



## 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(cliente\_id\*, 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)

 cliente_id	cliente_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



ESCOLA DE CIÊNCIAS EXATAS E COMPUTAÇÃO  
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
DISCIPLINA MODELAGEM CONCEITUAL DE DADOS



vendas.estado: 5 registros totais (aproximadamente)

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

vendas.grupo: 3 registros totais (aproximadamente)

 grupo_codigo	grupo_nome
1	ESCRITÓRIO
2	LIMPEZA
3	INFORMÁTICA

vendas.pedido: 10 registros totais (aproximadamente)

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

**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**

```
1 select * from cliente
```

cliente_id	cliente_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

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

```
1 select * from cliente order by cliente_nome
```

cliente (3x7)		
cliente_id	cliente_nome	estado_codigo
6	ANTONIO CARLOS	2
4	CARLOS ANTONIO	4
2	CARLOS PEREIRA	2
7	JOÃO CARLOS	3
1	JOSE DA SILVA	1
3	JOSE SILVEIRA	3
5	SILVIO JOSE	1

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

```
1 select cliente_nome from cliente order by cliente_nome desc
```

cliente (1x7)	
cliente_nome	
SILVIO JOSE	
JOSE SILVEIRA	
JOSE DA SILVA	
JOÃO CARLOS	
CARLOS PEREIRA	
CARLOS ANTONIO	
ANTONIO CARLOS	

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

```
1 select * from cliente where cliente_nome like 'j%'
```

cliente (3x3)		
cliente_id	cliente_nome	estado_codigo
1	JOSE DA SILVA	1
3	JOSE SILVEIRA	3
7	JOÃO CARLOS	3

**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
```

cliente (3x2)		
cliente_id	cliente_nome	estado_codigo
2	CARLOS PEREIRA	2
6	ANTONIO CARLOS	2

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

```
1 select * from cliente where cliente_nome like '%N%'
```

cliente (3x2)		
cliente_id	cliente_nome	estado_codigo
4	CARLOS ANTONIO	4
6	ANTONIO CARLOS	2

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

```
1 select * from cliente where cliente_nome like '%A' order by cliente_nome
```

cliente (3x3)		
cliente_id	cliente_nome	estado_codigo
2	CARLOS PEREIRA	2
1	JOSE DA SILVA	1
3	JOSE SILVEIRA	3

- 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

```
1 select cliente.cliente_nome, estado.estado_nome from cliente
2 inner join estado on estado.estado_codigo = cliente.estado_codigo
3 order by estado.estado_nome, cliente.cliente_nome
```

Resultado #1 (2x7)	
cliente_nome	estado_nome
ANTONIO CARLOS	DF
CARLOS PEREIRA	DF
JOSE DA SILVA	GO
SILVIO JOSE	GO
CARLOS ANTONIO	MG
JOÃO CARLOS	SP
JOSE SILVEIRA	SP

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

```
1 select cliente.cliente_nome, estado.estado_nome from cliente
2 inner join estado on estado.estado_codigo = cliente.estado_codigo
3 order by estado.estado_nome, cliente.cliente_nome desc
```

Resultado #1 (2x7)	
cliente_nome	estado_nome
CARLOS PEREIRA	DF
ANTONIO CARLOS	DF
SILVIO JOSE	GO
JOSE DA SILVA	GO
CARLOS ANTONIO	MG
JOSE SILVEIRA	SP
JOÃO CARLOS	SP

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

```
1 select produto.produto_nome, grupo.grupo_nome from produto
2 inner join grupo on grupo.grupo_codigo = produto.grupo_codigo
3 order by grupo.grupo_nome, produto.produto_nome
```

Resultado #1 (2x6)	
produto_nome	grupo_nome
CANETA ESFEROGRÁFICA	ESCRITÓRIO
LAPIS PRETO	ESCRITÓRIO
USB 32	INFORMÁTICA
USB 8	INFORMÁTICA
DETERGENTE	LIMPEZA
SABONETE LIQUIDO	LIMPEZA

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

```
1 select * from pedido where pedido_data>='2017-05-14' and pedido_data<='2017-06-12'
```

pedido (4x6)			
pedido_id	cliente_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
9	7	2017-06-12	VPR

### Com between

```
1 select * from pedido where pedido_data between '2017-05-14' and '2017-06-12'
```

pedido (4x6)

pedido_id	cliente_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
9	7	2017-06-12	VPR

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

```
1 select pedido.pedido_id, pedido.pedido_data, cliente.cliente_nome from pedido
2 inner join cliente on cliente.cliente_id = pedido.cliente_id
3 order by cliente.cliente_nome
```

Resultado #1 (3x10)

pedido_id	pedido_data	cliente_nome
8	2017-06-13	ANTONIO CARLOS
6	2017-06-14	CARLOS ANTONIO
3	2017-05-18	CARLOS PEREIRA
4	2017-05-25	CARLOS PEREIRA
9	2017-06-12	JOÃO CARLOS
10	2017-05-12	JOÃO CARLOS
2	2017-05-30	JOSE DA SILVA
1	2017-05-14	JOSE DA SILVA
5	2017-06-01	JOSE SILVEIRA
7	2017-06-14	SILVIO JOSE

- 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

```
1 select pedido.pedido_id, pedido.pedido_data, cliente.cliente_nome, item.item_id, produto.produto_nome from pedido
2 inner join item on item.pedido_id = pedido.pedido_id
3 inner join produto on produto.produto_codigo = item.produto_id
4 inner join cliente on cliente.cliente_id = pedido.cliente_id
5 order by cliente.cliente_nome, pedido.pedido_data
```

Resultado #1 (5x19)

pedido_id	pedido_data	cliente_nome	item_id	produto_nome
8	2017-06-13	ANTONIO CARLOS	20	USB 8
6	2017-06-14	CARLOS ANTONIO	16	DETERGENTE
6	2017-06-14	CARLOS ANTONIO	17	LAPIS PRETO
3	2017-05-18	CARLOS PEREIRA	8	USB 32
3	2017-05-18	CARLOS PEREIRA	9	LAPIS PRETO
4	2017-05-25	CARLOS PEREIRA	11	SABONETE LIQUIDO
4	2017-05-25	CARLOS PEREIRA	12	CANETA ESFEROGRÁFICA
4	2017-05-25	CARLOS PEREIRA	10	DETERGENTE
10	2017-05-12	JOÃO CARLOS	23	LAPIS PRETO
9	2017-06-12	JOÃO CARLOS	21	USB 32
1	2017-05-14	JOSE DA SILVA	1	SABONETE LIQUIDO
1	2017-05-14	JOSE DA SILVA	4	CANETA ESFEROGRÁFICA
1	2017-05-14	JOSE DA SILVA	5	USB 8
2	2017-05-30	JOSE DA SILVA	7	USB 8
2	2017-05-30	JOSE DA SILVA	6	SABONETE LIQUIDO
5	2017-06-01	JOSE SILVEIRA	14	USB 8
5	2017-06-01	JOSE SILVEIRA	15	USB 32
7	2017-06-14	SILVIO JOSE	18	SABONETE LIQUIDO
7	2017-06-14	SILVIO JOSE	19	CANETA ESFEROGRÁFICA

- 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

- 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

```
1 select * from item limit 5
```

item (5×5)

item_id	pedido_id	produto_id	item_qtde	item_punit
1	1	1	10	7,00
4	1	2	8	1,00
5	1	3	2	38,00
6	2	1	3	8,00
7	2	3	1	40,00

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

```
1 select pedido.pedido_id, cliente.cliente_nome from pedido
2 inner join cliente on cliente.cliente_id = pedido.cliente_id
3 where pedido.pedido_tipopg='VPR'
4 order by cliente.cliente_nome
```

Resultado #1 (2×5)

pedido_id	cliente_nome
3	CARLOS PEREIRA
9	JOÃO CARLOS
2	JOSE DA SILVA
5	JOSE SILVEIRA
7	SILVIO JOSE