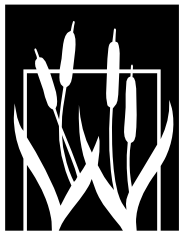

Whitemarsh
Information Systems Corporation

Whitemarsh Metabase: Database Objects 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

List of Figures

ReadMe.1st

This metabase module introduces a completely new concept, database objects. These are NOT objects stored in a database. Rather, they are object classes defined completely through DBMS facilities. In a nutshell, database objects are very robust data structures that are acted upon by processes such that they proceed through defined value states. Database objects are intrinsically data that proceeds through value state changes. An example is Employee, where the data structure for an employee may include 20+ traditional relational tables, and then the processes that take an employee object from the *null state* to the *requisition state*, ..., *assigned state*, ... *inactive state*, ... *removed state*. Whether an object can be in the "null state" is an interesting question. In the case of DBMS, it would signify that there is only DDL (i.e., metadata) and no other form of data. An object similarly would be returned to the null state when it is completely removed from the database. Null states serve the purpose of defining the two end-states of the database object. The set of all metadata for EMPLOYEE could be called EMPLOYEE CLASS. Further, all the metadata within the metabase comprises the database object's class.

The methods that effect change are two types: database object table processes, and database object information systems. A database object table process is one that acts solely against one table during a row insert, change, or delete. A database object information system is one that acts to transform a collection of rows from one or more database object tables from one value state to the next.

Various SQL based DBMSs implement database objects differently. This ranges from "not at all" to "somewhat." The SQL 1999 standard largely addresses database objects by defining data structures within the SQL schema object, Table. The SQL 2003 standard, and the unfolding SQL 2007 standard are expanding database object capabilities.

There are other materials regarding database objects on the Whitemarsh website, www.wiscorp.com. Included in these materials are methodology, book, papers, and courses.

1 Introduction

There are generally considered to be three classes of objects: display objects, wholly contained process objects, and business objects. Display objects embrace buttons on a screen, a drop list of menu choices, a graphical user interface (GUI), or complete engineering drawings. Wholly contained process objects are for example, the COSINE function, a nautical distance function that when given two geographical coordinates returns the geographic distance between them, or a well defined process that takes standard arguments and returns a specific value such as asking for the net asset value for a business given all assets and liabilities. Finally, business objects encompass business components like an insurance policy [information system] that accomplishes whole business transactions in a certain manner.

While three object classes have their proponents and detractors, what all three object classes have in common is that they are first and foremost self contained software modules/systems in the form of an executable that behaves according to certain fixed rules.

A database object is none of these. It is its own fourth class. While database objects share some common names and definitions with the other three object classes, that is, encapsulation, inheritance, and polymorphism, database objects are unique to both database and DBMS.

Database objects are identified, designed, implemented, operated through, and evolved, or maintained through just one type of data processing facility, a database management system (DBMS). If the available DBMS is an ANSI SQL3 DBMS (download SQL_BOM from the www.wiscorp.com website) then database object definition and use can be direct. Otherwise, database objects can only be indirectly approached through proprietary facilities in one or more DBMSs.

Database objects are essential to the proper understanding, specification, implementation, and maintenance of world-wide heterogeneous databases. Database objects fit within the enterprise's Knowledge Worker Framework (download the Knowledge Worker Framework book from www.wiscorp.com).

Database objects are not new. They were started in certain DBMS types (e.g., IDMS, IDS, GIM, Inquire, and Adabas) in the late 1960s. Relational DBMS such as DB2, Oracle, Informix, and Sybase, however, stopped the march to database objects dead in its tracks. It was not until the ANSI SQL3 data model moved away from relational and not until a whole programming language was incorporated into ANSI SQL3 that the march to database objects restarted. The newest versions of IDMS, DB2, Oracle, Informix, and Sybase have all started to support the data structure and process features essential for database objects. Even if the twenty-year delay had not happened, computers, networks, languages and operating systems were just not sophisticated enough to make database objects successful.

Database objects were formulated almost 20 years ago by the late Matt Flavin in his 1979 Yourdon Monograph, *Fundamentals of Information Modeling*. During the Seventies, Matt who worked for Infodata of Rochester, NY and Fairfax, VA. Infodata accomplished very early database management system research and development. Infodata's DBMS, Inquire, was widely used in the U.S. Federal Government. Matt represented Infodata to the X3H2, the ANSI Database Languages committee in the late Seventies.

Database objects existed only on paper only until the ANSI database languages committee, X3H2, working since early 1993 on the specification of SQL, formulated the essential linguistic components of database objects.

Distributed, client/server data processing either through traditional or Internet processing is here to stay, and rightly so. Not only are they empowering, these are essential because enterprises are highly distributed and world wide. Enterprises must be able to respond to local needs, laws, customs and mores. But, if business are designed and tuned to respond to local situations, how can they act in concert within their world wide communities? How can you have world-wide consistency and semantics without suffocating local needs and practices? How can both ends of the information resources spectrum be satisfied?

Business data needs far exceed today's DBMS's two dimensional table capabilities and simple column based constraints. Businesses cry out for semantically rich data management to meet business needs across world-wide, heterogeneous hardware and operating system environments. Business data management environments must behave consistently regardless of their host computing hardware environment, operating systems, or DBMS vendors, and must be easy to specify, implement, use and maintain.

Businesses require hierarchies of complex data tables, collections of integrated rules for data integrity, well defined procedure sets, and fixed transformations that move a business policy–data is just executed policy--from one well defined state to another. Examples of business needs include insurance policies and claims, court cases and documents, public safety incidents, sales and marketing databases that contain customers, sales organizations, forecasts, orders, deliveries, and product sales statistics, inventory control and deployment, and human resources.

Database objects "live" entirely within the domain of the DBMS. Database objects can be both persistent and non-persistent, and can span single or multiple-tables.

A persistent database object is one that is stored and is retrievable over long periods of time. An example is an insurance policy along with its full compliment of payments, renewals, and claims. Another example of a persistent object are the rotating-three dimensional views of a mechanical part.

A non-persistent database object is one that is materialized and displayed but is not able to be re-materialized because some of its components are not retained after the database object's display is terminated.

Non-persistent objects are dynamically produced from database data and exist only for the life of their "display." An example might be evening news weather displays. The weather map, that is, the states, cities, streams, rivers, etc are all persistent database data. The actual streams of clouds, high and low pressure fronts, cloud formations, and the like are time-sequenced BLOBs (binary large objects) that are dynamically displayed across the screen. While the displayed database objects may be recorded via videotape and redisplayed at a later time, the detailed components, which upon retrieval make up the non-persistent database objects, is not stored.

By the time the news cast is over, the BLOB parts are discarded. Other than for a videotape replay, the complete set of the non-persistent database objects are gone. The persistent part is traditional data structures with the appropriate quantity of indexes. The BLOBs are just non-indexed streams of binary data that are stored in a very primitive format.

Persistent database objects are those that are stored in a database on a permanent basis. Included are traditional "relational" data, abstract data types of complex structures (like an entire auto accident claim that might include BLOBs, free text streams, etc.).

Single table database objects are those that are fully defined within a single row of an SQL/1999 table structure. With SQL/1999, columns can support very complex structures, such as simple values, lists, sets, multi-sets, and abstract data types of arbitrary complexity. This capability is quite common in hierarchical DBMSs like System 2000 and in independent logical file data model DBMSs such as Adabas, Model 204, Inquire, and Datacom/DB.

For example, in a product sales database, the single table called sales has product number as the primary key with other columns for product name, product description, and the like. The sales table sales column in contrast, contains product sales by year by month by region, district and territory by salesman. That's a single column with six dimensions of values. Prior to SQL/1999 such product sales information would require multiple tables with the attendant keys, joins, and computer processing melt-downs. Since the salesman's object identifier is contained as a integral component of the sales data, the salesman's full set of data is accessible through normal SQL language processing. A referenced database object, that is, the referenced salesman's data is not considered a formal part of a single table database object.

Multi-table objects are those that are implemented across multiple tables. For example, an insurance policy may have several dozen tables that make up its full definition. One and only one table is considered as the database object's root table. The database object root table contains among many things, the object's identity column. A row from the root table is the head-row of the database object. All other related tables within the multi-table database object contain other information related to the object. In the insurance policy example, a claim might be a contained in one or more database object tables. Other database object tables would contain the underwriting information regarding the person about whom the policy was issued. The person is a different single or multi-table database object. Each table within a multi-table object may consist of single valued columns, it can also support lists, sets, multi-sets, and abstract data types of arbitrary complexity.

Not only do most businesses contain multi-table database objects, the majority of business applications are examples of multi-table database objects. A quick look at business applications reveals that inescapable conclusion.

Database objects, regardless of persistence and regardless of whether single or multi-table contain the same four-part composition:

- Data Structure: the set of data structures (simple and complex collections of tables) that map onto the different value sets for real world database objects such as an auto accident, vehicle and emergency medicine incident.

- Database Object Process: the set of database object processes that enforce the integrity of columns (simple or complex), references between database objects and actions among contained data structure segments, the proper computer-based rules governing data structure segment insertion, modification, and deletion. For example, the proper and complete storage of an auto accident.

- Database Object Information System: the set of specifications that control, sequence, and iterate the execution of various database object processes that cause changes in database object states to achieve specific value-based states in conformance to the requirements of business policies. For example, the reception and database posting of data from business information system activities (screens, data edits, storage, interim reports, etc.) that accomplish entry of the auto accident information.

- Database Object State: The value states of a database object that represent the after-state of the successful accomplishment of one or more recognizable business events. Examples of business events are auto accident initiation, involved vehicle entry, involved person entry, and auto accident DUI (driving under the influence of alcohol/drugs) involvement. Database object state changes are initiated through named business events that are contained in business functions. The business function, auto accident investigation includes the business event, auto-accident-incident initiation, which in turn causes the incident initiation database object information system to execute, which in turn causes several database object processes to cause the auto accident incident to be materialized in the database.

A database object is specified to the SQL/1999 DBMS through the SQL/1999 definition language (DDL). All four components of a database object operate within the "firewall" of the DBMS. This ensures that database objects are protected from improper access or manipulation by 3GLs, or 4GLs.

This metabase users guide is coupled with a book, course titled, *Database Objects, The Foundation Stones of Enterprise Database*. Readers of this metabase users guide encouraged to become members of the Whitemarsh website, www.wiscorp.com to obtain the full set of database object materials. This metabase users guide is all about how to define, update, and report database objects.

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 database for database objects consists of 16 different tables. Of these 16, six tables are read only tables as they are created in other metabase modules. The tables that are created within database objects are:

- Database Domain and Database Object
- Database Object
- Database Object Information System
- Database Object Information Systems and Database Object Process
- Database Object Process
- Database Object Process Column
- Database Object State
- Database Object State and Database Object Information System
- Database Object Table
- Membership Rational

The tables that are created in other metabase modules are:

- Column (created in the implemented data model module)
- Data Element (created in the data element module)
- Database (created in the implemented data model module)
- Database Domain (created in the mission, organization, function module)
- Schema (created in the implemented data model module)
- Table (created in the implemented data model module)

Of these six, five are created within the data modeling metabase module. The database design is presented in Figure 1.

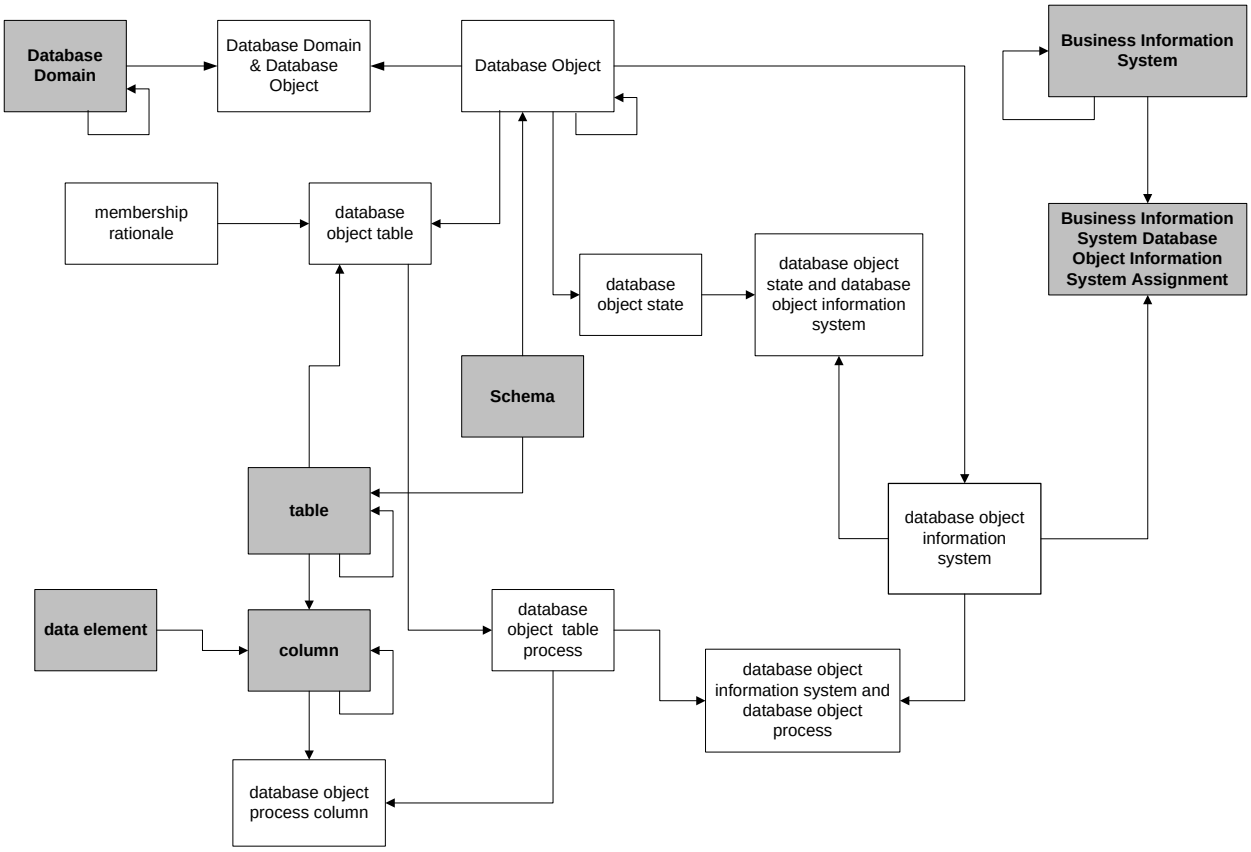

Figure 1. Database Objects Meta Model.

Explicit in this database are the following structures:

- A database object is a collection of one or more tables. The rows from these tables are transformed from one valid state to another via database object table processes and database object information systems. Database objects are related to one or more database domains.

- A Database Domain and Database Object is the association of one or more database objects with one or more database domains.

- A database object is a collection of one or more tables. The rows from these tables are transformed from one valid state to another via database object table processes and database object information systems. Database objects are related to one or more database domains.

- A Database Object Information System is a collection of processes defined within the domain of the DBMS usually as a stored procedure that transforms one or more rows of a database object from one valid state to another. A database object information system accomplishes one or more database object table processes.

- A Database Object Information Systems and Database Object Process is the association of one or more Database Object Information Systems and Database Object Processes. These are set into a sequence for proper accomplishment.

- A Database Object Table Process is a process such as insert, change, or delete that occurs against one row of a single table within a database object. A table owns (and is thus acted upon by) one or more database object table processes. A database object table process may be invoked by one or more database object information systems.

- A Database Object Table Process Column is an association of a specific database object table process and a specific column of a table.

- A Database Object State is a well defined value state of a database object. States occur in a particular sequence, typically from the null state through a set of value states and returning to a null state. A database object state is accomplished through one or more database object information systems.

- A Database Object State and Database Object Information System is the association of one or more Database Object States and Database Object Information Systems. The association exists within a sequence so that the state is properly achieved.

- A Database Object Table is an association of a table with a database object. Membership rationale classifies the reason why a table belongs to the database object.

- Membership Rational is a classification of the reasons why a table belongs to a database object as evidenced through the database object table.

- Schema and database object is the association between a schema and a database object. This means that a database object may reside in more than one schema.

These statements must be carefully studied as they represent the paradigm of database objects. Because a table can be in multiple database objects, this paradigm support multiple inheritance. In this special case, there are only two roles for tables. The first is a proper table, such as job-skills of an employee. The other is a referenced table, that is, the job-skills-types. Reference tables are merely abstraction representations, and could be completely encapsulated. They are not so as to achieve data standardization across multiple database objects within the enterprise.

4 Reference Data

The only reference data table is membership rationale. The purpose of membership rationale is to identify the reason why a table is included in a particular database object. Since there is only one reference data table it has been included in the database object Data Structure menu item.

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

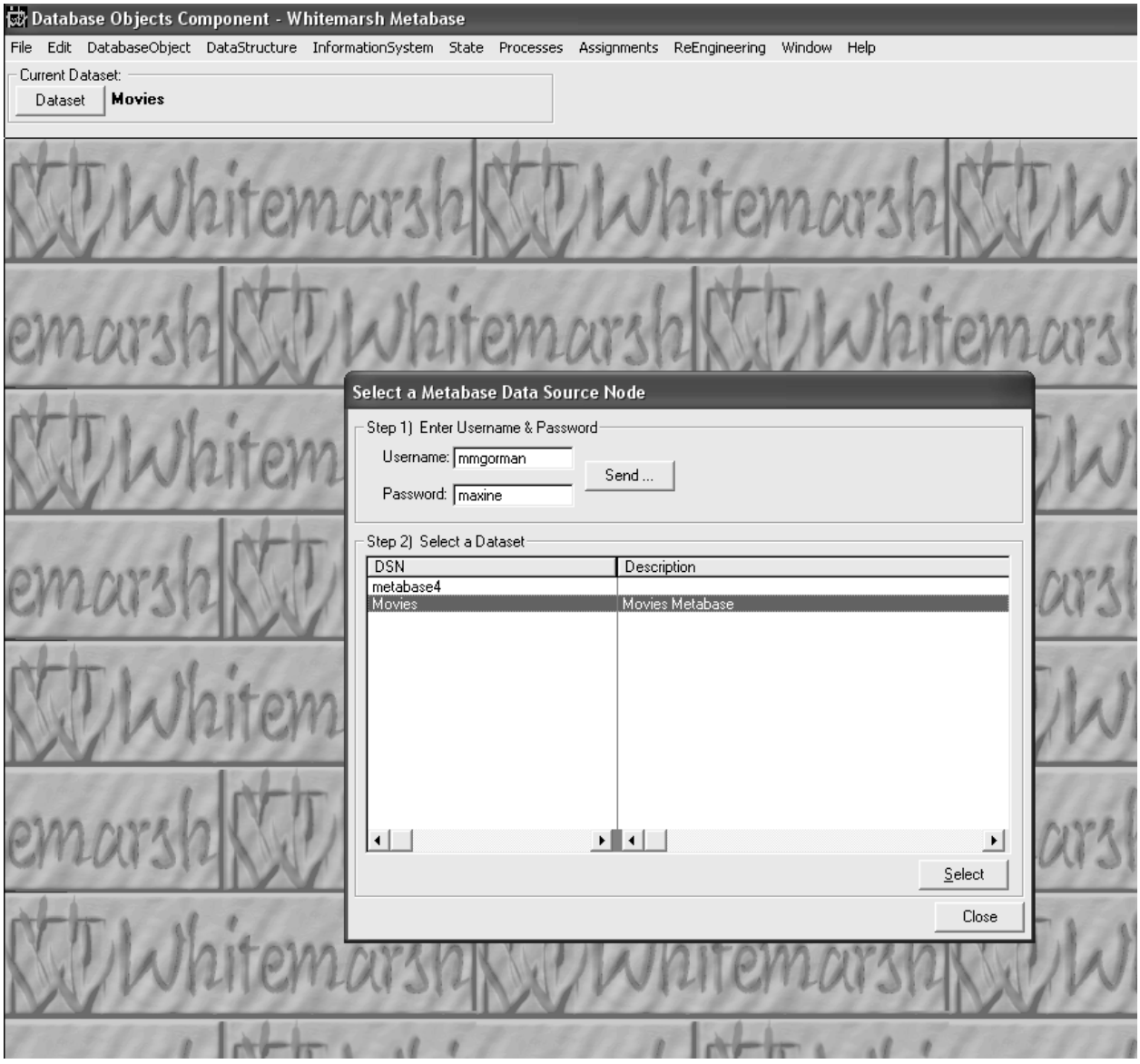

Figure 2. Login screen.

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.

6 Process Model

The information needs analysis process model consists of four classes of processes:

- Reference Data
- Fact Data Entry
- Assignments
- Reports

The top level menu for database objects contains the following top level items:

- Database Object– defines the specific database objects
- Data Structure– database object tables and rationale for database object table membership in a database object.
- Processes–defines the database object processes for a database object
- Information Systems–defines the database object information systems
- State–defines the states through which the database object proceed
- Assignments–interconnects the various components of database objects. For example, the database object information systems to a particular database object state
- Reports–the canned and table based reports produced by the database objects metabase module.

A complete menu is provided in the table that follows:

-- DatabaseObject
    -- Database Objects

-- DataStructure
    -- Database Object Tables
    -- Database Object Table Membership Rationales

-- InformationSystem
    -- Database Object Information System

-- State
    -- Database Object State

-- Processes
    -- Database Object Tables Processes

-- Assignments
    -- Assign Database Domains to Database Objects
    -- Assign Database Objects to Tables
    -- Assign Database Object Table Processes to Columns
    -- Assign Database Object State to Database Object Information System
    -- Assign Database Object Information System to DBO Table Processes

-- ReEngineering
    -- Reassign Database Object to Database Domain
    -- Reassign Database Object to Schema
    -- Reassign Database Object Tables
    -- Reassign Database Object Table Process Columns
    -- Reassign Database Object State to Database Object Information Systems
    -- Reassign Database Object Processes to Database Object Information Systems

Menu for Database Objects

6.1 Reference Data

The DBO reference data consists only of membership rationale. This is included under database object data structure in the Fact Data section below.

6.2 Fact Data

Database object fact data consists of entering the following:

- Database Objects
- Database Object Table Structure (menu: Data structure)
- Database Object Table Process (menu: Process)
- Database Object Information System (menu: Information systems)
- Database Object State (menu: State)

6.2.1 Database Objects

Database objects consist of the following:

- Database object creation
- Database-object database-object domain assignment
- Database-object schema assignment
- Reassign of database object to database domain
- Reassign of database object to schema

6.2.1.1 Database Objects

The browse list and data entry screen for database objects is shown in Figure 3. The information necessary about the database object itself is its name and description. All other information, that is, its data structure, processes, information systems, and states is entered in other screens. Because database objects can be hierarchical, then to create a subordinate database object, highlight the parent database object and press the Insert button. If the database object being created is at the top of a database object tree then just highlight the “hierarchy” entry and press Insert. In either case, Figure 4 presents the update screen for the database object. Enter the appropriate information and press enter. Prior to creating a new database object its containing schema must be selected (see Figure 3). Once the schema is chosen for a database object, the metabase software ensures that all assigned database object tables belong to the same schema.

In this example, Movies, the majority of the database objects only contains one table. Typically, however, database objects contain 10 to 15 tables. The reason they do not in this example is because this example is restricted to a small database design for movie rentals in a

movie store. There is also a data warehouse for this same application. Given that most databases have about 200 tables, there is likely to be between 15 to 20 database objects. “Seeing” them on a well drawn entity relationship diagram is quite easy. Focus on those entities that have a root entity and then a collection of “child” entities. That collection is most likely a database object.

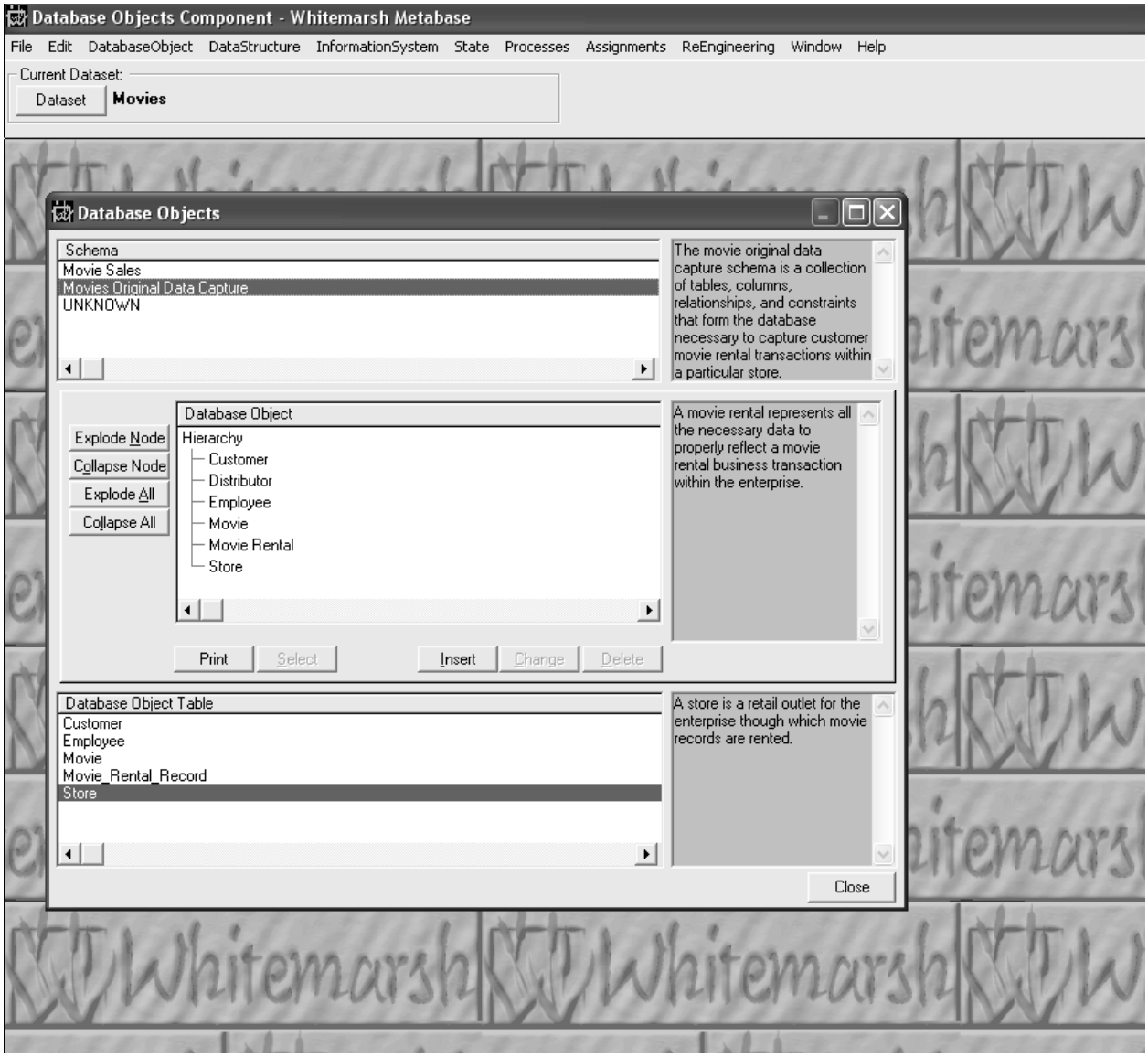

Figure 3. List of Database Objects.

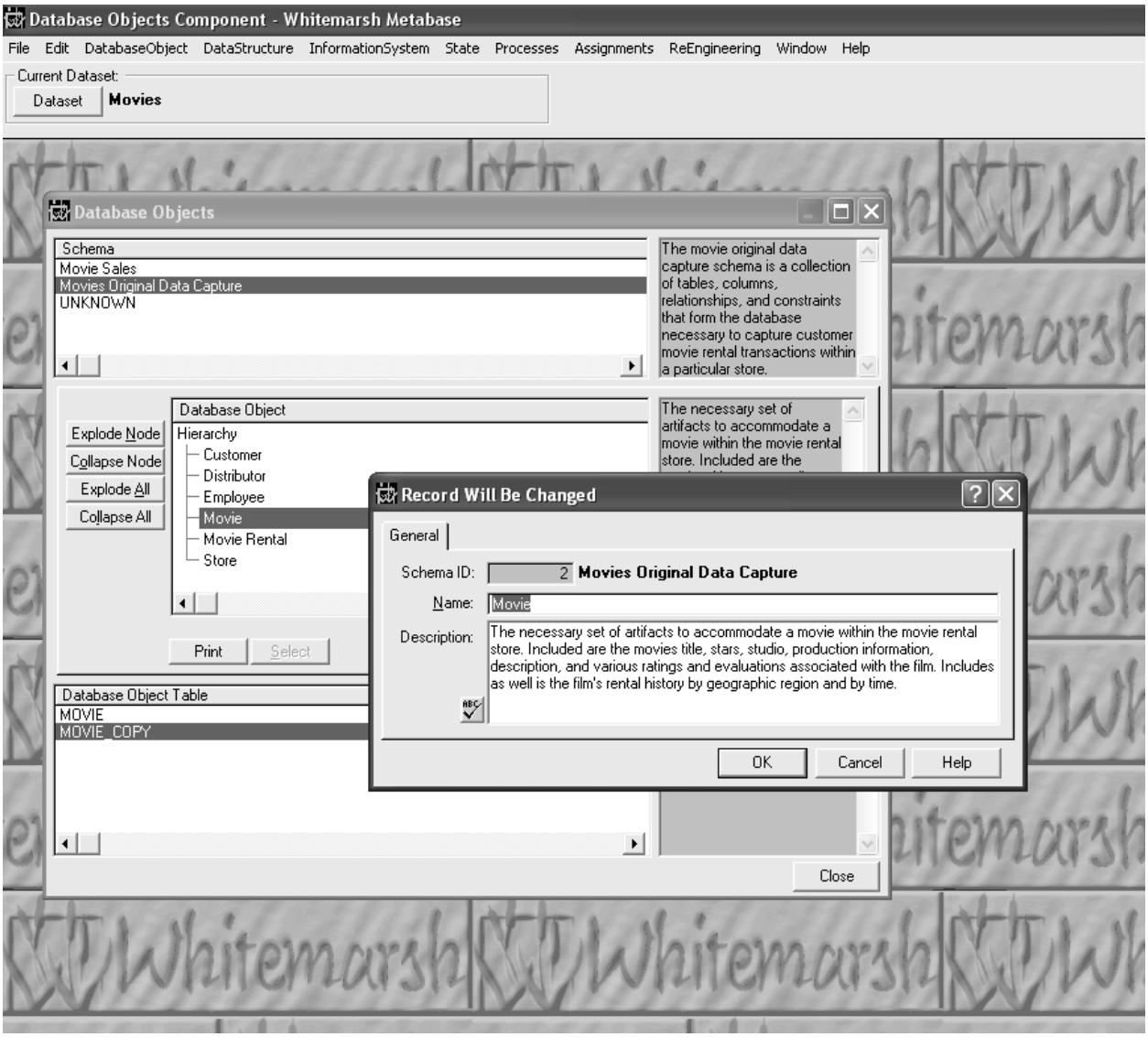

Figure 4. Database Object update screen.

6.2.1.2 Database Domain Database Object Assignment

Database objects do not exist in isolation. They are discovered through an analysis of the set of all nouns enumerated within the development of database domains. Figure 5 presents the assignment screen for allocating database objects to database domains. In this window, database objects are specifically assigned database domains. In this window, the particular database object is identified and tagged in the upper left window. One only database object should be tagged. If more than one is tagged, they will be ignored. Then, tag as many different database domains as appropriate. Finally, press the Build button. The database domains that are then associated with the database object will appear in the bottom window. If any records already exist in the bottom window, they have already been assigned to the highlighted database object.

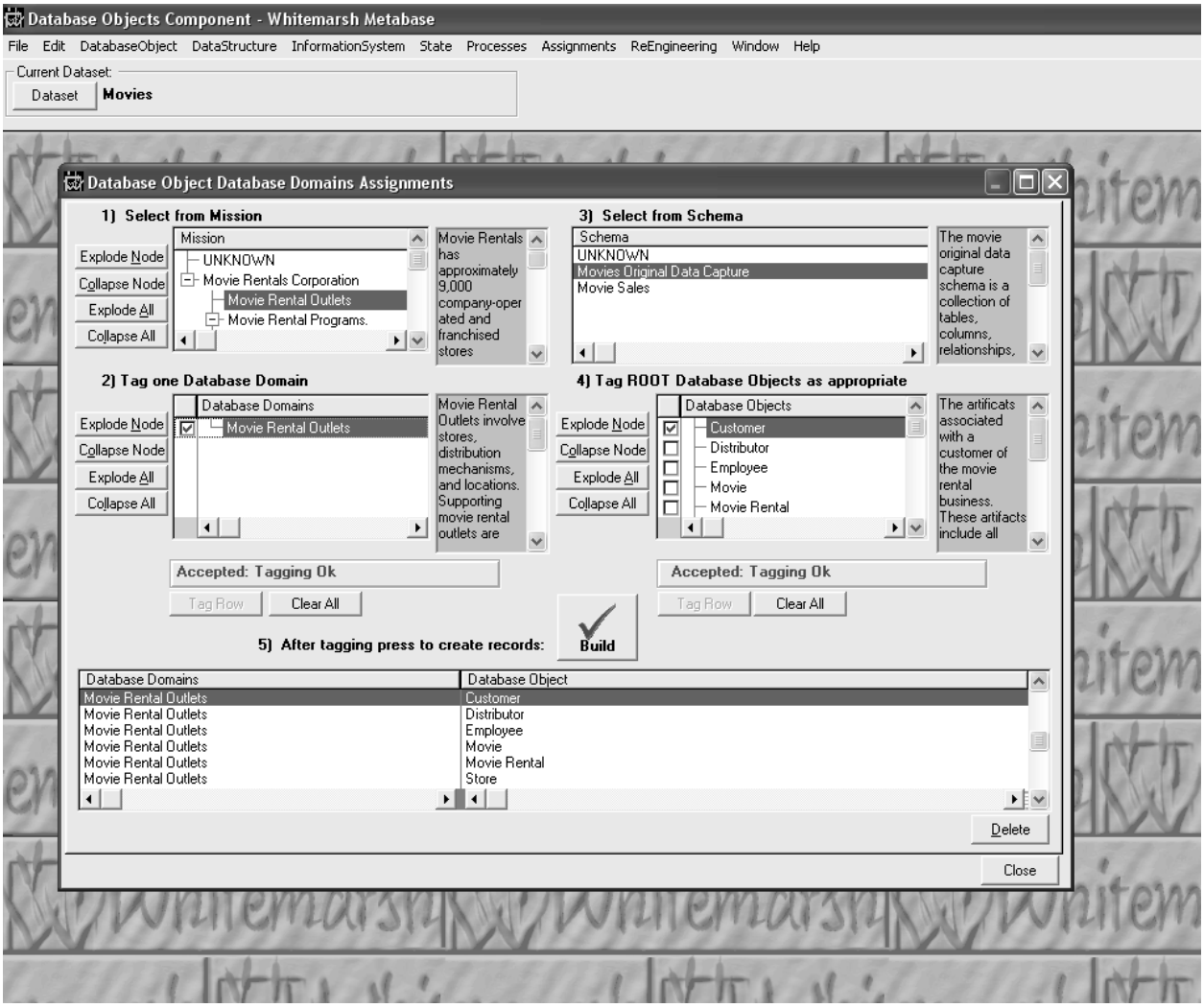

Figure 5. Database Object Database Domain assignment.

6.2.1.3 Reassign of Database Object to Database Domain

Once a database object has been assigned to a database domain, it can be reassigned. The process that accomplishes that is presented in Figure 6. A schema is selected. Then a database object. The current database domains for the highlighted object is shown. Tag it. Then on the right brows, tag the database domain to which the database object is to be moved. When the Press button is pressed the database object is reassigned from the "From" database domain to the "To" database domain. The new database domain assignments for the database object are then shown in the window.

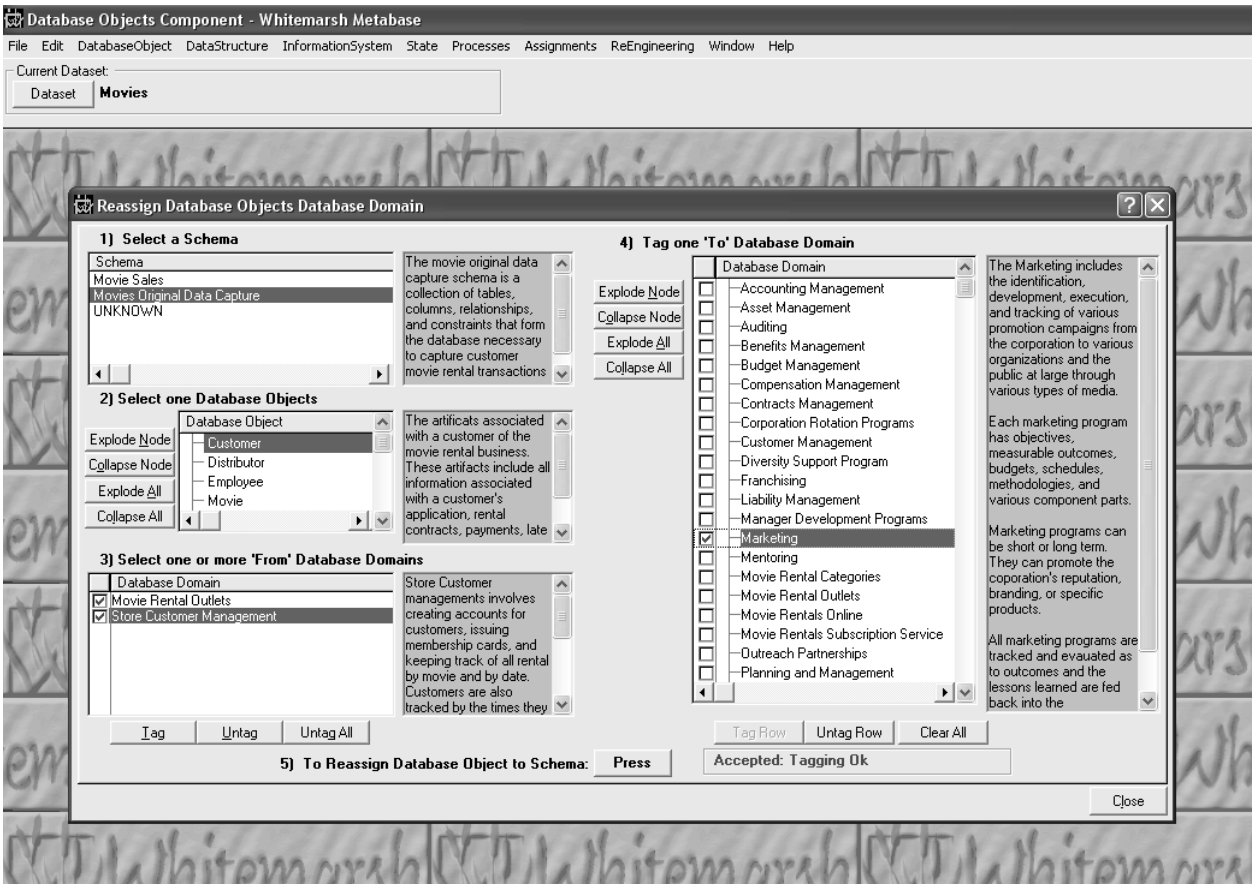

Figure 6. Database Object Database Domain reassignment.

6.2.1.5 Reassign of Database Object to Schema

Once a database object has been assigned to a schema, it can be reassigned. The process that accomplishes that is presented in Figure 7. One or more database objects are tagged in the left browse. Then one schema is tagged in the right browse. When the Press button is pressed the database objects are reassigned from one schema to another. The new assignments are then shown in the window. If there are any tables assigned to the schema then the schema re-assignment cannot take place as that would in effect be reassigning the set of database object tables from one schema to another. Additionally, only root-level database objects can be re-assigned. Their contained subtyped database objects are re-assigned automatically. There are two message areas, one for the subtype tagging and the other for the schema tables message.

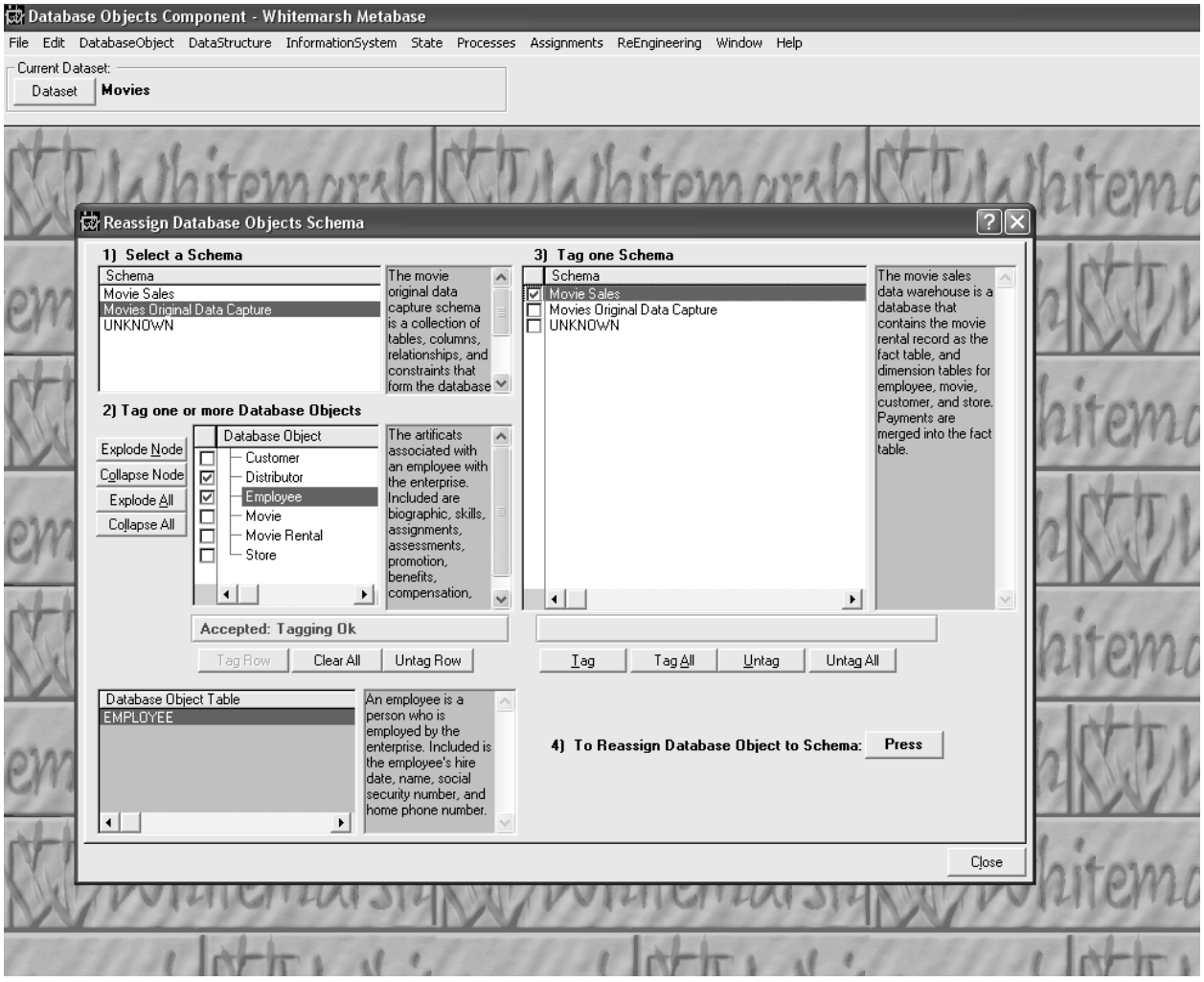

Figure 7. Database Object Schema reassignment.

6.2.2 Database Object Structures

Database object structures consist of the following:

- Database object tables
- Database object table rationale
- Database object table assignment

6.2.2.1 Database Object Tables

The structure of a database object is the set of tables that belong to the database object. Figure 8 shows a list of database object tables. The top browse shows the schema, the contained database objects, and then within a highlighted one, the list of database object tables. Each table then shows the columns. Each included table has the table name and the membership rationale. There are no insert, change, delete buttons on this screen. That is accomplished by the Assign Database Objects to Tables, and the Reassign Database Object Tables processes.

6.2.2.2 Database Object Table Rationale

The reason for the membership must set into the relationship between a table and the database object.

There are two types of membership rationale, natural and multiple inheritance. A natural membership is strictly hierarchical.

Natural membership implies that there is single inheritance and that the contained table is subordinate to the parent table. For example, both the invoice header table and the invoice detail table are natural members of the Invoice Database Object.

Multiple inheritance of for a database object table means that the table belongs to two or more database objects. This occurs in two circumstances. Factoring, or Interrelationship. Factoring implies that a proper column of data, say, Gender, or an entire data structure, say Address has been factored out of a main table. E.g. Person's gender, or a common address for multiple people. In either case the factored out data has its own database object and there is a relationship between the two database objects.

Figure 9 presents a browse list for membership rationale. This reference data is accessed under the data structure menu item. The name of the basis for membership rationale and a description of the basis can be entered. Figure 10 presents the update screen for membership rationale.

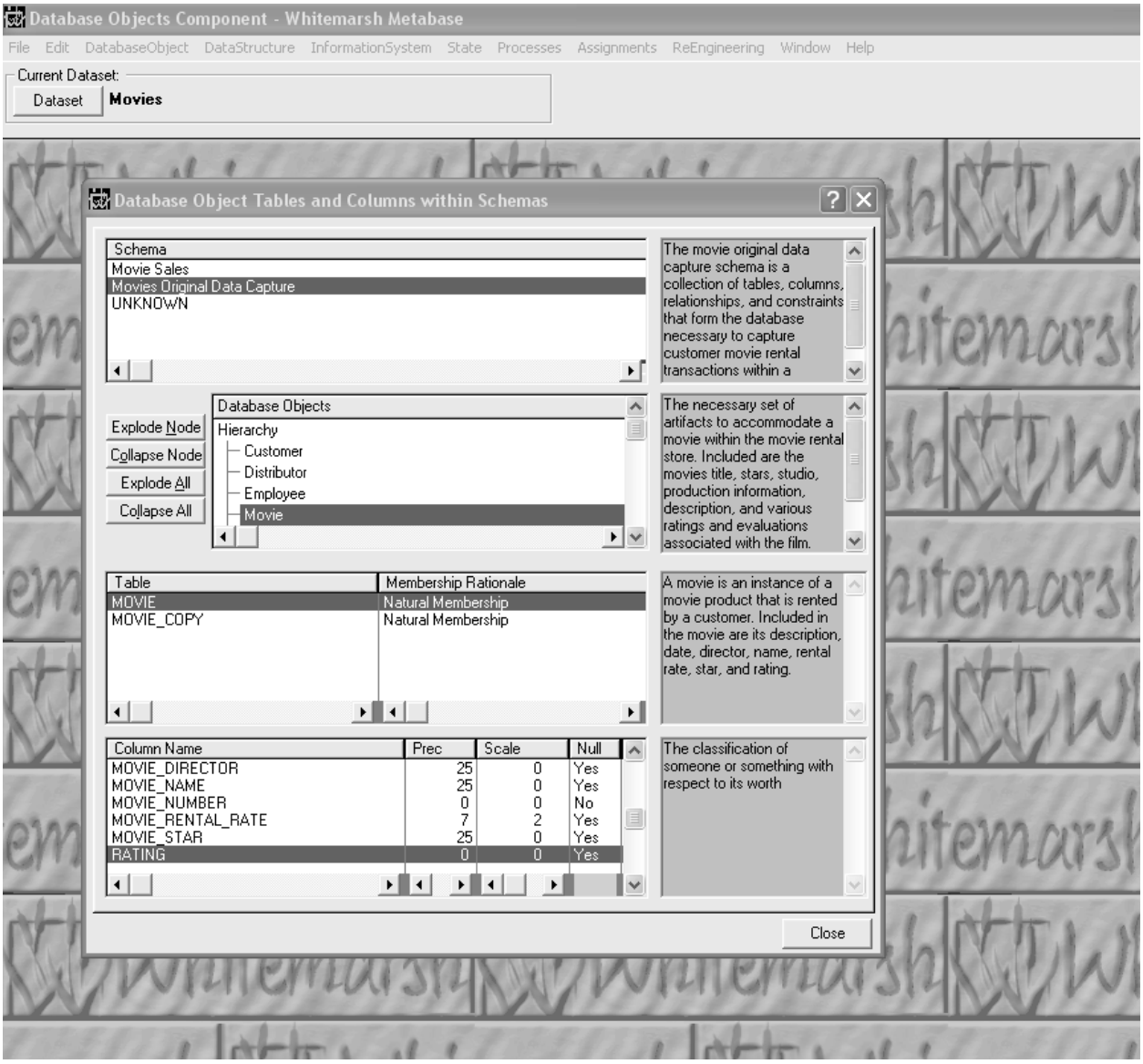

Figure 8. List of tables within database objects.

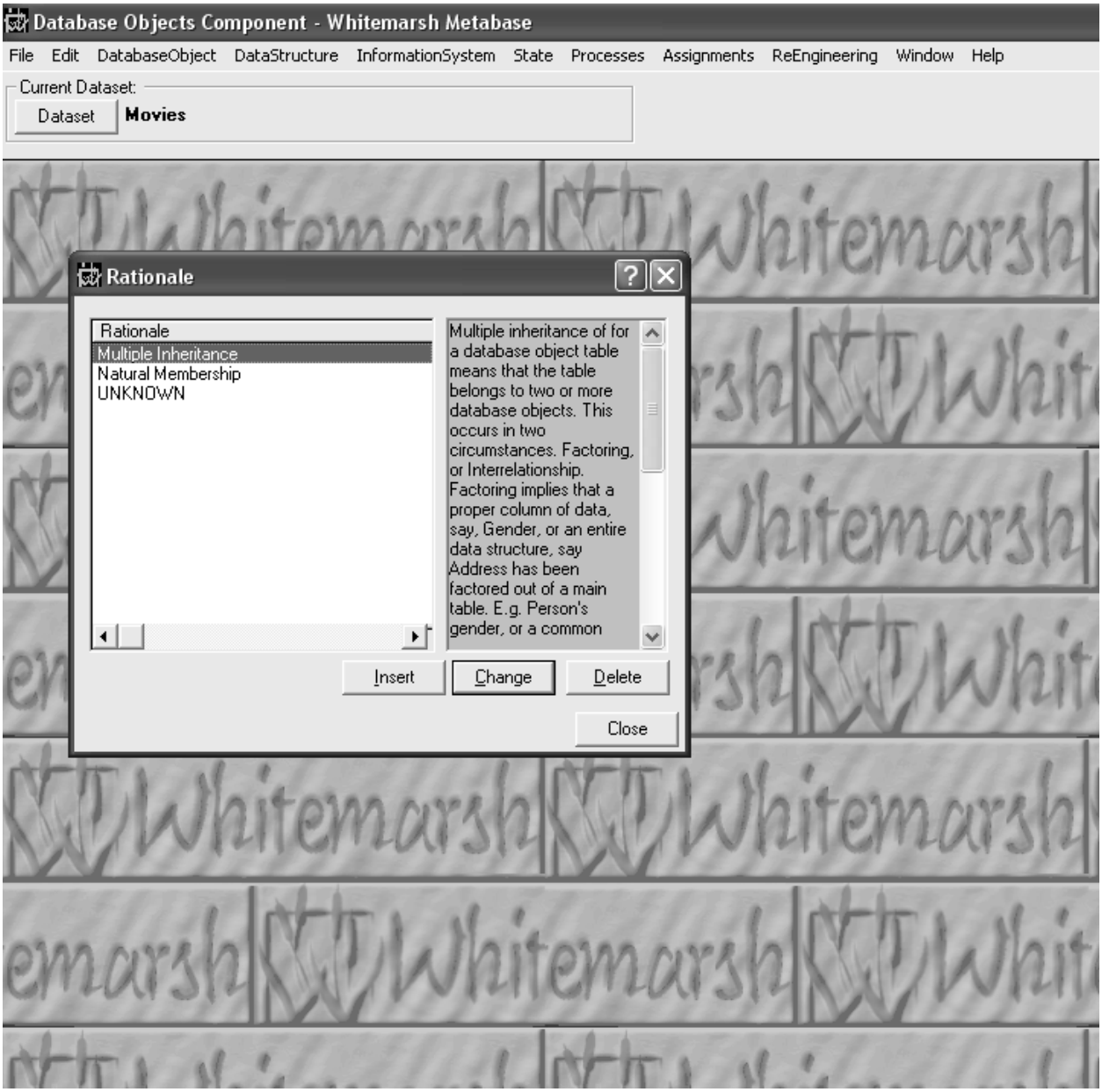

Figure 9. Database Object Table membership rationales.

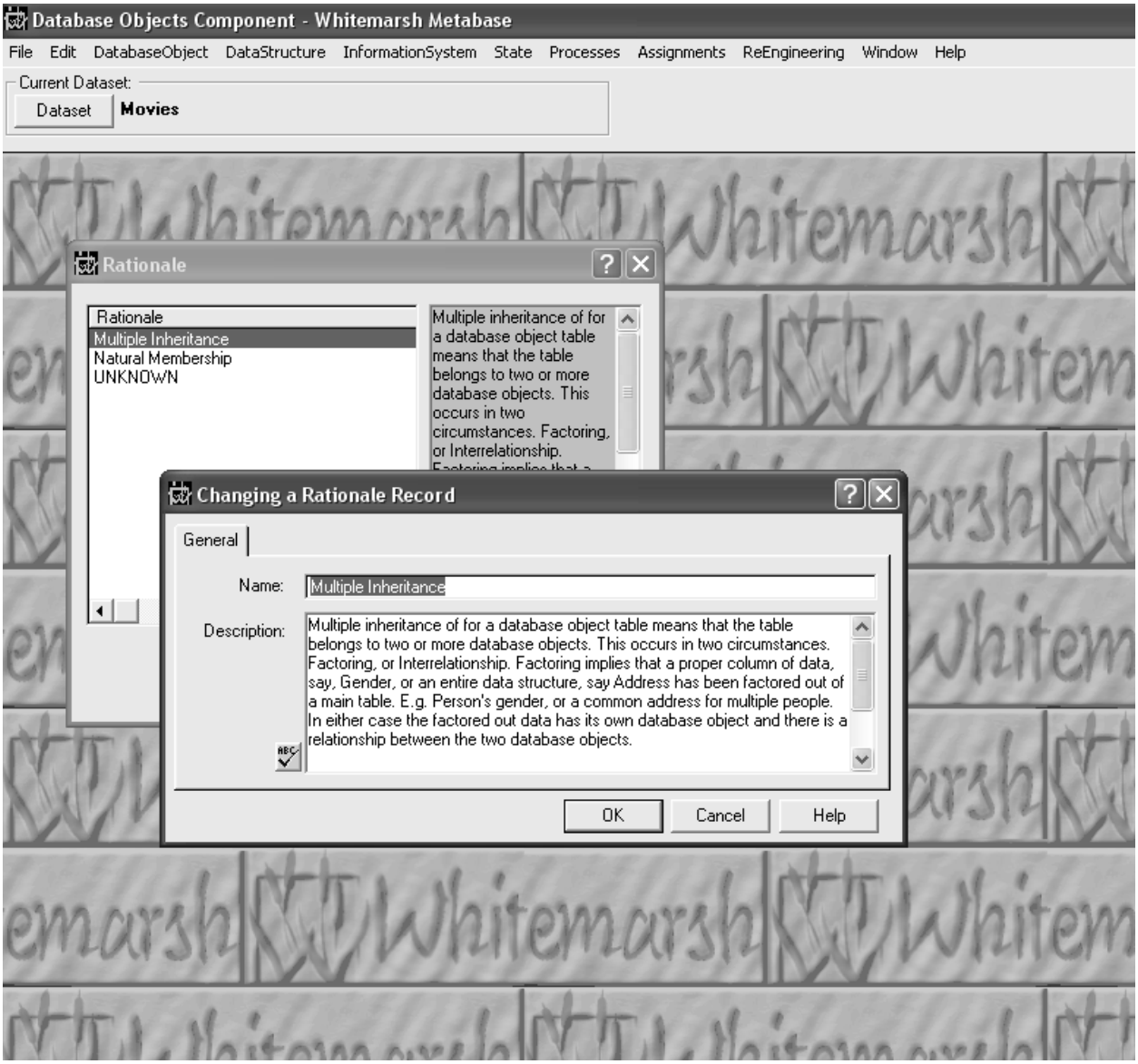

Figure 10. Updating a Database Object Table membership rationale.

6.2.2.3 Database Object Table Assignment

Once the rationale is created, tables can be assigned to database objects. This is accomplished in the Database Object Table Assignment. The database object table assignments' window is shown in Figure 11. In this window, tables are assigned to database objects. In this window, the particular database object is identified and tagged in the upper left window. One only database object should be tagged. If more than one is tagged, they will be ignored. Then, tag as many different tables as appropriate. Finally, press the Build button. The tables that are then assigned to the database object will appear in the bottom window. If any records already exist in the bottom window, they have already been assigned to the highlighted database object. A default rationale is automatically assigned.

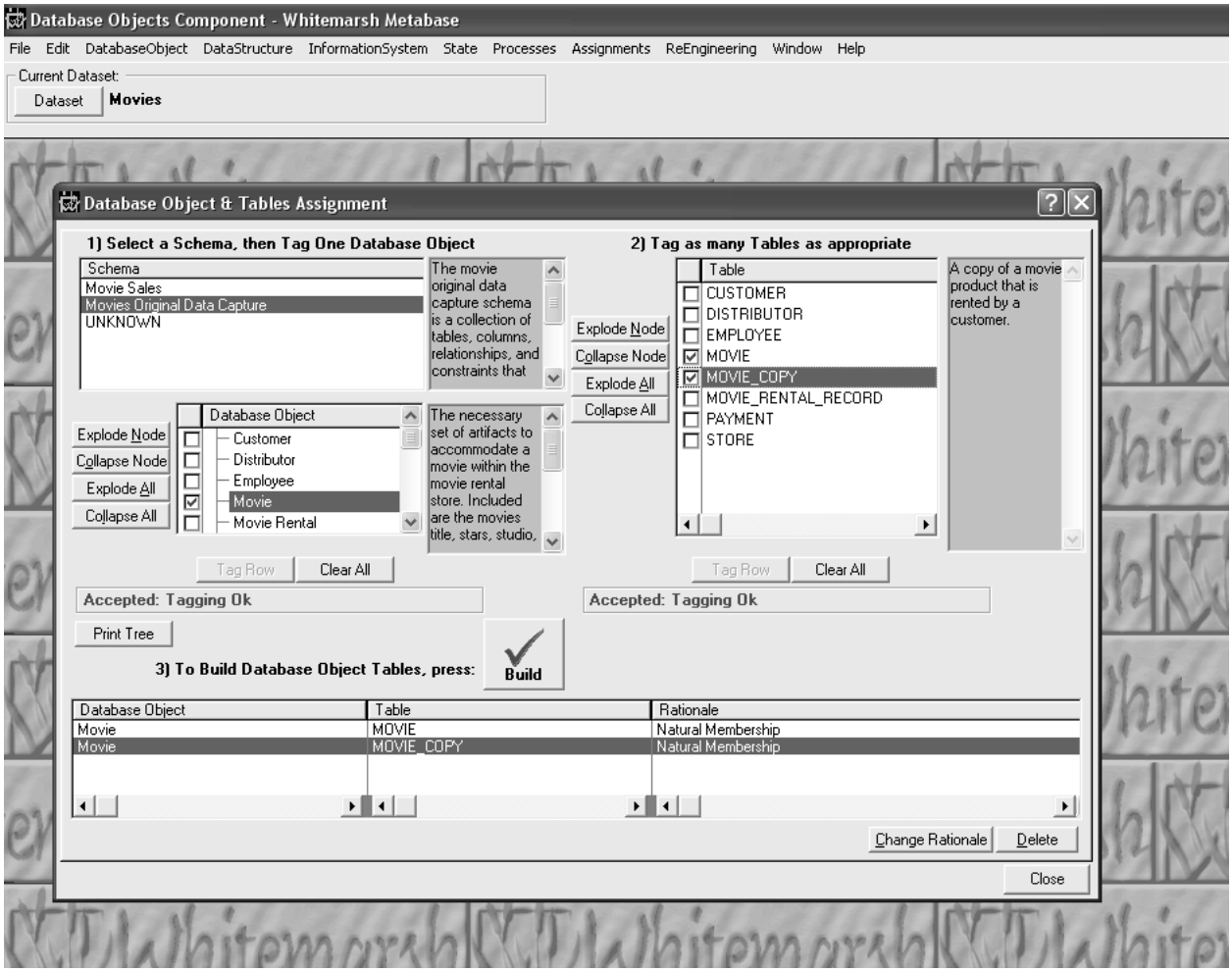

Figure 11. Database Object Table assignment.

6.2.2.4 Database Object Table Rationale Assignment

The database object table rational can be changed if it is inappropriate. To change that rationale, press the Change Rationale button. Figure 12 then results. Choose the appropriate rationale. Entering a zero and then tabbing through causes a list to appear. Select from the list.

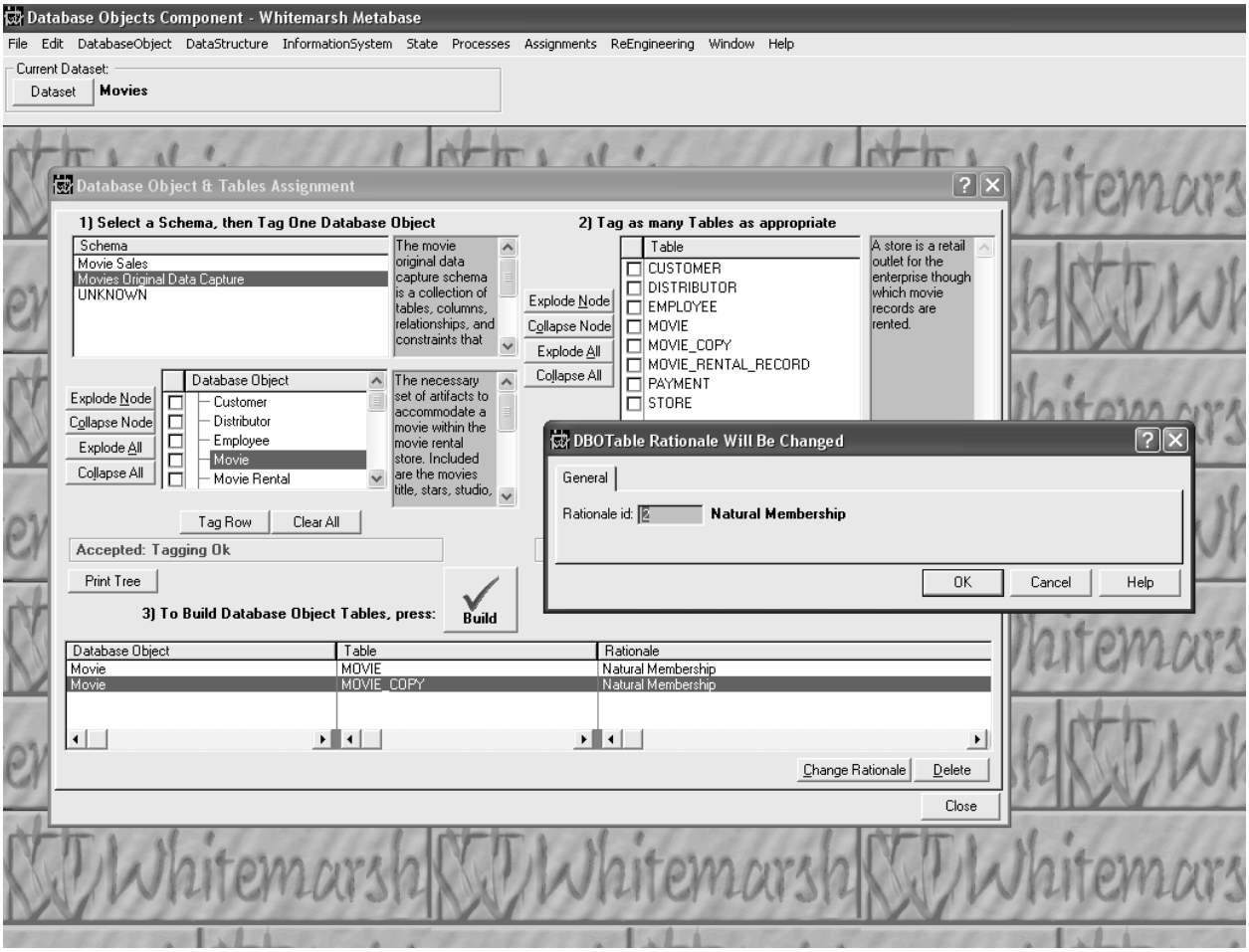

Figure 12. Changing a database object table membership rationale.

6.2.3 Database Object Table Process

Database object table process consists of the following:

- Database object table process
- Database object table process column assignment

6.2.3.1 Database Object Table Process

Database object table processes are owned by only one table. They, as would be required for referential integrity purposes, may involve additional tables within the database object or an outward reference to another table. This second case occurs when a database object employs foreign keys to another database object table for two reasons:

- Data factoring, such as the lookup of the full name of a State when only the State_id as a foreign key is represented within the proper database object table.

- Data joining, such as when two database objects are formally joined as would be the case when an employee database object (in some discrete state) is joined with a work project database object on an employee work task. One of these database objects has to be identified as owning the intersection table, however.

Figure 13 presents the browse window for database object table processes. The browse window in Figure 13 is a four part browse. That is, there are three nested browses. These three are interconnected in a hierarchy of:

- Database schemas and their contained database objects
- Database objects and their contained
- Tables and their currently assigned
- Database object tables processes

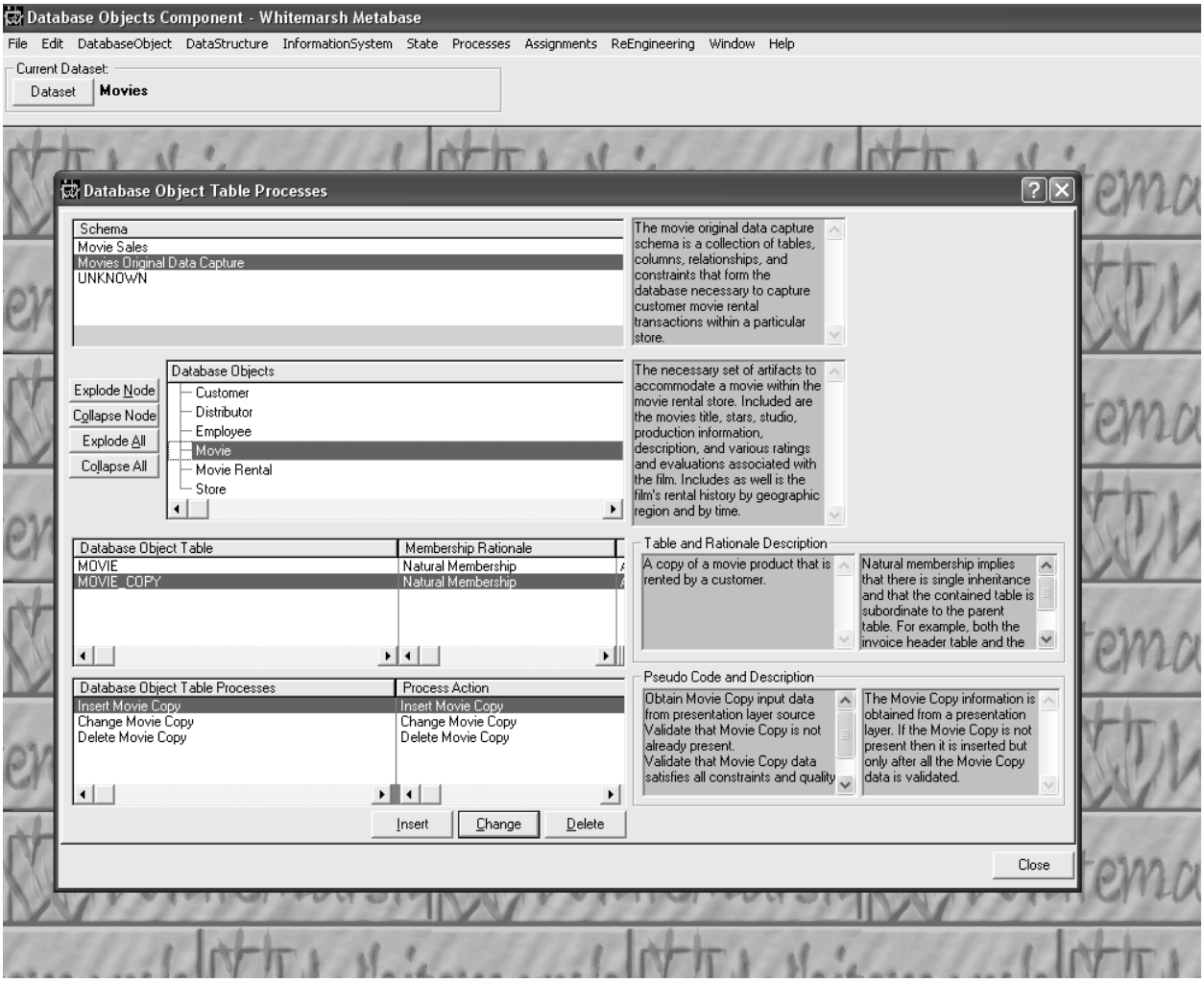

Figure 13. Database Object Table Processes.

When a database object is highlighted, only the database object tables associated with that database object appear in the tables' window. And, when a table is highlighted, only the database object table processes associated with that database object table appear. If no database object table processes appear then none are assigned. To create a database object table process, press the Insert key. The update window then appears, as shown in Figure 14. This table contains the following fields:

- Name–a short descriptive name for the database object table process. This should be expressed as an action phrase
- Action, the action-purpose of the database object table process
- Pseudo Code, the actual computer-based actions against the database object table when the database object table process executes. Note: at this time, the pseudo code language is a text string. It is recommended that the SQL constraint and persistent stored module language should be employed.
- Description, a set of descriptive text associated with the database object table process.

Even though specified through this screen, the database object table process is not yet complete. Two additional activities must occur:

- The columns from the database object table involved in the database object table process must be identified
- The specific position within the sequence of database object table process executions within a database object information system must be specified.

These are specified in the assignment processes. Since a database object table process may be invoked by many different database object information systems, its execution sequence position is specified apart from the database object table process.

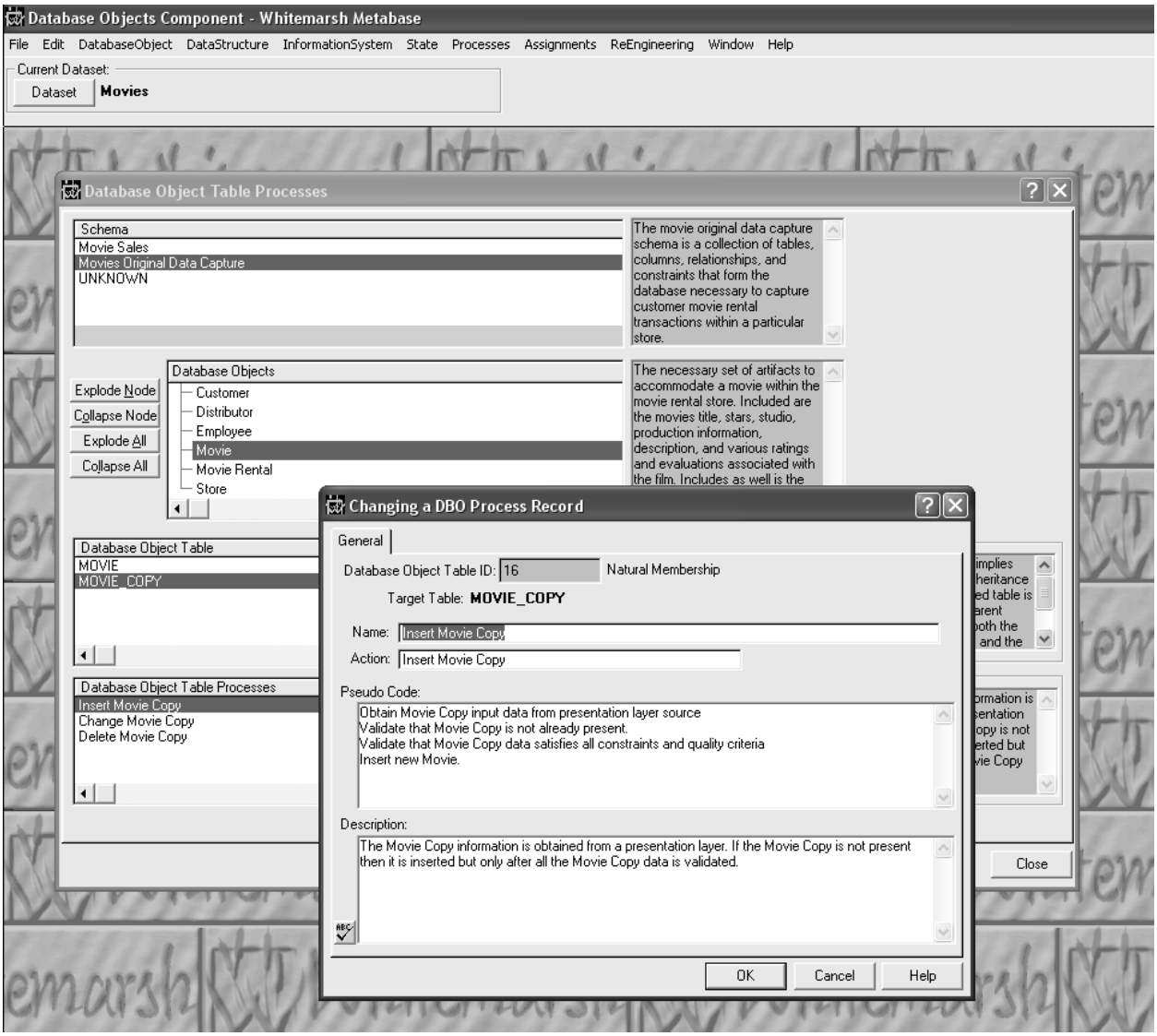

Figure 14. Database Object Table Process update screen.

6.2.3.2 Database Object Table Process Column Assignment

The database object table process column assignments' window is shown in Figure 15. In this window, database object table processes are assigned to database object tables. The database object table process column assignments are created by identifying the appropriate database object table process and then tagging the appropriate database object columns. The database object process is selected by highlighting the database object in the upper left window. At that point, the assigned database object tables appear in the middle left window. Highlight the appropriate one and then the assigned database object table processes appear. Tag only one database object table process.

When the database object table is highlighted in the middle left window, the columns assigned to that table appear in the upper right window. Tag as many columns as are to be acted upon by the database object table process. Then press the Build button. The columns that are then assigned to the specific database object table process will appear in the lower right window.

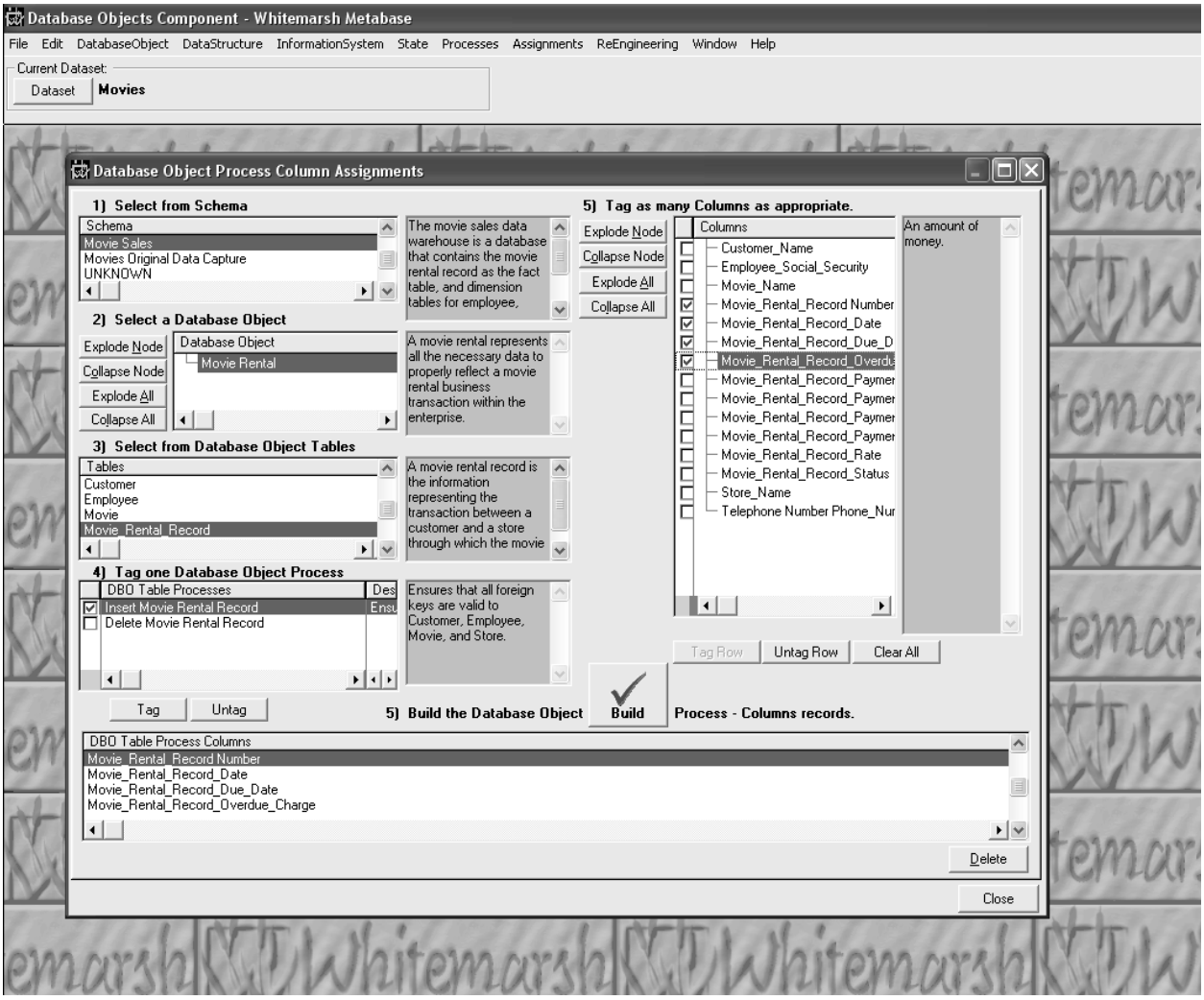

Figure 15. Database Object Process Column assignments.

6.2.4 Database Object Information System

Database object information systems consist of the following:

- Database object information systems
- Database object information systems database object table process assignments

6.2.4.1 Database Object Information Systems

The database object information system is the component of a database object that transforms the database object from one state to the next. There may be multiple database object information systems associated with one state transformation. In addition, a database object information system may be involved in more than one state transformation.

Database object information systems are different from business information systems in that database object information systems are ideally specified entirely within the SQL language. Thus, database object information systems have only these purposes:

- Receive data from a business information system
- Cause the execution of a specific sequence of database object table processes that transform database object tables
- Receive data from database object table processes and make that data available to a business information system

Database object information systems thus do not contain a presentation layer as would be the case with other types of information systems that are programmed in other ANSI standard languages.

Figure 16 presents a browse screen for a database object information system. The information for the database object information system is simple, name and description. Each is listed within the context of their database objects and database object schemas. As with database object table process, the position within the execution sequence of a set of database object information systems is set when the database object information system is assigned to be a part of the accomplishment of a database object state. Figure 17 is an update screen for a database object information system.

In all essence, the database object information system is the sequenced collection of database object table processes with one additional feature. If all the database object table processes compete then so too does the database object information system. Otherwise, it rolls back all the database object tables to their prior state. Either the object is wholly transformed or it is not transformed at all.

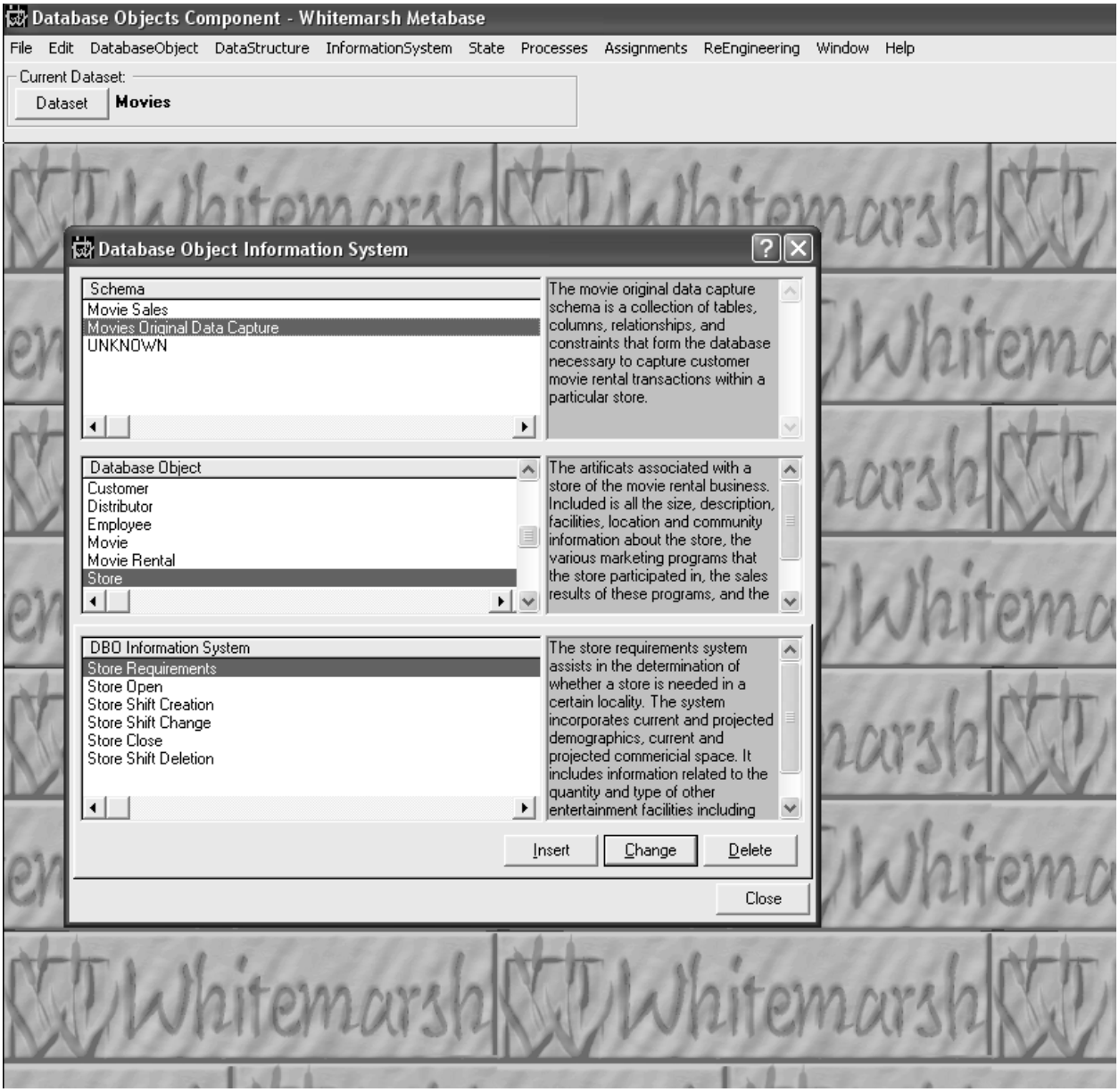

Figure 16. Database Object Information Systems.

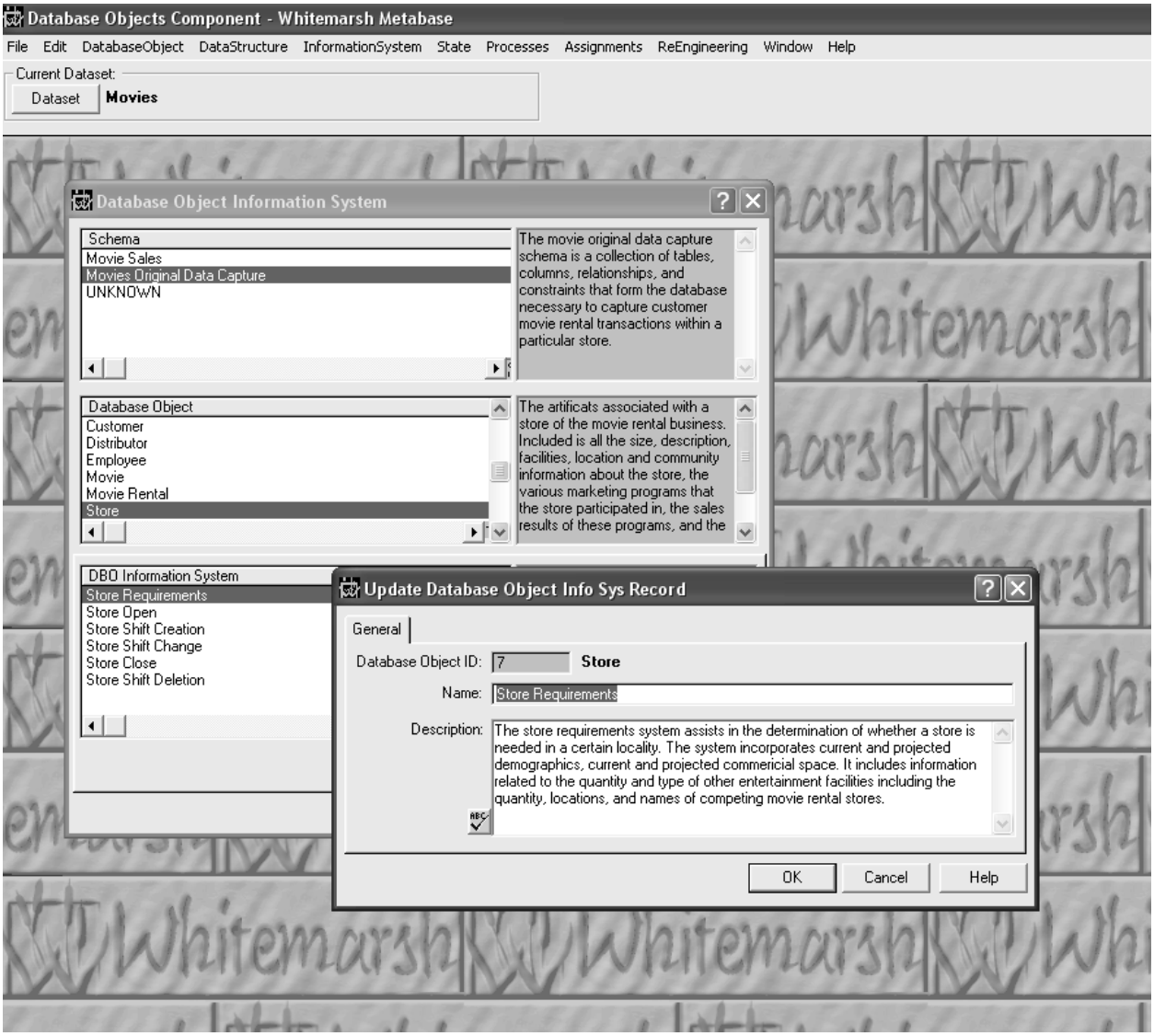

Figure 17. Database Object Information System update screen.

6.2.4.2 Database Object Information Systems DBO Table Process Assignment

The database object information system DBO table process assignments' window is shown in Figure 18. In this window, database object table processes are assigned to specific database object information systems. The purpose of this assignment is to specifically identify those database object table processes that are accomplished when the database object information system is executed.

This assignment is accomplished by identifying the appropriate database object assigned information system and then tagging the appropriate database object table processes. The database object assigned information system is identified by first highlighting the appropriate database object in the upper left window. Once highlighted, the assigned database object assigned information systems appear in the middle left window. Tag only one.

The highlighted database object causes the assigned database object tables to appear in the upper right window. As the appropriate one is highlighted, the database object assigned table processes appear in the middle right window. Tag as many as appropriate. Then press the Build button. The assigned database object table process will then appear in the bottom window. A default sequence is established. The default sequence is modified by pressing the Update Sequence button. As the sequence is changed the bottom window's browse list is adjusted.

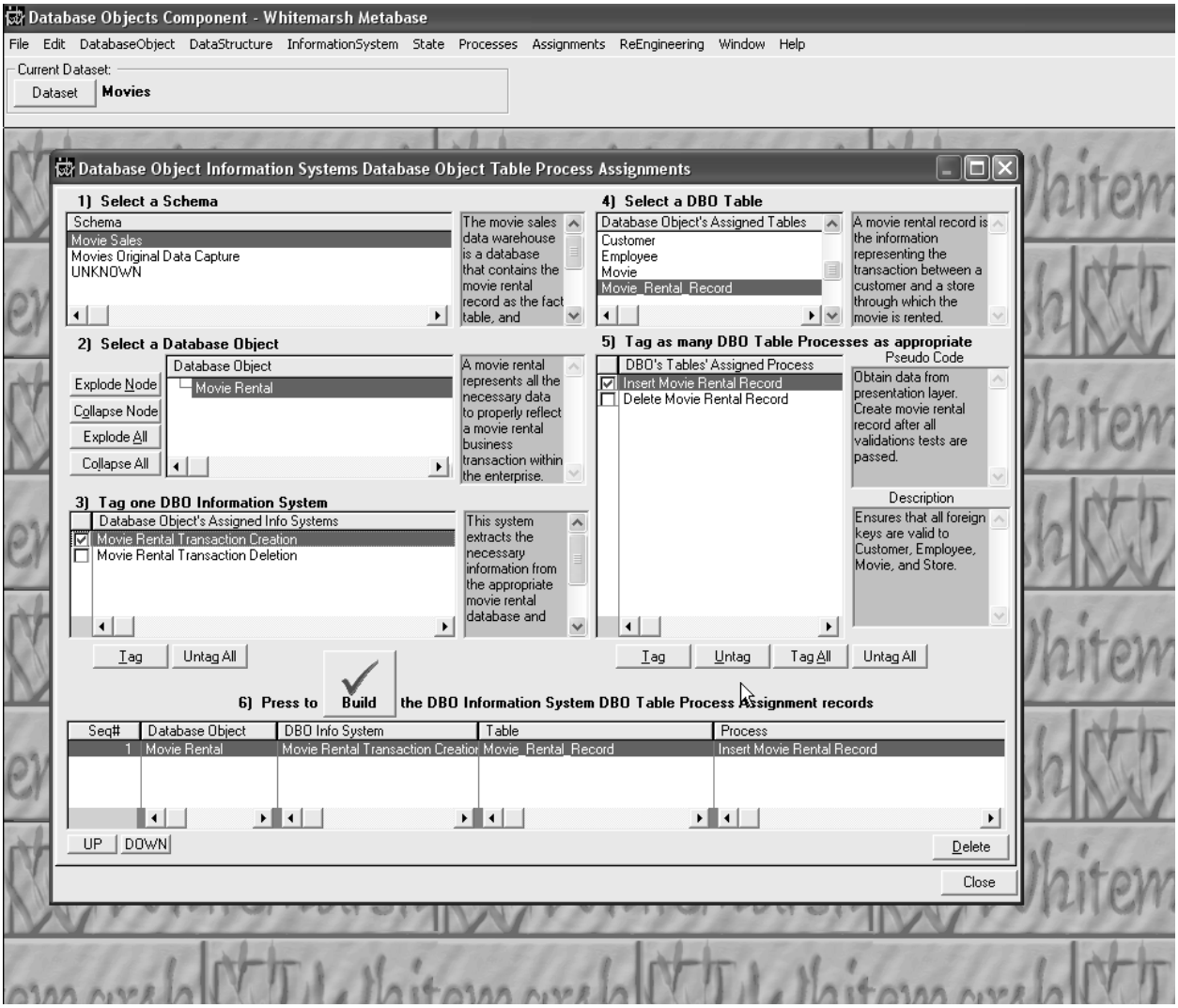

Figure 18. Database Object Information System Database Object Table Process assignment.

6.2.5 Database Object State

Database object state consist of the following:

- Database object information systems
- Database object information systems database object table process assignments

6.2.5.1 Database Object State

The database object state represents the successful accomplishment of the execution of the set of database object information systems. Each database object information system, in turn causes the accomplishment of the associated database object table processes in their defined sequence. If any of the database object information systems fail, and one would if any of its contained database object table processes fail, then the accomplishment the database object state transformation also fails. In this case, the underlying DBMS causes a complete roll-back to the prior database object state.

Database integrity is thus preserved, from one business defined state to the next. The initial state of a database object is the null state. The database object then proceeds through a set of non-null states, where each is defined to reflect a business recognized value state of the database object. For example, the value states for an employee may be:

- Employee Requisition (01)
- Employee Candidate (02)
- Employee New-hire (03)
- Employee Assigned (04)
- Employee Promotable (05)
- Employee in Probation (06)
- Employee Separated (07)

Each of these states must be identified and defined. Further, the exact data that represents a completed state must be clearly specified along with all the database object information systems and in turn the database object table processes necessary to achieve the state. Adjacent to each employee state is the state's sequence number.

The data that is needed for the database object state is depicted in the browse and update screen in Figure 20 and consists of the database object state name, its description and then its sequence within the context of the database object.

To change the sequence, the second state menu item presents only a browse screen. This screen, presented in Figure 21 enables a state's sequence to be changed when the Update Sequence Number button is pressed. Each sequence number can be changed directly. As each number is changed, the list automatically resorts. Duplicate sequence numbers are permitted so as to avoid the problem of having to do "creative" renumbering to avoid duplicates (i.e., switching two states). Despite being allowed, a duplicate sequence number is a mistake.

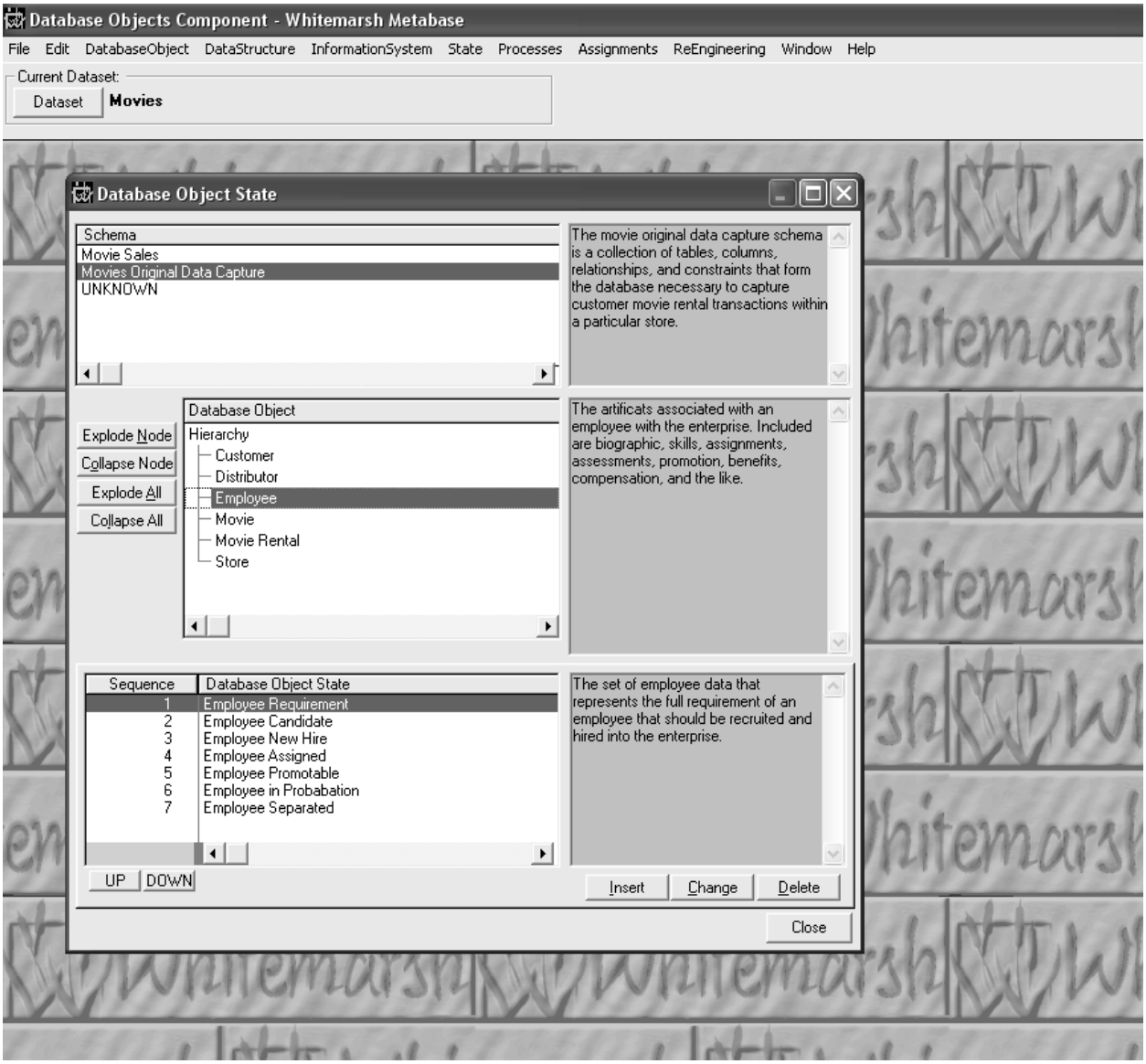

Figure 19. Database Object States.

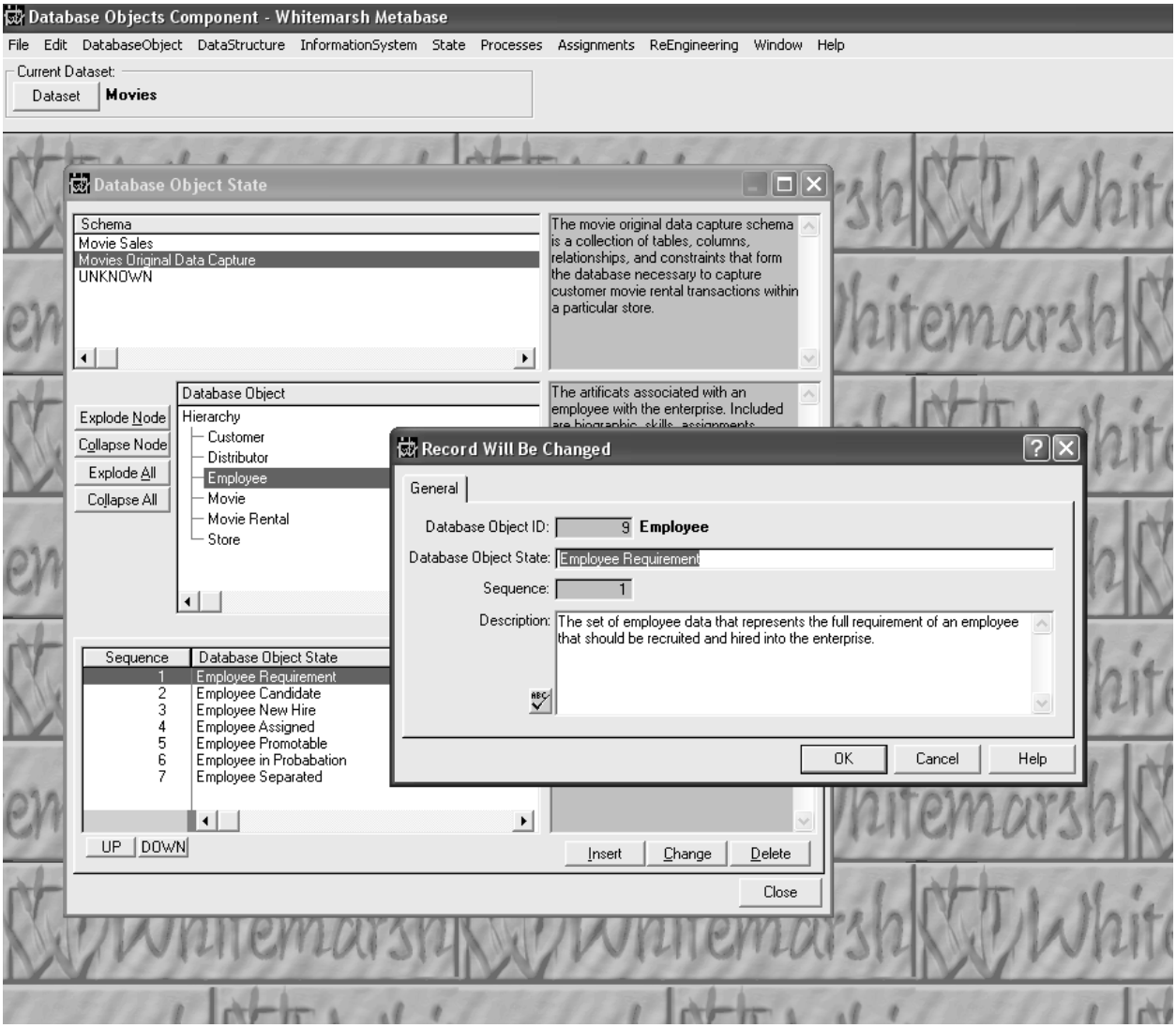

Figure 20. Database Object State update screen.

6.2.5.2 Database Object State DBO Information System Assignments

The database object state DBO information system assignments' window is shown in Figure 22. In this window, database object information systems are assigned to specific database object states. The purpose of this assignment is to specifically identify those database object information systems that must be accomplished to achieve the requirements of the database object state.

This assignment is accomplished by identifying the appropriate database object assigned state and then tagging the appropriate database object information systems. The database object state is identified by first highlighting the appropriate database object in the upper left window. Once highlighted, the assigned database object assigned states appear in the middle left window. Tag only one.

The highlighted database object causes the assigned database object information systems to appear in the upper right window. Tag as many as appropriate. Then press the Build button. The assigned database object state information systems will then appear in the bottom window. A default sequence is established. The default sequence is modified by pressing the Update Sequence button. As the sequence is changed the bottom window's browse list is adjusted.

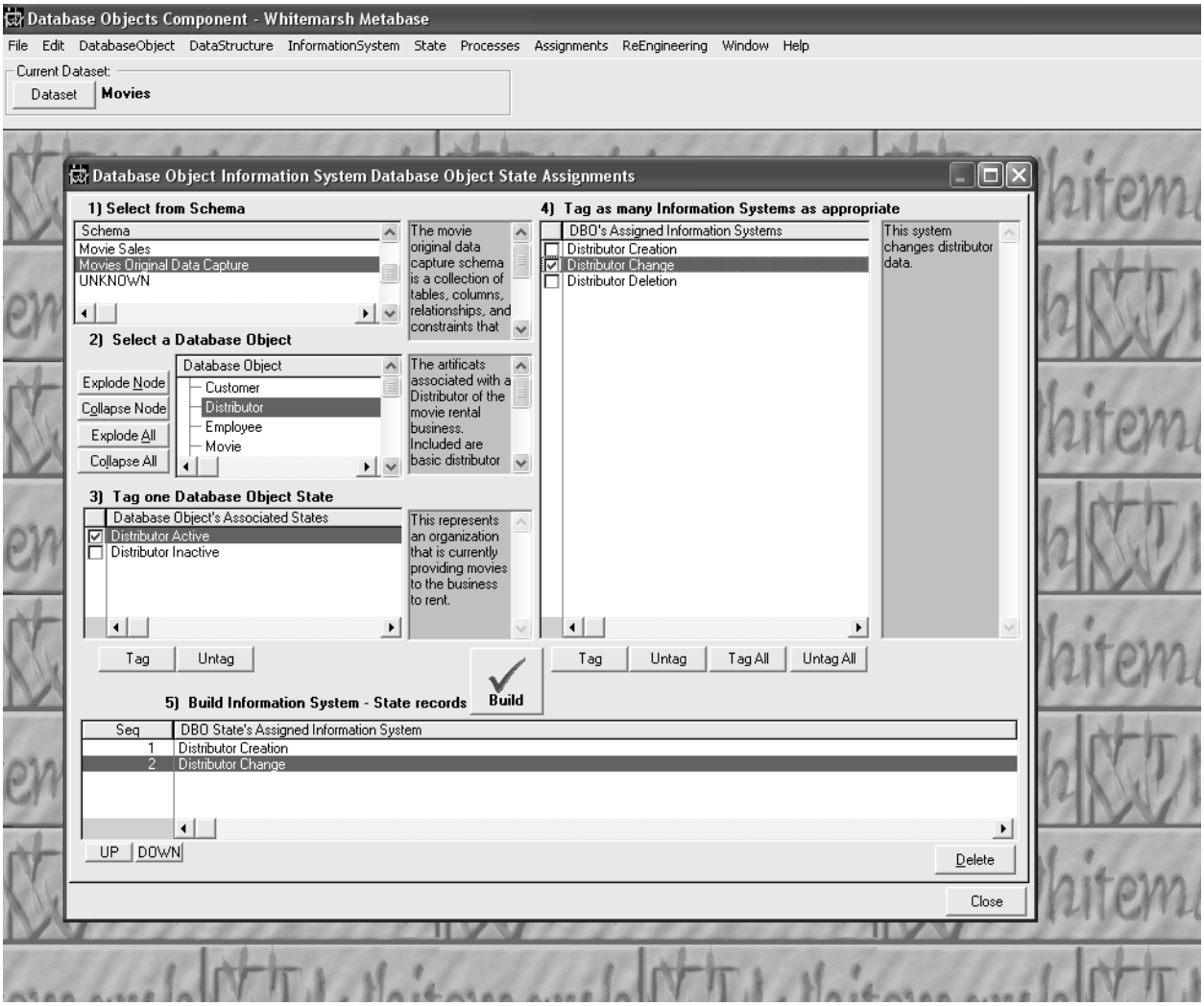

Figure 21. Database Object State Information System assignment.

6.4 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.