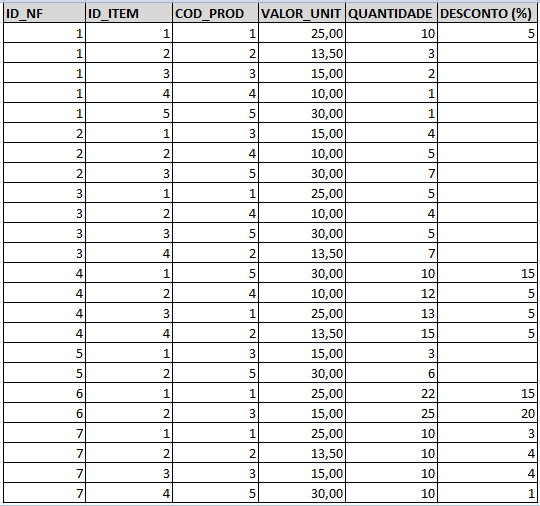
**Lista de Exercicios – Liguagem de banco de dados**

**Aluno: Leticia Araujo Costa – Ciencia de dados 3° semestre**

1. Crie, no seu banco de dados, a tabela abaixo, insira os valores apresentados e em seguida escreva as consultas solicitadas abaixo.

OBS: Os valores em branco devem ser nulos no banco de dados.



**create database venda; use venda;**

**set sql\_safe\_updates =0;**

**CREATE TABLE vendas ( id\_nf int not null, id\_item smallint not null, cod\_prod int not null, valor\_unit float not null, quantidade int null,**

**desconto float null,**

**primary key (id\_nf, id\_item)**

**)**

**engine = innodb;**

**INSERT INTO `venda`.`vendas` (`id\_nf`,**

**`id\_item`,**

**`cod\_prod`,**

**`valor\_unit`,**

**`quantidade`,**

**`desconto`**

**)**

**VALUES**

**(1, 1, 1, 25.00, 10, 5),**

**(1, 2, 2, 13.50, 3, null),**

**(1, 3, 3, 15.00, 2, null),**

**(1, 4, 4, 10.00, 1, null),**

**(1, 5, 5, 30.00, 1, null),**

**(2, 1, 3, 15.00, 4, null),**

**(2, 2, 4, 10.00, 15, null),**

**(2, 3 , 5, 30.00, 7, null),**

**(3, 1, 1, 25.00, 5, null),**

**(3, 2, 4, 10.00, 4, null),**

**(3, 3, 5, 30.00, 5, null),**

**(3, 4, 2, 13.50, 7, null),**

**(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, null),**

**(5, 2, 5, 30.00, 6, null),**

**(6, 1, 1, 25.00, 22, 15),**

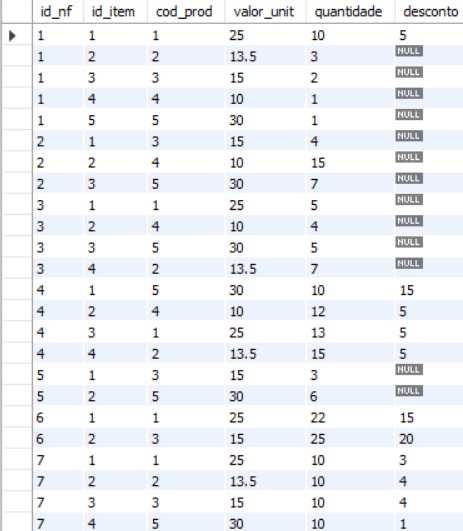
**(6, 2, 3, 15.00, 25, 20),**

**(7, 1, 1, 25.00, 10, 3),**

**(7,2, 2, 13.50, 10, 4),**

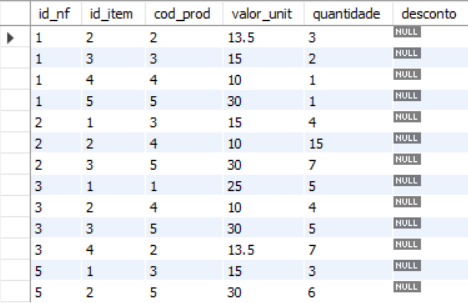
**(7, 3, 3, 15.00, 10, 4),**

**(7, 4, 5, 30.00, 10, 1);**



1. Pesquise os itens que foram vendidos sem desconto. As colunas presentes no resultado da consulta são: ID\_NF, ID\_ITEM, COD\_PROD E VALOR\_UNIT.

SELECT (id\_nf), (id\_item), (cod\_prod), (valor\_unit), (quantidade), (desconto) From vendas where desconto is null;

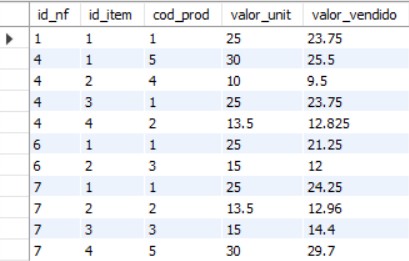


1. Pesquise os itens que foram vendidos com desconto. As colunas presentes no resultado da consulta são: ID\_NF, ID\_ITEM, COD\_PROD, VALOR\_UNIT E O VALOR VENDIDO. OBS: O valor vendido é igual ao VALOR\_UNIT -(VALOR\_UNIT\*(DESCONTO/100)).

SELECT id\_nf, id\_item, cod\_prod, valor\_unit, (valor\_unit - (valor\_unit \* (desconto / 100))) AS valor\_vendido

FROM vendas

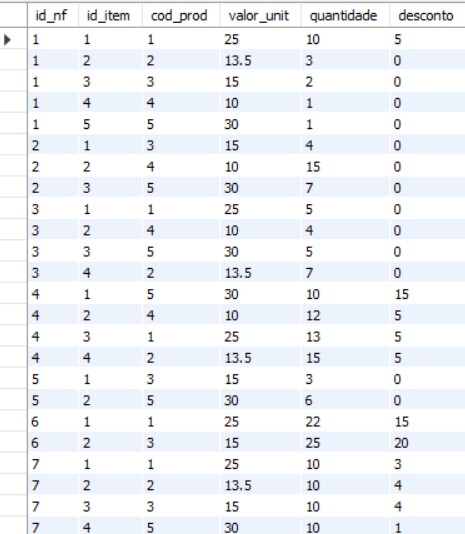
WHERE desconto IS NOT NULL;



1. Altere o valor do desconto (para zero) de todos os registros onde este campo é nulo.

UPDATE vendas SET desconto = 0

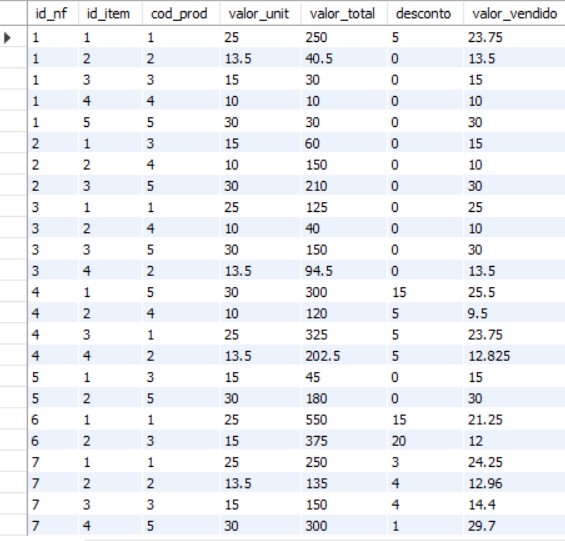
WHERE desconto IS NULL;

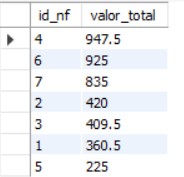


1. Pesquise os itens que foram vendidos. As colunas presentes no resultado da consulta são: ID\_NF, ID\_ITEM, COD\_PROD, VALOR\_UNIT, VALOR\_TOTAL, DESCONTO, VALOR\_VENDIDO. OBS: O VALOR\_TOTAL é obtido pela fórmula: QUANTIDADE \* VALOR\_UNIT. O VALOR\_VENDIDO é igual a VALOR\_UNIT - (VALOR\_UNIT\*(DESCONTO/100)).

select id\_nf, id\_item, cod\_prod, valor\_unit, (quantidade \* valor\_unit) AS valor\_total, desconto,

(valor\_unit - (valor\_unit \*(desconto / 100))) AS valor\_vendido From vendas;



1. Pesquise o valor total das NF e ordene o resultado do maior valor para o menor. As colunas presentes no resultado da consulta são: ID\_NF, VALOR\_TOTAL. OBS: O VALOR\_TOTAL é obtido pela fórmula: ∑ QUANTIDADE \* VALOR\_UNIT. Agrupe o resultado da consulta por ID\_NF.

SELECT id\_nf, SUM(quantidade \* valor\_unit) AS valor\_total

FROM vendas GROUP BY id\_nf

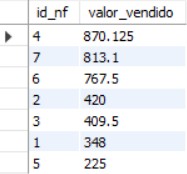
ORDER BY valor\_total DESC;

1. Pesquise o valor vendido das NF e ordene o resultado do maior valor para o menor. As colunas presentes no resultado da consulta são: ID\_NF, VALOR\_VENDIDO. OBS: O VALOR\_TOTAL é obtido pela fórmula: ∑ QUANTIDADE \* VALOR\_UNIT. O VALOR\_VENDIDO é igual a ∑ VALOR\_UNIT - (VALOR\_UNIT\*(DESCONTO/100)). Agrupe o resultado da consulta por ID\_NF.

select

id\_nf,

sum((valor\_unit \* quantidade) - (valor\_unit \* (desconto / 100) \* quantidade)) as valor\_vendido from

vendas group by

id\_nf order by

valor\_vendido desc;

1. Consulte o produto que mais vendeu no geral. As colunas presentes no resultado da consulta são: COD\_PROD, QUANTIDADE. Agrupe o resultado da consulta por COD\_PROD.

select cod\_prod,

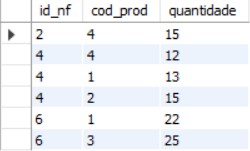
sum(quantidade) as quantidade from

vendas group by

cod\_prod order by

quantidade desc limit 1;

1. Consulte as NF que foram vendidas mais de 10 unidades de pelo menos um produto. As colunas presentes no resultado da consulta são: ID\_NF, COD\_PROD, QUANTIDADE. Agrupe o resultado da consulta por ID\_NF, COD\_PROD.

select

id\_nf, cod\_prod,

sum(quantidade) as quantidade from

vendas where

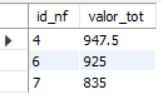
quantidade > 10 group by

id\_nf, cod\_prod;

1. Pesquise o valor total das NF, onde esse valor seja maior que 500, e ordene o resultado do maior valor para o menor. As colunas presentes no resultado da consulta são: ID\_NF, VALOR\_TOT. OBS: O VALOR\_TOTAL é obtido pela fórmula: ∑ QUANTIDADE

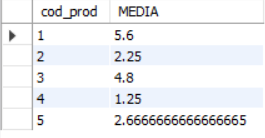
\* VALOR\_UNIT. Agrupe o resultado da consulta por ID\_NF.

SELECT id\_nf, SUM(quantidade \* valor\_unit) AS valor\_tot FROM vendas

GROUP BY id\_nf HAVING valor\_tot > 500

ORDER BY valor\_tot DESC;

1. Qual o valor médio dos descontos dados por produto. As colunas presentes no resultado da consulta são: COD\_PROD, MEDIA. Agrupe o resultado da consulta por COD\_PROD.

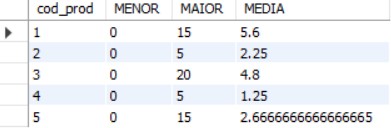


SELECT cod\_prod, AVG(desconto) AS MEDIA FROM vendas

WHERE desconto IS NOT NULL GROUP BY cod\_prod;

1. Qual o menor, maior e o valor médio dos descontos dados por produto. As colunas presentes no resultado da consulta são: COD\_PROD, MENOR, MAIOR, MEDIA. Agrupe o resultado da consulta por COD\_PROD.

SELECT

cod\_prod,

MIN(desconto) AS MENOR, MAX(desconto) AS MAIOR, AVG(desconto) AS MEDIA

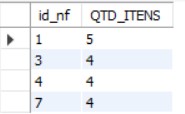
FROM

vendas WHERE

desconto IS NOT NULL GROUP BY

cod\_prod;

1. Quais as NF que possuem mais de 3 itens vendidos. As colunas presentes no resultado da consulta são: ID\_NF, QTD\_ITENS. OBS:: NÃO ESTÁ RELACIONADO A QUANTIDADE VENDIDA DO ITEM E SIM A QUANTIDADE DE ITENS POR NOTA FISCAL. Agrupe o resultado da consulta por ID\_NF.

SELECT

id\_nf,

COUNT(id\_item) AS QTD\_ITENS FROM

vendas GROUP BY

id\_nf HAVING

COUNT(id\_item) > 3;

1. Crie uma base de dados Universidade com as tabelas a seguir: Alunos (MAT, nome, endereço, cidade)

Disciplinas (COD\_DISC, nome\_disc, carga\_hor)

Professores (COD\_PROF, nome, endereço, cidade)

Turma (COD\_DISC, COD\_TURMA, COD\_PROF, ANO, horário) COD\_DISC referencia Disciplinas

COD\_PROF referencia Professores

Histórico (MAT, COD\_DISC, COD\_TURMA, COD\_PROF, ANO, frequência, nota) MAT referencia Alunos

COD\_DISC, COD\_TURMA, COD\_PROF, ANO referencia Turma

INSIRA OS SEGUINTES REGISTROS:

ALUNOS:

(2015010101, JOSE DE ALENCAR, RUA DAS ALMAS, NATAL) (2015010102, JOÃO JOSÉ, AVENIDA RUY CARNEIRO, JOÃO PESSOA) (2015010103, MARIA JOAQUINA, RUA CARROSSEL, RECIFE) (2015010104, MARIA DAS DORES, RUA DAS LADEIRAS, FORTALEZA)

(2015010105, JOSUÉ CLAUDINO DOS SANTOS, CENTRO, NATAL) (2015010106, JOSUÉLISSON CLAUDINO DOS SANTOS, CENTRO, NATAL)

DISCIPLINAS:

(BD, BANCO DE DADOS, 100)

(POO, PROGRAMAÇÃO COM ACESSO A BANCO DE DADOS, 100) (WEB, AUTORIA WEB, 50)

(ENG, ENGENHARIA DE SOFTWARE, 80)

PROFESSORES:

(212131, NICKERSON FERREIRA, RUA MANAÍRA, JOÃO PESSOA) (122135, ADORILSON BEZERRA, AVENIDA SALGADO FILHO, NATAL) (192011, DIEGO OLIVEIRA, AVENIDA ROBERTO FREIRE, NATAL)

TURMA:

(BD, 1, 212131, 2015, 11H-12H)

(BD, 2, 212131, 2015, 13H-14H)

(POO, 1, 192011, 2015, 08H-09H)

(WEB, 1, 192011, 2015, 07H-08H)

(ENG, 1, 122135, 2015, 10H-11H)

HISTÓRICO:

INSIRA VALORES PARA TODOS OS ALUNOS EM TODAS AS DISCIPLINAS

**create database universidade; use universidade;**

**CREATE TABLE Alunos (**

**MAT VARCHAR(100) NOT NULL, nome VARCHAR(100) NOT NULL,**

**endereco VARCHAR(255), cidade VARCHAR(100), PRIMARY KEY (MAT)**

**);**

**CREATE TABLE Disciplinas (**

**COD\_DISC VARCHAR(100) NOT NULL,**

**nome\_disc VARCHAR(100) NOT NULL, carga\_hor INT NOT NULL,**

**PRIMARY KEY (COD\_DISC)**

**);**

**CREATE TABLE Professores (**

**COD\_PROF VARCHAR(100) NOT NULL, nome VARCHAR(100) NOT NULL,**

**endereco VARCHAR(255), cidade VARCHAR(100), PRIMARY KEY (COD\_PROF)**

**);**

**CREATE TABLE Turma (**

**COD\_DISC VARCHAR(10) NOT NULL, COD\_TURMA INT NOT NULL, COD\_PROF VARCHAR(10) NOT NULL, ANO INT NOT NULL,**

**horario VARCHAR(20),**

**PRIMARY KEY (COD\_DISC, COD\_TURMA, ANO),**

**FOREIGN KEY (COD\_DISC) REFERENCES Disciplinas(COD\_DISC), FOREIGN KEY (COD\_PROF) REFERENCES Professores(COD\_PROF)**

**);**

**CREATE TABLE Historico (**

**MAT VARCHAR(100) NOT NULL, COD\_DISC VARCHAR(100) NOT NULL, COD\_TURMA INT,**

**COD\_PROF VARCHAR(10) NOT NULL, ANO INT,**

**frequência DECIMAL(5, 2),**

**nota DECIMAL(5, 2),**

**PRIMARY KEY (MAT, COD\_DISC, COD\_TURMA), FOREIGN KEY (MAT) REFERENCES Alunos(MAT),**

**FOREIGN KEY (COD\_DISC) REFERENCES Disciplinas(COD\_DISC), FOREIGN KEY (COD\_TURMA) REFERENCES Turma(COD\_TURMA),**

**FOREIGN KEY (COD\_PROF) REFERENCES Professores(COD\_PROF)**

**);**

**CREATE INDEX idx\_cod\_turma ON Turma(COD\_TURMA);**

**ALTER TABLE Historico**

**ADD CONSTRAINT fk\_cod\_turma FOREIGN KEY (COD\_TURMA) REFERENCES Turma(COD\_TURMA);**

**INSERT INTO Alunos (MAT, nome, endereco, cidade) VALUES ('2015010101', 'JOSE DE ALENCAR', 'RUA DAS ALMAS', 'NATAL'), ('2015010102', 'JOÃO JOSÉ', 'AVENIDA RUY CARNEIRO', 'JOÃO PESSOA'), ('2015010103', 'MARIA JOAQUINA', 'RUA CARROSSEL', 'RECIFE'), ('2015010104', 'MARIA DAS DORES', 'RUA DAS LADEIRAS', 'FORTALEZA'), ('2015010105', 'JOSUÉ CLAUDINO DOS SANTOS', 'CENTRO', 'NATAL'), ('2015010106', 'JOSUÉLISSON CLAUDINO DOS SANTOS', 'CENTRO', 'NATAL');**

**INSERT INTO Disciplinas (COD\_DISC, nome\_disc, carga\_hor) VALUES**

**('BD', 'BANCO DE DADOS', 100),**

**('POO', 'PROGRAMAÇÃO COM ACESSO A BANCO DE DADOS', 100), ('WEB', 'AUTORIA WEB', 50),**

**('ENG', 'ENGENHARIA DE SOFTWARE', 80);**

**INSERT INTO Professores (COD\_PROF, nome, endereco, cidade) VALUES**

**(212131, 'NICKERSON FERREIRA', 'RUA MANAÍRA', 'JOÃO PESSOA'), (122135, 'ADORILSON BEZERRA', 'AVENIDA SALGADO FILHO', 'NATAL'), (192011, 'DIEGO OLIVEIRA', 'AVENIDA ROBERTO FREIRE', 'NATAL');**

**INSERT INTO Turma (COD\_DISC, COD\_TURMA, COD\_PROF, ANO, horario) VALUES**

**('BD', 1, 212131, 2015, '11H-12H'),**

**('BD', 2, 212131, 2015, '13H-14H'),**

**('POO', 1, 192011, 2015, '08H-09H'),**

**('WEB', 1, 192011, 2015, '07H-08H'),**

**('ENG', 1, 122135, 2015, '10H-11H');**

**INSERT INTO Historico (MAT, COD\_DISC, COD\_TURMA, COD\_PROF, ANO, frequência, nota) VALUES**

**('2015010101', 'BD', 1, 212131, 2015, 80.00, 4.2),**

**('2015010101', 'POO', 1, 192011, 2015, 85.00, 7.5),**

**('2015010101', 'WEB', 1, 192011, 2015, 90.00, 3.9),**

**('2015010101', 'ENG', 1, 122135, 2015, 95.00, 6.4),**

**('2015010102', 'BD', 2, 212131, 2015, 80.00, 8.5),**

**('2015010102', 'POO', 1, 192011, 2015, 85.00, 5.2),**

**('2015010102', 'WEB', 1, 192011, 2015, 90.00, 7.3),**

**('2015010102', 'ENG', 1, 122135, 2015, 95.00, 2.8),**

**('2015010103', 'BD', 1, 212131, 2015, 80.00, 9.1),**

**('2015010103', 'POO', 1, 192011, 2015, 85.00, 4.8),**

**('2015010103', 'WEB', 1, 192011, 2015, 90.00, 6.7),**

**('2015010103', 'ENG', 1, 122135, 2015, 95.00, 3.5),**

**('2015010104', 'BD', 2, 212131, 2015, 80.00, 6.2),**

**('2015010104', 'POO', 1, 192011, 2015, 85.00, 5.6),**

**('2015010104', 'WEB', 1, 192011, 2015, 90.00, 4.3),**

**('2015010104', 'ENG', 1, 122135, 2015, 95.00, 9.9),**

**('2015010105', 'BD', 1, 212131, 2015, 80.00, 3.1),**

**('2015010105', 'POO', 1, 192011, 2015, 85.00, 8.4),**

**('2015010105', 'WEB', 1, 192011, 2015, 90.00, 9.7),**

**('2015010105', 'ENG', 1, 122135, 2015, 95.00, 2.0),**

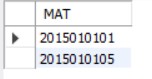
**('2015010106', 'BD', 2, 212131, 2015, 80.00, 5.4),**

**('2015010106', 'POO', 1, 192011, 2015, 85.00, 1.3),**

**('2015010106', 'WEB', 1, 192011, 2015, 90.00, 6.1),**

**('2015010106', 'ENG', 1, 122135, 2015, 95.00, 4.9);**

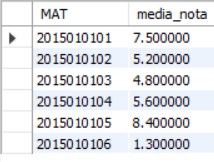
* 1. Encontre a MAT dos alunos com nota em BD em 2015 menor que 5 (obs: BD = código da disciplinas).

-- Letra A-- SELECT MAT

FROM Historico

WHERE COD\_DISC = 'BD' AND ANO = 2015

AND nota < 5;

* 1. Encontre a MAT e calcule a média das notas dos alunos na disciplina de POO em 2015.

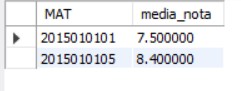
-- Letra B --

SELECT MAT, AVG(nota) AS media\_nota FROM Historico

WHERE COD\_DISC = 'POO' AND ANO = 2015

GROUP BY MAT;

* 1. Encontre a MAT e calcule a média das notas dos alunos na disciplina de POO em 2015 e que esta média seja superior a 6.

SELECT MAT, AVG(nota) AS media\_nota FROM Historico

WHERE COD\_DISC = 'POO' AND ANO = 2015

GROUP BY MAT HAVING AVG(nota) > 6;

* 1. Encontre quantos alunos **não** são de Natal.

SELECT COUNT(\*) AS quantidade\_alunos FROM Alunos

WHERE cidade <> 'NATAL';