Sri Lanka Institute of Information Technology

BSc Honors in Information Technology Specializing in Cyber Security

IE3062- Data and Operating Systems Security

Group Assignment
Database and OS Security

Submitted by: Group 54

Student Name	Student ID
M.V.G.R.S. Gunasekara (Leader)	IT21162596
S.M.J.Y Wickramasinghe	IT21076534
Perera U.L.S. A	IT21278976
W.M.M. Gunasekara	IT21226496

Date: 26.10.2023

Contents

Task 1

1. Linux distributions

Five well-known Linux distributions are compared in this comparative analysis: Ubuntu, CentOS, Debian, Fedora, and Oracle Linux. It gives information on their user base, release cycles, package management systems, community support, stability, and security upgrades, assisting users in selecting the best Linux distribution.

1. **Ubuntu:**
    - **Audience:** General users, servers, and desktops.
    - **Release Cycle:** Regular releases every six months with Long Term Support (LTS) versions every two years.
    - **Package Management:** Uses Debian's APT (Advanced Package Tool).
    - **Community Support:** Strong community support and a wide user base.
    - **Stability:** LTS releases are known for their stability.
    - **Security Updates:** Prompt and reliable security updates.
2. **CentOS:**
    - **Audience:** Enterprises, data centers, and servers.
    - **Release Cycle:** Stable and conservative, based on RHEL releases.
    - **Package Management:** Uses RPM (Red Hat Package Manager) and YUM (Yellowdog Updater, Modified).
    - **Community Support:** Strong community support, but the support model has shifted with CentOS Stream.
    - **Stability:** Known for its stability and long support life.
    - **Security Updates:** Reliable security updates, following Red Hat's security practices.

3. **Debian:**
    - **Audience:** General users, servers, and developers.
    - **Release Cycle:** Relatively slow-release cycle with a strong focus on stability.
    - **Package Management:** Uses APT with DEB packages.
    - **Community Support:** Strong and diverse community support.
    - **Stability:** Renowned for its stability, making it ideal for servers.
    - **Security Updates:** Reliable and prompt security updates.
4. **Fedora:**
    - **Audience:** Developers and enthusiasts.
    - **Release Cycle:** Frequent releases with new features and technologies.
    - **Package Management:** Uses DNF (Dandified YUM) and RPM.
    - **Community Support:** Active and responsive community support.
    - **Stability:** Known for being more bleeding-edge, not as stable as some other distributions.
    - **Security Updates:** Timely security updates.
5. **Oracle Linux:**
    - **Audience:** Enterprises, particularly when using Oracle database products.
    - **Release Cycle:** Stable and conservative, following the Red Hat Enterprise Linux (RHEL) model.
    - **Package Management:** Uses RPM and YUM.
    - **Community Support:** Strong community support, but with a shift towards Oracle's CentOS Stream.
    - **Stability:** Known for stability, especially when used in Oracle environments.

- **Security Updates:** Reliable security updates, aligned with Oracle's security practices [1].

2. Selection and Justification:

Oracle Linux is a suitable choice for setting up a new server for running a HR system database for several reasons.

1. **Optimized for Database Workloads:** When used to run Oracle databases, Oracle Linux is expertly designed to ensure seamless interaction with Oracle database solutions, enhancing performance and guaranteeing consistent stability.

2. **Direct Support from Oracle:** Oracle Linux, created and maintained by Oracle Corporation, guarantees regular updates, software compatibility, and single-vendor responsibility for support.

3. **Security Emphasis:** Oracle Linux benefits from Oracle Corporation's ongoing dedication to security when it is installed on a server hosting sensitive HR data. This includes regular updates and strong features created to protect against known vulnerabilities.

4. **Certification for Oracle Software:** A reliable database server is guaranteed by Oracle Linux certification, which guarantees compatibility, performance, and comprehensive testing for its database and software solutions.

5. **Audit and Compliance Support:** Oracle Linux, when combined with Oracle database products, can streamline compliance efforts for HR systems, enhancing security and auditing standards.

6. **Community and Documentation:** Oracle Linux boasts an active user community and extensive official documentation, providing valuable resources for troubleshooting, best practices, and security guidance.

In conclusion, Oracle Linux is a good option for hosting the database server for an HR system because of its compatibility, support, and security emphasis.

Analyze security threats and choose 10 settings to harden in order to safeguard Oracle Linux for the HR database system server.

3. Security Risks:

- **Configurations**
  1. Unauthorized Access: Unauthorized users gaining access to sensitive HR data.
  2. Data Breach: Unauthorized disclosure, alteration, or destruction of HR data.
  3. Denial of Service (DoS) Attacks: Attackers disrupting system availability.
  4. Insider Threats: Employees or authorized users with malicious intent.
  5. Malware and Vulnerabilities: Exploitation of software vulnerabilities or malware infection.
  6. Weak Authentication: Weak or default passwords, or insufficient user authentication.
  7. Unpatched Software: Failing to apply security updates promptly.
  8. Inadequate Network Security: Weak firewall and network security settings.
  9. Data Loss: Inadvertent data loss due to misconfigurations or errors.
  10. Lack of Monitoring: Insufficient monitoring of system activities and potential security incidents.

- **Configurations to Harden Oracle Linux:**
  1. User Access Control:
     - Configuration: Implement proper user access control with the principle of least privilege. Use strong, unique passwords, and consider multi-factor authentication (MFA).
     - Security Benefits: Reduces the risk of unauthorized access and data breaches.

  2. Firewall Configuration:
     - Configuration: Configure a firewall to restrict incoming and outgoing network traffic, allowing only necessary ports and services.
     - Security Benefits: Mitigate unauthorized network access and DoS attacks.

3. SELinux/AppArmor:
    - Configuration: Enable Security-Enhanced Linux (SELinux) or AppArmor to confine processes and protect system resources.
    - Security Benefits: Isolates and limits the impact of security breaches and malware.

4. Regular Patch Management:
    - Configuration: Establish a robust process for regularly applying OS and software updates and patches.
    - Security Benefits: Addresses known vulnerabilities and reduces the risk of exploitation.

5. Intrusion Detection System (IDS):
    - Configuration: Deploy an IDS to monitor network traffic and detect suspicious activities or potential security breaches.
    - Security Benefits: Provides early detection of security incidents.

6. Strong Audit Policies:
    - Configuration: Implement comprehensive audit policies to record system and user activities. Regularly review audit logs.
    - Security Benefits: Helps in identifying security issues, detecting insider threats, and meeting compliance requirements.

7. Data Encryption:
    - Configuration: Encryption is crucial for both rest and transit data, particularly for sensitive HR data.
    - Security Benefits: Protects data from unauthorized access and breaches.

8. Backup and Disaster Recovery:

- Configuration: Implement regular backups and disaster recovery procedures to ensure data availability and integrity.
- Security Benefits: Mitigates data loss and system downtime.

9. Access Control Lists (ACLs):
    - Configuration: Implement access control lists to control file and directory access.
    - Security Benefits: Restricts unauthorized access to sensitive HR data.

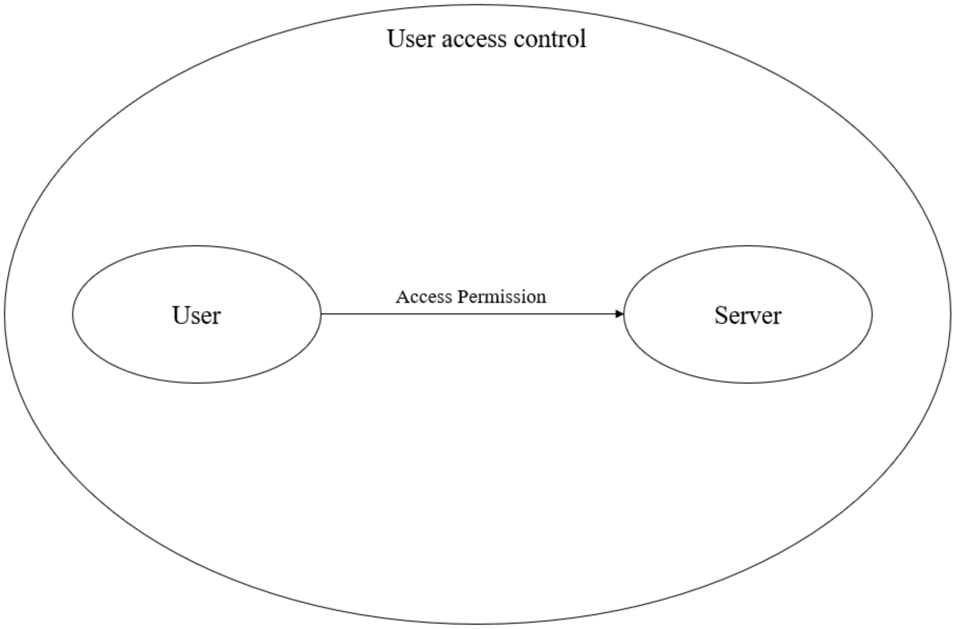
10. Security Monitoring and Incident Response:
    - Configuration: Implement continuous security monitoring and create an incident response plan.
    - Security Benefits: Improves the ability to detect and respond to security incidents.

Regular updates and monitoring are essential for maintaining Oracle Linux server security, mitigating risks like unauthorized access, data breaches, and malware [2].

4. Diagrams of the configurations

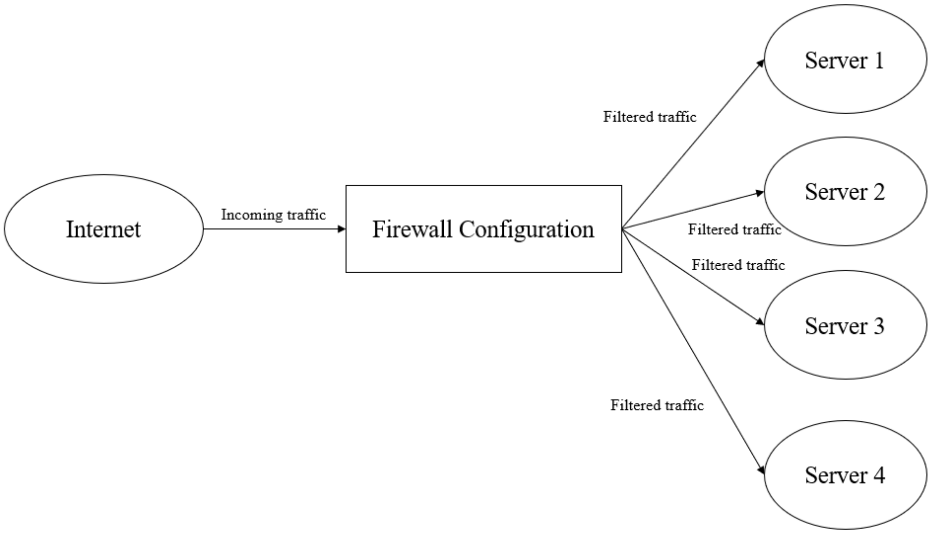
1. User Access Control

User Access Control is a crucial security mechanism that manages user permissions and rights, ensuring data confidentiality, integrity, and availability by controlling access and actions.



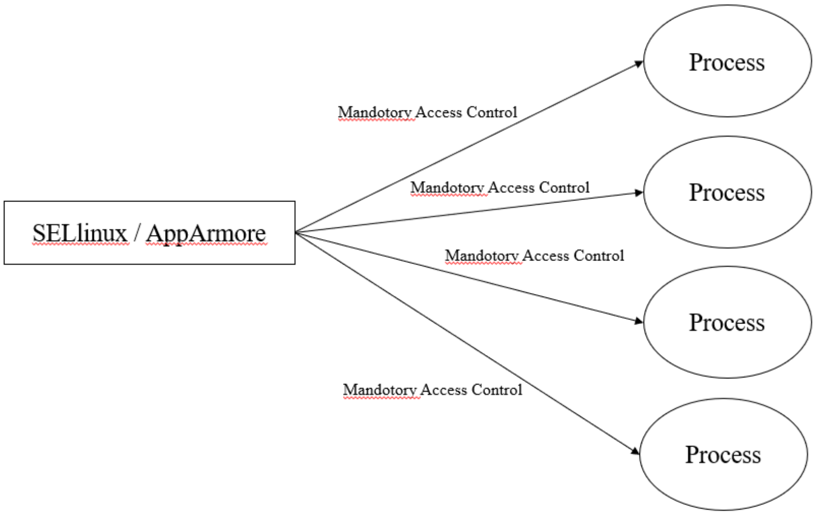
2. Firewall Configuration

Firewall configuration is a crucial network security measure that establishes a firewall to safeguard against unauthorized access and threats from untrusted networks, regulating incoming and outgoing traffic.



3. SELinux/AppArmor

SELinux and AppArmor are security frameworks in Linux-based operating systems that enforce access control policies, providing fine-grained permissions and restrictions for processes and applications.



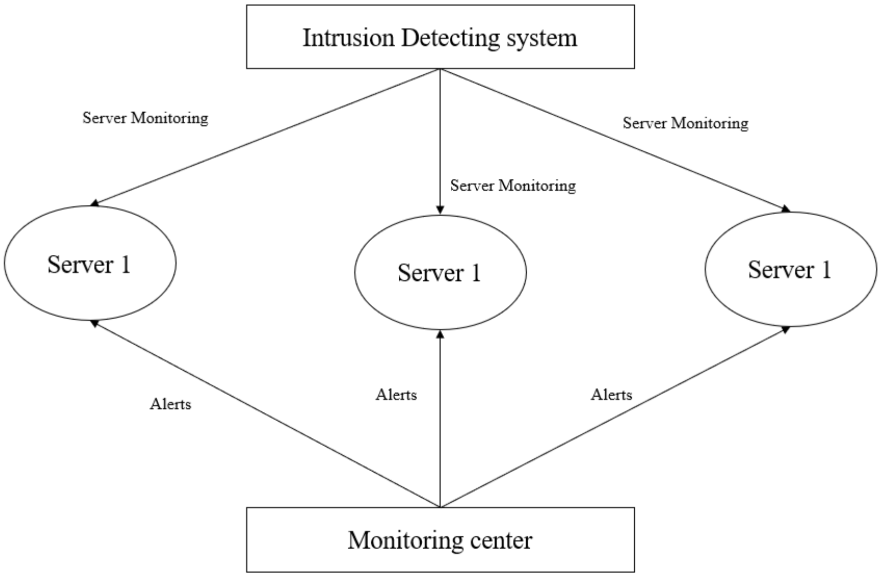
4. Regular Patch Management

Regular patch management is a systematic process of identifying, testing, and applying updates to software, operating systems, and applications to improve security, functionality, and system performance.

Server 1
Regular Updates
Patch Management
Regular Updates
Server 2

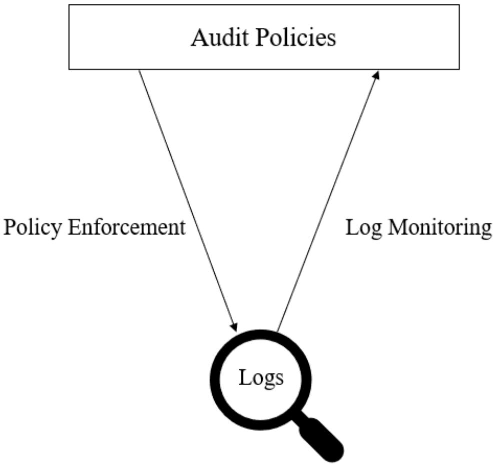
5. Intrusion Detection System (IDS)

An IDS is a vital security tool that monitors, analyzes, and identifies potential security threats, proactively alerting administrators to suspicious or malicious activity.



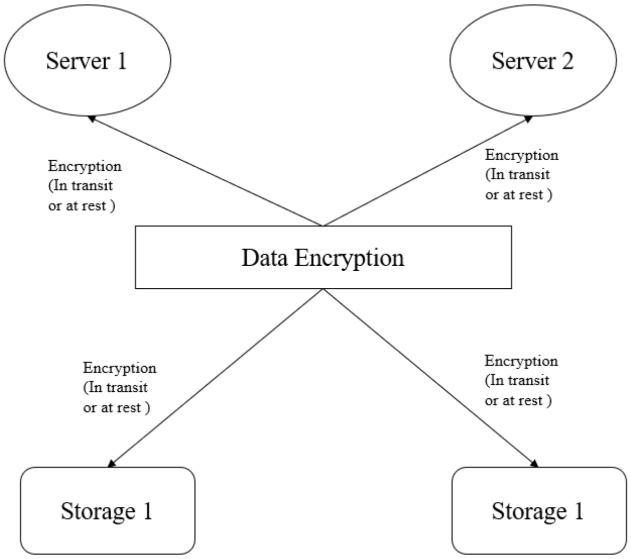
6. Strong Audit Policies

Strong Audit Policies are rules and configurations that govern the generation, collection, and storage of audit logs in an organization's computer systems, servers, and network devices.



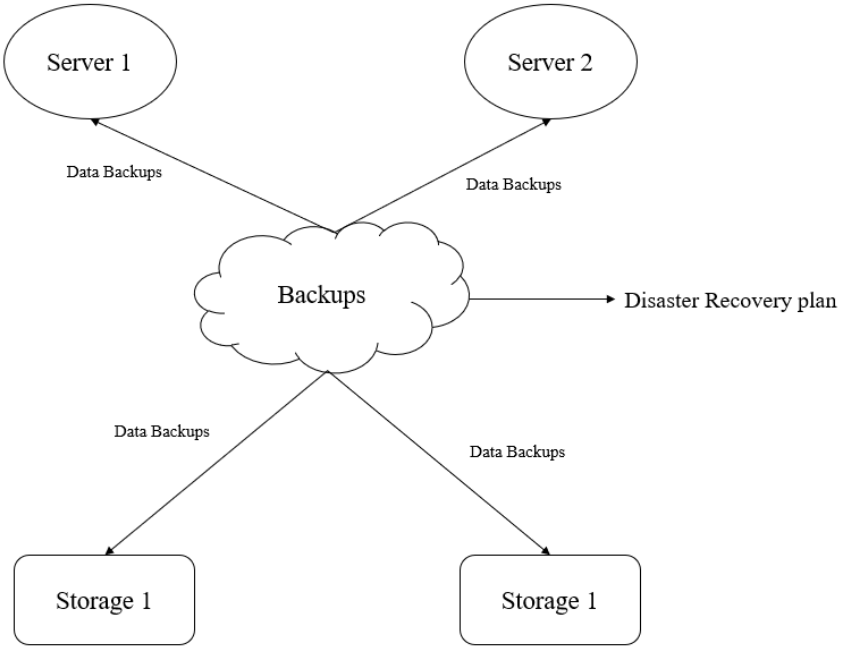
7. Data Encryption

Data encryption is a crucial security technique that converts plaintext data into ciphertext, requiring the use of a decryption key to ensure its security.


Server 1
Server 2
Encryption (In transit or at rest)
Encryption (In transit or at rest)
Data Encryption
Encryption (In transit or at rest)
Encryption (In transit or at rest)
Storage 1
Storage 1

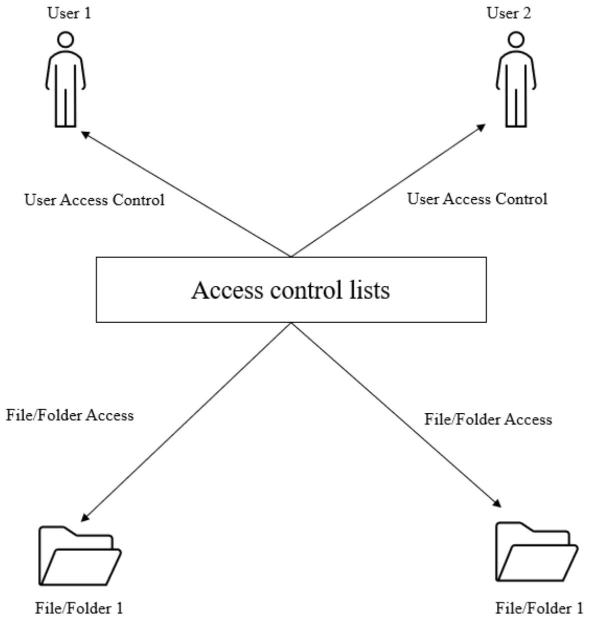
8. Backup and Disaster Recovery

Backup and Disaster Recovery (BDR) is a strategy implemented by organizations to ensure operational continuity and data availability amidst disruptions such as hardware failures, data corruption, natural disasters, and cyberattacks.



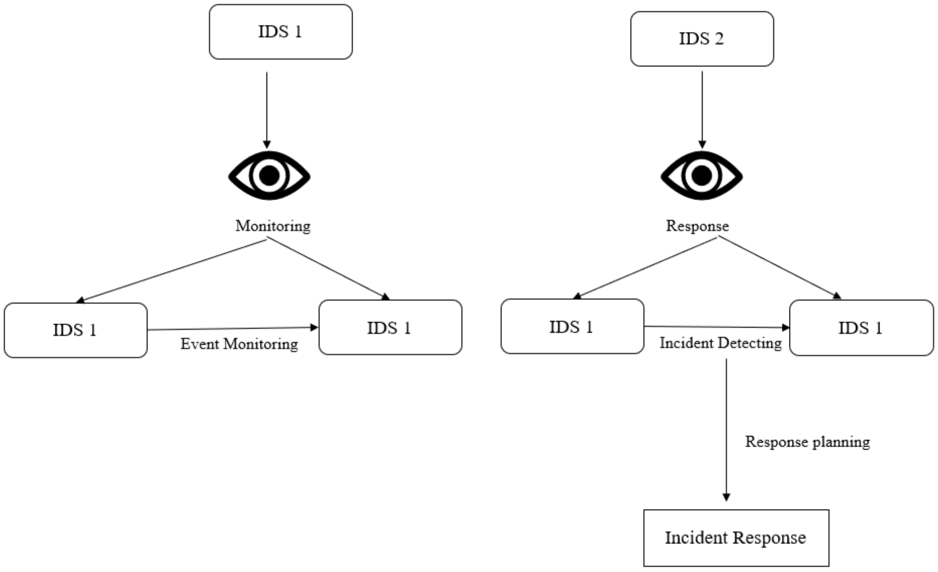
9. Access Control Lists (ACLs)

Access Control Lists (ACLs) are crucial security mechanisms that manage user and system resource permissions, ensuring fine-grained control over access and actions in operating systems and network devices.



10.Security Monitoring and Incident Response

Security Monitoring and Incident Response are essential practices in information security that help organizations proactively identify and mitigate security threats and respond effectively when security incidents occur [3].



Task 2

2. Configuration distribution and increase the security

a. Restrict Database Privileges:

Explanation: By revoking unnecessary privileges from users and roles, you minimize the risk of privilege escalation attacks.

Implement: Use Oracle's built-in role and privilege management to grant only the permissions necessary for specific users or roles.

b. Enable Database Auditing:

Explanation: Enabling auditing captures a record of actions taken within the database, aiding in identifying and responding to unauthorized or suspicious activities.

Implement: Set up audit policies to track access to sensitive data, database schema changes, and security-relevant events.

c. Implement Transparent Data Encryption (TDE):

Explanation: TDE protects data at rest by encrypting it. It safeguards against data breaches, ensuring that even if physical storage is compromised, the data remains secure.

Implement: Configure TDE for the specific database columns or tables that hold sensitive information. Properly manage encryption keys and wallets.

d. Database Patching and Updates:

Explanation: Regularly apply Oracle's Critical Patch Updates (CPUs) to address known vulnerabilities and security issues. Timely patching is essential to keep the database secure.

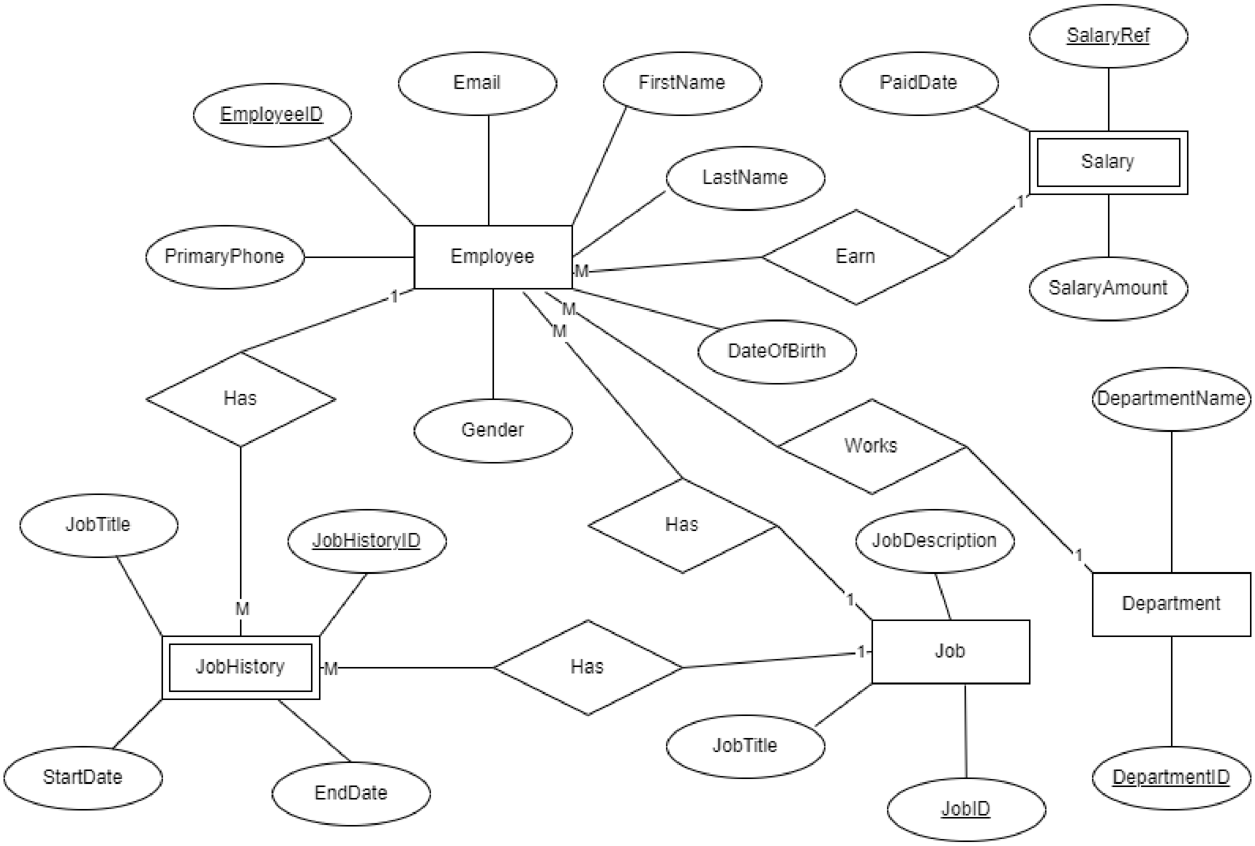
Implement: Stay informed about Oracle's security patches and establish a patch management process to apply updates promptly.

e. Fine-Grained Auditing (FGA):

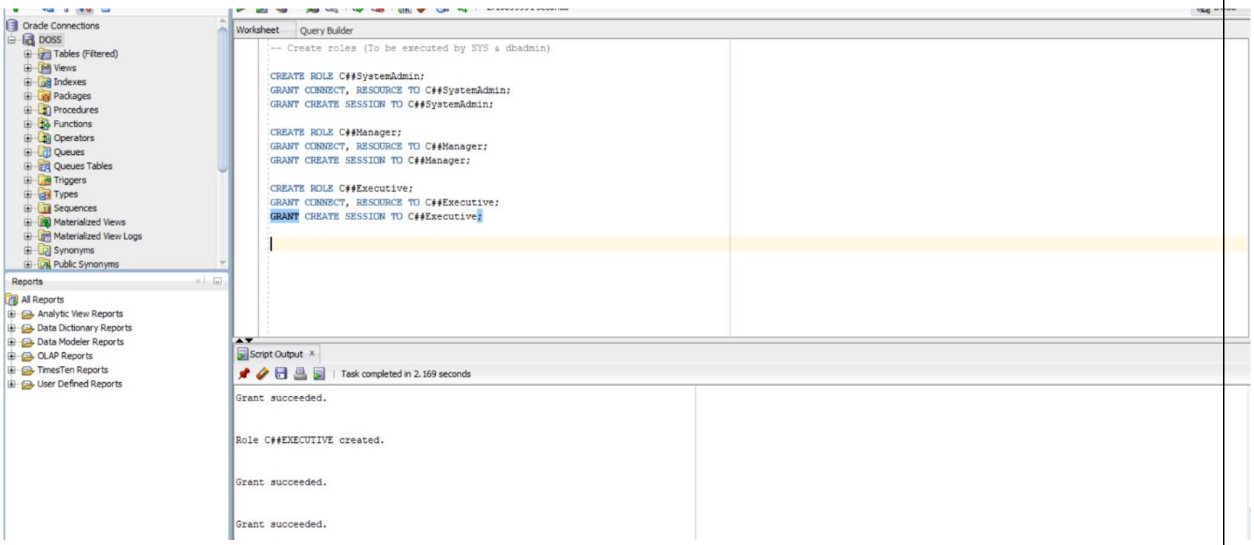
Explanation: Implement FGA policies to track and log specific data access activities based on predefined criteria. This is crucial for monitoring and auditing sensitive data.

Implement: Define FGA policies that specify which actions on specific tables or columns trigger auditing. Customize audit actions and reporting.

3. ER Diagram



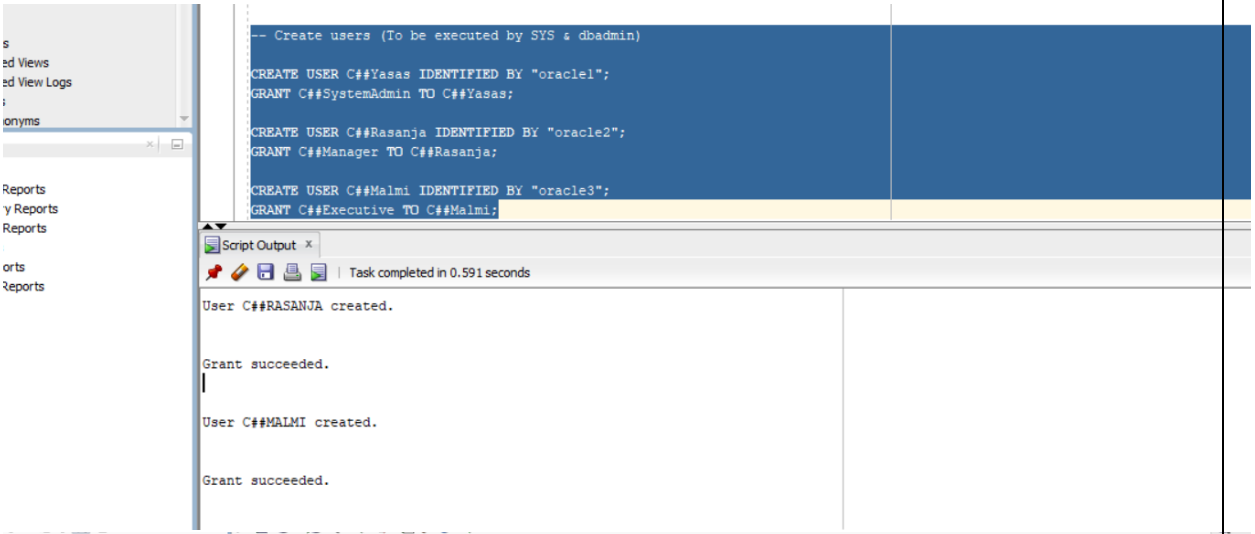
4. Tables



```
-- Create roles (To be executed by SYS & dbadmin)

CREATE ROLE C##SystemAdmin;
GRANT CONNECT, RESOURCE TO C##SystemAdmin;
GRANT CREATE SESSION TO C##SystemAdmin;

CREATE ROLE C##Manager;
GRANT CONNECT, RESOURCE TO C##Manager;
GRANT CREATE SESSION TO C##Manager;

CREATE ROLE C##Executive;
GRANT CONNECT, RESOURCE TO C##Executive;
GRANT CREATE SESSION TO C##Executive;
```

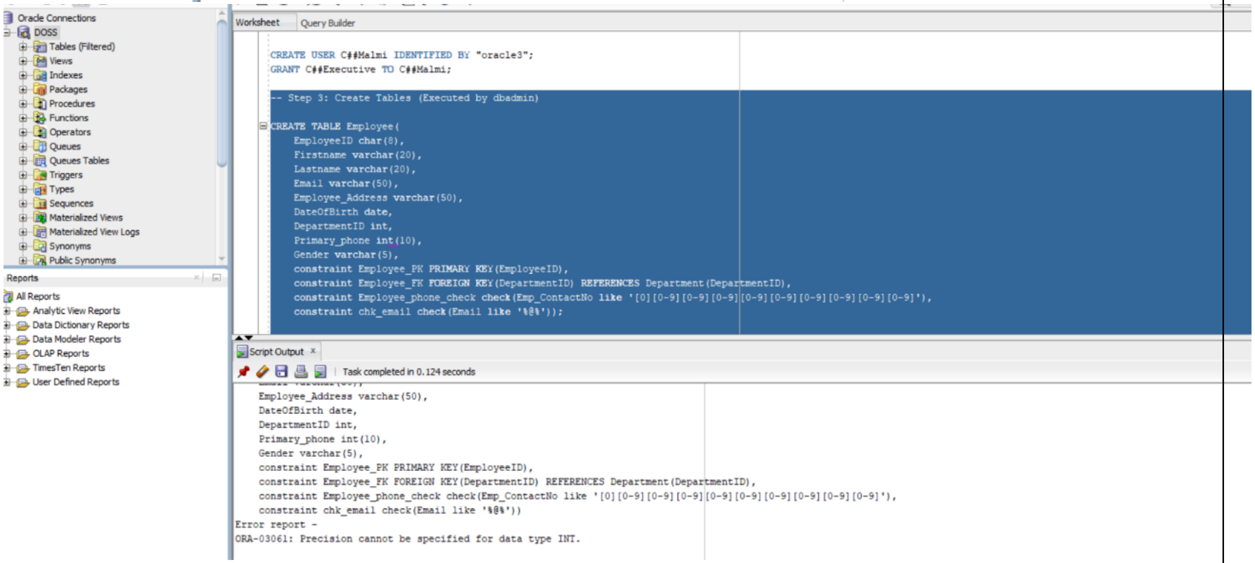
Task completed in 2.169 seconds

```
Grant succeeded.


Role C##EXECUTIVE created.


Grant succeeded.


Grant succeeded.
```



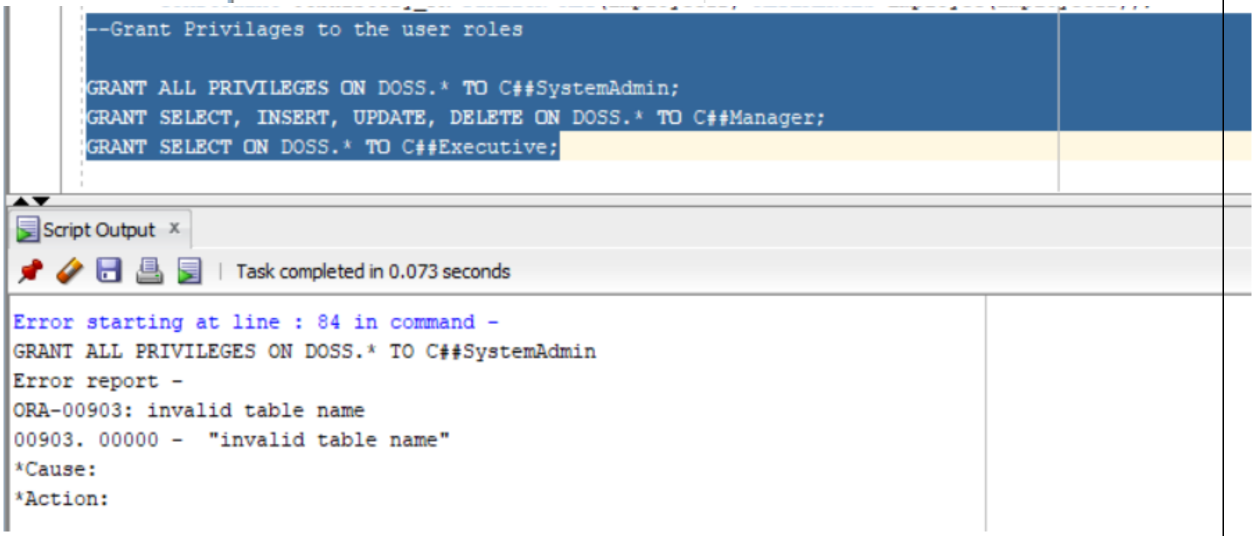
```
-- Create users (To be executed by SYS & dbadmin)

CREATE USER C##Yasas IDENTIFIED BY "oracle1";
GRANT C##SystemAdmin TO C##Yasas;

CREATE USER C##Rasanja IDENTIFIED BY "oracle2";
GRANT C##Manager TO C##Rasanja;

CREATE USER C##Malmi IDENTIFIED BY "oracle3";
GRANT C##Executive TO C##Malmi;
```

```
User C##RASANJA created.


Grant succeeded.


User C##MALMI created.


Grant succeeded.
```



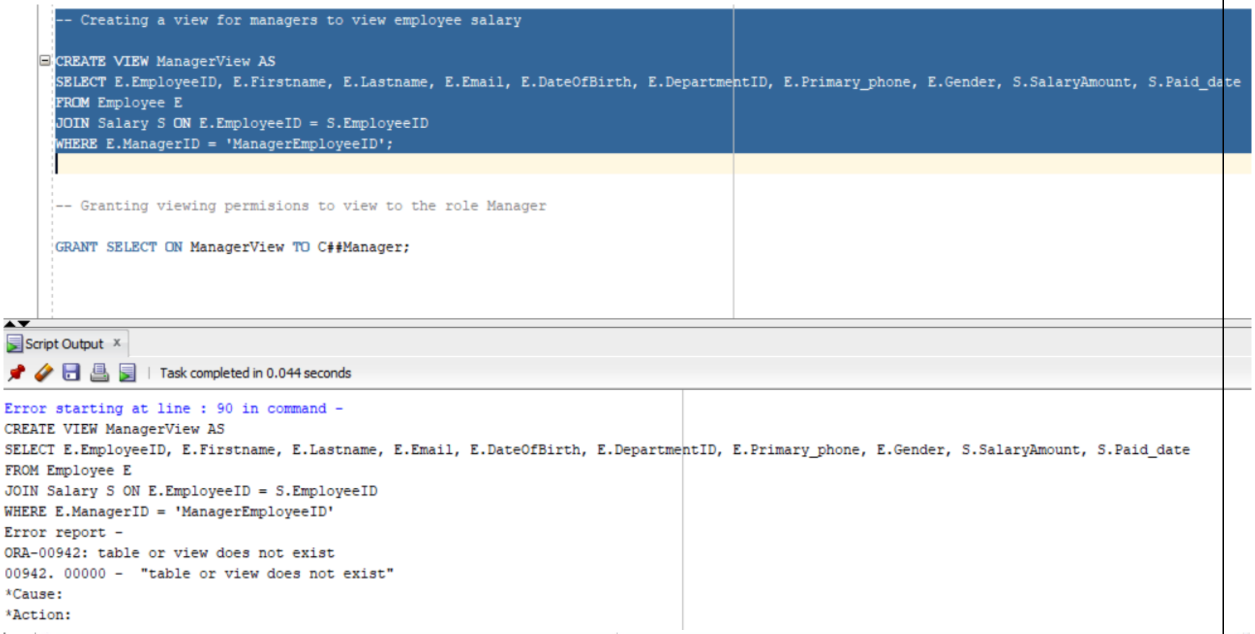
```
CREATE USER C##Malmi IDENTIFIED BY "oracle3";
GRANT C##Executive TO C##Malmi;

-- Step 3: Create Tables (Executed by dbadmin)

CREATE TABLE Employee(
    EmployeeID char(8),
    Firstname varchar(20),
    Lastname varchar(20),
    Email varchar(50),
    Employee_Address varchar(50),
    DateOfBirth date,
    DepartmentID int,
    Primary_phone int(10),
    Gender varchar(5),
    constraint Employee_PK PRIMARY KEY(EmployeeID),
    constraint Employee_FK FOREIGN KEY(DepartmentID) REFERENCES Department(DepartmentID),
    constraint Employee_phone_check check(Emp_ContactNo like '[0][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    constraint chk_email check(Email like '%@%'));
```

```
    Employee_Address varchar(50),
    DateOfBirth date,
    DepartmentID int,
    Primary_phone int(10),
    Gender varchar(5),
    constraint Employee_PK PRIMARY KEY(EmployeeID),
    constraint Employee_FK FOREIGN KEY(DepartmentID) REFERENCES Department(DepartmentID),
    constraint Employee_phone_check check(Emp_ContactNo like '[0][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    constraint chk_email check(Email like '%@%'))
Error report -
ORA-03061: Precision cannot be specified for data type INT.
```



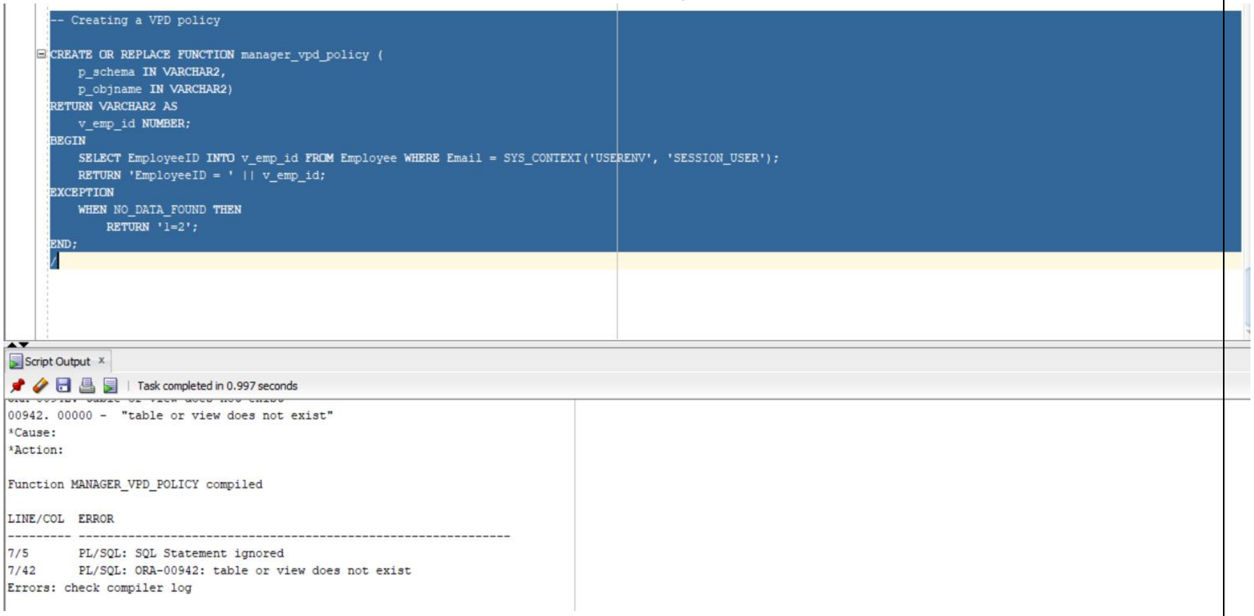
```
--Grant Privilages to the user roles

GRANT ALL PRIVILEGES ON DOSS.* TO C##SystemAdmin;
GRANT SELECT, INSERT, UPDATE, DELETE ON DOSS.* TO C##Manager;
GRANT SELECT ON DOSS.* TO C##Executive;
```

```
Error starting at line : 84 in command -
GRANT ALL PRIVILEGES ON DOSS.* TO C##SystemAdmin
Error report -
ORA-00903: invalid table name
00903. 00000 -  "invalid table name"
*Cause:
*Action:
```



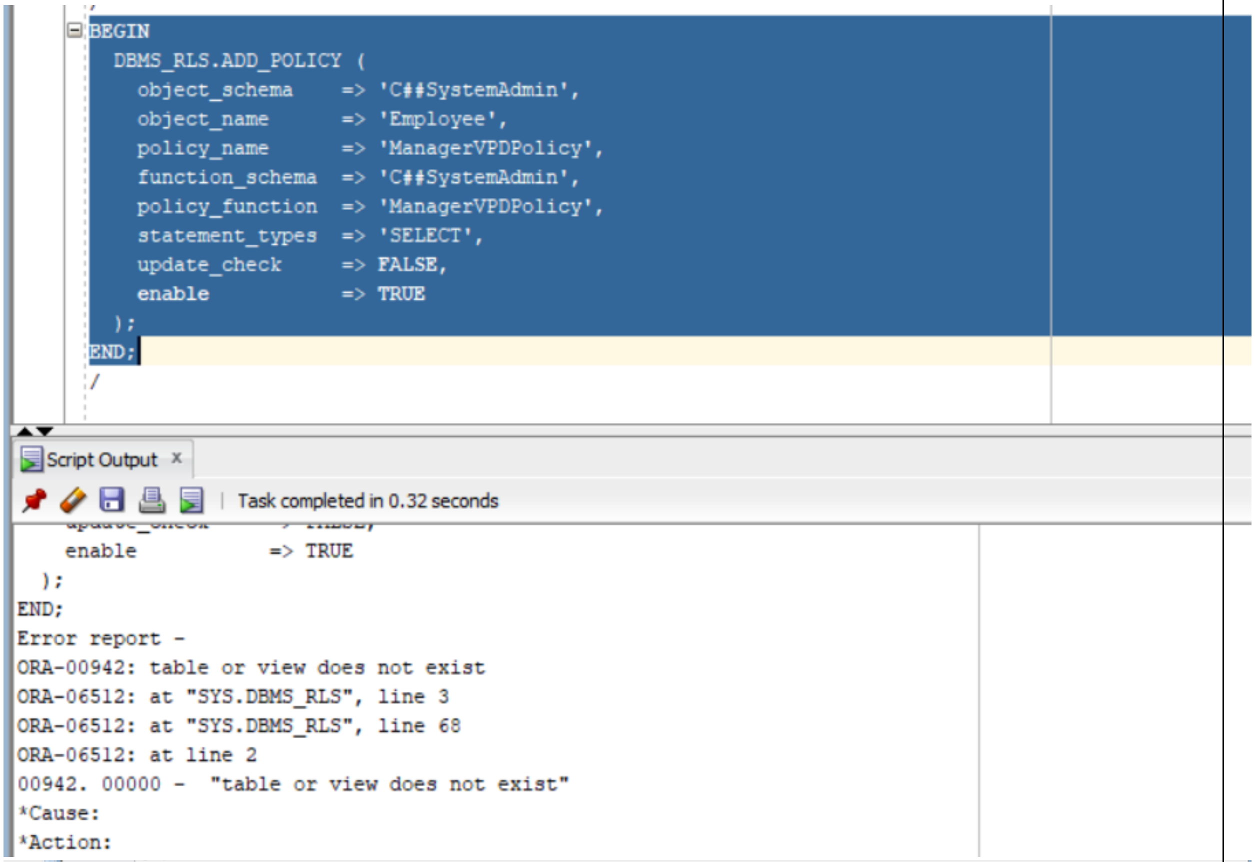
```
-- Creating a view for managers to view employee salary

CREATE VIEW ManagerView AS
SELECT E.EmployeeID, E.Firstname, E.Lastname, E.Email, E.DateOfBirth, E.DepartmentID, E.Primary_phone, E.Gender, S.SalaryAmount, S.Paid_date
FROM Employee E
JOIN Salary S ON E.EmployeeID = S.EmployeeID
WHERE E.ManagerID = 'ManagerEmployeeID';


-- Granting viewing permisions to view to the role Manager

GRANT SELECT ON ManagerView TO C##Manager;
```

```
Error starting at line : 90 in command -
CREATE VIEW ManagerView AS
SELECT E.EmployeeID, E.Firstname, E.Lastname, E.Email, E.DateOfBirth, E.DepartmentID, E.Primary_phone, E.Gender, S.SalaryAmount, S.Paid_date
FROM Employee E
JOIN Salary S ON E.EmployeeID = S.EmployeeID
WHERE E.ManagerID = 'ManagerEmployeeID'
Error report -
ORA-00942: table or view does not exist
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
```



```
-- Creating a VPD policy

CREATE OR REPLACE FUNCTION manager_vpd_policy (
    p_schema IN VARCHAR2,
    p_objname IN VARCHAR2)
RETURN VARCHAR2 AS
    v_emp_id NUMBER;
BEGIN
    SELECT EmployeeID INTO v_emp_id FROM Employee WHERE Email = SYS_CONTEXT('USERENV', 'SESSION_USER');
    RETURN 'EmployeeID = ' || v_emp_id;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN '1=2';
END;
/
```

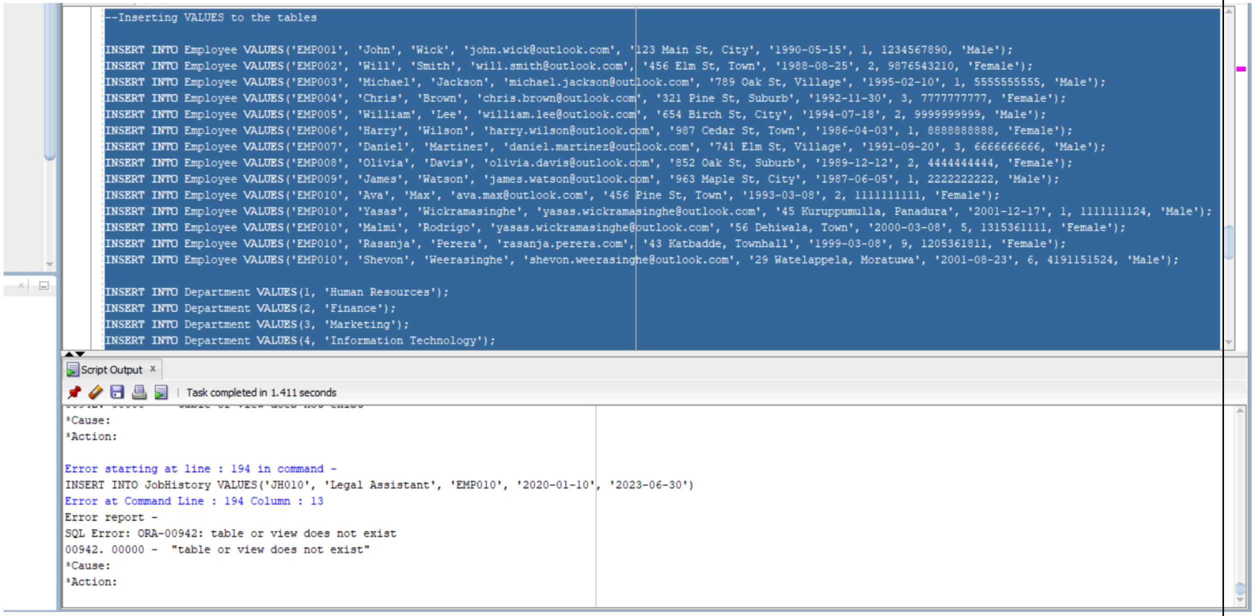
```
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:

Function MANAGER_VPD_POLICY compiled

LINE/COL  ERROR
--------- -------------------------------------------------------------
7/5       PL/SQL: SQL Statement ignored
7/42      PL/SQL: ORA-00942: table or view does not exist
Errors: check compiler log
```



```
BEGIN
  DBMS_RLS.ADD_POLICY (
    object_schema     => 'C##SystemAdmin',
    object_name       => 'Employee',
    policy_name       => 'ManagerVPDPolicy',
    function_schema   => 'C##SystemAdmin',
    policy_function   => 'ManagerVPDPolicy',
    statement_types   => 'SELECT',
    update_check      => FALSE,
    enable            => TRUE
  );
END;
/
```

Task completed in 0.32 seconds

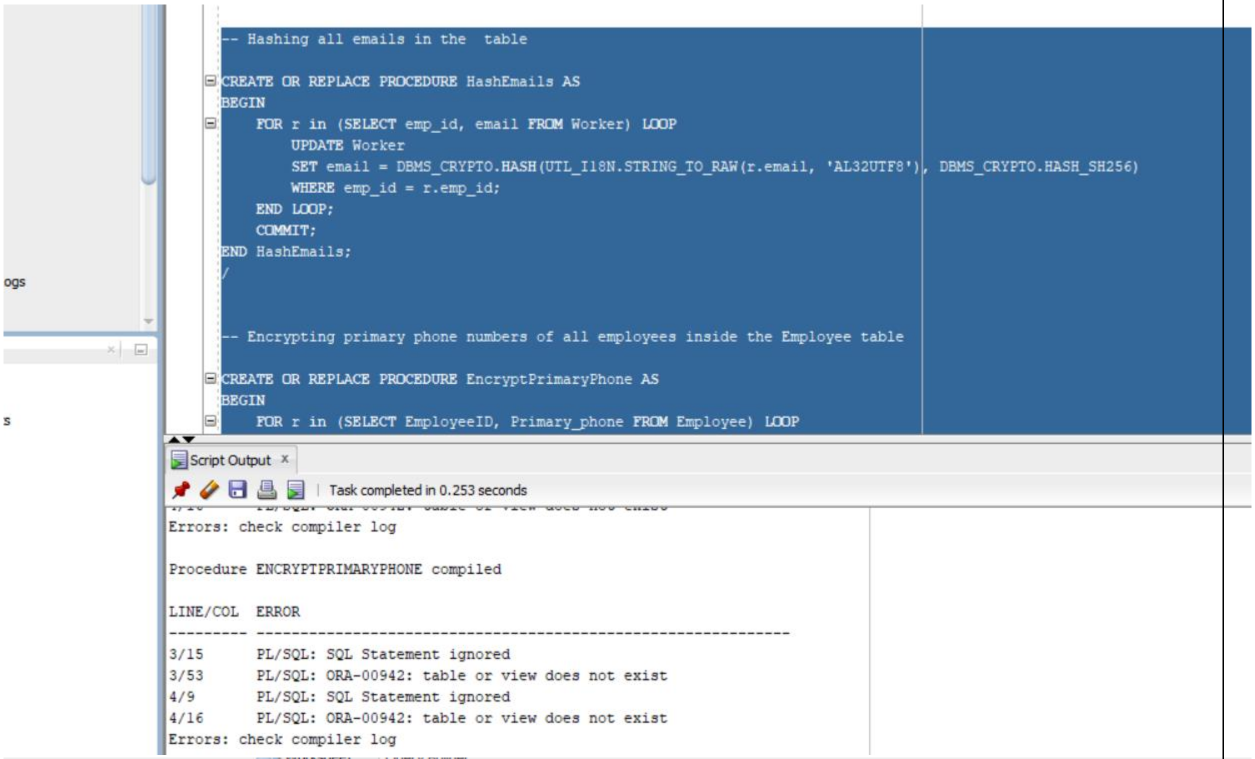
```
    update_check       => FALSE,
    enable             => TRUE
  );
END;
Error report -
ORA-00942: table or view does not exist
ORA-06512: at "SYS.DBMS_RLS", line 3
ORA-06512: at "SYS.DBMS_RLS", line 68
ORA-06512: at line 2
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
```



```
--Inserting VALUES to the tables

INSERT INTO Employee VALUES('EMP001', 'John', 'Wick', 'john.wick@outlook.com', '123 Main St, City', '1990-05-15', 1, 1234567890, 'Male');
INSERT INTO Employee VALUES('EMP002', 'Will', 'Smith', 'will.smith@outlook.com', '456 Elm St, Town', '1988-08-25', 2, 9876543210, 'Female');
INSERT INTO Employee VALUES('EMP003', 'Michael', 'Jackson', 'michael.jackson@outlook.com', '789 Oak St, Village', '1995-02-10', 1, 5555555555, 'Male');
INSERT INTO Employee VALUES('EMP004', 'Chris', 'Brown', 'chris.brown@outlook.com', '321 Pine St, Suburb', '1992-11-30', 3, 7777777777, 'Female');
INSERT INTO Employee VALUES('EMP005', 'William', 'Lee', 'william.lee@outlook.com', '654 Birch St, City', '1994-07-18', 2, 9999999999, 'Male');
INSERT INTO Employee VALUES('EMP006', 'Harry', 'Wilson', 'harry.wilson@outlook.com', '987 Cedar St, Town', '1986-04-03', 1, 8888888888, 'Female');
INSERT INTO Employee VALUES('EMP007', 'Daniel', 'Martinez', 'daniel.martinez@outlook.com', '741 Elm St, Village', '1991-09-20', 3, 6666666666, 'Male');
INSERT INTO Employee VALUES('EMP008', 'Olivia', 'Davis', 'olivia.davis@outlook.com', '852 Oak St, Suburb', '1989-12-12', 2, 4444444444, 'Female');
INSERT INTO Employee VALUES('EMP009', 'James', 'Watson', 'james.watson@outlook.com', '963 Maple St, City', '1987-06-05', 1, 2222222222, 'Male');
INSERT INTO Employee VALUES('EMP010', 'Ava', 'Max', 'ava.max@outlook.com', '456 Pine St, Town', '1993-03-08', 2, 1111111111, 'Female');
INSERT INTO Employee VALUES('EMP010', 'Yasas', 'Wickramasinghe', 'yasas.wickramasinghe@outlook.com', '45 Kuruppumulla, Panadura', '2001-12-17', 1, 1111111124, 'Male');
INSERT INTO Employee VALUES('EMP010', 'Malmi', 'Rodrigo', 'yasas.wickramasinghe@outlook.com', '56 Dehiwala, Town', '2000-03-08', 5, 1315361111, 'Female');
INSERT INTO Employee VALUES('EMP010', 'Rasanja', 'Perera', 'rasanja.perera.com', '43 Katbadde, Townhall', '1999-03-08', 9, 1205361811, 'Female');
INSERT INTO Employee VALUES('EMP010', 'Shevon', 'Weerasinghe', 'shevon.weerasinghe@outlook.com', '29 Watelappela, Moratuwa', '2001-08-23', 6, 4191151524, 'Male');

INSERT INTO Department VALUES(1, 'Human Resources');
INSERT INTO Department VALUES(2, 'Finance');
INSERT INTO Department VALUES(3, 'Marketing');
INSERT INTO Department VALUES(4, 'Information Technology');
```

Task completed in 1.411 seconds

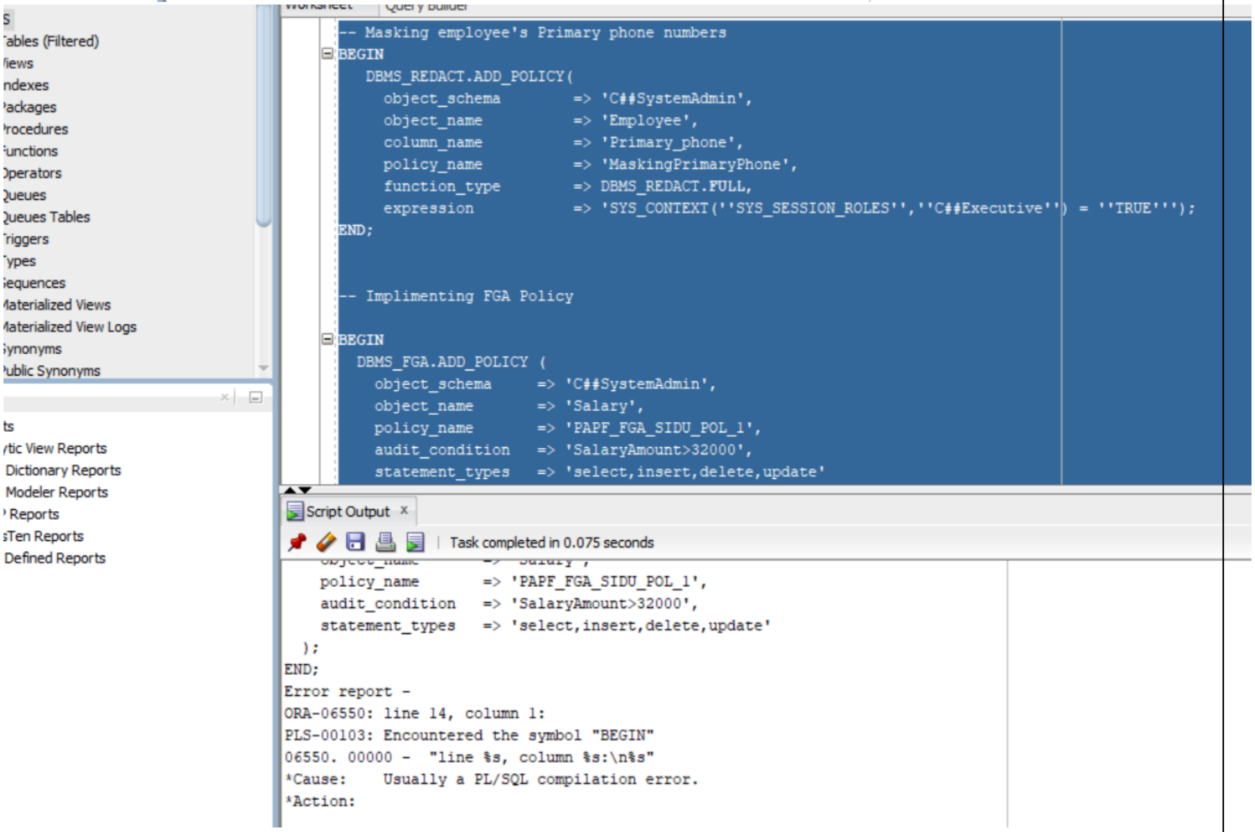
```
*Cause:
*Action:

Error starting at line : 194 in command -
INSERT INTO JobHistory VALUES('JH010', 'Legal Assistant', 'EMP010', '2020-01-10', '2023-06-30')
Error at Command Line : 194 Column : 13
Error report -
SQL Error: ORA-00942: table or view does not exist
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
```



```
-- Hashing all emails in the  table

CREATE OR REPLACE PROCEDURE HashEmails AS
BEGIN
    FOR r in (SELECT emp_id, email FROM Worker) LOOP
        UPDATE Worker
        SET email = DBMS_CRYPTO.HASH(UTL_I18N.STRING_TO_RAW(r.email, 'AL32UTF8'), DBMS_CRYPTO.HASH_SH256)
        WHERE emp_id = r.emp_id;
    END LOOP;
    COMMIT;
END HashEmails;
/


-- Encrypting primary phone numbers of all employees inside the Employee table

CREATE OR REPLACE PROCEDURE EncryptPrimaryPhone AS
BEGIN
    FOR r in (SELECT EmployeeID, Primary_phone FROM Employee) LOOP
```

Task completed in 0.253 seconds

```
Errors: check compiler log

Procedure ENCRYPTPRIMARYPHONE compiled

LINE/COL  ERROR
--------- -------------------------------------------------------------
3/15      PL/SQL: SQL Statement ignored
3/53      PL/SQL: ORA-00942: table or view does not exist
4/9       PL/SQL: SQL Statement ignored
4/16      PL/SQL: ORA-00942: table or view does not exist
Errors: check compiler log
```



```
-- Masking employee's Primary phone numbers
BEGIN
   DBMS_REDACT.ADD_POLICY(
     object_schema        => 'C##SystemAdmin',
     object_name          => 'Employee',
     column_name          => 'Primary_phone',
     policy_name          => 'MaskingPrimaryPhone',
     function_type        => DBMS_REDACT.FULL,
     expression           => 'SYS_CONTEXT(''SYS_SESSION_ROLES'',''C##Executive'') = ''TRUE''');
END;


-- Implimenting FGA Policy

BEGIN
  DBMS_FGA.ADD_POLICY (
    object_schema     => 'C##SystemAdmin',
    object_name       => 'Salary',
    policy_name       => 'PAPF_FGA_SIDU_POL_1',
    audit_condition   => 'SalaryAmount>32000',
    statement_types   => 'select,insert,delete,update'
```

Task completed in 0.075 seconds

```
    policy_name       => 'PAPF_FGA_SIDU_POL_1',
    audit_condition   => 'SalaryAmount>32000',
    statement_types   => 'select,insert,delete,update'
  );
END;
Error report -
ORA-06550: line 14, column 1:
PLS-00103: Encountered the symbol "BEGIN"
06550. 00000 -  "line %s, column %s:\n%s"
*Cause:    Usually a PL/SQL compilation error.
*Action:
```

Task 3

Literature Review: Big Data Security

This literature review explores the security landscape of Big Data, focusing on its unique requirements, common attack vectors, and necessary controls for protection.

a. Requirements and traditional databases

- **Security Requirements of Big Data**

1. **Scalability:**
   - Big Data systems necessitate scalable security solutions that can manage a range of data types and volumes without compromising security, which is why they are crucial for data expansion.
2. ***Data Variety:***
   - The diversity of data types, including structured, semi-structured, and unstructured data, demands security measures capable of addressing complex data integration and protection issues (Chen et al., 2014).
3. ***Real-time Processing:***
   - Big Data often involves real-time or near-real-time processing, necessitating security controls that enable quick detection and response to threats (Luo et al., 2016).

- **Comparison with Traditional Databases**

  Big Data systems handle vast volumes of heterogeneous data from diverse sources, necessitating more adaptable and scalable security mechanisms compared to traditional databases.

b. Common Attacks on Big Data

1. **Data Breaches:**
   - The risk of unauthorized access to Big Data repositories, which could lead to sensitive data exposure, necessitates the

implementation of advanced access control mechanisms. (Zheng et al., 2019).

2. **Distributed Denial of Service (DDoS):**

   - DDoS attacks pose a significant threat to big data systems, necessitating effective mitigation strategies and traffic monitoring to ensure system availability. (Sakib et al., 2017).

3. **Insider Threats:**

   - User behavior analytics and monitoring aid in detecting and preventing insider threats from authorized users, including employees and contractors, early. (Jin et al., 2019).

4. **Data Injections:**

   - Data injection attacks are crucial to prevent malicious data injections into Big Data environments, requiring robust data validation and sanitization. (Sahin et al., 2016).

5. **Evasion Attacks:**

   - SIEM solutions enable effective security monitoring, detecting, and responding to evasion attacks that attempt to bypass security controls and policies. (Li et al., 2018).

c. Security Controls for Big Data

1. Data Breaches:

The text emphasizes the importance of implementing security controls to prevent unauthorized access to sensitive data in Big Data environments:

a. **Access Control and Authentication:** - Role-based access control (RBAC) and robust authentication mechanisms ensure authorized users access specific data and systems, categorizing users into roles with specific permissions.

b. **Encryption of Data at Rest and in Transit:** - Transparent Data Encryption and secure communication protocols like TLS/SSL are essential in preventing unauthorized access to Big Data repositories and data transfer.

2. Distributed Denial of Service (DDoS) Attacks:

DDoS attacks can disrupt Big Data processing and analysis. To mitigate this threat, implement the following security controls:

a. **Traffic Analysis and Rate Limiting:** - Utilize traffic analysis tools and rate-limiting policies to detect and manage unusual traffic patterns, ensuring the system can handle specific requests per unit of time.

b. **Content Delivery Network (CDN):** - A Content Delivery Network (CDN) distributes content and mitigates DDoS attacks by spreading load across geographically distributed servers, absorbing traffic during attacks, and maintaining service availability.

3. Insider Threats:

Insider threats from authorized users are persistent security challenges. To mitigate this threat, consider these security controls:

a. **User Behavior Analytics (UBA):** - UBA systems continuously monitor and analyze user behavior, detecting unusual or suspicious activities by collecting and analyzing data to establish a baseline and generate alerts for further investigation.

b. **Least Privilege Principle:** - PoLP minimizes insider threats and unauthorized access to data or systems by restricting user and application permissions to the minimum required for tasks.

References

[1] "https://www.stackscale.com/blog/popular-linux-distributions/," *stackscale,* 2023.

[2] D. Varma, "https://pub.towardsai.net/operating-systems-types-and-security-f319bec1078b," *Medium,* 2023.

[3] "https://www.researchgate.net/figure/Operating-System-Security-management_fig4_235090678," *Research Gate.*

[4] "https://www.datamation.com/big-data/big-data-security/," *Datamation.*

[5] "https://wearebrain.com/blog/cloud-based-vs-traditional-databases-comparison/," *Wearebrain.*

[6] D. Kumar, "https://hevodata.com/learn/big-data-security/," *Hevo Data,* 2023.

[7] "https://www.turing.com/resources/big-data-security," *Turing.*