

FIAP

FIAP GRADUAÇÃO

Tecnologia em Análise e Desenvolvimento de Sistemas

Prof^o Ms. Alexandre Barcelos
profalexandre.barcelos@fiap.com.br

2019

Database Application Development

Prof^o Ms. Alexandre Barcelos
profalexandre.barcelos@fiap.com.br

2019



Manipulando Dados



Ao concluir esta lição, você será capaz de:

- **Descrever cada instrução DML (Data Manipulation Language)**
- **Inserir linhas em uma tabela**
- **Atualizar linhas em uma tabela**
- **Deletar linhas de uma tabela**
- **Controlar transações**

Objetivo

Nesta lição, você aprenderá a usar instruções DML para inserir linhas em uma tabela, atualizar as linhas existentes em uma tabela e deletar as linhas existentes de uma tabela. Você também aprenderá a controlar transações com as instruções COMMIT, SAVEPOINT e ROLLBACK.

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

Data Manipulation Language

A DML (Data Manipulation Language) é uma parte essencial de SQL. Para adicionar, atualizar ou deletar dados no banco de dados, execute uma instrução DML. Um conjunto de instruções DML que formam uma unidade lógica de trabalho é chamado de *transação*.

Considere um banco de dados de uma instituição bancária. Quando um cliente do banco transfere dinheiro da poupança para a conta corrente, a transação pode consistir em três operações distintas: diminuição da poupança, aumento da conta corrente e registro da transação no diário de transações. O servidor Oracle deve garantir a execução de todas as três instruções SQL para manter as contas com os saldos corretos. Quando algo impedir a execução de uma das instruções da transação, será necessário desfazer as outras.

Adicionando uma Nova Linha a uma Tabela

FIAP

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
110	Accounting	205	1700
190	Contracting		1700

70	Public Relations	100	1700
----	------------------	-----	------

Nova
linha

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
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700
70	Public Relations	100	1700

Adicionando uma Nova Linha a uma Tabela

O slide mostra a adição de um novo departamento à tabela DEPARTMENTS.

- Adicione novas linhas a uma tabela usando a instrução **INSERT**:

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

- Com esta sintaxe, apenas uma linha é inserida por vez.

1-8

Adicionando uma Nova Linha a uma Tabela (continuação)

É possível adicionar novas linhas a uma tabela executando a instrução **INSERT**.

Na sintaxe:

<i>table</i>	é o nome da tabela
<i>column</i>	é o nome da coluna da tabela a ser preenchida
<i>value</i>	é o valor correspondente da coluna

Observação: Esta instrução com a cláusula **VALUES** adiciona somente uma linha por vez a uma tabela.

- Insira uma nova linha com valores para cada coluna.
- Liste os valores na ordem default das colunas na tabela.
- Como alternativa, liste as colunas na cláusula INSERT.

```
INSERT INTO departments(department_id,  
                        department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);  
1 row created.
```

- Delimite os valores de caractere e data com aspas simples.

1-9

Adicionando uma Nova Linha a uma Tabela (continuação)

Como é possível inserir uma nova linha com valores para cada coluna, a lista de colunas não é necessária na cláusula INSERT. No entanto, se você não usar a lista de colunas, os valores deverão ser listados de acordo com a ordem default das colunas na tabela e um valor deverá ser fornecido para cada coluna.

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

Para fins de clareza, use a lista de colunas na cláusula INSERT.

Delimite os valores de caractere e data com aspas simples; não é recomendável delimitar valores numéricos com aspas simples.

Os valores numéricos não devem ser delimitados com aspas simples, já que poderá ocorrer conversão implícita dos valores numéricos designados às colunas com o tipo de dados

NUMBER se forem incluídas aspas simples.

- **Método implícito: omita a coluna da lista de colunas.**

```
INSERT INTO departments (department_id,  
                           department_name )  
VALUES (30, 'Purchasing');  
1 row created.
```

- **Método explícito: especifique a palavra-chave NULL na cláusula VALUES.**

```
INSERT INTO departments  
VALUES (100, 'Finance', NULL, NULL);  
1 row created.
```

1-10

Métodos para Inserir Valores Nulos	
Implícito	Omita a coluna da lista de colunas.
Explícito	Especifique a palavra-chave NULL na lista VALUES; especifique a string vazia (' ') na lista VALUES para strings de caracteres e datas.

Certifique-se de que a coluna de destino permita valores nulos verificando o status `Null?` com o comando `DESCRIBE` do *SQL Developer*.

O servidor Oracle impõe automaticamente todos os tipos de dados, faixas de dados e constraints de integridade de dados. Todas as colunas não listadas explicitamente obtêm um valor nulo na nova linha.

Erros comuns que podem ocorrer durante a entrada do usuário:

- Valor obrigatório ausente para uma coluna NOT NULL
- Constraint de exclusividade violada por valor duplicado
- Constraint de chave estrangeira violada
- Constraint CHECK violada

- Incompatibilidade de tipo de dados
- Valor muito extenso para caber na coluna

Alterando os Dados de uma Tabela

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_F
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	60	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	60	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	60	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

Atualize as linhas da tabela **EMPLOYEES**:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSIO
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	30	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	30	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	30	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

Alterando os Dados de uma Tabela

O slide mostra a alteração do número de departamento dos funcionários do departamento 60 para o departamento 30.

- **Modifique as linhas existentes com a instrução UPDATE:**

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

- **Atualize mais de uma linha por vez (se necessário).**

1-12

Atualizando Linhas

É possível modificar linhas existentes com a instrução UPDATE.

Na sintaxe:

<i>table</i>	é o nome da tabela
<i>column</i>	é o nome da coluna da tabela a ser preenchida
<i>value</i>	é o valor correspondente ou a subconsulta da coluna
<i>condition</i>	identifica as linhas a serem atualizadas e é composta de nomes de colunas, expressões, constantes, subconsultas e operadores de comparação

Para confirmar a operação de atualização, consulte a tabela para exibir as linhas atualizadas.

Para obter mais informações, consulte "UPDATE" no manual *Oracle Database SQL Reference*.

Observação: Em geral, use a chave primária para identificar uma única linha. O uso de outras colunas pode resultar na atualização inesperada de várias linhas. Por exemplo, identificar uma única linha da tabela EMPLOYEES por nome é perigoso, pois mais de um funcionário pode ter o mesmo nome.

- Uma ou mais linhas específicas são modificadas quando a cláusula **WHERE** é especificada:

```
UPDATE employees
SET    department id = 70
WHERE  employee_id = 113;
1 row updated.
```

- Se você omitir a cláusula **WHERE**, todas as linhas da tabela serão modificadas:

```
UPDATE    copy_emp
SET       department_id = 110;
22 rows updated.
```

1-13

Atualizando Linhas (continuação)

A instrução **UPDATE** modifica linhas específicas quando a cláusula **WHERE** é especificada. O exemplo do slide transfere o funcionário 113 (Popp) para o departamento 70.

Se você omitir a cláusula **WHERE**, todas as linhas da tabela serão modificadas.

```
SELECT last name, department id
```

LAST_NAME	DEPARTMENT_ID
King	110
Kochhar	110
De Haan	110
Hunold	110
Ernst	110
Lorentz	110

■■■
22 rows selected.

Observação: A tabela **COPY_EMP** tem os mesmos dados que a tabela **EMPLOYEES**.

Removendo uma Linha de uma Tabela

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
100	Finance		
50	Shipping	124	1500
60	IT	103	1400

Delete uma linha da tabela DEPARTMENTS:

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
50	Shipping	124	1500
60	IT	103	1400

Removendo uma Linha de uma Tabela

O exemplo do slide remove o departamento Finance da tabela DEPARTMENTS (pressupondo que não haja constraints definidas nessa tabela).

É possível remover as linhas existentes de uma tabela com a instrução DELETE:

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

1-15

Deletando Linhas

É possível remover linhas existentes com a instrução DELETE.

Na sintaxe:

table é o nome da tabela
condition identifica as linhas a serem deletadas e é composta de nomes de colunas, expressões, constantes, subconsultas e operadores de comparação

Observação: Se nenhuma linha for deletada, a mensagem "0 rows deleted" será exibida.

Para obter mais informações, consulte "DELETE" no manual *Oracle Database SQL Reference*.

- Se você usar a cláusula **WHERE**, as linhas específicas serão deletadas:

```
DELETE FROM departments
WHERE department_name = 'Finance';
1 row deleted.
```

- Se você omitir a cláusula **WHERE**, todas as linhas da tabela serão deletadas:

```
DELETE FROM copy_emp;
22 rows deleted.
```

1-16

Deletando Linhas (continuação)

É possível deletar linhas específicas usando a cláusula **WHERE** na instrução **DELETE**. O exemplo do slide deleta o departamento Finance da tabela **DEPARTMENTS**. Para confirmar a operação de deleção, exiba as linhas deletadas com a instrução **SELECT**.

```
SELECT *
FROM departments
WHERE department_name = 'Finance';
no rows selected.
```

Se você omitir a cláusula **WHERE**, todas as linhas da tabela serão deletadas. O segundo exemplo do slide deleta todas as linhas da tabela **COPY_EMP** porque nenhuma cláusula **WHERE** foi especificada.

Exemplo

Remova as linhas identificadas na cláusula **WHERE**.

```
DELETE FROM employees WHERE employee_id = 114;
1 row deleted.
```

```
DELETE FROM departments WHERE department_id IN (30, 40);
```

2 rows deleted.

Uma transação de banco de dados consiste em uma das seguintes instruções:

- **Instruções DML que constituem uma alteração consistente nos dados**
- **Uma instrução DDL**
- **Uma instrução DCL (Data Control Language)**

Transações de Banco de Dados

O servidor Oracle garante a consistência de dados com base em transações. As transações permitem mais flexibilidade e controle durante a alteração de dados e garantem a consistência de dados em caso de falha de processo do usuário ou falha do sistema.

As transações consistem em instruções DML que formam uma alteração consistente dos dados. Por exemplo, uma transferência de fundos entre duas contas deve incluir o débito em uma conta e o crédito em outra conta no mesmo valor. As duas ações deverão apresentar falha ou ser bem-sucedidas; o crédito não deverá ser submetido a commit sem o débito.

Tipos de Transação

Tipo	Descrição
DML (Data Manipulation Language)	Consiste em qualquer número de instruções DML que o servidor Oracle trata como uma entidade única ou uma unidade de trabalho lógica
(Data Definition Language)	Consiste em uma única instrução DDL
DCL (Data Control Language)	Consiste em uma única instrução DCL

- **Começam quando a primeira instrução SQL DML é executada**
- **Terminam com um destes eventos:**
 - Uma instrução COMMIT ou ROLLBACK é executada.
 - Uma instrução DDL ou DCL é executada (commit automático).
 - O usuário sai do SQL Developer.
 - Ocorre uma falha do sistema.

Quando uma Transação Começa e Termina?

Uma transação começa quando a primeira instrução DML é encontrada e termina quando uma destas ações ocorre:

- Uma instrução COMMIT ou ROLLBACK é executada.
- Uma instrução DDL, como CREATE, é executada.
- Uma instrução DCL é executada.
- O usuário sai do *SQL Developer*.
- Ocorre uma falha de máquina ou do sistema.

Após o término de uma transação, a próxima instrução SQL executável inicia automaticamente a transação seguinte.

Uma instrução DDL ou DCL é submetida a commit automaticamente e, portanto, encerra uma transação de forma implícita.



Com as instruções COMMIT e ROLLBACK, é possível:

- **Garantir a consistência de dados**
- **Visualizar alterações de dados antes de torná-las permanentes**
- **Agrupar operações relacionadas logicamente**

COMMIT e ROLLBACK

As instruções COMMIT e ROLLBACK permitem controlar as alterações permanentes nos dados.

- Um commit automático ocorre nas seguintes circunstâncias:
 - Uma instrução DDL é executada
 - Uma instrução DCL é executada
 - Saída normal do *SQL Developer*, sem a execução implícita de instruções `COMMIT` ou `ROLLBACK`
- Um rollback automático ocorre em decorrência do encerramento anormal do *SQL Developer* ou de uma falha do sistema.

Processamento de Transação Implícita

Status	Circunstâncias
Automatic commit	A instrução DDL ou DCL é executada. O <i>iSQL*Plus</i> foi encerrado normalmente, sem a execução explícita dos comandos <code>COMMIT</code> ou <code>ROLLBACK</code> .
Automatic rollback	Encerramento anormal do <i>iSQL*Plus</i> ou falha do sistema.

Observação: Um terceiro comando está disponível no *SQL Developer*. É possível alternar o comando `AUTO COMMIT` entre os estados ativado e desativado. Se ele for *ativado*, cada instrução DML será submetida a commit logo após sua execução. Não é possível fazer rollback das alterações. Se ele for *desativado*, a instrução `COMMIT` ainda poderá ser executada explicitamente. Além disso, a instrução `COMMIT` será executada quando uma instrução DDL for executada ou quando você sair do *SQL Developer*.

Processamento de Transação Implícita (continuação)

Falhas do Sistema

Quando uma transação for interrompida por uma falha do sistema, será feito rollback automaticamente de toda a transação. Isso impede que o erro cause alterações indesejadas nos dados e retorna as tabelas para o estado em que se encontravam no momento do último commit. Dessa forma, o servidor Oracle protege a integridade das tabelas.

No *SQL Developer*, para sair normalmente da sessão, você deve clicar no botão Exit. No *SQL*Plus*, para executar essa mesma ação, você deve digitar o comando `EXIT` no prompt. O fechamento da janela é interpretado como uma saída anormal.



Nesta lição, você aprendeu a usar as seguintes instruções:

Function	Descrição
INSERT	Adiciona uma nova linha à tabela
UPDATE	Modifica as linhas existentes da tabela
DELETE	Remove as linhas existentes da tabela
COMMIT	Torna permanentes todas as alterações pendentes
ROLLBACK	Descarta todas as alterações de dados pendentes

Sumário

Nesta lição, você aprendeu a manipular dados no banco de dados Oracle com as instruções INSERT, UPDATE e DELETE, bem como controlar alterações nos dados com as instruções COMMIT, SAVEPOINT e ROLLBACK.

O servidor Oracle garante uma view consistente dos dados em todos os momentos.

Bibliografia Utilizada

FIAP

Database SQL Language Reference: <http://docs.oracle.com/database/121/SQLRF/toc.htm>

Manuais Oracle – Oracle Database 12c: SQL Workshop I/II

*Esta apresentação possui material de referência com propriedade da Oracle.
Copyright © 2004, Oracle. Todos os direitos reservados.*





Este exercício aborda os seguintes tópicos:

- **Inserção de linhas nas tabelas**
- **Atualização e deleção de linhas na tabela**
- **Controle de transações**

Exercício: Visão Geral

Neste exercício, você adicionará linhas à tabela MY_EMPLOYEE, atualizará e deletará dados na tabela e controlará as transações.

Exercício

O departamento de recursos humanos deseja criar instruções SQL para inserir, atualizar e deletar dados de funcionários. Como protótipo, use a tabela MY_EMPLOYEE antes de fornecer as instruções ao departamento de recursos humanos.

Insira dados na tabela MY_EMPLOYEE.

1. Execute a instrução no script lab_08_01.sql para criar a tabela MY_EMPLOYEE a ser usada no exercício.
2. Descreva a estrutura da tabela MY_EMPLOYEE para identificar os nomes de colunas.

Name	Null?	Type
ID	NOT NULL	NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)

3. Crie uma instrução INSERT para *adicionar a primeira linha* de dados à tabela MY_EMPLOYEE usando estes dados de amostra. Não liste as colunas na cláusula INSERT. *Não informe todas as linhas ainda.*

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audrey	aropebur	1550

4. Preencha a tabela MY_EMPLOYEE com a segunda linha de dados de amostra da lista anterior. Desta vez, liste as colunas explicitamente na cláusula INSERT.
5. Confirme a adição à tabela.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860

Exercício (continuação)

- Preencha a tabela MY_EMPLOYEE com a terceira linha de dados de amostra da lista anterior.
- Preencha a tabela MY_EMPLOYEE com a quarta linha de dados de amostra da lista anterior.
- Confirme as adições à tabela

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750

- Torne as adições de dados permanentes.

Atualize e delete dados na tabela MY_EMPLOYEE.

- Altere o sobrenome do funcionário 3 para Drexler.
- Altere o salário de todos os funcionários com salário inferior a US\$ 900 para US\$ 1.000.
- Verifique as alterações na tabela.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
2	Dancs	Betty	bdancs	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

- Delete Betty Dancs da tabela MY_EMPLOYEE.
- Confirme as alterações na tabela.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Exercício (continuação)

15. Submeta todas as alterações pendentes a commit.

Controle a transação de dados na tabela **MY_EMPLOYEE**.

16. Preencha a tabela com a última linha dos dados de amostra.

17. Confirme a adição à tabela.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

18. Marque um ponto intermediário no processamento da transação.

19. Esvazie a tabela inteira.

20. Confirme se a tabela está vazia.

21. Descarte a operação DELETE mais recente sem descartar a operação INSERT anterior.

22. Confirme se a nova linha permanece intacta.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

23. Torne a adição de dados permanente.

Script lab_08_01.sql

CREATE TABLE my_employee

(id NUMBER(4) CONSTRAINT my_employee_id_nn NOT NULL,

last_name VARCHAR2(25),

first_name VARCHAR2(25),

userid VARCHAR2(8),

salary NUMBER(9,2));