

ACH2025

Laboratório de Bases de Dados

Aula 11

Visões

Professora:

➤ **Fátima L. S. Nunes**



Introdução

- ✓ Nos exemplos mostrados nas aulas anteriores, consideramos o modelo lógico → relações usadas são aquelas reais, armazenadas no BD.
- ✓ Todos os usuários podem ver o modelo lógico inteiro?

Introdução

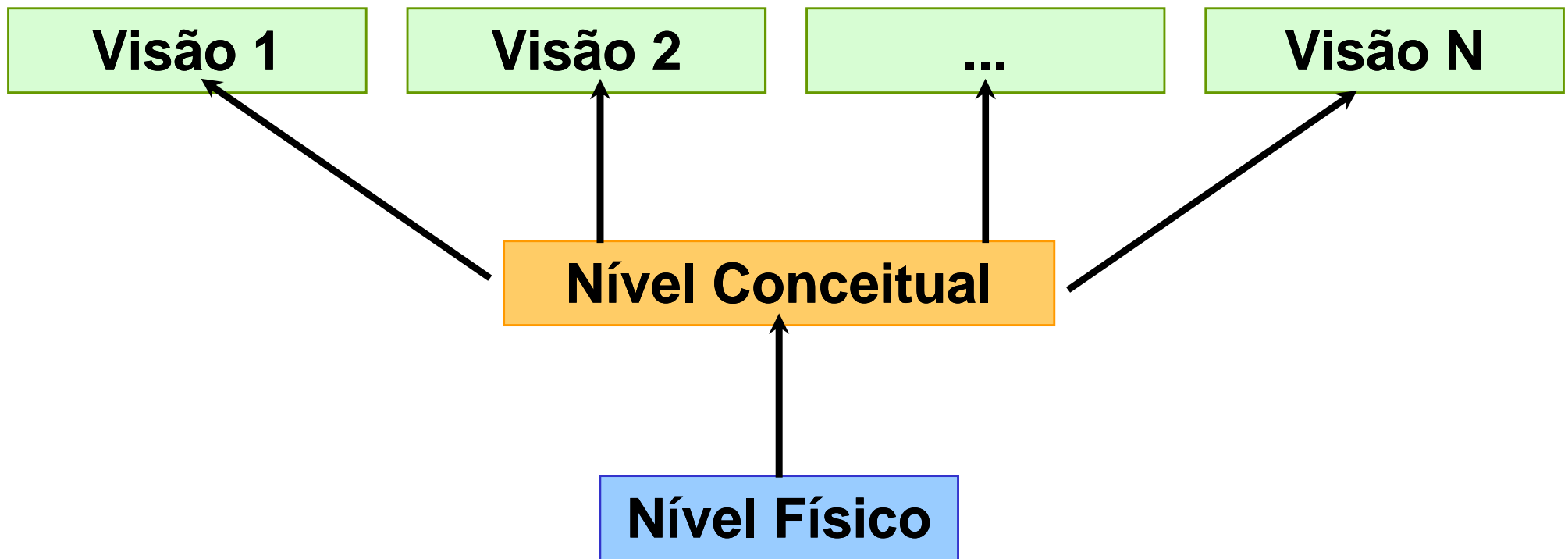
- ✓ Nos exemplos mostrados nas aulas anteriores, consideramos o modelo lógico → relações usadas são aquelas reais, armazenadas no BD.
- ✓ Todos os usuários podem ver o modelo lógico inteiro?
NÃO !!!!!!!!!
- ✓ Por exemplo: não posso permitir que qualquer funcionário saiba o salário de todos os funcionários da empresa!

Introdução

- ✓ Nos exemplos mostrados nas aulas anteriores, consideramos o modelo lógico → relações usadas são aquelas reais, armazenadas no BD.
- ✓ Todos os usuários podem ver o modelo lógico inteiro?
NÃO !!!!!!!!!
- ✓ Por exemplo: não posso permitir que qualquer funcionário saiba o salário de todos os funcionários da empresa!
- ✓ Qualquer relação que não seja parte do modelo lógico, mas que é visível a um usuário como uma **relação virtual**, é chamada de **view**.

Visões

- ✓ **Segurança do BD** → é desejável filtrar as informações do BD → cada categoria de usuário deve “enxergar” somente a parte que lhe é permitida.
- ✓ **Visões** → permitem esta diferenciação.



Definição de visões em SQL

✓ Comando *create view*.

✓ Sintaxe:

```
create view v as <expressões de consulta>
```

✓ *v* é o nome da visão

✓ <expressões de consulta> é uma expressão de consulta válida em SQL.

Visões

✓ Exemplos:

- ▶ *Conta* (numero_conta, nome_cliente, saldo, codigo_agencia)
- ▶ *Agencia* (codigo_agencia, nome_agencia, cidade)
- ▶ *Emprestimo* (numero_emprestimo, nome_cliente, valor, codigo_agencia)

Criar uma visão consistindo de agências e seus clientes.

```
create view Clientes_agencia as
    (select nome_agencia, nome_cliente
     from conta, agencia
     where conta.codigo_agencia = agencia.codigo_agencia)
UNION
    (select nome_agencia, nome_cliente
     from emprestimo, agencia
     where emprestimo.codigo_agencia =
       agencia.codigo_agencia)
```



Visões

✓ Visão definida → pode usar o seu nome para fazer referência à relação virtual criada por essa visão.

✓ Exemplos:

Usando a visão ***Clientes_agencia*** criada, selecionar todos os clientes da agência Centro.

► ***Clientes_agencia (nome_agencia, nome_cliente)***

```
select nome_cliente  
from clientes_agencia  
where nome_agencia= 'CENTRO';
```



Visões

- ✓ Nomes de atributos de uma *view* podem ser especificados de forma explícita:
- ✓ **Exemplos:**
 - ▶ *Agencia* (codigo_agencia, nome_agencia, cidade)
 - ▶ *Emprestimo* (numero_emprestimo nome_cliente, valor, codigo_agencia)

```
create view emprestimo_total_agencia
(nome_agencia, emprestimo_total)
as
select nome_agencia, sum(valor)
from emprestimo, agencia
where emprestimo.codigo_agencia =
      agencia.codigo_agencia)
group by nome_agencia;
```



Visões X Tabelas

- ✓ A definição de visão é diferente da operação de criação de uma nova tabela a partir de outra.
- ✓ Exemplo:
 - A partir da tabela EMP, crie uma **tabela** denominada **empdep10** que contenha os números, nomes e cargos de todos funcionários do departamento 10.

```
SQL> desc emp;
```

Nome	Nulo?	Tipo
-----	-----	-----
EMPNO	NOT NULL	NUMBER(4)
JOB		VARCHAR2(20)
ENAME		VARCHAR2(40)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)
MGR		NUMBER(4)

Visões X Tabelas

- ✓ A definição de visão é diferente da operação de criação de uma nova tabela a partir de outra.
- ✓ Exemplo:

```
create table empdep10
as
select empno,ename,job
from emp
where deptno=10;
```

```
desc empdep10
```

Name	Null?	Type
-----	-----	----
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)

Visões X Tabelas

- ✓ A definição de visão é diferente da operação de criação de uma nova tabela a partir de outra.
- ✓ Exemplo:

```
create table empdep10
as
select empno,ename,job
from emp
where deptno=10;
```

```
select * from empdep10;
```

EMPNO	ENAME	JOB
7839	KING	PRESIDENT
7782	CLARK	MANAGER
7934	MILLER	CLERK

Visões X Tabelas

- ✓ A definição de visão é diferente da operação de criação de uma nova tabela a partir de outra.
- ✓ Exemplo:
 - A partir da tabela EMP, crie uma **visão** denominada **vempdep10** que contenha os números, nomes e cargos de todos funcionários do departamento 10.

```
SQL> desc emp;
```

Nome	Nulo?	Tipo
-----	-----	-----
EMPNO	NOT NULL	NUMBER(4)
JOB		VARCHAR2(20)
ENAME		VARCHAR2(40)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)
MGR		NUMBER(4)

Visões X Tabelas

- ✓ A definição de visão é diferente da operação de criação de uma nova tabela a partir de outra.
- ✓ Exemplo:

```
create view vempdep10
as
select empno,ename,job
from emp
where deptno=10;
```

```
desc vempdep10
```

Name	Null?	Type
-----	-----	-----
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)

Visões X Tabelas

- ✓ A definição de visão é diferente da operação de criação de uma nova tabela a partir de outra.
- ✓ Exemplo:

```
create view vempdep10
as
select empno,ename,job
from emp
where deptno=10;
```

```
select * from vempdep10;
```

EMPNO	ENAME	JOB
7839	KING	PRESIDENT
7782	CLARK	MANAGER
7934	MILLER	CLERK

Visões X Tabelas

- ✓ A definição de visão é diferente da operação de criação de uma nova tabela a partir de outra.
- ✓ Exemplo:

```
create table empdep10  
as  
select empno,ename,job  
from emp  
where deptno=10;
```

```
create view vempdept10  
as  
select empno,ename,job  
from emp  
where deptno=10;
```

Qual é a diferença???



Visões X Tabelas

- ✓ Quando uma visão é definida, o **SGBD armazena a sua definição** propriamente dita, em vez do resultado da expressão que a gerou.
- ✓ Sempre que uma consulta é desenvolvida, a relação visão é recalculada.
- ✓ Vantagens e desvantagens?

Visões X Tabelas

- ✓ Quando uma visão é definida, o **SGBD armazena a sua definição** propriamente dita, em vez do resultado da expressão que a gerou.
- ✓ Sempre que uma consulta é desenvolvida, a relação visão é recalculada.
- ✓ Vantagens e desvantagens?
 - *view* está sempre atualizada – mesmo que tabelas sejam modificadas;
 - *view* exige maior tempo de processamento.

Visões X Tabelas

- ✓ Alguns SGBDs permitem que relações de visões sejam armazenadas → **visões materializadas**:
 - consultas baseadas nessas visões têm tempo de resposta menor;
 - maior custo de armazenamento e atualizações.

Visões

- ✓ Nomes das *views* podem aparecer em qualquer lugar onde um nome de relação pode aparecer.
- ✓ Mas, como ficam as atualizações?

Visões

- ✓ Nomes das *views* podem aparecer em qualquer lugar onde um nome de relação pode aparecer.
- ✓ Mas, como ficam as atualizações?
 - Em princípio, nenhuma operação de atualização pode ser executada em uma *view*.

– Por quê?

Atualizações em Visões

- ✓ Devido à definição de visões → atualizações podem gerar problemas no esquema do BD.
- ✓ Atualização em uma visão → reflete atualizações nas relações envolvidas.
- ✓ Problema → atualizações indevidas ou incompletas.

Atualizações em Visões

✓ Exemplo:

```
create view Clientes_agencia as
    (select nome_agencia,nome_cliente
     from conta, agencia
     where conta.codigo_agencia = agencia.codigo_agencia)
UNION
    (select nome_agencia,nome_cliente
     from emprestimo, agencia
     where emprestimo.codigo_agencia =
        agencia.codigo_agencia)
```

✓ O usuário poderia escrever:

```
insert into Clientes_agencia values ('Acapulco', 'Manoel')
```



Atualizações em Visões

✓ Exemplo:

✓ O usuário poderia escrever:

```
insert into Clientes_agencia values ('Acapulco', 'Manoel')
```

✓ Relações envolvidas na definição da view:

- ▶ *Conta (numero-conta, nome_cliente, saldo, codigo_agencia)*
- ▶ *Agencia (codigo_agencia, nome_agencia, cidade)*

O que aconteceria ???



Atualizações em Visões

✓ Exemplo:

✓ O usuário poderia escrever:

```
insert into Clientes_agencia values ('Acapulco', 'Manoel')
```

✓ Relações envolvidas na definição da *view*:

- ▶ *Conta* (numero-conta, nome_cliente, saldo, codigo_agencia)
- ▶ *Agencia* (codigo_agencia, nome_agencia, cidade)

✓ Isto significa criar tuplas em conta e agência com alguns atributos nulos!

Conta: (nulo, “Manoel”, nulo, nulo)

Agencia: (nulo, “Acapulco”, nulo)

Problemático!!



EACH

Atualizações em Visões

Por problemas como esses, não são permitidas atualizações em visões na maioria dos SGBDs !!



Visões a partir de visões

- ✓ É possível criar novas visões a partir de visões já existentes.

```
create view Clientes_agencia as
    (select nome_agencia,nome_cliente
    from conta, agencia
    where conta.codigo_agencia = agencia.codigo_agencia)
UNION
    (select nome_agencia,nome_cliente
    from emprestimo, agencia
    where emprestimo.codigo_agencia =
        agencia.codigo_agencia);

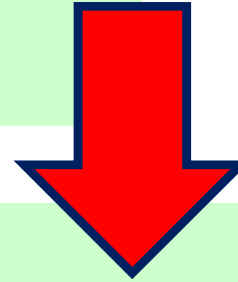
create view Clientes_centro as
    (select nome_cliente
    from Clientes_agencia
    where nome_agencia = 'CENTRO');
```

Visões a partir de visões

- ✓ Ao processo de criar uma *view* a partir de outras *views* dá-se o nome de ***expansão de views***.
- ✓ O processo considera que as definições de *views* não são recursivas:
 - ✓ nenhuma *view* é usada em sua própria definição – direta ou indiretamente.
- ✓ O que de fato acontece na expansão de *views*? O que o SGBD faz?

Visões a partir de visões

```
create view Clientes_centro as  
  (select nome_cliente  
   from Clientes_agencia  
   where nome_agencia = 'CENTRO');
```

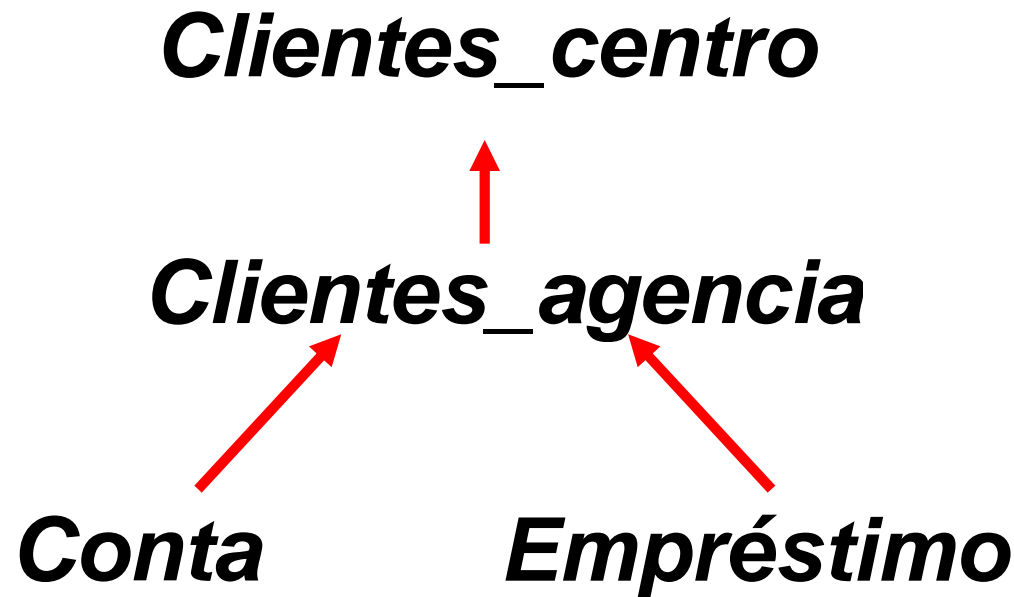


```
create view Clientes_centro as  
  (select nome_cliente from  
   ((select nome_agencia,nome_cliente  
    from conta, agencia  
    where conta.codigo_agencia = agencia.codigo_agencia)  
   UNION  
   (select nome_agencia,nome_cliente  
    from emprestimo, agencia  
    where emprestimo.codigo_agencia= agencia.codigo_agencia))  
   where nome_agencia = 'CENTRO');
```

Visões a partir de visões

- ✓ Uma visão v_1 é diretamente dependente de uma visão v_2 se v_2 é usada na expressão que define v_1 .
- ✓ Uma visão v_1 é dependente de uma visão v_2 se existir um caminho no gráfico de dependência de v_2 para v_1 .

Gráfico de dependência:



Exercícios

✓ Considere os seguintes esquemas de relação:

- ▶ *Professor* (número-prof, profnome, profrua, profcidade)
- ▶ *Aluno* (número-aluno, alunome, alurua, alucidade)
- ▶ *Matrícula* (número-aluno, código-disc, ano)
- ▶ *Disciplina* (código-disc, nome-disciplina, nome-curso, quantidade de aulas)
- ▶ *ProfDisc* (código-disc, número-prof, ano)

1. Crie uma visão com todas as disciplinas oferecidas neste ano, constando os nomes das disciplinas, nomes e cidades dos professores responsáveis e nomes dos cursos das disciplinas que os professores ministram.
2. A partir da visão criada no exercício 1, mostre todos os nomes dos professores do curso de Sistemas de Informação do ano corrente e os nomes das disciplinas pelas quais são responsáveis.
3. Crie uma visão com todos os nomes de alunos matriculados no ano corrente os nomes das disciplinas em que estão matriculados, a cidade onde moram e os nomes dos cursos das disciplinas.
4. A partir das visões criadas, mostre todos os alunos de Manaus que cursam disciplinas do Curso de Sistemas de Informação no ano corrente.
5. A partir das visões criadas, mostre os nomes de todos os professores que não moram em São Paulo e que ministram alguma disciplina no curso de Sistemas de Informação.
6. Crie uma visão com os dados dos professores (número e nome) e a quantidade total de aulas que ministram.



Exercícios

✓ Considere os seguintes esquemas de relação:

- ▶ *Professor* (número-prof, profnome, profrua, profcidade)
- ▶ *Aluno* (número-aluno, alunome, alurua, alucidade)
- ▶ *Matrícula* (número-aluno, código-disc, ano)
- ▶ *Disciplina* (código-disc, nome-disciplina, nome-curso, quantidade de aulas)
- ▶ *ProfDisc* (código-disc, número-prof, ano)

7. Crie uma visão com o nome dos professores o total de aulas que ministram em cada curso. A visão deve conter o número e nome do professor, o nome do curso e o total de aulas no curso.
8. A partir da visão criada, mostre o total de aulas de cada professor no curso de Sistemas de Informação.
9. Crie uma visão com o nome do aluno, nome do curso, código e nome das disciplinas matriculadas no ano corrente.
10. A partir da visão criada, mostre e a quantidade total de disciplinas de cada aluno do curso de Sistemas de Informação no ano corrente.
11. A partir da visão criada, mostre o total de alunos matriculados no curso de Sistemas de Informação no ano corrente.
12. O que acontece com as visões criadas se matricularmos a aluna Michele Silva na disciplina 986? Quais visões seriam afetadas?
13. O que acontece com as visões criadas se atribuirmos a disciplina “Estruturas de Dados” para o professor “Carlos”? Quais visões seriam afetadas?
14. Faça um gráfico de dependência das visões criadas nos exercícios anteriores.



ACH2025

Laboratório de Bases de Dados

Aula 11

Visões

Professora:

➤ **Fátima L. S. Nunes**

