```
TO_CHAR(last_analyzed, 'YYYY') = '2024'
alter session set "_oracle_script" = true;
create user USERNAME identified by PASSWORD;
grant create session, DBA, .. to USERNAME;
-- 1b)Phân quyền đảm bảo các user này có thể tạo bất kỳ bảng nào trong tablespace với quota 10M
GRANT CREATE TABLE TO John;
ALTER USER John QUOTA 10M ON USERS;
```

```
create or replace view uv_NhanVienCoBan_NHANSU
as
    select *
    from NHANSU
    where MANV = sys_context('userenv', 'session_user')
    with check option;
```

```
create or replace procedure usp_createusernhanvien
as
    cursor cur is (
        select MaNhanVien
        from NHANVIEN
        where MaNhanVien not in (select username from all_users)
    );
    usr char(4);
    strsql varchar2(1000);
begin
    open cur;
    strsql := 'alter session set "_oracle_script" = true';
    execute immediate strsql;

    loop
        fetch cur into usr;
        exit when cur%notfound;

        strsql := 'create user ' || usr || ' identified by ' || usr;
        execute immediate strsql;
    end loop;

    strsql := 'alter session set "_oracle_script" = false';
    execute immediate strsql;

    close cur;
end;
/
```

```
-- 5b) Cho phép user John quyền cấp quyền cho các user khác
grant grant any privilege to John;

-- 5c) Gán tất cả các quyền mà user John có cho user Beth.
conn John/JOHN;
set role DataEntry identified by mgt;
declare
   v_sql varchar2(4000);
begin
   for res in (select privilege from USER_SYS_PRIVS where username = 'JOHN')
   loop
      v_sql := 'grant ' || res.privilege || ' to Beth';
      execute immediate v_sql;
   end loop;
end;
/
```

```
-- Câu 2: Viết lệnh script để tạo ra các tài khoảng Oracle DB tương ứng với
create or replace procedure usp_createAccount
as
    cursor cur is (
        select MaNV
        from NHANVIEN
        where MaNV not in (select username from all_users)
    );
    usr char(5);
    strsql varchar2(1000);
begin
    open cur;
    strsql := 'alter session set "_oracle_script" = true';
    execute immediate strsql;
    loop
        fetch cur into usr;
        exit when cur%notfound;

        strsql := 'create user ' || usr || ' identified by ' || upper(usr);
        execute immediate strsql;
    end loop;
    strsql := 'alter session set "_oracle_script" = false';
    execute immediate strsql;
    close cur;
end;
/
```

*3.2.2 Auditing trên các lệnh PL/SQL và đối tượng dữ liệu*

Người quả trị có thể audit các hành động phổ biến trên các đối tượng dữ liệu bảng, view, procedure, trigger, function, sequence,… như CREATE, AUDIT, INSERT,… Danh sách đầy đủ các hành động có thể được audit có thể xem thêm tại link sau:

http://docs.oracle.com/cd/E16655_01/network.121/e17607/audit_config.htm#CHDFACCE

Ví dụ sau thực hiện audit trên thao tác SELECT trong bảng SYS.USER$.

```
CREATE AUDIT POLICY select_user_dictionary_table_pol ACTIONS SELECT
ON SYS.USER$;

AUDIT POLICY select_user_dictionary_table_pol;
```

Bên cạnh đó, một audit policy có thể dùng để audit cùng lúc nhiều hành động khác nhau trên một đối tượng dữ liệu

```
CREATE AUDIT POLICY actions_on_hr_emp_pol1
 ACTIONS EXECUTE, GRANT
 ON app_lib;

AUDIT POLICY actions_on_hr_emp_pol1 BY jrandolph, phawkins;
```

Hoặc kết hợp audit các quyền và các hành động trên đối tượng dữ liệu

```
CREATE AUDIT POLICY tab_pol
 PRIVILEGES CREATE ANY TABLE
 ACTIONS CREATE TABLE;

AUDIT tab_pol BY jackson;
```

Trong trường hợp muốn audit tất cả các hành động trên một bảng dữ liệu, người dùng có thể sử dụng từ khoá ALL như trong ví dụ sau:

```
CREATE AUDIT POLICY all_actions_on_hr_emp_pol
 ACTIONS ALL ON HR.EMPLOYEES;
```

Ngoài ra, người dùng cũng có thể thêm các điều kiện vào trong audit policy.

```
CREATE AUDIT POLICY os_users_priv_pol
 PRIVILEGES SELECT ANY TABLE, CREATE LIBRARY
 WHEN 'SYS_CONTEXT (''USERENV'', ''OS_USER'') IN (''psmith'',
''jrawlins'')'
 EVALUATE PER SESSION;

AUDIT POLICY os_users_priv_pol;
```



```
strsql := 'AUDIT ALL BY ' || usr || ' BY ACCESS';
execute immediate (strsql);

strsql := 'AUDIT SELECT TABLE, UPDATE TABLE, INSERT TABLE, DELETE TABLE BY ' || usr || ' BY ACCESS';
execute immediate (strsql);
```

```
create or replace trigger utrig_GiaoVu_KHMO
instead of insert or update on uv_NhanVienC
for each row
begin
    if (inserting) then
        insert into KHMO values (:new."MA H
    elsif (updating) then
        update KHMO set
        MAHP = :new."MA HOC PHAN",
        HK = :new."HOC KY",
        NAM = :new.NAM,
        MACT = :new."MA CHUONG TRINH"
        where MAHP = :old."MA HOC PHAN"
        and HK = :old."HOC KY"
        and NAM = :old.NAM
        and MACT = :old."MA CHUONG TRINH";
    end if;
end;
/
```

```
elsif (deleting) then
    delete from DANGKY
    where MASV = :old.MASV
    and MAGV = :old.MAGV
    and MAHP = :old.MAHP
    and HK = :old.HK
    and NAM = :old.NAM
    and MACT = :old.MACT;
end if;
```

```sql
CREATE OR REPLACE FUNCTION no_dept10
 p_schema  IN  VARCHAR2,
 p_object  IN  VARCHAR2)
 RETURN VARCHAR2
 AS
BEGIN
 RETURN 'deptno != 10';
END;
```

```sql
BEGIN DBMS_RLS.add_policy
(object_schema      => 'SCOTT',
 object_name        => 'EMP',
 policy_name        => 'quickstart',
 policy_function    => 'no_dept10');
END;
```

```sql
CREATE OR REPLACE FUNCTION dept_less_4 (
 p_schema  IN  VARCHAR2 DEFAULT NULL,
 p_object  IN  VARCHAR2 DEFAULT NULL)
 RETURN VARCHAR2
 AS
BEGIN
 RETURN 'deptno < 4';
END;

BEGIN DBMS_RLS.add_policy
(object_schema      => 'SCOTT',
 object_name        => 'EMP',
 policy_name        => 'EMP_IU',
 function_schema    => 'SEC_MGR',
 policy_function    => 'dept_less_4',
 statement_types    => 'INSERT,UPDATE',
 update_check       => TRUE);
END;
```

```sql
   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      => 'SALARY',
         sec_relevant_cols_opt  =>
DBMS_RLS.all_rows);
   END;
```

```sql
-- 4b
begin
    dbms_fga.add_policy(
        object_schema   => 'ADMIN',
        object_name     => 'NHANVIEN',
        policy_name     => 'FGA_POLICY_LUONGPHUCAP',
        audit_condition => 'MANV <> sys_context(''USERENV'', ''SESSION_USER'')',
        audit_column    => 'LUONG, PHUCAP',
        statement_types => 'select'
    );
end;
/
```

```sql
-- 4c
--Viết function 'f_CheckRLTaiChinh' trả về vị từ là 1
begin
    dbms_fga.add_policy(
        object_schema   => 'ADMIN',
        object_name     => 'NHANVIEN',
        policy_name     => 'FGA_POLICY_LUONGPHUCAP',
        audit_condition => 'f_CheckRLTaiChinh = 0',
        audit_column    => 'LUONG, PHUCAP',
        statement_types => 'update'
    );
end;
/
```

- Enable chính sách.
- Xem nhật ký trên bảng DBA_FGA_AUDIT_TRAIL.

```sql
-- Thêm role tương ứng cho user nhân viên
create or replace procedure sp_AddRoleNV
    (strrole varchar2, loainv nvarchar2)
authid current_user
as
    cursor cur is (
        select MANV
        from NHANSU
        where MANV in (select username from all_users) and VAITRO = loainv
    );
    strsql varchar2(1000);
    usr varchar2(100);
begin
    open cur;
    loop
        fetch cur into usr;
        exit when cur%notfound;

        strsql := 'grant '|| strrole || ' to ' || usr;
        execute immediate (strsql);
    end loop;
    close cur;
end;
/
```

u1: 01 giám đốc có thể đọc được toàn bộ dữ liệu (GD:MB,SX,GC:B,T,N)

u2: 01 trưởng phòng phụ trách lĩnh vực sản xuất miền Nam (TP:SX:N)

u3: 01 giám đốc phụ trách cả 3 lĩnh vực ở chi nhánh miền Bắc (có thể đọc được toàn bộ dữ liệu theo đúng cấp bậc và không phân biệt lĩnh vực). (GD:MB,SX,GC:B)

u41 là 1 giám đốc có thể đọc được các dòng dữ liệu dành cho tất cả các giám đốc không đề cập lĩnh vực, chi nhánh (GĐ)

u42: 1 giám đốc có thể đọc được các dòng dữ liệu dành cho tất cả các giám đốc bất kể dữ liệu đó liên quan đến lĩnh vực, chi nhánh nào (GĐ:MB,SX,GC:B,T, N)

u5: 01 giám đốc phụ trách lĩnh vực sản xuất và gia công ở miền Nam. (GD:SX,GC:N)

u9: 01 trưởng phòng phụ trách tất cả các lĩnh vực ở tất cả các vùng miền (TP:MB,SX,GC:B,T,N)

t1 đến tất cả trưởng phòng phụ trách tất cả các lĩnh vực không phân biệt chi nhánh (TP:MB,SX,GC)

t2 đến trưởng phòng phụ trách lĩnh vực sản xuất ở miền Trung (TP:SX:T)

t3: phát tán đến tất cả các giám đốc (GD)

t4: phát tán đến tất cả các trưởng phòng. (TP)

t5: phát tán đến giám đốc, trưởng phòng phụ trách lĩnh vực mua bán ở miền Trung (TP:MB:T)

t6: phát tán đến mọi người dùng trong hệ thống. (NV)

t7: phát tán đến nhân viên không phân biệt lĩnh vực hay vùng miền (NV)

t8: phát tán đến trưởng phòng không phân biệt lĩnh vực hay vùng miền (TP)