

MODELAGEM DE DADOS
Prof. Milton Palmeira Santana



Crie um banco de dados chamado **SALA_DE_AULA**.
Coloque o banco em **USO**.

Crie uma tabela **ALUNOS** com os seguintes dados:

- » NUM_ALUNO INTEIRO, CHAVE PRIMÁRIA,
 AUTOINCREMENTO
- » NOME ALFANUMÉRICO NÃO NULO,
- » DATA_NASCIMENTO DATETIME,
- » IDADE INTEIRO,
- » E_MAIL ALFANUMÉRICO
- » FONE_RES ALFANUMÉRICO
- » FONE_COM ALFANUMÉRICO
- » CELULAR ALFANUMÉRICO
- » PROFISSAO ALFANUMÉRICO
- » EMPRESA ALFANUMÉRICO

Insira 5 registros na tabela ALUNOS.

SELECT

- » O comando SELECT é utilizado para consultar todos os dados de uma fonte de dados ou uma parte deles.
- » Utiliza-se a seguinte sintaxe:



```
SELECT * FROM TABELA
```

- » Onde * refere-se a todos os campos da tabela.
- » Caso se deseje apenas um campo da tabela basta trocar o * pelo nome do campo.

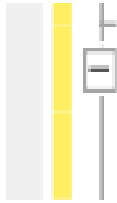
SELECT

- » Exemplo:
- » Vamos consultar o nome, Data de Nascimento e Idade de todos os alunos.

```
-- Consultar Nome, Data de Nascimento e idade  
-- de todos os alunos  
SELECT NOME, DATA_NASCIMENTO, IDADE FROM ALUNOS
```

SELECT - ALIAS

- » O nome de uma coluna pode ser substituído por uma espécie de apelido criado para facilitar a visualização.
- » Utilizamos para isso a cláusula **AS**.



```
SELECT NOME, DATA_NASCIMENTO AS [DATA DE NASCIMENTO]  
FROM ALUNOS
```

Ordenação de dados

- » Utilizamos o comando **ORDER BY** para ordenar os dados de uma consulta.

```
-- Ordenando os dados  
SELECT * FROM ALUNOS ORDER BY NOME
```

- » Podemos também ordenar em ordem DECRESCENTE.

```
-- Ordenando os dados em ordem decrescente  
SELECT * FROM ALUNOS ORDER BY NOME DESC
```

TOP

» Exemplos:

```
-- Lista os 5 primeiros ALUNOS
```

```
SELECT TOP 5 * FROM ALUNOS
```

```
-- Lista os 5 primeiros ALUNOS por ordem  
-- de NOME
```

```
SELECT TOP 5 * FROM ALUNOS  
ORDER BY NOME
```


Cláusula WHERE

- » Podemos utilizar filtros em nossas consultas. Para isso utilizamos a cláusula WHERE.
- » Ela servirá para verificarmos quais dados queremos consultar e através de quais filtros.

```
-- Selecionar alunos menores de idade (18)  
SELECT * FROM ALUNOS  
      WHERE IDADE < 18
```

Utilizando o comando SELECT elabore os seguintes relatórios:

- » Mostre o NOME, DATA_NASCIMENTO, IDADE de todos os alunos
- » Mostre o NOME, IDADE, E_MAIL, CELULAR E PROFISSAO de todos os alunos
- » Mostre o NOME, FONE_RES, CELULAR, PROFISSAO e EMPRESA de todos os alunos.
- » Mostre o NOME, DATA_NASCIMENTO e PROFISSAO do ALUNO com CÓDIGO = 1.

- » Mostre os alunos com PROFISSAO de Informática.
- » Mostre todos os alunos que tem profissão de ANALISTA DE SISTEMAS.
- » Mostre todos os alunos que trabalham na empresa ESCOLA.
- » Mostre todos os alunos que possuem idade superior a 18 anos.
- » Mostre todos os alunos que possuem idade inferior a 18 anos.

Cláusula WHERE – Operadores Lógicos

- » Os operadores lógicos serão utilizados quando desejarmos combinar expressões.
- » Operadores Lógicos: **AND**, **OR** ou **NOT**.
- » Exemplo:
- » Mostre todos os alunos que têm idade entre 16 e 18 anos.

```
-- Selecionar alunos idade entre 16 e 18 anos  
SELECT * FROM ALUNOS  
      WHERE IDADE >= 16 AND IDADE <= 18
```

- » Crie um banco de dados chamado PEDIDOS_VENDA;
- » Criar uma tabela chamada PRODUTOS, com os respectivos campos:

TABELA PRODUTOS

Campos	Descrição
Código do produto	INT IDENTITY PRIMARY
Nome do Produto	VARCHAR(50)
Código da Unidade	INT NOT NULL
Código da Categoria	INT NOT NULL
Quantidade em Estoque	INT
Quantidade Mínima	INT
Preço de Custo	DECIMAL(8,2)
Preço de Venda	DECIMAL(8,2)
Características Técnicas	VARCHAR(MAX)

- » Agora vamos realizar as consultas:
- » Listar da tabela PRODUTOS as colunas Código, DESCRIÇÃO, PRECO DE CUSTO, PRECO DE VENDA e calcule o lucro unitário.
- » Listar todos os campos da tabela PRODUTOS e calcule o lucro unitário, mostrando apenas os registros com lucro abaixo de 1,00.
- » Listar da tabela PRODUTOS todos os itens com valor de venda entre R\$ 100,00 e R\$ 450,00.
- » Listar da tabela PRODUTOS todos os itens com valor de venda entre R\$ 100,00 e R\$ 450,00 e com estoque abaixo do mínimo.

- » Listar da tabela PRODUTOS todos os itens com estoque abaixo do mínimo ou com estoque abaixo de 10.
- » Listar da tabela PRODUTOS todos os itens que não tenham quantidade em estoque.
- » Listar da tabela PRODUTOS todos os itens que não tiverem valor de venda entre R\$ 100,00 e R\$ 450,00.

Cláusula BETWEEN

A cláusula BETWEEN permite filtrar dados em uma consulta tendo como base uma faixa de valores.

Podemos utilizá-la ao invés de escrever uma série de WHERE contendo operadores relacionais.

Exemplo: Mostre todos os alunos que têm idade entre 16 e 18 anos. (WHERE)

```
-- Selecionar alunos idade entre 16 e 18 anos
```

```
SELECT * FROM ALUNOS
```

```
WHERE IDADE >= 16 AND IDADE <= 18
```

Cláusula BETWEEN

A cláusula BETWEEN permite filtrar dados em uma consulta tendo como base uma faixa de valores.

Podemos utilizá-la ao invés de escrever uma série de WHERE contendo operadores relacionais.

Exemplo: Mostre todos os alunos que têm idade entre 18 e 16 anos. (WHERE)

```
-- Selecionar alunos idade entre 16 e 18 anos
```

```
SELECT * FROM ALUNOS
```

```
WHERE IDADE BETWEEN 16 AND 18
```

Operador LIKE

O operador **LIKE** é empregado em situações em que utilizamos como base para realizar pesquisas as colunas que estão no formato texto (varchar, etc).

Podemos utilizá-lo para retornar consultas que se iniciam com determinada letra ou palavra.

O caractere % representa uma espécie de coringa.

Exemplo: Retorne todos os alunos com o nome Maria.

```
-- Retornar todos alunos com nome MARIA  
SELECT * FROM ALUNOS  
WHERE NOME LIKE 'MARIA%'
```

Operador LIKE

Um outro caractere utilizado com o operador **LIKE** é o sinal _

Ele equivale a um único caractere qualquer.

Exemplo: Selecionar todos os alunos com nome que sejam iniciados por qualquer caractere e tenha a segunda letra O.

```
-- Retornar todos alunos com nome iniciados  
-- por qualquer caractere e segunda letra seja O  
SELECT * FROM ALUNOS  
WHERE NOME LIKE '_O%'
```

Operador IN

O operador **IN** permite verificar se os valores indicados por uma lista estão presentes na tabela.

Exemplo:

Selecionar todos os alunos que contenham os códigos 1, 2, 3 ou 7.

```
-- SELECIONAR TODOS ALUNOS COM OS CÓDIGOS  
-- 1, 2, 3 ou 7  
SELECT * FROM ALUNOS  
       WHERE NUM_ALUNO IN (1,2,3,7)
```

Exercício

Após fazer compras em uma pequena loja de material de construção, você percebeu que o controle dos produtos levados pelos clientes era realizado em um caderno com fichas coladas para serem preenchidas de forma manual. Somente clientes antigos e de confiança possuem a regalia de pegar os produtos e pagar depois. Ao saberem que você é da área de informática, pediram que fizesse um banco de dados para agilizar o processo de controle.

Exercício

FICHA DE CONTROLE DE RETIRADA DE MATERIAIS

Controle Ficha nº:		Data:	
Cliente:	RG:	CPF:	
Endereço:	Cidade:	UF:	

Produtos				
Código	Descrição	Quantidade	Preço Unitário	Preço Total
Valor total a Pagar:				

Exercício

Nº de controle da ficha

- Data da nota
- Nome cliente
- RG cliente
- CPF cliente
- UF
- Cidade
- Código produto
- Descrição produto
- Quantidade
- Preço unitário
- Preço total item
- Valor total da nota

Exercício – 1 FN

Para aplicar a primeira forma normal temos que adicionar a chave primária.

Ficha (#numControle, DtNota, valorTotal, Cidade, Estado)

Cliente (#CPF, Nome, RG, Endereco)

Produto (#codProduto, Descricao, precoUnitario, quantidade, precoTotal)

Exercício – 2 FN

Para aplicar a segunda forma normal temos que criar as tabelas CIDADE e ESTADO e inserir as chaves estrangeiras (todos campos dependem exclusivamente da chave primária e não parte dela apenas).

Ficha (#numControle, DtNota, valorTotal, &idCidade, &siglaEstado)

Cidade (#idCidade, Cidade)

Estado (#siglaEstado, Estado)

Exercício – 3 FN

Para aplicar a terceira forma normal temos eliminar todos os campos dependentes (não poderá haver dependências funcionais entre os campos). Retirar preço total da tabela Produto.

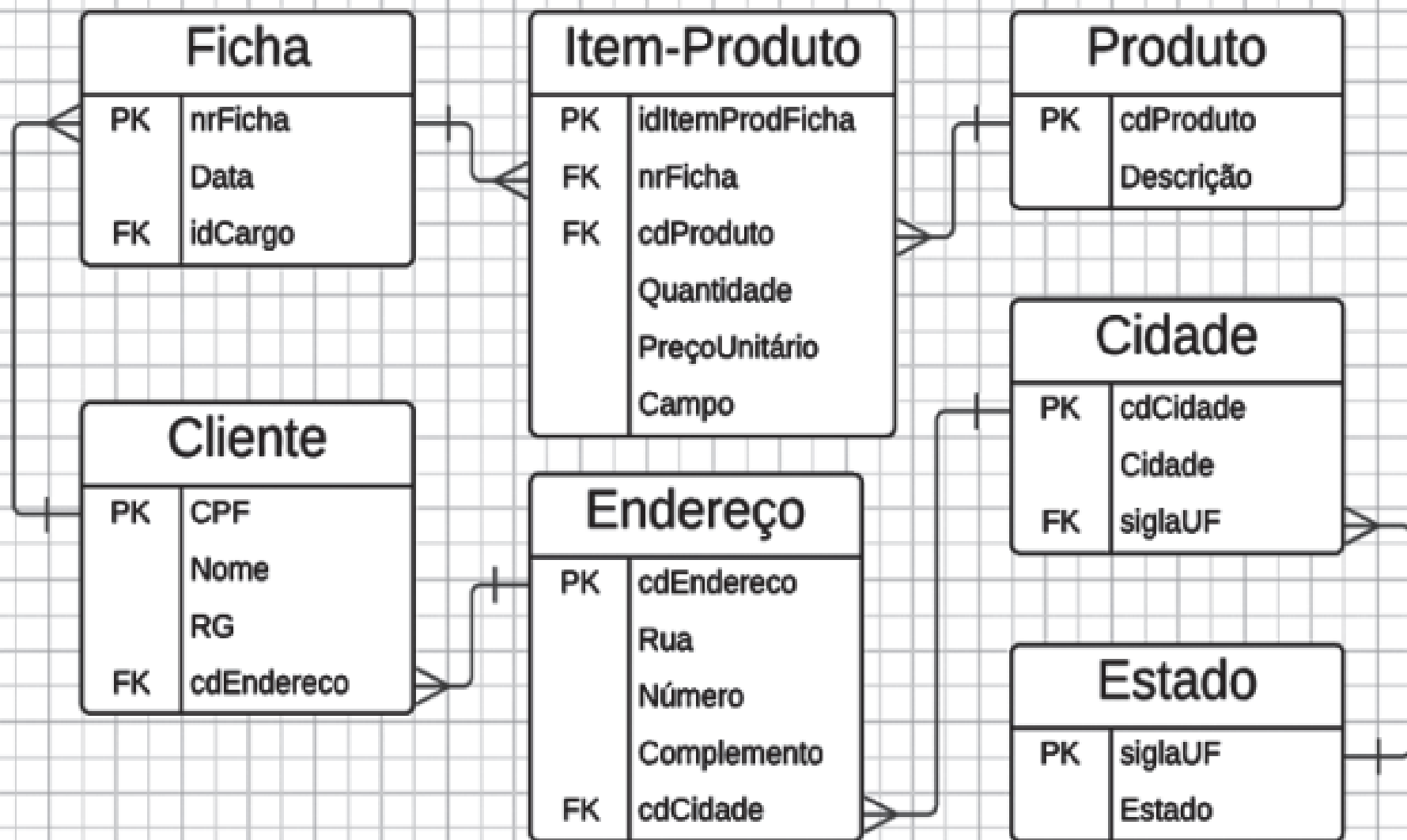
Exercício – 4 FN

Verificar se não há campos multivalorados e não atômicos.

Ficha (#numControle, DtNota, valorTotal, &idCidade, &siglaEstado)

Cidade (#idCidade, Cidade)

Estado (#siglaEstado, Estado)



BARBOZA, Fabrício Felipe Meleto; FREITAS, Pedro Henrique Chagas. **Modelagem e desenvolvimento de banco de dados**. Porto Alegre: SAGAH, 2018.

WERLICH, Claudia. **Modelagem de dados**. Londrina: Editora e Distribuidora Educacional S.A, 2018.

MANZANO, Jose Augusto Navarro Garcia. **Microsoft SQL Server 2016 Express Edition Interativo**. [S. l.]: ÉRICA, 2016.

MACHADO, Felipe Nery Rodrigues; ABREU, Mauricio Pereira de. **Projeto de Banco de Dados: Uma Visão Prática** - Edição Revisada e Ampliada. [S. l.]: ÉRICA, 2009.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de Gerenciamento de Bancos de Dados. [S. l.]: Amgh Editora, 2011.

ALVES, WILLIAM PEREIRA. Banco de Dados. São Paulo: Saraiva, 2014

CARDOSO, VIRGÍNIA M.; CARDOSO, GISELLE CRISTINA. SISTEMA DE BANCO DE DADOS. São Paulo: Saraiva, 2013



Anhanguera