

SCRIPTS

Definindo Novos Tipos de Dados

Adicionando

```
sp_addtype 'TCEP', 'char(10)', 'null'  
GO  
sp_addtype 'TEstado', 'char(2)', 'null'  
GO  
sp_addtype 'TQuantidade', 'numeric(10,2)', 'not null'  
GO  
sp_addtype 'TTelefone', 'varchar (20)', 'null'  
GO  
sp_addtype 'TTipoOperacao', 'SmallInt', 'null'  
go  
sp_addtype 'TValorGrante', 'numeric(15,2)'
```

Removendo

```
sp_droptype TTipoOperacao  
GO  
sp_droptype TValorGrante
```

Criando Tabelas

```
/****** Table CategoriaContato *****/  
  
CREATE TABLE CategoriaContato (  
    CodCategoria int NOT NULL,  
    Nome          varchar (60) NULL  
)  
GO  
  
/****** Table Contato *****/  
  
CREATE TABLE Contato (  
    Tipo          char(1) NOT NULL CHECK (Tipo in ('P','E')),  
    Codigo        int NOT NULL,  
    CodigoSub     int NOT NULL,  
    Nome          varchar (60) NULL  
)  
GO
```

```
/****** Table Empresa *****/
```

```
CREATE TABLE Empresa (  
    CodEmpresa    int NOT NULL,  
    Nome          varchar (60) NULL,  
    RazaoSocial   varchar (60) NULL,  
  
    -- campos adicionais  
    DataCadastro  datetime NULL,  
    Notas         text NULL  
)  
GO
```

```
/****** Table Produto *****/
```

```
CREATE TABLE Produto (  
    CodProduto    int NOT NULL,  
    Nome          varchar(60) NULL,  
    Descricao     varchar(60) NULL,  
    QuantDisponivel TQuantidade NOT NULL CHECK (QuantDisponivel >= 0)  
DEFAULT 0,  
    QuantMinima   TQuantidade NULL,  
    Localizacao   varchar (50) NULL,  
    Preco         money NOT NULL CHECK (Preco > 0)  
)  
GO
```

```
/****** Table MovimentacaoProduto *****/
```

```
CREATE TABLE MovimentacaoProduto (  
    CodMovProduto int NOT NULL,  
  
    -- chave estrangeira da tabela Contato  
    TipoContato   char(1) NULL,  
    CodContato    int NULL,  
    CodSubContato int NULL,  
  
    -- chave estrangeira da tabela Produto  
    CodProduto    int NOT NULL,  
  
    Quantidade     TQuantidade NOT NULL,  
    DataMov        datetime NOT NULL DEFAULT (getdate()),  
    -- E = Entrada, S = Saída  
    TipoMov        char(1) NOT NULL CHECK (TipoMov in ('E','S'))  
)  
GO
```

```
/****** Table Pessoa *****/
```

```
CREATE TABLE Pessoa (  
    CodPessoa    int NOT NULL,  
  
    Nome         varchar (50) NULL,  
    Sexo        char(1) NOT NULL,  
    Fone        TTelefone NULL,  
    Fax         TTelefone NULL  
)  
GO
```

```
/****** Table Subdivisao *****/
```

```
CREATE TABLE Subdivisao (  
    -- chave primária  
    CodEmpresa    int NOT NULL,  
    CodSubdivisao int NOT NULL,  
    Nome         varchar (50) NULL,  
    Fone        TTelefone NULL,  
    Fax         TTelefone NULL,  
  
    -- colunas de endereço  
    Rua         varchar (50) NULL,  
    Bairro      varchar (25) NULL,  
    Cidade      varchar (40) NULL,  
    Estado      TEstado NULL,  
    CEP         TCEP NULL,  
  
    CGC         varchar (18) NULL,  
  
    -- colunas adicionais  
    DataCadastro datetime NULL,  
    Notas       text NULL  
)  
GO
```

```
/****** Table RelEmpresaCategoria *****/
```

```
CREATE TABLE RelEmpresaCategoria (  
    CodEmpresa    int NOT NULL,  
    CodCategoria  int NOT NULL  
)  
GO
```

```
/****** Table RelPessoaCategoria *****/
```

```
CREATE TABLE RelPessoaCategoria (  
    CodPessoa    int NOT NULL,  
    CodCategoria int NOT NULL  
)  
GO
```

```
/****** Table RelSubdivisaoPessoa *****/
```

```
CREATE TABLE RelSubdivisaoPessoa (  
    CodEmpresa    int NOT NULL,  
    CodSubdivisao int NOT NULL,  
    CodPessoa     int NOT NULL,  
  
    Cargo         varchar (30) NULL  
)  
GO
```

```
/****** Table MovAcumulado *****/
```

```
CREATE TABLE MovAcumulado (  
    CodProduto    int NOT NULL,  
    TotalVendas   TQuantidade,  
    TotalCompras  TQuantidade  
)  
GO
```

```
/***** Table Temporaria *****/
```

```
CREATE TABLE Temporaria (  
    Codigo        int NOT NULL,  
    Nome          varchar (50) NULL  
)
```

```
/***** Table Temporaria1 *****/
```

```
CREATE TABLE Temporaria (  
    Codigo        int NOT NULL,  
    Nome          varchar (50) NULL  
)
```

Alter Table

Campo

```
ALTER TABLE Pessoa
ADD Rua varchar(60) null,
Cidade varchar(30) null,
Bairro varchar(30) null,
CEP TCEP NULL,
Estado TEstado NULL,
CPF varchar (14) NULL,
DataCadastro datetime NULL DEFAULT (getdate()),
Notas text NULL
```

Renomear Tabela

```
sp_rename 'Temporaria1', TemporariaTeste
GO
sp_rename 'Temporaria.codigo', cod
```

Remover Tabela

```
drop table Temporaria, TemporariaTeste
```

Inserir Diversos Dados nas Tabelas já Criadas

```
dump transaction Contatos
with truncate_only, no_log
GO
```

```
/***** Pessoa *****/
```

```
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
values (1, 'Abilio Frenkel', 'F', '222-2870', '031-295-4095', 'Av.
Goi s, 345 sala 2', '', 'Goifnia', 'GO', '')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
values (2, 'Adão Dias', 'F', '261-8263', '222-4280', 'Rua C- 146,
661', '', 'Goifnia', 'GO', '')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
values (3, 'Adão Pereira', 'F', '234-7755', 'n0o lembra', 'Rua
2, 151', 'Setor Universit rio', 'Goifnia', 'GO', '')
```

```

insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (4, 'Adebaldo Nepomuceno', 'F', '243-6465', '291-3257',
'Av. 24 de Outubro, 45', '', 'Goifnia', 'GO', '74215-030')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (5, 'Adriano Jesus', 'M', '281-6779', '295-4217', 'Av.
Perimetral Norte, 3231', ' ', 'Goifnia', 'GO', ' ')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (6, 'Advar Roberto', 'M', '213-2000', '217-1573', 'Rua Dr.
Constfncio Gomes, 81 Setor Crimeia Leste', '', 'Goifnia', 'GO', '')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (7, 'Agnaldo LeEo', 'M', '271-7300', '222-4845', 'Rua 805,
130 Vila Santa Isabel', '', 'Goifnia', 'GO', ' ')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (8, 'Agnaldo Carlos', 'M', '201-3366', '201-5577', 'Av.
Par , 400 Campinas', '', 'Goifnia', 'GO', '')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (9, 'Alan Gualberto', 'F', '206-8080', '222-3566',
'Rodovia BR-060 Km 204 Zona Rural', 'Centro', 'Goifnia', 'GO',
'74005-010')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (10, 'Alessandro Pereira', 'F', '243-0143', '205-2700',
' ', 'Setor Rodovi rio', 'Goifnia', 'GO', '74.430-13')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (11, 'Alex Bueno', 'F', '215-1011', '295-1937', ' ', ' ',
'Goifnia', 'GO', '74.210-02')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (12, 'Alexandre Ant"nio', 'F', '292-7171', '212-2526',
'Rua 18, 30', '', 'Goifnia', 'GO', '')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (13, 'Alexandre Branco Soares Ferreira', 'F', '222-7100',
'205-4216', '', 'Centro', 'Goifnia', 'GO', '74055-110')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (14, 'Alfredo JEnior', 'F', '831-2090', '291-6499', 'Rua
125 n. 59', '', 'Goiania', 'Go', '')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CE
P)
  values (15, 'Ana Barbosa', 'F', '239-2600', '281-3077', 'Rua 9-A,
315', '', 'Goifnia', 'GO', '74.430-020')

```

```

insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (16, 'Ana Mariano', 'F', '222-4075', '235-1027', '',
'Setor Bueno', '', '', '74223-050')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (17, 'Ana de Barros', 'F', '234-1171', '234-6121', 'Av.
Independência, 5.597 Setor Aeroporto', 'Setor Aeroporto',
'Goifnia', 'GO', '74040-010')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (18, 'Andr, da Silva', 'F', '293-1100', '225-3400', '',
'', 'Goifnia', 'GO', '')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (19, 'Antônio Rodrigues Guimarães', 'M', '295-2020',
'292-6677', 'Av. T-9 321', 'Cidade Jardim', 'Goifnia', 'GO',
'74423-230')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (20, 'Antônio dos Reis', 'M', '220-1622', '234-3169', 'Av.
Goi s, 5.106', '', 'Goifnia', 'GO', '74.110-020')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (21, 'Antônio das Chagas', 'M', '297-4327', '2905901',
'Rua Getúlio Vargas, 205 Parque Anhanguera', 'Parque Amazonas',
'Goifnia', 'GO', '74835-620')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (22, 'Antônio Camargo', 'M', '290-1700', '207-1060', '',
'', 'Goifnia', 'GO', '')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (23, 'Antônio Silva', 'M', '201-6000', '281-5099', 'Av.
Anhanguera, 7830', 'Vila Beteu', 'Goifnia', 'GO', '74410-040')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (24, 'Aparecida da Silva', 'F', '281-3077', '281-3350',
'Rua Anxietá, 104 Setor Rodoviário', '', 'Goifnia', 'GO', '')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
  values (25, 'Armindo Gonzales', 'F', '205-3434', '222-7555', 'Av.
Castelo Branco, 10.133', '', 'Goifnia', 'GO', '')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CE
P)
  values (26, 'Arthur Gomes', 'M', '295-5626', '281-6779', ' ',
'Vila João Vaz', 'Goifnia', 'GO', '74445-190')
insert into
Pessoa(CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)

```

```

values (27, 'Ataides de Jesus', 'F', '261-8041', '201-4253', 'Av
C 03 Q 44 L 05', '', 'Goiania', 'GO', ' ')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
values (28, 'Augusto de Rezende', 'F', '215-1295', '201-4706',
'Av. Presidente Kennedy, 1655 St. SÆo Judas Thadeu', ' ', 'Goifnia',
'GO', ' ')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
values (29, 'Carlos da Silva Gomes', 'M', '281-0886', '234-8977',
'Av. Henrique Leal Qd. 22 lt. 14', 'Vila Boa', 'Goifnia', 'GO',
'74360-370')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
values (30, 'Carlos Paula', 'M', '291-5599', '201-3569', 'Rua 86,
117 Setor Sul', ' ', 'Goifnia', 'GO', ' ')
insert into
Pessoa (CodPessoa, Nome, Sexo, Fone, Fax, Rua, Bairro, Cidade, Estado, CEP)
values (31, 'Carlos Correia', 'M', '295-4119', '233-2637', 'Av.
Portugal esq/24', 'Setor Aerovi rio', 'Goifnia', 'GO', '74435-090')

```

Consultando Dados

Consultando Versão do SQL Server

```
Select @@version
```

Mostrando Valor de uma String

```
Select 'Teste'
```

Consultando Todas as Colunas

* Mostrar quais são as pessoas existentes nos contatos.

```
Select * from pessoa
```

Consultando em outro Banco de Dados

* Mostrar quais são os autores cadastrados no banco de dados Pubs?

```
select * from pubs..authors
```

Consultando outras Colunas

* Mostrar o nome, telefone e fax das pessoas cadastradas nos contatos.

```
Select nome, fone, fax from pessoa
```

Consultando outras Colunas, Mudando o Cabeçalho das Colunas Retornadas

* Mostrar as empresas cadastradas nos contatos.
select nome 'Nome Fantasia', razaosocial 'Razão Social',
DataCadastro Cadastro from empresa

Usando Condições

* Quais são as pessoas, cujo, estado é igual a 'Go' e cidade igual a 'Goiânia'?

```
select Nome, Fone  
from pessoa  
where cidade = 'Goiânia' and estado = 'Go'
```

Manipulando Expressões

* Mostrar os preços de cada produto após um aumento de 10%.

```
select nome Produto, Preco Preço, (preco * 1.1) "Preço com 10% de  
aumento"  
from produto
```

Funções Matemáticas

* Mostrar os preços de cada produto após um aumento de 10%, arredondando o valor com duas casas decimais?

```
select nome Produto, Preco Preço,  
round((preco * 1.1), 2) "Preço arredondado"  
from produto
```



```
select power(4,2), pi(), ascii('A')
```

Funções de Caracteres

* Mostrar os produtos com as descrições?

```
select Nome + '-' + descricao 'Produto/Descrição'  
from produto
```

* Mostrar os produtos cujos nomes podem aparecer com 20 caracteres, e o preço com 6 casas decimais antes da vírgula e 2 casas depois da vírgula.

```
Select substring(Nome, 1, 20) Produto , str(preco, 6, 2) Preço  
from produto
```

* Repetir a letra a dez vezes.

```
select replicate('a', 10)
```

Funções de Data/hora

```
select datacadastro 'Data Cadastro',
dateAdd(mm, 1, datacadastro) 'Adicionando 1 Mês' ,
dateDiff(dd, datacadastro, getdate())
'Subtraindo Datas',
datepart(yy, datacadastro) Ano,
datepart(dw, datacadastro) Semana
from empresa
```

Conversão de Dados

```
select convert(char(10), nome) 'Nome Subdivisão',
convert(char, datacadastro, 103)
'Data Formato Brasileiro'
from subdivisao
```

Condições de Pesquisa

* Quais são os produtos cuja quantidade disponível em estoque é menor que a quantidade mínima permitida?

```
Select nome , quantdisponivel, quantMinima
from produto
Where quantDisponivel < quantMinima
```

* Quais as pessoas que não foi informado o telefone?

```
Select nome from pessoa
where fone is null
```

* Quais as subdivisões que estão no estado de Goiás e Tocantins?

```
select nome, fone, estado
from subdivisao
where estado in ('Go', 'To')
```

* Quais as saídas realizadas no período de 01/06/98 a 04/06/98?

```
select codproduto, quantidade, datamov
from movimentacaoproduto
where dataMov between '06/01/98' and '06/04/98'
```

* Quais as pessoas que contém as letras 'cris' no meio (ou no início ou no fim)?

```
select nome from pessoa
where nome like '%cris%'
```

* Quais as cidades onde a empresa possui clientes, independente de ser pessoa ou subdivisão?

```
select cidade from pessoa
union
select cidade from subdivisao
```

Inserindo Linhas

```
INSERT INTO Pessoa
VALUES (400, 'PESSOA UM', 'M', '222-2222', '', 'R. Teste', 'B. DOS
LIMOEIROS', 'Goiânia', 'GO', '74090-123',
'2222222-21', '07/22/1998', 'Obs. Nenhuma')
```

```
INSERT INTO Pessoa (CodPessoa, Nome, Sexo)
VALUES (401, 'Pessoa401', 'F')
```

Insert com select

Para testar, iremos criar a tabela copiaempresa com a mesma estrutura da tabela empresa.

```
CREATE TABLE CopiaEmpresa (
  CodEmpresa      int NOT NULL,
  Nome            varchar (60) NULL,
  RazaoSocial     varchar (60) NULL,

  -- campos adicionais
  DataCadastro   datetime NULL DEFAULT (getdate()),
  Notas          text NULL
)
```

Copiar as Empresas Cujo Nome seja Maior que 'M'

```
insert into copiaempresa
select * from empresa
where nome > 'm'
```

Para incluir os dados numa tabela com a estrutura diferente Faça:

Criar a tabela copiapessoa

```
CREATE TABLE CopiaPessoa (  
    CodPessoa      int NOT NULL,  
    Nome           varchar (50) NULL,  
    Sexo           char(1) NOT NULL CHECK (Sexo in ('M','F')),  
    Fone           TTelefone NULL  
)
```

Copiar as Pessoas cujo Sexo = 'F'

```
insert into CopiaPessoa  
select codpessoa, nome, sexo, fone from pessoa  
where sexo = 'F'
```

Atualizando Dados

*** Alterar o estado das pessoas para 'Go' , quando a cidade for igual a 'Goiânia'**

```
update Pessoa  
set Estado = 'Go'  
where Cidade = 'Goiânia'
```

Update com select

Atualizar o total das vendas e compras quando seu valor for null. Mas não existe nenhum registro cadastrado na tabela movacumulado, portanto iremos incluir todos os produtos nesta tabela para depois fazer as alterações.

```
drop table movacumulado
```

```
go  
CREATE TABLE MovAcumulado (  
    CodProduto int NOT NULL ,  
    TotalVendas TQuantidade NULL ,  
    TotalCompras TQuantidade NULL  
)
```

```

go
insert movacumulado(codproduto)
        select codproduto from produto
go

select * from movacumulado

UPDATE MovAcumulado
SET TotalVendas =
    (select sum(Quantidade)
     from MovimentacaoProduto mp
     where mp.CodProduto =
           MovAcumulado.CodProduto
     and mp.TipoMov = 'S'),
TotalCompras =
    (select sum(Quantidade)
     from MovimentacaoProduto mp
     where mp.CodProduto =
           MovAcumulado.CodProduto
     and mp.TipoMov = 'E')
where TotalVendas is null
       or TotalCompras is null

select * from movacumulado

```

Excluindo Dados

*** Excluir todos os registros da tabela copiapessoa**

```
delete from copiapessoa
```

*** Excluir os registros da tabela copiaempresa, quando o código da empresa for igual a 3**

```
delete from copiaempresa where codempresa = 3
```

Exclusão usando Subconsulta

Excluir as linhas da tabela copiaempresa que existem na tabela empresa.

```
Delete from copiaempresa
where codempresa in (select codempresa from empresa)
```

Coloca Dados na Tabela de Funcionários

```
drop table funcionario
go
create table funcionario
```

```
(
  codigo int identity,
  nome char(60),
  tipo char(1),
  salario money
)

go

insert into funcionario
values('func1', 'T', 300)
go

insert into funcionario
values('func2', 'P', 300)
go
insert into funcionario
values('func3', 'P', 1000)
go
insert into funcionario
values('func4', 'T', 400)
go
insert into funcionario
values('func5', 'T', 1200)
go
insert into funcionario
values('func6', 'P', 1300)
go
insert into funcionario
values('func7', 'P', 200)
go
insert into funcionario
values('func8', 'P', 600)
go
insert into funcionario
values('func9', 'P', 700)
go
insert into funcionario
values('func10', 'T', 800)
```

Dados de Resumo

*** Quantas pessoas estão cadastradas na tabela de pessoas?**

```
Select count(*) from Pessoa
```

*** Quantas pessoas existem por categoria?**

```
select CodCategoria 'Código Categoria',  
       Count(*) 'Quantidade Pessoa'  
from RelPessoaCategoria  
group by CodCategoria
```

*** Relação de produtos da empresa com sua quantidade total vendida, e ordenados pela quantidade.**

```
select CodProduto, SUM(Quantidade)  
from MovimentacaoProduto  
where TipoMov = 'S'  
and DataMov between '7/6/98' and '7/7/98'  
group by CodProduto  
order by SUM(Quantidade) DESC
```

Funções Agregadas

```
select max(quantidade) 'Maior Quantidade',  
       min(quantidade) 'Menor Quantidade',  
       count(quantidade) 'Quantidade Total',  
       avg(quantidade) 'Média',  
       sum(quantidade) 'Total'  
from movimentacaoProduto
```

*** Qual a última venda realizada para cada código do produto?**

```
Select codproduto, max(datamov)  
from movimentacaoproduto  
where tipomov = 'S'  
Group by codproduto
```

*** Qual a menor quantidade vendida por cada código do produto?**

```
Select codproduto, min(quantidade) from movimentacaoproduto  
Where tipomovimentacao = 'S'  
Group by codproduto
```

*** Qual a quantidade total de vendas realizadas para cada código do produto?**

```
Select codproduto, sum(quantidade)  
from movimentacaoproduto  
Where tipomov = 'S'  
       Group by codproduto
```

Having

*** Qual a menor saída realizada para cada código do produto, quando a quantidade vendida for maior que 51?**

```
Select codproduto, min(quantidade)
from movimentacaoproduto
Where tipomov = 'S'
Group by codproduto
      Having sum(quantidade) > 51
```

Junções de Tabelas

*** Quais os produtos foram vendidos na empresa?**

```
Select distinct p.nome
from movimentacaoproduto m , produto p
where m.codproduto = p.codproduto
and m.tipoMov = 'S'
```

*** Quais as pessoas pertencem à categoria 'CL Clientes', mostrando as pessoas em ordem alfabética?**

```
Select p.nome
from Pessoa p, RelPessoaCategoria r, CategoriaContato c
where p.codpessoa = r.codpessoa and c.codcategoria =
r.codcategoria
and c.nome = 'CL Clientes' order by p.nome
```

Junção Usando Inner Join

*** Quais as subdivisões existentes para cada empresa?**

```
select e.Nome, s.Nome
from Empresa e inner join Subdivisao s on e.codEmpresa =
s.codEmpresa
order by e.nome, s.nome
```

Junção Cruzada ou Irrestrita

```
select e.Nome, s.Nome
from Empresa e cross join Subdivisao s
order by e.nome, s.nome
```

Junção Exterior

```
Select * from produto left join movimentacaoproduto
      On produto.codproduto = movimentacao.codproduto
```

Junção com Mais de Duas Tabelas

*** Quais as empresas que pertencem à categoria 'CL Clientes', mostrando as empresas em ordem alfabética?**

```
select e.nome
from RelEmpresaCategoria r
inner join Empresa e on r.codempresa = e.codempresa
inner join CategoriaContato c on r.codcategoria = c.codcategoria
where c.nome = 'CL Clientes'
order by e.nome
```

*** Relação das empresas e subdivisões com os nomes dos funcionários.**

```
select e.nome 'Empresa', p.nome 'Funcionários'
from RelSubdivisaoPessoa r
inner join Pessoa P on r.codpessoa = p.codpessoa
inner join subdivisao s on (r.codsubdivisao = s.codsubdivisao and
r.codempresa = s.codempresa)
inner join Empresa E on S.codEmpresa = E.codEmpresa
order by E.nome , P.nome
```

*** Listagem das empresas com nome, fax, bairro, cidade, estado e que pertencem à categoria igual a 2.**

```
select s.Nome,Fax,Bairro,Cidade,Estado
from Subdivisao s
      inner join Empresa e
      on s.CodEmpresa = e.CodEmpresa
      inner join RelEmpresaCategoria r
      on r.CodEmpresa = e.CodEmpresa
where r.CodCategoria = 2
order by Estado, Cidade, Bairro, s.Nome
```

Sub-Consultas

*** Porcentagem da quantidade de um produto em relação ao total de quantidades dos produtos que foram comprados pela empresa.**

```
select CodProduto, 100.0 * sum(Quantidade) / (select
sum(Quantidade) from MovimentacaoProduto where TipoMov = 'E') as
Porcentagem from MovimentacaoProduto where TipoMov = 'E' group by
CodProduto
```

```
Select Nome, (select sum(Quantidade) from MovimentacaoProduto m
where TipoMov = 'S' and m.CodProduto = p.CodProduto) as TotalVendas,
(select sum(Quantidade) from MovimentacaoProduto m where TipoMov
= 'E' and m.CodProduto = p.CodProduto) as TotalCompras from Produto
p order by nome
```

Para as próximas consultas iremos criar a tabela de funcionário. Para isso, executar os seguintes comandos:

```
create table funcionario ( codigo int identity, nome char(60), tipo
char(1), salario money )
```

Executar o script [dados_func.htm](#) para acrescentar dados na tabela de funcionários. A coluna tipo, se for P, indica professor e se for T, indica técnico.

Teste de Existência

*** Mostrar os professores somente se houver técnicos com o salário igual a 1200.**

```
Select nome , salario
From funcionario
Where funcionario.tipo = 'P' and
Exists (select * from funcionario where tipo = 'T' and salario =
1200)
```

*** Quais as empresas que não possuem pessoas cadastradas?**

```
Select CodEmpresa,CodSubdivisao,Nome
from Subdivisao s
where NOT EXISTS
(select *
from RelSubdivisaoPessoa rsp
where rsp.CodEmpresa = s.CodEmpresa
and rsp.CodSubdivisao = s.Codsubdivisao)
```

*** Qual a data da última venda de cada produto com suas respectivas quantidades?**

```
select codproduto, max(datamov),
       (select quantidade from
        movimentacaoproduto p
        where p.codproduto = mp.codproduto
        and p.datamov = max(mp.datamov))
from movimentacaoproduto mp
where tipomov = 'S'
group by codproduto
```

*** Relação das empresas e subdivisões com a quantidade de funcionários pertencente a cada empresa.**

```
select e.nome 'Empresa', count(*)
from RelSubdivisaoPessoa r
inner join Pessoa P on r.codpessoa = p.codpessoa
inner join subdivisao s on (r.codsubdivisao = s.codsubdivisao and
r.codempresa = s.codempresa)
inner join Empresa E on S.codEmpresa = E.codEmpresa
group by e.nome
```

Índices

(Índices clustered geralmente são utilizados quando não é feita muita inclusão de dados na tabela.)

```
CREATE UNIQUE CLUSTERED INDEX indNome ON CategoriaContato(nome)
```

```
CREATE CLUSTERED INDEX indNome ON Empresa(nome)
```

```
CREATE INDEX indNome ON Pessoa(nome)
```

Identity

```
Create table Departamento
(
           Codigo int not null identity,
           Nome varchar(60)
)
```

Default

```
create default
       DataAtual as getdate()
go
sp_bindefault DataAtual, 'empresa.datacadastro'
```

```
exec sp_bindefault DataAtual, 'subdivisao.datacadastro'
```

```
Alter table Subdivisao  
add default 'GO' for Estado
```

```
Alter table Pessoa  
add default 'GO' for Estado
```

Regras

```
create rule RegraSexo as @sexo in ('F', 'M')  
go  
sp_bindrule RegraSexo, 'pessoa.sexo'
```

```
ALTER TABLE MovimentacaoProduto  
add check (@TipoMov in ('E', 'S'))
```

Restrições

```
/***** Object: Table CategoriaContato *****/  
ALTER TABLE CategoriaContato  
ADD PRIMARY KEY NONCLUSTERED (CodCategoria)
```

```
/***** Object: Table Contato *****/  
ALTER TABLE Contato  
ADD CONSTRAINT PkContato PRIMARY KEY CLUSTERED (Tipo,  
Codigo, CodigoSub)
```

```
/***** Object: Table Empresa *****/  
ALTER TABLE Empresa  
ADD CONSTRAINT PkEmpresa PRIMARY KEY NONCLUSTERED (CodEmpresa)
```

```
/***** Object: Table Produto *****/  
ALTER TABLE Produto  
ADD CONSTRAINT PkProduto PRIMARY KEY NONCLUSTERED (CodProduto)
```

```
/***** Object: Table MovimentacaoProduto *****/  
ALTER TABLE MovimentacaoProduto  
ADD PRIMARY KEY NONCLUSTERED (CodMovProduto),  
FOREIGN KEY (CodProduto) REFERENCES  
Produto(CodProduto),  
FOREIGN KEY (TipoContato, CodContato, CodSubContato)  
REFERENCES Contato(Tipo, Codigo, CodigoSub)
```

```
/***** Object: Table Pessoa *****/  
ALTER TABLE Pessoa  
ADD PRIMARY KEY NONCLUSTERED (CodPessoa)
```

```
/****** Object: Table Subdivisao *****/  
ALTER TABLE Subdivisao  
    ADD PRIMARY KEY NONCLUSTERED (CodEmpresa, CodSubdivisao)
```

```
/****** Object: Table RelEmpresaCategoria *****/  
ALTER TABLE RelEmpresaCategoria  
    ADD CONSTRAINT PkRelEmpresaCategoria PRIMARY KEY  
NONCLUSTERED (CodEmpresa,CodCategoria),  
    CONSTRAINT FkCodEmpresaCategoria FOREIGN KEY  
(CodCategoria) REFERENCES  
    CategoriaContato(CodCategoria)
```

```
/****** Object: Table RelPessoaCategoria *****/  
ALTER TABLE RelPessoaCategoria  
    ADD PRIMARY KEY (CodPessoa, CodCategoria),  
    CONSTRAINT FkCodPessoaCategoria FOREIGN KEY  
(CodCategoria) REFERENCES  
    CategoriaContato(CodCategoria)
```

```
/****** Object: Table RelSubdivisaoPessoa *****/  
ALTER TABLE RelSubdivisaoPessoa  
    ADD PRIMARY KEY (CodEmpresa,CodSubdivisao,CodPessoa),  
    FOREIGN KEY (CodEmpresa,CodSubdivisao) REFERENCES  
Subdivisao,  
    FOREIGN KEY (CodPessoa) REFERENCES Pessoa
```

*** Criar estas tabelas que utilizam as restrições de integridade com o comando create.**

```
CREATE TABLE copiaPessoa (  
    CodPessoa int NOT NULL constraint PK_Pessoa primary key,  
    Nome varchar (50) NULL constraint U_NomePessoa Unique,  
    Fone TTelefone NULL ,  
    Fax TTelefone NULL ,  
    Sexo char (1) check (sexo in('F','M')),  
    Rua varchar (60) NULL ,  
    DataCadastro datetime default getdate() ,  
    Cidade varchar (30) default 'Goiânia' ,  
    Bairro varchar (30) NULL ,  
    CEP TCEP NULL ,  
    Estado TEstado NULL ,  
    CPF varchar (14) NULL ,  
    Notas text NULL )
```

```
CREATE TABLE CopiaRelPessoaCategoria (  
    CodPessoa int NOT NULL constraint  
FK_CopiaPessoa foreign key references copiapessoa,
```

```
        CodCategoria int NOT NULL
        primary key (codpessoa, codcategoria)
    )
```

Visões

```
CREATE VIEW EmpresaCategoria
    (CodEmpresa, Empresa, Categoria)
AS
select e.CodEmpresa, e.Nome, c.Nome
from Empresa e
    INNER JOIN RelEmpresaCategoria rec
    ON e.CodEmpresa = rec.CodEmpresa
    INNER JOIN CategoriaContato c
    ON rec.CodCategoria = c.CodCategoria
```

```
CREATE VIEW ProdutoRestrito
AS SELECT CodProduto, Nome, Localizacao
    FROM Produto
```

```
CREATE VIEW PessoasEmpresa as
select p.Nome 'Pessoa', e.nome 'Empresa', s.nome 'Subdivisao',
rs.cargo
from Pessoa p
    inner join RelSubdivisaoPessoa rs
    on p.CodPessoa = rs.CodPessoa
    inner join Subdivisao s
    on (rs.CodEmpresa = s.CodEmpresa
    and rs.CodSubdivisao = s.CodSubdivisao)
    inner join Empresa e
    on (s.codempresa = e.codempresa)
```

```
CREATE VIEW PessoaView
AS SELECT Nome, Cidade, Estado
    FROM Pessoa
    WHERE Nome like 'A%'
```

```
CREATE VIEW Saidas
AS SELECT DataMov, CodProduto, Quantidade, TipoMov
FROM MovimentacaoProduto
WHERE TipoMov = 'S'
WITH CHECK OPTION
```

Procedimentos

```
CREATE PROCEDURE BuscaPessoa
    @nome varchar(50)
AS
    declare @pesquisa varchar(50)
    declare @contagem int

    select @pesquisa = '%' + @nome + '%'
    select @contagem = count(*)
        from Pessoa
        where Nome like @pesquisa

    declare @mensagem varchar(100)

    IF @contagem != 0
    begin
        select @mensagem =
            convert(varchar,@contagem) +
            ' pessoas encontradas'
        print @mensagem

        select CodPessoa, Nome
        from Pessoa
        where Nome LIKE @pesquisa
        order by Nome
    end
    ELSE
    begin
        select @mensagem =
            'Não foi encontrado "' + @nome + "'"
        print @mensagem
    end
GO

/*****/

create procedure BuscaPessoaCategoria
    @nome varchar(50),
    @codCategoria int = 0
AS
    if @codCategoria = 0
    begin
        select * from Pessoa
        where Nome like '%' + @nome + '%'

        return 1
    end
    else begin
        select * from Pessoa
        where Nome like '%' + @nome + '%'
        and CodPessoa in
            (select CodPessoa from RelPessoaCategoria
```

```

        where CodCategoria = @codCategoria)

    return 2
end
GO

/*****/

create procedure ConsultaVendasProduto
    @codProduto int,
    @dataIni     datetime,
    @dataFim     datetime
as
select Quantidade, DataMov,
    TipoContato, CodContato, CodSubContato
from MovimentacaoProduto
where CodProduto = @codProduto
and   TipoMov = 'S'
and   DataMov between @dataIni
                        and   @dataFim
order by DataMov
/*****/

create procedure InicializarContato
as
delete from Contato

insert into Contato
    select 'P', CodPessoa, 0, Nome
    from Pessoa

insert into Contato
    select 'S', CodEmpresa, CodSubdivisao,
        Nome
    from Subdivisao
GO

/*****/

create procedure AtualizarVendasCompras
as
delete from MovAcumulado

insert into MovAcumulado (CodProduto, TotalVendas, TotalCompras)
select CodProduto,
    Preco*(select sum(Quantidade)
        from MovimentacaoProduto m
        where m.TipoMov = 'S'
        and m.CodProduto = p.CodProduto),
    Preco*(select sum(Quantidade)
        from MovimentacaoProduto m
        where m.TipoMov = 'E'
        and m.CodProduto = p.CodProduto)

```

```

from Produto p
GO
/*****/
CREATE PROCEDURE IncrementaProduto
@codProduto int, @QuantAdicionada TQuantidade
AS
    update produto
    set quantdisponivel = quantdisponivel +
        @quantAdicionada
    where
        codproduto = @codproduto
    if @@Error != 0
        return @@Error
    else
        return 0
GO
/*****/
CREATE PROCEDURE DecrementarProduto
@CodProduto int, @QuantRetirada TQuantidade
AS
    if (select quantdisponivel
        from produto
        where codproduto = @codproduto) <
        @QuantRetirada
    begin
        RaisError('Quantidade Insuficiente!',
            16, 1)
        return 1
    end
    update produto
    set quantdisponivel = quantdisponivel -
        @quantretirada
    where codproduto = @codproduto

    if @@Error != 0
        return @@Error
    else
        return 0
GO

```

Usar o procedimento Documentador, porque ele utiliza cursor e é um exemplo de como utilizar tabelas de sistemas.

```

create procedure Documentador
@tabela varchar(50) = NULL
as
if @tabela IS NULL

```

```

declare SOBJ cursor for          //declarando cursor com nome SOBJ
  select id,name from sysobjects
  where type = 'U'
  order by name
else
  declare SOBJ cursor for
  select id,name from sysobjects
  where name = @tabela
declare @id int, @nomeTabela varchar(50)
declare @msg varchar(100)

// Depois de criado, tenho que abrir o cursor. Após a abertura, ele
fica posicionado no início.

open SOBJ

fetch next from SOBJ into @id, @nomeTabela // busca o próximo
registro e guarda esses valores nas variáveis id, nomeTabela, para
percorrer a tabela tem que colocar num laço.

while @@fetch_status != -1
begin
  select @msg = '***** Tabela: ' + @nomeTabela
  print @msg
  select sc.colid Num, sc.name Coluna,
  convert(varchar(20),
  case
    when st.name in ("char", "varchar", "binary")
      then st.name + "(" + convert(varchar,sc.length)+")"
    when st.name in ("numeric", "decimal")
      then st.name + "(" + convert(varchar,sc.prec) + "," +
      convert(varchar,sc.scale) + ")"
    else st.name
  end) Tipo,
  case when sc.status & 0x08 = 0 then "NOT NULL " else "NULL " end
+
  case when sc.status & 0x80 != 0 then "IDENTITY" else null end
  Opções,
  sc.length 'Tam.bytes', sc.cdefault, sc.domain
from syscolumns sc LEFT JOIN systypes st
ON sc.usertype = st.usertype
where sc.id = @id

  fetch next from SOBJ into @id, @nomeTabela
end
close SOBJ
deallocate SOBJ // destroi o cursor do banco de dados
GO

```