Relational Databases vs. Object-Relational Databases

Bryan Mack

MSCD 640 – Oracle Database Administration

Brad Blake

1-30-08

Abstract

The relational database (RDB) was introduced in 1970 by E.F. Codd. This has since served as the standard for a typical database management system (DMBS). In recent times, a greater need for the incorporating of objects and greater scalability as brought about object-relational databases. Object-Relational Databases (ORDB) are a hybrid of a relational database and an object-oriented database. Oracle, as well as other major DBMS vendors, have options to implement either type of database enabling the database to maximize the benefits of the features of both system types.

Analysis and Discussion

In 1970, E.F. Codd introduced the relational database, which is run by a relational database management system (RDBMS). This two-dimensional methodology is constructed of tables containing tuples (rows) which each represent a record. Each record is broken down into a defined number of attributes, represented by the vertical fields in the table. Each row must be uniquely identified within a table by one or a combination of multiple attributes. These unique identifiers are known as keys; a key which uniquely identifies a row in a particular table is called the table's primary key. Separate tables can be joined together by connecting unique row identifiers; these unique identifiers in other tables are known as foreign keys. (Devarakonda, R. n.d.)

Tables and values within tables in a relational database are constructed through the use of data definition language (DDL). In this DDL, the table and its attributes are all named, the attributes are assigned datatypes, and the table is assigned rules (constraints) to which its data must follow. DDL is a form of structured query language (SQL).

Another form of SQL is data manipulation language (DML). DML is used to modify, insert, and query data within the database.

With the development of the relational database over the years, needs have increased to include objects and other rich data types within the storage mechanism. For this reason, the object-relational DBMS (ORDBMS) was developed. The ORDB uses SQL in all forms just as the RDB uses it. In fact, the ORDBMS contains all the features of an RDBMS, but adds the capability for users to define datatypes and functions. Similarly, complex objects such as GIS data, audio, video, and images can all be stored within an object-relational database. Simply put, the makeup and structure of an ORDB is the same as an RDB, however, the ORDB contains complex data whereas the RDB contains simple data (Devarakonda, R., n.d., Table 1).

Videos, GIS data, audio, and images can all be stored in an ORDB as objects. Oracle can store these in fields such as BFILE or Binary Large Objects (BLOB). A BFILE links to a file external to the database, meaning the file is not actually stored in the database and resides elsewhere on a local machine or on a networked server. A BLOB is an internally stored document. The BLOB stores the composition of the object (whether it be video, PDF, or audio) within data blocks for the schema object (table) which it is to be stored in.

One of the main benefits of an ORDB is the ability to construct custom datatypes for which an attribute must conform to. One of the areas an ORDB is able to inherit from object-oriented programming languages is the ability to recognize these datatypes. In object-oriented languages such as Java, a datatype model can be constructed and subsequently imported to Oracle for use within the DDL when defining an Oracle table's

field datatype(s). Jeff Ullman states (1998) that objects can also be created (or defined) in SQL by using the CREATE TYPE AS OBJECT syntax. If a custom datatype is to be removed, whether it's been constructed in an object-oriented style or defined using SQL, all tables and other types that reference this datatype must be removed from the database.

Custom datatypes can also have defined methods for the values of the attributes using this datatype. When creating a datatype, the method is declared as a MEMBER FUNCTION or MEMBER PROCEDURE in the DDL statement. The code for the function or procedure itself is contained in a CREATE TYPE BODY statement. The method can be used to describe the data contained in the custom datatype (Ullman, 1998). An example of this would be a method which returns the length of the attribute. In a RDB, this would return the number of characters, in an ORDB, this can be customized to list whatever it is you would like the method to display.

In Oracle, not only can columns be datatypes, but a custom datatype can be substituted for the schema elements (relations) when creating a table by using the DDL syntax CREATE TABLE tablename OF custom_datatype_name. (Ullman, 1998). This allows for the basic structure of a table to remain the same as it would in an RDB, however the object-based methods can now be used on the rows. These types are called row types.

Other objects that Oracle can store include the VARRAY and nested tables. A nested table is when the entire contents of a table are stored within one attribute (field) of another table. The data within a nested table follows the same guidelines as all parent tables. Fields within the table may have a NULL value, and they can be removed and/or modified without affecting the other attributes of the nested table. A VARRAY is an

array which acts as a single object. This object type can also be input as a single field within a table. Unlike a nested table, the values within a VARRAY are sequential and can not be altered. Data is mandatory within the VARRAY and in order to modify it, all data must be removed, the array size adjusted (if needed), and the VARRAY must be recreated with the new data. Nested tables are useful when the data may change from time to time, while VARRAYs should be used when the data is static with little-to-no chance of needing modification (How To Use, 2007).

Oracle also provides a feature which can convert ordinary relations to object relations. If an existing table contains the DDL of standard SQL datatypes (such as NUMBER, INT, or DECIMAL), it can be converted by simply inserting the values into an existing object row type (Ullman, 1998). This conversion would allow the new table to utilize the standards of the object datatype and the methods which were created for it.

Conclusion

When a relational database needs to utilize complex datatypes within its tables, the use of an object-relational database allows for this. Oracle provides several options for defining and customizing object-relational databases while keeping this type of system similar in structure to relational databases while providing a high level of scalability.

References

Krishanmurthy, V., Banerjee, S., Nori, A. (June 1999), "Bringing object-relational technology to the mainstream." Proceedings of the 1999 ACM SIGMOD international conference on Management of data
ACM Digital Library ISBN:1-58113-084-8

Ullman, J. (1998). Object-Relational Features of Oracle (site visited 1-30-08) http://infolab.stanford.edu/~ullman/fcdb/oracle/or-objects.html

REMOVE HYPERLINKS (UNDERLINES)

Database Models (site visited 1-30-08)
http://www.unixspace.com/context/databases.html

Devarakonda, R. (Site visited 1-30-08). Object-Relational Databasy Systems – The Road Ahead
http://www.acm.org/crossroads/xrds7-3/ordbms.html

How To Use Collection in Oracle (2007)
http://www.digcode.com/default.aspx?page=ed51cde3-d979-4daf-afae-fa6192562ea9&article=bfe8e303-0987-4dec-8c50-20d54bf3db68

ORDER THESE ALPHABETICALLY.

GOOD INFO!

A-