Bài thực hành số 7

ORACLE LABEL SECURITY (2)

❖ *Tóm tắt nội dung:*

- Các loại nhãn người dùng
- Các quyền đặc biệt trên chính sách
- Các điều kiện áp dụng chính sách
- Áp dụng chính sách cho bảng

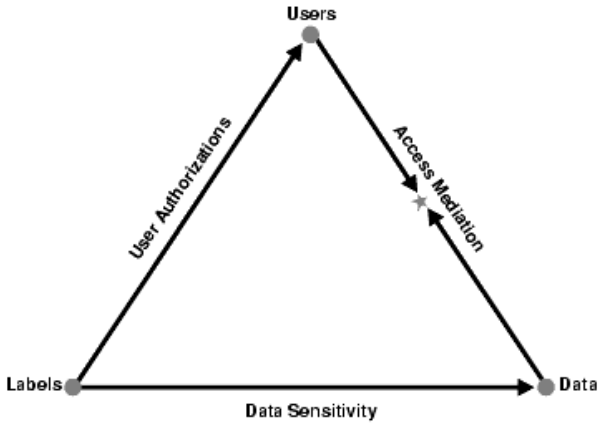
I. Các loại nhãn người dùng

A. Lý thuyết

- Trong bài *Lab 6 - Oracle Label Security (1)*, ở phần *I.A.4*, chúng ta đã nhắc đến quy trình cơ bản để xây dựng một chính sách OLS. Theo đó:
  - ✓ B4: Gán chính sách trên cho các bảng hoặc schema mà bạn muốn bảo vệ.
  - ✓ B5: Gán các giới hạn quyền, các nhãn người dùng hoặc các quyền truy xuất đặc biệt cho những người dùng liên quan.
- Thứ tự của 2 bước trên như vậy là hợp lý, vì trong OLS, khi một chính sách được chỉ định bảo vệ cho một bảng/schema, kể từ thời điểm đó bất kỳ người dùng nào cũng không thể truy xuất vào bảng/schema đó trừ khi được gán cho các *nhãn người dùng* (user label) thích hợp hoặc được cấp những quyền đặc biệt đối với chính sách đó.
- Tuy nhiên, để hiểu được tác dụng của các tùy chọn áp dụng chính sách ở bước 4, ta cần phải hiểu về các ràng buộc đối với người dùng khi truy xuất các bảng và schema được bảo vệ. Do vậy, để việc tìm hiểu về OLS được dễ dàng hơn, trong bài lab này sẽ tạm hoán đổi thứ tự tìm hiểu và thực hiện của bước 4 và bước 5. Khi đã hiểu và biết cách hiện thực một chính sách OLS, các bạn hãy thực hiện các bước theo đúng thứ tự của nó để đảm bảo tính bảo mật và toàn vẹn cho dữ liệu .

1. Nhãn người dùng (user label)

- Tại mỗi thời điểm, mỗi người dùng đều có một nhãn gọi là ***nhãn người dùng*** (***user label***). Nhãn này có tác dụng cho biết mức độ tin cậy của người dùng đối với những dữ liệu được chính sách đó bảo vệ. Nhãn người dùng cũng gồm các thành phần giống như nhãn dữ liệu. Khi một người dùng truy xuất trên bảng được bảo vệ, nhãn người dùng sẽ được so sánh với nhãn dữ liệu của mỗi dòng trong bảng để quyết định những dòng nào người dùng đó có thể truy xuất được. Hình bên dưới minh họa mối quan hệ tương ứng của ***user label*** và ***data label***.



Các nhãn thể hiện các quyền truy xuất (user authorization) được gán cho các user. Các nhãn thể hiện mức độ nhạy cảm của dữ liệu (data sensitivity) được gán cho dữ liệu. Để có thể truy xuất được dữ liệu, 2 loại nhãn này phải tương thích với nhau (access mediation).

- OLS cung cấp cho chúng ta 2 cách thức để quản lý các ***user label***: gán cụ thể từng thành phần của nhãn cho user hoặc gán nguyên nhãn cho user. Trong các phần sau sẽ trình bày kỹ hơn về 2 cách quản lý này.
- Dù sử dụng hình thức quản lý nào, mỗi người dùng cũng có một ***tập xác thực quyền*** (***set of authorizations***) để lưu giữ thông tin về quyền hạn truy xuất đối với những dữ liệu được chính sách đó bảo vệ. Tập xác thực quyền gồm có:
  - ✓ ***Level cao nhất*** (***User Max Level***) của người dùng trong các tác vụ read và write.
  - ✓ ***Level thấp nhất*** (***User Min Level***) của người dùng trong các tác vụ write. User Min Level phải thấp hơn hoặc bằng User Max Level.
  - ✓ ***Tập các compartment*** được truy xuất.

- ✓ ***Tập các group*** được truy xuất.

  (Đối với mỗi compartment và group có lưu kèm thông tin quyền truy xuất được phép là quyền "**chỉ đọc**" (***read-only***) hay quyền "**đọc-viết**" (***read-write***))

- Với tập xác thực quyền, ta có thể hình thành nên nhiều tổ hợp các thành phần của nhãn. Do vậy mỗi người dùng có thể có nhiều user label khác nhau nhưng vẫn nằm trong giới hạn của tập xác thực quyền.
- ***Session label***:
  - ✓ Session label là một user label mà người dùng sử dụng để truy xuất dữ liệu trong một session làm việc. Session label có thể là một tổ hợp bất kỳ các thành phần nằm trong giới hạn tập xác thực quyền của user đó.
  - ✓ Người quản trị có thể mô tả session label mặc định cho người dùng khi thiết lập tập xác thực quyền cho người dùng đó.
  - ✓ Bản thân người dùng có thể thay đổi session label của mình thành một nhãn bất kỳ với điều kiện là nhãn mới nằm trong giới hạn xác thực quyền của họ.
- ***Row label:***
  - ✓ Khi một hàng mới được insert vào một bảng đang được bảo vệ, cần có một nhãn dữ liệu (data label) được chỉ định cho hàng dữ liệu mới đó. Hoặc khi một hàng được update, nhãn dữ liệu của hàng đó cũng có thể bị thay đổi.
  - ✓ Những nhãn dữ liệu trong các trường hợp vừa nói ở trên có thể được gán cho dòng dữ liệu tương ứng theo một trong những cách sau:
    - Người update/insert hàng dữ liệu chỉ định một cách tường mình ngay khi thực hiện tác vụ update/insert đó.
    - Hàm gán nhãn (labeling function) của bảng đó tự sinh nhãn theo những điều kiện được hiện thực trong function tương ứng.
    - Bằng giá trị mặc định do người quản trị quy định khi gán quyền hạn truy xuất cho người dùng đó.
    - Bằng giá trị của session label của người dùng đó.
  - ✓ Tùy ngữ cảnh và trường hợp mà giá trị nhãn mới thêm vào sẽ rơi vào trường hợp nào trong các trường hợp kể trên.
  - ✓ ***Row label*** là từ dùng để chỉ những nhãn được áp dụng cho các hàng dữ liệu khi hàng đó được update hoặc insert.

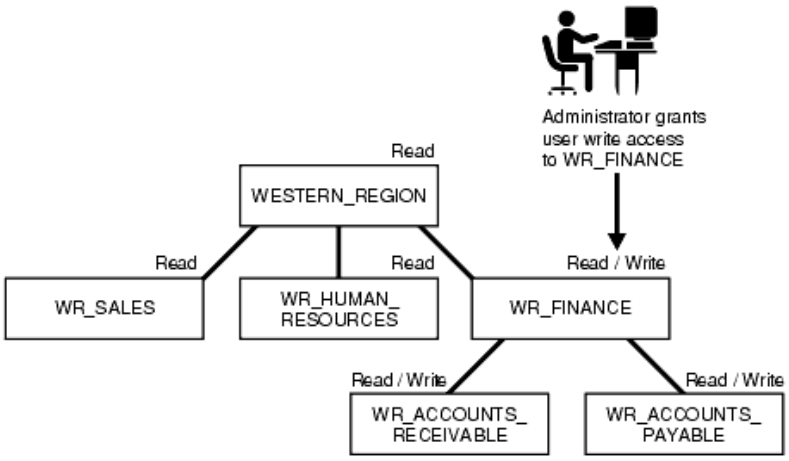
- ✓ Khi insert/update, người dùng có thể mô tả tường minh row label cho dòng dữ liệu mới được insert/update, với điều kiện row label phải thỏa **đồng thời** các điều kiện sau:
  - Level thấp hơn hoặc bằng max level của người dùng đó.
  - Level cao hơn hoặc bằng min level của người dùng đó.
  - Chỉ được chứa các compartment xuất hiện trong session label hiện tại của người dùng đó và người dùng có quyền *viết (write)* trên các compartment đó.
  - Chỉ được chứa các group xuất hiện trong session label hiện tại của người dùng đó và người dùng có quyền *viết (write)* trên các group đó.

2. Quản lý người dùng theo từng loại thành phần của nhãn

- Để gán quyền theo cách này ta cần chỉ định ra cụ thể các *level, compartment, group* mà một user có thể truy xuất.
- Để dễ hiểu phần này, người học cần nhớ lại quy tắc quản lý truy xuất của OLS mà ta đã nêu lên trong bài *Lab 6 – Oracle Label Security (1)*: **"no read up - no write up - limited write down".**
- *Quản lý các level*: gồm có 4 thông số:
  - ✓ *max_level*: level cao nhất mà người dùng có quyền đọc và viết. Vì quy tắc quản lý đòi hỏi "*no read up – no write up*" (không được đọc và viết lên những dữ liệu có độ bảo mật cao hơn độ tin cậy của user) nên *max level* chính là "giới hạn trên" cho việc truy xuất (đọc và viết) của người dùng.
  - ✓ *min_level*: level thấp nhất mà người dùng có quyền write. Vì quy tắc quản lý yêu cầu "limited write down" (chỉ viết lên những dữ liệu có độ bảo mật thấp hơn độ tin cậy của người dùng ở một mức giới hạn nào đó) nên *min level* chính là "giới hạn dưới" cho tác vụ viết của người dùng. "Giới hạn dưới" cho tác vụ đọc chính là level thấp nhất mà chính sách đó quy định.
  - ✓ *def_level*: level cho *session label* mặc định của người dùng (phải thỏa *min level <= default level <= max level*). Nếu người quản trị bảo mật không mô tả thông số này thì *default level* sẽ là *max level*.
  - ✓ *row_level*: level cho *row label* mặc định của người dùng, dùng để gán nhãn cho dữ liệu mà user đó tạo khi truy xuất bảng được bảo vệ bởi chính sách (phải thỏa

min level <= *row level* <= *max level*). Nếu người quản trị bảo mật không mô tả thông số này thì *default row level* sẽ là *default level*.

- *Quản lý các compartment*: Gồm có 4 thông số chính:
  - ✓ *read_comps*: danh sách các compartment mà người dùng được quyền đọc.
  - ✓ *write_comps*: danh sách các compartment mà người dùng được quyền viết (danh sách này phải là tập con của danh sách *read_comps*).
  - ✓ *def_comps*: danh sách các compartment cho *session label* mặc định của người dùng đó (danh sách này phải là tập con của danh sách *read_comps*).
  - ✓ *row_comps*: danh sách các compartment cho *row label* mặc định của người dùng, dùng để gán nhãn cho dữ liệu mà người dùng đó tạo khi truy xuất bảng được bảo vệ bởi chính sách (danh sách này phải là tập con của danh sách *read_comps* và *write_comps*).
- *Quản lý các group*: Gồm có 4 thông số chính:
  - ✓ *read_groups*: danh sách các groups mà người dùng được quyền đọc.
  - ✓ *write_ groups*: danh sách các groups mà người dùng được quyền viết (danh sách này phải là tập con của danh sách *read_ groups*).
  - ✓ *def_ groups*: danh sách các groups cho *session label* mặc định của người dùng đó (danh sách này phải là tập con của danh sách *read_ groups*).
  - ✓ *row_ groups*: danh sách các groups cho *row label* mặc định của người dùng đó, dùng để gán nhãn cho dữ liệu mà người dùng đó tạo ra khi truy xuất bảng được bảo vệ bởi chính sách (danh sách này phải là tập con của danh sách *read_ groups* và *write_ groups*).
- **Lưu ý**: nếu người dùng có quyền đọc trên một group thì đồng thời cũng có quyền đọc trên tất cả các group con (trực tiếp và gián tiếp) của group đó. Tương tự đối với quyền viết cũng vậy. Hình bên dưới minh họa cho việc thừa kế quyền đọc và viết trên các group. Trong hình, người dùng có quyền đọc trên group WESTERN_REGION nên cũng có quyền đọc trên tất cả các group con còn lại. Bên cạnh đó, người dùng chỉ được cấp quyền viết trên group WR_FINANCE nên chỉ có quyền viết trên group này và 2 group con của nó chứ không có quyền viết trên các group WR_SALES, WR_HUMAN_RESOURCES, WESTERN_REGION.



3. Quản lý người dùng thông qua các nhãn

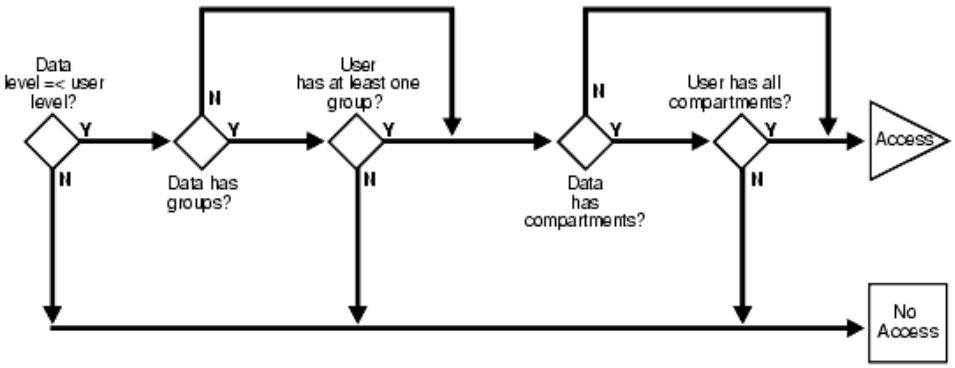
- Để tiện lợi hơn, OLS cũng cho phép người quản trị thiết lập tập xác thực quyền cho người dùng thông qua việc gán các nhãn thay vì phải chỉ định từng thành phần riêng.
- Các loại nhãn cần mô tả:
  - ✓ *max_read_label*: nhãn thể hiện mức truy xuất cao nhất đối với tác vụ đọc. Nó bao gồm level cao nhất (*max_level*) cho tác vụ đọc, tất cả các compartment và group mà người dùng được phép đọc (*read_comps* và *read_groups*). Đây là nhãn mà người quản trị **bắt buộc** phải gán cho người dùng nếu chọn cách quản lý quyền truy xuất của người dùng thông qua nhãn.
  - ✓ *max_write_label*: nhãn thể hiện mức truy xuất cao nhất đối với quyền viết. Nó bao gồm level cao nhất (*max_level*) cho tác vụ viết, tất cả các compartment và group mà người dùng được phép viết (*write_comps* và *write_groups*). Nếu người quản trị không thiết lập giá trị cho loại nhãn này, nó sẽ lấy giá trị bằng giá trị của *max_read_label*.
  - ✓ *min_write_label*: nhãn thể hiện mức truy xuất thấp nhất đối với tác vụ viết. Nhãn này chỉ chứa level thấp nhất (*min_level*) của người dùng đó, không chứa bất kỳ compartment và group nào.
  - ✓ *def_read_label*: là session label mặc định cho các tác vụ đọc của người dùng. Nó là tập con của *max_read_label*. Nếu người quản trị không thiết lập giá trị cho loại nhãn này, nó sẽ lấy giá trị bằng giá trị của *max_read_label*.
  - ✓ *def_write_label*: là session label mặc định cho tác vụ write của người dùng. Nó

là tập con của *def_read_label* (có level bằng level của *def_read_label*; chứa tất cả các compartment và group mà người dùng có quyền viết trong *def_read_label*). Giá trị của nhãn này sẽ được tính một cách **tự động** bởi OLS từ giá trị của *def_read_label*. Nói cách khác, người quản trị sẽ không mô tả giá trị cho nhãn này.

- ✓ *row_label*: nhãn mặc định dùng để gán nhãn cho các dòng dữ liệu mà user tạo ra trong bảng được chính sách bảo vệ. Nhãn này là tập con của *max_write_label* và *def_read_label*. Nếu người quản trị không thiết lập giá trị cho loại nhãn này, nó sẽ lấy giá trị bằng giá trị của *def_write_label*.

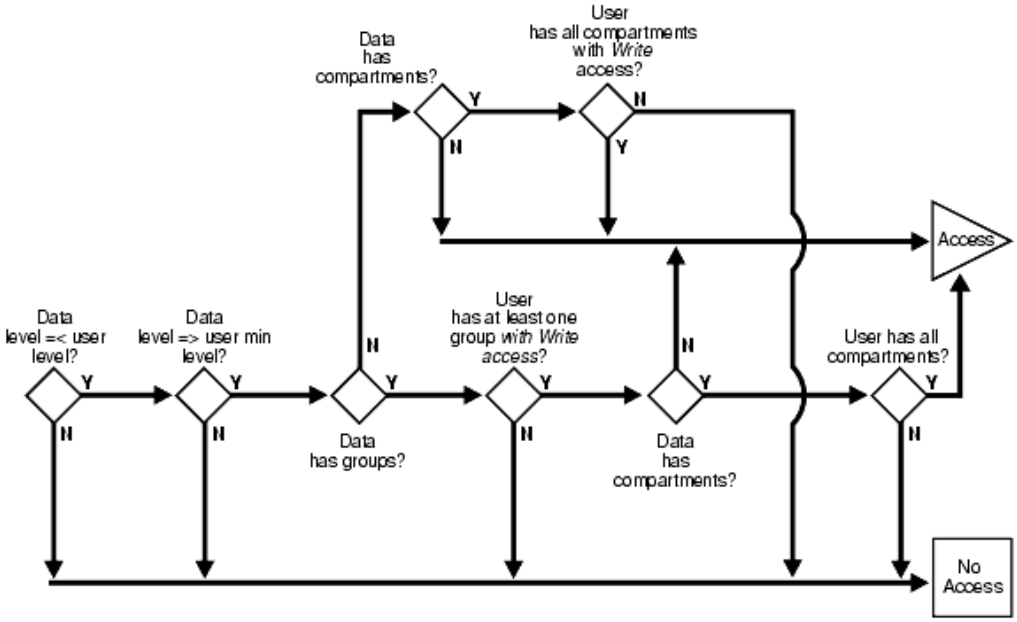
- ***Lưu ý:*** do *def_write_label* là nhãn được tính tự động từ *def_read_label*, người quản trị không cần phải thao tác trên nó nên trong các tài liệu hướng dẫn của Oracle *def_read_label* thường được gọi là *def_label*. Kể từ các phần sau, bài thực hành của chúng ta cũng sẽ áp dụng cách gọi như vậy.

4. Giải thuật bảo mật của OLS đối với tác vụ đọc



- Hình trên mô tả một cách rõ ràng cách thức mà OLS so sánh nhãn dữ liệu và nhãn người dùng tại thời điểm đó (session label) để quyết định xem người dùng có quyền đọc dòng dữ liệu đó hay không.
- Trong OLS, tác vụ đọc tương đương với lệnh SELECT.
- Nói một cách ngắn gọn, người dùng chỉ có thể đọc được dữ liệu khi thỏa đồng thời các điều kiện sau:
  - ✓ Level của session label cao hơn hoặc bằng level của dữ liệu.
  - ✓ Session label có chứa ít nhất một group nằm trong các group của data label hoặc có chứa group cha của ít nhất một group nằm trong data label.
  - ✓ Session label có chứa tất cả các compartment xuất hiện trong data label.

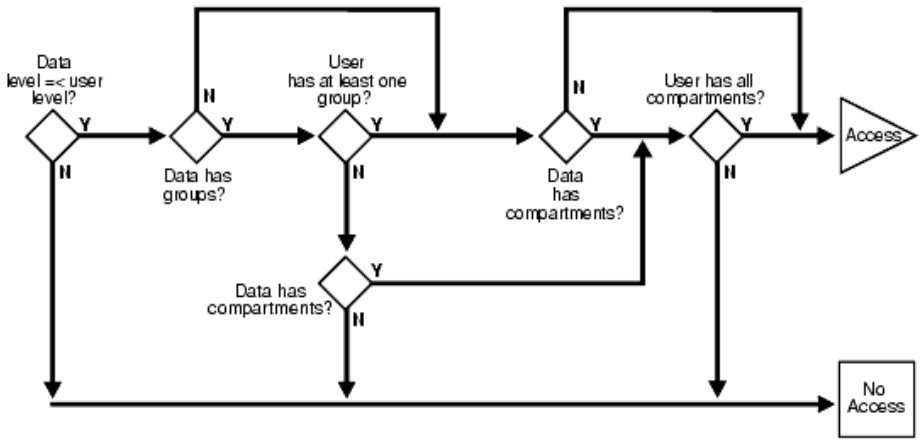
5. Giải thuật bảo mật của OLS đối với tác vụ viết



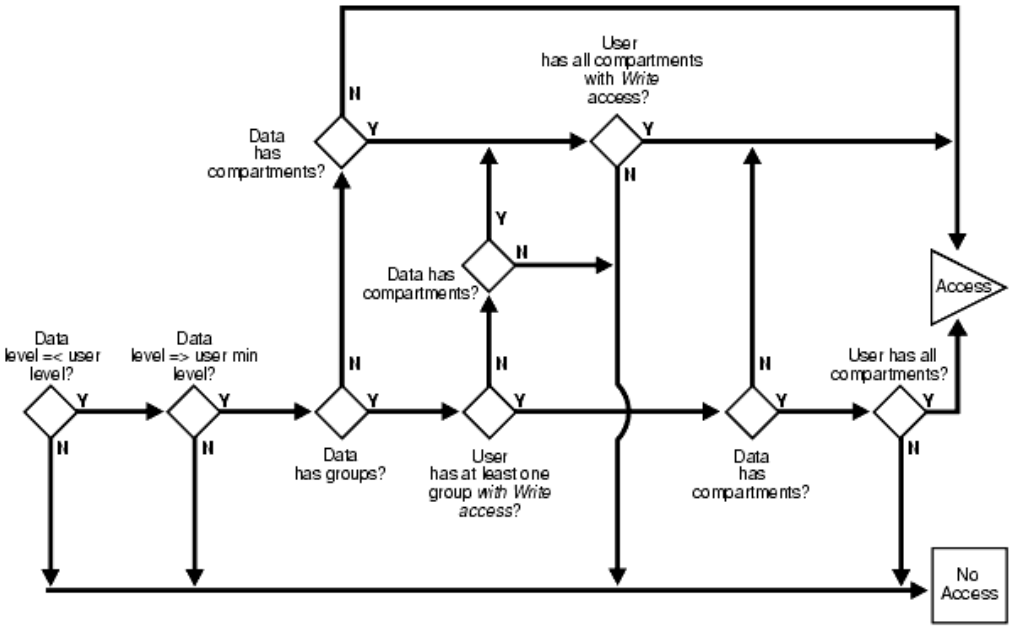
- Hình trên mô tả một cách rõ ràng cách thức mà OLS so sánh nhãn dữ liệu và nhãn người dùng để quyết định xem người dùng có quyền viết dòng dữ liệu đó hay không.
- Trong OLS, tác vụ viết tương đương với một trong các lệnh: UPDATE, INSERT, DELETE.
- Nói một cách ngắn gọn, người dùng chỉ có thể viết được dữ liệu khi đồng thời thỏa 2 điều kiện sau:
  - ✓ Điều kiện về level: Level của data label phải thấp hơn hoặc bằng level của session label hiện tại và cao hơn hoặc bằng min_level của người dùng.
  - ✓ Điều kiện về group và compartment: phải thỏa một trong 2 điều kiện sau:
    - Nếu data label không có group: session label của người dùng phải có quyền viết đối với tất cả các compartment mà data label đó có.
    - Nếu data label có chứa group: session label phải có quyền viết trên ít nhất một group trong data label hoặc có quyền viết trên group cha của ít nhất một group trong data label. Bên cạnh đó, session label cũng phải chứa tất cả các compartment xuất hiện trong data label (tức là có quyền đọc trên tất cả các compartment của data label, còn quyền write có hoặc không có cũng được).

6. Các quyền đặc biệt trong OLS

- Vì một số lý do đặc biệt, một người dùng có thể được cấp những quyền đặc biệt trong OLS để thực hiện một số tác vụ chuyên biệt hoặc truy xuất đến dữ liệu nằm ngoài giới hạn truy xuất được quy định trong tập xác thực quyền của người dùng đó.
- Các quyền đặc biệt được OLS định nghĩa gồm có 2 nhóm: quyền truy xuất đặc biệt (Special Access Privilege), quyền đặc biệt trên row label (Special Row Label Privilege).
- *Quyền truy xuất đặc biệt*:
  - ✓ *READ:* cho phép người dùng có quyền xem (SELECT) tất cả các dữ liệu do chính sách này bảo vệ, ngay cả khi người này không được gán bất cứ tập xác thực quyền nào.
  - ✓ *FULL:* cho phép người dùng có quyền viết và xem tất cả các dữ liệu do chính sách này bảo vệ.
  - ✓ *COMPACCESS:* quyền COMPACCESS cho phép người dùng truy xuất dữ liệu dựa trên các compartment của nhãn dữ liệu, không quan tâm đến các group mà nhãn dữ liệu đó đang chứa. Nếu nhãn dữ liệu đó không chứa compartment, việc truy xuất được xác định dựa trên các group như bình thường. Nếu dữ liệu đó có chứa các compartment và người dùng có quyền truy xuất (đọc/viết) đến chúng thì việc xác thực các group sẽ được bỏ qua. Hai hình bên dưới lần lượt minh họa cho quy trình xác thực tác vụ đọc và tác vụ viết đối với người dùng có quyền COMPACCESS.



Quy trình xác thực tác vụ đọc



Quy trình xác thực tác vụ viết

- ✓ *PROFILE_ACCESS:* cho phép thay đổi các session label của bản thân người dùng đó và session privilege của người dùng khác. Đây là một quyền rất "mạnh", vì người có quyền này có thể ngầm trở thành người có quyền FULL.

- *Quyền đặc biệt trên row label*:
  - ✓ *WRITEUP:* cho phép người dùng nâng level của một hàng dữ liệu nhưng không làm thay đổi các compartment và group của nó. Người dùng chỉ được nâng tối đa đến max_level của chính họ.
  - ✓ *WRITEDOWN:* cho phép người dùng hạ level của một hàng dữ liệu nhưng không làm thay đổi các compartment và group của nó. Người dùng chỉ được phép hạ tối đa xuống đến min_level của họ, không được hạ thấp hơn mức này.
  - ✓ *WRITEACROSS:* cho phép người dùng thay đổi compartment và group của một hàng dữ liệu nhưng không thay đổi level của nó. Người dùng có thể thay đổi các compartment và group đó thành bất cứ compartment và group nào có định nghĩa trong chính sách.

B. Thực hành

(Ý nghĩa các tham số của các thủ tục trong phần thực hành đã được giải thích trong phần lý thuyết nên sẽ không nhắc lại trong phần này).

1. Gán quyền người dùng theo các thành phần của nhãn

- Louise Doran là nhân viên thuộc phòng Sales nên ta sẽ gán các level, compartment và group phù hợp với phòng ban và cấp bậc của Louise.
- Để gán level cho người dùng, ta dùng thủ tục SA_USER_ADMIN.SET_LEVELS.

```
CONN hr_sec/hrsec;
BEGIN
    sa_user_admin.set_levels
       (policy_name => 'ACCESS_LOCATIONS',
       user_name    => 'LDORAN',
       max_level    => 'CONF',
       min_level    => 'PUB',
       def_level    => 'CONF',
       row_level    => 'CONF');
END;
/
```

- Gán compartment cho người dùng, ta dùng thủ tục SA_USER_ADMIN.SET_COMPARTMENTS.

```
CONN hr_sec/hrsec;
BEGIN
    sa_user_admin.set_compartments
       (policy_name => 'ACCESS_LOCATIONS',
       user_name    => 'LDORAN',
       read_comps   => 'SM,HR',
       write_comps  => 'SM',
       def_comps    => 'SM',
       row_comps    => 'SM');
END;
/
```

- Gán group cho người dùng, ta dùng thủ tục SA_USER_ADMIN.SET_GROUPS.

```
CONN hr_sec/hrsec;
BEGIN
    sa_user_admin.set_groups
       (policy_name => 'ACCESS_LOCATIONS',
       user_name    => 'LDORAN',
       read_groups  => 'UK,CA',
       write_groups => 'UK',
       def_groups   => 'UK',
       row_groups   => 'UK');
END;
/
```

2. **Gán quyền người dùng theo các nhãn**

   - Karen Partner là trưởng phòng Sales. Ta sẽ gán các nhãn phù hợp với phòng ban và cấp bậc của Karen.

```
CONN hr_sec/hrsec;
BEGIN
    sa_user_admin.set_user_labels
       (policy_name     => 'ACCESS_LOCATIONS',
       user_name        => 'KPARTNER',
       max_read_label   => 'SENS:SM,HR:UK,CA',
       max_write_label  => 'SENS:SM:UK',
       min_write_label  => 'CONF',
       def_label        => 'SENS:SM,HR:UK',
       row_label        => 'SENS:SM:UK');
END;
/
```

3. **Gán các quyền đặc biệt**

   - Steven King là tổng giám đốc có toàn quyền trên cơ sở dữ liệu, nên ta cấp quyền FULL cho người dùng này.

```
CONN hr_sec/hrsec;
BEGIN
    sa_user_admin.set_user_privs
       (policy_name   => 'ACCESS_LOCATIONS',
       user_name      => 'SKING',
       PRIVILEGES     => 'FULL');
END;
/
```

- Neena Kochhar là giám đốc điều hành nên ta có thể cấp quyền READ để người này có thể xem toàn bộ dữ liệu.

```
CONN hr_sec/hrsec;
BEGIN
    sa_user_admin.set_user_privs
       (policy_name   => 'ACCESS_LOCATIONS',
       user_name      => 'NKOCHHAR',
       PRIVILEGES     => 'READ');
END;
/
```

- *Lưu ý:* tham số user_name trong các procedure vừa sử dụng ở phần ***B – Thực hành*** không nhất thiết phải là một user thật sự của hệ thống. Nó cũng có thể là role, tên của ứng dụng,…

II. Áp dụng chính sách OLS

A. Lý thuyết

1. Đối tượng được bảo vệ

- OLS cho phép ta gán các chính sách cho các đối tượng cần được bảo vệ theo 2 cấp độ: cấp schema và cấp bảng. Khi 1 bảng cần được bảo vệ bởi 1 chính sách nào đó, ta gán chính sách đó cho cụ thể bảng đó. Nếu muốn tất cả các bảng thuộc 1 schema đều được bảo vệ bởi 1 chính sách, ta gán chính sách đó cho schema đó.
- Lưu ý: Nếu 1 chính sách được gán cho 1 schema và đồng thời cũng được gán tường minh cho 1 bảng thuộc schema đó thì các tùy chọn, thao tác ở cấp độ bảng sẽ override các tùy chọn, thao tác ở cấp độ schema.

2. **Các thao tác quản trị việc gán chính sách cho table/schema**
   - Áp dụng chính sách (Apply): ta gán chính sách cho cụ thể một bảng/schema cần được bảo vệ.
   - Loại bỏ chính sách (Remove): loại bỏ sự bảo vệ của 1 chính sách khỏi bảng/schema. Lưu ý là khi loại bỏ như vậy, cột chứa nhãn của chính sách đó vẫn còn trong table, trừ khi ta xóa cột đó một cách tường minh.
   - Ta có thể Enable/Disable một chính sách đang được gán cho 1 schema/bảng nào đó trong một khoảng thời gian.
   - Để thay đổi những thiết lập tùy chọn của một chính sách đối với 1 bảng thì trước hết ta phải remove chính sách đó ra rồi sau đó apply trở lại với những thay đổi trong tùy chọn.
3. **Các tùy chọn cho việc áp dụng chính sách**
   - Các tùy chọn này cho phép ta quy định một số ràng buộc trong việc áp dụng các chính sách:
     - ✓ LABEL_DEFAULT: Sử dụng row label mặc định của người dùng hiện tại để làm nhãn cho hàng dữ liệu mới được insert vào trừ khi row label được chỉ định tường minh bởi người insert hoặc hàm gán nhãn.
     - ✓ LABEL_UPDATE: bình thường, một người dùng khi update dữ liệu có thể thay đổi nhãn dữ liệu kèm theo. Tuy nhiên, nếu tham số này được bật lên, một người muốn thay đổi nhãn dữ liệu thì người đó phải có ít nhất một trong các quyền sau: WRITEUP, WRITEDOWN, hoặc WRITEACROSS.
     - ✓ CHECK_CONTROL: nếu tùy chọn này được thiết lập, mỗi khi dữ liệu được update/insert và nhãn dữ liệu bị thay đổi/tạo mới, OLS sẽ kiểm tra xem nhãn dữ liệu mới có vượt quá giới hạn quyền của người update/insert hay không để tránh xảy ra tình trạng một người sau khi update/insert dữ liệu thì không thể truy xuất lại dữ liệu đó.
     - ✓ READ_CONTROL: Chỉ những hàng có xác nhận quyền (thoả 3 điều kiện trong phần I.A.4) mới có thể được truy xuất bởi các thao tác SELECT, UPDATE và DELETE.
     - ✓ WRITE_CONTROL: xác định khả năng INSERT, UPDATE và DELETE dữ liệu tại 1 hàng. Nếu tùy chọn này được kích hoạt, người dùng phải được xác

thực quyền đầy đủ trước khi thực hiện các lệnh INSERT, UPDATE, DELETE.

- ✓ INSERT_CONTROL: có tác dụng giống tùy chọn WRITE_CONTROL nhưng chỉ đối với loại câu lệnh INSERT.
- ✓ DELETE_CONTROL: có tác dụng giống tùy chọn WRITE_CONTROL nhưng chỉ đối với loại câu lệnh DELETE.
- ✓ UPDATE_CONTROL: có tác dụng giống tùy chọn WRITE_CONTROL nhưng chỉ đối với loại câu lệnh UPDATE.
- ✓ ALL_CONTROL: áp dụng mọi ràng buộc tùy chọn.
- ✓ NO_CONTROL: không áp dụng bất cứ ràng buộc nào của chính sách.

4. Gán nhãn cho dữ liệu

- Có 3 cách để một hàng dữ liệu được gán nhãn chính sách:
  - ✓ Gán tường minh nhãn cho từng dòng dữ liệu thông qua các lệnh INSERT (cho dữ liệu mới) và UPDATE (cho dữ liệu đang tồn tại).
  - ✓ Thiết lập tùy chọn LABEL_DEFAULT.
  - ✓ Viết một function dùng cho việc gán nhãn cho các hàng dữ liệu của 1 bảng tùy theo nội dung của dữ liệu. Function này sẽ tự động được gọi cho mọi lệnh INSERT và UPDATE và nó độc lập với việc xác nhận quyền của mọi user.
- Tuy nhiên trong phạm vi bài thực hành này chúng ta sẽ chỉ dùng cách 1 để gán nhãn dữ liệu cho chính sách.

B. Thực hành

1. Áp dụng chính sách cho bảng

- Để gán chính sách cho các bảng ta dùng thủ tục SA_POLICY_ADMIN.APPLY_TABLE_POLICY

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

- Cần nhớ một điều quan trọng là khi một bảng được bảo vệ bởi 1 chính sách, các hàng dữ liệu chưa được gán nhãn sẽ không thể được truy xuất. Do đó khi áp dụng một chính sách bảo vệ cho bảng có chứa sẵn dữ liệu, đầu tiên ta chọn tùy chọn 'NO_CONTROL' để chính sách tuy được gán cho bảng (cột OLS_COLUMN được thêm vào bảng) nhưng những ràng buộc của chính sách chưa áp dụng lên bảng.
- Ta đăng nhập vào tài khoản HR để xem sự thay đổi của bảng sau khi gán chính sách:

```
CONN hr/hr;
DESCRIBE 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)
OLS_COLUMN                   NUMBER(10)
```

2. Gán nhãn cho dữ liệu

- Để sec_admin có thể thiết lập nhãn cho các dòng dữ liệu ta cần gán quyền cho sec_admin:

```
CONN hr/hr;
GRANT select, insert, update ON locations TO sec_admin;
```

- Khi đã có đủ quyền ta gán nhãn cho các dòng dữ liệu. Đầu tiên ta gán nhãn CONF cho mọi dữ liệu trong bảng:

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

- Tiếp theo ta cập nhật các nhãn của các dòng dữ liệu về các nước Mỹ, Anh, Canada:

```
CONN sec_admin/secadmin;
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';
```

- Giả sử có một số địa chỉ là thông tin đặc biệt cần bảo mật, nên ta gán cho những dòng này nhãn có độ bảo mật cao hơn:

```
CONN sec_admin/secadmin;
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 ;
```

- Lưu ý trong các lệnh trên ta có dùng thủ tục CHAR_TO_LABEL. Do giá trị các nhãn được lưu trong bảng thực chất là tag number. Cho nên ta phải dùng hàm này để chuyển từ dạng chuỗi ngắn của nhãn thành dạng số của nó.

- Tới đây thì ta đã thực hiện xong 5 bước trong quy trình hiện thực OLS.
- Do ở trên chúng ta đã thiết lập tùy chọn 'NO_CONTROL' cho việc áp dụng chính sách nên ở đây chúng ta cần remove chính sách khỏi bảng rồi áp dụng lại chính sách với tùy chọn mới để chính sách có thể được kích hoạt bảo vệ cho bảng.

```
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    =>
                 'READ_CONTROL,WRITE_CONTROL,CHECK_CONTROL');
END;
/
```

III. Bài tập

1. Tạo bảng CUSTOMERS để áp dụng chính sách *region_policy* (đã tạo trong phần bài tập của Lab 06). Sau đó insert dữ liệu vào.

```
customers (
 id                  NUMBER(10) NOT NULL,
 cust_type           VARCHAR2(10),
 first_name          VARCHAR2(30),
 last_name           VARCHAR2(30),
 region              VARCHAR2(5),
 credit              NUMBER(10,2),
 CONSTRAINT customer_pk PRIMARY KEY (id));
```

Vùng giá trị của một số cột:

- cust_type : silver, gold, platinum

- region: north, west, east, south
- credit: SV cần nhập dữ liệu đủ cho 3 trường hợp tương ứng với 3 khoảng giá trị >2000, từ 500 đến 2000, < 500.

Tạo ra các user: sales_manager, sales_north, sales_west, sales_east, sales_south. Cấp quyền để các user này kết nối vào CSDL. Gán user label cho các user vừa tạo (SV tự xác định user label cho từng user sao cho hợp lý).