

FIAP GRADUAÇÃO

# BUILDING RELATIONAL DATABASE

Prof. Diogo Alves  
profdiogo.alves@fiap.com.br

**COMANDOS DML (SQL)**  
**(DML→LINGUAGEM DE MANIPULAÇÃO DE DADOS)**

- Objetivo
- Conceitos referentes a linguagem de manipulação de dados
- Revisão dos Conceitos
- Exercícios

# Objetivos

- ❑ Aplicar os conceitos da linguagem SQL durante a implementação do banco de dados

# Conteúdo Programático referente a esta aula

- ☐ Linguagem de definição de dados
  - ☐ DML (Insert, Update, Delete)
  - ☐ Exercícios

# Linguagem SQL

## SQL: Structured Query Language

### (Linguagem Estruturada de Consulta)



É uma **linguagem de definição, manipulação e controle de banco de dados**.

As instruções de manipulação de dados, pertencem a divisão DML, da linguagem SQL.

**DML: Data Manipulation Language**, ou linguagem para manipulação de dados.

As operações de manipulação de dados são:

- Inclusão de dados nas tabelas
- Alteração ou atualização de dados
- Exclusão de dados

Também encontramos o subconjunto denominado DCL (Data Control Language), utilizado para controlar transações que ocorrem no bando de dados.

# Linguagem SQL

SQL Structured Query Language

(Linguagem Estruturada de Consulta)



## DML: Linguagem de Manipulação de Dados

Utilizada para manipular os dados de uma tabela. Utilizamos para inserir, atualizar e apagar dados.

**Exemplos: INSERT, UPDATE, DELETE.**

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



### COMANDO INSERT

Permite a inclusão de um registro em uma tabela.

```
INSERT INTO <nome-tabela>  
    [ (<nome-coluna>, [<nome-coluna>], ...) ]
```

#### VALUES

```
(<conteúdo>, [<conteúdo>], ...);
```

Onde:

**<nome-tabela>** - Tabela onde será efetuada a inserção da linha.

**<nome-coluna>** - Colunas que terão os valores atribuídos.

**<conteúdo>** - Relação de valores a serem incluídos.



# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



### COMANDO UPDATE

Permite a alteração do conteúdo de um ou mais campos pertencentes a um ou mais registros em uma tabela.

**UPDATE** <nome-tabela>

**SET** <nome-coluna> = <novo conteúdo para o campo>

[,<nome-coluna> = <novo conteúdo para o campo>]

[**WHERE** <condição>]

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



### COMANDO UPDATE

**UPDATE** <nome-tabela>

**SET** <nome-coluna> = <novo conteúdo para o campo>

[,<nome-coluna> = <novo conteúdo para o campo>]

[**WHERE** <condição>]

Onde:

<nome-tabela> - Tabela onde será efetuada a atualização.

<nome-coluna> - Colunas que terá seu valor atualizado.

<conteúdo> - Novo conteúdo para a coluna.

<condição> - Critério que indicará quais linhas serão atualizadas.

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



### COMANDO DELETE

Permite a exclusão de um ou mais registros em uma tabela.

```
DELETE FROM <nome-tabela>  
[WHERE <condição>]
```

Onde:

**<nome-tabela>** - Tabela onde será excluída a linha.

**<condição>** - Critério que indicará quais linhas serão excluídas.

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL

### Possíveis situações de ERROS durante a execução dos comandos DML



#### Comando INSERT

- ☐ Violação de Chave Primária (**Primary Key**);
- ☐ Violação de **UNIQUE** constraint;
- ☐ Valor informado excede o tamanho de coluna ;
- ☐ Tipo de dado incompatível com a coluna (caracteres em coluna);
- ☐ Violação de **Check** constraint;
- ☐ Violação de chave estrangeira (**Foreign Key**);
- ☐ Violação de **not null**.

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL

### Possíveis situações de ERROS durante a execução dos comandos DML



#### Comando UPDATE

- ☐ Violação de Chave Primária (**Primary Key**);
- ☐ Violação de **UNIQUE** constraint;
- ☐ Valor informado excede o tamanho de coluna ;
- ☐ Tipo de dado incompatível com a coluna (caracteres em coluna);
- ☐ Violação de **Check** constraint;
- ☐ Violação de chave estrangeira (**Foreign Key**);
- ☐ Violação de **not null**.

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL

Possíveis situações de ERROS durante a execução dos comandos DML



### Comando DELETE

- ❑ Violação de chave estrangeira (**Foreign Key**);

## Manipulação dos Dados utilizando a linguagem SQL

Possíveis situações de ERROS durante a execução dos comandos DML



### CONSIDERAÇÕES: Comando INSERT/UPDATE/DELETE

- ☐ Quando um erro ocorrer o registro NÃO SERÁ INSERIDO/ALTERADO/EXCLUÍDO na base.
- ☐ O comando é efetivado somente após o **COMMIT**. É possível desfazer a operação através do **ROLLBACK**. Obs: após o COMMIT dos comandos, não será possível desfazer as operações (nem mesmo utilizando o ROLLBACK);
- ☐ Em alguns SGBDS, as ferramentas podem realizar COMMIT automático (SQL Server). Em programação pode ser necessário que se use os comandos `commit` ou `rollback`.

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL

### Possíveis situações de ERROS durante a execução dos comandos DML



#### PERMISSÕES INSERT/UPDATE/DELETE

- ❑ Para realizar essas operações o usuário deve ser o proprietário da tabela, isto é, deve ter criado a tabela em seu esquema, ou possuir privilégios para realizar os comandos;
- ❑ Esquema ou, em inglês, schema, é uma coleção de objetos dentro de um banco de dados. Serve para agrupar objetos que se referem a áreas de interesse comum dentro de um modelo de dados.



# Manipulação dos Dados utilizando a linguagem SQL



## DATA no ORACLE

Formato padrão de data no Oracle é: dia, mês e ano com dois dígitos.

Para alterar o formato padrão usamos:

```
ALTER SESSION SET NLS_DATE_FORMAT = [date_format]
```

### Exemplo:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'dd/mm/yyyy hh24:mi:ss';
```

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY HH:MI';
```

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY HH24:MI:SS.sssss';
```

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



### DATA no ORACLE

O date format pode assumir os seguintes valores:

YY	= ano com 2 dígitos
YYYY	= ano com 4 dígitos
NM	= o número do mês
MONTH	= o nome completo do mês
MON	= o nome abreviado do mês. Exemplo: jan, fev, mar
DDD	= o dia do ano
DD	= o dia do mês
D	= o dia da semana
DAY	= o nome do dia
HH	= a hora do dia, no formato de 12 horas
HH24	= a hora do dia, no formato 24 horas
MI	= os minutos
SS	= os segundos
SS.sssss	= segundos com milissegundos
SS.FF ou SSXFF	= segundos com frações de segundos

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



## EXEMPLOS: INSERT

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE INSERÇÃO DE DADOS EM UMA TABELA

T_SIP_DEPTO		
P *	cd_depto	NUMBER (3)
U *	nm_depto	VARCHAR2 (30)
*	sg_depto	CHAR (3)
PK_SIP_DEPTO (cd_depto)		
UN_SIP_DEPTO_NOME (nm_depto)		

#### Atenção:

Neste exemplo declaramos o nome de todas as colunas no comando INSERT.

O valores para cada coluna deve ser informado na mesma ordem da declaração, separados por vírgula.

Dados do tipo caractere deve ser informado entre aspas simples.

```
-- INSERIR UMA LINHA/REGISTRO NA TABELA
INSERT INTO T_SIP_DEPTO (CD_DEPTO, NM_DEPTO, DS_SIGLA) -- NOMES DAS COLUNAS/CAMPO DA TABELA
VALUES (1, 'FINANCEIRO', 'FIN'); -- VALORES QUE SERÃO ARMZENADOS NAS COLUNAS/CAMPOS

COMMIT;
```

#### Observação:

Ao final de cada **INSERT** é importante aplicar o comando COMMIT, caso contrário, ao fechar a ferramenta as informações inseridas serão perdidas.



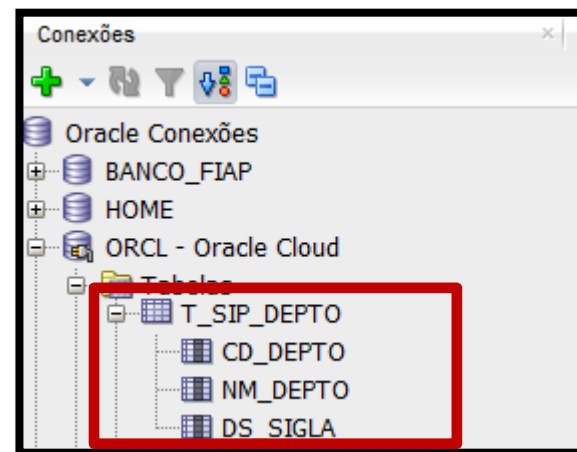
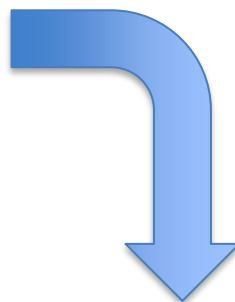
# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE INSERÇÃO DE DADOS EM UMA TABELA

T_SIP_DEPTO		
P *	cd_depto	NUMBER (3)
U *	nm_depto	VARCHAR2 (30)
*	sg_depto	CHAR (3)
PK_SIP_DEPTO (cd_depto)		
UN_SIP_DEPTO_NOME (nm_depto)		



```
-- INSERIR UMA LINHA/REGISTRO NA TABELA
INSERT INTO T_SIP_DEPTO VALUES (2, 'MARKETING', 'MKT');
-- VALORES QUE SERÃO ARMZENADOS NAS COLUNAS/CAMPOS

COMMIT;
```

### Observação:

Podemos omitir a declaração do nome das colunas, no momento da inserção, desde que seja respeitado a ordem física das colunas ao inserir os respectivos valores.

Para auxiliar, basta expandir a lista de colunas da tabela, no SQL Developer, conforme imagem a direita e informar os valores na ordem visualizada.



# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### VISUALIZANDO O DADOS INSERIDOS NA TABELA DEPARTAMENTO

Utilize o comando SELECT para realizar a consulta simples na tabela Departamento.

Neste exemplo, estamos consultando todas as colunas/campos (representado pelo asterisco \* na linha de comando) da tabela DEPARTAMENTO.

```
-- VISUALIZANDO A TABELA DEPARTAMENTO
SELECT * FROM T_SIP_DEPTO;
```

Resultado da consulta:

CD_DEPTO	NM_DEPTO	DS_SIGLA
1	FINANCEIRO	FIN
2	MARKETING	MKT

**Nota: Veremos comando SELECT em detalhes em nossas próximas aulas.**



# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

EXEMPLO DE INSERÇÃO DE DADOS EM UMA TABELA, CONTENDO CONTEÚDO DATA/HORA.

T_SIP_PROJETO		
P *	cd_projeto	NUMBER (5)
U *	nm_projeto	VARCHAR2 (30)
*	dt_inicio	DATE
	dt_termino	DATE
PK_SIP_PROJETO (cd_projeto)		
UN_SIP_PROJETO_NOME (nm_projeto)		

TO DATE(<data e/ou hora>, <formato>) =>

função do Oracle que converte o parâmetro informado (cadeia de caracteres) em uma data no formato especificado .

(FUNÇÃO utilizada em PL/SQL → Procedure Language/Structured Query Language)

```
-- INSERÇÃO CONTENDO DATA/HORA
INSERT INTO T_SIP_PROJETO VALUES (1, 'PROJETO ABC', -- VALORES PARA AS COLUNAS CODIGO E NOME DO PROJETO
    TO_DATE('01/02/2018','DD/MM/YYYY'), -- DATA ARMAZENADA NA DATA DE INICIO DO PROJETO
    TO_DATE('15/03/2020','DD/MM/YYYY') ); -- DATA ARMAZENADA NA DATA TERMINO DO PROJETO

COMMIT;
```

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

EXEMPLO DE INSERÇÃO DE DADOS EM UMA TABELA, CONTENDO CONTÉÚDO NUMÉRICO DECIMAL.

### T\_SIP\_FUNCIONARIO

P *	nr_matricula	NUMBER (5)
F *	cd_depto	NUMBER (3)
*	nm_funcionario	VARCHAR2 (50)
*	dt_nascimento	DATE
*	dt_admissao	DATE
*	ds_endereco	VARCHAR2 (100)
*	vl_salario	NUMBER (7,2)

PK\_SIP\_FUNCIONARIO (nr\_matricula)

FK\_SIP\_FUNC\_DEPTO (cd\_depto)

### Observação:

Utilizamos o ponto decimal.

Não armazenamos a formatação do banco. A máscara para formatação poderá ser aplicada no momento da consulta aos dados no banco ou na apresentação dos dados ao usuário, através da linguagem de programação utilizada na camada de apresentação dos dados.

```
-- INSERINDO CONTENDO VALORES NUMERICOS DECIMAIS
INSERT INTO T_SIP_FUNCIONARIO VALUES (12345, 1, 'JOAO DA SILVA XAVIER',
TO_DATE('18/03/1985','DD/MM/YYYY'),
TO_DATE('21/07/2012','DD/MM/YYYY'),
'AVENIDA PAULISTA, 1900 - 10. ANDAR',
3189.52 ); -- UTILIZAMOS O PONTO DECIMAL
```

COMMIT;



# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### VISUALIZANDO O DADOS INSERIDOS NA TABELA FUNCIONARIO

```
-- VISUALIZANDO A TABELA FUNCIONARIO
SELECT * FROM T_SIP_FUNCIONARIO;
```

Resultado da consulta:

NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDERECO	VL_SALARIO
12345	1	JOAO DA SILVA XAVIER	18/03/85	21/07/12	AVENIDA PAULISTA, 1900 - 10. ANDAR	3189,52

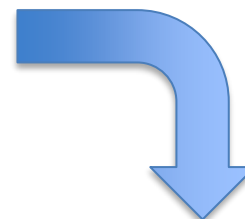
# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

EXEMPLO DE INSERÇÃO DE DADOS EM UMA TABELA, COM VALORES NULOS PARA CAMPOS OPCIONAIS.

T_SIP_PROJETO		
P *	cd_projeto	NUMBER (5)
U *	nm_projeto	VARCHAR2 (30)
*	dt_inicio	DATE
	dt_termino	DATE
PK_SIP_PROJETO (cd_projeto)		
UN_SIP_PROJETO_NOME (nm_projeto)		



```
-- INSERINDO VALORES NULOS PARA CAMPOS OPCIONAIS
-- MÉTODO IMPLÍCITO - OMITA A COLUNA DA LISTA DE COLUNAS
INSERT INTO T_SIP_PROJETO (CD_PROJETO, NM_PROJETO, DT_INICIO)
VALUES (2, 'PROJETO DEF', -- VALORES PARA AS COLUNAS CODIGO E NOME DO PROJETO
        TO_DATE('14/03/2019', 'DD/MM/YYYY')); -- DATA ARMAZENADA NA DATA DE INICIO DO PROJETO
-- OBSERVE QUE OMITIMOS O CAMPO DT_TERMINO NA DECLARAÇÃO DAS COLUNAS, DESTA FORMA,
-- NÃO INFORMAMOS A DATA NA LISTA DE VALORES DA CLAUSULA VALUES PARA INSERÇÃO
COMMIT;

-- MÉTODO EXPLÍCITO - ESPECIFIQUE A PALAVRA-CHAVE NULL, NA CLAUSULA VALUES
INSERT INTO T_SIP_PROJETO
VALUES (3, 'PROJETO GHI', -- VALORES PARA AS COLUNAS CODIGO E NOME DO PROJETO
        TO_DATE('14/03/2019', 'DD/MM/YYYY'), -- DATA ARMAZENADA NA DATA DE INICIO DO PROJETO
        NULL ); -- ENVIO DE NULO PARA O CAMPO OPCIONAL DATA TERMINO DO PROJETO
COMMIT;
```

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### VISUALIZANDO O DADOS INSERIDOS NA TABELA PROJETO

```
-- VISUALIZANDO A TABELA PROJETO  
SELECT * FROM T_SIP_PROJETO;
```

Resultado da consulta:

CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO
1	PROJETO ABC	01/02/18	15/03/20
2	PROJETO DEF	14/03/19	(null)
3	PROJETO GHI	14/03/19	(null)

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### COPIANDO LINHAS DE UMA TABELA PARA OUTRA UTILIZANDO O COMANDO INSERT E COMANDO SELECT

Para realizarmos este teste, será necessário criar a tabela abaixo, para que possamos demonstrar a cópia dos dados.

```
-- CRIAR UMA TABELA PARA REALIZAR A CÓPIA
CREATE TABLE T_SIP_COPY_PROJETO_FIN
(
    CD_PROJETO          NUMBER(5)          NOT NULL ,
    NM_PROJETO          VARCHAR2(40)       NOT NULL ,
    DT_INICIO           DATE               NOT NULL ,
    DT_TERMINO          DATE               NOT NULL
);

-- CHAVE PRIMARIA DA TABELA COPIA PROJETOS FINALIZADOS
ALTER TABLE T_SIP_COPY_PROJETO_FIN
ADD CONSTRAINT PK_SIP_COPY_PROJETO_FIN
PRIMARY KEY (CD_PROJETO);
```

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### COPIANDO LINHAS DE UMA TABELA PARA OUTRA UTILIZANDO O COMANDO INSERT E COMANDO SELECT (INSERÇÃO A PARTIR DE UMA SUBCONSULTA)

Abaixo temos o comando INSERT e SELECT necessários para realizar a cópia dos dados de uma tabela para outra.

Neste exemplo, a ideia é copiar apenas os dados referentes aos projetos finalizados, ou seja, não será copiada a tabela inteira.

```
-- REALIZANDO A COPIA DOS DADOS DE UMA TABELA PARA OUTRA
-- UTILIZANDO O COMANDO INSERT E O COMANDO SELECT
INSERT INTO T_SIP_COPY_PROJETO_FIN (CD_PROJETO, NM_PROJETO, DT_INICIO, DT_TERMINO)
    (SELECT CD_PROJETO, NM_PROJETO, DT_INICIO, DT_TERMINO
     FROM T_SIP_PROJETO
     WHERE DT_TERMINO IS NOT NULL);

-- OU
INSERT INTO T_SIP_COPY_PROJETO_FIN
    (SELECT *
     FROM T_SIP_PROJETO
     WHERE DT_TERMINO IS NOT NULL);
```

#### Atenção:

No 1º. Exemplo declaramos todas as colunas no comando INSERT e no comando SELECT, é importante ressaltar, que as colunas das tabelas sejam equivalentes e estejam na mesma ordem.

#### Atenção:

No 2º. Exemplo omitimos as colunas no comando INSERT e no comando SELECT, isto é possível pois as colunas das duas tabelas estão na mesma ordem.

#### Atenção:

A cláusula WHERE no comando SELECT, permite criar filtros durante a consulta. O operador especial IS NOT NULL, permite validar se o conteúdo da coluna não é nulo. Mais adiante veremos o comando SELECT em detalhes.

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



## EXEMPLOS: UPDATE

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE ATUALIZAÇÃO DE DADOS EM UMA TABELA

Conteúdo da tabela antes da atualização:

NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDERECO	VL_SALARIO
12345	1	JOAO DA SILVA XAVIER	18/03/85	21/07/12	AVENIDA PAULISTA, 1900 - 10. ANDAR	3189,52

COMANDO UPDATE PARA REALIZAR A ALTERAÇÃO:

```
-- ATUALIZAR DUAS OU MAIS COLUNAS EM UM COMANDO DE ATUALIZACAO
-- ALTERAR O NOME E A DATA DE NASCIMENTO DE UM FUNCIONARIO ESPECIFICO
UPDATE T_SIP_FUNCIONARIO
  SET NM_FUNCIONARIO = 'JOÃO DA SILVA XAVIER DE CAMARGO' ,
      DT_NASCIMENTO = TO_DATE ('18/05/1985', 'DD/MM/YYYY')
 WHERE NR_MATRICULA = 12345;
```

Conteúdo da tabela depois da execução da atualização:

NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDERECO	VL_SALARIO
12345	1	JOÃO DA SILVA XAVIER DE CAMARGO	18/05/85	21/07/12	AVENIDA PAULISTA, 1900 - 10. ANDAR	3189,52

### ATENÇÃO:

A clausula WHERE no comando UPDATE, garante que serão modificadas apenas as linhas/registros que atendem a condição.

Em um UPDATE sem WHERE, todas as linhas da tabela serão modificadas.



# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE ATUALIZAÇÃO DE DADOS EM UMA TABELA

Conteúdo da tabela antes da atualização:

NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDERECO	VL_SALARIO
12345	1	JOÃO DA SILVA XAVIER DE CAMARGO	18/05/85	21/07/12	AVENIDA PAULISTA, 1900 - 10. ANDAR	3189,52

COMANDO UPDATE PARA REALIZAR A ALTERAÇÃO:

```
-- ATUALIZAR O SALARIO DOS FUNCIONARIOS
-- TODOS OS FUNCIONARIOS IRÃO RECEBER UM AUMENTO DE 5%
UPDATE T_SIP_FUNCIONARIO
    SET VL_SALARIO = VL_SALARIO * 1.05;
-- NÃO TEMOS O WHERE (CONDICAO), POIS O AUMENTO SERÁ A TODOS FUNCIONARIOS
```

Conteúdo da tabela depois da execução da atualização:

NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDERECO	VL_SALARIO
12345	1	JOÃO DA SILVA XAVIER DE CAMARGO	18/05/85	21/07/12	AVENIDA PAULISTA, 1900 - 10. ANDAR	3349

### ATENÇÃO:

Neste exemplo utilizamos um UPDATE sem a clausula WHERE, pois a ideia é conceder o aumento de 5% no salário de todos os funcionários.





# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE ATUALIZAÇÃO UTILIZANDO SUBCONSULTA

Abaixo temos o comando UPDATE e SELECT necessários para alterar o conteúdo de um campo/coluna de uma tabela, a partir do conteúdo de outra tabela.

```
-- ATUALIZACAO UTILIZANDO SUBCONSULTA
-- ALTERAR A DATA TERMINO DOS PROJETOS QUE ESTÃO ARMAZENADOS NA TABELA PROJETO,
-- UTILIZANDO A DATA TERMINO QUE ESTÁ ARMAZENADO NO PROJETO FINALIZADO DE
-- CÓDIGO IGUAL A 1, QUE ESTÁ NA TABELA COPIA DOS PROJETOS FINALIZADOS
UPDATE T_SIP_PROJETO
  SET DT_TERMINO =
      (SELECT DT_TERMINO
       FROM T_SIP_COPY_PROJETO_FIN
       WHERE CD_PROJETO = 1);
```

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE ATUALIZAÇÃO UTILIZANDO SUBCONSULTA

Observe o conteúdo de cada tabela, antes da execução do comando apresentado na página anterior.

**TABELA: PROJETO**

T_SIP_PROJETO				
Colunas	Dados	Model	Constraints	Concessões
Classificar..	Filtro:			
CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO	
1	1 PROJETO ABC	01/02/18	15/03/20	
2	2 PROJETO DEF	14/03/19	(null)	
3	3 PROJETO GHI	14/03/19	(null)	

**TABELA: COPY\_PROJETO\_FIN**

T_SIP_COPY_PROJETO_FIN				
Colunas	Dados	Model	Constraints	Concessões
Classificar..	Filtro:			
CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO	
1	1 PROJETO ABC	01/02/18	15/03/20	

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE ATUALIZAÇÃO UTILIZANDO SUBCONSULTA

Observe o conteúdo da tabela PROJETO, após a execução do comando de alteração.

Observe que todos os projetos, foram modificados para a mesma data término do projeto finalizado que está armazenado na tabela COPY\_PROJETO\_FIN.

#### TABELA: PROJETO

	CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO
1	1	PROJETO ABC	01/02/18	15/03/20
2	2	PROJETO DEF	14/03/19	15/03/20
3	3	PROJETO GHI	14/03/19	15/03/20

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL



## EXEMPLOS: DELETE

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE DELEÇÃO (ELIMINAÇÃO) DE DADOS EM UMA TABELA

Conteúdo da tabela antes da deleção:

NR_MATRICULA	CD_DEPTO	NM_FUNCIONARIO	DT_NASCIMENTO	DT_ADMISSAO	DS_ENDERECO	VL_SALARIO
12345	1	JOÃO DA SILVA XAVIER DE CAMARGO	18/05/85	21/07/12	AVENIDA PAULISTA, 1900 - 10. ANDAR	3189,52

COMANDO DELETE PARA REALIZAR A DELEÇÃO/ELIMINAÇÃO:

```
-- ELIMINAR OS FUNCIONARIOS DO DEPTO FINANCEIRO (COD=1) E QUE
-- FORAM ADMITIDOS A PARTIR DE 2012
DELETE FROM T_SIP_FUNCIONARIO
WHERE CD_DEPTO = 1 AND
      DT_ADMISSAO >= TO_DATE ('01/01/2012', 'DD/MM/YYYY');
```

Conteúdo da tabela depois da execução da deleção:

NR_MAT...	CD_DEP...	NM_FUN...	DT_NAS...	DT_ADM...	DS_END...	VL_SAL...

### ATENÇÃO:

A clausula WHERE no comando DELETE, garante que serão deletadas apenas as linhas/registros que atendem a condição.

Em um DELETE sem WHERE, todas as linhas da tabela serão APAGADAS.



# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE DELEÇÃO/ELIMINAÇÃO UTILIZANDO SUBCONSULTA

Abaixo temos o comando DELETE e SELECT necessários para apagar um registro/linha, a partir do conteúdo de outra tabela.

```
-- DELEÇÃO COM SUBCONSULTA
-- ELIMINAR O PROJETO NA TABELA PROJETO, CORRESPONDENTE AO PROJETO QUE FOI
-- COPIADO PARA A TABELA COPY_PROJETO_FIN
DELETE FROM T_SIP_PROJETO
    WHERE CD_PROJETO =
        (SELECT CD_PROJETO FROM T_SIP_COPY_PROJETO_FIN WHERE CD_PROJETO=1);
```

# Linguagem SQL



## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO DE DELEÇÃO/ELIMINAÇÃO UTILIZANDO SUBCONSULTA

Observe o conteúdo de cada tabela, antes da execução do comando apresentado na página anterior.

**TABELA: PROJETO**

Colunas	Dados	Model	Constraints	Concessões	Estatísticas	Triggers	Flashback	Dependências	Detalhes	Partições	Índices
	CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO							
1	1	PROJETO ABC	01/02/18	15/03/20							
2	2	PROJETO DEF	14/03/19	15/03/20							
3	3	PROJETO GHI	14/03/19	15/03/20							

**TABELA: COPY\_PROJETO\_FIN**

Colunas	Dados	Model	Constraints	Concessões	Estatísticas	Triggers	Flashback	Dependências	Detalhes	Partições	Índices	SQ
	CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO								
1	1	PROJETO ABC	01/02/18	15/03/20								



## Manipulação dos Dados utilizando a linguagem SQL

## EXEMPLO DE DELEÇÃO/ELIMINAÇÃO UTILIZANDO SUBCONSULTA

Observe o conteúdo da tabela PROJETO, após a execução do comando de alteração.

Observe que o projeto de código igual a 1 foi apagado.

### TABELA: PROJETO

T\_SIP\_PROJETO

ColunasDadosModelConstraintsConcessõesEstatísticasTriggersFlashbackDependênciasDetalhesPartições

Classificar..

Filtro:

	CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO
1	2	PROJETO DEF	14/03/19	15/03/20
2	3	PROJETO GHI	14/03/19	15/03/20



# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL

**ATENÇÃO!**

## Manipulação dos Dados utilizando a linguagem SQL

- ❑ Todo comando DDL (Create, Alter e Drop) e DCL (Grant e Revoke) provocam um COMMIT implícito, ou seja, após a execução de alguma cláusula de DML (Insert, Update ou Delete), se for executado qualquer comando de DDL, os registros serão automaticamente salvos no SGBD.
- ❑ Um ROLLBACK automático ocorre quando há finalização anormal de uma transação (exemplo: falha no sistema).

## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO



#### Instrução COMMIT

```
1 UPDATE  PESSOA_FISICA
2 SET     nom_pessoa = 'Miguel Ambrósio da Silva',
3         dat_nascimento = '13/07/1965'
4 WHERE   num_pessoa_pf = 34513;
5
6 COMMIT;
```

Observe que a transação é iniciada pela instrução UPDATE. A partir dessa instrução, uma cópia dos dados anteriores à alteração é armazenada no segmento de ROLLBACK ou no segmento de UNDO (linhas 1 a 4).

Quando a operação realizada pela instrução UPDATE é concluída, a alteração fica disponível em memória, ou seja, ela ainda não foi refletida fisicamente no banco de dados e, caso outros usuários acessem os dados, visualizarão a cópia anterior à operação (linha 4).

A instrução COMMIT (linha 6) efetiva a operação. Consequentemente, a transação é concluída e a alteração dos dados é refletida de forma física na base de dados. Nesse momento, caso outros usuários acessem o registro, visualizarão os dados atualizados. A imagem antiga é retirada do segmento de ROLLBACK ou UNDO.



#### PONTO DE ATENÇÃO

É importante lembrar que, após a emissão da instrução COMMIT, não há como reverter a operação realizada.

## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO



#### Instrução ROLLBACK

```
1 DELETE FROM PESSOA_FISICA  
2 WHERE   = 34513;  
3  
4 ROLLBACK;
```

Na sintaxe, a transação é iniciada com a instrução DELETE. Com esta instrução, uma cópia dos dados é armazenada no segmento de ROLLBACK ou UNDO.

Da mesma forma como ocorreu no Exemplo 12, quando a operação DELETE é concluída, os dados ainda não são removidos fisicamente do banco de dados. Como a instrução COMMIT não foi acionada, a exclusão está armazenada em memória.

Nesse momento, ainda é possível descartar a transação. Para isso, a instrução ROLLBACK (linha 4) é acionada e cancela a operação de exclusão, retornando os dados ao estado anterior à operação. Isto é, os dados do segmento de ROLLBACK ou UNDO são gravados novamente na tabela.

# Linguagem SQL

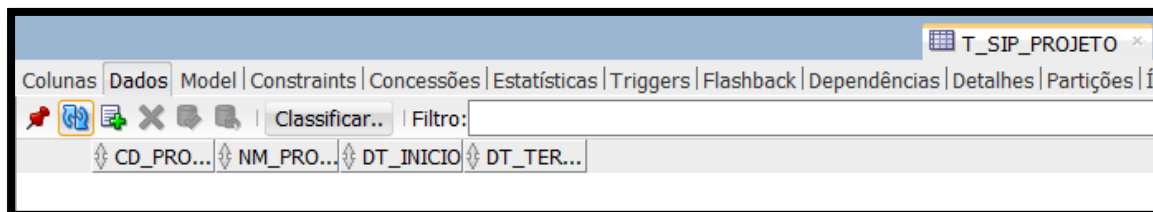
## Manipulação dos Dados utilizando a linguagem SQL

### EXEMPLO COM ROLLBACK

O comando abaixo apagará todas as linhas/registros da tabela PROJETO.

```
DELETE FROM T_SIP_PROJETO;
```

Tabela PROJETO, após a execução do comando DELETE.

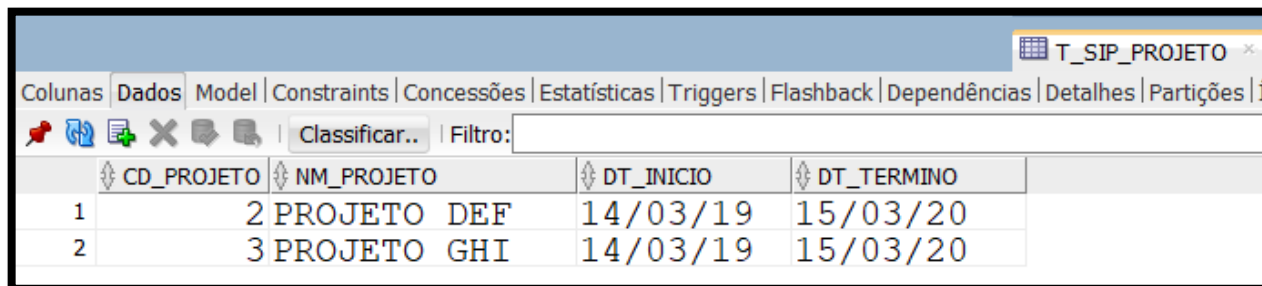


CD_PRO...	NM_PRO...	DT_INICIO	DT_TER...
-----------	-----------	-----------	-----------

A instrução ROLLBACK, irá desfazer a deleção executada anteriormente, uma vez que a deleção não foi efetivada ainda (não foi executada a instrução COMMIT).

```
ROLLBACK;
```

Tabela PROJETO, após a execução da instrução ROLLBACK. Observe que a deleção foi desfeita.



	CD_PROJETO	NM_PROJETO	DT_INICIO	DT_TERMINO
1	2	PROJETO DEF	14/03/19	15/03/20
2	3	PROJETO GHI	14/03/19	15/03/20

## Manipulação dos Dados utilizando a linguagem SQL

### ERROS COMUNS

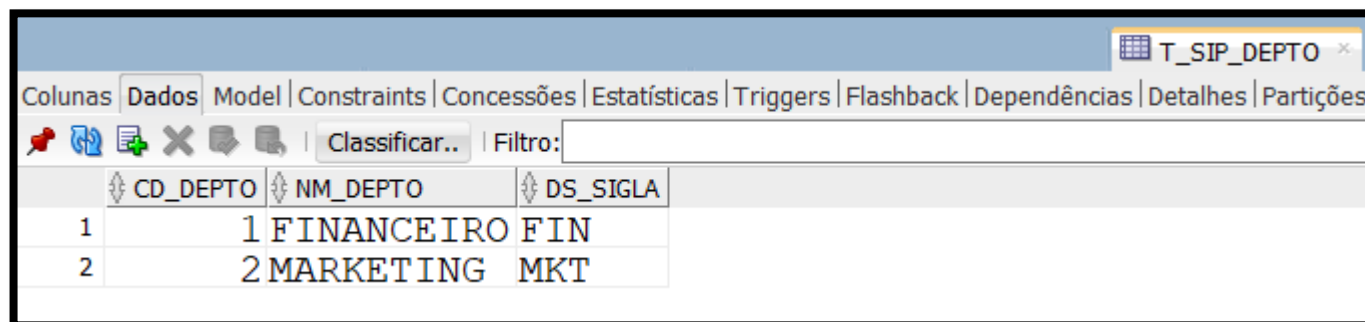
## Manipulação dos Dados utilizando a linguagem SQL

### ERRO DE VIOLAÇÃO DE CHAVE PRIMÁRIA

Tentativa de inserção de um novo registro, utilizando uma chave primária já cadastrada/existente.

```
-- ERRO DE VIOLAÇÃO DE CHAVE PRIMARIA
-- INFORMAR UMA CHAVE PRIMARIA JÁ EXISTENTE
INSERT INTO T_SIP_DEPTO VALUES (1, 'TESTE', 'TST');
/*
Erro a partir da linha : 197 no comando -
INSERT INTO T_SIP_DEPTO VALUES (1, 'TESTE', 'TST')
Relatório de erros -
ORA-00001: restrição exclusiva (ADMIN.PK_SIP_DEPTO) violada
*/
```

Observe que já existe um registro na tabela DEPTO, cadastrado com código igual a 1.



The screenshot shows a database management tool interface with a tab for the table T\_SIP\_DEPTO. The 'Dados' (Data) tab is selected, displaying a table with three columns: CD\_DEPTO, NM\_DEPTO, and DS\_SIGLA. The table contains two rows of data.

	CD_DEPTO	NM_DEPTO	DS_SIGLA
1	1	FINANCEIRO	FIN
2	2	MARKETING	MKT

## Manipulação dos Dados utilizando a linguagem SQL

### ERRO DE VIOLAÇÃO DE UNIQUE CONSTRAINT

Tentativa de inserção de um novo registro, utilizando um nome de departamento já cadastrado/existente.

```
-- ERRO DE VIOLAÇÃO DE UNIQUE CONSTRAINT
-- INFORMAR UM NOME DE DEPTO QUE JÁ EXISTE
INSERT INTO T_SIP_DEPTO VALUES (3, 'MARKETING', 'MKT');
/*
Erro a partir da linha : 177 no comando -
INSERT INTO T_SIP_DEPTO VALUES (3, 'MARKETING', 'MKT')
Relatório de erros -
ORA-00001: restrição exclusiva (ADMIN.UN_SIP_DEPTO_NOME) violada
*/
```

Observe que já existe um registro na tabela DEPTO, cadastrado com o nome 'MARKETING'.

T_SIP_DEPTO			
Colunas	Dados	Model	Constraints
CD_DEPTO	NM_DEPTO	DS_SIGLA	
1	1 FINANCEIRO	FIN	
2	2 MARKETING	MKT	



## ERRO DE VIOLAÇÃO DE CHAVE ESTRANGEIRA

Tentativa de inserção de um novo registro, utilizando um código de departamento que não existe na tabela DEPTO.

```
-- ERRO DE VIOLAÇÃO DE CHAVE ESTRANGEIRA
-- INFORMAR UM CODIGO DE DEPTO QUE NÃO ESTA CADASTRADO NA TAB. DEPTO
INSERT INTO T_SIP_FUNCIONARIO VALUES
(12349, 99, -- CODIGO DO DEPTO INVALIDO
 'TESTE', TO_DATE('01/01/1900','DD/MM/YYYY'), TO_DATE('01/01/1900','DD/MM/YYYY'),
 'RUA TESTE', 1895.33);
/*
Erro a partir da linha : 187 no comando -
INSERT INTO T_SIP_FUNCIONARIO VALUES
(12349, 99, -- CODIGO DO DEPTO INVALIDO
 'TESTE', TO_DATE('01/01/1900','DD/MM/YYYY'), TO_DATE('01/01/1900','DD/MM/YYYY'),
 'RUA TESTE', 1895.33)
Relatório de erros -
ORA-02291: restrição de integridade (ADMIN.FK_SIP_FUNC_DEPTO) violada - chave mãe não localizada
*/
```

Observe que não existe um departamento na tabela DEPTO, cadastrado com o código 99.

49

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL

### ERRO DE VIOLAÇÃO DE CHECK CONSTRAINT

Tentativa de inserção de um novo registro, utilizando um valor para o salário menor que 1045.

Violando a restrição CHECK, onde foi configurado a regra de validação para o salário.

```
-- ERRO DE VIOLAÇÃO DE CHECK CONSTRAINT
-- INFORMAR UM SALARIO MENOR QUE O VALOR QUE TEMOS NA CHECK
-- CHECK -> VL_SALARIO >= 1045
INSERT INTO T_SIP_FUNCIONARIO VALUES
(12349, 1,
'TESTE', TO_DATE('01/01/1900','DD/MM/YYYY'), TO_DATE('01/01/1900','DD/MM/YYYY'),
'RUA TESTE', 600 ); -- INFORMOU UM SALARIO MENOR QUE 1045
/*
Erro a partir da linha : 204 no comando -
INSERT INTO T_SIP_FUNCIONARIO VALUES
(12349, 1,
'TESTE', TO_DATE('01/01/1900','DD/MM/YYYY'), TO_DATE('01/01/1900','DD/MM/YYYY'),
'RUA TESTE', 600 )
Relatório de erros -
ORA-02290: restrição de verificação (ADMIN.CK_SIP_FUNC_SALARIO) violada
*/
```

Observe a regra de validação configurada para a tabela FUNCIONÁRIO, no campo VL\_SALARIO.

Tabela - T_SIP_FUNCIONARIO			
Restrições do Nível da Tabela			
<div> <div></div> <div></div> </div>			
	Nome	Regra de Validação	Gerar em DDL
1	CK_SIP_FUNC_SALARIO	VL_SALARIO >= 1045	<input checked="" type="checkbox"/>

```
-- CONSTRAINT CHECK NA TABELA FUNCIONARIO - VALIDANDO O SALARIO
ALTER TABLE T_SIP_FUNCIONARIO
ADD CONSTRAINT CK_SIP_FUNC_SALARIO
CHECK (VL_SALARIO >= 1045);
```

## Manipulação dos Dados utilizando a linguagem SQL

### ERRO REFERENTE AO TAMANHO DE DADOS DE UM CAMPO NUMÉRICO (MAIOR QUE A CAPACIDADE)

Tentativa de inserção de um novo registro, utilizando um código de departamento, que excede o tamanho máximo especificado para o campo.

```
-- ERRO REFERENTE AO TAMANHO DE DADOS
-- INFORMAR 5 DÍGITOS PARA O CÓDIGO DO DEPTO, ONDE SÃO ESPERADOS APENAS 3
INSERT INTO T_SIP_DEPTO VALUES (12345, 'DEPTO TESTE', 'TST');
/*
Erro a partir da linha : 220 no comando -
INSERT INTO T_SIP_DEPTO VALUES (12345, 'DEPTO TESTE', 'TST')
Relatório de erros -
ORA-01438: valor maior que a precisão especificada usado para esta coluna
*/
```

Observe a estrutura da tabela DEPTO, onde o código do departamento, foi definido com no máximo 3 dígitos numéricos inteiros.

T_SIP_DEPTO				
Colunas   Dados   Model   Constraints   Concessões   Estatísticas   Triggers   Flashback   Dependências   Detalhes   Partições				
Ações...				
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA
1	CD DEPTO	NUMBER (3, 0)	No	(nu
2	NM DEPTO	VARCHAR2 (30 BYTE)	No	(nu
3	DS SIGLA	CHAR (3 BYTE)	No	(nu

## Manipulação dos Dados utilizando a linguagem SQL

### ERRO REFERENTE AO TIPO DE DADOS INVÁLIDO

Tentativa de inserção de um novo registro, utilizando um código de departamento com conteúdo caractere, onde é especificado um número.

```
-- ERRO DE TIPO DE DADOS
-- INFORMAR UM CONTEÚDO CARACTERE, ONDE E ESPERADO NUMERO
INSERT INTO T_SIP_DEPTO VALUES ('XXX', 'DEPTO TESTE', 'TST');
/*
Erro a partir da linha : 230 no comando -
INSERT INTO T_SIP_DEPTO VALUES ('XXX', 'DEPTO TESTE', 'TST')
Relatório de erros -
ORA-01722: número inválido
*/
```

Observe a estrutura da tabela DEPTO, onde o código do departamento, foi definido com no máximo 3 dígitos numéricos inteiros.

T_SIP_DEPTO				
Colunas   Dados   Model   Constraints   Concessões   Estatísticas   Triggers   Flashback   Dependências   Detalhes   Partições				
Ações...				
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA
1	CD DEPTO	NUMBER (3, 0)	No	(nu
2	NM DEPTO	VARCHAR2 (30 BYTE)	No	(nu
3	DS SIGLA	CHAR (3 BYTE)	No	(nu

# Linguagem SQL

## Manipulação dos Dados utilizando a linguagem SQL

### ERRO REFERENTE A VALORES INSUFICIENTES PARA INSERÇÃO

Tentativa de inserção de um novo registro, utilizando menos valores (conteúdos dos campos) do que o esperado na tabela, conforme definição de sua estrutura.

```
-- ERRO DE VALORES INSUFICIENTES PARA A INSERÇÃO
INSERT INTO T_SIP_DEPTO VALUES ('XXX', 'DEPTO TESTE');
/*
Erro a partir da linha : 240 no comando -
INSERT INTO T_SIP_DEPTO VALUES ('XXX', 'DEPTO TESTE')
Erro na Linha de Comandos : 240 Coluna : 13
Relatório de erros -
Erro de SQL: ORA-00947: não há valores suficientes
00947. 00000 - "not enough values"
*Cause:
*Action:
*/
```

Observe a estrutura da tabela DEPTO, durante a inserção, são esperados valores (conteúdo) para três colunas e foram informados apenas 2 valores.

T_SIP_DEPTO				
Colunas   Dados   Model   Constraints   Concessões   Estatísticas   Triggers   Flashback   Dependências   Detalhes   Partições				
Ações...				
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA
1	CD DEPTO	NUMBER (3, 0)	No	(nu
2	NM DEPTO	VARCHAR2 (30 BYTE)	No	(nu
3	DS SIGLA	CHAR (3 BYTE)	No	(nu

## Manipulação dos Dados utilizando a linguagem SQL

### ERRO REFERENTE A VALORES DEMAIS ENVIADOS PARA INSERÇÃO

Tentativa de inserção de um novo registro, utilizando mais valores (conteúdos dos campos) do que o esperado na tabela, conforme definição de sua estrutura.

```
-- ERRO DE VALORES A MAIS DO QUE O ESPERADO
INSERT INTO T_SIP_DEPTO VALUES (5, 'DEPTO TESTE', 'TST', 'XXX'); -- UM VALOR A MAIS ENVIADO NA INSERCAO
/*
Erro a partir da linha : 253 no comando -
INSERT INTO T_SIP_DEPTO VALUES (5, 'DEPTO TESTE', 'TST', 'XXX')
Erro na Linha de Comandos : 253 Coluna : 13
Relatório de erros -
Erro de SQL: ORA-00913: valores demais
00913. 00000 - "too many values"
*Cause:
*Action:
*/
```

Observe a estrutura da tabela DEPTO, durante a inserção, são esperados valores (conteúdo) para três colunas e foram informados 4 valores.

T_SIP_DEPTO				
Colunas   Dados   Model   Constraints   Concessões   Estatísticas   Triggers   Flashback   Dependências   Detalhes   Partições				
Ações...				
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA
1	CD DEPTO	NUMBER (3, 0)	No	(nu
2	NM DEPTO	VARCHAR2 (30 BYTE)	No	(nu
3	DS SIGLA	CHAR (3 BYTE)	No	(nu

# Próxima aula estudaremos

- ☐ Revisão de conceitos através de exercícios

# REFERÊNCIAS



- MACHADO, Felipe Nery R. Banco de Dados - Projeto e Implementação. Érica, 2004.
- Páginas: 330, 331.
  
- ELMASRI, R.; NAVATHE, S.B. Sistemas de Banco de Dados: Fundamentos e Aplicações. Pearson, 2005. Páginas: 153, 154.
  
- PRICE, JASON, ORACLE DATABASE 11 g – SQL Domine SQL e PL-SQL no banco de Dados Oracle, Bookman, 2008. Capítulo: 8. Páginas: 280 a 301.
  
- Puga, S.; França. Ed.; Goya, M. Banco de Dados – Implementação em SQL, PL/SQL e Oracle 11g. Pearson, 2014.



Copyright © 2023 Prof. Diogo Alves

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).