

# Database and Data Warehouse Project

---

Michael Verwijst  
12/1/2017

## Table of Contents

|  |    |
|--|----|
| Section 1 – Requirements .....                                       | 2  |
| Business Description .....   | 2  |
| Business Requirements .....  | 2  |
| Conceptual Data Model .....  | 3  |
| Database Requirements .....  | 3  |
| Section 2 – Data Models .....  | 4  |
| Logical Data Model .....   | 4  |
| Entity Description .....   | 4  |
| Data Model Balancing .....   | 5  |
| Section 3 – Physical Database .....                                  | 6  |
| Convert Logical Model to Tables (Already in 2nd Normal Form) .....   | 6  |
| 3rd Normal Form (Showing Changes) .....                              | 8  |
| SQL Code – Create Tables (12 Total with Normalization Changes) ..... | 10 |
| SQL Code – Load Tables .....   | 12 |
| Section 4 – Database Queries .....                                   | 19 |
| Simple Queries .....   | 19 |
| Complex Queries .....  | 21 |
| Section 5 – Data Warehouse Design and Coding .....                   | 23 |
| Data Warehouse Design .....  | 23 |
| Data Mart Design .....   | 25 |
| Data Warehouse Creation – Code .....                                 | 26 |
| Data Warehouse Queries – Code .....                                  | 28 |

## Section 1 – Requirements

### Business Description

This Project will focus on creating a data warehouse for a steel service center's inventory needs. The general structure of the business is to purchase steel products from vendors (manufacturers/mills) in bulk at the lowest price available, store the steel in warehouses as cost effectively as possible, and then sell it to the end customer at top dollar to maximize profit margins. On top of buying and re-selling, the business also offers its own manufacturing services such as cutting and sizing to specific customer requirements to increase sales revenue.

The steel service center has 12 strategically placed facilities across the Midwest to minimize transportation costs, both from vendors and to customers. Each branch has its own inventory to keep track of, with general products as well as products unique to that branch's customers. The company has over 12,000 unique product numbers of varying sizes, shapes, and weight. If a product is cut, shaped, or modified in any way in the manufacturing process, it is given a new product number. The purchasing of products is done at the corporate level as well as at the branch-level, as needed.

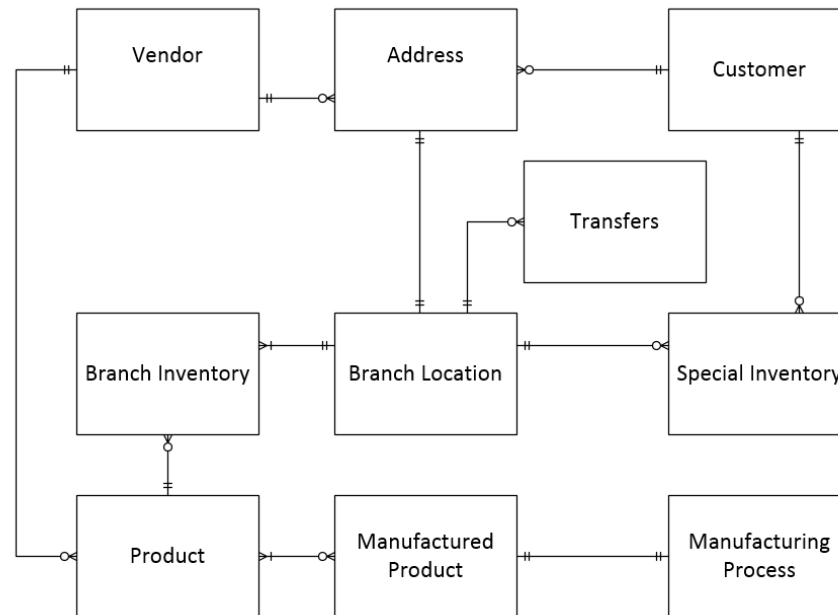
This project will attempt to design a data warehouse that solves the following business problems:

- Provide a central repository of integrated data between distribution and manufacturing systems for Enterprise inventory reporting
- Provide an accessible location to store historical inventory data to allow for trend analysis and pattern recognition
- Provide inventory visibility at the Branch as well as at the corporate level to support all reporting needs company-wide
- Provide business users the ability to analyze at the product level for profitability reporting

### Business Requirements

1. View current Inventory at the Branch Level (For a specific Branch)
2. Identify which products have low inventory and need to be re-ordered
3. Identify which products have been over-allocated in error
4. Identify which products are being transferred from a specific branch
5. Identify which products are unique to a customer (Special Inventory)
6. Identify which products use the same Raw Material
7. Identify the highest cost Raw Material
8. Identify Prospects to focus on converting to Customers
9. Identify how many products are being Cut or Bent
10. Determine Vendors or Customers in a specific State
11. Determine which Vendor the Raw Material for a Product came from
12. Determine the most costly product to Manufacture (Raw Material + Manufacturing Cost)
13. Determine the Inventory level of a Special Product and the accompanied Branch
14. Determine the phone number to call if there is an issue with a specific Raw Material

## Conceptual Data Model

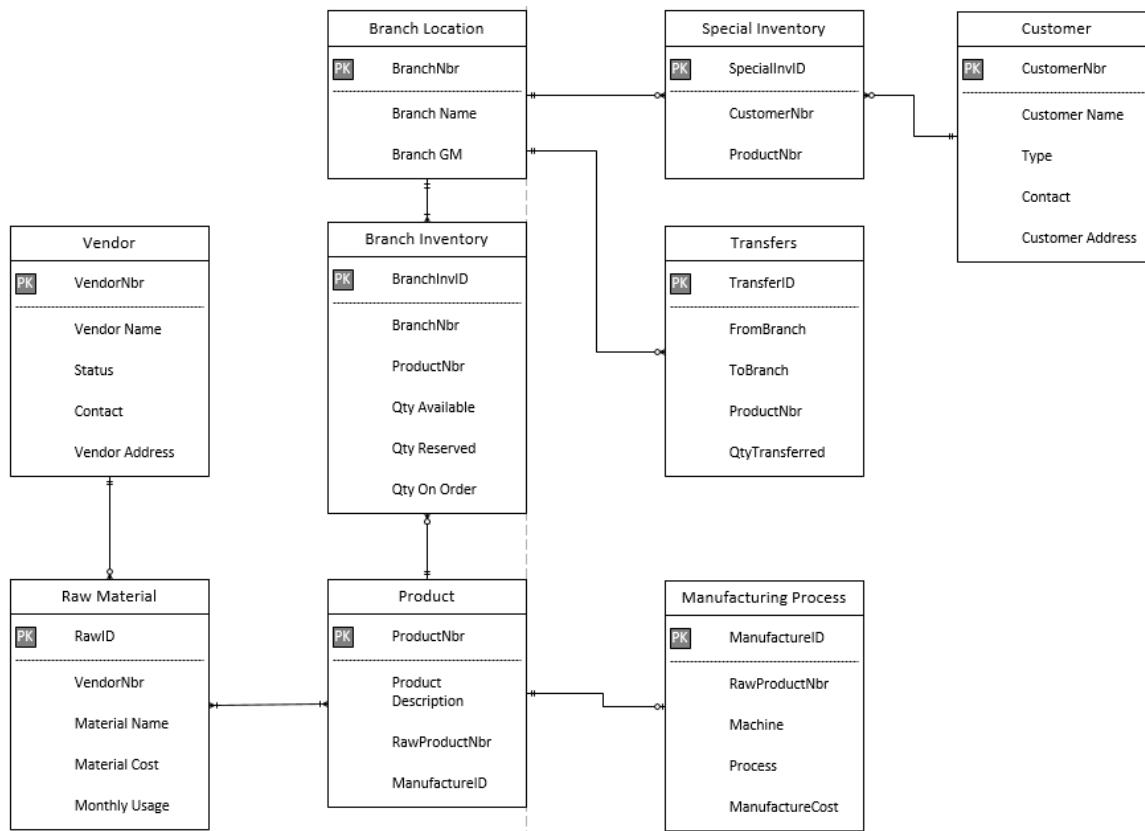


## Database Requirements

1. Each Product is supplied by exactly one Vendor. Conversely, each Vendor supplies any number of Products.
2. Each Supplier can have any number of Addresses. Conversely, each Address is for exactly one Supplier.
3. Each Customer can have any number of Addresses. Conversely, each Address is for exactly one Customer.
4. Each Customer can have any number of Special Inventory Products. Conversely, Each Special Inventory Product is for exactly one customer.
5. Each Special Inventory Product is tied to exactly one Branch Location. Conversely, Each Branch Location can have any number of Special Inventory Products
6. Each Branch Location has exactly one Address. Conversely, Each Address is for exactly one Branch Location
7. Each Branch Location can send any number of Transfers. Conversely, Each Transfer can be sent by exactly one Branch
8. Each Branch Location has any number of Product Inventories. Conversely, Each Branch Inventory is for exactly one Branch Location
9. Each Branch Inventory contains exactly one Product. Conversely, Each Product has Inventory and any number of Branches
10. Each Product can be turned into any number of Manufactured Products. Conversely, Each Manufactured Product is made up of any number of Raw Materials
11. Each Manufactured Product has exactly one Manufacturing Process. Conversely, Each Manufacturing Process is for exactly one Manufactured Product

## Section 2 – Data Models

### Logical Data Model



### Entity Description

- **Vendor** – Contains information on each Vendor that Raw Material has been purchased from
- **Raw Material** – Contains information on each Raw Material that is purchased. Also contains links to the Vendor in which the Raw Material was purchased from
- **Product** – Contains information on each Product that is sold. Also contains links to which Raw Material is used as well as the manufacturing process used to create the product
- **Manufacturing Process** – Contains information regarding the raw material and process required to manufacture and create the final product to be sold.
- **Branch Inventory** - Contains information regarding the inventory levels of each product at each of the company's Branches
- **Branch Location** – Contains information on each Branch. Contains links to inter-branch transfers as well to special products created for only certain customers
- **Transfers** – Contains information regarding origin and destination for inter-branch transfers
- **Special Inventory** – Contains information regarding inventory that is regarded as 'Special', meaning it is created for one particular customer. Links that customer to a specific branch
- **Customer** – Contains information on each Customer that Product has been sold to

### Data Model Balancing

1. Each Vendor supplies any number of Raw Materials. Conversely, each Raw Material is supplied by exactly one Vendor.
2. Each Raw Material is used to create one or more Products. Conversely, each Product uses one or more Raw Materials.
3. If a Product is manufactured, it has exactly one Manufacturing Process (otherwise it does not one). Conversely, each Manufacturing Process is to create exactly one Product.
4. Each Product has Inventory at any number of Branches. Conversely, Each Branch Inventory is for exactly one Product.
5. Each Branch Inventory is for exactly one Branch Location. Conversely, Each Branch Location has any number of Branch Inventories.
6. Each Branch Location sends any number of Transfers. Conversely, Each Transfer is sent by exactly one Branch.
7. Each Branch Location has any number of Special Inventory Products. Conversely, Each Special Inventory Item is tied to exactly one Branch Location.
8. Each Special Inventory Item is for exactly one Customer. Conversely, Each Customer can have any number of Special Inventory Products

## Section 3 – Physical Database

### Convert Logical Model to Tables (Already in 2nd Normal Form)

#### VENDOR

VendorNbr --> VendNbr (PK)

Vendor Name --> VendName

Status --> Status

Contact --> Contact

Vendor Address --> VendAddress

|         |          |        |           |           |
|---------|----------|--------|-----------|-----------|
| VendNbr | VendName | Status | ContactID | AddressID |
|---------|----------|--------|-----------|-----------|

#### RAW\_MATERIAL

RawID --> RawID (PK)

VendorNbr --> VendNbr (FK)

Material Name --> MaterialName

Material Cost --> MaterialCost

|       |         |              |              |
|-------|---------|--------------|--------------|
| RawID | VendNbr | MaterialName | MaterialCost |
|-------|---------|--------------|--------------|

#### PRODUCT

Product Nbr --> ProdNbr (PK)

Product Description --> ProdDesc

RawProductNbr --> RawID (FK)

ManufactureID --> ManufactureID (FK)

|         |          |       |               |
|---------|----------|-------|---------------|
| ProdNbr | ProdDesc | RawID | ManufactureID |
|---------|----------|-------|---------------|

|         |          |       |               |
|---------|----------|-------|---------------|
| ProdNbr | ProdDesc | RawID | ManufactureID |
|---------|----------|-------|---------------|

#### MANUFACTURE\_PROCESS

ManufactureID --> ManufactureID (PK)

RawProductNbr --> RawID (FK)

Machine --> Machine

Process --> Process

ManufactureCost --> ManufactureCost

|               |       |         |         |                 |
|---------------|-------|---------|---------|-----------------|
| ManufactureID | RawID | Machine | Process | ManufactureCost |
|---------------|-------|---------|---------|-----------------|

|               |       |         |         |                 |
|---------------|-------|---------|---------|-----------------|
| ManufactureID | RawID | Machine | Process | ManufactureCost |
|---------------|-------|---------|---------|-----------------|

## BRANCH\_INVENTORY

BranchInvID --&gt; InventoryID (PK)

BranchNbr --&gt; BranchNbr (FK)

ProductNbr --&gt; ProdNbr (FK)

Qty Available --&gt; QtyAvailable

Qty Reserved --&gt; QtyReserved

Qty On Order --&gt; QtyOnOrder

|             |           |         |              |             |            |
|-------------|-----------|---------|--------------|-------------|------------|
| InventoryID | BranchNbr | ProdNbr | QtyAvailable | QtyReserved | QtyOnOrder |
|-------------|-----------|---------|--------------|-------------|------------|

## BRANCH\_LOCATION

BranchNbr --&gt; BranchNbr (PK)

Branch Name --&gt; BranchName

Branch GM --&gt; BranchGM

|           |            |          |
|-----------|------------|----------|
| BranchNbr | BranchName | BranchGM |
|-----------|------------|----------|

|           |            |          |
|-----------|------------|----------|
| BranchNbr | BranchName | BranchGM |
|-----------|------------|----------|

## TRANSFER

TransferID --&gt; TransferID (PK)

FromBranch --&gt; FromBranch

ToBranch --&gt; ToBranch

ProductNbr --&gt; ProdNbr (FK)

Qty Transferred --&gt; QtyTransferred

|            |            |          |         |                |
|------------|------------|----------|---------|----------------|
| TransferID | FromBranch | ToBranch | ProdNbr | QtyTransferred |
|------------|------------|----------|---------|----------------|

## SPECIAL\_INVENTORY

SpecialInvID --&gt; SpecialInvID (PK)

CustomerNbr --&gt; CustNbr (FK)

ProductNbr --&gt; ProdNbr (FK)

|              |         |         |
|--------------|---------|---------|
| SpecialInvID | CustNbr | ProdNbr |
|--------------|---------|---------|

## CUSTOMER

CustomerNbr --&gt; CustNbr (PK)

Customer Name --&gt; CustName

Type --&gt; Type

Contact --&gt; Contact

Customer Address --&gt; CustAddress

|         |          |      |           |           |
|---------|----------|------|-----------|-----------|
| CustNbr | CustName | Type | ContactID | AddressID |
|---------|----------|------|-----------|-----------|



### 3rd Normal Form (Showing Changes)

VENDOR and CUSTOMER --> VENDOR, CUSTOMER and ADDRESSES

VENDOR

VendAddress --> AddressID (FK)

|         |          |        |           |           |
|---------|----------|--------|-----------|-----------|
| VendNbr | VendName | Status | ContactID | AddressID |
|---------|----------|--------|-----------|-----------|

CUSTOMER

CustAddress --> AddressID (FK)

|         |          |      |         |           |
|---------|----------|------|---------|-----------|
| CustNbr | CustName | Type | Contact | AddressID |
|---------|----------|------|---------|-----------|

ADDRESSES Table

|           |               |      |       |     |
|-----------|---------------|------|-------|-----|
| AddressID | StreetAddress | City | State | Zip |
|-----------|---------------|------|-------|-----|

VENDOR and CUSTOMER --> VENDOR, CUSTOMER and CONTACT\_INFO

VENDOR

Contact --> ContactID

|         |          |        |           |           |
|---------|----------|--------|-----------|-----------|
| VendNbr | VendName | Status | ContactID | AddressID |
|---------|----------|--------|-----------|-----------|

CUSTOMER

Contact --> ContactID

|         |          |      |           |           |
|---------|----------|------|-----------|-----------|
| CustNbr | CustName | Type | ContactID | AddressID |
|---------|----------|------|-----------|-----------|

New CONTACT\_INFO Table

|           |           |          |       |       |
|-----------|-----------|----------|-------|-------|
| ContactID | FirstName | LastName | Phone | Email |
|-----------|-----------|----------|-------|-------|

|         |          |        |           |           |
|---------|----------|--------|-----------|-----------|
| VendNbr | VendName | Status | ContactID | AddressID |
|---------|----------|--------|-----------|-----------|

BRANCH\_LOCATION --> BRANCH\_LOCATION and EMPLOYEE

Branch GM --> BranchGM\_EmployeeID

|           |            |                     |
|-----------|------------|---------------------|
| BranchNbr | BranchName | BranchGM_EmployeeID |
|-----------|------------|---------------------|

New EMPLOYEE Table

|            |              |             |           |
|------------|--------------|-------------|-----------|
| EmployeeID | EmpFirstName | EmpLastName | BranchNbr |
|------------|--------------|-------------|-----------|



SQL Code – Create Tables (12 Total with Normalization Changes)

```
CREATE TABLE ADDRESSES (
  AddressID INT NOT NULL PRIMARY KEY,
  StreetAddress VARCHAR (50) NOT NULL,
  City VARCHAR (20) NOT NULL,
  "State" VARCHAR (10) NOT NULL,
  Zip INT NOT NULL );
```

```
CREATE TABLE CONTACT_INFO (
  ContactID INT NOT NULL PRIMARY KEY,
  FirstName VARCHAR (20) NOT NULL,
  LastName VARCHAR (20) NOT NULL,
  Phone VARCHAR (12) NOT NULL,
  Email VARCHAR (20) NOT NULL );
```

```
CREATE TABLE VENDOR (
  VendNbr INT NOT NULL PRIMARY KEY,
  VendName VARCHAR (30) NOT NULL,
  "Status" VARCHAR (10) NOT NULL,
  ContactID INT FOREIGN KEY REFERENCES CONTACT_INFO(ContactID) NOT NULL,
  AddressID INT FOREIGN KEY REFERENCES ADDRESSES(AddressID) NOT NULL );
```

```
CREATE TABLE CUSTOMER (
  CustNbr INT NOT NULL PRIMARY KEY,
  CustName VARCHAR (30) NOT NULL,
  "Type" VARCHAR (10) NOT NULL,
  ContactID INT FOREIGN KEY REFERENCES CONTACT_INFO(ContactID) NOT NULL,
  AddressID INT FOREIGN KEY REFERENCES ADDRESSES(AddressID) NOT NULL );
```

```
CREATE TABLE RAW_MATERIAL (
  RawID INT NOT NULL PRIMARY KEY,
  VendNbr INT FOREIGN KEY REFERENCES VENDOR(VendNbr) NOT NULL,
  MaterialName VARCHAR (20) NOT NULL,
  MaterialCost DECIMAL NOT NULL );
```

```
CREATE TABLE MANUFACTURE_PROCESS (
  ManufactureID INT NOT NULL PRIMARY KEY,
  RawID INT FOREIGN KEY REFERENCES RAW_MATERIAL(RawID) NOT NULL,
  Machine VARCHAR (10) NOT NULL,
  Process VARCHAR (30) NOT NULL,
  ManufactureCost DECIMAL NOT NULL );
```

```
CREATE TABLE PRODUCT (
  ProdNbr INT NOT NULL PRIMARY KEY,
  ProdDesc VARCHAR (50) NOT NULL,
  RawID INT FOREIGN KEY REFERENCES RAW_MATERIAL(RawID) NOT NULL,
  ManufactureID INT FOREIGN KEY REFERENCES MANUFACTURE_PROCESS(ManufactureID) NOT NULL );
```

```
CREATE TABLE EMPLOYEE (  
EmployeeID INT NOT NULL PRIMARY KEY,  
EmpFirstName VARCHAR (20) NOT NULL,  
EmpLastName VARCHAR (20) NOT NULL,  
BranchNbr INT NOT NULL );  
  
CREATE TABLE BRANCH_LOCATION (  
BranchNbr INT NOT NULL PRIMARY KEY,  
BranchName VARCHAR (10) NOT NULL,  
BranchGM_EmployeeID INT FOREIGN KEY REFERENCES EMPLOYEE(EmployeeID) NOT NULL );  
  
CREATE TABLE BRANCH_INVENTORY (  
InventoryID INT NOT NULL PRIMARY KEY,  
BranchNbr INT FOREIGN KEY REFERENCES BRANCH_LOCATION(BranchNbr) NOT NULL,  
ProdNbr INT FOREIGN KEY REFERENCES PRODUCT(ProdNbr) NOT NULL,  
QtyAvailable DECIMAL NOT NULL,  
QtyReserved DECIMAL NOT NULL,  
QtyOnOrder DECIMAL NOT NULL );  
  
CREATE TABLE [TRANSFER] (  
TransferID INT NOT NULL PRIMARY KEY,  
FromBranch INT FOREIGN KEY REFERENCES BRANCH_LOCATION(BranchNbr) NOT NULL,  
ToBranch INT FOREIGN KEY REFERENCES BRANCH_LOCATION(BranchNbr) NOT NULL,  
ProdNbr INT FOREIGN KEY REFERENCES PRODUCT(ProdNbr) NOT NULL,  
QtyTransferred DECIMAL NOT NULL );  
  
CREATE TABLE SPECIAL_INVENTORY (  
SpecialInvID INT NOT NULL PRIMARY KEY,  
CustNbr INT FOREIGN KEY REFERENCES CUSTOMER(CustNbr),  
ProdNbr INT FOREIGN KEY REFERENCES PRODUCT(ProdNbr) );
```



## SQL Code – Load Tables

```

INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('1', '11 Branch', 'Romeoville', 'Illinois', '60446') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('2', '12 Branch', 'Fond du Lac', 'Wisconsin', '54937') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('3', '13 Branch', 'St Paul', 'Minnesota', '55121') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('4', '24 Customer', 'Lake Ville', 'Illinois', '60046') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('5', '25 Customer', 'Grayslake', 'Illinois', '60047') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('6', '26 Customer', 'Burlington', 'Wisconsin', '51805') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('7', '37 Vendor', 'Chicago', 'Illinois', '60606') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('8', '38 Vendor', 'Proctor', 'Minnesota', '55810') ;
INSERT INTO dbo.ADDRESSES (AddressID, StreetAddress, City, [State], Zip)
VALUES ('9', '39 Vendor', 'Milwaukee', 'Illinois', '51810') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | AddressID | StreetAddress | City        | State     | Zip   |
|---|-----------|---------------|-------------|-----------|-------|
| 1 | 1         | 11 Branch     | Romeoville  | Illinois  | 60446 |
| 2 | 2         | 12 Branch     | Fond du Lac | Wisconsin | 54937 |
| 3 | 3         | 13 Branch     | St Paul     | Minnesota | 55121 |
| 4 | 4         | 24 Customer   | Lake Ville  | Illinois  | 60046 |
| 5 | 5         | 25 Customer   | Grayslake   | Illinois  | 60047 |
| 6 | 6         | 26 Customer   | Burlington  | Wisconsin | 51805 |
| 7 | 7         | 37 Vendor     | Chicago     | Illinois  | 60606 |
| 8 | 8         | 38 Vendor     | Proctor     | Minnesota | 55810 |
| 9 | 9         | 39 Vendor     | Milwaukee   | Illinois  | 51810 |

```

INSERT INTO dbo.CONTACT_INFO (ContactID, FirstName, LastName, Phone, Email)
VALUES ('1', 'Betty', 'Crocker', '555-111-1111', 'bcrocker@email.com') ;
INSERT INTO dbo.CONTACT_INFO (ContactID, FirstName, LastName, Phone, Email)
VALUES ('2', 'Abe', 'Lincoln', '555-111-2222', 'alincoln@email.com') ;
INSERT INTO dbo.CONTACT_INFO (ContactID, FirstName, LastName, Phone, Email)
VALUES ('3', 'Mike', 'Jordan', '555-111-3333', 'mjordan@email.com') ;
INSERT INTO dbo.CONTACT_INFO (ContactID, FirstName, LastName, Phone, Email)
VALUES ('4', 'Frank', 'Sinatra', '555-111-4444', 'fsinatra@email.com') ;
INSERT INTO dbo.CONTACT_INFO (ContactID, FirstName, LastName, Phone, Email)
VALUES ('5', 'Aaron', 'Rodgers', '555-111-5555', 'arodgers@email.com') ;
INSERT INTO dbo.CONTACT_INFO (ContactID, FirstName, LastName, Phone, Email)
VALUES ('6', 'Fred', 'Flinstone', '555-111-6666', 'fflinstone@email.com') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | ContactID | FirstName | LastName  | Phone        | Email                |
|---|-----------|-----------|-----------|--------------|----------------------|
| 1 | 1         | Betty     | Crocker   | 555-111-1111 | bcrocker@email.com   |
| 2 | 2         | Abe       | Lincoln   | 555-111-2222 | alincoln@email.com   |
| 3 | 3         | Mike      | Jordan    | 555-111-3333 | mjordan@email.com    |
| 4 | 4         | Frank     | Sinatra   | 555-111-4444 | fsinatra@email.com   |
| 5 | 5         | Aaron     | Rodgers   | 555-111-5555 | arodgers@email.com   |
| 6 | 6         | Fred      | Flinstone | 555-111-6666 | fflinstone@email.com |

```

INSERT INTO dbo.VENDOR (VendNbr, VendName, "Status", ContactID, AddressID)
VALUES ('1', 'Big Steel Inc', 'Active', '4', '7') ;
INSERT INTO dbo.VENDOR (VendNbr, VendName, "Status", ContactID, AddressID)
VALUES ('2', 'New Age Metals', 'Active', '5', '8') ;
INSERT INTO dbo.VENDOR (VendNbr, VendName, "Status", ContactID, AddressID)
VALUES ('3', 'Old World Pipe', 'Not Active', '6', '9') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

|   | VendNbr | VendName       | Status     | ContactID | AddressID |
|---|---------|----------------|------------|-----------|-----------|
| 1 | 1       | Big Steel Inc  | Active     | 4         | 7         |
| 2 | 2       | New Age Metals | Active     | 5         | 8         |
| 3 | 3       | Old World Pipe | Not Active | 6         | 9         |

```

INSERT INTO dbo.CUSTOMER (CustNbr, CustName, "Type", ContactID, AddressID)
VALUES ('1', 'Big Rig Digs', 'Customer', '1', '4') ;
INSERT INTO dbo.CUSTOMER (CustNbr, CustName, "Type", ContactID, AddressID)
VALUES ('2', 'Valves and Fittings', 'Customer', '2', '5') ;
INSERT INTO dbo.CUSTOMER (CustNbr, CustName, "Type", ContactID, AddressID)
VALUES ('3', 'Roller Coasters of America', 'Prospect', '3', '6') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

|   | CustNbr | CustName                   | Type     | ContactID | AddressID |
|---|---------|----------------------------|----------|-----------|-----------|
| 1 | 1       | Big Rig Digs               | Customer | 1         | 4         |
| 2 | 2       | Valves and Fittings        | Customer | 2         | 5         |
| 3 | 3       | Roller Coasters of America | Prospect | 3         | 6         |

```

INSERT INTO dbo.RAW_MATERIAL (RawID, VendNbr, MaterialName, MaterialCost)
VALUES ('1', '1', 'Structural Tubing', '100.00') ;
INSERT INTO dbo.RAW_MATERIAL (RawID, VendNbr, MaterialName, MaterialCost)
VALUES ('2', '1', 'Hot Rolled Coil', '80.50') ;
INSERT INTO dbo.RAW_MATERIAL (RawID, VendNbr, MaterialName, MaterialCost)
VALUES ('3', '2', 'Stainless & Aluminum', '160.25') ;
INSERT INTO dbo.RAW_MATERIAL (RawID, VendNbr, MaterialName, MaterialCost)
VALUES ('4', '2', 'Boiler Tubes', '145.00') ;
INSERT INTO dbo.RAW_MATERIAL (RawID, VendNbr, MaterialName, MaterialCost)
VALUES ('5', '3', 'Big Pipe', '80.00') ;
INSERT INTO dbo.RAW_MATERIAL (RawID, VendNbr, MaterialName, MaterialCost)
VALUES ('6', '3', 'Little Pipe', '40.00') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | RawID | VendNbr | MaterialName         | MaterialCost |
|---|-------|---------|----------------------|--------------|
| 1 | 1     | 1       | Structural Tubing    | 100          |
| 2 | 2     | 1       | Hot Rolled Coil      | 81           |
| 3 | 3     | 2       | Stainless & Aluminum | 160          |
| 4 | 4     | 2       | Boiler Tubes         | 145          |
| 5 | 5     | 3       | Big Pipe             | 80           |
| 6 | 6     | 3       | Little Pipe          | 40           |

```

INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('1', '1', 'Laser', 'Cut into 5 foot pieces', '20') ;
INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('2', '2', 'Laser', 'Cut 5x5 foot squares', '30') ;
INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('3', '3', 'Bender', 'Bend at 90 degree angle', '10') ;
INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('4', '4', 'Laser', 'Cut into 10 foot pieces', '20') ;
INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('5', '5', 'Laser', 'Cut into 8 foot pieces', '15') ;
INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('6', '5', 'Bender', 'Bend at 45 degree angle', '10') ;
INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('7', '6', 'Laser', 'Cut into 2 foot pieces', '10') ;
INSERT INTO dbo.MANUFACTURE_PROCESS(ManufactureID, RawID, Machine, Process,
ManufactureCost)
VALUES ('8', '6', 'Bender', 'Bend at 45 degree angle', '10') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | ManufactureID | RawID | Machine | Process                 | ManufactureCost |
|---|---------------|-------|---------|-------------------------|-----------------|
| 1 | 1             | 1     | Laser   | Cut into 5 foot pieces  | 20              |
| 2 | 2             | 2     | Laser   | Cut 5x5 foot squares    | 30              |
| 3 | 3             | 3     | Bender  | Bend at 90 degree angle | 10              |
| 4 | 4             | 4     | Laser   | Cut into 10 foot pieces | 20              |
| 5 | 5             | 5     | Laser   | Cut into 8 foot pieces  | 15              |
| 6 | 6             | 5     | Bender  | Bend at 45 degree angle | 10              |
| 7 | 7             | 6     | Laser   | Cut into 2 foot pieces  | 10              |
| 8 | 8             | 6     | Bender  | Bend at 45 degree angle | 10              |

```

INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('1', '5 ft Structural Tubing', '1', '1') ;
INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('2', '5x5 ft Steel Sheet', '2', '2') ;
INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('3', 'Stainless Corner Bend', '3', '3') ;
INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('4', '10 ft Boiler Tube', '4', '4') ;
INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('5', '8 ft Big Pipe', '5', '5') ;
INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('6', '8 ft Big Pipe Angled', '5', '6') ;
INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('7', '2 ft Small Pipe', '6', '7') ;
INSERT INTO dbo.PRODUCT(ProdNbr, ProdDesc, RawID, ManufactureID)
VALUES ('8', '2 ft Small Pipe Angled', '6', '8') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | ProdNbr | ProdDesc               | RawID | ManufactureID |
|---|---------|------------------------|-------|---------------|
| 1 | 1       | 5 ft Structural Tubing | 1     | 1             |
| 2 | 2       | 5x5 ft Steel Sheet     | 2     | 2             |
| 3 | 3       | Stainless Corner Bend  | 3     | 3             |
| 4 | 4       | 10 ft Boiler Tube      | 4     | 4             |
| 5 | 5       | 8 ft Big Pipe          | 5     | 5             |
| 6 | 6       | 8 ft Big Pipe Angled   | 5     | 6             |
| 7 | 7       | 2 ft Small Pipe        | 6     | 7             |
| 8 | 8       | 2 ft Small Pipe Angled | 6     | 8             |

```

INSERT INTO dbo.EMPLOYEE(EmployeeID, EmpFirstName, EmpLastName, BranchNbr)
VALUES ('1', 'Vaughn', 'Vietti', '1') ;
INSERT INTO dbo.EMPLOYEE(EmployeeID, EmpFirstName, EmpLastName, BranchNbr)
VALUES ('2', 'Rod', 'RanRite', '2') ;
INSERT INTO dbo.EMPLOYEE(EmployeeID, EmpFirstName, EmpLastName, BranchNbr)
VALUES ('3', 'Bob', 'Barville', '3') ;
INSERT INTO dbo.EMPLOYEE(EmployeeID, EmpFirstName, EmpLastName, BranchNbr)
VALUES ('4', 'Mike', 'Verwijst', '1') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | EmployeeID | EmpFirstName | EmpLastName | BranchNbr |
|---|------------|--------------|-------------|-----------|
| 1 | 1          | Vaughn       | Vietti      | 1         |
| 2 | 2          | Rod          | RanRite     | 2         |
| 3 | 3          | Bob          | Barville    | 3         |
| 4 | 4          | Mike         | Verwijst    | 1         |



```

INSERT INTO dbo.BRANCH_LOCATION(BranchNbr, BranchName, BranchGM_EmployeeID)
VALUES ('1', 'Chicago', '1') ;
INSERT INTO dbo.BRANCH_LOCATION(BranchNbr, BranchName, BranchGM_EmployeeID)
VALUES ('2', 'Wisconsin', '2') ;
INSERT INTO dbo.BRANCH_LOCATION(BranchNbr, BranchName, BranchGM_EmployeeID)
VALUES ('3', 'Minnesota', '3') ;

```

(1 row affected)

|                  | BranchNbr | BranchName | BranchGM_EmployeeID |
|------------------|-----------|------------|---------------------|
| (1 row affected) | 1         | Chicago    | 1                   |
|                  | 2         | Wisconsin  | 2                   |
| (1 row affected) | 3         | Minnesota  | 3                   |

```

INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('11', '1', '1', '20', '50', '10') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('12', '2', '1', '10', '20', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('13', '3', '1', '10', '20', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('21', '1', '2', '50', '100', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('22', '2', '2', '25', '40', '5') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('23', '3', '2', '25', '40', '5') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('31', '1', '3', '10', '10', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('32', '2', '3', '10', '5', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('33', '3', '3', '10', '5', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('41', '1', '4', '10', '10', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('42', '2', '4', '10', '5', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('43', '3', '4', '10', '5', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('51', '1', '5', '50', '20', '20') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('52', '2', '5', '0', '0', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('53', '3', '5', '0', '0', '0') ;

```

```

INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('61', '1', '6', '20', '10', '10') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('62', '2', '6', '0', '0', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('63', '3', '6', '0', '0', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('71', '1', '7', '0', '0', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('72', '2', '7', '100', '200', '50') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('73', '3', '7', '0', '0', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('81', '1', '8', '0', '0', '0') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('82', '2', '8', '60', '40', '10') ;
INSERT INTO dbo.BRANCH_INVENTORY(InventoryID, BranchNbr, ProdNbr, QtyAvailable,
QtyOnOrder, QtyReserved)
VALUES ('83', '3', '8', '0', '0', '0') ;

```

|    | InventoryID | BranchNbr | ProdNbr | QtyAvailable | QtyReserved | QtyOnOrder |
|----|-------------|-----------|---------|--------------|-------------|------------|
| 1  | 11          | 1         | 1       | 20           | 10          | 50         |
| 2  | 12          | 2         | 1       | 10           | 0           | 20         |
| 3  | 13          | 3         | 1       | 10           | 0           | 20         |
| 4  | 21          | 1         | 2       | 50           | 0           | 100        |
| 5  | 22          | 2         | 2       | 25           | 5           | 40         |
| 6  | 23          | 3         | 2       | 25           | 5           | 40         |
| 7  | 31          | 1         | 3       | 10           | 0           | 10         |
| 8  | 32          | 2         | 3       | 10           | 0           | 5          |
| 9  | 33          | 3         | 3       | 10           | 0           | 5          |
| 10 | 41          | 1         | 4       | 10           | 0           | 10         |
| 11 | 42          | 2         | 4       | 10           | 0           | 5          |

```

INSERT INTO dbo.TRANSFER(TransferID, FromBranch, ToBranch, ProdNbr,
QtyTransferred)
VALUES ('1', '1', '2', '1', '15') ;
INSERT INTO dbo.TRANSFER(TransferID, FromBranch, ToBranch, ProdNbr,
QtyTransferred)
VALUES ('2', '1', '3', '4', '5') ;
INSERT INTO dbo.TRANSFER(TransferID, FromBranch, ToBranch, ProdNbr,
QtyTransferred)
VALUES ('3', '2', '1', '8', '25') ;
INSERT INTO dbo.TRANSFER(TransferID, FromBranch, ToBranch, ProdNbr,
QtyTransferred)
VALUES ('4', '2', '3', '8', '15') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | TransferID | FromBranch | ToBranch | ProdNbr | QtyTransferred |
|---|------------|------------|----------|---------|----------------|
| 1 | 1          | 1          | 2        | 1       | 15             |
| 2 | 2          | 1          | 3        | 4       | 5              |
| 3 | 3          | 2          | 1        | 8       | 25             |
| 4 | 4          | 2          | 3        | 8       | 15             |

```

INSERT INTO dbo.SPECIAL_INVENTORY(SpecialInvID, CustNbr, ProdNbr)
VALUES ('1', '1', '5') ;
INSERT INTO dbo.SPECIAL_INVENTORY(SpecialInvID, CustNbr, ProdNbr)
VALUES ('2', '2', '6') ;
INSERT INTO dbo.SPECIAL_INVENTORY(SpecialInvID, CustNbr, ProdNbr)
VALUES ('3', '3', '7') ;
INSERT INTO dbo.SPECIAL_INVENTORY(SpecialInvID, CustNbr, ProdNbr)
VALUES ('4', '3', '8') ;

```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

|   | SpecialInvID | CustNbr | ProdNbr |
|---|--------------|---------|---------|
| 1 | 1            | 1       | 5       |
| 2 | 2            | 2       | 6       |
| 3 | 3            | 3       | 7       |
| 4 | 4            | 3       | 8       |

## Section 4 – Database Queries

### Simple Queries

1. View current Inventory at the Branch Level (For a specific Branch)

```
SELECT *
FROM BRANCH_INVENTORY
WHERE BranchNbr = 1
```

|   | InventoryID | BranchNbr | ProdNbr | QtyAvailable | QtyReserved | QtyOnOrder |
|---|-------------|-----------|---------|--------------|-------------|------------|
| 1 | 11          | 1         | 1       | 20           | 10          | 50         |
| 2 | 21          | 1         | 2       | 50           | 0           | 100        |
| 3 | 31          | 1         | 3       | 10           | 0           | 10         |
| 4 | 41          | 1         | 4       | 10           | 0           | 10         |
| 5 | 51          | 1         | 5       | 50           | 20          | 20         |
| 6 | 61          | 1         | 6       | 20           | 10          | 10         |
| 7 | 71          | 1         | 7       | 0            | 0           | 0          |
| 8 | 81          | 1         | 8       | 0            | 0           | 0          |

2. Identify which products have low inventory and need to be re-ordered

```
SELECT *
FROM BRANCH_INVENTORY
WHERE QtyAvailable < 10
AND BranchNbr = 1
```

|   | InventoryID | BranchNbr | ProdNbr | QtyAvailable | QtyReserved | QtyOnOrder |
|---|-------------|-----------|---------|--------------|-------------|------------|
| 1 | 71          | 1         | 7       | 0            | 0           | 0          |
| 2 | 81          | 1         | 8       | 0            | 0           | 0          |

3. Identify which products have been over-allocated in error

```
SELECT *
FROM BRANCH_INVENTORY
WHERE QtyReserved > QtyAvailable
```

|   | InventoryID | BranchNbr | ProdNbr | QtyAvailable | QtyReserved | QtyOnOrder |
|---|-------------|-----------|---------|--------------|-------------|------------|
| 1 | 41          | 1         | 4       | 10           | 20          | 10         |

4. Identify which products are being transferred from a specific branch

```
SELECT *
FROM [TRANSFER]
WHERE FromBranch = 1
```

|   | TransferID | FromBranch | ToBranch | ProdNbr | QtyTransferred |
|---|------------|------------|----------|---------|----------------|
| 1 | 1          | 1          | 2        | 1       | 15             |
| 2 | 2          | 1          | 3        | 4       | 5              |

5. Identify which products are unique to a customer (Special Inventory)

```
SELECT *
FROM SPECIAL_INVENTORY
```

|   | SpecialInvID | CustNbr | ProdNbr |
|---|--------------|---------|---------|
| 1 | 1            | 1       | 5       |
| 2 | 2            | 2       | 6       |
| 3 | 3            | 3       | 7       |
| 4 | 4            | 3       | 8       |

## 6. Identify which products use the same Raw Material

```

SELECT ProdNbr, ProdDesc, RawID
FROM PRODUCT
WHERE RawID = 5
      OR RawID = 6

```

|   | ProdNbr | ProdDesc               | RawID |
|---|---------|------------------------|-------|
| 1 | 5       | 8 ft Big Pipe          | 5     |
| 2 | 6       | 8 ft Big Pipe Angled   | 5     |
| 3 | 7       | 2 ft Small Pipe        | 6     |
| 4 | 8       | 2 ft Small Pipe Angled | 6     |

## 7. Identify the highest cost Raw Material to focus on lowering

```

SELECT RawID, MaterialName, MaterialCost
FROM RAW_MATERIAL
WHERE MaterialCost = (SELECT MAX(MaterialCost) FROM RAW_MATERIAL)

```

|   | RawID | MaterialName         | MaterialCost |
|---|-------|----------------------|--------------|
| 1 | 3     | Stainless & Aluminum | 160          |

## 8. Identify Prospects to focus on converting to Customers

```

SELECT CustNbr, CustName, Type
FROM CUSTOMER
WHERE Type = 'Prospect'

```

|   | CustNbr | CustName                   | Type     |
|---|---------|----------------------------|----------|
| 1 | 3       | Roller Coasters of America | Prospect |

## 9. Identify how many products are being Cut or Bent

```

SELECT COUNT(RawID) as NbrCut
FROM MANUFACTURE_PROCESS
WHERE Machine = 'Laser'

```

|   | NbrCut |
|---|--------|
| 1 | 5      |

```

SELECT COUNT(RawID) as NbrBent
FROM MANUFACTURE_PROCESS
WHERE Machine = 'Bender'

```

|   | NbrBent |
|---|---------|
| 1 | 3       |

## Complex Queries

### 10. Determine Vendors or Customers in a specific State

```
SELECT a.State, b.VendNbr, b.VendName
FROM ADDRESSES a
      JOIN VENDOR b on a.AddressID = b.AddressID
WHERE a.State = 'Illinois'
```

|   | State    | VendNbr | VendName       |
|---|----------|---------|----------------|
| 1 | Illinois | 1       | Big Steel Inc  |
| 2 | Illinois | 3       | Old World Pipe |

```
SELECT a.State, b.CustNbr, b.CustName
FROM ADDRESSES a
      JOIN CUSTOMER b on a.AddressID = b.AddressID
WHERE a.State = 'Illinois'
```

|   | State    | CustNbr | CustName            |
|---|----------|---------|---------------------|
| 1 | Illinois | 1       | Big Rig Digs        |
| 2 | Illinois | 2       | Valves and Fittings |

### 11. Determine which Vendor the Raw Material for a Product came from

```
SELECT a.ProdNbr, a.ProdDesc, a.RawID, b.MaterialName, b.VendNbr, c.VendName
FROM PRODUCT a
      JOIN RAW_MATERIAL b on a.RawID=b.RawID
      JOIN VENDOR c on b.VendNbr = c.VendNbr
```

|   | ProdNbr | ProdDesc               | RawID | MaterialName         | VendNbr | VendName       |
|---|---------|------------------------|-------|----------------------|---------|----------------|
| 1 | 1       | 5 ft Structural Tubing | 1     | Structural Tubing    | 1       | Big Steel Inc  |
| 2 | 2       | 5x5 ft Steel Sheet     | 2     | Hot Rolled Coil      | 1       | Big Steel Inc  |
| 3 | 3       | Stainless Corner Bend  | 3     | Stainless & Aluminum | 2       | New Age Metals |
| 4 | 4       | 10 ft Boiler Tube      | 4     | Boiler Tubes         | 2       | New Age Metals |
| 5 | 5       | 8 ft Big Pipe          | 5     | Big Pipe             | 3       | Old World Pipe |
| 6 | 6       | 8 ft Big Pipe Angled   | 5     | Big Pipe             | 3       | Old World Pipe |
| 7 | 7       | 2 ft Small Pipe        | 6     | Little Pipe          | 3       | Old World Pipe |
| 8 | 8       | 2 ft Small Pipe Angled | 6     | Little Pipe          | 3       | Old World Pipe |

### 12. Determine the Total Cost to Manufacture a Product (Raw Material + Manufacturing Cost)

```
SELECT a.ProdNbr, a.ProdDesc, b.ManufactureCost, c.MaterialCost,
(b.ManufactureCost + c.MaterialCost) as TotalCost
FROM PRODUCT a
      JOIN MANUFACTURE_PROCESS b on a.ManufactureID = b.ManufactureID
      JOIN RAW_MATERIAL c on a.RawID = c.RawID
WHERE ProdNbr = 4
```

|   | ProdNbr | ProdDesc          | ManufactureID | ManufactureCost | RawID | MaterialCost | TotalCost |
|---|---------|-------------------|---------------|-----------------|-------|--------------|-----------|
| 1 | 4       | 10 ft Boiler Tube | 4             | 20              | 4     | 145          | 165       |

## 13. Determine the Inventory level of a Special Product at each Branch

```

SELECT a.SpecialInvID, a.ProdNbr, b.BranchNbr, b.QtyAvailable, b.QtyOnOrder,
b.QtyReserved
FROM SPECIAL_INVENTORY a
      JOIN BRANCH_INVENTORY b on a.ProdNbr = b.ProdNbr
WHERE a.SpecialInvID = 1

```

|   | SpecialInvID | ProdNbr | BranchNbr | QtyAvailable | QtyOnOrder | QtyReserved |
|---|--------------|---------|-----------|--------------|------------|-------------|
| 1 | 1            | 5       | 1         | 50           | 20         | 20          |
| 2 | 1            | 5       | 2         | 0            | 0          | 0           |
| 3 | 1            | 5       | 3         | 0            | 0          | 0           |

## 14. Determine who to contact if there is an issue with a specific Raw Material

```

SELECT a.RawID, a.MaterialName, b.VendNbr, c.FirstName, c.LastName, c.Phone,
c.Email
FROM RAW_MATERIAL a
      JOIN VENDOR b on a.VendNbr = b.VendNbr
      JOIN CONTACT_INFO c on b.ContactID = c.ContactID
WHERE a.RawID = 6

```

|   | RawID | MaterialName | VendNbr | FirstName | LastName  | Phone        | Email                |
|---|-------|--------------|---------|-----------|-----------|--------------|----------------------|
| 1 | 6     | Little Pipe  | 3       | Fred      | Flinstone | 555-111-6666 | fflinstone@email.com |

## Section 5 – Data Warehouse Design and Coding

### Data Warehouse Design

As defined in the Business Description section, the goals of this project are as follows:

- Provide a central repository of integrated data between distribution and manufacturing systems for Enterprise inventory reporting
- Provide an accessible location to store historical inventory data to allow for trend analysis and pattern recognition
- Provide inventory visibility at the Branch as well as at the corporate level to support all reporting needs company-wide
- Provide business users the ability to analyze at the product level for profitability reporting

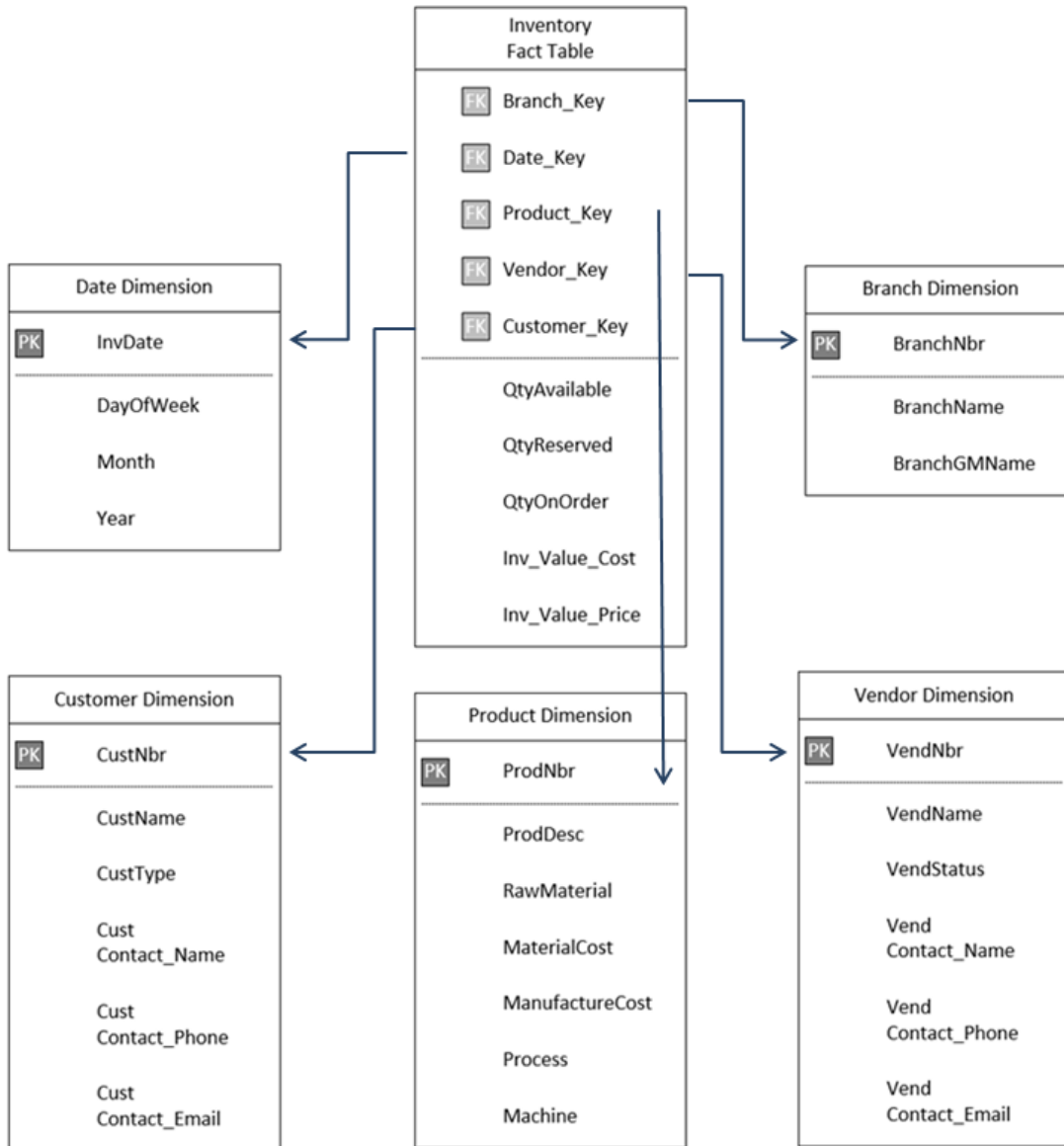
In order to achieve these goals, a Data Warehouse was designed using the Inventory Snapshot Fact Table Model with some added dimensions for Vendor and Customer information. The Inventory Fact Table and all four of the supporting Dimension Tables (Branch, Product, Customer, and Vendor) were designed and created based off the attributes in the database tables listed in the above sections. The Date dimension, a crucial aspect of the Data Warehouse, was added afterwards in order to get a daily snapshot of Inventory.

Although the operational database tables will only store and show current inventory values as of today, this Data Warehouse will track the history. Each night, the inventory levels will be captured at their atomic level and be pushed here, which will allow for the analysis needed to reach the objectives stated above. The branch dimension will allow inventory to be analyzed at the Branch level, and summaries can be calculated to achieve corporate requirements. The product dimension will provide product level granularity and allow for analysis of each individual item. Although not directly required, the Customer and Vendor dimensions will allow for some additional analysis of product origin and special inventory-customer analyses.

In future renditions of this project, it would be very beneficial to add in more detail around the Vendor and Customer metrics. Purchase Order and Sales Order information would have extended the scope of this project significantly, and was thus ignored. However, these additional important aspects of the business would be very beneficial to add at a later date.

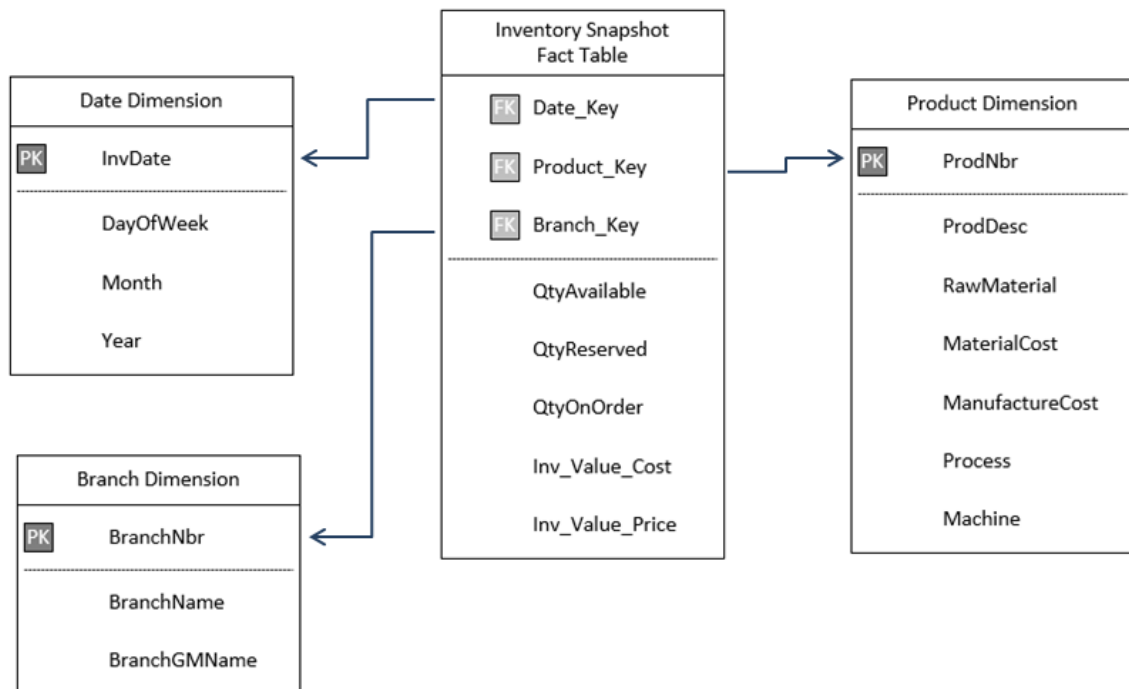


The diagram below shows the relationships between the Fact Table and supporting Dimension Tables:



## Data Mart Design

The Data Warehouse above contains supplementary information regarding Vendor and Customer contact and addresses, but that information may not be relevant to all business users. In the case where a user is only interested in the more traditional inventory attributes, the Inventory Snapshot Fact Table and supporting Dimension table can be used as a Data Mart. This will allow the Inventory data to be queried more quickly and speed up reporting and analytics. Of course, the Vendor and Customer information would still be available for access in the Data Warehouse above.



## Data Warehouse Creation – Code

### BranchDimension

```
SELECT a.BranchNbr, a.BranchName, (b.EmpFirstName + ' ' + b.EmpLastName) as
BranchGMName
INTO BranchDimension
FROM BRANCH_LOCATION a
      Left Join EMPLOYEE b on a.BranchGM_EmployeeID = b.EmployeeID
```

|   | BranchNbr | BranchName | BranchGMName  |
|---|-----------|------------|---------------|
| 1 | 1         | Chicago    | Vaughn Vietti |
| 2 | 2         | Wisconsin  | Rod RanRite   |
| 3 | 3         | Minnesota  | Bob Barville  |

### VendorDimension

```
SELECT a.VendNbr, a.VendName, a.Status as VendStatus, (b.FirstName + b.LastName)
as VendContactName, b.Phone as VendContactPhone, b.Email as VendContactEmail,
(c.StreetAddress + ', ' + c.City + ', ' + c.State) as VendAddress
INTO VendorDimension
FROM VENDOR a
      Left Join CONTACT_INFO b on a.ContactID = b.ContactID
      Left Join ADDRESSES c on a.AddressID = c.AddressID
```

|   | VendNbr | VendName       | VendStatus | VendContactName | VendContactPhone | VendContactEmail     |
|---|---------|----------------|------------|-----------------|------------------|----------------------|
| 1 | 1       | Big Steel Inc  | Active     | FrankSinatra    | 555-111-4444     | fsinatra@email.com   |
| 2 | 2       | New Age Metals | Active     | AaronRodgers    | 555-111-5555     | arodgers@email.com   |
| 3 | 3       | Old World Pipe | Not Active | FredFlinstone   | 555-111-6666     | fflinstone@email.com |

### CustomerDimension

```
SELECT a.CustNbr, a.CustName, a.Type as CustType,
(b.FirstName + ' ' + b.LastName) as CustContactName, b.Phone as CustContactPhone,
b.Email as CustContactEmail, (c.StreetAddress + ', ' + c.City + ', ' + c.State) as
CustAddress, d.ProdNbr as SpecialProdNbr
INTO CustomerDimension
FROM CUSTOMER a
      Left Join CONTACT_INFO b on a.ContactID = b.ContactID
      Left Join ADDRESSES c on a.AddressID = c.AddressID
      Left Join SPECIAL_INVENTORY d on a.CustNbr = d.CustNbr
```

|   | CustNbr | CustName                | Cust Type | CustContactName | CustContactPhone | CustContactEmail    |
|---|---------|-------------------------|-----------|-----------------|------------------|---------------------|
| 1 | 1       | Big Rig Digs            | Customer  | BettyCrocker    | 555-111-1111     | bcrocker@email.com  |
| 2 | 2       | Valves and Fittings     | Customer  | AbeLincoln      | 555-111-2222     | alincolin@email.com |
| 3 | 3       | Roller Coasters of A... | Prospect  | MikeJordan      | 555-111-3333     | mjordan@email.com   |

### ProductDimension

```
SELECT a.ProdNbr, a.ProdDesc, b.MaterialName as RawMaterial, b.MaterialCost,
c.ManufactureCost, c.Process, c.Machine
INTO ProductDimension
FROM PRODUCT a
      Left Join RAW_MATERIAL b on a.RawID = b.RawID
      Left Join MANUFACTURE_PROCESS c on a.ManufactureID = c.ManufactureID
```

|   | ProdNbr | ProdDesc               | RawMaterial          | MaterialCost | ManufactureCost | Process                 | Machine |
|---|---------|------------------------|----------------------|--------------|-----------------|-------------------------|---------|
| 1 | 1       | 5 ft Structural Tubing | Structural Tubing    | 100          | 20              | Cut into 5 foot pieces  | Laser   |
| 2 | 2       | 5x5 ft Steel Sheet     | Hot Rolled Coil      | 81           | 30              | Cut 5x5 foot squares    | Laser   |
| 3 | 3       | Stainless Corner Bend  | Stainless & Aluminum | 160          | 10              | Bend at 90 degree angle | Bender  |
| 4 | 4       | 10 ft Boiler Tube      | Boiler Tubes         | 145          | 20              | Cut into 10 foot pieces | Laser   |
| 5 | 5       | 8 ft Big Pipe          | Big Pipe             | 80           | 15              | Cut into 8 foot pieces  | Laser   |
| 6 | 6       | 8 ft Big Pipe Angled   | Big Pipe             | 80           | 10              | Bend at 45 degree angle | Bender  |
| 7 | 7       | 2 ft Small Pipe        | Little Pipe          | 40           | 10              | Cut into 2 foot pieces  | Laser   |
| 8 | 8       | 2 ft Small Pipe Angled | Little Pipe          | 40           | 10              | Bend at 45 degree angle | Bender  |

## DateDimension

```
CREATE TABLE DateDimension (
  InvDate DATE NOT NULL PRIMARY KEY,
  "DayOfWeek" VARCHAR (10) NOT NULL,
  "Month" VARCHAR (20) NOT NULL,
  "Year" Integer NOT NULL);
INSERT INTO dbo.DateDimension (InvDate, DayOfWeek, Month, Year)
VALUES ('2017-11-01', 'Wednesday', 'November', '2017');
INSERT INTO dbo.DateDimension (InvDate, DayOfWeek, Month, Year)
VALUES ('2017-11-02', 'Thursday', 'November', '2017');
INSERT INTO dbo.DateDimension (InvDate, DayOfWeek, Month, Year)
VALUES ('2017-11-03', 'Friday', 'November', '2017');
```

|   | InvDate    | DayOfWeek | Month    | Year |
|---|------------|-----------|----------|------|
| 1 | 2017-11-01 | Wednesday | November | 2017 |
| 2 | 2017-11-02 | Thursday  | November | 2017 |
| 3 | 2017-11-03 | Friday    | November | 2017 |

## Inventory Fact Table

```
SELECT b.BranchNbr, f.InvDate, c.ProdNbr, d.VendNbr, e.CustNbr,
a.QtyAvailable, a.QtyReserved, a.QtyOnOrder,
(a.QtyAvailable * (c.MaterialCost*c.ManufactureCost)) as Inv_Value_Cost
INTO InventoryFactTable
FROM BRANCH_INVENTORY a
  Left Join BranchDimension b on a.BranchNbr = b.BranchNbr
  Left Join ProductDimension c on a.ProdNbr = c.ProdNbr
  Left join VendorDimension d on c.VendNbr = d.VendNbr
  Left Join CustomerDimension e on a.ProdNbr = e.SpecialProdNbr
  Left Join DateDimension f on a.InvDate = f.InvDate
```

|    | BranchNbr | InvDate    | ProdNbr | VendNbr | CustNbr | QtyAvailable | QtyReserved | QtyOnOrder | Inv_Value_Cost |
|----|-----------|------------|---------|---------|---------|--------------|-------------|------------|----------------|
| 1  | 1         | 2017-11-03 | 1       | 1       | NULL    | 20           | 10          | 50         | 40000          |
| 2  | 2         | 2017-11-03 | 1       | 1       | NULL    | 10           | 0           | 20         | 20000          |
| 3  | 3         | 2017-11-03 | 1       | 1       | NULL    | 10           | 0           | 20         | 20000          |
| 4  | 1         | 2017-11-03 | 2       | 1       | NULL    | 50           | 0           | 100        | 121500         |
| 5  | 2         | 2017-11-03 | 2       | 1       | NULL    | 25           | 5           | 40         | 60750          |
| 6  | 3         | 2017-11-03 | 2       | 1       | NULL    | 25           | 5           | 40         | 60750          |
| 7  | 1         | 2017-11-03 | 3       | 2       | NULL    | 10           | 0           | 10         | 16000          |
| 8  | 2         | 2017-11-03 | 3       | 2       | NULL    | 10           | 0           | 5          | 16000          |
| 9  | 3         | 2017-11-03 | 3       | 2       | NULL    | 10           | 0           | 5          | 16000          |
| 10 | 1         | 2017-11-03 | 4       | 2       | NULL    | 10           | 20          | 10         | 29000          |
| 11 | 2         | 2017-11-03 | 4       | 2       | NULL    | 10           | 0           | 5          | 29000          |
| 12 | 3         | 2017-11-03 | 4       | 2       | NULL    | 10           | 0           | 5          | 29000          |
| 13 | 1         | 2017-11-03 | 5       | 3       | 1       | 50           | 20          | 20         | 60000          |
| 14 | 2         | 2017-11-03 | 5       | 3       | 1       | 0            | 0           | 0          | 0              |
| 15 | 3         | 2017-11-03 | 5       | 3       | 1       | 0            | 0           | 0          | 0              |
| 16 | 1         | 2017-11-03 | 6       | 3       | 2       | 20           | 10          | 10         | 16000          |
| 17 | 2         | 2017-11-03 | 6       | 3       | 2       | 0            | 0           | 0          | 0              |
| 18 | 3         | 2017-11-03 | 6       | 3       | 2       | 0            | 0           | 0          | 0              |
| 19 | 1         | 2017-11-03 | 7       | 3       | 3       | 0            | 0           | 0          | 0              |
| 20 | 2         | 2017-11-03 | 7       | 3       | 3       | 100          | 50          | 200        | 40000          |
| 21 | 3         | 2017-11-03 | 7       | 3       | 3       | 0            | 0           | 0          | 0              |
| 22 | 1         | 2017-11-03 | 8       | 3       | 3       | 0            | 0           | 0          | 0              |
| 23 | 2         | 2017-11-03 | 8       | 3       | 3       | 60           | 10          | 40         | 24000          |
| 24 | 3         | 2017-11-03 | 8       | 3       | 3       | 0            | 0           | 0          | 0              |

## Data Warehouse Queries – Code

What is the Total Inventory Value at all Branches?

```
SELECT b.BranchName, sum(a.Inv_Value_Cost) as TotInv_Value
FROM InventoryFactTable a
      Left Join BranchDimension b on a.BranchNbr = b.BranchNbr
GROUP BY b.BranchName
```

|   | BranchName | TotInv_Value |
|---|------------|--------------|
| 1 | Chicago    | 282500       |
| 2 | Minnesota  | 125750       |
| 3 | Wisconsin  | 189750       |

How much Raw Material is currently on Order from each Vendor?

```
SELECT b.VendName, c.RawMaterial, sum(a.QtyOnOrder) as Total_QtyOnOrder
FROM InventoryFactTable a
      Left Join VendorDimension b on a.VendNbr = b.VendNbr
      Left Join ProductDimension c on a.ProdNbr = c.ProdNbr
WHERE a.QtyOnOrder > 0
GROUP BY b.VendName, c.RawMaterial
ORDER BY b.VendName, c.RawMaterial
```

|   | VendName       | RawMaterial          | Total_QtyOnOrder |
|---|----------------|----------------------|------------------|
| 1 | Big Steel Inc  | Hot Rolled Coil      | 180              |
| 2 | Big Steel Inc  | Structural Tubing    | 90               |
| 3 | New Age Metals | Boiler Tubes         | 20               |
| 4 | New Age Metals | Stainless & Aluminum | 20               |
| 5 | Old World Pipe | Big Pipe             | 30               |
| 6 | Old World Pipe | Little Pipe          | 240              |

How much Inventory is currently available for a specific Customer?

```
SELECT b.CustName, c.ProdDesc, sum(a.QtyAvailable) as Total_CurrentlyAvailable
FROM InventoryFactTable a
      Left Join CustomerDimension b on a.CustNbr = b.CustNbr
      Left Join ProductDimension c on a.ProdNbr = c.ProdNbr
WHERE b.SpecialProdNbr IS NOT NULL
GROUP BY b.CustName, c.ProdDesc
ORDER BY b.CustName, c.ProdDesc
```

|   | CustName                   | ProdDesc               | Total_CurrentlyAvailable |
|---|----------------------------|------------------------|--------------------------|
| 1 | Big Rig Digs               | 8 ft Big Pipe          | 50                       |
| 2 | Roller Coasters of America | 2 ft Small Pipe        | 200                      |
| 3 | Roller Coasters of America | 2 ft Small Pipe Angled | 120                      |
| 4 | Valves and Fittings        | 8 ft Big Pipe Angled   | 20                       |