CWI SOFTWARE

módulo 2 | banco de dados

2 - Comandos DML - SQL

André Luís Nunes

Tipos de comandos

→ DDL: Data Definition Language

→ Comandos que permitem definir a estrutura de objetos.

→ DML: Data Manipulation Language

→ Comandos que permitem alterar os dados de uma tabela.

→ Inserir, alterar, excluir e consultar regitros.

→ DCL: Data Control Language

→ Comandos que permitem conceder ou revogar permissões à determinados objetos de uma base dados.

DML - Manipulação

Comandos que manipulam dados:

- → INSERT
- → UPDATE
- → DELETE
- → SELECT

```
Insert into Cidade
 (IDCidade, Nome, UF)
Values
 (1, 'São Leopoldo', 'RS');
```

» O comando SELECT não manipula nenhuma informação, mas é considerado um DML.

DML – Manipulação → Insert

- O número de colunas da claúsula INTO deve ser o mesmo da VALUES;
- Se uma coluna não for especificada o valor padrão (default) é usado;
- Os tipos de objetos que permitem este comando são: tabelas de views*;

» No comando de INSERT todas as colunas obrigatórias devem ser informadas*.

*Exceções: quando a coluna possuir um valor *default* (inicial) ou quando for auto-incremento.

» **Boa prática**: sempre informe as colunas no comando **insert**.

TIPOS DE COMANDOS: DML

mãos à obra

Inserção de registros, execute o script Lab2.sql

- Cidade
- Associado

CursoSQL

Termos mais utilizados para o comando Insert:

- Criar;
- Popular;
- Carregar;
- Inserir;
- Gerar.

TIPOS DE COMANDOS: DML – Copiando 1 tabela (CTA)

Para copiar uma tabela a partir de um comando SQL:

```
Select *
into CopiaCidade
From Cidade;
```

- ✓ Serão copiados todos os atributos e os dados;
- ✓ Não serão copiados as contraints e os índices.

Neste exemplo foi utilizado apenas 1 tabela, e utilizado todos os atributos, Mas poderíamos ter utilizado apenas alguns atributos e também utilizado mais de uma tabela.

TIPOS DE COMANDOS: DML – Manipulação > Update

Comandos que manipulam dados:

→ INSERT

→ UPDATE

→ DELETE

→ SELECT

```
Update <TableName>
Set    <Column1> = <expression>
      ,<Column2> = <expression>
       ...
Where  <predicate>;
```

```
Update Cidade
Set    Nome = 'Novo Hamburgo'
Where  IDCidade = 1;
```

» Permite alteração em massa (vários registros), conforme a condição (Where).

» Se for omitida a cláusula Where todos os registos serão alterados.

» Possibilita que uma ou todas as colunas tenham seu contéudo alterado em uma instrução.

TIPOS DE COMANDOS: DML – Manipulação > Update

Termos normalmente utilizados:

- Atualizar;
- Alterar;
- Definir;
- Setar;

TIPOS DE COMANDOS: DML – Manipulação > Delete

Comandos que manipulam dados:

→ INSERT

→ UPDATE

→ **DELETE**

→ SELECT

```
Delete From <TableName>
Where  <predicate>;
```

```
Delete Cidade
Where  IDCidade = 1;
```

» Permite exclusão em massa (vários registros), conforme a condição (Where).

» Se for omitida a cláusula Where todos os registos serão excluídos*.

* Desde que o registro não esteja sendo referenciado em outra tabela.

TIPOS DE COMANDOS: DML – Manipulação > Delete

Termos normalmente utilizados:

- Excluir;
- Deletar;
- Eliminar;
- Cancelar;
- Limpar;

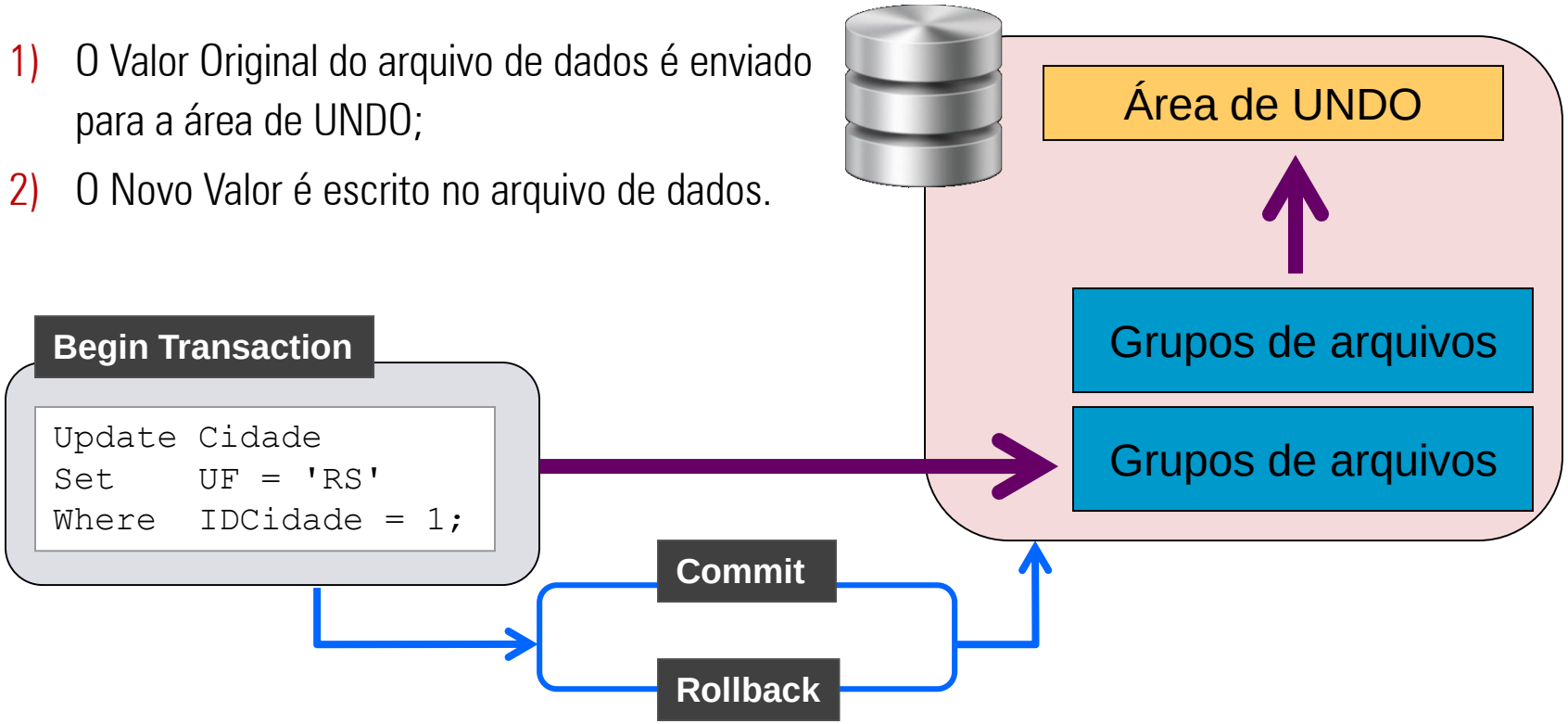
TRANSAÇÃOES

- **Transações:** as operações de manipulação de dados nos bancos de dados (insert, update e delete) são realizadas através de transações. E para que outros usuários tenham acesso aos dados alterados por um processo é necessário efetivar a transação, com o comando COMMIT. Para desfazer as alterações é necessário um ROLLBACK.
  - Toda transação deve ser iniciada de forma explícita, **BEGIN TRANSACTION**.
  - COMMIT: efetiva as alterações;
  - ROLLBACK: desfaz as alterações.

TRANSAÇÕES

As operações de manipulação de dados (insert, update e delete) são realizadas através de transações.

1) O Valor Original do arquivo de dados é enviado para a área de UNDO;
2) O Novo Valor é escrito no arquivo de dados.



TIPOS DE COMANDOS: DML – Manipulação > Truncate

O comando DELETE utiliza transação, e permite o uso de ROLLBACK, o truncate é mais rápido, porém não permite desfazer;

```
Truncate Table Cidade;
```

Não é possível utilizar filtros (where) no truncate.

TIPOS DE COMANDOS: DML – exercícios

TIPOS DE COMANDOS: DML – Manipulação > Select

Comandos que manipulam dados:

→ INSERT

→ UPDATE

→ DELETE

→ SELECT

```
Select <Column1>, <Column2>
From   <TableName>
[Where  <predicate>]
[Order by <column>];
```

```
Select IDCidade, Nome
From   Cidade;
```

» Utilizado para retornar os registros de uma tabela (ou mais).

» Os comandos SQL **não** são CASE SENSITIVE.

TIPOS DE COMANDOS: DML – Manipulação > Select

Comandos que manipulam dados:

- A cláusula **SELECT** corresponde a **projeção** dos dados.
- A cláusula **FROM** corresponde ao produto cartesiano, são listadas todas as **fontes** (origem) de dados, que podem ser tabelas ou visões.
- A cláusula **WHERE** corresponde à seleção da álgebra relacional. Consiste em um predicado (**condição**) envolvendo as colunas das tabelas que aparecem na cláusula FROM. Os relacionamentos podem estar nesta cláusula também.

SQL – Manipulação > Select

Comando mais utilizado (exemplos):

- Selecionando o nome e UF da cidade, onde o ID é igual a 1:

```
Select Nome, UF
From   Cidade
Where  IDCidade = 1;
```

- Selecionando o nome e ID das cidades de UF igual a 'RS':

```
Select IDCidade, Nome
From   Cidade
Where  UF = 'RS';
```

- **Boa prática:** sempre informe as colunas no comando select, evite o * (asterisco).

SQL – Manipulação > Select - alias

- Colocando apelido para as colunas e tabelas:

Alterando o nome (exibição) das colunas:

```
Select Nome_Cidade as Nome,
       UF          as Estado
From   Cidade;
```

Pode ser utilizado para tabelas também (útil quando utilizado mais de 1 tabela):

```
Select c.IDCidade    as "ID Cidade",
       c.Nome_Cidade as "Nome"
From   Cidade c;
```

TIPOS DE COMANDOS: DML

mãos à obra

Criando tabelas, execute o script Lab3.sql

- Empregado
- Departamento

CursoSQL

» Estrutura auxiliar utilizada nos próximos exemplos.

SQL – Select > Operadores

Operadores matemáticos:

- Selecionando os distintos estados existentes na tabela Cidade:

```
+   Soma                /   Divisão
-   Subtração           *   Multiplicação
```

- Exemplo: selecionando o salário anual de cada funcionário:

```
Select NomeEmpregado as Nome,
       salario       as SalarioMensal,
       (salario*12)  as SalarioAnual
From  Empregado;
```

- **Boas práticas:** utilize parenteses para operações de cálculos.

SQL – Select > Order by

Ordenando o resultado de uma consulta:

- Permite ordernar de forma ascendente (default) e descrescente.

```
Select <columns>
From   <table>
Order by <column> [ASC|DESC];
```

- Ordenando o resultado pelo nome da cidade:

```
Select Nome, UF
From   Cidade
Order by Nome ASC;
```

- **Boas práticas:** não utilize o índice referente a coluna no *order by*, sempre informe o nome/alias.

SQL – Select > Where

Corresponde as condições de filtro da operação:

- Permite o uso de operadores, expressões e funções

```
Select <columns>
From   <table>
Where  <column> <operator> <value> ;
```

- Consultando todas as cidades do RS:

```
Select IDCidade, Nome
From   Cidade
Where  UF = 'RS';
```

- A clausula WHERE não se aplica somente ao comando SELECT.

SQL – Select > Where > Operadores lógicos

Principais operadores permitidos na cláusula WHERE:

Operador	Significado	Exemplo
=	igual a	Coluna1 = 30
>	maior que	Coluna2 > 100
>=	maior ou igual que	Coluna2 >= 100
<	menor que	Coluna3 < 1000
<=	menor ou igual que	Coluna3 <=1000
<>	diferente	Coluna4 <> 47
!=	diferente	Coluna4 != 47

» Consultando todos os empregados com salário maior ou igual a 1500:

```
Select NomeEmpregado, IDEmpregado
From   Empregado
Where  Salario >= 1500;
```

SQL – Select > Where > Operadores SQL

Aplicados em vários os tipos de dados:

Operador	Significado	Exemplo
between	entre dois valores (inclusive)	Col1 between 1 and 5
in	verifica um argumeno de pesquisa em uma lista	Col2 in (1,2,4)
like	pesquisa parte de um valor (string)	Col3 like 'RIO%'
is null	é um valor nulo	Col4 is null
exists	verifica se o registro existe em uma subquery	Exists (subquery)

- Consultando todas as cidades dos estados da região sul:

```
Select IDCidade, Nome, UF
From   Cidade
Where  UF in ('RS', 'SC', 'PR');
```

SQL – Select > Where > Operadores SQL negação

Aplicados em todos os tipos de dados:

Operador	Significado
not between	não está entre dois valores (inclusive)
not in	não está na lista
not like	não é como a parte pesquisada.
is not null	não é um valor nulo
not exists	não existe na subquery

» Consultando todas as cidades dos estados diferentes da região sul:

```
Select IDCidade, Nome, UF
From   Cidade
Where  UF not in ('RS', 'SC', 'PR');
```

SQL – Select > Where > Operadores AND e OR

» O operador AND determina que todas as condições devem ser verdadeiras.

» O operador OR determina que apenas uma das condições seja verdadeira.

» Todos os empregados com salário de 1000 e 1500 (inclusive) e que tenham comissão:

```
Select IDEmpregado, NomeEmpregado
From   Empregado
Where  salario >= 1000
  and  salario <= 1500
  and  comissao > 0;
```

» Todas as cidades do RS ou SC:

```
Select IDCidade, Nome, Uf
From   Cidade
Where  uf = 'RS'
   OR  uf = 'SC';
```

SQL – Select > Where > Exemplos 1/2

- Todos os empregados sem comissão:

```
Select IDEmpregado, NomeEmpregado
From   Empregado
Where  comissao IS NULL;
```

- Todos os empregados onde o nome inicia com S:

```
Select IDEmpregado, NomeEmpregado
From   Empregado
Where  NomeEmpregado LIKE 'S%';
```

- Todos os empregados com comissão maior que 0 (zero):

```
Select IDEmpregado, NomeEmpregado
From   Empregado
Where  comissao > 0;
```

SQL – Select > Where > Exemplos 2/2

- Todos os empregados com salário entre 1000 e 1500 (inclusive):

```
Select IDEmpregado, NomeEmpregado
From   Empregado
Where  salario BETWEEN 1000 and 1500;
```

- Todos os empregados com salário de 1000 e 1500 (inclusive):

```
Select IDEmpregado, NomeEmpregado
From   Empregado
Where  salario >= 1000
  and  salario <= 1500;
```

Exercícios

Crescer

André Luís Nunes

andre.nunes@cwi.com.br