

Objetivos da Aula:

-Apresentação:

-***COMANDOS DDL – PARTE 2***

Professora: Juliana Forin Pasquini Martinez
17/02/2020 – semana 03

Roteiro

- Comandos
 - ❖ ALTER
 - ❖ DROP
 - ❖ CASCADE
- Cláusulas DDL equivalentes
- Exercícios

Comando ALTER TABLE

- Uma vez criadas as tabelas, podem surgir a necessidade de alterações em suas estruturas como:
 - *Aumentar o tamanho de uma coluna;*
 - *Acrescentar uma nova coluna;*
 - *Adicionar uma constraint;*

ALTER TABLE: Sintaxe

ALTER TABLE NOME_DA_TABELA

[**ADD** *definição de coluna*,]

[**MODIFY** *definição de coluna*,]

[**DROP COLUMN** *nome*,]

[**RENAME COLUMN** *antigo* **TO** *novos*,]

[**ADD CONSTRAINT** *definição de constraint*,]

[**MODIFY CONSTRAINT** *definição de constraint*,]

[**DROP CONSTRAINT** *nome*,]

[**RENAME CONSTRAINT** *antigo* **TO** *novos*,]

[**ENABLE** | **DISABLE** *constraint*,]

[**RENAME TO** *novos_nome*];

Exemplos do comando ALTER TABLE

ALTER TABLE PEDIDO

ADD VALOR_TOTAL NUMBER(8,2)

ALTER TABLE ENDERECO

MODIFY COD_ENDERECO NUMBER(3)

ALTER TABLE AUTOR **DROP COLUMN** OBSERCACAO

ALTER TABLE CLIENTE **RENAME COLUMN** US_COD **TO** CLI_COD

ALTER TABLE VENDEDOR **ADD CONSTRAINT** PK_VENDEDOR

PRIMARY KEY (VEM_COD)

Adicionando colunas

ALTER TABLE NOME_DA_TABELA

ADD (*definição de coluna*)

Exemplo:

ALTER TABLE PEDIDO ADD PED_DATA DATE

- ❖ A coluna torna-se a última coluna da tabela
- ❖ Se já existirem registros na tabela, então a coluna será NULA para todos os registros da tabela

Modificando colunas

ALTER TABLE NOME_DA_TABELA
MODIFY *[definição de coluna]*

Exemplo:

ALTER TABLE PEDIDO MODIFY PED_COD NUMBER(8)

Apenas é possível para tipos equivalentes:

- Exemplo:
- number(4) -> number(8)
- varchar2(100) -> varchar2(200)
- ***Demais modificações só são permitidas se a coluna não tiver nenhum dado inserido***

Excluindo colunas

```
ALTER TABLE NOME_DA_TABELA  
DROP COLUMN NOME_COLUNA
```

Exemplo:

```
ALTER TABLE PEDIDO DROP COLUMN PED_DATA
```

- O comando DROP é utilizado para excluir objetos do esquema do banco de dados.
- Não são permitidas exclusões de colunas com alguma *constraint* associada! É preciso excluir a restrição antes.

Renomeando colunas

```
ALTER TABLE NOME_DA_TABELA  
RENAME COLUMN NOME_ANTIGO TO NOME_NOVO
```

Exemplo:

```
ALTER TABLE PEDIDO  
RENAME COLUMN PED_DATA TO PED_DATA_CADASTRO
```

Incluindo constraints

SINTAXE:

ALTER TABLE NOME_DA_TABELA

ADD CONSTRAINT NOME_RESTRICAO **PRIMARY KEY** (COLUNA_PK)

ALTER TABLE NOME_DA_TABELA **ADD CONSTRAINT** NOME_RESTRICAO
FOREIGN KEY (COLUNA_FK)

REFERENCES TABELA_REFERIDA (COLUNA_PK)

ALTER TABLE NOME_DA_TABELA **ADD CONSTRAINT** NOME_RESTRICAO
UNIQUE (NOME_COLUNA)

Exemplos

Primary Key:

```
ALTER TABLE PEDIDO ADD CONSTRAINT PK_PEDIDO  
PRIMARY KEY (PED_COD)
```

Foreign Key:

```
ALTER TABLE PEDIDO ADD CONSTRAINT  
FK_CLI_COD FOREIGN KEY (CLI_COD)  
REFERENCES CLIENTE (CLI_COD)
```

Unique Key:

```
ALTER TABLE CLIENTE  
ADD CONSTRAINT UK_CLI_CPF UNIQUE (CLI_CPF)
```

Excluindo *constraints*

```
ALTER TABLE NOME_DA_TABELA  
DROP CONSTRAINT NOME_RESTRICAO [CASCADE]
```

- Comando válido para qualquer tipo de restrição
- A cláusula CASCADE elimina as *constraints* dependentes.

Exemplo:

```
ALTER TABLE FUNCIONARIO DROP CONSTRAINT fu_cod_pk cascade
```

Exemplo

```
create table funcionario
```

```
(fun_cod number (4) constraint fu_cod_pk primary key,
```

```
fun_nome varchar2(30));
```

```
create table dependente
```

```
(dep_cod number(4) constraint dep_cod_pk primary key, dep_nome  
varchar2(40),
```

```
dep_fun_cod number constraint dep_fun_cod_fk references  
funcionario(fun_cod));
```

Exemplo

```
Select CONSTRAINT_NAME, CONSTRAINT_TYPE, STATUS,  
SEARCH_CONDITION  
From USER_CONSTRAINTS  
Where TABLE_NAME = 'FUNCIONARIO';
```

Results Explain Describe Saved SQL History

CONSTRAINT_NAME	CONSTRAINT_TYPE	STATUS	SEARCH_CONDITION
FU_COD_PK	P	ENABLED	-

1 rows returned in 0.00 seconds [Download](#)

```
Select CONSTRAINT_NAME, CONSTRAINT_TYPE, STATUS,  
SEARCH_CONDITION  
From USER_CONSTRAINTS  
Where TABLE_NAME = 'DEPENDENTE';
```

CONSTRAINT_NAME	CONSTRAINT_TYPE	STATUS	SEARCH_CONDITION
DEP_COD_PK	P	ENABLED	-
DEP_FUN_COD_FK	R	ENABLED	-

2 rows returned in 0.00 seconds [Download](#)



- Realizar o comando e verificar em
USER_CONSTRAINTS

- **ALTER TABLE** FUNCIONARIO **DROP**
CONSTRAINT fu_cod_pk CASCADE

Modificando constraints

```
ALTER TABLE NOME_DA_TABELA  
  RENAME CONSTRAINT NOME_ANTIGO TO NOME_NOVO
```

Comando MODIFY? Uma constraint não pode ser modificada, porém, para adicionar uma restrição NOT NULL, pode ser utilizando a cláusula MODIFY.

Exemplos:

```
ALTER TABLE PROJETO RENAME CONSTRAINT  
PK_PROJETO_COD_PROJ TO PK_PROJETO
```

```
ALTER TABLE CLIENTE MODIFY CLI_NOME NOT NULL
```


Ativando/Desativando constraints

```
ALTER TABLE NOME_DA_TABELA [ENABLE | DISABLE]  
CONSTRAINT NOME_CONSTRAINT [CASCADE]
```

- **ENABLE:** habilita uma *constraint*
- **DISABLE:** desabilita uma *constraint*
- **CASCADE:** essa cláusula aplica a mesma regra a todas as *constraints dependentes*.

Exemplo:

```
ALTER TABLE CLIENTE  
DISABLE CONSTRAINT PK_CLI_COD CASCADE
```

Renomeando e excluindo tabelas

- Renomeando uma tabela:

ALTER TABLE NOME_DA_TABELA

RENAME TO NOVO_NOME_TABELA

Exemplo: **ALTER TABLE** CLIENTE **RENAME TO** CLI

- Excluindo uma tabela:

DROP TABLE NOME_DA_TABELA

[CASCADE CONSTRAINTS]

Exemplo: **DROP TABLE** CLIENTE

Comando DELETE CASCADE

➤ Exemplo:

```
ALTER TABLE FUNCIONARIO  
ADD CONSTRAINT FK_COD_DEPARTAMENTO  
FOREIGN KEY(COD_DEPARTAMENTO)  
REFERENCES DEPARTAMENTO(COD_DEPARTAMENTO)  
ON DELETE CASCADE
```

Exclui automaticamente os registros da tabela FUNCIONARIO que fazem referência a um registro excluído da tabela DEPARTAMENTO (PK), ou seja:

DEPARTAMENTO.COD_DEPARTAMENTO= FUNCIONARIO.COD_DEPARTAMENTO

DEPARTAMENTO

COD_DEPARTAMENTO	DEP_DESCRICAO
<u>1</u>	RH
<u>2</u>	VENDAS

FUNÇÃO

FUN_COD	FUN_NOME	COD_DEPARTAMENTO
101	MARIA	<u>1</u>
102	JOÃO	<u>1</u>
103	PEDRO	<u>1</u>

Resumo

- Comandos DDL estudados:
 - CREATE TABLE
 - Tipos de dados, constraints, default, not null, etc
 - ALTER TABLE
 - Incluir colunas / *constraints*
 - Alterar colunas
 - Excluir colunas / *constraints* / *tabelas*
 - Renomear colunas / *constraints* / *tabelas*
 - Ativar/desativar *constraints*
 - Cascade

Cláusulas equivalentes

- Algumas cláusulas da linguagem SQL podem ser escritas de diversas formas.
- Deve ser analisada a melhor forma de se escrever uma cláusula, para isso:
 - Considerar simplicidade
 - Facilidade de entendimento por outros analistas

Criando uma UNIQUE KEY: forma unificada

```
CREATE TABLE ALUNO  
(  
  COD_ALUNO number(4) NOT NULL,  
  NOME varchar2(100) NOT NULL,  
  CPF varchar2(20),  
  COD_CURSO number(4) NOT NULL,  
  CONSTRAINT UK_CPF UNIQUE(CPF)  
);
```

Criando uma UNIQUE KEY: forma desmembrada

```
CREATE TABLE ALUNO
```

```
(  
  COD_ALUNO number(4) NOT NULL,  
  NOME varchar2(100) NOT NULL,  
  CPF varchar2(20),  
  COD_CURSO number(4) NOT NULL  
);
```

```
ALTER TABLE ALUNO ADD CONSTRAINT UK_CPF UNIQUE(CPF);
```


Criando uma PRIMARY KEY: forma unificada

```
CREATE TABLE ALUNO
```

```
(
```

```
COD_ALUNO number(4) CONSTRAINT PK_ALUNO PRIMARY KEY,
```

```
NOME varchar2(100) NOT NULL,
```

```
CPF varchar2(20),
```

```
COD_CURSO number(4) NOT NULL
```

```
);
```

Criando uma PRIMARY KEY: forma desmembrada

```
CREATE TABLE ALUNO
```

```
(  
  COD_ALUNO number(4),  
  NOME varchar2(100) NOT NULL,  
  CPF varchar2(20),  
  COD_CURSO number(4) NOT NULL  
);
```

```
ALTER TABLE ALUNO ADD CONSTRAINT PK_ALUNO  
PRIMARY KEY(COD_ALUNO);
```

Criando uma FOREIGN KEY: forma unificada

```
CREATE TABLE ALUNO
```

```
(  
  COD_ALUNO number(4) NOT NULL,  
  NOME varchar2(100) NOT NULL,  
  CPF varchar2(20),  
  COD_CURSO number(4) NOT NULL,  
  CONSTRAINT FK_COD_CURSO FOREIGN KEY(COD_CURSO)  
  REFERENCES CURSO(COD_CURSO)  
);
```

Criando uma FOREIGN KEY: forma desmembrada

```
CREATE TABLE ALUNO
```

```
(  
  COD_ALUNO number(4) NOT NULL,  
  NOME varchar2(100) NOT NULL,  
  CPF varchar2(20),  
  COD_CURSO number(4) NOT NULL  
);
```

```
ALTER TABLE ALUNO ADD CONSTRAINT FK_COD_CURSO  
  FOREIGN KEY (COD_CURSO) REFERENCES  
  CURSO(COD_CURSO)
```

Tabela completa: forma unificada

```
CREATE TABLE ALUNO
(
  COD_ALUNO number(4),
  NOME varchar2(100) NOT NULL,
  CPF varchar2(20),
  COD_CURSO number(4) NOT NULL,
  CONSTRAINT UK_CPF UNIQUE(CPF),
  CONSTRAINT PK_ALUNO PRIMARY KEY(COD_ALUNO),
  CONSTRAINT FK_COD_CURSO FOREIGN KEY(COD_CURSO)
  REFERENCES CURSO(COD_CURSO)
);
```

Tabela completa: forma desmembrada

```
CREATE TABLE ALUNO
```

```
(
```

```
COD_ALUNO number(4) NOT NULL,
```

```
NOME varchar2(100) NOT NULL,
```

```
CPF varchar2(20),
```

```
COD_CURSO number(4) NOT NULL
```

```
);
```

```
ALTER TABLE ALUNO ADD CONSTRAINT PK_ALUNO
```

```
PRIMARY KEY (COD_ALUNO);
```

```
ALTER TABLE ALUNO ADD CONSTRAINT FK_COD_CURSO
```

```
FOREIGN KEY (COD_CURSO) REFERENCES CURSO(COD_CURSO);
```

```
ALTER TABLE ALUNO ADD CONSTRAINT UK_CPF UNIQUE (CPF);
```

Para refletir!!!

- Considerando as cláusulas equivalentes apresentadas na aula de hoje, qual a melhor forma de se escrever um comando DDL?
- Cláusulas unificadas? ou
- Cláusulas desmembradas?

Qual tipo de cláusula você usaria?

