



SQL COMPLETO

1º Exercicio

```
Untitled-1 •
1 A EQUIPE DE MARKETING PRECISA DE FAZER UMA PERSQUISA SOBRE NOMES MAIS COMUNS DE SEUS CLIENTES
2 E PRECISA DO NOME E SOBRENOME DE TODOS OS CLIENTES QUE ESTÃO CADASTRADOS NO SISTEMA.
3
4 first name , last name|
```

Resposta:

```
SELECT FirstName, LastName
FROM person.Person;
```

100 %

	FirstName	LastName
1	Syed	Abbas
2	Catherine	Abel
3	Kim	Abercrombie
4	Kim	Abercrombie
5	Kim	Abercrombie
6	Hazem	Abolrous
7	Sam	Abolrous
8	Humberto	Acevedo
9	Gustavo	Achong
10	Pilar	Ackeman
11	Pilar	Ackeman
12	Aaron	Adams

2º Exercício

```
1  ### DESAFIO 2 ###
2  QUANTOS NOMES SOBRENOMES ÚNICOS TEMOS EM NOSSA TABELA PERSON.PERSON ?
```

Resposta: 1206

The screenshot shows a SQL query window with the following content:

```
SELECT DISTINCT LastName
FROM person.Person;
```

The results pane displays a table with the following data:

	Last Name
1	Abbas
2	Abel
3	Abercrombie
4	Abolrous
5	Acevedo
6	Achong
7	Ackerman
8	Adams
9	Adina
10	Agborile
11	Agcaouli
12	Aguilar

At the bottom of the window, there is a status bar with the following information:

Consulta executada com êxito. | DESKTOP-FD3EPE3\SQLEXPRESS ... | DESKTOP-FD3EPE3\ddaiw ... | AdventureWorks2017 | 00:00:00 | 1.206 linhas

3º Exercício

```
1  -- DESAFIO
2
3  /*a equipe de produção de produtos precisa do nome de todas as peças que pesam mais que 500kg mas não
   mais que 700kg para inspecção*/
4
```

```
-- DESAFIO 2  
Foi pedido pelo marketing um relação de todos os empregados(employees) que são casados  
(single=solteiro, married =casado) e são asalariados(salaried)
```

```
-- DESAFIO 3  
Um usuário chamado Peter Krebs está devendo um pagamento, consiga o email dele para que possamos  
enviar uma cobrança!  
( você vai ter que usar a tabela person.person e depois a tabela person.emailaddress)
```

Resposta

Desafio 1-

The screenshot shows a SQL query window and a results grid. The query is:

```
SELECT name  
FROM Production.Product  
WHERE Weight > 500 and Weight < 700;
```

The results grid displays the following data:

	name
1	HL Road Front Wheel
2	LL Crankset
3	ML Crankset
4	HL Crankset

Desafio 2 -

```

SELECT *
FROM HumanResources.Employee
WHERE MaritalStatus = 'm' AND SalariedFlag = 'true';

```

100 %

Resultados Mensagens

	BusinessEntityID	NationalIDNumber	LoginID	OrganizationNode	OrganizationLevel	JobTitle	BirthDate	MaritalStatus	Gender	HireDate
1	3	509647174	adventure-works\roberto0	0x5AC0	2	Engineering Manager	1974-11-12	M	M	2007
2	5	695256908	adventure-works\gail0	0x5ADA	3	Design Engineer	1952-09-27	M	F	2008
3	6	998320692	adventure-works\jossef0	0x5ADE	3	Design Engineer	1959-03-11	M	M	2008
4	7	134969118	adventure-works\dylan0	0x5AE1	3	Research and Development Manager	1987-02-24	M	M	2005
5	9	658797903	adventure-works\gigi0	0x5AE168	4	Research and Development Engineer	1979-01-21	M	F	2005
6	10	879342154	adventure-works\michael6	0x5AE178	4	Research and Development Manager	1984-11-30	M	M	2005
7	15	56920285	adventure-works\sharon0	0x5AE7	3	Design Engineer	1961-05-02	M	F	2011
8	26	277173473	adventure-works\peter0	0x7AC0	2	Production Control Manager	1982-11-03	M	M	2008
9	212	885059826	adventure-works\peng0	0x7B56	3	Quality Assurance Supervisor	1976-03-18	M	M	2008
10	227	141165819	adventure-works\gary1	0x7C20	2	Facilities Manager	1971-02-18	M	M	2005
11	228	553069203	adventure-works\christian0	0x7C2B	3	Maintenance Supervisor	1976-01-18	M	M	2008

Desafio - 3

```

SELECT *
FROM Person.Person
WHERE FirstName = 'peter' and LastName = 'krebs';

SELECT *
FROM Person.EmailAddress
WHERE BusinessEntityID = 26;

```

100 %

Resultados Mensagens

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Demographics
1	26	EM	0	NULL	Peter	J	Krebs	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.microso...

	BusinessEntityID	EmailAddressID	EmailAddress	rowguid	ModifiedDate
1	26	26	peter0@adventure-works.com	2153CB40-E8B1-43C3-9791-6BFF91E75D77	2008-11-24 00:00:00.000

4º Exercicio

```
/*DESAFIO 1  
eu quero saber quantos produtos temos cadastrados em nossa tabela de produtos  
(production.product)
```

```
|  
DESAFIO 2  
eu quero saber quantos tamanhos de produtos temos cadastrado em nossa tabela  
(production.product) |
```

```
eu quero saber quantos tamanhos de produtos temos cadastrado em nossa tabela  
*/
```

Resposta

Desafio 1-

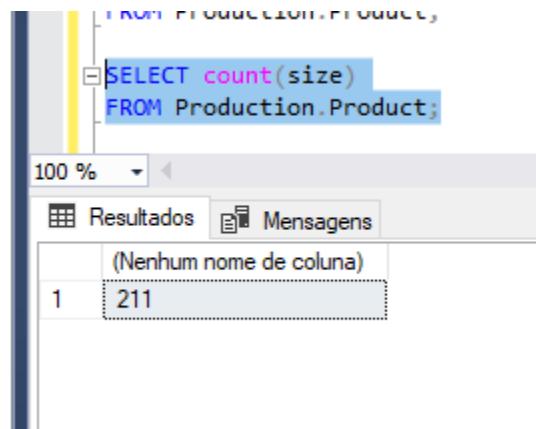
A screenshot of a SQL query editor window. The query entered is:

```
SELECT count (*)
FROM Production.Product;
```

The results pane shows a single row with the value 504.

	(Nenhum nome de coluna)
1	504

Desafio 2-



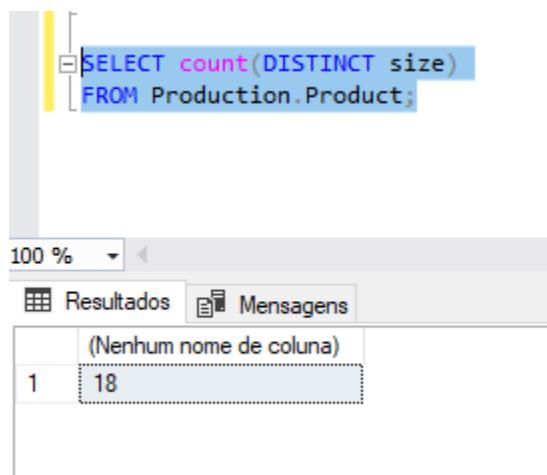
A screenshot of the SQL Server Management Studio interface. The query window contains the following SQL code:

```
FROM Production.Product  
SELECT count(size)  
FROM Production.Product;
```

The results pane shows a single row of data:

	(Nenhum nome de coluna)
1	211

Desafio 3-



A screenshot of the SQL Server Management Studio interface. The query window contains the following SQL code:

```
SELECT count(DISTINCT size)  
FROM Production.Product;
```

The results pane shows a single row of data:

	(Nenhum nome de coluna)
1	18

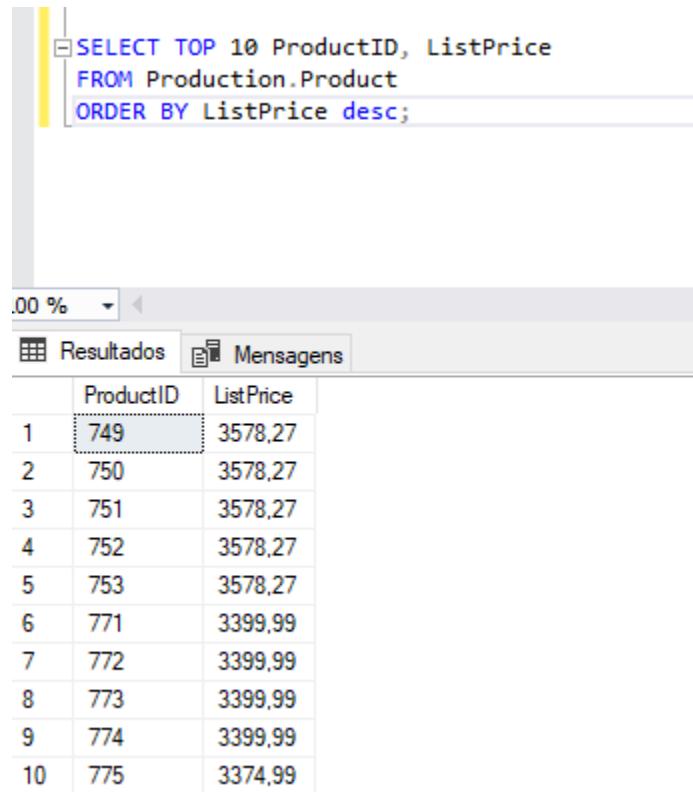
5º Exercicio

```
--DESAFIO 1--  
/*  
Obter o ProductId dos 10 produtos mais caros cadastrados no sistema, listando do  
mais caro para o mais barato  
*/
```

```
/*DESAFIO 2-
Obter o nome e numero do produto dos produtos que tem o ProductID entre 1~4
*/
```

Respostas

Desafio 1-



A screenshot of the SQL Server Management Studio (SSMS) interface. The top pane shows a query window with the following T-SQL code:

```
SELECT TOP 10 ProductID, ListPrice
FROM Production.Product
ORDER BY ListPrice desc;
```

The bottom pane displays the results of the query in a grid format. The grid has two columns: 'ProductID' and 'ListPrice'. The data is as follows:

	ProductID	ListPrice
1	749	3578,27
2	750	3578,27
3	751	3578,27
4	752	3578,27
5	753	3578,27
6	771	3399,99
7	772	3399,99
8	773	3399,99
9	774	3399,99
10	775	3374,99

Desafio 2-

```
SELECT TOP 4 Name, ProductNumber
FROM Production.Product
Order By ProductID asc;
```

	Name	ProductNumber
1	Adjustable Race	AR-5381
2	Bearing Ball	BA-8327
3	BB Ball Bearing	BE-2349
4	Headset Ball Bearings	BE-2908

6º Exercicio

1- Quantos produtos temos cadastrado no sistema que custam mais que 1500 dolares?

2- Quantas pessoas temos com o sobrenome que inicia com a letra P ?

3- Em quantas cidades únicas estão cadastrados nossos clientes ?

4- Quais são as cidades únicas que temos cadastrados em nosso sistema ?

5- Quantos produtos vermelhos tem preço entre 500 a 1000 dolares?

6- Quantos produtos cadastrados tem a palavra 'road' no nome deles ?

Respostas

Desafio 1-

The screenshot shows a SQL query window with the following content:

```
SELECT COUNT(*)
FROM Production.Product
WHERE ListPrice > 1500;
```

The results pane shows the output of the query:

	Resultados
(Nenhum nome de coluna)	
1	39

Desafio 2-

The screenshot shows a SQL query window with the following content:

```
SELECT COUNT(*)
FROM Person.Person
WHERE LastName like 'p%';
```

The results pane shows the output of the query:

	Resultados
(Nenhum nome de coluna)	
1	1187

Desafio 3-

```
SELECT COUNT(DISTINCT CITY)
FROM Person.Address;
```

100 %

Resultados Mensagens

	(Nenhum nome de coluna)
1	575

Desafio 4-

```
SELECT DISTINCT city
FROM Person.Address;
```

100 %

Resultados Mensagens

	city
1	Cheltenham
2	Kingsport
3	Suresnes
4	Baltimore
5	Reading
6	Fontana

Desafio 5-

```
SELECT COUNT(*)
FROM Production.Product
WHERE Color = 'red' and ListPrice BETWEEN 500 AND 1000;
```

100 %

Resultados Mensagens

	(Nenhum nome de coluna)
1	11

Desafio 6-

```
SELECT COUNT(*)
FROM Production.Product
WHERE name like '%road%';
```

0 %

Resultados Mensagens

	(Nenhum nome de coluna)
	103

7º Exercício

1- Eu preciso saber quantas pessoas tem o mesmo MiddleName agrupadas por o MiddleName

2-Eu preciso saber em média qual é a quantidade(quantity) que cada produto é vendido na loja.

3- Eu quero saber qual foram as 10 vendas que no total tiveram os maiores valor de venda(line total) por produto do maior valor para o menor

4-Eu preciso saber quantos produtos e qual é quantidade media de produtos temos cadastrados nas nossas ordem de serviço (WorkOrder), agrupados por productId

RESPOSTAS

```
SELECT middleName, COUNT(middleName) AS "CONTAGEM"
FROM Person.Person
GROUP BY MiddleName;

SELECT ProductID, AVG(OrderQty) AS "MEDIA"
FROM Sales.SalesOrderDetail
GROUP BY ProductID;

SELECT TOP 10 ProductID, SUM(linetotal) AS "TOTAL"
FROM sales.SalesOrderDetail
GROUP BY ProductID
ORDER BY SUM(linetotal) DESC;

SELECT ProductId, COUNT(ProductId) AS "CONTAGEM",
AVG(OrderQty) AS "MEDIA"
FROM Production.WorkOrder
GROUP BY ProductID;
```

8º Exercicio

1. Estamos querendo identificar as províncias(stateProvinceId) com o maior numero de cadastros no nosso sistema, então é preciso encontrar quais províncias (stateProvinceId) estão registradas no banco de dados mais que 1000 vezes

2. Desafio

Sendo que se trata de uma multinacional os gerentes querem saber quais produtos (productId) não estão trazendo em média no mínimo 1 milhão em total de vendas (lineTotal)

Respostas

```
SELECT StateProvinceId, COUNT(StateProvinceId) AS "QUANTIDADE"
FROM Person.Address
GROUP BY StateProvinceId
HAVING COUNT(StateProvinceId) > 1000;

SELECT ProductId, AVG(linetotal) AS "MEDIA"
FROM sales.SalesOrderDetail
GROUP BY ProductID
HAVING AVG(linetotal) < 1000000;
```

9º Exercicio

- ```
1) Encontrar o FirstName e LastName person.person
2) ProductNumber da tabela production.product "Numero do Produto"
3) sales.SalesOrderDetail unitPrice "Preço Unitário"
```

## Respostas

```
SELECT FirstName as "Nome", LastName as "Sobrenome"
FROM Person.Person;

SELECT ProductNumber as "Número do Produto"
FROM Production.Product;

SELECT unitPrice as "Preço Unitário"
FROM Sales.SalesOrderDetail;
```

## 10º Exercicio

INNER JOIN

```
select top 10 *
from person.PhoneNumberType

select top 10 *
from person.PersonPhone

--BusinessEntityId,Name,PhoneNumberTypeId,PhoneNumber
```

```
SELECT TOP 10 *
FROM person.stateprovince

SELECT TOP 10 *
FROM person.address
-- quero AddressId,City,StateProvinceId,Nome do Estado
```

RESPOSTA

Desafio 1-

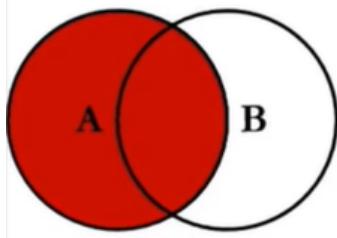
```
SELECT pn.PhoneNumberTypeID, pn.Name, pp.PhoneNumber, pp.BusinessEntityID
FROM Person.PhoneNumberType AS pn
INNER JOIN Person.PersonPhone pp ON pp.PhoneNumberTypeID = pn.PhoneNumberTypeID;
```

Desafio 2-

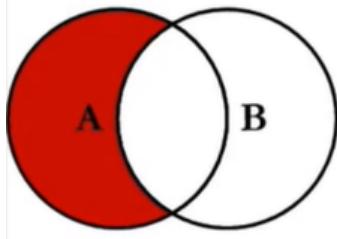
```
SELECT pa.AddressId, pa.city, ps.StateProvinceId, ps.name
FROM person.Address AS pa
INNER JOIN Person.StateProvince ps ON pa.StateProvinceID= ps.StateProvinceID;
```

## SQL JOINS - DIAGRAMA

# SQL JOINS

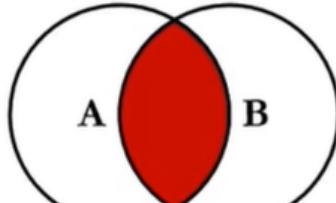


```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

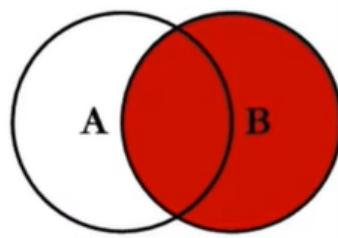


```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

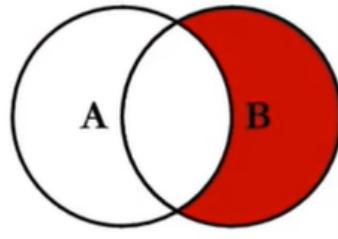
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

© C.L. Moffatt, 2008

## 11º Exercicio

-----DESAFIO-----

-- Encontre para mim todos os endereços que estão no estado de 'Alberta', Pode trazer todas informações  
-- Usar Person.Address e Person.StateProvince

I

Resposta

```
]SELECT AddressLine1
FROM Person.Address
WHERE StateProvinceID IN (
 SELECT StateProvinceID
 FROM Person.StateProvince
 WHERE Name = 'Alberta');

]SELECT pa.AddressLine1
FROM Person.Address pa
INNER JOIN Person.StateProvince ps ON pa.StateProvinceID = ps.StateProvinceID
AND name = 'Alberta';
```

## 12º Exercício

```
-- DESAFIO
-- Eu quero saber na tabela detalhe do pedido [Order Details] quais produtos
-- tem o mesmo percentual de desconto
```

Resposta -

```
]SELECT A.ProductID, A.Discount, B.ProductId, B.Discount
FROM [Order Details] A, [Order Details] B
WHERE A.Discount = B.Discount;
```

## TIPOS DE DADOS

- 1- Booleanos
- 2- Caractere
- 3- Numeros
- 4- Temporais

- 1- Booleanos

Por padrão ele é inicializado como nulo, e pode receber tanto 1 ou 0

## 2- Caracteres

Tamanho fixo - char // permite inserir ate uma quantidade fixa de caractere e sempre ocupa todo espaco reservado 10/50

tamanhos variaveis - varchar ou nvarchar // permite inserir ate uma quantidade que for definida, porem so usa o espaco que for preenchido 10/50

## 3- Numeros

### #### Valores exatos

1. TINYINT - nao tem parte valor fracionados (ex: 1.43, 24.23)
2. SMALLINT - mesma coisa porem limite maior
3. INT - mesma coisa porem limite maior
4. BIGINT - mesma coisa porem limite maior
5. NUMERIC OU DECIMAL - valores exatos, porem permite ter parte fracionados, que tambem pode ser especificado a precisao e escala(escala é o número de digitos na parte fracional) -ex: NUMERIC (5,2) 113,44

### #### Valores Aproximados

1. REAL - Tem precisao aproximada de ate 15 digitos
2. FLOAT - mesmo conceito de REAL

### #### Temporais

1. DATE - armazena data no formato aaaa/mm/dd
2. DATETIME - armazena data e horas no formato aaaa/mm/dd:hh:mm:ss
3. DATETIME2 - data e horas com adição de milisegundos no formato aaaa/mm/dd:hh:mm:ssssss

4. SMALLDATETIME - data e hora nos respeitando o limite entre '1900-01-01:00:00:00' até '2079-06-06:23:59:59'.
5. TIME - horas, minutos, segundos e milissegundos respeitando o limite de '00:00:00.0000000' to '23:59:59.9999999'
6. DATETIMEOFFSET - permite armazena informações de data e horas incluindo o fuso horário.

## Chave Primária e Estrangeira

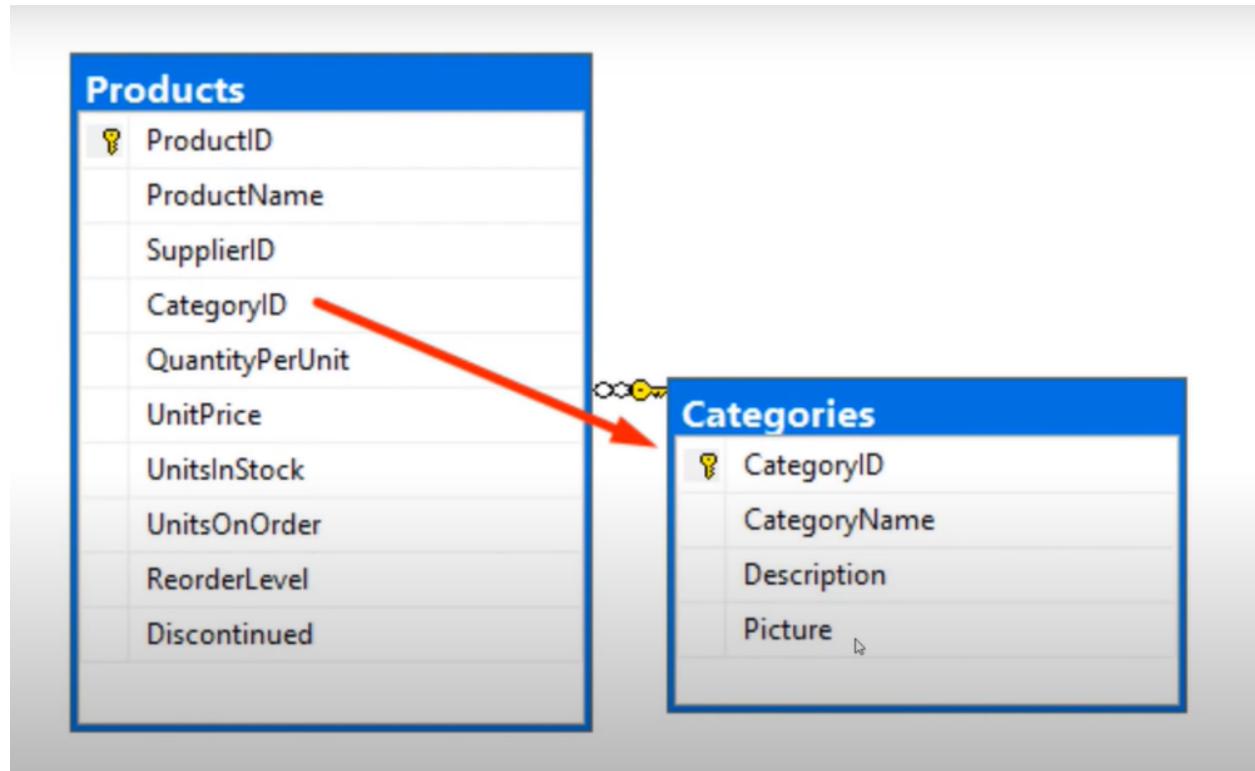
```
CREATE TABLE nome_Tabela (
 nomeColuna tipoDeDados PRIMARY KEY
 nomeColuna tipoDeDados ...
)
```

> o que é uma Chave estrangeira  
\* Uma chave estrangeira é uma coluna ou grupo de colunas em uma tabela que identifica unicamente uma linha em outra tabela.  
  
\* ou seja, uma chave estrangeira é definida em uma tabela onde ela é apenas uma referência e não contém todos os dados ali  
  
\* então uma chave estrangeira é simplesmente uma coluna ou grupo de colunas que é uma chave primária em outra tabela  
  
\* A tabela que contém a chave estrangeira é chamada de tabela referenciadora ou tabela filho. E a tabela na qual a chave estrangeira é referenciada é chamada de tabela referenciada ou tabela pai.  
  
\* uma tabela pode ter mais de uma chave estrangeira dependendo do seu relacionamento com as outras tabelas

\* No SQL Server você define uma chave estrangeira através de um "Foreign Key Constraint" ou Restrição de chave estrangeira

\* Uma Restrição de Chave Estrangeira indica que os valores em uma coluna ou grupo de colunas na tabela filho correspondem aos valores na tabela pai

\* Nos podemos entender que uma chave estrangeira mantém a "integridade referencial "



## CREATE TABLE

```
CREATE TABLE nomeTabela (
 coluna1 tipo restricaoDaColuna,
 coluna2 tipo restricaoDaColuna,
 coluna3 tipo restricaoDaColuna,

);
```

restrições:

Principais tipos de restrições que podem ser aplicadas

NOT NULL - Não permite nulos

UNIQUE - Força que todos os valores em uma coluna sejam diferentes

PRIMARY KEY - uma junção de NOT NULL e UNIQUE

FOREIGN KEY - identifica únicamente uma linha em outra tabela

CHECK - FORÇA uma condição específica em uma coluna

DEFAULT - força um valor padrão quando nenhum valor é passado

YouTube Lite

| Canal               |                   |
|---------------------|-------------------|
| + CanalId           | int PK            |
| + Nome varchar(150) | not null          |
| + ContagemInscritos | int default 0     |
| + DataCriacao       | datetime not null |

| Video               |               |
|---------------------|---------------|
| + Videoid           | int PK        |
| + Nome varchar(150) | not null      |
| + Vizualizações     | int default 0 |
| + Likes             | int default 0 |
| + Dislikes          | int default 0 |
| + Duracao           | int not null  |
| + CanalId           | FK            |



```
SQL AULA 27- CREATE TABLE + DESAFIO

DESAFIO

[Encontre] duas coisas na sua casa que podem tornar tabelas e crie 2 tabelas que tem no mínimo 1
relacionamento com a outra
```

```
CREATE DATABASE Desafio;

USE Desafio;

CREATE TABLE TipoAlimento(
 TipoAlimentoId INT PRIMARY KEY,
 Nome VARCHAR(150) NOT NULL
);

CREATE TABLE Alimentos(
 AlimentoId INT PRIMARY KEY,
 Nome VARCHAR(150) NOT NULL,
 TipoAlimentoId INT FOREIGN KEY REFERENCES TipoAlimento(TipoAlimentoId)
);
```

## INSERT INTO

```
CREATE TABLE Desafio(
 id INT PRIMARY KEY,
 nome VARCHAR(100)
);

INSERT INTO Desafio(id, nome)
VALUES(1, 'linha1');

INSERT INTO Desafio(id, nome)
VALUES
 (2, 'linha2'),
 (3, 'linha3'),
 (4, 'linha4');

CREATE TABLE Desafio2(
 id INT PRIMARY KEY,
 nome VARCHAR(100)
);

INSERT INTO Desafio2(id, nome)
VALUES (1, 'linha1');
```

### Desafio + Coloque sua resposta nos comentários

1. Crie uma tabela nova
2. Insira uma linha de dados nela
3. Insira 3 linhas de dados ao mesmo tempo
4. Crie uma segunda tabela
5. Insira 1 linha nessa tabela nova
6. Copie os dados da segunda tabela para a primeira

## UPDATE

```
/* UPDATE */

UPDATE aula
SET nome = 'newname';

SELECT *
FROM Aula;

UPDATE Aula
SET nome = 'newname1'
WHERE id = 3;

UPDATE Aula
SET nome = 'newname2'
WHERE id = 2;

UPDATE Aula
SET nome = 'newname3'
WHERE id = 1;
```

100 %

Resultados Mensagens

|   | id | nome     |
|---|----|----------|
| 1 | 1  | newname3 |
| 2 | 2  | newname2 |
| 3 | 3  | newname1 |
| 4 | 4  | newname  |

```
#SQL AULA 29 - UPDATE + DESAFIO

Desafio + Poste resposta nos comentários
Altere o nome de 2 linhas diferente e poste
nos comentários o código que você usou para fazer isso!
```

## DELETE

```
/* DELETE */

DELETE FROM Aula
WHERE id = 4;

SELECT *
FROM Aula;
```

## ALTER TABLE

> *Syntaxe*

ALTER TABLE nomeDaTabela

ACAO

> *Exemplos de o que pode ser feito:*

- Add, Remover, or alterar uma coluna
- Set valores padrões para uma coluna
- Add ou Remover restrições de colunas
- Renomear uma tabela

### ### DESAFIO

CRIAR uma tabela nova

| com 3 colunas e depois

- 1) Alterar o tipo de uma coluna!
- 2) Renomear o nome de uma coluna!
- 3) Renomear o nome da tabela que você criou!
- 4) Postar nos comentários o código que você usou para fazer isso!

```

CREATE TABLE Desafio3(
 id INT PRIMARY KEY,
 nome VARCHAR(100),
 email VARCHAR(150)
);

ALTER TABLE Desafio3
 ALTER COLUMN nome VARCHAR(200);
EXEC sp_RENAME 'Desafio3.nome', 'desafioNome', 'COLUMN';
EXEC sp_RENAME 'Desafio3', 'DesafioRename';

SELECT *
FROM DesafioRename;

/* DROP TABLE */

```

100 %

|  | id | desafioNome | email |
|--|----|-------------|-------|
|  |    |             |       |

## DROP TABLE

```

Desafio Crie 2 tabelas e depois exclua elas

* Depois poste o código que usou nos comentários abaixo
* Example drop table person.Address
* Example drop table Production.Culture
* Example drop table Sales.CreditCard

```

```

CREATE TABLE tabela1 (
 id INT PRIMARY KEY,
 nome VARCHAR(200)
);

CREATE TABLE tabela2 (
 id INT PRIMARY KEY,
 nome VARCHAR(200)
);

DROP TABLE tabela1;
DROP TABLE tabela2;

```

## CHECK CONSTRAINT

```
CREATE TABLE CarteiraMotorista (
 Id int NOT NULL,
 Nome varchar(255) NOT NULL ,
 Idade int CHECK (Idade >= 18)
);
```

```
CREATE TABLE MontanhaRussa(
 id INT NOT NULL,
 nome VARCHAR(255) NOT NULL,
 altura INT CHECK (altura >= 150)
);

CREATE TABLE EleicaoVotos(
 id INT NOT NULL,
 nome VARCHAR(255) NOT NULL,
 idade INT CHECK (idade >= 16)
);
```

```
Desafio Crie 2 tabelas novas e crie duas restrições para elas
* Depois poste o código que usou nos comentários abaixo
```

## NOT NULL CONSTRAINT

```
CREATE TABLE CarteiraMotorista (
 Id int NOT NULL,
 Nome varchar(255) NOT NULL ,
 Idade int CHECK (Idade >= 18)
);
```

```
CREATE TABLE MontanhaRussa(
 id INT NOT NULL,
 nome VARCHAR(255) NOT NULL,
 altura INT CHECK (altura >= 150)
);

CREATE TABLE EleicaoVotos(
 id INT NOT NULL,
 nome VARCHAR(255) NOT NULL,
 idade INT CHECK (idade >= 16)
);
```

# Desafio Crie 2 tabelas novas e não permita campos vazios em duas colunas

## UNIQUE

```
/ * UNIQUE */
CREATE TABLE Usuario(
 id INT PRIMARY KEY,
 nome VARCHAR(200) NOT NULL,
 email VARCHAR(200) UNIQUE
);

CREATE TABLE Admin(
 id INT PRIMARY KEY,
 nome VARCHAR(200) NOT NULL,
 chaveacesso VARCHAR(200) UNIQUE
);
```

# Desafio Crie 2 tabelas novas e defina que esses campos precisam ter valores únicos

\* Depois poste o código que usou nos comentários abaixo

## VIEWS

> *Syntaxe*

```
CREATE VIEW [Pessoas Simplificado] AS
SELECT FirstName, MiddleName, LastName
FROM Person.Person
WHERE Title = 'Ms.'
```