## **COMANDOS APRENDIDOS:**

### **CREATE TABLE, AUTO\_INCREMENT(IDENTITY) & INSERT**

CREATE TABLE TabelaA (

id inst. identity(1,1) primary Key,

name varchar(50)

);

CREATE TABLE Tabelab (

id int primary Key identity(1,1),

name varchar(50)

);

insert into TabelaB

values

('Espada'),

('robo'),

('Mario'),

('Samurai')

;

### **DISTINCT**

select distinct person.person.LastName

from person.person;

(seleciona todos os registros desse campo, porém, não repete os que estão duplicados)

### **WHERE**

select \*

from Person.person a

where a.FirstName = 'Peter' and a.LastName = 'Krebs'

select \*

from Person.EmailAddress a

where a.BusinessEntityID = 26

### **COUNT**

select count(\*) /\*mostra quantos registros existem no total da tabela\*/

from person.person

select count(distinct firstname) /\*mostra quantos registros existem com esse campo sem repetir e não nulos\*/

from person.person

select count(firstname) /\*mostra quantos registros existem com esse campo não nulos\*/

from person.person

### **TOP**

select top 10 \*

from person.person

/\*limita para 10 registros que retornarão do select, evitando travar\*/

### **ORDER BY**

select coluna1, coluna2

from tabela

order by coluna 1 asc/desc

/\*vai ordenar na ordem crescente ou decrescente com base em alguma coluna, no caso a 1\*/

/\*pode fazer assim também\*/

select firstname,lastname

from person.person

order by firstname asc, lastname desc

### **TOP E ORDER BY**

select top 10 ProductID

from production.Product a

order by a.ListPrice desc

select top 4 name, ProductNumber

from production.Product a

order by a.ProductID asc

### **BETWEEN**

select \*

from Production.Product a

where a.ListPrice between 1000 and 1500;

select \*

from Production.product a

where a.ListPrice not between 1000 and 1500;

select \*

from HumanResources.Employee a

where a.HireDate between '2009/01/01' and '2010/01/01'

order by a.HireDate;

### **IN**

select \*

from Production.Product a

where a.Color in ('blue', 'red');

/\*será a mesma coisa de:

select \*

from Production.Product a

where a.color = 'blue'

or a.color = 'red';\*/

select \*

from Production.Product a

where a.Color not in ('blue', 'red');

### **LIKE**

select \*

from Production.product a

where a.color like '%ue';

select \*

from Production.product a

where a.color like 'blu%';

select \*

from Production.product a

where a.color like '%lu%';

/\*Aprendeu com o Barretinha em\*/

select \*

from Production.product a

where a.color like '%lu\_';

/\*a diferença desse é que o \_ substitui apenas uma letra\*/

/\*NO SQL SERVER O LIKE NÃO É CASE SENSITIVE\*/

### **SUM, MIN, MAX E AVG**

select top 10 sum(a.LineTotal) as soma

from sales.SalesOrderDetail a

select top 10 min(a.LineTotal) as minimo

from sales.SalesOrderDetail a

select top 10 max(a.LineTotal) as maximo

from sales.SalesOrderDetail a

select top 10 avg(a.LineTotal) as media

from sales.SalesOrderDetail a

### **GROUP BY**

select a.SpecialOfferID, sum(a.UnitPrice) as soma

from sales.SalesOrderDetail a

group by a.SpecialOfferID

/\*soma o unitPrice de cada SpecialOfferId, ex:

soma o unitPrice de todos SpecialOfferId = a 9 em uma linha;

soma o unitPrice de todos SpecialOfferId = a 7 em outra linha...

assim sem misturar a soma de unitPrice de specialofferId diferentes

se não entender roda uma query assim que já vai entender\*/

/\*QUERO SABER QUANTO (em dinheiro) DE CADA PRODUTO FOI VENDIDO ATÉ HOJE\*/

SELECT a.ProductID, sum(a.LineTotal)

FROM Sales.SalesOrderDetail a

group by a.ProductID

/\*QUERO SABER QUANTO (em quantidade) DE CADA PRODUTO FOI VENDIDO ATÉ HOJE\*/

select a.ProductID, count(a.ProductID) as contagem

from Sales.SalesOrderDetail a

group by a.ProductID

/\*QUERO SABER QUANTOS NOMES DE CADA NOME TEMOS EM NOSSO DATABASE\*/

select a.FirstName, count(a.FirstName) as contagemNames

from person.person a

group by a.FirstName

order by contagemNames;

/\*QUERO SABER A MEDIA DE PREÇOS PARA PRODUTOS QUE SÃO PRATAS (SILVER)\*/

select a.Color, avg(a.ListPrice) as mediaPreco

from Production.Product a

where a.color = 'silver'

group by a.color;

### **DESAFIOS**

select a.MiddleName, count(a.MiddleName) as contagem

from Person.Person a

group by a.MiddleName

order by contagem

select a.ProductID, avg(a.OrderQty) as quantidadeMedia

from sales.SalesOrderDetail a

group by a.ProductID

order by quantidadeMedia

select top 10 a.ProductID, sum(a.LineTotal) as maximoValor

from Sales.SalesOrderDetail a

group by a.ProductID

order by maximoValor desc

select \*

from production.WorkOrder

select a.ProductID,

count(a.ProductID) as qtyProducts,

avg(a.OrderQty) as mediaProducts

from production.WorkOrder a

group by a.ProductID

### **HAVING**

/\*o having é aplicado depois que os dados já foram agrupados e o Where é aplicado antes do agrupamento\*/

select a.FirstName, count(a.FirstName) as ocorrencia

from Person.Person a

group by a.FirstName

having count(a.FirstName) > 10

select a.ProductID, sum(a.LineTotal) as "totalVendas"

from Sales.SalesOrderDetail a

group by a.ProductID

having sum(a.LineTotal) between 162000 and 500000;

select a.FirstName, count(a.FirstName) as total

from Person.Person a

where a = 'Mr.'

group by a.FirstName

having count(a.FirstName) > 10

select \*

from person.Address

select a.StateProvinceID, count(a.StateProvinceID)

from person.Address a

group by a.StateProvinceID

having count(a.StateProvinceID) > 1000

select a.ProductID, avg(a.LineTotal) as media

from sales.SalesOrderDetail a

group by a.ProductID

having avg(a.LineTotal) < 1000000

order by media desc

**AS**

/\*quando é nome composto precisa de aspas\*/

select top 10 ListPrice as Preço

from Production.Product

select top 10 ListPrice as "Preço do produto"

from Production.Product

### **INNER JOIN**

Uma imagem contendo Gráfico

Descrição gerada automaticamente

SELECT C.clienteId, C.nome, C.rua, E.cidade

FROM Cliente C

INNER JOIN Endereco E

ON E.enderecoId = C.enderecoId

SELECT P.BusinessEntityID, P.FirstName, P.LastName, E.EmailAddress

FROM Person.Person P

INNER JOIN Person.EmailAddress E

ON E.BusinessEntityID = P.BusinessEntityID

SELECT P.Name, P.ListPrice, Sub.Name as subcatergoryName

FROM Production.Product P

INNER JOIN Production.ProductSubcategory Sub

ON Sub.ProductSubcategoryID = P.ProductSubcategoryID

SELECT TOP 10 \*

from Person.BusinessEntityAddress P

inner join Person.Address Pa

ON Pa.AddressID = P.AddressID

SELECT pp.BusinessEntityID, py.name as PhoneType, pp.PhoneNumberTypeID, pp.PhoneNumber

FROM Person.PersonPhone pp

INNER JOIN Person.PhoneNumberType py

ON pp.PhoneNumberTypeID = py.PhoneNumberTypeID

SELECT pa.AddressID, pa.City, pa.StateProvinceID, ps.Name as nameState

FROM Person. StateProvince ps

INNER JOIN Person.Address pa

ON ps.StateProvinceID = pa.StateProvinceID

### **TYPE OF JOINS**

Diagrama

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente com confiança média

O RIGTH OUTER JOIN funciona da mesma forma, só muda o lado

### **OUTER JOIN**

/\*mostra as pessoas que não tem cartão de crédito cadastrado\*/

select \*

from person.person pp

left join sales.PersonCreditCard sp

on pp.BusinessEntityID = sp.BusinessEntityID

where sp.BusinessEntityID is null

### **UNION**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

\*TEM QUE SER A MESMA QUANTIDADE DE COLUNAS E COLUNAS DO MESMO TIPO

\*ELE REMOVE DADOS DUPLICADOS, A NÃO SER QUE UTILIZE O union all

select p.ProductID, p.Name, p.ProductNumber

from production.product p

where p.name like '%chain%'

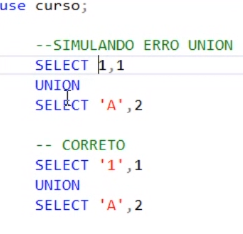
union

select pp.ProductID, pp.name, pp.ProductNumber

from production.product pp

where pp.name like '%decal%'

order by name desc



### **SELF JOIN**

/\*todos os clientes que moram na mesma região\*/

select a.ContactName, a.region, b.ContactName, b.region

from Customers a, customers B

where a.Region = b.Region

order by a.Region

/\*encontrar o nome e a data de contratação de todos os funcionários que foram contratados no mesmo ano\*/

select a.FirstName, a.HireDate, b.FirstName, b.HireDate

from Employees a, Employees b

where datepart(year,a.hiredate) = datepart(year,b.HireDate)

### **SUBQUERIES**

select p.ProductID, p.Name

from Production.Product p

where p.ListPrice > (SELECT AVG(pp.ListPrice)

FROM Production.Product pp

)

select p.FirstName

from person.person p

where p.BusinessEntityID in (

select h.BusinessEntityID

from HumanResources.Employee h

where h.JobTitle = 'design engineer'

)

select \*

from person.Address

select \*

from person.Address pa

where pa.StateProvinceID = (

select ps.StateProvinceID

from person.StateProvince ps

where ps.Name = 'Alberta'

)

### **DATEPART**

/\*Quero saber o mês de um pedido\*/

/\*aqui eu pego o mês e informo em qual coluna essa informação está\*/

SELECT ss.SalesOrderID, DATEPART(month, ss.OrderDate) as mes

FROM sales.SalesOrderHeader ss

SELECT AVG(ss.TotalDue) as media, DATEPART(MONTH, ss.OrderDate) as mes

FROM sales.SalesOrderHeader ss

group by DATEPART(month, ss.OrderDate)

<https://docs.microsoft.com/pt-br/sql/t-sql/functions/datepart-transact-sql?view=sql-server-ver15>

### **STRING**

SELECT CONCAT(pp.FirstName,' ', pp.LastName)

FROM PERSON.PERSON pp

SELECT UPPER(pp.FirstName) as nome, LOWER(pp.LastName) as sobrenome

FROM person.person pp

SELECT pp.firstName, LEN(pp.FirstName) as QuantCaracteres

FROM person.person pp

/\*captura parte do dado, do índice 1 ao 3)\*/

SELECT pp.firstName, SUBSTRING(pp.FirstName, 1, 3) as parteNome

FROM person.person pp

/\*qual campo tem o dado, qual parte do texto vai ser substituida e pelo que vai ser substituida\*/

SELECT pp.ProductNumber, REPLACE(pp.ProductNumber, '-', '%')

FROM Production.Product pp

### **OPERAÇÕES MATEMÁTICAS**

SELECT UnitPrice + LineTotal

FROM sales.SalesOrderDetail

SELECT UnitPrice - LineTotal

FROM sales.SalesOrderDetail

SELECT UnitPrice \* LineTotal

FROM sales.SalesOrderDetail

SELECT UnitPrice / LineTotal

FROM sales.SalesOrderDetail

/\*arredondamento decimal com precisão de 2 casas decimais\*/

SELECT ss.LineTotal, ROUND(LineTotal, 2)

FROM sales.SalesOrderDetail ss

/\*raiz quadrada\*/

SELECT SQRT(ss.LineTotal)

FROM sales.SalesOrderDetail ss

### **TIPOS DE DADOS**

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

### **CHAVE PRIMÁRIA E ESTRANGEIRA**

### **CREATE TABLE**

Texto

Descrição gerada automaticamente

CREATE TABLE Canal (

CanalId int Primary Key,

Nome varchar(150) not null,

ContagemInscritos int default 0,

DataCriacao datetime not null

);

Create TABLE Video (

VideoId int primary key,

Nome varchar(150),

Vizualizacoes int default 0,

Likes int default 0,

Dislikes int default 0,

Duracao int not null,

CanalId int foreign key references Canal(CanalId)

);

### **INSERT INTO**

INSERT INTO Aua(id, nome)

Values(1, 'nome 1')

INSERT INTO Aua(id, nome)

Values

(2, 'nome 2'),

(3, 'nome 3'),

(4, 'nome 4')

/\*criar tabela dando select \*/

SELECT \* INTO TabelaNova FROM Aua

/\*agora a TabelaNova possui todas as informações da tabela Aua\*/

### **UPDATE**

UPDATE Aua

SET nome = 'nome alterado',

id = 10

where id = 1

### **DELETE**

DELETE FROM Aua

WHERE id = 2

### **ALTER TABLE**

Texto

Descrição gerada automaticamente

CREATE TABLE YouTube2(

id int primary key,

nome varchar(200) not null unique,

categoria varchar(200) not null,

dataCriacao datetime not null

);

/\*adicionando uma nova coluna\*/

ALTER TABLE YouTube2

ADD ativo bit

/\*alterando a propriedade de uma coluna\*/

ALTER TABLE YouTube2

ALTER COLUMN categoria varchar(300) not null;

/\*alterar o nome da coluna\*/

/\*informo o nome atual da coluna, o novo nome e o tipo de alteraçõa que é de coluna\*/

EXEC sp\_RENAME 'YouTube2.nome', 'nomeCanal', 'COLUMN'

/\*renomear a tabela\*/

EXEC sp\_rename 'YouTube2', 'youtube3'

### **DROP TABLE**

DROP TABLE youtube3;

### **TRUNCATE TABLE**

/\*apaga todos os dados de uma tabela mas mantém a tabela existindo\*/

TRUNCATE TABLE youtube3;

### **CHECK CONSTRAINT**

/\*cria uma regra dos dados que podem ser inseridos em um campo da tabela\*/

CREATE TABLE cnh(

id int primary key not null,

nome varchar(255) not null,

idade int check ( idade >= 18 )

);

### **VIEWS**

Texto

Descrição gerada automaticamente

create view [pessoas simplificado] as

select a.FirstName, a.MiddleName, a.LastName

from Person.Person a

where Title = 'Ms.'

select \*

from [pessoas simplificado]