Chapter 30.

July 2009

Welcome to the July 2009 edition of IBM InfoSphere Information Server Developer's Notebook. This month we answer the question;

> In a prior edition of this document, you discussed federated gateways including use of IBM MQ/Series. Can you offer detail on how to implement query re-write and materialized query tables using IBM InfoSphere Federation Server?
>
> *Excellent question! We love the benefits of federated gateways, and while one might assume that adding another access layer decreases observed performance, that simply is not true, as we will prove here.*
>
> In addition to the technical features detailed below, we also spend a bit of time on common practice and increasing maintainability of one's database related applications. Its one thing to design and deliver a given system, what are the challenges we need also to address over time-

Software versions

All of these solutions were *developed and tested* on (IBM) InfoSphere Information Server (IIS) version 8.1.1, using the Microsoft Windows XP/SP3 platform to support IIS client programs, and a RedHat Linux Advanced Server 5 (RHAS 5) 32 bit SMP server (Linux kernel version 2.6.9-67.EL-smp) to support the IIS server side components.

IBM InfoSphere Information Server allows for a single, consistent, and accurate view of data across the full width of the corporate enterprise, be it relational or non-relational, staged or live data. As a reminder, the IBM InfoSphere Information Server product contains the following major components;

> WebSphere Business Glossary Anywhere™, WebSphere Information Analyzer™, WebSphere Information Services Director™, WebSphere DataStage™, WebSphere QualityStage™, WebSphere Metadata Server and Metabridges™, WebSphere Metadata Workbench™, InfoSphere Federation Server™, Classic Federation™, Event Publisher™, Replication Server™, InfoSphere Data Architect™, DataMirror Transformation Server™, and others.

Obviously, IBM InfoSphere Information Server is a large and capable product, addressing many strategic needs across the enterprise, and supporting different roles and responsibilities.

30.1 Terms and core concepts

Federated gateways

We will let the April/2009 edition of this document act as our primer on the topic of federated gateways; overview what they are, how they function. An example of Why we use them, and the problems they solve might be demonstrated in the true story below-

> A regional health care provider built their two primary business application systems; application source code, data models, the whole thing. Over time, 400+ users have built 1000's of custom reports using MS/Access that have become critical to the business. These reports access the source systems natively/directly using ODBC.
>
> Laws and regulatory oversight have changed and increased, and while this customer was reasonably satisfied, meeting the minimum and core needs of their operational systems, they are unable to further adopt the increased application programming burden of operational systems, plus reporting, plus governance and more. While it would be easy and cost effective to replace the now legacy, home grown applications systems with commercial off the shelf,
>
> > *What do we do to continue to support the 1000's of now critical custom reports-*
>
> Answer: Install a federated gateway.

Figure 30-1 below, displays the current end user access to data from the story above in the form of red arrows. The blue arrows display access through a controlled access point, through a federated gateway.

Will this new software layer cause a delay in observed perform- No, as we will see and prove in the examples that follow.

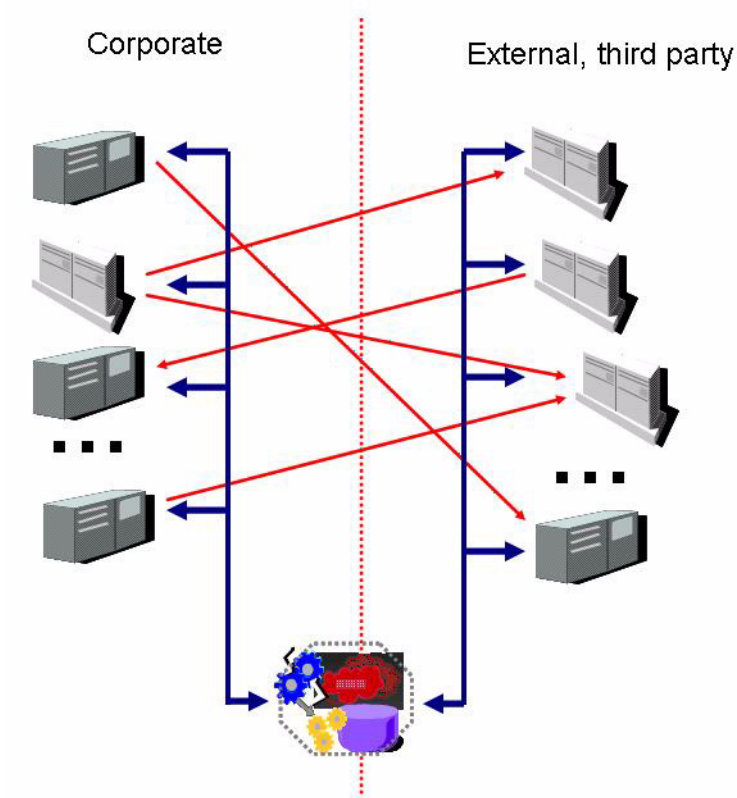

Figure 30-1 Red arrows, direct access. Blue arrows, directed access via gateways.

Physical tiers, IBM InfoSphere Information Server

As a single, integrated software product IBM InfoSphere Information Server (IIS) has 3 physical server side tiers. Those are;

- A shared meta data repository, in the form of a SQL database server.

  A free limited use license of DB2/UDB is included with IIS, or you can use Oracle or MS/SQLServer.

- A J2EE compliant application server.

  A free limited use license of WebSphere Application Server (WAS) is also provided.

- And the DSEngine proper, which provides the parallel framework on which all procedures and Jobs are able to run; 1 CPU or many, 1 node or multiple concurrent nodes.

The embedded version of DB2/UDB comes with a limited use license, or you can use IBM Infosphere Federation Server in this same role. While you have to have licensed access from DB2/Federated to Oracle, DB2/Federated to MS/SQLServer and others, access from DB2 to DB2 or DB2 to IBM Informix IDS functionality is enabled right out of the box.

Note: IBM InfoSphere Federation Server is an integrated, optional component to IBM InfoSphere Information Server (IIS).

IBM InfoSphere Federation Server has a stand alone product name, because it can operate without an installation of IIS.

IFS contains an embedded version of DB2/UDB and thus can act as a replacement to DB2.UDB. That is why IFS can serve as the meta data repository to IIS.

IBM Informix IDS being our favorite database server, we will detail access to that system. If you installed Informix before installing DB2/UDB or IBM Information Server, you are already configured for use. If you are not using Informix, and are using instead Oracle, the concepts, procedures and capabilities are nearly identical.

IBM InfoSphere Federation Server, Object hierarchy

As with any software system, IBM InfoSphere Federation Server (IFS) has an object hierarchy; physical and logical terms and ideas. Figure 30-2 below, offers these terms and relationships between one another. A code review follows.

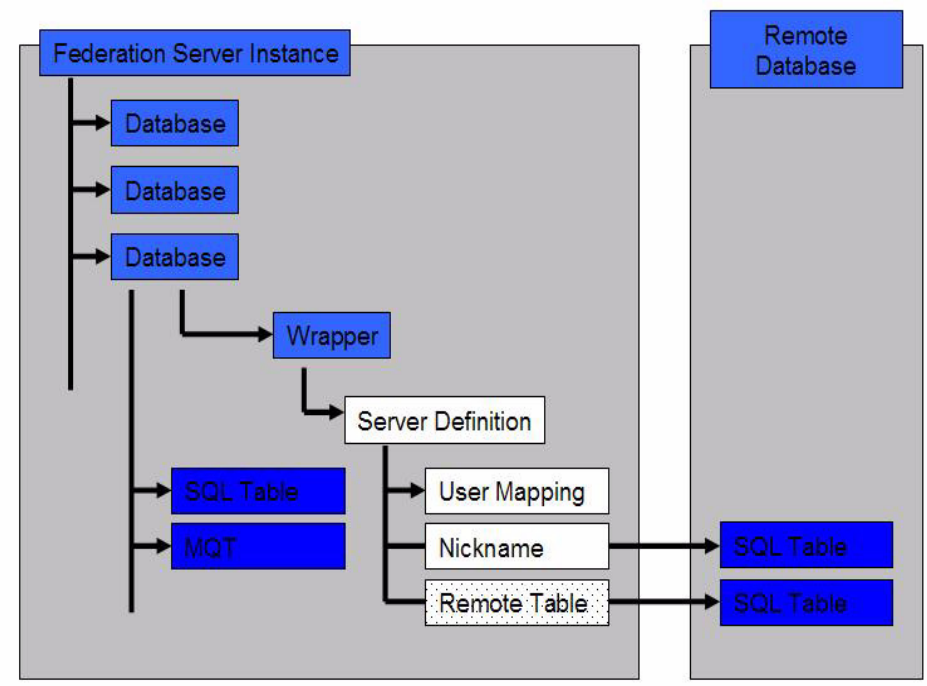

Figure 30-2 Federation Server object hierarchy.

Related to Figure 30-2, the following is offered;

- The IBM InfoSphere Federation Server (IFS) proper operates much like a DB2/UDB database server instance, with its own memory and process allocation at the server instance level.

  The IFS instance contains zero or more databases. Again, like DB2/UDB, these databases each have their own allocations of memory, process, and disk.

  Contrast this to say, IBM Informix IDS, where databases are more like logical entities; having all of the memory, process and disk allocated at the server instance level.

- A given IFS database contains zero or more objects entitled, Wrapper (definitions). These Wrappers are essentially a reference to an ODBC driver package, and are thus, vendor specific; Informix, Oracle, MS/SQLServer, etcetera.

  Wrappers may be created via SQL DDL, as we will demonstrate.

- A given Wrapper (definition) contains zero or more objects entitled, Server (definitions). A Server specifies connection criteria to a given database in a given (presumably) remote database server instance.

  Servers may be created via SQL DDL, as we will demonstrate.

- A given Server (definition) contains zero or more User Mappings. A User Mapping specific the user name and password.

  User Mappings may be created via SQL DDL, as we will demonstrate. We will not demonstrate integrating User Mappings with a LDAP security and authentication server.

- A given Server (definition) contains zero or more Nicknames. A Nickname is a reference to a single, whole table from a single, remote (Server).

  A Nickname is a logical term, requiring no real physical disk space allocation; it is merely a reference to the remote SQL Object to which it points.

Note: If you wish to refer to something more granular than a whole (remote) table, you have 2 simple options;

- Create a SQL View on the remote system, have the Nickname refer to that Object.
- Create a local SQL View or local SQL Materialized Query Table, and refer to that Object.

- A given Server (definition) contains zero or more Remote Tables.

  IBM InfoSphere Federation Server (IFS) has a feature called 'Transparent DDL'; from IFS, from a Server definition, you can create SQL tables in the *remote database server instance* using IFS SQL DDL syntax. While not entirely transparent (you can not make extended data types like Clobs or Blobs; they just are SQL Standard enough), this is still a pretty handy feature.

  This is one of the few sub-topics listed that we do not provide an example for in this document.

- And then back to the Database layer, a (federated) database can contain zero or more standard SQL tables, or zero or more Materialized Query Tables (MQTs).

  We give quite a bit of attention to MQTs and automatic query rewrite in this document.

Note: Materialized Query Tables (MQTs) are one of the funnier sub-topics in the history of SQL database servers.

At the time, there was quite a bit of competition in the area of database server performance; audited TPC Benchmarks for both OLTP and DSS style systems, and related. Oracle was the first vendor to deliver MQTs, calling their (same) feature Materialized Views. At the time, the TPC-D benchmark had 17 or so specified DSS style queries. Oracle pre-ran the queries and stored the results in SQL (result) tables. Using a feature called query rewrite, Oracle then ran the TPC-D queries against the source tables, internally rewrote the queries to use the pre-populated result tables. Oracle then claimed a 'shattering' nearly sub-second solution to the TPC-D benchmark, ignoring the time to load and pre-calculate the result tables. Classic Oracle.

Still, MQTs are a nice feature and can offer performance gains. *So what exactly is an MQT?*

An MQT is exactly as described above; a SQL table storing the results of an associated SQL SELECT statement, a query.

Combined with a feature called 'query rewrite', the database server has the opportunity to receive a query with calculations or other conditions, and redirect the query to use all or part of a precalculated result set. If you use the MQT often enough, the cost to create and maintain this table (this result set) is outweighed by the cost to have to create is live over and over.

30.2 Complete the following examples

In order to complete the following examples, you need the following;

- The examples that follow detail DB2/UDB using Federation Server capabilities to IBM Informix IDS. (You don't need Federation Server when going DB2/UDB to DB2/UDB or DB2/UDB to Informix.)

  Because we installed Informix first, then installed DB2/UDB, our system automatically configured. We do not detail how to manually configure DB2/UDB for federated capability with Informix; although we will say, its not that hard.

- If you wish to run these examples against Oracle, MS/SQLServer, or another supported third party database, you will need a full license to Federation Server, not just DB2/UDB.

  At that point, the examples detailed below translate quite well.

1. In your remote IBM Informix IDS database server, create the following;

   Our examples use the given identifiers-

   – Linux host name, rhhost.grid or rhhost
   – Informix database server instance name, rhhost_soc
   – Informix database name, root.DailyData_200907

   Watch case sensitivity below.

   The following sample SQL DDL/DML script is offered as example, Example 30-1. A code review follows.

Example 30-1 Remote table definition.

```
CREATE TABLE root.DailyData_200907
  (
  pk_col        INTEGER,
    --
  sale_item     CHAR(32),
  sale_date     DATE,
  sale_units    SMALLINT
  ) LOCK MODE ROW;

INSERT INTO root.DailyData_200907 VALUES
  (101, "Kayak    ", "12/15/1999", 5);
INSERT INTO root.DailyData_200907 VALUES
  (102, "Kayak    ", "12/15/1999",10);
INSERT INTO root.DailyData_200907 VALUES
  (103, "Paddle   ", "12/15/1999", 5);
INSERT INTO root.DailyData_200907 VALUES
  (104, "Life Vest", "12/15/1999", 5);

INSERT INTO root.DailyData_200907 VALUES
  (105, "Kayak    ", "12/16/1999", 5);
INSERT INTO root.DailyData_200907 VALUES
  (106, "Helmet   ", "12/16/1999", 5);
INSERT INTO root.DailyData_200907 VALUES
  (107, "Life Vest", "12/16/1999", 5);
INSERT INTO root.DailyData_200907 VALUES
  (108, "Kayak    ", "12/16/1999",10);

INSERT INTO root.DailyData_200907 VALUES
  (200, "Kayak    ", "12/16/2000",10);
INSERT INTO root.DailyData_200907 VALUES
  (200, "Kayak    ", "12/16/2000",10);

UPDATE STATISTICS;
```


A code review related to Example 30-1 contains;

- Basically this is just a SQL CREATE TABLE.
- We actually repeated the last 2 SQL INSERT statements 10,000 times in order to give this table some weight.
- The SQL UPDATE STATISTICS is considered key, since Federation Server will use that data for its own calculations.

2. In Federation Server, or as is our actual case, in DB2/UDB, create the following;

   Ultimately we wish to demonstrate query re-write, which is disabled by default. (Preserve the legacy behavior before query rewrite existed.) And we have some diagnostic statements below that are not required.

   A code review follows Example 30-2.

Example 30-2 Making the Federation Server objects,.

```
UPDATE DB CFG USING dft_refreshage ANY;

FORCE APPLICATIONS ALL;

DEACTIVATE DB cust_db;


CONNECT TO cust_db
  USER db2admin
  USING password;

CREATE WRAPPER informix;

CREATE SERVER ids_iisdn
  TYPE informix VERSION 9.50
  WRAPPER informix
  OPTIONS (node 'rhhost_soc',
    DBNAME 'ids_iisdn');

CREATE USER MAPPING FOR db2admin
  SERVER ids_iisdn
  OPTIONS
  (
  REMOTE_AUTHID 'informix',
  REMOTE_PASSWORD 'password'
  );


SET PASSTHRU ids_iisdn;
```



```
SELECT * FROM systables;

SET PASSTHRU RESET;


CREATE NICKNAME IDS_DailyData_200907
  FOR ids_iisdn."root"."dailydata_200907";
```

A code review related to Example 30-2 contains;

- We run all of these examples through either the DB2 Command Editor (db2ce) or the DB2 Command Line Processor (db2).
- The UPDATE DB CFG statement basically enables the query rewrite capability to consider using MQT tables.

  This line will make this change globally. Alternately, you could set this condition SQL session be session with the command,

  SET CURRENT REFRESH AGE ANY;

  Making this change to the DB CFG requires that we restart at least our given database; as demonstrated in the next 2 lines, FORCE APPLICATIONS and DEACTIVATE.

  The named database will restart automatically when we access it next.
- The CONNECT statement connects us to our database.
- CREATE WRAPPER, CREATE SERVER, and CREATE USER MAPPING create the federated objects detailed above.
- The 2 SET PASSTHRU statements and the SQL SELECT statement located between them are not required.

  We use this statement group merely as a test.

  A PASSTHRU allows us to speak natively and directly with our remote (database) SERVER.
- And the CREATE NICKNAME creates our federated Nickname object.

  At this point we could execute a,

  SELECT * FROM IDS_DailyData_200907;

  and it would return the content of this remote table.

  We could also run two-phase distributed transactions between this database, the remote database, and any other Federated Database Servers.

Note: Informix does have one odd behavior. Even though we created our table inside Informix as, DailyData_200907, the table name inside the Informix systems catalog is recorded as dailydata_200907 (all lowercase).

If we don't double quote the identifiers in the CREATE NICKNAME statement, Federation Server will fold them into all upper case; also not useful.

If all we were demonstrating were federation, we'd be done at this point; run a few queries, a few distributed transactions with update against numerous remote servers, etcetera. But we wish to go further. Next we delve into MQTs.

3. In Federation Server, or as is our actual case, in DB2/UDB, create the following;

   A code review follows Example 30-3.

Example 30-3 Making MQTs, which reference our remote, federated table.

```
CONNECT TO cust_db
  USER db2admin
  USING password;


DROP TABLE local_DailyData_200907;

CREATE TABLE local_DailyData_200907
  AS
  (
  SELECT pk_col, sale_item,
    sale_date, sale_units
  FROM IDS_DailyData_200907
  )
  DATA INITIALLY DEFERRED
  REFRESH DEFERRED
  MAINTAINED BY SYSTEM
  ENABLE QUERY OPTIMIZATION;

REFRESH TABLE local_DailyData_200907;

CREATE INDEX i1
  ON local_DailyData_200907
  (sale_date);

RUNSTATS ON TABLE
  db2admin.local_DailyData_200907
```



```
  AND INDEXES ALL;


DROP TABLE agg_DailyData_200907;

CREATE TABLE agg_DailyData_200907
  AS
  (
  SELECT sale_date, sale_item,
    SUM(sale_units) sum_units
  FROM IDS_DailyData_200907
  GROUP BY sale_date, sale_item
  )
  DATA INITIALLY DEFERRED
  REFRESH DEFERRED
  MAINTAINED BY SYSTEM
  ENABLE QUERY OPTIMIZATION;

REFRESH TABLE agg_DailyData_200907;

RUNSTATS ON TABLE
  db2admin.agg_DailyData_200907
  AND INDEXES ALL;
```

A code review related to Example 30-3 contains;

– The first CREATE TABLE statement makes our first Materialized Query Table (MQT).

  This table is not loaded until the REFRESH TABLE statement that follows its creation. There are many strategies for complete reloads, incremental reload, etcetera. That topic is not expanded upon here.

  We also make an index on this table, and perform RUNSTATS on it, so that the local query optimizer may best judge its use.

Note: This first MQT table is an exact copy of its source. If the source (remote) table or server were down or otherwise unavailable, we could still execute queries, by accessing this MQT table.

As we will see below, a direct and unfiltered (no WHERE clause) query against the source table will no cause (allow for) query rewrite; the query will prefer to run against the source, remote table. If we add WHERE clauses or the like, however, cool things start to happen.

– The second CREATE TABLE makes our second MQT table; this one with aggregation.

4. Testing for query rewrite, part-1.

   If you invoke the DB2 Command Editor (db2ce), you can connect to databases, run queries, and view a query optimizer visual explain. Example as shown in Figure 30-3.

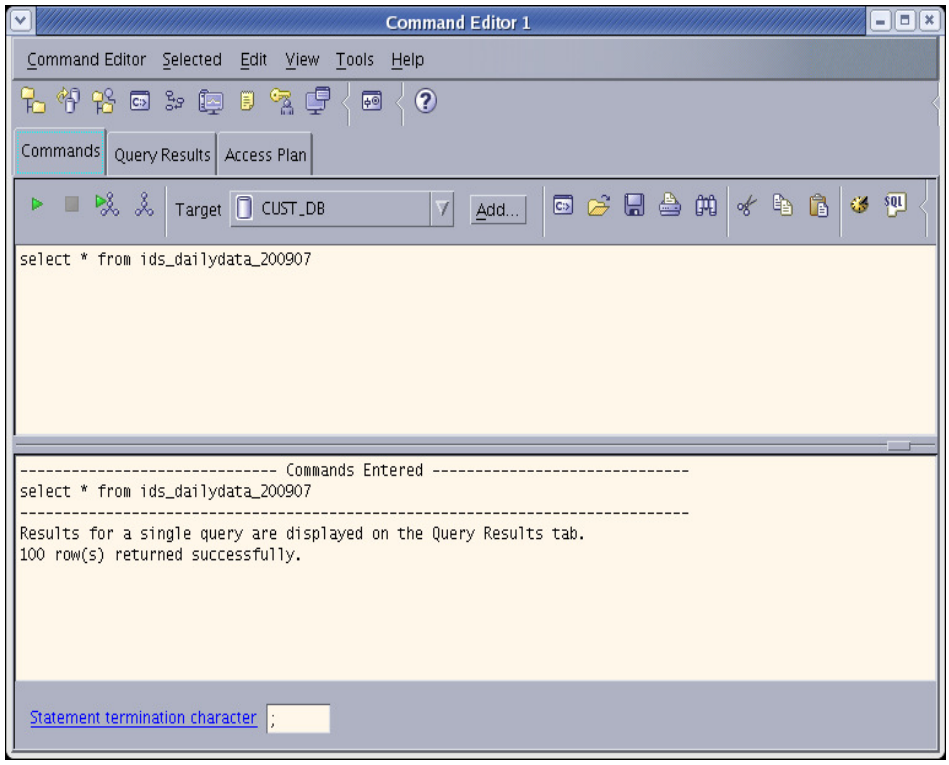

Figure 30-3 Running a query in DB2 Command Editor (db2ce).

   To the right of the green arrow for Run, are other buttons for Run and Explain, or just Explain. Examples as shown in Figure 30-4 and Figure 30-5.

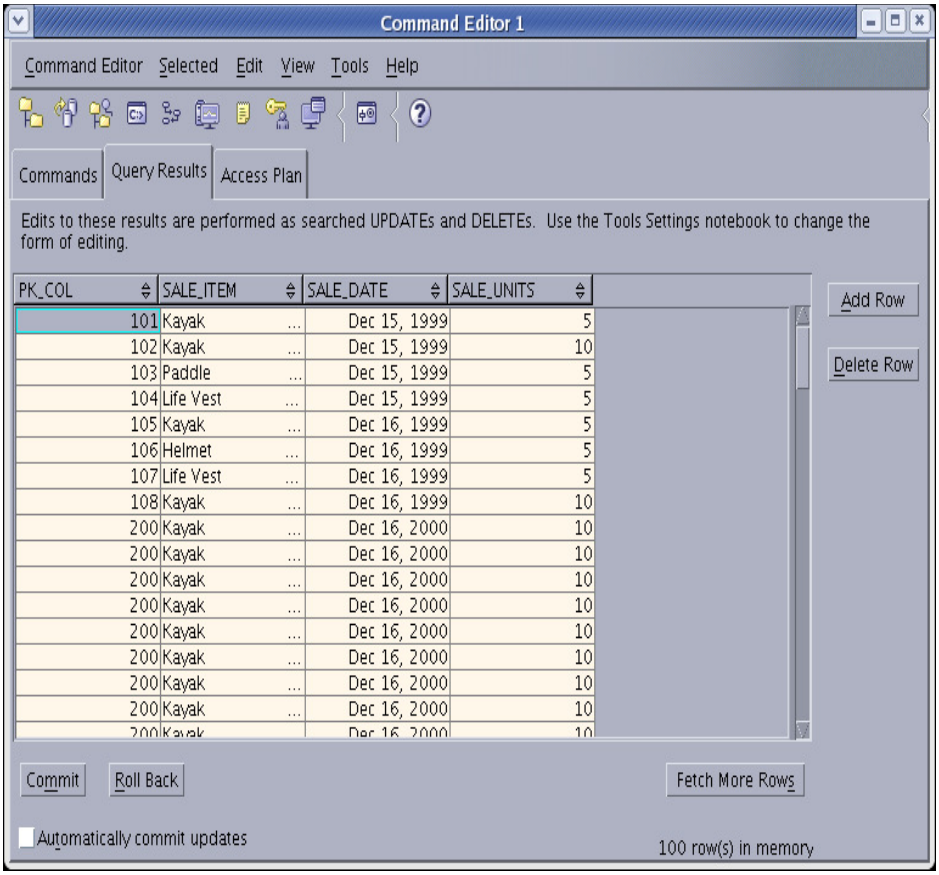

Figure 30-4 Viewing query results set.

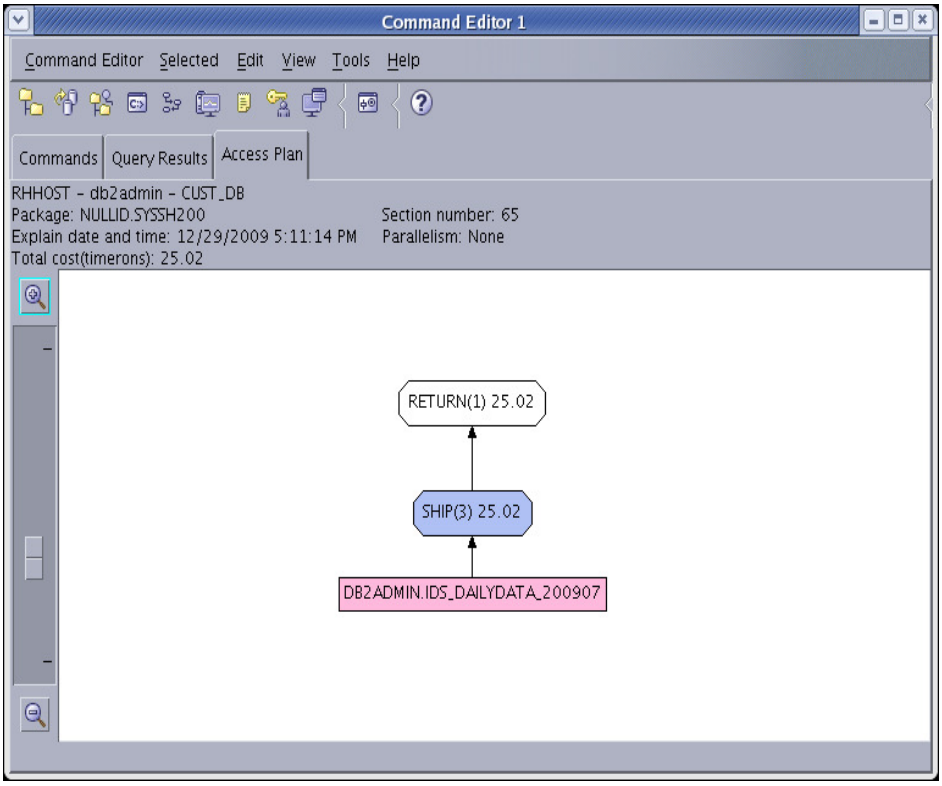

Figure 30-5 Seeing Visual Explain.

Note: We were running DB2/UDB version 9.1.0.3, (run db2level to get your exact version). This version had bug report JR-27405, where a SQL SELECT with a WHERE clause did not accurately present the graphical query plan.

For that reason, and also for those who prefer command line utilities, we detail how to get query plans via the command line.

Run the above first, however, so that the query plan tables get created automatically.

5. Test for query rewrite, part 2.

   In the DB2 Command Editor (db2ce), we are going to run 2 SQL SELECT statements, 1 at a time. And we are going to place statements before and

after these 2 SELECTS, and run an additional command to get the textual query plan.

The 2 SQL SELECT statements are,

```
SELECT *
FROM ids_dailydata_200907
WHERE YEAR(sale_date) = 1999;
```

and,

```
SELECT sale_item, sale_date,
  sum(sale_units) sum_units
FROM ids_dailydata_200907
GROUP BY sale_item, sale_date;
```

Note: The SET CURRENT EXPLAIN MODE EXPLAIN command allows us to analyze a query plan without actually running it. this can save a lot of time when dealing with large data sets.

In each case, we wish to place statements around the SELECT to connect to the database and generate the query plan data. Example as shown,

```
CONNECT TO cust_db
  USER db2admin
  USING password;

SET CURRENT EXPLAIN MODE EXPLAIN;

SELECT *
FROM ids_dailydata_200907
WHERE YEAR(sale_date) = 1999;

SET CURRENT EXPLAIN MODE NO;
```

After each SELECT, we need to run the following diagnostic command. Example as shown,

```
db2exfmt -d cust_db -g TIC -w -1 -n % -s % -# 0 -o MyFile.txt
```

(That's a zero, follows my a small 'o', above.)

Below are the sample query plans, with lines deleted. A code review follows.

Example 30-4 Sample query plan for first query, with lines deleted.

```
Original Statement:
------------------
SELECT *
FROM ids_dailydata_200907
WHERE YEAR(sale_date) = 1999


Optimized Statement:
-------------------
SELECT Q1.PK_COL AS "PK_COL",
  Q1.SALE_ITEM AS "SALE_ITEM",
  Q1.SALE_DATE AS "SALE_DATE",
  Q1.SALE_UNITS AS "SALE_UNITS"
FROM DB2ADMIN.IDS_DAILYDATA_200907
  AS Q1
WHERE ('01/01/1999' <= Q1.SALE_DATE)
  AND (Q1.SALE_DATE <= '12/31/1999')

Access Plan:
-----------
    Total Cost: 21.9488
    Query Degree:1

          Rows
         RETURN
         (   1)
          Cost
           I/O
           |
           0.8
         FETCH
         (   2)
         21.9488
           2.9
       /-----+-----\
     0.8           9008
   IXSCAN      TABLE: DB2ADMIN
   (   3)   LOCAL_DAILYDATA_200907
   15.8973
    2.1
    |
   9008
 INDEX: DB2ADMIN
    I1
```



```
Extended Diagnostic Information:
--------------------------------

Diagnostic Identifier: 1
Diagnostic Details:
  EXP0062W  The following MQT or statistical
  view was not eligible because one or more
  columns or expressions referenced in the
  query were not found in the MQT:
  "DB2ADMIN"."AGG_DAILYDATA_200907".

Diagnostic Identifier: 2
Diagnostic Details:
  EXP0148W  The following MQT or statistical
  view was considered in query matching:
  "DB2ADMIN"."AGG_DAILYDATA_200907".

Diagnostic Identifier: 3
Diagnostic Details:
  EXP0148W  The following MQT or statistical
  view was considered in query matching:
  "DB2ADMIN"."LOCAL_DAILYDATA_200907".

Diagnostic Identifier: 4
Diagnostic Details:
  EXP0149W  The following MQT was used
  (from those considered) in query matching:
  "DB2ADMIN"."LOCAL_DAILYDATA_200907".


Plan Details:
-------------
{lines deleted}
```

A textual representation of the query graph gets lost in a cut and paste, here is a screen print of that same graph, Figure 30-6.

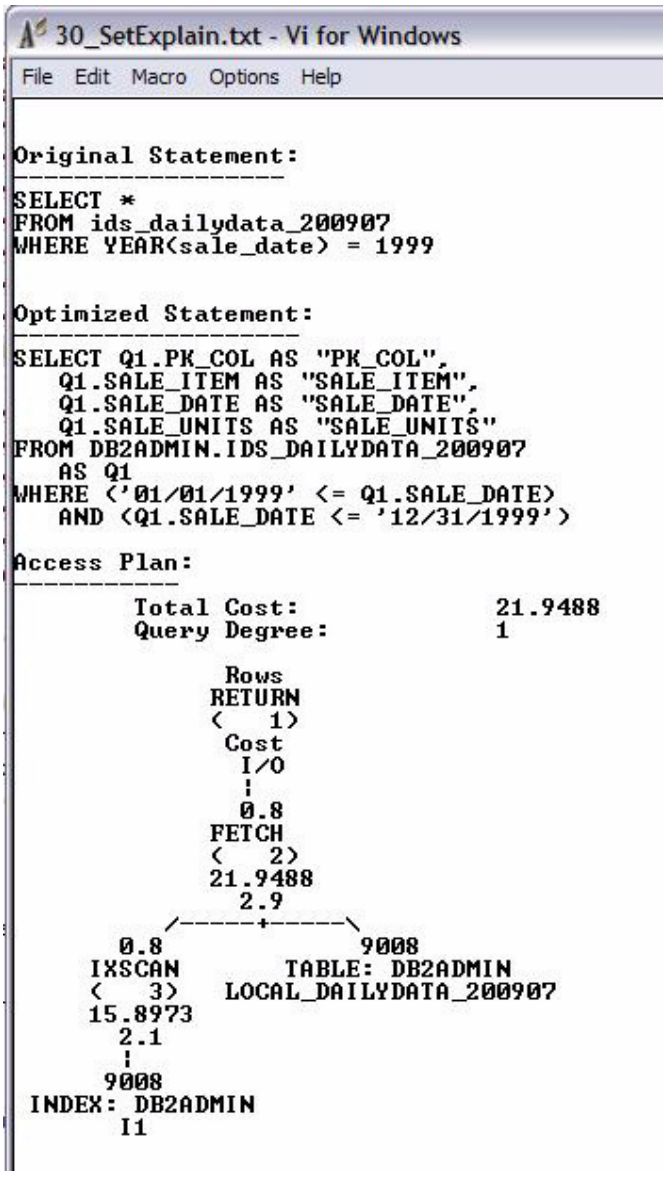

Figure 30-6 Screen print of query graph.

From the image and text above, we see that the query has been rewritten, to include;

– The original source query referenced only the one, remote table; IDS_DailyData_200907.

Query re-write calls to use the index, db2admin.i1, which was placed on the local table, local_DailyData_200907.

(Indexes don't have table names as they may be placed on one or more tables.)

– After an indexed retrieval of the key from the local index structure for efficiency, we then go to the remote table to get the freshest data possible, returning only the few number of rows which met our query criteria.

How cool is that?

Recall too that out source table had 1000 or more rows, nearly all of which from year 2000. Our query criteria called for the few number of rows found to exist for 1999.

And, we used an expression on that index, YEAR(col), not just a direct reference to the index column value.

How cool is that?

Note: The second query is both easier and harder. There is an MQT which more closely resembles the source (original) query, but the query rewrite capability also has to recognize that.

The second query is detailed in the image below, Figure 30-7.

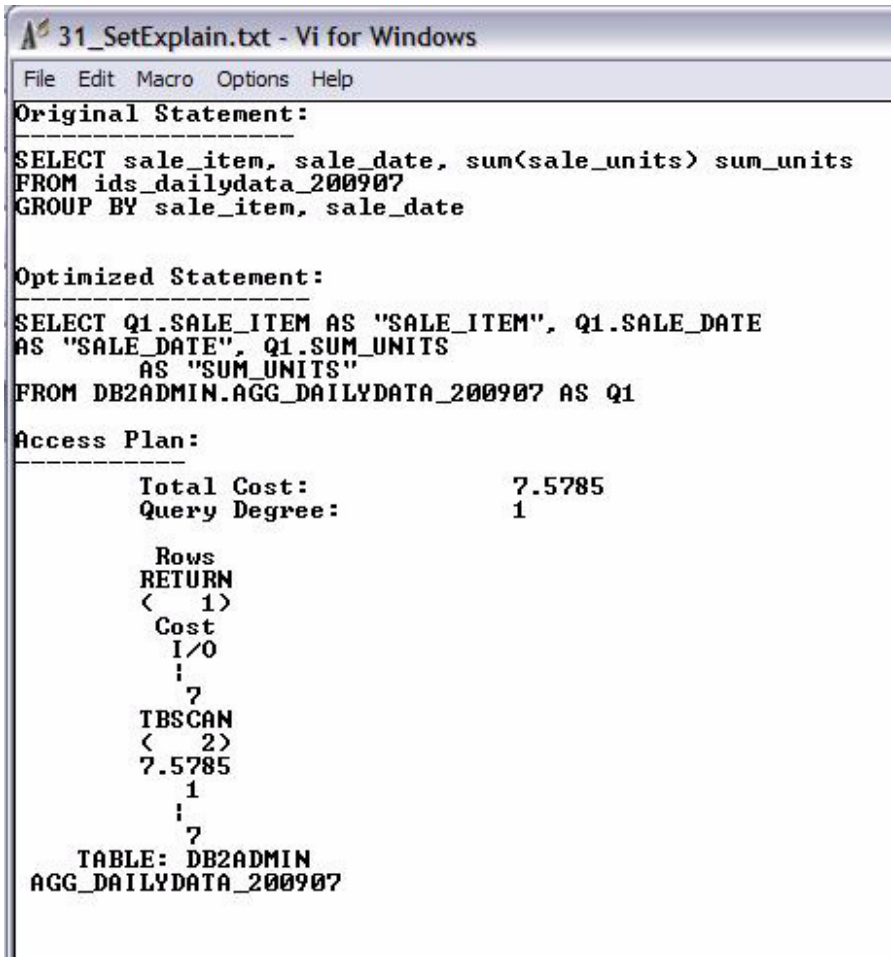

Figure 30-7 Query plan from second query.

In Figure 30-7, we see a simpler replacement of the source table with the MQT table; the MQT table being less cost and meeting all of our needs.

30.3 In this document, we reviewed or created:

We detailed the how to create and configure IBM InfoSphere Federated Server (IFS) objects, including a review of the IFS object hierarchy. And we detailed Materialized Query Tables (MQTs), and demonstrated query rewrite on a number of queries.

Persons who help this month.

Mr.Federated, Sean Byrd.

Additional resources:

Legal statements:

Special attributions:

The listed trademarks of the following companies require marking and attribution:

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel trademark information

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Other company, product, or service names may be trademarks or service marks of others.

