Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research

www.ijmter.com

Overview of Indexing In Object Oriented Database

Rutuja P Fale[1], Prof. P J Pursani[2]
Final CSE (Direct), H.V.P.M.'s C.O.E.T., Amravati
Assistant Professor, CSE Dept.,H.V.P.M.'s C.O.E.T., Amravati

Abstrac-In conventional database an index is maintain on an attribute of single class to speed up association research. In object oriented database the access scope of query against a class in general includes not only the class but also all subclass of the class. This means that to support the evaluation of a query, the system must maintain one index on an attribute for each classes involve in query.

I. Introduction

Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. In object oriented database software indexing is essential for improve performance, as linear search is inefficient for large databases company many technologies are work in indexing there is research on indexing.

II. Database indexing

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure. Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.

An index is a copy of select columns of data from a table that can be searched very efficiently that also includes a low level disk block address or direct link to the complete row of data it was copied from. Some databases extend the power of indexing by letting developers create indices on functions or expressions. For example, an index could be created on upper (last name), which would only store the upper case versions of the last name field in the index. Another option sometimes supported is the use of partial indices, where index entries are created only for those records that satisfy some conditional

expression. A further aspect of flexibility is to permit indexing on user-defined functions, as well as expressions formed from an assortment of built-in functions.

A proposal is presented for a general indexing method, called a generalized index, for object-oriented databases which can be implemented on all kinds of hierarchical and nonhierarchical structures. A generalized index is a domain based indexing structure in which are maintained the value and object identities of an attribute, the class to which that value belongs, and the name of the attribute and/or the path to which that attribute value belongs. This information is stored in a tree structure referred to as Index Tree.

Indexes are essential components in database systems to speed up the evaluation of queries. To evaluate a query without an index structure, the system needs to check through the whole file to look for the desired tuple.In RBDS, indexes are especially useful when the user wishes to select a small subset of a relation tuples based on the value of a specific attribute. In this case, the system looks up the desired attribute value in the index (stored in B-trees, or hash tables) and then retrieves the page that contains the desired tuples. Using index for searching influences the performance of producing the result but not the result itself. Indexing in OODBS is a lot more complicated than in RBDS. One difference between objects and relational tuples is that objects are not flat. Therefore one should be able to index on instance variables that are nested several levels deep in an object to be indexed. Indexing for OODBS is first proposed for the data model. It is a generalization of an indexing technique for path expressions.

III. Issues in Indexing Object Oriented Database

The basic need for complex structure is to efficiently select from a collection whose members meeting a selection criterion. All the objects that either contain given object, or contain an object equal to a given object have to be found.

3.1 Index on classes

Authorization problems occur if indexing on classes. For example, a user may have access to a Student object but is prohibited to the instance variable course History Allowing a user to build an index on Students could allow him to access some unauthorized information. On the other hand, if a user is prohibited to access one or some of the instances of a class, how should indexes be built in this class? For example, a professor may have access to Students that attend his lectures, but not other Students. To authorize access to certain student objects is complicated if indexing is applied on the

Student class. An alternative is to apply index on collections, and only add desired members to a collection; but then each object must be able to reference a number of indexes to support update, as an object may be contained in several collections.

3.2 Indexing over type hierarchy

The authorization issue is also raised here when all objects of indexed object subclasses are also indexed. The evaluation of a query over super class objects will retrieve also objects of its subclass. For example, the Manager class is a subclass of the Employee class. By applying index on Employee including Manager, a user who is prohibited to access the Manager instances can get the attribute of a manager through querying on the Employee. However, if indexing on super class and its subclasses individually, the evaluation of a query over the class hierarchy involves a lookup in several index structures and a union of the results.

3.3 Uni-directional or bidirectional index

Uni-directional index is a one-way reference from one object to another, as bi-directional index does two-way links. Two-way links have the advantage of supporting both forward and backward queries, whereas one-way link supports only one of them. Two-way link is however problematic, as an object may be the value of an instance variable in several objects. For example, the same Publisher instance can fill the published variable of many Book objects.

IV. Literature review:

1. In this paper [Bertino & Kim, 1989] author have discuss the basic functionalities of database system (DBMSs) is to be able to process declarative user queries. The first generation of object oriented DBMSs did not provide declarative queries. The first generation of object oriented DBMSs did not provide declaration query capabilities. However, the last decade has seen significant research in defining query models (including calculi, algebra and user languages) and in techniques for processing and optimizing them. Many of the current commercial system provide at least rudimentary query capabilities. In this chapter we discuss the techniques that have been developed for processing object-oriented queries. Our particular emphasis is on extensible query processing architectures and techniques.

2. In this paper [Trzeciak and Sexton, 1997] authors have discuss object oriented database have become one of the high pro areas of data and software engineering. Much work has been done to improve query executions, performance. Index are crucial in cutting the costs of updating and retrieval operations performed on database. In this paper we present various indexing techniques from join indexes to the enhanced nested-inherited index.

3. In this paper [Pursani and Raut, 2014] author have discuss the indexing technique for Fuzzy Object Oriented Database Model to support complex data and to handle fuzziness in the complex database along with indexing to speed up the evaluation of fuzzy queries. In today's era as traditional commercial market is changing rapidly towards specialized market where need of internet is increasing. So to satisfied market Object Oriented Database (OOD) has been developed which would soon become the primary database technology as Relational database were not designed to handle the type of multimedia data present on the web is high uncertain. So to handle the uncertainty on the web, Fuzzy Object Orient Database (FOOD) model is required. Due in this paper FOOD index is proposed which deals with various kinds of fuzziness and provides indexing technique based on R tree indexing which supports various fuzzy queries. Index Terms: Object Oriented Database, Fuzzy Set theory, Fuzzy queries, Fuzzy Indexing, R tree indexing.

4. In this paper [David Maier] author have introduced the indexing in Object Oriented Database and their importance in it.

5. In this paper [Heller stein] author have describe the implementation of complex types in Object-Relational database system required the development of efficient access methods. In this paper we describe the RD-Tree, an index structure for set-valued attributes. The RD-Tree is an adaptation of the R-Tree that exploits a natural analogy between spatial objects and sets. A particular engineering difficulty arises in representing the keys in an RD-Tree. We propose several different representations, and describe the tradeoffs of using each. An implementation and validation of this work is underway in the SHORE object repository.

V. Conclusion:

To enhance the performance of Object Oriented Database, indexing technique is used. Indexing aims to retrieve results fast from the database and hence it increase query processing. Thus research is going on to develop various indexing techniques on Object Oriented Database.

References

[1]. [Bertino & Kim, 1989] E. Bertino and W. Kim "Indexing Techniques for Queries on Nested Objects", *IEEE Tans. On Knowledge and Data Engineering*, vol. 1, no. 2, 1989

[2]. [Trzeciak and Sexton, 1997] Arthur Alan Trzeciak and P. Sexton *A Survey of Indexing Techniques for Object-Oriented Databases,* December 1997

[3]. [Pursani and Raut, 2014] Priyanka J. Pursani and A.B. Raut "FI: Fuzzy Object Oriented Database (Food) Index", 2014

[4]**. [**Maier**]** David Maier *Object-Oriented Database Theory An Introduction & Indexing in OODBS.*

[5]. [Heller stein] Joseph M. Heller stein *THE RD-TREE: AN INDEX STRUCTURE FOR SETS.*

[6]. [Kappel and Retschitzegger,2001] Kappel, G., RauschSchott,S., Retschitzegger, W. Bottom Up Design of Active Object Oriented Databases. Communications of the ACM, 2001

[7]. [Fayad and Schmidt et, 1997,1992] Fayad, Mohamed E., Schmidt, Douglas C. Object Oriented Application Frameworks. Communications of the ACM, 1997.Loomis, Mary E. S. ODBMS versus Relational. JOOP Focus on ODBMS, 1992

[8]. **[**Atkinson and Banchilhon et.,1989**]**M. ATKINSON, F. BANCILHON, D. DEWITT, K. DITTRICH, D. MAIER, S. ZDONIK, the Object-Oriented Database System Manifesto. In *Proceedings of the First International Conference on Deductive and Object-Oriented Databases*, pages 223-40, Kyoto, Japan, December 1989.

[9]. **[**Kim and Lochovsky, 1989**]**W. KIM AND LOCHOVSKY (EDS), *Object-Oriented Concepts, Databases, and Applications,* Addison-Wesley (Reading MA), 1989

[10].**[**Ullman, 1989] JEFFREY D. ULLMAN, *Principles of Database and Knowledge-based Systems, Vol.2,* Computer Science Press, 1989(2nd volume of 2-volume textbook.)

[11]. [Michael, 1998] Kofler Michael, Ph.D. Thesis, *R-trees for Visualizing and Organizing Large 3D GIS Databases,* Technischen Universidad Graz, July1998.

