Dimensional modeling with IBM InfoSphere Data Architect, Part 1: Forward engineering in InfoSphere Data Architect

Multidimensional modeling

Yun Feng Bai
Prabhudoss Samuel

July 28, 2011

IBM® InfoSphere® Data Architect (IDA) is a collaborative data design solution that helps you discover, model, relate, and standardize diverse and distributed data assets. It is a pivotal component of the IBM initiative to enable an integrated data management environment throughout the entire data management lifecycle. In this series of articles, learn how to build a dimensional data model using IBM InfoSphere Data Architect that efficiently captures analytical requirements at the logical and physical levels of detail. Part 1 focuses on using forward engineering to achieve a multidimensional data model.

Introduction

Beginning with InfoSphere Data Architect V7.5.3, you can create relational data model and multidimensional data models. This series uses three user scenarios to demonstrate how it helps accelerate multidimensional data modeling and how users can benefit from InfoSphere Data Architect V7.5.3 adoption. The three user scenarios are multidimensional data modeling through forward engineering, data modeling through reverse engineering, and data model transformations between InfoSphere Data Warehouse and Cognos® Framework Manager.

Scenario overview

A retail company is planning to develop one system to manage sale transactions and another system to analyze the business. Now it has created normalized data models, including products, employees, customers, and stores, in addition to sales for the transaction system. For the business analysis system, the company needs to create multidimensional models based on the normalized data model.
To fulfill the requirements for business analysis, a typical workflow will be introduced to show you how to create multidimensional data models through forward engineering using InfoSphere Data Architect.

Key steps in the workflow include:

- Discovering multidimensional information based on a normalized data model
- Transforming the normalized logical data model to a de-normalized dimensional logical data model
- Transforming the dimensional logical data model to dimensional physical data model
- Transforming the dimensional physical data model to a Cubing or Cognos model

**Know your model**

The retail company has created a logical model as shown in Figure 1, where we will get a basic understanding of the model. It is assumed that you would have created a data design project and have successfully created the below model using InfoSphere Data Architect V7.5.3 or later.

**Figure 1. The retail sales model**

Drag and drop all the entities onto the diagram. You will see that the entities contained in the model depict the following relationships:

- Employee and the corresponding departments denoted by:
  - Employees
  - Department
- Individual stores and their locations denoted by:
  - Store
  - Store_Region
- Customers and their locations denoted by:
• Customers
• Customer_Type
• Region
• Territories
• Products and their suppliers denoted by:
  • Products
  • Brand
  • Packaging
  • Categories
  • Supplier
  • Supplier_Type
• Billing denoted by:
  • Store_Billing
  • Store_Billing_Details

Enabling dimensional notation

The first step for enabling dimensional notation is to enable dimensional capability in the logical data model. Right-click on the data model and then choose the Use Dimensional Notation menu item. Your model can now hold dimensional properties.

Figure 2. Enabling dimensional notation
In a similar way, you can remove the dimensional capabilities of the model by unchecking the option.

Note: Once you have some dimensional information put in the models, unchecking the option would only remove the dimensional properties from your view. Internally, the information is still persisted in the model. This is a soft removal of dimensional properties and can be brought back by enabling the notations again.

Creating a normalized dimensional logical data model

Taking a look at the model now, you should probably get to know that the Store_Billing entity must be a Fact entity. You can change a dimensional property of this entity in the following way:

1. Select the entity and open the Properties view.
2. Find the tab labelled Dimensional and select the Change the dimensional entity type checkbox.
3. The Type panel is enabled and will appear as shown in the figure below.
4. Select the Fact option. The Store_Billing entity will now be a Fact.

Figure 3. Setting dimensional properties.

Note: As you may have guessed, selecting None would make the entity a normal one. This is a hard removal, as the dimensional information is removed at the model level itself.

But isn't this a slower way? Don't we need a faster way to add dimensional properties? Read on.

Add dimensional properties by automated discovery

InfoSphere Data Architect provides a powerful feature that automates the identification of entities to different dimensional properties. You can do this as follows:

1. In the Data Project Explorer, select the data model.
2. Right-click, and on the pop-up that appears, click Discover Facts and Dimensions.
3. A box will pop up asking you if you want the hierarchy to be generated for any entities of type Dimension. Choose **No**. You can learn and use more on hierarchies after the transformation process.

4. After completion of discovery, as shown in the figure below, you will have different dimensional properties being applied to the entities. This is a normalized dimensional logical model.
   - The entity Brand has been discovered as an Outrigger.
   - The entity Products has been discovered as a Dimension.
   - The entity Store_Billing_Details has been discovered as a Fact. The attributes Unit Price and Quantity have been discovered as Measure.
   - The entity Territories was left as such.
Note: The above discovery process is just a recommendation based on InfoSphere Data Architect's logic and is not required. Using the manual method outlined above, you can still change the properties if desired. It is also worth mentioning that the discovery logic depends upon the existing dimensional properties of the model. Hence, it is recommended that you apply as much dimensional information that you are sure of before initiating the discovery process. This being the case, the resulting model will be more aligned with your requirements.

Creating a de-normalized dimensional logical data model

Having generated the normalized model, you need to de-normalize it to suit your business needs. Transformation mechanisms available in InfoSphere Data Architect will help us achieve that:

1. Click on the logical data model node.
2. Choose **Data > Transform > New Configuration**.

   **Figure 6. The transformation menu**

   ![Transformation Menu](image)

   3. This will open the transformation configuration options window.
4. Specify **LDM2DLDM** for the configuration.

5. Choose **Logical Data Model to Dimensional-Logical Data Model** option.

**Figure 7. The transformation options — file name and type**

6. Click **Next**.

7. Choose the input logical model and the output folder as shown in the screenshot.
8. Click **Next**.

9. Choose the following options in the next screen.
   - Create a star schema.
   - Create the date and time dimension if applicable.
   - Enable the generate traceability option.

**Figure 9. The transformation options — schema type, date dimension and traceability**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degenerate and, if dimensional notation exists, denormalize to</td>
<td>Star schema</td>
</tr>
<tr>
<td>Generate Date and Time dimension entities (if applicable)</td>
<td>true</td>
</tr>
<tr>
<td>generate traceability</td>
<td>true</td>
</tr>
</tbody>
</table>

10. Click **Finish**.

11. In the transformation configuration window that opens, click **Run**.

12. A new file, Package1_D.ldm, is created. This is a de-normalized version of your logical model.

13. Take a quick look at the file and you would know that:
   - Date and time entities have been added. They have been classified as Dimension entities.
   - The multiple entities in the normalized model have been de-normalized to denote data with a reduced number of tables.
   - Dimension entities have been retained.
   - The fact entity Billing_Details has been retained.
Figure 10. The de-normalized dimensional logical model

• The numeric columns in fact entity has been classified as Measure.

Figure 11. The measures classified in Fact entity

14. A closer look at the Date entity reveals that:
• Two hierarchies named FiscalYear and Year have been created.
• They have individual levels defined that correspond to Year, Quarter, Month, and Date.
• These levels actually relate to the drill-down reports. In other words, the query can answer the sales information that took place:
  i. based on Year
  ii. based on Quarter
  iii. based on Month
  iv. based on Date
15. Click on the level **FiscalYear** and check its properties in the **Properties** view.

16. Each level should have exactly one caption attribute. We will add this to our de-normalized logical dimensional model at this point of time.

17. Check the box below the caption column for the level FiscalYear.

**Figure 13. Adding caption attributes for a level**

18. Repeat the above process for all levels available for FiscalYear and Year hierarchies.

19. Take a look at the Store Billing Details fact entity.

20. You can infer that new relationships have been created with the newly created entities Date and Time.

**Figure 14. References to new entities — Date and Time**

The process of creating a de-normalized dimensional logical model is now complete. By now, you should be able to relate that the fact Store Billing Details has got the actual data of a transaction and is at the Center. The details of the individual transaction can be seen in the Dimension entities surrounding it. Does this resemble a Star schema? Please continue with the diagramming section below.

Note: There are three measure types: Non-additive, Additive, and Semi-additive. The default measure classified from auto-discovery is additive measure with SUM as aggregation function. You can update the additive measure to other aggregation function and can also classify non-additive measures and semi-additive measures.
Visualize your dimensional model

Having created the de-normalized model, it would be a nice time to know that you can view them in dimension-specific diagrams:

1. In the Data Project Explorer, right-click on the **Diagrams** node.
2. Click on the **New Dimensional Blank Diagram** menu item.

   **Figure 15. Creating a dimensional data diagram**

3. You can see a new diagram (Diagram1) has been created, and the diagram editor opens on the right side.
4. Select all the entities in your normalized dimensional model, and drag them and drop them into the Diagram editor.
5. You should now be able to see the model in all its glory. See that this now resembles a star.
Figure 16. The star schema as shown in dimensional diagram

Note: As an aside, you can directly add the dimensional entities in the diagram by using the Dimensional widget that is available on the right side of the diagram editor.

Publishing the model

It is always recommended that you have the model reviewed with people in your organization. To facilitate this, you can publish your current model in HTML format and share it across the organization as applicable. Here’s how:
1. Right-click on **Package1** in the Package1_D.ldm.
2. Click **Data >> Publish >> Web**.
3. Fill in the required information as below.

   **Figure 17. Web publish — Dimensional logical data model**

   ![Publish to Web dialog box](image)

   4. Click **OK**.
   5. Open the index.html in the C:\MyDDLDMReport folder. You should see the entire model has been converted to HTML format. You can then share this across your organization.

### Transform de-normalized dimensional logical data model to dimensional physical data model

In the section above, we have the de-normalized dimensional logical data model reviewed by stakeholders. Before the dimensional logical data model is finalized, you can update the dimensional logical data model based on the feedback from stakeholders and continue with the review process.

In this section, we are going to transform the de-normalized dimensional logical data model to dimensional physical data model.

1. Right click the de-normalized dimensional logical data model node, and click **Transform to Physical Data Model** from the context menu.
Figure 18. Transform to physical data model from context menu

2. In the Transform To Physical Data Model wizard, select **Create new model** and then click **Next**.
3. Keep the **Destination folder** and **File name** set at their defaults. As we are going to transform the model to DB2® for Linux®, UNIX®, and Windows® V9.7, select Database as **DB2 for Linux, UNIX, and Windows**, and Version as **V9.7**, then click **Next**.
Figure 20. Specify database, version, and location for transformation

4. Select **Generate traceability**, which can be used for object trace in future, and update Schema name as `RETAIL_SALES`, then click **Next**.
Figure 21. Specify options for transformation

5. In the Output page, the transformation status is displayed. Click Finish to generate the dimensional physical data model.
Now we have the dimensional physical data model generated with the dimensional notations added to the source dimensional logical data model. You can add more database-specific information to the dimensional physical data model, but we will not introduce much here.

To make sure the transformed dimensional physical data model is compliant with enterprise standards, analyzing the model is always recommended. We can use the Analyze Model function to analyze the transformed dimensional physical data model.

**Analyze transformed dimensional physical data model**

1. Right-click on the schema RETAIL_SALES in the transformed dimensional physical data model in the Data Project Explorer, then click **Analyze Model**.
2. In the Analyze Model wizard, all the analyze rules under category physical data model are selected by default. Seven rules are added in InfoSphere Data Architect V7.5.3 for dimensional physical data model validation. Click **Finish** to run the analyze model process.
3. The analysis results are displayed in the Problems view, similar to the snapshot below. No error message is found.

**Figure 25. Analyze results for The Transformed Dimensional Physical Data Model**

<table>
<thead>
<tr>
<th>Property</th>
<th>SQL Results</th>
<th>Problems</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 errors, 136 warnings, 22 others (filter matched 122 of 158 items)</td>
<td>Description</td>
<td>Resource</td>
<td>Path</td>
</tr>
<tr>
<td>Warnings (100 of 136 items)</td>
<td>Brand ID might contain either lower case or RETAIL_SALES... Retail_Sales</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Generate DDL from Dimensional Physical Data Model**

Now the user can generate DDL, which can be used for the dimensional schema deployment later, from the dimensional physical data model:

1. Right-click on the schema node **RETAIL_DETAILS** in the Data Project Explorer, then click the context menu item **Generate DDL**.
Figure 26. Generate DDL menu item to generate DDL for selected object

2. Customize the options to generate DDL and leave the options in the Generate DDL wizard as default, then click Next.
Figure 27. Customize options to generate DDL

3. Customize the objects to generate DDL and leave the objects in the Generate DDL wizard as default, then click Next.
4. Now the DDL for schema RETAIL_SALES is generated, and it will be saved to the specified file with the specified folder. You can run the DDL on the specified server and open the DDL file for editing after the Generate DDL process completes. Leave the properties as default and click Next.
5. The summary page of the Generate DDL wizard lists the details for the Generate DDL process. Click **Finish**.
Transform the dimensional physical data model to Cubing/Cognos model

In the section above, one valid dimensional physical data model is transformed from the de-normalized dimensional logical data model. To make sure the dimensional model could be used within business intelligence tools, we need to transform the dimensional physical data model to the InfoSphere Warehouse Cubing model or Cognos Framework Manager model. In InfoSphere Data Architect V7.5.3, one new transformation is added to transform dimensional physical data model to the Cubing/Cognos model.

Transform the dimensional physical data model to Cubing model:

1. Right-click on the dimensional physical data model node and click context menu item New > Transformation Configuration.
Figure 31. Create new transformation configuration

2. In the New Transformation Configuration wizard, specify the name and the destination for the transformation configuration, select the transformation **Dimensional-Physical Data Model to Cognos/Cubing Model**, then click **Next**.
3. Select the schema RETAIL_SALES as the source from the left tree and select the project as the target for transformed model, then click Next.
4. Four properties are available for the transformation. The first two are useful only when **Target dimensional model** is selected as the Cognos model. Select **Name** for the property **Name source of table and column for Logical/Dimensional View** and **Cubing Model** as the target model, then click **Finish**.

**Figure 34. Specify transformation properties**
5. The transformation configuration is opened in the editor. The user can view the properties of the transformation configuration and update if necessary. Click the toolbar button **Validate the transformation configuration** to validate the transformation configuration created above, and no validation error is expected in the Console view.

**Figure 35. Validate the transformation configuration**

6. Click **Run** to run the transformation process. Once the process completes, one Cubing model is generated under the XML Schemas folder in the target project.

**Figure 36. Run the transformation configuration and generate Cubing model**

The Cognos model can be transformed following the steps above, but two more properties are available for the transformation to Cognos. For detailed introduction of the properties, please refer to the [Information Center](https://www.ibm.com/ibm/en).

**Import transformed Cubing/Cognos model to InfoSphere Warehouse/ Cognos Framework Manager**

Now we have the Cubing and Cognos models from the transformation in the section above, and you can import the model to related products for further update and deployment. In this section, we are going to import the Cubing model generated above to InfoSphere Warehouse Design Studio:

1. Create a Data Design Project in InfoSphere Warehouse Design Studio.
2. Create a Physical Data Model with OLAP in the Data Design project above.

**Figure 37. Create new physical data model with OLAP**

3. Right-click on the physical data model node and select **Import** from the context menu.

**Figure 38. Import Cubing model**

4. In the Import wizard, select **Data Warehousing > OLAP Metadata**, then click **Next**.
5. Specify the Cubing model generated above and the target as the database node of the physical data model, then click **Next**.
Figure 40. Specify source and target to import

6. The OLAP objects to be imported are listed. Click Finish.
7. Click OK on the pop-up dialogs to complete the import. The OLAP objects are imported to the physical data model in the Data Project Explorer.
In the Cubing model, most OLAP objects are generated from the dimensional physical data model in InfoSphere Data Architect, such as cube models, facts, dimensions, measures, hierarchies, and levels. But no cube is generated. So you need to add cube to the Cubing model before it can be deployed. There are also some other gaps between InfoSphere Data Architect dimensional model and InfoSphere Warehouse Cubing model.

**Conclusion**

We have completed the workflow to create multidimensional data models through forward engineering using InfoSphere Data Architect V7.5.3. The retail company can use the dimensional schema for database deployment and use the Cubing or Cognos model for business intelligence deployment.

InfoSphere Data Architect can help accelerate multidimensional data modeling from design to deployment. You can greatly benefit from the features InfoSphere Data Architect provides, such as dimensional information discovery, model de-normalization to dimensional schema, and transformation from dimensional physical data model to InfoSphere Warehouse Cubing model or IBM Cognos Framework Manager model.

**Acknowledgements**

Thanks to Erin Wilson, Qi Yun Liu, and Bo Yuan for the review of this article.
# Downloadable resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source model</td>
<td>DMinIDA_FE_SourceModel.zip</td>
<td>20KB</td>
</tr>
</tbody>
</table>
Related topics

• Read "Dimensional Modeling: In a Business Intelligence Environment," an IBM Redbooks® publication that guides the user to design dimensional modeling in a business intelligence environment.
• Download a trial version of InfoSphere Data Architect V7.5.3 and learn how to create a dimensional model efficiently.
• In the InfoSphere Data Architect area on developerWorks, get the resources you need to advance your data modeling skills.

© Copyright IBM Corporation 2011
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)