1. **Considere o seguinte esquema relacional.**

**ItemVenda**(NumeroNF, NumeroProd, QtdeItem, PrecoItem)

NumeroProd referencia Produto

NumeroNF referencia Venda

**Produto** (NumeroProd, CodigoTipoProd, DescricaoProd)

CodigoTipoProd referencia TipoProd

**TipoProd**(CodigoTipoProd, DescricaoTipoProd)

**Venda**(NumeroNF, DataVenda, CodEmp)

CodEmp referencia Empregado

**Empregado**(CodEmp, NomeEmp, CodDept, Salario, DataAdmissao)

CodDept referencia Departamento

**Departamento**(CodDept, NomeDepartamento)

**2. Comandos para criação das tabelas.**

CREATE SEQUENCE CodDept \_sq1

MINVALUE 1

MAXVALUE 30

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE CodigoTipoProd\_sq1

MINVALUE 1

MAXVALUE 2000

START WITH 1

INCREMENT BY 1

CACHE 150

NOCYCLE;

CREATE SEQUENCE CodEmp\_sq1

MINVALUE 1

MAXVALUE 100

START WITH 1

INCREMENT BY 1

CACHE 10

NOCYCLE;

CREATE SEQUENCE NumeroNF\_sq1

MINVALUE 1

MAXVALUE 10000000000

START WITH 1

INCREMENT BY 1

CACHE 100

NOCYCLE;

CREATE SEQUENCE NumeroProd\_sq1

MINVALUE 1

MAXVALUE 1000000

START WITH 1

INCREMENT BY 1

CACHE 100

NOCYCLE;

-------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Departamento (

CodDept NUMBER NOT NULL,

NomeDepartamento VARCHAR(255) NOT NULL,

CONSTRAINT PK\_CodDept PRIMARY KEY (CodDept )

);

CREATE TABLE TipoProd(

CodigoTipoProd NUMBER NOT NULL,

DescricaoTipoProd VARCHAR(510),

CONSTRAINT PK\_CodigoTipoProd PRIMARY KEY (CodigoTipoProd )

);

CREATE TABLE Empregado (

CodEmp NUMBER NOT NULL,

CodDept NUMBER,

NomeEmp VARCHAR(255) NOT NULL,

Salario NUMBER(4,2),

DataAdmissao DATE NOT NULL,

CONSTRAINT PK\_CodEmp PRIMARY KEY ( CodEmp ),

CONSTRAINT FK\_CodDept FOREIGN KEY (CodDept ) REFERENCES Departamento(CodDept )

);

CREATE TABLE Venda(

NumeroNF NUMBER NOT NULL,

CodEmp NUMBER,

DataVenda DATE NOT NULL,

CONSTRAINT PK\_NumeroNF PRIMARY KEY (NumeroNF),

CONSTRAINT FK\_CodEmp FOREIGN KEY (CodEmp ) REFERENCES Empregado(CodEmp)

);

CREATE TABLE Produto(

NumeroProd NUMBER NOT NULL,

CodigoTipoProd NUMBER NOT NULL,

DescricaoProd VARCHAR(510),

CONSTRAINT PK\_NumeroProd PRIMARY KEY (NumeroProd ),

CONSTRAINT FK\_CodigoTipoProd FOREIGN KEY (CodigoTipoProd) REFERENCES TipoProd(CodigoTipoProd)

);

CREATE TABLE ItemVenda(

NumeroNF NUMBER NOT NULL,

NumeroProd NUMBER NOT NULL,

QtdeItem NUMBER NOT NULL,

PrecoItem NUMBER NOT NULL,

CONSTRAINT PK\_NumeroNF PRIMARY KEY(NumeroNF ,NumeroProd),

CONSTRAINT FK\_NumeroNF FOREIGN KEY (NumeroNF) REFERENCES Venda(NumeroNF),

CONSTRAINT FK\_NumeroProd FOREIGN KEY (NumeroProd) REFERENCES Produto(NumeroProd)

);

**3. Comandos de inserção de 5 registros em cada tabela.**

INSERT INTO Departamento VALUES (CodDept \_sq1.NEXTVAL, RH);

INSERT INTO Departamento VALUES (CodDept \_sq1.NEXTVAL, Almoxarifado);

INSERT INTO Departamento VALUES (CodDept \_sq1.NEXTVAL, TI);

INSERT INTO Departamento VALUES (CodDept \_sq1.NEXTVAL, LOJA);

INSERT INTO Departamento VALUES (CodDept \_sq1.NEXTVAL, LIMPEZA);

-------------------------------------------------------------------------------------------------------------------------------------------------

INSERT INTO TipoProd VALUES (CodigoTipoProd\_sq1.NEXTVAL, SMARTPHONE);

INSERT INTO TipoProd VALUES (CodigoTipoProd\_sq1.NEXTVAL, NOTEBOOK );

INSERT INTO TipoProd VALUES (CodigoTipoProd\_sq1.NEXTVAL, TV);

INSERT INTO TipoProd VALUES (CodigoTipoProd\_sq1.NEXTVAL, MONITOR);

INSERT INTO TipoProd VALUES (CodigoTipoProd\_sq1.NEXTVAL, COMPUTADOR DESKTOP);

-------------------------------------------------------------------------------------------------------------------------------------------------

INSERT INTO Empregado VALUES (

CodEmp\_sq1.NEXTVAL,

1,

‘Anderson Fuhr Souza’,

5000.00,

‘25/DEZ/2007’,

);

INSERT INTO Empregado VALUES (

CodEmp\_sq1.NEXTVAL,

2,

‘Gabriele Peschke Baum’,

5000.00,

‘12/AGO/2015’,

);

INSERT INTO Empregado VALUES (

CodEmp\_sq1.NEXTVAL,

3,

‘Jhonata Ávila’,

5000,00,

‘05/JUL/2011’,

);

INSERT INTO Empregado VALUES (

CodEmp\_sq1.NEXTVAL,

5,

‘Maria de Souza’,

1500.00,

‘01/AGO/2018’,

);

INSERT INTO Empregado VALUES (

CodEmp\_sq1.NEXTVAL,

4,

‘Mario Antonio de Souza’,

3000.00,

‘11/DEZ/2009’,

);

-------------------------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------------------------

INSERT INTO Venda VALUES (

NumeroNF\_sq1.NEXTVAL,

5,

‘10/JAN/2010’,

);

INSERT INTO Venda VALUES (

NumeroNF\_sq1.NEXTVAL,

5,

‘10/JAN/2010’,

);

INSERT INTO Venda VALUES (

NumeroNF\_sq1.NEXTVAL,

5,

‘11/JAN/2010’,

);

INSERT INTO Venda VALUES (

NumeroNF\_sq1.NEXTVAL,

5,

‘15/JAN/2010’

);

INSERT INTO Venda VALUES (

NumeroNF\_sq1.NEXTVAL,

5,

‘16/JAN/2010’,

);

-------------------------------------------------------------------------------------------------------------------------------------------------

INSERT INTO Produto VALUES (

NumeroProd\_sq1.NEXTVAL,

1,

‘Samsung Galaxy S10 Azul, com Tela Infinita de 6,1", 4G, 128GB e Câmera Tripla de 12MP+12MP+16MP - SM-G973FZBJZTO’,

);

INSERT INTO Produto VALUES (

NumeroProd\_sq1.NEXTVAL,

1,

‘Samsung Galaxy S10 Plus Ceramic Black, Tela Infinita 6,4", 4G, 128GB e Câmera Tripla de 12MP+16MP+12MP - SM-G975FCKJZTO’,

);

INSERT INTO Produto VALUES (

NumeroProd\_sq1.NEXTVAL,

2,

‘Notebook Lenovo, Intel® Core™ i7-8265U, 8GB, 256GB, 15,6", Placa NVIDIA® GeForce® MX110, IdeaPad S145 - 81S9000HBR’,

);

INSERT INTO Produto VALUES (

NumeroProd\_sq1.NEXTVAL,

3,

‘Samsung Smart TV Crystal UHD TU8000 4K 75", Borda Infinita, Alexa built in, Controle Único, Visual Livre de Cabos’,

);

INSERT INTO Produto VALUES (

NumeroProd\_sq1.NEXTVAL,

4,

‘Kit Monitor Dell LED Full HD IPS 23.8'' SE2419HR + Mouse Wireless Sem Fio e Bluetooth MS3320W’,

);

-------------------------------------------------------------------------------------------------------------------------------------------------

INSERT INTO ItemVenda VALUES (

1,

1,

1,

4999.00

);

INSERT INTO ItemVenda VALUES (

2,

3,

1,

2000.00,

);

INSERT INTO ItemVenda VALUES (

3,

1,

1,

4999.00,

);

INSERT INTO ItemVenda VALUES (

4,

1,

2,

4999.00,

);

INSERT INTO ItemVenda VALUES (

5,

4,

5,

5500.00,

);

**4. Mostrar a quantidade de produtos cadastrados**

SELECT COUNT **(NumeroProd)**

FROM **Produto;**

**5. Mostrar o nome do produto e a descrição do tipo do produto para todos os produtos cadastrados.**

SELECT **Produto.DescricaoProd, TipoProd.DescricaoTipoProd**

FROM **Produto**

LEFT JOIN **TipoProd** ON

**Produto.CodigoTipoProd =**

**TipoProd.CodigoTipoProd;**

**6. Mostrar a quantidade de venda por nome empregado (ou seja, para cada empregado apresentar a sua quantidade de venda).**

**Venda**(NumeroNF, DataVenda, CodEmp)

SELECT Emp.NomeEmp, COUNT(\*) as TotalVendas

FROM Empregado as Emp

INNER JOIN Venda ON

Venda.CodEmp = Emp.CodEmp

GROUP BY Emp.NomeEmp;

**7. Mostrar a quantidade de venda por nome empregado (ou seja, para cada empregado apresentar a sua quantidade de venda), considerando quando o empregado realizou duas ou mais vendas.**

SELECT Emp.NomeEmp, COUNT(\*) as TotalVendas

FROM Empregado as Emp

INNER JOIN Venda ON

Venda.CodEmp = Emp.CodEmp

GROUP BY Emp.NomeEmp

HAVING COUNT(\*) >= 2;

**8. Mostrar o nome do empregado, o nome do departamento e a data de admissão dos empregados que possuem o maior salário.**

SELECT Emp.NomeEmp, Emp.DataAdmissao, Departamento.NomeDepartamento

FROM Empregado as Emp

LEFT [OUTER] JOIN Departamento ON

Emp.CodDept = Departamento.CodDept

WHERE Emp.Salario = (SELECT MAX(Emp2.Salario) FROM Empregado Emp2);

OBS: Usamos a função de agregação MAX pois pede o “Maior salário” e não os maiores acima da média! Bem como usamos o sinal de “=” pois haverá apenas um maior salário e não deve haver ninguém mais acima deste, somente igual.

**9. Mostrar o número da NF e o nome do empregado que realizou aquela venda, para todas as vendas realizadas em uma data específica.**

SELECT Venda.NumeroNF, Emp.NomeEmp

FROM Empregado as Emp

RIGHT JOIN Venda ON

Emp.CodEmp = Venda.CodEmp

WHERE Venda.DataVenda = ‘16/JAN/2010’;

**10. Crie as seguintes visões (View):**

**10.1. com o número do produto, CodigoTipoProd e a descrição do tipo de produto. Após, apresente uma consulta a essa view**

CREATE VIEW DadosProdView as

SELECT p.NumeroProd, p.CodigoTipoProd, tp.DescricaoTipoProd

FROM Produto p

INNER JOIN TipoProd tp ON

p.CodigoTipoProd = tp.CodigoTipoProd;

SELECT NumeroProd

FROM DadosProdView

WHERE CodigoTipoProd = 1 ;

**10.2. com a quantidade total de venda por produto. Deve ser feita a multiplicação entre quantidade de 1itens e Preco de item. Após, apresente uma consulta a essa view.**

CREATE VIEW DadosProdVendaView

SELECT Produto.DescricaoProd, SUM((ItemVenda.QtdeItem \* ItemVenda.PrecoItem)) as ValorTotalVendas

FROM ItemVenda

INNER JOIN Produto ON

ItemVenda.NumeroProd = Produto.NumeroProd

GROUP BY Produto.DescricaoProd;

SELECT ValorTotalVendas

FROM DadosProdVendaView

WHERE DescricaoProd = ’Samsung Smart TV Crystal UHD TU8000 4K 75", Borda Infinita, Alexa built in, Controle Único, Visual Livre de Cabos’;

1