

### 1. Utworzenie użytkownika:

```
CONNECT SYS/SYS AS SYSDBA;
```

```
CREATE USER ols IDENTIFIED BY ols DEFAULT TABLESPACE users TEMPORARY  
TABLESPACE temp;  
GRANT CONNECT, RESOURCE, SELECT_CATALOG_ROLE TO ols;
```

### 2. Nadanie użytkownikowi odpowiednich praw:

```
CONNECT lbacsys/lbacsys
```

```
GRANT EXECUTE ON sa_components TO ols WITH GRANT OPTION;  
GRANT EXECUTE ON sa_user_admin TO ols WITH GRANT OPTION;  
GRANT EXECUTE ON sa_user_admin TO ols WITH GRANT OPTION;  
GRANT EXECUTE ON sa_label_admin TO ols WITH GRANT OPTION;  
GRANT EXECUTE ON sa_policy_admin TO ols WITH GRANT OPTION;  
GRANT EXECUTE ON sa_audit_admin TO ols WITH GRANT OPTION;
```

```
GRANT LBAC_DBA TO ols;  
GRANT EXECUTE ON sa_sysdba TO ols;  
GRANT EXECUTE ON to_lbac_data_label TO ols;
```

### 3. Utworzenie definicji etykiet:

```
CONNECT ols/ols
```

```
exec SA_SYSDBA.DROP_POLICY('prawo', TRUE);
```

```
BEGIN  
  SA_SYSDBA.CREATE_POLICY(  
    policy_name => 'prawo',  
    column_name => 'etykieta');  
END;  
/
```

```
GRANT prawo_DBA TO ols;
```

### 3. Przypisanie do definicji etykiet poziomów, grup i przegródek:

```
EXECUTE SA_COMPONENTS.CREATE_LEVEL('prawo',20,'P1','Poziom 1');  
EXECUTE SA_COMPONENTS.CREATE_LEVEL('prawo',40,'P2','Poziom 2');  
EXECUTE SA_COMPONENTS.CREATE_LEVEL('prawo',60,'P3','Poziom 3');
```

```
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('prawo',90,'P','PREZES');  
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('prawo',100,'D','DYREKTOR');  
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('prawo',110,'K','KIEROWNIK');  
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('prawo',120,'N','PRACOWNIK');
```

```
EXECUTE SA_COMPONENTS.CREATE_GROUP('prawo',20,'KW','WSCHOD');  
EXECUTE SA_COMPONENTS.CREATE_GROUP('prawo',40,'KZ','ZACHOD');  
EXECUTE SA_COMPONENTS.CREATE_GROUP('prawo',60,'KP','POLNOC');  
EXECUTE SA_COMPONENTS.CREATE_GROUP('prawo',80,'KD','POLUDNIE');  
EXECUTE SA_COMPONENTS.CREATE_GROUP('prawo',10,'K','KRAJ');
```

```
EXECUTE SA_USER_ADMIN.SET_USER_PRIVS('prawo','ols','FULL,PROFILE_ACCESS');
```

#### 4. Stworzenie schematu:

```
CONNECT ols/ols
```

```
CREATE TABLE sprzedawcy (  
  id          NUMBER(10) NOT NULL,  
  rodzaj_klienta  VARCHAR2(10),  
  imie          VARCHAR2(30),  
  nazwisko       VARCHAR2(30),  
  region        VARCHAR2(10),  
  kredyt        NUMBER(10,2),  
  CONSTRAINT klienci_pk PRIMARY KEY (id));
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON sprzedawcy TO PUBLIC;
```

#### 5. Zasilenie schematu danymi:

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 1, 'srebrny', 'Pawel', 'Bik', 'WSCHOD', 11000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 2, 'srebrny', 'Jacek', 'Gigola', 'WSCHOD', 2000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 3, 'srebrny', 'Anna', 'Bajor', 'ZACHOD', 500.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 4, 'srebrny', 'Pawel', 'Bialy', 'POLUDNIE', 1000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 5, 'srebrny', 'Robert', 'Uprzejmy', 'POLNOC', 20000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 6, 'zloty', 'Alicja', 'Lopez', 'ZACHOD', 500.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 7, 'zloty', 'Katarzyna', 'Broszka', 'WSCHOD', 1000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 8, 'zloty', 'Maria', 'Drop', 'ZACHOD', 1000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES ( 9, 'zloty', 'Dani', 'Minogue', 'POLUDNIE', 20000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES (10, 'zloty', 'Anna', 'Jopek', 'POLNOC', 500.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES (11, 'platynowy', 'Adam', 'Ec', 'POLUDNIE', 500.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES (12, 'platynowy', 'Anna', 'Klucha', 'WSCHOD', 2000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES (13, 'platynowy', 'Ewa', 'Prasna', 'ZACHOD', 10000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES (14, 'platynowy', 'Dariusz', 'Bobik', 'POLNOC', 2000.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES (15, 'platynowy', 'Jan', 'Wstretny', 'POLNOC', 100.00);
```

```
INSERT INTO sprzedawcy (id, rodzaj_klienta, imie, nazwisko, region, kredyt)  
VALUES (50, 'diamentowy', 'Iwan', 'Grozny', 'KRAJ', 100000.00);
```

```
COMMIT;
```

#### 6. Utworzenie funkcji zarządzającej etykietami wierszy:

```

CREATE OR REPLACE FUNCTION get_sprzedawca_label (
  p_rodzaj_klienta IN VARCHAR2,
  p_region IN VARCHAR2,
  p_kredyt IN NUMBER)
RETURN LBACSYS.LBAC_LABEL AS
  v_label VARCHAR2(80);
BEGIN
  IF p_kredyt > 10000 THEN
    v_label := 'P3: ';
  ELSIF p_kredyt > 200 THEN
    v_label := 'P2: ';
  ELSIF p_kredyt <= 200 THEN
    v_label := 'P1: ';
  END IF;

  IF p_rodzaj_klienta = 'diamentowy' THEN
    v_label := v_label || 'P: ';
  ELSIF p_rodzaj_klienta = 'platynowy' THEN
    v_label := v_label || 'D: ';
  ELSIF p_rodzaj_klienta = 'zloty' THEN
    v_label := v_label || 'K: ';
  ELSIF p_rodzaj_klienta = 'srebrny' THEN
    v_label := v_label || 'N: ';
  END IF;

  IF p_region = 'WSCHOD' THEN
    v_label := v_label || 'KW';
  ELSIF p_region = 'ZACHOD' THEN
    v_label := v_label || 'KZ';
  ELSIF p_region = 'POLNOC' THEN
    v_label := v_label || 'KP';
  ELSIF p_region = 'POLUDNIE' THEN
    v_label := v_label || 'KD';
  ELSIF p_region = 'KRAJ' THEN
    v_label := v_label || 'K';
  END IF;

  RETURN TO_LBAC_DATA_LABEL('prawo',v_label);
END get_sprzedawca_label;
/

```

SHOW ERRORS

7: Powiązanie obsługi etykiet i funkcji zarządzającej z tabelą:

```
connect ols/ols
```

```
exec SA_POLICY_ADMIN.REMOVE_TABLE_POLICY('prawo','OLS','sprzedawcy');
```

```

BEGIN
  SA_POLICY_ADMIN.REMOVE_TABLE_POLICY('prawo','OLS','sprzedawcy');
  SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
    policy_name => 'prawo',
    schema_name => 'OLS',
    table_name => 'sprzedawcy',
    table_options => 'READ_CONTROL,WRITE_CONTROL,CHECK_CONTROL',

```

```
label_function =>
'ols.get_sprzedawca_label(:new.rodzaj_klienta,:new.region,:new.kredyt)',
predicate => NULL);
END;
/
```

8: Aktualizacja danych:

```
UPDATE sprzedawcy
SET id = id;
```

```
COMMIT;
```

9. Testy:

```
CONNECT sys/sys as sysdba
```

```
CREATE USER a IDENTIFIED BY a;
```

```
GRANT CONNECT TO a;
```

```
CONNECT ols/ols
```

```
exec SA_USER_ADMIN.SET_USER_LABELS('prawo','a','P3:P,N:K');
```

a:

```
connect a/a
```

```
select to_char(id) || '-' || label_to_char(etykieta) from ols.sprzedawcy;
```

b: dodanie węzła grupy 'Kraj' za pomocą Policy Managera

```
connect a/a
```

```
select to_char(id) || '-' || label_to_char(etykieta) from ols.sprzedawcy;
```

c: procedury i funkcje

```
CONNECT ols/ols
```

```
create or replace function suma_kredytow RETURN number IS
suma NUMBER;
```

```
BEGIN
```

```
select sum(kredyt) into suma from sprzedawcy;
```

```
RETURN suma;
```

```
END suma_kredytow;
```

```
/
```

```
grant execute on suma_kredytow to public;
```

```
connect a/a
```

```
select ols.suma_kredytow from dual;
```

```
connect ols/ols
```

```
exec SA_USER_ADMIN.SET_PROG_PRIVS ('prawo','OLS','suma_kredytow','READ');
```

```
connect a/a
```

```
select ols.suma_kredytow from dual;
```