	<b>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO</b>	<b>CAMPUS</b>  <b>São Paulo</b>
<b>Banco de Dados</b>		<b>Aula 19</b>
<b>Professor: Francisco Veríssimo Luciano e Prof. Glauber da Rocha Balthazar</b> <b>E-mail: fvluciano@hotmail.com</b>		

## 8 - Aula 8 - Modelo Físico de Banco de Dados - Parte II

### 8.1 - Populando as Tabelas

A seguir são apresentados alguns conceitos avançados sobre construção de tabelas.

1. Execute os comandos abaixo, caso não tenham sido feitos anteriormente.

```
CREATE DATABASE bd_aluno --criação do BD

USE bd_aluno - acionar o uso do BD

--criação das tabelas

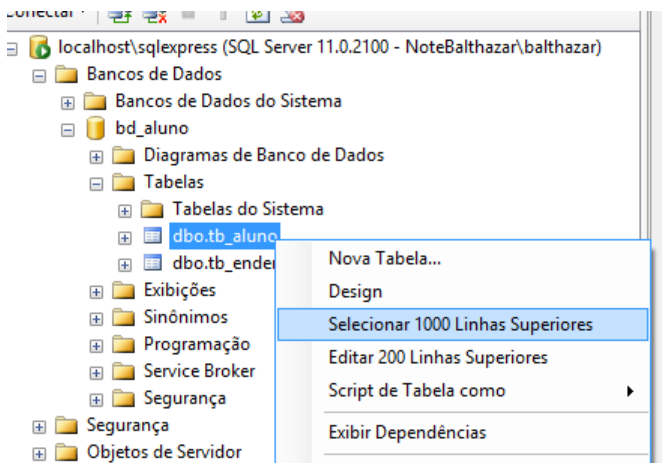
CREATE TABLE tb_aluno (
prontuario int PRIMARY KEY IDENTITY(0,1),
nome varchar(255)
)

CREATE TABLE tb_endereco (
cod_endereco INT PRIMARY KEY IDENTITY(0,1),
rua VARCHAR(200),
numero INT,
bairro VARCHAR(150),
cidade VARCHAR(150),
estado VARCHAR(2),
cep VARCHAR(7),
prontuario int REFERENCES tb_aluno(prontuario)
)
```

2. Para popular os dados de uma tabela, utilize o comando INSERT INTO:

```
INSERT INTO tb_aluno (nome) values ('joao')
INSERT INTO tb_aluno (nome) values ('maria')
INSERT INTO tb_aluno (nome) values ('pedro')
INSERT INTO tb_aluno (nome) values ('ana')
INSERT INTO tb_aluno (nome) values ('clovis')
```

3. Para visualizar os dados clique com o botão direito sobre a tabela e escolha "Selecionar 1000 linhas superiores"



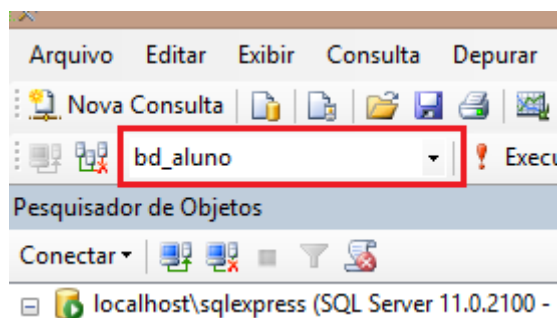
4. Os resultados serão exibidos:

The screenshot shows the 'Resultados' (Results) pane in SQL Server Enterprise Manager. It displays a table with two columns: 'prontuario' and 'nome'. The data is as follows:

	prontuario	nome
1	0	joao
2	1	maria
3	2	pedro
4	3	ana
5	4	clovis

## 8.2 - Selecionar dados de uma tabela

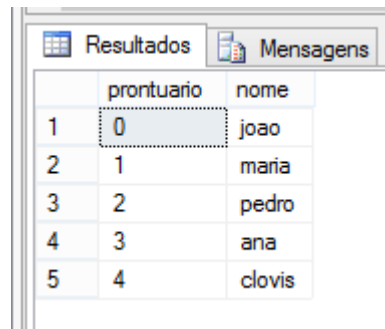
1. Obs.: antes, visualize se a base em uso é a BD\_ALUNO:



2. Para selecionar os dados inseridos em uma tabela basta utilizar o comando SELECT:

```
SELECT
    tb_aluno.*
FROM
    tb_aluno
```

3. Os dados serão exibidos:



	prontuario	nome
1	0	joao
2	1	maria
3	2	pedro
4	3	ana
5	4	clovis

## 8.3 - Conceitos avançados em SELECT

### 1. Definindo alias

Um alias (apelido) é um apelido que pode ser dado a coluna. Funciona apenas para a exibição do apelido. O nome da coluna permanece inalterado na tabela. Quando forem usados dois ou mais nomes no alias é obrigatório usar apóstrofos (exemplo: ~~Nome do Aluno~~ 'Nome do Aluno')

```
SELECT
    tb_aluno.prontuario AS PRONT,
    tb_aluno.nome AS NOME_ALUNO
FROM
    tb_aluno
```

### 2. Definindo ordenação

```
SELECT
    *
FROM
    tb_aluno
ORDER BY
    nome DESC --ou ASC
```

### 3. Contando a quantidade de registros

```
SELECT
    Count(*)
FROM
    tb_aluno
```

### 4. Selecionando registros distintos

Antes, acrescente alguns nomes duplicados:

```
INSERT INTO tb_aluno (nome) values ('pedro')
INSERT INTO tb_aluno (nome) values ('ana')
INSERT INTO tb_aluno (nome) values ('clovis')
```

Em seguida, faça um select normal:

```
SELECT
    *
FROM
    tb_aluno
```

Em seguida, utilize o DISTINCT:

```
SELECT
    DISTINCT (nome)
FROM
    tb_aluno
```

## 5. Selecionando uma quantidade exata de registros

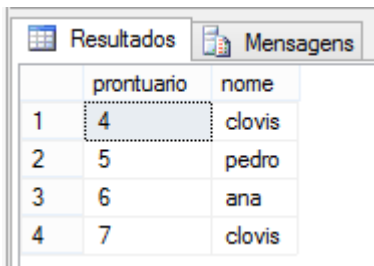
```
SELECT TOP 4
    *
FROM
    tb_aluno
```

## 8.4 - Selecionar dados com cláusula WHERE

1. Para realizar consultas com condições utilize a cláusula where, veja o exemplo a seguir:

```
SELECT
    *
FROM
    tb_aluno
WHERE
    prontosuario >3
```

2. O resultado será apresentado:



The screenshot shows a window titled 'Resultados' (Results) with a 'Mensagens' (Messages) tab. It displays a table with two columns: 'prontosuario' and 'nome'. The table contains four rows of data. The first row has '1' in the first column and '4' in the second column, with 'clovis' in the third column. The second row has '2' in the first column and '5' in the second column, with 'pedro' in the third column. The third row has '3' in the first column and '6' in the second column, with 'ana' in the third column. The fourth row has '4' in the first column and '7' in the second column, with 'clovis' in the third column.

	prontosuario	nome
1	4	clovis
2	5	pedro
3	6	ana
4	7	clovis

## 8.5 - Condições avançadas com a cláusula WHERE

- Para realizar consultas em intervalos

```
SELECT
    *
FROM
    tb_aluno
WHERE
    prontosuario BETWEEN 2 AND 5
```

- Partes de uma string

1. Todos que iniciam com 'a'

```
SELECT
    *
FROM
    tb_aluno
WHERE
    nome LIKE 'a%'
```

## 2. Todos que terminam com 'o'

```
SELECT
    *
FROM
    tb_aluno
WHERE
    nome LIKE '%o'
```

## 3. Todos que tenham 'a' em qualquer parte da palavra

```
SELECT
    *
FROM
    tb_aluno
WHERE
    nome LIKE '%a%'
```

## 4. Todos que NÃO tenham 'a' em qualquer parte da palavra

```
SELECT
    *
FROM
    tb_aluno
WHERE
    nome NOT LIKE '%a%'
```

### • Especificar múltiplos valores

```
SELECT
    *
FROM
    tb_aluno
WHERE
    nome IN ('PEDRO', 'ANA')
```

### • Operadores Lógicos

Operador	Descrição
=	Igual
<>	Não é igual. Nota: Em algumas versões do SQL este operador pode ser escrito como !=
>	Maior que
<	Menor que
>=	Maior ou igual
<=	Menor ou igual
BETWEEN ... AND ...	Entre ... E ...
LIKE	Procura um padrão
IN	Especifica vários valores possíveis para uma coluna

## 8.6 - UPDATE e DELETE com WHERE

- **UPDATE:**

Serve para atualizar os dados de um determinado registro na tabela.

```
UPDATE
    tb_aluno
SET
    nome = 'joaquim'
WHERE
    prontuario = 1
```

**obs.:** caso a cláusula WHERE venha a ser omitida **TODOS** os registros serão afetados.

- **DELETE:**

Serve para remover (deletar) um ou mais registro(s) de uma tabela, de acordo com condição(ões) pre definida(s).

DELETE FROM <nome\_tabela> WHERE <condição>

```
DELETE
    tb_aluno
WHERE
    prontuario = 1
```

**Obs1.:** somente o registro com prontuário = 1 será afetado.

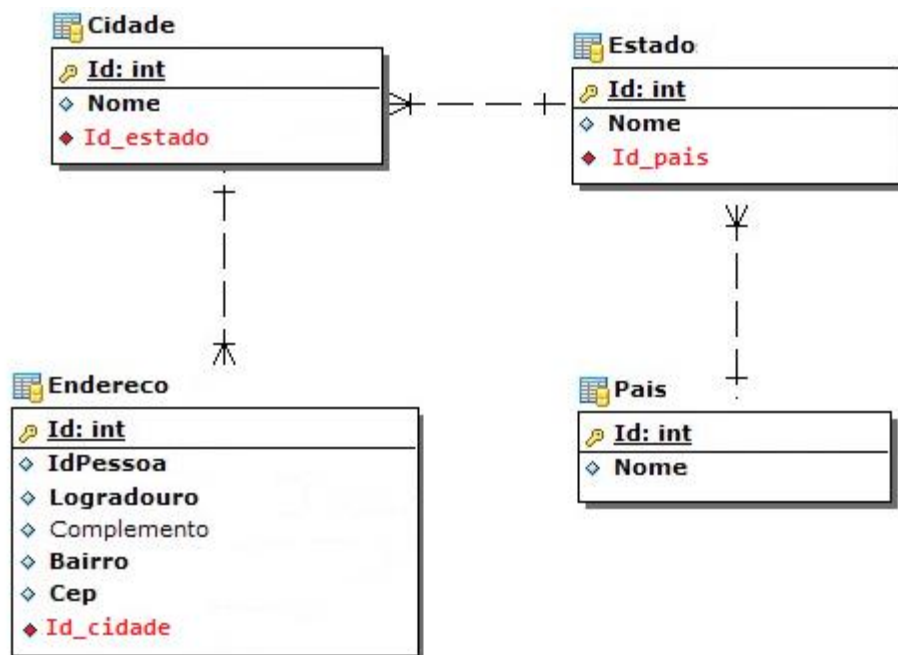
**Obs2.:** caso a cláusula WHERE venha a ser omitida **TODOS** os registros serão afetados. Ou seja, para excluir todas as linhas de uma tabela, existem duas opções. Uma é usar um comando DELETE sem condição WHERE. A outra opção é o comando TRUNCATE TABLE, que quase sempre é mais rápido que o DELETE, especialmente em tabelas grandes:

Sintaxe: **TRUNCATE TABLE** <nome da tabela>

A básica entre os dois comandos está no fato de o TRUNCATE TABLE não guardar logs.

## 8.7 - Exercícios

Para o modelo apresentado a seguir, faça:



1. Construa um banco de dados chamado "BD\_ENDERECO"
2. Construa as tabelas, lembrando-se:
  - todas as chaves primárias devem ser especificadas
  - todas as chaves estrangeiras devem ser especificadas
  - todas as chaves primárias deverão ser **IDENTITY(1,1)**
  - deverão existir alguns campos **NOT NULL** (todos em negrito)
  - um novo índice deverá ser criado para tabela (a seu critério o atributo)
3. Popule as tabelas com:
  - 1 país
  - 3 estados
  - 3 cidades para cada estado
  - 2 endereços para cada cidade
4. Altere o nome de um estado para "Paraná"
5. Acrescente a cidade "Cascavel" ao estado do "Paraná"
6. Delete 3 endereços
7. Recupere as cidades que comecem com a letra 'c'
8. Altere o nome da cidade "Cascavel" para "Curitiba"

- **Referências**

MECENAS, Ivan. OLIVEIRA, Vivianne. **Banco de Dados: Do modelo conceitual à implementação física**. Rio de Janeiro: Alta Books, 2005. 180 p.

ELMASRI, Ramez E. & NAVATHE, Shamkant B, **Sistema de Bancos de Dados É Fundamentos e Aplicações**, São Paulo: ADDISON WESLEY BRASIL, 2005.

MACEDO, Diego. **Modelagem Conceitual, Lógica e Física de Dados**. 2011. Acessado em <14/02/2015>. Disponível em <<http://www.diegomacedo.com.br/modelagem-conceitual-logica-e-fisica-de-dados/>>.

NASCIMENTO, Manuela Cicco do. **Br2Oracle :Geração automática de esquema relacional a partir da ferramenta BrModelo para o SGBD Oracle**. Monografia para conclusão de graduação em Ciência da Computação. Orientador: Prof. Fernando da Fonseca de Souza. Recife, 30 de Janeiro de 2008. Universidade Federal de Pernambuco - Centro de Informática.

RABELO, Ricardo J. **Banco de Dados. MaterialDAS5316**. 2009. Acessado em <14/02/2015>. Disponível em <[http://www.das.ufsc.br/~rabelo/Ensino/DAS5316/MaterialDAS5316/Banco\\_de\\_Dados/bd\\_intro.pdf](http://www.das.ufsc.br/~rabelo/Ensino/DAS5316/MaterialDAS5316/Banco_de_Dados/bd_intro.pdf)>.

MACHADO, Felipe Nery Rodrigues. **Projeto de banco de dados: uma visão prática**. -- 16 ed. rev.e atual. -- São Paulo: Érica, 2009.

ROCHA, Ricardo. **Bases de Dados 2005/2006. Parte VIII: Normalização**. DCC - Faculdade de Ciências da Universidade do Porto. Acessado em <18/02/2015>. Disponível em <<http://www.dcc.fc.up.pt/~ricroc/aulas/0506/bd/apontamentos/parteVIII.pdf>>.

PIVETTA, Elisa Maria. BANCO DE DADOS: Modelagem de Dados . **Normalização**. UFSM/CAFW, 2011. Acessado em <18/02/2015>. Disponível em <<http://www.cafw.ufsm.br/~elisa/bd/normalizacao1.pdf>>.

FROZZA, Angelo Augusto. **BANCO DE DADOS - NORMALIZAÇÃO**. Instituto Federal Catarinense. 2011. Acessado em <18/02/2015>. Disponível em <<http://www.ifc-camboriu.edu.br/~frozza/2011.1/IE10/IE10-BDD-Aula003-Normalizacao-Revisao.pdf>>.

IFSudesteMG. **Exercícios sobre normalização**. Acessado em <19/02/2015>. Disponível em <[http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1997751601\\_Exemplos\\_sobre\\_normizacao%20resolvido.pdf](http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1997751601_Exemplos_sobre_normizacao%20resolvido.pdf)>.

SANTOS, Ricardo. **Módulo 2: correção exercícios - normalização**. Acessado em <19/02/2015>. Disponível em <[http://ricardosantos.net/arquivos/Normalizacao\\_resolvidos.pdf](http://ricardosantos.net/arquivos/Normalizacao_resolvidos.pdf)>.

LOPES, ABRAHÃO. **AULA 11-12 Modelo Conceitual, Lógico e Físico, Entidade-Relacionamento**. Instituto Federal do Rio Grande do Norte, campus Mossoró, 2014. Acessado em <19/02/2015>. Disponível em <<http://docente.ifrn.edu.br/abrahamlopes/2014.1-integrado/3.2401.1v-prog-b-dados/modelo-conceitual-logico-fisico-relacionamento-cardinalidade>>.

BIANCHI, Wagner. **Entendendo e usando índices - Parte 1**. 2008. Acessado em <19/02/2015>. Disponível em <<http://www.devmedia.com.br/entendendo-e-usando-indices-parte-1/6567>>.