# I. Lý thuyết

## 1. Quyền EXEMPT ACCESS POLICY

|  |
| --- |
| CREATE OR REPLACE FUNCTION no\_records (p\_schema IN VARCHAR2 DEFAULT NULL, p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS  BEGIN  RETURN '1=0';  END; |

A screenshot of a computer

Description automatically generated

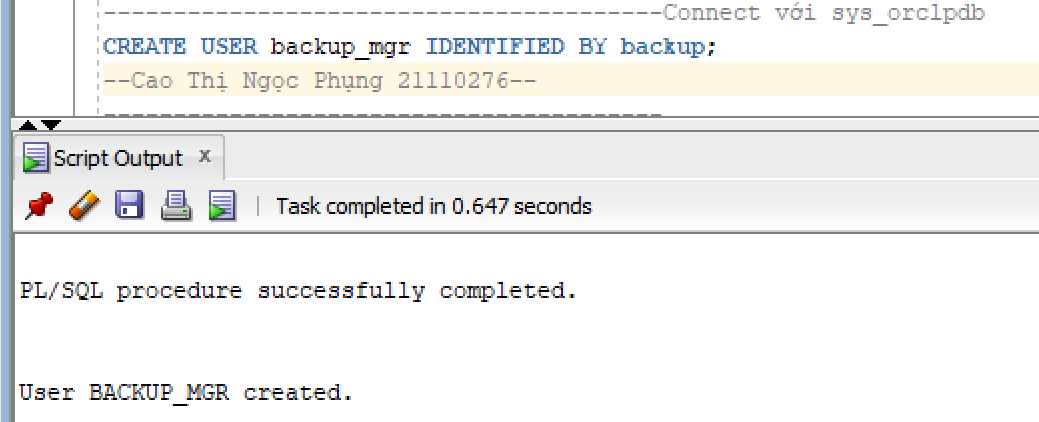
|  |
| --- |
| BEGIN  DBMS\_RLS.add\_policy  (object\_schema => 'SCOTT',  object\_name => 'EMP',  policy\_name => 'NOT\_DELETE',  function\_schema => 'SEC\_MGR',  policy\_function => 'No\_Records',  statement\_types => 'DELETE');  END; |

A screenshot of a computer

Description automatically generated

Tạo user

|  |
| --- |
| CREATE USER backup\_mgr IDENTIFIED BY backup; |



|  |
| --- |
| GRANT dba TO backup\_mgr IDENTIFIED BY backup; |

A screenshot of a computer error

Description automatically generated

Kết nối với scott\_orclpdb bằng user backup\_mgr

A screenshot of a computer error

Description automatically generated

Do BACKUP\_MGR bị ảnh hưởng bởi các policy function nên user này không xóa được record nào

|  |
| --- |
| DELETE FROM scott.emp; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| GRANT EXEMPT ACCESS POLICY TO backup\_mgr; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| DELETE FROM scott.emp; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| ROLLBACK; |

A screenshot of a computer

Description automatically generated

## 2. Giám sát quyền EXEMPT ACCESS POLICY

|  |
| --- |
| SELECT grantee FROM dba\_sys\_privs  WHERE PRIVILEGE = 'EXEMPT ACCESS POLICY'; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| AUDIT EXEMPT ACCESS POLICY BY ACCESS |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| DELETE FROM scott.emp;  ROLLBACK;  SET SERVEROUTPUT ON; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| SELECT \* FROM DBA\_AUDIT\_TRAIL WHERE USERNAME = 'HAN' AND ACTION\_NAME = 'DELETE'; SELECT \* FROM DBA\_AUDIT\_TRAIL WHERE USERNAME = 'HAN' AND ACTION\_NAME = 'DELETE'; |

A screenshot of a computer

Description automatically generated

## 3. Xử lý lỗi cho Policy Function

Để minh họa cho lỗi của một policy function, trước tiên ta tạo một bảng sẽ được truy xuất bởi policy function

|  |
| --- |
| CREATE TABLE test(id int); |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| INSERT INTO test VALUES(1); |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| CREATE PUBLIC SYNONYM testTable FOR test; |

A screenshot of a computer

Description automatically generated

Tạo ra policy function có lệnh truy xuất đến bảng TEST:

|  |
| --- |
| CREATE OR REPLACE FUNCTION pred\_function (p\_schema IN VARCHAR2 DEFAULT NULL, p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS total NUMBER;  BEGIN  SELECT COUNT (\*) INTO total FROM testTable;  RETURN '1 <= ' || total;  END; |

A screenshot of a computer

Description automatically generated

Gán policy function trên cho bảng DEPT của SCOTT

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| SELECT COUNT(\*) FROM dept; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| DROP TABLE test; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| SELECT COUNT(\*) FROM dept; |

A screenshot of a computer

Description automatically generated

Để sửa lỗi trên ta chỉ cần recover lại việc xóa bảng bằng lệnh Flashback Drop. Khi đó việc thực thi của chính sách bảo mật cũng sẽ được phục hồi

|  |
| --- |
| FLASHBACK TABLE test TO BEFORE DROP; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| SELECT COUNT(\*) FROM dept; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| CREATE OR REPLACE FUNCTION pred\_function (p\_schema IN VARCHAR2 DEFAULT NULL, p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS total NUMBER;  BEGIN  EXECUTE IMMEDIATE 'SELECT COUNT (\*) FROM testTable'  INTO total;  RETURN '1 <= ' || total;  EXCEPTION  WHEN OTHERS THEN RETURN '1 = 0';  END; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| SELECT COUNT(\*) FROM dept; |

A screenshot of a computer

Description automatically generated

## 4. Column Sensitive VPD

|  |
| --- |
| CREATE OR REPLACE FUNCTION user\_only (p\_schema IN VARCHAR2 DEFAULT NULL, p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS  BEGIN  RETURN 'ename = user';  END; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| BEGIN  DBMS\_RLS.add\_policy  (object\_schema => 'SCOTT',  object\_name => 'EMP',  policy\_name => 'people\_sel\_sal',  function\_schema => 'SEC\_MGR',  policy\_function => 'user\_only',  statement\_types => 'SELECT',  sec\_relevant\_cols => 'SAL,HIREDATE');  END; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| SELECT ename,job  FROM emp  WHERE ename >= 'S'; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| SELECT ename,sal  FROM emp; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| SELECT ename,hiredate  FROM emp; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| SELECT ename,hiredate,sal  FROM emp; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| SELECT ename  FROM scott.emp  WHERE ename >= 'S' AND sal > 1000; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| SELECT ename  FROM emp  WHERE ename >= 'S' AND sal > 1000; |

**A screenshot of a computer

Description automatically generated**

|  |
| --- |
| BEGIN  -- Xóa chính sách hiện tại  DBMS\_RLS.drop\_policy  (object\_schema => 'SCOTT',  object\_name => 'EMP',  policy\_name => 'people\_sel\_sal');  -- Tạo lại chính sách với thay đổi ở  -- tham số SEC\_RELEVANT\_COLS\_OPT  DBMS\_RLS.add\_policy  (object\_schema => 'SCOTT',  object\_name => 'EMP',  policy\_name => 'people\_sel\_sal',  function\_schema => 'SEC\_MGR',  policy\_function => 'user\_only',  statement\_types => 'SELECT',  sec\_relevant\_cols => 'SAL,HIREDATE',  sec\_relevant\_cols\_opt => DBMS\_RLS.all\_rows);  END; |

**A screenshot of a computer

Description automatically generated**

|  |
| --- |
| SELECT ename,job,sal,hiredate  FROM emp  WHERE ename >= 'S' ; |

**A screenshot of a computer

Description automatically generated**

# II. Bài tập

## 1. Hoàn thiện chính sách bảo mật ở bài thực hành số 4 (HolidayControl)

|  |
| --- |
| CREATE OR REPLACE FUNCTION holiday\_control  (p\_schema IN VARCHAR2 DEFAULT NULL,  p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS  BEGIN  RETURN 'Name = USER'; -- Chỉ cho phép xem thông tin của chính nhân viên  EXCEPTION -- Khi xảy ra lỗi, trả về một điều kiện an toàn để bảo vệ thông tin  WHEN OTHERS THEN  RETURN '1 = 0';  END; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| BEGIN  DBMS\_RLS.add\_policy(  object\_schema => 'SCOTT',  object\_name => 'EMPHOLIDAY',  policy\_name => 'holiday\_control\_policy',  function\_schema => 'SEC\_MGR',  policy\_function => 'holiday\_control',  statement\_types => 'SELECT');  END; |

A screenshot of a computer

Description automatically generated

## 2. Chỉnh sửa lại chính sách bảo mật ở câu 1

Cho phép An xem được thông tin EmpNo và Name của các nhân viên khác trong bảng EmpHoliday nhưng chỉ xem được ngày nghỉ của chính mình.

|  |
| --- |
| CREATE OR REPLACE FUNCTION holiday\_control(  p\_schema IN VARCHAR2 DEFAULT NULL,  p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS  BEGIN  IF SYS\_CONTEXT('USERENV', 'SESSION\_USER') = 'AN' THEN  RETURN 'Name = USER'; -- Chỉ cho phép "An" xem thông tin của chính mình  ELSE  RETURN '1 = 0'; -- Không cho phép người dùng khác xem bất kỳ thông tin nào  END IF;  EXCEPTION -- Khi xảy ra lỗi, trả về một điều kiện an toàn để bảo vệ thông tin  WHEN OTHERS THEN  RETURN '1 = 0';  END; |

A screenshot of a computer

Description automatically generated

Thêm sec\_relevant\_cols => 'Holiday'

|  |
| --- |
| BEGIN  -- Xóa chính sách cũ nếu đã tồn tại  DBMS\_RLS.drop\_policy(  object\_schema => 'SCOTT',  object\_name => 'EMPHOLIDAY',  policy\_name => 'holiday\_control\_policy');  -- Thêm chính sách mới  DBMS\_RLS.add\_policy(  object\_schema => 'SCOTT',  object\_name => 'EMPHOLIDAY',  policy\_name => 'holiday\_control\_policy',  function\_schema => 'SEC\_MGR', -- Đảm bảo schema chứa function là đúng  policy\_function => 'holiday\_control',  statement\_types => 'SELECT', -- Áp dụng cho các câu lệnh SELECT  sec\_relevant\_cols => 'Holiday');  END; |

A screenshot of a computer program

Description automatically generated

Đăng nhập với user An để kiểm tra ta có thể thấy An thấy đc số và tên của mọi người nhưng chỉ thấy đc holiday của bản thân

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Đăng nhập với user Han để kiểm tra, ta thấy HAN có thể thấy hết thông tin của mọi người

A screenshot of a computer

Description automatically generated

## 3. Từ chính sách HolidayControl ở câu 2

Thiết lập quyền không ảnh hưởng và đảm bảo sự giám sát đối với chính sách này cho user Han.

|  |
| --- |
| GRANT EXEMPT ACCESS POLICY TO HAN; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| AUDIT EXEMPT ACCESS POLICY BY ACCESS; |

A computer screen shot of a computer

Description automatically generated

Kết nối với user Han và xóa bảng

|  |
| --- |
| DELETE FROM scott.EMPHOLIDAY; |

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

## 4. Cho bảng có cấu trúc như sau thuộc schema của sec\_mgr

**Tạo bảng Employee (empno, ename, email, salary, deptno)**

|  |
| --- |
| CREATE TABLE Employee (  empno NUMBER PRIMARY KEY,  ename VARCHAR2(100),  email VARCHAR2(100),  salary NUMBER,  deptno NUMBER); |

A screenshot of a computer

Description automatically generated

Thêm dữ liệu vào bảng

|  |
| --- |
| INSERT INTO EMPLOYEE (empno, ename, email, salary, deptno) VALUES (1, 'PHUNG', 'phung@gmail.com', 50000, 10);  INSERT INTO EMPLOYEE (empno, ename, email, salary, deptno) VALUES (2, 'TIEN', 'tien@gmail.com', 55000, 20);  INSERT INTO EMPLOYEE (empno, ename, email, salary, deptno) VALUES (3, 'BACH', 'bach@gmail.com', 60000, 10);  INSERT INTO EMPLOYEE (empno, ename, email, salary, deptno) VALUES (4, 'THUONG', 'thuong@gmail.com', 65000, 30);  INSERT INTO EMPLOYEE (empno, ename, email, salary, deptno) VALUES (5, 'KHANG', 'khang@gmail.com', 70000, 20); |

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Thêm user tương ứng và gán quyền select và update

|  |
| --- |
| CREATE USER PHUNG IDENTIFIED BY phung;  CREATE USER TIEN IDENTIFIED BY tien;  CREATE USER BACH IDENTIFIED BY bach;  CREATE USER THUONG IDENTIFIED BY thuong;  CREATE USER KHANG IDENTIFIED BY khang;  GRANT CONNECT TO PHUNG;  GRANT CONNECT TO TIEN;  GRANT CONNECT TO BACH;  GRANT CONNECT TO THUONG;  GRANT CONNECT TO KHANG;  GRANT SELECT, UPDATE ON EMPLOYEE TO PHUNG;  GRANT SELECT, UPDATE ON EMPLOYEE TO TIEN;  GRANT SELECT, UPDATE ON EMPLOYEE TO BACH;  GRANT SELECT, UPDATE ON EMPLOYEE TO THUONG;  GRANT SELECT, UPDATE ON EMPLOYEE TO KHANG; |

A screenshot of a computer

Description automatically generated

Tạo function lấy phòng ban hiện tại của user

|  |
| --- |
| CREATE OR REPLACE FUNCTION select\_policy (  p\_schema IN VARCHAR2 DEFAULT NULL,  p\_object IN VARCHAR2 DEFAULT NULL  )  RETURN VARCHAR2  AS user\_deptno sec\_mgr.EMPLOYEE.deptno%TYPE;  BEGIN  SELECT deptno INTO user\_deptno  FROM sec\_mgr.EMPLOYEE  WHERE ename = USER;  RETURN 'deptno = ' || user\_deptno;  EXCEPTION  WHEN OTHERS THEN  RETURN '1 = 0'; -- Trả về điều kiện không hợp lệ khi có lỗi xảy ra  END; |

**A screenshot of a computer

Description automatically generated**

**Thêm policy**

|  |
| --- |
| BEGIN  DBMS\_RLS.DROP\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE',  policy\_name => 'select\_employee\_policy');  DBMS\_RLS.ADD\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE',  policy\_name => 'select\_employee\_policy',  function\_schema => 'sec\_mgr',  policy\_function => 'select\_policy',  statement\_types => 'SELECT',  sec\_relevant\_cols => 'email, salary');  END; |

**A screenshot of a computer program

Description automatically generated**

**Kiểm tra sẽ thấy nhân viên của phòng ban đó chỉ thông tin các nhân viên phòng cùng phòng ban**

**A screenshot of a computer

Description automatically generated**

**Tạo function chỉ có user mới thay đổi email của mình**

|  |
| --- |
| CREATE OR REPLACE FUNCTION update\_policy (  p\_schema IN VARCHAR2 DEFAULT NULL,  p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS  BEGIN  RETURN 'ename = USER';  EXCEPTION  WHEN OTHERS THEN  RETURN '1 = 0';  END; |

**A screenshot of a computer

Description automatically generated**

|  |
| --- |
| BEGIN  DBMS\_RLS.DROP\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE',  policy\_name => 'update\_employee\_policy'  );  DBMS\_RLS.ADD\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE',  policy\_name => 'update\_employee\_policy',  function\_schema => 'sec\_mgr',  policy\_function => 'update\_policy',  statement\_types => 'UPDATE',  sec\_relevant\_cols => 'empno, ename, salary, deptno');  END; |

**A screenshot of a computer

Description automatically generated**

**User Phung không thể thay đổi email của BACH**

**A screenshot of a computer

Description automatically generated**

**Chỉ có USER update được mail của bản thân**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

## 5. Cho bảng có cấu trúc như sau thuộc schema của sec\_manager:

|  |
| --- |
| CREATE TABLE EMPLOYEE\_NEXT (  EMPNO NUMBER PRIMARY KEY,  ENAME VARCHAR2(50),  EMAIL VARCHAR2(100),  SALARY NUMBER,  DEPTNO NUMBER,  MANAGER NUMBER); |

A screenshot of a computer

Description automatically generated

Thêm dữ liệu vào bảng

|  |
| --- |
| INSERT INTO EMPLOYEE\_NEXT (EMPNO, ENAME, EMAIL, SALARY, DEPTNO, MANAGER) VALUES (1, 'PHUNG', 'phung@gmail.com', 50000, 10, NULL);  INSERT INTO EMPLOYEE\_NEXT (EMPNO, ENAME, EMAIL, SALARY, DEPTNO, MANAGER) VALUES (2, 'TIEN', 'tien@gmail.com', 55000, 10, 1);  INSERT INTO EMPLOYEE\_NEXT (EMPNO, ENAME, EMAIL, SALARY, DEPTNO, MANAGER) VALUES (3, 'BACH', 'bach@gmail.com', 60000, 20, 5);  INSERT INTO EMPLOYEE\_NEXT (EMPNO, ENAME, EMAIL, SALARY, DEPTNO, MANAGER) VALUES (4, 'THUONG', 'thuong@gmail.com', 65000, 30, NULL);  INSERT INTO EMPLOYEE\_NEXT (EMPNO, ENAME, EMAIL, SALARY, DEPTNO, MANAGER) VALUES (5, 'KHANG', 'khang@gmail.com', 70000, 20, NULL); |

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Gán quyền select cho nhân viên, số 1,5 là manager nên gán thêm quyền insert, update, delete cho 2 user đó

|  |
| --- |
| GRANT SELECT, INSERT, UPDATE, DELETE ON EMPLOYEE\_NEXT TO PHUNG;  GRANT SELECT ON EMPLOYEE\_NEXT TO TIEN;  GRANT SELECT ON EMPLOYEE\_NEXT TO BACH;  GRANT SELECT ON EMPLOYEE\_NEXT TO THUONG;  GRANT SELECT, INSERT, UPDATE, DELETE ON EMPLOYEE\_NEXT TO KHANG; |

A screenshot of a computer

Description automatically generated

Chỉ xem đc

|  |
| --- |
| CREATE OR REPLACE FUNCTION sec\_mgr.select\_policy\_next (  p\_schema IN VARCHAR2,  p\_object IN VARCHAR2  ) RETURN VARCHAR2  AS  user\_deptno sec\_mgr.EMPLOYEE\_NEXT.deptno%TYPE;  BEGIN  SELECT deptno INTO user\_deptno  FROM sec\_mgr.EMPLOYEE\_NEXT  WHERE ename = USER;  RETURN 'deptno = ' || user\_deptno;  EXCEPTION  WHEN OTHERS THEN  RETURN '1 = 0';  END; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| BEGIN  DBMS\_RLS.DROP\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'select\_employee\_next\_policy');  DBMS\_RLS.ADD\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'select\_employee\_next\_policy',  function\_schema => 'sec\_mgr',  policy\_function => 'select\_policy\_next',  statement\_types => 'SELECT',  sec\_relevant\_cols => 'SALARY');  END; |

Quản lý có thể update, insert, delete các user trong phòng ban

|  |
| --- |
| CREATE OR REPLACE FUNCTION sec\_mgr.manager\_policy (  p\_schema IN VARCHAR2,  p\_object IN VARCHAR2  ) RETURN VARCHAR2  AS  user\_deptno sec\_mgr.EMPLOYEE\_NEXT.deptno%TYPE;  is\_manager NUMBER(1);  BEGIN  SELECT deptno,  CASE  WHEN manager IS NULL THEN 1  ELSE 0  END  INTO user\_deptno, is\_manager  FROM sec\_mgr.EMPLOYEE\_NEXT  WHERE ename = USER;  IF is\_manager = 1 THEN  RETURN 'deptno = ' || user\_deptno;  ELSE  RETURN '1 = 0';  END IF;  EXCEPTION  WHEN OTHERS THEN  RETURN '1 = 0';  END; |

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| BEGIN  DBMS\_RLS.DROP\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'manager\_policy\_function');  DBMS\_RLS.ADD\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'manager\_policy\_function',  function\_schema => 'sec\_mgr',  policy\_function => 'manager\_policy',  statement\_types => 'INSERT,UPDATE,DELETE',  update\_check => TRUE);  END; |

A screenshot of a computer

Description automatically generated

Các nhân viên khác có thể select (empno, ename, email, deptno, manager) trừ salary của nhân viên cùng phòng ban

|  |
| --- |
| CREATE OR REPLACE FUNCTION sec\_mgr.select\_policy\_next (  p\_schema IN VARCHAR2,  p\_object IN VARCHAR2  ) RETURN VARCHAR2  AS  user\_deptno sec\_mgr.EMPLOYEE\_NEXT.deptno%TYPE;  BEGIN  SELECT deptno INTO user\_deptno  FROM sec\_mgr.EMPLOYEE\_NEXT  WHERE ename = USER;  RETURN 'deptno = ' || user\_deptno;  EXCEPTION  WHEN OTHERS THEN  RETURN '1 = 0';  END; |

A screenshot of a computer program

Description automatically generated

Thêm policy

|  |
| --- |
| BEGIN DBMS\_RLS.DROP\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'select\_policy\_function\_next');  DBMS\_RLS.ADD\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'select\_policy\_function\_next',  function\_schema => 'sec\_mgr',  policy\_function => 'select\_policy\_next',  statement\_types => 'SELECT',  sec\_relevant\_cols => 'empno, ename, email, deptno, manager');  END; |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| CREATE OR REPLACE FUNCTION salary\_policy (  p\_schema IN VARCHAR2 DEFAULT NULL,  p\_object IN VARCHAR2 DEFAULT NULL)  RETURN VARCHAR2  AS  BEGIN  RETURN 'ename = USER';  EXCEPTION  WHEN OTHERS THEN  RETURN '1 = 0';  END; |

A screenshot of a computer

Description automatically generated

Thêm policy

|  |
| --- |
| BEGIN  DBMS\_RLS.DROP\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'salary\_policy\_function'  );  DBMS\_RLS.ADD\_POLICY(  object\_schema => 'sec\_mgr',  object\_name => 'EMPLOYEE\_NEXT',  policy\_name => 'salary\_policy\_function',  function\_schema => 'sec\_mgr',  policy\_function => 'salary\_policy',  statement\_types => 'SELECT',  sec\_relevant\_cols => 'salary');  END; |

A screenshot of a computer program

Description automatically generated

User chỉ có thể xem salary của chính mình

A screenshot of a computer

Description automatically generated