



ORACLE

Academy



Database Foundations

6-4

Linguagem de Manipulação de Dados (DML)

ORACLE
Academy



Roteiro

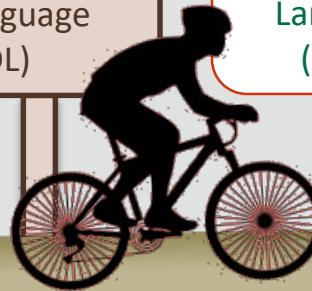
Introdução ao
Oracle
Application
Express (APEX)

Structured
Query
Language
(SQL)

Data
Definition
Language
(DDL)

Data
Manipulation
Language
(DML)

Transaction
Control
Language
(TCL)



Recuperand
o Dados com
SELECT

Restringindo
Dados com
WHERE

Classificando
Dados com
ORDER BY

Unindo Tabelas
com JOIN

Parte 6

Objetivos

- Esta lição abrange os seguintes objetivos:
 - Descrever a finalidade do DML (Data Manipulation Language)
 - Explicar as operações DML necessárias para gerenciar os dados de uma tabela de banco de dados:
 - INSERT
 - UPDATE
 - DELETE





Data Manipulation Language

- Uma instrução DML é executada quando você:
 - Adiciona novas linhas a uma tabela (INSERT)
 - Modifica as linhas existentes em uma tabela (UPDATE)
 - Remove as linhas existentes de uma tabela (DELETE)
- Uma transação consiste em um conjunto de instruções DML que formam uma unidade de trabalho lógica



Adicionando uma Nova Linha a uma Tabela

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

...

Nova linha

70	Public Relations	204	2700
----	------------------	-----	------

Insira uma nova linha
na tabela
DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	149	2500

Sintaxe da Instrução INSERT

- Adicione linhas a uma tabela usando a instrução INSERT:

```
INSERT INTO table [(column [, column...])]  
VALUES                (value [, value...]);
```

- Com essa sintaxe, somente uma linha é inserida por vez

Sintaxe da Instrução INSERT

- Antes de executar operações DML em uma tabela, faça uma cópia da tabela:

```
CREATE TABLE copy_departments  
AS (SELECT * FROM departments);
```

- Lembre-se de que, quando você copia uma tabela dessa maneira, todas as restrições não são copiadas (somente NOT NULL)

Inserindo Linhas

- Se você inserir uma linha que contenha valores para cada coluna, a lista de colunas não será necessária na cláusula INSERT
- Liste os valores na ordem padrão das colunas na tabela.
- Deve ser fornecido um valor para cada coluna

```
INSERT INTO copy_departments  
VALUES (40, 'Advertising', 201, 1800);
```

Inserindo Linhas

- Como opção, liste as colunas na cláusula INSERT

```
INSERT INTO copy_departments(department_id,  
                             department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);
```

- Liste os valores na mesma ordem que os campos listados
- Coloque os valores de data e caractere entre aspas simples

Cenário de Caso: Inserindo Linhas

Entendo que a instrução INSERT é usada para adicionar linhas a uma tabela. Será possível inserir uma linha se algumas colunas não tiverem valores?



Aluno



Corpo Docente

Com certeza. Você pode usar a instrução INSERT para especificar as colunas para as quais não serão inseridos valores.



Inserindo Linhas com Valores Nulos

- Método explícito : Omita a coluna da lista de colunas

```
INSERT INTO copy_departments(department_id,  
                             department_name)  
VALUES (30, 'Purchasing');
```

- Método implícito : Especifique a palavra-chave NULL na cláusula VALUES

```
INSERT INTO copy_departments  
VALUES (100, 'Finance', NULL, NULL);
```

Cenário de Caso: Inserindo Linhas

```
1 row(s) inserted.  
1 row(s) inserted.  
1 row(s) inserted.
```

```
INSERT INTO AUTHORS  
VALUES ('002','Oscar  
Wilde');  
  
INSERT INTO AUTHORS  
VALUES ('003','George  
Shaw');  
  
INSERT INTO AUTHORS  
VALUES ('004','Leo');
```

Linhas inseridas
com êxito





Inserindo Valores Especiais

- Você pode usar funções para inserir valores especiais na tabela
- A função SYSDATE registra a data e a hora atuais

```
INSERT INTO copy_employees (employee_id, first_name,  
                             last_name, email, phone_number,  
                             hire_date, job_id, salary,  
                             commission_pct, manager_id,  
                             department_id)  
  
VALUES( 113, 'Louis',  
        'Popp', 'LPOPP', '515.124.4567',  
        SYSDATE, 'AC_ACCOUNT', 6900,  
        NULL, 205,  
        110);
```

Inserindo Valores Específicos de Data e Hora

- Adicione um funcionário

```
INSERT INTO copy_employees  
VALUES(114, 'Den', 'Raphealy', 'DRAPHEAL',  
      '515.127.4561',  
      TO_DATE('Dec 7, 2002', 'MON DD, YYYY'),  
      'SA_REP', 11000, 0.2, 100, 60, NULL);
```

1 rows inserted

- Verifique sua inclusão

114	Den	Raphealy	DRAPHEAL	515.127.4561	07-Dec-2002
-----	-----	----------	----------	--------------	-------------

Exercício 1 do Projeto

- DFo_6_4_1_Project
 - Banco de Dados da Oracle Baseball League
 - Usando operações DML para gerenciar tabelas de banco de dados: Inserindo registros



Alterando Dados em uma Tabela

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	-	100	90
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-1987	AD_ASST	4400	-	101	10

...

Atualize as linhas da tabela EMPLOYEES:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	-	-	60
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	-	100	60
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	-	100	60
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-1987	AD_ASST	4400	-	101	10

...

Sintaxe da Instrução UPDATE

É recomendável
deixar a instrução
UPDATE em uma
linha própria

- Modifique os valores existentes em uma tabela com a instrução UPDATE:

```
UPDATE      table  
SET         column = value [, column = value, ...]  
[WHERE      condition];
```

- Em geral, use a coluna de chave primária na cláusula WHERE a fim de identificar uma única linha para atualização
- Atualize mais de uma linha por vez (se necessário)

Atualizando Linhas em uma Tabela

- Os valores de uma linha ou linhas específicas serão modificados se você especificar a cláusula WHERE:

```
UPDATE copy_employees  
SET    department_id = 50  
WHERE  employee_id = 113;
```

- Os valores de todas as linhas da tabela serão modificados se você omitir a cláusula WHERE:

```
UPDATE copy_employees  
SET    department_id = 110;
```

Atualizando Linhas em uma Tabela

- Especifique SET column_name= NULL para atualizar um valor de coluna para NULL.

```
UPDATE    copy_employees
SET       department_id = NULL
WHERE     employee_id = 124;
```

Violando Restrições

```
UPDATE employees
SET department_id = 55
WHERE department_id = 110;
```

Error starting at line 1 in command:

```
UPDATE employees
SET department_id = 55
WHERE department_id = 110
```

Error report:

```
SQL Error: ORA-02291: integrity constraint (ORA1.EMP_DEPT_FK) violated - parent key not found
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause:      A foreign key value has no matching primary key value.
*Action:     Delete the foreign key or add a matching primary key.
```

- O departamento 55 não existe na tabela pai – DEPARTMENTS

Removendo uma Linha de uma Tabela

- Este slide mostra que o departamento Public Relations foi removido da tabela DEPARTMENTS (supondo-se que não haja restrições violadas nessa tabela)

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
70	Public RElations	204	2700
80	Sales	149	2500

**Linha excluída da
tabela DEPARTMENTS:**

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700



Instrução DELETE

- Você pode remover as linhas existentes de uma tabela usando a instrução DELETE:

```
DELETE [FROM] table  
[WHERE condition];
```



Excluindo Linhas de uma Tabela

- Linhas específicas serão excluídas se você incluir a cláusula WHERE:

```
DELETE FROM copy_departments  
WHERE department_name = 'Purchasing';
```

- Todas as linhas da tabela serão excluídas se você omitir a cláusula WHERE.

```
DELETE FROM copy_departments;
```


Violando Restrições

- Não é possível excluir uma linha que contenha uma chave primária usada como uma chave estrangeira em outra tabela

```
DELETE
FROM   departments
WHERE  department_id = 60;
```

Error starting at line 1 in command:

```
DELETE FROM departments
WHERE department_id = 60
```

Error report:

SQL Error: ORA-02292: integrity constraint (ORA1.JHIST_DEPT_FK) violated - child record found
02292. 00000 - "integrity constraint (%s.%s) violated - child record found"

*Cause: attempted to delete a parent key value that had a foreign dependency.

*Action: delete dependencies first then parent or disable constraint.

Cenário de Caso: Excluindo Linhas

É possível excluir todas as linhas de uma tabela, mas deixar a estrutura intacta?



Aluno



Corpo Docente

Com certeza! Você pode usar a instrução TRUNCATE para fazer isso.



Instrução TRUNCATE

- Remove todas as linhas de uma tabela, deixando-a vazia e a sua estrutura intacta
- É uma instrução DDL, e não uma instrução DML; não é possível desfazê-la facilmente
- Sintaxe:

```
TRUNCATE TABLE table_name;
```

- Exemplo:

```
TRUNCATE TABLE copy_employees;
```

- ** Consulte as observações sobre o truncamento das tabelas pai

Exercício 2 do Projeto

- DFo_6_4_2_Project
 - Banco de Dados da Oracle Baseball League
 - Usando operações DML para gerenciar tabelas de banco de dados: Atualizando e excluindo registros



Resumo

- Nesta lição, você deverá ter aprendido a:
 - Descrever a finalidade do DML
 - Listar as operações DML necessárias para gerenciar os dados de uma tabela de banco de dados:
 - INSERT
 - UPDATE
 - DELETE





ORACLE

Academy

