

**NHÓM 9**

**6.3 FINE-GRAINED  
ACCESS CONTROL**



# CÁC PHẦN TRÌNH BÀY

**1. VPD**

**2. APPLICATION CONTEXT**

**3. DEMO**

“ *Fine-grained access control (FGAC): cho phép người quản trị dùng các function để hiện thực các chính sách bảo mật và liên kết các chính sách bảo mật đó với các table, view hoặc synonym. Việc gán các chính sách như vậy khiến cho những người dùng với quyền hạn khác nhau sẽ thấy được những “khung nhìn” khác nhau đối với đối tượng được bảo vệ.*

1

VPD



# MỤC LỤC VPD

1. Khái niệm
2. Cách làm việc của VPD
3. Column-level và Column Masking
4. VPD SECURITY POLICY
5. Phạm vi VPD



# 1. KHÁI NIỆM

- Virtual Private Database (Oracle VPD) là một cơ chế AC của Oracle kết hợp việc sử dụng Công cụ điều khiển truy cập (Fine Grained Access Control) và Ngưỡng ứng dụng (Application Context).
- VPD cho phép định nghĩa các chính sách an toàn đến mức từng đối tượng đối tượng cho từng thao tác.

# 1. KHÁI NIỆM

Start Page x SIMPLEACConn x NV01 x NV02 x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN;`

Query Result x

SQL | All Rows Fetched: 4 in 0 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	100	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM



Start Page x SIMPLEACConn x NV01 x NV02 x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN;`

Query Result x

SQL | All Rows Fetched: 4 in 0.141 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	(null)	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	(null)	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	(null)	NTHIEN@GMAIL.COM

Start Page x SIMPLEACConn x NV01 x NV02 x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.BENHNHAN;`

Query Result x

SQL | All Rows Fetched: 3 in 0.016 seconds

	ID_BENHNHAN	EMAIL_BACSI	BENH
1	BN0001	HHONG@GMAIL.COM	SOI THAN
2	BN0002	BDUY@GMAIL.COM	HIV
3	BN0003	NTHIEN@GMAIL.COM	UNG THU



Start Page x SIMPLEACConn x NV01 x NV02 x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.BENHNHAN;`

Query Result x

SQL | All Rows Fetched: 1 in 0.016 seconds

	ID_BENHNHAN	EMAIL_BACSI	BENH
1	BN0001	HHONG@GMAIL.COM	SOI THAN



## 2. CÁCH LÀM VIỆC CỦA VPD

Dùng hàm trả về vị từ (predicate) thêm vào truy vấn của người dùng sau mệnh đề WHERE để kiểm soát dữ liệu họ truy cập dựa trên cách cài đặt

The screenshot shows a SQL query editor with a toolbar and a query window. The query is as follows:

```
SELECT * FROM SIMPLEAC.BENHNHAN WHERE  
EMAIL_BACSI IN (SELECT NV.EMAIL FROM SIMPLEAC.NHANVIEN NV WHERE SYS_CONTEXT('USERENV', 'SESSION_USER') = USERNAME)  
AND ('BACSI' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT  
WHERE NVVT.USERNAME = SYS_CONTEXT('USERENV', 'SESSION_USER')));
```

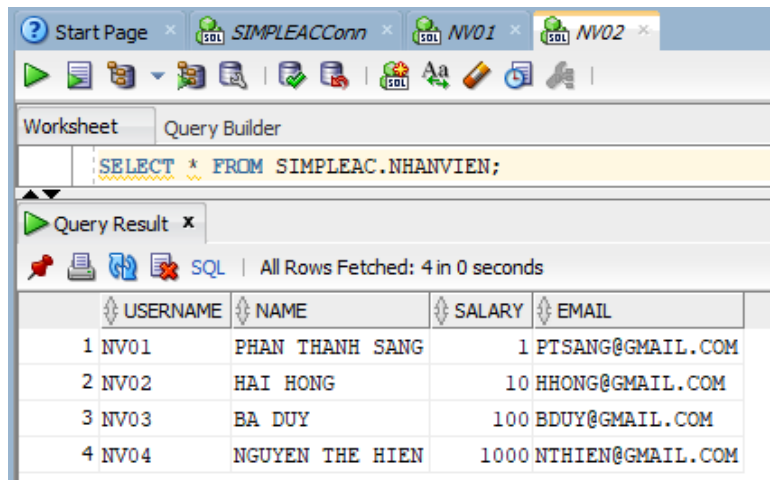
Below the query, the "Query Result" window shows the following data:

ID_BENHNHAN	EMAIL_BACSI	BENH
1 BN0001	HHONG@GMAIL.COM	SOI THAN



### 3. COLUMN LEVEL VÀ COLUMN MASKING

- Cung cấp các truy cập chi tiết hơn trên dữ liệu
- Gắn hàm chính sách trên cột cần hạn chế truy cập, thay vì gắn trên table hoặc view



Start Page x SIMPLEACConn x NV01 x NV02 x

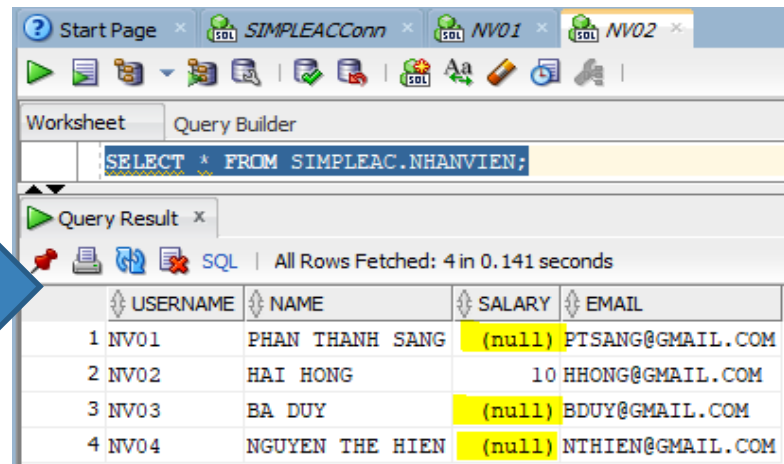
Worksheet Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN;`

Query Result x

SQL | All Rows Fetched: 4 in 0 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	100	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM



Start Page x SIMPLEACConn x NV01 x NV02 x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN;`

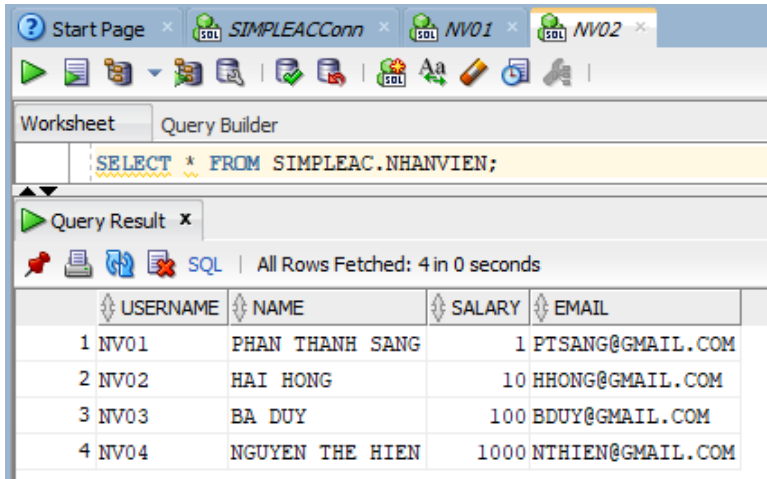
Query Result x

SQL | All Rows Fetched: 4 in 0.141 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	(null)	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	(null)	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	(null)	NTHIEN@GMAIL.COM

### 3. COLUMN LEVEL VÀ COLUMN MASKING

- Dùng để hiển thị các cột chứa dữ liệu nhạy cảm dưới dạng giá trị NULL



Start Page | SIMPLEACConn | NV01 | NV02

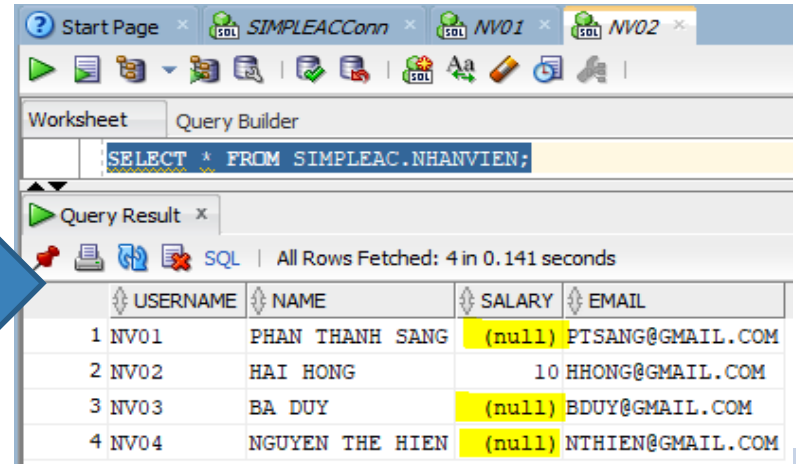
Worksheet | Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN;`

Query Result x

SQL | All Rows Fetched: 4 in 0 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	100	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM



Start Page | SIMPLEACConn | NV01 | NV02

Worksheet | Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN;`

Query Result x

SQL | All Rows Fetched: 4 in 0.141 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	(null)	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	(null)	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	(null)	NTHIEN@GMAIL.COM



## 4. VPD SECURITY POLICY

THỦ TỤC	MÔ TẢ
DBMS_RLS.ADD_POLICY	Thêm policy vào table, view, synonym
DBMS_RLS.ENABLE_POLICY	Cho phép (hay vô hiệu hóa) một policy trước đó mà ta đã thêm vào table, view, synonym
DBMS_RLS.ALTER_POLICY	Thay đổi liên kết hay không liên kết thuộc tính với policy
DBMS_RLS.REFRESH_POLICY	Vô hiệu hóa các cursor liên kết với policy, làm mới policy.
DBMS_RLS.DROP_POLICY	Xóa một policy ra khỏi table, view, synonym



## 4. VPD SECURITY POLICY

```
select * from ALL_POLICIES;
```

	OBJECT_OWNER	OBJECT_NAME	POLICY_GROUP	POLICY_NAME	PF_OWNER
1	SIMPLEAC	NHANVIEN	SYS_DEFAULT	VPD_NHANVIEN_POLICY	SIMPLEAC
2	SIMPLEAC	BENHNHAN	SYS_DEFAULT	VPD_BENHNHAN_POLICY	SIMPLEAC

THAM SỐ	Kiểu dữ liệu	CÓ THỂ NULL	Mô tả
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner của table, view, hay synonym (thuộc quyền sở hữu)
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Tên của table, view, hay synonym
POLICY_GROUP	VARCHAR2(30)	NOT NULL	Tên của nhóm policy
POLICY_NAME	VARCHAR2(30)	NOT NULL	Tên của policy
PF_OWNER	VARCHAR2(30)	NOT NULL	Owner hàm của policy



## 4. VPD SECURITY POLICY

```
select * from ALL_POLICIES;
```

PACKAGE	FUNCTION	SEL	INS	UPD	DEL	IDX	CHK_OPTION	ENABLE	STATIC_POLICY	POLICY_TYPE	LONG_PREDICATE
(null)	VPD_NHANVIEN	YES	NO	NO	NO	NO	NO	YES	NO	DYNAMIC	NO
(null)	VPD_BENHNHAN	YES	NO	YES	YES	NO	NO	YES	NO	DYNAMIC	NO

THAM SỐ	Kiểu dữ liệu	Mô tả
CHK_OPTION	VARCHAR2(3)	Cho biết có được kiểm tra option khi thực thi policy hay không (yes – no)
ENABLE	VARCHAR2(3)	Cho biết policy đang ở trạng thái cho phép – enable hay vô hiệu hóa – disable (yes – no)
STATIC_POLICY	VARCHAR2(3)	Cho biết policy có đang ở trạng thái static hay không (yes – no)
POLICY_TYPE	VARCHAR2(24)	Policy type: STATIC, SHARED_STATIC, CONTEXT_SENSITIVE, SHARED_CONTEXT_SENSITIVE DYNAMIC



## 5. PHẠM VI VPD

- Khi export data, chính sách VPD không được thi hành khi export data trực tiếp từ đường dẫn. Khi xuất dữ liệu trực tiếp từ một đường dẫn, Oracle Database đọc dữ liệu từ đĩa vào bộ đệm và chuyển dữ liệu trực tiếp đi.



## 5. PHẠM VI VPD

```
Enter user-name: sys/123456 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> grant dba to kda;

Grant succeeded.

SQL> conn kda/123456
Connected.
SQL> select * from simpleac.benhnhan;

no rows selected

SQL> conn kda/123456 AS SYSDBA
Connected.
SQL> select * from simpleac.benhnhan;

ID_BENHNHAN          EMAIL_BACSI
-----
BENH
-----
BN0001              HHONG@GMAIL.COM
SOI THAN

BN0002              BDUY@GMAIL.COM
HIV

BN0003              NTHIEN@GMAIL.COM
UNG THU
```

SYS USER, SYS SCHEMA

SYSDBA CONNECT



## 5. PHẠM VI VPD

```
SQL> conn kda/123456
Connected.
SQL> select * from simpleac.benhnhan;

no rows selected

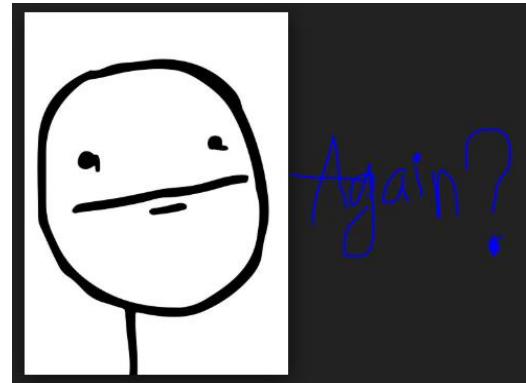
SQL> conn sys/123456 as sysdba
Connected.
SQL> grant EXEMPT ACCESS POLICY to kda;

Grant succeeded.

SQL> conn kda/123456
Connected.
SQL> select * from simpleac.benhnhan;
```

ID_BENHNHAN	EMAIL_BACSI
BENH	
BN0001	HHONG@GMAIL.COM
SOI THAN	
BN0002	BDUY@GMAIL.COM
HIV	
BN0003	NTHIEN@GMAIL.COM
UNG THU	

## EXEMPT ACCESS POLICY





# 2

## APPLICATION CONTEXT



# APPLICATION CONTEXT

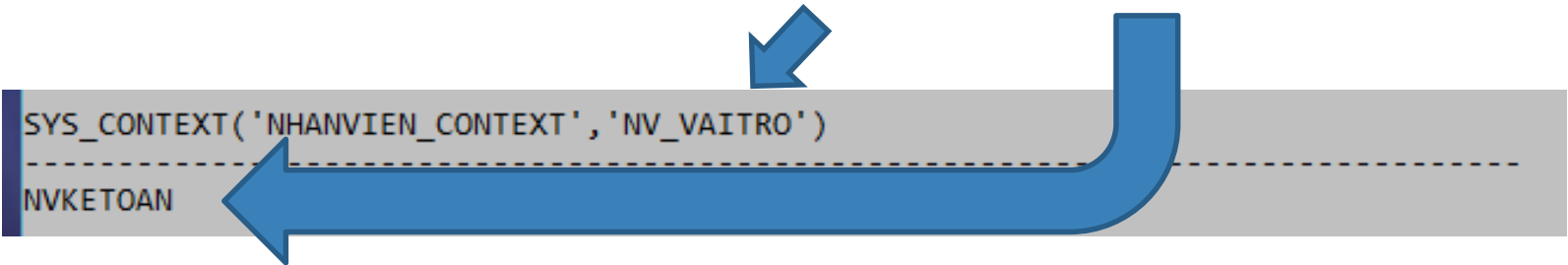
- Khái niệm
- Các đặt trưng
- Cú pháp tạo context
- Các loại Application Context
- Lợi ích



# KHÁI NIỆM

## Application Context (Ngữ cảnh ứng dụng):

- ▶ Là một phần của Cơ sở dữ liệu riêng ảo trong Oracle (VPD)
- ▶ Gồm 2 thành phần chính là attribute và value





## CÁC ĐẶC TRƯNG

- Ngữ cảnh ứng dụng thường nhóm các thuộc tính có liên quan thành một nhóm và được truy cập theo namespace. Giúp việc truy xuất nhanh hơn.
- Ngữ cảnh ứng dụng được xem như một biến toàn cục chứa thông tin được truy cập trong cơ sở dữ liệu.
- Các ngữ cảnh ứng dụng thường chứa các thuộc tính như tên người dùng, tiêu đề, ... và các chính sách bảo mật vẫn có thể đến các thuộc tính này khi người dung đăng nhập.
- Được xác định, thiết lập và lấy ra bởi người dùng hoặc các ứng dụng



## CÚ PHÁP CREATE CONTEXT

**CREATE OR REPLACE CONTEXT** <Context\_name> **USING**  
[Schema\_name].[Trusted\_procedure\_name]

- Context\_name : Tên của ngữ cảnh ứng dụng
- Trusted\_procedure\_name : Tên của thủ tục (thường là một gói Packet)



## CÚ PHÁP SET CONTEXT

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

Tham số	Ghi chú
namespace	Namespace của ngữ cảnh ứng dụng được đặt, tối đa 30 byte
attribute	Thuộc tính của ngữ cảnh ứng dụng, tối đa 30 byte
value	Giá trị của ngữ cảnh ứng dụng, giới hạn 4 Kilobyte
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



## CÁC LOẠI NGỮ CẢNH ỨNG DỤNG

- Client Session-based Application Context
- Database session-based Application Contexts
- Global Application Context

Initialized locally

Initialized externally

Initialized globally



## LỢI ÍCH

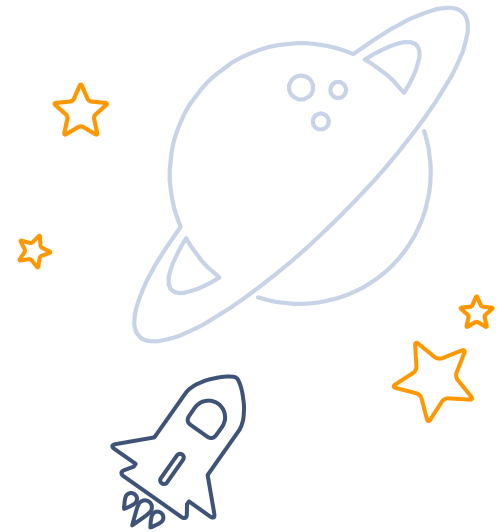
- Bảo mật danh tính người dùng trên các môi trường khác nhau.
- Giúp các biện pháp bảo mật trở nên mạnh hơn
- Giúp việc thiết lập các chính sách bảo mật(VPD) trở nên dễ dàng hơn



3

DEMO

# GIẢ ĐỊNH





# TÌNH HUỐNG

Start Page x SIMPLEACConn x

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN;
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.011 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	100	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

SIMPLEACConn x

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN_VAITRO;
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.001 seconds

	USERNAME	VAITRO
1	NV01	NVKETOAN
2	NV02	BACSI
3	NV03	BACSI
4	NV04	BACSI

SIMPLEACConn x

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.073 seconds

	ID_BENHNHAN	EMAIL_BACSI	BENH
1	BN0001	HHONG@GMAIL.COM	SOI THAN
2	BN0002	BDUY@GMAIL.COM	HIV
3	BN0003	NTHIEN@GMAIL.COM	UNG THU



## TÌNH HUỐNG

```
-----  
grant select on SIMPLEAC.NHANVIEN to NV01;  
grant select on SIMPLEAC.NHANVIEN to NV02;  
grant select on SIMPLEAC.NHANVIEN to NV03;  
grant select on SIMPLEAC.NHANVIEN to NV04;  
grant select on SIMPLEAC.NHANVIEN_VAITRO to NV01;  
grant select on SIMPLEAC.NHANVIEN_VAITRO to NV02;  
grant select on SIMPLEAC.NHANVIEN_VAITRO to NV03;  
grant select on SIMPLEAC.NHANVIEN_VAITRO to NV04;  
grant select on SIMPLEAC.BENHNHAN to NV01;  
grant select on SIMPLEAC.BENHNHAN to NV02;  
grant select on SIMPLEAC.BENHNHAN to NV03;  
grant select on SIMPLEAC.BENHNHAN to NV04;  
-----
```

- Cấp quyền cho các nhân viên xem các bảng để làm việc.
- Mọi người xem lương, biết chức vụ của mình và làm công việc của mình



# TÌNH HUỐNG

SIMPLEACConn x NV01 x

0.017 seconds

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN;  
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Query Result x Query Result 1 x Script Output x

Task completed in 0.017 seconds

USERNAME	NAME	SALARY	EMAIL
NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
NV02	HAI HONG	10	HHONG@GMAIL.COM
NV03	BA DUY	100	BDUY@GMAIL.COM
NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

ID_BENHNHAN	EMAIL_BACSI	BENH
BN0001	HHONG@GMAIL.COM	SOI THAN
BN0002	BDUY@GMAIL.COM	HIV
BN0003	NTHIEN@GMAIL.COM	UNG THU

- Sao kế toán lại xem được bệnh nhân?
- Họ đâu có phụ trách khám?



# TÌNH HUỐNG

SQL SIMPLEACConn x NV01 x NV02 x

0.021 seconds

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN;  
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Script Output x

Task completed in 0.021 seconds

USERNAME	NAME	SALARY	EMAIL
NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
NV02	HAI HONG	10	HHONG@GMAIL.COM
NV03	BA DUY	100	BDUY@GMAIL.COM
NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

ID_BENHNHAN	EMAIL_BACSI	BENH
BN0001	HHONG@GMAIL.COM	SOI THAN
BN0002	BDUY@GMAIL.COM	HIV
BN0003	NTHIEN@GMAIL.COM	UNG THU

- NV02 là bác sĩ HAIHONG, sao có thể xem bệnh nhân của 2 bác sĩ còn lại?
- Không ổn!



# TÌNH HUỐNG

The screenshot shows the SQL Developer interface with a query window and a script output window. The query window contains two SQL statements: `SELECT * FROM SIMPLEAC.NHANVIEN;` and `SELECT * FROM SIMPLEAC.BENHNHAN;`. The script output window shows the results of the first query, which is a table with four columns: USERNAME, NAME, SALARY, and EMAIL. The results are as follows:

USERNAME	NAME	SALARY	EMAIL
NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
NV02	HAI HONG	10	HHONG@GMAIL.COM
NV03	BA DUY	100	BDUY@GMAIL.COM
NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

Below the first table, there is another table with three columns: ID\_BENHNHAN, EMAIL\_BACSI, and BENH. The results are as follows:

ID_BENHNHAN	EMAIL_BACSI	BENH
BN0001	HHONG@GMAIL.COM	SOI THAN
BN0002	BDUY@GMAIL.COM	HIV
BN0003	NTHIEN@GMAIL.COM	UNG THU

- Không thể trông cậy mọi người tự giác tuân thủ.
- Chúng ta phải làm cho họ chỉ thấy thông tin họ nên thấy.



# CHÍNH SÁCH ĐỀ RA

Start Page x SIMPLEACConn x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN;`

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.011 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	100	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

SIMPLEACConn x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.NHANVIEN_VAITRO;`

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.001 seconds

	USERNAME	VAITRO
1	NV01	NVKETOAN
2	NV02	BACSI
3	NV03	BACSI
4	NV04	BACSI

SIMPLEACConn x

Worksheet Query Builder

`SELECT * FROM SIMPLEAC.BENHNHAN;`

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.073 seconds

	ID_BENHNHAN	EMAIL_BACSI	BENH
1	BN0001	HHONG@GMAIL.COM	SOI THAN
2	BN0002	BDUY@GMAIL.COM	HIV
3	BN0003	NTHIEN@GMAIL.COM	UNG THU

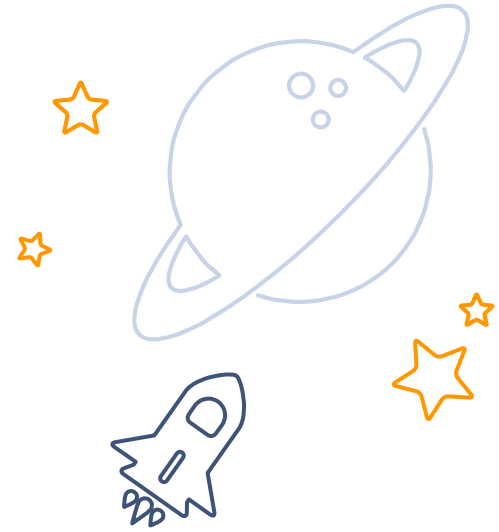
1. Các BACSI không xem lương của nhau, KETOAN xem lương tất cả
2. Các BACSI không xem bệnh nhân không phải của mình





# VPD

VIRTUAL PRIVATE DATABASE





# CHÍNH SÁCH VPD MỘT

Bước 1. Ta tạo ra hàm trả về vị từ cho chính sách tên là VPD\_NHANVIEN

```
-----  
Create or replace function VPD_NHANVIEN(schema varchar2,object varchar2)  
  return varchar2  
as  
  user varchar2(100);  
begin  
  if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then  
    return '';  
  else  
    user:= SYS_CONTEXT('userenv', 'SESSION_USER');  
    return 'USERNAME = user      OR (''NVKETOAN'' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT  
      WHERE NVVT.USERNAME = USER))';  
  end if;  
end;  
/  
-----
```



## CHÍNH SÁCH VPD MỘT

Bước 1. Ta tạo ra hàm trả về vị từ cho chính sách tên là VPD\_NHANVIEN

```
if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then  
    return '';  
else
```

Nếu là người dung SIMPLEAC thì trả về vị từ là "" tương đương với ko có điều kiện khi select dòng và sẽ trả về toàn bộ bảng.



Bước 1. Ta tạo ra hàm trả về vị từ cho chính sách tên là VPD\_NHANVIEN

```

else
    user:= SYS_CONTEXT('userenv', 'SESSION_USER');
    return 'USERNAME = user      OR (''NVKETOAN'' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
        WHERE NVVT.USERNAME = USER))';
end if;

```

Nếu khác thì trả về vị từ cột USERNAME = người dùng đang đăng nhập HOẶC 'NVKETOAN' có trong select dòng trong bảng NVVT với CỘT USERNAME = tên người dùng đang đăng nhập



# CHÍNH SÁCH VPD MỘT

SIMPLEACConn NV01 NV02

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN WHERE  
  USERNAME = SYS_CONTEXT('USERENV', 'SESSION_USER') OR ('NVKETOAN' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT  
    WHERE NVVT.USERNAME = SYS_CONTEXT('USERENV', 'SESSION_USER')));
```

Query Result x Query Result 1 x Script Output x Query Result 2 x

SQL | All Rows Fetched: 4 in 0.003 seconds

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	100	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

Kết quả tương tự như khi xài VPD cho user NV01.



# CHÍNH SÁCH VPD MỘT

The screenshot shows the SQL Developer interface with three tabs: SIMPLEACConn, NV01, and NV02. The Query Builder tab is active, displaying the following SQL query:

```
SELECT * FROM SIMPLEAC.NHANVIEN WHERE  
  USERNAME = SYS_CONTEXT('USERENV', 'SESSION_USER') OR ('NVKETOAN' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT  
    WHERE NVVT.USERNAME = SYS_CONTEXT('USERENV', 'SESSION_USER')));
```

Below the query, the Query Result tab shows the execution results. The status bar indicates "All Rows Fetched: 1 in 0.002 seconds". The results are displayed in a table with the following columns: USERNAME, NAME, SALARY, and EMAIL.

	USERNAME	NAME	SALARY	EMAIL
1	NV02	HAI HONG	10	HHONG@GMAIL.COM

Kết quả tương tự như khi xài VPD cho user NV02.



## CHÍNH SÁCH VPD MỘT

Bước 2. Ta áp dụng HÀM trả về vị từ đã viết VPD\_NHANVIEN vào trong bảng NHANVIEN của người dùng SIMPLEAC.

```
-----  
BEGIN  
  DBMS_RLS.ADD_POLICY  
  (  
    OBJECT_SCHEMA      => 'SIMPLEAC',  
    OBJECT_NAME        => 'NHANVIEN',  
    POLICY_NAME        => 'VPD_NHANVIEN_POLICY',  
    POLICY_FUNCTION     => 'VPD_NHANVIEN',  
    SEC_RELEVANT_COLS   => 'SALARY',  
    SEC_RELEVANT_COLS_OPT => DBMS_RLS.ALL_ROWS  
  );  
END;  
/  
-----
```



# CHÍNH SÁCH VPD MỘT

The screenshot shows the SQL Developer interface with the 'Query Result' window open. The query 'SELECT \* FROM SIMPLEAC.NHANVIEN;' is executed. The result table has 4 rows. The first row, corresponding to user NV01, shows a salary of 1. The other rows show salaries of 10, 100, and 1000.

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	100	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

The screenshot shows the SQL Developer interface with the 'Query Result' window open. The query 'SELECT \* FROM SIMPLEAC.NHANVIEN;' is executed. The result table has 4 rows. The first row, corresponding to user NV02, shows a salary of (null). The other rows show salaries of 10, (null), and (null).

	USERNAME	NAME	SALARY	EMAIL
1	NV01	PHAN THANH SANG	(null)	PTSANG@GMAIL.COM
2	NV02	HAI HONG	10	HHONG@GMAIL.COM
3	NV03	BA DUY	(null)	BDUY@GMAIL.COM
4	NV04	NGUYEN THE HIEN	(null)	NTHIEN@GMAIL.COM

Thông tin xem được của 2 người dùng NV01 và NV02 khác nhau dù cùng 1 truy vấn.





# CHÍNH SÁCH VPD MỘT

The screenshot shows the SQL Developer interface with a query window titled 'Query Builder'. The query is:

```
SELECT OBJECT_OWNER, OBJECT_NAME, POLICY, PREDICATE  
FROM V$VPD_POLICY;
```

Below the query, the 'Query Result' window shows the results of the query. The status bar indicates 'All Rows Fetched: 2 in 0.015 seconds'.

	OBJECT_OWNER	OBJECT_NAME	POLICY	PREDICATE
1	SIMPLEAC	NHANVIEN	VPD_NHANVIEN_POLICY	USERNAME = user OR ('NVKETOAN' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
2	SIMPLEAC	NHANVIEN	VPD_NHANVIEN_POLICY	(null)

Thông tin được xem qua truy vấn về VPD trên SYS.



## CHÍNH SÁCH VPD HAI

Bước 1. Ta tạo ra hàm trả về vị từ cho chính sách tên là VPD\_BENHNHAN

```
=====
Create or replace function VPD_BENHNHAN(schema varchar2,object varchar2)
    return varchar2
as
    user varchar2(100);
begin
    if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then
        return '';
    else
        user:= SYS_CONTEXT('userenv', 'SESSION_USER');
        return 'EMAIL_BACSI IN (SELECT EMAIL FROM SIMPLEAC.NHANVIEN WHERE USER = USERNAME)
                AND ('BACSI' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
                                WHERE NVVT.USERNAME = USER))';
    end if;
end;
/
=====
```



## CHÍNH SÁCH VPD HAI

Bước 1. Ta tạo ra hàm trả về vị từ cho chính sách tên là VPD\_BENHNHAN

```
if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then  
    return '';  
else
```

Nếu là người dung SIMPLEAC thì trả về vị từ là "" tương đương với ko có điều kiện khi select dòng và sẽ trả về toàn bộ bảng.



## CHÍNH SÁCH VPD HAI

Bước 1. Ta tạo ra hàm trả về vị từ cho chính sách tên là VPD\_BENHNHAN

```
else
  user:= SYS_CONTEXT('userenv', 'SESSION_USER');
  return 'EMAIL_BACSI IN (SELECT EMAIL FROM SIMPLEAC.NHANVIEN WHERE USER = USERNAME)
        AND ('BACSI' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
                          WHERE NVVT.USERNAME = USER))';
end if;
```

Ngắn gọn là xét một dòng bệnh nhân, nếu người đang đăng nhập có email giống như của bác sĩ quản lý bệnh nhân VÀ người đang đăng nhập có vai trò là BACSI  
=> Trả về dòng đó cho người đang đăng nhập.



# CHÍNH SÁCH VPD HAI

The screenshot displays the SQL Developer interface. The top toolbar includes icons for running queries, saving, and other database operations. The 'Query Builder' tab is active, showing a SQL query that filters data from the 'SIMPLEAC.BENHNHAN' table based on the user's session context. The query is as follows:

```
SELECT * FROM SIMPLEAC.BENHNHAN WHERE  
    EMAIL_BACSI IN (SELECT NV.EMAIL FROM SIMPLEAC.NHANVIEN NV WHERE SYS_CONTEXT('USERENV', 'SESSION_USER') = USERNAME)  
    AND ('BACSI' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT  
        WHERE NVVT.USERNAME = SYS_CONTEXT('USERENV', 'SESSION_USER')));
```

Below the query editor, the 'Query Result' window shows the execution status: 'All Rows Fetched: 0 in 0.09 seconds'. A table with three columns is visible:

ID_BENH...	EMAIL_BA...	BENH
------------	-------------	------

Kết quả tương tự như khi xài VPD cho user NV01.



# CHÍNH SÁCH VPD HAI

The screenshot shows the SQL Developer interface with the following components:

- Top Bar:** Contains connection tabs for SIMPLEACConn, NV01, NV02, and sysConn. Below the tabs is a toolbar with icons for running queries, saving, and other database operations.
- Worksheet:** The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT * FROM SIMPLEAC.BENHNHAN WHERE  
EMAIL_BACSI IN (SELECT NV.EMAIL FROM SIMPLEAC.NHANVIEN NV WHERE SYS_CONTEXT('USERENV', 'SESSION_USER') = USERNAME)  
AND ('BACSI' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT  
WHERE NVVT.USERNAME = SYS_CONTEXT('USERENV', 'SESSION_USER')));
```
- Query Result:** Below the query, the 'Query Result' tab shows the execution status: 'All Rows Fetched: 1 in 0.003 seconds'. The results are displayed in a table with three columns: ID\_BENHNHAN, EMAIL\_BACSI, and BENH.

	ID_BENHNHAN	EMAIL_BACSI	BENH
1	BN0001	HHONG@GMAIL.COM	SOI THAN

Kết quả tương tự như khi xài VPD cho user NV02.



## CHÍNH SÁCH VPD HAI

Bước 2. Ta áp dụng HÀM trả về vị từ đã viết VPD\_BENHNHAN vào trong bảng BENHNHAN của người dùng SIMPLEAC.

```
-----  
BEGIN  
  DBMS_RLS.ADD_POLICY  
  (  
    OBJECT_SCHEMA => 'SIMPLEAC',  
    OBJECT_NAME   => 'BENHNHAN',  
    POLICY_NAME   => 'VPD_BENHNHAN_POLICY',  
    POLICY_FUNCTION => 'VPD_BENHNHAN'  
  );  
END;  
/  
-----
```



# CHÍNH SÁCH VPD HAI

Worksheet | Query Builder

```
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Query Result x

SQL | All Rows Fetched: 0 in 0.078 seconds

ID_BENH...	EMAIL_BA...	BENH
------------	-------------	------

Worksheet | Query Builder

```
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Query Result x

SQL | All Rows Fetched: 1 in 0.003 seconds

ID_BENHNHAN	EMAIL_BACSI	BENH
1 BN0001	HHONG@GMAIL.COM	SOI THAN

Thông tin xem được của 2 người dùng NV01 và NV02 khác nhau dù cùng 1 truy vấn.





# CHÍNH SÁCH VPD HAI

SQL SIMPLEACConn x NV01 x NV02 x NV03 x sysConn x

0.038 seconds

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN;  
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Script Output x

Task completed in 0.038 seconds

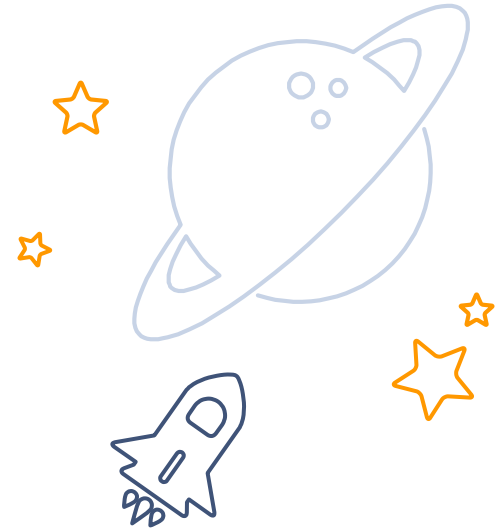
USERNAME	NAME	SALARY	EMAIL
NV01	PHAN THANH SANG		PTSANG@GMAIL.COM
NV02	HAI HONG		HHONG@GMAIL.COM
NV03	BA DUY	100	BDUY@GMAIL.COM
NV04	NGUYEN THE HIEN		NTHIEN@GMAIL.COM

ID_BENHNHAN	EMAIL_BACSI	BENH
BN0002	BDUY@GMAIL.COM	HIV

Thông tin được giới hạn như mong muốn qua 2 chính sách VPD.

# APPLICATION CONTEXT





## VẤN ĐỀ

- Mỗi người dung có 1 email, 1 username, 1 vai trò.
- Nhưng khi cần?

```
if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then
    return '';
else
    user:= SYS_CONTEXT('userenv', 'SESSION_USER');
    return 'USERNAME = user OR (''NVKETOAN'' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
    WHERE NVVT.USERNAME = USER));';
end if;
```



## VẤN ĐỀ

- Khó kiểm soát.
- Khó hiểu.
- Khó mở rộng.

```
if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then
    return '';
else
    user:= SYS_CONTEXT('userenv', 'SESSION_USER');
    return 'USERNAME = user' || OR ('NVKETOAN' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
    WHERE NVVT.USERNAME = USER));
end if;
```



## VẤN ĐỀ

```
if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then
    return '';
else
    user:= SYS_CONTEXT('userenv', 'SESSION_USER');
    return 'USERNAME = user OR ('NVKETOAN' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
    WHERE NVVT.USERNAME = USER))';
end if;
```

```
else
    user:= SYS_CONTEXT('userenv', 'SESSION_USER');
    return 'EMAIL_BACSI IN (SELECT EMAIL FROM SIMPLEAC.NHANVIEN WHERE USER = USERNAME)
    AND ('BACSI' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
    WHERE NVVT.USERNAME = USER))';
end if;
```



# VẤN ĐỀ

```
Create or replace function VPD_PHIEUKHAM(schema varchar2,object varchar2)
return varchar2
as
user varchar2(100);
begin
if (((SYS_CONTEXT('userenv', 'SESSION_USER'))='QUANLYBENHVIEN') OR
    ((SYS_CONTEXT('userenv', 'SESSION_USER'))='QUANLYCHINHSACH')) then
return '';
else
user:= SYS_CONTEXT('userenv', 'SESSION_USER');
return 'MABS = user OR (''QLCHUYENMON'' IN (SELECT TKVT.VAITRO FROM QUANLYBENHVIEN.TAIKHOAN_VAITRO TKVT
    WHERE TKVT.USERNAME = ' || 'user' || '))
OR (''TIEPTAN'' IN (SELECT TKVT.VAITRO FROM QUANLYBENHVIEN.TAIKHOAN_VAITRO TKVT
    WHERE TKVT.USERNAME = ' || 'user' || '))
    OR (''QLTAINGUYENNHANSU'' IN (SELECT TKVT.VAITRO FROM QUANLYBENHVIEN.TAIKHOAN_VAITRO TKVT
    WHERE TKVT.USERNAME = ' || 'user' || '))
    OR (''QLTAIVU'' IN (SELECT TKVT.VAITRO FROM QUANLYBENHVIEN.TAIKHOAN_VAITRO TKVT
    WHERE TKVT.USERNAME = ' || 'user' || '))');
end if;
end;
/
```



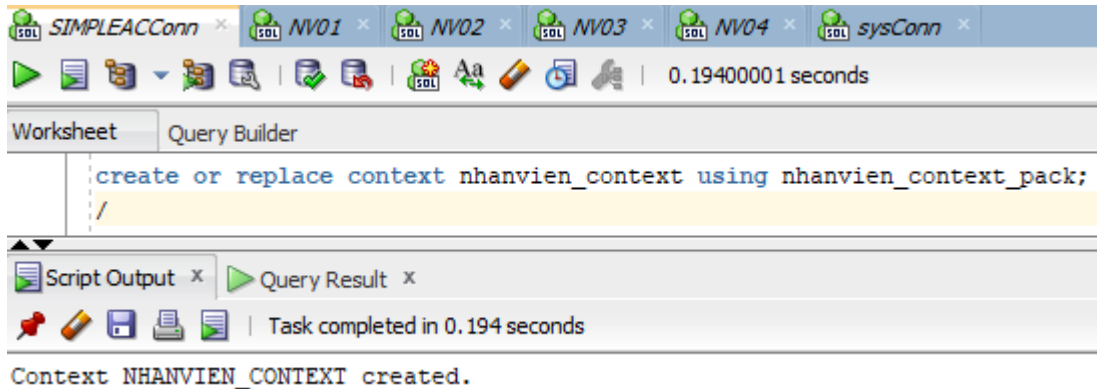
## VẤN ĐỀ

- Tạo 3 thuộc tính cho một tài khoản vaitro, email, username mỗi khi đăng nhập để xác định cho các chính sách điều khiển truy cập



# CÀI ĐẶT APPLICATION CONTEXT

## Bước 1: Tạo context trong một gói







# CÀI ĐẶT APPLICATION CONTEXT

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The main editor displays the following SQL code:

```
create or replace package nhanvien_context_pack
is
    procedure initialize;
end;
/
create or replace package body nhanvien_context_pack
is
    procedure initialize
    is
        t_vaitro nvarchar2(30);
        t_email nvarchar2(30);
        t_username nvarchar2(30);
    begin
        select NVVI.VAITRO, NV.EMAIL, NV.USERNAME
        into t_vaitro, t_email, t_username
        from SIMPLEAC.NHANVIEN NV, SIMPLEAC.NHANVIEN_VAITRO NVVT
        where NV.USERNAME = sys_context('USERENV', 'SESSION_USER') AND NV.USERNAME = NVVT.USERNAME;
        DBMS_SESSION.set_context('nhanvien_context', 'nv_vaitro', t_vaitro);
        DBMS_SESSION.set_context('nhanvien_context', 'nv_email', t_email);
        DBMS_SESSION.set_context('nhanvien_context', 'nv_username', t_username);
    end;
end;
```

The bottom status bar indicates: "Task completed in 1.048 seconds". Below the status bar, the following messages are displayed:

```
Package NHANVIEN_CONTEXT_PACK compiled
Package body NHANVIEN_CONTEXT_PACK compiled
```

Bước 2: tạo thủ tục khởi tạo giá trị trong gói đã tạo.



# CÀI ĐẶT APPLICATION CONTEXT

The screenshot shows the SQL Developer interface with several tabs at the top: SIMPLEACConn, NV01, NV02, NV03, NV04, and sysConn. The 'Query Builder' tab is active, displaying the following SQL query:

```
select NVVT.VAITRO, NV.EMAIL, NV.USERNAME
  from SIMPLEAC.NHANVIEN NV, SIMPLEAC.NHANVIEN_VAITRO NVVT
 where NV.USERNAME = sys_context('USERENV', 'SESSION_USER') AND NV.USERNAME = NVVT.USERNAME;
```

Below the query, the 'Query Result' tab shows the execution status: 'All Rows Fetched: 1 in 0.004 seconds'. The result is displayed in a table with three columns: VAITRO, EMAIL, and USERNAME.

	VAITRO	EMAIL	USERNAME
1	NVKETOAN	PTSANG@GMAIL.COM	NV01

Nguyên tắc là chọn ra 1 dòng chứa các dữ liệu cần đặt sau đó dùng lệnh `DBMS_SESSION.set_context('context_name', 'attribute', <giá trị sẽ đặt>)`



# CÀI ĐẶT APPLICATION CONTEXT

```
create or replace package body nhanvien_context_pack
is
  procedure initialize
  is
    t_vaitro nvarchar2(30);
    t_email nvarchar2(30);
    t_username nvarchar2(30);
  begin
    select NVVT.VAITRO, NV.EMAIL, NV.USERNAME
    into t_vaitro, t_email, t_username
    from SIMPLEAC.NHANVIEN NV, SIMPLEAC.NHANVIEN_VAITRO NVVT
    where NV.USERNAME = sys_context('USERENV', 'SESSION_USER') AND NV.USERNAME = NVVT.USERNAME;
    DBMS_SESSION.set_context('nhanvien_context', 'nv_vaitro', t_vaitro);
    DBMS_SESSION.set_context('nhanvien_context', 'nv_email', t_email);
    DBMS_SESSION.set_context('nhanvien_context', 'nv_username', t_username);
  end;
end;
/
```

Trong ví dụ là kết 2 bảng rồi cho các giá trị vaitro, email, username vào SET\_CONTEXT.



## CÀI ĐẶT APPLICATION CONTEXT

The screenshot shows the SQL Developer interface with the following components:

- Top Tab Bar:** Contains several tabs labeled `SIMPLEACConn`, `NV01`, `NV02`, `NV03`, `NV04`, and `sysConn`.
- Toolbar:** Includes icons for running queries, saving, and other database operations. A timer shows `0.141 seconds`.
- Worksheet:** The `Query Builder` tab is active, displaying the following SQL code:

```
create or replace trigger emp_logon
after logon on database
begin
    nhanvien_context_pack.initialize;
end;
/
```
- Bottom Panel:** Contains two tabs: `Script Output` and `Query Result`. The `Script Output` tab is active, showing the message: `Task completed in 0.141 seconds` and `Trigger EMP_LOGON compiled`.

Bước 3: tạo trigger chạy mỗi lần đăng nhập. Chạy thủ tục đã tạo trong packet để tạo ra 3 context cần dùng.



# CÀI ĐẶT APPLICATION CONTEXT

The screenshot shows the SQL Developer interface with a query window and a script output window. The query window contains three SQL statements to retrieve context values from the 'dual' table. The script output window shows the results of these queries, with the first result being 'NVKETOAN'.

```
select sys_context('nhanvien_context', 'nv_vaitro') from dual;  
select sys_context('nhanvien_context', 'nv_email') from dual;  
select sys_context('nhanvien_context', 'nv_username') from dual;
```

Script Output x

Task completed in 0.018 seconds

```
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_VAITRO')  
-----  
NVKETOAN  
  
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_EMAIL')  
-----  
PTSANG@GMAIL.COM  
  
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_USERNAME')  
-----  
NV01
```

Kết quả chúng ta có:

■ `sys_context('nhanvien_context', 'nv_vaitro')` =  
'NVKETOAN' có được từ  
bảng NHANVIEN\_VAITRO.

■ ...



# CÀI ĐẶT APPLICATION CONTEXT

SIMPLEACConn sysConn NV01 NV02 NV03 NV04

0.02 seconds

Worksheet Query Builder

```
select sys_context('nhanvien_context', 'nv_vaitro') from dual;  
select sys_context('nhanvien_context', 'nv_email') from dual;  
select sys_context('nhanvien_context', 'nv_username') from dual;
```

Script Output x

Task completed in 0.02 seconds

```
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_VAITRO')  
-----  
BACSI  
  
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_EMAIL')  
-----  
HHONG@GMAIL.COM  
  
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_USERNAME')  
-----  
NV02
```

SIMPLEACConn sysConn NV01 NV02 NV03 NV04

0.02 seconds

Worksheet Query Builder

```
select sys_context('nhanvien_context', 'nv_vaitro') from dual;  
select sys_context('nhanvien_context', 'nv_email') from dual;  
select sys_context('nhanvien_context', 'nv_username') from dual;
```

Script Output x

Task completed in 0.02 seconds

```
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_VAITRO')  
-----  
BACSI  
  
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_EMAIL')  
-----  
BDUY@GMAIL.COM  
  
SYS_CONTEXT('NHANVIEN_CONTEXT', 'NV_USERNAME')  
-----  
NV03
```



# CÀI ĐẶT APPLICATION CONTEXT

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The main editor displays the following SQL code:

```
CREATE OR REPLACE FUNCTION VPD_NHANVIEN(schema varchar2, object varchar2)
  RETURN VARCHAR2
AS
BEGIN
  IF ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') THEN
    RETURN '';
  ELSE
    RETURN '(USERNAME = sys_context(''nhanvien_context'', ''nv_username''))
    OR ''NVKETOAN'' = sys_context(''nhanvien_context'', ''nv_vaitro'')';
  END IF;
END;
```

The status bar at the bottom indicates 'Function VPD\_NHANVIEN compiled'.



# CÀI ĐẶT APPLICATION CONTEXT

The screenshot shows the SQL Developer interface with a query editor window. The title bar includes tabs for 'SIMPLEACConn', 'sysConn', and four 'NV' connections. The toolbar shows various SQL tools and a timer indicating '0.026 seconds'. The 'Worksheet' tab is active, displaying a PL/SQL function definition. The code defines two variables, 'nv\_vaitro' and 'nv\_email', and a function that checks the application context 'userenv' for 'SESSION\_USER'. If it is 'SIMPLEAC', it returns an empty string; otherwise, it returns a string containing 'EMAIL\_BACSI' and 'BACSI' values from the 'nhanvien\_context' schema. The bottom status bar indicates 'Task completed in 0.026 seconds' and 'Function VPD\_BENHNHAN compiled'.

```
as
  nv_vaitro nvarchar2(100);
  nv_email nvarchar2(100);
begin
  if ((SYS_CONTEXT('userenv', 'SESSION_USER'))='SIMPLEAC') then
    return '';
  else
    return 'EMAIL_BACSI = sys_context(''nhanvien_context'', 'nv_email')
           AND ''BACSI'' = sys_context(''nhanvien_context'', 'nv_vaitro')';
  end if;
end;
/
```

Script Output x Query Result x

Task completed in 0.026 seconds

Function VPD\_BENHNHAN compiled





# CÀI ĐẶT APPLICATION CONTEXT

SIMPLEACConn sysConn NV01 NV02 NV03 NV04

0.009 seconds

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN;  
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Script Output x

Task completed in 0.009 seconds

USERNAME	NAME	SALARY	EMAIL
NV01	PHAN THANH SANG	1	PTSANG@GMAIL.COM
NV02	HAI HONG	10	HHONG@GMAIL.COM
NV03	BA DUY	100	BDUY@GMAIL.COM
NV04	NGUYEN THE HIEN	1000	NTHIEN@GMAIL.COM

no rows selected

SIMPLEACConn sysConn NV01 NV02 NV03 NV04

0.014 seconds

Worksheet Query Builder

```
SELECT * FROM SIMPLEAC.NHANVIEN;  
SELECT * FROM SIMPLEAC.BENHNHAN;
```

Script Output x

Task completed in 0.014 seconds

USERNAME	NAME	SALARY	EMAIL
NV01	PHAN THANH SANG		PTSANG@GMAIL.COM
NV02	HAI HONG	10	HHONG@GMAIL.COM
NV03	BA DUY		BDUY@GMAIL.COM
NV04	NGUYEN THE HIEN		NTHIEN@GMAIL.COM

ID_BENHNHAN	EMAIL_BACSI	BENH
BN0001	HHONG@GMAIL.COM	SOI THAN



# CÀI ĐẶT APPLICATION CONTEXT

SQL Developer interface showing the Query Builder and Query Result tabs.

**Query Builder:**

```
SELECT OBJECT_OWNER, OBJECT_NAME, POLICY, PREDICATE
FROM V$VPD_POLICY;
SELECT * FROM V$VPD_POLICY;
```

**Query Result:**

Script Output x | Query Result x | All Rows Fetched: 2 in 0 seconds

	POLICY	PREDICATE
1	VPD_NHANVIEN_POLICY	(USERNAME = sys_context('nhanvien_context', 'nv_username')) OR 'NVKETOAN' = sys_context('nhanvien_context', 'nv_vaitro')
2	VPD_BENHNHAN_POLICY	EMAIL_BACSI = sys_context('nhanvien_context', 'nv_email') AND 'BACSI' = sys_context('nhanvien_context', 'nv_vaitro')



# CÀI ĐẶT APPLICATION CONTEXT

## ■ Trước khi cài: Logic cho VPD\_NHANVIEN

```
else
  user:= SYS_CONTEXT('userenv', 'SESSION_USER');
  return 'USERNAME = user      OR (''NVKETOAN'' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
                                     WHERE NVVT.USERNAME = USER))';
end if;
```

## ■ Sau khi cài Application Context: Logic cho VPD\_NHANVIEN

```
else
  return '(USERNAME = sys_context(''nhanvien_context'', 'nv_username'))
  OR ''NVKETOAN'' = sys_context(''nhanvien_context'', 'nv_vaitro'))';
end if;
```



# CÀI ĐẶT APPLICATION CONTEXT

## ■ Trước khi cài: Logic cho VPD\_BACSI

```
else
  user:= SYS_CONTEXT('userenv', 'SESSION_USER');
  return 'EMAIL_BACSI IN (SELECT EMAIL FROM SIMPLEAC.NHANVIEN WHERE USER = USERNAME)
  AND ('BACSI' IN (SELECT NVVT.VAITRO FROM SIMPLEAC.NHANVIEN_VAITRO NVVT
  WHERE NVVT.USERNAME = USER));
end if;
```

## ■ Sau khi cài Application Context: Logic cho VPD\_BACSI

```
else
  return 'EMAIL_BACSI = sys_context('nhanvien_context', 'nv_email')
  AND 'BACSI' = sys_context('nhanvien_context', 'nv_vaitro');
end if;
```



## CHIA SẺ CHO TẤT CẢ NGƯỜI DÙNG APPLICATION CONTEXT GLOBAL

- Ta có tài khoản NV01 không thể select job\_role từ context global\_kda\_ctx.

```
Enter user-name: NV01/PWNV01
Last Successful login time: Thu Dec 20 2018 00:12:57 +07:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> SELECT SYS_CONTEXT('global_kda_ctx', 'job_role') job_role FROM DUAL;

JOB_ROLE
-----
```



## CHIA SẺ CHO TẤT CẢ NGƯỜI DÙNG APPLICATION CONTEXT GLOBAL

- Đăng nhập tài khoản KDA để tạo Context globaly.

```
SQL> conn kda/123456
Connected.
SQL> create or replace context global_kda_ctx using kda_ctx_pkg ACCESSED GLOBALLY;

Context created.

SQL> CREATE OR REPLACE PACKAGE kda_ctx_pkg
2   AS
3   PROCEDURE set_kda_ctx(sec_level IN VARCHAR2);
4   PROCEDURE clear_kda_context;
5   END;
6   /

Package created.
```



# CHIA SẺ CHO TẤT CẢ NGƯỜI DÙNG APPLICATION CONTEXT GLOBAL

```
SQL> CREATE OR REPLACE PACKAGE BODY kda_ctx_pkg
 2  AS
 3  PROCEDURE set_kda_ctx(sec_level IN VARCHAR2)
 4  AS
 5  BEGIN
 6      DBMS_SESSION.SET_CONTEXT(
 7          namespace => 'global_kda_ctx',
 8          attribute => 'job_role',
 9          value      => sec_level);
10  END set_kda_ctx;
11  PROCEDURE clear_kda_context
12  AS
13  BEGIN
14      DBMS_SESSION.CLEAR_CONTEXT(
15          namespace      => 'global_kda_ctx',
16          attribute       => 'job_role');
17  END clear_kda_context;
18  END;
19  /
```

Package body created.

- Tạo phần thân package trả về bất cứ sec\_level nào mà KDA muốn cài đặt.



## CHIA SẺ CHO TẤT CẢ NGƯỜI DÙNG APPLICATION CONTEXT GLOBAL

- Người dùng KDA đặt mức sec\_level ở mức NHANVIEN.

```
SQL> BEGIN
2   kda_ctx_pkg.set_kda_ctx('NHANVIEN');
3   END;
4   /
```

PL/SQL procedure successfully completed.





## CHIA SẺ CHO TẤT CẢ NGƯỜI DÙNG APPLICATION CONTEXT GLOBAL

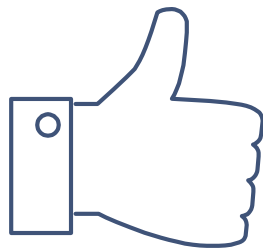
- Những người dùng đều có thể truy cập context global\_kda\_ctx với attribute là NHANVIEN (do người dùng KDA đặt).

```
SQL> CONN NV01/PWNV01
Connected.
SQL> SELECT SYS_CONTEXT('global_kda_ctx', 'job_role') job_role FROM DUAL;

JOB_ROLE
-----
NHANVIEN

SQL> CONN NV02/PWNV02
Connected.
SQL> SELECT SYS_CONTEXT('global_kda_ctx', 'job_role') job_role FROM DUAL;

JOB_ROLE
-----
NHANVIEN
```



# THANKS!

**Any questions?**

[1512454@student.hcmus.edu.vn](mailto:1512454@student.hcmus.edu.vn)

**038 608 5765**