IUG 2018 Polaris SQL Forum

NOTE: Direct SQL access (such as using SSMS) is permitted for turnkey customers (who purchase/maintain their own servers) or for “dedicated hosted” customers (meaning they have their own instance of SQL server shared with no other customers, this feature is part of your contract).

Q: How can you design a report to have multiple values for a parameter?
A: This question was in context of utilizing reports through the desktop client toolbar menu. The parmeter has to be prefixed in the code with GIS_MS_*.

Q: Is there documentation on Polaris database?
A: Yes! Download Polaris database repository from Supportal, which you can find by searching “polaris database respository”. Make sure to download the version appropriate to your installed version of Polaris. You will need to unblock the file to use it (right click file, go to Properties). An extremely useful add-in for SSMS is “SQL Search” available from RedGate.
https://www.red-gate.com/products/sql-development/sql-search/

TIP: An SSMS add-in called SQL Complete Express can be used as an alternative to Intellisense (the built in schema/table completion tool). A paid version is available with more features and Visual Studio compatibility.
https://www.devart.com/dbforge/sql/sqlcomplete/

TIP: In addition to tables, there are also views in the database. A view is a virtual table that retrieves data from one or more tables (or other views), similar to how a SELECT statement returns a data set. SQL Server views abstract the underlying table schema while presenting data in rows and columns like a regular table. For the purposes of the user, it is another “table” you can call on to view information (simplifying your query by calling on one view instead of several tables). Some sample views called out during the session are below.

```sql
--ViewTransactions is helpful in determining which transactionsubtypes are available for querying with a particular TransactionType
SELECT * FROM PolarisTransactions.Polaris.ViewTransactions WITH (NOLOCK) WHERE TransactionTypeID = '6002'
```

```sql
--RWRITER is the prefix for views used by SimpliyReports
SELECT TOP 100 * FROM Polaris.Polaris.RWRITER_BibDerivedDataView WITH (NOLOCK)
```

Q: With version 6.0, there is a new transaction for resetting the due date (modify due date) with the possibility of including a reason or notes with the action. How do you see this reason in the transactions database?

A: All text fields which are stored in the Transactions database are kept in the TransactionStrings table. Reference this table in combination with the TransactionDetails and TransactionHeaders to find out the reason or notes accompany a due date reset. Below is the code to see the reasons for all due date resets since a particular day.

```
SELECT *
FROM PolarisTransactions.Polaris.TransactionHeaders th WITH (NOLOCK)
INNER JOIN PolarisTransactions.Polaris.TransactionDetails td1 WITH (nolock)
ON th.TransactionID = td1.TransactionID AND td1.TransactionSubTypeID IN ('321','322')
LEFT OUTER JOIN PolarisTransactions.Polaris.TransactionDetailStrings tds WITH
(NOLOCK)
ON td1.numValue = tds.TransactionStringID
WHERE th.TransactionDate > '20180401'
```