

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RONDÔNIA  
Campus Ji-Paraná

# Linguagem de Consulta de Dados *Select*

PROFESSOR: JACKSON HENRIQUE

# Consulta de Dados

- ▶ Uma das principais funções de um banco de dados é a manipulação de dados inseridos nas tabelas;
- ▶ Um das principais funções utilizadas é o ***select***;
- ▶ O *select* serve para consultar qualquer registro inserido em uma tabela ou mais de uma tabela de forma personalizada;

# Consulta de Dados

- ▶ **Síntese:**

- ▶ **SELECT**

- ▶ atributo1, atributo2, ...

- ▶ **FROM**

- ▶ nome\_da\_tabela1, nome\_da\_tabela2, ...

- ▶ **WHERE**

- ▶ condição;

# Consulta de Dados

## ► Exemplos:

➤ `Select * from Cliente;`

► O \* serve para indicar que a consulta irá mostrar TODOS os atributos da tabelas com seus devidos registros;

# Consulta de Dados

- ▶ Também é possível escolher quais atributos serão mostrados no resultado da consulta;
- ▶ Exemplos:
  - **SELECT nome\_cli, email\_cli FROM Cliente;**
  - **SELECT idade\_cli FROM Cliente WHERE cod\_cli = 1;**
  - **SELECT nome\_cli FROM Cliente WHERE sexo\_cli = 'Masculino';**

# Select com Cláusula Where

- ▶ Exemplos:
- **SELECT nome\_cli, email\_cli FROM Cliente WHERE nome\_cli = 'José da Silva';**
- ▶ Se você analisar, o comando *select* segue a mesma lógica dos comandos *Update* e *Delete*;
- ▶ Uma vez que este comando utiliza uma condição (**WHERE**) para determinar qual registro da tabela será consultado;

# Select com Operador de Comparação

- ▶ Você também pode utilizar qualquer um dos operadores de comparação (<, <=, >, >=, <>, =) que você aprendeu a utilizar nas disciplinas de programação, pode ser utilizado para testar os valores dos atributos na cláusula WHERE;
- ▶ < : menor;
- ▶ > : maior;
- ▶ <= : menor e igual;
- ▶ >= : maior e igual;
- ▶ <> : diferente;

# Select com Operador de Comparação

## ▶ Exemplos:

- **SELECT** titulo\_fil FROM **filmes** WHERE preco\_fil < 3;
- **SELECT** titulo\_fil FROM **filmes** WHERE preco\_fil > 3;
- **SELECT** titulo\_fil FROM **filmes** WHERE preco\_fil >= 3;
- **SELECT** titulo\_fil FROM **filmes** WHERE preco\_fil <= 3;



# Select com Operador de Comparação

- ▶ Além dos operadores de comparação, podemos utilizar os operadores lógicos AND (E) e OR (OU) para fazermos combinações de condições na cláusula WHERE;
- ▶ O operador lógico AND deve ser utilizado quando se quer fazer uma consulta a registros que obedecem a todas as condições impostas;
- ▶ Já o operador lógico OR deve ser utilizado para exibir registros quando quaisquer das condições sejam atendidas, ou seja, pelo menos uma condição seja verdadeira. Analise os seguintes exemplos:

# Select com Operador de Comparação

## ► Exemplos:

- `SELECT titulo_fil FROM Filmes WHERE (preço_fil < 3) AND (situação_fil <> 'alugado');`
- `SELECT titulo_fil FROM Filmes WHERE ((genero_fil = 'romance') OR (genero_fil= 'animação'));`

# Select com Palavras Chaves

- ▶ Além dos operadores de comparação e dos operadores lógicos, podemos utilizar algumas palavras chaves na cláusula WHERE para facilitar a elaboração dos comandos, tais como:
- ▶ **BETWEEN** - Usado para verificar se o valor de um atributo está em um intervalo de valores. Especifica um intervalo a ser testado;

# Select com Between

## ► Exemplos:

- SELECT titulo\_fil FROM Filmes WHERE (cod\_fil **BETWEEN** 2 AND 20);
- SELECT nome\_fil FROM Clientes WHERE (dataNasc\_cli **BETWEEN** '1990-1-1' AND '2000-1-1');

# Select com Like

- ▶ **LIKE** - Utilizada para comparar cadeias de caracteres usando padrões de comparação para um ou mais caracteres. O coringa percentual % substitui zero, um ou mais caracteres e o coringa sublinha \_ substitui um único caractere;
- ▶ **Exemplo:**
  - ▶ SELECT nome\_cli FROM Clientes WHERE email\_cli **LIKE** '%gmail.com';

# Select com In

- ▶ **IN** - Usado para verificar se o valor de um atributo está em um conjunto de valores entre parênteses. Quando o valor for compatível com um dos valores do conjunto, o registro é exibido;
- ▶ **Exemplo:**
- ▶ `SELECT titulo_fil FROM Filmes WHERE genero_fil IN ('comedia', 'romance', 'acao');`

# Select com Is Null

- ▶ **IS NULL** – Usado para selecionar diretamente valores que sejam NULL (nulos) ou NOT NULL (não nulos);
- ▶ **Exemplo:**
  - SELECT nome\_cli FROM Clientes WHERE email\_cli **IS NULL**;
  - SELECT nome\_cli FROM Clientes WHERE email\_cli **IS NOT NULL**;

# Select com Order By

- ▶ O comando **ORDER BY** ordena os registros selecionados em uma consulta;
- ▶ **Exemplo:**
- *SELECT nome\_cli, email\_cli FROM Clientes ORDER BY nome\_cli;*
- *SELECT nome\_cli, email\_cli FROM Clientes ORDER BY nome\_cli DESC;*
- ▶ O **DESC** mostra na ordem **decrescente**;



# Select com Order By

- ▶ O comando pode ser usado para ordenar qualquer tipo de resultado, mesmo que tenha sido utilizado uma cláusula WHERE;
- ▶ **Exemplo:**
  - *SELECT nome\_cli, email\_cli FROM Clientes WHERE sexo\_cli = 'Masculino' **ORDER BY** nome\_cli;*

# Select com Group By

- ▶ O comando **GROUP BY** agrupa valores iguais em uma consulta;
- ▶ **Exemplo:**
  - *SELECT preco\_fil FROM Filmes **GROUP BY** preco\_fil;*
- ▶ **Exemplo:** Caso tenha 10 filmes com o **preço\_fil de 10,00** apenas **01 registro** com o valor 10,00 será mostrado;

# Select com Função

- ▶ O principal uso da cláusula GROUP BY ocorre em conjunto com funções de agregação. As funções de agregação retornam um único valor como resultado de um conjunto de valores de entrada. As principais funções de agregação usadas no *MySQL* são:
- ▶ **COUNT()** – Função que retorna a quantidade total de registros não-nulos de um atributo;
- ▶ **SUM()** – Função que retorna a soma dos valores de um atributo;
- ▶ **AVG()** – Função que retorna a média dos valores de um atributo;
- ▶ **MIN()** – Função que retorna o menor valor de um atributo;
- ▶ **MAX()** – Função que retorna o maior valor de um atributo.

# Select com COUNT()

- ▶ **Exemplos:**

- ▶ Calcular o número total de filmes cadastrados no banco de dados de uma Locadora. **Exemplo:**

- SELECT **COUNT**(titulo\_fil) FROM Filmes;

- ▶ Mostrar quantos títulos diferentes de filmes estão cadastrados no banco de dados de uma Locadora. **Exemplo:**

- SELECT **COUNT**(DISTINCT titulo\_Fime) FROM Filmes;

# Select com MIN() e MAX()

- ▶ Comando para selecionar os filmes com o **MENOR** preço de uma locadora. **Exemplo:**
  - SELECT **MIN**(preco\_fil) FROM Filmes;
- ▶ Comando para selecionar os filmes com o **MAIOR** preço de uma locadora. **Exemplo:**
  - SELECT **MAX**(preco\_fil) FROM Filmes;

# Select com SUM() e AVG()

- ▶ Comando para selecionar os filmes com o menor preço de uma locadora. Exemplo:
- ▶ `SELECT MIN(preco_fil), AVG(preco_fil), MAX(preco_fil) FROM Filmes;`
- ▶ Comando para selecionar os filmes com o menor preço de uma locadora. Exemplo:
- ▶ `SELECT MIN(preco_fil), AVG(preco_fil), MAX(preco_fil) FROM Filmes;`
- ▶ Obs: As funções SUM() e AVG() são utilizadas apenas em atributos do tipo numérico

# Exemplos de Problemas

- ▶ **Exemplos:**

- ▶ **1. *Quantos clientes do sexo feminino têm na locadora?***

- ▶ `SELECT sexo_cli, COUNT(sexo_cli) FROM Filmes GROUP BY sexo_cli;`

- ▶ **2. *Quantos exemplares de um mesmo filme estão cadastrados na locadora?***

- ▶ `SELECT fil_til, COUNT(fil_til) AS Unidades FROM filmes GROUP BY fil_fil;`

# Select com Múltiplas Tabelas

- ▶ Existem duas maneiras de consultas dados em múltiplas tabelas em um banco de dados;
  - ❑ **A comparação direta**
  - ❑ **A comparação com Join**
- ▶ Veremos primeiro como realizar a comparação direta de chaves;

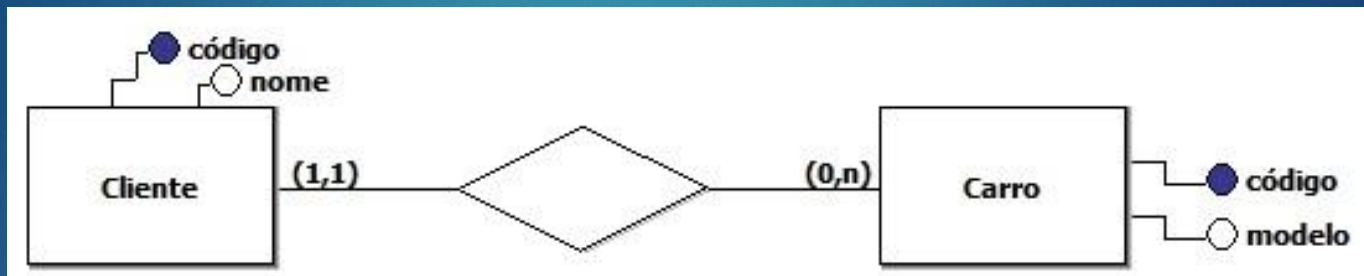


# Select com Múltiplas Tabelas

- ▶ Na comparação direta o que é comparado são os atributos comuns entre **02 tabelas** relacionadas;
- ▶ Ou seja, a **chave estrangeira** da tabela com **N** com a **chave primária** da tabela com o **1**;

# Exemplo

Veja este exemplo:



**Podemos afirmar que:**

- ▶ 1 Cliente está relacionado a Muitos carros;
- ▶ 1 Carro está relacionado a Um cliente;

**Logo:**

- ▶ Carro está com o N, então recebe a Chave Primária de Cliente que está com o 1;

# Exemplo com 02 Tabelas

- ▶ **Tarefa:** Selecione todos os registros de Carro, porém, substitua a Chave Estrangeira de Cliente pelo Nome do Cliente.

- ▶ **Solução:**

- ▶ **SELECT**

- ▶ CARRO.cód\_car,
- ▶ CARRO.modelo\_car,
- ▶ CLIENTE.nome\_cli

Selecione os atributos a serem mostrados, mas adicione o **nome da tabela** de origem do atributo

- ▶ **FROM**

- ▶ CLIENTE, CARRO

Informe quais tabelas estão sendo selecionadas

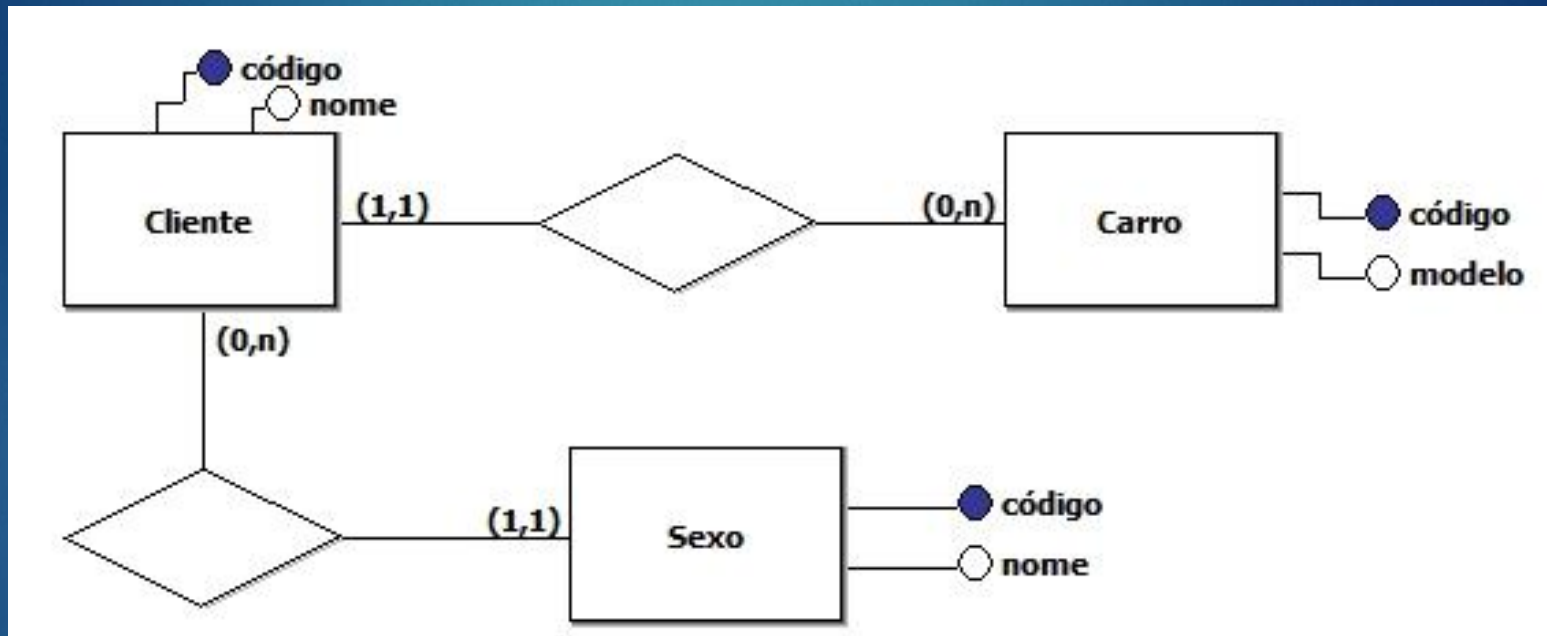
- ▶ **WHERE**

- ▶ (CARRO.cod\_cli = CLIENTE.cod\_cli);

Compare as **chaves comuns** entre as tabelas, ou seja, a chave **estrangeira** com a chave **primária**

# Exemplo com 03 Tabelas

Veja agora este **exemplo**:



**Tarefa:** Selecione o nome do cliente, nome do sexo e o modelo do seu carro.

# Exemplo com 03 Tabelas

- **SELECT**

- CLIENTE.nome\_cli,

- SEXO.nome\_sex,

- CARRO.nome\_car

- **FROM**


- CLIENTE, SEXO, CARRO

- **WHERE**

- (CARRO.cod\_cli = CLIENTE.cod\_cli) **AND**

- (CLIENTE.cod\_sex = SEXO.cod\_sex);

Comparação entre  
as FK e PK que une  
as 03 tabelas



# Exercício Avaliativo

- ▶ Com os novos conhecimentos sobre o **Select** faça a **Parte III** da Lista de Exercício II disponível no AVA;