

BÀI 5. ORACLE LABEL SECURITY

1. Thực hành 1

1.1. Cài đặt OLS

⇒ Từ phiên bản 12c tính năng OLS đã được cài đặt sẵn khi cài oracle.

1.2. Cấu hình để sử dụng OLS

⇒ Từ phiên bản 12c tính năng OLS đã được cài đặt sẵn khi cài oracle không cần cấu hình.

1.3. Kích hoạt tài khoản LBACSYS

- Đăng nhập: sys

```
Command Prompt - sqlplus / as sysdba
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Tan>sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 30 20:54:07 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> _
```

- Kiểm tra OLS đã bật hay chưa:

```
SQL> select value from v$option where parameter = 'Oracle Label Security';

VALUE
-----
FALSE

SQL>
```

- Kích hoạt user lbacsys – tài khoản này dùng để triển khai OLS

```
SQL> alter user lbacsys account unlock identified by 123;

User altered.

SQL> _
```

- Kích hoạt OLS

```
SQL> exec lbacsys.configure_ols;

PL/SQL procedure successfully completed.

SQL> exec lbacsys.ols_enforcement.enable_ols;

PL/SQL procedure successfully completed.

SQL>
```

- Khởi động lại Database:

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> STARTUP;
ORACLE instance started.

Total System Global Area 2617241712 bytes
Fixed Size                  9031792 bytes
Variable Size              570425344 bytes
Database Buffers          2030043136 bytes
Redo Buffers                7741440 bytes
Database mounted.
Database opened.
```

- Kiểm tra lại:

```
SQL> select value from v$option where parameter = 'Oracle Label Security';

VALUE
-----
TRUE

SQL>
```

1.4. Chuẩn bị dữ liệu

- Sử dụng các đối tượng dữ liệu thuộc schema HR (có sẵn khi cài đặt Oracle Database, chứa dữ liệu quản lý nhân sự của một công ty - Human Resources) như là những đối tượng dữ liệu cần được bảo vệ. HR là user tạo ra, quản lý và sở hữu (về mặt nội dung) dữ liệu cần được bảo vệ. Đổi mật khẩu cho schema HR: dùng tài khoản SYSDBA để mở khóa và tạo user.

```
100
189 ALTER USER hr IDENTIFIED BY hr ACCOUNT UNLOCK;
190

Script Output x
Task completed in 0.123 seconds

User HR altered.
```

- Tiếp theo, tạo mới user HR_SEC chịu trách nhiệm quản lý những user nào được phép truy xuất dữ liệu trong schema HR:

```
188 GRANT connect, create user, drop user,
189 create role, drop any role
190 TO hr_sec IDENTIFIED BY hrsec;
191

Script Output x
Task completed in 0.047 seconds

Grant succeeded.
```

- Tạo user SEC_ADMIN chịu trách nhiệm quản lý chính sách bảo mật dành cho dữ liệu trong HR.

```
188 GRANT connect TO sec_admin IDENTIFIED BY secadmin;

Script Output x
Task completed in 0.04 seconds

Grant succeeded.
```

- Tạo ra các user là nhân viên trong công ty và role cho các nhân viên:

<pre> 190 CREATE ROLE emp_role; 191 GRANT connect TO emp_role; 192 </pre>	<pre> 193 -- Steven King (Tổng Giám đốc) 194 CREATE USER sking IDENTIFIED BY sking; 195 GRANT emp_role TO sking; 196 </pre>
<p>Script Output x</p> <p>Task completed in 0.041 seconds</p> <p>Role EMP_ROLE created.</p> <p>Grant succeeded.</p>	<p>Script Output x</p> <p>Task completed in 0.032 seconds</p> <p>User SKING created.</p> <p>Grant succeeded.</p>
<pre> 197 -- Neena Kochhar (Giám đốc điều hành) 198 CREATE USER nkochhar IDENTIFIED BY nkochhar; 199 GRANT emp_role TO nkochhar; 200 </pre>	<pre> 202 -- Karen Partner (Trưởng phòng Sales) 203 CREATE USER kpartner IDENTIFIED BY kpartner; 204 GRANT emp_role TO kpartner; 205 </pre>
<p>Script Output x</p> <p>Task completed in 0.047 seconds</p> <p>User NKOCHHAR created.</p> <p>Grant succeeded.</p>	<p>Script Output x</p> <p>Task completed in 0.05 seconds</p> <p>User KPARTNER created.</p> <p>Grant succeeded.</p>
<pre> 207 -- Louise Doran (Nhân viên thuộc phòng Sales) 208 CREATE USER ldoran IDENTIFIED BY ldoran; 209 GRANT emp_role TO ldoran; 210 211 </pre>	
<p>Script Output x</p> <p>Task completed in 0.039 seconds</p> <p>User LDORAN created.</p> <p>Grant succeeded.</p>	

- Vì HR là người quản lý về mặt nội dung đối với dữ liệu trong của phòng nhân sự nên HR là người cấp quyền xem dữ liệu cho các nhân viên:

```

C:\Users\Tan>sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 30 21:32:55 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> CONN hr/123;
Connected.
SQL> GRANT select ON hr.locations TO emp_role;

Grant succeeded.

SQL>

```

2. Thực hành 2: Chính sách trong Oracle Label Security

2.1. Tạo ra một chính sách dùng để điều khiển các truy xuất đến bảng LOCATIONS của HR với tên gọi là “ACCESS_LOCATIONS” và có cột chứa nhãn tên là “OLS_COLUMN”

```
211 -- kết nối user lbacsys để tạo chính sách ACCESS_LOCATIONS
212 BEGIN
213     SA_SYSDBA.CREATE_POLICY (
214         policy_name => 'ACCESS_LOCATIONS',
215         column_name => 'OLS_COLUMN');
216 END;
```

Script Output x
Task completed in 0.469 seconds

PL/SQL procedure successfully completed.

2.2. Để SEC_ADMIN có thể quản lý và duy trì hoạt động của chính sách. Cấp cho user này role quản trị của chính sách và các quyền thực thi trên các package liên quan: dùng user sys

```
218
219 GRANT access_locations_dba TO sec_admin;
220
```

Script Output x
Task completed in 0.042 seconds

Grant succeeded.

```
221 -- Package dùng để tạo ra các thành phần của nhãn
222 GRANT execute ON sa_components TO sec_admin;
223
```

Script Output x
Task completed in 0.05 seconds

Grant succeeded.

```
224 -- Package dùng để tạo các nhãn
225 GRANT execute ON sa_label_admin TO sec_admin;
226
```

Script Output x
Task completed in 0.05 seconds

Grant succeeded.

```
227 -- Package dùng để gán chính sách cho các table/schema
228 GRANT execute ON sa_policy_admin TO sec_admin;
229
```

Script Output x
Task completed in 0.044 seconds

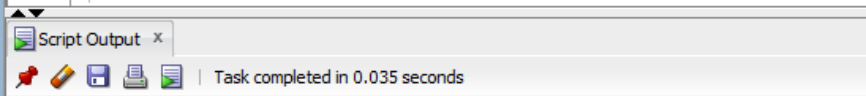
Grant succeeded.

2.3. Để HR_SEC có thể quản lý việc truy xuất của các user, ta cũng cần cấp cho user này role quản trị của chính sách và các quyền thực thi trên các package liên quan:

```

230 GRANT access_locations_dba TO hr_sec;
231

```

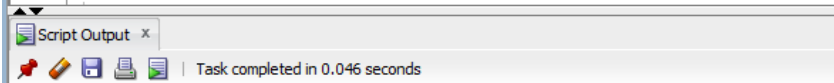


Grant succeeded.

```

232 -- Package dùng để gán các label cho user
233 GRANT execute ON sa_user_admin TO hr_sec;
234

```



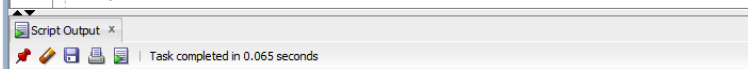
Grant succeeded.

- 2.4. Lưu ý: đối với mỗi user quản lý chính sách, ta cấp cho user đó các quyền thực thi trên các package tương ứng. Tuy nhiên có các quyền trên chưa phải là điều kiện đủ để user đó có thể quản lý các chính sách. Nếu muốn user đó quản lý chính sách nào, ta cần gán thêm role quản trị của chính sách đó cho user.

```

235 -- dùng lbacsys Tạo 1 policy mới nhưng không gán role
236 BEGIN
237     sa_sysdba.create_policy
238     (policy_name => 'Different_Policy');
239 END;

```

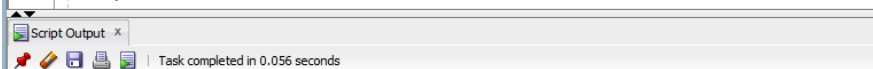


PL/SQL procedure successfully completed.

```

241 /** Thử quản lý policy mới tạo ra. Nhưng sẽ bị thất
242 bại vì sec_admin không được gán role cần thiết.**/
243 -- kết nối user sec_admin/secadmin để thực hiện:
244 BEGIN
245     sa_components.create_level
246     (policy_name => 'Different_Policy',
247      long_name => 'foo',
248      short_name => 'bar',
249      level_num => 9);
250 END;

```



Error report -

```

ORA-12407: unauthorized operation for policy Different_Policy
ORA-06512: at "LBACSYS.SA_COMPONENTS", line 466
ORA-06512: at "LBACSYS.LBAC_STANDARD", line 20
ORA-06512: at "LBACSYS.SA_COMPONENTS", line 69
ORA-06512: at "LBACSYS.SA_COMPONENTS", line 434
ORA-06512: at line 2
12407. 00000 - "unauthorized operation for policy %s"
*Cause:      The policy did not authorize the database session to
              perform the requested operation.
*Action:     Grant the user or program unit the necessary policy
              privilege or additional authorizations.

```

- 2.5. Dùng SA_SYSDBA.DROP_POLICY để xóa chính sách 'Different_Policy' ở trên:

```

253 | -- kết nối user lbacsys/123 để thực hiện:
254 | BEGIN
255 |     sa_sysdba.drop_policy
256 |     (policy_name => 'Different_Policy',
257 |      drop_column => true);
258 | END;
259 | /

```

Script Output x

Task completed in 0.419 seconds

PL/SQL procedure successfully completed.

3. Thực hành 3: Ở phần này ta sẽ tạo các thành phần của nhãn (ứng với bước 2 trong quy trình hiện thực OLS) cho chính sách ACCESS_LOCATIONS mà ta đã tạo trong phần Chính sách trong Oracle Label Security

3.1. Tạo level

- Ta quy định chính sách ACCESS_LOCATIONS của ta có 3 level (theo thứ tự mức độ bảo mật giảm dần): SENSITIVE, CONFIDENTIAL, PUBLIC. Ta dùng thủ tục SA_COMPONENTS.CREATE_LEVEL để tạo ra các level:

```

261 | -- kết nối user sec_admin/secadmin để thực hiện:
262 | BEGIN
263 |     sa_components.create_level
264 |     (policy_name => 'ACCESS_LOCATIONS',
265 |      long_name => 'PUBLIC',
266 |      short_name => 'PUB',
267 |      level_num => 1000);
268 | END;
269 | /

```

Script Output x

Task completed in 0.038 seconds

PL/SQL procedure successfully completed.

```

271 | EXECUTE sa_components.create_level('ACCESS_LOCATIONS',2000,'CONF','CONFIDENTIAL');

```

Script Output x

Task completed in 0.036 seconds

PL/SQL procedure successfully completed.

```

275 | EXECUTE sa_components.create_level('ACCESS_LOCATIONS',3000,'SENS','SENSITIVE');
276 | /

```

Script Output x

Task completed in 0.033 seconds

PL/SQL procedure successfully completed.

Đoạn code trên cho ta thấy 2 cách khác nhau để thực thi thủ tục. Người đọc cũng cần chú ý cách chọn số cho các level.

- Để thay đổi tên đầy đủ và tên rút gọn của level, ta dùng thủ tục SA_COMPONENTS.ALTER_LEVEL. Nếu level đang được dùng bởi ít nhất một nhãn dữ liệu nào đó, ta có thể thay đổi tên đầy đủ của nó nhưng không thể thay đổi tên rút gọn. Trong mọi trường hợp, ta đều không thể thay đổi số đại diện của level.

```

274 | EXECUTE sa_components.create_level('ACCESS_LOCATIONS',4000,'HS','HIGHLY SECRET');
275 | /

```

Script Output x

Task completed in 0.044 seconds

PL/SQL procedure successfully completed.

```

276 -- kết nối user sec_admin/secadmin để thực hiện:
277 BEGIN
278     sa_components.alter_level
279     (policy_name => 'ACCESS_LOCATIONS',
280      level_num => 4000,
281      new_short_name => 'TS',
282      new_long_name => 'TOP SECRET');
283 END;
284 /

```

Script Output x
Task completed in 0.049 seconds

PL/SQL procedure successfully completed.

```

285 BEGIN
286     sa_components.alter_level
287     (policy_name => 'ACCESS_LOCATIONS',
288      short_name => 'TS',
289      new_long_name => 'TOP SENSITIVE');
290 END;
291 /

```

Script Output x
Task completed in 0.038 seconds

PL/SQL procedure successfully completed.

- Để xóa một level ta dùng thủ tục SA_COMPONENTS.DROP_LEVEL. Nếu level đang được sử dụng bởi bất kỳ nhãn dữ liệu nào, ta không thể xóa nó.

```

292 -- kết nối user sec_admin/secadmin để thực hiện:
293 BEGIN
294     sa_components.drop_level
295     (policy_name => 'ACCESS_LOCATIONS',
296      short_name => 'TS');
297 END;

```

Script Output x
Task completed in 0.034 seconds

PL/SQL procedure successfully completed.

3.2. Tạo compartment

- Giả sử chúng ta có 3 compartment là: Finance, Sales & Marketing, Human Resources. Để tạo compartment chúng ta dùng procedure SA_COMPONENTS.CREATE_COMPARTMENT:

```

300 -- kết nối user sec_admin/secadmin để thực hiện:
301 BEGIN
302     sa_components.create_compartment
303     (policy_name => 'ACCESS_LOCATIONS',
304      long_name => 'SALES_MARKETING',
305      short_name => 'SM',
306      comp_num => 2000);
307 END;

```

Script Output x
Task completed in 0.055 seconds

PL/SQL procedure successfully completed.

```

310 EXECUTE sa_components.create_compartment('ACCESS_LOCATIONS',3000,'FIN','FINANCE');
311 EXECUTE sa_components.create_compartment('ACCESS_LOCATIONS',1000,'HR','HUMAN RESOURCES');

```

Script Output x

Task completed in 0.042 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

- Để thay đổi tên đầy đủ và tên rút gọn của compartment, dùng thủ tục SA_COMPONENTS.ALTER_COMPARTMENT. Các điều kiện của việc thay đổi thuộc tính của compartment giống như đối với level.

```

314 -- kết nối user sec_admin/secadmin để thực hiện:
315 EXECUTE sa_components.create_compartment('ACCESS_LOCATIONS',4000, 'PR', 'PUBLIC RELATIONS');

```

Script Output x

Task completed in 0.031 seconds

PL/SQL procedure successfully completed.

```

317 BEGIN
318     sa_components.alter_compartment
319     (policy_name => 'ACCESS_LOCATIONS',
320      comp_num => 4000,
321      new_short_name => 'PU',
322      new_long_name => 'PURCHASING');
323 END;
324 /

```

Script Output x

Task completed in 0.031 seconds

PL/SQL procedure successfully completed.

```

325 BEGIN
326     sa_components.alter_compartment
327     (policy_name => 'ACCESS_LOCATIONS',
328      short_name => 'PU',
329      new_long_name => 'PURCHASE');
330 END;

```

Script Output x

Task completed in 0.044 seconds

PL/SQL procedure successfully completed.

- Để xóa một compartment dùng thủ tục SA_COMPONENTS.DROP_COMPARTMENT. Nếu compartment đang được sử dụng bởi bất kỳ nhãn dữ liệu nào, không thể xóa.

```

332 BEGIN
333     sa_components.drop_compartment
334     (policy_name => 'ACCESS_LOCATIONS',
335      short_name => 'PU');
336 END;
337 /

```

Script Output x

Task completed in 0.044 seconds

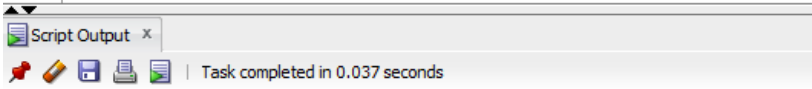
PL/SQL procedure successfully completed.

3.3. Tạo group

Chính sách của chúng ta sẽ có 1 group cấp cao nhất là Corporate (CORP) tương ứng với cấp độ toàn công ty. Công ty này có các chi nhánh hoạt động ở 3 nước: Mỹ (American United States), Anh (United Kingdom) và Canada. Ứng với mỗi khu vực đó ta tạo 1 group con cho group CORP. -- ket noi user sec_admin/secadmin de thuc hien:

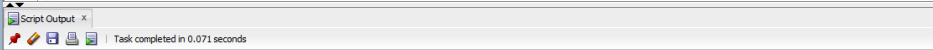
- Dùng procedure SA_COMPONENTS.CREATE_GROUP để tạo ra các group:

```
339 BEGIN
340     sa_components.create_group
341     (policy_name => 'ACCESS_LOCATIONS',
342     long_name => 'CORPORATE',
343     short_name => 'CORP',
344     group_num => 10,
345     parent_name => NULL);
346 END;
347 /
```



PL/SQL procedure successfully completed.

```
348 EXECUTE sa_components.create_group ('ACCESS_LOCATIONS',30,'US','UNITED STATES','CORP');
349 EXECUTE sa_components.create_group ('ACCESS_LOCATIONS',50,'UK','UNITED KINGDOM','CORP');
350 EXECUTE sa_components.create_group ('ACCESS_LOCATIONS',70,'CA','CANADA','CORP');
```



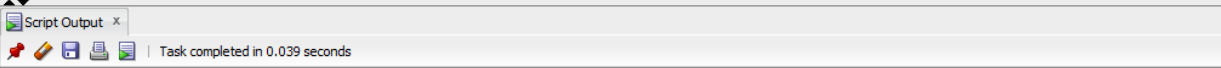
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

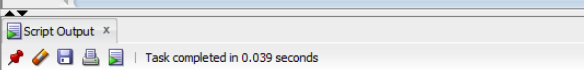
- Để thay đổi tên đầy đủ và tên rút gọn của group, dùng thủ tục SA_COMPONENTS.ALTER_GROUP. Các điều kiện của việc thay đổi thuộc tính của group giống như đối với level.

```
352 EXECUTE sa_components.create_group('ACCESS_LOCATIONS',90,'FR','FRANCE','CORP');
```



PL/SQL procedure successfully completed.

```
354 BEGIN
355     sa_components.alter_group
356     (policy_name => 'ACCESS_LOCATIONS',
357     group_num => 90,
358     new_short_name => 'RFR',
359     new_long_name => 'REPUBLIC FRANCE');
360 END;
```



PL/SQL procedure successfully completed.

```

362 BEGIN
363     sa_components.alter_group
364     (policy_name => 'ACCESS_LOCATIONS',
365      short_name => 'RFR',
366      new_long_name => 'PURCHASE');
367 END;

```

Script Output x

Task completed in 0.039 seconds

PL/SQL procedure successfully completed.

- Để xóa một group dùng thủ tục SA_COMPONENTS.DROP_GROUP. Nếu group đang được sử dụng bởi bất kỳ nhãn dữ liệu nào, ta không thể xóa nó.

```

368 -- kết nối user sec_admin/secadmin để thực hiện:
369 BEGIN
370     sa_components.drop_group
371     (policy_name => 'ACCESS_LOCATIONS',
372      short_name => 'RFR');
373 END;

```

Script Output x

Task completed in 0.033 seconds

PL/SQL procedure successfully completed.

4. Thực hành 4: Chi tiết về nhãn dữ liệu

- Phần này thực hiện bước 3 trong quy trình hiện thực OLS tạo các nhãn thật sự cần dùng từ các thành phần đã tạo ở phần III.
- Để tạo nhãn dùng thủ tục SA_LABEL_ADMIN.CREATE_LABEL. Khi sử dụng thủ tục này để tạo nhãn, ta phải tự định ra label tag (là một số nguyên có tối đa 8 chữ số) cho nhãn được tạo.

```

375 -- kết nối user sec_admin/secadmin để thực hiện:
376 BEGIN
377     sa_label_admin.create_label
378     (policy_name => 'ACCESS_LOCATIONS',
379      label_tag => 10000,
380      label_value => 'PUB');
381 END;

```

Script Output x

Task completed in 0.046 seconds

PL/SQL procedure successfully completed.

```

384 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',20000,'CONF');

```

Script Output x

Task completed in 0.036 seconds

PL/SQL procedure successfully completed.

```
386 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',20010,'CONF::US');
387 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',20020,'CONF::UK');
388 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',20030,'CONF::CA');

Script Output x
Task completed in 0.048 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

390 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',21020,'CONF:HR:UK');
391 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',22040,'CONF:SM:UK,CA');

Script Output x
Task completed in 0.047 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

393 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',34000,'SENS:SM,FIN');
394 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',39090,'SENS:HR,SM,FIN:CORP');

Script Output x
Task completed in 0.048 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```

- Thông thường, khi xây dựng chính sách nên xây dựng hệ thống quy ước đặt label tag để tiện lợi trong việc quản lý. Trong đoạn code trên, quy ước chữ số đầu tiên biểu diễn level (1 là PUB, 2 là CONF, 3 là SENS), 2 chữ số kế tiếp biểu diễn các compartment (00 cho biết không có compartment), 2 chữ số cuối biểu diễn group (00 cho biết không có group).

(Lưu ý là trong phần thực hành này chỉ tạo một số nhãn để minh họa, chứ không tạo hết tất cả các nhãn cần thiết).

- Để thay đổi nhãn, dùng thủ tục SA_LABEL_ADMIN.ALTER_LABEL. Có thể thay đổi giá trị của nhãn nhưng không thể thay đổi giá trị của label tag. Do vậy, độ nhạy cảm của dữ liệu có thể thay đổi được mà không cần phải cập nhật lại bảng chứa dữ liệu đó, do trong bảng chỉ lưu label tag chứ không lưu giá trị của nhãn.

```
397 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',30000,'SENS');
398 EXECUTE sa_label_admin.create_label('ACCESS_LOCATIONS',30090,'SENS::CORP');
```

Script Output x
Task completed in 0.043 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

```
400 BEGIN
401     sa_label_admin.alter_label
402     (policy_name => 'ACCESS_LOCATIONS',
403      label_tag => 30000,
404      new_label_value => 'SENS:SM');
405     sa_label_admin.alter_label
406     (policy_name => 'ACCESS_LOCATIONS',
407      label_value => 'SENS:SM',
408      new_label_value => 'SENS:HR');
409 END;
```

Script Output x
Task completed in 0.04 seconds

PL/SQL procedure successfully completed.

- Ta có thể xóa nhãn bằng thủ tục SA_LABEL_ADMIN.DROP_LABEL:

```
412 BEGIN
413     sa_label_admin.drop_label
414     (policy_name => 'ACCESS_LOCATIONS',
415      label_value => 'SENS:HR');
416 END;
417 /
```

Script Output x
Task completed in 0.045 seconds

PL/SQL procedure successfully completed.

```
418 BEGIN
419     sa_label_admin.drop_label
420     (policy_name => 'ACCESS_LOCATIONS',
421      label_tag => 30090);
422 END;
```

Script Output x
Task completed in 0.03 seconds

PL/SQL procedure successfully completed.

5. Bài tập

5.1. Tạo user ols_test và cấp quyền để user này truy cập vào hệ thống được. Cấp quyền thực thi trên các gói thủ tục cần thiết để user này quản lý được một chính sách.

```
13 -- tạo user ols_test
14 CREATE USER ols_test IDENTIFIED BY 123
15 DEFAULT TABLESPACE users
16 TEMPORARY TABLESPACE temp
17 QUOTA UNLIMITED ON users;
```

Script Output x
Task completed in 0.037 seconds

User OLS_TEST created.

```

274 -- gan quyen de user truy cap vao he thong bang sys
275 GRANT connect TO ols_test;
276 GRANT CREATE TABLE TO ols_test;
277 GRANT select any TABLE TO ols_test;
278 GRANT update any TABLE TO ols_test;
279 GRANT delete any TABLE TO ols_test;
280 GRANT insert any TABLE TO ols test;

```

Script Output x
Task completed in 0.039 seconds

Grant succeeded.

Grant succeeded.

Grant succeeded.

Grant succeeded.

```

282 GRANT execute ON to_lbac_data_label TO ols_test WITH GRANT OPTION;
283 GRANT execute ON SA SYSDBA TO ols test;

```

Script Output x
Task completed in 0.03 seconds

Grant succeeded.

Grant succeeded.

```

285 -- Package dùng để tạo ra các thành phần của nhân
286 GRANT execute ON sa_components TO ols_test;
287 -- Package dùng để tạo các nhân
288 GRANT execute ON sa_label_admin TO ols_test;
289 -- Package dùng để gán chính sách cho các table/schema
290 GRANT execute ON sa_policy_admin TO ols_test;
291 -- Package dùng để gán các label cho user
292 GRANT execute ON sa_user_admin TO ols_test;

```

Script Output x
Task completed in 0.041 seconds

Grant succeeded.

Grant succeeded.

Grant succeeded.

Grant succeeded.

5.2. Tạo chính sách region_policy với tên cột chính sách là region_label. Thực hiện lệnh cần thiết để ols_test trở thành người quản lý chính sách này.

```

310 --5.2 tạo chính sách
311 -- dùng user lbacsys để tạo chính sách
312 BEGIN
313 SA_SYSDBA.CREATE_POLICY (
314 policy_name => 'region_policy',
315 column_name => 'region_label');
316 END;
317 /
318 -- user gán quyền cho user lbacsys
319 GRANT SELECT ON DBA_USERS TO LBACSYS;
320
321 -- thủ tục select tất cả user
322 create or replace procedure pro_select_all_users
323 (v_out out sys_refcursor)
324 is
325 begin
326 open v_out for
327 SELECT username
328 FROM dba_users
329 ORDER BY username ASC;
330 end;
331

```

```

333 -- user lbacsys gan quyền để user ols_test quản lý chính sách này
334 GRANT region_policy_DBA TO ols_test;
335
336 -- thủ tục tạo chính sách OLS
337 create or replace procedure pro_create_policy
338 (policyName in VARCHAR2 ,colName in VARCHAR2)
339 is
340 begin
341     SA_SYSDBA.CREATE_POLICY (
342         policy_name => policyName,
343         column_name => colName);
344 end ;
345 /
-- thủ tục gán quyền đến 1 user quản lý 1 chính sách
create or replace procedure pro_grant_policy
(policyName in VARCHAR2 ,username in VARCHAR2)
is
text varchar2(100);
begin
    text := 'GRANT ' || policyName || '_DBA TO ' || username;
    execute immediate text;
    commit work;
end;

--thủ tục select tất cả policy ols hiện có
create or REPLACE procedure pro_select_OLS_POLICIES
( v_out out SYS_REFCURSOR )
is
begin
open v_out for
    select * from ALL_SA_POLICIES;
end ;

```

- Giao diện bài tập: bắt buộc sử dụng lại giao diện để Đăng nhập trước khi thực hiện:

- Thêm dòng code để hiển thị form khi thực hiện Login thành công

```

1 reference
private void btn_login_Click(object sender, EventArgs e)
{
    string host = txt_host.Text;
    string port = txt_port.Text;
    string sid = txt_sid.Text;
    string user = txt_user.Text;
    string pass = txt_password.Text;

    if (Check_Textbox(host, port, sid, user, pass))
    {
        Database.Set_Database(host, port, sid, user, pass);
        if (Database.Connect())
        {
            OracleConnection c = Database.Get_Connect();
            MessageBox.Show("Đăng nhập thành công\nServerVersion: " + c.ServerVersion);
            // Hiện thị form2
            lab5_baitap5_2 form2 = new lab5_baitap5_2();
            this.Hide(); // Ẩn form1
            form2.Show();
        }
        else
        {
            MessageBox.Show("Đăng nhập thất bại");
        }
    }
}

```

- Giao diện bài tập: chữ màu đỏ: tên Design được định nghĩa khi tạo thành phần

```
public partial class taoPolicy_GanQuyenQL : Form
{
    private OracleConnection conn;

    1 reference
    public taoPolicy_GanQuyenQL()
    {
        InitializeComponent();
        CenterToScreen();
        conn = Database.Get_Connect();
    }

    1 reference
    private void lab5_baitap5_2_Load(object sender, EventArgs e)
    {
        LoadPolicyComboBox(conn);
        load_Cbo_User(conn);
    }
}
```

- Code hiển thị chức năng tạo policy: viết hàm addPolicy để gọi chạy:

```
1 reference
private void addPolicy(OracleConnection conn, string policyName, string columnName)
{
    try
    {
        using (OracleCommand cmd = new OracleCommand("pro_create_policy", conn))
        {
            cmd.CommandType = CommandType.StoredProcedure;

            // Thêm tham số vào stored procedure
            cmd.Parameters.Add("policyName", OracleDbType.Varchar2).Value = policyName;
            cmd.Parameters.Add("colName", OracleDbType.Varchar2).Value = columnName;

            try
            {
                // Thực thi stored procedure
                cmd.ExecuteNonQuery();
                MessageBox.Show("Policy created successfully!");
                LoadPolicyComboBox(conn);
            }
            catch (OracleException ex)
            {
                MessageBox.Show("Error creating policy: " + ex.Message);
            }
        }
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Error creating policy: " + ex.Message);
    }
}
```

- Code hiển thị những policy hiện có: viết hàm loadPolicy với sự kiện loadForm

```

}
References
private void LoadPolicyComboBox(OracleConnection connection)
{
    try
    {
        using (OracleCommand command = new OracleCommand("pro_select_OLS_POLICIES", connection))
        {
            command.CommandType = CommandType.StoredProcedure;

            // Tạo tham số output
            OracleParameter outParam = new OracleParameter("v_out", OracleDbType.RefCursor);
            outParam.Direction = ParameterDirection.Output;
            command.Parameters.Add(outParam);

            // Thực thi thủ tục
            command.ExecuteNonQuery();

            // Lấy dữ liệu từ tham số output
            using (OracleDataReader reader = command.ExecuteReader())
            {
                cboName.Items.Clear();
                while (reader.Read())
                {
                    string policyName = reader.GetString(0);
                    cboName.Items.Add(policyName);
                    cboName.SelectedIndex = 0;
                }
            }
        }
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Error Select policy: " + ex.Message);
    }
}

```

- Code gán quyền quản lí 1 chính sách cho 1 user:


```

1 reference
private void btnGan_Click(object sender, EventArgs e)
{
    // Lấy giá trị từ các control trên form
    object selectedItem = cboName.SelectedItem;
    string selectedName = cboName.SelectedItem.ToString();
    string userName = txtUser.Text;
    runPro_grant_policy(conn, selectedName, userName);
}

1 reference
private void runPro_grant_policy(OracleConnection conn, string policyName, string userName)
{
    try
    {
        using (OracleCommand cmd = new OracleCommand("pro_grant_policy", conn))
        {
            cmd.CommandType = CommandType.StoredProcedure;

            // Thêm tham số vào stored procedure
            cmd.Parameters.Add("policyName", OracleDbType.Varchar2).Value = policyName;
            cmd.Parameters.Add("username", OracleDbType.Varchar2).Value = userName;

            try
            {
                // Thực thi stored procedure
                cmd.ExecuteNonQuery();
                MessageBox.Show("Policy granted successfully!");
                LoadPolicyComboBox(conn);
            }
            catch (OracleException ex)
            {
                MessageBox.Show("Error granted policy: " + ex.Message);
            }
        }
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Error creating policy: " + ex.Message);
    }
}

```

- Code hiển thị danh sách user ở combobox

1 reference

```
private void load_Cbo_User(OracleConnection conn)
{
    try
    {
        using (OracleCommand command = new OracleCommand("pro_select_all_users", conn))
        {
            command.CommandType = CommandType.StoredProcedure;

            // Tạo tham số output
            OracleParameter outParam = new OracleParameter("v_out", OracleDbType.RefCursor);
            outParam.Direction = ParameterDirection.Output;
            command.Parameters.Add(outParam);

            // Thực thi thủ tục
            command.ExecuteNonQuery();

            // Lấy dữ liệu từ tham số output
            using (OracleDataReader reader = command.ExecuteReader())
            {
                cbo_User.Items.Clear();
                while (reader.Read())
                {
                    string userName = reader.GetString(0);
                    cbo_User.Items.Add(userName);
                    cbo_User.SelectedIndex = 0;
                }
            }
        }
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Error Select user: " + ex.Message);
    }
}
```

5.3. Disable thủ tục đã tạo ở câu 2. Sau đó enable nó lại.

- Đây là những thủ tục sẽ được thực thi ở bài C#:

Dùng user lbacsys để tạo các thủ tục, PACKAGE

```
325 -- thu tục DISABLE POLICY
326 create or replace procedure pro_disable_policy
327 (policyName in VARCHAR2 )
328 is
329 BEGIN
330     SA_SYSDBA.DISABLE_POLICY (
331     policy_name => policyName);
332 END;
```

Script Output x

Task completed in 0.029 seconds

Procedure PRO_DISABLE_POLICY compiled

```
334 -- thu tục ENABLE POLICY
335 create or replace procedure pro_enable_policy
336 (policyName in VARCHAR2 )
337 is
338 BEGIN
339     SA_SYSDBA.ENABLE_POLICY
340     ( policy_name => policyName );
341 END;
```

Script Output x

Task completed in 0.042 seconds

Procedure PRO_ENABLE_POLICY compiled

```

390 --thu tuc select tat ca policy ols hien co
391 create or REPLACE procedure pro_select_OLS_POLICIES
392 ( v_out out SYS_REFCURSOR )
393 is
394 begin
395 open v_out for
396 select * from ALL_SA_POLICIES;
397 end ;
398 /

```

```

399 --thu tuc select status 1 policy ols hien co
400 create or REPLACE procedure pro_select_Status_OLS_POLICIES
401 ( policyName in VARCHAR2, v_out out SYS_REFCURSOR )
402 is
403 begin
404 open v_out for
405 select status from ALL_SA_POLICIES where POLICY_NAME = policyName ;
406 end ;
407 /

```

- Hoặc đúng ta sẽ dùng PACKAGE ở bài này. Tương tự như thủ tục:

```

408 -- tao packet cho chuc nang disable va enable policy OLS
409 CREATE OR REPLACE PACKAGE packg_disable_enable_policy IS
410 PROCEDURE pro_disable_policy (policyName IN VARCHAR2);
411 PROCEDURE pro_enable_policy (policyName IN VARCHAR2);
412 PROCEDURE pro_select_OLS_POLICIES (v_out OUT SYS_REFCURSOR);
413 PROCEDURE pro_select_Status_OLS_POLICIES
414 (policyName IN VARCHAR2, v_out OUT SYS_REFCURSOR);
415 END packg_disable_enable_policy;
416 /

```

```

418 CREATE OR REPLACE PACKAGE BODY packg_disable_enable_policy IS
419 -- Thu tuc DISABLE_POLICY
420 PROCEDURE pro_disable_policy(policyName IN VARCHAR2) IS
421 BEGIN
422     SA_SYSDBA.DISABLE_POLICY(
423         policy_name => policyName);
424 END pro_disable_policy;
425
426 -- Thu tuc ENABLE_POLICY
427 PROCEDURE pro_enable_policy(policyName IN VARCHAR2) IS
428 BEGIN
429     SA_SYSDBA.ENABLE_POLICY(
430         policy_name => policyName );
431 END pro_enable_policy;
432
433 -- Thu tuc SELECT_OLS_POLICIES
434 PROCEDURE pro_select_OLS_POLICIES(v_out OUT SYS_REFCURSOR) IS
435 BEGIN
436     OPEN v_out FOR
437         SELECT * FROM ALL_SA_POLICIES;
438 END pro_select_OLS_POLICIES;
439
440 -- Thu tuc SELECT_STATUS_OLS_POLICIES
441 PROCEDURE pro_select_Status_OLS_POLICIES(policyName IN VARCHAR2, v_out OUT SYS_REFCURSOR) IS
442 BEGIN
443     OPEN v_out FOR
444         SELECT status FROM ALL_SA_POLICIES WHERE POLICY_NAME = policyName;
445 END pro_select_Status_OLS_POLICIES;
446 END packg_disable_enable_policy;

```

- Giao diện enable và disable policy: bắt buộc sử dụng lại giao diện để Đăng nhập trước khi thực hiện:

The screenshot shows a 'Login' window with the following fields and values:

Field	Value
Host:	localhost
Port:	1521
Sid:	orcl
User:	
PassWord:	

At the bottom center, there is a button labeled 'Login'.

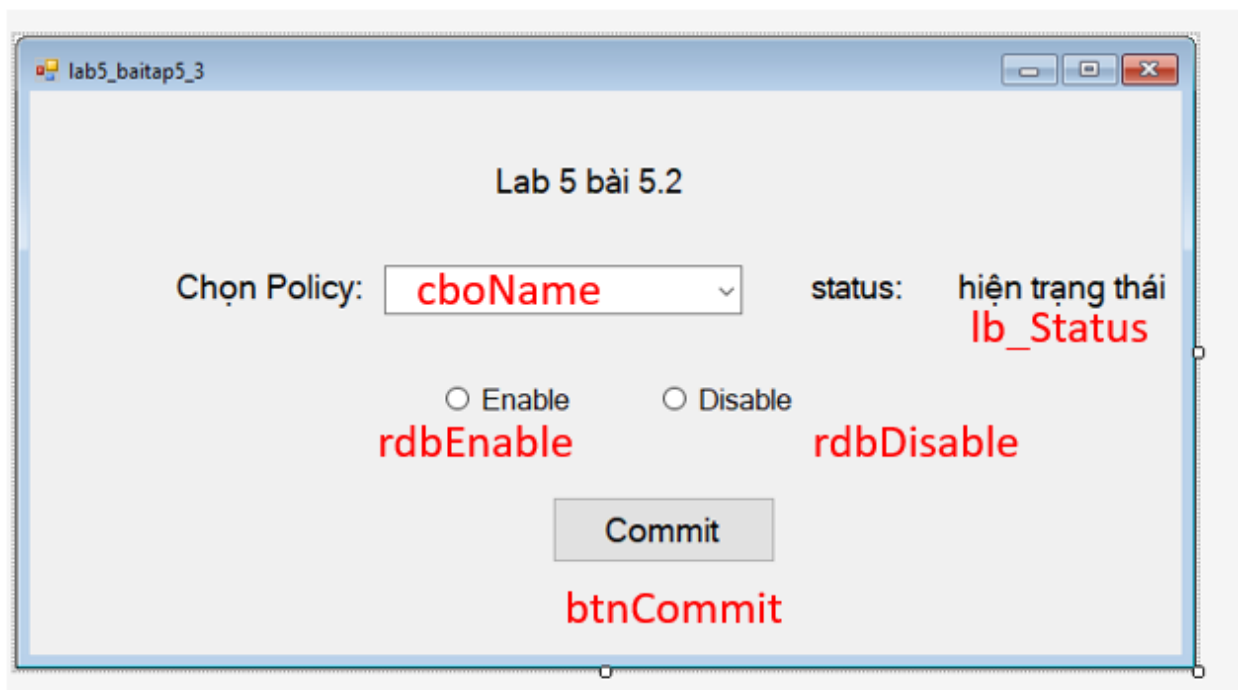
```

1 reference
private void btn_login_Click(object sender, EventArgs e)
{
    string host = txt_host.Text;
    string port = txt_port.Text;
    string sid = txt_sid.Text;
    string user = txt_user.Text;
    string pass = txt_password.Text;

    if (Check_Textbox(host, port, sid, user, pass))
    {
        Database.Set_Database(host, port, sid, user, pass);
        if (Database.Connect())
        {
            OracleConnection c = Database.Get_Connect();
            MessageBox.Show("Đăng nhập thành công\nServerVersion: " + c.ServerVersion);
            // Hiện thị form2
            lab5_baitap5_3 form2 = new lab5_baitap5_3();
            this.Hide(); // Ẩn form1
            form2.Show();
        }
        else
        {
            MessageBox.Show("Đăng nhập thất bại");
        }
    }
}

```

- Giao diện bài tập: phần chữ màu đỏ là tên Design được định nghĩa khi tạo thành phần, giúp dễ gọi và sử dụng trong code C#



- Code hiển thị policy hiện có:

4 references

```
public partial class lab5_baitap5_3 : Form
{
```

```
    private OracleConnection conn;
```

1 reference

```
    public lab5_baitap5_3()
```

```
    {
```

```
        InitializeComponent();
```

```
        CenterToScreen();
```

```
        conn = Database.Get_Connect();
```

```
    }
```

1 reference

```
    private void lab5_baitap5_3_Load(object sender, EventArgs e)
```

```
    {
```

```
        LoadPolicyComboBox(conn);
```

```
        LoadStatus(conn);
```

```
    }
```

1 reference

```
private void LoadPolicyComboBox(OracleConnection connection)
```

```
{
```

```
    try
```

```
    {
```

```
        using (OracleCommand command = new OracleCommand("packg_disable_enable_policy.pro_select_OLS_POLICIES", connection))
```

```
        {
```

```
            command.CommandType = CommandType.StoredProcedure;
```

```
            // Tạo tham số output
```

```
            OracleParameter outParam = new OracleParameter("v_out", OracleDbType.RefCursor);
```

```
            outParam.Direction = ParameterDirection.Output;
```

```
            command.Parameters.Add(outParam);
```

```
            // Thực thi thủ tục và nhận tập kết quả
```

```
            using (OracleDataReader reader = command.ExecuteReader())
```

```
            {
```

```
                cboName.Items.Clear();
```

```
                while (reader.Read())
```

```
                {
```

```
                    string policyName = reader.GetString(0);
```

```
                    cboName.Items.Add(policyName);
```

```
                    cboName.SelectedIndex = 0;
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
    catch (OracleException ex)
```

```
    {
```

```
        MessageBox.Show("Error Select policy: " + ex.Message);
```

```
    }
```

```
}
```

- Nút hiển thị trạng thái của policy

```

4 references
private void LoadStatus(OracleConnection connection)
{
    try
    {
        string policyName = cboName.SelectedItem.ToString();
        using (OracleCommand command = new OracleCommand("packg_disable_enable_policy.pro_select_Status_OLS_POLICIES", connection))
        {
            command.CommandType = CommandType.StoredProcedure;

            // Thêm tham số đầu vào
            command.Parameters.Add("policyName", OracleDbType.Varchar2).Value = policyName;

            // Tạo tham số output
            OracleParameter outParam = new OracleParameter("v_out", OracleDbType.RefCursor);
            outParam.Direction = ParameterDirection.Output;
            command.Parameters.Add(outParam);

            // Thực thi thủ tục và nhận tập kết quả
            using (OracleDataReader reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    string userName = reader.GetString(0);
                    lb_Status.Text = userName;
                }
            }
        }
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Error Select Status policy: " + ex.Message);
    }
}

1 reference
private void cboName_SelectedIndexChanged(object sender, EventArgs e)
{
    LoadStatus(conn);
}

```


- Code nút commit:

```

1 reference
private void btnCommit_Click(object sender, EventArgs e)
{
    object selectedItem = cboName.SelectedItem;
    string policyName = cboName.SelectedItem.ToString();
    // Kiểm tra xem đã chọn RadioButton chưa
    if (rdbEnable.Checked || rdbDisable.Checked)
    {
        // Lấy ra giá trị đã chọn
        if (rdbEnable.Checked)
        {
            using (OracleCommand cmd = new OracleCommand("pro_enable_policy", conn))
            {
                cmd.CommandType = CommandType.StoredProcedure;
                // Thêm tham số vào stored procedure
                cmd.Parameters.Add("policyName", OracleDbType.Varchar2).Value = policyName;
                try
                {
                    // Thực thi stored procedure
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Policy Enable successfully!");
                }
                catch (OracleException ex)
                {
                    MessageBox.Show("Error Enable policy: " + ex.Message);
                }
            }
        }
        else
        {
            using (OracleCommand cmd = new OracleCommand("pro_disable_policy", conn))
            {
                cmd.CommandType = CommandType.StoredProcedure;

                // Thêm tham số vào stored procedure
                cmd.Parameters.Add("policyName", OracleDbType.Varchar2).Value = policyName;

                try
                {
                    // Thực thi stored procedure
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Policy Disable successfully!");
                }
                catch (OracleException ex)
                {
                    MessageBox.Show("Error Disable policy: " + ex.Message);
                }
            }
        }
    }
    else
    {
        MessageBox.Show("Vui lòng chọn Enable hoặc Disable để tiếp tục");
    }
}

```

5.4. Tạo các thành phần nhãn cho chính sách region_policy:

5.4.1. Level: level 1, level 2, level 3

```
9  --5.4.1 tao level
10 -- ket noi user ols_test de thuc hien:
11 EXECUTE sa_components.create_level('region_policy',300,'LV3','level 3');
12 EXECUTE sa_components.create_level('region_policy',200,'LV2','level 2');
13 EXECUTE sa_components.create_level('region_policy',100,'LV1','level 1');
14
```

Script Output x

Task completed in 0.094 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

```
15 -- thu tuc tao level
16 create or replace procedure pro_create_level
17 (policyName in VARCHAR2,lvnum in int,shortName in VARCHAR2,longName in VARCHAR2 )
18 is
19 BEGIN
20     sa_components.create_level
21     (policy_name => policyName ,
22     level_num => lvnum,
23     short_name => shortName,
24     long_name => longName);
25 END;
26 /
```

Script Output x

Task completed in 0.131 seconds

Procedure PRO_CREATE_LEVEL compiled

5.4.2. Compartment: MANAGEMENT, EMPLOYEE

```
27 --5.4.2 tao compartment
28 EXECUTE sa_components.create_compartment ('region_policy',110,'MAN','MANAGEMENT');
29 EXECUTE sa_components.create_compartment ('region_policy',100,'EMP','EMPLOYEE');
30
```

Script Output x
Task completed in 0.043 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

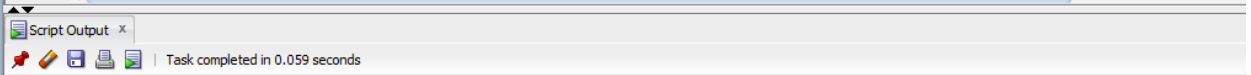
```
32 -- thu tuc tao compartment
33 create or replace procedure pro_create_compartment
34 (policyName in VARCHAR2,cpnum in int,shortName in VARCHAR2,longName in VARCHAR2 )
35 is
36 BEGIN
37     sa_components.create_compartment
38     (policy_name => policyName ,
39     comp_num => cpnum,
40     short_name => shortName,
41     long_name => longName);
42 END;
43 /
```

Script Output x
Task completed in 0.046 seconds

Procedure PRO_CREATE_COMPARTMENT compiled

5.4.3. Group: REGION NORTH, REGION SOUTH, REGION EAST, REGION WEST

```
44 --5.4.3 tao group
45 EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',50,'RN','REGION NORTH');
46 EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',40,'RS','REGION SOUTH');
47 EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',30,'RW','REGION WEST');
48 EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',20,'RE','REGION EAST');
```



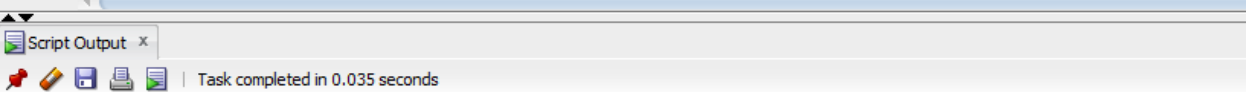
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

```
49 -- thu tuc tao group
50 create or replace procedure pro_create_group
51 (policyName in VARCHAR2, cpnum in int, shortName in VARCHAR2,
52 longName in VARCHAR2 , parentName in VARCHAR2)
53 is
54 BEGIN
55     sa_components.CREATE_GROUP
56     (policy_name => policyName ,
57     group_num => cpnum,
58     short_name => shortName,
59     long_name => longName,
60     parent_name => parentName);
61 END;
```



Procedure PRO_CREATE_GROUP compiled

```
63 --thu tuc select tat ca policy ols hien co
64 create or REPLACE procedure pro_select_OLS_POLICIES
65 ( v_out out SYS_REFCURSOR )
66 is
67 begin
68 open v_out for
69 select * from ALL_SA_POLICIES;
70 end ;
71 /
```

Script Output x Query Result x
Task completed in 0.032 seconds

Procedure PRO_SELECT_OLS_POLICIES compiled

- Những thủ tục dùng để hiển thị các thành phần ra chi tiết của từng thành phần trong 1 chính sách của OLS:
- Thủ tục select short name level của policy đã chọn

```

73 --thu tục select level ung voi 1 policy
74 create or REPLACE procedure pro_select_levels
75 (policyName in VARCHAR2 , v_out out SYS_REFCURSOR )
76 is
77 begin
78 open v_out for
79 select Short_name from ALL_SA_LEVELS
80 where POLICY_NAME = policyName order by Short_name ASC ;
81 end ;
82 /

```

Script Output x
Task completed in 0.041 seconds

Procedure PRO_SELECT_LEVELS compiled

- Thủ tục select chi tiết level dựa vào short name của policy đã chọn

```

83 -- thu tục select chi tiet level tuong ung
84 create or REPLACE procedure pro_select_CTlevels
85 (policyName in VARCHAR2 , v_out out SYS_REFCURSOR , ShortName in VARCHAR2)
86 is
87 begin
88 open v_out for
89 select * from ALL_SA_LEVELS
90 where POLICY_NAME = policyName and Short_name = ShortName ;
91 end ;
92 /

```

Script Output x
Task completed in 0.025 seconds

Procedure PRO_SELECT_CTLEVELS compiled

- Thủ tục select short name compartments của policy đã chọn

```
93 --thu tục select COMPARTMENTS ung voi 1 policy
94 create or REPLACE procedure pro_select_COMPARTMENTS
95 (policyName in VARCHAR2 , v_out out SYS_REFCURSOR )
96 is
97 begin
98 open v_out for
99 select Short_name from ALL_SA_COMPARTMENTS
100 where POLICY_NAME = policyName ;
101 end ;
102 /
```

Script Output x
Task completed in 0.034 seconds

Procedure PRO_SELECT_COMPARTMENTS compiled

- Thủ tục select chi tiết compartments dựa vào short name của policy đã chọn

```
103 -- thu tục select chi tiet COMPARTMENTS tuong ung
104 create or REPLACE procedure pro_select_CTCOMPARTMENTS
105 (policyName in VARCHAR2 , v_out out SYS_REFCURSOR ,ShortName in VARCHAR2)
106 is
107 begin
108 open v_out for
109 select * from ALL_SA_COMPARTMENTS
110 where POLICY_NAME = policyName and Short_name = ShortName ;
111 end ;
112 /
```

Script Output x
Task completed in 0.034 seconds

Procedure PRO_SELECT_CTCOMPARTMENTS compiled

- Thủ tục select short name group của policy đã chọn

```
113 --thu tuc select GROUPS ung voi 1 policy
114 create or REPLACE procedure pro_select_GROUPS
115 (policyName in VARCHAR2 , v_out out SYS_REFCURSOR )
116 is
117 begin
118 open v_out for
119     select Short_name from ALL_SA_GROUPS where POLICY_NAME = policyName ;
120 end ;
121 /
```

Script Output x
Task completed in 0.022 seconds

Procedure PRO_SELECT_GROUPS compiled

- Thủ tục select chi tiết group dựa vào short name của policy đã chọn

```
122 -- thu tuc select chi tiet GROUPS tuong ung
123 create or REPLACE procedure pro_select_CTGROUPS
124 (policyName in VARCHAR2 , v_out out SYS_REFCURSOR ,ShortName in VARCHAR2)
125 is
126 begin
127 open v_out for
128     select * from ALL_SA_GROUPS
129     where POLICY_NAME = policyName and Short_name = ShortName ;
130 end ;
```

Script Output x
Task completed in 0.037 seconds

Procedure PRO_SELECT_CTGROUPS compiled

- Giao diện tạo thành phần của 1 policy: sử dụng form đăng nhập đã xây dựng ở những bài trước để đăng nhập
- Phần chữ màu đỏ là tên Design của cái thành phần, giúp dễ đọc code hơn
- Các sử dụng:
 - o Chọn policy
 - o Chọn thành phần cần tạo
 - o Nhập thông tin thành phần
 - o Nhấn tạo.

baiTap5_4

Tạo Oracle Label Security

Chọn Policy: cboPolicy

Chọn thành phần cboThanhPhan Chi tiết: cboChiTiet

Number: txtNumber

Short name: txtShortName

Long name: txtLongName

Parent name: cbo_groupParent

btnCreate Tạo Thoát btnClose

- Code

```
4 references
public partial class baiTap5_4 : Form
{
    private OracleConnection conn;
    string tenPolicy, tenThanhPhan, CT_ThanhPhan;
    1 reference
    public baiTap5_4()
    {
        InitializeComponent();
        CenterToScreen();
        conn = Database.Get_Connect();
    }

    1 reference
    private void baiTap5_4_Load(object sender, EventArgs e)
    {
        loadPolicyComboBox(conn);
        GanThanhPhanComboBox();
        cbo_groupParent.Enabled = false;
    }

    //Hàm lấy danh sách policy hiện có
    1 reference
    private void loadPolicyComboBox(OracleConnection connection)
    {
        resetText();
        try
        {
            using (OracleCommand command = new OracleCommand("LBACSYS.pro_select_OLS_POLICIES", connection))
            {
                command.CommandType = CommandType.StoredProcedure;

                // Tạo tham số output
                OracleParameter outParam = new OracleParameter("v_out", OracleDbType.RefCursor);
                outParam.Direction = ParameterDirection.Output;
                command.Parameters.Add(outParam);

                // Thực thi thủ tục và nhận tập kết quả
                using (OracleDataReader reader = command.ExecuteReader())
                {
                    cboPolicy.Items.Clear();
                    while (reader.Read())
                    {
                        string policyName = reader.GetString(0);
                        cboPolicy.Items.Add(policyName);
                        cboPolicy.SelectedIndex = 0;
                    }
                }
            }
        }
        catch (OracleException ex)
        {
            MessageBox.Show("Error Select policy: " + ex.Message);
        }
    }
}
```

```

69 // hiển thị policy thành phần của từng level, group, compartment tương ứng
70 private void loadThanhPhan(OracleConnection conn)
71 {
72     resetText();
73     cboChiTiet.Items.Clear();
74     tenPolicy = cboPolicy.SelectedItem.ToString();
75     tenThanhPhan = cboThanhPhan.SelectedItem.ToString();
76
77     string query = "";
78
79     try
80     {
81         if (tenThanhPhan.Equals("Level"))
82         {
83             query = "LBACSYS.pro_select_levels";
84         }
85         else if (tenThanhPhan.Equals("Compartment"))
86         {
87             query = "LBACSYS.pro_select_COMPARTMENTS";
88         }
89         else if (tenThanhPhan.Equals("Group"))
90         {
91             query = "LBACSYS.pro_select_GROUPS";
92         }
93     }
94     using (OracleCommand command = new OracleCommand(query, conn))
95     {
96         command.CommandType = CommandType.StoredProcedure;
97         // Thêm tham số đầu vào
98         command.Parameters.Add("1", OracleDbType.Varchar2).Value = tenPolicy;
99         // Tạo tham số output
100         OracleParameter outParam = new OracleParameter("v_out", OracleDbType.RefCursor);
101         outParam.Direction = ParameterDirection.Output;
102         command.Parameters.Add(outParam);
103
104         // Thực thi thủ tục
105         command.ExecuteNonQuery();
106
107         // Lấy dữ liệu từ tham số output
108         using (OracleDataReader reader = command.ExecuteReader())
109         {
110             while (reader.Read())
111             {
112                 cboChiTiet.Items.Add(reader.GetString(0));
113                 cboChiTiet.SelectedIndex = 0;
114             }
115         }
116     }
117 }
118
119 catch (OracleException ex)
120 {
121     MessageBox.Show("Error select policy: " + query + ex.Message);
122 }
123
124 }
125

```

```

126 // hiển thị policy chi tiết theo level, group, compartment tương ứng
127 1 reference private void load_ChiTiet_ThanhPhan(OracleConnection connection)
128 {
129     tenPolicy = cboPolicy.SelectedItem.ToString();
130     tenThanhPhan = cboThanhPhan.SelectedItem.ToString();
131     CT_ThanhPhan = cboChiTiet.SelectedItem.ToString();
132     string query = "";
133     try
134     {
135         if (tenThanhPhan.Equals("Level"))
136         {
137             query = "LBACSYS.pro_select_CTlevels";
138         }
139         else if (tenThanhPhan.Equals("Compartment"))
140         {
141             query = "LBACSYS.pro_select_CTCOMPARTMENTS";
142         }
143         else if (tenThanhPhan.Equals("Group"))
144         {
145             query = "LBACSYS.pro_select_CTGROUPS";
146         }
147     }
148     using (OracleCommand command = new OracleCommand(query, connection))
149     {
150         command.CommandType = CommandType.StoredProcedure;
151
152         // Thêm tham số đầu vào
153         command.Parameters.Add("policyName", OracleDbType.Varchar2).Value = tenPolicy;
154
155         // Tạo tham số output
156         OracleParameter outParam = new OracleParameter("v_out", OracleDbType.RefCursor);
157         outParam.Direction = ParameterDirection.Output;
158         command.Parameters.Add(outParam);
159
160         command.Parameters.Add("ShortName", OracleDbType.Varchar2).Value = CT_ThanhPhan;
161         // Lấy dữ liệu từ tham số output
162         using (OracleDataReader reader = command.ExecuteReader())
163         {
164             while (reader.Read())
165             {
166                 cbo_groupParent.Items.Clear();
167                 cbo_groupParent.Text = string.Empty;
168
169                 txtNumber.Text = reader.GetInt32(1).ToString();
170                 txtShortName.Text = reader.GetString(2);
171                 txtLongName.Text = reader.GetString(3);
172                 if (tenThanhPhan.Equals("Group") && !reader.IsDBNull(5))
173                 {
174                     cbo_groupParent.Items.Add(reader.GetString(5));
175                     cbo_groupParent.Items.Add("");
176                     cbo_groupParent.SelectedIndex = 0;
177                 }
178             }
179         }
180     }
181     catch (OracleException ex)
182     {
183         MessageBox.Show("Error select policy: " + query + ex.Message);
184     }
185 }

```

1 reference

```
193 private void cboThanhPhan_SelectedIndexChanged(object sender, EventArgs e)
194 {
195
196     loadThanhPhan(conn);
197     if (tenThanhPhan.Equals("Group"))
198     {
199         cbo_groupParent.Enabled = true;
200     }
201     else
202     {
203         cbo_groupParent.Enabled = false;
204     }
205 }
206
207 }
208
```

```

208
209 //hàm gán giá trị cho combobox thành phần
    3 references
210 private void GanThanhPhanComboBox()
211 {
212     cboThanhPhan.Items.Clear();
213     cboThanhPhan.Items.Add("Level");
214     cboThanhPhan.Items.Add("Compartment");
215     cboThanhPhan.Items.Add("Group");
216     cboThanhPhan.SelectedIndex = 0;
217
218 }
219
220 // hàm xóa hết giá trị trước khi gán giá trị mới
    2 references
221 private void resetText()
222 {
223
224     txtNumber.Text = string.Empty;
225     txtShortName.Text = string.Empty;
226     txtLongName.Text = string.Empty;
227     cbo_groupParent.Items.Clear();
228 }
    1 reference
229 private void btnClose_Click(object sender, EventArgs e)
230 {
231     this.Close();
232     Application.Exit();
233 }
    1 reference
235 private void cboChiTiet_SelectedIndexChanged(object sender, EventArgs e)
236 {
237
238
239     load_ChiTiet_ThanhPhan(conn);
240 }
    1 reference
242 private void btnCreate_Click(object sender, EventArgs e)
243 {
244     create(conn);
245 }
246

```

```

247 // hàm tạo
248 1 reference
249 private void create(OracleConnection connection)
250 {
251     tenPolicy = cboPolicy.SelectedItem.ToString();
252
253     try
254     {
255         cboChiTiet.Items.Clear();
256         string query = "";
257
258         tenThanhPhan = cboThanhPhan.SelectedItem.ToString();
259         //int number = int.Parse(txtNumber.Text);
260
261         int number;
262
263         if (!int.TryParse(txtNumber.Text, out number))
264         {
265             MessageBox.Show("Số nhập vào không hợp lệ.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
266             return;
267         }
268         string shortName = txtShortName.Text;
269         string longName = txtLongName.Text;
270         string parentName = "";
271         if (tenThanhPhan.Equals("Level"))
272         {
273             query = "LBACSYS.pro_create_level";
274         }
275         else if (tenThanhPhan.Equals("Compartment"))
276         {
277             query = "LBACSYS.pro_create_compartment";
278         }
279         else if (tenThanhPhan.Equals("Group"))
280         {
281             query = "LBACSYS.pro_create_group";
282             parentName = cbo_groupParent.SelectedItem.ToString();
283         }
284
285         using (OracleCommand command = new OracleCommand(query, connection))
286         {
287             command.CommandType = CommandType.StoredProcedure;
288
289

```

```

290
291 // Thêm tham số đầu vào
292 command.Parameters.Add(new OracleParameter("1", OracleDbType.Varchar2)).Value = tenPolicy;
293 command.Parameters.Add(new OracleParameter("2", OracleDbType.Int16)).Value = number;
294 command.Parameters.Add(new OracleParameter("3", OracleDbType.Varchar2)).Value = shortName;
295 command.Parameters.Add(new OracleParameter("4", OracleDbType.Varchar2)).Value = longName;
296 if (tenThanhPhan.Equals("Group"))
297 {
298     command.Parameters.Add("parentName", OracleDbType.Varchar2).Value = parentName;
299 }
300
301 DialogResult result = MessageBox.Show(
302     "Bạn có chắc chắn muốn tạo " + tenThanhPhan + " không ?", "Xác nhận tạo",
303     MessageBoxButtons.YesNo, MessageBoxIcon.Question);
304
305 if (result == DialogResult.Yes)
306 {
307     // Thực thi thủ tục
308     command.ExecuteNonQuery();
309     MessageBox.Show("Thêm mới thành công.");
310 }
311 }
312
313 GanThanhPhanComboBox();
314 }
315 catch (OracleException ex)
316 {
317     MessageBox.Show("Error create : " + ex.Message);
318 }
319 }
320
321 }
322
323
324 1 reference
325 private void cboPolicy_SelectedIndexChanged(object sender, EventArgs e)
326 {
327     GanThanhPhanComboBox();
328     loadThanhPhan(conn);
329 }
330
331 }
332

```