IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA

Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITB10780

Chapter III

Object-Relational Spatial Indexing

Hans-Peter Kriegel, University of Munich, Germany

Martin Pfeifle, University of Munich, Germany

Marco Pötke, sd&m AG, Germany

Thomas Seidl, RWTH Aachen University, Germany

Jost Enderle, RWTH Aachen University, Germany

Abstract

In order to generate efficient execution plans for queries comprising spatial data types and predicates, the database system has to be equipped with appropriate index structures, query processing methods and optimization rules. Although available extensible indexing frameworks provide a gateway for seamless integration of spatial access methods into the standard process of query optimization and execution, they do not facilitate the actual implementation of the spatial access method. An internal enhancement of the database kernel is usually not an option for database developers. The embedding of a custom, block-oriented index structure into concurrency control, recovery services and buffer management would cause extensive implementation efforts and maintenance

cost, at the risk of weakening the reliability of the entire system. The server stability can be preserved by delegating index operations to an external process, but this approach induces severe performance bottlenecks due to context switches and inter-process communication. Therefore, we present the paradigm of object-relational spatial access methods that perfectly fits to the common relational data model, and is highly compatible with the extensible indexing frameworks of existing object-relational database systems, allowing the user to define application-specific access methods.

Introduction

Users of database systems want to manage data of very different types, depending on the particular application area. While office applications, for example, mainly perform simple access and update operations on records of simple data types, spatial data usually have a complex structure and demand specialized operations. It is not a choice for vendors of database management systems to provide data types and management functions for each conceivable domain. So the design of *extensible architectures* allowing users to adapt systems to their special needs represents an important area in database research.

Traditional relational database systems support very limited extensibility. All data have to be mapped on rows of flat tables consisting of attributes with such simple types as numbers, character strings or dates. For the retrieval and manipulation of data, there exist only generic operations for selecting, inserting, updating and deleting (parts of) rows within tables. Data of more complex types cannot be stored directly as a unit in the database but have to be split across several tables. To restore the data from the system, complex queries with many joins have to be performed. Alternatively, the data can be coded within a large object that prevents direct access to single components of the data using the database language. Operations on complex types have to be implemented within the application and cannot be used within the database language directly.

Object-oriented database management systems (OODBMS) seem to provide solutions for most of the cited problems of relational databases. An OODBMS has an extensible type system that allows the user to define new data types (by the nested application of type constructors) together with corresponding operations. The resulting *object types* then describe the structure as well as the behavior of the objects based on this type. Furthermore, subtypes (inheriting the properties of their supertypes) can be derived of existing object types.