Universidade Estadual da Paraíba

Banco de Dados

Modelos, Esquemas e Linguagens

Prof. Dr. Vladimir Costa Alencar

valencar@gmail.com

https://www.valencar.com/

Modelo de dados

É um conjunto de conceitos para descrever um BD

- É o Modelo (estrutura) de referência a partir do qual os dados são organizados logicamente;

- Instrumento que permite uma representação do mundo real a partir de informações;

- Permite a interação entre analistas e usuários;

Modelo de dados

-Nele são representados basicamente:

1. Entidades (propriedades e restrições de integridade)
2. Relacionamentos entre as entidades
3. Eventos (resultados de triggers, etc)
4. Regras de estruturação e acesso a dados

Modelo de dados

É uma coleção de ferramentas conceituais para descrever:

- Dados
- Relações de dados
- Semântica de dados
- Restrições de consistência

Modelo de dados

Um modelo de dados oferece uma maneira de descrever o projeto de um banco de dados no nível físico, lógico e de visões.

1. Modelo Relacional

- Usa uma coleção de tabelas para representar os dados e as relações entre eles.
- Cada Tabela possui colunas
- É um modelo baseado em Registros (Tuplas)

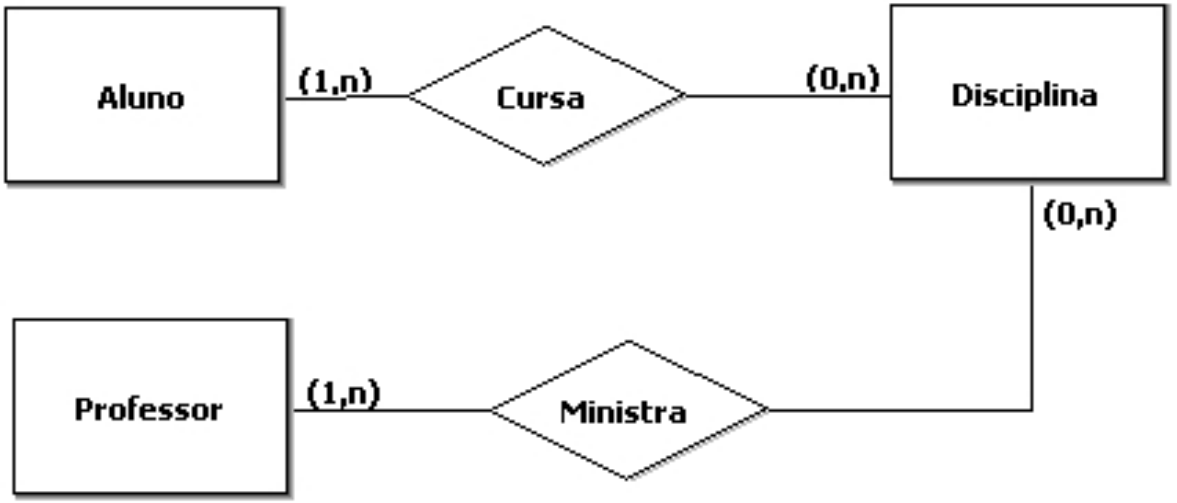
Table: Cliente

Database Connection: Agencia-Cliente

Cod_Cliente	Nome	CPF	Rua	Número	Cidade	UF
1	Paula	22222222222	Av Chile	234	Campina Grande	PB
2	João	56745454533	Rua 14	888	Patos	PB
3	Maria	23412356709	Rua Floriano Peixoto	123	Campina Grande	PB
4	Antonio	34567891234	Rua Y	902	João Pessoa	PB
5	Marcos	29102034455	Rua H	321	João Pessoa	PB

2. Modelo Entidade/Relacionamento

- É baseado na percepção de um mundo real
- Consiste em uma coleção de objetos básicos
- Esses objetos são chamados de entidades
- Existem relações entre as entidades

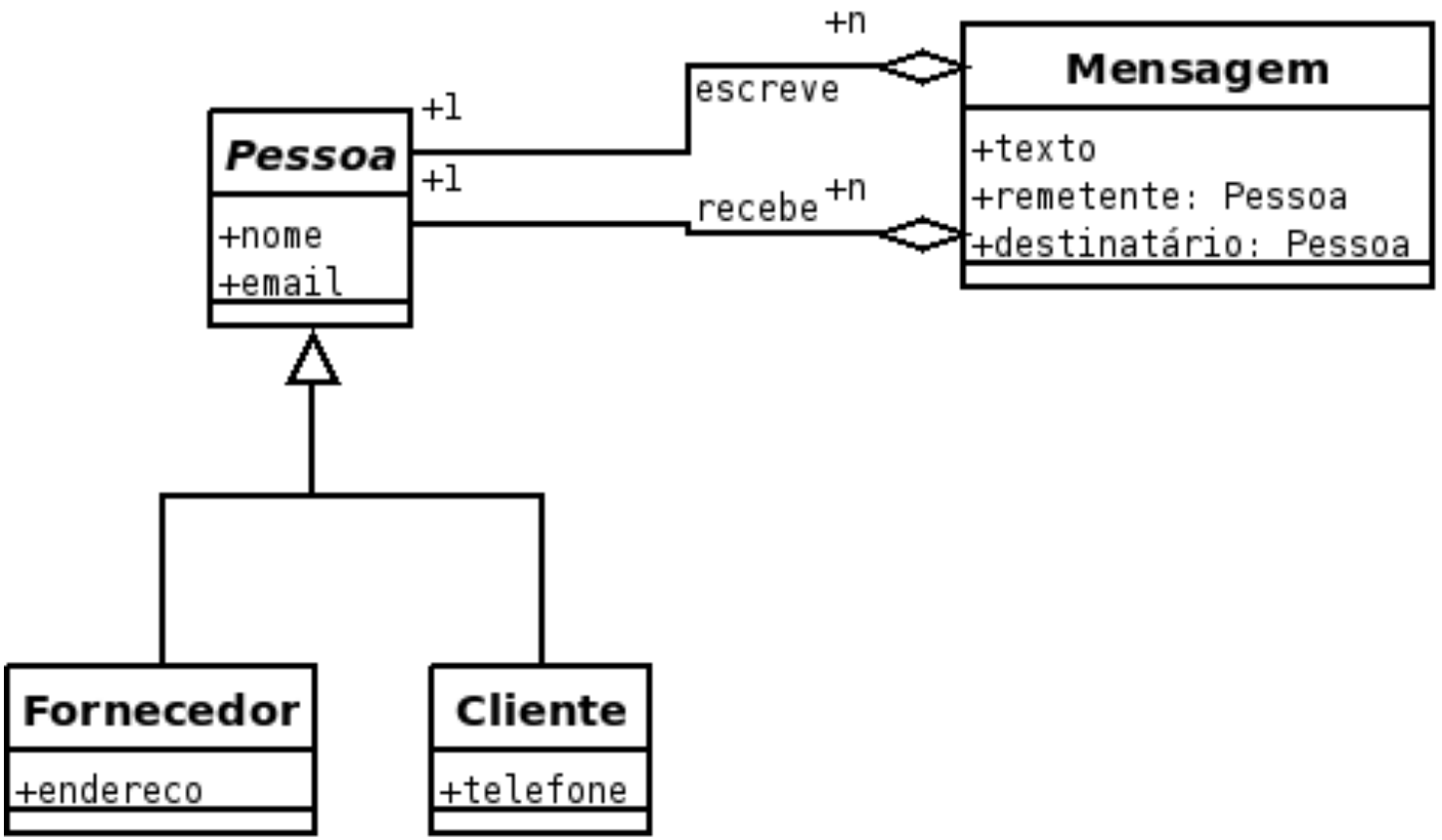


3. Modelo Baseado em Objetos

- Ele pode ser visto como uma extensão do modelo E-R com noções de:
  1. Encapsulamento
  2. métodos (funções)
  3. identidade de objeto.

- O modelo Relacional de Objetos combina recursos do MER com o modelo de objetos

3. Modelo Baseado em Objetos



4. Modelo de dados Semi-estruturado

- Nota-se atualmente que boa parte dos dados disponíveis para acesso eletrônico não estão mantidos em BDs.

- Alguns exemplos são diretórios de arquivos de documentos (atas de reuniões, processos, etc) de uma organização ou informações acessíveis através da Web.

- A justificativa para tal fato decorre da própria natureza destes dados.

4. Modelo de dados Semi-estruturado

- Dados Web, por exemplo, apresentam uma organização bastante heterogênea, que pode variar de um texto sem nenhuma formatação até um conjunto de registros bem formatados

- Além disso, o volume destes dados pode ser grande e com muitos relacionamentos

- A alta heterogeneidade desses dados torna complexa as atividades de pesquisa de dados, uma vez que não existe um esquema uniforme a partir do qual uma consulta possa ser formulada

4. Modelo de dados Semi-estruturado

- Dados semi-estruturados apresentam uma representação estrutural heterogênea, não sendo nem completamente não-estruturados nem estritamente tipados.

- A Extensible Markup Language (XML) é amplamente usada para representar dados semi-estruturados.

4. Modelo de dados Semi-estruturado

Dados Semi-Estruturados

Ronaldo Mello
Carlos Heuser

UFRGS
e-mails: {ronaldo,heuser}@inf.ufrgs.br

Resumo
---

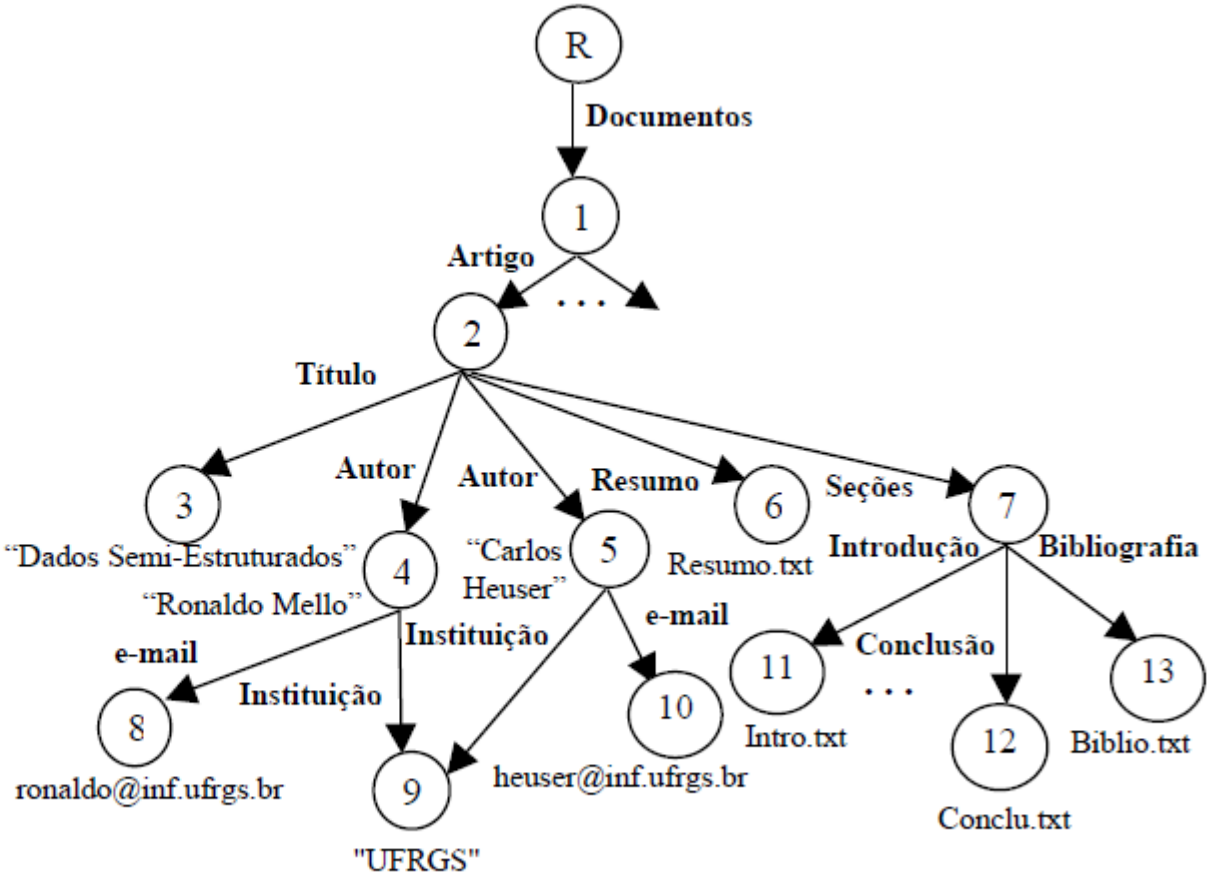
1 Introdução

. . .

n Conclusão

Bibliografia

(a)



(b)

Exemplo de um objeto semi-estruturado (a) e sua representação em um modelo de dados baseado em grafo (b)

4. Modelo de dados Semi-estruturado

Ex. A Extensible Markup Language (XML)

```
<?xml version="1.0"?>
<artigo xmlns="artigo.xsd">
  <título> Dados Semi-Estruturados </título>
  <autor>
      <nomea> Ronaldo Mello </nomea>
      <instituição>  <nomei> UFRGS </nomei>
                    <endereço> Bento Gonçalves, 9500 </endereço>
      </instituição>
      <instituição>  <nomei> UFSC </nomei>
      </instituição>
      <email> ronaldo@inf.ufrgs.br </email>
  </autor>
  <autor>
      <nomea> Carlos A. Heuser </nomea>
      <instituição>   <nomei> UFRGS </nomei>
                     <endereço> Bento Gonçalves, 9500 </endereço>
      </instituição>
      <email> heuser@inf.ufrgs.br </email>
  </autor>
  <resumo> ... </resumo>
  <seção> <nomes> Introdução </nomes>
          <texto> ... </texto> </seção>
   ...
</artigo>
```

4. Modelo de dados Semi-estruturado

Sample JSON Schema

```
{
        "title": "Example Schema",
        "type": "object",
        "properties": {
                "firstName": {
                        "type": "string"
                },
                "lastName": {
                        "type": "string"
                },
                "age": {
                        "description": "Age in years",
                        "type": "integer",
                        "minimum": 0
                }
        },
        "required": ["firstName", "lastName"]
}
```

Modelo de dados

- Historicamente, o **modelo de dados de rede** e o **modelo de dados hierárquico** precederam o **modelo relacional**.

-Estes modelos estavam intimamente relacionados com a implementação

-Eles complicavam a tarefa de modelar dados

-São pouco usados atualmente

Esquema de dados

- É a coleção de informações armazenadas no banco de dados em um determinado momento

-É a Descrição de um BD segundo um modelo de dados

- Contém a Descrição da estrutura de um BD

-Pode ser textual ou gráfico

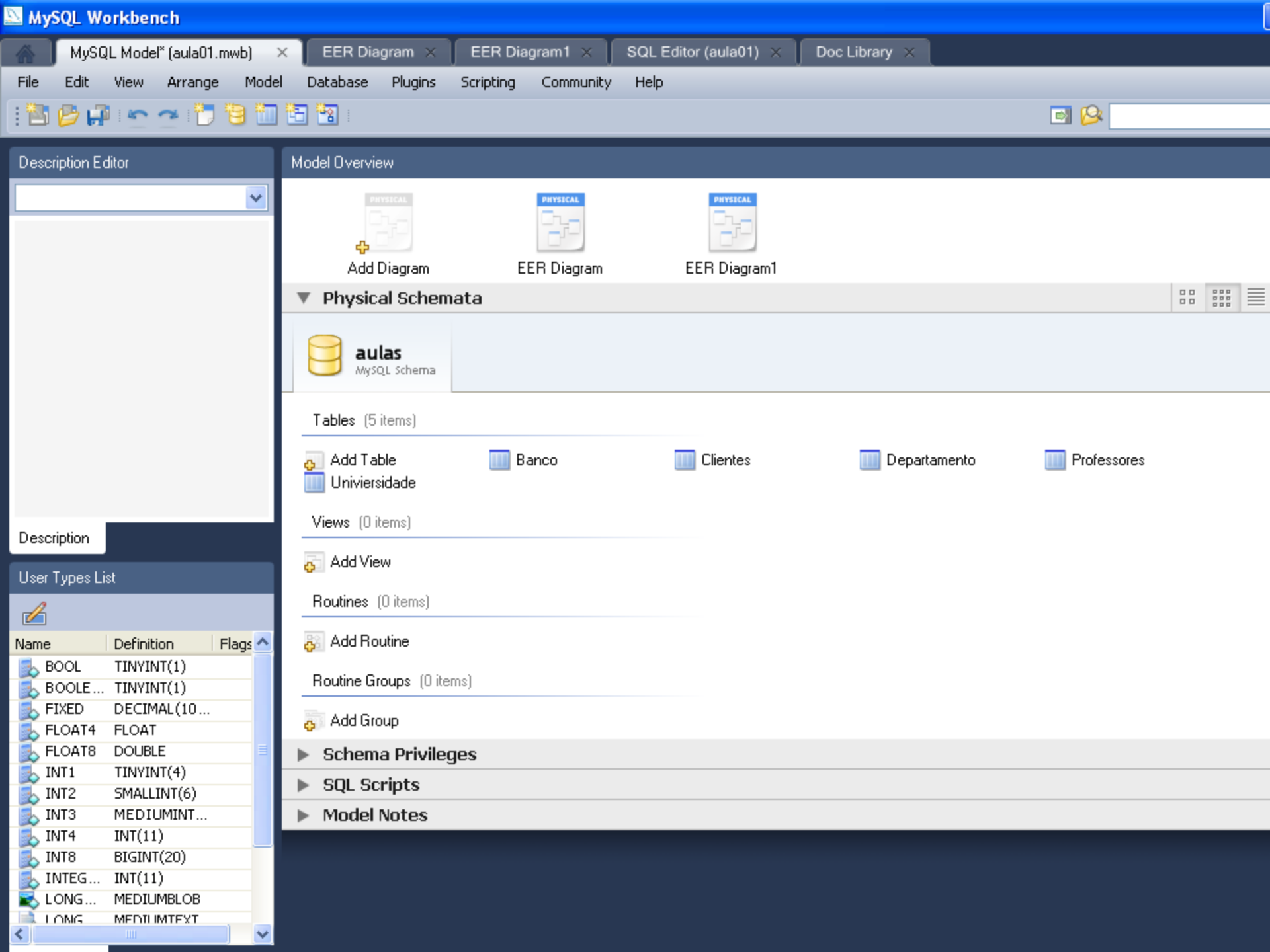
Esquema de dados

- Fazendo analogia com uma linguagem de programação:

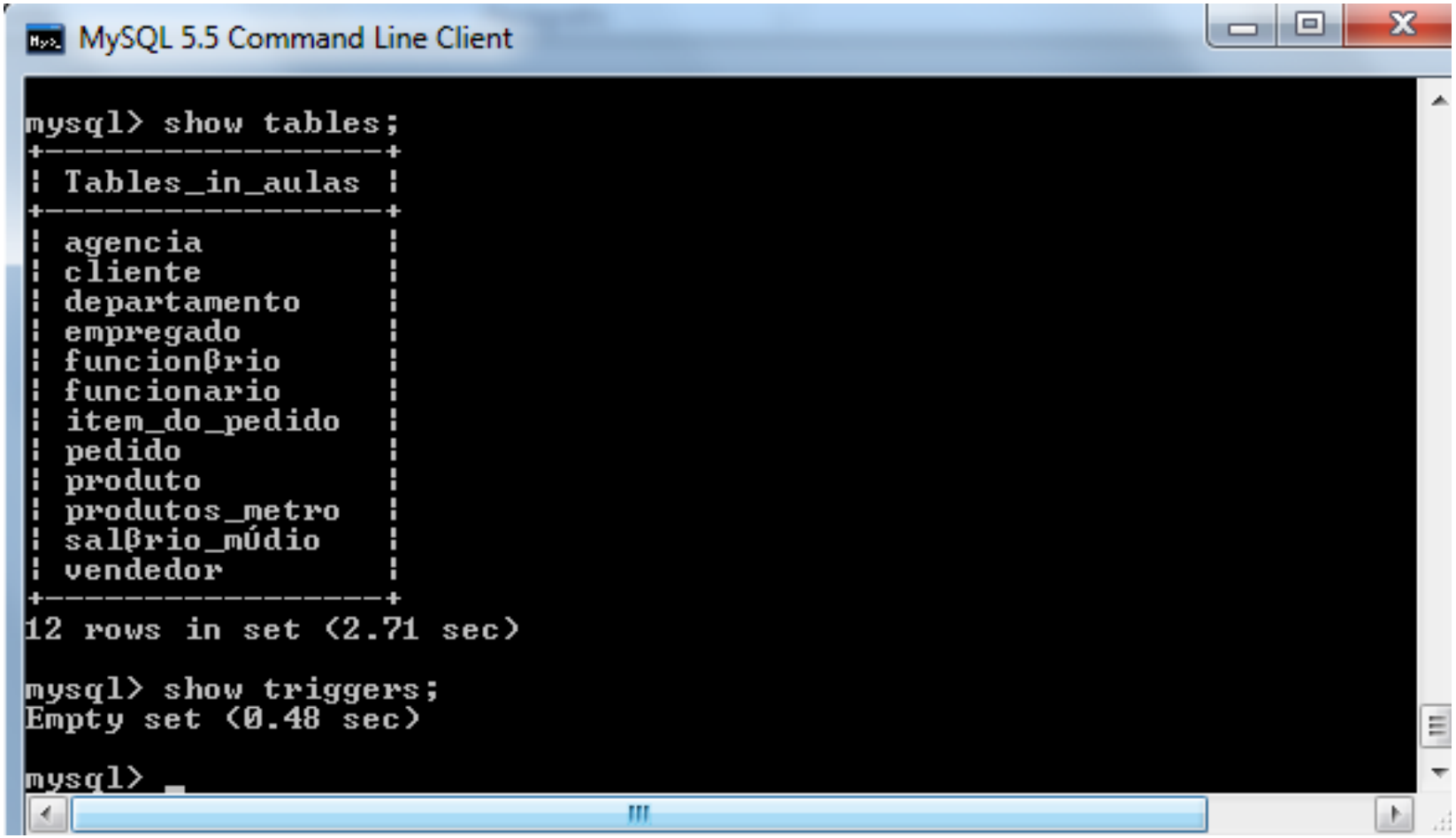
- Corresponde às declarações de variável

- Juntamente com as definições de tipo associadas

- Cada variável possui um valor em um dado instante (instância)


MySQL Workbench
MySQL Model* (aula01.mwb)
EER Diagram
EER Diagram1
SQL Editor (aula01)
Doc Library
File
Edit
View
Arrange
Model
Database
Plugins
Scripting
Community
Help
Description Editor
Description
User Types List
Name
Definition
Flags
BOOL
TINYINT(1)
BOOLE...
TINYINT(1)
FIXED
DECIMAL(10...
FLOAT4
FLOAT
FLOAT8
DOUBLE
INT1
TINYINT(4)
INT2
SMALLINT(6)
INT3
MEDIUMINT...
INT4
INT(11)
INT8
BIGINT(20)
INTEG...
INT(11)
LONG...
MEDIUMBLOB
Model Overview
Add Diagram
EER Diagram
EER Diagram1
Physical Schemata
aulas
MySQL Schema
Tables (5 items)
Add Table
Banco
Clientes
Departamento
Professores
Univiersidade
Views (0 items)
Add View
Routines (0 items)
Add Routine
Routine Groups (0 items)
Add Group
Schema Privileges
SQL Scripts
Model Notes

Esquema de dados



Instância de um BD

É o estado do esquema de um BD num dado instante.

São os dados atuais armazenados no BD em um momento particular.

Instância de um BD

Table: Cliente

Database Connection: Agencia-Cliente

Cod_Cliente	Nome	CPF	Rua	Número	Cidade	UF	Saldo
1	Paula	22222222222	Av Chile	234	Campina Grande	PB	3000
2	João	56745454533	Rua 14	888	Patos	PB	1200
3	Maria	23412356709	Rua Floriano Peixoto	123	Campina Grande	PB	3900
4	Antonio	34567891234	Rua Y	902	João Pessoa	PB	3980
5	Marcos	29102034455	Rua H	321	João Pessoa	PB	4000

Instância da tabela Cliente

Abstração de dados

- O sistema precisa recuperar dados de maneira eficiente (sistema funcional)

- Necessidade de eficiência → projetistas usam estruturas de dados complexas

- Os projetistas ocultam a complexidade dos usuários sob níveis de abstração:

1. Nível físico
2. Nível lógico
3. Nível de View (visão)

Abstração de dados

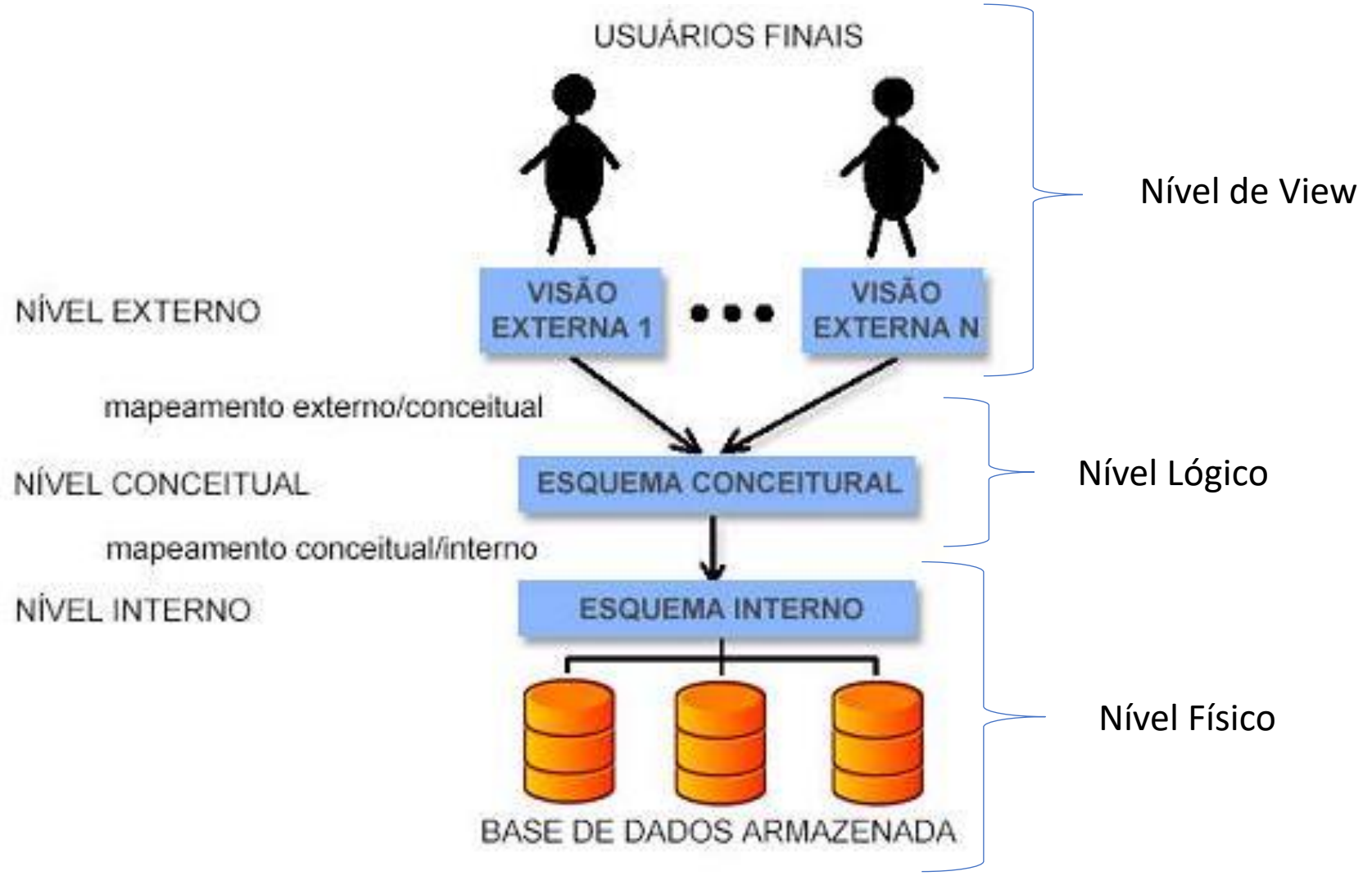
- O sistema precisa recuperar dados de maneira eficiente (sistema funcional)

- Necessidade de eficiência → projetistas usam estruturas de dados complexas

- Os projetistas ocutam a complexidade dos usuários sob níveis de abstração:

1. Nível físico
2. Nível lógico
3. Nível de View (visão)

Abstração de Dados



1. Nível físico

- Descreve como os dados são realmente armazenados
- Detalhes de estruturas complexas de baixo nível

1. Nível físico

- Forma física de armazenamento dos dados

```
RELATION VOOS   [
   KEY = {VOO}
   ATTRIBUTES = {
        VOO:            CHAR (5)
        TARIFA:         NUMERIC (8)
        ASS:            NUMERIC (6)
        CIA:            CHAR (20)
    }
]
INTERNAL REL VOOS   [
   INDEX ON V# CALL VOINX
   FIELD = {
        HEADER:         BYTE (1)
        V#:             BYTE (5)
        TARIFA:         NUMERIC (8)
        ASS:            NUMERIC (6)
        CIA:            CHAR (20)
    }
]
```

2. Nível Lógico

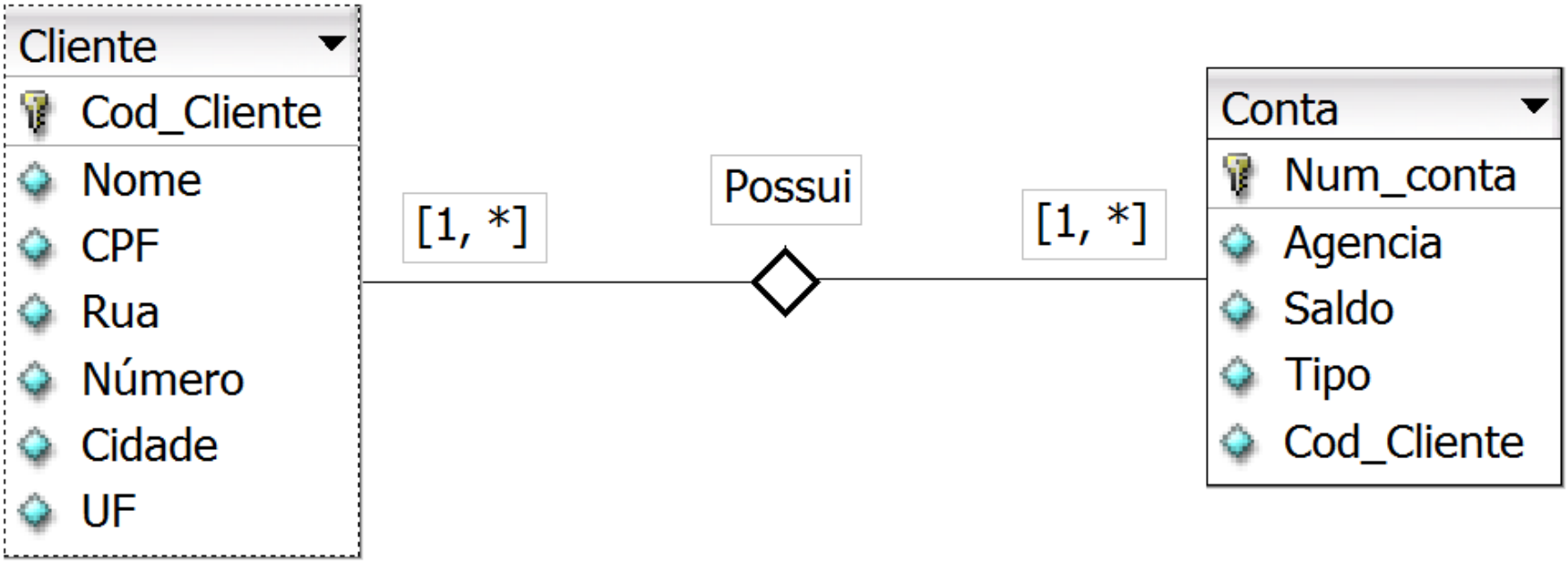
- Descreve que dados estão armazenados no banco de dados e que relações existem entre eles.

- Descreve o banco de dados inteiro em termos de um pequeno número de estruturas relativamente simples

- O BD é estruturado por meio de registros de formato fixo, simplificando a implementação do banco de dados no nível físico.

2. Nível Lógico

Conceitual: Quais os dados armazenados e relações entre eles

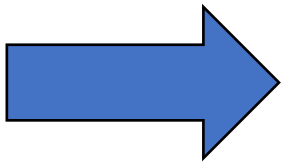


3. Nível de View (visão)

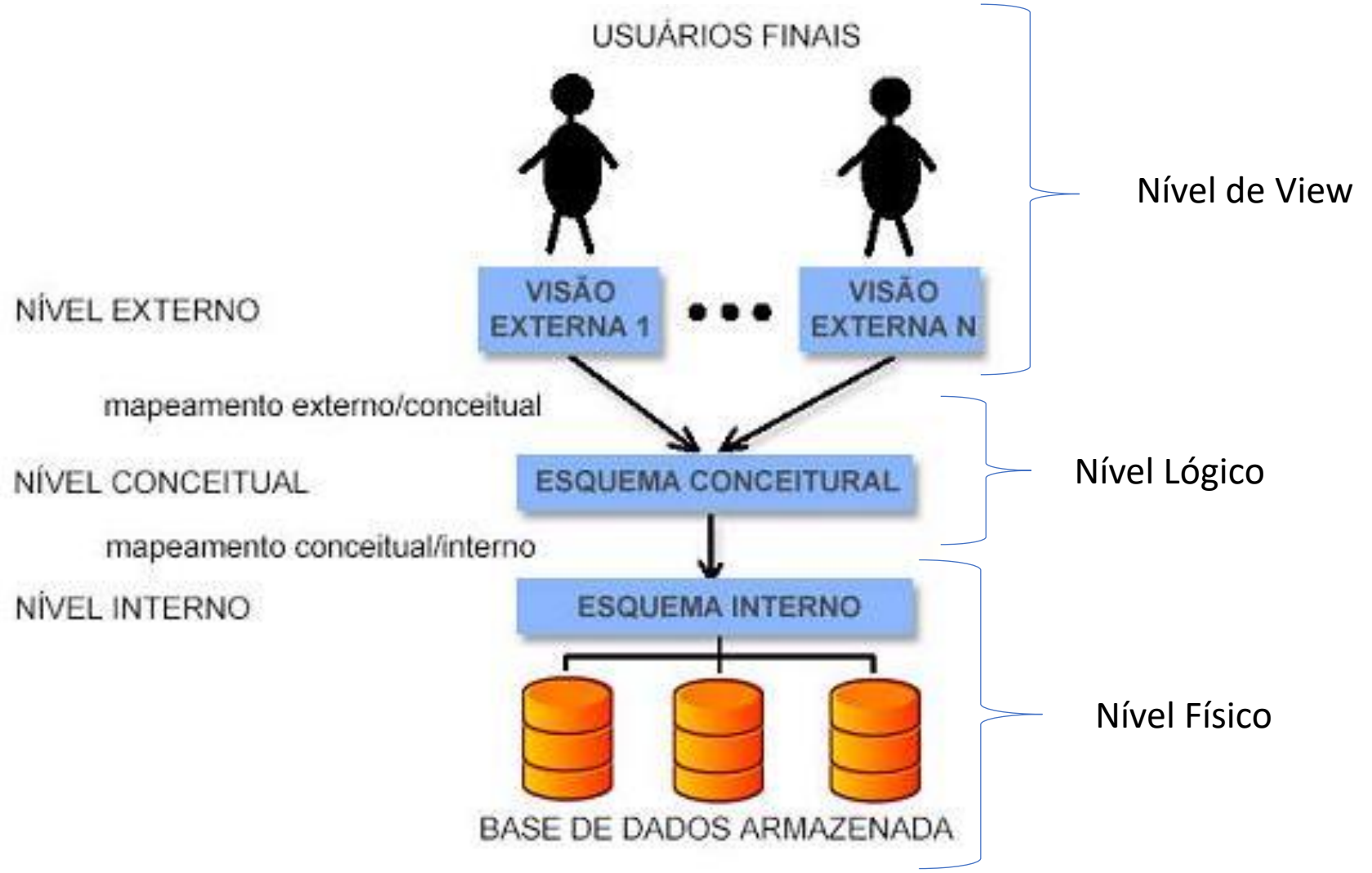
- Nível de abstração mais alto
- Descreve apenas parte do banco de dados
- O sistema pode oferecer muitas visões para o mesmo banco de dados
- Existe para simplificar sua interação com o sistema

3. Nível de View (visão)

Visões do utilizador



Níveis de Abstração de Dados



Níveis de Abstração de Dados

Gerente

Supervisor

EMPREGADO
Matrícula
Salário
Departamento

EMPREGADO
Matrícula
Departamento

Nível de View

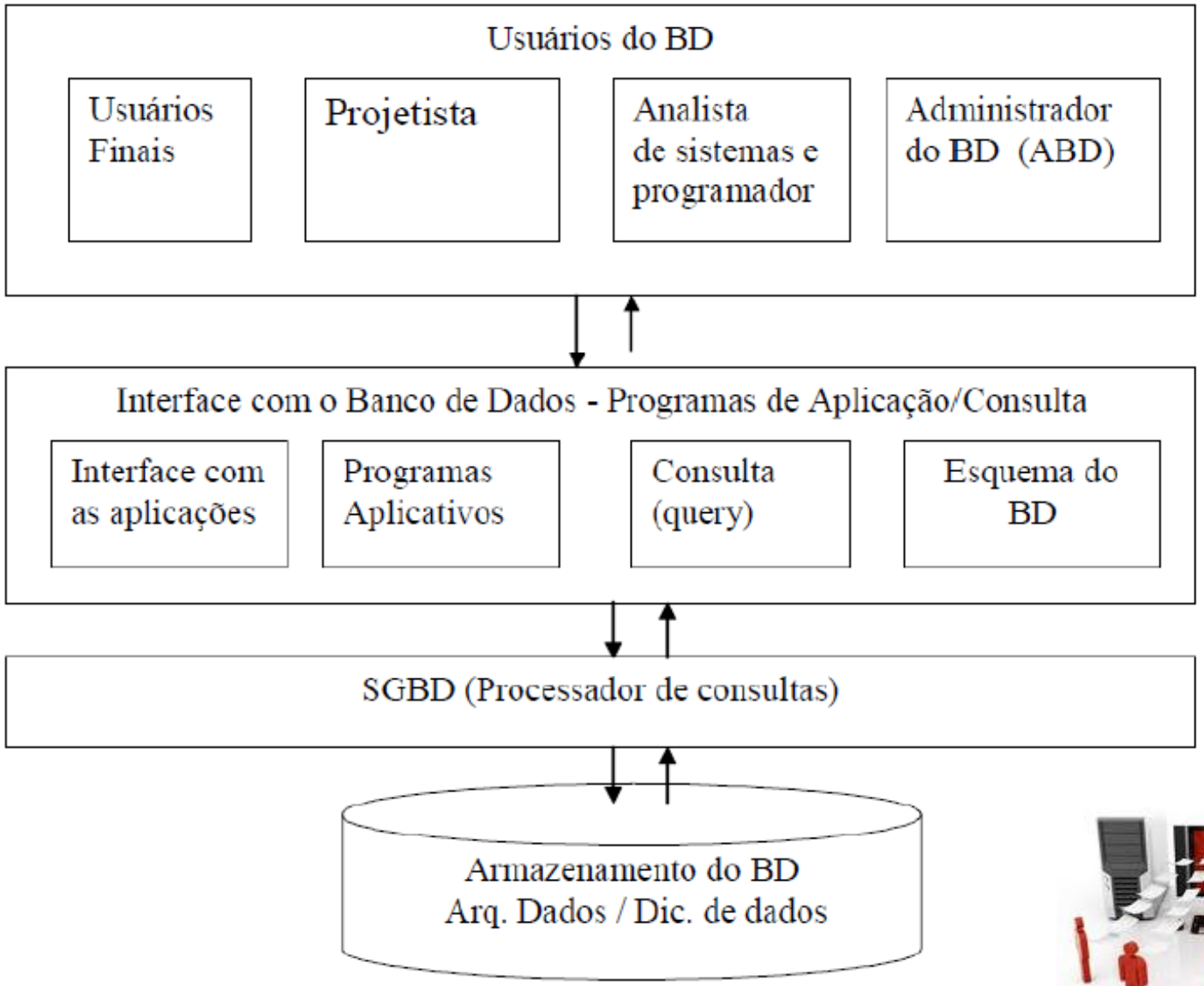
EMPREGADO
Matrícula Inteiro
Salário Real
Departamento Caracter (20)

Nível Lógico

EMPREGADO_ARMAZENADO
EMP# Indice EMPX
...

Nível Físico

BD – Características - usuários



Visão Geral da estrutura de um BD

Usuários de Banco de Dados

1) Administrador do BD (DBA):

- Definição do esquema conceitual
- Definição da estrutura de armazenamento e métodos de acesso
- Modificação do esquema conceitual, estrutura de armazenamento e métodos de acesso
- Concessões de autorização de acesso
- Especificação das estratégias de recovery
- Manutenção (Backups, gerenciamento de espaço livre, Desempenho)

Usuários de Banco de Dados

2) Projetista de BD
-Projeta os Esquemas Lógicos e Externo do BD

3) Analista de Sistemas
-Especifica programas que acessam o BD
(*programas de aplicação ou aplicativos)*
-*Usa ferramentas de consulta*

4) Programador de Aplicações
- Implementa aplicativos para acesso aos dados

Usuários de Banco de Dados

5) Usuários Finais

- Leigos: utilizam o SGBD, através de aplicações através de menus formulários, relatórios, etc.

- Usuários Sofisticados: programam suas próprias consultas no SGBD utilizando linguagens declarativas (SQL).

Linguagens de Banco de Dados

Um SGBD fornece:

- Uma **linguagem de definição de dados** para especificar o esquema de banco de dados;
- Uma **linguagem de manipulação de dados** para manipular as consultas e
- Uma **linguagem de controle de dados** para aspectos de autorização dos dados.

- Na prática, fazem parte de uma única linguagem, com exemplo a linguagem SQL

Linguagem de Controle de Dados (DCL)

- Usada pelo Administrador de Banco de Dados (DBA) para controlar:
- Os aspectos de autorização de dados
- Licenças de usuários para controlar quem tem acesso para ver ou manipular dados dentro do banco de dados.

Linguagem de Controle de Dados (DCL)

Duas palavras-chaves da DCL:

GRANT - autoriza ao usuário executar ou setar operações.

REVOKE - remove ou restringe a capacidade de um usuário de executar operações.

- Outros comandos DCL:

ALTER PASSWORD
DROP VIEW

Linguagem de Manipulação de Dados (DML)

- Permite aos usuários acessar ou manipular dados conforme são organizados pelo modelo de dados apropriado

- Os tipos de acesso são:

1. Recuperação de informações armazenadas no banco de dados
2. Inserção de novas informações no BD
3. Exclusão de informações no BD
4. Modificação de informações armazenadas no banco de dados

Linguagem de Manipulação de Dados (DML)

Ex. DML (Linguagem SQL)

```
Select Nome, Rua, Número
From Cliente
Where Saldo > 1000;
```

Nome	Rua	Número
João	Av das Américas	888
Maria	Rua Floriano Peixoto	123
Antonio	Rua Alfredo Dantes	902
Marcos	Rua Argemiro Figueiredo	321

Linguagem de Manipulação de Dados (DML) Embutida

- O padrão SQL define incorporações da SQL, em diversas linguagens de programação, como C, Cobol, Pascal, Java, Fortran (linguagens host).

- Essas linguagens host podem acessar e atualizar dados armazenados em um BD

- Essa forma embutida da SQL extende ainda mais a capacidade do programador de manipular Banco de dados.

Linguagem de Manipulação de Dados (DML) Embutida

- Código na linguagem C:

```
Char *consulta_sql = "select nome from conta
                              where saldo >  ?";
EXEC SQL prepare prog-sql from :consulta_sql;
Float saldo = 1000
EXEC SQL execute prog-sql using :saldo
```

- Esta sintaxe exige extensões para a linguagem ou um pré-processador

Linguagem de Manipulação de Dados (DML) Embutida - Java

A API JDBC (java) encapsula:

1. o estabelecimento da conexão com o BD
2. o envio de comandos SQL
3. o processamento dos resultados

Linguagem de Manipulação de Dados (DML) Embutida (Java)

```
String url = "jdbc:mysql://" + “localhost” + "/" + “aulas”;

try {
     con = DriverManager.getConnection(url, “root”, “root-password”);
     stmt = con.createStatement();

     rs = stmt.executeQuery("SELECT * FROM pessoa");
     rs.beforeFirst();

     while (rs.next()) {
               int codigo = rs.getInt("cod");
               String nome = rs.getString("nome");
               System.out.println(codigo+" - "+nome );
    }
    con.close();
} catch (SQLException e) {
               e.printStackTrace();
               System.out.print("Erro ao conexao.");
}
```

Linguagem de Manipulação de Dados (DML) Embutida

- Dois padrões muito utilizados para se conectar a um banco de dados SQL são:

ODBC

JDBC

ODBC

- O padrão Open Database Connectivity (ODBC) define uma maneira para um programa se comunicar com um servidor de banco de dados.

- O ODBC define uma **interface de programa de aplicação** (API) que as aplicações podem usar para abrir uma conexão com um banco de dados, enviar consultas e atualizações e trazer resultados

JDBC

- O padrão JDBC define uma API que os programas Java podem usar para se conectar a servidores de bancos de dados.

- O usuário se conecta a um servidor SQL, estabelecendo uma sessão SQL, executa uma série de instruções, e desconecta a sessão.

Linguagem de Definição de Dados (DDL)

- Através dela é especificado um esquema de banco de dados por um conjunto de definições

- Essas instruções definem os detalhes de implementação dos esquemas de bancos de dados, que normalmente estão ocultos dos usuários

- Valores de dados armazenados no banco de dados precisam satisfazer certas **restrições de consistência**

Linguagem de Definição de Dados (DDL) – Restrições de consistência

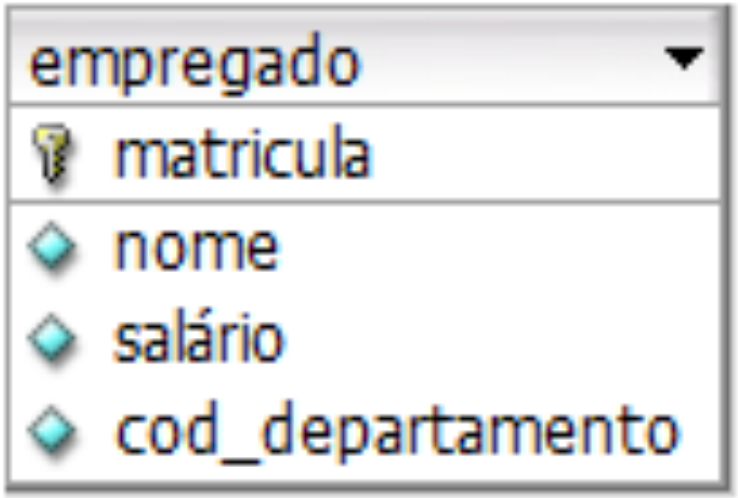
- **Restrições de domínio**
  Inteiro, caracter, data, hora, ...

- **Integridade Referencial**
  Relação entre tabelas

- **Assertivas / Triggers**
  Condição que o BD precisa satisfazer
  Ex. “Toda conta não pode ter **saldo** negativo”
  → qualquer modificação no BD só será permitida se a assertiva não for violada

Linguagem de Definição de Dados (DDL)

```
CREATE TABLE empregado (
  matricula INTEGER(10) UNSIGNED NOT NULL,
  nome CHAR(20) NOT NULL,
  salário FLOAT NOT NULL,
  cod_departamento INTEGER(10) UNSIGNED NOT NULL,
  PRIMARY KEY(matricula)
);
```



Linguagem e Interface

= DDL + DML + DCL

SQL = *Structured Query Language*, ou Linguagem de Consulta Estruturada