

Engenharia de Computação  
Laboratório de BD I – 1º Sem/2024  
Prof. Evandrino G. Barros (evandrino@cefetmg.br)

Aula 03 – Alteração em esquemas de bancos de dados relacionais e manipulação de dados - 05/04/2024.

Roteiro de aula prática sobre comandos de definição de dados (DDL – *Data Definition Language*) e de manipulação de dados (DML – *Data Manipulation Language*). O script completo desse roteiro também está disponível no AVA.

Siga o roteiro abaixo, executando cada um dos exemplos na sequência dada.

### I) Comandos de Definição de Dados em SQL (DDL – *Data Definition Language*)

Com comandos DDL é possível a:

1. Criação de tabelas
2. Alteração de tabelas (altera uma tabela já criada).

As alterações possíveis são:

- a. Inclusão de chave (primary key)
- b. Inclusão de coluna
- c. Remoção de coluna
- d. Remoção de chave
- e. Inclusão de chave estrangeira (foreign key)
- f. Remoção de chave estrangeira (remover constraint fk)
- g. Alteração de nome de coluna
- h. Alteração do tamanho de coluna
- i. Adição de chave única (unique key)
- j. Remoção de tabelas
- k. Adição de regra de validação (check)  
**Remoção de regra de validação (check)**
- l. Adição de valor default em uma coluna
- m. Remoção de um valor default em uma coluna

#### 1) Criação de tabelas

```
create table DEPART  
(deptno number(2),  
  nome varchar2(20),  
  criacao date default sysdate); -- sysdate é data atual
```

#### 2) Alteração de tabelas

- a) Inclusão de uma chave primária. A chave primária só será criada se não houver valores duplicados já existentes na(s) coluna(s) da chave.

```
alter table DEPART  
add constraint pk_DEPART primary key (deptno);
```

Se violarmos a chave primária, o sgbd trará o erro com o nome da regra:

```
insert into DEPART (Deptno, nome)  
values (10,'Vendas');
```

```
select * from DEPART;
```

```
insert into DEPART (Deptno, nome)  
values (10,'Vendas');
```

- b) Inclusão de uma nova coluna (coluna local varchar2(20)). Só é possível incluir colunas obrigatórias se a tabela estiver vazia. Se não for especificado que a coluna é not null, ela é null por DEFAULT.

```
alter table DEPART  
add local varchar2(20) not null;
```

Se a tabela já tiver dados, a nova coluna terá nulo em todas as linhas, mas somente se tiver sido permitidos valores nulos para ela (não é not null);

Para resolver isso: remover todas as linhas ou permitir null;

Para remover todas as linhas: delete from DEPART;

- c) Remoção de uma coluna.

```
alter table DEPART  
drop column criacao;
```

Se a coluna a ser removida tiver uma chave estrangeira que a referencia, a remoção não será possível. Para que a remoção realmente aconteça, é necessário o uso de opção CASCADE CONSTRAINTS, conforme exemplo a seguir.

```
Create table TESTE1 (      t1c1 number(5) primary key,  
                           t1c2 number(5));  
Create table TESTE2 (      t2c1 number(5) references TESTE1,  
                           t2c2 number(5));  
alter table TESTE1 drop column t1c1; -- s/ cascade  
                                   -- remoção proibida  
alter table TESTE1 drop column t1c1 CASCADE CONSTRAINTS;
```

- d) Remoção de chaves (constraints). A coluna que servia de chave, permanece, mas sem a regra.

```
alter table DEPART  
drop constraint pk_DEPART;
```

- e) Inclusão de chave estrangeira. Se houver dados na tabela que violem a chave estrangeira, tal chave não poderá ser estabelecida. Observe que uma coluna só se torna chave estrangeira quando a restrição de chave estrangeira é definida sobre ela. O default no comportamento da remoção da pk é de bloqueio.

```
Create table EMPREGADO  
(mat number (5) primary key,  
nome varchar2(20),  
deptno number(2));
```

```
alter table DEPART  
add constraint pk_dept  
primary key (deptno);
```

```
alter table EMPREGADO  
add constraint fk_emp_dept  
foreign key (deptno)  
references DEPART(deptno);
```

- f) Remoção de chave estrangeira: remoção da integridade referencial sem remoção da coluna que servia de chave estrangeira. Dados existentes não são afetados, mas não há mais verificação de referencia.

```
Create table teste3 (c1 number(5) primary key);  
Create table teste4 (c2 number(5) constraint fk_teste4_teste3  
references teste3);
```

```
alter table teste4 drop constraint fk_teste4_teste3;
```

- g) Alteração de nome de coluna

```
Alter table EMPREGADO  
rename column mat to matricula;
```

- h) Alteração do tamanho de uma coluna. A redução só será aceita, se os dados já existentes na coluna couberem na redução.

```
Alter table EMPREGADO  
modify nome varchar2(30);
```

Alteração para NULL ou NOT NULL, deve ser feita com a ALTER ... MODIFY

```
Alter table EMPREGADO  
modify nome not null;
```

O inverso também é feito com modify.

```
Alter table EMPREGADO
```

```
modify nome null;
```

- i) Adicionando chave única. Uma tabela pode ter várias chaves únicas, mas somente uma chave primária. As chaves únicas podem ser compostas e chaves estrangeiras podem ser criadas referenciando chaves únicas também. Outra diferença é que uma chave única pode ter valores vazios. Novamente, se existirem dados que violem a chave única, tal chave não poderá ser criada.

```
alter table EMPREGADO  
add constraint uk_nome unique (nome);
```

- j) Remoção de tabelas (a opção cascade constraints permite remover as chaves estrangeiras que referenciam a chave primária da tabela sendo removida – somente as regras de chave estrangeira são removidas, mas não os dados já existentes).

```
Drop table <nome da tabela> [cascade constraints];
```

```
Create table teste5 (c1 number(5) primary key);  
Create table teste6 (c2 number(5) references teste5);  
Drop table teste5; -- sem cascade constraints  
-- não é possível a remoção
```

```
Drop table teste5 cascade constraints; -- remocao ok
```

```
Drop table teste1;  
Drop table teste2;  
Drop table teste3;  
Drop table teste4;  
Drop table teste6;
```

- k) Adicionando um regra de validação  
Exemplo: sexo só pode ser 'M' ou 'F'. Para este exemplo, vamos acionar antes a coluna sexo em EMPREGADO. Novamente, se existirem dados que violem a regra, ela não poderá ser criada.

```
alter table EMPREGADO add sexo char (1);  
alter table EMPREGADO add constraint sexo_emp_ck  
check (sexo in ('M', 'F'));
```

- l) Adicionando um valor default

```
Drop table teste1 cascade constraints;  
Create table teste1 (c1 number(5) primary key,  
                    c2 number(5));  
alter table teste1 modify c1 default 1;
```

- m) Removendo um valor default (nesse caso, o valor default é *null*, que é o padrão)

```
alter table teste1 modify c1 default (null);
```

## II) Comandos de Manipulação de Dados (DML – *Data Manipulation Language*)

São comandos de manipulação de dados:

- Insert (inserção)
- Update (alteração)
- Delete (remoção)

1) Comando insert

```
Insert into DEPART (deptno, nome, local)
values (10, 'Vendas','BH');
```

```
Insert into DEPART
values (20, 'TI','BH');
```

```
select * from DEPART;
```

```
Insert into EMPREGADO
values (1000, 'Antonio',10, 'M');
```

2) Comando Update

```
Update EMPREGADO set matricula=3000
Where matricula=1000;
```

```
Select * from EMPREGADO;
```

3) Comando Delete

```
Delete from EMPREGADO
Where matricula=3000;
```

```
Select * from EMPREGADO;
```

4) Para confirmar as manipulações, emitimos o comando COMMIT;

Então, digite COMMIT e confirme.