January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

Preserving Privacy and Optimizing Performance by setting Multi-Level Access Restrictions using VPD Policies with Locks

H. Lakshmi[1], Dr. K. Nageswara Rao[2]
[1]Research Scholar, Rayalaseema University, Kurnool, Andhra Pradesh, India
[2]Principal, Potti Sriramulu Chalavadi Mallikarjuna Rao College of Engineering & Technology, Andhra Pradesh, India
itslakshmi.h@gmail.com[1], principal@pscmr.ac.in[2]

Article Info
Volume 82
Page Number: 7087 - 7098
Publication Issue:
January-February 2020

Article History
Article Received: *18 May 2019*
Revised: *14 July 2019*
Accepted: *22 December 2019*
Publication: *03 February 2020*

Abstract:
Several organizations use databases that are mostly secured against impostors. Organizations take many network security precautions such as firewalls, and build network-based systems that detect intrusion. Although network security systems are important in protecting databases in this regard, securing the database systems themselves, and the procedures/functions and data within them, has possibly become more critical as networks are increasingly opened to wider access through the Internet. In order to prevent unauthorized access of data, several programs and procedures have been put in place for users to provide authentication before accessing a data. In this paper, we primarily focused on discussing a multi-level access control mechanism that uses a Virtual private database. Database returns customized results to each user using Virtual Private Database. We tried to overcome some of drawbacks of VPD such as mechanisms to project out certain attributes i.e., column level security and difficulties in writing predicates involving cases of cross-reference, joins of tables. This Research Paper also talk about preventing data loss for advanced database management systems, focuses issues such as policies in database as well as the essential capabilities: Protect, Monitor, Discover and Manage of the policies as it protect the privacy of the data and shouldn't disrupt the lawful activities of any organization.

Keywords: *Data Privacy, Column Level Security, Multi-Level Access Restrictions*

I INTRODUCTION

Usually, a database application must allow each department access to only its own data: data entered by Department ABC should be visible only to Department ABC. This is a typical situation for a virtual private database. One single server may serve all the sectors/departments at various locations, and data for a specific entity from various locations resides in one relation. Every individual employee of distinct departments at various locations can see only his/her own data. This paper illustrates this concept and demonstrates how to enable mutli-level access restrictions which ensures performance, accuracy and security.

1.1 Database Security

The subject of Information security is vast and Database security is one of the specializations within this subject and requires special focus. Primarily, three objectives should be considered in order to design a secure database application to safeguard the information. They are:

1. Secrecy
2. Integrity

January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

3. Availability

Secrecy:

Limiting access to information and unauthorized user access is what Secrecy or Confidentiality refers to. Secrecy addresses certain aspects of security that have subtle differences. The first aspect is about preventing unauthorized individuals from identifying and accessing the confidential information. The second aspect is protecting confidential information and disclosing it only to those individuals who are authorized. This can be done by classifying information based on the violation of two factors—a breach of confidentiality and a risk for the security of information. By properly classifying information and having a design process in place can help in implementing and enforcing confidentiality. A simple example is to ensure that a Doctor from a certain department is not allowed access to the patient details of another department.

Integrity: Sometime, employees or other users may modify or change certain information accidentally. Such actions can lead to questioning the integrity of the data that loses consistency after being modified. It is important to address the integrity aspect of information security because it focuses on the most treasured asset, data, which eventually becomes information. Any Data that has not been tampered with intentionally or inadvertently can be considered accurate and has integrity. For achieving complete integrity, data must be protected at all levels. For example, a few researchers may be allowed to view patient details, but may not be provided rights (obviously!) to modify them.

Availability: Any user, who is authorized to access certain data, should be able to view it, update it, or perform any other actions that he is authorized to. This is called Availability with respect to database security. It is the system that should be able to control what an individual can do with that data she has access to. For example, if a concerned Doctor is authorized to change the patient details, then he should be allowed to do so.

1.2 About the Paper

The paper is planned in such a way that second segment explains various Access Controls such as DAC, MAC, RBAC, Content-based and Context-based. Segment 3 states the benefits of VPD while comparing with views, Purpose of Application Context and Security Policies. Segment 4 summarizes the implementation of VPD policies with locks, Multi level secure system, locking the relations, granting and revoking attributes. Lastly, Segment 5 discusses Conclusion and Limitations.

II SECURITY MECHANISMS

Enabling particular entity to either allow/disallow to user referred as Access control [2] policy. Today, all the permissions to the users are happening through roles only which allows user to manipulate the data in the database objects. A security level is assigned to each object. A user can access or manipulate data if he/she owns the permission to manipulate on that specific database object.

1) Discretionary Access Control

At times, users can request access to certain information. Providing access to such users on request is considered as discretionary access control [3]. To get the access user should own the permission on that database object. This owning the permission on specific objects is referred as authorization. The set of rules for authorization tells the users, how much data he/she can access and how many database objects he/she can manipulate. **Example:** Consider S, subject, O, Object, and A, Action. Subject is nothing but users; Objects are created and maintained by the subjects like relation/synonym/view and Action means execute/write.

Table 1 is a sample access matrix, tells that student can only read the RESULTS. But FACULTY can update the RESULTS.

January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

Table 1
Privileges associated with each object

SUBJECT	RESULTS	WORKING HOURS	SALARIES
Management	Read	Read	Write
HOD	Read	Write	Read
Faculty	Write	Read	Read
Student	Read		

For organizations that are operate on medium or large scale with a huge number of clients or users, access control possibly will not be very flexible. Some of the issues with this mechanism are: If one employee leaves the company/Industry, the same set of security levels to be assigned to the new employee. The access control mechanisms will work and allow the new employee to access the database objects and treats as same as old one. These controls cannot identify the user as hacker/intruder. This is why because he/she owns the roles and permissions on the database objects. Now days it became a critical issue for any company who are working with live and confidential data.

2) Mandatory Access Control

To avoid the issues raised in Discretionary Access Control there is a need of central authorization. A central authority puts in place regulations to enforce the access controls. These are referred as Mandatory Security Policies. Here all the employees are classified into different levels based on their permission/access levels on objects.

Level 0 employees/users can access only few tuples of database objects on which they are allowed to access. Level 1 employee can access level 1 tuples and as well as level 0 tuples of database objects. It means, level 1 employee has high security access level authority and can access more tuples when compared with level 0 employee. In this way, subjects and objects are assigned to a particular access level. Now, it is easy to tag on employee's operations. This access control increases the work for administrative department. Admin role is very crucial because he has to divide and assign the roles to each access level subject and object.

3) RBAC- Role Based Access Control

Role means a command or privilege on particular object. Usually these roles are associated with tables, views or any other database objects. To restrict the tasks of users up to that role, RBAC [4] is taken into consideration as a security aspect. If the object is a procedure then the role will be execute command on that procedure. Now this role will be granted to a subject. The user who owns the role can execute that procedure until it is revoked by the owner of the procedure.

There are two steps to define the access control policy: First, the permissions associated with each and every user is defined by the administrator. Next, those commands will be sanctioned to users by the DBA. A hierarchy must be followed in the assignment of roles to the users. One user may get any number of roles and one role may be granted to any number of users. Users holding the same set of roles are defined as one group of users. Additionally, users may activate or deactivate roles at any time. One point to be noted here is deactivating the membership in a group is not possible. RBAC also suits to commercial and close environments. This is an added advantage for RBAC over DAC and MAC. In a company or a close environment, a person's responsibilities are more important over his identity to access a system.

Example: Consider any Education institution. Generally university has many faculty members. Faculty members offer different courses to students.

```
SQL> CREATE ROLE COURSE;
Role created.
SQL> GRANT INSERT, UPDATE ON WARD
  2      TO COURSE;
Grant succeeded.
SQL> GRANT COURSE TO USER;
Grant succeeded.
```

Here, COURSE is a role created by the

administrator and it is a set of various DML commands like INSERT, UPDATE privileges on WARD relation. COURSE role is granted to the entire teaching faculty members in the University. So that faculty may insert and update the attendance and marks of the wards in the appropriate database objects.

4) Content – Based Access Control

In order to implement an access control mechanism there are certain requirements should be satisfied by any data management system. One of the primary requirements is content-based access control. An access control mechanism that is content based ensures that any decisions pertaining to the content was made depending on what the data is. For example, let's consider a data table that has information related to all employees working in an organization. Let's assume that the content-based access control policy states that "The IT manager can access employee information who are in the IT department." If the IT manager issues a query, then the system filters the results and displays only those items related to employees working in the IT department and report to the IT manager. SQL language supports in implementation of such access control mechanisms. SQL is primarily used for data management because it has queries that are based on conditions that are applied against the data systems.

To implement content-based access control mechanism, relations DBMSs use views. For this, views are categorized as protection views and shorthand views. Protection views help in controlling access based on content while shorthand views in simplifying query writing.

A view enables the selection of a subsets of columns and rows. Using a view definition query, these subsets can be specified. The query is usually query connected to the view's name. View composition is a mechanism by which a query can be modified whenever it is issued against a view. In this mechanism, the view that is referenced in the query will be replaced by its definition. For example, if there is a "where clause" in the query then it will be combined with the "where clause" of the view definition query through the through the AND Boolean connective. This way, the query filters out the tuples that are not in accordance with the view.

With this approach, several benefits can be accrued. Access policies that are content-based are articulated such that they are in consistency with the query language. If a certain data is modified, then it may not be necessary to modify the control policies too. Employees can modify their data without carrying worry about the access control policies. Consider a typical hospital database:

Example:

SQL> CREATE OR REPLACE VIEW DOC_PAT
2 AS
3 SELECT PAT_ID, PAT_NAME, HIV_STATUS
4 FROM PAT_HIV_INS
5 WHERE DOC_ID = 'USER';
View created.

Here DOC_PAT is a view. USER may be any one such as DOC1, DOC2 or RESERACHER. If DOC1 get the read access permission on this DOC_PAT view then he/she can be limited to retrieve the details o patients under him/her. If any new patient is admitted in the hospital and updated the base table PAT then this view will be automatically modified. The doctor who is authorized to access the newly admitted patient can only view the details. By accessing this view DOC_PAT every doctor can only access the patients under him.

5) Context – Based Access Control

The validity of an authorization will intact unless a revoke operation is performed to remove the authorization. There are, however, systems in which the authorizations will remain valid for a limited time frame. Sometimes, periodic authorizations are applied for certain other systems. Some organizations will customize authorizations based on the form of operations carried out in the

January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

organization. That is the reason employees in these organizations will have access authorizations only for a set time frame. This works in a way similar to the "need-to-know" security principle. For example, "all employees can access their files every working day except on Saturdays and Sundays." Most DBMSs will implement such policies as a code in application programs. This complicates the approach making it difficult to verify and modify the access control policies. A proposed authorization model addresses such complexities by having a temporal interval of authorization validity; this indicates that an authorization is valid only during this specified interval. After the time interval, the authorization is removed automatically. This means that no revoke operations will be performed by the security administrator. The interval linked to an authorization may also have set time frames. The proposed model provides deductive temporal rules that support the automatic derivation of new authorizations depending upon the existence or non-existence of other authorizations during the specified time periods.

The new model is beneficial by providing flexibility to a large extent. It also ensures that several protection requirements are met. Traditional access controls have several limitations with respect to the number of requirements being met. Now, there is a need to include the time as a factor in access controls. Time and location will play a major role in deciding the permissions for duration and sites. These two factors will be incorporated in context-based access control model. This model is recently started implementing in various database management systems.

Example: Consider an example which creates a view to grant the permission on mentioned dates:

SQL> CREATE VIEW DOC_VIEW
2 AS
3 SELECT PAT_ID, HIV_STATUS
4 FROM PAT
5 WHERE TO_CHAR (SYSDATE,'D') IN (1, 2, 3, 4, 5);

View created.

This view specifies that all the doctors can access their patient files every working day except on Saturdays and Sundays.

III VPD

Virtual Private Database, VPD ensures that security is enforced on views, tables, or synonyms. The security policies are applied on these database objects directly and therefore the security is robust and cannot be bypassed whenever a user accesses the data.

If a user accesses a database object that is protected using a Virtual Private Database policy, then the SQL statement of the user will automatically get modified. A WHERE condition is created by a function as mentioned in the policy. Based on the values returned by the function and user's requirement VPD modifies the statement dynamically. Users can apply VPD policies to the DML statements and as well as INDEX statements. VPD makes administrator tasks easier because Virtual Private Database always works based on session attributes using 2 unique characteristics known as FGAC and Application Context. Fine – grained access control is always works along with security policies to database objects such as tables, views, procedures or triggers and Application Context usually identifies and accesses the session columns.

a) Benefits of VPD

Several benefits can be accrued by attaching VPD security policies to database objects such as views, synonyms, and database tables.

Simple: Let's imagine there is a relation called SAMPLE on which hundreds of views are created. If DBA wants to restrict the access to some of the fields of SAMPLE relation then he has to change the view definitions undoubtedly. This is very time consuming process and not suggestible. If a policy is attached to this SAMPLE relation then DBA can easily restrict the data of specific fields from the users.

January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

Scalable: To create personal databases, views are also considered as a solution. For example, consider HIV_STATUS relation which contains 1 million patient records all over India. To create a personal database for every patient there is a need of creating 1 million views which is not correct. This is the benefit of VPD because this personal database can be easily generated with a single policy.

Protection: If a policy is attached to a relation, synonym, procedure, or view then it is highly secured from all types of threats. Authorized users also intentionally or unknowingly can't insert wrong data or modify existing data in the relations.

Fine grained access control always secures the data from all types of threats and breaches. There will be no chance of data loss by attaching a policy to the databases. However a user tries to modify the data in the database, security policies will work in force and preserves the privacy of customer's data.

b) Application context

Today, organizations want to secure multiple attributes from unauthorized access. That's why they greatly prefer application contexts to preserve multiple security attributes.

With the help of application context retrieving multiple attributes is very fast and performance will also be increased if the data is available. Retrieving host name, IP address of the system, user name of the system is possible with the help of SYS_CONTEXT function. USERENV attribute is used to get these details.

Example:

```
SQL> SELECT SYS_CONTEXT ('USERENV',
        'IP_ADDRESS') FROM DUAL
  2    /
SYS_CONTEXT ('USERENV','IP_ADDRESS')
--------------------------------------------------------------
-----
172.148.2.30
```

When the application context data is running in the system then the total responsibility will be taken by RDBMS like oracle or SQL SERVER. The cache memory which is filled with application context data values will be cleared automatically once the user exits the session. If abnormal terminations happen due to internet failure or power failure then the user connection with the RDBMS will be lost. But the background process will cleans up the application context data. User has no need to take the responsibility of cleaning the context data from cache memory.

There are many purposes to use the application context:

i. Over multi user systems user privacy will be preserved
ii. Security will be ensured by fine-grained access control
iii. Performance increases at a commendable rate. Because when user repeatedly retrieves the same tuples by placing loops then the cache saves the time by saving the data in the memory itself and no need to access the attributes every time.
iv. Data loss will be reduced
v. Usage of application context is always recommendable, because data is not in control of user instead procedure will handle the security issues
vi. The application performance levels will get increased automatically due to no force is applied repeatedly to applications to retrieve the data from the relation.

c) VPD Security Policies

A collection of security policies associated with the same application forms a policy group. To indicate the effect of policy group user can a policy context. Database always checks the application context and confirms the permissions to allow the user to access the database objects. VPD security policies are very helpful in various situations like various applications access the same relation or view.

Example: Consider a relation INCENTIVE owned by an organization and shared among two different

January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

organizations. These two organizations access the relation by 2 distinct applications. These two applications use two different security policies and are owned by two departments marketing and Sales. Marketing department policy rules says that depending on ranking they give the authorization to the users. Sales department policy rules convey that depending on sales they give the authorization to the users. There is a need to join these two policies of 2 organizations to get access from the INCENTIVE relation. But it won't works in a proper way and not an accurate solution. In this situation, application context enforces the set of security policies to the database objects and distinct set of security policies will be implemented by two departments.

d) Detouring the limitations

In some situations, users want to terminate or revoke the constraints or limitations on the database objects so that owners of the relation can access all the attributes. The solution to attain this is administrator need to pass a NULL value. This NULL value will be returned by the predicate which is appended to the statement. Once the function returns NULL then it means the policy allows all the tuples without any restriction.

To access the tuples of the database objects without any restrictions the policy function will look like this:

```
SQL> CREATE FUNCTION GET_DISEASE_ID
(R_SCHEMA CHAR, R_REL CHAR) RETURN
CHAR IS
Q_DISEASE_ID NUMBER;
BEGIN
        IF (R_SCHEMA = USER) THEN
                RETURN NULL;
        END IF;
        SELECT DISEASE_ID
        INTO Q_DISEASE_ID
        FROM DISEASE
        WHERE USER_NAME = USER;
        RETURN 'DISEASE_ID =
'||Q_DISEASE_ID;
END;
```

Function created.

The code which is written in bold is the actual logic which returns the null value. If the schema owner is the current user who is trying to access the details of disease then the security policy function returns null which means that no predicate is added to the statement. This is just like DBA role. If DBA logins and tries to access the tuples of specific database object then no restrictions will be there and no data will be hidden. This function will works as same now. The user will act as a DBA here, but it is always important to keep in mind that sanctioning such privileges on the database objects to the users may cause threats in future.

e) Working of VPD

Let's know the working of Virtual Private Database in detail: After adding VPD policies to the RDBMS if a user tries to access a relation or view:

i. The policy function is invoked by the RDBMS server.
ii. Depending on the session attributes mentioned in the query the policy function will return a predicate.
iii. Now the query will be rewritten by adding the WHERE clause and submitted again to the RDBMS server.
iv. The modified statement will be executed by the server and appropriate results will be displayed.

"Dynamic Query Modification" is the scheme on which Fine-grained access control will work and ensures the security of the database objects which are associated with the policies.

Query may be of any statement such as insert, delete or update. It may not be limited to select. Predicate is nothing but a WHERE condition returned by a function which implements the VPD security policy. Any database object which is linked with a policy will return a predicate and appends dynamically to the query submitted by the user. Before getting submitted to the RDBMS all the queries are executed and modified based on the security policies defined by the DBA.

January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

Figure 1 shows the architecture of the implementation and explains the functionality of the modules.

i. *Query Modification Module:* The query submitted by the user will be rewritten by this module in such a way that only permitted tuples are displayed as a result. Adding a specific WHERE condition is the miracle of this module. This modified query subtracts the tuples from the original query which is actually submitted by the user.

ii. *Policy Engine Module:* The policy will get executed by this module and it returns the predicate to the query modification module which dynamically modifies the query.

iii. *RDBMS Access:* Once the query is modified then it is sent to RDBMS. Now RDBMS accesses its database and returns the data set which includes only allowed database objects directly to the user.

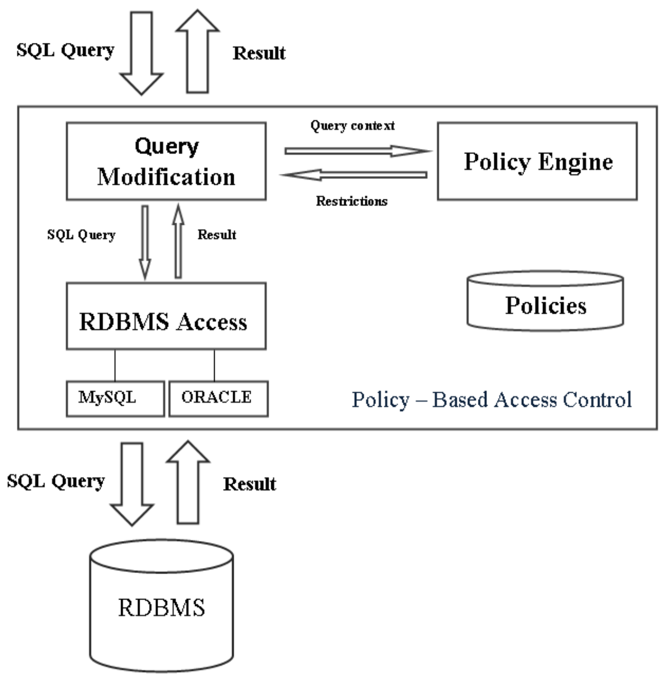

Fig.1. Architecture of Implementation

f) Oracle Policy Manager

Users prefer using interface tools that are easy to understand. Oracle provides a tool called Policy Manager, a GUI administration tool. Oracle uses Policy manager to manage its Virtual Private Database (VPD) policies and application contexts. Policy manager can also be used for managing policies for the Oracle9*i* Label Security. If DBA is working for large implementations on security policies associated with various relations, views or synonyms then Policy Manager is very helpful to him. It is also important to note that this policy manager cannot be a replacement for coding segment involved in VPD. On the other hand, this tool definitely reduces the burden concerned with maintaining the security policies and application contexts. It really creates a deep curiosity and attention towards Virtual Private Database for administrator.

Example:

The following function permits users to access the tuples of patients who has HIV negative i.e., they can't access the details of the patients whose status HIV POSITIVE. In this way it limits the access of users to specific tuples only.

Policy Function which returns predicate to the policy:

```
SQL> CREATE FUNCTION HIV_PATIENT
(R_SCHEMA IN CHAR, R_OBJECT IN CHAR)
RETURN CHAR
AS
BEGIN
RETURN  ' STATUS_NEGATIVE! =0 ';
END;
```

Syntax for adding a Policy:

```
SQL> BEGIN
  2    DBMS_RLS.ADD_POLICY
  3    ("Object schema name",
  4      "Object name",
  5      "Policy name",
  6      "Function schema name",
  7      "Function name",
  8      "Statement type");
  9     END;
```

Fig 2 is the Oracle policy manager using which administrator can easily enable and disable the security policies. By using Oracle Policy Manager Interface user can:

- Policies can be created
- Valid labels will be specified
- Relations will be associated with policies
- Users can get authorization or Permission
- Authorize trusted program units
- Configure auditing

Oracle policy Manager performs the following tasks:

- Policy Group will be created
- Policy Group can be dropped
- A Policy can be added to a Group
- A Policy can be dropped from a Group

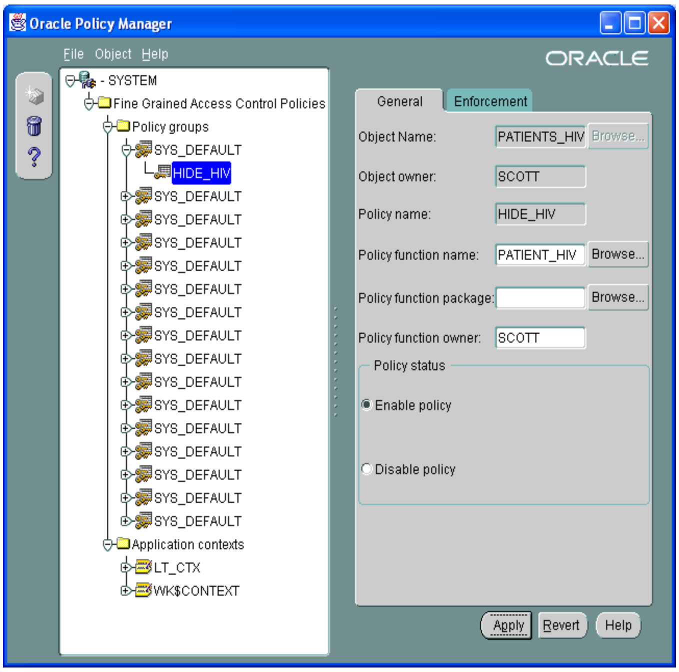

Fig.2. Oracle Policy Manager

IV OPTIMIIZING PERFORMANCE BY SETTING MULTI-LEVEL ACCESS RESTRICTIONS USING VPD POLICIES WITH LOCKS

Preserving privacy, protecting confidentiality, maintaining integrity, attaining loyalty, providing reliability, restricting unlawful activities, accomplishing regulations, achieving trust and optimizing performance are the key issues and concerns for any RDBMS. Preventing data loss and in detail restricting the unauthorized access are the primary tasks of administrators. For this reason, since many years several database security communities have implemented various techniques to ensure confidentiality and maintain integrity and make availability. Existing solutions to the database security issues include cryptographic techniques, authorization concepts, and risk analysis[16-21].

Views are the basic security mechanisms to protect the data privacy through applications in various organizations. Because views can limit access to the information associated with the base relations by content or context. With the support of views granular access control can be achieved fairly. But they have some limitations which make them less than optimal for very fine-grained access control. To enforce the user policies views are not practical due to their limitations. If user updates the views then the security policy will fall into risk. Updatable views always create complications for administrator.

This research paper mentioned several features of the Virtual Private Database, allows fine-grained access control down to the tuple level based on the use of predicates. The VPD disallows various sensitive data to abscond the security boundaries of organizations. This VPD is capable in providing both row-level and column-level security: access control, and authorization policies in RDBMSs. Implementing Locks with VPD policies are particularly suitable for organizations like: Online banking, Health care, Insurance, etc.

VPD policies with locks mechanism is very much helpful in the organizations where preserving the data privacy is of paramount importance or mandatory as per the privacy legislations like United States' HIPAA, CISP, Gramm-Leach-Bliley Act, Sarbanes-Oxley Act.

Multilevel Secure System: This system provides security in levels based on the authorization. It is a security policy which follows some hierarchy. The two primary goals of multi-level secure system are:

i. At first it provides some controls which prevent users from accessing the data at a higher level compared to their levels of authorization.
ii. Secondly it provides some controls which stops the individuals from declassifying the data.

Example: Suppose imagine there are 2 levels of security policy mechanisms. Those levels are labeled as “top secret” and “secret”. A user holding the security policy level as “secret” cannot write the data at the security policy level “top secret”.

For Internet access, the Virtual Private Database can ensure that online banking customers can see only their own accounts. It is possible to maintain the same database to store the patients private data sets of multiple branches, while permitting each doctor to see only his/her patient’s data. Security can be built once, in the data server, rather than in each application that accesses data. Security is stronger, because it is enforced by the database, no matter how a user accesses the data.

Large scale industries, National and Multi-national software companies typically work with millions of clients and thousands of employees at various locations. VPD policies sometimes may create issues because if one employee leaves the company/Industry, the same set of security levels to be assigned to the new employee. Imagine this new employee is a hacker/intruder. But if once the company recruits the new employee, immediately assigns the roles, security policies to him/her and he/she gains the permissions on the database objects. Now days it became a critical issue for any company who are working with live and confidential data.

Column Masking of Name and Designation of Patients:

PATIENT ID	DISEASE	NAME	DESIGNATION	DIAGNOSIS_NAME	STAFF ID	MEDICINE
PATIENT1	FLU			LFT	NURSE3	M-CIN
PATIENT2	TYPHOID			PFT	NURSE2	MONTECK

Locking relations if the existing user is replaced by new one:

```
SQL> LOCK TABLE CUSTOMER
  2      EXCLUSIVE MODE;
table(s) locked.
```

Revoking privileges if the old employee is replaced by new one:

```
SQL>  REVOKE UPDATE, INSERT
  2      ON CUSTOMER
  3      FROM NEW_EMP;
Grant succeeded.
```

Granting specific columns if the existing employee is replaced by new one:

```
SQL>    GRANT    SELECT(PH_NUM), INSERT(ENAME)
  2      ON CUSTOMER
  3      TO NEW_EMP;
Grant succeeded.
```

V CONCLUSION AND LIMITATIONS

One of the most important technologies that allow the usage of mission-critical systems for online users is the Virtual Private Database. Cloud computing usually uses fine-grained access control schemes. These schemes combined with application contexts that are secure will allow companies to store data in the server and also safeguard them. This is to ensure that a uniform access control policy is enforced to all users irrespective of how they gain access to certain data. With the help of the Virtual Private Database, users will have access to their own data while others will not be able to view it.

The customer records will be safely segregated by the telecommunications firms and all the applications runs to access these records must follow the complex rules designed by the VPD. By setting Multi-level access restrictions using Locks and VPD policies helps to implement their RDBMSs with lower cost. It is just like one time investment policy for securing their data and protection from data loss. These locks and policies are directly placed in the data server, instead of implementing access control

January-February 2020
ISSN: 0193-4120 Page No. 7087 - 7098

mechanisms in every application that accesses the data. It thus curbs the "application security problem". Finally, this VPD with locks solution complements the most common application models to achieve secure, scalable, and simple RDBMSs by solving their Row-level, Multi-level security issues as well as high performance rate is automatically achieved.

VI REFERENCES

[1] Simon Liu and Rick Kuhn, "Data Loss Prevention", Published by the IEEE Computer Society ©2010 IEEE

[2] *Meg Coffin Murray*, "Database Security: What Students Need to Know", Journal of Information Technology Education Volume 9, 2010.

[3] Sohail IMRAN, "Security Issues in Databases", 2009 Second International Conference on Future Information Technology and Management Engineering

[4] Ravi Sandhu, David Ferraiolo and Richard Kuhn,"The NIST Model for Role - Based Access Control: Towards a Unified Standard."

[5] Xiaolei Qian, Computer Science Laboratory, SRI International "View - Bases Access Control with High Assurance."

[6] Ravi S. Sandhu and Sushil Jajodia, " Data and Database Security and Controls", Handbook of Information Security Management, Auerbach Publishers, 1993

[7] The virtual private database in oracle9ir2: An oracle technical white paper. http://www.cgisecurity.com/database/oracle/pdf/VPD9ir2twp.pdf

[8] E.Bertino, L.M. Haas, B.G.Lindsay, View Management In Distributed Data Base Systems

[9] Surajit Chaudhuri, Raghav Kaushik, Ravi Ramamurthy, Microsoft Research, "Database Access Control & Privacy: Is There a Common Ground?"

[10] Lakshmi, B., et al. "Data Confidentiality and Loss Prevention using Virtual Private Database." *International Journal on Computer Science and Engineering* 5.3 (2013): 143.

[11] Yadav, Rajesh and Sharma, Anand, A Critical Review of Data Security in Cloud Computing Infrastructure (January 14, 2019). International Journal of Advanced Studies of Scientific Research, Volume 3, Issue 9, 2018. Available at SSRN: https://ssrn.com/abstract=3315422

[12] Sudhakar, Kumar, S. An emerging threat Fileless malware: a survey and research challenges. *Cybersecur* **3,** 1 (2020) doi:10.1186/s42400-019-0043-x

[13] Tung Bui, Eric Clemons, Introduction to Information Security and Privacy in Business and Society Minitrack, Proceedings of the 52nd Hawaii International Conference on System Sciences | 2019,

[14] Pawitar Dulari, Brijender Bhushan, "A Novel Approach for Cloud Data Security Enhancement through Cryptography and Biometricin the Government Cloud Environment,International Journal of Computer Science and Mobile Computing, Vol.8 Issue.12, December- 2019, pg. 59-63.

[15] Anil Lamba, Satinderjeet Singh, Balvinder Singh, Sivakumar Sai Rela Muni, A STUDY PAPER ON SECURITY RELATED ISSUE BEFORE ADOPTINGCLOUD COMPUTING SERVICE MODEL, International Journal For Technological Research In Engineering Volume 3, Issue 4, December-2015ISSN (Online): 2347 -4718.

[16] Sari, S.P., Dwidiyanti, M., Wijayanti, D.Y. and Sarjana, W., 2017. Prevalence, demographic, clinical features and its association of comorbid depressive symptoms in patients with schizophrenia. International Journal of Psychosocial Rehabilitation, 21(2).

[17] Fuadi, L., 2017. Influence of Behavioral Counseling Techniques, Token Economy and Parent's Parenting Class of Behaviour Prosocial X Syamsulhude Tegallinggah. International Journal of Psychosocial Rehabilitation, 21(2).

[18] Uhrmann, L.S., Nordli, H., Fekete, O.R. and Bonsaksen, T., 2017. Perceptions of a Norwegian clubhouse among its members: A psychometric evaluation of a user satisfaction tool. International Journal of Psychosocial Rehabilitation, 21(2).

[19] Kurian, J., Christoday, R.J. and Uvais, N.A., 2018. Psychosocial factors associated with repeated hospitalisation in men with alcohol dependence: A hospital based cross sectional study. International Journal of Psychosocial Rehabilitation. Vol 22 (2) 84, 92.

[20] Melnichuk, M., 2018. Psychosocial Adaptation of International Students: Advanced Screening. International Journal of Psychosocial Rehabilitation. Vol 22 (1) 101, 113.

[21] Daly, A., Arnavut, F., Bohorun, D., Daly, A., Arnavut, F. and Bohorun, D., The Step-Down Challenge. International Journal of Psychosocial Rehabilitation, Vol 22(1) 76, 83.

AUTHORS' BIOGRAPHY

Mrs. H.Lakshmi is currently working as an Asst. Professor, Department of Computer Applications, VRSEC (Autonomous), Vijayawada, Andhra Pradesh. She has 13 years of teaching experience. Her areas of interest include Database Security, Database Management Systems, Data Warehousing and Data Mining. She had received M.C.A from Acharya Nagarjuna University and M.Tech in CSE from JNTUK, Kakinada. She has ratified under Acharya Nagarjuna University and JNTUK, Kakinada. She had completed the OCA certification. She has achieved top 5 in Database Management Systems course, Funded by the MHRD, India. She is a Member of CSI. She is a valued reviewer for IJERT during the year of 2019.

Dr. K. Nageswara Rao Garu is currently working as Principal, Potti Sriramulu Chalavadi Mallikharjuna Rao College of Engineering & Technology, Vijayawada-7. He had completed his Ph.D. in computer science and system engineering and former professor of Andhra University, Vishakhapatnam. He has an excellent academic and research experience. He has contributed a number of research papers in various National and International Journals. His area of interest includes Robotics, Data warehousing and Data Mining and Database Security.