SEMANTIC OBJECT MODELING SYSTEM FOR CREATING RELATIONAL DATABASE SCHEMAS

This application is a continuation of application Ser. No. 08/145,997, filed Oct. 29, 1993 now U.S. Pat. No. 5,548,749.

FIELD OF THE INVENTION

The present invention relates to computer systems in general and, in particular, to systems for creating database schemas.

BACKGROUND OF THE INVENTION

As the cost of computers decrease, more people are purchasing computer systems that have sophisticated computing and data storage capabilities. In addition to the word processing and spread sheet programs used with such computer systems, another common use for such systems is to store and retrieve data using commercially available database management system (DBMS) products. For example, a home computer system user may wish to store and access information regarding an investment portfolio on the computer. Similarly, the owner of a small business may desire to use a computer system to keep track of its customers, inventory, and sales orders. While commercially available database programs allow a user to store and retrieve these types of information, currently available database programs have a steep learning curve that discourages the average computer user from developing any but the most simple of databases.

Most conventional database programs use what is commonly known as a relational database architecture. These relational databases consist of one or more two-dimensional tables that contain numerous rows and columns. Each row of the relational table defines a record of the data. For example, in a customer table, each row refers to a particular customer in the database. The columns of the tables store particular attributes of each record. For example, one column may hold a customer's last name, another the customer's first name, and another the customer's street or post office box number, etc. The structure of the relational tables that store the data comprising the database is commonly referred to as the database schema.

The difficulty with using a commercially available database program is knowing how to define the database schema. Typically, in order to create anything but the most simple database, the user must either become an expert database designer or hire a consultant to create the database. Both options are generally undesirable because most computer users do not have the time or the desire to become expert database designers, and the cost of hiring a consultant can be prohibitive.

Another problem with current database programs is that they force the user to define the data that they wish to store in the database in a way that is determined by the database program. For example, to create a new database, most database programs present the user with a blank table and ask the user to identify the type of data to be placed in each column. If more than one table is required for defining the database schema, the user must create each additional table and define a key field or attribute that is common to two or more tables in order to relate one table to another. The problem with this method is that the user most likely does not think of the data to be stored in terms of tables, columns, and keys. For example, the user who wishes to create a database for the investment portfolio data most likely thinks of the data in terms of the names of companies in which stock is held, the number of shares of each company owned, a purchase price, a purchase date, a price-to-earnings ratio, etc. By requiring the user to define relational tables in conformance with rigid rules, the commercial database program forces the user to think of and characterize the data to be stored in a way that is unnatural.

Therefore, there is a need for a system that allows a user to create a relational database schema in a way that does not require the user to be familiar with the underlying database technology or rules for defining a database. The system should be easy to use and be able to run on commonly available computer systems. In addition, the user should be able to define the data to be stored in a way that mirrors the user's view of the data.

SUMMARY OF THE INVENTION

To address these problems associated with prior art database systems, the present invention is a system for allowing a user to easily produce a database schema. The system allows a user to create an album that defines a semantic object data model of a plurality of relational database tables that define the database schema.

The semantic object data model defined by the album includes one or more semantic objects, each of which includes one or more attributes that define a characteristic of the semantic objects. The attributes are defined as being: (a) simple value attributes that describe a single characteristic of the semantic object; (b) group attributes that include one or more member attributes, which collectively describe a characteristic of the semantic object; (c) formula attributes that set forth a formula, which describes a characteristic of a semantic object; or (d) object link attributes that define a relationship between one or more semantic objects. Once the album has been created, the system validates the album to determine if the user has created any modeling errors. After the album has been validated, the system transforms the semantic objects and their included attributes contained in the album into a plurality of relational database tables that will store data as defined by the semantic object data model.

The present invention also comprises a method for validating an album to determine if any semantic objects in the album are unconnected to the remaining semantic objects of the album. The validation method also determines if a semantic object in the album is uniquely identified by another semantic object, which in turn is uniquely identified by the first semantic object. The validation method also determines if a formula attribute is recursively defined.

The present invention further comprises a method for interpreting formula attributes. The method includes the steps of searching the semantic objects for the names of the terms used in an expression property of a formula attribute. The system performs an expanding ring search from the formula attribute to the semantic object that contains the formula attribute and to the remaining semantic objects in the semantic object data model. The search produces one or more lists of attributes that are located at the same logical distance from the formula attribute. The lists are searched to find all instances of attributes having the name of the attribute used in the formula expression property, as well as all paths to a single attribute used in the formula. If there exist two or more attributes within the semantic object data model having the same name or two distinct paths to the same attribute used in a formula, the system prompts the user to decide which attribute or path to the attribute is to be