GRADUAÇÃO

UniCesumar

Banco de Dados II

Professor Msc. Aparecido Vilela Junior

aparecido.vilela@unicesumar.edu.br

GRADUAÇÃO

UniCesumar

Segurança e Auditoria de BD Parte/01

Aparecido Vilela Junior

Segurança e Auditoria de BD

- Para proteger um dos recursos mais vitais de uma empresa – seus dados – um DBA precisa estar profundamente ciente da maneira como o Oracle protege os dados corportativos e as diferentes ferramentas que ele têm a sua disposição.
- As ferramentas e mecanismos oferecidos pelo Oracle divide-sem em três amplas categorias: Autenticação, Autorização e Auditoria.

Autenticação

- Autenticação inclui os métodos utilizados para identificar quem está acessando o banco de dados, assegurando que você é que diz ser, independente dos recursos que solicita do BD.

Autorização

- Fornece acesso a vários objetos no BD depois que você é autenticado pelo BD.
- Há métodos de autorização adicional disponíveis aos administradores de BD, devido ao poder extremo que um DBA tem.
- A autorização vai muito além do simples acesso a uma tabela ou relatório; ela também inclui os direitos de uso dos recursos do sistema no BD e privilégios para realizar certas ações no BD.

Auditoria

- Inclui diferentes níveis de monitoramento no BD. Em um nível alto, a auditoria pode registrar tanto tentativas bem-sucedidas de login como malsucedidas, acessar um objeto ou realizar uma ação.
- Os DBAs precisam utilizar a auditoria criteriosamente para que não haja um numero excessivo de registros de auditoria ou muito overhead implemenando uma auditoria contínua
- Por outro lado, a auditoria pode ajudar a proteger o patrimônio da empresa, monitorando quem utiliza mais recursos, a que horas e com qual frequencia.

Autenticação do DBA

- O Bd nem sempre está disponível para autenticar um BDA, quando ele está parado devido a uma falha não prevista ou para realizar um backup off-line no BD.
- Para resolver essa situação o Oracle utiliza um arquivo de senhas para manter uma lista de usuários do BD que têm permissão para realizar funções como iniciar e desligar o BD, iniciar Backups, etc.

Autenticação do DBA

- Há dois privilégios de sistema em particular que fornecem aos administradores autenticação especial no banco de dados: SYSDBA e SYSOPER.
- Um administrador com o privilégio SYSOPER pode iniciar e desligar o BD, realizar backups on-line ou off-line, arquivar os arquivos redo log atuais e conectar-se ao BD quando ele está no modo RESTRICTED SESSION.
- O privilégio SYSDBA contém todos os direitos de SYSOPER, mais a capacidade de criar um BD e conceder o privilégio SYSDBA ou SYSOPER a outros usuários do BD.

Autenticação do DBA

- Para conectar-se ao BD a partir de uma sessão SQL*PLUS, acrescente AS SYSDBA ou AS SYSOPER ao seu comando connect.
  - Sqlplus /nolog
  - connect scott/tiger as sysdba;
  - Show user;
- Além dos privilégios adicionais disponíveis a usuários que se conectam como SYSDBA ou SYSOPER, o esquema padrão também é diferente para esses usuários quando eles se conectam ao BD.
- Os usuários que se conectam com o privilégio SYSDBA o fazem como usuário SYS; o privilégio SYSOPER configura o arquivo como PUBLIC;

Contas de usuários

- A fim de ganhar acesso ao BD, um usuário deve fornecer um nome de usuário para acessar os recursos associados com essa conta.
- Cada nome de usuário deve ter uma senha e estar associado a um e somente um esquema no BD; algumas contas talvez não tenham objeto no esquema, mas, em vez disso, receberiam os privilégios para que essa conta acesse objetos em outros esquemas.

Criando usuários

- O comando create user é relativamente simples e direto. Ele tem alguns parâmetros:

Parâmetro	Uso
Nome do usuário	Nome do esquema.
IDENTIFIED {BY senha} | EXTERNALLY | GLOBALLY AS 'nomeexterno'	Especifica a maneira como o usuário será autenticado: BD com senha, SO (local ou remoto; ou por serviço OID
DEFAULT TABLESPACE espaço de tabela	Espaço de tabela em que os objetos permanentes são criados.
TEMPORARY TABLESPACE	Espaço de tabela
QUOTA {tamanho |UNLIMITED} ON espaço	Qtde de espaço permitida
PROFILE perfil	Perfil atribuído ao usuário
PASSWORD EXPIRE	Primeiro logon, altera senha
ACCOUNT {LOCK|UNLOCK}	Bloqueada ou desbloqueada

Criando usuários

- No exemplo a seguir, criaremos um usuário (AUDITOR:
  - CREATE USER AUDITOR IDENTIFIED BY AUDITOR
  - ACCOUNT UNLOCK
  - DEFAULT TABLESPACE USERS
  - TEMPORARY TABLESPACE TEMP;
- Tanto o espaço de tabela permanente padrão como o espaço de tabela temporário padrão são definidos no nível do BD, assim as duas últimas linhas do comando não são requeridas a menos que voce queira um espaço de tabela permanente padrão diferente ou um espaço de tabela temporário diferente para o usuário.

Criando usuários

- Mesmo que tenha sido atribuída ao usuário AUDITOR, explícita ou implicitamente, um espaço de tabela permanente padrão, ele não poderá criar nenhum objeto no BD até fornecermos uma cota e os direitos de criar objetos no seu próprio esquema.
- Uma quota é simplesmente um limite de espaço, por espaço de tabela, para um dado usuário.
- A menos que uma quota seja atribuída explicitamente ou seja concedido ao usuário o privilégio UNLIMITED TABLESPACE, o usuário não poderá criar objetos no seu próprio esquema.

Criando usuários

- No exemplo a seguir atribuiremos uma cota de 250MB no espaço de tabela USERS para a conta AUDITOR:
  - Alter user AUDITOR quota 250M on users;
- A menos que sejam concedidos alguns privilégios básicos a uma nova conta, essa conta nem ao menos poderá efetuar o login, portanto precisamos conceder pelo menos o privilégio CREATE SESSION ou o profile CONNECT.
- O profile CONNECT contém o privilégio CREATE SESSION, juntamente com outros privilégios básicos, como CREATE TABLE e ALTER SESSION.
  - Grant connect to AUDITOR

Alterando Usuários

- A alteração das características de um usuário é feita por meio do comando alter user.
- A sintaxe é quase idêntica a create user, exceto que alter user permite atribuir profiles e conceder direitos a uma aplicação na camada intermediária para que ela realize funções em favor do usuário.
  - Alter user AUDITOR
  - Default tablespace USER2
  - Quota 500M on user2;
- Agora o usuário AUDITOR ainda pode criar objetos no espaço de tabela USERS, mas ele deve explicitamente especificar USERS em quaisquer comandos create table e create index.

Eliminando usuários

- Descartar usuários é muito simples e se realiza com o comando drop user.
- Os únicos parâmetros são o nome do usuário a ser descartado e a opção cascade; quaisquer objetos possuídos pelo usuário devem ser explicitamente descartados ou movidos para um outro esquema se a opção cascade não for utilizada.
  - DROP USER NOMEUSUARIO CASCADE;

Métodos de Autorização pelo BD

- Depois que um usuário é autenticado no BD, o próximo passo é determinar os tipos de objetos, privilégios e recursos que o usuário tem permissão de acessar ou utilizar.
- Os perfis (profiles) podem controlar não apenas a maneira como as senhas são gerenciadas, mas também a maneira como os perfis podem impor limites sobre os vários tipos de recursos do sistema.

Gerenciamento de Perfil

- Parece que nunca há capacidade suficiente de CPU, espaço em disco ou largura de banda de E/S para executar a consulta de um usuário.
- Com a limitação desses recursos, o Oracle fornece um mecanismo para controlar quanto desses recursos um usuário pode utilizar.
- Um perfil Oracle é um conjunto identificado de limites de recursos que oferece esse mecanismo.

Gerenciamento de Perfil

- Os perfis também podem ser utilizados como um mecanismo de autorização para controlar a maneira como as senhas dos usuários são criadas, reutilizadas e validadas.
- Por exemplo, poderíamos querer impor um comprimento mínimo de senha, juntamente com um requisito de que pelo menos uma letra maiúscula e uma minúscula apareçam na senha.

O comando CREATE PROFILE

- O comando create profile realiza um trabalho duplo; podemos criar um perfil para limitar o tempo de conexão a um usuário para 120 minutos:
  - Create profile lim_connect limit connect_time 120;
- De maneira semelhante, podemos limitar o número de vezes consecutivas que um login pode falhar antes de a conta ser bloqueada:
  - Create profile lim_fail_login limit failed_login_attempts 8;
- Ou podemos combinar os dois tipos de limites em um único perfil:
  - Create profile lim_connectime_faillog limit connect_time 120 failed_login_attempts 8;

O comando CREATE PROFILE

- A maneira como o Oracle responde a um dos limites de recursos que é excedido depende do tipo de limite.
- Quando um dos limites, tempo de conexão ou tempo de inatividade, é atingido (como CPU_PER_SESSION), a transação em progresso é revertida e a sessão é desconectada.
- O Oracle fornece o perfil DEFAULT, que é aplicado a qualquer novo usuário se nenhum outro perfil for especificado. Essa consulta na visão de dicionários de dados DBA_PROFILES mostra os limites do perfil DEFAULT.

  - SELECT * FROM DBA_PROFILES WHERE PROFILE = 'DEFAULT‘
- A única restrição real no perfil DEFAULT limita o número de tentativas mal-sucedidas de logins consecutivos a dez antes de a conta ser bloqueada.
- Além disso, nenhuma função de verificação de senha é ativada.

Perfis e controle por senha

- Todas as unidades de tempo são especificadas em dias (para especificar qualquer um desses parâmetros em minutos, por exemplo, dividida em 1440):
  - Create profile lim_lock limit password_lock_time 5/1440;

- Neste exemplo, uma conta só será bloqueada por cinco minutos depois que o número especificado de logins falhar.
- Um valor de parâmetro unlimited signifca que não há limites sobre quanto de um dado recurso pode ser utilizado. Default significa que esse parâmetro recebe os seus valores do perfil DEFAULT.

Perfis e controle por senha

- Os parâmetros password_reuse_time e password_reuse_max devem ser utilizados juntos;
- Configurar um sem o outro não tem nenhum efeito útil.
  - Create profile lim_reuse_pass limit
  - Password_reuse_time 20
  - Password_reuse_max 5;
- Neste exemplo a seguir, criamos um perfil que configura password_reuse_time como 20 dias e password_reuse_max como 5.

Perfis e controle por senha

- Para usuários com esse perfil, suas senhas podem ser reutilizadas depois de 20 dias e se a senha tiver sido alterada pelo menos cinco vezes.
- Se você especificar um valor para qualquer um desses e UNLIMITED para o outro, um usuário nunca pode poderá reutilizar uma senha.

Perfis e controle por senha

Parâmetro de Senha	Descrição
FAILED_LOGIN_ATTEMPTS	O número de tentativas de login malsucedidas antes de a conta ser bloqueada
PASSWORD_LIFE_TIME	O número de dias que a senha pode ser utilizada antes de precisar ser alterada
PASSWORD_REUSE_TIME	O número de dias que um usuário deve esperar antes de reutilizar uma senha;
PASSSWORD_REUSE_MAX	O número de alterações de senha que tem de ocorrer antes de uma senha pode ser reutilizada;
PASSWORD_LOCK_TIME	Quantos dias a conta permanece bloqueada depois de tentativas FAILED_LOGIN_ATTEMPTS;
PASSWORD_GRACE_TIME	Número de dias depois do qual uma senha expirada deve ser alterada.
PASSWORD_VERIFY_FUNCTION	Um script PL/SQL para fornecer uma rotina avançada de verificação de senha.

Perfis e controle por senha

- O Oracle fornece um template para impor a diretiva de senha para uma empresa. Ele está localizado em $ORACLE_HOME/RDBMS/ADMIN/UTLPWDMG.SQL. O script fornece a seguinte funcionalidade para complexidade de senha:
  - Assegura que a senha não seja a mesma do nome do usuário;
  - Assegura que a senha tenha pelo menos quatro caracteres;
  - Verifica se a senha não é uma palavra simples e óbvia, como Oracle ou Database
  - Requer que a senha contenha uma letra, um dígito e um sinal de pontuação
  - Assegura que a senha seja diferente da senha anterior por pelo menos três caracteres.

Privilégios de objeto

- Exemplo:
  - GRANT INSERT, UPDATE, DELETE ON SCOTT.EMP TO AUDITOR;
  - GRANT SELECT ON SCOTT.EMP TO AUDITOR WITH GRANT OPTION;

PRIVILÉGIOS DE TABELA

- Os tipos de privilégios que podem ser concedidos em uma tabela dividem-se em três duas amplas categorias: operações DML e operações DDL.
- As operações DML incluem delete, insert, select e update, enquanto operações DDL incluem adicionar, descartar e modificar colunas colunas nas tabelas, com como criar índices na tabela.

PRIVILÉGIOS DE TABELA

- Ao conceder operações DML em uma tabela, é possível restringir essas operações somente a certas colunas.
  - REVOKE UPDATE ON SCOTT.EMP FROM AUDITOR;
  - GRANT UPDATE (EMPNO, ENAME, JOB, MGR,HIREDATE, COMM, DEPTNO, SEXO) ON SCOTT.EMP TO AUDITOR;

Criando, atribuindo e mantendo Role

- Um Role é um grupo identificado de privilégios, privilégios de sistema ou privilégios de objeto ou uma combinação dos dois, que facilita a administração de privilégios.
- Em vez de conceder privilégios de sistema ou de objeto individualmente a cada usuário, voce pode conceder o grupo de privilégios de sistema ou de objeto a um rolee, por sua vez, o papel pode ser concedido ao usuário.
- Isso reduz tremendamente o trabalho administrativo envolvido na manutenção de privilégios para usuários.

Criando um descartando um Role

- Para criar um role, utilize o comando create role e você deverá ter o privilégio de sistema CREATE ROLE. Em geral, esse role é concedido apenas a administradores do BD ou a administradores da aplicação.
  - Create role admin not indentified;

Role Padrão

- Por padrão, todos os papéis concedidos a um usuário são ativados quando o usuário se conecta.
- Se um Role só for utilizado dentro do contexto da aplicação, inicialmente esse role poderá estar desativado quando o usuário estiver conectado;
- Ele pode ser ativado e desativado dentro da aplicação.
- Se o usuário AUDITOR tiver os papéis CONNECT, RESOURCE, SCOTT_CLERK E DEPT30 e se quisermos especificar que SCOTT_CLERCK E DEPT30 não seja ativados por padrão, podemos fazer o seguinte:
  - ALTER USER AUDITOR DEFAULT ROLE ALL EXCEPT SCOTT_CLERK, DEPT30;

Role Padrão

- Quando AUDITOR conecta-se ao BD, ele tem automaticamente todos os privilégios concedidos com todos os papéis, exceto SCOTT_CLERCK E DEPT30;
  - AUDITOR poderia ativar explicitamente um papel na sua sessão utilizando set role:
    - SET ROLE DEPT030;
  - Quando ele terminar de acessar as tabelas para o departamento 30, ele pode desativar o papel na sua sessão:
    - SET ROLE ALL EXCEPT DEPT30;

Visões de Dicionário de dados sobre papéis

- A visão DBA_ROLE_PRIVS é uma boa maneira de descobrir quais papéis são concedidos a um usuário e também se pode passar esse papel para um outro usuário (ADMIN_OPTION) e se esse papel é ativado por padrão (DEFAULT_ROLE)
  - SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE = 'AUDITOR';

Visões de Dicionário de dados sobre papéis

- De maneira semelhante, podemos descobrir quais papéis atribuímos a algum role:
  - select * from dba_role_privs where grantee = 'ALL_DEPTS';
- Recuperar os privilégios concedidos diretamente é simples e direto:
  - SELECT GRANTEE, OWNER, TABLE_NAME, GRANTOR, PRIVILEGE, GRANTABLE
  - FROM DBA_TAB_PRIVS
  - WHERE GRANTEE = 'AUDITOR';

Visões de Dicionário de dados sobre papéis

- Para recuperar privilégios de tabela concedidos via roles, precisamos criar uma junção entre DBA_ROLE_PRIVS e ROLE_TAB_PRIVS:
  - SELECT DRP.GRANTEE, RTP.OWNER, RTP.TABLE_NAME, RTP.PRIVILEGE, RTP.GRANTABLE, RTP.ROLE
  - FROM ROLE_TAB_PRIVS RTP
  - JOIN DBA_ROLE_PRIVS DRP ON RTP.ROLE = DRP.GRANTED_ROLE
  - WHERE DRP.GRANTEE = 'SCOTT'

Auditoria

Auditoria

- O Oracle utiliza diferentes métodos de auditoria para monitorar quais tipos de privilégios são utilizados e quais objetos são acessados.
- A auditoria não impede o uso desses privilégios, mas pode fornecer informações úteis para revelar o abuso ou mal emprego dos privilégios

Auditoria

- A monitoração ou auditoria deve ser parte integrante dos procedimentos de segurança.
- As ferramentas internas de auditoria do Oracle incluem:
  - **Auditoria de Banco de Dados;**
    - Captura várias informações sobre um evento auditado.
  - **Auditoria baseada em valor**
    - Alterações de dados (DML);
  - **FGA (Fine-Grainte Auditing)**
    - Auditoria de Instruções de SQL, estende a auditoria de BD, capturando a instrução SQL real estendida.

Auditoria

Valor do Parâmetro	Ação
NONE, FALSE	Desativa a auditoria
OS	Ativa a auditoria. Envia os registros da auditoria a um arquivo do SO
DB, TRUE	Ativa a auditoria. Envia os registros da auditoria à tabela SYS.AUD$
DB, EXTEND	Ativa a auditoria. Envia os registros da auditoria para a tabela SYS.AUD$ e registra as informações adicionais nas colunas SQLBIND CLOB e SQLTEXT

Auditoria de Instrução

- Todos os tipos de auditoria utilizam o comando audit para ativar a auditoria e noaudit para desativá-la.
- Para auditoria de instrução, o formato do comando audit, tem a seguinte sintaxe:

- AUDIT cláusula_instrução_sql BY {SESSION | ACCESS} WHENEVER [NOT] SUCCESSFUL;

Auditoria de Instrução

- **Cláusula_instrução_sql**
  - Contém informações, como o tipo de instrução SQL que queremos auditar e quem estamos auditando.
- **By {Session|Access}:**
  - Auditar a ação toda vez que ela acontece (by access) ou somente uma vez (by session).
- **Whenever [not] successful:**
  - Auditar ações bem-sucedidas (sem mensagem de erro) ou [Not] quando os comandos que utilizam as instruções auditadas falham, devido as violações de privilégios, espaço insuficiente, espaço de tabela ou erros de sintaxe.

Auditoria de BD Padrão

- Ativada por meio do parâmetro AUDIT_TRAIL
- Pode fazer auditoria do(s):
  - Eventos de Login;
  - Exercício dos privilégios de Sistema
  - Exercício dos privilégios de objeto;
  - Uso de instruções SQL;
  - Exemplo:
    - AUDIT TABLE;
    - AUDIT DELETE ON SCOTT.EMP WHENEVER SUCCESSFUL;

Exemplos

- Também poderíamos querer auditar rotineiramente tanto os logins bem-sucedidos como os malsucedidos. Isto requer dois comandos de auditoria:
  - **Audit session whenever successful;**
  - **Audit session whenever not successful;**
  - SELECT USERNAME, TO_CHAR(TIMESTAMP,'DD/MM/YY HH24:MI') TIMESTAMP, OBJ_NAME, RETURNCODE, ACTION_NAME, SQL_TEXT FROM DBA_AUDIT_TRAIL WHERE ACTION_NAME IN ('LOGON', 'LOFOFF') ORDER BY TIMESTAMP

Exemplo

- Se quisermos auditar todos os comandos insert e update na tabela DEPT, independente de quem está fazendo a atualização e toda vez que a ação ocorrer, podemos utilizar o comando de auditoria:
  - AUDIT INSERT, UPDATE ON SCOTT.DEPT BY ACCESS WHENEVER SUCCESSFUL;
  - SELECT USERNAME, TO_CHAR(TIMESTAMP,'DD/MM/YY HH24:MI') TIMESTAMP, OBJ_NAME, RETURNCODE, ACTION_NAME, SQL_TEXT FROM DBA_AUDIT_TRAIL WHERE ACTION_NAME IN (‘INSERT', ‘UPDATE') ORDER BY TIMESTAMP

Protegendo a trilha de Auditoria

- A própria trilha de auditoria precisa ser protegida, especialmente se usuários que não são do sistema precisarem acessar a tabela SYS.AUD$.
- O profile DELETE_ANY_CATALOG é uma das maneiras de usuário não-SYS podem ter acesso à trilha de auditoria (arquivar e truncar a trilha de auditoria)
- Para configurar a auditoria na própria trilha de auditoria, conecte-se como SYSDBA e execute o comando:
  - AUDIT ALL ON SYS.AUD$ BY ACCESS;

Exemplos

- Podemos auditar comandos de DDL e DML, assim como alguns eventos do sistema:

- DDL (CREATE, ALTER & DROP de objetos)
- DML (INSERT UPDATE, DELETE, SELECT, EXECUTE).
- SYSTEM EVENTS (LOGON, LOGOFF etc.)

Especificando Opções de Auditoria

- Auditoria de Instruções de SQL
  - AUDIT TABLE;
- Auditoria (não específica e específica) de privílégios de sistema:
  - AUDIT SELECT ANY TABLE, CREATE ANY TRIGGER,
  - AUDIT SELECT ANY TABLE BY DEPT BY SESSION

Especificando Opções de Auditoria

- Auditoria (não específica e específica) de privilégios de objeto
  - AUDIT ALL ON SCOTT.EMP;
  - AUDIT UPDATE, DELETE ON SCOTT.EMP BY ACCESS;
- Auditoria de Sessões:
  - AUDIT SESSION WHENEVER NOT SUCCESSFUL;

Exemplos

- SHOW PARAMETER AUDIT;
- ALTER SYSTEM SET audit_trail=db SCOPE=SPFILE;
- SHUTDOWN IMMEDIATE;
- STARTUP;
- SHOW PARAMETER AUDIT;

Exemplos... Cont.

- CONNECT sys AS SYSDBA
- CREATE USER AUDITORIA IDENTIFIED BY AUDITORIA DEFAULT TABLESPACE users
- TEMPORARY TABLESPACE temp
- QUOTA UNLIMITED ON users;
- GRANT connect TO AUDITORIA;
- GRANT create table, create procedure TO AUDITORIA;

Exemplos

- Agora iremos definir o que vai ser auditado para o usuário AUDITORIA:
  - AUDIT ALL BY AUDITORIA BY ACCESS;
  - AUDIT SELECT TABLE, UPDATE TABLE, INSERT TABLE, DELETE TABLE BY AUDITORIA BY ACCESS;
  - AUDIT EXECUTE PROCEDURE BY AUDITORIA BY ACCESS;

Exemplos

- CONN AUDITORIA/AUDITORIA

- CREATE TABLE TESTE (id NUMBER);
- INSERT INTO TESTE (id) VALUES (1);
- UPDATE TESTE SET id = id;
- SELECT * FROM TESTE;
- DELETE FROM TESTE;
- DROP TABLE TESTE;

- Para saber quais as Views de informações de Auditoria:

  - SELECT view_name FROM dba_views
  - WHERE view_name LIKE 'DBA%AUDIT%'
  - ORDER BY view_name;

Exemplos

- COLUMN username FORMAT A10
- COLUMN owner FORMAT A10
- COLUMN obj_name FORMAT A10
- COLUMN extended_timestamp FORMAT A35

  - SELECT username, extended_timestamp, owner, obj_name, action_name
  - FROM dba_audit_trail
  - WHERE owner = 'AUDITORIA‘
  - ORDER BY timestamp
  - /