

Sistemas de Informação

Design e Desenvolvimento de Banco de Dados

PROF. MILTON

Criando e Gerenciando Tabelas

Objetivos

- Depois de completar esta lição, você poderá fazer o seguinte:
 - Descrever os principais objetos do banco de dados
 - Criar tabelas
 - Descrever os tipos de dados que podem ser usados ao especificar a definição da coluna
 - Alterar definições de tabela
 - Eliminar, renomear e truncar tabelas

Objetivo da Lição

Nesta lição, você aprenderá sobre os principais objetos do banco de dados e o relacionamento entre eles. Você também aprenderá como criar, alterar e eliminar tabelas.

Objetos do Banco de Dados

| Objeto | Descrição |
|-----------|---|
| Tabela | Unidade básica de armazenamento, composta de linhas uma ou mais tabelas |
| View | Representa logicamente subconjuntos de dados de uma ou mais tabelas |
| Seqüência | Gera valores de chave primária |
| Índice | Melhora o desempenho de algumas consultas |
| Sinônimo | Atribui nomes alternativos a objetos |

Objetos do Banco de Dados

Um banco de dados Oracle pode conter várias estruturas de dados. Cada estrutura deve ser descrita no projeto do banco de dados para que possa ser criada durante o estágio de desenvolvimento do banco de dados.

Tabela: Armazena dados

View: Subconjunto de dados de uma ou mais tabelas

Seqüência: Gera valores de chave primária

Índice: Melhora o desempenho de algumas consultas

Sinônimo: Atribui nomes alternativos a objetos

Estruturas de Tabela do Oracle

As tabelas podem ser criadas a qualquer momento, até mesmo quando os usuários estiverem usando o banco de dados.

Não é necessário especificar o tamanho de nenhuma tabela. O tamanho é definido pela quantidade de espaço alocada no banco de dados como um todo.

Entretanto, é importante estimar a quantidade de espaço que uma tabela usará.

A estrutura da tabela pode ser modificada on-line.

Observação: Há mais objetos de banco de dados disponíveis que não são abordados neste curso.

Convenções para Nomeação

- Deve começar com uma letra
- Pode ter de 1 a 30 caracteres
- Deve conter somente A–Z, a–z, 0–9, _, \$ e #
- Não deve duplicar o nome de outro objeto de propriedade do mesmo usuário
- Não deve ser uma palavra reservada pelo Oracle Server

Regras para Nomeação

Nomeie tabelas e colunas do banco de dados de acordo com as regras default de nomeação de qualquer objeto do banco de dados Oracle:

Nomes de tabela e de colunas devem começar com uma letra e podem ter de 1 a 30 caracteres.

Os nomes devem conter somente os caracteres A-Z, a-z, 0-9, _ (sublinhado), \$ e # (caracteres legais, mas evite usá-los).

Os nomes não devem duplicar o nome de outro objeto de propriedade do mesmo usuário do Oracle Server.

Os nomes não devem ser uma palavra reservada do Oracle Server.

Diretrizes para Nomeação

Use nomes descritivos para tabelas e outros objetos do banco de dados.

Nomeie a mesma entidade de modo consistente em diferentes tabelas. Por exemplo, a coluna do número do departamento é chamada DEPTNO nas tabelas EMP e DEPT.

Observação: Os nomes não fazem distinção entre maiúsculas de minúsculas.

Por exemplo, EMP é tratada com o mesmo nome que eMP ou eMp.

Para obter mais informações, consulte o *Oracle Server SQL Reference*, "Object

Names and Qualifiers".

A Instrução CREATE TABLE

– Você deve ter:

- privilégio CREATE TABLE
- Uma área de armazenamento

```
CREATE [GLOBAL TEMPORARY] TABLE [esquema.]tabela
      (tipo de dados da coluna
      [DEFAULT expr][, ...]);
```

– Especifique:

- Nome da tabela
- Nome da coluna, tipo de dados da coluna e tamanho da coluna

A Instrução CREATE TABLE

Crie tabelas para armazenar dados executando a instrução SQL CREATE TABLE. Ela é uma das instruções **DDL (data definition language)** a ser abordada nas próximas lições. As instruções DDL são um subconjunto de instruções SQL usadas para criar, modificar ou remover estruturas do banco de dados Oracle8. Essas instruções possuem um efeito imediato no banco de dados e também registram informações no dicionário de dados. Para criar uma tabela, o usuário deve ter o privilégio **CREATE TABLE** e uma área de armazenamento na qual criar objetos. O administrador do banco de dados usa instruções DCL (data control language), que serão abordadas mais tarde.

Na sintaxe:

| | |
|----------------------|---|
| GLOBAL TEMPORARY | especifica que a tabela é temporária e que sua definição está visível em todas as sessões. Os dados em uma tabela temporária são visíveis somente na sessão que insere dados na tabela. |
| <i>esquema</i> | é o mesmo do nome do proprietário |
| <i>tabela</i> | é o nome da tabela |
| DEFAULT <i>expr</i> | especifica um valor default se um valor estiver omitido na instrução INSERT |
| <i>coluna</i> | é o nome da coluna |
| <i>tipo de dados</i> | é o tipo de dados e o comprimento da coluna |

Fazendo Referência a Tabelas de Outro Usuário

- As tabelas que pertencem a outros usuários não estão no esquema do usuário.
- Você deve usar o nome do proprietário como um prefixo da tabela.

Fazendo Referência a Tabelas de Outro Usuário

Um *esquema* é um conjunto de objetos. Os objetos de **esquema** são as estruturas lógicas que fazem referência diretamente aos dados no banco de dados. Os objetos de esquema incluem tabelas, views, sinônimos, seqüências, procedimentos armazenados, índices, clusters e vínculos com o banco de dados.

Se uma tabela não pertencer ao usuário, o nome do proprietário deve ser prefixado à tabela.

A Opção DEFAULT

- Especifique um valor default para uma coluna durante uma inserção.

```
... hiredate DATE DEFAULT SYSDATE, ...
```

- Valores legais são um valor literal, expressão ou função SQL.
- Valores ilegais são outro nome da coluna ou pseudocoluna.
- O tipo de dados default deve corresponder ao tipo de dados da coluna.

A Opção DEFAULT

Um valor default pode ser dado à uma coluna usando-se a opção **DEFAULT**. Essa opção impede que valores nulos entrem nas colunas se uma linha for inserida sem um valor para a coluna. O valor default pode ser um literal, uma expressão ou uma função SQL, como SYSDATE e USER, mas o valor não pode ser o nome de outra coluna ou pseudocoluna como NEXTVAL ou CURRVAL. A expressão default deve corresponder ao tipo de dados da coluna.

Criando Tabelas

– Crie a tabela.

```
SQL> CREATE TABLE dept
2      (deptno NUMBER(2),
3      dname  VARCHAR2(14),
4      loc    VARCHAR2(13));
Table created.
```

```
SQL> DESCRIBE dept
```

| Name | Null? | Type |
|--------|-------|--------------|
| DEPTNO | | NUMBER(2) |
| DNAME | | VARCHAR2(14) |
| LOC | | VARCHAR2(13) |

Criando Tabelas

O exemplo no slide cria a tabela DEPT, com três colunas — chamadas, DEPTNO, DNAME e LOC. Ele também confirma a criação da tabela emitindo o comando DESCRIBE.

Como criar uma tabela é uma instrução DDL, um commit automático ocorre quando essa instrução é executada.

Tabelas no Banco de Dados Oracle

– Tabelas do Usuário

- Conjunto de tabelas criadas e mantidas pelo usuário
- Contêm informações sobre o usuário

– Dicionário de dados

- Conjunto de tabelas criadas e mantidas pelo Oracle server
- Contêm informações sobre o banco de dados

Tabelas no Banco de Dados Oracle

Tabelas do usuário são tabelas criadas pelo usuário, como EMP. Há outro conjunto de tabelas e views no banco de dados Oracle conhecido como o *dicionário de dados*. Esse conjunto é criado e mantido pelo Oracle Server e contém informações sobre o banco de dados.

Todas as **tabelas de dicionário** de dados são de propriedade do usuário SYS. As tabelas-base dificilmente são acessadas pelo usuário porque a informação contida nelas não é de fácil compreensão. Assim, os usuários geralmente acessam as views do dicionário de dados porque as informações são apresentadas em um formato mais fácil de entender. As informações armazenadas no dicionário de dados incluem nomes dos usuários do Oracle Server, privilégios concedidos a usuários, nomes de objeto do banco de dados, restrições de tabela e informações sobre auditoria. Há quatro categorias de views do dicionário de dados, cada categoria contém um prefixo distinto que reflete o uso pretendido.

| Prefixo | Descrição |
|---------|---|
| USER_ | Estas views contêm informações sobre objetos de propriedade do usuário. |
| ALL_ | Estas views contêm informações sobre todas as tabelas (de objeto e relacionais) acessíveis ao usuário. |
| DBA_ | Estas views são restritas. Estas views somente podem ser acessadas por pessoas que tenham sido atribuídas o DBA total. |
| V\$_ | Estas views contêm informações sobre views de desempenho dinâmico, desempenho do servidor do banco de dados e bloqueio. |

Consultando o Dicionário de Dados

– Descreva tabelas de propriedade do usuário.

```
SQL> SELECT *  
2 FROM user_tables;
```

– Exiba tipos de objetos distintos de propriedade do usuário.

```
SQL> SELECT DISTINCT object_type  
2 FROM user_objects;
```

– Exiba tabelas, views, sinônimos e seqüências de propriedade do usuário.

```
SQL> SELECT *  
2 FROM user_catalog;
```

11

Consultando o Dicionário de Dados

Você pode consultar as tabelas de dicionário de dados para exibir vários objetos de banco de dados de sua propriedade. As tabelas de dicionário de dados frequentemente usadas são:

USER_TABLES

USER_OBJECTS

USER_CATALOG

Observação: USER_CATALOG possui um sinônimo chamado CAT. Você pode usar esse sinônimo no lugar de USER_CATALOG nas instruções SQL.

```
SELECT *  
FROM CAT;
```

Tipos de Dados

FIAP

| Tipo de Dados | Descrição |
|----------------------------|---|
| VARCHAR2(<i>tamanho</i>) | Dados de caractere de comprimento variável |
| CHAR(<i>tamanho</i>) | Dados de caractere de comprimento fixo |
| NUMBER(<i>p,s</i>) | Dados numéricos de comprimento variável |
| DATE | Valores de data e hora |
| LONG | Dados de caractere de comprimento variável até 2 gigabytes |
| CLOB | Dados de caractere de um byte de até 4 gigabytes |
| RAW e LONG RAW | Dados binários brutos |
| BLOB | Dados binários de até 4 gigabytes |
| BFILE | Dados binários armazenados em um arquivo externo de até 4 gigabytes |

12

Tipos de Dados

| Tipo de Dados | Descrição |
|----------------------------|---|
| VARCHAR2(<i>tamanho</i>) | Dados de caractere de comprimento variável (Um <i>tamanho</i> máximo deve ser especificado). O <i>tamanho</i> default e mínimo é 1; o <i>tamanho</i> máximo é 4000.) |
| CHAR(<i>tamanho</i>) | Dados de caractere de comprimento fixo de bytes de <i>tamanho</i> de comprimento (O <i>tamanho</i> default e mínimo é 1; o <i>tamanho</i> máximo é 2000.) |
| NUMBER(<i>p,s</i>) | Número contendo a precisão <i>p</i> e a escala <i>s</i> (A precisão é o número total de dígitos decimais e a escala é o número de dígitos à direita do ponto decimal. A precisão pode variar de 1 a 38 e a escala, de -84 a 127.) |
| DATE | Valores de data e hora entre 1º de janeiro, 4712 A.C. e 31 de dezembro, 9999 D.C. |
| LONG | Dados de caractere de comprimento variável até 2 gigabytes |
| CLOB | Dados de caractere de um byte de até 4 gigabytes |

Tipos de Dados

| Tipo de Dados | Descrição |
|----------------------------|---|
| VARCHAR2(<i>tamanho</i>) | Dados de caractere de comprimento variável |
| CHAR(<i>tamanho</i>) | Dados de caractere de comprimento fixo |
| NUMBER(<i>p,s</i>) | Dados numéricos de comprimento variável |
| DATE | Valores de data e hora |
| LONG | Dados de caractere de comprimento variável até 2 gigabytes |
| CLOB | Dados de caractere de um byte de até 4 gigabytes |
| RAW e LONG RAW | Dados binários brutos |
| BLOB | Dados binários de até 4 gigabytes |
| BFILE | Dados binários armazenados em um arquivo externo de até 4 gigabytes |

Tipos de dados (continuação)

| Tipo de Dados | Descrição |
|-----------------------|--|
| RAW(<i>tamanho</i>) | Dados binários brutos de <i>tamanho</i> de comprimento (Um <i>tamanho</i> máximo deve ser especificado. O <i>tamanho</i> máximo é 2000.) |
| LONG RAW | Dados binários brutos de comprimento variável de até 2 gigabytes |
| BLOB | Dados binários de até 4 gigabytes |
| BFILE | Dados binários armazenados em um arquivo externo; até 4 gigabytes |

Criando uma Tabela Usando uma Subconsulta

FIAP

- Crie uma tabela e insira linhas combinando a instrução CREATE TABLE e a opção da *subconsulta* AS.

```
CREATE TABLE tabela  
[(coluna, coluna...)]  
AS subconsulta;
```

- Faça a correspondência do número de colunas especificadas com o número de colunas da subconsulta.
- Defina colunas com nomes de colunas e valores default.

14

Criando uma Tabela a Partir de Linhas em Outra Tabela

Um segundo método para criar uma tabela é aplicar a cláusula da *subconsulta* AS para criar a tabela e inserir linhas retornadas da subconsulta.

Na sintaxe:

| | |
|--------------------|--|
| <i>tabela</i> | é o nome da tabela. |
| <i>coluna</i> | é o nome da coluna, valor default e restrição de integridade. |
| <i>subconsulta</i> | é a instrução SELECT que define o conjunto de linhas a serem inseridas |
| | sua nova tabela. |

Diretrizes

A tabela será criada com os nomes de coluna especificados e as linhas recuperadas pela instrução SELECT serão inseridas na tabela.

A definição da coluna pode conter somente o nome da coluna e o valor default.

Se forem dadas especificações para a coluna, o número de colunas deve ser igual ao número de colunas na lista SELECT da subconsulta.

Se não forem dadas especificações para a coluna, os nomes de coluna da tabela serão os mesmos que os nomes de coluna na subconsulta.

Criando uma Tabela Usando uma Subconsulta

```
SQL> CREATE TABLE dept30
2 AS
3 SELECT empno, ename, sal*12 ANNSAL, hiredate
4 FROM emp
5 WHERE deptno = 30;
Table created.
```

```
SQL> DESCRIBE dept30
```

| Name | Null? | Type |
|----------|----------|---------------|
| EMPNO | NOT NULL | NUMBER (4) |
| ENAME | | VARCHAR2 (10) |
| ANNSAL | | NUMBER |
| HIREDATE | | DATE |

Criando uma Tabela a Partir de Linhas em Outra Tabela (continuação)

O exemplo do slide cria uma tabela, DEPT30, que contém detalhes sobre todos os funcionários que trabalham no departamento 30. Observe que os dados da tabela DEPT30 vêm da tabela EMP.

Você pode verificar a existência de uma tabela do banco de dados e verificar as definições da coluna usando o comando DESCRIBE do SQL*Plus.

Forneça um apelido para a coluna ao selecionar uma expressão.

A Instrução ALTER TABLE

- Use a instrução ALTER TABLE para:
 - Adicionar uma nova coluna
 - Modificar uma coluna existente
 - Definir um valor default para a nova coluna

```
ALTER TABLE tabela
ADD          (tipo de dados da coluna [DEFAULT expr]
             [, tipo de dados da coluna]...);
```

```
ALTER TABLE tabela
MODIFY       (tipo de dados da coluna [DEFAULT expr]
             [, tipo de dados da coluna]...);
```

16

Instrução ALTER TABLE

Após criar suas tabelas, você talvez precise alterar as estruturas da tabela porque omitiu uma coluna ou a definição da coluna precisa ser alterada. Você pode fazer isso usando a instrução ALTER TABLE.

Você pode adicionar colunas a uma tabela usando a instrução **ALTER TABLE** com a cláusula ADD.

Na sintaxe:

| | |
|----------------------|--|
| <i>tabela</i> | é o nome da tabela |
| <i>coluna</i> | é o nome da nova coluna |
| <i>tipo de dados</i> | é o tipo de dados e o comprimento da nova coluna |
| DEFAULT <i>expr</i> | especifica o valor default para a nova coluna |

Você pode modificar colunas existentes em uma tabela usando a instrução ALTER TABLE com a cláusula MODIFY.

Observação: O slide fornece a sintaxe abreviada para ALTER TABLE. ALTER TABLE será abordada com mais detalhes em uma lição mais adiante.

Adicionando uma Coluna

| DEPT30 | | | | Nova coluna |
|--------|--------|--------|-----------|-------------|
| EMPNO | ENAME | ANNSAL | HIREDATE | JOB |
| 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 DEPT30..."

| DEPT30 | | | | |
|--------|--------|--------|-----------|-----|
| EMPNO | ENAME | ANNSAL | HIREDATE | JOB |
| 7698 | BLAKE | 34200 | 01-MAY-81 | |
| 7654 | MARTIN | 15000 | 28-SEP-81 | |
| 7499 | ALLEN | 19200 | 20-FEB-81 | |
| 7844 | TURNER | 18000 | 08-SEP-81 | |
| ... | | | | |

Adicionando uma Coluna

O gráfico adiciona a coluna JOB à tabela DEPT30. Observe que a nova coluna torna-se a última coluna na tabela.

Adicionando uma Coluna

– Use a cláusula ADD para adicionar colunas.

```
SQL> ALTER TABLE dept30
2 ADD (job VARCHAR2(9));
Table altered.
```

– A coluna nova torna-se a última coluna.

| EMPNO | ENAME | ANNSAL | HIREDATE | JOB |
|-------|--------|--------|-----------|-----|
| 7698 | BLAKE | 34200 | 01-MAY-81 | |
| 7654 | MARTIN | 15000 | 28-SEP-81 | |
| 7499 | ALLEN | 19200 | 20-FEB-81 | |
| 7844 | TURNER | 18000 | 08-SEP-81 | |
| ... | | | | |

6 rows selected.

Diretrizes para Adicionar uma Coluna

Você pode adicionar ou modificar colunas, mas não pode removê-las de uma tabela.

Você não pode especificar onde a coluna deve aparecer. A coluna nova torna-se a última coluna.

O exemplo no slide adiciona uma coluna chamada JOB à tabela DEPT30. A coluna JOB torna-se a última coluna na tabela.

Observação: Se uma tabela já contiver linhas quando uma coluna for adicionada, então a nova coluna será inicialmente nula para todas as linhas.

Modificando uma Coluna

- Você pode alterar um tipo de dados, tamanho e valor default de uma coluna.

```
SQL> ALTER TABLE dept30  
2 MODIFY (ename VARCHAR2(15));  
Table altered.
```

- Uma alteração no valor default afeta somente as inserções subseqüentes à tabela.

Modificando uma Coluna

Você pode modificar uma definição de coluna usando a instrução ALTER TABLE com a cláusula **MODIFY**. A modificação da coluna pode incluir alterações ao tipo de dados, tamanho e valor default da coluna.

Diretrizes

- Aumente a largura ou precisão de uma coluna numérica.
- Diminua a largura de uma coluna se ela contiver somente valores nulos e se a tabela não tiver linhas.
- Altere o tipo de dados se a coluna contiver valores nulos.
- Converta uma coluna CHAR para o tipo de dados VARCHAR2 ou converta uma coluna VARCHAR2 para o tipo de dados CHAR se a coluna contiver valores nulos ou se você não alterar o tamanho.
- Uma alteração no valor default de uma coluna afeta somente as inserções subseqüentes à tabela.

Eliminando uma Coluna

Use a cláusula DROP COLUMN para eliminar colunas que você não precisa mais na tabela.

```
SQL> ALTER TABLE dept30  
2 DROP COLUMN job ;  
Table altered.
```

20

Eliminando uma Coluna

Você pode eliminar uma coluna de uma tabela usando a instrução ALTER TABLE com a cláusula **DROP COLUMN**. Esse é um recurso disponível a partir do Oracle8i.

Diretrizes

- A coluna pode ou não conter dados.

- Somente uma coluna pode ser eliminada por vez.

- A tabela deve permanecer com pelo menos uma coluna após ser alterada.

- Depois que a coluna for eliminada, não poderá ser recuperada.

Opção SET UNUSED

- Use a opção SET UNUSED para marcar uma ou mais colunas como não usadas.
- Use a opção DROP UNUSED COLUMNS para remover as colunas marcadas como UNUSED.

```
ALTER TABLE tabela  
SET UNUSED (coluna);  
  
OR  
  
ALTER TABLE tabela  
SET UNUSED COLUMN coluna;
```

```
ALTER TABLE tabela  
DROP UNUSED COLUMNS;
```

Opção SET UNUSED

A opção **SET UNUSED** marca uma ou mais colunas como não usadas para que possam ser eliminadas quando a demanda nos recursos do sistema for menor. Esse recurso está disponível no Oracle8i. Ao especificar esta cláusula você, na verdade, não remove as colunas de destino de cada linha na tabela (ou seja, não restaura o espaço em disco usado por essas colunas). Por isso, o tempo de resposta é mais rápido do que seria se você executasse a cláusula DROP. As colunas não usadas são tratadas como se fossem eliminadas, embora os dados da coluna permaneçam nas linhas da tabela. Após uma coluna ter sido marcada como não usada, você não terá acesso a essa coluna. Uma consulta "SELECT *" não recuperará os dados de colunas não usadas. Além disso, os nomes e tipos de colunas marcadas como não usadas não serão exibidos durante DESCRIBE e você poderá adicionar à tabela uma nova coluna com o mesmo nome que uma coluna não usada.

Opção DROP UNUSED COLUMNS

DROP UNUSED COLUMNS remove da tabela todas as colunas marcadas atualmente como não usadas. Você pode usar essa instrução quando quiser recuperar o espaço em disco extra de colunas não usadas na tabela. Se a tabela não contiver colunas não usadas, a instrução retornará sem erros.

```
SQL> ALTER TABLE dept30  
2 SET UNUSED (ename);  
Table altered.
```

```
SQL> ALTER TABLE dept30  
2 DROP UNUSED COLUMNS;  
Table altered.
```

Eliminando uma Tabela

- Todos os dados e estrutura da tabela serão excluídos.
- Todas as transações pendentes sofrerão commit.
- Todos os índices serão eliminados.
- Você *não pode* fazer roll back desta instrução.

```
SQL> DROP TABLE dept30;  
Table dropped.
```

23

Eliminando uma Tabela

A instrução **DROP TABLE** remove a definição de uma tabela do Oracle8. Quando você elimina uma tabela, o banco de dados perde todos os dados na tabela e todos os índices associados a ela.

Sintaxe

```
DROP TABLE tabela;
```

onde: *tabela*

é o nome da tabela

Diretrizes

Todos os dados são deletados da tabela.

As views e sinônimos permanecerão, mas serão inválidos.

Todas as transações pendentes sofrerão commit.

Somente o criador da tabela ou um usuário com o privilégio DROP ANY TABLE poderá remover uma tabela.

A instrução DROP TABLE, uma vez executada, é irreversível. O Oracle Server não questiona a ação quando você emite a instrução DROP TABLE. Se você possuir tal tabela ou tiver um privilégio de nível superior, então a tabela será imediatamente removida. Todas as instruções DDL emitem um commit, tornando assim a transação permanente.

Alterando o Nome de um Objeto

- Para alterar o nome de uma tabela, view, seqüência ou sinônimo, execute a instrução RENAME.

```
SQL> RENAME dept TO department;  
Table renamed.
```

- Você deve ser o proprietário do objeto.

Renomeando uma Tabela

As instruções DDL adicionais incluem a instrução **RENAME**, que é usada para renomear uma tabela, view, seqüência ou sinônimo.

Sintaxe

```
RENAME      old_name TO new_name;
```

onde: *old_name* é o nome antigo da
tabela, da view, da seqüência ou do sinônimo

new_name é o novo nome da tabela,
da view, da seqüência ou do sinônimo

Você deve ser o proprietário do objeto que renomear.

Truncando uma Tabela

– A Instrução TRUNCATE TABLE:

- Remove todas as linhas de uma tabela
- Libera o espaço de armazenamento usado por esta tabela

```
SQL> TRUNCATE TABLE department;  
Table truncated.
```

- Você não pode fazer roll back da remoção da linha ao usar TRUNCATE.
- Como alternativa, você pode remover as linhas usando a instrução DELETE.

25

Truncando uma Tabela

Outra instrução DDL é a instrução **TRUNCATE TABLE**, que é usada para remover todas as linhas de uma tabela e para liberar o espaço de armazenamento usado por essa tabela. Ao usar a instrução TRUNCATE TABLE, você não pode fazer roll back da remoção de linha.

Sintaxe

```
TRUNCATE TABLE tabela;
```

onde: *tabela*

é o nome

da tabela

Você deve ser o proprietário da tabela ou ter privilégios de sistema DELETE TABLE para truncar a tabela.

A instrução DELETE também pode remover todas as linhas de uma tabela, mas não libera o espaço de armazenamento.

Adicionando Comentários a uma Tabela

- Você pode adicionar comentários a uma tabela ou coluna usando a instrução **COMMENT**.

```
SQL> COMMENT ON TABLE emp  
2 IS 'Employee Information';  
Comment created.
```

- Os comentários podem ser exibidos através das views do dicionário de dados.
 - ALL_COL_COMMENTS
 - USER_COL_COMMENTS
 - ALL_TAB_COMMENTS
 - USER_TAB_COMMENTS

Adicionando um Comentário a uma Tabela

Você pode adicionar um comentário de até 2.000 bytes sobre uma coluna, tabela, view ou instantâneo usando a instrução **COMMENT**. O comentário é armazenado no dicionário de dados e pode ser visto em uma das seguintes views do dicionário de dados na coluna COMMENTS:

ALL_COL_COMMENTS
USER_COL_COMMENTS
ALL_TAB_COMMENTS
USER_TAB_COMMENTS

Sintaxe

```
COMMENT ON TABLE tabela | COLUMN tabela.coluna  
IS 'text';
```

| | |
|----------------------------|----------------------------------|
| onde: <i>tabela</i> | é o nome da tabela |
| <i>coluna</i> | é o nome da coluna em uma tabela |
| <i>texto</i> | é o texto do comentário |

Você pode eliminar um comentário do banco de dados definindo-o como uma string vazia (' ').

```
SQL> COMMENT ON TABLE emp IS ' ';
```

Sumário

| Instrução | Descrição |
|--------------|---|
| CREATE TABLE | Cria uma tabela |
| ALTER TABLE | Modifica as estruturas da tabela |
| DROP TABLE | Remove as linhas e estrutura da tabela |
| RENAME | Altera o nome de uma tabela, view, seqüência ou sinônimo |
| TRUNCATE | Remove todas as linhas de uma tabela e libera o espaço de armazenamento |
| COMMENT | Adiciona comentários a uma tabela ou view |

CREATE TABLE Cria uma tabela.

Cria uma tabela baseada em outra tabela usando uma subconsulta.

ALTER TABLE Modifica as estruturas da tabela.

Altera larguras da coluna, tipos de dados da coluna e adiciona colunas.

DROP TABLE Remove linhas e estrutura de uma tabela.

Uma vez executada, não é possível fazer roll back dessa instrução.

RENAME Renomeia uma tabela, view, seqüência ou sinônimo.

TRUNCATE Remove todas as linhas de uma tabela e libera o espaço de armazenamento usado pela tabela.

A instrução DELETE remove somente linhas.

COMMENT Adiciona um comentário a uma tabela ou coluna.

Consulta o dicionário de dados para ver o comentário.

Visão Geral do Exercício

- Criando novas tabelas
- Criando uma nova tabela usando a sintaxe `CREATE TABLE AS`
- Modificando definições da coluna
- Verificando que as tabelas existem
- Adicionando comentários às tabelas
- Eliminando tabelas
- Alterando tabelas

Visão Geral do Exercício

Crie novas tabelas usando a instrução `CREATE TABLE`. Confirme se a nova tabela foi adicionada ao banco de dados. Crie a sintaxe no arquivo de comandos e execute-o para criar a tabela.

Exercício 8

1. Crie a tabela DEPARTMENT de acordo com tabela de exemplo a seguir.

| Name | Null? | Type |
|-------|-------|---------------|
| ----- | ----- | ----- |
| ID | | NUMBER (7) |
| NAME | | VARCHAR2 (25) |

2. Create Crie a tabela EMPLOYEE de acordo com a tabela de exemplo a seguir.

| Name | Null? | Type |
|------------|-------|---------------|
| ----- | ----- | ----- |
| ID | | NUMBER (7) |
| LAST_NAME | | VARCHAR2 (25) |
| FIRST_NAME | | VARCHAR2 (25) |
| DEPT_ID | | NUMBER (7) |

Exercício 8 (continuação)

3. Modifique a tabela EMPLOYEE para aceitar os sobrenomes longos dos funcionários.

| Name | Null? | Type |
|------------|-------|---------------|
| ----- | ----- | ----- |
| ID | | NUMBER (7) |
| LAST_NAME | | VARCHAR2 (50) |
| FIRST_NAME | | VARCHAR2 (25) |
| DEPT_ID | | NUMBER (7) |

4. Confirme se as tabelas DEPARTMENT e EMPLOYEE foram armazenadas no dicionário de dados. (*Dica: USER_TABLES*)

```
TABLE_NAME
-----
DEPARTMENT
EMPLOYEE
```

5. Crie a tabela EMPLOYEE2 de acordo com a estrutura da tabela EMPLOYEE.
6. Elimine a tabela EMPLOYEE.
7. Renomeie a tabela EMPLOYEE2 para EMPLOYEE.
8. Adicione um comentário às definições das tabelas DEPARTMENT e EMPLOYEE, descrevendo as tabelas. Confirme as adições ao dicionário de dados.
9. Elimine a coluna LAST_NAME da tabela EMPLOYEE.