How to use a "Range Where" clause that is based on more than one table in Dexterity in Microsoft Dynamics GP

Partner Only Article

Article ID : 922056
Last Review : 2006-07-27
Revision : 2.0

INTRODUCTION

This article describes how to use a **Range Where** clause that is based on more than one table. You can use this kind of **Range Where** clause in Dexterity in Microsoft Dynamics GP. The **Range Where** clause allows for a **Where** clause to be passed to Microsoft SQL Server. This lets you use the **Range Where** clause to restrict the records that are returned by SQL Server to Dexterity.

MORE INFORMATION

Typically, you use the **Range Where** clause only to work with columns that are in the same table. This limitation exists because the **Range Where** clause does not support joins in SQL Server. Therefore, you must use a subquery when you want to use a **Range Where** clause on more than one table.

When you use a subquery, you can set up a **Range Where** clause that is based on more than one table. You can do this only when the additional tables are fully qualified.

The following sample **Range Where** clause restricts the Sales Order Processing Header table so that Dexterity can only display the transactions for one class of customer.

```
inout table SOP_HDR_WORK;
in 'Class ID' IN_Class_ID;

local text l_where_clause;

pragma(disable warning LiteralStringUsed);

clear l_where_clause;
l_where_clause = l_where_clause + physicalname('Customer Number' of table SOP_HDR_WORK) + CH_SPACE + "in" + CH_SPACE + CH_LEFTPAREN;

{ Create Subquery }
l_where_clause = l_where_clause + "select" + CH_SPACE + physicalname('Customer Number' of table RM_Customer_MSTR) + CH_SPACE;
l_where_clause = l_where_clause + "from" + CH_SPACE + 'Intercompany ID' of globals + CH_PERIOD + SQL_DEFAULT_OWNER  + CH_PERIOD;
l_where_clause = l_where_clause + physicalname(table RM_Customer_MSTR) + CH_SPACE;
l_where_clause = l_where_clause + "where" + CH_SPACE + physicalname('Customer Class' of table RM_Customer_MSTR) + CH_SPACE;
l_where_clause = l_where_clause+ CH_EQUAL + CH_SPACE + SQL_FormatStrings(IN_Class_ID) + CH_SPACE;

l_where_clause = l_where_clause + CH_RIGHTPAREN;

pragma(enable warning LiteralStringUsed);

range table SOP_HDR_WORK where l_where_clause;
```

Additionally, the following sample line adds a **Where** clause to the end of each table command that is sent from Dexterity to SQL Server. The sample line uses a class of **DEFAULT** together with the TWO demonstration database.

```
WHERE CUSTNMBR in (select CUSTNMBR from TWO.dbo.RM00101 where CUSTCLAS = 'DEFAULT' )
```

APPLIES TO

- Dexterity, when used with:
  Microsoft Dynamics GP 9.0
  Microsoft Business Solutions –Great Plains 8.0
  Microsoft Business Solutions –Great Plains 7.5
  Microsoft Great Plains eEnterprise 7.0
  Microsoft Great Plains Dynamics 7.0

Keywords: kbhowto kbinfo kbmbsmigrate kbexpertiseadvanced kbexpertiseinter KB922056