330230 – Danimar Henrique Varisa

SQL Básico (MySQL)

1. Execute o comando SHOW DATABASES para verificar os bancos de dados existentes. Quais os esquemas extras que apareceram no resultado e que não são listados na lista dos esquemas do MySQL Workbench?

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. Uma imagem contendo Interface gráfica do usuário

   Descrição gerada automaticamenteRode o comando CREATE DATABASE dev\_ti\_2022 para criar o banco de dados. Atualize (dê um *refresh*) na lista de esquemas do Workbench e faça um *print* da tela colando-o abaixo.

Texto, Tabela

Descrição gerada automaticamente

1. Execute o comando USE apropriado para selecioná-lo. Confira para ver se o nome do banco está em negrito significando que está selecionado. Faça um *print* da tela do Workbench e cole abaixo.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. Use o comando SHOW TABLES para listar as tabelas existentes neste banco. A consulta não deve retornar nenhum resultado.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

1. Cria a tabela com o comando SQL abaixo:

**Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente**

1. Use o comando DESCRIBE pessoa para verificar a estrutura da tabela. Cole o *print* do resultado abaixo.

Tabela

Descrição gerada automaticamente

1. Insira na tabela os seguintes registros.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id\_pessoa | nome | endereco | estado | data\_nascimento | sexo | salario |
| 1 | André | Av.X, 10 | RJ | 1980-01-01 | m | 1.000,00 |
| 2 | Maria | Av.X, 50 | RJ | 1970-10-31 | f | 3.000,00 |
| 3 | Luis | Rua Y, 25 | PR | 1990-05-15 | m | 1.000,00 |
| 4 | Ana | Rua Y, 50 | PR | 2000-02-28 | f | 4.300,00 |
| 5 | Augusto | Av.Z, 80 | SC | 1995-08-17 | m | 8.600,00 |
| 6 | Ricardo | Av.X, 50 | PR | 2005-07-16 | m | 1.500,00 |
| 7 | Thais | Rua Z, 20 | SP | 1960-06-06 | f | 10.000,00 |
| 8 | Pedro | Rua Z, 100 | SP | 1980-09-05 | m | 8.900,00 |
| 9 | Flavia | Av.Z, 40 | SC | 1980-01-01 | f | 2.300,00 |
| 10 | Patrícia | Alameda W, 42 | SP | 1985-11-23 | f | 15.000,00 |

O primeiro registro pode ser inserido com o comando a seguir. Use-o como base para inserir todos os restantes.

INSERT INTO pessoa VALUES(null, 'André', 'Av.X, 10', 'RJ', '1980-01-01', 'm', 1000.00);

Tabela

Descrição gerada automaticamente

1. Modifique a data de nascimento de Ana para 7 de setembro de 1998 (use o campo id\_pessoa para fazer isso).

UPDATE pessoa SET data\_nascimento = '1998-09-07' WHERE id\_pessoa = 4;

Aplicativo, Tabela, Teams

Descrição gerada automaticamente

1. Modifique o salário de Pedro para 9 mil reais e o de Augusto para 8500 (use dois comandos diferentes).

UPDATE pessoa SET salario = 9000.00 WHERE id\_pessoa = 8;

UPDATE pessoa SET salario = 8500.00 WHERE id\_pessoa = 5;

1. Experimente agora inserir o registro abaixo:

INSERT INTO pessoa VALUES(null, 'Fulano', 'Rua F', '2000-01-01', 'p', 1000.00);

Foi possível inserir o registro? Se não, qual foi o motivo?

A coluna ID é do tipo auto\_increment e não pode ser nula ou vazia.

1. Tente inserir o registro abaixo:

INSERT INTO pessoa VALUES(null, 'Fulano', 'Rua F', '2000-01-01', 'm', -1000.00);

Foi possível inserir o registro? Se não, qual foi o motivo?

A coluna salario possui uma checagem que não permite inserir números menores que zero.

1. Faça agora as seguintes consultas e escreva os comandos utilizados:
2. Liste todos os registros e todas as colunas.

SELECT \* FROM pessoa;

1. Liste todos os registros, mas somente as colunas nome e salario.

SELECT nome, salario FROM pessoa;

1. Mostre somente os nomes em ordem crescente.

SELECT nome FROM pessoa ORDER BY nome ASC;

1. Mostre os nomes com respectivas datas de nascimento e salários em ordem decrescente de nascimento e, para nascimentos iguais, ordem crescente de salário.

SELECT nome, data\_nascimento, salario FROM pessoa ORDER BY data\_nascimento DESC, salario ASC;

1. Exiba o campo nome somente dos registros cujo sexo = 'f'.

SELECT nome FROM pessoa WHERE sexo = 'f';

1. Mostre a média salarial, agrupada por sexo e dê apelido de ‘média salarial’ para a coluna.

SELECT sexo, avg(salario) as 'Média salarial' FROM pessoa GROUP BY sexo;

1. Melhore a consulta anterior usando a função round() para arredondar o resultado para 2 dígitos.

SELECT sexo, ROUND(AVG(salario), 2) AS 'Média salarial' FROM pessoa GROUP BY sexo;

1. Melhore ainda mais a consulta mostrando o símbolo ‘R$’ (utilize a função concat()). Em vez da função round() use a função format() para formatar a média salarial no formato monetário.

SELECT sexo, concat('R$ ', format(avg(salario), 2, 'de\_DE')) AS 'Média salarial' FROM pessoa GROUP BY sexo;

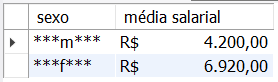
1. Agora, após aplicar a formatação monetária mas antes de concatenar com o ‘R$’ use a função lpad() para produzir uma saída de 20 caracteres preenchidas com espaços à esquerda, como mostrado abaixo.

Interface gráfica do usuário, Aplicativo, Tabela, Teams

Descrição gerada automaticamente

SELECT sexo, concat('R$ ', lpad(format(avg(salario), 2, 'de\_DE'), 20, ' ')) AS 'Média salarial' FROM pessoa GROUP BY sexo;

1. Finalmente, tente unir as funções lpad() e rpad() para centralizar o caractere do campo sexo, preenchendo com 3 asteriscos à direita e 3 à esquerda. Dê o apelido de ‘sexo’ para esta coluna.



SELECT rpad(lpad(sexo, 4, '\*'), 7, '\*') AS sexo, concat('R$ ', lpad(format(avg(salario), 2, 'de\_DE'), 20, ' ')) AS 'Média salarial' FROM pessoa GROUP BY sexo;

1. Utilize a cláusula HAVING para mostrar somente o sexo cuja média salarial é maior ou igual a 5 mil reais.

SELECT sexo, concat('R$ ', format(avg(salario), 2, 'de\_DE')) AS 'Média salarial' FROM pessoa GROUP BY sexo HAVING avg(salario) >= 5000;

1. Mostre o maior e o menor dos salários na mesma consulta e modifique o nome das colunas produzindo o resultado abaixo.

Tabela

Descrição gerada automaticamente com confiança média

SELECT max(salario) AS 'maior salário', min(salario) AS 'menor salário' FROM pessoa;

1. Liste a média dos salários por estado ordenando de forma decrescente por média salarial.

SELECT estado, concat('R$ ', format(avg(salario), 2, 'de\_DE')) AS 'Média salarial' FROM pessoa GROUP BY estado ORDER BY avg(salario) DESC;

1. Liste os estados em ordem decrescente de média salarial e cuja média dos salários seja maior ou igual a   
   R$ 5.000,00.

SELECT estado, concat('R$ ', format(avg(salario), 2, 'de\_DE')) AS 'média salarial' FROM pessoa GROUP BY estado HAVING avg(salario) >= 5000 ORDER BY avg(salario) DESC;

1. Mostre o nome e o estado das pessoas que moram na região Sul.

SELECT nome, estado FROM pessoa WHERE estado = 'SC' OR estado = 'PR';

1. Utilize a função year() para mostrar o nome, data de nascimento e estado de todas as pessoas que nasceram em 1980 ou antes (<=).

SELECT nome, estado, data\_nascimento FROM pessoa WHERE year(data\_nascimento) <= 1980;

1. Faça uma consulta que retorne o nome, data de nascimento e estado de todas as pessoas que nasceram em 1980 ou antes (<=) e que sejam de ‘SP’.

**Interface gráfica do usuário, Tabela

Descrição gerada automaticamente**

SELECT nome, estado, data\_nascimento FROM pessoa WHERE estado = 'SP' AND year(data\_nascimento) <= 1980;

1. Insira a pessoa com o comando a seguir.

**INSERT** **INTO** pessoa (nome, endereco)

**VALUES(**'Fulano', 'Rua dos Fulanos'**);**

Execute uma consulta que busque somente pelo nome 'Fulano' e mostre todos seus campos. Qual é o conteúdo dos campos data de nascimento, estado, sexo e salário? Por que o MySQL atribuiu esses valores aos campos?

Por que no INSERT foi informado apenas os campos nome e endereço.

1. Exclua o registro criado na questão anterior utilizando para isso o campo de identificação da pessoa.

DELETE FROM pessoa WHERE id\_pessoa = 11;

1. Mostre todas as pessoas que morem em ruas.

SELECT \* FROM pessoa WHERE endereco LIKE 'Rua %';

1. Mostre todas as pessoas que não morem em alamedas.

SELECT \* FROM pessoa WHERE endereco NOT LIKE 'Alameda %';

1. Utilize o operador BETWEEN para mostrar todas as pessoas nascidas entre 1º de novembro de 1985 e 15 de abril de 2000. Use a função date\_format() com a máscara '%d de %M de %Y' para formatar a saída. Dê o apelido de ‘nascimento’ a esse campo.

SELECT nome, date\_format(data\_nascimento, '%d de %M de %Y') as nascimento FROM pessoa WHERE data\_nascimento BETWEEN '1985-11-01' AND '2000-04-15';

1. O resultado da consulta anterior retornou as datas em inglês. Modifique a ‘localização’ do sistema para português do Brasil com o comando SET lc\_time\_names='pt\_BR' e refaça a consulta.

SET lc\_time\_names='pt\_BR'; SELECT nome, date\_format(data\_nascimento, '%d de %M de %Y') as nascimento FROM pessoa WHERE data\_nascimento BETWEEN '1985-11-01' AND '2000-04-15';

1. Utilize a função count() para retornar o número de registros que atenderam ao critério da consulta da questão anterior.

SELECT count(\*) FROM pessoa WHERE data\_nascimento BETWEEN '1985-11-01' AND '2000-04-15';

1. Faça uma consulta que liste todos os registros mas não deve mostrar nem as pessoas que se chamem ‘André’ e nem as pessoas que ganhem R$ 1.000,00. Quantos registros atenderam a esse critério?

SELECT \* FROM pessoa WHERE nome != 'André' AND salario != 1000.00;

1. Faça uma consulta que liste todos os registros exceto se for uma pessoa que ganhe R$ 1.000,00 e ao mesmo tempo se chame ‘André’. Quantos registros atenderam a esse critério?

SELECT \* FROM pessoa WHERE NOT (nome = 'André' AND salario = 1000.00);

1. Execute o comando para truncar (apagar todos os dados) da tabela pessoa. Dê um *refresh* e verifique que a tabela continua dentro do banco, mas está vazia.

TRUNCATE pessoa;

1. Agora remova realmente a tabela pessoa. Dê um *refresh* para se assegurar que a tabela não existe mais.

DROP TABLE pessoa;

1. Execute o comando que apaga o banco de dados dev\_ti\_2022

DROP DATABASE dev\_ti\_2022;