

1 A Linguagem SQL

SQL é a linguagem através do qual pedimos ao SGBD para realizar determinadas tarefas. Esta linguagem segue um padrão mundialmente aceito entre os principais fabricantes de sistema de gerenciadores de banco de dados. O seu aprendizado é de extrema importância pois, é através dela que realizamos as operações de manutenção do esquema (DDL), além da manipulação dos dados armazenados e realização de consultas sobre estes (DML). Veremos alguns dos principais comandos da linguagem SQL.

1.1 Create Table

Cria as tabelas embaixo do usuário no qual estamos conectados. Sua sintaxe segue o seguinte padrão.

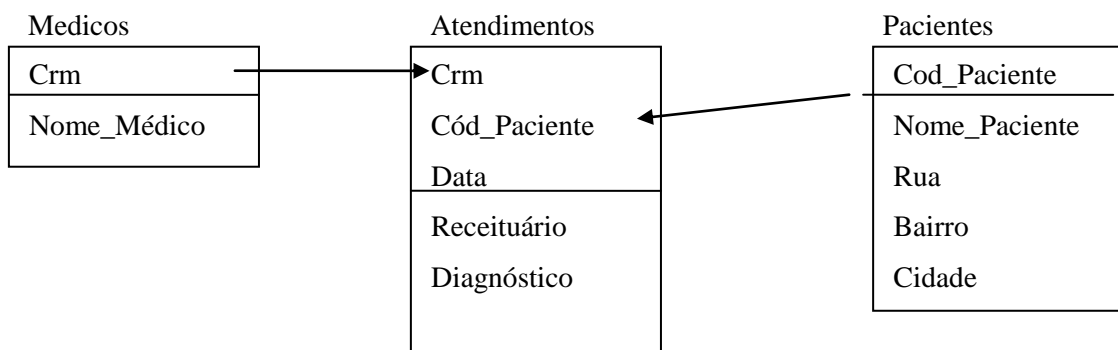
```
Create Table nome_tabela (  
    Campo1 Tipo1 [restricao sobre o campo1],  
    Campo2 Tipo2 [restricao sobre o campo2],  
    ...  
    CampoN TipoN [restricao sobre o campoN],  
    Primary Key (campo,...),  
    [Foreign Key (campo,...) references tabela_referenciada]  
)
```

restricao = Not Null – campo não pode conter nulos

= Default: valor a ser colocado no campo quando não for informado o conteúdo em uma inserção.

tipo = CHAR(tamanho), VARCHAR2(tamanho), NUMBER(tamanho,decimal), LONG, RAW(tamanho), LONG RAW(tamanho), DATE, ROWID, MLSLABEL

exemplo :



```
Create Table Medicos (  
Crm char(20) Not Null,  
Nome_Medico char(40) Not Null,  
Primary Key (Crm)  
);
```

```
Create Table Pacientes(  
Cod_Paciente Number(7,0) Not Null,  
Nome_Paciente char(40) Not Null,  
Rua char(40),  
Bairro char(20),  
Cidade char(20) default 'Campos dos Goitacazes',  
Primary Key (Cod_Paciente)  
)
```

```
Create Table Atendimento (  
Crm char(20) not null,  
Cod_Paciente Number(7,0) Not Null,  
Data Date,  
Receituário char(40),  
Diagnóstico char(40),  
Primary Key(Crm, Cod_Paciente,Data),  
Foreign Key (Crm) references Medicos,  
Foreign Key (Cod_Paciente) references Pacientes  
)
```

1.2 ALTER TABLE

Comando para alterar as definições originais de uma tabela. Ex:

Adicionar Atributo cidade na tabela de Clientes

```
Alter Table clientes ADD cidade char(20)  
Default 'Campos dos Goitacazes';
```

Apagar um Atributo

```
Alter Table clientes Drop cidade;
```

Sql Básico

Adicionar uma restrição de chave estrangeira (caso não tenha sido colocada)

Alter Table Itens_Vendidos Add

Foreign Key (codpro) references Produtos(codpro);

Alter Table Itens_Vendidos Add

Foreign Key (codvenda) references Vendas(codvenda) on delete cascade;

* on delete cascade faz com que caso a venda tenha sido apagada, os itens vendidos referentes aquela venda serão apagados também.

1.3 DROP TABLE

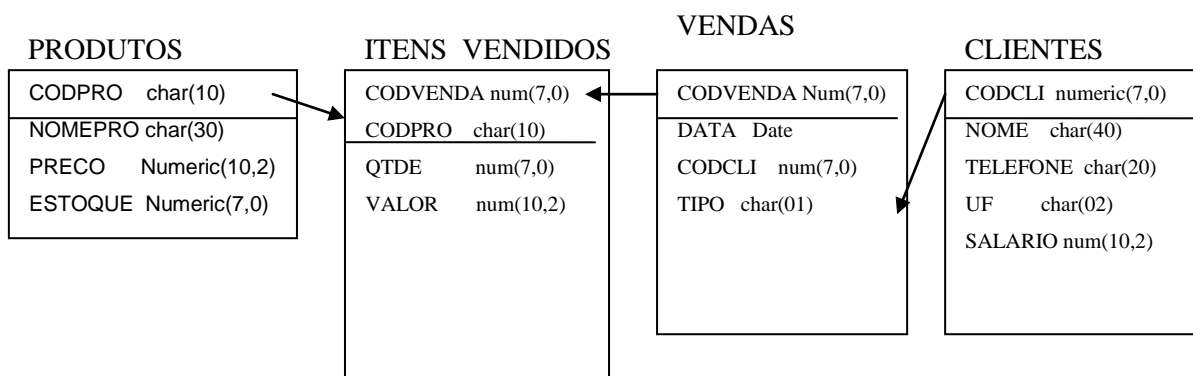
O comando drop table serve para apagar uma tabela, esta tabela não pode ter outras tabelas dependentes, assim para apagar a tabela de produtos temos que retirar a chave estrangeira da tabela de itens_vendidos que a referencia, ou colocar a palavra cascade na frente.

DROP TABLE MEDICOS;

1.3.1 Exercícios :

- Crie um usuário
- Considere o esquema e crie no seu usuário as tabelas dos esquema abaixo;
- Consulte a criação das tabelas através da visão user_tables e dos comandos desc;

:



1.4 INSERT

1.4.1 Inserção de tuplas (na ordem em que os atributos foram declarados)

Inserir o produto de código='00021', nome = 'Sabão em pó', Valor = 1,80 e quantidade = 20

```
Insert into Produtos values ('00021','Sabão em pó', 1.80 , 20);
```

```
Insert into Produtos (codpro,nomepro) values ('00022','Amaciante de Roupas');
```

1.4.2 Exercícios

Insira nas tabelas de 10 produtos, 3 clientes, 5 vendas e 15 itens vendidos nas tabelas criadas.

1.5 DELETE

Eliminar todos os registros de uma relação.

```
Delete From Clientes;
```

Eliminar os registros de uma relação que atendam uma determinada condição.

```
Delete From clientes  
where salario < 200;
```

Eliminar um registro específico

```
Delete From clientes  
where codcli = 2;
```

1.6 UPDATE

Serve para alterar os campos de um conjunto de registro. A cláusula set determina os campos que receberão os valores; A cláusula where determina em quais registros a mudança ocorrerá. A ausência da cláusula where determina que a mudança ocorrerá em todos os registros da tabela.

Reajuste todos os produtos em 20 %

- Update produtos
Set preco = preco * 1.20;

Altere o telefone do cliente 1 para '722-3456'

- Update clientes
Set telefone = '722-3456'
Where codcli = 1;

1.6.1 Exercícios

- Inserir o produto '00023', nome = 'Detergente', Quantidade=100 e preço = 1.20;
- Diminuir o preço de todos os produtos em 10%
- Apagar os clientes que ganham menos de 300;

1.7 Select

```
<comando-de-seleção> ::=  
SELECT      <lista-de-colunas>  
    FROM <lista-de-tabelas-ou-visões>  
    [WHERE   <condição-de-pesquisa>]  
    [GROUP BY <lista-de-colunas>  
        [HAVING   <condição-de-pesquisa>] ]  
    [ORDER BY <lista-de-colunas>]
```

1.7.1 Consultas Simples:

Visualizando o conteúdo de uma tabela

```
Select *  
From produtos
```

Listar todos os atributos selecionando os registros

```
Select *  
From Vendas  
Where data = '15/05/2000';
```

Sql Básico

1.7.2 Renomeando colunas

```
Select codpro as produto, nomepro as Descrição, preco  
From produtos  
Where preco >= 1.00;
```

1.7.3 Expressões com valores em colunas

Forneça o inventário dos produtos

```
Select codpro, nomepro, estoque * preco as inventario  
From produtos;
```

Codpro	Nomepro	Inventário
00001	Carne de 1 ^a .	1770.00
00002	Arroz	212.00
....

1.7.4 Uso de uma string resposta em uma coluna

```
Select nome, 'Marajá' as status  
From clientes  
Where salario > 1000;
```

Nome	Status
Joao	Marajá
Paulo	Marajá
....

1.7.5 Uso de %, 'like', 'not like'

Localize os clientes que têm telefone começando por 733

```
Select nome
From Clientes
Where telefone Like '733%';
```

1.7.6 Unindo dados de mais de uma tabela

Diga o nome dos clientes datas que eles realizaram compras

```
Select cliente.nome, vendas.data
From Clientes, Vendas
Where Clientes.codcli = Vendas.Codcli
      ou
Select C.nome, V.data
From Clientes C, Vendas V
Where C.codcli = V.codcli
```

Diga o nome dos clientes que realizaram compras no dia 15/05/2000

```
Select C.nome, V.data
From Clientes C, Vendas V
Where C. codcli = V. codcli {clausula de join}
And V.Data = '15/05/2000'
```

Diga o nome dos clientes e os produtos que eles já compraram

```
Select C.nome, P.nomepro
From Clientes C, Vendas V, Itens_Vendidos I, Produtos P
Where C. codcli = V.codcli {clausula de join}
And V.CodVenda = I.CodVenda {clausula de join}
And I.CodPro = P.Codpro {clausula de join}
```


1.7.7 Uso de In e Not In:

Liste os produtos que nunca foram vendidos.

```
Select P.codpro, P.nomepro  
From Produtos  
Where codpro not in  
    (Select codpro  
     From itens_vendidos);
```

Os atributos devem ser
compatíveis

Liste as os Clientes que não realizaram compras entre '01/05/2000' e 15/05/2000.

```
Select nome  
From Clientes  
Where codcli not in  
    (Select codcli  
     From Vendas  
     Where Data >= '01/05/2000'  
     And Data <= '15/05/2000');
```

1.7.8 Quantificadores: Any (existencial), All (universal)

Diga produtos que foram vendidos por valor menor que o preço de todos os produtos

```
Select I.Produtos  
From Itens_Vendidos I  
Where I.Valor <= All  
    (Select valor  
     From Produtos);
```

1.7.9 União (Union), Intercessão(Intersect), Diferença (Difference, Except ou Minus - Oracle)

Liste os produtos que foram vendidos no dia 15/05/2000 e também foram vendidos no dia 16/05/2000

```
Select Nomepro  
From Itens_vendidos I, Vendas V, Produtos P  
Where I.codvenda = V.codvenda  
And I.codpro = P.codpro  
And V.Data = '15/05/2000'
```

Intersect

```
Select Nomepro  
From Itens_vendidos I, Vendas V, Produtos P  
Where I.codvenda = V.codvenda  
And I.codpro = P.codpro  
And V.Data = '16/05/2000'
```

1.7.10 Agregações : Sum, avg, min, max e count

Liste o preço médio que foi vendido o produto '0001'.

```
Select avg(valor)  
From Itens_Vendidos  
Where codpro = '00001';
```

Liste o número compras do cliente 'MARIA'

```
Select Count(codvenda)  
From Vendas V, Clientes C  
Where V.codcli = C.codcli  
And C.Nome = 'MARIA'
```

1.7.11 Group by: ordena por grupos de acordo com os atributos da causa Select; agregações p/ grupo

Listar o preço médio de Vendem de cada produto.

```
Select codpro, avg(valor)
      From Itens_Vendidos
      Group by codpro;
```

Liste o número compras por cliente

```
Select C.codcli, Count(V.codvenda)
      From Vendas V, Clientes C
      Where V.codcli = C.codcli
      Group by C.codcli;
```

Liste os produtos e a quantidade vendida no mês de maio de 2000

```
Select P.Nome, Sum(I.Qtde) as TotQtd,
      Sum(I.Qtde * I.Valor) as Total_Valor
      From Produtos P, Itens_Vendidos I, Vendas V
      Where P.codpro = I.codpro
      And I.codvenda = V.codvenda
      And V.data >= '01/05/2000'
      And V.data <= '31/05/2000'
      Group by P.Nome
```

1.7.12 Uso de 'Having'

Liste os clientes que fizeram mais de 2 compras

```
Select C.codcli, C.Nome, Count(V.codvenda)
      From Vendas V, Clientes C
      Where V.codcli = C.codcli
      Group by C.codcli
      Having Count(V.codvenda) > 2;
```

1.8 *Uso do Order By*

Liste os clientes em ordem alfabética

```
Select Nome, Codcli, Telefone  
From Clientes  
Order by Nome Ascending
```

Liste os produtos e a quantidade vendida em ordem de faturamento (do maior para o menor)

```
Select P.Nome, Sum(I.Qtde) as TotQtd,  
Sum(I.Qtde * I.Valor) as Total_Valor  
From Produtos P, Itens_Vendidos I  
Where P.codpro = I.codpro  
Group by P.Nome  
Order by Sum(I.Qtde * I.Valor)
```

1.8.1 Exercícios

- Mostre todos os clientes que ganham mais de 850 e moram no 'RJ'
- Mostre os produtos cujo estoque esteja maior do que o somatório de suas vendas
- Mostre os clientes que fizeram menos de 2 compras
- Mostre os clientes e o numero de produtos distintos comprados, em ordem alfabética;

1.9 Create View <nome> AS < consulta >

Modo de salvar uma consulta como se fosse uma tabela, para uso futuro. Depois podemos referenciar esta tabela de maneira mais simples

Criar uma visão dos produtos cujo estoque esteja abaixo de 200

```
Create View Produtos_Estoque_Baixo As
Select codpro, nomepro, quantidade qtd
From Produtos
Where Quantidade < 200;
```

Uma vez criada podemos selecionar na cláusula from da instrução select

```
Select *
From Produtos_Estoque_Baixo;
```

CodPro	NomePro	Qtd
00002	Arroz	100
00010	Papel Toalha	1

Liste os produtos de estoque baixo que já foram vendidos :

```
Select PB.codpro
From Produtos_Estoque_Baixo PB, Itens_Vendidos I
Where PB.codpro = I.codPro
```

1.9.1 Exercícios

- Crie uma visão com os produtos mais vendidos
- Crie uma visão com todos os clientes que ganham mais de 850 e moram no 'RJ'

1.10 Índices

Create Indexes <nome> on <tabela(campos...)>

Os índices são estruturas que aceleram a busca de informações sobre os dados.

Devemos criá-los para as seguintes condições:

- Chaves Estrangeiras (ex: campo codcli da tabela de vendas);
- Campos de pesquisa ou ordenação (ex: campo nome da tabela de clientes);
- Campos de condições de filtro (ex: campo data da tabela de vendas);

Os índices deixam mais lentas as operações de inserção, alteração e deleção sobre as tabelas que os contiverem, por isso não devem ser utilizados indiscriminadamente.

Não devemos utilizá-los nas seguintes condições:

- Campos cujo o número de ocorrências diferentes for inferior a 4% (ex: Sexo, só existem 2 logo $100/2 = 50\%$);
- Tabelas pequenas demais, pois o Oracle prefere mantê-las inteiras na memória, desprezando a utilização dos índices.
- Chave primária (o Oracle já cria automaticamente);

Exemplo :

```
Create Indexe CLIENTES_NOME on CLIENTES(NOME)
```

1.10.1 Exercícios

- Crie os índices que você achar adequado no esquema estoque