SneakerDoodle Shop

DBMS by Farheen Zaman
BCS260

INTRODUCTION

This is the database requirement for the retail store SneakerDoodle.They sale shoes and sneakers accessories .Their store needs the database to manage their stock , inventory, sales,customer and their shipping information.

BUSINESS RULES

In this database should have all the store items and their quantity availability which needs to be updated as soon as a customer makes a purchase. For example, a purchase is done by a customer ,the product inventory quantity should be decreased based on the quantity purchased by the customer and it should also give the notification for the items which are below the specific limit in the store in order to restock asap. Each item should have a unique ID Also the database should store all the transaction information to get the complete idea of sales and profit.As this is the database deal with the retail store it will be used by the store manager to update the quantities and the items, and the employees while the time of billing and inventory manager to update the stock in the inventory or to create the order requests based on the stock available in the market.Each sale will have a unique Id as well in order to track sold item and purchase history etc.ach employee must be assigned a unique Id in order to track who has sold certain item.In addition, this database will store the customer information such as name and mobile number, email , address for subscriptions to give some special offers based on their purchases and give them store points which we can find as points redeemable.

Entities and Relationships

ITEM (ItemID,ItemName,ItemDescription,ItemType,QuantityOnHand,ItemPrice)

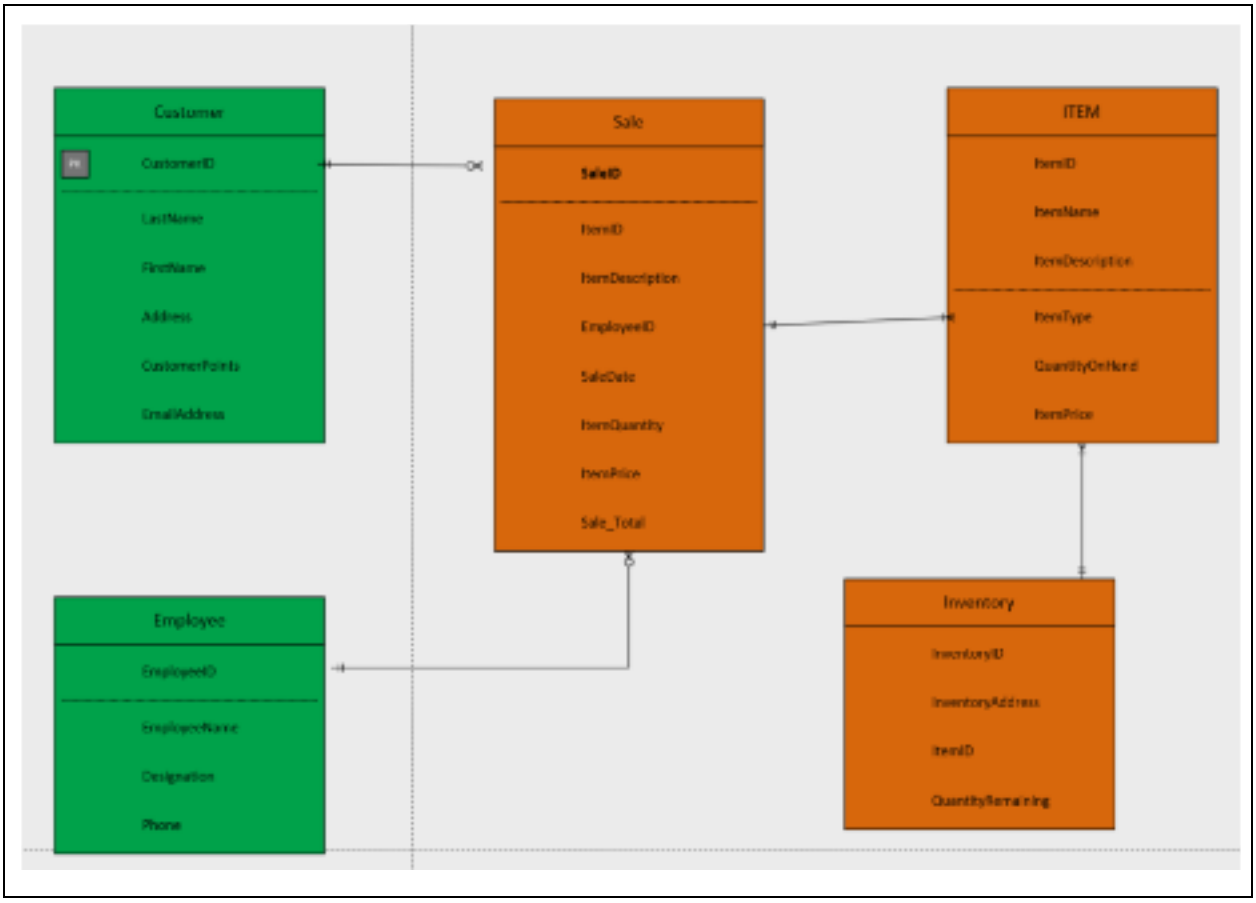
EMPLOYEE(EmployeeID,EmployeeName,Designation,Phone)

SALE (SaleID, ItemID,EmployeeID,ItemDescription,SaleDate,ItemQuantity,ItemPrice,Sale_Total)

CUSTOMER (CustomerID, CustomerLastName,CustomerFirstName, Email,CustomerPoints)

INVENTORY(InventoryID,InventoryAddress,ItemID,QuantityRemaining)

ERD Diagram


Customer
CustomerID
LastName
FirstName
Address
CustomerPoints
EmailAddress
Sale
SaleID
ItemID
ItemDescription
EmployeeID
SaleDate
ItemQuantity
ItemPrice
Sale_Total
ITEM
ItemID
ItemName
ItemDescription
ItemType
QuantityOnHand
ItemPrice
Employee
EmployeeID
EmployeeName
Designation
Phone
Inventory
InventoryID
InventoryAddress
ItemID
QuantityRemaining

Relation among Entities

Relationship		Cardinality	
Entity 1	Entity 2	Max	Min
Customer/strong	Sale	1:N	M-O
Employee/strong	sale	1:N	M-O
Sale/Weak	Item	1:N	M-M
Inventory/weak	Item	1:N	M-M

Sql to create entities

CUSTOMER ENTITY

```
CREATE TABLE CUSTOMER(CustomerID INT NOT NULL,
FirstName VARCHAR(25),
LastName VARCHAR(25),
EmaillAddress VARCHAR(75),
Customerpoints INT NOT NULL,
PRIMARY KEY(CustomerID));

CREATE TABLE ITEM (ItemID int primary key,
ItemName varchar (10) not null,
ItemDescription varchar(10),
ItemType varchar(10),
QuantityOnHand int not null,
ItemPrice decimal(5,2));

Create table EMPLOYEE (EmployeeID int primary key,
EmployeemName varchar(10) not null,
Designation varchar (10) not null);

create table INVENTORY(InventoryID int primary key,
InventoryAddress varchar(20) not null,
 ItemID int,
QuantityRemaining int,
foreign key(ItemID) references ITEM(ItemID));

Create table SALE (SaleID int primary key,
ItemID int not null,
SaleDate date not null,
Itemquantity int not null,
Itemprice decimal(5,2) UNIQUE,
EmployeeID int not null,
sale_total decimal(6,2),
foreign key(ItemID) references ITEM(ItemID),
foreign key(EmployeeID) references EMPLOYEE(EmployeeID));
```

Queries:

1) **DIsplay all the column from Item table**
   Select* From ITEM;

EmployeeID5Sale | SALE | ITEM | INVENTORY | Query1

ItemID	ItemName	ItemDescript	ItemPrice	QuantityOnH
1	Xshoe lace	Shoe lace	$51.00	125
2	Color poli	polisher	$53.00	125
3	Sgloss	polisher	$93.00	132
4	Cboad	body	$57.00	151
5	kicks	Sneakerkit	$19.00	186
6	Runner	Sneakrkit	$75.00	62
7	skyup	Scating kt	$11.00	145
8	Sking	Skee kit	$56.00	345
9	Flace	Lace kit	$38.00	101
10	Rom9	polisher	$59.00	122
				0

2)Customers listed who has more than 200 points
SELECT CustomerID, LastName,FirstName
From CUSTOMER
WHERE CustomerPoints >200;

CUSTOMER | Customerpoints>200

CustomerID	LastName	FirstName
7	Peterffy	Desiree
8	Andreou	Fidel
9	Prescote	Welbie
10	Davidofski	Betta
6	Fookes	Adelaide
1	Berthomier	Goldi
2	Petruskevich	Josselyn
4	Cawcutt	Mariann
5	Pitceathly	Felicle

3) list the sale done by employeeID 5

```
SELECT Sale.SaleID,Sale.Sale_total,Item.ItemName
From ITEM Inner join SALE
ON  ITEM.ItemID=SALE.ItemID
Where EmployeeID = 5 ;
```

Query1 | SALE | ITEM

SaleID	Sale_total	ItemName
1	$135.18	Xshoe lace
9	$456.00	Cboad
7	$352.00	skyup

4) Display all the Items with names ending with “kit” with less than 120 quantity on .

```
Select ItemID,ItemName,ItemDescription ,ItemPrice
```

```
 FROM ITEM
 where   QuantityOnHand < 120
and trim(ItemDescription) like '%kit';
```

EmployeeID5Sale | SALE | ITEM | INVENTORY | Query

ItemID	ItemName	ItemDescript	ItemPrice
6	Runner	Sneakrkit	$75.00
9	Flace	Lace kit	$38.00

5) Item that has less than 30 remaining stock.

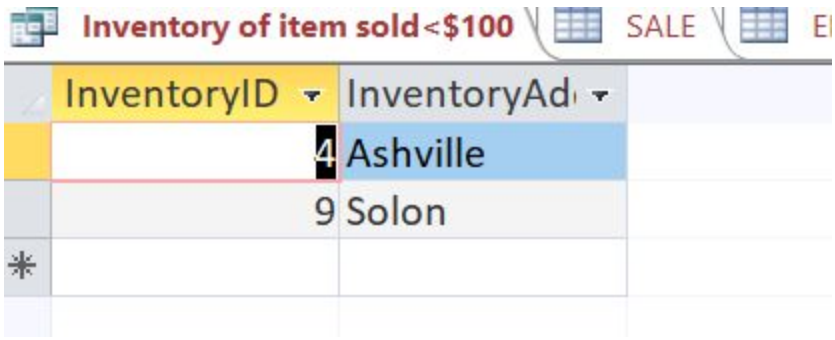
```
SELECT Item.ItemDescription,Inventory.InventoryID,Item.ItemName
From ITEM Inner join INVENTORY
ON  ITEM.ItemID=INVENTORY.ItemID
Where QuantityRemaining < 30 ;
```

EmployeeID5Sale | SALE | ITEM | Query

ItemDescript	InventoryID	ItemName
Shoe lace	1	Xshoe lace
Sneakerkit	5	kicks
Sneakrkit	6	Runner
Scating kt	7	skyup
Skee kit	9	Sking
Lace kit	10	Flace

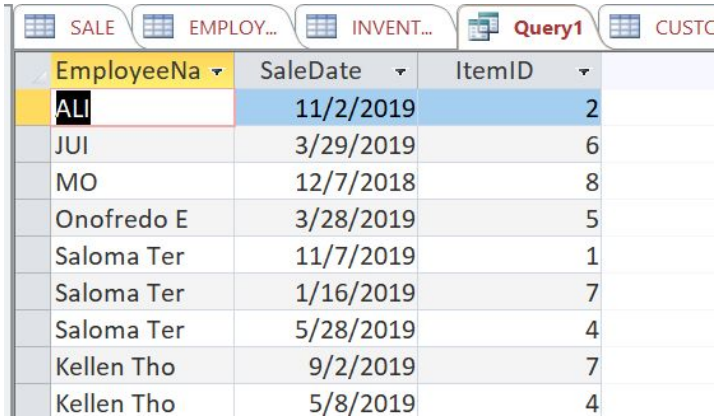
6) List the InventoryID and InventoryAddress of items that has Sale_total less than $100.

```
SELECT InventoryID,InventoryAddress
FROM Inventory
WHERE ItemID IN
        ( SELECT ItemID
from   Sale
WHERE Sale_total <100.);
```



7) Display the Names of all the employees and list theirx SaleDate and ItemID of the sold Item.

```
SELECT Employee.EmployeeName,Sale.SaleDate,Sale.ItemID
FROM  Employee
INNER JOIN Sale
ON Employee.EmployeeID = Sale.EmployeeID;
```



EmployeeNa	SaleDate	ItemID
ALI	11/2/2019	2
JUI	3/29/2019	6
MO	12/7/2018	8
Onofredo E	3/28/2019	5
Saloma Ter	11/7/2019	1
Saloma Ter	1/16/2019	7
Saloma Ter	5/28/2019	4
Kellen Tho	9/2/2019	7
Kellen Tho	5/8/2019	4