**Resumo aulas de SQL DEV APRENDER**

Ícone

Descrição gerada automaticamente

**Sumario**

[1. Primeira Parte (Básico) 4](#_Toc80391223)

[1.1 Criando e Apagando banco de dados 4](#_Toc80391224)

[1.2 Restaurando um banco de dados 4](#_Toc80391225)

[1.3 Comando SELECT 4](#_Toc80391226)

[1.4 Comando DISTINCT 5](#_Toc80391227)

[1.5 Comando WHERE 5](#_Toc80391228)

[1.6 Comando COUNT 6](#_Toc80391229)

[1.7 Comando TOP 6](#_Toc80391230)

[1.8 Comando ORDER BY 6](#_Toc80391231)

[1.9 Comando BETWEEN 7](#_Toc80391232)

[1.10 Comando IN 7](#_Toc80391233)

[1.11 Comando LIKE 8](#_Toc80391234)

[1.12 Desafios para resumo 8](#_Toc80391235)

[2. Segunda Parte (Intermediário) 10](#_Toc80391236)

[2.1 Comandos MIN, MAX, SUM, AVG (funções de agregação) 10](#_Toc80391237)

[2.2 Comando GROUP BY 10](#_Toc80391238)

[2.3 Comando HAVING 11](#_Toc80391239)

[2.4 Comando INNER JOIN 13](#_Toc80391240)

[2.5 Tipos de JOIN 14](#_Toc80391241)

[2.5.1 INNER JOIN 15](#_Toc80391242)

[2.5.2 FULL OUTER JOIN 15](#_Toc80391243)

[2.5.3 LEFT OUTER JOIN 16](#_Toc80391244)

[2.6 Comando UNION 16](#_Toc80391245)

[3. Terceira Parte 18](#_Toc80391246)

[3.1 COMANDO DATEPART 18](#_Toc80391247)

[3.2 Operações com strings 18](#_Toc80391248)

[3.3 Comando SUBQUERY (SUBSELECT) 19](#_Toc80391249)

[3.4 Comando SELF JOIN 20](#_Toc80391250)

[3.5 Tipos de Dados 21](#_Toc80391251)

[3.5.1 Boleanos 21](#_Toc80391252)

[3.5.2 Caracteres 21](#_Toc80391253)

[3.5.3 Números 21](#_Toc80391254)

[3.5.4 Temporais 22](#_Toc80391255)

[3.6 Chave Primária e Estrangeira 22](#_Toc80391256)

[3.7 Comando CREATE TABLE 23](#_Toc80391257)

[3.8 Comando INSER INTO 24](#_Toc80391258)

[3.9 Comando UPDATE 26](#_Toc80391259)

[3.10 Comando DELETE 26](#_Toc80391260)

[3.11 Comando ALTER TABLE 27](#_Toc80391261)

[3.12 Comando DROP TABLE 27](#_Toc80391262)

[4. Referências 27](#_Toc80391263)

# Primeira Parte (Básico)

## Criando e Apagando banco de dados

* *create database name\_databse;* (cria um banco)
* *drop database name\_database;* (remove um banco)

## Restaurando um banco de dados

* É necessário ter um arquivo .bak (backup), no nosso caso está sendo utilizado o “AdventureWorks2017.bak”
* No SSMS clicar com o botão direito no Bancos de Dados e clicar no Restaurar Banco de Dados, depois disso basta localizar o arquivo no dispositivo.

## Comando SELECT

Faz uma query contento a coluna 1 e 2 de uma tabela em específico.

* *SELECT column1, column2*

*FROM table\_name;*

Faz uma query com todas as colunas da tabela.

* *SELECT \**

*FROM table\_name;*

## Desafio 1 ## – A equipe de marketing precisa fazer uma pesquisa sobre nomes mais comuns de seus clientes, dessa forma é necessário obter o nome (first name) e sobrenome (last name) de todos os clients que estão cadastrados no sistema.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Comando DISTINCT

Adicionando o DISTINCT a query descarta conteúdos repetidos.

* *SELECT DISTINCT column1, column2*

*FROM table\_name;*

## Desafio 2 ## Quantos sobrenomes distintos estão presentes na tabela Person.Person?

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

## Comando WHERE

A query é feita, porém utilizando condições matemáticas.

* *SELECT column1, column\_n FROM table\_name WHERE condição;*

|  |  |
| --- | --- |
| **Operador** | **Descrição** |
| = > < >= <= <> AND  OR | Igual  Maior que  Menor que  Maior que ou igual  Menor que ou igual  Diferente de  Operador lógico E  Operador lógico OU |

## Desafio 3 ## Foi pedido pelo marketing uma relação de todos os empregados(employees) que são casados (married), são asalariados (salaried) e tem horas de férias (vacation hours) maior que 20.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

## Comando COUNT

Vai contar o número de características diferentes dentro da column1.

* *SELECT COUNT (DISTINCT column1)*

*FROM table\_name*

## Desafio 4 ## Quantos tamanhos diferentes de produtos estão cadastrados na tabela production.product?

Interface gráfica do usuário, Aplicativo, Tabela

Descrição gerada automaticamente

## Comando TOP

Faz uma query cosiderando apenas as 10 primeiras linhas.

* *SELECT TOP 10 column1, column\_n FROM table\_name*;

## Comando ORDER BY

Vai ordenar os dados de forma descendente ou ascendente.

* *SELECT column1, column\_n*

*FROM table\_name*

*ORDER BY column1 ASC/DESC;*

## Desafio 5 ## Obter o ProductId dos 10 produtos mais caros cadastrados no sistema, listando do mais caro para o mais barato.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

## Comando BETWEEN

Vai fazer uma query com os valores dentro do intervalo estabelecido. Basta adicionar o NOT para pegar os valores fora desse intervalo.

* *SELECT column1, column\_n*

*FROM table\_name*

*WHERE column1 BETWEEN 1000 AND 1500;*

Mesmo princípio, contudo utilizando para datas.

* *SELECT \**

*FROM table\_name*

*WHERE column\_with\_date BETWEEN ‘2009/01/01’ and ‘2010/01/01’*

## Comando IN

* *SELECT \**

*FROM table\_name*

*WHERE column\_with\_info IN (info\_target1, info\_target2, info\_target3);*

Equivale à:

* *SELECT \**

*FROM table\_name*

*WHERE column\_with\_info = info\_target1*

*OR column\_with\_info = info\_target2*

*OR column\_with\_info = info\_target3;*

## Comando LIKE

Utilizado para fazer uma query onde não se lembra exatamente o que se deseja. Seguindo o exemplo temos a string gab e isso vai ser responsável por trazer nomes que contém gab – gabriel, gabriela etc.

* *SELECT \**

*FROM table\_name*

*WHERE FirstName LIKE ‘%gab%’;*

## Desafios para resumo

## Desafio 6 ## Quantos produtos temos cadastrado no sistema que custam que 1500 dolares?

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Desafio 7 ## Quantas pessoas temos com o sobrenome que inicia com a letra P?

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Desafio 8 ## Em quantas cidades únicas estão cadastrados nossos clientes?

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Desafio 9 ## Quais são as cidades únicas que temos cadastrados em nosso sistema?

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Desafio 10 ## Quantos produtos vermelhos tem preço entre 500 a 1000 dolares?

Tabela

Descrição gerada automaticamente com confiança média

# Segunda Parte (Intermediário)

## Comandos MIN, MAX, SUM, AVG (funções de agregação)

Realiza operações com intervalos de dados selecionados. Mínimo, Máximo, Soma e Média.

* *SELECT \* MIN/MAX/SUM/AVG (column\_target) AS “name\_to\_column\_query”*

*FROM table\_name;*

O comando AS serve para dar nome a coluna de resposta

## Comando GROUP BY

Permite realizar operações a partir de colunas e agrupando elas em uma única resposta em função da column1.

* *SELECT column1, agregation\_function (column2)*

*FROM table\_name*

*GROUP BY column1;*

## Desafio 11 ## Preciso saber quantas pessoas tem o mesmo MiddleName agrupadas pelo MiddleName.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## Desafio 12 ## Preciso saber em média qual é a quantidade que cada produto é vendido na loja.

Tabela

Descrição gerada automaticamente

## Comando HAVING

É basicamente um WHERE para dados agrupados, ou seja, serve para filtrar dados vindos de um agrupamento.

* *SELECT column1, agregation\_function (column2)*

*FROM table\_name*

*GROUP BY column1*

*HAVING condiction;*

A diferença entre os dois é que o HAVING é usado depois do agrupamento e o WHERE é usado antes. Exemplo de uso para os dois juntos:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

## Desafio 13 ## Estamos querendo identificar as provincias (StateProvinceId) com maior número de cadastros no nosso sistema, então é preciso encontrar quais províncias (StateProvinceId) estão registradas no banco de dados mais que 1000 vezes.

Tabela

Descrição gerada automaticamente

## Desafio 14 ## 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).

Uma imagem contendo Tabela

Descrição gerada automaticamente

## Comando INNER JOIN

Esse comando serve basicamente para relacionar dados de duas tabelas diferentes, dessa forma permitindo fazer uma query em função de lógicas entre duas colunas de duas tabelas diferentes. Abaixo é mostrada essa relação em que na tabela cliente temos o id do endereço que pode ser usado na tabela endereço.

* *SELECT C.ClientId, C.Nome, E.Rua, E.Cidade*

*FROM Cliente C*

*INNER JOIN Endereço E ON E.EndereçoId = C.EndereçoId*

Exemplo de resposta = |2| Bruno| Rua Norte | São Paulo|

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

## Desafio 15 ## Faça uma query mostrando as seguintes colunas BusinessEntityId, PhoneNumber da tabela (person.PersonPhone) e Name, PhoneNumberTypeId da tabela (person.PhoneNumberType).

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

## Desafio 16 ## Faça uma query mostrando as seguintes colunas: AddressId, City da tabela (person.Address) e StateProvinceId, Name da tabela (person.StateProvince).

Tabela

Descrição gerada automaticamente

## Tipos de JOIN

Diagrama

Descrição gerada automaticamente

### INNER JOIN

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

### FULL OUTER JOIN

Diagrama

Descrição gerada automaticamente

### LEFT OUTER JOIN

Uma imagem contendo Diagrama de Venn

Descrição gerada automaticamente

## Comando UNION

O operador UNION combina dois ou mais resultados de um select em um resultado apenas. Caso haja duplicatas entre os dois ele ignora e traz somente um.

* *SELECT column1, column2*

*FROM table\_1*

*UNION*

*SELECT column1, column2*

*FROM table\_2;*

## Desafio 17 ## Utilize o comando UNION para unir duas queries, de modo que os dois SELECTS devem retornar FirstName, Title, MiddleName da tabela person.Person, contudo o primeiro SELECT devem filtrar para o Title “Mr” e o segundo para o MiddleName “A”.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

# Terceira Parte

## COMANDO DATEPART

Esse tipo de comando serve para trabalhar com datas no banco, basta passar o argumento que se deseja extrair (month, year...) e a coluna que contém esse formato de data.

* *SELECT column1, DATEPART (argument\_datepart, column\_name\_with\_data)*

*FROM table\_name;*

Os argumentos estão presentes no link a seguir: <https://docs.microsoft.com/pt-br/sql/t-sql/functions/datepart-transact-sql?view=sql-server-ver15>

## Desafio 18 ## Mostre a média da divida total por ano (TotalDue) da tabela sales.SalerOrderHeader.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Operações com strings

Concatena duas colunas, por exemplo um nome, espaço vazio, sobrenome.

* *SELECT CONCAT (column1, ‘ ‘, column2)*

*FROM table\_name;*

UPPER deixa todas as letras maiúsculas e o LOWER todas minúsculas.

* *SELECT UPPER (column1), LOWER (column2)*

*FROM table\_name;*

Exibe o tamanho da palavra, quantos caracteres ela possuiu.

* *SELECT LEN (column1)*

*FROM table\_name;*

Faz uma query da coluna um contendo apenas um intervalo entre o primeiro e o terceiro caracter.

* *SELECT SUBSTRING (column1, 1, 3)*

*FROM table\_name;*

Substitui um caracter por outro.

* *SELECT REPLACE (column1, ‘will\_get\_out’, ‘will\_get\_in’)*

*FROM table\_name;*

Para ter acesso a todos os comandos de manipulação de strings, basta acessar o link a seguir: <https://docs.microsoft.com/pt-br/sql/t-sql/functions/string-functions-transact-sql?view=sql-server-ver15>

## Comando SUBQUERY (SUBSELECT)

Serve para fazer um select que depende do valor de um outro select.

* *SELECT \**

*FROM table\_name*

*WHERE column1 > (SELECT AVG (column1) FROM table\_name);*

## Desafio 19 ## Encontre todos os endereços que estão no estado de “Alberta”, contendo todas as informações.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

## Comando SELF JOIN

Para o restante dos estudos foi alterado o banco de dados em que se estava trabalhando, basta copiar todo conteúdo do arquivo “Northwind\_db”, clicar com o botão direito em database no SSMS, nova query, colar e rodar para que um novo banco seja criado. Lembre-se de atualizar.

O SELF JOIN serve para comparar dados de uma mesma tabela, para isso se faz uso do AS para “fazer um clone” da tabela permitindo utilizar métodos de comparação.

* *SELECT column1*

*FROM table\_name AS A, table\_name AS B*

*WHERE condiction;*

## Desafio 20 ## Quero saber na tabela detalhe do pedido quais produtos tem o mesmo percentual de desconto.

Tabela

Descrição gerada automaticamente

## Tipos de Dados

### Boleanos

***BIT*** - Por padrão ele é inicializado como nulo, e pode receber tanot 1 ou 0.

### Caracteres

***Char*** – Tamanho fixo, permite inseir até uma quantidade fixa de caracteres e sempre ocupatodo o espaço reservado.

EXEMPLO: Reservei um espaço de 50 caracteres na memória, porém só escrevi ‘Gabriel’, ele vai ocupar os 50 espaços na memória.

***Vchar ou nvchar*** – permite inserir até uma quantidade que for definida, porém só usa o espaço que for preenchido. Apesar de parecer melhor em relação ao char, não deve ser utilizado sempre no lugar dele porque esse tipo de dado tem alocação dinâmica e pode contribuir para redução de desempenho.

### Números

Valores exatos:

***TINYINT*** – não tem parte do valor fracionados, somente a parte inteira para um número pequenho.

***SMALLINT*** – mesma coisa, porém com o limite maior que o TINYINT

***INT*** - mesma coisa, porém com o limite maior que o SMALLINT

***BIGINT*** - mesma coisa, porém com o limite maior que o INT

***NUMERIC* ou DECIMAL** – valores exatos, porém permite ter parte fracionados, que também pode ser especificado a precisão e escala. Exemplo: o número 133 para um NUMERIC (5,2) fica 133,00.

Valores aproximados:

***REAL*** – Tem precisão aproximada de até 15 dígitos

***FLOAT*** – Mesmo conceito do REAL

### Temporais

***DATE*** – armazena data no formato aaaa/mm/dd

***DATETIME*** – armazena data e horas no formatto aaaa/mm/dd:hh:mm:ss

***DATETIME2*** – data e horas com adição de milissegundos no formato aaaa/mm/dd:hh:mm:ssssss

***SMALLDATETIME*** – data e hora respeitando o limite entre ‘1900-01-01:00:00:00’ até ‘2079-06-06:23:59:59’

***TIME*** – horas, minutos, segundos e milisegundos respeitando o limite de ’00:00:00.0000000’ até ’23:59:59.9999999’

***DATETIMEOFFSET*** – permite armazenar informações de data e horas incluindo o fuso horário.

## Chave Primária e Estrangeira

Uma chave primária é basicamente uma coluna ou grupo de colunas, usada para identificar unicamente uma linha em uma tabela. É possível criar essas chaves primárias através de restrições (ou constrints em inglês), que são regras que você define quando está criando uma coluna.

Feito isso, você está criando um índice único para aquela coluna ou grupo de colunas.

* *CREATE TABLE table\_name (*

*column1 datatype PRIMARY KEY*

*column2 datatype*

*column3 datatype ...*

*)*

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 inde 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.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Comando CREATE TABLE

* *CREATE TABLE table\_name (*

*column1 datatype column\_constraint*

*column2 datatype column\_constraint*

*column3 datatype column\_constraint*

*...*

*);*

A restrição da coluna é um argumento que pode ou não estar presente. As principais restrições estão listadas a seguir:

***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 unicamente uma linha em outra tabela

***CHECK*** – Força uma condição específica em uma coluna

***DEFAULT*** – força um valor padrão quando nenhum é passado

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

## Comando INSER INTO

Esse comando serve para inserir valores em colunas de uma tabela em específico.

* *INSERT INTO table\_name (column1, column2, …)*

*VALUES*

*(value1, value2)*

*(value3, value4)*

*(value5, value6);*

Essa query faz que seja inserida a coluna 2 da tabela B na coluna 1 da tabela A.

* *INSERT INTO table\_nameA (colunm1)*

*SELECT colunm2*

*FROM table\_nameB;*

## Desafio 21 ## (Vou usar o banco de dados do exemplo do Youtube)

* Crie uma tabela nova
* Insira uma linha de dados nela
* Insira 3 linhas de dados ao mesmo tempo
* Crie uma segunda tabela
* Insira 1 linha nessa tabela nova

Texto

Descrição gerada automaticamente

## Comando UPDATE

* *UPDATE table\_name*

*SET*

*column1 = valor1*

*column2 = valor2*

*WHERE condiction;*

É importante especifica a condição, porque caso não aborde esse parâmetro todas as linhas da coluna serão setadas com o valor indicado.

## Desafio 22 ## Altere o nome de dois canais do banco de dados Youtube

Uma imagem contendo Tabela

Descrição gerada automaticamente

## Comando DELETE

Deleta a linha inteira onde o dado da coluna1 é igual a alguma coisa em específico.

* *DELETE FROM table\_name*

*WHERE column1 = “something”;*

## Comando ALTER TABLE

Adicionar uma nova coluna em uma tabela existente:

* *ALTER TABLE table\_name*

*ADD new\_column datatype;*

Alterar propriedades de uma coluna existente em uma tabela:

* *ALTER TABLE table\_name*

*ALTER COLUMN colunm1 new\_datatype new\_constraint;*

Existem mais outras variáveis para esse tipo de comando, basta pesquisar uma situação mais específica.

## Comando DROP TABLE

Serve para excluir uma tabela, porém essa só pode ser excluída caso não tenha nenhuma outra tabela que esteja se referenciando a ela (foreign key).

* *DROP TABLE table\_name;*

Existe também um comando que serve para excluir somente o contéudo da tabela, preservandoa.

* *TRUNCATE TABLE table\_name;*

# Referências

<https://www.youtube.com/watch?v=rX2I7OjLqWE>