Comando SELECT

* comando mais utilizado, toda query começa com select
* serve para selecionar as colunas de tabelas

select coluna\_1, coluna\_2, coluna\_3

from schema\_1.tabela\_1

Exemplo 1 : seleção de uma coluna de uma tabela

– liste os e-mails dos clientes da tabela sales.customers

select email

from sales.customers

* apos clicar para executar esse comando no query tools, ele vai fornecer todos os emails prontos para colocar em uma planilha excel

Exemplo 2 : seleção de mais de uma coluna de uma tabela

– Liste os emails e nomes dos clientes da tabela sales.customers

select email, first\_name, last\_name

from sales.customers

Exemplo 3: Seleção de todas as colunas de uma tabela

– liste todas as informações dos clientes da tabela sales.customers

select \*

from sales.customers

RESUMO

1. Comando usado para selecionar colunas de tabelas ( select )
2. Quando selecionar mais de uma coluna, elas devem ser separadas por vírgula sem conter vírgula antes do comando FROM
3. Pode-se utilizar o asterisco (\*) para selecionar todas as colunas da tabela

Comando DISTINCT

* serve para remover linhas duplicadas e mostrar apenas linhas distintas
* Muito usado na etapa de exploração de bases
* Sintaxe :

select distinct coluna\_1, coluna\_2, coluna\_3

from schema\_1.tabela\_1

Exemplo 1: Seleção de uma coluna sem Distinct

– liste as marcas de carro que constam na tabela products

select brand

from sales.products

* retorna 333 resultados repetidos

Exemplo 2: Seleção de uma coluna com Distinct

– liste as marcas de carro distintas que constam na tabela products

select distinct brand

from sales.products

* retorna 40 resultados sem repetição

Exemplo 3 : Seleção de mais de uma coluna com distinct

– Liste as marcas e anos de modelo distintos que constam na tabela products

select distinct brand, model\_year

from sales.products

* encontrou todas as combinações distintas nessas 2 colunas

Resumo

1. Comando Distinct é usado para remover linhas duplicadas e mostrar apenas linhas distintas.
2. Muito utilizado na etapa de exploração dos dados
3. caso mais de uma coluna seja selecionada, o comando SELECT DISTINCT irá

retornar todas as combinações distintas

Comando WHERE

* Serve para filtrar as linhas da tabela de acordo com uma condição
* trabalhando com dados em forma de texto devemos utilizar **aspas simples** e **CASE SENSITIVE**
* sintaxe

select coluna\_1, coluna\_2, coluna\_3

from schema\_1.tabela\_1

where condição\_x=true

simplificando:

1. selecionar as colunas que desejamos trazer na seleção
2. de qual tabela iremos trazer
3. e qual a condição verdadeira para a linha aparecer na seleção

Exemplo 1 : filtro com condição única

– Liste os emails dos clientes da nossa base que moram no estado de santa catarina

select email, state

from sales.customers

where state = ‘Santa catarina’

porém deu erro, precisamos ver como os dados estão inseridos:

select distinct state

from sales.customers

* os dados estão abreviados e em uppercase, então mudamos a sintaxe

select email, state

from sales.customers

where state = ‘SC’

Exemplo 2 : Filtro com mais de uma condição

– Liste os emails dos clientes da nossa base que moram no estado de Santa Catarina ou Mato grosso do sul

select email, state

from sales.customers

where state = ‘SC’ or state = ‘MS’

Exemplo 3 : Filtro de condição com data

– liste os emails dos clientes da nossa base que moram no estado de santa catarina ou mato grosso do sul e que tem mais de 30 anos

* verificar como estão os dados das datas

select distinct birth\_date

from sales.customers

select email, state, birth\_date

from sales.customers

where (state = ‘SC’ or state = ‘MS’) and birth\_date < ‘1993-06-28’

ou

select email, state, birth\_date

from sales.customers

where (state = ‘SC’ or state = ‘MS’) and birth\_date < ‘19930628’

Resumo:

1. Commando utilizado para filtrar linhas de acordo com uma condição
2. No PostgreSQL são utilizadas aspas simples para delimitar strings
3. string = sequência de caracteres = texto
4. pode-se combinar mais de uma condição utilizando os operadores lógicos
5. No PostgreSQL as datas são escritas no formato ‘YYYY-MM-DD’ ou ‘YYYYMMDD’

Comando ORDER BY

* serve para ordenar a seleção de acordo com uma regra definida pelo usuário

Exemplo 1 : Ordenação de valores numéricos

– liste produtos da tabela products na ordem crescente com base no preço :

select \*

from sales.products

order by price

para fazer na ordem decrescente:

select \*

from sales.products

order by price desc

Exemplo 2: Ordenação de texto

– Liste os estados distintos da tabela customers na ordem crescente

select distinct state

from sales.customers

order by state

Resumo

1. Comando utilizado para ordenar a seleção de acordo com uma regra definida
2. Por padrão o comando ordena na ordem crescente. Para mudar para decrescente devemos usar o Comando DESC
3. No caso de strings a ordenação seguirá a ordem alfabética

Comando LIMIT

* Serve para limitar o nº de linhas de consulta
* Muito utilizado na etapa de exploração de dados
* limit N(numero de linha)

Exemplo 1: Seleção das N primeiras linhas usando LIMIT

– liste as 10 primeiras linhas da tabela funnel

select \*

from sales.funnel

limit 10

Exemplo 2: Seleção das N primeiras linhas usando LIMIT e ORDER BY

– liste os 10 produtos mais caros da tabela products

select \*

from sales.products

order by price desc

limit 10

Resumo:

1. Comando utilizado para limitar o nº de linhas da consulta
2. Muito utilizado na etapa de exploração dos dados
3. Muito utilizado em conjunto com o comando ORDER BY quando o que importa são os TOP N. Ex: “N pagamentos mais recentes”, “N produtos mais caros”