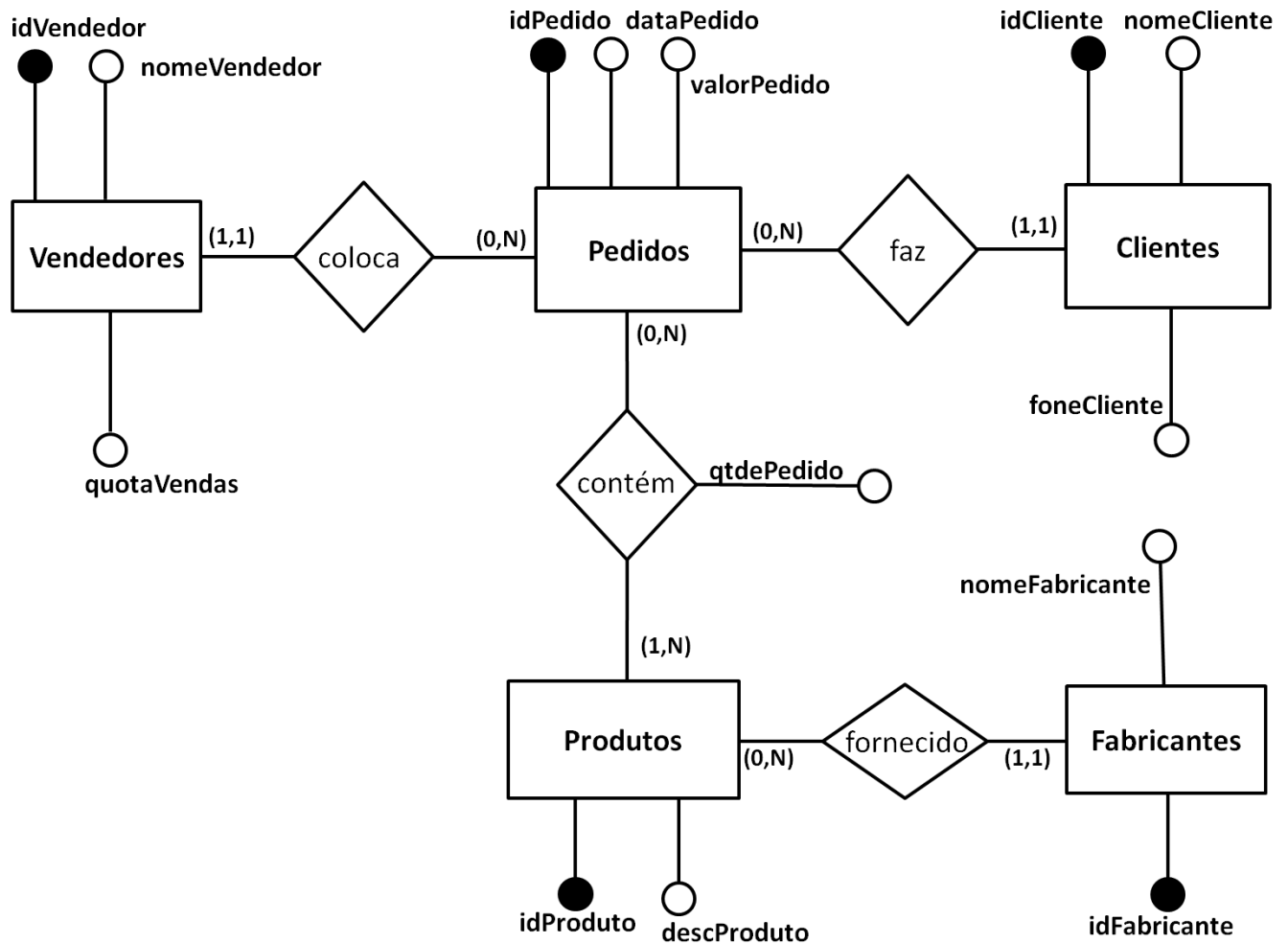


TTI 102 - Banco de Dados Relacionais – Tarefa T15 - Solução

Queries Multi-Tabelas – MySQL – Prof. Dr. Aparecido Freitas

1. Iniciar o servidor de Banco de Dados **MySQL**. Criar um banco de dados chamado **dbpedidos**;
2. Para a criação e população do Banco de dados considere o Script **TarefaT15.sql**;
3. Considere o seguinte Diagrama Entidade-Relacionamento para o Banco de Dados;



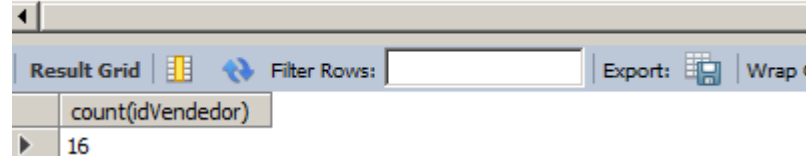
4. Escrever a query **SQL** para se obter a quota média de todos os vendedores;

```
177 •      select avg(quotaVendas) from vendedores;
178
179
```

Result Grid		Filter Rows:	Export:	Wrap Cell C
avg(quotaVendas)				
10455.006250				

5. Escrever a query **SQL** para se obter a quantidade de vendedores cadastrados no banco de dados;

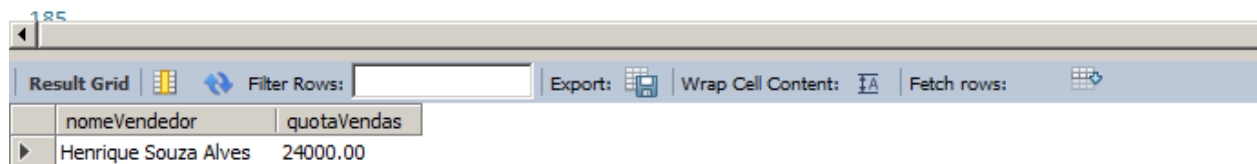
```
181 • select count(idVendedor) from vendedores;
```



The screenshot shows a SQL query result grid. The query is `select count(idVendedor) from vendedores;`. The result grid has a single column header `count(idVendedor)` and a single row with the value `16`.

6. Escrever a query **SQL** para se obter o nome do Vendedor que possui a maior quota de vendas. Exibir também juntamente com o nome do vendedor a sua quota;

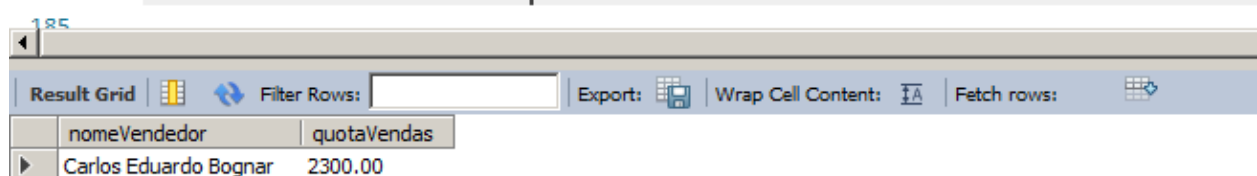
```
184 • select nomeVendedor, quotaVendas from vendedores order by quotaVendas desc limit 1;
```



The screenshot shows a SQL query result grid. The query is `select nomeVendedor, quotaVendas from vendedores order by quotaVendas desc limit 1;`. The result grid has two columns: `nomeVendedor` and `quotaVendas`. The single row shows `Henrique Souza Alves` and `24000.00`.

7. Escrever a query **SQL** para se obter o nome do Vendedor que possui a menor quota de vendas. Exibir também juntamente com o nome do vendedor a sua quota;

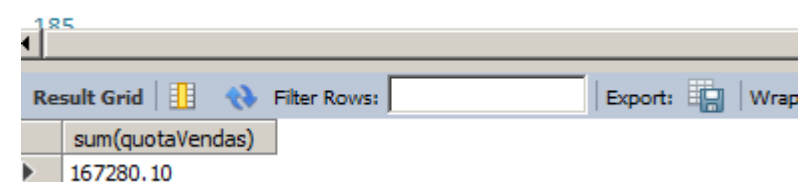
```
184 • select nomeVendedor, quotaVendas from vendedores order by quotaVendas limit 1;
```



The screenshot shows a SQL query result grid. The query is `select nomeVendedor, quotaVendas from vendedores order by quotaVendas limit 1;`. The result grid has two columns: `nomeVendedor` and `quotaVendas`. The single row shows `Carlos Eduardo Bognar` and `2300.00`.

8. Escrever a query **SQL** para se obter a somatória das quotas de vendas de todos os vendedores cadastrados no banco de dados;

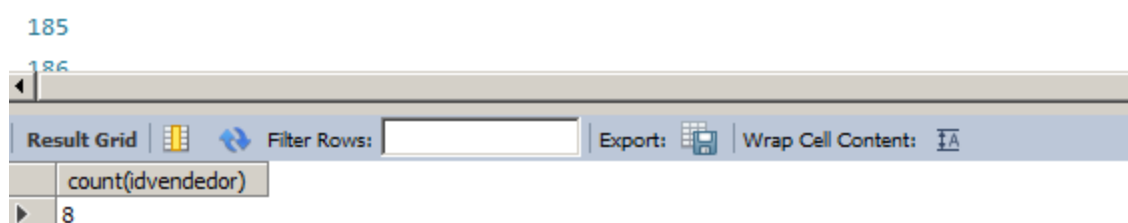
```
184 • select sum(quotaVendas) from vendedores;
```



The screenshot shows a SQL query result grid. The query is `select sum(quotaVendas) from vendedores;`. The result grid has a single column header `sum(quotaVendas)` and a single row with the value `167280.10`.

9. Escrever a query **SQL** para se saber quantos vendedores têm quota de vendas superior a 8000.00;

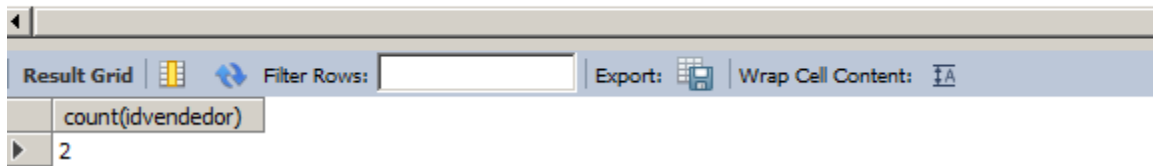
```
184 • select count(idvendedor) from vendedores where quotaVendas > 8000;
```



The screenshot shows a SQL query result grid. The query is `select count(idvendedor) from vendedores where quotaVendas > 8000;`. The result grid has a single column header `count(idvendedor)` and a single row with the value `8`.

10. Escrever a query **SQL** para se saber quantos pedidos o vendedor 1056 colocou ?

```
183 • select count(idvendedor) from vendedores where quotaVendas > 8000;  
184
```

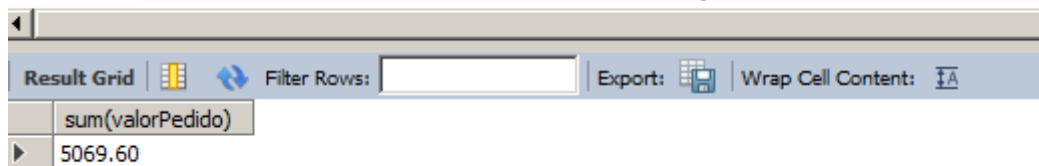


The screenshot shows a SQL query result grid. The query is `select count(idvendedor) from vendedores where quotaVendas > 8000;`. The result grid has a single column labeled `count(idvendedor)` and a single row with the value `2`.

count(idvendedor)
2

11. Escrever a query **SQL** para se saber o valor total dos pedidos colocados pelo vendedor 1002 ?

```
187 • select sum(valorPedido) from pedidos where idvendedor = 1002;
```

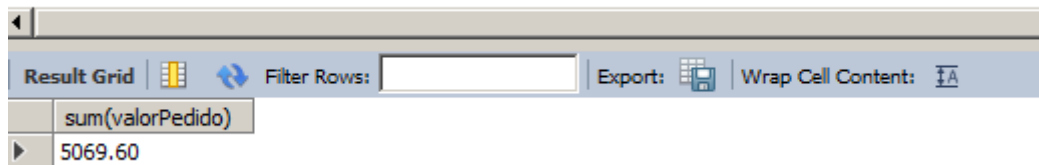


The screenshot shows a SQL query result grid. The query is `select sum(valorPedido) from pedidos where idvendedor = 1002;`. The result grid has a single column labeled `sum(valorPedido)` and a single row with the value `5069.60`.

sum(valorPedido)
5069.60

12. Escrever a query **SQL** para se saber o valor total dos pedidos colocados pelo vendedor 1002 ?

```
187 • select sum(valorPedido) from pedidos where idvendedor = 1002;
```

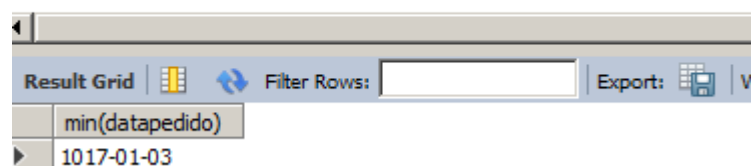


The screenshot shows a SQL query result grid. The query is `select sum(valorPedido) from pedidos where idvendedor = 1002;`. The result grid has a single column labeled `sum(valorPedido)` and a single row with the value `5069.60`.

sum(valorPedido)
5069.60

13. Escrever a query **SQL** para se saber qual o pedido mais antigo registrado no Banco de Dados ?

```
190 • select min(datapedido) from pedidos;  
191
```



The screenshot shows a SQL query result grid. The query is `select min(datapedido) from pedidos;`. The result grid has a single column labeled `min(datapedido)` and a single row with the value `1017-01-03`.

min(datapedido)
1017-01-03

14. Escrever a query **SQL** para se saber quantos pedidos estão com valor total acima de 12.000.00 ?

```
195 • select count(valorPedido) from pedidos where valorPedido > 12000;
196
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

count(valorPedido)
2

15. Escrever a query **SQL** para se saber qual a média dos valores dos pedidos colocados por cada vendedor ?

```
197 • select idvendedor, avg(valorPedido) from pedidos
198       group by idvendedor order by idvendedor;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

idvendedor	avg(valorPedido)
1000	1200.600000
1001	134.600000
1002	2534.800000
1021	23000.600000
1032	7483.933333
1034	9875.300000
1045	1230.600000
1056	16438.600000
1067	8765.600000
1068	5432.600000
1078	3210.600000
1081	2109.600000
1098	8765.600000
1099	6498.800000

16. Escrever a query **SQL** para se saber o total de pedidos colocados por cada vendedor ?

```
197 • select idvendedor, count(idvendedor) from pedidos
198       group by idvendedor order by idvendedor;
199
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

idvendedor	count(idvendedor)
1000	1
1001	1
1002	2
1021	1
1032	3
1034	1
1045	1
1056	2
1067	1
1068	1
1078	1
1081	1
1098	1
1099	3

17. Escrever a query **SQL** para se saber a quantidade máxima do valor do Pedido e a quantidade mínima do valor do Pedido para cada vendedor ?

```
201 • select idvendedor, min(valorPedido), max(valorPedido) from pedidos
202      group by idvendedor;
```

Result Grid			
		Filter Rows:	
		Export:	Wrap Cell Content: IA
	idvendedor	min(valorPedido)	max(valorPedido)
▶	1000	1200.60	1200.60
	1001	134.60	134.60
	1002	500.60	4569.00
	1021	23000.60	23000.60
	1032	6000.60	8800.60
	1034	9875.30	9875.30
	1045	1230.60	1230.60
	1056	9876.60	23000.60
	1067	8765.60	8765.60
	1068	5432.60	5432.60
	1078	3210.60	3210.60
	1081	2109.60	2109.60
	1098	8765.60	8765.60
	1099	4321.80	8631.00

18. Escrever a query **SQL** para se saber a quantidade máxima do valor do Pedido e a quantidade mínima do valor do Pedido para cada vendedor que vendeu no mínimo 10000 ?

```
202 • select idvendedor, min(valorPedido), max(valorPedido) from pedidos
203      where valorPedido >= 10000.00
204      group by idvendedor
205      order by idvendedor;
206
```

Result Grid			
		Filter Rows:	
		Export:	Wrap Cell Content: IA
	idvendedor	min(valorPedido)	max(valorPedido)
▶	1021	23000.60	23000.60
	1056	23000.60	23000.60

19. Escrever a query **SQL** para se saber a quantidade de pedidos que um vendedor colocou valor de pedido superior a 1000.00, ordenado pelo código do vendedor.

```
207 • select idvendedor, count(*) as quantidadePedidosacimade1000 from pedidos
208     where valorPedido >= 1000.00
209     group by idvendedor
210     order by idvendedor;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
idvendedor	quantidadePedidosacimade1000			
1000	1			
1002	1			
1021	1			
1032	3			
1034	1			
1045	1			
1056	2			
1067	1			
1068	1			
1078	1			
1081	1			
1098	1			
1099	3			

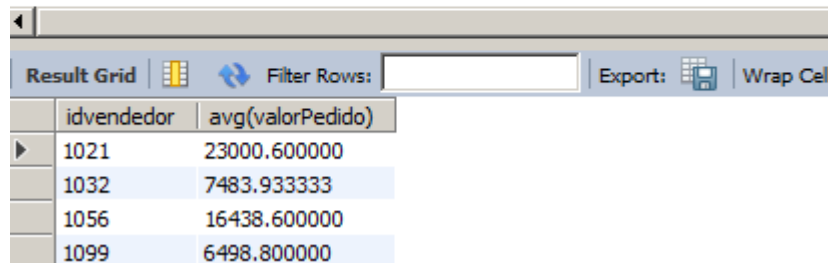
20. Escrever a query **SQL** para se saber a quantidade de pedidos que um vendedor colocou valor de pedido superior a 1000.00, ordenado pelo código do vendedor. Incluir na consulta o valor mínimo do pedido do vendedor e o valor máximo.

```
207 • select idvendedor, count(*) as quantidadePedidosacimade1000 from pedidos
208     where valorPedido >= 1000.00
209     group by idvendedor
210     order by idvendedor;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
idvendedor	count(*)	min(valorPedido)	max(valorPedido)	
1000	1	1200.60	1200.60	
1002	1	4569.00	4569.00	
1021	1	23000.60	23000.60	
1032	3	6000.60	8800.60	
1034	1	9875.30	9875.30	
1045	1	1230.60	1230.60	
1056	2	9876.60	23000.60	
1067	1	8765.60	8765.60	
1068	1	5432.60	5432.60	
1078	1	3210.60	3210.60	
1081	1	2109.60	2109.60	
1098	1	8765.60	8765.60	
1099	3	4321.80	8631.00	

21. Escrever a query **SQL** para se saber o valor médio do valor de cada pedido colocado pelos vendedores cuja soma dos pedidos esteja acima de 20000.

```
212 • select idvendedor, avg(valorPedido) from pedidos
213      group by idvendedor
214      having sum(valorPedido) > 10000
215      order by idvendedor;
216
```

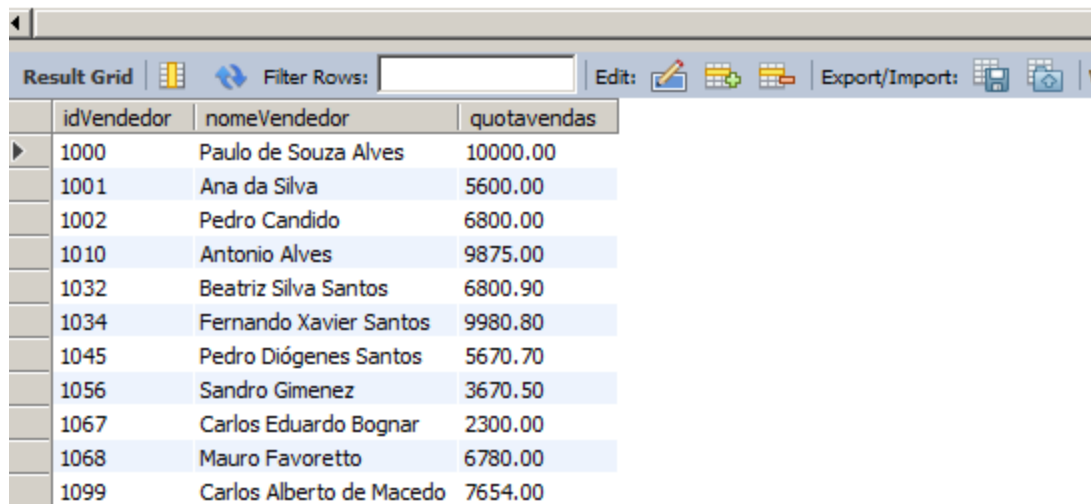


The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of the SQL query for question 21. The columns are 'idvendedor' and 'avg(valorPedido)'. The data is as follows:

idvendedor	avg(valorPedido)
1021	23000.600000
1032	7483.933333
1056	16438.600000
1099	6498.800000

22. Escrever a query **SQL** para se saber quais os vendedores cuja quota é menor que a quota de vendas de toda a empresa.

```
217 • select idVendedor, nomeVendedor, quotavendas from vendedores where
218      quotavendas <
219      (select avg(quotavendas) from vendedores);
220
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of the SQL query for question 22. The columns are 'idVendedor', 'nomeVendedor', and 'quotavendas'. The data is as follows:

idVendedor	nomeVendedor	quotavendas
1000	Paulo de Souza Alves	10000.00
1001	Ana da Silva	5600.00
1002	Pedro Candido	6800.00
1010	Antonio Alves	9875.00
1032	Beatriz Silva Santos	6800.90
1034	Fernando Xavier Santos	9980.80
1045	Pedro Diógenes Santos	5670.70
1056	Sandro Gimenez	3670.50
1067	Carlos Eduardo Bognar	2300.00
1068	Mauro Favoretto	6780.00
1099	Carlos Alberto de Macedo	7654.00

Script:

```
create database dbpedidos;
```

```
use dbpedidos;
```

```
create table fabricantes (idFabricante integer not null primary key,  
descricaoFabricante char(80) not null );
```

```
insert into fabricantes (idFabricante, descricaoFabricante) values
```

```
(10, 'Acer'),
```

```
(20, 'Adata'),
```

```
(30, 'Asus'),
```

```
(40, 'Corsair'),
```

```
(50, 'Dell'),
```

```
(60, 'Intel'),
```

```
(70, 'HP'),
```

```
(80, 'Logitech'),
```

```
(90, 'Samsung'),
```

```
(100, 'Seagate'),
```

```
(110, 'Western Digital'),
```

```
(120, 'Xiommi');
```

```
select * from fabricantes;
```

```
create table vendedores (idVendedor integer not null primary key,  
nomeVendedor char(80) not null, cpfVendedor char(14) not null unique,  
quotaVendas decimal(8,2) not null);
```

```
insert into vendedores (idVendedor, nomeVendedor, cpfVendedor,
quotaVendas) values
```

```
(1000, "Paulo de Souza Alves", "234.876.855-10", 10000.00),
(1001, "Ana da Silva", "666.876.855-10", 5600.00),
(1002, "Pedro Candido", "444.876.855-10", 6800.00),
(1010, "Antonio Alves", "269.876.855-45", 9875.00),
(1021, "Henrique Souza Alves", "903.876.855-42", 24000.00),
(1032, "Beatriz Silva Santos", "084.457.123-42", 6800.90),
(1034, "Fernando Xavier Santos", "991.670.120-77", 9980.80),
(1045, "Pedro Diógenes Santos", "042.668.234-22", 5670.70),
(1056, "Sandro Gimenez", "816.803.129-33", 3670.50),
(1067, "Carlos Eduardo Bognar", "775.662.127-33", 2300.00),
(1068, "Mauro Favoretto", "225.662.127-33", 6780.00),
(1078, "Maria Aparecida Flores", "388.800.120-49", 23700.80),
(1081, "Saulo de Souza", "691.389.114-91", 10790.50),
(1098, "Ivo de Alcantara", "892.339.966-66", 12876.40),
(1099, "Carlos Alberto de Macedo", "447.100.590-19", 7654.00),
(1101, "Renato Carioca", "778.124.801-17", 20780.50);
```

```
select * from vendedores;
```

```
create table clientes (idcliente integer not null primary key, nomeCliente
char(80) not null, cpfCliente char(14) not null unique, foneCliente char(20)
null);
```

```
insert into clientes (idCliente, nomeCliente, cpfCliente, foneCliente) values
```

```
(9000, "Pedro de Souza Alves", "666.876.855-10", "011 9 6546-0099"),
(9010, "Beatriz Souza", "124.871.855-10", "011 9 6790-0356"),
(9020, "Carlos Bognar Xavier", "098.876.855-10", "011 9 1234-8711"),
```

```
(9025, "Marcelo Alves", "230.876.855-45", "011 9 2356-0976"),
(9026, "Silvio Matheus Alves", "120.876.855-42", "011 9 3456-9855"),
(9028, "Carolina Silva Santos", "665.457.123-42", "011 9 3377-0978"),
(9030, "Ricardo Xavier Santos", "092.670.120-77", "011 9 2256-8872"),
(9035, "Paulo Freitas Silas", "873.668.234-22", "011 9 0987-1233"),
(9040, "Carla Andrade", "129.803.129-33", "011 9 9988-4780"),
(9041, "Benedito Silva Bognar", "270.662.127-33", "011 9 8870-2344"),
(9042, "Silton Favoretto", "309.662.127-33", "011 9 9866-9988"),
(9043, "Sueli Candida Flores", "129.800.120-49", "011 9 9990-5567"),
(9050, "Silas de Andrade", "855.389.114-91", "011 9 9888-3455"),
(9060, "Renato de Savio", "669.339.966-66", "011 9 9984-3481"),
(9065, "Adelsi Alberto de Macedo", "113.100.590-19", "011 9 8722-0981"),
(9070, "Karen Andrade", "008.124.801-17", "011 9 9000-3466");
```

```
select * from clientes;
```

```
create table pedidos (idPedido integer not null primary key, dataPedido date
not null, valorPedido decimal (8,2) not null, idCliente integer not null,
idVendedor integer not null,
foreign key(idCliente) references clientes(idCliente) ,
foreign key(idVendedor) references vendedores(idVendedor));
```

```
insert into pedidos (idPedido, dataPedido, valorPedido, idCliente,
idVendedor) values
```

```
(10, '2020-10-10', 23000.60, 9000, 1056),
(20, '2019-11-01', 1200.60, 9010, 1000),
(30, '2018-12-02', 134.60, 9010, 1001),
(40, '1017-01-03', 500.60, 9010, 1002),
(50, '2020-02-04', 4569.00, 9020, 1002),
(60, '2020-03-05', 23000.60, 9025, 1021),
```

```
(70, '2020-03-06', 6000.60, 9028, 1032),  
(80, '2020-04-07', 8800.60, 9035, 1032),  
(90, '2020-04-08', 7650.60, 9035, 1032),  
(100, '2020-05-09', 9875.30, 9040, 1034),  
(110, '2020-05-11', 1230.60, 9040, 1045),  
(120, '2020-06-11', 9876.60, 9040, 1056),  
(130, '2020-06-12', 8765.60, 9041, 1067),  
(140, '2020-07-13', 5432.60, 9042, 1068),  
(150, '2020-08-14', 3210.60, 9042, 1078),  
(160, '2020-09-15', 2109.60, 9043, 1081),  
(170, '2020-10-16', 8765.60, 9050, 1098),  
(180, '2020-10-17', 6543.60, 9050, 1099),  
(190, '2020-11-20', 4321.80, 9060, 1099),  
(200, '2020-12-25', 8631.00, 9060, 1099);
```

```
select * from pedidos;
```

```
create table produtos (idProduto integer not null primary key, descProduto  
char(80) not null, idFabricante integer not null,  
foreign key(idFabricante) references fabricantes(idFabricante) );
```

```
insert into produtos (idProduto, descProduto, idFabricante) values  
(200, 'Monitor', 90),  
(201, 'Caixa de Som', 80),  
(202, 'Placa de Vídeo', 40),  
(203, 'Placa de Vídeo', 20),  
(204, 'Notebook', 10),  
(205, 'Notebook', 30),  
(206, 'Notebook', 90),  
(207, 'Teclado', 40),  
(208, 'Teclado', 80),
```

```
(209, 'Teclado', 70),  
(210, 'Impressora', 70),  
(211, 'Fone de Ouvido', 70),  
(212, 'Fone de Ouvido', 120),  
(213, 'Fone de Ouvido', 40),  
(214, 'Headset', 40),  
(215, 'Headset', 70),  
(216, 'Headset', 80),  
(217, 'Cooler', 40),  
(218, 'Disco Rígido', 40),  
(219, 'Disco Rígido', 100),  
(220, 'Computador Gamer', 10),  
(221, 'Computador Gamer', 50),  
(222, 'Computador Gamer', 70);
```

```
select * from produtos;
```

```
create table itensPedidos (idPedido integer not null, idProduto integer not  
null, qtdePedida integer not null,  
primary key(idPedido, idProduto, qtdePedida),  
foreign key(idPedido) references pedidos(idPedido),  
foreign key(idProduto) references produtos(idProduto) );
```

```
insert into itensPedidos (idPedido, idProduto, qtdePedida) values
```

```
(10, 200, 10),  
(10, 210, 5),  
(10, 207, 2),  
(20, 207, 10),  
(20, 208, 40),  
(30, 214, 60),
```

(30, 215, 40),
(40, 220, 5),
(50, 200, 10),
(50, 201, 4),
(60, 212, 50),
(60, 214, 7),
(70, 219, 10),
(80, 218, 5),
(80, 221, 60),
(90, 215, 10),
(90, 216, 16),
(100, 220, 10),
(100, 221, 40),
(110, 215, 7),
(110, 209, 12),
(120, 204, 10),
(120, 205, 6),
(120, 206, 8),
(130, 211, 30),
(140, 212, 60),
(140, 213, 6),
(140, 200, 10),
(140, 201, 20),
(150, 218, 40),
(160, 218, 50),
(170, 205, 10),
(170, 206, 40),
(180, 208, 50),
(190, 209, 30),
(200, 202, 10),
(200, 203, 40);

```
select * from itensPedidos;
```