

# Banco de Dados

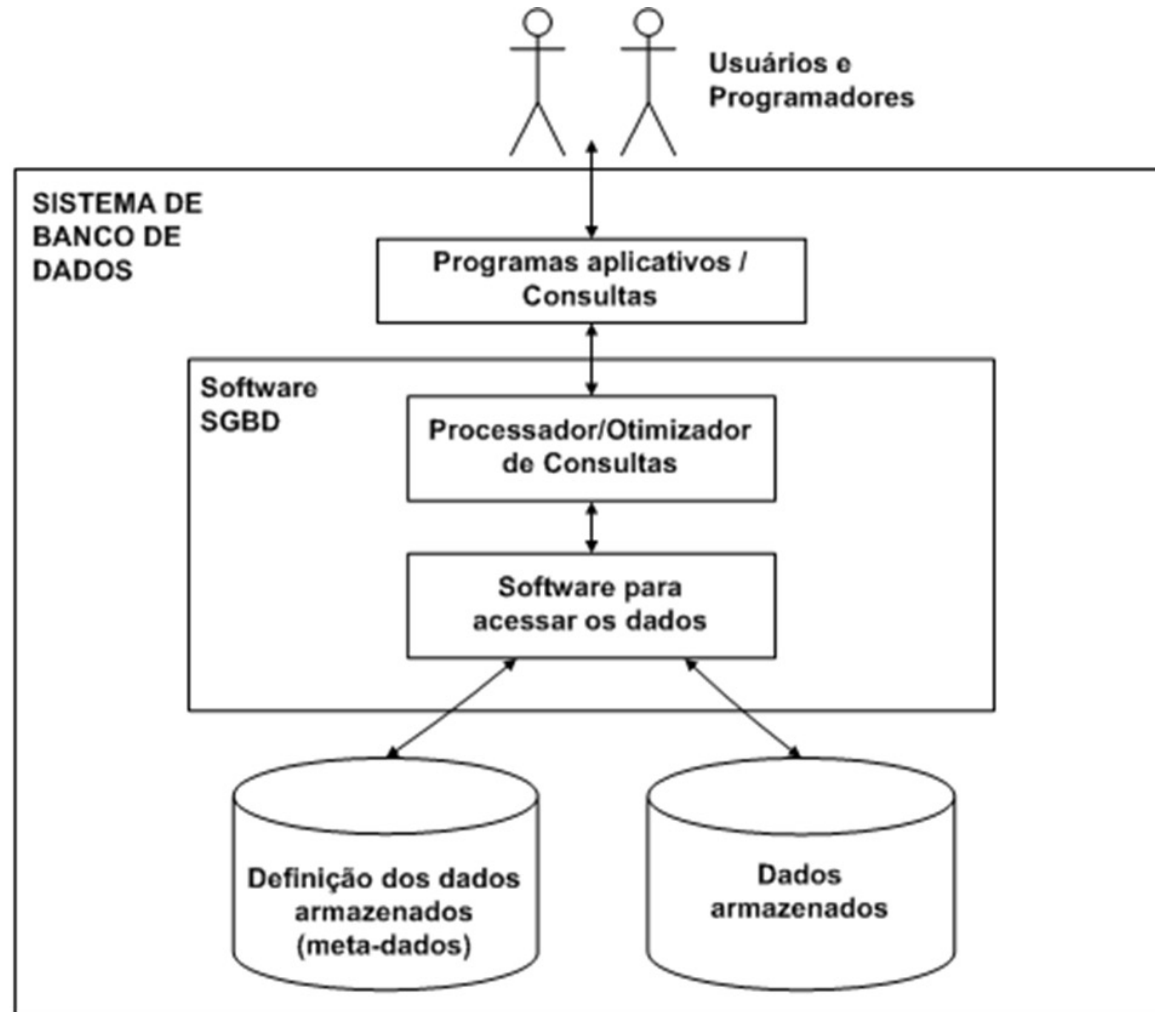
# Modelagem física

---

PROGRAMAÇÃO PARA BD

PROF. EDER SOTTO

# Sistema de Banco de Dados (SBD)



# O que é um SGBD?

Sistema Gerenciador de Banco de Dados é um conjunto de programas e ferramentas utilizadas para configurar, atualizar e manter um banco de dados.

- Recursos para administrar usuários/permisões.
- Recursos para criar/alterar tabelas e banco de dados.
- Recursos para backup e restauração de dados.
- Recursos para otimizar a performance do banco.

# Alguns SGBDs



[www.oracle.com](http://www.oracle.com)



**DB2®**

[www-01.ibm.com/software/data/db2/](http://www-01.ibm.com/software/data/db2/)



[www.firebirdsql.org/](http://www.firebirdsql.org/)



[www.sybase.com.br/](http://www.sybase.com.br/)



[www.microsoft.com/sqlserver/en/us/default.aspx](http://www.microsoft.com/sqlserver/en/us/default.aspx)



[www.postgresql.org/](http://www.postgresql.org/)



[www.mysql.com/](http://www.mysql.com/)

# CONCEITOS

---

Domínio

Atributo

Tupla

Relação

Chave

# DOMÍNIO

---

Conjunto de valores atômicos, determina os valores que podem ser utilizados em cada atributo:

- Inteiro (*integer*), Literal (*string*)
- Data (*date*), Dia e hora (*datetime*)
- ['masculino', 'feminino'] (domínio definido)

# ATRIBUTO

---

Um dado com nome e domínio definido, também chamado de campo ou coluna da tabela:

- nome: string
- idade: integer
- sexo: ['m', 'f']

# TUPLA

---

Um conjunto de atributos com valores determinados:

- Define a relação entre dados através da identificação de todos os atributos necessários para determinar um fato ou relacionamento entre fatos



# EXEMPLO

Atributos

Tuplas

Relação: Alunos

<u>matricula</u>	Nome	cpf
1000125	João Antônio	123.123.123-25
1000234	Maria Aparecida	121.121.121-22
1000322	Genésio Alves	432.123.456-25

# RELAÇÃO

---

Conjunto de tuplas composto por um **cabeçalho** e um **corpo**:

- **Cabeçalho**: Apresenta os atributos de forma não ambígua
- **Corpo**: Número variável de tuplas

# EXEMPLO

Atributos

Tuplas

Relação: Alunos

<u>matricula</u>	Nome	cpf
1000125	João Antônio	123.123.123-25
1000234	Maria Aparecida	121.121.121-22
1000322	Genésio Alves	432.123.456-25

# CHAVE

---

**Chave primária:** Utilizada para identificar unicamente uma tupla em uma relação

**Chave estrangeira:** Estabelece uma equivalência de valor com uma chave primária de outra relação. Toda chave-estrangeira referencia uma chave-primária.

# EXEMPLO

Atributos

Tuplas

Relação: Alunos

<u>matricula</u>	Nome	cpf
1000125	João Antônio	123.123.123-25
1000234	Maria Aparecida	121.121.121-22
1000322	Genésio Alves	432.123.456-25

# EXEMPLO

---

## Aluno

matricula
Nome
cpf
sigla_curso

## Curso

sigla
descricao



# EXEMPLO

ALUNO			
<u>matricula</u>	nome	cpf	sigla_curso
1000125	João Antônio	123.123.123-25	SIS
1000234	Maria Aparecida	121.121.121-22	ADM
1000322	Genésio Alves	432.123.456-25	SIS



CURSO	
<u>sigla</u>	descricao
SIS	Sistemas de Informação
ADM	Administração

# RESTRIÇÕES DE INTEGRIDADE

---

Oferece consistência de dados através de regras

Domínio

Entidade

Referencial



# RESTRIÇÕES DE INTEGRIDADE

## **Integridade de Domínio**

---

O SGBD garante que as informações armazenadas em cada atributo estejam de acordo com o domínio definido.

## **Entidade**

O SGBD garante a integridade de suas entidades (tabelas) através das chaves-primárias e suas respectivas validações.

## **Referencial**

É garantida pelo uso das chaves-estrangeiras. Onde toda chave-estrangeira deve corresponder à sua respectiva chave-primária.

# EXEMPLO

INSERIR EM ALUNOS:

(1000436 , Luiz Augusto , 310.310.410-56 , CCO)

ALUNOS			
<u>Matrícula</u>	Nome	CPF	Curso
1000125	João Antônio	123.123.123-25	SIS
1000234	Maria Aparecida	121.121.121-22	ADM
1000322	Genésio Alves	432.123.456-25	SIS

CURSOS	
<u>Sigla</u>	Descrição
SIS	Sistemas de Informação
ADM	Administração

# Modelo físico

---

Chamamos de modelagem física, modelo físico ou esquema físico, qualquer esquema de banco de dados que detalhe a estrutura de suas tabelas, de acordo com o SGBD em que será utilizado.

Um modelo físico pode ser um dicionário de dados, um diagrama físico ou um conjunto de comandos SQL DDL (*data definition language*)

# Linguagem SQL

---

A linguagem SQL ou *Structured Query Language* é uma linguagem declarativa e a linguagem padrão para comunicação com os SGBD's relacionais.

A linguagem SQL pode ser dividida em três grupos:

- DDL – *Data definition language* – Linguagem de definição de dados
- DCL – *Data control language* – Linguagem de controle de dados
- DML – *Data manipulation language* – Linguagem de manipulação de dados

# Linguagem SQL - DDL

---

A DDL é a linguagem que utilizamos para criação dos nossos bancos de dados, e também para realizar qualquer modificação em sua estrutura.

A DDL é composta principalmente pelos comandos: **Create, alter e drop**

# Linguagem SQL - DCL

---

A DCL é a linguagem que utilizamos para criar usuários de banco de dados e definir ou retirar permissões de qualquer usuário no banco de dados.

A DCL é composta principalmente pelos comandos: **Grant e revoke.**

# Linguagem SQL - DML

---

A DML é a linguagem que utilizamos para manipular dados no banco de dados. Pode ser considerada a principal parte da linguagem SQL.

A manipulação de dados é composta pelas operações: Inserção, alteração, exclusão e consulta.

É importante ressaltar que a DML não altera nenhuma estrutura de banco de dados e suas tabelas.

A DML é composta pelos comandos: **Insert, update, delete e select.**

# Principais domínios do SQL Server

---

## Numéricos inteiros

- bit
- tinyint
- int
- bigint

## Numéricos racionais

- float
- numeric\*
- real
- money

## Datas (AAAA-MM-DD)

- date (somente a data)
- datetime (data e hora)  
ex: 2020-05-12 21:48:00
- datetime2 (data e hora) mais precisão para hora. ex: 2020-05-12 21:48:00.158

## Strings

- char (campo de tamanho fixo)
- varchar (campo de tamanho variável)
- nvarchar (campo de tam. variável UNICODE)



# Principais tipos de Restrição

---

PK – Primary key (chave-primária)

FK – Foreign Key (chave-estrangeira)

UQ – Unique (único / chave-candidata)

CK – Check (restrições customizáveis). Ex: Validação de CPF.

DEFAULT – Valor padrão do atributo (preenchimento automático).

# Exemplo de Restrição do tipo PK (primary key)

---

```
CREATE TABLE FUNCIONARIO(  
    CODIGO INT NOT NULL IDENTITY(1,1),  
    NOME VARCHAR(50) NOT NULL,  
    EMAIL VARCHAR(50) NOT NULL,  
    CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(CODIGO)  
);
```

# Exemplo de Restrição do tipo FK (foreign key)

---

```
CREATE TABLE DEPENDENTE(  
    CODIGO INT NOT NULL IDENTITY(1,1),  
    NOME VARCHAR(50) NOT NULL,  
    COD_FUNCIONARIO INT NOT NULL,  
    CONSTRAINT PK_DEPENDENTE PRIMARY KEY(CODIGO),  
    CONSTRAINT FK_DEPENDENTE_FUNCIONARIO FOREIGN  
    KEY(COD_FUNCIONARIO) REFERENCES FUNCIONARIO(CODIGO)  
);
```

# Exemplo de Restrição do tipo UQ (unique) (chave-candidata)

---

```
CREATE TABLE FUNCIONARIO(  
    CODIGO INT NOT NULL IDENTITY(1,1),  
    NOME VARCHAR(50) NOT NULL,  
    EMAIL VARCHAR(50) NOT NULL,  
    CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(CODIGO),  
    CONSTRAINT UQ_FUNCIONARIO_EMAIL UNIQUE(EMAIL)  
);
```

# Exemplo de Restrição do tipo CK (CHECK)

---

```
CREATE TABLE FUNCIONARIO(  
    CODIGO INT NOT NULL IDENTITY(1,1),  
    NOME VARCHAR(50) NOT NULL,  
    SEXO CHAR(1) NOT NULL,  
    CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(CODIGO),  
    CONSTRAINT CK_FUNCIONARIO_SEXO CHECK(SEXO IN('F','M'))  
);
```

# Exemplo de Restrição do tipo DEFAULT

---

```
CREATE TABLE FUNCIONARIO(  
    CODIGO INT NOT NULL IDENTITY(1,1),  
    NOME VARCHAR(50) NOT NULL,  
    DATA_HORA_CADASTRO DATETIME NOT NULL DEFAULT GETDATE(),  
    ATIVO BIT NOT NULL DEFAULT 1,  
    CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(CODIGO),  
);
```

# Nulabilidade

---

Anulável – Pode ser nulo – Opcional – “NULL”

Não-anulável – Não pode ser nulo – Obrigatório – “NOT NULL”

# Auto incremento

---

O auto incremento permite delegar ao SGBD a responsabilidade pelo preenchimento automático da chave-primária.

Importante: só podemos definir o auto incremento para atributos que forem chave primária.

Observações: Para que isso seja possível, a chave primária deve ser do tipo numérica (int, smallint, bigint) e deve ser não-composta.

No SQL Server, o auto incremento é declarado através do predicado `IDENTITY`.



# Exemplo de uso do auto incremento no SQL Server (identity)

---

```
CREATE TABLE FUNCIONARIO(  
    CODIGO INT NOT NULL IDENTITY(1,1),  
    NOME VARCHAR(50) NOT NULL,  
    CONSTRAINT PK_FUNCIONARIO PRIMARY KEY(CODIGO)  
);
```

# Exemplo 1: BD Livraria

AUTORIA			
	Nome da Coluna	Tipo Condensado	Permite Valor Nulo
🔑	COD_LIVRO	int	Não
🔑	MAT_AUTOR	int	Não

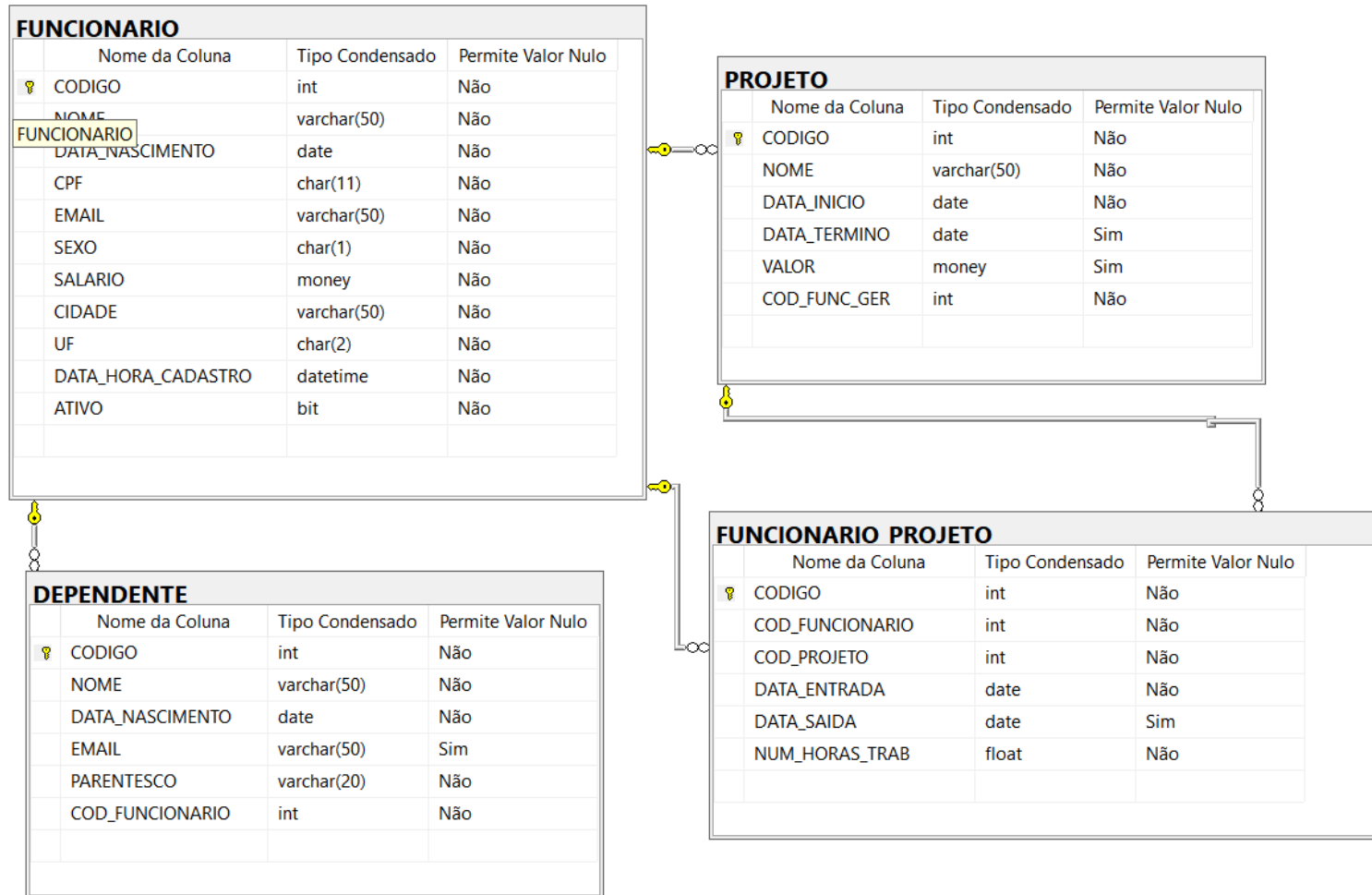
AUTOR			
	Nome da Coluna	Tipo Condensado	Permite Valor Nulo
🔑	MATRICULA	int	Não
	NOME	varchar(50)	Não
	DATA_NASCIM...	date	Sim
	NACIONALIDA...	varchar(50)	Sim

LIVRO			
	Nome da Coluna	Tipo Condensado	Permite Valor Nulo
🔑	CODIGO	int	Não
	TITULO	varchar(50)	Não
	DATA_LANCA...	date	Sim
	VALOR	money	Sim
	DATA_HORA_C...	datetime	Não
	ATIVO	bit	Não
	COD_EDITORA	int	Não
	SIGLA_ASSUNTO	char(3)	Não

ASSUNTO			
	Nome da Coluna	Tipo Condensado	Permite Valor Nulo
🔑	SIGLA	char(3)	Não
	DESCRICAO	varchar(50)	Não

EDITORIA			
	Nome da Coluna	Tipo Condensado	Permite Valor Nulo
🔑	CODIGO	int	Não
	NOME	varchar(50)	Não

# Exercício: BD Construtora



# Restrições

---

Atributo DATA\_HORA\_CADASTRO deve ter default e ser preenchido com a data/hora atuais (DEFAULT).

Atributo ATIVO deve ter default 1

Atributos EMAIL e CPF devem ter UNIQUE

Atributo SEXO deve permitir apenas M ou F (CHECK)

Atributo DATA\_HORA\_SAIDA não pode ter valor anterior a DATA\_HORA\_ENTRADA (CHECK)

Atributo DATA\_TERMINO não pode ter valor anterior a DATA\_INICIO (CHECK)