IBM Software
White Paper

Data lifecycle management strategies

The importance of the complete business object

Sunil Dravida

WW InfoSphere Optim CTP Leader,
IBM Corporation

Tom Forlenza

WW InfoSphere Optim Center of Excellence,
IBM Corporation

Contents

Introduction

Consider the following scenario: one of your organization's business partners has run into some legal trouble. During an SEC raid, the business partner implicated your organization in the crime. As a result, your organization is now required to produce financial information about your operations from six years ago. However, the company archives financial data after two years and migrated to a different financial application about three years ago. Even if the six-year-old records were effectively archived and secured, how will they be accessed? Can the data be represented in a meaningful way? What will happen to your organization if the data cannot be accurately accessed?

This example is just one of many scenarios that demonstrate the importance of effective data retention, data archiving and e-discovery practices—not to mention practical security and business applications. It is critical that an organization understands relationships between data, its business value and ways to ensure long-term and accurate data access. One key concept that helps organizations better understand their data is the complete business object.

An organization's information can be grouped into many small units, often referred to as objects. Every process—defined or otherwise—within the organization creates information as the process is executed. This information constitutes the complete business object.

This white paper defines the complete business object as it applies to organizations and its importance to an organization's data governance strategies. This paper also examines how the complete business object is implemented in several IBM® data lifecycle management solutions.

Overview of the complete business object

Business objects represent the fundamental building blocks of your application data records, including the full and extended information related to the data entities. From a business perspective, a business object could be a payment, invoice, paycheck or customer record. From a database perspective, a business object represents a group of related rows from related tables across one or more applications, together with its related metadata—information about the structure of the database and about the data itself. Capturing the complete business object offers a complete view of the business activity surrounding a particular transaction.

Understanding the complete business object

The complete business object is defined around a key construct: the relational database. A relational database is a structured set of data held in a computer. The sets are a combination of one or more tables that are represented as row-and-column matrices of information.

Typically, a relational database is normalized. Normalization is a systematic way of ensuring that a database structure is suitable for general-purpose querying and is free of certain undesirable characteristics—insertion, update and deletion anomalies—that could lead to a loss of data integrity and performance.

During normalization, sets of data are broken out into smaller sets and stored in many tables. For example, an invoice may contain pieces of information that are stored in various tables and sometimes even different databases and physical locations (see the sidebar, "Case in point: Archiving invoice data").

A complete business object is designed to account for the fact that the data of a defined object within an organization may be scattered across an organization's many divisions and physical locations. To describe the data and its relationships, a complete business object comprises lists of tables and the relationships between the tables, as well as definitions on how data is mapped and traversed within the business object. Complete business objects can include several types of tables:

- **Object tables** contain the central information of the business object. Typically, these tables experience a high volume of transactions and handle most of the system's activities. A complete business object includes one or more object tables.
- **Context tables** contain information relating to an object table but do not necessarily apply specifically to the object itself. They help put the data of the object table in context. A complete business object includes zero, one or more of these tables.
- **Reference tables** provide additional information that helps clarify the data in the primary object table. These are typically a table of codes, such as postal zip codes, a table of set values, such as a monetary exchange rate, or a table that defines a data element within the object or context table. A complete business object includes zero, one or more of these tables.

Case in point: Archiving invoice data

In this example use case, a sales representative takes an order from a customer by filling out a paper invoice, a physical form that includes various labels and corresponding blank spaces for the written information. The representative writes the customer's name, address and phone number and the appropriate part number, description, price and quantity for the products being purchased. Sometimes the representative must cross-reference another document to look up the part number and product description.

After it is filled out, the paper invoice is stored digitally. However, all the invoice data may not be entered in a single database table. In fact, good database design practices recommend storing the information in separate tables for efficiency.

In this case, the invoice is stored as a record in the database table INVOICE that contains only an invoice number and a customer number with a date and summary information, such as total sale (see Figure 1). Another table, CUSTOMER, contains the customer number, name, address and phone number. The customer address is only a street number, name and ZIP code. The city and state are stored in another table with the ZIP code for cross-reference. Information about the items purchased on the invoice is stored in the INVOICE DETAIL table, with each record containing the invoice number and a part number. The part number is a reference to the PART table, which contains the description and other information about the part.

If the object tables (the INVOICE table and the INVOICE DETAIL table) were the only tables archived, critical information about the order would be missing. For example, there would be no context information, such as customer name, street address and product description. In addition, reference data—such as the city and state of the customer and the location in the warehouse where the product is stored—would not be available. Providing a complete picture of the invoice requires information from the related context and reference tables. This combination of related object, context and reference tables helps define the complete business object (see Figure 2).

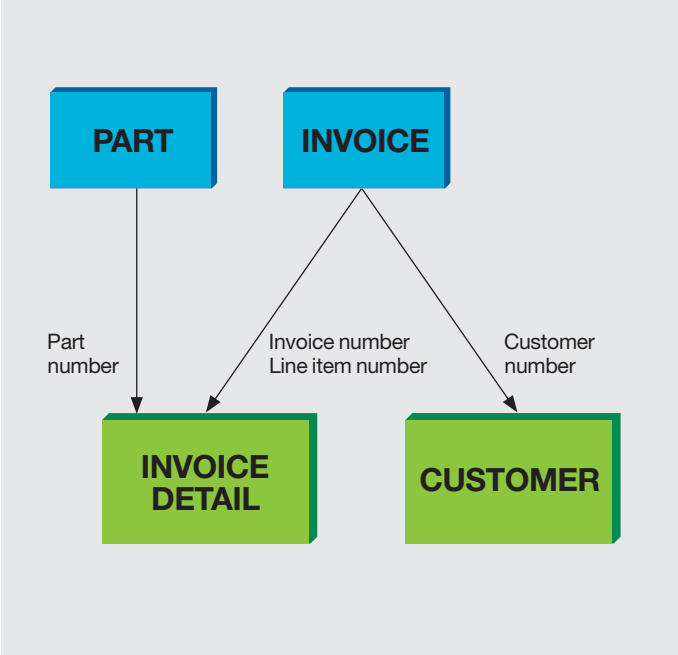

Figure 1: Database tables that make up the invoice business object.

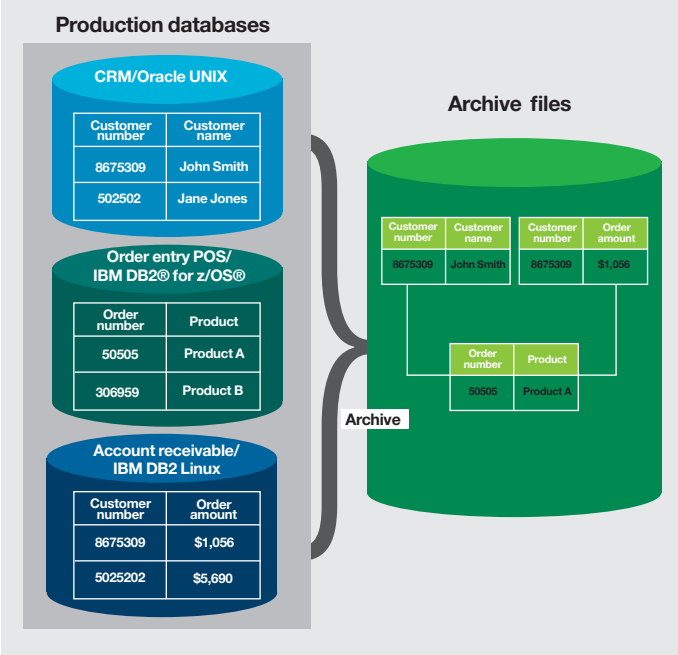

Figure 2: Data can be archived and accessed across multiple systems to make up the complete business object.

Generating the list of tables

The complete business object contains a list of object, context and reference tables. But how big does that list need to be for the business object to be complete? If the business object has too many tables, redundant data may be likely. On the other hand, a business object with too few tables may be missing important information.

A process that leverages extensive research performed by the IBM Engineering and Center of Excellence organizations can be used to quickly generate a business object's list of tables:

- **Frame the business object.** The IBM InfoSphere® Optim™ expert starts with an object table and traverses multiple levels into the data model using known database relationships to produce a superset of the table and the relationships of a business object. The depth of table selections can be customized to limit the number of parent-child-parent table traversals, and there are best-practice guidelines for when the traversal of a given path should end to include the optimal amount of data in the archive.
- **Refine the tables.** You must perform the appropriate validations to determine whether a table belongs in the object. For example, does a selected table currently contain any data? If not, this table probably is not used by either the application or the customer's particular use of the application.
- **Refine the relationships.** The refinement of relationships is a critical stage in the process, because incorrect relationships could result in a bad business object. Such an object may incorporate the wrong data, which in turn could make a data growth management process perform poorly or archive duplicate data. Or a subsetting process could extract all stored data of an object despite the use of selection criteria.

Examining relationships

The complete business object requires relationships to provide a structure for the data model so that the data in the object can be accessed. Relationships also help identify extraction paths for consistent, accurate archiving or subsetting processes.

Relationships can be direct and obvious, such as when the key value of one table is in the column of a foreign key field of another table. They can also be indirect, obscured or application-defined. For example, an application retrieves data from one or more tables, processes the information and using an algorithmic function, builds a key value for retrieving data from another table. Such relationships are difficult to find, and expert domain knowledge of the application is required to discover and decipher the relationships. Some enterprise resource planning (ERP) or customer relationship management (CRM) applications may use an indirect method for relating data, and many data governance software vendors do not consider indirect relationships in their data governance solutions. IBM employs domain experts in the engineering organization to develop and support indirect relationships of most third-party ERP and CRM solutions.

Relationships can also be data driven. Consider multiple tables that contain pay rate data for employees, as shown in Figure 3. Each table contains the pay rates for a specific geography indexed by pay grade or level. The EMPLOYEE table must relate to one of the specific geography tables for rate data, depending on the employee's residence and the employee's pay grade level. A complete business object must consider such relationships when traversing the data model.

Finally, relationships may need to be defined externally to an application. In some cases, a complete business object may encompass data across databases and applications. Workflows are a good example. A workflow is a process that drives the operation of a business, sometimes across different departments. When each department has its own database, workflow actions may be stored in different locations. Effective data lifecycle management solutions should have a federated capability to manage data across systems and applications.

The list of relationships contained in a complete business object can be quite simple. For example, if the business object consists of a single table, then the object probably has no relationships. However, this is an unlikely scenario; even a simple model containing three tables has multiple relationships. As shown in Figure 4, a minimum of two relationships is required: one to relate Table A to Table B and another to relate Table B to Table C. Sometimes two tables can have multiple relationships between them, as depicted in Figure 5.

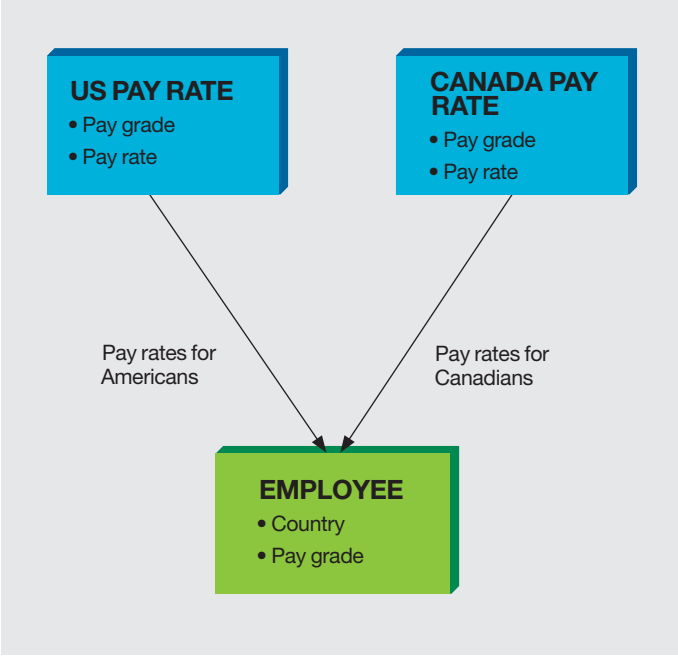

Figure 3: Tables for storing employee pay rate information.

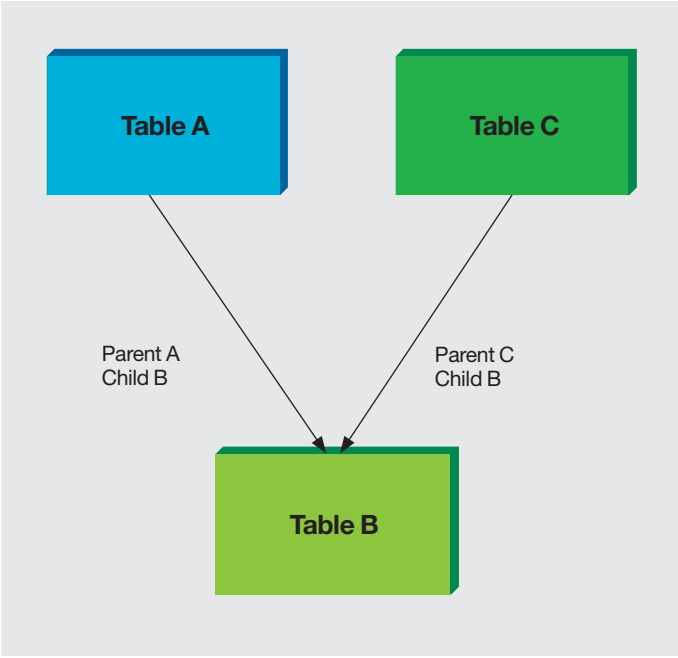

Figure 4: Two relationships in a three-table example.

Even though many relationships exist between tables, not all of the relationships are required for a given business object. Many normalized databases contain tables that are used as part of multiple objects. Take, for example, an activities table, which records information as it relates to the activity of a business object. An object that describes a service call may record each of the service call's activities in the activities table. Another object that describes a sales opportunity may record sales activities in the same activities tables. As a result, the activities table contains records for both the sales opportunity object and the service call object. If the business object to archive in this example is the service call object, then activities for sales opportunities do not need to be archived—the relationship between the activities and the sales opportunity object is not required. The complete business object would need to be aware that only the service call activities should be archived.

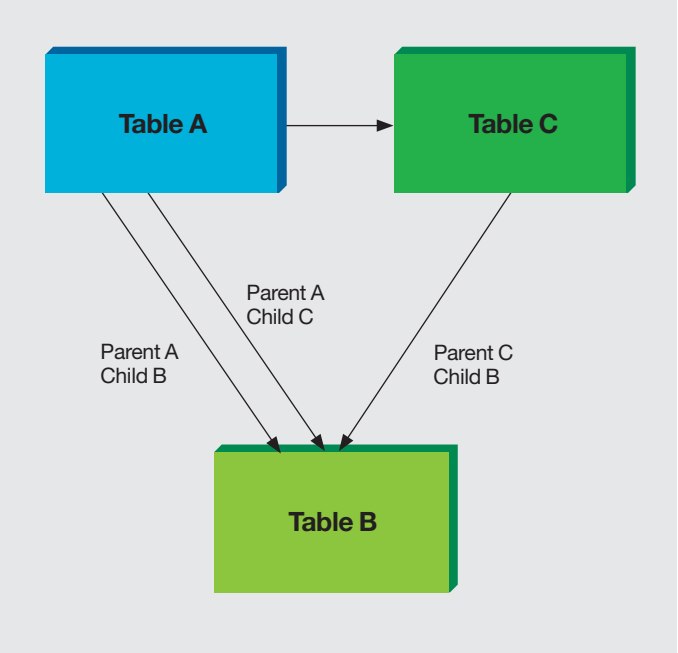

Figure 5: Multiple relationships in a three-table example.

In some situations, multiple relationships may relate a table to itself. For example, in a not-so-normalized database, a customer record may have a field for a billing contact and a field for a shipping contact. Both fields contain key values pointing to corresponding records in the contacts table. Thus, the contacts table is related to the customer table by two relationships. In this particular example, it would be prudent to preserve both relationships for the customer business object.

It is also possible that such multiple relationships exist but are not all required for the business object. For example, the aforementioned activities table could be related to the service call table multiple times: once as the activity of that object and also as a reference to a parent service call (see Figure 5). However, the parent service call is also related to the child service call. So the relationship to the parent from the activity can be considered redundant and not required as a part of the business object.

In this example, the relationship between the service call parent and child could potentially initiate a cycle during the extraction process that causes an excess number of records to be extracted. Although the self-referring relationship is important to the object, it must be used carefully when extracting data for the purpose of obtaining and storing the business objects.

Importance of the complete business object in data lifecycle management

Many applications are not designed with data governance in mind, so when an organization implements a data lifecycle management practice, it requires software and solutions specific to such a practice. Because an application typically does not have any knowledge of the data lifecycle management solutions, those solutions must understand the metadata of the application data.

When an ERP or CRM application creates an object as part of the daily transactions that occur in an organization, the application generates records that contain necessary information, such as any information the user (or actor if automated) enters at the time of creation and any reference values that point to data in other context and reference tables. Because the application generates these records transparently to the user, the complete business object must be considered when acting on this data outside of the application that created it.

For example, an organization that archives only the object tables to manage data growth may reduce storage costs, but in the long term, such a strategy is contrary to the goals of an effective data growth management and data retention practice. It is crucial that an archiving strategy start with the complete business object to help ensure long-term, accurate access to historical data.

IBM InfoSphere Optim solutions for data lifecycle management

The IBM InfoSphere Optim suite of data management solutions leverages the complete business object as one of its core foundations. These solutions are designed to manage data from requirements to retirement, helping organizations boost performance, empower collaboration and improve governance across applications, databases and platforms.

IBM InfoSphere Discovery

As previously discussed, tables for complete business objects can be found by traversing known data relationships. However, in some circumstances relationships are not defined by the database, and no known documentation exists to describe them—making it difficult to define a complete business object.

IBM InfoSphere Discovery delivers leading-edge capabilities to automate the identification and definition of data relationships across the complex, heterogeneous environments prevalent in IT today. Covering simple to complex data relationships, InfoSphere Discovery provides a 360-degree view of data assets. It analyzes the data values and patterns from one or more sources to capture and uncover hidden correlations. Once an end user identifies and validates these correlations, the software can then define a business object that is usable by other IBM lifecycle management solutions.

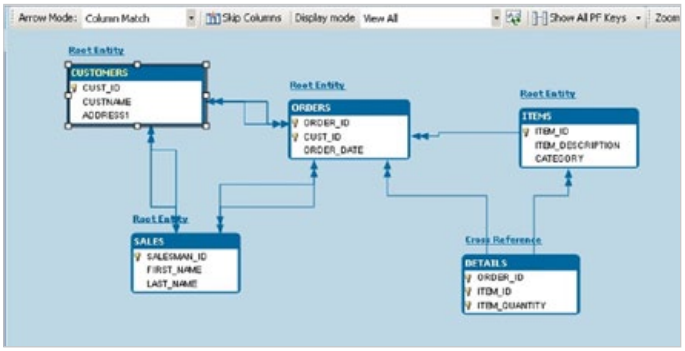

Figure 6: Entity relationship diagram as displayed by InfoSphere Discovery.

The InfoSphere Discovery interface allows end users to mine the data and review the identified relationships. For example, an entity relationship diagram of the found relationships for customer orders helps users understand the data model of the business object (see Figure 6). InfoSphere Discovery also presents a list of found tables and relationships in a data set (see Figure 7). Each relationship can be selected and reviewed for accuracy. Using InfoSphere Discovery, end users can view statistics showing how the relationships were identified (see Figure 8).

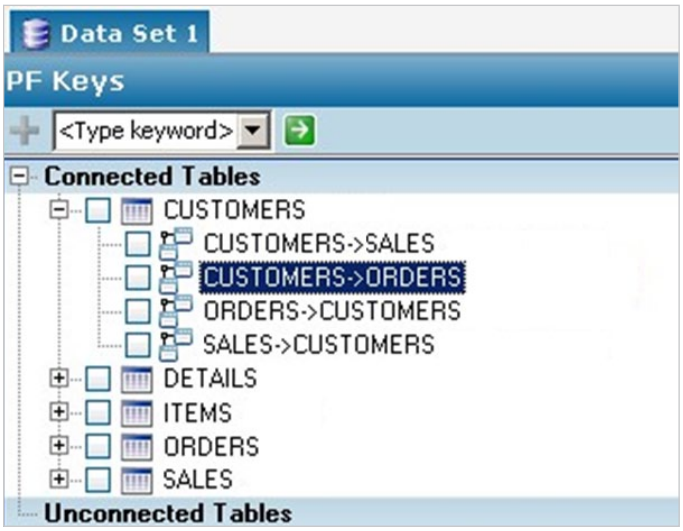

Figure 7: InfoSphere Discovery data set showing found tables and relationships.

IBM InfoSphere Optim Data Growth Solution

As a recognized best practice, archiving segregates inactive application data from current activity and safely moves it to secure storage. Managing data growth through effective archiving helps streamline application databases, reduce storage costs and improve application performance and availability. The InfoSphere Optim Data Growth Solution is designed to address data growth issues at the source by managing enterprise application data. InfoSphere Optim enables organizations to intelligently archive historical transaction records that may be infrequently accessed but still have business value. Historical data is archived securely and cost-effectively, and can be easily accessed for analysis, audits or e-discovery requests. And with less data to sift through, organizations can speed reporting and complete mission-critical business processes on time.

In InfoSphere Optim, archive processes are policy-driven; organizations can specify business rules for archiving. For example, an organization may choose to archive all closed orders that are more than two years old. InfoSphere Optim identifies the transactions that meet these criteria and moves them into an accessible archive file. The business rules, along with the complete business object, are captured in access definitions, central objects that are leveraged by the InfoSphere Optim Data Growth Solution. Access definitions facilitate the ability to navigate the business object data within InfoSphere Optim Data Growth, allowing end users to easily review and manage the complete business object and the archiving business rules.

CUSTOMERS->ORDERS	Row Hit Rate		Value Hit Rate		Selectivity		Notes
Expression	ORDERS	CUSTOMERS	ORDERS	CUSTOMERS	ORDERS	CUSTOMERS	
ORDERS.ORDER_ID= CUSTOMERS.CREDITCARD...	7% (632/9321)	95% (3334/3520)	7% (632/9321)	95% (632/667)	100% (9321/93...	19% (667/3520)	
ORDERS.ORDER_ID= CUSTOMERS.CREDITCARD...	1% (120/9321)	100% (3520/3520)	1% (120/9321)	100% (120/120)	100% (9321/93...	3% (120/3520)	

Figure 8: Hit statistics displayed by InfoSphere Discovery.

End users can view and modify the list of tables that make up the complete business object (see Figure 9). InfoSphere Optim provides several options for customizing the software to perform the archive process on any particular table related to the complete business object.

Likewise, InfoSphere Optim shows the relationships between the tables (see Figure 10). Relationships that are database-defined are automatically displayed. End users can also define relationships within InfoSphere Optim when the database lacks these constraints. Relationships can be selected for extraction, which allows InfoSphere Optim to use those relationships when archiving or creating subsets of data. Unselected relationships will not be traversed but will be kept with the data to keep the business object information intact.

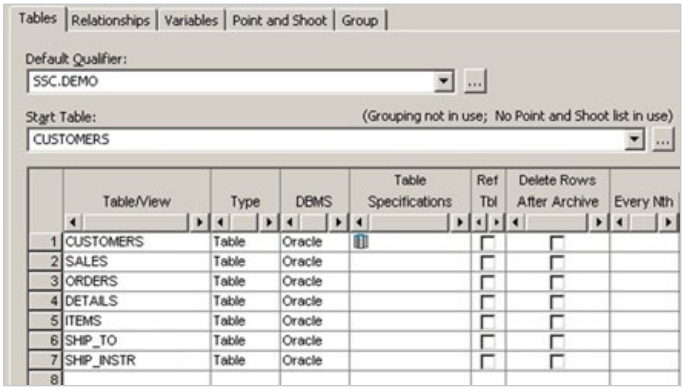

Figure 9: List of tables archived in the Customers complete business object.

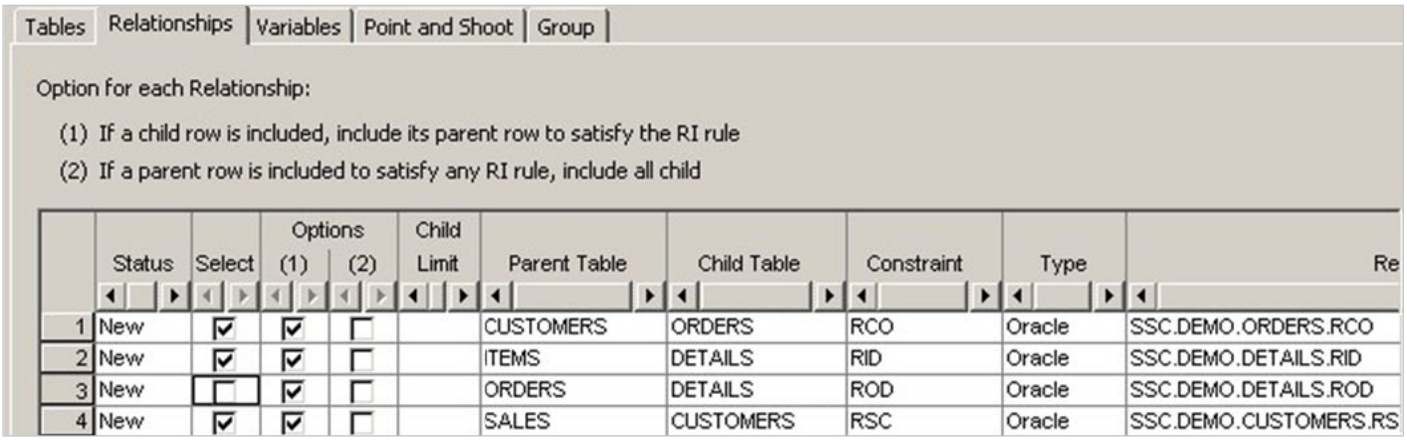

Figure 10: List of relationships displayed in InfoSphere Optim.

A strategy for maintaining data integrity and accessibility

Complete business objects may be only a part of a data lifecycle management strategy, but they play a critical role in helping to ensure that data lifecycle management activities such as archiving, subsetting and masking are successful.

Archiving data with the complete business object in mind enables organizations to improve their data governance processes and helps them comply with regulatory requirements. Each archived object contains all data that pertains to the object at the time of archive. The data and its relationships have been preserved to help ensure data integrity and accessibility—even if the original enterprise application is no longer available.

About IBM InfoSphere

IBM InfoSphere Optim is a part of the InfoSphere portfolio, which creates an integrated platform for defining, integrating, protecting and managing trusted information across your systems. The InfoSphere platform provides the foundational building blocks of trusted information, including data integration, data warehousing, master data management and information governance, all integrated around a core of shared metadata and models. The portfolio is modular, helping you to mix and match InfoSphere software building blocks with components from other vendors or to deploy multiple building blocks together for increased acceleration and value. The InfoSphere platform offers an enterprise-class foundation for information-intensive projects, designed to provide the performance, scalability, reliability and acceleration needed to simplify difficult challenges and accelerate the delivery of trusted information to your business.

For more information

To learn more about IBM InfoSphere and InfoSphere Optim solutions, please contact your IBM sales representative or visit:

- **ibm.com**/software/data/infosphere
- **ibm.com**/software/data/optim

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
June 2012

IMW14622-USEN-00