Chapter XII

A Taxonomy for Object-Relational Queries

David Taniar
Monash University, Australia

Johanna Wenny Rahayu
La Trobe University, Australia

Prakash Gaurav Srivastava
La Trobe University, Australia

ABSTRACT

A comprehensive study of object-relational queries gives not only an understanding of full capability of object-relational query language but also a direction for query processing and optimization. This chapter classifies object-relational queries into REF queries, aggregate queries and inheritance queries. REF queries are queries involving REF pointers, whereas aggregation queries use either nested table structures or index on clusters. Finally, inheritance queries are queries on inheritance hierarchies.

INTRODUCTION

There have been many notions in recent times about Object Relational Database Management Systems (ORDBMS), but still different people have different understandings of the concept (Dorsey & Hudicka, 1999). ORDBMS is based on SQL3, which consists of a basic relational model along with objects, row types, collections and abstract data types (Fuh et al., 1999). Incorporation of a new object feature has given rise

to object thinking in which data, its associated operations and methods are considered together while designing and developing the system. The increase in the size and complexity in the data is demanding a new type of database, which should efficiently handle both the issues without affecting the overall performance. The advantage associated with a database based on the SQL3 model is that the system is more flexible and robust so that the changes in the business process or newly discovered requirements can be accommodated easily (Fuh et al., 1999).

The Relational Database Management System (RDBMS) has been investigated well; yet, in all respects the same has not been done for ORDBMS. Most papers have only introduced the new data structures in SQL3 (Carey, 1992; Stonebraker & Moore, 1996; Fuh et al., 1999). These new data structures have given rise to new types of queries, an area that has not been investigated much. This chapter attempts to develop a framework of queries which arise due to new data structures that have been introduced in SQL3. The aim of this chapter is not to introduce the new object features, but to give a clear understanding of the full capability of the new queries that arise due to new data structures.

The chapter is organized as follows. The first section gives a brief introduction about the new data structures and the way they are implemented in SQL 3 standards. Next, the various types of object relational queries based on new data structures are described, and finally, we discuss the essence of this chapter and how the work in this chapter can be extended so as to cover various other areas in object relational queries.

NEW DATA STRUCTURES IN SQL3

The model given in Figure 1 is the base model for the work, which has been carried out in this chapter. The model reflects the database schema of a Web-based tutor payment system for a given university. The data model given in Figure 1 will be used for all our running example queries in this chapter.

The new data structures that have been introduced in Object Relational Databases can be broadly classified into REF, nested tables, index on clusters and inheritance. Nested tables and clusters are commonly used to implement the aggregation concepts.

REF Structure

REF is similar to pointers in object-oriented programming language and is used to define the link between two tables. REF is essentially a logical pointer, which can be used outside the scope of the database (Dorsey & Hudicka, 1999). It is incorporated into a database by defining one attribute in the table, which holds the REF information of the attribute, which belongs to the other table. REF is used when there is an association relationship between two objects. Association relationships can be of three types: *many to many*, *one to many*, and *one to one* (Loney & Koch 2000). REF is used in different ways to establish an association relationship depending on the type of association relationship. The general syntax for implementing REF is given in

An example of *REF* implementation is between Person and Login in Figure 1. This relationship can be implemented in ORDBMS, which in our case is Oracle 9i by the following syntax (see Figure 2).

Related Content

A Socio-Technical Interpretation of IT Convergence Services: Applying a Perspective from Actor Network Theory and Complex Adaptive Systems
Myeong Ho Lee (2009). *Advanced Principles for Improving Database Design, Systems Modeling, and Software Development (pp. 308-328).*
www.irma-international.org/chapter/socio-technical-interpretation-convergence-services/4304

Databases for Mobile Applications
Indranil Bose, Wang Ping, Mok Wai Shan, Wong Ka Shing, Yip Yee Shing, Chan Lit Tinand Shiu Ka Wai (2005). *Encyclopedia of Database Technologies and Applications (pp. 162-169).*
www.irma-international.org/chapter/databases-mobile-applications/11140

Conceptual Modeling for XML: A Myth or a Reality
Sriram Mohanand Arijit Sengupta (2009). *Selected Readings on Database Technologies and Applications (pp. 148-173).*
www.irma-international.org/chapter/conceptual-modeling-xml/28551

Object-Oriented Database Benchmarks
Jerome Darmontand Michel Schneider (2002). *Advanced Topics in Database Research, Volume 1 (pp. 34-57).*
www.irma-international.org/chapter/object-oriented-database-benchmarks/4321

Collaboration Matrix Factorization on Rate and Review for Recommendation
Zhicheng Wu, Huafeng Liu, Yanyan Xuand Liping Jing (2019). *Journal of Database Management (pp. 27-43).*
www.irma-international.org/article/collaboration-matrix-factorization-on-rate-and-review-for-recommendation/232720