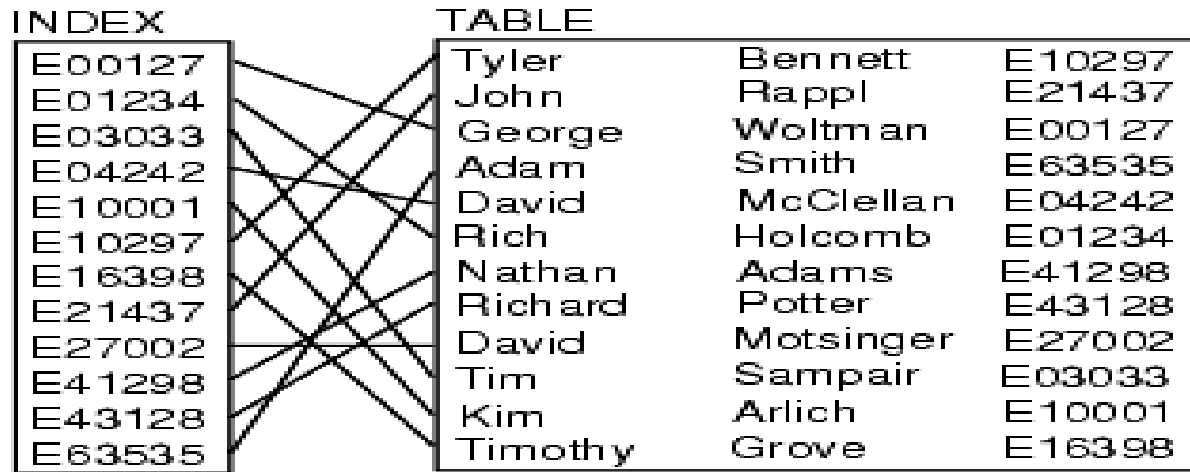


Unidade 5.1: DDL – Criação de Visões e Índices (CREATE VIEW e CREATE INDEX)

Prof. Rodrigo Baroni

ÍNDICES (CREATE INDEX)

- Estruturas físicas opcionais de banco de dados criadas para melhorar a performance no acesso.



ÍNDICES (CREATE INDEX)

- Caberá ao otimizador do SGBD optar ou não pelo uso do índice (DML não-procedural no SQL).
- Na falta de índices, o SGBD faz uma varredura completa da tabela (*full table scan*).
- Os comandos SELECT que envolvem ORDER BY (ordenação) tendem a ficar mais rápidos após a criação de índices pelos campos de ordenação.
- Os índices devem ser criados com prudência.

ÍNDICES (CREATE INDEX)

- Para escolher bem os índices, analisa-se quais campos da tabela participam das expressões WHERE de comandos de SELECT, UPDATE e DELETE.
- No entanto, o uso excessivo de índices pode ser prejudicial ao desempenho, pois todo comando que atualiza a tabela origem (Insert, Update, Delete) pode gerar uma alteração no índice.

ÍNDICES (CREATE INDEX)



- Os índices geralmente são definidos em conjunto pelo DBA e pelo desenvolvedor / engenheiro de software, que conhecem quais são as consultas mais críticas que são mais frequentes e envolvem um maior volume de acesso a dados.
- Quando linhas são adicionadas, alteradas ou apagadas, todos os índices são atualizados sem necessidade de ação externa alguma.
- Índices são partes importantes do “*Tuning*” do SGBD

ÍNDICES (CREATE INDEX)

- Os índices são criados usualmente para as chaves primárias e para as chaves estrangeiras, visando agilizar os comandos que envolvem junção de tabelas.
- O índice deve ser o mais seletivo possível.
- Colunas muito usadas em cláusulas WHERE também são boas candidatas para índices.



ÍNDICES (CREATE INDEX)



- Os índices podem ou não ser únicos (UNIQUE).
- O índice é único quando não se permitem repetições no conjunto de campos que compõem o índice. Assim sendo, o índice pela chave primária é sempre UNIQUE.
- Em alguns SGBDs, este índice pela chave primária já é gerado automaticamente.

Índice Único e Não Único

Índices podem ser únicos ou não únicos

Índice único garante que nenhum conjunto de duas linhas de uma tabela tenham valor repetido na coluna ou colunas da chave do índice.

Chaves primárias e únicas de tabelas possuem índices únicos que garantem unicidade de valor.

Índices não únicos permitem valores duplicados para a coluna ou colunas do índice.

Índice Composto

Múltiplos índices podem existir para a mesma tabela

Vários índices usando as mesmas colunas podem ser criados especificando ordens diferentes das colunas

```
CREATE INDEX empregado_ix1  
ON employees  
(department_id, manager_id);
```

```
CREATE INDEX empregado_ix2  
ON employees  
(manager_id, department_id);
```

ÍNDICES (CREATE INDEX)

- CREATE [UNIQUE] [**CLUSTERED** | **NONCLUSTERED**] INDEX
Nome-do-índice ON tabela (coluna ou lista-de-colunas)
- DROP INDEX *Nome-do-índice*
- Exemplos:

```
CREATE UNIQUE INDEX CLUSTERED IX_Funcionarios ON  
Funcionarios (Matricula)
```

```
CREATE INDEX IX_salario ON Funcionarios (CodDepto, Salario)
```

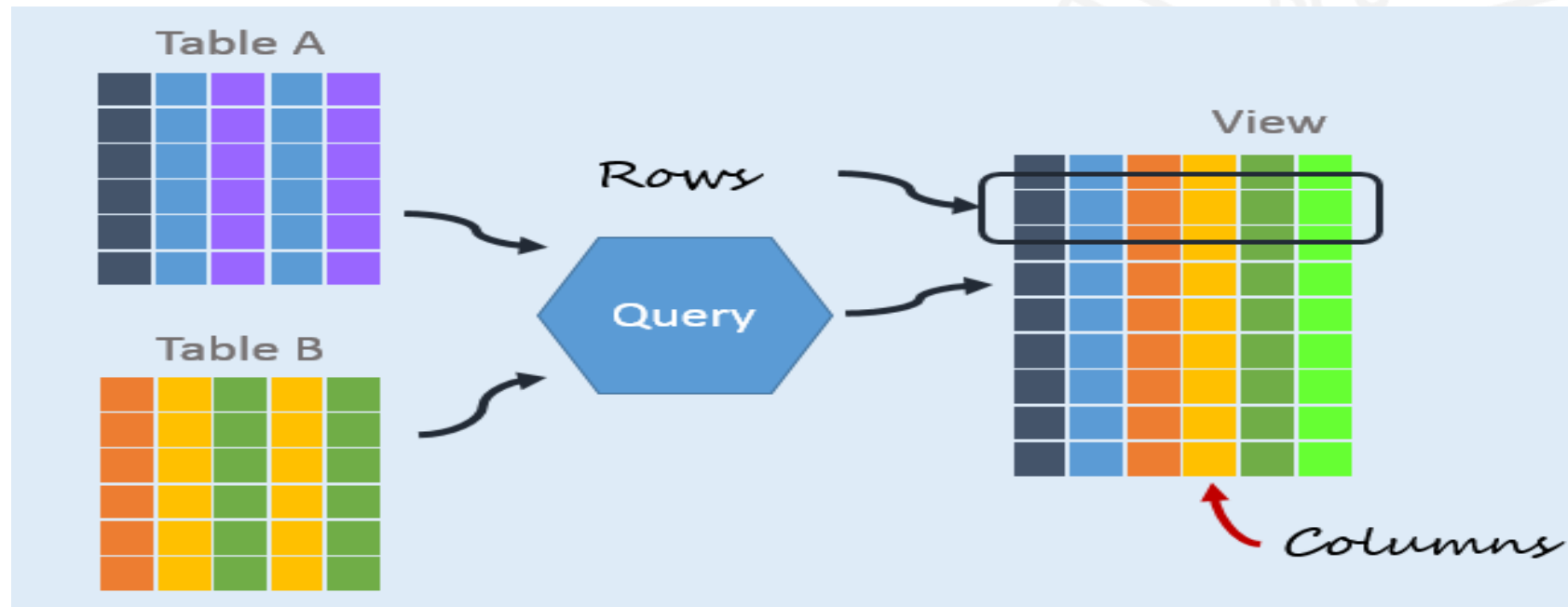
```
CREATE INDEX Ind_Regional ON Clientes (Estado Asc, Cidade Asc)
```

Visões (CREATE VIEW)

- A visão é uma estrutura lógica que é montada dinamicamente a partir de um SELECT.
- Através da visão, consegue-se selecionar as linhas e colunas desejadas de uma tabela ou mais tabelas e dar um nome lógico (nome da visão) para o resultado.



Visões (CREATE VIEW)



Visões (CREATE VIEW)

- Utiliza-se visão para armazenar os selects mais frequentes no próprio banco de dados
- SGBD não armazena as linhas resultantes da execução da visão.
- A segurança é uma das vantagens, pois pode-se dar permissão de acesso para a visão e não para a tabela inteira
- Com isso, os programas aplicativos ou usuários finais podem acessar a visão, simplificando a sintaxe dos comandos.
- As visões podem ser encadeadas, mas isto provoca uma queda de performance.

Utilidade de Visões

Prover nível adicional de segurança de tabela por restringir acesso para um determinado conjunto de linhas e colunas da tabela

Esconder a complexidade dos dados

- Junções de tabelas e cálculos podem estar presentes na definição da visão

Apresentar os dados em uma perspectiva diferente da tabela base

- Renomear nome de coluna

Isolar aplicações de mudanças na definição de tabelas base

- A inserção de nova coluna na tabela base não afeta a visão

Criação de Visão

```
• CREATE VIEW staff AS  
SELECT employee_id, last_name, job_id,  
       manager_id, department_id  
FROM employees;
```

Colunas data de
admissão e salário
não são mostradas
pela visão

Remoção de Visão

```
DROP VIEW staff;
```

Base
Table

| employees | | | | | | |
|-------------|-----------|------------|------------|-----------|--------|---------------|
| employee_id | last_name | job_id | manager_id | hire_date | salary | department_id |
| 203 | marvis | hr_rep | 101 | 07-Jun-94 | 6500 | 40 |
| 204 | baer | pr_rep | 101 | 07-Jun-94 | 10000 | 70 |
| 205 | higgins | ac_rep | 101 | 07-Jun-94 | 12000 | 110 |
| 206 | gietz | ac_account | 205 | 07-Jun-94 | 8300 | 110 |

View

| staff | | | | |
|-------------|-----------|------------|------------|---------------|
| employee_id | last_name | job_id | manager_id | department_id |
| 203 | marvis | hr_rep | 101 | 40 |
| 204 | baer | pr_rep | 101 | 70 |
| 205 | higgins | ac_rep | 101 | 110 |
| 206 | gietz | ac_account | 205 | 110 |

Visões (CREATE VIEW)

CREATE VIEW nome-da-visão (apelidos para os campos – opcional) AS SELECT

- Exemplo:

```
CREATE VIEW SalarioDepto (Depto, Media_Salarial, Numero)
AS SELECT Depto, AVG(Salario), COUNT(*)
FROM Funcionarios
GROUP BY Depto
```



Visões (CREATE VIEW)

- Pode-se criar uma visão de uma visão

```
CREATE VIEW MaioresSalarios
```

```
AS SELECT * FROM SalarioDepto
```

```
WHERE Media_Salarial > 8000
```





PUC Minas

Virtual