Database Security and Auditing: Protecting Data Integrity and Accessibility

Chapter 6
Virtual Private Databases

Objectives

- Define the term “virtual private database” and explain its importance
- Implement a virtual private database by using the VIEW database object
- Implement a virtual private database by using Oracle’s application context
- Implement the Oracle virtual private database feature

Objectives (continued)

- Use a data dictionary to view an Oracle virtual private database
- Use Policy Manager to view an Oracle virtual private database
- Implement row-level and column-level security

Overview of Virtual Private Databases

- A VPD deals with data access
- VPD controls data access at the row or column level
- SQL Server 2000: use VIEW data object
- Oracle10*g:*
  - Specific functions
  - Row-level security, fine-grained access (FGA)

Overview of Virtual Private Databases (continued)

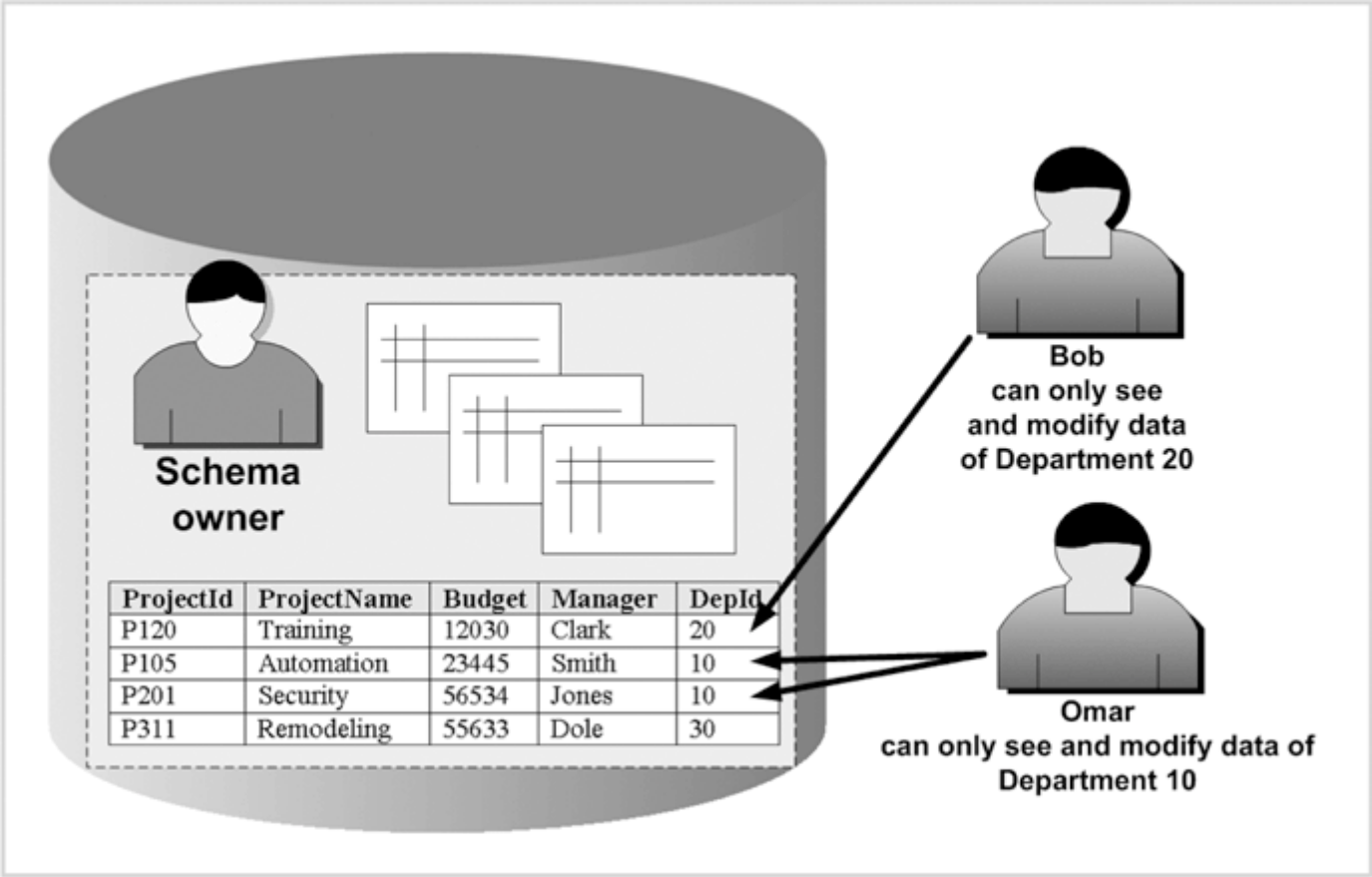

FIGURE 6-2 Virtual private database example

Overview of Virtual Private Databases (continued)

- Shared database schema:
  - Containing data that belongs to different users
  - User view or update only data he or she owns
- Purposes/benefits:
  - Security requirements necessitate data access be restricted at row or column level (FGA)
  - One database schema serves multiple unrelated groups or entities

Implementing a VPD Using Views

- View object limits what users can see and do with existing data: hides columns or rows from users
- CREATE VIEW statement: creates data views
- Views can become hard to administer; solution is VPD
- Implementation is limited and requires careful design and development

Implementing a VPD Using Views (continued)

- Example implementation steps:
  - Logon as DBSEC schema
  - Display the EMPLOYEES table
  - Create the table EMPLOYESS_VER1
  - Create a VIEW object to display rows that belong only to the logged on user
  - Grant SELECT and INSERT on this view to another user
  - Insert a row using EMPLOYEES_VIEW1

Implementing a VPD Using Views (continued)

- Example implementation steps (continued)
  - Logon as the other user
  - Select the EMPLOYEES_VIEW1 VIEW object; you see only rows that belongs to the other user

Hiding Rows Based on the Current User

- System function USER:
  - Returns database user
  - Used to implement row-based security
- Implementing row-based security with views:
  - Need a column in your tables for the row's owner
  - Preface it with "CTL"

Hiding Rows Based on the Current User (continued)

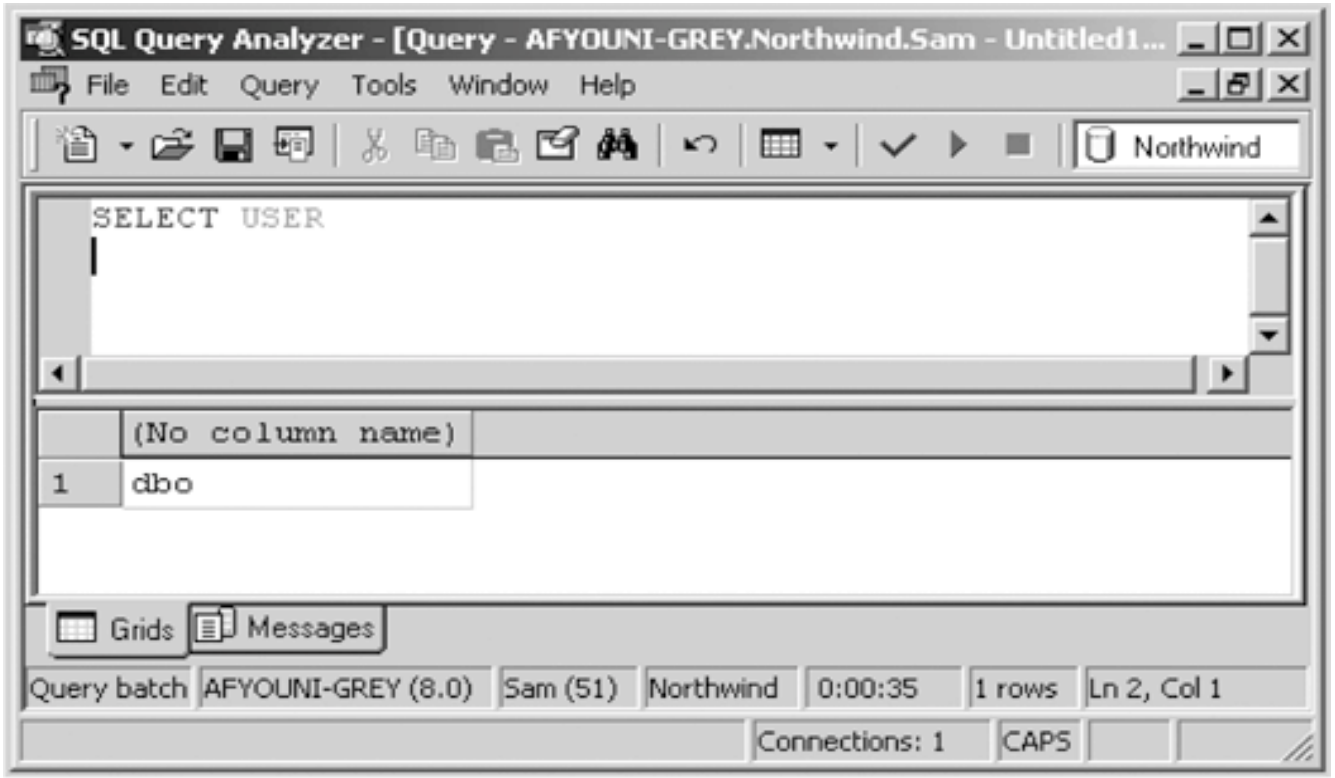

FIGURE 6-3 Command to display current user

Hiding Rows Based on the Current User (continued)

- Steps:
  - Alter table to contain column CTL_UPD_USER
  - Create database users in the database
  - Create the view users will use to access table
  - Test the code:
    - Login as one user and do some actions
    - Then login as a different user

Hiding Rows Based on the Current User (continued)

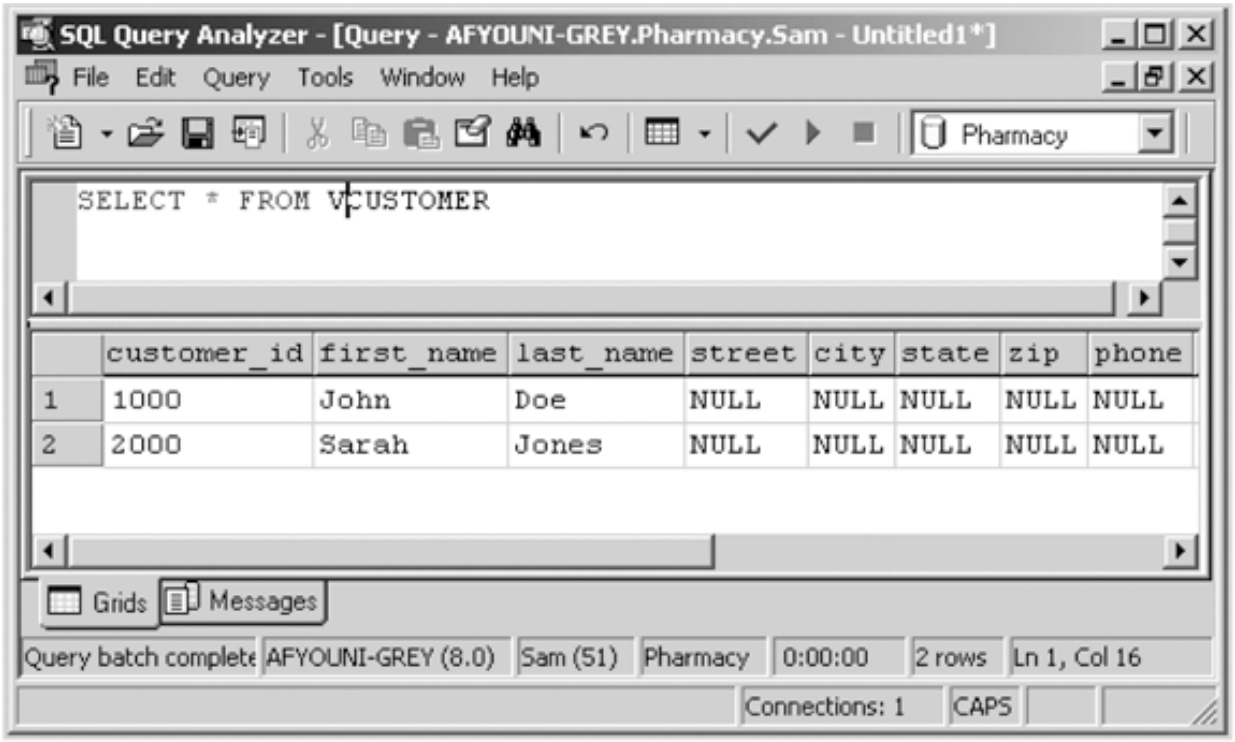

FIGURE 6-4 Query to show content of VCUSTOMER table before security enforcement

Hiding Rows Based on the Current User (continued)

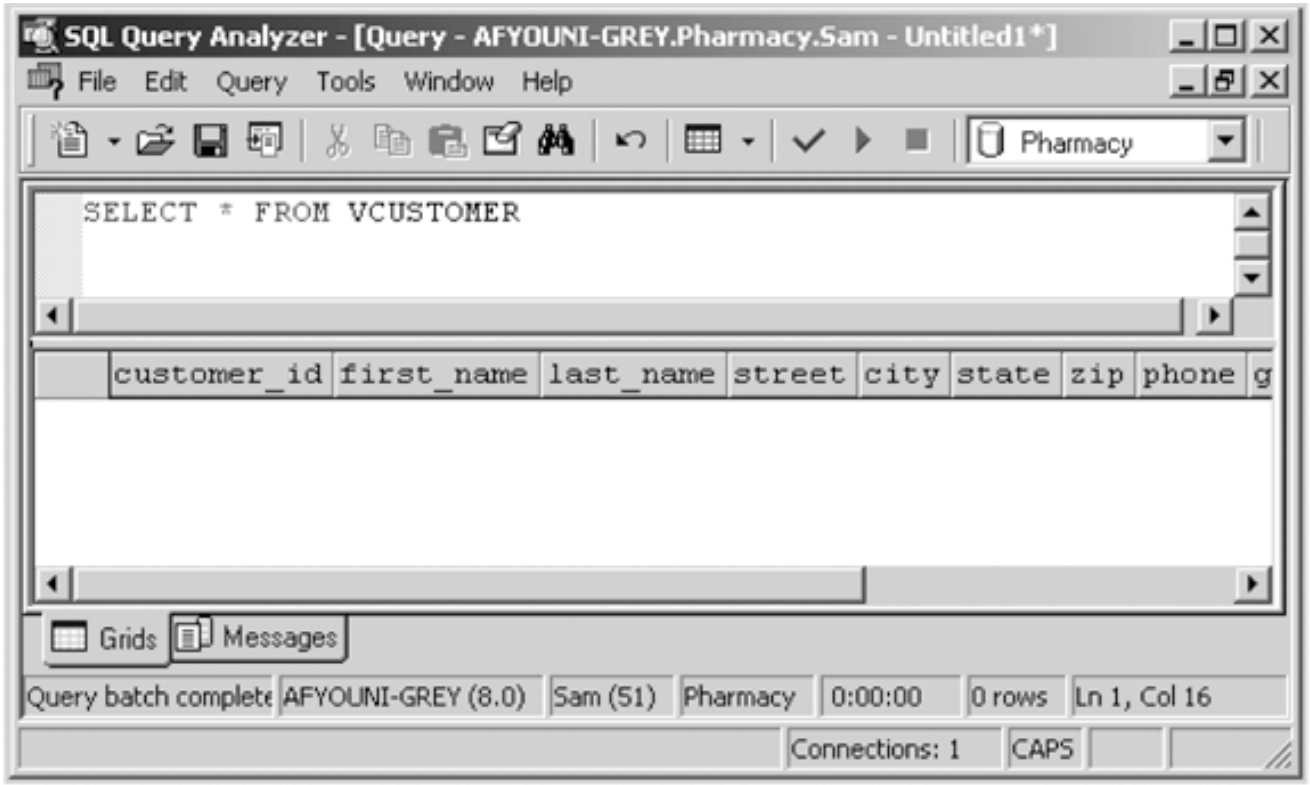

FIGURE 6-5 Query to show content of VCUSTOMER table after security enforcement

Implementing a VPD Using Application Context in Oracle

- Triggers
- Application context:
  - Functionality specific to Oracle
  - Allows to set database application variables that can be retrieved by database sessions
  - Variables can be used for security context-based or user-defined environmental attributes
- Dynamic performance view V$SESSION
- Application context function SYS_CONTEXT

Implementing a VPD Using Application Context in Oracle (continued)

Table 6-1 Common USERENV namespaces

Attribute	Description of What the Attribute Returns
TERMINAL	Operating system terminal name for the current connected session
IP_ADDRESS	Network IP address for the current connected session
HOST	Name of the host machine for the current connected session
DB_NAME	Name of the database to which the current session is connected
CURRENT_USER	Database name for the current connected session
DB_DOMAIN	Network domain name for the database to which the current session is connected
OS_USER	Operating system user name for the current connected session
SERVER_HOST	Name of the host machine to which the current database session is connected
SESSIONID	Auditing session identifier for the current connected session
ISDBA	Information to indicate whether the connected session has DBA privileges or not; the returned value is a Boolean TRUE or FALSE

The information in this table is derived from the online documentation that Oracle provides at the Oracle Technology Network site: *www.otn.oracle.com*.

Implementing a VPD Using Application Context in Oracle (continued)

- Set your own application context: use Oracle PL/SQL package DBMS_SESSION (SET_CONTEXT procedure)
- Example steps:
  - Using DBSEC that has privileges to create tables and other database objects:
    - Application context table APP_CONTEXT_USERS
    - ORDERS table

Implementing a VPD Using Application Context in Oracle (continued)

- Example steps (continued):
  - As DBSEC insert rows into:
    - ORDERS table
    - APP_CONTEXT_USERS table
  - As DBSEC create a VIEW object to display rows based on application context SECURITY_LEVEL
  - As DBSEC create context for ORDERS_APP
  - Create a package; can be owned by SYS, SYSTEM or DBSEC

Implementing a VPD Using Application Context in Oracle (continued)

- Example steps (continued):
  - Grant the user CREATE ANY CONTEXT privilege and the execute privilege to DBSEC
  - Create a logon database trigger
  - Connect as HR and select from the view
- Complex implementation: use VPD instead

Implementing Oracle Virtual Private Databases

- VPDs are a more direct solution
- User functions:
  - DBSEC users: application schema owner
  - CUSTOMERS: used to demonstrate VPDs
  - *VPD_CLERK1, VPD_CLERK2,* and *VPD_CLERK3* users: database users that are used to test VPDs

Implementing Oracle Virtual Private Databases (continued)

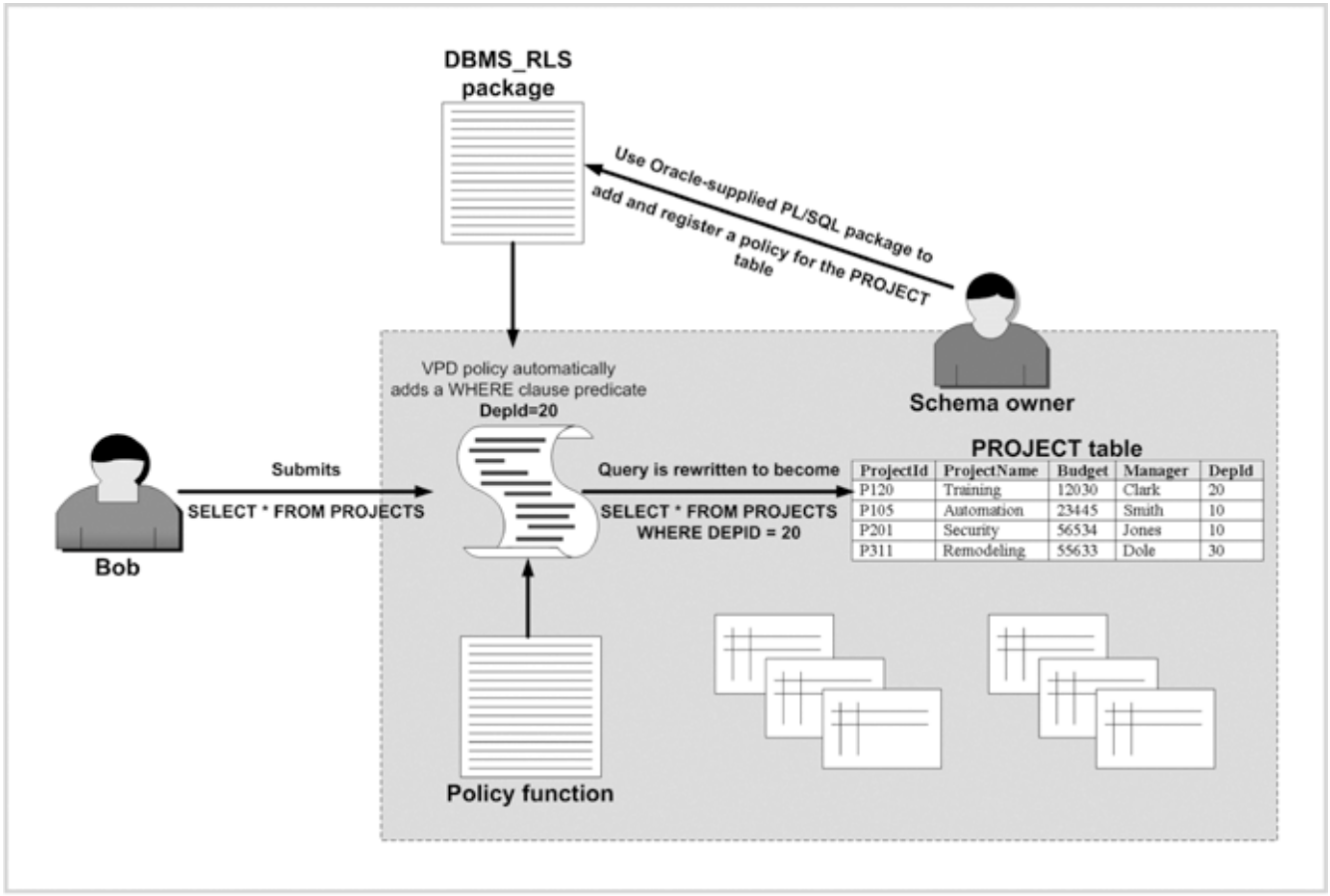

FIGURE 6-6 Architecture of Oracle virtual private database feature

Implementing Oracle Virtual Private Databases (continued)

- Create table for customer users:
  - Create the CUSTOMERS table
  - Insert rows into the CUSTOMERS table
  - Create three users for testing, VPD_CLERK1, VPD_CLERK2, and VPD_CLERK3
  - Grant the necessary privileges on the CUSTOMERS table to use each test
- ROW_OWNER security: row-level security based on user that owns row

Implementing Oracle Virtual Private Databases (continued)

- Steps:
  - Create a policy function to add a predicate to the WHERE clause
  - Using DBMS_RLS add the VPD policy: Oracle-supplied package
  - Log in as VPD_CLERK1; display number of records that this user can see
  - Disable this policy

Implementing Oracle Virtual Private Databases (continued)

Table 6-2 Most commonly used procedure of DBMS_RLS

Procedure	Description
PROCEDURE ADD_POLICY	Adds and registers a VPD policy for a table
PROCEDURE ADD_POLICY_CONTEXT	Adds an application context to a policy
PROCEDURE DROP_POLICY	Removes a VPD policy from a table
PROCEDURE ENABLE_POLICY	Enables or disables a policy

The information in this table is derived from the online documentation that Oracle provides at the Oracle Technology Network site: *www.otn.oracle.com*.

Implementing Oracle Virtual Private Databases (continued)

- APPLICATION-CONTEXT security: allows specific users to see only rows for a specific sales representative
- Steps:
  - Create the DBSEC_CUSTOMERS_APP_CONTEXT table
  - Insert rows into this table
  - Create a trusted package that allows DBSEC to execute DBMS_SESSION

Implementing Oracle Virtual Private Databases (continued)

- Steps (continued):
  - Create an application context for this policy
  - Create a new VPD function policy to add a WHERE clause predicate
  - Add a VPD policy for the CUSTOMERS table
  - Create an after-logon trigger
  - Now log on as VPD_CLERK2

Implementing Oracle Virtual Private Databases (continued)

- ROLE SECURITY LEVEL:
  - Detects the role of the user
  - A predicate is used to filter the rows that can be seen by each user
- Steps:
  - Disable any policies on the CUSTOMERS table
  - Disable the AFTER LOGON database trigger
  - Create three new roles

Implementing Oracle Virtual Private Databases (continued)

Table 6-3 Setup for using roles to determine security levels

User	Assigned Role	Security Level
VPD_CLERK1	CLERK_ROLE	2
VPD_CLERK2	MANAGER_ROLE	3
VPD_CLERK3	ADMIN_ROLE	4

Implementing Oracle Virtual Private Databases (continued)

- Steps (continued):
  - Create application context for the security level
  - Create application context package to set the application context
  - Create a policy function to implement row-level security (VPD)
  - Create a policy to enforce a WHERE clause predicate

Implementing Oracle Virtual Private Databases (continued)

- Steps (continued):
  - Application logs on as VPD_CLERK3; run PKG_DBSEC_ROLE_SECURITY_LEVEL package
  - Repeat last step, this time use VPD_CLERK1
- VPD policies can be grouped for organizational purposes
- Oracle enforces row-level security using all the DML statements

Viewing VPD Policies and Applications Context Using the Data Dictionary

Table 6-4 Oracle policy and application context data dictionary views

View Name	Description
DBA_POLICIES	Contains all policies that are created in the database and their attributes
ALL_POLICIES	Contains all policies that the current user owns and has access to and their attributes
USER_POLICIES	Contains all policies owned by the current user and their attributes
V$VPD_POLICY	Lists all policies that have been used and executed and are still cached in memory
ALL_CONTEXT	Lists all contexts that the current user owns or has privileges to view
SESSION_CONTEXT	Lists all active contexts for the current session

Viewing VPD Policies and Applications Context Using Policy Manager

- Graphical tool called Policy Manage
- Use SYSTEM credentials to log in
- FGA control policies are divided into two parts:
  - Policy groups
  - Application context

Viewing VPD Policies and Applications Context Using Policy Manager (continued)

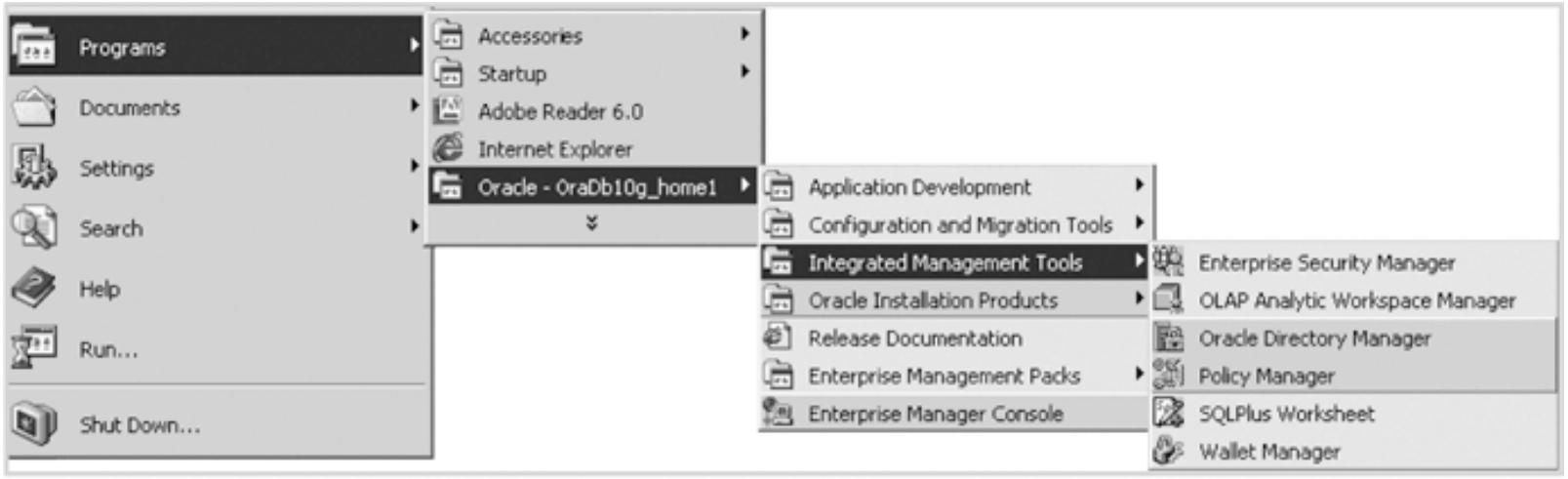

FIGURE 6-7 Policy Manager shortcut in the Start menu

Viewing VPD Policies and Applications Context Using Policy Manager (continued)

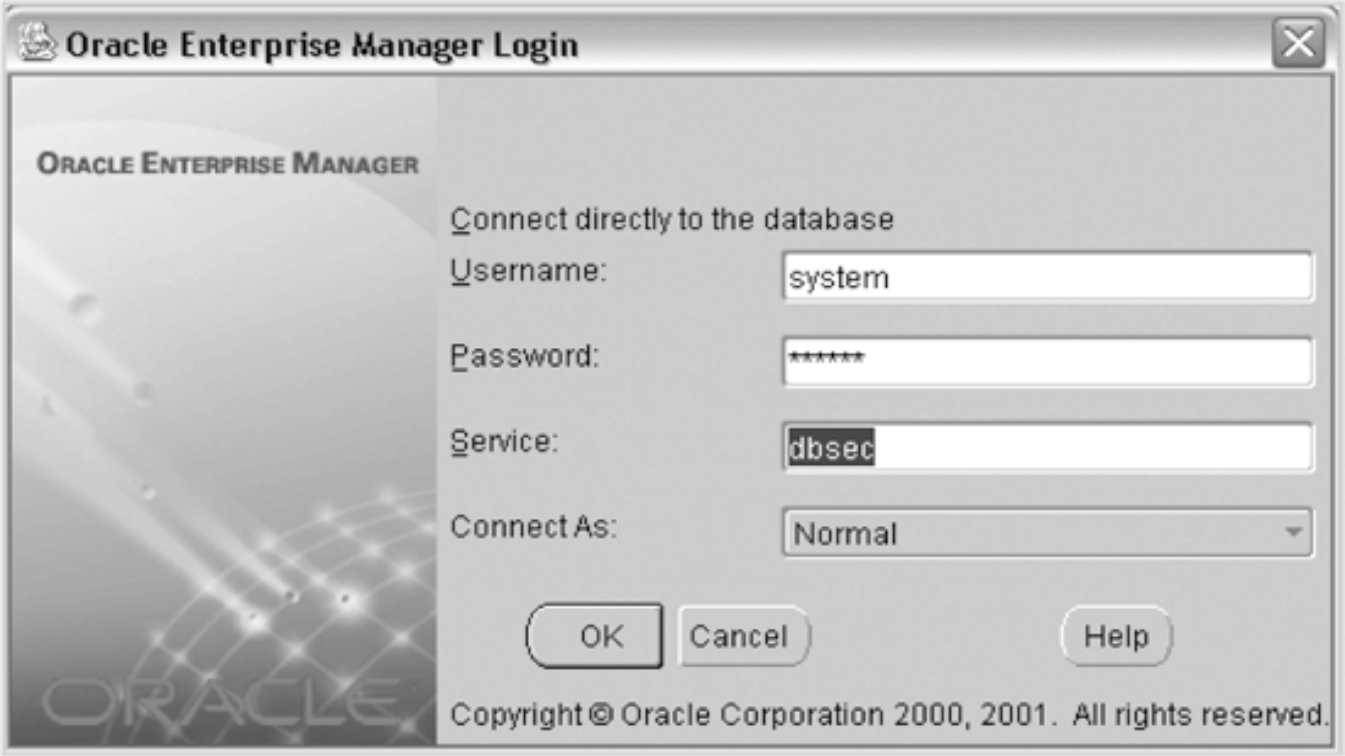

FIGURE 6-8 Logging into Oracle Policy Manager

Viewing VPD Policies and Applications Context Using Policy Manager (continued)

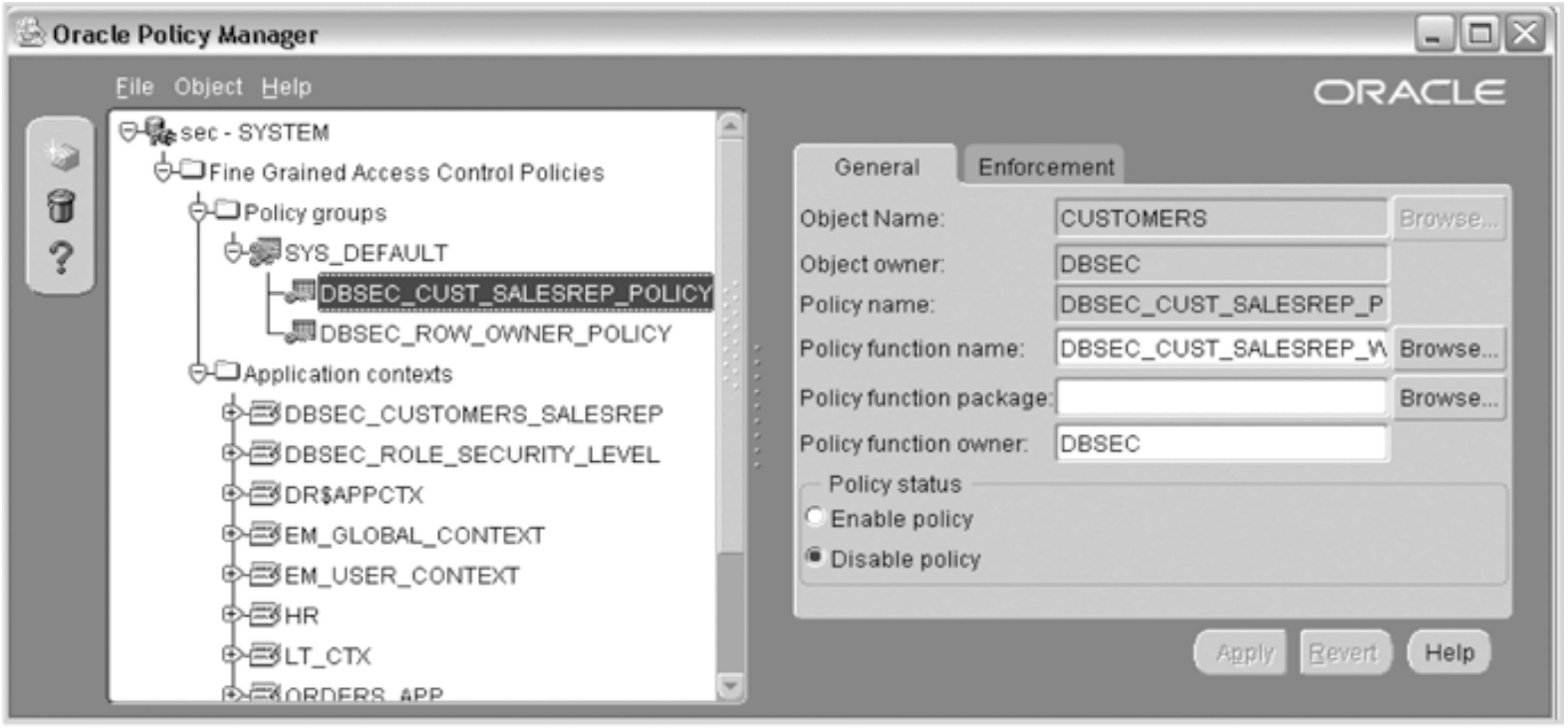

FIGURE 6-9 Oracle Policy Manager

Implementing Row- and Column-level Security with SQL Server

- SQL Server 2000 does not support VPDs; you can mimic their functionality
- Use views and expand security models

Row-based Security Using Access Levels

- Variation of both:
  - Application table-based security model
  - Application function-based security model
- Access levels:
  - 0 = No access
  - 1 = select
  - 2 = select, insert
  - 3 = select, insert, update

Row-based Security Using Access Levels (continued)

- Access levels (continued):
  - 4 = select, insert, update, delete
  - 5 = administrator access
- Steps:
  - Create the APPLICATION USERS table
  - Alter the CUSTOMER table to include the ACCESS CONTROL column
  - With the security structure in place use a view to retrieve data

Row-based Security Using Application Functions

- Steps (continued): apply privileges
- Drawbacks: it allows insertion, update, and deletion of records
- Alternatives:
  - Use stored procedures
  - Use application functions: access table list a function instead of a level

Column-based Security

- VPD and Column Access Using Oracle steps:
  - Log in as VPD_CLERK2 and view rows and columns in the CUSTOMERS table
  - Log in as the DBSEC user and recreate the policy on customers
  - Log in as VPD_CLERK2 and query the CUSTOMERS table

Column-based Security

- Column privileges in Oracle steps:
  - Log in as DBSEC and create a table called TEST
  - Grant SELECT on the TEST table to SCOTT
  - Grant UPDATE only on the column TEXT in the TEST table to SCOTT
  - Insert a row into the TEST table and save it
  - Log in as SCOTT and query the TEST table owned by DBSEC

Column-based Security (continued)

- Column privileges in Oracle steps (continued):
  - Update the TEXT column in the TEST table
  - Try to update the NUM column in the TEST table

Column-based Security (continued)

- Access-level control with SQL Server steps:
  - Create the APP_TABLES table
  - Create the APP_COLUMNS columns
  - All access to the tables must be performed with stored procedures

Column-based Security (continued)

- Column Privileges with SQL Server steps: set update permissions for sam on the column phone in the Customer table
- Customized column-access control facilitates management

Summary

- A virtual private database allows or prevents data access at the row or column level; implemented using VIEW database object
- VPDs are also referred to as row-level security (RLS) or fine-grained access (FGA)
- SQL Server does not support VPDs
- Microsoft SQL Server 2000 system function of USER

Summary (continued)

- Oracle Application context:
  - Allows setting of database application be retrieved by database sessions
  - SYS_CONTEXT function
  - PL/SQL package DBMS_SESSION
  - SET_CONTEXT procedure
- Use Oracle-supplied package DBMS_RLS to add the VPD policy
- Oracle data dictionary views

Summary (continued)

- Oracle Policy Manager: graphical tool used to administer VPD policies
- Oracle has the capability to restrict updates or inserts on columns, using GRANT UPDATE(column) and INSERT(column)