

## Views

- uma view é uma “tabela virtual” .
- não existe fisicamente na BD.
- é definida à custa de outras tabelas ou views.
- sintaxe:

```
CREATE VIEW <nome> AS <query>;
```

## Exemplo

Criar uma view que dá o nome, ano e duração de todos os filmes da Disney.

```
CREATE VIEW filmes_disney (nome,ano,realizador) AS
  SELECT nome, ano, nomeRealizador
  FROM Filmes
  WHERE nomeEstudio = 'Disney';
```

Agora, podemos fazer queries à view `filmes_disney` tal e qual como se fosse uma tabela.

```
-- Quais os filmes da Disney feitos nos anos 90?
SELECT *
FROM filmes_disney
WHERE ano>=1990 AND ano<=2000;
```

## Outro exemplo

```
CREATE VIEW filmes_atores (filme, ano, estudio,
                           realizador, actor) AS
SELECT F.nome, F.ano, F.nomeEstudio,
       F.nomeRealizador, P.nomeActor
FROM Filmes AS F, Participa AS P
WHERE F.nome = P.nomeFilme
      AND F.ano = P.anoFilme;
```

Quais os estúdios que já trabalharam com a Nicole Kidman?

```
SELECT estudio
FROM filmes_atores
WHERE actor = 'Nicole Kidman';
```

## Indices

- um índice é uma estrutura de dados que é utilizada para acelerar as pesquisas.
- os índices aceleram as pesquisas mas tornam mais lentos os inserts, deletes, e updates.
  - porque o SGBD para além de modificar tabelas, também têm de modificar a estrutura de dados.
- muitos SGBDs criam índices automaticamente para as chaves primárias, é o caso do PostgreSQL.

## Indices (cont.)

- vamos supor que temos de procurar muitas vezes por filmes cuja duração seja maior que  $x$  minutos.
- se não tivermos um índice, o SGBD terá de fazer uma pesquisa sequencial por todos os tuplos da tabela.
- se tivermos um índice para o atributo duracao, o SGBD poderá utilizar um método de pesquisa mais sofisticado.

```
-- criar um índice para 'duracao'  
CREATE INDEX IndiceDuracao ON Filmes(duracao);  
  
-- apagar o índice  
DROP INDEX IndiceDuracao;
```

## Permissões/Autorização

- cada objecto da BD tem um dono (o utilizador que criou o objecto).
- por exemplo, se o utilizador 'figo' criar a tabela de filmes, apenas o 'figo' pode aceder e modificar os dados dessa tabela.
- o SQL oferece um mecanismo de permissões de modo a que o 'figo' possa dar permissões a outros utilizadores.

## **Permissões/Autorização (cont.)**

Em SQL usa-se o comando GRANT. Existe vários tipos de permissões:

- SELECT
- INSERT
- DELETE
- UPDATE

## Exemplos

Dar permissão ao utilizador a33333 para fazer SELECTs na tabela de filmes.

```
GRANT SELECT ON Filmes TO a33333;
```

Dar permissão ao utilizador a33333 para fazer SELECTs, e UPDATEs à tabela de filmes.

```
GRANT SELECT, UPDATE ON Filmes TO a33333;
```

Dar permissão ao utilizador a33333 para fazer SELECTs, e UPDATEs apenas à coluna aCores

```
GRANT SELECT, UPDATE(aCores) ON Filmes  
TO a33333;
```

## Exemplos (cont.)

Dar todas as permissões (SELECT, DELETE, e UPDATE) ao utilizador a33333 para aceder e/ou modificar a tabela de filmes.

```
GRANT ALL ON Filmes TO a44444;
```

Também se pode dar permissões a vários utilizadores.

```
GRANT SELECT ON Filmes  
TO a33333, a44444, a55555;
```

Também se pode dar permissões a todos os utilizadores.

```
GRANT SELECT ON Filmes TO PUBLIC;
```

## **Permissões/Autorização (cont.)**

O comando REVOKE retira as permissões.

- O comando REVOKE retira as permissões.
- Exemplo:

```
REVOKE SELECT ON Filmes TO a33333;
```

## Restrições

Já vimos vários tipos de restrições:

- PRIMARY KEY
- UNIQUE
- NOT NULL
- FOREIGN KEY

## Mais restrições

Podemos ainda,

- restringir valores ao nível do atributo.
- restringir valores ao nível do tuplo.
- restringir valores para a BD em geral.

## Restrições ao nível do atributo

- coloca-se CHECK( <condição> ) a seguir à definição do atributo.
- a condição pode referir o nome do respectivo atributo. Apenas pode referir outros atributos ou outras tabelas num subquery.
- Exemplo:

```
CREATE TABLE Actores(  
    ...  
    sexo CHAR CHECK (sexo IN ('F','M')),  
    nomeEstudio VARCHAR(30) CHECK (  
        nomeEstudio IN  
        (SELECT nome FROM Estudios)),  
    ...  
);
```

## Restrições ao nível do atributo (cont.)

Podemos ainda,

- o check só é verificado quando existe um insert ou update no atributo respectivo.
- Ex 1: `CHECK( sexo IN ('F','M'))` é verificado cada vez que há um insert ou update ao valor de `sexo`.
  - se o valor não for 'F' nem 'M', o SGBD rejeita a modificação.
- Ex 2: `CHECK( nomeEstudio IN (SELECT nome FROM Estudios))` não é verificado se apagarmos um estúdio da tabela de Estúdios.
  - é diferente de uma chave estrangeira.

## Restrições ao nível do tuplo

- podemos ter um CHECK( <condição> ) como um elemento à parte na definição de uma tabela.
- a condição pode referir-se a qualquer atributo da tabela. Apenas pode referir outros atributos ou outras tabelas num subquery.
- novamente, o check só é verificado em inserts e updates.
- se a condição do check for falsa, o insert ou update do respectivo tuplo é rejeitado.

## Exemplo

Os filmes anteriores a 1939 não podem ser a cores.

```
CREATE TABLE Filmes(  
    ...  
    CHECK (NOT (ano<1939 AND aCores=TRUE))  
);
```

## Restrições para a BD em geral (Asserções)

- uma asserção é uma restrição mais geral que pode especificar várias tabelas.
- uma vez definida, a asserção faz parte do esquema da BD.
- as asserções são verificadas cada vez que há uma modificação no estado da BD.
- não está implementado em PostgreSQL.

```
CREATE ASSERTION <nome>  
    CHECK ( <condição> );
```

## Exemplo

Não pode haver realizadores que também sejam actores (este exemplo é obviamente fictício)

```
CREATE ASSERTION NaoHaActoresRealizadores
  CHECK( NOT EXISTS
    (SELECT nome FROM Actores
     INTERSECT
     SELECT nome FROM Realizadores
    )
  );
```

## Outro exemplo

A duração média de todos os filmes de um determinado estúdio não pode exceder 200 minutos.

```
CREATE ASSERTION DuracaoMedia
  CHECK( 200 >= ALL
        (SELECT AVG(duracao)
         FROM Filmes
         GROUP BY nomeEstudio
        )
  );
```

- se fizéssemos apenas um check ao nível da tabela de filmes, a restrição poderia falhar quando houvesse deletes, visto que as restrições ao nível da tabela apenas são testadas em inserts e updates.