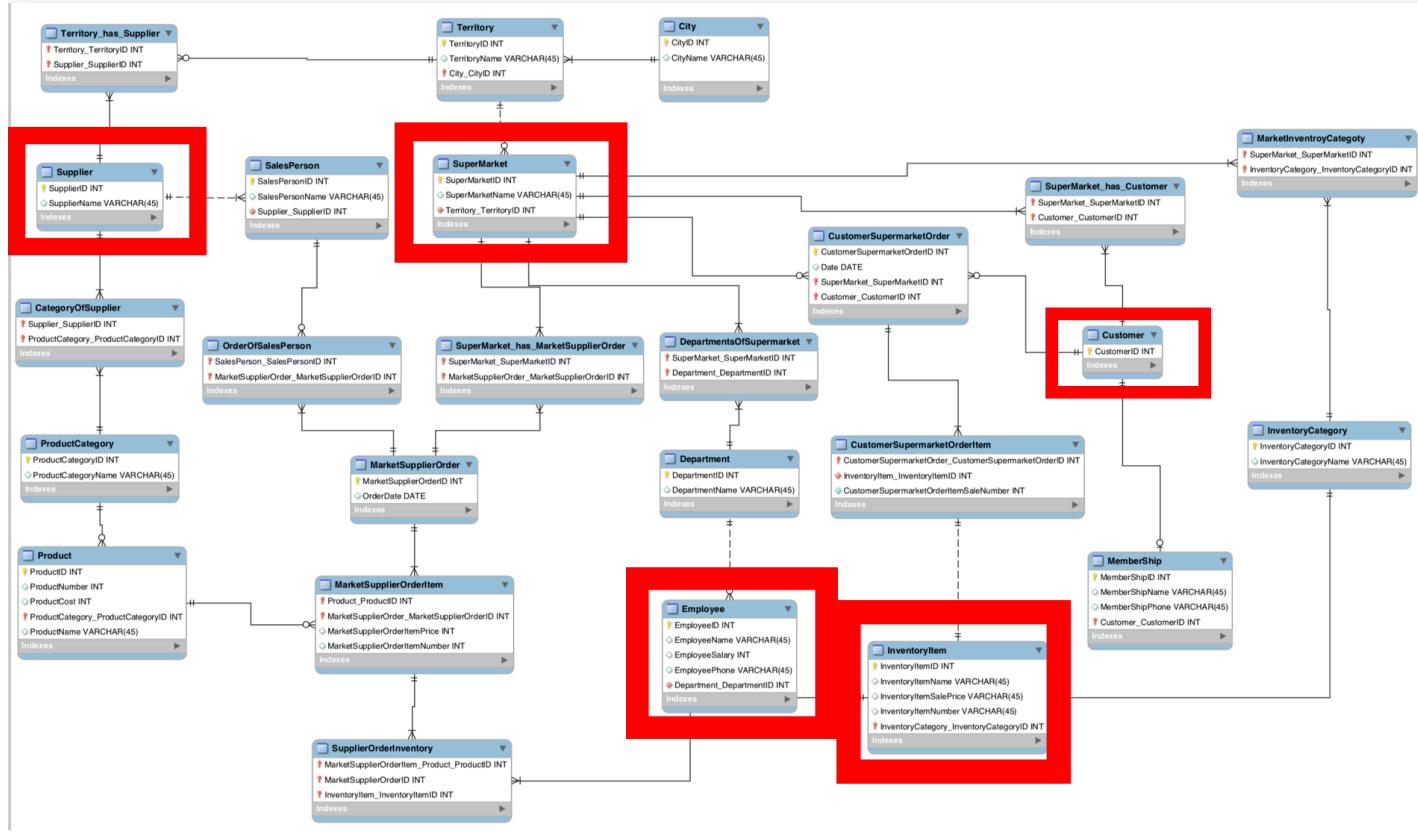
1. Proposal

This is a supermarket management system to manage the SUPPLIER, INVENTORY and CUSTOMER of the supermarket. The supermarket first need to bought PRODUCTS from different suppliers through sales person and the products will be the inventory items of the supermarket. Then the supermarket will SELL the inventory items to different customers to complete the trade.

2. ER Diagram



3.Main Tables



One City must have at least one Territory

One Territory can have but not must have many Supermarkets

One Territory can have many Suppliers and one Supplier can serve many Territories

One Supplier has many Sales Person to sell the products to Supermarkets.

One Supplier has many Categories of the products and one Category has many Products in it.

One Supermarket has many Departments for different responsibilities and in one Department there will be many Employees.

One Supermarket has many Customers and Some of the Customers will become the Membership so we can have the information for them.

One Supermarket has many Inventory Categories and each Category can belong to many Supermarkets.

One Inventory Category can have many Inventory Item

4.Relations

1)Supplier and Supermarket

Supplier sells the products by Sales Person to Supermarket.

One Supermarket can have many Orders with many Sales Person. At the same time, one Sales Person can also sell many Products to many Supermarkets.

There are two kinds of Orders. One can be seen from Sales Person aspect, the other one can be seen from Supermarket aspect. So they can check and manage their own orders separately.

One Order must have one or more Order Item.

For the Order Item, from the Supplier aspect, one Order Item will be one Product of the Supplier.

From the Supermarket aspect, after bought these Order Item, they will become the Inventory Item of the Supermarket

2)Supermarket and Customer

One Supermarket can have Orders with many Customers and one Customer can also have Orders with many different Supermarkets.

One Order must have at least one Order Item, and actually, the Order Item is the Inventory Item of the Supermarket.

5.Views

1) show all the Customers who has any Order of any Product

```
-- create a view that show all the customers who has any order of any product
create view product_customer as
select CustomerID,InventoryItemName as ItemName,CustomerSupermarketOrderItemSaleNumber as ItemNumber,
CustomerSupermarketOrderID as OrderID,SuperMarketName,InventoryItemSalePrice*CustomerSupermarketOrderItemSaleNumber as TotalPrice
from
Customer,CustomerSuperMarketOrder,SuperMarket,CustomerSuperMarketOrderItem,InventoryItem
where
Customer.CustomerID=CustomerSuperMarketOrder.Customer_CustomerID
and CustomerSuperMarketOrder.SuperMarket_SuperMarketID=SuperMarket.SuperMarketID
and CustomerSuperMarketOrder.CustomerSupermarketOrderID=
CustomerSuperMarketOrderItem.CustomerSuperMarketOrder_CustomerSupermarketOrderID
and CustomerSuperMarketOrderItem.InventoryItem_InventoryItemID=InventoryItem.InventoryItemID
;

select * from product_customer;
-- drop view product_customer;

-- select the specific customer's order
select ItemName,ItemNumber,OrderID,TotalPrice
from
product_customer
where CustomerID=1;

-- select specific order's total price
select OrderID,sum(TotalPrice) as OrderTotalPrice
from
product_customer
where OrderID=1;
```

```

14 ;
15
16 • select * from product_customer;
17 -- drop view product_customer;
18
19

```

100% 11:16

Result Grid



Filter Rows:

Search

Export:



CustomerID	ItemName	ItemNumber	OrderID	SuperMarketName	TotalPrice
1	MarketStoryBook	1	1	StarMarket	25
1	MarketNoteBook	2	1	StarMarket	68
1	MarketIPhone8	3	1	StarMarket	3000
1	MarketCannonM5	1	1	StarMarket	600
2	MarketStoryBook	1	2	StarMarket	25
2	MarketIPhone8	4	2	StarMarket	4000
3	MarketCannonM5	1	3	Walmart	600
3	MarketIPhone8	2	3	Walmart	2000
4	MarketCannonM5	3	4	Walmart	1800
4	MarketSonyA7	1	4	Walmart	10
5	MarketStoryBook	2	5	Target	50
5	MarketChildre...	1	5	Target	11
5	MarketChildre...	4	6	Target	44
6	MarketStoryBook	2	7	StopShop	50
1	MarketStoryBook	3	8	StarMarket	75
1	MarketNoteBook	4	8	StarMarket	136
1	MarketStoryBook	2	8	StarMarket	50

-- Select specific customer's Order

```

19 -- select the specific customer's order
20 • select ItemName,ItemNumber,OrderID,TotalPrice
21 from
22 product_customer
23 where CustomerID=1;
24

```

100% 11:23

Result Grid



Filter Rows:

Search

Export:



ItemName	ItemNumber	OrderID	TotalPrice
MarketStoryBook	1	1	25
MarketNoteBook	2	1	68
MarketIPhone8	3	1	3000
MarketCannonM5	1	1	600
MarketStoryBook	3	8	75
MarketNoteBook	4	8	136
MarketStoryBook	2	8	50

-- Select specific order's Total Price

```
25 -- select specific order's total price
26 • select OrderID,sum(TotalPrice) as OrderTotalPrice
27 from
28 product_customer
29 where OrderID=2;
30
31
```

100% 20:26

Result Grid | Filter Rows: Search Export:

OrderID	OrderTotalPrice
2	4025

2) show all the suppliers' all the products

```
-- create a view that shows all the suppliers' all the products
create view prodcut_supplier as
select SupplierID,SupplierName,ProductCategoryName,ProductName
from Supplier, CategoryOfSupplier,ProductCategory,Product
where
Supplier.SupplierID=CategoryOfSupplier.Supplier_ID
and CategoryOfSupplier.ProductCategory_ID=ProductCategory.ProductCategoryID
and ProductCategory.ProductCategoryID=Product.ProductCategory_ID;
```

```
10 • select * from prodcut_supplier;
11
12
```

100% 92:7

Result Grid | Filter Rows: Search Export:

SupplierID	SupplierName	ProductCategoryName	ProductName
1	One	NoteBook	NoteBookOne
1	One	Phone	IPhone 8
1	One	Phone	SumSung A7
1	One	Phone	SumSung A8
1	One	Camera	Canon M5
1	One	Camera	Sony A7
1	One	Milk	ChildrenMilk
1	One	Milk	CaMilk
1	One	Milk	NoSugarMilk
2	Two	StoryBook	StoryBookOne
2	Two	NoteBook	NoteBookOne
3	Three	StoryBook	StoryBookOne
3	Three	NoteBook	NoteBookOne
3	Three	Camera	Canon M5
3	Three	Camera	Sony A7

-- Select all the products of specific supplier

```
12 -- select all the products of specific supplier
13 • select ProductCategoryName,ProductName
14   from prodcut_supplier
15   where SupplierID=1;
16
```

Result Grid | Filter Rows: Search | Export:

ProductCategoryName	ProductName
NoteBook	NoteBookOne
Phone	IPhone 8
Camera	Canon M5
Camera	Sony A7
Milk	ChildrenMilk
Milk	CaMilk
Milk	NoSugarMilk
Phone	SumSung A7
Phone	SumSung A8

-- Select distinct products of all supplier

```
17 -- select distinct products of all supplier
18 • select distinct ProductName
19   from prodcut_supplier;
20
```

Result Grid | Filter Rows: Search | Export:

ProductName
StoryBookOne
NoteBookOne
IPhone 8
Canon M5
Sony A7
ChildrenMilk
CaMilk
NoSugarMilk
SumSung A7
SumSung A8

6.Procedures

1)Calculate the Total Price of the Market Supplier Order according to Order ID

```
1 -- calculate the total price of the MarketSupplierOrder according to OrderID
2 delimiter $
3 • create procedure product_pricing
4   (
5     in orderID int,
6     out priceSum int
7   )
8
9   begin
10  select sum(MarketSupplierOrderItemPrice*MarketSupplierOrderItemNumber)
11    from MarketSupplierOrderItem
12   where
13     MarketSupplierOrder_MarketSupplierOrderID=orderID
14   into priceSum;
15   end
16 $
```

```

17
18 delimiter ;
19 • select * from MarketSupplierOrderItem;
20 • call product_pricing(1,@sum);
21 • select @sum;

```

Result Grid | Filter Rows: Search | Edit: Export/Import:

Product_ProductID	MarketSupplierOrder_MarketSupplierOrderID	MarketSupplierOrderItemPrice	MarketSupplierOrderite...
1	1	20	30
1	2	22	45
1	6	5	67
1	8	25	5
2	1	30	25
2	2	33	66
3	3	980	45
4	3	560	77
4	4	570	89
5	5	7	13
6	7	4	78

```

18 delimiter ;
19 • select * from MarketSupplierOrderItem;
20 • call product_pricing(1,@sum);
21 • select @sum;

```

Result Grid | Filter Rows: Search | Export:

@sum
1350

2)Check specific day of customer 's order and order details

```

1 -- check specific day of customer 's order and order details
2
3 delimiter $
4 • create procedure customer_date
5 (
6   in orderDate date,
7   out OrderID int,
8   out totalPriceOfDate int
9 )
10
11 begin
12
13   create view OrderDetailsDate as
14     select CustomerSupermarketOrderID, CustomerSupermarketOrder.Date, CustomerID, InventoryItemName,
15       InventoryItemSalePrice, InventoryItemNumber,sum(InventoryItemSalePrice*InventoryItemNumber) as totalPrice
16     FROM
17     CustomerSupermarketOrder,CustomerSupermarketOrderItem,InventoryItem,Customer
18     where
19     CustomerSupermarketOrder.CustomerSupermarketOrderID=CustomerSupermarketOrderItem.CustomerSupermarketOrder_CustomerSupermarketOrderID
20     and
21     CustomerSupermarketOrder.Customer_CustomerID=Customer.CustomerID
22     and
23     CustomerSupermarketOrderItem.InventoryItem_InventoryItemID=InventoryItem.InventoryItemID;
24
25
26   select CustomerSupermarketOrderID from OrderDetailsDate
27   where Date=orderDate
28   limit
29   into OrderID;
30
31   select totalPrice from OrderDetailsDate
32   where Date=orderDate
33   limit
34   into totalPriceOfDate;
35
36 end
37
38

```

```

37   $ 
38 
39   delimiter ;
40 • drop procedure customer_date;
41 • drop view OrderDetailsDate;
42 • select * from OrderDetailsDate;

```

Result Grid | Filter Rows: Search | Export:

CustomerSuper...	Date	CustomerID	InventoryItemNa...	InventoryItemSa...	InventoryItemNu...	totalPrice
1	2017-05-12	1	MarketStoryBook	25	12	255188

```

43 
44 -- show create procedure customer_date;
45 • call customer_date('2017-05-12',@OrderID,@totalPriceOfDate);
46 • select @OrderID,@totalPriceOfDate;
47 

```

Result Grid | Filter Rows: Search | Export:

@OrderID	@totalPriceOfDate
1	255188

7.Triggers

1) Record Product Information if it is updated

```

1 • create table product_update
2 (
3   productUpdateID INT NOT NULL,
4   productUpdateDate date
5 );
6 
7 delimiter $
8 • create trigger tr_productUpdate
9 before update on Product
10 for each row
11 
12 begin
13   insert into product_update
14   set productUpdateID=OLD.ProductID,
15   productUpdateDate=now();
16 end
17 $

```

```

23 update Product
24 set ProductNumber=150
25 where
26 ProductID=1;
27
28 • select * from product_update;

```

100% 10:28

Result Grid Filter Rows: Search Export

productUpdateID	productUpdate...
1	2017-12-04
1	2017-12-10

2) Record Employee Information if he/she left the supermarket

```

1 • create table leave_employee
2 (
3     EmployeeID INT NOT NULL,
4     EmployeeName VARCHAR(45) NULL,
5     EmployeeLeaveDate date,
6     EmployeeStatus VARCHAR(45) NULL
7 );
8
9 delimiter $
10 • create trigger tr_leaveEmployee
11 before delete on Employee
12 for each row
13
14 begin
15     insert into leave_employee
16         set EmployeeID=OLD.EmployeeID,
17             EmployeeName=OLD.EmployeeName,
18             EmployeeLeaveDate=now(),
19             EmployeeStatus='Leave';
20     end
21 $
22

```

```

24 delimiter ;
25 • select * from Employee;
26
27 • INSERT INTO Employee ('EmployeeID', `EmployeeName`, `EmployeeSalary`, `EmployeePhone`, `Department_DepartmentID`)
28 VALUES (9, 'Mengting Yang', 5000, '857-209-1386', 1);
29
30 • INSERT INTO Employee ('EmployeeID', `EmployeeName`, `EmployeeSalary`, `EmployeePhone`, `Department_DepartmentID`)
31 VALUES (10, 'Chiwen Shi', 6000, '857-744-1222', 2);
32
33 • delete from Employee where EmployeeID=9;
34 • delete from Employee where EmployeeID=10;
35 • select * from leave_employee;

```

100% 13:35

Result Grid Filter Rows: Search Export

EmployeeID	EmployeeName	EmployeeLeave...	EmployeeStatus
9	Mengting Yang	2017-12-04	Leave
10	Chiwen Shi	2017-12-10	Leave

3) Reduce the Inventory Number if Customer buy new things

```
3 delimiter $  
4 • create trigger tr_productOrder  
5 after insert on CustomerSupermarketOrderItem  
6 for each row  
7  
8 begin  
9 update InventoryItem  
10 set InventoryItem.InventoryItemNumber=InventoryItem.InventoryItemNumber-new.CustomerSupermarketOrderItemSaleNumber  
11 where  
12 InventoryItem.InventoryItemID=new.InventoryItem_InventoryItemID;  
13 end  
14 $  
15  
16 delimiter ;  
17
```

27
28 • select * from InventoryItem;
29
30

100% 34:30

Result Grid Filter Rows: Search Edit: Export/Import:

InventoryItemID	InventoryItemNa...	InventoryItemSa...	InventoryItemNumber	InventoryCatego...
1	MarketStoryBook	25	12	1
2	MarketNoteBook	34	24	2
3	MarketPhone8	1000	44	3
4	MarketCannonM5	600	66	3
5	MarketSonyA7	10	45	3
6	MarketChildre...	11	23	4
NULL	NULL	NULL	NULL	NULL

23
24 • INSERT INTO `mydb`.`CustomerSupermarketOrderItem`
25 (`CustomerSupermarketOrder_CustomerSupermarketOrderID`, `InventoryItem_InventoryItemID`, CustomerSupermarketOrderItemSaleNumber)
26 VALUES (7, 2, 1);
27
28 • select * from InventoryItem;
29
30

100% 19:28

Result Grid Filter Rows: Search Edit: Export/Import:

InventoryItemID	InventoryItemNa...	InventoryItemSa...	InventoryItemNu...	InventoryCatego...
1	MarketStoryBook	25	12	1
2	MarketNoteBook	34	23	2
3	MarketPhone8	1000	44	3
4	MarketCannonM5	600	66	3
5	MarketSonyA7	10	45	3
6	MarketChildre...	11	23	4
NULL	NULL	NULL	NULL	NULL