



FUNDAÇÃO UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO  
CAMPUS JUAZEIRO  
CURSO DE ENGENHARIA DA COMPUTAÇÃO

**JONAS CASSIANO DOS SANTOS**  
**MATHEUS DOS ANJOS MARTINS**

## **PROJETO DE BANCO DE DADOS II: FARMÁCIA**

Juazeiro, BA

2019

**JONAS CASSIANO DOS SANTOS**  
**MATHEUS DOS ANJOS MARTINS**

## **PROJETO DE BANCO DE DADOS II: FARMÁCIA**

Relatório requerido pelo professor Mario Godoy Neto, como parte das exigências de avaliação da disciplina Banco de Dados II, do curso de Engenharia da Computação, da Universidade do Vale do São Francisco.

Juazeiro, BA

2019

## Sumário

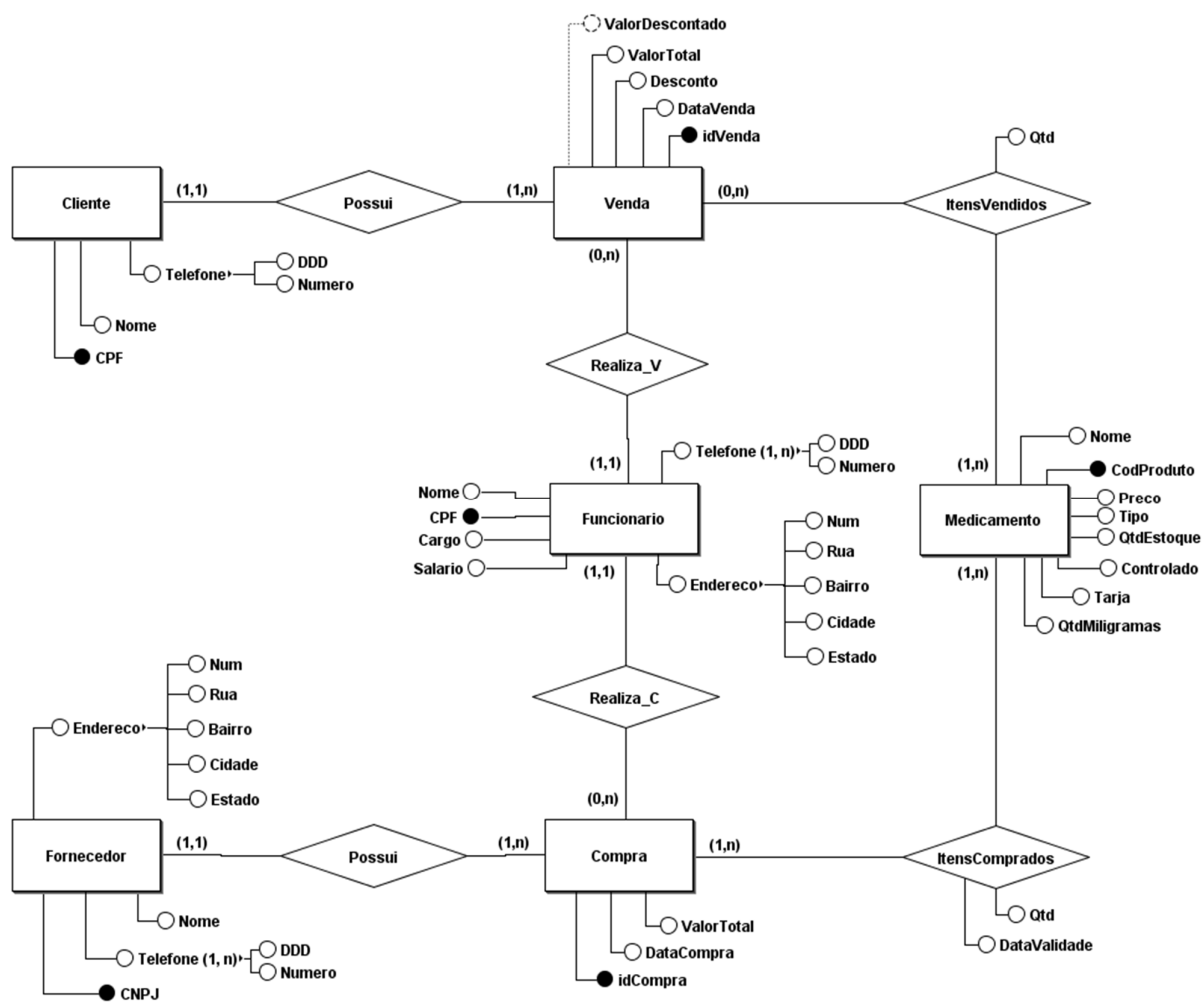
1. MODELO CONCEITUAL .....	5
1.1. Diagrama Entidade Relacionamento (DER).....	6
2. Diagrama Relacional (DR) .....	7
3. MODELO FÍSICO .....	8
3.1. CRIAÇÃO DE TABELAS .....	8
3.2. INSERÇÕES NAS TABELAS .....	12
3.2.1. FUNCIONARIOS .....	12
3.2.2. CLIENTES .....	12
3.2.3. Medicamento .....	13
3.2.4. FORNECEDOR .....	13
3.2.5. VENDA .....	14
3.2.6. COMPRA.....	14
3.2.7. ITENS VENDIDOS .....	14
3.2.8. ITENS COMPRADOS .....	15
3.2.9. Telefone_FUN .....	15
3.2.10. Telefone_FOR .....	16
3.3. Dois exemplos de cada comando .....	16
3.3.1. INSERT .....	16
3.3.2. ALTER TABLE.....	17
3.3.3. Funções de Conjuntos .....	17
3.3.4. SELECT e suas variações .....	18
3.3.4.1. Expressões.....	18
3.3.4.2. Aliases.....	18
3.3.4.3. Concat.....	19
3.3.4.4. Literais.....	19
3.3.4.5. Null .....	20
3.3.4.6. Distinct.....	21
3.3.4.7. Order BY .....	21
3.3.4.8. Where.....	22
3.3.4.9. Where between .....	22
3.3.4.10. Where in .....	23
3.3.4.11. Where like .....	23
3.3.4.12. Where is null.....	24
3.3.4.13. Where NOT .....	24
3.3.4.14. JOIN .....	25

3.3.4.15.	SELECT, Cláusulas X Operadores .....	25
3.3.4.16.	SELECT Dentro de outro SELECT .....	26
3.3.4.17.	UNION.....	26
3.3.4.18.	UNION ALL .....	27
3.3.4.19.	INTERSECT .....	28
3.3.4.20.	MINUS.....	29
3.3.4.21.	Funções de conversão (UPPERCASE, LOWERCASE etc) .....	29
3.3.4.22.	Manipulação de caractere.....	30
3.3.4.23.	Funções numéricas .....	31
3.3.5.	Delete .....	31
3.3.6.	View .....	32
3.3.7.	Update.....	33
3.3.8.	Group By .....	33
3.3.9.	Having .....	34
3.3.10.	Subquery.....	34
3.3.11.	PL/SQL .....	35
3.3.11.1.	SELECT INTO .....	35
3.3.11.2.	IF .....	36
3.3.11.3.	LOOPS .....	37
3.3.12.	STORED PROCEDURES .....	39
3.3.13.	TRIGGER.....	40

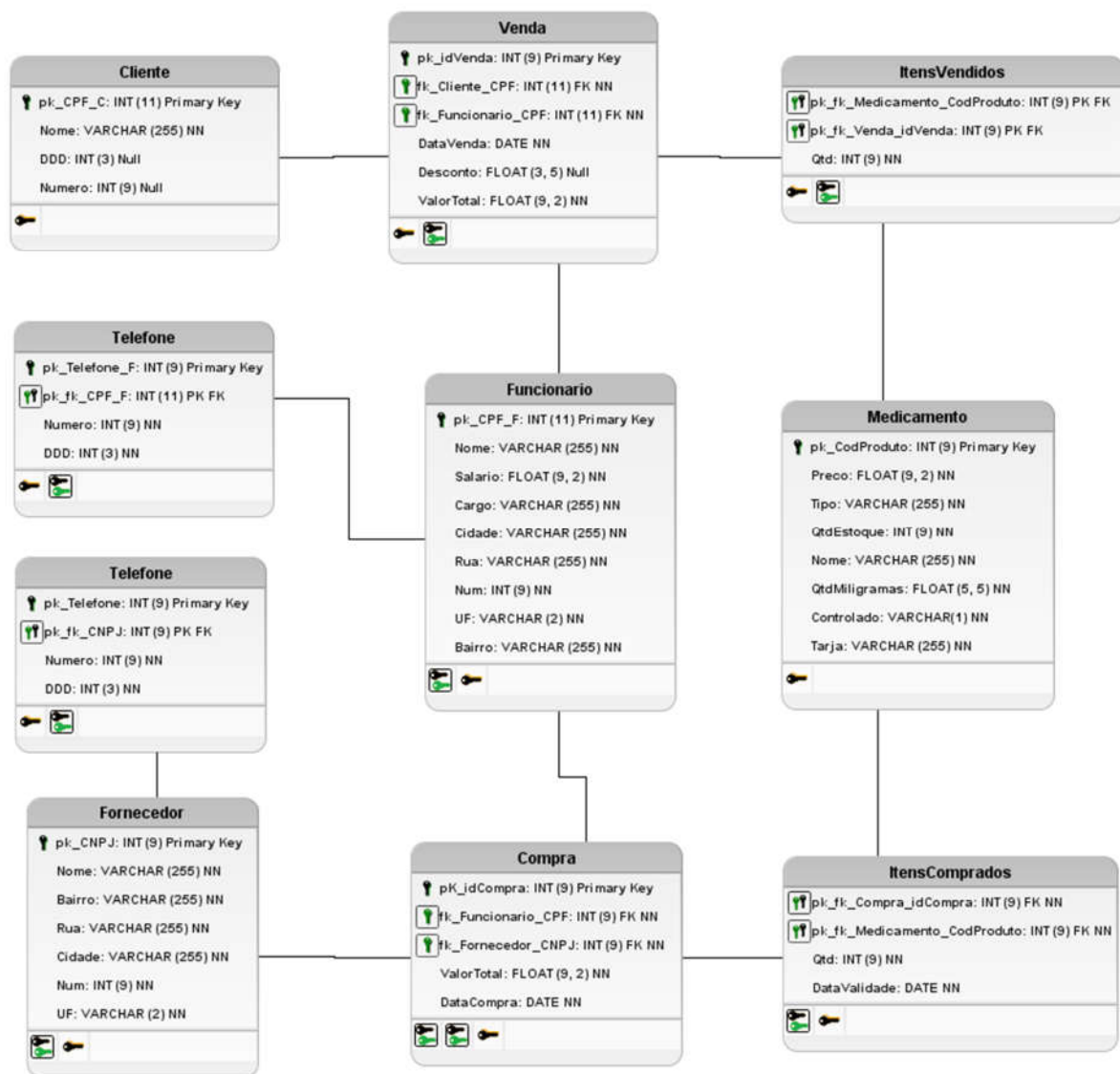
## **1. MODELO CONCEITUAL**

Em uma farmácia os funcionários possuem um cadastro com seu nome, endereço, CPF, cargo, salário e telefone(s). Os funcionários realizam o atendimento em vendas, que são cadastradas com o código da venda, data, além de poderem contar com algum desconto sendo este efetivamente guardado no sistema bem como o valor total da venda. Os clientes possuem um cadastro com seu CPF, nome e telefone. Nas vendas são vendidos apenas medicamentos, cada venda realizada tem em seu cadastro a quantidade de medicamentos vendidos. Estes por sua vez possuem um cadastro contendo nome, código, preço, tipo, quantidade em estoque, se é de uso controlado, qual a tarja e a quantidade em miligramas. Cada medicamento pode ser comprado de diferentes fornecedores sendo que cada compra é realizada por um funcionário. Um fornecedor possui um CNPJ, endereço, nome e telefone(s). Em cada compra é gerado um código, a data em que foi realizada, o valor, a quantidade do medicamento e é atribuída a data de validade.

### 1.1. Diagrama Entidade Relacionamento (DER)



## 2. Diagrama Relacional (DR)









```
CREATE TABLE TELEFONE_FUN (
    pk_Telefone_F NUMBER(9) NOT NULL,
    pk_fk_CPF_F NUMBER(11) NOT NULL,
    Numero NUMBER(9) NOT NULL,
    DDD NUMBER(3) NOT NULL,
    CONSTRAINT pk_Telefone_fun PRIMARY KEY (pk_Telefone_F, pk_fk_CPF_F),
    CONSTRAINT fk_CPF_F FOREIGN KEY (pk_fk_CPF_F) REFERENCES
FUNCIONARIO(PK_CPF_F)
);
```

### Resultado:

Object Type **TABLE** Object **TELEFONE\_FUN**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TELEFONE_FUN	PK_TELEFONE_F	Number	-	9	0	1	-	-	-
	PK_FK_CPF_F	Number	-	11	0	2	-	-	-
	NUMERO	Number	-	9	0	-	-	-	-
	DDD	Number	-	3	0	-	-	-	-
1 - 4									

```
CREATE TABLE Compra (
    pk_idCompra NUMBER(9) PRIMARY KEY,
    fk_Funcionario_CPF NUMBER(11) NOT NULL,
    fk_Fornecedor_CNPJ NUMBER(14) NOT NULL,
    ValorTotal NUMBER(9,2) NOT NULL,
    DataCompra DATE NOT NULL,
    CONSTRAINT fk_Compra_Funcionario_CPF FOREIGN KEY (fk_Funcionario_CPF)
REFERENCES funcionario(pk_CPF_F),
    CONSTRAINT fk_Compra_Fornecedor_CNPJ FOREIGN KEY (fk_Fornecedor_CNPJ)
REFERENCES fornecedor(pk_CNPJ)
);
```

### Resultado:

Object Type **TABLE** Object **COMPRA**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COMPRA	PK_IDCOMPRA	Number	-	9	0	1	-	-	-
	FK_FUNCIONARIO_CPF	Number	-	11	0	-	-	-	-
	FK_FORNECEDOR_CNPJ	Number	-	14	0	-	-	-	-
	VALORTOTAL	Number	-	9	2	-	-	-	-
	DATACOMPRA	Date	7	-	-	-	-	-	-
1 - 5									

```
CREATE TABLE TELEFONE_FOR (
    pk_Telefone_FOR NUMBER(9) NOT NULL,
    pk_fk_CNPJ NUMBER(14) NOT NULL,
    Numero NUMBER(9) NOT NULL,
    DDD NUMBER(3) NOT NULL,
    CONSTRAINT pk_Telefone_fornecedor PRIMARY KEY (pk_Telefone_FOR,
pk_fk_CNPJ),
    CONSTRAINT fk_CPNJ_FOR FOREIGN KEY (pk_fk_CNPJ) REFERENCES
FORNECEDOR(pk_CNPJ)
);
```

**Resultado:**Object Type **TABLE** Object **TELEFONE\_FOR**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TELEFONE_FOR	PK_TELEFONE_FOR	Number	-	9	0	1	-	-	-
	PK_FK_CNPJ	Number	-	14	0	2	-	-	-
	NUMERO	Number	-	9	0	-	-	-	-
	DDD	Number	-	3	0	-	-	-	-
1 - 4									

```

CREATE TABLE Venda (
    pk_idVenda NUMBER(9) PRIMARY KEY,
    fk_Cliente_CPF NUMBER (11) NOT NULL,
    fk_Funcionario_CPF NUMBER (11) NOT NULL,
    DataVenda DATE NOT NULL,
    Desconto NUMBER(5,2),
    ValorTotal NUMBER(9,2) NOT NULL,
    CONSTRAINT fk_Venda_Cliente_CPF FOREIGN KEY (fk_Cliente_CPF) REFERENCES
Cliente(pk_CPF_C),
    CONSTRAINT fk_Venda_Funcionario_CPF FOREIGN KEY (fk_Funcionario_CPF)
REFERENCES funcionario(pk_CPF_F)
);

```

**Resultado:**Object Type **TABLE** Object **VENDA**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VENDA	PK_IDVENDA	Number	-	9	0	1	-	-	-
	FK_CLIENTE_CPF	Number	-	11	0	-	-	-	-
	FK_FUNCIONARIO_CPF	Number	-	11	0	-	-	-	-
	DATAVENDA	Date	7	-	-	-	-	-	-
	DESCONTO	Number	-	5	2	-	✓	-	-
	VALORTOTAL	Number	-	9	2	-	-	-	-
1 - 6									

```

CREATE TABLE ItensVendidos (
    pk_fk_Medicamento_CodProduto NUMBER(9) NOT NULL,
    pk_fk_Venda_idVenda NUMBER(9) NOT NULL,
    Qtd NUMBER(9) NOT NULL,
    CONSTRAINT pk_ItensVendidos PRIMARY KEY (pk_fk_Medicamento_CodProduto,
pk_fk_Venda_idVenda),
    CONSTRAINT pk_fk_Med_Produto FOREIGN KEY (pk_fk_Medicamento_CodProduto)
REFERENCES Medicamento(pk_CodProduto),
    CONSTRAINT pk_fk_Vend_idVenda FOREIGN KEY (pk_fk_Venda_idVenda) REFERENCES
Venda(pk_idVenda)
);

```

**Resultado:**Object Type **TABLE** Object **ITENS VENDIDOS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
ITENS VENDIDOS	PK_FK_MEDICAMENTO_CODPRODUTO	Number	-	9	0	1	-
	PK_FK_VENDA_IDVENDA	Number	-	9	0	2	-
	QTD	Number	-	9	0	-	-

```
CREATE TABLE ItensComprados (
    pk_fk_Compra_idCompra NUMBER(9) NOT NULL,
    pk_fk_Medicamento_CodProduto NUMBER(9) NOT NULL,
    Qtd NUMBER(9) NOT NULL,
    DataValidade DATE NOT NULL,
    CONSTRAINT pk_ItensComprados PRIMARY KEY (pk_fk_Compra_idCompra,
    pk_fk_Medicamento_CodProduto),
    CONSTRAINT pk_fk_Compr_idCompra FOREIGN KEY (pk_fk_Compra_idCompra)
REFERENCES Compra(pk_idCompra),
    CONSTRAINT pk_fk_Med_CodProduto FOREIGN KEY (pk_fk_Medicamento_CodProduto)
REFERENCES Medicamento(pk_CodProduto)
);
```

### Resultado:

Object Type **TABLE** Object **ITENSCOMPRADOS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
ITENSCOMPRADOS	PK_FK_COMPRA_IDCOMPRA	Number	-	9	0	1	-
	PK_FK_MEDICAMENTO_CODPRODUTO	Number	-	9	0	2	-
	QTD	Number	-	9	0	-	-
	DATAVALIDADE	Date	7	-	-	-	-

## 3.2. INSERÇÕES NAS TABELAS

### 3.2.1. FUNCIONARIOS

```
BEGIN
    INSERT INTO funcionario VALUES (10123453401, 'Jaime Rodrigues Araújo',
    1500.00, 'Operador de Caixa', 'Juazeiro', 'Rua 10', '100', 'PE', 'Loteamento
    Recife');
    INSERT INTO funcionario VALUES (11113398043, 'Jorge Lucas Star Wars',
    2500.00, 'Operador de Caixa', 'Petrolina', 'Rua 21', '420', 'PE', 'Vila
    Eduardo');
    INSERT INTO funcionario VALUES (94428187084, 'Maria Carolina Araújo',
    4300.00, 'Gerente', 'Petrolina', 'Rua 31', '200', 'PE', 'Cohab');
END;
```

### Resultado:

PK_CPF_F	NOME	SALARIO	CARGO	CIDADE	RUA	NUM	UF	BAIRRO
10123453401	Jaime Rodrigues Araújo	1500	Operador de Caixa	Juazeiro	Rua 10	100	PE	Loteamento Recife
11113398043	Jorge Lucas Star Wars	2500	Operador de Caixa	Petrolina	Rua 21	420	PE	Vila Eduardo
94428187084	Maria Carolina Araújo	4300	Gerente	Petrolina	Rua 31	200	PE	Cohab

### 3.2.2. CLIENTES

```
BEGIN
    INSERT INTO cliente VALUES(19006207071,'Eronildo Junior',81,
    22010112); INSERT INTO cliente VALUES(30765713039,'Edwildson Filho da
    Natureza',74, 23110212);
    INSERT INTO cliente VALUES(82871596018,'Tiago Tito Liro',74, 24160311);
    INSERT INTO cliente VALUES(19006217171,'Joao Pinto',null,null);
END;
```

Resultado:

PK_CPF_C	NOME	DDD	NUMERO
19006207071	Eronildo Junior	81	22010112
30765713039	Edwildson Filho da Natureza	74	23110212
82871596018	Tiago Tito Liro	74	24160311
19006217171	Joao Pinto	-	-

### 3.2.3. Medicamento

```
BEGIN
  INSERT INTO medicamento VALUES (01,12.00,'XAROPE',10, 'ACEBROFILINA',
25.00,'S','Vermelha');
  INSERT INTO medicamento VALUES (02,140.67,'COMPRIMIDO',4, 'RITALINA',
10.00,'S','Preta');
  INSERT INTO medicamento VALUES (03,5.00,'GEL',13, 'DICLOFENACO SÓDICO',
10.00,'N','N');
  INSERT INTO medicamento VALUES (04,8.00,'COMPRIMIDO',33, 'IBUPROFENO',
200.00,'N','N');
  INSERT INTO medicamento VALUES (05,4.00,'COMPRIMIDO',20, 'NIMESULIDA',
100.00,'N','N');
END;
```

Resultado:

PK_CODPRODUTO	PRECO	TIPO	QTDESTOQUE	NOME	QTMILIGRAMAS	CONTROLADO	TARJA
1	12	XAROPE	10	ACEBROFILINA	25	S	Vermelha
2	140.67	COMPRIMIDO	4	RITALINA	10	S	Preta
3	5	GEL	13	DICLOFENACO SÓDICO	10	N	N
4	8	COMPRIMIDO	33	IBUPROFENO	200	N	N
5	4	COMPRIMIDO	20	NIMESULIDA	100	N	N

### 3.2.4. FORNECEDOR

```
BEGIN
  INSERT INTO fornecedor VALUES (00012301443002,'DB MEDICAMENTOS', 'MARIA
DAS DORES','RUA TREZE', 'PETROLINA' , 440, 'PE');
  INSERT INTO fornecedor VALUES (00012301443232,'BANCO MEDICAMENTOS
CONTROLADOS', 'JOSE DE MATOS' , 'RUA DAS GRAÇAS', 'JUAZEIRO' , 770, 'BA');
  INSERT INTO fornecedor VALUES (00022303113002,'DB SP MED', 'ALTO
NORTE','RUA 7', 'SAO PAULO' , 120, 'SP');
END;
```

Resultado:

PK_CNPJ	NOME	BAIRRO	RUA	CIDADE	NUM	UF
12301443002	DB MEDICAMENTOS	MARIA DAS DORES	RUA TREZE	PETROLINA	440	PE
12301443232	BANCO MEDICAMENTOS CONTROLADOS	JOSE DE MATOS	RUA DAS GRAÇAS	JUAZEIRO	770	BA
22303113002	DB SP MED	ALTO NORTE	RUA 7	SAO PAULO	120	SP

### 3.2.5. VENDA

```
BEGIN
  INSERT INTO Venda VALUES(01,19006207071, 10123453401,
TO_DATE('12/12/2018','MM/DD/YYYY'), 0.2,100.00);
  INSERT INTO Venda VALUES(02,30765713039, 10123453401,
TO_DATE('12/12/2018','MM/DD/YYYY'), null, 10.00);
  INSERT INTO Venda VALUES(03,30765713039, 11113398043,
TO_DATE('03/01/2019','MM/DD/YYYY'), null, 50.00);
END;
```

Resultado:

PK_IDVENDA	FK_CLIENTE_CPF	FK_FUNCIONARIO_CPF	DATAVENDA	DESCONTO	VALORTOTAL
1	19006207071	10123453401	12-DEC-18	.2	100
2	30765713039	10123453401	12-DEC-18	-	10
3	30765713039	11113398043	01-MAR-19	-	50

### 3.2.6. COMPRA

```
BEGIN
  INSERT INTO COMPRA VALUES(01,10123453401, 00012301443002 , 320,
TO_DATE('10/10/2018','MM/DD/YYYY'));
  INSERT INTO COMPRA VALUES(02,94428187084, 00012301443232 , 180,
TO_DATE('09/10/2018','MM/DD/YYYY'));
  INSERT INTO COMPRA VALUES(03,94428187084 ,00022303113002 ,703.35
,TO_DATE('08/10/2018','MM/DD/YYYY'));
  INSERT INTO COMPRA VALUES(04,94428187084 ,00022303113002 ,65,
TO_DATE('08/10/2018','MM/DD/YYYY'));
  INSERT INTO COMPRA VALUES(05,94428187084 ,00022303113002 ,80,
TO_DATE('08/10/2018','MM/DD/YYYY'));
END;
```

Resultado:

PK_IDCOMPRA	FK_FUNCIONARIO_CPF	FK_FORNECEDOR_CNPJ	VALORTOTAL	DATACOMPRA
1	10123453401	12301443002	320	10-OCT-18
2	94428187084	12301443232	180	10-SEP-18
3	94428187084	22303113002	703.35	10-AUG-18
4	94428187084	22303113002	65	10-AUG-18
5	94428187084	22303113002	80	10-AUG-18

### 3.2.7. ITENS VENDIDOS

```
BEGIN
  INSERT INTO ITENSVENDIDOS VALUES(01,01,10);
  INSERT INTO ITENSVENDIDOS VALUES(02,02,15);
  INSERT INTO ITENSVENDIDOS VALUES(03,03,5);
  INSERT INTO ITENSVENDIDOS VALUES(02,03,4);
END;
```

Resultado:

PK_FK_MEDICAMENTO_CODPRODUTO	PK_FK_VENDA_IDVENDA	QTD
1	1	10
2	2	15
3	3	5
2	3	4

### 3.2.8. ITENS COMPRADOS

```
BEGIN
  INSERT INTO ITENSCOMPRADOS VALUES(01,04,40, TO_DATE('10/10/2020'
, 'MM/DD/YYYY'));
  INSERT INTO ITENSCOMPRADOS VALUES(02,01,15, TO_DATE('10/10/2020'
, 'MM/DD/YYYY'));
  INSERT INTO ITENSCOMPRADOS VALUES(03,02,5, TO_DATE('10/10/2021'
, 'MM/DD/YYYY'));
  INSERT INTO ITENSCOMPRADOS VALUES(04,03,13, TO_DATE('10/10/2021'
, 'MM/DD/YYYY'));
  INSERT INTO ITENSCOMPRADOS VALUES(05,05,20, TO_DATE('10/10/2024'
, 'MM/DD/YYYY'));
END;
```

Resultado:

PK_FK_COMPRA_IDCOMPRA	PK_FK_MEDICAMENTO_CODPRODUTO	QTD	DATAVALIDADE
1	4	40	10-OCT-20
2	1	15	10-OCT-20
3	2	5	10-OCT-21
4	3	13	10-OCT-21
5	5	20	10-OCT-24

### 3.2.9. Telefone\_FUN

```
BEGIN
  INSERT INTO TELEFONE_FUN VALUES(01,10123453401,988786755, 87);
  INSERT INTO TELEFONE_FUN VALUES(02,10123453401,988665555, 87);
  INSERT INTO TELEFONE_FUN VALUES(01,11113398043,123443215, 74);
  INSERT INTO TELEFONE_FUN VALUES(01,94428187084,123412345, 74);
END;
```

Resultado:

PK_TELEFONE_F	PK_FK_CPF_F	NUMERO	DDD
1	10123453401	988786755	87
2	10123453401	988665555	87
1	11113398043	123443215	74
1	94428187084	123412345	74

### 3.2.10. Telefone\_FOR

BEGIN

```
INSERT INTO TELEFONE_FOR VALUES(01,00012301443002,988786733, 87);
INSERT INTO TELEFONE_FOR VALUES(02,00012301443002,988665533, 87);
INSERT INTO TELEFONE_FOR VALUES(01,00012301443232,123443233, 74);
INSERT INTO TELEFONE_FOR VALUES(01,00022303113002,123412333, 74);
```

END;

Resultado:

PK_TELEFONE_FOR	PK_FK_CNPJ	NUMERO	DDD
1	12301443002	988786733	87
2	12301443002	988665533	87
1	12301443232	123443233	74
1	22303113002	123412333	74

### 3.3. Dois exemplos de cada comando

#### 3.3.1. INSERT

Cadastra um novo funcionário com os seguintes dados:

CPF: 11123453411

Nome: Ruan dos Santos

Cargo: Operador de Caixa

Salário: 1500

Endereço: Rua 20, nº 200, Loteamento Alto 2, Petrolina, PE

```
INSERT INTO funcionario VALUES (11123453411, 'Ruan dos Santos', 1500.00,
'Operador de Caixa', 'Petrolina', 'Rua 20', '200', 'PE', 'Loteamento Alto 2');
```

Resultado:

PK_CPF_F	NOME	SALARIO	CARGO	CIDADE	RUA	NUM	UF	BAIRRO
10123453401	Jaime Rodrigues Araújo	1500	Operador de Caixa	Juazeiro	Rua 10	100	PE	Loteamento Recife
11113398043	Jorge Lucas Star Wars	2500	Operador de Caixa	Petrolina	Rua 21	420	PE	Vila Eduardo
94428187084	Maria Carolina Araújo	4300	Gerente	Petrolina	Rua 31	200	PE	Cohab
11123453411	Ruan dos Santos	1500	Operador de Caixa	Petrolina	Rua 20	200	PE	Loteamento Alto 2

Cadastra um novo cliente com os seguintes dados:

CPF: 32765713139

Nome: Edmarcio da Silva

Telefone: 74 13111212

```
INSERT INTO cliente VALUES(32765713139,'Edmarcio da Silva',74, 13111212);
```



*Resultado:*

PK_CPF_C	NOME	DDD	NUMERO
32765713139	Edmario da Silva	74	13111212
19006207071	Eronildo Junior	81	22010112
30765713039	Edwildson Filho da Natureza	74	23110212
82871596018	Tiago Tito Liro	74	24160311
19006217171	Joao Pinto	-	-

### 3.3.2. ALTER TABLE

Muda o tipo de dado do desconto para 3 dígitos, sendo 2 deles decimais.

```
ALTER TABLE venda MODIFY desconto number(3,2);
```

Adiciona a coluna, data de validade para a tabela ItensVendidos.

```
ALTER TABLE itensVendidos ADD DataValidade date;
```

### 3.3.3. Funções de Conjuntos

Informa quantos medicamentos não possuem tarja.

```
SELECT COUNT (*) AS QUANTIDADE FROM medicamento WHERE tarja= 'N'
```

*Resultado:*

QUANTIDADE
3

Informa o menor salário.

```
SELECT MIN (salario) AS "Menor Salario" FROM funcionario
```

*Resultado:*

Menor Salario
1500

### 3.3.4. SELECT e suas variações

#### 3.3.4.1. Expressões

Seleciona a coluna nome da tabela funcionario e calcula o valor do salario acumulado em 12 meses de cada funcionario:

```
SELECT nome, salario * 12 FROM funcionario
```

*Resultado:*

NOME	SALARIO*12
Jaime Rodrigues Araújo	18000
Jorge Lucas Star Wars	30000
Maria Carolina Araújo	51600
Ruan dos Santos	18000

Seleciona a coluna nome da tabela medicamento e calcula a quantidade em miligramas de cada medicamento:

```
SELECT nome,qtdestoque*qtdmiligramas AS TotalMiligramas FROM medicamento
```

*Resultado:*

NOME	TOTALMILIGRAMAS
ACEBROFILINA	250
RITALINA	40
DICLOFENACO SÓDICO	130
IBUPROFENO	6600
NIMESULIDA	2000

#### 3.3.4.2. Aliases

Dá um apelido a coluna do salario acumulado:

```
SELECT nome, salario * 12 "Salário acumulado" FROM funcionario
```

*Resultado:*

NOME	Salário Acumulado
Jaime Rodrigues Araújo	18000
Jorge Lucas Star Wars	30000
Maria Carolina Araújo	51600
Ruan dos Santos	18000

Dá um apelido a coluna da quantidade em miligramas em estoque:

```
SELECT nome,qtdestoque*qtdmiligramas "Miligramas em estoque" FROM medicamento
```

*Resultado:*

NOME	Miligramas Em Estoque
ACEBROFILINA	250
RITALINA	40
DICLOFENACO SÓDICO	130
IBUPROFENO	6600
NIMESULIDA	2000

### 3.3.4.3. Concat

Concatena a coluna nome com a coluna tipo da tabela medicamento:

```
SELECT nome || ' ' || tipo AS Concat FROM medicamento
```

*Resultado:*

CONCAT
ACEBROFILINA XAROPE
RITALINA COMPRIMIDO
DICLOFENACO SÓDICO GEL
IBUPROFENO COMPRIMIDO
NIMESULIDA COMPRIMIDO

Concatena a coluna nome com a coluna cargo da tabela funcionario:

```
SELECT nome || ' ' || cargo AS Concat FROM funcionario
```

*Resultado:*

CONCAT
Jaime Rodrigues Araújo Operador de Caixa
Jorge Lucas Star Wars Operador de Caixa
Maria Carolina Araújo Gerente
Ruan dos Santos Operador de Caixa

### 3.3.4.4. Literais

Informa a quantidade em estoque de cada medicamento:

```
SELECT 'O medicamento ' || nome || ' tem ' || qtdestoque || ' unidade(s) em
estoque' AS Query FROM medicamento
```

*Resultado:*

QUERY
O medicamento ACEBROFILINA tem 10 unidade(s) em estoque
O medicamento RITALINA tem 4 unidade(s) em estoque
O medicamento DICLOFENACO SÓDICO tem 13 unidade(s) em estoque
O medicamento IBUPROFENO tem 33 unidade(s) em estoque
O medicamento NIMESULIDA tem 20 unidade(s) em estoque

Informa o CPF do funcionário responsável por uma determinada venda:

```
SELECT 'A venda ' || pk_idVenda || ' foi realizada pelo funcionario de CPF '
|| fk_Funcionario_CPF AS Query FROM venda
```

*Resultado:*

QUERY
A venda 1 foi realizada pelo funcionario de CPF 10123453401
A venda 2 foi realizada pelo funcionario de CPF 10123453401
A venda 3 foi realizada pelo funcionario de CPF 11113398043

### 3.3.4.5. Null

Substitui o valor null por 0.0 na coluna desconto da tabela venda e seleciona o id e o valor de cada venda:

```
SELECT pk_idVenda, ValorTotal, NVL (desconto, 0.00) FROM venda
```

*Resultado:*

PK_IDVENDA	VALORTOTAL	NVL(DESCONTO,0.00)
1	100	.2
2	10	0
3	50	0

Seleciona os telefones dos clientes e substitui o valor null por 0 na coluna número e ddd da tabela cliente:

```
SELECT NVL (ddd, 00),NVL (numero, 0) FROM cliente
```

*Resultado:*

NVL(DDD,00)	NVL(NUMERO,0)
74	13111212
81	22010112
74	23110212
74	24160311
0	0

### 3.3.4.6. Distinct

Seleciona os cpfs de funcionarios que realizaram alguma venda:

```
SELECT DISTINCT fk_Funcionario_CPF FROM venda
```

*Resultado:*

FK_FUNCIONARIO_CPF
10123453401
11113398043

Seleciona os cpfs de funcionarios que realizaram alguma compra:

```
SELECT DISTINCT fk_Funcionario_CPF FROM compra
```

*Resultado:*

FK_FUNCIONARIO_CPF
94428187084
10123453401

### 3.3.4.7. Order BY

Seleciona o id e a data de cada venda e ordena de acordo com a data:

```
SELECT pk_idVenda, DataVenda FROM venda ORDER BY DataVenda
```

*Resultado:*

PK_IDVENDA	DATAVENDA
1	12-DEC-18
2	12-DEC-18
3	01-MAR-19

Seleciona o id e a data de cada compra e ordena de acordo com a data:

```
SELECT pk_idCompra, DataCompra FROM compra ORDER BY DataCompra
```

*Resultado:*

PK_IDCOMPRA	DATACOMPRA
3	10-AUG-18
4	10-AUG-18
5	10-AUG-18
2	10-SEP-18
1	10-OCT-18

### 3.3.4.8. Where

Seleciona as vendas que foram realizadas no dia 12/12/2018:

```
SELECT pk_idVenda FROM venda WHERE DataVenda = TO_DATE ('12/12/2018',  
'mm/dd/yyyy');
```

*Resultado:*

PK_IDVENDA
1
2

Seleciona as compras que foram realizadas pelo funcionario de CPF 10123453401:

```
SELECT pk_idCompra FROM compra WHERE fk_Funcionario_CPF=10123453401
```

*Resultado:*

PK_IDCOMPRA
1

### 3.3.4.9. Where between

Seleciona o nome dos funcionários com salário entre 2000 e 3000:

```
SELECT nome FROM funcionario WHERE salario BETWEEN 2000 AND 3000
```

*Resultado:*

NOME
Jorge Lucas Star Wars

Seleciona o id das vendas entre 100 e 200:

```
SELECT pk_idVenda FROM venda WHERE ValorTotal BETWEEN 100 AND 200
```

*Resultado:*

PK_IDVENDA
1

### 3.3.4.10. Where in

Seleciona os funcionários que ganham 1500 ou 4300:

```
SELECT * FROM funcionario WHERE salario in(1500,4300)
```

*Resultado:*

PK_CPF_F	NOME	SALARIO	CARGO	CIDADE	RUA	NUM	UF	BAIRRO
10123453401	Jaime Rodrigues Araújo	1500	Operador de Caixa	Juazeiro	Rua 10	100	PE	Loteamento Recife
94428187084	Maria Carolina Araújo	4300	Gerente	Petrolina	Rua 31	200	PE	Cohab
11123453411	Ruan dos Santos	1500	Operador de Caixa	Petrolina	Rua 20	200	PE	Loteamento Alto 2

Seleciona os fornecedores com CNPJ 00022303113002 ou 00012301443232:

```
SELECT * FROM fornecedor WHERE pk_CNPJ in(00022303113002,00012301443232)
```

*Resultado:*

PK_CNPJ	NOME	BAIRRO	RUA	CIDADE	NUM	UF
12301443232	BANCO MEDICAMENTOS CONTROLADOS	JOSE DE MATOS	RUA DAS GRAÇAS	JUAZEIRO	770	BA
22303113002	DB SP MED	ALTO NORTE	RUA 7	SAO PAULO	120	SP

### 3.3.4.11. Where like

Seleciona os clientes com nome que começam com E:

```
SELECT * FROM cliente WHERE nome like 'E%'
```

*Resultado:*

PK_CPF_C	NOME	DDD	NUMERO
32765713139	Edmario da Silva	74	13111212
19006207071	Eronildo Junior	81	22010112
30765713039	Edwildson Filho da Natureza	74	23110212

Seleciona os funcionários com nome que começam com J:

```
SELECT * FROM funcionario WHERE nome like 'J%'
```

*Resultado:*

PK_CPF_F	NOME	SALARIO	CARGO	CIDADE	RUA	NUM	UF	BAIRRO
10123453401	Jaime Rodrigues Araújo	1500	Operador de Caixa	Juazeiro	Rua 10	100	PE	Loteamento Recife
11113398043	Jorge Lucas Star Wars	2500	Operador de Caixa	Petrolina	Rua 21	420	PE	Vila Eduardo

### 3.3.4.12. Where is null

Seleciona as vendas sem desconto(nulo):

```
SELECT * FROM venda WHERE desconto IS NULL
```

*Resultado:*

PK_IDVENDA	FK_CLIENTE_CPF	FK_FUNCIONARIO_CPF	DATAVENDA	DESCONTO	VALORTOTAL
2	30765713039	10123453401	12-DEC-18	-	10
3	30765713039	11113398043	01-MAR-19	-	50

Seleciona os clientes sem telefone:

```
SELECT * FROM cliente WHERE numero IS NULL
```

*Resultado:*

PK_CPF_C	NOME	DDD	NUMERO
19006217171	Joao Pinto	-	-

### 3.3.4.13. Where NOT

Seleciona os funcionários cujo salário não está entre 2000 e 3000:

```
SELECT * FROM funcionario WHERE salario not BETWEEN 2000 AND 3000
```

*Resultado:*

PK_CPF_F	NOME	SALARIO	CARGO	CIDADE	RUA	NUM	UF	BAIRRO
10123453401	Jaime Rodrigues Araújo	1500	Operador de Caixa	Juazeiro	Rua 10	100	PE	Loteamento Recife
94428187084	Maria Carolina Araújo	4300	Gerente	Petrolina	Rua 31	200	PE	Cohab
11123453411	Ruan dos Santos	1500	Operador de Caixa	Petrolina	Rua 20	200	PE	Loteamento Alto 2

Seleciona os funcionários com nome que não começam com J:

```
SELECT * FROM funcionario WHERE nome not like 'J%'
```



*Resultado:*

PK_CPF_F	NOME	SALARIO	CARGO	CIDADE	RUA	NUM	UF	BAIRRO
94428187084	Maria Carolina Araújo	4300	Gerente	Petrolina	Rua 31	200	PE	Cohab
11123453411	Ruan dos Santos	1500	Operador de Caixa	Petrolina	Rua 20	200	PE	Loteamento Alto 2

### 3.3.4.14. JOIN

Seleciona o nome do empregado e o id da(s) venda(s) que ele realizou:

```
SELECT funcionario.nome, venda.pk_idVenda FROM funcionario,venda WHERE
funcionario.pk_CPF_F = venda.fk_Funcionario_CPF
```

*Resultado:*

NOME	PK_IDVENDA
Jaime Rodrigues Araújo	1
Jaime Rodrigues Araújo	2
Jorge Lucas Star Wars	3

Seleciona o nome do empregado e o id da(s) compra(s) que ele realizou:

```
SELECT funcionario.nome, compra.pk_idCompra FROM funcionario,compra WHERE
funcionario.pk_CPF_F = compra.fk_Funcionario_CPF
```

*Resultado:*

NOME	PK_IDCOMPRA
Jaime Rodrigues Araújo	1
Maria Carolina Araújo	2
Maria Carolina Araújo	3
Maria Carolina Araújo	4
Maria Carolina Araújo	5

### 3.3.4.15. SELECT, Cláusulas X Operadores

Seleciona nome dos funcionários com venda de 100 ou 50:

```
SELECT funcionario.nome FROM funcionario, venda WHERE venda.valortotal IN
(100.00,50.00) AND funcionario.pk_CPF_F = venda.fk_Funcionario_CPF
```

*Resultado:*

NOME
Jaime Rodrigues Araújo
Jorge Lucas Star Wars

Seleciona o nome do cliente que comprou mais de 10 medicamentos em uma venda e o nome do medicamento vendido:

```
SELECT cliente.nome, medicamento.nome FROM cliente, medicamento, itensvendidos,
venda WHERE itensvendidos.qtd>10 AND
itensvendidos.pk_fk_Venda_idVenda=venda.pk_idVenda AND
itensvendidos.pk_fk_Medicamento_CodProduto=medicamento.pk_CodProduto AND
cliente.pk_CPF_C=venda.fk_Cliente_CPF
```

*Resultado:*

NOME	NOME
Edwildson Filho da Natureza	RITALINA

### 3.3.4.16. SELECT Dentro de outro SELECT

Seleciona os nomes dos funcionários que fizeram alguma venda no dia 12/12/2018:

```
SELECT funcionario.nome FROM funcionario WHERE funcionario.pk_CPF_F IN (
SELECT fk_Funcionario_CPF FROM venda WHERE DataVenda = TO_DATE ('12/12/2018',
'mm/dd/yyyy'))
```

*Resultado:*

NOME
Jaime Rodrigues Araújo

Seleciona os nomes dos clientes que fizeram alguma compra no dia 12/12/2018:

```
SELECT cliente.nome FROM cliente WHERE cliente.pk_CPF_C IN ( SELECT
fk_Cliente_CPF FROM venda WHERE DataVenda = TO_DATE ('12/12/2018',
'mm/dd/yyyy'))
```

*Resultado:*

NOME
Eronildo Junior
Edwildson Filho da Natureza

### 3.3.4.17. UNION

Informa todos os telefones do banco de dados, assim como o nome do proprietário.

```
SELECT nome, ddd, numero FROM cliente UNION
```

```
SELECT funcionario.nome, telefone_fun.ddd, telefone_fun.numero FROM
funcionario, telefone_fun WHERE funcionario.pk_cpf_f =
telefone_fun.pk_fk_cpf_f UNION SELECT fornecedor.nome, telefone_for.ddd,
telefone_for.numero FROM fornecedor, telefone_for WHERE fornecedor.pk_cnpj =
telefone_for.pk_fk_cnpj
```

*Resultado:*

NOME	DDD	NUMERO
BANCO MEDICAMENTOS CONTROLADOS	74	123443233
DB MEDICAMENTOS	87	988665533
DB MEDICAMENTOS	87	988786733
DB SP MED	74	123412333
Edmario da Silva	74	13111212
Edwildson Filho da Natureza	74	23110212
Eronildo Junior	81	22010112
Jaime Rodrigues Araújo	87	988665555
Jaime Rodrigues Araújo	87	988786755
Joao Pinto	-	-
More than 10 rows available. Increase rows selector to view more rows.		

Informa o nome e o telefone de funcionários e fornecedores da cidade de Petrolina

```
SELECT fornecedor.nome, telefone_for.ddd, telefone_for.numero FROM fornecedor,
telefone_for WHERE fornecedor.pk_cnpj = telefone_for.pk_fk_cnpj AND
UPPER(fornecedor.cidade) = 'PETROLINA' UNION SELECT funcionario.nome,
telefone_fun.ddd, telefone_fun.numero FROM funcionario, telefone_fun WHERE
funcionario.pk_cpf_f = telefone_fun.pk_fk_cpf_f AND UPPER(funcionario.cidade)
= 'PETROLINA'
```

*Resultado:*

NOME	DDD	NUMERO
DB MEDICAMENTOS	87	988665533
DB MEDICAMENTOS	87	988786733
Jorge Lucas Star Wars	74	123443215
Maria Carolina Araújo	74	123412345

### 3.3.4.18. UNION ALL

Informa o nome de todos os funcionarios que realizaram compras e vendas de medicamentos, com repetição.

```
SELECT funcionario.nome FROM funcionario, compra WHERE funcionario.pk_cpf_f =
compra.fk_funcionario_cpf UNION ALL SELECT funcionario.nome FROM funcionario,
venda WHERE funcionario.pk_cpf_f = venda.fk_funcionario_cpf
```

*Resultado:*

NOME
Jaime Rodrigues Araújo
Maria Carolina Araújo
Maria Carolina Araújo
Maria Carolina Araújo
Maria Carolina Araújo
Jaime Rodrigues Araújo
Jaime Rodrigues Araújo
Jorge Lucas Star Wars

Informa estado onde se encontram funcionários e os fornecedores. Com repetição.

```
SELECT UF FROM fornecedor UNION ALL SELECT UF FROM funcionario
```

*Resultado:*

UF
PE
BA
SP
PE
PE
PE
PE

### 3.3.4.19. INTERSECT

Informa o nome dos medicamentos sem tarja com menos de 20 unidades em estoque.

```
SELECT nome FROM medicamento WHERE UPPER(tarja) = 'N' intersect select nome
FROM medicamento WHERE qtdEstoque < 20
```

*Resultado:*

NOME
DICLOFENACO SÓDICO

Informa o nome dos funcionários que moram em petrolina e realizaram alguma venda em 2018

```
SELECT funcionario.nome FROM funcionario, venda WHERE funcionario.pk_cpf_f =
venda.fk_funcionario_cpf AND venda.dataVenda BETWEEN TO_DATE ('01/01/2018',
'mm/dd/yyyy') AND TO_DATE ('12/12/2018', 'mm/dd/yyyy') intersect SELECT nome
FROM funcionario WHERE UPPER(cidade) = 'PETROLINA'
```

*Resultado:*

no data found

### 3.3.4.20. MINUS

Informe o nome dos fornecedores que não se situam em Pernambuco

```
SELECT nome FROM fornecedor MINUS SELECT nome FROM fornecedor WHERE UF = 'PE'
```

*Resultado:*

NOME
BANCO MEDICAMENTOS CONTROLADOS
DB SP MED

Informe o nome dos medicamentos que não possuem tarja preta.

```
SELECT nome FROM medicamento minus SELECT nome FROM medicamento WHERE
UPPER(tarja) = 'PRETA'
```

*Resultado:*

NOME
ACEBROFILINA
DICLOFENACO SÓDICO
IBUPROFENO
NIMESULIDA

### 3.3.4.21. Funções de conversão (UPPERCASE, LOWERCASE etc)

Informa o nome dos funcionários com letras MAIÚSCULAS;

```
SELECT UPPER(nome) FROM funcionario
```

Resultado:

UPPER(NOME)
JAIME RODRIGUES ARAÚJO
JORGE LUCAS STAR WARS
MARIA CAROLINA ARAÚJO
RUAN DOS SANTOS

informa o nome dos fornecedores com letras MINÚSCULAS;

```
SELECT LOWER(nome) FROM fornecedor
```

Resultado:

LOWER(NOME)
db medicamentos
banco medicamentos controlados
db sp med

### 3.3.4.22. Manipulação de caractere

Informa o ddd e número dos funcionários em um só campo de nome Telefone.

```
SELECT CONCAT(ddd,numero) AS Telefone FROM telefone_fun
```

Resultado:

TELEFONE
87988786755
87988665555
74123443215
74123412345

Informa o endereço dos fornecedores em um só campo de nome Endereço.

```
SELECT CONCAT(Rua, ' ', ' ') || CONCAT('Nº', NUM) || CONCAT(' ', BAIRRO, ' ', BAIRRO) || CONCAT(' ', ' ', CIDADE) || CONCAT(' - ', UF) AS Endereco FROM fornecedor
```

*Resultado:*

ENDereco
RUA TREZE , N°440, BAIRRO MARIA DAS DORES, PETROLINA - PE
RUA DAS GRAÇAS , N°770, BAIRRO JOSE DE MATOS, JUAZEIRO - BA
RUA 7 , N°120, BAIRRO ALTO NORTE, SAO PAULO - SP

### 3.3.4.23. Funções numéricas

Arredonda o preço dos medicamentos

```
SELECT ROUND(preco, 0) FROM medicamento;
```

*Resultado:*

ROUND(PRECO,0)
12
141
5
8
4

Trunca o preço dos medicamentos

```
SELECT TRUNC(preco) FROM medicamento;
```

*Resultado:*

TRUNC(PRECO)
12
140
5
8
4

### 3.3.5. Delete

Remove todos os medicamentos da base de dados

```
Delete FROM Medicamento
```

Remove os fornecedores que não se situam em Pernambuco

```
Delete FROM fornecedor WHERE UPPER(UF) != 'PE'
```

### 3.3.6. View

Cria uma view com uma coluna que mostra o valor descontado de cada Venda.

```
CREATE view venda_valorDescontado as
  SELECT fk_Cliente_CPF, fk_Funcionario_CPF, DataVenda, Desconto, ValorTotal,
  CASE WHEN Desconto IS NULL OR Desconto = 0 THEN ValorTotal ELSE ValorTotal *
  (1-Desconto) END AS ValorDescontado
  FROM venda;
```

*Resultado:*

FK_CLIENTE_CPF	FK_FUNCIONARIO_CPF	DATAVENDA	DESCONTO	VALORTOTAL	VALORDESCONTADO
19006207071	10123453401	12-DEC-18	.2	100	80
30765713039	10123453401	12-DEC-18	-	10	10
30765713039	11113398043	01-MAR-19	-	50	50

Cria uma view que mostra o nome de cada medicamento comprado e vendido e uma coluna informando a operação.

```
CREATE view operacao as
SELECT DISTINCT medicamento.nome, CASE WHEN medicamento.pk_codproduto =
itensvendidos.pk_fk_Medicamento_CodProduto THEN 'Venda' ELSE
CASE WHEN medicamento.pk_codproduto =
itenscomprados.pk_fk_Medicamento_CodProduto THEN 'Compra' END END AS Operacao
  FROM venda, compra, medicamento, itensvendidos, itensComprados WHERE
(medicamento.pk_codproduto = itensvendidos.pk_fk_Medicamento_CodProduto AND
venda.pk_idvenda = itensVendidos.pk_fk_venda_idVenda) OR
(medicamento.pk_codproduto = itenscomprados.pk_fk_Medicamento_CodProduto AND
compra.pk_idcompra = itenscomprados.pk_fk_compra_idcompra);
```

*Resultado:*

NOME	OPERACAO
ACEBROFILINA	Venda
ACEBROFILINA	Compra
DICLOFENACO SÓDICO	Compra
RITALINA	Venda
NIMESULIDA	Compra
DICLOFENACO SÓDICO	Venda
RITALINA	Compra
IBUPROFENO	Compra



### 3.3.7. Update

Aumenta o salário de 'Operador de Caixa' em 5%

```
UPDATE funcionario SET salario = salario * 1.05 WHERE UPPER(cargo) = 'OPERADOR DE CAIXA'
```

Resultado:

PK_CPF_F	NOME	SALARIO	CARGO	CIDADE	RUA	NUM	UF	BAIRRO
10123453401	Jaime Rodrigues Araújo	1575	Operador de Caixa	Juazeiro	Rua 10	100	PE	Loteamento Recife
11113398043	Jorge Lucas Star Wars	2625	Operador de Caixa	Petrolina	Rua 21	420	PE	Vila Eduardo
94428187084	Maria Carolina Araújo	4300	Gerente	Petrolina	Rua 31	200	PE	Cohab
11123453411	Ruan dos Santos	1575	Operador de Caixa	Petrolina	Rua 20	200	PE	Loteamento Alto 2

Diminui o preço dos medicamentos sem tarja em 10%

```
UPDATE medicamento SET preco = preco * 0.9 WHERE UPPER(tarja) = 'N';
```

Resultado:

PK_CODPRODUTO	PRECO	TIPO	QTDESTOQUE	NOME	QTMILIGRAMAS	CONTROLADO	TARJA
1	12	XAROPE	10	ACEBROFILINA	25	S	Vermelha
2	140.67	COMPRIMIDO	4	RITALINA	10	S	Preta
3	4.5	GEL	13	DICLOFENACO SÓDICO	10	N	N
4	7.2	COMPRIMIDO	33	IBUPROFENO	200	N	N
5	3.6	COMPRIMIDO	20	NIMESULIDA	100	N	N

### 3.3.8. Group By

Informa a quantidade de medicamentos de cada tarja

```
SELECT tarja, count(*) AS Quantidade FROM medicamento group by tarja
```

Resultado:

TARJA	QUANTIDADE
Preta	1
N	3
Vermelha	1

Informa o menor, o maior e a média dos salários separados por Estado

```
SELECT UF, min(salario) AS "Menor Salario", avg(salario) AS "Média", max(salario) AS "Maior Salário" FROM funcionario group by UF
```

Resultado:

UF	Menor Salario	Média	Maior Salário
PE	1575	2518.75	4300

### 3.3.9. Having

Informa o nome dos funcionários que realizaram mais de uma venda, seguido da respectiva quantidade.

```
SELECT funcionario.nome, count(*) AS Quantidade FROM funcionario, venda WHERE
funcionario.pk_cpf_f = venda.fk_funcionario_cpf group by funcionario.nome
having count(*) > 1
```

Resultado:

NOME	QUANTIDADE
Jaime Rodrigues Araújo	2

Informa as tarjas na qual a soma da quantidade total de medicamentos em estoque é inferior a 20.

```
SELECT tarja, sum(qtdEstoque) AS Quantidade FROM medicamento group by tarja
having sum(qtdEstoque) < 20
```

Resultado:

TARJA	QUANTIDADE
Preta	4
Vermelha	10

### 3.3.10. Subquery

Informa os medicamentos cujo preço é maior que o preço de todos medicamento sem tarja

```
SELECT * FROM medicamento WHERE Preco > ALL ( SELECT Preco FROM medicamento
WHERE tarja='N');
```

Resultado:

PK_CODPRODUTO	PRECO	TIPO	QTDESTOQUE	NOME	QTMILIGRAMAS	CONTROLADO	TARJA
1	12	XAROPE	10	ACEBROFILINA	25	S	Vermelha
2	140.67	COMPRIMIDO	4	RITALINA	10	S	Preta

Informa o nome dos funcionários que já fizeram compra e venda de medicamentos

```
SELECT nome FROM funcionario WHERE pk_cpf_f = ANY(SELECT fk_funcionario_cpf
FROM venda) AND pk_cpf_f = ANY(SELECT fk_funcionario_cpf FROM compra)
```

*Resultado:*

NOME
Jaime Rodrigues Araújo

### 3.3.11. PL/SQL

#### 3.3.11.1. SELECT INTO

Informa qual foi o funcionário que fez a última venda, seguida de sua respectiva data.

```
DECLARE
  VarNOME FUNCIONARIO.NOME%TYPE;
  VarData VENDA.DATAVENDA%TYPE;
BEGIN
  SELECT *
  INTO VarNOME, VarData
  FROM (SELECT FUNCIONARIO.NOME, VENDA.DATAVENDA
        FROM VENDA, FUNCIONARIO
        WHERE VENDA.FK_FUNCIONARIO_CPF = FUNCIONARIO.PK_CPF_F
        ORDER BY DATAVENDA DESC)
  WHERE ROWNUM = 1;
  DBMS_OUTPUT.PUT_LINE(VarData || ' - ' || VarNome);
END;
```

*Resultado:*

01-MAR-19 - Jorge Lucas Star Wars

Statement processed.

0.00 seconds

Informa o DDD e o Número do primeiro fornecedor da farmácia.

```
DECLARE
  VarDDD TELEFONE_FOR.DDD%TYPE;
  VarNUM TELEFONE_FOR.NUMERO%TYPE;
BEGIN
  SELECT *
  INTO VarDDD, VarNUM
  FROM (SELECT TELEFONE_FOR.DDD, TELEFONE_FOR.NUMERO
```

```

        FROM COMPRA, TELEFONE_FOR, FORNECEDOR
        WHERE COMPRA.FK_FORNECEDOR_CNPJ = FORNECEDOR.PK_CNPJ AND
FORNECEDOR.PK_CNPJ = TELEFONE_FOR.PK_FK_CNPJ ORDER BY DATACOMPRA )
    WHERE ROWNUM = 1;
DBMS_OUTPUT.PUT_LINE(VarDDD || ' - ' || VarNUM);
END;

```

*Resultado:*

74 - 123412333

Statement processed.

0.00 seconds

### 3.3.11.2. IF

Adiciona desconto na compra caso o cliente já tenha alguma compra anterior.

```

DECLARE
    VarDesconto VENDA.desconto%TYPE;
    VarCPF Cliente.pk_CPF_C%TYPE;
    VarCount number(4);
BEGIN
    VarCPF := 19006207071;
    select count(*) AS quantidade INTO VarCount
    FROM VENDA, CLIENTE
    WHERE VENDA.FK_CLIENTE_CPF = CLIENTE.PK_CPF_C AND CLIENTE.PK_CPF_C = VarCPF
group by CLIENTE.PK_CPF_C;
    If VarCount > 0 then
        VarDesconto := 0.1;
    Else
        VarDesconto := 0;
    End If;
    INSERT INTO Venda
VALUES(04,VarCPF,10123453401,TO_DATE('06/14/2019','MM/DD/YYYY'),VARDesconto,10
0.00);
    DBMS_OUTPUT.PUT_LINE(VarCPF || ' ' || VarDesconto || ' ' || VarCount);
END;

```

*Resultado:*

19006207071 .1 1

Statement processed.

0.00 seconds

Abaixa o preço do medicamento mais caro em 5% (se o valor do mesmo for acima de R\$100,00)

```
DECLARE
  VarMEDID MEDICAMENTO.PK_CODPRODUTO%TYPE;
  VarPRECO MEDICAMENTO.PRECO%TYPE;
BEGIN
  SELECT * INTO VarMEDID, VarPRECO
  FROM (SELECT MEDICAMENTO.PK_CODPRODUTO, MEDICAMENTO.PRECO
        FROM MEDICAMENTO
        ORDER BY PRECO DESC)
  WHERE ROWNUM = 1;
  IF VarPreco > 100 THEN
    update MEDICAMENTO
    set preco= preco * 0.95
    WHERE pk_codproduto=VarMEDID;
  END IF;
  DBMS_OUTPUT.PUT_LINE(VarMEDID || ' - ' || VarPRECO );
END;
```

*Resultado:*

2 - 140.67

1 row(s) updated.

PK_CODPRODUTO	PRECO	TIPO	QTDESTOQUE	NOME	QTMILIGRAMAS	CONTROLADO	TARJA
1	12	XAROPE	10	ACEBROFILINA	25	S	Vermelha
2	133.64	COMPRIMIDO	4	RITALINA	10	S	Preta
3	4.5	GEL	13	DICLOFENACO SÓDICO	10	N	N
4	7.2	COMPRIMIDO	33	IBUPROFENO	200	N	N
5	3.6	COMPRIMIDO	20	NIMESULIDA	100	N	N

### 3.3.11.3. LOOPS

Aplica desconto de 5% no preço de todos os medicamentos.

```
DECLARE
  VarCountMED Number(4);
  VarPRECO MEDICAMENTO.PRECO%TYPE;
  VarMEDID MEDICAMENTO.PK_CODPRODUTO%TYPE;
  VarNOME MEDICAMENTO.NOME%TYPE;
  VarX Number;
BEGIN
  SELECT COUNT(*)
  INTO VarCountMED
  FROM MEDICAMENTO;

  FOR varX IN 1..VarCountMED
  LOOP
```

```

SELECT PRECO, NOME
INTO VarPRECO, varNOME
FROM MEDICAMENTO
WHERE PK_CODPRODUTO = varX;
dbms_output.put_line ('PRECO ATUAL DO ' || VARNOME || ': ' || VarPRECO);

UPDATE MEDICAMENTO
SET preco= preco * 0.95
WHERE pk_codproduto=varX;
SELECT PRECO, NOME
INTO VarPRECO, varNOME
FROM MEDICAMENTO
WHERE PK_CODPRODUTO = varX;
dbms_output.put_line ('PRECO APÓS DESCONTO DO ' || VARNOME || ': ' ||
VarPRECO);
END LOOP;
END;

```

### Resultado:

```

PRECO ATUAL DO ACEBROFILINA: 12
PRECO APÓS DESCONTO DO ACEBROFILINA: 11.4
PRECO ATUAL DO RITALINA: 133.64
PRECO APÓS DESCONTO DO RITALINA: 126.96
PRECO ATUAL DO DICLOFENACO SÓDICO: 4.5
PRECO APÓS DESCONTO DO DICLOFENACO SÓDICO: 4.28
PRECO ATUAL DO IBUPROFENO: 7.2
PRECO APÓS DESCONTO DO IBUPROFENO: 6.84
PRECO ATUAL DO NIMESULIDA: 3.6
PRECO APÓS DESCONTO DO NIMESULIDA: 3.42

```

PK_CODPRODUTO	PRECO	TIPO	QTDESTOQUE	NOME	QTMILIGRAMAS	CONTROLADO	TARJA
1	11.4	XAROPE	10	ACEBROFILINA	25	S	Vermelha
2	126.96	COMPRIMIDO	4	RITALINA	10	S	Preta
3	4.28	GEL	13	DICLOFENACO SÓDICO	10	N	N
4	6.84	COMPRIMIDO	33	IBUPROFENO	200	N	N
5	3.42	COMPRIMIDO	20	NIMESULIDA	100	N	N

Informa o nome e cargo de todos os funcionários. Caso o cargo seja 'Gerente', informe também o salário.

```

DECLARE
  VarX Number;
BEGIN
  FOR varX IN (SELECT NOME, CARGO, SALARIO FROM FUNCIONARIO)
  LOOP
    If UPPER(varX.Cargo) = 'GERENTE' then
      dbms_output.put_line ('Nome: ' || varX.NOME || ' | Cargo: ' ||
VarX.Cargo || ' | Salário: ' || VarX.Salario);
    Else
      dbms_output.put_line ('Nome: ' || varX.NOME || ' | Cargo: ' ||
VarX.Cargo);
    End If;
  END LOOP;
END;

```

**Resultado:**

Nome: Jaime Rodrigues Araújo | Cargo: Operador de Caixa  
 Nome: Jorge Lucas Star Wars | Cargo: Operador de Caixa  
 Nome: Maria Carolina Araújo | Cargo: Gerente | Salário: 4300  
 Nome: Ruan dos Santos | Cargo: Operador de Caixa

Statement processed.

**3.3.12. STORED PROCEDURES**

Retira do estoque os medicamentos vendidos a partir do ID da VENDA.

```
CREATE OR REPLACE PROCEDURE Drop_Estoque(
  varIDVENDA IN VENDA.PK_IDVENDA%TYPE
)
IS
  varX NUMBER;
  VarNOMEMED VARCHAR(255);
  totalQTD NUMBER;
BEGIN
  FOR varX IN (SELECT pk_fk_MEDICAMENTO_CODPRODUTO, QTD
               FROM Itensvendidos
               WHERE pk_fk_Venda_idVenda = varIDVENDA)
  LOOP
    UPDATE MEDICAMENTO
    SET qtdEstoque = qtdEstoque - varX.QTD
    WHERE varX.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

    SELECT qtdEstoque INTO totalQTD FROM MEDICAMENTO
    WHERE varX.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;
    SELECT NOME INTO varNOMEMED FROM MEDICAMENTO
    WHERE varX.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

    dbms_output.put_line ('Nova quantidade de estoque de ' || varNOMEMED
    || ' : ' || totalQTD);
  END LOOP;
END Drop_Estoque;
```

**Resultado:**

```
begin
  Drop_estoque('3');
end;
```

Nova quantidade de estoque de DICLOFENACO SÓDICO : 8  
 Nova quantidade de estoque de RITALINA : 0

Statement processed.

PK_CODPRODUTO	PRECO	TIPO	QTDESTOQUE	NOME	QTD MILIGRAMAS	CONTROLADO	TARJA
1	11.4	XAROPE	10	ACEBROFILINA	25	S	Vermelha
2	126.96	COMPRIMIDO	0	RITALINA	10	S	Preta
3	4.28	GEL	8	DICLOFENACO SÓDICO	10	N	N
4	6.84	COMPRIMIDO	33	IBUPROFENO	200	N	N
5	3.42	COMPRIMIDO	20	NIMESULIDA	100	N	N

Adiciona no estoque os medicamentos comprados a partir do ID da COMPRA.

```
CREATE OR REPLACE PROCEDURE ADD_Estoque(
    varIDCOMPRA IN COMPRA.PK_IDCOMPRA%TYPE
)
IS
    varX NUMBER;
    VarNOMEMED VARCHAR(255);
    totalQTD NUMBER;
BEGIN
    FOR varX IN (SELECT pk_fk_MEDICAMENTO_CODPRODUTO, QTD
                  FROM ItensComprados
                  WHERE pk_fk_compra_idCompra = varidCompra )
    LOOP
        UPDATE MEDICAMENTO
        SET qtdEstoque = qtdEstoque + varX.QTD
        WHERE varX.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

        SELECT qtdEstoque INTO totalQTD FROM MEDICAMENTO
        WHERE varX.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;
        SELECT NOME INTO varNOMEMED FROM MEDICAMENTO
        WHERE varX.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

        dbms_output.put_line ('Nova quantidade de estoque de ' || varNOMEMED
        || ' : ' || totalQTD);
    END LOOP;
END ADD_Estoque;
```

*Chamada e Resultado:*

```
begin                                Nova quantidade de estoque de IBUPROFENO : 73
    ADD_estoque('1');
end;                                Statement processed.
```

### 3.3.13. TRIGGER

Coloque um gatilho para reduzir automaticamente a quantidade em estoque de um medicamento quando o mesmo for vendido.

```
CREATE OR REPLACE TRIGGER Drop_estoque_auto
AFTER insert ON itensVendidos
FOR EACH ROW
DECLARE
    VarNOMEMED VARCHAR(255);
    totalQTD NUMBER;
BEGIN
    UPDATE MEDICAMENTO
    SET qtdEstoque = qtdEstoque - :NEW.qtd
```



```

WHERE :NEW.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

SELECT qtdEstoque INTO totalQTD FROM MEDICAMENTO WHERE
:NEW.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;
SELECT NOME INTO varNOMEMED FROM MEDICAMENTO WHERE
:NEW.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

dbms_output.put_line ('Nova quantidade de estoque de ' || varNOMEMED || ' :
'|| totalQTD);
END;

```

### *Chamada e Resultado:*

```
INSERT INTO ITENSVENDIDOS VALUES(01,03,6);
```

Nova quantidade de estoque de ACEBROFILINA : 19

1 row(s) inserted.

Coloque um gatilho para aumentar automaticamente a quantidade em estoque de um medicamento quando o mesmo for comprado.

```

CREATE OR REPLACE TRIGGER ADD_estoque_auto
AFTER insert ON itensComprados
FOR EACH ROW
DECLARE
    VarNOMEMED VARCHAR(255);
    totalQTD NUMBER;
BEGIN
    UPDATE MEDICAMENTO
    SET qtdEstoque = qtdEstoque + :NEW.qtd
    WHERE :NEW.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

    SELECT qtdEstoque INTO totalQTD FROM MEDICAMENTO WHERE
:NEW.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;
    SELECT NOME INTO varNOMEMED FROM MEDICAMENTO WHERE
:NEW.pk_fk_MEDICAMENTO_CODPRODUTO = pk_CODPRODUTO;

    dbms_output.put_line ('Nova quantidade de estoque de ' || varNOMEMED || ' :
'|| totalQTD);
END;

```

### *Resultado:*

```
INSERT INTO ITENSCOMPRADOS VALUES(04,01,10, TO_DATE('10/12/2024' , 'MM/DD/YYYY'));
```

Nova quantidade de estoque de ACEBROFILINA : 29

1 row(s) inserted.