
ORACLE
Academy

Programação de Banco de Dados com SQL

15-1
Criando Views

ORACLE Academy

Objetivos

Esta lição abrange os seguintes objetivos:

- Listar três usos para views do ponto de vista do administrador do banco de dados
- Explicar, da perspectiva dos negócios, por que é importante ser capaz de criar e usar subconjuntos lógicos de dados derivados de uma ou mais tabelas
- Criar uma view com e sem aliases de coluna na subconsulta usando uma única tabela básica

Objetivos

Esta lição abrange os seguintes objetivos:

- Criar uma view complexa que contém funções de grupo para exibir valores de duas tabelas
- Recuperar dados de uma view

Finalidade

- Pare um pouco e pense no que você aprendeu até agora como aluno do Oracle Academy.
- Seria fácil explicar o que você sabe para alguém que não fez este curso?
- Você devia se dar parabéns!
- O nível de conhecimento que você adquiriu é compreendido apenas por algumas pessoas.

Finalidade

- Agora, imagine-se como o Administrador do Banco de Dados de uma empresa.
- O que você faz quando um gerente pede para tornar possível que ele recupere e insira dados usando o banco de dados da empresa?
- "Não faça nada muito complicado. Só quero conseguir preparar relatórios sobre todas as nossas operações".

Finalidade

- Os funcionários devem ter acesso a todos os dados da empresa?
- Como vão executar comandos que exijam condições de junção?
- É uma boa ideia permitir que qualquer pessoa insira dados?
- Essas são perguntas que você, como DBA, precisa saber responder.
- Nesta seção, você aprenderá a criar "views": representações virtuais de tabelas personalizadas para atender às necessidades específicas do usuário.

View

- Assim como uma tabela, uma view é um objeto de banco de dados.
- No entanto, views não são tabelas "reais".
- Elas são representações lógicas de tabelas existentes ou de outra view.
- Views não contêm dados próprios.
- Elas funcionam como uma janela através da qual se pode ver ou alterar os dados das tabelas.

View

- As tabelas nas quais uma view se baseia são chamadas de tabelas "básicas".
- A view é uma consulta armazenada como uma instrução SELECT no dicionário de dados.

```
CREATE VIEW view_employees
AS SELECT employee_id,first_name,
last_name, email
FROM employees
WHERE employee_id BETWEEN 100 and 124;
```

```
SELECT *
FROM view_employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
100	Steven	King	SKING
101	Neena	Kochhar	NKOCHHAR
102	Lex	De Haan	LDEHAAN
124	Kevin	Mourgos	KMOURGOS
103	Alexander	Hunold	AHUNOLD
104	Bruce	Ernst	BERNST
107	Diana	Lorentz	DLORENTZ

Por Que Usar Views?

- As views restringem o acesso aos dados da tabela básica, pois podem exibir colunas selecionadas.
- As views podem ser usadas para reduzir a complexidade da execução de consultas com base em instruções SELECT mais complicadas.
- Por exemplo, o criador da view pode construir instruções de junção que recuperem dados de várias tabelas.
- O usuário da view não vê o código subjacente nem o modo como ele é criado.
- Através da view, o usuário interage com o banco de dados usando consultas simples.

Por Que Usar Views?

- As views podem ser usadas para recuperar dados de várias tabelas, oferecendo independência de dados para os usuários.
- Os usuários podem exibir os mesmos dados de maneiras diferentes.
- As views fornecem a grupos de usuários acesso aos dados de acordo com suas permissões ou seus critérios particulares.

Criando uma View

- Para criar uma view, incorpore uma subconsulta na instrução CREATE VIEW.
- A sintaxe de uma instrução de view é a seguinte:

```
CREATE [OR REPLACE] [FORCE| NOFORCE] VIEW view  [(alias [,
 alias]...)] AS subconsulta
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

Criando uma View

OR REPLACE	**Recria a view, caso ela já exista.**
FORCE	**Cria a view, mesmo que as tabelas básicas existam ou não.**
NOFORCE	**Cria a view apenas se a tabela básica existir (padrão).**
view_nome	**Especifica o nome da view.**
alias	**Especifica um nome para cada expressão selecionada pela consulta da view.**
subconsulta	**É uma instrução SELECT completa. Você pode usar aliases para as colunas na lista SELECT. A subconsulta pode conter sintaxe SELECT complexa.**

Criando uma View

WITH CHECK OPTION	**Especifica que as linhas permanecem acessível à view após operações de inserção ou atualização.**
CONSTRAINT	**É o nome atribuído à constraint CHECK OPTION.**
WITH READ ONLY	**Garante que nenhuma operação DML possa ser executada na view.**

Criando uma View

- Exemplo:

```
CREATE OR REPLACE VIEW view_euro_countries
AS SELECT country_id, region_id, country_name, capitol
 FROM wf_countries
 WHERE location LIKE '%Europe';
```

```
SELECT * FROM view_euro_countries
ORDER BY country_name;
```

COUNTRY_ID	REGION_ID	COUNTRY_NAME	CAPITOL
22	155	Bailiwick of Guernsey	Saint Peter Port
203	155	Bailiwick of Jersey	Saint Helier
387	39	Bosnia and Herzegovina	Sarajevo
420	151	Czech Republic	Prague
298	154	Faroe Islands	Torshavn
49	155	Federal Republic of Germany	Berlin
33	155	French Republic	Paris
…	…	…	…

Diretrizes da Criação de uma View

- A subconsulta que define a view pode conter sintaxe SELECT complexa.
- Por razões de desempenho, a subconsulta que define a view não deve conter uma cláusula ORDER BY. Essa cláusula é melhor especificada quando você recupera dados da view.
- Você pode usar a opção OR REPLACE para mudar a definição da view sem precisar eliminá-la ou conceder novamente os privilégios de objeto que já pertenceram a ela.
- Os aliases podem ser usados para os nomes de colunas na subconsulta.

Recursos de CREATE VIEW

- São usadas duas classificações de views: simples e complexa.
- A tabela resume os recursos de cada view.

Recurso	Views Simples	Views Complexas
Número de tabelas usadas para derivar dados	Uma	Uma ou mais
Pode conter funções	Não	Sim
Pode conter grupos de dados	Não	Sim
Pode executar operações DML (INSERT, UPDATE, DELETE) através de uma view	Sim	Nem sempre

View Simples

- A view mostrada abaixo é um exemplo de uma view simples.
- A subconsulta deriva dados de apenas uma tabela e não contém uma função de junção nem funções de grupo.
- Como se trata de uma view simples, as operações INSERT, UPDATE, DELETE e MERGE que afetam a tabela básica podem ser executadas através da view.

```
CREATE OR REPLACE VIEW view_euro_countries
AS SELECT country_id, country_name, capitol
 FROM wf_countries
 WHERE location LIKE '%Europe';
```

View Simples

- Os nomes das colunas na instrução SELECT podem ter aliases, como mostrado abaixo.
- Observe que os aliases também podem ser listados após a instrução CREATE VIEW e antes da subconsulta SELECT.

```
CREATE OR REPLACE VIEW view_euro_countries
AS SELECT country_id AS "ID", country_name AS "Country",
  capitol AS "Capitol City"
 FROM wf_countries
 WHERE location LIKE '%Europe';
```

```
CREATE OR REPLACE VIEW view_euro_countries("ID", "Country",
  "Capitol City")
AS SELECT country_id, country_name, capitol
 FROM wf_countries
 WHERE location LIKE '%Europe';
```

View Simples

- É possível criar uma view, mesmo que as tabelas básicas existam ou não.
- O acréscimo da palavra FORCE à instrução CREATE VIEW cria a view.
- Para um DBA, essa opção poderia ser útil durante o desenvolvimento de um banco de dados, principalmente se ele estiver esperando que os privilégios necessários para o objeto referenciado sejam concedidos em breve.
- A opção FORCE criará a view, mesmo que seja inválida.
- A opção NOFORCE é o padrão quando se cria uma view.

View Complexa

- As views complexas são aquelas que podem conter funções de grupo e junções.
- O exemplo a seguir cria uma view que deriva dados de duas tabelas.

```
CREATE OR REPLACE VIEW view_euro_countries
 ("ID", "Country", "Capitol City", "Region")
AS SELECT c.country_id, c.country_name, c.capitol, r.region_name
 FROM wf_countries c JOIN wf_world_regions r
 USING (region_id)
 WHERE location LIKE '%Europe';
```

```
SELECT *
FROM view_euro_countries;
```

View Complexa

ID	Country	Capitol City	Region
375	Republic of Belarus	Minsk	Eastern Europe
48	Republic of Poland	Warsaw	Eastern Europe
421	Slovak Republic	Bratislava	Eastern Europe
36	Republic of Hungary	Budapest	Eastern Europe
90	Republic of Turkey	Ankara	Eastern Europe
40	Romania	Bucharest	Eastern Europe
373	Republic of Moldova	Chisinau	Eastern Europe
370	Republic of Lithuania	Vilnius	Eastern Europe
371	Republic of Latvia	Riga	Eastern Europe
372	Republic of Estonia	Tallinn	Eastern Europe
…	…	…	…

View Complexa

- Funções de grupo também podem ser adicionadas a instruções de view complexa.

```
CREATE OR REPLACE VIEW view_high_pop
 ("Region ID", "Highest population")
AS SELECT region_id, MAX(population)
 FROM wf_countries
 GROUP BY region_id;
```

```
SELECT * FROM view_high_pop;
```

Region ID	Highest population
5	188078227
9	20264082
11	131859731
13	107449525
14	74777981
15	78887007
17	62660551
18	44187637
21	298444215
…	…

Modificando uma View

- Para modificar uma view existente sem precisar eliminá-la e depois recriá-la, use a opção OR REPLACE na instrução CREATE VIEW.
- A view antiga é substituída pela nova versão.
- Por exemplo:

```
CREATE OR REPLACE VIEW view_euro_countries
AS SELECT country_id, region_id, country_name, capitol
 FROM wf_countries
 WHERE location LIKE '%Europe';
```

Terminologia

Estes são os principais termos usados nesta lição:

- Alias
- View complexa
- CREATE VIEW
- FORCE
- NOFORCE
- REPLACE

Terminologia

Estes são os principais termos usados nesta lição:

- View simples
- Subconsulta
- View
- VIEW_NOME

Resumo

Nesta lição, você deverá ter aprendido a:

- Listar três usos para views do ponto de vista do administrador do banco de dados
- Explicar, da perspectiva dos negócios, por que é importante ser capaz de criar e usar subconjuntos lógicos de dados derivados de uma ou mais tabelas
- Criar uma view com e sem aliases de coluna na subconsulta usando uma única tabela básica

Resumo

Nesta lição, você deverá ter aprendido a:

- Criar uma view complexa que contém funções de grupo para exibir valores de duas tabelas
- Recuperar dados de uma view


ORACLE
Academy