Pete’s Snack Corp Database

[redacted]

[redacted]

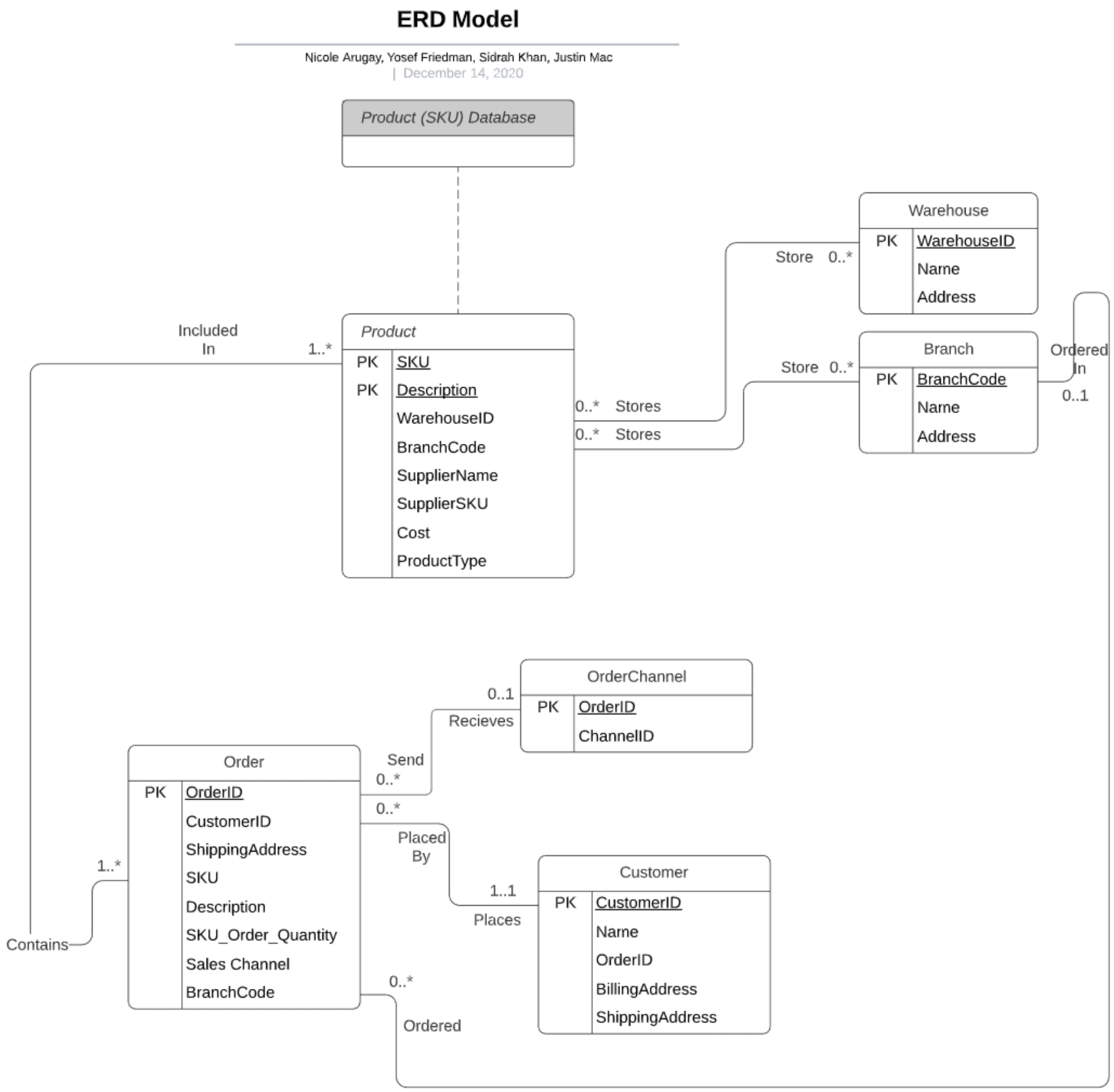
Business Scenario

The business we chose to focus on is a supplier named “Pete’s Snack Corp” that distributes snacks and beverages to offices. The company needs an operational database to monitor products which includes where they came from, where they are now, and where they were sold to. The company needs a database that allows its employees to view orders and supplier data. We created a relational model and displayed the associations for each part with an ERD. Our database consists of products, suppliers, customers, branches, warehouses, and orders.

Our company’s challenge includes recording accurate inventory levels for its popular products so that it always has enough stock to supply customers. It must be able to properly supply its various sales channels with the correct amount of inventory. The company must keep track of where each product is sold so customers can find their products.

Our company needs a database to track products, inventory levels, sales, customers, orders, and suppliers. This is integral to any business that needs to properly manage its retail operations. It will give visibility into replenishing low stock, forecast demand, profit per item, and allow the business to function properly.

By creating a product-based database, we will display inventory, purchases, and product attributes. This information will educate current/prospective employees and assist in managing supplier relationships and client (customer) relationships. Our product database will also assist in information navigation and in reducing redundant data.



ER Model Using UML Notation

Relationship Sentences:

1. One **order** can contain many **products**.
2. One **warehouse** can store many **products**.
3. One **branch** can store many **products**.
4. One **branch** can place many **orders**.
5. One **customer** can place many **orders.**
6. One **order** can only belong to one **customer.**
7. One **order channel** can receive multiple **orders.**

Converting ERD to a Relational Model

To begin with, we referenced our Entity Relation Diagram to create a set of relations. We analyzed all the relations in the model and distributed information to the most logical relation, accordingly. After we established the relational model, we reviewed which relations would require normalization. Such relations were identified if they had any partial dependency (i.e. Product relation). This was resolved by transforming the entity by employing splitting. The relationship was split into several components in order to achieve third normal form.

Channel_Order (ChannelID, OrderID(fk))
Product (SKU, Description, WarehouseID, BranchCode, SupplierName, SupplierSKU, Cost, ProductType)
Product_Order (OrderID(fk), SKU (fk))
Order (OrderID, CustomerID (fk), BranchCode (fk), SKU, ShippingAddress, Description, SKU_Order_QTY, Price, SalesChannel)
Customer (CustomerID, OrderID (fk), Name, ShippingAddress, BillingAddress)
Product_Warehouse (SKU (fk), Description (fk), WarehouseID (fk))
Warehouse (WarehouseID, Name, Address)
Product_Branch(SKU (fk) , Description (fk), BranchCode (fk))
Branch (BranchCode, Name, Address)

Normalization

Channel_Order (ChannelID, OrderID(fk))
FD1: ChannelID -> OrderID
Any key? Yes 1NF
Any partial dependency? No
Any transitive dependency? No

Product (SKU, Description, WarehouseID, BranchCode, SupplierName, SupplierSKU, Cost, ProductType)
FD1: SKU->SupplierSKU
FD2: Description-> ProductType, Cost, SupplierName, WarehouseID, BranchCode
FD3: ProductType -> Cost
FD4: SupplierSKU -> SupplierName
Any key? Yes 1NF
Any partial dependency? Yes, ProductType -> Cost, SupplierSKU-> SupplierName
Splitting:
R1(ProductType, Cost)

R2(SupplierSKU, SupplierName)
R3(SKU, Description, Cost, WarehouseID, BranchCode, SupplierSKU)

Checking any partial dependency? No
Any transitive dependency? No

Product_Order (OrderID(fk), SKU (fk))
FD1: OrderID -> SKU
Any key? Yes 1NF
Any partial dependency? No
Any transitive dependency? No

Order (OrderID, CustomerID (fk), BranchCode (fk), SKU, ShippingAddress, Description, SKU_Order_QTY, Price, SalesChannel)
FD1: OrderID -> Name, ShippingAddress, SKU, SKU_Order_Quantity
FD2: BranchCode -> SalesChannel
Any key? Yes 1NF
Any partial dependency? No
Any transitive dependency? No

Customer (CustomerID, OrderID (fk), Name, ShippingAddress, BillingAddress)
FD1: CustomerID -> Name, ShippingAddress, BillingAddress
Any key? Yes 1NF
Any partial dependency? No
Transitive dependency? No

Product_Warehouse (SKU (fk), Description (fk), WarehouseID (fk))
FD1: SKU -> Description, WarehouseID
Any key? Yes 1NF
Any partial dependency? No
Transitive dependency? No

Warehouse (WarehouseID, Name, Address)
FD1: WarehouseID -> Name, Address
Any key? Yes 1NF
Any partial dependency? No
Any transitive dependency? No

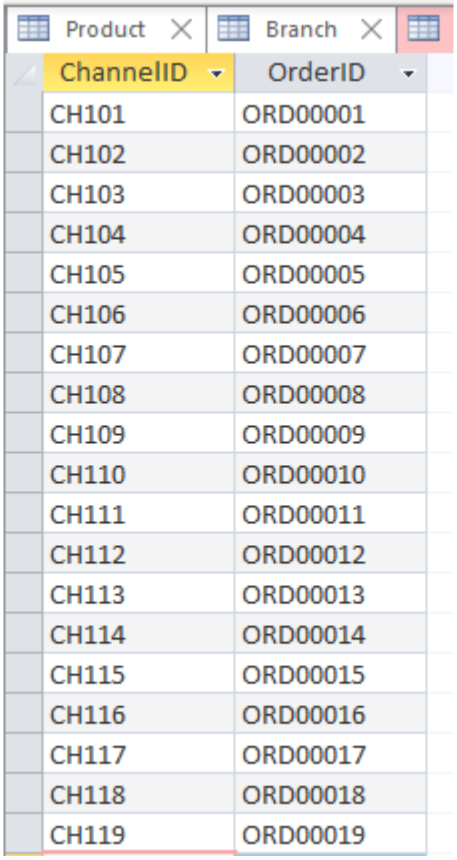
Product_Branch(SKU (fk) , Description (fk), BranchCode (fk))
FD1: SKU -> Description, BranchCode

Any key? Yes 1NF
Any partial dependency? No
Any transitive dependency? No

Branch (BranchCode, Name, Address)
FD1: BranchCode -> Name, Address
Any key? Yes 1NF
Any partial dependency? No
Any transitive dependency? No

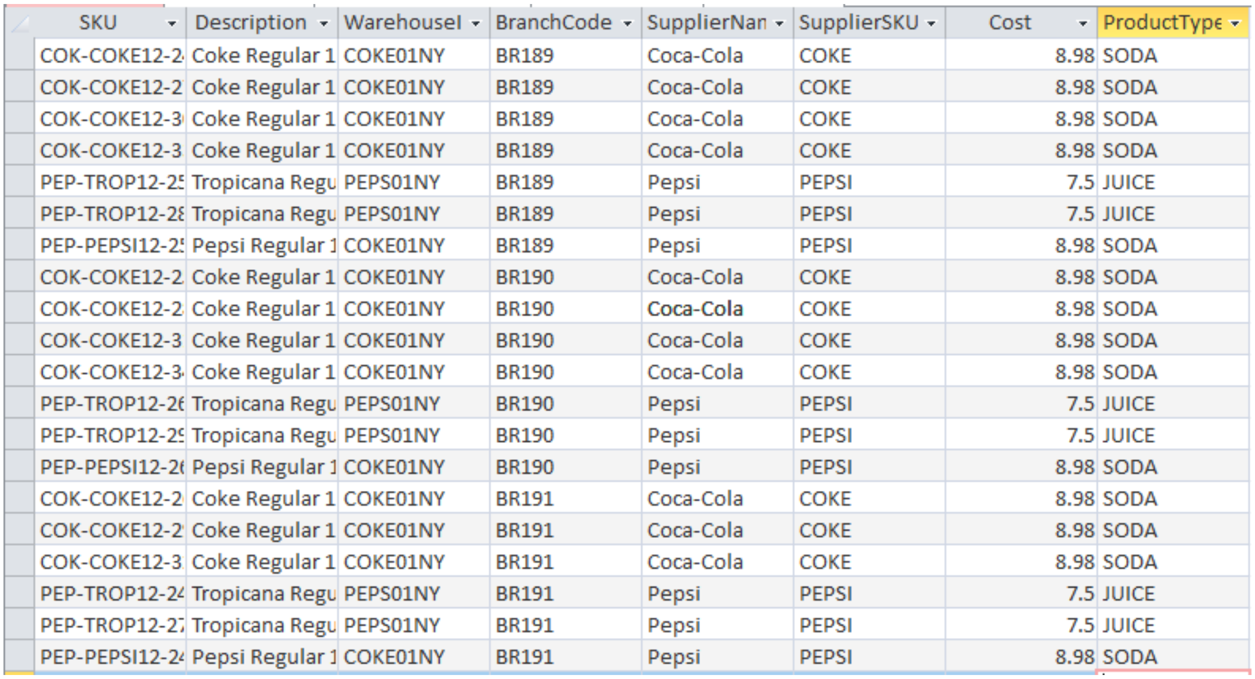
Creating Tables with SQL

```
CREATE TABLE Channel_Order
(
ChannelID VARCHAR(20) NOT NULL,
OrderID VARCHAR(20) NOT NULL
);

ALTER TABLE Channel_Order
  ADD CONSTRAINT pk1_OrderID
    PRIMARY KEY (OrderID);

ALTER TABLE Channel_Order
  ADD CONSTRAINT fk_OrderID
   FOREIGN KEY (OrderID)
     REFERENCES Orders (OrderID);

INSERT INTO Channel_Order VALUES ('CH101', 'ORD00001');
INSERT INTO Channel_Order VALUES ('CH102', 'ORD00002');
```



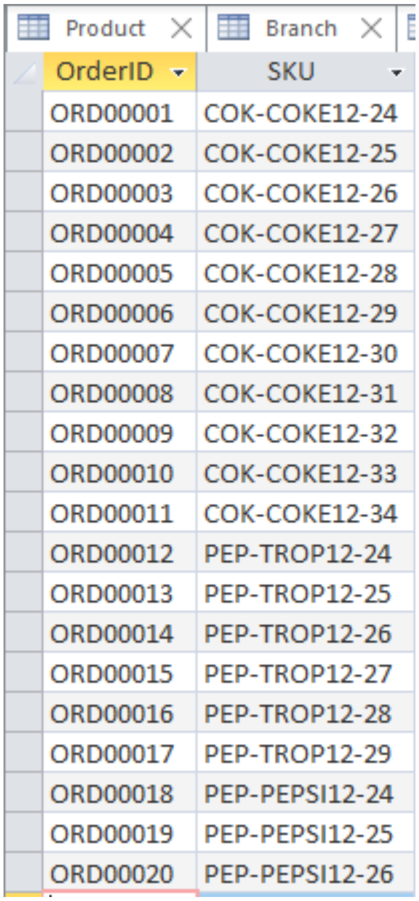
ChannelID	OrderID
CH101	ORD00001
CH102	ORD00002
CH103	ORD00003
CH104	ORD00004
CH105	ORD00005
CH106	ORD00006
CH107	ORD00007
CH108	ORD00008
CH109	ORD00009
CH110	ORD00010
CH111	ORD00011
CH112	ORD00012
CH113	ORD00013
CH114	ORD00014
CH115	ORD00015
CH116	ORD00016
CH117	ORD00017
CH118	ORD00018
CH119	ORD00019

```
CREATE TABLE Product(
    SKU VARCHAR(10) NOT NULL,
    Description VARCHAR(30) NOT NULL,
    WarehouseID VARCHAR(10) NOT NULL,
    BranchCode VARCHAR(10) NOT NULL,
    SupplierName VARCHAR(30) NOT NULL,
    SupplierSKU VARCHAR(30) NOT NULL,
    Cost VARCHAR(10) NOT NULL,
    ProductType VARCHAR(20) NOT NULL
);

ALTER TABLE Product
  ADD CONSTRAINT pk_product_sku
   PRIMARY KEY (SKU);

INSERT INTO Product VALUES ('COK-COKE12-24', 'Coke Regular 12 Oz Can 24 Cans',
'COKE01NY', 'BR189', 'Coca-Cola', 'COKE', '8.98', 'SODA');
INSERT INTO Product VALUES
('COK-SPRT12-24', 'Sprite Regular 12 Oz Can, 24 Cans', 'COKE02NY', 'BR190', 'Coca-Cola',
'COKE', '8.98', 'SODA');
```



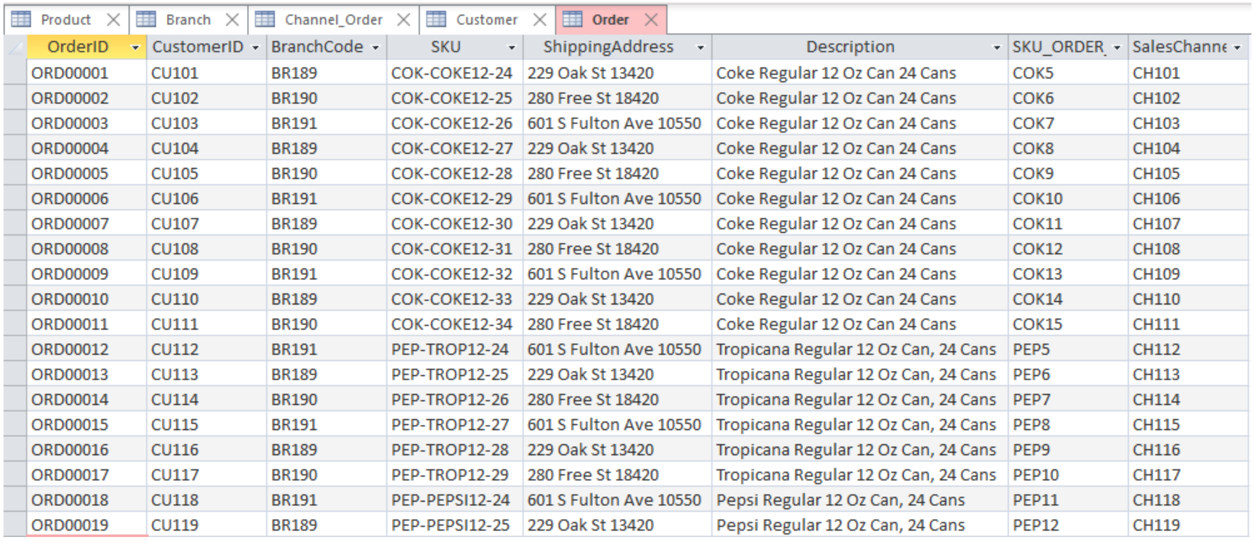
SKU	Description	Warehousel	BranchCode	SupplierNan	SupplierSKU	Cost	ProductType
COK-COKE12-24	Coke Regular 1	COKE01NY	BR189	Coca-Cola	COKE	8.98	SODA
COK-COKE12-27	Coke Regular 1	COKE01NY	BR189	Coca-Cola	COKE	8.98	SODA
COK-COKE12-30	Coke Regular 1	COKE01NY	BR189	Coca-Cola	COKE	8.98	SODA
COK-COKE12-3	Coke Regular 1	COKE01NY	BR189	Coca-Cola	COKE	8.98	SODA
PEP-TROP12-25	Tropicana Regu	PEPS01NY	BR189	Pepsi	PEPSI	7.5	JUICE
PEP-TROP12-28	Tropicana Regu	PEPS01NY	BR189	Pepsi	PEPSI	7.5	JUICE
PEP-PEPSI12-25	Pepsi Regular 1	COKE01NY	BR189	Pepsi	PEPSI	8.98	SODA
COK-COKE12-2	Coke Regular 1	COKE01NY	BR190	Coca-Cola	COKE	8.98	SODA
COK-COKE12-2	Coke Regular 1	COKE01NY	BR190	Coca-Cola	COKE	8.98	SODA
COK-COKE12-3	Coke Regular 1	COKE01NY	BR190	Coca-Cola	COKE	8.98	SODA
COK-COKE12-34	Coke Regular 1	COKE01NY	BR190	Coca-Cola	COKE	8.98	SODA
PEP-TROP12-26	Tropicana Regu	PEPS01NY	BR190	Pepsi	PEPSI	7.5	JUICE
PEP-TROP12-29	Tropicana Regu	PEPS01NY	BR190	Pepsi	PEPSI	7.5	JUICE
PEP-PEPSI12-26	Pepsi Regular 1	COKE01NY	BR190	Pepsi	PEPSI	8.98	SODA
COK-COKE12-2	Coke Regular 1	COKE01NY	BR191	Coca-Cola	COKE	8.98	SODA
COK-COKE12-2	Coke Regular 1	COKE01NY	BR191	Coca-Cola	COKE	8.98	SODA
COK-COKE12-3	Coke Regular 1	COKE01NY	BR191	Coca-Cola	COKE	8.98	SODA
PEP-TROP12-24	Tropicana Regu	PEPS01NY	BR191	Pepsi	PEPSI	7.5	JUICE
PEP-TROP12-27	Tropicana Regu	PEPS01NY	BR191	Pepsi	PEPSI	7.5	JUICE
PEP-PEPSI12-24	Pepsi Regular 1	COKE01NY	BR191	Pepsi	PEPSI	8.98	SODA

```
CREATE TABLE Product_Order
(
OrderID VARCHAR(20) NOT NULL,
SKU VARCHAR(20) NOT NULL
);

ALTER TABLE Product_Order
 ADD CONSTRAINT fk2_OrderID
  FOREIGN KEY (OrderID)
   REFERENCES Orders (OrderID);

INSERT INTO Product_Order VALUES ('O189', 'COK-COKE12-24');
INSERT INTO Product_Order VALUES ('O190', 'COK-COKE12-24');
```



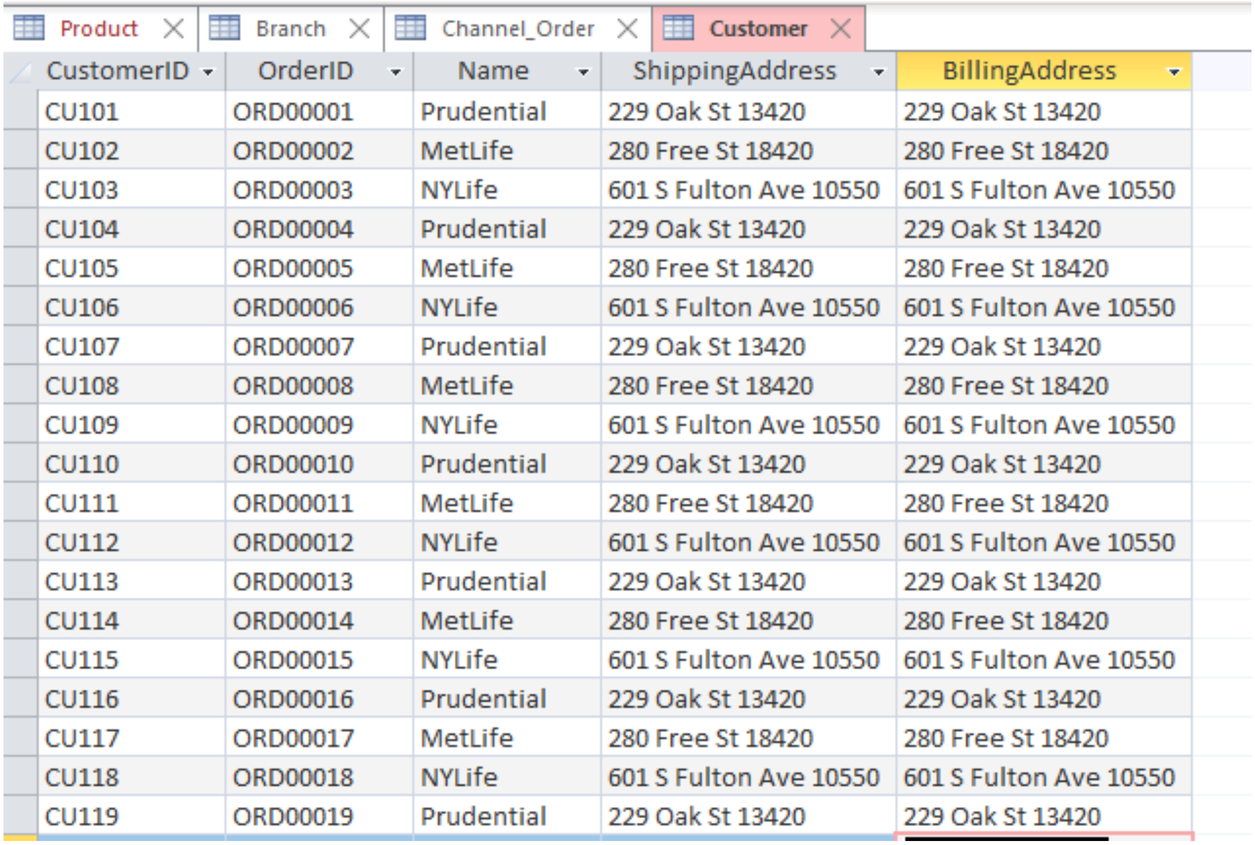
OrderID	SKU
ORD00001	COK-COKE12-24
ORD00002	COK-COKE12-25
ORD00003	COK-COKE12-26
ORD00004	COK-COKE12-27
ORD00005	COK-COKE12-28
ORD00006	COK-COKE12-29
ORD00007	COK-COKE12-30
ORD00008	COK-COKE12-31
ORD00009	COK-COKE12-32
ORD00010	COK-COKE12-33
ORD00011	COK-COKE12-34
ORD00012	PEP-TROP12-24
ORD00013	PEP-TROP12-25
ORD00014	PEP-TROP12-26
ORD00015	PEP-TROP12-27
ORD00016	PEP-TROP12-28
ORD00017	PEP-TROP12-29
ORD00018	PEP-PEPSI12-24
ORD00019	PEP-PEPSI12-25
ORD00020	PEP-PEPSI12-26

```
CREATE TABLE Orders(
   OrderID VARCHAR(10) NOT NULL,
   CustomerID VARCHAR(30) NOT NULL,
   BranchCode VARCHAR(30) NOT NULL,
   SKU VARCHAR(10) NOT NULL,
   ShippingAddress VARCHAR(30) NOT NULL,
   SalesChannel VARCHAR(30) NOT NULL,
   Description VARCHAR(30) NOT NULL
);

ALTER TABLE Orders
  ADD CONSTRAINT pk_OrderID
    PRIMARY KEY (OrderID);

INSERT INTO Orders VALUES ('ORD00001', 'CH190', 'BRANCHA', 'COK-COKE12-24', '229
Oak St 13420','Coke Regular 12 Oz Can 24 Cans');
INSERT INTO Orders VALUES  ('ORD00002', 'CH191', 'BRANCHB', 'COK-COKE12-24', '280
Oak St 18420','Coke Regular 12 Oz Can 24 Cans');
```



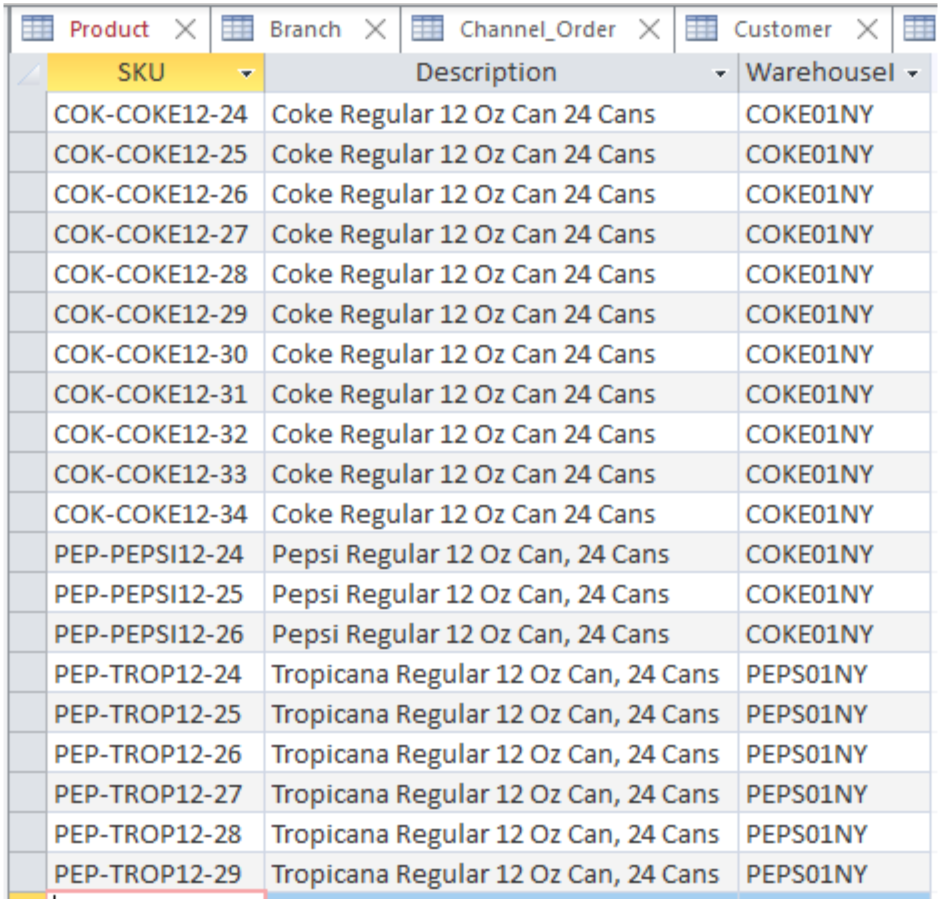
OrderID	CustomerID	BranchCode	SKU	ShippingAddress	Description	SKU_ORDER_	SalesChanne
ORD00001	CU101	BR189	COK-COKE12-24	229 Oak St 13420	Coke Regular 12 Oz Can 24 Cans	COK5	CH101
ORD00002	CU102	BR190	COK-COKE12-25	280 Free St 18420	Coke Regular 12 Oz Can 24 Cans	COK6	CH102
ORD00003	CU103	BR191	COK-COKE12-26	601 S Fulton Ave 10550	Coke Regular 12 Oz Can 24 Cans	COK7	CH103
ORD00004	CU104	BR189	COK-COKE12-27	229 Oak St 13420	Coke Regular 12 Oz Can 24 Cans	COK8	CH104
ORD00005	CU105	BR190	COK-COKE12-28	280 Free St 18420	Coke Regular 12 Oz Can 24 Cans	COK9	CH105
ORD00006	CU106	BR191	COK-COKE12-29	601 S Fulton Ave 10550	Coke Regular 12 Oz Can 24 Cans	COK10	CH106
ORD00007	CU107	BR189	COK-COKE12-30	229 Oak St 13420	Coke Regular 12 Oz Can 24 Cans	COK11	CH107
ORD00008	CU108	BR190	COK-COKE12-31	280 Free St 18420	Coke Regular 12 Oz Can 24 Cans	COK12	CH108
ORD00009	CU109	BR191	COK-COKE12-32	601 S Fulton Ave 10550	Coke Regular 12 Oz Can 24 Cans	COK13	CH109
ORD00010	CU110	BR189	COK-COKE12-33	229 Oak St 13420	Coke Regular 12 Oz Can 24 Cans	COK14	CH110
ORD00011	CU111	BR190	COK-COKE12-34	280 Free St 18420	Coke Regular 12 Oz Can 24 Cans	COK15	CH111
ORD00012	CU112	BR191	PEP-TROP12-24	601 S Fulton Ave 10550	Tropicana Regular 12 Oz Can, 24 Cans	PEP5	CH112
ORD00013	CU113	BR189	PEP-TROP12-25	229 Oak St 13420	Tropicana Regular 12 Oz Can, 24 Cans	PEP6	CH113
ORD00014	CU114	BR190	PEP-TROP12-26	280 Free St 18420	Tropicana Regular 12 Oz Can, 24 Cans	PEP7	CH114
ORD00015	CU115	BR191	PEP-TROP12-27	601 S Fulton Ave 10550	Tropicana Regular 12 Oz Can, 24 Cans	PEP8	CH115
ORD00016	CU116	BR189	PEP-TROP12-28	229 Oak St 13420	Tropicana Regular 12 Oz Can, 24 Cans	PEP9	CH116
ORD00017	CU117	BR190	PEP-TROP12-29	280 Free St 18420	Tropicana Regular 12 Oz Can, 24 Cans	PEP10	CH117
ORD00018	CU118	BR191	PEP-PEPSI12-24	601 S Fulton Ave 10550	Pepsi Regular 12 Oz Can, 24 Cans	PEP11	CH118
ORD00019	CU119	BR189	PEP-PEPSI12-25	229 Oak St 13420	Pepsi Regular 12 Oz Can, 24 Cans	PEP12	CH119

```
CREATE TABLE Customer(
   CustomerID VARCHAR(30) NOT NULL,
   OrderID VARCHAR(10) NOT NULL,
   Name VARCHAR(10) NOT NULL,
   ShippingAddress VARCHAR(30) NOT NULL,
   BillingAddress VARCHAR(30) NOT NULL
);

ALTER TABLE Customer
  ADD CONSTRAINT pk_CustomerID
    PRIMARY KEY (CustomerID);

INSERT INTO Customer VALUES ('C100', 'ORD00001', 'Jane Doe',  '229 Oak St, 13420',  '229
Oak St, 13420');
INSERT INTO Customer VALUES ('C102', 'ORD00002', 'John Smith',  '2324 Bell St, 10016',
'2324 Bell St, 10016');
```



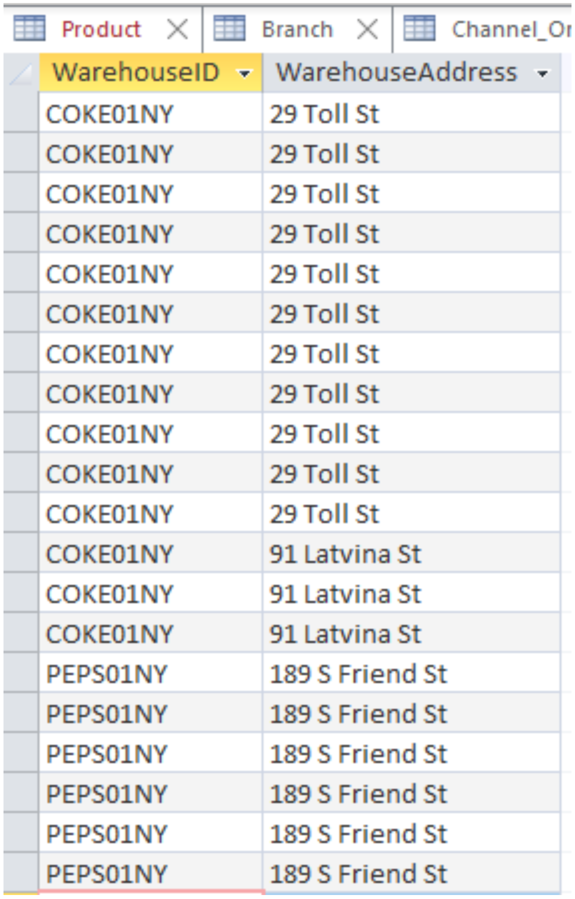
CustomerID	OrderID	Name	ShippingAddress	BillingAddress
CU101	ORD00001	Prudential	229 Oak St 13420	229 Oak St 13420
CU102	ORD00002	MetLife	280 Free St 18420	280 Free St 18420
CU103	ORD00003	NYLife	601 S Fulton Ave 10550	601 S Fulton Ave 10550
CU104	ORD00004	Prudential	229 Oak St 13420	229 Oak St 13420
CU105	ORD00005	MetLife	280 Free St 18420	280 Free St 18420
CU106	ORD00006	NYLife	601 S Fulton Ave 10550	601 S Fulton Ave 10550
CU107	ORD00007	Prudential	229 Oak St 13420	229 Oak St 13420
CU108	ORD00008	MetLife	280 Free St 18420	280 Free St 18420
CU109	ORD00009	NYLife	601 S Fulton Ave 10550	601 S Fulton Ave 10550
CU110	ORD00010	Prudential	229 Oak St 13420	229 Oak St 13420
CU111	ORD00011	MetLife	280 Free St 18420	280 Free St 18420
CU112	ORD00012	NYLife	601 S Fulton Ave 10550	601 S Fulton Ave 10550
CU113	ORD00013	Prudential	229 Oak St 13420	229 Oak St 13420
CU114	ORD00014	MetLife	280 Free St 18420	280 Free St 18420
CU115	ORD00015	NYLife	601 S Fulton Ave 10550	601 S Fulton Ave 10550
CU116	ORD00016	Prudential	229 Oak St 13420	229 Oak St 13420
CU117	ORD00017	MetLife	280 Free St 18420	280 Free St 18420
CU118	ORD00018	NYLife	601 S Fulton Ave 10550	601 S Fulton Ave 10550
CU119	ORD00019	Prudential	229 Oak St 13420	229 Oak St 13420

```
CREATE TABLE Product_Warehouse(
   SKU VARCHAR(10) NOT NULL,
   Description VARCHAR(30) NOT NULL,
   WarehouseID VARCHAR(30) NOT NULL
);

ALTER TABLE Product_Warehouse
  ADD CONSTRAINT pk_SKU
   PRIMARY KEY (SKU);

INSERT INTO Product_Warehouse VALUES ('COK-COKE12-24', 'Coke Regular 12 Oz Can 24 Cans', 'COKE01NY');
INSERT INTO Product_Warehouse VALUES ('COK-SPRT12-24', 'Sprite Regular 12 Oz Can, 24 Cans', 'COKE02NY');
INSERT INTO Product_Warehouse VALUES ('PEP-PEPSI12-24', 'Pepsi Regular 12 Oz Can, 24 Cans', 'PEPS01NY');
```



SKU	Description	WarehouseI
COK-COKE12-24	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-25	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-26	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-27	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-28	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-29	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-30	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-31	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-32	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-33	Coke Regular 12 Oz Can 24 Cans	COKE01NY
COK-COKE12-34	Coke Regular 12 Oz Can 24 Cans	COKE01NY
PEP-PEPSI12-24	Pepsi Regular 12 Oz Can, 24 Cans	COKE01NY
PEP-PEPSI12-25	Pepsi Regular 12 Oz Can, 24 Cans	COKE01NY
PEP-PEPSI12-26	Pepsi Regular 12 Oz Can, 24 Cans	COKE01NY
PEP-TROP12-24	Tropicana Regular 12 Oz Can, 24 Cans	PEPS01NY
PEP-TROP12-25	Tropicana Regular 12 Oz Can, 24 Cans	PEPS01NY
PEP-TROP12-26	Tropicana Regular 12 Oz Can, 24 Cans	PEPS01NY
PEP-TROP12-27	Tropicana Regular 12 Oz Can, 24 Cans	PEPS01NY
PEP-TROP12-28	Tropicana Regular 12 Oz Can, 24 Cans	PEPS01NY
PEP-TROP12-29	Tropicana Regular 12 Oz Can, 24 Cans	PEPS01NY

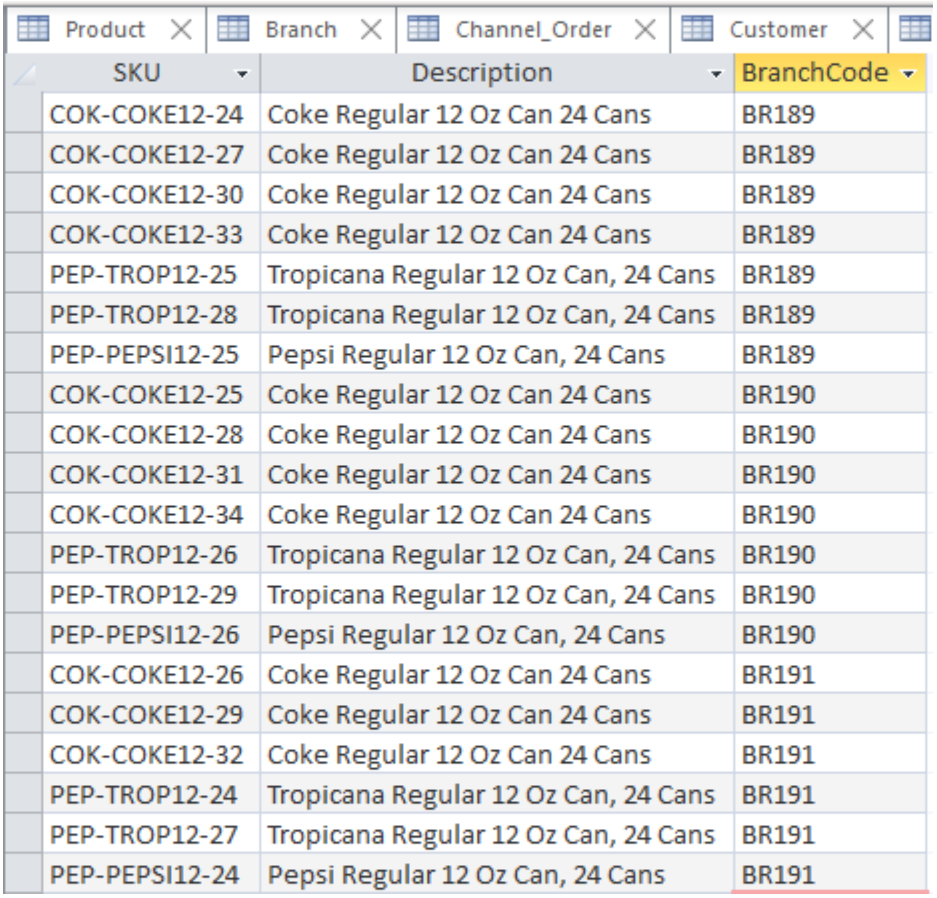
```
CREATE TABLE Warehouse
(
WarehouseID VARCHAR(10) NOT NULL,
WarehouseName VARCHAR(10) NOT NULL,
WarehouseAddress VARCHAR(10) NOT NULL
);
INSERT INTO Warehouse VALUES ('COKE01NY', 'Liberty Coca-Cola Beverages', '54-80
Borden Ave, 11378');
INSERT INTO Warehouse VALUES ('COKE02NY', 'Liberty Coca-Cola Beverages', '977 E
149th St, 10455');
INSERT INTO Warehouse VALUES ('PEPS01NY', 'Pepsi Cola-Co', '601 S Fulton Ave, 10550');
```



WarehouseID	WarehouseAddress
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	29 Toll St
COKE01NY	91 Latvina St
COKE01NY	91 Latvina St
COKE01NY	91 Latvina St
PEPS01NY	189 S Friend St
PEPS01NY	189 S Friend St
PEPS01NY	189 S Friend St
PEPS01NY	189 S Friend St
PEPS01NY	189 S Friend St
PEPS01NY	189 S Friend St

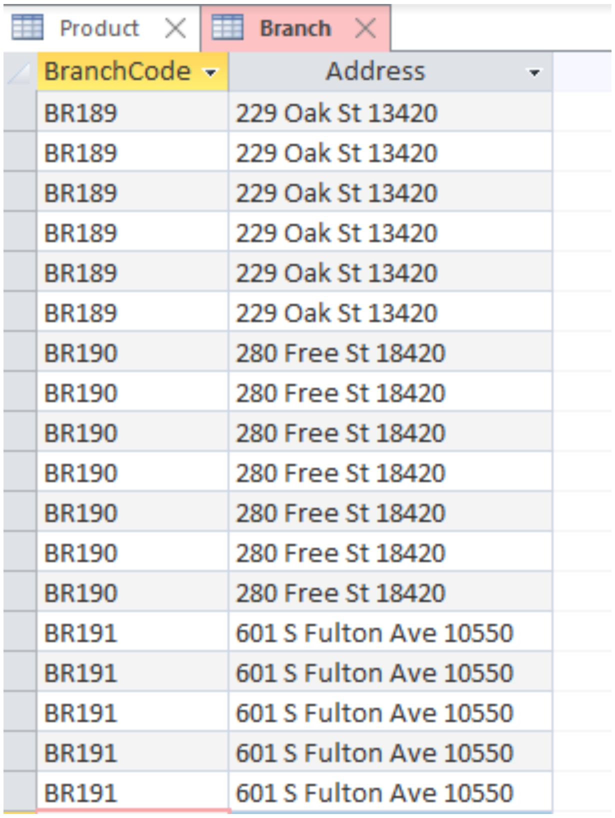
```
CREATE TABLE Product_Branch
(
SKU VARCHAR(30) NOT NULL,
Description VARCHAR(250),
BranchCode VARCHAR(10) NOT NULL
);

ALTER TABLE ProductBranch
  ADD CONSTRAINT fk2_SKU
    FOREIGN KEY (SKU)
      REFERENCES Product (SKU);

ALTER TABLE ProductBranch
  ADD CONSTRAINT fk_BranchCode
    FOREIGN KEY (BranchCode)
      REFERENCES Branch (BranchCode);


INSERT INTO ProductBranch VALUES ('COK-COKE12-24', 'Coke Regular 12 Oz Can, 24
Cans', 'BRANCHA');
```

```
INSERT INTO ProductBranch VALUES ('PEP-PEPSI12-24', 'Pepsi Regular 12 Oz Can, 24
Cans', 'BR189');
```



SKU	Description	BranchCode
COK-COKE12-24	Coke Regular 12 Oz Can 24 Cans	BR189
COK-COKE12-27	Coke Regular 12 Oz Can 24 Cans	BR189
COK-COKE12-30	Coke Regular 12 Oz Can 24 Cans	BR189
COK-COKE12-33	Coke Regular 12 Oz Can 24 Cans	BR189
PEP-TROP12-25	Tropicana Regular 12 Oz Can, 24 Cans	BR189
PEP-TROP12-28	Tropicana Regular 12 Oz Can, 24 Cans	BR189
PEP-PEPSI12-25	Pepsi Regular 12 Oz Can, 24 Cans	BR189
COK-COKE12-25	Coke Regular 12 Oz Can 24 Cans	BR190
COK-COKE12-28	Coke Regular 12 Oz Can 24 Cans	BR190
COK-COKE12-31	Coke Regular 12 Oz Can 24 Cans	BR190
COK-COKE12-34	Coke Regular 12 Oz Can 24 Cans	BR190
PEP-TROP12-26	Tropicana Regular 12 Oz Can, 24 Cans	BR190
PEP-TROP12-29	Tropicana Regular 12 Oz Can, 24 Cans	BR190
PEP-PEPSI12-26	Pepsi Regular 12 Oz Can, 24 Cans	BR190
COK-COKE12-26	Coke Regular 12 Oz Can 24 Cans	BR191
COK-COKE12-29	Coke Regular 12 Oz Can 24 Cans	BR191
COK-COKE12-32	Coke Regular 12 Oz Can 24 Cans	BR191
PEP-TROP12-24	Tropicana Regular 12 Oz Can, 24 Cans	BR191
PEP-TROP12-27	Tropicana Regular 12 Oz Can, 24 Cans	BR191
PEP-PEPSI12-24	Pepsi Regular 12 Oz Can, 24 Cans	BR191

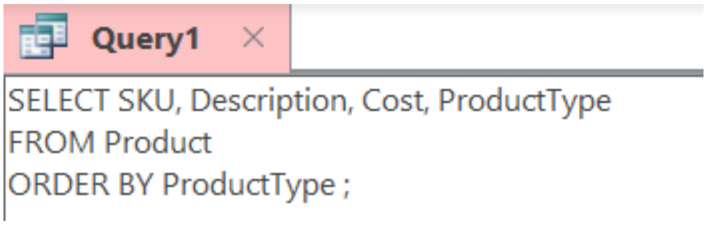
```
CREATE TABLE ProductBranch
(
SKU VARCHAR(30) NOT NULL,
Description VARCHAR(250),
BranchCode VARCHAR(10) NOT NULL
);

ALTER TABLE Branch
  ADD CONSTRAINT pk_BranchCode
    PRIMARY KEY (BranchCode);

INSERT INTO Branch VALUES ('BRANCHA', 'Liberty Coca-Cola Bottling Co', '400 Western
Ave, 10303');
INSERT INTO Branch VALUES('BRANCHB',  'Pepsi Bottling Group', '9701 Avenue D,
11236');
```



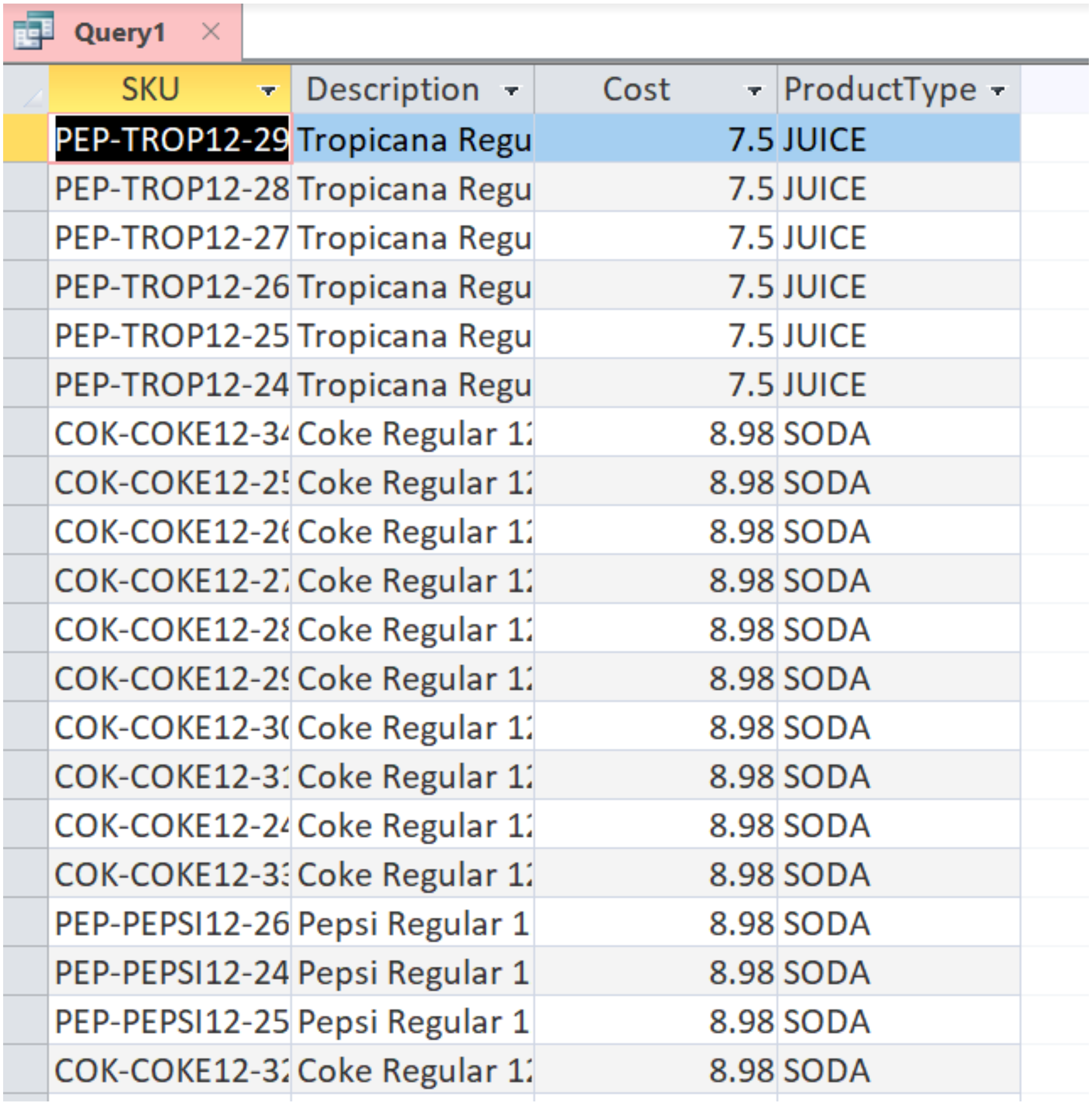
BranchCode	Address
BR189	229 Oak St 13420
BR189	229 Oak St 13420
BR189	229 Oak St 13420
BR189	229 Oak St 13420
BR189	229 Oak St 13420
BR189	229 Oak St 13420
BR190	280 Free St 18420
BR190	280 Free St 18420
BR190	280 Free St 18420
BR190	280 Free St 18420
BR190	280 Free St 18420
BR190	280 Free St 18420
BR190	280 Free St 18420
BR191	601 S Fulton Ave 10550
BR191	601 S Fulton Ave 10550
BR191	601 S Fulton Ave 10550
BR191	601 S Fulton Ave 10550
BR191	601 S Fulton Ave 10550

Scenarios

1. One use of the database would be to create a catalog sorted by any parameter (such as product type):

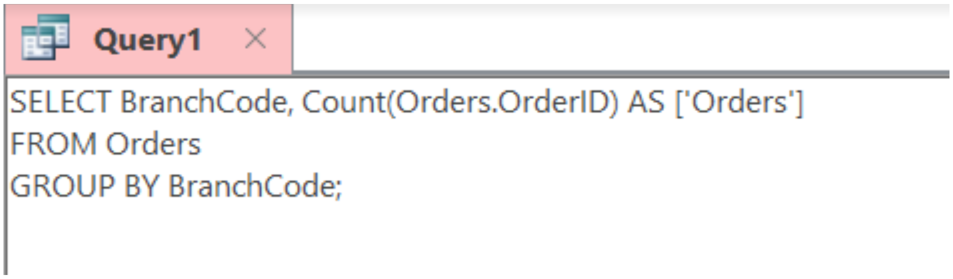


```
SELECT SKU, Description, Cost, ProductType
FROM Product
ORDER BY ProductType ;
```

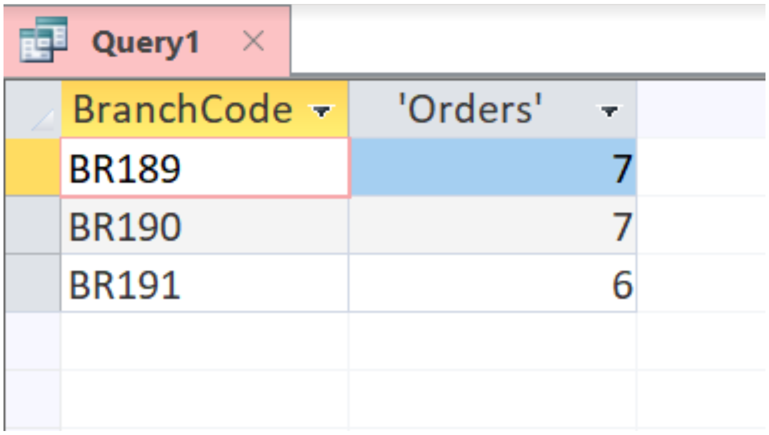


SKU	Description	Cost	ProductType
PEP-TROP12-29	Tropicana Regu	7.5	JUICE
PEP-TROP12-28	Tropicana Regu	7.5	JUICE
PEP-TROP12-27	Tropicana Regu	7.5	JUICE
PEP-TROP12-26	Tropicana Regu	7.5	JUICE
PEP-TROP12-25	Tropicana Regu	7.5	JUICE
PEP-TROP12-24	Tropicana Regu	7.5	JUICE
COK-COKE12-34	Coke Regular 12	8.98	SODA
COK-COKE12-25	Coke Regular 12	8.98	SODA
COK-COKE12-26	Coke Regular 12	8.98	SODA
COK-COKE12-27	Coke Regular 12	8.98	SODA
COK-COKE12-28	Coke Regular 12	8.98	SODA
COK-COKE12-29	Coke Regular 12	8.98	SODA
COK-COKE12-30	Coke Regular 12	8.98	SODA
COK-COKE12-31	Coke Regular 12	8.98	SODA
COK-COKE12-24	Coke Regular 12	8.98	SODA
COK-COKE12-33	Coke Regular 12	8.98	SODA
PEP-PEPSI12-26	Pepsi Regular 1	8.98	SODA
PEP-PEPSI12-24	Pepsi Regular 1	8.98	SODA
PEP-PEPSI12-25	Pepsi Regular 1	8.98	SODA
COK-COKE12-32	Coke Regular 12	8.98	SODA

2. Another scenario would be if a salesperson wanted to know how many orders we get from a certain branch they can do execute the following:

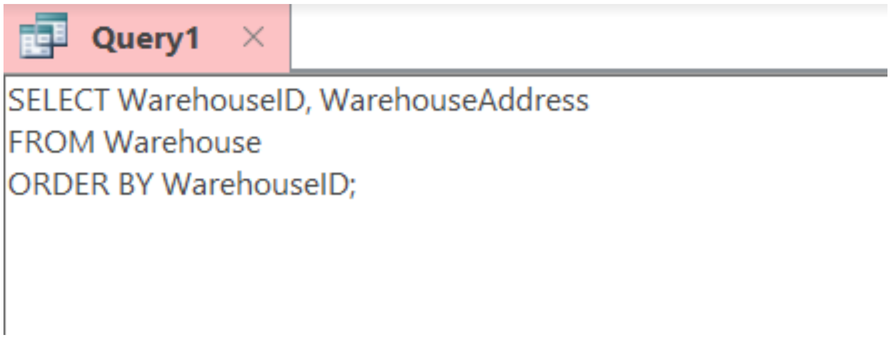


```
SELECT BranchCode, Count(Orders.OrderID) AS ['Orders']
FROM Orders
GROUP BY BranchCode;
```

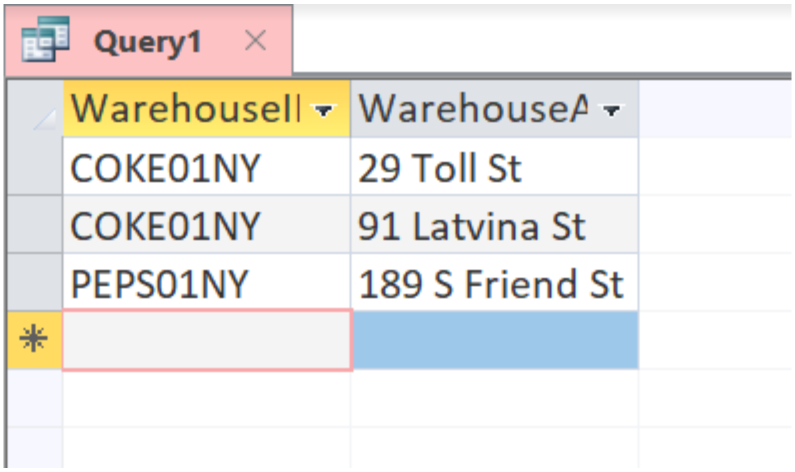


BranchCode	'Orders'
BR189	7
BR190	7
BR191	6

3. Additionally, we can see all the information about our warehouses by completing a simple query like this:

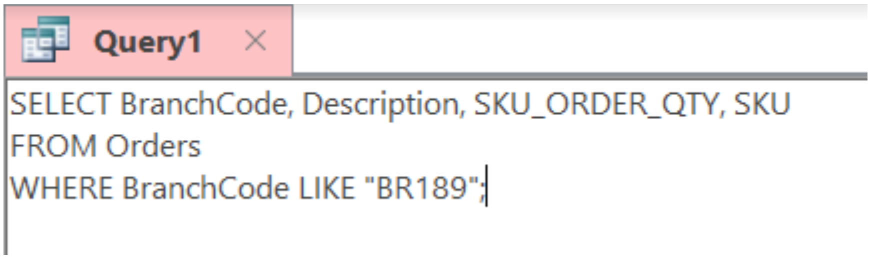


```
SELECT WarehouseID, WarehouseAddress
FROM Warehouse
ORDER BY WarehouseID;
```

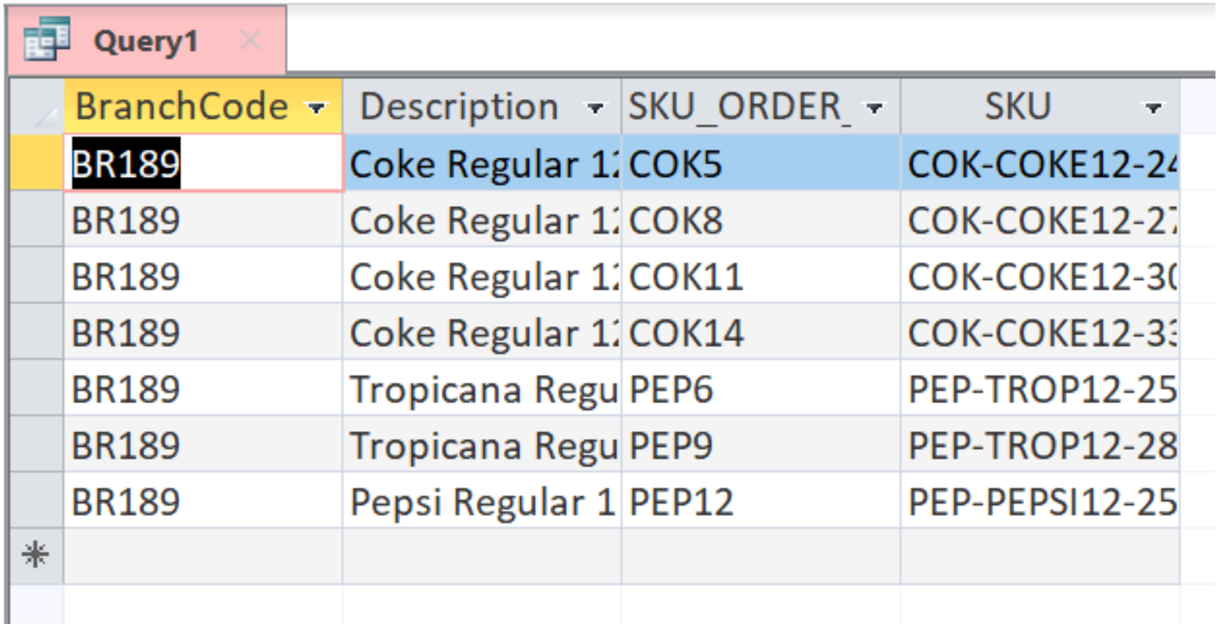


WarehouseI	WarehouseA
COKE01NY	29 Toll St
COKE01NY	91 Latvina St
PEPS01NY	189 S Friend St

4. Finally, another scenario would be if a particular branch wanted to view their order descriptions, order quantity and SKU.

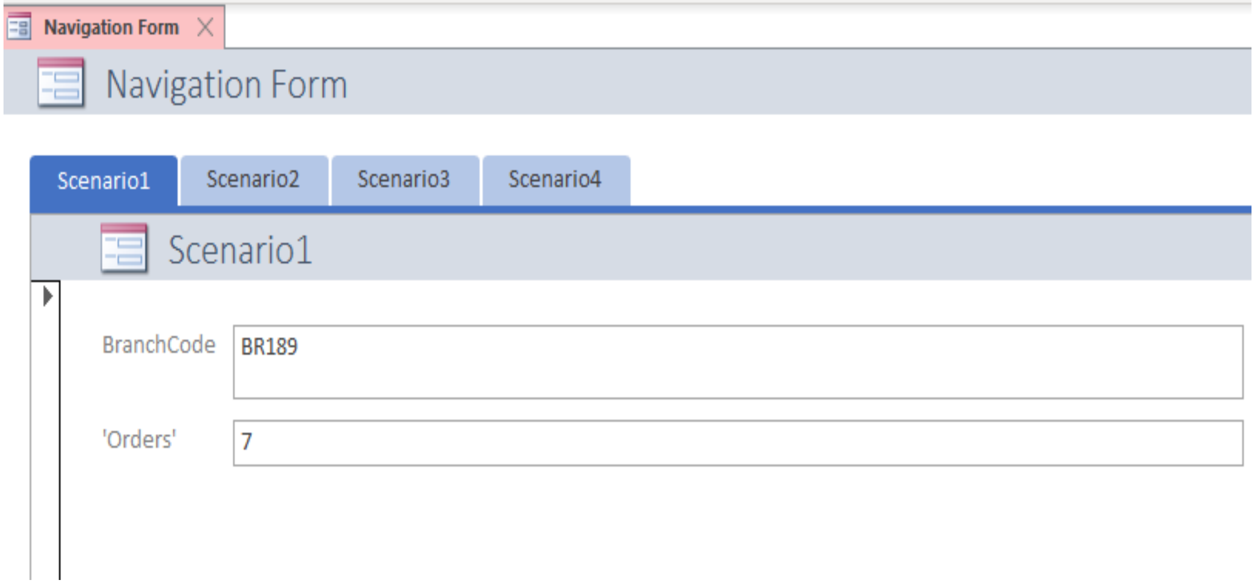


```
SELECT BranchCode, Description, SKU_ORDER_QTY, SKU
FROM Orders
WHERE BranchCode LIKE "BR189";
```

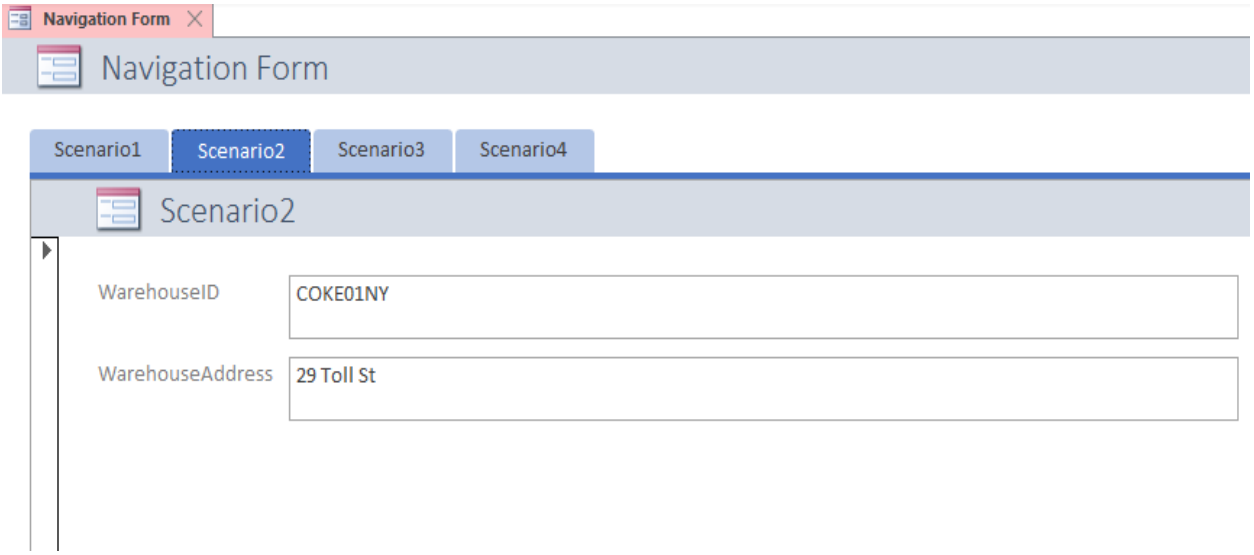


BranchCode	Description	SKU_ORDER_	SKU
BR189	Coke Regular 1:	COK5	COK-COKE12-24
BR189	Coke Regular 1:	COK8	COK-COKE12-27
BR189	Coke Regular 1:	COK11	COK-COKE12-30
BR189	Coke Regular 1:	COK14	COK-COKE12-33
BR189	Tropicana Regu	PEP6	PEP-TROP12-25
BR189	Tropicana Regu	PEP9	PEP-TROP12-28
BR189	Pepsi Regular 1	PEP12	PEP-PEPSI12-25

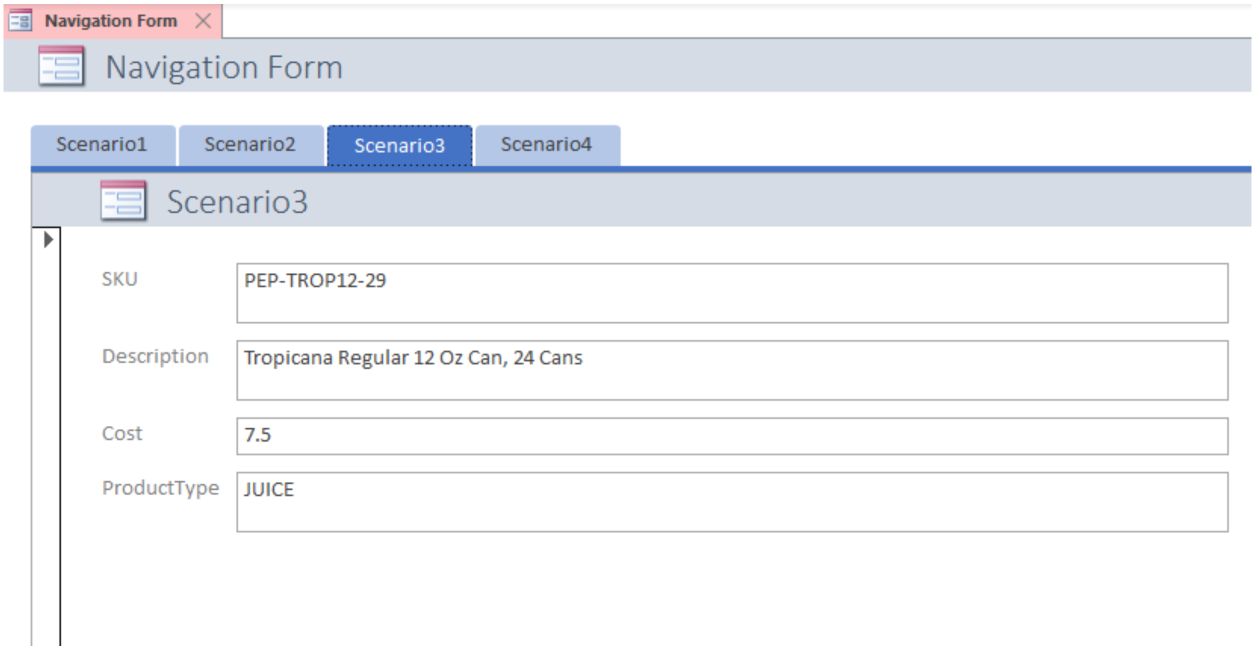
Database Application



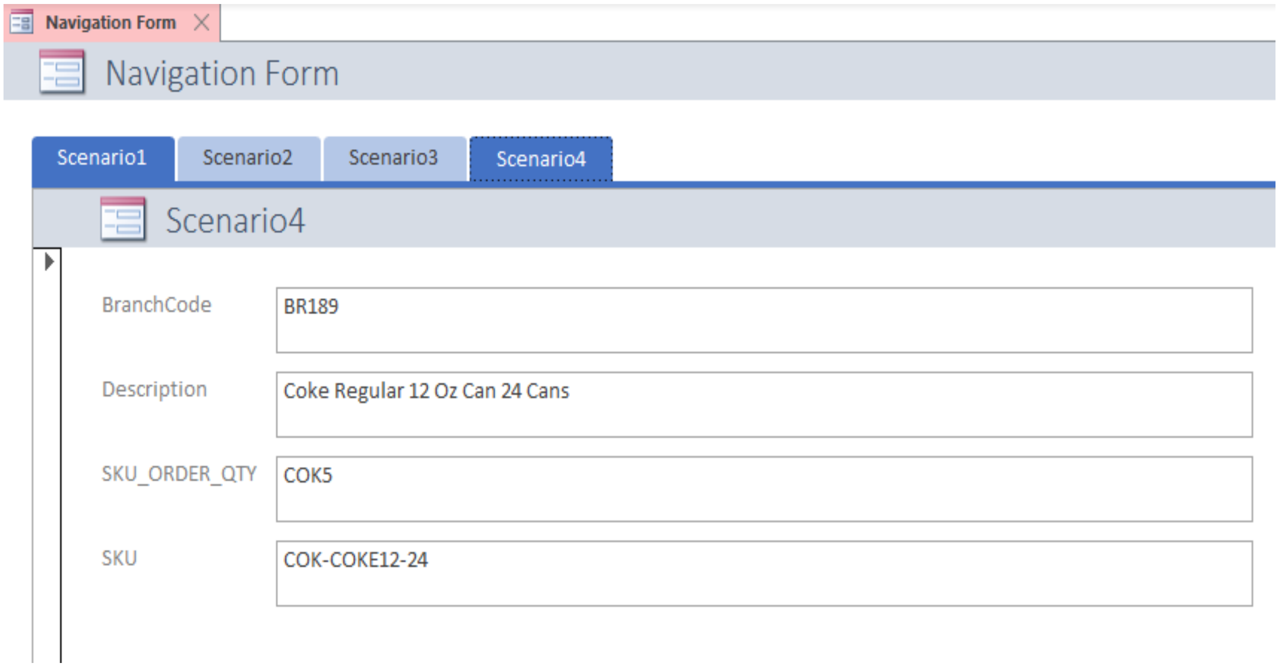
BranchCode	BR189
'Orders'	7



WarehouseID	COKE01NY
WarehouseAddress	29 Toll St



SKU	PEP-TROP12-29
Description	Tropicana Regular 12 Oz Can, 24 Cans
Cost	7.5
ProductType	JUICE



BranchCode	BR189
Description	Coke Regular 12 Oz Can 24 Cans
SKU_ORDER_QTY	COK5
SKU	COK-COKE12-24

Conclusion

Our group found the ERD, RDM and normalization components of the project the easiest to fulfill. To begin with, our project proposal was succinct yet pointed. This enabled us to easily visualize the accompanying entity relationship diagram. Consequently, our RDM was made effortlessly as it was based off of the ERD. We followed the interconnective flow of the ERD to produce relevant relations. Following the completion of the RDM, we implemented normalizations as needed.

The SQL portion of the project was the most challenging. Our process began with referencing the RDM in order to create SQL statements. Firstly, we wrote 'CREATE TABLE' statements of entities and their attributes. Next, we integrated 'ALTER TABLE' statements for the key associated with aforementioned entities. By doing so, we set designated constraints for entities based on primary and foreign keys. Finally, we incorporated 'INSERT INTO' statements. Our insert statements enabled us to add new records/rows of data to our database. Despite having correct and uniform syntax for these statements, this is where parts of our code began to fail and we experienced complications. We had to remedy this by going back to revise our code with correct key associations. The keys and variables did not align correctly in different tables which made inserting data impossible.

If we had to do this step differently, we would ensure to make the code and its syntax more cohesive and consistent. Additionally, we would have used a preset naming system so we could keep track of associated variables. We would have started creating tables in the order of our relational model and moved forward from there. From this project, we learned unexpected ways to troubleshoot the creation of our database. We went back several times to review our code and found that our create table and alter statements were accurate; however, our insert statements were sometimes lacking/faulty and had to be rectified.

Some issues we faced when implementing our database were problems related to Microsoft Access bug issues. We had issues implementing our queries that were fixed with obsolete solutions, and the navigation form had some bugs. Nonetheless, our database proposal fulfilled the proposed benefits of exhibiting demanded information on behalf of an employee or customer. To illustrate, our queries satisfy any potential audits conducted by a worker of "Pete's Snack Corp". This can be demonstrated if a particular salesperson or manager requires detailed information regarding a product branch (i.e. their location and associated product) and employs the database. Furthermore, our database extends the boundaries of internal use. Employees may provide this information to a customer who is seeking their order details or wants to know more about the company's operations. Overall, our product database bridges the gap between internal and external business communication.