IDENTIFICATION AND ANALYSIS OF INTEGRITY CONSTRAINTS OF OODBMS IN THE PERFORMANCE VIEW

Sahebrao N Shinde1*, Chitra G Desai2

1 Computer Science Department K.T.H.M. College, Nashik, Maharastra, India.
2 Department of MCA, MIT Aurangabad, Maharastra, India.

ARTICLE INFO

Corresponding Author:
Sahebrao N Shinde
Computer Science Department
K.T.H.M. College, Nashik, Maharastra, India.
sns110@gmail.com,
chitragdesai@gmail.com.

ABSTRACT

An Object Oriented Database Management System (OODBMS) is the combination of objects oriented language and database. In this paper OODBMS introduction as well as integrity constraints in OODBMS is given. Integrity constraints are identified and analyzed with the help of oracle 9i, 9i and 10g. We have identified and analyzed integrity constraints in the view of performance with comparison of ORDBMS.

©2011, JPRO, All Right Reserved.

KeyWords: OODBMS, chaining.

INTRODUCTION

An Object Oriented Database Management System is combination of object oriented programming with the database management system features [1]. The object oriented features are implemented using the virtual allocation of storage [1]. In addition to these features we need to represent the data handling features using DBMS and RDBMS.

An integrity constraint can define to enforce business rules on data in the database tables. Business rules specify conditions and relationships that must always be true or false based on the situation [3]. Business rules are varying from organization to organization and the policies of the organization. The rules are to implement at the time of design the database. For example employee salary, inventory tracking, trading system etc.

For an integrity constraint applies to a table, all data in the table must conform to the corresponding rule[4]. When you issue a SQL statement that modifies data in the table, Oracle ensures that the new data satisfies the integrity constraints, without the need to do any checking within your program. You can enforce rules by defining integrity constraints more reliably than by adding logic to the application. Oracle can check that all the data in a table follows an integrity constraint faster than an application procedure:

Create a separate table space with bigger block size i.e., of 4k block size and transfer all the tables of which average row length exceeds the block size into the table space of the bigger block size.

This will improve performance because instead of selecting row from multiple blocks, now it will select a row from a single block[6].

Row chaining takes place in every DBMS, it may be ORDBMS or OODBMS but row chaining can be removed in ORDBMS and not in OODBMS. But in case of OODBMS a row chaining cannot be removed.

2) No separate memory:

In case of Oracle database there is a separate memory called SGA (Shared or System Global Area). SGA is memory exclusively allocated to the Oracle Server, hence no other application data will come into SGA nor Oracle Data goes out of SGA. Due to the separate memory the data is not scattered throughout the complete RAM. Thus enables the data retrieval very fast. Whereas there is no special memory for OODBMS and thus objects data can be...
scattered anywhere in the RAM and the retrieval become slower.

In case of relational databases like oracle it has got a separate memory called SGA (Shared Global Area). It is called shared memory because the contents in this memory are shared by all the clients. It is divided into some of the following components[3].

1) DB buffer cache
2) Shared Pool
3) Redo log buffer

1) DB buffer cache

It contains the blocks read from the data file. If a query is issued against the table the server process will check whether the tables blocks are there in the buffer pool or db buffer cache or not. If they are found then there will be no physical read thus increases the performance. Less the physical reads more the performance, more the physical reads less the performance.

If a user requires a table’s data frequently then the table’s data can be kept in the buffer pool or cache. It will be there till the database is shutdown. If the tables datablocks are not cached then these blocks will be flushed if there is no enough space for the other requested table data blocks. This buffer pool or cache works on the LRU (Least Recently used). LRU blocks will be flushed. To make the table's data blocks MRU then cache the table. Table can be cached by giving the following command.

Alter table <table_name> Cache;

If you want to make the cached tables data blocks as LRU blocks (If you wants to flush the tables data blocks then give the option no cache).

E.g. alter table <table_name> Cache;

2) Shared Pool

This component is further divided into the following components.

a. SLC(Shared Library Cache)
b. DDC(Data Dictionary Cache)
a. SLC (Shared Library Cache)

It contains the execution plan, compiled SQL statement, compiled PL/SQL blocks i.e. procedures, packages, triggers, text format of the SQL statement. The process optimizer will pick up the eplan(execution plan) if it finds and gives it to server process for fetching the data. To increase the performance, the compiled or parsed SQL statement and the eplan must be found in this cache. If these plans and the parsed SQL statement is flushed then the performance will be degraded as the optimizer has to prepare the execution plan and also parsing has to take place. Then only the server process can fetch the data.

b. DDC (Data Dictionary Cache)

It contains the definition of the objects, data files, indexes, extends information etc. which will be used for the purpose of semantic analysis. To increate the performance this contents should be found as far as possible in the DDC.

3) Redo log buffer [Refer to Oracle architecture]

It contains the changes made to the database which will be written to the log files on the hard disk. If the database is in archive mode then the contents of the log file will be written in the form of archive files by separate process called ARC at the destination mentioned in the parameter files.

In Oracle the size of SGA can be increased and decreased thus utilizing the memory in an efficient manner which lacks in OODBMS.

No separate memory is allocated to objects in OODBMS to keep the objects in the OODBMS. In Oracle there is separate memory called SGA(Shared Global Area). Only Oracle’s data is stored and no data of any other application is stored in the SGA. The result of this retrieval and other database operations will be faster because it will not read any other applications data hence there will be less burden on server.

3) Lack of different processes:

Unlike Oracle databases there are no separate processes for reading, writing etc. For eg. Reading will be done by a server process, writing the changes from the db buffer cache will be done by database writer(dbwr) process, releasing the resources like memory will be done by the checkpoint process. Due to the different processes the load will be balanced ie., there will not be load on a single process, it results in better performance. Due to the lack of different processes in OODBMS, reading and writing will be done by single process. Thus these will slowdown the eading and writing operations[4].

4) Indexing is not available:

Index is an optional structure associated with the table which will speed up the execution of select statement as it stores the location of row which is called ROWID.

In case of Oracle RDBMS there are different types of indexes suitable for different types of applications. E.g. The types of indexes are B-Tree and Bitmap indexes. Bitmap indexes are used where cardinality is less and btree indexes are used where the cardinality is high. These indexes are not available in OODBMS. Indexing will speed up the database operation because index contains the rowid. Rowid is nothing but physical address of the row. Rowid contains the following:

Fileno. Blockno. Rowno

Types of indexes are 1)btree and 2)bitmap

Create index if the cardinality is very high.

To create btree index:

Create index <index_name> on <table_name> (column_name);

E.g. Assume if the searching is on the column empno of table emp.

SQL> create index emp_empno_ind on emp(empno, job);

To view the list of indexes on the table emp give the following command:

SQL> select index_name from user_indexes where table_name='emp';

Create the bitmap index where the cardinality is very low.

Syntax:

Create BITMAP index <index_name> on <table_name> (column_name);

E.g. Assume if you are searching the column result which contain the value pass

and fail on the table student.

SQL> create BITMAP index bit_result on student(result);

Creation of index is done by the developer but it is used internally by the database management system. It is never used by developer and the contents cannot be displayed in the index.

The index structure contains the column name with corresponding rowid, values and its corresponding rowid. Rowid consists of the field. Blocked row no To
increase the performance create the proper index on a proper column, otherwise performance will be degraded [5].

5) Clustering

It is an optional method of shoring tablespace. A cluster is a group of tables that share the same data blocks because they share common column and are often used together[6]. When you cluster the emp and dept both the emp and dept tables in the same data blocks. Following are the steps:

1) Create cluster
SQL>create cluster emp_dept(deptno number);
2) Associating tables
SQL>create table dept(deptno number primary key, dname varchar2(20)), emp(deptno) ;
SQL>create table emp_dept(deptno number, empno number, name varchar2(20)) cluster emp_dept(deptno);
3) Create the index on the cluster by giving the following command:
SQL>create index emp_dept_index on cluster emp_dept;
Clustering facility is not allowed in OODBMS. It has not provided indexing and clustering together.

6) Partitioning

Partitioning means splitting the large table into smaller tables. These smaller tables are called partitions[5]. With partitioning data can be read and written parallel, because each partition can be stored on different hard disks. That is the reason read and write operations can be perform parallel. One more advantage is if you are creating the index or rebuilding the indexes, you can create the index on the partition and not on the whole table, thus you can prevent generating huge of temporary data. If you generate huge amount of temporary data then there are possibilities that it may sort in the hard disk which slows down the performance. First oracle will sort the data in the memory and if the memory is not sufficient then it sorts in the hard disk. It means memory is not wasted by creating partition. Another advantage of partitioning is the backup can be taken of individual partition. If the data is lost of specific partition then you have to restore only that partition data. It means time is saved while taking the backup and while restoring. Hence we can say a large table’s data can be managed efficiently by creating partitions.

Types of partitions:

a. Range partitioning
b. Hash partitioning
c. Composite partitioning

a. Range partitioning
Partitioning is done on the basis of range.
E.g. If you want to share data from Jan-Mar on one hard disk, Apr-Jun on second hard disk, Jul-Sep on third hard disk and Oct-Dec on fourth hard disk.
Create table sales (custome_id number, sales_date date, amount number)
Partition by range(sales_date)
(Partition P1 values less than(’01-apr-2011’) tablespace ts1,
Partition P2 values less than (’01-jun-2011’) tablespace ts2,
Partition P3 values less than (’01-oct-2011’) tablespace ts3,
Partition P4 values less than (max value) table space ts4);
b. Hash partitioning
Oracle uses hash function to handle this type of partitioning.
E.g.
Create table sales(customer_id number, sales_date date, amount number)
Partition by hash(sales_date)
Partition by P1 tablespace ts1,
Partition P2 tablespace ts2,
Partition P3 tablespace ts3,
Partition P4 tablespace ts4);
c. Composite partitioning
It is a combination of range and hash partitioning.
E.g.
Create table sales(customer_id number, sales_date date, amount number)
Partition by hash(customer_id) sub partitions 4 store in(ts1,ts2,ts3,ts4)
(partition P1 values less than(’01-apr-2011’) tablespace ts1,
partition P2 values less than(’01-jun-2011’) tablespace ts2,
partition P3 values less than(’01-oct-2011’) tablespace ts3,
partition P4 values less than(maxvalue) tablespace ts4);
In case of OODBMS if the file is read the entire content is loaded in the memory. This will leading to paging and swapping and you cannot partition the objects data.

7) View

It is available in the relational databases. It is not a table but it is a virtual table. It contains only the select statement but not the data[4]. Data will be in the table. The advantages of view:

1) It will make complex query
If you are joining 2 tables and if it includes a subquery, if this query is required frequently then store the above query in the view.
If you want to execute the above query then write the query against the view.
E.g. If the query to to display deptno, empno, sal from table emp and dept, loc from table dept then the query will be
Select emp.deptno, empno, sal, dname, loc From emp, dept Where emp.deptno=dept.deptno;
The above query can be store in the view by giving the following command.
Create view emp_dept_view
As
Select emp.deptno, empno, sal, dname, loc From emp, dept
Where emp.deptno=dept.deptno;
2) It restricts data access
With view you can hide the access to the specific column. This is also called data abstraction or data hiding.
E.g. If you do not want to give access to the column sal then do not include the sal in the view.
Create view emp_dept_view_no_sal
As
Select emp.deptno, empno, dname, loc From emp, dept
Where emp.deptno=dept.deptno;
3) It presents multiple forms of the same data.
E.g. If you want to display deptwise sum of salaries then create a view as follows:
Create view dept_total_sal
As
Select deptno, sum(sal) total_sal
from emp
Group by deptno;
If you want to display deptwise, jobwise total salaries then create a view as follows:
Create view dept_job_total_sal
As
Select deptno, job, sum(sal) total_sal from emp
Group by deptno, job;
If you don't use the view concept then there is only one way to create a table but it occupies space hence it is not recommended. View will not occupy any space.
If you want to see what the dept_total_sal contains give the following query.
Select * from dept_total_sal;
4) Data independence
It means the user is unaware from which table data is coming and if the user drops the view the table is not affected.
View can be shared by multiple users but the object in the ORDBMS cannot be shared by multiple users. If the user U1 has created the view then other users U2 and U3 can access the view V1 if user U1 grants the select privilege on V1 by giving the following command.
SQL>grant select on dept_job_total_sal to U1,U2;
The advantages are lacking in ORDBMS.
8) No separate space on hard disk
Objects in ORDBMS are scattered throughout the hard disk because it doesn't have separate space of its own to store the respective objects[6]. This will result in slow retrieval of data because the data is scattered throughout the hard disk as a result it has to search every block.
Unlike in Oracle there is no concept of tablespace in ORDBMS. In Oracle relational database Oracle stores data in its own area called tablespace which contains physical datafiles.
With the concept of tablespace, Oracle data will be stored in the tablespace i.e., it will not be scattered throughout the hard disk nor no other application data will be stored in the tablespace. Tablespace means the logical storage of oracle data which consists of physical data files[5]. The result of the tablespace is oracle searches the data in its own area and other part of hard disk is not searched. As the tablespace stores only oracle data that is oracle data will not go outside the tablespace and it will not allow other applications data in the tablespace. With the tablespace concept performance will be improved[2].
The following is the example of creation of tablespace and storing the table in the corresponding tablespace.
Create tablespace userdatatbs
Datafile 'c:\pks\df\userdatatbs1.dbf' size 10m;
Associating / storing table table in the above created table space.
Create table emp(a number)
Table space user database;
From the above scenario we can say the ORDBMS lacks the control of storing the objects.
9) You can configure different block sizes in ORDBMS
In the relational database like Oracle there is provision of configuring different block sizes depends on the length of the row of table. The block sizes in oracle are : 2K, 4K, 8K, 16K and 32K[2].
Block size is responsible for row chaining and row chaining is one of the reason of poor performance. Row chaining takes place when the row size exceeds the block size. In this case oracle will split the row in multiple blocks. To remove the row chaining there is a provision which is configuring the different block size.
E.g. If row exceeds 2K block size then create a tablespace with 4K blocksize and move the table into the 4K tablespace.
1)Assume that the table was created with 2K blocksize which resulted in row chaining.
2)Create a tablespace with bigger block size i.e. 4K block size like given below.
Create tablespace userdatatbs4k
Datafile 'c:\ord\df\userdatatbs4k1.dbf' size 10m
Blocksize 4k;
3)Move the table to the tablespace with 4K blocksize by giving the following command.
Alter table table_name
Move tablesapce userdatatbs4k;
CONCLUSION
From the above features of ORDBMS we can say that there are many features in ORDBMS but still above features are lacking which results in poor performance as the burden will be on a single process and the complete hard disk is searched. We can say that the above features of ORDBMS are the limitations of ORDBMS.
REFERENCES
[1]. Won Kim "Introduction to Object-Oriented Databases", The MIT press 1990.