

Ao final deste assunto, você será capaz de:

- Fazer a atualização dos dados de uma tabela previamente criada num banco de dados através da utilização de sentenças SQL

Neste momento, o que se estará tratando são as instâncias de dados e, para tanto, pode ser necessário fazer a atualização dos dados de tabelas através da inclusão, alteração ou exclusão de dados dessas tabelas.

3.2.1. Inclusão de Dados

Uma vez criadas as tabelas, a primeira coisa a ser feita é popular essas tabelas, ou seja, incluir dados, de modo que as tabelas possam ter dados para estarem disponíveis quando necessário.

A sentença SQL que possibilita essa inclusão de dados nas linhas de uma tabela é apresentada através do exemplo a seguir:

```
INSERT INTO ESTADO VALUES ('PA', 'PARÁ');  
INSERT INTO ESTADO VALUES ('AM', 'AMAZONAS');
```

Desta forma, estarão sendo incluídos os dados dos estados do PARÁ e AMAZONAS.

Outra forma de escrever as mesmas sentenças é a seguinte:

```
INSERT INTO ESTADO (NOME_EST, COD_EST)  
VALUES ('PARÁ', 'PA');  
  
INSERT INTO ESTADO (NOME_EST, COD_EST)  
VALUES ('AMAZONAS', 'AM');
```

A visualização do resultado dessa inclusão de dados seria o seguinte:

COD_EST	NOME_EST
AM	AMAZONAS
PA	PARÁ

3.2.2. Alteração de Dados

Como parte do processamento normal dos sistemas, os dados inseridos em tabelas precisam ser atualizados e, neste caso, isso pode implicar em modificação de valores de algumas colunas de tabelas.

A sentença SQL que possibilita essa alteração de dados nas linhas de uma tabela é apresentada através do exemplo a seguir

Supondo que foi criada uma palestra na tabela PALESTRA, vista a seguir:

COD_PAL	TIT_PAL	COD_PALESTRANTE	DATA_PAL	HORA_PAL	SALA_PAL
110	DESIGN PATTERNS	114	14/10/2003	18:30	D-200
213	QoS EM VOZ SOBRE IP	23	23/11/2003	19:30	LA-211
235	DATA MINING UTILIZANDO REDES NEURAIAS	142	05/09/2004	20:00	D-200

Pode ser necessário fazer alteração em valores das linhas dessa tabela, como a mudança de sala de uma determinada palestra, como na sentença seguinte:

```
UPDATE PALESTRA SET SALA_PAL = 'B-100' WHERE COD_PAL = 213;
```

Após essa modificação a tabela ficaria da seguinte forma:

COD_PAL	TIT_PAL	COD_PALESTRANTE	DATA_PAL	HORA_PAL	SALA_PAL
110	DESIGN PATTERNS	114	14/10/2003	18:30	D-200
213	QoS EM VOZ SOBRE IP	23	23/11/2003	19:30	B-100
235	DATA MINING UTILIZANDO REDES NEURAIAS	142	05/09/2004	20:00	D-200

3.2.3. Exclusão de Dados

Ainda, como parte do processamento normal dos sistemas, os dados, é possível que se precise eliminar linhas de uma tabela, ou por erro ou simplesmente como parte da lógica de algum tipo de rotina.

A sentença SQL que possibilita essa exclusão de dados nas linhas de uma tabela é apresentada através do exemplo a seguir.

Considerando que a tabela PALESTRA tem as seguintes linhas:

COD_PAL	TIT_PAL	COD_PALESTRANTE	DATA_PAL	HORA_PAL	SALA_PAL
110	DESIGN PATTERNS	114	14/10/2003	18:30	D-200
213	QoS EM VOZ SOBRE IP	23	23/11/2003	19:30	B-100
235	DATA MINING UTILIZANDO REDES NEURAIAS	142	05/09/2004	20:00	D-200

Ao submeter ao SGBD a seguinte sentença:

```
DELETE FROM PALESTRA WHERE SALA_PAL = 'D-200';
```

Teremos o seguinte resultado:

COD_PAL	TIT_PAL	COD_PALESTRANTE	DATA_PAL	HORA_PAL	SALA_PAL
213	QoS EM VOZ SOBRE IP	23	23/11/2003	19:30	B-100

3.2.4. Controle de Transações

Quando for necessária a execução de sentenças de atualização em bloco, ou seja, diversas sentenças INSERT, UPDATE e DELETE tiverem que ser submetidas ao SGBD de modo que todas sejam aceitas ou rejeitadas em conjunto, é necessário que se utilize outras sentenças SQL para esse fim:

- Para aceitar o bloco de atualizações é preciso submeter a sentença **COMMIT** após esse conjunto de sentenças;

- Para rejeitar o bloco de atualizações é preciso submeter a sentença **ROLLBACK** após esse conjunto de sentenças.

Observações:

- Alguns SGBDs tem a submissão de COMMIT automaticamente após a submissão de INSERTs, UPDATEs e DELETEs. Neste caso, o ROLLBACK não irá funcionar.
- Em alguns SGBDs a configuração de COMMIT automático é configurável.