



**UGF - Universidade Gama Filho**

Campus Piedade – T.305/2012.2 – Período da Noite

Prof. Cláudio Passos INF434 – Banco de Dados II

## **PROJETO DB MITSAN**

:

Alunos: Rodrigo Alues de Souza Mat. 2011107620-4

Sérgio da Silva Pereira Mat. 2010160941-8

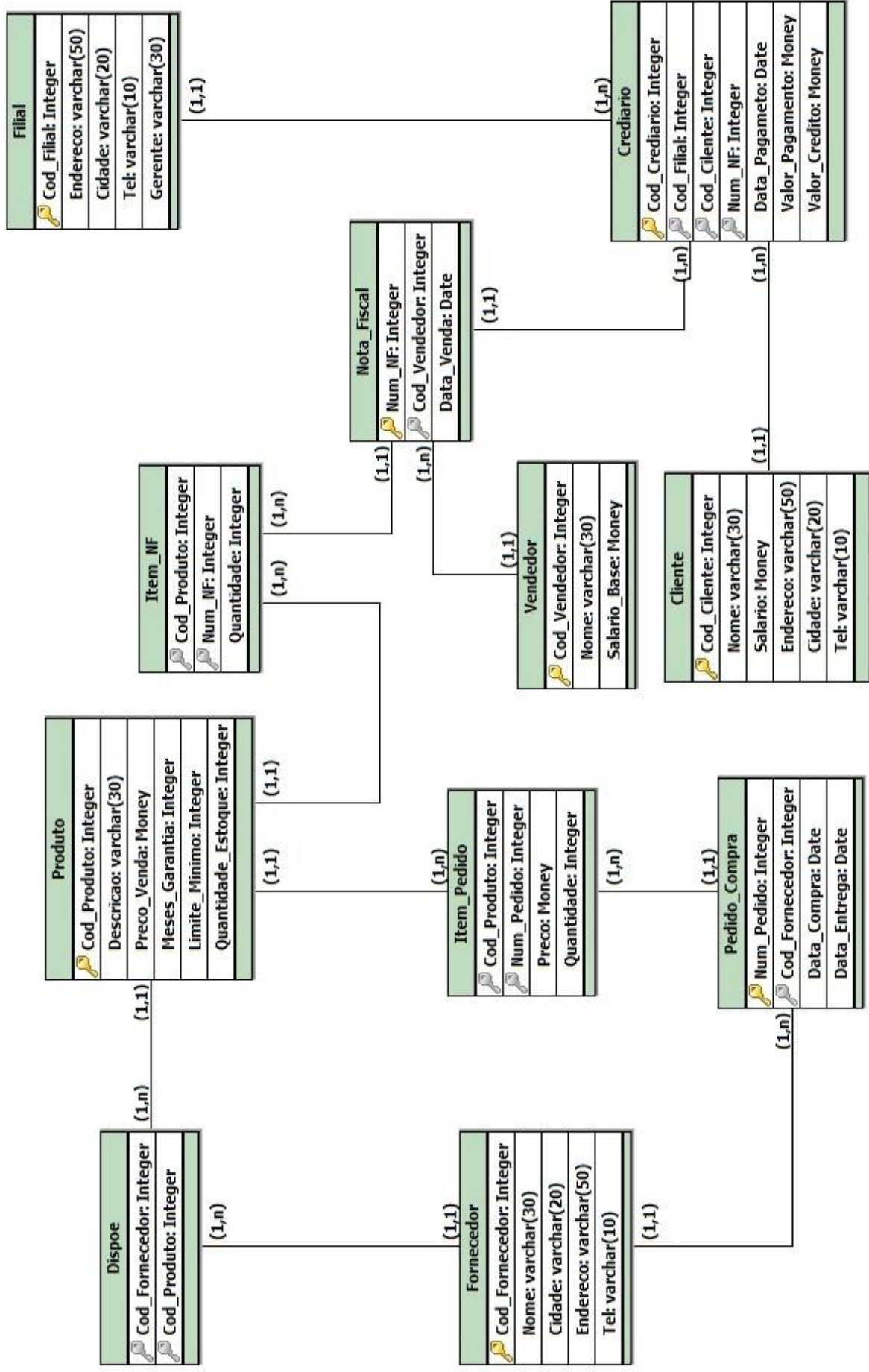
**Rio de Janeiro – Novembro de 2012**

## **1 – Objetivo:**

- a) Diagrama de entidade e relacionamento;
- b) Os create's tables;
- c) Os devidos insert's
- d) As consultas;
- e) A relação dos trigger's;
- f) Os grant's da parte de segurança.

## **2 – Diagramas:**





### 3 – CREATE'S TABLES

```
CREATE TABLE Produto (  
Cod_Produto Integer PRIMARY KEY,  
Descricao varchar(30),  
Preco_Venda Money,  
Meses_Garantia Integer,  
Limite_Minimo Integer,  
Quantidade_Estoque Integer  
);
```

```
CREATE TABLE Item_Pedido (  
Cod_Produto Integer,  
Num_Pedido Integer,  
Preco_Compra Money,  
Quantidade Integer,  
FOREIGN KEY(Cod_Produto) REFERENCES Produto (Cod_Produto) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Filial (  
Cod_Filial Integer PRIMARY KEY,  
Endereco varchar(50),  
Cidade varchar(20),  
Tel varchar(10),  
Gerente varchar(30)  
);
```

```
CREATE TABLE Vendedor (  
Cod_Vendedor Integer PRIMARY KEY,  
Nome varchar(30),  
Salario_Base Money  
);
```

```
CREATE TABLE Nota_Fiscal (  
Num_NF Integer PRIMARY KEY,  
Cod_Vendedor Integer,  
Data_Venda Date,  
FOREIGN KEY(Cod_Vendedor) REFERENCES Vendedor (Cod_Vendedor) ON UPDATE  
CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE Item_NF (  

```

```
Cod_Produto Integer,  
Num_NF Integer,  
Quantidade Integer,  
FOREIGN KEY(Cod_Produto) REFERENCES Produto (Cod_Produto) ON UPDATE CASCADE  
ON DELETE CASCADE,  
FOREIGN KEY(Num_NF) REFERENCES Nota_Fiscal (Num_NF) ON UPDATE CASCADE ON  
DELETE CASCADE  
);
```

```
CREATE TABLE Fornecedor (  
Cod_Fornecedor Integer PRIMARY KEY,  
Nome varchar(30),  
Cidade varchar(20),  
Endereco varchar(50),  
Tel varchar(10)  
);
```

```
CREATE TABLE Pedido_Compra (  
Num_Pedido Integer PRIMARY KEY,  
Cod_Fornecedor Integer,  
Data_Compra Date,  
Data_Entrega Date,  
FOREIGN KEY(Cod_Fornecedor) REFERENCES Fornecedor (Cod_Fornecedor) ON UPDATE  
CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE Dispoe (  
Cod_Fornecedor Integer,  
Cod_Produto Integer,  
FOREIGN KEY(Cod_Fornecedor) REFERENCES Fornecedor (Cod_Fornecedor) ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY(Cod_Produto) REFERENCES Produto (Cod_Produto) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Cliente (  
Cod_Cliente Integer PRIMARY KEY,  
Nome varchar(30),  
Salario Money,  
Endereco varchar(50),  
Cidade varchar(20),  
Tel varchar(10)  
);
```

```

CREATE TABLE Crediario (
Cod_Crediario Integer PRIMARY KEY,
Cod_Filial Integer,
Cod_Cliente Integer,
Num_NF Integer,
FOREIGN KEY(Cod_Filial) REFERENCES Filial (Cod_Filial) ON UPDATE CASCADE ON
DELETE CASCADE,
FOREIGN KEY(Cod_Cliente) REFERENCES Cliente (Cod_Cliente) ON UPDATE CASCADE
ON DELETE CASCADE,
FOREIGN KEY(Num_NF) REFERENCES Nota_Fiscal (Num_NF) ON UPDATE CASCADE ON
DELETE CASCADE
);

```

```

CREATE TABLE Fatura (
Cod_Faturamento Integer PRIMARY KEY,
Cod_Crediario Integer,
Data_Vencimento Date,
Valor_Fatura Money,
Data_Pagamento Date,
Valor_Pago Date,
FOREIGN KEY(Cod_Crediario) REFERENCES Crediario (Cod_Crediario) ON UPDATE
CASCADE
);

```

```

ALTER TABLE Item_Pedido ADD FOREIGN KEY(Num_Pedido) REFERENCES Pedido_Compra
(Num_Pedido) ON UPDATE CASCADE ON DELETE CASCADE;

```

#### 4 – Insert's

```

INSERT INTO Vendedor(Cod_Vendedor, Nome, Salario_Base) VALUES(1, 'Roberto
Fontes', 1200);
INSERT INTO Vendedor(Cod_Vendedor, Nome, Salario_Base) VALUES(2, 'Gilson
Pedrosa', 1300);
INSERT INTO Vendedor(Cod_Vendedor, Nome, Salario_Base) VALUES(3, 'João
Caldas', 890);
INSERT INTO Vendedor(Cod_Vendedor, Nome, Salario_Base) VALUES(4, 'José
Medeiros', 1100);
INSERT INTO Vendedor(Cod_Vendedor, Nome, Salario_Base) VALUES(5, 'Wilson
Ferreira', 1230);

```

---

```
INSERT INTO Cliente(Cod_Cliente, Nome, Salario, Endereco, Cidade, Tel)
VALUES(1, 'Simone Neves', 15076.94, 'Rua do Ouvidor, 35', 'Rio de Janeiro',
'3908-7848');
INSERT INTO Cliente(Cod_Cliente, Nome, Salario, Endereco, Cidade, Tel)
VALUES(2, 'Beth Coutinho', 4945.60, 'Rua das Flores, 9', 'Rio de Janeiro',
'5743-7643');
INSERT INTO Cliente(Cod_Cliente, Nome, Salario, Endereco, Cidade, Tel)
VALUES(3, 'Irene Rodrigues', 1508.56, 'Rua Pau d'Água, 39', 'Petrópolis',
'2343-6754');
INSERT INTO Cliente(Cod_Cliente, Nome, Salario, Endereco, Cidade, Tel)
VALUES(4, 'Rafael Souza', 635.89, 'Av. Quebra Vento', 'Duque de Caxias',
'6468-8976');
INSERT INTO Cliente(Cod_Cliente, Nome, Salario, Endereco, Cidade, Tel)
VALUES(5, 'Miguel Bastos', 38240.27, 'Est. Rio D'Ouro', 'Paraty', '2345-
6753');
```

---

```
INSERT INTO Filial(Cod_Filial, Endereco, Cidade, Tel, Gerente) VALUES(1, 'Rua
Gal. Osório, 30', 'Rio de Janeiro', '2345-6734', 'Peixoto Correia');
INSERT INTO Filial(Cod_Filial, Endereco, Cidade, Tel, Gerente) VALUES(2, 'Rua
do Imperador, 30', 'Petrópolis', '6734-2345', 'Roberto Salmão');
INSERT INTO Filial(Cod_Filial, Endereco, Cidade, Tel, Gerente) VALUES(3, 'Rua
Sarapuí, 62', 'Duque de Caxias', '4563-6785', 'Gustavo Frei');
INSERT INTO Filial(Cod_Filial, Endereco, Cidade, Tel, Gerente) VALUES(4, 'Av.
Brasil, 14062', 'Rio de Janeiro', '3890-5432', 'Samuel Gouveia');
INSERT INTO Filial(Cod_Filial, Endereco, Cidade, Tel, Gerente) VALUES(5,
'Est. dos Tindibas, 104', 'Rio de Janeiro', '3398-9483', 'Renato Pessoa');
```

---

```
INSERT INTO Fornecedor(Cod_Fornecedor, Nome, Cidade, Endereco, Tel) VALUES(1,
'Rei do Rio Construções Ltda.', 'Rio de Janeiro', 'Est. dos Três Rios, 1234',
'4322-7643');
INSERT INTO Fornecedor(Cod_Fornecedor, Nome, Cidade, Endereco, Tel) VALUES(2,
'Fábrica de Cimento TUPI S/A', 'Rio de Janeiro', 'Av. Paraná, 2453', '4226-
6532');
INSERT INTO Fornecedor(Cod_Fornecedor, Nome, Cidade, Endereco, Tel) VALUES(3,
'Pedreira Rocha Lascada S/A.', 'Teresópolis', 'Trv. Projetada, s/n', '2345-
76544');
INSERT INTO Fornecedor(Cod_Fornecedor, Nome, Cidade, Endereco, Tel) VALUES(4,
'Cerâmica Barro Firme Ltda.', 'Magé', 'Prç. José Barroso, 32', '7546-7654');
INSERT INTO Fornecedor(Cod_Fornecedor, Nome, Cidade, Endereco, Tel) VALUES(5,
'Wamel Ferramentas Ltda.', 'Volta Redonda', 'Av. Aço Tenaz, 238', '5678-
5468');
```

---



```
INSERT INTO Produto(Cod_Produto, Descricao, Preco_Venda, Meses_Garantia,
Limite_Minimo, Quantidade_Estoque) VALUES(1, 'Cimento', 100.00, 1, 20, 100);
INSERT INTO Produto(Cod_Produto, Descricao, Preco_Venda, Meses_Garantia,
Limite_Minimo, Quantidade_Estoque) VALUES(2, 'Areia', 60.00, 1, 50, 500);
INSERT INTO Produto(Cod_Produto, Descricao, Preco_Venda, Meses_Garantia,
Limite_Minimo, Quantidade_Estoque) VALUES(3, 'Brita', 80.00, 3, 70, 700);
INSERT INTO Produto(Cod_Produto, Descricao, Preco_Venda, Meses_Garantia,
Limite_Minimo, Quantidade_Estoque) VALUES(4, 'Tijolo', 1.20, 3, 1000, 3000);
INSERT INTO Produto(Cod_Produto, Descricao, Preco_Venda, Meses_Garantia,
Limite_Minimo, Quantidade_Estoque) VALUES(5, 'Penera', 15.00, 3, 5, 50);
```

---

```
INSERT INTO Pedido_Compra(Num_Pedido, Cod_Fornecedor, Data_Compra,
Data_Entrega) VALUES(1, 3, '21/10/12', '28/11/12');
INSERT INTO Pedido_Compra(Num_Pedido, Cod_Fornecedor, Data_Compra,
Data_Entrega) VALUES(2, 5, '25/10/12', '27/11/12');
INSERT INTO Pedido_Compra(Num_Pedido, Cod_Fornecedor, Data_Compra,
Data_Entrega) VALUES(3, 1, '27/10/12', '30/11/12');
INSERT INTO Pedido_Compra(Num_Pedido, Cod_Fornecedor, Data_Compra,
Data_Entrega) VALUES(4, 4, '28/10/12', '16/11/12');
INSERT INTO Pedido_Compra(Num_Pedido, Cod_Fornecedor, Data_Compra,
Data_Entrega) VALUES(5, 2, '30/10/12', '20/11/12');
```

---

```
INSERT INTO Item_Pedido(Cod_Produto, Num_Pedido, Preco_Compra, Quantidade)
VALUES(3, 1, 40, 100);
INSERT INTO Item_Pedido(Cod_Produto, Num_Pedido, Preco_Compra, Quantidade)
VALUES(5, 2, 20, 500);
INSERT INTO Item_Pedido(Cod_Produto, Num_Pedido, Preco_Compra, Quantidade)
VALUES(2, 3, 26, 700);
INSERT INTO Item_Pedido(Cod_Produto, Num_Pedido, Preco_Compra, Quantidade)
VALUES(4, 4, 0.30, 3000);
INSERT INTO Item_Pedido(Cod_Produto, Num_Pedido, Preco_Compra, Quantidade)
VALUES(1, 5, 6.0, 50);
```

---

```
INSERT INTO Dispoe(Cod_Fornecedor, Cod_Produto) VALUES(1, 2);
INSERT INTO Dispoe(Cod_Fornecedor, Cod_Produto) VALUES(2, 1);
INSERT INTO Dispoe(Cod_Fornecedor, Cod_Produto) VALUES(3, 3);
INSERT INTO Dispoe(Cod_Fornecedor, Cod_Produto) VALUES(4, 4);
INSERT INTO Dispoe(Cod_Fornecedor, Cod_Produto) VALUES(5, 5);
```

---

```
INSERT INTO Nota_Fiscal(Num_NF, Cod_Vendedor, Data_venda) VALUES(1, 2,
'02/10/12');
```

```
INSERT INTO Nota_Fiscal(Num_NF, Cod_Vendedor, Data_venda) VALUES(2, 4, '08/10/12');
INSERT INTO Nota_Fiscal(Num_NF, Cod_Vendedor, Data_venda) VALUES(3, 5, '13/10/12');
INSERT INTO Nota_Fiscal(Num_NF, Cod_Vendedor, Data_venda) VALUES(4, 1, '22/10/12');
INSERT INTO Nota_Fiscal(Num_NF, Cod_Vendedor, Data_venda) VALUES(5, 3, '30/10/12');
```

---

```
INSERT INTO Item_NF(Num_NF, Cod_Produto, Quantidade) VALUES(1, 1, 10);
INSERT INTO Item_NF(Num_NF, Cod_Produto, Quantidade) VALUES(2, 4, 1000);
INSERT INTO Item_NF(Num_NF, Cod_Produto, Quantidade) VALUES(3, 2, 200);
INSERT INTO Item_NF(Num_NF, Cod_Produto, Quantidade) VALUES(4, 5, 5);
INSERT INTO Item_NF(Num_NF, Cod_Produto, Quantidade) VALUES(5, 3, 100);
```

---

```
INSERT INTO Creditorio(Cod_Creditorio, Cod_Filial, Cod_Cliente, Num_NF)
VALUES(1, 4, 3, 1);
INSERT INTO Creditorio(Cod_Creditorio, Cod_Filial, Cod_Cliente, Num_NF)
VALUES(2, 4, 2, 2);
INSERT INTO Creditorio(Cod_Creditorio, Cod_Filial, Cod_Cliente, Num_NF)
VALUES(3, 4, 1, 3);
INSERT INTO Creditorio(Cod_Creditorio, Cod_Filial, Cod_Cliente, Num_NF)
VALUES(4, 4, 5, 4);
INSERT INTO Creditorio(Cod_Creditorio, Cod_Filial, Cod_Cliente, Num_NF)
VALUES(5, 4, 4, 5);
```

---

```
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(1, 1, '15/12/12', 6000);
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(2, 1, '15/01/13', 6000);
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(3, 2, '15/12/12', 600);
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(4, 2, '15/01/13', 600);
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(5, 3, '15/12/12', 12000);
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(6, 4, '15/12/12', 25);
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(7, 4, '15/01/13', 25);
INSERT INTO Fatura(Cod_Faturamento, Cod_Creditorio, Data_Vencimento,
Valor_Fatura) VALUES(8, 4, '15/02/13', 25);
```

```
INSERT INTO Fatura(Cod_Faturamento, Cod_Crediarario, Data_Vencimento, Valor_Fatura) VALUES(9, 5, '15/12/12', 8000);
```

## 5 – Consultas

### 5.1.1

```
Select Fornecedor.Nome, Item_Pedido.Preco_Compra,
Pedido_Compra.Data_entrega, Pedido_Compra.Data_Compra From Fornecedor,
Pedido_Compra, Item_Pedido, Produto
Where (Produto.Descricao = 'Cimento') and
(Item_Pedido.Cod_Produto = Produto.Cod_Produto) and
(Pedido_Compra.Num_Pedido = Item_Pedido.Num_Pedido) and
(Fornecedor.Cod_Fornecedor = Pedido_Compra.Cod_Fornecedor) and
(Fornecedor.Cidade = 'Rio de Janeiro');
```

### 5.1.2

```
Select Produto.Descricao From Produto, Dispoe
where (Produto.Cod_Produto = Dispoe.Cod_Produto) and (Dispoe.Cod_Fornecedor =
(Select Fornecedor.Cod_Fornecedor From Fornecedor, Pedido_Compra,
Item_Pedido, Produto
Where (Produto.Descricao = 'Cimento') and
(Item_Pedido.Cod_Produto = Produto.Cod_Produto) and
(Pedido_Compra.Num_Pedido = Item_Pedido.Num_Pedido) and
(Fornecedor.Cod_Fornecedor = Pedido_Compra.Cod_Fornecedor) and
(Fornecedor.Cidade='Rio de Janeiro')));
```

### 5.2.1

```
Select Cliente.Nome from Cliente, Crediarario
where (Cliente.Cod_Cliente = Crediarario.Cod_Cliente) and
(Crediarario.Num_NF in
(Select Item_NF.Num_NF from Item_NF, Produto
group by Item_NF.Num_NF
Order by sum(Produto.Preco_Venda * Item_NF.Quantidade) Desc));
```

Observação: Poderia ser utilizado, após Desc, Limit para selecionar um numero determinado de clientes que mais comprem, mas o enunciado mencionou ‘Todos’.

### 5.2.2

```
Select Vendedor.Nome, Nota_Fiscal.Num_NF,
sum(Produto.Preco_Venda * Item_NF.Quantidade) as Valor,
sum(Produto.Preco_Venda * Item_NF.Quantidade * 0.05) as Comissao
from Item_NF, Produto, Nota_Fiscal, Vendedor
```

```

where (Item_NF.Cod_Produto = Produto.Cod_Produto) and (Nota_Fiscal.Num_NF =
Item_NF.Num_NF) and (Nota_Fiscal.Cod_Vendedor = Vendedor.Cod_Vendedor)
group by Nota_Fiscal.Num_NF, Vendedor.Nome
Order by Vendedor.Nome;

```

### 5.2.3

```

select Cliente.Nome, sum(Valor_Fatura) as Divida, sum(Cliente.Salario * 0.3)
as Credito from Cliente, Crediario, Fatura
Where (Cliente.Cod_Cliente = Crediario.Cod_Cliente) and
(Crediario.Cod_Crediario = Fatura.Cod_Crediario)
Group by Cliente.Nome
Order by Cliente.Nome;

```

### 5.2.4

```

select Cliente.Nome, sum(Valor_Fatura) as Divida, sum(Cliente.Salario * 0.3)
as Credito, sum((Cliente.Salario * 0.3) - Valor_Fatura) as Disponivel_Compra
from Cliente, Crediario, Fatura
Where (Cliente.Cod_Cliente = Crediario.Cod_Cliente) and
(Crediario.Cod_Crediario=Fatura.Cod_Crediario)
Group by Cliente.Nome
Order by Cliente.Nome;

```

### 5.3.1

```

select Descricao from Produto
where Quantidade_Estoque <= Limite_Minimo;

```

### 5.3.2

```

(Select Produto.Descricao, Fornecedor.Nome, Sum(Produto.Preco_venda *
Item_NF.Quantidade) as Acumulado from Produto, Dispoe, Fornecedor, Item_NF
where (Dispoe.Cod_Produto = Produto.Cod_Produto) and
(Fornecedor.Cod_Fornecedor = Dispoe.Cod_Fornecedor) and
(Item_NF.Cod_Produto = Produto.Cod_Produto)
Group by Produto.Descricao, Fornecedor.Nome
Order by Acumulado desc
Limit 1)
UNION ALL
(Select Produto.Descricao, Fornecedor.Nome, Sum(Produto.Preco_venda *
Item_NF.Quantidade) as Acumulado from Produto, Dispoe, Fornecedor, Item_NF
where (Dispoe.Cod_Produto = Produto.Cod_Produto) and
(Fornecedor.Cod_Fornecedor = Dispoe.Cod_Fornecedor) and
(Item_NF.Cod_Produto = Produto.Cod_Produto)
Group by Produto.Descricao, Fornecedor.Nome

```

```
Order by Acumulado Asc
Limit 1);
```

### 5.3.3

```
select Descricao from Produto
where Meses_Garantia = 3;
```

## 6 – Relação dos trigger's

```
CREATE OR REPLACE FUNCTION Capacidade_Estoque()
RETURNS TRIGGER AS
$$
BEGIN
    IF NOT EXISTS (Select Quantidade_Estoque From Produto
    Where (Produto.Cod_Produto = New.Cod_Produto) and
    (Produto.Quantidade_Estoque >= New.Quantidade))
    THEN
        RAISE NOTICE 'Estoque com baixa capacidade para atender esse pedido!';
    ELSE
        RETURN NEW;
    END IF;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER Estoque_Produto
BEFORE INSERT OR UPDATE ON Item_NF
FOR EACH ROW
EXECUTE PROCEDURE Capacidade_Estoque();
```

---

```
CREATE OR REPLACE FUNCTION Atualiza_Estoque()
RETURNS TRIGGER AS
$$
BEGIN
    UPDATE Produto
    SET Quantidade_Estoque = Quantidade_Estoque - NEW.Quantidade
    WHERE Produto.Cod_Produto = NEW.Cod_Produto;

    IF NOT EXISTS (Select Produto.Quantidade_Estoque From Produto
    Where (Produto.Cod_Produto = New.Cod_Produto) and
    (Produto.Quantidade_Estoque >= Produto.Limite_Minimo))
```

```

        THEN
            RAISE NOTICE 'Setor de Compras: Item do Estoque no LIMITE MÍNIMO.';
        END IF;
        RETURN NULL;
    END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER Atualiza_Quatidade_Estoque
AFTER INSERT OR UPDATE ON Item_NF
    FOR EACH ROW
        EXECUTE PROCEDURE Atualiza_Estoque();

```

## 7 – Grant's da parte de segurança

```

REVOKE ALL ON Cliente, Pedido_Compra FROM public;
GRANT ALL ON Cliente IN SCHEMA public TO Gerente, Vendedor;
GRANT ALL ON Pedido_Compra IN SCHEMA public TO Setor_Compras;

```

## 8 – Referências

**PASSOS**, Cláudio. **Notas de aulas de Banco de Dados II da matéria INF434 ministradas.** Laboratório da UGF – Universidade Gama Filho, campus da Piedade, Rio de Janeiro: 2º Período de 2012.