

PUC-Rio  
Data: 07/04/2018  
Aluno: Wellington Bezerra Chaves  
Matrícula: 1413383

## Laboratório 2

### **Estruturas de dados**

#### **1.0 TABELAS**

Em SQLServer, as tabelas são armazenadas no banco de dados como um conjunto de páginas de tamanho fixo 8 KB. As páginas da tabela são armazenadas por padrão em uma única partição no grupo de arquivos padrão PRIMARY.

Uma tabela também pode ser armazenada em várias partições, em que cada grupo de linhas será armazenado em uma partição específica, em um ou mais grupos de arquivos, com base em uma coluna específica.

Cada partição de tabela contém linhas de dados em um heap ou estrutura de índice clusterizada:

##### 1.1 Tabela Heap

Tabela HEAP não possui um índice clusterizado de nenhum tipo, ou seja, é uma tabela em que os dados são gravados conforme a ordem cronológica das inserções.

##### 1.2 Tabela clusterizada

Uma tabela clusterizada possui um índice predefinido em uma coluna ou várias colunas da tabela que define a ordem de armazenamento das linhas nas páginas de dados e a ordem das páginas na tabela, com base na chave de índice clusterizada.

A estrutura usada para a elaboração do índice depende do SGBD e do tipo de dados das colunas usadas no índice: árvore B, árvore R, entre outras.

#### **2.0 SQL\_VARIANT**

Um tipo de dado que armazena valores de vários tipos de dados. Ele permite que esses objetos de banco de dados sejam compatíveis com valores de outros tipos de dados.

Pode ser usado em colunas, parâmetros, variáveis e retorno de funções. Por exemplo, uma definida como sql\_variant pode armazenar valores int, binary e char.

### **DICIONÁRIO DE DADOS**

Um dicionário de dados é um grupo de tabelas habilitadas apenas para leitura ou consulta que mantém as seguintes informações:

- Descrição de objetos
- Alocação de espaço
- Índices
- Privilégio dos usuários

No Dicionário de Dados, tem-se a liberdade de definir novos campos, com seus respectivos tipos, além de ser possível a criação de novas abas para colocar esses novos campos.

Seu benefício é a consistência entre itens de dados através de diferentes tabelas. Por exemplo, diversas tabelas podem conter CPF: utilizando uma definição de um dicionário de dados bem feito, o formato do campo 'cpf' definido com "999.999.999-99" deverá ser obedecido em todas as tabelas que utilizarem esta informação.

### 1.0 COL\_NAME:

Retorna o nome da coluna da tabela, baseado no id da tabela e da coluna.

Sintaxe: **col\_name( table\_id, column\_id )**

table\_id : número de identificação da tabela que contém a coluna (int)

column\_id: número de identificação da coluna (int)

### 2.0 COL\_LENGTH:

Retorna o tamanho da coluna em bytes

Sintaxe: **col\_lenght( 'table', 'column' )**

'table': nome da tabela

'column': nome da coluna

### 3.0 DB\_ID:

Retorna o número de especificação do database

Sintaxe: **DB\_ID ( [ 'database\_name' ] )**

'database\_name': nome do database

### 4.0 OBJECT\_NAME:

Retorna o nome do objeto de banco de dados para objetos com escopo de esquema.

Sintaxe: **OBJECT\_NAME ( object\_id [, database\_id ] )**

object\_id: identificador do objeto

database\_id: identificador do database que contem o objeto