

BỘ MÔN HỆ THỐNG THÔNG TIN - KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐẠI HỌC QUỐC GIA TP.HCM

CHUYÊN ĐỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NÂNG CAO

TOPIC 6.3: FINE-GRAINED ACCESS CONTROL

Nhóm sinh viên thực hiện: NHÓM 16

Giáo viên phụ trách: ThS. Tuấn Nguyễn Hoài Đức

ThS. Tiết Gia Hồng

ThS. Hồ Thị Hoàng Vy

BÀI TẬP MÔN HỌC - CHUYÊN ĐỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NÂNG CAO
HỌC KỲ II – NĂM HỌC 2022-2023

MỤC LỤC

BẢNG THÔNG TIN CHI TIẾT NHÓM	3
BẢNG PHÂN CÔNG & ĐÁNH GIÁ HOÀN THÀNH CÔNG VIỆC	4
A. Giới thiệu	5
I. Fine-grained Access Control là gì?	5
II. Lợi ích của Fine-grained Access Control	5
B. VPD	6
I. Khái niệm về VPD.....	6
1. VPD là gì?	6
2. Áp dụng VPD khi nào?	6
3. Nguyên lí VPD hoạt động.....	7
4. Lợi ích khi áp dụng VPD	9
5. Những lưu ý khi sử dụng VPD	10
II. Column-level VPD và Column Masking	10
1. Column-level VPD.....	11
2. Column Masking.....	13
3. Sự khác biệt giữa Column-level và Column-Masking	15
III. Chính sách bảo mật của VPD	15
1. DBMS_RLS package	15
2. Áp dụng chính sách trên table, view hoặc synonym.....	17
3. Tổ chức các chính sách theo nhóm	23
4. Các chính sách mặc định (Default policy).....	29
IV. Cấu hình VPD cho fine_grained access control	30
1. Sử dụng DBMS_RLS Package và lệnh Create Context command	30
2. Sử dụng Oracle Policy Manager Interface	35
2.1 Tạo VPD Policy Groups	35
2.2 Hàm tạo chính sách (policy function)	40
2.3 Thêm chính sách (policy) vào nhóm chính sách (policy group)	43
V. Giới hạn phạm vi của VPD use (trong Sechemas)	51
C. Application Context	54
I. Đặc trưng của Application context và cách thiết lập.....	55
1. Đặc trưng của Application context	55
2. Thiết lập Application context.....	55

II. Session-based/Global application context.....	57
1. Session-based application context	57
1.1 Database session-based application context	57
1.2 Client session-base dapplication context	57
2. Global application context	58
III. Cách áp dụng Application Context cho Fine-grained Access Control.....	59
IV. Kiểm tra tác dụng policy: View V\$VPD_POLICY	65
D. Lưu ý của nhóm	66
E. Kết luận.....	67
TÀI LIỆU THAM KHẢO	68

BẢNG THÔNG TIN CHI TIẾT NHÓM

Mã nhóm:	16		
Tên nhóm:	NHÓM 16		
Số lượng:	5		
MSSV	Họ tên	Email	Điện thoại
20120429	Nguyễn Quốc Anh	20120429@student.hcmus.edu.vn	0395639633
20120431	Tô Trần Sơn Bá	20120431@student.hcmus.edu.vn	0822926156
20120439	Hoàng Văn Cầu	20120439@student.hcmus.edu.vn	0845718717
20120446	Nguyễn Đình Cường	20120446@student.hcmus.edu.vn	0338707803
20120477	Nguyễn Trung Hiếu	20120477@student.hcmus.edu.vn	0378813690

BẢNG PHÂN CÔNG & ĐÁNH GIÁ HOÀN THÀNH CÔNG VIỆC

Công việc thực hiện	Người thực hiện	Mức độ hoàn thành	Đánh giá của nhóm
Phân giới thiệu	Nguyễn Quốc Anh	100%	100%
Khái niệm về VPD	Hoàng Văn Cầu Nguyễn Trung Hiếu	100%	100%
Column-level VPD và Column Masking	Tô Trần Sơn Bá Nguyễn Đình Cường	100%	100%
Chính sách bảo mật của VPD	Nguyễn Quốc Anh Hoàng Văn Cầu Nguyễn Trung Hiếu	100%	100%
Cấu hình cho fine_grained access control	Tô Trần Sơn Bá Nguyễn Đình Cường Nguyễn Quốc Anh	100%	100%
Giới hạn phạm vi của VPD use (trong Schema)	Hoàng Văn Cầu Nguyễn Trung Hiếu	100%	100%
Đặc trưng của Application context và cách thiết lập	Hoàng Văn Cầu Tô Trần Sơn Bá	100%	100%
Session-based/Global application context	Nguyễn Quốc Anh Nguyễn Đình Cường Nguyễn Trung Hiếu	100%	100%
Cách áp dụng Application Context cho Fine-grained Access Control	Nguyễn Trung Hiếu Tô Trần Sơn Bá	100%	100%
Kiểm tra tác dụng policy: View V\$VPD_POLICY	Hoàng Văn Cầu	100%	100%
Phản lưu ý của nhóm	Nguyễn Đình Cường	100%	100%
Phản Kết luận	Nguyễn Quốc Anh	100%	100%

A. Giới thiệu

I. Fine-grained Access Control là gì?

Fine-grained Access Control (kiểm soát truy cập chi tiết) là phương pháp kiểm soát người dùng truy cập lượng dữ liệu nhất định, cho phép hoặc từ chối yêu cầu sử dụng dữ liệu. So với Generalized Data Access Control (kiểm soát truy cập dữ liệu tổng quát), còn được gọi là Coarse-grained Access Control (kiểm soát truy cập chi tiết thô), Fine-grained Access Control sử dụng các phương pháp đa dạng và phong phú hơn để cho phép truy cập.

Fine-grained Access Control trong Oracle cho phép áp dụng các quy tắc bảo mật chi tiết đến các đối tượng cơ sở dữ liệu như bảng, cột, thủ tục và hàm. Fine-grained Access Control có thể xác định các quyền truy cập cho từng người dùng hoặc nhóm người dùng cụ thể dựa trên các tiêu chí như giá trị cột, điều kiện WHERE, thời gian và các nguyên tắc khác. Điều này cho phép người dùng tạo ra các quyền truy cập tùy chỉnh và chi tiết hơn so với các quyền truy cập tổng quát được cung cấp bởi Oracle

Ví dụ: Một cá nhân có thể được cấp quyền truy cập để chỉnh sửa và thực hiện các thay đổi đối với một phần dữ liệu, trong khi một người khác có thể chỉ được cấp quyền truy cập để đọc dữ liệu mà không thực hiện bất kỳ thay đổi nào

II. Lợi ích của Fine-grained Access Control

Fine-grained Access Control đem lại nhiều yêu cầu thiết yếu:

- Nguyên tắc bảo mật của đặc quyền tối thiểu:

FGAC cung cấp mức kiểm soát truy cập cần thiết để thực hiện các chức năng công việc hợp pháp và không có gì khác. Cách làm này dẫn đến sự ổn định và bảo mật hệ thống tốt hơn

- Quyền riêng tư và bảo mật:

Rò rỉ dữ liệu và luật riêng tư đã khiến các tổ chức chịu áp lực ngày càng tăng trong việc bảo vệ dữ liệu của người tiêu dùng. Các công ty cũng phải đáp ứng tất cả các yêu cầu về quy định và tuân thủ khi xử lý thông tin nhạy cảm, chẳng hạn như hồ sơ chăm sóc sức khỏe hoặc dữ liệu tài chính. Mà FGAC giúp giảm nguy cơ lộ dữ liệu và cho phép các tổ chức tuân thủ các quy định của chính phủ hiệu quả hơn

Thông tin thêm: Theo Pew Research Center, 93% người Mỹ cảm thấy điều quan trọng là phải kiểm soát những người có thể xem và sử dụng thông tin cá nhân của họ.

- Điện toán đám mây và lưu trữ dữ liệu tập trung:

Các nguồn dữ liệu được sử dụng trong điện toán đám mây thường được lưu trữ cùng nhau. FGAC hỗ trợ việc ngăn tất cả dữ liệu nội bộ có sẵn cho mọi người trong tổ chức đảm bảo bảo mật và quyền riêng tư đầy đủ đồng thời cho phép các tổ chức mở rộng quy mô và nhận ra lợi ích của việc lưu trữ dữ liệu tập trung

- Sự chính xác và tính rõ ràng:

FGAC cung cấp các phương pháp thực hiện kiểm soát truy cập chính xác hơn, đặc biệt đối với dữ liệu nhạy cảm. Mỗi tài nguyên có thể được chỉ định chính sách thay vì dựa vào phân loại vai trò rộng rãi. Các chính sách cũng cho phép triển khai kiểm soát truy cập chính xác hơn, rõ ràng tỉ mỉ hơn.

- Hỗ trợ bên thứ ba và nhân viên truy cập từ xa:

Trong một vài trường hợp có yêu cầu truy cập dữ liệu từ bên thứ ba, FGAC hỗ trợ việc giới hạn phạm vi dữ liệu được chia sẻ sẽ giảm thiểu rủi ro bảo mật và quyền có thể bị thu hồi khi bên thứ ba không còn yêu cầu. Đồng thời, FGAC cũng cho phép đặt quyền một cách linh hoạt cho các vai trò nhân viên từ xa dựa trên thời gian trong ngày và địa điểm, cũng có thể đặt quyền truy cập cho các nội dung có độ nhạy cảm cao để chỉ cho phép truy cập tại chỗ

B. VPD

I. Khái niệm về VPD

1. VPD là gì?

VPD (viết tắt cho Virtual Private Database) là một cơ chế bảo mật mạnh mẽ và cũng là một trong những tính năng bảo mật phổ biến nhất trong cơ sở dữ liệu, một tính năng của Oracle Database 11g Enterprise Edition, đã được giới thiệu trong Oracle8i để cung cấp kiểm soát truy cập dữ liệu chi tiết.

Lưu ý: VPD không phải là giải pháp cơ sở dữ liệu bảo mật duy nhất có sẵn, và sự phổ biến của nó có thể thay đổi tùy thuộc vào nhu cầu cụ thể và sự ưa thích của các tổ chức khác nhau

2. Áp dụng VPD khi nào?

VPD được sử dụng khi các đặc quyền đối tượng tiêu chuẩn và các vai trò cơ sở dữ liệu liên quan không đủ để đáp ứng các yêu cầu bảo mật của ứng dụng mà không cần VPD. Chính sách VPD có thể đơn giản hoặc phức tạp tùy thuộc vào yêu cầu bảo mật của bạn. VPD có thể được sử dụng kết hợp với tính năng "Application context" để thực

thì các yêu cầu bảo mật cấp độ hàng và/hoặc cấp độ cột phức tạp đối với quyền riêng tư và tuân thủ quy định[*]

VPD cho phép xác định và áp dụng các quy tắc truy cập riêng biệt cho từng người dùng hoặc nhóm người dùng dựa trên các điều kiện cụ thể, và chỉ cho phép họ truy cập vào các phần của dữ liệu mà họ được ủy quyền.

Dưới đây là một số trường hợp nên sử dụng VPD:

- **Bảo mật dữ liệu:** Khi muốn đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập vào dữ liệu nhạy cảm, VPD là một giải pháp hữu ích. Bạn có thể thiết lập các quy tắc truy cập cụ thể để chỉ cho phép các người dùng nhất định xem hoặc chỉnh sửa các phần cụ thể của dữ liệu.
- **Tuân thủ quy định pháp lý:** Khi bạn hoạt động trong một lĩnh vực yêu cầu tuân thủ các quy định pháp lý nghiêm ngặt về quyền riêng tư và bảo mật dữ liệu, VPD có thể giúp bạn đáp ứng các yêu cầu đó. Bằng cách xác định các quy tắc truy cập chính xác, bạn có thể đảm bảo rằng chỉ có người dùng được ủy quyền mới có thể truy cập vào dữ liệu nhạy cảm.
- **Đa tài khoản hoặc đa khách hàng:** Khi bạn cung cấp dịch vụ cho nhiều tài khoản hoặc khách hàng, bạn có thể sử dụng VPD để tạo ra các lớp truy cập riêng biệt cho từng tài khoản hoặc khách hàng. Điều này giúp đảm bảo rằng mỗi tài khoản chỉ có thể truy cập vào dữ liệu của chính mình mà không thể xem dữ liệu của các tài khoản khác.
- **Kiểm soát quyền truy cập động:** VPD cho phép bạn xác định các quy tắc truy cập dựa trên các thông tin ngữ cảnh như thông tin người dùng, vị trí hoặc thời gian. Điều này giúp bạn tạo ra các quy tắc truy cập linh hoạt và đáp ứng các yêu cầu truy cập động.

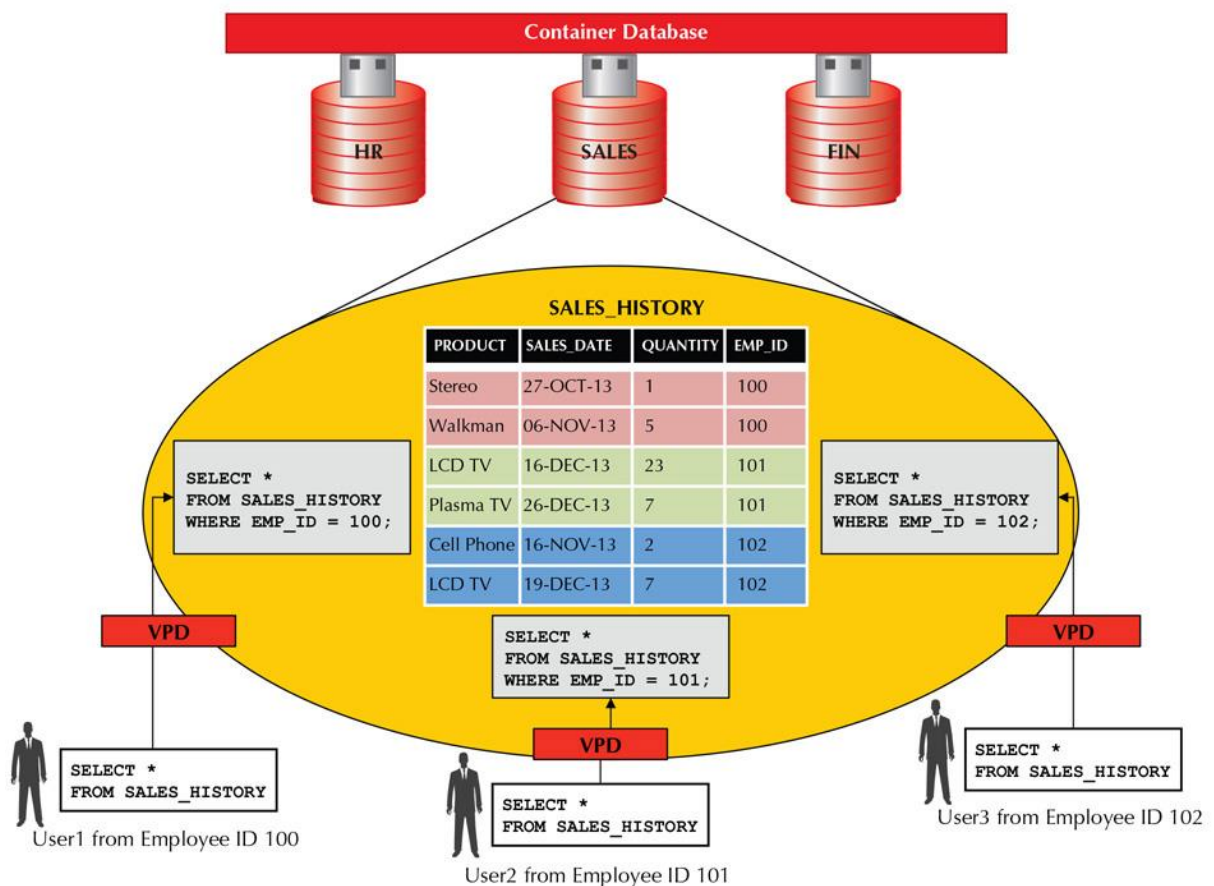
Từ đó, trước khi triển khai VPD, nên xem xét tổng quan về yêu cầu bảo mật, quyền truy cập dữ liệu và khả năng quản lý của hệ thống của bạn để đảm bảo rằng VPD là giải pháp phù hợp cho nhu cầu của bạn.

Lưu ý: Chính sách VPD có thể áp dụng trên các lệnh SQL như: SELECT, INSERT, UPDATE, INDEX và DELETE. VPD không hỗ trợ các ngôn ngữ định nghĩa dữ liệu như: TRUNCATE hay ALTER TABLE.

** Được trình bày ở phần sau*

3. Nguyên lý VPD hoạt động

Về bản chất, nếu các table (bảng), view (khung nhìn) hoặc synonym (tên thay thế cho các đối tượng như bảng, khung nhìn, các thủ tục được lưu và các đối tượng CSDL khác) trong Oracle áp dụng chính sách VPD để giữ bảo mật về dữ liệu, khi người dùng trực tiếp hoặc gián tiếp truy cập tới table, view hay synonym được bảo vệ bởi chính sách VPD này, thì Oracle sẽ tự động sửa lại câu truy vấn SQL của người dùng bằng cách thêm vào một mệnh đề WHERE (điều kiện WHERE, hay còn được gọi là vị từ, được trả về bởi một hàm thực hiện chính sách bảo mật và được thực thi trong suốt phiên truy cập đối với người dùng) Oracle sửa đổi câu lệnh một cách linh hoạt, minh bạch cho người dùng, sử dụng bất kỳ điều kiện nào có thể được biểu thị hoặc trả về bởi một hàm. Kết quả là mỗi người dùng có một khung nhìn khác nhau với dữ liệu ban đầu, tùy thuộc vào đặc quyền của họ.



4. Lợi ích khi áp dụng VPD

Việc đính kèm chính sách bảo mật VPD của Oracle vào các table, view hoặc synonym của cơ sở dữ liệu, thay vì triển khai các biện pháp kiểm soát truy cập trong tất cả các ứng dụng, sẽ mang lại các lợi ích sau:

- **Tính bảo mật:** Việc liên kết chính sách các table, view hoặc synonym trong cơ sở dữ liệu có thể giải quyết vấn đề bảo mật ứng dụng nghiêm trọng tiềm ẩn. Giả sử một người dùng được phép sử dụng một ứng dụng, sau đó dựa trên các đặc quyền được liên kết với ứng dụng đó, sửa đổi sai cơ sở dữ liệu bằng cách sử dụng một công cụ truy vấn đặc biệt, chẳng hạn như SQL Developer. Bằng cách đính kèm trực tiếp các chính sách bảo mật vào các table, view hoặc từ synonym, Fine-grained Access Control đảm bảo rằng cùng một biện pháp bảo mật có hiệu lực, bất kể người dùng truy cập dữ liệu bằng cách nào. Ngoài ra, dựa theo nguyên lý hoạt động của VPD, sẽ không có cách nào để bỏ qua bảo mật của VPD
- **Tính đơn giản:** Bạn chỉ thêm chính sách bảo mật các table, view hoặc synonym một lần thay vì thêm liên tục vào từng ứng dụng dựa trên các table, view hoặc từ synonym của mình.
- **Tính linh hoạt:** Bạn có thể có một chính sách bảo mật cho các câu lệnh SELECT, một chính sách khác cho các câu lệnh INSERT và các chính sách khác cho các câu lệnh UPDATE và DELETE.

Ví dụ: Bạn có thể muốn cho phép nhân viên Nhân sự có đặc quyền SELECT đối với tất cả hồ sơ nhân viên trong bộ phận của họ, nhưng chỉ cập nhật lương cho những nhân viên trong bộ phận của họ có họ bắt đầu bằng A đến F.

Ngoài ra, bạn có thể tạo nhiều chính sách cho mỗi table, view hoặc synonym và nó có thể được sửa đổi một cách dễ dàng mà không làm gián đoạn luồng điều khiển. Đồng thời, bạn có thể kiểm soát hiệu suất của các chức năng chính sách bằng cách định cấu hình các mà Oracle tổ chức lưu trữ các vị từ của VPD theo các cách:

- Đánh giá chính sách một lần cho mỗi truy vấn (chính sách tĩnh).
- Chỉ đánh giá chính sách khi ngữ cảnh ứng dụng trong chức năng chính sách thay đổi (chính sách nhạy cảm với ngữ cảnh).
- Đánh giá chính sách mỗi khi nó được chạy (chính sách động)
- **Tính truy cập cao:** Người dùng có thể dễ dàng truy cập dữ liệu từ bất cứ đâu, bất cứ khi nào
- **Tỷ lệ khôi phục cao:** Dữ liệu có thể được khôi phục rất dễ dàng (nhưng có hạn chế về tính thời gian)

- Tính bảo mật động: Không cần tạo các vai trò (roles) phức tạp.

5. Những lưu ý khi sử dụng VPD

Bên cạnh những lợi ích mà VPD mang lại, chúng ta vẫn còn một số vấn đề cần giải quyết khi áp dụng VPD cần được xem xét

- Tính bảo mật: VPD khó khăn trong việc bảo mật mức cột
- Hiệu năng: Khi sử dụng VPD, cần xem xét một số yếu tố như hiệu suất hệ thống và quản lý quy tắc truy cập. Nếu không thiết lập đúng các quy tắc VPD, có thể dẫn đến sự chậm trễ trong truy vấn và xử lý dữ liệu.
- Bảo trì: Khó kiểm tra

II. Column-level VPD và Column Masking

Column-level VPD và Column-masking là một trong những tính năng bảo mật của Oracle Database, cho phép kiểm soát quyền truy cập vào các cột hoặc thuộc tính cụ thể trong database dựa trên các điều kiện hoặc quy tắc cụ thể, thay vì gắn trên table, view hoặc synonym.

Column-level VPD và Column-masking đều áp dụng các quy tắc và chính sách dữ trên vai trò và quyền của người dùng đang truy cập.

Các chính sách của Column-level VPD và Column-masking chỉ được áp dụng khi có yêu cầu truy cập vào cột đã được áp cài đặt trong chính sách.

Ví dụ: User DEMO có một cơ sở dữ liệu gồm 1 bảng NHANVIEN và 5 user. Mỗi user chỉ có quyền xem thông tin của chính mình khi truy vấn vào bảng NHANVIEN. Ví dụ dưới đây sẽ có 2 chính sách về Column-level và Column masking sử dụng chung một policy function.

```
-- policy function
CREATE OR REPLACE FUNCTION DEMO_FUNC_POLICY(O_SCHEMA NVARCHAR2, O_NAME NVARCHAR2)
RETURN NVARCHAR2
AS
BEGIN
    RETURN 'MANV = SYS_CONTEXT(''USERENV'', ''SESSION_USER'')';
END;
```

- Trước khi áp dụng Column-level và Column masking:

The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the query 'SELECT * FROM DEMO.NHANVIEN'. Below the query, the 'Query Result' window shows the fetched data. The status bar indicates 'All Rows Fetched: 5 in 0.007 seconds'.

	MANV	TENNV	LUONG	VAITRO
1	NV03	David Johnson	7000.8	TRƯỞNG PHÒNG
2	NV04	Sarah Williams	5500.25	NHÂN VIÊN
3	NV05	Emily Brown	4500.9	NHÂN VIÊN
4	NV01	John Doe	5000.5	TRƯỞNG PHÒNG
5	NV02	Jane Smith	6000.75	NHÂN VIÊN

1. Column-level VPD

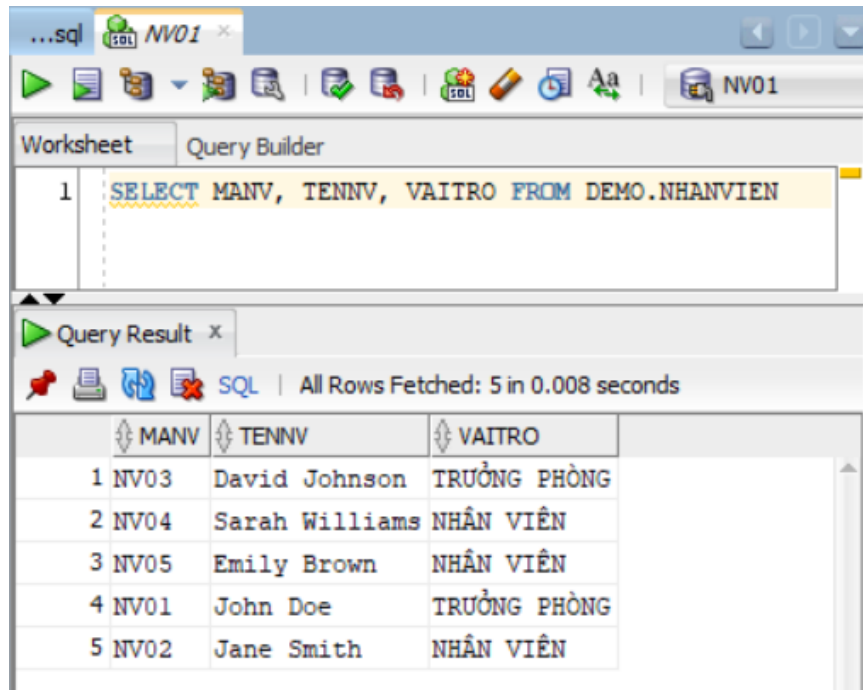
Trong Oracle, chính sách column-level thực thi bảo mật dữ liệu ở mức dòng khi có yêu cầu truy vấn tới những cột dữ liệu đã được bảo mật bởi những chính sách này. Kết quả trả về là các dòng dữ liệu thỏa chính sách.

- **Tiếp tục ví dụ trên:** Cài đặt chính sách Column-level ở cột LUONG như sau:

```
-- column level
BEGIN
  DBMS_RLS.ADD_POLICY(
    OBJECT_SCHEMA=> 'DEMO',
    OBJECT_NAME=> 'NHANVIEN',
    POLICY_NAME=> 'POLICY_COLUMN_LEVEL_DEMO',
    POLICY_FUNCTION=> 'DEMO_FUNC_POLICY',
    SEC_RELEVANT_COLS => 'LUONG'
  );
END;
```

*** Phần chi tiết DBMS_RLS.ADD_POLICY
được trình bày ở phần sau*

- Khi truy vấn tới bảng NHÂN VIÊN nhưng không truy vấn tới cột LUONG.



Worksheet Query Builder

```
1 SELECT MANV, TENNV, VAITRO FROM DEMO.NHANVIEN
```

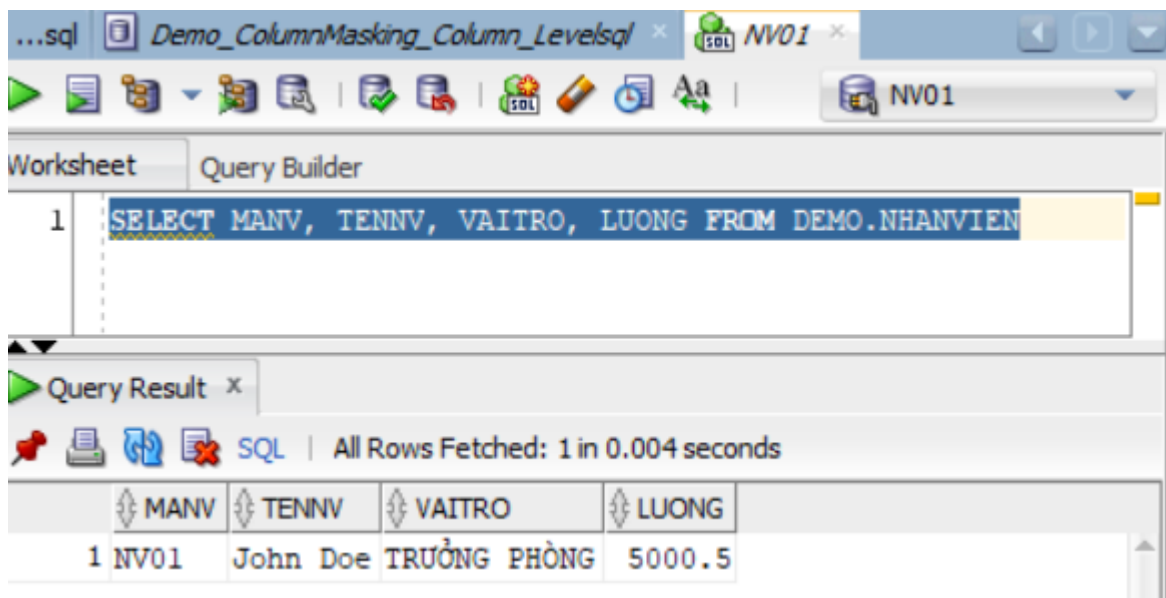
Query Result x

SQL | All Rows Fetched: 5 in 0.008 seconds

	MANV	TENNV	VAITRO
1	NV03	David Johnson	TRƯỞNG PHÒNG
2	NV04	Sarah Williams	NHÂN VIÊN
3	NV05	Emily Brown	NHÂN VIÊN
4	NV01	John Doe	TRƯỞNG PHÒNG
5	NV02	Jane Smith	NHÂN VIÊN

- Vì câu truy vấn không tham chiếu tới cột LUONG nên chính sách DEMO_FUNC_POLICY đã không được thực thi. Kết quả nhận được là bảng toàn bộ các dòng dữ liệu

- Khi truy vấn tới bảng NHÂN VIÊN có truy vấn trên cột LUONG.



Worksheet Query Builder

```
1 SELECT MANV, TENNV, VAITRO, LUONG FROM DEMO.NHANVIEN
```

Query Result x

SQL | All Rows Fetched: 1 in 0.004 seconds

	MANV	TENNV	VAITRO	LUONG
1	NV01	John Doe	TRƯỞNG PHÒNG	5000.5

- Lúc này câu truy vấn đã tham chiếu tới cột LUONG, khi đó chính sách column-level DEMO_FUNC_POLICY đã được thực thi từ đó dẫn đến việc giới hạn dữ liệu trả về thỏa yêu cầu của vị từ của policy function.

2. Column Masking

Khi có yêu cầu truy vấn tới những cột chứa dữ liệu nhạy cảm, thì column-level VPD sẽ hạn chế số lượng dòng dữ liệu được trả về. Vì vậy, Column-Masking được tạo ra để xử lý hạn chế đó.

Column-masking cho phép che giấu các giá trị của các cột trong một bảng dữ liệu dựa trên các quy tắc và chính sách được xác định.

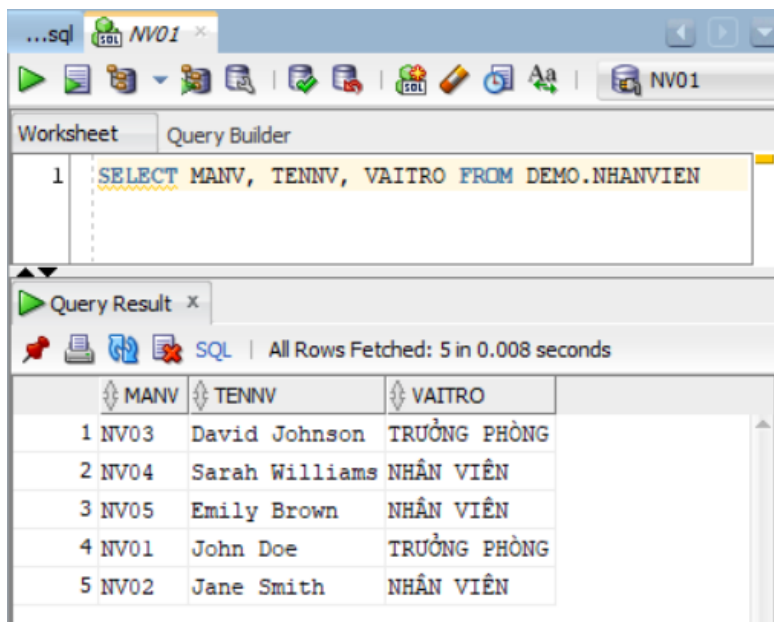
Mục tiêu: Column-Masking giữ cho dữ liệu nhạy cảm không hiển thị hoặc được che giấu với những người dùng không có quyền truy cập nhưng vẫn hiện đầy đủ record, thậm chí là các cột dữ liệu nhạy cảm. Tuy nhiên, các cột dữ liệu nhạy cảm sẽ hiển thị giá trị là NULL.

Để kích hoạt Column-masking, tham số SEC_RELEVANT_COLS_OPT phải được cài đặt trong thủ tục DBMS_RLS.ADD_POLICY.

Ví dụ: Ta áp dụng chính sách Column Masking ở cột LUONG

```
-- column MASKING
BEGIN
    DBMS_RLS.ADD_POLICY(
        OBJECT_SCHEMA=> 'DEMO',
        OBJECT_NAME=> 'NHANVIEN',
        POLICY_NAME=> 'POLICY_COLUMN_MASKING_DEMO',
        POLICY_FUNCTION=> 'DEMO_FUNC_POLICY',
        SEC_RELEVANT_COLS => 'LUONG',
        SEC_RELEVANT_COLS_OPT => DBMS_RLS.ALL_ROWS
    );
END;
```


- Khi truy vấn tới bảng NHANVIEN nhưng không truy vấn tới cột LUONG.



Worksheet Query Builder

```
1 SELECT MANV, TENNV, VAITRO FROM DEMO.NHANVIEN
```

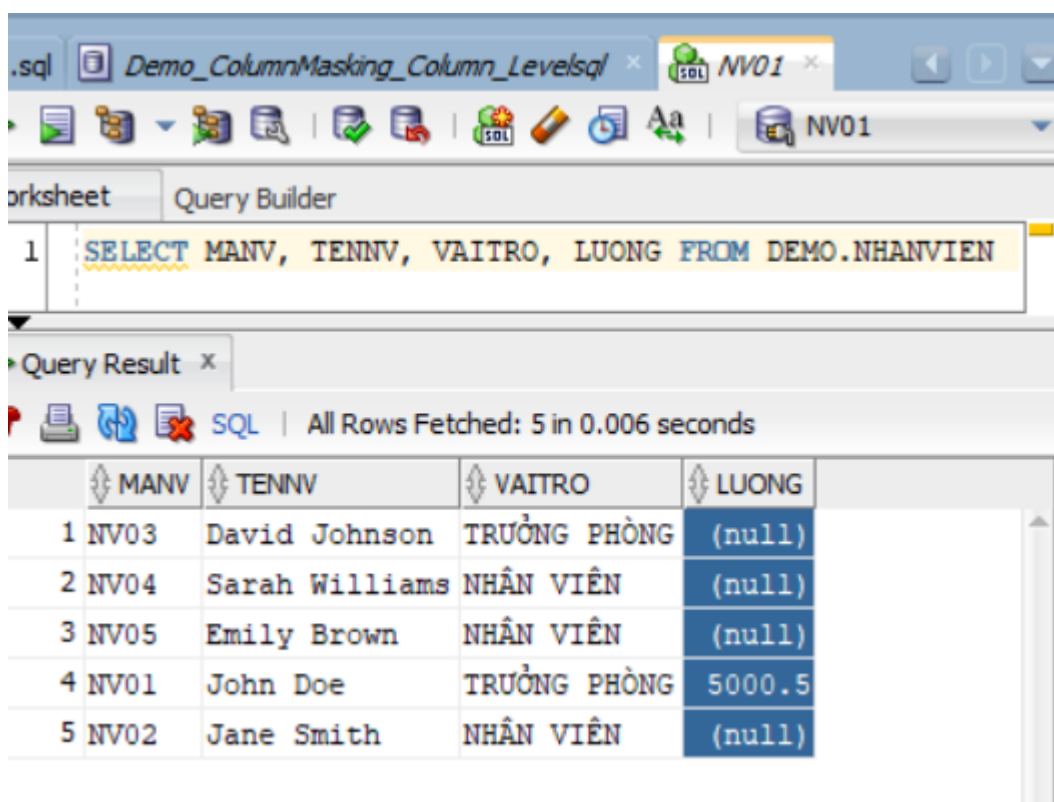
Query Result x

SQL | All Rows Fetched: 5 in 0.008 seconds

	MANV	TENNV	VAITRO
1	NV03	David Johnson	TRƯỞNG PHÒNG
2	NV04	Sarah Williams	NHÂN VIÊN
3	NV05	Emily Brown	NHÂN VIÊN
4	NV01	John Doe	TRƯỞNG PHÒNG
5	NV02	Jane Smith	NHÂN VIÊN

- Câu truy vấn không tham chiếu tới cột LUONG nên chính sách đã không được thực thi nên kết quả trả về như bình thường

- Khi truy vấn tới bảng NHANVIEN có truy vấn tới cột LUONG.



Worksheet Query Builder

```
1 SELECT MANV, TENNV, VAITRO, LUONG FROM DEMO.NHANVIEN
```

Query Result x

SQL | All Rows Fetched: 5 in 0.006 seconds

	MANV	TENNV	VAITRO	LUONG
1	NV03	David Johnson	TRƯỞNG PHÒNG	(null)
2	NV04	Sarah Williams	NHÂN VIÊN	(null)
3	NV05	Emily Brown	NHÂN VIÊN	(null)
4	NV01	John Doe	TRƯỞNG PHÒNG	5000.5
5	NV02	Jane Smith	NHÂN VIÊN	(null)

- Câu truy vấn đã tham chiếu tới cột LUONG, khi đó chính sách column-masking DEMO_FUNC_POLICY đã được thực thi và thay thế giá trị của cột LUONG thành

giá trị NULL trong các dòng dữ liệu không thỏa mãn điều kiện trả về của policy function.

3. Sự khác biệt giữa Column-level và Column-Masking

	Column-Level	Column Masking
Giống	<ul style="list-style-type: none">• Bảo vệ dữ liệu nhạy cảm khỏi người dùng không có quyền truy cập.• Cho phép quản lý quyền truy cập dữ liệu mức cột dựa trên các quy tắc và chính sách.• Có thể áp dụng cho các cột cụ thể trong bảng	
Khác	Cho phép thấy một phần hoặc không hoàn toàn nhìn thấy dòng dữ liệu khi truy vấn vào dữ liệu nhạy cảm	Cho phép nhìn thấy được toàn bộ dữ liệu truy vấn nhưng giá trị của các cột dữ liệu nhạy cảm sẽ được thay thế bằng giá trị NULL

Từ ví dụ ở phần trên, ta đã thấy được những điểm hỗ trợ chung và riêng giữa Column-level và Column-Masking. Vậy nên, chúng ta nên dựa vào quy tắc và chính sách bảo mật để lựa chọn sử dụng 2 tính năng trên phù hợp với ứng dụng.

III. Chính sách bảo mật của VPD

Trong trường hợp cần phải bảo mật dữ liệu với các ngữ cảnh khác nhau, có thể thực hiện yêu cầu đó bằng cách kiểm soát quyền và lượng dữ liệu khi xem, xóa, cập nhật, ... với các đối tượng người dùng khác nhau. Từ đó, có thể hạn chế tối đa nhất có thể việc mất mát thông tin. Các chính sách bảo mật VPD của Oracle có thể hỗ trợ thực hiện điều đó: Các chính sách bảo mật VPD liên kết các function VPD với table, view và synonym trong cơ sở dữ liệu.

1. DBMS_RLS package

Để tạo một chính sách, Oracle cung cấp cho người dùng DBMS_RLS package để hỗ trợ thực hiện công việc đó. DBMS_RLS package được sử dụng để triển khai các chính sách VPD. VPD cho phép bạn áp dụng kiểm soát truy cập chi tiết trên các table và view của cơ sở dữ liệu dựa trên các điều kiện cụ thể của người dùng, chẳng hạn như vai trò người dùng, ngữ cảnh ứng dụng hoặc bất kỳ tiêu chí nào khác.

DBMS_RLS package cung cấp một số thủ tục và hàm cho phép bạn xác định, quản lý và kích hoạt các chính sách VPD.

Thủ tục		Mô tả
Thủ tục xử lý policies riêng lẻ	DBMS_RLS.ADD_POLICY	Thêm một chính sách vào bảng, view, hoặc synonym
	DBMS_RLS.ENABLE_POLICY	Kích hoạt (vô hiệu hoá) một chính sách đã được thêm vào bảng, view, hoặc synonym trước đây.
	DBMS_RLS.ALTER_POLICY	Liên kết một ngữ cảnh ứng dụng với chính sách
	DBMS_RLS.REFRESH_POLICY	Phân tích lại các câu lệnh được lưu trong bộ nhớ cache liên quan đến chính sách được phân tích lại
	DBMS_RLS.DROP_POLICY	Xoá một chính sách khỏi table, view, hoặc synonym
Thủ tục xử lý policies group	DBMS_RLS.CREATE_POLICY_GROUP	Tạo một policy group
	DBMS_RLS.ALTER_GROUPED_POLICY	Thêm các thay đổi liên quan đến Application Context
	DBMS_RLS.DELETE_POLICY_GROUP	Xoá một policy group
	DBMS_RLS.ADD_GROUPED_POLICY	Thêm một policy vào một policy group được chỉ định
	DBMS_RLS.ENABLE_GROUPED_POLICY	Kích hoạt một policy trong một group
	DBMS_RLS.REFRESH_GROUPED_POLICY	Phân tích lại câu lệnh SQL liên kết với một group đã được làm mới
	DBMS_RLS.DISABLE_GROUPED_POLICY	Vô hiệu hoá một policy trong một group
	DBMS_RLS.DROP_GROUPED_POLICY	Xoá một policy khỏi một group được chỉ định

Thủ tục xử lý Application Contexts	DBMS_RLS.ADD_POLICY_CONTEXT	Thêm một context vào một ứng dụng đang hoạt động
	DBMS_RLS.DROP_POLICY_CONTEXT	Xoá một context khỏi một ứng dụng

2. Áp dụng chính sách trên table, view hoặc synonym

Để thêm một chính sách vào table, view hoặc synonym trong cơ sở dữ liệu thì package PL/SQL DBMS_RLS có thể hỗ trợ thực hiện công việc đó.

❖ Đầu tiên, phải tạo một policy function.

```
CREATE OR REPLACE FUNCTION function_name (
    Object_schema nvarchar2, Object_name nvarchar2)
RETURN NVARCHAR2
IS | AS
    [declaration_section]
BEGIN
    executable_section
[EXCEPTION
    exception_section]
END [function_name];
```

trong đó cần lưu ý:

Khi khởi tạo policy function nhất định phải truyền vào 2 tham số có giá trị (nvarchar2, nvarchar2) và trả về giá trị là NVARCHAR2

Ví dụ tạo một function policy như sau:

```
CREATE OR REPLACE FUNCTION FUNC_POLICY_NHANVIEN_1(O_SCHEMA VARCHAR2, O_NAME VARCHAR2)
RETURN VARCHAR2
AS
    V_ROLE INT;
    USERSESSION VARCHAR2(50);
BEGIN
    SELECT USER INTO USERSESSION FROM DUAL;
    IF (USERSESSION = 'HQT_CSDLNC') THEN
        RETURN '';
    ELSE
        SELECT COUNT(*) INTO V_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTEE = 'R_NHANVIEN';
        IF V_ROLE = 1 THEN
            RETURN 'MANV = sys_context(''userenv'', ''session_user'')';
        ELSE
            RETURN '';
        END IF;
    END IF;
END;
```

❖ Tiếp theo, sử dụng DBMS_RLS.ADD_POLICY để tạo 1 policy mới.

- Cú pháp tổng quát của DBMS_RLS.ADD_POLICY như sau:

```
DBMS_RLS.ADD_POLICY (
    object_schema          IN VARCHAR2          DEFAULT NULL,
    object_name            IN VARCHAR2,
    policy_name            IN VARCHAR2,
    function_schema        IN VARCHAR2          DEFAULT NULL,
    policy_function         IN VARCHAR2,
    statement_types        IN VARCHAR2          DEFAULT NULL,
    update_check           IN BOOLEAN           DEFAULT FALSE,
    enable                 IN BOOLEAN           DEFAULT TRUE,
    static_policy          IN BOOLEAN           DEFAULT FALSE,
    policy_type            IN BINARY_INTEGER    DEFAULT NULL,
    long_predicate         IN BOOLEAN           DEFAULT FALSE,
    sec_relevant_cols      IN VARCHAR2          DEFAULT NULL,
    sec_relevant_cols_opt  IN BINARY_INTEGER    DEFAULT NULL,
    namespace             IN VARCHAR2          DEFAULT NULL,
    attribute              IN VARCHAR2          DEFAULT NULL;
```

Trong đó:

Tham số	Mô tả
object_schema	Schema chứa table, view hoặc synonym. Lưu ý: Nếu không có object_schema nào được chỉ định hoặc là <i>NULL</i> , thì lược đồ hiện tại sẽ được sử dụng.
object_name	Tên của table, view hoặc synonym mà chính sách được thêm vào.
policy_name	Tên của chính sách sẽ được thêm vào. Lưu ý: <ul style="list-style-type: none"> - Nó phải là duy nhất cho cùng một table hoặc view. - Không nhập các ký tự đặc biệt như dấu cách hoặc dấu phẩy. Nếu bạn muốn sử dụng các ký tự đặc biệt cho tên chính sách, hãy đặt tên đó trong dấu ngoặc kép.
function_schema	Schema sở hữu function policy. Lưu ý: Nếu không có function_schema nào được chỉ định hoặc là <i>NULL</i> , thì lược đồ hiện tại sẽ được sử dụng.
policy_function	Tên của hàm tạo vị từ cho chính sách. Lưu ý: Nếu chức năng được định nghĩa trong một package, thì tên của package đó phải được chỉ định.

*** Phần tham số chính sách ADD_POLICY đã gần như đầy đủ nên không nhắc lại các tham số bị trùng ở phần dưới

statement_types	<p>Các loại câu lệnh mà chính sách áp dụng. Nó có thể là bất kỳ sự kết hợp nào của INDEX, SELECT, UPDATE hoặc DELETE.</p> <p>Lưu ý: Mặc định là áp dụng cho tất cả các loại này ngoại trừ INSERT và INDEX</p>
update_check	<p>Đổi số tùy chọn cho loại câu lệnh INSERT hoặc UPDATE.</p> <p>Lưu ý:</p> <ul style="list-style-type: none"> - Mặc định là <i>FALSE</i>. Giả sử bạn định sử dụng loại câu lệnh INSERT thì bạn phải đặt update_check thành <i>TRUE</i>. Nếu không, giá trị đầu vào ORA-28104 cho chuỗi không hợp lệ sẽ được tạo ra. - update_check chỉ áp dụng cho các cột liên quan đến bảo mật được bao gồm trong định nghĩa chính sách. Nói cách khác, thao tác INSERT hoặc UPDATE sẽ chỉ thất bại nếu cột liên quan đến bảo mật được định nghĩa trong chính sách được thêm hoặc cập nhật trong câu lệnh INSERT hoặc UPDATE
enable	<p>Cho biết liệu chính sách có được bật hay không khi nó được thêm vào.</p> <p>Lưu ý: Mặc định là đúng.</p>
static_policy	<p>Mặc định này FALSE. Nếu nó được đặt thành TRUE, thì máy chủ nhận định rằng chức năng chính sách cho chính sách tĩnh tạo ra cùng một chuỗi vị ngữ cho bất kỳ ai truy cập đối tượng, ngoại trừ SYS hoặc người dùng đặc quyền có đặc quyền CHÍNH SÁCH TRUY CẬP MIỄN PHÍ.</p>
policy_type	<p>Mặc định là NULL, có nghĩa là loại chính sách được quyết định bởi giá trị của chính sách tĩnh. Các loại chính sách có sẵn được liệt kê trong Bảng 138-5. Việc chỉ định bất kỳ loại chính sách nào trong số này sẽ ghi đè giá trị của static_policy</p>
long_predicate	<p>Hàm chính sách có thể trả về một vị từ có độ dài lên tới 4000 byte.</p> <p>Lưu ý:</p> <ul style="list-style-type: none"> - Mặc định là FALSE - Nếu mang giá trị TRUE có nghĩa là độ dài chuỗi văn bản vị ngữ có thể lên tới 32K byte. Các chính sách hiện có trước khi có sẵn tham số này giữ giới hạn 32K
sec_relevant_cols	<p>Bật VPD ở cấp độ cột, giúp thực thi các chính sách bảo mật khi một cột chứa thông tin nhạy cảm được tham chiếu trong</p>

	<p>một yêu cầu truy vấn. Chỉ định danh sách các tên cột hợp lệ được phân tách bằng dấu phẩy hoặc dấu cách của đối tượng được bảo vệ bằng chính sách.</p> <p>Lưu ý:</p> <ul style="list-style-type: none"> - Mặc định là tất cả các cột do người dùng xác định cho đối tượng. - Áp dụng được cho table và view, nhưng không áp dụng cho synonym - Chính sách này chỉ được thực thi nếu một cột cụ thể được tham chiếu (hoặc, đối với một cột kiểu dữ liệu trừu tượng, các thuộc tính của nó được tham chiếu) trong câu lệnh SQL của người dùng hoặc định nghĩa view cơ bản của nó.
sec_relevant_cols_opt	<p>Sử dụng với sec_relevant_cols để hiển thị tất cả các hàng cho các truy vấn được lọc VPD cấp cột (chỉ SELECT), nhưng ở đó các cột nhảy cảm xuất hiện dưới dạng NULL.</p> <p>Gợi ý: Đặt thành DBMS_RLS.ALL_ROWS để hiển thị tất cả các hàng, nhưng với các giá trị cột nhảy cảm, được lọc theo sec_relevant_cols, được hiển thị dưới dạng NULL.</p> <p>Lưu ý:</p> <ul style="list-style-type: none"> - Mặc định được đặt thành NULL, cho phép bộ lọc được xác định bằng sec_relevant_cols có hiệu lực
namespace	Tên xác định không gian tên bối cảnh ứng dụng
attribute	Thuộc tính xác định tên thuộc tính bối cảnh ứng dụng

Thông tin thêm: Thông thường người dùng không sử dụng hết tất cả tham số trong đó mà thông dụng một vài thông số trong đó thôi. Ví dụ như sau:

```

begin
    dbms_rls.add_policy(
        OBJECT_SCHEMA=> 'HQT_CSDLNC',
        OBJECT_NAME=> 'nhanvien',
        POLICY_NAME=> 'policy_nhanvien_1',
        FUNCTION_SCHEMA=> 'HQT_CSDLNC',
        POLICY_FUNCTION=> 'func_policy_nhanvien_1'
    );
end;
```

Lưu ý 1: Nếu trong câu lệnh có tham số namespace và attribute thì policy_type phải thuộc CONTEXT_SENSITIVE hoặc SHARE_CONTEXT_SENSITIVE

Lưu ý 2:

- Số lượng tối đa của các chính sách có thể tạo cho một đối tượng duy nhất là 255
 - Việc làm cho các đối tượng phụ thuộc trở thành không hợp lệ (bằng cách thêm một chính sách VPD vào đối tượng gốc, đối tượng được thêm chính sách từ ban đầu, của chúng) và khiến chúng cần phải được biên dịch lại có thể làm giảm hiệu suất trong hệ thống tổng thể. Oracle khuyến nghị chỉ nên thêm chính sách VPD vào một đối tượng có các đối tượng phụ thuộc trong thời gian không sử dụng hoặc trong thời gian chế độ ngừng hoạt động.
- ❖ Để kích hoạt hoặc vô hiệu hóa 1 chính sách, sử dụng procedure DBMS_RLS.ENABLE_POLICY.
- Cú pháp:

```
DBMS_RLS.ENABLE_POLICY (  
    object_schema IN VARCHAR2 NULL,  
    object_name   IN VARCHAR2,  
    policy_name   IN VARCHAR2,  
    enable        IN BOOLEAN TRUE);
```

- Ví dụ:

```
begin  
    DBMS_RLS.ENABLE_POLICY (  
        OBJECT_SCHEMA=> 'HQT_CSDLNC',  
        OBJECT_NAME=> 'NHANVIEN',  
        POLICY_NAME=> 'policy_truongphong_1',  
        enable      => TRUE);  
end;
```

- ❖ Để thêm một ngữ cảnh, dùng procedure DBMS_RLS.ALTER_POLICY.
- Cú pháp:

```
DBMS_RLS.ALTER_POLICY (  
    object_schema IN VARCHAR2 DEFAULT NULL,  
    object_name   IN VARCHAR2,
```

```

policy_name      IN VARCHAR2,
alter_option      IN NUMBER,
namespace         IN VARCHAR2 DEFAULT NULL,
attribute         IN VARCHAR2 DEFAULT NULL);

```

- Trong đó:

Tham số	Mô tả
alter_option	Được sử dụng để xác định ngữ cảnh ứng dụng được thêm hay xóa trong policy

- Ví dụ:

```

BEGIN
  DBMS_RLS.ALTER_POLICY(
    OBJECT_SCHEMA=> 'HQT_CSDLNC',
    OBJECT_NAME=> 'NHANVIEN',
    POLICY_NAME=> 'policy_truongphong_1',
    alter_option   => DBMS_RLS.ADD_ATTRIBUTE_ASSOCIATION,
    namespace      => 'hqtcsdl_user_ctx',
    attribute       => 'hqtcsdl_role');
END;

```

❖ Để phân tích lại các câu lệnh được lưu trong bộ nhớ cache liên quan đến policy, dùng procedure DBMS_RLS.REFRESH_POLICY.

- Cú pháp:

```

DBMS_RLS.REFRESH_POLICY (
  object_schema IN VARCHAR2 NULL,
  object_name   IN VARCHAR2 NULL,
  policy_name   IN VARCHAR2 NULL);

```

- Ví dụ:

```

begin
  dbms_rls.REFRESH_policy(
    OBJECT_SCHEMA=> 'HQT_CSDLNC',
    OBJECT_NAME=> 'nhanvien',
    POLICY_NAME=> 'policy_truongphong_1');

```

```
end;
```

❖ Để xóa một chính sách, dùng procedure DBMS_RLS.DROP_POLICY.

- Cú pháp:

```
DBMS_RLS.DROP_POLICY (  
    object_schema IN VARCHAR2 NULL,  
    object_name   IN VARCHAR2 NULL,  
    policy_name   IN VARCHAR2 NULL);
```

- Ví dụ:

```
begin  
    dbms_ols.DROP_policy(  
        OBJECT_SCHEMA=> 'HQT_CSDLNC',  
        OBJECT_NAME=> 'nhanvien',  
        POLICY_NAME=> 'policy_truongphong_1');  
end;
```

3. Tổ chức các chính sách theo nhóm

Nhóm chính sách là một tập hợp các chính sách bảo mật lại với nhau thuộc về một ứng dụng. Có thể chỉ định một application context (được biết đến như driving context hoặc policy context) để chỉ ra nhóm chính sách có hiệu lực. Sau đó, khi người dùng truy cập table, view hoặc synonym, Oracle sẽ tra cứu application context để xác định nhóm chính sách có hiệu lực. Nó thực thi tất cả các chính sách liên quan thuộc về nhóm chính sách.

Nhóm chính sách hữu ích cho các tình huống trong đó nhiều ứng dụng có nhiều chính sách bảo mật chia sẻ cùng một table, view hay synonym. Điều này cho phép xác định những chính sách sẽ có hiệu lực khi table, view hay synonym được truy cập.

❖ Để tạo một nhóm chính sách, dùng DBMS_RLS.CREATE_POLICY_GROUP.

Cú pháp:

```
DBMS_RLS.CREATE_POLICY_GROUP (  
    object_schema IN VARCHAR2 NULL,
```



```
object_name      IN VARCHAR2,  
policy_group     IN VARCHAR2);
```

- Trong đó:

Tham số	Mô tả
policy_group	Tên của nhóm chính sách

- Ví dụ:

```
BEGIN  
  DBMS_RLS.CREATE_POLICY_GROUP (  
    OBJECT_SCHEMA=> 'HQT_CSDLNC',  
    OBJECT_NAME=> 'NHANVIEN',  
    policy_group   => 'hqtcsqlnc_group');  
END; /
```

- ❖ Để xóa một nhóm chính sách, dùng dùng procedure DBMS_RLS.
DELETE_POLICY_GROUP

- Cú pháp:

```
DBMS_RLS.DELETE_POLICY_GROUP (  
  object_schema  IN VARCHAR2 NULL,  
  object_name    IN VARCHAR2,  
  policy_group   IN VARCHAR2);
```

- Ví dụ:

```
BEGIN  
  DBMS_RLS.DELETE_POLICY_GROUP (  
    OBJECT_SCHEMA=> 'HQT_CSDLNC',  
    OBJECT_NAME=> 'NHANVIEN',  
    policy_group   => 'hqtcsqlnc_group');  
END;
```

- ❖ Để thêm một chính sách vào group, dùng
DBMS_RLS.CREATE_GROUPED_POLICY

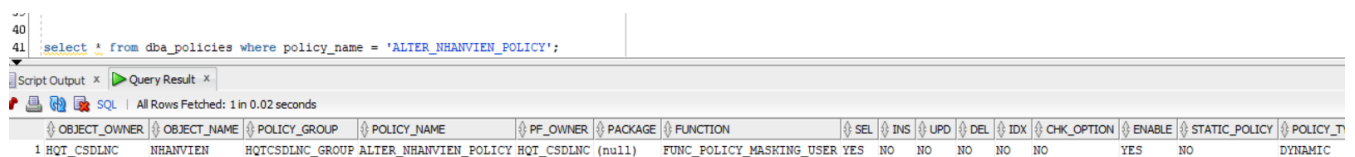
- **Cú pháp:**

```
DBMS_RLS.ADD_GROUPED_POLICY(
    object_schema          IN VARCHAR2          DEFAULT NULL,
    object_name            IN VARCHAR2,
    policy_group           IN VARCHAR2          DEFAULT 'SYS_DEFAULT',
    policy_name            IN VARCHAR2,
    function_schema        IN VARCHAR2          DEFAULT NULL,
    policy_function         IN VARCHAR2,
    statement_types        IN VARCHAR2          DEFAULT NULL,
    update_check           IN BOOLEAN           DEFAULT FALSE,
    enable                 IN BOOLEAN           DEFAULT TRUE,
    static_policy          IN BOOLEAN           DEFAULT FALSE,
    policy_type            IN BINARY_INTEGER    DEFAULT NULL,
    long_predicate         IN BOOLEAN           DEFAULT FALSE,
    sec_relevant_cols      IN VARCHAR2,
    sec_relevant_cols_opt  IN BINARY_INTEGER    DEFAULT NULL,
    namespace             IN VARCHAR2          DEFAULT NULL,
    attribute              IN VARCHAR2          DEFAULT NULL);
```

- **Ví dụ:**

```
BEGIN
    DBMS_RLS.ADD_GROUPED_POLICY(
        OBJECT_SCHEMA => 'HQT_CSDLNC',
        OBJECT_NAME   => 'NHANVIEN',
        policy_group   => 'hqtcsdlnc_group',
        policy_name     => 'alter_nhanvien_policy',
        policy_function => 'func_policy_masking_user',
        policy_type     => DBMS_RLS.CONTEXT_SENSITIVE,
        statement_types => 'select');
END; /
```

➤ **Kết quả:**



The screenshot shows a SQL Developer window with a query executed: `select * from dba_policies where policy_name = 'ALTER_NHANVIEN_POLICY';`. The results are displayed in a table with 14 columns. The first row of data corresponds to the policy created in the example.

	OBJECT_OWNER	OBJECT_NAME	POLICY_GROUP	POLICY_NAME	PF_OWNER	PACKAGE	FUNCTION	SEL	INS	UPD	DEL	IDX	CHK_OPTION	ENABLE	STATIC_POLICY	POLICY_TYPE
1	HQT_CSDLNC	NHANVIEN	HQTCSDLNC_GROUP	ALTER_NHANVIEN_POLICY	HQT_CSDLNC	(null)	FUNC_POLICY_MASKING_USER	YES	NO	NO	NO	NO	NO	YES	NO	DYNAMIC

- ❖ Để xóa một chính sách trong một nhóm chính sách, dùng procedure DBMS_RLS.
DROP_GROUPED_POLICY.

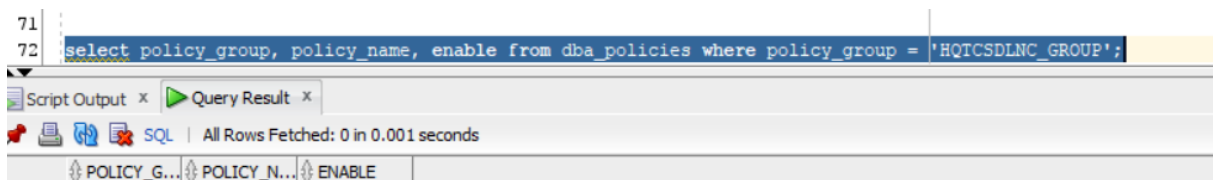
- Cú pháp:

```
DBMS_RLS.DROP_GROUPED_POLICY (  
    object_schema    IN VARCHAR2 NULL,  
    object_name      IN VARCHAR2,  
    policy_group     IN VARCHAR2 'SYS_DEFAULT',  
    policy_name      IN VARCHAR2);
```

- Ví dụ:

```
BEGIN  
    DBMS_RLS.DROP_GROUPED_POLICY(  
        OBJECT_SCHEMA => 'HQT_CSDLNC',  
        OBJECT_NAME   => 'NHANVIEN',  
        policy_group   => 'hqtcsdlnc_group',  
        policy_name    => 'alter_nhanvien_policy');  
END;
```

➤ Kết quả:



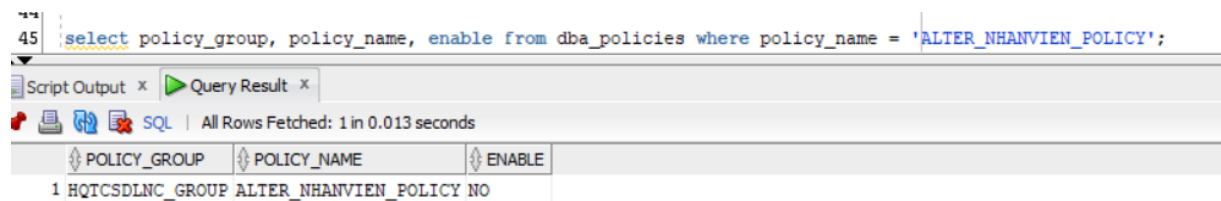
- ❖ Để vô hiệu hóa một chính sách trong nhóm chính sách, dùng procedure
DBMS_RLS.DISABLE_GROUPED_POLICY.

```
DBMS_RLS.DISABLE_GROUPED_POLICY (  
    object_schema    IN VARCHAR2 NULL,  
    object_name      IN VARCHAR2,  
    policy_group     IN VARCHAR2 'SYS_DEFAULT',  
    policy_name      IN VARCHAR2);
```

- Ví dụ:

```
BEGIN
  DBMS_RLS.DISABLE_GROUPED_POLICY (
    DBMS_RLS.DISABLE_GROUPED_POLICY (
      'HQT_CSDLNC',
      'NHANVIEN',
      'HQTCSLNC_GROUP',
      'ALTER_NHANVIEN_POLICY');
END;
```

➤ Kết quả



The screenshot shows a SQL query in the 'Script Output' pane: `select policy_group, policy_name, enable from dba_policies where policy_name = 'ALTER_NHANVIEN_POLICY';`. Below the query, the 'Query Result' pane displays a table with the following data:

POLICY_GROUP	POLICY_NAME	ENABLE
1 HQTCSLNC_GROUP	ALTER_NHANVIEN_POLICY	NO

- Để kích hoạt một chính sách trong nhóm chính sách, DBMS_RLS.
DBMS_RLS.ENABLE_GROUPED_POLICY.

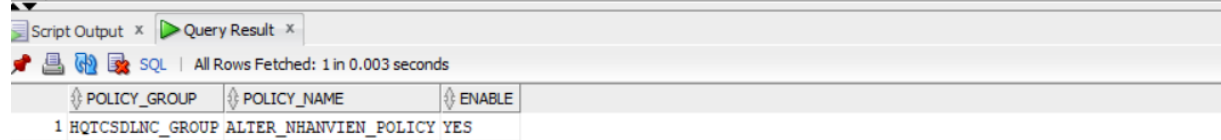
```
DBMS_RLS.ENABLE_GROUPED_POLICY (
  object_schema  IN VARCHAR2 NULL,
  object_name    IN VARCHAR2,
  group_name     IN VARCHAR2,
  policy_name    IN VARCHAR2,
  enable         IN BOOLEAN TRUE);
```

- Ví dụ:

```
BEGIN
  DBMS_RLS.ENABLE_GROUPED_POLICY (
    'HQT_CSDLNC',
    'NHANVIEN',
    'HQTCSLNC_GROUP',
    'ALTER_NHANVIEN_POLICY',
    TRUE);
END;
```

➤ Kết quả:

```
44 BEGIN
45 DBMS_RLS.ENABLE_GROUPED_POLICY ('HQT_CSDLNC','NHANVIEN', 'HQTCSLNC_GROUP','ALTER_NHANVIEN_POLICY', TRUE);
46 END;
47 select policy_group, policy_name, enable from dba_policies where policy_name = 'ALTER_NHANVIEN_POLICY';
```



The screenshot shows the SQL Developer interface. The top pane contains a PL/SQL block with lines 44 to 47. Line 45 calls `DBMS_RLS.ENABLE_GROUPED_POLICY` with parameters: `('HQT_CSDLNC','NHANVIEN', 'HQTCSLNC_GROUP','ALTER_NHANVIEN_POLICY', TRUE);`. Line 47 is a query: `select policy_group, policy_name, enable from dba_policies where policy_name = 'ALTER_NHANVIEN_POLICY';`. The bottom pane shows the 'Query Result' tab with a table containing one row: `HQTCSLNC_GROUP ALTER_NHANVIEN_POLICY YES`. The status bar indicates 'All Rows Fetched: 1 in 0.003 seconds'.

POLICY_GROUP	POLICY_NAME	ENABLE
HQTCSLNC_GROUP	ALTER_NHANVIEN_POLICY	YES

- Để phân tích lại các câu lệnh SQL liên kết với một chính sách được làm mới, dùng procedure `DBMS_RLS.REFRESH_GROUPED_POLICY`.

```
DBMS_RLS.REFRESH_GROUPED_POLICY (
    object_schema    IN VARCHAR2 NULL,
    object_name      IN VARCHAR2,
    group_name       IN VARCHAR2,
    policy_name      IN VARCHAR2);
```

- Ví dụ:

```
BEGIN
    DBMS_RLS.REFRESH_GROUPED_POLICY (
        'HQT_CSDLNC',
        'NHANVIEN',
        'HQTCSLNC_GROUP',
        'ALTER_NHANVIEN_POLICY');
END;
```

- Để chỉnh sửa ngữ cảnh liên quan đến chính sách trong nhóm chính sách, sử dụng `DBMS_RLS.ALTER_GROUPED_POLICY`.

```
DBMS_RLS.ALTER_GROUPED_POLICY (
    object_schema    IN VARCHAR2 DEFAULT NULL,
    object_name      IN VARCHAR2,
    policy_group     IN VARCHAR2 DEFAULT SYS_DEFAULT,
    policy_name      IN VARCHAR2,
    alter_option     IN NUMBER,
    namespace        IN VARCHAR2 DEFAULT NULL,
```

```
attribute          IN VARCHAR2 DEFAULT NULL);
```

- Ví dụ:

```
BEGIN
  DBMS_RLS.ALTER_GROUPED_POLICY (
    OBJECT_SCHEMA    => 'HQT_CSDLNC',
    OBJECT_NAME      => 'NHANVIEN',
    policy_group     => 'hqtcsdlnc_group',
    policy_name      => 'alter_nhanvien_policy',
    alter_option     => DBMS_RLS.ADD_ATTRIBUTE_ASSOCIATION,
    namespace       => 'hqtcsdl_user_ctx',
    attribute        => 'hqtcsdl_role');
END;
```

4. Các chính sách mặc định (Default policy)

Hàm SYS_CONTEXT: truy xuất thông tin về môi trường của oracle

o Cú pháp: Hàm trả về giá trị là chuỗi

```
SYS_CONTEXT
(
  namespace,
  parameter,
  [length]
)
```

Trong đó:

- *namespace*: Một Oracle namespace đã được tạo trước đó. Nếu Oracle namespace là 'USERENV' dùng để truy thông tin phiên bản của Oracle hiện tại.
- *length*: là độ dài của giá trị trả về tính bằng byte. Nếu tham số này bị bỏ qua, hay có nhập nhưng không hợp lệ thì mặc định là 256 byte.
- *parameter*: Tham số đã được định nghĩa sẵn; một số tham số hợp lệ của namespace là 'USERENV' (không phải tất cả các tham số đều hợp lệ trong tất cả phiên bản của Oracle)

IV. Cấu hình VPD cho fine_grained access control

1. Sử dụng DBMS_RLS Package và lệnh Create Context command

- Đầu tiên, ta có 1 user tên là HQT_CSDLNC có schema chứa 2 bảng là NHANVIEN và PHONGBAN. Có 2 Role là R_NHANVIEN và R_TRUONGPHONG và 5 user.
 - R_NHANVIEN có quyền truy cập vào thông tin của chính mình trong bảng nhân viên.
 - R_TRUONGPHONG có quyền xem thông tin của các nhân viên thuộc phòng ban mình quản lý nhưng không được cột LUONG.
- Sau đó, ta tạo và cấp quyền cho user HQT_CSDLNC:

```
create user HQT_CSDLNC identified by 123;  
grant create session to HQT_CSDLNC;  
grant dba to HQT_CSDLNC;
```

- Tiếp theo, ta đăng nhập vào oracle bằng user HQT_CSDLNC. Sau đó, tạo 2 bảng NHANVIEN, PHONGBAN và chèn dữ liệu. Ta được kết quả như sau.

MANV	TENNV	LUONG VAITRO	PHG
NV0001	John Doe	5000.5 TRƯỞNG PHÒNG	PB0001
NV0002	Jane Smith	6000.75 NHÂN VIÊN	PB0001
NV0003	David Johnson	7000.8 TRƯỞNG PHÒNG	PB0002
NV0004	Sarah Williams	5500.25 NHÂN VIÊN	PB0002
NV0005	Emily Brown	4500.9 NHÂN VIÊN	PB0001
MAPB	TENPB	TRPHG	
PB0001	Phòng Nhân Sự	NV0001	
PB0002	Phòng Kế Toán	NV0003	

- Ta tạo 2 Role R_NHANVIEN, R_TRUONGPHONG, 5 user và cấp quyền cho chúng.

```
create ROLE R_NHANVIEN;  
CREATE ROLE R_TRUONGPHONG;  
  
GRANT R_NHANVIEN TO NV0002;  
GRANT R_NHANVIEN TO NV0004;  
GRANT R_NHANVIEN TO NV0005;  
GRANT R_TRUONGPHONG TO NV0001;  
GRANT R_TRUONGPHONG TO NV0003;
```

```
GRANT SELECT ON PHONGBAN TO R_TRUONGPHONG;
GRANT SELECT ON NHANVIEN TO R_TRUONGPHONG;
GRANT SELECT ON NHANVIEN TO R_NHANVIEN;
```

- Kết quả thực hiện truy vấn trước khi cài đặt VPD.

	MANV	TENNV	LUONG	VAITRO	PHG
1	NV0001	John Doe	5000.5	TRƯỞNG PHÒNG	PB0001
2	NV0002	Jane Smith	6000.75	NHÂN VIÊN	PB0001
3	NV0003	David Johnson	7000.8	TRƯỞNG PHÒNG	PB0002
4	NV0004	Sarah Williams	5500.25	NHÂN VIÊN	PB0002
5	NV0005	Emily Brown	4500.9	NHÂN VIÊN	PB0001

- Bây giờ chúng ta bắt đầu làm chính sách thỏa khi truy vấn vào bảng NHANVIEN không phải là trưởng phòng thì chỉ thấy thông tin bản thân, nếu phải là trưởng phòng thì thấy thông tin nhân viên trong phòng. Chúng ta tạo 1 hàm chính sách trước khi tạo 1 chính sách.

```
create or replace function func_polic_truongphong_1(o_schema varchar2,
o_name varchar2)
return varchar2
as
    userSession varchar2(50);
begin
    userSession:= sys_context('userenv', 'session_user');
    if(userSession = 'HQT_CSDLNC') then
        return ' ';
```



```

ELSE
    RETURN 'MANV = sys_context(''userenv'', ''session_user'')
    OR
    PHG IN (SELECT MAPB FROM HQT_CSDLNC.PHONGBAN WHERE TRPHG =
sys_context(''userenv'', ''session_user''))';
end if;
end;

```

➤ Hàm chính sách trên đầu tiên kiểm tra session-user có phải là HQT_CSDLNC. Nếu phải thì trả về vị từ bằng rỗng.

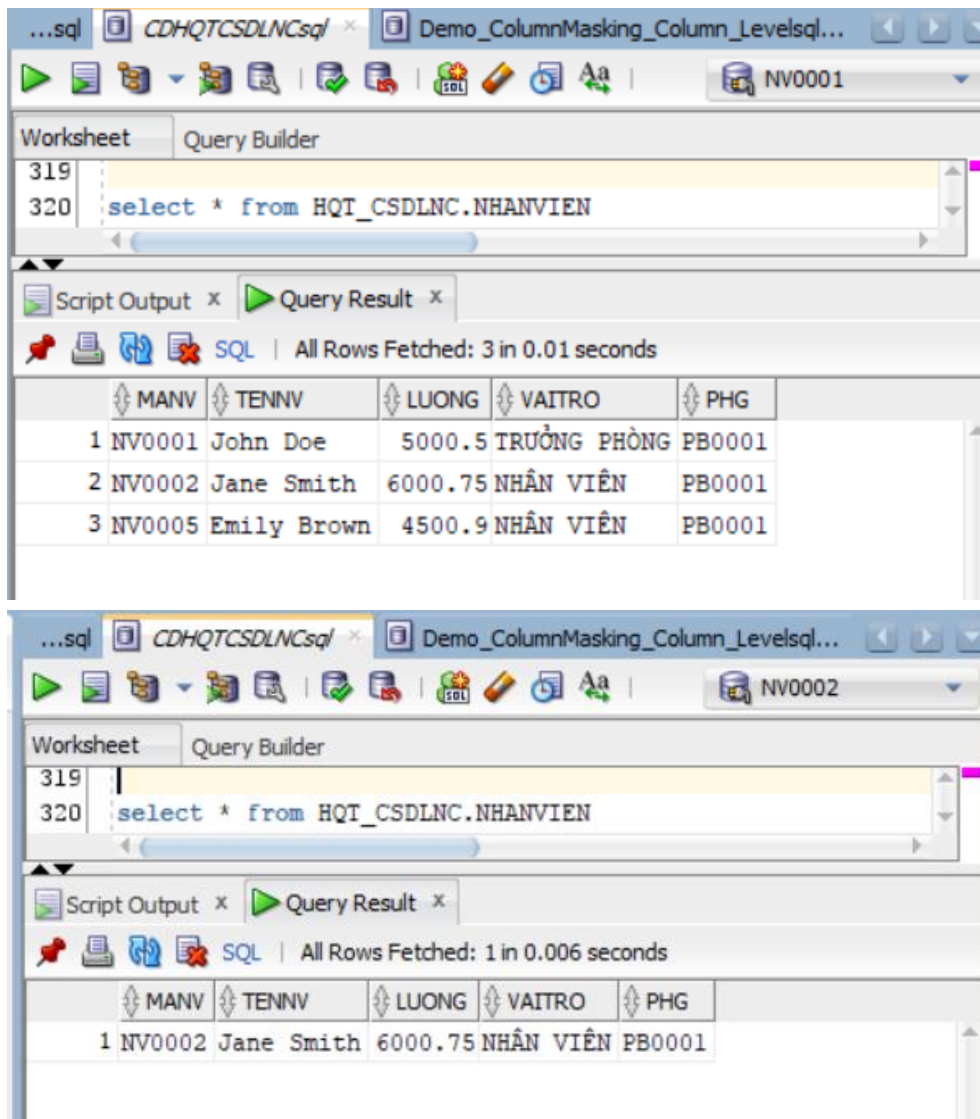
- Tiếp theo, ta tạo 1 chính sách áp dụng hàm chính sách trên.

```

begin
    dbms_ols.add_policy(
        OBJECT_SCHEMA=> 'HQT_CSDLNC',
        OBJECT_NAME=> 'NHANVIEN',
        POLICY_NAME=> 'policy_truongphong_1',
        POLICY_FUNCTION=> 'func_polic_truongphong_1'
    );
end;

```

- Đây là kết quả khi 2 user NV0001 (R_TRUONGPHONG) và NV0002 (R_NHANVIEN) khi truy vấn vào bảng NHANVIEN sau khi áp dụng chính sách.



- Để che giấu thông tin cột lương của nhân viên khỏi trưởng phòng. Ta viết một chính sách column masking để làm việc này.

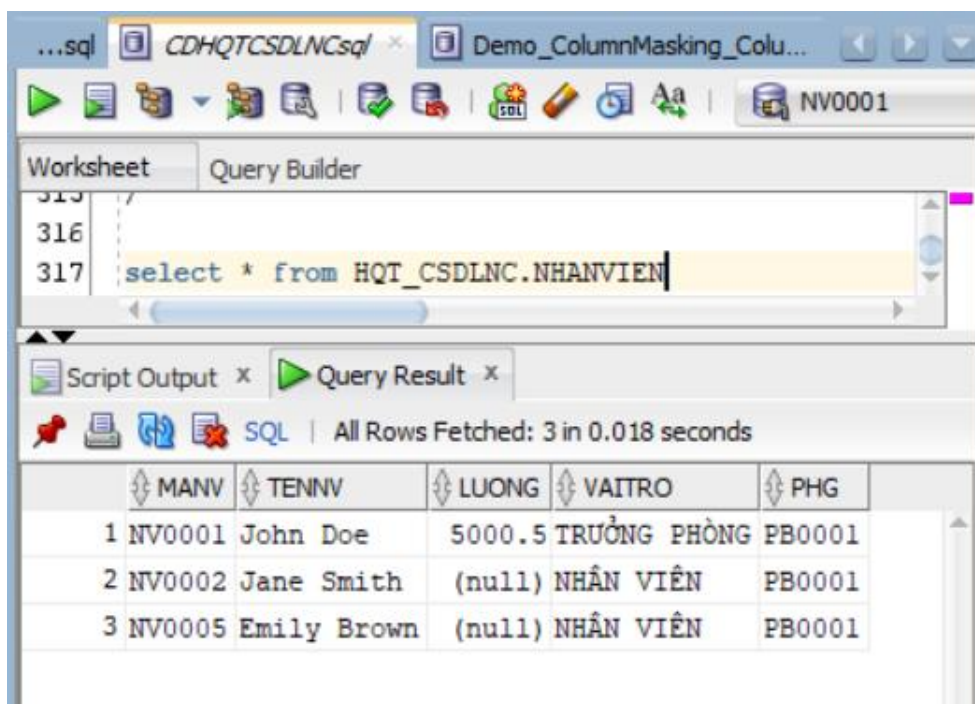
```
create or replace function func_policy_masking_user
    (o_schema nvarchar2, o_name nvarchar2)
return nvarchar2
as
begin
    userSession:= sys_context('userenv', 'session_user'); -- nv0002
    if(userSession = 'HQT_CSDLNC') then
        return '';
    ELSE
        RETURN 'MANV = sys_context(''userenv'', ''session_user'')';
    end if;
end;
```

```

begin
    DBMS_RLS.ADD_POLICY(
        OBJECT_SCHEMA=> 'HQT_CSDLNC',
        OBJECT_NAME=> 'NHANVIEN',
        POLICY_NAME=> 'policy_masking_user',
        POLICY_FUNCTION=> 'func_policy_masking_user',
        SEC_RELEVANT_COLS =>'LUONG',
        SEC_RELEVANT_COLS_OPT => DBMS_RLS.ALL_ROWS
    );
end;

```

➤ Kết quả đạt được là:



	MANV	TENNV	LUONG	VAITRO	PHG
1	NV0001	John Doe	5000.5	TRƯỞNG PHÒNG	PB0001
2	NV0002	Jane Smith	(null)	NHÂN VIÊN	PB0001
3	NV0005	Emily Brown	(null)	NHÂN VIÊN	PB0001

- Để xóa 2 chính sách, ta thực hiện lệnh DBMS_RLS.DROP_POLICY ở dưới đây:

```

begin
    dbms_ols.DROP_policy(
        OBJECT_SCHEMA=> 'HQT_CSDLNC',
        OBJECT_NAME=> 'nhanvien',
        POLICY_NAME=> 'policy_truongphong_1'
    );
end;
/
begin
    dbms_ols.DROP_policy(

```

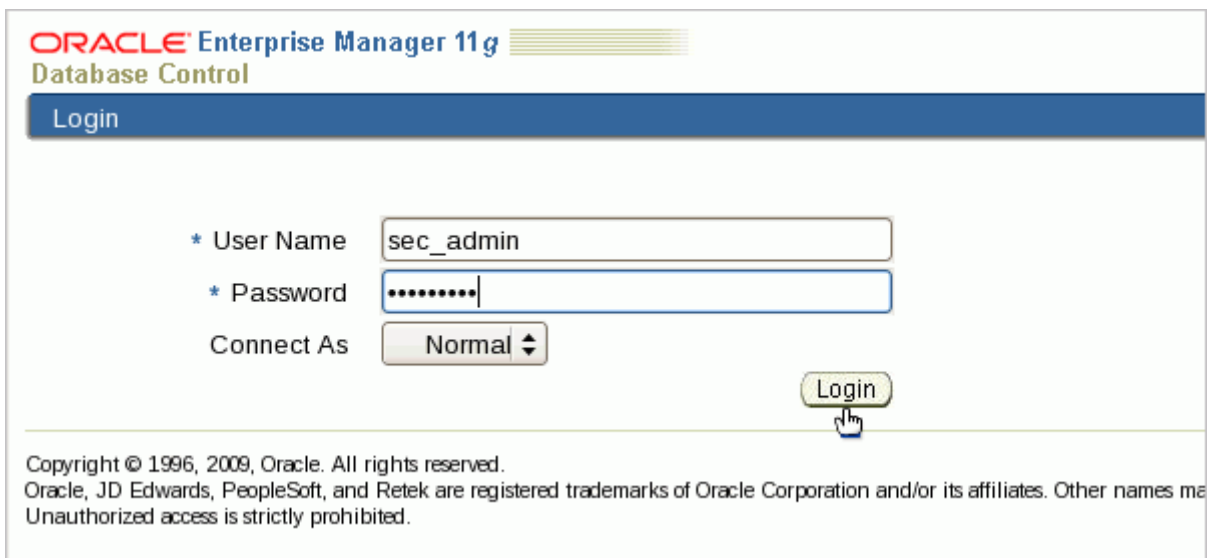
```
OBJECT_SCHEMA=> 'HQT_CSDLNC',  
OBJECT_NAME=> 'nhanvien',  
POLICY_NAME=> 'policy_masking_user'  
);  
end;
```

2. Sử dụng Oracle Policy Manager Interface

2.1 Tạo VPD Policy Groups

a) Tạo nhóm chính sách **PROVIDER_A_GROUP**

- Đăng nhập vào Enterprise Manager Database Control bằng tài khoản có quyền quản trị về bảo mật (**SEC_ADMIN**.)



ORACLE® Enterprise Manager 11g
Database Control

Login

* User Name

* Password

Connect As

Login

Copyright © 1996, 2009, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
Unauthorized access is strictly prohibited.


- Nhấn vào Server

Database Instance: orcl.example.com

Home Performance Availability **Server** Schema Data Movement Soft

Page Refreshed Sep 17, 2009 11:50:

General

 Shutdown Black Out

Status **Up**
 Up Since **Sep 2, 2009 4:17:31 PM EDT**
 Instance Name **orcl**
 Version **11.2.0.1.0**
 Host **host01.example.com**
 Listener **LISTENER_host01.example.com**

[View All Properties](#)

Host CPU

1.0
0.5
0.0

Loading...

Load **0.00** Paging **0.00**

- Trong Security của trang Server, chọn Virtual Private Database...

Logged in As SEC_ADMIN

chema Data Movement Software and Support

base Configuration

[ory Advisors](#)
[natic Undo Management](#)
[ization Parameters](#)
[Database Feature Usage](#)

Oracle Scheduler

[Jobs](#)
[Chains](#)
[Schedules](#)
[Programs](#)
[Job Classes](#)
[Windows](#)
[Window Groups](#)
[Global Attributes](#)
[Automated Maintenance Tasks](#)

Source Manager

[ing Started](#)
[umer Groups](#)
[umer Group Mappings](#)
[s](#)
[igs](#)
[tics](#)

Security

[Users](#)
[Roles](#)
[Profiles](#)
[Audit Settings](#)
[Transparent Data Encryption](#)
[Oracle Label Security](#)
[Virtual Private Database](#)
[Application Contexts](#)
[Database Vault](#)

- Nhấn vào Advanced.

Database Instance: [orcl.example.com](#) >

Virtual Private Database Policies

Virtual Private Database Policies enable to build applications that enforce row-level security policies to objects and dynamically appending predicates (WHERE clauses) to SQL statements.

Policy **Advanced**

Security Policies can be applied to Tables, Views or Synonyms (Synonyms to tables and views only Access Control (FGAC)).

Search

Specify an object name to list the policies associated with it. Optionally provide a policy name to

Schema Name

Object Name

Policy Name

- Nhấn vào Create để thêm VPD policy group.

Virtual Private Database Policies

Virtual Private Database Policies enable to build applications that enforce row-level security policies at the object security policies to objects and dynamically appending predicates (WHERE clauses) to SQL statements that query

Policy **Advanced**

A Policy Group is a set of security policies applied to a table, view or synonym that belong to an application. Apply indicate the policy group in effect for a user session and thus together implement Partitioned Fine Grained Access

Search

Specify an object name to list the policies associated with it. Optionally provide a policy name to filter the data

Schema Name

Object Name

Policy Group

Selection Mode

Select Policy Group	Schema	Object
No Policies Found		

- Nhập **provider_a_group** vào Policy Group Name.
- Nhập **APP_SERVER.RFID_DATA** vào Object Name.
- Nhấn OK.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.example.com > Virtual Private Database Policies >

Logged in As SEC_ADMIN

Create Policy Group

Select a object (table, view or synonym) and choose a unique group name for the selected object.

• Policy Group Name:

• Object Name:
Example: Schema Name Object Name

- Ta nhận được một tin nhắn xác nhận policy group đã được tạo

Database Instance: orcl.example.com >

Virtual Private Database Policies

Update Message

The object has been created successfully

Virtual Private Database Policies enable to build applications that enforce row-level security policies at the object security policies to objects and dynamically appending predicates (WHERE clauses) to SQL statements that query

Policy **Advanced**

A Policy Group is a set of security policies applied to a table, view or synonym that belong to an application. Appli indicate the policy group in effect for a user session and thus together implement Partitioned Fine Grained Access

Search

Specify an object name to list the policies associated with it. Optionally provide a policy name to filter the data

Schema Name:

Object Name:

Policy Group:

Selection Mode:

Select	Policy Group	Schema	Object
<input checked="" type="radio"/>	PROVIDER_A_GROUP	APP_SERVER	RFID_DATA

b) Tạo nhóm chính sách **PROVIDER_B_GROUP Policy**

- Trên trang Virtual Private Database (Advanced tab), chọn Create để tạo policy group khác.

Virtual Private Database Policies

Update Message
The object has been created successfully

Virtual Private Database Policies enable to build applications that enforce row-level security policies at the object security policies to objects and dynamically appending predicates (WHERE clauses) to SQL statements that query

Policy **Advanced**

A Policy Group is a set of security policies applied to a table, view or synonym that belong to an application. Appl indicate the policy group in effect for a user session and thus together implement Partitioned Fine Grained Access

Search
Specify an object name to list the policies associated with it. Optionally provide a policy name to filter the data

Schema Name
Object Name
Policy Group

Selection Mode

Select	Policy Group	Schema	Object
<input checked="" type="radio"/>	<u>PROVIDER_A_GROUP</u>	APP_SERVER	RFID_DATA

- Nhập **provider_b_group** vào Policy Group Name.
Nhập **APP_SERVER.RFID_DATA** vào Object Name. Nhấn OK.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.example.com > Virtual Private Database Policies > Logged in As SEC_ADMIN

Create Policy Group
Select a object (table, view or synonym) and choose a unique group name for the selected object

* Policy Group Name

* Object Name
Example: Schema Name.Object Name

- Ta nhận được một tin nhắn xác nhận policy group đã được. Chọn đường dẫn Database để trả về Database Home page.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.example.com > Help Logout Database

Logged in As SEC_ADMIN

Virtual Private Database Policies

Update Message
The object has been created successfully

Virtual Private Database Policies enable to build applications that enforce row-level security policies at the object (table, view or synonym) level by attaching security policies to objects and dynamically appending predicates (WHERE clauses) to SQL statements that query them.

Policy **Advanced**

A Policy Group is a set of security policies applied to a table, view or synonym that belong to an application. Application contexts (known as driving context) indicate the policy group in effect for a user session and thus together implement Partitioned Fine Grained Access (PFGAC).

Search
Specify an object name to list the policies associated with it. Optionally provide a policy name to filter the data that is displayed in the result.

Schema Name

Object Name

Policy Group

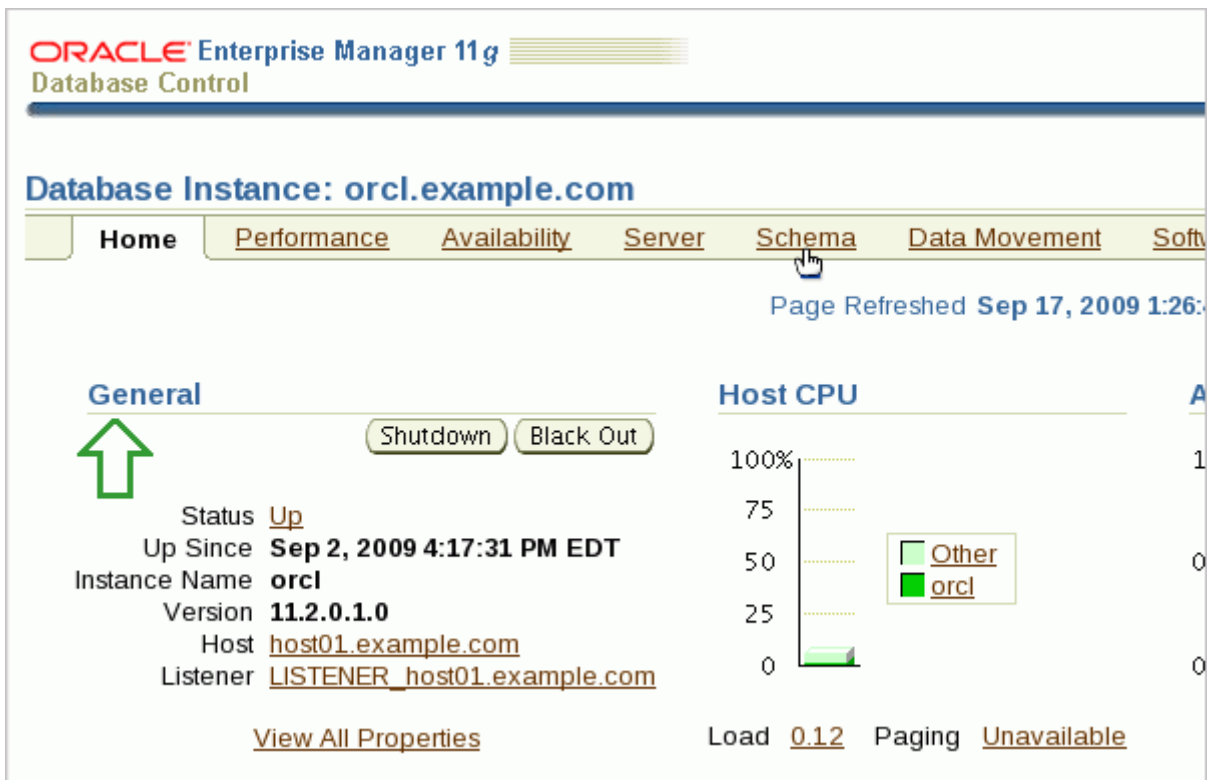
Selection Mode

Select	Policy Group	Schema	Object
<input checked="" type="radio"/>	PROVIDER_A_GROUP	APP_SERVER	RFID_DATA
<input type="radio"/>	PROVIDER_B_GROUP	APP_SERVER	RFID_DATA

2.2 Hàm tạo chính sách (policy function)

Để tạo hàm chính sách (policy function), thực hiện các bước:

- Trên trang Database Home, chọn Schema.



- Chọn Functions trong Programs.



- Nhấn Create.

Functions

Object Type: **Function**

Search
Enter a schema name and an object name to filter the data that is displayed in your results set.

Schema:

Object Name:

Status: **All**

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Select Schema	Function Name	Created	Last Modified	Status
No search conducted				

- Nhập **f_hide_cols** vào Name. Copy và paste theo đoạn code dưới đây vào Source . Nhấn OK.

```
(schema in varchar2, tab in varchar2)
return varchar2 as predicate varchar2(8) default '1=2';
begin
return predicate;
end;
```

ORACLE Enterprise Manager 11g

Database Control

Database Instance: [orcl.example.com](#) > [Functions](#) >

Logged in As **SEC_ADMIN**

Create Function

Name:

Schema:

Source:

```
(schema in varchar2, tab in varchar2)
return varchar2 as predicate varchar2(8) default '1=2';
begin
return predicate;
end;
```

- Ta nhận được thông báo xác nhận function đã được tạo. Chọn Database để trở về trang Database Home.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.example.com > Logged in As SEC_ADMIN

Confirmation
Function SEC_ADMIN.F_HIDE_COLS has been created successfully

Functions

Object Type: Function

Search
Enter a schema name and an object name to filter the data that is displayed in your results set.

Schema: SEC_ADMIN
Object Name:
Status: All
Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode: Single Create

Edit View Delete Actions Create Like Go

Select	Schema	Function Name	Created	Last Modified	Status
<input checked="" type="radio"/>	SEC_ADMIN	F_HIDE_COLS	Sep 17, 2009 1:40:02 PM EDT	Sep 17, 2009 1:40:02 PM EDT	VALID

2.3 Thêm chính sách (policy) vào nhóm chính sách (policy group)

1. Thêm chính sách vào PROVIDER_A_GROUP

- Chọn Server trên trang Database Home.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.example.com

Home Performance Availability **Server** Schema Data Movement Softw

Page Refreshed Sep 17, 2009 2:10:5

General

Shutdown Black Out

Status **Up**
Up Since **Sep 2, 2009 4:17:31 PM EDT**
Instance Name **orcl**
Version **11.2.0.1.0**
Host **host01.example.com**
Listener **LISTENER_host01.example.com**

[View All Properties](#)

Host CPU

100% 75 50 25 0

Other orcl

Load **0.32** Paging **0.00**

Active

1.0 0.5 0.0

- Chọn Virtual Private Database trong Security.

Logged in As SEC_ADMIN

[Schema](#) [Data Movement](#) [Software and Support](#)

Database Configuration

- [Diagnostic Advisors](#)
- [Automatic Undo Management](#)
- [Initialization Parameters](#)
- [Database Feature Usage](#)

Oracle Scheduler

- [Jobs](#)
- [Chains](#)
- [Schedules](#)
- [Programs](#)
- [Job Classes](#)
- [Windows](#)
- [Window Groups](#)
- [Global Attributes](#)
- [Automated Maintenance Tasks](#)

Source Manager

- [Logging Started](#)
- [Consumer Groups](#)
- [Consumer Group Mappings](#)
- [Streams](#)
- [Streams Pumps](#)
- [Stream Pumps](#)
- [Stream Pumps](#)

Security

- [Users](#)
- [Roles](#)
- [Profiles](#)
- [Audit Settings](#)
- [Transparent Data Encryption](#)
- [Virtual Private Database](#)
- [Application Contexts](#)
- [Database Vault](#)

- Chọn Advanced.

Virtual Private Database Policies

Virtual Private Database Policies enable to build applications that enforce row-level security policies to objects and dynamically appending predicates (WHERE clauses) to SQL statements.

Policy
Advanced

Security Policies can be applied to Tables, Views or Synonyms (Synonyms to tables and views only). Access Control (FGAC).

Search

Specify an object name to list the policies associated with it. Optionally provide a policy name to filter the results.

Schema Name

- Chọn **PROVIDER_A_GROUP** và chọn Manage.

Selection Mode
Single
Create

Create Like
View
Manage
Delete

Select	Policy Group	Schema	Object
<input checked="" type="radio"/>	PROVIDER_A_GROUP	APP_SERVER	RFID_DATA
<input type="radio"/>	PROVIDER_B_GROUP	APP_SERVER	RFID_DATA

- Trong phần Manage Group Policies, chọn Add.

Manage Policy Group: PROVIDER_A_GROUP

Select a object (table, view or synonym) and choose a unique group name for the selected object.

Policy Group Name **PROVIDER_A_GROUP**

Object Name **APP_SERVER.RFID_DATA**

Manage Group Policies

Choose to add or remove policies from a group below.

Select	Policy Name	Enabled
	No items found	

Add

- Nhập **hide_cols_b_from_a** vào Policy Name.
Chọn **SHARED_CONTEXT_SENSITIVE** trong Policy Type menu. Đảm bảo **Enabled** đã được chọn. Nhập **SEC_ADMIN.F_HIDE_COLS** trong Policy Function.

Add Policy

General

* Policy Name

Object Name **APP_SERVER.RFID_DATA**


Policy Group **PROVIDER_A_GROUP**

Policy Type **SHARED_CONTEXT_SENSITIVE**

☒ Enabled
Check this box to enable the policy after creation

Policy Function

Specify a policy function to return a predicate for filtering the data. The function can also reside in a package.

* Policy Function 

Example: Schema.Policy Function

☐ Long Predicate
Check this box to allow policy function to return a predicate with a length up to 32k. Default is 4k.

- Trong phần Enforcement, đảm bảo chỉ có **SELECT** được chọn (bỏ chọn INSERT, UPDATE, and DELETE). Chọn **Enable Column Masking Behavior**. Chọn Add.

Enforcement

Select operation types to which the policy applies. It can be any combination of SELECT, INSERT, UPDATE, and DELETE.

☐ INSERT
☐ UPDATE
☐ DELETE
☒ SELECT
☐ INDEX
☐ Insert/Update Check (CHECK OPTION)
Check this to allow changes to the row if they are still visible to the user after update. Can be specified only if INSERT and UPDATE are selected.

Security Relevant Columns

Specify security relevant columns if the policy being created is meant to apply to Column-Level Virtual Private Database (VPD).

☒ **Enable Column Masking Behavior**
Check this box to enable Column Masking Behavior of Column-Level Virtual Private Database.

Add

Remove

Select Name

No items Found

Overview

Column-level security policy controls access to information in the VPD application.

It can be used to:

- Default query that

- Chọn **STORAGE_B** và **DATE_B**. Chọn Select.

Database Instance: orcl.example.com > Logged in As SEC_ADMIN

Security Relevant Columns Cancel Select

Result

Select All | Select None

Select Name

<input type="checkbox"/>	STORAGE_A
<input type="checkbox"/>	DATE_A
<input checked="" type="checkbox"/>	STORAGE_B
<input checked="" type="checkbox"/>	DATE_B

- Trở về trang Add Policy. Chọn Continue.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.example.com > Logged In As SEC_ADMIN

Add Policy

Cancel Continue

General

Policy Name:

Object Name: APP_SERVER.RFID_DATA

Policy Group: PROVIDER_A_GROUP

Policy Type: SHARED_CONTEXT_SENSITIVE

☒ Enabled
Check this box to enable the policy after creation

Overview Of Policy Types

STATIC - For this type of policies the policy function is executed once and then the predicate is cached in the SGA (System Global Area) for fast performance. Applies to only one object.

SHARED_STATIC - Same as STATIC except that the server first looks for a cached predicate generated by the same policy function of the same policy type. Shared across multiple objects.

CONTEXT_SENSITIVE - For this type of policies, the server re-evaluates the policy function at statement execution time if it detects context changes. Server always executes the policy function.

- Trong Manage Policy Group: **PROVIDER_A_GROUP**, chọn OK.

Database Instance: orcl.example.com > Virtual Private Database Policies > Logged In As SEC_ADMIN

Manage Policy Group: PROVIDER_A_GROUP

Select a object (table, view or synonym) and choose a unique group name for the selected object.

Policy Group Name: PROVIDER_A_GROUP

Object Name: APP_SERVER.RFID_DATA

Manage Group Policies

Choose to add or remove policies from a group below.

Add

Select Policy Name	Enabled
<input checked="" type="radio"/> hide_cols_b_from_a	<input checked="" type="checkbox"/>

Delete

- Thông báo xác nhận nhóm chính sách đã được điều chỉnh thành công.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.example.com >

Virtual Private Database Policies

Update Message

POLICY GROUP PROVIDER_A_GROUP has been modified successfully

Virtual Private Database Policies enable to build applications that enforce row-level security policies on objects and dynamically appending predicates (WHERE clauses) to SQL statements that query them.

Policy Advanced

A Policy Group is a set of security policies applied to a table, view or synonym that belong to an application group in effect for a user session and thus together implement Partitioned Fine Grained Access (PFGA).

2. Thêm chính sách cho **PROVIDER_B_GROUP**

- Chọn **PROVIDER_B_GROUP** và chọn Manage.

Selection Mode Single

Create

Create Like View Manage Delete

Select	Policy Group	Schema	Object
<input type="radio"/>	PROVIDER_A_GROUP	APP_SERVER	RFID_DATA
<input checked="" type="radio"/>	PROVIDER_B_GROUP	APP_SERVER	RFID_DATA

- Trong Manage Group Policies, chọn Add.

Manage Policy Group: PROVIDER_B_GROUP

Select a object (table, view or synonym) and choose a unique group name for the selected object.

Policy Group Name **PROVIDER_B_GROUP**

Object Name **APP_SERVER.RFID_DATA**

Manage Group Policies

Choose to add or remove policies from a group below.

Add

Select	Policy Name	Enabled
	No items found	

- Nhập **hide_cols_a_from_b** vào Policy Name.
 Chọn **SHARED_CONTEXT_SENSITIVE** trong Policy Type. Đảm bảo **Enabled** đã được chọn. Nhập **SEC_ADMIN.F_HIDE_COLS** vào Policy Function.

Add Policy

General

* Policy Name

Object Name **APP_SERVER.RFID_DATA**


Policy Group **PROVIDER_B_GROUP**

Policy Type

☒ Enabled
Check this box to enable the policy after creation

Policy Function

Specify a policy function to return a predicate for filtering the data. The function can also reside in a package.

* Policy Function 

Example: Schema.Policy Function

☐ Long Predicate
Check this box to allow policy function to return a predicate with a length up to 32k. Default is 4k.

- Kéo xuống dưới trang. Vào Enforcement, đảm bảo chỉ **SELECT** được chọn (không chọn INSERT, UPDATE, và DELETE). Chọn **Enable Column Masking Behavior**. Chọn Add.

Enforcement

Select operation types to which the policy applies. It can be any combination of SELECT, INSERT, UPDATE, DELETE, and INDEX.

☐ INSERT
 ☐ UPDATE
 ☐ DELETE
 ☒ SELECT
 ☐ INDEX
 ☐ Insert/Update Check (CHECK OPTION)

Check this to allow changes to the row if they are still visible to the user after update. Can be specified only if INSERT, UPDATE, or DELETE is selected.

Security Relevant Columns

Specify security relevant columns if the policy being created is meant to apply to Column-Level Virtual Private Database (VPD).

☒ Enable Column Masking Behavior

Check this box to enable Column Masking Behavior of Column-Level Virtual Private Database.

Remove

Add

Select Name
No items Found

Overview

Column-level security policies control access to information in the database. VPD applies to queries and DML statements.

It can be used to:

- Default query the

- Chọn **STORAGE_A** và **DATE_A**. Chọn Select.

Database Instance: orcl.example.com >

Logged in As SEC_ADMIN

Security Relevant Columns

Cancel Select

Result

Select All | Select None

Select Name
<input checked="" type="checkbox"/> STORAGE_A
<input checked="" type="checkbox"/> DATE_A
<input type="checkbox"/> STORAGE_B
<input type="checkbox"/> DATE_B

- Ta trở về trang Add Policy. Chọn Continue.

Database Instance: orcl.example.com >

Logged in As SEC_ADMIN

Add Policy

Cancel Continue

General

Policy Name:

Object Name: **APP_SERVER.RFID_DATA**

Policy Group: **PROVIDER_B_GROUP**

Policy Type: **SHARED_CONTEXT_SENSITIVE**

☒ Enabled
Check this box to enable the policy after creation

Overview Of Policy Types

STATIC - For this type of policies the policy function is executed once and then the predicate is cached in the SGA (System Global Area) for fast performance. Applies to only one object.

SHARED_STATIC - Same as STATIC except that the server first looks for a cached predicate generated by the same policy function of the same policy type. Shared across multiple objects.

CONTEXT_SENSITIVE - For this type of policies, the server re-evaluates the policy function at statement execution time if it detects context changes. Server always executes the policy function on statement parsing and does not cache the value returned by the

- Trên Manage Policy Group: trang **PROVIDER_B_GROUP**, nhấn OK.

Database Instance: orcl.example.com > Virtual Private Database Policies > Logged in As SEC_ADMIN

Manage Policy Group: PROVIDER_B_GROUP

Select a object (table, view or synonym) and choose a unique group name for the selected object. Show SQL Cancel OK

Policy Group Name **PROVIDER_B_GROUP**
Object Name **APP_SERVER.RFID_DATA**

Manage Group Policies
Choose to add or remove policies from a group below.

[Delete](#) [Add](#)

Select Policy Name	Enabled
<input checked="" type="radio"/> hide_cots_a_from_b	✓

- Ta nhận được thông báo policy group đã được chỉnh sửa. Chọn Database để trở về trang Database Home.

ORACLE Enterprise Manager 11g Database Control Help Logout Database

Database Instance: orcl.example.com > Logged in As SEC_ADMIN

Virtual Private Database Policies

Update Message
POLICY GROUP PROVIDER_B_GROUP has been modified successfully

Virtual Private Database Policies enable to build applications that enforce row-level security policies at the object (table, view or synonym) level by attaching security policies to objects and dynamically appending predicates (WHERE clauses) to SQL statements that query them.

[Policy](#) [Advanced](#)

A Policy Group is a set of security policies applied to a table, view or synonym that belong to an application. Application contexts (known as driving context) indicate the policy group in effect for a user session and thus together implement Partitioned Fine Grained Access (PFGAC).

V. Giới hạn phạm vi của VPD use (trong Sechemas)

Phạm vi của VPD được giới hạn trong cơ sở dữ liệu hiện tại. Nếu bạn đang ở trong CDB root và tạo chính sách VPD, thì chỉ những đối tượng trong bộ chứa gốc mới bị ảnh hưởng. Nếu bạn đang ở trong PDB(Pluggable database) và tạo chính sách VPD, thì chỉ các đối tượng trong cơ sở dữ liệu có thể cấm mới bị ảnh hưởng. Bạn không thể tạo một chính sách duy nhất trong bộ chứa gốc có thể được thực thi trong tất cả các cơ sở dữ liệu có thể cấm được

Bạn không thể áp dụng chính sách VPD lên đối tượng trong SYS schema. Người dùng SYS và những người tạo được connect as SYSDBA thì chính sách VPD không tác động lên hành động của họ.

Lưu ý: Mặc dù vậy, bạn có thể Audit người dùng SYS để xem những hành động của họ thực hiện lên database và lưu trong một chỗ an toàn trong hệ điều hành.

```
Enter user-name: HQT_CSDLNC/123 as sysdba
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> GRANT DBA TO SCOPE_LIMIT;

Grant succeeded.

SQL> CONN SCOPE_LIMIT/123
Connected.
SQL> SELECT * FROM hqt_CSDLNC.NHANVIEN;

no rows selected

SQL> CONN SCOPE_LIMIT/123 AS SYSDBA
Connected.
SQL> SELECT * FROM hqt_CSDLNC.NHANVIEN;

MANV
-----
TENNV                                LUONG
-----
VAITRO
-----
PHG
-----
NV0001
John Doe                            5000.5
TRU NG PHENG
PB0001

MANV
-----
TENNV                                LUONG
-----
VAITRO
-----
PHG
-----
NV0002
Jane Smith                          6000.75
```

Ở ví dụ này, khi user SCOPE_LIMIT không đăng nhập dưới quyền SYSDBA thì khi SELECT trên bảng NHANVIEN thì không vượt qua được do chính sách VPD được kích hoạt nên không nhận được dòng dữ liệu nào trả về.

Mặt khác, khi đăng nhập dưới quyền SYSDBA thì hoàn toàn bỏ qua được phạm vi chính sách VPD từ đó đọc được tất cả dòng dữ liệu trên bảng NHANVIEN

Những người dùng được cấp quyền EXEMPT ACCESS POLICY (GRANT EXEMPT ACCESS POLICY TO) dù là qua trực tiếp cấp hay trong Role cấp có đều vượt qua được VPD. Quyền này thường dùng để cấp cho các người dùng quản trị chẳng hạn như cài đặt, đặt dữ liệu và xuất dữ liệu không phải thuộc sys schema.

```
SQL> CONN SCOPE_LIMIT/123
Connected.
SQL> SELECT * FROM hqt_CSDLNC.NHANVIEN;

no rows selected

SQL> CONN HQT_CSDLNC/123 as sysdba
Connected.
SQL> GRANT EXEMPT ACCESS POLICY TO SCOPE_LIMIT;

Grant succeeded.

SQL> CONN SCOPE_LIMIT/123
Connected.
SQL> SELECT * FROM hqt_CSDLNC.NHANVIEN;
```

MANV	TENNV	LUONG
VAITRO		
PHG		
NV0001	John Doe	5000.5
TRUONG PHENG		
PB0001		

MANV	TENNV	LUONG
VAITRO		
PHG		
NV0002	Jane Smith	6000.75
NHATN VINH		
PB0001		

Ở ví dụ này, khi user SCOPE_LIMIT không được cấp quyền EXEMPT ACCESS POLICY thì khi SELECT trên bảng NHANVIEN thì không vượt qua được do chính sách VPD được kích hoạt nên không nhận được dòng dữ liệu nào trả về.

Mặt khác, khi user SCOPE_LIMIT được cấp quyền EXEMPT ACCESS POLICY thì hoàn toàn bỏ qua được phạm vi chính sách VPD từ đó đọc được tất cả dòng dữ liệu trên bảng NHANVIEN

C. Application Context

Application Context là một tập các cặp tên – giá trị mà cơ sở dữ liệu Oracle lưu trong bộ nhớ. Nó được xác định, thiết lập và lấy ra bởi người dùng và các ứng dụng. Các giá trị liên quan được nhóm lại thành một nhóm được truy cập theo không gian tên miền (namespace) hay tên của nó. Bằng cách lưu trữ các giá trị và các thuộc tính trong bộ nhớ (UGA hoặc SGA), sau đó chia sẻ chúng dựa trên ngưỡng, sẽ giúp việc truy xuất các giá trị nhanh chóng hơn. Khi tạo một AC, ta cần phải đặt tên và liên kết nó với một gói PL/SQL chứa các hàm định nghĩa giá trị của các thuộc tính.

AC gồm được tạo nên bởi 2 thành phần:

- **Name:** Tên của bộ thuộc tính được liên kết với giá trị.

Ví dụ: Ta có AC empno_ctx truy xuất một ID nhân viên từ bảng HR. EMPLOYEES, thì nó có thể có tên (name) là employee_id.

- **Value:** Một giá trị được đặt bởi thuộc tính.

Ví dụ: Cho AC empno_ctx, để truy xuất ID nhân viên từ bảng HR. EMPLOYEES, ta có thể tạo một giá trị (value) gọi là emp_id để đặt giá trị cho ID.

Application Context hữu ích cho các mục đích sau:

- Thực thi kiểm soát truy cập chi tiết (ví dụ: trong chính sách Cơ sở dữ liệu riêng ảo của Oracle).
- Bảo vệ danh tính người dùng trên các môi trường nhiều lớp.
- Thực thi bảo mật mạnh mẽ hơn cho các ứng dụng của bạn, vì AC được kiểm soát bởi một quy trình đáng tin cậy chứ không phải người dùng.
- Tăng hiệu suất bằng cách đóng vai trò là bộ đệm dữ liệu an toàn cho các thuộc tính cần thiết cho ứng dụng để kiểm tra chi tiết hoặc để sử dụng trong các câu lệnh hoặc vòng lặp điều kiện PL/SQL.

Bộ đệm này tiết kiệm chi phí truy vấn cơ sở dữ liệu lặp đi lặp lại mỗi khi cần các thuộc tính này. Bởi vì AC lưu trữ dữ liệu phiên trong bộ đệm thay vì buộc các ứng dụng của bạn truy xuất dữ liệu này nhiều lần từ một table, nên nó cải thiện đáng kể hiệu suất của các ứng dụng của bạn.

- Đóng vai trò là khu vực lưu trữ cho các cặp tên-giá trị mà ứng dụng có thể xác định, sửa đổi và truy cập.

I. Đặc trưng của Application context và cách thiết lập

1. Đặc trưng của Application context

Application Context có các đặc trưng nổi bật sau:

- Chỉ định thuộc tính cho từng ứng dụng.
- Cung cấp quyền truy cập vào các thuộc tính được xác định trước thông qua namespace USERENV (USERENV là một hàm hệ thống được cung cấp bởi cơ sở dữ liệu Oracle. Nó cung cấp thông tin về môi trường phiên làm việc hiện tại và các thuộc tính liên quan).
- Tăng tính bảo mật cho ứng dụng vì application context được kiểm soát bởi một thủ tục đáng tin mà không phải là người dùng.
- Tăng hiệu suất bằng cách phục vụ như một bộ đệm dữ liệu an toàn cho các thuộc tính cần thiết cho một ứng dụng để kiểm tra chi tiết (fine-grained auditing) hoặc để sử dụng trong các câu lệnh hoặc vòng lặp có điều kiện PL/SQL.
- Đóng vai trò là một vùng lưu trữ các cặp giá trị name-value mà ứng dụng có thể xác định, sửa đổi và truy cập.
- Bảo vệ danh tính người dùng trong môi trường nhiều lớp.

2. Thiết lập Application context

Để tạo một context, ta sử dụng procedure DBMS_SESSION.SET_CONTEXT trong DBMS_SESSION được Oracle Database cung cấp.

Cú pháp:

```
DBMS_SESSION.SET_CONTEXT (  
    namespace VARCHAR2,  
    attribute  VARCHAR2,  
    value      VARCHAR2,  
    username   VARCHAR2,  
    client_id  VARCHAR2 );
```

Tham số	Mô tả
namespace	Tên namespace của ngữ cảnh được cài đặt
attribute	Tên của thuộc tính trong ngữ cảnh ứng dụng
value	Giá trị mới của thuộc tính
username	Tên người dùng CSDL của ngữ cảnh ứng dụng.

	Mặc định: NULL
client_id	Giá trị cụ thể của id máy khách của ngữ cảnh ứng dụng. Tối đa 64 byte. Mặc định: NULL

GIÁ TRỊ CẬP THUỘC TÍNH	Mô tả
Username = null Client_id = null	Tất cả mọi người đều có thể truy cập vào ngữ cảnh ứng dụng. Cặp giá trị này cũng sử dụng để cho ngữ cảnh ứng dụng dựa trên phiên truy cập (thường thì không định nghĩa, 2 thuộc tính tự null).
Username = <giá trị> Client_id = null	Cho phép một ngữ cảnh ứng dụng có thể được truy cập bởi nhiều session khác nhau, miễn là cùng một username
Username = null Client_id = <giá trị>	Cho phép một ngữ cảnh ứng dụng được truy cập bởi nhiều session của nhiều người dùng khác nhau
Username = <giá trị> Client_id = <giá trị>	Có 2 trường hợp: <ul style="list-style-type: none"> - Lightweight users: Nếu người dùng không có một tài khoản trong CSDL, thuộc tính username trở thành chủ của các kết nối. Thuộc tính client_id trở thành liên kết với người dùng CSDL không đăng nhập. - Database users: đối với người dùng là một người có tài khoản thì sẽ có thể sử dụng cho stateless web session

Ví dụ

```
DBMS_SESSION.SET_CONTEXT('hqtcsdl_user_ctx', 'hqtcsdl_user_phg', 'nv001');
```

II. Session-based/Global application context

1. Session-based application context

1.1 Database session-based application context

Loại ngữ cảnh ứng dụng này sử dụng thủ tục PL/SQL trong Oracle để truy xuất, đặt và bảo mật dữ liệu mà nó quản lý

Bối cảnh ứng dụng dựa trên phiên cơ sở dữ liệu được quản lý hoàn toàn trong Oracle. Oracle đặt các giá trị và sau đó khi người dùng thoát khỏi phiên, sẽ tự động xóa các giá trị AC được lưu trong bộ đệm. Nếu kết nối người dùng kết thúc bất thường, chẳng hạn như trong khi mất điện, thì quy trình nền PMON sẽ dọn sạch dữ liệu AC. Bạn không cần phải xóa rõ ràng AC khỏi bộ đệm

Ưu điểm của việc Database session-base application context là có thể tập trung hóa việc quản lý AC. Bất kỳ ứng dụng nào truy cập cơ sở dữ liệu này sẽ cần sử dụng AC này để cho phép hoặc ngăn người dùng truy cập vào ứng dụng đó. Điều này mang lại lợi ích cả về hiệu suất được cải thiện và bảo mật mạnh mẽ hơn

Lưu ý: Nếu người dùng là người dùng ứng dụng, nghĩa là người dùng không có trong cơ sở dữ liệu, hãy cân nhắc sử dụng ngữ cảnh ứng dụng chung (Global application context) để thay thế.

Có 3 loại bối cảnh ứng dụng dựa trên phiên cơ sở dữ liệu (Database session-based application context):

- **Initialized locally** (Khởi tạo cục bộ): Khởi tạo ngữ cảnh ứng dụng cục bộ cho phiên của người dùng.
- **Initialized externally** (Khởi tạo bên ngoài): Khởi tạo AC từ ứng dụng giao diện cuộc gọi Oracle (OCI), quy trình xếp hàng công việc hoặc liên kết cơ sở dữ liệu người dùng được kết nối. Cách khởi tạo này giúp tăng hiệu suất nhờ việc lưu trữ application context ở UGA (User Global Area).
- **Initialized globally** (Khởi tạo toàn cục): Sử dụng các thuộc tính và giá trị từ một vị trí tập trung, chẳng hạn như thư mục LDAP (khi đó nó có thể được sử dụng một cách toàn cục từ thư mục LDAP).

1.2 Client session-base application context

Loại application context này sử dụng các chức năng Oracle Call Interface của Oracle ở phía máy khách để đặt dữ liệu phiên của người dùng, sau đó thực hiện kiểm tra bảo mật cần thiết để hạn chế quyền truy cập của người dùng.

Lưu ý: Các chức năng Oracle Call Interface (OCI) có thể thiết lập và xóa thông tin theo phiên của người dùng trên User Global Area (UGA).

Ưu điểm của loại Application Context này :

- Trong Session-based Application Context là một ứng dụng riêng lẻ có thể kiểm tra dữ liệu theo phiên của người dùng phi cơ sở dữ liệu cụ thể, thay vì để cơ sở dữ liệu thực hiện tác vụ này
- Các lệnh gọi để đặt giá trị ngưỡng của ứng dụng được bao gồm trong lệnh gọi tiếp theo tới máy chủ, giúp cải thiện hiệu suất.

Tuy nhiên, hãy lưu ý rằng bảo mật của application context bị xâm phạm với client session-base application context: bất kỳ người dùng ứng dụng nào cũng có thể thiết lập client application context và không có kiểm tra nào được thực hiện trong cơ sở dữ liệu.

2. Global application context

Global Application Context (GAC) cho phép các giá trị ngưỡng ứng dụng có thể truy cập được trong các phiên cơ sở dữ liệu, bao gồm các phiên bản Oracle RAC. Oracle lưu trữ thông tin GAC trong System Global Area (SGA) (hoặc đôi khi có thể được gọi là Shared Global Area) để có thể sử dụng nó cho các ứng dụng sử dụng mô hình không phiên, chẳng hạn như các ứng dụng tầng giữa trong kiến trúc ba tầng. Các ứng dụng này không thể sử dụng Session-based AC xác thực với ứng dụng và sau đó ứng dụng này thường kết nối với cơ sở dữ liệu dưới dạng một danh tính duy nhất. Oracle khởi tạo GAC một lần, thay vì cho từng phiên người dùng. Điều này cải thiện hiệu suất vì các kết nối được sử dụng lại từ một nhóm kết nối.

Có ba cách sử dụng chung cho GAC

- Phải chia sẻ giá trị ứng dụng trên toàn cầu cho tất cả người dùng cơ sở dữ liệu
- Người dùng cơ sở dữ liệu phải di chuyển từ ứng dụng này sang ứng dụng khác.
- Phải xác thực người dùng không phải cơ sở dữ liệu, nghĩa là người dùng không biết đến cơ sở dữ liệu

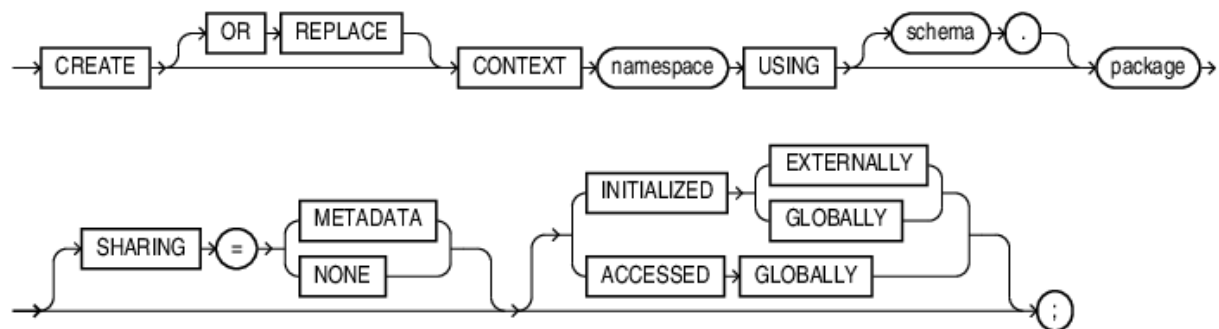
Lưu ý: có thể xóa giá trị GAC bằng cách chạy câu lệnh SQL ALTER SYSTEM FLUSH GLOBAL_CONTEXT.

III. Cách áp dụng Application Context cho Fine-grained Access Control

Lưu ý: Demo ở phần này sẽ tiếp tục phần demo ở phần B.IV.1 (sử dụng DBMS_RLS package).

Bước 1: Tạo namespace context:

Cú pháp:



```
CREATE [ OR REPLACE ] CONTEXT namespace
  USING [ schema. ] package
  [ SHARING "=" ( METADATA | NONE ) ]
  [ INITIALIZED { EXTERNALLY | GLOBALLY }
  | ACCESSED GLOBALLY] ;
```

Tham số	Mô tả
namespace	Tên namespace của namespace context được tạo. Context namespace luôn được lưu trữ trong schema SYS
schema	Schema sẽ sở hữu package. Nếu bỏ thông tham số schema, thì giá trị của tham số sẽ là scheme hiện tại.
package	Tên package PL/SQL sẽ được tạo ra hoặc đặt lại các thuộc tính ngữ cảnh trong namespace của session-user
sharing	Tùy chọn này cho phép chỉ định chế độ chia sẻ của ngữ cảnh ứng dụng. Có hai giá trị có thể là METADATA hoặc NONE.
INITIALIZED Clause	Cho phép chỉ định cách khởi tạo giá trị của ngữ cảnh ứng dụng. Có hai giá trị có thể là EXTERNALLY hoặc GLOBALLY

	<p>EXTERNALLY chỉ định rằng ngữ cảnh được khởi tạo từ bên ngoài.</p> <p>GLOBALLY chỉ định rằng ngữ cảnh được khởi tạo tự động khi cơ sở dữ liệu được khởi động</p>
ACCESSED GLOBALLY	Tùy chọn này cho phép chỉ định rằng giá trị của ngữ cảnh ứng dụng có thể được truy cập từ bất kỳ session nào.

Ví dụ:

```
CREATE OR REPLACE CONTEXT hqtcsdl_user_ctx USING set_hqtcsdl_user_ctx_pkg;
```

Bước 2: Tạo context package

```
CREATE OR REPLACE PACKAGE set_hqtcsdl_user_ctx_pkg
IS
    PROCEDURE set_hqtcsdl_role_context;
END set_hqtcsdl_user_ctx_pkg;
```

Bước 3: Định nghĩa Procedure trong package

```
CREATE OR REPLACE PACKAGE BODY set_hqtcsdl_user_ctx_pkg
IS
    PROCEDURE set_hqtcsdl_role_context
    IS
        check_role nvarchar2(50);
        role_name nvarchar2(50);
        u_phg nvarchar2(50);
    begin
        select vaitro, phg into check_role, u_phg from HQT_CSDLNC.NHANVIEN
        where manv = sys_context('userenv', 'session_user');
        if(check_role = 'NHÂN VIÊN') THEN
            role_name := 'R_NHANVIEN';
        ELSIF (check_role = 'TRƯỞNG PHÒNG') THEN
            role_name := 'R_TRUONGPHONG';
        ELSE
            role_name := '123';
        END IF;
    end;
```

```

        DBMS_SESSION.SET_CONTEXT('hqtcsdl_user_ctx', 'hqtcsdl_role',
role_name);

        DBMS_SESSION.SET_CONTEXT('hqtcsdl_user_ctx', 'hqtcsdl_user_phg',
u_phg);

    end;

end;

```

Bước 4: Tạo trigger để mỗi lần đăng nhập, procedure trong package được chạy để tạo context.

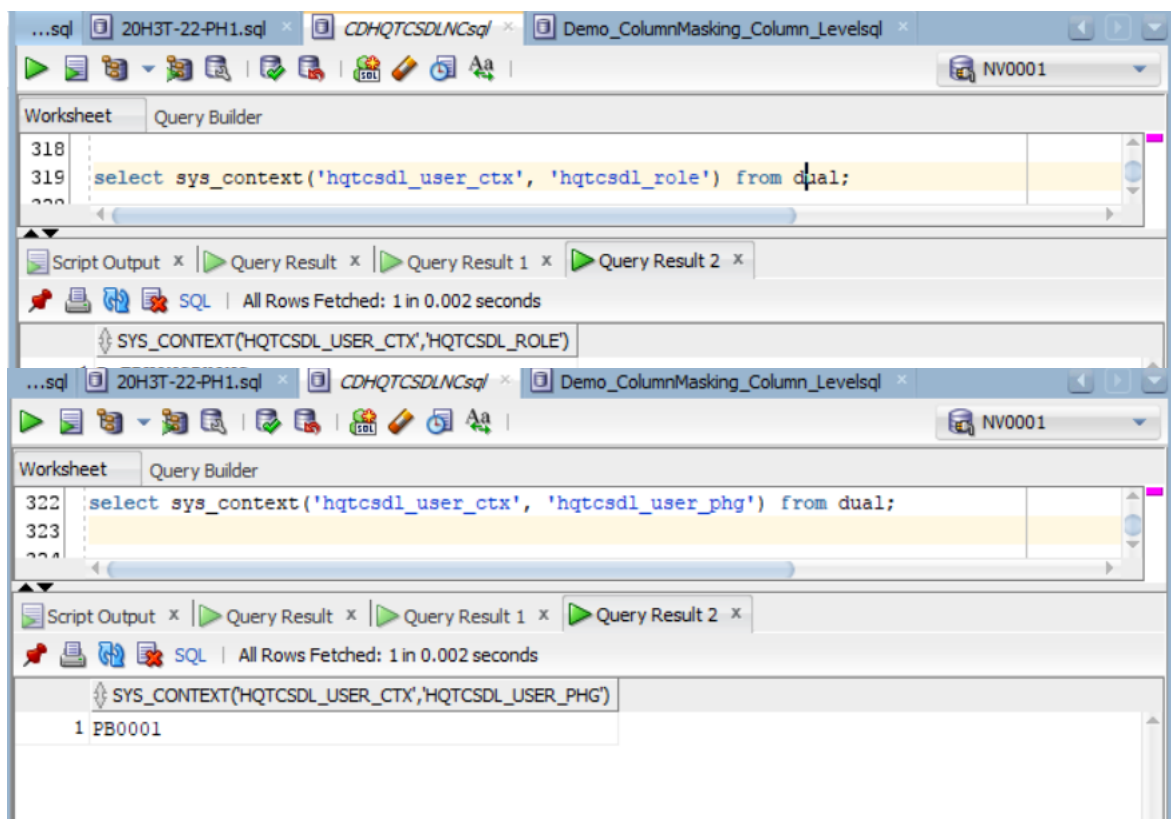
```

CREATE OR REPLACE TRIGGER hqtcsdl_trigger_role_ctx
AFTER LOGON ON DATABASE
BEGIN
    set_hqtcsdl_user_ctx_pkg.set_hqtcsdl_role_context;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        dbms_output.put_line(SUBSTR(SQLERRM, 1 , 64));
end;

```

- **Lưu ý:** Khi thực hiện demo, sau khi đã tạo context, nếu user đã được đăng nhập vào database trước khi tạo context thì context sẽ không có hiệu lực. Để xử lý vấn đề trên, hãy disconnect user và đăng nhập lại vào database. Khi đó context sẽ có hiệu lực.

➤ **Kết quả:**



Bước 5: Áp dụng application context vào policy function

```
CREATE OR REPLACE FUNCTION FUNC_POLIC_TRUONGPHONG_1(O_SCHEMA VARCHAR2,
O_NAME VARCHAR2)
RETURN VARCHAR2
AS
    V_ROLE VARCHAR2(50);
    USERSESSION VARCHAR2(50);
    PB VARCHAR2(50);
BEGIN
    USERSESSION:= SYS_CONTEXT('userenv', 'session_user'); -- nv0002
    IF(USERSESSION = 'HQT_CSDLNC') THEN
        RETURN '';
    ELSE
        V_ROLE := SYS_CONTEXT('hqtcsdl_user_ctx', 'hqtcsdl_role');
        IF(UPPER(V_ROLE) = 'R_TRUONGPHONG') THEN
            RETURN 'PHG = sys_context(''hqtcsdl_user_ctx'',
            ''hqtcsdl_user_phg'')';
        ELSIF (UPPER(V_ROLE) = 'R_NHANVIEN') THEN
            RETURN 'manv = sys_context(''userenv'', ''session_user'')';
        ELSE
```

```

        RETURN '';
    END IF;
END IF;
END;
```

- Ở policy function trên, sau khi đã áp dụng application context, ta có thể điều hướng kết quả trả về của function theo vai trò và rút gọn được vị từ trả về.

Kết quả truy vấn vẫn không thay đổi:

Worksheet | Query Builder

365 `select * from HQT_CSDLNC.NHANVIEN`

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.009 seconds

	MANV	TENNV	LUONG	VAITRO	PHG
1 NV0002	Jane Smith	6000.75	NHÂN VIÊN	PB0001	

Worksheet | Query Builder

350
351 `select * from HQT_CSDLNC.NHANVIEN`

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.01 seconds

	MANV	TENNV	LUONG	VAITRO	PHG
1 NV0001	John Doe	5000.5	TRƯỞNG PHÒNG	PB0001	
2 NV0002	Jane Smith	(null)	NHÂN VIÊN	PB0001	
3 NV0005	Emily Brown	(null)	NHÂN VIÊN	PB0001	

Tuy nhiên, về mặt chính sách thực thi (Execution Plan) đã có những thay đổi khá đáng kể như sau:

- Khi không áp dụng AC

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 3
FILTER				
Filter Predicates				
OR				
MANV=SYS_CONTEXT('userenv','session_user')				
EXISTS (SELECT 0 FROM HQT_CSDLNC.PHONGBAN PHONGBAN WHERE MAPB=:B1 AND CASE MAPB WHEN SYS_CONTEXT('hqtcsdl_user_ctx','hqtcsdl_user_phg') THEN TRPHG ELSE NULL END =SYS_CONTE				
TABLE ACCESS	HQT_CSDLNC.NHANVIEN	FULL	5	3
TABLE ACCESS	HQT_CSDLNC.PHONGBAN	BY INDEX ROWID	1	1
Filter Predicates				
CASE MAPB WHEN SYS_CONTEXT('hqtcsdl_user_ctx','hqtcsdl_user_phg') THEN TRPHG ELSE NULL END =SYS_CONTEXT('userenv','session_user')				
INDEX	HQT_CSDLNC.ATBMHITT_TABLE_PHO...	UNIQUE SCAN	1	0
Access Predicates				
MAPB=:B1				
Other XML				

- Khi áp dụng AC

420

421

SELECT * FROM hqt_CSDLNC.NHANVIEN;

Script Output x

Query Result x

Explain Plan x

SQL

0.05 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
<div>SELECT STATEMENT</div>			3	3
<div> <div>TABLE ACCESS</div> <div>Filter Predicates</div> <div>PHG=SYS_CONTEXT('hqtcsdl_user_ctx','hqtcsdl_user_phg')</div> <div>Other XML</div> </div>	<div>HQT_CSDLNC.NHANVIEN</div>	<div>FULL</div>	3	3

Ta có thể thấy thay vì phải nhọc nhằn thực hiện nhiều lần truy xuất phức tạp để có được thông tin session_users thì dựa vào AC đã tối ưu được rất nhiều bước, truy xuất và thời gian để có được session_users đó cũng là cơ sở để AC tối ưu được hiệu suất

IV. Kiểm tra tác dụng policy: View V\$VPD_POLICY

View V\$VPD_POLICY hiển thị tất cả các chính sách bảo mật chi tiết và các vị từ bảo mật liên quan đến các con trỏ hiện tại nằm trong bộ đệm.

Chi tiết trong bảng sau:

Tên	Loại dữ liệu	Mô tả
ADDRESS	RAW (4 8)	Địa chỉ con trỏ
PARADDR	RAW (4 8)	Địa chỉ con trỏ tạo ra con trỏ Address
SQL_HASH	NUMBER	Số hash câu SQL
SQL_ID	VARCHAR2(13)	Id SQL
CHILD_NUMBER	NUMBER	Số con trỏ con mà Paraddr tạo ra
OBJECT_OWNER	VARCHAR2(30)	Người sở hữu đối tượng gắn với chính sách
OBJECT_NAME	VARCHAR2(30)	Tên của đối tượng gắn với chính sách
POLICY_GROUP	VARCHAR2(30)	Tên của nhóm chính sách
POLICY	VARCHAR2(30)	Tên của chính sách
POLICY_FUNCTION_OWNER	VARCHAR2(30)	Người sở hữu hàm gắn với chính sách
PREDICATE	VARCHAR2(4000)	Vị từ trả về của chính sách (giới hạn ở 4000 bytes)
CON_ID	NUMBER	0: được sử dụng cho các dòng có dữ liệu liên quan đến toàn bộ Container Database. Đồng thời cũng được sử dụng cho các dòng dữ liệu không liên quan đến Container Database. 1: Giá trị này được sử dụng cho các dòng dữ liệu có liên quan đến thư mục root. n: số dòng chứa dữ liệu liên quan đến ID container.

Kết quả:

```
324 select object_owner, object_name, policy, policy_group, predicate from V$VPD_POLICY
325
```

OBJECT_OWNER	OBJECT_NAME	POLICY	POLICY_GROUP	PREDICATE
1 HQT_CSDLNC	NHANVIEN	POLICY_TRUONGPHONG_1	SYS_DEFAULT	phg = sys_context('hqtcsdl_user_ctx', 'hqtcsdl_user_phg')
2 HQT_CSDLNC	NHANVIEN	POLICY_MASKING_USER	SYS_DEFAULT	MANV = sys_context('userenv', 'session_user')
3 HQT_CSDLNC	NHANVIEN	POLICY_TRUONGPHONG_1	SYS_DEFAULT	(null)
4 HQT_CSDLNC	NHANVIEN	POLICY_MASKING_USER	SYS_DEFAULT	MANV = sys_context('userenv', 'session_user')

D. Lưu ý của nhóm

Trong quá trình tìm hiểu và thực hiện demo, nhóm có một số kinh nghiệm muốn chia sẻ:

- **ORA-28113: policy predicate has error:** Đây là lỗi vị từ, hãy kiểm tra vị từ trả về của bạn chắc chắn đúng. Bạn có thể kiểm tra file trace mới nhất để xem lỗi như thế nào. Bạn có thể lấy đường dẫn tới bằng câu lệnh “*select value from v\$diag info where name='Diag Trace'.*”.
- Khi thực hiện truy vấn View USER_ROLE_PRIVS, kết quả sẽ trả về role của current-user (Owner của function), không phải là role của session-user mà bạn muốn truy vấn.
 - Ví dụ, có bảng TEST được áp dụng 1 policy sử dụng function Func_demo thuộc schema của user DEMO có role DBA, NHANVIEN. Trong Func_demo, có một câu lệnh “*Select granted_role into <tên biến> from USER_ROLE_PRIVS*”.
 - User NV001 có role thuộc NHANVIEN, khi truy vấn vào bảng TEST, thì kết quả trả về của view USER_ROLE_PRIVS trong policy function là DBA, NHANVIEN (role của current-user DEMO).
 - Do đó, câu lệnh “*Select granted_role into <tên biến> from USER_ROLE_PRIVS*” sẽ xảy ra exception “TOO_MANY_ROWS: ORA-01422: When a ‘SELECT’ statement with INTO clause returns more than one row”. Nhưng lỗi được thông báo là **ORA-28112**.
- Ngược lại với USER_ROLE_PRIVS, view SESSION_ROLES sẽ trả về role đang kích hoạt của session-user. Khi truy vấn vào view này ở trong policy function, sẽ không có kết quả trả về.

- Do đó, sẽ xảy ra exception “NO_DATA_FOUND: ORA-01403: When ‘SELECT’ statement that contains INTO clause fetches no rows”. Nhưng lỗi được thông báo là **ORA-28112**.
- Hai view được nêu ở trên cũng trả về kết quả tương tự khi sử dụng với procedure và function trong package thuộc namespace context. Vì vậy, hãy cân nhắc thật kĩ khi sử dụng 2 view trên trong function và procedure.

E. Kết luận

Việc bảo vệ an toàn cho CSDL luôn là một việc vô cùng quan trọng. Tuy nhiên, bảo vệ như thế nào lại là một vấn đề không hề đơn giản. Với những nội dung được trình bày trong bài báo cáo này, chúng ta có thể thấy rằng bên trong các hệ quản trị CSDL luôn có các cơ chế an toàn được hỗ trợ sẵn. Fine-grained Access Control là một biện pháp tốt, nó tận dụng tốt các hỗ trợ của CSDL Oracle như VPD, Application Context... VPD tốt trong việc hỗ trợ bảo vệ CSDL ở mức hàng và mức cột. Application Context tốt trong việc tối ưu hiệu suất làm việc của VPD điều đó đưa ra một tổ hợp tốt cho các nhà quản trị khi cần thiết lập các biện pháp bảo vệ cho dữ liệu.

Tuy nhiên, trong bài báo cáo, nhóm cũng chỉ ra được những điều đáng lưu tâm mà khi người quản trị có ý sử dụng FGAC (với tổ hợp VPD và AC) để kiểm soát dữ liệu của mình. Lời khuyên của nhóm là hãy chú ý đến mục đích sử dụng của yêu cầu, đánh giá tốt mục đích để đưa ra những biện pháp phù hợp nhất chứ không khẳng định sử dụng một và chỉ một, đồng thời, lưu tâm đến những lưu ý và lời khuyên của Oracle hay những thứ mà nhóm đã hỗ trợ đưa ra.

Tóm lại, nhóm hy vọng bài báo cáo có thể giúp bạn hiểu được phần nào về FGAC, VPD và AC. Hãy vận dụng tốt những kiến thức và bài báo cáo để có thể đảm bảo an toàn CSDL tốt hơn.

TÀI LIỆU THAM KHẢO

[1] Cơ chế an toàn dựa vào nhãn và CSDL trên Oracle, *Ban cơ yếu chính phủ an toàn thông tin*:

[Cơ chế an toàn dựa vào nhãn và CSDL trên Oracle - Tạp chí An toàn thông tin \(antoanthongtin.gov.vn\)](http://antoanthongtin.gov.vn)

[2] PL/SQL Packages and Types Reference, *Oracle Help Center*:

[DBMS_QLS \(oracle.com\)](http://dbms_qls.oracle.com)

[3] 13 Using Virtual Private Database to Implement Application Security Policies, *Oracle® Database Security Guide*:

[13 Using Virtual Private Database to Implement Application Security Policies \(oracle.com\)](http://oracle.com)

[4] 14 Implementing Application Context and Fine-Grained Access Control, *Oracle® Database Security Guide:s*

[14 Implementing Application Context and Fine-Grained Access Control \(oracle.com\)](http://oracle.com)

[5] Oracle in a Nutshell, *O'Reilly Media, Inc.:*

[Application Context | Oracle in a Nutshell \(oreilly.com\)](http://oreilly.com)

[6] 14 Using Oracle Virtual Private Database to Control Data Access, *Oracle Help Center*:

[Using Oracle Virtual Private Database to Control Data Access](http://oracle.com)

[7] Virtual Private Database (VPD), *GeeksforGeeks*:

<https://www.geeksforgeeks.org/virtual-private-database-vpd/>

[8] Virtual Private Database, *Oracle*:

[Virtual Private Database \(oracle.com\)](http://oracle.com)

[9] Oracle Virtual Private Database (VPD) – 1, *IT Tutorial*:

<https://ittutorial.org/oracle-virtual-private-database-vpd/>

[10] What is Fine-Grained Access Control? (And Why It's So Important), *Immuta*:

[What is Fine-Grained Access Control? \(And Why It's So Important\) \(immuta.com\)](http://immuta.com)

[11] 7 Defaults and Policies, *ORACLE Help Center*:

https://docs.oracle.com/cd/E91325_01/OBREF/defaults_policies.htm#OBREF383