**Prova SQL Samsung**

**Parte 1**

1 – C

2 – D

3 – D

4 – B

5 – E

6 – A

7 – D

8 – A

9 – C

10 – D

**Parte 2**

1 – O ACID impede que os dados do sei Banco de Dados sejam corrompidos

2 – As operações são CREATE, READ, UPDATE e DELETE e em português ficaria Criar, Ler, Atualizar e Remover

3 –

4 – BLOB – Armazena até 4GB de dados Binários, útil para armazenar imagens, áudio ou vídeo.

CLOB – Armazena até 4 GB de dados, útil para armazenar texto

5 – VIEW – Usada para consultas ou restrição de acesso

MATERILIZED VIEW – Armazena dados e melhora o desempenho de uma consulta

**Parte 3**

1 – SELECT \* FROM Cliente;

2 – SELECT

c.Cliente\_ID,

c.Nome AS Nome,

i.Descricao

FROM

Cliente c

JOIN

Nota\_Fiscal nf ON c.Cliente\_ID = nf.Cliente\_ID

JOIN

Nota\_Fiscal\_Item nfi ON nf.Nota\_Numero\_ID = nfi.Nota\_Numero\_ID

JOIN

Item i ON nfi.Item\_ID = i.Item\_ID

WHERE

i.Descricao = ‘NP-123’;

3 – SELECT

c.Cliente\_ID AS ID,

c.Nome AS Nome,

COUNT(nfi.Quantidade) AS QtdProdutosComprados

FROM

Cliente c,

JOIN

Nota\_Fiscal nf ON c.Cliente\_ID = nf.Cliente\_ID

JOIN

Nota\_Fiscal\_Itens nfi ON nf.Nota\_Numero\_ID = nfi.Nota\_Numero\_ID

JOIN

Item i ON nfi.Item\_ID = i.Item\_ID

GROUP BY

c.Cliente\_ID, c.Nome;

4 – Basicamente é realizado o JOIN até chegar na tabela de Item com isso já iremos ter os resultados atendendo o relacionamento de todas as tabelas necessárias, por final é agrupado pelo Cliente onde teremos pra cada cliente a quantidade total de produtos comprados.

5 – UPDATE Item

SET Volume = Volume \* 3;

COMMIT;

**Prova Java Samsung**

**Parte 1**

1 – B

2 – E

3 – V

F

F

F

V

V

4 – V

F

V

V

5 – D

**Parte 2**

a)

A diagram of a flowchart

Description automatically generated

b)

|  |  |
| --- | --- |
| **SALBASE** | **GRATIF** |
| 3000 | 1200 |
| 1200 | 400 |
| 500 | 100 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SALBASE** | **GRATIF** | **SALBRUTO** | **IR** | **SALLIQ** |
| 3000 | 1200 | 4200 | 630 | 3570 |
| 1200 | 400 | 1600 | 240 | 1360 |
| 500 | 100 | 600 | 120 | 480 |

|  |
| --- |
| **SALLIQ** |
| 3360 |
| 1360 |
| 480 |

**Parte 3 (Java)**

1 – E

2 – E

3 –

**Public** **static** **X** main(String[] args) {

Carros carro = **new** Carros("Fiat", "Stilo", "2007");

String[] opcoes = **new** String **X**{"Transmissão Manual", "1.8 cc", "Preto"};

carro.setOptions(opcoes)**X**);

Carros carro2 = **new** Carros("Chevrolet", "Vectra", "2010");

String[]opcoes2 = **new** String **X**{"Transmissão Automática","2.0 cc","Cinza"}

carro2.setOptions(options2);

**X** system.*out*.println("Os carros 1 e 2 são iguais(): " , carro2.equals(carro2));

}

4 – C

5 – E

6 – B

7 – A, B, C