

1. Criação de Banco de Dados (início dos comandos **DDL**)

Sintaxe geral:

```
create database nome  
[on {default | device1} [= size]  
    [,device2 [= size]...]  
[log on device3 [= size]  
    [,device4 [= size]...]  
[for load]
```

Exemplos:

```
create database vendas  
create database vendas on default = 256  
create database vendas on default = 50, novosdados = 25
```

2. Tipos de dados para utilizar nos create table

PARA DADOS	TIPO	TAMANHO
Caractere	<i>char</i> (<i>n</i>), <i>varchar</i> (<i>n</i>), <i>nvarchar</i> (<i>n</i>), <i>nchar</i> (<i>n</i>)	até <i>n</i> bytes
Numérico exato	<i>decimal</i> (<i>p,e</i>) ou <i>numeric</i> (<i>p,e</i>)	depende
Numérico aproximado	<i>float</i> , <i>real</i>	8,4 bytes
Numérico inteiro	<i>int</i> , <i>smallint</i> , <i>tinyint</i>	4,2,1 bytes
Monetário	<i>money</i> , <i>smallmoney</i>	8,4 bytes
Data hora	<i>datetime</i> , <i>smalldatetime</i>	8,4 bytes
Binário	<i>binary</i> (<i>n</i>), <i>varbinary</i> (<i>n</i>)	<i>n</i> bytes
Texto e imagens	<i>text</i> , <i>image</i> , <i>ntext</i>	-variável-
Outros	<i>bit</i> , <i>timestamp</i>	1 bit, 8 bytes

3. Criação das Tabelas de Dados

Sintaxe geral:

```
create table <nome-tabela>  
    (<descrição das colunas>);  
    (<descrição das chaves>);
```

Exemplos:

```
create table Cliente
(
    codcliente smallint not null unique,
    nomecliente char(20),
    endereco char(30),
    IE char(20)
);
```

Ou indicando a criação de índices, que agilizará o acesso nessa tabela através dessa chave:

```
create table Cliente (
    codcliente smallint not null unique,
    nomecliente char(20),
    endereco char(30),
    IE char(20))
CONSTRAINT PK_Cliente Primary Key (codcliente);
```

Indicando chave estrangeira:

```
create table Pedido (
    num_pedido int not null unique,
    prazo_entrega smallint not null,
    codcliente smallint not null,
    codvendedor
FOREIGN KEY (codcliente) references Cliente,
FOREIGN KEY (codvendedor) references Vendedor);
```

* references estabelece a restrição de integridade referencial (neste caso: obriga que só poderemos incluir dados de um pedido se tal codcliente já esteja cadastrado na tabela Cliente e se tal codvendedor já esteja cadastrado na tabela Vendedor). Para usar tal cláusula, as tabelas referenciadas já devem ter sido criadas.

```
create table Vendedor
(
    codvendedor smallint not null unique,
    nome_vendedor char(20),
    salario_fixo money,
    faixa_comissao char(1)
PRIMARY KEY (codvendedor)
);
```

```

create table ITEMPEDIDO
(
    numpedido int not null unique,
    codprod smallint not null unique,
    quant decimal,
    CONSTRAINT PK_ITEMPEDIDO
        PRIMARY KEY (numpedido, codprod),
    CONSTRAINT FK_PEDIDO
        FOREIGN KEY (numpedido)
        REFERENCES (Pedido),
    CONSTRAINT FK_PRODUTO
        FOREIGN KEY (codprod)
        REFERENCES Produto
);

```

3. Eliminação de uma tabela:

Drop table <tabela>

Ex.: Drop table Pedido

4. Alteração da Estrutura de uma tabela:

É possível mudar várias opções de definição de uma tabela depois que foi criada (create). Colunas podem ser acrescentadas, modificadas ou excluídas. As restrições PRIMARY KEY e FOREIGN KEY podem ser acrescentadas ou excluídas. A mudança de tipo de dados pode causar truncamento de dados ou mesmo ser impossível de ser realizada.

Como a criação de tabelas, se já registrar todas restrições, obriga uma ordem para criação das mesmas, muitas vezes faz-se a criação das tabelas, sem as restrições, e as altera depois.

Exemplos:

```

ALTER TABLE Cliente
    ADD CONSTRAINT PK_Cliente
        PRIMARY KEY (codcli)

```

```

ALTER TABLE Pedido
    ADD CONSTRAINT FK_Pedido
        FOREIGN KEY (codcli)
        REFERENCES Cliente (codcli)

```

```

ALTER TABLE Produto
    ALTER COLUMN valor_unitario char(12)

```

5. Obtendo dados de uma tabela (início dos comandos **DML**):

Uma das operações mais comuns realizadas com o SQL é a verificação (seleção) de dados armazenados. Esta operação é realizada com o comando SELECT.

A Sintaxe básica contém palavras-chave:

- SELECT – especifica as colunas que queremos selecionar
- FROM – especifica as tabelas que serão consultadas
- WHERE – especifica o critério de seleção das linhas desejadas

Exemplos:

5.1. select sem WHERE, lista todas as linhas da tabela, pois não foi imposto nenhum critério de seleção:

```
SELECT descricao, unidade_produto, val_unitario FROM Produto;
```

5.2. select que listará todas as colunas e todas as linhas da tabela Cliente:

```
SELECT * FROM Cliente;
```

5.3. uso de cabeçalho alternativo para as colunas de dados que serão apresentadas:

```
SELECT numero = codvendedor, nome = nome_vendedor, rendimentos = salario_fixo FROM Vendedor
```

5.4. aplicação de derivação ou de cálculos diversos:

```
SELECT nome_vendedor, salario_fixo = (salario_fixo *2) FROM Vendedor
```

5.5. Utilizando critério de seleção de linhas (seleção de somente algumas linhas):

5.5.1. Sintaxe básica:

```
SELECT <nome da(s) coluna(s)>  
FROM <tabela(s)>  
WHERE <critério de seleção>;
```

5.5.2. Utilizando operadores de comparação (= igualdade, <> ou != diferente, !> não maior, etc)

```
SELECT num_pedido, codprod, qtdprod FROM Item_pedido WHERE qtdprod = 35;
```

5.5.3. Utilizando operadores lógicos (and, or, no)

```
SELECT descr_prod FROM Produto WHERE unidprod = 'M' AND val_unit = 1.05;
```

```
SELECT nomecli, endereco FROM Cliente WHERE (CEP >= '30077000' AND CEP <= '300790999') OR cidade = 'São Paulo';
```

```
SELECT num_pedido FROM Pedido WHERE (prazo_entrega <> 15);
```

5.5.4. Utilizando operadores BETWEEN e NOT BETWEEN:

```
SELECT codprod, descr_prod FROM Produto WHERE val_unit BETWEEN 0.32 AND 2.00;
```

5.5.5. Utilizando operadores para strings (cadeia de caracteres) LIKE e NOT LIKE

```
SELECT * FROM Produto WHERE descr_prod LIKE 'Lápis Preto';
```

```
SELECT * FROM Produto WHERE descr_prod LIKE 'Lápis %';
```

* Neste caso recuperará as descrições com Lápis Preto, Lápis Cera, Lápis Borracha

```
SELECT * FROM Produto WHERE descr_prod LIKE 'Broca N_';
```

* Neste caso recuperará as descrições com Broca N1, Broca N9, Broca Nx

```
SELECT * FROM Produto WHERE descr_prod LIKE '%ão';
```

* Neste caso recuperará as descrições que terminam com "ão"

```
SELECT * FROM Produto WHERE descr_prod LIKE '[CM]%';
```

* Neste caso recuperará as descrições que começam com C ou com M

```
SELECT * FROM Produto WHERE descr_prod LIKE '[C-X]%';
```

* Neste caso recuperará as descrições que começam com C ou D ou E ... até X

```
SELECT * FROM Produto WHERE descr_prod LIKE 'M[^o]%';
```

* Neste caso recuperará as descrições que comece com M e não tenha o 'o' como segunda letra

5.5.6. Utilizando operadores baseados em listas IN ou NOT IN

```
SELECT Nome_vendedor FROM Vendedor WHERE faixa_comissao IN ('A', 'D', 'E');
```

5.5.7. Utilizando operadores baseados em valores não preenchidos (IS NULL) ou preenchidos (IS NOT NULL)

```
SELECT * FROM Cliente WHERE IE IS NULL;
```

5.6. Utilizando algumas funções interessantes:

5.6.1. Função de contagem de linhas (COUNT)

```
SELECT COUNT(*) FROM Vendedor WHERE Sal_fixo > 2500;
```

*Apresentará a quantidade de vendedores que tem salário fixo maior que 2500.

5.6.2. Função de somatório (SUM)

```
SELECT SUM(quant) FROM Item_pedido WHERE codprod = 78;
```

5.6.3. Função de mínimo e máximo (MIN, MAX)

```
SELECT MIN(Sal_fixo), MAX(Sal_fixo) FROM Vendedor;
```

5.6.4. Função de média (AVG)

```
SELECT AVG(Sal_fixo) FROM Vendedor;
```

5.6.5. Clausula DISTINCT (seleciona todos os conteúdos distintos de uma coluna)

```
SELECT DISTINCT unidade FROM Produto;
```

5.7. Listando informações agrupadas (GROUP BY)

```
SELECT Num_pedido, Total_Itens = COUNT(*) FROM Item_pedido GROUP BY  
num_pedido;
```

5.8. Seleção de colunas de tabelas relacionadas (JOIN) não pertence ao programa deste módulo.
Nem criação de VIEW.

6. Adição de Linhas em uma tabela (Insert)

```
INSERT INTO <nome da tabela> (<nome da(s) coluna(s)> VALUES (<valores>);
```

Exemplos:

```
INSERT INTO Produto VALUES (108, 'Parafuso', 'Kg', 1.25);
```

```
INSERT INTO Cliente (codcli, nome_cliente) VALUES (35, 'João da Silva');
```

7. Atualização de Linhas de uma tabela (UPDATE)

UPDATE <nome da tabela> SET <nome da(s) coluna(s)> = valor WHERE <condição>;

Exemplos:

UPDATE Produto SET val_unitario = 1.62 WHERE descricao = 'Parafuso';

UPDATE Vendedor SET Sal_fixo = (Sal_fixo * 1.27) + 100.00;

8. Remoção de linhas de uma tabela (DELETE)

DELETE FROM <nome da tabela> WHERE <condição>

Exemplo:

DELETE FROM Vendedor WHERE faixa_comissao IS NULL;