

**DEFINICAO**

Consultas com expressoes no comando SELECT. Consultas com o uso da clausula WHERE. Agrupamento de dados.

**PROP6SITO**

Saber construir comandos SQL com o uso de expressoes no comando SELECT, bem coma a especificac;ao de condic;oes na clausula WHERE, que representam tarefas importantes no projeto de consultas em sistemas gerenciadores de banco de dados (SGBD). Para o desenvolvimento

de relat6rios e consultas analiticas, e fundamental saber trabalhar com agrupamento de dados. Essas atividades sao relacionadas ao dia a dia de programadores, analistas de sistemas e

desenvolvedores.

**PREPARACAO**

Antes de iniciar o conteudo deste tema, certifique-se deter baixado e instalado o SGBD PostgreSQL em seu computador.

## OBJETIVOS

###### MODULO 1

Operar consultas com o comando SELECT

###### MODULO 2

Operar consultas usando a clausula WHERE

###### MODULO 3

Operar consultas envolvendo agrupamento de dados

## INTRODUCAO

Ao longo deste tema, vamos explorar diversos exemplos de consultas envolvendo uma tabela. Aprenderemos a codificar consultas abrangendo tanto a recuperagao de colunas da pr6pria tabela quanta o uso de express6es no comando SELECT. Quando projetamos um banco de dados para determinado domfnio de neg6cio, em geral, sao criadas diversas tabelas que serao manipuladas pelas aplica96es desenvolvidas para acessar os recursos do banco de dados.

Diversas opera96es que manipulam tabelas em um banco de dados necessariamente estao associadas a alguma operagao de consulta. Por exemplo, se resolvermos aumentar em 10% o

salario de todos os funcionarios que ganham ate R$ 4.000, sera necessario programarmos um comando de consulta para que o sistema gerenciador de banco de dados (SGBD) selecione os registros dos funcionarios alvo da atualizagao. Assim, aprender de maneira efetiva a programar consultas trara beneffcios, tanto para atividades de construc;ao de relat6rios, quanta para o projeto de operac;oes de remoc;ao e atualizagao de dados.

**Clique aqui** para baixar o arquivo com todos os c6digos que serao utilizados nas consultas dos m6dulos a seguir.



# MODULO 1

**0 Operar consultas com o comando SELECT**

# ESTRUTURA BASICA DE UM COMANDO SELECT

0 comando SELECT e usado para exibir dados resultantes de uma consulta. Os dados podem ser colunas ffsicas de uma tabela, colunas calculadas ou mesmo resultado do uso de expressoes e func;oes. Uma sintaxe **basica** para o comando SELECT esta expressa a seguir:

SELECT COLUNA 1 [[AS] APELIDOCOLUNA 1], COLUNA2 [[AS] APELIDOCOLUNA2],

COLUNAN [[AS] APELIDOCOLUNAN] FROM TABELA;

E importante ressaltar que estamos diante de uma sintaxe simplificada o suficiente para entendimento dos exemplos que iremos explorar ao longo do modulo. A sintaxe completa abrange

todos os recursos do PostgreSQL.

Uma sintaxe complexa envolve uma serie de clausulas e recursos bastante uteis para consultas de maior complexidade.

Na pratica, o comando SELECT, dependendo da consulta desejada, pode ser usado de diferentes formas para obter o mesmo resultado. E importante frisar que a clausula SELECT realiza a operac;ao de projec;ao da Algebra Relacional.

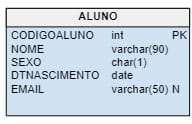
Caso haja interesse em exibir todas as colunas especificadas em uma consulta, basta adicionar um"\*", conforme a seguir: SELECT\* FROM TABELA;

###### 0 VOCE SABIA

Alguns SGBDs, coma o PostgreSQL, implementam uma forma simplificada do comando SELECT\* FROM TABELA, que e simplesmente TABLE tabela (voce pode testar isso no PostgreSQL).

Vamos estudar alguns exemplos?

Construiremos as consultas com base na tabela ALUNO, conforme figura a seguir:



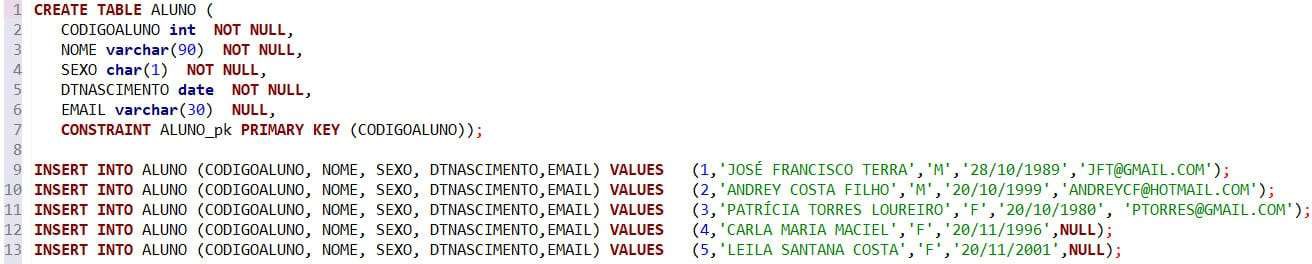
Fonte: 0 autor Tabela ALUNO.

Recomendamos que voce crie a tabela e insira algumas linhas, o que pode ser feito usando o

*script* a seguir, a partir da ferramenta de sua preferencia.

Para isso, tenha em mente que e necessario estar conectado ao PostgreSQL e acessando algum

*database* criado por vocEL



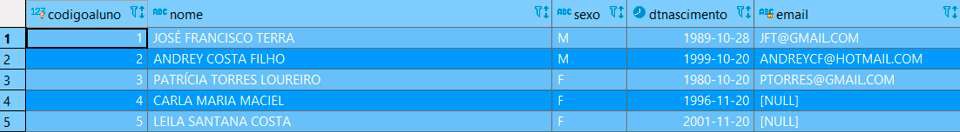
Fonte: 0 autor.

Vamos ver alguns exemplos de consultas?

###### CONSULTA 01

Exibir todas as informac;oes dos alunos. SELECT\* FROM ALUNO;

A tabela a seguir apresenta os resultados da consulta.



Fonte: 0 autor.

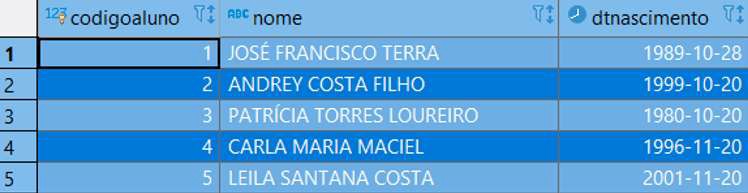
**aJ** Resultados da consulta O1.

Ao executar a consulta, o SGBD percorre todos os registros da tabela ALUNO e exibe as colunas dessa tabela.

###### CONSULTA02

Retornar o c6digo, o nome ea data de nascimento de todos os alunos. SELECT CODIGOALUNO, NOME, DTNASCIMENTO

FROMALUNO;



Fonte: 0 autor.

**aJ** Resultados da consulta 02.

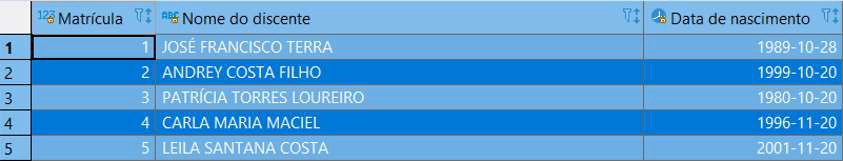
Na consulta 02, foram especificadas tres colunas da tabela ALUNO para serem exibidas ao usuario.

Em especial, pode ser interessante "renomear'' as colunas resultantes da consulta, visando tornar os resultados mais "apresentaveis" ao usuario da aplicac;ao. Par exemplo, a consulta 02 pode ser reescrita conforme a seguir:

SELECT CODIGOALUNOAS "Matrfcula",

NOME AS "Nome do discente", DTNASCIMENTO AS "Data de nascimento" FROMALUNO;,

0 resultado dessa consulta seria este:



Fonte: 0 autor

**aJ** Resultados da segunda versao da consulta 02.

E importante ressaltar que, na tabela anterior, o name apresentado para cada coluna nao existe fisicamente no banco de dados.

Vamos aprender a seguir que nem toda coluna resultante de uma consulta representa necessariamente uma coluna de alguma tabela.

**FUNC0ES DE DATA E HORA**

Quando desenvolvemos consultas, e comum manipularmos colunas e func;oes que envolvem dados representativos de datas.

Func;oes de data do PostgreSQL.

**Um resumo contendo algumas func;oes de data do PostgreSQL pode ser visualizado na tabela a seguir:**

|  |  |
| --- | --- |
| **Func;ao** | **0 que retorna?** |
| current date | data de hoje |
| current time | hora do dia |
| current\_timestamp | data ea hora |
| extract (campo from fonte) | subcampos de data e hora: seculo, ano, dia, mes |

**Atenc;ao!** Para visualizagao completa da tabela utilize a rolagem horizontal Fonte: 0 autor.

Um quadro complete contendo informagoes sabre fungoes de data e hora pode ser encontrado na documentagao oficial do PostgreSQL.

Vamos estudar alguns exemplos? Observe com atengao o c6digo:



Fonte: 0 autor

Agora veja na tabela a seguir os resultados da consulta:



Fonte: 0 autor.

II Resultados da consulta envolvendo fungoes de data e hora.

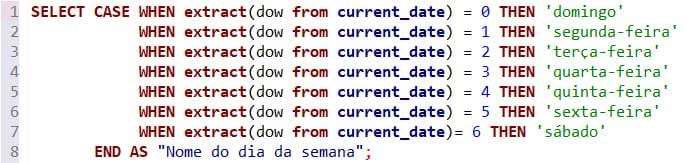
Observe que utilizamos o qualificador *AS ''Apelido"para* facilitar o entendimento do retorno de cada fun9ao. Note tambem que nao ha clausula FROM na consulta, visto que todas as colunas retornadas representam o resultado de fun96es do PostgreSQL sem envolver qualquer tabela do domfnio da aplica9ao.

###### ATENCAO

Convem ressaltar que, no padrao SOL, a clausula FROM e obrigat6ria. No entanto, o PostgreSQL permite executar um comando SELECT **sem** a clausula FROM. Experimente executar SELECT 5+5;

#### EXIBINDO O NOME DO DIADASEMANA

Perceba que a linha 6 do c6digo acima retorna um inteiro representativo do dia da semana. No entanto, se houver necessidade de exibir o dia da semana, voce pode usar o c6digo a seguir:



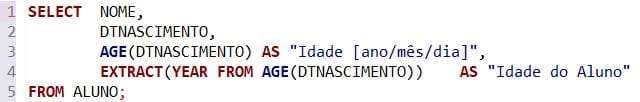
Fonte: 0 autor

Observe que construfmos uma 16gica utilizando o comando CASE, que e equivalente ao comando IF:, cada linha com a clausula WHEN avalia expressao que retorna um inteiro representativo do dia da semana, caso a expressao tenha valor 16gico verdadeiro.

#### CALCULANDO IDADE E FAIXA ETARIA

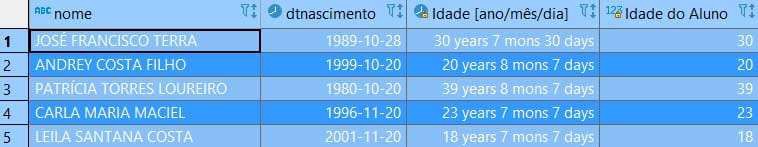
Em geral, quando estamos diante de alguma coluna representativa da data de nascimento de uma pessoa, e comum extrair informa96es derivadas, tais como idade e faixa etaria. Por

exemplo, o c6digo a seguir retorna o nome, a data de nascimento ea idade dos alunos:



Fonte: 0 autor

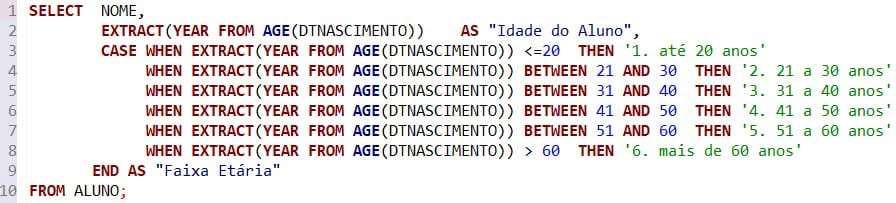
Perceba que na linha 3 utilizamos a furn;ao AGE, que retorna uma frase representativa da informai;ao sabre a idade em questao. Na linha 4, usamos a funi;ao EXTRACT para exibir a idade do aluno. A figura a seguir apresenta o resultado dessa consulta:



Fonte: 0 autor

d'J Exibindo a idade dos alunos.

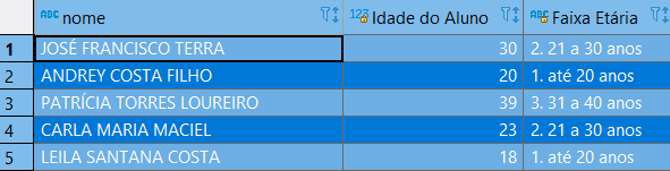
Muito bem, agora, vamos exibir o nome, a idade ea faixa etaria dos alunos. Observe o c6digo SOL a seguir:



Fonte: 0 autor

Perceba que cada linha com a clausula WHEN avalia a expressao que retorna uma faixa etaria de acordo com a idade do aluno.

A seguir, o resultado da consulta:



Fonte: 0 autor.

**aJ** Resultados da consulta envolvendo idade e faixa etaria dos alunos.

# FUNCOES DE RESUMO OU DE AGREGACAO

As fung6es a seguir sao uteis para obtermos resumo dos dados de alguma tabela: Fung6es para resumo de dados.

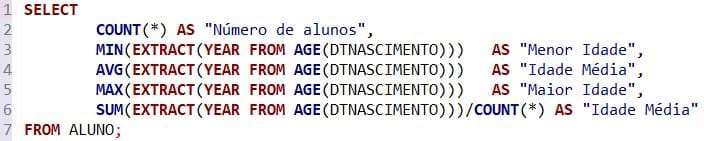
|  |  |
| --- | --- |
| **Fun ao** | **0 que retorna?** |
| COUNT(\*) | numero de linhas da consulta |
| MIN(COLUNA/EXPRESSAO) | menor de uma coluna ou expressao |
| AVG(COLUNA/EXPRESSAO) | valor media da coluna ou expressao |
| MAX(COLUNA/EXPRESSAO) | maior valor de uma coluna ou expressao |
| SUM(COLUNA/EXPRESSAO) | soma dos valores de uma coluna ou expressao |
| STDDEV(COLUNA/EXPRESSAO) | desvio padrao dos valores de uma coluna ou expressao |
| **VARIANCE(COLUNA/EXPRESSAO)** | variancia dos valores de uma coluna ou |

expressao

**Aten ao!** Para visualizagao completa da tabela utilize a rolagem horizontal Fonte: 0 Autor.

Vamos estudar um exemplo?

Observe o c6digo a seguir:



Fonte: 0 autor

Perceba que, como estamos usando somente o comando SELECT/FROM, cada fungao e calculada levando em consideragao todos os registros da tabela.

Veja na figura a seguir, o resultado da consulta:



Fonte: 0 autor.

**aJ** Resultado da consulta envolvendo fungoes para extrair resumo a partir da tabela aluno.

**'l::l ATENCAO**

Perceba, tambem, que o c6digo da linha 6 e equivalente ao da linha 4: ambos calculam a idade media dos alunos.

#### LISTANDO RESUMOS EM UMA LINHA

Suponha que haja interesse em conhecer os quantitativos de cursos, disciplinas e alunos do nosso banco de dados.

Poderfamos submeter ao SGBD as consultas a seguir:

##### CURSO

SELECT COUNT(\*) NCURSOS FROM CURSO;

##### DISCIPLINA

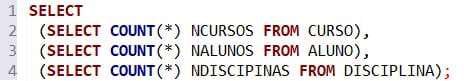
SELECT COUNT(\*) NDISCIPINAS FROM DISCIPLINA;

##### ALUNO

SELECT COUNT(\*) NALUNOS FROM ALUNO;

Estamos diante de tres consultas. No entanto, pode ser mais interessante mostrarmos os resultados em apenas uma linha.

Podemos, entao, submeter o c6digo a seguir:



Fonte: 0 autor

0 que fizemos?

Como cada consulta (linhas 2 a 4) retorna somente um valor, utilizamos um SELECT externo (linha 1) para exibir cada coluna resultante.

Observe o resultado a seguir:



Fonte: 0 autor.

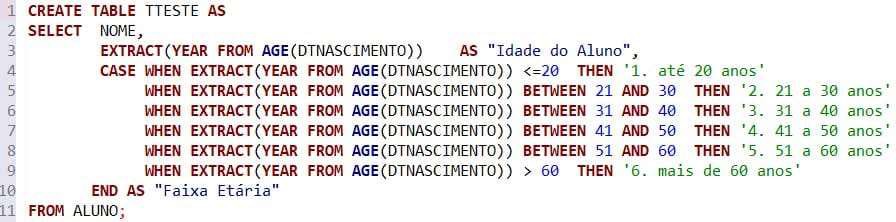
**aJ** Resultado da consulta envolvendo quantitativos de cursos, alunos e disciplinas.

Convem ressaltar que o comando e valido, visto que, no PostgreSQL, a clausula FROM nao e obrigat6ria.

## CRIANDO TABELAA PARTIR DE CONSULTA

Em alguns momentos, voce tera interesse em salvar os resultados de uma consulta em uma nova tabela.

Para isso, basta usar o comando CREATE TABLE <CONSULTA>.



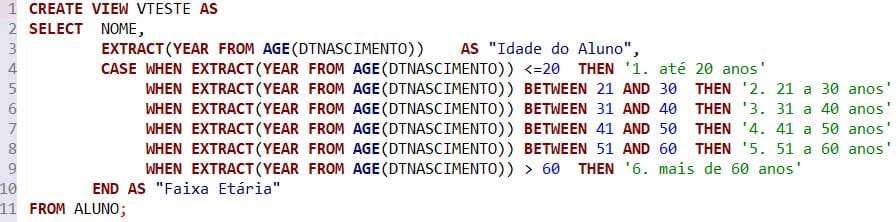
Fonte: 0 autor

No exemplo apresentado, o SGBD criara uma tabela denominada TTESTE e armazenara os dados resultantes da consulta (linhas 2 a 11) em questao.

### CRIANDO *VIEW* A PARTIR DE CONSULTA

Outro recurso interessante, diretamente relacionado ao processo de constrU<;ao de consultas, e o

objeto view(visao). Uma viewencapsula a complexidade da consulta SOL, que a forma. Para criar esse objeto, usa-se o comando CREATE VIEW <CONSULTA>.

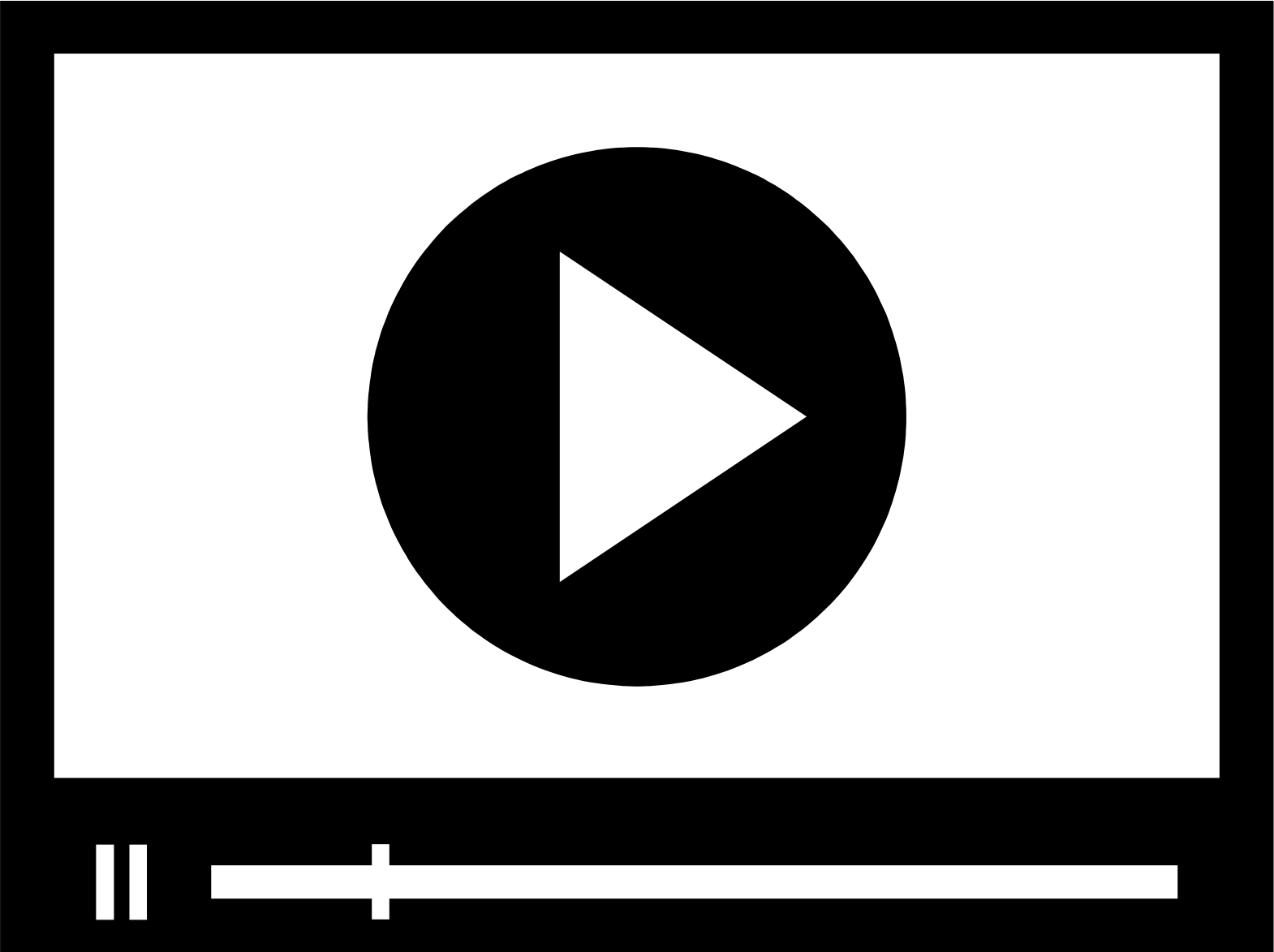


Fonte: 0 autor

No exemplo, o SGBD criara uma viewdenominada VTESTE. Na pratica, quando usuario

submeter, por exemplo, a consulta SELECT\* FROM VTESTE, o SGBD executara o c6digo associado a *viewem* questao.

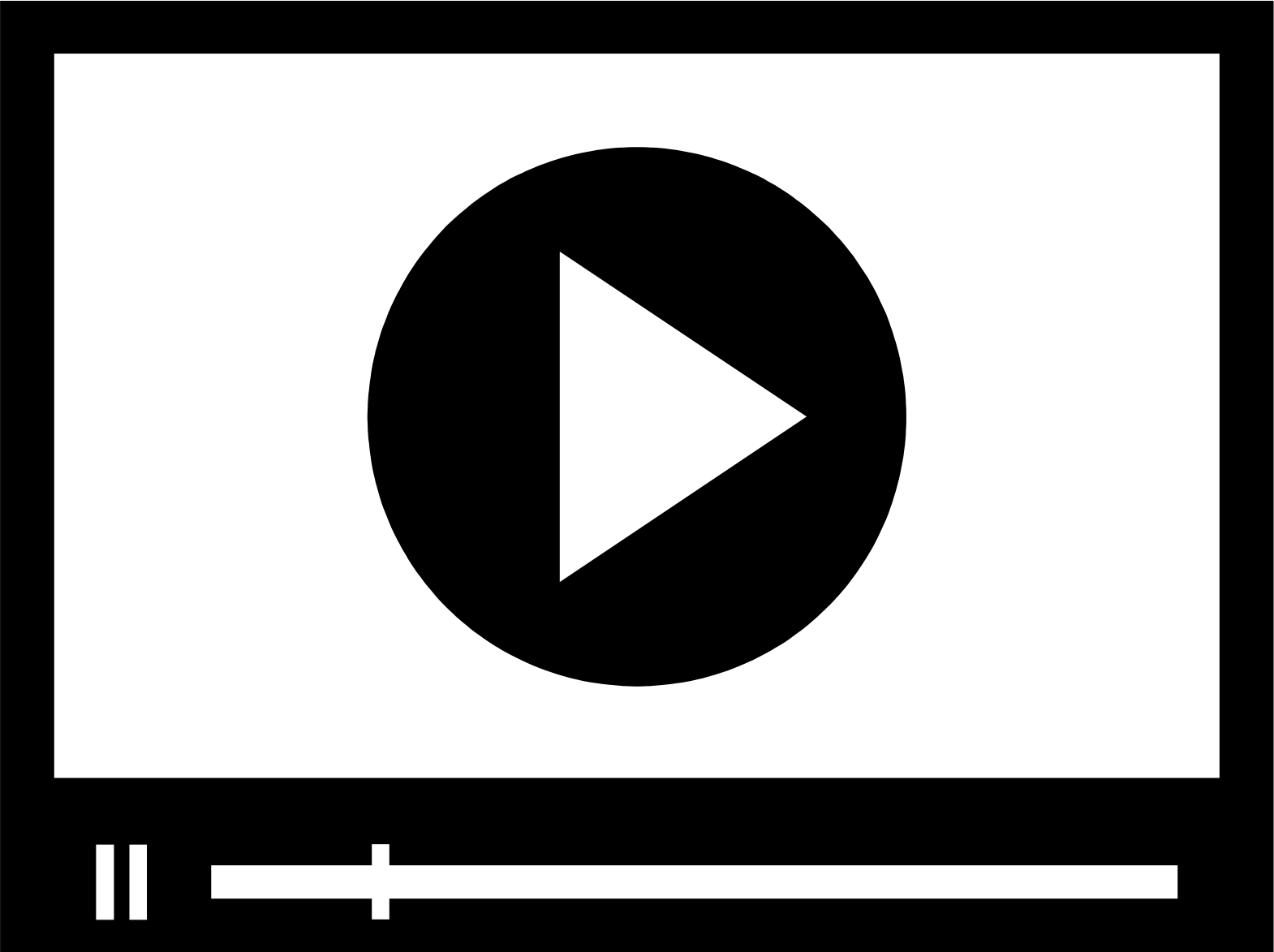
Atem;ao: Antes de assistir o vfdeo a seguir realizar o download do arquivo **EMPRESA.SQL** e criar o banco de dados utilizando o PGADMIN.



## CONSULTAS COM O COMANDO SELECT UTILIZANDO PGADMIN, INCLUINDO CRIACAO DE TABELA E VIEW

Veja agora o vfdeo sobre Cria ao de tabela e view a partir de consulta





## CONSULTAS COM O COMANDO SELECT UTILIZANDO PLSQL

Veja agora o vfdeo sobre **Consultas simples com o comando SELECT no PostgreSQL**

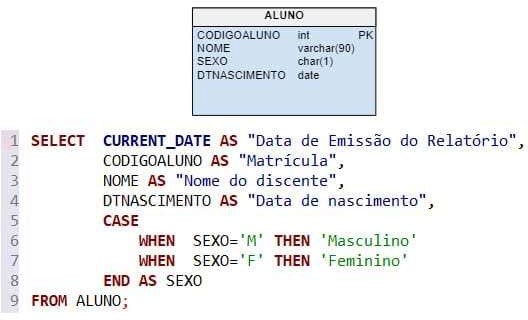


Ao longo da nossa jornada, estudamos a constru9ao de consultas envolvendo a extra9ao de informa9ao a partir de uma tabela. Alem disso, foram exibidas fun9oes de data e fun9oes para resumir dados de uma tabela.

Agora e com voce! Vamos realizar as atividades a seguir?

## VERIFICANDO O APRENDIZADO

1. CONSIDERE A TABELA E O CODIGO SQL A SEGUIR:



FONTE: 0 AUTOR

ANALISE AS SEGUINTES PROPOSICOES:

I-ACONSULTARETORNAINFORMACOES SOBRE CINCO COLUNAS EXISTENTES NA TABELAALUNO.

11-ACONSULTARETORNAINFORMACOES SOBRE TODOS OSALUNOS

CADASTRADOS.

111- PODE EXISTIR REGISTRO COM VALOR "MASCULINO" ARMAZENADO NA COLUNA SEXO.

IV -0 RESULTADO DE CURRENT \_DATE (LINHA 1) ESTAARMAZENADO EM UMA COLUNA DAT ABELAALUNO.

1. ACONSULTARETORNAINFORMACOES SOBRE QUATRO COLUNAS EXISTENTES NA TABELAALUNO.

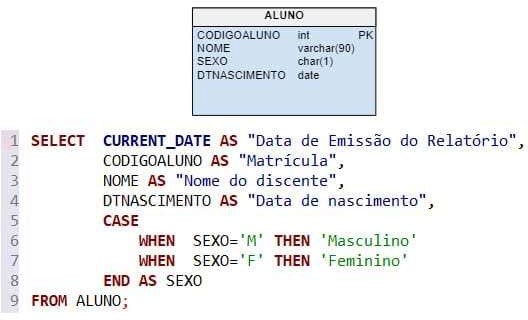
SAO PROPOSICOES VERDADEIRAS:

* 1. le II.
  2. lleV.
  3. II, Ille IV.
  4. llleV.

1. SEJA UMA TABELAASSIM DEFINIDA: FUNCIONARIO (IDFUNC, NOME, DATANASCIMENTO, SALARIO). QUAL CONSULTASQL RETORNAO MAIOR SALARIO?
   1. SELECT> SALARIO FROM FUNCIONARIO.
   2. SELECT MAX(SALARIO) FROM FUNCIONARIO.
   3. SELECT AVG(SALARIO) FROM FUNCIONARIO.
   4. SELECT FUNCIONARIO FROM SALARIO.

GABARITO

* + 1. Considere a tabela e o c6digo SQL a seguir:



Fonte: 0 autor

**Analise as seguintes proposic;oes:**

**1-Aconsulta retorna informac;oes sobre cinco colunas existentes na tabela ALUNO. 11-Aconsulta retorna informac;oes sobre todos os alunos cadastrados.**

Ill - **Pode existir registro com valor "Masculino" armazenado na coluna SEXO.**

1. **-0 resultado de CURRENT\_DATE (linha 1) esta armazenado em uma coluna da tabela ALUNO.**
2. **-Aconsulta retorna informac;oes sobre quatro colunas existentes na tabela ALUNO.**

**Sao proposic;oes verdadeiras:**

A alternativa **118** 11 esta correta.

A proposi9ao II e verdadeira, pois nao ha condi9ao de filtro na consulta. A proposi9ao Ve

verdadeira, pois retorna informa96es a respeito de todas as colunas da tabela ALUNO. As demais proposi96es sao falsas.

* + 1. **Seja uma tabela assim definida: FUNCIONARIO (IDFUNC, NOME, DATANASCIMENTO, SALARIO). Qual consulta SQL retorna o maior salario?**

A alternativa **118** 11 esta correta.

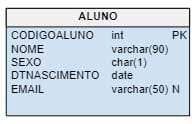
Na alternativa B, foi usado o comando MAX para retornar o maior valor da coluna SALARIO da tabela FUNCIONARIO.

# M6DUL02

0 **Operar consultas usando a clausula WHERE**

# CLAUSULA WHERE E OPERADORES DA SQL

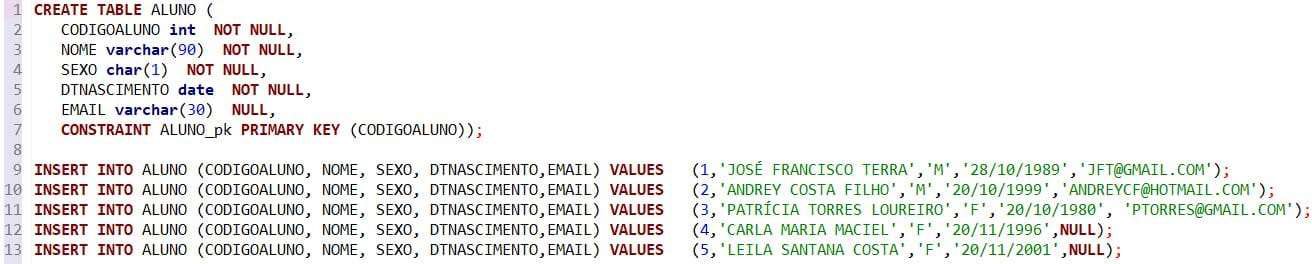
Em nossas consultas, usaremos como base a tabela ALUNO, conforme figura a seguir:



Fonte: 0 autor

**rl** Tabela ALUNO.

Recomendamos que voce crie a tabela e insira algumas linhas, o que pode ser feito usando o *script* a seguir, a partir da ferramenta de sua preferencia. Para isso, tenha em mente que e necessario estar conectado ao PostgreSQL e acessando algum *database* criado por voce.



Fonte: 0 autor

## RECUPERANDO DADOS COM SELECT/FROM/WHERE/ORDER BY

Uma sintaxe basica para o comando SELECT, com o uso das clausulas WHERE e ORDER BY, esta expressa a seguir:

SELECT COLUNA 1 [[AS] APELIDOCOLUNA 1], COLUNA2 [[AS] APELIDOCOLUNA2],

COLUNAN [[AS] APELIDOCOLUNAN] FROM TABELA

WHERE <CONDl<;AO>

ORDER BY EXPRESSA01[ASCIDESC] [NULLS {FIRSTILAST}], [EXPRESSA02[ASCIDESC] [NULLS {FIRSTILAST} ... ];

0 prop6sito do SELECT e declararmos as colunas da consulta. No FROM, informamos a tabela alvo da consulta. No WHERE, especificamos alguma condic;ao, simples ou composta, para filtrar registros que serao recuperados pelo SGBD. No ORDER BY, declaramos uma ou mais colunas como criteria de ordenac;ao, com possibilidade de especificarmos se valores NULL aparecem no infcio ou no final do resultado.

E importante frisar que a clausula WHERE realiza a operac;ao de restric;ao da Algebra Relacional, tambem conhecida como selec;ao - nao confundir com o comando SELECT.

Ainda, a construc;ao de uma condic;ao na clausula WHERE envolve operadores relacionais, conforme tabela a seguir:

Operadores relacionais.

|  |  |
| --- | --- |
| **Operador** | **Significado** |
| < | menor |
| <= | menor ou igual a |
| > | maior |
| >= | maior ou igual a |
| = | igual |
| **<>OU!=** | > diferente |

**Atenc;ao!** Para visualizac;ao completa da tabela utilize a rolagem horizontal Fonte: 0 autor.

Alem dos operadores relacionais, a construc;ao de uma condic;ao na clausula WHERE pode fazer

uso dos seguintes operadores 16gicos: Operadores 16gicos.

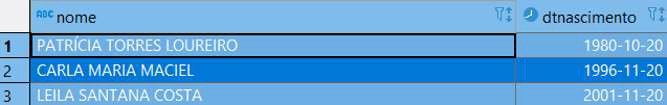
|  |  |
| --- | --- |
| **Operador** | **Significado** |
| AND | conjum;ao |
| OR | disjunc;ao |
| NOT | negac;ao |

**Atenc;ao!** Para visualizac;ao completa da tabela utilize a rolagem horizontal Fonte: 0 autor.

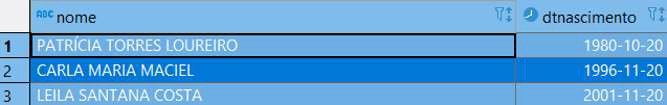
Vamos estudar alguns exemplos de consultas com o uso da clausula WHERE?

###### CONSULTA 01 + RESULTADO

Mostrar o nome ea data de nascimento das professoras.



Fonte: 0 autor



Fonte: 0 autor

Resultado consulta 01.

Perceba que foi criada uma condic;ao simples de igualdade envolvendo a coluna SEXO da tabela ALUNO. 0 SGBD percorre cada registro da tabela ALUNO, avalia a condic;ao (linha 3) e exibe as colunas NOME e DTNASCIMENTO para cada registro cuja avaliac;ao da condic;ao retorne verdadeiro.

###### CONSULTA 02 + RESULTADO

Mostrar o nome ea data de nascimento das professoras que fazem aniversario em novembro.



Fonte: 0 autor



Fonte: 0 autor

**aJ** Resultado consulta 02.

Perceba que foi criada uma condic;ao composta envolvendo uma conjunc;ao. 0 SGBD retornara os registros que possuem o valor "F" para a coluna SEXO e o inteiro 11 como valor do mes

referente a data de nascimento.

## RECUPERANDO DADOS COM O USO DO OPERADOR IN

0 operador [NOT] IN pode ser utilizado em consultas que envolvam comparac;oes usando uma lista de valores.

###### CONSULTA 03 + RESULTADO

Li star o nome dos alunos que fazem aniversario no segundo semestre.



Fonte: 0 autor



Fonte: 0 autor

II Resultado consulta 03.

Note que a expressao na clausula WHERE compara o mes de nascimento de cada aluno junto aos valores da lista contendo os inteiros correspondentes aos meses do segundo semestre.

## RECUPERANDO DADOS COM O USO DO OPERADOR BETWEEN

0 operador [NOl] BElWEEN verifica se determinado valor encontra-se no intervalo entre dois valores.

Por exemplo, *X BETWEEN Y AND* Z e equivalente a *X>=Y AND* X<=Z. De modo semelhante, *X NOT BETWEEN Y AND* Z e equivalente a *X<Y OR X>Z.*

CONSULTA 04 + RESULTADO

Listar o nome dos alunos nascidos entre 1985 e 2005.



Fonte: 0 autor



Fonte: 0 autor

II Resultado consulta 04.

Note que a expressao na clausula WHERE compara o anode nascimento de cada aluno junto ao intervalo especificado pelo operador BElWEEN. Caso quisessemos extrair o mesmo resultado

sem o uso do BElWEEN, poderfamos programar um comando equivalente, conforme a seguir:



Fonte: 0 autor

## RECUPERANDO DADOS COM O USO DO OPERADOR LIKE

0 uso do [NOT] LIKE permite realizar buscas em uma cadeia de caracteres.

Trata-se de um recurso bastante utilizado em buscas textuais. Voce pode utilizar os simbolos especiais a seguir:

\_ para ignorar qualquer caractere especffico;

(Underline)

% para ignorar qualquer padrao.

(Percentual)

Vamos estudar alguns exemplos?

###### CONSULTA 05 + RESULTADO

Listar o nome dos alunos que possuem a *string* COSTA em qualquer parte do nome.



Fonte: 0 autor



Fonte: 0 autor

Resultado consulta 05.

0 SAIBAMAIS

0 uso do padrao '%COSTA%' significa que nao importa o conteudo localizado antes e depois da

*string* "COSTA".

CONSULTA 06 + RESULTADO

Listar o name dos alunos que possuem a letra "A" na segunda posic;ao do name.



Fonte: 0 autor



Fonte: 0 autor

**m** Resultado consulta 06.

Note que, para especificar o "A" na segunda posic;ao, o SGBD desprezara qualquer valor na primeira posic;ao da *string,* nao importando o que estiver localizado a direita do "A".

CONSULTA 07 + RESULTADO

Listar o name e a data de nascimento dos alunos que nao possuem a *string* "MARIA" fazendo parte do name.



Fonte: 0 autor



Fonte: 0 autor

**m** Resultado consulta 07.

Estamos diante de um caso semelhante ao da consulta 05.

No entanto, utilizamos o operador de negac;ao para retornar os registros de interesse.

###### CONSULTA 08 + RESULTADO

Quantos alunos possuem conta de e-mail no gmail?



Fonte: 0 autor



Fonte: 0 autor

**!'I** Resultado consulta 08.

Note que, mais uma vez, estamos diante de um caso semelhante ao da consulta 05. Buscamos pela string "@GMAIL. "em qualquer posic;ao da coluna EMAIL.

## RECUPERANDO DADOS COM O USO DO OPERADOR NULL

Quando uma coluna e opcional, significa que existe possibilidade de que algum registro nao possua valor cadastrado para a coluna em questao. Nessa hip6tese, ha entendimento de que o valor da coluna e "desconhecido" ou "nao aplicavel".

Para testar se uma coluna possui valor cadastrado, usa-se a expressao COLUNA IS NOT NULL.

Vamos estudar alguns exemplos?

###### CONSULTA 09 + RESULTADO

Listar o nome, a data de nascimento e o e-mail dos alunos que tern enderec;o eletr6nico cadastrado.



Fonte: 0 autor



Fonte: 0 autor

**m** Resultado consulta 09.

0 SGBD retorna os registros onde ha algum conteudo cadastrado na coluna EMAIL.

CONSULTA 10 + RESULTADO

Retornar o nome dos alunos sem e-mail cadastrado no banco de dados.



Fonte: 0 autor



Fonte: 0 autor

**m** Resultado consulta 10.

0 SGBD retorna os registros sobre os quais nao ha valor cadastrado na coluna EMAIL.

## RECUPERANDO DADOS USANDO ORDENACAO DOS RESULTADOS

Para melhor organizar os resultados de uma consulta, n6s podemos especificar criterios de ordenac;ao. Vejamos alguns exemplos:

CONSULTA 11 + RESULTADO

Retamar o name ea data de nascimento dos alunos, ordenando as resultados par name, de maneira ascendente.



Fonte: 0 autor



Fonte: 0 autor

**al** Resultado consulta 11.

0 SGBD retoma as registros da tabela ALUNO, obedecendo ao criteria de ordenagao especificado na linha 3 da consulta. 0 padrao ascendente (ASC) e opcional.

CONSULTA 12 + RESULTADO

Retamar o name ea data de nascimento dos alunos, ordenando as resultados de modo ascendente pelo mes de nascimento e, em seguida, pelo name, tambem de modo ascendente.



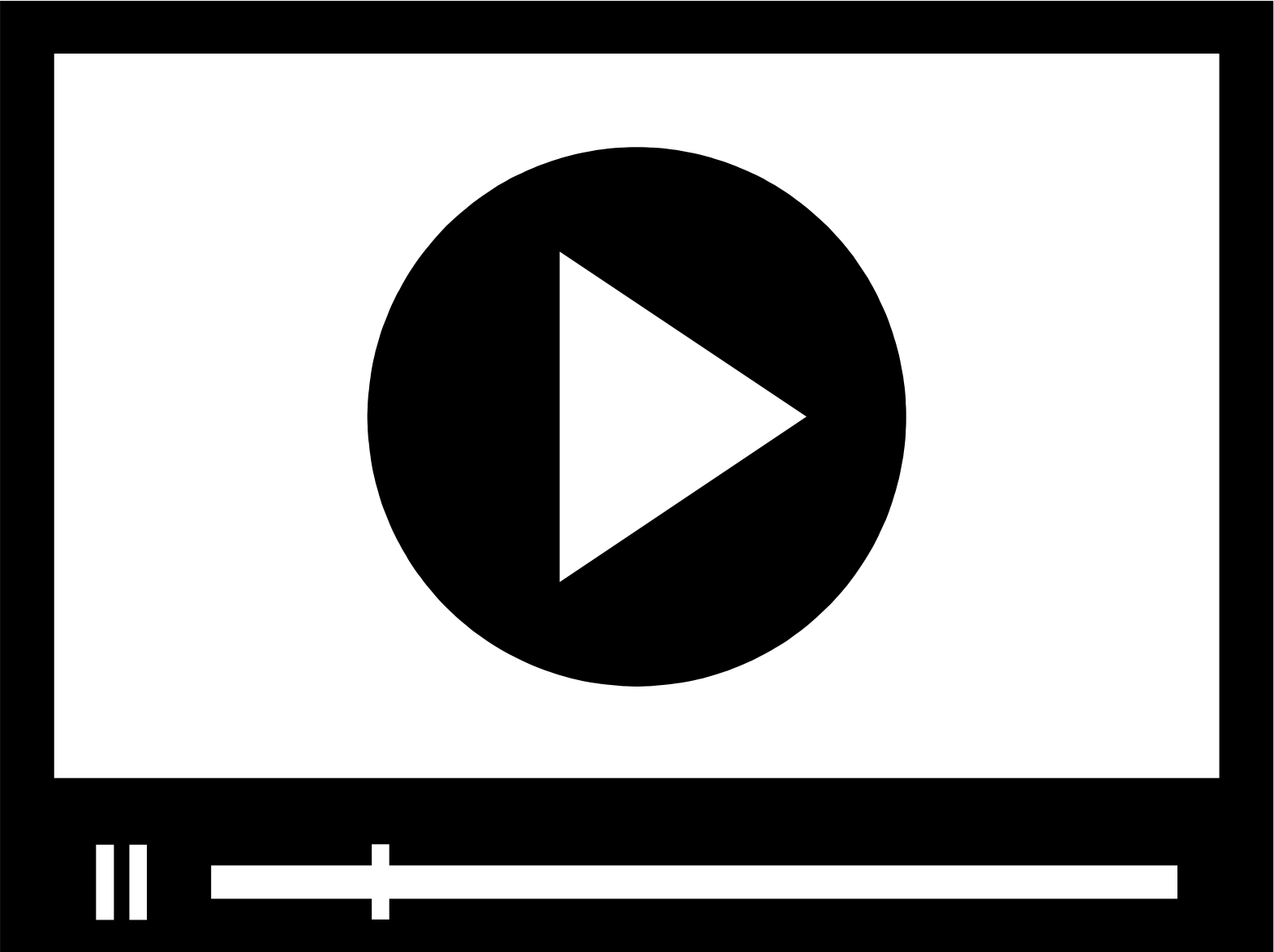
Fonte: 0 autor



Fonte: 0 autor

**al** Resultado consulta 12.

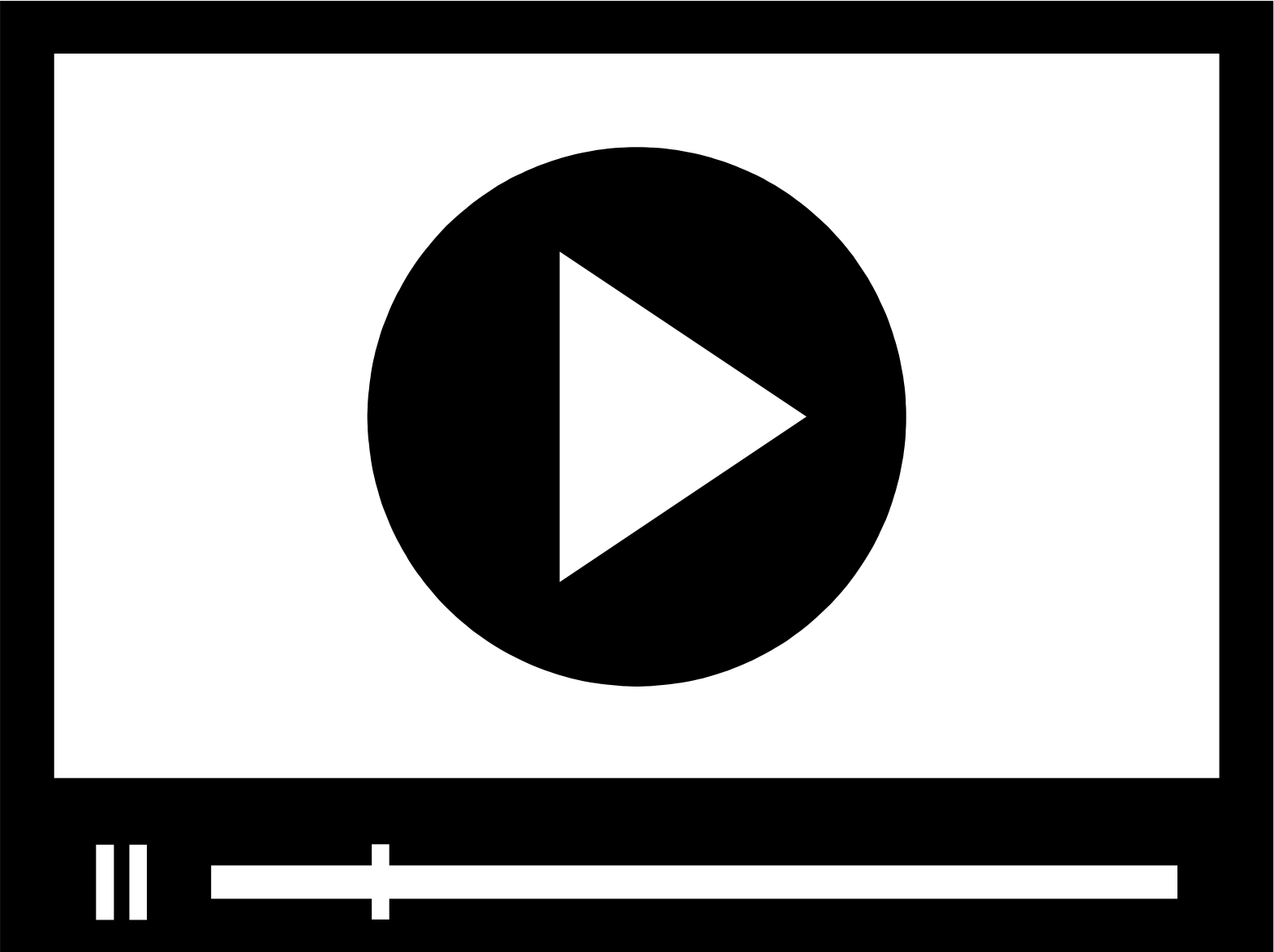
0 SGBD retoma as registros da tabela ALUNO, levando em conta o criteria de ordenagao especificado na linha 3 da consulta. Foi realizada ordenagao pelo mes de nascimento; em seguida, pelo name.



# CONSULTAS COM O COMANDO SELECT EA CLAUSULA WHERE UTILIZANDO PGADMIN

A seguir, veja o vfdeo: Estudos de casos com o comando SELECT ea clausula WHERE





# CONSULTAS COM O COMANDO SELECT EA CLAUSULA WHERE UTILIZANDO PLSQL

A seguir, veja o vfdeo: **Consultas com o comando SELECT ea clausula WHERE**



Trabalhamos o uso de consultas com auxnio da clausula WHERE, fazendo uso de operadores relacionais e 16gicos na composigao de condigoes 16gicas. Alem disso, estudamos como estabelecer criterios para ordenagao dos resultados de consultas.

Agora e com voce! Vamos realizar as atividades a seguir?

VERIFICANDO O APRENDIZADO

1. GABRIELE ANALISTADE SISTEMAS DE UMAEMPRESADE TECNOLOGIADE INFORMACAO E RECEBEUATAREFADE RECUPERAR TODOS OS REGISTROS DA TABELACLIENTE ONDE O VALOR DA COLUNA "NOMECLIENTE" CONTENHAACADEIA "DASILVA" EM QUALQUER PARTE DO NOME. ASSINALE AALTERNATIVA CORRETA QUE PERMITAA GABRIEL EXECUTAR SUA TAREFA.
   1. SELECT\* FROM CLIENTE WHERE NOMECLIENTE = 'da Silva'.
   2. SELECT\* FROM CLIENTE WHERE NOMECLIENTE != 'da Silva'.
   3. SELECT\* FROM CLIENTE WHERE NOME LIKE '%da Silva%'.
   4. SELECT\* FROM CLIENTE WHERE NOMECLIENTE LIKE '%da Silva%'.
2. UM PROGRAMADOR RECUPEROU OS DADOS DOS BAIRROS PENHA, IPANEMA, FLAMENGO E CENTRO GRAVADOS NACOLUNABAIRRO DA TABELACLIENTE, ASEGUIR ESPECIFICADA.

**CLIENTE (IDCLIENTE, NOME, ENDERECO, BAIRRO, CIDADE, UF, CEP) A SINTAXE SQL CORRETA USADA POR ELE PARA REALIZAR ESSA ATIVIDADE FOi SELECT** \* **FROM CLIENTE**

* 1. WHERE BAIRRO IN ('Penha', 'lpanema ', 'Flamengo ', 'Centro').
  2. WHERE BAIRRO =('Penha', 'lpanema ', 'Flamengo ', 'Centro').
  3. WHEN BAIRRO =('Penha', 'lpanema ', 'Flamengo ', 'Centro').
  4. WHERE BAIRRO BETWEEN ('Penha', 'lpanema ', 'Flamengo ', 'Centro').

GABARITO

* + 1. **Gabriele analista de sistemas de uma empresa de tecnologia de informa ao e recebeu a tarefa de recuperar todos os registros da tabela CLIENTE onde o valor da coluna "NOMECLIENTE" contenha a cadeia "da Silva" em qualquer parte do nome. Assinale a alternativa correta que permita a Gabriel executar sua tarefa.**

A alternativa **11D** II esta correta.

Para recuperar os registros que contenham "da Silva" em qualquer parte do name, utiliza-se o comando LIKE com auxnio do"%" coma forma do SGBD desconsiderar qualquer padrao a esquerda ea direita da string de interesse.

* + 1. **Um programador recuperou os dados dos bairros Penha, lpanema, Flamengo e Centro gravados na coluna BAIRRO da tabela CLIENTE, a seguir especificada.**

**CLIENTE (IDCLIENTE, NOME, ENDERECO, BAIRRO, CIDADE, UF, CEP)**

**A sintaxe SQL correta usada por ele para realizar essa atividade foi SELECT\* FROM CLIENTE**

A alternativa **11A** II esta correta.

Para recuperar os registros de interesse, foi utilizado o operador IN com o uso de uma lista contendo os bairros em questao. 0 SGBD compara o bairro do cliente junta aos elementos especificados na lista de bairros em questao.

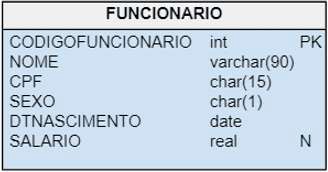
## M6DULO 3



**0 Operar consultas envolvendo agrupamento de dados**

## CONSULTAS COM GROUP BYE HAVING

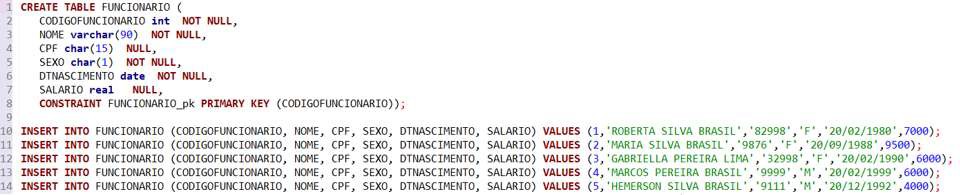
Em nossas consultas, usaremos coma base a tabela FUNCIONARIO, conforme figura a seguir:



Fonte: 0 autor

rl Tabela FUNCIONARIO.

Recomendamos que voce crie a tabela e insira algumas linhas, o que pode ser feito usando o *script* a seguir, a partir da ferramenta de sua preferencia. Para isso, tenha em mente que e necessario estar conectado ao PostgreSQL e acessando algum *database* criado por voce.



Fonte: 0 autor.

Ap6s a criac;ao da tabela ea inserc;ao dos registros, podemos utilizar o c6digo a seguir para exibir todo o seu conteudo:



Fonte: 0 autor

0 resultado da consulta sera semelhante a este:



Fonte: 0 autor

rl Registros da tabela FUNCIONARIO.

## GRUPO DE DADOS

Nas pr6ximas segoes, vamos aprender a projetar consultas com o uso de agrupamento de dados, com auxnio dos comandos GROUP BY e HAVING.

Vamos perceber que a maior parte dessas consultas esta atrelada ao uso de alguma fungao de resumo, por exemplo, SUM, AVG, MIN e MAX, as quais representam, respectivamente, soma, media, mfnimo e maxima.

Logo, essas consultas sao uteis para quern tern interesse em construir relat6rios e aplicagoes de natureza mais gerencial e analftica. Os valores de determinada coluna podem formar grupos sobre os quais podemos ter interesse em recuperar dados.

Por exemplo, se avaliarmos o resultado da consulta anterior, podemos naturalmente dividir os registros de acordo com o valor da coluna sexo. Terfamos, entao, uma estrutura conforme a seguir:

M {4,5} {MARCOS PEREIRA BRASIL, HEMERSON SILVA BRASIL}{... }

F {1,2,3} {ROBERTA SILVA BRASIL, MARIA SILVA BRASIL, GABRIELLA PEREIRA LIMA}

{... }

**Aten ao!** Para visualizagao completa da tabela utilize a rolagem horizontal

###### 0 SAIBA MAIS

Todas as linhas com o mesmo valor para a coluna sexo formam um grupo. Representamos as linhas com mais de um valor com o uso de chaves para fins ilustrativos, dado que estamos diante de linhas dentro de colunas, uma representagao que nao existe no modelo relacional.

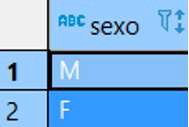
A estrutura anterior possui somente um grupo, o qual e formado pela coluna SEXO da tabela FUNCIONARIO. Como exibir esse grupo em SQL?

Lima solugao e adicionar a clausula DISTINCT ao comando SELECT, conforme a seguir:



Fonte: 0 autor

0 resultado da consulta anterior pode ser visualizado na figura a seguir:



Fonte: 0 autor.

**GI** Grupo de dados baseado na coluna SEXO da tabela FUNCIONARIO.

Vamos perceber que em SQL a clausula mais adequada para trabalhar com agrupamento de

**dados e O GROUP BY.**

## GRUPO DE DADOS COM GROUP BY

A clausula GROUP BY serve para exibir resultados de consulta de acordo com um grupo especificado. Ela e declarada ap6s a clausula FROM, ou ap6s a clausula WHERE, caso exista na consulta. Por exemplo, para obter o mesmo resultado do comando anterior, podemos usar o

c6digo a seguir:



Fonte: 0 autor

No entanto, vamos perceber que o uso mais conhecido da clausula GROUP BY ocorre quando associada a func;oes de agregac;ao, tais como COUNT, MIN, MAX e AVG.

Lima tabela com o nome e o significado dessas func;oes foi apresentada na sec;ao **"Func;oes de resumo ou agregac;ao"** no modulo 1.

Vamos estudar alguns exemplos?

CONSULTA 01 + RESULTADO

Retamar o numero de funcionarios par sexo.

Fonte: 0 autor

Fonte: 0 autor

***ii*** Resultado consulta 01.

0 SGBD realiza o agrupamento de dados de acordo com as valores da coluna SEXO. Em seguida, para cada grupo encontrado, a furn;ao COUNT(\*) e executada e o resultado exibido.

E se tivessemos interesse em exibir as resultados da consulta anterior em uma unica linha? Poderfamos usar o c6digo a seguir:



Fonte: 0 autor



Fonte: 0 autor.

**m** Numero de funcionarios par sexo: informac;;oes exibidas em uma unica linha.

CONSULTA 02 + RESULTADO

Retamar a media salarial par sexo.

Fonte: 0 autor

Fonte: 0 autor

***ii*** Resultado consulta 02.

0 SGBD realiza o agrupamento de dados de acordo com os valores da coluna SEXO. Em seguida, para cada grupo encontrado, a furn;ao AVG (SALARIO) e executada; e o resultado, exibido.

CONSULTA 03 + RESULTADO

Retornar, por mes de aniversario, a quantidade de colaboradores, o menor salario, o maior salario e o salario medio. Ordene os resultados por mes de aniversario.

Fonte: 0 autor

Fonte: 0 autor

**m** Resultado consulta 03.

0 SGBD realiza o agrupamento de dados de acordo com o mes de nascimento dos funcionarios. Depois, para cada grupo encontrado, as func;oes de agregac;ao sao executadas e, em seguida, exibidos os resultados. Perceba tambem que, na linha 4, utilizamos a func;ao ROUND com objetivo de mostrar ao usuario final somente a parte inteira dos valores resultantes da media salarial.

CONSULTA 04 + RESULTADO

Retornar, por mes de aniversario, o mes, o sexo e a quantidade de colaboradores. Apresentar os resultados ordenados pelo mes.

Fonte: 0 autor

Fonte: 0 autor

**m** Resultado consulta 04.

O SGBD realiza o agrupamento de dados de acordo com os valores do mes de aniversario. Em seguida, no contexto de cada mes encontrado, mais um grupo e construfdo por sexo. Finalmente, para cada ocorrencia mes/sexo, o numero de colaboradores e calculado.

### GRUPO DE DADOS COM GROUP BYE HAVING

Ate o momenta, utilizamos a clausula WHERE para programar filtros em consultas, com condig6es simples ou compostas envolvendo colunas da tabela ou func;6es de data.

Contudo, voce vai vivenciar situac;6es onde sera necessario estabelecer algum tipo de filtro, tendo coma base um calculo originado a partir de uma func;ao de agregac;ao, nao sendo possfvel usar a clausula WHERE. Nesses casos, utilizamos a clausula HAVING, que serve justamente para esse prop6sito.

Vamos ver a seguir um exemplo de quando utilizar essa clausula.

###### CONSULTA 05 + RESULTADO

Suponha que o departamento de recursos humanos esteja estudando a viabilidade de oferecer bonus de 5% aos funcionarios par mes de nascimento, mas limitado somente aos casos onde ha mais de um colaborador aniversariando. Assim, para cada mes em questao, deseja-se Iistar o mes, o numero de colaboradores e o valor do bonus.

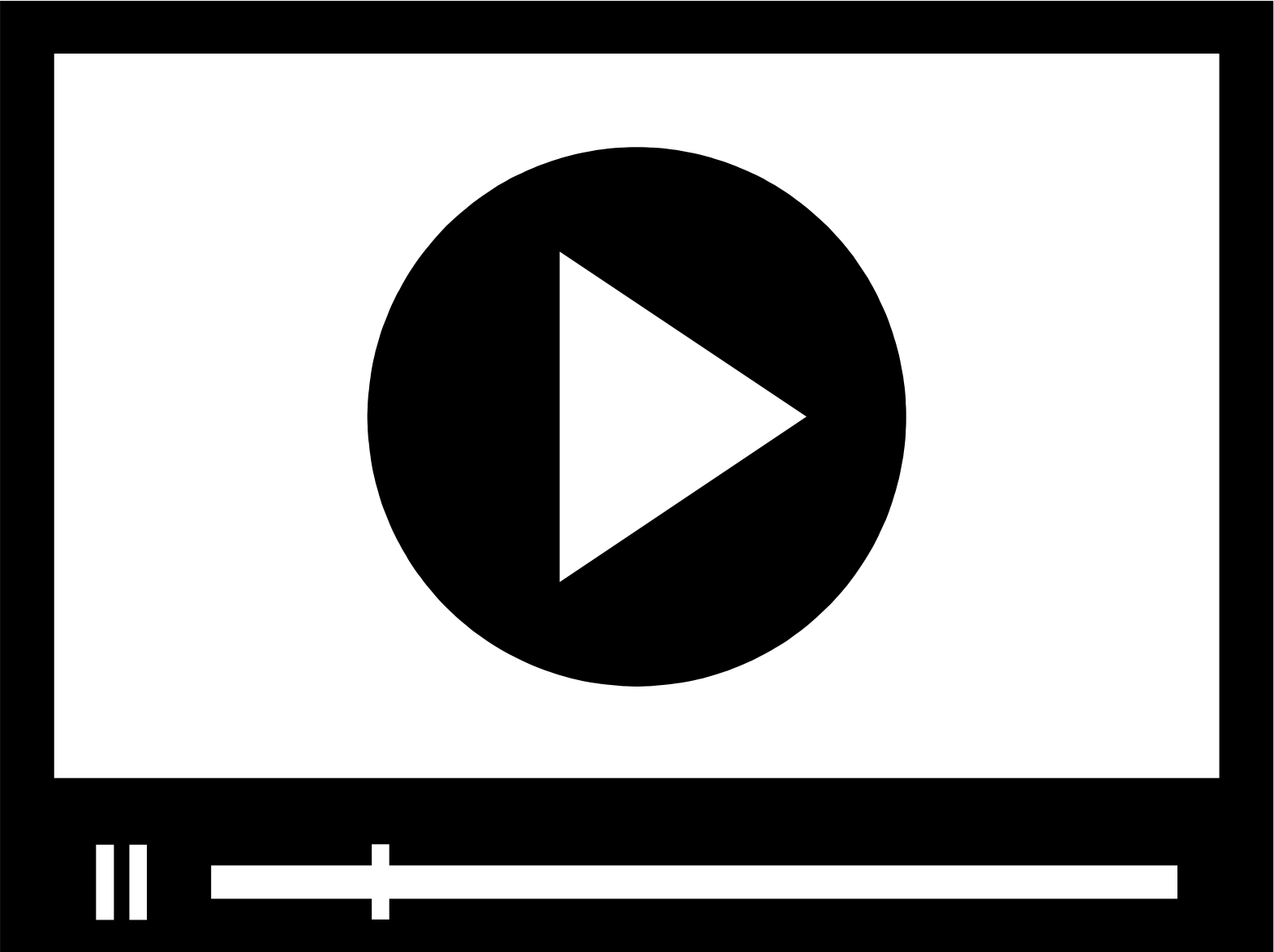
Soluc;ao:

Fonte: 0 autor

Fonte: 0 autor

Resultado consulta 05.

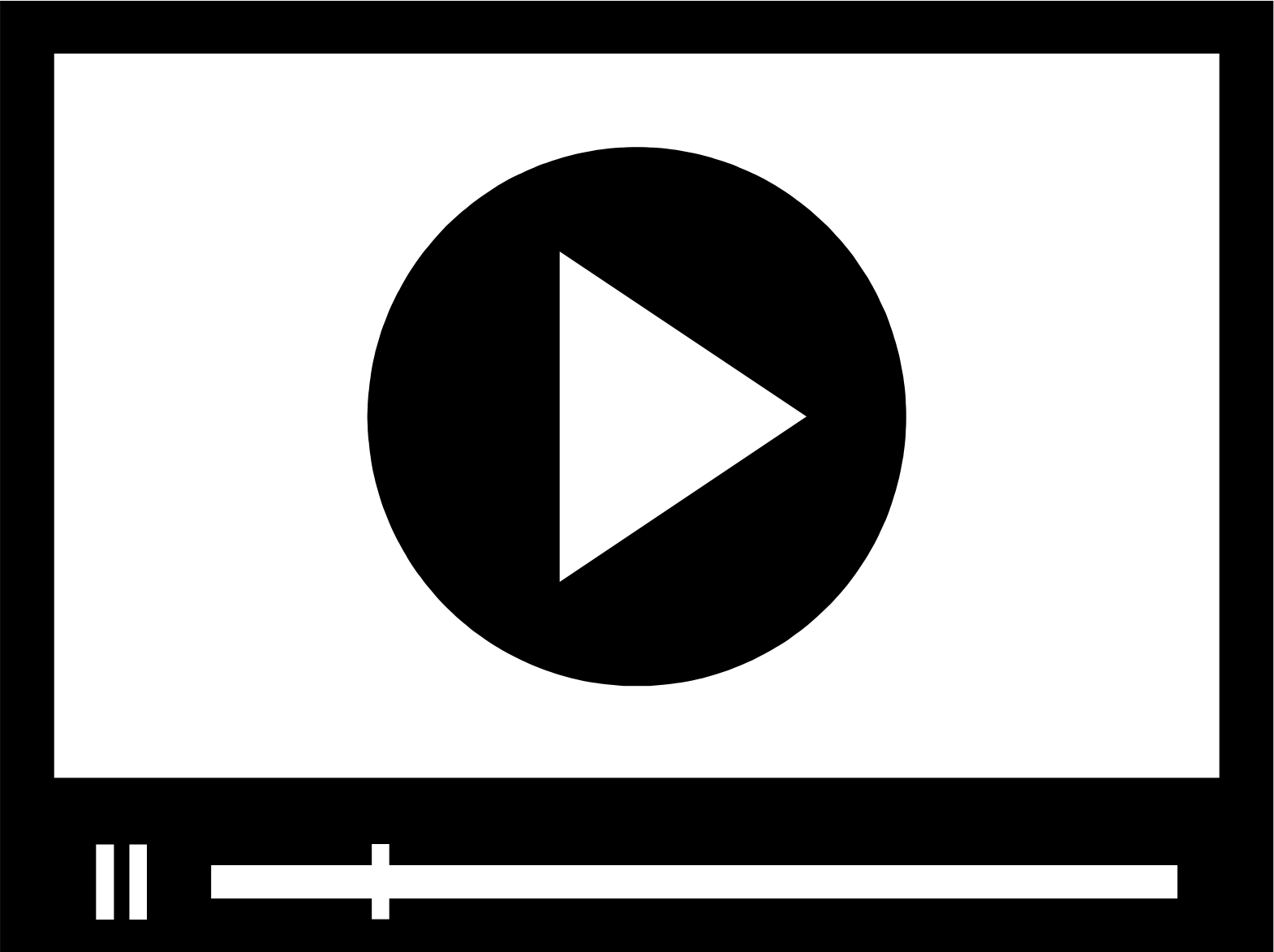
Note que estamos diante de uma estrutura de consulta muito similar ao c6digo da consulta 03. Porem, estamos interessados em retornar somente o(s) registro(s) cujo valor da coluna **quantidade** seja maior que a unidade. Acontece que **quantidade** e uma coluna calculada com aux0io de uma func;ao de agregac;ao, nao sendo possfvel programar um filtro na clausula WHERE (WHERE QUANTIDADE>1 ). Assim, declaramos o filtro de interesse fazendo uso da clausula HAVING, conforme linha 6 da consulta.



# CONSULTAS COMO COMANDO SELECT E AS CLAUSULAS GROUP BYE HAVING UTILIZANDO PGADMIN

Veja no vfdeo a seguir como realizar Estudos de casos com os comandos **GROUP BYe HAVING**





# CONSULTAS COMO COMANDO SELECT E AS CLAUSULAS GROUP BYE HAVING UTILIZANDO PLSQL

Veja no video a seguir como realizar **Consultas com o comando SELECT e as clausulas GROUP BYe HAVING**



Ao longo da nossa jornada, estudamos o projeto de consultas com o uso de agrupamento de dados. Percebemos que esse recurso e imprescindfvel quando temos interesse na extrac;ao de

informac;6es de carater mais analltico a partir de alguma tabela, fazendo uso de func;6es de agregac;ao associadas a uma ou diversas colunas.

Ainda, percebemos que, as vezes, a natureza do problema que estamos resolvendo requer o uso de filtro tendo como base o uso de alguma func;ao de agregac;ao. Para isso, fizemos uso da clausula HAVING.

Agora e com voce! Vamos realizar as atividades a seguir?

## VERIFICANDO O APRENDIZADO

1. SUPONHAQUE EXISTAEM UM BANCO DE DADOS UMATABELA DENOMINADA CLIENTE, ASSIM ESTRUTURADA: CLIENTE (CODIGOCLIENTE, NOME, SEXO, BAIRRO, RENDA). VOCE FOi SOLICITADOAESCREVER UM COMANDO SQL PARAOBTERARENDA MEDIA DOS CLIENTES POR BAIRRO.

**0 COMANDO CORRETO E:**

1. SELECT BAIRRO, MIN(RENDA) FROM CLIENTE

GROUP BY BAIRRO

1. SELECT SEXO, SUM(RENDA) FROM CLIENTE

GROUP BY BAIRRO

1. SELECT BAIRRO, AVG(RENDA) FROM CLIENTE

GROUP BY BAIRRO

1. SELECT BAIRRO,MAX(RENDA) FROM CLIENTE

GROUP BY SEXO

1. SUPONHAA EXISTENCIA DE UMA TABELA NO POSTGRESQL COM A SEGUINTE ESTRUTURA: PRODUTO (CODIGOP, NOME,ANO QUANTIDADE). SUPONHA TAMBEM QUE ATABELA TENHAOS SEGUINTES REGISTROS:

|  |  |  |  |
| --- | --- | --- | --- |
| CODIGO P | NOME | ANO | QUANTIDADE |
| 1 | VIRTUS | 2020 | 3 |
| 2 | FIESTA | 2014 | 1 |
| 3 | CRUZE | 2020 | 4 |
| 4 | CAMARO | 2018 | 1 |
| 5 | KOMBI | 1996 | 4 |
| 6 | FOCUS | 2016 | 3 |

ATENCAO! PARA VISUALIZACAOCOMPLETA DA TABELA UTILIZE A ROLAGEM HORIZONTAL

QUAL CONSULT AA SEGUIR RETORNA MAIS DE DOIS RESULTADOS?

1. SELECTANO,SUM(QUANTIDADE)AS TOTAL FROM PRODUTO

GROUP BYANO

HAVING SUM(QUANTIDADE)>1;

1. SELECT ANO,SUM(QUANTIDADE)AS TOTAL FROM PRODUTO

GROUP BY ANO;

1. SELECT SUM(QUANTIDADE)AS TOTAL FROM PRODUTO;
2. SELECT ANO, COUNT(\*) AS TOTAL FROM PRODUTO

WHERE QUANTIDADE>5 GROUP BY ANO;

GABARITO

1. **Suponha que exista em um banco de dados uma tabela denominada CLIENTE, assim estruturada: CLIENTE (CODIGOCLIENTE, NOME, SEXO, BAIRRO, RENDA). Voce foi**

**solicitado a escrever um comando SQL para obter a renda media dos clientes por bairro. 0 comando correto e:**

A alternativa **"C** " esta correta.

Para recuperar corretamente as registros de interesse, e necessario agrupar as dados pela coluna BAIRRO e em seguida usar a furn;ao de media (AVG), tendo coma base a coluna RENDA.

1. **Suponha a existencia de uma tabela no PostgreSQL com a seguinte estrutura: PRODUTO (CODIGOP, NOME, ANO QUANTIDADE). Suponha tambem que a tabela tenha os seguintes registros:**

|  |  |  |  |
| --- | --- | --- | --- |
| **CODIGO P** | **NOME** | **ANO** | **QUANTIDADE** |
| **1** | **VIRTUS** | **2020** | **3** |
| **2** | **FIESTA** | **2014** | **1** |

|  |  |  |  |
| --- | --- | --- | --- |
| **3** | **CRUZE** | **2020** | **4** |
| **4** | **CAMARO** | **2018** | **1** |
| **5** | **KOMBI** | **1996** | **4** |
| **6** | **FOCUS** | **2016** | **3** |

**Aten ao! Para visualiza aocompleta da tabela utilize a rolagem horizontal**

**Qual consulta a seguir retorna mais de dois resultados?**

A alternativa **"A"** esta correta.

Para recuperar os registros de interesse, a consulta em questao retorna o total de autom6veis por ano, no entanto, levando em conta somente os grupos em que o total seja maior que 1. Na pratica, os anos 2014 e 2018 nao farao parte dos resultados da consulta e os demais o farao, totalizando tres resultados.

# CONCLUSAO

**CONSIDERAC0ES FINAIS**

Neste tema, tratamos do comando SELECT da SQL no PostgreSQL. Vimos a sua sintaxe basica para consulta a uma tabela, no formato SELECT ... FROM ... WHERE.

Reconhecemos que, na clausula SELECT, sao especificadas as colunas da tabela a serem selecionadas, o que corresponde a operagao de projegao da Algebra Relacional.

Aprendemos que e possfvel especificar express6es e func;6es nesta clausula. No caso especffico do PostgreSQL, vimos que a execuc;ao de func;oes pre-definidas e realizada especificando o nome e os parametros da func;ao na clausula SELECT, omitindo as demais clausulas do

comando, inclusive a clausula FROM. Em seguida, estudamos o uso da clausula WHERE, que especifica a condic;ao de selec;ao de linhas da tabela, o que corresponde a operac;ao de restric;ao

ou selec;ao da Algebra Relacional. Por fim, aplicamos clausulas adicionais do comando SELECT, como ORDER BY, GROUP BYe HAVING, todas implementadas no PostgreSQL em compatibilidade com o padrao da linguagem SQL.





## REFERENCIAS

AWS. **Tarefas comuns do administrador de banco de dados para PostgreSQL.** *In:* AWS. Consultado em meio eletronico em: 30 mai. 2020.

BILECKI, L. F.; KALEMPA, V. C. **EasyRA:** Uma ferramenta para traduc;ao de consultas em algebra relacional para SQL. In: Computer On The Beach, 2015, Florian6polis. Computer on the Beach, 2015. p. 21-30.

CAMPOS, N. S. **Notas de Aula sobre Banco de Dados da professora Nathielly Campos.**

Disponivel sob licenc;a Creative Commons BRAtribuic;ao - CC BY, 2020.

ELMASRI, R.; NAVATHE, S. Sistemas de Banco de Dados. 7. ed. Sao Paulo: Pearson, 2019.

POSTGRESQL. **Chapter 9. Functions and Operators.** *In:* PostgreSQL. Consultado em meio eletr6nico em: 30 mai. 2020.

POSTGRESQL. **PostgreSQL Downloads.** *In:* PostgreSQL. Consultado em meio eletr6nico em: 30 mai. 2020.

POSTGRESQL. **SELECT.** *In:* PostgreSQL. Consultado em meio eletr6nico em: 30 mai. 2020.

## EXPLORE+

Para aprofundar os seus conhecimentos sabre o assunto deste tema, leia:

"Tarefas comuns do administrador de banco de dados para PostgreSQL", um interessante material sabre tarefas do dia a dia de um administrador de banco de dados que atue com o PostreSQL. Voce pode encontra-lo no site da Amazon Web Services.

BILECKI, L. F.; KALEMPA, V. C. **EasyRA: Uma ferramenta para tradu ao de consultas em algebra relacional para SQL.** *In:* Computer on the Beach, 2015. E importante saber que parte consideravel da linguagem SQL e baseada na teoria de Algebra Relacional.

Trata-se de uma algebra que envolve diversos operadores sabre relac;oes. Neste trabalho, voce aprendera sabre as operac;oes basicas da Algebra Relacional, e conhecera uma ferramenta para praticar comandos em algebra e visualizar o respectivo comando na linguagem SQL.

## CONTEUDISTA

Nathielly de Souza Campos

**c§> CURRiCULO LATTES**