BANCO DE DADOS

DDL (DATA DEFINITION LANGUAGE)

PROFº Bruno A de Moraes

bmoraes@sp.senai.br

SENAI

DDL (DATA DEFINITION LANGUAGE)

Linguagem de Definição de Dados

Permite a definição da estrutura do Banco de Dados. A seguir alguns comandos:

CREATE
ALTER
DROP

CRIAÇÃO DE BANCO DE DADOS

CREATE DATABASE

- Cria um novo Banco de Dados
- Sintaxe pode mudar a depender do SGBD
- Deve-se escolher o BD master para executar o comando

Sintaxe: **CREATE DATABASE [IF NOT EXISTS]** **<nome_do_BD>[;]**

Exemplo: **CREATE DATABASE** **db_Solar;**

SENAI

REMOÇÃO DE BANCO DE DADOS

DROP DATABASE

- Remove um Banco de Dados

Sintaxe: **DROP DATABASE [IF NOT EXISTS]** **<nome_do_BD>[;]**

Exemplo: **DROP DATABASE db_Solar;**

Obs: O Banco de Dados só será removido se não estiver aberto ou em uso

SENAI

DDL (DATA DEFINITION LANGUAGE)

Instruções para definição do esquema do BD:

CREATE TABLE:

- Cria uma nova tabela na base de dados, especificando nome, atributos e restrições.

ALTER TABLE:

- Altera as definições de uma tabela

DROP TABLE

- Remove uma tabela, quando não é mais necessária

SENAI

TIPO DE DADOS

São utilizados para facilitar o armazenamento e representação dos dados armazenados.

Para cada campo de uma tabela, um tipo de dados apropriado é utilizado.

TIPO DE DADOS (NÚMERICO)

Todos os tipos numéricos possíveis, o que inclui exatos, aproximados, inteiros, etc.

Tipo de Dado	Tamanho
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(X)	4 ou 8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes
DOUBLE PRECISION	8 bytes
REAL	8 bytes
DECIMAL(M,D)	Variável
NUMERIC(M,D)	Variável

SENAI

TIPO DE DADOS (DATA E HORA)

Esses são os tipos possíveis para armazenar dados relacionados a data e hora.

Tipo de Dado	Tamanho
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte

TIPO DE DADOS

Tipos de dados	Descrição
char(n)	Sequência de caracteres de tamanho fixo. Máximo 8000 caracteres
varchar(n)	Sequência de caracteres de tamanho variável. Máximo 8000 caracteres
varchar(max)	Sequência de caracteres de tamanho variável. Máximo 1.073.741.824 caracteres
text	Sequência de caracteres de tamanho variável. Máximo 2GB de dados de texto

TIPO DE DADOS

nchar(n)	Dados Unicode de comprimento fixo. Máximo 4000 caracteres
nvarchar(n)	Dados Unicode de comprimento variável. Máximo 4000 caracteres
nvarchar(max)	Dados Unicode de comprimento variável. Máximo 536.870.912 caracteres
ntext	Dados Unicode de comprimento variável. Máximo 2GB de dados de texto

DIFERENÇA ENTRE CHAR E VARCHAR

CHAR e VARCHAR são tipos de dados caractere, a diferença é que **CHAR** é um tipo de dado de comprimento fixo e **VARCHAR** é de comprimento variável.

Usamos **CHAR** quando os tamanhos que desejamos armazenar na coluna de uma tabela são de tamanho consistentes e semelhantes. *Exemplo: Número de telefone, CEP, CPF,* CGC, etc. O tipo CHAR possui um tamanho fixo, assim se você tentar armazenar um valor maior que o definido numa coluna do tipo CHAR ele será truncado.

SENAI

DIFERENÇA ENTRE CHAR E VARCHAR

Se você definir uma coluna da tabela (*campo*) como **CHAR(10)** e armazenar um caractere apenas ele vai armazenar mais nove espaços em branco.(*Por causa desta característica o tipo de dados CHAR é chamado de tipo de dados com tamanho fixo.)*

Use **VARCHAR** quando os tamanhos a serem armazenados na coluna da tabela variam consideravelmente. *Ex: Endereço,Nomes, URL, etc.* Dessa forma um valor menor irá ocupar menos espaço que um valor maior.

SENAI

DIFERENÇA ENTRE CHAR E VARCHAR

O tipo de dados **VARCHAR** armazena somente a quantidade de caracteres que foram definidos na usa criação.

Assim se você definir uma coluna da tabela (*campo*) como **VARCHAR(10)** e armazenar um caractere ele vai armazenar somente o caractere sem colocar espaços para completar o tamanho definido na criação.

SENAI

QUANDO USAR CHAR E VARCHAR?

O tipo CHAR deve ser usado quando sabemos que todos os dados armazenados em determinada coluna não são variáveis como, por exemplo, uma coluna que armazena a sigla do estado ou o cep que sempre terão o mesmo tamanho.

Já o VARCHAR deve ser utilizado quando não sabemos o que vamos armazenar. Um exemplo pode ser o nome do cliente, endereço, o email que sempre variam de tamanho.

SENAI

CRIAÇÃO DE TABELAS

CREATE TABLE

- Colunas são especificadas primeira, sob a forma:

  <nomeCol> <dominio> <restrição>

- Depois Chaves, integridade referencial e restrições de integridade

Sintaxe:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] <nome_da_tabela>(

<nome_da_col1> <tipo_da_col1> [NULL/NOT NULL],

<nome_da_col2> <tipo_da_col2> [NULL/NOT NULL],

...

PRIMARY KEY <lista_de_nomes_de_col>,

FOREIGN KEY <nomes_de_col> REFERENCES

<nome_tab_ref>(<nome_da_col_ref>)

)[;]

SENAI

CRIAÇÃO DE TABELAS

CREATE TABLE

As partes da declaração que se encontram entre [ ] (colchetes) são opcionais:

TEMPORARY: indica que a tabela criada será temporária, o que significa que ela expira

assim que a sua sessão no MySQL terminar. Use-a sempre que estiver fazendo testes.

IF NOT EXISTS: verifica a prévia existência da tabela e evita uma interrupção do script,

causada por erro.

SENAI

CRIAÇÃO DE TABELAS

- **CREATE TABLE**

```
CREATE TABLE Fornecedor(
Fcodigo INT NOT NULL,
Fnome VARCHAR(30) NOT NULL,
Status INT,
Cidade VARCHAR (30)
);
```

SQL CONSTRAINTS (RESTRIÇÕES)

NOT NULL

A constraint **NOT NULL** impõe a uma coluna a NÃO aceitar valores **NULL**.

- Constraint **NOT NULL** obriga um campo a sempre possuir um valor.
- Não é possível inserir um registro (ou atualizar) sem entrar com um valor neste campo.

SENAI

SQL CONSTRAINTS (RESTRIÇÕES)

DEFAULT

Restrição estabelecida para a coluna, onde o sistema assume que ele dever ser utilizado quando o usuário deixa de preencher o campo com o valor desejado.

O valor padrão será adicionado a todos os novos registros caso nenhum outro valor seja especificado na hora de inserir dados.

SENAI

CRIAÇÃO DE TABELAS

Especificação de chaves:

Primária

Primary Key(<nomeCol>),

Estrangeira

FOREIGN KEY(<nomeCol>) REFERENCES <NomeTabRefer> ,

Alternativa

UNIQUE(<nomeCol>),

CRIAÇÃO DE TABELAS

- **Exemplo com chave primária:**

```
CREATE TABLE Departamento
(
Dcodigo INT NOT NULL,
Dnome VARCHAR(30) NOT NULL,
Cidade VARCHAR(30),
Primary Key (Dcod)
);
```

CRIAÇÃO DE TABELAS

Exemplo com chave primária composta:

```
CREATE TABLE Empregado
(
Ecod INT NOT NULL,
Enome VARCHAR(40) NOT NULL,
CPF VARCHAR(15) NOT NULL,
Salario DECIMAL(7,2),
Dcodigo INT NOT NULL,
PRIMARY KEY(Ecodigo,Enome)
);
```

SENAI

CRIAÇÃO DE TABELAS

Exemplo com chave estrangeira (completo):

```
CREATE TABLE Empregado
(
Ecod INT NOT NULL,
Enome VARCHAR(40) NOT NULL,
CPF VARCHAR(15) NOT NULL,
Salario DECIMAL(7,2),
Cod_Dept INT NOT NULL,
PRIMARY KEY(Ecodigo),
[CONSTRAINT fk_DCodigo]FOREIGN KEY (Dcodigo) REFERENCES Departamento(Dcodigo)
);
```

CRIAÇÃO DE TABELAS

- **Exemplo com chave estrangeira (simplificado):**

```
CREATE TABLE Empregado
(
Ecod INT NOT NULL,
Enome VARCHAR(40) NOT NULL,
CPF VARCHAR(15) NOT NULL,
Salario DECIMAL(7,2),
Dcodigo INT NOT NULL,
PRIMARY KEY(Ecodigo),
FOREIGN KEY (Dcodigo) REFERENCES Departamento
);
```

RESTRIÇÃO DE INTEGRIDADE

Chave Estrangeira (FOREIGN KEY)

Uma restrição de chave estrangeira só pode ser criada se as tabelas a que se referem já foram anteriormente criadas.

Um índice não é criado automaticamente para chaves estrangeiras, embora seja recomendável criá-lo para maior desempenho.

RESTRIÇÃO DE INTEGRIDADE

Chave Estrangeira (FOREIGN KEY)

Por padrão a coluna da tabela em references tomada como chave estrangeira é a chave primária.

Caso deseje uma outra coluna, desde que seja UNIQUE, fosse chave estrangeira, basta especificar em references o nome da coluna, por exemplo:

FOREIGN KEY(Ccliente) **REFERENCES** Cliente (Ccodigo)

RESTRIÇÃO

Chave Alternativa (UNIQUE)

A restrição UNIQUE identifica de forma única cada registro em uma tabela de um banco de dados.

As constraints UNIQUE e PRIMARY KEY garantem a unicidade em uma coluna ou conjunto de colunas.

RESTRIÇÃO

Chave Alternativa (UNIQUE)

•Uma constraint **PRIMARY KEY** automaticamente possui uma restrição **UNIQUE** definida, portanto não é necessário especificar essa constraint neste caso.

•É possível termos várias constraints **UNIQUE** em uma mesma tabela, mas apenas uma **PRIMARY KEY** por tabela.

SENAI

REMOÇÃO DE TABELAS

- **DROP TABLE**
  - Elimina completamente a tabela (vazia ou não)

  DROP TABLE <nome_da_tabela>;

  Exemplo:

  DROP TABLE Empregado;

  Obs: Não há como recuperar a tabela removida.

ALTERAÇÃO DE TABELAS

- **ALTER TABLE**
  - Altera as propriedades da tabela selecionada

ALTER TABLE <nome_da_tabela> <alteração>[;]

ALTERAÇÃO PODE SER ADD OU DROP.

Exemplo:
ALTER TABLE Funcionario
ADD sexo **CHAR(1)**;

ALTER TABLE Funcionario
DROP COLUMN sexo;

SENAI

ALTERAÇÃO DE TABELAS

- Alteração do nome de uma tabela

Permite alterar o nome de uma tabela após a sua criação.

ALTER TABLE tabela
RENAME TO nome;

Exemplo:

ALTER TABLE empregado
RENAME TO funcionario;

ALTERAÇÃO DE TABELAS

- Alteração do nome de um campo da tabela

Permite alterar o nome de um campo de uma tabela.

ALTER TABLE tabela
CHANGE campo nome tipo_dado;

Exemplo:
ALTER TABLE funcionario
CHANGE CIC CPF int;

ALTERAÇÃO DE TABELAS

- Alteração do tipo dados de colunas

ALTER TABLE tabela

MODIFY COLUMN campo tipo_dado;

Exemplo:

Alterar o tipo de dados da coluna nome da tabela funcionario para varchar(30):

ALTER TABLE funcionario

MODIFY COLUMN nome **varchar(30)**;

ALTERAÇÃO DE TABELAS

- **Adicionando valores default a colunas**

ALTER TABLE nome_tabela
MODIFY COLUMN coluna tipo_dados
DEFAULT ‘valor_padrao’;

Exemplo:

Adicionar um default a coluna uf da tabela cliente

ALTER TABLE funcionario
MODIFY COLUMN uf char(2)
DEFAULT ‘SP’;

ALTERAÇÃO DE TABELAS

- Adicionar chave Primária
  - Pode ser feita através do comando abaixo:

ALTER TABLE tabela
ADD **PRIMARY KEY**(campo);

ALTERAÇÃO DE TABELAS

- Exclusão de uma chave Primária

  Pode ser feita através do comando abaixo:

ALTER TABLE tabela

DROP **PRIMARY KEY**;

ALTERAÇÃO DE TABELAS

- Exclusão de uma chave estrangeira
  - Pode ser feita somente se ela recebeu um nome

ALTER TABLE tabela

DROP FOREIGN KEY chave_estrangeira