

Relatório de Laboratório: Administração de Usuários Oracle

Gabriel Vasconcelos

23 de setembro de 2025

Resumo

Este relatório documenta os passos executados em um laboratório de administração de usuários no Oracle Database. O objetivo é estudar os comandos de gerenciamento de usuários, privilégios e *roles*, além de explorar as visões do dicionário de dados relevantes para monitorar a segurança e a estrutura do banco de dados.

Sumário

1	Objetivo do Laboratório	3
2	Conexão com SYSTEM e Análise Inicial	3
2.1	Visão v\$version	3
2.2	Visão dba_users	3
3	Criação e Configuração do Usuário USR_LAB01	3
3.1	Comando CREATE USER	3
3.2	Comando GRANT CONNECT, RESOURCE	4
3.3	Teste de Conexão	4
4	Gerenciamento de Senha e Sessão	4
4.1	Alteração de Senha	4
4.2	Verificação da Sessão Ativa	4
4.3	Tentativa de Nova Conexão	5
5	Propriedade e Criação de Objetos	5
5.1	Criação de Tabelas pelo SYSTEM	5
5.2	Verificação de Propriedade pelo USR_LAB01	5
6	Concessão de Privilégios de Objeto	5
6.1	Criação do USR_LAB02 e Concessão de Privilégios	6
6.2	Verificação de Privilégios Concedidos	6
6.3	Operações do USR_LAB02	6
7	Gerenciamento de Privilégios com Roles	6
7.1	Acesso Restrito ao Dicionário de Dados	7
7.2	Criação e Concessão da Role new_dba	7
7.3	Verificação do Acesso	7

8	Análise Final de Privilégios	7
9	Apêndice: Comandos Utilitários Oracle XE	8
10	Conclusão	8

1 Objetivo do Laboratório

O objetivo principal deste laboratório é aprofundar o conhecimento sobre a criação e administração de usuários no Oracle. Isso inclui a concessão de privilégios de sistema e de objeto, o uso de *roles* para simplificar a gestão de permissões e a consulta a visões do dicionário de dados para auditar e verificar as configurações de segurança aplicadas.

2 Conexão com SYSTEM e Análise Inicial

A primeira etapa consiste em conectar-se com o usuário **SYSTEM**, que possui privilégios administrativos, e verificar informações básicas sobre o ambiente.

2.1 Visão v\$version

```
1 SELECT * FROM v$version;
```

Finalidade: A visão dinâmica **v\$version** fornece informações sobre a versão do software Oracle Database e seus componentes instalados. Ela é útil para identificar a edição do banco de dados (ex: Enterprise, Standard, Express), a versão principal e de patch, e a compatibilidade do hardware. O resultado mostra, por exemplo:

- Oracle Database 19c Enterprise Edition Release 19.0.0.0.0
- PL/SQL Release 19.0.0.0.0
- TNS for Linux: Version 19.0.0.0.0

2.2 Visão dba_users

```
1 SELECT username FROM dba_users;
```

Finalidade: A visão **dba_users** é parte do dicionário de dados e contém informações sobre todos os usuários definidos no banco de dados. Um administrador (DBA) pode usar esta visão para listar os usuários existentes, verificar seus status (ex: **OPEN**, **LOCKED**), *tablespaces* padrão, data de criação, etc. A consulta retorna uma lista com todos os nomes de usuário cadastrados na instância.

3 Criação e Configuração do Usuário USR_LAB01

3.1 Comando CREATE USER

```
1 CREATE USER USR_LAB01
2 IDENTIFIED BY SENHA
3 DEFAULT TABLESPACE users
4 QUOTA UNLIMITED ON users;
```

Explicação do Comando:

- **CREATE USER USR_LAB01:** Comando que inicia a criação de um novo usuário chamado **USR_LAB01**.

- IDENTIFIED BY SENHA: Define a senha inicial do usuário como SENHA.
- DEFAULT TABLESPACE users: Especifica que todos os objetos criados por este usuário (como tabelas e índices) serão armazenados na *tablespace* chamada **users**, a menos que outra seja explicitamente indicada.
- QUOTA UNLIMITED ON users: Concede ao usuário uma cota ilimitada de espaço de armazenamento na *tablespace* **users**. Sem uma cota, o usuário não poderia criar objetos nela.

3.2 Comando GRANT CONNECT, RESOURCE

```
1 GRANT CONNECT, RESOURCE to USR_LAB01;
```

Finalidade das Roles CONNECT e RESOURCE:

- CONNECT: É uma *role* que agrupa privilégios básicos de sistema. O mais fundamental é o CREATE SESSION, que permite ao usuário conectar-se ao banco de dados. Sem este privilégio, a tentativa de login falharia com o erro "ORA-01045: user lacks CREATE SESSION privilege".
- RESOURCE: Esta *role* concede privilégios para criar objetos no próprio *schema* do usuário, como CREATE TABLE, CREATE PROCEDURE, CREATE TRIGGER, entre outros.

Observação: A Oracle atualmente recomenda a criação de *roles* customizadas com os privilégios mínimos necessários, em vez de usar CONNECT e RESOURCE, que podem conceder mais permissões do que o necessário.

3.3 Teste de Conexão

Após a criação e a concessão da *role* CONNECT, abriu-se uma nova janela de terminal/SQL Developer e foi possível conectar com o usuário USR_LAB01 e a senha SENHA. (figura pendente).

4 Gerenciamento de Senha e Sessão

4.1 Alteração de Senha

O comando abaixo foi executado na janela do usuário SYSTEM.

```
1 ALTER USER USR_LAB01 IDENTIFIED BY new_password;
```

Este comando altera a senha do usuário USR_LAB01 para new_password.

4.2 Verificação da Sessão Ativa

Na janela onde o usuário USR_LAB01 já estava conectado, o seguinte comando foi executado:

```
1 SELECT table_name FROM all_tables;
```

Análise: O comando funcionou normalmente. A alteração de senha por um DBA **não invalida as sessões já existentes** daquele usuário. A nova senha só será exigida para novas tentativas de conexão.

4.3 Tentativa de Nova Conexão

Após encerrar a sessão de USR_LAB01, tentou-se conectar novamente.

- **Usando a senha antiga (SENHA):** A conexão falhou com o erro "ORA-01017: invalid username/password".
- **Usando a nova senha (new_password):** A conexão foi bem-sucedida.

O que aconteceu? O comando ALTER USER efetivou a troca da senha no dicionário de dados. Qualquer nova autenticação deve, obrigatoriamente, usar a credencial atualizada.

5 Propriedade e Criação de Objetos

Esta seção explora como a propriedade de objetos (*schema*) funciona no Oracle.

5.1 Criação de Tabelas pelo SYSTEM

Os comandos foram executados na janela do usuário SYSTEM.

```
1 SHOW USER; -- Retorna: USER is "SYSTEM"
2 CREATE TABLE xyz (name VARCHAR2(30));
```

Em qual usuário a tabela foi criada? A tabela xyz foi criada no *schema* do usuário que executou o comando, ou seja, no *schema* SYSTEM.

```
1 CREATE TABLE USR_LAB01.xyz (name VARCHAR2(30));
```

Em qual usuário a tabela foi criada? A tabela xyz foi criada no *schema* do usuário USR_LAB01.

Que nível de privilégio foi necessário? Para criar um objeto no *schema* de outro usuário, é necessário o privilégio de sistema CREATE ANY TABLE. O usuário SYSTEM possui este privilégio por padrão.

5.2 Verificação de Propriedade pelo USR_LAB01

Na janela do USR_LAB01, os comandos abaixo foram executados.

```
1 DESC xyz
```

O comando funcionou, exibindo a estrutura da tabela xyz, confirmando que ela pertence ao *schema* USR_LAB01.

```
1 DESC system.xyz
```

Esse comando funcionou? Não. O comando resultou no erro "ORA-04043: object system.xyz does not exist" ou "ORA-01031: insufficient privileges".

O que falta ao usuário USR_LAB01? Falta o privilégio de acessar objetos do *schema* SYSTEM. Para que o comando DESC (e outros como SELECT) funcione, o usuário USR_LAB01 precisaria receber o privilégio SELECT sobre a tabela system.xyz ou um privilégio mais amplo como SELECT ANY TABLE.

6 Concessão de Privilégios de Objeto

Nesta etapa, criamos um segundo usuário e concedemos a ele acesso a um objeto específico do USR_LAB01.

6.1 Criação do USR_LAB02 e Concessão de Privilégios

Na janela do SYSTEM:

```
1 CREATE USER USR_LAB02 IDENTIFIED BY SENHA DEFAULT TABLESPACE users;
2
3 GRANT INSERT, DELETE, SELECT ON USR_LAB01.XYZ TO USR_LAB02;
4
5 GRANT CONNECT TO USR_lab02;
```

Que operação está acontecendo? O comando GRANT INSERT, DELETE, SELECT ON USR_LAB01.XYZ TO USR_LAB02; está concedendo privilégios de objeto. Especificamente, o usuário USR_LAB02 está recebendo permissão para executar as operações de INSERT, DELETE e SELECT na tabela XYZ que pertence ao *schema* USR_LAB01.

6.2 Verificação de Privilégios Concedidos

```
1 select * from dba_tab_privs where grantee = 'USR_LAB02';
```

Qual o significado do resultado? Esta consulta à visão dba_tab_privs (privilégios de tabelas) mostra todos os privilégios de objeto concedidos a USR_LAB02. O resultado exibirá três linhas, indicando que o GRANTEE (beneficiário) USR_LAB02 recebeu os privilégios (PRIVILEGE) INSERT, DELETE e SELECT na tabela (TABLE_NAME) XYZ do proprietário (OWNER) USR_LAB01.

6.3 Operações do USR_LAB02

Conectado como USR_LAB02:

```
1 -- Insercao de dados na tabela de outro schema
2 insert into usr_lab01.xyz values ('teste de nome');
3 commit;
4
5 -- Consulta aos dados
6 select * from usr_lab01.xyz;
```

Resultado e Explicação: O comando SELECT funcionou e retornou a linha inserida: "teste de nome". Ele funcionou porque o usuário USR_LAB02 recebeu explicitamente o privilégio SELECT sobre a tabela usr_lab01.xyz. O INSERT também funcionou pelo mesmo motivo.

```
1 select * from system.xyz;
2 select * from xyz;
```

Explicação do porquê NÃO funcionaram:

- `select * from system.xyz;`: Falhou porque USR_LAB02 não tem permissão de acesso à tabela xyz do *schema* SYSTEM.
- `select * from xyz;`: Falhou porque, ao omitir o *schema*, o Oracle procura por uma tabela chamada xyz no *schema* do próprio usuário (USR_LAB02). Como essa tabela não existe, um erro é retornado.

7 Gerenciamento de Privilégios com Roles

Esta seção demonstra como usar *roles* para gerenciar o acesso a visões do dicionário de dados.

7.1 Acesso Restrito ao Dicionário de Dados

Na janela do USR_LAB01:

```
1 select * from dba_sys_privs;
```

O comando funcionou? Não. O acesso a visões DBA_ é restrito a usuários com altos privilégios, como SYSTEM ou aqueles com o privilégio SELECT ANY DICTIONARY ou a *role* SELECT_CATALOG_ROLE.

7.2 Criação e Concessão da Role new_dba

Na janela do SYSTEM:

```
1 CREATE ROLE new_dba;
2 GRANT CONNECT TO new_dba;
3 GRANT SELECT ANY TABLE TO new_dba;
4 GRANT SELECT_CATALOG_ROLE TO new_dba;
5
6 GRANT new_dba TO USR_LAB01;
```

7.3 Verificação do Acesso

Na janela do USR_LAB01, após reconectar para que os novos privilégios da *role* sejam carregados na sessão:

```
1 select * from dba_sys_privs;
```

Explicação do processo:

1. Foi criada uma *role* customizada chamada **new_dba**.
2. A esta *role* foram concedidos privilégios de sistema importantes: **CONNECT**, **SELECT ANY TABLE** (permite ler dados de qualquer tabela em qualquer *schema*, exceto SYS) e **SELECT_CATALOG_ROLE** (uma *role* pré-definida que concede privilégio de leitura sobre as visões do dicionário de dados).
3. A *role* **new_dba**, contendo todos esses privilégios, foi então concedida ao usuário USR_LAB01.
4. Ao herdar os privilégios da *role*, USR_LAB01 passou a ter permissão para consultar a visão **dba_sys_privs** e muitas outras, tornando o comando bem-sucedido.

8 Análise Final de Privilégios

As seguintes visões permitem auditar como os privilégios e *roles* foram distribuídos.

- **select * from dba_sys_privs**: Mostra os privilégios de sistema concedidos diretamente a usuários ou *roles*.
- **SELECT * FROM DBA_ROLE_PRIVS**: Lista quais *roles* foram concedidas a quais usuários (ex: mostra que USR_LAB01 possui a *role* NEW_DBA).
- **SELECT * FROM ROLE_ROLE_PRIVS**: Mostra *roles* que foram concedidas a outras *roles*.

- `SELECT * FROM ROLE_SYS_PRIVS`: Detalha quais privilégios de sistema uma *role* possui (ex: mostra que `NEW_DBA` tem o privilégio `SELECT ANY TABLE`).
- `SELECT * FROM ROLE_TAB_PRIVS`: Detalha quais privilégios de objeto (em tabelas, views, etc.) uma *role* possui.

9 Apêndice: Comandos Utilitários Oracle XE

Os comandos abaixo são úteis para administrar o ambiente Oracle Database Express Edition (XE) em um sistema Windows.

- `set path=C:\oraclexe\app\oracle\product\11.2.0\server\bin;%path%:`
Adiciona o diretório de executáveis do Oracle ao `PATH` do sistema, permitindo que comandos como `sqlplus` e `lsnrctl` sejam executados de qualquer local no terminal.
- `lsnrctl status`: Utilitário de controle do Listener. O comando `status` verifica o estado do Oracle Listener, que é o processo responsável por "escutar" por requisições de conexão de clientes e direcioná-las para a instância do banco de dados.
- `netstat -a`: Comando do sistema operacional que lista todas as portas de rede ativas e conexões. É útil para verificar se a porta padrão do Oracle Listener (geralmente 1521) está aberta e escutando.
- `sqlplus /nolog`: Inicia o cliente de linha de comando SQL*Plus sem tentar se conectar a um banco de dados, útil para se conectar a diferentes instâncias posteriormente.
- `conn user/senha`: Dentro do SQL*Plus, este comando é usado para conectar-se (ou trocar de conexão) a um banco de dados com o usuário e senha especificados.

10 Conclusão

Este laboratório proporcionou uma experiência prática e fundamental sobre a arquitetura de segurança do Oracle Database. Através da execução de comandos `CREATE USER`, `GRANT`, `ALTER USER` e `CREATE ROLE`, foi possível compreender a separação entre usuários (*schemas*), a distinção entre privilégios de sistema e de objeto, e o poder das *roles* para uma administração de permissões eficiente e escalável. A exploração das visões do dicionário de dados demonstrou como auditar e validar a correta implementação das políticas de segurança.