Support Bulletin # 110
Common ABM Table Relationships

Table of Contents

Introduction

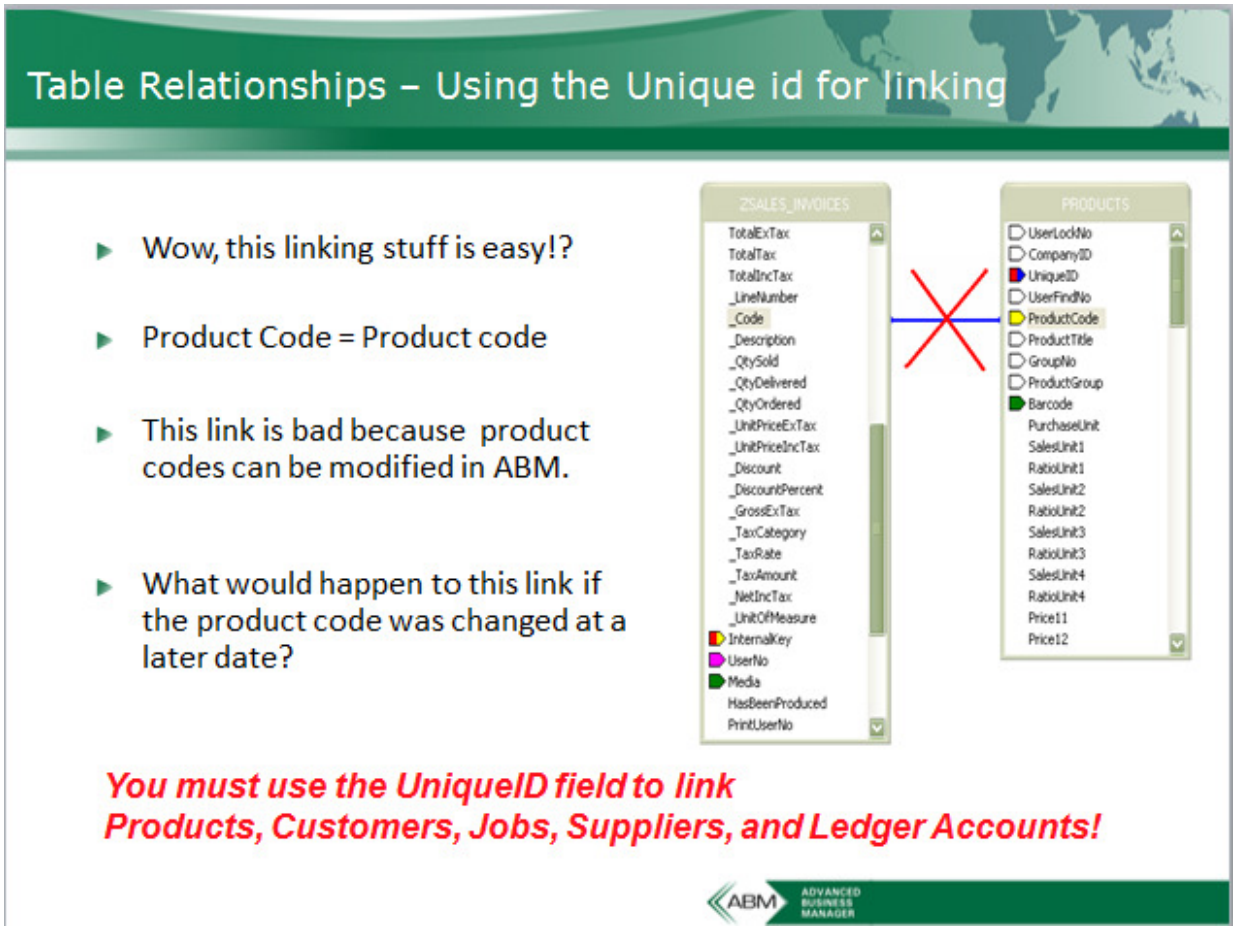
This document outlines some table relationships in ABM that we commonly get enquiries on. The original contented was in a Powerpoint Presentation in an Advanced Training Session.

Using the uniqueid for linking

It's not always obvious how the tables in ABM should relate. Product codes, Customer codes, Supplier codes, Job codes, and Ledger codes seem like a logical way to link, but the problem with this approach is that these codes can be modified in ABM.

In the example below we have the standard Sales Invoice form with the product table added and linked to the ZSales_invoice table. The link is between the ZSales_Invoice._Code field and the Products.ProductCode field. Both of these fields represent the product code so the link appears to be correct. If the product code was changed after this invoice was created then the link may be pointing to the wrong product or even no product at all.

When Customers, Suppliers, Products, Ledger Accounts, and Jobs are created ABM assigns a special id that is unique to each record in each table. This field is called the UNIQUEID and can't be edited. If you use this field then the link will always point to the correct record.



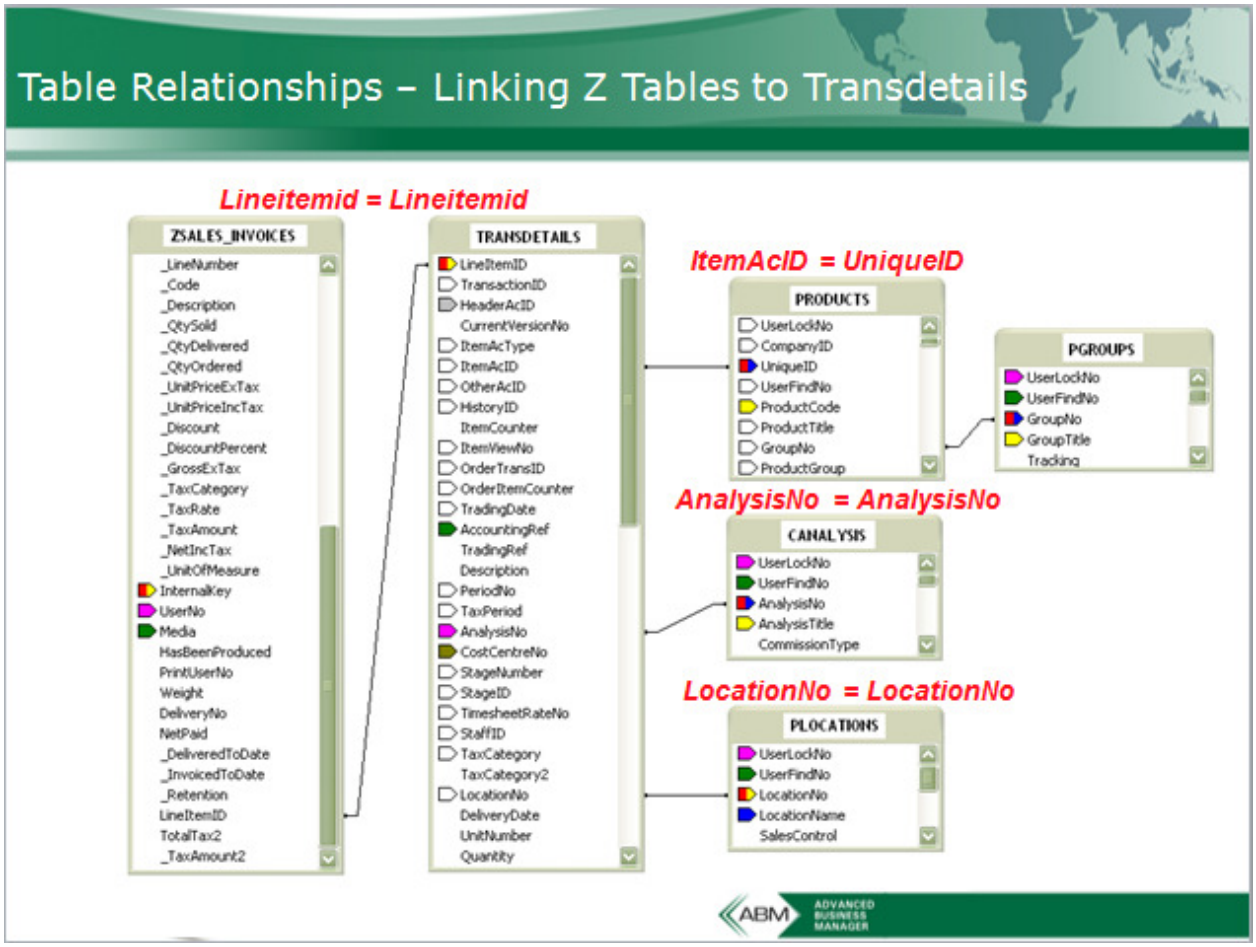
Linking Z Tables to Transdetails

Not only is this the correct relationship (see below) between Z tables and the products table, but also illustrates how to link all z tables to the transdetails table.

Z_Sales_Invoice.LineItemid = Transdetails.Lineitemid

This relationship gives you access to all the information on the transdetails and the ability to link to many other tables in the database.

Transdetails.ItemAcID = Products.UniqueID
Transdetails.AnalysisNo = Canalysis.AnalysisNo
Transdetails.LocationNo = Plocations.LocationNo



Examples of other tables that can be linked from the transdetails table:

Transheaders
Customers
CGroups – Customer Group Title
CAnalysis – Customer Sales Analysis Title
Suppliers
SGroups – Supplier Group Title
Products
Plocations – Product Location Title
PGroups – Product Group Title

Head Office and Branch Relationships

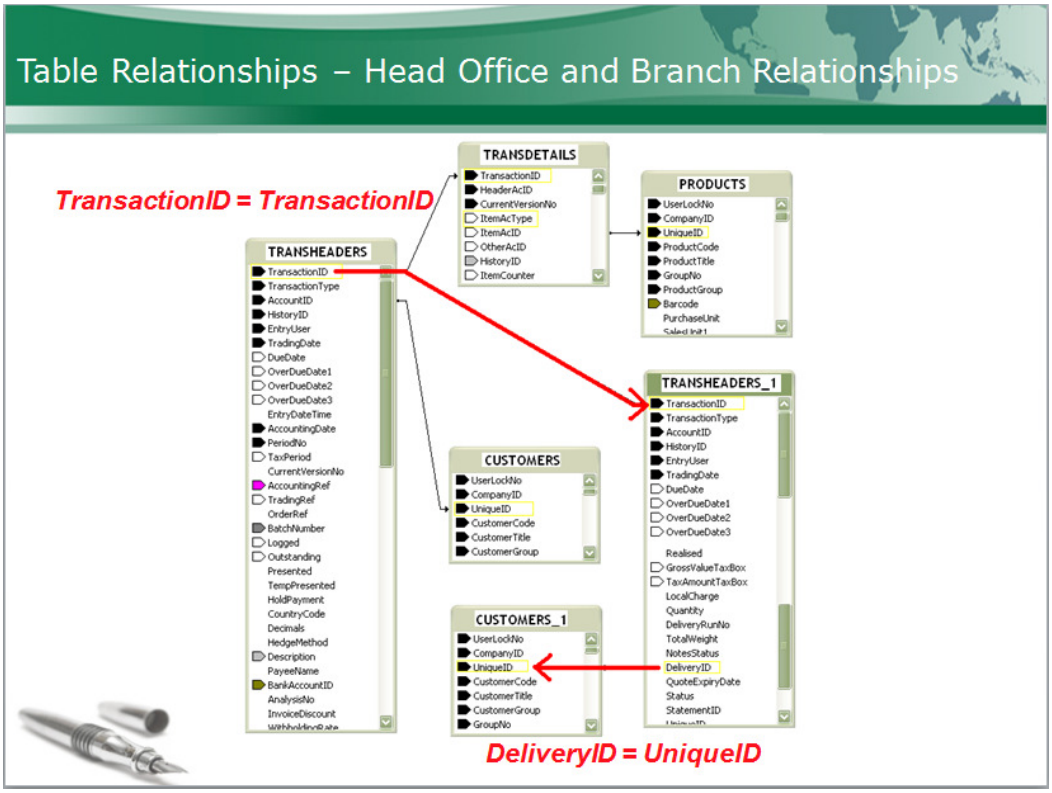
We have had a lot of requests for a report that shows Sales by Branch. This isn't as easy as it sounds because in a Branch and Head office relationship, ABM allocates the

delivery/return transaction to the Branch and the invoice/credit transaction to the Headoffice. If a Headoffice / Branch relationship exists then the invoice/credit transaction contains the UniqueID of the branch in a field called transheader.deliveryid.

Transheaders.TransactionID = Transdetails.TransactionID
Transheaders.TransactionID = Transheaders_1.TransactionID
Tranheaders.AccountID = Customers.UniqueID
Transheader_1.DeliveryID = Customers_1.UniqueID
Transdetails.ItemAcID = Products.UnqueID

These relationships can be used to build a report that will report the sales for the head office buy grouping and totalling sales by the branch associated with each sales transaction. Normally the Transheaders table is linked from the Customers.UniqueID or Supplier.UniqueID to the Transheaders.AccountID field. This works to get the Head Office, but we also need to get the Branch.

The "Trick" to doing this is to add the TRANSHEADERS and CUSTOMERS table to the report twice. This way we can link the Transheaders to the second occurrence (Transheaders_1) by the Transactionid which gives us the same record, but instead of linking the customer from the AccountID of the second Transheader table, we link the customer to the DeliveryId which gives us the Branch account. We can then use the Branch account to group the invoice and credit transactions in our report.



Left Outer Join

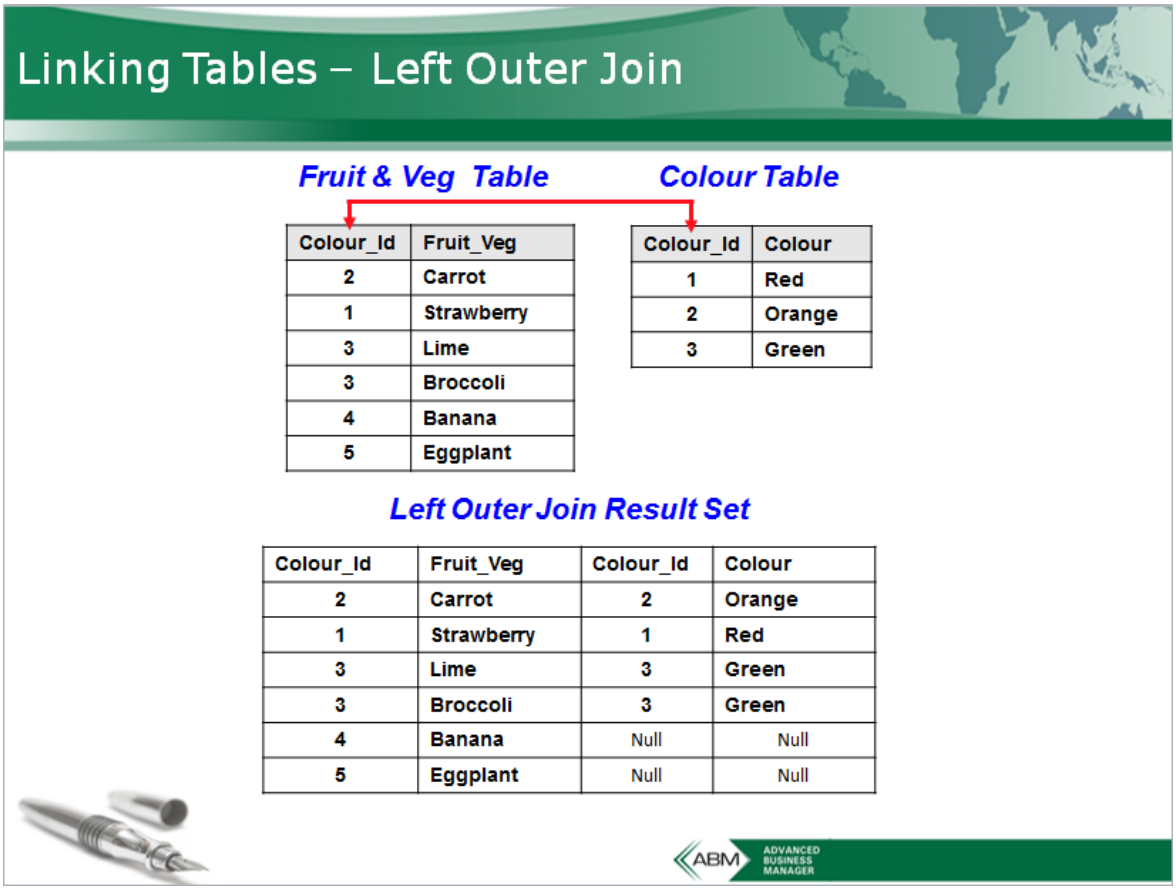
Joins together with links determine what records are returned in the result set of the tables involved. In the example below there are two table that are linked on the Color_Id field.

An Inner join returns all records where the linked fields are equal.

A Left Outer join returns all records from the left table, even if there are no matches from the other table(s).

If we were to create a Inner Join the result would be all the records from the Fruit & Veg table that match a record from the Colour Table. Banana and Eggplant are not in the result set because there are no colours that match Colour_Id 4 and 5.

If we select the same tables using a Left Outer Join we get all of the Fruit & Veg records, even if there are no colour matches.



Colour_Id	Fruit_Veg
2	Carrot
1	Strawberry
3	Lime
3	Broccoli
4	Banana
5	Eggplant

Colour_Id	Colour
1	Red
2	Orange
3	Green

Colour_Id	Fruit_Veg	Colour_Id	Colour
2	Carrot	2	Orange
1	Strawberry	1	Red
3	Lime	3	Green
3	Broccoli	3	Green
4	Banana	Null	Null
5	Eggplant	Null	Null

Tips for Joins

- An Inner Join will return only records that match
- Left Outer Joins return all records in the left table even if there are no matches in the lookup table
- ***The default join in Crystal Reports is an Inner Join!***
- Relationships involving Z tables and Transdetails may require a Left Outer Join so that all records are included in the report. An example of this is the relationship between the Transdetails and the Products table in ABM
- Why? Not all records on transaction lines are Products!