



Aula 12

Prof: Henrique Augusto Maltauro

Implementar Banco de Dados Para WEB

SQL

- DQL: ALTER

O comando ALTER é responsável por alterar os elementos da estrutura do nosso banco de dados.

SQL

- DQL: ALTER (TABLE)

O comando ALTER em conjunto com o subcomando TABLE vai alterar uma tabela na estrutura do banco de dados.

Logo após o ALTER TABLE é necessário informar o nome da tabela que será alterada.

SQL

- DQL: ALTER (TABLE)

Depois de informado o nome da tabela, é informado as definições das alterações que serão realizadas.

É realizado apenas uma alteração por vez.

SQL

- **DQL: ALTER (TABLE)**

Com o **ALTER TABLE** nós podemos:

- Adicionar uma coluna
- Remover uma coluna
- Renomear uma coluna
- Alterar o tipo de um coluna
- Adicionar uma constraint
- Remover uma constraint

SQL

- DQL: ALTER (TABLE: ADD)

O comando `ADD` vai adicionar uma coluna nova na tabela.

SQL

- DQL: ALTER (TABLE: ADD)

```
ALTER TABLE NOME_DA_TABELA  
ADD NOME_DA_COLUNA TIPO_DO_DADO
```

SQL

- **DQL: ALTER (TABLE: DROP COLUMN)**

O comando **DROP COLUMN** vai remover uma coluna existente da tabela.

SQL

- DQL: ALTER (TABLE: DROP COLUMN)

```
ALTER TABLE NOME_DA_TABELA  
DROP COLUMN NOME_DA_COLUNA
```

SQL

- DQL: ALTER (TABLE: RENAME COLUMN)

O comando RENAME COLUMN vai renomear uma coluna existente da tabela.

SQL

- DQL: ALTER (TABLE: RENAME COLUMN)

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

SQL

- DQL: ALTER (TABLE: ALTER COLUMN)

O comando `ALTER COLUMN` vai alterar uma coluna existente da tabela, permitindo alterar o tipo do dado, ou alguma constraint não nomeável.

SQL

- DQL: ALTER (TABLE: ALTER COLUMN)

ALTER TABLE *NOME_DA_TABELA*

ALTER COLUMN *NOME_DA_COLUNA* *TIPO_DO_DADO*

SQL

- DQL: ALTER (TABLE: ALTER COLUMN)

ALTER TABLE *NOME_DA_TABELA*

ALTER COLUMN *NOME_DA_COLUNA* TIPO_DO_DADO NOT NULL

SQL

- DQL: ALTER (TABLE: ADD CONSTRAINT)

O comando ADD CONSTRAINT vai adicionar uma **constraint** na tabela.

SQL

- DQL: ALTER (TABLE: ADD CONSTRAINT)

```
ALTER TABLE NOME_DA_TABELA  
ADD CONSTRAINT UC_TABELA UNIQUE (NOME_DA_COLUNA)
```


SQL

- DQL: ALTER (TABLE: ADD CONSTRAINT)

```
ALTER TABLE NOME_DA_TABELA  
ADD CONSTRAINT PK_TABELA PRIMARY KEY (NOME_DA_COLUNA)
```

SQL

- DQL: ALTER (TABLE: ADD CONSTRAINT)

```
ALTER TABLE NOME_DA_TABELA  
ADD CONSTRAINT FK_TABELA FOREIGN KEY (NOME_DA_COLUNA)  
REFERENCES TABELA_DE_REFERENCIA (NOME_DA_COLUNA)
```

SQL

- DQL: ALTER (TABLE: ADD CONSTRAINT)

```
ALTER TABLE NOME_DA_TABELA  
ADD CONSTRAINT CHK_TABELA CHECK (NOME_DA_COLUNA >= 18)
```

SQL

- DQL: ALTER (TABLE: ADD DEFAULT)

A **constraint** de **DEFAULT** é a única que funciona diferente, é preciso do comando **ADD DEFAULT** para adicioná-lo de uma coluna já existente.

SQL

- DQL: ALTER (TABLE: ADD DEFAULT)

```
ALTER TABLE NOME_DA_TABELA  
ADD DEFAULT 'string' FOR NOME_DA_COLUNA
```

SQL

- DQL: ALTER (TABLE: DROP CONSTRAINT)

O comando DROP CONSTRAINT vai remover uma **constraint** na tabela.

SQL

- DQL: ALTER (TABLE: DROP CONSTRAINT)

```
ALTER TABLE NOME_DA_TABELA  
DROP CONSTRAINT NOME_DA_CONSTRAINT
```

SQL

- DQL: ALTER (TABLE: DROP DEFAULT)

A **constraint** de **DEFAULT** é a única que funciona diferente, é preciso do comando **DROP DEFAULT** para removê-lo de uma coluna já existente.

SQL

- DQL: ALTER (TABLE: DROP DEFAULT)

ALTER TABLE *NOME_DA_TABELA*

ALTER COLUMN *NOME_DA_COLUNA* DROP DEFAULT

SQL

- DQL: DROP

O comando DROP é responsável por remover os elementos da estrutura do nosso banco de dados.

SQL

- DQL: DROP (TABLE)

O comando DROP em conjunto com o subcomando TABLE vai remover uma tabela na estrutura do banco de dados.

SQL

- DQL: DROP (TABLE)

DROP TABLE *NOME_DA_TABELA*

Exercício

Exercício

gg.gg/SenacBD12

github.com/hmaltaurodev/slides