

Bases de Dados



Universidade do Porto

Faculdade de Engenharia

FEUP

1

- **INTRODUÇÃO**
- **MODELOS CONCEPTUAIS**
 - Diagrama de Classes UML
 - Modelo Entidade-Associação (E-A)
- **MODELO RELACIONAL**
- **LINGUAGEM DE DEFINIÇÃO DE DADOS**
- **INTERROGAÇÃO DE DADOS**
 - Álgebra relacional
 - Linguagem de Manipulação de Dados (LMD)
 - Datalog (bases de dados dedutivas)

Observação: baseado em slides desenvolvidos pelo Prof. Jeffrey D. Ullman

Índice

2

- Vistas
- Vistas materializadas
- Índices

Vistas

3

- Uma *vista* é uma relação definida com base em tabelas (designadas *tabelas base*) e outras vistas.
- Há dois tipos de vistas:
 1. *Virtual*: não é guardada na base de dados; É, tão só, uma instrução SELECT para construir a relação.
 2. *Materializada*: é guardada na base de dados.

Vistas

4

- Declara-se da seguinte forma:
`CREATE [MATERIALIZED] VIEW
 <nome> AS <instrução SELECT>;`
- Por defeito é virtual.
- A opção *MATERIALIZED* não existe em SQLite.

Vistas

5

Definição de vistas

- **PodeBeber(cliente, cerveja)** é uma vista com os pares cliente-cerveja sendo que esse cliente frequenta pelo menos um bar que serve essa cerveja:

```
CREATE VIEW PodeBeber AS
  SELECT cliente, cerveja
  FROM Frequenta, Vende
  WHERE Frequenta.bar = Vende.bar;
```

Vistas

6

Definição de vistas

- Utiliza-se uma vista como se fosse uma tabela.
- Exemplo de uma consulta:

```
SELECT cerveja FROM PodeBeber  
WHERE cliente = 'Mariana';
```

Vistas

7

Definição de vistas

- Tipicamente, é impossível modificar uma vista virtual pelo simples facto de ela não existir.
- No entanto, os gatilhos INSTEAD OF deixam-nos interpretar as alterações a vistas de forma a que elas façam sentido.
- **Exemplo:** a vista Sinergias tem tuplos (cliente, cerveja, bar) tais que o bar serve a cerveja, o cliente frequenta o bar e gosta da cerveja.

Vistas

8

Exemplo

CREATE VIEW Sinergias AS

SELECT Gosta.cliente, Gosta.cerveja, Vende.bar

FROM Gosta, Vende, Frequenta

WHERE Gosta.cliente = Frequenta.cliente

AND Gosta.cerveja = Vende.cerveja

AND Vende.bar = Frequenta.bar;

Vistas

9

Exemplo

- Não se pode inserir registros em Sinergias --- por ser uma vista virtual.
- Mas podemos utilizar um gatilho INSTEAD OF de forma a converter a inserção do tuplo (cliente, cerveja, bar) em três inserções, uma para cada tabela Gosta, Vende, e Frequenta.
 - Vende.preco tem de ser NULL.

Vistas

10

Exemplo: o gatilho **INSTEAD OF**

CREATE TRIGGER GatilhoVista

INSTEAD OF INSERT ON Sinergias

FOR EACH ROW

BEGIN

INSERT INTO Gosta VALUES (NEW.cliente, NEW.cerveja);

INSERT INTO Vende(bar, cerveja) VALUES (NEW.bar, NEW.cerveja);

INSERT INTO Frequenta VALUES (NEW.cliente, NEW.bar);

END;

Vistas materializadas

11

- **Problema:** sempre que uma tabela base muda, a vista materializada deve mudar.
 - É problemático refazer as vistas sempre que há uma mudança em alguma tabela base.
- **Solução:** Reconstrução Periódica da vista materializada, sob o risco de ficar desatualizada.

Vistas materializadas

12

Exemplo: inscrições dos alunos às UCs no SIFEUP

- A lista de alunos inscritos na turma 1MIEIC2 de Bases de Dados se não é podia ser uma vista materializada feita a partir da informação das inscrições.
- Vamos admitir que essa vista existe e que é actualizada uma vez por semana.
 - É possível um aluno inscrever-se nessa turma de Bases de Dados e não constar na lista dos alunos inscritos a essa mesma turma de Bases de Dados.

Vistas materializadas

13

Exemplo: um Armazém de Dados

- A cadeia de hipermercados Continente guarda todas as vendas realizadas em todos os hipermercados da cadeia numa base de dados.
- Durante a noite, as vendas efectuadas durante o dia são utilizadas para atualizar um *Armazém de Dados* = vistas materializadas das vendas.
- O Armazém de Dados é utilizado por analistas para analisar tendências e sazonalidades permitindo colocar os bens onde eles são mais necessários.

Índices

14

- *Índice* = estrutura de dados utilizada para acelerar as consultas à base de dados.
- Pode ser uma tabela de dispersão (tabela *hash*), mas é quase sempre uma árvore de pesquisa balanceada com nós gigantes chamada *B-tree*.
- Pode ainda ser do tipo *bitmap* especialmente quando:
 - O atributo tem poucos valores distintos (por ex.: género, masculino ou feminino) numa tabela com muitos registos;
 - Consultas que envolvem operadores de agregação pelo atributo indexado.

Índices

15

Declaração

- Sintaxe:

```
CREATE INDEX idxCerveja
```

```
ON Cervejas(empr);
```

```
CREATE INDEX idxVende ON Vende(bar,  
cerveja);
```

Índices

16

Utilização de índices (1)

- **Exemplo:** utilizar idxCerveja e idxVende para encontrar os preços de cervejas produzidas pela UNICER e vendidas no bar Pipa Velha.

Índices

17

Utilização de índices (2)

```
SELECT preco FROM Cervejas, Vende  
WHERE empr = 'UNICER' AND  
       Cervejas.nome = Vende.cerveja AND  
       bar = 'Pipa Velha' ;
```

1. Utiliza idxCerveja para obter todas as cervejas feitas pela UNICER.
2. Depois utiliza idxVende para obter os preços das cervejas, com bar = 'Pipa velha'

Índices

18

Afinação da base de dados

- A decisão mais importante a tomar pelo administrador da base de dados para que ela seja convenientemente rápida diz respeito à decisão sobre os índices a criar.
- **Vantagens:** Um índice acelera as consultas que dele tiram proveito;
- **Desvantagens:** Um índice torna mais lentas as modificações na tabela sobre a qual está definido, porque o índice também tem de ser modificado.

Índices

19

Exemplo: afinação da base de dados

- Suponhamos que as únicas coisas que fazemos sobre a base de dados das cervejas é:
 1. Inserir registos novos na tabela Cervejas (10%).
 2. Procurar o preço de uma dada cerveja num dado bar, que se obtem da tabela Vende (90%).
- Assim **idxVende** sobre Vende(bar, cerveja) seria útil, mas **idxCerveja** sobre Cervejas(empr) seria prejudicial.

Índices

20

A criação de índices

- As chaves primárias e candidatas são índices, não sendo por isso necessário criá-los.
- Tabelas com poucos registos não necessitam de índices.
- Aconselha-se a criação de índices nas seguintes situações:
 1. Tabelas grandes das quais se selecciona frequentemente uma pequena percentagem de registos;
 2. Colunas usadas para junção de tabelas (são tipicamente as chaves externas);
 3. Colunas com poucos valores repetidos ou com uma gama de valores grande;
 4. Colunas com poucos valores distintos em tabelas com muitos registos (índices bitmap, mas não existem em todos os SGBD).

Índices

21

Afinação automática (*tuning advisor*) - 1

- Uma ferramenta muito útil.
 - Porque a afinação manual consome muito tempo.
- O aconselhamento é feito a partir do *histórico das consultas*, por ex.:
 1. Escolhe aleatoriamente consultas do histórico das consultas executadas sobre a base de dados, ou
 2. O administrador da base de dados fornece uma amostra de consultas.

Índices

22

Afinação automática (*tuning advisor*) - 2

- O afinador automático gera candidatos a índices e avalia cada um deles na amostra.
 - Executa cada consulta da amostra assumindo que só existe esse índice.
 - Mede a melhoria/degradação do tempo médio de execução das consultas.

Índices

23

EXPLAIN

- EXPLAIN SELECT ...;
- EXPLAIN QUERY PLAN SELECT ...;