



INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Colatina

BANCO DE DADOS I

Linguagem de Modelagem de Dados (DML)

Prof. Gustavo Ludovico Guidoni



Adicionando Linhas em uma Tabela

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

```
INSERT INTO tabela [(coluna [, coluna...])]  
VALUES              (valor [, valor...])
```



Adicionando Dados em uma Tabela

- Inserindo listando os nomes das colunas na cláusula INSERT

```
INSERT INTO fita (cod_fita, nom_fita, cor, sit_fita, cod_genero)
VALUES          (300, 'Psicose', 'AM', 2, 6)
```

- Inserindo omitindo os nomes das colunas na cláusula INSERT

– Deve respeitar a ordem de criação das colunas

```
INSERT INTO fita
VALUES (310, 'Loucademia de polícia', 'AM', 3, 2)
```



Adicionando Dados em uma Tabela

- Adicionando linhas com valores nulos
 - **Método implícito:** omitindo a coluna na lista de colunas

```
INSERT INTO locacao (cod_cliente, cod_fita)
VALUES      (110, 300)
```

- **Método explícito:** especificando a palavra-chave NULL

```
INSERT INTO locacao (cod_cliente, cod_fita, dat_inclusao)
VALUES      (110, 300, NULL)
```



Adicionando Dados em uma Tabela

- Adicionando diversos valores
 - Colocam-se os valores separados por vírgula

```
INSERT INTO produto (nome, preco, qtde)
VALUES ('Arroz', 15.50, 300) ,
       ('Feijão', 8.90, 200),
       ('Picanha', 65.99, 150)
```



Adicionando Dados em uma Tabela

- Campos auto numeração:
 - Não deve ser passado no INSERT
- Chave estrangeira (FK):
 - Deve existir na tabela de origem
- Campos com valor **DEFAULT**:
 - Se não for passado no INSERT, será inserido com o valor DEFAULT, definido na criação da tabela



Adicionando Dados em uma Tabela

- Inserindo dados em uma tabela a partir de outra

```
INSERT INTO tabela [(coluna [, coluna...])]  
SELECT      ...
```

- A quantidade e o tipo das colunas do **select** deve ser a mesma do **insert**



Adicionando Dados em uma Tabela

- Exemplo: insere as locações que serão incluídas a partir de uma data de acerto

```
INSERT INTO locacao (cod_cliente, cod_fita)
SELECT      cod_cliente_aux, cod_fita_aux
FROM        acerto
WHERE       dat_acerto > "2000-01-01"
```




Alterando Dados de uma Tabela

- Altere linhas existentes em uma tabela utilizando a instrução **UPDATE**

```
[ WITH [ RECURSIVE ] with_query [, ...] ]  
UPDATE [ ONLY ] table_name [ * ] [ [ AS ] alias ]  
  SET { column_name = { expression | DEFAULT } |  
    ( column_name [, ...] ) = [ ROW ] ( { expression | DEFAULT } [, ...] ) |  
    ( column_name [, ...] ) = ( sub-SELECT )  
  } [, ...]  
[ FROM from_item [, ...] ]  
[ WHERE condition | WHERE CURRENT OF cursor_name ]  
[ RETURNING { * | output_expression [ [ AS ] output_name ] } [, ...] ]
```



Alterando Dados de uma Tabela

- Especifique a tabela a ser atualizada na cláusula **UPDATE**
- Especifique as colunas e os novos valores na cláusula **SET**
- Observe que há a cláusula **FROM** na sintaxe do comando
 - É possível fazer **JOIN** no comando **UPDATE**
- Especifique na cláusula **WHERE** as condições para restrição das linhas a atualizar
 - Caso esta cláusula seja omitida, toda a tabela será alterada



Alterando Dados de uma Tabela

- Atualizando toda a tabela

```
UPDATE produto  
SET  preco = preco * 1.50
```

- Atualizando linhas específicas da tabela

```
UPDATE cliente  
SET  end_cliente = 'Av. Amazonas, 100/500',  
      bairro = 'Centro'  
WHERE cod_cliente = 160
```



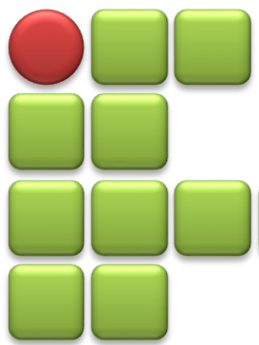
Alterando Dados de uma Tabela

- Caso exista necessidade de utilizar outra(s) tabela(s) para restringir as linhas a serem atualizadas:
 - Especifique a tabela a ser alterada na cláusula **UPDATE**
 - Utilize a cláusula **FROM** para fazer **JOINS** com as demais tabelas
 - O Join da cláusula **UPDATE** e da primeira tabela da clausula **FROM** deve ser feito no **WHERE**

Alterando Dados de uma Tabela

- Exemplo:
 - Alterar a data de devolução para a data prevista de devolução de todas as fitas do gênero “Drama” e do cliente 2, que ainda não foram devolvidas

```
UPDATE locacao l
SET    dat_devolucao    = l.dat_prevista_devolucao
FROM    fita fit
JOIN    filme fil
        ON      fit.cod_filme      = fil.cod_filme
JOIN    genero g
        ON      fil.cod_genero     = g.cod_genero
WHERE   l.cod_fita      = fit.cod_fita
AND     g.nom_genero    = 'Drama'
AND     cod_cliente    = 2
AND     l.dat_devolucao IS NULL
```

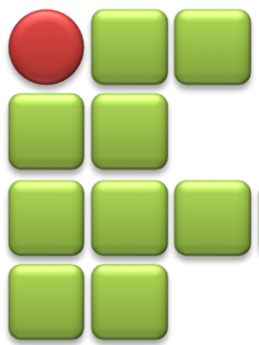


Removendo Linhas de uma Tabela

- Remova linhas de uma tabela utilizando a instrução **DELETE**

```
DELETE FROM tabela  
[WHERE condicao]
```

- Especifique na cláusula **DELETE** a tabela de onde as linhas serão removidas



Removendo Linhas de uma Tabela

- Se necessário, especifique na cláusula **WHERE** as condições para restrição das linhas a remover
 - Caso esta cláusula seja omitida, todas as linhas da tabela serão removidas. Entretanto, a estrutura da tabela será mantida



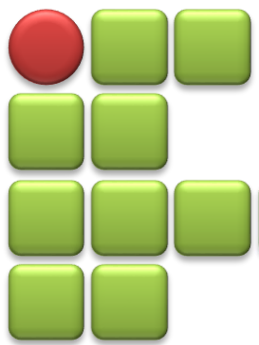
Removendo Linhas de uma Tabela

- Removendo todas as linhas de uma tabela

```
DELETE FROM genero
```

- Removendo linhas específicas de uma tabela

```
DELETE FROM cliente  
WHERE cid_cliente = 'Sete Lagoas'
```

Removendo Linhas de uma Tabela

- Removendo linhas com chave estrangeira (FK)
 - Se a FK foi criada com a restrição "ON DELETE RESTRICT", então um erro ocorrerá e o registro não será excluído.
 - Se a FK foi criada com a restrição "ON DELETE CASCADE", então todos os registros da tabela de origem, com o mesmo valor da FK, serão excluídos.

