

Improve performance of mixed OLTP workloads with DB2 shadow tables

Real-time analytics and enhanced insight

Ayesha Zaka (ayeshaz@ca.ibm.com)

Software Engineer
IBM

25 September 2014

DB2® with BLU Acceleration has an innovative new feature to make analytics faster and simpler. In this article, learn how shadow tables use BLU Acceleration technologies to improve the performance of analytic queries within your OLTP environment. Experience the power of complex reporting on real-time data in a single database. Examples walk you through the simple steps to set up your environment to start using shadow tables and tap into their power.

Growing need for OLTP systems

Business organizations have a growing need to bring disparate systems for transactional and analytic processing together for enhanced reporting on real-time operational data. The varied performance and storage requirements between online transaction processing (OLTP) and online analytical processing (OLAP) systems make it difficult to have a single system that performs well on both fronts. There is a need for *online transactional analytic processing (OLTP)* systems that can efficiently perform complex analytics on real-time transactional data with minimal trade-off.

[Download a trial of IBM DB2 with BLU Acceleration](#) (Includes all InfoSphere CDC software components required for shadow tables)

DB2 with BLU Acceleration has transformed complex analytic processing. The introduction of in-memory, CPU, and I/O optimized column-organized processing has drastically enhanced analytic workloads. DB2 introduced *shadow tables*, which bring these technologies together for hybrid OLTP processing of your data. Instead of offloading transactional data into a separate OLAP system for further data crunching, systems that process a higher percentage of OLTP workloads than OLAP workloads can now use shadow tables to leverage the columnar-processing power of DB2 BLU alongside their row-organized data.

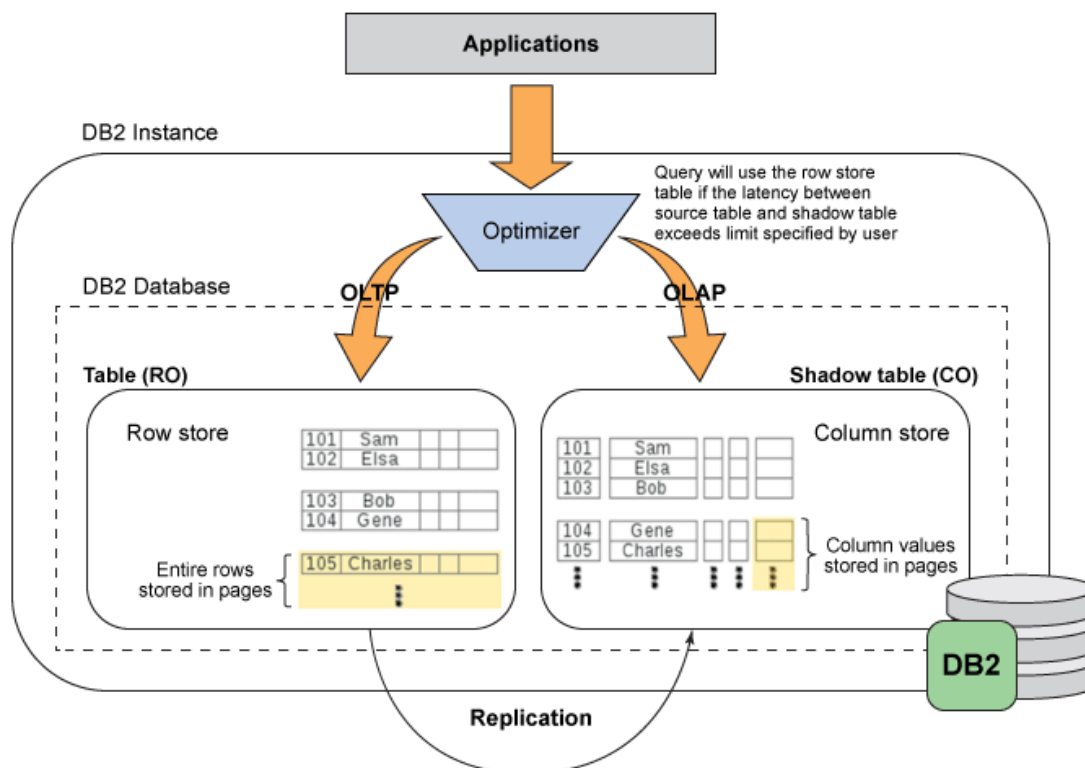
Shadow tables make OLTP simpler

The shadow table feature takes advantage of BLU Acceleration technologies to eliminate the overhead involved in the traditional handling of OLTP workloads, making processing faster,

simpler, and easier. Shadow tables eliminate the need for multiple indices to support OLAP or mixed workloads in a predominantly OLTP system.

Shadow tables are up-to-date column organized copies of their row-organized source tables. Shadow tables can "shadow" existing row-organized (source) tables by their associated column-organized tables, allowing the DB2 optimizer to choose the optimal execution environment for both OLAP and OLTP queries. The copy is maintained by replication without any affect on how the data is queried by applications. Applications continue to query the source table, as always, without any changes to their query structures. Queries identified as benefiting from column organization are transparently rerouted to run against shadow tables, ensuring that incoming queries are always executed using the best execution environment regardless of their nature (OLTP vs. OLAP). Figure 1 shows a high-level view of DB2 optimizer routing queries, based on their nature and execution plan, to row-organized tables and their shadow table counterparts.

Figure 1. Overview of shadow tables



OLAP queries issued against typical row-organized tables are automatically rerouted to shadow tables to take advantage of BLU Acceleration. The DB2 optimizer can base query routing decisions on the latency between the source table and the shadow table, as in the user-defined limit, resulting in a flexible environment optimized for the mixed OTLAP workload.

Shadow table architecture

Shadow tables are implemented within DB2 as a form of MQTs maintained by continuous replication using InfoSphere® Data Replication Change Data Capture (CDC). The replication

solution comprises InfoSphere CDC with DB2 to ensure that the shadow tables are up to date with their associated source tables. The shadow table feature uses InfoSphere CDC 10.2.1.

InfoSphere CDC is responsible for capturing changes to the source tables and applying those changes to the associated columnar tables within the same database system. Changes to the source tables are captured by reading the database transaction log stream. InfoSphere CDC is also responsible for feeding latency information back to DB2. The information shared by InfoSphere CDC is used by the DB2 optimizer when making routing decisions based on how current the shadow table data is.

Figure 2. Architecture for DB2 shadow table replication

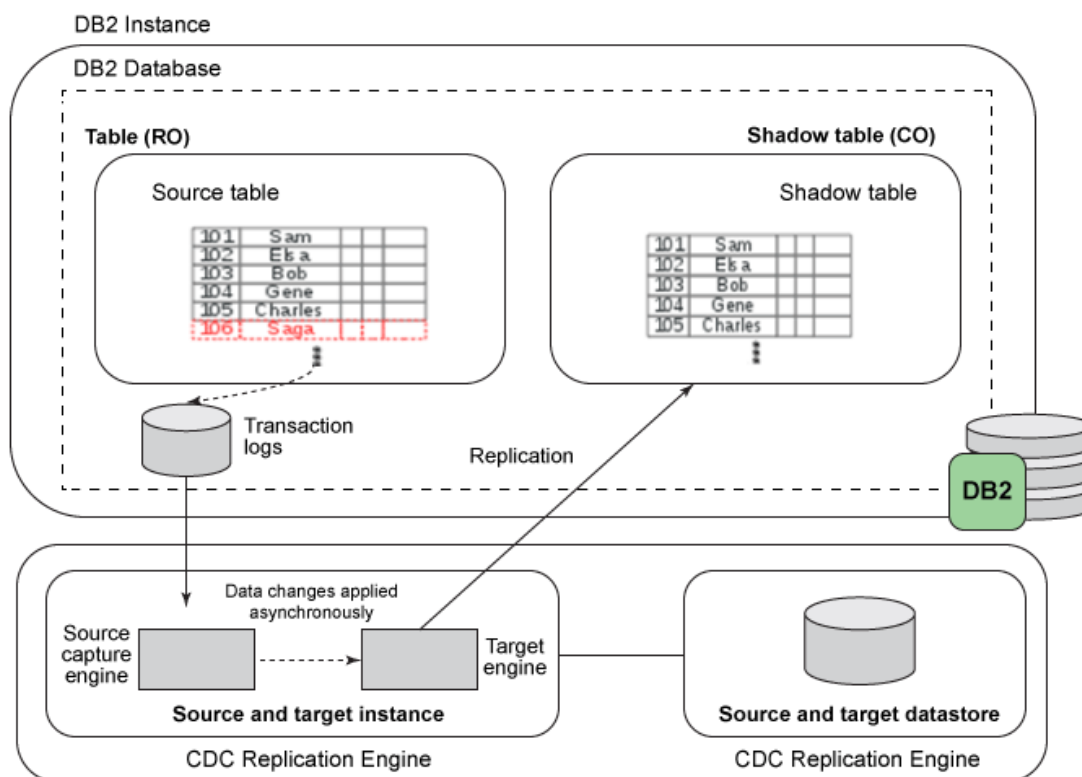


Figure 2 shows the complete replication solution in more detail. It is composed of a single CDC instance and DB2 instance. The InfoSphere CDC Access Server maintains the communication and metadata used by the InfoSphere CDC Replication Engine for DB2 Linux®, UNIX®, and Windows® (LUW). The database transaction logs are read by the InfoSphere CDC Replication Engine to identify data manipulation (DML) changes to the source tables that are to be applied by the apply agent to the target shadow tables.

Setting up shadow tables

This section covers the high-level tasks to set up the complete replication solution for shadow tables and the configuration process for using InfoSphere CDC to maintain and administer shadow tables.

Deployment roadmap

To successfully deploy the replication solution and begin using shadow tables, use the following required high-level steps:

1. [Install DB2 10.5 Fix Pack 4 as root user](#), then:
 - a. Create a DB2 instance.
 - b. [Create and configure the database for shadow table use](#).
2. [Install InfoSphere CDC and configure replication for shadow tables](#):
 - a. Install InfoSphere CDC Access Server.
 - b. Install InfoSphere CDC for DB2 for LUW 10.2.1.
 - c. Install the InfoSphere CDC Management Console (optional). The CDC instance can be managed via the command-line interface CHCCLP or by using the graphical CDC Management Console on the Windows platform.
 - d. Create the InfoSphere CDC Replication Engine instance — one CDC instance per database.
 - e. Configure a datastore.
 - f. Create the SYSTOOLS.REPL_MQT_LATENCY table.
3. [Create shadow tables](#).
4. [Configure and begin replication to shadow tables](#):
 - a. Create an InfoSphere CDC subscription and table mappings.
 - b. Start the InfoSphere CDC subscription and replicate data to a shadow table.
5. [Enable query routing to shadow tables](#):
 - a. Enable query routing within a DB2 command-line processor (CLP) session.
 - b. Enable query routing at the application level.

Product installation

To use the shadow tables feature, DB2 must be installed as root and the database must be configured for shadow table use. Two important user accounts required for shadow tables: the user for the DB2 instance where replication is to be carried out (for example, *db2inst1*), and a user for the CDC installation with DATAACCESS and DBADM or SYSADM authority on DB2 (for example, *cdc-user*). The CDC user should have access to the db2-instance-dir/db2-inst/name/NODE0000/sqlldbidir directory with drwxrwxr-x permissions.

To support shadow tables, you must install the following software versions using their supported platforms:

- IBM DB2 for LUW Cancun Release 10.5.0.4. Supported platforms are listed in the DB2 LUW Knowledge Center.
- IBM InfoSphere Data Replication 10.2.1 — Change Data Capture for DB2 LUW with interim Fix 12 and later releases. Supported for platforms that run DB2 10.5 FP4 for LUW.
- IBM InfoSphere CDC Access Server 10.2.1 Interim Fix 5 and later releases.
- IBM InfoSphere Data Replication 10.2.1 — CDC Management Console Interix Fix 5 and later releases (optional).

The components for the replication solution can be installed within the same system or on separate systems. The suggested layout for this tutorial consists of InfoSphere CDC replication

and InfoSphere CDC Access Server installed on the same server as the DB2 instance, and InfoSphere CDC Management Console installed on a dedicated Windows server or workstation.

Configuring DB2 for shadow tables

When using shadow tables, the database is not configured for BLU Acceleration because the system is to be used for a mixed OLTP workload. The BLU workload setting `DB2_WORKLOAD` to `ANALYTICS`, while good for OLAP, is not optimal for OLTP workloads. For shadow tables, use database manager and database configuration parameter settings to exploit BLU Accelerations.

This article highlights the following important configuration parameters. ([DB2 server configuration for shadow tables](#) has a complete list of configuration settings.)

Configuration parameter	Value	Example
DB2_EXTENDED_OPTIMIZATION	<code>OPT_SORTHEAP_EXCEPT_COLUMN</code>	<pre>\$ db2 get db cfg for oltpdb grep -i sortheap Sort list heap (4KB) (SORTHEAP) = 10000 \$ db2set DB2_EXTENDED_OPTIMIZATION="OPT_SORTHEAP_EXCEPT_COLUMN 10000"</pre>

Reason: Environments with shadow tables have higher sort heap memory requirements than databases in OLTP environments. To increase the sort heap memory without affecting existing OLTP queries, specify the override value for queries that do not reference shadow tables. The existing value of the `sortheap` database configuration parameter can be used as the override value.

Configuration parameter	Value	Example
Sortheap and sheapthres_shr	<code>NOT AUTOMATIC</code>	<pre>\$ db2 update db cfg for oltpdb using sheapthres_shr 12000000 sortheap 600000</pre>

Reason: Shadow tables require more sort heap memory than OLTP environments. To ensure that there is memory for the CDC components, assign the following suggested memory amounts to the two parameters:

- Set `sheapthres_shr` to 50 percent of the database memory.
- Set `sortheap` to 5-20 percent of `sheapthres_shr`.

For example, 50 percent of 96 GB is 48 GB or 12 million 4K pages, and 5 percent of 48 GB is 2.4 GB or 600,000 4K pages.

Configuration parameter	Value	Example
logarchmeth1	<code>LOGRETAIN</code> or similar (not <code>NONE</code>)	<pre>\$ db2 update db cfg for oltpdb using logarchmeth1 logretain</pre>

Reason: InfoSphere CDC uses transaction logs for data replication and to read from non-active logs, so the database transaction logs need to be retained.

Configuration parameter	Value	Example
<code>util_heap_sz</code>	AUTOMATIC (with appropriate starting value)	\$ db2 update db cfg for oltpdb using util_heap_sz 1000000 AUTOMATIC

Reason: The `LOAD` command is used for the refresh operation on shadow tables. Set the `util_heap_sz` database configuration parameter to Automatic with a larger starting value to address the requirements of the `LOAD` command usage by InfoSphere CDC. A good starting value is 1 million 4K pages. To accommodate higher memory requirements, such as when running concurrent workloads of the `LOAD` command, you will need to increase the `util_heap_sz` parameter.

Configuring CDC components for shadow tables

Before moving on to configuration of the InfoSphere CDC components, you might want to review some CDC concepts that are important in the context of shadow tables, as listed in Table 1.

Table 1. InfoSphere CDC terminology

Term	Definition
InfoSphere CDC instance	An instance of the replication engine. For shadow table replication, create only one single CDC instance for all shadow tables in a database.
Datastore	An abstraction that represents a CDC instance. It holds metadata about the database and data files that are required for replication. For shadow tables, we only require one datastore because the source and target are the same database.
Subscriptions	A container for table mappings that controls replication behavior. For shadow tables, replication requires that you create one single persistent subscription that provides fault tolerant replication for all shadow tables in the database.
Table mappings	Contain information on how individual tables (or a subset of columns of tables) are replicated from the source to the target datastores. Shadow tables use standard replication with a 1:1 table mapping between a source (row-organized) table and the shadow table.
Mirror	The process of replicating changed data from the source table to a target table. The replication method for shadow tables is continuous mirroring, which continuously replicates changes to shadow tables in your subscription.
Refresh	The process that synchronizes the shadow table with the current contents of the source table. The initial CDC refresh will load data into the shadow table.
Latency	Query routing relies heavily on this CDC concept. Latency signifies how synchronized the shadow table is with its associated source table.
CHCCLP command-line interface	Command-line tool used to administer and monitor CDC subscriptions and replication. It can be run in batch or interactive mode.

The next section discusses configuring InfoSphere CDC, which consists of setting up the InfoSphere CDC Access Server and creating and setting up InfoSphere CDC objects that are required for shadow tables.

Create CDC instance and configure CDC Replication

Use the CDC Replication Engine `dmconfigurets` command to launch the interactive instance configuration tool. Create a new instance of the CDC Replication Engine using the values in Table 2. The configuration tool is in the `/bin` subfolder for the CDC replication installation folder, as the `cdc-user`.

```
cd [...]/InfoSphereChangeDataCapture/ReplicationEngineforDB2/bin/
./dmconfigurets
```

Table 2. CDC instance parameters

Parameter	Value example	Description
Name	<code>cdcinst1</code>	Instance name that you want to create.
Staging Store Disk Quota (GB)	<code>variable</code>	Staging storage on the source.
DB2 instance	<code>db2inst1</code>	DB2 instance involved in replication.
Database name	<code>oltpdb</code>	Database with tables to replicate.
Username	<code>db2inst1</code>	DB2 username for the specified database.
Password	<code><password to db2inst1></code>	Database password.
Metadata schema	<code>db2inst1</code>	Schema name used by the CDC for DB2 for metadata table.
Refresh loader path	<code>/db2/db2inst1_db1/ cdc_refresh_loader</code>	The file path <code>cdc_refresh_loader</code> is later used as a staging location for table refreshes with CDC data replication.

After the instance is set up, use the `dmts64` command to configure the latency settings for the instance and activate the instance for replication.

```
cd [...]/InfoSphereChangeDataCapture/ReplicationEngineforDB2/bin/
dmset -I cdcinst1 maintain_replication_mqt_latency_table=true
dmset -I cdcinst1 acceptable_latency_in_seconds_for_column_organized_tables=50
dmset -I cdcinst1 mirror_auto_restart_interval_minutes=2
nohup ./dmts64 -I cdcinst1 &
```

These parameters control the latency information communicated to DB2 by the CDC replication solution. The parameters signify that:

- The latency information is to be sent to the database table `SYSTOOLS.REPL_MQT_LATENCY`.
- The acceptable latency for changes to be applied to shadow tables from source tables is to be set to 50s.

The `mirror_auto_restart_interval_minutes` parameter sets the delay after which persistent subscriptions are restarted in the case of replication shutdown or restart.

For the latency information to be referenced by the DB2 optimizer, the database table storing this information has to be built manually. The following example shows the stored procedure creating the table within table space `TBSP4K`:

```
CALL SYSPROC.SYSINSTALLOBJECTS('REPL_MQT', 'C', 'TBSP4K', CAST (NULL AS VARCHAR(128)))
```

Create CDC Access Server user and datastore

With the CDC instance created and configured, you need to create the administration user for CDC Access Server and specify the user as the SYSADMIN using the `dmcreateuser` command. The commands are accessible as the *cdc-user* from the bin subfolder of your CDC Access Server installation:

```
cd [...]InfoSphereChangeDataCapture/AccessServer/bin/
```

The following commands show how to create the CDC administration user, the datastore, and the connection to associate it with your database.

Table 3. Setting up the administration user and datastore

Create administration user for CDC Access Server	<pre>./dmcreateuser <user_name><full_name><description><password><role><manager>< For example: ./dmcreateuser cdc-admin "N/A" "administrator" p@ssw0rd sysadmin true false false</pre>
Create datastore	<pre>dmcreatedatastore <datastore-name> "description" hostname for database port for datastore For example: ./dmcreatedatastore ds1 "db2inst1 OLTPDB Shadow Tables" prodhost 10901</pre>
Add connection	<pre>dmaddconnection <cdc-username> <datastore-name> <db-name> <db2-user> <db-password> <alwaysPrompt> <showParams> <writeProtected> For example: ./dmaddconnection cdc-admin ds1 oltpdb db2inst1 pa\$\$w0rd false true false true</pre>

For more detail about these commands, see the InfoSphere CDC documentation in [Resources](#).

Creating shadow tables

Now that the InfoSphere CDC components are set up and configured, you are ready to create and set up shadow tables. DB2 provides tooling to assist in selecting an appropriate source table. You can use the Optim™ Query Workload Tuner to provide recommendations of valid candidates for shadow tables. When creating a shadow table, it is important to make sure that the selected source table has a primary or unique key constraint defined. A 1:1 mapping is required for each row in both tables.

To create a shadow table, issue the `CREATE TABLE` statement with the `MAINTAINED BY REPLICATION` clause. This clause identifies the table as a shadow table. The source table can be replicated in its entirety, or a subset of columns relevant to your query workload can be selected for replication. Our example below shows the shadow table being created from source table `TRADE`. Make sure that the primary key of the source table is part of the select list.

```
create table TRADE_SHADOW as
(select * from TRADE)
data initially deferred
refresh deferred
enable query optimization
maintained by replication
organize by column;
```


Upon creation, the new shadow table is in *set integrity pending* state. Issue a command to bring the shadow table out of pending state. Alter the table to add a primary key constraint; the shadow table requires a primary key that matches the enforced primary/unique constraint on the source table. Issue the following statements on the shadow table:

```
set integrity for TRADE_SHADOW all immediate unchecked;  
alter table TRADE_SHADOW add constraint TRADE_SHADOW_PK primary key (trans_id);
```

The shadow tables are now set up. You have successfully installed and configured replication for shadow tables and created a shadow table. The next section discusses how to set up the subscription and how replication is performed.

Setting up replication for the shadow table

When you use the shadow table feature, all table mappings are grouped under a single subscription for the database. Users can create subscriptions using the graphical CDC Management Console or the command-line driven CHCCLP utility. This article showcases the CHCCLP commands for brevity, but new users are advised to use the CDC Management Console for its ease of use.

The commands below are an example of creating a persistent subscription *sub1* in the datastore *ds1*. To create a subscription, the user has to connect to the access server and datastore associated with the database containing shadow tables.

```
$cd [...]InfoSphereChangeDataCapture/AccessServer/bin/  
$./chcclp  
set verbose;  
connect server hostname prodhost port 10101 username cdc-admin password p@ssw0rd;  
connect datastore name ds1;  
add subscription name sub1 persistency true;
```

With the subscription successfully created, a table mapping is created for each source and shadow table pairing, and mirroring is initiated. While being connected to the datastore, use the following command within the CHCCLP.

```
select subscription name sub1;  
add table mapping sourceSchema BASE sourceTable TRADE targetSchema BASE targetTable TRADE_SHADOW  
targetIndexMode index targetIndexName BASE.TRADE_SHADOW_PK type standard method mirror;  
start mirroring method continuous;  
disconnect datastore name ds1;  
disconnect server;  
exit;
```

The setup and configuration of replication for the newly created shadow table is now complete. The next step would be to verify that the queries are being routed to the correct type of tables based on your settings. To verify that query routing is enabled, you can use DB2 monitoring table functions or the EXPLAIN utility. The [DB2 Knowledge Center](#) has more information about monitoring shadow tables and query routing to shadow tables.

Enabling query routing to shadow tables

To enable the DB2 Optimizer to reroute queries to shadow tables, certain parameters must be set to required values. Along with the replication latency being within the defined limit, `intra_parallel` has to be enabled (set to 'ANY') to allow runtime access to the shadow tables. The following special registers have to be set to allow for use of the latency-based routing:

- `CURRENT_REFRESH_AGE` special register has to be set to a duration other than zero or ANY. Latency-based query routing occurs when the source-shadow latency is within the interval specified by this special register.
- `CURRENT_MAINTAINED_TABLE_TYPES_FOR_OPTIMIZATION` special register is set to contain only REPLICATION.
- The `CURRENT_QUERY_OPTIMIZATION` special register should be set to 2 or a value greater than or equal to 5.

To learn more about these registers, visit the [DB2 Knowledge Center](#).

After you are compliant with the requirements for latency-based routing, you can execute the following SQL statements to enable shadow table support for query routing.

```
$ db2 CONNECT TO oltpdb
$ db2 "CALL ADMIN_SET_INTRA_PARALLEL ('YES')"
```

```
$ db2 SET CURRENT DEGREE 'ANY'
$ db2 SET CURRENT REFRESH AGE 500
$ db2 SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION REPLICATION
$ db2 SET CURRENT QUERY OPTIMIZATION 2
```

Example

The following example demonstrates the conditions under which the DB2 optimizer chooses to route a query to a shadow table instead of the source table. We will use the `TRADE` and `TRADE_SHADOW` table defined earlier. We will also leverage an example of a view that checks on the current latency between the source-shadow pair and checks whether the latency falls within the specified limit. For a description of the latency view, read the "enablement of query routing to shadow tables" in the [DB2 Knowledge Center](#).

Query on the source table

```
SET CURRENT REFRESH AGE 500;
SELECT COUNT(*) AS N_TRADES,
SUM(TRADE_PRICE) AS TRADE_PRICE,
YEAR(TRANS_DATE) AS YEAR
FROM TRADE
GROUP BY YEAR(TRANS_DATE)
ORDER BY YEAR(TRANS_DATE)
```

Sample output from the latency view for the source-shadow pair

```
SELECT * FROM REPL_LATENCY;
```

REFRESH_AGE	LATENCY	CUR_TS	REFRESH_TS	COMMIT_POINT	DELAY_OFFSET
500.000000	3.572204	2014-05-07-14.03.20.572204	2014-05-07-14.03.17.000000	1399485797	0

1 record(s) selected.

The result of the latency view shows that the query should be rerouted to the shadow table. The latency between the source-shadow pair (3.57s) falls well within the user-defined limit of 5 minutes. Upon further verification using the EXPLAIN utility, it is confirmed that the query rerouting was successful and the cost is 77 percent less than the traditional row-based access plan referenced prior to setting up shadow tables. Table 4 shows a snippet of the EXPLAIN output.

Table 4. Access plans for querying with row-based and shadow tables

<pre> Access Plan: ----- Total Cost: 767030 Query Degree: 8 Rows RETURN (1) Cost I/O 12 GRPBY (2) 767030 152935 12 LMTQ . . . 4.28e+06 TABLE: DTW TRADE Q1 ... </pre>	<pre> Access Plan: ----- Total Cost: 175824 Query Degree: 8 Rows RETURN (1) Cost I/O 12 . . . LMTQ Extended Diagnostic Information: ----- Diagnostic Identifier: 1 Diagnostic Details: EXP0148W The following MQT or statistical view was considered in query matching:"DTW".TRADE_SHADOW". Diagnostic Identifier: 2 Diagnostic Details: EXP0149W The following MQT was used (from those considered) in query matching:"DTW".TRADE_SHADOW". </pre>
---	---

There are some administrative changes to database operations regarding the shadow table feature and some operational best practices that help you better administer and tune your system while using shadow tables. You can also monitor CDC replication for your subscription using the CDC Management tool. To learn more about administrative changes, best practices, and using the CDC Management tool, see Chapter 3 in the IBM Redbooks® publication titled "[Architecting and Deploying IBM DB2 with BLU Acceleration in Your Analytical Environment](#)."

Summary

Shadow tables powered by BLU Acceleration technologies are the real enabler for analytics on OLTP workloads. They make the processing of mixed workloads faster, simpler, and easier. By having data in both traditional row-organized format and in column-organized format within the same database, organizations get the combined power of an OLTP and OLAP optimized query execution environment. Begin using shadow tables to experience the improved performance for your mixed OLTP workloads.

Resources

Learn

- For product documentation, visit the [IBM Knowledge Center for DB2 Version 10.5 for Linux, UNIX, and Windows](#).
- The IBM Knowledge Center for InfoSphere CDC has help for:
 - [Installing Access Server](#)
 - [Starting Access Server](#)
- The [IBM Knowledge Center for InfoSphere CDC Replication](#) has the latest documentation.
- Learn about the [InfoSphere CDC Management Console](#).
- Visit the [developerWorks Information Management zone](#) to find more resources for DB2 developers and administrators.
- Stay current with [developerWorks technical events and webcasts](#) focused on a variety of IBM products and IT industry topics.
- Follow [developerWorks on Twitter](#).
- Watch [developerWorks on-demand demos](#) ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.

Get products and technologies

- [Evaluate IBM products](#) in the way that suits you best: Download a product trial, try a product online, or use a product in a cloud environment.

Discuss

- Get involved in the [developerWorks community](#). Connect with other developerWorks users while exploring the developer-driven blogs, forums, groups, and wikis.

About the author

Ayesha Zaka



Ayesha Zaka is a software engineer on the Information Management Data Ecosystem team at the IBM Toronto Lab. She currently works on DB2 with BLU Acceleration and has more than four years of experience conducting technical enablement and consulting for various IM products. She started her career with DB2 9.7. At IBM, she has worked with various IM products, including DB2, InfoSphere BigInsights, InfoSphere Streams, and pureData for Hadoop. She has enabled business partners and clients around the world in adopting IM solutions within their organizations, delivering technical training sessions worldwide and speaking at the annual IBM Information on Demand (now Insight) conference.

© Copyright IBM Corporation 2014

(www.ibm.com/legal/copytrade.shtml)

[Trademarks](#)

(www.ibm.com/developerworks/ibm/trademarks/)