World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:11, 2007

Contribution to the Query Optimization in the Object-Oriented Databases

Minyar Sassi, and Amel Grissa-Touzi

***Abstract*—**Appeared toward 1986, the object-oriented databases management systems had not known successes knew five years after their birth. One of the major difficulties is the query optimization. We propose in this paper a new approach that permits to enrich techniques of query optimization existing in the object-oriented databases. Seen success that knew the query optimization in the relational model, our approach inspires itself of these optimization techniques and enriched it so that they can support the new concepts introduced by the object databases.

***Keywords*—**Query, query optimization, relational databases, object-oriented databases.

I. Introduction

ONE of the relational databases successes is that the optimization of its queries have been studied well and showed its proofs in the speed of query execution. However, the relational model only permits the alphanumeric data management. Nowadays, the necessity to support complex data in databases is intensified. Models trying to answer to these needs appeared as the object-oriented and the object-relational model [1].

Some either the chosen model, the goal always remained after having stocked these data, to be able to interrogate the basis and to have an answer in one appropriate time. The definition of a query in these new models becomes larger than the one in the relational model, since it must take in consideration the new concepts introduced by these models.

Query optimization in relational databases is benefited a lot from the simplicity of the data model. This is not the case with the object model. Indeed, a query must use the new concepts introduced by object model as multi-valuated attributes, drawers etc [2].

To answer such a query, it is necessary to try to surround the problems introduced by these concepts [1]-[2]-[3].

The paper is organized in six sections. Basic concepts are presented in Section II. Query notion in the object databases is presented in section III. Query optimization approach that we propose is described in Section IV. In section V, we compare our approach to the others. The balance of this work and its future perspectives are discussed in Section VI.

Manuscript received May 20, 2005.
M. Sassi, A. Grissa-Touzi., with the Department of Electrical Engineering of National School of Engineers of Tunis, Tunisia (e-mail: amel.touzi@ enit.rnu.tn, sassi_minyar@yahoo.fr).

II. Basic Concepts

A. Object-Oriented Databases

We suppose known notions of databases and of relational databases. Us we limit to present, in this section the basic concepts of object model.

Object models are descended of the semantic networks and object programming languages. They aim to permit the reuse of structures and operations to construct some more complex entities. The new concepts introduced by this model are:

The **object** is a triplet <OID, class, state>, the OID is identifying of the object. It's unique and invariant during the program, contrary to its state (value of attributes of the object) that can vary.

The **attribute** is defined by its name and its type. An operation is a function that permits to modify the state of an object or to send back a value.

The **class** is a data abstract type permitting to define properties of a whole of objects regrouped in two categories: attributes and operations [1]-[4]-[5].

The **inheritance** is a transmission mechanism of properties of a class toward one under class. The inheritance is simple if the property is inherited of only one on-class. It is multiple when the property is present in several on-classes [1]-[2].

The **polymorphism** is the fact to arrange operations having one same name but of the different parameters in number or in types.

An object database is a coherent organization of shared obstinate objects by user's competitors [1].
Example:

Let's the database School describing Professor, Student, Country and City. Professor is described by NCI, Name, Birthday, Nationality, Type, and Salary. Student is described by NCI, Name, Birthday, Nationality and NCE.

Country is described by Name, Population Capital and Population. City is described by Name and Population.

It is clear that the classes Professors and Student can be considered as classes inherited the class personne describing by NCI, Name, Birthday and Nationality.

The object diagram of the database is presented in Fig. 1.

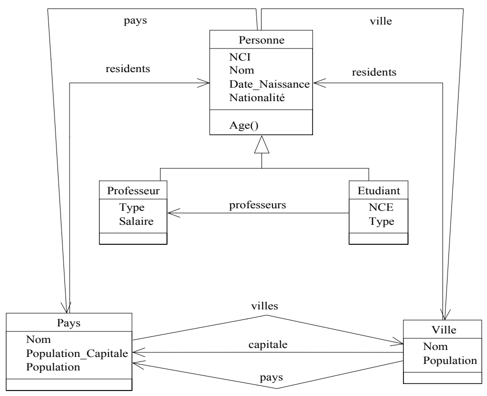

Fig.1 Object Database Diagram

B. *Relational Query Notion and Query Optimization in the Relational Databases*

A query is an expression that describes information that one wants to search for in a database. The result of a query is a relation described with the simple attributes.

The query optimization is the process of selection of the best path of access to data in a database [5]-[6]-[7]-[8].
Process of optimization is summarizes in three steps (see Fig.2). *Rewrite step* consists in a syntactic and semantic rewrite of the query in the goal to determine simpler equivalent queries [1]. The result of this step is the generation of a query graph. *Ordering operations step* is takes place in two phases: generation and assessment of the plans which determined in the first phase. *Execution step* permits to choose the optimal execution plan and to execute it. Two approaches present themselves, by materialization or by pipeline.

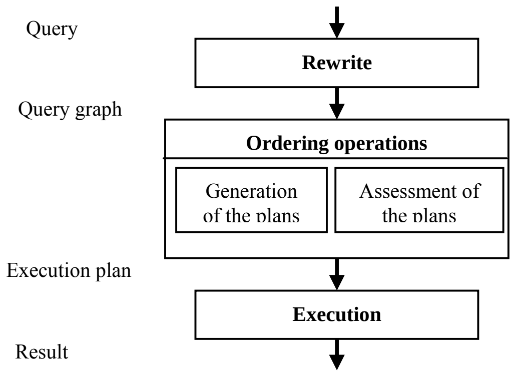

Fig.2 Optimization Process

III. Query Notion in the Object Database

An object query is a question that permits to extract data of the object database. The result can be of type: objects, values or OID.

A. *Types of Query Results*

In relational model, there are only two concepts: the relation and the attribute. The result of a query is always a new relation (with its attributes).

In object model, there are more concepts: objects, values, OIDs, methods. It is necessary to choose between them, the one (those) that will be the type of the result: objects, value, OIDs [2]-[3].

B. *Object Model Problems*

Object queries integrate some inexpressible constructions in relational model. In this fact, they put some specific problems. The new points are:

1) The support of methods and operators defined by the user.
2) Path crossings.
3) Complex structure of objects: complex attributes, multi-valuated.
4) The hold in account of the inheritance and the polymorphism.
5) The Encapsulation

Therefore, the object query optimization reveals the main problems follows:

-To be capable to generate all possible execution plans for a complex query. It requires, first of all, of the new techniques of query rewrite using the new object concepts.

-Choice of algorithms and the most effective access methods. This choice is richer than in the relational case, because the joints can take place directly by crossings of drawers [1]-[2].

-Choice of the best plan. This choice requires a cost model permitting to estimate the cost of a plan to the compilation. This model must take in account the courses of drawers.

IV. Query Optimization Approach in Object-Oriented Databases

Seen the success that the query optimization knew in the relational model, we tried in our approach to adapt it to the new object concepts.

Data are represented in the basis as of objects. Associations are implemented by the direct ties via object identifying that permit a fast navigational access between the different objects. While specifying in the class diagram a mono-valuated or multi-valuated tie toward another class, queries can follow this ties in order to attend the searched objects [3]-[9].

In this case, alternatives of calculation for such a query are not constructed anymore while using the algebraic properties of operations, but, mainly, while using the semantics of ties that joins these objects.

This semantic knowledge can be expressed by constraints of integrities and, while using them as of rewrite rules, the optimizer can determine the equivalence of the two expressions.

Our approach bases itself on the idea of an adaptation of the optimizer definite for the relational databases to object databases. Therefore, optimizer must follow the three steps of optimization described previously, has known: rewrite, ordering operations and execution.

A. *Query Rewrite Step*

In this step, we defined the following rewrite rules:

Rule1

If a class C1 has a tie ***a*** toward a class C2 and the class C2 has a tie ***b*** toward the class C3, it is possible that the class C1 possesses a direct tie toward the class C3 in order to avoid the navigation through objects of the class C2. This tie is the composition of ties ***a*** and ***b***.

Rule2

Transform the navigation between objects in joints.

Rule3

Transform the independent joints in dependent joints.

Example:

We considers in the following the object-oriented database described in the Fig.1.

Either the expression corresponding to the following query Q: To "search for all people living to Tunis, in a city named Nabeul ".

(Q)

Select x
from x in Personne
where x.pays.Nom="Tunis" and x.ville.Nom="Nabeul"

While applying the rule 1, we get the equivalent expression Q_1 to Q (to replace the x.pays under-expression by x.ville.pays)

(Q_1)

Select x
from x in Personne where x.ville.pays.Nom="Tunis" and x.ville.Nom="Nabeul"

While applying the second rule, we get the equivalent expression Q_2 to Q1:

(Q_2)

Select x
from x in Personne, y in Ville
where x.ville=y and y.pays.Nom="Tunis" and y.Nom="Nabeul"

While applying the third rule, we get the equivalent expression Q_3 to Q2:

(Q_3)

Select x
from y in Villes, x in y.residents
where y.pays.Nom="Tunis" and y.Nom="Nabeul"

We get three formulations of the same query with the potentially different assessment costs.

B. *Ordering Operations Step*

By analogy to the relational model, this step takes place in two stages. The passage to an object model complicates the problem of ordering operations for several reasons:

1) The query structure is different. Indeed, a “From” clause collection of an expression can be specified:
   - By another expression (overlapped queries),
   - By a simple expression,

   The order in which “where” clause conjunctions of an expression are valued influences on query semantics.

   In ordering operations step, one must take in consideration the following results:

   -The notion of joint graph representing the entrance of this step must be spread in order to integrate the generality of the OQL language;

Example:

Let's consider the following query:

(Q)

Select x.ville
from x in Personne, y in Pays, z in y.villes
where x.age>=50 and x.pays=y and x in z.professeurs and x.salaire>=50.000

The joint graph of this query can be represented in the Fig.3. It represents the existing assessment constraints in the query expression.

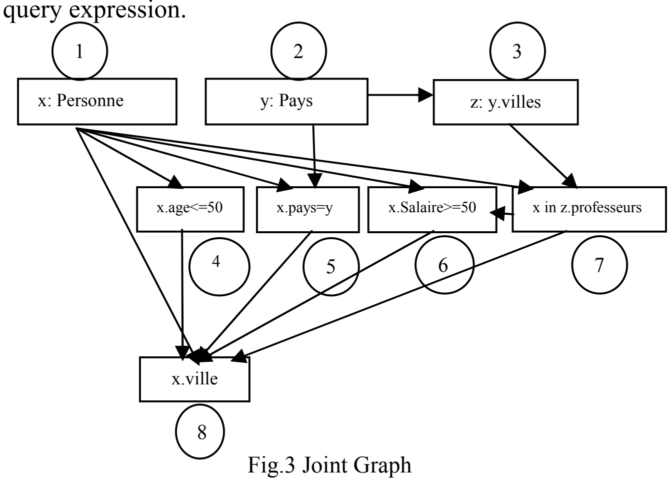

Fig.3 Joint Graph

-The operator of object joint doesn't verify the algebraic properties of commutability and associativity [10]-[11]. It is necessary to examine other techniques or algorithms permitting to generate a whole of equivalent binary joint trees from a joint graph.

2) To spread the method of tree generation. This method must take in entrance the joint graph corresponding to a defined OQL query that verifies the whole of assessment constraints specified by the graph. The algorithm must define for every collection a whole of predecessors. If, for a collection data, the whole of these predecessors is hopeless, this last is susceptible to begin pipeline chain.

Example:

In the Fig. 3, collections Personne and Countries are susceptible to begin pipeline chain. On the other hand, the y.villes collection cannot be considered.

The construction of the first operation of pipeline chain makes in two steps:

- Transformation of the initial query in an overlapped query, having like interior query the expression that one can form with the collection Personne and the predicate x.age <=50. Graphically, it is represented by the unification of nodes corresponding to these three expressions in only one node, whose label is the internal expression. To this point, alone collection Country is susceptible to be visited.

```
a: select x
    from x in Personne
   where x.age<=50
```

- Construction of the second operation of pipeline chain, with in parameters, collection Pays and the predicate a.pays=y. By a similar transformation, the previous query is transformed in an overlapped query.

```
b: select [F1:=a, F2:=y]
   from a in (select x
              from x in Personne
               where x.age>=50 ),
        y in Pays
where a.pays=y
```

Finally, the y.villes collection is visited, forming the last operation of pipeline chain.

```
Select b.F1.ville
from b in (select [F1:=a, F2:=y]
           from a in (select x from x in Personne
                          where x.age>=50),
                  y in Payss
             where a.pays=y)
  z in b.F2.villes
where b.F1 in z.professeurs and b.F1.Salaire>=50.000
```

After generation phase, we must spend to the assessment phase. An expression is only valued when all variables used in its construction are instantiated. The ascending strategy of tree construction is useful to generate only trees whose cost is lower to a certain limit that is the parameter of the procedure. At the time of the construction of one sub-tree whose cost already passes the L limit, the procedure doesn't continue research on this branch, because the cost of a tree is bigger than the cost of a sub-tree.

C. *Execution Step*

The notion of pipeline can be used also in our execution model, while replacing the notion of relation by the one of collection and the notion of tuple by the one of element of collection.

V. Comparaisons with Others Works

Several methods of query optimization have been proposed, GemStone [12], O2 [13], Orion [14] and Blackboard [15]. In our approach, the supplementary difficulties brought by the object context have been put in evidence. By analogy to the relational databases, we treated the object query optimization in two main phases.

The first translated query in those equivalents by application of the semantic rules using the new object concepts. Then, we presented the problem of ordering operations and the extension of the joint graph to capture the whole of assessment constraints to which a query is submitted. Finally, we presented a model of execution of similar query trees to the model of execution in the relational systems.

VI. Conclusion

We proposed in this paper an approach of object query optimization. We enriched techniques of relational query optimization so that they can support the new concepts introduced by the object model. We proposed formalism for the specification of course paths that can exist in object database, as well as rules of query rewrite in equivalent queries that use this type of information. Construction of an efficient execution plan is the most important phase in the query optimization. In the relational systems, this phase is the joint's ordering. It takes in entrance the joint graph. In the object context, the notion of joint graph must be spread in order to integrate the generality of the OQL language.

This work can be spread while proposing to adapt our method to the Object-relational databases.

References

[1] G. Gardarin, "*Object and relational databases*" , Eyrolles, 1999.

[2] G. Gardarin and P. Valduriez., "*Relational Databases and Knowledge Bases.Addison-Wesley Publishing Company*", Reading, Massachusetts, 1990.

[3] R.G.G. Catell, "*Object-Oriented Data Management: Object-Oriented and Extended Relational Database Sytems*", Addison-Wesley Publishing, Inc., 1994.

[4] R.G.G. Catell, "*Object-Oriented Data Management: Object-Oriented and Extended Relational Database Systems*", Addison-Wesley Publishing, Inc., 1994.

[5] T.K Sellis., L. Shapiro, "*Optimization of Extended Database Query languages*". ACM-SIGMOD, Austin 1985.

[6] W. Kim, "*Global Optimization of Relational Queries: A First step*". In: Kim W., D.S.Reiner, D.S.Batory;"Query Processing in Data Base Systems".Springer Verlag,1985.

[7] M. Jarke, "*Range Nesting: a Fast Method to Evaluate Quantified Queries*".Int.Conf. ACM-SIGMOD, 1983.

[8] T. Bannon, S. Ford, P. Pazandak, C. Thompson, and D. Wells, "*Object Services and Consulting, Inc.*". Available: http://www.objs.com/x3h7/sql3.htm 30-Jun-1997.

[9] B. Finance and G. Gardarin, "*A rule-based optimizer with multiple search strategies.* IEEE Data and Knowledge Engineering", 13(2), 1994.

[10] K. Ono and G. Lohman., "*Measuring the complexity of join enumeration in query optimization.* In Proc. Int. Conf. on Very Large Data Bases", Brisbane, Australia, August 1990.

[11] B. Finance and G. Gardarin., "*A rule-based optimizer with multiple search strategies.* IEEE Data and Knowledge Engineering", 13(2), 1994.

[12] R. Breitel, D. Maier, A. Otis, J.Penney, B. Schuchardt, J. Stein, H. Wiliams, and M. Williams, "*The Gemstone data magement system. In Object Oriented Concepts, Databases and Applications*", eds. W. Kim and F. H. Lochovsky 1988.

[13] F. Bancilhon, S. Cluet, and C. Delobel, "*A query language for O2. In Building an Object-Oriented Database System-The Story of O2*", Morgan Haufmann Publishers, San Mateo, Ca., 1992.

[14] B.P. Jenq, D. Woelk, W. Kim, and W.-L. Lee, "*Query processing in distributed ORION.* In Proc. EDBT", Venice, Italy, 1990.

[15] A. Kemper, G. Moerkotte, and K. Peithner, "*A blackbord architecture for query optimization in object bases.* In Proc. Int. on Very Large Data Bases", Dublin, Ireland, August 1993.