Bài thực hành số 8

ORACLE LABEL SECURITY (3)

❖ *Tóm tắt nội dung:*

- Che dấu cột thông tin chính sách
- Sử dụng hàm gán nhãn
- Các view của OLS

I. Một số kỹ thuật nâng cao trong OLS

A. Lý thuyết

1. Che giấu cột thông tin nhãn dữ liệu

- Để tránh việc hiển thị cột chứa thông tin nhãn của chính sách[1], người quản trị có thể thiết lập tùy chọn HIDE khi gán chính sách cho bảng.
- Một khi chính sách đã được áp dụng, trạng thái Ẩn/Không Ẩn của cột không thể được thay đổi trừ khi ta remove chính sách khỏi bảng với tham số DROP_COLUMN bằng TRUE. Sau đó chính sách có thể được áp dụng lại với trạng thái mới.
- Khi người dùng INSERT dữ liệu vào bảng có trạng thái ẩn cột chính sách, giá trị của cột chứa nhãn sẽ không bị yêu cầu phải insert.
- Câu lệnh SELECT * sẽ không tự động trả về giá trị của cột ẩn, trừ khi nó được truy xuất trực tiếp.
- Câu lệnh DESCRIBE cũng sẽ không hiển thị thông tin cột ẩn.

2. Hàm gán nhãn

- Trong bài lab 7 sinh viên đã được giới thiệu một số cách để gán nhãn chính sách cho một hàng dữ liệu và đã được thực hành gán tường minh nhãn cho từng dòng dữ liệu. Tuy nhiên, có những bảng dữ liệu lớn, không thể gán nhãn cho từng trường hợp. Thay vào đó, ta có thể sử dụng một hàm (function) do mình hiện thực để OLS

[1] Sinh viên xem lại khái niệm label tag ở lab 6.

sẽ tự động gán nhãn mỗi khi có hàng mới được insert vào bảng dữ liệu được bảo vệ. Xem phần thực hành để hiểu rõ hơn về cách thức làm việc này.

- Hàm gán nhãn sẽ override 2 tùy chọn LABEL_DEFAULT và LABEL_UPDATE.
- Kết quả trả về của hàm gán nhãn thuộc kiểu dữ liệu LBACSYS.LBAC_LABEL. Hàm TO_ LBAC_DATA_LABEL dùng để chuyển đổi một nhãn ở kiểu chuỗi thành kiểu LBACSYS.LBAC_LABEL. Lưu ý, tài khoản LBACSYS phải có quyền EXECUTE trên hàm gán nhãn. Chủ sỡ hữu của hàm gán nhãn phải có quyền EXECUTE trên hàm TO_ LBAC_DATA_LABEL với tuỳ chọn WITH GRANT OPTION.

B. Thực hành

1. Che dấu cột thông tin chính sách

- Do trong bài lab trước, ta đã áp dụng chính sách cho bảng mà không có tùy chọn HIDE nên trong bài lab này ta phải remove chính sách (xóa cả cột thông tin chính sách), thực hiện lại đoạn code gán nhãn trong bài lab trước và gán lại chính sách.

```
CONN sec_admin/secadmin;
BEGIN
    sa_policy_admin.remove_table_policy
       (policy_name    => 'ACCESS_LOCATIONS',
       schema_name     => 'HR',
       table_name      => 'LOCATIONS',
       drop_column     => true);
END;
/
SELECT * FROM hr.locations;
```

Ta nhận thấy bây giờ cột OLS_COLUMN đã được xóa.

- Gán lại chính sách cho bảng với NO_CONTROL và HIDE:

```
CONN sec_admin/secadmin;
BEGIN
   sa_policy_admin.apply_table_policy
```

```
      (policy_name     => 'ACCESS_LOCATIONS',
      schema_name      => 'HR',
      table_name       => 'LOCATIONS',
      table_options    => 'HIDE,NO_CONTROL');
END;
/
```

- Gán lại nhãn cho dữ liệu trong bảng (do lúc remove đã xóa mất cột chứa thông tin chính sách):

```
CONN sec_admin/secadmin;
UPDATE hr.locations SET ols_column = char_to_label
('ACCESS_LOCATIONS', 'CONF');

UPDATE hr.locations SET ols_column = char_to_label
('ACCESS_LOCATIONS', 'CONF::US')
WHERE country_id = 'US';

UPDATE hr.locations SET ols_column = char_to_label
('ACCESS_LOCATIONS', 'CONF::UK')
WHERE country_id = 'UK';

UPDATE hr.locations SET ols_column = char_to_label
('ACCESS_LOCATIONS', 'CONF::CA')
WHERE country_id = 'CA';
UPDATE hr.locations SET ols_column = char_to_label
('ACCESS_LOCATIONS', 'CONF:SM:UK,CA')
WHERE (country_id = 'CA' and city = 'Toronto')
or (country_id = 'UK' and city = 'Oxford');

UPDATE hr.locations SET ols_column = char_to_label
('ACCESS_LOCATIONS', 'CONF:HR:UK')
WHERE country_id = 'UK' and city = 'London';

UPDATE hr.locations SET ols_column = char_to_label
```

```
('ACCESS_LOCATIONS', 'SENS:HR,SM,FIN:CORP')
WHERE country_id = 'CH' and city = 'Geneva';
COMMIT ;
```

- Tiếp theo ta cần gán lại chính sách với tùy chọn HIDE và READ_CONTROL:

```
CONN sec_admin/secadmin;
BEGIN
    sa_policy_admin.remove_table_policy
       (policy_name   => 'ACCESS_LOCATIONS',
       schema_name    => 'HR',
       table_name     => 'LOCATIONS');

       sa_policy_admin.apply_table_policy
       (policy_name    => 'ACCESS_LOCATIONS',
       schema_name     => 'HR',
       table_name      => 'LOCATIONS',
       table_options    =>
       'HIDE,READ_CONTROL,WRITE_CONTROL,CHECK_CONTROL');
END;
/
```

- Bây giờ ta thử truy xuất bảng Locations:

```
CONN sec_admin/secadmin;
SELECT * FROM hr.locations;

no rows selected

DESCRIBE hr.locations;

Name              Null?      Type
--------------- ---------- ------
LOCATION_ID       NOT NULL   NUMBER(4)
STREET_ADDRESS               VARCHAR2(40)
POSTAL_CODE                  VARCHAR2(12)
```

```
CITY                NOT NULL    VARCHAR2(30)
STATE_PROVINCE                  VARCHAR2(25)
COUNTRY_ID                      CHAR(2)
```

- Kết quả của lệnh SELECT là "no rows selected". Chỉ có lệnh DESCRIBE có trả về kết quả. Nguyên nhân là do bây giờ bảng này đã được bảo vệ, chỉ những người được cấp quyền OLS cụ thể mới có thể truy xuất. Ta log in lại bằng user SKING:

```
CONN sking/sking;
SELECT * FROM hr.locations WHERE city = 'Bern';
SELECT  label_to_char (ols_column) as label, locations.*
       FROM hr.locations WHERE city = 'Bern';
```

- Ta thấy trong câu lệnh SELECT thứ 2, ta có chỉ định rõ cột *ols_column* nên cột này mới xuất hiện. Trong kết quả truy vấn của câu SELECT thứ nhất không có cột thông tin chính sách này.

2. Dùng hàm gán nhãn

- Trong phần thực hành này, ta sẽ dùng bảng Employees của schema HR để minh họa.
- Cấp các quyền cần thiết cho sec_admin trên bảng Employees:

```
CONN system/system;
GRANT select, insert, update ON hr.employees TO sking;
GRANT select, insert, update ON hr.employees TO sec_admin;
GRANT create procedure TO sec_admin;

CONN lbacsys/lbacsys;
GRANT execute ON to_lbac_data_label
       TO sec_admin WITH GRANT OPTION;
```

- Tiếp theo ta viết một hàm gán nhãn dựa trên điều kiện của thông tin nhân viên:

```
CONN sec_admin/secadmin;
CREATE OR REPLACE FUNCTION sec_admin.gen_emp_label
     (Job varchar2, Sal number)
```

```
RETURN LBACSYS.LBAC_LABEL
AS
  i_label varchar2(80);
BEGIN
/************* Xác định level *************/
   IF Sal > 17000 THEN
      i_label := 'SENS:';
   ELSIF Sal > 10000 THEN
      i_label := 'CONF:';
   ELSE
      i_label := 'PUB:';
   END IF;
/************* Xác định compartment *************/
   IF Job LIKE '%HR%' THEN
      i_label := i_label||'HR:';
   ELSIF (Job LIKE '%MK%') OR (Job LIKE '%SA%')  THEN
      i_label := i_label||'SM:';
   ELSIF Job LIKE '%FI%' THEN
      i_label := i_label||'FIN:';
   ELSE
      i_label := i_label||':';
   END IF;
/************* Xác định groups *************/
   i_label := i_label||'CORP';
   RETURN TO_LBAC_DATA_LABEL('ACCESS_LOCATIONS',i_label);
END;
/
```

- Ta cần gán cho LBACSYS quyền thực thi trên hàm gán nhãn vừa được tạo:

```
CONN sec_admin/secadmin;
GRANT execute ON sec_admin.gen_emp_label TO lbacsys;
```

- Ta chỉ định thủ tục vừa hiện thực làm hàm gán nhãn cho bảng `Employees`:

```
CONN sec_admin/secadmin;
```

```
BEGIN
   SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
      policy_name       => 'ACCESS_LOCATIONS',
      schema_name       => 'HR',
      table_name        => 'EMPLOYEES',
      table_options     =>
      'READ_CONTROL,WRITE_CONTROL,CHECK_CONTROL',
      label_function    => 'sec_admin.gen_emp_label
              (:new.job_id, :new.salary)',
      PREDICATE => NULL);
END;
/
```

- Sinh viên tự kiểm tra kết quả của hàm gán nhãn, bằng cách log in vào tài khoản có quyền thao tác trên bảng EMPLOYEES (ví dụ SKING) rồi insert, update một dòng dữ liệu trong bảng và xem xét giá trị của cột chứa nhãn dữ liệu.

II. Các view thông tin của OLS

- Các thông tin về các chính sách của OLS được lưu trong data dictionary. Ta có thể xem các thông tin này thông qua các view của OLS.
- View DBA_SA_USERS: hiển thị thông tin về tất cả các chính sách có trong CSDL.
- DBA_SA_USER_LEVELS: hiển thị thông tin level của mọi người dùng.
- DBA_SA_USER_COMPARTMENTS: hiển thị thông tin compartment của mọi người dùng.
- DBA_SA_USER_GROUPS: hiển thị thông tin group của mọi người dùng.
- Để xem tất cả các view trên cần log in vào tài khoản LBACSYS hoặc được cấp quyền SELECT từ LBACSYS.

```
conn lbacsys/lbacsys;
select * from DBA_SA_USERS;
select * from DBA_SA_USER_LEVELS;
select * from DBA_SA_USER_COMPARTMENTS;
select * from DBA_SA_USER_GROUPS;
```

III. Bài tập

1. Viết hàm gán nhãn GET_CUSTOMER_LABEL cho các khách hàng trong bảng CUSTOMERS đã tạo ở bài lab 7 theo điều kiện sau:
   - Credit > 2000: level 3; 500 < credit <= 2000: level 2; còn lại level 1.
   - Cust_type = 'Platinum' thì compartment là Manager, còn lại là Employee.
   - Group gán theo region.

2. Thực hiện các câu lệnh cần thiết để bảng trên được gán nhãn và được áp dụng chính sách REGION_POLICY đã tạo trong bài lab 6.

3. Thực hiện một số câu lệnh để kiểm tra tác dụng của chính sách.