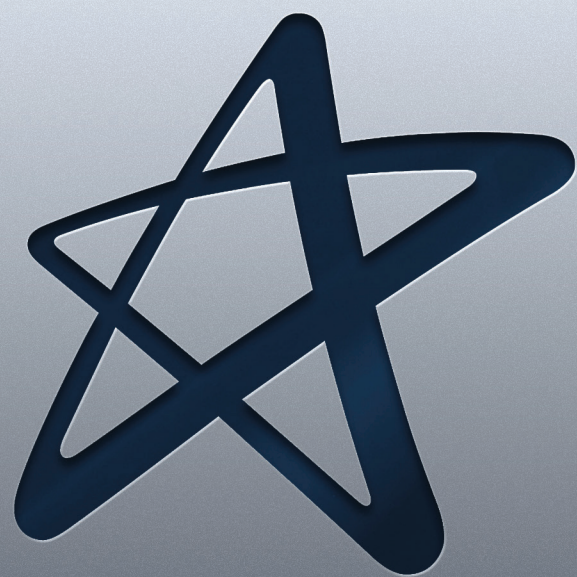


# Banco de Dados



**Cruzeiro do Sul Virtual**  
Educação a distância



# Material Teórico



Organização de Dados

**Responsável pelo Conteúdo:**

Prof. Ms. Alexander Gobbato Albuquerque

**Revisão Textual:**

Profa. Esp. Vera Lúcia de Sá Cicarone



# UNIDADE

## Organização de Dados



- Introdução
- Regras de Nomeação
- Restrições - Constraint
- Alteração de Tabela



**Objetivo de  
APRENDIZADO**

- Nesta Unidade veremos alguns comandos para a organização de dados, ou seja, comandos DDL (*Data Definition Language*) e SQL significa “*Structured Query Language*” (Linguagem Estruturada de Consulta).



### Atenção

Para um bom aproveitamento do curso, leia o material teórico atentamente antes de realizar as atividades. É importante também respeitar os prazos estabelecidos no cronograma.



## Contextualização

Vocês devem estar se perguntando o que seria esse DBA ou AD?

Com o conhecimento que irão adquirir no decorrer do curso, vocês poderão se tornar um Database Administration ou um Analista de Dados.

Nessa unidade iremos detalhar o modo de armazenamento interno [tamanho de campos, índices, tipo de preenchimento dos campos, nomenclaturas e etc] através da linguagem denominada SQL.

Com essas informações seremos capazes de criar uma estrutura para o armazenamento e visualização de qualquer informação.

Em nosso curso utilizaremos o SGDB Oracle 10g, que possui um material complementar de como instalar e proceder com a liberação do usuário HR.

## Introdução



A SQL foi desenvolvida originalmente pela IBM Research no início da década de 1970. Representa o padrão para linguagens de SGBD Relacionais, tendo sido padronizada pelo comitê ANSI/ISO.

Em 1999 foi publicado o atual padrão SQL/99 ou SQL3.

Quando os Bancos de Dados Relacionais estavam sendo desenvolvidos, foram criadas linguagens destinadas à sua manipulação. O Departamento de Pesquisas da IBM desenvolveu a SQL como forma de interface para o sistema de banco de dados relacional denominado SYSTEM R, no início dos anos 70.

Em 1986, o *American National Standard Institute* (ANSI) publicou um padrão SQL.

A SQL estabeleceu-se como linguagem padrão de Banco de Dados Relacional.

A SQL apresenta uma série de comandos que permitem a definição dos dados, chamada de DDL (*Data Definition Language*), composta, entre outros, pelos comandos *Create*, *Alter* e *Drop*.

## Regras de Nomeação



Os nomes das tabelas e colunas devem conter de 1 a 30 caracteres, sendo o primeiro caractere alfabético.

Os nomes devem conter apenas caracteres de a-z, A-Z, 0-9, \_, S e #.

Os nomes não podem ser iguais às palavras reservadas do Oracle.

## Tipo de Dados

Ao se criar a estrutura de uma tabela, é necessário que o usuário forneça, para cada coluna, as seguintes informações:

- » Tipo de Dado
- » Tamanho
- » Restrições

Tipo de Dado	Descrição
CHAR(n)	Campo fixo com tamanho máximo de 2000 bytes.
DATE	Permite data entre 1 de janeiro de 4712 AC até 31 de dezembro de 4712 DC
LONG	Caractere variável com tamanho de até 2 Gb
VARCHAR2(n)	Campo do tipo caractere com tamanho variável e limitado a 4000 bytes.
NUMBER(n,d)	Onde n é o número de dígitos e d o número de casas decimais.

## Restrições - Constraint



As restrições são regras básicas estabelecidas para o preenchimento de uma ou mais colunas da tabela e são definidas ao final da especificação de cada coluna ou ao final do comando.

Entre as restrições encontram-se:

- » chaves primárias;
- » chaves únicas;
- » chaves estrangeiras;
- » identificadores de campos obrigatórios;
- » condições para valores permitidos para determinado campo.

Constraint	Descrição
Not Null	Especifica que esta coluna não pode conter valores nulos.
Unique	Especifica uma coluna ou combinação de colunas que terão seus valores únicos na tabela.
Primary Key	Identifica a unicidade de cada linha na tabela.
Foreign Key References	Estabelece um relacionamento entre a chave estrangeira e a chave primária da tabela referenciada.
Check	Especifica uma condição que deve ser verdadeira, obedecendo a uma regra do negócio.

### Criação de Tabela

```
CREATE TABLE nome_tabela
( nome_coluna tipo de dado | constraint_tabela ),
( nome_coluna tipo de dado | constraint_tabela )
```

#### Onde:

Nome\_tabela à é o nome da tabela

Nome\_coluna à é o nome da coluna

Tipo de dado à é o tipo de dado da coluna

Constraint\_tabela à é a “constraint” ou restrição para a coluna.

### Constraint Primary Key

Especifica uma ou mais colunas que compõem a chave primária de uma tabela.

Uma forma mais organizada, após definirmos todos os campos da tabela, é definirmos as restrições.



### CREATE TABLE Cliente

```
(
    cd_cliente          number (4),
    nm_cliente          varchar2 (50),
    ds_endereco         varchar2 (70),
    cd_municipio         number (5),
    sg_estado           char (2),
    nr_cep               varchar2 (8),
    nr_ddd               number (3),
    nr_fone              number(7),
    ie_sexo              char(1),
    constraint cliente_cd_cliente_pk primary key (cd_cliente)
)
```

Padronizando a restrição, seguiremos o seguinte padrão:

Constraint=> **nome\_tabela\_nome\_campo\_tipodaconstraint**

## Constraint Primary Key Composta

Aqui é apresentado um exemplo de uma PK (Primary Key) composta por 2 atributos: cd\_cliente e dt\_compra

### CREATE TABLE Histórico

```
(
    cd_cliente          number (4),
    dt_compra           date,
    vl_compra           number (12,2),
    CONSTRAINT Historico_PK PRIMARY KEY (cd_cliente, dt_Compra)
)
```

## Constraint Unique

Define uma ou mais colunas que não podem ter valor repetido em mais de uma linha da tabela.

Por exemplo, não existem duas pessoas com o mesmo CPF ou número do PIS, mas esses campos não serão colocados como chave primária.

**CREATE TABLE Estado**

```
(  
Sg_Estado      char(2) primary key,  
Nm_Estado      varchar2 (35),  
constraint Estado_nm_Estado_UN UNIQUE (nm_Estado)  
)
```

**Constraint FOREIGN KEY**

Referencia um atributo que é chave primária de outra tabela com o propósito de implementar o relacionamento entre tabelas.

**CREATE TABLE Cliente**

```
(  
cd_cliente      number (4),  
    nm_cliente      varchar2 (50),  
    ds_endereco      varchar2 (70),  
    cd_municipio      number (5),  
    sg_estado      char (2),  
    nr_cep      varchar2 (8),  
    nr_ddd      number (3),  
    nr_fone      number(7),  
    ie_sexo      char(1),  
    constraint cliente_sg_estado_fk foreign key (sg_estado) references Estado(sg_estado)  
)
```

Neste tipo de constraint, relacionamentos que utilizem mais de uma coluna (chave composta) podem ser criados.

Para definição desta constraint, utilizamos o padrão [nome\_tabela/atributo/tipo\_constraint]

**Regras:**

Caso o tipo de dados da coluna na tabela inicial e na tabela referenciada sejam diferentes, será apresentado um erro;

Caso a tabela referenciada não possua chave primária (a foreign key será estabelecida sobre a chave primária da tabela referenciada);

O uso de chaves estrangeiras garante que não existirão linhas órfãs nas tabelas-filhas (tabelas que possuem dados que devem estar cadastrados previamente em outra tabela, denominada tabela mãe).

## Constraint CHECK

Define um conjunto de valores permitidos ou condição para inserção de valores em uma determinada coluna.

### **CREATE TABLE Cliente**

```
(  
cd_cliente          number (4),  
    nm_cliente      varchar2 (50),  
    ds_endereco     varchar2 (70),  
    cd_municipio    number (5),  
    sg_estado       char (2),  
    nr_cep          varchar2 (8),  
    nr_ddd          number (3),  
    nr_fone         number(7),  
    ie_sexo         char(1),  
    constraint cliente_ie_sexo_ck check(ie_sexo in ('F', 'M'))  
)
```

## Constraint NOT NULL

Indica que é obrigatória a inserção de algum valor nessa coluna. Somente pode ser declarado junto à coluna e não recebe nome da restrição.

### **CREATE TABLE Cliente**

```
(  
cd_cliente          number (4),  
    nm_cliente      varchar2 (50) not null,  
    ds_endereco     varchar2 (70) not null,  
    cd_municipio    number (5),  
    sg_estado       char (2),  
    nr_cep          varchar2 (8),  
    nr_ddd          number (3),  
    nr_fone         number(7),  
    ie_sexo         char(1)  
)
```

## Alteração de Tabela



Após a criação da estrutura de uma tabela, pode-se incluir (ADD), excluir (DROP) ou modificar (MODIFY) colunas ou constraints e desabilitar constraints dessa tabela. Utilizaremos o comando ALTER.

### **Acrescentar campos em uma tabela**

```
ALTER TABLE CLIENTE  
ADD( IE_FISICA_JURIDICA CHAR(1))
```

### **Alterando obrigatoriedade de atributos em uma tabela.**

```
ALTER TABLE CLIENTE  
MODIFY (nm_cliente not null);
```

### **Modificando o tipo de atributos em uma tabela.**

```
ALTER TABLE CLIENTE  
MODIFY (DS_ENDERECO NUMBER(3));
```

### **Modificando o tamanho dos atributos em uma tabela.**

```
ALTER TABLE CLIENTE  
MODIFY (NR_CEP VARCHAR2(12));
```

### **Acrescentar restrições a uma tabela.**

```
ALTER TABLE CLIENTE  
ADD(CONSTRAINT CLIENTE_IE_FISICA_JURIDICA_CK  
CHECK(IE_FISICA_JURIDICA IN('F','J')));
```

### **Desabilitar uma restrição de uma tabela.**

```
ALTER TABLE CLIENTE  
DISABLE CONSTRAINT CLIENTE_IE_FISICA_JURIDICA_CK;
```

### **Excluir uma restrição de uma tabela.**

```
ALTER TABLE CLIENTE  
DROP CONSTRAINT CLIENTE_IE_FISICA_JURIDICA_CK
```

## **Exclusão de Tabela**

Após a criação da estrutura de uma tabela ou alteração da sua estrutura, podemos excluí-la através do comando DROP.

Ao excluir uma tabela, todas as constraints e os dados inseridos são deletados fisicamente.

`DROP TABLE CLIENTE.`

## Material Complementar

- [http://arquivos.cruzeirosulvirtual.com.br/materiais/disc\\_graduacao/2014/1sem/1mod/ban\\_dad/un\\_I/mat\\_comp/mat\\_comp.pdf](http://arquivos.cruzeirosulvirtual.com.br/materiais/disc_graduacao/2014/1sem/1mod/ban_dad/un_I/mat_comp/mat_comp.pdf)



## Referências

COSTA, Rogério Luis de C. **SQL: guia prático**. 2. ed. Rio de Janeiro: Brasport, 2006.

MORELLI, Eduardo M. Terra, 1996. **Oracle 9i Fundamental: Sql, Pl/SQL e Administração**. São Paulo: Érica, 2002

SILBERSCHATZ, A. **Sistema de bancos de dados**. São Paulo: Pearson Education do Brasil, 2004.



**Cruzeiro do Sul**  
Educatonal