Banco de Dados - Ciclo 3 - Webaula 17

WEBAULA 17

- Introdução ao Ciclo
- Criação e Gerenciamento de Tabelas
- Referências

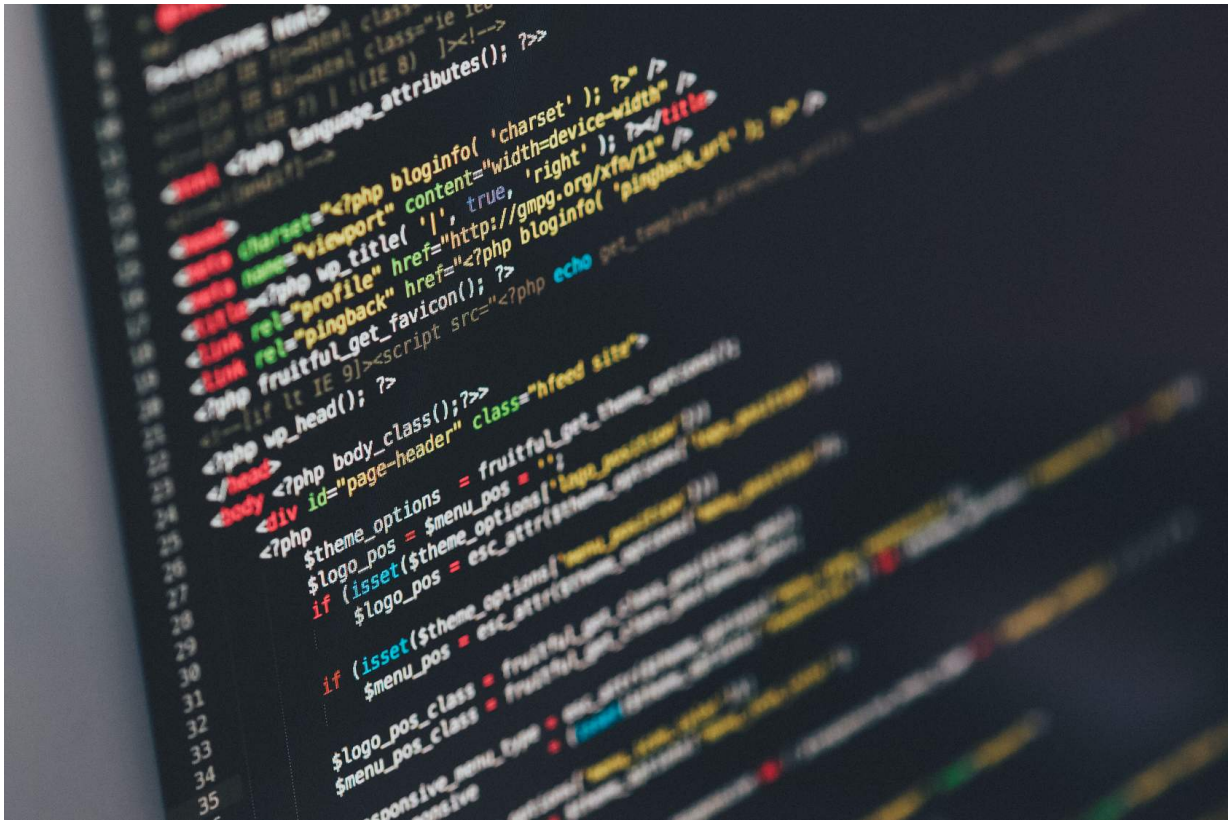
QUESTION BANKS

Lição 1 de 3

Introdução ao Ciclo

Esta etapa da nossa disciplina é dedicada à linguagem SQL, que é utilizada para interagir com os principais SGBDs relacionais através de instruções/comandos.

O objetivo ao término deste Ciclo é que você conheça a linguagem SQL utilizada para definir estruturas e manipular os dados em SGBDs relacionais e, desta forma, tenha totais condições de explorar as principais funcionalidades deste modelo de banco de dados. Mas para isso é muito importante que você pratique o uso da linguagem SQL, reproduzindo os exemplos passados ao longo do conteúdo, realizando as atividades propostas e questionando sempre que necessário.



Nosso ponto de partida será a criação de manipulação de tabelas, ou seja, o principal objeto mantido pelos SGBDs relacionais, responsáveis pelo armazenamento dos dados informados pelo usuário.

Dica

A linguagem SQL pode ser dividida em categorias. As mais conhecidas e utilizadas são a DDL (Data Definition Language), em que os comandos são responsáveis pela criação e gerenciamento das estruturas, e a DML (Data Manipulation Language), onde temos os comandos que possibilitam a manutenção dos dados, ou seja, a inserção, alteração e exclusão dos dados armazenados nas estruturas (tabelas).

CONTINUE

Criação e Gerenciamento de Tabelas

Neste tópico vamos começar a conhecer a linguagem SQL. Alguns dos comandos usados nessa linguagem serão padrão nos SGBDs relacionais que seguem o padrão ANSI (American National Standards Institute), porém outros poderão apresentar pequenas alterações decorrentes de particularidades, como tipos de dados ou palavras reservadas específicas.

Como dito anteriormente, os exemplos de comandos apresentados neste e nos módulos seguintes têm como base o padrão ANSI e foram executados no SGBD Oracle. Destacamos que ao longo dos conteúdos faremos correlações com outros SGBDs relacionais. Mesmo assim, se você se interessar em executar as mesmas ações em outro SGBD relacional e encontrar divergências, poderá reportá-las sempre que julgar necessário por meio dos nossos canais de comunicação, para que possamos enriquecer juntos em conhecimento.

Vamos dar início à nossa aprendizagem criando e manipulando estruturas conhecidas como tabelas. 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.

Glossário

Tabela: unidade básica de armazenamento, composta de uma ou mais linhas.

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

Existem algumas recomendações para a nomeação de tabelas e colunas:

- comece sempre com uma letra;
- utilize no máximo 30 caracteres;
- dê preferência aos 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.

> **Dica**
>
> Apesar de cada SGBD relacional apresentar características específicas quanto às possibilidades de utilização dos identificadores, a adoção das recomendações acima poderá minimizar eventuais problemas com a nomenclatura usada para nomes de tabelas e colunas.

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. Logo à frente teremos um exemplo para melhor ilustrar esta correlação a partir dos nomes das colunas envolvendo o número do departamento, que é chamada de DEPTNO nas tabelas EMP e DEPT.

A instrução utilizada para criar tabelas é a CREATE TABLE, que é uma das instruções DDL (Data Definition Language). 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 apresentada no Quadro 1:

Quadro 1 – Sintaxe para a criação de tabela

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

Como você pode observar (Quadro 1), ao criarmos colunas é necessário informar o tipo de dados. O tipo de dados e os argumentos variam de acordo com o SGBD relacional adotado, tornando necessário verificar a documentação para utilizar os tipos corretamente. Para auxiliá-lo, inicialmente, segue uma tabela com os tipos de dados mais comuns e sua representação nos SGBDs relacionais já comentados na disciplina (Figura 37):

	FIREBIRD	MYSQL	ORACLE	POSTGRESQL
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

Figura 37

Saiba mais

Clique no nome do SGBD cuja lista completa de tipos de dados você deseja verificar:

- FIREBIRD;
- MYSQL;
- ORACLE;
- POSTGRESQL.

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. São valores permitidos: um valor literal, expressão ou função SQL. E não são permitidos valores provenientes de outro nome da coluna.

O Quadro 2 apresenta um exemplo de criação de tabela e a utilização da opção default:

Quadro 2 – Exemplo de criação de tabela

```
CREATE TABLE pessoa
(
  cd_pessoa INTEGER,
  nm_pessoa VARCHAR(255),
  fl_estado_civil CHAR(1) DEFAULT 'S'
 );
```

Agora que você já conheceu a instrução básica 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 modelo entidade-relacionamento (MER) da Figura 38.

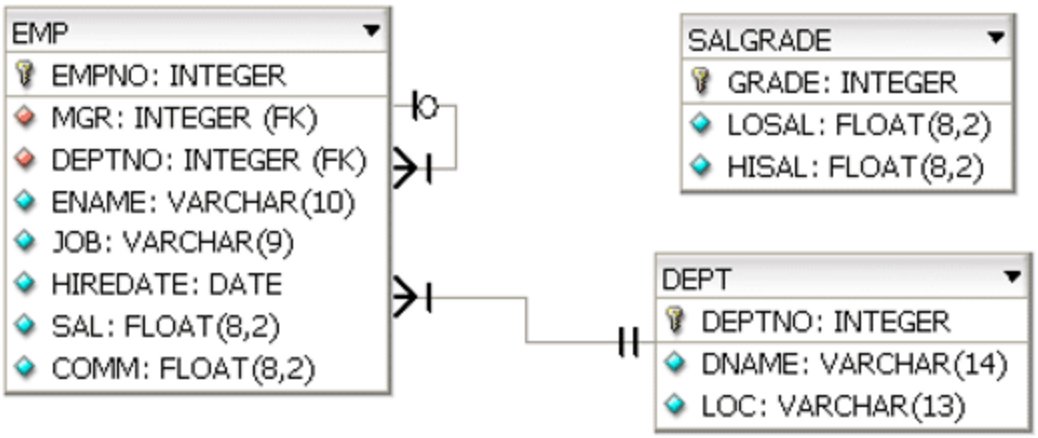


Figura 38

Dica

A estrutura utilizada no exemplo de MER (Figura 38) foi extraída da documentação do SGBD Oracle versão 11g (ORACLE, 2017). Trata-se de uma estrutura que contempla elementos capazes de demonstrar as principais funcionalidades do SGBD Oracle, o qual

procuramos adaptar para os demais SGBDs relacionais mais comuns no mercado atualmente. Assim, destacamos que ao longo da explicação dos conteúdos estas tabelas (EMP, DEPT e SALGRADE) estarão presentes.

Para auxiliá-lo, apresentamos a sentença de criação da tabela DEPT conforme apresentado no Quadro 3. Este comando é compatível com os principais SGBDs relacionais existentes, pois se utiliza de tipos de dados para colunas que seguem o padrão ANSI.

Quadro 3 – Comando SQL para a criação da tabela DEPT

```
CREATE TABLE DEPT
(
  DEPTNO INTEGER NOT NULL,
  DNAME VARCHAR(14),
  LOC VARCHAR(13)
 );
```

Na sequência, utilize-se da ferramenta gráfica de interação com o SGBD relacional escolhido e procure observar nas propriedades da tabela criada se a estrutura corresponde à representação gráfica apresentada no MER.

Dica

Atenção para o tipo de dado FLOAT utilizado no MER (Figura 38). Este tipo de dado pode apresentar varrições dependendo do SGBD relacional que você irá utilizar. Por isso, pesquise se há um tipo correlato caso ele apresente a mensagem de tipo incompatível.

Há uma segunda forma de criar tabelas, que se baseia em uma consulta, também conhecida como subconsulta. Para isso você precisa adicionar ao comando CREATE TABLE, já visto, a cláusula AS e a subconsulta na sequência. Veja a sintaxe no Quadro 4:

Quadro 4 – Sintaxe de criação de tabela a partir de estrutura existente

```
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.

> **Dica**
>
> O termo "subconsulta" é utilizado por se tratar de uma consulta dentro de um comando principal. Mas não se preocupe: neste momento não é fundamental que você conheça e domine o conceito de subconsulta, pois ele será visto mais adiante.

Observe o código na linguagem SQL apresentado no Quadro 5, que exemplifica a criação de uma tabela a partir de uma consulta realizada em uma estrutura (tabela) já existente:

Quadro 5 – Exemplo de criação de tabela

```
CREATE TABLE dept30
AS
```

```
SELECT empno, ename, sal, hiredate
FROM emp
WHERE   deptno = 30;
```

No exemplo, foi criada uma tabela chamada DEPT30. Nela há informações de colaboradores que pertencem ao departamento de número 30.

Se após a criação de uma tabela você necessitar fazer alguma alteração, poderá fazê-la utilizando a instrução ALTER TABLE. Use a instrução ALTER TABLE para:

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

Considere que no exemplo criado anteriormente (tabela DEPT30) houvesse a necessidade da inclusão de uma nova coluna, por exemplo, “job”. Observe na Figura 39 a ilustração que representa a adição de uma nova coluna:

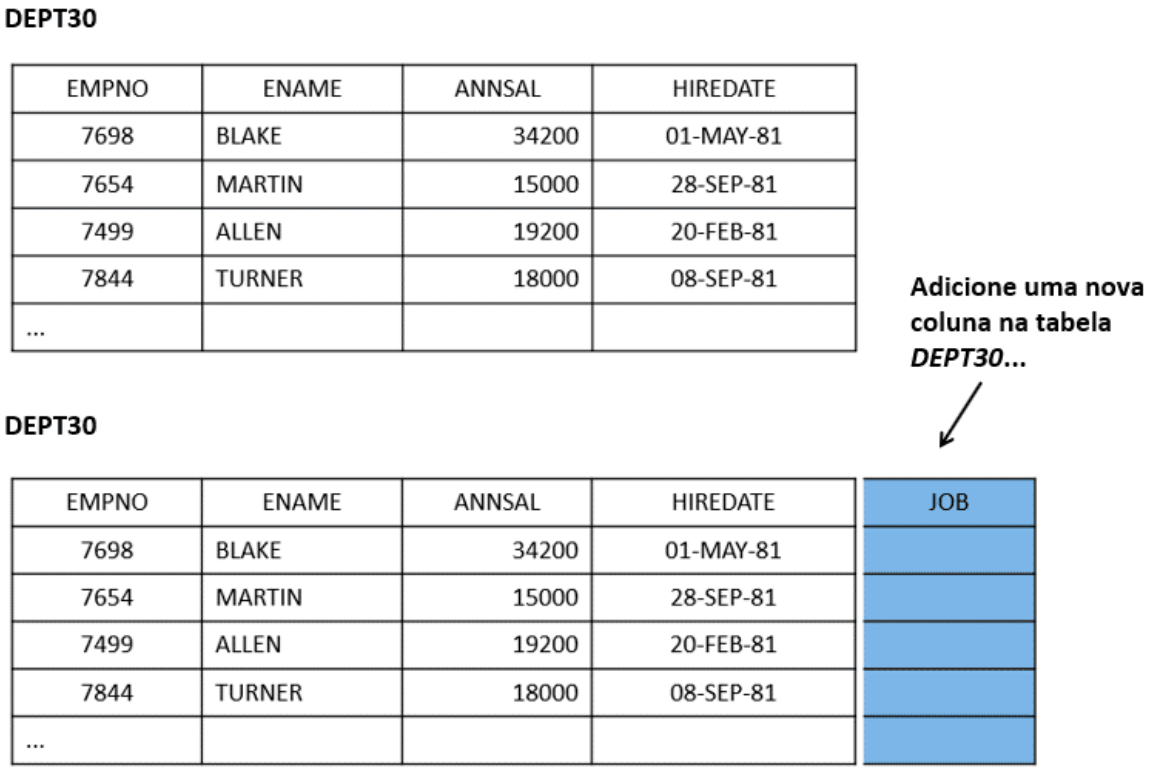


Figura 39
Fonte: Adaptado de Oracle (2017)

Para adicionar colunas após a criação de uma tabela utilizando a instrução ALTER TABLE, é necessário a utilização da cláusula ADD COLUMN, conforme se observa no Quadro 6.

Quadro 6 – Sintaxe para adição de coluna a uma tabela

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

Aproveite a tabela DEPT30 e execute o comando para a adição de uma nova coluna chamada JOB na tabela DEPT30. Essa coluna deve permitir o armazenamento do cargo do funcionário em no máximo 9 caracteres.

> **Dica**
>
> Há SGBDs relacionais que não necessitam da palavra reservada "column", limitando-se ao uso do "add" para adicionar coluna. Fique atento e, se tiver dúvidas, sempre consulte o manual de utilização do respectivo SGBD. Estas pequenas variações são comuns.

Veja no Quadro 7 como ficaria a instrução SQL para o comando de adição da coluna "job" à tabela DEPT30.

Quadro 7 – Sintaxe para adição de coluna a uma tabela

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

Assim como você pode adicionar colunas em uma tabela existente, também é possível modificar propriedades de uma coluna existe na tabela. Para isto, utilizamos também a instrução ALTER TABLE, porém agora com a cláusula ALTER COLUMN. Observe no Quadro 8 a sintaxe de utilização deste recurso:

Quadro 8 – Sintaxe para alteração de coluna em uma tabela

```
-- sintaxe  no SGBD MySQL
```

```
ALTER TABLE tabela
   ALTER COLUMN coluna tipo_de_dado;


-- sintaxe  no SGBD Oracle
ALTER TABLE tabela
   MODIFY coluna tipo_de_dado;
```

> **Dica**
>
> Observe que no Quadro 8 já apresentamos duas possibilidades de alteração na definição de uma coluna. Isto se dá porque não há um padrão entre os SGBDs relacionais. O MySQL, por exemplo, utiliza a primeira sintaxe, enquanto o Oracle utiliza a segunda. Novamente, fique atento à documentação específica do SGBD que você irá utilizar.

Vejamos um exemplo de utilização da funcionalidade. Considere que seja necessário aumentar o tamanho da coluna "job" da tabela DEPT30, passando dos atuais 9 caracteres para o tamanho de 15 caracteres. Veja no Quadro 9 como ficaria a sentença SQL correspondente:

Quadro 9 – Exemplo de alteração de coluna em uma tabela

```
-- exemplo  no SGBD MySQL
ALTER TABLE dept30
   ALTER COLUMN job varchar(15);


-- exemplo  no SGBD Oracle
ALTER TABLE depto30
   MODIFY job varchar(15);
```

Há cenários em que uma determinada coluna não se faz mais necessária na estrutura de uma tabela. Neste caso, é possível removê-la utilizando também o comando ALTER TABLE, porém agora com a cláusula DROP COLUMN. Veja a sintaxe no Quadro 10.

Quadro 10 – Sintaxe de exclusão de coluna em uma tabela

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

Dica

De igual forma, aqui há SGBDs relacionais que não necessitam da palavra reservada "column", limitando-se ao uso do "drop" para remover a coluna.

Agora vejamos na prática a utilização da instrução, considerando que não necessitamos mais da coluna "job" da tabela DEPT30. O Quadro 11 ilustra a sentença SQL correspondente.

Quadro 11 – Sintaxe de exclusão de coluna em uma tabela

```
ALTER TABLE nome_tabela
  DROP COLUMN job;
```

Da mesma forma como podemos identificar que uma coluna não se faz mais necessária em uma tabela, também podemos entender que uma tabela não é mais necessária em uma estrutura de banco de dados. Neste caso, podemos remover a tabela utilizando o comando DROP TABLE. O Quadro 12 apresenta a sintaxe do comando:

Quadro 12 – Sintaxe de exclusão de uma tabela

```
DROP TABLE nome_tabela;
```

Consideremos o cenário no qual não necessitamos mais da tabela DEPT30, criada anteriormente para exemplificar o uso de funcionalidades da linguagem SQL. Para isto, você poderia removê-la utilizando a instrução apresentada no Quadro 13.

Quadro 13 – Exemplo de exclusão da tabela DEPT30

```
DROP TABLE dept30;
```

Dica

ATENÇÃO! Ao executar o comando DROP TABLE, todos os dados estruturados serão perdidos. Além disso, não será possível reverter a execução do comando.

Se consideramos um cenário no qual necessitamos eliminar todos os dados, porém mantendo a estrutura de uma tabela, não podemos utilizar o comando DROP TABLE que acabamos de ver. Para isso há uma instrução conhecida como TRUNCATE TABLE, que remove todas as linhas de uma tabela,

liberando o espaço por ela ocupado e, diferentemente do comando DROP TABLE, este comando não destrói a estrutura, apenas elimina os dados lá existentes. Veja a sintaxe do comando no Quadro 14:

Quadro 14 – Sintaxe de truncamento de tabela

```
TRUNCATE TABLE tabela;
```

Um exemplo de sentença SQL para eliminar todas as linhas da tabela DEPT pode ser visualizada no Quadro 15.

Quadro 15 – Exemplo de instrução para truncamento da tabela DEPT

```
TRUNCATE TABLE dept;
```

É importante destacar que o comando TRUNCATE TABLE não permite a escolha pela eliminação parcial dos dados da tabela, ou seja, todos os dados da tabela serão excluídos. De igual forma como no comando DROP TABLE, aqui não há a possibilidade de reversão do comando executado.

Encerramento

Chegamos ao fim deste tópico, onde você teve a oportunidade de conhecer os comandos para a criação, alteração e exclusão da estrutura básica de um banco de dados, a tabela. Na sequência você irá aprender a inserir, modificar e eliminar as restrições nas tabelas.

CONTINUE

Referências

Referências

ORACLE. Documentação de utilização do sistema (Versão 11g). Califórnia: Oracle Corp., 2017.

CARDOSO, Vírginia M. **Linguagem SQL**: fundamentos e práticas. São Paulo: Saraiva, 2009.

Referências de Imagens

Divisão de Modalidades de ensino (DME), Fundação Universidade Regional de Blumenau (FURB), 2019.