CS3 Technology

# Customer Pricing Generic Inquiry

Prepared by: Shawn Slavin
Date Prepared: 12/07/17

During a recent consulting engagement, a customer asked for an easy way to look up a customer's current, effective price for a range of items.  While there isn't a pre-existing report available in Acumatica, it is possible to build generic inquiry that comes close.

*Note:* Before we begin, I need to give credit where credit is due. While my skills developing generic inquires (GI) have grown rather strong, this GI was created by my friend Tim Rodman while he was working as a business consultant with Acumatica and even he had to reach out to one or more internal 'techno-dev geniuses' like Gabriel Michaud.

For those not familiar with how sales prices are established in Acumatica, we have a white paper covering the details you can pull down from here.

The challenge with compiling all the possible prices for an item on a single line of a report is that except for the items default price, all the remaining prices are stored as individual rows in the same table.

There are potentially seven different types of sales prices for a single item assuming we aren't taking advantage of multiple price breaks. These seven are applied in the following order to determine which is used by Acumatica on a Sales Order or Invoice:

- Customer specific promotional price
- Price Class specific promotional price
- Base promotional price
- Customer specific price
- Price Class specific price
- Base price
- Item Default price

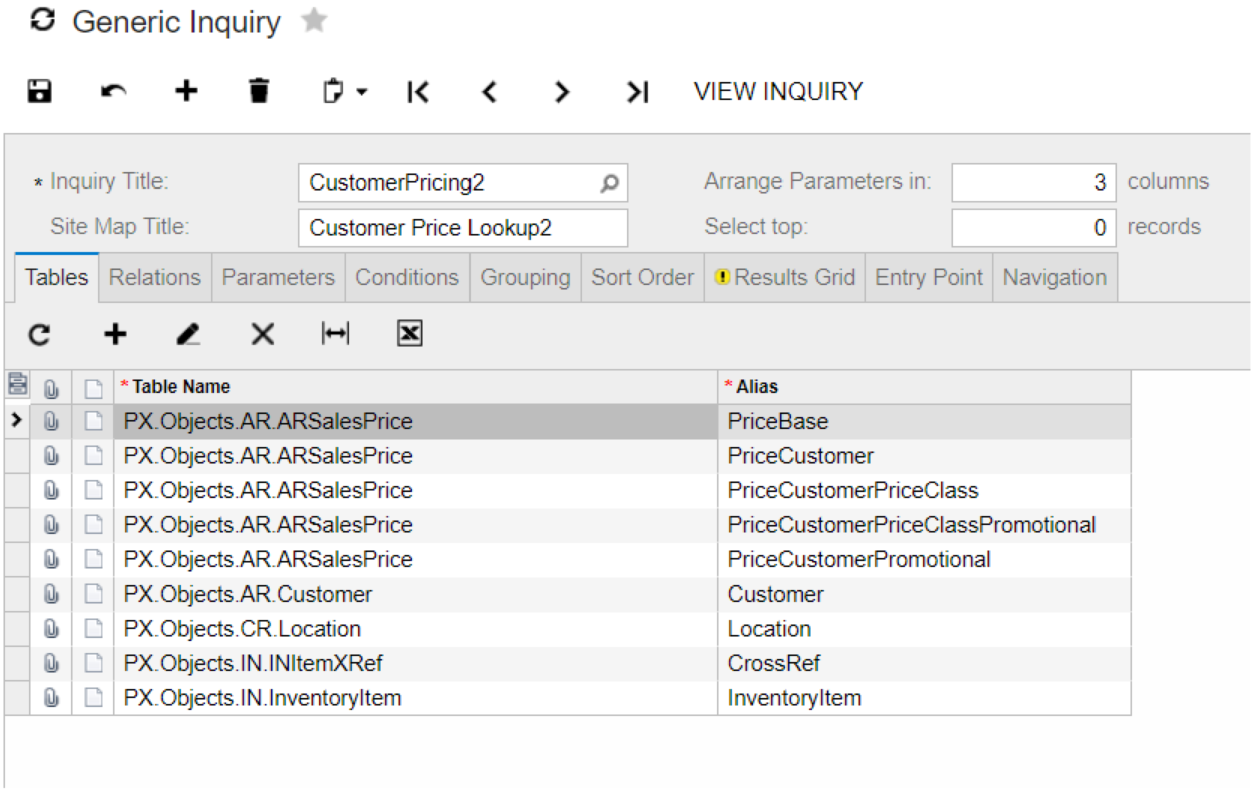
Acumatica does not look for the lowest possible price but rather walks down the list in the order listed above. The first effective price found is used.

means that to get all possible price options on a single row of a generic inquiry, we will need to reference the table holding sales prices (ARSalesPrice) a maximum of six times. The number of times you reference the table will depend on how many of the above sales price types you use in your Acumatica deployment.

In the GI we built and documented below, we use six of the seven options listed above. The only one we did not use was a Base Promotional pricing option.

### Tables

The following shows the tables involved and the aliases assigned.



Inquiry Title: CustomerPricing2

Site Map Title: Customer Price Lookup2

Arrange Parameters in: 3 columns

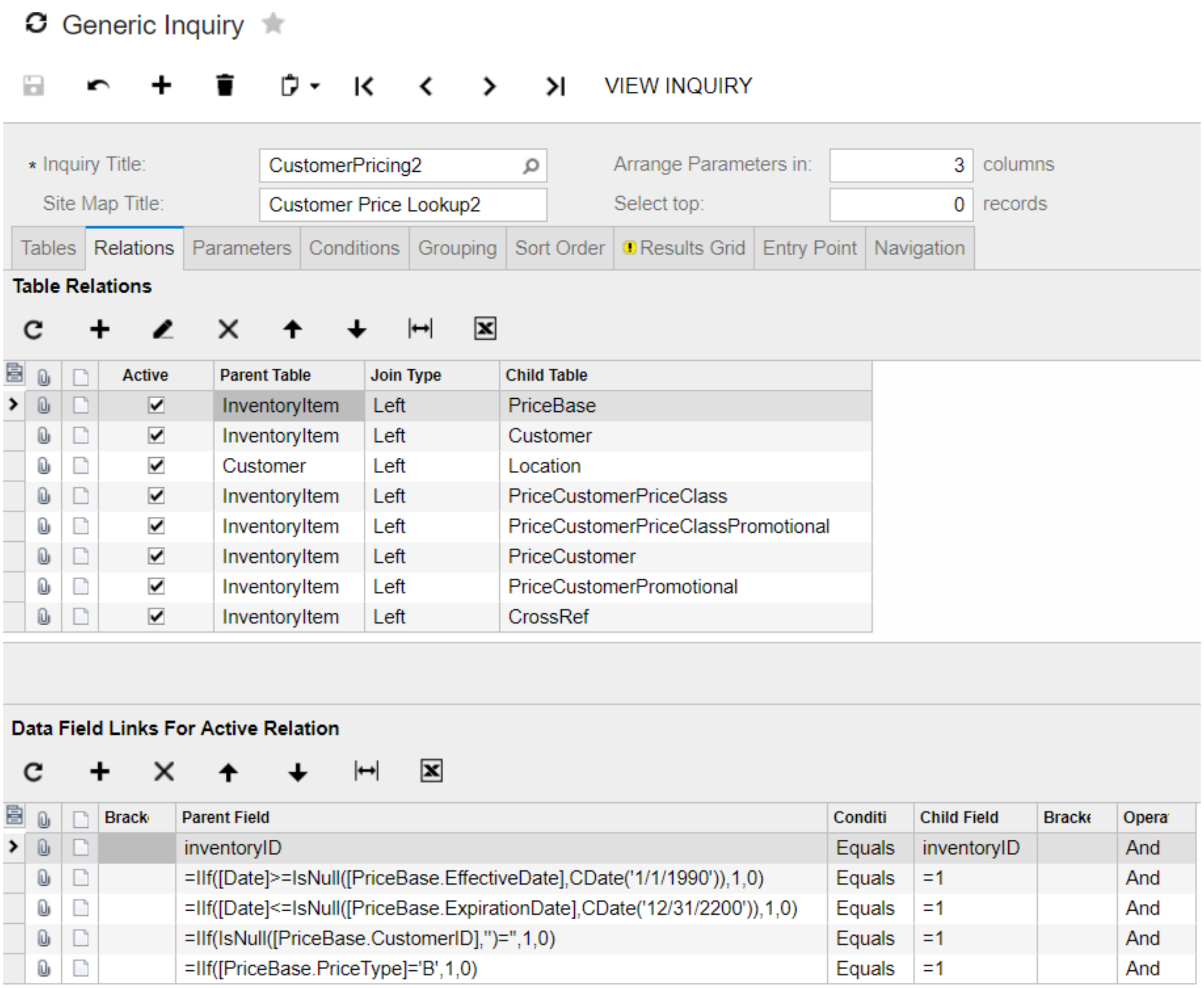
Select top: 0 records

| * Table Name | * Alias |
|---|---|
| PX.Objects.AR.ARSalesPrice | PriceBase |
| PX.Objects.AR.ARSalesPrice | PriceCustomer |
| PX.Objects.AR.ARSalesPrice | PriceCustomerPriceClass |
| PX.Objects.AR.ARSalesPrice | PriceCustomerPriceClassPromotional |
| PX.Objects.AR.ARSalesPrice | PriceCustomerPromotional |
| PX.Objects.AR.Customer | Customer |
| PX.Objects.CR.Location | Location |
| PX.Objects.IN.INItemXRef | CrossRef |
| PX.Objects.IN.InventoryItem | InventoryItem |

## Relations

Unlike any other GI I have ever built; this GI makes very unique use of table relationships to link specific rows from the ARSalesPrice table.

## Base Price



|  |  |  |  |
|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 records |

**Table Relations**

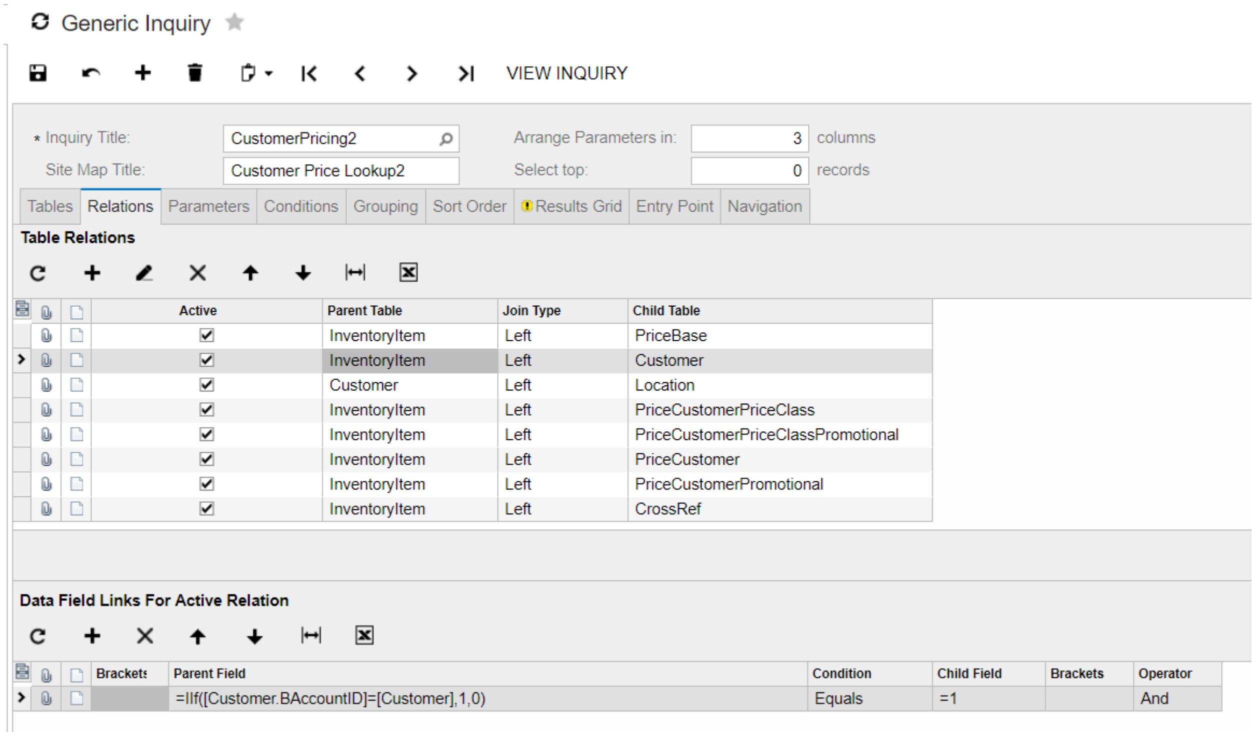
| Active | Parent Table | Join Type | Child Table |
|---|---|---|---|
| ☑ | InventoryItem | Left | PriceBase |
| ☑ | InventoryItem | Left | Customer |
| ☑ | Customer | Left | Location |
| ☑ | InventoryItem | Left | PriceCustomerPriceClass |
| ☑ | InventoryItem | Left | PriceCustomerPriceClassPromotional |
| ☑ | InventoryItem | Left | PriceCustomer |
| ☑ | InventoryItem | Left | PriceCustomerPromotional |
| ☑ | InventoryItem | Left | CrossRef |

**Data Field Links For Active Relation**

| Brack | Parent Field | Conditi | Child Field | Bracke | Opera |
|---|---|---|---|---|---|
|  | inventoryID | Equals | inventoryID |  | And |
|  | =IIf([Date]>=IsNull([PriceBase.EffectiveDate],CDate('1/1/1990')),1,0) | Equals | =1 |  | And |
|  | =IIf([Date]<=IsNull([PriceBase.ExpirationDate],CDate('12/31/2200')),1,0) | Equals | =1 |  | And |
|  | =IIf(IsNull([PriceBase.CustomerID],'')='',1,0) | Equals | =1 |  | And |
|  | =IIf([PriceBase.PriceType]='B',1,0) | Equals | =1 |  | And |

As the above shows, there is a lot of logic built into most of the links. In our example, the user may enter an effective date so valid prices can be determined at any time. The last link condition above limits the returned value to Base Prices only.

## Customer



* Inquiry Title: CustomerPricing2

Site Map Title: Customer Price Lookup2

Arrange Parameters in: 3 columns

Select top: 0 records

**Table Relations**

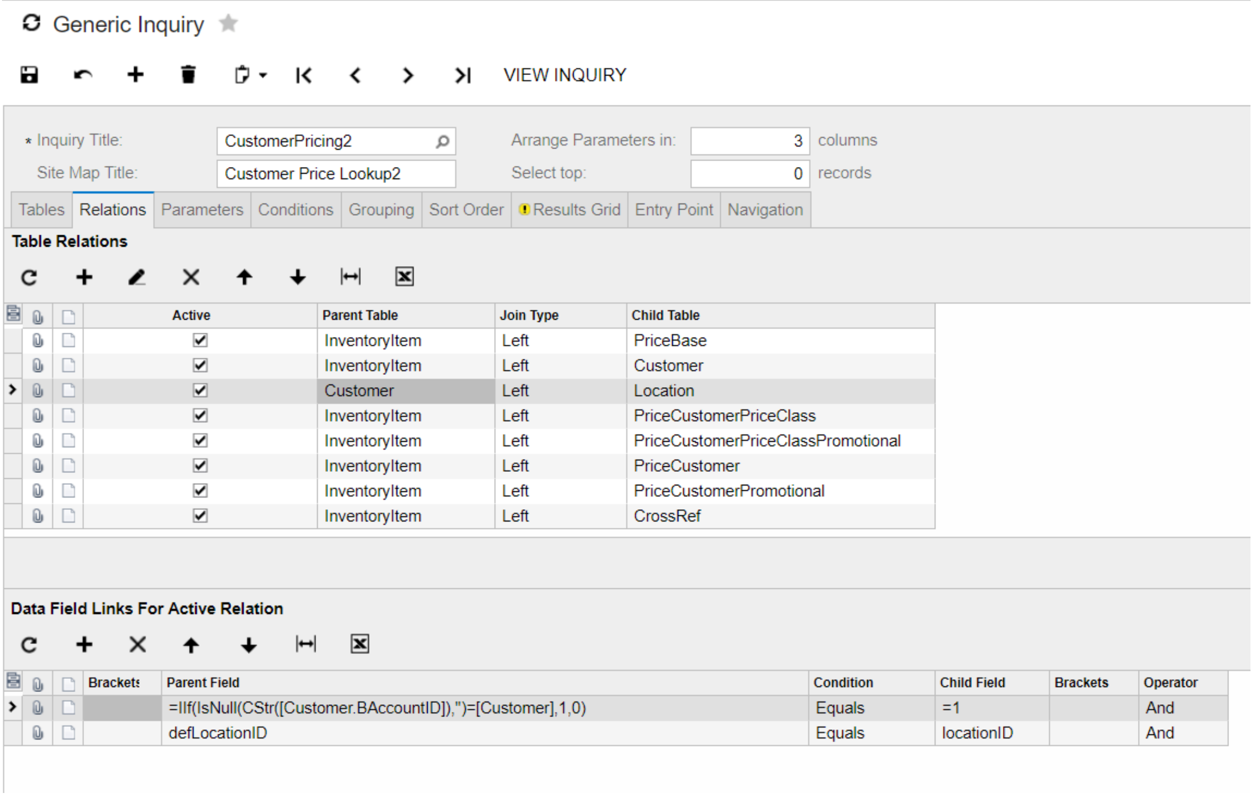
| Active | Parent Table | Join Type | Child Table |
|---|---|---|---|
| ☑ | InventoryItem | Left | PriceBase |
| ☑ | InventoryItem | Left | Customer |
| ☑ | Customer | Left | Location |
| ☑ | InventoryItem | Left | PriceCustomerPriceClass |
| ☑ | InventoryItem | Left | PriceCustomerPriceClassPromotional |
| ☑ | InventoryItem | Left | PriceCustomer |
| ☑ | InventoryItem | Left | PriceCustomerPromotional |
| ☑ | InventoryItem | Left | CrossRef |

**Data Field Links For Active Relation**

| Brackets | Parent Field | Condition | Child Field | Brackets | Operator |
|---|---|---|---|---|---|
| | =IIf([Customer.BAccountID]=[Customer],1,0) | Equals | =1 | | And |

The customer involved is limited to the customer selected a parameter defined below.

## Location



| | | | |
|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 records |

**Table Relations**

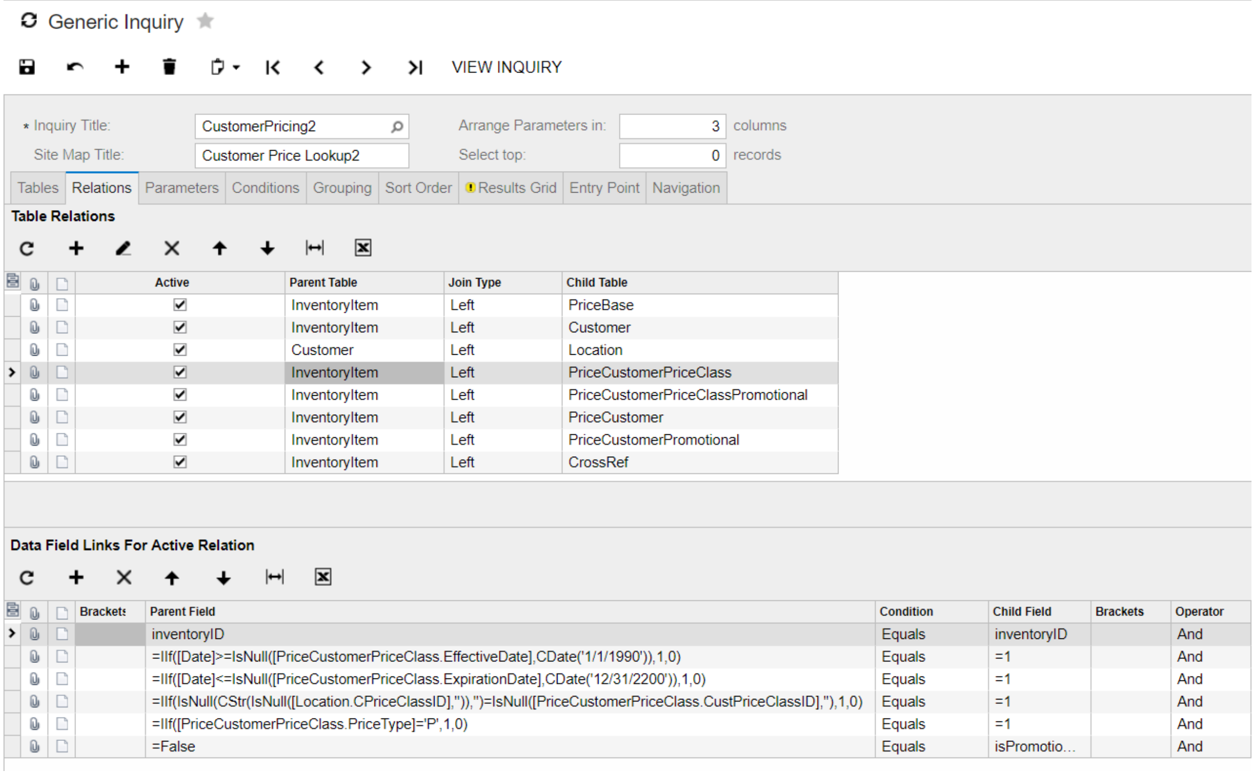
| Active | Parent Table | Join Type | Child Table |
|---|---|---|---|
| ☑ | InventoryItem | Left | PriceBase |
| ☑ | InventoryItem | Left | Customer |
| ☑ | Customer | Left | Location |
| ☑ | InventoryItem | Left | PriceCustomerPriceClass |
| ☑ | InventoryItem | Left | PriceCustomerPriceClassPromotional |
| ☑ | InventoryItem | Left | PriceCustomer |
| ☑ | InventoryItem | Left | PriceCustomerPromotional |
| ☑ | InventoryItem | Left | CrossRef |

**Data Field Links For Active Relation**

| Brackets | Parent Field | Condition | Child Field | Brackets | Operator |
|---|---|---|---|---|---|
| | =IIf(IsNull(CStr([Customer.BAccountID]),'')=[Customer],1,0) | Equals | =1 | | And |
| | defLocationID | Equals | locationID | | And |

Location is required when a customer has multiple ship-to addresses.

## Customer Price Class



| | | | |
|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 records |

Table Relations

| Active | Parent Table | Join Type | Child Table |
|---|---|---|---|
| ☑ | InventoryItem | Left | PriceBase |
| ☑ | InventoryItem | Left | Customer |
| ☑ | Customer | Left | Location |
| ☑ | InventoryItem | Left | PriceCustomerPriceClass |
| ☑ | InventoryItem | Left | PriceCustomerPriceClassPromotional |
| ☑ | InventoryItem | Left | PriceCustomer |
| ☑ | InventoryItem | Left | PriceCustomerPromotional |
| ☑ | InventoryItem | Left | CrossRef |

Data Field Links For Active Relation

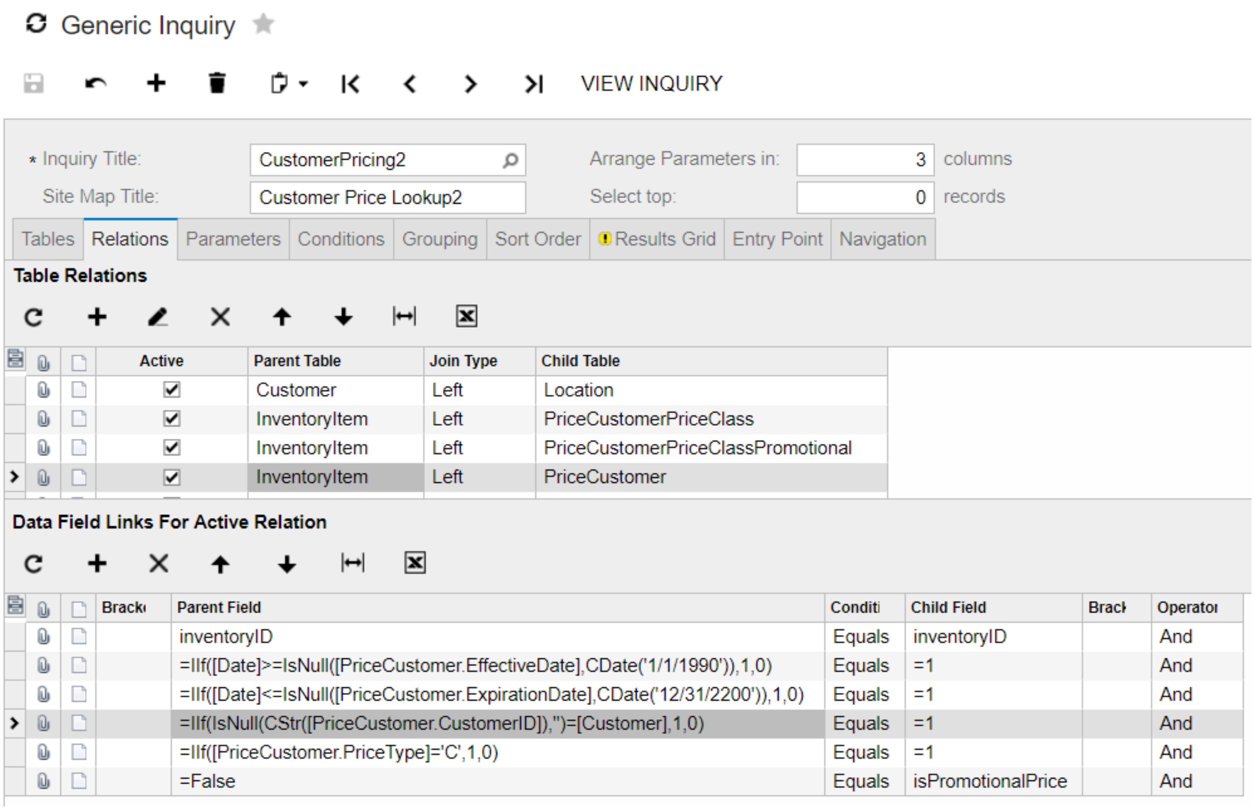
| Brackets | Parent Field | Condition | Child Field | Brackets | Operator |
|---|---|---|---|---|---|
| | inventoryID | Equals | inventoryID | | And |
| | =IIf([Date]>=IsNull([PriceCustomerPriceClass.EffectiveDate],CDate('1/1/1990')),1,0) | Equals | =1 | | And |
| | =IIf([Date]<=IsNull([PriceCustomerPriceClass.ExpirationDate],CDate('12/31/2200')),1,0) | Equals | =1 | | And |
| | =IIf(IsNull(CStr(IsNull([Location.CPriceClassID],'')),'')=IsNull([PriceCustomerPriceClass.CustPriceClassID],''),1,0) | Equals | =1 | | And |
| | =IIf([PriceCustomerPriceClass.PriceType]='P',1,0) | Equals | =1 | | And |
| | =False | Equals | isPromotio… | | And |

As discussed earlier, a date parameter is used to filter by effective dates. The fifth link limits us to Customer Price Class Sales Prices.  The last link ensures we are excluding promotional prices.

Customer Price Class Promotional

This link is identical the previous link except the last condition looks for “=True” value rather than “=False”.

Customer Specific Sales Price



| | | | | |
|---|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 | columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 | records |

Table Relations

| Active | Parent Table | Join Type | Child Table |
|---|---|---|---|
| ☑ | Customer | Left | Location |
| ☑ | InventoryItem | Left | PriceCustomerPriceClass |
| ☑ | InventoryItem | Left | PriceCustomerPriceClassPromotional |
| ☑ | InventoryItem | Left | PriceCustomer |

Data Field Links For Active Relation

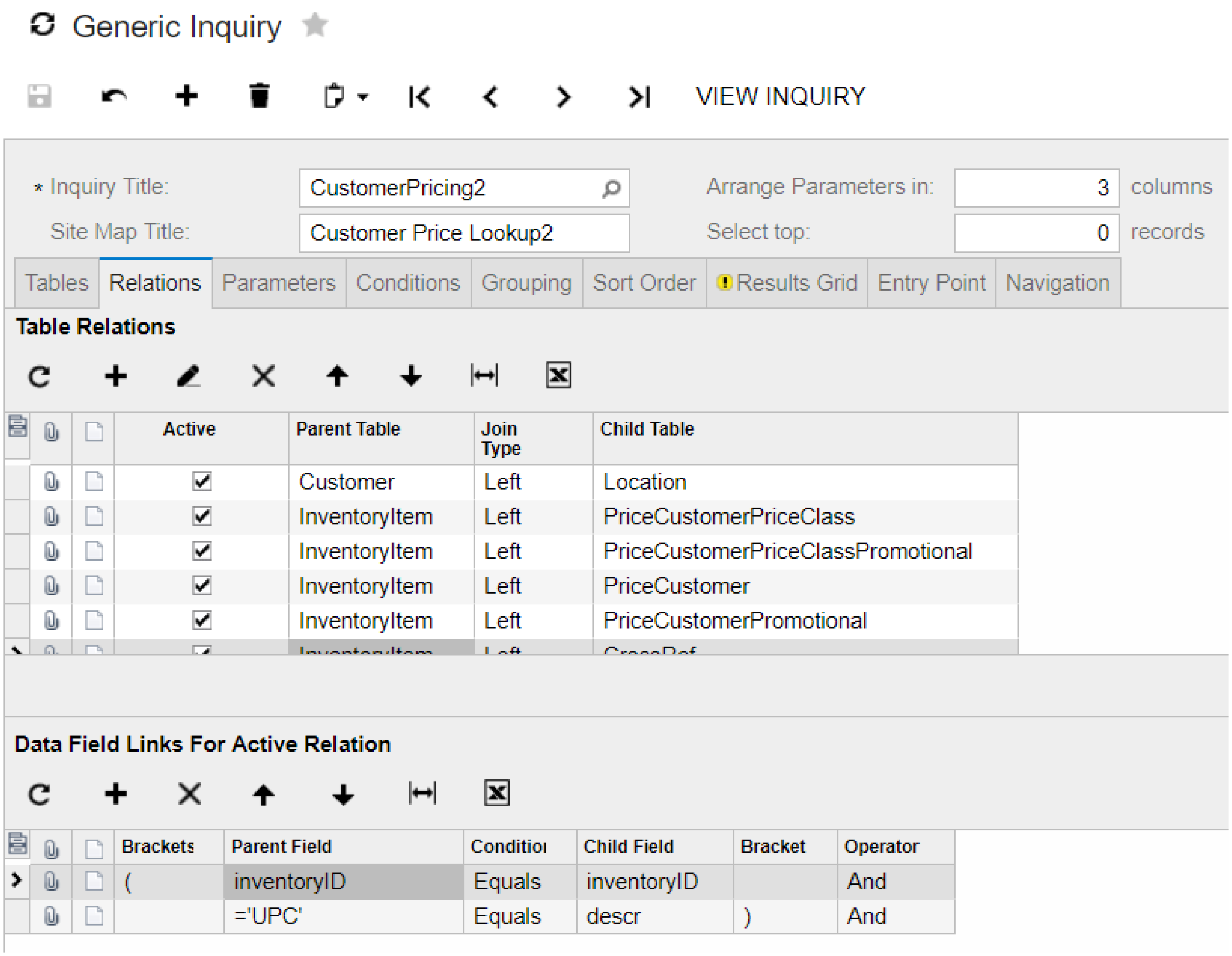
| Brack | Parent Field | Conditi | Child Field | Brack | Operator |
|---|---|---|---|---|---|
| | inventoryID | Equals | inventoryID | | And |
| | =IIf([Date]>=IsNull([PriceCustomer.EffectiveDate],CDate('1/1/1990')),1,0) | Equals | =1 | | And |
| | =IIf([Date]<=IsNull([PriceCustomer.ExpirationDate],CDate('12/31/2200')),1,0) | Equals | =1 | | And |
| | =IIf(IsNull(CStr([PriceCustomer.CustomerID]),'')=[Customer],1,0) | Equals | =1 | | And |
| | =IIf([PriceCustomer.PriceType]='C',1,0) | Equals | =1 | | And |
| | =False | Equals | isPromotionalPrice | | And |

Once again, the linkages filter based using an effective date and limits to non-promotional prices. The fifth linkage establishes that only customer specific sales prices are used.

Customer Specific Promotional Sales Price

Once again, this link is identical to its non-promotional sibling except it looks for a “=True” value in isPromotionalPrice.

Cross Reference



|  |  |  |  |
|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 records |

**Table Relations**

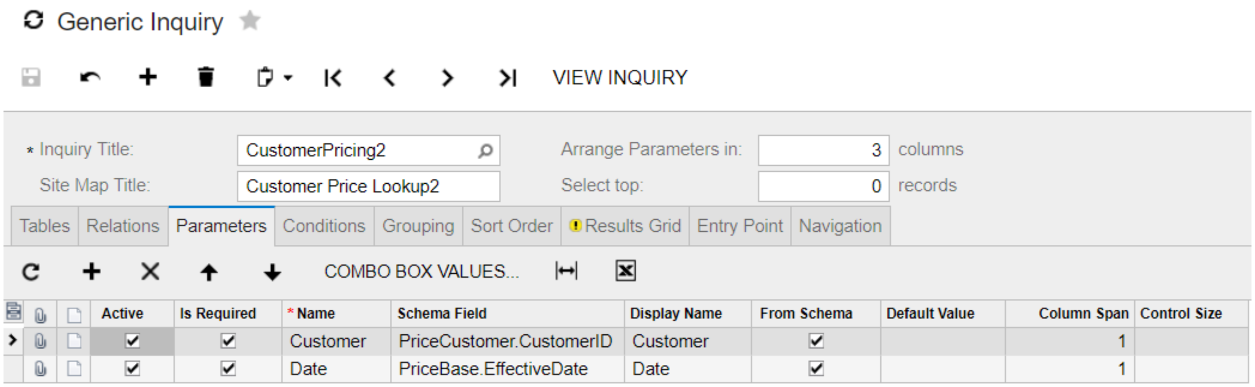
| Active | Parent Table | Join Type | Child Table |
|---|---|---|---|
| ☑ | Customer | Left | Location |
| ☑ | InventoryItem | Left | PriceCustomerPriceClass |
| ☑ | InventoryItem | Left | PriceCustomerPriceClassPromotional |
| ☑ | InventoryItem | Left | PriceCustomer |
| ☑ | InventoryItem | Left | PriceCustomerPromotional |
| ☑ | InventoryItem | Left | CrossRef |

**Data Field Links For Active Relation**

| Brackets | Parent Field | Conditio | Child Field | Bracket | Operator |
|---|---|---|---|---|---|
| ( | inventoryID | Equals | inventoryID |  | And |
|  | ='UPC' | Equals | descr | ) | And |

In our example, we also wanted to display the inventory item's UPC code within the output.

## Parameters

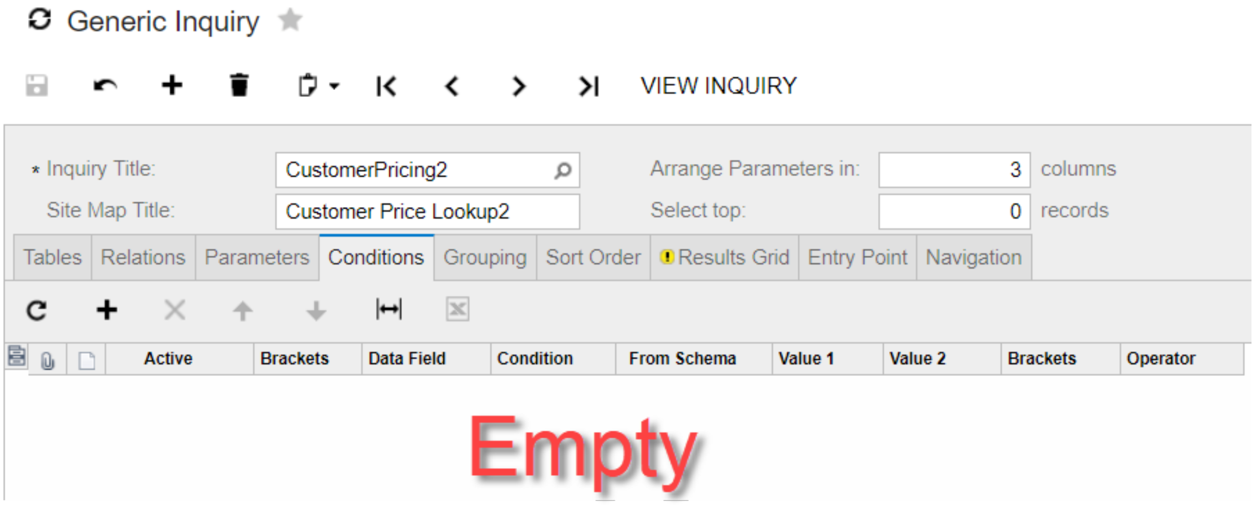


|  |  |  |  |
|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 records |

| Active | Is Required | * Name | Schema Field | Display Name | From Schema | Default Value | Column Span | Control Size |
|---|---|---|---|---|---|---|---|---|
| ☑ | ☑ | Customer | PriceCustomer.CustomerID | Customer | ☑ |  | 1 |  |
| ☑ | ☑ | Date | PriceBase.EffectiveDate | Date | ☑ |  | 1 |  |

Two parameters are required to limit our rows to a specific customer and a specific effective date.

## Conditions



* Inquiry Title: CustomerPricing2

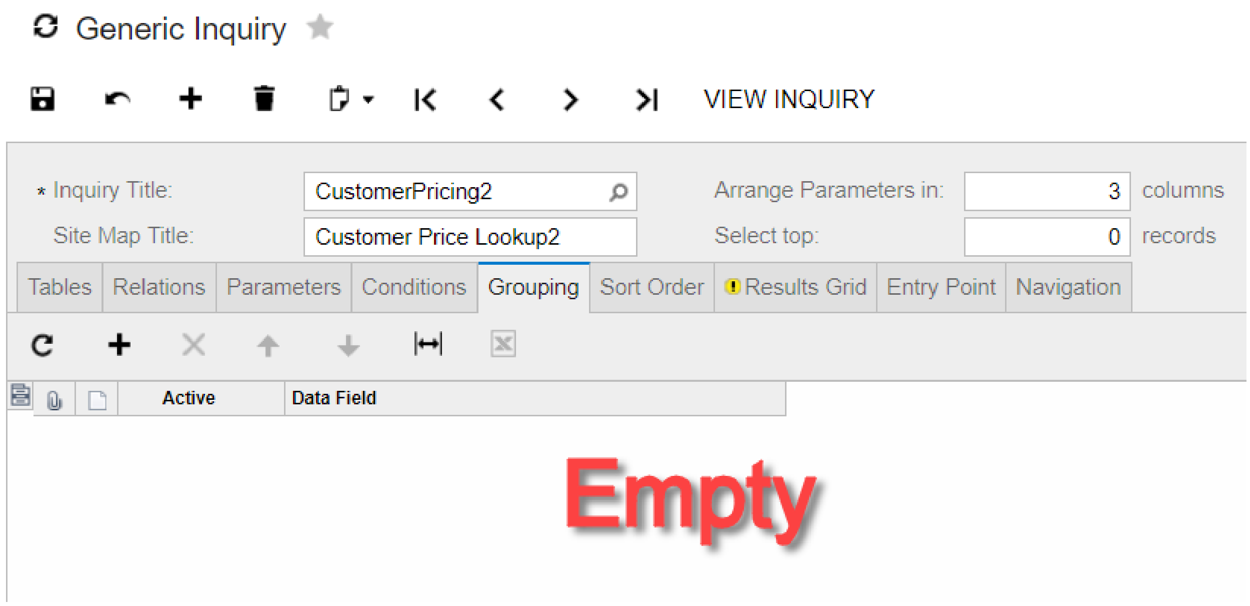
Site Map Title: Customer Price Lookup2

Arrange Parameters in: 3 columns

Select top: 0 records

Typically, parameters are used on the conditions tab to limit the rows returned. However, in this GI, the parameters were used to limit rows through link conditions.

Grouping



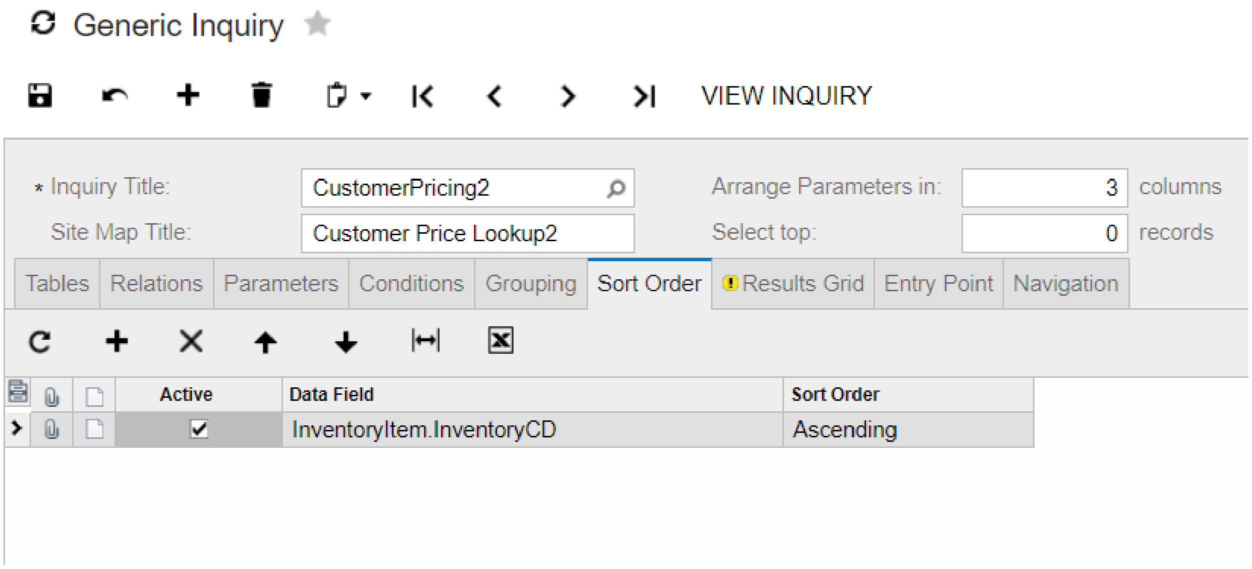
* Inquiry Title: CustomerPricing2

Site Map Title: Customer Price Lookup2

Arrange Parameters in: 3 columns

Select top: 0 records

## Sort Order



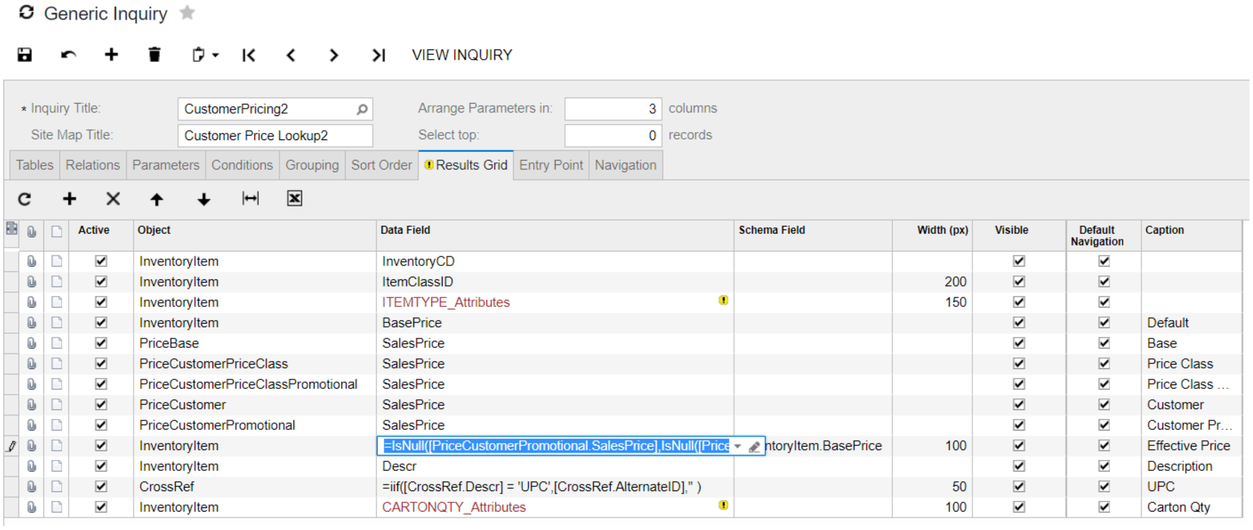
Generic Inquiry

VIEW INQUIRY

| | | | |
|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 records |

| Active | Data Field | Sort Order |
|---|---|---|
| ☑ | InventoryItem.InventoryCD | Ascending |

## Results Grid



Generic Inquiry

VIEW INQUIRY

| | | | |
|---|---|---|---|
| * Inquiry Title: | CustomerPricing2 | Arrange Parameters in: | 3 columns |
| Site Map Title: | Customer Price Lookup2 | Select top: | 0 records |

| Active | Object | Data Field | Schema Field | Width (px) | Visible | Default Navigation | Caption |
|---|---|---|---|---|---|---|---|
| ☑ | InventoryItem | InventoryCD | | | ☑ | ☑ | |
| ☑ | InventoryItem | ItemClassID | | 200 | ☑ | ☑ | |
| ☑ | InventoryItem | ITEMTYPE_Attributes | | 150 | ☑ | ☑ | |
| ☑ | InventoryItem | BasePrice | | | ☑ | ☑ | Default |
| ☑ | PriceBase | SalesPrice | | | ☑ | ☑ | Base |
| ☑ | PriceCustomerPriceClass | SalesPrice | | | ☑ | ☑ | Price Class |
| ☑ | PriceCustomerPriceClassPromotional | SalesPrice | | | ☑ | ☑ | Price Class ... |
| ☑ | PriceCustomer | SalesPrice | | | ☑ | ☑ | Customer |
| ☑ | PriceCustomerPromotional | SalesPrice | | | ☑ | ☑ | Customer Pr... |
| ☑ | InventoryItem | =IsNull([PriceCustomerPromotional.SalesPrice],IsNull([Price | ntoryItem.BasePrice | 100 | ☑ | ☑ | Effective Price |
| ☑ | InventoryItem | Descr | | | ☑ | ☑ | Description |
| ☑ | CrossRef | =iif([CrossRef.Descr] = 'UPC',[CrossRef.AlternateID],'' ) | | 50 | ☑ | ☑ | UPC |
| ☑ | InventoryItem | CARTONQTY_Attributes | | 100 | ☑ | ☑ | Carton Qty |

The results grid is pretty straight forward. However, determining the effective date requires us to build a formula to follow the hierarchy outlined earlier.  This formula is as follows:

```
=IsNull([PriceCustomerPromotional.SalesPrice],
    IsNull([PriceCustomer.SalesPrice],
        IsNull([PriceCustomerPriceClassPromotional.SalesPrice],
            IsNull([PriceCustomerPriceClass.SalesPrice],
                IsNull([PriceBase.SalesPrice],[InventoryItem.BasePrice])))))
```

Entry Point and Navigation

These tabs are not used in this inquiry

## In Conclusion

As stated at the start of the white paper, this is generic inquire uses the most complex table join logic I have ever encountered.  The benefit of working through this GI is the limitless number of possibilities it reveals for those with creative minds.

Enjoy and explore!