

# Desing e Desenvolvimento de Banco de Dados

## Manipulação de Dados (DML)



Luciano Melo  
*profluciano.melo@fiap.com.br*



## Data Manipulation Language

- Os comandos que alteram informações no banco de dados fazem parte da linguagem SQL chamada de DML – (Linguagem de Manipulação de Dados)
- Uma instrução DML é executada quando você:
  - Adiciona novas linhas a uma tabela (*INSERT*)
  - Modifica as linhas existentes de uma tabela (*UPDATE*)
  - Remove as linhas existentes de uma tabela (*DELETE*)

## Adicionando uma Nova Linha a uma Tabela

DEPARTMENTS

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

Nova  
linha

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

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

## Sintaxe da Instrução INSERT

- 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.

## Sintaxe da Instrução INSERT

- **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.**

## Inserindo Novas Linhas

- 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.

## 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	

## Sintaxe da Instrução UPDATE

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

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

- **Pode se atualizar mais de uma linha por vez (se necessário).**
- **A quantidade de linhas alteradas depende do filtro utilizado. Se não existir filtro, todas as linhas da tabela são alteradas**



## Atualizando Linhas em uma Tabela

- 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.
```

## 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

## Instrução DELETE

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

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

## Deletando Linhas de uma Tabela

- 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.
```

## Instrução TRUNCATE

- Remove todas as linhas de uma tabela, esvaziando a tabela e mantendo a estrutura intacta.
- É muito mais rápido que remover (usando delete) todas as linhas da tabela.
- É uma instrução DDL (Data Definition Language), e não DML; não pode ser desfeita facilmente
- Sintaxe:

```
TRUNCATE TABLE table_name;
```

- Exemplo:

```
TRUNCATE TABLE copy_emp;
```

## Transações de Banco de Dados

- Uma transação consiste em um conjunto de instruções DML que formam uma unidade lógica de trabalho.
- Uma transação pode iniciar automaticamente, quando o primeiro comando DML é executado ou então explicitamente, como no caso do SQL Server (begin transaction)
- Uma transação termina da seguinte maneira:
  - ✓ **COMMIT:** Efetiva todas as alterações feitas pela transação
  - ✓ **ROLLBACK:** Desfaz todas as alterações feitas pela transação.

## Estado dos Dados antes de COMMIT OU ROLLBACK

- É possível recuperar o estado anterior dos dados.
- O usuário atual pode verificar os resultados das operações DML usando a instrução `SELECT`.
- Outros usuários *não podem* visualizar os resultados das instruções DML executadas pelo usuário atual.
- As linhas afetadas são *bloqueadas*; outros usuários não podem alterar os dados nessas linhas.

## Efetivando as Alterações da Transação

- Efetue as alterações:

```
DELETE FROM employees
WHERE  employee_id = 99999;
1 row deleted.

INSERT INTO departments
VALUES (290, 'Corporate Tax', NULL, 1700);
1 row created.
```

- Submeta as alterações a commit:

```
COMMIT;
Commit complete.
```



## Estado dos Dados após COMMIT

- **As alterações de dados tornam-se permanentes no banco de dados.**
- **O estado anterior dos dados é perdido permanentemente.**
- **Todos os usuários podem visualizar os resultados.**
- **Os bloqueios nas linhas afetadas são liberados; essas linhas estão disponíveis para manipulação por outros usuários.**

## Desfazendo as Alterações da Transação

**Descarte todas as alterações pendentes com a instrução ROLLBACK:**

- **As alterações de dados são desfeitas.**
- **O estado anterior dos dados é restaurado.**
- **Os bloqueios nas linhas afetadas são liberados.**

```
DELETE FROM copy_emp;  
22 rows deleted.  
ROLLBACK ;  
Rollback complete.
```

## Exemplo

```
DELETE FROM test;  
25,000 rows deleted.
```

```
ROLLBACK;  
Rollback complete.
```

```
DELETE FROM test WHERE id = 100;  
1 row deleted.
```

```
SELECT * FROM test WHERE id = 100;  
No rows selected.
```

```
COMMIT;  
Commit complete.
```



*profluciano.melo@fiap.com.br*



# Atividade

*profluciano.melo@fiap.com.br*