

Tạo tài khoản admin để tạo CSDL QLBH và thực hiện các thao tác theo yêu cầu (không sử dụng tài khoản sys để thực hiện vì sys không thể tạo được trigger và cài các chính sách trên VPD)

B1. TẠO TÀI KHOẢN COMMON C##ADMIN (PASS : 123) TRONG CDB\$ROOT => TÀI KHOẢN COMMON LUÔN BẮT ĐẦU BẰNG C## VÀ TẠO TRONG CDB\$ROOT

CREATE USER C##ADMIN IDENTIFIED BY 123;

B2. CẤP QUYỀN SYSDBA CHO C##ADMIN

-- ADD ROLE MEMBER CHO ADMIN LÀ DBA

GRANT DBA TO C##ADMIN;

GRANT EXECUTE ANY PROCEDURE TO C##ADMIN;

--CẤP QUYỀN TRÊN TOÀN BỘ CONTAINER

GRANT CREATE SESSION TO C##ADMIN CONTAINER = ALL;

--CONNECT VÀO USER C##ADMIN

New / Select Database Connection

Connection Name	Connection Details
CADMIN	C##ADMIN@//o...
conn	G004@//localhos...
QLDETAI	G001@//localhos...
QLDETAI1	hongtg@//localh...
QLDT	sys@//localhost:...
QLSV	sys@//localhost:...

Name: CADMIN

Database Type: Oracle

User Info: Proxy User

Authentication Type: Default

Username: C##ADMIN

Role: default

Save Password: ☐

Connection Type: Basic

Details: Advanced

Hostname: localhost

Port: 1521

☒ SID: ORCL

☐ Service name

Status:

Help Save Clear Test Connect Cancel

--XEM QUYỀN CỦA CURRENT USER

```
SELECT * FROM USER_SYS_PRIVS;
```

```
SELECT * FROM USER_TAB_PRIVS;
```

```
SELECT * FROM USER_ROLE_PRIVS;
```

```
--TẠO CSDL QLBH TRONG ORACLE
```

```
CHẠY CRIPT TẠO CSDL QLBH
```

```
B3. CẤP QUYỀN SYSDBA CHO C##ADMIN
```

CÀI ĐẶT CHÍNH SÁCH 1 TRONG BÀI TẬP: CS#1 - Người dùng thuộc loại “Khách hàng cá nhân” có quyền truy cập dữ liệu:

1. Xem thông tin cá nhân của mình, chỉnh sửa thông tin cá nhân (HoTen, Phai, NGSinh, DiaChi, DienThoai)

SỬ DỤNG VIEW: ĐỂ LỌC DÒNG THÔNG TIN CỦA KHÁCH HÀNG THUỘC LOẠI “CÁ NHÂN”

```
--TẠO VIEW XEM THÔNG TIN CÁ NHÂN
```

```
CREATE OR REPLACE VIEW UV_XEMTHONGTINCANHAN  
AS
```

```
SELECT HOTEN, PHAI,NGSINH,DIACHI,DIENTHOAI  
FROM BH_KHACHHANG  
WHERE MAKH = SYS_CONTEXT('USERENV','USERCONTEXT')  
AND LOAIKH = 'CÁ NHÂN';
```

```
--VIẾT THỦ TỤC ĐỂ TẠO USER TRÊN ORACLE
```

```
CREATE OR REPLACE PROCEDURE USP_CREATEUSER  
AS
```

```
CURSOR CUR IS (SELECT MAKH  
FROM BH_KHACHHANG  
WHERE MAKH NOT IN (SELECT USERNAME  
FROM ALL_USERS)
```

```
);
```

```
STRSQL VARCHAR(2000);
```

```
USR VARCHAR2(5);
```

```
BEGIN
```

```
OPEN CUR;
```

```
STRSQL := 'ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE';
```

```
EXECUTE IMMEDIATE(STRSQL);
```

```
LOOP
```

```
FETCH CUR INTO USR;
```

```
EXIT WHEN CUR%NOTFOUND;
```

```
STRSQL := 'CREATE USER '||USR||' IDENTIFIED BY '||USR;
```

```

EXECUTE IMMEDIATE(STRSQL);
STRSQL := 'GRANT CONNECT TO '||USR;
EXECUTE IMMEDIATE(STRSQL);
END LOOP;
STRSQL := 'ALTER SESSION SET "_ORACLE_SCRIPT" = FALSE';
EXECUTE IMMEDIATE(STRSQL);
CLOSE CUR;
END;
--TẠO ROLE RL_KHACHHANGCN

ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE;
CREATE ROLE RL_KHACHHANGCN;
--XEM ROLE ĐÃ TẠO

SELECT * FROM DBA_ROLES WHERE ROLE LIKE '%RL_%' ;
--XEM QUYỀN TRÊN TABLE CỦA ROLE

SELECT * FROM ROLE_TAB_PRIVS WHERE ROLE LIKE '%RL_%';
--XEM DANH SÁCH MEMBER CỦA ROLE

SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTED_ROLE LIKE '%RL_%';
--THÊM USER VÀO ROLE

CREATE OR REPLACE PROCEDURE USP_ADDUSRMEM
(STRROLE VARCHAR, LOAI VARCHAR)
AS
CURSOR CUR IS (SELECT MAKH
                FROM BH_KHACHHANG
                WHERE MAKH IN (SELECT USERNAME
                              FROM ALL_USERS)
                AND LOAIKH = LOAI);
STRSQL VARCHAR(2000);
USR VARCHAR2(5);
BEGIN
OPEN CUR;
LOOP
    FETCH CUR INTO USR;
    EXIT WHEN CUR%NOTFOUND;

    STRSQL := 'GRANT '||STRROLE||' TO '||USR;
    EXECUTE IMMEDIATE (STRSQL);
END LOOP;
CLOSE CUR;
END;
--THỰC THI THỦ TỤC

EXEC USP_ADDUSRMEM ('C##RL_KHACHHANG','CÁ NHÂN');

```

--TEST USER: SỬ DỤNG USER KH1

→ CONNECT VÀO KH1

→ XEM THÔNG TIN CỦA MÌNH (PHẢI GÕ TÊN SCHEMA = TÊN USER CHỨA CÁC BẢNG CỦA CSDL QL BH)

SELECT * FROM C##ADMIN.UV_XEMTHONGTINCANHAN;

→ CHỈNH SỬA THÔNG TIN CỦA MÌNH

UPDATE C##ADMIN.UV_XEMTHONGTINCANHAN

SET PHAI = 'NỮ'

2. Xem thông tin SANPHAM, CUAHANG => RBAC

-- CẤP QUYỀN SELECT CHO ROLE C##RL_KHACHHANG'

GRANT SELECT ON BH_SANPHAM TO C##RL_KHACHHANG;

GRANT SELECT ON BH_CUAHANG TO C##RL_KHACHHANG;

--TEST USER: SỬ DỤNG USER KH1

→ XEM THÔNG TIN SANPHAM, CUAHANG BẰNG USER KH1

SELECT * FROM C##ADMIN.BH_SANPHAM;

SELECT * FROM C##ADMIN.BH_CUAHANG;

3. Xem thông tin DONHANG, CT_DONHANG của mình, chỉnh sửa chi tiết đơn hàng (đổi số lượng) nếu đơn hàng chưa được thanh toán. => VPD

--B1: TẠO FUNCTION POLICY

CREATE OR REPLACE FUNCTION PC1_FUNCTION

(P_SCHEMA VARCHAR2, P_OBJ VARCHAR2)

RETURN VARCHAR2

AS

USER VARCHAR(100);

BEGIN

USER:= SYS_CONTEXT('USEREVN','SESSION_USER');

RETURN 'MAKH = ''' || USER || ''';

END;

--B2: GẮN POLICY VỚI ĐỐI TƯỢNG CẦN KIỂM TRA

BEGIN

DBMS_RLS.ADD_POLICY(

OBJECT_SCHEMA => 'C##ADMIN',

```

    OBJECT_NAME=>'BH_DONHANG',
    POLICY_NAME =>'PC1',
    POLICY_FUNCTION=>'PC1_FUNCTION',
    STATEMENT_TYPES=>'SELECT'
);
END;
--CẤP QUYỀN SELECT CHO ROLE KHACHHANG

```

```

GRANT SELECT ON BH_DONHANG TO C##RL_KHACHHANG;
--TEST USER: SỬ DỤNG USER KH1

```

```

SELECT * FROM C##ADMIN.BH_DONHANG;

```

```

--B1: TẠO FUNCTION POLICY PC2: CHO PHÉP XEM, CHỈNH SỬA CT_DONHANG CHƯA
THANH TOÁN CỦA MÌNH

```

```

CREATE OR REPLACE FUNCTION PC2_FUNCTION
(P_SCHEMA VARCHAR2, P_OBJ VARCHAR2)
RETURN VARCHAR2
AS
    MA VARCHAR2(5);
    STRSQL VARCHAR2(2000);
    CURSOR CUR IS (SELECT MADH
                    FROM BH_DONHANG
                    WHERE MAKH = SYS_CONTEXT('USERENV','SESSION_USER')
                    AND THANHTOAN = 'CHƯA THANH TOÁN');
BEGIN
    OPEN CUR;
    LOOP
        FETCH CUR INTO MA;
        EXIT WHEN CUR%NOTFOUND;

        IF (STRSQL IS NOT NULL) THEN
            STRSQL := STRSQL || ',';
        END IF;
        STRSQL := STRSQL || MA;
    END LOOP;
    RETURN 'MADH IN ('||STRSQL||')';
END;

```

```

--B2: GẮN POLICY VỚI ĐỐI TƯỢNG CẦN KIỂM TRA

```

```

BEGIN
    DBMS_RLS.ADD_POLICY(
        OBJECT_SCHEMA =>'C##ADMIN',
        OBJECT_NAME=>'BH_CT_DONHANG',
        POLICY_NAME =>'PC2',
        POLICY_FUNCTION=>'PC2_FUNCTION',
        STATEMENT_TYPES=>'UPDATE,SELECT',
        SEC_RELEVANT_COLS =>'SOLUONG',
        UPDATE_CHECK => TRUE
    );

```

```

);
END;
--CẤP QUYỀN UPDATE,SELECT CHO ROLE KHACHHANG

GRANT SELECT,UPDATE ON BH_CT_DONHANG TO C##RL_KHACHHANG;

--TEST USER: SỬ DỤNG USER KH1

SELECT * FROM C##ADMIN.BH_CT_DONHANG;

```

- Được xóa đơn hàng của mình trong khoảng thời gian 2 ngày kể từ ngày đặt với điều kiện đơn hàng chưa được thanh toán. => VPD

```

--B1: TẠO FUNCTION_POLICY => VIẾT FUNCTION_POLICY ĐỂ XÓA ĐƠN HÀNG CỦA MÌNH +
CHƯA THANH TOÁN + CÁCH 2 NGÀY ĐẶT

```

```

CREATE OR REPLACE FUNCTION PC3_FUNCTION
(P_SCHEMA VARCHAR2, P_OBJ VARCHAR2)
RETURN VARCHAR2
AS
MA VARCHAR2(5);
STRSQL VARCHAR2(2000);
CURSOR CUR IS (SELECT MADH
                FROM BH_DONHANG
                WHERE MAKH = SYS_CONTEXT('USERENV','SESSION_USER')
                AND THANHTOAN = 'CHƯA THANH TOÁN'
                AND NGAYDAT + INTERVAL '2' DAY = TRUNC(SYSDATE));
BEGIN
OPEN CUR;
LOOP
    FETCH CUR INTO MA;
    EXIT WHEN CUR%NOTFOUND;

    IF (STRSQL IS NOT NULL) THEN
        STRSQL := STRSQL || ',';
    END IF;
    STRSQL := STRSQL || MA;
END LOOP;
RETURN 'MADH IN ('||STRSQL||')';
END;

```

```

--B2: GẮN POLICY VỚI ĐỐI TƯỢNG CẦN KIỂM TRA

```

```

BEGIN
dbms_ols.add_policy(
    OBJECT_SCHEMA => 'C##ADMIN',
    OBJECT_NAME => 'BH_DONHANG',
    POLICY_NAME => 'PC3',

```

```

    POLICY_FUNCTION=>'PC3_FUNCTION',
    STATEMENT_TYPES=>'DELETE',
    UPDATE_CHECK => TRUE
);
END;
--TRIGGER XOA CT_DONHANG KHI XÓA ĐƠN HÀNG

CREATE OR REPLACE TRIGGER UTR_DEL_DONHANG
BEFORE DELETE
ON BH_DONHANG
FOR EACH ROW
BEGIN
    DELETE FROM BH_CT_DONHANG
    WHERE MADH = :OLD.MADH;
END;
--CẤP QUYỀN DELETE CHO ROLE KHACHHANG

GRANT DELETE ON BH_DONHANG TO C##RL_KHACHHANG;
--TEST USER: SỬ DỤNG USER KH1

DELETE FROM C##ADMIN.BH_DONHANG;

```

Thực hiện OLS cho các thông tin sau

- Bảng dữ liệu: OLS_KHACHHANG

ID	NUMBER(10)
CUST_TYPE	VARCHAR2(10)
FIRST_NAME	VARCHAR2(30)
LAST_NAME	VARCHAR2(30)
REGION	VARCHAR2(5)
CREDIT	NUMBER(10,2)

ID	CUST_TYPE	FIRST_NAME	LAST_NAME	REGION	CREDIT
1	SILVER	HARRY	HILL	NORTH	11000
2	SILVER	VIC	REEVES	NORTH	2000
3	SILVER	BOB	MORTIMER	WEST	500
4	SILVER	PAUL	WHITEHOUSE	SOUTH	1000
5	SILVER	HARRY	ENFIELD	EAST	20000
6	GOLD	JENIFER	LOPEZ	WEST	500
7	GOLD	KYLIE	MINOGUE	NORTH	1000
8	GOLD	MARIA	CAREY	WEST	1000
9	GOLD	DANI	MINOGUE	SOUTH	20000
10	GL0D	WHITNEY	HOUSTON	EAST	500
11	PLATINUM	ROBBIE	WILLIAMS	SOUTH	500
12	PLATINUM	THOM	YORKE	NORTH	2000
13	PLATINUM	GARETH	GATES	WEST	10000
14	PLATINUM	DARIUS	DINESH	EAST	2000

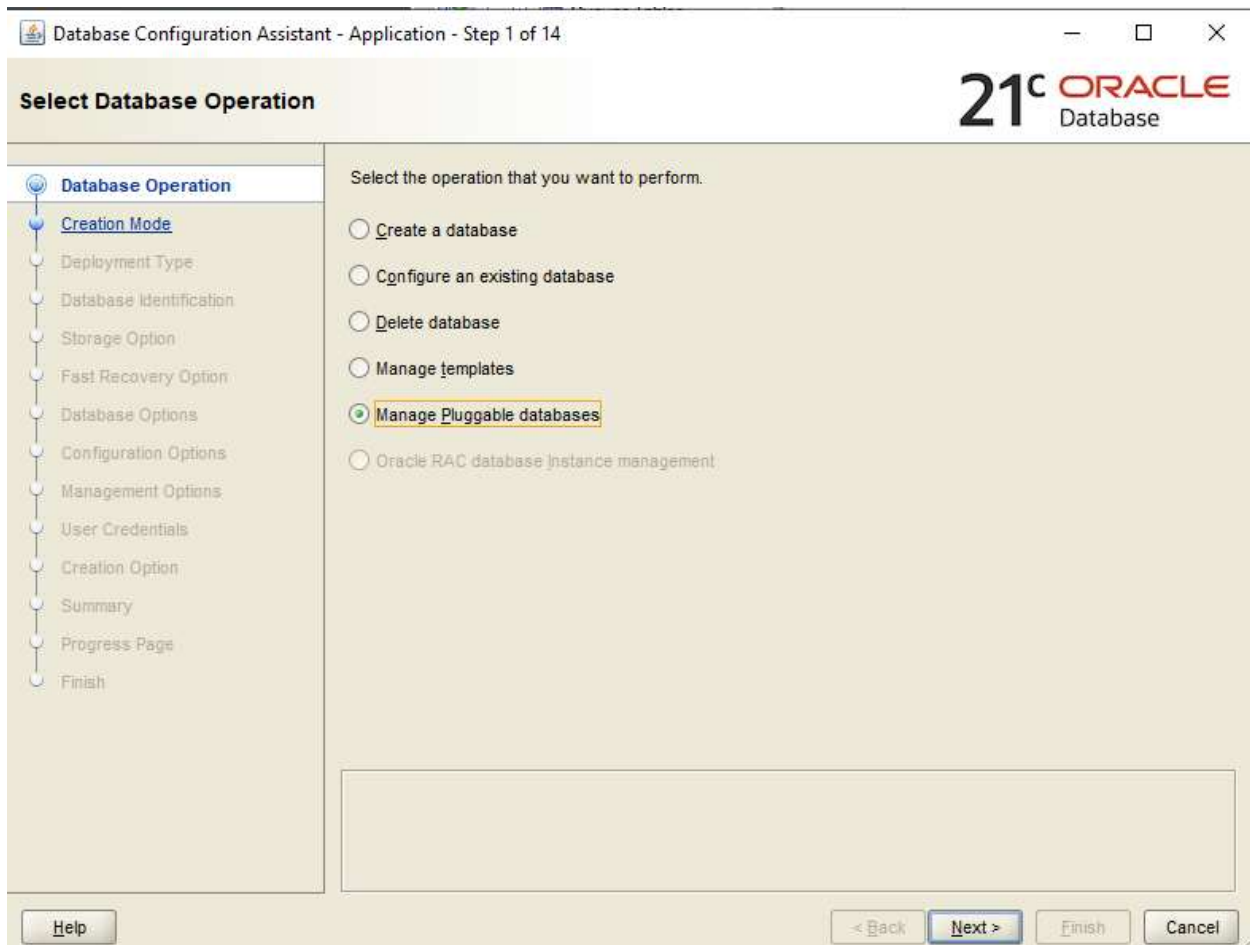
15	PLATINUM	WILL	YOUNG	EAST	100
----	----------	------	-------	------	-----

- NHÃN

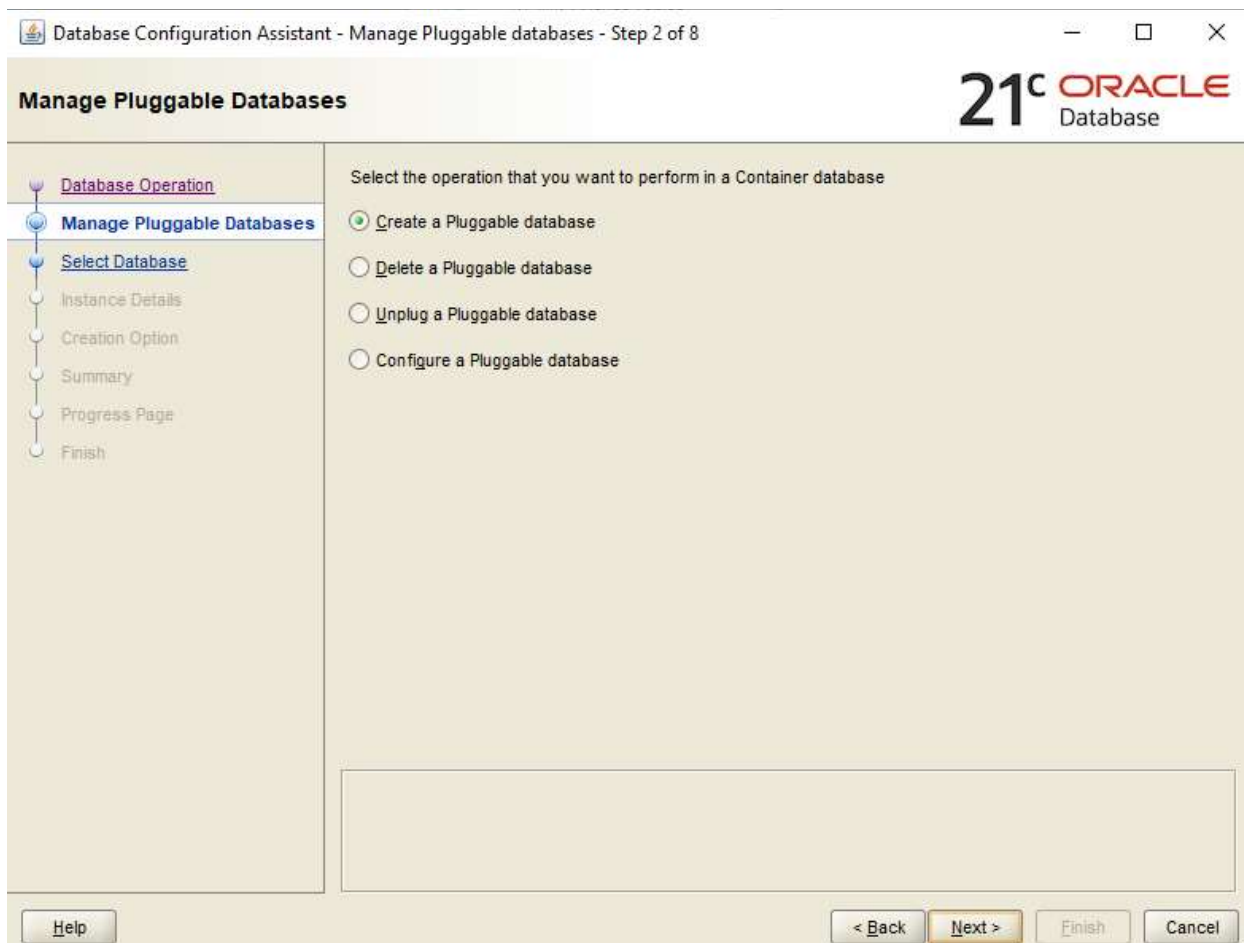
LEVEL		
LEVEL_NUM	SHORT_NAME	LONG_NAME
20	L1	LEVEL 1
40	L2	LEVEL 2
60	L3	LEVEL 3
COMPARTMENT		
COMP_NUM	SHORT_NAME	LONG_NAME
100	M	MANAGEMENT
120	E	EMPLOYEE
GROUP		
GROUP_NUM	SHORT_NAME	LONG_NAME
20	R20	REGION NORTH
40	R40	REGION SOUTH
60	R60	REGION EAST
80	R80	REGION WEST

- Cài đặt

--B1: TẠO PDB BẰNG DATABASE CONFIGURATION ASSISTANT



CHỌN TẠO PDB



CHỌN CDB@ROOT MUỐN TẠO PDB BÊN TRONG

Database Configuration Assistant - Manage Pluggable databases - Step 3 of 8

21^c ORACLE

Database

Select Source Database

Database Operation

Manage Pluggable Databases

Select Database

Instance Details

Creation Option

Summary

Progress Page

Finish

Select a Container database within which Pluggable database needs to be created.

Database	Local instance	Type
<input checked="" type="radio"/> ORCL	ORCL	Single Instance
<input type="radio"/> ORCL1	ORCL1	Single Instance
<input type="radio"/> ORCL200	ORCL200	Single Instance

DBCA will connect to the database using OS based authentication. Database credentials may be needed if OS based authentication is disabled. Specify the credentials, if needed.

User name:

Password:

TDE Wallet Password:

Help

< Back

Next >

Finish

Cancel

CHỌN THIẾT LẬP NHƯ HÌNH

Database Configuration Assistant - Manage Pluggable databases - Step 4 of 9

Create Pluggable Database

21^c ORACLE Database

Database Operation
Manage Pluggable Databases
Select Database
Create Pluggable Database
PDB Identification
Pluggable Database Options
Summary
Progress Page
Finish

☐ Create a new Pluggable database from another PDB

Select Pluggable database: PDB\$SEED

☒ Create Pluggable database from an unplugged PDB

☒ Create as clone

☐ Create from PDB archive

Pluggable database archive: Browse...

☒ Create using PDB file set

Pluggable database metadata file: db_home\assistants\dbca\templates\pdbseed.xml Browse...

Pluggable database datafile backup: db_home\assistants\dbca\templates\pdbseed.dfb Browse...

Help < Back Next > Finish Cancel

ĐẶT TÊN CHO PDB

Database Configuration Assistant - Manage Pluggable databases - Step 5 of 9

Pluggable Database Identification Options

21^c ORACLE Database

Database Operation
Manage Pluggable Databases
Select Database
Create Pluggable Database
PDB Identification
Pluggable Database Options
Summary
Progress Page
Finish

Pluggable database name: PDBQLBH

☐ Create a new administrator

Administrator user name:

Administrator password:

Confirm administrator password:

☐ Lock all existing PDB users

Help < Back Next > Finish Cancel

CHỌN NEXT

Database Configuration Assistant - Manage Pluggable databases - Step 6 of 9

Pluggable Database Options

21^c ORACLE Database

Database Operation
Manage Pluggable Databases
Select Database
Create Pluggable Database
PDB Identification
Pluggable Database Options
Summary
Progress Page
Finish

PDB storage options

Specify the location for PDB datafiles.

Storage type: File System

Database location: C:\ORACLE21C\ORADATA\ORCL\{PDB_NAME} Browse...

☐ Create default user tablespace

Help < Back Next > Finish Cancel

CHỌN FINISH

Summary

- [Database Operation](#)
- [Manage Pluggable Databases](#)
- [Select Database](#)
- [Create Pluggable Database](#)
- [PDB Identification](#)
- [Pluggable Database Options](#)
- [Summary](#)**
- [Progress Page](#)
- [Finish](#)

Database Configuration Assistant**Create Pluggable Database**

- Container Database: ORCL
- Pluggable Database Name: PDBQLBH
- Pluggable Database Source: PDB File Set
- Datafile location: C:\ORACLE21C\ORADATA\ORCL\{PDB_NAME}\
- Configure Oracle Database Vault: No
- Configure Oracle Label Security: No
- Configure Oracle Label Security with OID: No
- Configure Enterprise Manager (EM) database express: No

[Help](#)

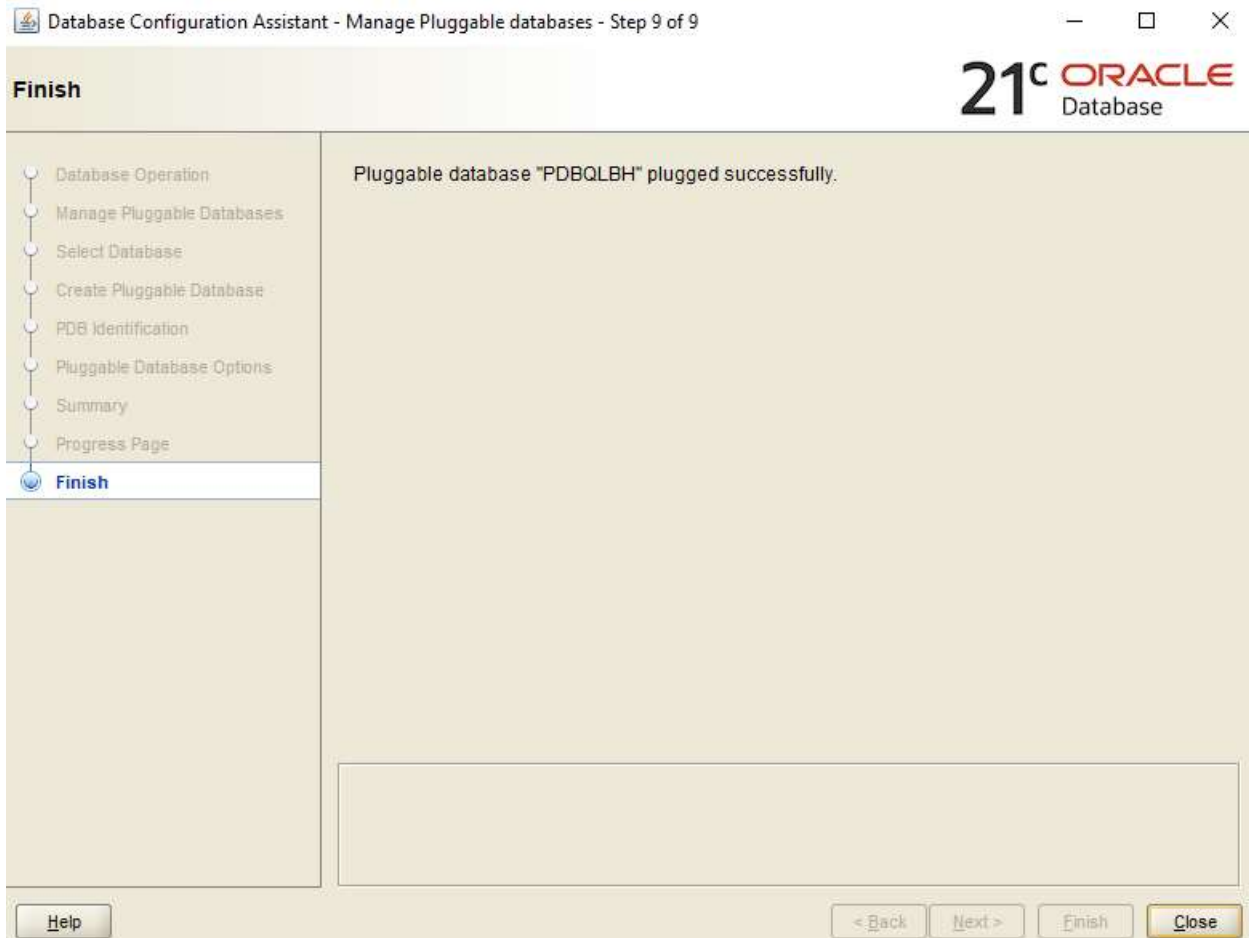
< Back

Next >

Finish

Cancel

HOÀN TẤT TẠO PDB



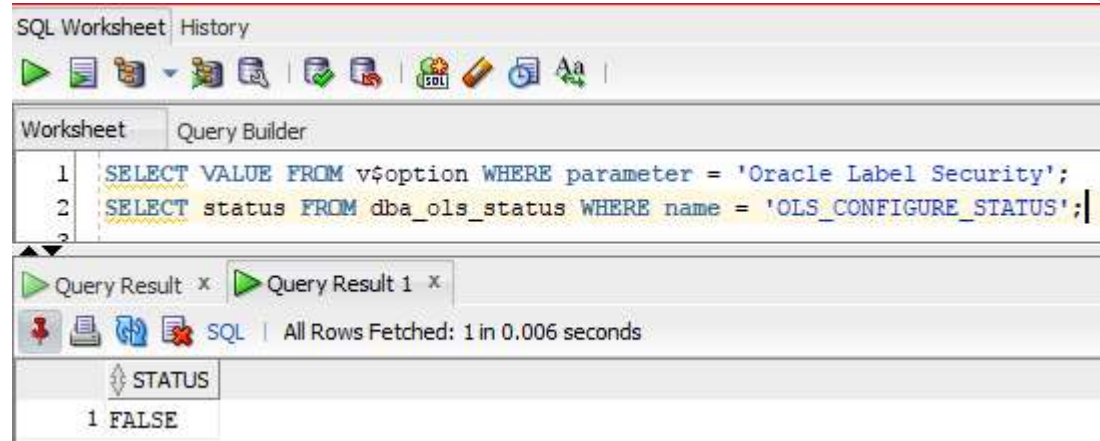
CONNECT VÀO PDB BẰNG SYS

--B2: THIẾT LẬP TRƯỚC KHI TẠO CÁC CHÍNH SÁCH OLS

---> KIỂM TRA OLS ĐÃ ĐƯỢC BẬT CHƯA (CONNECT = SYS USER)

SELECT VALUE FROM v\$option WHERE parameter = 'Oracle Label Security';

SELECT status FROM dba_ols_status WHERE name = 'OLS_CONFIGURE_STATUS';



---> BẬT OLS NẾU CHƯA BẬT/ CHƯA ĐĂNG KÝ THÌ ĐĂNG KÝ OLS VÀ BẬT OLS

EXEC LBACSYS.CONFIGURE_OLS; -- This procedure registers Oracle Label Security.

EXEC LBACSYS.OLS_ENFORCEMENT.ENABLE_OLS; -- This procedure enables it

---> KHỞI ĐỘNG LẠI

SHUTDOWN IMMEDIATE;

STARTUP;

---> KIỂM TRA PDB CÓ CHƯA (VÌ KO THỂ TẠO OLS TREN CDB)

select * from v\$services;

SERVICE_ID	NAME	NAME_HASH	NETWORK_NAME	CREATION_DATE	CREATION_DATE_HASH	GOAL
1	5 orclXDB	3468872077	orclXDB	23-FEB-22	1435368159	NONE
2	10 pdbqlbh	464492181	pdbqlbh	07-MAR-24	3089310120	NONE
3	6 orcl	2392458149	orcl	23-FEB-22	1435368159	NONE
4	1 SYS\$BACKGROUND	165959219	(null)	28-SEP-21	858462688	NONE
5	2 SYS\$USERS	3427055676	(null)	28-SEP-21	858462688	NONE
6	0 orclpdb	3680576210	orclpdb	(null)		0 NONE

---> UNLOCK LBACSYS (OLS ADMIN)

ALTER USER lbacsys IDENTIFIED BY lbacsys ACCOUNT UNLOCK;

---> NẾU CÓ RỒI THÌ MỞ PDB

ALTER PLUGGABLE DATABASE PDBQLBH OPEN READ WRITE;

---> CHUYỂN QUA PDB

ALTER SESSION SET CONTAINER= PDBQLBH;

SHOW CON_NAME;

--TẠO USER TEST OLS = ADMIN_OLS

---> TẠO ADMIN OLS & CẤP QUYỀN CHO ADMIN_OLS

CREATE USER ADMIN_OLS IDENTIFIED BY 123 CONTAINER = CURRENT;

GRANT CONNECT,RESOURCE TO ADMIN_OLS; --CẤP QUYỀN CONNECT VÀ RESOURCE

GRANT UNLIMITED TABLESPACE TO ADMIN_OLS; --CẤP QUOTA CHO ADMIN_OLS

GRANT SELECT ANY DICTIONARY TO ADMIN_OLS; --CẤP QUYỀN ĐỌC DICTIONARY

---> CẤP QUYỀN EXECUTE CHO ADMIN_OLS

GRANT EXECUTE ON LBACSYS.SA_COMPONENTS TO ADMIN_OLS WITH GRANT OPTION;

GRANT EXECUTE ON LBACSYS.sa_user_admin TO ADMIN_OLS WITH GRANT OPTION;

GRANT EXECUTE ON LBACSYS.sa_label_admin TO ADMIN_OLS WITH GRANT OPTION;

GRANT EXECUTE ON sa_policy_admin TO ADMIN_OLS WITH GRANT OPTION;

GRANT EXECUTE ON char_to_label TO ADMIN_OLS WITH GRANT OPTION;

---> ADD ADMIN_OLS VÀO LBAC_DBA

GRANT LBAC_DBA TO ADMIN_OLS;

GRANT EXECUTE ON sa_sysdba TO ADMIN_OLS;

GRANT EXECUTE ON TO_LBAC_DATA_LABEL TO ADMIN_OLS; -- CẤP QUYỀN THỰC THI

--B3: BẮT ĐẦU TẠO CHÍNH SÁCH OLS (CONNECT VÀO TÀI KHOẢN ADMIN_OLS)

-- TẠO CHÍNH SÁCH OLS (KHỞI ĐỘNG LẠI SQLDEV ĐỂ CẬP NHẬT OLS ENABLE)

BEGIN

SA_SYSDBA.CREATE_POLICY(

policy_name => 'region_policy',

column_name => 'region_label'

```
);
END;
--ENABLE POLICY VỪA TẠO -> KHOI DONG LẠI SQLDEV
EXEC SA_SYSDBA.ENABLE_POLICY ('region_policy');
--TẠO COMPONENT CỦA LABEL
--->TẠO LEVEL
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',20,'L1','LEVEL 1');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',40,'L2','LEVEL 2');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',60,'L3','LEVEL 3');
--->TẠO COMPARTMENT
EXECUTE
SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',100,'M','MANAGEMENT');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',120,'E','EMPLOYEE');
```

--->TẠO GROUP

```
EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',20,'R20','REGION NORTH');
EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',40,'R40','REGION SOUTH');
EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',60,'R60','REGION EAST');
EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',80,'R80','REGION WEST');
```

--B4: CONNECT SYS KIỂM TRA COMPONENT TẠO ĐÚNG CHƯA

```
SELECT * FROM DBA_SA_LEVELS;
```

	POLICY_NAME	LEVEL_NUM	SHORT_NAME	LONG_NAME
1	REGION_POLICY	20	L1	LEVEL 1
2	REGION_POLICY	40	L2	LEVEL 2
3	REGION_POLICY	60	L3	LEVEL 3

```
SELECT * FROM DBA_SA_COMPARTMENTS;
```

	POLICY_NAME	COMP_NUM	SHORT_NAME	LONG_NAME
1	REGION_POLICY	120	E	EMPLOYEE
2	REGION_POLICY	100	M	MANAGEMENT

```
SELECT * FROM DBA_SA_GROUPS;
```

	POLICY_NAME	GROUP_NUM	SHORT_NAME	LONG_NAME	PARENT_NUM	PARENT_NAME
1	REGION_POLICY	60	R60	REGION EAST	(null)	(null)
2	REGION_POLICY	20	R20	REGION NORTH	(null)	(null)
3	REGION_POLICY	40	R40	REGION SOUTH	(null)	(null)
4	REGION_POLICY	80	R80	REGION WEST	(null)	(null)

```
SELECT * FROM DBA_SA_GROUP_HIERARCHY;
```

	POLICY_NAME	HIERARCHY_LEVEL	GROUP_NAME
1	REGION_POLICY	1	R20 - REGION NORTH
2	REGION_POLICY	1	R40 - REGION SOUTH
3	REGION_POLICY	1	R60 - REGION EAST
4	REGION_POLICY	1	R80 - REGION WEST

--B5: TẠO BẢNG DỮ LIỆU TEST

```
CREATE TABLE OLS_KHACHHANG
```

```

(
  ID NUMBER(10),
  CUST_TYPE VARCHAR2(10),
  FIRST_NAME VARCHAR2(30),
  LAST_NAME VARCHAR2(30),
  REGION VARCHAR2(30),
  CREDIT NUMBER(10,2),

  PRIMARY KEY(ID)
);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(1,'SILVER', 'HARRY','HILL','NORTH',11000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(2,'SILVER','VIC','REEVES','NORTH',2000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(3,'SILVER','BOB','MORTIMER','WEST',500);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(4,'SILVER','PAUL','WHITEHOUSE','SOUTH',1000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(5,'SILVER','HARRY','ENFIELD','EAST',20000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(6,'GOLD','JENIFER','LOPEZ','WEST',500);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(7,'GOLD','KYLIE','MINOGUE','NORTH',1000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(8,'GOLD','MARIA','CAREY','WEST',1000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(9,'GOLD','DANI','MINOGUE','SOUTH',20000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(10,'GLOD','WHITNEY','HOUSTON','EAST',500);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(11,'PLATINUM','ROBBIE','WILLIAMS','SOUTH',500);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(12,'PLATINUM','THOM','YORKE','NORTH',2000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(13,'PLATINUM','GARETH','GATES','WEST',10000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(14,'PLATINUM','DARIUS','DINESH','EAST',2000);
INSERT INTO OLS_KHACHHANG(ID, CUST_TYPE,FIRST_NAME,LAST_NAME,REGION,CREDIT)
VALUES(15,'PLATINUM','WILL','YOUNG','EAST',100);
SELECT * FROM OLS_KHACHHANG;
--B7: CẬP NHẬT NHÃN TRONG BẢNG
BEGIN
  SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
    POLICY_NAME => 'REGION_POLICY',
    SCHEMA_NAME => 'ADMIN_OLS',

```

```

TABLE_NAME => 'OLS_KHACHHANG',
TABLE_OPTIONS => 'NO_CONTROL'
);
END;
--B6:TAO NHAN
---TAO THU CONG BANG LNH
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L3:E:R20')
WHERE ID = 1;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L2:E:R20')
WHERE ID = 2;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L1:E:R80')
WHERE ID = 3;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L2:E:R40')
WHERE ID = 4;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L3:E:R60')
WHERE ID = 5;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L1:E:R80')
WHERE ID = 6;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L2:E:R20')
WHERE ID = 7;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L2:E:R80')
WHERE ID = 8;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L5:E:R40')
WHERE ID = 9;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L1:E:R60')
WHERE ID = 10;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L1:M:R40')
WHERE ID = 11;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L2:M:R20')
WHERE ID = 12;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L3:M:R80')
WHERE ID = 13;
UPDATE OLS_KHACHHANG

```

```

SET region_label = CHAR_TO_LABEL('REGION_POLICY','L2:M:R60')
WHERE ID = 14;
UPDATE OLS_KHACHHANG
SET region_label = CHAR_TO_LABEL('REGION_POLICY','L1:M:R60')
WHERE ID = 15;

```

ID	CUST_TYPE	FIRST_NAME	LAST_NAME	REGION	CREDIT	REGION_LABEL
1	SILVER	HARRY	HILL	NORTH	11000	1000000008
2	SILVER	VIC	REEVES	NORTH	2000	1000000009
3	SILVER	BOB	MORTIMER	WEST	500	1000000010
4	SILVER	PAUL	WHITEHOUSE	SOUTH	1000	1000000011
5	SILVER	HARRY	ENFIELD	EAST	20000	1000000004
6	GOLD	JENIFER	LOPEZ	WEST	500	1000000010
7	GOLD	KYLIE	MINOGUE	NORTH	1000	1000000009
8	GOLD	MARIA	CAREY	WEST	1000	1000000012
9	GOLD	DANI	MINOGUE	SOUTH	20000	
10	GLOD	WHITNEY	HOUSTON	EAST	500	1000000013
11	PLATINUM	ROBBIE	WILLIAMS	SOUTH	500	1000000014
12	PLATINUM	THOM	YORKE	NORTH	2000	1000000015
13	PLATINUM	GARETH	GATES	WEST	10000	1000000016
14	PLATINUM	DARIUS	DINESH	EAST	2000	1000000017
15	PLATINUM	WILL	YOUNG	EAST	100	1000000006

15 rows selected.

```

--B8: UPDATE ĐỂ KHỞI TẠO LABEL TRONG BẢNG OLS_KHACHHANG (OPTION=
NO_CONTROL)

```

```

UPDATE OLS_KHACHHANG

```

```

SET region_label = CHAR_TO_LABEL('REGION_POLICY','L1');

```

```

--B9: ÁP DỤNG OLS VÀO BẢNG

```

```

BEGIN

```

```

SA_POLICY_ADMIN.REMOVE_TABLE_POLICY('REGION_POLICY','ADMIN_OLS','OLS_KHACHH
ANG');

```

```

SA_POLICY_ADMIN.APPLY_TABLE_POLICY (

```

```

    policy_name => 'REGION_POLICY',

```

```

    schema_name => 'ADMIN_OLS',

```

```

    table_name => 'OLS_KHACHHANG',

```

```

    table_options => 'READ_CONTROL',

```

```

    predicate => NULL

```

```

);

```

```

END;

```

```

--B10:UPDATE BẢNG OLS_KHACHHANG (ĐỂ GỌI UPDATE LABEL)

```

```

UPDATE customers

```

```

SET first_name = first_name;

```

```

COMMIT;

```

```

--B11: TẠO USER ĐỂ TEST (CONNECT VÀO SYS)

```

```

CREATE USER sales_manager IDENTIFIED BY 123;

```

```

CREATE USER sales_north IDENTIFIED BY 123;

```

```

CREATE USER sales_south IDENTIFIED BY 123;

```

```

CREATE USER sales_east IDENTIFIED BY 123;

```

```

CREATE USER sales_west IDENTIFIED BY 123;

```


--B12: CẤP QUYỀN CONNECT CHO CÁC TÀI KHOẢN TEST (CONNECT VÀO SYS)

GRANT CONNECT TO sales_manager, sales_north, sales_south, sales_east, sales_west;

GRANT SELECT ON ADMIN_OLS.OLS_KHACHHANG TO sales_manager, sales_north,

sales_south, sales_east, sales_west;

--B13: GÁN LABEL CHO USER

BEGIN

SA_USER_ADMIN.SET_USER_LABELS('region_policy','sales_manager','L3:M,E:R20,R40,R60,R80');

SA_USER_ADMIN.SET_USER_LABELS('region_policy','sales_north','L3:E:R20,R40');

SA_USER_ADMIN.SET_USER_LABELS('region_policy','sales_south','L3:E:R20,R40,R60,R80');

SA_USER_ADMIN.SET_USER_LABELS('region_policy','sales_east','L3:E:R60');

SA_USER_ADMIN.SET_USER_LABELS('region_policy','sales_west','L3:E:R80');

END;

44 SELECT * FROM ALL_SA_LABELS;






POLICY_NAME	LABEL	LABEL_TAG	LABEL_TYPE
1 REGION_POLICY	L1	1000000000	USER LABEL
2 REGION_POLICY	L3:M,E:R20,R40,R60,R80	1000000001	USER/DATA LABEL
3 REGION_POLICY	L3:E:R20,R40	1000000002	USER/DATA LABEL
4 REGION_POLICY	L3:E:R20,R40,R60,R80	1000000003	USER/DATA LABEL
5 REGION_POLICY	L3:E:R60	1000000004	USER/DATA LABEL
6 REGION_POLICY	L3:E:R80	1000000005	USER/DATA LABEL
7 REGION_POLICY	L1:M:R60	1000000006	USER/DATA LABEL
8 REGION_POLICY	L2:M:R40	1000000007	USER LABEL
9 REGION_POLICY	L3:E:R20	1000000008	USER LABEL
10 REGION_POLICY	L2:E:R20	1000000009	USER LABEL
11 REGION_POLICY	L1:E:R80	1000000010	USER LABEL
12 REGION_POLICY	L2:E:R40	1000000011	USER LABEL
13 REGION_POLICY	L2:E:R80	1000000012	USER LABEL
14 REGION_POLICY	L1:E:R60	1000000013	USER LABEL
15 REGION_POLICY	L1:M:R40	1000000014	USER LABEL
16 REGION_POLICY	L2:M:R20	1000000015	USER LABEL
17 REGION_POLICY	L3:M:R80	1000000016	USER LABEL
18 REGION_POLICY	L2:M:R60	1000000017	USER LABEL

--B14:CONNECT VÀO USER TEST ĐỂ TEST THỬ (UER: SALES_MANAGER)

1

SELECT * FROM ADMIN_OLS.OLS_KHACHHANG. ;

Script Output x

     Task completed in 0.036 seconds

ID	CUST_TYPE	FIRST_NAME	LAST_NAME	REGION	CREDIT	REGION_LABEL
1	SILVER	HARRY	HILL	NORTH	11000	1000000008
2	SILVER	VIC	REEVES	NORTH	2000	1000000009
3	SILVER	BOB	MORTIMER	WEST	500	1000000010
4	SILVER	PAUL	WHITEHOUSE	SOUTH	1000	1000000011
5	SILVER	HARRY	ENFIELD	EAST	20000	1000000004
6	GOLD	JENIFER	LOPEZ	WEST	500	1000000010
7	GOLD	KYLIE	MINOGUE	NORTH	1000	1000000009
8	GOLD	MARIA	CAREY	WEST	1000	1000000012
10	GLD	WHITNEY	HOUSTON	EAST	500	1000000013
11	PLATINUM	ROBBIE	WILLIAMS	SOUTH	500	1000000014
12	PLATINUM	THOM	YORKE	NORTH	2000	1000000015
ID	CUST_TYPE	FIRST_NAME	LAST_NAME	REGION	CREDIT	REGION_LABEL
13	PLATINUM	GARETH	GATES	WEST	10000	1000000016
14	PLATINUM	DARIUS	DINESH	EAST	2000	1000000017
15	PLATINUM	WILL	YOUNG	EAST	100	1000000006

14 rows selected.

14 rows selected.