VOLTAR

Criando VIEWS - Visões

Criar e administrar VIEWS em banco de dados.

NESTE TÓPICO

Marcar tópico

Visões (VIEWS)

Uma VIEW funciona de forma semelhante a uma tabela. É utilizada em comandos SELECT, INSERT, UPDATE e DELETE para recuperação e manipulação de dados (com restrições), porém, não armazena estes dados.

A VIEW tem suas linhas e colunas calculadas dinamicamente através de um SELECT pré-estabelecido, cada vez que o solicitamos apenas a sua definição é armazenada no dicionário de dados.

Podemos dizer que se trata de uma tabela virtual, pois não possui linhas próprias, mas as obtém em tempo de execução e disponibiliza-as em memória para acesso por uma instrução.

Uma VIEW é como se fosse uma janela que dá acesso aos dados da tabela, mas possui restrições. Vejamos a sintaxe do comando:

```
1.  CREATE [OR REPLACE] [FORCE I NOFORCE] VIEW nome_view
2.  [(alias1 [, alias2] [, ....] ) ]
3.  AS subquery
4.  [WITH CHECK OPTION [CONSTRAINT constraint ] ]
5.  [WITH READ ONLY]
```

Onde:

Cláusula	Descrição

OR REPLACE	Se existente, a VIEW é recriada.
FORCE	Cria a VIEW mesmo que a tabela referenciada não exista.
NOFORCE	Cria a VIEW apenas se existir a tabela. É *default*.
Nome VIEW	É o nome VIEW.
aliasn	Especifica os nomes que atuarão como pseudocolunas da VIEW, a partir de expressões e/ ou colunas oriundas da tabela referenciada.
subquery	É o comando SELECT que originará a VIEW.
WITH CHECK OPTION	Especifica que operações de INSERT e UPDATE sobre a VIEW têm que resultar em linhas de possível acesso a VIEW.
constraint	É o nome da restrição CHECK OPTION
WITH READ ONLY	Garante que nenhuma operação DML pode ser feita sobre a VIEW.

Exemplo:

```
1.  CREATE VIEW exemplo AS SELECT c.nomecargo, f.nomefuncionario
2.  from cargo c INNER JOIN join funcionario f
3.  ON c.codcargo = f.codcargo;
```

Agora, podemos utilizar EXEMPLO como se fosse uma tabela.

A diferença é que ela obtém os dados dinamicamente, por meio da instrução especificada na própria definição da VIEW.

Recuperando Dados Através da View:

```
1.  SELECT * from exemplo;
```

Tipos de Views

Existem, basicamente, dois tipos de VIEWS: simples e complexas.

O tipo simples é composto por apenas um SELECT, utiliza apenas uma tabela, suas colunas são formadas por colunas da tabela original, sem cálculos ou funções.

A VIEW complexa é aquela em que há um JOIN entre tabelas na *subquery*, conforme visto no exemplo acima.

Com uma VIEW simples será possível executarmos comandos INSERT, UPDATE e DELETE (além do SELECT).

A manipulação dos dados através de uma VIEW não desabilita as *constraints* das tabelas as quais os mesmos se referem.

Cada coluna definida para VIEWS deve ter um nome de coluna válida.

Caso seja uma fórmula, deve possuir um alias (apelido).

A Cláusula WITH CHECK OPTION

Podemos garantir que os dados inseridos ou atualizados numa tabela através de uma VIEW poderão ser visualizados através da própria VIEW com esta opção.

Criando uma View:

```
1.  CREATE OR REPLACE VIEW alunos_v_uf
2.  AS SELECT * from alunos WHERE estado = 'RJ'
3.  WITH CHECK OPTION CONSTRAINT ALUNOS_V_UK_CK;
```

Vejamos a operação de atualização na view a seguir e o resultado:

Atualizando a View alunos_v_uf

```
1.  UPDATE alunos_v_uf SET estado = 'BA';
2.  ERROR at line 2:
3.  ORA-01402: view WITH CHECK OPTION where-clause violation
```

O erro é causado porque estamos tentando, através da VIEW que somente visualiza o estado 'RJ', alterá-la para 'BA', o que certamente impediria o acesso à tabela pela VIEW, pois esta tentaria construir a instrução restringindo pelo 'RJ'.

A Cláusula WITH READ ONLY

Também podemos impedir operações DML sobre a VIEW, criando a opção WITH READ ONLY, o que a restringe apenas a operações de leitura.

Criando uma VIEW:

```
1.  CREATE OR REPLACE VIEW ALUNOS_V_UF
2.  AS SELECT * from alunos WHERE estado = 'RJ'
3.  WITH READ ONLY;
```

A seguir, vejamos a operação de deleção na VIEW e o resultado:

Deletando dados da View ALUNOS_V_UF

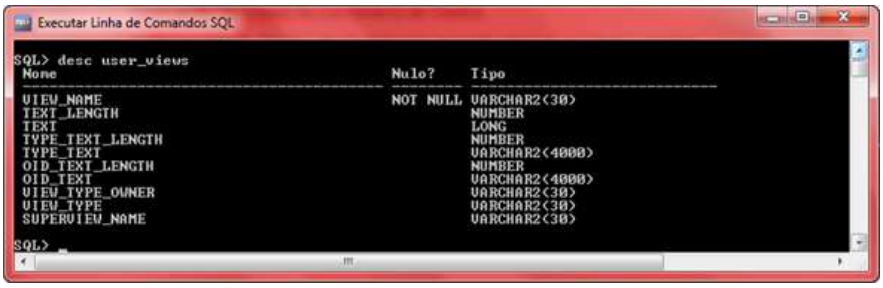
```
1.  DELETE from alunos_v_uf;
2.  ERROR at line 1:
3.  ORA-01752: cannot delete from view without exactly one key preserved table
```

A VIEW está protegida para operações DML, portanto causará erro em qualquer tentativa de inserção, atualização ou deleção.

Checando as VIEWS no Dicionário de Dados: o Dicionário de Dados do Oracle nos permite checar as VIEWS criadas.

Checando o Dicionário de Dados, exemplo:

```
1.  desc user_views
```



Selecionando

Exemplo:

```
1.  SELECTview_name, text_length, text from user_views
2.  WHERE view_name = 'EXEMPLO';
```

O texto da VIEW é exibido somente até 80 caracteres e o mecanismo de criação da view define o nome de todas as colunas da tabela origem, se colocarmos '*'.

Para eliminar uma VIEW o comando DROP VIEW apaga uma VIEW do Dicionário de Dados. Nenhum efeito ocorrerá sobre as Tabelas referenciadas, bastando para isso ter apenas o privilégio DROP VIEW ou DBA.

Vejamos a sintaxe do comando:

```
1.  DROP VIEW nome_view;
```

Referências

BEIGHLEY, Lynn. *Use a Cabeça SQL*. Rio de Janeiro: Alta Books, 2008.

FANDERUFF, Damaris. *Dominando o Oracle 9i: Modelagem e Desenvolvimento,* São Paulo: Makron, 2003.

GRAVES, Mark. *Projeto de banco de dados com XML*. São Paulo: Pearson, 2003.

MORELLI, Eduardo Terra. *Oracle 9i Fundamental: SQL, PL/SQL e Administração,* São Paulo, Editora Érica, 2002.

PRICE, Jason. *Oracle Database 11g SQL*. (tradução: João Eduardo Nóbrega Tortello). Porto Alegre: Bookman, 2009.

SILVA, Robson. *Oracle Database 10g Express Edition*. São Paulo: Editora Érica, 2007.

Avalie este tópico

ANTERIOR
Operações com conjuntos

Índice

Biblioteca (https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)
Portal Uninove (http://www.uninove.br)
Mapa do Site

Ajuda? (https://ava.uninove.br/seu/AVA/topico/topico.php?idCurso=)