IDENTIFICATION AND ANALYSIS OF INTEGRITY CONSTRAINTS OF OODBMS IN THE AVAILABILITY AND CONSISTENCY 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.

ABSTRACT

An Object Oriented Database Management System is the combination of object oriented language and database together. This paper gives OODBMS introduction as well as integrity constraints. Integrity constraints are identified and analyzed with the help of oracle study using versions 8i, 9i and 10g. The constraints identified and analyzed in the view of availability and consistency. This analysis is done with examples.

©2011, JPRO, All Right Reserved.

TICLE INFO

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

KeyWords: OODBMS, chaining.

INTRODUCTION

An Object Oriented Database Management System is combined study of object oriented programming with the database management system features[3]. The object oriented features are implemented using the virtual allocation of storage. In addition to these features we need to represent the data handling features using DBMS.

An integrity constraint can define to enforce business rules on data in the database tables[1]. Business rules specify conditions and relationships that must always be true or false based on the situation. Business rules are varying according to the organization policies[2]. The rules are to implement at the time of design the database. For example employee payroll, inventory management, trading system etc.

When an integrity constraint applies to a table, all data in the table must conform to the corresponding rule[1]. When we 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 our program. We 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 can. As database engine in oracle resides the current database in cache.

Integrity constraints are applicable to minimize space and time overhead with proper rules defined in the definition of database tables. Integrity constraints are used to prevent the entry of invalid information into tables[2].

Constraints or limitations of OODBMS in the view of availability and consistency

The OODBMS is having very good features of database handling with the object oriented facilities. In addition to the same there are some constraints in the angle of availability and consistency are:

1) Backups are not possible: Unlike Oracle RDBMS there are no different types of backups like online, incremental.

Due to lack of proper backups data will not be available 24X7. There is no recovery mechanism hence there are chances of loss of data[4].

In case of Oracle there are different types of backups those are

1) Offline / Cold backup: This backup can be taken only if the database is in shutdown mode. Taking the backup is nothing but copying the files of corresponding database by issuing operating system command.
2) Online / Hot backup: This backup can be taken while the database is up and running. The command of taking is same as the offline backup. But before issuing copy command the table spaces must be kept in begin backup mode. After copying the corresponding files again the tablespace must be kept in end backup mode.
3) Logical backups: Logical backup can be taken with the oracle utility (if the database is Oracle) exp (export) and data can be restored if the database is crashed with imp (import).

In case of OODBMS due to lack of different types of backups there is always threat to the data loss. The only way to take the backup is Serialization which is time consuming.

As there are various types of backup, if there is any loss of data, it can be recovered quickly and there will not be any loss of data. You can archive the data to more than 1 destination which is helpful in the disaster recovery[5].

2) Lack of Flashback features: Flashback makes it possible to correct user errors that is if the user delete some data. In Oracle you can get even the dropped tables by going in a flashback mode. In case of OODBMS if you delete data it cannot be recovered. Again you have to enter the data.

Flashback is nothing but retrieving different version of the specific rows or data that existed during a certain time[6]. E.g. If the table is dropped then you can recover the dropped table by giving the following command...
Syntax
    Flashback table <table_name> to before drop;
Assume that the table products are dropped then it can be recovered by giving the following command.
    SQL> flashback table products to before drop;
If you have removed from array/collection in OODBMS is not possible to recover. With this feature there will be no data loss even if user commits errors. No data loss must be one of the feature of any database management system which ORDBMS is satisfying.

3) Logging: Logging is nothing but keeping track of all the changes made to the database in the redo log buffer. From the redo log buffer the changes are made permanent by writing the changes from the log buffer into the log files on the hard disk[5]. The advantage of logging is you can retrieve data of any time provided the database must be configured in the archive log and the changes made to the database will be in the archive files. If the user commits any errors that is delete or update wrong records and given commit statement still data can be restored to the original state[1]. This is possible because every statement has got two statements:
1)The original statement and
2)Opposite of the original statement
If the following statement is issued :
    Sql> delete from emp where deptno=10;
    Sql> commit;
If the database is running in archive log mode, then you can get from the archive files by using log miner, because it stores the original statement which is also called SQL REDO and the opposite statement which will be insert statement. You can get back all the deptno 10 deleted records by logmining activity[4].

4) Authorization :
    With proper privileges the data will be consistent and no threat to the data as the user cannot access the table without sufficient privileges.
    In most of the cases for handling database different users want different privileges. If the requirement is giving different privileges for different users like for some users you want to give select privilege, some users select, insert, some users insert, update, delete and for some users no privileges, some users all privileges[5]. This kind of complex accessing method lacks in OODBMS. It can be achieved easily in the relational database by giving privileges.
    E.g. To give select privilege to users U1 and U2 on table emp.

    SQL>grant select on emp to U1, U2;
    To give select and insert privileges to users U3 and U4 on table emp.
    SQL>grant select, insert on emp to U3, U4;
    To give insert, update, delete privileges to users U5 and U6 on table emp.
    SQL>grant insert, update, delete on emp to U5, U6;
    To give all privileges to users U7 and U8 on table emp.
    SQL>grant insert, update, delete on emp to U7, U8;
    All includes select, insert, update, delete.
    It is not only easy to grant accessing methods in the relational database (Oracle) but also we can revoke. In short in the relational database table can be accessed or denied can be done dynamically.
    Even you can restrict the user from giving a particular attribute like DROP, INSERT, DELETE...etc. With the above privileges you can get complete control over the users
    And keep the data consistent.

5) Auditing: With auditing we can put check on the user who has got the privilege to access the data, i.e., we can know which user has access which table along with time and data and also the privilege[6].
    If we are suspicious on a particular user then we can enable auditing. For e.g., if you are suspicious about user u1 then we can enable auditing on user u1 like If you want to the login time of any user then you can audit like following:
    SQL> audit create session;
    Even you can restrict a user from giving DROP command.
    Due to triggers you can make the auditing powerful because you can log the old values also and also the logout time like:
    Create or replace trigger logout
    Before logoff on schema
    Begin
    Insert into log_tab values(user,sysdate);
    End;
    / Create the table log_tab
    Create table log_tab(username varchar(20), logout_date date);
If any table is dropped by u1, Oracle will keep the history of date and time, username, hostname from where the table is dropped. With this we can say even after giving authorization towards the data which is crucial, you can put a check on the user, thus the data will be in a consistent mode.

CONCLUSION
    From the above constraints we can say that there are many features in OODBMS but still above constraints are not built in. These constraints can also implement in the OODBMS by which the software will be full fledged and applicable for object oriented as well as relational database management system.
    Due to different types of backup, due to archiving facility and flashback, logminer the data can be recovered as there will not be loss of data. As the function of any DBMS is managing data then data should be available 24 X 7. Due to authorization and Auditing data can be accessed by multiple users but you can put check on the user and by this data will be always consistent.

REFERENCES