



PELO FUTURO DO TRABALHO

01 - Criação e manipulação de tabelas

Professor Hermano Roepke

Contextualização

Uma **tabela** é uma unidade básica de armazenamento do SGBD que nos permite **organizar os dados** armazenados em uma estrutura bidimensional.

As tabelas são **compostas por colunas e linhas** (razão pela qual as denominamos bidimensionais). Em uma tabela, cada linha contém um conjunto de colunas ou, se preferir, campos.

Agora que você já conhece o conceito de tabela, podemos seguir para a parte prática de criação e manipulação de tabelas. Antes, porém, veja algumas sugestões para convencionarmos o nome das tabelas e das colunas.

Convenções para nomeação

Seguem algumas recomendações para nomeação de tabelas e colunas:

- comece sempre com uma letra;
- utilize no máximo 30 caracteres;
- de preferência os caracteres A-Z, a-z, 0-9 e _;
- não duplique o nome de outro objeto de propriedade do mesmo usuário;
- não use uma palavra reservada para o SGBD utilizado.

Convenções para nomeação

É interessante que as tabelas e outros objetos do banco de dados possuam nomes descritivos. Também é interessante manter a compatibilidade entre os nomes de colunas em tabelas diferentes.

Por exemplo, a coluna do número do departamento é chamada de ID_DEPARTAMENTO nas tabelas EMPREGADO e DEPARTAMENTO.

Instrução CREATE TABLE

A instrução utilizada para criar tabelas é a CREATE TABLE, que é uma das instruções DDL (Data Definition Language). Para utilizá-la é necessário ter privilégio CREATE TABLE e uma área disponível para armazenamento.

Para a criação de uma tabela é necessário que você informe, basicamente:

- nome da tabela;
- nome da coluna, tipo de dados da coluna e o tamanho, quando necessário.

A sintaxe do comando é:

```
CREATE TABLE nome_tabela (  
  coluna1 tipo_do_dado,  
  coluna2 tipo_do_dado,  
  coluna3 tipo_do_dado, ...);
```

Instrução CREATE TABLE

Ao criarmos colunas é necessário informar o **tipo de dados**. O tipo de dados e os argumentos variam de acordo com o SGBD adotado, tornando necessário verificar a documentação para utilizar os tipos corretamente. Segue uma tabela com os tipos de dados mais comuns e a sua representação nos SGBDS já comentados na unidade curricular.

	<u>FIREBIRD</u>	<u>MYSQL</u>	<u>ORACLE</u>	<u>POSTGRESQL</u>
Numérico inteiro	integer	int	integer	integer
Numérico flutuante	numeric	decimal	number	decimal
Variável de caracteres	varchar	varchar	varchar2	character varying
Data	date	date	date	date
Caractere fixo	char	char	char	char
Tempo	timestamp	timestamp	timestamp	timestamp

Opção DEFAULT

Você pode especificar um valor default para uma coluna sempre que efetuar uma inserção. É importante saber que somente no momento em que você omitir um valor para a coluna o SGBD irá utilizar esse valor definido previamente. O valor da opção DEFAULT deve corresponder ao tipo de dados da coluna.

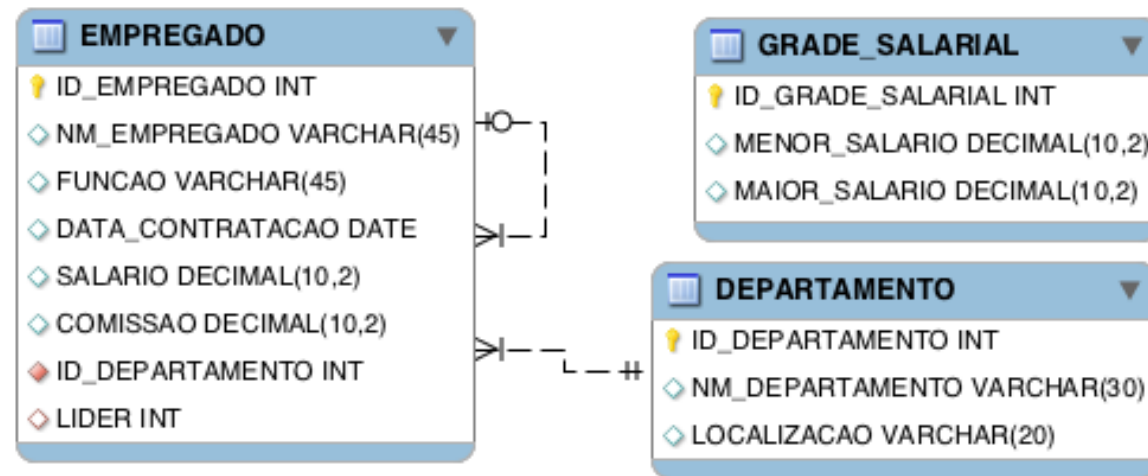
- **Valores permitidos:** um valor literal, expressão ou função SQL.
- **Valores não permitidos:** outro nome da coluna.

Veja um exemplo:

```
CREATE TABLE pessoa (  
  cd_pessoa integer,  
  nm_pessoa varchar(255),  
  fl_estado_civil char(1) DEFAULT 'S');
```


Exemplo de uso

Agora você já conheceu as instruções básicas para a criação de uma tabela chegou a hora de praticar. Escolha um dos SGBDs apresentados na disciplina e construa os comandos para a definição das tabelas ilustradas no seguinte diagrama:



Na sequência, execute o comando DESC ou DESCRIBE, que permitirá conhecer a estrutura de uma tabela. Esses comandos não são ANSI, portanto não são compatíveis com todos os SGBDs.

Exemplo de uso

A criação da tabela no SGBD MySQL seria realizada da seguinte maneira:

```
CREATE TABLE DEPARTAMENTO(  
ID_DEPARTAMENTO INT NOT NULL,  
NM_DEPARTAMENTO VARCHAR(30),  
LOCALIZACAO VARCHAR(20));
```

Você pode confirmar a criação da tabela no SGBD MySQL através do comando DESC ou DESCRIBE;

```
DESC DEPARTAMENTO;
```

O resultado apresentado será:

Field	Type	Null ^	Key	Default	Extra
ID_DEPARTAMENTO	int	NO	PRI	NULL	
NM_DEPARTAMENTO	varchar(30)	YES		NULL	
LOCALIZACAO	varchar(20)	YES		NULL	

Criando tabelas com base em subconsulta

Há uma segunda forma de criar tabelas, que se baseia em uma subconsulta. Para isso você precisa adicionar ao comando CREATE TABLE, a cláusula AS e a subconsulta na sequência. Veja a sintaxe:

```
CREATE TABLE table_name  
  (column_name, column_name, ...)  
AS  
  subconsulta
```

Fique atento:

- não esqueça de definir os nomes das colunas e valores default, se necessário;
- verifique se o número de colunas informadas é igual ao número de colunas da subconsulta.

Criando tabelas com base em subconsulta

Para que você tenha conhecimento deste recurso, criamos uma tabela (exemplo) como base em subconsulta. Veja:

```
CREATE TABLE departamento30  
AS  
  SELECT id_empregado, nm_empregado, salario,  
         data_contratacao  
  FROM empregado  
 WHERE id_departamento = 30;
```

No exemplo, foi criada uma tabela chamada DEPARTAMENTO30. Nela há informações de funcionários que pertencem ao departamento de número 30 execute o comando CREATE TABLE acima e confira como ficará a estrutura:

Field	Type	Null	Key	Default	Extra
id_empregado	int	NO		NULL	
nm_empregado	varchar(45)	YES		NULL	
salario	decimal(10,2)	YES		NULL	
data_contratacao	date	YES		NULL	

Instrução **ALTER TABLE**

Use a instrução **ALTER TABLE** para:

- adicionar ou remover uma coluna;
- modificar uma coluna existente;
- definir um valor default para nova coluna.

Instrução ALTER TABLE – Adicionando coluna

DEPARTAMENTO30

ID_EMPREGADO	NM_EMPREGADO	SALARIO_ANUAL	DATA_CONTRATACAO
7698	BLAKE	34200	01-MAY-81
7654	MARTIN	15000	28-SEP-81
7499	ALLEN	19200	20-FEB-81
7844	TURNER	18000	08-SEP-81
...			

“... adicione uma nova coluna na tabela departamento30”

DEPARTAMENTO30

ID_EMPREGADO	NM_EMPREGADO	SALARIO_ANUAL	DATA_CONTRATACAO	TAREFA
7698	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	
...				



Instrução ALTER TABLE – Adicionando coluna

Você pode adicionar colunas após a criação de uma tabela utilizando a instrução ALTER TABLE com a cláusula ADD.

```
ALTER TABLE nome_tabela  
ADD coluna tipo_de_dado;
```

Aproveite a tabela DEPARTAMENTO30 e execute o comando para a adição de uma nova coluna chamada TAREFA na tabela DEPARTAMENTO30. Essa coluna deve permitir o armazenamento da tarefa do funcionário em no máximo 9 caracteres. Veja como ficaria o comando:

```
ALTER TABLE departamento30  
ADD tarefa varchar(9);
```

Instrução ALTER TABLE – Modificando coluna

Você pode modificar colunas após a criação de uma tabela utilizando a instrução ALTER TABLE com a cláusula MODIFY.

```
ALTER TABLE nome_tabela  
MODIFY coluna tipo_de_dado;
```

Agora que você possui a coluna TAREFA na tabela DEPARTAMENTO30, aumente seu tamanho para 15 caracteres utilizando a cláusula MODIFY.

```
ALTER TABLE departamento30  
MODIFY TAREFA varchar(15);
```

Obs.: Há SGBDS onde a palavra reservada “modify” deve ser substituída por “alter”.

Instrução ALTER TABLE – Removendo coluna

Caso você identifique que uma coluna não é mais necessária, é possível removê-la utilizando o comando ALTER TABLE com a cláusula DROP COLUMN. Veja a sintaxe:

```
ALTER TABLE nome_tabela  
DROP COLUMN coluna;
```

Para exercitar a funcionalidade, remova a coluna JOB da tabela DEPT30 com o seguinte comando:

```
ALTER TABLE departamento30  
DROP COLUMN tarefa;
```

Eliminando uma tabela

Você pode remover uma tabela caso não necessite mais dela. Para executar esta ação, utilize o comando DROP TABLE. Conheça a sintaxe do comando:

```
DROP TABLE departamento30;
```

Como não necessitamos mais da tabela DEPARTAMENTO30 que foi criada. Você poderá removê-la utilizando o comando DROP TABLE.

ATENÇÃO! Ao executar o comando DROP TABLE, todos os dados e estruturas de dados serão perdidos...

Além disso, não será possível reverter a execução do comando.

Truncando uma tabela

A instrução conhecida como TRUNCATE TABLE remove todas as linhas de uma tabela, liberando o espaço por ela ocupado. Diferentemente do comando DROP TABLE, este comando não destrói a estrutura, apenas elimina os dados lá existentes. Veja a sintaxe.

```
TRUNCATE TABLE nome_tabela;
```

Como você ainda não inseriu dados nas tabelas criadas, a execução deste comando não terá efeito nenhum. Mesmo assim, apresento a sua funcionalidade.

```
TRUNCATE TABLE departamento30;
```

ATENÇÃO! Após a execução do comando, não será possível reverter as modificações.

RESUMO

Instrução	Descrição
CREATE TABLE	Cria uma tabela
ALTER TABLE	Modifica as estruturas da tabela
DROP TABLE	Remove linhas e estruturas da tabela
TRUNCATE	Remove linhas, liberando o espaço armazenado



PELO FUTURO DO TRABALHO

sesisc.org.br     **sc.senai.br**

0800 048 1212

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC