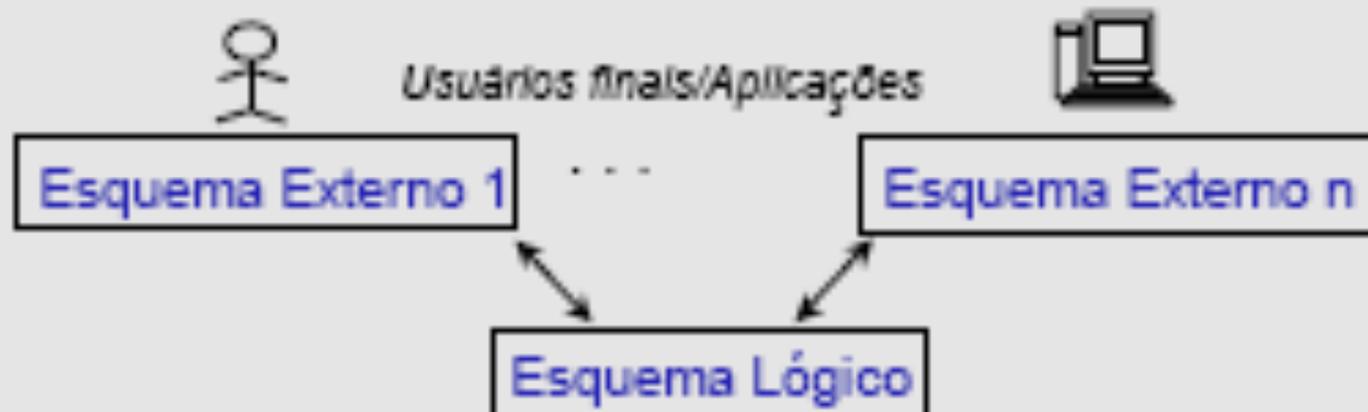


Visões

Prof. Alexandre Tagliari Lazzaretti
lazzaretti10@gmail.com

Visões

- Visão
 - **tabela derivada** a partir das tabelas do BD
 - **tabela virtual** (por *default*)
 - isto é transparente para usuários e aplicações
 - visões são manipuladas como tabelas normais do BD
- Visões fazem parte dos esquemas externos da arquitetura de um SGBD



Definição de Visões

- Exemplo
 - o setor de câncer do hospital lida apenas com dados de pacientes que têm esta doença
- Definição em SQL

```
create view PacCâncer (código, paciente, idade) as
select codp, nome, idade
from Pacientes → tabela base
where problema = 'câncer';
```

Operações DML sobre Visões

- São executadas na(s) tabela(s) base
- Exemplos

```
select *  
from PacCâncer  
where paciente like 'J%';
```



```
select codp, nome, idade  
from Pacientes  
where problema = 'câncer'  
and nomep like 'J%';
```

```
update PacCâncer  
set idade = idade + 1  
where código = 1;
```



```
update Pacientes  
set idade = idade + 1  
where problema = 'câncer'  
and codp = 1;
```

- Considerações
 - toda visão é passível de consulta
 - nem toda a visão é passível de atualização

Atualização de Visões

- Situação 1

```
create view DadosMed as
```

```
    select codm, nome, RG, especialidade  
    from Médicos;
```

```
create view Horários as
```

```
    select data, hora  
    from Consultas;
```

→ I / A / E nas visões **DadosMed** e **Horários**?

Atualização de Visões

- Situação 2

```
create view ConsultasMed (código, nroConsultas) as
select codm, count(*)
from Consultas
group by codm;
```

→ inclusão/alteração de dados em ConsultasMed?

- Conclusão 2

- uma visão atualizável deve conter atributos que tenham correspondência direta com atributos da tabela base
 - mapeamento 1-1 entre atributo da visão e atributo da tabela base

Atualização de Visões

- Situação 3
 - supor que a chave primária na tabela Consultas é (codm, codp, data, hora)

```
create view ConsultasMP (codigoM, especialidade,  
                        codigoP, data) as  
select Médicos.codm, especialidade, codp, data  
from Médicos join Consultas  
on Médicos.codm = Consultas.codm;
```

Atualização de Visões

- Situação 3

```
create view ConsultasMP (codigoM, especialidade,  
                        codigoP, data) as  
select Médicos.codm, especialidade, codp, data  
from Médicos join Consultas  
on Médicos.codm = Consultas.codm;
```

- Conclusão 3

- uma visão atualizável deve ser derivada de apenas uma tabela base
 - garantia de um mapeamento sempre 1-1 entre uma tupla da visão e uma tupla do BD

Atualização de Visões

- Atributo definido no predicado da visão **não** é um atributo da visão

```
create view PacCâncer (código, paciente, idade) as
select codp, nome, idade
from Pacientes
where problema = 'câncer';
```

- tuplas incluídas em **PacCâncer** não serão vistas por ela!
 - solução: *trigger* que seta problema = "câncer"

Atualização de Visões

- Atributo definido no predicado da visão é um atributo da visão

```
create view MédicosJovens as
  select código, nome, especialidade, idade
  from Médicos
  where idade < 35;
```

[with check option;]

- cláusula *with check option*
 - controle de valores válidos para os atributos da visão que fazem parte do seu predicado
 - Inclusão ou alteração em *MédicosJovens*
 - » apenas *idades < 35* serão permitidas!

Uso de Visões

- Vantagens
 - independência lógica dos dados
 - cada usuário ou aplicação "enxerga" a porção do BD que deseja
 - segurança de acesso
 - cada usuário ou aplicação "enxerga" a porção do BD que tem permissão
- Problemática de visões atualizáveis
 - limita o uso de visões
 - tema de pesquisa na área de BD

Sintaxe visões

```
CREATE [OR REPLACE] VIEW  
NOME_VIEW (COLUNAS)  
AS consulta
```

Exemplo: **Mostras as pessoas físicas cadastradas**

```
CREATE VIEW FISICAS (CODIGO, NOME, TELEFONE)  
AS SELECT COD_CLI, NOME_CLI, TEL_CLI  
FROM CLIENTES WHERE FIS_JUR='F'
```