

Instituto Federal
Campus Goiânia

Bacharelado em Sistemas de Informação

Banco de Dados II



Prof. Dory Gonzaga Rodrigues





Agenda

- Realizando Cálculos no comando Select
- Utilizando Funções no comando Select
 - Retorno de valores Numérico
 - Manipulação de Datas
 - Manipulação de Caracteres (Alfanuméricas)
- Funções de Grupo
- Conversão de Tipos de Dados
- Agrupamento

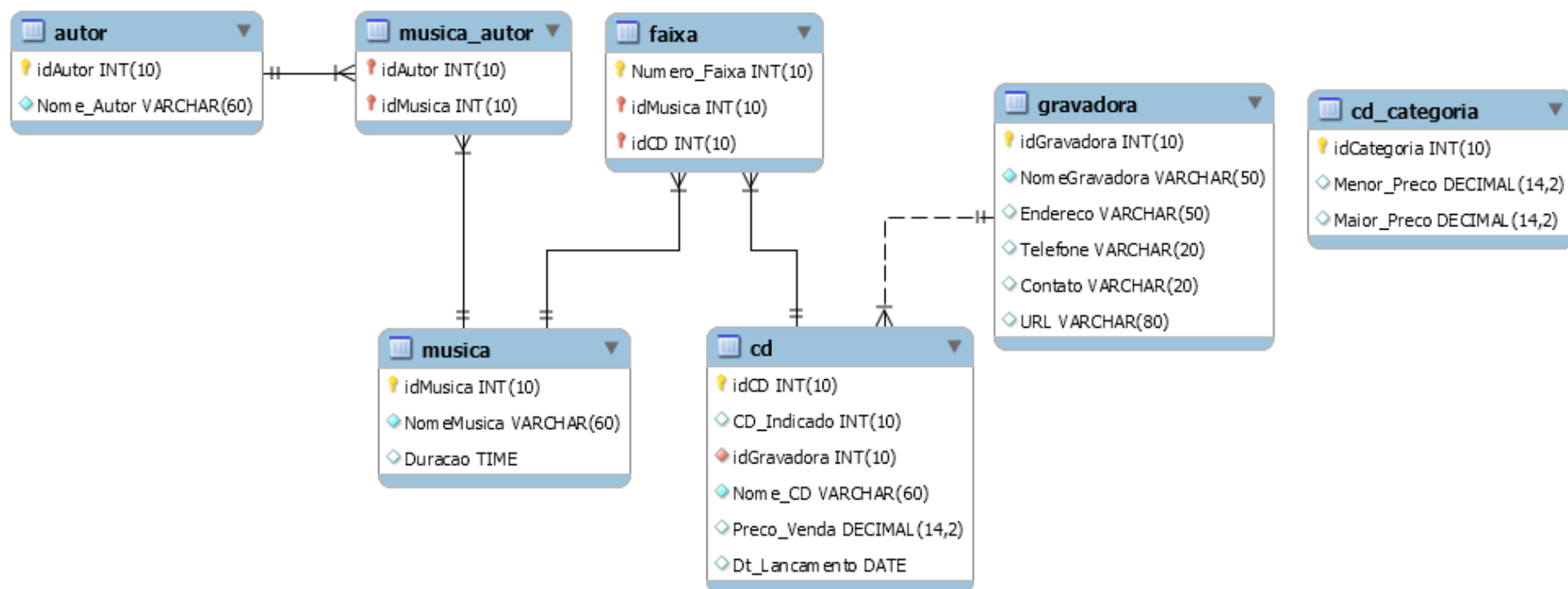




DML BANCO DE DADOS

BANCO DE DADOS REFERÊNCIA

- Utilizaremos nesta aula o banco de dados: Catálogo de CDs



Arquivo enviado por e-mail: [DDL_Catalogo_CDs.sql](#)
[DML_Catalogo_CDs.sql](#)





DML SELECT

SQL – DML

- Manipulação dos dados em Tabela: **SELECT**

O comando **SELECT** é sem sombra de dúvidas o comando mais utilizado e por isso o mais importante da linguagem SQL. Este comando permite selecionar os dados armazenados no Banco de Dados.

A sintaxe:

```
SELECT      [ DISTINCT | ALL ]      { * , coluna1, coluna1, ..., colunaN }  
FROM       nome da tabela  
[ ORDER BY <campo>      ]  
[ WHERE     <condição>  ]  
[ GROUP BY <campo>      ]  
[ HAVING    <condição>  ]
```

[] Opcional





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Cálculos e Funções

Através do comando SELECT podemos fazer cálculos e totalizações de valores unitários. Esta funcionalidade é bastante utilizada e importante para os sistemas informatizados.

- Os cálculos são feitos utilizando operadores Aritméticos e as colunas das tabelas que fazem parte da busca.
- Os operadores aritméticos dentro do SQL Padrão são:

Operador	Ação
+	Soma
-	Subtração
*	Multiplicação
/	Divisão





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Operações Aritméticas

São fornecidos operadores matemáticos para muitos tipos de dado do PostgreSQL.

Operador	Descrição	Exemplo	Resultado
+	adição	2 + 3	5
-	subtração	2 - 3	-1
*	multiplicação	2 * 3	6
/	divisão (divisão inteira trunca o resultado)	4 / 2	2
%	módulo (resto)	5 % 4	1
^	exponenciação	2.0 ^ 3.0	8
/	raiz quadrada	/ 25.0	5
/	raiz cúbica	/ 27.0	3





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Operações Aritméticas

São fornecidos operadores matemáticos para muitos tipos de dado do PostgreSQL.

Operador	Descrição	Exemplo	Resultado
!	fatorial	5 !	120
!!	fatorial (operador de prefixo)	!! 5	120
@	valor absoluto	@ -5.0	5
&	AND bit a bit	91 & 15	11
	OR bit a bit	32 3	35
#	XOR bit a bit	17 # 5	20
~	NOT bit a bit	~1	-2
<<	deslocamento à esquerda bit a bit	1 << 4	16
>>	deslocamento à direita bit a bit	8 >> 2	2

Fonte: <http://pgdocptbr.sourceforge.net/pg80/functions-math.html>





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Cálculos

Exemplos

- Selecionar o nome, preço de venda , o imposto sobre a venda (10%), custo da mercadoria vendida (45%), despesas variáveis de (15%), despesas fixas (20%) e o lucro de cada um dos os CDs.





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Cálculos

Exemplos

- Selecionar o nome, preço de venda , o imposto sobre a venda (10%), custo da mercadoria vendida (45%), despesas variáveis de (15%), despesas fixas (20%) e o lucro de cada um dos os CDs.

Resp: **SELECT** Nome_CD,
Preco_Venda,
Preco_Venda * 0.45 **AS** Custo,
Preco_Venda * 0.10 **AS** Imposto,
Preco_Venda * 0.15 **AS** Despesas_Var,
Preco_Venda * 0.20 **AS** Despesas_Fixas,
Preco_Venda * 0.10 **AS** Lucro
FROM CD;





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Operadores e Funções de Cadeias de Caracteres

Funções e operadores para cadeias de caracteres. No PostgreSQL, cadeia de caracteres inclui todos os valores dos tipos `character`, `character varying` e `text`.

- Número de caracteres presentes na cadeia_de_caracteres.

`length(cadeia_de_caracteres)`

- Converte as letras da cadeia de caracteres em maiúsculas

`upper(cadeia_de_caracteres)`

- Extrai a subcadeia de caracteres

`substr(cadeia_de_caracteres, origem [, tamanho])`

- Concatenação de cadeias de caracteres

`cadeia_de_caracteres || cadeia_de_caracteres`

Fonte: <http://pgdocptbr.sourceforge.net/pg80/functions-string.html>





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **CARACTERES**

- Selecione todos os autores retornando a quantidade de caracteres contidos em cada nome de autor.





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **CARACTERES**

- Selecione todos os autores retornando a quantidade de caracteres contidos em cada nome de autor.

Resp: **SELECT** Nome_Autor, **LENGTH**(Nome_Autor)
FROM autor;

- Selecione todos os autores retornando o nome com todas as letras maiúsculas.





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **CARACTERES**

- Selecione todos os autores retornando a quantidade de caracteres contidos em cada nome de autor.

Resp: **SELECT** Nome_Autor, **LENGTH**(Nome_Autor)
FROM autor;

- Selecione todos os autores retornando o nome com todas as letras maiúsculas.

Resp: **SELECT UPPER**(Nome_Autor)
FROM autor;





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **CARACTERES**

- Selecione todos os autores retornando a abreviação (3 primeiros caracteres dos nomes) e o nome de autor.





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **CARACTERES**

- Selecione todos os autores retornando a abreviação (3 primeiros caracteres dos nomes) e o nome de autor.

Resp: **SELECT SUBSTR**(Nome_Autor, 1, 3) **AS** Abrev, Nome_Autor
FROM autor;

- Concatenação entre o nome e sobrenome do autor

Resp: **SELECT** Nome_Autor || ' ' || SobreNome_Autor
FROM autor;





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Operadores com Date/Time

Trabalhando com **Data / Time**

Operador	Exemplo	Resultado
+	date '2001-09-28' + integer '7'	date '2001-10-05'
+	interval '1 day' + interval '1 hour'	interval '1 day 01:00'
+	time '01:00' + interval '3 hours'	time '04:00'
-	date '2001-10-01' - date '2001-09-28'	integer '3'
-	date '2001-10-01' - integer '7'	date '2001-09-24'
*	interval '1 hour' * double precision '3.5'	interval '03:30'
/	interval '1 hour' / double precision '1.5'	interval '00:40'

Fonte: <http://pgdocptbr.sourceforge.net/pg80/functions-datetime.html>





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Operadores e Funções de Data/Time

Trabalhando com **Data / Time**

Funções para os valores dos tipos **DATE** e **TIME**.

- Data de hoje.

current_date

- Hora do dia

current_time

- Data e Hora corrente

now()

- Subtrai das datas o intervalo de anos e meses

age(timestamp, timestamp)





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **DATAS**

- Selecionar o nome, a data do lançamento e uma data de inicio da venda que deverá ser 31 dias depois do lançamento dos CD's.





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **DATAS**

- Selecionar o nome, a data do lançamento e uma data de inicio da venda que deverá ser 31 dias depois do lançamento dos CD's.
- Resp: **SELECT** nome_cd, dt_lancamento, dt_lancamento + 31 **AS** Ini_Vendas **FROM** cd;
- Selecionar o nome, a data do lançamento e quantos dias já se passaram do lançamento dos CD's até a data de hoje (CURRENT_DATE).





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com DATAS

- Selecionar o nome, a data do lançamento e uma data de inicio da venda que deverá ser 31 dias depois do lançamento dos CD's.

Resp: **SELECT** nome_cd, dt_lancamento, dt_lancamento + 31 **AS** Ini_Vendas
FROM cd;

- Selecionar o nome, a data do lançamento e quantos dias já se passaram do lançamento dos CD's até a data de hoje (CURRENT_DATE).

Resp: **SELECT** nome_cd,
current_date,
dt_lancamento,
current_date - dt_lancamento **AS** Dias
FROM cd;





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **DATAS**

- Selecionar o nome e o ano de lançamento e quantos anos já se passaram desde o lançamento dos CD's.





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções

Trabalhando com **DATAS**

- Selecionar o nome e o ano de lançamento e quantos anos já se passaram desde o lançamento dos CD's.

Resp:

```
SELECT nome_cd, dt_lancamento,  
       date_trunc('year', current_date) – date_trunc('year', dt_lancamento) AS anos  
FROM cd;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

- Funções de Grupo operam conjuntos de linhas visando a fornecer um resultado para o grupo;
- Quando dizemos Funções de Grupo significa que a função trabalha utilizando grupos de linhas. Esses grupos podem ser constituídos de subgrupos da tabela até a tabela toda.
- Existem diversas funções de grupo, as principais são:
 - COUNT()
 - SUM()
 - AVG()
 - MIN()
 - MAX()

Fonte: <http://pgdocptbr.sourceforge.net/pg80/functions-aggregate.html>





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **COUNT**

- Retorna o número de linhas que atende a uma determinada condição. O operador especial * pode ser utilizado para que seja contadas as linhas independente de haver ou não linhas nulas.

EXEMPO

- Selecione a quantidade de autores.





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **COUNT**

- Retorna o número de linhas que atende a uma determinada condição. O operador especial * pode ser utilizado para que seja contadas as linhas independente de haver ou não linhas nulas (NULL).

EXEMPO

- Selecione a quantidade de autores.

```
SELECT COUNT(*)  
FROM autor;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **COUNT**

- Para retornar o número de linhas que atende a uma determinada condição, e que seja contadas as linhas que não estejam nulas: deve-se informar a coluna.

EXEMPO

- Selecione a quantidade de gravadoras com endereços cadastrados.

```
SELECT COUNT(endereco)  
FROM gravadora;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **COUNT com a cláusula DISTINCT**

- Para retornar o número de linhas que atende a uma determinada condição, e que seja contadas as linhas sem repetições devemos utilizar a cláusula DISTINCT na coluna.

EXEMPO

- Selecione a quantidade de músicas distintas que possuem 1 ou mais autores.

```
SELECT COUNT( DISTINCT idMusica)  
FROM musica_autor;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **SUM**

- Para retornar o valor total de uma determinada coluna em um determinado grupo de linhas que atenda às condições.

EXEMPO

- Selecione o valor total dos preços de venda dos CD's.

```
SELECT SUM( preco_Venda)  
FROM cd;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **AVG**

- Para retornar a média aritmética de uma determinada coluna em um determinado grupo de linhas que atenda às condições.

EXEMPO

- O preço médio de venda dos CD's.

```
SELECT AVG( preco_Venda)  
FROM cd;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **MIN**

- Para retornar o menor valor de uma determinada coluna em um determinado grupo de linhas que atenda às condições. Podemos utilizá-la para colunas do tipo Data ou Alfanuméricas.

EXEMPO

- O preço do CD mais barato.

```
SELECT MIN( preco_Venda)  
FROM cd;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **MIN**

- Para retornar o menor valor de uma determinada coluna em um determinado grupo de linhas que atenda às condições. Podemos utilizá-la para colunas do tipo Data ou Alfanuméricas.

EXEMPO

O preço do CD mais barato.

```
SELECT MIN( preco_Venda )  
FROM cd;
```

O CD mais antigo.

```
SELECT MIN( dt_Lancamento )  
FROM cd;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Funções de Grupo (Totais)

Função: **MAX**

- Para retornar o maior valor de uma determinada coluna em um determinado grupo de linhas que atenda às condições. Podemos utilizá-la para colunas do tipo Data ou Alfanuméricas.

EXEMPO

O preço do CD mais caro.

```
SELECT MAX( preco_Venda )  
FROM cd;
```

O CD mais novo.

```
SELECT MAX( dt_Lancamento )  
FROM cd;
```





DML SELECT CÁLCULOS E FUNÇÕES

SQL – DML – Conversão de Tipos de Dados

Função: **CAST**

- Utilizado para realizar uma conversão entre tipos de dados para determinadas situações.
- O preço da Venda dos CD's foi definido com domínio DECIMAL(14,2). Quando calculamos a média dos preços, o sistema nos retorna um número com diversas casas decimais. Neste caso, para manter o padrão dos preços com duas casas utilizamos a função CAST com a formatação desejada.

```
SELECT CAST ( AVG( preco_Venda) AS DECIMAL(14,2) )  
FROM cd;
```





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

- Uma importante característica do comando SELECT é o poder de agrupar linhas com base em valores de determinadas colunas
- A cláusula **GROUP BY** deve vir depois do WHERE e antes do ORDER BY (caso existam).

EXEMPLOS

Quantas músicas existem em cada CD ?





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

- Uma importante característica do comando SELECT é o poder de agrupar linhas com base em valores de determinadas colunas
- A cláusula **GROUP BY** deve vir depois do WHERE e antes do ORDER BY (caso existam).

EXEMPLOS

Quantas músicas existem em cada CD ?

```
SELECT idCD, COUNT(*)  
FROM faixa  
GROUP BY idCD;
```





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

GROUP BY

EXEMPLOS

Qual o preço médio de venda de cada CD por Gravadora ?

Qual o preço médio de vendas e a quantidade de CD's de cada Gravadora ?





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

GROUP BY

EXEMPLOS

Qual o preço médio de venda de cada CD por Gravadora ?

```
SELECT idGravadora, AVG(preco_Venda)
FROM cd
GROUP BY idGravadora;
```

Qual o preço médio de vendas e a quantidade de CD's de cada Gravadora ?

```
SELECT idGravadora, AVG(preco_Venda), COUNT(*)
FROM cd
GROUP BY idGravadora;
```





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

GROUP BY

EXEMPLOS

Qual o preço médio de venda de cada CD por Gravadora ?

```
SELECT idGravadora, AVG(preco_Venda)
FROM cd
GROUP BY idGravadora;
```

Qual o preço médio de vendas e a quantidade de CD's de cada Gravadora ?

```
SELECT idGravadora, AVG(preco_Venda), COUNT(*)
FROM cd
GROUP BY idGravadora;
```





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

Cláusula HAVING

A cláusula **HAVING** é utilizada para restringir o resultado de pesquisa em uma tabela de acordo com a condição desejada;

```
SELECT idMusica, COUNT(*)  
FROM musica_autor  
WHERE idMusica < 15;  
GROUP BY idMusica;
```

```
SELECT idMusica, COUNT(*)  
FROM musica_autor  
GROUP BY idMusica  
HAVING idMusica < 15;
```





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

Cláusula WHERE

- Algumas informações quanto ao uso da cláusula WHERE:
 - As linhas são filtradas PRIMEIRO e DEPOIS são realizados os agrupamentos;
 - Deve vir antes do GROUP BY;
 - Podemos utilizar qualquer coluna das tabelas existentes na cláusula FROM;
 - Não podemos utilizar funções de grupo na cláusula WHERE.





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

Cláusula HAVING

- Algumas informações quanto ao uso da cláusula HAVING:
 - As linhas são agrupadas PRIMEIRO e filtradas DEPOIS.
 - Deve vir após o GROUP BY;
 - Só podemos utilizar as colunas que fazem parte do GROUP BY;
 - Podemos utilizar uma função de grupo (total) para filtrar os grupos;
- Devemos analisar se usamos

HAVING (muitos agrupamentos e com uso da maioria das linhas da tabela)

ou

WHERE (menor parte das linhas da tabela)





DML SELECT AGRUPAMENTO

SQL – DML – Agrupamento de Resultados

Cláusula HAVING – usando função de grupo para filtrar as linhas

```
select idCategoria, AVG(preco_venda)
from cd
group by idcategoria
having AVG(preco_venda) < 10
```

- Considerando que a cláusula HAVING agrupa primeiro
- Considerando que a cláusula HAVING se aplica ao grupo definido na cláusula GROUP BY, a média dos preços de venda será calculada para cada idCategoria.
- No exemplo acima, a média que está sendo utilizada no HAVING é uma para cada grupo (por categoria).

