



CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

DISCIPLINA: BANCO DE DADOS

LISTA DE EXERCÍCIOS SOBRE LINGUAGEM SQL

TRABALHO INDIVIDUAL

PARTE 1 – CRIAÇÃO DA BASE DE DADOS E INSERÇÃO DE DADOS NO SGBD MYSQL.

Siga as etapas:

ETAPA 1: Acessar a ferramenta MYSQL Command Line Cliente, ou use o Prompt de comando do Windows/Linux (exemplo: [link para guia comandos-basicos-mysql-no-terminal](#))

ETAPA 2: Criar a base de dados:

```
create database bdlista1;
```

ETAPA 3: acessar a base de dados:

```
use bdlista1;
```

ID_NF	ID_ITEM	COD_PROD	VALOR_UNIT	QUANTIDADE	DESCONTO (%)
1	1	1	25,00	10	5
1	2	2	13,50	3	
1	3	3	15,00	2	
1	4	4	10,00	1	
1	5	5	30,00	1	
2	1	3	15,00	4	
2	2	4	10,00	5	
2	3	5	30,00	7	
3	1	1	25,00	5	
3	2	4	10,00	4	
3	3	5	30,00	5	
3	4	2	13,50	7	
4	1	5	30,00	10	15
4	2	4	10,00	12	5
4	3	1	25,00	13	5
4	4	2	13,50	15	5
5	1	3	15,00	3	
5	2	5	30,00	6	
6	1	1	25,00	22	15
6	2	3	15,00	25	20
7	1	1	25,00	10	3
7	2	2	13,50	10	4
7	3	3	15,00	10	4
7	4	5	30,00	10	1

figura 1 - tabela vendas

Exercício 1: Crie um banco de dados chamado bdlista1, contendo uma tabela chamada vendas, contendo as colunas conforme figura 1. Não é necessário criar nenhuma chave primária para essa tabela. **Inserir abaixo o comando SQL DDL de criação da tabela (comando**

create table);

R.: create table vendas(id_nf int not null, id_item int not null, cod_prod int not null, valor_unit dec(10, 2) not null, quantidade int, desconto int);

Exercício 2: Inserir na tabela Vendas os dados da figura 1. Os valores em branco devem ser nulos no banco de dados (NULL). **Inserir abaixo os comandos SQL DML de insert into vendas values (...); São 24 linhas de dados, então deve-se ter 24 comandos: Insert into vendas values (...);**

R.:

```
insert into vendas values(1,1,1,25.00,10,5);
insert into vendas values (1,2,2,13.50,3,null);
insert into vendas values (1,3,3,15.00,2,null);
insert into vendas values (1,4,4,10.00,1,null);
insert into vendas values (1,5,5,30.00,1,null);
insert into vendas values (2,1,3,15.00,4,null);
insert into vendas values (2,2,4,10.00,5,null);
insert into vendas values (2,3,5,30.00,7,null);
insert into vendas values (3,1,1,25.00,5,null);
insert into vendas values (3,2,4,10.00,4,null);
insert into vendas values (3,3,5,30.00,5,null);
insert into vendas values (3,4,2,13.50,7,null);
insert into vendas values (4,1,5,30.00,10,15);
insert into vendas values (4,2,4,10.00,12,5);
insert into vendas values (4,3,1,25.00,13,5);
insert into vendas values (4,4,2,13.50,15,5);
insert into vendas values (5,1,3,15.00,3,null);
insert into vendas values (5,2,5,30.00,6,null);
insert into vendas values (6,1,1,25,22,15);
insert into vendas values (6,2,3,15,25,20);
insert into vendas values (7,1,1,25,10,3);
insert into vendas values (7,2,2,13.5,10,4);
insert into vendas values (7,3,3,15,10,4);
insert into vendas values (7,4,5,30,10,1);
```

PARTE 2 – COMANDOS DE SELEÇÃO DE DADOS NO SGBD MYSQL.

INSTRUÇÕES: Para os exercícios a seguir elabore as consultas SQL utilizando o comando SQL DML Select e insira os comandos na resposta. Alguns exercícios já estão respondidos, para fins de exemplo.

Exercício 3: utilizando o comando SELECT, pesquise todas os itens de venda, da nota fiscal 7 (ID_NF = 7). Devem aparecer no resultado todas as colunas da tabela.

R.: select * from vendas where id_nf = 7;

Exercício 4: utilizando o comando SELECT, pesquise todas os itens de venda, da nota fiscal 7 (ID_NF = 7), mas ordene o resultado por Valor Unitário (coluna VALOR_UNIT) de forma decrescente (do Maior p/ o menor).

R.: select * from vendas where id_nf = 7 order by valor_unit desc;

Exercício 5: utilizando o comando SELECT, pesquise todas os itens de venda, da nota fiscal 7 (ID_NF = 7), mas ordene o resultado por Valor Unitário (coluna VALOR_UNIT) de forma decrescente (do Maior p/ o menor). As colunas presentes no resultado da consulta são: ID_NF, ID_ITEM, COD_PROD E VALOR_UNIT.

R.: select id_nf, id_item, cod_prod, valor_unit from vendas where id_nf =7 order by valor_unit desc;

Exercício 6: utilizando a função MAX, mostre o produto que tem o valor unitário maior (coluna VALOR_UNIT) entre todos os produtos da tabela.

R.: select * from vendas where valor_unit = (select max(valor_unit) from vendas);

Exercício 7: Mostre todas as Nota fiscais (mostra no resultado somente a coluna ID_NF) diferente de 7. Deve-se eliminar dados repetidos (utilize a clausula *distinct*).

R.: select distinct id_nf as nota_fiscal from vendas where id_nf != 7;

Exercício 8: utilizando o comando SELECT, pesquise todas os itens de venda, da nota fiscal 1 (ID_NF = 1) e que tenha quantidade maior ou igual a 3, ordene o resultado por Quantidade de forma crescente. As colunas presentes no resultado da consulta são: ID_NF, ID_ITEM, COD_PROD E QUANTIDADE.

R.: select id_nf, id_item, cod_prod, quantidade from vendas where id_nf = 1 and quantidade >= 3 order by quantidade;

Exercício 9: Pesquise itens que foram vendidos sem desconto. As colunas presentes no resultado da consulta são: ID_NF, ID_ITEM, COD_PROD E VALOR_UNIT.

R.: select id_nf, id_item, cod_prod, valor_unit from vendas where desconto is null;

Exercício 10: Pesquise o total de cada Item ($\text{VALOR_UNIT} * \text{QUANTIDADE}$), desconsiderando o desconto. As colunas presentes no resultado da consulta são: ID_NF, ID_ITEM, VALOR_UNIT, QUANTIDADE E VALOR_TOTAL_ITEM.

R.: `select id_nf, id_item, valor_unit, quantidade, (valor_unit * quantidade) as valor_total_item from vendas order by id_nf, id_item;`

RESPOSTA_Professor: `Select ID_NF, ID_ITEM, VALOR_UNIT, QUANTIDADE, VALOR_UNIT*QUANTIDADE AS VALOR_TOTAL_ITEM from vendas;`

Exercício 11: Utilizando o comando `update`, altere o valor do desconto para zero de todos os registros onde este campo é nulo (NULL). Só para avisar, embora este exercício esteja na parte 2 (referente a seleção de dados) o mesmo é referente a alteração de dados, utilizando o comando `update`.

R.: `update vendas set desconto = 0 where desconto is null;`

Exercício 12: Pesquise os itens que foram vendidos com desconto. As colunas presentes no resultado da consulta são: ID_NF, ID_ITEM, COD_PROD, VALOR_UNIT E O VALOR VENDIDO. OBS: O valor vendido é igual ao $\text{VALOR_UNIT} - (\text{VALOR_UNIT} * (\text{DESCONTO} / 100))$.

R.: `select id_nf, id_item, cod_prod, valor_unit, round(valor_unit - (valor_unit * (desconto / 100)), 2) as valor_vendido from vendas;`

RESPOSTA_Professor: `Select ID_NF, ID_ITEM, COD_PROD, VALOR_UNIT, VALOR_UNIT - (VALOR_UNIT*(DESCONTO/100)) AS VALOR_VENDIDO from vendas;`

Exercício 13: Pesquise o valor final de cada item vendido. As colunas presentes no resultado da consulta são: ID_NF, ID_ITEM, COD_PROD, VALOR_UNIT, DESCONTO, VALOR_VENDIDO, VALOR_FINAL_ITEM. OBS: O VALOR_VENDIDO é igual ao $\text{VALOR_UNIT} - (\text{VALOR_UNIT} * (\text{DESCONTO} / 100))$ AS VALOR_VENDIDO. O VALOR_FINAL_ITEM é obtido pela fórmula: $\text{QUANTIDADE} * (\text{VALOR_UNIT} - (\text{VALOR_UNIT} * (\text{DESCONTO} / 100)))$ AS VALOR_FINAL_ITEM.

R.: `select id_nf, id_item, cod_prod, valor_unit, desconto, round((valor_unit - (valor_unit*(desconto/100))), 2) as valor_vendido, round((quantidade * (valor_unit - (valor_unit*(desconto/100))), 2) as valor_total_item from vendas;`