

--Criar uma nova base de dados

Banco de Dados > Criar novo Banco de Dados

Nova Consulta > Create database nomedodatabase;

--Remover uma base de dados

Drop database nomedodatabase;

AULA 02 – SELECT

Feito para mostrar na tela coluna(s) de determinada tabela pretendido pelo usuário.

```
>> SELECT coluna1, colunaxTrouble On My Mind (feat. Tyler, The Creator);
```

```
>> FROM nometabela;
```

--Extrair do banco de dados todas as informações que estão nessa(s) coluna(s) da tabela

```
>> SELECT *
```

```
>> FROM nometabela;
```

--Retornar todas as colunas da tabela

--Colocar ponto & virgula (;) no final da linha de comando

DESAFIO: A EQUIPE DE MARKETING PRECISA DE FAZER UMA PESQUISA SOBRE NOMES MAIS COMUNS DE SEUS CLIENTES E PRECISA DO NOME E SOBRENOME DE TODOS OS CLIENTES QUE ESTÃO CADASTRADOS NO SISTEMA.

AULA 03 – DISTINCT

Usado para omitir dados duplicados de uma tabela; retornar apenas dados únicos

```
>> SELECT DISTINCT coluna1, colunax
```

```
>> FROM nometabela;
```

1. DESAFIO: QUANTOS NOMES SOBRENOMES ÚNICOS TEMOS EM NOSSA TABELA PERSON.PERSON

AULA 04- WHERE

Extrair apenas alguns dados/Informações de uma tabela

```
>> SELECT coluna1, colunax
```

```
>> FROM nometabela
```

```
>> WHERE condição;
```

Operadores Lógicos:

= Igual

> Maior que

< Menor que

>= Maior que ou igual

<= Menor que ou igual

<> Diferente de

AND Operador Lógico E

OR Operador Lógico Ou

1. DESAFIO: A EQUIPE DE PRODUÇÃO DE PRODUTOS PRECISA DO NOME DE TODAS AS PEÇAS QUE PESAM MAIS QUE 500KG, MAS NÃO MAIS QUE 700KG PARA INSPEÇÃO. PESO = WEIGHT
2. DESAFIO: FOI PEDIDO PELO MARKETING UMA RELAÇÃO DE TODOS OS EMPREGADOS (EMPLOYEES) QUE SÃO CASADOS (SINGLE = SOLTEIRO, MARRIED = CASADO) E SÃO ASSALARIADOS (SALARIED)
3. DESAFIO: 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.EMAILADDRES)

Aula 05 – COUNT

Retornar o número de linhas que são *True* à condição imposta

```
>> SELECT COUNT (coluna1)
```

```
>> FROM nometabela;
```

```
>> SELECT COUNT (DISTINCT (coluna1)) --DISTINCT usa p não incluir valores nulos da coluna
```

```
>> FROM nometabela
```

1. DESAFIO: EU QUERO SABER QUANTOS PRODUTOS TEMPOS CADASTRADOS EM NOSSA TABELA DE PRODUTOS
2. DESAFIO: EU QUERO SABER QUANTOS TAMANOS DE PRODUTOS TEMOS CADASTRADO EM NOSSA TABELA

Aula 06 – TOP

Filtrar as linhas dos dados retornados de um SELECT

```
>> SELECT TOP (qtdDeLinhas) coluna1
```

```
>> FROM nometabela
```

Aula 07 – ORDER BY

Ordena os resultados em ordem (de)crescente

```
>> SELECT coluna1, coluna2
```

```
>> FROM nometabela
```

```
>> ORDER BY coluna1 asc/desc
```

--Sempre insira no SELECT a coluna da tabela que você irá ordenar no ORDER BY

1. DESAFIO: OBTER O PRODUCTID DOS 10 PRODUTOS MAIS CAROS CADASTRADOS NO SISTEMA, LISTANDO DO MAIS CARO PARA O MAIS BARATO
2. DESAFIO: OBTER O NOME E NUMERO DO PRODUTO DOS PRODUTOS QUE TEM O PRODUCTID ENTRE 1~4

Aula 08 – BETWEEN

Usado junto com o WHERE para encontrar valor entre um valor mínimo e valor máximo

```
>> SELECT coluna1
```

```
>> FROM nometabela
```

```
>> WHERE col/do valor BETWEEN mínimo AND máximo
```

--É um “col/do valor >= mínimo AND <= máximo;” no comando WHERE

--Colocar NOT BETWEEN não inclui os dados entre min AND max

```
>> SELECT coluna1
```

```
>> FROM nometabela
```

```
>> WHERE col/datas BETWEEN ‘AAAA/MM/DD’ AND ‘AAAA/MM/DD’
```

Aula 09 – IN

Usado junto com o WHERE, verifica se um valor corresponde com qualquer valor passado na lista de valores.

```
>> WHERE valor IN (valor1, valor)
```

```
>> WHERE valor IN (SELECT valor FROM nometabela)
```

Aula 10 – LIKE

Usado para buscar algum dado que você não sabe a informação completa, como o nome de uma pessoa

```
>> SELECT *
```

```
>> FROM Person.Person
```

```
>> WHERE FirstName LIKE 'vini%'
```

--Colocando underline ao invés de % no final da procura, limita-se a apenas 1 caractere à direita do procurado

--Input('%ro%') -> Output['Robert','Cameron']

--Input('%ro_') -> Output['Cameron']

DESAFIOS FUNDAMENTOS SQL

1. DESAFIO: QUANTOS PRODUTOS TEMOS CADASTRADO NO SISTEMA QUE CUSTAM MAIS QUE 1500 DOLARES?
2. DESAFIO: QUANTAS PESSOAS TEMOS COM O SOBRENOME QUE INICIA COM A LETRA P?
3. DESAFIO: 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

Aula 12 – MIN MAX SUM AVG

Usado para agregar/combinar dados de uma tabela em um único resultado

```
>> SELECT coluna1 sum(coluna1) AS "nome coluna"
```

```
>> FROM nometabela
```

```
>> SELECT MIN(coluna1)
```

```
FROM nometabela
```

--Usando o AS ele dá um nome para a nova tabela criada para mostrar o resultado

--AVG = Média

Aula 13 – GROUP BY

Usado para dividir o resultado da pesquisa em grupos (calcular soma de itens, contar o número de itens naquele grupo)

```
>> SELECT coluna1, funcaoAgregacao(coluna2)
```

```
>> FROM nometabela
```

```
>> GROUP BY coluna1;
```

--Se eu quiser ver a soma de gols por equipe numa competição por exemplo

1. DESAFIO: EU PRECISO SABER QUANTAS PESSOAS TEM O MESMO MIDDLENAME AGRUPADAS POR MIDDLENAME
2. DESAFIO: EU PRECISO SABER EM MÉDIA QUAL É A QUANTIDADE QUE CADA PRODUTO É VENDIDO NA LOJA
3. DESAFIO: 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. DESAFIO: EU PRECISO SABER QUANTOS PRODUTOS E QUAL A QUANTIDADE MEDIA DE PRODUTOS TEMOS CADASTRADOS NAS NOSSAS ORDEM DE SERVIÇO (WORKORDER), AGRUPADOS POR PRODUCTID

Aula 14 – HAVING

É muito usado em junção com o GROUP BY para filtrar resultados de um agrupamento (um where para dados agrupados)

```
>> SELECT coluna1, funcaoAgregacao(coluna2)
>> FROM nometabela
>> GROUP BY coluna1
>> HAVING condição
```

--A grande diferença entre HAVING e WHERE é que o WHERE é aplicado antes dos dados serem agrupados

1. DESAFIO: ESTAMOS QUERENDO IDENTIFICAR AS PROVINCIAS (STATEPROVINCEID) COM O MAIOR NUMERO DE CADASTROS NO NOSSO SISTEMA, ENTÃO É PRECISO ENCONTRAR QUAIS PROVINCIAS ESTÃO REGISTRADAS NO BANCO DE DADOS MAIS QUE 1000 VEZES
2. DESAFIO: SENDO QUE SE TRATA DE UMA MUTINACIONAL, OS GERENTES QUEREM SABER QUAIS PRODUTOS (PRODUCTID) NÃO ESTÃO TRAZENDO EM MÉDIA NO MÍNIMO 1 MILHÃO EM TOTAL DE VENDAS (LINETOTAL)

Aula 15 – AS

É usado para dar/modificar nomes de uma coluna da tabela

```
>> SELECT coluna1 AS 'novonome'
```

--Se for nome composto precisa declarar como string, senão, não precisa

1. DESAFIO: ENCONTRAR O FIRST NAME E O LAST NAME COM NOME EM PORTUGUES
2. DESAFIO: PRODUCTNUMBER DA TABELA PRODUCTION.PRODUCT PARA NUMERO DO PRODUTO

Aula 16 – INNER JOIN

É usado para unificar informações de outras tabelas.

Numa tabela há *premier key* que é o valor que identifica determinada tabela e *foreign key* são os outros dados

```
>> SELECT C.ClientId, C.Nome, E.Rua, E.Cidade
>> FROM Cliente as C
>> INNER JOIN Endereco as E ON E.Enderecoid = C.Enderecoid
```

--o AS serve para transformar todo o diretório da tabela para uma única variável

Aula 17 – OUTER JOIN

O LEFT OUTER JOIN pode ser escrito apenas como LEFT JOIN...

```
>> SELECT
```

```
--
```

Aula 18 – UNION

Usado para combinar de alguma forma, a grande diferença é que o UNION ajuda a combinar 2 ou mais resultados de um SELECT em um único resultado.

```
>> SELECT coluna1, coluna2
>> FROM tabela1
>> UNION
>> SELECT coluna1, coluna2
>> FROM tabela2
```

--Quantidade de colunas e o tipo de dados tem que ser igual para as uniões

--UNION remove resultados duplicados, para INCLUIR dados duplicados é utilizado o UNION ALL

--Usa-se UNION em tabelas não normalizadas

Aula 19 – DATEPART

Usado para extrair diversas informações de um dado tipo timestamp

```
>> SELECT DATEPART (datepart, colunaData)
```

```
>> FROM nometabela
```

```
>> GROUP BY colunaData
```

--Muito útil utilizado junto com o GROUP BY

datepart	abreviações
year	yy, yyyy
quarter	qq, q
month	mm, m
dayofyear	dy, y
day	dd, d
week	wk, ww
weekday	dw

hour	hh
minute	mi, n
second	ss, s
millisecond	ms
microsecond	mcs
nanosecond	ns
tzoffset	tz
iso_week	isowk, isoww

Aula 20 – MANIPULAÇÃO DE STRING

Usado para

```
>> SELECT ASCII(parâmetros)
```

--

ASCII
CONCAT
FORMAT
LOWER
PATINDEX
REPLICATE
RTRIM
STR
STRING_SPLIT
TRANSLATE
UPPER

CHAR
CONCAT_WS
LEFT
LTRIM
QUOTENAME
REVERSE
SOUNDEX
STRING_AGG
STUFF
TRIM
CHARINDEX

DIFFERENCE
LEN
NCHAR
REPLACE
RIGHT
SPACE
STRING_ESCAPE
SUBSTRING
UNICODE

Aula 21 – OPERAÇÕES MATEMATICAS

```
>> ROUND(nomeColuna, NumDeCasasDecimais)
```

```
>> SQRT(nomeColuna) --Raiz Quadrada
```

Aula 22 – SUBQUERY

SELECT DENTRO DE UM SELECT

```
>> SELECT *
```

```
>> FROM nometabela
```

```
>> WHERE nomecoluna (condição) (SELECT * FROM nometabela WHERE ...)
```

--AVG pode ser qualquer outro, SUM COUNT

1. DESAFIO: EU QUERO TODOS OS ENDEREÇOS QUE ESTÃO NO ESTADO DE 'ALBERTA'

Aula 23 – SELF JOIN

Usado para agrupar/ordenar dados dentro de uma mesma tabela

```
>> SELECT(coluna1)
```

```
>> FROM tabela1 as t1, tabela2 as t2
```

```
>> WHERE condição
```

--Só pode ser usado utilizando o AS

1. DESAFIO: EU QUERO NOME E DATA DE CONTRATAÇÃO DE TODOS OS FUNCIONÁRIOS QUE FORAM CONTRATADOS NO MESMO ANO.
2. DESAFIO: EU QUERO SABER NA TABELA DETALHE DO PEDIDO QUAIS PRODUTOS TEM O MESMO PERCENTUAL DE DESCONTO

Aula 24 – TIPOS DE DADOS

1. Booleanos

Por padrão ele é iniciado como Null, e pode receber tanto 1 ou 0

2. Caractere

- a. *Tamanho Fixo (CHAR)*: Permite inserir até uma quantidade fixa de caracteres e sempre ocupa todo o espaço reservado 10/50
- b. *Tamanhos Variáveis (VARCHAR ou NVARCHAR)*: Permite inserir até uma quantidade que for definida, porém só usa o espaço que for preenchido 10/50

3. Números

a. Valores Exatos

- i. *TINYINT*: Apenas números Inteiros
- ii. *SMALLINT*: Limite > i.
- iii. *INT*: Limite > ii.
- iv. *BIGINT*: Limite > iii.
- v. *NUMERIC ou DECIMAL*: Valores exatos, porém, permitem terem partes fracionados, que também podem ser especificados a precisão e escala. EX: `>> NUMERIC (5,2),`
`OUTPUT 113,44`

b. Valores Aproximados

- i. *REAL*: Tem precisão aproximada de até 15 dígitos
- ii. *FLOAT*: Mesmo conceito de Real

4. Temporais

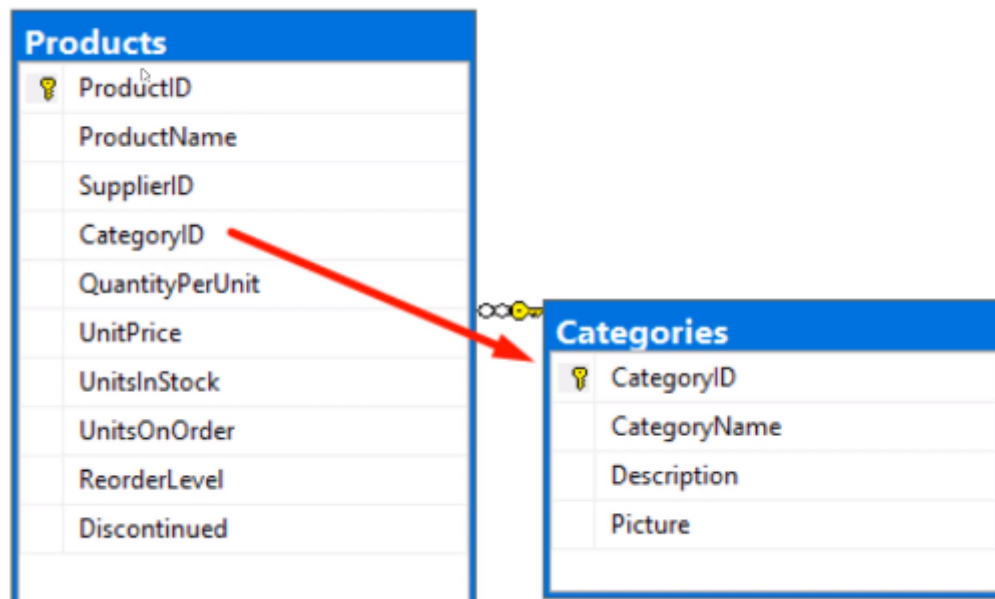
- a. *DATE*: aaaa/mm/dd
- b. *DATETIME*: aaaa/mm/dd:hh:mm:ss
- c. *DATETIME2*: aaaa/mm/dd:hh:mm:ssssss
- d. *SMALLDATETIME*: Limite entre '1900-01-01:00:00:00' & '2079-06-06:23:59:59'
- e. *TIME*: Limite entre '00:00:00.0000000' & '23:59:59.9999999
- f. *DATETIMEOFFSET*: Permite armazenar informações de data e horas incluindo o fuso horário

Aula 25 – CHAVES PRIMÁRIAS E ESTRANGEIRAS

- Uma chave primária é uma coluna ou grupo delas, usada para identificar unicamente uma linha em uma tabela (nunca existirá linha repetida com os dados que estão naquela linha)
- Consegue-se criar essas chaves primárias através de restrições (*constraints*) que são regras que você define quando está criando uma coluna
- Assim, quando você o faz, você está criando um índice único para aquela coluna ou grupo de colunas.

```
>> CREATE TABLE nometabela (  
>>     nomeColuna tipoDeDados PRIMARY KEY  
>>     nomeColuna tipodeDados ...  
>> )
```

- Uma chave estrangeira é definida em uma tabela onde ela é apenas uma referencia e não contem todos os dados ali. Dito isso, é simplesmente uma coluna ou grupo de colunas que é uma FOREIGN KEY 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 outras tabelas.
- Define-se uma chave estrangeira através de uma “FOREIGN KEY CONSTRAINT”
- 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
- Podemos entender que uma chave estrangeira mantém a “integridade referencial”



Aula 26 – CREATE TABLE

Criar tabelas.

```
>> CREATE TABLE nometabela (  
>> nomecoluna1 tipo restricaoDaColuna  
>> nomecoluna2 tipo restricaoDaColuna  
>> ... )
```

--Restrição da Coluna não é obrigatória

PRINCIPAIS RESTRIÇÕES	
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 CODIÇÃO ESPECIFICA 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

1. DESAFIO: EU QUERO NOME E DATA DE CONTRATAÇÃO DE TODOS OS FUNCIONÁRIOS QUE FORAM CONTRATADOS NO MESMO ANO

Aula 26 – INSERT INTO

Usado para inserir dados numa tabela

```
>> INSERT INTO nomeTabela(coluna1, coluna2,...)
>> VALUES (valor1, valor2, ...)
>> (valor1, valor2, ...)
>> [...]
```

--Qtd de dados nos parênteses devem ser iguais a qtd de colunas

```
>> INSERT INTO TabelaA(coluna1)
>> SELECT  coluna2
>> FROM TabelaB
--Para copiar dados de outra coluna de outra tabela
>> SELECT *
>> INTO tabelNova
>>FROM Escolas
```

Aula 26 – UPDATE

Usado para alterar linhas de uma tabela

```
>> UPDATE nomeTabela
>> SET coluna1 = valor1
>> coluna2 = valor2
>> WHERE condicao
--Se você não quer alterar todos os dados da coluna, utilize o WHERE
```

Aula 26 – DELETE

Usado para deletar linhas de uma tabela

```
>> DELETE FROM nomeTabela
>> WHERE condicao
--Se você não quer deletar todos os dados da coluna, utilize o WHERE
```

Aula 26 – ALTER TABLE

Usado para alterar as estruturas de uma tabela

```
>> ALTER TABLE nomeTabela
>> ACAO
```

Add, remover ou alterar uma coluna
Set valores padrões para uma coluna
Add ou remover restrições de colunas
Renomear uma tabela

--bit é um arquivo booleano no SQL

```
>>EXEC sp_rename 'nometabelaatual.nomecolunaatual', 'nomecolunanova'
>>EXEC sp_rename 'nometabelaatual', 'nometabelanova'
--Para renomear uma tabela/coluna
```

```
>>ALTER TABLE FilhosdeFlavio
>>ALTER COLUMN Nome varchar (100) not null
--Para alterar o tipo do dado na coluna
```