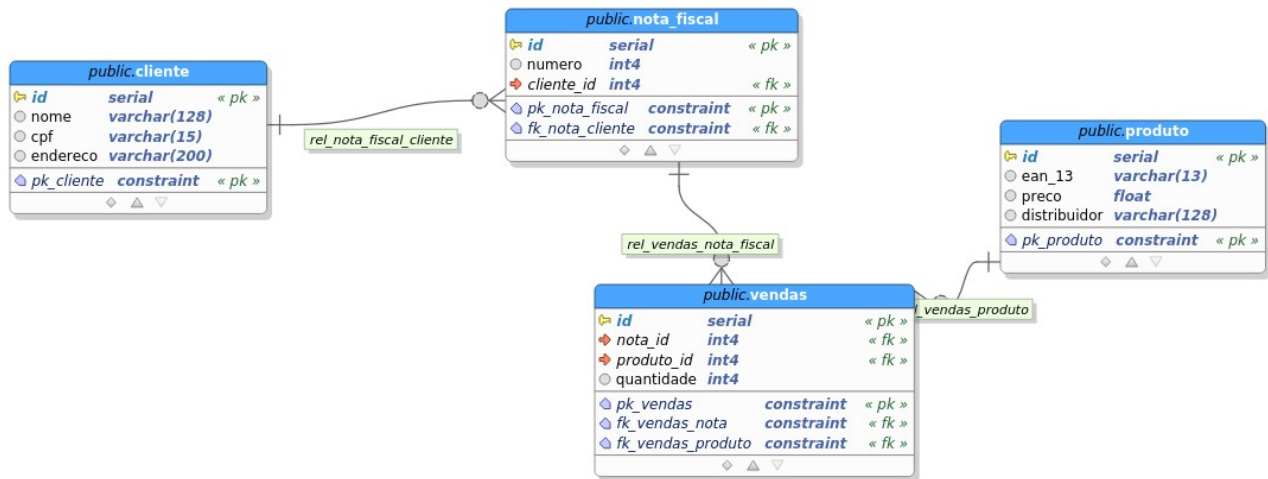
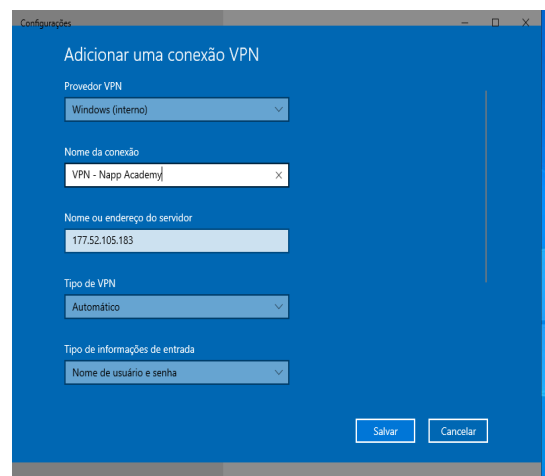
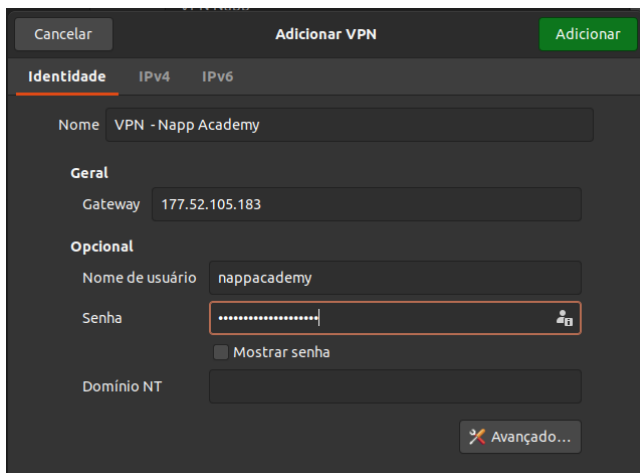


## Exercícios – Sprint 2

O contexto desta *sprint* trata-se de um mini-mundo de Vendas. No banco de dados indicado, existem quatro tabelas, conforme imagem a seguir:



Para acessar o banco de dados, é necessário ter acesso via VPN. Configure sua máquina com a VPN da Napp, conforme imagens a seguir:



**Gateway:** 177.52.105.183  
**Usuário:** nappacademy  
**Senha:** 286eXeGBN2XM9Cw5gz1K  
**Tipo Conexão:** PPTP

**Após o acesso via VPN**, para acesso ao banco de dados PostgreSQL, é necessário ter as seguintes credenciais:


**Host:** 38.242.156.52  
**Porta:** 5432  
**Usuário:** user\_alunos  
**Senha:** hGaXqzvQG2CHvGK

# Conexão via CloudBeaver

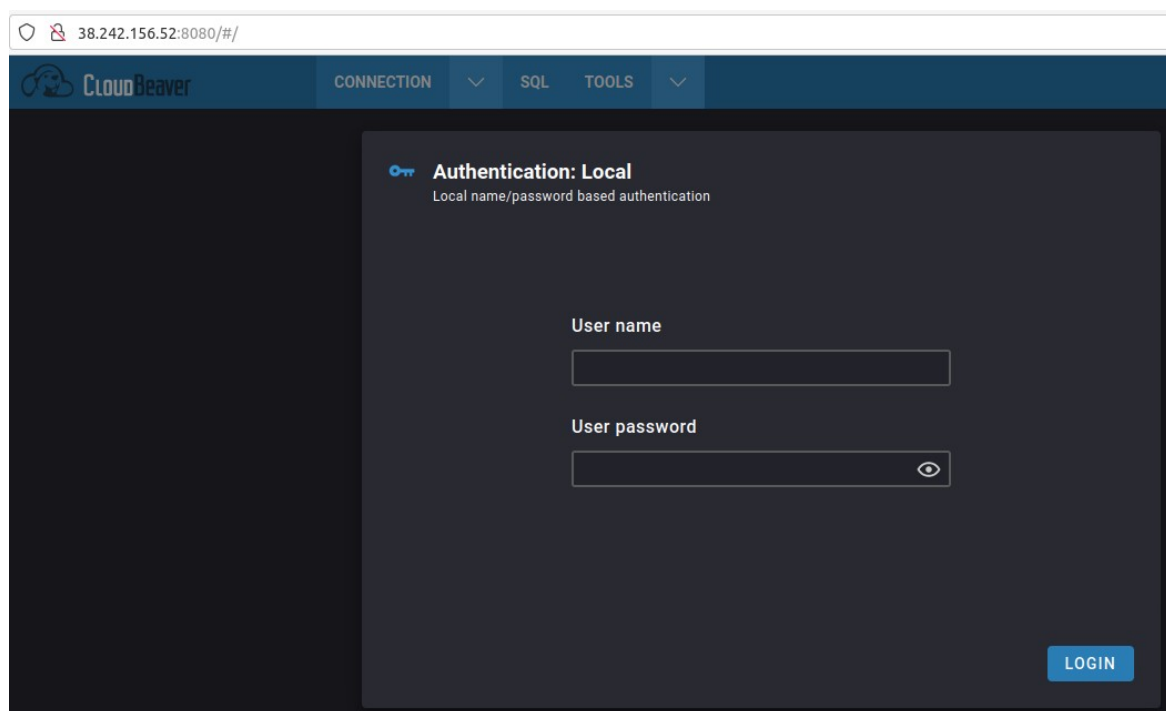
Para conectar ao banco de dados da Sprint 2, você também pode conectar-se via CloudBeaver. Ao acessar o moodle, clique no link “Web Database”

## Sprint 1

Nesta primeira sprint, conheceremos os conceitos básicos sobre Banco de Dados, Sistemas Gerenciadores de Banco de Dados (SGBDs), o que é SQL e seu histórico.

 Web Database	<input checked="" type="checkbox"/>
 Exercícios - Sprint 1	<input checked="" type="checkbox"/>
 Exercícios - Sprint 1	<input checked="" type="checkbox"/>
 Questões Bônus - Sprint 1	<input checked="" type="checkbox"/>
 Slides - Sprint 1	<input checked="" type="checkbox"/>
 Gravação da aula - sprint 1	<input checked="" type="checkbox"/>

Ao acessar o painel CloudBeaver, inclua as credenciais:



The screenshot shows the CloudBeaver web interface. The browser address bar displays the URL 38.242.156.52:8080/#/. The interface has a dark theme with a top navigation bar containing the CloudBeaver logo and tabs for CONNECTION, SQL, and TOOLS. The main content area is titled 'Authentication: Local' with the subtitle 'Local name/password based authentication'. It features two input fields: 'User name' and 'User password'. The 'User password' field includes a toggle icon for password visibility. A blue 'LOGIN' button is located in the bottom right corner of the authentication panel.

**Usuário:** cb\_alunos

**Senha:** crs&9Q#eyX&K6Gd

S2Q1) Ao executar a query **SELECT COUNT(id) FROM produto;**  
O resultado será :

- a) 10
- b) 25
- c) 100
- d) 200
- e) 250

S2Q2) Ao executar a query **SELECT COUNT(id) FROM cliente;**  
O resultado será :

- a) 10
- b) 25
- c) 100
- d) 200
- e) 250

S2Q3) Ao executar a query **SELECT COUNT(id) FROM nota\_fiscal;**  
O resultado será :

- a) 10
- b) 25
- c) 100
- d) 200
- e) 250

S2Q4) Ao executar a query **SELECT COUNT(id) FROM vendas;**  
O resultado será :

- a) 100
- b) 250
- c) 500
- d) 800
- e) 1000

S2Q5) Ao executar a query **SELECT id, ean\_13 FROM produto ORDER BY preco DESC LIMIT 5;**  
A primeira tupla encontrada será :

- a) 181 | 9986838978746
- b) 199 | 1631021120126
- c) 171 | 0565248721570
- d) 70 | 4049551183882
- e) 187 | 5187317946889

S2Q6) Para exibir o id da nota e o número de itens em cada nota na tabela vendas ordenado do maior número de itens para o menor, a query adequada é a opção:

- a) **SELECT nota\_id as nota, count(\*) as total FROM vendas GROUP BY nota\_id ORDER BY nota DESC;**
- b) **SELECT nota\_id as nota, count(\*) as total FROM vendas group BY nota\_id ORDER BY total DESC;**
- c) **SELECT nota\_id as nota, count(\*) as total FROM vendas group BY nota\_id ORDER BY total ASC;**
- d) **SELECT nota\_id as nota, count(\*) as total FROM vendas GROUP BY nota\_id ORDER BY nota;**
- e) **SELECT nota\_id as nota, count(\*) as total FROM vendas group BY nota\_id ORDER BY total;**

S2Q7) O resultado da query

**SELECT p.ean\_13 AS produto, v.quantidade AS quantidade , p.preco AS preco FROM vendas v, produto p WHERE p.id = v.produto\_id ORDER BY quantidade DESC**

é equivalente a query:

- a) **SELECT p.ean\_13 AS produto, v.quantidade AS quantidade , p.preco AS preco FROM vendas v, produto p WHERE p.id = v.produto\_id ORDER BY quantidade ASC**
- b) **SELECT p.ean\_13 AS produto, v.quantidade AS quantidade , p.preco AS preco FROM vendas v, produto p WHERE p.id = v.produto\_id ORDER BY quantidade**
- c) **SELECT p.ean\_13 AS produto, v.quantidade AS quantidade , p.preco AS preco FROM vendas v, produto p WHERE p.id = v.produto\_id ORDER BY 1 DESC**
- d) **SELECT p.ean\_13 AS produto, v.quantidade AS quantidade , p.preco AS preco FROM vendas v, produto p WHERE p.id = v.produto\_id ORDER BY 2 DESC**
- e) **SELECT p.ean\_13 AS produto, v.quantidade AS quantidade , p.preco AS preco FROM vendas v, produto p WHERE p.id = v.produto\_id ORDER BY 3 DESC**

S2Q8) A query **SELECT nf.numero AS NF, v.quantidade , p.ean\_13 , p.preco FROM vendas v, produto p, nota\_fiscal nf WHERE p.id = v.produto\_id and v.nota\_id = nf.id AND nf.numero = 1 order by 4** apresenta:

- a) Apresenta o número da nota fiscal, quantidade, código EAN-13 do produto e preço de todas as compras
- b) Apresenta o número da nota fiscal, quantidade, código EAN-13 do produto e preço da compra feita com a nota fiscal número quatro, ordenando-se pelo primeiro campo em ordem crescente.
- c) Apresenta o número da nota fiscal, quantidade, código EAN-13 do produto e preço da compra feita com a nota fiscal número quatro, ordenando-se pelo primeiro campo em ordem decrescente.
- d) Apresenta o número da nota fiscal, quantidade, código EAN-13 do produto e preço da compra feita com a nota fiscal número um, ordenando-se pelo quarto campo em ordem crescente.
- e) Apresenta o número da nota fiscal, quantidade, código EAN-13 do produto e preço da compra feita com a nota fiscal número um, ordenando-se pelo quarto campo em ordem decrescente.

S2Q9) O uso da função CONCAT está correto na query:

- a) **SELECT concat( c.nome '+' c.cpf ) AS pessoa FROM cliente c**
- b) **SELECT concat( c.nome , ' ', c.cpf ) AS pessoa FROM cliente c**
- c) **SELECT concat( c.nome . ' ' . c.cpf ) AS pessoa FROM cliente c**
- d) **SELECT concat( c.nome ' ' c.cpf ) AS pessoa FROM cliente c**
- e) **SELECT concat( c.nome & ' ' & c.cpf ) AS pessoa FROM cliente c**

S2Q10) Ao executar a query **SELECT id FROM produto ORDER BY preco DESC OFFSET 5 LIMIT 5** apresenta os seguintes ids:

- a) 199, 181, 171, 70, 187
- b) 125, 96, 61, 56, 160
- c) 86, 42, 182, 109, 37
- d) 183, 18, 52, 139, 192
- e) 21, 88, 13, 93, 117

S2Q11) Ao executar a query

**SELECT distribuidor, preco FROM produto ORDER BY id OFFSET 4 LIMIT 2**

apresenta as seguintes tuplas:

- a) (Freitas Barbosa – EI, 64.74), (Teixeira S.A., 42)
- b) (Duarte, 37.73), (Cavalcanti, 49.55)
- c) (Ribeiro Peixoto S.A., 68.53), ( da Paz, 24.22)
- d) (Dias – ME, 18.87), (da Cunha, 5.7)
- e) ( Oliveira e Filhos, 33.81), (Barros S/A, 45.76)

S2Q12) Ao executar a query **SELECT numero, cliente\_id FROM nota\_fiscal ORDER BY cliente\_id OFFSET 4 LIMIT 2** é retornado as seguintes tuplas:

- a) ((5, 17), (6, 18))
- b) ((96,22), (95, 6))
- c) ((12, 2), (97, 2))
- d) ((3, 24), (9,23))
- e) ((42, 2), (1, 2))

S2Q13) Ao executar a query **SELECT nome, cpf, endereco FROM cliente OFFSET 2 LIMIT 4** o resultado apresenta:

- a) Três registros e quatro campos
- b) Três registros e três campos
- c) Quatro registros e três campos
- d) Quatro registros e quatro campos
- e) Dois registros e três campos

S2Q14) Ao executar a query **SELECT DISTINCT nota\_id FROM vendas ORDER BY nota\_id** apresenta a seguinte quantidade de tuplas:

- a) 55
- b) 65
- c) 95
- d) 100
- e) 800

S2Q15) Para selecionar todos os ids das notas fiscais, ids dos produtos e quantidades que possuem a quantidade máxima, a query mais adequada é a opção:

- a) **SELECT nota\_id, produto\_id, max(quantidade) FROM vendas group by nota\_id**
- b) **SELECT nota\_id, produto\_id, max(quantidade) FROM vendas group by quantidade**
- c) **SELECT nota\_id, produto\_id, max(quantidade) FROM vendas**
- d) **SELECT nota\_id, produto\_id, quantidade FROM vendas where quantidade = max( quantidade)**
- e) **SELECT nota\_id, produto\_id, quantidade FROM vendas where quantidade = (SELECT max(quantidade) FROM vendas )**

S2Q16) Para selecionar o id da nota fiscal e a contagem de quantos itens possui cada nota fiscal, a query adequada é a opção:

- a) **SELECT nota\_id, count(quantidade) as quantidade FROM vendas GROUP BY quantidade ORDER BY 2 DESC**
- b) **SELECT nota\_id, count(quantidade) as quantidade FROM vendas GROUP BY nota\_id ORDER BY 2 DESC**
- c) **SELECT nota\_id, max(quantidade) as quantidade FROM vendas GROUP BY nota\_id ORDER BY 2 DESC**
- d) **SELECT nota\_id, min(quantidade) as quantidade FROM vendas GROUP BY nota\_id ORDER BY 2 DESC**
- e) **SELECT nota\_id, avg(quantidade) as quantidade FROM vendas GROUP BY quantidade ORDER BY 2 DESC**

S2Q17) Ao executar a query

```
SELECT nota_id,  
CASE  
    WHEN count(quantidade) > 10 THEN 'Caixa Normal'  
    ELSE 'Caixa Rápido'  
END AS condição FROM vendas  
where nota_id in (1,2,3)  
GROUP BY nota_id order by nota_id ;
```

O resultado esperado é:

- a) ((1, 'Caixa Rápido'), (2, 'Caixa Rápido'), (3, 'Caixa Rápido'))
- b) ((1, 'Caixa Rápido'), (2, 'Caixa Normal'), (3, 'Caixa Rápido'))
- c) ((1, 'Caixa Rápido'), (2, 'Caixa Normal'), (3, 'Caixa Normal'))
- d) ((1, 'Caixa Normal'), (2, 'Caixa Normal'), (3, 'Caixa Normal'))
- e) ((1, 'Caixa Normal'), (2, 'Caixa Rápido'), (3, 'Caixa Rápido'))

S2Q18) Ao executar a query

```
SELECT nota_id,  
CASE  
    WHEN count(quantidade) > 10 THEN 'Caixa Normal'  
    ELSE 'Caixa Rápido'  
END AS condição FROM vendas  
GROUP BY nota_id ORDER BY nota_id ;
```

O resultado esperado é:

- a) Encontrar todas os os ids das notas e a partir da quantidade de itens em cada nota, classificar como “Caixa Normal” ou “Caixa Rápido”
- b) Um erro de sintaxe próximo ao comando CASE
- c) Um erro de sintaxe próximo ao comando GROUP BY
- d) Um erro de sintaxe próximo ao comando ORDER BY
- e) Um erro de sintaxe próximo ao comando FROM

S2Q19) Para se obter um relatório do total de vendas por id de nota fiscal e a soma do total de venda de cada item ( quantidade \* preço), a query adequada é a opção:

- a) **SELECT nf.numero , sum(v.quantidade \* p.preco) as total FROM vendas v, nota\_fiscal nf , produto p where p.id = v.produto\_id group by nf.numero**
- b) **SELECT nf.numero , sum(v.quantidade \* p.preco) as total FROM vendas v, nota\_fiscal nf , produto p where v.nota\_id = nf.id and p.id = v.produto\_id group by nf.numero**
- c) **SELECT nf.numero , sum(v.quantidade \* p.preco) as total FROM vendas v, nota\_fiscal nf , produto p where v.nota\_id = nf.id group by nf.numero**
- d) **SELECT nf.numero , sum(v.quantidade \* p.id) as total FROM vendas v, nota\_fiscal nf , produto p where p.id = v.produto\_id group by nf.numero**
- e) **SELECT nf.numero , sum(v.quantidade \* p.preco) as total FROM vendas v, nota\_fiscal nf , produto p where v.nota\_id = nf.id and p.id = v.produto\_id order by 2 DESC**

S2Q20) Ao executar a query

```
SELECT nf.numero,p.ean_13, v.quantidade, p.preco, (v.quantidade * p.preco) as total  
FROM vendas v, nota_fiscal nf , produto p  
where v.nota_id = nf.id  
and p.id = v.produto_id  
and nf.numero = 20  
order by p.ean_13 ;
```

O resultado esperado é:

- a) Um relatório de todos os itens na nota fiscal número vinte ordenado pelo EAN do produto
- b) Um relatório de todos os itens de todas as notas fiscais ordenado pelo EAN do produto
- c) Um erro de sintaxe de SQL
- d) Um relatório de todos os itens na nota fiscal número vinte ordenado pelo número da nota fiscal
- e) Um relatório de todos os itens de todas as notas fiscais ordenado pelo número da nota fiscal

S2Q21) Ao executar a query

```
SELECT nf.numero,p.ean_13,p.distribuidor , v.quantidade, p.preco, (v.quantidade * p.preco) as total  
FROM vendas v, nota_fiscal nf , produto p  
WHERE v.nota_id = nf.id  
AND p.id = v.produto_id  
AND p.distribuidor = 'Azevedo'  
ORDER BY quantidade ;
```

É possível obter um relatório de vendas do distribuidor “Azevedo”. Qual o produto com menor quantidade de vendas ?

- a) 2004971104826
- b) 5119402964430
- c) 5706981021274
- d) 8700638543802
- e) 0720664680379

S2Q22) Ao executar a query

```
SELECT c.nome , c.cpf as CPF, nf.id as NOTA, p.ean_13  
FROM vendas v, nota_fiscal nf , produto p, cliente c  
where v.nota_id = nf.id  
and p.id = v.produto_id  
and c.id =nf.cliente_id  
AND p.ean_13 = '8173599959737'
```

É possível obter um relatório de todos os clientes que consumiram o produto ‘8173599959737’ Quais dos clientes listados consumiram o produto com o referido código EAN ?

- a) Dr. Diego Santos
- b) Lucas Martins
- c) Ana Vitória Teixeira
- d) Gabrielly da Costa
- e) João Felipe Silva

S2Q23) Ao executar a query

```
SELECT c.nome , c.cpf as CPF, nf.id as NOTA, p.ean_13  
FROM vendas v, nota_fiscal nf , produto p, cliente c  
where v.nota_id = nf.id  
and p.id = v.produto_id  
and c.id =nf.cliente_id  
AND p.ean_13 = '7182948303559'
```

É possível obter um relatório de todos os clientes que consumiram o produto '7182948303559' Quais dos clientes listados consumiram o produto com o referido código EAN ?

- a) Dr. Diego Santos
- b) Lucas Martins
- c) Ana Vitória Teixeira
- d) Gabrielly da Costa
- e) João Felipe Silva

S2Q24) Ao executar a query

```
SELECT DISTINCT ean_13 FROM produto ;
```

Qual a quantidade de produtos cadastrados no banco de dados ?

- a) 10
- b) 25
- c) 100
- d) 149
- e) 200

S2Q25) Ao executar a query

```
SELECT DISTINCT cliente_id FROM nota_fiscal ;
```

Qual a quantidade de clientes que possuem vínculo com pelo menos uma nota fiscal ?

- a) 10
- b) 25
- c) 100
- d) 149
- e) 200

S2Q26) Ao executar a query

```
SELECT p.ean_13, v.produto_id , v.quantidade, nf.id as nota FROM nota_fiscal nf JOIN vendas v ON  
nf.id = v.nota_id JOIN produto p ON v.produto_id =p.id where p.id = 155 order by nota
```

O resultado será o mesmo com a seguinte query:

- a) `select p.ean_13, v.produto_id , v.quantidade, nf.id as nota from nota_fiscal nf , vendas v , produto p where nf.id = v.nota_id and v.produto_id = p.id and p.id =155 order by 4`
- b) `select p.ean_13, v.produto_id , v.quantidade, nf.id as nota from nota_fiscal nf , vendas v , produto p where nf.id = v.nota_id and v.produto_id = p.id order by 155`
- c) `select p.ean_13, v.produto_id , v.quantidade, nf.id as nota from nota_fiscal nf , vendas v , produto p where v.produto_id = p.id and p.id =155 and nf.id = v.nota_id order by 1`
- d) `select p.ean_13 as nota, v.produto_id , v.quantidade, nf.id from nota_fiscal nf , vendas v , produto p where v.produto_id = p.id and p.id =155 and nf.id = v.nota_id order by nota`
- e) `select p.ean_13, v.produto_id , v.quantidade, nf.id from nota_fiscal nf , vendas v , produto p where v.produto_id = p.id and p.id =155 and nf.id = v.nota_id order by nota`



S2Q27) Ao executar a query

**SELECT vendas.nota\_id, vendas.nota\_id FROM vendas WHERE EXISTS (SELECT ean\_13 FROM produto WHERE produto.id = vendas.produto\_id)**

O resultado esperado é:

- a) Um erro de sintaxe
- b) Um relatório de todos os produtos presentes na tabela vendas
- c) Um relatório de todos os produtos presentes na tabela nota\_fiscal
- d) Um relatório de todos os produtos ausentes na tabela vendas
- e) Um relatório de todos os produtos ausentes na tabela nota\_fiscal

S2Q28) Ao executar a query

**SELECT c.nome, count(nf.numero) AS total FROM cliente c , nota\_fiscal nf WHERE c.id = nf.cliente\_id;**

O resultado esperado é:

- a) Um erro de sintaxe
- b) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota
- c) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota ordenado pelo nome
- d) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota ordenado pelo nome em ordem decrescente
- e) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota ordenado pelo total

S2Q29) Ao executar a query

**SELECT c.nome, count(nf.numero) AS total FROM cliente c , nota\_fiscal nf WHERE c.id = nf.cliente\_id GROUP BY c.id ORDER BY 2;**

O resultado esperado é:

- a) Um erro de sintaxe
- b) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota
- c) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota ordenado pelo nome
- d) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota ordenado pelo nome em ordem decrescente
- e) Um relatório de todos os clientes e o total de compras de cada cliente em cada nota ordenado pelo total

S2Q30) Ao executar a query

**SELECT c.nome, count(v.produto\_id) AS total FROM cliente c , nota\_fiscal nf , vendas v WHERE c.id = nf.cliente\_id AND nf.id = v.nota\_id GROUP BY c.id ORDER BY 2 DESC;**

O nome do cliente com maior quantidade de produto comprado é o cliente :

- a) Gabrielly Porto
- b) Sr. Rodrigo Cardoso
- c) Ryan da Conceição
- d) Dr. Vitor Farias
- e) Sra. Mariane Aragão

S2Q31) Ao executar a query

```
SELECT cpf, numero FROM cliente c JOIN nota_fiscal nf ON c.id =nf.cliente_id WHERE  
cpf='526.398.410-33';
```

Qual a quantidade de registro(s) encontrado(s):

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

S2Q32) Ao executar a query

```
SELECT cpf, numero FROM cliente c JOIN nota_fiscal nf ON nf.cliente_id = c.id WHERE  
cpf='526.398.410-33';
```

Qual a quantidade de registro(s) encontrado(s):

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

S2Q33) Ao executar a query

```
SELECT nf.numero as nota_num, v.produto_id FROM nota_fiscal nf JOIN vendas v ON nf.id =  
v.nota_id WHERE v.nota_id = 18;
```

Qual a quantidade de registro(s) encontrado(s):

- a) 0
- b) 2
- c) 4
- d) 6
- e) 8

S2Q34) Ao executar a query

```
SELECT nf.numero as nota_num, v.produto_id FROM nota_fiscal nf JOIN vendas v ON nf.id =  
v.nota_id WHERE v.nota_id = 60;
```

Qual a quantidade de registro(s) encontrado(s):

- a) 0
- b) 2
- c) 4
- d) 6
- e) 8

S2Q35) Ao executar a query

```
SELECT nf.numero as nota_num, v.produto_id  
FROM nota_fiscal nf JOIN vendas v ON nf.id = v.nota_id  
WHERE v.nota_id IN  
( SELECT nf.numero as nota_num  
FROM cliente c  
JOIN nota_fiscal nf  
ON c.id =nf.cliente_id  
WHERE cpf='526.398.410-33') order by 1;
```

O resultado esperado é:

- a) Um erro de sintaxe
- b) Um relatório de todas as notas e produto\_id do cliente com cpf número ' 526.398.410-33'
- c) Um relatório de todas as notas fiscais do cliente com cpf número ' 526.398.410-33'
- d) Um relatório de todas as vendas do produto com código ean número ' 526.398.410-33'
- e) Um relatório de todas as notas fiscais do produto com código ean número ' 526.398.410-33'

S2Q36) Ao executar a query

I.

```
SELECT nf.numero as nota_num, v.produto_id  
FROM nota_fiscal nf JOIN vendas v ON nf.id = v.nota_id  
WHERE v.nota_id IN  
( SELECT nf.numero as nota_num  
FROM cliente c  
JOIN nota_fiscal nf  
ON c.id =nf.cliente_id  
WHERE cpf='526.398.410-33') order by 1;
```

II.

```
SELECT nf.numero as nota_num, v.produto_id FROM nota_fiscal nf JOIN vendas v ON nf.id =  
v.nota_id JOIN cliente c ON c.id =nf.cliente_id WHERE cpf='526.398.410-33' order by 1;
```

III.

```
SELECT nf.numero as nota_num, v.produto_id FROM vendas v JOIN nota_fiscal nf ON nf.id =  
v.nota_id JOIN cliente c ON c.id =nf.cliente_id WHERE cpf='526.398.410-33' order by 1;
```

O mesmo resultado pode ser obtido com a seguinte query:

- a) Todas apresentam erros de sintaxe
- b) Somente a query (I) apresenta erro de sintaxe
- c) Nenhuma apresenta erro de sintaxe. A query II e III apresentam resultados equivalentes.
- d) Nenhuma apresenta erro de sintaxe. A query I e III apresentam resultados equivalentes.
- e) Nenhuma apresenta erro de sintaxe. Todas apresentam resultados equivalentes.

S2Q37) Durante uma pesquisa (*query*) em uma base de dados, as tabelas e colunas podem receber nomes temporários para facilitar a clareza da leitura da sequência de comandos. Este nome temporário é chamado de

- a) Redundância.
- b) Clone.
- c) Alias.
- d) Herança.
- e) Generalização.

S2Q38) Ao se utilizar a função COUNT em uma expressão SQL (Structured Query Language), deseja-se

- a) permitir a ordenação de uma pesquisa de forma ascendente e descendente.
- b) filtrar apenas alguns atributos de uma pesquisa genérica.
- c) obter a soma de um atributo numérico.
- d) obter o número de registros de acordo um critério específico.
- e) obter o menor valor de um conjunto de registros

S2Q39) O sistema de linguagem SQL (*Structured Query Language*, ou Linguagem de Consulta Estruturada) possui a seguinte característica:

- a) está centrado no conceito de adição ou remoção de servidores de banco de dados de forma escalável e sem interrupção dos serviços
- b) está baseado no relacionamento estruturado para endereços de nomes de domínios com seus respectivos endereços de máquina
- c) está baseado na pesquisa declarativa padrão para banco de dados relacional; muitas de suas características foram inspiradas na álgebra relacional
- d) está centrado em bancos de dados distribuídos, onde dados são armazenados em múltiplos pontos de processamento e normalmente em muitos servidores
- e) nenhuma das alternativas

S2Q40) Assinale, das alternativas abaixo, a única que identifica somente funções de agregação usadas dentro de uma cláusula SELECT do SQL (*Structured Query Language*):

- a) AVG – COUNT – SUM – MAX – MIN
- b) AVG – LIKE – UNION – MAX – MIN
- c) AVG – COUNT – AND – MAX – LIKE
- d) FROM – COUNT – SUM – MAX – MIN
- e) FROM – COUNT – SUM – AND – UNION

## Bônus – Sprint 2

41 ) Observe a query a seguir:

```
SELECT p.ean_13 , v.quantidade , v.nota_id FROM vendas v JOIN produto p ON  
v.produto_id = p.id WHERE v.nota_id = 30
```

Reescreva a query acima **sem o uso** de join

42 ) Observe a query a seguir:

```
SELECT nf.id , v.quantidade , v.produto_id FROM vendas v JOIN nota_fiscal nf ON nf.id =  
v.nota_id WHERE v.nota_id = 30
```

Reescreva a query acima **sem o uso** de join

43 ) Observe a query a seguir:

```
SELECT nf.id , p.ean_13 , p.preco , v.quantidade FROM vendas v JOIN nota_fiscal nf ON  
nf.id = v.nota_id JOIN produto p ON p.id = v.produto_id WHERE v.nota_id = 30
```

Reescreva a query acima **sem o uso** de join

44 ) Observe a query a seguir:

```
SELECT nf.id , c.nome , p.ean_13 , p.preco, v.quantidade FROM vendas v JOIN nota_fiscal nf ON nf.id = v.nota_id JOIN produto p on p.id = v.produto_id JOIN cliente c on nf.cliente_id = c.id WHERE v.nota_id = 30
```

Reescreva a query acima **sem o uso** de join

45 ) Observe a query a seguir:

```
SELECT nf.id as nota, c.nome , p.ean_13 , p.preco, v.quantidade FROM vendas v JOIN nota_fiscal nf ON nf.id = v.nota_id JOIN produto p on p.id = v.produto_id JOIN cliente c on nf.cliente_id = c.id WHERE c.nome like 'A%' order by nota
```

Reescreva a query acima **sem o uso** de join

46 ) Observe a query a seguir:

```
select p.ean_13 , v.quantidade , v.nota_id from vendas v, produto p where v.produto_id = p.id and v.nota_id = 54;
```

Reescreva a query acima **com o uso** de join

47 ) Observe a query a seguir:

```
select nf.id , v.quantidade , v.produto_id from vendas v, nota_fiscal nf WHERE nf.id = v.nota_id AND v.nota_id = 54
```

Reescreva a query acima **com o uso** de join

48 ) Observe a query a seguir:

```
select nf.id , p.ean_13 , p.preco , v.quantidade from vendas v, nota_fiscal nf , produto p WHERE nf.id = v.nota_id AND p.id = v.produto_id AND v.nota_id = 54
```

Reescreva a query acima **com o uso** de join

49 ) Observe a query a seguir:

```
select nf.id , c.nome , p.ean_13 , p.preco, v.quantidade from vendas v, nota_fiscal nf, cliente c , produto p WHERE nf.id = v.nota_id AND p.id = v.produto_id AND nf.cliente_id = c.id AND v.nota_id = 54
```

Reescreva a query acima **com o uso** de join

50 ) Observe a query a seguir:

```
SELECT nf.id as nota, c.nome , p.ean_13 , p.preco, v.quantidade FROM vendas v, nota_fiscal nf, produto p , cliente c WHERE nf.id = v.nota_id and p.id = v.produto_id AND nf.cliente_id = c.id AND c.nome like 'Dr.%' and c.nome like '%Farias' order by nota
```

Reescreva a query acima **com o uso** de join