



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Course: Network and Information Security

Course Code- ITE4001

Slot - F2+TF2

Faculty- Aswani Kumar Cherukuri

Review 3

1. Title of the Project: Database Security For Health ID

2. Name of the Students: Nikhil Khurana

Chitresh Gupta

Vanshika Singh

Kritika Garg

3. Registration Numbers:18BIT0092

18BIT0109

18BIT0110

18BIT0427

4. Problem Definition:

Precise problem statement reflecting your project.

With coronavirus disease (Covid-19) bringing health sector into focus, the Prime Minister Narendra Modi announced that the National Digital Health Mission will commence on the Independence Day. "From today, the national digital health mission will begin. It will revolutionise Indian healthcare sector. Every Indian will be issued a Health ID that will act like a healthcare account, storing details of all the tests done, existing diseases, diagnoses, medicines prescribed," said the prime minister in his speech. This database would be extremely confidential and any breach of this data will lead to catastrophic consequences. Database security refers to the various measures organizations take to ensure their databases are protected from internal and external threats. Database security includes protecting the database itself, the data it contains, its database management system, and the various applications that access it. Security concerns for internet-based attacks are some of the most persistent challenges to database security. Hackers devise new ways to infiltrate databases and steal data almost daily. Organizations must ensure their database security measures are strong enough to withstand these attacks.

5. List of Objectives of to be achieved:

[3 Marks]

A list of objectives (min of 3 & max of 5) indicating what are to be achieved in the project.

Preferably with qualitative statements.

- Protection of Data: -The goal of database security is the protection of data against threats such as accidental or intentional loss, destruction or misuse. These threats pose problems to the database integrity and access.
- Authorization: - Authorization is the process of verifying which people of an organization can access which data. Authorization is a process managed by the DB2 Database manager. The manager obtains information about the current authenticated user, that indicates which database operation the user can perform or access. Eg: - Manager can access information of all the patients.
- Authentication: - Authentication is the process of confirming that a user logs in only in accordance with the rights to perform the activities he is authorized to perform. User authentication can be performed at operating system level or database level itself. Eg: - only a user or a member of organization can login.

6. Survey of the Literature:

[4 Marks]

A detailed survey of the literature indicating min of 15 articles (from reputed international journals). It should be written on your own. It is better if you arrange the survey in the form of chronological order & in a table as well. Indicate the research gap also.

S.no	Paper Title	Survey/Research gap
1	Database Security -Threats & Prevention	<p>Making sure about databases is a significant worry for the CIOs. To make preparations for these assaults, despite the fact that database data is powerless against an enormous number of assaults, it is conceivable to significantly diminish hazard by concentrating on the most basic dangers. In managing dangers, organizations should meet the consistence and hazard impediment prerequisites of the most profoundly controlled worldwide enterprises. They are likewise required to approach experts (through reviews) to check their IT security rehearses. It ought to be noticed that dismissing security is extremely regularly emotional. Without a doubt, the ubiquity of IT in organizations suggests that all touchy data is contained in databases or possibly in a worker or PC associated with the network and in this way they are conceivably piratables. markers will in general demonstrate that sooner rather than later assaults endeavors will be increasingly normal and in this way, we should recollect the significance of doing, notwithstanding all specialized measures, anticipation to clients, particularly those brought to deal with sensitive information. In the event that we produce a solicitation, we can seize the underlying solicitation to execute our preferred code.</p>
2	A database system security framework	<p>Database security is a developing worry as the measure of touchy information gathered and held in databases is quickly developing and the greater part of these information are being made available by means of the web. Most of the organizations, associations, and educating and learning establishments store delicate information in databases. As the vast majority of these information are electronically gotten to, It can, hence, be accepted that the trustworthiness of these various and delicate information is inclined to various types of dangers such as{Unauthorized access, robbery also access refusal. In this way, the requirement for making sure about databases has additionally expanded The essential targets of database security are to forestall unapproved admittance to information, forestall unapproved altering or alteration of information, and to likewise guarantee that, these information stay accessible at whatever point required. In this paper, we built up a database security system by joining diverse security instruments on a sensitive understudy data database application intended for Shehu Shagari College of Education Sokoto (SSCOE) with the point of limiting and keeping the information from Confidentiality, Integrity and Availability dangers.</p>
3	Database Security Model using Access Control Mechanism in Student Data Management	<p>Database security implies the assurance of information against unapproved divulgence, change, devastation. This paper presents a methodology to execute a DataAccess Policy to</p>

		<p>guarantee the insurance of the security privileges of understudies' records inside the understudy information the board framework. As per the framework, the organization of various security levels, assets, clients, assignments, and so forth is basic. This paper concentrated on an understudy information the executives framework by utilizing the Data Access Control model. As per the idea, just the head has the benefit to oversee or regulate the information. She/he gives a wide range of benefits required to look after clients, their approval and access, and the approved assets. The overseer controls the biggest data. This framework we present DAC access control instrument utilizing a MySQL database.</p>
4	Detection of Malicious Transactions in DBMS	<p>Database Management Systems (DBMS) are a key part in the data foundation of most associations and speak to a definitive layer in forestalling unapproved information gets to. A few components expected to ensure information, for example, confirmation, client benefits, encryption, and inspecting, have been executed in business DBMS. Pernicious exchanges executed by unapproved clients that may access the database by investigating framework weaknesses and unapproved database exchanges executed by approved clients can't be identified and halted by run of the mill security components. In this paper, we propose another system for the identification of malignant exchanges in DBMS.</p> <p>This paper proposed another component for the discovery of noxious exchanges in DBMS. The proposed system utilizes a chart that speaks to the profile of legitimate exchanges to recognize unapproved exchanges and comprises of two unique stages: exchange profiling and interruption discovery.</p> <p>Interruption discovery comprises in the recognition of clients executing arrangements of orders that conceivably speak to interruption endeavors.</p>
5	Database Security: What Students Need to Know	<p>Database security is a developing concern confirmed by an expansion in the quantity of detailed episodes of loss of or unapproved presentation to touchy information. As the measure of information gathered, held, and shared electronically extends, so does the need to comprehend database security. The Defense Information Systems Agency of the US Department of Defense (2004), in its Database Security Technical Implementation Guide, expresses that database security ought to give controlled, ensured admittance to the substance of a database just as protect the uprightness, consistency, and generally nature of the information. Understudies in the registering disciplines must build up a comprehension of the issues and moves identified with database security and must have the option to distinguish potential arrangements.</p> <p>At its center, database security endeavors to guarantee that lone confirmed clients perform approved exercises at approved occasions. While database security consolidates a wide exhibit of security themes, in any case, physical security, network security, encryption, and validation, this paper centers around the ideas and systems specific to making sure about info.</p>

		<p>Inside that unique situation, database security incorporates three builds: privacy or assurance of information from unapproved revelation, honesty or anticipation from unapproved information access, and accessibility or the ID of and recuperation from equipment and programming mistakes or malignant action bringing about the forswearing of information accessibility.</p> <p>In the figuring discipline educational plans, database security is regularly included as a theme in a starting database or initial PC security course. This paper presents a lot of sub-subjects that may be remembered for a database security segment of such a course. Planning to the three develops of information security, these points incorporate access control, application access, weakness, derivation, and inspecting components. Access control is the cycle by which rights and benefits are allocated to clients and database objects. Application access delivers the need to appoint proper access rights to outside applications requiring a database association. Weakness alludes to shortcomings that permit noxious clients to abuse assets. Induction alludes to the utilization of genuine information to surmise obscure data without reserving the privilege to straightforwardly recover that data. Database inspecting tracks database access and client movement giving an approach to distinguish breaks that have happened so restorative move may be made.</p>
6	Web and Database Security	<p>The continuous event of security occurrences, ventures and associations have now understood the significance of planning a security data framework. Today, data frameworks intensely depend on web and database advances, along these lines the dangers and dangers those advances confronted will likewise influence the security of data frameworks. Web and database security innovations can guarantee the classification, trustworthiness, and convenience of information in the data framework, and can viably ensure the security and dependability of the data framework. Subsequently, to more readily make sure about the data frameworks, we have to learn Web and database security-related information. This paper covers broadly useful and valuable information on web and database security.</p> <p>Web and database advances are in a fast development guide, for instance, web3.0 and diagram database (Angles, 2008) is getting increasingly more consideration. Simultaneously, related security issues will show up, however the principal security rules will continue as before. In this part, we quickly diagram the serious web and database protections, security structure standards, and security review rules and strategies. Because of the impediment of part length and variable programming dialects, most substance in each segment are general rules and rules. While sending functional data frameworks, we have to plan those standards to genuine usage. Data frameworks can be more made sure about on the off chance that we know and apply those advancements.</p>

7	<u>One approach to the testing of security of proposed database application software</u>	<p>This paper presents the idea of database arrangement and improvement considering security issues particularly when associated with the web. Notwithstanding insurances on security weaknesses executed on different degrees of database condition, for example, network, working framework, customer application, it is essential to ensure the database itself by maintaining a strategic distance from notable database security issues. To demonstrate that the proposed arrangement has an elevated level of security assurance, security testing must be performed. The general objective of security testing is to diminish weaknesses inside a product framework and we have proposed testing strategy including code survey and weakness appraisal that speak to the most far reaching of best practices for programming security affirmation.</p> <p>To shield programming with a database from an assailant, it is expected to ceaselessly screen and research every distributed instance of programming weaknesses. One of the approaches to improve the security of the database is to insert a shield on input information that may cause the potential security imperfections, by utilizing put away methods with worked in approvals on input information and to ensure the exchange of accreditations for client verification through standard web encryption procedures. As full security can't be ensured, the methodical way to deal with the testing of such a database must be performed by utilizing assault situations.</p>
8	<u>A HYBRID INTRUSION PREVENTION SYSTEM (HIPS) FOR WEB DATABASE SECURITY</u>	<p>Web database security is a difficult issue that ought to be thought about when structuring and fabricating business based web applications. Those applications as a rule incorporate basic cycles, for example, electronic-trade web applications that incorporate cash move by means of visa or ace cards. Security is a basic issue in other online applications, for example, destinations for military weapons organizations and public security of nations. The fundamental commitment of this paper is to present another web database security model that incorporates a blend of triple framework ; (I) Host Identity Protocol(HIP) in another verification strategy called DSUC (Data Security Unique Code), (ii) a solid separating decides that distinguish interlopers with high exactness, and (iii) an ongoing checking framework that utilizes the Uncertainty Degree Model (UDM) utilizing fluffy sets hypothesis. It was demonstrated that the mix of those three amazing security issues brings about a solid security model. In like manner, the proposed web database security model can recognize and give ongoing avoidance of interloper access with high accuracy. Exploratory outcomes have demonstrated that the proposed model presents acceptable web database security levels which reach sometimes to recognize and forestall over 93% of the gatecrashers. We have proposed another model for web database security utilizing the Ultra Hybrid security framework dependent on DSUC and Uncertainty Degree Model. Our tests and exploratory outcomes show that our framework is productive and fit for blocking</p>

		<p>gatecrashers from hacking into our framework and find dubious practices of inside and approved framework clients. We can ensure triple security layers, and the test shows that our framework can square 93% of aggressors on high burden.</p>
9	Approaches and Challenges in Database Intrusion Detection	<p>Databases regularly uphold undertaking business and store its privileged insights. This implies making sure about them from information harm and data spillage is basic. To manage interruptions against database frameworks, DatabaseIntrusion Detection Systems (DIDS) are often utilized. This paper presents an overview on the primary database interruption recognition strategies as of now accessible and examines the issues concerning the application at the database worker layer.</p> <p>The distinguished shaky areas show that most DIDS deficiently manage numerous attributes of explicit database frameworks, for example, impromptu remaining burdens and ready administration issues in information warehousing situations, for instance. In view of this investigation, research difficulties are introduced, prerequisites and rules for the structure of new or improved DIDS are proposed. The fundamental finding is that the turn of events and benchmarking of explicitly custom fitted DIDS for the setting where they work is a significant issue, and stays a test.</p> <p>We believe this work gives a solid motivating force to open the conversation between both the security and database research networks</p>
10	Unsupervised Visualization of SQL Attacks by Means of the SCMAS Architecture	<p>This presents presents an improvement of the SCMAS design pointed at making sure about SQL-run databases. The primary objective of such design is the recognition also, avoidance of SQL infusion assaults. The improvement comprises in the fuse of unaided projection models for the visual investigation of SQL traffic. Through the got projections, SQL infusion inquiries can be recognized and resulting moves can be made. It permits the location of SQL infusion assaults by separating them from typical SQL questions. This arrangement joins the benefits of MASs, for example, self-governance and appropriated critical thinking, with the perception, learning, and variation abilities of solo neural projection models. The proposed approach settles one of the absences of the SCMAS engineering: the perception of the information in a successful and natural manner. Further work will concentrate on the mix of the new representation capacities with the grouping cycle.</p>

11	<u>TRDBAC: Temporal reflective database access control</u>	<p>Database access control approaches can turn out to be very confused and complex in enormous databases, for example, clinic clinical frameworks, banks and undertaking asset arranging frameworks of huge endeavors and so on. The unpredictability in access control arrangements may brings about security penetrates if the approaches are questionable, not very much characterized and actualized mistakenly. for example HSBC database security break detailed in year 2006 in which an ex-representative swiped away just about 24,000 clients accounts because of wrong access arrangements. The entrance control approaches characterize the rights and benefits of clients on database objects. So as to keep these database frameworks secure, the database security ought to give controlled, ensured admittance to the substance of a database just as protect the honesty, consistency, and in general nature of the information. So as to execute the reliable database access control approaches, various models have been created by the database security network, for example, discretionary (DAC) and mandatory (MAC) access control models, role-based access control model (RBAC), reflective database access control (RDBAC). RDBAC is a moderately new and more expressive access control model that gives a more fine-grained level control than the past models. Move over database benefit is communicated as a database query itself, instead of as a static benefit contained in an entrance control framework.</p> <p>We propose Temporal Reflective Database Access Control (TRDBAC)- another entrance control strategy intended to address an impediment of RDBAC: the powerlessness to communicate time imperatives, similarly as TRBAC stretches out RBAC to consolidate the idea of time. To show how our new approach functions we have shown a contextual analysis on understudies' outcome data frameworks, in which arrangements are written in a period based augmentation of intelligent database access control (RDBAC) and changed over to SQL inquiries. At last we dissect the conduct of our new model.</p>
12	<u>An Adaptive Mechanism to Protect Databases against SQL Injection</u>	<p>The reason is to introduce versatile and savvy machines that can deal with SQL infusion assaults. This proposition centers around coordinating a case-based thinking (CBR) component with a neural network. The proposed arrangement consequently adjusts to changes in assault designs and gives the capacity to identify assaults freely of their advancement.</p> <p>A contextual investigation was proposed to approve the adequacy of the SQLCBR classifier model. The tests were led with a basic web application with database access, MySQL 5.0. The passages were computerized by utilizing the SQLMap 0.6.3tool, with which an underlying case base was set up for preparing the SQLCBR-Classifer. Other than the noxious inquiries to be examined by our answer were executed by this apparatus. The observational outcomes show that the best techniques are those that include the utilization of a neural network. This technique is more exact than factual strategies for</p>

		<p>recognizing assaults to databases on the grounds that the conduct of the programmer isn't direct, however unique and tumultuous. The adequacy of our answer was shown by the outcomes acquired.</p>
13	<p><u>Comparative Analysis of Various Biometric Techniques for Database Security</u></p>	<p>Biometric acknowledgment alludes to the utilization of various physiological attributes like unique mark acknowledgment, face acknowledgment, retina acknowledgment, hand math acknowledgment, iris acknowledgment, and so on and social qualities, for example, voice acknowledgment, walk acknowledgment, signature acknowledgment, and so forth. Called biometric identifiers of biometrics. For validation purposes, these highlights are utilized in a PC based security framework. The ID of an individual is getting significant as the ID cards, username, mystery secret word, and PIN are utilized for individual distinguishing proof. The ID can be taken by somebody and the PIN Number can be overlooked however the biometric methods beat every one of these issues. The biometric framework offers different focal points over the customary verification framework. The issue of data security gives assurance of data guaranteeing just approved clients can get to the data. The creator has reasoned that while the iris procedure gives to be the most secure, voice and face biometric methods had the most significant level of client acknowledgment, the unique mark strategy is the quick and exact biometric method for a more solid and secure framework and offered the best by and large arrangement.</p>
14	<p><u>Database Security Threats and Challenges in Database Forensic: A Survey</u></p>	<p>Realational Database Management Systems (RDBMS) is an assortment of utilizations that deal with the capacity, recovery, and control of database information. At the business level SQL Server, Oracle, Sybase, DB2, MySQL, and other mainstream database applications are generally acknowledged as RDBMSs. As in the current situation, enormous information security penetrates are happening at an exceptionally high rate so we point here to unearth the database frameworks which makes a few repetitive duplicates of touchy information that can be found in the table stockpiling, review logs, appeared sees, information word reference, SQL worker antiquities, and so forth for scientific examination. Likewise, a lot of measurable information is lying around a database framework to do an appropriate examination, and the most data important to bits together an occurrence afterward. So in this paper, we present a study that investigates the different convictions upon database criminology through various systems utilizing criminological calculations and apparatuses for examinations.</p>

		<p>Database Forensics is an exceptionally new field with little writing and not many instruments. This paper moved toward its undertaking by distinguishing different elements of Database Forensics. Different approaches for alter recognition are talked about. Significant difficulties are delineated dependent on the study offering new open doors for exploration and educating.</p> <p>Basically, this paper is planned to draw consideration towards Database Forensics with the desire for invigorating examination in this significant territory.</p>
--	--	---

7. Techniques to be Used & Experimental Setup:

[3 Marks]

Identify the techniques & experimental setup (S/W & H/W) you wish to use to address the list of objectives.

There are two types of database security mechanisms:

- Discretionary security mechanisms: These are used to grant privileges to users, including the capability to access specific data files, records, or fields in a specified mode (such as read, insert, delete, or update).
- Mandatory security mechanisms: These are used to enforce multilevel security by classifying the data and users into various security classes (or levels) and then implementing the appropriate security policy of the organization. Actions that will be performed:
 1. Account creation. This action creates a new account and password for a user or a group of users to enable access to the DBMS.
 2. Privilege granting. This action permits the DBA to grant certain privileges to certain accounts.
 3. Privilege revocation. This action permits the DBA to revoke (cancel) certain privileges that were previously given to certain accounts.
 4. Security level assignment. This action consists of assigning user accounts to the appropriate security clearance level.

The softwares that we will be using are:

MySQL: MySQL is a database management system that allows you to manage relational databases. It is open source software backed by Oracle.

8. Proposed Model / Algorithm / Framework

In this project we are using **Discretionary Access Control (DAC)** which is a type of Access Control mechanism which uses user identification procedures to identify and restrict access. DACs are discretionary because the subject (owner) can transfer authenticated objects or information access to other users. In other words, the owner determines object access privileges.

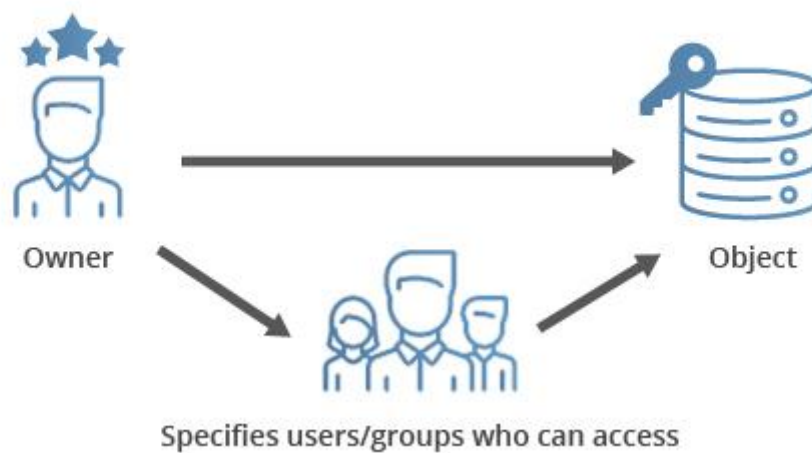
A mechanism implementing a DAC policy must be able to answer the question: "Does subject S have right R for object O?" Abstractly, the information needed to answer this question can be represented as a mathematical relation D on subjects, objects, and rights: if (S,O,R) is in D, then S does have right R for object O; otherwise, S does not.

More practically, the same information could also be represented as an *access control matrix* [Lampson 1971]. Each row of the matrix corresponds to a subject and each column to an object. Each cell of the matrix contains a set of rights. For example:

	file1	file2
Alice	rwX	r-x
Bob	r--	rw-

Real systems typically store the information from this matrix either by columns or by rows.

Discretionary Access Control (DAC)



Features:

- For each user, the credentials which are username and password will be authenticated before giving access to the database.
- After authentication, depending on your position in the firm you will be granted access to various resources.
- The Database Administrator is responsible for granting and revoking privileges from the users. He is the owner of this resource(database) and is responsible for deciding who does and does not have access, and exactly what access they are allowed to have.

Advantages of using this method:

- **Data Security** : Discretionary access control minimizes security risks. It creates a firewall against malware attacks, unauthorized access. This goes further to increase reliability in the organization.

- **Fast Authentication:** Unlike the manual control and authentication of access, DAS authentication is done in a matter of seconds. The DAC system automates the whole network such that it does not take more than few seconds to assess, verify and authorize or deny access.
- **Efficiency:** The security protocol is fail-proof. The components are structured in the most efficient way to monitor and restrict access. DAS devices are innovative enough to deal with attempts to override them and gain forceful entry into unauthorized areas of an organization.
- **Minimizes Cost:** This type of access control is also cost-effective, reducing the number of resources used in policing an organization's network.

9. Compare the list of objectives given and achieved so far

The objectives of our project were protection of data, authorisation, authentication.

The first objective of our project that we have achieved is protection of data i.e. Database security encompasses a range of security controls designed to protect the Database Management System. Protecting data in the database includes access control, data integrity, encryption, and auditing. Here we are encrypting data as an additional measure of security. Database security can include the secure management of encryption keys, protection of the encryption system, management of a secure, off-site encryption backup, and access restriction protocols.

Next objective of our project achieved is Authorization i.e. process where the database manager gets information about the authenticated user. Part of that information is determining which database operations the user can perform and which data objects a user can access. Users of the database can only view the contents they are authorized to view. The rest of the database is out of bounds to them.

The categories of authorization are:

- **Superintendent** - This is the highest administrative authorization for a user. Users with this authorization can also execute some database administrator commands such as restore or upgrade a database. Superintendent can access the data of doctor, patients, staff, hospital and update them.
- **Doctor** - Users with this authorization can access the data of staff, hospital and patients. They can restore or upgrade a database the data of the patients according to their visits or prescription.
- **Nurse** - Users with this authorization can access the data of hospital and doctor but they cannot modify the data.

- **Patient** - Users with this authorization can access the data of staff, hospital and doctor but they cannot modify the data.
- **Pharmacy** - Users with this authorization can access the data of hospital and patients.

Other objective for our project that we achieved is authentication i.e. the process or act of confirming that a user who is attempting to log in to a database is authorized to do so, and is only accorded the rights to perform activities that he or she has been authorized to do. The client has to establish the identity of the server and the server has to establish the identity of the client. This is done often by means of shared secrets. It can also be achieved by a system of higher authority which has previously established authentication. In client-server systems where data (not necessarily the database) is distributed, the authentication may be acceptable from a peer system. Note that authentication may be transmissible from system to system.

Triggers – A trigger in SQL is a procedural code that is automatically executed in response to certain events on a specified table. It is important to understand how these small codes make such a huge difference in database performance.

Triggers are, in fact, written to be executed in response to any of the following events –

- A **database manipulation (DML)** statement (DELETE, INSERT, or UPDATE)
- A **database definition (DDL)** statement (CREATE, ALTER, or DROP).
- A **database operation** (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN)

We need to create triggers so that whenever a user tries to manipulate the data in the database or the database an event will occur. In this way we can ensure that **integrity** is maintained

10. Experimental Results Obtained

Table Creation: -

```

SQL*Plus: Release 11.2.0.1.0 Production on Sun Oct 11 16:27:05 2020

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create table Aadhar_card
  2 (aadharc_no varchar(12) primary key,
  3 name varchar(25),
  4 address varchar(20),
  5 gender varchar(2));

Table created.

SQL> create table Hospital
  2 (hospital_id number(5) primary key,
  3 hospital_name varchar(20),
  4 no_of_beds number(5),
  5 address varchar(20));

Table created.

SQL> create table Doctor
  2 (doctor_id number(5) primary key,
  3 doctor_name varchar(25),
  4 aadharc_no references Aadhar_card,
  5 field varchar(25),
  6 salary number(8),
  7 hospital_id references Hospital);

Table created.

```

```

SQL> create table Staff
  2 (staff_id number(5) primary key,
  3 staff_name varchar(25),
  4 field varchar(15),
  5 aadharc_no references Aadhar_card);

Table created.

```

```

SQL> create table Patient
  2 (patient_id number(5) primary key,
  3 aadharc_no references Aadhar_card,
  4 doctor_id references Doctor,
  5 hospital_id references Hospital,
  6 health_issue varchar(20),
  7 medications varchar(20));

Table created.

SQL> create table visit
  2 (visit_id number(5),
  3 doctor_id references Doctor,
  4 patient_id references patient,
  5 prescription number(10));

Table created.

```

Inserting values: -

```

SQL> insert into Aadhar_card values ('18BIT0092', 'Nikhil Khurana', 'Jaipur', 'M');
1 row created.

SQL> insert into Aadhar_card values ('18BIT0109', 'Chitresh Gupta', 'Jaipur', 'M');
1 row created.

SQL> insert into Aadhar_card values ('18BIT0110', 'Vanshika Singh', 'Delhi', 'F');
1 row created.

SQL> insert into Aadhar_card values ('18BIT0427', 'Kritika Garg', 'Delhi', 'F');
1 row created.

SQL> select * from Aadhar_card;

AADHAR_CARD_ NAME          ADDRESS          GE
-----
18BIT0092     Nikhil Khurana   Jaipur          M
18BIT0109     Chitresh Gupta   Jaipur          M
18BIT0110     Vanshika Singh   Delhi           F
18BIT0427     Kritika Garg     Delhi           F

SQL>

```

```
SQL> insert into hospital values (1, 'Fortis Hospital', 300, 'Delhi');
insert into hospital values (1, 'Fortis Hospital', 300, 'Delhi')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SYS_C001105) violated

SQL> insert into hospital values (2, 'Fortis Hospital', 300, 'Delhi');
1 row created.

SQL> select * from hospital;

HOSPITAL_ID HOSPITAL_NAME      NO_OF_BEDS ADDRESS
-----
1 SMS Hospital      300 Jaipur
2 Fortis Hospital   300 Delhi
```

```
SQL> insert into Doctor values(201, 'Nikhil Khurana', '18BIT0092', 'Neurosurgeon', 20000000,1);
1 row created.

SQL> insert into Doctor values(201, 'Chitresh Gupta', '18BIT0109', 'Physician', 10000000,2);
insert into Doctor values(201, 'Chitresh Gupta', '18BIT0109', 'Physician', 10000000,2)
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SYS_C001106) violated

SQL> insert into Doctor values(501, 'Chitresh Gupta', '18BIT0109', 'Physician', 10000000,2);
1 row created.

SQL> select * from Doctor
2 ;

DOCTOR_ID DOCTOR_NAME      AADHAR_CARD_ FIELD
-----
SALARY HOSPITAL_ID
-----
201 Nikhil Khurana      18BIT0092  Neurosurgeon
20000000 1
501 Chitresh Gupta      18BIT0109  Physician
10000000 2
```

```
SQL> insert into staff values(371, 'Ramesh', 'Nurse', '18XXX0001');
1 row created.

SQL> insert into staff values(372, 'Suresh', 'Nurse', '18XXX0002');
1 row created.

SQL> select * from staff;

STAFF_ID STAFF_NAME      FIELD      AADHAR_CARD_
-----
371 Ramesh      Nurse      18XXX0001
372 Suresh      Nurse      18XXX0002
```

```
SQL> insert into Patient values(1, 'Vanshika Singh', '18BIT0110', 201,1, 'Brain Damage', 'Sleeping pills');
1 row created.

SQL> insert into Patient values(2, 'Kritika Garg', '18BIT0427', 501,2, 'Back Pain', 'pill');
1 row created.

SQL> select * from Patient
2 ;

PATIENT_ID NAME      AADHAR_CARD_ DOCTOR_ID HOSPITAL_ID
-----
HEALTH_ISSUE      MEDICATIONS
-----
1 Vanshika Singh      18BIT0110      201      1
Brain Damage      Sleeping pills
2 Kritika Garg      18BIT0427      501      2
Back Pain      pill
```

Creating users: -


```

SQL> create user Superintendent identified by sup123;
User created.
SQL> create user Doctor identified by doc123;
User created.
SQL> create user nurse identified by nurse123;
User created.
SQL> create user patient identified by pat123;
User created.
SQL> create user pharmacy identified by phar123;
User created.
SQL>

```

Granting Sessions: -

```

SQL> grant create session to Superintendent;
Grant succeeded.
SQL> grant create session to Doctor;
Grant succeeded.
SQL> grant create session to Nurse;
Grant succeeded.
SQL> grant create session to Patient;
Grant succeeded.
SQL> grant create session to Pharmacy;
Grant succeeded.

```

Granting tables: -

```

SQL> grant select, insert, update, delete on Hospital to Superintendent with grant option;
Grant succeeded.
SQL> grant select, insert, update, delete on Doctor to Superintendent with grant option;
Grant succeeded.
SQL> grant select, insert, update, delete on Staff to Superintendent with grant option;
Grant succeeded.
SQL> grant select, insert, update, delete on Patient to Superintendent with grant option;
Grant succeeded.
SQL> grant select, insert, update, delete on Visit to Superintendent with grant option;
Grant succeeded.
SQL> grant select on Aadhar_card to Superintendent;
Grant succeeded.

```

```

SQL> grant select on Hospital to Doctor;
Grant succeeded.
SQL> grant select, insert, update, delete on Doctor to Doctor;
Grant succeeded.
SQL> grant select on Staff to Doctor with grant option;
Grant succeeded.
SQL> grant select, insert, update, delete on Patient to Doctor with grant option;
Grant succeeded.
SQL> grant select, insert, update, delete on Visit to Doctor with grant option
2 ;
Grant succeeded.
SQL> grant select on Aadhar_card to Doctor;
Grant succeeded.

```

```

SQL> grant select on Hospital to Nurse;
Grant succeeded.

SQL> grant select on Doctor to Nurse;
Grant succeeded.

SQL> grant select, insert, update, delete on Staff to Nurse;
Grant succeeded.

SQL> grant select on Patient to Nurse;
Grant succeeded.

SQL> grant select on Visit to Nurse;
Grant succeeded.

SQL> grant select on Aadhar_card to Nurse;
Grant succeeded.

```

```

SQL> grant select on Hospital to Patient;
Grant succeeded.

SQL> grant select on Doctor to Patient;
Grant succeeded.

SQL> grant select on Staff to Patient;
Grant succeeded.

SQL> grant select, insert, update, delete on Patient to Patient with grant option;
Grant succeeded.

SQL> grant select, insert, update, delete on Visit to Patient with grant option;
Grant succeeded.

SQL> grant select, insert, update, delete on Aadhar_card to Patient;
Grant succeeded.

```

```

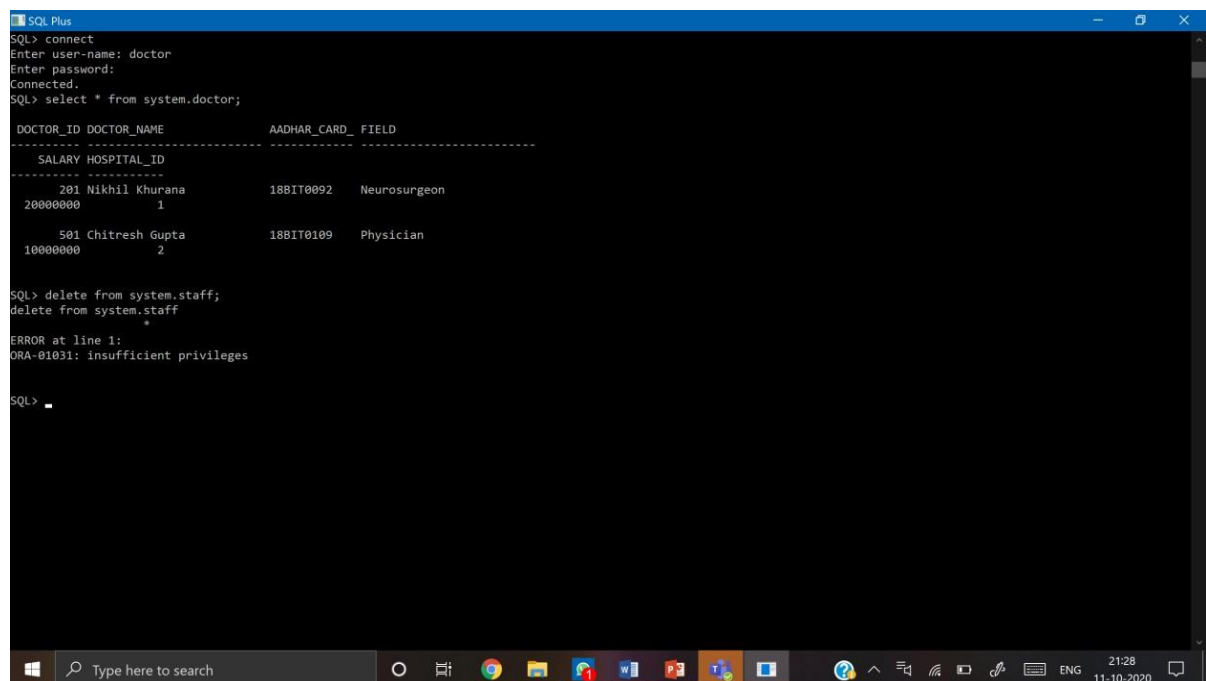
SQL> grant select on Hospital to Pharmacy;
Grant succeeded.

SQL> grant select on Patient to Pharmacy;
Grant succeeded.

```

Final example: -

The same user which logged in (Authentication) can view the Doctors information but can't delete the staff details as he is not authorized to do that.



```

SQL Plus
SQL> connect
Enter user-name: doctor
Enter password:
Connected.
SQL> select * from system.doctor;

DOCTOR_ID DOCTOR_NAME      AADHAR_CARD_FIELD
-----
SALARY HOSPITAL_ID
-----
201 Nikhil Khurana      18BIT0092  Neurosurgeon
20000000 1
501 Chitresh Gupta      18BIT0109  Physician
10000000 2

SQL> delete from system.staff;
delete from system.staff
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>

```

Triggers:

```
SQL> Create table doctor_changes (doctor_id number(5) , doctor_name varchar(25), aadhar_card_no varchar(12), field varchar(25), salary number(8), hospital_id number(5), operation varchar(25), user1 varchar(30));
```

Table created.

```
SQL> create or replace trigger Doctor_change
2  after insert or update or delete on doctor
3  for each row
4  declare
5  V varchar(30);
6  opr varchar(20);
7  begin
8  V:=USER;
9  if inserting then opr := 'insert';
10 elsif updating then opr := 'update';
11 else opr := 'delete';
12 end if;
13 Insert into doctor_changes (doctor_id , doctor_name , aadhar_card_no , field, salary, hospital_id , operation, user1 )
14 values (:old.doctor_id, :old.doctor_name, :old.aadhar_card_no, :old.field, :old.salary, :old.hospital_id, opr, V);
15 End;
16 /
```

Trigger created.

```
SQL>
```

```
SQL> create table Aadhar_card_changes
2  (aadhar_card_no varchar(12),
3  name varchar(25),
4  address varchar(20),
5  gender varchar(2),
6  operation varchar(25),
7  user1 varchar(30));
```

Table created.

```
SQL> create or replace trigger aadhar_card_change
2  after insert or update or delete on aadhar_card
3  for each row
4  declare
5  V varchar(30);
6  opr varchar(20);
7  begin
8  V:=USER;
9  if inserting then opr := 'insert';
10 elsif updating then opr := 'update';
11 else opr := 'delete';
12 end if;
13 Insert into aadhar_card_changes (aadhar_card_no , name, address, gender, operation, user1)
14 values (:old.aadhar_card_no, :old.name, :old.address, :old.gender, opr, V);
15 End;
16 /
```

Trigger created.

```
SQL>
```

```

SQL> create or replace trigger hospital_change
  2 after insert or update or delete on hospital
  3 for each row
  4 declare
  5 V varchar(30);
  6 opr varchar(20);
  7 begin
  8 V:=USER;
  9 if inserting then opr := 'insert';
 10 elsif updating then opr := 'update';
 11 else opr := 'delete';
 12 end if;
 13 Insert into hospital_changes (hospital_id , hospital_name, no_of_beds, address, operation, user1)
 14 values (:old.hospital_id, :old.hospital_name, :old.no_of_beds, :old.address, opr, V);
 15 End;
 16 /

```

Trigger created.

```

SQL> create table Staff_changes
  2 (staff_id number(5),
  3 staff_name varchar(25),
  4 field varchar(15),
  5 aadhar_card_no varchar(12),
  6 operation varchar(25),
  7 user1 varchar(30));

```

Table created.

```

SQL> create or replace trigger staff_change
  2 after insert or update or delete on staff
  3 for each row
  4 declare
  5 V varchar(30);
  6 opr varchar(20);
  7 begin
  8 V:=USER;
  9 if inserting then opr := 'insert';
 10 elsif updating then opr := 'update';
 11 else opr := 'delete';
 12 end if;
 13 Insert into staff_changes (staff_id , staff_name, field, aadhar_card_no, operation, user1)
 14 values (:old.staff_id, :old.staff_name, :old.field, :old.aadhar_card_no, opr, V);
 15 End;
 16 /

```

Trigger created.

SQL>

```
SQL> create table Patient_changes
  2 (patient_id number(5) ,
  3 Name varchar(20),
  4 aadhar_card_no varchar(12),
  5 doctor_id number(5),
  6 hospital_id number(5),
  7 health_issue varchar(20),
  8 medications varchar(20),
  9 operation varchar(25),
 10 user1 varchar(30));
```

Table created.

```
SQL> create or replace trigger patient_change
  2 after insert or update or delete on patient
  3 for each row
  4 declare
  5 V varchar(30);
  6 opr varchar(20);
  7 begin
  8 V:=USER;
  9 if inserting then opr := 'insert';
 10 elsif updating then opr := 'update';
 11 else opr := 'delete';
 12 end if;
 13 Insert into patient_changes (patient_id , name, aadhar_card_no, doctor_id, hospital_id, health_issue, medications, operation, user1)
 14 values (:old.patient_id, :old.name, :old.aadhar_card_no, :old.doctor_id, :old.hospital_id, :old.health_issue, :old.medications, opr, V);
 15 End;
 16 /
```

Trigger created.

```
SQL> create table visit_changes
  2 (visit_id number(5),
  3 doctor_id number(5),
  4 patient_id number(5),
  5 prescription number(10),
  6 operation varchar(25),
  7 user1 varchar(30));
```

Table created.

```
SQL> create or replace trigger visit_change
  2 after insert or update or delete on visit
  3 for each row
  4 declare
  5 V varchar(30);
  6 opr varchar(20);
  7 begin
  8 V:=USER;
  9 if inserting then opr := 'insert';
 10 elsif updating then opr := 'update';
 11 else opr := 'delete';
 12 end if;
 13 Insert into visit_changes (visit_id , doctor_id, patient_id, prescription, operation, user1)
 14 values (:old.visit_id, :old.doctor_id, :old.patient_id, :old.prescription, opr, V);
 15 End;
 16 /
```

Trigger created.

SQL>

Example: -

```

SQL> connect
Enter user-name: doctor
Enter password:
Connected.
SQL> select * from system.doctor;

```

DOCTOR_ID	DOCTOR_NAME	AADHAR_CARD_FIELD
201	Nikhil	18BIT0092 Neurosurgeon
501	Chitresh Gupta	18BIT0109 Physician

```

SQL> update system.doctor set doctor_name='Niks' where doctor_id=201;
1 row updated.
SQL> select * from system.doctor;

```

DOCTOR_ID	DOCTOR_NAME	AADHAR_CARD_FIELD
201	Niks	18BIT0092 Neurosurgeon
501	Chitresh Gupta	18BIT0109 Physician

```

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> select * from doctor_changes;

```

DOCTOR_ID	DOCTOR_NAME	AADHAR_CARD_FIELD
201	Nikhil	18BIT0092 Neurosurgeon

SALARY	HOSPITAL_ID	OPERATION	USER1
20000000	1	update	DOCTOR

We get the old information and the user who changes the name.

11.Conclusion:

We conclude that using this we can secure the data of the users that are the patients, hospital and the staff. We have achieved the security of the database by authentication, authorization and maintaining the integrity.

12. References:

[2 Marks]

Arrange all the references in APA style of referencing. All the references should be cited in the literature survey.

- Database Security Threats & Prevention by Simanta Shekhar Sarmah
- A DATABASE SYSTEM SECURITY FRAMEWORK by Habiba Muhammad Sani and Muhammad Mika'ilu Yab
- Database Security Model using Access Control Mechanism in Student Data Management by Aye Mon Win and Khin Lay Myint
- Detection of Malicious Transactions in DBMS by Ayushi, Anisha Sharma & Reena Bansal
- Database Security: What Students Need to Know by Meg Coffin Murray
- Web and Database Security by Jiping Xiong, Lifeng Xuan, Jian Zhao and Tao Huang
- One approach to the testing of security of proposed database application software by SINIŠA S. ILIĆ, LJUBOMIR LAZIĆ and PETAR SPALEVIĆ
- A HYBRID INTRUSION PREVENTION SYSTEM (HIPS) FOR WEB DATABASE SECURITY by Eslam Mohsin Hassib, Aida Osman Abdelgwad and Ahmed Ibrahim Saleh.
- Approaches and Challenges in Database Intrusion Detection by Macro Vieira
- Unsupervised Visualization of SQL Attacks by Means of the SCMAS Architecture Álvaro Herrero, Cristian I. Pinzón, Emilio Corchado, and Javier Bajo*
- TRDBAC: Temporal reflective database access control by Zahid Rashid and Abdul Basit
- An Adaptive Mechanism to Protect Databases against SQL Injection by Cristian I. Pinzón, Juan F. De Paz, Javier Bajo, Juan M. Corchado
- Comparative Analysis of Various Biometric Techniques for Database Security by Harpreet Saini, Kanwal Garg
- Database Security Threats and Challenges in Database Forensic: A Survey by Harmeet Kaur Khanuja and D .S. Adane