



++ Tabela EMP

Nome	Nulo?	Tipo
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

++ Tabela DEPT

Nome	Nulo?	Tipo
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

## CRIAÇÃO E MANIPULAÇÃO DE TABELAS: AULA DE LABORATÓRIO NR. 1

**Objetivo:** Aprender a criar tabelas, inserir, alterar e excluir dados na tabela criada em um banco de dados utilizando a linguagem SQL padrão ANSI 92. Nessa aula serão introduzidos os primeiros comandos DDL da linguagem SQL. Antes de iniciar essa aula, crie o usuário lab com senha lab: **create user lab identified by lab;** Dê privilégios: **Grant all privileges to lab;** Conect: **connect lab/lab;**

**TEORIA:** O comando SQL que permite a criação de uma tabela no banco de dados é o **CREATE TABLE**. Apesar de ser um comando fácil de utilizar, possui muitas particularidades que o torna bastante versátil, mas de certa forma, complexo. A melhor forma de se conhecer esse comando é exercitando exemplos do seu uso e ter uma referência rápida do comando.

**Exemplo1:** Criação da tabela DEPT descrita anteriormente.

```
CREATE TABLE DEPT(
    DEPTNO    NUMERIC(2)    DEFAULT 99 NOT NULL,
    DNAME     VARCHAR(14),
    PRIMARY KEY (DEPTNO),
    UNIQUE    (DNAME)
);
```

No comando anterior, a linha 2 possui a cláusula DEFAULT que permite que um valor padrão seja assumido pela coluna, caso o usuário não informe o valor desejado no comando insert. Na linha 5, DNAME é alterada para NÃO aceitar valores repetidos (cláusula UNIQUE).

**Exemplo2:** Comando para criar a tabela Emp com dados dos empregados.

```
CREATE TABLE EMP (
    Empno     NUMERIC(4)    NOT NULL,
    Ename     VARCHAR(10) ,
    Job       VARCHAR(9) ,
    Mgr       NUMERIC(4) ,
    Hiredate  DATE ,
    Sal       NUMERIC(7,2) ,
    Comm      NUMERIC(7,2) ,
    Deptno    NUMERIC(2) not null,
    PRIMARY KEY (empno) ,
    FOREIGN KEY (mgr) REFERENCES Emp(empno) ,
    FOREIGN KEY (Deptno) REFERENCES Dept(Deptno)
);
```

O comando acima cria a tabela **Emp**. Para cada coluna da tabela, existe uma linha com o seguinte padrão (linhas 2 a 9) :  
**nome da coluna domínio NULL/NOT NULL**  
 Na linha 10, o comando informa que a coluna **empno** é chave primária da tabela, enquanto que as linhas 11 e 12, informam que **mgr** e **Depto** são colunas estrangeiras (FOREIGN KEY). A primeira coluna referencia a coluna **empno** da própria tabela **Empregado** e, a segunda, referencia a coluna **deptno** da tabela **DEPT** criada anteriormente.

A referência do comando CREATE TABLE completa:

```
CREATE [ [ LOCAL ] { TEMPORARY | TEMP } ] TABLE nome_da_tabela (
  { nome_da_coluna tipo_de_dado [ DEFAULT expressão_padrão ] [
    restrição_de_coluna [, ... ] ]
    | restrição_de_tabela } [, ... ]
)
[ INHERITS ( tabela_ascendente [, ... ] ) ]
[ WITH OIDS | WITHOUT OIDS ]
```

onde restrição\_de\_coluna é:

```
[ CONSTRAINT nome_da_restrição ]
{ NOT NULL | NULL | UNIQUE | PRIMARY KEY |
  CHECK (expressão) |
  REFERENCES tabela_referenciada [ ( coluna_referenciada ) ] [ MATCH FULL |
  MATCH PARTIAL ]
  [ ON DELETE ação ] [ ON UPDATE ação ] }
[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```

e restrição\_de\_tabela é:

```
[ CONSTRAINT nome_da_restrição ]
{ UNIQUE ( nome_da_coluna [, ... ] ) |
  PRIMARY KEY ( nome_da_coluna [, ... ] ) |
  CHECK ( expressão ) |
  FOREIGN KEY ( nome_da_coluna [, ... ] ) REFERENCES tabela_referenciada [ (
  coluna_referenciada [, ... ] ) ]
  [ MATCH FULL | MATCH PARTIAL ] [ ON DELETE ação ] [ ON UPDATE ação ] }
[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```

Para uma referência completa, acesse:

<http://www.postgresql.org.br/referencia/sql-createtable.html>

**TEORIA:** Uma vez criada a tabela, deve-se incluir os dados utilizando a linguagem SQL. O comando utilizado é o INSERT, que tem forma geral a seguir:

```
INSERT INTO nome_da_tabela (coluna1, ..., coluna n) VALUES (valor1,..., valor n);
```

**Exemplo 3: Comando para inserir um departamento na tabela DEPT.**

```
Insert into DEPT (deptno, dname) values (1, 'RH')
```

**Atenção:** Os dados das colunas do tipo string ou character e DATA são colocados entre aspas simples.

**TEORIA:** Para alterar uma informação que já foi incluída na tabela, utiliza-se o comando UPDATE, que tem forma geral a seguir:

```
UPDATE nome_da_tabela SET coluna = valor WHERE condição;
```

**Exemplo4: Comando para atualizar linha(s) de uma tabela.**

```
Update DEPT set DNAME = 'VENDAS' where DEPTNO = 1;
```

O comando anterior muda o nome do departamento de número 1. Caso não se coloque a cláusula WHERE, o SQL atualiza todas as linhas da tabela, portanto deve-se ter cuidado ao emitir o comando acima.

**TEORIA:** Para excluir uma linha da tabela, utiliza-se o comando DELETE, que tem forma geral a seguir:

```
DELETE FROM nome_da_tabela WHERE condição;
```

**Exemplo5: Comando para excluir linha(s) de uma tabela.**

```
delete from DEPT where DEPTNO = 1;
```

O comando anterior exclui o departamento de número 1. Caso não se coloque a cláusula WHERE, o SQL EXCLUI TODAS as linhas da tabela departamento, portanto deve-se ter cuidado ao realizar o comando acima. Excluir a tabela completamente (dados e estrutura) use DROP TABLE DEPT.

## EXERCÍCIOS PRÁTICOS

1. Crie a tabela DEPT, mostrada no início da aula de laboratório;
2. Crie a tabela EMP mostrada no início da aula de laboratório;
3. Inclua pelo menos dois departamentos e três empregados;
4. Inclua o empregado José que trabalha no departamento de número 1;
5. Altere o nome de José para Paulo José;
6. Aumente o salário de Paulo José em 10%;
7. Altere a tabela EMP para que o valor DEFAULT do sal (salário) seja 300 reais;  
Solução: ALTER TABLE EMP MODIFY SAL DEFAULT 300;
8. Altere a comissão (COMM) de cada funcionário para que seja 5% do seu salário;