

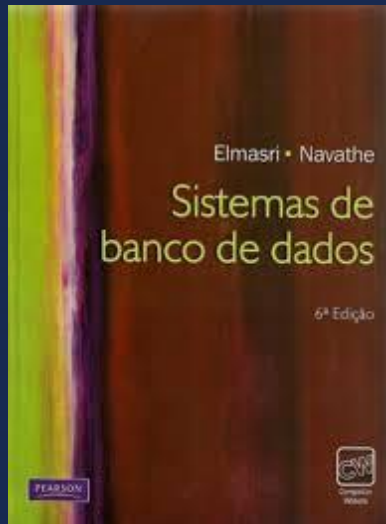


## Unidade 15 – Introdução às técnicas de Programação SQL



Prof. Aparecido V. de Freitas  
Doutor em Engenharia  
da Computação pela EPUSP  
[aparecidovfreitas@gmail.com](mailto:aparecidovfreitas@gmail.com)

# Bibliografia



Sistemas de Banco de Dados  
Elmasri / Navathe 6ª edição



Sistema de Banco de Dados  
Korth, Silberschatz – Sixth Edition

# Programação de Banco de Dados

- ❖ A maioria dos SGBD's possui um interface interativa na qual comandos **SQL** podem ser digitados interativamente em uma aplicação cliente (monitor) para execução no sistema gerenciador de banco de dados;
- ❖ A interface interativa é muito conveniente para a criação de esquemas ou consultas ad hoc ocasionais;
- ❖ Porém, na prática, a maioria das interações de banco de dados é executada por programas que foram cuidadosamente desenvolvidos e testados. Esses programas são conhecidos por aplicações de banco de dados.



# Técnicas para programação de Banco de Dados

- ✓ SQL embutido;
- ✓ API;
- ✓ Linguagem específica de Banco de Dados;



# SQL Embutido

- ❁ A técnica corresponde em embutir comandos do Banco de Dados em uma linguagem de programação de uso geral;
- ❁ Nessa técnica, os comandos do banco de dados são embutidos (embedded) na linguagem de programação hospedeira, mas são identificados por um prefixo especial;
- ❁ Por exemplo, o prefixo geralmente é **EXEC SQL** que precede todos os comandos **SQL** em um programa da linguagem hospedeira;

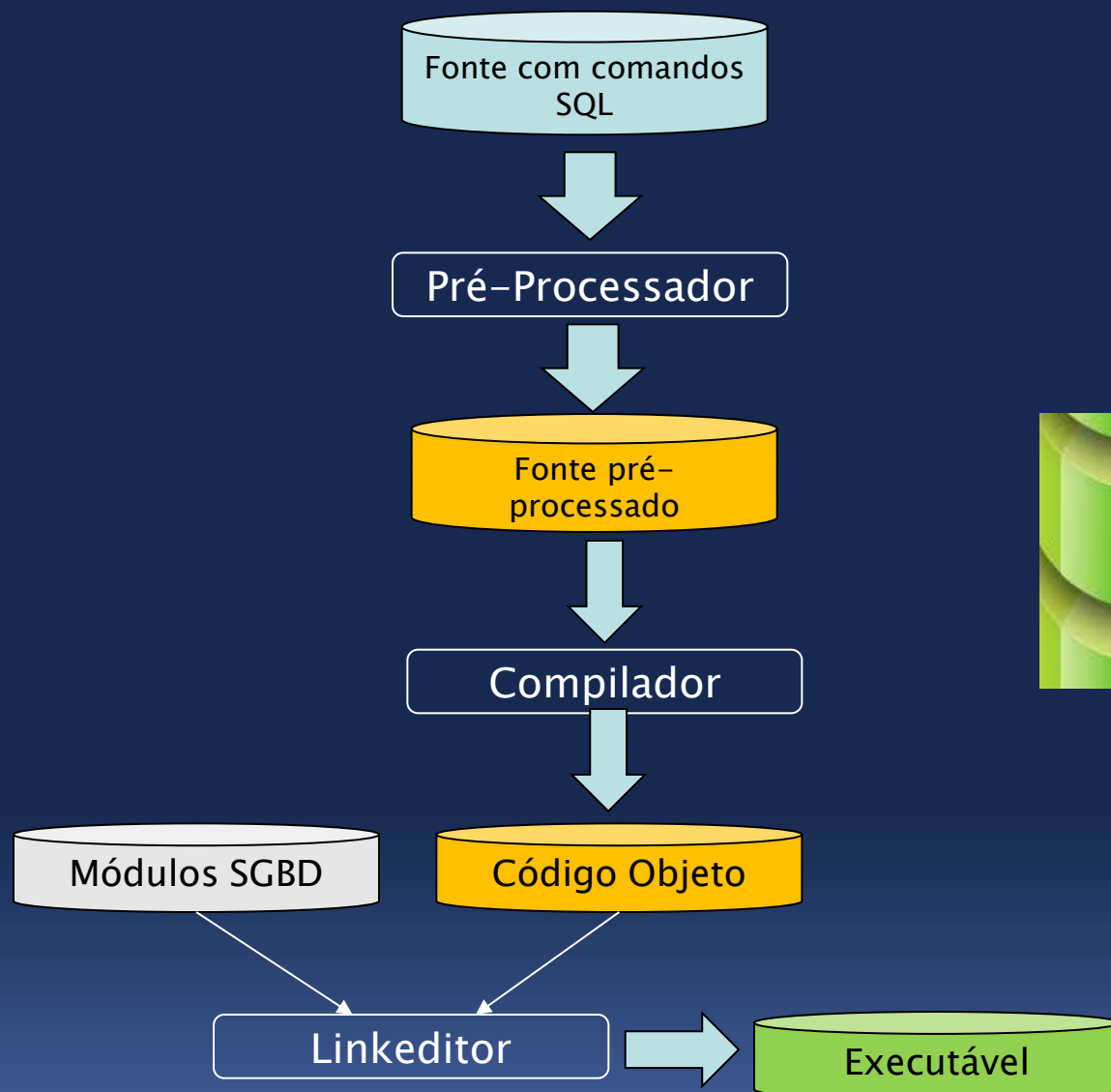


# SQL Embutido – Processamento

- ❖ Para a compilação do programa, um pré-processador deve varrer inicialmente o código para identificar os comandos do banco de dados e extraí-los para processamento pelo **SGBD**;
- ❖ Os comandos de banco de dados presentes no programa devem ser substituídos por chamadas de função ao **SGBD**;
- ❖ Essa técnica é geralmente conhecida por SQL embutido;



# Geração de Código com SQL Embutido



# API – Application Program Interface

- ❖ Uma **biblioteca de funções** se torna disponível à linguagem de programação hospedeira para chamadas de banco de dados;
- ❖ Os comandos reais de consulta e atualização do banco de dados são incluídos como parâmetros nas chamadas de função;
- ❖ Essa técnica é chamada de **interface de programação de aplicação (API)**



Fonte: Navathe/Elmasri



# Linguagem Nova

- ❖ Uma linguagem de programação totalmente **nova** é projetada do zero para ser compatível com o modelo de banco de dados e a linguagem de consulta;
- ❖ Estruturas de programação adicionais, como **loops** e **instruções condicionais**, são acrescentadas à linguagem de banco de dados para convertê-la em uma linguagem de programação completa;
- ❖ Exemplo: **PL/SQL da Oracle**.



# Técnicas para programação de Banco de Dados

- ✓ SQL embutido;
- ✓ API;
- ✓ Linguagem específica de Banco de Dados;



Das três técnicas quais são as mais convenientes ?

# Técnicas para programação de Banco de Dados



- ✓ SQL embutido;
- ✓ API;
- ✓ Linguagem específica de Banco de Dados;

- ✿ Na prática, as duas primeiras técnicas são mais **comuns**, pois muitas aplicações já são escritas em linguagens de programação de uso geral, mas exigem algum acesso ao banco de dados;
- ✿ A terceira técnica é mais apropriada para aplicações que possuem intensa interação com o banco de dados;
- ✿ Um dos principais problemas com as duas primeiras técnicas é a **DIVERGÊNCIA DE IMPEDÂNCIA**, que não ocorre na terceira técnica.

# Divergência de Impedância



- ❖ Corresponde ao problema que ocorre devido às diferenças entre o modelo de banco de dados e o modelo da linguagem de programação;
- ❖ Por exemplo, os tipos de dados da linguagem de programação podem diferir dos tipos de dados de atributos;
- ❖ Por exemplo, em **MySQL**:

Tipo	Bytes	De	Até
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807

Alguns desses tipos de dados podem não estar presentes na linguagens de programação !!!



Como resolver o problema de compatibilidade de tipos entre Linguagem de Programação e SQL ?



# Incompatibilidade de tipos entre SQL e Linguagem de Programação



- ❏ É necessário ter-se um vínculo entre a linguagem de programação hospedeira e os tipos de atributos compatíveis em **SQL**;
- ❏ Será necessário um vínculo diferente para cada linguagem de programação, uma vez que diferentes linguagens possuem diversos tipos de dados;
- ❏ Por exemplo, os tipos de dados disponíveis em **C** e **Java** são diferentes, e ambos diferem dos tipos de dados **SQL**, que são os tipos padrão para bancos de dados relacionais.



Fonte: Navathe/Elmasri

# Divergência de Impedância

- ✿ Outro problema ocorre porque os resultados da maioria das consultas **SQL** são **conjuntos** de **tuplas** (linhas) e cada tupla é formada por uma sequência de valores de atributo.

- ✿ Exemplo:

```
SELECT nomeCliente, codCliente FROM TABCLIENTE  
WHERE codCliente > 10  
ORDER BY nomeCliente;
```



Conjunto de  
registros



Antônio Carlos	2345
Beatriz Silva	1230
Carlos Souza	8764
Paula de Andrade	3487

# Divergência de Impedância

- ❖ No entanto, o programa geralmente trabalha com o **acesso de dados individuais** nas tuplas individuais para processamento ou geração de relatórios.
- ❖ Assim, é preciso que haja um **vínculo** para **mapear** a estrutura de dados do resultado da consulta, que é uma tabela, para uma estrutura de dados apropriada na linguagem de programação.
- ❖ É necessário que haja um mecanismo para percorrer as tuplas em um resultado de consulta a fim de se acessar uma tupla de cada vez e extrair valores individuais dela.



Fonte: Navathe/Elmasri



Como mapear a estrutura de dados da consulta SQL para uma estrutura de dados do programa ?



# Mapeamento da consulta para o programa

- Os valores de atributo da consulta SQL devem ser **copiados** para variáveis de programa apropriadas para que o programa continue processando;
- Um **cursor** ou **variável de iteração** normalmente é utilizada para **percorrer** as tuplas em um resultado de consulta;
- Os valores individuais dentro de cada tupla são então extraídos para variáveis do programa distintas do tipo apropriado;



```

EDIT          AGY0157.DEMO.SRCLIB(PROG45) - 01.07          Columns 00001 00072
Command ==>          Scroll ==> CSR
=====+-----+-----+-----+-----+-----+-----+-----+
=COLS> 1-----2-----3-----4-----5-----6-----7--
000025      EXEC SQL
000026          DECLARE EMP_CSR CURSOR FOR
000027              SELECT EMPID,ENAME,SALARY,JDATE
000028                  FROM   EMPLOYEE
000029          END-EXEC.
000030
000031      EXEC SQL
000032          INCLUDE SQLCA
000033      END-EXEC.
  
```

# Divergência de Impedância

- ✿ É um problema menor quando uma linguagem de programação de banco de dados especial é projetada para usar o mesmo modelo e tipos de dados do banco de dados;
- ✿ Por exemplo, **PL/SQL** da Oracle;



# Sequência típica de Operações

- ❁ Uma arquitetura comum para o acesso ao banco de dados é o **modelo cliente/servidor**, no qual um programa cliente trata da lógica de uma aplicação de software, mas inclui algumas **chamadas** para o servidor de banco de dados;
- ❁ Ao se escrever tal programa, emprega-se a seguinte sequência de operações:
  1. O programa precisa inicialmente estabelecer **conexão** com o sistema gerenciador de banco de dados. Isso envolve especificar o endereço da máquina onde o servidor está localizado, além de fornecer nome de conta e senha para acesso ao banco de dados;
  2. Uma vez estabelecida a conexão, o programa pode interagir com o banco de dados, **submetendo consultas** e outros comandos de banco de dados;
  3. Quando o programa não precisar mais acessar o banco de dados, ele deverá terminar ou **fechar** essa **conexão**.

