


# **Best practices: Deploying the IBM Banking Data Warehouse to IBM DB2 10.5 with BLU Acceleration**

**Enda McCallig**  
*Data Warehouse Specialist*  
*IBM Dublin Lab*

## Table of Contents

Best practices: Deploying the IBM Banking Data Warehouse to IBM DB2 10.5 with BLU Acceleration.....	1
Introduction.....	4
IBM DB2 10.5 with BLU Acceleration.....	5
The Banking Data Warehouse industry model .....	6
The Dimensional Warehouse Model in the IBM big data reference architecture.....	8
Implementing an industry model as a physical database.....	9
Understanding database design challenges.....	10
Scoping the logical model and transforming into a physical data model. .	11
Scoping the logical model.....	11
Transforming the logical data model into a physical data model.....	14
Preparing the physical data model for deployment as a DB2 BLU database .....	16
Choosing column or row organized tables.....	16
Implementing a table space strategy.....	17
Customizing keys and data types.....	18
Primary keys .....	18
Referential constraints.....	18
Identity keys and surrogate keys.....	19
Refining data types.....	19
Nullable columns.....	20
Populating the physical database.....	20
LOAD utility considerations.....	20
Optimizing load for compression and speed.....	20
Optimizing first-time load into a column-organized table.....	21
Conclusion.....	23
Best practices.....	23
Appendix A. Test environment.....	24
Appendix B. DDL statements.....	25
Further reading.....	29



Contributors.....	29
Notices.....	30
Trademarks.....	31

## Introduction

Implementing industry models can help accelerate projects and reduce risk in a wide variety of industry sectors when enterprises develop and implement business intelligence applications.

The IBM Industry Models solutions cover a range of industries that include banking, healthcare, retail, and telecommunications. The IBM Industry Models solutions provide you with an extensive and extensible data model for your industry sector. Use the logical data model as provided by IBM to build a physical model that is customized for your reporting requirements, and then deploy and populate a best practice IBM DB2® 10.5 with BLU Acceleration database.

This paper introduces the logical data model concepts and then focuses on what you must do to transform a non-vendor-specific logical data model into a best-practice production IBM DB2 10.5 with BLU Acceleration schema.

Specifically, this paper uses the IBM Industry Models Banking Data Warehouse model pertaining to Involved Party and Social Media entities. It guides you through the recommended process for transforming the logical data model for dimensional warehousing into a physical database design for production use in your environment. The key steps described are:

- Create a subset of the data model from the supplied logical data model.
- Prepare the physical data model for deployment as an IBM DB2 10.5 with BLU Acceleration schema.
- Generate the DDL to deploy the schema to DB2.

Using the recommendations in this paper can help you transform an IBM Industry Model Banking Data Warehouse dimensional warehouse solution into a database that is ready for production use.

## IBM DB2 10.5 with BLU Acceleration

BLU Acceleration is a new collection of technologies for analytic queries that are introduced in DB2 for Linux, UNIX, and Windows Version 10.5 (DB2 10.5). At its heart, BLU Acceleration is about providing faster answers to more questions and analyzing more data at a lower cost. DB2 with BLU Acceleration, also known as DB2 BLU, is about providing order-of-magnitude benefits in performance, storage savings, and time to value.

These goals are accomplished by using multiple complementary technologies, including the following technologies:

- *Column-organized tables* mean that I/O is performed only on those columns and values that satisfy a particular query.
- *Actionable compression* on the column data preserves order so that the data can be used without decompression, resulting in huge storage and CPU savings and a significantly higher density of useful data held in memory.
- *Parallel vector processing*, with multi-core parallelism and single instruction, multiple data (SIMD) parallelism, provides improved performance and better utilization of available CPU resources.
- *Data skipping* avoids the unnecessary processing of irrelevant data, thereby further reducing the I/O that is required to complete a query.

These and other technologies combine to provide an in-memory, CPU-optimized, and I/O-optimized solution that is greater than the sum of its parts.

BLU Acceleration is fully integrated into DB2 10.5, so that much of how you leverage DB2 in your analytics environment today still applies when you adopt BLU Acceleration. The simplicity of BLU Acceleration changes how you implement and manage a BLU-accelerated environment. Gone are the days of having to define secondary indexes or aggregates, or having to make SQL or schema changes to achieve adequate performance.

## The Banking Data Warehouse industry model

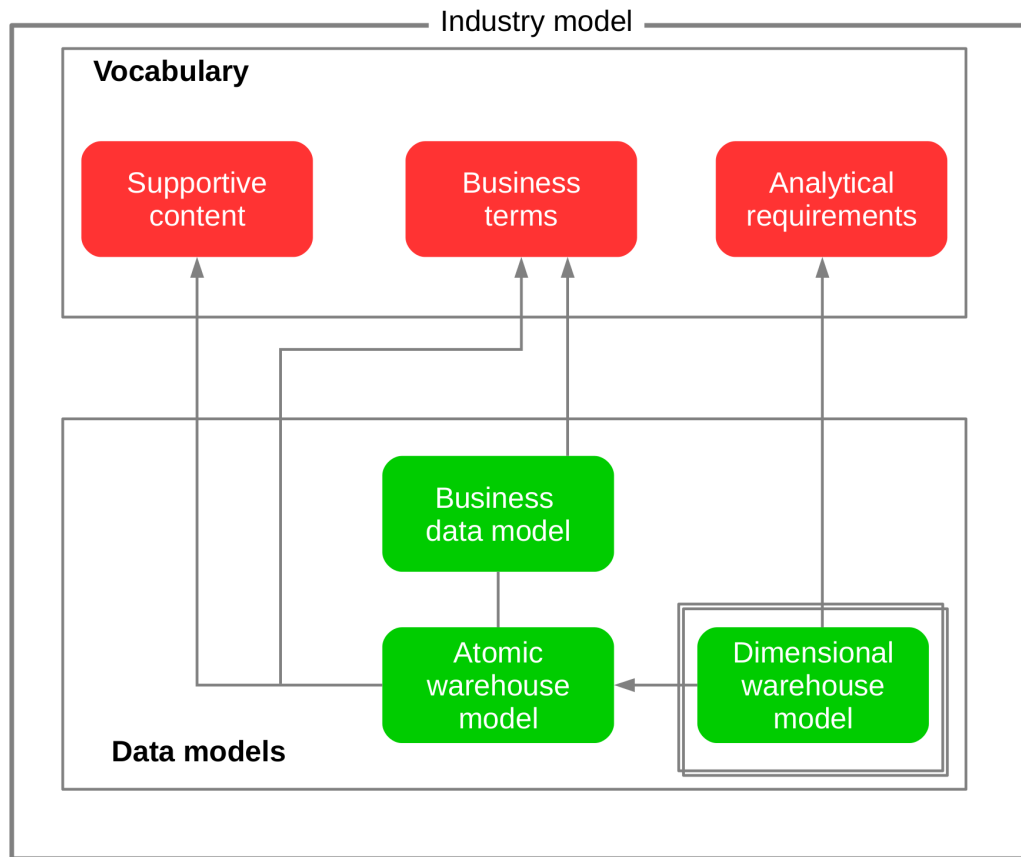
The Banking Data Warehouse (BDW) model is an industry-specific blueprint that provides data warehouse design models, business terminology, and analysis templates to help accelerate the development of business applications. The software can improve reporting, credit and risk management, and consolidate trusted information across multiple viewpoints.

The BDW contains support for social media through the requirements, analysis, and design models. The BDW allows a financial institution to harness social media data into a form that it can use to derive real business insight. This paper uses social media as an example of one of the many types of information in the BDW that can be deployed to IBM InfoSphere® BigInsights™.

The BDW social media content has various business applications. For example, a financial institution might want to calculate various social media metrics such as share of voice and reach, or distinguish between positive, negative, and neutral postings on the company's page. A financial institution might also want to understand what trends and patterns are emerging so that it can make better business decisions.

The model content also allows a financial institution to link a customer and a social media persona, bringing together the traditional sources of data with the new social media data that's available. The financial institution can use this extended view of the customer to identify any concerns early that might affect a customer's creditworthiness. Social persona analysis also allows new relationships to be uncovered and identifies possible cross-selling opportunities that are linked to life events that are uncovered through social media.

Figure 1 illustrates the various components of the BDW. The vocabulary describes the business content that is supported by the data models, and provides a consistent terminology and helps to understand the information used by related IT assets.



*Figure 1. The Dimensional Warehouse Model within the context of industry models*

The business terms are industry concepts in plain business language and with no modeling. The mapping of business terms to the data models allows the transformation of requirements into IT data structures. Analytical requirements are high level groups of business information to express business measures along axes of analysis, which are named dimensions. The analytical requirements are the basis for building the dimensional warehouse model.

The data models are used to build reporting solutions, such as business intelligence and standardized reporting. They are mapped back to the vocabulary and act as a bridge between the business terminology and the deployed IT assets. The data models consist of business, atomic warehouse, and dimensional warehouse models.

Of particular interest to this paper is the dimensional warehouse model, which is described in the next section.

## The Dimensional Warehouse Model in the IBM big data reference architecture

The following figure shows the Dimensional Warehouse Model (DWM) within the context of the IBM big data reference architecture.

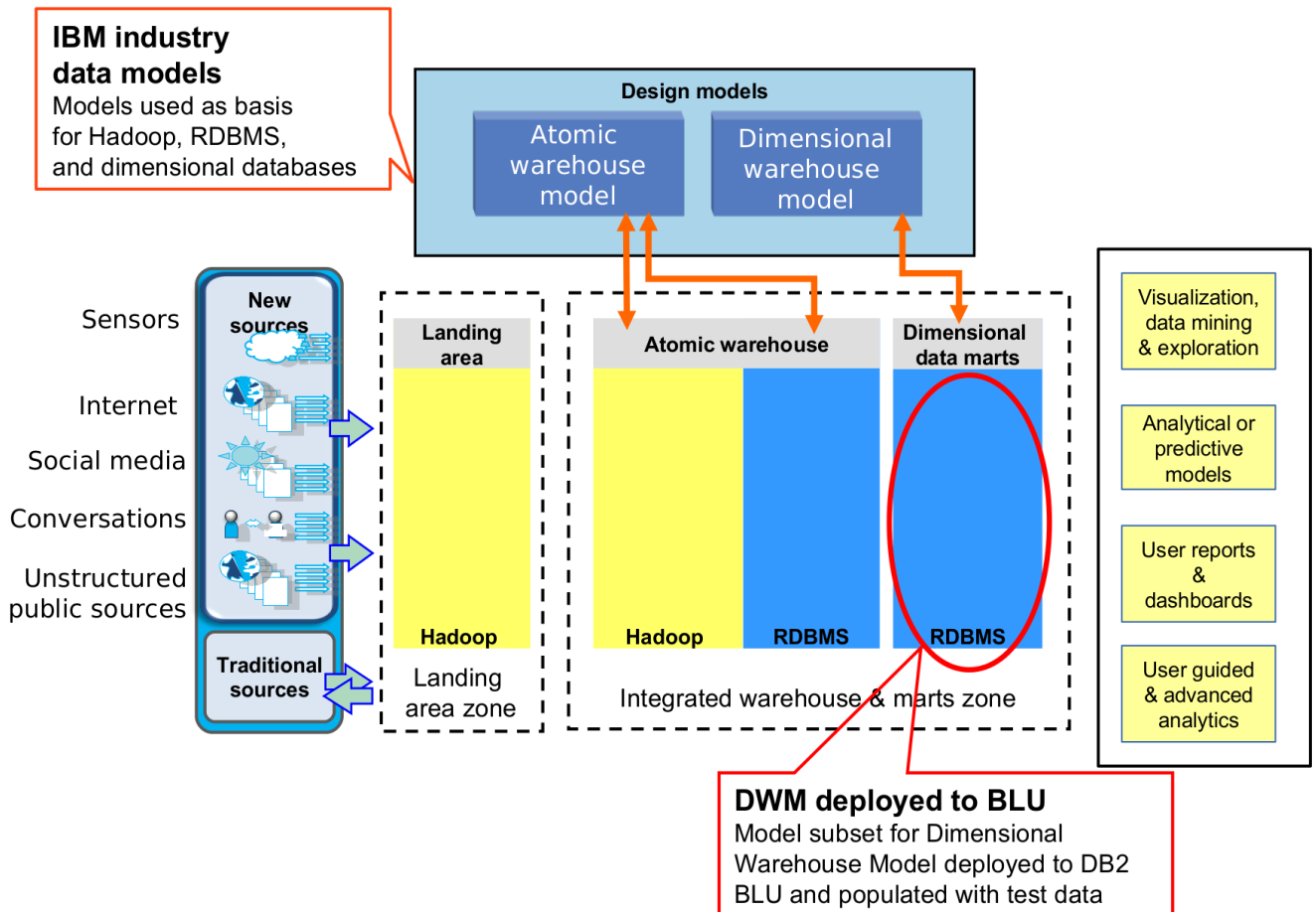


Figure 2. The dimensional warehouse model within the context of the IBM big data reference architecture

In the IBM big data reference architecture, the dimensional warehouse model contains the dimensional data that is populated by aggregating from both Hadoop and DB2 databases.

A dimensional data warehouse typically features the following items:

- Queries that have grouping, aggregation, range scans, or joins
- Queries that access a subset of a table's columns
- Database designs that often include a star or snowflake schema.

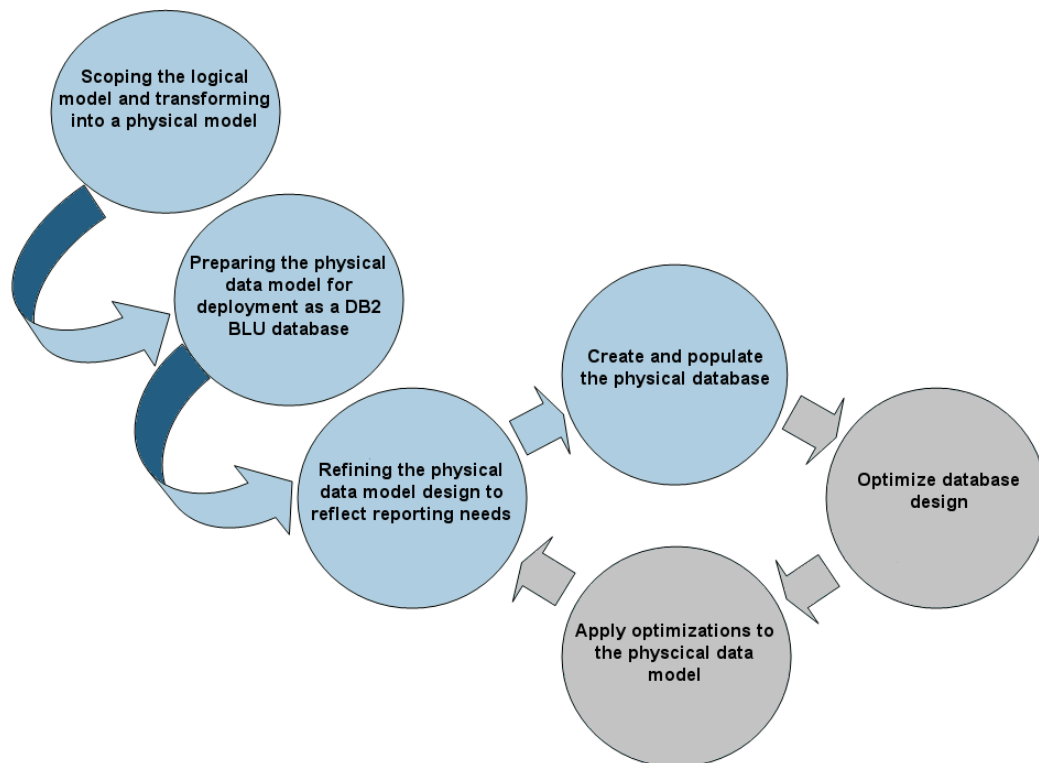


In general, column-organized tables significantly speed up analytic workloads or data mart types of workloads.

DB2 BLU is therefore a natural choice for the DWM.

## Implementing an industry model as a physical database

Implementing a logical data model as a physical database presents technical challenges. There are several steps through which you typically create and optimize your physical database for production use



*Figure 3. Typical deployment patterns for creating a physical database*

By using DB2 10.5 with BLU Acceleration, the need for the iterative refinement of the physical data model is not required, so you can avoid the steps “Optimize database design” and “Apply optimizations to the physical data model” from Figure 3. There’s no need to create indexes, define multidimensional clustering tables, build materialized query tables (MQTs), and so on.

This leaves the following phases:

1. Scoping the logical model and transforming it into a physical model.
2. Map reporting requirements to the logical data model to determine the scope of the model. Include only those entities and attributes for which you have a reporting requirement.
3. Preparing the physical data model for deployment as a DB2 BLU database.  
  
Update the physical data model to be compatible with the target DB2 BLU database architecture.
4. Populating the physical database.

### *Understanding database design challenges*

The logical data model as provided contains no vendor specific database features. You must implement the features of your database software in the physical data model.

Focus your data warehouse design decisions on the following aspects:

- **Query performance.** Efficient query performance minimizes resource usage.  
  
Using BLU Acceleration eliminates the need for indexes, MQTs, or time-consuming database tuning to maximize query performance. There is a new option for the DB2\_WORKLOAD registry variable to automatically set configuration parameters that are most relevant to BLU Acceleration and the performance of analytic workloads.
- **Intelligent table space design.** This is the ability to manage, move, and archive data as it ages.  
  
Intelligent table space design facilitates flexibility in backup and recovery strategies.
- **Data ingest.** The design should allow for ingest of data with minimal effect on data availability.  
  
Implement an architecture in which data ingest and data backup can operate concurrently.
- **Online maintenance operations.** It is important to design for concurrent database operations.  
  
Reduce the number of operations that are needed to maintain data availability and query performance and enable online database operations.



**Focus on performance, scalability, and balance when you design your database environment.**

## Scoping the logical model and transforming into a physical data model

Scoping the industry model is the process of selecting the business objects that you need from the logical data model to build a valid physical data model. The physical data model must reflect your analytical and reporting needs.

As a prerequisite to the scoping phase, you must model your business requirements and map these to analytical requirements. The quality and availability of your data sources must also be understood and assessed. However, these tasks are outside the scope of this document, which focuses on the implementation of a production physical database.

When you complete the process of scoping, you can transform your logical data model into a physical data model by selecting a menu option in IBM InfoSphere Data Architect.

**IBM InfoSphere Data Architect** is a data design tool that you can use to scope, transform, and customize the data models that are supplied in the Industry Models solutions. The examples in this paper reference InfoSphere Data Architect and you can reference the “**Further Reading**” section for more details about the product.

### *Scoping the logical model*

The logical data model is designed to meet all aspects of reporting for an industry sector. Your enterprise might not need all of the objects that are provided. Scoping is the process, by using InfoSphere Data Architect or other modeling tools, of selecting those entities from the logical data model that align with your analytical requirements.



**Refine your scope as much as possible to address your current data and reporting needs.**

When you scope the industry model to create your logical data model, use these steps with InfoSphere Data Architect:

1. Create a diagram, into which you can drag those entities that you need, to address your warehousing and reporting needs.

Creating a diagram for your logical data model avoids directly changing the base model. This method allows you to more easily accept future industry model upgrades.

2. Navigate through the **Aggregate Facts** section and drag the aggregate facts that you need into your new diagram.

Aggregate facts are related to the supporting fact tables that can be identified and included when you are completing the scoping process. Supporting entities can be identified under the heading “DWM Source”.

3. Use InfoSphere Data Architect to identify and include all related entities, both fact and dimensions tables, in the diagram.

Avoid manually moving individual related entities because this can affect the integrity of the resulting database. Let InfoSphere Data Architect identify and automatically add all related entities; you can then prune those entities that you do not need.

Figure 4 depicts a fictitious reporting requirement based on the Banking Data Warehouse industry model which is used for this paper.

**Risk Management Dashboard augmented with Social Media Sentiment**

**Top 10 Exposure at Default Watchlist**

Customer	Industry	Exposure at Loan Default	Number of Loans	Negative Social Sentiment %
Neon Chromium Partners	G Trade Repair Vehicles	\$8,365,262.96	14	80%
Molybdenum Wild Rice Partners	N Health & Social Work	\$7,531,730.72	12	81%
Tungsten Napa Cabbage Incorporated	D Manufacturing	\$6,328,300.87	11	21%
Lead Pineapple Enterprises	D Manufacturing	\$5,898,305.83	11	18%
Zirconium Durian Development	N Health & Social Work	\$5,730,963.70	8	17%
Lutetium Bell Pepper Incorporated	D Manufacturing	\$5,675,722.04	11	17%
Tin green peas ABF	E Electricity Gas & Water Supply	\$5,674,673.39	9	17%
Cobalt Kale ABF	F Construction	\$5,351,558.62	8	18%
Astatine Bamboo Shoots & Associates	M Education	\$5,198,692.59	9	20%
Yttrium Damson Associates	D Manufacturing	\$5,118,164.95	8	18%

Please select a customer:

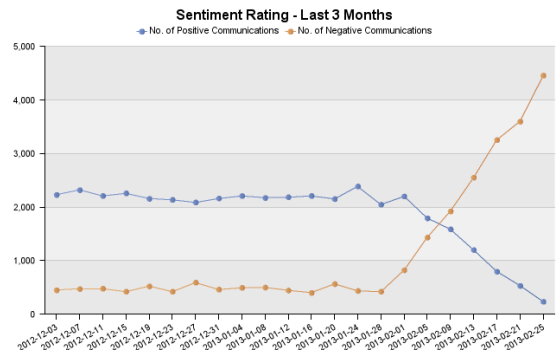
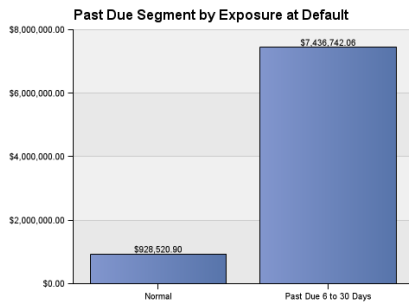


Figure 4. Sample report augmented with Social Media Sentiment

Based on this requirement, the following are a few examples of entities that are required to be scoped:

- The *Exposure at Loan Default* in the **Top 10 Exposure at Default Watchlist** report, requires the *Exposure at Default EAD* attribute from the *Customer Credit Risk Profile* entity to be scoped.
- The **Past Due Segment by Exposure at Default** report requires the *Arrangement Past Due Aging Segment* attribute from the *Finance Service Arrangement Mini Dimension* entity to be scoped.
- The *Negative Social Sentiment %* in the **Top 10 Exposure at Default Watchlist** report, as well as the **Sentiment Rating – Last 3 Months** report require the measures *Number of Negative Communications*, *Number of Neutral Communications* and *Number of Positive Communications* to be scoped.

Seven main entities are accordingly identified as candidates to be scoped to meet this reporting need:

- **Social Media Sentiment Analysis**  
This contains an analysis of the social media activity relating to the financial institution or subject of interest. It looks at measures such as the social media sentiment, the exposure of the social media page, and the number of new leads generated during a social media campaign.
- **Customer**  
A Customer is a role played by an Involved Party that is considered to be receiving services or products from the Financial Institution or one of its Organization Units, or who is a potential recipient of such services or products.
- **Calendar Date**  
Calendar Date contains the dates of the Gregorian calendar and is used to facilitate analysis by time dimension at day level.  
Calendar month, quarter, and year are in separate entities to allow more flexibility to join the time dimension to fact tables where the grain is day, month, quarter, or year.
- **Organization Mini Dimension**  
This is a grouping of attributes used to perform analysis on instances of Organization that share the same characteristics.
- **Finance Service Arrangement**  
This identifies an Arrangement wherein the Financial Institution puts its assets at risk, for a fee, for the benefit of its customer's use.

- **Finance Service Arrangement Mini Dimension**

This is a grouping of attributes used to perform analysis on instances of Debt Arrangement that share the same characteristics.

- **Customer Credit Risk Profile**

This entity is used to determine profiles of Customer Credit Risk in terms of the amount of credit in arrears, average balances, credit score and customer balance sheet, and thereby help to reduce the risk of customer credit by forecasting the profile of the customer most likely to incur credit risk, and give preventative advice.

The resulting Logical Data Model diagram is as follows:

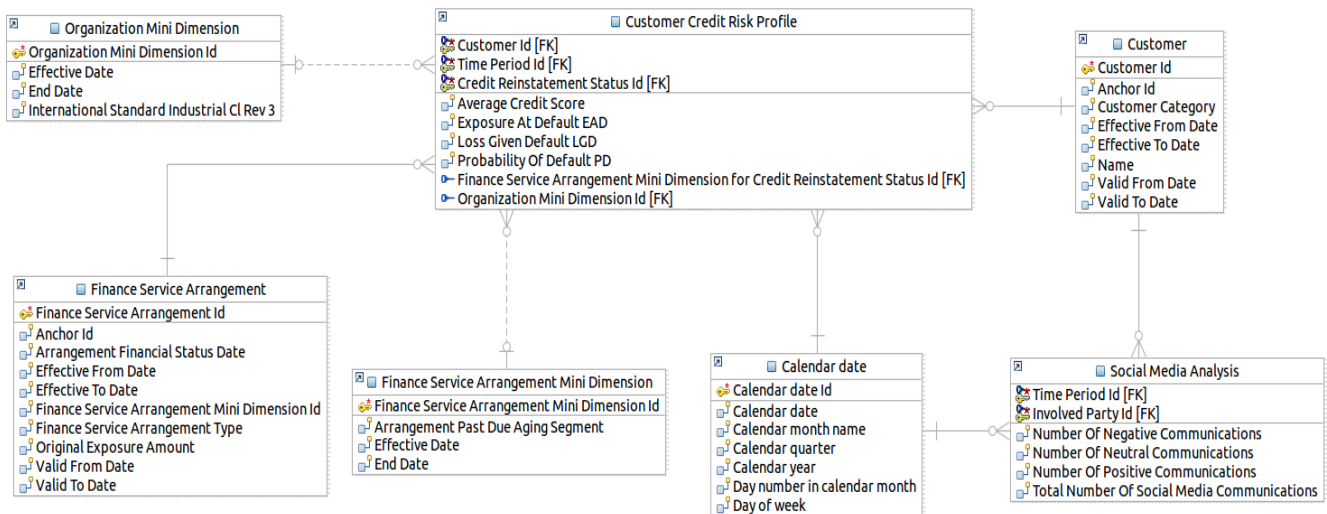


Figure 5. The scoped logical data model

## Transforming the logical data model into a physical data model

Because the logical data model applies to all databases, minimize the database architecture and design changes that you make to the logical data model. Instead, implement those changes in the physical data model. This strategy provides the following benefits:

- Easier upgrade strategy for future releases of industry models as only semantic differences will exist between your model and the industry model.
- Focus technical modeling effort on the physical data model and retain a logical data model that is suited for all databases.

- More easily control changes, in both the logical and physical data models, by assigning clear roles to each model. The logical model functions as a semantic master while the physical model is the technical master.

Entities can be added to the diagram at a later stage and merged into an existing physical data model by using the compare and merge functionality in InfoSphere Data Architect. Select this approach to build your physical data model incrementally over time.



**Apply architecture and database design changes that are specific to DB2 databases to the physical database model rather than the logical data model to help accommodate future upgrades.**

Transform your logical data model into a physical data model by selecting a blank area in the diagram and, from the InfoSphere Data Architect main menu, selecting **Data > Transform > Physical Data Model**.

When prompted for further details, select the DB2 database version that you require. For DB2 BLU choose DB2 V10.5.

Use the default settings that are provided, apart from to specify the Schema Name, and complete the transformation process.

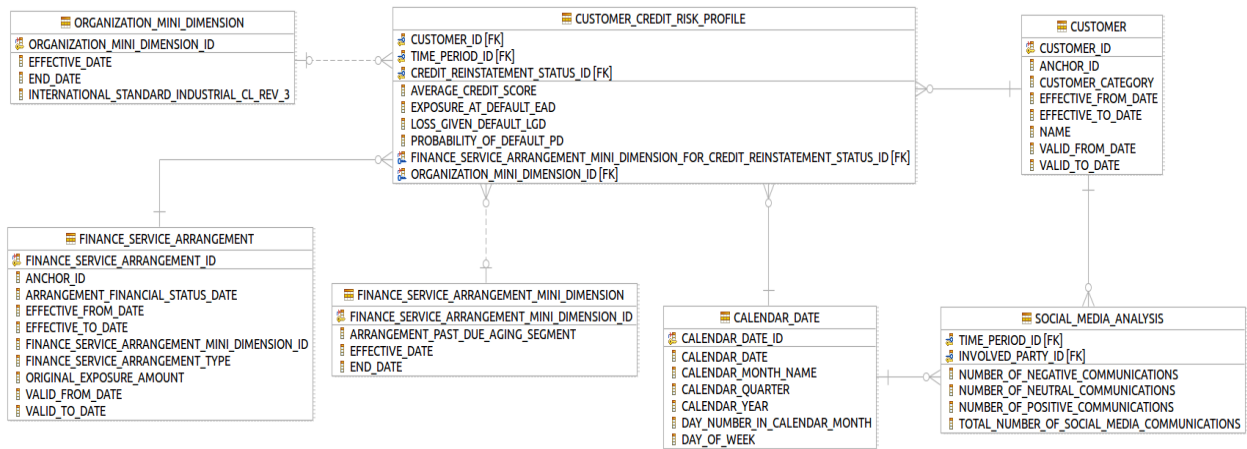


Figure 6. The physical data model

## Preparing the physical data model for deployment as a DB2 BLU database

The physical data model is a representation of your logical data model that is compatible with DB2. However, the model does not yet reflect a data warehouse architecture and design.

Several best practice recommendations for data warehousing can be applied to the physical data model before you generate DDL statements that are suitable for a DB2 BLU database environment. These improvements include the following items:

- Choosing table organization
- Implementing a table space strategy
- Customizing keys and data types

Refer to the “**Further reading**” section and the best practices paper called “*Physical database design for data warehouse environments*” for detailed explanations and examples.

Outside of these recommendations, BLU Acceleration eliminates the need for time-consuming database tuning to achieve top performance and storage efficiency.

After following these recommendations, validate your DB2 installation and the physical data model by generating the DDL statements to create a test database. From the main menu, select **Data > Generate DDL**.

### *Choosing column or row organized tables*

DB2 10.5 with BLU Acceleration supports using both row-organized and column-organized tables in the same database, even in the same table spaces and buffer pools. The benefit of having column organization and row organization in the same database is that you can choose to optimize the table organization based on the workload.

Row-organized tables have long been optimized for online transactional processing (OLTP) environments.

In general, column-organized tables significantly speed up analytic workloads or data mart types of workloads. A data mart or warehouse typically features queries that contain:

- Grouping, aggregation, range scans or joins
- Queries that access a subset of a table’s columns
- Database designs that often include a star or snowflake schema



Now with column-organized tables placed next to row-organized tables, you have the best of both worlds in the same database.

In the reporting example for the BDW, because the tables are based on a star-schema design, the tables involved are column-organized. Because it is an analytic workload, set the DB2 registry variable DB2\_WORKLOAD to ANALYTICS. This setting specifies that any tables created in the DB2 database are, by default, column organized. In InfoSphere Data Architect set the default for the database table organization to COLUMN in the physical data model. This setting is found in the Defaults tab for the database in the physical data model. By setting this default, this means that if a table is column organized then no table organization will be specified in the Infosphere Data Architect generated CREATE TABLE statement. Because the DB2 registry variable DB2\_WORKLOAD is set to ANALYTICS these tables will be created as column organized in the DB2 database.

Although changing an existing table from row organized to column organized using the db2convert utility does update a table's organization setting, update the physical data model to reflect the change in the table's organization.

### *Implementing a table space strategy*

For improved manageability, create fact tables in their own table space. For example, with a fact table in one table space and dimension tables in a different table space, you can have different storage groups and buffer pools for each table space and its associated table type, and you can backup and restore fact tables independently.

The table space for each table can be specified in the physical data model by right-clicking on the database and then clicking **Add Data Object**. For the BDW example, the TS\_BLU\_FCT1 and TS\_BLU\_FCT2 table spaces were specified for the two fact tables and the TS\_BLU\_DIM table space for the dimensions.



#### **Intelligent table space design facilitates concurrent database maintenance operations**

Correcting a poor table space design strategy post production can have a negative effect on resource usage and data availability:

- Significant resources are needed to physically move data from one table space to another post production.
- Having too few table spaces restricts your flexibility in performing data-specific backup and restore operations, performing maintenance on specific ranges of data or tables, and managing the data life cycle.
- Having too many table spaces creates unnecessary overhead for the database manager when you activate the database and maintain recovery history. It can also require too many database operations to be issued in parallel to complete tasks.

## *Customizing keys and data types*

The physical data model, when generated, uses default keys, constraints, and data type values that you need to modify based on your source data and your approach to data ingest. Consider the recommendations in the following areas:

### **Primary keys**

The logical data model implements a composite primary key on fact tables. The primary key includes all dimension foreign keys that make the primary key unique.

By default, InfoSphere Data Architect makes these primary keys ENFORCED. DB2 10.5 introduces an option for specifying not enforced, or informational, primary key constraints or unique constraints. Because the NOT ENFORCED clause does not guarantee uniqueness, use this option only when you are certain that the data is unique.

Not enforced primary key constraints or unique constraints require significantly less storage and less time to create, because no internal B-tree structure is involved. Informational primary key, foreign key, and check constraints can be very beneficial to the query optimizer.

An enforced primary key constraint consumes space because a unique index is required to enforce the constraint. Like any index, this unique index increases the cost of insert, update, or delete operations.

If your data has been cleansed through the use of rigorous extract, transform, and load (ETL) procedures, take advantage of informational constraints, especially primary key constraints.

InfoSphere Data Architect does not support non-enforced primary keys. In the example the DDL statements generated by InfoSphere Data Architect were manually updated to use the NOT ENFORCED clause for the primary key CREATE statements.

For example, for the CALENDAR\_DATE table:

```
ALTER TABLE "DB2_BLU"."CALENDAR_DATE" ADD CONSTRAINT  
"CALENDAR_DATE_PK" PRIMARY KEY  
("CALENDAR_DATE_ID") NOT ENFORCED;
```

### **Referential constraints**

Because the logical data model is suitable for any database, referential constraints are created by default as enforced constraints. Enforced constraints are not supported on DB2 BLU for foreign keys, so change these constraints in the physical data model to informational constraints.

In a warehousing environment, because informational constraints can be used by the DB2 optimizer when compiling access plans, this use helps improve query performance.

In the example, for each of the foreign key constraints in the physical data model the “Enforced” check box was unchecked.



**Use information constraints instead of enforced constraints to minimize the effect of unique index maintenance when you are populating fact tables.**

## Identity keys and surrogate keys

The logical data model implements identity keys for the primary key, whose purpose is as a surrogate key, on each dimension table.

For dimension entities, the logical data model defines certain attributes as primary and surrogate keys.

Generated columns are not supported in DB2 BLU.

Within the physical data model, clear the GENERATED check box for each of the identity columns. In the banking example, there were five such identity columns to be updated: *Customer ID*, *Calendar Date ID*, *Financial Service Arrangement ID*, *Financial Service Arrangement Mini Dimension ID* and *Organization Mini Dimension ID*.

## Refining data types

The physical data model, when generated from your logical data model, uses default data-type settings for integer and character columns. Use the following guidelines to refine the default values but avoid over-pruning column lengths. Modifying column data-types is not supported in DB2 BLU.

- Change the default CHAR(x) data type to CHAR(18) for those columns you anticipate to be no more than 18 characters long.
- Change the default CHAR(x) data type to VARCHAR(y) for those columns you anticipate to be greater than 18 characters long.
- Ensure that the modified data types of columns that are used in table joins match for optimal query access plans.
- Use the BIGINT data type where you expect the values in the columns to be greater than the capacity of the integer data type.

DB2 BLU uses synopsis tables to optimize data skipping during the processing of specific queries. The synopsis table stores minimum and maximum values, identified by tuple sequence number, for all numeric, datetime, primary key, and foreign key columns. If you have a column that stores dates, use the DATE data type rather than character values so that the synopsis table enables effective data skipping.

Equally, if you have a non-primary key or non foreign key character column which could be represented as a numeric, then using a numeric data type allows it to be used in the synopsis table.

## Nullable columns

When possible, in column-organised tables, use NOT NULL as the default attribute for columns, except in cases where you actually need to store null values. Every nullable column in a column-organized table is actually a two-column group of columns: one for the nullability attribute and one for the actual column. These nullable columns cause extra work to be done on each row during query processing.

Select the **Not Null** check box for any column which does not need to store null values

## Populating the physical database

### *LOAD utility considerations*

Loading data using the LOAD utility into DB2 BLU column-organized tables works in much the same way as for row-organized tables in that input data parsing. Key options for the LOAD utility, and general load semantics for column-organized tables and row-organized tables are the same. There are, however, a few key differences:

- Loading data into column-organized tables has a new ANALYZE phase.
- Data is converted from row-organized input into fully formatted and compressed column-organized pages.
- The synopsis tables are maintained.

BLU Acceleration builds column compression dictionaries as part of the initial load operation on a column-organized table. Additional page-level dictionaries might be created to take advantage of local data clustering at the page level and to further compress the data.

To optimize the compression dictionaries and ensure optimal compression of the data, ensure that the initial load operation, or dictionary creation, uses a representative sample of the data. However, new values that are not covered by column compression dictionaries can still be compressed by page-level dictionaries.

### **Optimizing load for compression and speed**

For optimal compression and load performance, there are two important considerations.

First, the UTIL\_HEAP\_SZ database configuration parameter directly affects how many different values can be maintained during the compression dictionary build process. It is therefore crucial to review the UTIL\_HEAP\_SZ value. If the UTIL\_HEAP\_SZ parameter is set to AUTOMATIC, ensure that the database memory is large enough so that there is ample memory available for this parameter. Otherwise, update the UTIL\_HEAP\_SZ value to be as large as possible prior to a first load operation into a column-organized table (which builds the histograms and compression dictionary), and then reduce the value after the load operation completes. UTIL\_HEAP\_SZ can be updated dynamically.

Second, if possible, presort the data by columns that are frequently referenced by predicates that filter the fact table, or columns that are used to join highly filtered dimension tables. This type of presorting can improve compression ratios and improves the likelihood of data skipping, and better performance, by ensuring that a particular column value is clustered on fewer data pages.

## Optimizing first-time load into a column-organized table

To reduce the total time that is required to load large input data files into a column-organized table for the first time, you can reduce the duration of the ANALYZE phase by creating a column compression dictionary with only a portion of the input data file. This approach is outlined as follows:

1. Create the table. Do not create any primary keys at this time.
2. Obtain a subset of representative data from all of the data files that will be loaded into the database. If the data is already in a DB2 database, perhaps in a row-organized table, use a cursor with a sampling SELECT statement to obtain the sample.
3. Load the sample for the sole purpose of building the column compression dictionary, as in the following example:

```
LOAD FROM sample_data.csv OF DEL
REPLACE RESETDICTIONARYONLY INTO mytable;
```

4. Load and compress the full set of source data by using this prebuilt column compression dictionary, as in the following example:

```
LOAD FROM all_data.csv OF DEL
INSERT INTO mytable;
```

Step 4 does not trigger the ANALYZE phase. Repeat the LOAD command until all of the data has been loaded.

5. Create enforced or informational primary key constraints and informational foreign key constraints.

6. Issue the RUNSTATS command to ensure a complete view of the table that was populated.

This approach reduces the total load time but can also reduce the compression efficiency if the sampled data is not a representative subset of the entire data set.

The new PCTENCODED column in the SYSCAT.COLUMNS catalog view represents the percentage of values that are encoded as a result of compression for a column in a column-organized table. If the overall compression ratio for your column-organized table is too low, check this statistic to see if values in specific columns were left uncompressed. If you see many columns with a very low value (or even 0) for PCTENCODED, the utility heap might have been too small when the column compression dictionaries were created. You might also see very low values for columns that were incrementally loaded with data that was outside of the scope of the column compression dictionaries.

## Conclusion

You can successfully deploy the Banking Data Warehouse to DB2 with BLU Acceleration by following the recommendations in this paper.



## Best practices

- Refine your scope as much as possible to address your current data and reporting needs.
- Apply architecture and database design changes specific to DB2 databases to the physical database model rather than the logical data model to help accommodate future upgrades.
- Use information constraints instead of enforced constraints to minimize the effect of unique index maintenance when you are populating fact tables.
- If you have a column that stores dates, use the DATE data type rather than character values so that the synopsis table enables effective data skipping.
- Use NOT NULL as the default attribute for columns, except in cases where you actually need to store null values.
- Reduce the total time required to load large input data files into a column-organized table for the first time by first creating a column compression dictionary with only a portion of the input data file.

## **Appendix A. Test environment**

The test environment that was used in the research and development of this paper was an IBM Power 7 Server running AIX 6.1 with IBM DB2 10.5 with BLU Acceleration installed.

The physical data model was created using IBM InfoSphere Data Architect V9.1 and the database was populated with 400 million rows.

The IBM Industry Model that was used was BFMDW86.



## Appendix B. DDL statements

The DDL statements for the Banking Data Warehouse Social Media example, for deployment to a database configured for WORKLOAD=ANALYTICS, are as follows:

```
--<ScriptOptions statementTerminator=";" />

CREATE TABLE "DB2_BLU"."CALENDAR_DATE" (
    "CALENDAR_DATE_ID" CHAR(10) NOT NULL,
    "CALENDAR_DATE" CHAR(10) NOT NULL,
    "CALENDAR_MONTH_NAME" CHAR(10) NOT NULL,
    "CALENDAR_QUARTER" CHAR(10) NOT NULL,
    "CALENDAR_YEAR" CHAR(10) NOT NULL,
    "DAY_NUMBER_IN_CALENDAR_MONTH" CHAR(10) NOT NULL,
    "DAY_OF_WEEK" CHAR(10) NOT NULL
)
DATA CAPTURE NONE
IN "TS_BLU_DIM";

CREATE TABLE "DB2_BLU"."CUSTOMER" (
    "CUSTOMER_ID" CHAR(10) NOT NULL,
    "ANCHOR_ID" CHAR(10),
    "CUSTOMER_CATEGORY" CHAR(10),
    "EFFECTIVE_FROM_DATE" CHAR(10) NOT NULL,
    "EFFECTIVE_TO_DATE" CHAR(10) NOT NULL,
    "NAME" CHAR(10),
    "VALID_FROM_DATE" CHAR(10) NOT NULL,
    "VALID_TO_DATE" CHAR(10) NOT NULL
)
DATA CAPTURE NONE
IN "TS_BLU_DIM";

CREATE TABLE "DB2_BLU"."CUSTOMER_CREDIT_RISK_PROFILE" (
    "AVERAGE_CREDIT_SCORE" CHAR(10),
    "EXPOSURE_AT_DEFAULT_EAD" CHAR(10),
    "LOSS_GIVEN_DEFAULT_LGD" CHAR(10),
    "PROBABILITY_OF_DEFAULT_PD" CHAR(10),
    "CREDIT_REINSTATEMENT_STATUS_ID" CHAR(10) NOT NULL,
    "CUSTOMER_ID" CHAR(10) NOT NULL,
    "TIME_PERIOD_ID" CHAR(10) NOT NULL,
    "FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION_FOR_CREDIT
_REINSTATEMENT_STATUS_ID" CHAR(10),
    "ORGANIZATION_MINI_DIMENSION_ID" CHAR(10)
)
DATA CAPTURE NONE
IN "TS_BLU_FCT1";

CREATE TABLE "DB2_BLU"."FINANCE_SERVICE_ARRANGEMENT" (
    "FINANCE_SERVICE_ARRANGEMENT_ID" CHAR(10) NOT NULL,
    "ANCHOR_ID" CHAR(10),
    "ARRANGEMENT_FINANCIAL_STATUS_DATE" CHAR(10),
    "EFFECTIVE_FROM_DATE" CHAR(10) NOT NULL,
    "EFFECTIVE_TO_DATE" CHAR(10) NOT NULL,
```

```

        "FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION_ID"
CHAR(10),
        "FINANCE_SERVICE_ARRANGEMENT_TYPE" CHAR(10),
        "ORIGINAL_EXPOSURE_AMOUNT" CHAR(10),
        "VALID_FROM_DATE" CHAR(10) NOT NULL,
        "VALID_TO_DATE" CHAR(10) NOT NULL
    )
    DATA CAPTURE NONE
    IN "TS_BLU_DIM";

CREATE TABLE
"DB2_BLU"."FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION" (
    "FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION_ID"
CHAR(10) NOT NULL,
    "ARRANGEMENT_PAST_DUE_AGING_SEGMENT" CHAR(10),
    "EFFECTIVE_DATE" CHAR(10) NOT NULL,
    "END_DATE" CHAR(10) NOT NULL
)
    DATA CAPTURE NONE
    IN "TS_BLU_DIM";

CREATE TABLE "DB2_BLU"."ORGANIZATION_MINI_DIMENSION" (
    "ORGANIZATION_MINI_DIMENSION_ID" CHAR(10) NOT NULL,
    "EFFECTIVE_DATE" CHAR(10) NOT NULL,
    "END_DATE" CHAR(10) NOT NULL,
    "INTERNATIONAL_STANDARD_INDUSTRIAL_CL_REV_3" CHAR(10)
)
    DATA CAPTURE NONE
    IN "TS_BLU_DIM";

CREATE TABLE "DB2_BLU"."SOCIAL_MEDIA_ANALYSIS" (
    "NUMBER_OF_NEGATIVE_COMMUNICATIONS" CHAR(10),
    "NUMBER_OF_NEUTRAL_COMMUNICATIONS" CHAR(10),
    "NUMBER_OF_POSITIVE_COMMUNICATIONS" CHAR(10),
    "TOTAL_NUMBER_OF_SOCIAL_MEDIA_COMMUNICATIONS"
CHAR(10),
    "TIME_PERIOD_ID" CHAR(10) NOT NULL,
    "INVOLVED_PARTY_ID" CHAR(10) NOT NULL
)
    DATA CAPTURE NONE
    IN "TS_BLU_FCT2";

ALTER TABLE "DB2_BLU"."CALENDAR_DATE" ADD CONSTRAINT
"CALENDAR_DATE_PK" PRIMARY KEY
("CALENDAR_DATE_ID") NOT ENFORCED;

ALTER TABLE "DB2_BLU"."CUSTOMER" ADD CONSTRAINT "CUSTOMER_PK"
PRIMARY KEY
("CUSTOMER_ID") NOT ENFORCED;

ALTER TABLE "DB2_BLU"."CUSTOMER_CREDIT_RISK_PROFILE" ADD
CONSTRAINT "CUSTOMER_CREDIT_RISK_PROFILE_PK" PRIMARY KEY
("CUSTOMER_ID",
"TIME_PERIOD_ID",
"CREDIT_REINSTATEMENT_STATUS_ID") NOT ENFORCED;

```

```

ALTER TABLE "DB2_BLU"."FINANCE_SERVICE_ARRANGEMENT" ADD
CONSTRAINT "FINANCE_SERVICE_ARRANGEMENT_PK" PRIMARY KEY
("FINANCE_SERVICE_ARRANGEMENT_ID") NOT ENFORCED;

ALTER TABLE
"DB2_BLU"."FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION" ADD
CONSTRAINT "FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION_PK"
PRIMARY KEY
("FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION_ID") NOT
ENFORCED;

ALTER TABLE "DB2_BLU"."ORGANIZATION_MINI_DIMENSION" ADD
CONSTRAINT "ORGANIZATION_MINI_DIMENSION_PK" PRIMARY KEY
("ORGANIZATION_MINI_DIMENSION_ID") NOT ENFORCED;

ALTER TABLE "DB2_BLU"."SOCIAL_MEDIA_ANALYSIS" ADD CONSTRAINT
"SOCIAL_MEDIA_ANALYSIS_PK" PRIMARY KEY
("TIME_PERIOD_ID",
"INVOLVED_PARTY_ID") NOT ENFORCED;

ALTER TABLE "DB2_BLU"."CUSTOMER_CREDIT_RISK_PROFILE" ADD
CONSTRAINT "CUSTOMER_CREDIT_RISK_PROFILE_CALENDAR_DATE_FK"
FOREIGN KEY
("TIME_PERIOD_ID")
REFERENCES "DB2_BLU"."CALENDAR_DATE"
("CALENDAR_DATE_ID")
NOT ENFORCED;

ALTER TABLE "DB2_BLU"."CUSTOMER_CREDIT_RISK_PROFILE" ADD
CONSTRAINT "CUSTOMER_CREDIT_RISK_PROFILE_CUSTOMER_FK" FOREIGN
KEY
("CUSTOMER_ID")
REFERENCES "DB2_BLU"."CUSTOMER"
("CUSTOMER_ID")
NOT ENFORCED;

ALTER TABLE "DB2_BLU"."CUSTOMER_CREDIT_RISK_PROFILE" ADD
CONSTRAINT
"CUSTOMER_CREDIT_RISK_PROFILE_FINANCE_SERVICE_ARRANGEMENT_FK"
FOREIGN KEY
("CREDIT_REINSTATEMENT_STATUS_ID")
REFERENCES "DB2_BLU"."FINANCE_SERVICE_ARRANGEMENT"
("FINANCE_SERVICE_ARRANGEMENT_ID")
NOT ENFORCED;

ALTER TABLE "DB2_BLU"."CUSTOMER_CREDIT_RISK_PROFILE" ADD
CONSTRAINT
"CUSTOMER_CREDIT_RISK_PROFILE_FINANCE_SERVICE_ARRANGEMENT_MINI_D
IMENSION_FK" FOREIGN KEY
("FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION_FOR_CREDIT_REINSTAT
EMENT_STATUS_ID")
REFERENCES
"DB2_BLU"."FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION"
("FINANCE_SERVICE_ARRANGEMENT_MINI_DIMENSION_ID")
NOT ENFORCED;

```

```
ALTER TABLE "DB2_BLU"."CUSTOMER_CREDIT_RISK_PROFILE" ADD
CONSTRAINT
"CUSTOMER_CREDIT_RISK_PROFILE_ORGANIZATION_MINI_DIMENSION_FK"
FOREIGN KEY
  ("ORGANIZATION_MINI_DIMENSION_ID")
REFERENCES "DB2_BLU"."ORGANIZATION_MINI_DIMENSION"
  ("ORGANIZATION_MINI_DIMENSION_ID")
NOT ENFORCED;

ALTER TABLE "DB2_BLU"."SOCIAL_MEDIA_ANALYSIS" ADD CONSTRAINT
"SOCIAL_MEDIA_ANALYSIS_CALENDAR_DATE_FK" FOREIGN KEY
  ("TIME_PERIOD_ID")
REFERENCES "DB2_BLU"."CALENDAR_DATE"
  ("CALENDAR_DATE_ID")
NOT ENFORCED;

ALTER TABLE "DB2_BLU"."SOCIAL_MEDIA_ANALYSIS" ADD CONSTRAINT
"SOCIAL_MEDIA_ANALYSIS_CUSTOMER_FK" FOREIGN KEY
  ("INVOLVED_PARTY_ID")
REFERENCES "DB2_BLU"."CUSTOMER"
  ("CUSTOMER_ID")
NOT ENFORCED;
```

## Further reading

- “Governing and managing enterprise models”  
(<http://www.ibm.com/developerworks/rational/library/10/governingandmanagingenterprisemodels-series/index.html>)
- “DB2 for Linux, UNIX, and Windows best practices”  
(<http://www.ibm.com/developerworks/data/bestpractices/db2luw/>)
- “Best practices: Optimizing analytic workloads using DB2 10.5 with BLU Acceleration”  
(<https://ibm.biz/BdDrnq>)
- “Scoping the IBM Industry Model for banking using Enterprise Model Extender and InfoSphere Data Architect”  
(<http://www.ibm.com/developerworks/data/tutorials/dm-1003bankindustrymodel/>)

## Contributors

Austin Clifford

*Data Warehouse Specialist*

Pat G. O’Sullivan

*Senior Technical Staff Member, Industry Models Architecture*

Gary Thompson

*Information Architect, Industry Models Architecture*

Bryan Tierney

*Business Analyst, Industry Models*

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment does so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE: © Copyright IBM Corporation 2014. All Rights Reserved.

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

## **Trademarks**

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

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.

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