PUC

EXERCÍCIOS PRÁTICOS DE REVISÃO

Banco de Dados

1.1. Criação de um Banco de Dados

Crie o banco de dados vendas que possua a seguinte estrutura:

CLIENTE(<u>cliente_id</u>*, cliente_nome,estado_codigo*)

ESTADO(estado_codigo*,estado_nome)

GRUPO(grupo_codigo,grupo_nome)

PRODUTO(produto_codigo*,produto_nome,produto_punit,produto_qtde,grupo_codigo*)

ITEM(item_id*,pedido_id*,produto_id*,item_qtde,item_punit)

PEDIDO(pedido_id*,pedido_tipopg,cliente_id*,pedido_data)

1.2 Crie o banco e as tabelas indicadas

```
Cria o banco: Create database vendas
Cria as tabelas:
CREATE TABLE 'estado' (
'estado codigo' INT(11) NOT NULL AUTO INCREMENT,
'estado nome' VARCHAR(30) NULL DEFAULT NULL,
PRIMARY KEY ('estado codigo')
ENGINE=InnoDB
CREATE TABLE 'cliente' (
'cliente id' INT(11) NOT NULL AUTO INCREMENT,
'cliente nome' VARCHAR(40) NULL DEFAULT NULL,
'cliente_perc' DOUBLE(6,2) NULL DEFAULT NULL,
'estado codigo' INT(11) NULL DEFAULT NULL,
PRIMARY KEY ('cliente_id'),
CONSTRAINT
                `estado cliente 01`
                                   FOREIGN
                                               KEY
                                                      (`estado_codigo`)
                                                                        REFERENCES
                                                                                         `estado`
('estado_codigo')
ENGINE=InnoDB
CREATE TABLE `grupo` (
`grupo_codigo` INT(11) NOT NULL,
`grupo_nome` VARCHAR(30) NOT NULL,
PRIMARY KEY ('grupo_codigo')
ENGINE=InnoDB
CREATE TABLE 'produto' (
`produto_codigo` INT(11) NOT NULL,
```



```
'produto nome' VARCHAR(40) NOT NULL,
'produto_punit' DECIMAL(10,2) NOT NULL,
`produto_qtde` INT(11) NOT NULL,
`grupo codigo` INT(11) NULL DEFAULT NULL,
PRIMARY KEY ('produto_codigo'),
INDEX 'produto grupo 01' ('grupo codigo'),
CONSTRAINT 'produto grupo 01' FOREIGN KEY ('grupo codigo') REFERENCES 'grupo' ('grupo codigo')
ENGINE=InnoDB
CREATE TABLE `pedido` (
'pedido id' INT(11) NOT NULL,
`cliente_id` INT(11) NOT NULL,
'pedido-data' DATE NOT NULL,
'pedido tipopg' VARCHAR(3) NOT NULL,
PRIMARY KEY ('pedido_id'),
INDEX `pedido_cliente_01` (`cliente_id`),
CONSTRAINT 'pedido cliente 01' FOREIGN KEY ('cliente id') REFERENCES 'cliente' ('cliente id')
ENGINE=InnoDB
CREATE TABLE 'item' (
'item id' INT(11) NOT NULL,
'pedido_id' INT(11) NOT NULL,
`produto_id` INT(11) NOT NULL,
'item_qtde' INT(11) NOT NULL,
'item_punit' DECIMAL(6,2) NOT NULL,
PRIMARY KEY ('item_id'),
INDEX `item_produto_01` (`produto_id`),
INDEX 'item pedido 01' ('pedido id'),
CONSTRAINT 'item produto 01' FOREIGN KEY ('produto id') REFERENCES 'produto '('produto codigo'),
CONSTRAINT 'item pedido 01' FOREIGN KEY ('pedido id') REFERENCES 'pedido' ('pedido id')
ENGINE=InnoDB;
NOTA: ao copiar esse script, observe se ocorrerá erro devido as aspas simples. Se ocorrer, retire as aspas
simples.
```

1.3 Dados cadastrados

vendas.cliente: 7 registros totais (aproximadamente)

diente_id	diente_nome	estado_codigo
1	JOSE DA SILVA	1
2	CARLOS PEREIRA	2
3	JOSE SILVEIRA	3
4	CARLOS ANTONIO	4
5	SILVIO JOSE	1
6	ANTONIO CARLOS	2
7	JOÃO CARLOS	3



vendas.estado: 5 registros totais (aproximadamente)

1 GO 2 DF 3 SP 4 MG 5 SC	estado_codigo	estado_nome
3 SP 4 MG	1	GO
4 MG	2	DF
	3	SP
5 SC	4	MG
	5	SC

vendas.grupo: 3 registros totais (aproximadamente)

grupo_nome
ESCRITÓRIO
LIMPEZA
INFORMÁTICA

vendas.pedido: 10 registros totais (aproximadamente)

pedido_id	diente_id	pedido-data	pedido_tipopg
1	1	2017-05-14	VVI
2	1	2017-05-30	VPR
3	2	2017-05-18	VPR
4	2	2017-05-25	VVI
5	3	2017-06-01	VPR
6	4	2017-06-14	VVI
7	5	2017-06-14	VPR
8	6	2017-06-13	VVI
9	7	2017-06-12	VPR
10	7	2017-05-12	VVI

vendas.produto: 6 registros totais (aproximadamente)

produto_codigo	produto_nome	produto_punit	produto_qtde	grupo_codigo
1	SABONETE LIQUIDO	8,25	10	2
2	CANETA ESFEROGRÁFICA	2,00	100	1
3	USB 8	40,00	50	3
4	USB 32	80,00	20	3
5	LAPIS PRETO	1,00	80	1
6	DETERGENTE	2,00	120	2

vendas.item: 19 registros totais (aproximadamente)

item_punit	item_qtde	produto_id	pedido_id	item_id
7,00	10	1	1	1
1,00	8	2	1	4
38,00	2	3	1	5
8,00	3	1	2	6
40,00	1	3	2	7
78,00	2	4	3	8
0,00	20	5	3	9
2,00	30	6	4	10
8,00	5	1	4	11
2,00	10	2	4	12
40,00	3	3	5	14
80,00	1	4	5	15
2,00	3	6	6	16
1,00	20	5	6	17
8,00	12	1	7	18
2,00	20	2	7	19
35,00	4	3	8	20
73,00	5	4	9	21
1,00	25	5	10	23

- 1.4 Crie os scripts para inserir os dados indicados nas tabelas anteriores
- 1.5 Crie alguns scripts para alterar alguns dados nas tabelas
- 1.6 Crie alguns scripts para excluir alguns registros da tabela item
- 1.7 Escreva scripts para as seguintes situações:
 - a. Mostre todos os campos de todos clientes





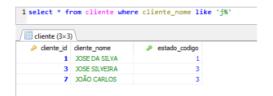
b. Mostre todos os campos de todos os clientes por ordem de nome do cliente



c. Mostre apenas o campo nome do cliente de todos os clientes cadastrados em ordem decrescente de nome

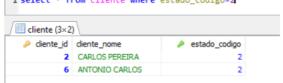


d. Mostre o nome e o código do estado apenas dos clientes que possuem o nome iniciado com J

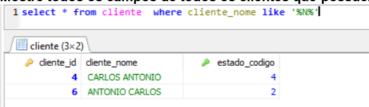


e. Mostre apenas o nome de todos os clientes do estado de código 2 em ordem de nome do cliente

1 select * from cliente where estado_codigo=2



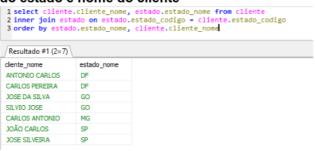
f. Mostre todos os campos de todos os clientes que possuem a letra N em algum lugar do nome



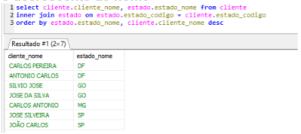
g. Mostre todos os campos de todos os clientes cujo nome termina com a letra a, em ordem de nome



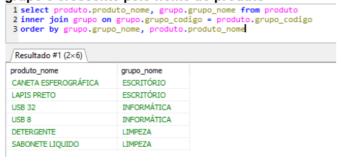
h. Mostre o nome do cliente e o nome do estado que o cliente reside, por ordem crescente do nome do estado e nome do cliente



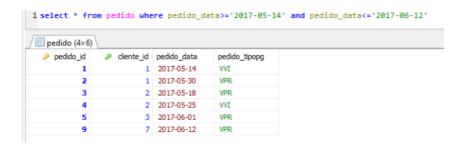
 Mostre o nome do cliente e o nome do estado que o cliente reside, por ordem crescente de estado e decrescente de nome



 Mostre o nome de cada produto e o nome do seu respectivo grupo, por ordem crescente de grupo e crescente pelo nome do produto

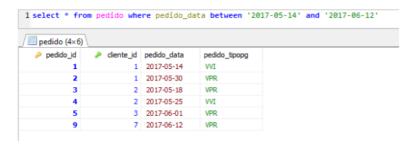


k. Mostre todos os campos dos pedidos com venda entre 14.05.2017 a 12.06.2017





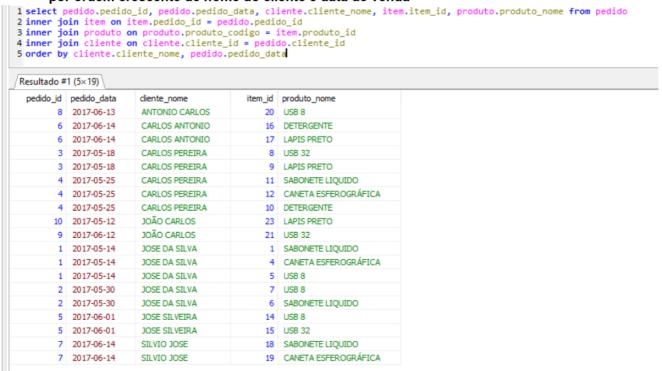
Com between



 Mostre o número do pedido, data de venda e nome do cliente por ordem crescente do nome do cliente



m. Mostre o número do pedido, data de venda, nome do cliente, código do item, descrição do item, por ordem crescente de nome do cliente e data de venda



n. Mostre o total da venda de todos os pedidos cadastrados no sistema

```
1 select sum(item.item_qtde * item.item_punit) as total from item

| Resultado #1 (1×1) | total | 1.386,00 |
```

o. Mostre o total da venda apenas dos pedidos vendidos entre 12.05.2017 a 30.05.2017

```
1 select sum(item.item_qtde * item.item_punit) from item
2 inner join pedido on pedido.pedido_id = item.pedido_id
3 where pedido.pedido_data>='2017-05-12' and pedido.pedido_data<='2017-05-30'

Resultado #1 (1×1)

sum(item.item_qtde * item.item_punit)

519,00</pre>
```

p. Mostre o total de venda de todos os pedidos do cliente João Carlos

```
1 select sum(item.item_qtde * item.item_punit) as total from item
2 inner join pedido on pedido.pedido_id = item.pedido_id
3 inner join cliente on cliente.cliente_id = pedido.cliente_id
4 where cliente.cliente_nome='JOÃO CARLOS'

Resultado #1 (1×1)

total
390,00
```

q. Apresente a média de preço unitário de todo o estoque

```
1 select avg(produto.produto_punit) 'média de preço' from produto

Resultado #1 (1×1)

média de preço
22,208333
```

r. Apresente o maior preço unitário do estoque

```
1 select max(produto.produto_punit) 'maior preço unitário' from produto

Resultado #1 (1×1) \
maior preço unitário
80,00
```

s. Apresente o menor preço unitário do estoque

```
1 select min(produto.produto_punit) 'menor preço unitário' from produto

Resultado #1 (1×1)

menor preço unitário

1,00
```

t. Mostre a quantidade de registros de itens vendidos

```
1 select count(*) from item

Resultado #1 (1×1) \

count(*)

19
```

u. Mostre a quantidade de itens vendidos

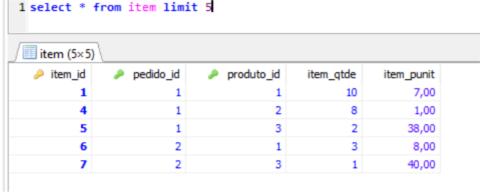
```
1 select sum(item.item_qtde) from item

Resultado #1 (1×1) \

sum(item.item_qtde)

184
```

v. Mostre apenas os 5 primeiros registros da tabela item



 w. Mostre o número do pedido e o nome do cliente de todos os pedidos vendidos a prazo (VPR) por ordem crescente pelo nome do cliente