

Whitemarsh
Information Systems Corporation

Whitemarsh Metabase Data Modeler: Specified Data Model Users Guide

December 2007

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

Table of Contents

| | | |
|-----------|---|----|
| 1 | Introduction | 1 |
| 2 | Software Installation | 2 |
| 3 | Database Design | 2 |
| 4 | Reference Data | 4 |
| 5 | Operation | 4 |
| 6 | Process Model | 7 |
| 6.1 | Fact Data | 8 |
| 6.1.1 | Subjects & Entities | 8 |
| 6.1.1.1 | Subjects | 8 |
| 6.1.1.2 | Entities | 11 |
| 6.1.2 | Attributes | 14 |
| 6.1.2.1 | Creation | 14 |
| 6.1.2.2 | Maintenance | 17 |
| 6.1.2.2.1 | Maintain Attributes | 17 |
| 6.1.2.2.2 | Maintain Attribute Value Domains | 22 |
| 6.1.2.2.3 | Maintain Attribute Meta Category Values | 22 |
| 6.1.2.3 | Data Hierarchies | 24 |
| 6.1.3 | Keys | 26 |
| 6.1.3.1 | Primary | 26 |
| 6.1.3.1.1 | Primary Key Definition | 26 |
| 6.1.3.1.2 | Allocation of Attributes to the Primary Key | 29 |
| 6.1.3.2 | Foreign Keys | 30 |
| 6.1.3.3 | Candidate | 35 |
| 6.1.3.3.1 | Candidate Key Definition | 35 |
| 6.1.3.3.2 | Allocation of Attributes to the Candidate Key | 35 |
| 6.1.4 | Reverse Engineering | 39 |
| 6.1.4.1 | Reassigning Subjects to Subject | 39 |
| 6.1.4.2 | Reassigning Entities to Subject | 39 |
| 6.1.4.3 | Reassigning Entities to Entities | 39 |
| 6.1.4.4 | Reassigning Attributes to Data Elements | 40 |
| 6.1.4.5 | Reassigning Attributes to Entities | 40 |
| 6.1.4.6 | Synchronize Attributes to Local Definitions | 40 |
| 6.1.4.6 | Promoting Attribute to Data Element | 47 |
| 6.1.4.7 | Removing Attribute Meta Category Values | 48 |
| 6.1.5 | SQL DDL | 49 |
| 6.1.5.1 | Export | 49 |



| | |
|--|----|
| 6.1.5.2 Import SQL DDL | 55 |
| 6.1.5.1 Creating the SQL DDL Text File | 55 |
| 6.1.5.2 Importing the SQL DDL | 55 |
| 6.1.6 Data Model Trees | 63 |
| 6.2 Reports | 66 |



List of Figures

| | |
|---|----|
| Figure 1. Specified data model meta entity model. | 3 |
| Figure 2. Login screen. | 6 |
| Figure 3. Subjects. | 9 |
| Figure 4. Subject update screen. | 10 |
| Figure 5. Entities | 12 |
| Figure 6. Entity update screen. | 13 |
| Figure 7 Creating a single attribute. | 15 |
| Figure 8. Create many attributes. | 16 |
| Figure 9. Attributes within Subjects and Entities. | 17 |
| Figure 10. Attribute update screen. | 19 |
| Figure 11. Restriction on updating a Foreign Key Attribute. | 20 |
| Figure 12. List of selectable data elements for the creation of an attribute. | 21 |
| Figure 13. Assigning an attribute value domain. | 22 |
| Figure 14. Assigning an attribute meta category value. | 23 |
| Figure 15. Attribute cross reference from Data Element through to DBMS Columns. | 25 |
| Figure 16. List of Primary Keys. | 27 |
| Figure 17. Primary Key update screen. | 28 |
| Figure 18. Primary Key Attribute assignment screen. | 29 |
| Figure 19. Foreign Keys. | 30 |
| Figure 20. Foreign Key update screen. | 31 |
| Figure 21. Selecting a Primary Key for Foreign Key selection. | 32 |
| Figure 22. Selecting the target Entity for Foreign Key creation. | 33 |
| Figure 23. Entering a verb phrase to create Foreign Key Name. | 34 |
| Figure 24. Candidate Keys. | 36 |
| Figure 26. Allocating attributes to a Candidate Key. | 38 |
| Figure 27. Reassign Subjects to Subjects. | 41 |
| Figure 28. Reassign Entity to Subject. | 42 |
| Figure 29. Reassigning Entity to Entity. | 43 |
| Figure 30. Reassign Attribute to Data Element. | 44 |
| Figure 31. Reassign Attribute to Entity. | 45 |
| Figure 32. Synchronize Attribute Local Definitions. | 46 |
| Figure 33. Promote Attribute to Data Element. | 47 |
| Figure 34. Remove Attribute Meta Category Values. | 48 |
| Figure 35. SQL DDL Export screen. | 50 |
| Figure 36. Displayed data model. | 51 |
| Figure 37. Selected SQL DDL output file for export. | 52 |
| Figure 38. Generation message for creating SQL DDL for the Subject: Person. | 53 |
| Figure 39. Displayed SQL DDL for the Person Subject area. | 54 |
| Figure 40. List of Subjects. | 57 |
| Figure 41. SQL DDL Import screen. | 58 |
| Figure 42. Selecting the SQL DDL file for importing. | 59 |



| | |
|--|----|
| Figure 43. Viewing SQL DDL that is to be imported. | 60 |
| Figure 44. Entities and Attributes of Imported Movies SQL DDL. | 61 |
| Figure 45. Viewing the SQL Import Log File. | 62 |
| Figure 46. Selecting a root entity for displaying a data model tree. | 64 |
| Figure 47. Movies Subject area Data Model Tree display. | 65 |



1 Introduction

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 relationships 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.

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.

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:

- Attribute
- Attribute & meta category value
- Entity
- Entity candidate key
- Entity candidate key & attribute
- Entity foreign key
- Entity foreign key & attribute
- Entity primary key
- Entity primary key & attribute
- Subject

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

- Attributes are the manifestation of the semantics of a data element within an entity of a subject. Attributes may have additional semantics that further refine it within the context of the data element.
- Attribute value domain are collections of a more refined set of value domains for specific attributes. Attribute value domains are a subset of data element value domains.
- Attribute & meta category values are the relationship that exists between an attribute and its set of assigned meta category value semantics. These assigned semantics are always a subset of those assigned to the attributes “parent” data element through processes within the data modeler.



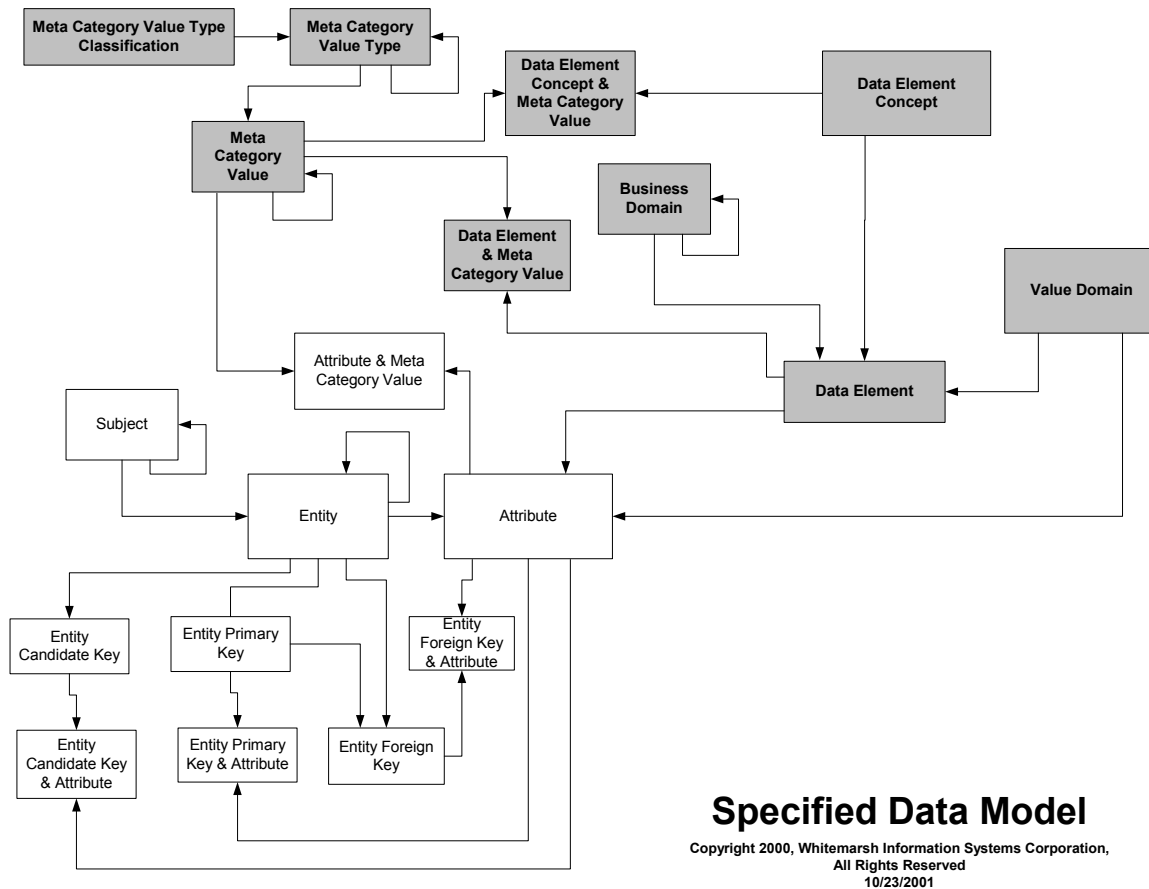


Figure 1. Specified data model meta entity model.

- An entity is intended to be a well defined expression of one policy within a subject area. The collection of all the entities within a subject area should define the complete set of policy for that area. Some entities and even some subject areas may never be represented within implemented data models. Additionally, some attributes within an entity may never be employed. Entities may be subtyped.
- Entity candidate keys represent a collection of attributes within an entity that when their values are collectively employed would result in the retrieval or update of a single row of data for that entity if that entity had actually been a table. There may be multiple candidate keys within an entity. Attributes of candidate keys are not allowed to overlap each other or the entity's primary key.
- Entity candidate key & attributes are the relationship between an entity candidate key and the attributes that comprise the key. The attributes exist within a specified sequence. Candidate key attributes are not allowed to include any attributes within the entity's primary key.



- Entity foreign keys represent a related entity's primary key. The name of the foreign key should match closely the relationship that the key is to represent. The attributes of the foreign key should be able to be deleted entirely from the entity without any loss of policy. The attributes of the foreign key are not allowed to overlap the attributes of the entity's primary key. In addition to the foreign key's attributes there are additional rules governing inserts, updates, and deletes.
- Entity foreign key & attributes are the relationship between an entity foreign key and the attributes that comprise the key. The attributes exist within a specified sequence. Foreign key attributes are not allowed to include any attributes within the entity's primary key.
- Entity primary keys represent a collection of attributes within an entity that when their values are collectively employed would result in the retrieval or update of a single row of data for that entity if that entity had actually been a table. There can only be one primary key within an entity. Attributes of primary key are not allowed to overlap each other or the entity's candidate key.
- Entity primary key & attribute are the relationship between an entity candidate key and the attributes that comprise the key. The attributes exist within a specified sequence. Candidate key attributes are not allowed to include any attributes within the entity's primary key.
- Subjects represent a cohesive area of discipline within the enterprise. Subjects can be hierarchical and likely match the essential resources of the business. Specified data models are cast within the domain of a subject area.

4 Reference Data

There is no reference data in the specified 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.



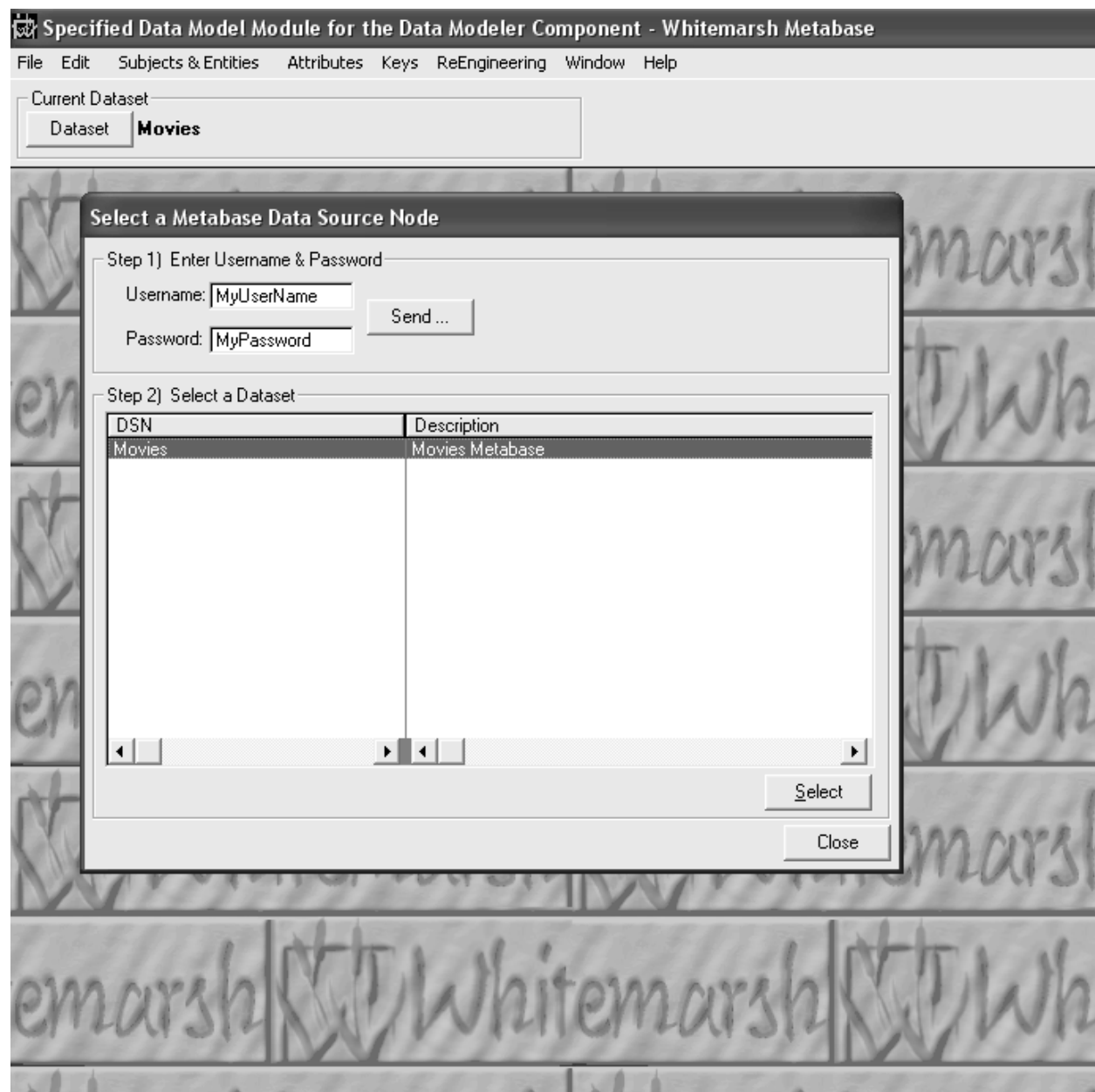


Figure 2. Login screen.



6 Process Model

The specified 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:

- Subjects & Entities
- Attributes
- Keys

Each menu item contains as appropriate, nested subordinate menu items. The complete menu is presented 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.

| |
|---|
| <ul style="list-style-type: none">-- Subjects Entities<ul style="list-style-type: none">-- Subject-- Entity-- Data Model Tree-- SQL DDL<ul style="list-style-type: none">-- Import-- Export-- Attributes<ul style="list-style-type: none">-- Creation One Attribute-- Create Many Attributes-- Data Hierarchies-- Maintenance<ul style="list-style-type: none">-- Maintain Attributes-- Attribute Value Domains-- Attribute Meta Category Value-- Keys<ul style="list-style-type: none">-- Entity Primary Keys-- Entity Primary Key Attributes-- Entity Foreign Keys-- Entity Candidate Keys-- Entity Candidate Key Attributes |
|---|



- ReEngineering
 - Reassign Subjects to Subjects
 - Reassign Entities to Subjects
 - Reassign Entities to Entities
 - Reassign Attributes to Data Elements
 - Reassign Attributes to Entities
 - Synchronize Attribute Local Definitions
 - Promote Attribute to Data Element
 - Remove Attribute Meta Category Values

Menu for Specified Data Model

6.1 Fact Data

The fact data consists of:

- Subjects & Entities
- Attributes
- Keys

6.1.1 Subjects & Entities

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

- Subject
- Entity
- Reverse Engineering
- SQL DDL

6.1.1.1 Subjects

Subjects with respect to the specified data model are expressions of enterprise policy within a defined area. Subjects areas can be hierarchical. Figure 3 presents a list of subjects.

If a entirely new subject is to be entered, highlight the top of the tree, “hierarchy” and then press Insert. A screen like Figure 4 is presented. The subject’s name, abbreviations, and description can then be entered. If a subordinate subject is to be created, highlight its parent and then press Insert. After the data is completely entered, press Enter. The new list of subjects then appears in alphabetical order.



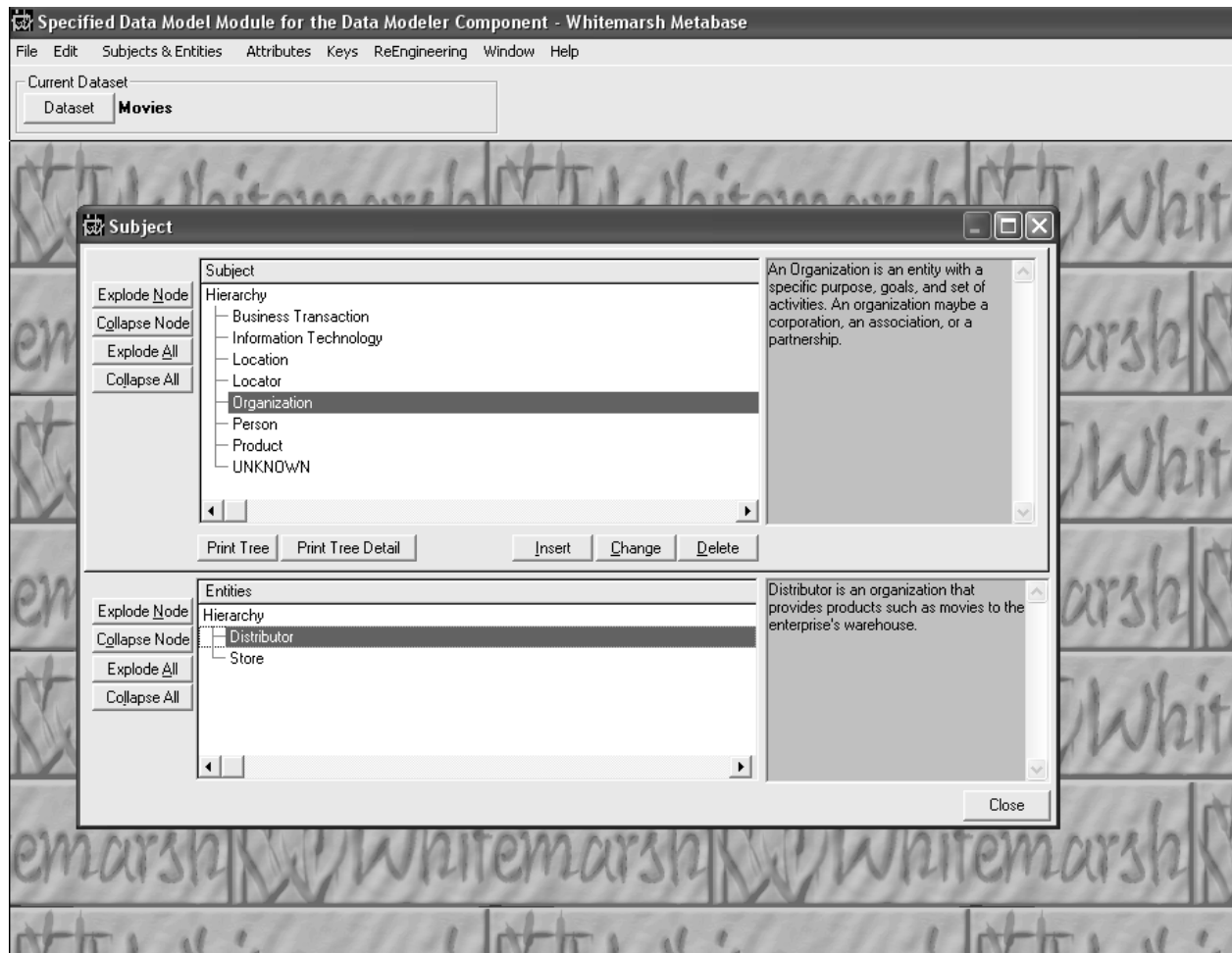


Figure 3. Subjects.



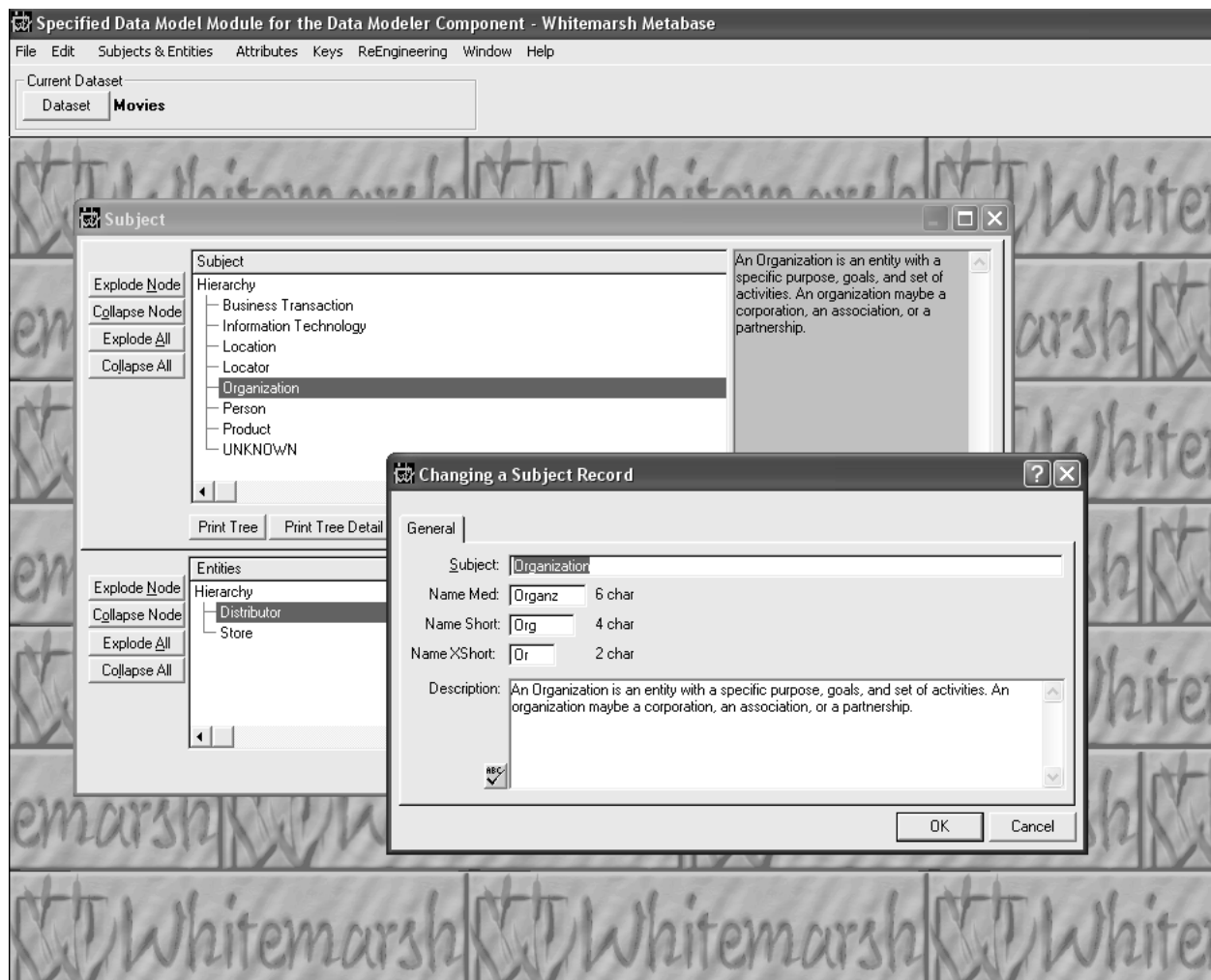


Figure 4. Subject update screen.



6.1.1.2 Entities

An entity is a self contained aspect of policy within a subject area. The attributes within the entity provide for the values that instantiate the policy. A collection of entities within the same subject area provide a comprehensive view of the subject's policy.

Figure 5 presents a list of entities. Within the subject, customer management, there are four entities. Within the highlighted address entity are a collection of attributes.

There are two buttons on this screen: Delete Entities for Selected Subject, and Auto Make Primary Key.

The first button, Delete Entities for Selected Subject, deletes all the entities for the subject. Deleted also are all associated attributes, keys, and attribute meta-category values). If any attribute of any of the entities within the subject are related to any column of any table in an Implemented Data Model then the delete operation will fail.

The second button, Auto Make Primary Key, creates a primary key for the entity of the form, <entity name> Primary Key. Additionally, an attribute is created, <entity name> ID.

Every entity has two definitions: local, and in context. The local definition should be very short, just a simple phrase. Do not make this a sentence. Do not start it with a capital letter, nor end it with a period. The contextual definition is automatically created through processes activated by the AutoDef button that is on the entity update screen. At the end of the AutoDef process, the local definition is included within the first sentence.

To add a new entity, highlight the containing subject and then press the Insert button. A screen like Figure 6 is displayed. The entity's name, abbreviations, and description can be added or changed. The attributes associated with an entity are addressed in Section 6.1.2.

This is the screen where you make the local definition and the contextual definition. As stated above, just make a simple phrase for the local definition. When the AutoDef button is pressed then all the contextual parts of the entities definition are gathered and employed in a more comprehensive description of the entity

If an entity subtype is to be created, highlight the appropriate parent entity and press Insert. Enter information as per Figure 6. The sub-typed entity then appears underneath its parent entity.



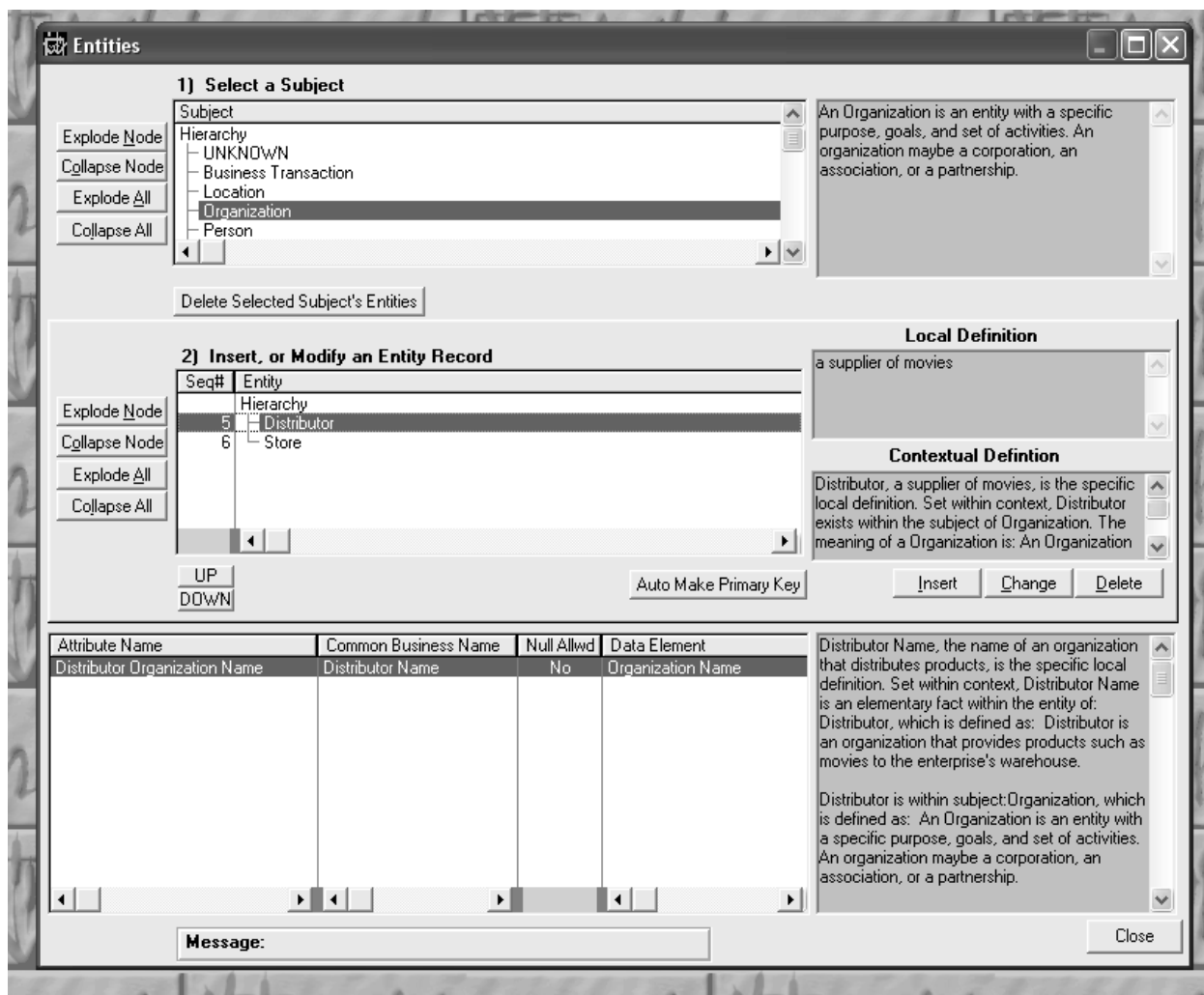


Figure 5. Entities



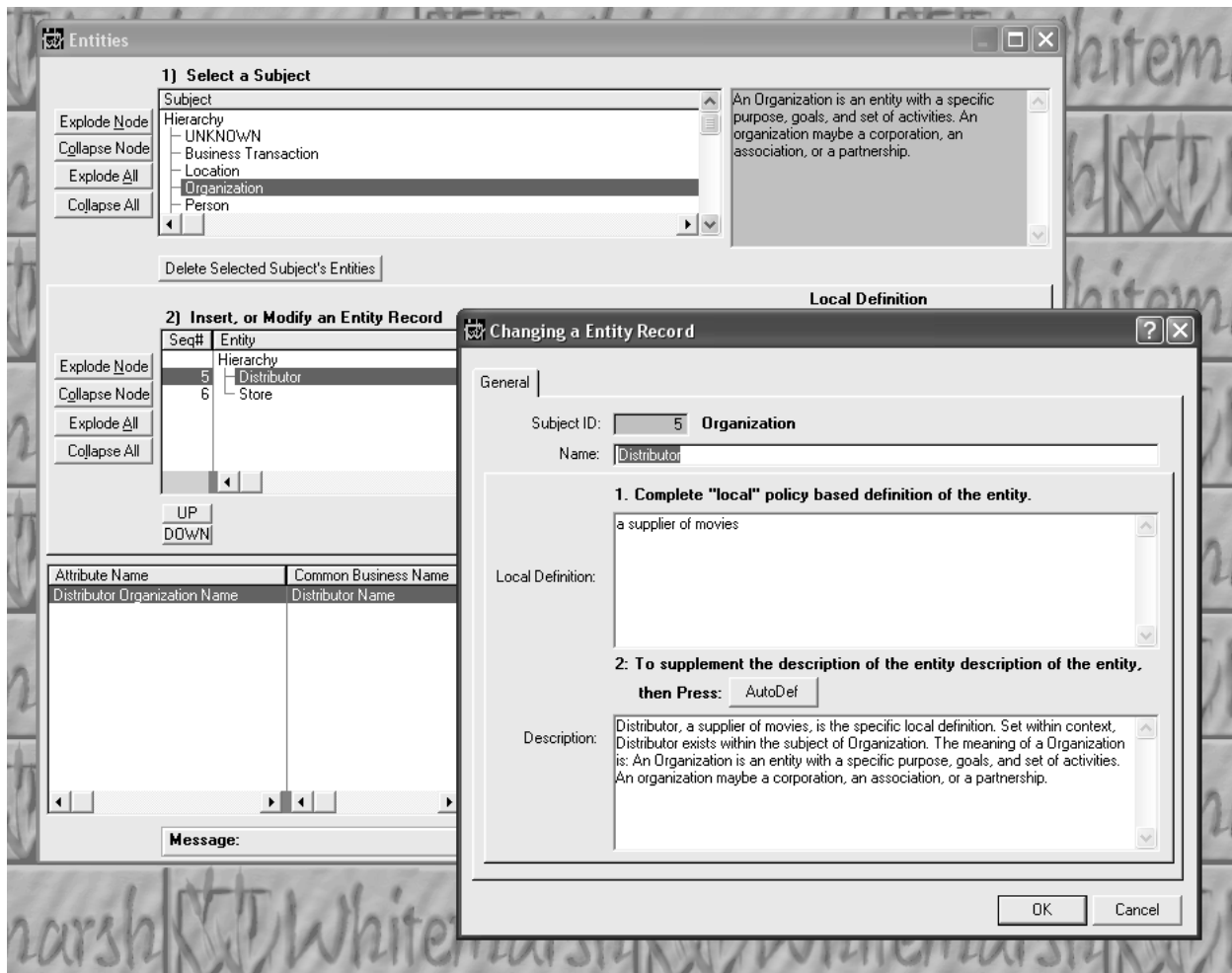


Figure 6. Entity update screen.



6.1.2 Attributes

Attributes are the value based characteristics of an entity. Included in the definition of attributes are:

- Creation
- Maintenance
- Data Hierarchies

6.1.2.1 Creation

Figure 7 presents the screen for creating a multiple attributes within a single entity. From the point of view of the specified data model, an attribute is really an “intersection record” between data element and entity. First highlight the subject then the entity. Tag the appropriate entity. Then highlight the business domain, and then tag one or more data elements. Once an entity and one or more data elements are tagged, press the Build button. The underlying process then creates attribute table rows within the highlighted entity. Each newly created attribute is mapped to the previously tagged data element. The complete set of attributes for the entity are then displayed in the bottom window.

Figure 8 presents the screen for creating multiple attributes, one each for the tagged entities. Select the business domain and then the data element. On the right side, select the subject and then tag the entities. You can tag multiple entities across multiple subjects. Once the tagging is accomplished, press the Build button. The attributes are created as above.



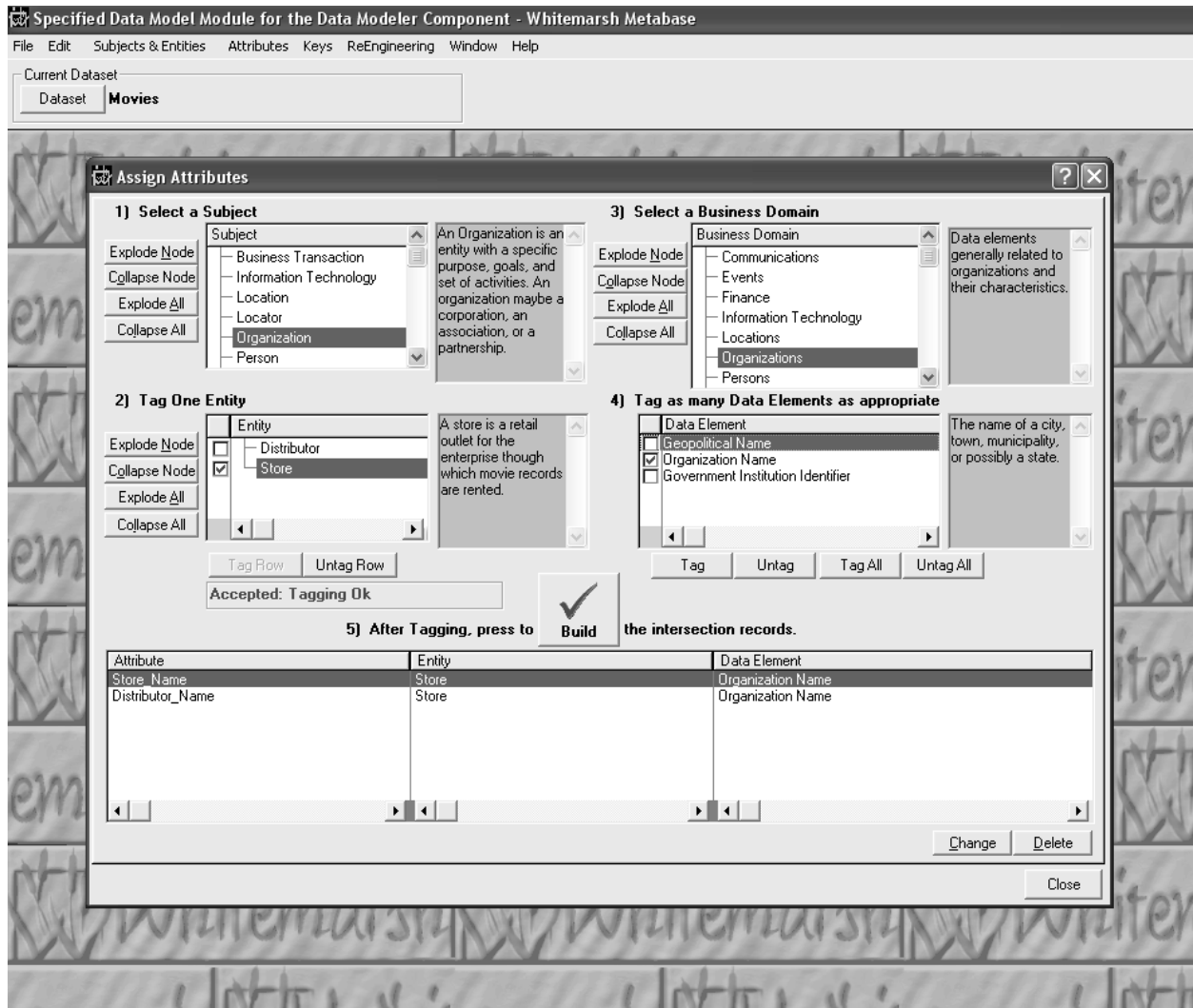


Figure 7 Creating a single attribute.



6.1.2.2 Maintenance

Included in the maintenance of attributes are:

- Maintaining attributes
- Maintaining attribute value domains
- Maintaining attribute meta category values

6.1.2.2.1 Maintain Attributes

Maintenance of attributes starts with a list of attributes. Figure 9 shows the list of attributes for highlighted subject and entity. Below the highlighted attribute are all the meta category values that have been assigned.

1) Select Subject

Subject

- UNKNOWN
- Business Transaction
- Location
- Organization**
- Person
- Product
- Locator
- Information Technology
- Movies

An Organization is an entity with a specific purpose, goals, and set of activities. An organization maybe a corporation, an association, or a partnership.

2) Select Entity

Entity

- Distributor**
- Store

Distributor, a supplier of movies, is the specific local definition. Set within context, Distributor exists within the subject of Organization. The meaning of a

For Keys Within Selected Entity
Refresh Foreign Key Attribute Names to Match Primary Key Attribute Names: **Press**

For all attributes within entity: **AutoDef**

3) Select Attribute then press Insert, Change or Delete

| Attribute Name | Common Business Name | User Set Name | Null Allowed |
|-------------------------------|----------------------|------------------|--------------|
| Distributor Organization Name | Distributor Name | Distributor Name | No |

Local Definition
the name of an organization that distributes products

Contextual Definition
Distributor Name, the name of an organization that distributes products, is the specific local definition. Set within context, Distributor Name is an elementary fact within the entity of: Distributor, which is defined

| Class | MCV Type Name | Meta Category Value |
|-------|---------------|---------------------|
|-------|---------------|---------------------|

Insert Change Delete

Close

Figure 9. Attributes within Subjects and Entities.



There are three buttons on this screen of special interest:

- Replace Underscores with Blanks
- For Pkeys within Selected Entity.... Press
- For all attributes within entity...AutoDef

In the event that you have imported SQL DDL from some outside source into the Specified Data Model, then underscores are inappropriate. Blanks are much better because at this level of specification, there models are just conceptual versus bound to a particular DBMS. Underscores are required for the DBMS-based models and the metabase accomplishes that for you automatically. Pressing Replace Underscores with Blanks does exactly what its name says for all attributes of all entities within the Selected subject.

The second button, For Pkeys within Selected Entity.... Press, this refreshes all the attributes that participate in a Foreign Key to match those in the source entity's Primary Key (Pkey).

The final button, For all attributes within entity...AutoDef, this causes all the attributes in the Selected entity to have the AutoDef function execute. The AutoDef process takes the local definition of an entity and sets it within the context of all the meta objects that are the "parents" of the entity.

If an attribute is to be changed then press the Change button. Figure 10 is then presented. If the attribute is not within a foreign key then all the meta attributes of the attribute can be changed. If, however, the attribute is part of foreign key Figure 11 is displayed with the message that the only meta attribute that can be changed is its description.

New attributes can also be created through Figure 9 by pressing the Insert button. Figure 10 is the update screen. Via this approach the data element must be chosen. Tab through the zero within the data element entry field, and then Figure 12, the list of data elements is displayed. Highlight the business domain and then the appropriate data element. Once the appropriate data element is identified then press the Select button.

The entity is automatically chosen for you because it was within the context of a selected entity from Figure 9 that the insert-attribute button was pressed. The value domain cannot be chosen because its assignment is handled in the Attribute Value Domains menu item from the Attributes main menu item. For every attribute, enter a common business name. Then press the Reset button. Default names will then automatically appear in the Name, and User Set Name fields. Select if null (i.e., "don't know the value").



Figure 10 also shows the UnAbbrev process. In the event that an SQL data definition language stream is imported, the physical names, that is abbreviated may be what is imported. For example, Empl_Hr_Dt. Under this update process, once the Business Domain has been chosen via the Select Abbreviation Business Domain button, the UnAbbrev button can be pressed. If there has been a good quantity of abbreviations established on various business domains, then the physical name can be automatically translated to a logical name. If there are multiple logical names for a given physical abbreviation, a screen presents itself and an appropriate logical name can be chosen for a given word. In this example, the full name becomes, Employee Hire Date.

Finally, enter the Local Definition phrase. Keep it just a phrase without a capital letter at the start and no period or comma at the end. For a complete contextual definition, then press the AutoDef button.

After all this, press the OK button and the attribute has been fully defined.

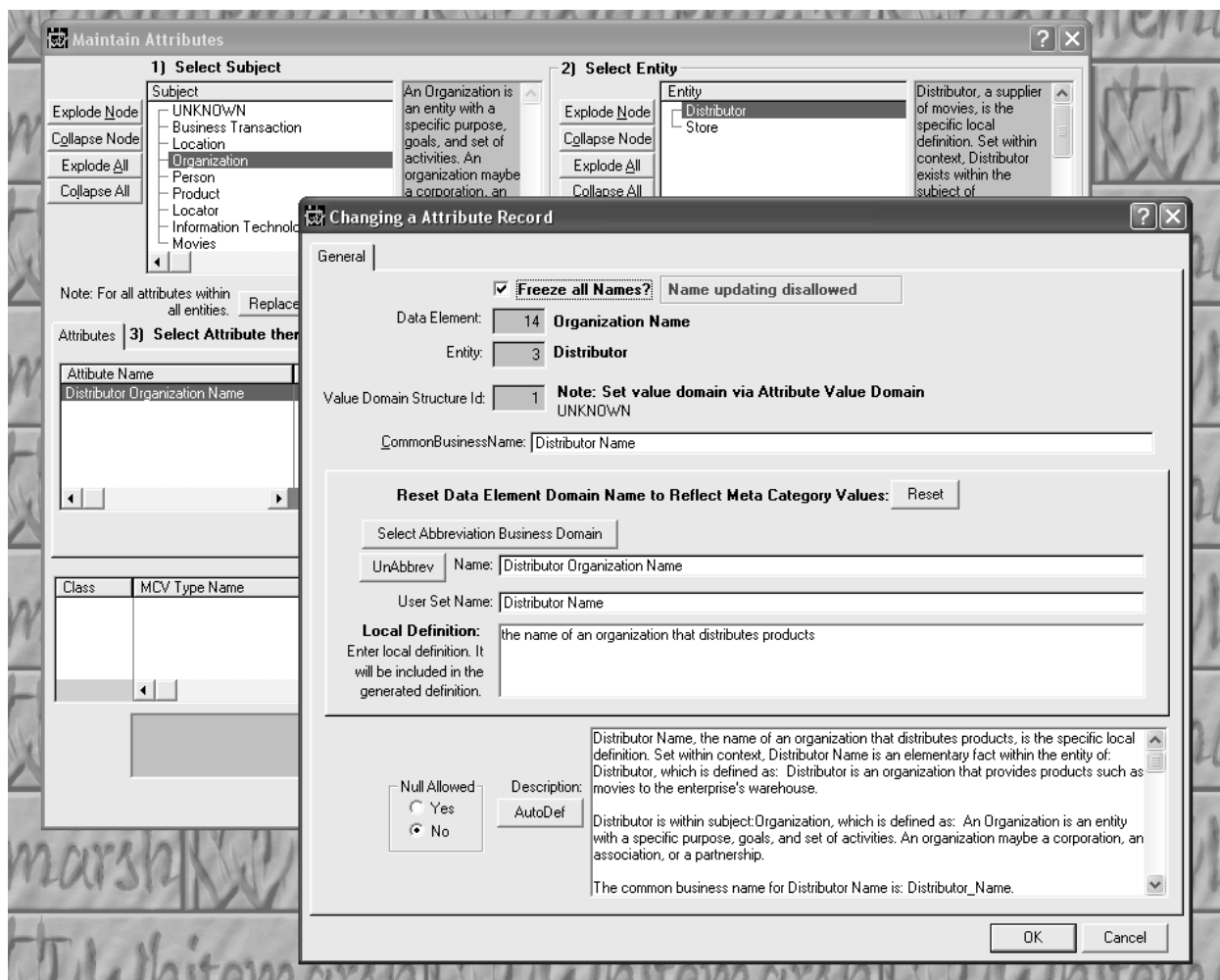


Figure 10. Attribute update screen.



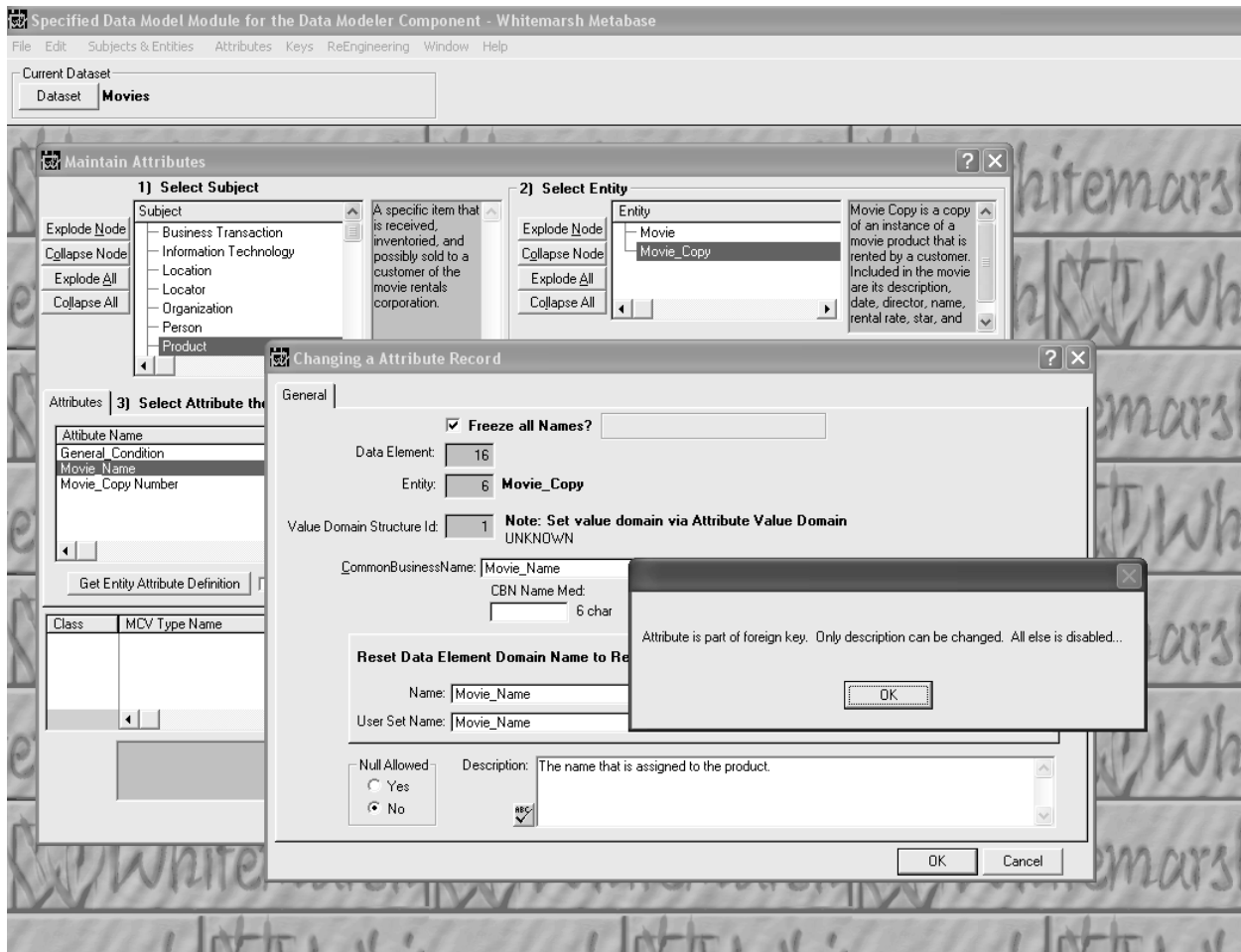


Figure 11. Restriction on updating a Foreign Key Attribute.



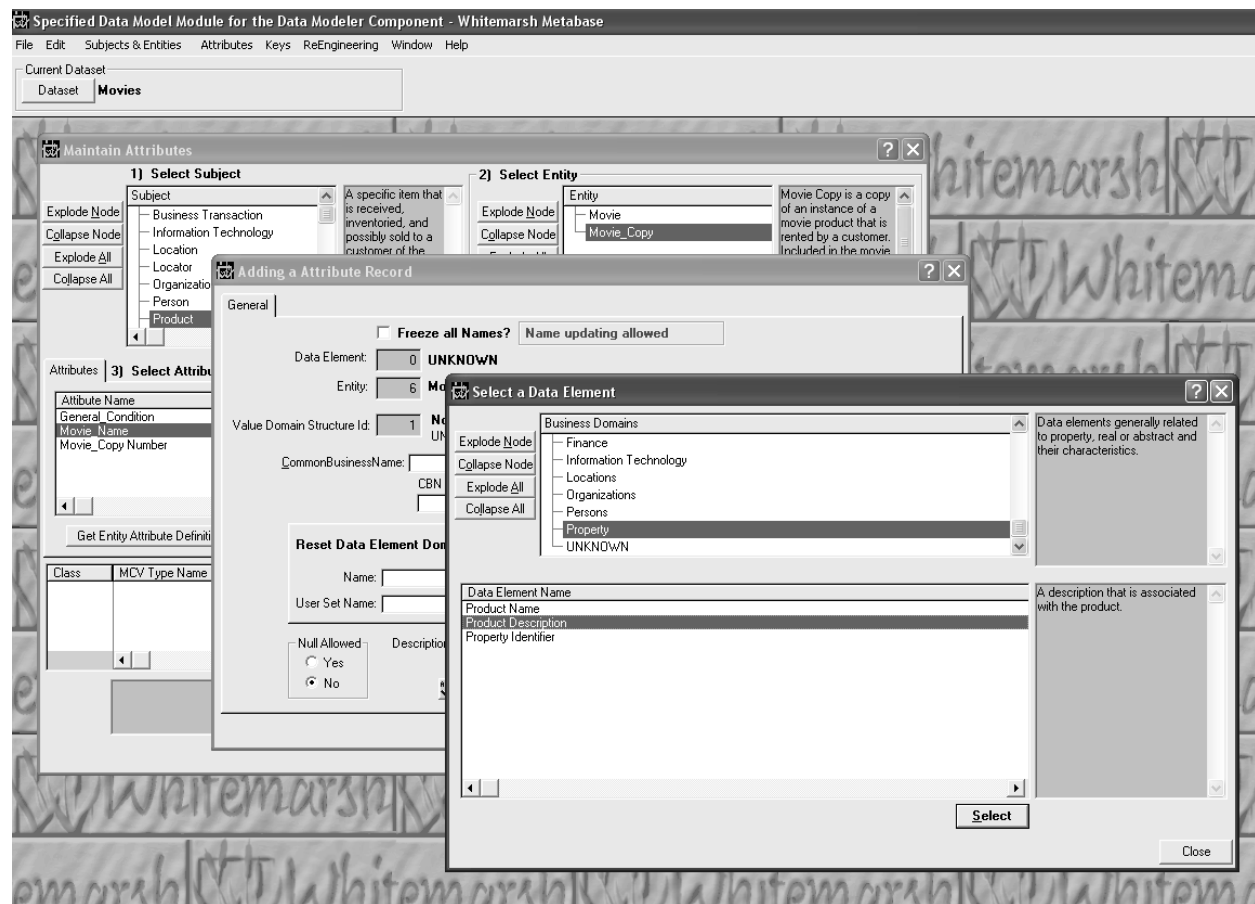


Figure 12. List of selectable data elements for the creation of an attribute.



6.1.2.2.2 Maintain Attribute Value Domains

Attribute value domains exist within the context of data element value domains which in turn exist within the context of data element concepts and value domains. Figure 13 presents the screen for viewing and then maintaining value domains. To see the specific value domain for an attribute, highlight the subject then entity. Attributes are then shown. Highlight a specific entity. If an attribute is mapped to a data element and if a data element has already been assigned a value domain, then the value domains able to be assigned to the attribute are shown. To see those value domains, highlight the Value Domain structure type. If a value domain can be assigned it is shown. If no valued domains are shown then none can be assigned. In this example, the value domain assigned to the data element has a “child” value domain. Because of that, the “child” value domain can be assigned to the attribute.

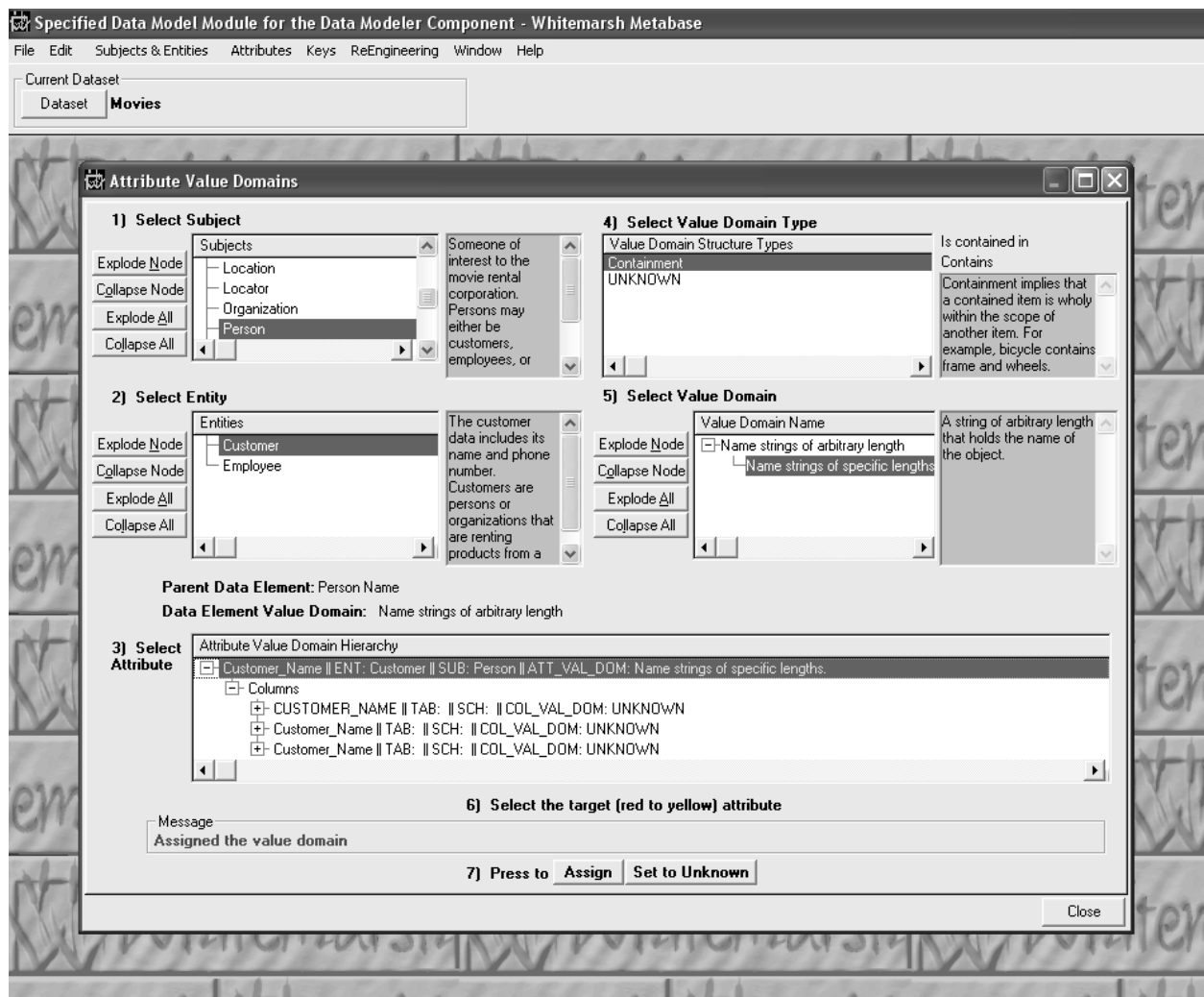


Figure 13. Assigning an attribute value domain.

6.1.2.2.3 Maintain Attribute Meta Category Values



The semantics of attributes are both inherited and assigned. They are inherited through its selected data element. Assigning semantics is illustrated in Figure 14. In this assignment process, the appropriate subject, entity, and attribute is highlighted. If there are any already assigned semantics then they are shown in the bottom browse window. The right two windows contain the set of meta category value types and type classes, and the meta category values for a particular highlighted meta category value. Only one meta category value from each meta category value type can be tagged. For example, it makes no sense to assign both estimated and projected to the same attribute.

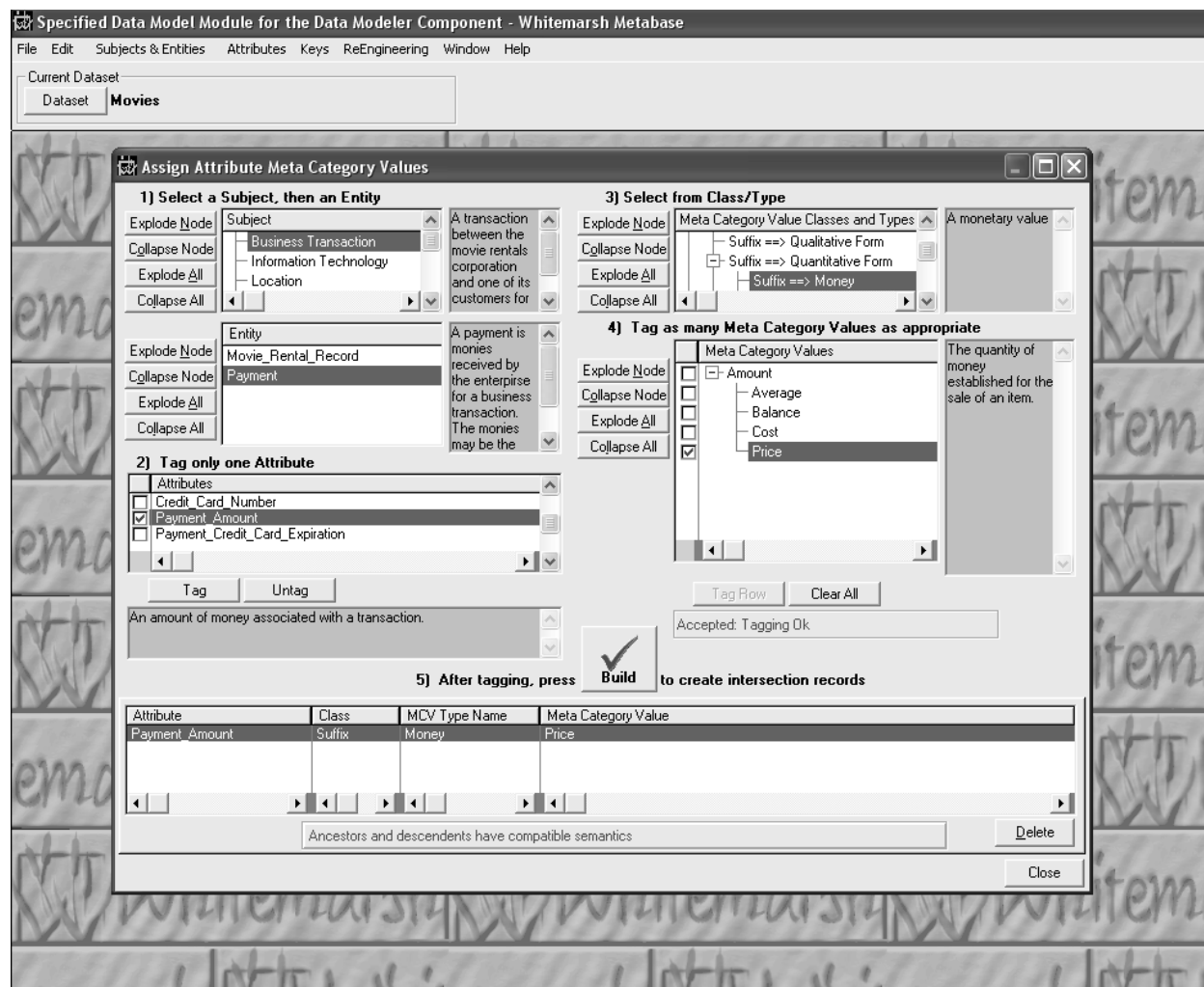


Figure 14. Assigning an attribute meta category value.



Suffix meta category values will then appear at after the attribute's name. Prefix meta category values will appear prior to the data element's common business name. The order of appearance is not arbitrary or able to be change by a metabase end-user. The order is specified within the table, meta category type. So, if the geographic meta category value type's sequence number is 2 and the temporal meta category value type's is 3 then regardless of the sequence of tagging, geographic meta category values will always be prefixed into the attribute's constructed name before those for the temporal meta category value type.

Once the meta category values are tagged and the build button is pressed, the meta category values are assigned to the attribute. The assignment process determines one by one that the tagged meta category values are semantic subsets of those already assigned either to a "parent" data element and data element domain, or to a "child" column. If the tagged semantic is a semantically acceptable then it is assigned. Otherwise the assignment for the offending semantic is rejected.

Once a semantic is assigned, if the attribute is employed in the creation of a column, or if a column is employed in the creation of a DBMS column, then none of the semantics for the attribute can be deleted. New ones can be added, but none can be deleted.

When ever semantics are added, the attribute's name remains the same until it is reset in the attribute update screen.

6.1.2.3 Data Hierarchies

A key value in the entire set of data modeler modules is the presentation of data hierarchies. For the specified data model, any highlighted attribute has only one parent data element and data element domain. Hence they are shown above the traditional browse.

As a browse progresses from one attribute to the next the respective data element and data element domain changes as well as the related set of implemented data model columns and then operational data model DBMS columns. If the IDM column browse is traversed then the ODM DBMS columns change as well.



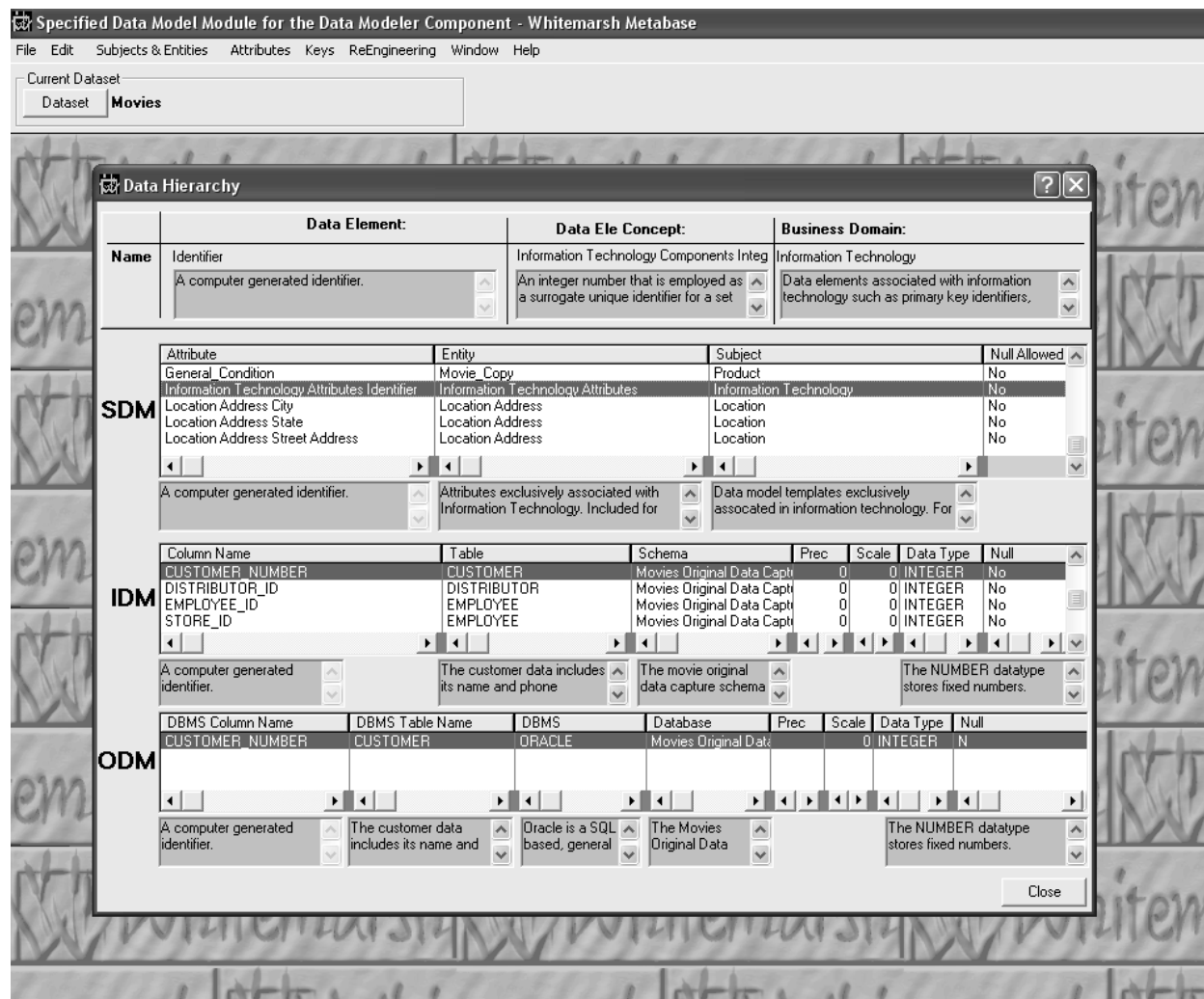


Figure 15. Attribute cross reference from Data Element through to DBMS Columns.



6.1.3 Keys

Entities are accessed through the use of keys. The keys included in the specified data model include:

- Primary
- Foreign
- Candidate

6.1.3.1 Primary

Included in the definition of a complete primary key is:

- Primary key definition
- Allocation of attributes to the primary key

6.1.3.1.1 Primary Key Definition

A primary key of an entity is a set of one or more attributes that represent values that when employed result in only one selected row. Figure 16 shows the current set of primary keys. There can, of course, only be one primary key for each entity. To see the attributes assigned to a particular primary key, highlight the appropriate subject, and then entity. To add a primary key if there is none for an entity press Insert.

If a primary key is already defined then an error message is displayed. If the primary key has already been used as the basis of relationships with other tables (that then has the primary key manifest as a foreign key) then a severe warning message is given before the delete operation is permitted to continue. If a delete operation is tried and if the primary key attribute is employed as a column in the implemented data model the operation fails.

If the Insert or change action succeeds, the Figure 17 is presented. On an Insert, the name is automatically constructed as the concatenation of the entity name and the string, "Primary Key." The name can be changed. In addition to the name a description can be added or changed.



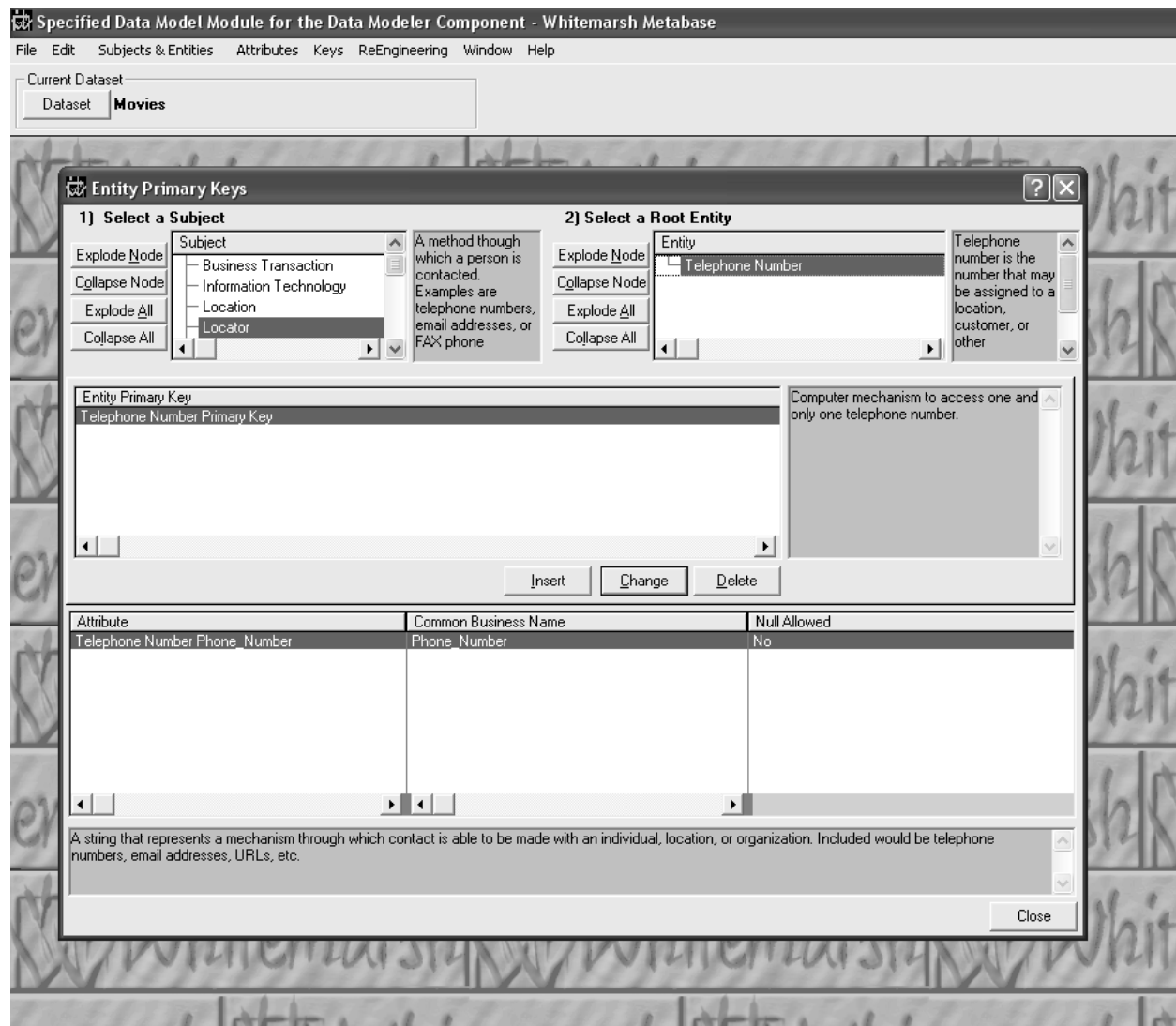


Figure 16. List of Primary Keys.



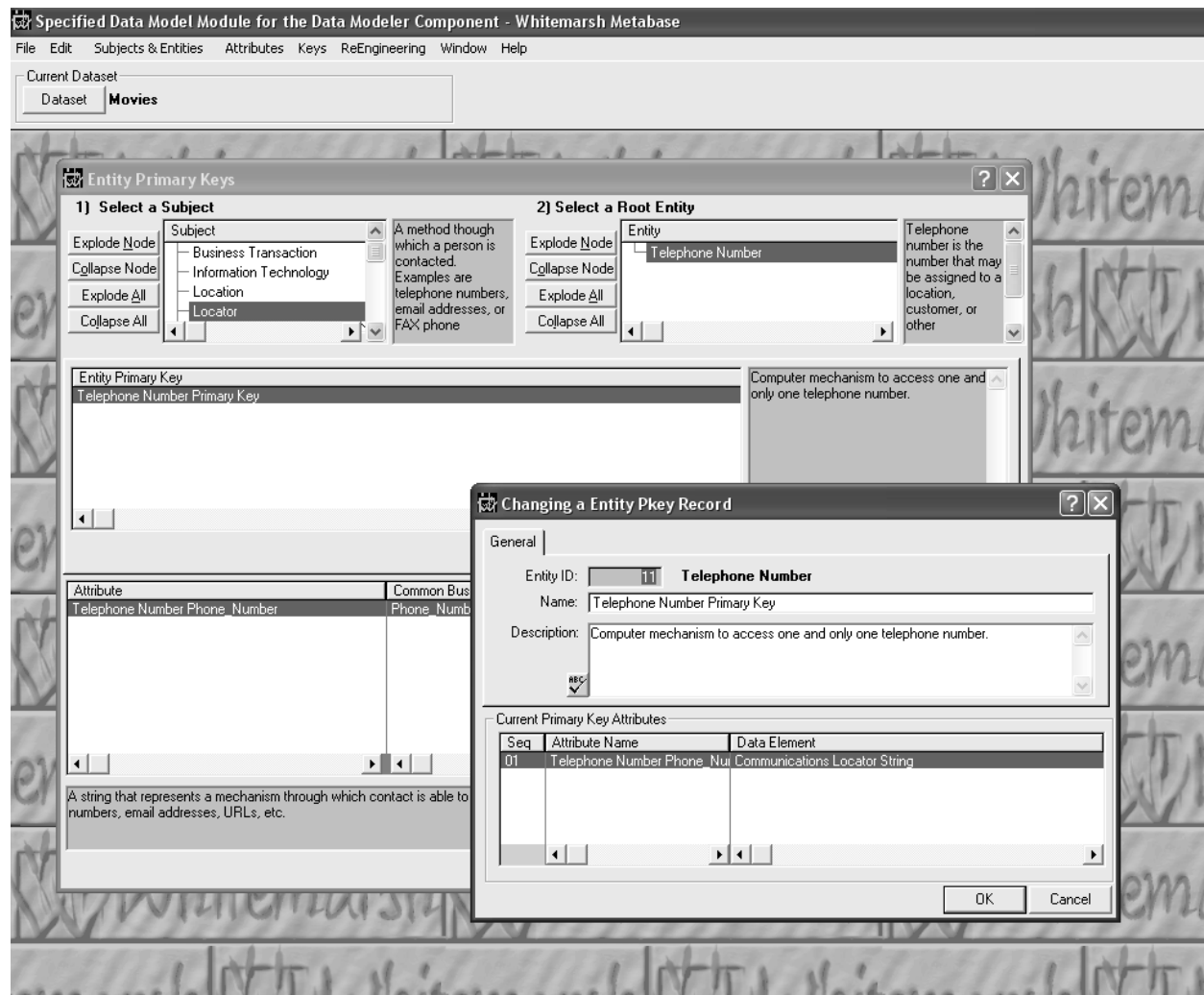


Figure 17. Primary Key update screen.



6.1.3.1.2 Allocation of Attributes to the Primary Key

A primary key is a collection of attributes. The order of the attributes within the primary key imply the order of the values used to select rows of data given that the entity is in fact a table. Figure 18 presents a list of the primary keys and the current set of attributes assigned to each. To assign an attribute to a primary key, highlight the subject then the entity, then tag the entity primary key. Then, from the entity's shown attributes, tag the one or more attributes that comprise the primary key. Finally, press the Build button. The attributes that then comprise the primary key are shown in the bottom window. Once the attributes are assigned, the Up|Down buttons can be used to change the sequence of the attributes within the primary key.

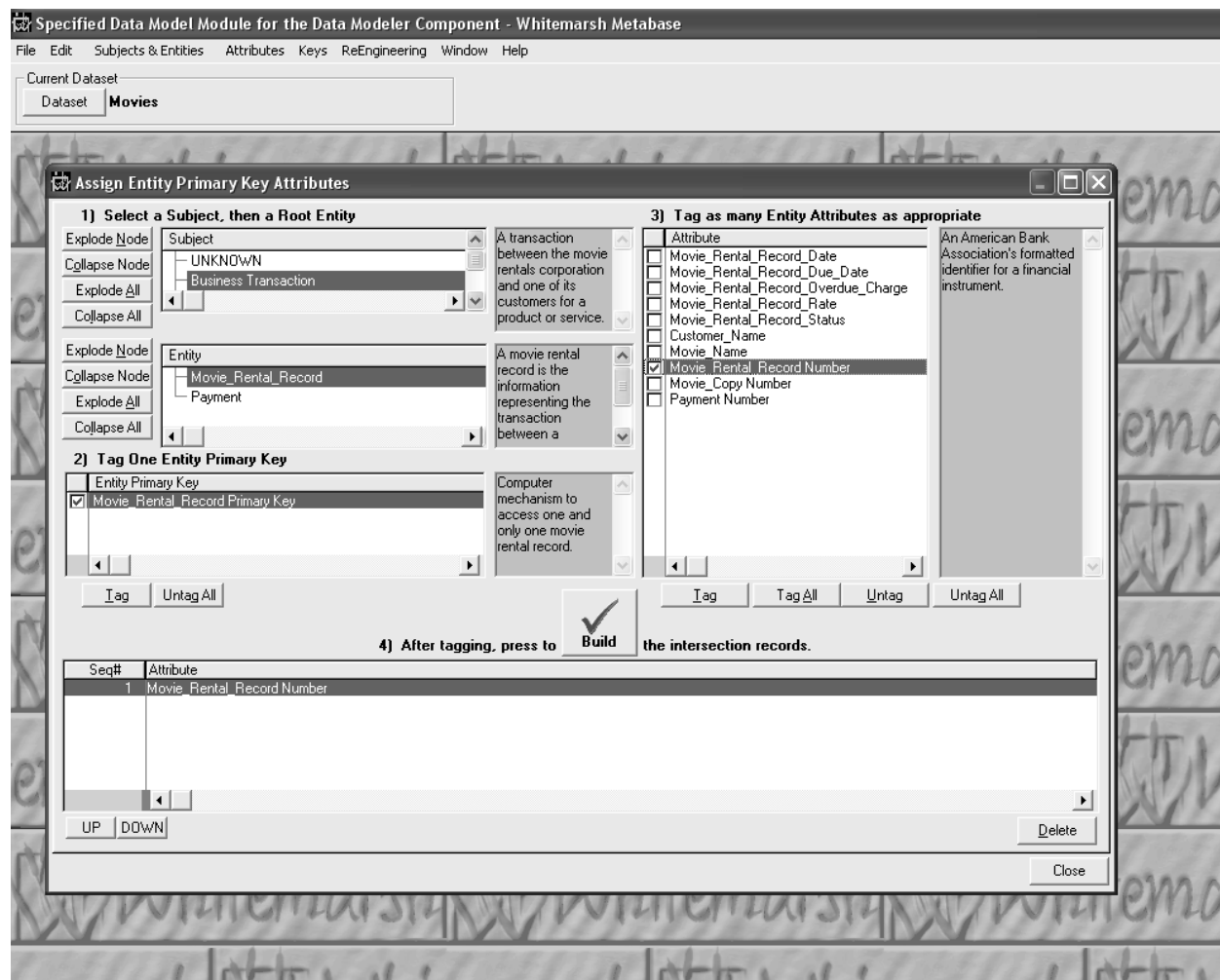


Figure 18. Primary Key Attribute assignment screen.



6.1.3.2 Foreign Keys

Foreign keys are primary keys from another entity. Hence this key is a *foreign* key. There are only two critical pieces of information necessary to create a foreign key are the specific primary key of the source entity, and the target entity. Once these two pieces of information are provided then the rest is automatic. That is, the foreign key's name, and attributes that comprise it.

Figure 19 presents the current list of foreign keys. To enter a new foreign key, highlight the subject, and then the entity that is to contain the foreign key. The current list of foreign keys within that entity are listed. To then create a new foreign key, press Insert. To change an existing foreign key press Change, and to delete an existing foreign key press Delete.

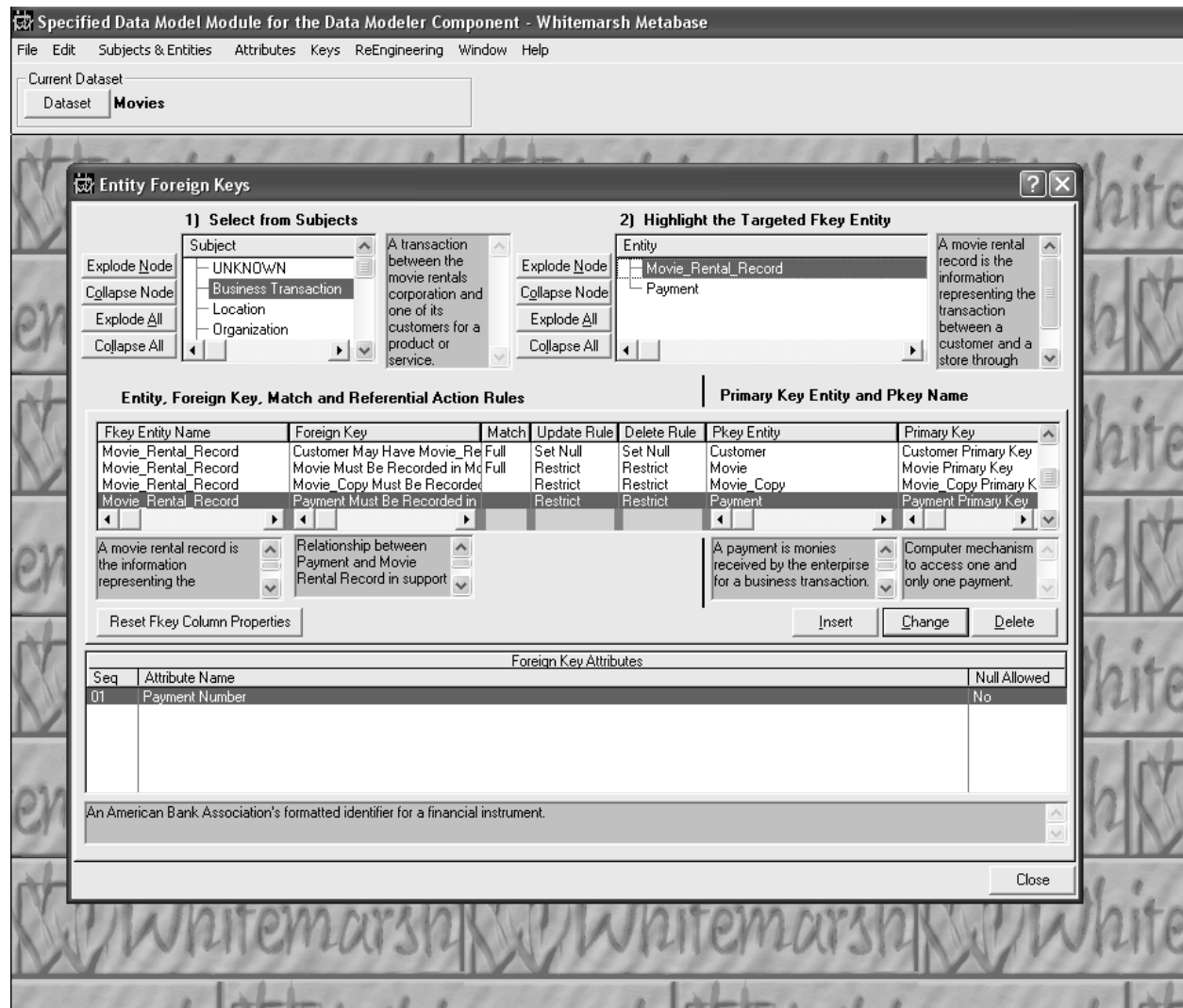


Figure 19. Foreign Keys.



Figure 20 presents the form for entering a new foreign key. The first step is the enter the value for Entity Pkey Id. If valued with a zero and then the Tab key is pressed, a list like the one in Figure 21 is presented. Highlight the appropriate subject and then the appropriate primary key, which also shows the source entity name. Then press Select.

If the entity that was to contain the foreign key had been previously highlighted before the Insert button was pressed then the entity's Id and name appears as the second data entry item. If a zero appears, then press Tab to cause a list of possible target entities. Figure 22 shows that list. Highlight the appropriate entity and press Select. The only entities that will appear are those that are not sub-typed nor outside the existing schema.

Specified Data Model Module for the Data Modeler Component - Whitemarsh Metabase

File Edit Subjects & Entities Attributes Keys ReEngineering Window Help

Current Dataset
Dataset **Movies**

Entity Foreign Keys

1) Select from Subjects

Subject: UNKNOWN, Business Transaction, Location, Organization

2) Highlight the Targeted Fkey Entity

Entity: Movie_Rental_Record, Payment

Entity, Foreign Key, Match and

Fkey Entity Name: Movie_Rental_Record, Customer Primary Key

Entity Pkey ID: 17

Entity ID: 7

Entity: Movie_Rental_Record

3) Enter a short, singular tense action phrase: Have

Interim or Resulting Fkey Name: Customer May Have Movie_Rental_Record

4) Select Key Match Option

Match Type: Full, Partial, Unique, Y, N

5) Choose the Referential Actions

Update Rule: Cascade, Set Null, Set Default, Restrict, No Action

Delete Rule: Cascade, Set Null, Set Default, Restrict, No Action

6) Describe Foreign Key Purpose:

Relationship between a customer and all the movies that have been rented by the customer.

7) If Fkey attribute is to be manually picked

Then: Select Attribute for Fkey Or Suggest

8) Press to Create Foreign Key or To Commit Changes

Cancel

Figure 20. Foreign Key update screen.



Then, the third step is to enter a singular present tense action phrase. This phrase, for example, have, is employed to construct the foreign key's name. Figure 23 illustrates the entry of the singular present tense action phrase. Immediately below the phrase, Have, is the constructed foreign key name, Customer must have Movie Order Record. Customer is the source entity. Movie Order Record is the target entity, and "have" is the action phrase. The word must results from the default or selected Referential Actions, No Action. A complete set of the meanings of the Referential actions is contained in the Data Modeler Architecture book from the Whitemarsh website.

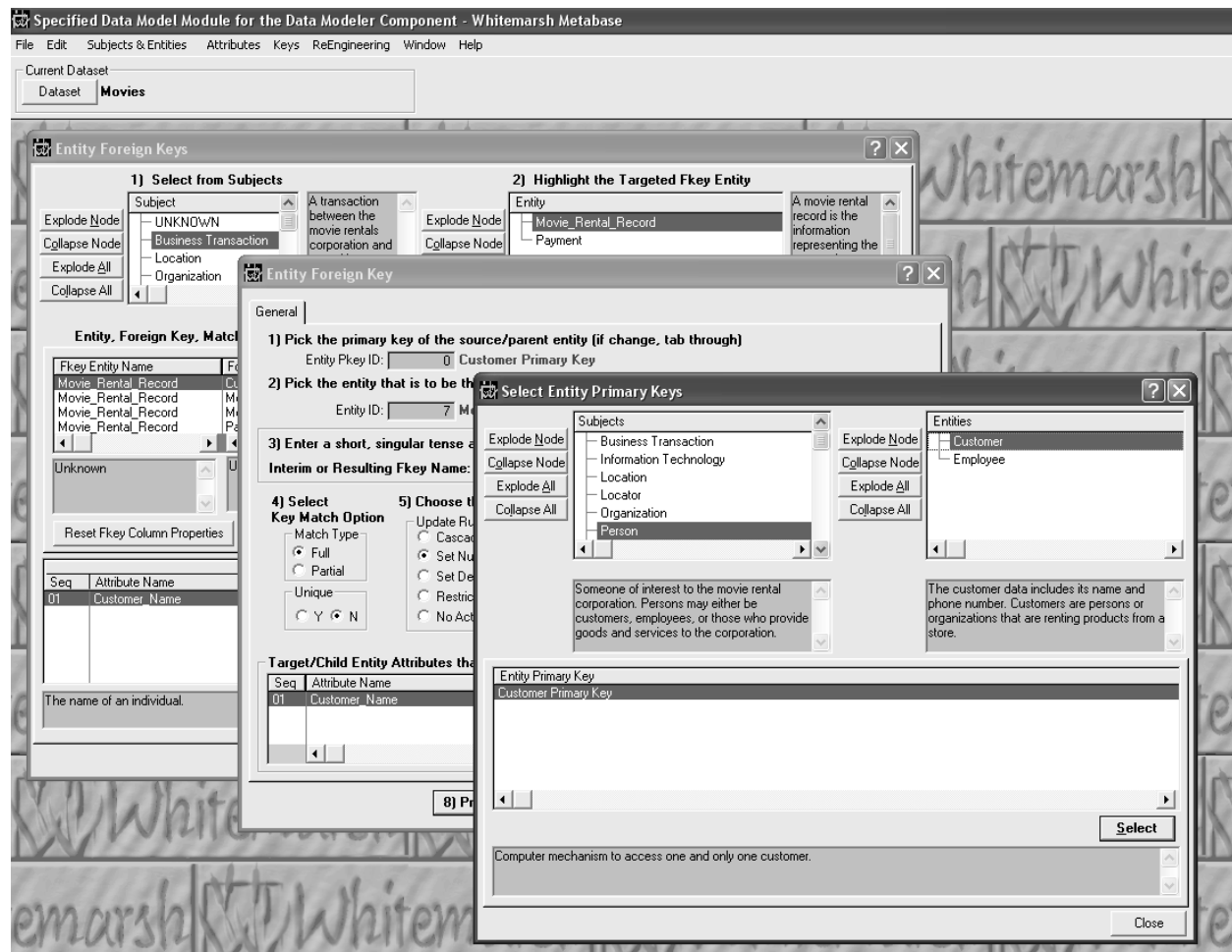


Figure 21. Selecting a Primary Key for Foreign Key selection.



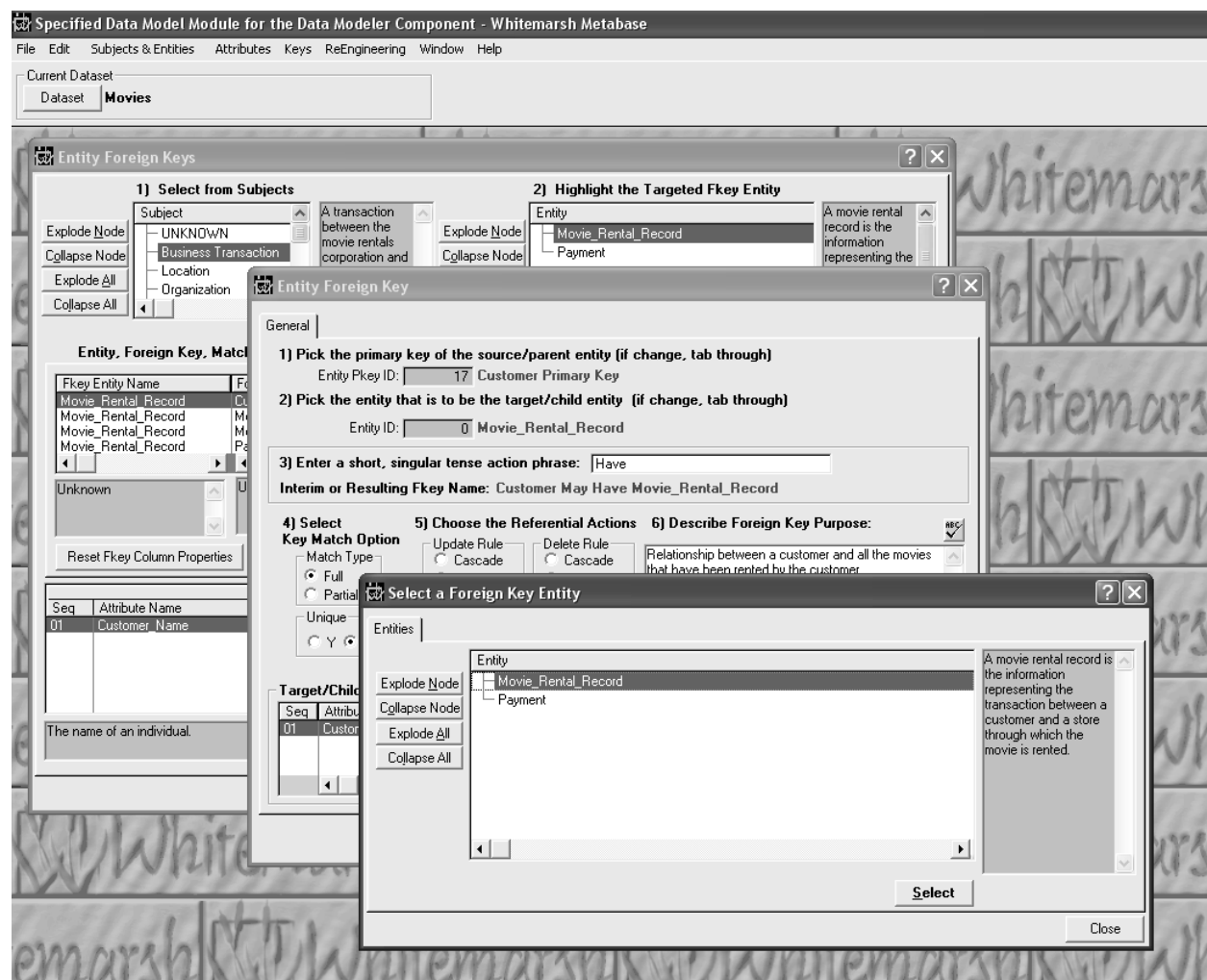


Figure 22. Selecting the target Entity for Foreign Key creation.



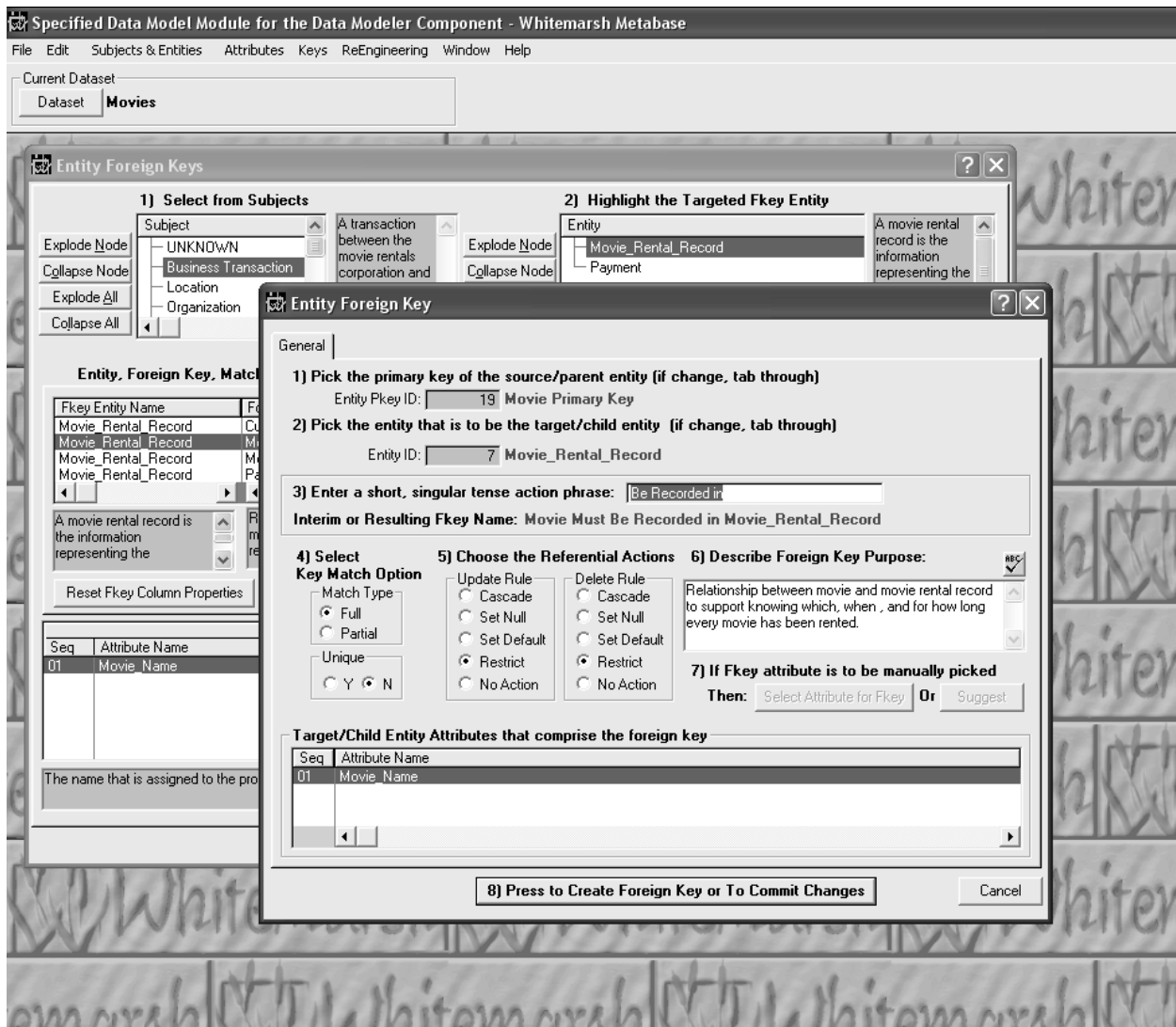


Figure 23. Entering a verb phrase to create Foreign Key Name.



6.1.3.3 Candidate

Included in the definition of a complete candidate key is:

- Candidate key definition
- Allocation of attributes to the candidate key

6.1.3.3.1 Candidate Key Definition

A candidate key of an entity is a set of one or more attributes that represent values that when employed result in only one selected row. Figure 24 shows the current set of candidate keys. There can be multiple candidate keys for each entity. To see the attributes assigned to a particular candidate key, highlight the appropriate subject, and then entity.

To add a candidate for an entity press Insert. If the Insert or change action succeeds, the Figure 25 is presented. On an Insert, the name is automatically constructed as the concatenation of the entity name and the string, "Candidate Key." The name can be changed. In addition to the name a description can be added or changed.

6.1.3.3.2 Allocation of Attributes to the Candidate Key

A candidate key is a collection of attributes. The order of the attributes within the candidate key imply the order of the values used to select rows of data given that the entity is in fact a table. Figure 26 presents a list of the candidate keys and the current set of attributes assigned to each. To assign an attribute to a candidate key, highlight the subject then the entity, then tag the appropriate entity candidate key. Then, from the entity's shown attributes, tag the one or more attributes that comprise the candidate key. Finally, press the Build button. The attributes that then comprise the candidate key are shown in the bottom window. Once the attributes are assigned, the Up|Down buttons can be used to change the sequence of the attributes within the candidate key.



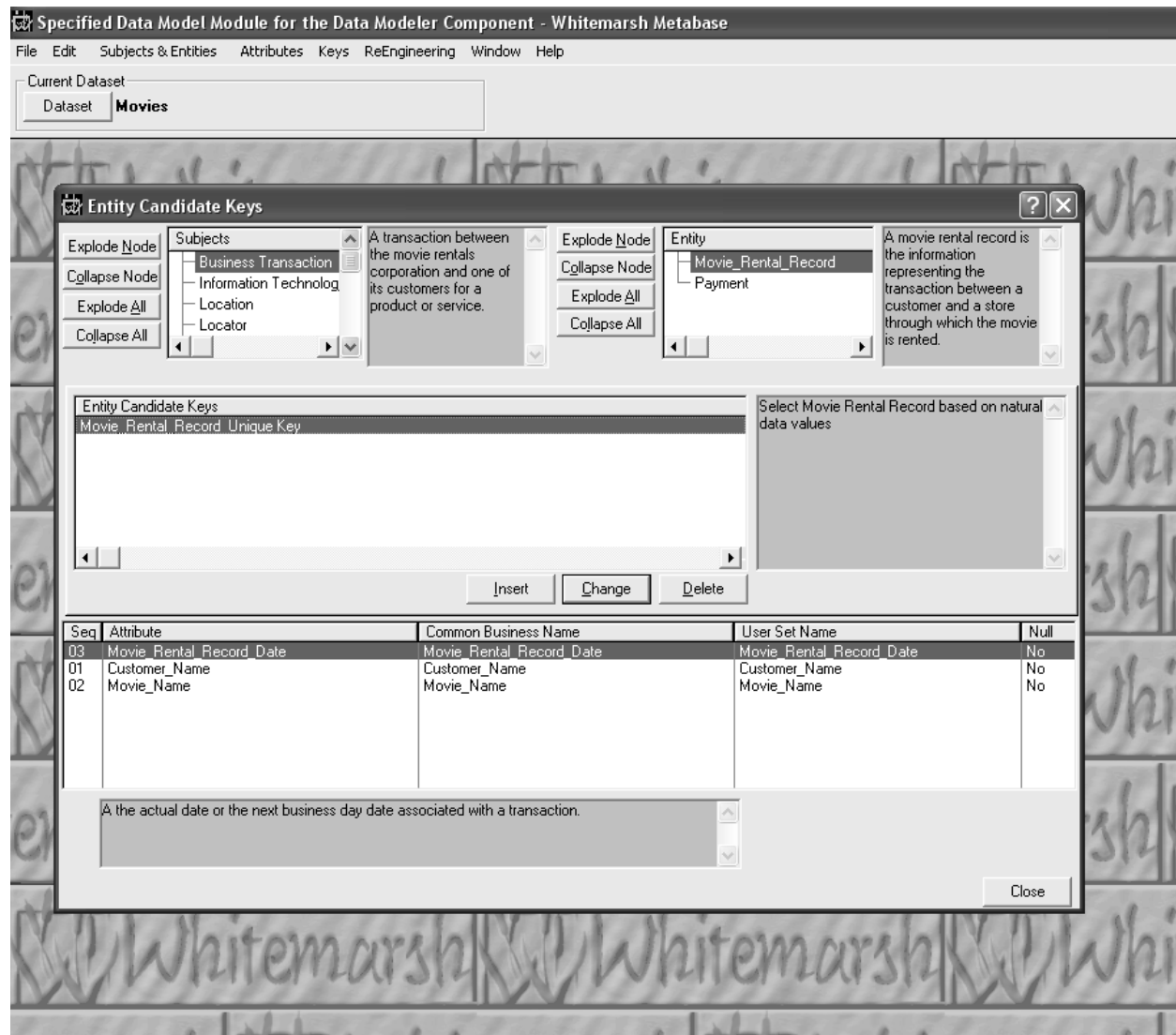


Figure 24. Candidate Keys.



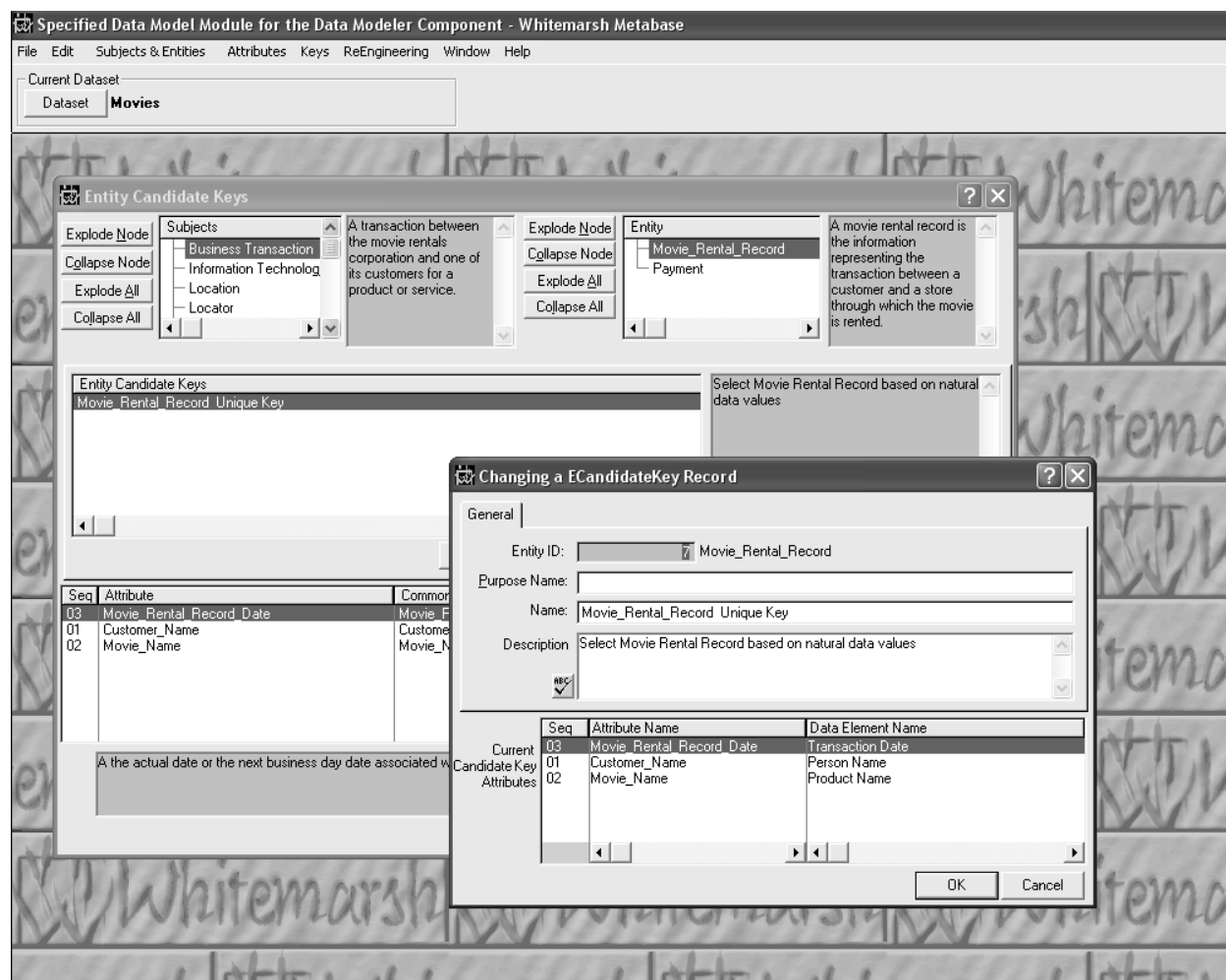


Figure 25. Candidate Key update screen.



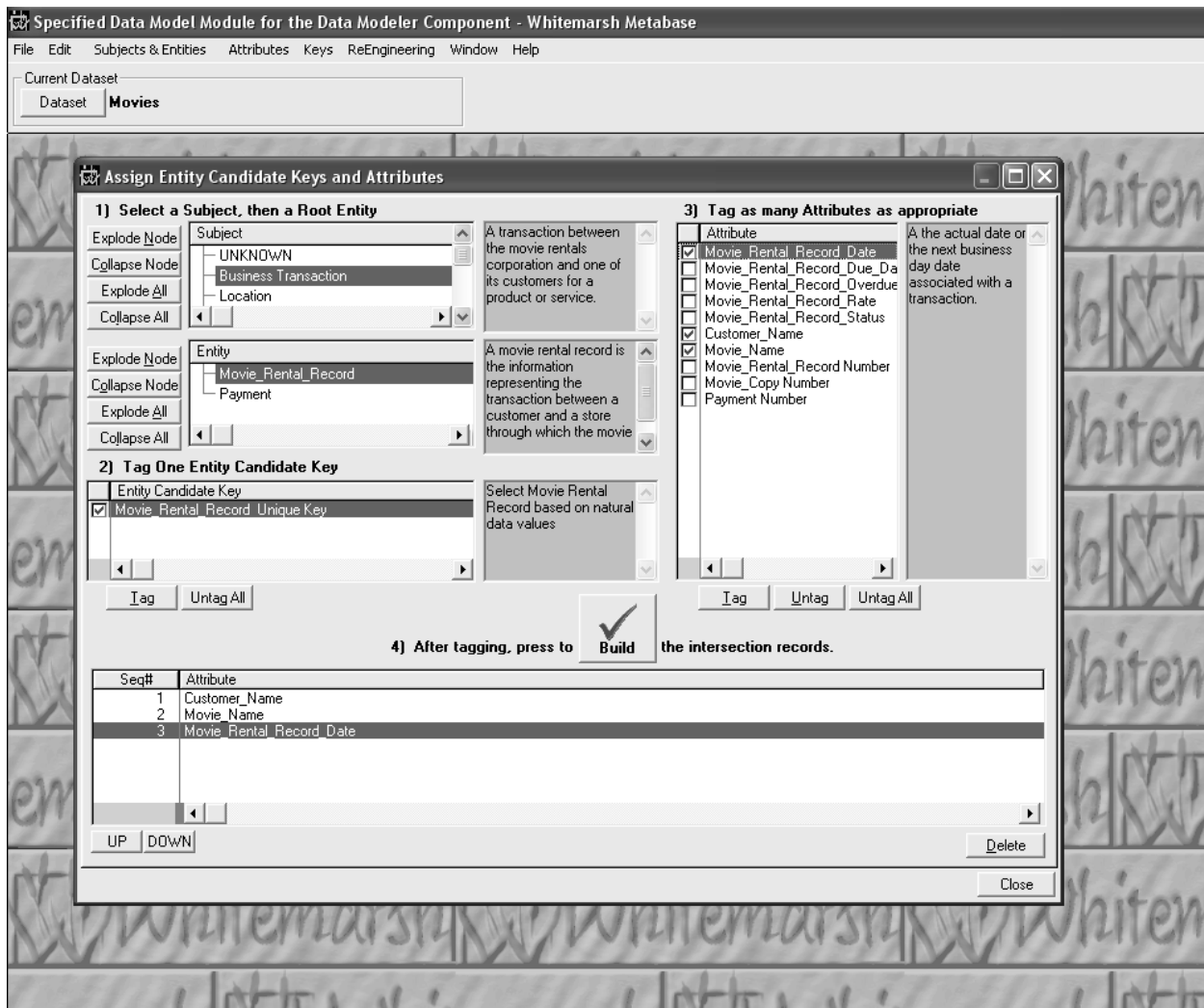


Figure 26. Allocating attributes to a Candidate Key.



6.1.4 Reverse Engineering

Reverse engineering is the process of recasting an already created collection of specified data model components. Included are:

- Reassigning Subjects to Subject
- Reassigning Entities to Subject
- Reassigning Entities to Entities
- Reassigning Attributes to Data Elements
- Reassigning Attributes to Entities
- Promoting Attribute to Data Element
- Remove Attribute Meta Category Values

6.1.4.1 Reassigning Subjects to Subject

Subjects can have subordinate subjects. This is shown in Figure 27. An subject might be pushed too far down in the subject family. In this case the reassignment allows the “parent” of an subject to be changed. To accomplish this, tag one or more subjects in the left window and then the new parent in the right window. Then press the Re-assign button. If a re-assign message is appropriate on either the left or right window it will be displayed. These messages are designed to prevent inappropriate re-assignments. For example, making a different root subject.

6.1.4.2 Reassigning Entities to Subject

Figure 28 displays the process for re-assigning entities to different subjects. The process starts with highlighting the entity that is to be re-assigned. Tag one or more entities that are to be re-assigned. Then highlight and tag the new subject. Once the entities and a subject is tagged, press the Re-Assign button. The underlying process then re-assigns the entity from the current subject to the new subject. Once all the tagged entities are reassigned the windows are redisplayed with the newly assigned subjects.

If a root entity is tagged then all its sub-typed entities are moved as well. Also automatically moved are all entities that are related to the root entity by primary-to-foreign key relationships. To move just single entities, start with “leaves” and work backwards towards the entity’s root.

6.1.4.3 Reassigning Entities to Entities

As stated above, entities can have sub-typed entities. This is shown in Figure 29. An entity might be pushed too far down in the entity family. In this case the reassignment allows the “parent” of an sub-typed entity to be changed. To accomplish this, tag one or more entities in the left



window and then the new parent in the right window. Then press the Re-assign button. If a re-assign message is appropriate on either the left or right window it will be displayed. These messages are designed to prevent inappropriate re-assignments. For example, making a different root entity.

6.1.4.4 Reassigning Attributes to Data Elements

Figure 30 presents the screen through which attributes are re-assigned to different data elements. Tag one or more attributes in the left window. Then Tag one data element in the right window and then press the Re-Assign button. Once the re-assignment process is complete the reallocated attributes appear in the left window.

6.1.4.5 Reassigning Attributes to Entities

Entities can be sub-typed. Essentially that means that the complete set of attributes can be spread across an entity family. An entity family is a root entity and one or more sub-entities or their sub-entities. During the process of creating attributes within an entity it might be discovered that the attribute really belongs in a different entity. This type of reassignment supports moving an attribute from one entity within an entity family to another entity.

Figure 31 illustrates the process of reassigning one or more attributes from one entity to another entity within the same family. Highlight the subject and then the entity. Then tag one or more attributes that are to be moved. Then tag the entity to which the attributes are to be moved. Press the Build button. The attributes are then moved.

6.1.4.6 Synchronize Attributes to Local Definitions

Attributes have local definitions. Some attributes within different entities and subjects are to have essentially the same local definition. For example, in several of the entities within this movies demo database has a Movie Name attribute. For sure, in all cases the local definition would be, “the name of the movie.” If this local definition is different across the attributes of the different entities and subjects, users will wonder why the difference exists. What’s the hidden meaning because of the difference? Figure 32 provides the ability to synchronize local definitions across attributes. On the left side, select a subject, entity, and then tag the attribute that is to be source of the local definitions. On the right side of Figure 32, select as many different attributes as may be appropriate. Do this by selecting the subjects, entities, and attributes. Different attributes from the different entities, and schemas can be selected and tagged. Once selected and tagged, press the Synchronize Definition button. Note: the only definition that is synchronized is the local definition. The contextual definition will have to be re-generated.



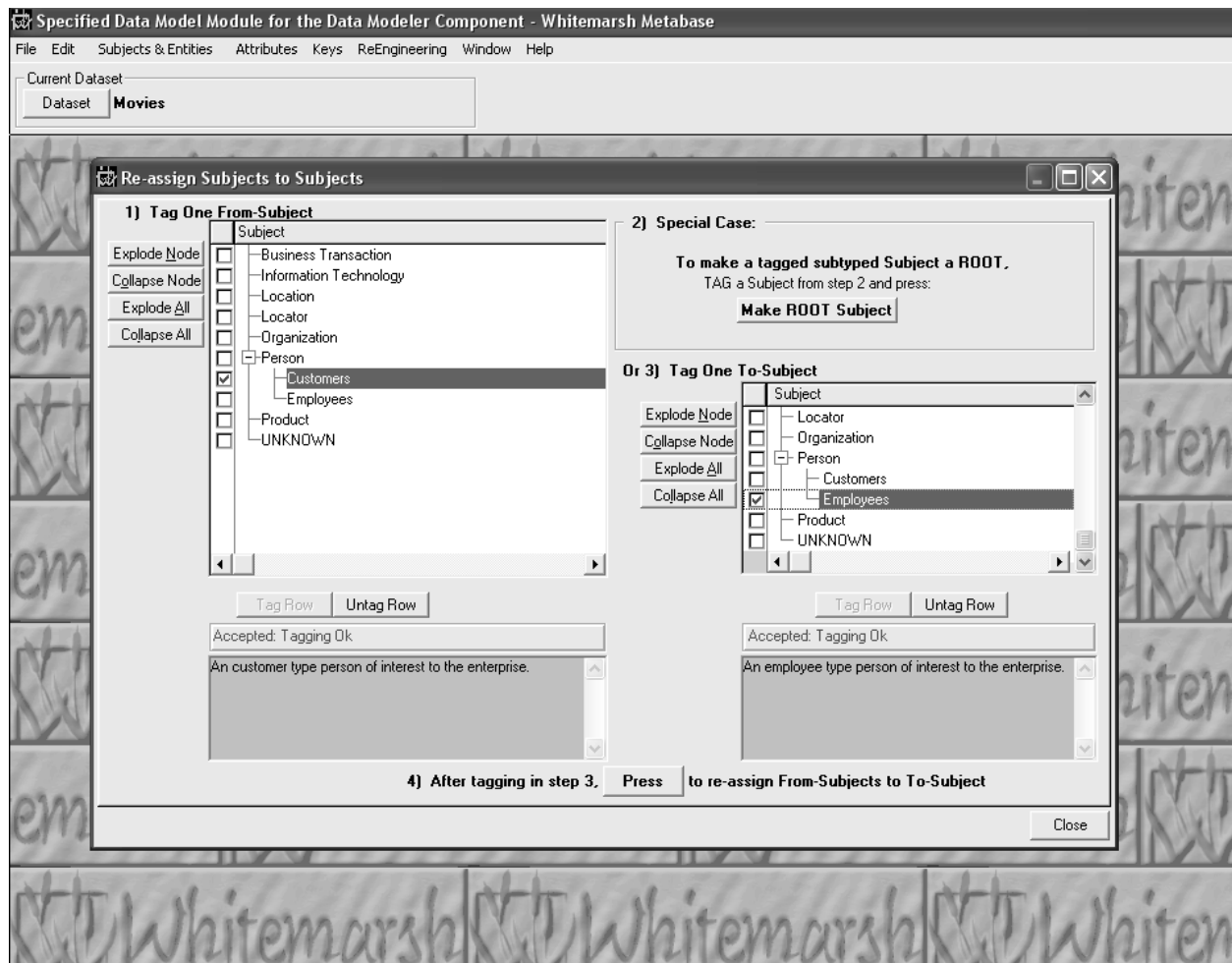


Figure 27. Reassign Subjects to Subjects.



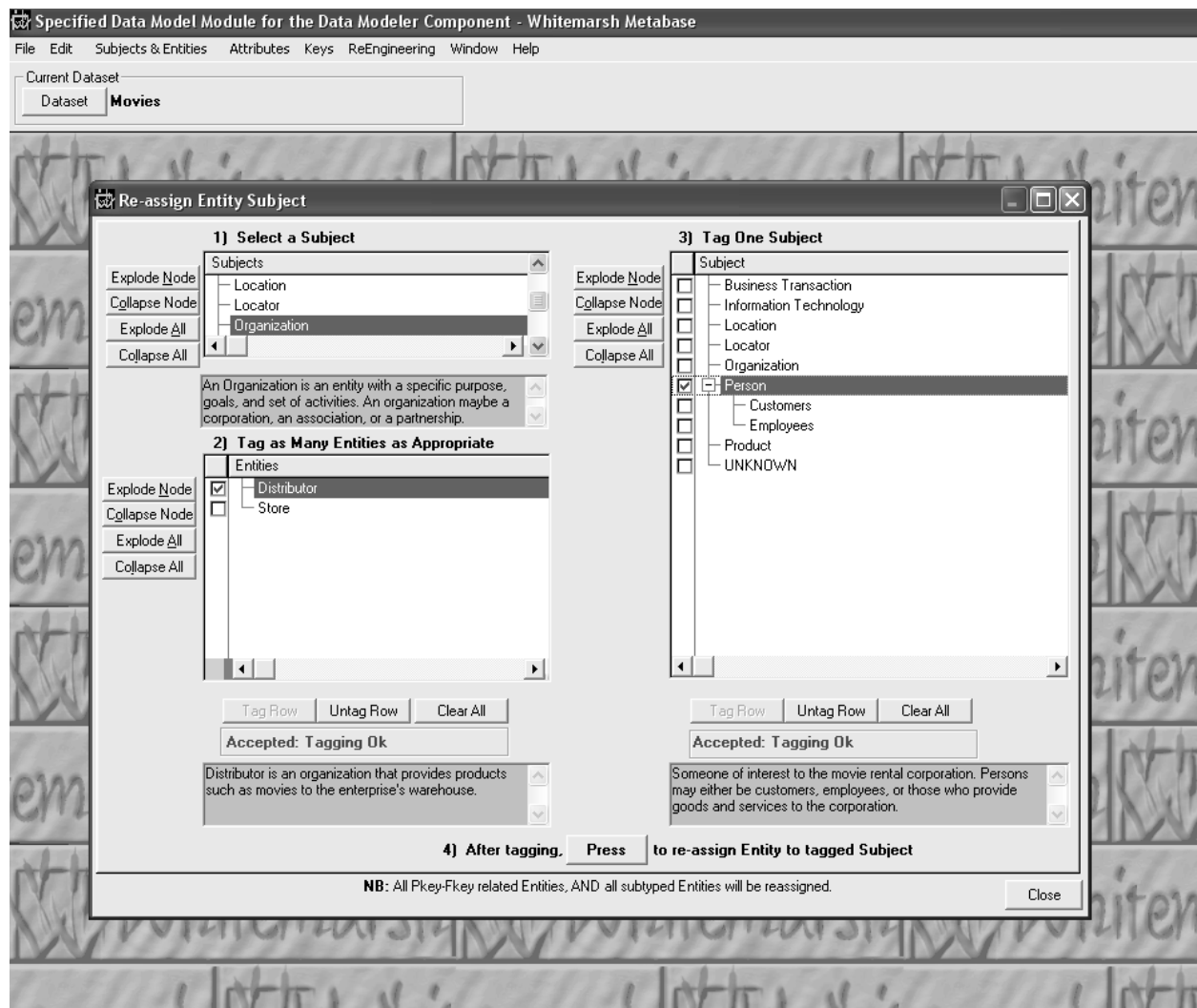


Figure 28. Reassign Entity to Subject.



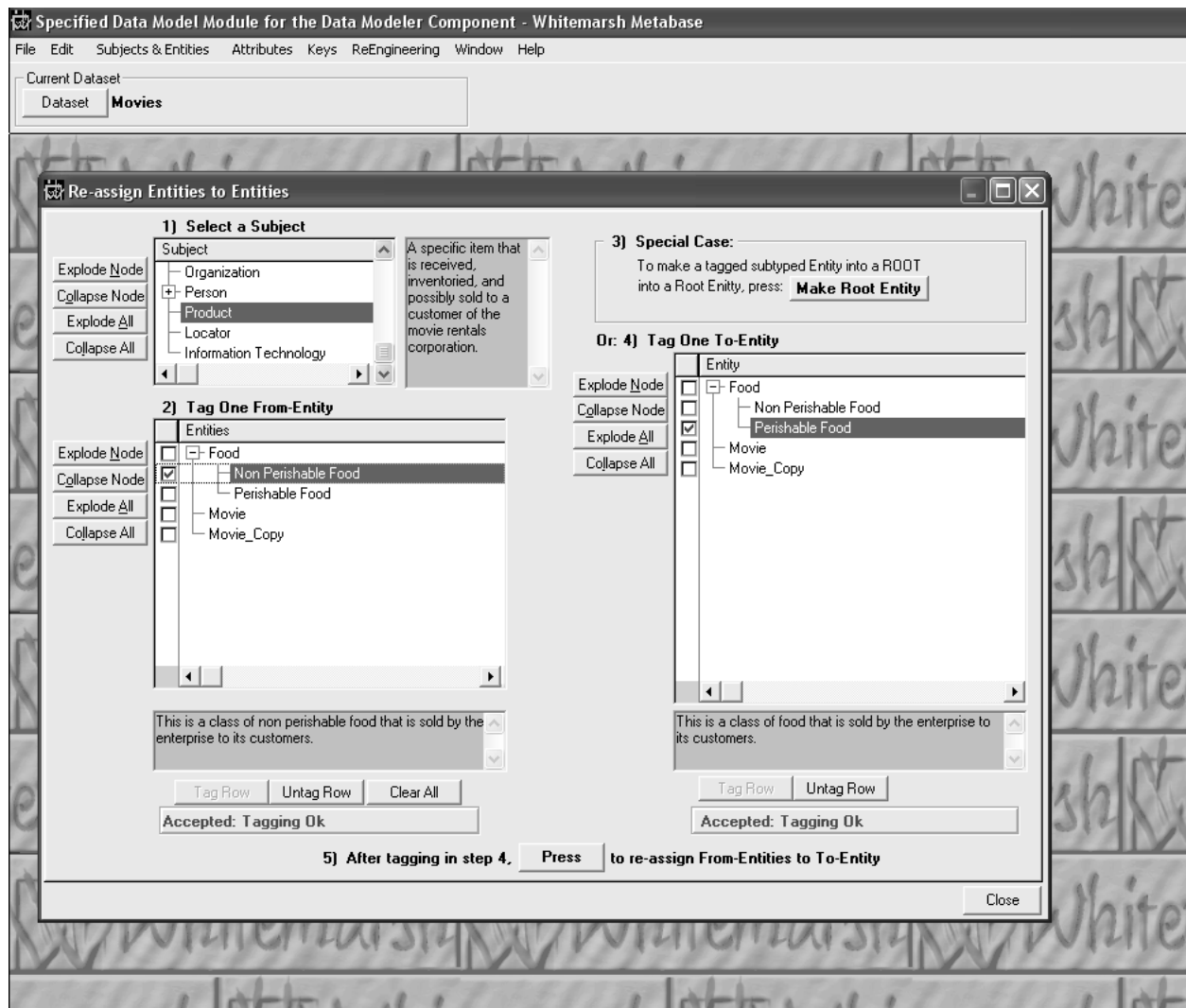


Figure 29. Reassigning Entity to Entity.



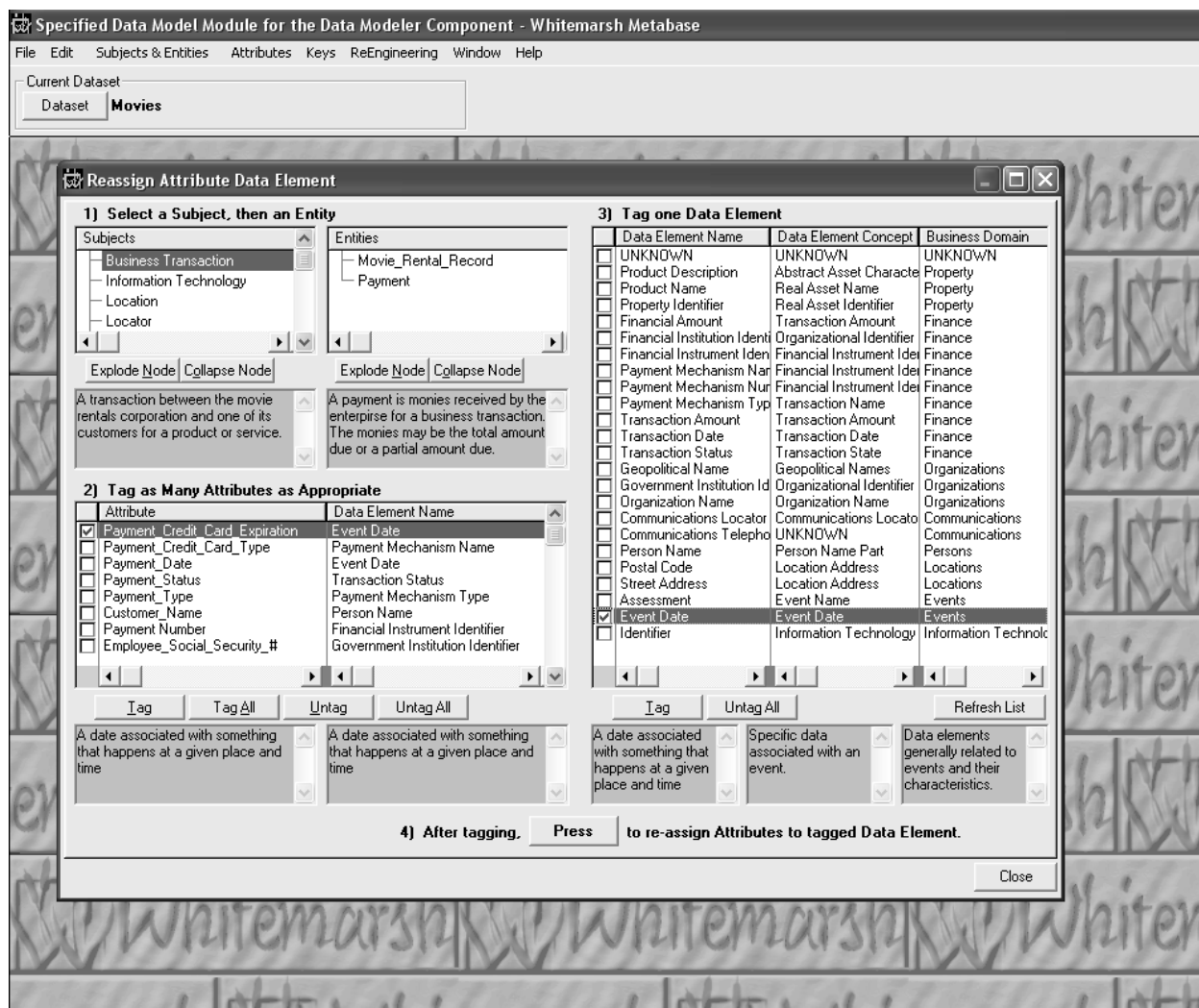


Figure 30. Reassign Attribute to Data Element.



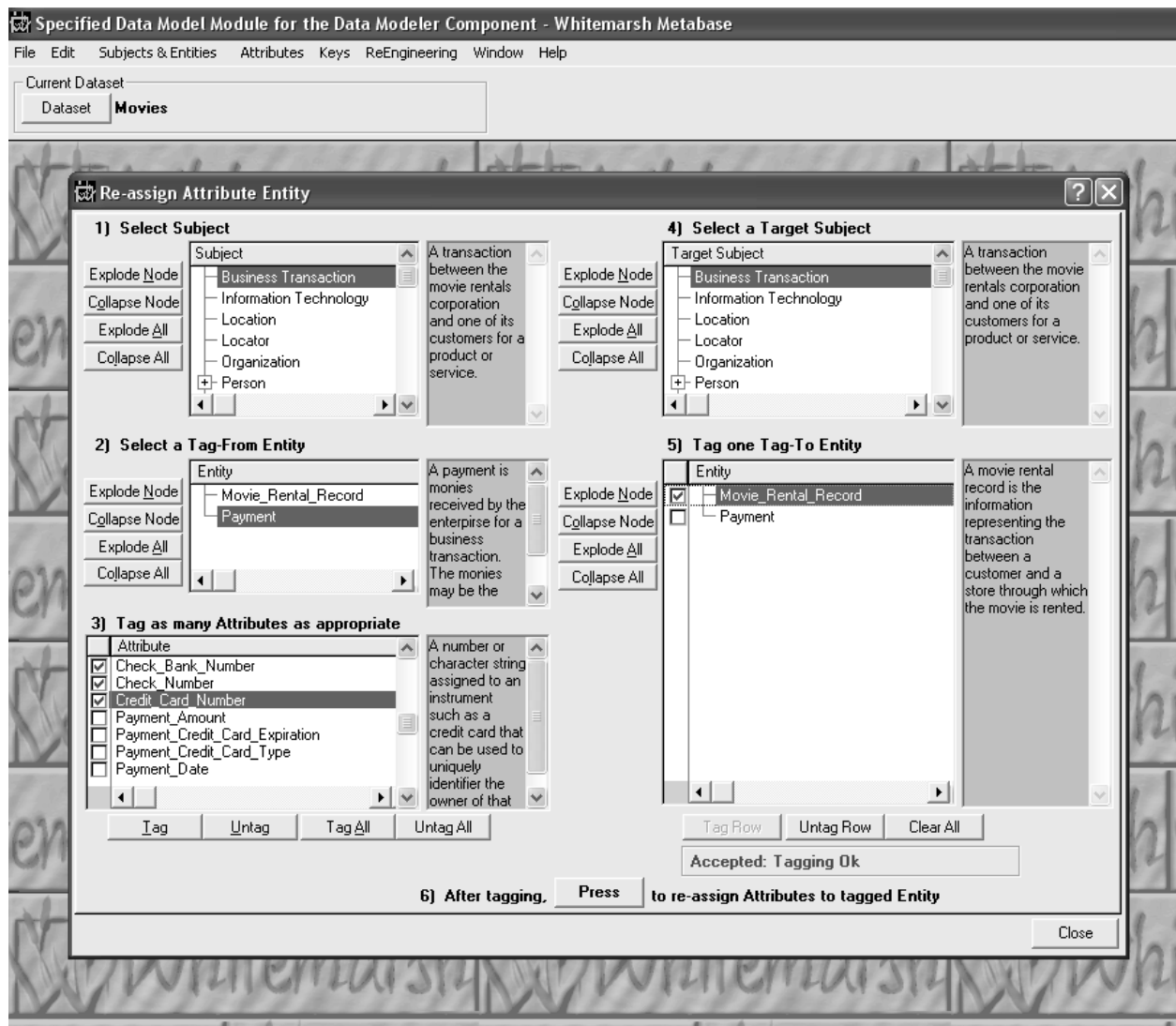


Figure 31. Reassign Attribute to Entity.



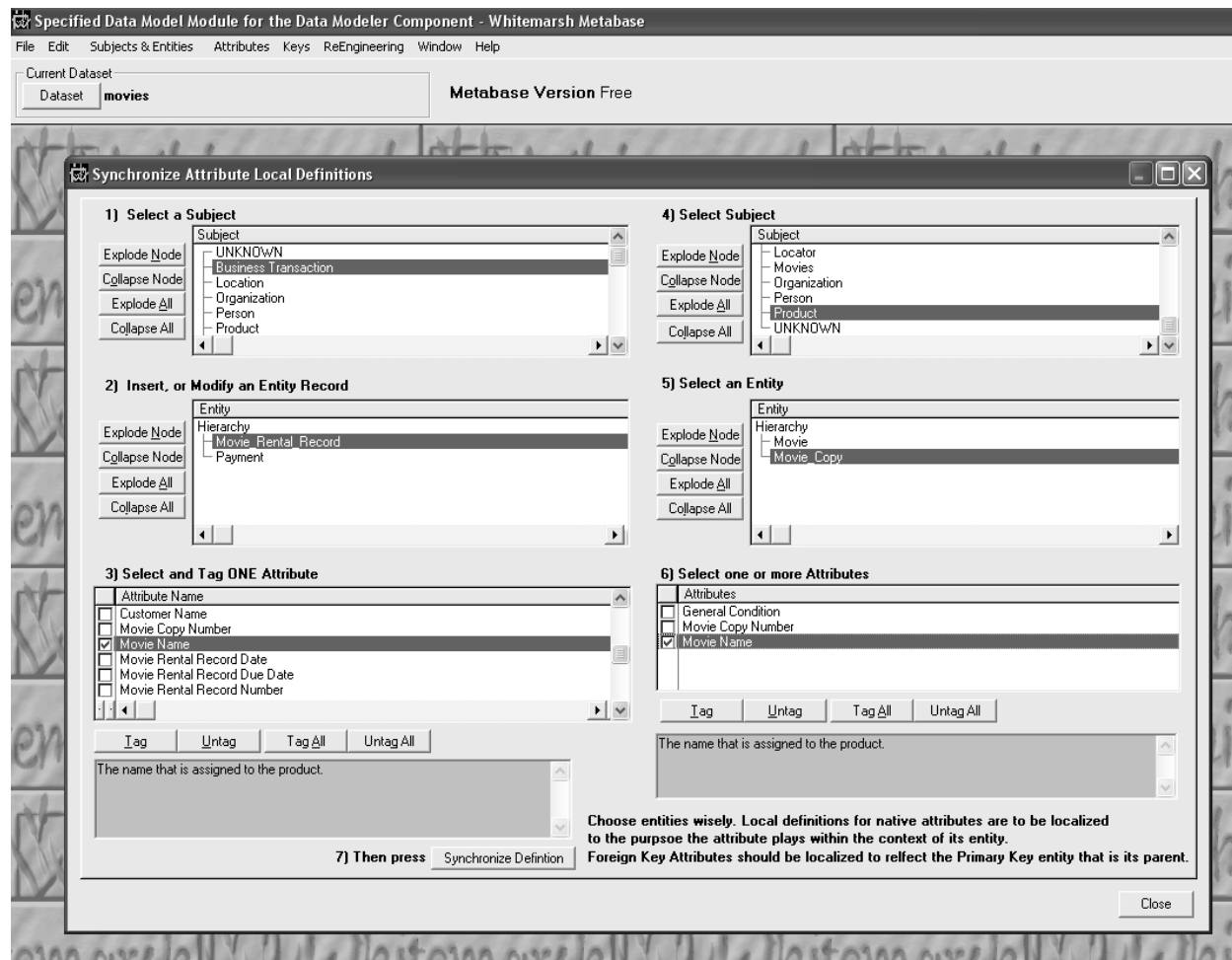


Figure 32. Synchronize Attribute Local Definitions.



6.1.4.6 Promoting Attribute to Data Element

Figure 33 displays the window for promoting an attribute to a data element. The attribute is identified by highlighting the subject then the entity, and finally the attribute. Then highlight the data element concept. Once the attribute and the data element concept is highlighted, press the Promote button. The underlying process a new data element is created from attribute and assigns to that newly created data element all the semantics assigned to the promoted attribute. The semantics assigned to the attribute are then deleted as they are then inherited from the newly created data element. The existing attribute is not automatically removed as it may have been the “parent” of one or more columns. If a delete operation is attempted, and the attribute is not a parent of a column then the delete process will succeed.

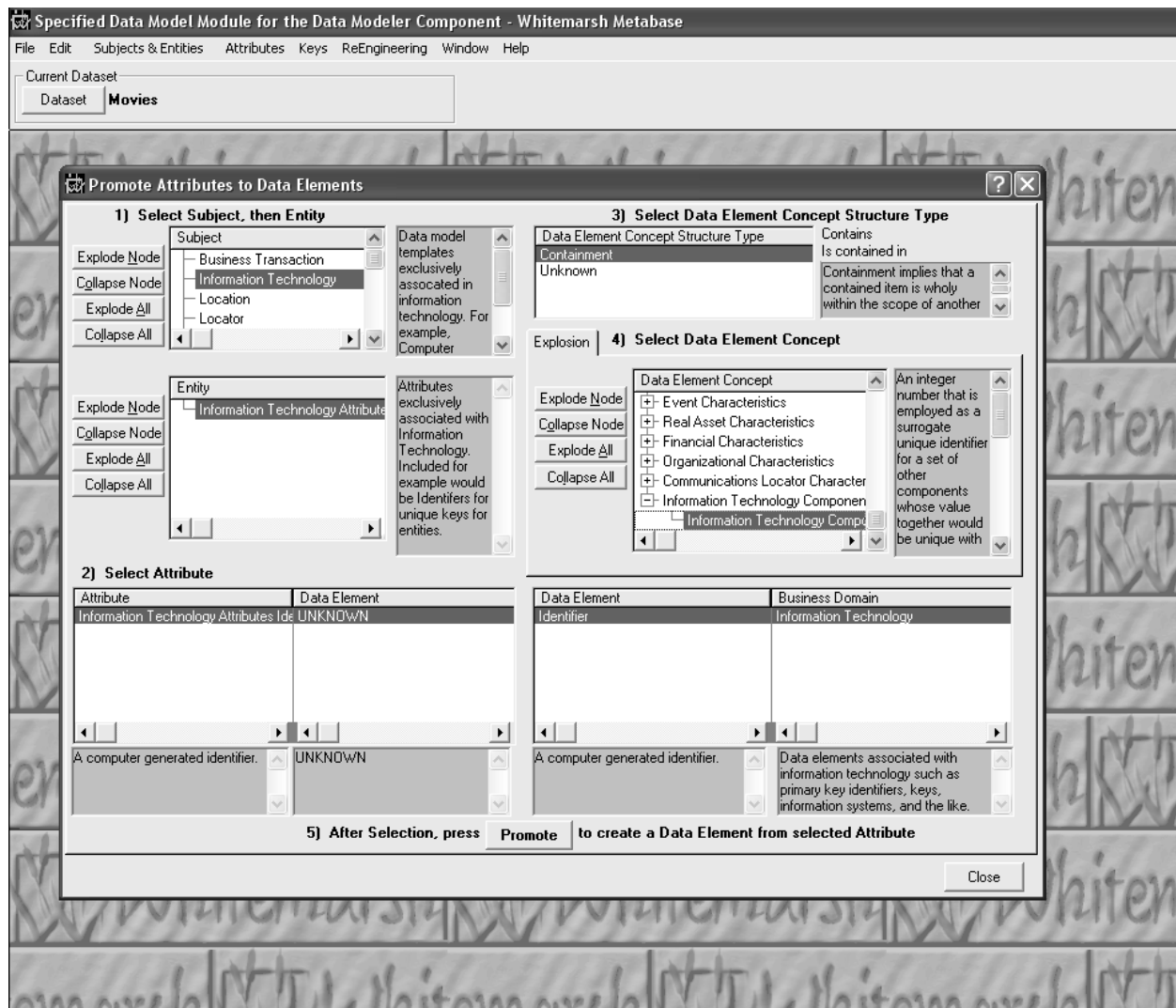


Figure 33. Promote Attribute to Data Element.



6.1.4.7 Removing Attribute Meta Category Values

Figure 34 displays the window for removing meta category values from an attribute. Select the subject, entity, and attribute. Select the specific meta category value that is to be deleted. Then press the Delete button.

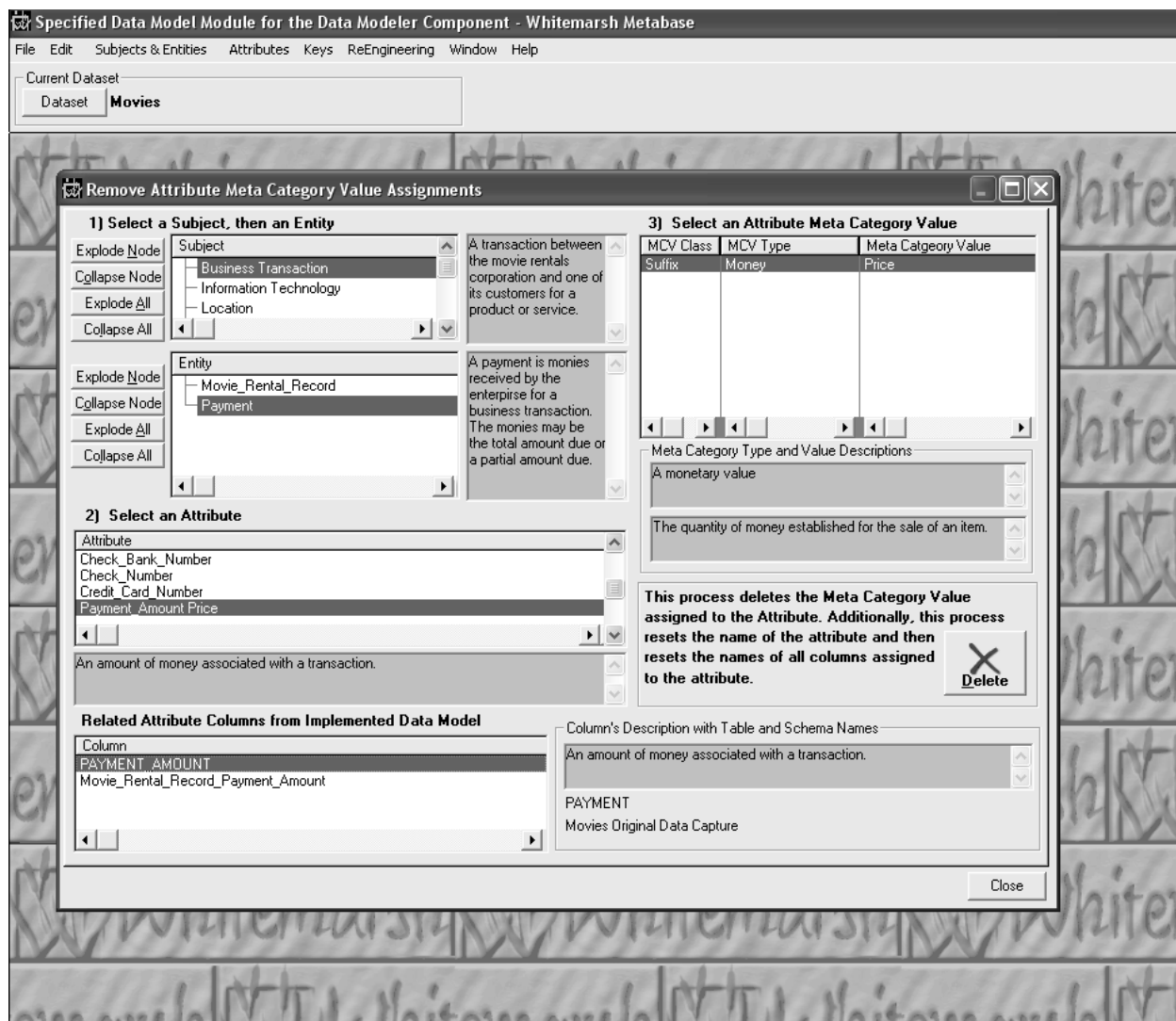


Figure 34. Remove Attribute Meta Category Values.



6.1.5 SQL DDL

Once a specified data model has been completely entered, that is, its entities, attributes and relationships, it can be displayed either graphically within the specified data modeler, or through an ER modeler such as DeZine (www.datanamic.com). To diagram within DeZiner, the data model must be imported. DeZiner, like many tools, has an SQL DDL import facility. Therefore, the metabase has an SQL DDL export facility. Additionally, another data modeling tool may have been used to create what is seen as a specified data model. Thus, the metabase has an SQL DDL import facility.

6.1.5.1 Export

Figure 35 presents the first screen in this process. The top browse shows the subject areas. The second browse shows an alphabetical listing of the entities within that subject area.

Obviously the root entity is not always the first. If the root entity is highlighted and then the button Display Specified Data Model is pressed, then a screen like that of Figure 35 is presented.

The process that is invoked starts with the highlighted entity and traces through all its foreign keys to then “know” all related entities. The hierarchy that is displayed in Figure 35 is all the descendant of the entity and all the direct ancestors (no uncles or aunts). When an ancestor entity is displayed its color is red. Descendent entities are blue and the originally highlighted entity's color is black. Subtyped entities are teal.

As each entity in Figure 36 is highlighted, the surrounding browses then display the entity's primary key and attributes, foreign keys and attributes, and attributes. If the Print Tree button is pressed a hierarchy tree is sent to the default printer. If the Print Tree Detail button is pressed the data model tree is printed along with an additional level of detail.

Figure 35 also has a set of buttons at the bottom. The first button, Select Output File for Generated DDL, causes a Select File display as presented in Figure 37. A default file name is presented that can be accepted or changed. The default directory is the current working directory.

Once the output file is selected, two buttons are then available for selection. The first is Generate Subject Based Data Model button. If selected, Figure 38 is displayed and all the entities within the subject area are accessed and then linguistically expressed using SQL. The value of this DDL file is that it can then be imported by another software package. Since attributes within the specified data model do not have data types (character, integer, etc.) their data types are default to Char(1).

The Generate Entity Based Data Model button only generates SQL DDL for all entities that are descendants and direct ancestors of the highlighted entity.

There are three options for representing subtyped entities: One, Each, and SQL:1999. The one option causes all the attributes from the contained subtyped entities to appear within the root entity. The Each option causes each subtyped entity to be expressed as a separate SQL table with its primary key the same as the root table's and the foreign key column reference to also be the same as its primary key. This ensures a 1:1 relationship.



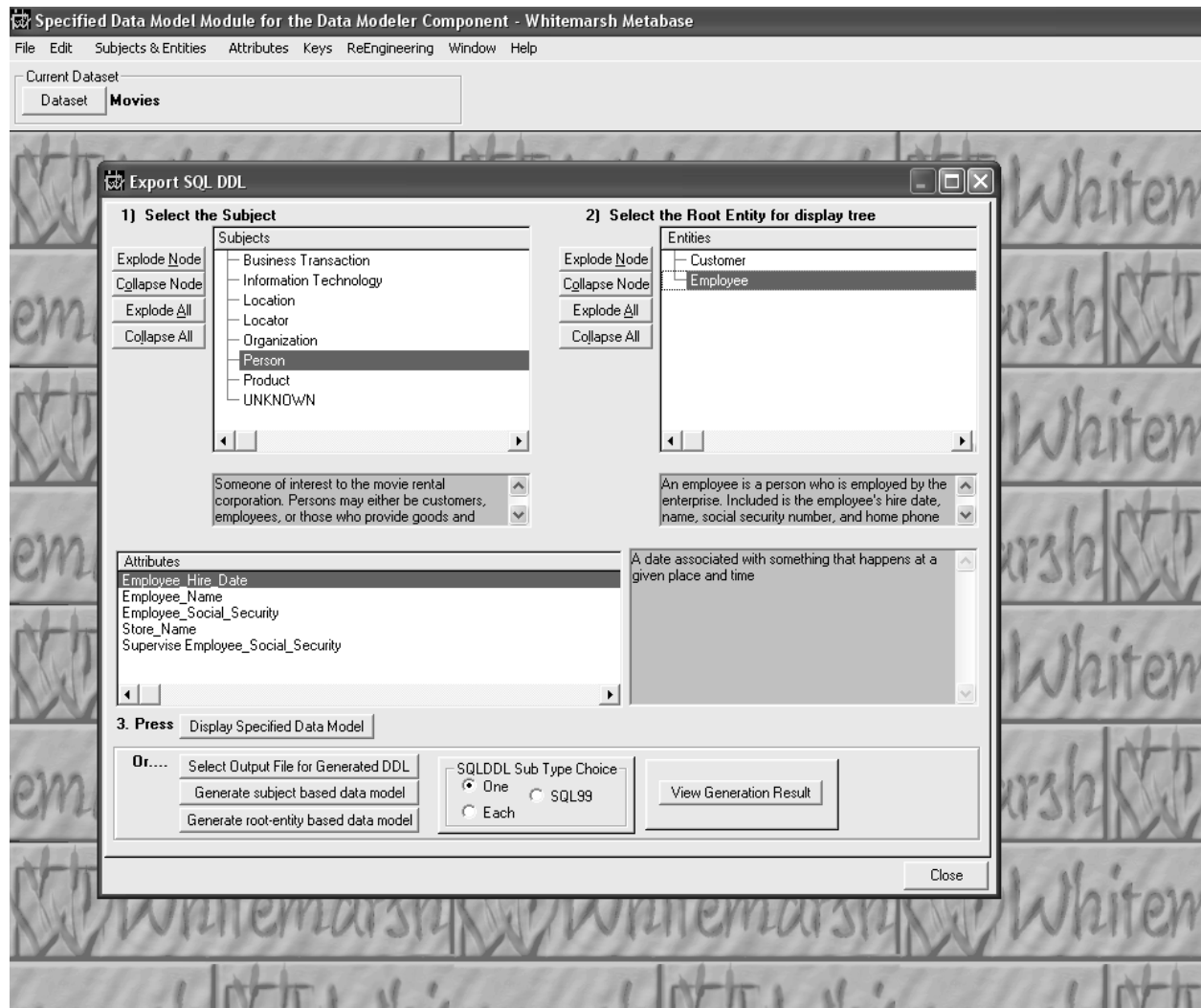


Figure 35. SQL DDL Export screen.



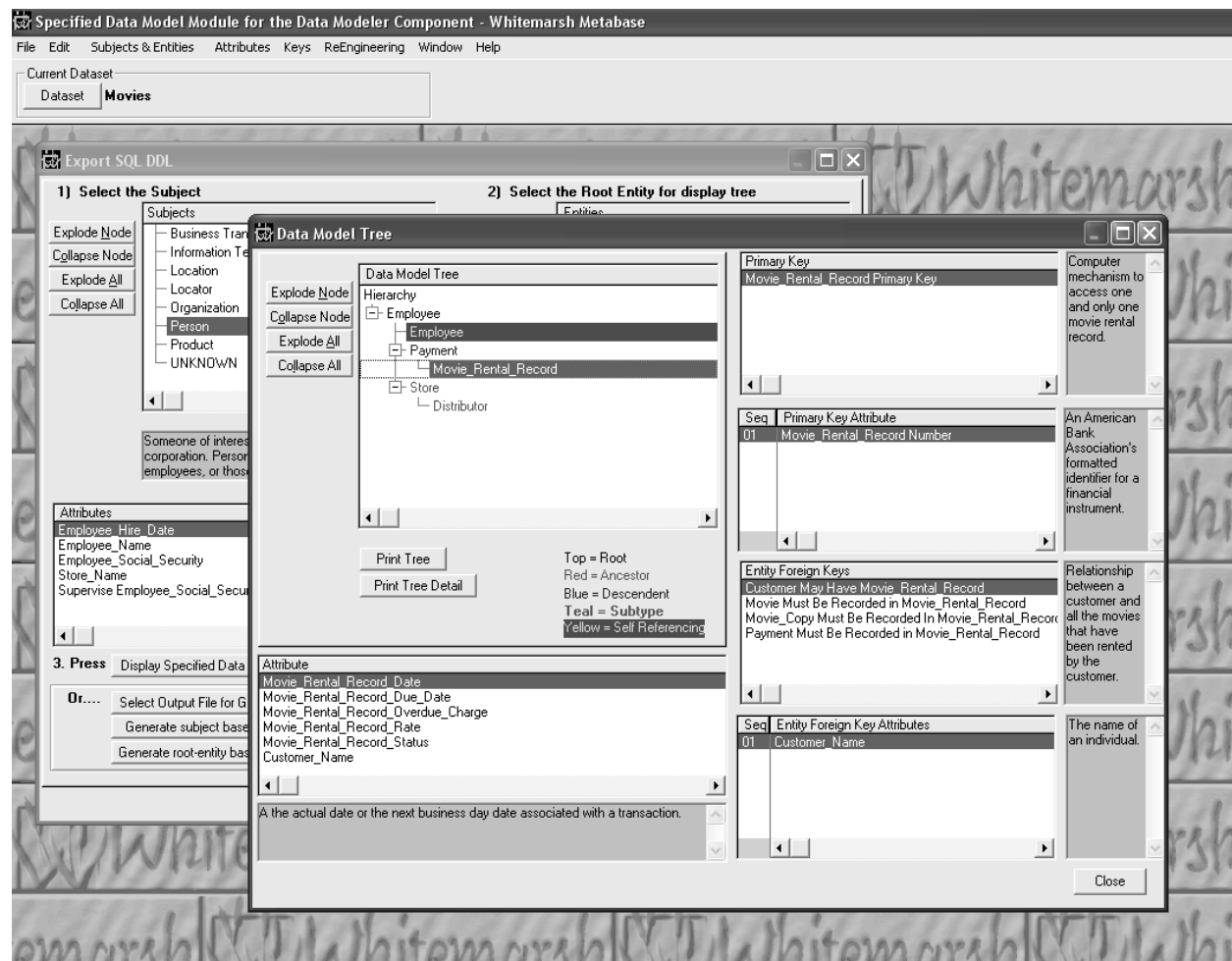


Figure 36. Displayed data model.



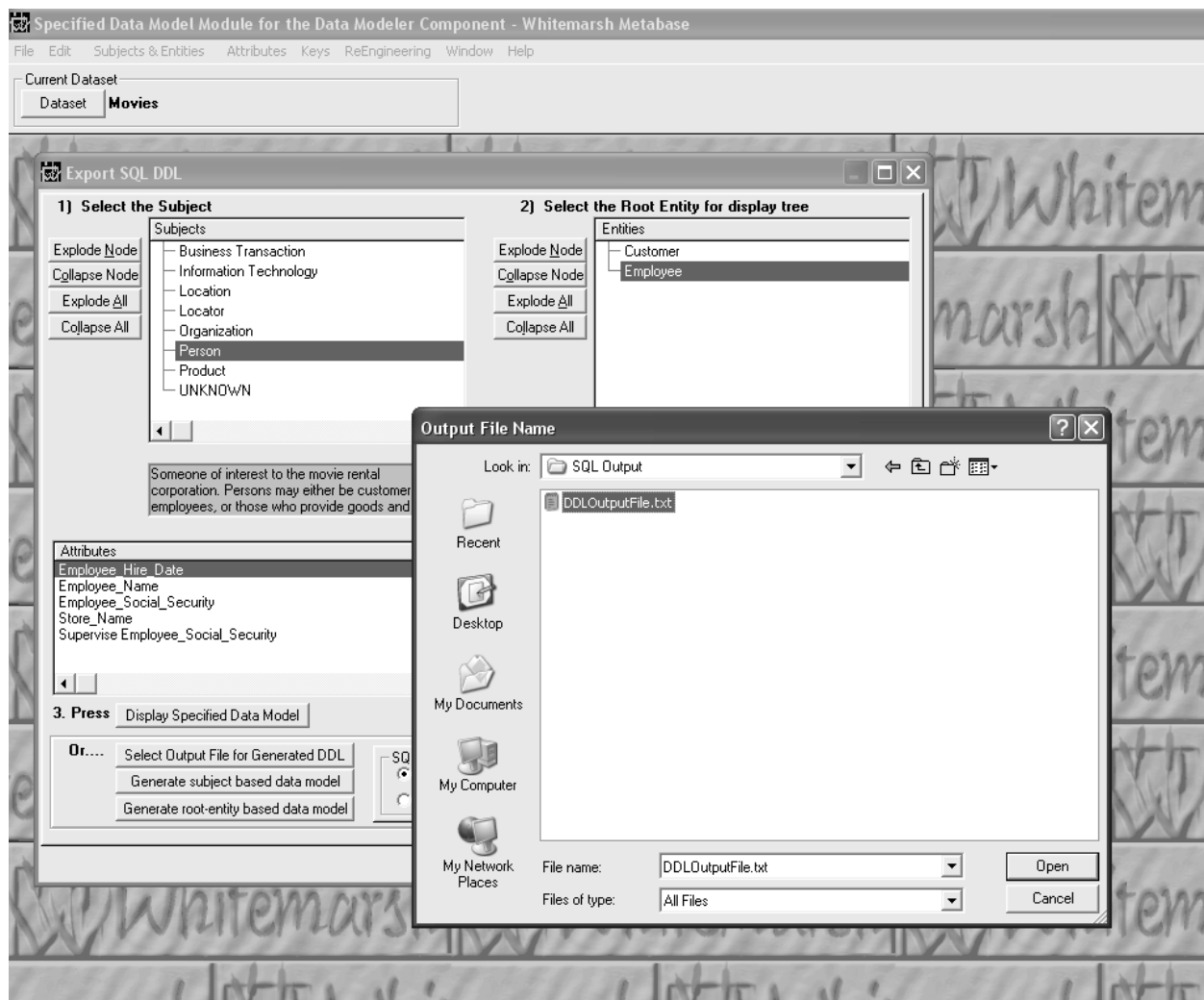


Figure 37. Selected SQL DDL output file for export.



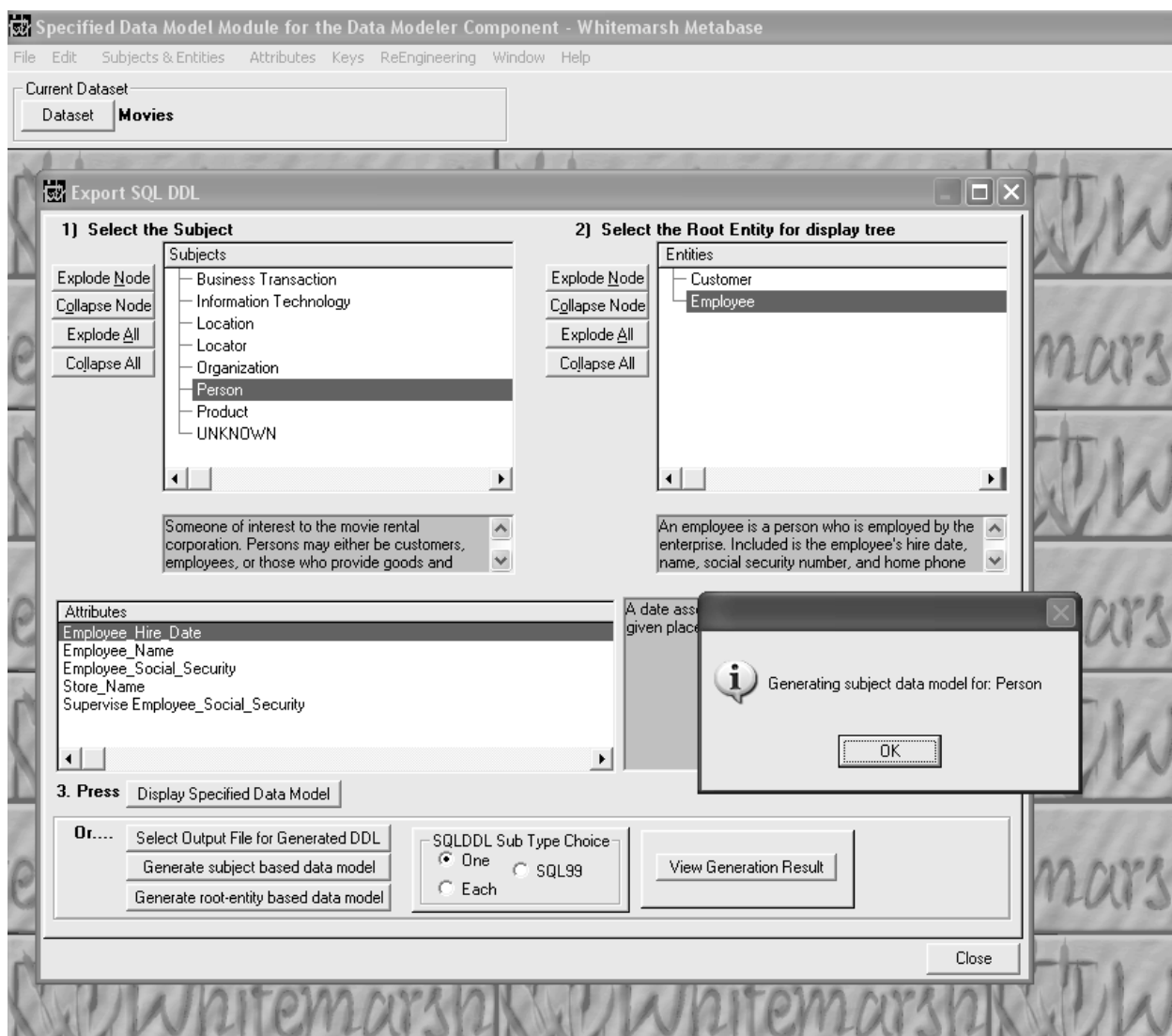


Figure 38. Generation message for creating SQL DDL for the Subject: Person.



To display the SQL DDL file, press the button, View Generation Result. Another Select File window is presented. Highlight the file from within the working directory and then press the Open button. SQL DDL as presented in Figure 39 is displayed.

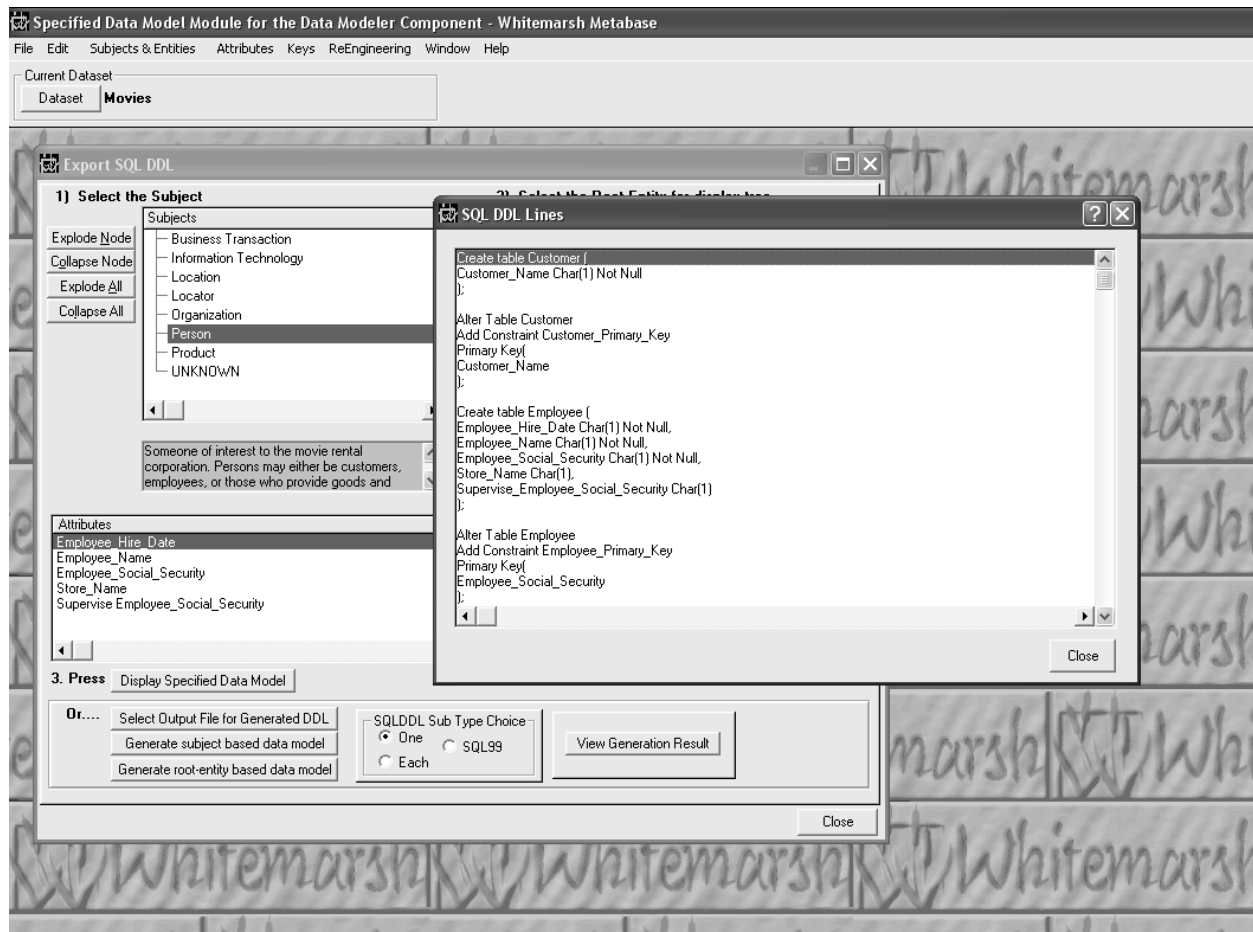


Figure 39. Displayed SQL DDL for the Person Subject area.



6.1.5.2 Import SQL DDL

The process of importing a SQL database schema into the metabase consists of:

- Creating the SQL DDL text file
- Importing the SQL DDL

When the SQL DDL is imported it will be scanned and, if acceptable, will create data records for the following Specified Data Model tables:

- Entity
- Attribute
- Primary Key
- Primary Key Column
- Unique Key
- Unique Key Column
- Foreign Key
- Foreign Key Column

6.1.5.1 Creating the SQL DDL Text File

It must be stated at the outset that the metabase will not import every possible variation of SQL DDL. There are too many variations on the theme. Focus has been on ISO/ANSI standard SQL DDL. Thus, what the metabase will import are the fundamental SQL constructs dealing with

- Table
- Column
- Primary Key in the Alter Table format
- Unique Keys in the Alter Table format
- Foreign Keys in the Alter Table format

Work is underway to import Primary and Unique keys via Constraint clauses within tables, and also various default and check constraints.

6.1.5.2 Importing the SQL DDL

The process of importing the SQL DDL consists of creating the Specified Data Model's Subject. Subject will then be the "container" within which the entities, attributes and keys will be defined. Thereafter, the actual import process can begin.

The first step is to create the subject. Figure 40 presents that screen that contains the current set of subjects. Within this screen there are a number of subjects. For the purposes of this example, create the Movies subject.



Now, to actually do the SQL DDL importing, activate the SQL import menu item. This brings up the screen in Figure 41. This screen shows that the subject, Movies exists. If it did not exist then you could create it through the Insert button. Now, select the SQL DDL file with the button, SQL DDL on the right. Browse till you find the file using the file selection dialog at the bottom of this figure. Once found, highlight the file (see Figure 42) and press the Open button. The window automatically closes and the name of the selected file appears across the middle of the window. Now, you can view the file with the View button. This is shown in Figure 43. Once you are satisfied, there's one last step. Do you want to create a log of all the actions? If so, then press the radio button yes. If you want the file "cleared" before you start importing, press the clear button. Then press the Import button.

The importing process first constructs complete SQL statements eliminating all unnecessary short lines, and the like. It then processes the file, one SQL statement at a time. As statements are successfully processed, metadata is built in the metabase. Once the process is finished, a message appears. Figure 44 shows the entities that were built for the Movies SQL DDL script. Since this is an import process, all attributes are mapped to the data element, unknown.

If there is an error, then you can view the log file to see all that did happen and what the statement was that was being processed at the time of the failure. For example, if a foreign key creation statement references a non-existing table and/or column then the load will fail. The log file is shown in Figure 45.

You should view the imported model through the process, Data Model Tree. Select the Subject, Movies, and then pick an entity, for example, Movies Rental Record, and then press the Display Data Model button. Figure 46 shows the data model tree for the Movies subject area.



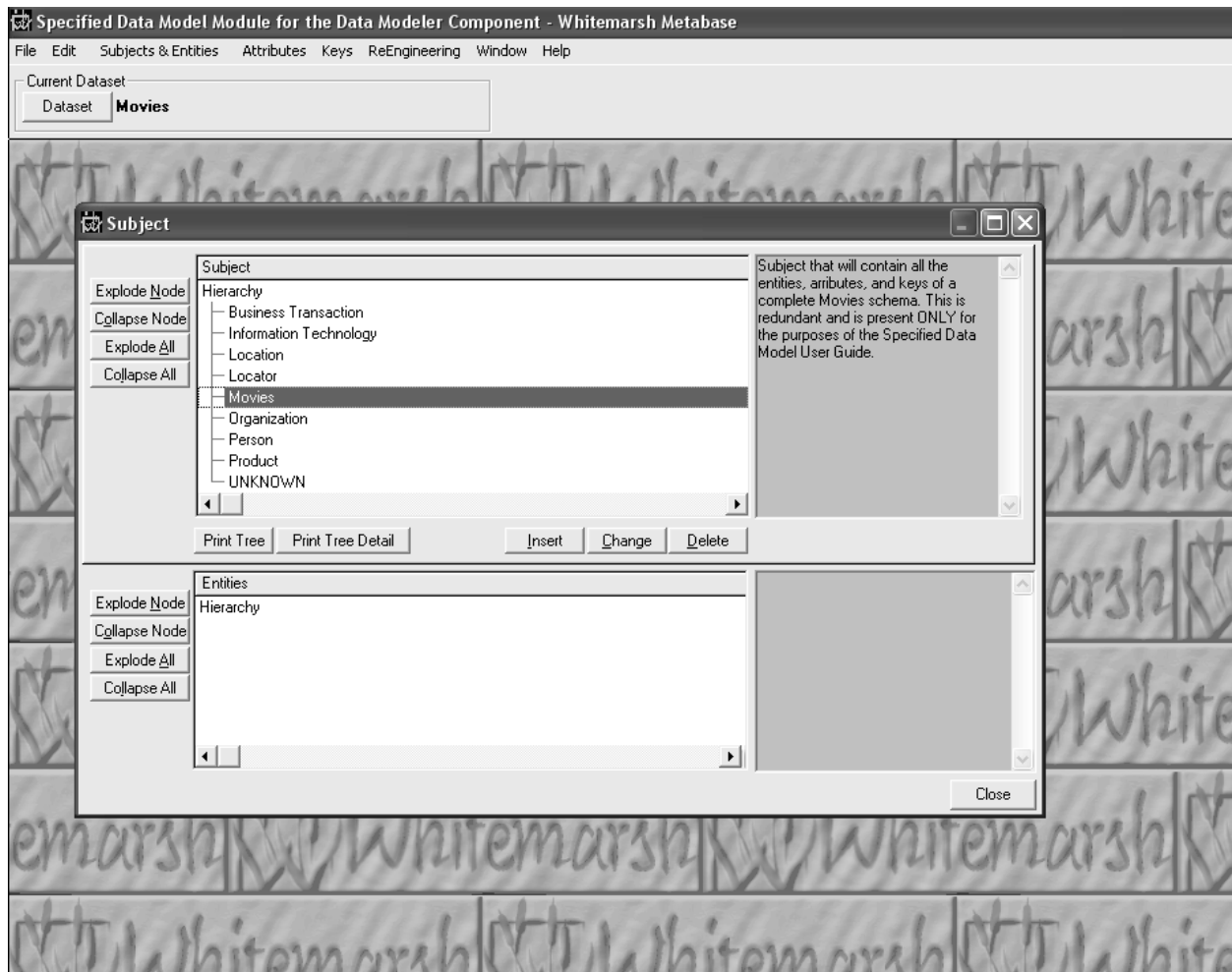


Figure 40. List of Subjects.



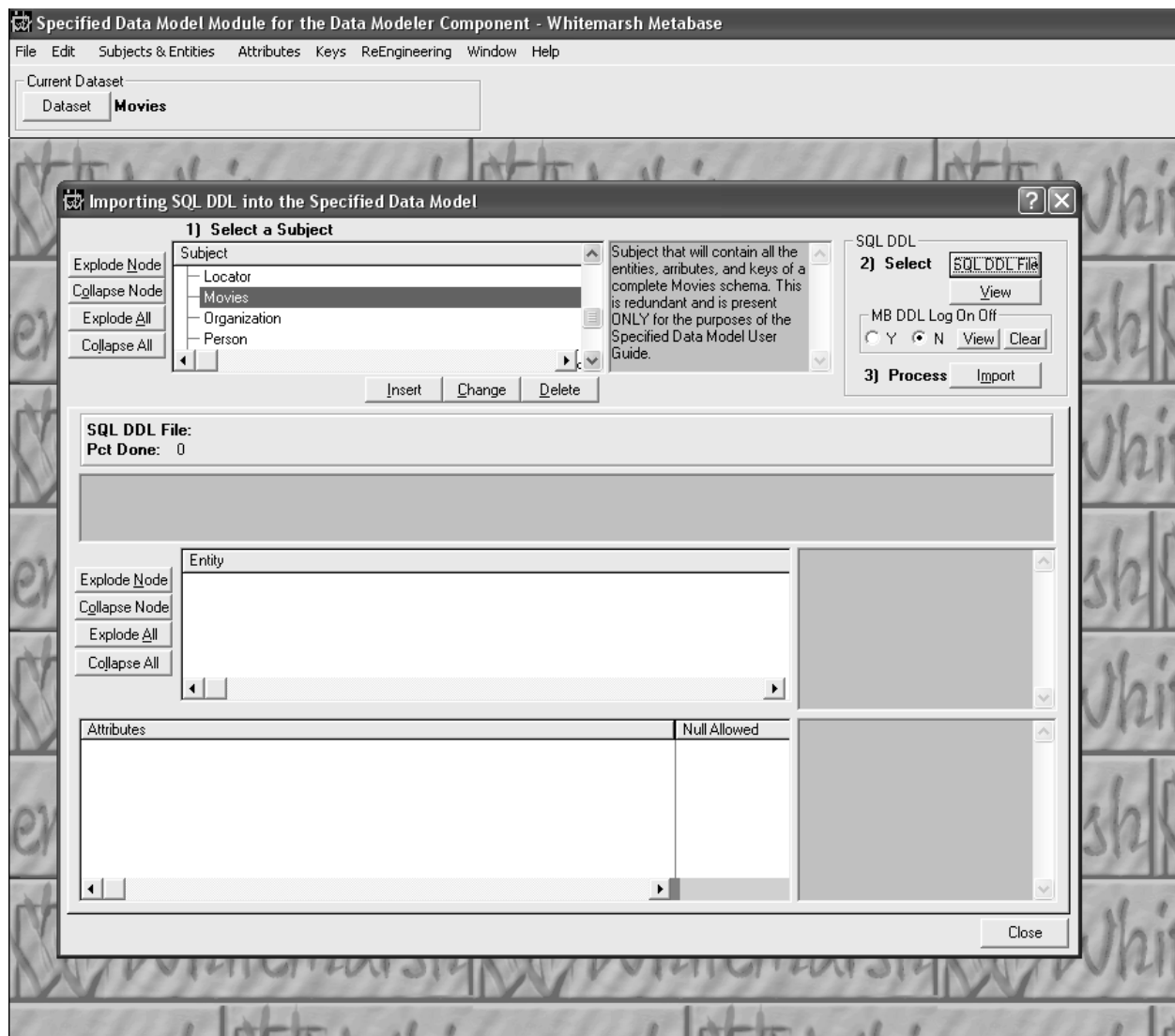


Figure 41. SQL DDL Import screen.



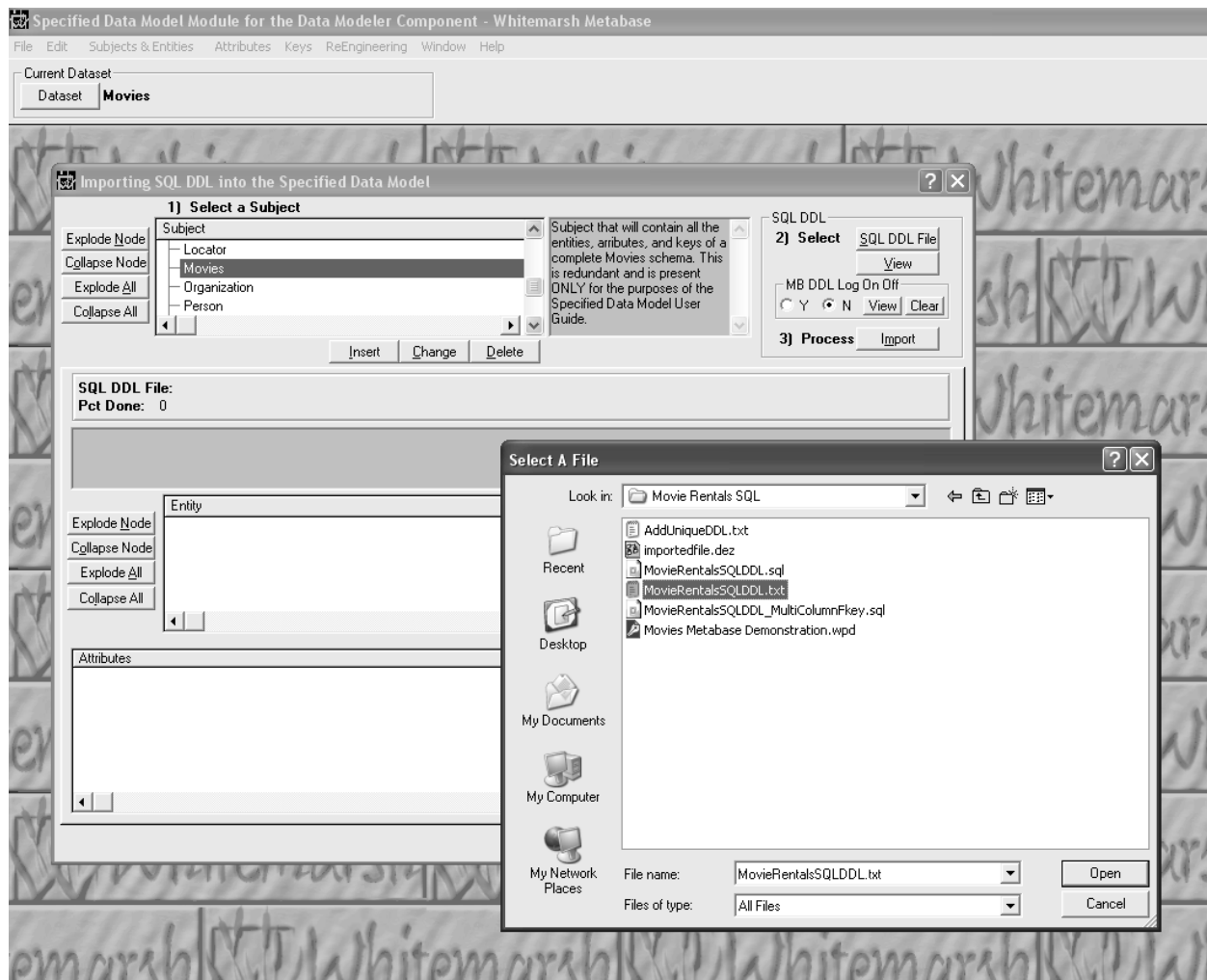


Figure 42. Selecting the SQL DDL file for importing.



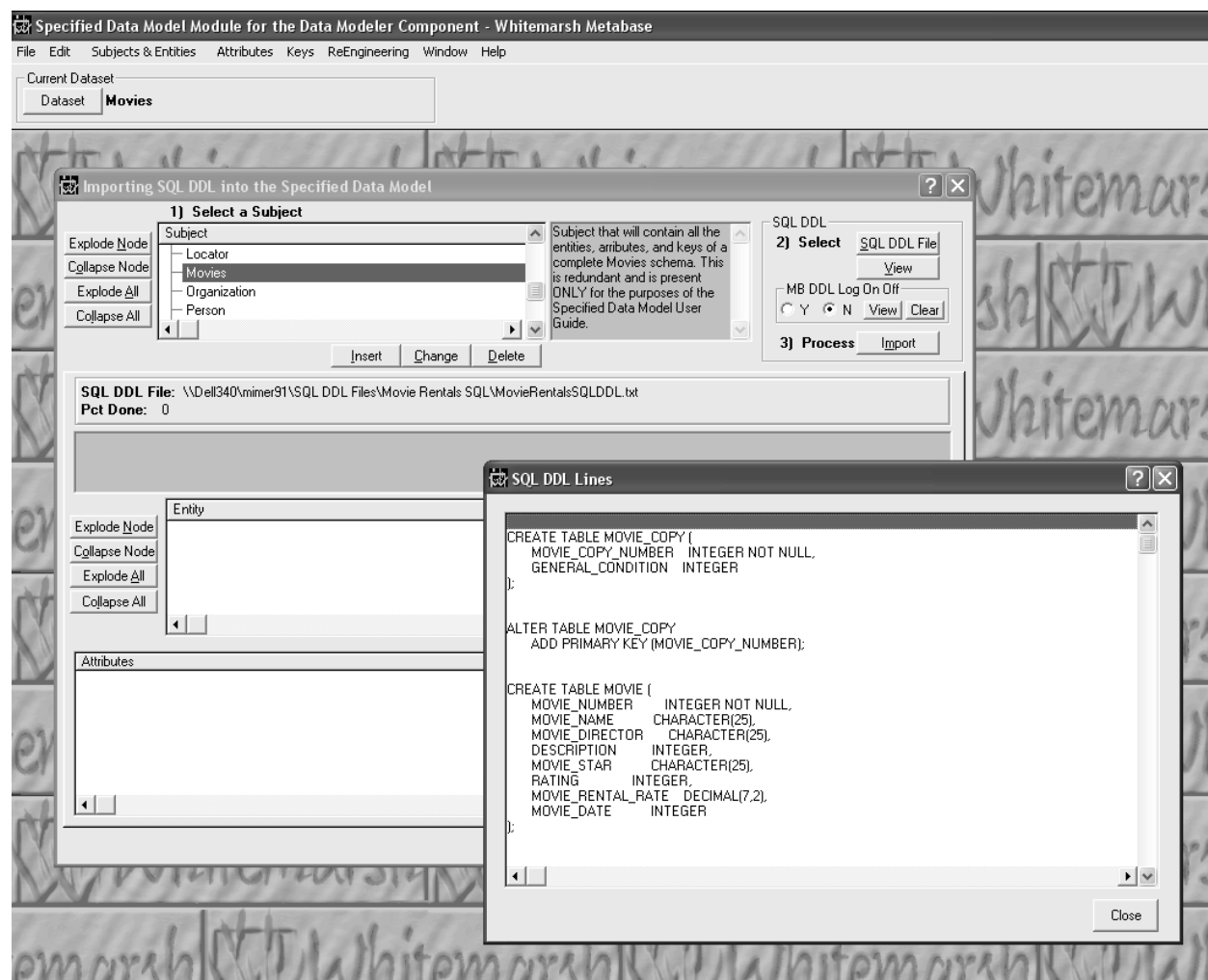


Figure 43. Viewing SQL DDL that is to be imported.



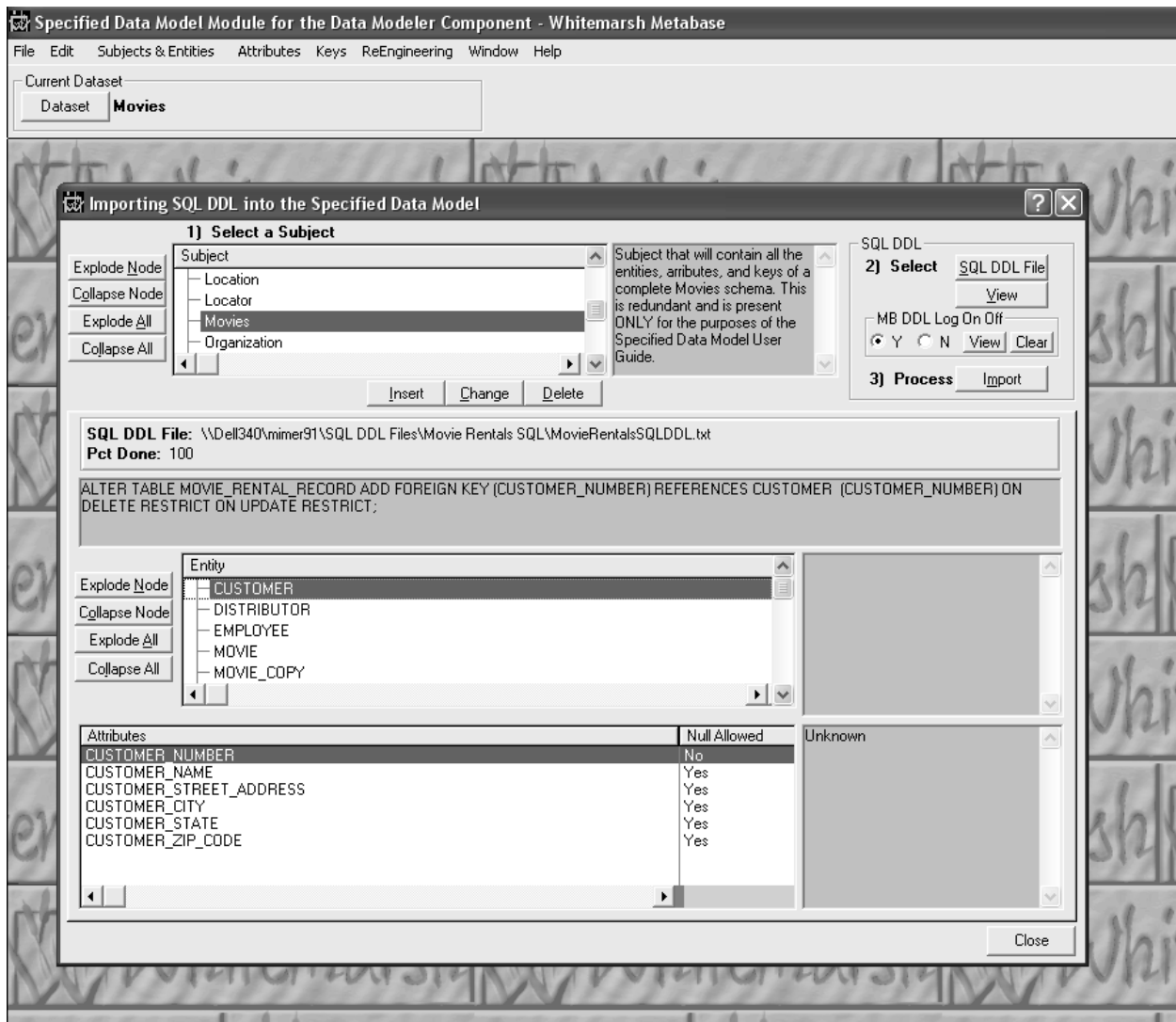


Figure 44. Entities and Attributes of Imported Movies SQL DDL.



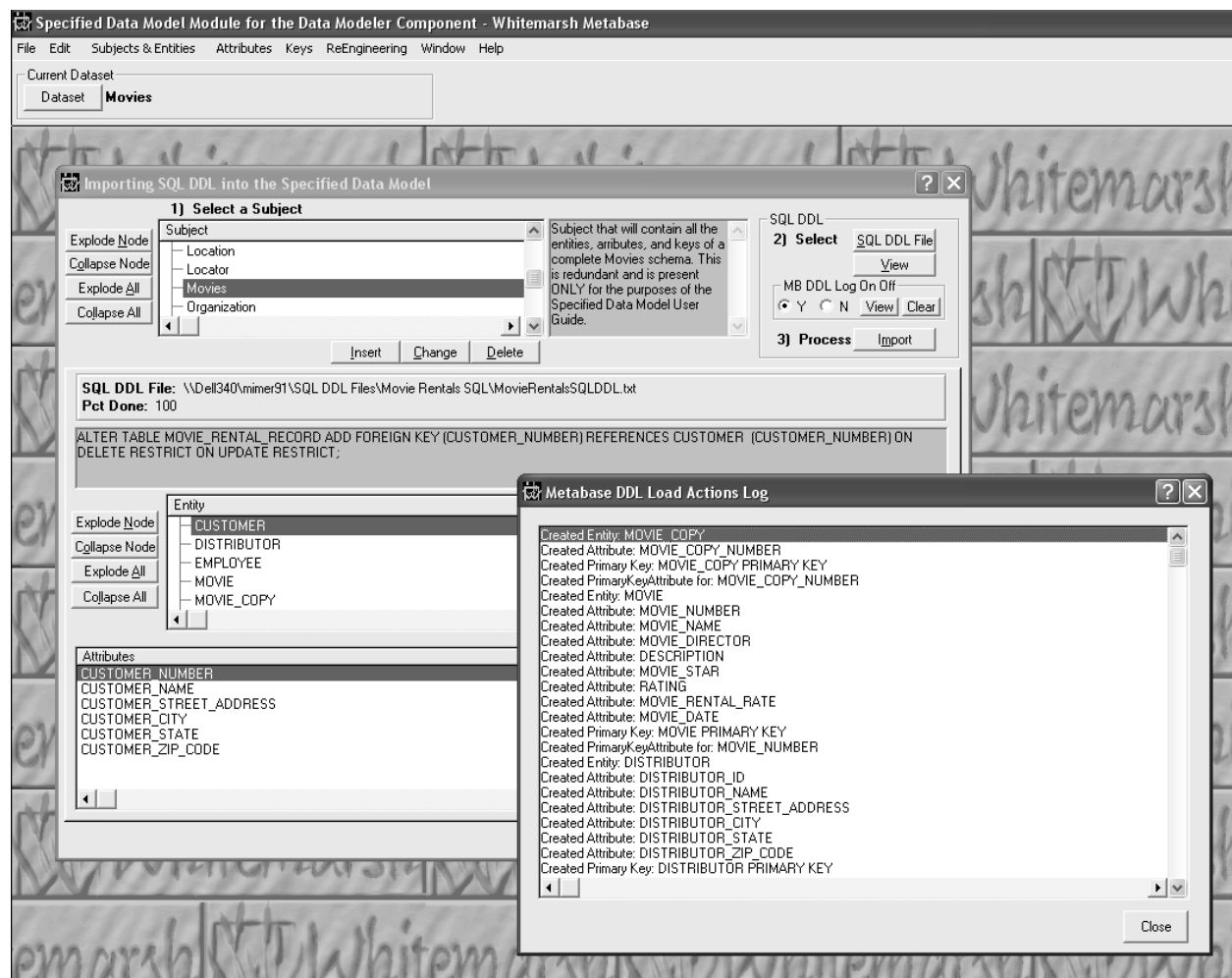


Figure 45. Viewing the SQL Import Log File.



6.1.6 Data Model Trees

The metabase can only display data in the form of trees. Entities can have descendants, ancestors, and loops. Entities are all related through primary and foreign keys. To distinguish descendants, ancestors, and loops for an entity colors are employed. While colors are not visible in this user guide, the top entity's name, around which the data model is displayed is in black text. Ancestors of that entity are shown in red. Descendants of that entity are shown in blue. Entities that loop are shown with a background of red and a type of yellow. Finally, entities that are proper subtypes of another entity are shown in teal.

Figure 46 shows the screen for selecting the root for a data model tree display. Select the subject then the "root" of the display tree. Here root means the entity around which the tree is formed. If the root is an ultimate "child" then only parent entities are shown. If the root is at the top of a tree, then only child entities are shown. If a middle entity is picked then both parents and children are shown. The paragraph above identifies all the different colors and reasons for the colors.

Figure 47 shows a displayed tree and then five browses around the tree that displays complete names and contextual definitions.



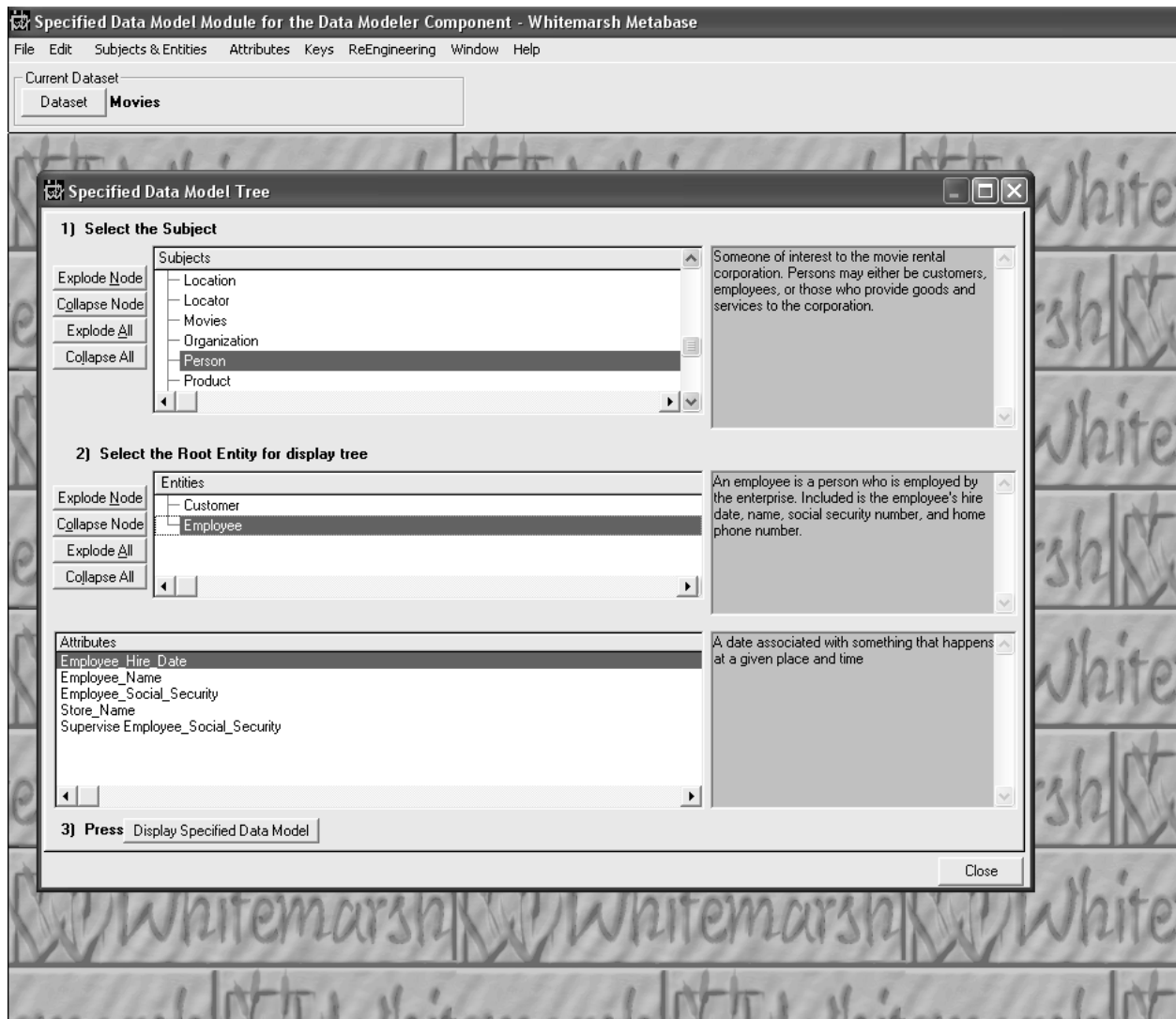


Figure 46. Selecting a root entity for displaying a data model tree.



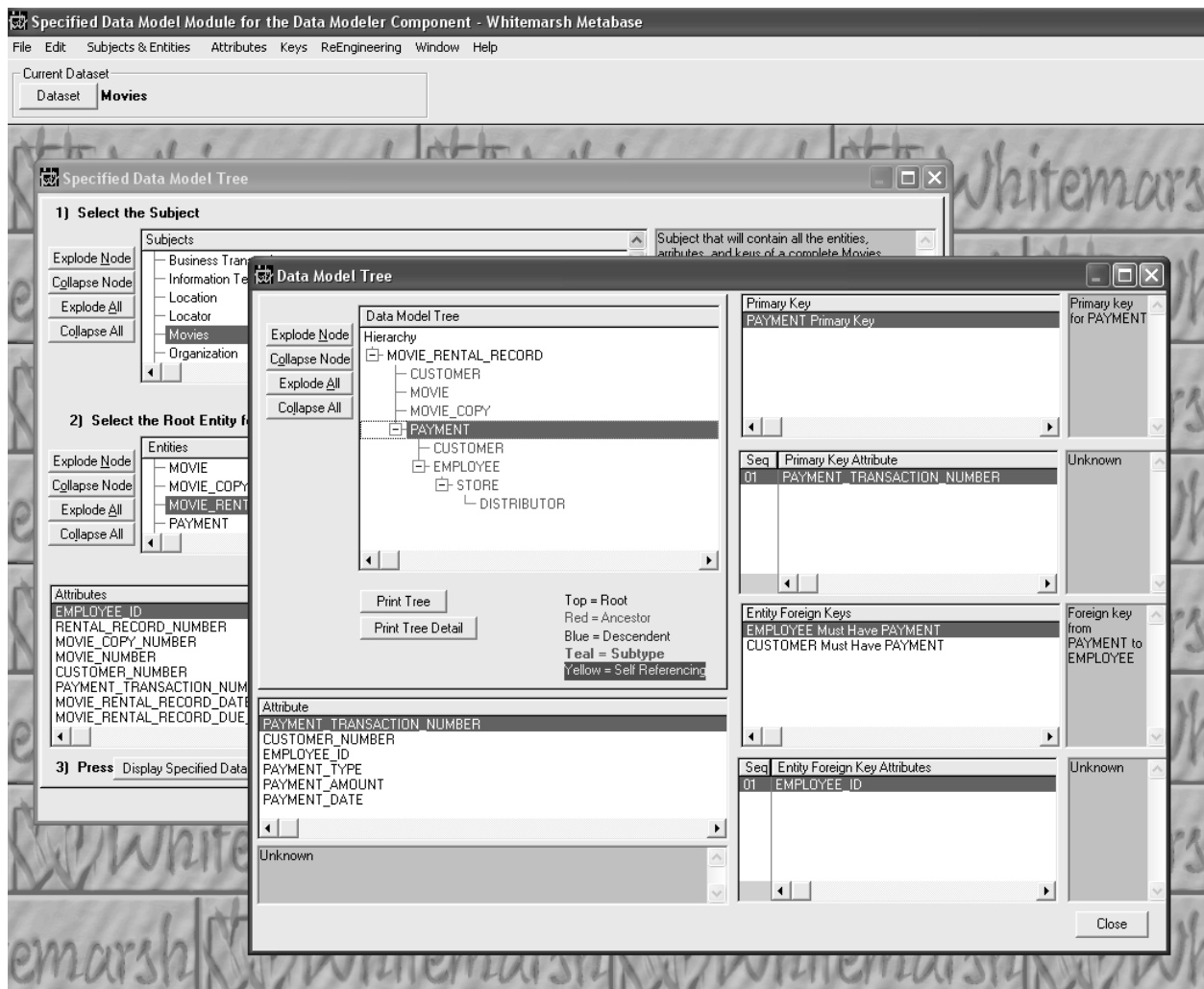


Figure 47. Movies Subject area Data Model Tree display.



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.

