



BOSCH
Invented for life

3.8 - Bancos de Dados Relacionais

inside.Docupedia Export

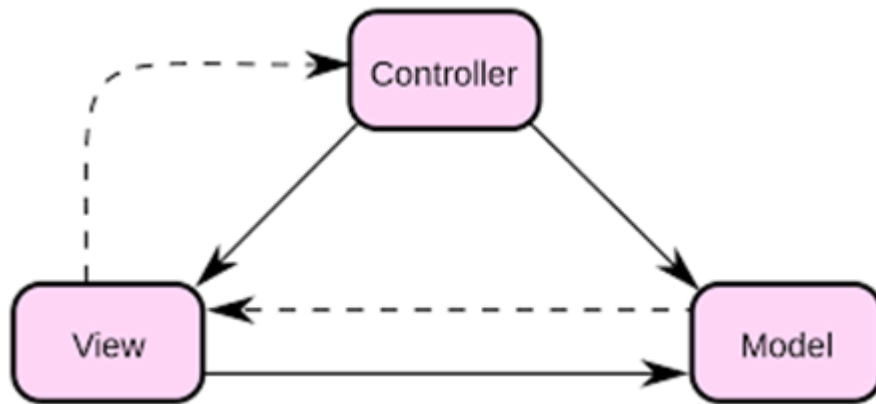
Author: Siqueira Joao (CtP/ETS)
Date: 20-Jan-2022 19:20

Table of Contents

1	Arquitetura de Software - MVC	3
2	O que é um banco de dados?	4
3	Banco de dados relacional	5
4	Banco de dados - MER	9
5	Banco de dados - tipos de dados	10
5.1	Varchar vs Char	10
6	Gerenciador de bancos de dados - SQL Server	12
7	Linguagem SQL	17
8	Preparando uma nova tabela	21

1 Arquitetura de Software - MVC

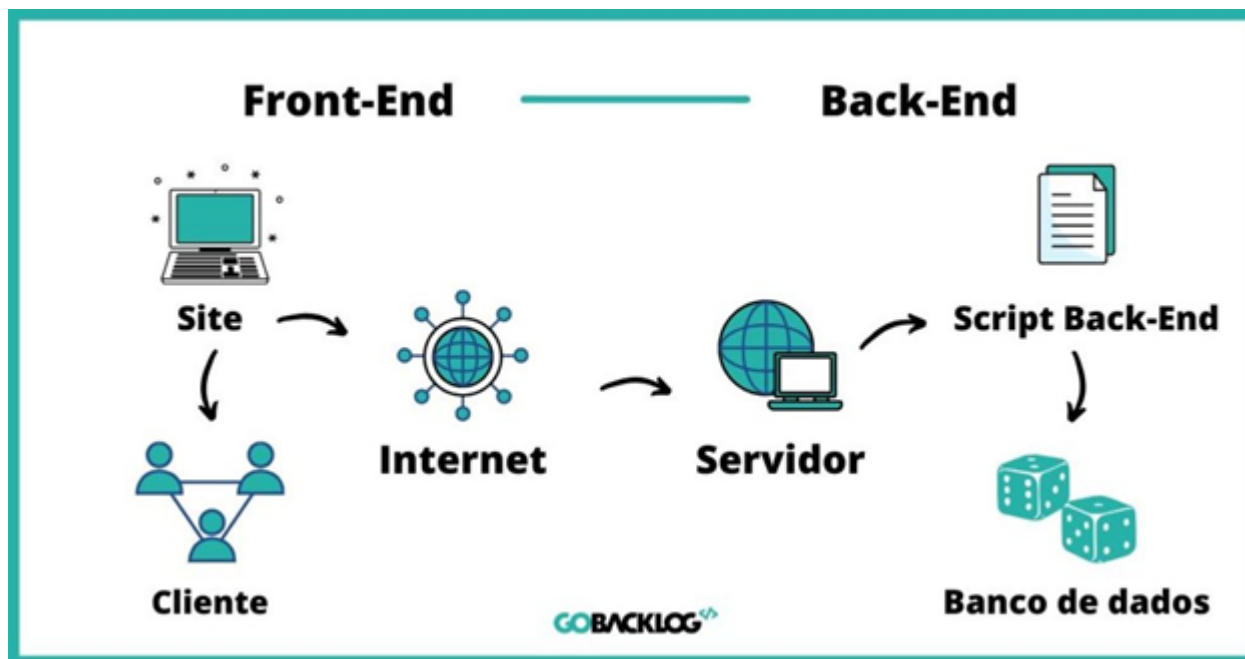
A arquitetura MVC é baseada em três camadas Model (Banco de dados) / View (Páginas Web)/ Controller (Processamento dos dados)



2 O que é um banco de dados?

Bancos de dados ou **bases de dados** são conjuntos de arquivos relacionados entre si com registros sobre pessoas, lugares ou coisas, etc. São coleções organizadas de dados que se relacionam de forma a criar algum sentido (Informação) e dar mais eficiência durante uma pesquisa ou estudo.

Ou seja, o Banco de dados é usada para armazenar informações de um sistema, como no esquema abaixo:



Por exemplo, temos a leitura de um código de barras e esse código está relacionado a um valor na tabela do banco de dados:



Código	Valor
1234567	R\$10,00

3 Banco de dados relacional

Existem diferentes modelos de bancos de dados, sendo o mais comumente utilizado o **modelo relacional**.

Banco de dados relacional: modela de uma forma que podemos acessar os dados em relação a outro dado no banco de dados, normalmente são organizados em tabelas.

Empregado

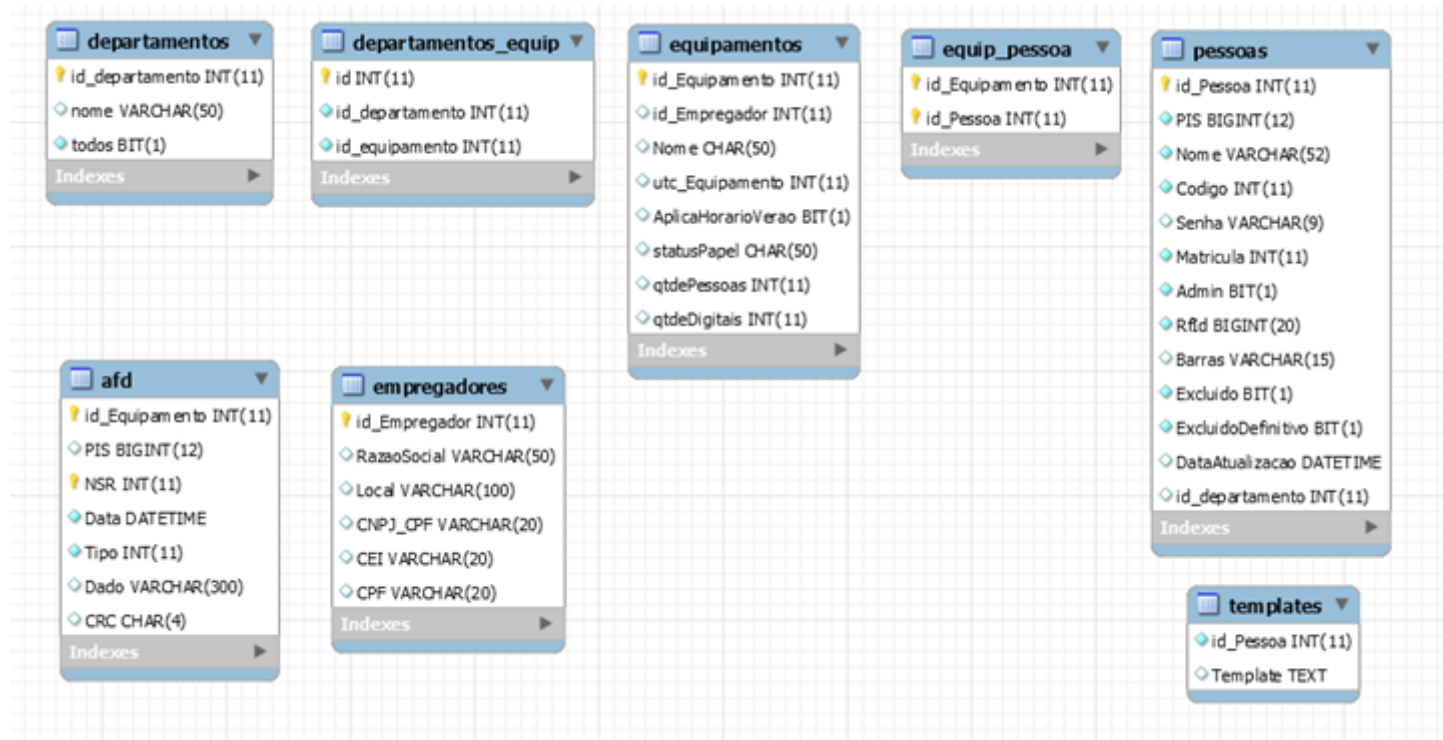
NumEmp	NomeEmp	Salário	Dept
032	J Silva	380	21
074	M Reis	400	25
089	C Melo	520	28
092	R Silva	480	25
112	R Pinto	390	21
121	V Simão	905	28
130	J Neves	640	28

Departamento

NumDept	NomeDept	Ramal
21	Pessoal	142
25	Financeiro	143
28	Técnico	144

Em **um banco de dados** podem existir **várias tabelas**, podendo ser limitadas pelo software utilizado, ou hardware disponível no equipamento.

Por exemplo, no banco de dados de uma loja:



Em uma tabelas podemos ter inúmeras colunas que vão separar os tipos de dados e inúmeras linhas que irão conter os dados específicos:

Results		Messages					
	marca	modelo	ano	cor	valor	portas	combustivel
1	Fiat	Uno	2010	branco	30600.00	2	flex
2	Fiat	Palio	2010	prata	35500.00	2	flex
3	Fiat	Uno	2008	prata	29600.00	4	GNV
4	Fiat	Stilo	2010	verde	42600.00	4	flex
5	Fiat	Uno	2008	preto	44900.00	4	flex
6	Fiat	Punto	2009	branco	25500.00	4	flex
7	Fiat	Uno	2008	branco	23000.00	2	flex
8	Fiat	Palio	2007	prata	26000.00	2	flex
9	Fiat	Uno	2006	verde	34500.00	4	GNV
10	Fiat	Stilo	2008	preto	27000.00	4	flex
11	Fiat	Uno	2007	preto	37000.00	4	gasolina
12	Fiat	Punto	2005	verde	39000.00	2	GNV
13	Fiat	Uno	2008	prata	28000.00	2	gasolina
14	Fiat	Stilo	2007	prata	35000.00	4	flex
15	Fiat	Uno	2006	prata	30000.00	2	flex

As tabelas associam-se entre si por meio de regras de relacionamentos, que consistem em associar um ou vários atributos de uma tabela com um ou vários atributos de outra tabela.

As tabelas relacionam-se umas as outras através de **chaves** que são conjuntos de um ou mais atributos que determinam a unicidade de cada registro (relação entre eles).

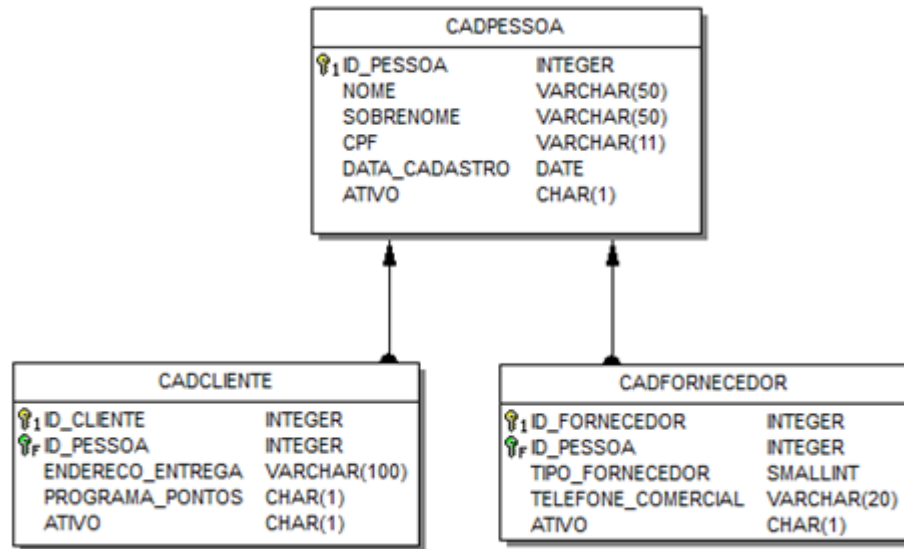


→ **Chave primária:** (PK - Primary Key) é um identificador exclusivo de todas as informações de cada, de acordo com a sua linha na tabela, por exemplo; Dando-lhe unicidade. A chave primária nunca se repetirá.

→ **Chave Estrangeira:** (FK - Foreign Key) é a chave formada através de um relacionamento com a chave primária de outra tabela. Define um relacionamento entre as tabelas e pode ocorrer repetidas vezes.

4 Banco de dados - MER

O **Modelo Entidade Relacionamento (MER)** representa de forma abstrata a estrutura que possuirá o banco de dados de uma aplicação. O MER de um banco de dados deve conter a especificação das colunas de cada tabela (nome e tipo da variável) e o relacionamento de cada tabela através das chaves.



5 Banco de dados - tipos de dados

5.1 Varchar vs Char

Cada caracter no banco de dados é equivalente a 1 byte, pode isso se colocamos um dados “Joao”, estamos consumindo 4 bytes. Se os dados da tabela forem tipados de uma forma ineficiente, vamos aumentar a quantidade de dados que será baixada no banco de dados, aumentando assim o tempo de resposta.

Para strings utilizamos Char ou Varchar.

- Char(6) - o tipo char não é redimensionável.

J	O	A	O		
---	---	---	---	--	--

- Varchar(6) - o tipo varchar irá se redimensionar conforme a quantidade de bytes da palavra.

--	--	--	--	--	--

J	O	A	O
---	---	---	---

- Int - usado para inserirmos números inteiros (o limite é de 11 dígitos)
- Float - usado para inserirmos números reais.
- Null- um dado que não possui nenhum valor.

This is not a NULL Value

StudentID	StudentName	City	marks
1	Raj		45
2	Sham	Mumbai	NULL
3	Tom	Pune	66
4	Ram	Pune	58
5	Joy	Mumbai	NULL
6	Robin	NULL	NULL

6 Gerenciador de bancos de dados - SQL Server

SQL (Structured Query Language) é a linguagem padrão universal para manipular Bancos de Dados relacionais através dos SGBDs. Isso significa que todos os SGBDRs (Sistema de Gerenciamento de Banco de Dados Relacionais) oferecem uma interface para acessar o banco de dados utilizando a linguagem SQL, embora com algumas variações. Logo, saber o que é SQL e como utilizá-la é fundamental para qualquer desenvolvedor de softwares. Utilizaremos para Práticas o Microsoft SQL Server.

MS SQL Server é um Sistema de Gerenciamento de Banco de Dados (SGBD), que utiliza a linguagem SQL.

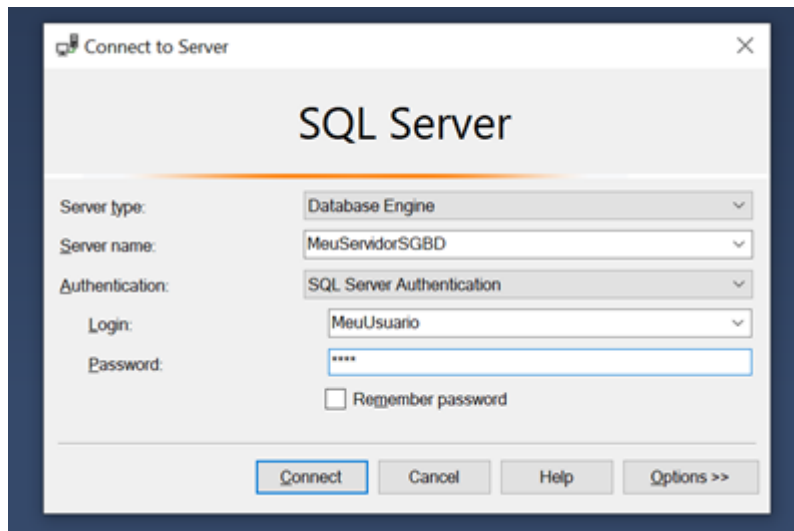


Abra o Microsoft SQL Server Management Studio e crie uma nova conexão de banco de dados com um nome de sua escolha – Não indique senha para o banco de dados. (não coloque nomes iguais para evitar interferência entre os bancos de dados).

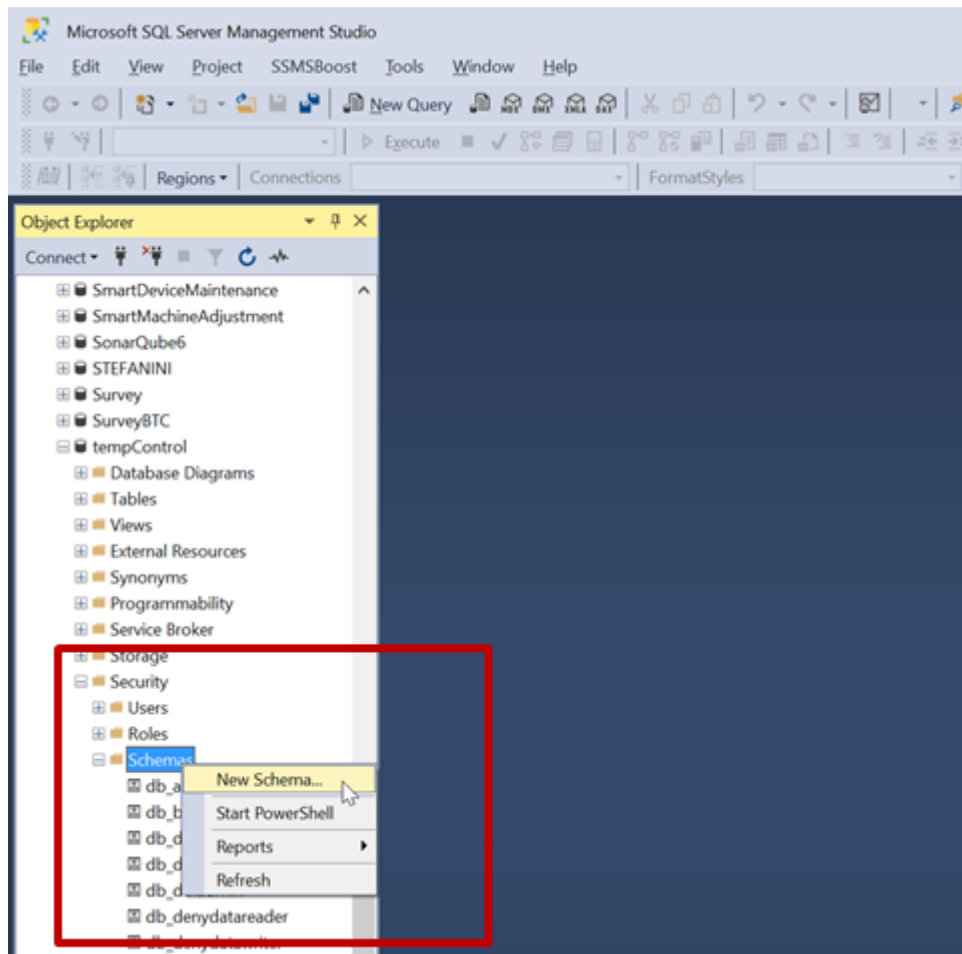


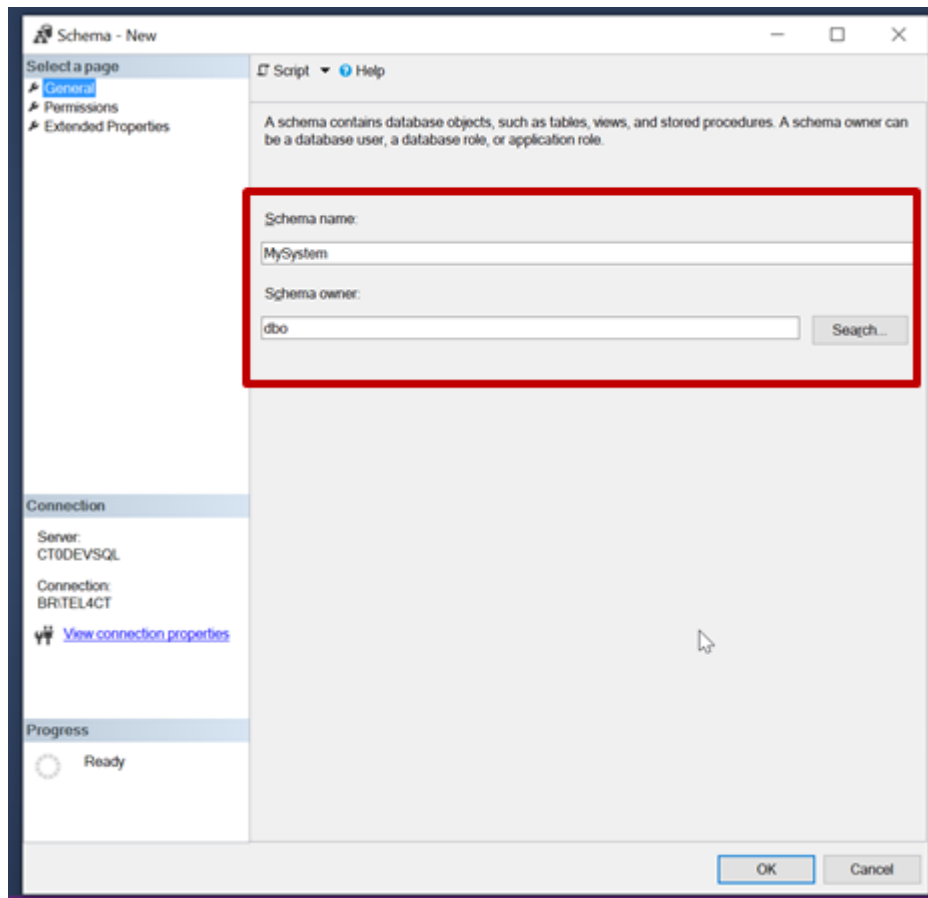
Microsoft SQL Server Management Studio 18

Aplicativo da área de trabalho

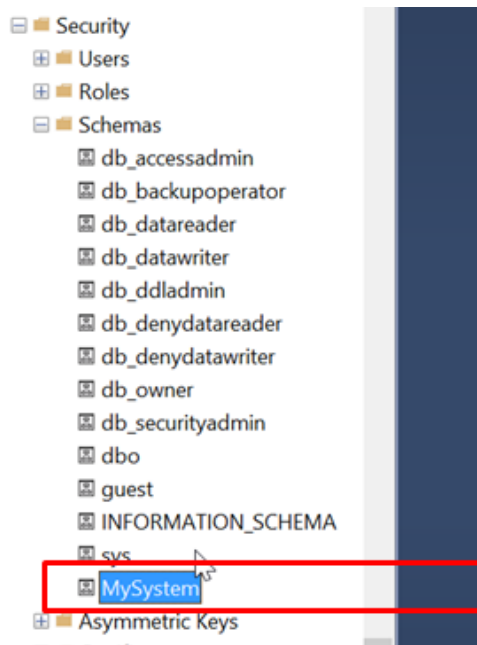


Clique com o botão direito no item Security na barra lateral esquerda, e em “Create Schema”, para criar um novo esquema. (Um Schema basicamente é um conjunto de tabelas). Em geral schemas são utilizados para agrupar tabelas de diferentes sistemas que compartilhem um mesmo banco de dados. Escolha um Nome para o Schema.





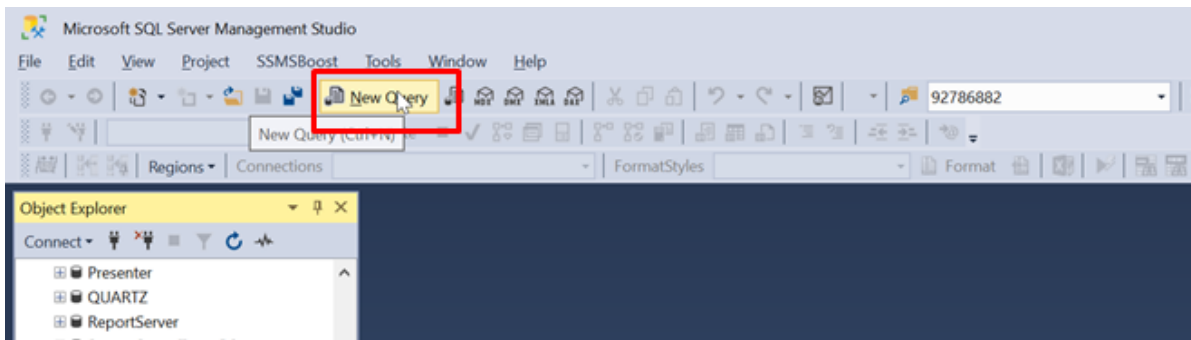
Você pode ver o Schema criado na barra lateral do lado esquerdo.



- As modificações feitas em Schemas podem não aparecer na hora, para isso, sempre que faze-las pressione o “Refresh”.

Agora vamos criar a nossa tabela.

Clique em “New Query” para escrevermos o comando de criação da tabela.



7 Linguagem SQL

USE → Indica qual banco de dados será usado:

USE

```
Use TempControl;
```

CREATE TABLE → Cria uma tabela no banco de dados selecionado:

CREATE TABLE

```
CREATE TABLE Colaboradores (EDV int primary key, Nome varchar(60), Idade int, Salario float);
```

Exemplo:

Colunas criadas

EDV: tipo inteiro, chave primária da tabela;

Nome: tipo string de caracteres (máximo 60 caracteres);

Idade: tipo inteiro;

Salario: tipo float.

INSERT INTO → Insere uma nova linha na tabela do banco de dados:

INSERT INTO

```
INSERT INTO dbo.Colaboradores(EDV, Nome, Idade, Salario) values (1, 'José', 52, 100000.95);  
INSERT INTO dbo.Colaboradores(EDV, Nome, Idade, Salario) values (2, 'Maria', 27, 500000.25);
```

SELECT → Seleciona os dados que escolhidos e apresenta ao usuário:

SELECT

```
SELECT * from dbo.Colaboradores;
```

O * representa a palavra “all”, então ele vai selecionar todos os dados que estão na tabela Turma A. (Por enquanto só adicionamos uma linha na tabela). O retorno seria ao similar à imagem abaixo:

O resultado deve ser o seguinte:

	EDV	Nome	Idade	Salario
1	1	José	52	100000.95
2	2	Maria	27	500000.25

Para selecionar somente uma coluna da tabela:

SELECT

```
SELECT Nome from dbo.Colaboradores;
```

	Nome
1	José
2	Maria

Para filtrar melhor a nossa seleção podemos utilizar o comando **WHERE**. Para utilizar o WHERE é necessário inserir uma condição.

Exemplo: Para Selecionar somente colaboradores com idade maior que 30 anos:

SELECT WHERE

```
SELECT Nome, Idade from dbo.Colaboradores WHERE Idade>30;
```

	Nome	Idade
2	José	52

Podemos ordenar os dados de acordo com os valores de uma coluna específica, basta adicionar o comando **ORDER BY** seguido do nome da coluna escolhida. O resultado retornado será a mesma tabela em ordenada de forma crescente em relação a coluna escolhida. Para ordenar de forma decrescente basta adicionar **DESC** no final do comando.

SELECT ORDER BY

```
SELECT Nome, Salario FROM dbo.Colaboradores ORDER BY Salario;
```

	Nome	Salario
1	Maria	500000.25
2	José	100000.95

UPDATE → Atualiza registros no banco de dados:

UPDATE

```
UPDATE dbo.Colaboradores SET Nome="Paulo" WHERE EDV=1;
```

	EDV	Nome	Idade	Salario
1	1	Paulo	52	100000.95
2	2	Maria	27	500000.25

DELETE → Deleta registros de uma tabela:

DELETE

```
DELETE from dbo.Colaboradores WHERE EDV=1;
```

	EDV	Nome	Idade	Salario
1	2	Maria	27	500000.25



8 Preparando uma nova tabela

Nosso objetivo é salvar dados de sensor no SQL Server. Então o primeiro passo é criar uma nova tabela para poder armazenar esses dados.

Iremos usar um sensor para temperatura e umidade. CURRENT_TIMESTAMP - irá inserir na linha o horário e dia em que essa linha foi criada. IDENTITY - irá criar automaticamente um valor de id que irá aumentar a cada linha criada.

```
Use tempdb; --Nome do banco de dados que você selecionou para o seu login
CREATE TABLE Sensor (id INT PRIMARY KEY IDENTITY, Temperatura FLOAT NULL, Umidade INT NULL, timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP);
```

	ID	TEMPERATURA	UMIDADE	TIMESTAMP
--	----	-------------	---------	-----------