TVET CERTIFICATE V in SOFTWARE DEVELOPMENT

DATABASE SECURITY

SFDDS501

Apply Database Security

Competence

Credits: 8

Learning hours: 80

Sector: ICT

Sub-sector: Software Development

Module Note Issue date: June, 2020

Purpose statement

This module describes the skills, knowledge, and attitudes required to describe database security standards, perform security of database on both system and data level and conduct database auditing

Table of Contents

Total Number of Pages: 58

Learning Unit 1 – Describe database security concepts and standards

LO 1.1 – Describe the scope of database security

- Topic 1: Introduction to database

Database security refers to the collective measures used to protect and secure a database or database management software from illegitimate use and malicious threats and attacks.

It is a broad term that includes a multitude of processes, tools and methodologies that ensure security within a database environment.

Database security covers and enforces security on all aspects and components of databases. This includes:

- Data stored in database
- Database server
- Database management system (DBMS)
- Other database workflow applications

Database security is generally planned, implemented and maintained by a database administrator and or other information security professional.

Some of the ways database security is analyzed and implemented include:

- Restricting unauthorized access and use by implementing strong and multifactor access and data management controls
- Load/stress testing and capacity testing of a database to ensure it does not crash in a distributed denial of service (DDoS) attack or user overload
- Physical security of the database server and backup equipment from theft and natural disasters
- Reviewing existing system for any known or unknown vulnerabilities and defining and implementing a road map/plan to mitigate them

Purpose of database security

Database security is more than just important: it is essential to any company with any online component. Sufficient database security prevents data bring lost or compromised, which may have

serious ramifications for the company both in terms of finances and reputation. Database security helps:

- Company's block attacks, including ransomware and breached firewalls, which in turn keeps sensitive information safe.
- Prevent malware or viral infections which can corrupt data, bring down a network, and spread to all end point devices.
- Ensure that physical damage to the server doesn't result in the loss of data.
- Prevent data loss through corruption of files or programming errors.

Importance of database security

- Confidentiality is the most important aspect of database security, and is most commonly enforced through encryption. Encryption should be done both for data-in-transit and data-at-rest.
- Integrity is yet another crucial aspect of database security, because it ensures that only the correct people will be able to see privileged company information. The integrity of a database is enforced through a User Access Control system that defines permissions for who can access which data.

The integrity aspect extends beyond simply permissions, however. Security implementations like authentication protocols, strong password policies, and ensuring unused accounts (like of employees that have left the company) are locked or deleted, further strengthen the integrity of a database.

- Availability relates to the need for databases to be up and available for use. Databases need to be dependable in order to be functional, which requires they be up and running whenever the organization is. This means downtimes should be planned on weekends and servers kept up-to-date.
- Separation of tasks and access control: Access control primarily aims at preventing malicious attacks and potential threats from within the organization. While there are instances of deliberate malicious attacks from insiders, more often than not these result from theft of login credentials.

  By separating duties and assigning functionality and privileges according to user requirements, you can limit the extent of damage in case of potential or actualized

threats. For instance, a user responsible for creating backup files to the DB needs never see the actual content in the DB. A tester will need access into the database, but not necessarily to actual data stored therein.

- **Data encryption**: Another form might be to protect the database externally i.e. ensure that all data leaving the DB in whatever format is protected from malicious access. Application of this is at the point of transportation through encryption of the transport channel, and is important for every item of information leaving the organization.

Types of database security

- Access authorization.
- Access controls.
- Views.
- Backup and recovery of data.
- Data integrity.
- Encryption of data.
- RAID technology.

What is a threat? A threat is any situation or event, whether intentionally or incidentally, can cause damage, which can reflect an adverse effect on the database structure and, consequently, the organization. A threat may occur by a situation or event involving a person or the action or situations that are probably to bring harm to an organization and its database.

BACKUP AND RECOVERY

Every Database Management System should offer backup facilities to help with the recovery of a database after a failure. It is always suitable to make backup copies of the database and log files at the regular period and for ensuring that the copies are in a secure location. In the event of a failure that renders the database unusable, the backup copy and the details captured in the log file are used to restore the database to the latest possible consistent state.

Encryption

Encryption is the method by which information is converted into secret code that hides the information's true meaning. The science of encrypting and decrypting information is called cryptography.

- Topic 2: Database Security Policies

Security policies are guidelines describing all (if possible) actions pertinent to the information system. Logical access control belongs to that area and so security policies should define principles on which is design of secure DBMS based.

To eliminate threats, it is necessary to define proper security policy. Security policies are governing principles adopted by organizations.

Generally, policies should give answers on basic security questions. Policies can be divided into two basic types - of minimal (army) and maximal (university) privilege. System with such a policy is called closed or opened, respectively.

Talking about access control, the way of administration of access rules should be determined.

- Hierarchical decentralized - central authorizer distributes responsibilities among dependent subauthorizers.
- Ownership based - owner of an object (its author) determines access to the object.
- Cooperative authorization - authorization of special rights for special resources is approved by all members of predefined group.

- Topic 3: Database security framework

The security framework was designed based on the core facets of database security mechanisms (CIA) to help address the issues of confidentiality, integrity and authenticity as well as availability of data. ... While on the other hand, the system rejects and denied unauthorized users access to the system and data.

Database security encompasses three constructs (security objectives/goals):

- **Confidentiality/secrecy:** *information is only disclosed to authorized users.* Protection of data from unauthorized disclosure.
- **Integrity:** *information is only modified by authorized users.*
  Prevention from unauthorized data access.
- **Availability:** *information is accessible by authorized users.*
  Identification and recovery from hardware and software errors or malicious activity resulting in the denial of data availability.

LO 1.2 – Elaborate database security models

- Topic 1: Definition of security models

Security models are the formal description of security policies. Security models are useful tools for evaluating and comparing security policies. Security models allow us to test security policies for completeness and consistency. They describe what mechanisms are necessary to implement a security policy.

Access Control

Access Control The purpose of access control is to limit the actions or operations that a legitimate user of a computer system can perform. Access control constraints what a user can do directly, as well as what programs executing on behalf of the users are allowed to do. In this way access control seeks to prevent activity that could lead to a breach of security.

Authentication and Authorization

Both the terms are often used in conjunction with each other in terms of security, especially when it comes to gaining access to the system. Both are very crucial topics often associated with the web as key pieces of its service infrastructure. However, both the terms are very different with totally different concepts. While it's true that they are often used in the same context with the same tool, they are completely distinct from each other.

Difference between Authentication and Authorization.

Authentication means confirming your own identity, while **authorization** means granting access to the system. In simple terms, authentication is the process of verifying who you are,
while authorization is the process of verifying what you have access to.

Authentication is about validating your credentials like User Name/User ID and password to verify your identity. Authorization, on the other hand, occurs after your identity is successfully authenticated by the system, which ultimately gives you full permission to access the resources such as information, files, databases, funds, locations, almost anything.

Access philosophies and management

Discretionary control is where specific privileges are assigned on the basis of specific assets, which authorised users are allowed to use in a particular way. The security DBMS has to construct an access matrix including objects like relations, records, views and operations for each user - each entry separating create, read, insert and update privileges. This matrix becomes very intricate as authorisations will vary from object to object. The matrix can also become very large, hence its implementation frequently requires the kinds of physical implementation associated with sparse matrices. It may not be possible to store the matrix in the computer's main memory.

- Topic 2: Description of database access matrix models

Security models are described in terms of the following elements (database access matrix models):

• **Subjects:** Entities that request access to objects.

• **Objects:** Entities for which access request is being made by subjects.

• **Access Modes:** Type of operation performed by subject on object (read, write, create etc.).

• **Policies:** Enterprise wide accepted security rules.

• **Authorizations:** Specification of access modes for each subject on each object.

• Administrative Rights: Who has rights in system administration and what responsibilities administrators have.

• Axioms: Basic working assumptions.

LO 1.3 – Explore database security principles

- Topic 1: Elaborate Database attacks

1. Cloud database configuration errors

Barely a week goes by without a new data breach caused by insecurely configured cloud databases or storage services.

Public Cloud service IP addresses are not secret and are continually scanned for vulnerabilities by malicious persons and security researchers.

2. SQL injection

SQL injection vulnerabilities occur when application code contains dynamic database queries which directly include user supplied input.

3. Weak Authentication

Weak authentication schemes allow attackers to assume the identity of legitimate database users by stealing or otherwise obtaining login credentials.

4. Privilege abuse

Users may abuse legitimate data access privileges for unauthorised purposes. For example, a user in sales with privileges to view individual customer records may abuse that privilege to retrieve all customer records to pass to a competitor.

5. Excessive privileges

If users hold privileges that exceed the requirements of their job function, these privileges may be abused by the individual, or an attacker who compromises their account. When people move roles, they may be given the new privileges they need without those they no-longer require being removed.

6. Inadequate logging and weak auditing

Logging and auditing are key to deterring and detecting misuse and enabling adequate investigation of suspected data compromise. In this context, logging is the collection of data - and auditing is someone actually looking at it.

7. Denial of service

Network level Denial of Service (DoS) attacks from the internet, can overwhelm your system regardless of the capacity of its internet connection. Cloud based DoS protection services are the usual defence against this and many offer a free protection tier.

8. Exploiting unpatched services

While up-to-date patching won't make you secure, operating vulnerable unpatched services will significantly increase the likelihood of being compromised.

9. Insecure system architecture

While controls against specific database threats are important, they must form part of a design which is secure overall. This is a big topic, but some pointers are given below:

10. Inadequate Backup

Theft of database backup tapes and hard disks has long been a concern, but new threats to the availability of data have arisen and these must not be ignored.

11. Weak Audit Trail

Failure to collect detailed **audit** records of **database** activity represents a serious organizational risk on many levels. Organizations with **weak** (or sometimes non-existent) **database audit** mechanisms will increasingly find that they are at odds with industry and government regulatory requirements.

12. Exposure of backup data

Sensitive data exposure occurs when an application, company, or other entity inadvertently exposes personal data. Sensitive data exposure differs from a data breach, in which an attacker accesses and steals information.

Sensitive data exposure occurs as a result of not adequately protecting a database where information is stored. This might be a result of a multitude of things such as weak encryption, no encryption, software flaws, or when someone mistakenly uploads data to an incorrect database.

- Topic 2: Elaborate database control methods

Access Control: The purpose of access control is to limit the actions or operations that a legitimate user of a computer system can perform. Access control constraints what a user can do directly, as well as what programs executing on behalf of the users are allowed to do. In this way access control seeks to prevent activity that could lead to a breach of security.

Inference Policy: Inference is a database system technique used to attack databases where malicious users infer sensitive information from complex databases at a high level. In basic terms, inference is a data mining technique used to find information hidden from normal users.

User Identification /Authentication: Identification is the ability to identify uniquely a user of a system or an application that is running in the system. Authentication is the ability to prove that a user or application is genuinely who that person or what that application claims to be.

Accountability and auditing: The Board aims to present a clear and meaningful assessment of the Company's financial positions and their reports to the shareholders, investors and regulatory authorities. This assessment is primarily provided in the annual financial statements, quarterly result announcements as well as the Chairman's statement and review of the operations in the annual report.

Encryption: Encryption is the method by which information is converted into secret code that hides the information's true meaning. The science of encrypting and decrypting information is called cryptography.

Learning Unit 2 – Perform system security

LO 2.1 – Define system security coverage.

- Topic 1: Definition of system security terms

System security Control of access to a computer system's resources, specially its data and operating system files.

The objective of system security is the protection of information and property from theft, corruption and other types of damage, while allowing the information and property to remain accessible and productive. System security includes the development and implementation of security countermeasures. There are a number of different approaches to computer system security, including the use of a firewall, data encryption, passwords and biometrics.

Elements of system security

1. Availability: As the name suggests, availability specifies whether the data or resource is available when it is required or requested by the client. The information that has been requested will possess the actual value only when legitimate users can take access to those resources at the right time. But cybercriminals seize those data so that the request to access those resources gets denied (leads to downtime of a working server), which is a conventional attack.

2. Integrity: This refers to the techniques to ensure that all the data or resources that can be accessed in real-time are legitimate, correct, and protected from unlawful user (hackers) modification. Data integrity has become a primary and essential component or element of information security because users have to trust online information to use them. Non-trusted data compromises the integrity and hence will violate one of the six elements. Data integrity is verified through techniques like checksums, change in hash values, and data comparison.

3. Authenticity: Authenticity is another essential element, and authentication can be defined as the process of ensuring and confirming that the identity of the user is genuine and legitimate. This process of authentication takes place when the user tries to gain access to any data or information (commonly done by login or biometric access). However, cybercriminals use more sophisticated tools and techniques to gain such access with the use of social engineering, password guessing, brute force techniques, or cracking ciphers.

4. Confidentiality: can be defined as permitting approved users for accessing to all sensitive as well as a protected information. Confidentiality takes care of the fact that confidential information and

other resources have to be revealed to legitimate and authorize users only. Confidentiality can be made certain by the use of role-based security techniques for ensuring user or viewer's authorization as well as access controls on any particular data.

5. Non-repudiation: can be defined as the way of assurance that message transmitted among two or more users via digital signature or through the use of encryption is accurate, and no one can deny the authentication of the digital signature on any document. Authentic data, as well as its origination, can be acquired with the help of a data hash.

6. Utility: as the name suggests is used for any purpose or reason and is accessed and then used by users. It is not entirely the type of element for security, but if the utility of any resource becomes vague or useless, then it is of no use. Cryptography is used to preserve the efficiency of any resource sent over the internet. Various encryption mechanisms are used for securing the message or data sent over the internet so that it is not altered during the transmission; otherwise, the utility of that resource will not prevail.

Access Control: The purpose of access control is to limit the actions or operations that a legitimate user of a computer system can perform. Access control constraints what a user can do directly, as well as what programs executing on behalf of the users are allowed to do. In this way access control seeks to prevent activity that could lead to a breach of security.

- Topic 2: Definition of fundamentals of access control

Access Control The purpose of access control is to limit the actions or operations that a legitimate user of a computer system can perform. Access control constraints what a user can do directly, as well as what programs executing on behalf of the users are allowed to do. In this way access control seeks to prevent activity that could lead to a breach of security.

Elements of access controls

The key to understanding access control security is to break it down. There are three core elements to access control.

1. **Identification**: For access control to be effective, it must provide some way to identify an individual. The weakest identification capabilities will simply identify someone as part of a vague, poorly defined group of users who should have access to the system. Your TechRepublic username, a PGP e-mail signature, or even the key to the server closet provides some form of identification.

2. **Authentication**: Identification requires authentication. This is the process of ensuring that the identity in use is authentic -- that it's being used by the right person. In its most common form in IT security, authentication involves validating a password linked to a username. Other forms of authentication also exist, such as fingerprints, smartcards, and encryption keys.
3. **Authorization**: The set of actions allowed to a particular identity makes up the meat of authorization. On a computer, authorization typically takes the form of read, write, and execution permissions tied to a username.

Types of Access Control

Access control types include:

- Administrative
- Physical
- Technical

Administrative Access Control

Administrative access control sets the access control policies and procedures for the whole organization, defines the implementation requirements of both physical and technical access control, and what the consequences of non-compliance will be. Some examples are: supervisory structure, staff and contractor controls, information classification, training, auditing, and testing.

Physical Access Control

Physical access control is critical to an organizations security and applies to the access or restriction of access to a place such as property, building or room. Some examples are: fences, gates, doors, turnstiles, etc. using locks, badges, biometrics (facial recognition, fingerprints), video surveillance cameras, security guards, motion detectors, mantrap doors, etc. to allow access to certain areas.

Technical or Logical Access Control

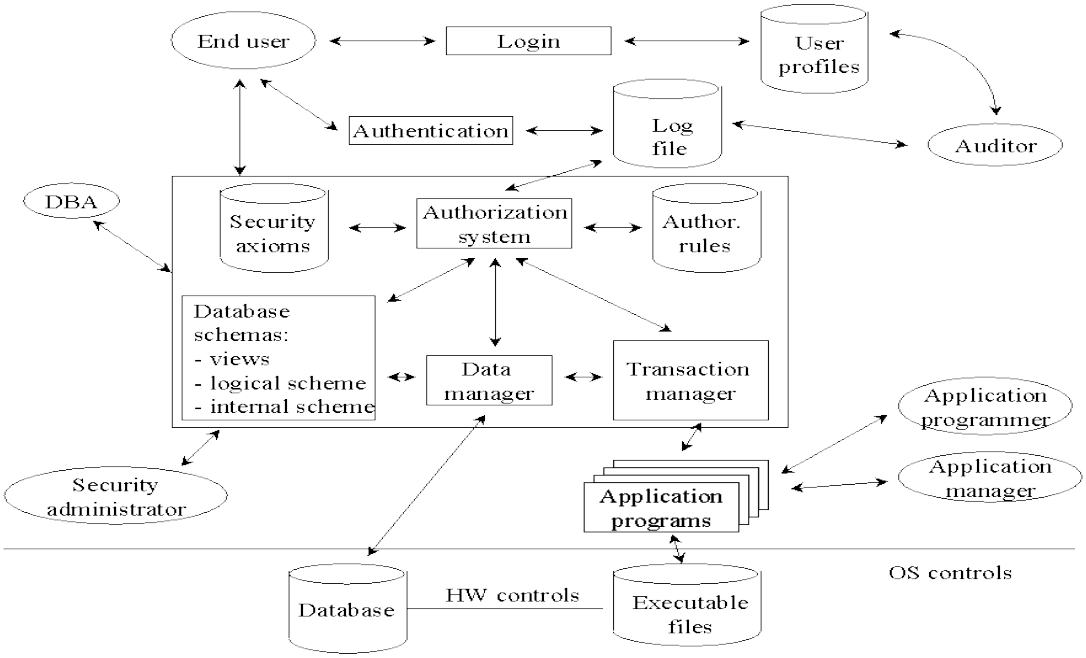
Technical or logical access control limits connections to computer networks, system files, and data. It enforces restrictions on applications, protocols, operating systems, encryptions mechanisms, etc.

In today's increasingly digital world, modern access control systems combine both administrative, physical and technical access control to limit access to sensitive data and physical locations, providing a much higher level of security. Some examples are: access control lists, intrusion detection systems, and antivirus software.

- Topic 3: Introduction to system database access control

Access control is responsible for control of rules determined by security policies for all direct accesses to the system. Traditional control systems work with notions subject, object and operation.

For better image look at the figure of secure DBMS.



What is a database

A database is an organized collection of structured data stored electronically in a computer system.

Oracle Database Architecture

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

- Topic 4: Oracle Database features

Oracle Database allows you to quickly and safely store and retrieve data. Here are the integration benefits of the Oracle Database:

- Oracle Database is cross-platform. It can run on various hardware across operating systems including Windows Server, Unix, and various distributions of GNU/Linux.
- Oracle Database has its networking stack that allows application from a different platform to communicate with the Oracle Database smoothly. For example, applications running on Windows can connect to the Oracle Database running on Unix.
- ACID-compliant – Oracle is ACID-compliant Database that helps maintain data integrity and reliability.
- Commitment to open technologies – Oracle is one of the first Database that supported GNU/Linux in the late 1990s before GNU/Linux become a commerce product. It has been supporting this open platform since then.

Oracle Database has several structural features that make it popular:

- **Logical data structure** – Oracle uses the logical data structure to store data so that you can interact with the database without knowing where the data is stored physically.
- **Partitioning** – is a high-performance feature that allows you to divide a large table into different pieces and store each piece across storage devices.
- **Memory caching** – the memory caching architecture allows you to scale up a very large database that still can perform at a high speed.
- **Data Dictionary** is a set of internal tables and views that support administer Oracle Database more effectively.
- **Backup and recovery** – ensure the integrity of the data in case of system failure. Oracle includes a powerful tool called Recovery Manager (RMAN) – allows DBA to perform cold, hot, and incremental database backups and point-in-time recoveries.

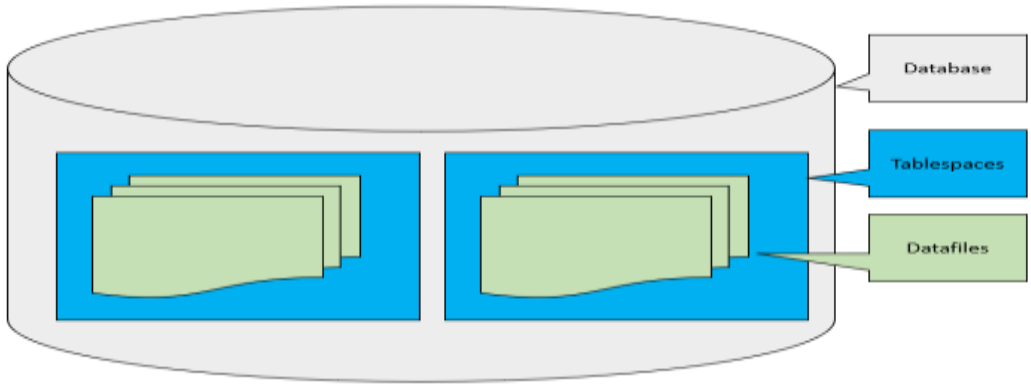
- **Clustering** – Oracle Real Application Clusters (RAC) – Oracle enables high availability that enables the system is up and running without interruption of services in case one or more server in a cluster fails.

- Topic 5: Creating table spaces in oracle

An Oracle database consists of one or more logical storage units called tablespaces, which collectively store all of the database's data. Each tablespace in an Oracle database consists of one or more files called datafiles, which are physical structures that conform to the operating system in which Oracle is running.

Oracle divides a database into one or more logical storage units called tablespaces. Each tablespace consists of one or more files called datafiles. A datafile physically stores the data objects of the database such as tables and indexes on disk. In other words, Oracle logically stores data in the tablespaces and physically stores data in datafiles associated with the corresponding tablespaces.

The following picture illustrates the relationship between a database, tablespaces, and datafiles:



Default tablespaces in Oracle

Oracle comes with the following default tablespaces: SYSTEM, SYSAUX, USERS, UNDOTBS1, and TEMP.

Privileges in oracle

A user privilege is a right to execute a particular type of SQL statement, or a right to access another user's object. The types of privileges are defined by Oracle

Some examples of privileges include the right to:

- Connect to the database (create a session)

- Create a table
- Select rows from another user's table
- Execute another user's stored procedure

There are two distinct categories of privileges:

- System privileges
- Schema object privilege

System Privileges

A system privilege is the right to perform a particular action, or to perform an action on any schema objects of a particular type. For example, the privileges to create tablespaces and to delete the rows of any table in a database are system privileges. There are over 60 distinct system privileges.

Schema Object Privileges

A schema object privilege is a privilege or right to perform a particular action on a specific schema object:

- Table
- View
- Sequence
- Procedure
- Function
- Package

- Topic 6: Roles in oracle

The types of privileges are defined by Oracle. Roles, on the other hand, are created by users (usually administrators) and are used to group together privileges or other roles. They are a means of facilitating the granting of multiple privileges or roles to users.

Use the CREATE ROLE statement to create a role, which is a set of privileges that can be granted to users or to other roles. You can use roles to administer database privileges. You can add privileges to a role and then grant the role to a user. The user can then enable the role and exercise the privileges granted by the role.

Users in Oracle Database

In Oracle terminology, a user is someone who can connect to a database (if granted enough privileges) and optionally (again, if granted the appropriate privileges) can own objects (such as tables) in the database.

Database Administrators

Each database requires at least one database administrator (DBA). An Oracle Database system can be large and can have many users. Therefore, database administration is sometimes not a one-person job, but a job for a group of DBAs who share responsibility.

A database administrator's responsibilities can include the following tasks:

- Installing and upgrading the Oracle Database server and application tools
- Allocating system storage and planning future storage requirements for the database system
- Creating primary database storage structures (tablespaces) after application developers have designed an application
- Creating primary objects (tables, views, indexes) once application developers have designed an application
- Modifying the database structure, as necessary, from information given by application developers
- Enrolling users and maintaining system security
- Ensuring compliance with Oracle license agreements
- Controlling and monitoring user access to the database
- Monitoring and optimizing the performance of the database
- Planning for backup and recovery of database information
- Maintaining archived data on tape
- Backing up and restoring the database
- Contacting Oracle for technical support

Database users interact with the database through applications or utilities. A typical user's responsibilities include the following tasks:

- Entering, modifying, and deleting data, where permitted
- Generating reports from the data

- Topic 7: Types of users in Oracle Database

A user is either

- A local user,
- An external user, or
- A global user

Local users

A local user needs a password to log on to the database.

External users

An external user, unlike a local user, doesn't need a password to log on to the database, instead, an external service (such as the operating system) authenticates the user when (s)he logs on the database.

Global users

A global user, like an external user, doesn't need a password to log on to the database, instead, (s)he is authenticated by an enterprise directory service.

Database rights (Privileges)

Users can be assigned rights what they're allowed to do in a database and what not. These rights are called privileges.

LO 2.2 – Manage users according to the system roles and privileges

- Topic 1: Creating users

The CREATE USER statement allows you to create a new database user which you can use to log in to the Oracle database.

The basic syntax of the CREATE USER statement is as follows:

```
CREATE USER username
  IDENTIFIED BY password
  [DEFAULT TABLESPACE tablespace]
  [QUOTA {size | UNLIMITED} ON tablespace]
  [PROFILE profile]
  [PASSWORD EXPIRE]
  [ACCOUNT {LOCK | UNLOCK}];
```

In this syntax:

CREATE USER username

Specify the name of the user to be created.

IDENTIFIED BY password

Specify a password for the local user to use to log on to the database. Note that you can create an external or global user, which is not covered in this tutorial.

DEFAULT TABLESPACE

Specify the tablespace of the objects such as tables and views that the user will create.

If you skip this clause, the user's objects will be stored in the database default tablespace if available, typically it is USERS tablespace; or the SYSTEM tablespace in case there is no database default tablespace.

QUOTA

Specify the maximum of space in the tablespace that the user can use. You can have multiple QUOTA clauses, each for a tablespace.

Use UNLIMITED if you don't want to restrict the size in the tablespace that user can use.

PROFILE profile

A user profile limits the database resources or password that the user cannot exceed. You can assign a profile to a newly created user. If you skip this clause, Oracle will assign the DEFAULT profile to the user.

PASSWORD EXPIRE

Use the PASSWORD EXPIRE if you want to force the user to change the password for the first time the user logs in to the database.

ACCOUNT {LOCK | UNLOCK}

Use ACCOUNT LOCK if you want to lock user and disable access. On the other hand, specify ACCOUNT UNLOCK to unlock user and enable access.

To execute the CREATE USER statement, you must have the CREATE USER system privilege. Once you create the new user, the privilege domain of the user will be empty. Therefore, if you want to the user to be able to login to the database, you should grant the CREATE SESSION system privilege to the user.

Oracle CREATE USER examples

Let's practice with the CREATE USER statement.

Using Oracle CREATE USER statement to create a new local user example

This example uses the CREATE USER statement to create a new local user named john with the password abcd1234:

```
CREATE USER john IDENTIFIED BY abcd1234;
```

To find a list of users with the OPEN status, you query the information from the dba_users:

```
SELECT username, default_tablespace, profile, authentication_type FROM dba_users WHERE
   account_status = 'OPEN';
```

Let's use the john account to log in the database.

Launch the SQL*Plus program and enter the following information:

Enter user-name: john@pdborcl

Enter password:<john_password>

Oracle issued the following error:

ERROR: ORA-01045:

user JOHN lacks CREATE SESSION privilege; logon denied

To enable the user john to log in, you need to grant the CREATE SESSION system privilege to the user john by using the following statement:

```
GRANT CREATE SESSION TO john;
```

Now, the user john should be able to log in the database.

Enter user-name: john@pdborcl

Enter password:

Connected to:

Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

- Topic 2: Allocation of users privileges

SQL GRANT REVOKE Commands

DCL commands are used to enforce database security in a multiple user database environment. Two types of DCL commands are GRANT and REVOKE. Only Database Administrator's or owner's of the database object can provide/remove privileges on a database object.

SQL GRANT Command

SQL GRANT is a command used to provide access or privileges on the database objects to the users.

The Syntax for the GRANT command is:

```
GRANT privilege_name
ON object_name
TO {user_name |PUBLIC |role_name}
[WITH GRANT OPTION];
```

- Privilege_name is the access right or privilege granted to the user. Some of the access rights are ALL, EXECUTE, and SELECT.
- Object_name is the name of an database object like TABLE, VIEW, STORED PROC and SEQUENCE.
- User_name is the name of the user to whom an access right is being granted.
- User_name is the name of the user to whom an access right is being granted.
- PUBLIC is used to grant access rights to all users.
- ROLES are a set of privileges grouped together.
- WITH GRANT OPTION - allows a user to grant access rights to other users.

For Example: GRANT SELECT ON employee TO user1;

This command grants a SELECT permission on employee table to user1.You should use the WITH GRANT option carefully because for example if you GRANT SELECT privilege on employee table to user1 using the WITH GRANT option, then user1 can GRANT SELECT privilege on employee table to another user, such as user2 etc. Later, if you REVOKE the SELECT privilege on employee from user1, still user2 will have SELECT privilege on employee table.

SQL REVOKE Command:

The REVOKE command removes user access rights or privileges to the database objects.

The Syntax for the REVOKE command is:

```
REVOKE privilege_name
ON object_name
FROM {user_name |PUBLIC |role_name}
```

For Example: REVOKE SELECT ON employee FROM user1;

This command will REVOKE a SELECT privilege on employee table from user1.When you REVOKE SELECT privilege on a table from a user, the user will not be able to SELECT data from that table anymore. However, if the user has received SELECT privileges on that table from more than one users, he/she can SELECT from that table until everyone who granted the permission revokes it. You cannot REVOKE privileges if they were not initially granted by you.

Privileges and Roles:

Privileges: Privileges defines the access rights provided to a user on a database object. There are two types of privileges.

1) **System privileges** - This allows the user to **CREATE, ALTER,** or **DROP** database objects.
2) **Object privileges** - This allows the user to **EXECUTE, SELECT, INSERT, UPDATE,** or **DELETE** data from database objects to which the privileges apply.

Few CREATE system privileges are listed below:

System Privileges	Description
CREATE object	allows users to create the specified object in their own schema.
CREATE ANY object	allows users to create the specified object in any schema.

The above rules also apply for ALTER and DROP system privileges.

Few of the object privileges are listed below:

Object Privileges	Description
INSERT	allows users to insert rows into a table.
SELECT	allows users to select data from a database object.

UPDATE	allows user to update data in a table.
EXECUTE	allows user to execute a stored procedure or a function.

Roles: Roles are a collection of privileges or access rights. When there are many users in a database it becomes difficult to grant or revoke privileges to users. Therefore, if you define roles, you can grant or revoke privileges to users, thereby automatically granting or revoking privileges. You can either create Roles or use the system roles pre-defined by oracle.

Some of the privileges granted to the system roles are as given below:

System Role	Privileges Granted to the Role
CONNECT	CREATE TABLE, CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE, CREATE SESSION etc.
RESOURCE	CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER etc. The primary usage of the RESOURCE role is to restrict access to database objects.
DBA	ALL SYSTEM PRIVILEGES

Creating Roles:

The Syntax to create a role is:

```
CREATE ROLE role_name
[IDENTIFIED BY password];
```

For Example: To create a role called "developer" with password as "pwd",the code will be as follows

```
CREATE ROLE testing
[IDENTIFIED BY pwd];
```

It's easier to GRANT or REVOKE privileges to the users through a role rather than assigning a privilege directly to every user. If a role is identified by a password, then, when you GRANT or REVOKE privileges to the role, you definitely have to identify it with the password.

We can GRANT or REVOKE privilege to a role as below.

For example: To grant CREATE TABLE privilege to a user by creating a testing role:

First, create a testing Role

```
CREATE ROLE testing
```

Second, grant a CREATE TABLE privilege to the ROLE testing. You can add more privileges to the ROLE.

```
GRANT CREATE TABLE TO testing;
```

Third, grant the role to a user.

```
GRANT testing TO user1;
```

To revoke a CREATE TABLE privilege from testing ROLE, you can write:

```
REVOKE CREATE TABLE FROM testing;
```

The Syntax to drop a role from the database is as below:

```
DROP ROLE role_name;
```

For example: To drop a role called developer, you can write:

```
DROP ROLE testing;
```

- Topic 3: Viewing privileges

How to Show All Oracle Database Privileges for a User?

Retrieving all user privileges within Oracle can range from a simple task using a basic SQL query to an advanced script; depending primarily on how involved the roles and privileges are configured within the server.

Querying DBA/USER Privilege Views

A database administrator (DBA) for Oracle can simply execute a query to view the rows in DBA_SYS_PRIVS, DBA_TAB_PRIVS, and DBA_ROLE_PRIVS to retrieve information about user privileges related to the system, tables, and roles, respectively.

For example, a DBA wishing to view all system privileges granted to all users would issue the following query:

SELECT * FROM DBA_SYS_PRIVS;

The DBA_SYS_PRIVS view contains three columns of data:

- GRANTEE is the name, role, or user that was assigned the privilege.
- PRIVILEGE is the privilege that is assigned.
- ADMIN_OPTION indicates if the granted privilege also includes the ADMIN option.

To determine which users have direct grant access to a table we'll use the DBA_TAB_PRIVS view:

SELECT * FROM DBA_TAB_PRIVS;

You can check the official documentation for more information about the columns returned from this query, but the critical columns are:

- GRANTEE is the name of the user with granted access.
- TABLE_NAME is the name of the object (table, index, sequence, etc).
- PRIVILEGE is the privilege assigned to the GRANTEE for the associated object.

Finally, querying the DBA_ROLE_PRIVS view has much of the same information but applicable to roles instead, where the GRANTED_ROLE column specifies the role in question:

SELECT * FROM DBA_ROLE_PRIVS;

QUERYING THE CURRENT USER'S PRIVILEGES

If DBA access isn't possible or necessary, it is also possible to slightly modify the above queries to view the privileges solely for the current user.

This is done by alternatively querying USER_ versions of the above DBA_ views. Thus, instead of looking at DBA_SYS_PRIVS we'd query USER_SYS_PRIVS, like so:

SELECT * FROM USER_SYS_PRIVS;

Since the USER_ privilege views are effectively the same as their DBA_ counterparts, but specific to the current user only, the type of returned data and column names are all identical to those when querying DBA_ views instead.

- Topic 4: Finding Information About User Privileges and Roles:

ALL_COL_PRIVS : Describes all column object grants for which the current user or PUBLIC is the object owner, grantor, or grantee

ALL_COL_PRIVS_MADE :Lists column object grants for which the current user is object owner or grantor.

ALL_COL_PRIVS_RECD : Describes column object grants for which the current user or PUBLIC is the

grantee

ALL_TAB_PRIVS : Lists the grants on objects where the user or PUBLIC is the grantee

ALL_TAB_PRIVS_MADE :Lists the all object grants made by the current user or made on the objects owned by the current user.

ALL_TAB_PRIVS_RECD :Lists object grants for which the user or PUBLIC is the grantee

DBA_COL_PRIVS :Describes all column object grants in the database

DBA_TAB_PRIVS :Lists all grants on all objects in the database

DBA_ROLES :This view lists all roles that exist in the database, including secure application roles. Note that it does not list the PUBLIC role.

DBA_ROLE_PRIVS :Lists roles granted to users and roles

DBA_SYS_PRIVS :Lists system privileges granted to users and roles

SESSION_ROLES :Lists all roles that are enabled for the current user. Note that it does not list the PUBLIC role.

ROLE_ROLE_PRIVS :This view describes roles granted to other roles. Information is provided only about roles to which the user has access.

ROLE_SYS_PRIVS :This view contains information about system privileges granted to roles. Information is provided only about roles to which the user has access.

ROLE_TAB_PRIVS :This view contains information about object privileges granted to roles. Information is provided only about roles to which the user has access.

USER_COL_PRIVS :Describes column object grants for which the current user is the object owner, grantor, or grantee

USER_COL_PRIVS_MADE: Describes column object grants for which the current user is the grantor

USER_COL_PRIVS_RECD :Describes column object grants for which the current user is the grantee

USER_ROLE_PRIVS :Lists roles granted to the current user

USER_TAB_PRIVS :Lists grants on all objects where the current user is the grantee

USER_SYS_PRIVS : Lists system privileges granted to the current user

USER_TAB_PRIVS_MADE: Lists grants on all objects owned by the current user

USER_TAB_PRIVS_RECD: Lists object grants for which the current user is the grantee

SESSION_PRIVS: Lists the privileges that are currently enabled for the user

SESSION_ROLES: Lists the roles that are currently enabled to the user.

LO 2.3 – Validate user accounts

- Topic 1: Set of database password policy

For users to access your database, you must create user accounts and grant appropriate database access privileges to those accounts. A user account is identified by a user name and defines the attributes of the user, including the following:

- Authentication method
- Password for database authentication
- Default tablespaces for permanent and temporary data storage
- Tablespace quotas
- Account status (locked or unlocked)
- Password status (expired or not)

When you create a user account, you must not only assign a user name, a password, and default tablespaces for the account, but you must also do the following:

- Grant the appropriate system privileges, object privileges, and roles to the account.
- If the user will be creating database objects, then give the user account a space usage quota on each tablespace in which the objects will be created.

Oracle recommends that you grant each user just enough privileges to perform his job, and no more. For example, a database application developer needs privileges to create and modify tables, indexes, views, and stored procedures, but does not need (and should not be granted) privileges to drop (delete) tablespaces or recover the database. You can create user accounts for database administration, and grant only a subset of administrative privileges to those accounts.

Criteria for a strong password

A strong password is one that is more secure by virtue of being difficult for a machine or a human to guess. Password strength can be achieved by incorporating the following characteristics; the more characteristics you incorporate into your password, the stronger it will be.

Characteristics of strong passwords

- At least 8 characters—the more characters, the better
- A mixture of both uppercase and lowercase letters
- A mixture of letters and numbers
- Inclusion of at least one special character, e.g., ! @ # ?]

  Note: do not use < or > in your password, as both can cause problems in Web browsers

Tips for keeping your password secure

- Change it regularly—once every three to six months.
- Change it if you have the slightest suspicion that the password has become known by a human or a machine.
- Never use it for other websites.
- Avoid typing it on computers that you do not trust; for example, in an Internet café.
- Never save it for a web form on a computer that you do not control or that is used by more than one person.
- Never tell it to anyone.
- Never write it down.

Account Locking

Account lockout keeps the account secure by preventing anyone or anything from guessing the username and password. When your account is locked, you must wait the set amount of time before being able to log into your account again.

- Topic 2: Description of Database User's Authentication methods

Authentication is the process of identifying users that request access to a system, network, or device. Access control often determines user identity according to credentials like username and password. Other authentication technologies like biometrics and authentication apps are also used to authenticate user identity.

5 Common Authentication Types

1. Password-based authentication

Passwords are the most common methods of authentication. Passwords can be in the form of a string of letters, numbers, or special characters. To protect yourself you need to create strong passwords that include a combination of all possible options.

2. Multi-factor authentication

Multi-Factor Authentication (MFA) is an authentication method that requires two or more independent ways to identify a user. Examples include codes generated from the user's smartphone, Captcha tests, fingerprints, or facial recognition.

3. Certificate-based authentication

Certificate-based authentication technologies identify users, machines or devices by using digital certificates. A digital certificate is an electronic document based on the idea of a driver's license or a passport.

4. Biometric authentication

Biometrics authentication is a security process that relies on the unique biological characteristics of an individual. Here are key advantages of using biometric authentication technologies:

- Biological characteristics can be easily compared to authorized features saved in a database.
- Biometric authentication can control physical access when installed on gates and doors.
- You can add biometrics into your multi-factor authentication process.

5. Token-based authentication

- Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange. You can then use the token to access protected systems instead of entering your credentials all over again. The digital token proves that you already have access permission. Use cases of token-based authentication include RESTful APIs that are used by multiple frameworks and clients.

Learning Unit 3 – Perform object security

LO 3.1 – Implement database security policies in line with object privileges and roles

- Topic 1: Introduction to data security policy

Managing passwords and resources using profiles

A profile is a set of resource limitations that can be assigned to a database user. Each Oracle database allows definition of a limitless number of profiles. They must be created and administered only if security policy requires that the use of database resources is limited. To use profiles, first we have to create types of similar user groups.

Profiles can be assigned to each user (using the CREATE USER command or ALTER USER) or we can define default profiles that are associated with all users who do not have a specific profile.

To create a profile is required the system privilege CREATE PROFILE. When we create a profile we can explicit the use limits of private resources or password parameters.

The Oracle can authenticate users using information stored in the database. The most important is the authentication information associated with a user password. This is encryptioned and stored in the data dictionary. The user can always change their own password.

To ensure confidentiality of passwords, encryption system allows their connections during network (client / server or server / server).

If this feature is enabled on both the client machine and the server system to encrypt passwords before sending them into the network, using a modified versiunde encryption algorithm DES (Data Encryption Standard).

If the user enters the wrong password for specified number of times the system locks the account associated with it. Depending on how the account was configured may be automatically unlocked after a specified period of time, or manually, by the database administrator.

After the profile was created, it can be associated to database users. It is not possible for a user to have multiple profiles simultaneously. If a profile is assigned to a user who already has one, the

new profile will replace the old one. Combination of profiles does not affect the current session. Profiles can be attributed only to users, and not to a roller or other profile.

Information about users and profiles

The system maintains a series of views in the data dictionary containing information about database users and profiles:

- DBA_TS_QUOTAS (describes the table space quotas for users); USER_PASSWORD_LIMITS (describes the parameters relating to passwords, set by CREATE PROFILE);
- DBA_PROFILES (lists all the profiles together with their limits); USER_RESOURCE_LIMITS (display resource limitations of the current user); RESOURCE_COST (displays the cost of each resource);
- V$SESSTAT (lists statistics about user sessions);
- V$STATNAM (displays the name of the statistics listed by previous view); PROXY_USERS (describes the database users who can assume the identity of other users), etc..

Managing privileges and roles

A privilege is the right to execute certain SQL commands. Privileges include the right to connect to the database, create tables, select lines of another user table, execution of stored procedures created by another user, etc.. Privileges should be granted only if users are absolutely necessary in such activity. Excessive granting of privileges can compromise the security of its database. Privileges can be system type or object type. System privileges can be classified as:

- System-specific privileges (eg CREATE SESSION, DROP TABLESPACE, ALTER TABLESPACE)
- Privileges for proper management of objects in any scheme (eg, CREATE ANY TABLE, DROP ANY INDEX). A role is a group of related privileges that can be granted or revoked simultaneously to users or other roles.

Using roles allows:

- To simplify administration privileges (rather than grant more privileges to one group of closely related users, we can create a role that contains all the necessary privileges and then grant the role each group member);
- Dynamic administration of privileges (if a user's group privileges must be changed, we will change their role that contains it and that will automatically be propagated to each user who is assigned to that role);

- Elective activation of the privileges (Roles can be selectively enabled or disabled so that it allows a high control of the privileges granted to users).

Roles have the following features:

- Users can be granted or revoked using the same commands as with system privileges;
- May include both system privileges and object privileges;
- Can be protected using passwords;
- Must have a unique name, different from user accounts and other roles in the database;
- There are not contained in any user scheme; their characteristics can be found in the data dictionary.

Oracle's Middleware Security

Multitier architectures have replaced the client-server applications in terms of preferred channel for access to applications and data processing. There are many reasons for this, including greater scalability, lower costs and more opportunities. Risks that arise when deploying applications on the Internet should not be overlooked. Such risks include limited knowledge of user identity, minimum control systems user behavior and the increased exposure of data to users malevolent attacks that exploit specific features open Internet, such as "worms", script of "cross-site", etc.

Web application developers have been forced to find solutions to such risks. About these security solutions in the middleware applications will be discussed in the following lines. Recent trends that multiply the possible risks that arise in web applications includes: deployment of several software applications through a single portal for business information, increasing the share of Java technology for web application development and the complexity and the need for scalability to run applications.

Through Oracle Application Server 10g, the Oracle provides a security framework for both internal components of OracleAS as well as third part applications running on OracleAS. OracleAS introduces the term Identity Management which provides support for the process of defining and managing user identity applications.

Security services provided by Oracle HTTP Server

Oracle HTTP Server extends Apache with a variety of standard optimizations or specific Oracle (the "modules" added the Apache server). These enhancements include the ability to allow / restrict

access to files and services based on user identity authentication standard established by operations through X.509 certificates and by IP address or hostname.

Another important feature of Oracle HTTP Server is the protection of data exchanged between client and server. This is provided by the SSL protocol, which ensures both data integrity and authentication of users and HTTP servers.

Although the Oracle HTTP Server is based on open source Apache Web server, it contains several enhancements that increase security access control. For example, Apache Server restrict access to directories with files with extension. htaccess. Processing these files is disabled by default in Oracle HTTP Server as file processing .htaccess requires both security issues and the decrease of performance. Oracle HTTP Server implements this type of access control through the modules / plug-ins that offer increased security and better performance for authorizing users to access resources.

Java Security in OracleAS

Java and Java 2 Enterprise Edition in particular has become the preferred development environment for many web applications. J2EE defines a security model type Java2 Security Model and a security framework known as the Java Authentication and Authorization Service (JAAS). OracleAS implements this framework through a J2EE compliant JAAS provider. JAAS Provider ensure accessibility to authentication services, authorization and delegation for developers of applications and allows integration of applications in J2EE environments.

OracleAS implementation for JAAS supports both the authorization information stored in OID and a simple implementation of the authorization API using XML as the encoding mechanism. This API allows Java applications to obtain information about users and roles through a secure mechanism from the operating system files.

OracleAS Portal - security features

Oracle 9iAS Portal is a key component of Oracle's product offerings in the category "enterprise portal". Web products in this class allow access to the newly formed business related information from internal networks of organizations. Although it was originally intended market for corporate portals, OracleAS can be configured to allow access to much larger communities, such as the Internet.

This portal allows users to manage applications and content published on the web and to structure the information logically. It also contains numerous tools to create users and keep track of those already existing, as well as their access to OracleAS Portal. OracleAS Portal provides a secure platform to integrate different applications into one portal, as well as effective management tools in this environment.

OracleAS Portal provides a consistent model for authorization based on individual and group privileges to give users access to applications and content of the portal. It also provides a flexible model to integrate applications into the portal login interface, allowing them to be classified as portlets, applications partner or external applications. OracleAS Portal also supports the type of security audit events via the event registration service.

Conclusions Security is a critical issue in the case of multitier applications. Oracle Middleware provides a solid framework for this type of web server applications using Oracle HTTP Server based on Apache, Oracle's J2EE framework and OracleAS Portal. Secure Application Server OracleAS starts from basic services, well-tested and easily configurable provided by Apache, add enhancements such single sign-on, authorization based on the OID and user management, security services and reaches Java2 security and Portal mechanisms to integrate applications. In addition, OracleAS supports a secure access to Oracle database using Oracle Advanced Security. These features ensure that the OracleAS Infrastructure is a smart choice for development and deployment of multitier applications in a secure environment.

- Topic 2: Elaboration of element of data security

Data security accountability

An **accountability**-based approach to **data protection** requires that organisations that collect, process or otherwise use personal **data** take **responsibility** for its **protection** and appropriate use beyond mere legal requirements, and are **accountable** for any misuse of the information that is in their care.

Accountability is one of the **data** protection principles - it makes you responsible for complying with the GDPR and says that you must be able to demonstrate your compliance. You need to put in place appropriate technical and organisational measures to meet the requirements of **accountability**.

System Data Security Policies

System Data Security Policies – The **security** configuration of all essential servers and operating **systems** is a critical piece of the **data security policy**. Rules regarding servers that run on the company's networks as well as the management of accounts and passwords must be clearly defined.

Policies that govern network services

A **network** security **policy** (NSP) is a generic document that outlines rules for computer **network** access, determines how **policies** are enforced and lays out some of the basic architecture of the company security/ **network** security environment.

The core elements of data security are confidentiality, integrity, and availability. Also known as the CIA triad, this is a security model and guide for organizations to keep their sensitive data protected from unauthorized access and data exfiltration.

- Confidentiality ensures that data is accessed only by authorized individuals;
- Integrity ensures that information is reliable as well as accurate; and
- Availability ensures that data is both available and accessible to satisfy business needs.

What are Data Security Considerations?

There are a few data security considerations you should have on your radar:

- Where is your sensitive data located? You won't know how to protect your data if you don't know where your sensitive data is stored.
- Who has access to your data? When users have unchecked access or infrequent permission reviews, it leaves organizations at risk of data abuse, theft or misuse. Knowing who has access to your company's data at all times is one of the most vital data security considerations to have.
- Have you implemented continuous monitoring and real-time alerting on your data? Continuous monitoring and real-time alerting are important not just to meet compliance regulations, but can detect unusual file activity, suspicious accounts, and computer behavior before it's too late.

Managing Patches

Patch management is the process that helps acquire, test and install multiple patches (code changes) on existing applications and software tools on a computer, enabling systems to stay

updated on existing patches and determining which patches are the appropriate ones. Managing patches thus becomes easy and simple.

Patch Management is mostly done by software companies as part of their internal efforts to fix problems with the different versions of software programs and also to help analyze existing software programs and detect any potential lack of security features or other upgrades.

Software patches help fix those problems that exist and are noticed only after the software's initial release. Patches mostly concern security while there are some patches that concern the specific functionality of programs as well.

How does an Automated Patch Management Solution Work?

- The automated patch management is used to automate the various stages of patching process.
- Scan the applications of devices for missing patches.
- Automate the downloading of missing patches that are released by the application vendors.
- Automated Patch Deployment ensures to automatically deploy patches based on the deployment policies, without any manual interference.
- Once the patches are deployed, reports on the status of the automated patch management tasks are updated.

With automated Patch Management solution, each enterprise is equipped to update its endpoints with latest patches irrespective of what OS they run and where they are located.

What is the Purpose of Patching?

Patching is a process to repair vulnerability or a flaw that is identified after the release of an application or software. Newly released patches can fix a bug or a security flaw, can help to enhance applications with new features, fix security vulnerability.

Unpatched software can make the device a vulnerable target of exploits. Patching a software as and when the patch is released is critical to deny malware access.

LO 3.2 – Implement data encryption in line with data protection requirements

- Topic 1: Elaboration of Type of Data Encryption.

Data encryption is a security method where information is encoded and can only be accessed or decrypted by a user with the correct encryption key. Encrypted data, also known as ciphertext, appears scrambled or unreadable to a person or entity accessing without permission.

Data encryption translates data into another form, or code, so that only people with access to a secret key (formally called a decryption key) or password can read it. Encrypted data is commonly referred to as ciphertext, while unencrypted data is called plaintext. Currently, encryption is one of the most popular and effective data security methods used by organizations. Two main types of data encryption exist - asymmetric encryption, also known as public-key encryption, and symmetric encryption.

How Data Encryption is Used

Data Encryption is used to deter malicious or negligent parties from accessing sensitive data. An important line of defense in cyber-security architecture, encryption makes using intercepted data as difficult as possible. It can be applied to all kinds of data protection needs ranging from classified government intel to personal credit card transactions. Data encryption software, also known as an encryption algorithm or cipher, is used to develop an encryption scheme which theoretically can only be broken with large amounts of computing power.

The purpose of data encryption is to protect digital data confidentiality as it is stored on computer systems and transmitted using the internet or other computer networks. The outdated data encryption standard (DES) has been replaced by modern encryption algorithms that play a critical role in the security of IT systems and communications.

- Topic 2: Classification of Encryption – Symmetric and Asymmetric

Encryption is often applied in two different forms, a symmetric key or an asymmetric key. A symmetric key, or secret key, uses one key to both encode and decode the information. This is best used for one to one sharing and smaller data sets. Asymmetric, or public key cryptography, uses two linked keys – one private and one public. The encryption key is public and can be used by anyone to encrypt. The opposite key is kept private and used to decrypt.

Symmetric vs. asymmetric encryption

Symmetric encryption: Symmetric encryption uses a single key to encrypt as well as decrypt data. The key needs to be shared with all authorized people.
Asymmetric encryption: Also called public key cryptography, asymmetric encryption uses two

separate keys—one public (shared with everyone) and one private (known only to the key's generator). The public key is used to encrypt the data and the private key helps to decrypt it.

The 4 common encryption types

There are different encryption methods based on the type of keys used, key length, and size of data blocks encrypted. Here we discuss some of the common encryption methods.

1. Advanced Encryption Standard (AES)

Advanced Encryption Standard is a symmetric encryption algorithm that encrypts fixed blocks of data (of 128 bits) at a time. The keys used to decipher the text can be 128-, 192-, or 256-bit long. The 256-bit key encrypts the data in 14 rounds, the 192-bit key in 12 rounds, and the 128-bit key in 10 rounds. Each round consists of several steps of substitution, transposition, mixing of plaintext, and more. AES encryption standards are the most commonly used encryption methods today, both for data at rest and data in transit.

2. Rivest-Shamir-Adleman (RSA)

Rivest-Shamir-Adleman is an asymmetric encryption algorithm that is based on the factorization of the product of two large prime numbers. Only someone with the knowledge of these numbers will be able to decode the message successfully. RSA is often used in digital signatures but works slower when large volumes of data need to be encrypted.

3. Triple Data Encryption Standard (TripleDES)

Triple Data Encryption Standard is a symmetric encryption and an advanced form of the DES method that encrypts blocks of data using a 56-bit key. TripleDES applies the DES cipher algorithm three times to each data block. TripleDES is commonly used to encrypt ATM PINs and UNIX passwords.

4. Twofish

Twofish is a license-free encryption method that ciphers data blocks of 128 bits. It's considered the successor to the Blowfish encryption method that ciphered message blocks of 64 bits. Twofish always encrypts data in 16 rounds regardless of the key size. Though it works slower than AES, the Twofish encryption method continues to be used by many file and folder encryption software solutions.

- Topic 3: Steps to implement an effective encryption strategy

Building and implementing an encryption strategy is a collaborative effort between your IT, operations, and management teams. Here are some steps that'll help you build an effective encryption strategy.

1. Classify data: You need to identify what data to encrypt. Understand and classify different types of data you transmit and store—card details, customer names and emails, company sales data, intellectual property data, and more—based on sensitivity, use, and regulatory impact.
2. Identify the right encryption tools: In most cases, you wouldn't need to implement a separate encryption software. Encryption features are present in commonly used apps and security tools such as email security, payment gateways, and cloud security software. But for encrypting databases or sensitive individual files, you might need separate encryption tools.

Encryption options in MEO file encryption software (Source)

3. Implement strong key management practices: If your keys fall into the wrong hands, your data security is at stake. You need to keep an inventory of all your encryption keys, along with information on who has access to them and how and when the keys have been used. Key management solutions help you to store and manage encryption keys.
4. Understand the limitations of encryption: Encryption does not help you prevent or detect cyberattacks. It only ensures that hackers will not be able to read your data. Hence, along with encrypting data, it is also important to implement other strong cybersecurity and intrusion detection measures such as anti-virus solutions and firewalls.

How does encryption work?

Encryption is the process of taking plain text, like a text message or email, and scrambling it into an unreadable format — called “cipher text.” This helps protect the confidentiality of digital data either stored on computer systems or transmitted through a network like the internet.

When the intended recipient accesses the message, the information is translated back to its original form. This is called decryption.

To unlock the message, both the sender and the recipient have to use a “secret” encryption key — a collection of algorithms that scramble and unscramble data back to a readable format.

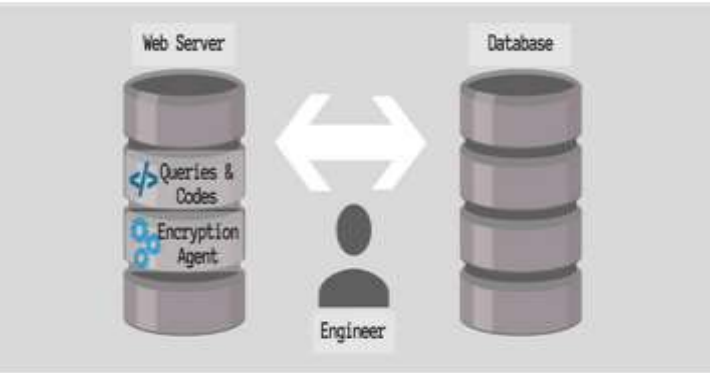
- Topic 4: Introduction to Database Encryption

Database encryption can generally be defined as a process that uses an algorithm to transform data stored in a database into "cipher text" that is incomprehensible without first being decrypted.

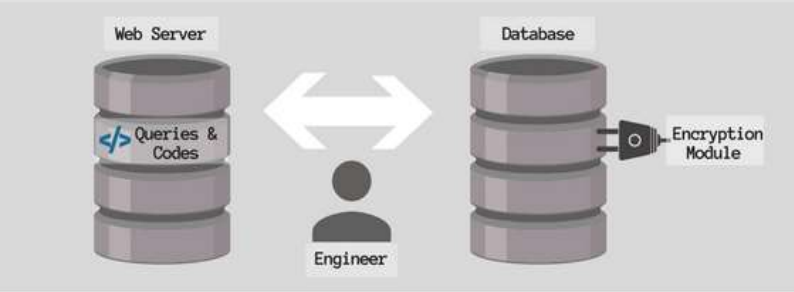
Common database encryption methods

It’s possible to encrypt data at a number of levels, from the application to the database engine. For an MSP considering how to help a customer choose an encryption method, it’s important to be clear on the purposes and requirements of these different encryption methods:

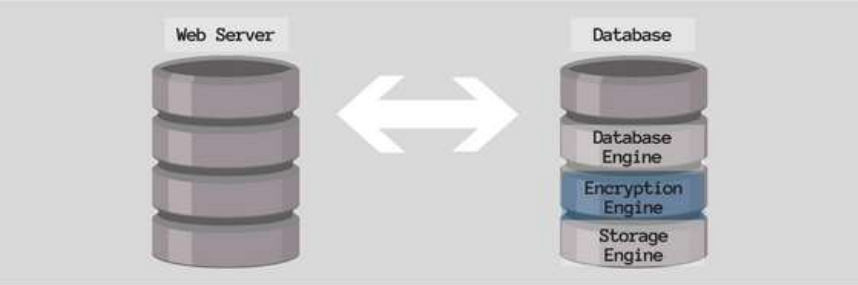
- **API Method**: This is application-level encryption that is appropriate across any database product (Oracle, MSSQL, etc). Queries within the encrypted columns are modified within the application, requiring hands-on work. If a business has an abundance of data, this can be a time-consuming approach. Additionally, encryption that functions at the application level can lead to increased performance issues.



- **Plug-In Method**: In this case, you'll attach an encryption module, or "package," onto the database management system. This method works independently of the application, requires less code management and modification, and is more flexible—you can apply this to both commercial and open-source databases. With this option, you will typically use column-level encryption.



- **TDE Method**: Transparent data encryption (TDE) executes encryption and decryption within the database engine itself. This method doesn't require code modification of the database or application and is easier for admins to manage. Since it's a particularly popular method of database encryption, TDE is explored in further detail below.



LO 3.3 – Restore data and Backup

- Topic 1: Introduction of data backup and restore

A backup is a copy of the information in a database, held in some physically separate location from your database. If the database becomes unavailable, perhaps because of damage to a disk drive, you can restore it from the backup. Depending on the nature of the damage, it is often possible to restore from backups all committed changes to the database up to the time it became unavailable.

Recovery happens when the operating system or database server crashes, or the database server does not shut down properly. The database server checks on database startup whether the database was shut down cleanly at the end of the previous session. If it was not, the server

executes an automatic recovery process to restore information. This mechanism recovers all changes up to the most recently committed transaction.

Purpose of Backup and Recovery

As a backup administrator, your principal duty is to devise, implement, and manage a backup and recovery strategy. In general, the purpose of a backup and recovery strategy is to protect the database against data loss and reconstruct the database after data loss. Typically, backup administration tasks include the following:

- Planning and testing responses to different kinds of failures
- Configuring the database environment for backup and recovery
- Setting up a backup schedule
- Monitoring the backup and recovery environment
- Troubleshooting backup problems
- Recovering from data loss if the need arises

As a backup administrator, you may also be asked to perform other duties that are related to backup and recovery:

- Data preservation, which involves creating a database copy for long-term storage
- Data transfer, which involves moving data from one database or one host to another

The purpose of this manual is to explain how to perform the preceding tasks.

Types of Backup

There are quite a number of backup types and terms used when it comes to backups of your digital content. This is a compilation of the most common types of backup with a brief explanation of their meaning, common examples, advantages and disadvantages of each backup type.

Full Backup

Full backup is a method of backup where all the files and folders selected for the backup will be backed up. When subsequent backups are run, the entire list of files and will be backed up again. The advantage of this backup is restores are fast and easy as the complete list of files are stored each time. The disadvantage is that each backup run is time consuming as the entire list of files is

copied again. Also, full backups take up a lot more storage space when compared to incremental or differential backups.

Incremental backup

Incremental backup is a backup of all changes made since the last backup. With incremental backups, one full backup is done first and subsequent backup runs are just the changes made since the last backup. The result is a much faster backup then a full backup for each backup run. Storage space used is much less than a full backup and less then with differential backups. Restores are slower than with a full backup and a differential backup.

Differential backup

Differential backup is a backup of all changes made since the last full backup. With differential backups, one full backup is done first and subsequent backup runs are the changes made since the last full backup. The result is a much faster backup then a full backup for each backup run. Storage space used is much less than a full backup but more then with Incremental backups. Restores are slower than with a full backup but usually faster then with Incremental backups.

Mirror Backup

Mirror backups are as the name suggests a mirror of the source being backed up. With mirror backups, when a file in the source is deleted, that file is eventually also deleted in the mirror backup. Because of this, mirror backups should be used with caution as a file that is deleted by accident or through a virus may also cause the mirror backups to be deleted as well.

Full PC Backup or Full Computer Backup

In this backup, it is not the individual files that are backed up but entire images of the hard drives of the computer that is backed up. With the full PC backup, you can restore the computer hard drives to its exact state when the backup was done. With the Full PC backup, not only can the work documents, picture, videos and audio files be restored but the operating system, hard ware drivers, system files, registry, programs, emails etc can also be restored.

Local Backup

Local backups are any kind of backup where the storage medium is kept close at hand or in the same building as the source. It could be a backup done on a second internal hard drive, an attached external hard drive, CD/ DVD –ROM or Network Attached Storage (NAS). Local backups protect

digital content from hard drive failures and virus attacks. They also provide protection from accidental mistakes or deletes. Since the backups are always close at hand they are fast and convenient to restore.

Offsite Backup

When the backup storage media is kept at a different geographic location from the source, this is known as an offsite backup. The backup may be done locally at first but once the storage medium is brought to another location, it becomes an offsite backup. Examples of offsite backup include taking the backup media or hard drive home, to another office building or to a bank safe deposit box.

Beside the same protection offered by local backups, offsite backups provide additional protection from theft, fire, floods and other natural disasters. Putting the backup media in the next room as the source would not be considered an offsite backup as the backup does not offer protection from theft, fire, floods and other natural disasters.

Online Backup

These are backups that are ongoing or done continuously or frequently to a storage medium that is always connected to the source being backed up. Typically the storage medium is located offsite and connected to the backup source by a network or Internet connection. It does not involve human intervention to plug in drives and storage media for backups to run. Many commercial data centres now offer this as a subscription service to consumers. The storage data centres are located away from the source being backed up and the data is sent from the source to the storage data centre securely over the Internet.

Remote Backup

Remote backups are a form of offsite backup with a difference being that you can access, restore or administer the backups while located at your source location or other location. You do not need to be physically present at the backup storage facility to access the backups. For example, putting your backup hard drive at your bank safe deposit box would not be considered a remote backup. You cannot administer it without making a trip to the bank. Online backups are usually considered remote backups as well.

Cloud Backup

This term is often used interchangeably with Online Backup and Remote Backup. It is where data is backed up to a service or storage facility connected over the Internet. With the proper login credentials, that backup can then be accessed or restored from any other computer with Internet Access.

FTP Backup

This is a kind of backup where the backup is done via FTP (File Transfer Protocol) over the Internet to an FTP Server. Typically the FTP Server is located in a commercial data centre away from the source data being backed up. When the FTP server is located at a different location, this is another form of offsite backup.

- Topic 2: Introduction to Backup in oracle database

A backup is a copy of data. This copy can include important parts of the database, such as the control file and datafiles. A backup is a safeguard against unexpected data loss and application errors. If you lose the original data, then you can reconstruct it by using a backup.

Backups are divided into physical backups and logical backups. Physical backups, which are the primary concern in a backup and recovery strategy, are copies of physical database files. You can make physical backups with either the Recovery Manager (RMAN) utility or operating system utilities. In contrast, logical backups contain logical data (for example, tables and stored procedures) extracted with an Oracle utility and stored in a binary file. You can use logical backups to supplement physical backups.

There are two ways to perform Oracle backup and recovery: Recovery Manager and user-managed backup and recovery.

Recovery Manager (RMAN) is an Oracle utility that can back up, restore, and recover database files. It is a feature of the Oracle database server and does not require separate installation.

You can also use operating system commands for backups and SQL*Plus for recovery. This method, also called user-managed backup and recovery, is fully supported by Oracle, although use of RMAN is highly recommended because it is more robust and greatly simplifies administration.

Whether you use RMAN or user-managed methods, you can supplement your physical backups with logical backups of schema objects made using the Export utility. The utility writes data from an Oracle database to binary operating system files. You can later use Import to restore this data into a database.

Disaster recovery plan

A disaster recovery plan (DRP) is a documented, structured approach that describes how an organization can quickly resume work after an unplanned incident. A DRP is an essential part of a business continuity plan (BCP). It is applied to the aspects of an organization that depend on a functioning IT infrastructure. A DRP aims to help an organization resolve data loss and recover system functionality so that it can perform in the aftermath of an incident, even if it operates at a minimal level.

Some types of disasters that organizations can plan for include:

- Application failure
- Communication failure
- Data center disaster
- Building disaster
- Campus disaster
- Citywide disaster
- Regional disaster
- National disaster
- Multinational disaster

Learning Unit 4 – Conduct database auditing

LO 4.1 – Describe auditing, types and records

- Topic 1: Introduction to database auditing overview

Actions

Audit Actions allow you to **audit** tables and stored procedures. **Audit actions** mandate that table row commands (select, insert, update, delete) and a few other commands be **audited** on specified **database** objects when performed by specified users or roles.

Oracle Database Auditing.

Auditing is the monitoring and recording of selected user database actions. It can be based on individual actions, such as the type of SQL statement executed, or on combinations of factors that can include user name, application, time, and so on. Security policies can trigger auditing when specified elements in an Oracle database are accessed or altered, including the contents within a specified object.

Auditing is always about accountability, and is frequently done to protect and preserve privacy for the information stored in databases. Concern about privacy policies and practices has been rising steadily with the ubiquitous use of databases in businesses and on the Internet. Oracle Database provides a depth of auditing that readily enables system administrators to implement enhanced protections, early detection of suspicious activities, and finely-tuned security responses.

Auditing is typically used to:

- Enable future accountability for current actions taken in a particular schema, table, or row, or affecting specific content
- Deter users (or others) from inappropriate actions based on that accountability
- Investigate suspicious activity

  For example, if some user is deleting data from tables, then the security administrator might decide to audit all connections to the database and all successful and unsuccessful deletions of rows from all tables in the database.
- Notify an auditor that an unauthorized user is manipulating or deleting data and that the user has more privileges than expected which can lead to reassessing user authorizations
- Monitor and gather data about specific database activities

For example, the database administrator can gather statistics about which tables are being updated, how many logical I/Os are performed, or how many concurrent users connect at peak times.

- Detect problems with an authorization or access control implementation

For example, you can create audit policies that you expect will never generate an audit record because the data is protected in other ways. However, if these policies do generate audit records, then you will know the other security controls are not properly implemented.

Auditing Types and Records

Oracle allows audit options to be focused or broad, enabling you to audit the following:

- Successful statement executions, unsuccessful statement executions, or both
- Statement executions once in each user session or once every time the statement is executed
- Activities of all users or of a specific user

Auditing Types and Descriptions

Type of Auditing (link to discussion)	Meaning/Description
Statement Auditing	Enables you to audit SQL statements by type of statement, not by the specific schema objects on which they operate. Typically broad, statement auditing audits the use of several types of related actions for each option. For example, `AUDIT TABLE` tracks several DDL statements regardless of the table on which they are issued. You can also set statement auditing to audit selected users or every user in the database.
Privilege Auditing	Enables you to audit the use of powerful system privileges that enable corresponding actions, such as `AUDIT CREATE TABLE`. Privilege auditing is more focused than statement auditing, which audits only a particular type of action. You can set privilege auditing to audit a selected user or every user in the database.

Schema Object Auditing	such as AUDIT SELECT ON employees. Schema object auditing is very focused, auditing only a single specified type of statement (such as SELECT) on a specified schema object. Schema object auditing always applies to all users of the database.
Fine-Grained Auditing	based on content, using any Boolean measure, such as value > 1,000,000. Enables auditing based on access to or changes in a column.

- Topic 2: Elaboration of Audit Records and Audit Trails

Audit records include information about the operation that was audited, the user performing the operation, and the date and time of the operation. Audit records can be stored in either a data dictionary table, called the database audit trail, or in operating system files, called an operating system audit trail.

The two general types of auditing are standard auditing, which is based on privileges, schemas, objects, and statements, and fine-grained auditing. Standard audit records can be written either to DBA_AUDIT_TRAIL (the sys.aud$ table) or to the operating system. Fine-grained audit records are written to DBA_FGA_AUDIT_TRAIL (the sys.fga_log$ table) and the DBA_COMMON_AUDIT_TRAIL view, which combines standard and fine-grained audit log records.

The following subsections describe these trails and records:

- Database Audit Trail (DBA_AUDIT_TRAIL)
- Operating System Audit Trail
- Syslog Audit Trail
- Operating System and Syslog Audit Records
- Records Always in the Operating System and Syslog Audit Trail

Database Audit Trail (DBA_AUDIT_TRAIL)

The database audit trail consists of a single table named SYS.AUD$ in the SYS schema of the data dictionary of each Oracle database. Several predefined views are provided to help you use the information in this table, such as DBA_AUDIT_TRAIL.

Audit trail records can contain different types of information, depending on the events audited and the auditing options set. The partial list in Table 8-2 shows columns that always appear in the audit trail. If the data they represent is available, then that data populates the corresponding column. (For certain columns, this list shows the column name displayed in the audit record in parentheses.)

If the database destination for audit records becomes full or is unavailable, and is therefore unable to accept new records, then an audited action cannot complete. Instead, it causes an error message and is not done. In some cases, an operating system log allows such an action to complete.

Operating System Audit Trail

Oracle Database allows audit trail records to be directed to an operating system audit trail if the operating system makes such an audit trail available to Oracle Database. If not, then audit records are written to a file outside the database. The target directory varies by platform: on the Solaris platform, it is $ORACLE_HOME/rdbms/audit, but for other platforms you must check the platform documentation to learn the correct target directory. In Windows, the information is accessed through Event Viewer.

If init.ora specifies AUDIT_TRAIL=XML, then audit records are written to the operating system as XML files. A new dynamic view, V$XML_AUDIT_TRAIL, makes such XML audit records available to DBAs through a SQL query, providing enhanced usability. Querying this view causes all XML files (all files with a .xml extension) in the AUDIT_FILE_DEST directory to be parsed and presented in relational table format. Because XML is a standard document format, many utilities are available to parse and analyze such XML data.

Syslog Audit Trail

One potential security vulnerability for an operating system audit trail is that a privileged user, such as a DBA, can modify or delete audit records. In order to minimize this risk, you can use a syslog audit trail. Syslog is a standard protocol on UNIX-based systems for logging information from different components of a network. Applications call the syslog() function to log information to the syslog daemon, which then determines where to log the information. You can configure syslog to log information to a file name syslog.conf, to the console, or to a remote, dedicated log host. You can also configure syslog to alert a specified set of users when information is logged.

Because applications, such as an Oracle process, use the syslog() function to log information to the syslog daemon, a privileged user does not need to have permissions to the file system where messages are logged. For this reason, audit records stored using a syslog audit trail can be more secure than audit records stored using an operating system audit trail. In addition to restricting permissions to a file system for a privileged user, for a syslog audit trail to be secure, neither privileged users nor the Oracle process should have root access to the system where the audit records are written.

Operating System and Syslog Audit Records

The operating system and syslog audit trails are encoded, but are decoded in data dictionary files and error messages. The following fields are included:

- Action code describes the operation performed or attempted. The AUDIT_ACTIONS data dictionary table contains a list of these codes and their descriptions.
- Privileges used describes any system privileges used to perform the operation. The SYSTEM_PRIVILEGE_MAP table lists all of these codes and their descriptions.
- Completion code describes the result of the attempted operation. Successful operations return a value of zero, and unsuccessful operations return the Oracle error code describing why the operation was unsuccessful.

Records Always in the Operating System and Syslog Audit Trail

Some database-related actions are always recorded into the operating system and syslog audit trails regardless of whether database auditing is enabled. The fact that these records are always created is sometimes referred to as mandatory auditing. The following actions are recorded:

- At instance startup, an audit record is generated that includes the operating system user starting the instance, the terminal identifier of the user, and the date and time stamp. This information is recorded into the operating system or syslog audit trails, because the database audit trail is not available until after startup has successfully completed.
- At instance shutdown, an audit record is generated that details the operating system user shutting down the instance, the terminal identifier of the user, and the date and time stamp.
- During connections made with administrator privileges, an audit record is generated that details the operating system user connecting to Oracle Database with administrator

privileges. This record provides accountability regarding users connected with administrator privileges.

On operating systems that do not make an audit trail accessible to Oracle Database, these audit trail records are placed in an Oracle audit trail file in the same directory as background process trace files, and in a similar format.

When Are Audit Records Created?

Standard auditing for the entire database is either enabled or disabled by the security administrator. If it is disabled, then no audit records are created

If database auditing is enabled by the security administrator, then individual audit options become effective. These audit options can be set by any authorized database user for database objects he owns.
When auditing is enabled in the database and an action set to be audited occurs, an audit record is generated during the execute phase of the statement.
SQL statements inside PL/SQL program units are individually audited, as necessary, when the program unit is executed.

The generation and insertion of an audit trail record is independent of a user transaction being committed. That is, even if a user transaction is rolled back, the audit trail record remains committed.

Statement and privilege audit options in effect at the time a database user connects to the database remain in effect for the duration of the session. Setting or changing statement or privilege audit options in a session does not take effect in that session. The modified statement or privilege audit options take effect only when the current session ends and a new session is created.

In contrast, changes to schema object audit options become effective for current sessions immediately.

Statement Auditing

Statement auditing is the selective auditing of related groups of statements regarding a particular type of database structure or schema object, but not a specifically named structure or schema object. These statements fall into the following categories:

- DDL statements: As an example, AUDIT TABLE audits all CREATE and DROP TABLE statements
- DML statements: As an example, AUDIT SELECT TABLE audits
  all SELECT ... FROM TABLE/VIEW statements, regardless of the table or view

Statement auditing can be broad or focused, for example, by auditing the activities of all database users or of only a select list.

Privilege Auditing

Privilege auditing audits statements that use a system privilege, such as SELECT ANY TABLE. For example, when AUDIT SELECT ANY TABLE is in force, all statements issued by users with the SELECT ANY TABLE privilege are audited.

You can audit the use of any system privilege. Like statement auditing, privilege auditing can audit the activities of all database users or of only a specified list.

If similar statement and privilege audit options are both set, then only a single audit record is generated. For example, if the statement clause TABLE and the system privilege CREATE TABLE are both audited, then only a single audit record is generated each time a table is created.

Thus privilege auditing does not occur if the action is already permitted by the existing owner and schema object privileges. Privilege auditing is triggered only if they are insufficient, that is, only if what makes the action possible is a system privilege.

Privilege auditing is more focused than statement auditing, because each privilege auditing option audits only specific types of statements, not a related list of statements. For example, the statement auditing clause, TABLE, audits CREATE TABLE, ALTER TABLE, and DROP TABLE statements. However, the privilege auditing option, CREATE TABLE, audits only CREATE TABLE statements, because only the CREATE TABLE statement requires the CREATE TABLE privilege.

Schema Object Auditing

Schema object auditing can audit all SELECT and DML statements permitted by schema object privileges, such as SELECT or DELETE statements on a given table.
The GRANT and REVOKE statements that control those privileges are also audited.

You can audit statements that reference tables, views, sequences, standalone stored procedures or functions, and packages, but not individual procedures within packages. Further discussion appears in the next section, entitled Schema Object Audit Options for Views, Procedures, and Other Elements.

Statements that reference clusters, database links, indexes, or synonyms are not audited directly. However, you can indirectly audit access to these schema objects, by auditing the operations that affect the base table.

Schema object audit options are always set for all users of the database. These options cannot be set for a specific list of users. You can set default schema object audit options for all auditable schema objects.

Fine-Grained Auditing

Fine-Grained Auditing (FGA) enables you to monitor data access based on content. A built-in audit mechanism in the database prevents users from bypassing the audit.

While Oracle Database triggers can potentially monitor DML actions such as INSERT, UPDATE, and DELETE, monitoring SELECT statements can be costly. In some cases, a trigger may audit too much, and in others, its effectiveness or completeness may be uncertain. Triggers also do not enable users to define their own alert action in response to a triggered audit, beyond simply inserting an audit record into the audit trail.

FGA provides an extensible interface for creating policies to audit SELECT and DML statements on tables and views. The DBMS_FGA package administers these value-based audit policies. Using DBMS_FGA, the security administrator creates an audit policy on the target object. If any rows returned from a query match the audit condition, then an audit event entry is inserted into the fine-grained audit trail. This entry includes all the information reported in the regular audit trail. See the Audit Records and Audit Trails section. Only one row of audit information is inserted into the audit trail for every FGA policy that evaluates to true. The extensibility framework in FGA also enables administrators optionally to define an appropriate audit event handler to process the event, for example by sending an alert page to the administrator.

The administrator uses the DBMS_FGA.ADD_POLICY interface to define each FGA policy for a table or view, identifying any combination of SELECT, UPDATE, DELETE, or INSERT statements.

Learning Outcome 4.2: Complete inspection of database security standards

- Topic 1: Introduction of database security standards

Database Hardening Best Practices

This checklist was developed by IST system administrators to provide guidance for securing databases storing sensitive or protected data. Implementing these security controls will help to prevent data loss, leakage, or unauthorized access to your databases.

Physical Database Server Security

- The physical machine hosting a database is housed in a secured, locked and monitored environment to prevent unauthorized entry, access or theft.
- Application and web servers are not hosted on the same machine as the database server.

Firewalls for Database Servers

- The database server is located behind a firewall with default rules to deny all traffic.
- The database server firewall is opened only to specific application or web servers, and firewall rules do not allow direct client access. If the development environment cannot meet this requirement, then protected data is not stored in the development database server and mock data is made up for development. Data obfuscation of production data is not sufficient.
- Firewall rule change control procedures are in place and notification of rule changes are distributed to System Administrators (SAs) and Database Administrators (DBAs).
- Firewall rules for database servers are maintained and reviewed on a regular basis by SAs and DBAs. If using the IST provided firewall service, the rules are also regularly reviewed by the Information Security Office (ISO).
- Regularly test machine hardening and firewall rules via network scans, or by allowing ISO scans through the firewall.

Database Software

- The database software version is currently supported by the vendor or open source project, as required by the campus minimum security standards.
- All unused or unnecessary services or functions of the database are removed or turned off.

- Unneeded default accounts are removed, or else passwords are changed from defaults.
- Null passwords are not used, and temporary files from the install process that may contain passwords are removed.
- Database software is patched to include all current security patches. Provisions are made to maintain security patch levels in a timely fashion.

Application / Web Servers / Application Code

- Destination systems (application/web servers) receiving protected data are secured in a manner commensurate with the security measures on the originating system. All servers and clients meet minimum security standards.
- All servers, applications and tools that access the database are documented.
- Configuration files and source code are locked down and only accessible to required OS accounts.
- Application code is reviewed for SQL injection vulnerabilities.
- No "Spyware" is allowed on the application, web or database servers.

User/Client Workstations

- If users are allowed protected data on their workstations, then client workstations meet the minimum security standards.
- If users are allowed protected data on their workstations, then the workstation is protected against unauthorized access to a session by deploying screen savers. Users understand the requirement to lock their workstations when leaving the station.
- If users are allowed protected data on their workstations, then the workstation should require an individual login and password.
- If users are allowed protected data on their workstations, then protected data on the client workstation is encrypted by the workstation's operating system.
- Protected data is not stored on transportable devices.
- Protected data is never sent via email, either in the body or as an attachment, by either users or as an automated part of the system.
- Protected data that is no longer needed is routinely deleted.

- If users are allowed protected data on their workstations, then no "Spyware" is allowed on the client workstations.

Administrator Accounts / Permissions / Passwords

- DBAs understand their responsibility for reviewing all requested script and database changes to ensure the security of the system is not compromised.
- Accounts with system administration capabilities are provided to as few individuals as is practical, and only as needed to support the application.
- All Developers, Vendors, SAs, DBAs & Contractors have signed a non-disclosure agreement.
- All developers, SAs, DBAs and contractors have passed a criminal background check if required by the background check policy.
- Operating system accounts used by DBA staff to login to dataserver machines for administrative duties are individual accounts, and not a shared group account.
    - When possible, the daemon OS account that is required to run the dataserver process does not allow a direct login.
    - Instead, individual OS accounts are used to login, then sudo or su to the daemon account (for UNIX) or disallow desktop login (Windows).
- Database accounts used by DBA staff for administrative duties are individual accounts, and not a shared group account.
    - A group account is permitted for running automated DBA maintenance and monitoring jobs, such as backups.
    - This group account is not used for daily interactive tasks by the DBA group, except when required to troubleshoot maintenance and monitoring jobs.
- Passwords for all DBA operating system accounts and database accounts are strong passwords, and are changed when administrators/contractors leave positions.
- If the DBA and developer roles are being filled by a single person, changes are approved by the Data Proprietor.

Learning Outcome 4.3: Efficient production of auditing report

- Topic 1: Common Vulnerabilities Found in Database Attacks

Phishing: Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity,

dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information. An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft.

SQL injection: SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.

Data exfiltration: is the unauthorized transfer of sensitive information from a target's network to a location which a threat actor controls. Data exfiltration is any unauthorized movement of data. It can also be known as data exfil, data exportation, data extrusion, data leakage and data theft. Whether information is stolen with a printer or a thumb drive, data exfil is a very real threat for organizations. Attacks can be conducted manually by an authorized employee with access to company systems or through external malicious actors who have gained access.

Staging server : is a type of server that is used to test a software, website or service in a production-similar environment before being set live. It is part of a staging environment or staging site, where it serves as a temporary hosting and testing server for any new software or websites.

Data redaction: Data Redaction or Dynamic Data Masking is the process of obfuscating or hiding sensitive data elements such as Credit Card Numbers in the SQL query results prior to display by applications. ... Sensitive data is redacted on-the-fly without changing actual data stored in the database.

Data masking: Data masking (also known as data scrambling and data anonymization) is the process of replacing sensitive information copied from production databases to test non-production databases with realistic, but scrubbed, data based on masking rules.

Data encryption: In Encryption, humans scramble data to make it unreadable. It uses an algorithm and a key to transform a plaintext into an encrypted text (also known as ciphertext). This protects your data from unauthorized access or a potential attacker.

Reference(s):

1. Learn PHP, MYSQL, JavaScript-with jQuery, CSS &HTML5 4 Edition-By Robin Nixon
2. HTML, XHTML, & CSS ALL-IN-ONE for dummies 2nd Edition – By Andy Harris
3. How to Do Everything with PHP and MySQL (McGraw-Hill, 2005, 0-07-146654-1) - by VikranWasani
4. https://www.formget.com/update-data-in-database-using-php/
5. https://www.tutorialspoint.com/php/php_validation_example.htm
6. https://www.w3schools.com/php/php_mysql_prepared_statements.asp
7. https://www.oracle.com/database/technologies/database12c-win64-downloads.html
8. https://tuto-computer.com/oracle-tutorial-pdf.html/amp
9. https://www.google.com/url?sa=t&source=web&rct=j&url=https://docs.oracle.com/cd/B28359_01/appdev.111/b31231.pdf&ved=2ahUKEwiom5-wk_TpAhViRxUIHWnOBJMQFjABegQIAhAI&usg=AOvVaw1K7DgV5ky8roBYLirTZNH7&cshid=1591686270498