

Uma visão diferente

Podemos utilizar a visão em consultas:

```
SELECT nome, loja
FROM comprasCuritiba, Produto
WHERE comprasCuritiba.produto = Produto.nome AND
      Produto.categoria = "calçados"
```

5

Roteiro

- Definição de visões
- Atualização de visões
- Respondendo consultas através de visões

2

O que acontece quando consultamos uma visão?

```
SELECT nome, loja
FROM comprasCuritiba, Produto
WHERE comprasCuritiba.produto = Produto.nome AND
      Produto.categoria = "calçados"
```



```
SELECT nome, Compra.loja
FROM Pessoa, Compra, Produto
WHERE Pessoa.cidade = "Curitiba" AND
      Pessoa.nome = Compra.comprador AND
      Compra.produto = Produto.nome AND
      Produto.categoria = "calçados"
```

6

Definição de Visões

Visões são relações, com a diferença que elas não são fisicamente armazenadas.

Para apresentar informações distintas para usuários distintos.
[Funcionario](#)(RG, nome, departamento, projeto, salario)

```
CREATE VIEW Projetistas AS
SELECT nome, projeto
FROM Funcionario
WHERE departamento = "Desenvolvimento"
```

Os usuários do Departamento Pessoal tem acesso a [Funcionario](#); outros usuários só podem acessar [Projetistas](#).

3

Tipos de Visões

- Visões virtuais:
 - Usadas em banco de dados
 - Calculadas sob demanda – pode ser lento
 - Sempre atualizada
- Visões materializadas:
 - Usadas em data warehouses
 - Precomputada offline – rápido
 - Pode conter dados desatualizados

7

Uma visão diferente

[Pessoa](#)(nome, cidade)
[Compra](#)(comprador, vendedor, produto, loja)
[Produto](#)(nome, fabricante, categoria)

```
CREATE VIEW comprasCuritiba AS

SELECT comprador, vendedor, produto, loja
FROM Pessoa, Compra
WHERE Pessoa.cidade = "Curitiba" AND
      Pessoa.nome = Compra.comprador
```

Temos uma nova tabela virtual:
[comprasCuritiba](#)(comprador, vendedor, produto, loja)

4

Atualização de Visões

Como inserir uma tupla em uma tabela que não existe?

[Funcionario](#)(RG, nome, departamento, RGdoChefe, salario)

```
CREATE VIEW Projetistas AS
SELECT RG, nome
FROM Funcionario
WHERE departamento = "Desenvolvimento"
```

Se fizermos a inserção:

```
INSERT INTO Projetistas
VALUES("123456789", "Jose")
```

Ela corresponde a:

```
INSERT INTO Funcionario(RG, nome, departamento, RGdoChefe, salario)
VALUES("123456789", "Jose", NULL, NULL, NULL)
```

8

Visões que não podem ser atualizadas

Funcionario(RG, nome, departamento, RGdoChefe, salario)

```
CREATE VIEW Diretores AS
SELECT DISTINCT x.RG, x.nome
FROM Funcionario x, Funcionario y, Funcionario z
WHERE y.RGdoChefe = x.RG
      and z.RGdoChefe = y.RG
```

```
INSERT INTO Diretores
VALUES ("123456789", "Joe")
```

Impossível de executar !

9

Reutilização de Visões Materializadas

- Suponha que temos **somente** o resultado de comprasCuritiba:

```
SELECT comprador, vendedor, produto, loja
FROM Pessoa, Compra
WHERE Pessoa.cidade = "Curitiba" AND
      Pessoa.nome = Compra.comprador
```

- e queremos responder a consulta:

```
SELECT comprador, vendedor
FROM Pessoa, Compra
WHERE Pessoa.cidade = 'Curitiba' AND
      Pessoa.nome = Compra.comprador AND
      Compra.produto = 'camera'.
```

Então, podemos reescrever a consulta utilizando a visão.

13

Padrão SQL- 92

- SQL- 92: permite atualização de visões definidas apenas com uma relação utilizando seleção e projeção

- visões que contém a chave primária (idAlu)
- visões que não contém a chave primária

Aluno(idAluno, nome, curso, mediaGeral)

```
CREATE VIEW bomAlunos (id, media)
AS SELECT idAlu, mediaGeral
FROM Aluno
WHERE mediaGeral >= 70
```

10

Reformulação de Consultas utilizando Visões

Consulta reescrita:

```
SELECT comprador, vendedor
FROM ComprasCuritiba
WHERE produto = 'camera'
```

Consulta original:

```
SELECT comprador, vendedor
FROM Pessoa, Compra
WHERE Pessoa.cidade = 'Curitiba' AND
      Pessoa.nome = Compra.comprador AND
      Compra.produto = 'camera'.
```

14

SQL- 99

- SQL- 99: permite atualização de atributos em visões que envolvem mais de uma relação, mas o atributo deve ser obtido apenas de uma relação e a chave da relação faz parte da visão

- visão “updatable” : uma atualização na visão é mapeada para exatamente atualização de uma tupla em uma relação base.

- visão “insertable” : não há ambiguidade em qual relação a inserção será realizada.

- visões com union, intersect e except não são “insertable”

11

Outro Exemplo

- Temos **somente** o resultado da visão:

```
SELECT comprador, vendedor, produto, loja
FROM Pessoa, Compra
WHERE Pessoa.cidade = "Curitiba" AND
      Pessoa.nome = Compra.comprador
```

- mas queremos responder a consulta:

```
SELECT comprador, vendedor
FROM Pessoa, Compra
WHERE Pessoa.cidade = "Curitiba" AND
      Pessoa.nome = Compra.comprador AND
      Pessoa.tel LIKE '206 543 %'.
```

15

Respondendo Consultas usando Visões

- O que acontece se quisermos usar uma visão para responder uma consulta?

- Para que?

- integração, visões materializadas
- responder consultas sobre fontes de dados na web.

12

Outro Exemplo

- Temos **somente** o resultado da visão:

```
SELECT comprador, vendedor, produto, loja
FROM Pessoa, Compra, Produto
WHERE Pessoa.cidade = "Curitiba" AND
      Pessoa.nome = Compra.comprador AND
      Compra.produto = Produto.nome
```

- mas queremos responder a consulta:

```
SELECT comprador, vendedor
FROM Pessoa, Compra
WHERE Pessoa.cidade = "Curitiba" AND
      Pessoa.nome = Compra.comprador
```

16

Mais um Exemplo

- Temos o resultado da consulta:

```
SELECT comprador, vendedor, SUM(preco)
FROM Compra
WHERE Compra.loja = 'Wal Mart'
Group By comprador, vendedor
```
- e queremos responder:

```
SELECT vendedor, SUM(preco)
FROM Compra
WHERE Compra.loja = 'Wal Mart'
Group By vendedor
```

E se tivéssemos a segunda e quiséssemos reponder a primeira? 17

Último Exemplo

- Temos o resultado da consulta:

```
SELECT vendedor, comprador, Count(*)
FROM Compra
WHERE Compra.loja = 'Wal Mart'
Group By vendedor, comprador
```
- e queremos responder a seguinte consulta:

```
SELECT vendedor, Count(*)
FROM Compra
WHERE Compra.loja = 'Wal Mart'
Group By vendedor
```

18

O Problema Geral

- Dado um conjunto de visões V_1, \dots, V_n , e uma consulta Q , podemos responder Q utilizando somente o resultado de V_1, \dots, V_n ?
- Por que a problema é importante?
 - Podemos responder consultas de forma mais eficiente
 - Podemos responder consultas de fontes WWW de maneira sistemática.

19

Consultas de fontes WWW

- Assuma um esquema virtual do WWW, por exemplo:
 - `Disciplina(num, univ, nome, prof, periodo)`
- Fontes de dados da web contêm respostas de uma visão do esquema virtual:

```
UFPR: SELECT num, titulo, prof
FROM Disciplina
WHERE univ= 'UFPR' AND periodo= '1/06'
USP: SELECT num, titulo, prof, periodo
FROM Disciplina
WHERE univ= 'USP'
```

Consulta: encontre todos os professores que lecionam "Banco de Dados" 20