

Whitemarsh
Information Systems Corporation

Whitemarsh Metabase Data Modeler: View Data Model Users Guide

November 2007

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: mmgorman@wiscorp.com
Web: www.wiscorp.com

Table of Contents

1	Introduction	1
2	Software Installation	2
3	Database Design	2
4	Reference Data	3
5	Operation	4
6.1	Fact Data	7
6.1.1	View Data Models	7
6.1.1.1	Views	7
6.1.1.2	Generate Views	10
6.1.1.2	View Columns	10
6.1.1.3	Generate View Columns	13
6.1.2	Assignments	13
6.1.2.1	Assign DBMS Columns to View Column	14
6.1.2.2	Assign View Column to Compound Data Element	14
6.1.2.3	Assign View Column to Derived Data Element	14
6.1.3	View Column Mapping	18
6.1.3.1	View Column Structure	18
6.1.3.2	View Column Structure Process	22
6.1.3.3	View Column Structure Type	24
6.2	Reports	26



List of Figures

Figure 1. View data model.	3
Figure 2. Login screen.	5
Figure 3. List of current views along with view columns an associated DBMS columns.	7
Figure 4. View update screen.	9
Figure 5. Generate default simple views (one view per table).	10
Figure 6. List of View Columns.	11
Figure 7. Updating a view column.	12
Figure 8. Generating view columns.	13
Figure 9. Assigning View Columns to DBMS Columns.	15
Figure 10. Assigning view columns to a compound data element.	16
Figure 11. Assigning view columns to derived data element.	17
Figure 12. View Column Structure rows in support of relating view columns from different views.	18
Figure 13. View Column Structures.	19
Figure 14. Selecting a View Column for a View Column Structure.	20
Figure 15. Selecting a View Column Process.	21
Figure 16. View Column Processes.	22
Figure 17. Updating a View Column Structure Process.	23
Figure 18. View Column Structure Types.	24
Figure 19. View Column Structure Type update screen.	25



1 Introduction

The view data model (VDM) component of the data modeler module is designed to capture specifications of models of data that represent the interface between a DBMS's database schema as expressed in the Operational Data Model, and the Business Information Systems' need for data.

The specified data model (SDM) component of the data modeler module is designed to capture specifications of models of data. That is, data models without regard to databases or DBMSs. Specified data models are sometimes called conceptual data models or even logical data models. The key characteristic of a specified data model is that its collection of entities, attributes, and relationship are all generally within a single subject area.

Databases, in contrast to specified data models often span subject areas and are intended to fulfill the data entry, update, and reporting needs of a particular constituent group. Database data models, without regard to operating system, DBMS, or actual computing environment are called Implemented Data Models (IDM).

Once the DBMS and operating system and computing environment is chosen, an implemented data model is often transformed to meet the required capacity and performance characteristics. This last type of data model is called the operational data model.

There is therefore a hierarchical relationship between the Specified, Implemented, and Operational data models, and then a final relationship to the View Data Model that represents the interface to Business Information Systems.

Data elements and the semantics that assist in data element definition are defined within the data element module of the metabase. In addition to being a source for attribute semantics, a function within the specified data model allows both the re-designation of the data element from which an attribute draws its semantics and also the ability to promote an attribute to be a data element.

Collectively the specified data models and their associated semantics act as data model templates for implemented data models. If an organization already has operational databases (usually only about 100% of the time), then, there is a promotion capability from the implemented data model to the specified data model. Once an implemented data model is highlighted and the promotion button is pressed, the complete implemented data model is "promoted" up to be a specified data model.

The document, *Data Modeler Architecture and Concept of Operations*, which can be downloaded from the Whitemarsh website, www.wiscorp.com is an essential prerequisite reading for the correct use of this data modeler component. It presents the "business problem" being addressed. This user guide only briefly presents how to accomplish the solution.

Presumed Knowledge

This user guide, and all the other metabase user guides presume that the reader has read and is completely familiar with the following documents: Metabase Common Processes, and Metabase Bill of Materials and Single File Recursion (BOM/SFR Guide). These two documents serve as metabase teaching guides for processes that commonly occur throughout the metabase system.



F7 invokes automatic spell checking on all text fields like names and descriptions.

Metabase Example

The metabase example, Movies, is a complete example of a business which is available from the Whitemarsh website. The Movies Rental Corporation was modeled after the largest movies rental corporation in the United States. As such, the example has national, regional, and retail outlets. There are two data models, one for an original data capture, store based system, and another which is a data warehouse for rented movies.

2 Software Installation

Metabase installation is explained in the Metabase Administrators Guide.

3 Database Design

The specified data model module depicted in Figure 1 has the following tables:

- View
- View Column
- View Column & Compound Data Model
- View Column & DBMS Column
- View Column & Derived Data Model
- View Column Structure
- View Column Structure Process
- View Column Structure Type

The database design is depicted in Figure 1. Explicit in this database design are the following:

- Views consist of view columns.
- A view column is mapped to one or more compound data elements, or one or more derived data elements, and one or more DBMS column. A view column also forms the basis for the interrelationship among views.
- View column & compound data elements are the relationship between a view column and one or more compound data elements. The view columns & compound data elements exist within a specified sequence.
- View column & derived data elements are the relationship between a view column and one or more derived data elements. The view columns & derived data elements exist within a specified sequence.





4 Reference Data

There is no reference data in the view data model. Readers are encouraged to thoroughly review and understand the Data Modeler Architecture and Concept of Operations book that is available from the Whitemarsh website, www.wiscorp.com.

5 Operation

Once the application is installed it is ready to use. Just invoke the software from the metabase program. The application is a traditional windows application. Metabase reports are accomplished through any ODBC class report writer such as Crystal Reports..

5.1 Log In Process

Figure 2 shows the log-in screen that appears immediately after the application is started. Entered is your user name and your password. These are created by the Metabase Administrator through the metabase administration module. Please contact your metabase administrator to set up your user name and password. Once a user name and password is established, all the user's information can be changed by the user through a restricted use version of the administrator software. Once the send button is pressed the specific metabase database instances that can be accessed by the user is presented. The metabase is such that users are allowed to use specific metabase instances and specific metabase modules.

In this particular example, the user, once they sent their user name and password are shown the metabase database that they can access, that is, Movies. Highlight the choice and press the Select button. Once that is done then the metabase name, Movies, is shown as the data set that is being accessed.



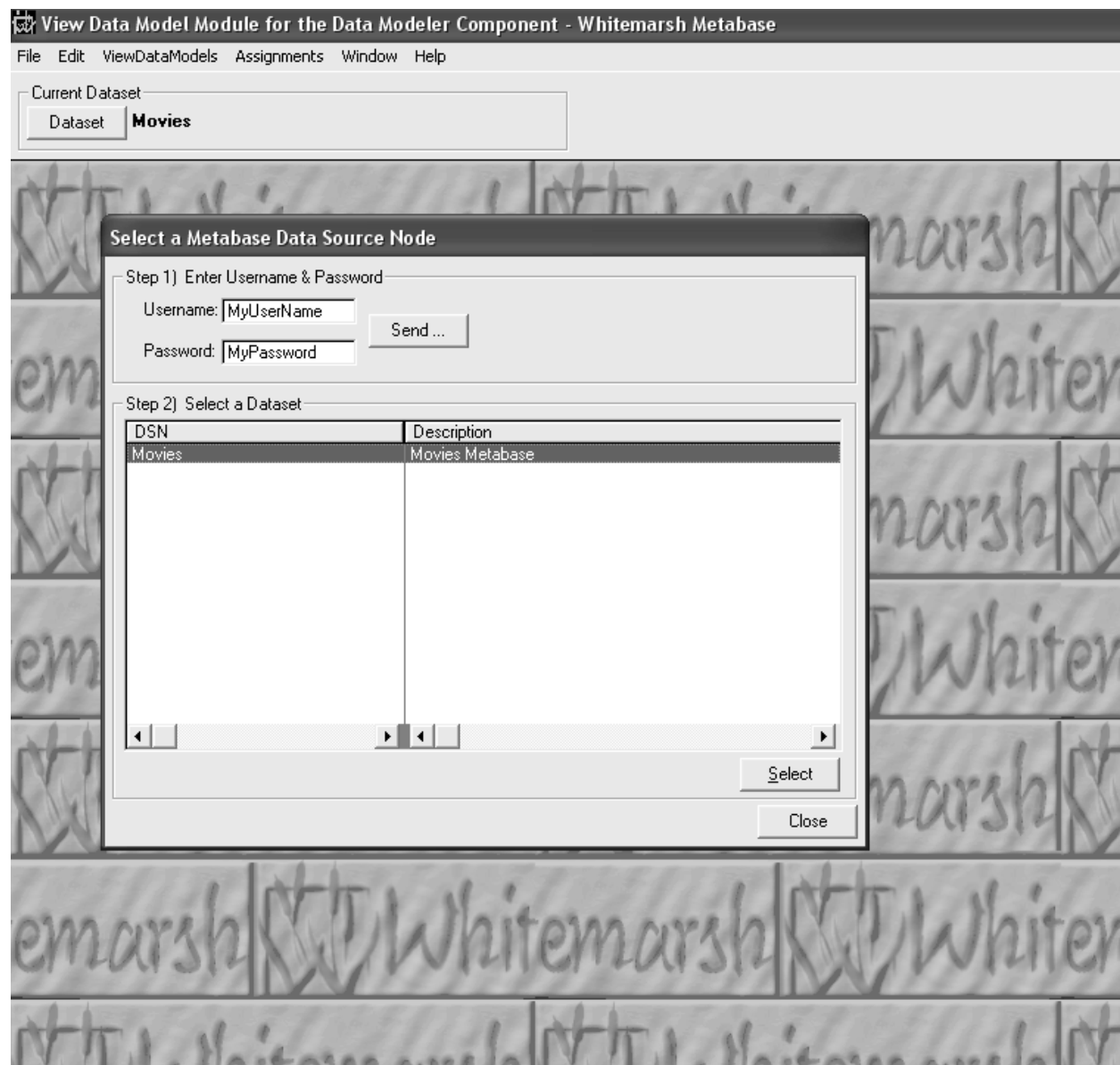


Figure 2. Login screen.



6 Process Model

The view data modeler process consists of two classes of processes:

- Fact Data Entry
- Reports

The top level menu for specified data model contains the following top level items:

- View data models
- Assignments
- Reports

Each menu item contains as appropriate, nested subordinate menu items. The complete menu is provide in the table that follows. When a actual process is activated, its existing list is presented. To add, change or delete an item on the browse list, the Insert, Change, or Delete button is pressed. The form that is then presented supports the entry of all the data that is needed.

-- View Data Models
-- Views
-- Views
-- Generate Views
-- View Columns
-- Generate View Columns
-- Multiple View Column Structures
-- View Column Structure Process
-- View Column Structure
-- View Column Structure Type
-- Assignments
-- Assign Compound Data Elements to View Columns
-- Assign Derived Data Elements to View Columns
-- Assign View Columns to &DBMS Columns
Menu for View Data Model



6.1 Fact Data

The fact data shown in consists of:

- View data models
- Assignments

6.1.1 View Data Models

The Subjects & Entities processes enable the entry and update of the main components of a specified data model. It consists of the following:

- Views
- Generate Views
- View Columns
- Generate View Columns

6.1.1.1 Views

Views and their attendant view columns represent the database interface between a DBMS and its database and a data accessing unit. A data accessing unit might be a SQL query, a 3GL program, or a 4GL procedure. A SQL view is a precise set of compilable syntax. That set of syntax is not able to be generated by the metabase. The role of the metabase is to contain a general definition of the view and an allocation of all DBMS columns that is to accessible by a view.



The list of current views, their associated view columns and associated DBMS tables and columns are shown in Figure 3. Also if an entirely new view is to be entered, or one is to be changed, then press Insert. A screen like Figure 4 is presented. At the lower right of the screen is shown the schema, database, and DBMS associated with the view.

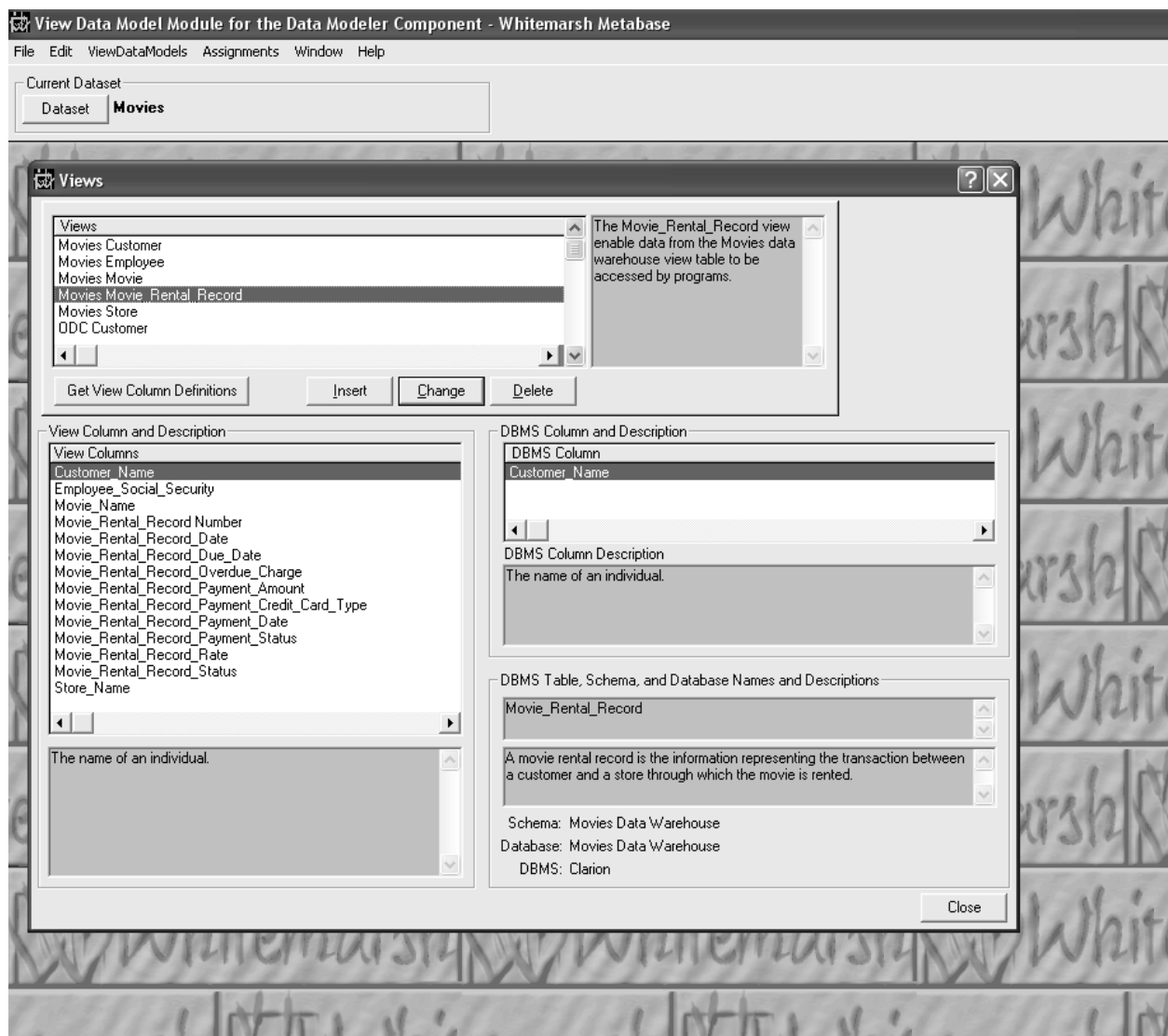


Figure 3. List of current views along with view columns and associated DBMS columns.



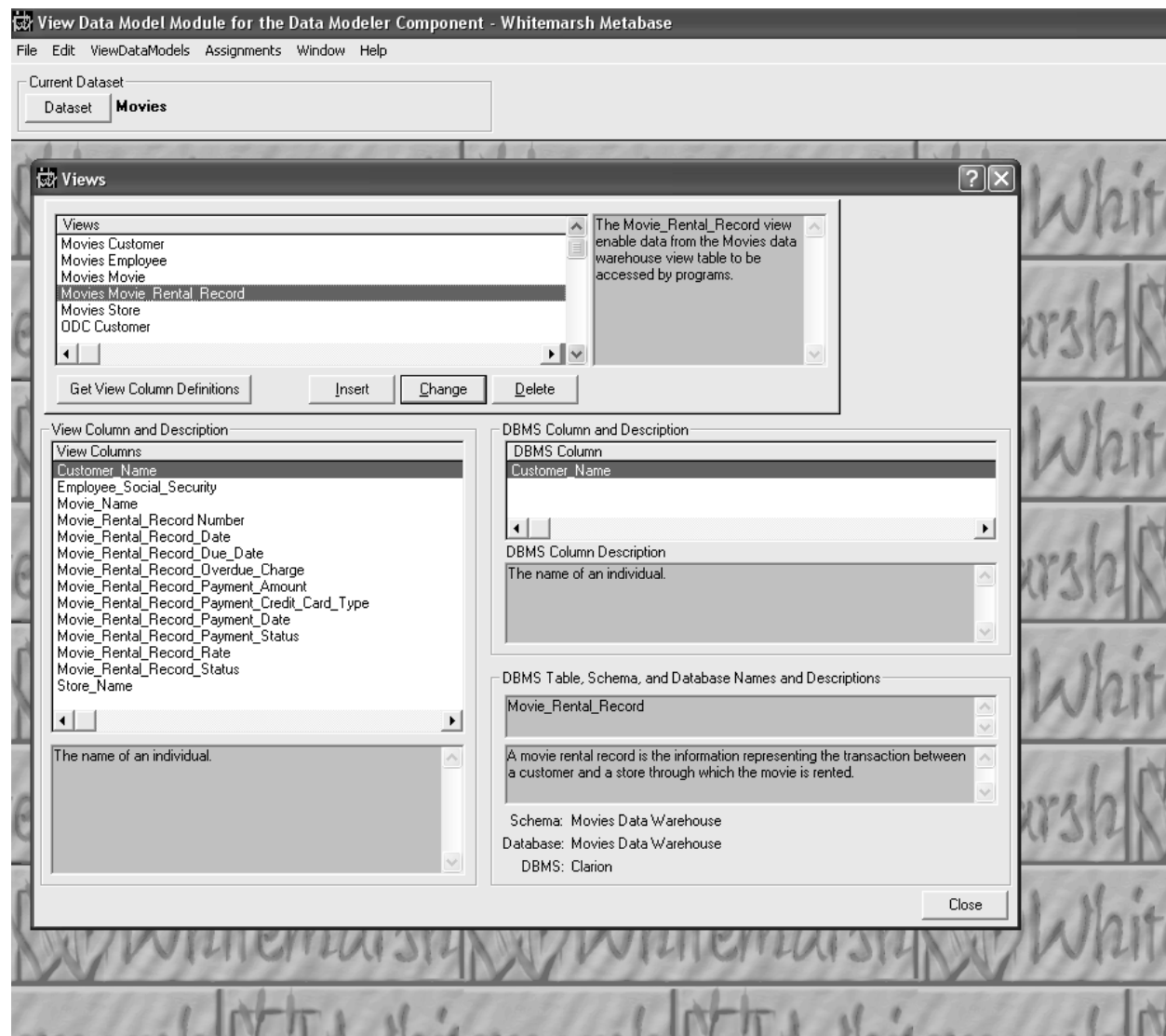


Figure 4. View update screen.



6.1.1.2 Generate Views

Figure 5 presents the screen for generating default views. A default view is a view table that corresponds exactly to a DBMS table. The process is straight forward. Highlight the appropriate database and then schema. Then tag one or more DBMS tables. Press the Build button. Views, view columns, and the relationship between the view column and the DBMS columns are built automatically. The resulting views are then displayed in the right-side windows.

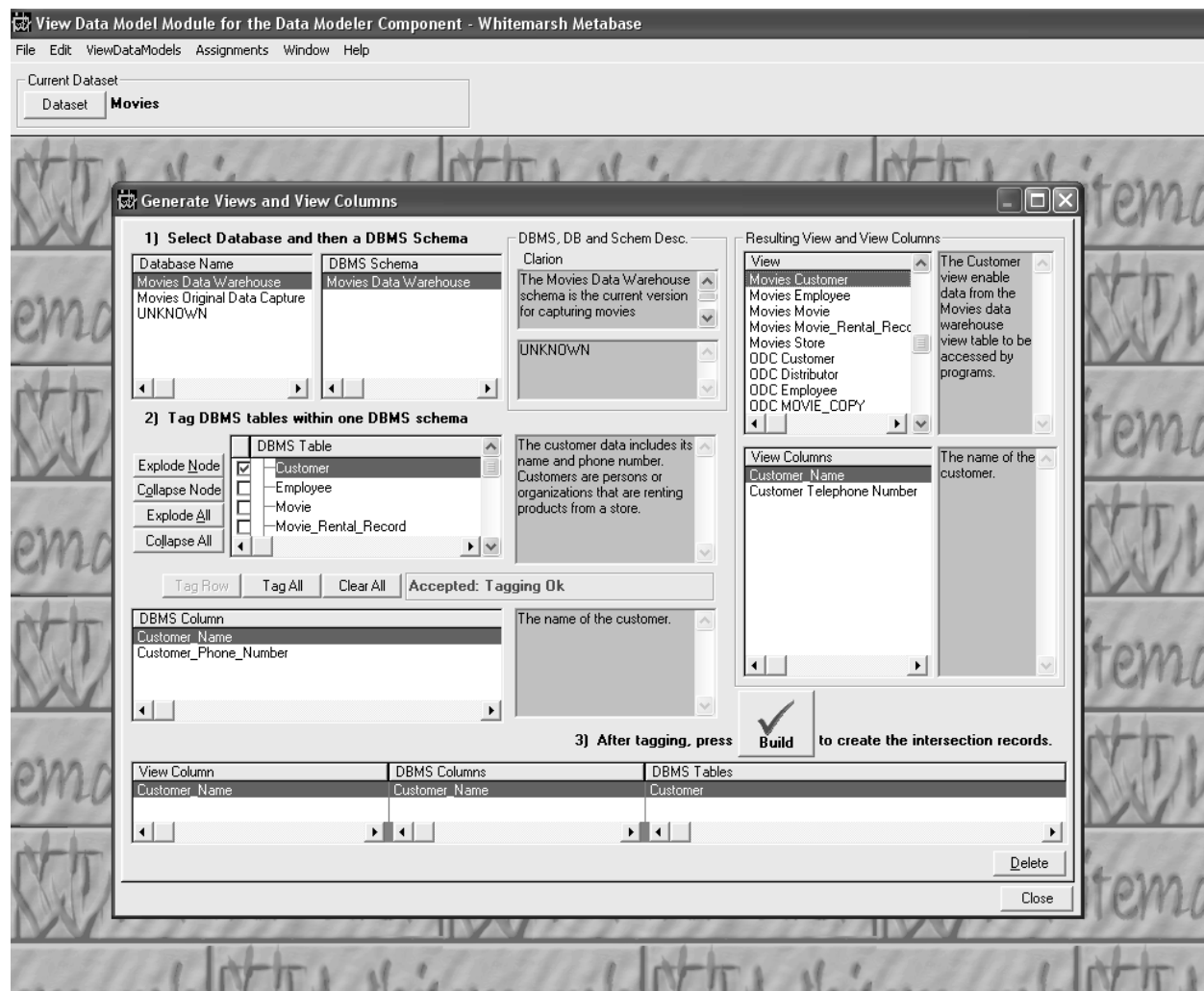


Figure 5. Generate default simple views (one view per table).

6.1.1.2 View Columns



A view column is a column within the view that is mapped to one or more DBMS columns. A view column may be mapped to one or more DBMS columns because the view column may either be a compound data element or a derived data element. In the event that a view column is not just a simple mapping of the DBMS column then the mapping process should be described in the view column.

Figure 6 presents a list of view columns for a view and the associated DBMS columns for each view column. To enter a new view column, press Insert. To modify a view column, highlight it and press enter.

Once a view column is created it may be updated. To update one, highlight it via Figure 6, and then press enter. Figure 7 presents the screen for view column updating.

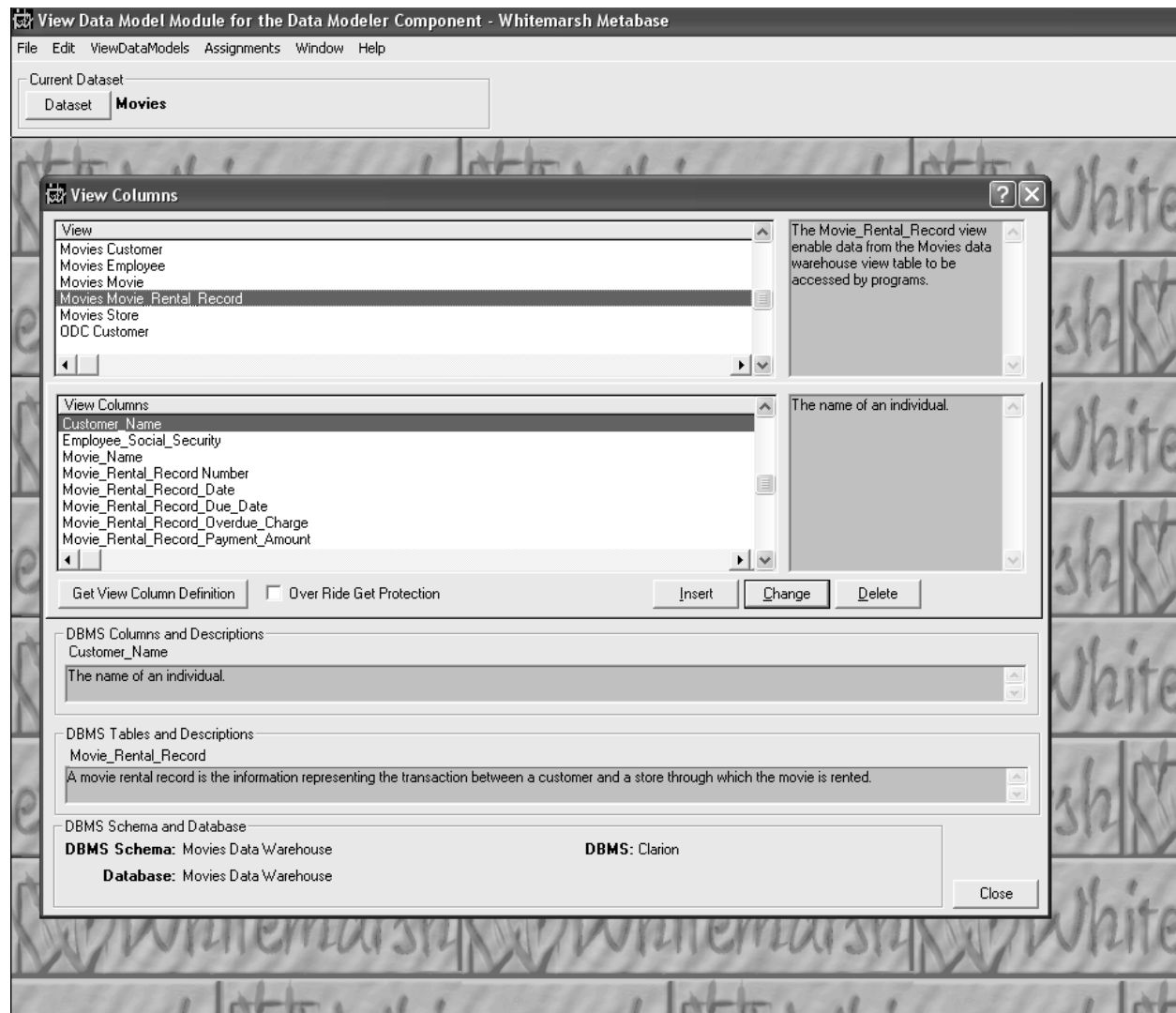


Figure 6. List of View Columns.



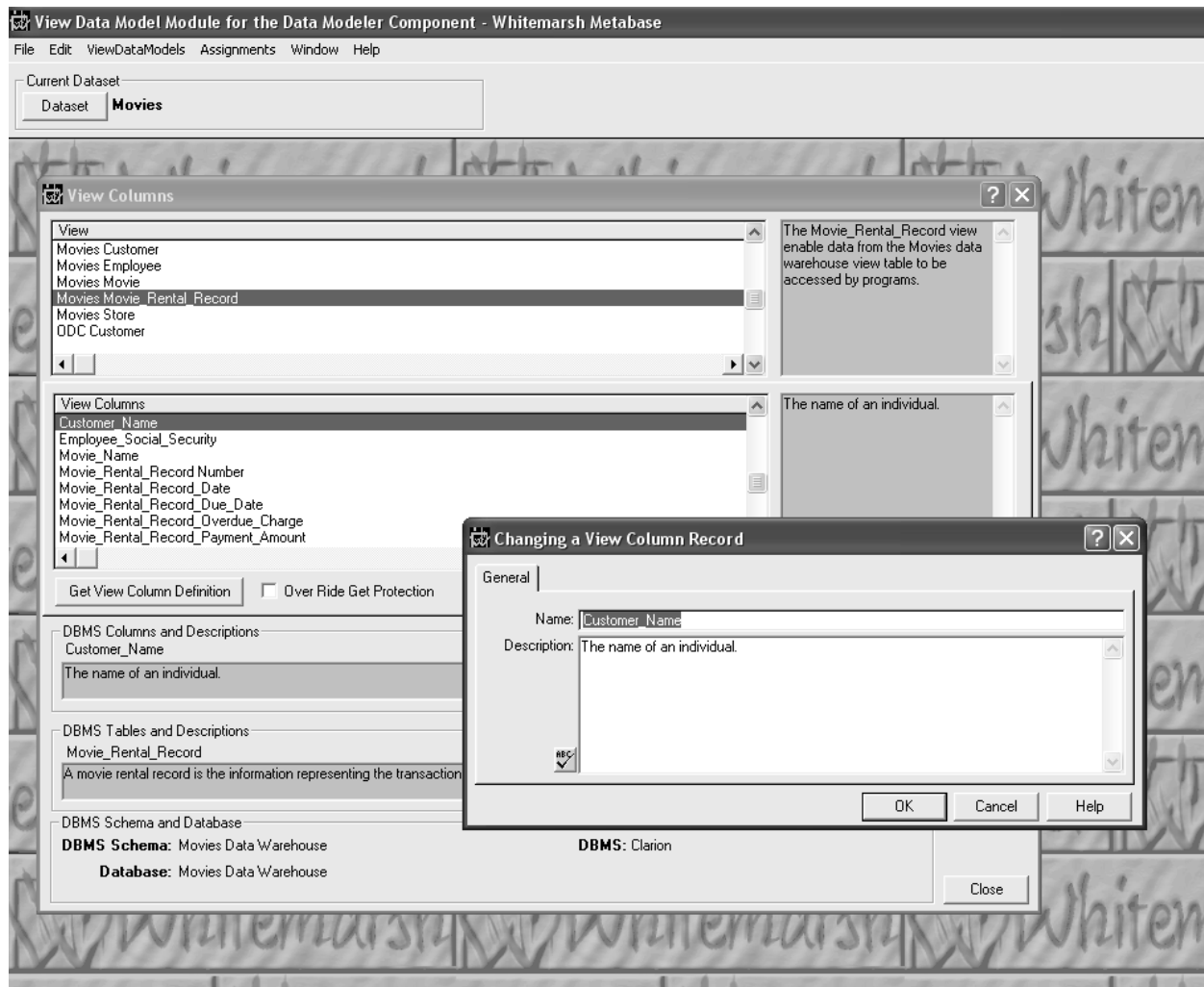


Figure 7. Updating a view column.



6.1.1.3 Generate View Columns

Figure 8 shows the screen for generating view columns from DBMS columns within DBMS tables. The process is simply to highlight the View, and then highlight and tag the DBMS columns that are to belong to the view. Once the Build button is pressed then the view columns and the relationship between the DBMS columns and the view columns are created. As can be seen, the view column names are identical to the DBMS column names.

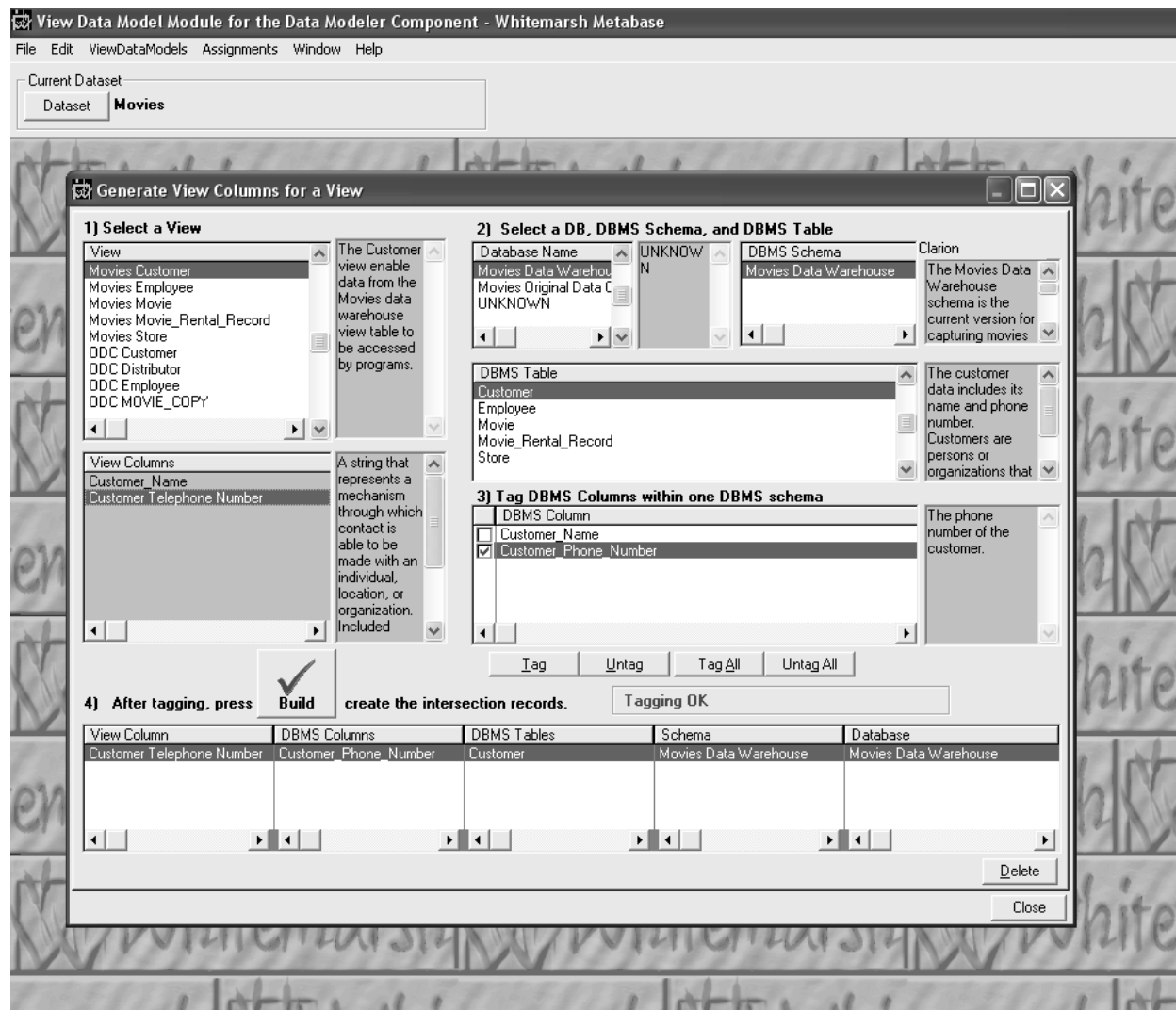


Figure 8. Generating view columns.

6.1.2 Assignments

There are three types of assignments:



- Assign DBMS columns to view column
- Assign view column to compound data element
- Assign view column to derived data element

6.1.2.1 Assign DBMS Columns to View Column

Figure 9 presents the screen that supports the process of assigning DBMS columns to a view. Highlight and tag a view column. Then tag one or more DBMS columns from one or more DBMS tables. Then press the build button. Mapped DBMS columns then appear in the bottom window.

6.1.2.2 Assign View Column to Compound Data Element

A view data element may be complex. That is, it may either be a compound data element, for example, like Person Full Name. In this case, the full set of semantics of the view column should be mapped to the appropriate compound data element. The assignment process is illustrated in Figure 10.

6.1.2.3 Assign View Column to Derived Data Element

A view data element may be complex. That is, it may either be a derived data element, for example, like Person Age. In this case, the full set of semantics of the view column should be mapped to the appropriate derived data element. The assignment process is illustrated in Figure 11.



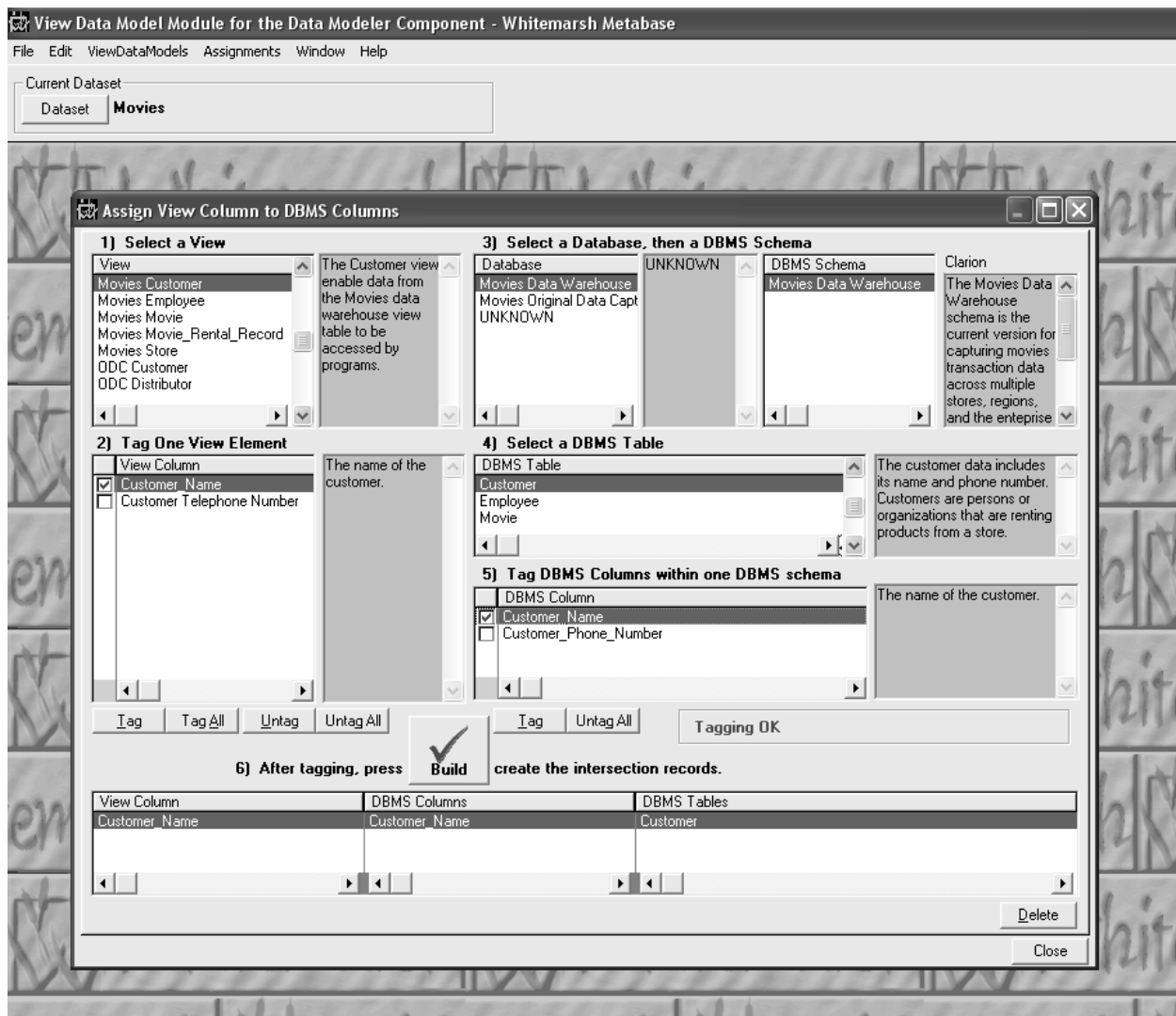


Figure 9. Assigning View Columns to DBMS Columns.



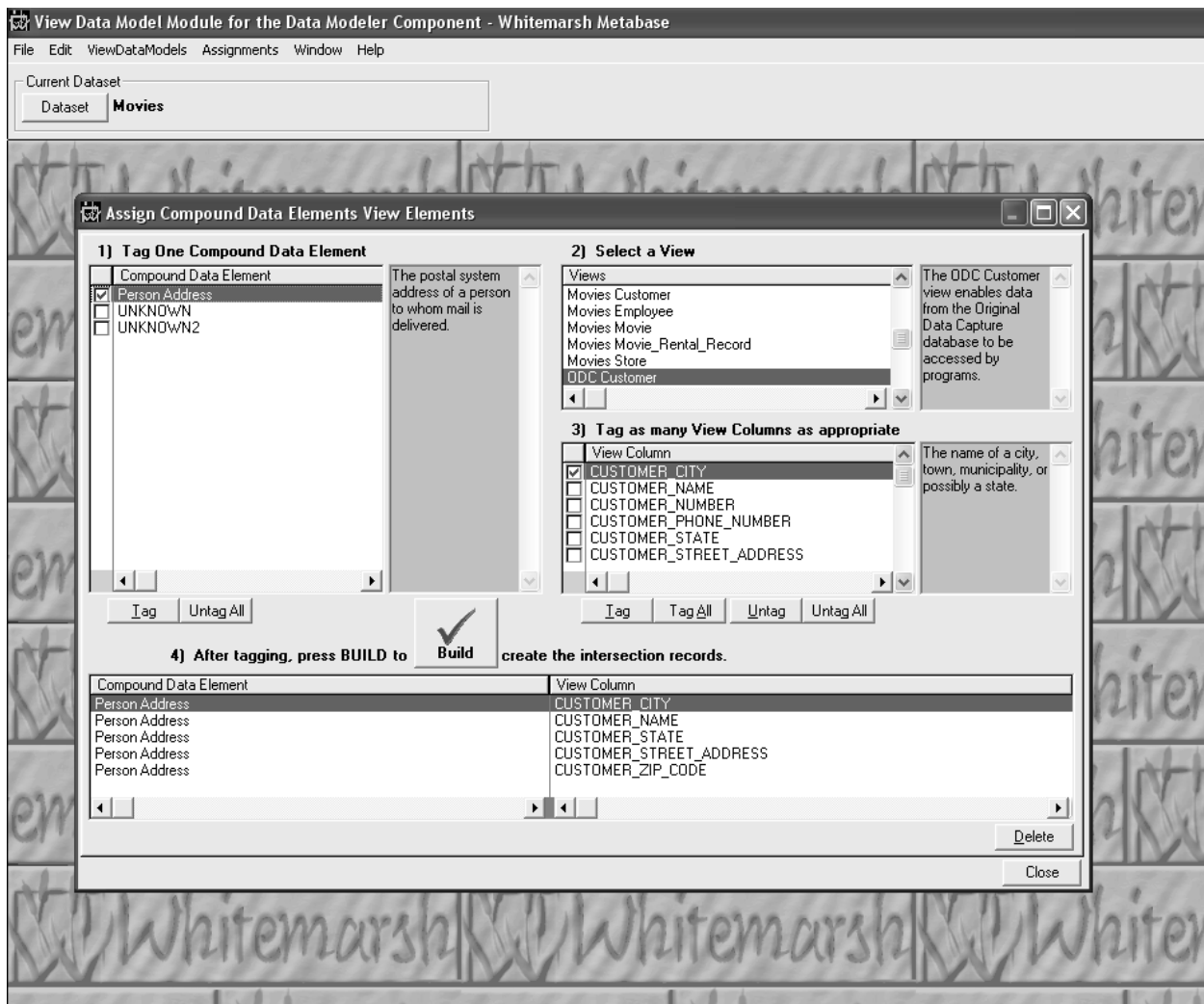


Figure 10. Assigning view columns to a compound data element.



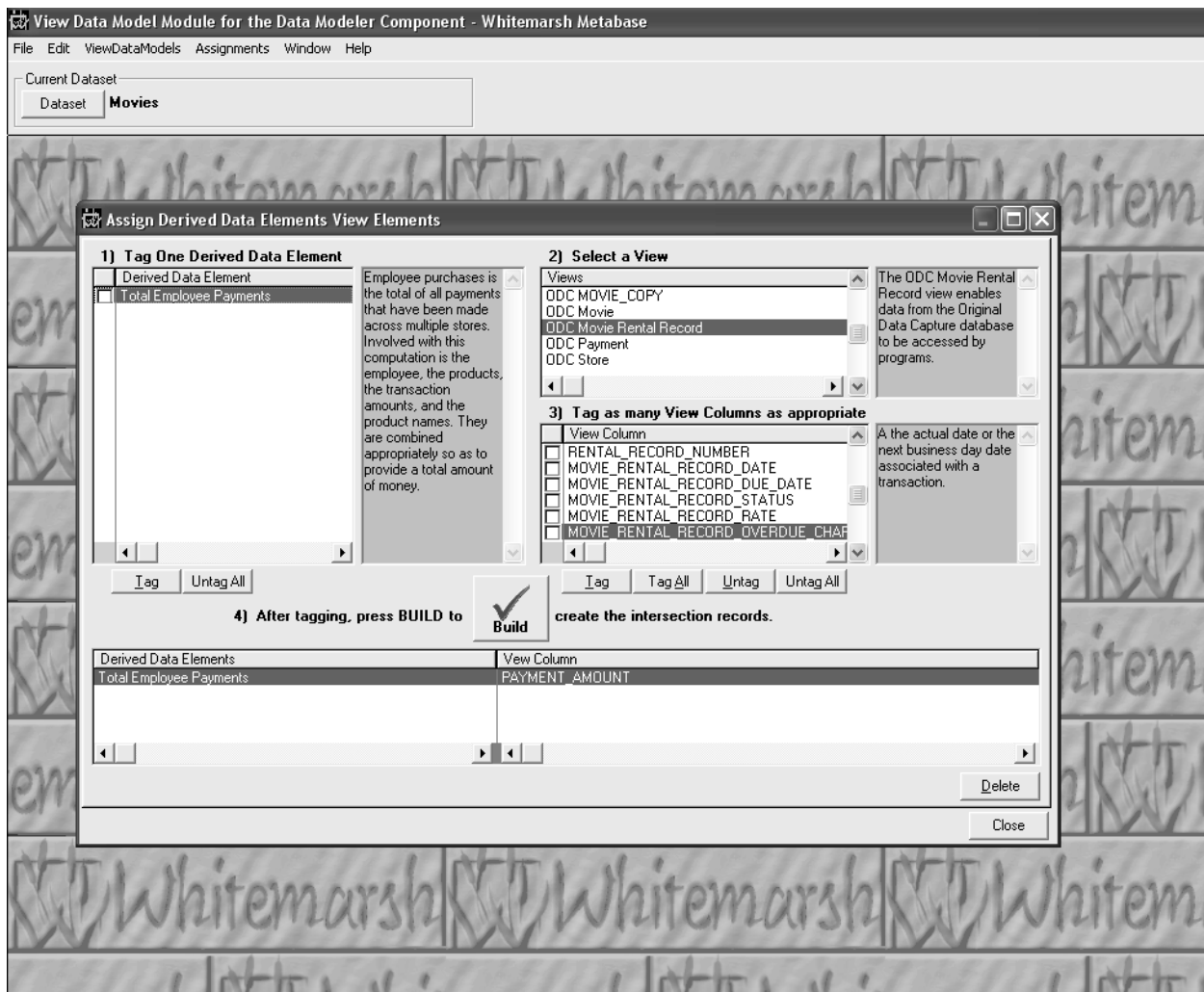


Figure 11. Assigning view columns to derived data element.



6.1.3 View Column Mapping

To assist in the tracing of data migration within a business information system, the view data model provides the ability to trace the mapping between view columns defined within two different views.

6.1.3.1 View Column Structure

View Column Structure is the relationship between two view columns of different views that are governed by a specific view column structure process. If for example one view is reading three columns, First Name, Middle Initial, and Last Name, and another view is outputting Full Name, then there would be three View Column Structure rows. The objective in this case is to create a set of view column structure rows that conform to those contained in Figure 12.

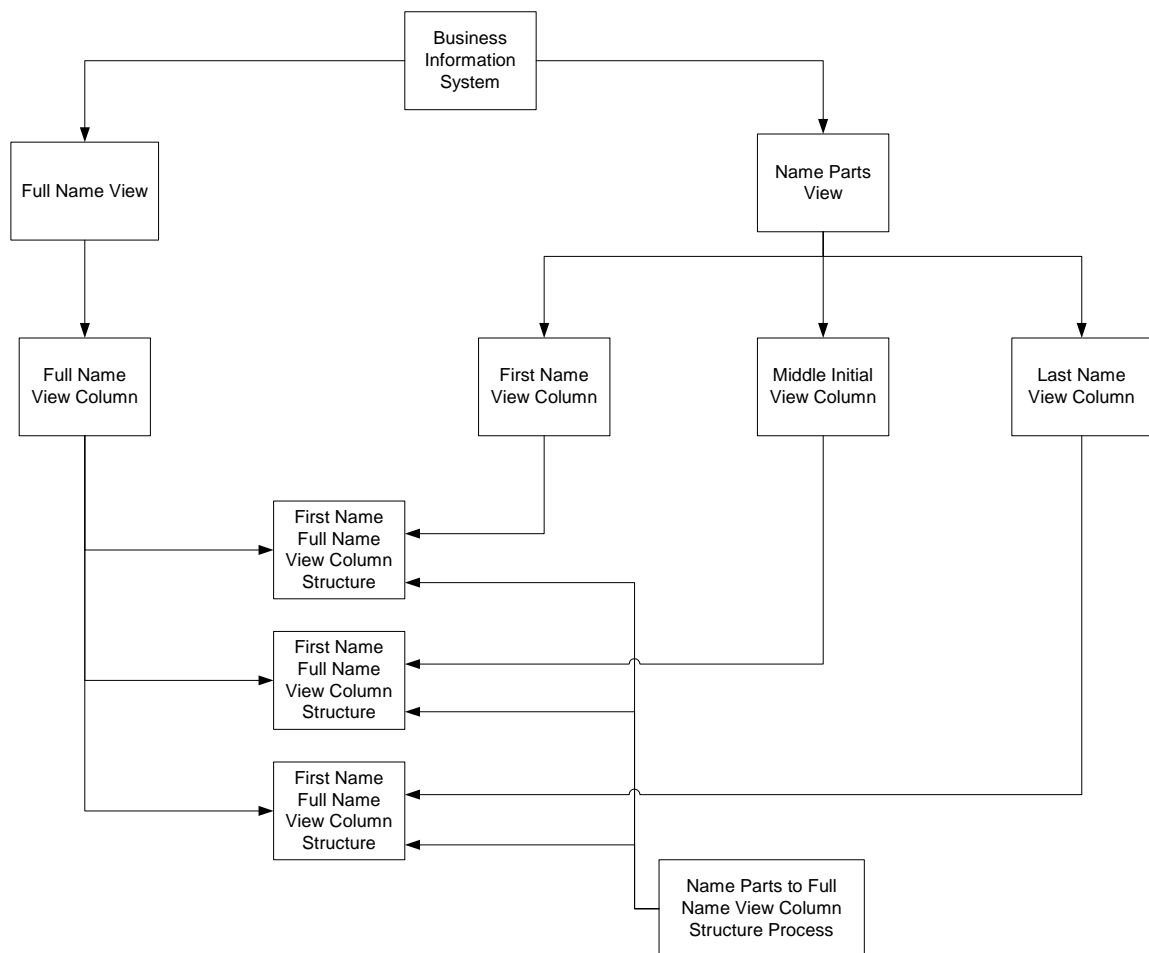


Figure 12. View Column Structure rows in support of relating view columns from different views.



Figure 13 shows the View Column Structure screen. In this particular example, the view, ODBC Customer Address consists of two view columns, Customer Full Name, and Customer Address Line 1. Needed is the Customer City-State-Zip. To insert that view column, The view column, press the Insert button and then Figure 14 is presented.

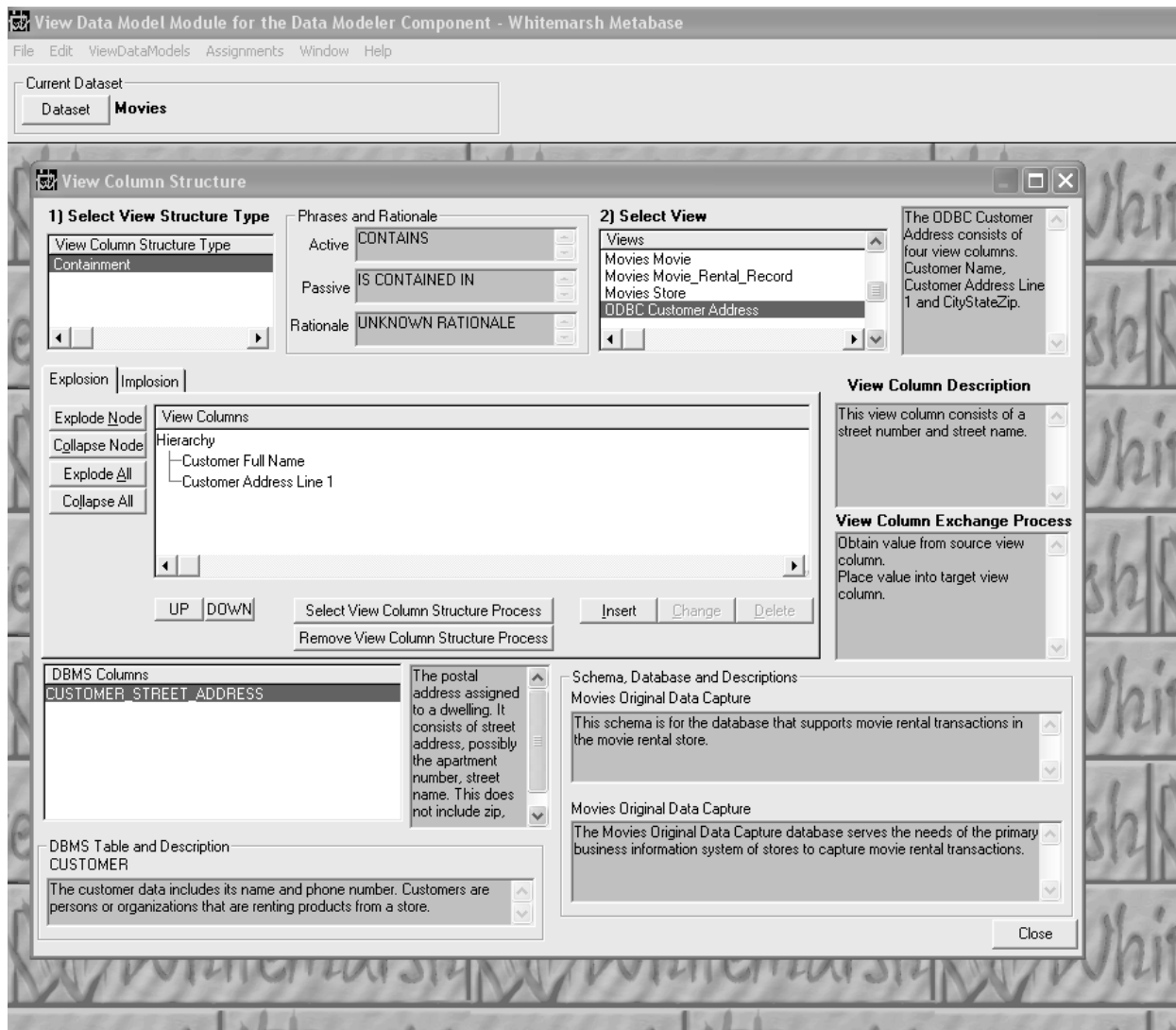


Figure 13. View Column Structures.



Highlight a view column. If a view column is highlighted from the same view as the column of the result column then the Select button is disabled and an error message is displayed. If a view column is highlighted that results in a recursion then an error message is also displayed. In this particular example, the view column, Customer CityStateZip is selected. Then press the Close button

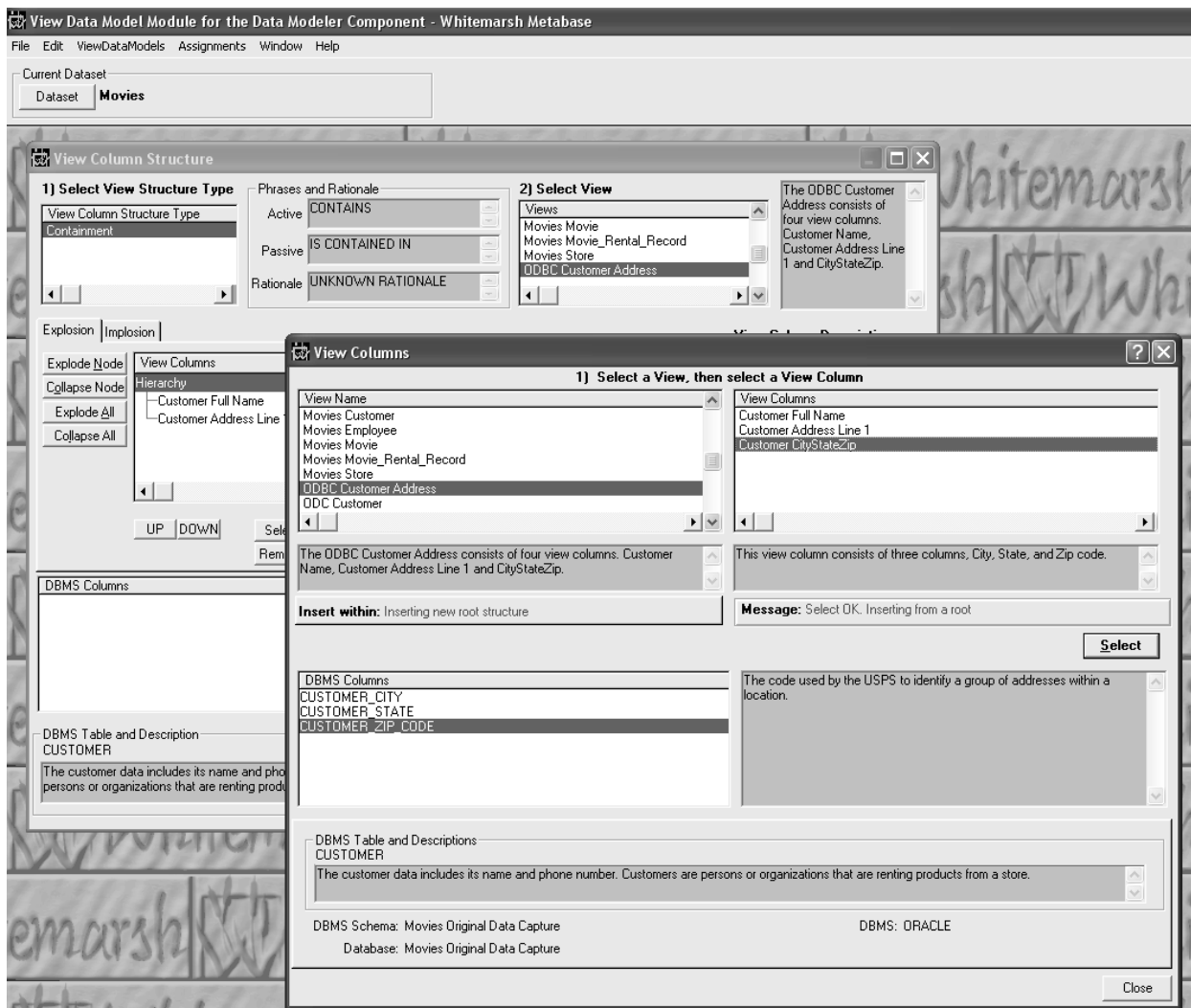


Figure 14. Selecting a View Column for a View Column Structure.



Once a inter-view column structure is completed, highlight the top of the structure and press the Select View Column Structure Process button. A set of processes, like those in Figure 15 are displayed. Select the appropriate one and press the Select button.

The final step in this process is to reselect the view. At that point, all the appropriate changes are displayed.

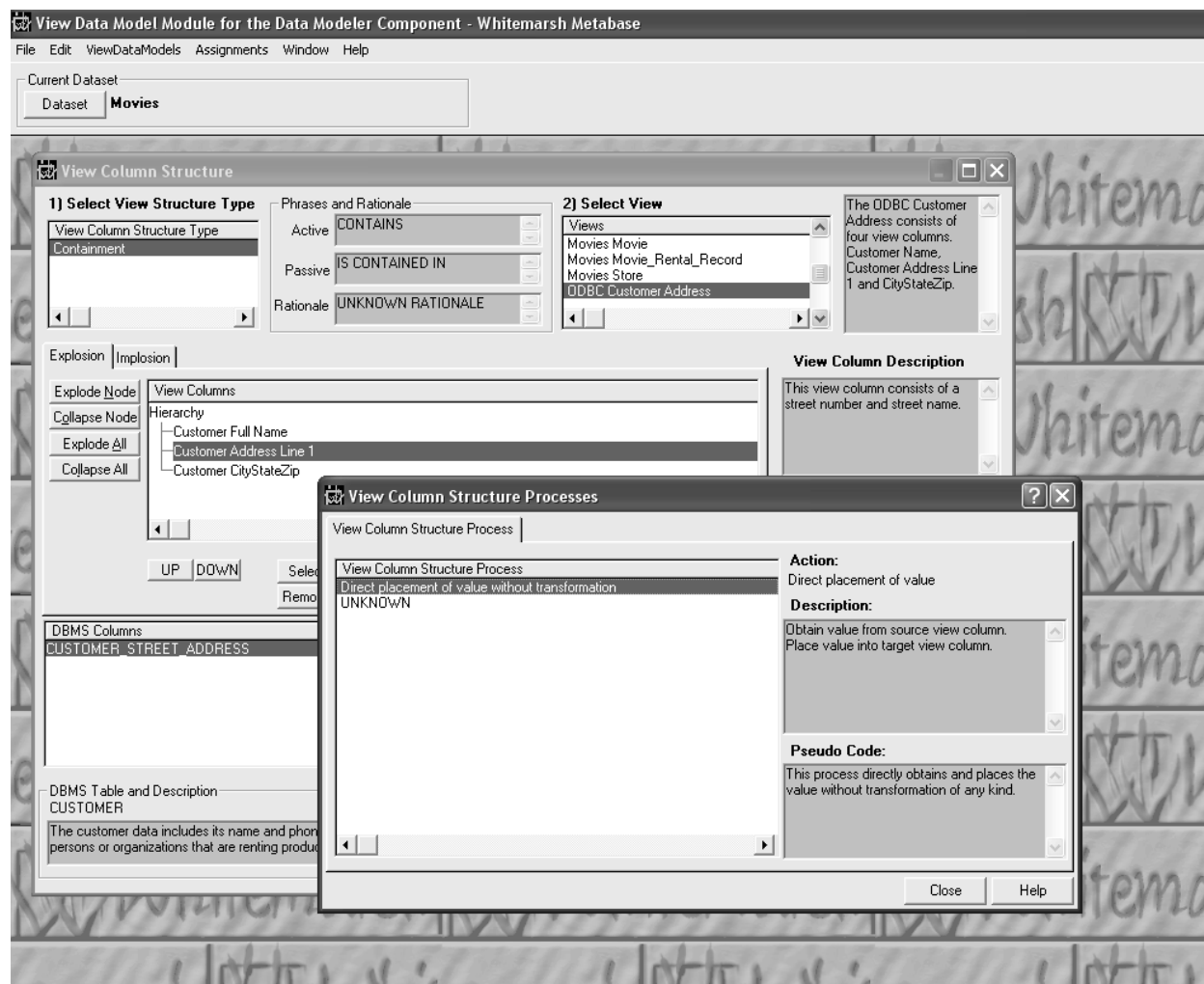


Figure 15. Selecting a View Column Process.



6.1.3.2 View Column Structure Process

View Column Structure process is the pseudo code that defines the relationship among all the view column referenced in a set of view column structure records. Figure 16 shows a list of the view column structure processes. To add, delete or change one press the appropriate button. On insert or change, Figure 17 is presented. Make the appropriate entries and close the screen.

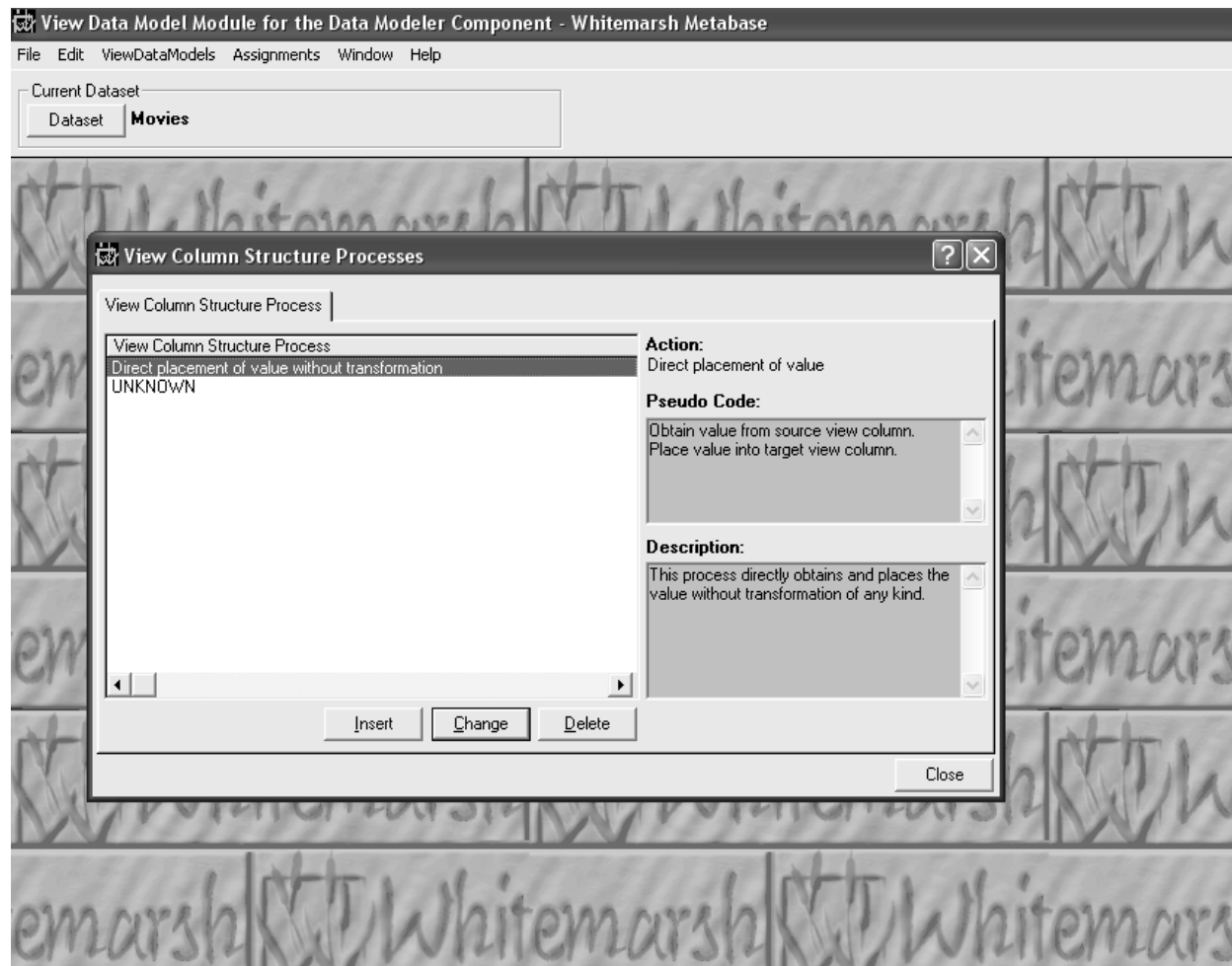


Figure 16. View Column Processes.



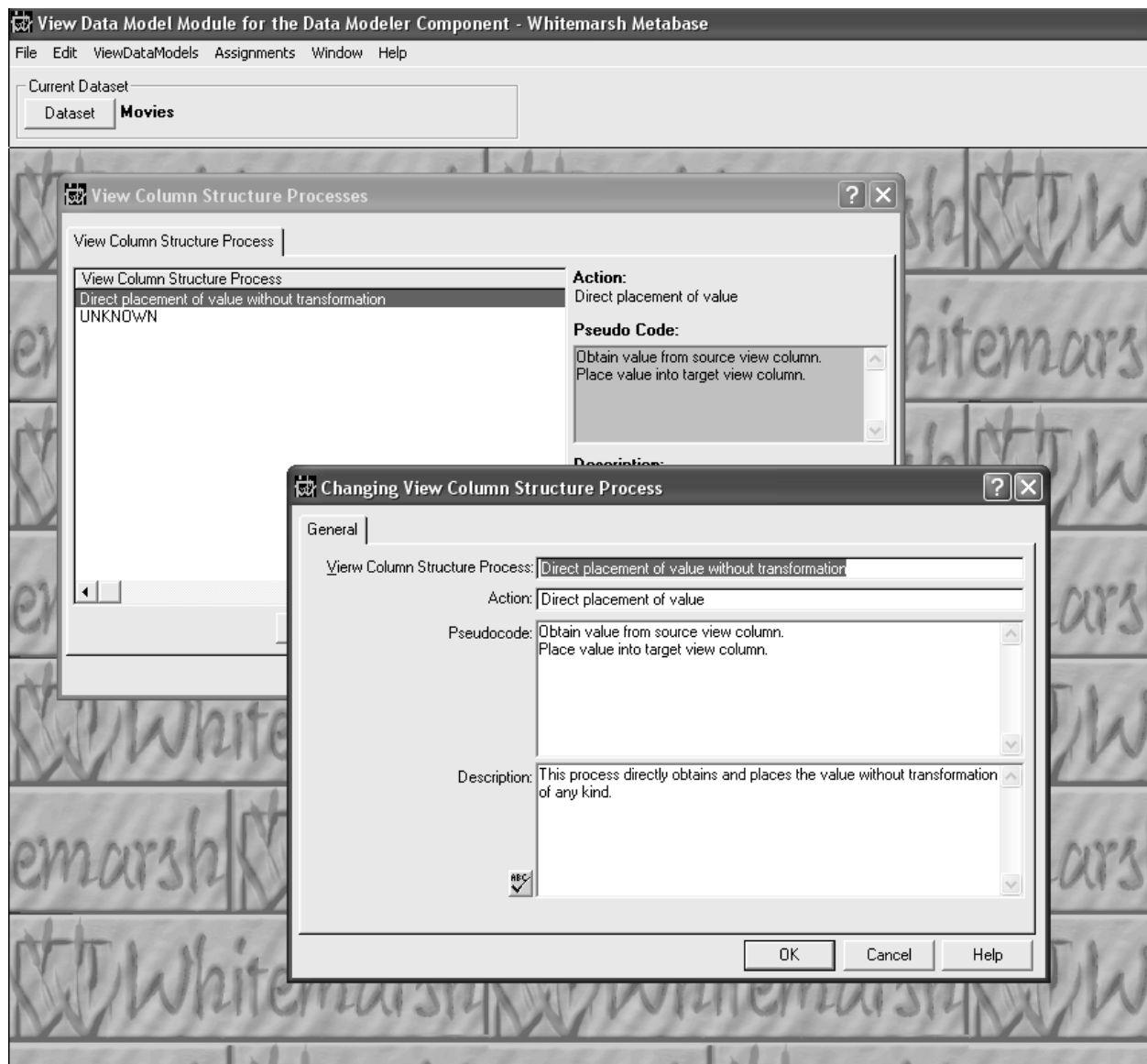


Figure 17. Updating a View Column Structure Process.



6.1.3.3 View Column Structure Type

A view column structure type represents the classification of structure among a set of view column structures. Figure 18 shows the set of view column structure types. The purpose is to classify the type of process that is to govern the transformation of the set of contained view columns into the result view column. Figure 19 presents the View Column Structure Type update screen.

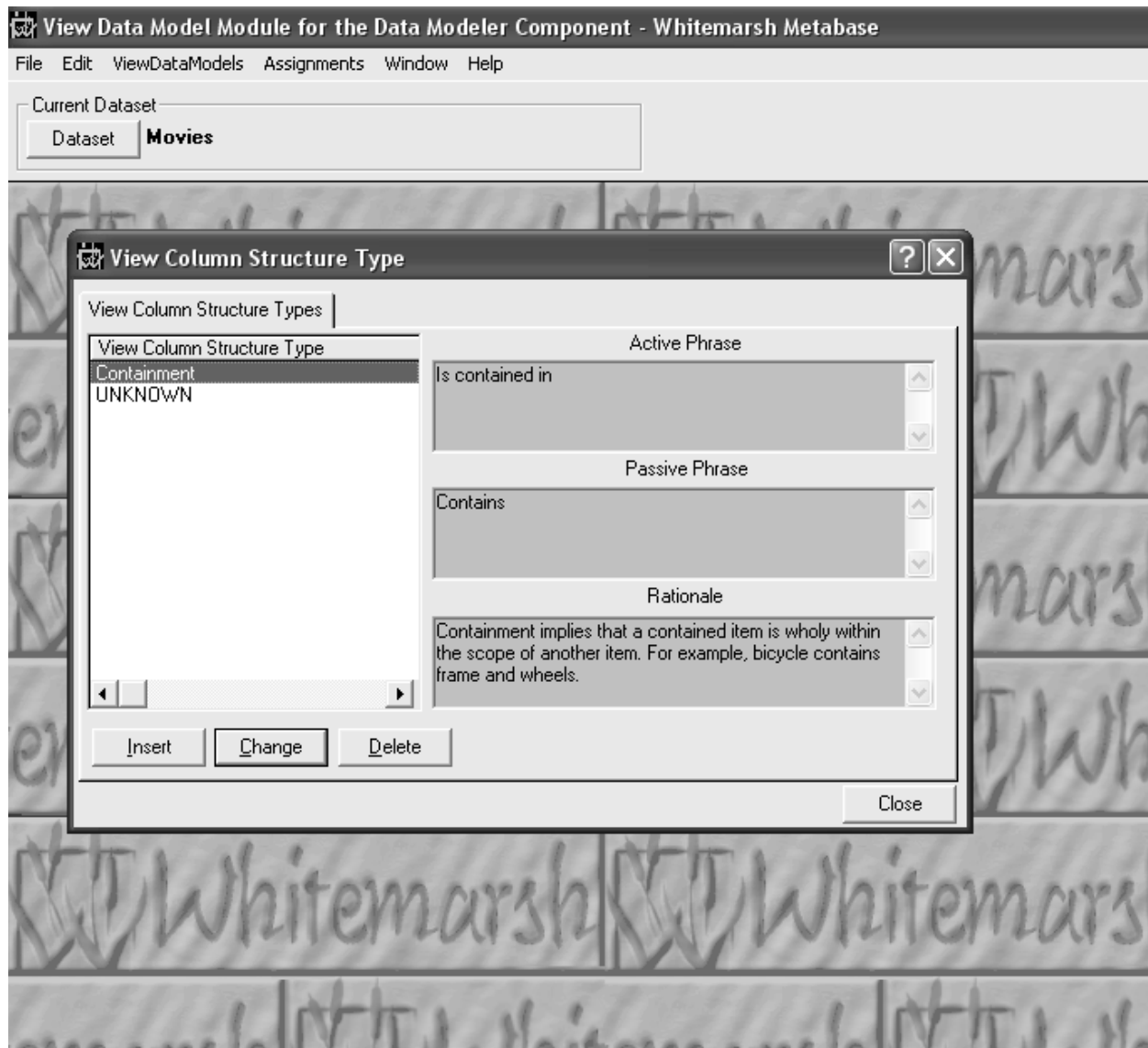


Figure 18. View Column Structure Types.



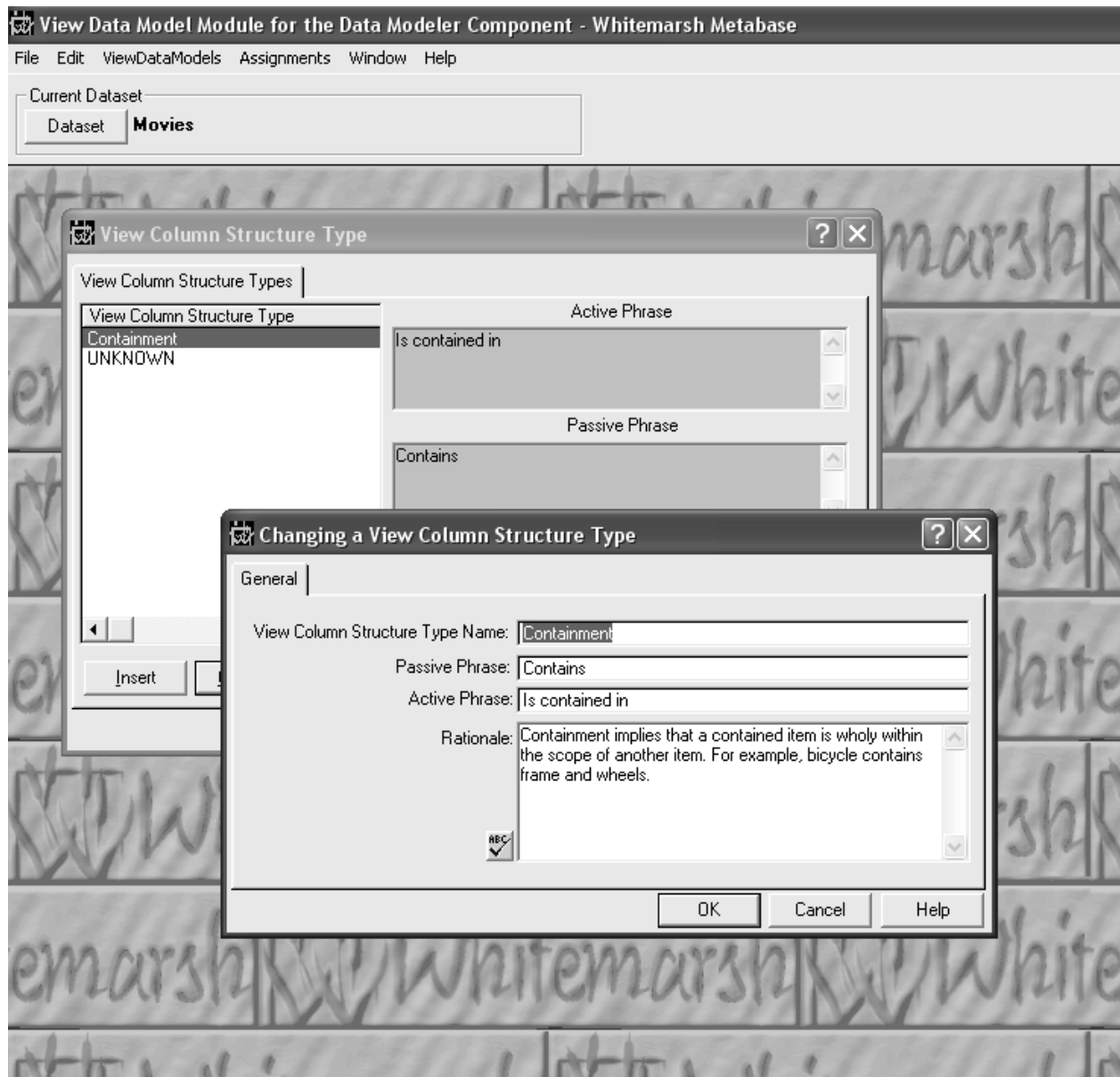


Figure 19. View Column Structure Type update screen.



6.2 Reports

Reports are accomplished through access to a particular metabase database instance through commercial report writers such as Crystal Reports. Whitemarsh provides about 100 such report templates for Crystal Report access from the Whitemarsh website.

