

Banco de Dados I

Professor Msc. Aparecido Vilela Junior
aparecido.vilela@unicesumar.edu.br

Manipulando Dados

- Uma manipulação de dados é executada quando:
 - Adiciona-se informações (linhas) na tabela
 - Modifica-se informações existentes na tabela
 - Remove-se informações existentes na tabela.

INSERT

- O comando INSERT é utilizado para inserir linhas em tabelas.
- A inclusão pode ser feita linha a linha ou os valores podem ser obtidos de outras tabelas.
- Podemos definir quais colunas serão preenchidas, informando valores para somente estas colunas.

INSERT - Sintaxe

- Adicionando novas linhas para uma tabela:
 - INSERT INTO table [(column [,column...])]
 - VALUES (value [,value...]);
- Somente uma linha por vez é inserida com esta sintaxe

INSERT

- A inserção de novas linhas (informações) poderá ser feita:
 - Inserindo uma nova linha contendo valores para cada coluna

```
INSERT INTO DEPT (DEPTNO, DNAME, LOC)  
VALUES (60, 'POS BANCO', 'LONDRINA')
```

- Associando valores a cada uma das colunas, não informando as colunas da tabela.

```
INSERT INTO DEPT VALUES (65, 'ORACLE', 'MARINGA', 'PR')
```

INSERT

- Inserindo linhas com o comando SELECT, para efetuar a inclusão de diversas linhas simultaneamente.
 - insert into dept (deptno, dname)
 - select 99, 'teste' from dual
- Strings e Datas dever ser escritas entre aspas simples.

INSERT

- Inserindo linhas com valores nulos.
 - Método Implícito: Omitir a coluna da lista.

```
INSERT INTO DEPT (DEPTNO, DNAME)  
VALUES (100,'RH');
```

Método Explícito: Especifica a palavra NULL nos valores.

```
INSERT INTO DEPT  
VALUES (110,'CONTABIL',NULL)
```


INSERT

- Inserindo valores especiais: Data Corrente

```
INSERT INTO EMP (EMPNO, ENAME, DEPTNO, HIREDATE)
VALUES
(1234,'NOME TESTE',10,SYSDATE)
```

Inserindo valores específicos da data.

```
INSERT INTO EMP (EMPNO, ENAME, DEPTNO, HIREDATE)
VALUES
(1234,'NOME TESTE',10,TO_DATE('JAN 3, 1969', 'MON DD, YYYY' ))
```

INSERT

- Inserindo valores utilizando Variáveis de Substituição.

```
INSERT INTO DEPT  
(DEPTNO, DNAME, DLOC)  
VALUES  
(&CODIGO, '&NOME', '&LOCAL')
```

Utilizar & para sinalizar uma variável de entrada de valores

INSERT

- Copiando Linhas a partir de uma outra tabela.

```
INSERT INTO EMP (EMPNO, ENAME, SAL, DEPTNO)
SELECT EMPNO + EMPNO, ENAME, SAL + 300, 10
FROM EMP
WHERE DEPTNO = 20
```

- Não utiliza a cláusula VALUES
- O número de colunas deve ser igual a lista de colunas da cláusula SELECT

UPDATE

- O comando UPDATE tem a finalidade de alterar informações já gravadas na base de dados.
- Possui uma cláusula WHERE, que determinará quais linhas serão modificadas.
- O Oracle faz um select implícito no banco de dados, para determinar as linhas que atendem a cláusula WHERE.

UPDATE

- Alterando linhas em uma tabela:

UPDATE TABLE

SET column = value [, column = value,...]

[WHERE condition];

UPDATE

- A linha ou linhas que serão alteradas são especificadas na cláusula WHERE.
- UPDATE EMP
SET DEPTNO = 20
WHERE EMPNO = 1234
- Todas as linhas serão alteradas se for omitida a cláusula WHERE
- UPDATE EMP
SET DEPTNO = 20

UPDATE

- Alterando linhas, utilizando uma SubQuery.

```
UPDATE EMP
```

```
SET DEPTNO = (SELECT DEPTNO FROM EMP WHERE  
EMPNO = 1234),
```

```
SAL = (SELECT SAL FROM EMP WHERE EMPNO =  
1234)
```

```
WHERE EMPNO = 1275
```

UPDATE

- Atualizando Linhas: Erro de Constraint de Integridade:

```
UPDATE EMP  
SET DEPTNO = 55  
WHERE DEPTNO = 110
```

- O departamento 55 não existe

Exercício

- Atualize os funcionários (EMP) que residam em CHICAGO ou NEW YORK, com os seguintes valores:
 - Eles passarão a residir em BOSTON.
 - Eles terão como acréscimo no salário de 10% da média do salário do seu departamento e na comissão 15% em relação a média do seu departamento.

DELETE

- Excluir linhas cadastradas no banco de dados.
- Da mesma forma que o UPDATE, a cláusula WHERE será responsável por determinar que linhas poderão ser removidas.
- Caso essa cláusula não seja informada, o comando tentará remover todas as linhas da tabela informada

DELETE

- Removendo linhas de uma tabela:

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

Obs.: Se nenhuma linha for excluída, a mensagem '0 rows deleted' é retornada

DELETE

- Especificando linhas a serem excluídas pela cláusula WHERE:

```
DELETE FROM DEPT  
WHERE DNAME = 'Finance'
```

- Todas as linhas da tabela serão excluídas se for omitida a cláusula WHERE:

```
DELETE FROM DEPT
```

DELETE

- Removendo linhas com base nos valores de outra tabela:

```
DELETE FROM EMP  
WHERE DEPTNO = (SELECT DEPTNO  
                FROM DEPT  
                WHERE DNAME LIKE  
                '%FINAN%');
```

DELETE

- Removendo Linhas: Erro de Constraint de Integridade:
 - Não pode ser excluída uma linha que contenha uma primary key, que é utilizada como foreign key em outra tabela.

CONTROLE DE TRANSAÇÕES

- Uma transação é uma sequência “indivisível” de comandos SQL, que modifica o conteúdo do banco de dados.
- Ex: Para transferir uma certa quantia de uma conta A para uma conta B, devemos:
 - Retirar o dinheiro da conta A;
 - Depositar o dinheiro na conta B;
- Para que a operação acima seja segura, ou os dois passos são executados ou nenhum deles deve ocorrer.

CONTROLE DE TRANSAÇÕES

- O início de uma transação é determinado pelo primeiro comando de manipulação de dados (INSERT, UPDATE, DELETE).
- O fim de uma transação é determinado por uma das seguintes formas:
 - Execução do comando COMMIT; todas as modificações são efetivadas no BD
 - Execução do comando ROLLBACK; nenhuma modificação é efetivada
 - Execução de comandos de definição de dados DDL
 - Término da sessão corrente equivale ao COMMIT

COMMIT e ROLLBACK

- Vantagens do uso de Commit e Rollback:
 - Assegura a consistência dos dados
 - Visualização prévia das mudanças dos dados antes de fazer mudanças permanentes.
 - Operações lógicas relacionados ao grupo

Transações Implícitas

- Um Commit automático, pode ocorrer nas seguintes circunstâncias:
 - Quando ocorre uma indicação DDL (CREATE, DROP, ALTER, RENAME, TRUNCATE)
 - Quando ocorre uma indicação DCL (GRANT, REVOKE)
 - Um Rollback automático ocorre, sob o término anormal do Sql*Plus ou falha do sistema.

Estado dos dados

- Antes de um commit ou rollback:
 - O estado prévio dos dados pode ser recuperado;
 - O usuário pode verificar o resultado das operações de DML, utilizando um select;
 - Outros usuários não podem verificar o resultado das operações de DML, utilizadas pelo usuário corrente;
 - As linhas alteradas são bloqueadas (locked), outros usuários não podem alterar os dados;

Estado dos dados

- Depois de um Commit:
 - Dados alterados ficarão permanentes no BD;
 - O estado anterior dos dados, serão perdidos;
 - Todos os usuários verão os resultados;
 - Bloqueios das linhas serão liberados;
 - Todos os Savepoints serão eliminados;

Estado dos dados

- Depois de um Rollback:
 - Alterações são desfeitas;
 - Estado anterior dos dados é mantido;
 - Bloqueio das linhas afetadas são liberados;