

Banco de Dados - Ciclo 3 - Webaula 24



WEBAULA 24

≡ Criação e Manipulação de Visões

≡ Conclusão do ciclo

≡ Referências

QUESTION BANKS

Criação e Manipulação de Visões

Nesta seção você vai estudar visões em SGBDs relacionais – conhecidas como views. No contexto dos SGBDs relacionais, uma view é uma relação que não armazena dados; é composta dinamicamente por uma consulta que é previamente analisada e otimizada. Isso significa que, diferentemente de uma tabela, uma view não é um objeto físico, ou seja, não ocupa espaço em disco. Dessa forma, quando se alteram os dados de uma tabela que são acessados por views, os resultados gerados pelas consultas armazenadas nessas views também são atualizados.

A view é uma tabela lógica baseada em uma tabela ou até mesmo em outra visão. Você pode apresentar combinações ou subconjuntos lógicos de dados criando views de tabelas. Uma view não contém dados próprios, mas é como uma janela através da qual os dados das tabelas podem ser vistos. As tabelas nas quais uma view é baseada são chamadas tabelas base. Observe a Figura 79, onde temos uma view sendo prospectada a partir da tabela EMP.

Tabela EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7782	CLARK	MANAGER	7839	09-JUN-81	1500	300	10
7934	MILLER	View VW_EMP10		23-JAN-82	1300		10
7566	JONES			02-APR-81	2975		20
7788	SCOTT						20
7876	ADAMS						20
7369	SMITH						20
7902	FORD						20
7698	BLAKE						30
7654	MARTIN						30
7499	ALLEN						30
7844	TURNER						30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30

Figura 79

Fonte: Adaptada de Oracle (2017)

A *view* é armazenada como uma instrução SELECT no dicionário de dados do SGBD relacional. No exemplo acima temos a criação de uma view tendo como base a tabela EMP, já conhecida por você, não é mesmo? Pois bem, observe que a view criada foi denominada “VW_EMP10” e é composta pelas colunas EMPNO, ENAME e JOB. A seguir apresentamos algumas das principais finalidades da utilização de *views*:

- para restringir o acesso a dados;
- para facilitar as consultas complexas;
- para permitir a independência dos dados;
- para apresentar diferentes views dos mesmos dados.

Detalhando um pouco mais estas finalidades, que ao mesmo tempo podem se confundir com vantagens, temos que as *views* restringem o acesso a dados porque uma *view* pode exibir, de forma seletiva, colunas a partir de uma tabela; elas permitem que o usuário faça consultas simples para recuperar resultados de consultas complexas, uma vez que a consulta complexa estaria na *view*. Por exemplo, uma *view* permite que usuários consultem informações de várias tabelas sem saber como criar uma instrução de junção entre essas tabelas. As *views* permitem independência de dados para usuários e programas aplicativos, pois sua origem estaria abstrata. Uma *view* pode ser usada para recuperar os dados de várias tabelas e de outras *views*, se necessário, apresentando diferentes formas de visualização dos mesmos dados.

Dica

Até algum tempo atrás, alguns SGBDs relacionais apresentavam restrições quanto à criação e utilização de views em seus repositórios de dados. Atualmente as visões são compatíveis com praticamente todos os principais SGBDs relacionais no mercado.

A seguir, no Quadro 102, vamos conhecer a sintaxe para a criação de uma *view*.

Quadro 102 – Sintaxe para criação de *view*

```
CREATE VIEW Identificador  
AS subconsulta
```

A partir da sintaxe apresentada, temos que:

- o nome da *view* (identificador) não pode duplicar o nome de outro objeto do SGBD relacional – por exemplo, ter o mesmo nome de uma tabela;
- a subconsulta pode conter uma sintaxe de uma sentença *SELECT* simples ou complexa, conforme a necessidade;
- a subconsulta não pode conter uma cláusula *ORDER BY*.

Você se lembra do cenário (Figura 79) utilizado para conceituar uma *view*? Pois bem, o código apresentado no Quadro 103 corresponde à criação de uma *view*, que contempla as informações (código, nome e atividade) dos empregados que trabalham no departamento 10:

Quadro 103 – Exemplo de criação de *view*

```
CREATE VIEW      vw_emp10  
AS  
  SELECT        empno, ename, job
```

```
FROM      emp
WHERE     deptno = 10;
```

i Dica

Aproximamo-nos do final do conteúdo sobre SQL previsto em nossa disciplina. Da mesma forma que nas demais seções de conteúdo, recomendamos que você vá executando as sentenças SQL para colocar em prática os conceitos apresentados.

Se você verificar a estrutura da *view* criada, vai constatar que a visão apresenta as colunas e os tipos de dados correspondentes à tabela base (de origem das colunas). A seguir, na Figura 80, vamos conhecer o comportamento do SGBD relacional quando uma *view* é acessada.

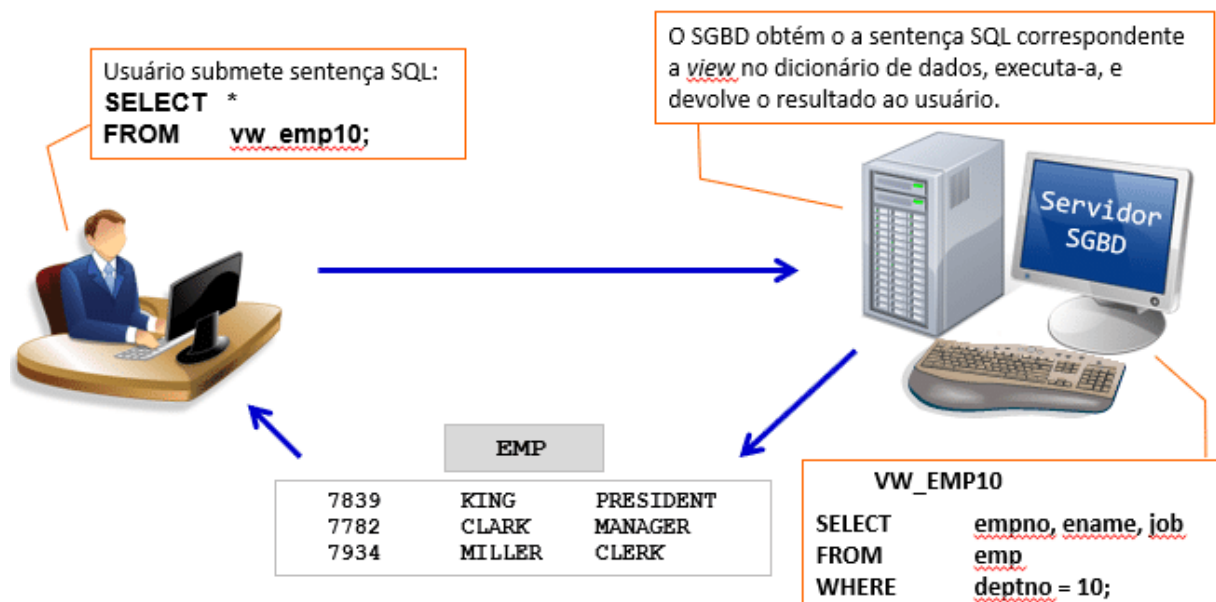


Figura 80

Como se pode observar na ilustração acima, quando você acessa os dados usando uma *view* as seguintes operações são executadas: (i) é recuperada a definição da *view* a partir do dicionário de dados; (ii) são verificados os privilégios de acesso para a tabela base da *view*; (iii) a sentença de recuperação é submetida à consulta da *view* em uma operação nos objetos subjacentes (de origem) ou de base; (iv) por fim o resultado é devolvido ao usuário. Desta forma você pode perceber que qualquer alteração sobre os dados nas tabelas base refletem o resultado da consulta realizada pela *view*.

A seguir temos o Quadro 104, que apresenta um exemplo de criação de uma *view* onde são atribuídos apelidos às colunas, os quais, na *view*, passam a ser os nomes das colunas.

Quadro 104 – Exemplo de criação de *view*

```
CREATE VIEW      vw_sal30
AS
  SELECT      empno CODIGO
              ename  NOME
              sal    SALARIO
FROM          emp
WHERE         deptno = 30;
```

No exemplo acima foi criada uma visão que recupera o código, o nome e o salário de todos os empregados que atuam no departamento 30. Sempre que houver a necessidade de acessar os dados provenientes desta visão, utilizamos o comando apresentado no Quadro 105 para que o SGBD relaciona os recupere.

Quadro 105 – Recuperando dados de uma *view*

```
SELECT  *

FROM    vw_sal30;
```

A execução da consulta realizada na view “VW_SAL30” apresenta o seguinte resultado (Quadro 106):

Quadro 106 – Recuperando dados de uma *view*

CODIGO	NOME	SALARIO
-----	-----	-----
7698	BLAKE	2850
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
7521	WARD	1250

Há situações em que desejamos alterar a consulta armazenada em uma visão. Para modificar uma *view* você deve utilizar a sentença CREATE OR REPLACE VIEW. Vamos exemplificar alterando a *view* VW_EMP10 através da adição de apelidos às colunas conforme o Quadro 107.

Quadro 107 – Alterando propriedades de uma *view*

```
CREATE OR REPLACE VIEW vw_emp10
    (codigo_empregado, nome_empregado, atividade)
AS
    SELECT      empno, ename, job
    FROM        emp
    WHERE       deptno = 10;
```

A opção “OR REPLACE” permite que uma *view* seja criada mesmo que uma já exista com esse nome, substituindo assim a versão antiga da visão para o seu proprietário. Isso significa que a *view* poderá ser alterada sem eliminar, recriar ou reconceder os privilégios de objeto. No exemplo, ao atribuir apelidos de coluna na cláusula CREATE VIEW, estes apelidos são listados na mesma ordem que as colunas na subconsulta.

A seguir (Quadro 108) é apresentado um exemplo de uma *view* considerada complexa, pois, além de utilizar dados provenientes de mais de uma tabela, são utilizadas funções de agrupamento de dados. Tais fatores caracterizam, por exemplo, não mais uma visão simples, e sim complexa.

Quadro 108 – Criando uma *view* complexa

```
CREATE VIEW      vw_dept_salarios
      (nome, menor, maior, media)
AS
  SELECT    d.name, MIN(e.sal), MAX(e.sal), AVG(e.sal)
  FROM      emp.e, dept d
  WHERE     e.deptno = d.deptno
  GROUP BY  d.dname
```

Para verificar o resultado produzido pela view criada no exemplo acima, podemos executar a sentença SQL do Quadro 109.

Quadro 109 – Acessando dados de uma *view* complexa

```
SELECT  *
FROM    vw_dept_salarios;
```


Observamos no exemplo acima que acessar uma *view* complexa se dá da mesma forma que outra visão qualquer. Como resultado serão exibidos dados como: nome do departamento, menor, maior e média dos salários de todos os departamentos que apresentarem dados armazenados.

Por fim, precisamos saber como fazemos para remover uma visão. Antes, é importante destacar que, quando se remove uma *view*, os dados, que estão baseados em tabelas subjacentes (de origem) no SGBD relacional, não sofrem qualquer modificação. Portanto apenas a descrição da visão é eliminada do dicionário do SGBD, assim como as referências e privilégios que porventura tenham sido concedidos a ela. A seguir (Quadro 110) temos a sintaxe do comando e em seguida um exemplo demonstrando este recurso.

Quadro 110 – Acessando dados de uma *view* complexa

Sintaxe:

```
DROP VIEW view;
```

Exemplo:

```
DROP VIEW vw_emp10;
```

Encerramento

Nesta seção aprendemos a utilizar visões ou views no SGBD relacional, que pode ser criada a partir de uma ou mais tabelas e/ou visões. A utilização de visões pode simplificar o acesso aos dados, assim como permitir uma certa independência das estruturas de armazenamento. Além disso, também podemos criar inúmeras visões sobre os mesmos dados.

CONTINUE

Conclusão do ciclo

Este último Ciclo de nossa disciplina foi dedicado à linguagem SQL, que é utilizada para interagir com os principais SGBDs relacionais através do uso de instruções/comandos.

Neste Ciclo você teve a oportunidade de conhecer comandos para estruturas e manipular os dados em SGBDs relacionais. Também explorou as funcionalidades do principal comando da SQL: o SELECT.

Apresentamos o comando SELECT iniciando por recursos básicos de recuperação de dados, caminhando para recursos envolvendo várias tabelas. Trabalhamos o agrupamento de dados, recurso que eleva significativamente o nível de abstração das operações em um SGBD relacional.

Em seguida trabalhamos com subconsultas, das mais simples às mais elaboradas, e terminados o Ciclo falando de visões (view). Acreditamos que após o acompanhamento dos conteúdos, exemplos e atividades propostas, você tenha condições de utilizar os principais recursos da linguagem SQL. Boa sorte, sucesso em suas escolhas e até uma próxima!

Referências

Referências

ORACLE. Documentação de utilização do sistema (Versão 11g). Califórnia: Oracle Corp., 2017.

CARDOSO, Vírínia M. **Linguagem SQL**: fundamentos e práticas. São Paulo: Saraiva, 2009.

Referências de Imagens

Divisão de Modalidades de ensino (DME), Fundação Universidade Regional de Blumenau (FURB), 2019.