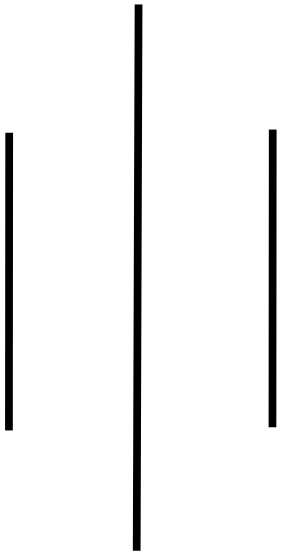
Deerwalk Institute Of Technology
Database Administration



Lab:

Submitted By:
Name: Sagar Giri
Roll No. 205

Submitted To:
Narayan Chalise

Date:

Virtual Private Database And policy types

Virtual Private Database (VPD) is a feature of Oracle Database 11g Enterprise Edition, and was introduced in Oracle8i. It is one of the most popular security features in the database. A virtual private database or VPD masks data in a larger database so that only a subset of the data appear to exist, without actually segregating data into different tables, schema or databases. Oracle Virtual Private Database enables us to create security policies to control database access at the row and column level.

Essentially, Oracle Virtual Private Database adds a dynamic WHERE clause to a SQL statement that is issued against the table, view. VPD policies can be simple or complex depending on the security requirements. A simple VPD example might restrict access to data during business hours and a more complex VPD example might read an application context during a login trigger and enforce row level security against the table. No matter how users connect to the protected table (via an application, a Web interface or SQL*Plus), the result is the same.

To implement VPD in oracle, we follow following procedure:

1. Create a arbitrary table, where we assign a policy later:

```
SQL> create table customer(lname varchar(20), fname varchar(30), climit int, mgr_id int);

Table created.
```

2. Fill in some dummy data:

```
SQL> insert into customer values('giri','sagar',1100,205);

1 row created.

SQL> insert into customer values('koirala','sameer',1200,206);

1 row created.

SQL> insert into customer values('aacharya','bidish',1200,204);

1 row created.
```

3. Verify that data are present in the newly created table:

```
SQL> select * from customer;

LNAME                FNAME                           CLIMIT     MGR_ID
-------------------- ------------------------------ ---------- ----------
giri                 sagar                            1100        205
koirala              sameer                           1200        206
aacharya             bidish                           1200        204

SQL>
```

4. Now we create a VPD policy and function, so that when a user executes select query, only the data of particular manager id is shown in the table.

The code for the function is written in the SQL file and is imported in the sql console.

PolicyFucntion.sql

```
CREATE OR REPLACE FUNCTION orders(
     schema_var IN VARCHAR2,
     table_var IN VARCHAR2
)
RETURN VARCHAR2
IS
  return_val VARCHAR2 (400);
BEGIN
  return_val := 'mgr_id = 205';
  RETURN return_val;
END orders;
/
```

The function is then imported:

```
SQL> @'C:\Users\IT Club\Desktop\policyFunction.sql'

Function created.
```

5. Now we write a VPD policy.

VpdPolicy.sql

```
BEGIN
  DBMS_RLS.ADD_POLICY(
     object_schema => 'system',
     object_name => 'customer',
     policy_name => 'secure',
     policy_function => 'orders',
     statement_types => 'SELECT');
END;
/
```

and then import it:

```
SQL> @'C:\Users\IT Club\Desktop\vpdPolicy.sql'

PL/SQL procedure successfully completed.
```

6. Now, when we execute select command, we can only view the details of member with MGR_ID=205.

```
SQL> select * from customer;

LNAME                FNAME                           CLIMIT     MGR_ID
-------------------- ------------------------------ ---------- ----------
giri                 sagar                            1100        205
```

7. We can drop the policy and function using DROP command. After dropping the function and policy, user can view all the data in the table.

```
SQL> DROP FUNCTION ORDERS;

Function dropped.

SQL> EXEC DBMS_RLS.DROP_POLICY('SYSTEM','CUSTOMER','SECURE');

PL/SQL procedure successfully completed.

SQL> select * from customer;

LNAME                FNAME                           CLIMIT     MGR_ID
-------------------- ------------------------------ ---------- ----------
giri                 sagar                            1100        205
koirala              sameer                           1200        206
aacharya             bidish                           1200        204
```