Sri Lanka Institute of Information Technology

IE3062 – Data and Operating System Security

Group Details

Group Number - 4

Registration Number	Name

1. OS Security

Comparison of 5 Linux Distributions

Distribution	Pros	Cons	Best for	Security Features
Ubuntu	- Excellent documentation & community support - Regular LTS releases (5-year support) - Large software repository - Easy to administer	- Can be resource-heavy - Some enterprise features require paid subscription	**General purpose servers, DevOps, cloud deployments**	- AppArmor by default - Unattended security updates - Strong SELinux/AppArmor support
CentOS	- Enterprise-grade stability - Binary compatible with RHEL - Long support cycles (10 years) - Conservative updates	- Slower package updates - Less beginner-friendly - CentOS Stream changes support model	**Enterprise environments, web servers, databases**	- SELinux enabled by default - Security team backports patches - NSA-developed security modules
Debian	- Extremely stable - Massive software repository - Free software philosophy - Lightweight	- Older package versions - Less frequent releases - Limited commercial support	**Mission-critical systems, embedded devices**	- Conservative security approach - Regular security updates - AppArmor optional
Red Hat Enterprise Linux (RHEL)	- Enterprise support available - Certified for enterprise software - Strong security compliance - Stable release cycle	- Expensive subscription required - Restricted repositories without subscription	**Large enterprises, government, financial institutions**	- SELinux mandatory access control - SCAP security profiles - FIPS 140-2 compliance
openSUSE	- YaST configuration tool - Btrfs filesystem by default - Strong KVM virtualization support - Regular rolling release available	- Smaller community than Ubuntu/RHEL - Less third-party software support	**System administrators, developers, mixed environments**	- AppArmor and SELinux support - Integrated security tools in YaST - Automated snapshots

Selection Justification: Ubuntu

1. Security Hardening Capabilities:

- **AppArmor** pre-installed and configured for major services
- **UFW (Uncomplicated Firewall)** for easy yet powerful network security
- **Automatic security updates** via unattended-upgrades package
- **Large security community** providing timely patches and guidance

2. Enterprise Readiness:

- **LTS (Long-Term Support)** releases provide 5 years of security updates
- **Oracle Database certification** - officially supported and tested
- **Stable package base** with security backports
- **Proven track record** in production environments (used by AWS, Google Cloud, IBM)

3. Administrative Efficiency:

- **GUI availability** simplifies initial setup and monitoring for junior administrators
- **Extensive documentation** reduces troubleshooting time
- **apt package management** provides reliable dependency resolution
- **Strong hardware compatibility** reduces driver issues

4. Cost-Effectiveness for Small Business:

- **Zero licensing costs** - crucial for small company budget
- **Reduced training time** due to widespread Ubuntu knowledge
- **Community support** eliminates need for expensive support contracts

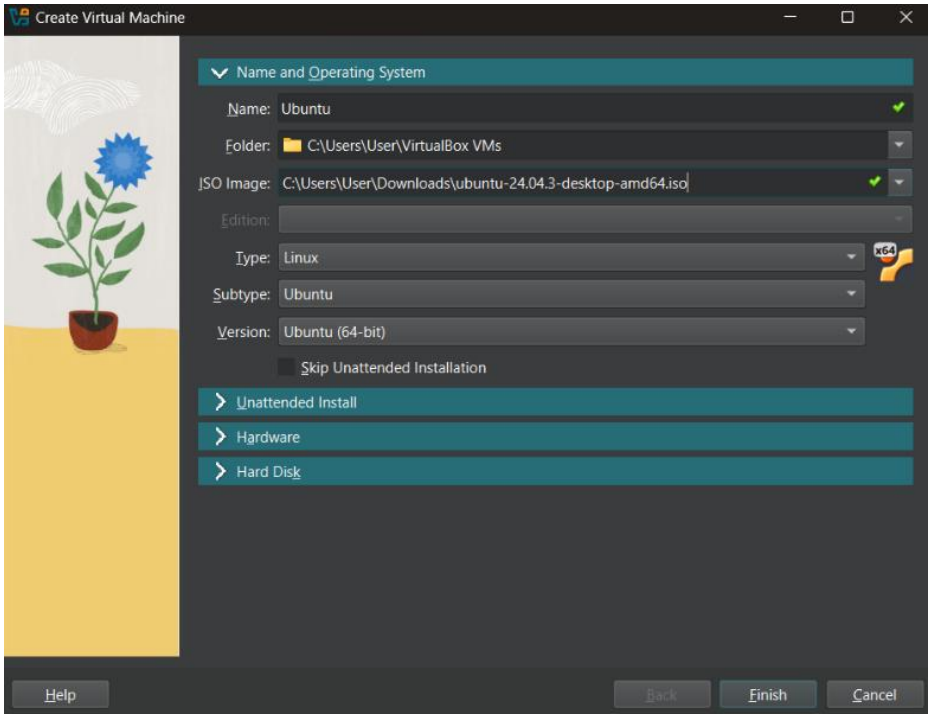
5. Specific HR Database Server Requirements:

- **Oracle DBMS compatibility** - Ubuntu is a supported platform
- **GUI access** allows visual monitoring of system resources
- **Security tools availability** for compliance with data protection regulations
- **Performance optimization** tools for database workloads

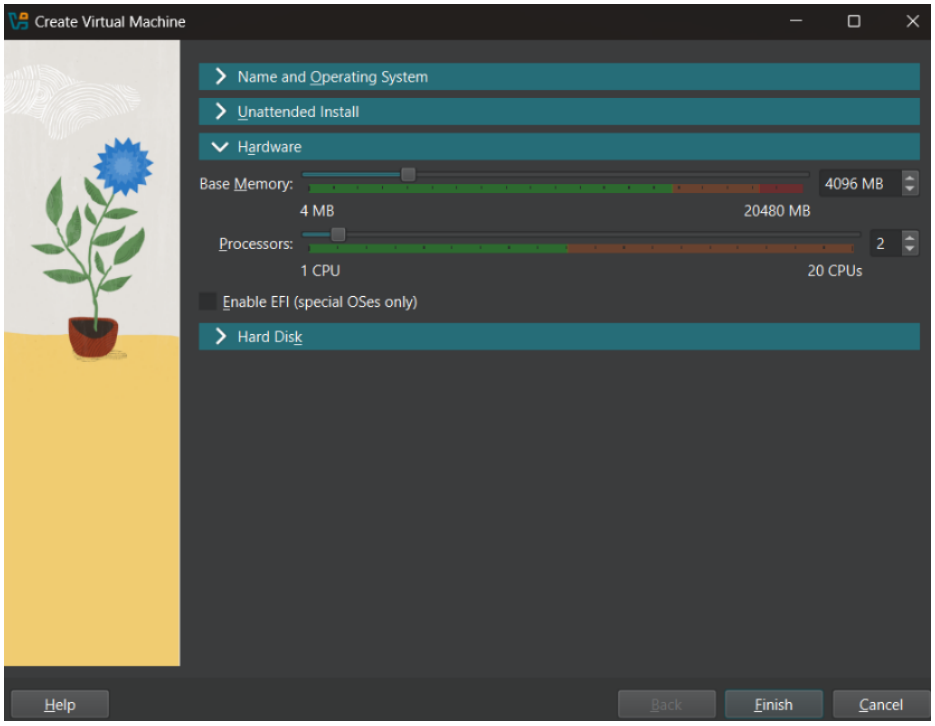
OS Installation and Initial Setup

Installation Steps:

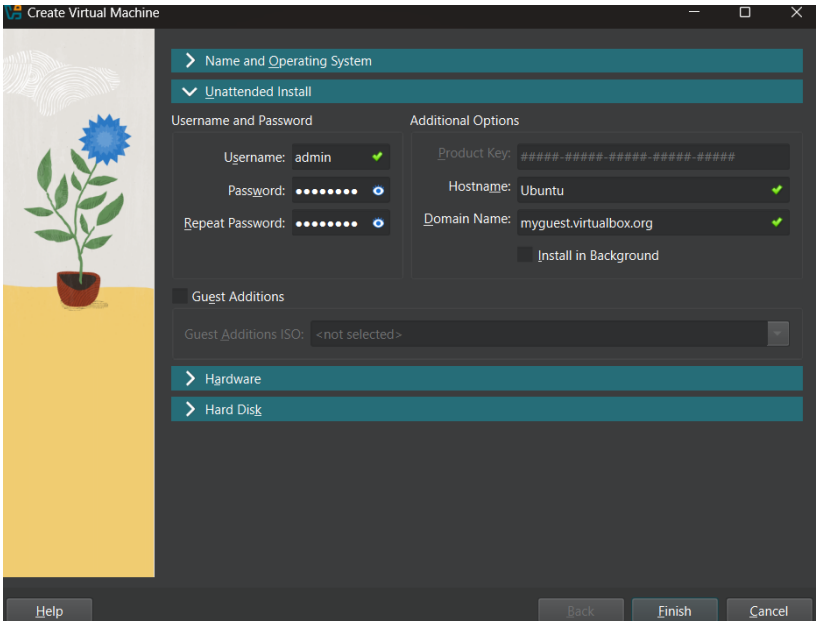
1. **Downloaded** Ubuntu Server 24.04 LTS ISO



2. **Created VM** with 4GB RAM, 2 CPUs, 25GB storage



3. **Configured** user admin with strong password



10 Security Hardening Configurations Implemented:

1. Firewall Configuration (UFW)

```
admin@Ubuntu:~$ sudo ufw enable
[sudo] password for admin:
Firewall is active and enabled on system startup
```

The system starts blocking/allowing network traffic according to the rules Admin define.

```
admin@Ubuntu:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
admin@Ubuntu:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Any device or program trying to connect to your computer from outside is **blocked by default**, unless Admin explicitly allows it & sets the **default policy for outgoing traffic** to **allow.**

```
admin@Ubuntu:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
```

Opens port **22/tcp**, which is the standard port for SSH (Secure Shell).

```
admin@Ubuntu:~$ sudo ufw allow 1521
Rule added
Rule added (v6)
```

Opens port **1521/tcp**, which is the default port used by **Oracle Database listener.**

```
admin@Ubuntu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
1521                       ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
1521 (v6)                  ALLOW IN    Anywhere (v6)
```

Displays the current rules in place.

2. SSH Service Hardening

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
MaxAuthTries 3
#MaxSessions 10
```

Disables **root** user login via SSH & Limits the number of **failed login attempts** to **3** per SSH connection.

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

Only **SSH key-based authentication** is allowed

```
ClientAliveInterval 300
ClientAliveCountMax 0
```

If a client doesn't respond, the server will eventually disconnect the idle session.

```
admin@Ubuntu:~$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.backup
admin@Ubuntu:~$ sudo nano /etc/ssh/sshd_config
admin@Ubuntu:~$ sudo systemctl restart ssh
admin@Ubuntu:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-10-05 17:22:20 UTC; 9s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 6737 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 6738 (sshd)
      Tasks: 2 (limit: 4604)
     Memory: 2.4M (peak: 2.8M)
        CPU: 17ms
     CGroup: /system.slice/ssh.service
             ├─6298 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
             └─6738 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Oct 05 17:22:20 Ubuntu systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 05 17:22:20 Ubuntu systemd[1]: ssh.service: Found left-over process 6298 (sshd) in control group while starting unit. Ignoring.
Oct 05 17:22:20 Ubuntu systemd[1]: ssh.service: This usually indicates unclean termination of a previous run, or service implementation deficiencies.
Oct 05 17:22:20 Ubuntu sshd[6738]: Server listening on 0.0.0.0 port 22.
Oct 05 17:22:20 Ubuntu sshd[6738]: Server listening on :: port 22.
Oct 05 17:22:20 Ubuntu systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

3. Fail2Ban Intrusion Prevention

```
admin@Ubuntu:~$ sudo apt install fail2ban -y
```

Installs the **Fail2Ban intrusion prevention tool**.

```
# "bantime" is the number of seconds that a host is banned.
bantime  = 1h

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 10m

# "maxretry" is the number of failures before a host get banned.
maxretry = 3
```

Sets the **ban duration** to **3600 seconds (1 hour)**, defines a **10-minute window (600 seconds)** during which Fail2Ban monitors failed login attempts & allows **only 3 failed login attempts** before banning the IP.

```
[selinux-ssh]

port = ssh
logpath = /var/log/auth.log
maxretry = 3
```

Allows only 3 failed attempts over SSH

```
admin@Ubuntu:~$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
admin@Ubuntu:~$ sudo nano /etc/fail2ban/jail.local
admin@Ubuntu:~$ sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban
admin@Ubuntu:~$ sudo systemctl start fail2ban
admin@Ubuntu:~$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-10-05 17:24:25 UTC; 7min ago
       Docs: man:fail2ban(1)
   Main PID: 7150 (fail2ban-server)
      Tasks: 5 (limit: 4604)
     Memory: 20.6M (peak: 21.1M)
        CPU: 803ms
     CGroup: /system.slice/fail2ban.service
             └─7150 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Oct 05 17:24:25 Ubuntu systemd[1]: Started fail2ban.service - Fail2Ban Service.
Oct 05 17:24:25 Ubuntu fail2ban-server[7150]: 2025-10-05 17:24:25,104 fail2ban.configreader   [7150]: WARNING 'allowipv6' not defined in 'Definition'. Using default one: 'auto'
Oct 05 17:24:25 Ubuntu fail2ban-server[7150]: Server ready
admin@Ubuntu:~$
```

4. Password Policy Enforcement

```
admin@Ubuntu:~$ sudo apt install libpam-pwquality -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libpam-pwquality is already the newest version (1.4.5-3build1).
libpam-pwquality set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 64 not upgraded.
```

Installs the **PAM (Pluggable Authentication Module) pwquality library**.

```
# here are the per-package modules (the "Primary" block)
password requisite pam_pwquality.so retry=3 minlen=12 difok=3 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1 reject_username
```

Sets the **minimum password length** to **12 characters**, requires that the new password **differs by at least 3 characters** from the old one, requires at least **one uppercase letter** in the password, requires at least **one lowercase letter**, requires at least **one numeric digit** in the password & requires at least **one special character** (symbol).

5. Service Minimization

```
admin@Ubuntu:~$ sudo systemctl disable bluetooth
Synchronizing state of bluetooth.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable bluetooth
Removed "/etc/systemd/system/bluetooth.target.wants/bluetooth.service".
Removed "/etc/systemd/system/dbus-org.bluez.service".
admin@Ubuntu:~$ sudo systemctl disable cups
Synchronizing state of cups.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable cups
Removed "/etc/systemd/system/multi-user.target.wants/cups.path".
Removed "/etc/systemd/system/multi-user.target.wants/cups.service".
Removed "/etc/systemd/system/sockets.target.wants/cups.socket".
Removed "/etc/systemd/system/printer.target.wants/cups.service".
Disabling 'cups.service', but its triggering units are still active:
cups.path, cups.socket
admin@Ubuntu:~$ sudo systemctl disable avahi-daemon
Removed "/etc/systemd/system/multi-user.target.wants/avahi-daemon.service".
Removed "/etc/systemd/system/sockets.target.wants/avahi-daemon.socket".
Removed "/etc/systemd/system/dbus-org.freedesktop.Avahi.service".
Disabling 'avahi-daemon.service', but its triggering units are still active:
avahi-daemon.socket
```

Disables the **Bluetooth service** from starting at boot, disables the **CUPS (Common UNIX Printing System)** service & disables the **Avahi daemon**, responsible for **mDNS and service discovery** (used by tools like AirDrop or network printers).

```
admin@Ubuntu:~$ sudo systemctl stop bluetooth
admin@Ubuntu:~$ sudo systemctl stop cups
Stopping 'cups.service', but its triggering units are still active:
cups.socket, cups.path
```

6. Automatic Security Updates

```
admin@Ubuntu:~$ sudo apt install unattended-upgrades -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unattended-upgrades is already the newest version (2.9.1+nmu4ubuntu1).
unattended-upgrades set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 64 not upgraded.
admin@Ubuntu:~$ sudo dpkg-reconfigure --priority=low unattended-upgrades
```

Installs the **Unattended Upgrades** package.

```
admin@Ubuntu:~$ sudo systemctl enable unattended-upgrades
Synchronizing state of unattended-upgrades.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable unattended-upgrades
admin@Ubuntu:~$ sudo systemctl start unattended-upgrades
admin@Ubuntu:~$ sudo systemctl status unattended-upgrades
● unattended-upgrades.service - Unattended Upgrades Shutdown
     Loaded: loaded (/usr/lib/systemd/system/unattended-upgrades.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-10-05 17:04:11 UTC; 41min ago
       Docs: man:unattended-upgrade(8)
   Main PID: 1221 (unattended-upgr)
      Tasks: 2 (limit: 4604)
     Memory: 11.2M (peak: 11.6M)
        CPU: 108ms
     CGroup: /system.slice/unattended-upgrades.service
             └─1221 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal

Oct 05 17:04:11 Ubuntu systemd[1]: Started unattended-upgrades.service - Unattended Upgrades Shutdown.
```

Opens a configuration wizard to **enable and customize automatic updates**.

7. File System Permissions

```
admin@Ubuntu:~$ sudo chmod 600 /etc/shadow
admin@Ubuntu:~$ sudo chmod 644 /etc/passwd
admin@Ubuntu:~$ sudo chmod 750 /home/*
admin@Ubuntu:~$ sudo chmod 640 /var/log/syslog
admin@Ubuntu:~$ sudo chmod 640 /var/log/auth.log
admin@Ubuntu:~$ sudo ls -l /etc/passwd /etc/shadow /etc/sudoers
-rw-r--r-- 1 root root   2952 Oct  5 17:12 /etc/passwd
-rw------- 1 root shadow 1379 Oct  5 17:12 /etc/shadow
-r--r----- 1 root root   1800 Jan 29  2024 /etc/sudoers
```

Restricts read/write access to the /etc/shadow file, sets permissions so the file is **readable by all**, but **writable only by root** & Restricts access to users' home directories.

8. System Security Audit

```
admin@Ubuntu:~$ sudo apt install lynis -y
```

Installs **Lynis**, a powerful open-source security auditing tool.

```
[+] Kernel
------------------------------------
  - Checking default run level                              [ RUNLEVEL 5 ]
  - Checking CPU support (NX/PAE)
    CPU support: PAE and/or NoeXecute supported             [ FOUND ]
  - Checking kernel version and release                     [ DONE ]
  - Checking kernel type                                    [ DONE ]
  - Checking loaded kernel modules                          [ DONE ]
      Found 86 active modules
  - Checking Linux kernel configuration file                [ FOUND ]
  - Checking default I/O kernel scheduler                   [ NOT FOUND ]
  - Checking for available kernel update                    [ OK ]
  - Checking core dumps configuration
    - configuration in systemd conf files                   [ DEFAULT ]
    - configuration in /etc/profile                         [ DEFAULT ]
    - 'hard' configuration in /etc/security/limits.conf     [ DEFAULT ]
    - 'soft' configuration in /etc/security/limits.conf     [ DEFAULT ]
    - Checking setuid core dumps configuration              [ PROTECTED ]
  - Check if reboot is needed                               [ NO ]
```

Runs a **comprehensive security audit** on the system.

9. Application Security (AppArmor)

```
admin@Ubuntu:~$ sudo systemctl enable apparmor
Synchronizing state of apparmor.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apparmor
admin@Ubuntu:~$ sudo systemctl start apparmor
admin@Ubuntu:~$ sudo aa-status
apparmor module is loaded.
152 profiles are loaded.
55 profiles are in enforce mode.
```

Enables **AppArmor** to start automatically at boot, starts the **AppArmor** service immediately

If a service gets hacked, AppArmor keeps it contained in its "jail" so it can't damage the rest of the system.

10. Advanced Auditing

```
admin@Ubuntu:~$ sudo apt install auditd -y
```

Installs the **Linux Audit Daemon (auditd)** package.

```
admin@Ubuntu:~$ sudo systemctl enable auditd
Synchronizing state of auditd.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable auditd
admin@Ubuntu:~$ sudo systemctl start auditd
```

Configures **auditd** to start automatically at system boot.

```
admin@Ubuntu: $ sudo auditctl -w /etc/passwd -p wa -k user_accounts
admin@Ubuntu: $ sudo auditctl -w /etc/shadow -p wa -k user_passwords
admin@Ubuntu: $ sudo auditctl -w /etc/ssh/sshd_config -p wa -k ssh_config
admin@Ubuntu: $ sudo ausearch -k user_accounts
----
time->Sun Oct  5 18:10:24 2025
type=PROCTITLE msg=audit(1759687824.781:287): proctitle=6175646974636C002D77002F6574632F706173737764002D70007761002D6B00757365725F6163636F756E7473
type=SYSCALL msg=audit(1759687824.781:287): arch=c000003e syscall=44 success=yes exit=1080 a0=4 a1=7ffcd14f0b40 a2=438 a3=0 items=0 ppid=141470 pid=141471 auid=1000 uid=0 gid=0 euid=0
suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=3 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1759687824.781:287): auid=1000 ses=3 subj=unconfined op=add_rule key="user_accounts" list=4 res=1
```

Adds a watch rule on the /etc/passwd file.

2. Database Security

2.1 Installation of the Oracle DBMS

Prepared the system environment by creating required Oracle groups and user (oinstall, dba, and oracle).

```
admin@Ubuntu:/tmp$ sudo groupadd -g 54321 oinstall
admin@Ubuntu:/tmp$ sudo groupadd -g 54322 dba
admin@Ubuntu:/tmp$ sudo useradd -u 54321 -g oinstall -G dba oracle
admin@Ubuntu:/tmp$ sudo chown -R oracle:oinstall /opt/oracle
```

Verified the oracle user, created and configured its home directory with default shell configuration files, and set appropriate ownership and permissions to initialize the user environment for Oracle database operations.

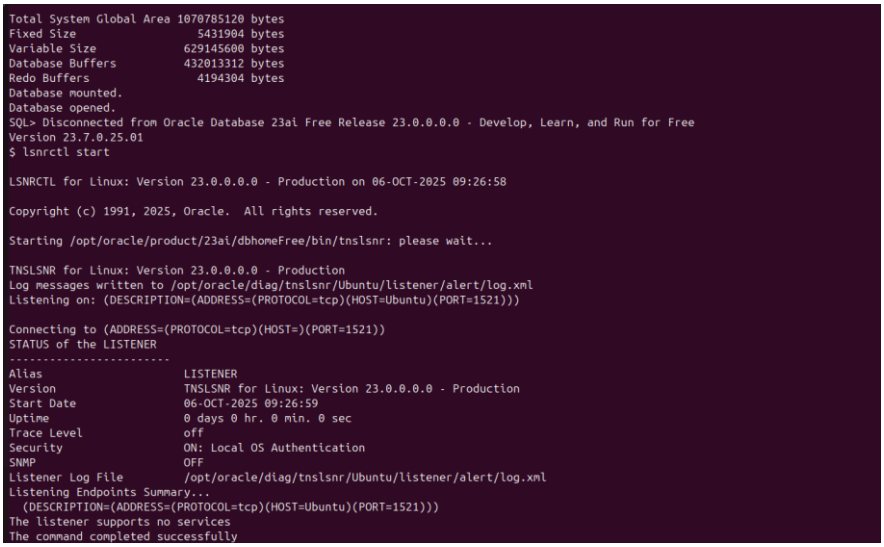
```
admin@Ubuntu:/tmp$ id oracle
uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),54322(dba)
admin@Ubuntu:/tmp$ sudo ls -la /home/oracle/
ls: cannot access '/home/oracle/': No such file or directory
admin@Ubuntu:/tmp$ sudo mkdir -p /home/oracle
admin@Ubuntu:/tmp$ sudo chown oracle:oinstall /home/oracle
admin@Ubuntu:/tmp$ sudo chmod 755 /home/oracle
admin@Ubuntu:/tmp$ sudo cp /etc/skel/.bashrc /home/oracle/
admin@Ubuntu:/tmp$ sudo cp /etc/skel/.profile /home/oracle/
admin@Ubuntu:/tmp$ sudo cp /etc/skel/.bash_logout /home/oracle/ 2>/dev/null || true
admin@Ubuntu:/tmp$ sudo chown oracle:oinstall /home/oracle/.bashrc /home/oracle/.profile /home/oracle/.bash_logout 2>/dev/null
admin@Ubuntu:/tmp$ sudo ls -la /home/oracle/
total 20
drwxr-xr-x 2 oracle oinstall 4096 Oct  6 07:19 .
drwxr-xr-x 4 root   root     4096 Oct  6 07:18 ..
-rw-r--r-- 1 oracle oinstall  220 Oct  6 07:19 .bash_logout
-rw-r--r-- 1 oracle oinstall 3771 Oct  6 07:19 .bashrc
-rw-r--r-- 1 oracle oinstall  807 Oct  6 07:19 .profile
```

Configure Oracle environment variables (ORACLE_BASE, ORACLE_HOME, etc.) for the oracle user.

```
admin@Ubuntu:/tmp$ sudo -u oracle bash -c 'echo "export ORACLE_BASE=/opt/oracle" >> ~/.bashrc'
admin@Ubuntu:/tmp$ sudo -u oracle bash -c 'echo "export ORACLE_HOME=/opt/oracle/product/23ai/dbhomeFree" >> ~/.bashrc'
admin@Ubuntu:/tmp$ sudo -u oracle bash -c 'echo "export ORACLE_SID=FREE" >> ~/.bashrc'
admin@Ubuntu:/tmp$ sudo -u oracle bash -c 'echo "export PATH=\$ORACLE_HOME/bin:\$PATH" >> ~/.bashrc'
admin@Ubuntu:/tmp$ sudo -u oracle bash -c 'echo "export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:\$LD_LIBRARY_PATH" >> ~/.bashrc'
```

Downloaded Oracle Database 23ai Free

Started the Oracle Database instance and listener service, confirming that the database was successfully mounted and opened



2.2 Oracle DBMS Hardening Configurations (5 Configurations)

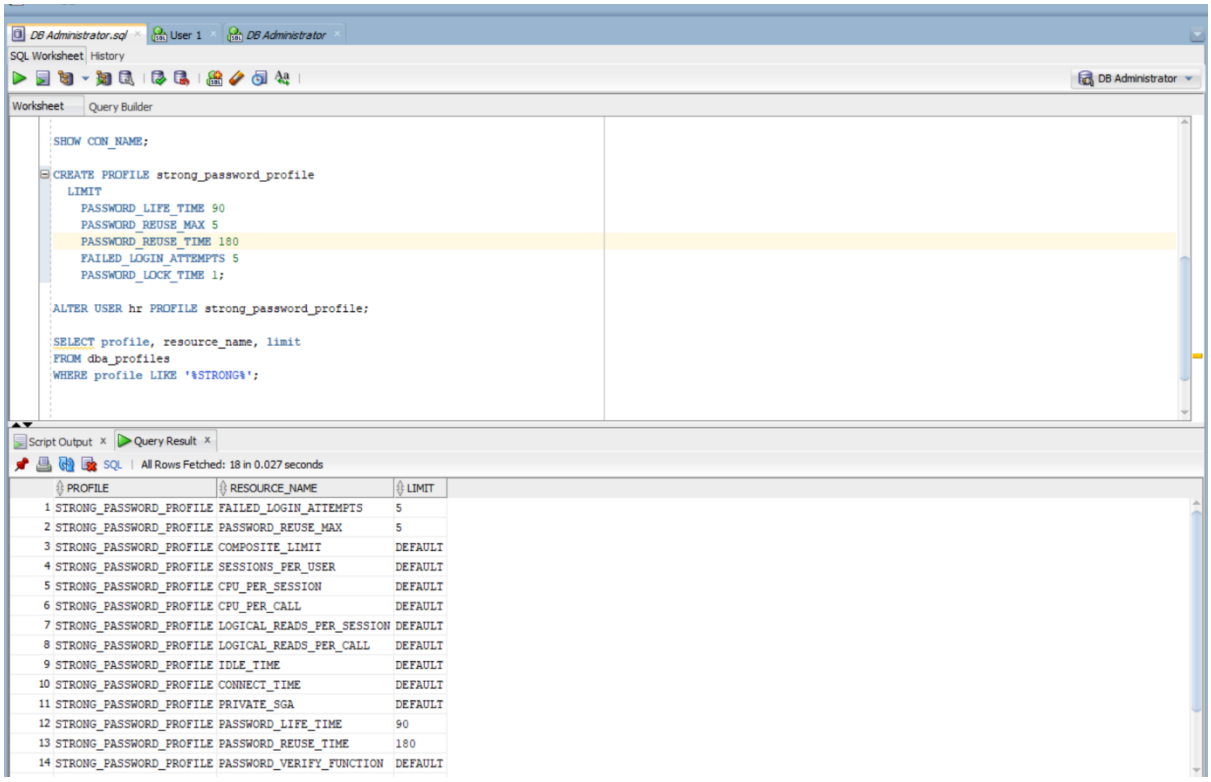
1. Security Configurations and Their Protective Impact

Configuration	Purpose	Attacks Prevented
Enforce Strong Password Policy	Ensures users use complex passwords and limits failed logins	Brute-force and unauthorized access
Enable Network Encryption	Secures data transmitted between client and server	Packet sniffing, MITM attacks
Remove Default Accounts	Disables unused accounts with known credentials	Default account exploitation
Enable Listener Administration Restrictions	Limits who can modify or stop the listener	Listener hijacking, DoS attacks
Enable Auditing	Tracks user activities and access to detect suspicious behavior	Insider threats and unauthorized data access

2. Implementation of configurations

1. Enforce Strong Password Policy

Created a profile with strong password setting and assigned profile to all users.



Verified the strong password policy.

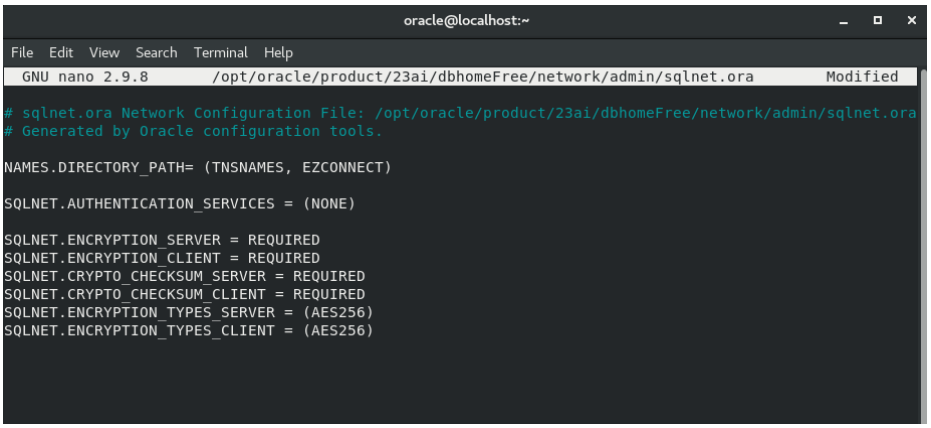
```
SQL> SHOW CON_NAME;

CON_NAME
------------------------------
FREEPDB1
SQL> CREATE PROFILE strong_password_profile
  LIMIT
    PASSWORD_LIFE_TIME 90
    PASSWORD_REUSE_MAX 5
    PASSWORD_REUSE_TIME 180
    FAILED_LOGIN_ATTEMPTS 5
    PASSWORD_LOCK_TIME 1;
  2    3    4    5    6    7  CREATE PROFILE strong_password_profile
*
ERROR at line 1:
ORA-02379: profile STRONG_PASSWORD_PROFILE already exists
Help: https://docs.oracle.com/error-help/db/ora-02379/


SQL> ALTER USER hr PROFILE strong_password_profile;

User altered.
```

2. Enable Network Encryption

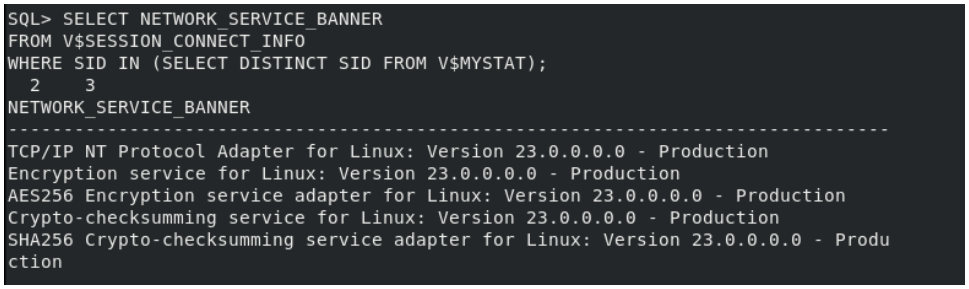
Opened the SQLNET.ORA file.

```
[oracle@localhost ~]$ sudo nano /opt/oracle/product/23ai/dbhomeFree/network/admin/sqlnet.ora
[sudo] password for oracle:
```

Added network encryption and integrity settings and saved it.



Verified encryption is active.



3. Remove Default Accounts

Check which users actually exist.

Look for any accounts that are unneeded or default accounts, such as `ANONYMOUS`, `XDB`, or other demo users.

```
SQL> SELECT username, account_status FROM dba_users;

USERNAME
--------------------------------------------------------------------------------
ACCOUNT_STATUS
--------------------------------
SYS
OPEN

APEX_PUBLIC_USER
OPEN

APEX_PUBLIC_ROUTER
OPEN


USERNAME
--------------------------------------------------------------------------------
ACCOUNT_STATUS
--------------------------------
AV
OPEN

PDBADMIN
OPEN

SYSRAC
OPEN


USERNAME
--------------------------------------------------------------------------------
ACCOUNT_STATUS
--------------------------------
ORDS_PUBLIC_USER
OPEN

ORDS_METADATA
OPEN

CO
OPEN
```

Lock accounts that exist and are unnecessary:

```
LOCKED


47 rows selected.

SQL> ALTER USER anonymous ACCOUNT LOCK;

User altered.
```

4. Enable Listener Administration Restrictions

Opened listener configuration file.

```
[oracle@localhost ~]$ sudo nano /opt/oracle/product/23ai/dbhomeFree/network/admin/listener.ora
[sudo] password for oracle:
```

Added ‘ADMIN_RESTRICTION_LISTENER = ON’ line at the bottom and saved the file.

```
oracle@localhost:~
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.8      /opt/oracle/product/23ai/dbhomeFree/network/admin/listener.ora       Modified

# listener.ora Network Configuration File: /opt/oracle/product/23ai/dbhomeFree/network/admin/listene$
# Generated by Oracle configuration tools.

DEFAULT_SERVICE_LISTENER = FREE

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )


SID_LIST_LISTENER =
 (SID_LIST =
 (SID_DESC =
 (GLOBAL_DBNAME = FREEPDB1)
 (SID_NAME = free)
 (ORACLE_HOME = /opt/oracle/product/23ai/dbhomeFree )
 )
 )

ADMIN_RESTRICTIONS_LISTENER = ON
```

Restricted OS access to listener configuration files.

```
[oracle@localhost ~]$ sudo chmod 600 /opt/oracle/product/23ai/dbhomeFree/network/admin/listener.ora
[oracle@localhost ~]$ sudo chown oracle:oinstall /opt/oracle/product/23ai/dbhomeFree/network/admin/li
stener.ora
```

Restricted listener commands to Oracle OS user.

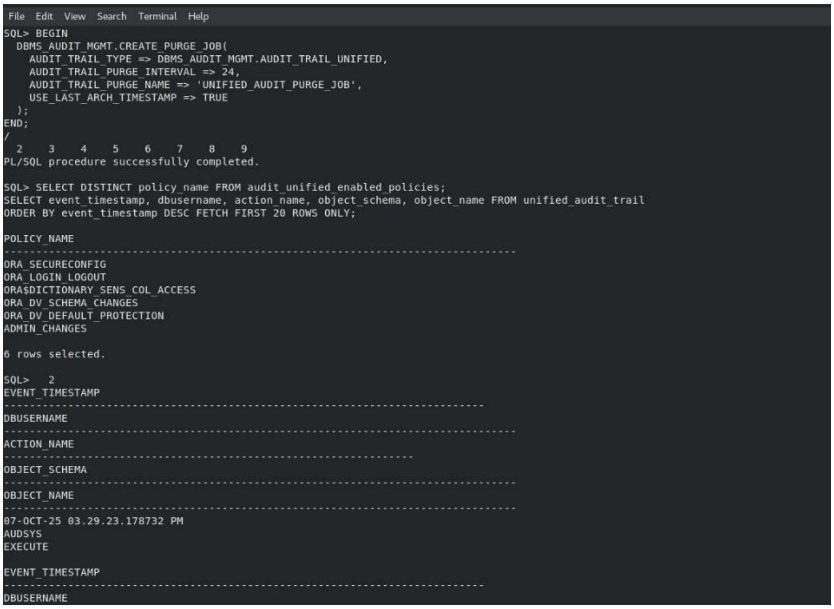
```
[oracle@localhost ~]$ sudo chmod 750 /opt/oracle/product/23ai/dbhomeFree/bin/lsnrctl
[oracle@localhost ~]$ sudo chown oracle:oinstall /opt/oracle/product/23ai/dbhomeFree/bin/lsnrctl
```

5. Enable Auditing

Checked unified auditing support and enabled recommended Oracle policies (predefined) and a custom admin policy

```
SQL> SELECT PARAMETER, VALUE FROM V$OPTION WHERE PARAMETER='Unified Auditing';

PARAMETER
----------------------------------------------------------------
VALUE
----------------------------------------------------------------
Unified Auditing
TRUE


SQL> -- Enable Oracle predefined policies (if available)
AUDIT POLICY ORA_SECURECONFIG;
AUDIT POLICY ORA_LOGIN_LOGOUT;

-- Create a focused admin-audit policy
CREATE AUDIT POLICY admin_changes
  PRIVILEGES CREATE USER, DROP USER, ALTER SYSTEM, GRANT ANY PRIVILEGE;
AUDIT POLICY admin_changes;
SQL>
Audit succeeded.

SQL>
Audit succeeded.

SQL> SQL> SQL>   2
Audit policy created.

SQL>
Audit succeeded.

SQL> BEGIN
  DBMS_AUDIT_MGMT.CREATE_PURGE_JOB(
    AUDIT_TRAIL_TYPE => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    AUDIT_TRAIL_PURGE_INTERVAL => 24,
    AUDIT_TRAIL_PURGE_NAME => 'UNIFIED_AUDIT_PURGE_JOB',
```

Directed unified audit trail retention/purge with DBMS_AUDIT_MGMT.

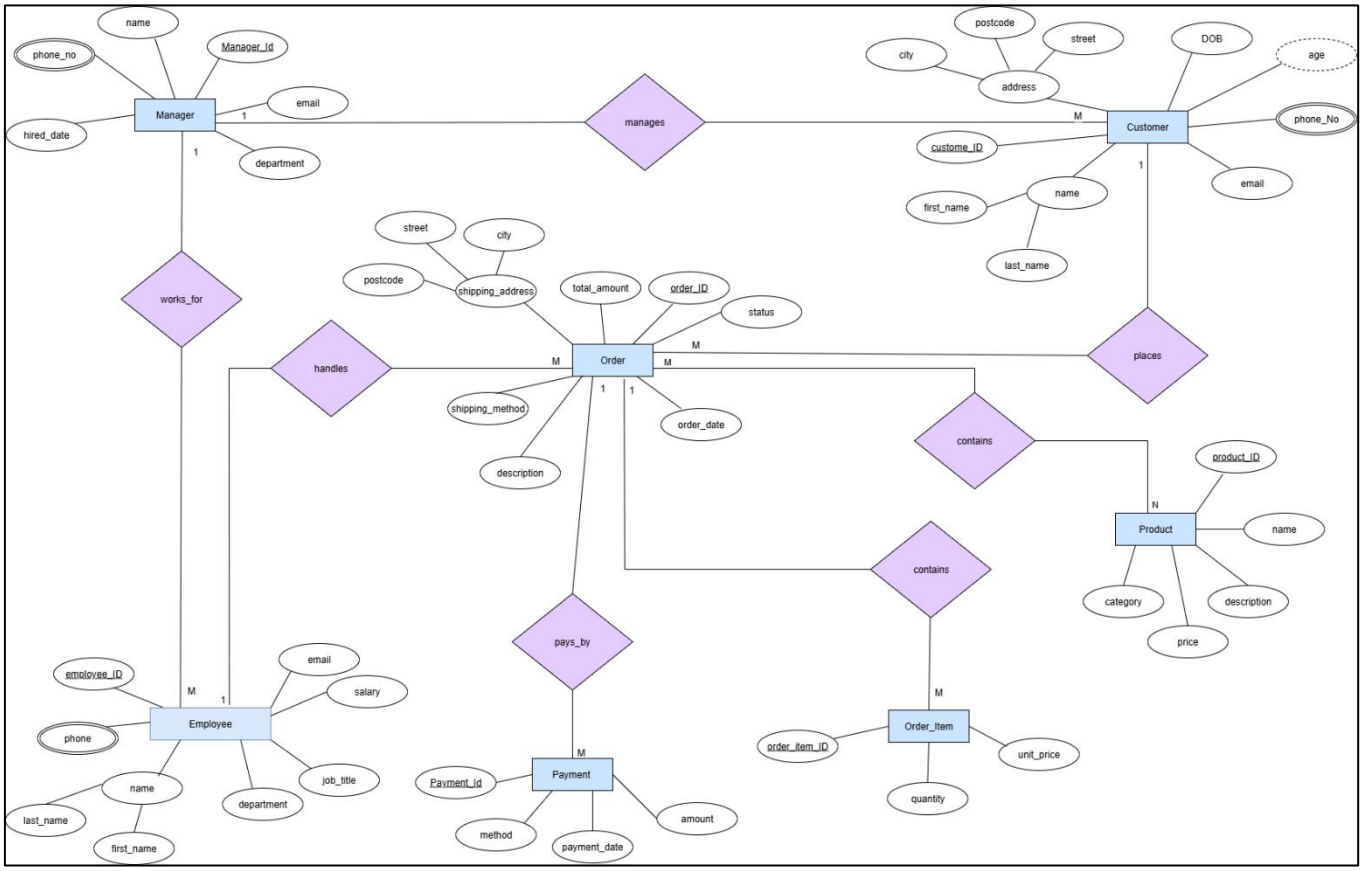


```
File  Edit  View  Search  Terminal  Help
SQL> BEGIN
  DBMS_AUDIT_MGMT.CREATE_PURGE_JOB(
    AUDIT_TRAIL_TYPE => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    AUDIT_TRAIL_PURGE_INTERVAL => 24,
    AUDIT_TRAIL_PURGE_NAME => 'UNIFIED_AUDIT_PURGE_JOB',
    USE_LAST_ARCH_TIMESTAMP => TRUE
  );
END;
/
  2    3    4    5    6    7    8    9
PL/SQL procedure successfully completed.

SQL> SELECT DISTINCT policy_name FROM audit_unified_enabled_policies;
SELECT event_timestamp, dbusername, action_name, object_schema, object_name FROM unified_audit_trail
ORDER BY event_timestamp DESC FETCH FIRST 20 ROWS ONLY;

POLICY_NAME
--------------------------------------------------------------------------------
ORA_SECURECONFIG
ORA_LOGIN_LOGOUT
ORA$DICTIONARY_SENS_COL_ACCESS
ORA_DV_SCHEMA_CHANGES
ORA_DV_DEFAULT_PROTECTION
ADMIN_CHANGES

6 rows selected.

SQL>   2
EVENT_TIMESTAMP
---------------------------------------------------------------------------
DBUSERNAME
--------------------------------------------------------------------------------
ACTION_NAME
----------------------------------------------------------------
OBJECT_SCHEMA
--------------------------------------------------------------------------------
OBJECT_NAME
--------------------------------------------------------------------------------
07-OCT-25 03.29.23.178732 PM
AUDSYS
EXECUTE

EVENT_TIMESTAMP
---------------------------------------------------------------------------
DBUSERNAME
```

```
SQL>   2
EVENT_TIMESTAMP
---------------------------------------------------------------------------
DBUSERNAME
--------------------------------------------------------------------------------
ACTION_NAME
----------------------------------------------------------------
OBJECT_SCHEMA
--------------------------------------------------------------------------------
OBJECT_NAME
--------------------------------------------------------------------------------
07-OCT-25 03.29.23.178732 PM
AUDSYS
EXECUTE

EVENT_TIMESTAMP
---------------------------------------------------------------------------
DBUSERNAME
--------------------------------------------------------------------------------
ACTION_NAME
----------------------------------------------------------------
OBJECT_SCHEMA
--------------------------------------------------------------------------------
OBJECT_NAME
--------------------------------------------------------------------------------
AUDSYS
DBMS_AUDIT_MGMT


EVENT_TIMESTAMP
---------------------------------------------------------------------------
DBUSERNAME
--------------------------------------------------------------------------------
ACTION_NAME
----------------------------------------------------------------
```

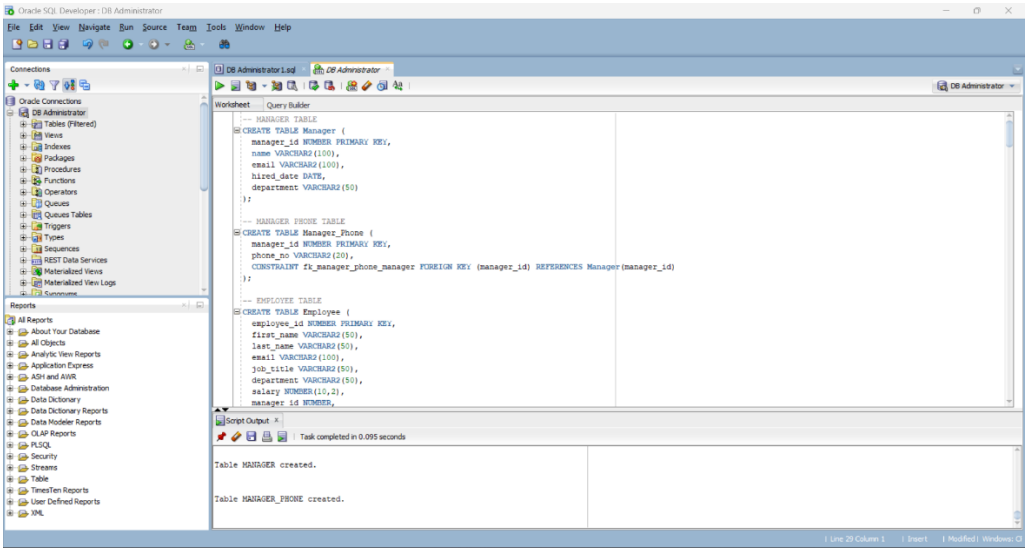
3.Database Design – Customer Management System (ERD)

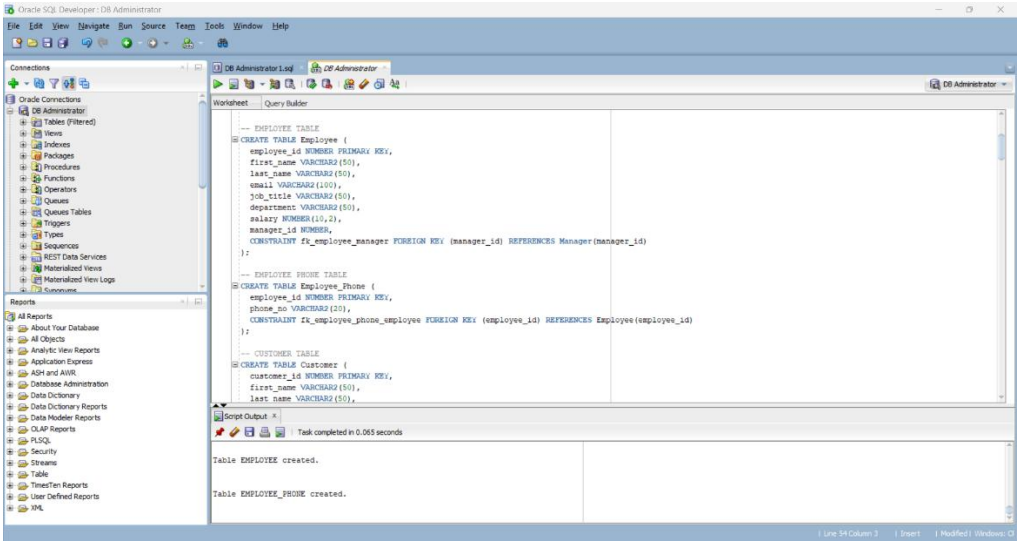
3.1 ERD

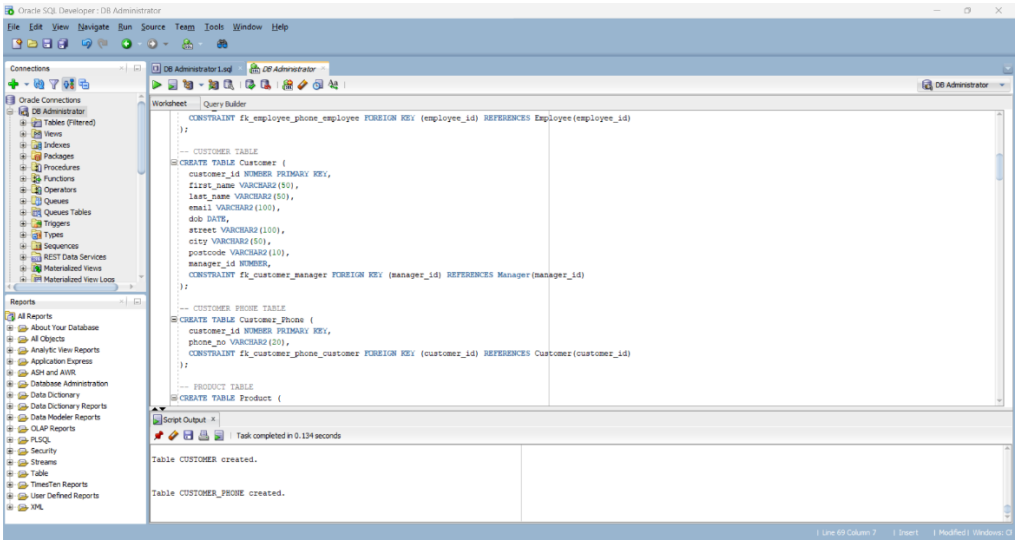

name
phone_no
Manager_Id
Manager
email
hired_date
department
1
1
manages
M
postcode
street
DOB
city
age
address
Customer
phone_No
custome_ID
email
first_name
name
last_name
1
street
city
postcode
shipping_address
total_amount
order_ID
status
works_for
handles
M
Order
M
M
places
shipping_method
1
1
order_date
description
contains
product_ID
N
Product
name
category
description
price
contains
pays_by
email
employee_ID
salary
M
1
Employee
phone
name
job_title
last_name
department
first_name
Payment_Id
M
Payment
method
payment_date
amount
M
order_item_ID
Order_Item
unit_price
quantity

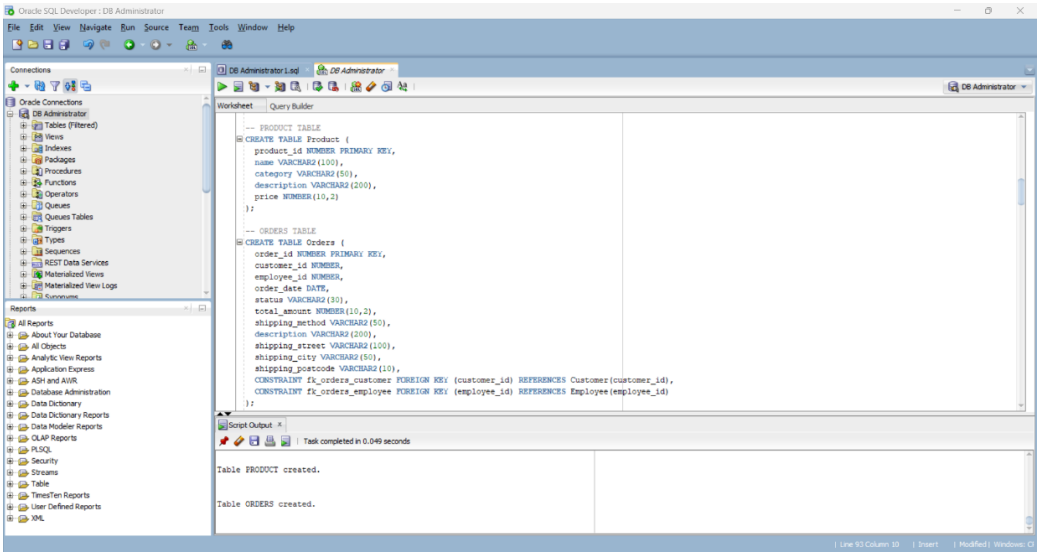
3.2 Table Definitions

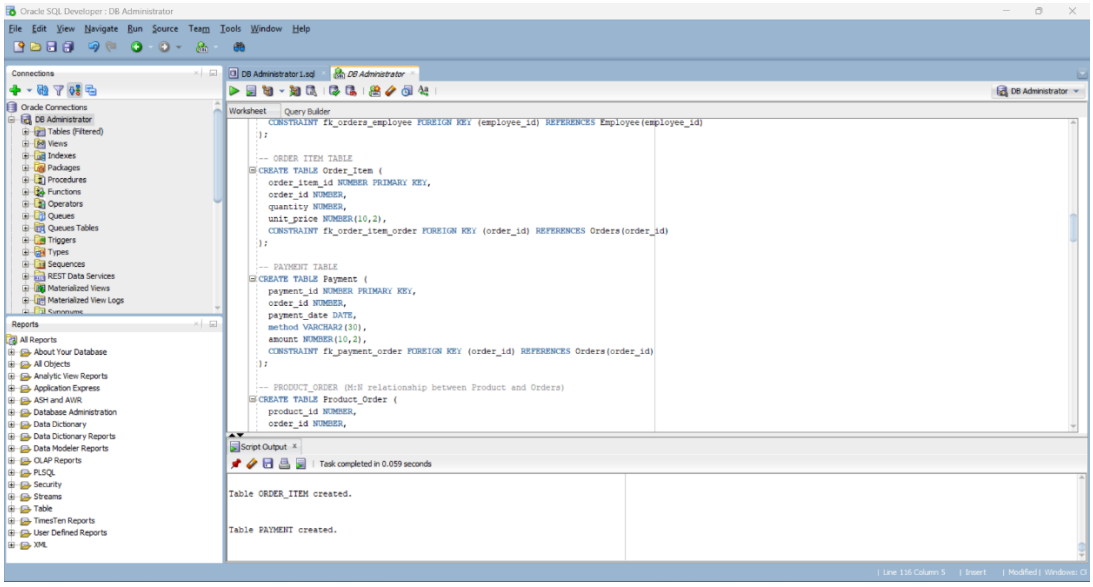
SQL table creation.

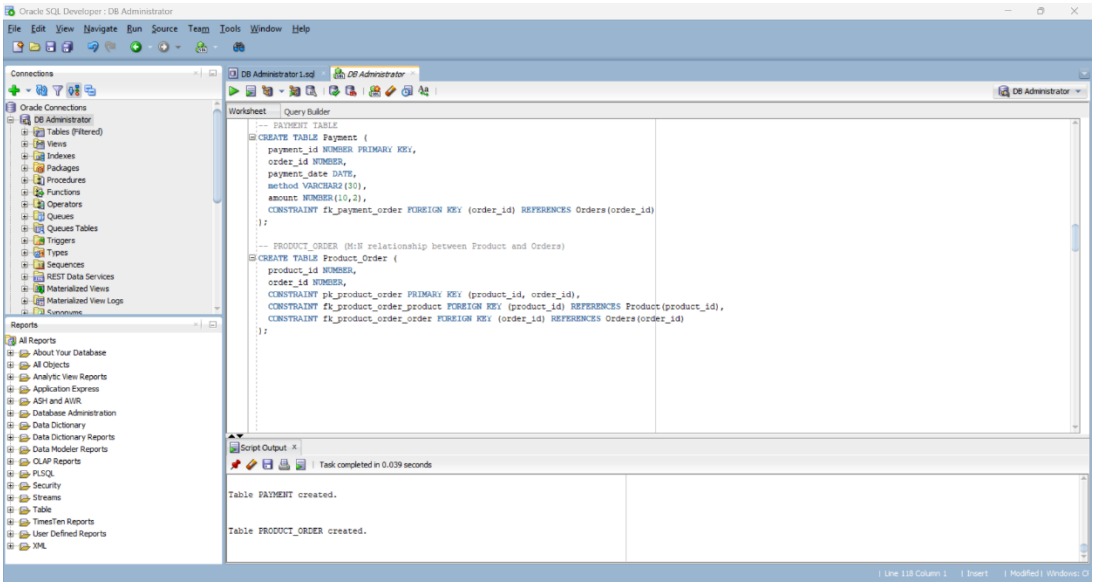




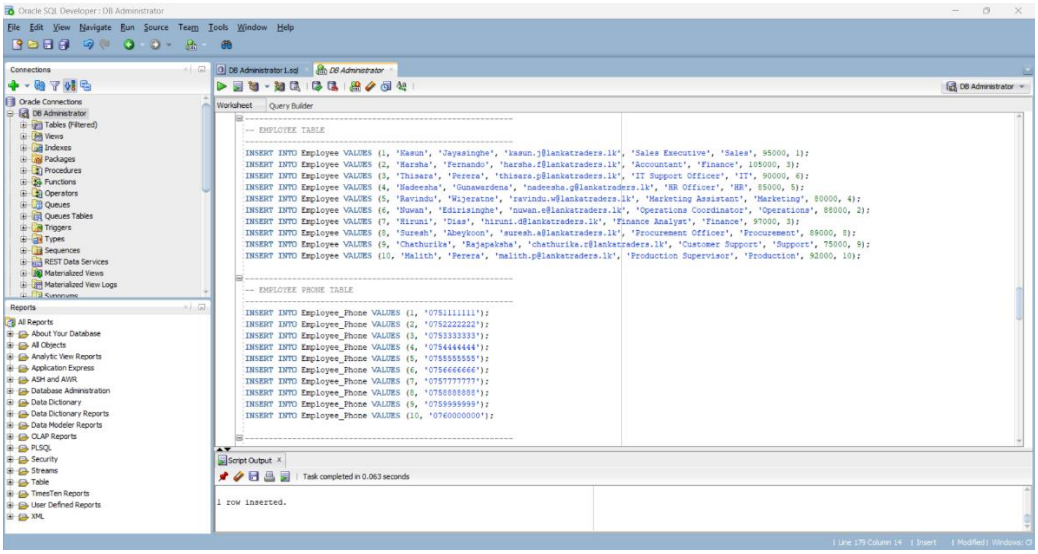


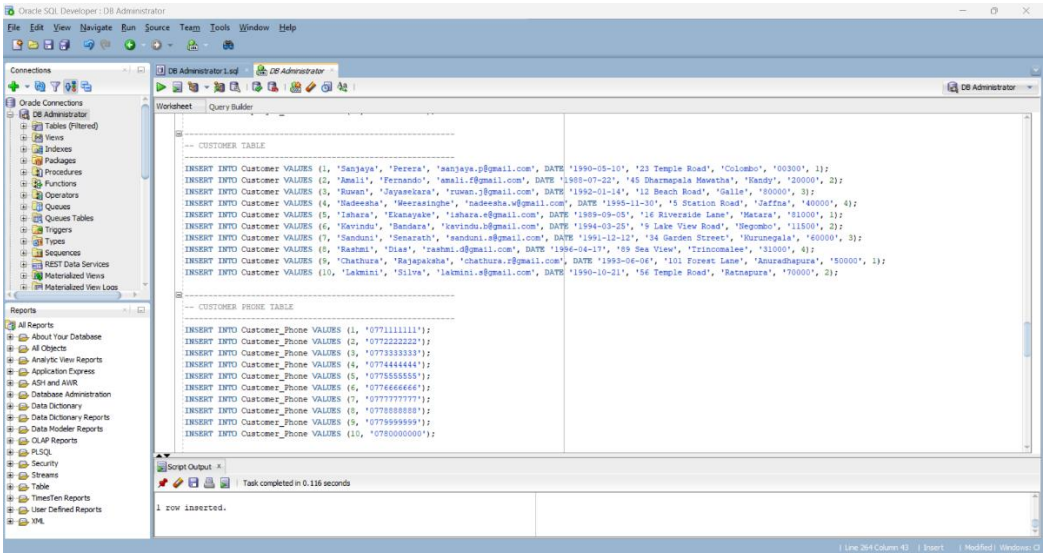


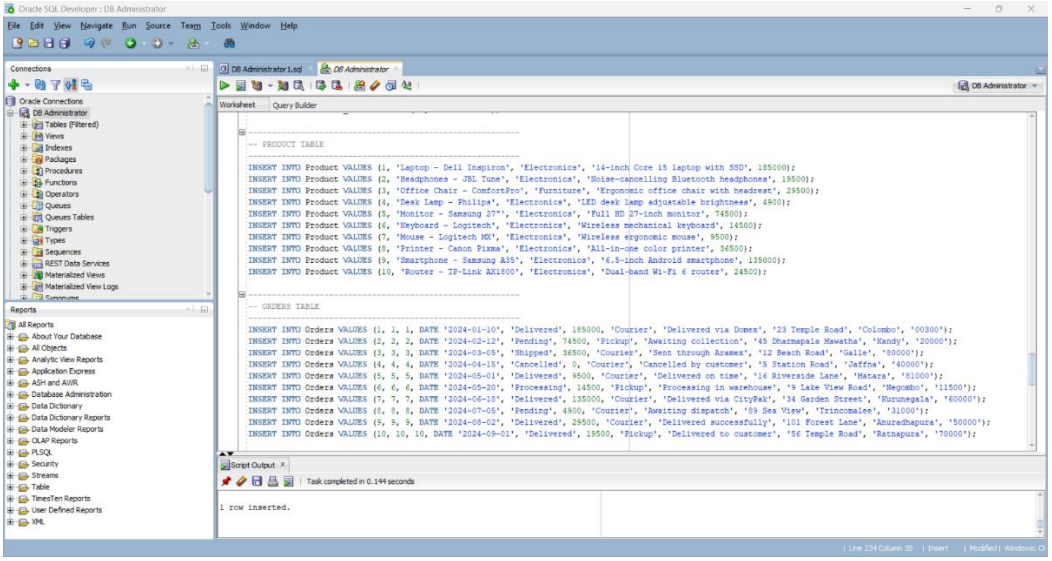


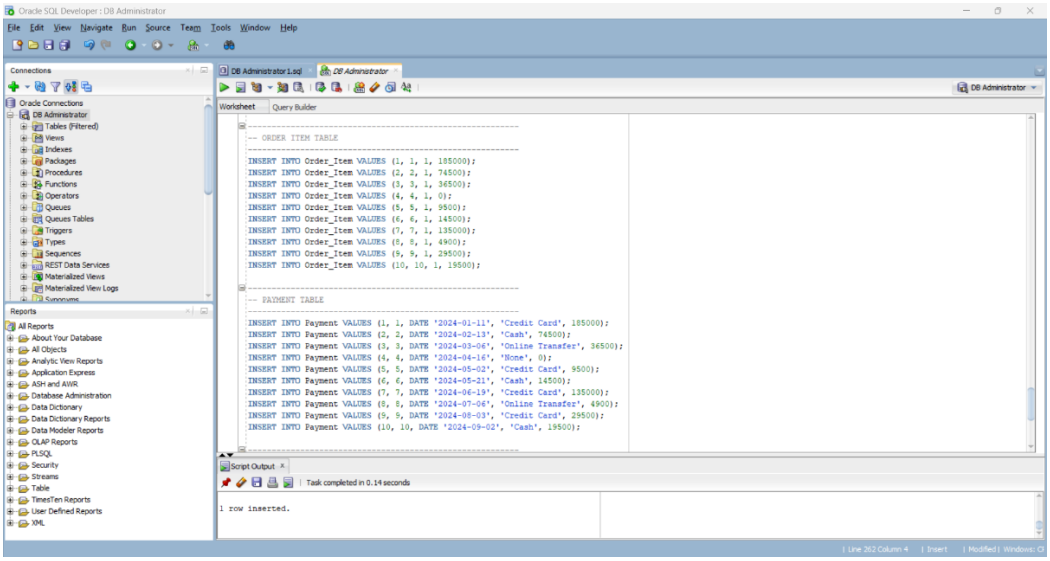


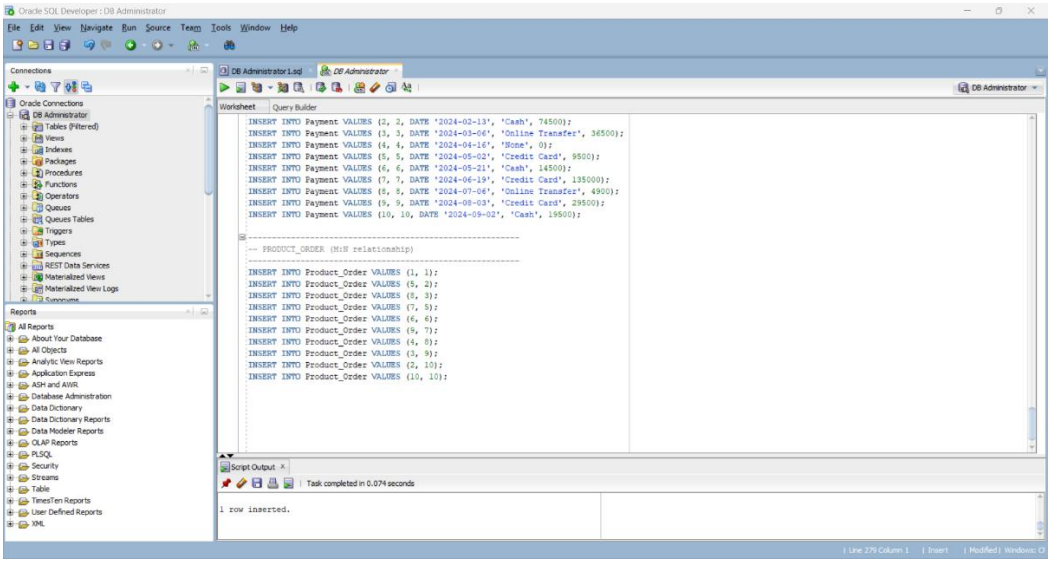
Sample data insertion.


-- EMPLOYEE TABLE
INSERT INTO Employee VALUES (1, 'Kasun', 'Jayasinghe', 'kasun.j@lankatraders.lk', 'Sales Executive', 'Sales', 95000, 1);
INSERT INTO Employee VALUES (2, 'Harsha', 'Fernando', 'harsha.f@lankatraders.lk', 'Accountant', 'Finance', 105000, 3);
INSERT INTO Employee VALUES (3, 'Thisara', 'Perera', 'thisara.p@lankatraders.lk', 'IT Support Officer', 'IT', 90000, 6);
INSERT INTO Employee VALUES (4, 'Nadeesha', 'Gunawardena', 'nadeesha.g@lankatraders.lk', 'HR Officer', 'HR', 85000, 5);
INSERT INTO Employee VALUES (5, 'Ravindu', 'Wijeratne', 'ravindu.w@lankatraders.lk', 'Marketing Assistant', 'Marketing', 80000, 4);
INSERT INTO Employee VALUES (6, 'Nuwan', 'Edirisinghe', 'nuwan.e@lankatraders.lk', 'Operations Coordinator', 'Operations', 88000, 2);
INSERT INTO Employee VALUES (7, 'Hiruni', 'Dias', 'hiruni.d@lankatraders.lk', 'Finance Analyst', 'Finance', 97000, 3);
INSERT INTO Employee VALUES (8, 'Suresh', 'Abeykoon', 'suresh.a@lankatraders.lk', 'Procurement Officer', 'Procurement', 89000, 8);
INSERT INTO Employee VALUES (9, 'Chathurika', 'Rajapaksha', 'chathurika.r@lankatraders.lk', 'Customer Support', 'Support', 75000, 9);
INSERT INTO Employee VALUES (10, 'Malith', 'Perera', 'malith.p@lankatraders.lk', 'Production Supervisor', 'Production', 92000, 10);
-- EMPLOYEE PHONE TABLE
INSERT INTO Employee_Phone VALUES (1, '0751111111');
INSERT INTO Employee_Phone VALUES (2, '0752222222');
INSERT INTO Employee_Phone VALUES (3, '0753333333');
INSERT INTO Employee_Phone VALUES (4, '0754444444');
INSERT INTO Employee_Phone VALUES (5, '0755555555');
INSERT INTO Employee_Phone VALUES (6, '0756666666');
INSERT INTO Employee_Phone VALUES (7, '0757777777');
INSERT INTO Employee_Phone VALUES (8, '0758888888');
INSERT INTO Employee_Phone VALUES (9, '0759999999');
INSERT INTO Employee_Phone VALUES (10, '0740000000');
1 row inserted.

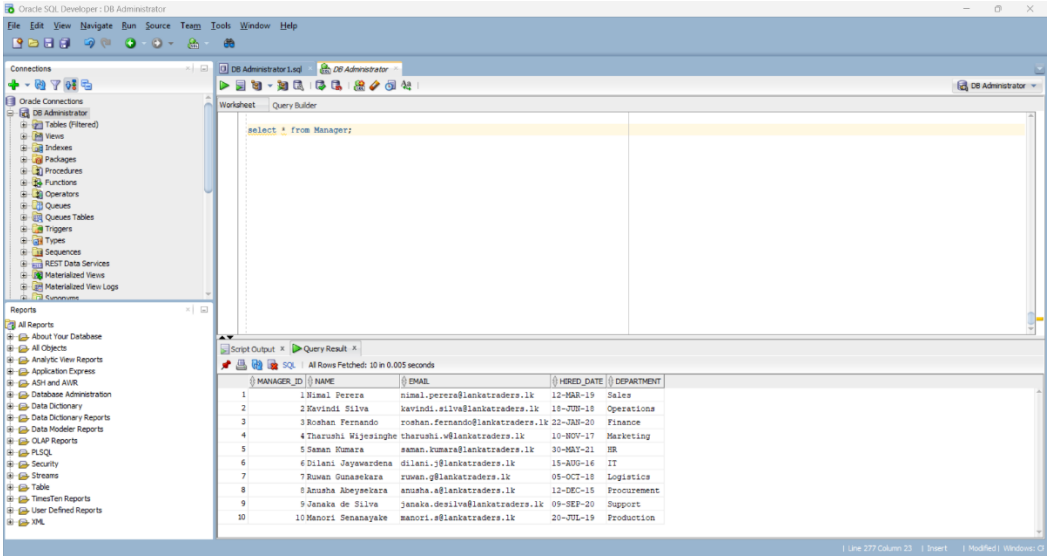

-- CUSTOMER TABLE
INSERT INTO Customer VALUES (1, 'Sanjaya', 'Perera', 'sanjaya.p@gmail.com', DATE '1990-05-10', '23 Temple Road', 'Colombo', '00300', 1);
INSERT INTO Customer VALUES (2, 'Amali', 'Fernando', 'amali.f@gmail.com', DATE '1988-07-22', '45 Dharmapala Mawatha', 'Kandy', '20000', 2);
INSERT INTO Customer VALUES (3, 'Ruwan', 'Jayasekara', 'ruwan.j@gmail.com', DATE '1992-01-14', '12 Beach Road', 'Galle', '80000', 3);
INSERT INTO Customer VALUES (4, 'Nadeesha', 'Weerasinghe', 'nadeesha.w@gmail.com', DATE '1995-11-30', '5 Station Road', 'Jaffna', '40000', 4);
INSERT INTO Customer VALUES (5, 'Ishara', 'Ekanayake', 'ishara.e@gmail.com', DATE '1989-09-05', '16 Riverside Lane', 'Matara', '81000', 1);
INSERT INTO Customer VALUES (6, 'Ravindu', 'Bandara', 'ravindu.b@gmail.com', DATE '1994-03-25', '9 Lake View Road', 'Negombo', '11500', 2);
INSERT INTO Customer VALUES (7, 'Sanduni', 'Senarath', 'sanduni.s@gmail.com', DATE '1991-12-12', '34 Garden Street', 'Kurunegala', '60000', 3);
INSERT INTO Customer VALUES (8, 'Rashmi', 'Dias', 'rashmi.d@gmail.com', DATE '1996-04-17', '89 Sea View', 'Trincomalee', '31000', 4);
INSERT INTO Customer VALUES (9, 'Chathura', 'Rajapaksha', 'chathura.r@gmail.com', DATE '1993-06-06', '101 Forest Lane', 'Anuradhapura', '50000', 1);
INSERT INTO Customer VALUES (10, 'Lakmini', 'Silva', 'lakmini.s@gmail.com', DATE '1990-10-21', '56 Temple Road', 'Ratnapura', '70000', 2);
-- CUSTOMER PHONE TABLE
INSERT INTO Customer_Phone VALUES (1, '0771111111');
INSERT INTO Customer_Phone VALUES (2, '0772222222');
INSERT INTO Customer_Phone VALUES (3, '0773333333');
INSERT INTO Customer_Phone VALUES (4, '0774444444');
INSERT INTO Customer_Phone VALUES (5, '0775555555');
INSERT INTO Customer_Phone VALUES (6, '0776666666');
INSERT INTO Customer_Phone VALUES (7, '0777777777');
INSERT INTO Customer_Phone VALUES (8, '0778888888');
INSERT INTO Customer_Phone VALUES (9, '0779999999');
INSERT INTO Customer_Phone VALUES (10, '0780000000');
1 row inserted.


-- PRODUCT TABLE
INSERT INTO Product VALUES (1, 'Laptop - Dell Inspiron', 'Electronics', '14-inch Core i5 laptop with SSD', 185000);
INSERT INTO Product VALUES (2, 'Headphones - JBL Tune', 'Electronics', 'Noise-cancelling Bluetooth headphones', 19500);
INSERT INTO Product VALUES (3, 'Office Chair - ComfortPro', 'Furniture', 'Ergonomic office chair with headrest', 29500);
INSERT INTO Product VALUES (4, 'Desk Lamp - Philips', 'Electronics', 'LED desk lamp adjustable brightness', 4900);
INSERT INTO Product VALUES (5, 'Monitor - Samsung 27"', 'Electronics', 'Full HD 27-inch monitor', 74500);
INSERT INTO Product VALUES (6, 'Keyboard - Logitech', 'Electronics', 'Wireless mechanical keyboard', 14500);
INSERT INTO Product VALUES (7, 'Mouse - Logitech MX', 'Electronics', 'Wireless ergonomic mouse', 9500);
INSERT INTO Product VALUES (8, 'Printer - Canon Pixma', 'Electronics', 'All-in-one color printer', 36500);
INSERT INTO Product VALUES (9, 'Smartphone - Samsung A35', 'Electronics', '6.5-inch Android smartphone', 135000);
INSERT INTO Product VALUES (10, 'Router - TP-Link AX1800', 'Electronics', 'Dual-band Wi-Fi 6 router', 24500);
-- ORDERS TABLE
INSERT INTO Orders VALUES (1, 1, 1, DATE '2024-01-10', 'Delivered', 185000, 'Courier', 'Delivered via Domex', '23 Temple Road', 'Colombo', '00300');
INSERT INTO Orders VALUES (2, 2, 2, DATE '2024-02-12', 'Pending', 74500, 'Pickup', 'Awaiting collection', '45 Dharmapala Mawatha', 'Kandy', '20000');
INSERT INTO Orders VALUES (3, 3, 3, DATE '2024-03-05', 'Shipped', 36500, 'Courier', 'Sent through Aramex', '12 Beach Road', 'Galle', '80000');
INSERT INTO Orders VALUES (4, 4, 4, DATE '2024-04-15', 'Cancelled', 0, 'Courier', 'Cancelled by customer', '5 Station Road', 'Jaffna', '40000');
INSERT INTO Orders VALUES (5, 5, 5, DATE '2024-05-01', 'Delivered', 9500, 'Courier', 'Delivered on time', '16 Riverside Lane', 'Matara', '81000');
INSERT INTO Orders VALUES (6, 6, 6, DATE '2024-05-20', 'Processing', 14500, 'Pickup', 'Processing in warehouse', '9 Lake View Road', 'Negombo', '11500');
INSERT INTO Orders VALUES (7, 7, 7, DATE '2024-06-18', 'Delivered', 135000, 'Courier', 'Delivered via CityPak', '34 Garden Street', 'Kurunegala', '60000');
INSERT INTO Orders VALUES (8, 8, 8, DATE '2024-07-05', 'Pending', 4900, 'Courier', 'Awaiting dispatch', '89 Sea View', 'Trincomalee', '31000');
INSERT INTO Orders VALUES (9, 9, 9, DATE '2024-08-02', 'Delivered', 29500, 'Courier', 'Delivered successfully', '101 Forest Lane', 'Anuradhapura', '50000');
INSERT INTO Orders VALUES (10, 10, 10, DATE '2024-09-01', 'Delivered', 18500, 'Pickup', 'Delivered to customer', '56 Temple Road', 'Ratnapura', '70000');
1 row inserted.





Successfully implemented all required tables and records for the Customer Management System.



4.User Roles and Permissions

User Creation and Permission Allocation

```
SQL> ALTER SESSION SET CONTAINER = mypdb;

Session altered.
```

The session has been set to mypdb because user accounts created inside a pluggable database are **local to that PDB** and will only have access to objects inside it.

```
SQL> CREATE USER sysadmin IDENTIFIED BY SysAdmin123;

User created.

SQL> GRANT DBA TO sysadmin;
```

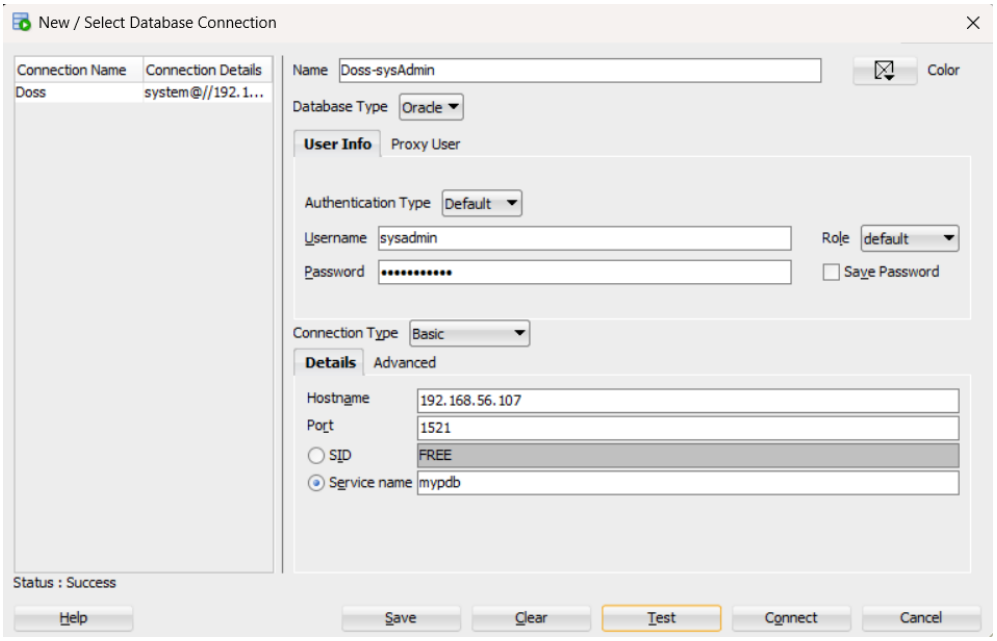
Full privileges for system administration.

```
SQL> CREATE USER manager IDENTIFIED BY Manager123;

User created.

SQL> GRANT CONNECT, RESOURCE TO manager;

Grant succeeded.

SQL> ALTER USER manager QUOTA UNLIMITED ON users;

User altered.
```

Operational access to manage tables (read/write).

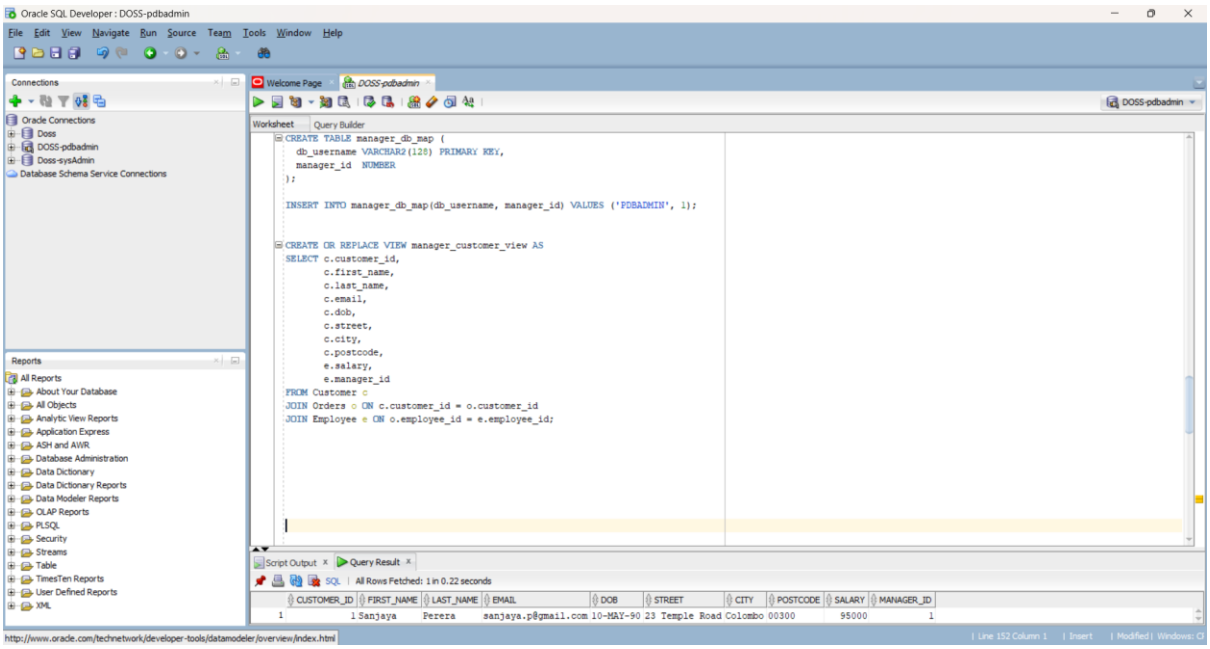
```
SQL> CREATE USER executive IDENTIFIED BY Executive123;

User created.

SQL> GRANT CONNECT TO executive;

Grant succeeded.

SQL> ALTER USER executive QUOTA UNLIMITED ON users;

User altered.
```

Read-only access to view data.



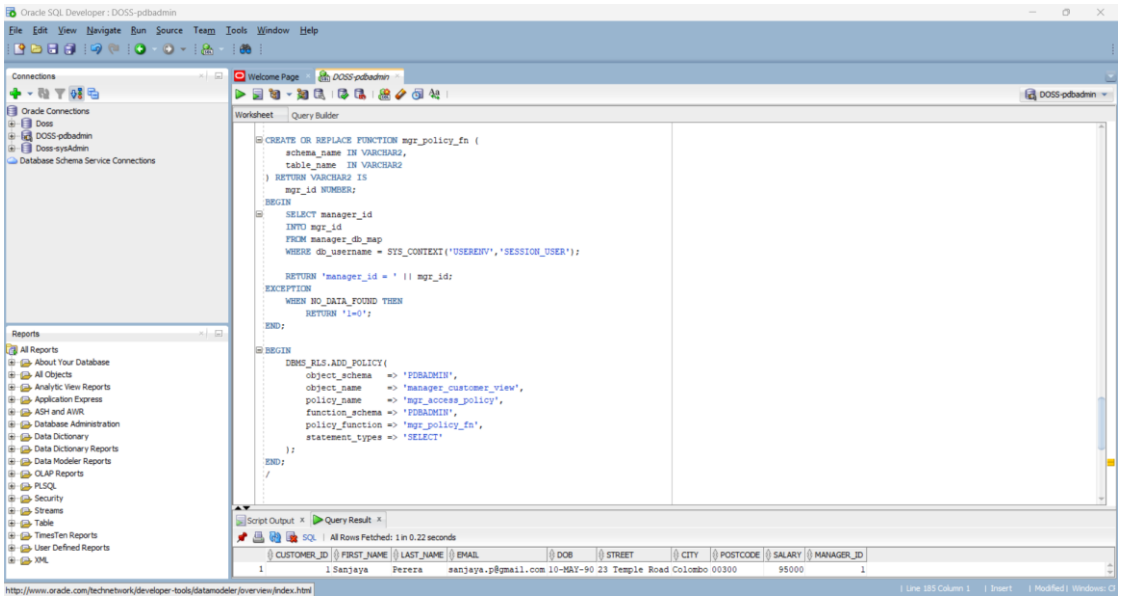
5.View Creation (Manager Access)

Created a **view** for managers to view customer details they manage.



6. VDP Creation That Matches The Functionality Of The View

```
CREATE TABLE manager_db_map (
  db_username VARCHAR2(128) PRIMARY KEY,
  manager_id  NUMBER
);

INSERT INTO manager_db_map(db_username, manager_id) VALUES ('PDBADMIN', 1);
```




```
SELECT * FROM manager_customer_view;
```

Script Output | Query Result

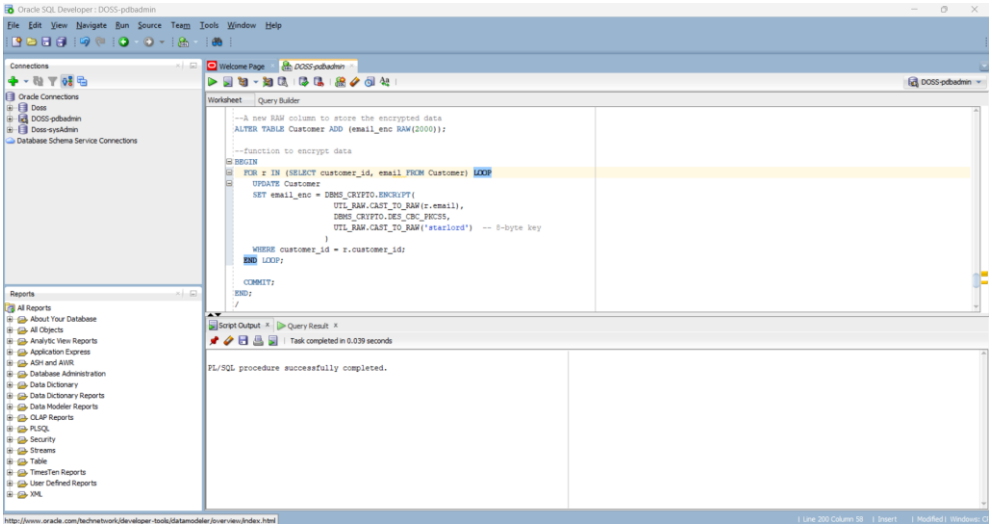
SQL | All Rows Fetched: 1 in 0.006 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	DOB	STREET	CITY	POSTCODE	SALARY	MANAGER_ID
1	1	Sanjaya	Perera	sanjaya.p@gmail.com	10-MAY-90	23 Temple Road	Colombo	00300	95000	1

7. Encryption of Sensitive Data

```
SQL> SELECT object_name, object_type, status
FROM dba_objects
WHERE object_name='DBMS_CRYPTO';
  2    3
OBJECT_NAME
--------------------------------------------------------------------------------
OBJECT_TYPE             STATUS
----------------------- -------
DBMS_CRYPTO
PACKAGE                 VALID

DBMS_CRYPTO
PACKAGE BODY            VALID

DBMS_CRYPTO
SYNONYM                 VALID
```

The DBMS_CRYPTO package was used for the encryption



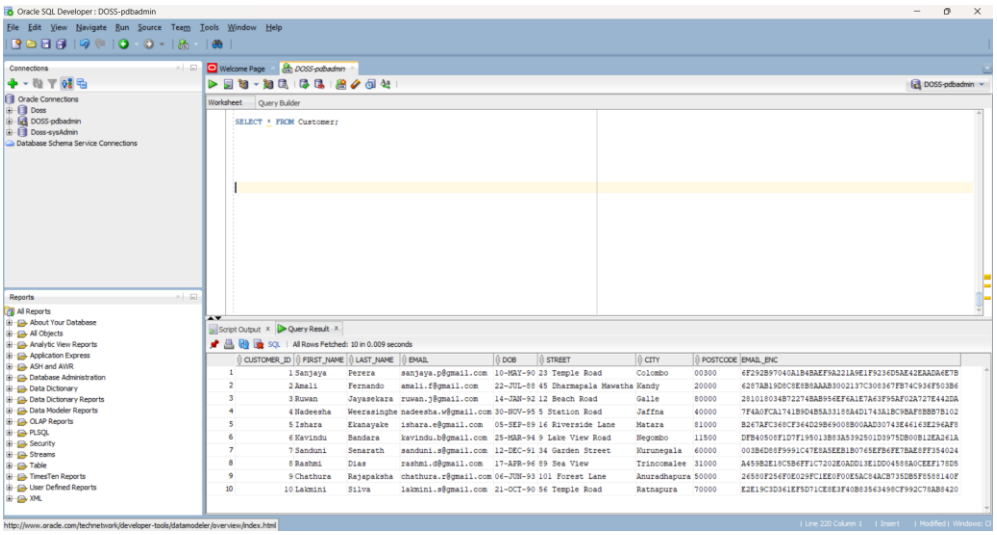
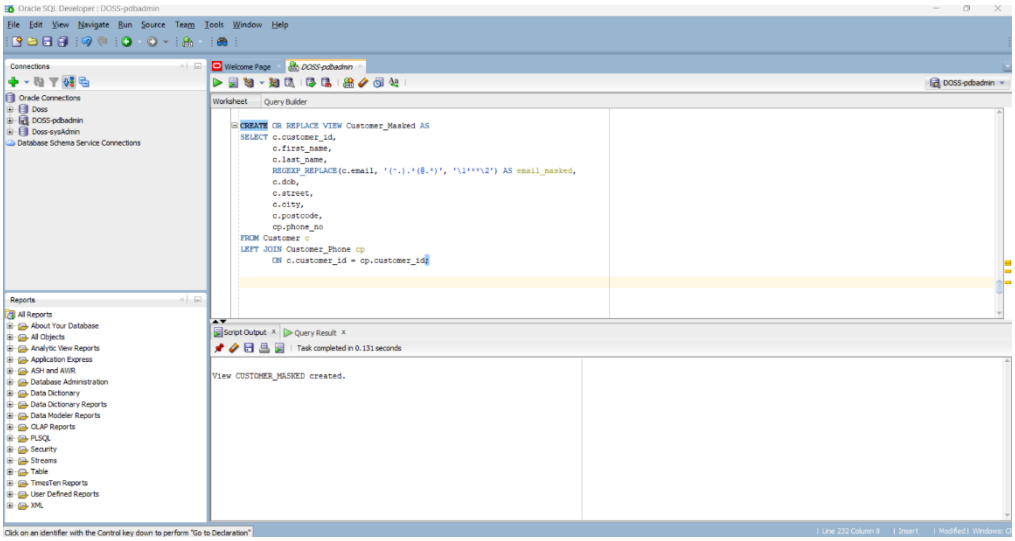


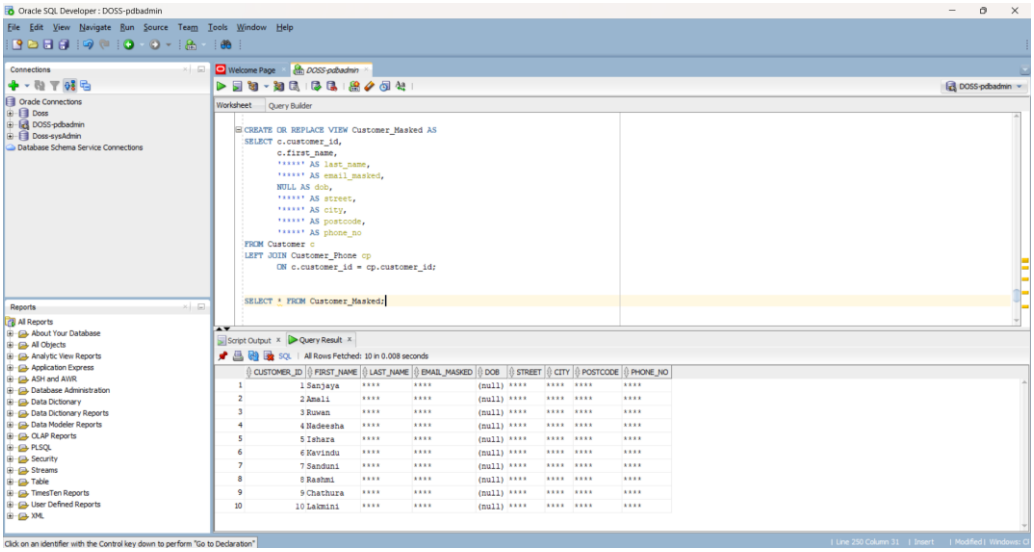
Table with encrypted data

8. DATA Masking

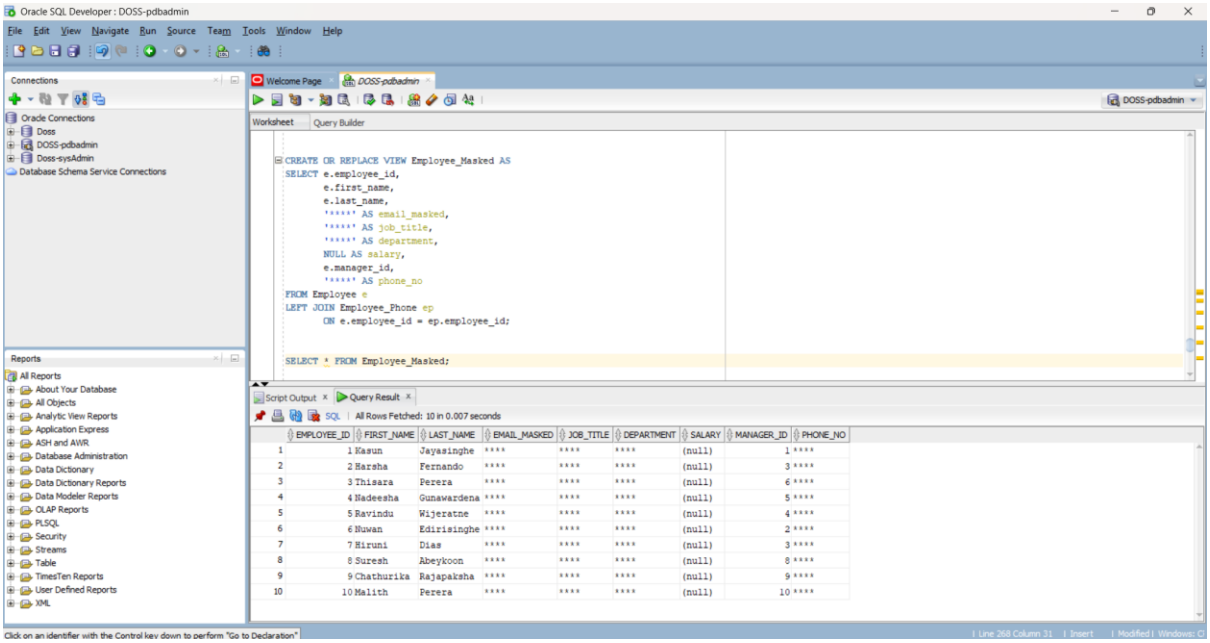
Created a view Customer_Masked that hides sensitive information by masking part of customers' email addresses using the REGEXP_REPLACE function, ensuring privacy while allowing limited data visibility.



Created a view Customer_Masked that completely hides all sensitive customer details

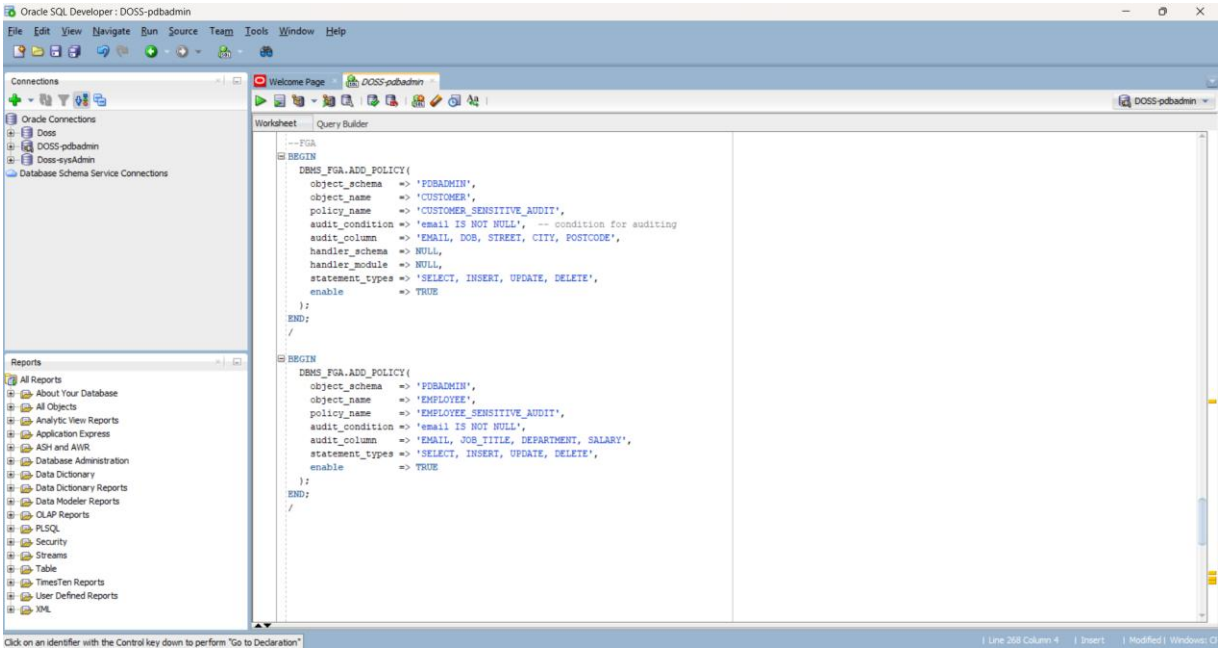


Created a view Employee_Masked that conceals sensitive employee information such as email, job title, department, salary, and phone number



9. FGA

Create policies on a table to audit specific columns or conditions.



10. Big Data Security

Introduction

Advancements in Big Data have revolutionized data collection and analysis but also introduced new security and privacy challenges. This review outlines major Big Data security issues, attack vectors, and mitigation measures based on frameworks like NIST and OWASP [1][2].

Security Requirements of Big Data vs. Traditional Databases

2.1 Main Security Requirements for Big Data

Big Data systems have operational characteristics that are significantly differentiated from traditional data systems, such as a larger scale, higher velocity, and more data diversity, which lead to implications for security, including:

- **End-to-End Protection for Data:**

Big Data must be protected both at rest and in transit. Encryption and centralized key management are essential to prevent data leakage during transfer. [1]

- **Strong Identity and Node Authentication:**

All nodes and users in Big Data environments should be authenticated using cryptographic protocols like mutual TLS (mTLS) to ensure trusted communication. [1]

- **Fine-Grained Access Control:**

Big Data systems should use dynamic, attribute-based (ABAC) or role-based (RBAC) access control to manage complex interactions between users and resources. [2]

- **Real-Time Monitoring and Anomaly Detection:**

Big Data's high-velocity data streaming necessitates real-time intrusion detection and behavioral analytics rather than relying on periodic auditing. [1]

- **Data Governance and Privacy Compliance:**

Big Data systems require governance, data lineage tracking, and techniques like anonymization and masking to protect PII and ensure privacy. [2]

2.2 Key Differences Between Big Data and Traditional Databases

Aspect	Big Data Systems	Traditional Databases	Reference
Scale & Distribution	Highly distributed environments that span numerous nodes, clouds, and data centers; involves secure communication across nodes.	Consolidated architecture with slightly more one role and easier boundary management.	[1]
Velocity & Processing	High-velocity data streaming in real time; requires low-latency security controls.	Occasional or bulk updates with static data processes	[1]
Variety (Schema Heterogeneity)	Handles structured, semi-structured, and unstructured data; difficult to classify and protect.	Primarily structured data with consistent schema and access permission.	[3]
Privacy & Provenance	Requires anonymization, lineage, and provenance tracking of personal/sensitive data	Minimal PII protection with simple access policies.	[1]
Toolchain Complexity	Involves diverse technologies (Hadoop, Spark, Kafka, etc.) with supply-chain vulnerabilities.	Built on a single DBMS with limited components.	[1]

Common Attacks on Big Data Systems

1) Data Breach / Data Exfiltration

This attack targets sensitive data such as personal, intellectual, or financial information. It often exploits compromised credentials, insecure storage, or vulnerable APIs. Big Data

systems are particularly vulnerable, as a single flaw or stolen credential can expose large volumes of diverse data. [1][2]

2) Node Impersonation / Compromised Cluster Nodes

This attack occurs when an attacker takes control of or impersonates a node in a distributed Big Data cluster, exploiting vulnerabilities, stolen credentials, or supply chain attacks. It can lead to data theft, alteration, or malicious job injection, compromising the confidentiality and integrity of the entire system. [1]

3) Distributed Denial of Service (DDoS) on Ingestion / Query Layers

This attack overwhelms Big Data systems by flooding ingestion endpoints or analytics engines with excessive or malformed traffic, exhausting resources like CPU, memory, or storage. The distributed architecture and multiple public-facing endpoints make these systems particularly vulnerable, leading to service degradation or pipeline failures. [4]

4) Inference Attacks / Privacy Leakage from Analytics

Inference attacks derive sensitive information from aggregated data or models without accessing raw records, using repeated queries or correlations. Big Data systems are especially at risk due to high-dimensional datasets, extensive data sharing, and collaborative analytics. [1][2]

5) Insider Threat / Misuse of Privileged Access

An insider threat occurs when an authorized user misuses their access to obtain, alter, or disclose sensitive data. Big Data systems are particularly vulnerable because their distributed, large-scale datasets make it easier for a single insider action to expose significant information, and detection is challenging due to complex logging. [1][5]

Mitigations for Common Big Data Security Attacks

1. Data Breach / Exfiltration

Mitigation 1 – Strong Encryption & Key Management:

Use strong encryption to secure data in transit and at rest. When protecting data, employ modern algorithms, such as AES-256 and TLS 1.3. Key Management System (KMS) must be implemented and centralized protocol key rotation, as well as audit logging, access control for keys must be set in place to prevent unauthorized decryption of sensitive information. [1]

Mitigation 2 – Principle of least Privilege & Fine-Grained Access Controls:

Role-based access control (RBAC) or attribute-based access control (ABAC) models should be implemented to ensure users and/or applications access only the data subset they justifiably need. In tandem, data masking and tokenization should be employed for non-privileged access that deems sensitive fields of information. [2]

2. Node Impersonation / Compromised Nodes

Mitigation 1 – Mutual Authentication & mTLS Between Nodes:

Mutual TLS (mTLS) will safeguard communication between cluster nodes. Certificates must be provided by a trusted Certificate Authority (CA) and rotated as needed to limit impersonation attacks. [1]

Mitigation 2 – Node Attestation & Runtime Integrity Checks:

Node integrity can be assured at startup through Trusted Platform Modules (TPMs) and Secure Boot. Done so, continuous runtime integrity check mechanisms can be developed along with a Software Bill of Materials (SBOM) to monitor for unauthorized changes, either binary or library. [1]

3. Distributed Denial of Service (DDoS) on Ingestion / Query Layers

Mitigation 1 – Rate Limiting & API Gateways:

Utilize API gateways that include rate limits, input validation, and authentication in ord manage data ingestion rates and prevent resource starvation in streaming or query interfaces.[4]

Mitigation 2 – Autoscaling + Anomaly Detection & Blackholing:

Deploy autoscaling mechanisms (to manage legitimate traffic) with real-time anomaly detection for malicious traffic and some form of traffic scrubbing or blackholing/forwarding at the network edge to eliminate DDoS. [4]

4. Inference Attacks / Privacy Leakage from Analytics

Mitigation 1 – Differential Privacy & Query Budgets:

Use differential privacy techniques in analytics pipelines so that statistical results do not inadvertently disclose particular individual record and configure query budgets and limit precision in response in order to compromise the risk of measure of re-identifying individuals. [1]

Mitigation 2 – Output Controls & Synthetic Data Generation:

Create output restrictions of precision for small data cohorts and provide synthetic or anonymized datasets for external sharing or modeling. This reduces the risk of leaking sensitive information in the analytics result process. [2]

5. Insider Threat / Abuse of Privileged Access

Mitigation 1 - Comprehensive auditing & behavior analytics:

Enable immutable audit logging of all access and modification events across the big data ecosystem. Leverage User and Entity Behavior Analytics (UEBA) to identify deviations from expected normal activity that suggest insider misuse. [1]

Mitigation 2 - Just-in-time privilege & separation of duties:

Implement just-in-time (JIT) access mechanisms for short durations of elevated privileges to limit the duration of persistent admin access. Institute separation of duties between admins, analysts, and auditors to mitigate the risk of abuse of privileges. [2]

References

[1] National Institute of Standards and Technology (NIST), *NIST Big Data Interoperability Framework — Volume 4: Security and Privacy*, NIST SP 1500-4r2, 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-4r2.pdf

[2] OWASP Foundation, *OWASP Data Security Top 10*, 2023. [Online]. Available: https://owasp.org/www-project-data-security-top-10/

[3] GeeksforGeeks, "Difference between Traditional Data and Big Data," 2021. [Online]. Available: https://www.geeksforgeeks.org/dbms/difference-between-traditional-data-and-big-data/

[4] CrowdStrike, "Common Types of Cyber Attacks," 2024. [Online]. Available: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/common-cyberattacks/

[5] SecureWorld, "Big Data Security and Privacy Best Practices," 2022. [Online]. Available: https://www.secureworld.io/industry-news/big-data-security-and-privacy-best-practices