Banco de Dados

CENTRO UNIVERSITÁRIO NEWTON PAIVA

Prof. Dr. João Paulo Aramuni









Banco de Dados

2° período

Prof. Dr. João Paulo Aramuni





Projeto Físico de Bando de Dados

Aula 12







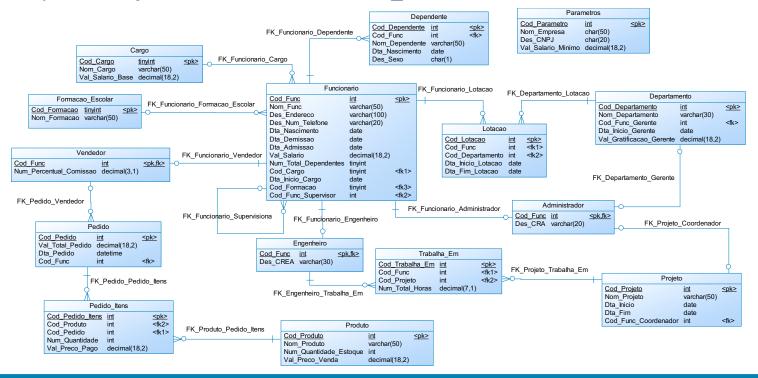
Divisão da Linguagem SQL





Vamos trabalhar com o BD do DER abaixo.

Execute o script de criação do banco de dados BD_RH.





- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.
- -- Para recuperar/consultar dados
- -- Sintaxe SQL básica para consultas em tabelas:

SELECT ...

FROM ...

WHERE ...

ORDER BY ...

OPERADORES BÁSICOS DE COMPARAÇÃO: =, <, <=, >, >=, <>

OPERADORES BÁSICOS LÓGICOS: AND, OR



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Exercícios:

- 1) Recupere o nome e data de nascimento dos funcionários com salario entre 2000 e 3000 mil reais. Ordene os dados pela idade dos funcionários em ordem decrescente.
- 2) Recupere o nome, data de nascimento dos dependentes do sexo feminimo nascidos após o dia 01/01/2000.
- 3) Recupere os pedidos realizados neste ano ou cujo valor total seja inferior a 2000 mil reais ordenados pelo código do pedido.



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADOR DISTINCT
OPERADOR BETWEEN
OPERADOR IS NULL E IS NOT NULL



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADOR **DISTINCT**

O operador **DISTINCT** é usado para filtrar os resultados de uma consulta, removendo registros duplicados das colunas selecionadas. Ele garante que apenas valores únicos sejam retornados na consulta.

Exemplo: Suponha que temos uma tabela "produtos" com uma coluna "categoria" e queremos listar todas as categorias únicas de produtos:

SELECT DISTINCT categoria **FROM** produtos;



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADOR BETWEEN

O operador **BETWEEN** é usado para filtrar resultados com base em um intervalo de valores. Ele permite selecionar valores dentro de um determinado intervalo, inclusive os limites.

Exemplo: Suponha que temos uma tabela "funcionarios" com uma coluna "salario" e queremos listar os funcionários com salário entre 3000 e 5000:

SELECT nome, salario FROM funcionarios WHERE salario BETWEEN 3000 AND 5000;



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES IS NULL e IS NOT NULL

Os operadores **IS NULL** e **IS NOT NULL** são usados para verificar se um valor em uma coluna é nulo ou não nulo, respectivamente.

Exemplo: Suponha que temos uma tabela "clientes" com uma coluna "telefone" que pode conter valores nulos para alguns clientes. Queremos listar os clientes que não têm número de telefone registrado:

SELECT nome **FROM** clientes **WHERE** telefone **IS NULL**;



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADOR IN E NOT IN OPERADOR LIKE SQL ALIAS Função DATEDIFF



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADOR IN E NOT IN

O operador **IN** é usado para filtrar resultados em uma consulta quando você deseja comparar um valor com uma lista de valores possíveis. Ele retorna resultados que correspondem a qualquer valor da lista especificada. O operador NOT IN faz o oposto, retornando resultados que não correspondem a nenhum valor da lista.

Exemplo com IN: Suponha que temos uma tabela "produtos" e queremos listar apenas os produtos com categorias específicas (por exemplo, "eletrônicos" e "vestuário"):

SELECT nome, categoria **FROM** produtos **WHERE** categoria IN ('eletrônicos', 'vestuário');



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADOR LIKE

O operador **LIKE** é usado para realizar buscas de padrões em dados de texto. Ele é frequentemente usado com os caracteres especiais "%" (corresponde a qualquer sequência de caracteres) e "_" (corresponde a um único caractere).

Exemplo com LIKE: Suponha que temos uma tabela "clientes" e queremos listar apenas os clientes cujo nome começa com "Jo":

SELECT nome **FROM** clientes **WHERE** nome **LIKE** 'Jo%';



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

SQL ALIAS

O SQL **Alias** permite dar um nome temporário a uma coluna ou a uma tabela em uma consulta. Ele é útil para tornar o resultado da consulta mais legível ou quando você precisa se referir a colunas por nomes diferentes.

Exemplo com SQL Alias: Suponha que temos uma tabela "funcionarios" e queremos listar os funcionários e seus salários, mas queremos dar nomes mais descritivos às colunas de resultado:

SELECT nome **AS** NomeFuncionario, salario **AS** SalarioMensal **FROM** funcionarios;



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Função **DATEDIFF**

A função **DATEDIFF** é usada para calcular a diferença entre duas datas em termos de unidades específicas (dias, meses, anos, etc.).

Exemplo com DATEDIFF: Suponha que temos uma tabela "pedidos" e queremos listar a quantidade de dias entre a data do pedido e a data de entrega planejada:

SELECT numero_pedido, **DATEDIFF**(data_entrega_planejada, data_pedido) AS dias_entrega FROM pedidos;



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Exercícios:

- 1) Mostre os dependentes que nasceram após o ano de 2000 que tenham silva no nome. Mostre os dependentes com maior idade primeiro.
- 2) Recupere os pedidos feitos há menos de cinco anos.
- 3) Recupere os funcionários admitidos há mais de cinco anos que não tenha supervisor e cujo salário esteja entre 2000 e 5000 mil reais. Ordene os dados mostrando primeiro os funcionários com menor idade.



• **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Exercícios:

- 4) Recupere os dependentes que terão 18 anos ou mais após o ano que vem.
- 5) Mostre o nome dos funcionários no cargo 1 que teria salário acima de 2000 reais caso seu salário atual fosse reajustado em 10 %. Exiba também qual seria o salário destes funcionários caso o salario fosse reajustado em 10%

Dica: Uso dos seguintes operadores aritméticos (*, /, -, +) na cláusula WHERE.



Obrigado!

joao.aramuni@newtonpaiva.br