

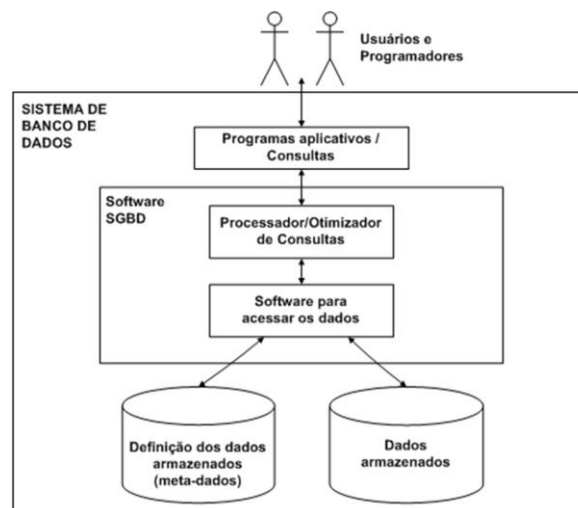
Banco de Dados

Modelagem física

PROGRAMAÇÃO PARA BD
PROF. EDER SOTTO

1

Sistema de Banco de Dados (SBD)



2

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.

3

Alguns SGBDs



www.oracle.com



www-01.ibm.com/software/data/db2/



www.firebirdsql.org/



www.sybase.com.br/



www.microsoft.com/sqlserver/en/us/default.aspx



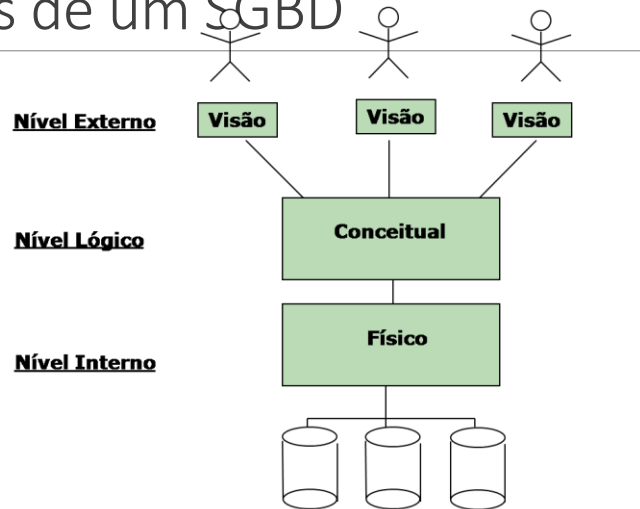
www.postgresql.org/



www.mysql.com/

4

Níveis de um SGBD



5

Níveis de um SGBD

Nível Externo

- É o nível em que o usuário possui acesso, ou seja, o usuário enxerga apenas os objetos aos quais possui permissão de acessar.

Nível Lógico

- Também chamado de esquema global, é um tipo de nível externo, mas com permissão de acesso a todos os objetos do BD.

Nível Interno

- É o nível onde somente o SGBD tem acesso. É neste nível onde o SGBD faz a alocação dos dados e meta-dados em disco.

6

CONCEITOS

Domínio

Atributo

Tupla

Relação

Chave



7

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)



8

ATRIBUTO

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

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



9

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



10

EXEMPLO

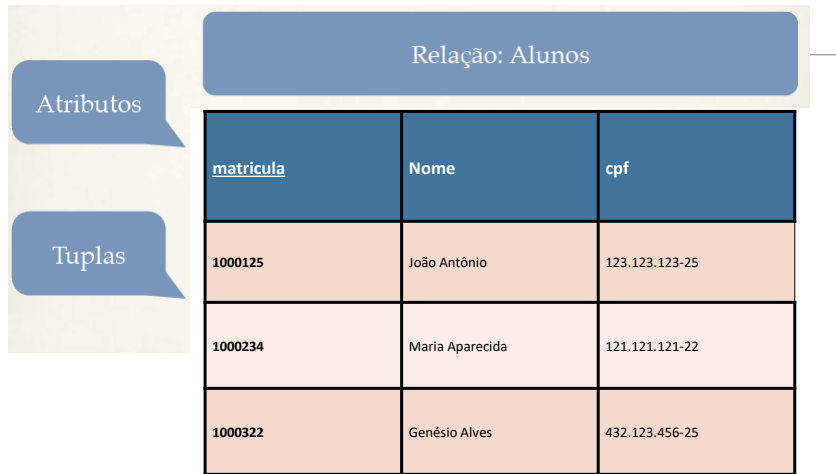


Diagram illustrating a table structure for a relation named "Relação: Alunos". The table has three columns: matricula, Nome, and cpf. The first column is labeled "Atributos" and the second and third columns are labeled "Tuplas".

<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

11

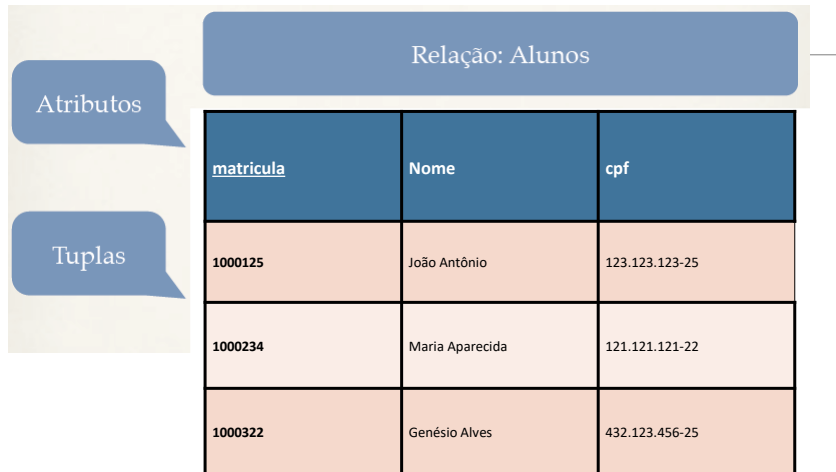
RELAÇÃO

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

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

12

EXEMPLO



The diagram illustrates a database relation named "Relação: Alunos". To the left of the table, two blue speech bubble-like boxes are labeled "Atributos" (Attributes) and "Tuplas" (Tuples). The table itself has three columns: "matricula" (underlined, indicating a primary key), "Nome", and "cpf". It contains three rows of data (tuples).

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

13

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.

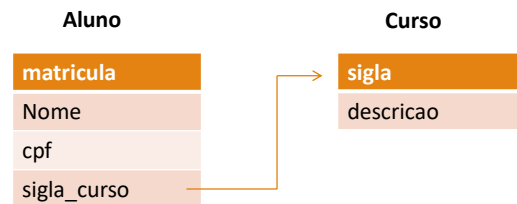
14

EXEMPLO

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

15

EXEMPLO



16

EXEMPLO



17

RESTRIÇÕES DE INTEGRIDADE

Oferece consistência de dados através de regras

Domínio

Entidade

Referencial

18

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.

19

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

20

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*)


21

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


22

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**



23

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.**



24

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.**

25

Principais tipos de dados 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)

26

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).


27

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)  
);
```


28

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)
);
```

29

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)
);
```

30

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'))
);
```

31

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),
);
```

32

Nulabilidade

Anulável – Pode ser nulo – Opcional – “NULL”

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



33

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.



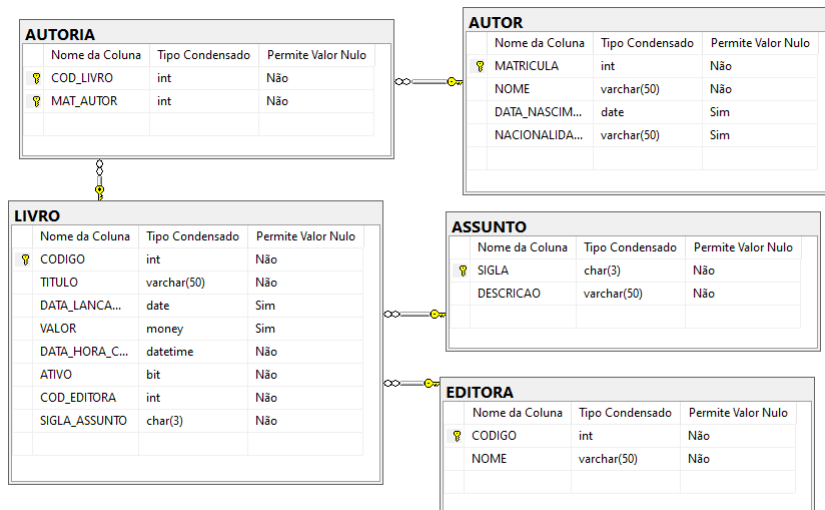
34

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)
);
```

35

Exercício 1: Livraria



36

Exercício 2: Construtora

