



Oracle Essbase XOLAP and Teradata

## Table of Contents

- 2 Scope
- 2 Overview
- 3 XOLAP Functional Summary
  - 4 XOLAP in a Partitioned Environment
  - 4 There are Restrictions with XOLAP
  - 5 XOLAP Cubes are Built with Oracle Essbase Studio
  - 5 XOLAP Application Development
- 5 XOLAP Performance Testing
  - 6 Setup
  - 6 Running the Tests
  - 6 Timing Results
  - 7 SQL Produced
  - 9 Test Summary
- 9 Conclusions
  - 9 For More Information

## Scope

This paper provides an overview of Oracle Essbase XOLAP and Oracle Essbase Studio and how they integrate with Teradata. This paper is targeted to BI administrators, system integrators, and database administrators. Readers are expected to have a basic understanding of the features of the Teradata® Database and Oracle Essbase.

## Overview

Oracle Essbase was first launched in 1992. Since then, many other OLAP tools have come and gone. Oracle Essbase has thrived and become a leading OLAP tool by continuously adding new features, adapting to new environments, and taking advantage of new technologies. Oracle Essbase XOLAP and Oracle Essbase Studio are examples of the latest improvements.

In the past, creating a multi-dimensional OLAP (MOLAP) cube was the only option for delivering an analytic solution that consistently provided fast OLAP results to users. Today, by taking advantage of Teradata Aggregate Join Indexes (AJIs), the Teradata Database and Oracle Essbase XOLAP (eXtended OnLine Analytic Processing) a relational OLAP (ROLAP) solution can be provided that satisfies the needs of OLAP users.

XOLAP is a variation of OLAP in business intelligence. Specifically, XOLAP is an Oracle Essbase multi-dimensional database available with the latest Oracle Essbase versions (e.g. 11.1.2) that stores only the outline metadata and retrieves data from Teradata at query time. XOLAP thus integrates a source relational database (Teradata) real time with an Oracle Essbase database, leveraging the scalability of Teradata with the more sophisticated analytic capabilities of a multi-dimensional database. Oracle Essbase still performs the calculations on the retrieved data. Oracle Essbase still provides the speed of dimensional navigation, member selection, and data/metadata security enforcement. The Teradata Database provides the horsepower that a traditional Oracle Essbase server may be lacking.

XOLAP thus integrates a source Teradata Database with an Oracle Essbase analytical engine, providing a ROLAP solution. This leverages the scalability of Teradata with the more sophisticated analytic capabilities of Oracle Essbase without moving data out of the database. XOLAP provides another option for your OLAP solutions.

	BSO	ASO	XOLAP
Storage Space	Large	Smaller	<b>Smallest</b>
Number of Dimensions	Limited	<b>Unlimited</b>	<b>Unlimited</b>
Location of Metadata	Essbase	Essbase	Essbase
Location of Fact Data	Essbase	Essbase	Database
Data Load Speed	Slowest	Faster	<b>Fastest</b>
History	Limited	<b>Unlimited</b>	<b>Unlimited</b>
Detail	Limited	<b>Unlimited</b>	<b>Unlimited</b>
Hierarchies	<b>Balanced, Ragged</b>	<b>Balanced, Ragged</b>	Balanced
Write-back	<b>Supported</b>	Limited	Not supported
Pre-calculation	<b>Supported</b>	Limited	Can be done in Database or in Essbase
Pre-calculation Method	Calc Scripts	MDX	SQL and/or MDX
Data Freshness	Load Required	Load Required	<b>Same as Warehouse</b>

Figure 1. Oracle Essbase Storage Options.

Business needs determine whether MOLAP or XOLAP is best suited to your environment.

XOLAP would typically be considered for any of these situations:

- Traditional cube is too large to load (more) data:
  - Cannot analyze/report against cube in timely manner.
  - Cannot keep history.
  - Cannot add detail.
  - Cannot bring in additional dimensions.
- Fact data is dynamic.
  - Users can't wait for a cube rebuild to get access to the latest data.
- Eliminate duplicate data store.
  - Reduces hardware costs of Essbase Server.

Figure 1 summarizes the different Oracle Essbase storage options. The text in orange highlights which option is optimal for each criteria.

## XOLAP Functional Summary

XOLAP cubes are defined in an aggregate storage format (ASO). This format is usually used when the Oracle Essbase outline contains a large number of sparse dimensions and one or no dense dimensions. In XOLAP, the metadata is located in the Oracle Essbase database and the underlying data remains in the source relational database. This model allows the aggregation to be done in the Teradata Database and additional calculations to be done in Oracle Essbase. At XOLAP cube creation time the dimensional values are loaded into the Oracle Essbase outline. The fact data is left in the Teradata Database.

When an end user queries an XOLAP cube the Oracle Essbase engine generates SQL to retrieve the data stored in the Teradata Database. With XOLAP, the SQL that the engine generates is Teradata friendly. Teradata has worked with Oracle to enable Oracle Essbase to produce SQL that the Teradata Database can execute efficiently.

An XOLAP cube can be made aggregate aware. Multiple tables can be defined as sources for a cube. The Oracle Essbase engine will generate SQL to retrieve the smallest applicable source or aggregate table. The Teradata Database is itself aggregate aware. If an AJI covers the query generated by Oracle Essbase, then the Teradata Database will automatically use the AJI. Oracle Essbase does not need to be made aware of AJIs; Teradata will choose the best source. This way the Teradata Database will perform the query aggregation while the other calculations are performed by Oracle Essbase. The data security is still defined in Oracle Essbase and is honored by XOLAP. There is further transparency of access for end users; any tool that works against Oracle Essbase today will continue to work with XOLAP.

### XOLAP in a Partitioned Environment

Each Oracle Essbase storage option (ASO, Block BSO, and XOLAP) has strengths that can be used to meet user requirements. Partitioning allows these different storage options to be used together. So for example, the ability to use write-back of forecasted data can be combined with low-level actuals coming from Teradata.

From the end-user's perspective, the use of a partitioned cube structure is transparent. Having upper levels of data stored in a cube and having Teradata Database supply the data on demand for lower levels can be achieved via a partitioned cube linking a physical MOLAP cube and XOLAP cube.

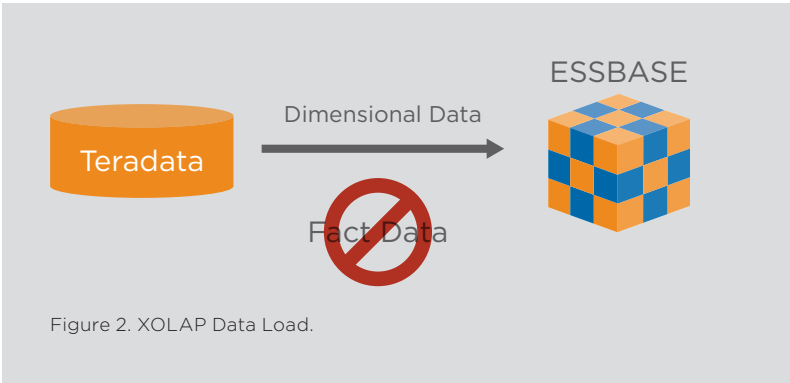


Figure 2. XOLAP Data Load.

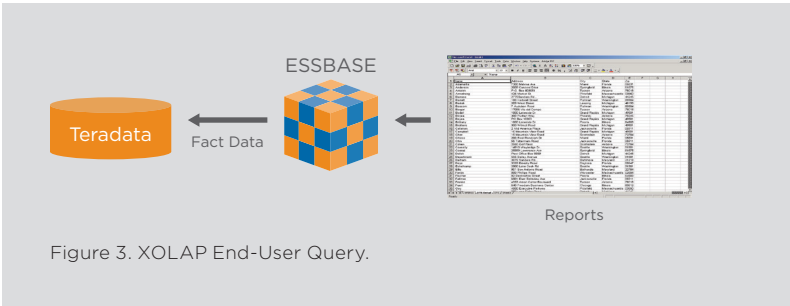


Figure 3. XOLAP End-User Query.

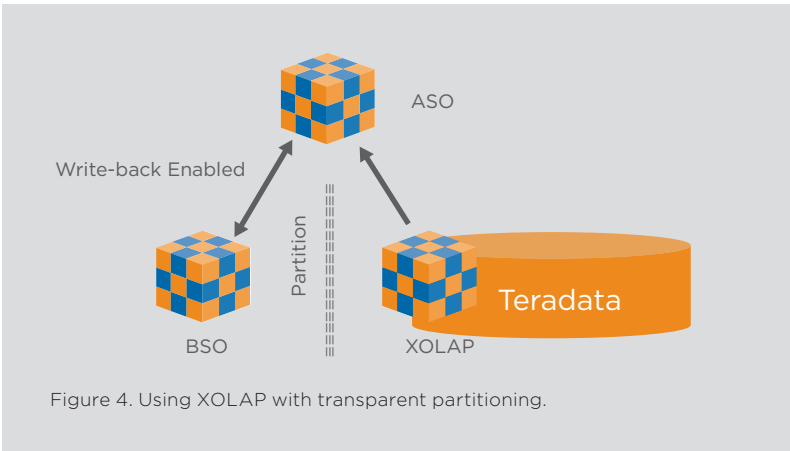


Figure 4. Using XOLAP with transparent partitioning.

### There are Restrictions with XOLAP

As with anything there are restrictions. Although the Outline Editor in Administration Services Console does not prevent you from modifying an XOLAP outline, it is not allowed. XOLAP operations do not automatically incorporate changes in the structures and the contents of the dimension tables after an outline is created. To modify an outline, you must modify the underlying Oracle Essbase model and redeploy to a new Oracle Essbase database.

Other restrictions with XOLAP:

- Flat files
- Ragged hierarchies
- Alternate hierarchies
- Recursive hierarchies
- Calendar hierarchies
- Security Filters
- Typed measures
- User defined members at the leaf level
- Multiple relational data sources

**There are workarounds for all of these restrictions.**

The workarounds require extra implementation time but should not be considered “show-stoppers” for an XOLAP model.

A recent version of Oracle Essbase (11.1.2.2 or newer) is required for XOLAP. Oracle made substantial changes to the way that SQL is generated. Older versions of Oracle Essbase will not perform well for an XOLAP deployment.

### XOLAP Cubes are Built with Oracle Essbase Studio

Oracle Essbase Studio, initially available with Oracle Essbase 11.1.1, is the next-generation environment for the development, deployment, and maintenance of enterprise scale Oracle Essbase analytic applications. Oracle Essbase Studio replaces Oracle Essbase Integration Server and portions of Oracle Essbase Administration Server. Studio is a new unified environment that helps significantly reduce the effort in building and maintaining enterprise Oracle Essbase.

Oracle Essbase Studio simplifies cube construction by delivering a single environment for data modeling, cube design, and analytic application construction tasks. By consolidating cube construction activities into one interface Oracle Essbase Studio provides a consistent platform for building outlines and loading data. A common metadata repository/catalog, captures all metadata related to all Oracle Essbase applications built in the enterprise and allows the reuse of metadata at the lowest level of granularity. Oracle Essbase Studio also supports lineage tracking through a rich graphical view of the metadata relationships, allowing users to follow application lineages

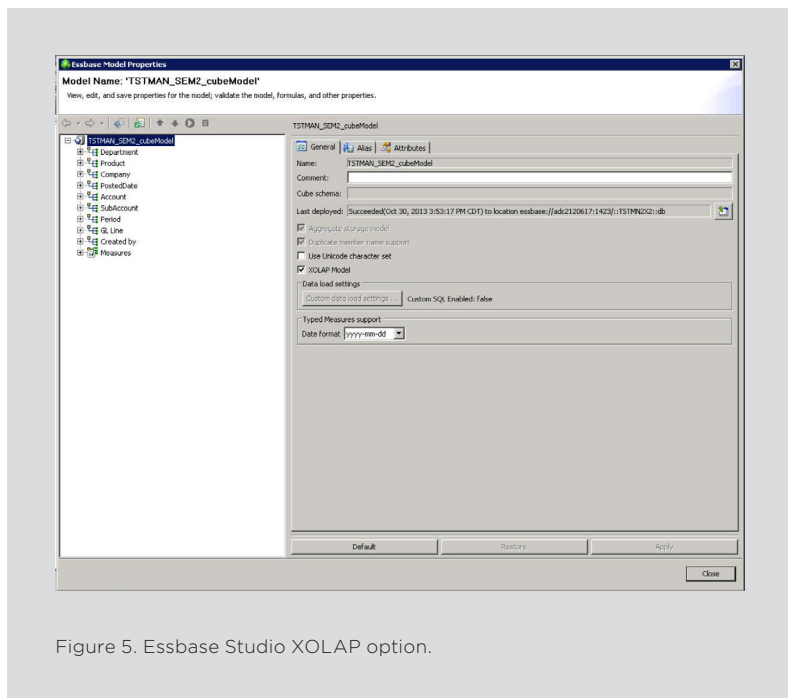


Figure 5. Essbase Studio XOLAP option.

to their metadata components and through to the data sources from which they were sourced.

Oracle Essbase Studio supports the modeling of Oracle Essbase cubes from relational databases, OBIEE, flat files, and the Dimension Server Library in EPM Architect. Oracle Essbase Studio supports several drill-through options: relational databases, OBIEE, URLs, custom SQL, and Java methods. In addition, Oracle Essbase supports drill-back to Oracle Hyperion Financial Data Quality Management.

### XOLAP Application Development

The workflow in an XOLAP environment is much like that of a non-XOLAP environment:

- The model is designed without concern for the type of the underlying database.
- The model is designated as XOLAP-enabled. There is a check box to determine if it is XOLAP or not.
- The cube is deployed; however, no fact data is loaded.
- The Oracle Essbase database is queried, using Smart View, or another reporting tool that can access an Oracle Essbase database.
- Oracle Essbase dynamically generates the required SQL to retrieve the data from the source relational database.

## 3NF Schema

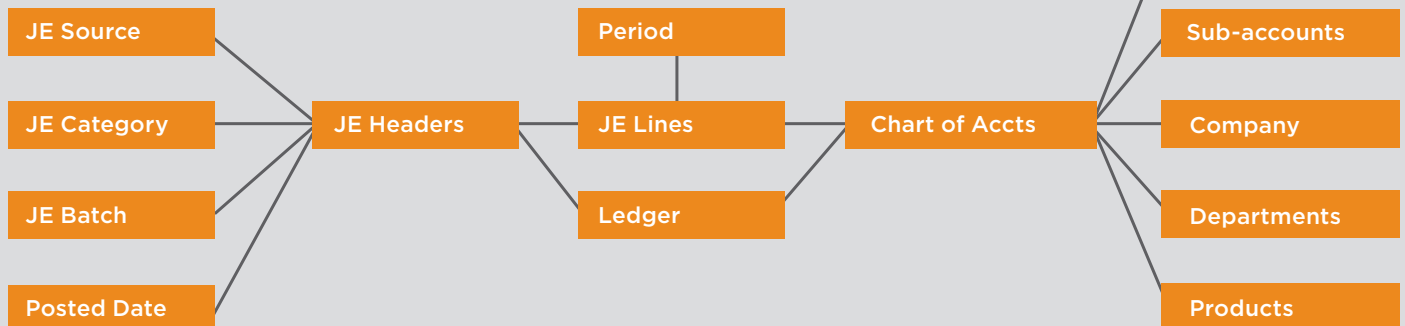


Figure 6. Third Normal Form Schema.

## XOLAP Performance Testing

Teradata and Oracle participated in joint performance testing of the older (initial) version of XOLAP against the latest functionality (11.1.2.3). The testing was done against Teradata Database 14. The goal was to measure the performance improvements of recent XOLAP enhancements. The secondary goal was to show performance improvements of a “BI-Ready” data model.

### Setup

Test data was obtained from sample data sourced from an Oracle Enterprise Resource Planning Application. The data was in third normal form (3NF) tables with views on top to simulate a star schema; one fact table with 20 dimensions. Diagrams of the 3NF schema and star schema are included in figures 6 and 7. The tables had decent primary indexes and statistics were up-to-date. The hierarchies were not ragged but they did have duplicate members; level numbers were concatenated to eliminate the duplicates. The fact to dimension joins are all inner joins.

The Teradata Database 14 was running on a 2-node Teradata 2690 system.

Oracle Essbase 11.1.2.3 was installed on a windows server. Metadata was built in Oracle Essbase Studio based on the star schema. Only a subset of the dimensions was actually built into the metadata.

The following nqconfig.ini settings were set:

- SSBULKGRIDPROCESSING ON
- SSOPTIMIZEDGRIDPROCESSING TRUE

### Running the Tests

Smart View was used to query the XOLAP cube.

The test scenario was to start with the default query. Then:

- Drill down on Department to G2.
- Drill down on Department to G3.
- Drill on Account to G2.
- Drill on Account to G3.
- Drill on Account to G4.
- Drill on Account to G5.
- Create an Asymmetrical query by deleting one of the rows in the spreadsheet and then refreshing the data.

### Timing Results

The results of the tests were measured by using a stopwatch for “clock time” and using DBQL for the database statistics. The timings show that data was coming back quickly. There were 8 mouse clicks in the test to perform the drill operations. Without AJs, results were coming back in 1.4 seconds per click. With AJs, the results were returned even faster at about 1 second per click. This is essentially half of the time of the previous version of XOLAP.

## Star Schema

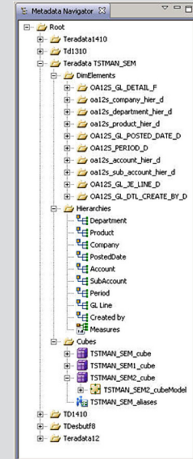
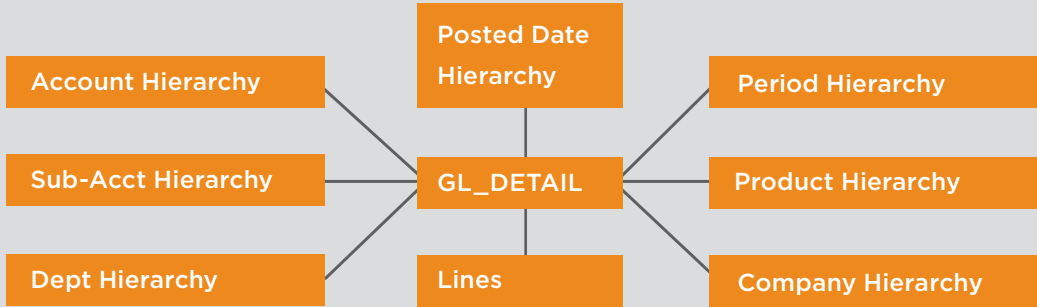


Figure 7. Star Schema.

## SQL Produced

The SQL that Oracle Essbase produced in the 11.1.2.2+ version is substantially better than previous versions. The number of SQL queries are dramatically reduced (3x), and the structure of the SQL is much simpler. The number of queries was reduced by intelligently generating queries at leaf levels of aggregation and the further summarizing of the results on the Oracle Essbase server. The multitude

of queries generated in an asymmetric situation has been resolved. The structure of the SQL was improved by applying filters in a more efficient manner. The older versions listed all of the leaf level members in the where clause of the SQL statement. The newer version removed the leaf level constraints and filtered based on the level in the hierarchy. See figure 10 for a sample of new SQL versus the older style of SQL.

A3 Parent Total									
	A	B	C	D	E	F	G	H	I
1			Company	PostedDate	SubAccount	Period	GL Line	Created by	Measures
2			Product						
3	Parent Total	100 Resources	95967774.59						
4	Parent Total	400 Sales	1755602689						
5	Parent Total	500 Manufacturing	2541916723						
6	Parent Total	700 Finance	50989653.9						
7	Parent Total	800 HR and Administration	190309923.3						
8	Parent Total	PT Parent Total	#Missing						
9	Parent Total	PT Parent Total	4634786764						
10	Parent Total	Department	4634786764						
11	Account	100 Resources	95967774.59						
12	Account	400 Sales	1755602689						
13	Account	500 Manufacturing	2541916723						
14	Account	700 Finance	50989653.9						
15	Account	800 HR and Administration	190309923.3						
16	Account	PT Parent Total	#Missing						
17	Account	PT Parent Total	4634786764						
18	Account	Department	4634786764						

Figure 8. Smart View output.

Table Name	Row Count	View Name	Row Count
OA123_COA	93,390	OA12S_ACCOUNT_D	500
OA123_COA_VALUE_DESC	48,681	OA12S_ACCOUNT_RLUP_D	396
OA123_CURRENCY	255	OA12S_COMPANY_D	18
OA123_DB_CALENDAR	10,592	OA12S_COMPANY_RLUP_D	16
OA123_GL_BALANCES	6,480,239	OA12S_CURRENCY_D	255
OA123_GL_JE_BATCH	72,800	OA12S_DEPARTMENT_D	96
OA123_GL_JE_HEADER	99,686	OA12S_DEPARTMENT_RLUP_D	84
OA123_GL_JE_LINE	2,449,353	OA12S_GL_ACCOUNT_TYPE_D	7
OA123_GL_OA_JE_CATEGORY	227	OA12S_GL_BAT_CATEGORY_D	227
OA123_GL_OA_JE_SOURCES	91	OA12S_GL_BUDGET_D	176
OA123_GL_PERIODS	5,050	OA12S_GL_DETAIL_F	402,765
OA123_LEDGERS	86	OA12S_GL_DTL_CREATE_BY_D	2,544
OA123_LOOKUP_VALUES	174,048	OA12S_GL_JE_LINE_D	402,765
OA123_OA_FLEX_ROLLUP	9,510	OA12S_GL_JE_SOURCES_D	91
OA123_USER_LIST	2,544	OA12S_GL_POSTED_DATE_D	10,592
OA12S_ACCOUNT_HIER_D	1,130	OA12S_LEDGER_D	86
OA12S_COMPANY_HIER_D	20	OA12S_PERIOD_D	236
OA12S_DEPARTMEN_HIER_D	496	OA12S_PRODUCT_D	33
OA12S_PRODUCT_HIER_D	73	OA12S_PRODUCT_RLUP_D	30
OA12S_SUB_ACCOU_HIER_D	257	OA12S_SUB_ACCOUNT_D	158
		OA12S_SUB_ACCOUN_RLUP_D	157
		OA12S_TRX_CURRENCY_D	255

Tables, Views, and Row Counts.



Test Description	Number of SQL Statements	Number of Steps in Explain Plan	Total Clock Time	Total I/O Count	AMP CPU Time
Previous version of XOLAP	30	930	16.7	672227	79.084
Current version of XOLAP	11	341	11.1	286263	45.496
Current version of XOLAP with AJIs in place	11	231	8.3	188394	43.65

Figure 9. Results Table

## Test Summary

With Oracle Essbase 11.1.2.2+, Oracle has substantially improved the XOLAP engine. Not only are fewer queries produced by the engine (3x reduction in the testing), but the structure of the queries is better. This results in quicker response times for the end-user (66 percent faster).

The latest version of the Oracle Essbase XOLAP engine should be a good fit for customers that need a ROLAP application. During the testing the engine has proven itself to be quick and efficient. The inefficiencies in versions prior to 11.1.2.2 have been resolved.

Tuning the Teradata Database at a customer's site for an XOLAP application should be straightforward due to the good and predictable SQL being sent. "Base" and "broad" AJIs can be created to efficiently handle the Oracle Essbase queries.

## Conclusions

Building, updating, maintaining, and enhancing MOLAP cubes can be difficult. Things such as the time it takes to build a cube, data latency, limited dimensionality, limited history, and limited detail can be major issues associated with MOLAP cubes. Teradata Database used with Oracle Essbase XOLAP is an alternative to alleviate these issues. Defining your cubes as XOLAP applications will enable a scalable solution that will result in analytics that are larger in dimensionality, detail, and history built in a fraction of the time it takes to build a MOLAP cube.

## For More Information

Oracle Essbase  
[www.oracle.com/technetwork/middleware/essbase/overview/index.html](http://www.oracle.com/technetwork/middleware/essbase/overview/index.html)

Teradata  
[Teradata.com/t/partners/oracle-corporation](http://Teradata.com/t/partners/oracle-corporation)

Contacts  
 Steve Kamyszek – [Stephen.Kamyszek@Teradata.com](mailto:Stephen.Kamyszek@Teradata.com)  
 Partner Integration Lab

Tom Slavinsky – [Tom.Slavinsky@Teradata.com](mailto:Tom.Slavinsky@Teradata.com)  
 Teradata Product Manager, Oracle Strategic Partnership

Todd Cannaday – [Todd.Cannaday@Teradata.com](mailto:Todd.Cannaday@Teradata.com)  
 Teradata Alliances Manager

## New SQL

```

SELECT(( cp_a300."LEVEL_2") || ('_D2')), (( cp_b297."LEVEL_1") || ('_P1')),
      (( cp_c282."LEVEL_1") || ('_C1')), (( cp_e275."LEVEL_4") || ('_A4')),
      (( cp_f283."LEVEL_1") || ('_S1')), SUM ( cp_303."BASE_BALANCE"),
      SUM ( cp_303."BASE_CREDIT_AMOUNT"), SUM ( cp_303."BASE_DEBIT_AMOUNT"),
      COUNT (*)
FROM   ((( ( "tstman_sem"."oal2s_department_hier_d" cp_a300 JOIN "tstman_sem"."oal2s_product_hier_d" cp_b297
ON      (1=1) ) JOIN "tstman_sem"."oal2s_company_hier_d" cp_c282
ON      (1=1) ) JOIN "tstman_sem"."oal2s_account_hier_d" cp_e275
ON      (1=1) ) JOIN "tstman_sem"."oal2s_sub_account_hier_d" cp_f283
ON      (1=1) ) JOIN "tstman_sem"."OAI2S_GL_DETAIL_F" cp_303
ON      ( cp_a300."DEPARTMENT_KEY" = cp_303."DEPARTMENT_KEY" )
AND     ( cp_b297."PRODUCT_KEY" = cp_303."PRODUCT_KEY" )
AND     ( cp_c282."COMPANY_KEY" = cp_303."COMPANY_KEY" )
AND     ( cp_e275."ACCOUNT__KEY" = cp_303."ACCOUNT__KEY" )
AND     ( cp_f283."SUB_ACCOUNT_KEY" = cp_303."SUB_ACCOUNT_KEY" )
WHERE  ((( ( cp_a300."LEVEL_1") || ('_D1')) IN ( 'PT_D1' ) ) )
AND    ((( ( cp_b297."LEVEL_1") || ('_P1')) IN ( 'T_P1' ) ) ) )
AND    ((( ( cp_c282."LEVEL_1") || ('_C1')) IN ( 'T_C1' ) ) ) )
AND    ((( ( cp_e275."LEVEL_1") || ('_A1')) IN ( 'PT_A1' ) ) ) )
AND    ((( ( cp_f283."LEVEL_1") || ('_S1')) IN ( 'T_S1' ) ) ) )
GROUP BY (( cp_a300."LEVEL_2") || ('_D2')), (( cp_b297."LEVEL_1") || ('_P1')), (( cp_c282."LEVEL_1") || ('_C1')), (( cp_e275."LEVEL_4") || ('_A4')), (( cp_
f283."LEVEL_1") || ('_S1'))

```

Figure 10. New vs. Older SQL.

## Older-Style SQL

```

SELECT ((cp_a300."LEVEL_2")||('_D2')),((cp_b297."LEVEL_1")||('_P1')),
((cp_c282."LEVEL_1")||('_C1')),((cp_e275."LEVEL_4")||('_A4')),
((cp_f283."LEVEL_1")||('_S1')),SUM(cp_303."BASE_BALANCE"),
SUM(cp_303."BASE_CREDIT_AMOUNT"),SUM(cp_303."BASE_DEBIT_AMOUNT"),COUNT(*)
FROM (((("tstman_sem"."oal2s_department_hier_d" cp_a300 JOIN "tstman_sem"."oal2s_product_hier_d" cp_b297
ON (1=1)) JOIN "tstman_sem"."oal2s_company_hier_d" cp_c282
ON (1=1)) JOIN "tstman_sem"."oal2s_account_hier_d" cp_e275
ON (1=1)) JOIN "tstman_sem"."oal2s_sub_account_hier_d" cp_f283
ON (1=1)) JOIN "tstman_sem"."OAI2S_GL_DETAIL_F" cp_303
ON (cp_a300."DEPARTMENT_KEY" = cp_303."DEPARTMENT_KEY")
AND (cp_b297."PRODUCT_KEY" = cp_303."PRODUCT_KEY")
AND (cp_c282."COMPANY_KEY" = cp_303."COMPANY_KEY")
AND (cp_e275."ACCOUNT_KEY" = cp_303."ACCOUNT_KEY")
AND (cp_f283."SUB_ACCOUNT_KEY" = cp_303."SUB_ACCOUNT_KEY"))
WHERE (((cp_a300."LEVEL_4")||('_D4')) IN ('500_D4',
'501_D4', '508_D4', '509_D4', '510_D4', '511_D4', '512_D4',
'514_D4', '515_D4', '516_D4', '517_D4', '519_D4', '520_D4',
'521_D4', '522_D4', '530_D4', '535_D4', '540_D4', '550_D4',
'555_D4', '556_D4', '560_D4', '570_D4', '575_D4', '580_D4',
'585_D4', '590_D4', '595_D4'))
AND (((cp_b297."LEVEL_2")||('_P2')) IN ('000_P2',
'100_P2', '110_P2', '120_P2', '130_P2', '140_P2', '150_P2',
'200_P2', '210_P2', '220_P2', '230_P2', '240_P2', '250_P2',
'260_P2', '270_P2', '280_P2', '291_P2', '292_P2', '293_P2',
'294_P2', '300_P2', '400_P2', '500_P2', '600_P2', '601_P2',
'700_P2', '701_P2', '702_P2', '800_P2', '801_P2', 'T_P2'))
AND (((cp_c282."LEVEL_2")||('_C2')) IN ('00_C2',
'01_C2', '02_C2', '03_C2', '05_C2', '26_C2', '27_C2', '28_C2',
'29_C2', '35_C2', '70_C2', '81_C2', '82_C2', '83_C2', '95_C2',
'99_C2', 'T_C2'))
AND (((cp_e275."LEVEL_4")||('_A4')) IN ('1000_A4',
'1100_A4', '1110_A4', '1120_A4', '1130_A4', '1140_A4', '1150_A4',
'1160_A4', '1170_A4', '1200_A4', '1210_A4', '1212_A4', '1214_A4',
'1216_A4', '1217_A4', '1218_A4', '1219_A4', '1220_A4', '1222_A4',
'1225_A4', '1230_A4', '1231_A4', '1232_A4', '1240_A4', '1241_A4',
'1245_A4', '1247_A4', '1250_A4', '1252_A4', '1260_A4', '1270_A4',
'1280_A4', '1300_A4', '1310_A4', '1320_A4', '1330_A4', '1332_A4',
'1333_A4', '1334_A4', '1335_A4', '1336_A4', '1340_A4', '1400_A4',
'1410_A4', '1411_A4', '1420_A4', '1421_A4', '1430_A4', '1431_A4',
'1440_A4', '1441_A4', '1450_A4', '1451_A4', '1460_A4', '1465_A4',
'1470_A4', '1471_A4', '1472_A4', '1473_A4', '1474_A4', '1475_A4',
'1480_A4', '1481_A4', '1482_A4', '1483_A4', '1499_A4', '1500_A4',
'1510_A4', '1520_A4', '1530_A4', '1540_A4', '1550_A4', '1560_A4',
'1565_A4', '1570_A4', '1571_A4', '1580_A4', '1590_A4', '1600_A4',
'1620_A4', '1630_A4', '1640_A4', '1650_A4', '1660_A4', '1665_A4',
'1670_A4', '1680_A4', '1690_A4', '1700_A4', '1720_A4', '1770_A4',
'1810_A4', '1811_A4', '1812_A4', '1813_A4', '1814_A4', '1815_A4',
'1816_A4', '1817_A4', '1818_A4', '1820_A4', '1900_A4', '1901_A4',
'1902_A4', '1903_A4', '1950_A4', '1952_A4', '1953_A4', '1954_A4',
'1955_A4', '1956_A4', '1957_A4', '1958_A4', '1959_A4', '1960_A4',
'1961_A4', '1962_A4', '1963_A4', '1965_A4', '2000_A4', '2100_A4',
'2110_A4', '2120_A4', '2130_A4', '2210_A4', '2215_A4', '2220_A4',
'2225_A4', '2230_A4', '2240_A4', '2254_A4', '2255_A4', '2370_A4',
'2371_A4', '2372_A4', '2373_A4', '2374_A4', '2375_A4', '2376_A4',
'2377_A4', '2378_A4', '2530_A4', '2300_A4', '2310_A4', '2320_A4',
'2330_A4', '2340_A4', '2350_A4', '2399_A4', '2400_A4', '2410_A4',
'2420_A4', '2430_A4', '2440_A4', '2450_A4', '2510_A4', '2511_A4',
'2512_A4', '2520_A4', '2540_A4', '2550_A4', '2551_A4', '2555_A4',
'2560_A4', '2570_A4', '2580_A4', '2599_A4', '2600_A4', '3000_A4',
'3100_A4', '3200_A4', '3310_A4', '3315_A4', '3320_A4', '3330_A4',
'3400_A4', '3500_A4', '3999_A4', '4000_A4', '4110_A4', '4120_A4',
'4130_A4', '4140_A4', '4150_A4', '4160_A4', '4161_A4', '4162_A4',
'4163_A4', '4164_A4', '4165_A4', '4166_A4', '4167_A4', '4168_A4',
'4170_A4', '4180_A4', '4501_A4', '4502_A4', '4503_A4', '4504_A4',
'4505_A4', '4506_A4', '4507_A4', '4508_A4', '4509_A4', '4510_A4',
'4511_A4', '4512_A4', '4513_A4', '4514_A4', '4515_A4', '4516_A4',
'4517_A4', '4518_A4', '4519_A4', '4520_A4', '4521_A4', '4522_A4',
'4523_A4', '4524_A4', '4525_A4', '4526_A4', '4527_A4', '4528_A4',
'4529_A4', '4530_A4', '4531_A4', '4532_A4', '4533_A4', '4534_A4',
'4900_A4', '5000_A4', '5020_A4', '5030_A4', '5040_A4', '5050_A4',
'5060_A4', '5110_A4', '5200_A4', '5210_A4', '5211_A4', '5220_A4',
'5221_A4', '5230_A4', '5240_A4', '5241_A4', '5250_A4', '5260_A4',
'5270_A4', '5280_A4', '5290_A4', '5300_A4', '5310_A4', '5312_A4',
'5315_A4', '5316_A4', '5320_A4', '5321_A4', '5322_A4', '5323_A4',
'5330_A4', '5340_A4', '5341_A4', '5360_A4', '5370_A4', '5380_A4',
'5390_A4', '5400_A4', '5500_A4', '5600_A4', '5700_A4', '5800_A4',
'5810_A4', '6000_A4', '6100_A4', '6110_A4', '6115_A4', '6120_A4',
'6125_A4', '6140_A4', '6145_A4', '6150_A4', '6155_A4', '6160_A4',
'6165_A4', '6170_A4', '6175_A4', '6176_A4', '6501_A4', '6502_A4',
'6503_A4', '6504_A4', '6505_A4', '6506_A4', '6507_A4', '6508_A4', '6509_A4',
'6510_A4', '6511_A4', '6512_A4', '6513_A4', '6514_A4', '6515_A4',
'6516_A4', '6527_A4', '7000_A4', '7020_A4', '7030_A4', '7040_A4',
'7050_A4', '7060_A4', '7070_A4', '7080_A4', '7090_A4', '7092_A4',
'7094_A4', '7096_A4', '7099_A4', '7100_A4', '7110_A4', '7120_A4',
'7130_A4', '7131_A4', '7151_A4', '7152_A4', '7153_A4', '7155_A4',
'7210_A4', '7220_A4', '7230_A4', '7299_A4', '7405_A4', '7410_A4',
'7420_A4', '7430_A4', '7440_A4', '7450_A4', '7460_A4', '7470_A4',
'7510_A4', '7515_A4', '7520_A4', '7530_A4', '7531_A4', '7540_A4',
'7560_A4', '7570_A4', '7580_A4', '7590_A4', '7610_A4', '7615_A4',
'7620_A4', '7621_A4', '7625_A4', '7630_A4', '7640_A4', '7650_A4',
'7660_A4', '7670_A4', '7675_A4', '7680_A4', '7685_A4', '7690_A4',
'7695_A4', '7699_A4', '7710_A4', '7720_A4', '7730_A4', '7740_A4',
'7750_A4', '7751_A4', '7752_A4', '7753_A4', '7754_A4', '7755_A4',
'7756_A4', '7757_A4', '7758_A4', '7766_A4', '7776_A4', '7777_A4',
'7778_A4', '7779_A4', '7870_A4', '7300_A4', '7320_A4', '7330_A4', '7340_A4',
'7350_A4', '7360_A4', '7365_A4', '7800_A4', '7810_A4', '7820_A4',
'7824_A4', '7825_A4', '7826_A4', '7827_A4', '7830_A4', '7840_A4',
'7842_A4', '7844_A4', '7846_A4', '7848_A4', '7850_A4', '7851_A4',
'7852_A4', '7853_A4', '7854_A4', '7855_A4', '7860_A4', '7900_A4',
'7910_A4', '7920_A4', '7930_A4', 'PT_A4'))
AND (((cp_f283."LEVEL_2")||('_S2')) IN ('0000_S2',
'1100_S2', '1200_S2', '1300_S2', '1400_S2', '2101_S2', '2102_S2',
'2103_S2', '2104_S2', '2105_S2', '2106_S2', '2107_S2', '2108_S2',
'2109_S2', '2110_S2', '2111_S2', '2112_S2', '2113_S2', '2114_S2',
'2115_S2', '2116_S2', '2117_S2', '2118_S2', '2119_S2', '2120_S2',
'2121_S2', '2122_S2', '2123_S2', '2124_S2', '2125_S2', '2126_S2',
'2127_S2', '2128_S2', '2129_S2', '2130_S2', '2131_S2', '2132_S2',
'2133_S2', '2134_S2', '2135_S2', '2136_S2', '2137_S2', '2138_S2',
'2139_S2', '2140_S2', '2141_S2', '2142_S2', '2143_S2', '2144_S2',
'2145_S2', '2146_S2', '2147_S2', '2148_S2', '2149_S2', '2150_S2',
'5001_S2', '5002_S2', '5003_S2', '5004_S2', '5005_S2', '5006_S2',
'5007_S2', '5008_S2', '5009_S2', '5010_S2', '5011_S2', '5012_S2',
'5013_S2', '5014_S2', '5015_S2', '5016_S2', '5017_S2', '5018_S2',
'5019_S2', '5020_S2', '5021_S2', '5022_S2', '5023_S2', '5024_S2',
'5025_S2', '5026_S2', '5027_S2', '5028_S2', '5029_S2', '5030_S2',
'5031_S2', '5032_S2', '5033_S2', '5034_S2', '5035_S2', '5036_S2',
'5037_S2', '5038_S2', '5039_S2', '5040_S2', '5041_S2', '5042_S2',
'5043_S2', '5044_S2', '5045_S2', '5046_S2', '5047_S2', '5048_S2',
'5049_S2', '5050_S2', '5051_S2', '5052_S2', '5053_S2', '5054_S2',
'5055_S2', '5056_S2', '5057_S2', '5058_S2', '5059_S2', '5060_S2',
'5061_S2', '5062_S2', '5063_S2', '5064_S2', '5065_S2', '5066_S2',
'5067_S2', '5068_S2', '5069_S2', '5070_S2', '5071_S2', '5072_S2',
'5073_S2', '5074_S2', '5075_S2', '5076_S2', '5077_S2', '5078_S2',
'5079_S2', '5080_S2', '5081_S2', '5082_S2', '5083_S2', '5084_S2',
'5085_S2', '5086_S2', '5087_S2', '5088_S2', '5089_S2', '5090_S2',
'5091_S2', '5092_S2', '5093_S2', '5094_S2', '5095_S2', '5096_S2',
'5097_S2', '5098_S2', '5099_S2', '5100_S2', 'T_S2'))
GROUP BY ((cp_a300."LEVEL_2")||('_D2')),((cp_b297."LEVEL_1")||('_P1'))),
((cp_c282."LEVEL_1")||('_C1')),((cp_e275."LEVEL_4")||('_A4')),
((cp_f283."LEVEL_1")||('_S1'))

```

Figure 10. New vs. Older SQL.

10000 Innovation Drive, Dayton, OH 45342 [Teradata.com](http://www.teradata.com)

Teradata and the Teradata logo are registered trademarks of Teradata Corporation and/or its affiliates in the U.S. and worldwide. Oracle is a registered trademark of Oracle Corporation. Teradata continually improves products as new technologies and components become available. Teradata, therefore, reserves the right to change specifications without prior notice. All features, functions, and operations described herein may not be marketed in all parts of the world. Consult your Teradata representative or Teradata.com for more information.

Copyright © 2014 by Teradata Corporation All Rights Reserved. Produced in U.S.A.

09.14 EB5844



TERADATA