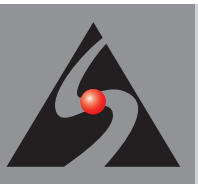


Back to the Future with RelationalCube – ODBC Access to SAP BW

By Amyn Rajan and George Chow

Simba Technologies Inc.™

SAP R/3 revolutionized ERP in the late 20th century and Business Information Warehouse ("BW") looks to extend SAP's leading position in these early days of the 21st century. As SAP is a leading technology company, BW is furnished with modern APIs (OLE DB for OLAP and XML for Analysis) and the usual "legacy" API (RFC BAPI). (You can find a paper that compares and contrasts these three APIs here: http://www.simba.com/docs/ODBO-BAPI-and-XMLA-Its-All-MDX-to-Me.pdf) What is curiously absent from this list is the industry standard ODBC API (or its cousin JDBC).

Simba RelationalCube was conceived to bridge this gap. ODBC is still important because it is still the most common interface to access any data source, and there are a very large number of people in the world who understand SQL. Despite the growth of OLAP and data warehousing technologies, the unassailable truth is that there are more SQL/ODBC client tools, both in the marketplace and in deployment.

At this point, an important detail about BW that is relevant to RelationalCube needs to be pointed out. BW, being built on a standard 3-tier architecture, uses a relational database at its foundation for persisting all data. One common way that people have taken to ODBC access for BW is to side step the SAP middle tier and to reach into the database tier directly. While technologically feasible, there are hosts of reasons this approach is neither sanctioned by SAP, nor suitable for mainstream work. Note that SAP lists the following in its SAP Service Marketplace.

> *Via the OLAP BAPI / OLE DB for OLAP interface the following features are ensured:*
> - *authority checking*
> - *multi-language support: SAP BW master data supports multiple language within a single query / single infocube*
> - *hierarchy handling: SAP BW's OLAP engine interpretes the various types of hierarchies (unique/non-unique, interval-based, time-dep./indep. etc.) and uses type-specific evaluation algorithms*
> - *time-dependent master data, calculated key figures, query structures, multi-currency handling...*

- *non-cumulative key figures: the values stored on the DB need to be interpreted by using SAP BW specific meta data; the pure values persisted in a fact table do not provide any reasonable information for an end-user*
- *rounding, precision: SAP BW's OLAP engine can guarantee precise calculations, which is particularly important in financial data; special handling is necessary as aggregated data is typically further aggregated etc.*
- *SAP-specific formats: currency values are always stored with 2 decimals; thus 100 Yen are stored as 1.00; the necessary shift back to 100 Yen is done in SAP BW's OLAP engine consistent SAP BW metadata handling: synchronization with any possible change within the SAP BW meta data (re-modelling, realignment and other issues)*
- *ODBO interface does not support mass data*
- *Open hub service enables access to InfoCubes, master data and ODS objects (DataStore objects); it supports mass data*
- *ODS-read BAPI is only intended for use by applications, particularly in the context of 'transactional' ODS objects*

Source: SAP AG. (2007) Third Party Data Access Retrieved January 9, 2007, from: http://service.sap.com/~sapidb/011000358700000819442003E/3rdparty_data_access.htm (subscription required)

RelationalCube avoids these issues by working through SAP BW. RelationalCube presents a virtual schema from SAP BW's metadata and uses RFC BAPI to access all BW data. Thus, there are no data quality and/or integrity issues. In fact, the same SAP credentials that you already have are used when connecting with RelationalCube. Therefore, all SAP security roles and authorization are observed.

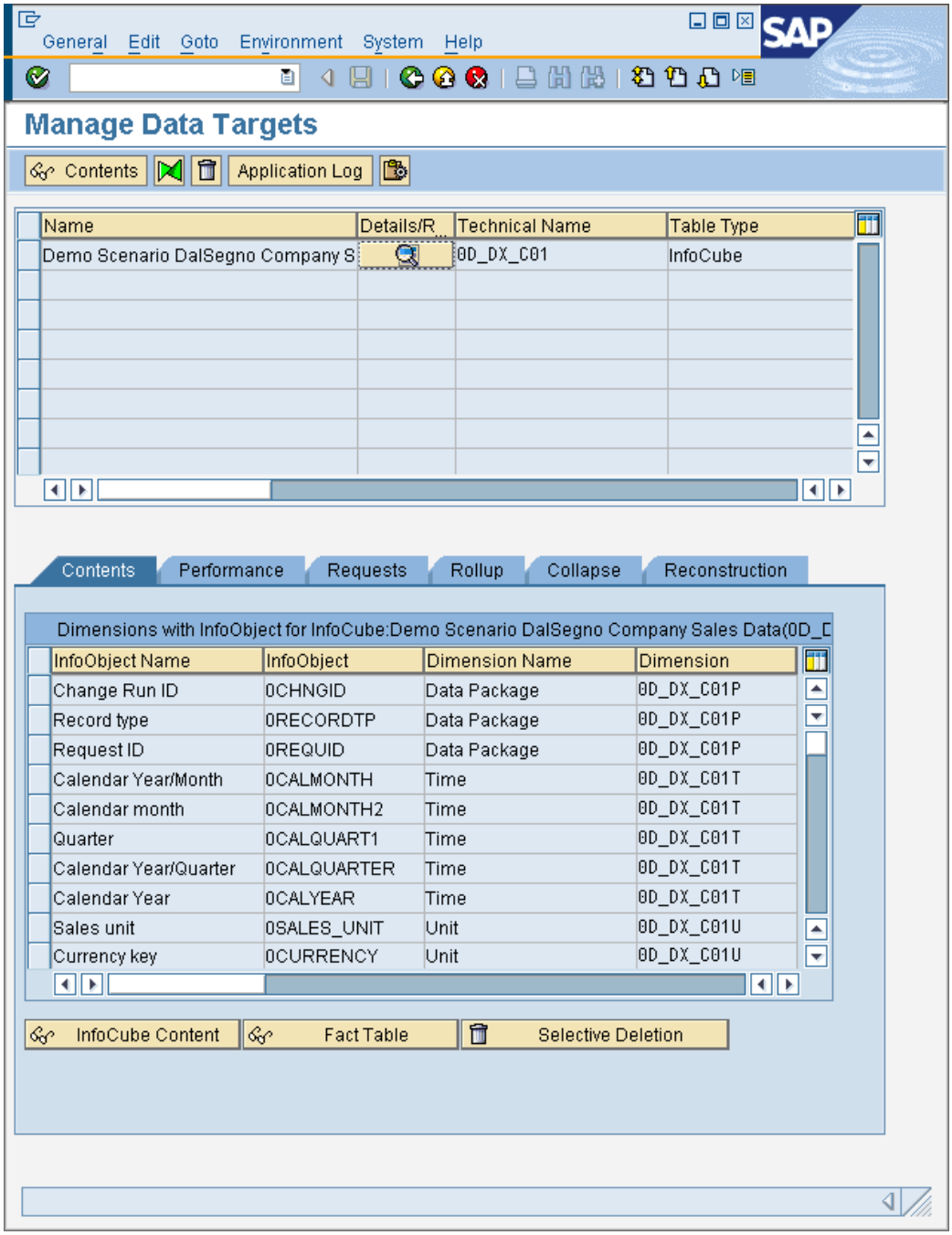
It is important to note that since RelationalCube is using a SAP approved API (RFC BAPI), you are working within the confines of your SAP license.

By now, reasons for using RelationalCube should be clear. Your organization already employs BW as a clearinghouse to consolidate data from across the enterprise. RelationalCube now enables you to connect any SQL tool with this consolidated view of your enterprise.

RelationalCube in Action

RelationalCube works by creating virtual star schemas from your pre-existing BW cubes. Because RelationalCube uses RFC BAPI along with MDX, it can access any InfoCube or QueryCube. If you are looking to use an ODS object, you will need to create a QueryCube from it first.

For example, consider the demo cube 0D_DX_C01:



You can report on it using MDX from any of the APIs above, as follows:

```
select
     {[OD_CUSTOMER].[LEVEL01].Members} on rows,
    {[Measures].Members} on columns
from [$OD_DX_C01]
```

The following is an excerpt of the results:

	Billed Quantity (SAP Demo)	Net Sales (SAP Demo)
Sandstone Ltd	370,018 ST	$ 1,211.87
Adept Services Inc	351,542 ST	$ 1,419.46
Brake Service Co Inc	203,377 ST	$ 594.55
Churko Industries Co Inc	318,562 ST	$ 1,317.11
Thompson, Inc	379,759 ST	$ 1,446.55
Winter Inc	101,485 ST	$ 600.37
Thompson, Inc	337,969 ST	$ 1,128.80
	334,297 ST	$ 1,210.89
	274,104 ST	$ 1,015.02
	238,511 ST	$ 1,121.71
Bear Express Co Inc	102,986 ST	$ 383.04
Genesis Baptist Church	438,871 ST	$ 1,680.76
Dawson Agency Inc	506,177 ST	$ 1,618.57
Acadia Transfer Inc	431,210 ST	$ 1,738.39
Evans Hotel Inc	475,629 ST	$ 1,637.41
Rainbow Ltd	195,404 ST	$ 907.04
Viking Automotive Inc	422,114 ST	$ 1,506.91
Fosters Ltd	336,583 ST	$ 1,159.93
Pepper Mill Clothes Group Inc	454,843 ST	$ 1,534.21
Corkill Cards and Gifts, Inc	435,660 ST	$ 1,463.51
Windsor Agency Ltd	320,354 ST	$ 1,444.85
Abstract Car Inc	427,403 ST	$ 1,409.58
Ramon Scissor Hands	274,095 ST	$ 1,170.52

However, with RelationalCube, you can now report on it from SQL, like so:

```
SELECT
     TOD_CUSTOMER_LEVEL01.CMEMBER_UNIQUE_NAME,
     TOD_CUSTOMER_LEVEL01.CMEMBER_CAPTION,
     Sum(TFactTable.CMeasures_OD_INV_QTY),
     Sum(TFactTable.CMeasures_OD_NETSALES)
FROM
     TOD_CUSTOMER_LEVEL01,
     TFactTable
WHERE
     TOD_CUSTOMER_LEVEL01.CMEMBER_UNIQUE_NAME = TFactTable.COD_CUSTOMER_LEVEL01_
CMEMBER_UNIQUE_NAME
GROUP BY
TFactTable.COD_CUSTOMER_LEVEL01_CMEMBER_UNIQUE_NAME,
    TOD_CUSTOMER_LEVEL01.CMEMBER_CAPTION
```

C0D_CUSTOMER_LEVEL01_CM	CMEMBER_CAPTION	Sum(CMeasures_0D_INV_	Sum(CMeasures_0D_
[0D_CUSTOMER].[DS1000]	Churko Industries Co Inc	318562	1317113
[0D_CUSTOMER].[DS1001]	Thompson, Inc	379759	1446547
[0D_CUSTOMER].[DS1002]	Winter Inc	101485	600372
[0D_CUSTOMER].[DS1003]	Thompson, Inc	337969	1128803
[0D_CUSTOMER].[DS1004]	Thompson, Inc	334297	1210885
[0D_CUSTOMER].[DS1005]	Thompson, Inc	274104	1015022
[0D_CUSTOMER].[DS1006]	Thompson, Inc	238511	1121710
[0D_CUSTOMER].[DS1007]	Bear Express Co Inc	102986	383044
[0D_CUSTOMER].[DS1008]	Genesis Baptist Church	438871	1680758
[0D_CUSTOMER].[DS1009]	Dawson Agency Inc	506177	1618570
[0D_CUSTOMER].[DS100]	Brake Service Co Inc	203377	594549
[0D_CUSTOMER].[DS1010]	Evans Hotel Inc	475629	1637414
[0D_CUSTOMER].[DS1011]	Rainbow Ltd	195404	907042
[0D_CUSTOMER].[DS1012]	Viking Automotive Inc	422114	1506905
[0D_CUSTOMER].[DS1013]	Fosters Ltd	336583	1159927
[0D_CUSTOMER].[DS1014]	Pepper Mill Clothes Group Inc	454843	1534206
[0D_CUSTOMER].[DS1015]	Corkill Cards and Gifts, Inc	435660	1463510
[0D_CUSTOMER].[DS1016]	Windsor Agency Ltd	320354	1444848
[0D_CUSTOMER].[DS1017]	Abstract Car Inc	427403	1409582
[0D_CUSTOMER].[DS1018]	Ramon Scissor Hands	274095	1170523
[0D_CUSTOMER].[DS1019]	Bishop Stores Group Inc	298577	1054897
[0D_CUSTOMER].[DS101]	Acadia Transfer Inc	431210	1738385
[0D_CUSTOMER].[DS1020]	Friends Memories	377915	1344091
[0D_CUSTOMER].[DS1021]	Williams B'S Mexican Food and F	338494	1236517
[0D_CUSTOMER].[DS1022]	Pencom Ltd	322500	1432821
[0D_CUSTOMER].[DS1023]	Jamieson Agency Co Inc	326365	1092252
[0D_CUSTOMER].[DS1024]	Carlington Styling Co Inc	262048	1199136

Note that the rows from "Churko Industries Co Inc" to "Abstract Car Inc" are identical between the two excerpts. The reason that the rows do not line up exactly is due to MDX and SQL's differing row ordering semantics. But, it should be clear that the two queries are equivalent.

Understandably, the SQL is notably more verbose than the MDX. However, the breadth of tools available for SQL more than compensates for this. As well, in either case, you are likely to be using a GUI tool to generate the required command and not writing one by hand.

RelationalCube in Comparison

Having shown the functional equivalence between RelationalCube and the other MDX-based API, the next natural topic is the performance characteristics. RelationalCube features Simba's SimbaEngine SQL engine and utilizes pushdown optimizations for table joins and filter clauses whenever possible. These optimizations keep the processing on the BW server side and minimize the amount of processing on the client end (where the RelationalCube driver is installed). Here is a summary of a few scenarios:

Query	Cells	OLE DB for OLAP/MDX	Rows	RelationalCube/SQL
1	10178	10 sec	5089	12 sec
2	22276	15 sec	11139	83 sec
3	81424	26 sec	40712	60 sec
4	89104	31 sec	44552	290 sec

The numbers above do not demonstrate a simple trend. For brevity, we have not included additional details about the queries and/or cubes. However, aside from the volume of data, the cube's structure also has a major effect on the performance.

RelationalCube in Details

Those of you who are familiar with SAP BW will no doubt be aware of the concept of SID (surrogate ID). SAP uses SID to implement advance features, such as time-dependent and multi-lingual master data. Note that SIDs are never exposed externally thru any SAP API because they are part of the internal details of BW. Since RelationalCube is built with SAP's RFC BAPI API, this means that RelationalCube also behaves like any other well-written SAP BW application. In particular:

1. RelationalCube observes all SAP security roles and authorization. The RelationalCube user only has access to cubes to which they have been granted. Of course, SAP also provides access control for the contents of the cubes and these will also be enforced.

2. RelationalCube supports all language available in a given SAP BW instance. The SQL metadata (table names, columns etc) provided by RelationalCube is invariant of your login language. The data – the contents of the tables – will depend on your login language.

3. RelationalCube uses MDX and shares the features and restrictions that MDX affords. MDX only allows access to InfoCubes and QueryCubes. SAP RemoteCube, RemoteCube and MultiProviders are supported since they're not distinguished from regular InfoCubes/QueryCubes.

4. RelationalCube currently does not support all variable types. Only cubes that do not require a variable value to be specified are supported by RelationalCube.

Conclusion

In this article, we've highlighted the key features of RelationalCube. We've seen that it is a viable option for retrieving data from SAP BW via SQL. More importantly, all of BW's strengths in data security, flexibility and integrity are preserved because RelationalCube is built with a supported SAP API.

About Simba Technologies

Simba Technologies Incorporated (http://www.simba.com) builds development tools that make it easier to connect disparate data analysis products to each other via standards like ODBC, JDBC, OLE DB, OLE DB for OLAP (ODBO), and XML for Analysis (XMLA). Independent software vendors that want to extend their proprietary architectures to include advanced analysis capabilities look to Simba for strategic data connectivity solutions. Customers use Simba to leverage and extend their proprietary data through high performance, robust, fully customized, standards-based data access solutions that bring out the strengths of their optimized data stores. Through standards-based tools, Simba solves complex connectivity challenges, enabling customers to focus on their core businesses.

About the Authors

Amyn Rajan is President and CEO of Simba Technologies. Amyn has over 17 years experience in custom software development, and is responsible for driving Simba's success as a leader in data connectivity solutions.

George Chow is a Development Manager with Simba Technologies. George has over 16 years experience in software and technology, and is a specialist in SAP BW, database access and data connectivity solutions.

Prepared for SAP Developer Network

Simba Technologies Incorporated
938 West 8th Avenue
Vancouver, BC Canada
V5Z 1E5

Tel. +1.604.633.0008
Fax. +1.604.633.0004

Email. solutions (at) simba.com
www.simba.com