

Bancos de Dados

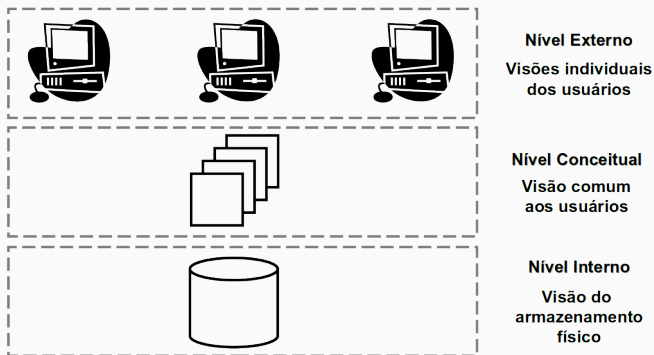
Prof. Leandro Correia

Novembro, 2020

Arquiteturas de Sistemas de Bancos de Dados

Arquitetura ANSI/SPARC

Arquitetura dividida em 3 níveis: interno, conceitual e externo.
Permite ao usuário interagir com o SGBD em diferentes níveis de abstração, ocultando seus detalhes de implementação.



Nível Externo:

- O nível mais próximo do usuário final.
- Visões individuais das informações do banco de dados.
- Maior abstração em relação ao nível interno.

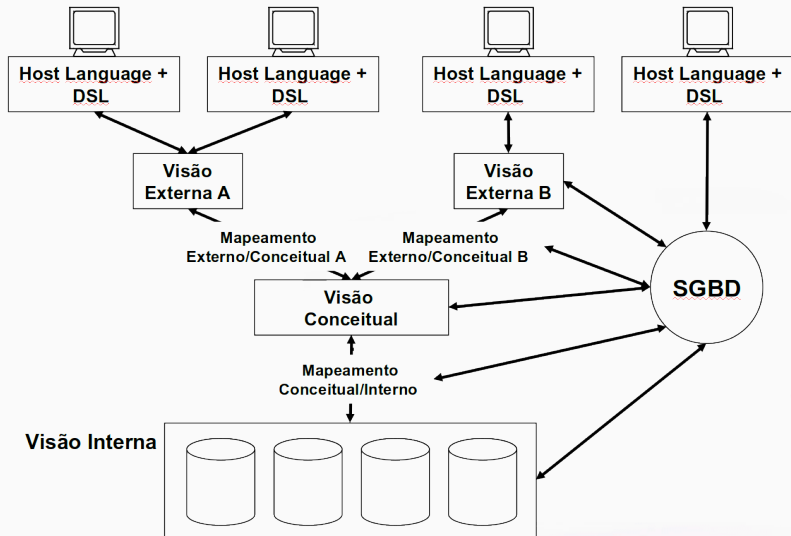
Nível Conceitual:

- Nível intermediário entre o interno e o externo.
- Visão de todo o conteúdo do banco de dados.
- Dados, regras de segurança e integridade.
- Encapsula os detalhes de implementação do nível interno.

Nível Interno:

- Nível mais próximo do armazenamento físico e mais afastado do usuário final.
- Define COMO os dados serão armazenados e recuperados.
- Estruturas de dados e métodos de acesso são descritos em detalhes nesse nível.

Arquitetura ANSI/SPARC



- Host Language (Linguagem Hospedeira): linguagem utilizada para desenvolvimento de aplicações (C++, Java, Python, etc).
- DSL (Data Sublanguage): linguagem utilizada especificamente para definir e manipular os objetos do banco de dados (SQL, Cypher, etc).
- Mapeamento Externo/Conceitual: define a correspondência entre uma visão particular externa e a visão conceitual do banco de dados (Visões, Funções, Permissões, etc).
- Mapeamento Conceitual/Interno: define a correspondência entre a visão conceitual e o banco de dados armazenado (Tablespaces, Filegroups, etc).

Arquitetura Cliente/Servidor

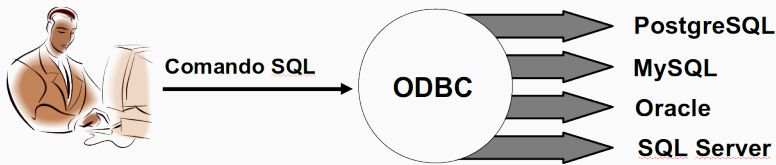
- Arquitetura na qual a funcionalidade do sistema e seu processamento são divididos entre o cliente (front-end) e o servidor (back-end).
- As funcionalidades do sistema, lógica de programação, regras de negócios e gerenciamento de dados são segregados entre as máquinas do cliente e do servidor. Formada por processos distintos cliente e servidor.
- A parte cliente e a parte servidor geralmente (mas nem sempre) operam em diferentes plataformas de computador.

Arquitetura Cliente/Servidor

- Tanto a máquina do cliente como a do servidor podem ser atualizadas independentemente, sem que seja necessário atualizar a outra.
- O servidor pode atender a vários clientes simultaneamente, e os clientes podem acessar vários servidores.
- Os sistemas cliente/servidor incluem algum tipo de capacidade para operar em rede.

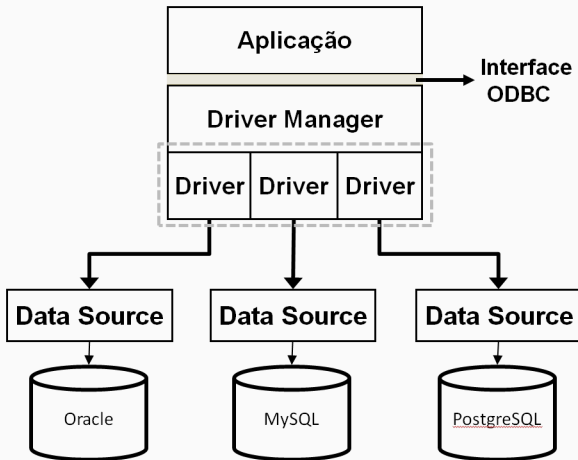
Arquitetura Cliente/Servidor

- ODBC (Open Database Connectivity): interface para acesso a dados em ambientes heterogêneos de Sistemas Gerenciadores de Banco de Dados (SGBD).



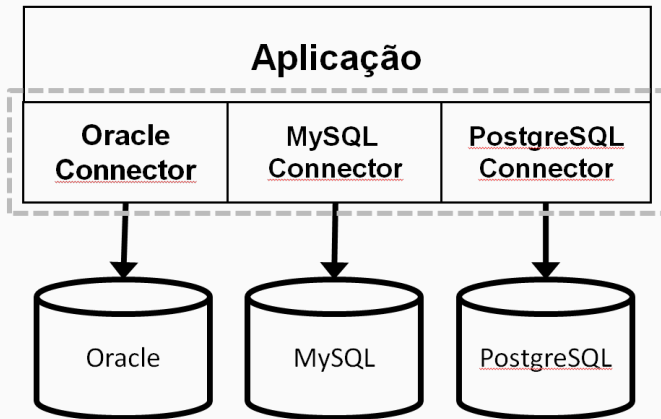
Arquitetura Cliente/Servidor

Arquitetura Baseada em ODBC



Arquitetura Cliente/Servidor

Arquitetura Baseada em Acesso Nativo



Acesso via ODBC

- Padronização
- Encapsulamento
- Portabilidade entre bancos de dados

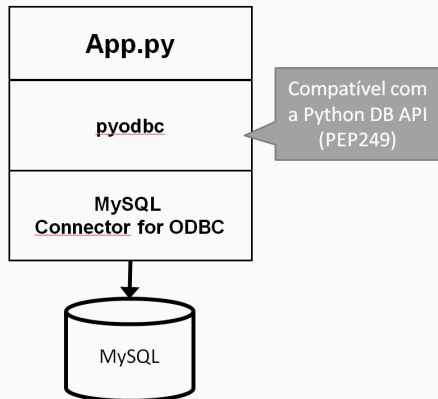
Acesso Nativo

- Performance
- API proprietária (específica para cada banco de dados)
- Maior complexidade na implementação

Arquiteturas de Banco de Dados

Exemplo 1:

- Linguagem: Python
- Banco de dados: MySQL
- Conexão via ODBC



Arquiteturas de Banco de Dados

Exemplo 2:

- Linguagem: Python
- Banco de dados: MySQL
- Conexão via acesso nativo

