Knowledgebase > Lasernet FO Connector > Lasernet FO Connector FAQs > How Do I Build a Query?

How Do I Build a Query?

Alice Petruzzella - 2024-09-13 - Comments (0) - Lasernet FO Connector FAQs

Lasernet for Dynamics 365

If you wish to build a Lasernet query to extract data from Dynamics 365 Finance and Operations, it is first necessary to find out where in the system that data is. Once the location of the data is known, the query must be built in a way that enables it to reach the data. This requires knowing how tables are linked together within FO.

Sometimes, this information is already known or can be easily worked out. For example, customer information is stored in the CustTable. However, when it is not so obvious, there are ways to find out.

Find the Table and Field Names

When building a query, the first thing to do is finding the data. Here are some suggestions about how to find out the exact location of the data that your customer wants to see in their report.

Ask for Specific Requirements

The best way to make sure that the data in the query is exactly what the customer wants, is to ask them to specify the exact table and field. An example of a common request from a customer would be:

“Add the date to the Free Text Invoice report”

However, in the Free Text Invoice header, we can see that there are actually six possible dates:

What we should, instead, be asking for is a more specific requirement. The requirement should show the table name, field name and, where possible, a screenshot showing the location of the field in the user interface:

“Add CustInvoiceTable.DocumentDate field to the FreeTextInvoiceReplacement query”

Having a well defined requirement, the extra effort needed to work out field names and relations is avoided.

Right-Click in the User Interface

If the customer cannot provide detailed requirements, it is possible to find out wich table and field names are needed based on a screenshot that the customer provides. For example, if the customer needs to add the Invoice Text to a Free Text Invoice, the customer's requirement would read:

“Please add the Invoice text to the Invoice lines on the Free Text Invoice Replacement report, as indicated in the screenshot:”

In this case, it is easy to get the table and field information.

1. Navigate to the screen shown by the customer in the screenshot. In this case, **Accounts Receivable** > **Invoices** > **All Free Text Invoices**.

2. Find the field on the form, and right click.

3. Point to **Form Information** and, when the flyout is shown, click **Form Name:xxxx**.

The **Form Information** will be displayed, which will show the **DataSource** and **Data Field**.

Warning

The name of the **DataSource** is the value between brackets. The two values are normally the same, but can sometimes vary. Therefore, ensure that you look at the value between brackets:

Find Table Relations

The table and field names can be found by right clicking in the system to find the data, or by asking the customer to be specific, however once these tables are identified, it is necessary to know how to link them in the query.

Browser Add In Table Browser

Browser add-ins are available to access the tables in the back end of FO. This one, for example, works for Google Chrome:

https://chrome.google.com/webstore/detail/table-browser-caller-for/khhnlfooiiffofchhdhjopgpegcjlodk

It will allow you to provide a URL, an entity, and log in, in order to see the tables in an environment.

The user can then either search for a specific table, or click on **Table List** to view a list of all the different tables in the system.

This can be useful when trying to understand and build relations as it also includes a search function that makes the process of finding intermediate tables easier. Ffor instance, between logistics tables: a relation may have been built to get an email address from the LogisticsElectronicAddress table, but the Role referenced in that table is just a RecID. By searching for “Logisticselec” in the table browser, we find another table specifically for this Role, from which we can get the label we need.

By opening the two tables, it is possible to cross reference the data and find the common features to identify how the tables are linked.

Native Table Browser

While a browser add-in is useful, some people are not able to or do not wish to install one. In this case, it is possible to use a native table browser within Finance and Operations. To do so, append the following text to the URL of your FO environment:

SysTableBrowser&tableName=<tableName>

Example

If the URL of an internal test environment were:

https://deliverytest-2-pu561bfff4965d53dee0aos.axcloud.dynamics.com/?cmp=usmf&mi=

To view the contents of the VendTable, it would be necessary to use the following URL:

https://deliverytest-2-pu561bfff4965d53dee0aos.axcloud.dynamics.com/?cmp=usmf&mi=SysTableBrowser&tablename=VendTable

Which would display the contents of the VendTable as:

This option is useful if the table name is known. However, it is not possible to search for a table as shown for the browser add-in.

Access to Visual Studio/SQL

One easy way to find the table relations is to ask a developer, or someone with access to Visual Studio or SQL. They will be able to tell you the relation between tables. This can save much work if many relations must be defined, or can be a good solution if the user already has access.

Most Used Relations

1. Customer

A. Full name

DirPartyTable.Party

Example from CustTable:

CustTable.RecId <-> DirPartyTable.RecId.

B. First name, middle name last name

DirPersonName.FirstName

DirPersonName.MiddleName

DirPersonName.LastName

From CustTable

CustTable.Party <-> DirPartyTable.RecId

DirPartyTable.RecId <->DirPerson.RecId

DirPerson.RecId <->DirPersonName.Person

C. Customer title (Mr. Mrs. Ms)

DirNameAffix.Affix

Example from CustTable

CustTable.Party <-> DirPartyTable.RecId

DirPartyTable.RecId <->DirPerson.RecId

DirPerson.PersonalTitle <-> DirNameAffix.RecId

Eksempel

Custaccount 122590

CustTable.Party=5637198169

DirPartyTable.RecId=5637198169 (It's therefore possible to jump directly from CustTable to DirPerson)

DirPerson.RecId=5637198169

DirPerson.PersonalTitle =5637146084

DirNameAffix.RecId=5637146084

CustAccount 122590 == Ms J

D. Contact informations (Email, phone number)

DirPartyTable.PrimaryContactEmail

DirPartyTable.PrimaryContactPhone

(The ones with checkmark in primary)

Example from CustTable

CustTable_Party <-> DirPartyTable_RecId

DirPartyTable.PrimaryContactPhone

E. Address

LogisticsPostalAddress.Address (Address in one string)

LogisticsPostalAddress.Street

LogisticsPostalAddress.ZipCode

LogisticsPostalAddress.City

LogisticsPostalAddress.CountryRegionId

Example from CustTable

CustTable_Party <-> DirPartyTable_RecId

DirPartyTable_PrimaryAddressLocation <-> LogisticsPostalAddress.Location

Country name (Russia instead of RUS)

LogisticsAddressCountryRegionTranslation.ShortName

LogisticsPostalAddress.CountryRegionId <-> LogisticsAddressCountryRegionTranslation.CountryRegionId

F. Delivery address

DirPartyPostalAddressView.Address (Address in one string)

(PartyLocation == 5637145327 (hardcodet for type "delivery" se knytning til type længere nede)

DirPartyPostalAddressView.City

DirPartyPostalAddressView.Street

DirPartyPostalAddressView.ZipCode

Knytning fra CustTable

CustTable.Party <-> DirPartyPostalAddressView.Party

Type (Delivery, Business etc)

LogisticsLocationRole_Type

DirPartyPostalAddresView_PartyLocation <-> LogisticsLocationRole_RecId

2. Translations

LanguageTxt.Txt

Example from CashDiscCode description added to the ProjectInvoice:

ProjInvoiceJour.CashDiscCode == CashDisc.CashDiscCode

CashDisc.RecId == LanguageTxt.TxtRecId

CashDisc.TableId == LanguageTxt.TableRecId

Calculated.ReportLanguage == LanguageId

Example from DlvTerm description added on a report in report language

CustTable.DlvTerm == DlvTerm.Code

DlvTerm.RecId == LanguageTxt.TxtRecId

DlvTerm.TableId == LanguageTxt.TableRecId

Calculated.ReportLanguage == LanguageId

3. Prospects

In AX common everyone starts as prospects and thereefter promoted to either customer or vendor.

Relation from prospect to customer is therefore relevant:

SmmBusRelTable.Party == CustTable.Party

(SmmBusRelTable is table for relevant prospectfields)

Bonus information

Field ”BusRelTypeId” indicates if the prospect is still prospect or has been promoted to customer or vendor.

4. Contacts Title, Profession

ContactPerson.Title

ContactPerson .Profession

ContactPerson_Function

Example from CustTable:

CustTable.AccountNum <-> ContactPerson.CustAccount

Bonus info:

”JobTitle” is a dropdownfield and therefor better for datadiciplin, where as ”Proffesion” is a ”freetextfield”.

5. CompanyInfo Contact Information

CompanyInfo.invoiceAddress().Address

CompanyInfo.postalAddress().Address

CompanyInfo.email

Example from Ledger:

CompanyInfo.RecId == Ledger.PrimaryForLegalEntity

Example from SalesTable:

CompanyInfo.DataArea == SalesTable.dataAreaId

Example from CustTable:

CompanyInfo.DataArea == CustTable.dataAreaId

6. BankAccountTable Information

BankAccountTable.AccountNum

BankAccountTable.Name

BankAccountTable.CurrencyCode

BankAccountTable.SWIFTNo

BankAccountTable.IBAN

BankAccountTable.GiroContract

Example from CompanyInfo:

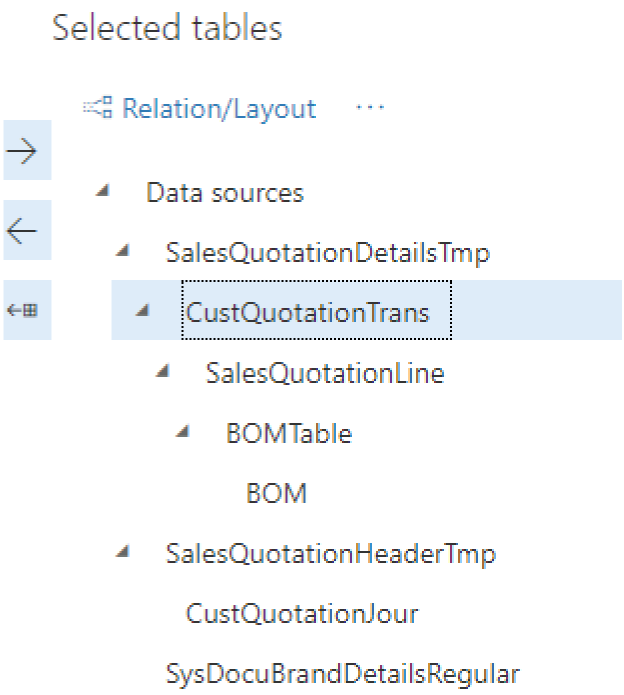
BankAccountTable.AccountId == CompanyInfo.Bank

BankAccountTable.dataAreaId == CompanyInfo.DataArea

7. Add Bomlines to Report

There can be several active BomVersions to the InvenDimId is key to get the correct Bom.

From Quotation:



Selected tables

- Data sources
  - SalesQuotationDetailsTmp
    - CustQuotationTrans
      - SalesQuotationLine
        - BOMTable
          - BOM
  - SalesQuotationHeaderTmp
    - CustQuotationJour
  - SysDocuBrandDetailsRegular

Add CustQuotationTrans

SalesQuotationDetailsTmp.LACTransRefRecId == CustQuotationTrans.RecId (OuterJoin 1:n)

CustQuotationTrans.InventDim == SalesQuotationLine.InventDim (OuterJoin 1:n)

CustQuotationTrans.InventTransId == SalesQuotationLine.InventTransId (OuterJoin 1:n)

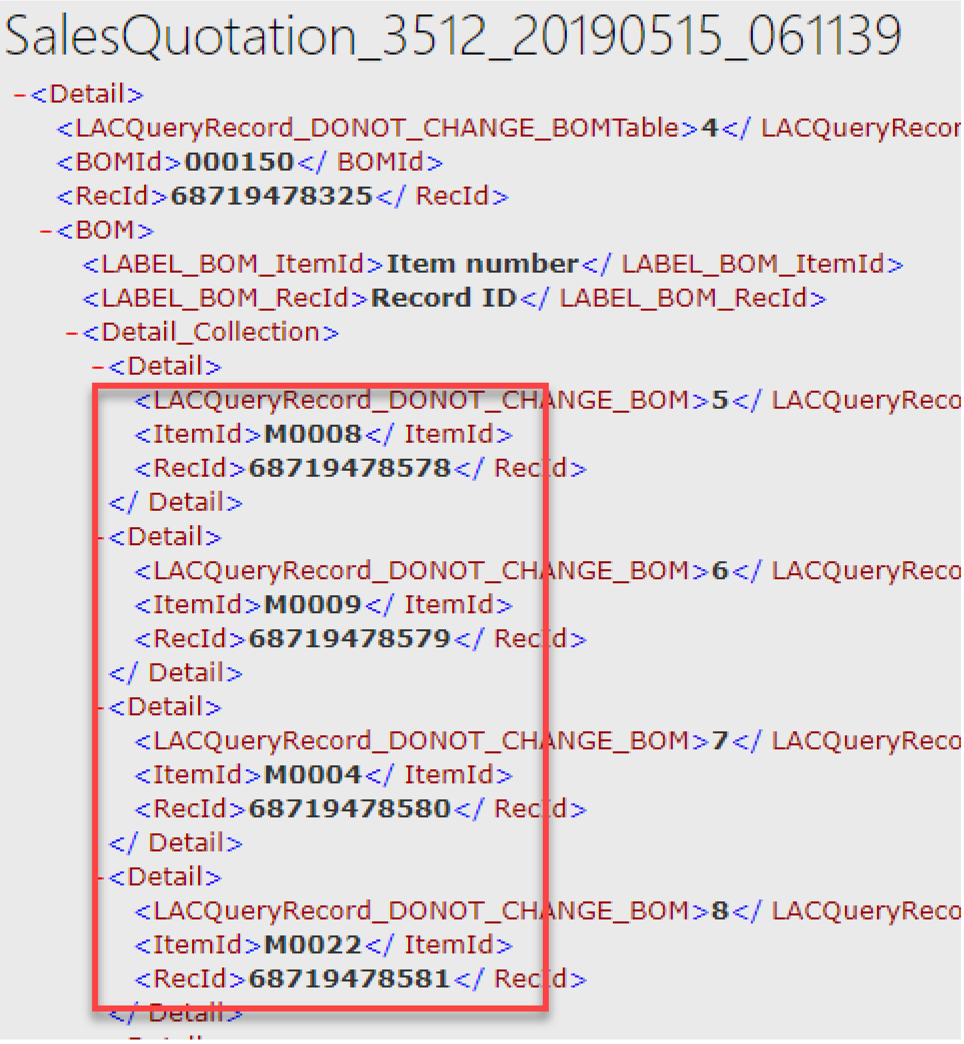
(Cause BOMId is on that table)

SalesQuotationLine.ItemBOMId == BOMTable.BOMId (OuterJoin 1:n)

BOMTable.BOMId == BOM.BOMId (OuterJoin 1:n)

The BOM item number is called BOMTable.BOMId

The BOM lines Item number is called BOM.ItemId



SalesQuotation_3512_20190515_061139

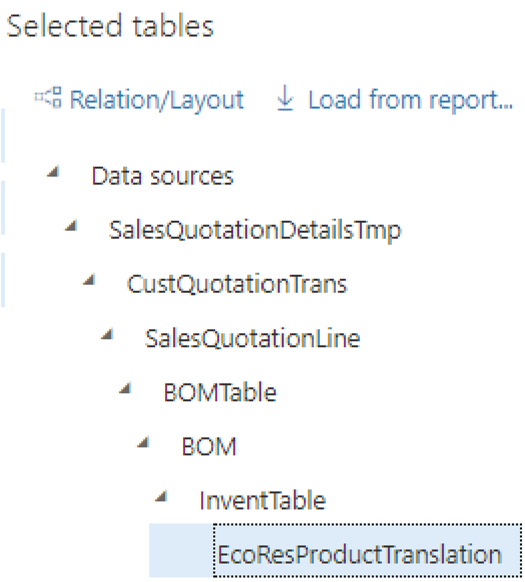
```
-<Detail>
   <LACQueryRecord_DONOT_CHANGE_BOMTable>4</ LACQueryRecor
   <BOMId>000150</ BOMId>
   <RecId>68719478325</ RecId>
 -<BOM>
     <LABEL_BOM_ItemId>Item number</ LABEL_BOM_ItemId>
     <LABEL_BOM_RecId>Record ID</ LABEL_BOM_RecId>
   -<Detail_Collection>
     -<Detail>
         <LACQueryRecord_DONOT_CHANGE_BOM>5</ LACQueryReco
         <ItemId>M0008</ ItemId>
         <RecId>68719478578</ RecId>
       </ Detail>
      -<Detail>
         <LACQueryRecord_DONOT_CHANGE_BOM>6</ LACQueryReco
         <ItemId>M0009</ ItemId>
         <RecId>68719478579</ RecId>
       </ Detail>
      -<Detail>
         <LACQueryRecord_DONOT_CHANGE_BOM>7</ LACQueryReco
         <ItemId>M0004</ ItemId>
         <RecId>68719478580</ RecId>
       </ Detail>
      -<Detail>
         <LACQueryRecord_DONOT_CHANGE_BOM>8</ LACQueryReco
         <ItemId>M0022</ ItemId>
         <RecId>68719478581</ RecId>
       </ Detail>
```

Add name to the bom:

BOM.ItemId == InventTable.ItemId

InventTable.Product == EcoResProductTranslation.Product

SalesQuotationTable.LanguageId == EcoResProductTranslation.languageId (SalesQuotationTable is added to SalesQuotationLine)

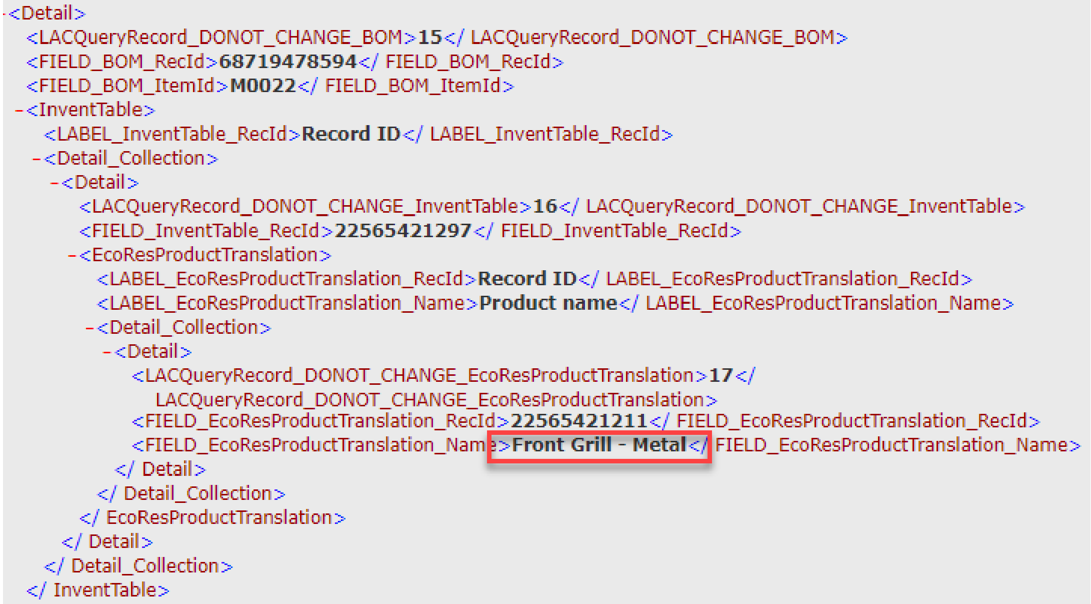


Selected tables

Relation/Layout Load from report...

- Data sources
  - SalesQuotationDetailsTmp
    - CustQuotationTrans
      - SalesQuotationLine
        - BOMTable
          - BOM
            - InventTable
              - EcoResProductTranslation

Name is EcoResProductTranslation.Name

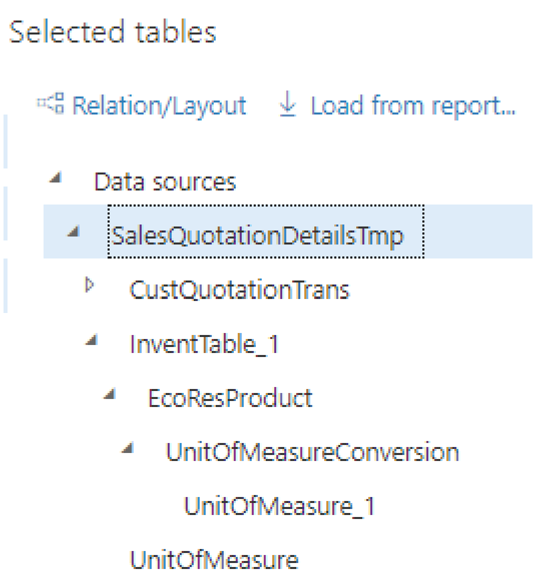


```
-<Detail>
  <LACQueryRecord_DONOT_CHANGE_BOM>15</ LACQueryRecord_DONOT_CHANGE_BOM>
  <FIELD_BOM_RecId>68719478594</ FIELD_BOM_RecId>
  <FIELD_BOM_ItemId>M0022</ FIELD_BOM_ItemId>
 -<InventTable>
    <LABEL_InventTable_RecId>Record ID</ LABEL_InventTable_RecId>
   -<Detail_Collection>
     -<Detail>
        <LACQueryRecord_DONOT_CHANGE_InventTable>16</ LACQueryRecord_DONOT_CHANGE_InventTable>
        <FIELD_InventTable_RecId>22565421297</ FIELD_InventTable_RecId>
       -<EcoResProductTranslation>
          <LABEL_EcoResProductTranslation_RecId>Record ID</ LABEL_EcoResProductTranslation_RecId>
          <LABEL_EcoResProductTranslation_Name>Product name</ LABEL_EcoResProductTranslation_Name>
         -<Detail_Collection>
           -<Detail>
              <LACQueryRecord_DONOT_CHANGE_EcoResProductTranslation>17</
                LACQueryRecord_DONOT_CHANGE_EcoResProductTranslation>
              <FIELD_EcoResProductTranslation_RecId>22565421211</ FIELD_EcoResProductTranslation_RecId>
              <FIELD_EcoResProductTranslation_Name>Front Grill - Metal</ FIELD_EcoResProductTranslation_Name>
            </ Detail>
          </ Detail_Collection>
        </ EcoResProductTranslation>
      </ Detail>
    </ Detail_Collection>
  </ InventTable>
```

8. Add UnitOfmeasureConversion

Let's say you want to display the product from unit and factor on a report:

(from unit) 1 box = 4 ea (to unit)



Selected tables

- Data sources
  - SalesQuotationDetailsTmp
    - CustQuotationTrans
    - InventTable_1
      - EcoResProduct
        - UnitOfMeasureConversion
          - UnitOfMeasure_1
    - UnitOfMeasure

Add UnitOfMeasure

SalesQuotationDetailsTmp.SalesUnitTxt == UnitOfMeasure.Symbol(InnerJoin 1:1)

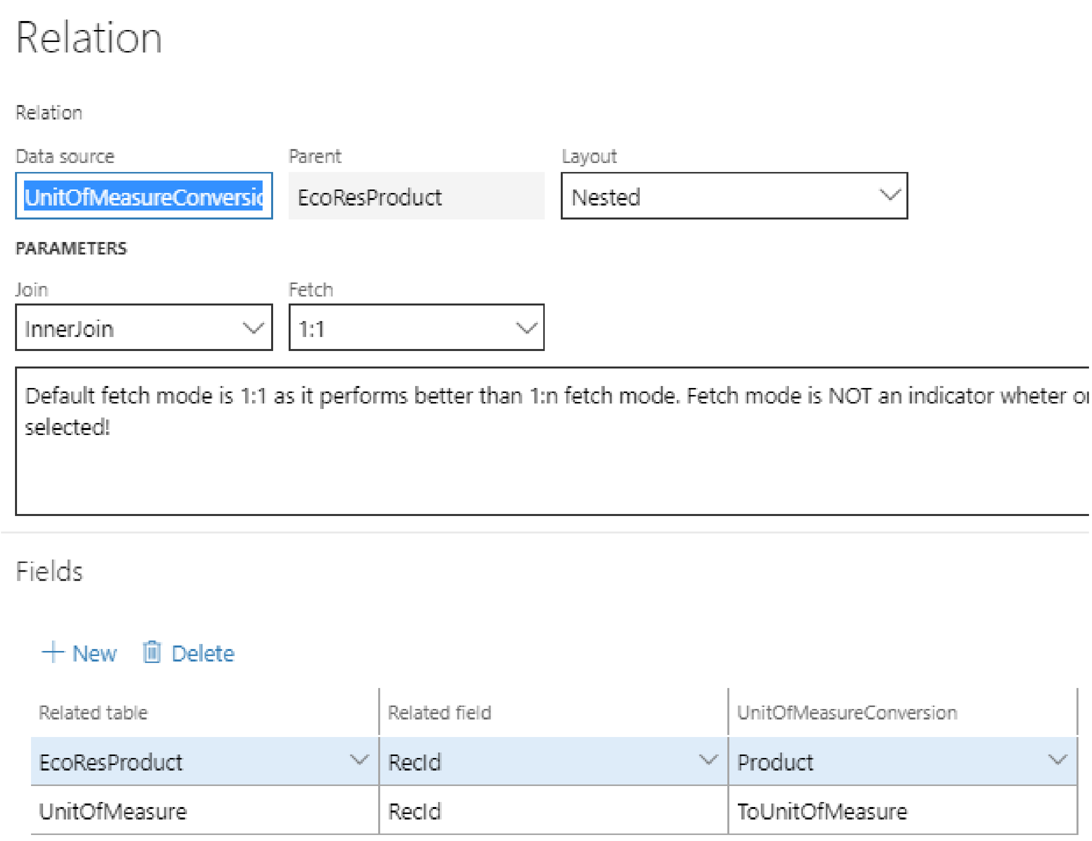
Add InventTable

SalesQuotationDetailsTmp.ItemId == InventTable.ItemId(InnerJoin 1:n)

EcoResProduct.RecId == InventTable.Product (InnerJoin 1:1)

UnitOfMeasureConversion.Product == EcoResProduct.RecId (InnerJoin 1:1)

UnitOfMeasure.RecId == UnitOfMeasureConversion.ToUnitOfMeasure (-||-)



Relation

Relation

Data source: UnitOfMeasureConversic

Parent: EcoResProduct

Layout: Nested

PARAMETERS

Join: InnerJoin

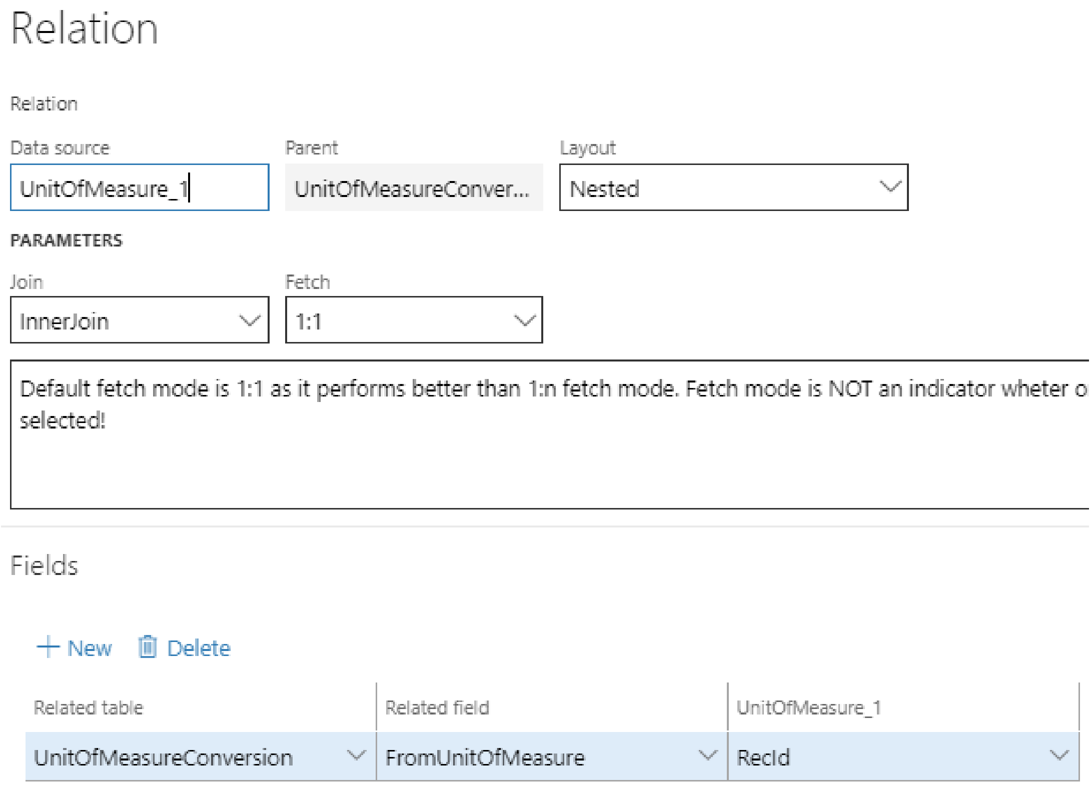
Fetch: 1:1

Default fetch mode is 1:1 as it performs better than 1:n fetch mode. Fetch mode is NOT an indicator wheter or selected!

Fields

Related table	Related field	UnitOfMeasureConversion
EcoResProduct	RecId	Product
UnitOfMeasure	RecId	ToUnitOfMeasure

UnitOfMeasure_1.RecId == UnitOfMeasureConversion.FromUnitOfMeasure



Relation

Relation

Data source: UnitOfMeasure_1

Parent: UnitOfMeasureConver...

Layout: Nested

PARAMETERS

Join: InnerJoin

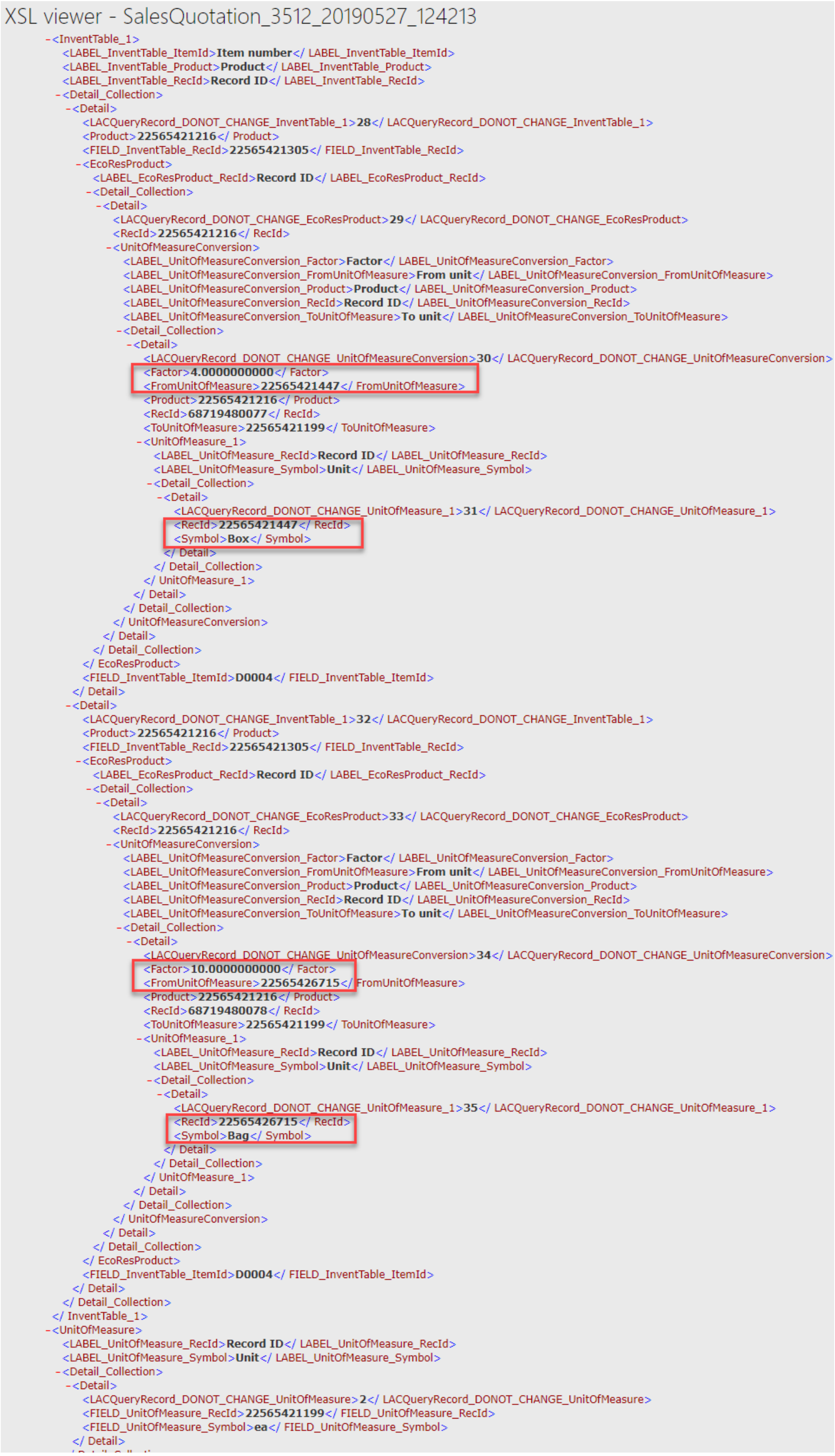
Fetch: 1:1

Default fetch mode is 1:1 as it performs better than 1:n fetch mode. Fetch mode is NOT an indicator wheter o selected!

Fields

Related table	Related field	UnitOfMeasure_1
UnitOfMeasureConversion	FromUnitOfMeasure	RecId

Result:



XSL viewer - SalesQuotation_3512_20190527_124213

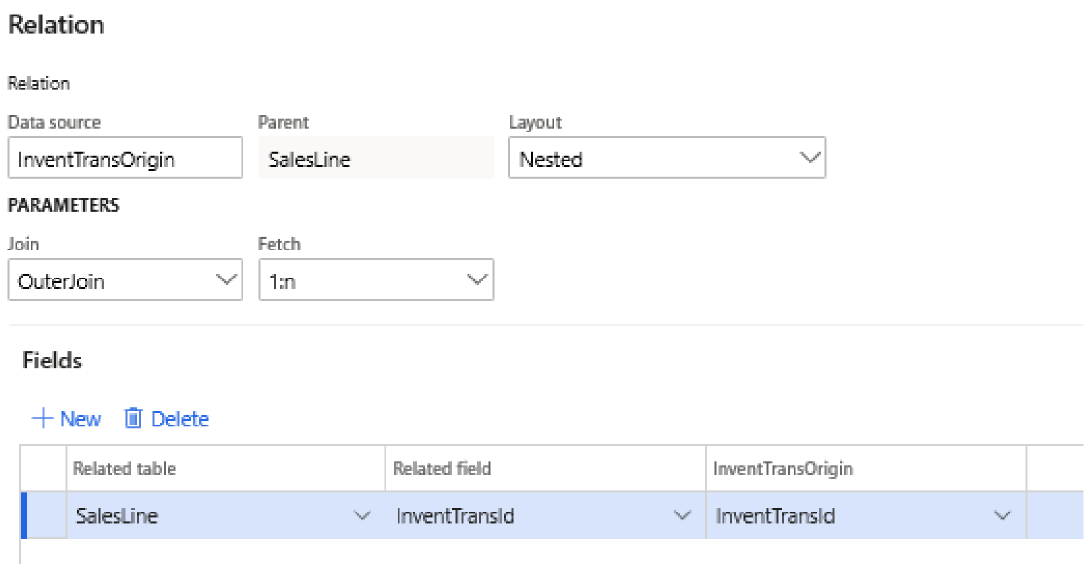
```
- <InventTable_1>
    <LABEL_InventTable_ItemId>Item number</ LABEL_InventTable_ItemId>
    <LABEL_InventTable_Product>Product</ LABEL_InventTable_Product>
    <LABEL_InventTable_RecId>Record ID</ LABEL_InventTable_RecId>
  - <Detail_Collection>
    - <Detail>
        <LACQueryRecord_DONOT_CHANGE_InventTable_1>28</ LACQueryRecord_DONOT_CHANGE_InventTable_1>
        <Product>22565421216</ Product>
        <FIELD_InventTable_RecId>22565421305</ FIELD_InventTable_RecId>
      - <EcoResProduct>
          <LABEL_EcoResProduct_RecId>Record ID</ LABEL_EcoResProduct_RecId>
        - <Detail_Collection>
          - <Detail>
              <LACQueryRecord_DONOT_CHANGE_EcoResProduct>29</ LACQueryRecord_DONOT_CHANGE_EcoResProduct>
              <RecId>22565421216</ RecId>
            - <UnitOfMeasureConversion>
                <LABEL_UnitOfMeasureConversion_Factor>Factor</ LABEL_UnitOfMeasureConversion_Factor>
                <LABEL_UnitOfMeasureConversion_FromUnitOfMeasure>From unit</ LABEL_UnitOfMeasureConversion_FromUnitOfMeasure>
                <LABEL_UnitOfMeasureConversion_Product>Product</ LABEL_UnitOfMeasureConversion_Product>
                <LABEL_UnitOfMeasureConversion_RecId>Record ID</ LABEL_UnitOfMeasureConversion_RecId>
                <LABEL_UnitOfMeasureConversion_ToUnitOfMeasure>To unit</ LABEL_UnitOfMeasureConversion_ToUnitOfMeasure>
              - <Detail_Collection>
                - <Detail>
                    <LACQueryRecord_DONOT_CHANGE_UnitOfMeasureConversion>30</ LACQueryRecord_DONOT_CHANGE_UnitOfMeasureConversion>
                    <Factor>4.0000000000</ Factor>
                    <FromUnitOfMeasure>22565421447</ FromUnitOfMeasure>
                    <Product>22565421216</ Product>
                    <RecId>68719480077</ RecId>
                    <ToUnitOfMeasure>22565421199</ ToUnitOfMeasure>
                  - <UnitOfMeasure_1>
                      <LABEL_UnitOfMeasure_RecId>Record ID</ LABEL_UnitOfMeasure_RecId>
                      <LABEL_UnitOfMeasure_Symbol>Unit</ LABEL_UnitOfMeasure_Symbol>
                    - <Detail_Collection>
                      - <Detail>
                          <LACQueryRecord_DONOT_CHANGE_UnitOfMeasure_1>31</ LACQueryRecord_DONOT_CHANGE_UnitOfMeasure_1>
                          <RecId>22565421447</ RecId>
                          <Symbol>Box</ Symbol>
                        </ Detail>
                      </ Detail_Collection>
                    </ UnitOfMeasure_1>
                  </ Detail>
                </ Detail_Collection>
              </ UnitOfMeasureConversion>
            </ Detail>
          </ Detail_Collection>
        </ EcoResProduct>
        <FIELD_InventTable_ItemId>D0004</ FIELD_InventTable_ItemId>
      </ Detail>
    - <Detail>
        <LACQueryRecord_DONOT_CHANGE_InventTable_1>32</ LACQueryRecord_DONOT_CHANGE_InventTable_1>
        <Product>22565421216</ Product>
        <FIELD_InventTable_RecId>22565421305</ FIELD_InventTable_RecId>
      - <EcoResProduct>
          <LABEL_EcoResProduct_RecId>Record ID</ LABEL_EcoResProduct_RecId>
        - <Detail_Collection>
          - <Detail>
              <LACQueryRecord_DONOT_CHANGE_EcoResProduct>33</ LACQueryRecord_DONOT_CHANGE_EcoResProduct>
              <RecId>22565421216</ RecId>
            - <UnitOfMeasureConversion>
                <LABEL_UnitOfMeasureConversion_Factor>Factor</ LABEL_UnitOfMeasureConversion_Factor>
                <LABEL_UnitOfMeasureConversion_FromUnitOfMeasure>From unit</ LABEL_UnitOfMeasureConversion_FromUnitOfMeasure>
                <LABEL_UnitOfMeasureConversion_Product>Product</ LABEL_UnitOfMeasureConversion_Product>
                <LABEL_UnitOfMeasureConversion_RecId>Record ID</ LABEL_UnitOfMeasureConversion_RecId>
                <LABEL_UnitOfMeasureConversion_ToUnitOfMeasure>To unit</ LABEL_UnitOfMeasureConversion_ToUnitOfMeasure>
              - <Detail_Collection>
                - <Detail>
                    <LACQueryRecord_DONOT_CHANGE_UnitOfMeasureConversion>34</ LACQueryRecord_DONOT_CHANGE_UnitOfMeasureConversion>
                    <Factor>10.0000000000</ Factor>
                    <FromUnitOfMeasure>22565426715</ FromUnitOfMeasure>
                    <Product>22565421216</ Product>
                    <RecId>68719480078</ RecId>
                    <ToUnitOfMeasure>22565421199</ ToUnitOfMeasure>
                  - <UnitOfMeasure_1>
                      <LABEL_UnitOfMeasure_RecId>Record ID</ LABEL_UnitOfMeasure_RecId>
                      <LABEL_UnitOfMeasure_Symbol>Unit</ LABEL_UnitOfMeasure_Symbol>
                    - <Detail_Collection>
                      - <Detail>
                          <LACQueryRecord_DONOT_CHANGE_UnitOfMeasure_1>35</ LACQueryRecord_DONOT_CHANGE_UnitOfMeasure_1>
                          <RecId>22565426715</ RecId>
                          <Symbol>Bag</ Symbol>
                        </ Detail>
                      </ Detail_Collection>
                    </ UnitOfMeasure_1>
                  </ Detail>
                </ Detail_Collection>
              </ UnitOfMeasureConversion>
            </ Detail>
          </ Detail_Collection>
        </ EcoResProduct>
        <FIELD_InventTable_ItemId>D0004</ FIELD_InventTable_ItemId>
      </ Detail>
    </ Detail_Collection>
  </ InventTable_1>
- <UnitOfMeasure>
    <LABEL_UnitOfMeasure_RecId>Record ID</ LABEL_UnitOfMeasure_RecId>
    <LABEL_UnitOfMeasure_Symbol>Unit</ LABEL_UnitOfMeasure_Symbol>
  - <Detail_Collection>
    - <Detail>
        <LACQueryRecord_DONOT_CHANGE_UnitOfMeasure>2</ LACQueryRecord_DONOT_CHANGE_UnitOfMeasure>
        <FIELD_UnitOfMeasure_RecId>22565421199</ FIELD_UnitOfMeasure_RecId>
        <FIELD_UnitOfMeasure_Symbol>ea</ FIELD_UnitOfMeasure_Symbol>
      </ Detail>
```

9. Add Reservation Status to your SalesLine

Add SalesLine

Add InventTransOrigin

SalesLine.InvenTransId == InventTransOrigin.InventTransId



Relation

Relation

Data source: InventTransOrigin

Parent: SalesLine

Layout: Nested

PARAMETERS

Join: OuterJoin

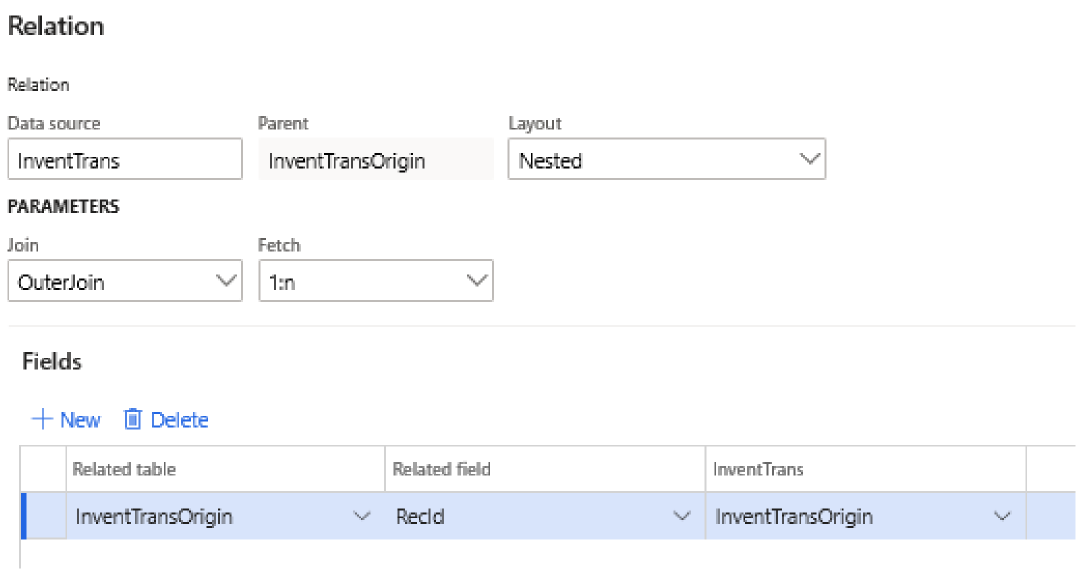
Fetch: 1:n

Fields

Related table	Related field	InventTransOrigin
SalesLine	InventTransId	InventTransId

Add InventTrans

InventTransOrigin.RecId == InventTrans.InventTransOrigin



Relation

Relation

Data source: InventTrans

Parent: InventTransOrigin

Layout: Nested

PARAMETERS

Join: OuterJoin

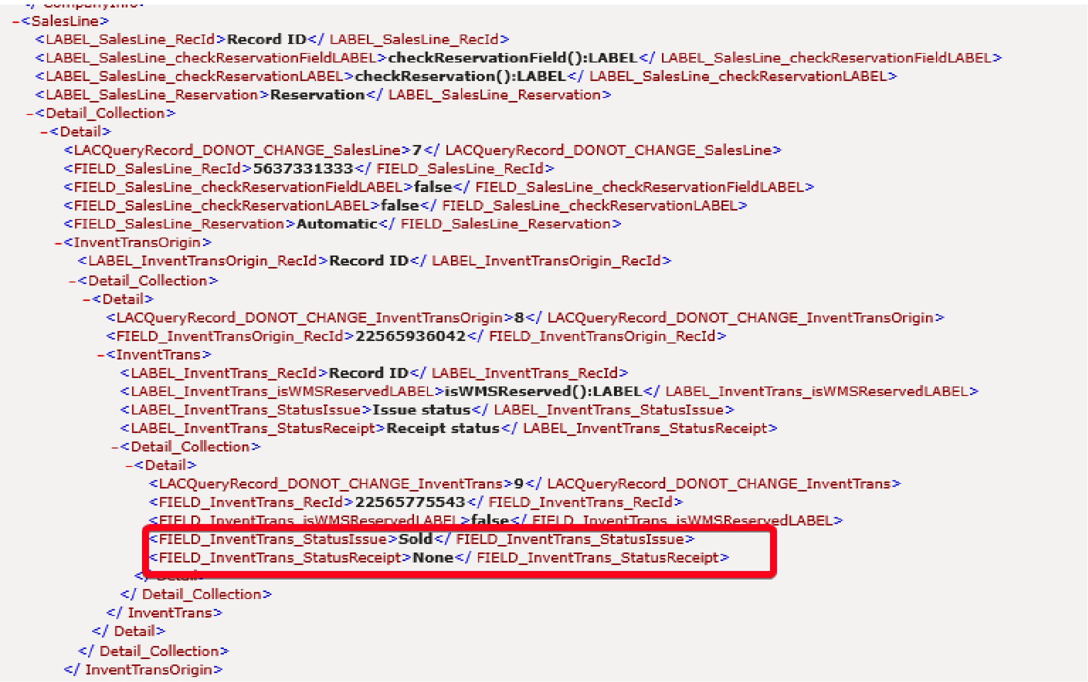
Fetch: 1:n

Fields

Related table	Related field	InventTrans
InventTransOrigin	RecId	InventTransOrigin

Add fields InventTrans.StatusIssue and InventTrans.StatusReceipt for reservation status.

Result:



```
  </ CompanyInfo>
-<SalesLine>
   <LABEL_SalesLine_RecId>Record ID</ LABEL_SalesLine_RecId>
   <LABEL_SalesLine_checkReservationFieldLABEL>checkReservationField():LABEL</ LABEL_SalesLine_checkReservationFieldLABEL>
   <LABEL_SalesLine_checkReservationLABEL>checkReservation():LABEL</ LABEL_SalesLine_checkReservationLABEL>
   <LABEL_SalesLine_Reservation>Reservation</ LABEL_SalesLine_Reservation>
  -<Detail_Collection>
    -<Detail>
       <LACQueryRecord_DONOT_CHANGE_SalesLine>7</ LACQueryRecord_DONOT_CHANGE_SalesLine>
       <FIELD_SalesLine_RecId>5637331333</ FIELD_SalesLine_RecId>
       <FIELD_SalesLine_checkReservationFieldLABEL>false</ FIELD_SalesLine_checkReservationFieldLABEL>
       <FIELD_SalesLine_checkReservationLABEL>false</ FIELD_SalesLine_checkReservationLABEL>
       <FIELD_SalesLine_Reservation>Automatic</ FIELD_SalesLine_Reservation>
      -<InventTransOrigin>
         <LABEL_InventTransOrigin_RecId>Record ID</ LABEL_InventTransOrigin_RecId>
        -<Detail_Collection>
          -<Detail>
             <LACQueryRecord_DONOT_CHANGE_InventTransOrigin>8</ LACQueryRecord_DONOT_CHANGE_InventTransOrigin>
             <FIELD_InventTransOrigin_RecId>22565936042</ FIELD_InventTransOrigin_RecId>
            -<InventTrans>
               <LABEL_InventTrans_RecId>Record ID</ LABEL_InventTrans_RecId>
               <LABEL_InventTrans_isWMSReservedLABEL>isWMSReserved():LABEL</ LABEL_InventTrans_isWMSReservedLABEL>
               <LABEL_InventTrans_StatusIssue>Issue status</ LABEL_InventTrans_StatusIssue>
               <LABEL_InventTrans_StatusReceipt>Receipt status</ LABEL_InventTrans_StatusReceipt>
              -<Detail_Collection>
                -<Detail>
                   <LACQueryRecord_DONOT_CHANGE_InventTrans>9</ LACQueryRecord_DONOT_CHANGE_InventTrans>
                   <FIELD_InventTrans_RecId>22565775543</ FIELD_InventTrans_RecId>
                   <FIELD_InventTrans_isWMSReservedLABEL>false</ FIELD_InventTrans_isWMSReservedLABEL>
                   <FIELD_InventTrans_StatusIssue>Sold</ FIELD_InventTrans_StatusIssue>
                   <FIELD_InventTrans_StatusReceipt>None</ FIELD_InventTrans_StatusReceipt>
                 </ Detail>
               </ Detail_Collection>
             </ InventTrans>
           </ Detail>
         </ Detail_Collection>
       </ InventTransOrigin>
```

Example - How Do I Add Shipment Header to SalesInvoiceReplacementReport?

The SalesInvoiceReplacement report does not come with shipment data. To add it, we need to first add the following table to its data sources through the Query wizard:

- WHSShipmentTable

To do so, follow these steps:

1. Navigate to **Lasernet** > **Common** > **Reports**.

2. In the list, find SalesInvoiceReplacement and select the desired record.

3. Select **Query wizard**.

4. Click **Next** until you get to Tables.

6. Use the Quick Filter to find records. In this case, find 'WHSShipmentTable'.

7. On the right pane, in the query data sources, select **SalesInvoiceHeaderFooterTmp**.

8. Click the right arrow button.

The WHSShipmentTable is added under the SalesInvoiceHeaderFooterTmp table.

9. To set the relation between these two tables, click the **Relation/Layout** button on the

top of the right pane.

10. On the Relation dialog, click **New**.

11. Set the following fields:

- Related table: SalesInvoiceHeaderFooterTmp
- Related field: SalesId
- WHSShipmentTable: OrderNum

12. Click **OK**.

13. Click **Next**.

14. In the Available fields under the table 'WHSShipmentTable', select the desired fields, for example ShipmentID, LoadID, OrderNum, ProNum, shipmentManifestTrackingNumber etc. and click the right arrow to move them to the Selected fields section.

15. Click **Finish** and exit the Query Wizard.

16. Generate a Sales Invoice that has shipment info associated to it.

17. In Lasernet Developer, for the SalesInvoiceReplacement form, grab the XML generated from the step above and update the form design to include newly added fields.

18. Commit and deploy your changes.

19. In Dynamics 365 Finance and Operations, regenerate the Sales Invoice report.

You should be able to see the newly added fields on the report.