

RELATIONAL SCHEMA FORMAT

TECHNICAL FIELD

[0001] This invention is related to data schemas, and more specifically, a way to represent a relational data schema in an XML format.

BACKGROUND OF THE INVENTION

[0002] Relational schemas are always implicitly stored in the database itself and can be retrieved by querying the database metadata. In addition, relational schemas can be represented by the DDL (Data Definition Language) statements that were used to create the schemas in the first place. However, neither of these formats can be used as a declarative, implementation-neutral format that can be easily stored in a file and used by applications to discover the relational schema.

[0003] What is needed is an improved architecture that represents relational schema in a declarative implementation-neutral and/or implementation-specific format.

SUMMARY OF THE INVENTION

[0004] The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is not intended to identify key/critical elements of the invention or to delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

[0005] The present invention disclosed and claimed herein, in one aspect thereof, comprises a Relational Schema Definition (RSD) language that allows representing any relational schemas in an alternative declarative format, for example, XML format. In one embodiment, the RSD format is expressed as an XML-based language used to represent relational schemas. The RSD language component uses a tool to extract metadata of the relational database, and to format the metadata in to an RSD file. The RSD file represents the precise structure of the database, and in more robust implementations, the data, such that the corresponding database can be reconstructed when remote and disconnected therefrom. The file also contains information on the database type system, as well as the routines (functions and procedures) that are exposed by the database. This is particularly suited by a traveling user, who may use the RSD file to interact with the database while disconnected.

[0006] In another aspect thereof, the user can update the disconnected database via the RSD file, such that the updated RSD file can now be remerged with the original database to reflect changes made thereto by the user while disconnected.

[0007] In yet another aspect thereof, the RSD language component can be used to facilitate mapping between at least two different data models. The RSD file is generated into an XML format, which XML format is easily processed by software that interacts with other types of data models to which data mapping is desired.

[0008] To the accomplishment of the foregoing and related ends, certain illustrative aspects of the invention are

described herein in connection with the following description and the annexed drawings. These aspects are indicative, however, of but a few of the various ways in which the principles of the invention may be employed and the present invention is intended to include all such aspects and their equivalents. Other advantages and novel features of the invention may become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 illustrates a general block diagram of a system of the present invention.

[0010] FIG. 2 illustrates a flow chart of a process for generating RSD from the relational database.

[0011] FIG. 3 illustrates a general block diagram of the RSD language component of FIG. 1.

[0012] FIG. 4 illustrates a general block diagram of an RSD component that further employs a classifier to perform one or more automated functions.

[0013] FIG. 5 illustrates a block diagram of a system where the RSD component is located remote to the relational database from at least one relational database.

[0014] FIG. 6 illustrates a block diagram of a distributed system where multiple relational databases each have respective RSD files that when retrieved can be combined provide a more comprehensive view of the overall database.

[0015] FIG. 7 illustrates a block diagram of a computer operable to execute the disclosed architecture.

[0016] FIG. 8 illustrates a schematic block diagram of an exemplary computing environment in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0017] The present invention is now described with reference to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It may be evident, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing the present invention.

[0018] As used in this application, the terms “component” and “system” are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

[0019] As used herein, the term “inference” refers generally to the process of reasoning about or inferring states of