UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

Relatórios Técnicos
do Departamento de Informática Aplicada
da UNIRIO
n° 0016/2010

Estudos do Oracle Database Vault

Sergio Puntar
Leonardo Guerreiro Azevedo

Departamento de Informática Aplicada

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
Av. Pasteur, 458, Urca - CEP 22290-240
RIO DE JANEIRO – BRASIL

Projeto de Pesquisa

Grupo de Pesquisa Participante

Patrocínio

PETROBRAS

Relatórios Técnicos do DIA/UNIRIO, No. 0016/2010
Editor: Prof. Sean W. M. Siqueira Novembro, 2010

Estudos do Oracle Database Vault *

Sergio Puntar, Leonardo Guerreiro Azevedo

Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec)
Departamento de Informática Aplicada (DIA) – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

sergio.puntar@uniriotec.br, azevedo@uniriotec.br

Abstract. Information security is an essential subject for commercial and governmental organizations, and its deployment requires implementations that guarantee only authorized users access data under security control. Besides, it is common that database administrators have direct access to sensitive data, even though these data are restricted to business users. This work studies the tool Database Vault from Oracle DBMS whose concern is to prevent internal threats, using responsibility separation between privileged database users, protecting sensitive data, but still enabling database administration.

Keywords: Access control, Oracle Database Vault, Separation of responsibilities, internal threats.

Resumo. Segurança da informação é um tema essencial em organizações comerciais e governamentais, e sua operacionalização requer a existência de funcionalidades que permitam garantir que apenas as pessoas devidamente autorizadas acessem os dados. Além disso, é muito comum que os administradores de banco de dados tenham acesso direto aos dados sensíveis, mesmo que haja controle de acesso para os usuários de negócio. Este trabalho apresenta um estudo da ferramenta Database Vault do SGBD Oracle, que se preocupa em impedir ameaças internas, através da separação de responsabilidades entre os usuários privilegiados do banco, protegendo os dados sensíveis, mas ainda permitindo a manutenção do banco pelos mesmos usuários.

Palavras-chave: Controle de acesso, Oracle Database Vault, Separação de responsabilidades, Ameaças internas.

* Trabalho patrocinado pela Petrobras.

Sumário

Índice de figuras

1 Introdução

Esse relatório tem como objetivo caracterizar a ferramenta de controle de acesso Database Vault, presente no Oracle Database versão 10.2.0.4, bem como avaliar o uso dessa ferramenta para autorização de acesso a banco de dados.

Este relatório foi produzido pelo Projeto de Pesquisa em Autorização de Informação como parte das iniciativas dentro do contexto do Projeto de Pesquisa do Termo de Cooperação entre UNIRIO/NP2Tec e a PETROBRAS/TIC-E&P/GDIEP.

Esse relatório está organizado em 5 capítulos, sendo o capítulo 1 a presente introdução. No capítulo 2, é caracterizada a ferramenta Database Vault do Oracle. No capítulo 3, são apresentados exemplos de uso da ferramenta. No capítulo 4, é apresentado um exemplo de integração do DBVault com o Virtual Private Database. Nos capítulos 5 e 6, são apresentadas as conclusões e referências do trabalho realizado, respectivamente.

2 Oracle Database Vault

Esta seção analisa as funcionalidades do Oracle Database Vault, no que se refere ao seu potencial de uso na definição de uma estratégia de controle de acesso de aplicativos aos dados existentes na BDIEP.

Com o Oracle Database Vault é possível restringir o acesso em áreas específicas do banco de dados a qualquer usuário, incluindo usuários com acesso de administração, permitindo um controle de acesso refinado [HUEY *et al.*, 2008]. O DB Vault preocupa-se com ameaças internas, portanto sugere uma separação de responsabilidades entre os usuários privilegiados, protegendo dados sensíveis desses usuários, mas ainda permitindo a manutenção do banco pelos mesmos usuários.

O Oracle Database Vault divide-se na série de componentes detalhados a seguir.

2.1 Componentes de controle de acesso

O Database Vault possui os seguintes componentes de controle de acesso:

- ***Realms***. Um *realm* (domínio) é um grupo de esquemas, objetos e papéis que precisam de segurança. Por exemplo, é possível agrupar um conjunto de esquemas, objetos e papéis e relacioná-los com uma área da companhia, como, por exemplo, área de vendas ou RH. Em seguida, esse *realm* pode ser utilizado para controlar privilégios de sistema para determinadas contas e papéis.
- ***Command rules***. Uma *command rule* (regra de comando) é uma regra especial que permite controlar como os usuários podem executar um comando SQL, incluindo SELECT, ALTER SYSTEM e comandos DDL (*database definition language*) e DML (*data manipulation language*). As *command rules* em conjunto com os *rule sets* determinam se um comando é permitido ou não.
- ***Factors***. Um *factor* (fator) é uma variável ou atributo como, por exemplo, a localização do usuário, o endereço IP do banco de dados ou o usuário de sessão, que o Database Vault pode reconhecer e proteger. Os *factors* podem ser utilizados em ações como, por exemplo, autorizar a conexão de contas específicas

com o banco de dados, ou criar filtros que restrinjam a visibilidade e gerenciamento dos dados. Cada *factor* pode possuir uma ou mais identidades, que são os valores reais de um *factor*.

- ***Rule Sets***. Um *rule set* (conjunto de regras) é uma coleção de uma ou mais regras que pode ser associada com um *realm*, *command rule*, *factor* ou *secure application role*. O conjunto de regras é avaliado como verdadeiro ou falso baseado na avaliação de cada regra do conjunto e no tipo de avaliação (todas verdadeiras ou alguma verdadeira). Cada regra é uma expressão PL/SQL que pode ser avaliada como verdadeira ou falsa, além disso, uma mesma regra pode aparecer em vários conjuntos de regra.
- ***Secure application roles***. Um *secure application role* (atribuição de aplicação segura) é um papel especial do Oracle que pode ser habilitado baseado na avaliação de um *rule set*.

2.2 Oracle Database Vault Administrator e Configuration Assistant

O Oracle Database Vault Administrator (DVA) é uma aplicação Java que utiliza as APIs PL/SQL do DB Vault para que gestores de segurança que não tenham conhecimento de PL/SQL possam configurar políticas de controle de acesso através de uma interface amigável. O DVA disponibiliza uma coleção de relatórios de segurança que ajudam na configuração de segurança. Já o Oracle Database Vault Configuration Assistant (DVCA) é um utilitário de linha de comando utilizado para manutenção da instalação do DB Vault.

2.3 Esquemas DVSYS, DVF, interfaces e pacotes PL/SQL

O Database Vault utiliza dois esquemas para armazenamento e recuperação de dados: o esquema DVSYS e o esquema DVF. O esquema DVSYS contém os papéis, visões, contas, funções e outros objetos necessários para processar os dados do Oracle para o DB Vault. O esquema DVF contém funções públicas utilizadas para recuperar (em tempo de execução) os valores dos *factors* definidos nas configurações do DB Vault.

Além destes esquemas, o Database Vault fornece uma coleção de interfaces e pacotes PL/SQL que permitem que gestores de segurança ou desenvolvedores configurem o controle de acesso necessário. As funções e *procedures* permitem que contas gerais do banco de dados operem dentro dos limites das políticas de controle de acesso, no contexto de uma sessão do banco de dados.

2.4 APIs PL/SQL do Oracle Label Security

O Oracle Database Vault possui capacidades de controle de acesso que podem ser integradas com o Oracle Label Security. No Oracle existe a aplicação Policy Manager a qual permite a definição e aplicação de políticas Label Security em objetos do banco de dados. O Oracle Label Security fornece ainda uma coleção de APIs PL/SQL que podem ser utilizadas pelos desenvolvedores para definir políticas Label Security.

A funcionalidade Label Security é um framework a parte do Oracle, que utiliza a tecnologia VPD para aplicar controle de acesso conhecido como MAC (Mandatory Access Control). O tipo de controle de acesso MAC foi proposto por DoD [1983]. Azevedo *et al*. [2010] apresentam estudos e avaliações do Label Security.

2.5 Ferramentas de monitoração e relatório

É possível gerar relatórios sobre as várias atividades que o DB Vault monitora. Além disso, é possível monitorar a mudança de políticas, tentativas de violação de segurança, configuração do banco de dados e mudanças estruturais.

2.6 Como o Database Vault protege os dados do acesso de um DBA

A Figura 1 ilustra como o Database Vault protege dados sensíveis do acesso de um DBA através de *Realms*. Nesse cenário, dois usuários, cada um responsável por um *Realm* diferente, possuem os mesmos privilégios de sistema. O dono de um *Realm* pode ser um usuário ou uma role do banco de dados. Dessa forma, cada uma das *roles*, OE_ADMIN e HR_ADMIN, pode ser protegida por um *Realm* como um objeto seguro e ainda ser configurada como dona de um *Realm*.

Além disso, somente um dono de *Realm*, como o OE_ADMIN, pode conceder e revogar *roles* do banco que são protegidas pelo *Realm*. Os donos de *Realm* não podem gerenciar *roles* protegidas por outros *Realms*, como a role DBA, criada pelo usuário SYS no *Realm* "*Oracle Data Dictionary*" durante a instalação do Database Vault.

Qualquer tentativa não autorizada de usar um privilégio de sistema para acessar um objeto protegido por um *Realm* criará uma violação de *Realm*, que pode ser auditada. Os poderes de cada dono de *Realm* são limitados ao seu próprio *Realm*. Por exemplo, o OE_ADMIN não tem acesso ao *Realm* "*Human Resource*"*s*, e o HR_ADMIN não tem acesso ao *Realm* "*Order Entry*", como ilustrado na Figura 1.

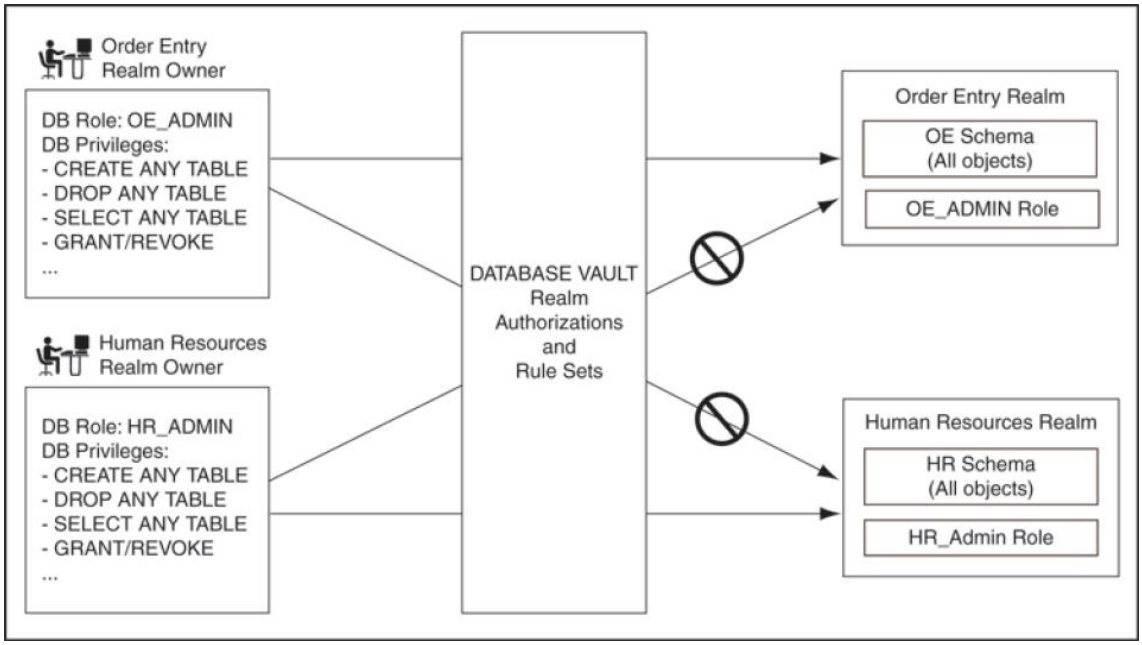


Figura 1 - Funcionamento de autorizações para *Realms* e donos de *Realms* [HUEY *et al.*, 2008]

2.7 Administração do Oracle Database Vault

Antes de apresentar os exemplos de uso do Database Vault, é importante caracterizar como é feita a sua administração. A administração do Oracle Database Vault é feita basicamente através de dois papéis: *Database Vault Owner*, identificado no sistema pela

atribuição DV_OWNER, e *Database Vault Account Manager*, identificado no sistema pela atribuição DV_ACCTMGR.

O *Database Vault Owner* é responsável pela criação de *realms*, regras de comando, fatores e conjuntos de regra, além da geração de relatórios e monitoramento do sistema. O *Database Vault Account Manager* é responsável por todo o gerenciamento de papéis e contas de usuário do sistema.

Após a instalação do Database Vault, o papel de *Account Manager* passa a ser necessário, pois os usuários privilegiados que já existem perdem os privilégios de criação, alteração e remoção de usuários e papéis. Essa perda de privilégios se dá pelo fato de que uma das principais intenções do Database Vault é limitar o poder de usuários privilegiados, portanto a centralização de uma função importante como o gerenciamento de contas é essencial para a aplicação dessa segurança.

Durante a instalação do Oracle Database Vault é criada uma conta de usuário que receberá a atribuição de *Database Vault Owner*, e opcionalmente uma conta de usuário que receberá a atribuição de *Account Manager.* Se somente uma conta for criada, ela receberá as duas atribuições.

Além desses dois papéis, o Oracle Database Vault fornece ainda um conjunto de atribuições menos poderosas que possuem certo poder de administração [HUEY *et al.*, 2008]. Porém, essencialmente, toda a administração pode ser feita através dos papéis *Database Vault Owner* e *Database Vault Account Manager*.

No âmbito desse relatório o papel *Database Vault Owner* foi atribuído ao usuário DBVOWNER e o papel *Database Vault Account Manager* foi atribuído ao usuário DBVMANAGER.

3 Exemplo de uso do Oracle Database Vault

Para ilustrar o funcionamento do Oracle Database Vault, foram elaborados exemplos de uso baseados nos tutoriais disponibilizados na OBE (*Oracle By Example*) [ORACLE, 2010a, 2010b]. Os exemplos apresentados a seguir utilizam as tabelas do esquema HR disponibilizado na instalação padrão do Oracle Enterprise 10.2.0.4.

3.1 Restringindo o acesso do DBA a dados sensíveis

Esta seção apresenta um exemplo de restrição de acesso aos dados do esquema HR por parte dos DBAs. Ela contempla os seguintes passos:

1. Criar um Realm para proteger os dados
2. Incluir os objetos sensíveis no Realm
3. Criar uma autorização de Realm para o dono dos dados

Considere um cenário muito comum onde a segurança é aplicada no nível de aplicação. Geralmente nesses casos, apesar dos usuários comuns estarem sujeitos a um controle de acesso, os DBAs têm total acesso às informações cadastradas na base de dados, o que aumenta consideravelmente o risco de ataques internos. Para contornar esse problema, o Database Vault permite bloquear o acesso a dados sensíveis por parte dos DBAs, sem que com isso eles percam o poder de administração do banco.

Para exemplificar esse conceito, foi feita uma conexão com o banco através do usuário privilegiado SYSTEM e realizada uma consulta sobre a tabela EMPLOYEES do esquema HR. Esses passos são apresentados na Figura 2.

```
SQL> connect system
Informe a senha:************
Conectado.
SQL> SELECT last_name, phone_number, salary
  2  FROM hr.employees
  3  WHERE employee_id < 110;

LAST_NAME                 PHONE_NUMBER            SALARY
------------------------- -------------------- ----------
King                      515.123.4567              24000
Kochhar                   515.123.4568              17000
De Haan                   515.123.4569              17000
Hunold                    590.423.4567               9000
Ernst                     590.423.4568               6000
Austin                    590.423.4569               4800
Pataballa                 590.423.4560               4800
Lorentz                   590.423.5567               4200
Greenberg                 515.124.4569              12000
Faviet                    515.124.4169               9000

10 linhas selecionadas.
```

Figura 2 - Consulta sem controle de acesso realizada pelo usuário privilegiado SYSTEM

Observe que, por enquanto, o usuário SYSTEM pode visualizar sem problemas os dados da tabela EMPLOYEES, pois possui o privilégio SELECT ANY TABLE. Para bloquear o acesso desse usuário é preciso criar um *Realm* (veja a sessão 2.1) para os dados sensíveis. Isso pode ser feito através do *Database Vault Administrator*, que é acessado pela URL http://<servidor>:<porta>/dva, como mostra a Figura 3.

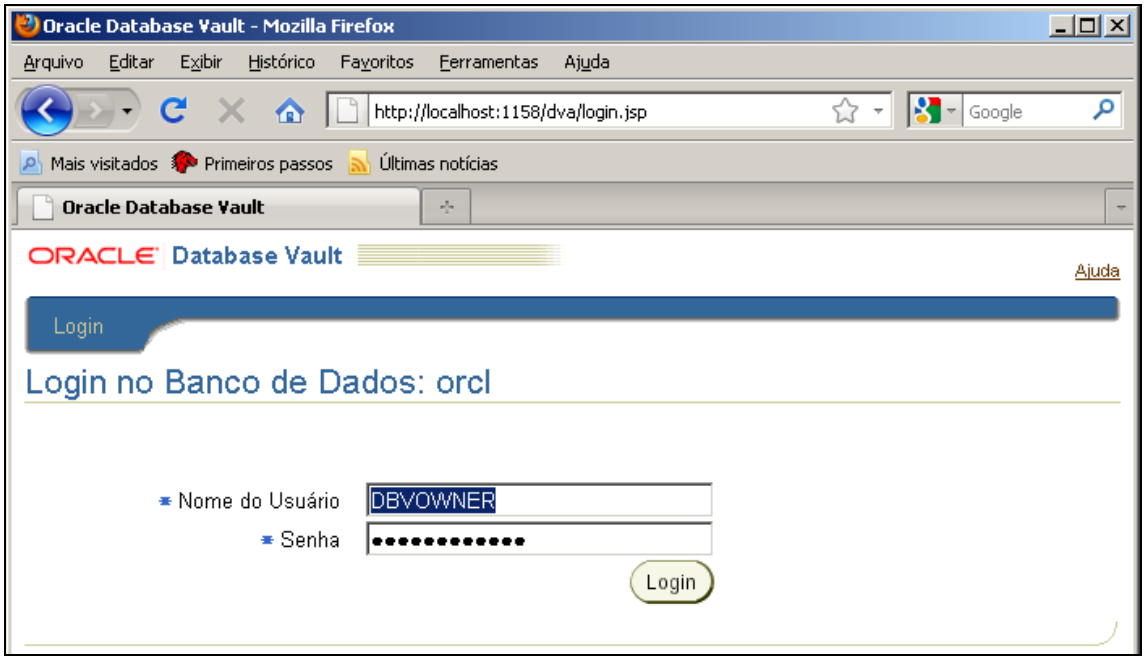


Figura 3 - *Database Vault Administrator* acessado via *browser*

Após entrar no DVA com o usuário DBVOWNER, foi criado o *Realm* através da opção "*Realms*" no menu principal da ferramenta, como mostra a Figura 4.

Figura 4 – Opção *Realms* no menu principal do *Database Vault Administrator*

Para criar o *Realm*, selecione a opção criar acima da tabela que apresenta os *Realms* já existentes, como mostra a Figura 5.

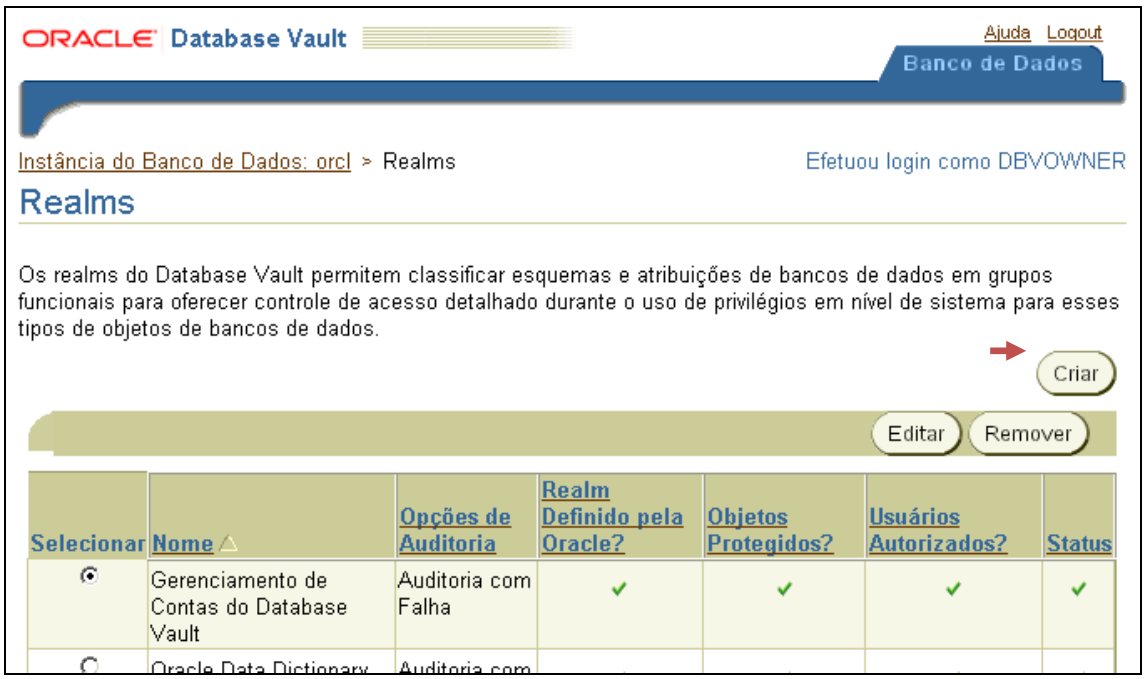


Figura 5 - Área de listagem de *Realms*

No próximo passo, foram inseridas as informações do *Realm*, tais como: nome, descrição, *status* e opções de auditoria, como mostra a Figura 6. As opções de auditoria

definem quando as informações de auditoria devem ser registradas: nunca, quando ocorrer alguma falha ou sempre. Neste caso, a segunda opção foi selecionada.

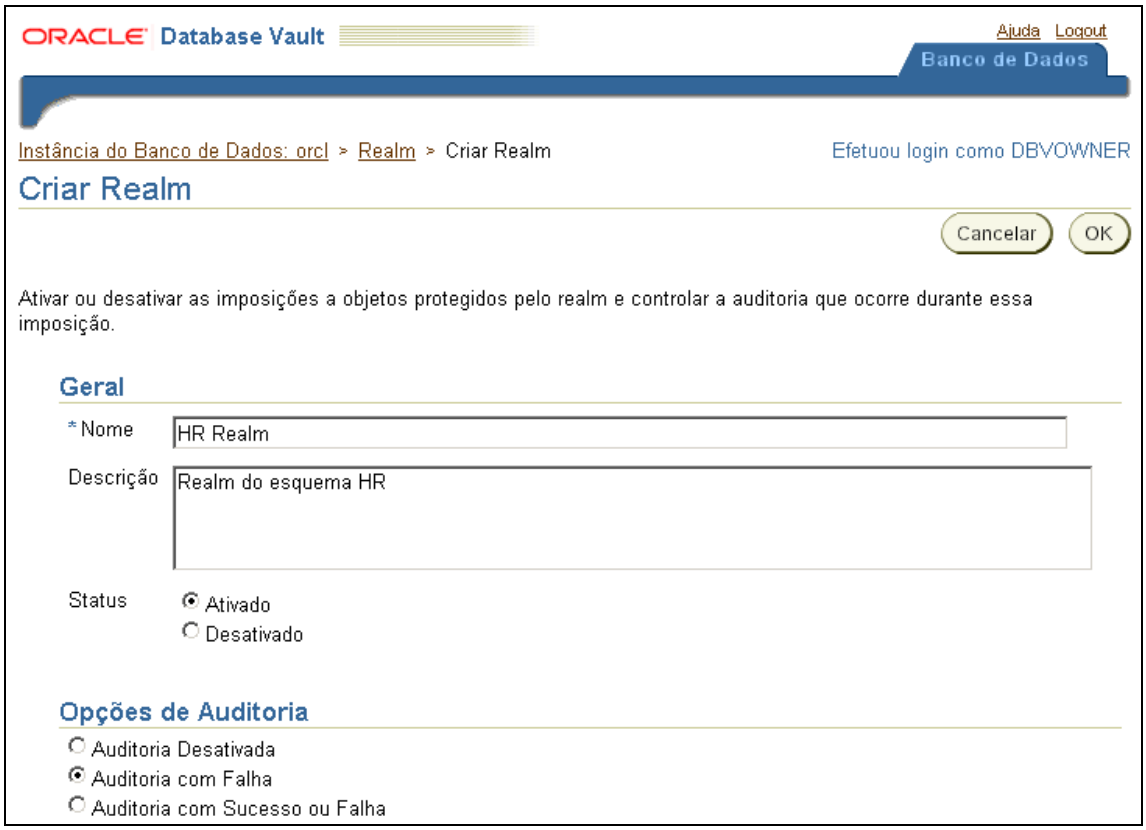


Figura 6 - Criação do *Realm* "HR Realm"

Clicando em OK, o *Realm* é criado, porém ainda não possui nenhum objeto. Para incluir os objetos voltamos à área de listagem de *Realms*, apresentada na Figura 5, foi selecionado o *Realm* recém criado e, em seguida ele foi editado (opção editar), como apresenta a Figura 7.

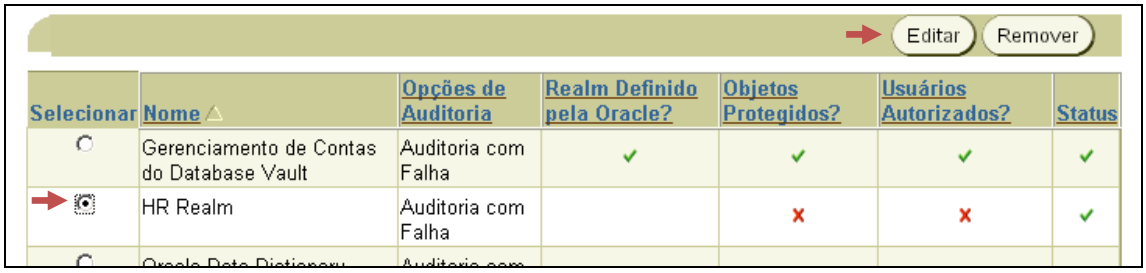


Figura 7 - Listagem dos *Realms* com o *Realm* recém criado

Na área de edição de *Realms* pode-se alterar os dados definidos quando o *Realm* foi criado e incluir objetos e autorizações para ao *Realm*. Por enquanto, foi incluído o esquema HR como objeto do *Realm*. Para isso, na área "Objetos Protegidos por *Realm*", que lista os objetos pertencentes ao *Realm*, é necessário clicar no botão criar, como mostra a Figura 8. Segue-se, então, para a tela apresentada na Figura 9.

Figura 8 - Listagem de objetos pertencentes ao *Realm*

Na opção "Proprietário do Objeto" (Figura 9), o nome do esquema que deve ser protegido foi inserido, nesse caso HR. Na opção "Tipo de Objeto", o valor '%' foi selecionado, indicando que devem ser protegidos todos os tipos de objetos. Finalmente, na opção "Nome do Objeto", o valor '%' foi colocado, pois deseja-se proteger todos os objetos do esquema.

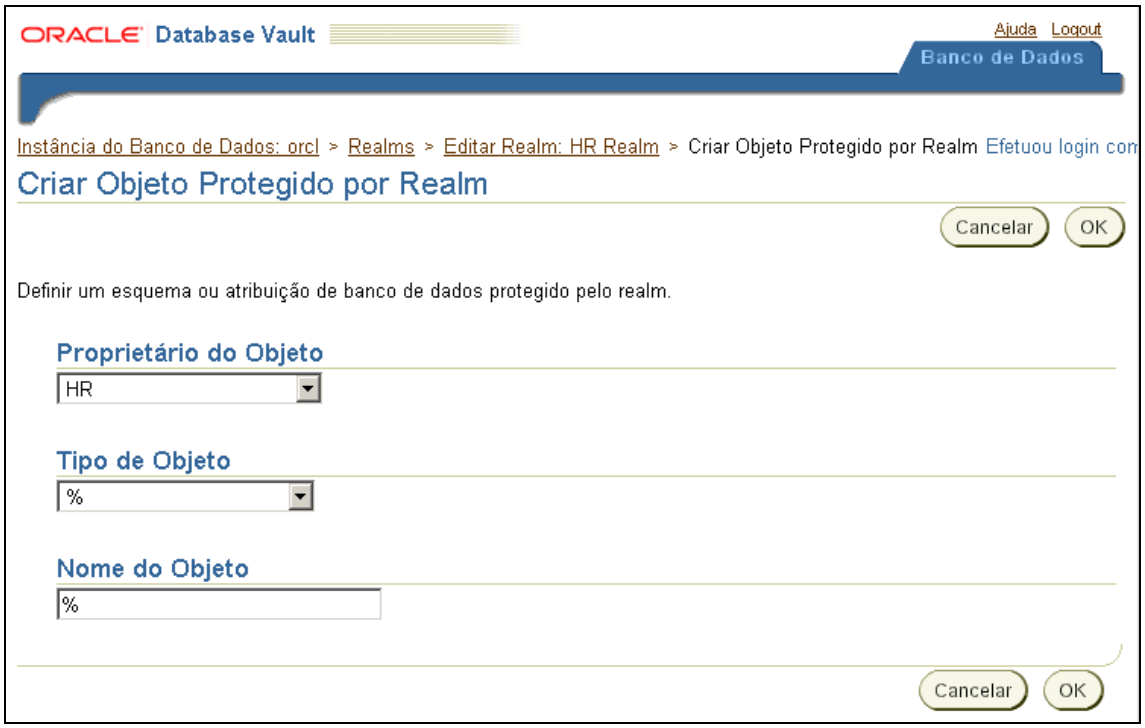


Figura 9 - Área de inclusão de objeto no *Realm*

Após clicar em OK, o esquema HR inteiro é incluído como um objeto do *Realm*. Podemos ver isso na listagem de objetos do *Realm* como mostra a Figura 10.

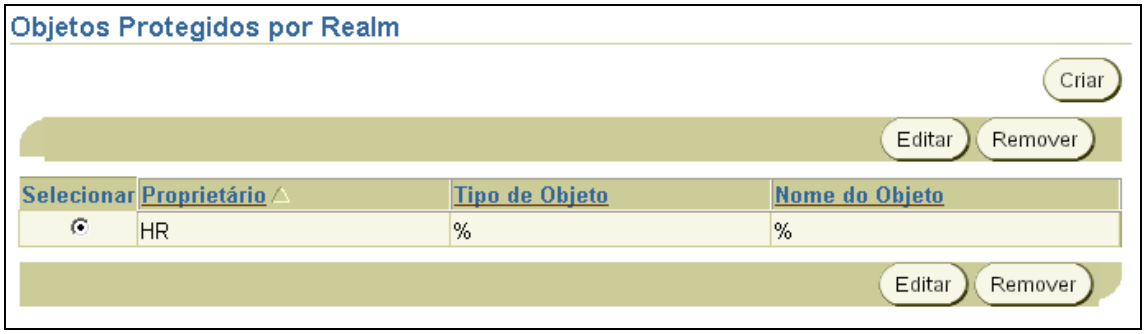


Figura 10 - Área de listagem de objetos do *Realm* com o esquema HR recém incluído

A partir desse momento, os dados do esquema HR estão bloqueados para todos os usuários do banco, menos para o usuário dono do esquema e para os usuários que

haviam recebido privilégios para algum objeto do esquema explicitamente (através do comando GRANT). Para verificar o funcionamento da segurança, a consulta sobre a tabela EMPLOYEES foi repetida com o usuário SYSTEM. O resultado da consulta pode ser visto na Figura 11. Observe que dessa vez um erro de privilégios insuficientes foi disparado impedindo a visualização dos dados. O mesmo vale para qualquer usuário poderoso que tinha acesso aos dados através do privilégio SELECT ANY TABLE.

A consulta foi então repetida com o usuário HR (Figura 12). Como este usuário é dono do esquema, ele tem acesso a todos os dados. Logo, a consulta ocorreu normalmente.

Com essa configuração o usuário privilegiado SYSTEM perde o acesso sobre os dados do esquema HR, mas mantém seus privilégios de gerenciamento no restante do banco. Contudo, isso também fez com que nenhum usuário do banco seja capaz de manipular objetos do esquema HR. Em outras palavras, mesmo o usuário HR, que é dono do esquema, não consegue mais executar operações de DDL sobre os objetos do esquema.

```
SQL> connect system
Informe a senha:*************
Conectado.
SQL> SELECT last_name, phone_number, salary
  2  FROM hr.employees
  3  WHERE employee_id < 110;
FROM hr.employees
     *
ERRO na linha 2:
ORA-01031: privilégios insuficientes
```

Figura 11 - Consulta com controle de acesso realizada pelo usuário privilegiado SYSTEM

```
SQL> connect HR
Informe a senha:*************
Conectado.
SQL> SELECT last_name, phone_number, salary
  2  FROM hr.employees
  3  WHERE employee_id < 110;

LAST_NAME                 PHONE_NUMBER              SALARY
------------------------- -------------------- ----------
King                      515.123.4567              24000
Kochhar                   515.123.4568              17000
De Haan                   515.123.4569              17000
Hunold                    590.423.4567               9000
Ernst                     590.423.4568               6000
Austin                    590.423.4569               4800
Pataballa                 590.423.4560               4800
Lorentz                   590.423.5567               4200
Greenberg                 515.124.4569              12000
Faviet                    515.124.4169               9000

10 linhas selecionadas.
```

Figura 12 - Consulta com controle de acesso realizada pelo usuário dono do esquema HR

Considere então que, após a definição do *Realm*, seja necessário criar uma nova tabela ou conceder o privilégio de SELECT na tabela EMPLOYEES para algum usuário. Nesse momento, não existe nenhum usuário capaz de realizar essa tarefa. A Figura 13

mostra a resposta do sistema para esses comandos quando executados pelo usuário SYSTEM, enquanto a Figura 14 mostra a resposta quando os comandos são executados pelo usuário HR. Em ambos os casos ocorreram erros na execução dos comandos.

```
SQL> connect system
Informe a senha:*************
Conectado.
SQL> GRANT SELECT ON HR.EMPLOYEES TO SCOTT;
GRANT SELECT ON HR.EMPLOYEES TO SCOTT
                    *
ERRO na linha 1:
ORA-00604: ocorreu um erro no nível 1 SQL recursivo
ORA-47401: Violação de Realm para grant object privilege em HR.EMPLOYEES
ORA-06512: em "DVSYS.AUTHORIZE_EVENT", line 55
ORA-06512: em line 13


SQL> CREATE TABLE HR.ADRESSES(
  2    EMPLOYEE_ID NUMBER(6,0) NOT NULL,
  3    COUNTRY VARCHAR2(30),
  4    STATE VARCHAR2(30),
  5    CITY VARCHAR2(30),
  6    ADRESS VARCHAR2(100)
  7  );
CREATE TABLE HR.ADRESSES(
*
ERRO na linha 1:
ORA-00604: ocorreu um erro no nível 1 SQL recursivo
ORA-47401: Violação de Realm para create table em HR.ADRESSES
ORA-06512: em "DVSYS.AUTHORIZE_EVENT", line 55
ORA-06512: em line 13
```

Figura 13 - Erro na execução de comando com o usuário SYSTEM sobre o *Realm* HR

```
SQL> connect hr
Informe a senha:*************
Conectado.
SQL> GRANT SELECT ON HR.EMPLOYEES TO SCOTT;
GRANT SELECT ON HR.EMPLOYEES TO SCOTT
                    *
ERRO na linha 1:
ORA-00604: ocorreu um erro no nível 1 SQL recursivo
ORA-47401: Violação de Realm para grant object privilege em HR.EMPLOYEES
ORA-06512: em "DVSYS.AUTHORIZE_EVENT", line 55
ORA-06512: em line 13


SQL> CREATE TABLE HR.ADRESSES(
  2    EMPLOYEE_ID NUMBER(6,0) NOT NULL,
  3    COUNTRY VARCHAR2(30),
  4    STATE VARCHAR2(30),
  5    CITY VARCHAR2(30),
  6    ADRESS VARCHAR2(100)
  7  );
CREATE TABLE HR.ADRESSES(
*
ERRO na linha 1:
ORA-00604: ocorreu um erro no nível 1 SQL recursivo
ORA-47401: Violação de Realm para create table em HR.ADRESSES
ORA-06512: em "DVSYS.AUTHORIZE_EVENT", line 55
ORA-06512: em line 13
```

Figura 14 - Erro na execução de comando com o usuário HR sobre o *Realm* HR

Para contornar esse problema devemos definir um usuário responsável pela manipulação de objetos do *Realm* HR. Esse usuário deve ter os privilégios para realizar as tarefas desejadas como, por exemplo, o próprio usuário HR. Essa definição deve ser feita criando uma autorização de *Realm*. Isso pode ser feito acessando a área de edição do *Realm* HR, indo à área "Autorizações de Realms", que lista as autorizações do *Realm*, e clicando no botão criar, como mostra a Figura 15.

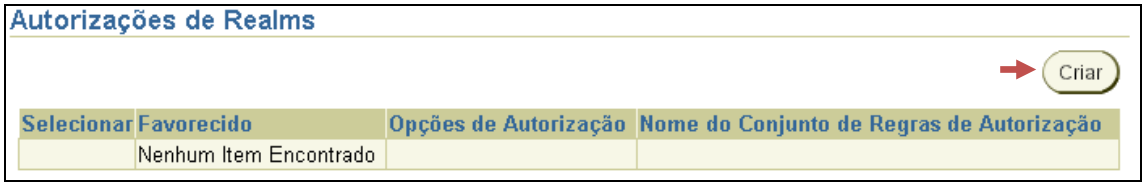


Figura 15 - Listagem de autorizações do *Realm*

Na opção "Favorecido", foi definido que o usuário que terá seus privilégios de sistema liberados para serem usados no *Realm*. Na opção "Tipo de Autorização", foi definido se esse usuário é dono do *Realm* ou participante do *Realm*. A única diferença entre o dono e participante é que o dono pode definir outros participantes. Por fim, na opção "Conjunto de Regras de Autorização", pode-se definir o conjunto de regras que deve ser validado para liberar a autorização. Neste exemplo, nenhum conjunto de regras foi selecionado, pois a autorização deve valer sempre, mas nos próximos exemplos este caso será tratado. A Figura 16 apresenta esta configuração.

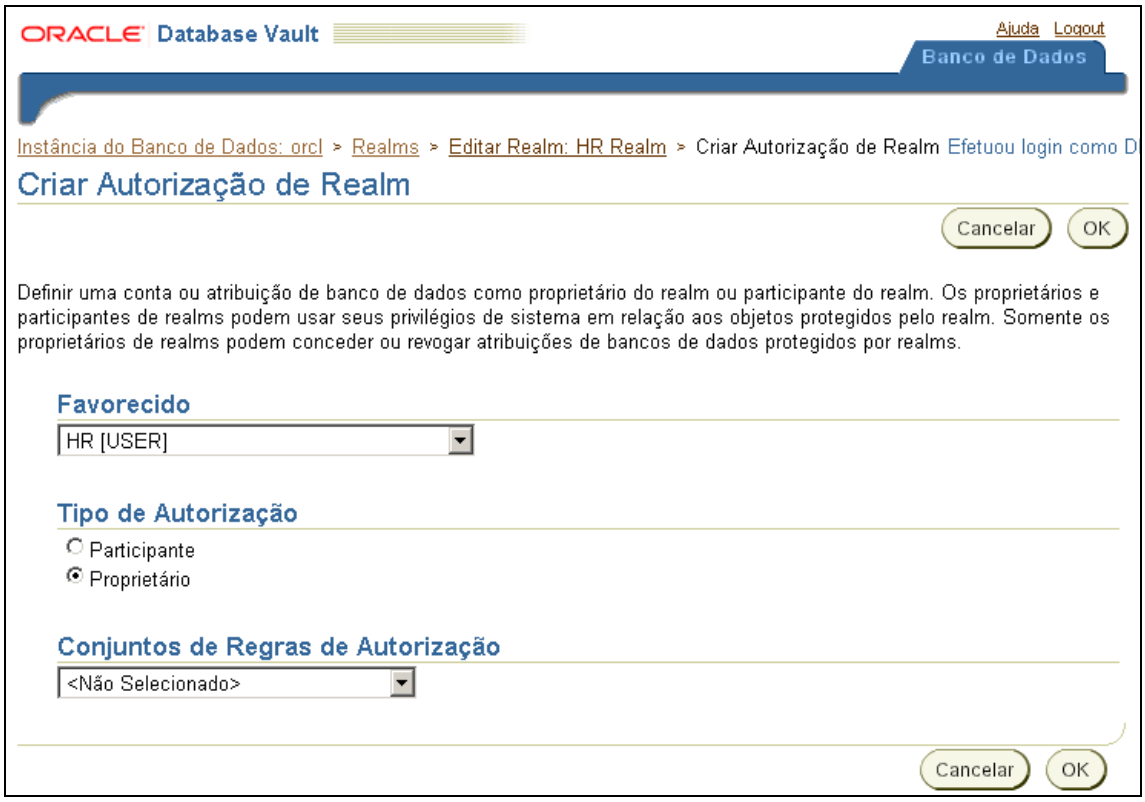


Figura 16 - Área de inclusão de autorização no *Realm*

Após clicar em OK, a autorização para o usuário HR é criada. Isso pode ser visto na listagem de autorizações do *Realm*, como mostra a Figura 17.

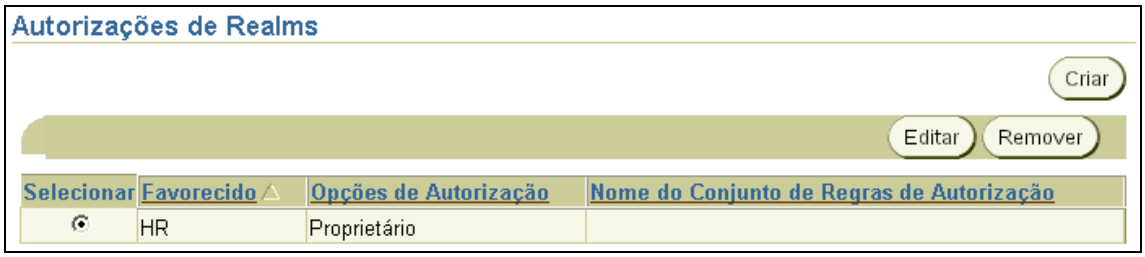


Figura 17 - Área de listagem de autorizações do *Realm* com a autorização recém incluída

Os comandos de concessão de privilégio e criação de tabela com o usuário HR foram repetidos e, desta vez, o resultado esperado foi obtido (Figura 18).

```
SQL> connect hr
Informe a senha:************
Conectado.
SQL> GRANT SELECT ON HR.EMPLOYEES TO SCOTT;

Concessão bem-sucedida.

SQL> CREATE TABLE HR.ADRESSES(
  2    EMPLOYEE_ID NUMBER(6,0) NOT NULL,
  3    COUNTRY VARCHAR2(30),
  4    STATE VARCHAR2(30),
  5    CITY VARCHAR2(30),
  6    ADRESS VARCHAR2(100)
  7  );

Tabela criada.
```

Figura 18 - Execução de comando no *Realm* HR com autorização de usuário

Nesse exemplo, foi definido um *Realm* sobre os objetos do esquema HR, bloqueando o acesso dos usuários poderosos a esses objetos, ignorando os privilégios de sistema desses usuários como, por exemplo, SELECT ANY TABLE. Além disso, os usuários que já possuíam privilégios definidos explicitamente sobre os objetos do *Realm* (através do comando GRANT) mantiveram seu acesso. Em seguida, foi definida uma autorização de *Realm* para o usuário HR, liberando que seus privilégios de sistema fossem usados sobre o *Realm*.

Ao concluir este exemplo, somente o usuário HR tem permissão para manipular os objetos do seu esquema e somente ele pode definir quais usuários podem acessar os dados, através de comandos GRANT.

3.2 Refinando privilégios do DBA em um *Realm* através de conjunto de regras

Esta seção apresenta um exemplo de refinamento de privilégios do DBA em *Realm* através de conjunto de regras. Neste caso, o acesso ao esquema HR é liberado para o usuário SYSTEM quando esse acesso for feito localmente. O exemplo contempla os seguintes passos:

1. Criar um conjunto de regras
2. Incluir regra de acesso local
3. Criar uma autorização de *Realm* baseada nesse conjunto de regras

Continuando no cenário do exemplo anterior, considere que agora deseja-se que o usuário SYSTEM manipule os objetos do *Realm* HR se o acesso for feito localmente no servidor. Para este caso, é necessário definir um conjunto de regras (veja a sessão 2.1) onde essa regra será incluída. Isso pode ser feito através da opção "Conjuntos de Regras", no menu principal do DVA, como mostra a Figura 19.

Figura 19 – Opção Conjuntos de Regras no menu principal do *Database Vault Administrator*

Para criar o conjunto de regras, selecione a opção criar acima da tabela que apresenta os conjuntos de regras já existentes, como mostra a Figura 20.

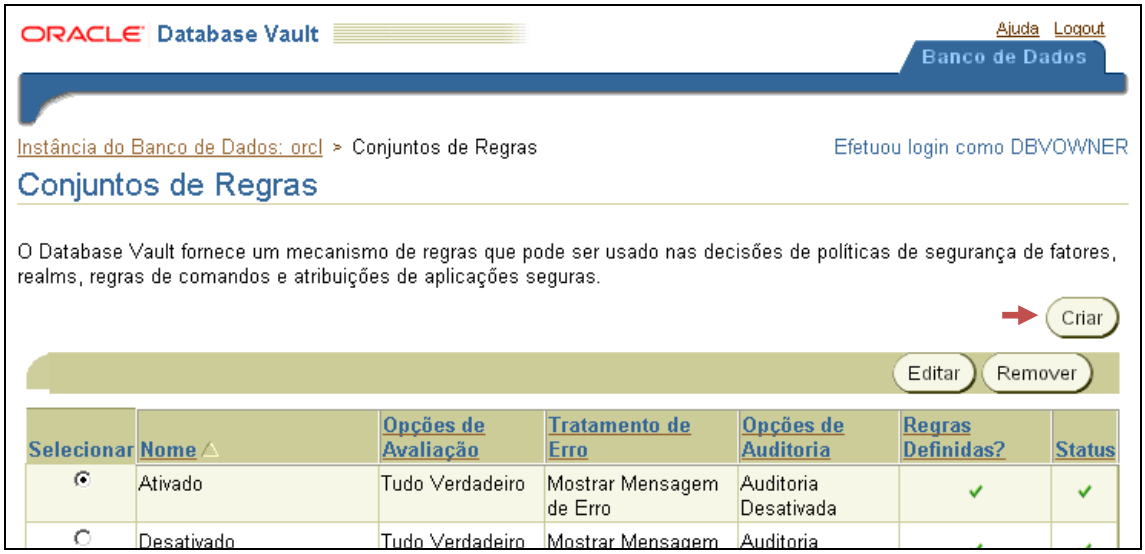


Figura 20 - Área de listagem de conjuntos de regras

Em seguida, são inseridas as informações do conjunto de regras, tais como: nome, descrição, status, opções de avaliação e opções de auditoria (Figura 21). As opções de avaliação definem como as regras são avaliadas: todas devem ser verdadeiras ou pelo menos uma deve ser verdadeira. Neste caso, foi selecionada a primeira opção. As opções de auditoria definem quando as informações de auditoria devem ser registradas: nunca, quando ocorrer alguma falha ou sempre. Nesse caso foi selecionada a segunda opção.

Figura 21 - 1ª parte da criação do conjunto de regras "Acesso Local"

Pode-se ainda definir opções de tratamento de erro para o conjunto de regras. Através dessas opções, é possível estipular um código e uma mensagem de erro para serem exibidos quando houver um erro na avaliação do conjunto de regras. Além disso, pode-se definir um *handler* para ser executado quando houver uma falha na avaliação ou sempre que as regras forem avaliadas. A Figura 22 apresenta essas configurações.

A *procedure handler* pode ser utilizada, por exemplo, para notificar via e-mail, através do pacote UTL_MAIL, as violações ocorridas. A infra-estrutura disponível para os testes realizados nesse relatório não contava com um servidor SMTP; portanto, a *procedure handler* implementada para simular a notificação, apresentada na Figura 23, cadastra as informações em uma tabela. Apesar do DVA gerar relatórios de auditoria, o *handler* tem a vantagem de disponibilizar essas informações em tempo real.

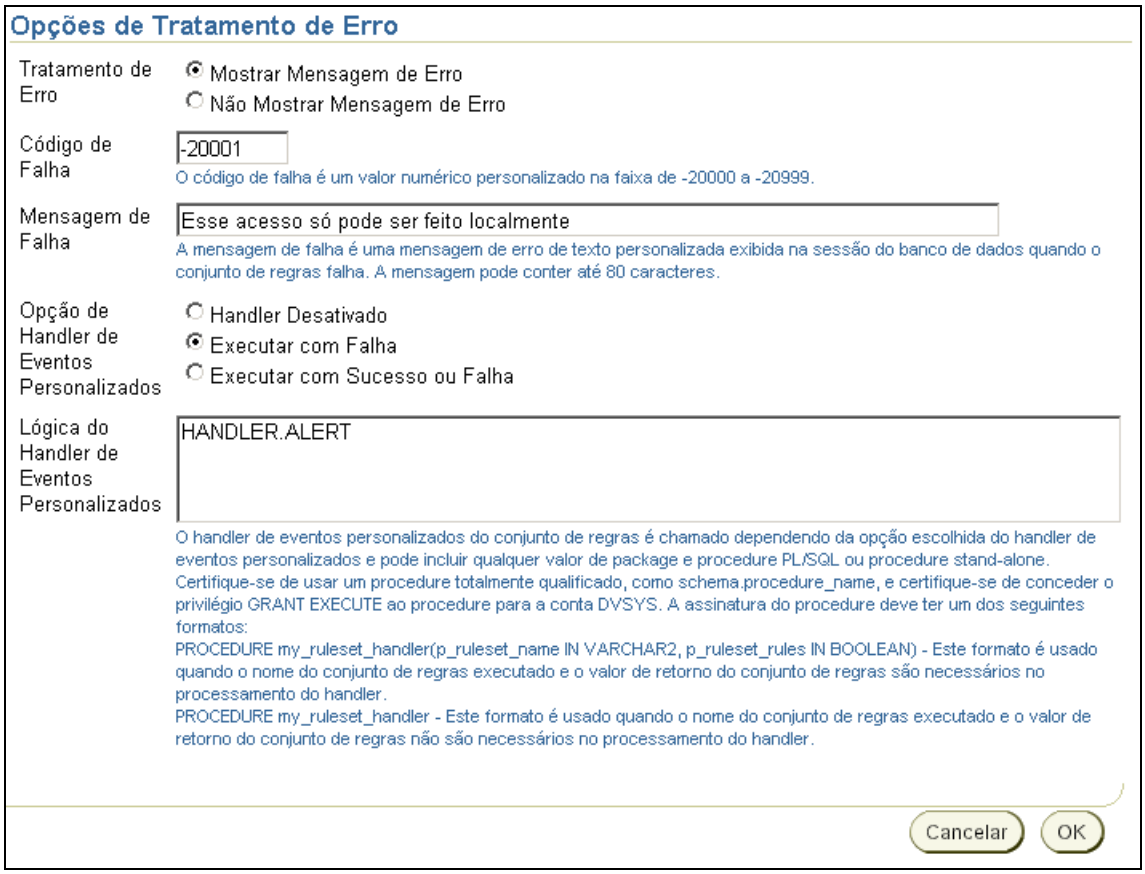


Figura 22 - 2ª parte da criação do conjunto de regras "Acesso Local"

```
create or replace
PROCEDURE ALERT AS
V_SENDER varchar2(100) := 'system@company.com';
V_RECIPIENTS varchar2(100) := 'dbadmin@company.com';
V_SUBJECT VARCHAR2(200) := 'Notificação de violação de Realm';
V_MENSAGEM varchar2(2000) := 'Tentativa de acesso externo não autorizado realizada em ';
BEGIN
  V_MENSAGEM := V_MENSAGEM || TO_CHAR(SYSDATE, 'Day DD MON, YYYY HH24:MI:SS');
  V_MENSAGEM := V_MENSAGEM || ' pelo usuário ' || SYS_CONTEXT('USERENV', 'SESSION_USER');

  INSERT INTO HANDLER.ALERT_TABLE (SENDER, RECIPIENTS, SUBJECT, MESSAGE)
  VALUES (V_SENDER, V_RECIPIENTS, V_SUBJECT, V_MENSAGEM);

  COMMIT;
END ALERT;
```

Figura 23 - *Procedure Handler*

Clicando em OK, o conjunto de regras é criado, porém ainda não possui nenhuma regra. Para incluir regras, deve-se voltar à área de listagem de conjuntos de regras apresentada na Figura 20, selecionar o conjunto de regras recém criado e clicar na opção editar, como mostra a Figura 24.

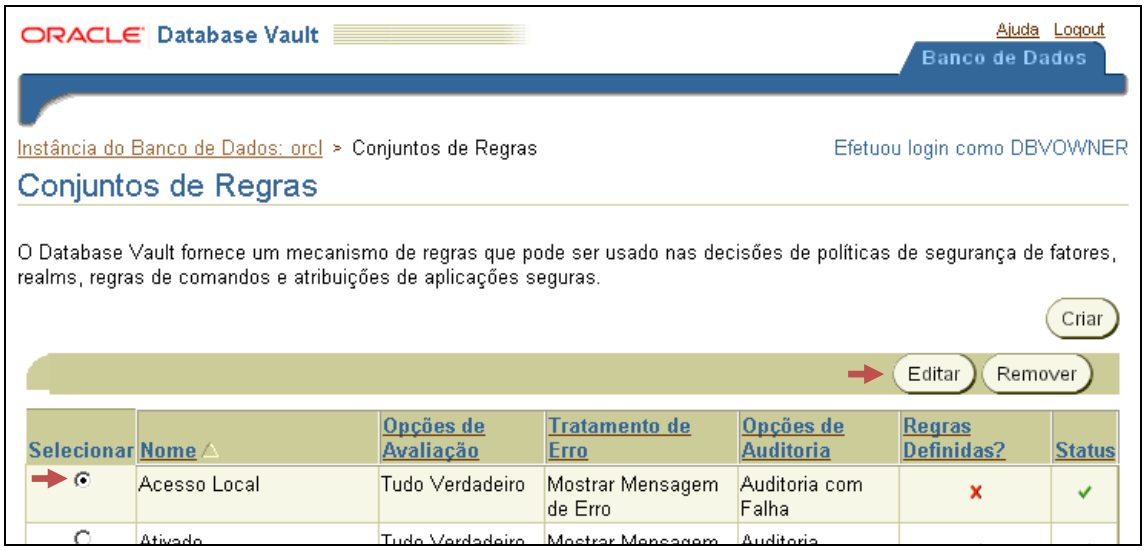


Figura 24 - Listagem dos conjuntos de regras com o conjunto recém criado

Na área de edição de conjunto de regras, pode-se alterar os dados definidos quando o conjunto foi criado bem como incluir regras ao conjunto. Para incluir a regra definida no início do exemplo, siga para a área "Regras Associadas ao Conjunto de Regras", que lista as regras pertencentes ao conjunto, e clique no botão criar, como mostra a Figura 25.

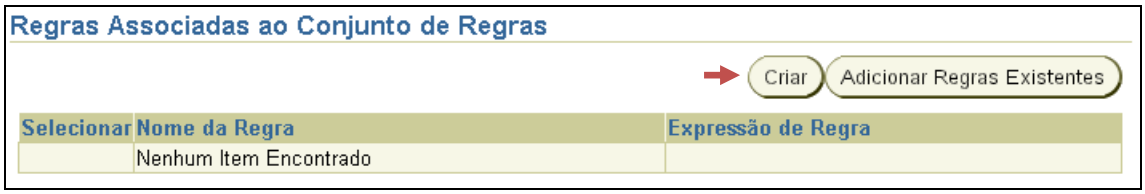


Figura 25 - Listagem de regras pertencentes ao conjunto de regras

A seguir, defina um nome e uma expressão para a regra. A expressão da regra deve retornar obrigatoriamente o valor Falso ou Verdadeiro. A expressão definida no exemplo utiliza a função do Oracle SYS_CONTEXT, acessando o contexto USERENV, que armazena diversas informações sobre a sessão do usuário. Neste caso, está sendo considerado o acesso pelo atributo IP_ADRESS, que armazena o IP de origem da conexão, e comparando com o IP do servidor (10.10.10.10). Dessa forma, a regra só é avaliada como verdadeira se a conexão partir do próprio servidor. A Figura 26 apresenta a criação da regra.

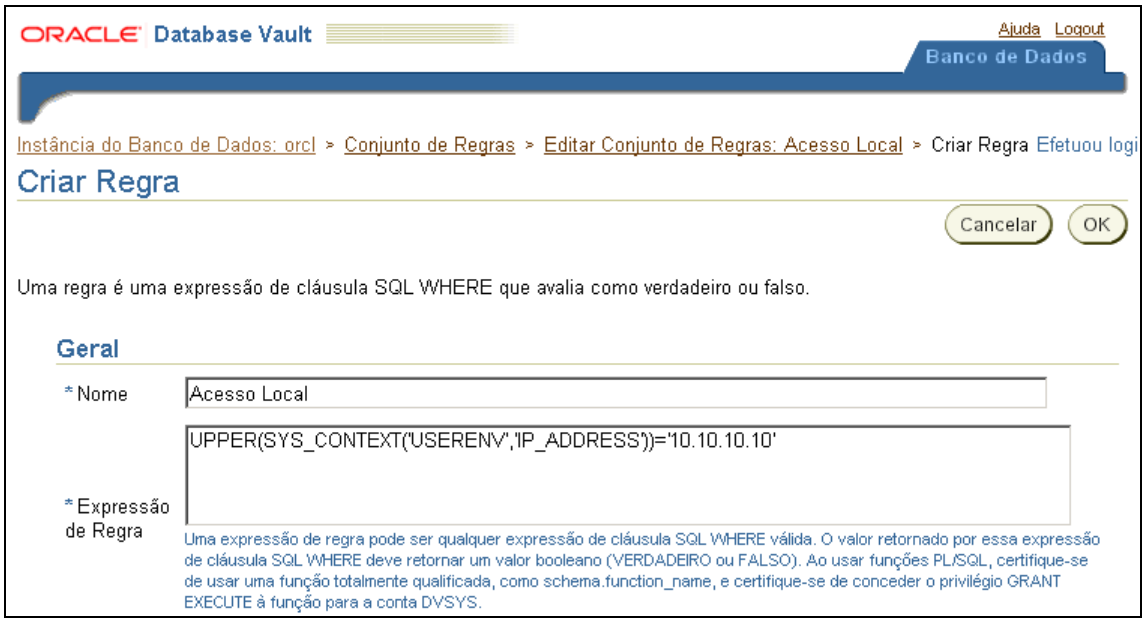


Figura 26 - Área de inclusão de regra no conjunto

Após clicar em OK, a regra é incluída no conjunto. Isto é visualizado na listagem de regras do conjunto, como exemplificado na Figura 27.

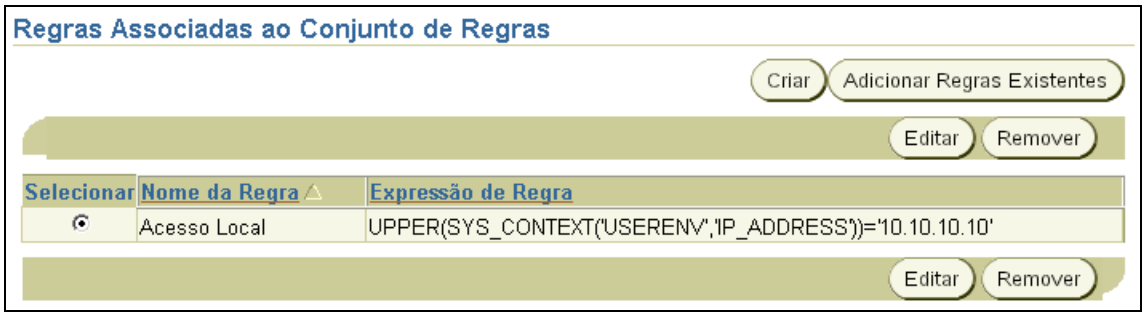


Figura 27 - Área de listagem de regras do conjunto com a regra recém incluída

O próximo passo consiste em incluir uma autorização no *Realm* HR para o usuário SYSTEM, da mesma forma que foi feito no exemplo anterior (Figura 15) para o usuário HR. Desta vez, é preenchido como "Favorecido" o usuário SYSTEM, como "Tipo de Autorização" foi selecionado"Participante" e como "Conjunto de Regras de Autorização" foi selecionado o conjunto "Acesso Local" recém criado. Isto significa que o SYSTEM só terá acesso ao *Realm* se o conjunto de regras for avaliado como verdadeiro.

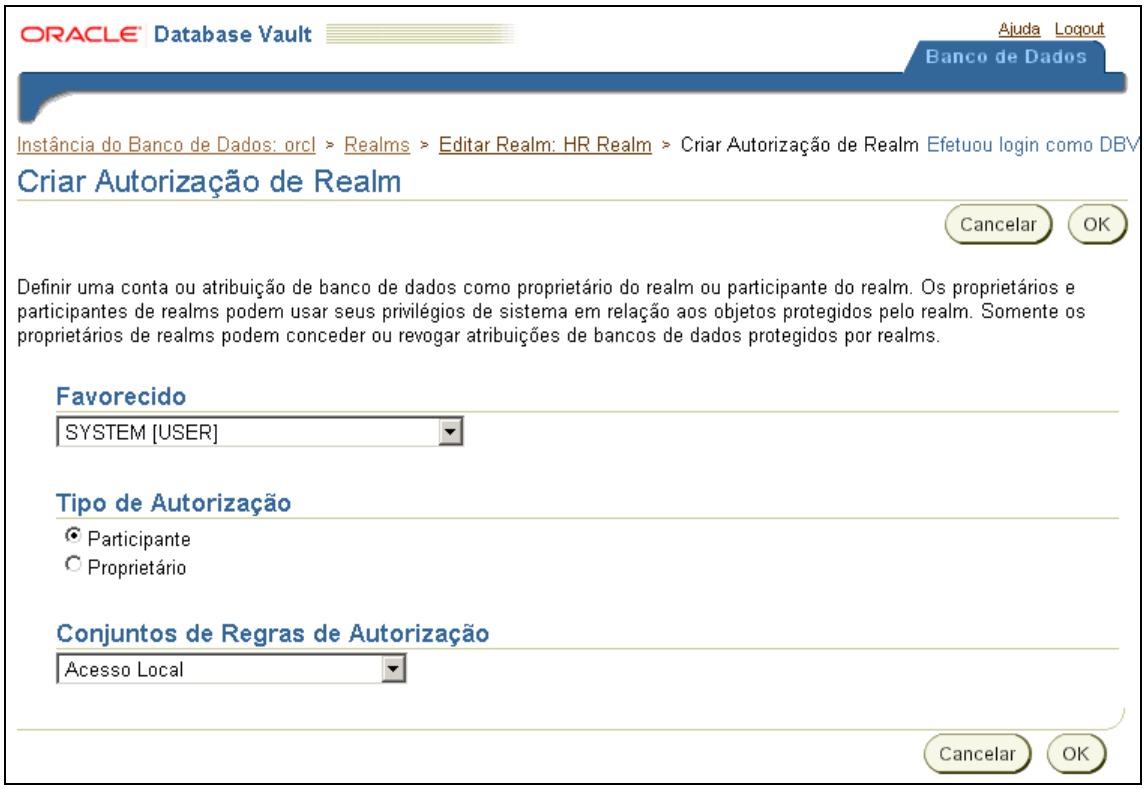


Figura 28 - Inclusão de Autorização de *Realm* para o usuário SYSTEM

A criação da autorização pode ser visualizada na listagem de autorizações do *Realm*, como mostra a Figura 29.

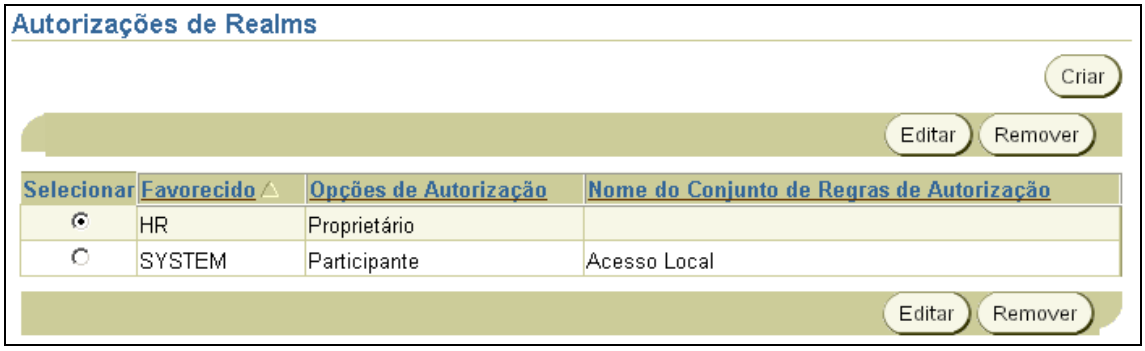


Figura 29 - Área de listagem de autorizações do *Realm* com a autorização para o usuário SYSTEM

O próximo passo deste exemplo consiste em testar as configurações realizadas. Logo, uma consulta na tabela EMPLOYEES com o usuário SYSTEM localmente e remotamente foram executadas. O resultado da primeira consulta é apresentado na Figura 30, enquanto o resultado da segunda consulta é apresentado na Figura 31.

```
SQL> connect system
Informe a senha:************
Conectado.
SQL> SELECT UPPER(SYS_CONTEXT('USERENV','IP_ADDRESS')) FROM DUAL;

UPPER(SYS_CONTEXT('USERENV','IP_ADDRESS'))
--------------------------------------------------------------------------------
10.10.10.10

SQL> SELECT last_name, phone_number, salary
  2  FROM hr.employees
  3  WHERE employee_id < 110;

LAST_NAME                 PHONE_NUMBER            SALARY
------------------------- -------------------- ----------
King                      515.123.4567             24000
Kochhar                   515.123.4568             17000
De Haan                   515.123.4569             17000
Hunold                    590.423.4567              9000
Ernst                     590.423.4568              6000
Austin                    590.423.4569              4800
Pataballa                 590.423.4560              4800
Lorentz                   590.423.5567              4200
Greenberg                 515.124.4569             12000
Faviet                    515.124.4169              9000

10 linhas selecionadas.

SQL> SELECT * from handler.alert_table;

não há linhas selecionadas
```

Figura 30 - Consulta na tabela EMPLOYEES realizada localmente pelo usuário SYSTEM

```
SQL> connect system
Informe a senha:************
Conectado.
SQL> SELECT UPPER(SYS_CONTEXT('USERENV','IP_ADDRESS')) FROM DUAL;

UPPER(SYS_CONTEXT('USERENV','IP_ADDRESS'))
--------------------------------------------------------------------------------
10.10.10.11

SQL> SELECT last_name, phone_number, salary
  2  FROM hr.employees
  3  WHERE employee_id < 110;
SELECT last_name, phone_number, salary
*
ERRO na linha 1:
ORA-01031: privilégios insuficientes


SQL> SELECT * from handler.alert_table;

SENDER
--------------------------------------------------------------------------------
RECIPIENTS
--------------------------------------------------------------------------------
SUBJECT
--------------------------------------------------------------------------------
MESSAGE
--------------------------------------------------------------------------------
system@company.com
dbadmin@company.com
Notificação de violação de Realm
Tentativa de acesso externo não autorizado realizada em Quarta-Feira  01 SET, 20
10 09:29:32 pelo usuário SYSTEM
```

Figura 31 - Consulta na tabela EMPLOYEES realizada remotamente pelo usuário

SYSTEM

Assim como esperado, a consulta retornou os dados da tabela quando realizada localmente, e um erro de privilégios insuficientes quando executada remotamente. Observe também que no primeiro caso nenhuma notificação foi cadastrada na tabela *alert_table*, enquanto no segundo caso a notificação da violação foi cadastrada normalmente.

Neste exemplo, o cenário do exemplo anterior foi evoluído para refinar os privilégios do usuário SYSTEM no *Realm* HR através de criação de uma autorização de *Realm* baseada em um conjunto de regras. Ao concluir este exemplo, o usuário SYSTEM obteve acesso aos objetos do *Realm* HR somente quando acessava o banco localmente, a partir do IP 10.10.10.10.

3.3 Controlando ações do DBA no banco com regras para comandos

Esta seção apresenta um exemplo de controle das ações do DBA no banco com regras para comandos. O exemplo contempla os seguintes passos:

1. Criar um fator que indica a hora do sistema
2. Criar um conjunto de regras e incluir a regra delimitando o horário de expediente utilizando o fator criado
3. Criar uma regra de comando para o comando ALTER SYSTEM baseada no conjunto de regras criado

Nos exemplos anteriores foi demonstrado que, através do Database Vault, é possível controlarmos as ações do DBA em áreas específicas do banco de dados. Contudo, também pode haver a necessidade de controlar alguma ação do DBA globalmente no sistema. Para isto o Database Vault permite a criação de regras para comandos (veja a seção 2.1 para explicação sobre regras para comandos). Assim pode-se aplicar uma regra que deve ser validada para que o usuário possa executar um comando específico. Este exemplo mostra como isto pode ser feito através da criação de uma regra sobre o comando ALTER SYSTEM para impedir que este comando seja realizado fora do horário de expediente (8:00 as 17:00). Para tratar este intervalo de tempo, será criado um fator (veja a sessão 2.1 para explicação sobre fatores) que represente a hora do sistema. Isto pode ser feito através da opção “Fatores” no menu principal do DVA, como mostra a Figura 32.

Figura 32 – Opção Fatores no menu principal do *Database Vault Administrator*

Para criar o fator selecione a opção criar acima da tabela que apresenta os fatores já existentes, como mostra a Figura 33.

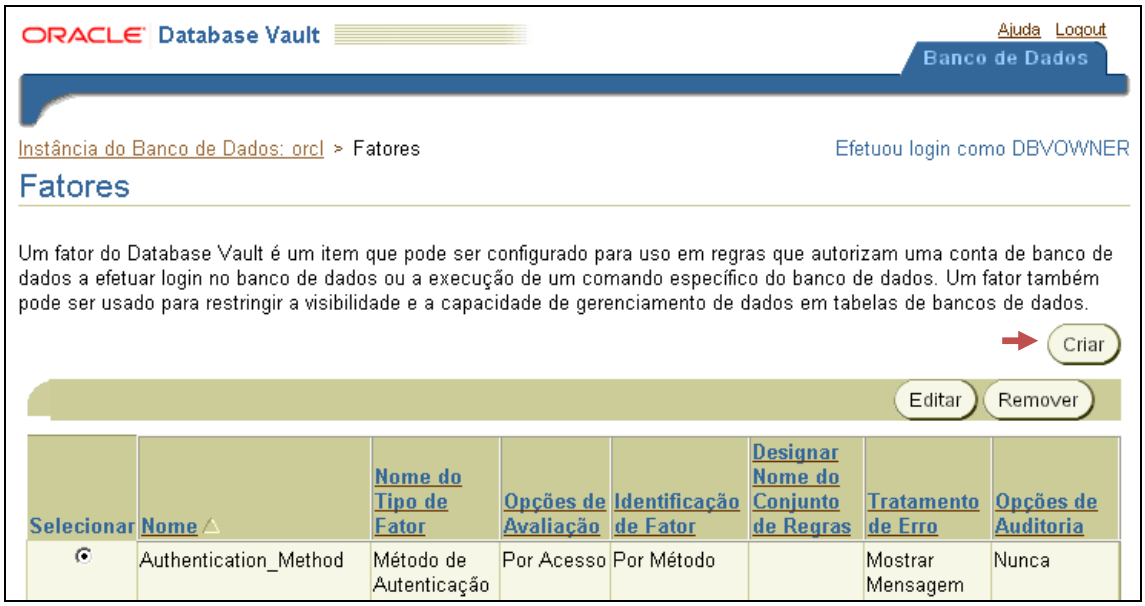


Figura 33 - Área de listagem de fatores

Em seguida, as informações do fator são inseridas, tais como: nome, descrição, tipo, identificação, avaliação, *label* e método de recuperação. Neste exemplo, foi realizada a configuração apresentada na Figura 34 e descrita a seguir.

O nome não deve possuir espaços, pois o Database Vault irá gerar um identificador para o fator que será exatamente o nome que for dado. Neste caso, o nome escolhido

foi “Hora_do_sistema” e o Database Vault gerou o identificador DVF.F$HORA_DO_SISTEMA, que será usado na definição da regra.

O tipo deve ser selecionado de uma lista de tipos pré-existentes. O tipo funciona apenas para categorizar os fatores, neste caso foi selecionado o tipo “Hora”. A identificação do fator define se é recuperado através de um método, se é uma constante ou se é definido por outros fatores, permitindo assim um relacionamento de pai e filho entre os fatores. Neste caso, a identificação selecionada foi por método, pois será utilizado um método do sistema para recuperar o fator.

A avaliação indica se o fator deve ser atualizado apenas quando uma sessão é iniciada ou toda vez que ele for referenciado. Nesse caso, como a hora do sistema muda constantemente, a avaliação por acesso foi selecionada.

A opção de *label* só é usada para integração com o Label Security. Esta integração não será tratada nesse relatório; portanto, foi selecionada a opção padrão.

Em método de avaliação, foi definida a expressão PL/SQL que recupera o fator, se ele foi definido como recuperado por método, ou o valor do fator, se ele foi definido como constante. Neste caso, foi especificada a expressão TO_CHAR(SYSDATE, 'hh24miss'). O comando SYSDATE retorna a hora atual do sistema, enquanto o método TO_CHAR permite formatar o valor de retornado. Neste caso, a formatação adotada foi horas seguidas imediatamente pelos minutos, seguidos imediatamente pelos segundos.

Figura 34 - 1ª parte da criação do fator "Hora_do_sistema"

Pode-se ainda definir para o fator um método de avaliação, um conjunto de regras de designação, opções de auditoria e opções de erro. O método de avaliação é uma expressão PL/SQL que deve retornar Verdadeiro ou Falso, e que será avaliada sempre que o valor do fator for definido. Essa expressão avalia o valor do fator e, se retornar Falso, redefine esse valor como NULL, senão mantém o valor inalterado. Em contra partida, o conjunto de regras de designação será avaliado antes da definição do valor do fator. Se esse conjunto for avaliado como verdadeiro a definição do valor prossegue, senão é impedida. Neste exemplo nenhuma dessas opções foi utilizada, pois esse fator deve ser definido sempre.

Nas opções de auditoria, pode-se definir se a auditoria não será feita nunca, sempre, ou selecionar em uma lista os casos específicos onde ela será realizada. Nas opções de erro, define-se se um erro deve ser lançado quando o fator não puder ser definido. Em ambas as opções, foram especificado os valores padrões. A Figura 35 mostra toda a configuração realizada.

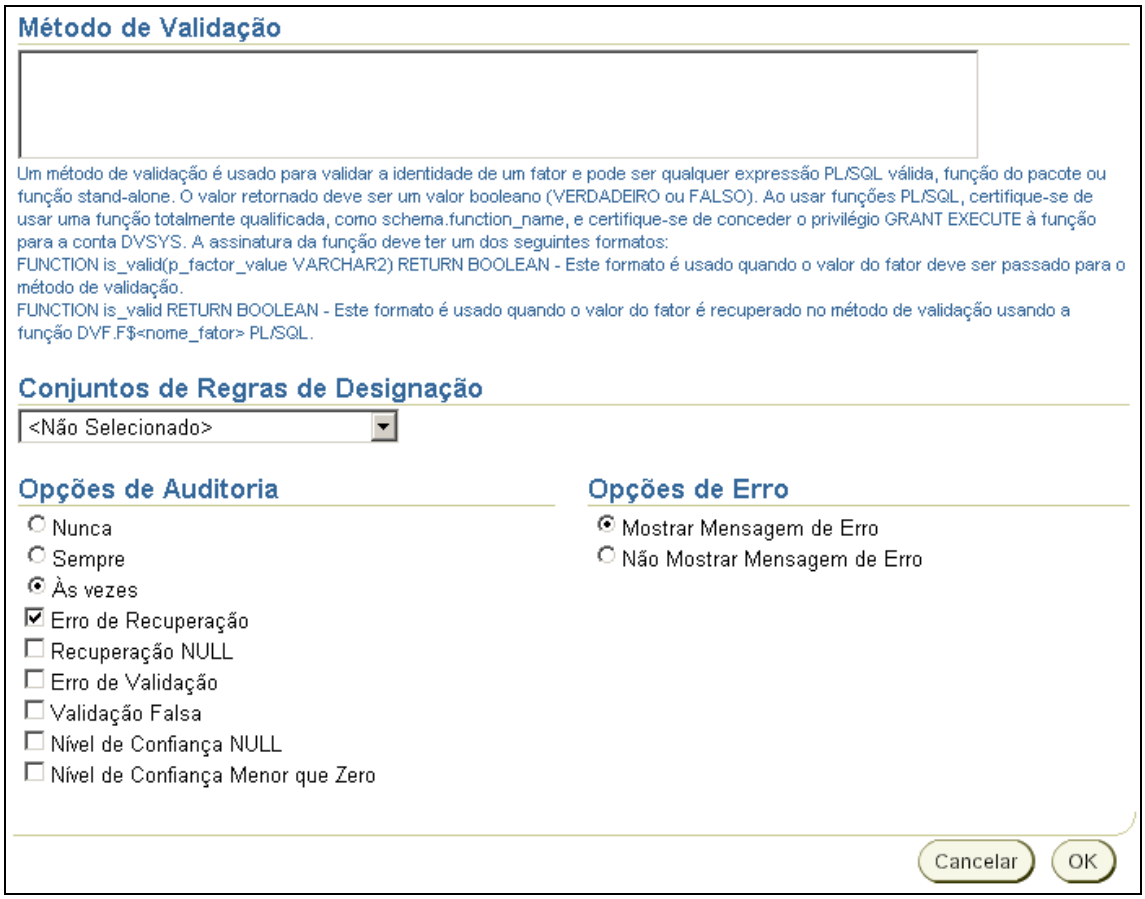
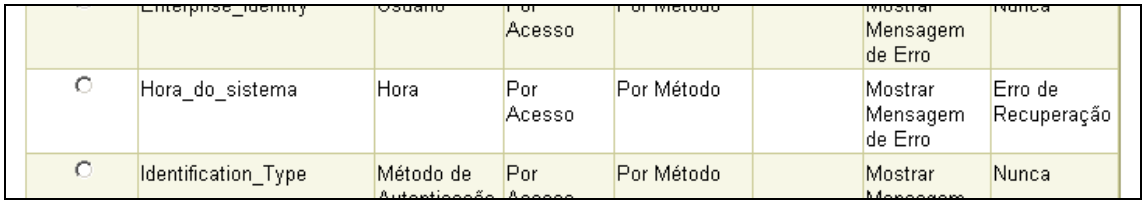


Figura 35 - 2ª parte da criação do fator "Hora_do_sistema"

Clicando em OK, o fator é criado. Isto é ilustrado na listagem de fatores, como exemplificado na Figura 36.



	Enterprise_Identity	Usuário	Por Acesso	Por Método		Mostrar Mensagem de Erro	Nunca
○	Hora_do_sistema	Hora	Por Acesso	Por Método		Mostrar Mensagem de Erro	Erro de Recuperação
○	Identification_Type	Método de Autenticação	Por Acesso	Por Método		Mostrar Mensagem	Nunca

Figura 36 - Área de listagem de fatores com o fator recém criado

Em seguida, foi criado um novo conjunto de regras onde a regra definida no início do exemplo foi incluída. O passo-a-passo para a criação de conjuntos de regra e regras é apresentado no exemplo anterior (seção 3.2).

Foi definido então o conjunto de regras "Horário de expediente" e incluída a regra apresentada na Figura 37. A expressão da regra referencia o fator recém criado através do identificador DVF.F$HORA_DO_SISTEMA e define que ele deve estar entre os valores 80000 (8:00:00) e 170000 (17:00:00).

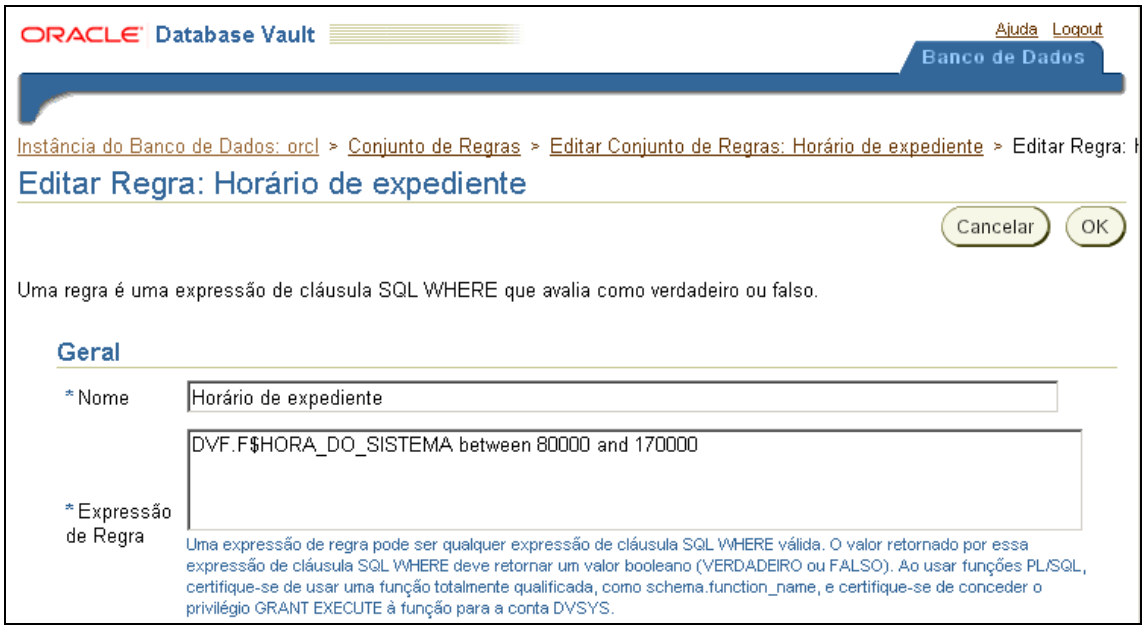


Figura 37 - Regra de horário de expediente

Em seguida, foi definida a regra de comando. Isto pode ser feito através da opção "Regras de Comandos" no menu principal do DVA, como mostra a Figura 38.

Figura 38 – Opção Regras de Comandos no menu principal do *Database Vault Administrator*

Para criar regra de comando, selecione a opção criar acima da tabela que apresenta as regras de comando já existentes, como mostra a Figura 39.

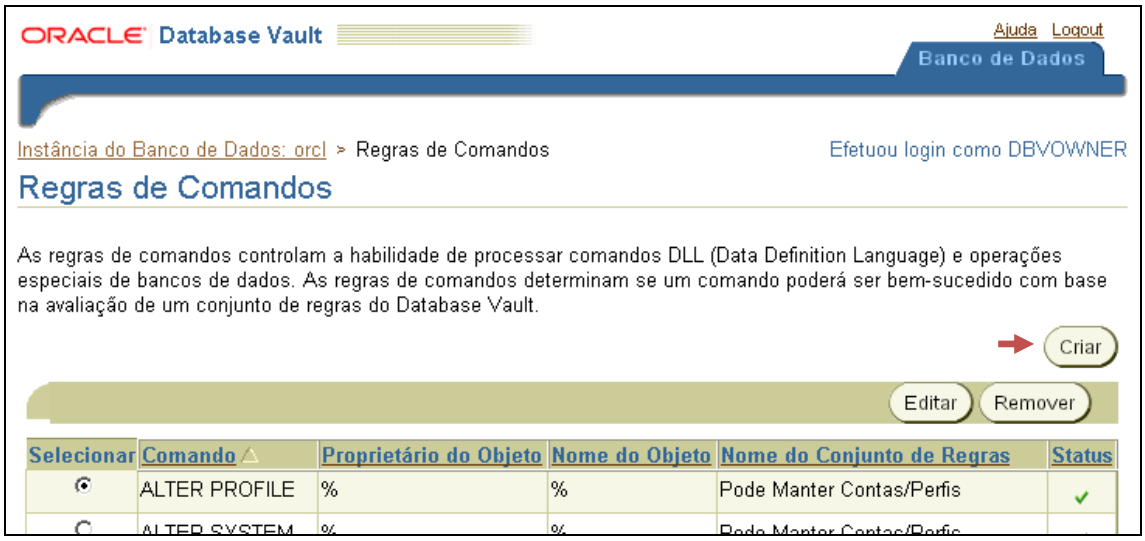


Figura 39 - Área de listagem de regras de comando

Em seguida, foram inseridas as informações da regra de comando, como comando, status, aplicabilidade e conjunto de regras. Em comando selecionamos o comando sobre o qual a regra será aplicada, nesse caso ALTER SYSTEM. Em status, foi definida a regra como ativada, e, em aplicabilidade, foram definidos os objetos sobre os quais esta regra valerá. Neste caso, foram selecionados todos os objetos do banco. Por fim, em conjunto de regras, foi definido o conjunto de regras que será aplicado sobre o comando. Neste caso, foi selecionado o conjunto "Horário de expediente" recém criado. A Figura 40 mostra toda essa configuração.

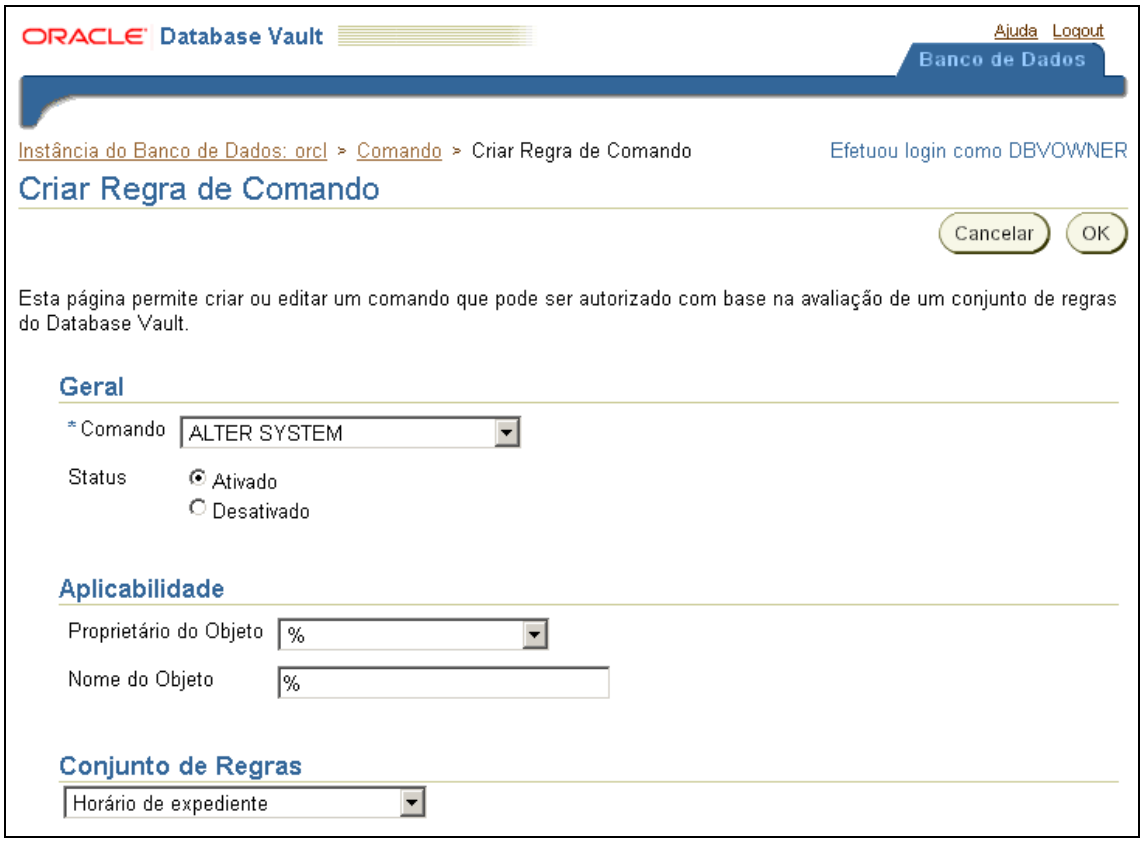


Figura 40 - criação da regra de comando sobre o comando ALTER SYSTEM

Observe que a configuração padrão do Database Vault já possui uma regra para o comando ALTER SYSTEM que utiliza o conjunto de regras "Permitir sessões". Por este motivo pode ocorrer um erro durante a criação dessa nova regra. Para contornar esse problema, pode-se incluir a regra de horário de expediente no conjunto de regras "Permitir sessões", ou alterar a regra do comando existente substituindo esse conjunto pelo conjunto "Horário de expediente".

Neste momento, a regra de horário de expediente já está ativa sobre o comando ALTER SYSTEM. Para verificar o funcionamento da regra, foi executado, com o usuário SYSTEM, um comando de alteração de arquivo de log durante o horário de expediente e fora do horário de expediente. O resultado da primeira execução é apresentado na Figura 41, enquanto o resultado da segunda execução é apresentado na Figura 42. Como esperado, o comando foi executando com sucesso durante o horário de expediente, e um erro foi lançando quando o comando foi executado fora do horário de expediente.

```
SQL> connect system
Informe a senha:*************
Conectado.
SQL> SELECT UPPER(TO_CHAR(SYSDATE, 'hh24:mi:ss')) FROM DUAL;

UPPER(TO
--------
13:16:01

SQL> ALTER SYSTEM SWITCH LOGFILE;

Sistema alterado.
```

Figura 41 - Execução do comando ALTER SYSTEM realizada em horário de expediente pelo usuário SYSTEM

```
SQL> connect system
Informe a senha:*************
Conectado.
SQL> SELECT UPPER(TO_CHAR(SYSDATE, 'hh24:mi:ss')) FROM DUAL;

UPPER(TO
--------
21:18:08

SQL> ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM SWITCH LOGFILE
*
ERRO na linha 1:
ORA-01031: privilégios insuficientes
```

Figura 42 - Execução do comando ALTER SYSTEM realizada fora do horário de expediente pelo usuário SYSTEM

Neste exemplo, o controle sobre as ações do DBA foi aumentado, incluindo uma regra sobre o comando ALTER SYSTEM que impede que esse comando seja executado fora do horário de expediente.

4 Integrando o Database Vault com o Virtual Private Database

Apesar de apresentar um conjunto de funcionalidades de segurança bastante úteis, a granularidade do controle de acesso a dados do Database Vault é muito baixa, permi-

tindo apenas o controle de acesso sobre objetos inteiros do banco. Contudo, esta granularidade pode ser aumentada através da integração do Database Vault com outras funcionalidades de controle e acesso do Oracle, como o Label Security e o Virtual Private Database.

No estudo sobre o Label Security e sobre o VPD apresentado em Azevedo *et al*. [2010], concluiu-se que o Label Security apresentava uma série de desvantagens em relação ao VPD, além de não possibilitar a aplicação de todas as regras definidas. Por esta razão, neste exemplo, o foco foi dado à integração do Database Vault com o VPD.

O gerenciamento de políticas VPD é feito através de funções do sistema pertencentes ao pacote DBMS_RLS. Portanto, o usuário que aplicará as políticas deve ter privilégio de execução sobre esse pacote. A instalação padrão do Database Vault cria uma regra de comando sobre o comando GRANT para o pacote DBMS_RLS cujo conjunto de regras (chamado "Pode Conceder Administração VPD") define que esse comando só pode ser executado pelo usuário dono do Database Vault, nesse caso o DBVOWNER. A Figura 43 apresenta essa regra de comando. A regra então restringe a execução do comando GRANT EXECUTE no *package* SYS_DBMS_RLS para ser executada apenas pelo DBVOWNER.

Figura 43 - Regra de comando padrão do Database Vault para o comando GRANT sobre o pacote DBMS_RLS

Observe que, além de possuir privilégio de execução sobre o pacote DBMS_RLS, o usuário também deve possuir o privilégio de alterar o objeto no qual a política será

aplicada. Neste caso, foi definido que o usuário responsável por aplicar as políticas seria o usuário HR, pois ele é o dono dos objetos do esquema. Como esse usuário já possui privilégio de alteração dos objetos, foi preciso apenas conceder o privilégio de execução sobre o pacote DBMS_RLS, como mostra a Figura 44.

```
SQL> connect DBVOWNER
Informe a senha:*************
Conectado.
SQL> grant execute on DBMS_RLS to HR;

Concessão bem-sucedida.
```

Figura 44 - Concessão de privilégio de execução sobre o pacote DBMS_RLS para o usuário HR

A integração do Dabatase Vault com o VPD se dá através dos fatores, de modo que uma função de política VPD pode utilizar um fator do Database Vault para criar o predicado da política. Definimos então a política onde um usuário comum pode acessar apenas as suas próprias informações na tabela EMPLOYEES, enquanto um usuário que é gerente de departamento pode visualizar as informações de todos os empregados do seu departamento.

Para exemplificar esta integração, foi criado um fator que define o departamento gerenciado por um usuário. O método utilizado para definir o valor do fator (veja sessão 3.3) é apresentado na Figura 45.

```
create or replace
FUNCTION F_GET_MANAGED_DEPARTMENT(USER_NAME IN VARCHAR2) RETURN VARCHAR2 AS
V_MANAGED_DEPARTMENT_NAME VARCHAR2 (30) default NULL;
BEGIN
  SELECT D.DEPARTMENT_NAME INTO V_MANAGED_DEPARTMENT_NAME
  FROM HR.DEPARTMENTS D
  INNER JOIN HR.EMPLOYEES E ON D.MANAGER_ID = E.EMPLOYEE_ID
  WHERE E.EMAIL = USER_NAME;

  RETURN V_MANAGED_DEPARTMENT_NAME;
END F_GET_MANAGED_DEPARTMENT;
```

Figura 45 - Método de definição de valor do fator Departamento Gerenciado

O método recebe o nome do usuário logado no banco e retorna o nome do departamento gerenciado por esse usuário. Para este exemplo, por simplificação, foi considerado que o *login* do usuário no banco é o mesmo e-mail cadastrado na tabela EMPLOYEES. Se o usuário não gerencia nenhum departamento, então a consulta não retorna nenhum valor e, conseqüentemente, o método retorna o valor NULL.

O método foi criado pelo usuário HR que concedeu privilégio de execução do método para o usuário DVSYS, que por sua vez é o responsável por recuperar os valores dos fatores para cada usuário que se conecta ao banco. O formulário de criação do fator é apresentado na Figura 46.

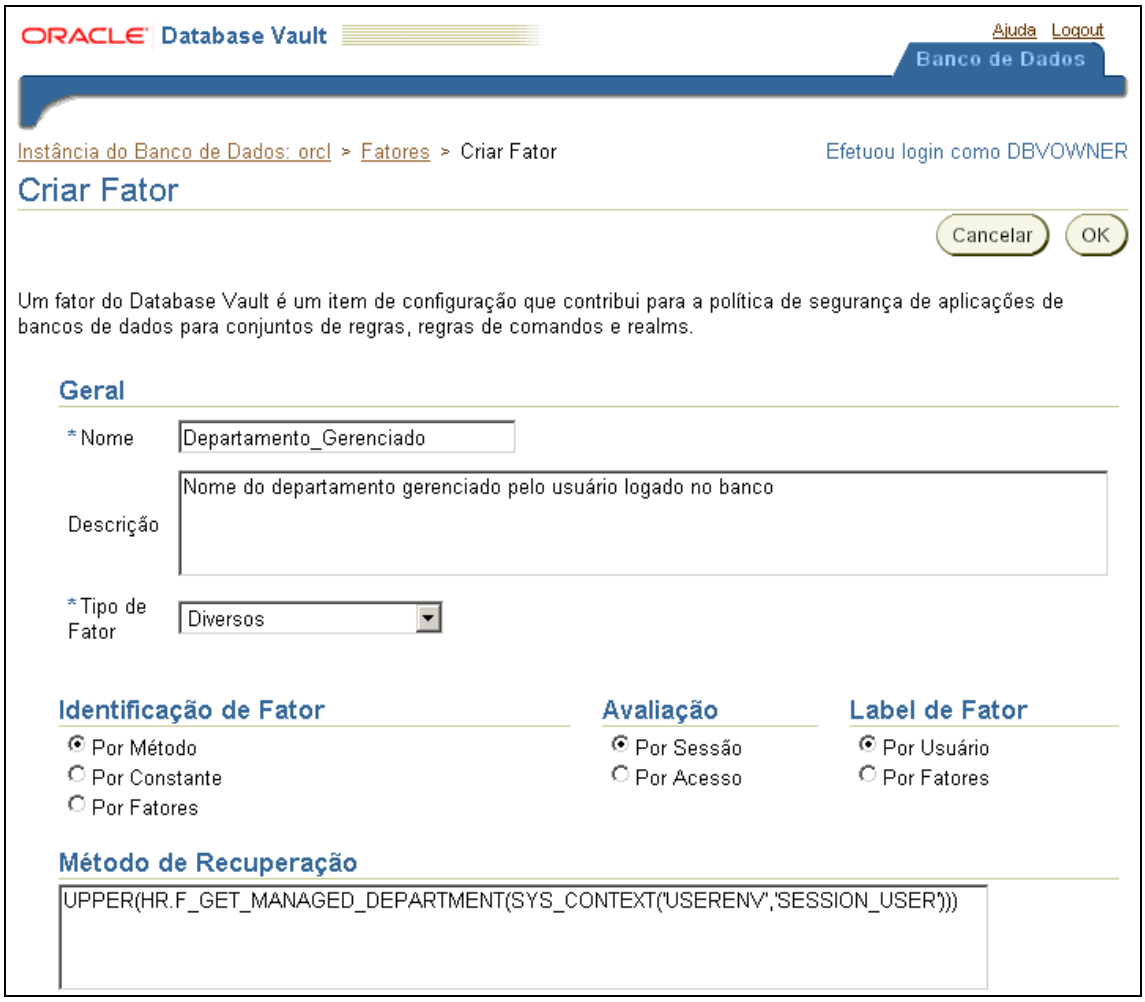


Figura 46 - Criação do fator Departamento Gerenciado

Observe que o usuário logado foi passado para o método de recuperação através da função SYS_CONTEXT, dessa forma garante-se que o retorno do método F_GET_MANAGED_DEPARTMENTS estará sempre de acordo com o usuário logado. A partir desse momento o valor do fator pode ser recuperado através do comando DVF.F$DEPARTAMENTO_GERENCIADO.

Em seguida, foi criada a função da política VPD para ser aplicada à tabela EMPLOYEES. A função criada é apresentada na Figura 47.

```
create or replace
FUNCTION F_POLICY(P_SCHEMA IN VARCHAR2, P_TAB IN VARCHAR2) RETURN VARCHAR2 AS
V_MANAGED_DEPARTMENT_ID NUMBER;
V_RETURN VARCHAR2 (100);
BEGIN
  IF DVF.F$DEPARTAMENTO_GERENCIADO IS NULL THEN
    V_RETURN := 'EMAIL = ''' || SYS_CONTEXT('USERENV','SESSION_USER') || '''';
  ELSE
    SELECT D.DEPARTMENT_ID INTO V_MANAGED_DEPARTMENT_ID
    FROM HR.DEPARTMENTS D
    WHERE UPPER(D.DEPARTMENT_NAME) = DVF.F$DEPARTAMENTO_GERENCIADO;

    V_RETURN := 'DEPARTMENT_ID = ''' || V_MANAGED_DEPARTMENT_ID || '''';
  END IF;

  RETURN V_RETURN;
END F_POLICY;
```

Figura 47 - Função da política VPD para restrição de acesso aos dados dos funcionários

O primeiro passo executado na função da política é verificar o valor do fator Departamento Gerenciado. Se o valor do fator for nulo, então o usuário não é gerente de departamento, portando o predicado é definido como "EMAIL = LOGIN_DO_USUARIO", liberando o acesso somente aos seus próprios dados. Se o valor do fator não for nulo significa que o usuário é gerente de departamento, portanto é realizada uma consulta para recuperar o ID do departamento e o predicado é definido como "DEPARTMENT_ID = ID_DEPARTAMENTO_GERENCIADO", liberando o acesso somente aos dados dos usuários do seu departamento.

Como a função é aplicada sobre a tabela EMPLOYEES, ela é invocada sempre que uma consulta é realizada sobre a tabela. Contudo, observe que o método que define o fator Departamento Gerenciado realiza uma consulta na tabela EMPLOYEES. Conseqüentemente esta consulta sempre é feita com o valor do fator ainda indefinido (NULL). Neste caso, a política sempre vai retornar o predicado de usuário comum, mesmo que o usuário seja gerente de departamento. Contudo, isso não afeta a definição do valor do fator, para isso só é preciso justamente a informação sobre o próprio usuário.

Antes de aplicar a política sobre a tabela EMPLOYEES, uma consulta foi realizada com dois usuários: JCHEN que é um funcionário comum do departamento financeiro e NGREENBE que é a gerente desse mesmo departamento. A Figura 48 apresenta o resultado da consulta realizada pelo usuário JCHEN, enquanto a Figura 49 apresenta o resultado da consulta realizada pelo usuário NGREENBE.

```
SQL> connect JCHEN
Informe a senha:*************
Conectado.
SQL> SELECT first_name, last_name, salary
  2  FROM hr.employees;

FIRST_NAME           LAST_NAME                      SALARY
-------------------- ------------------------- ----------
Donald               OConnell                         2600
Douglas              Grant                            2600
Jennifer             Whalen                           4400
Michael              Hartstein                       13000
Pat                  Fay                              6000
Susan                Mavris                           6500
Hermann              Baer                            10000
                           .....
Timothy              Gates                            2900
Randall              Perkins                          2500
Sarah                Bell                             4000
Britney              Everett                          3900
Samuel               McCain                           3200
Vance                Jones                            2800
Alana                Walsh                            3100
Kevin                Feeney                           3000

107 linhas selecionadas.
```

Figura 48 - Consulta sobre a tabela EMPLOYEES realizada sem controle de acesso pelo usuário JCHEN

```
SQL> connect NGREENBE
Informe a senha:*************
Conectado.
SQL> SELECT first_name, last_name, salary
  2  FROM hr.employees;

FIRST_NAME           LAST_NAME                      SALARY
-------------------- ------------------------- ----------
Donald               OConnell                         2600
Douglas              Grant                            2600
Jennifer             Whalen                           4400
Michael              Hartstein                       13000
Pat                  Fay                              6000
Susan                Mavris                           6500
Hermann              Baer                            10000
                           .....
Timothy              Gates                            2900
Randall              Perkins                          2500
Sarah                Bell                             4000
Britney              Everett                          3900
Samuel               McCain                           3200
Vance                Jones                            2800
Alana                Walsh                            3100
Kevin                Feeney                           3000

107 linhas selecionadas.
```

Figura 49 - Consulta sobre a tabela EMPLOYEES realizada sem controle de acesso pelo usuário NGREENBE

Ambos os usuário obtiveram acesso a todos os registros da tabela EMPLOYEE. Em seguida, a política foi aplicada sobre a tabela através do comando apresentado na Figura 50.

```
SQL> connect HR
Informe a senha:************
Conectado.
SQL> EXECUTE DBMS_RLS.ADD_POLICY('HR', 'EMPLOYEES', 'POLICY_MANAGED_EMPLOYEES', 'HR', 'F_POLICY');

Procedimento PL/SQL concluído com sucesso.
```

Figura 50 - Comando de aplicação da política sobre a tabela EMPLOYEES

A consulta foi então repetida por ambos os usuários. A Figura 51 apresenta o resultado da consulta realizada com controle de acesso pelo usuário JCHEN. Neste caso, foram retornadas apenas as informações do próprio JCHEN, pois ele não é gerente do departamento. A Figura 52 apresenta o resultado da consulta realizada com controle de acesso pelo usuário NGREENBE. Como este usuário é gerente, foram retornadas para ele as informações de todos os empregados do departamento.

```
SQL> connect JCHEN
Informe a senha:************
Conectado.
SQL> SELECT first_name, last_name, salary
  2  FROM hr.employees;

FIRST_NAME           LAST_NAME                      SALARY
-------------------- ------------------------- ----------
John                 Chen                             8200
```

Figura 51 - Consulta sobre a tabela EMPLOYEES realizada com controle de acesso pelo usuário JCHEN

```
SQL> connect NGREENBE
Informe a senha:************
Conectado.
SQL> SELECT first_name, last_name, salary
  2  FROM hr.employees;

FIRST_NAME           LAST_NAME                      SALARY
-------------------- ------------------------- ----------
Nancy                Greenberg                       12000
Daniel               Faviet                           9000
John                 Chen                             8200
Ismael               Sciarra                          7700
Jose Manuel          Urman                            7800
Luis                 Popp                             6900

6 linhas selecionadas.
```

Figura 52 - Consulta sobre a tabela EMPLOYEES realizada com controle de acesso pelo usuário NGREENBE

Neste exemplo, foi apresentado um exemplo de integração das funcionalidades Database Vault e Virtual Private Database. Foi criado um fator do Database Vault, e ele foi utilizado para aplicar uma política definida através do VPD.

5 Conclusões

Este relatório apresentou um estudo da ferramenta Database Vault da Oracle. A ferramenta foi caracterizada, apresentando os conceitos relacionados aos componentes de controle de acesso, ferramentas para configuração e administração, esquemas utilizado, APIs, ferramentas de monitoração de relatório, forma como os dados são protegidos.

Exemplos práticos e com passo-a-passo detalhados de uso do DBVault em circunstâncias comuns de necessidade de controle de acesso nas organizações foram apresentadas, tais como:

Restrição do acesso de DBAs a dados sensíveis através da criação de *Realms* utilizando a ferramenta *Database Vault Administrator*: Neste exemplo, o esquema HR foi considerado sensível e foi incluído em um *Realm*. A partir desse momento, o acesso aos dados do esquema HR ficaram bloqueados para todos os usuários do banco, menos para o usuário dono do esquema e para os usuários que haviam recebido privilégios para algum objeto do esquema explicitamente (através do comando GRANT). Em seguida, uma autorização de *Realm* foi criada para o usuário HR a fim de que ele pudesse executar comandos DDL (Data Definition Language).

Uso de conjunto de regras para refinar privilégios de DBA: Neste exemplo, foi habilitado, via conjunto de regras, a manipulação de dados pelo usuário SYSTEM, desde que ele acesse o banco de dados através do IP 10.10.10.10, que é o IP local (do computador onde o banco de dados está instalado). Foi apresentada também a definição de mensagens de erro, caso ocorra falha quando da avaliação do conjunto de regras. O IP do computador logando ao banco foi obtido através do uso de SYS_CONTEXT.

Definição de regras para comando: Neste caso, foi criada um conjunto de regras para ser avaliado quando o usuário tenta executar um comando no banco de dados. Se o conjunto de regra retornar falso, o usuário não pode executar o comando. Neste caso, foi criada regra sobre o comando ALTER SYSTEM para impedir que este comando seja realizado fora do horário de expediente (8:00 as 17:00). Um fator foi criado para retornar a hora do sistema.

Por fim, foi apresentado um exemplo de integração do Database Vault com Virtual Private Database. A integração apresentada considerou o uso de fatores para auxiliar na criação da política. Foi criado um fator para retornar o departamento do usuário logado e este fator foi utilizado na criação da política.

A partir dos exemplos apresentados, o Database Vault demonstrou-se como uma ferramenta fácil de usar e que requer conhecimento de PL/SQL apenas para criar expressões e invocar procedures, além do conhecimento do negócio e das necessidades de segurança para criar regras. No entanto, observou-se também que a ferramenta é muito mais apropriada para definir controle de acesso a usuários privilegiados do banco de dados do que para os usuários comuns. Isto porque o Database Vault tem uma granularidade de controle de acesso baixa (grossa), dado que é aplicado a objetos inteiros, comandos de DDL etc. Para definir autorização de acesso para usuários do negócio, a granularidade deve ser alta (fina) a fim de definir acesso a conjunto de tuplas e até mesmo a conjunto de colunas de determinadas tuplas. O VPD tem esta granularidade de controle de acesso.

No exemplo de integração entre Database Vault e VPD, foi utilizada a funcionalidade para criação de fatores, e o fator criado foi utilizado dentro da procedure do VPD. Uma vantagem no uso do fator é que o fator pode ser reutilizado em diferentes

procedures e, se for necessário realizar alguma alteração, sua definição será alterada em todas as procedures de autorização que o utilizam. Por outro lado, isto não justifica o uso do Database Vault, pois é um uso ainda limitado. Neste caso, poderia ser criada uma visão para retornar o valor do fator, surtindo o mesmo efeito. Além disso, o exemplo elaborado inclui uma recursão da procedure de autorização na mesma tabela que não é algo simples de lidar, mas que tem que ser considerado quando fatores forem criados considerando informação da própria tabela onde a política VPD está sendo aplicada.

Como trabalhos futuros, sugere-se uma investigação mais aprofundada, inclusive considerando exemplos reais, a fim de verificar outros usos do Database Vault para controle de acesso fino.

6 Referências Bibliográficas

AZEVEDO, L.; PUNTAR, S.; MELO, R.; BAIAO, F.; CAPPELLI, C. **Avaliação Prática de Funcionalidades para Autorização de Informações (Label Security e Virtual Private Database)**. Relatórios Técnicos do DIA/UNIRIO (RelaTe-DIA), RT-0002/2010, 2010. Disponível em: http://seer.unirio.br/index.php/monografiasppgi.

DoD. **Trusted Computer Security Evaluation Criteria. Department of Defense**. DoD 5200.28-STD, 1983.

HUEY, P.; BADNAR, P.; BEDNAR, T. *et al*. **Oracle Database Vault Administrator's Guide 10g Release 2**. Oracle, 2009

ORACLE. **Restricting Privileged Users from Accessing Private Application Data Using Oracle Database Vault**. Disponível em: http://www.oracle.com/technology/obe/10gr2_db_single/security/dv/datavault_otn.htm Acesso em: 15 jul. 2010a

ORACLE. **Restricting Command Execution Using Oracle Database Vault**. Disponível em: http://www.oracle.com/technology/obe/10gr2_db_single/security/dv/datavault2_otn.htm Acesso em: 15 jul. 2010b