Implementation of

IBM DB2 Cube Views

with

Siebel Business Analytics

November 2005
Brenda Nickelson
Jian Le
Dilip Kikla
IBM Silicon Valley Lab
San Jose, CA

Any recommendations provided here are intended only as a guide and should not be implemented on a production system without performance testing. Actual results will vary on an individual basis as determined by the tuning expertise available as well as hardware selection.
The recommendations and instructions contained in this document are provided for informational purposes only. Any implementation based on this document is undertaken at the user's exclusive risk.

The information in this paper is provided AS IS with no warranty of any kind, including implied warranties of merchantability or fitness for a particular purpose.

This paper includes illustrations of how IBM products perform in a customer environment. Many factors have contributed to the results and benefits described. IBM does not guarantee comparable results. IBM does not attest to its accuracy.

IBM, the IBM logo, DB2, Cube Views and Alphablox are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Siebel and Siebel Business Analytics are trademarks of Siebel Systems Inc. Other company, product and service names may be trademarks or service marks of others.

Parts of this document are copyrighted by IBM Corporation © Copyright IBM Corporation 2005

Table of Contents

Introduction to DB2 Cube Views

IBM® DB2® Cube Views™ is a feature of the IBM DB2 Universal Database™ (UDB) Data Warehouse Editions that improves the ability of DB2 UDB to perform OLAP processing. DB2 Cube Views can be used to simplify the deployment and management of OLAP solutions and improve the performance of OLAP tools and applications.

DB2 Cube Views has a multidimensional metadata management system and a query optimization system. The former includes an OLAP metadata repository, OLAP metadata management APIs, and an OLAP Center. OLAP Center is a GUI tool used to create, modify, import or export cube models and other metadata objects for use in OLAP tools. The multidimensional metadata is stored as part of a DB2 UDB database and consumed by other tools such as the query optimization system. The OLAP Center provides an access to the Optimization Advisor that analyzes the metadata stored in DB2 Cube Views' metadata repository and recommends summary tables. The summary tables store and index pre-aggregated data in sub-regions of a multidimensional cube (or data warehouse), depicted by the dimensions, levels and slices of an OLAP cube. SQL and OLAP queries whose results can be derived from these pre-aggregated data will benefit the most from these summary tables.

The metadata objects such as measures, dimensions, attributes, levels, hierarchies, cubes and cube models can either be created from scratch or be exchanged between the DB2 UDB catalog and OLAP tools. In the latter case, metadata bridges are often used to import and export metadata objects to and from the DB2 UDB catalog and are available for specific OLAP and database tools.

DB2 Cube Views exploits DB2 UDB features such as summary tables, different index schemas, functional dependency, aggregation functions and the DB2 Query Processor with its query rewrite functionality. The DB2 Query Processor rewrites incoming queries, transparently routing eligible queries to the appropriate summary tables for significant faster query performance.

For further information see IBM DB2 Cube Views Guide and Reference Version 8.2, SC18-7298-0.

Introduction to Siebel Business Analytics

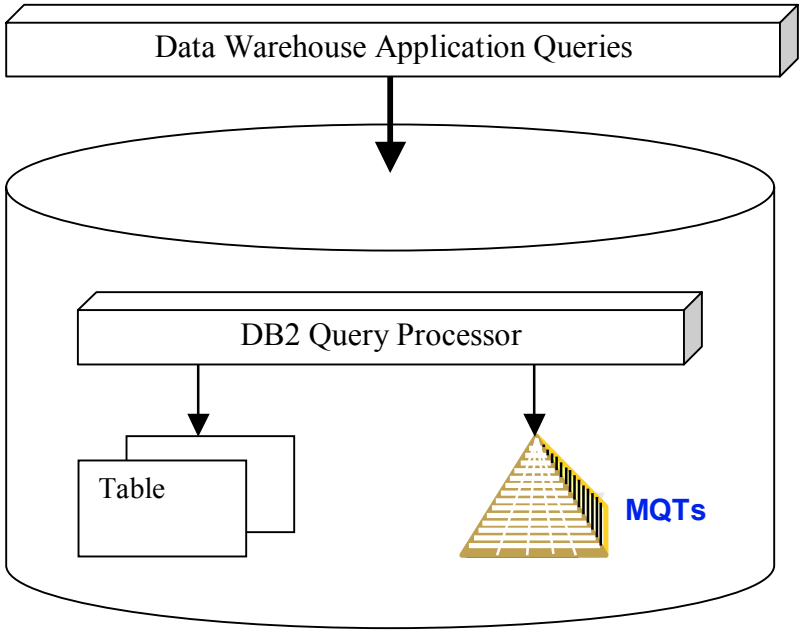
Siebel Business Analytics includes a comprehensive suite of market-leading analytic applications, as well as a next-generation enterprise BI platform. Siebel Business Analytics applications provide insight across the enterprise value chain with applications for optimizing business performance in sales, service, contact center, marketing, finance, supply chain, HR/workforce, and executive management. Siebel Business Analytics delivers complete, pre-built solutions that are based on industry and analytic best practices. Siebel Business Analytics applications drive faster deployment, lower TCO, and superior business value. The product is delivered as part of the analytic applications or available as an independent BI platform. This enables organizations to easily extend the analytic applications or build custom solutions to meet their unique business needs.

Just as sales models vary widely, analytics solutions must be tailored to meet the unique needs of each industry. Siebel Business Analytics applications are designed for more than 20 different industry sectors. They include data models encompassing industry-specific data, key industry metrics, and guided analytics and workflow incorporating industry best-practice processes. And, all of them can be easily and rapidly adapted to the specific needs of each individual organization.

Siebel Business Analytics applications come with pre-built extract, transform, and load (ETL) adapters and business logic to tap into a multitude of common operational applications and data sources, including Siebel, SAP, PeopleSoft, Oracle Applications, call center operational information such as IVR and CTI data, Web logs, and a host of other systems. These applications include an enterprise data warehouse design with common, conformed dimensions, enabling true cross-value-chain intelligence. In addition, the Siebel Business Analytics platform enables real-time access and intelligence across virtually any enterprise data source. The result is that Siebel Business Analytics provides a true cross-enterprise view, regardless of where data may be physically stored.

Siebel Analytics and DB2 Cube Views Integration

SQL queries in Data Warehouse applications usually involve large volumes of data stored in a relational database. To improve query performance of these applications, new functionality has been added to DB2 UDB that consistently reroutes application queries to some materialized query tables (MQTs). The following schematic flowchart depicts this query optimization process.



DB2 Cube Views product has been designed to recommend MQTs and indexes of MQTs that users can deploy in DB2 UDB to help them optimize their application queries. DB2 Cube Views product utilizes a user's data warehouse schema information and actual data to recommend MQTs and their indexes.

Siebel Business Analytics is a suite of applications that has a data modeling component. The objective in data modeling is to design a business model from a business analysts' point of view. To accomplish the business view, the administrator has to first define a physical model, and then construct a logical business model from this physical model. The design should be such that the business analyst can structure queries representing the business questions. The Siebel Analytics Server's repositories include both the physical and logical data models.

Siebel Analytics uses dimensional models to map objects logically to a business model. The business model is developed from relevant dimensional attribute data. Dimensional models are built on a star schema that model measurable facts by using one or more fact tables and a set of dimension tables that join fact tables to form a star-like topology.

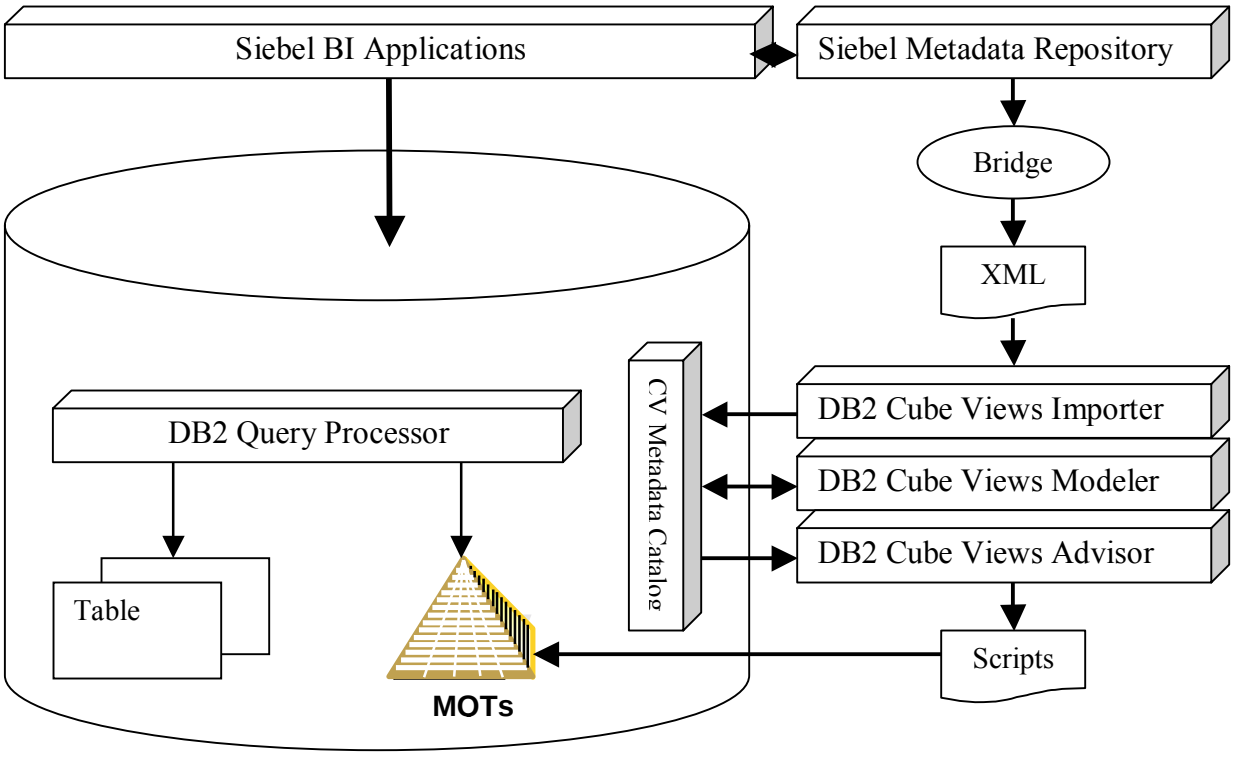
To understand the business model, the Administrator must first identify the
Measures,
Dimensions, and
Hierarchical relationships.

To understand the physical model, the Administrator must first identify
Entity relational schemas,
Dimensional schemas (star schema and snowflake),
Primary key-foreign key relationships, and
Unique key-foreign key relationships.

Siebel Analytics aids the Administrator in building repositories that consists of physical models, logical models and presentation models. With well designed business models, the Siebel Analytics tool can construct business oriented queries or logical queries.

DB2 Cube Views is another tool able to create physical, logical, and presentation modeling. The descriptions of these models are stored in the DB2 Cube Views' metadata. Since DB2 Cube Views query optimization component only works with DB2 Cube Views metadata, a user needs to convert the model information captured in Siebel's metadata into DB2 Cube Views metadata. To facilitate this metadata conversion process, a Siebel bridge program has been developed that uses the Siebel metadata as input and generates a metadata XML file compatible with DB2 Cube Views metadata. After the converted metadata is imported into DB2 Cube Views' metadata repository, a user can use this metadata information to recommend MQTs that contain pre-computed summary data related to a business model. After these recommended MQTs are created in DB2 UDB, the DB2 Query Processor will match every incoming Siebel query to these MQTs at query runtime to optimize query's performance.

A schematic flowchart of these operations is shown as follows.



Implementation Steps

The major steps to execute the Siebel Cube Views Generator (Siebel Bridge) and DB2 Cube Views Optimization are:

- Construct CubeViews Model
  - Execute the Siebel Bridge
  - Import XML metadata
  - Create new model
- Verify database integrity and settings
- Run Advisor
- Populate summary tables
- Verify query reroute
- Adjustments

Construct DB2 Cube Views Model

Execute the Siebel Cube Views Generator

The DB2 Cube Views Generator is invoked from the command line or embedded in a batch file. The command-line executable is named SACubeViewsGen.exe.

Syntax:

SACubeViewsGen -r "PathAndRepositoryFileName" -u <UserName> -p <Password> -f "InputFileNameAndPath"

where
-r repository filename (quotation marks are required when long format or has spaces)
-u username
-p password
-f input filename

Input File

The input file is a text file containing the parameters in the following 4 lines:
BUSINESS_MODEL = "*[name of business model]*"
PHYSICAL_DATABASE = "*[name of physical database]*"
RUN_AS_USER = "*[username]*"
OUTPUT_FOLDER = "*[full path and filename]*"

Output Files

Running the CubeViews Generator creates the following files in the specified output folder:

- XML files (encoded in UTF8)
  One XML file is created for each specified business model. It contains all DB2 Cube Views metadata objects that were mapped from metadata objects stored in Siebel Analytics' repository.

  The name of the XML file will match the business model name (without spaces), followed by the XML extension, for example, SalesResults.xml. The following is the syntax of an XML output file name:

  [BusinessModelNameWithNoSpaces].xml

- A SQL file containing the alias generation DLL

  SQL file is created for each specified business model only if aliases exist in the physical layer databases that are referenced in the specified business model. The alias file contains SQL commands that will be used to create aliases in the DB2 UDB database. The name of the SQL file will match the business model name (without spaces), followed by the SQL extension, for example, SalesResults-alias.sql. The following is the syntax of the alias-SQL file name:

  [BusinessModelNameWithNoSpaces]-alias.sql

Possible Errors from DB2 Cube Views Generator

- Unable to write to Log file : @1%ls.
  The log file specified in the NQSConfig.INI file might contain the wrong path, the user might not have write permissions to that folder, or the disk could be out-of-space.

- Run_as_user, @1%ls, is invalid.
  The user name is incorrect.

- Repository, @1%ls, is invalid or corrupt.
  The repository name might be incorrect, it might not exist in the given path, or the user might not have permission to read it.

- Physical Database, @1%ls, is invalid.
  The physical database name is incorrect.

- Business Model, @1%ls, is invalid.
  The business model name is correct.

- Authentication information provided is invalid.
  The specified username or password is incorrect.

- Path : "@1%ls" is invalid.
  The path or file name is incorrect.

DB2 Cube Views Metadata Objects

The following is a list of IBM DB2 Cube Views metadata objects:

- Attribute
- Join
- Measure
- Fact
- Dimension
- Hierarchy
- Level
- Cube Model
- Cube, Cube Fact, Cube Dimension, Cube Hierarchy, and Cube Level

The Siebel Cube Views Generator maps Siebel Analytics' metadata to DB2 Cube Views metadata with an exception that Siebel *Cube* objects are mapped to DB2 Cube Views *Cube Model* objects. The *Cube* objects in DB2 Cube Views represent logical sub cubes of a cube model.

Siebel Analytics Metadata Objects

Metadata	Description
Logical Fact Table	Made up of one or more logical fact table sources
Logical Fact Table Source	Made up of joins between multiple physical fact tables
Physical Fact Table	A table in the physical layer. It could also be an opaque view
Logical Dimension Table	Made up of one or more logical dimension table sources
Logical Dimension Table Source	Made up of joins between multiple physical dimension tables
Physical Dimension Table	A table in the physical layer. It could also be an opaque view. Might be shared with a physical fact table in the logical fact table source
Complex Join	A join defined using complex expressions
Key Join	A join defined using foreign key relationships
Dimensional Hierarchy	A hierarchy defined on dimensions. Can have multiple keys per level
Hierarchy Level	A level for each hierarchy. Should have a level key and related attributes
Measure	A column in which data is derived from expressions involving other attributes
Attribute	A column in the table
Alias	A synonym for a table name. Will be used in the place of physical tables in the cube view. DDL will be generated to create these in the database

Prepare Database for the DB2 Cube Views Metadata Deployment

The alias-SQL file generated by the DB2 Cube Views Generator should be executed before importing the XML file. The XML file generated by the DB2 Cube Views Generator contains the cube metadata in IBM DB2 Cube Views' metadata XML format.

To deploy cube metadata, first execute the alias-SQL File for IBM DB2 Cube Views and then import the XML file into the DB2 UDB database. This alias-SQL file can be executed using a SQL client on the database where the data warehouse is located. When executed, it creates aliases (synonyms) for tables in the database.

Sample alias-SQL file:

```
CREATE SYNONYM SIEBEL.ASYSMM_SCORES_29E1DA FOR
SIEBEL.ASYSMM_SCORES;
CREATE SYNONYM SIEBEL.W_ASSET_F_22627B FOR SIEBEL.W_ASSET_F;
CREATE SYNONYM SIEBEL.W_DAY_D_226152 FOR SIEBEL.W_DAY_D;
```

CREATE SYNONYM SIEBEL.W_DAY_D_226160 FOR SIEBEL.W_DAY_D;

See Siebel Analytics Administration Guide Version 7.8.2 for further information.

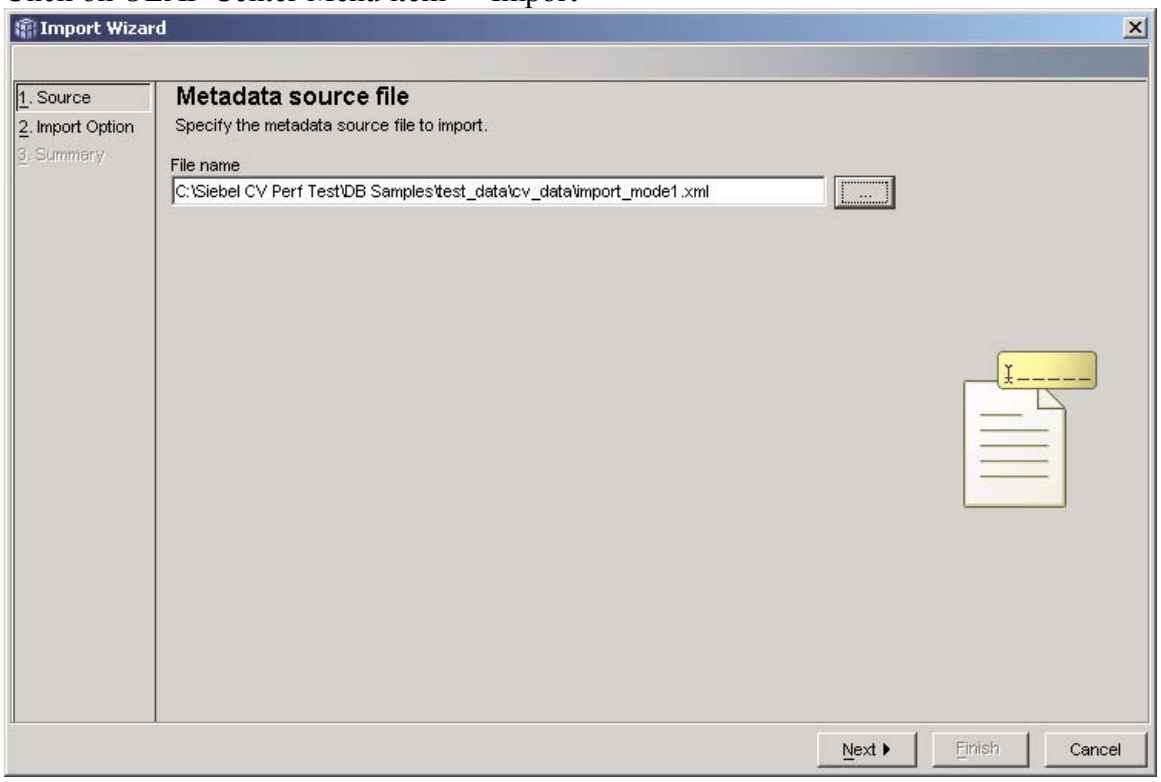
Import XML metadata

After executing the alias-SQL file, the XML file is imported into the DB2 UDB database. This file can be imported using the following IBM tools:

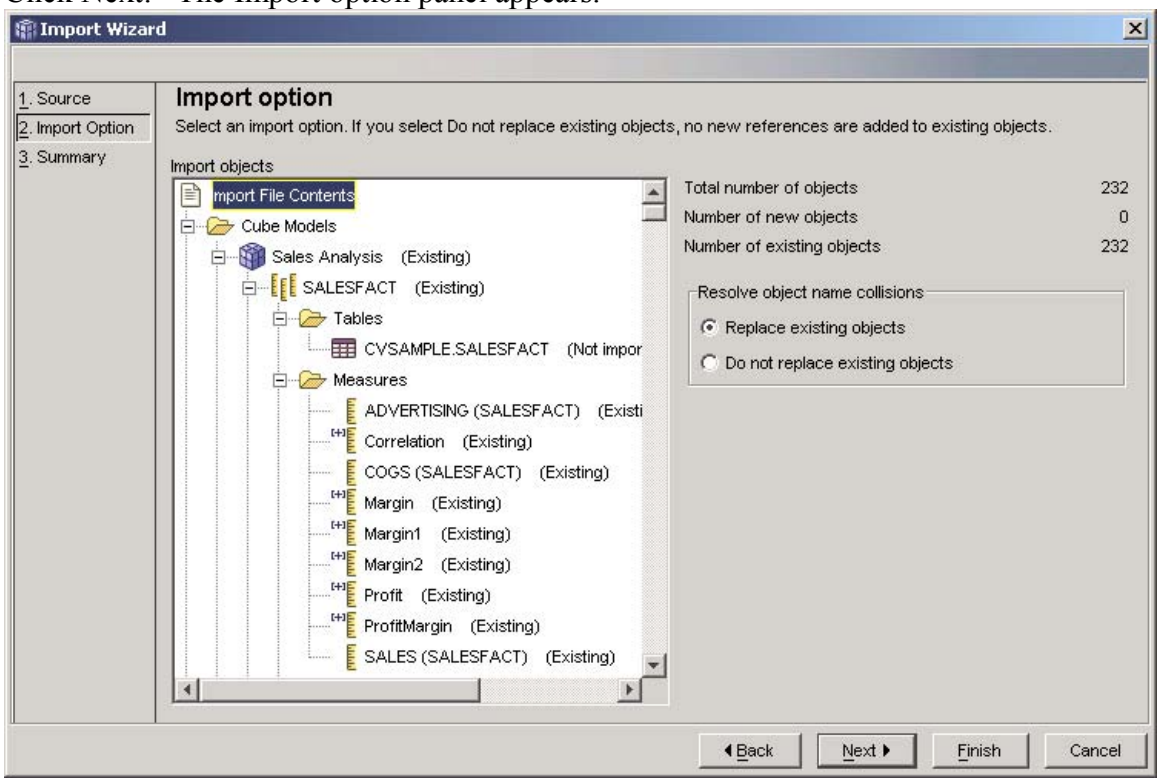
- IBM OLAP Center
- IBM command-line client utility (db2mdapiclient.exe). IBM ships this utility with DB2 UDB. For more information about using the command-line client utility, see the IBM DB2 Cube Views documentation.

Import using the IBM OLAP Center

Click on OLAP Center Menu item -> Import

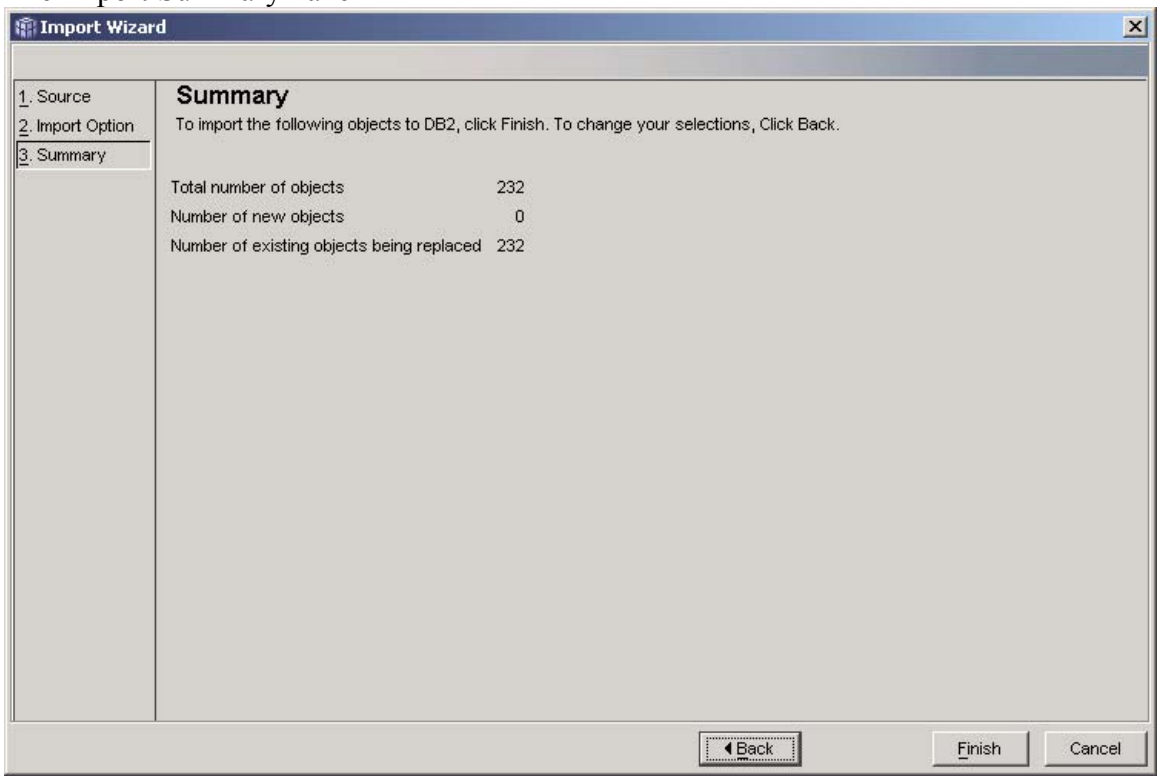


Click Next. The Import option panel appears.



There's an option to replace or not to replace existing objects. Click Next.

The Import Summary Panel



Click Finish to complete the import operation.

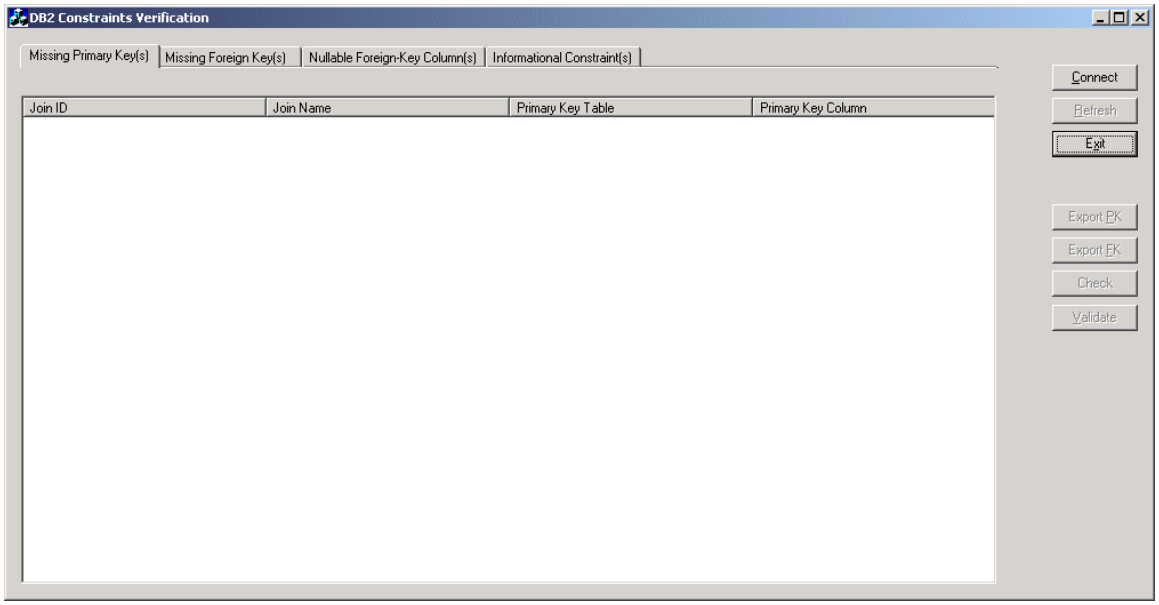
Create New Model

If there isn't a metadata model to import, a cube model can be constructed from scratch. On the left panel of the OLAP Center, click on the database name. Two options appear. One is to Create Cube Model and the other to Create Cube Model – Quick Start.
See IBM DB2 Cube Views User Guide on how to create a new cube model.

Verify Integrity and Settings

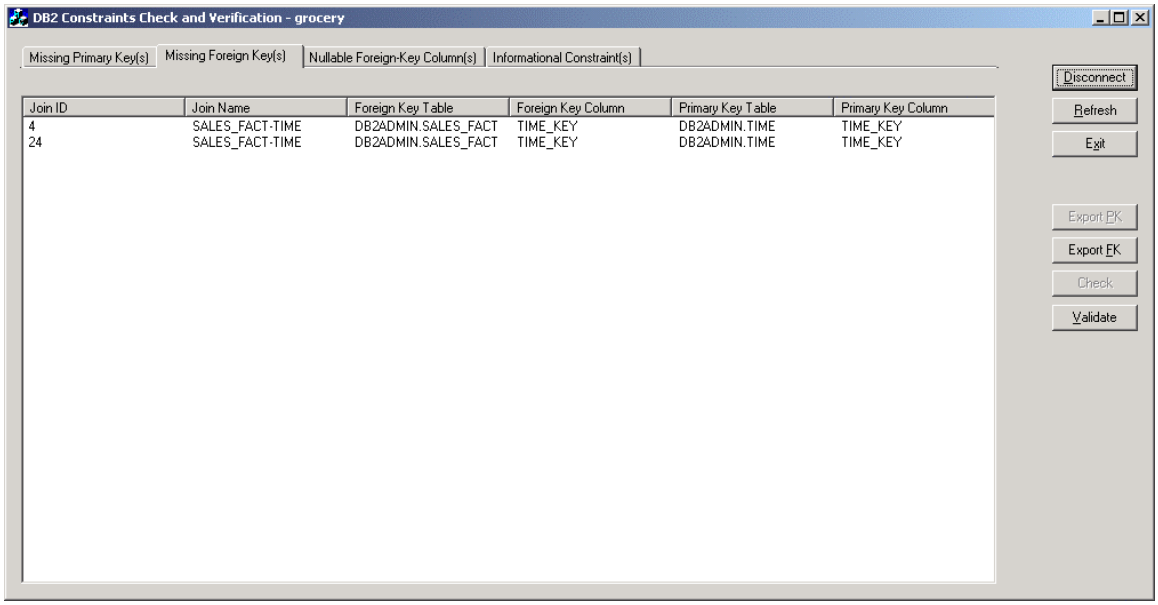
To ensure that the referential integrity constraints required for the DB2 Cube Views Advisor are available, the Referential Integrity Utility needs to be executed. A user also needs to enable a DB2 UDB database for query reroute and update the table and index statistics.

Execute Referential Integrity Utility



Click on Connect to enter database and provide a userid and password when prompted.

The utility checks for missing primary keys, foreign keys, nullable foreign key and validate informational constraints.



The above example has a missing foreign key. Click on the Export FK button. A DDL statement to create the missing foreign key is generated. A user can execute this DDL file in a SQL client.

Apply recommendations

Example of generated FK.txt

alter table DB2ADMIN.SALES_FACT add foreign key (TIME_KEY)
references DB2ADMIN.TIME(TIME_KEY) on delete restrict not enforced;

For details, see documentation Ref Integrity Users Guide 8.2.doc

Refresh Age

The DB2 UDB special register CURRENT REFRESH AGE affects whether an incoming query is matched to materialized query tables. When connected to the database, type

Update db cfg using dft_refresh_age any

in the DB2 UDB command window. The database will have to be restarted for the change to take effect.

Run Statistics

The Optimization Adviser uses the database statistics in making its recommendations. Hence, ensure that table and index statistics are current.

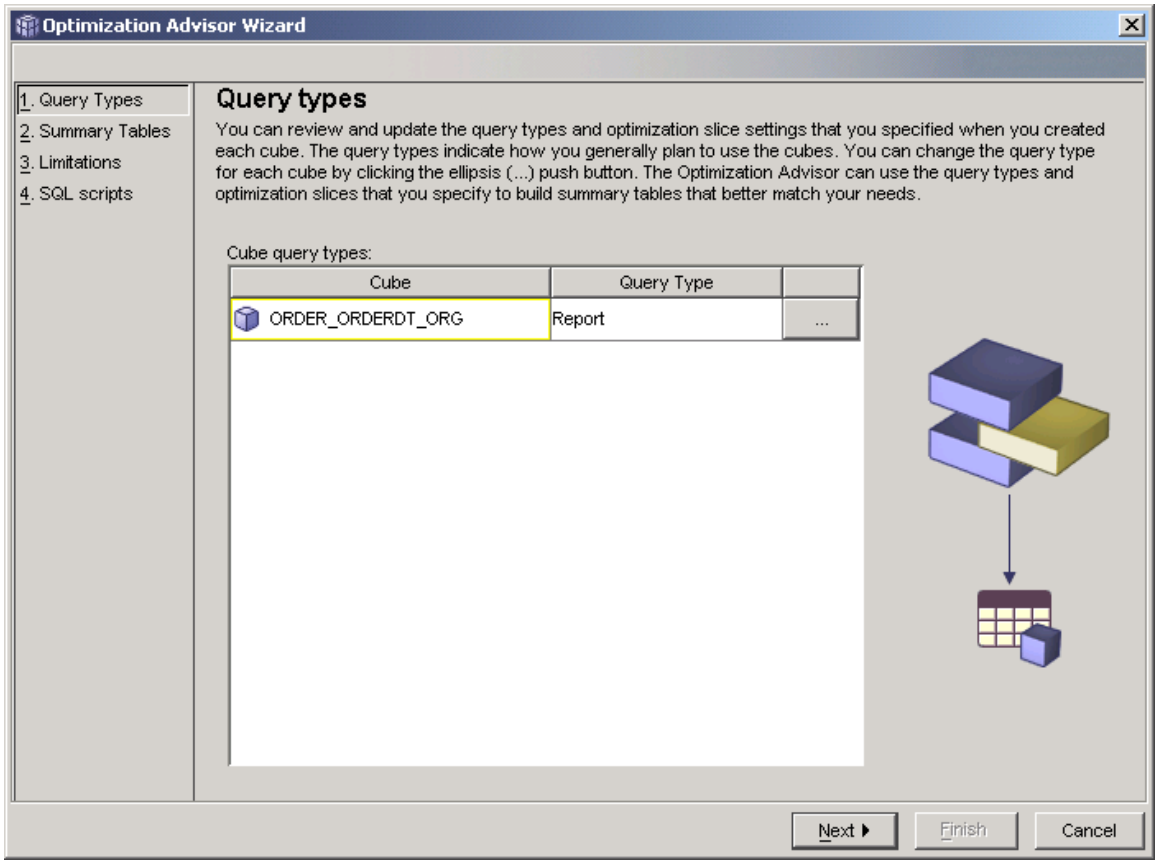
Run DB2 Cube Views Optimization Advisor

Run Advisor in Interactive Mode

The DB2 Cube Views Advisor can be initiated in two ways

1. On the left panel of the OLAP Center, highlight the Cube Model and right click. Select Optimization Advisor
2. From the Selected menu item, select Optimization Advisor

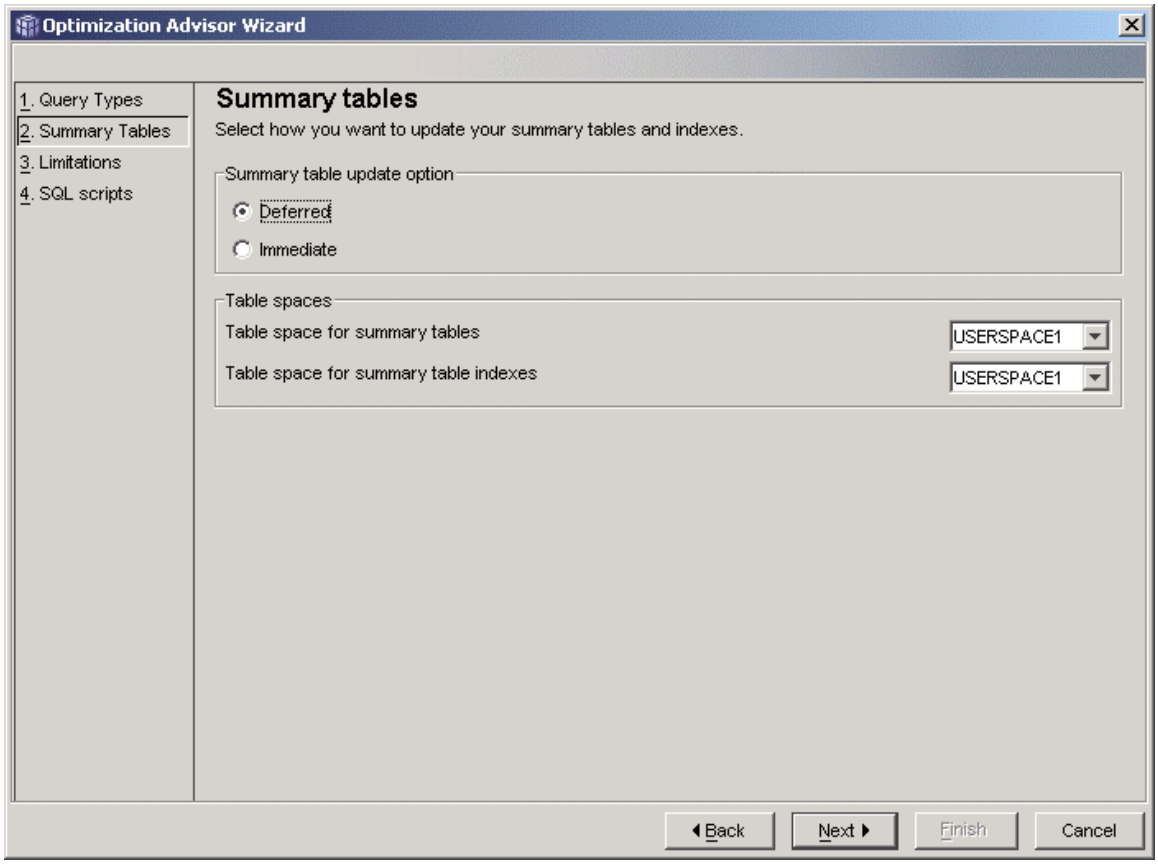
DB2 Cube Views Advisor will validate the cube model. An error message will appear if the cube model fails the validation process. If the query model passes the validation test, the Query Types panel appears.



This panel lists the cubes associated with the cube model. A query type can be selected from this panel. Since the Siebel Bridge maps Siebel Cube objects to DB2 Cube Views Cube Model objects, it is very unlikely that this panel list will appear.

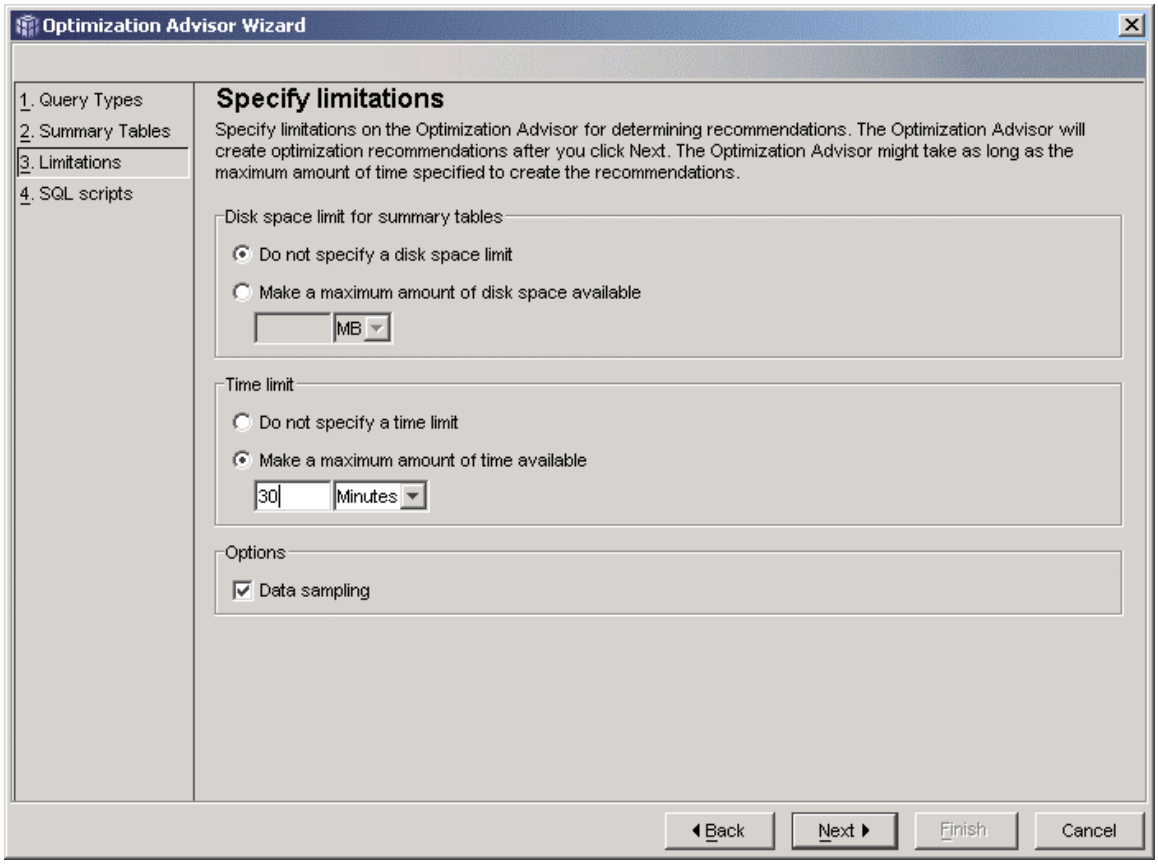
Click Next.

The Summary Tables panel appears. Select summary table update option Deferred or Immediate. Select table space(s) for summary table(s) and indexes.



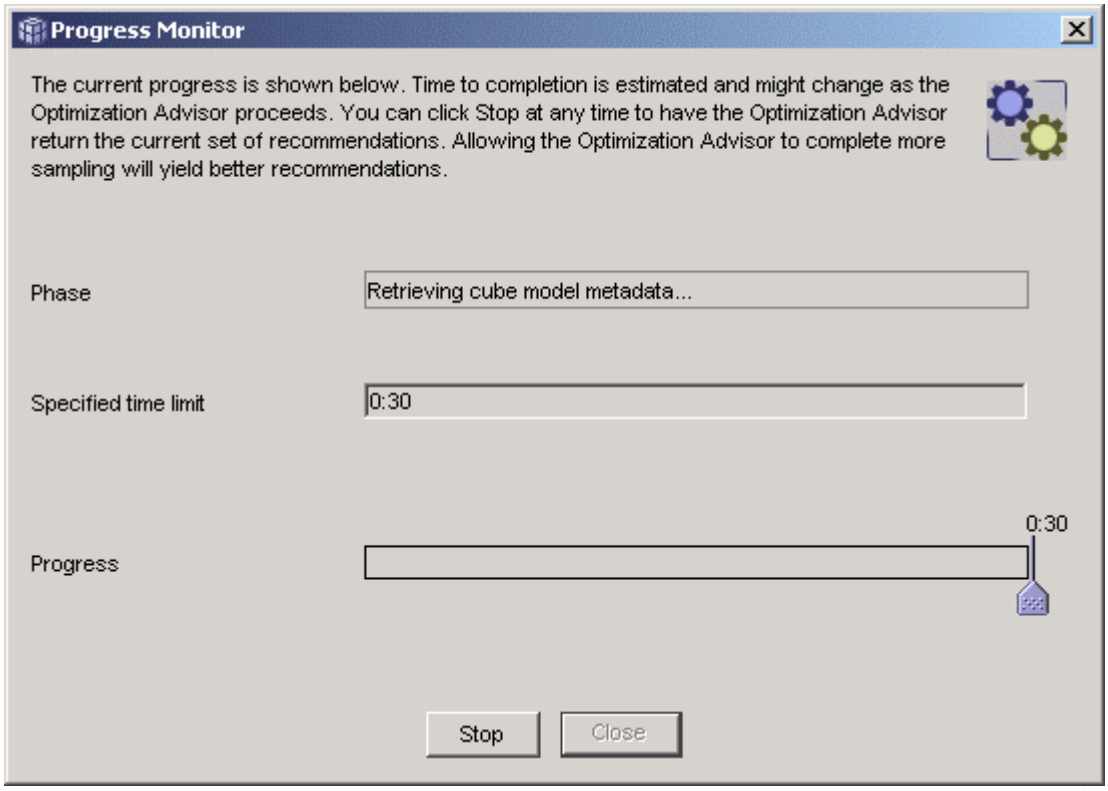
Click Next. The Limitations panel appears.

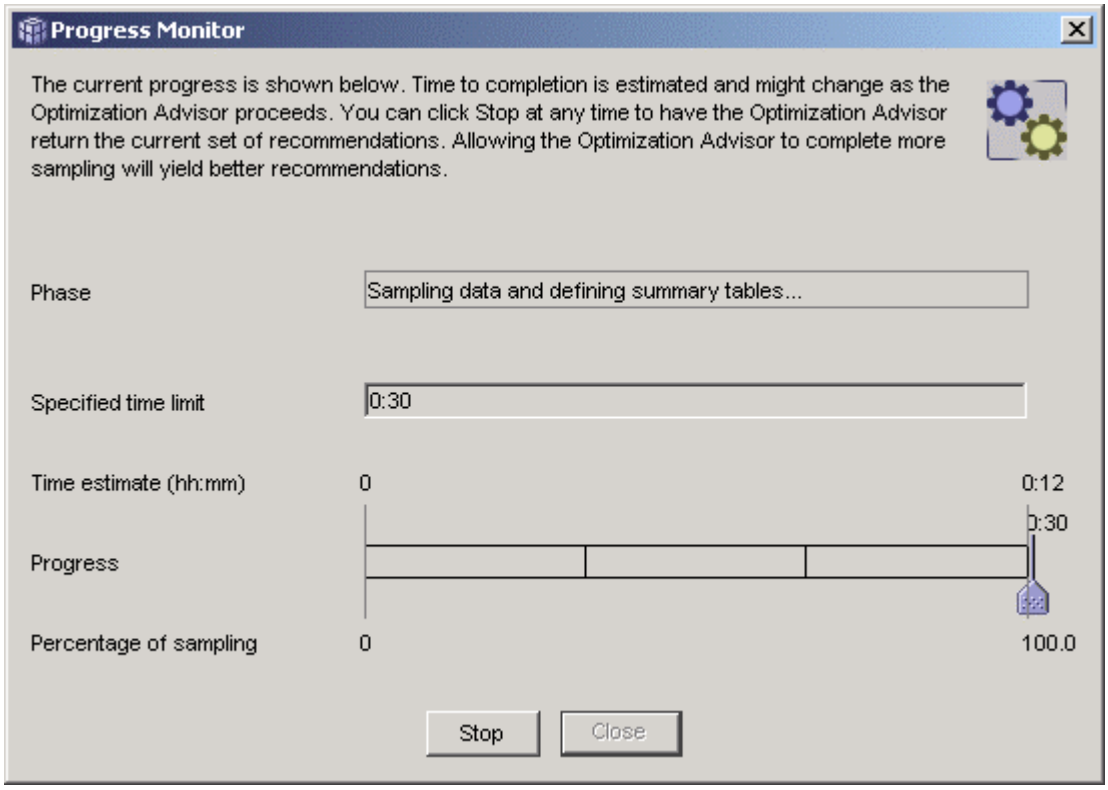
Specify disk space limit, time limit and data sampling option. It is not recommended to specify a time limit of more than two hours.

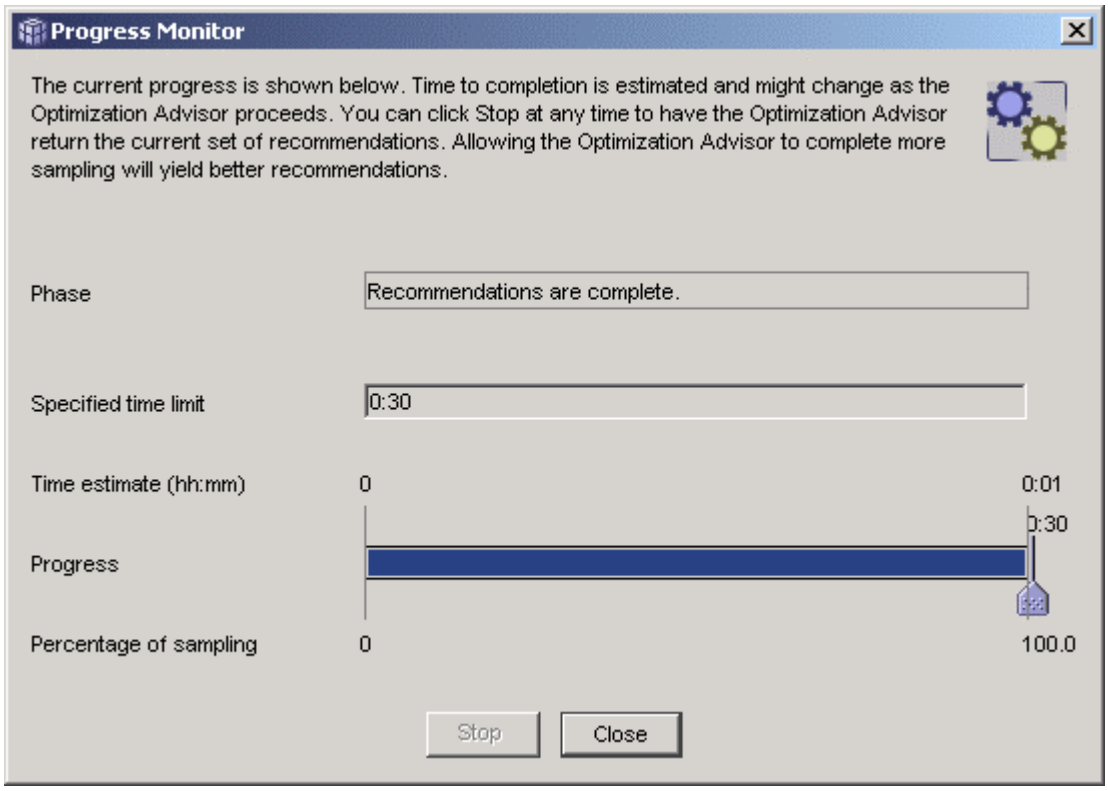


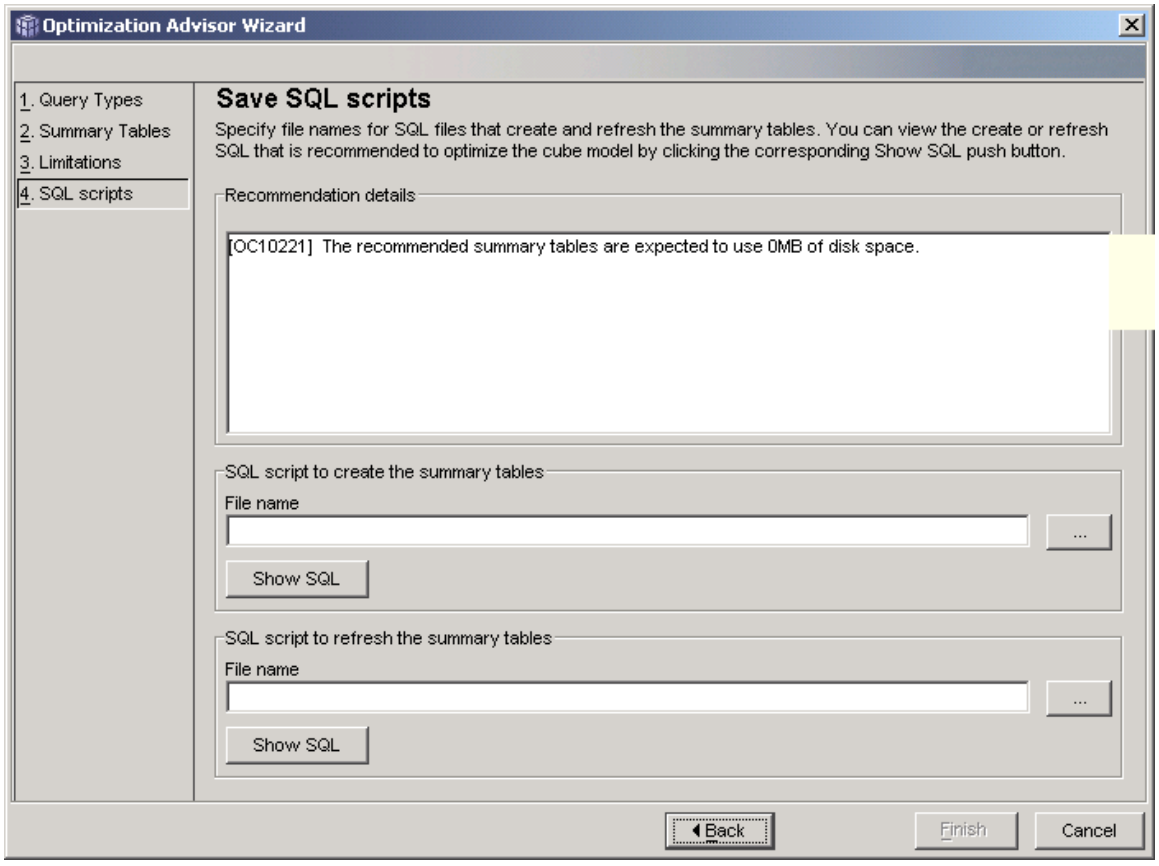
Click Next.

The Progress Monitor panel appears and it usually progresses through the following sample stages.






Progress Monitor
The current progress is shown below. Time to completion is estimated and might change as the Optimization Advisor proceeds. You can click Stop at any time to have the Optimization Advisor return the current set of recommendations. Allowing the Optimization Advisor to complete more sampling will yield better recommendations.
Phase
Recommendations are complete.
Specified time limit
0:30
Time estimate (hh:mm)
0
0:01
0:30
Progress
Percentage of sampling
0
100.0
Stop
Close


Optimization Advisor Wizard
1. Query Types
2. Summary Tables
3. Limitations
4. SQL scripts
Save SQL scripts
Specify file names for SQL files that create and refresh the summary tables. You can view the create or refresh SQL that is recommended to optimize the cube model by clicking the corresponding Show SQL push button.
Recommendation details
[OC10221] The recommended summary tables are expected to use 0MB of disk space.
SQL script to create the summary tables
File name
...
Show SQL
SQL script to refresh the summary tables
File name
...
Show SQL
Back
Finish
Cancel

From this panel, the generated SQL for creating and refreshing summary tables can be saved to the respective files. To populate the Summary tables, the generated SQL needs to be executed.

Run Advisor in batch mode

Create a batch file batchfilename.bat

batchfilename.bat contains:
db2mdapiclient -d *database* –u *userid* –p *password* -i adviseinput.xml -o adviseoutput.xml

where
-d database name
-u userid
-p password
-i input file name
-o output file name

adviseinput.xml is the input to the utility.

Example:

```
- <olap:request xmlns:olap="http://www.ibm.com/olap"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="8.2.0.1.0">
- <advise diskspaceLimit="0" timeLimit="0" sampling="yes" refresh="deferred">
  <cubeModelRef name="F_W_ORDER_F225C3" schema="SIEBEL" />
  </advise>
  </olap:request>
```

See sample in sqllib\samples\olap\xml\input.

Adviseoutput.xml is the output file containing the generated MQT scripts. The creating script and the refreshing script need to be manually separated from the output file.

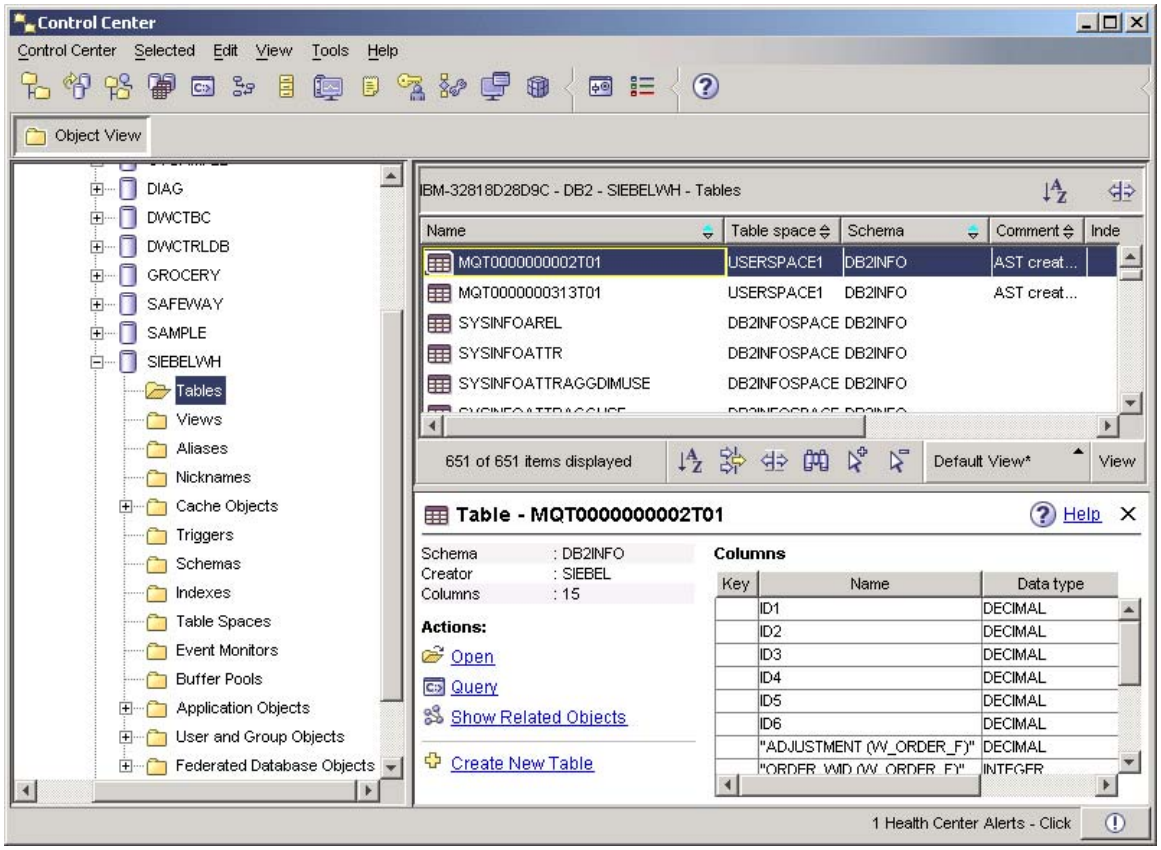
To execute, from a command prompt, type *batchfilename*

Populate Summary Tables

Execute generated SQL

From a DB2 UDB command window, type db2 –vtf *MQTCreateFileName.*

Check the output, to verify that there were no errors and the summary tables were created. The existence of summary tables can also be checked from the DB2 UDB Control Center. The MQTs should be listed as tables within the database. MQTs are named starting with MQT under a schema name of DB2INFO.



Verifying Query Reroute to MQT

Obtain test queries

To verify if queries are rerouted to MQTs, some test queries are needed. These test queries can be obtained by the following methods:

1) Supplied – e.g. Application log
2) Capture SQL using DB2 Event Monitor for statements
3) Other

For the Siebel integration project, these test queries were obtained by running Siebel Analytics application dashboards. The Siebel Analytics Application dashboards wrote the physical and logical SQLs to the Siebel log file. The physical SQL were extracted from the log file and used as the test query workload.

Execute Queries for timing

The next step is to execute the queries. First execute the queries without the MQTs to establish a performance baseline that will be compared with query execution time with MQTs. If the queries are issued against table columns whose data types are Vargraphics, DB2 Event Monitor can be used instead of DB2batch.

Set up the DB2 Event Monitor

From a DB2 UDB command window

1. type db2 connect to *database* user *userid* using *password*
   where *database* is database name
   userid is database user id
   password is database password
2. type db2 –vtf createmonitor.sql > createmonitor.out
   where the following statements are in the createmonitor.sql file

The output messages will go to the createmonitor.out file.

Createmonitor.sql file contains:

```
drop event monitor PMMON;

delete from stmt_PMMON;

update monitor switches using statement on;
```

```
CREATE EVENT MONITOR PMMON FOR STATEMENTS
       Where appl_id = 'XXXXXX.XXXX.XXXXXXXX'
        WRITE TO TABLE
          STMT (TABLE STMT_PMMON ,
            INCLUDES (
              APPL_ID, SEQUENCE_NO,STMT_OPERATION,STMT_TEXT,
START_TIME,STOP_TIME ));


set event monitor PMMON state = 1 ;
```

appl_id is obtained from DB2 UDB

On completion of the timing analysis, issue the following command from a DB2 UDB Command Window to stop the monitor.

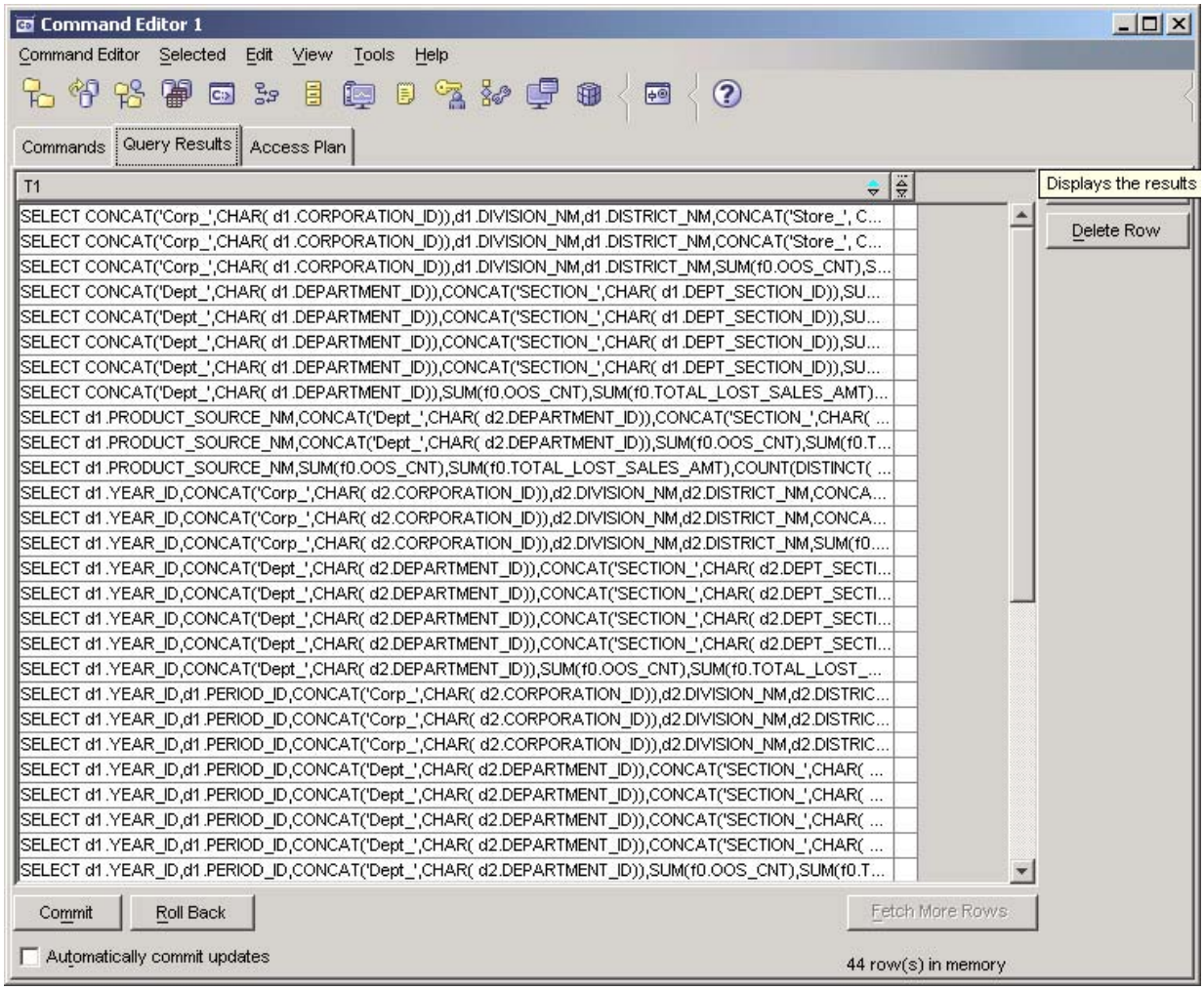
```
db2 set event monitor PMMON state = 0
```

From DB2 UDB Command Editor connect to the database and execute statement:

```
select char(substr(stmt_text,1,60)) as t1, stmt_operation , stop_time - start_time as t2
from siebel.stmt_pmmon
where stmt_operation = 6
order by t1, t2   ;
```

```
Note: We are filtering on the SELECT statements because the others are not
applicable. The monitor writes a row for many operations. By filtering on
operation 6, we will only get one entry for each statement.
```

The results will look like this:

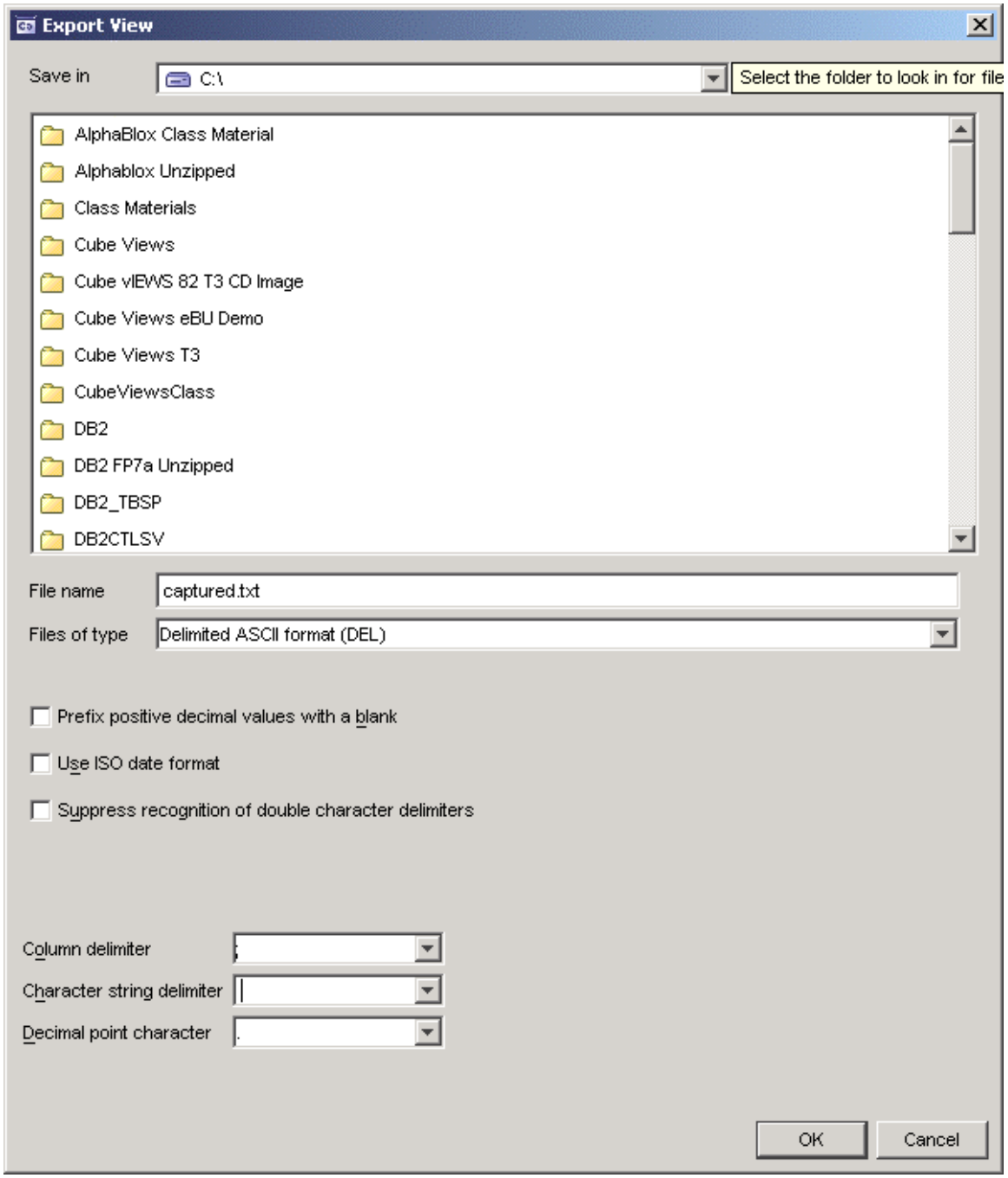


Click on the Selected menu item of this Command Editor and then click on Export

From Export View, select drive and path for exported file, specify a file name with a .txt extension in the File name edit box, and select ';' as column delimiter and enter space as character string delimiter.

Click OK

A file with the queries and timings has been saved. The timings can be captured in a spreadsheet program.


Export View
Save in
C:\
Select the folder to look in for file
AlphaBlox Class Material
Alphablox Unzipped
Class Materials
Cube Views
Cube vIEWS 82 T3 CD Image
Cube Views eBU Demo
Cube Views T3
CubeViewsClass
DB2
DB2 FP7a Unzipped
DB2_TBSP
DB2CTLSV
File name
captured.txt
Files of type
Delimited ASCII format (DEL)
Prefix positive decimal values with a blank
Use ISO date format
Suppress recognition of double character delimiters
Column delimiter
Character string delimiter
Decimal point character
OK
Cancel

Execute Queries with DB2 UDB Explain

From DB2 UDB command window, execute explain command as
db2expln -d *database* -u *username password* -o *explain.out* -f *queryfilename*.sql -g -z ;

db2expln
-d *database name*
- u *userid password*
-o *outputfile name*
- f *query statement filename*
-g indicates graphics mode
-z *termination character*

Search the output file for a string, "DB2INFO.MQT." When found, it indicates that a specific query was rerouted to the MQT.

Any queries that are not rerouted to a MQT should be analyzed.

Analyze Queries

To analyze queries that are not rerouted, the MQT scripts need to be compared with the query for attributes, measures and joins.

Check Joins

A query is considered for reroute if the query joins match the joins of a MQT, fully or partially.

Full match in Joins
Same set of joins in query as in MQT.

Partial match in Joins
If the query has extra join or MQT has extra joins–, the DB2 UDB optimizer will check to see if the extra joins in the MQT will affect the query results. The DB2 UDB optimizer will also check to see if the extra joins in a query are joinable with the MQT.

Check Attributes

A query is considered for reroute if its attributes match the attributes of a MQT, fully or partially.

Full Attribute match
Same attributes in MQT and in query

Example:

MQT

```
create table MYMQT as (
select s.REGION, SUM(f.SALES)
from FACT f, STORE s
where f.STORE_KEY=s.STORE_KEY
group by s.REGION
```

Query

```
select s.REGION, SUM(f.SALES)
from FACT f, STORE s
where f.STORE_KEY=s.STORE_KEY
and s.REGION='Pacific'
group by s.REGION
```

Rewritten Query

```
select REGION, SALES
from MYMQT
where REGION='Pacific'
```

The optimizer recognizes that the query is selecting the same attribute that is in the MQT.

Partial Attribute Match
The query has a subset of MQT attributes

Example:

MQT

```
create table MYMQT as (
select s.REGION, s.STATE, SUM(f.SALES)
from FACT f, STORE s
where f.STORE_KEY=s.STORE_KEY
group by s.REGION, s.STATE
```

Query

```
select s.REGION, SUM(f.SALES)
from FACT f, STORE s
where f.STORE_KEY=s.STORE_KEY
and s.REGION='Pacific'
group by s.REGION
```

Rewritten Query

```
select REGION, SUM(SALES)
from MYMQT
where REGION='Pacific'
group by REGION
```

In this case, the query only asks for one of the attributes in the MQTs group-by clause. The optimizer still uses the MQT. The rows in the summary table will have to be further summarized to return the result.

Partial Attribute Match
The query matches a subset of MQT attributes expanded by using functional dependency.

Example:

MQT

```
create table MYMQT as (
select s.STATE, SUM(f.SALES) as Sales
from FACT f, STORE s
where f.STORE_KEY=s.STORE_KEY
group by s.STATE
```

If there is a functional dependency defined between columns of s.State and s.Region, DB2 creates the following table at runtime:

MQT_2

```
select T.REGION, M.STATE, M.Sales
from MYMQT M, (Select distinct t.State, t.Region from Store t) as T(State, Region)
where M.State = T.State
```

Query

```
select s.REGION, SUM(f.SALES)
from FACT f, STORE s
where f.STORE_KEY=s.STORE_KEY
and s.REGION='Pacific'
group by s.REGION
```

This query matches a subset of MQT_2's attributes.

Check Measures

Check that the measures in the queries are in the MQT script. If missing, the measure needs to be added to the Cube Model.
Measures that can be derived from measures in a cube model are alright. Derived measures are measures in query can be calculated from measures in the MQT. For example if the MQT contains measure A, and measure B and the query contains measure A plus measure B, A plus B is a derived measure.

Query contains measures that do not match measures in MQT's definition query

Measure Matching
Direct
Same measures in query as in MQT

Indirect
Measure in query can be derived from measures in MQT

MQT contains:
sales, count

Query contains:
(sales / count)

The MQT is used because the query measure can be derived from measures in the MQT.

If the MQT has more measures than the query, the MQT will still be used.
If query contains measures missing from MQT, the MQT will not be used. The missing measures will have to be added to the Cube Model; rerun advisor

Distributed functions
Distributed functions such as sum, count, min, max are supported.
Higher levels can be summed from lower level aggregates.

Non-distributive functions
Non-distributive functions such as average, count distinct, stddev, variance, correlation, etc. must be matched at the same level. Higher level non-distributive aggregates - cannot be calculated from lower level non-distributive aggregates.

Adjustments

If the query is not rerouting to the MQT, adjustments may be needed to either or both, the DB2 Cube Views cube model and the Siebel Analytics business model.

1. Modify DB2 Cube Views Cube Model
   a. Define Joins
      Ensure joins in model match the joins in the query. If not, redefine the joins in the model
   b. Define Attributes
      Ensure that the attributes in the model match the attributes in the query.
   c. Define Measures
      Verify that the measures in the query exist in the model.
   d. Create a cube within cube model
      A cube (a subset of a cube model) that matches the query may have to be defined.
   e. Modify query type
      Modifying the query type (extract, drill down, report, custom) will vary the summarized area of the cube.
   f. Adjust time or space parameter

2. Modify Siebel Analytics business model
   Objects may be missing in the converted cube model. Modifying the Siebel business model can provide additional objects to the DB2 Cube Views cube model. After modifying the models and importing the modified model to IBM DB2 Cube Views, rerun the IBM DB2 Cube Views Optimization Advisor, repopulate the resultant summary tables, and re-verify the query reroute results to new MQTs.

Appendix

Siebel Analytics/DB2 Cube Views Integration Experience

Date: March 2005

Objectives

We are to test the Siebel Cube Views Generator program (a metadata bridge program) and verify that DB2 Cube Views MQTs do improve Siebel Analytics' query performance.

Plan

1. Obtain test queries from five Siebel Analytics warehouse apps
2. Benchmark the test queries without DB2 Cube Views' MQTs
3. Test the Siebel Cube Views bridge program by importing the bridge's output xml file into DB2 Cube Views metadata repository
4. Use the OLAP Center to verify the converted cube models
5. Run Optimization Advisor to recommend MQTs
6. Install the recommended MQTs in DB2 UDB
7. Benchmark the test queries with DB2 Cube Views' MQTs
8. Document query performance results

Challenges

We encountered a few limitations during our project such as

a. Missing referential integrity constraints from Siebel's physical model
b. Nullable foreign keys in Siebel's physical model
c. db2batch does not support vargraphics data types
d. db2 command line and db2 command editor do not support ODBC/CLI Escape Clauses
e. DB2 Cube Views does not work well with DB2 UDB aliases
f. Advisor quits with no recommendations
g. Missing measure and dimension objects from the converted Siebel models.

Referential Integrity

Prior to DB2 UDB V8.2.2, DB2 Cube Views Advisor requires that there must be a referential integrity constraint between a pair of tables in a data warehouse. We found many primary key, foreign key non conformances in Siebel's data warehouse models that had to be fixed. DB2 Cube Views Development staff has written a Referential Integrity Utility to assist users to detect and resolve the non conformance table-join objects. For details about this utility took, see Referential Integrity Utility for IBM DB2 Cube Views on http://www.alphaworks.ibm.com/tech. Starting from DB2 UDB V8.2.2, the referential integrity conformance is not strictly required. However, we still recommend our users to maintain them as much as they can in order to maximize the query coverage of MQTs.

Null Foreign Keys
We also found many null foreign keys in Siebel's data warehouse models. We had to redefine the foreign keys as not nullable in some cases. Starting from DB2 UDB V8.2.2, this step is not needed in most cases. However, in order to maximize the query coverage of MQTs, it is highly recommended that a user define foreign keys as not-null.

DB2BATCH
We tried to use DB2batch to get the query elapsed time. Since many queries contained column data defined as Vargraphics, db2batch could not be used, as Vargraphics is not supported. As an alternative for DB2batch, we used the DB2 UDB event monitor to measure the query time with and without MQTs.

DB2 UDB Command Line
Many queries contained ODBC escape clauses which are not supported by DB2 UDB command line processor and the DB2 UDB command editor.

Advisor quits with no MQT recommendation
Sometimes the Advisor would quit with no recommendations. This happened when the Facts object has 3 or more fact tables and a Cube Model object has more than 20 dimension tables. We got around this limitation by creating Cube objects with only a few dimensions determined by the query workload.

Missing measures from the input XML metadata file
Due to certain rules enforced within the Siebel bridge program, some measure objects were not exported to DB2 Cube Views metadata data. That led to mismatched measures in the recommended MQTs. When we checked for query reroute, we found that the measures referenced in the query were not included in the DB2 Cube Views cube model. We added the measures manually to the Cube Model and reran the DB2 Cube Views MQT recommendation process.

Performance Results at Siebel

	QueryType	RerouteStatus	QueryTime No MQT (sec)	QueryTime With MQT (sec)	% Increase
Sales Mgr					
Q3	Fact	Rerouted	0.134	0.007	94.8
Q4	Fact	Rerouted	0.792	0.01	98.7
Q5	Fact	Rerouted	0.252	0.006	97.6
Q6	Fact	Rerouted	0.025	0.008	68
Q7	Fact	Rerouted	0.475	0.006	98.7
Q8	Fact	Rerouted	0.098	0.006	93.9
Q11	Fact	Rerouted	0.635	0.014	97.8
Q12	Fact	Not rerouted	0.133	N/A	
Q13	Fact	Not rerouted	7.602	N/A	
Sales Rep					
Q2	Fact	Rerouted	0.408	0.017	95.8
Q3	Fact	Rerouted	0.494	0.014	97.2
Q4	Fact	Rerouted	0.038	0.008	78.9
Q5	Fact	Rerouted	0.145	0.06	58.6
Q11	Fact	Rerouted	2.538	0.048	98.1
Q13	Fact	Rerouted	0.388	0.007	98.2
Q14	Fact	Rerouted	0.824	0.01	98.8
Q15	Fact	Rerouted	0.792	0.01	98.7
Q16	Fact	Rerouted	0.24	0.006	97.5
Q17	Fact	Rerouted	0.026	0.009	65.4
Q18	Fact	Rerouted	0.495	0.006	98.8
Q19	Fact	Rerouted	0.099	0.006	93.9
Average Performance Increase					91

Sales Manager Queries Q12 and Q13 were not rerouted because of a missing measure in the XML metadata file, resulting in a missing measure in DB2 Cube Views' cube model. Consequently, the recommended MQT did not contain the missing measure either, resulting in a non reroute of the query.

Conclusion

We were able to see an improvement in performance on those queries that were routed to MQTs. The performance times were improved over 90% in most of the queries.

Related publications

Referential Integrity Utility for IBM DB2 Cube Views
http://www.alphaworks.ibm.com/tech

Siebel Analytics Administration Guide Version 7.8.2

IBM Redbooks

For information on ordering these publications, see
http://www.ibm.com/redbooks

DB2 Cube Views: A Primer, SG24-7002

Other publications

These publications are also relevant as further information sources:

IBM DB2 Cube Views Guide and Reference Version 8.2, SG18-7298-01
The Data Warehouse Toolkit by Ralph Kimball, ISBN 0-471-15337-0

Online resources

These Web sites and URLs are also relevant as further information sources:

IBM DB2 Cube Views Homepage
http://www.ibm.com/software/data/db2/db2md

IBM Software Homepage
http://www.software.ibm.com

IBM Information Management Homepage:
http://www.software.ibm.com/data

Notices

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give the reader any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, New York 10594, U.S.A.