

Curso de Jogos Digitais

Disciplina de Tecnologias Web

Aula 14 – Banco de Dados

Professor: André Flores dos Santos



Banco de dados

- Um **Sistema de Gerenciamento de Banco de Dados** (SGBD) consiste em uma **coleção de dados inter-relacionados e em um conjunto de programas para acessá-los**.
- Um conjunto de dados, normalmente referenciado como **banco de dados**, contém informações sobre um empreendimento particular.
- O principal objetivo do SGBD é **prover um ambiente** que seja **conveniente e eficiente** para recuperar e armazenar as informações de bancos de dados.

Banco de dados

Os SGBD são projetados para gerenciar **grandes grupos de informações**.

Para isso, são necessários:

Definição de **estruturas para armazenamento** de informações

Mecanismos para **manipular as informações**

O SGBD também deve fornecer mecanismos de **segurança e controles de acesso**

Banco de dados

Porquê banco de dados e não arquivos?

O SGBD tem como objetivo prover um **ambiente que seja adequado e eficiente para recuperar, armazenar e manter as informações pertencentes a um banco de dados**

Necessidade de haver a separação entre os problemas específicos de aplicações e problemas de armazenamento de dados, ou seja, **independência entre dados e programas**

Banco de dados

Portanto, *SGBDs* devem:

- Definir as **estruturas de armazenamento** dos dados;
- Definir os **mecanismos para manipulação** dos dados;
- Garantir a **segurança dos dados** contra ações externas;
- Garantir a **integridade dos dados** armazenados;
- Controlar o **acesso concorrente** aos dados;
- Permitir a **independência dos dados** armazenados;
- Permitir e manter o relacionamento entre os dados;
- Controlar a **redundância** dos dados;
- Garantir um bom **desempenho** mesmo com uma grande massa de dados.

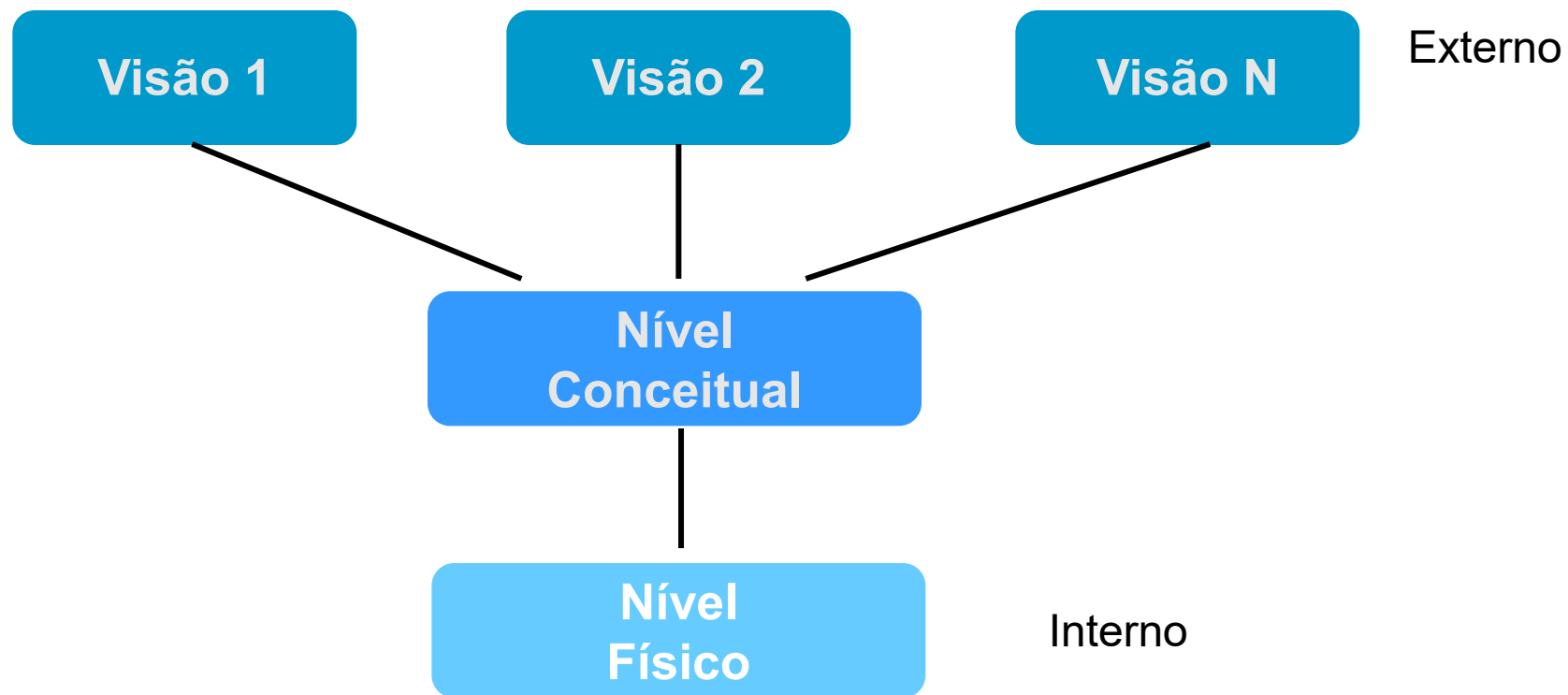
Abstração de dados

- O SGBD é composto por uma **coleção de arquivos** inter-relacionados e de um conjunto de programas específicos que permitem aos usuários **acessar** e **modificar** esses arquivos.
- O grande objetivo de um sistema de banco de dados é prover aos usuários uma **visão abstrata dos dados**.
- O sistema **omite certos detalhes** de como os dados são armazenados e mantidos, uma vez que muitos dos seus usuários não são especialistas.

Níveis de abstração

- A complexidade está escondida deles por meio de diversos **níveis de abstração**:
 - **Nível Interno ou Físico**: É o nível mais baixo de abstração e descreve como os dados estão realmente armazenados.
 - Neste nível, as estruturas complexas do SGBD são descritas;
 - **Nível Conceitual**: Descreve quais os dados que estão armazenados de fato no banco de dados e as relações existentes entre eles.
 - O BD é descrito nesse nível em um pequeno número de estruturas relativamente simples.
 - **Nível Externo ou de Visões**: É o mais alto nível de abstração e descreve o banco de dados em partes que são de interesse de cada usuário/aplicação.
 - Busca simplificar a interação dos usuários com o sistema, gerando assim diversas visões do BD.

Níveis de abstração



Modelos de dados

- Fundamentam as estruturas nas quais um banco de dados pode ser construído.
- São utilizados para **especificar a descrição, os relacionamentos, a semântica dos dados e suas restrições de consistência, e o conjunto de operações possíveis sobre os dados.**
- Como exemplos de modelos de dados têm-se:
 - Entidade-Relacionamento
 - Modelo Relacional

Banco de dados

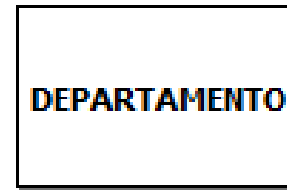
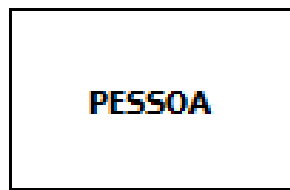
Diagrama Entidade-Relacionamento

Modelo Entidade-Relacionamento

- O modelo Entidade-relacionamento é baseado na **percepção do mundo real** que consiste em um **conjunto de objetos** chamados **entidades** e nos **relacionamentos** entre esses objetos
- Ele foi desenvolvido para **facilitar o projeto de banco de dados**.
- Este esquema representa a **estrutura lógica** geral do banco de dados.

Entidade

- Conjunto de objetos da realidade. São modelos sobre os quais deseja-se manter informações no Banco de Dados.
- Uma entidade pode ser:
 - Concreta: pessoa, livro
 - Abstrata: cargo, departamento
- Tem como objetivo modelar de forma abstrata um Banco de Dados.
- É representado por um retângulo contendo o nome da entidade:



Entidade

- Um **conjunto-entidade** representa um conjunto de entidades do mesmo tipo
 - O conjunto de todos os cliente de um banco
 - O conjunto de todas as contas de clientes de um banco
- Uma entidade é representada por um conjunto de **atributos**
 - Exemplos:
 - atributos de uma entidade **cliente**:
 - *nome, CPF, rua, cidade*
 - atributos de uma entidade **conta**:
 - *numero e saldo*
- Para cada atributo existe um **conjunto de valores permitidos**, chamado **domínio** daquele atributo:
 - O domínio do atributo *nome* pode ser o conjunto de cadeias de texto de um certo tamanho;
 - O domínio do atributo *numero* é o conjunto de números inteiros positivos

Entidade

- Um banco de dados inclui uma **coleção de conjuntos-entidade**, cada qual contendo qualquer número de entidades do mesmo tipo

João	111.222.333-44	Andradas	Santa Maria
Maria	111.222.333-55	Medianeira	Santa Maria
José	111.222.333-66	Dores	Santa Maria
Pedro	111.222.333-77	Tuiuti	Santa Maria
Ana	111.222.333-88	Bozano	Santa Maria

Conjunto-entidade *cliente*

003	1000
123	500
332	300
541	1150
599	50
776	8000
987	100

Conjunto-entidade
conta

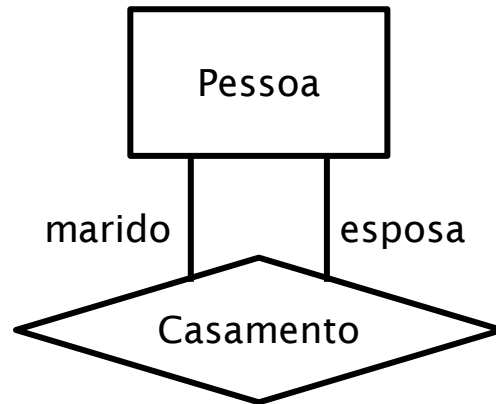
Relacionamento

- É o conjunto de associações entre ocorrências de entidades.
- É representado por um losango ligado por linhas aos retângulos representativos das entidades que participam do relacionamento:



Auto-Relacionamento

- Relacionamento entre ocorrências de uma mesma entidade.
- Exemplo:

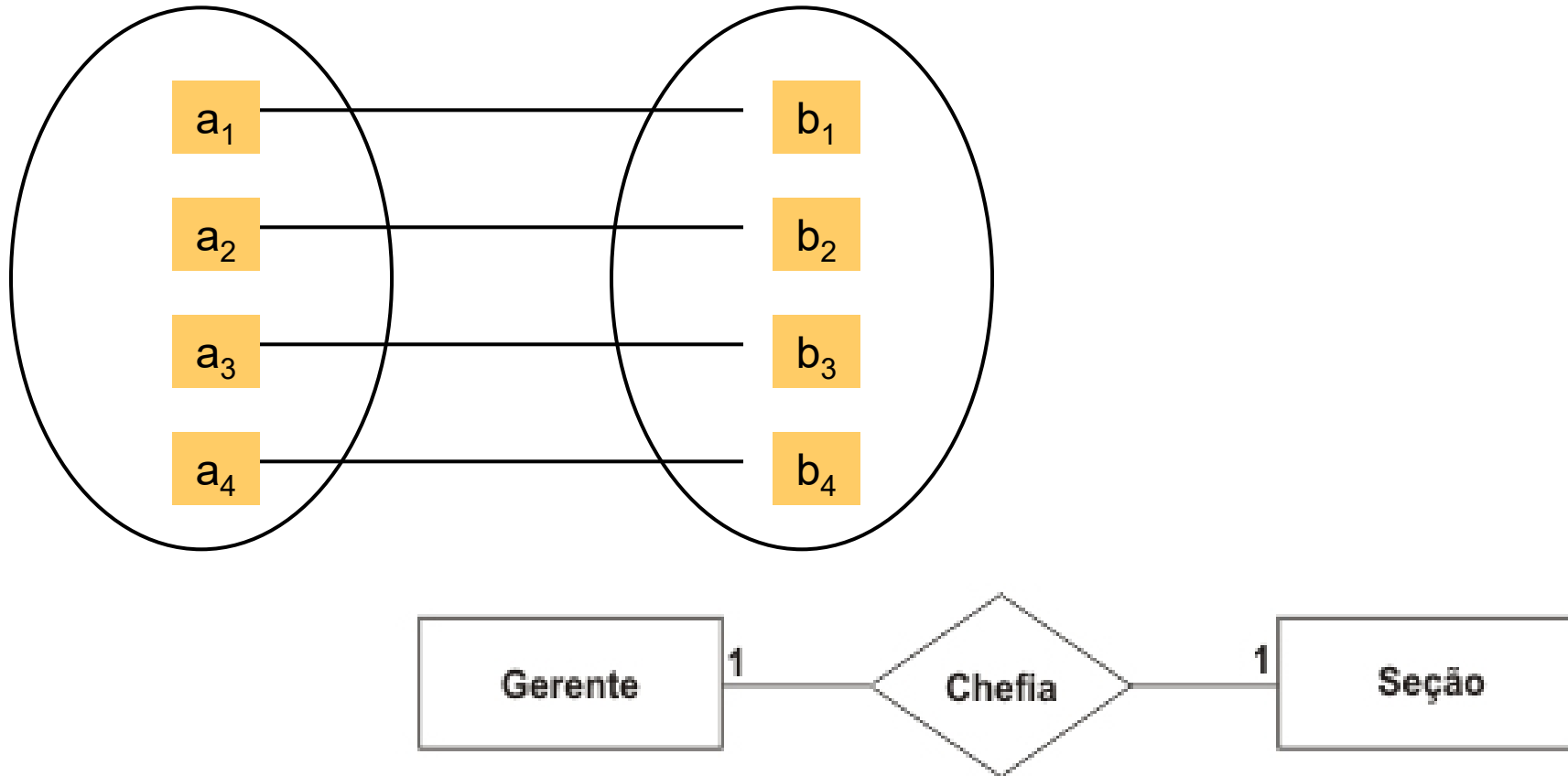


Cardinalidade de um relacionamento

- Número (**mínimo**, **máximo**) de ocorrências de entidade associadas a uma ocorrência da entidade em questão por meio do relacionamento.
- Podem ser:
 - Um-para-um;
 - Muitos-para-um;
 - Muitos-para-muitos;

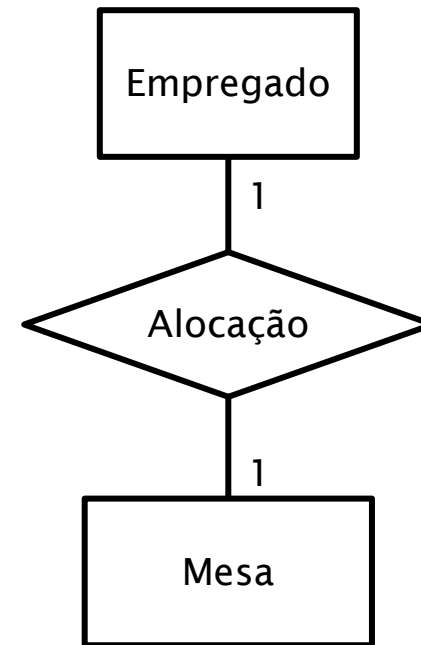
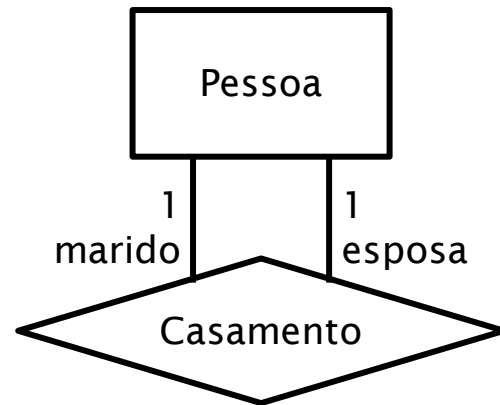
Um-para-um

- Uma entidade de A está associada no máximo a uma entidade de B, e vice-versa (1:1)



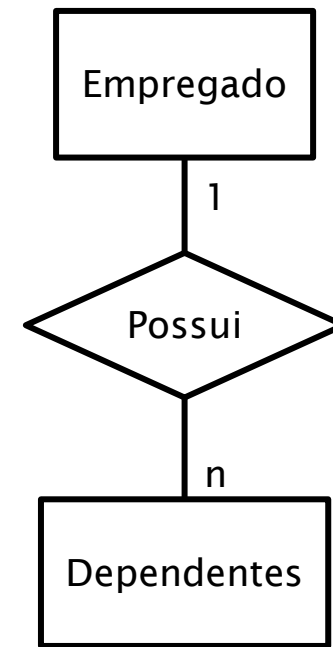
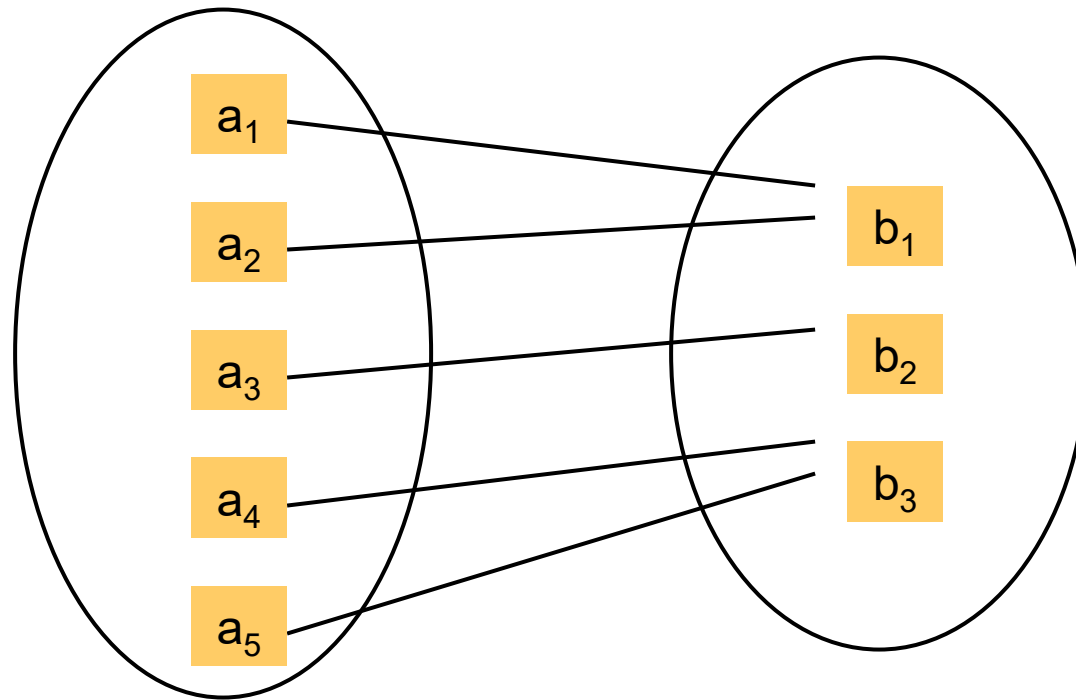
Um-para-um

Exemplos



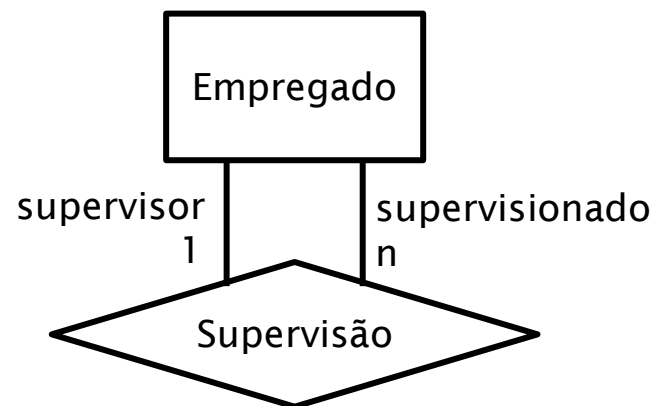
Muitos-para-um

- Uma entidade de A está associada no máximo a uma entidade de B. Uma entidade de B, entretanto, pode estar associada a qualquer número de entidades de A (**N:1**)



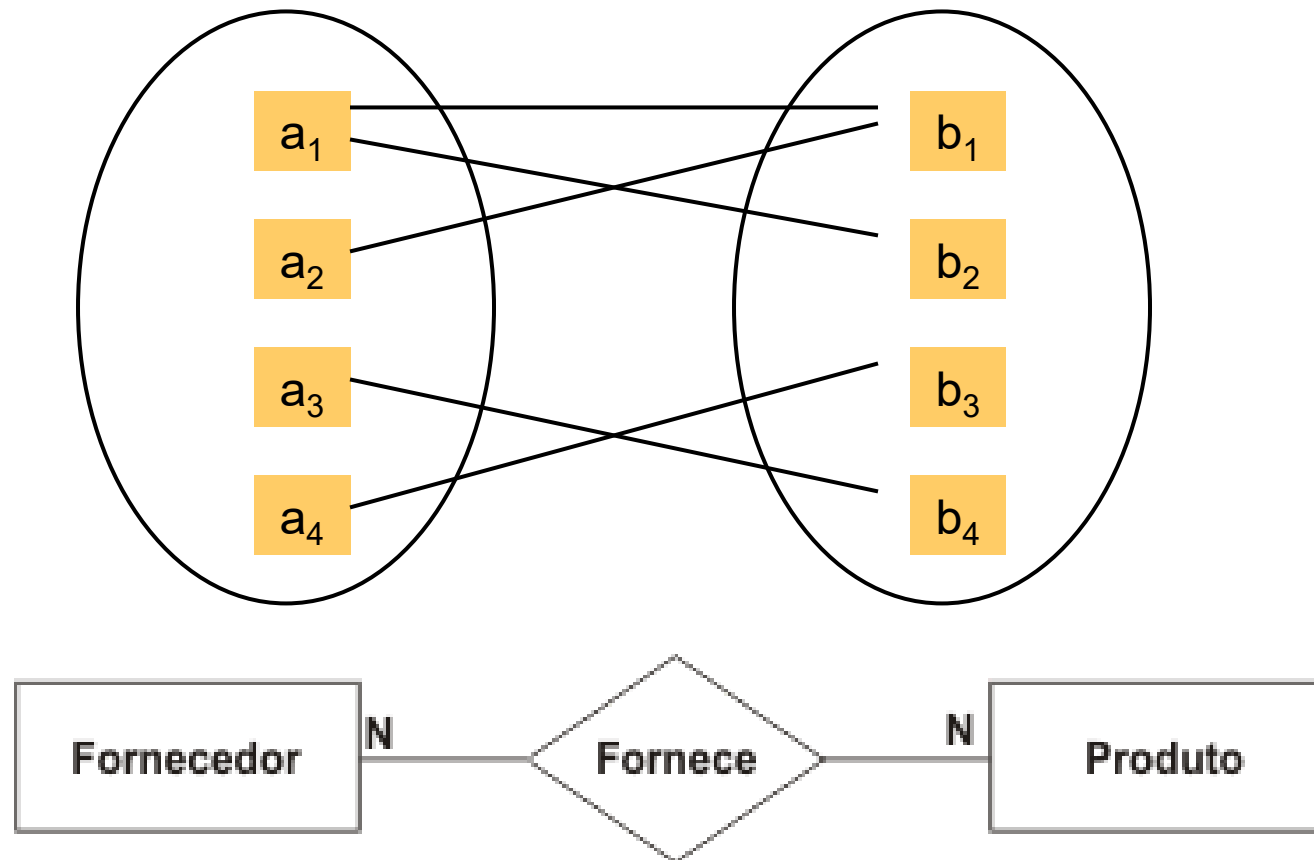
Muitos-para-um

- Exemplos



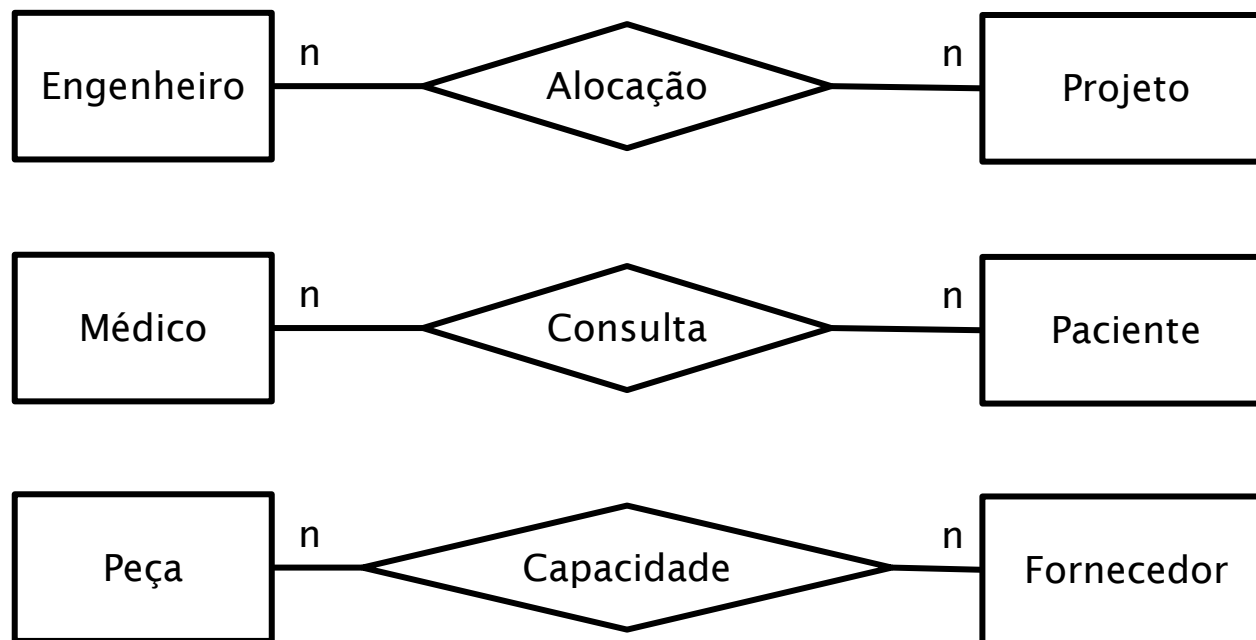
Muitos-para-muitos

- Uma entidade de A está associada a qualquer número de entidades de B, e vice-versa (**N:N**)



Muitos-para-muitos

- Exemplos

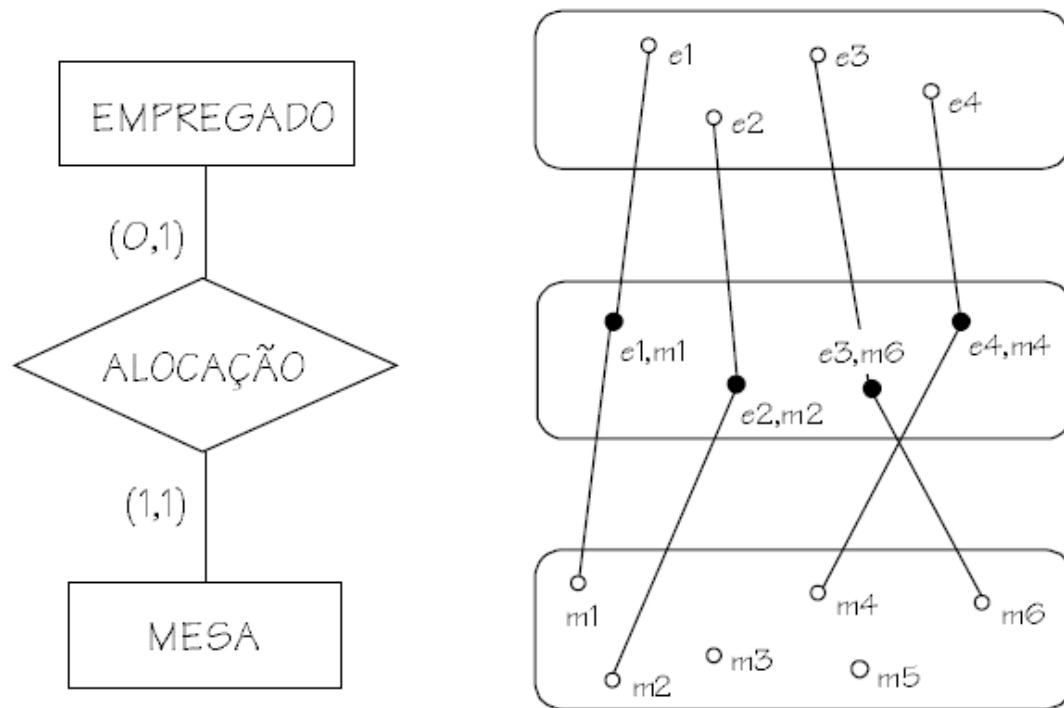


Cardinalidade mínima

- Número mínimo de ocorrências de entidade que são associadas a uma ocorrência de uma entidade através de um relacionamento
- Cardinalidade mínima 0: Associação Opcional
- Cardinalidade mínima 1: Associação Obrigatória
 - O relacionamento deve obrigatoriamente associar uma ocorrência de entidade a cada ocorrência da entidade em questão

Cardinalidade mínima

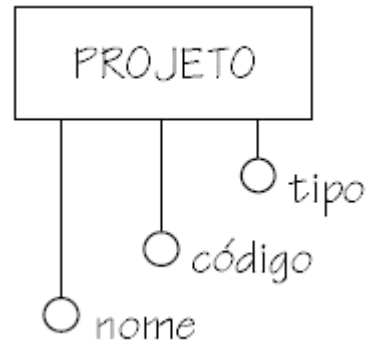
□ Exemplo:



Cada empregado deve ter a ele alocada obrigatoriamente uma mesa (cardinalidade mínima 1) e que uma mesa pode existir sem que a ela esteja alocado um empregado (cardinalidade mínima 0).

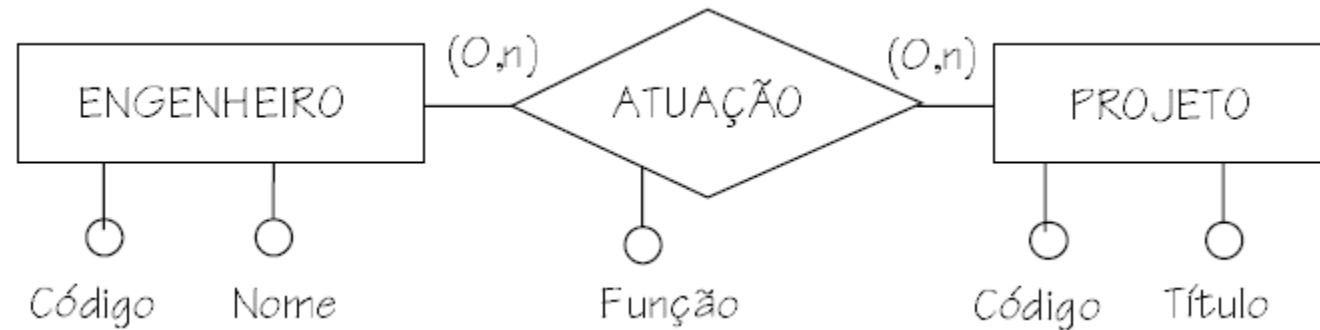
Atributos

- Dado que é associado a cada ocorrência de uma entidade ou de um relacionamento.



Atributos de Relacionamentos

- Assim como entidades possuem atributos, também relacionamentos podem possuir atributos.

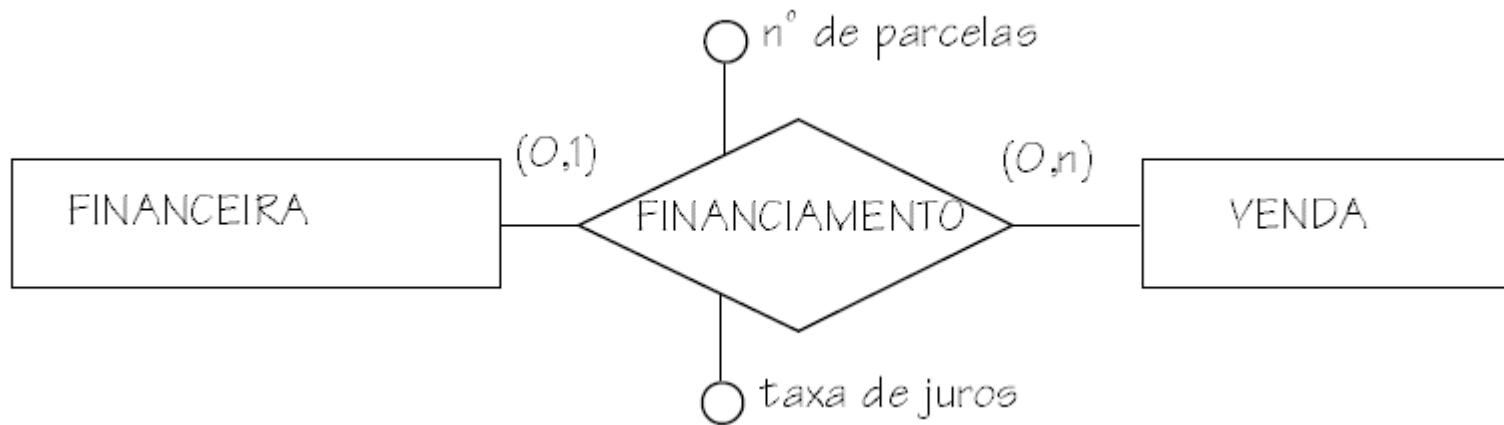


ATUAÇÃO possui um atributo, a função que um engenheiro exerce dentro de um projeto.

- Não pode ser considerada atributo de ENGENHEIRO, já que um engenheiro pode atuar em diversos projetos exercendo diferentes funções.
- Não é atributo de PROJETO, já que, em um projeto, podem atuar diversos engenheiros com funções diferentes.

Atributos de Relacionamentos

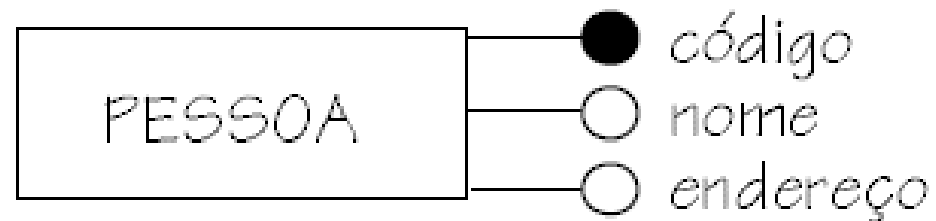
- Exemplo:



Algumas vendas são à vista, outras à prazo.
Vendas à prazo são relacionadas a uma financeira, através do relacionamento **FINANCIAMENTO**.
Os atributos nº de parcelas e taxa de juros são atributos do relacionamento.

Identificadores de entidades

- Cada entidade deve possuir um identificador.
- Um identificador é um conjunto de um ou mais atributos cujos valores servem para distinguir uma ocorrência da entidade das demais ocorrências da mesma entidade.



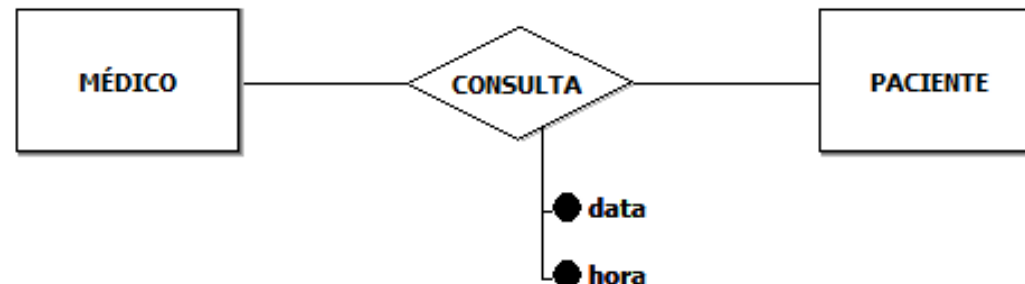
Identificadores de entidades

- Exemplo:
 - Considera-se um almoxarifado de uma empresa de ferragens organizado como segue.
 - Os produtos ficam armazenados em prateleiras.
 - Estas prateleiras encontram-se em armários organizados em corredores.
 - Os corredores são numerados sequencialmente a partir de um e as prateleiras são numeradas sequencialmente a partir de um dentro de um corredor.
 - Assim, para identificar uma prateleira é necessário conhecer seu número e o número do corredor em que se encontra.
 - Para cada prateleira deseja-se saber sua capacidade em metros cúbicos.



Identificando Relacionamentos

- Para um determinado médico e um determinado paciente podem haver diversas consultas.
- Há necessidade de algo que distinga uma consulta entre um médico e seu paciente das demais consultas entre este médico e este paciente.



Conceitos importantes

- **Esquema:** É chamado de esquema todo o **projeto lógico e físico** de um banco de dados
- **Linguagem de Definição de Dados (DDL):** É a linguagem utilizada para descrever o esquema do banco de dados
 - O resultado da compilação de comandos DDL resulta em um conjunto de tabelas que são armazenadas em um arquivo especial chamado **dicionário de dados**.

```
--Exemplo

CREATE TABLE Clientes (
    id          INT PRIMARY KEY,
    nome        VARCHAR(100) NOT NULL,
    cpf         CHAR(11) UNIQUE NOT NULL,
    telefone    VARCHAR(20)
);
```


Conceitos importantes

- **Dicionário de dados:** É um arquivo que contém *metadados*
 - Os *metadados* são os dados que descrevem os dados
 - O dicionário de dados é consultado antes que os dados reais sejam lidos ou modificados no banco de dados
- **Linguagem de Manipulação de Dados (DML):** É a linguagem que permite aos usuários ter acesso e manipular os dados do banco de dados:
 - Recuperação da informação armazenada
 - Inserção de novas informações
 - Eliminação de informações
 - Modificação de dados

--Exemplo

-- Inserção

```
INSERT INTO Clientes (id, nome, cpf, telefone, email)
VALUES (1, 'Ana Silva', '12345678901', '55-99999-0000', 'ana@exemplo.com');
```

-- Consulta

```
SELECT id, nome, cpf, telefone, email
FROM Clientes
WHERE nome LIKE 'Ana%';
```

Exercício 01:

- Uma transportadora aérea pretende implementar uma base de dados com a seguinte informação:
 - A transportadora tem vários aviões.
 - Cada avião tem, além da matrícula, um nome, o modelo do avião, o número de lugares, e a indicação da sua autonomia.
 - Na transportadora trabalham vários pilotos.
 - Sobre cada piloto pretende-se guardar o nome e número de licença, assim como quais os modelos de aviões que podem pilotar.
 - Pretende-se ainda, guardar a informação relativa ao nome, data de nascimento de cada um dos descendentes (caso existam) dos pilotos.
 - Cada avião faz vários voos.
 - Cada voo deve ter, pelo menos, a indicação da data e hora em que acontecerá dos locais de partida e de destino.
 - Cada voo de um dado avião é pilotado por um piloto.

Exercício 01:

Criar o modelo conceitual do exercício 01 no site BrModelo (<https://www.brmodeloweb.com/lang/pt-br/index.html>).

Colocar o print dentro de um arquivo PDF com identificação e entregar com as demais atividades de hoje.

OBS: Enviar num arquivo compactado (zip) todas as atividades da aula de hoje.

Parte II da aula – Banco de Dados

Exemplo de Banco de Dados Relacional - Tabelas

“Consideramos um contexto simples em que muitos estão familiarizados, um banco de dados UNIVERSIDADE para manter informações sobre alunos, disciplinas, professores e notas em um ambiente universitário”

PROFESSOR		
CPF	Nome	Salario
10120214450	Lucas Menezes	5420.00
11040540330	Bruno Santos	4200.00
10560930210	Marcos Lima	7150.00
10010220470	Elton Cardoso	6500.00
10190267390	Rafael Silva	2500.00
10432029020	Fernando Vaz	3880.00

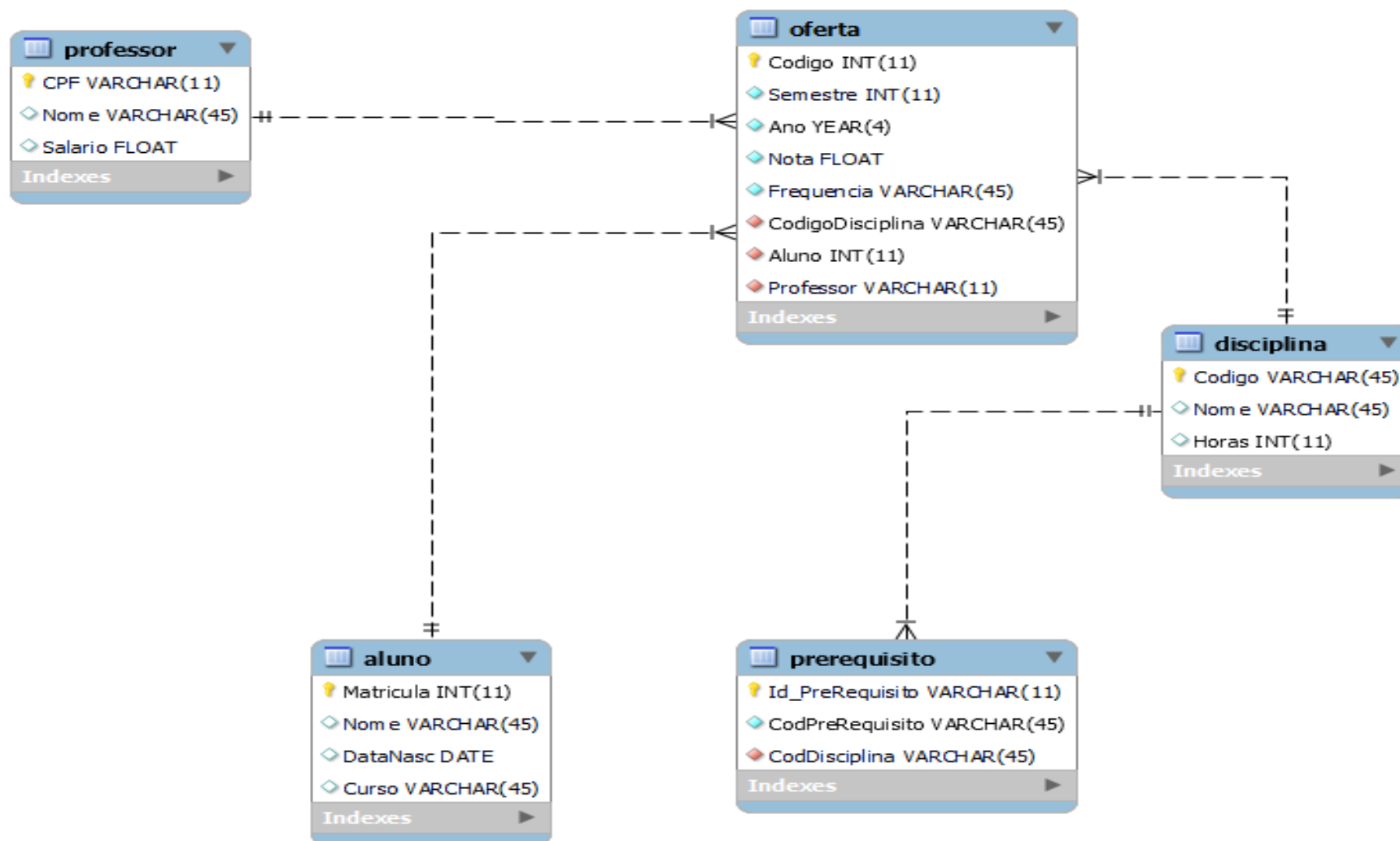
ALUNO			
Matricula	Nome	DataNascimento	Curso
1918023	Camila Braga	12/03/1999	COM
1918142	João Lage	20/11/2001	PRO
1828201	Pedro Santos	02/09/2000	COM

DISCIPLINA		
Codigo	Nome	Horas
CEA160	Cálculo Diferencial e Integral I	60
CSI030	Programação de Computadores I	60
CSI440	Banco de Dados I	60
CSI443	Matemática Discreta	60
CSI488	Algoritmos e Estruturas de Dados I	60

PREREQUISITO	
CodigoDisciplina	CodigoPrerequisito
CSI440	CSI443
CSI440	CSI488
CSI488	CSI030

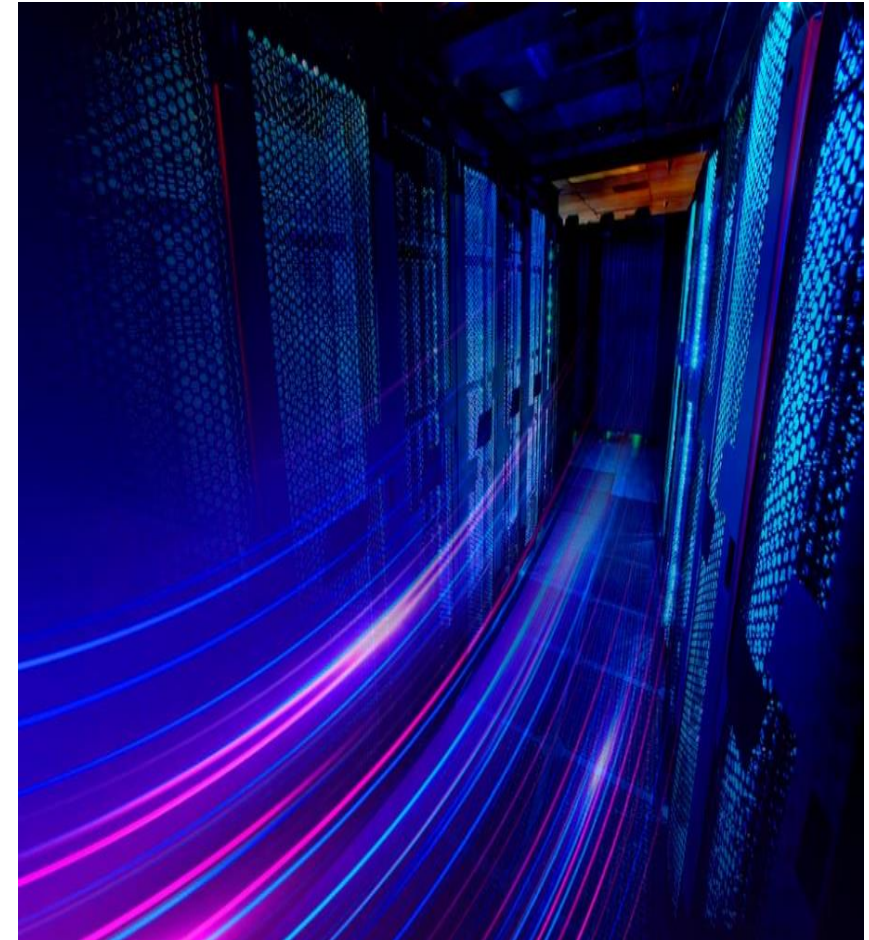
OFERTA							
Codigo	CodigoDisciplina	Semestre	Ano	Nota	Frequencia	Aluno	Professor
1	CSI030	1	2019	8.6	90%	1918023	10120214450
2	CSI030	1	2019	7.4	95%	1918142	10120214450
3	CSI030	1	2019	2.5	50%	1828201	10120214450
4	CSI440	1	2019	4.7	75%	1918142	11040540330
5	CSI160	1	2019	9.5	100%	1828201	10560930210
6	CSI443	1	2019	7.7	82%	1918023	10010220470
7	CSI030	2	2019	6.0	90%	1828201	10432029020
8	CSI440	2	2019	6.7	85%	1918142	10190267390
9	CSI160	2	2019	8.2	92%	1918023	10560930210
10	CSI443	2	2019	9.0	100%	1828201	10010220470

Exemplo de Banco de Dados Relacional – Modelo ER



Exemplo

Essa porção do banco de dados é organizada em cinco tabelas, onde cada qual armazena **registros** do mesmo tipo. Para *definir* esse banco de dados é necessário definir a estrutura dos registros em cada tabela especificando diferentes tipos de **elementos de dados** para serem armazenados em cada registro. É necessário ainda definir um **tipo de dados** para cada coluna da tabela (ou atributo)



Exemplo

Por exemplo, o **nome** na tabela **ALUNO** é uma cadeia de caracteres (varchar), já a **nota** na tabela **OFERTA** é um número decimal (float).



Exemplo

PROFESSOR		
CPF	Nome	Salario
10120214450	Lucas Menezes	5420.00
11040540330	Bruno Santos	4200.00
10560930210	Marcos Lima	7150.00
10010220470	Elton Cardoso	6500.00
10190267390	Rafael Silva	2500.00
10432029020	Fernando Vaz	3880.00

ALUNO			
Matricula	Nome	DataNascimento	Curso
1918023	Camila Braga	12/03/1999	COM
1918142	João Lage	20/11/2001	PRO
1828201	Pedro Santos	02/09/2000	COM

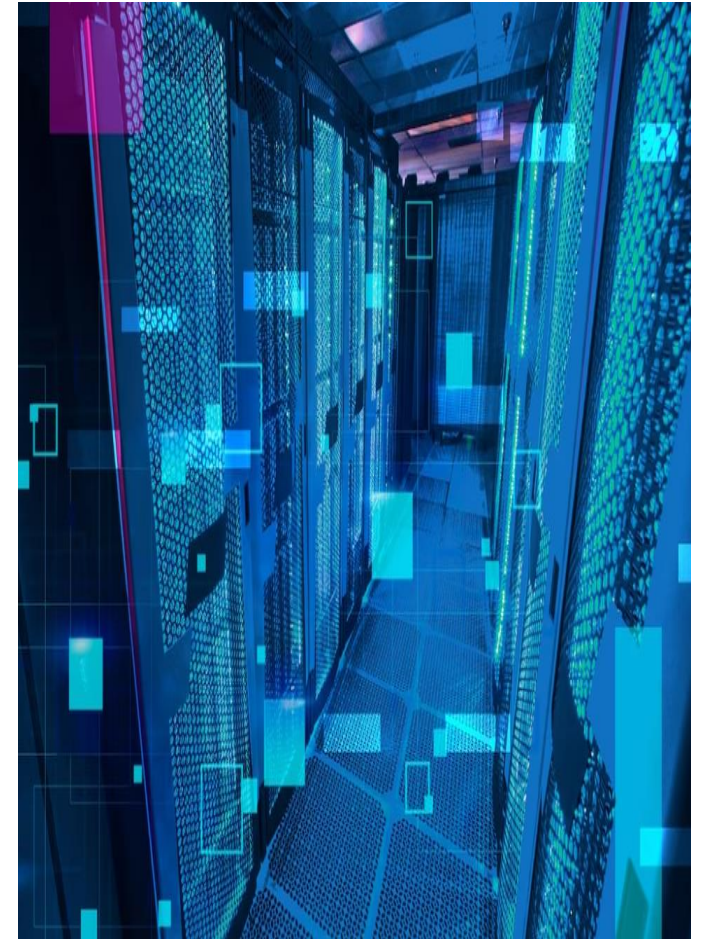
DISCIPLINA		
Codigo	Nome	Horas
CEA160	Cálculo Diferencial e Integral I	60
CSI030	Programação de Computadores I	60
CSI440	Banco de Dados I	60
CSI443	Matemática Discreta	60
CSI488	Algoritmos e Estruturas de Dados I	60

PREREQUISITO	
CodigoDisciplina	CodigoPrerequisito
CSI440	CSI443
CSI440	CSI488
CSI488	CSI030

OFERTA							
Codigo	CodigoDisciplina	Semestre	Ano	Nota	Frequencia	Aluno	Professor
1	CSI030	1	2019	8.6	90%	1918023	10120214450
2	CSI030	1	2019	7.4	95%	1918142	10120214450
3	CSI030	1	2019	2.5	50%	1828201	10120214450
4	CSI440	1	2019	4.7	75%	1918142	11040540330
5	CSI160	1	2019	9.5	100%	1828201	10560930210
6	CSI443	1	2019	7.7	82%	1918023	10010220470
7	CSI030	2	2019	6.0	90%	1828201	10432029020
8	CSI440	2	2019	6.7	85%	1918142	10190267390
9	CSI160	2	2019	8.2	92%	1918023	10560930210
10	CSI443	2	2019	9.0	100%	1828201	10010220470

Exemplo

Para *construir* o banco de dados, são armazenados dados para representar cada estudante, disciplina, oferta de disciplina, atribuição de nota e pré-requisito. Observe que várias tabelas podem estar relacionadas. Por exemplo, a tabela pré-requisito relaciona um elemento da tabela disciplina a outro da mesma tabela. A maioria dos bancos de dados apresentam **relacionamentos** entre suas tabelas.



Exemplo

PROFESSOR		
CPF	Nome	Salario
10120214450	Lucas Menezes	5420.00
11040540330	Bruno Santos	4200.00
10560930210	Marcos Lima	7150.00
10010220470	Elton Cardoso	6500.00
10190267390	Rafael Silva	2500.00
10432029020	Fernando Vaz	3880.00

ALUNO			
Matricula	Nome	DataNascimento	Curso
1918023	Camila Braga	12/03/1999	COM
1918142	João Lage	20/11/2001	PRO
1828201	Pedro Santos	02/09/2000	COM

DISCIPLINA		
Codigo	Nome	Horas
CEA160	Cálculo Diferencial e Integral I	60
CSI030	Programação de Computadores I	60
CSI440	Banco de Dados I	60
CSI443	Matemática Discreta	60
CSI488	Algoritmos e Estruturas de Dados I	60

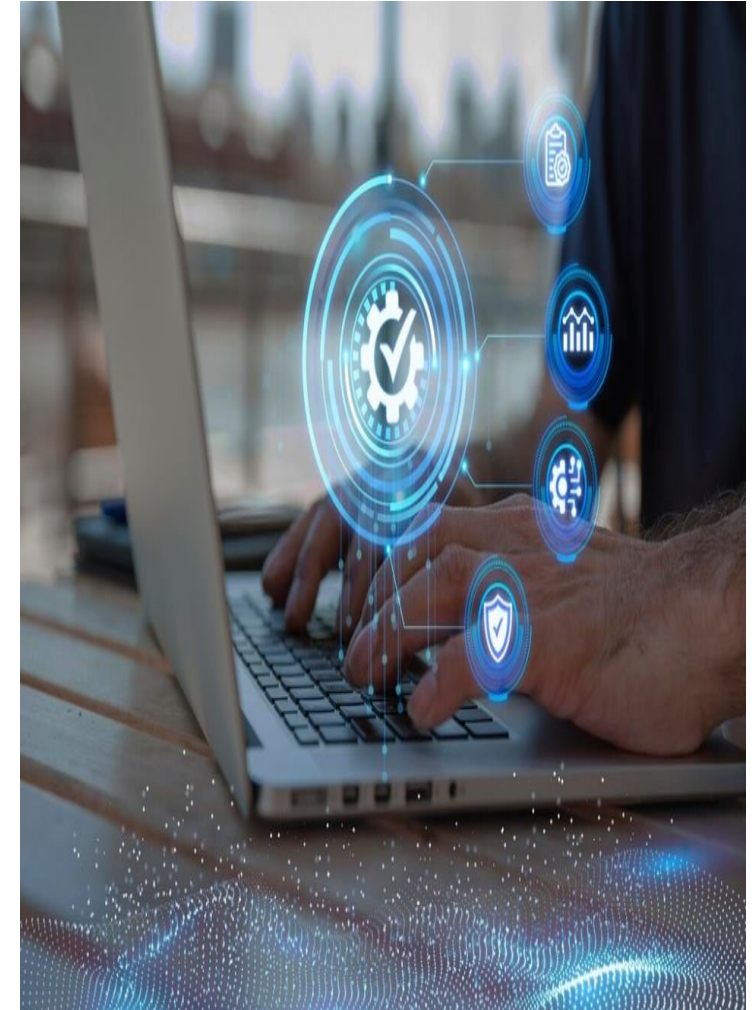
PREREQUISITO	
CodigoDisciplina	CodigoPrerequisito
CSI440	CSI443
CSI440	CSI488
CSI488	CSI030

OFERTA							
Codigo	CodigoDisciplina	Semestre	Ano	Nota	Frequencia	Aluno	Professor
1	CSI030	1	2019	8.6	90%	1918023	10120214450
2	CSI030	1	2019	7.4	95%	1918142	10120214450
3	CSI030	1	2019	2.5	50%	1828201	10120214450
4	CSI440	1	2019	4.7	75%	1918142	11040540330
5	CSI160	1	2019	9.5	100%	1828201	10560930210
6	CSI443	1	2019	7.7	82%	1918023	10010220470
7	CSI030	2	2019	6.0	90%	1828201	10432029020
8	CSI440	2	2019	6.7	85%	1918142	10190267390
9	CSI160	2	2019	8.2	92%	1918023	10560930210
10	CSI443	2	2019	9.0	100%	1828201	10010220470

Exemplo

A manipulação do banco de dados envolve consultas e atualização. Exemplos de consultas são:

- Recuperar as disciplinas ministradas pelo Prof. Bruno Santos;
- Selecionar os alunos que tirar nota igual ou superior a 6.0 em 'Matemática Discreta';
- Listar os pré-requisitos da disciplina 'Banco de Dados I'.



Exemplo

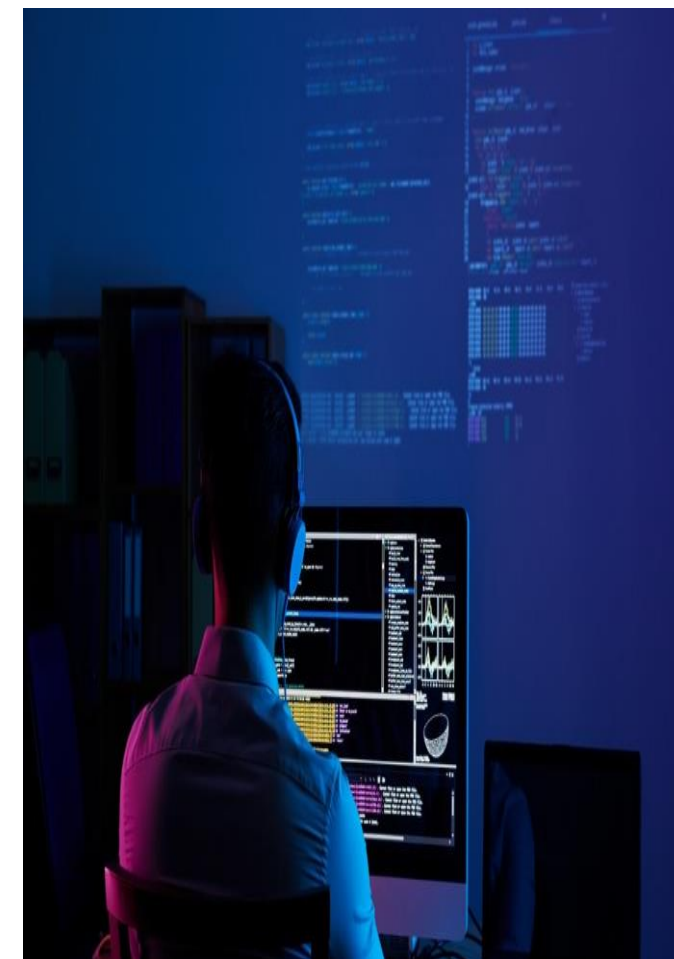
Exemplos de atualização incluem:

- Atualizar o curso do aluno Pedro Santos para PRO;
- Criar uma nova disciplina chamada 'Engenharia de Software';
- Alterar a nota de João Lage em 'Programação de Computadores I' para 8.0.



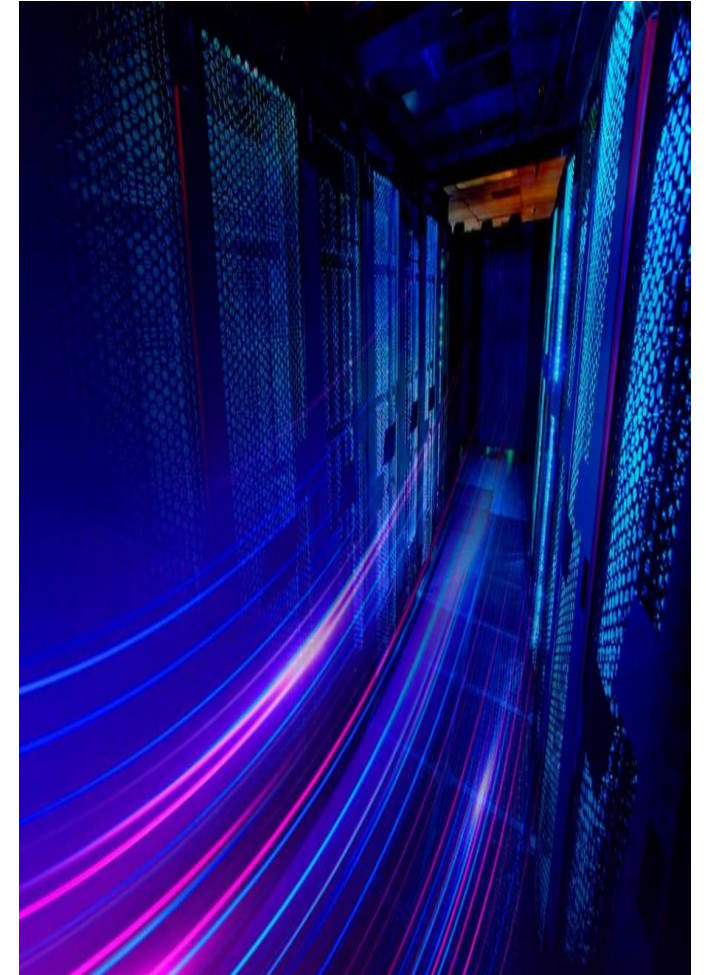
Características da Abordagem de Bancos de Dados

- Várias características distinguem a **abordagem de banco de dados** da **antiga abordagem de escrever programas personalizados para acessar dados em arquivos**.
- Na abordagem tradicional de processamento de arquivos, cada usuário define e implementa os arquivos necessários para o seu software como parte da programação da aplicação. Isso pode levar a problemas como desperdício de armazenamento, redundância de trabalho para manter os dados atualizados e maiores custos no desenvolvimento da aplicação.



Características da Abordagem de Bancos de Dados

- Na abordagem de banco de dados, um repositório único mantém os dados que são definidos apenas uma vez e acessados por uma variedade de usuários através de consultas e transações.



Características da Abordagem de Bancos de Dados

As principais características da abordagem de banco de dados frente à abordagem de processamento de arquivos são :

- Natureza Auto-Descritiva de um Sistema de Banco de Dados
- Isolamento entre Programa e Dados;
- Suporte a Múltiplas Visões dos Dados;
- Compartilhamento de Dados e Processamento de Transações Multiusuário;

Características da Abordagem de Bancos de Dados

As principais características da abordagem de banco de dados frente à abordagem de processamento de arquivos são :

- Natureza Auto-Descritiva de um Sistema de Banco de Dados
- Isolamento entre Programa e Dados;
- Suporte a Múltiplas Visões dos Dados;
- Compartilhamento de Dados e Processamento de Transações Multiusuário;

Natureza Auto-Descritiva de um Sistema de Banco de Dados

- Uma característica fundamental da abordagem de banco de dados é que um sistema de banco de dados não possui apenas o banco de dados mas também uma definição completa da estrutura do banco de dados e suas restrições.
- Essa definição é armazenada num catálogo de **meta-dados** do banco de dados.

Características da Abordagem de Bancos de Dados

As principais características da abordagem de banco de dados frente à abordagem de processamento de arquivos são :

- Natureza Auto-Descritiva de um Sistema de Banco de Dados
- Isolamento entre Programa e Dados;
- Suporte a Múltiplas Visões dos Dados;
- Compartilhamento de Dados e Processamento de Transações Multiusuário;

Isolamento entre Programa e Dados

- No processamento de arquivos tradicional, a estrutura dos arquivos de dados é embutida nas aplicações, assim, qualquer mudança na estrutura de um arquivo pode requerer alterações em todos os programas que acessar tal arquivo.
- Em contrapartida, aplicações que acessam SGBDs não precisam de tais mudanças na maioria dos casos.



Características da Abordagem de Bancos de Dados

As principais características da abordagem de banco de dados frente à abordagem de processamento de arquivos são :

- Natureza Auto-Descritiva de um Sistema de Banco de Dados
- Isolamento entre Programa e Dados;
- Suporte a Múltiplas Visões dos Dados;
- Compartilhamento de Dados e Processamento de Transações Multiusuário;

Suporte a Múltiplas Visões dos Dados

- Um banco de dados tem tipicamente muitos tipos de usuários, onde cada qual pode requerer uma visão diferente do BD. Uma visão pode ser um subconjunto dos dados ou conter dados virtuais que são derivados dos arquivos do banco de dados mas não são explicitamente armazenados.
- Em outras palavras, o banco de dados oferece a capacidade de apresentar diferentes perspectivas ou "visões" dos dados para diferentes usuários, dependendo de suas necessidades específicas. Cada usuário pode ter acesso a uma visão personalizada dos dados, que pode incluir apenas parte das informações ou informações calculadas dinamicamente a partir dos dados originais. Isso permite que o banco de dados seja flexível e adaptável às diferentes demandas e requisitos dos usuários que o utilizam.



Características da Abordagem de Bancos de Dados

As principais características da abordagem de banco de dados frente à abordagem de processamento de arquivos são :

- Natureza Auto-Descritiva de um Sistema de Banco de Dados
- Isolamento entre Programa e Dados;
- Suporte a Múltiplas Visões dos Dados;
- Compartilhamento de Dados e Processamento de Transações Multiusuário;

Compartilhamento de Dados e Processamento de Transações Multiusuário

Um SGBD permite que múltiplos usuários acessem o banco de dados ao mesmo tempo. Assim, os SGBDs têm que implementar o controle de concorrência para assegurar que quando vários usuários tentam atualizar os mesmos dados isso seja feito de forma controlada.



Vantagens ao Utilizar um SGBD

As **principais vantagens** da utilização de um SGBD são:

- Controle de Redundância;
- Restrição ao Acesso Não-Autorizado;
- Prover Armazenamento Persistente para Objetos de Programas;
- Prover Estruturas de Armazenamento e Técnicas de Busca;
- Prover Backup e Recuperação;
- Representar Relacionamentos Complexos entre Dados;

Controle de Redundância

Na abordagem de BD as visões de diferentes tipos de usuários são integradas no projeto do banco de dados.

Idealmente, deve-se ter um projeto de banco de dados que armazena cada item lógico de dados em apenas um lugar no BD. Isso é conhecido como **normalização de dados**, e assegura consistência e economiza espaço de armazenamento.



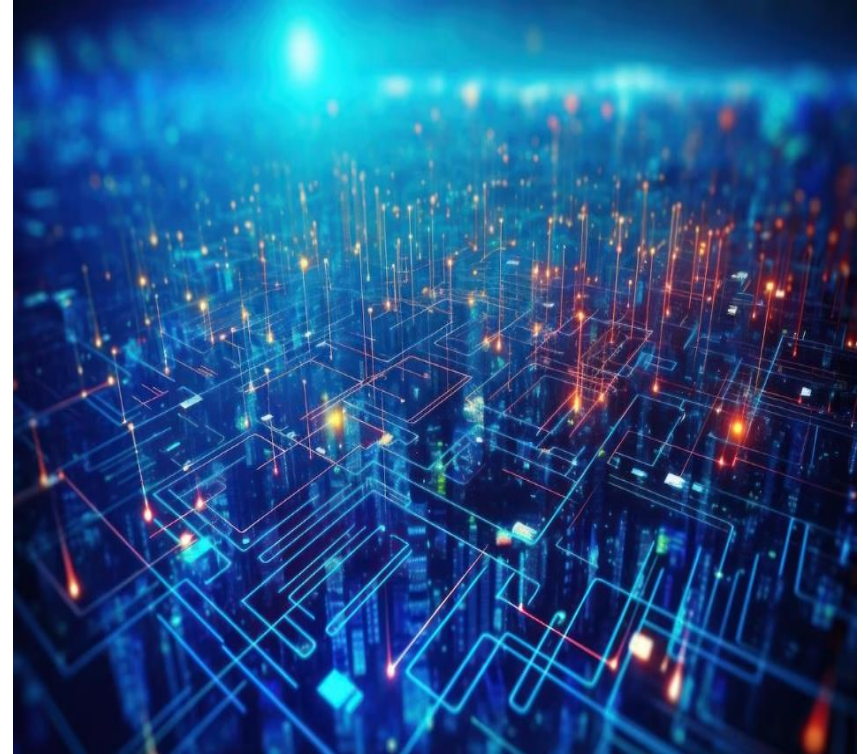
Vantagens ao Utilizar um SGBD

As principais vantagens da utilização de um SGBD são:

- Controle de Redundância;
- Restrição ao Acesso Não-Autorizado;
- Prover Armazenamento Persistente para Objetos de Programas;
- Prover Estruturas de Armazenamento e Técnicas de Busca;
- Prover Backup e Recuperação;
- Representar Relacionamentos Complexos entre Dados;

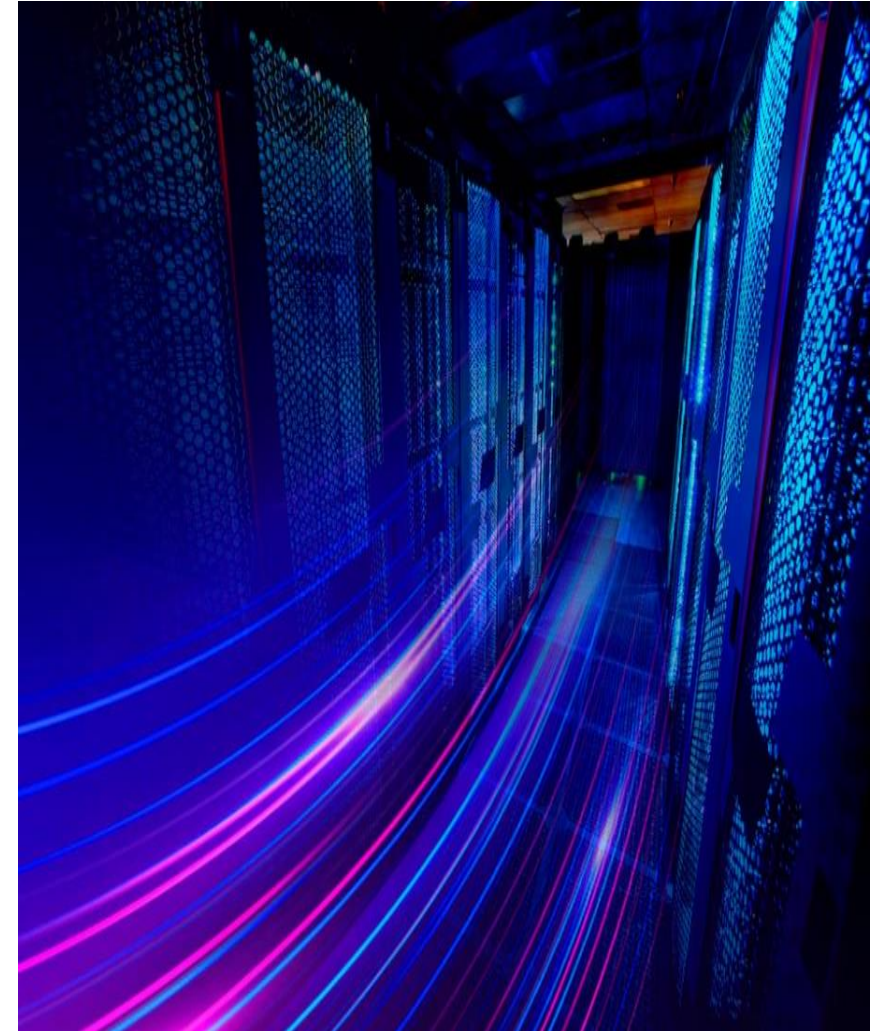
Restrição ao Acesso Não-Autorizado

Quando múltiplos usuários compartilham uma grande base de dados, é comum que muitos usuários não devam ter acesso a toda a informação no banco de dados.



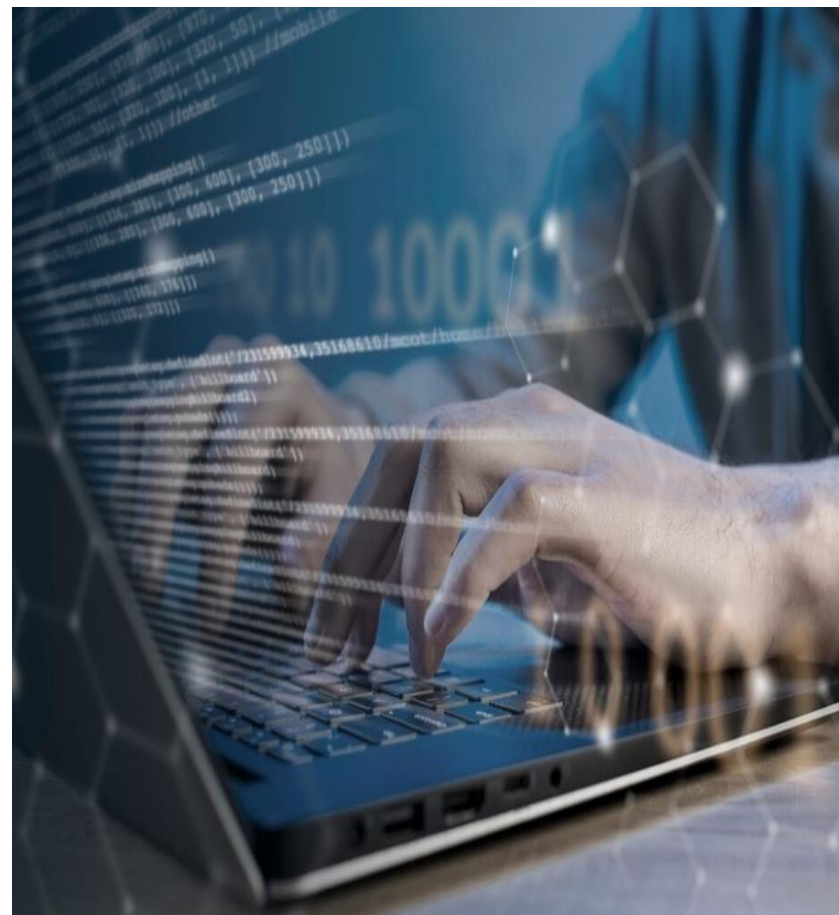
Restrição ao Acesso Não-Autorizado

- Por exemplo, dados financeiros como salários e bônus são comumente considerados confidenciais e apenas pessoas autorizadas podem acessar tais dados.
- Adicionalmente, alguns usuários podem ter permissão apenas para leitura dos dados enquanto outros podem fazer alterações.



Restrição ao Acesso Não-Autorizado

- Um SGBD deve prover um subsistema de segurança e autorização, o qual o administrador do banco de dados usa para criar contas e especificar restrições de acesso.
- Então, o SGBD deve assegurar essas restrições automaticamente.



Vantagens ao Utilizar um SGBD

As principais vantagens da utilização de um SGBD são:

- Controle de Redundância;
- Restrição ao Acesso Não-Autorizado;
- Prover Armazenamento Persistente para Objetos de Programas;
- Prover Estruturas de Armazenamento e Técnicas de Busca;
- Prover Backup e Recuperação;
- Representar Relacionamentos Complexos entre Dados;

Prover Armazenamento Persistente para Objetos de Programas

- Linguagens de programação tipicamente têm estruturas de dados complexas.
- Os valores das variáveis dos programas são descartados assim que o programa termina, a não ser que o programador explicitamente armazene eles em arquivos permanentes, que envolve converter essas estruturas complexas para um formato adequado ao armazenamento em arquivo.



Prover Armazenamento Persistente para Objetos de Programas

- Quando é necessário ler esses dados, o programador deve converter os dados do formato de arquivo para a estrutura de dados utilizada.
- SGBDs são compatíveis com as linguagens de programação mais populares, como Java, Python e C++ e realizam automaticamente todas as conversões necessárias

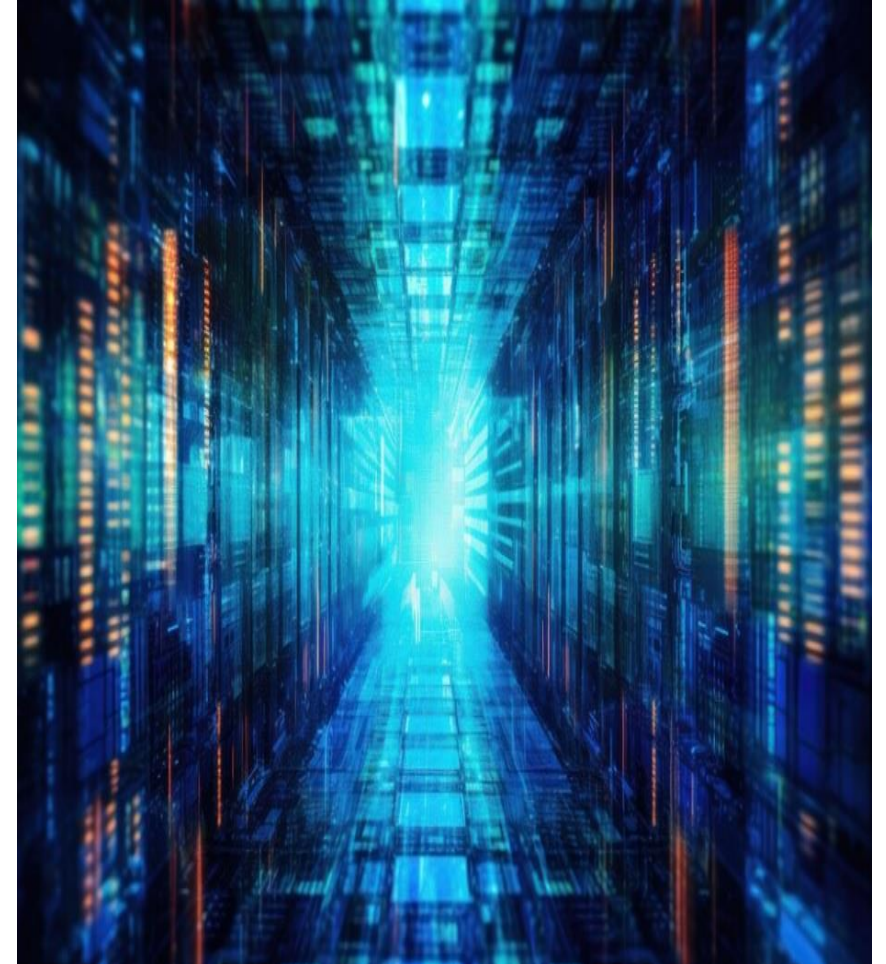


As principais vantagens da utilização de um SGBD são:

- Controle de Redundância;
- Restrição ao Acesso Não-Autorizado;
- Prover Armazenamento Persistente para Objetos de Programas;
- Prover Estruturas de Armazenamento e Técnicas de Busca;
- Prover Backup e Recuperação;
- Representar Relacionamentos Complexos entre Dados;

Prover Estruturas de Armazenamento e Técnicas de Busca

- Sistemas de bancos de dados devem prover capacidades para executar consultas e atualizações eficientemente.
- Dado que o banco de dados é armazenado em disco, o SGBD deve prover estruturas de dados especializadas e técnicas de busca para acelerar a busca no disco para os registros de interesse.



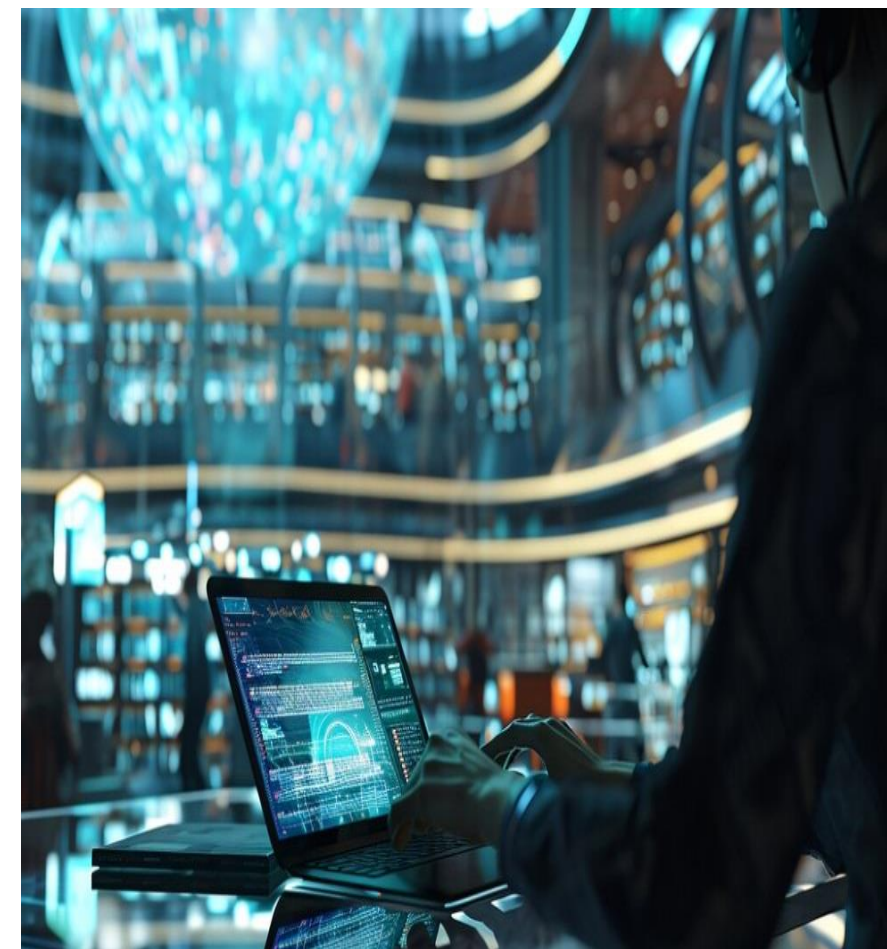
Vantagens ao Utilizar um SGBD

As principais vantagens da utilização de um SGBD são:

- Controle de Redundância;
- Restrição ao Acesso Não-Autorizado;
- Prover Armazenamento Persistente para Objetos de Programas;
- Prover Estruturas de Armazenamento e Técnicas de Busca;
- Prover Backup e Recuperação;
- Representar Relacionamentos Complexos entre Dados;

Prover Backup e Recuperação

- Um SGBD deve prover mecanismos para recuperar de falhas de software ou de hardware. O subsistema de backup e recuperação de um SGBD é responsável para tal.
- Por exemplo, se um computador desliga no meio de uma transação complexa de atualização, o sistema de recuperação é responsável por assegurar que o banco de dados é restaurado para o estado que estava antes da transação começar a executar.



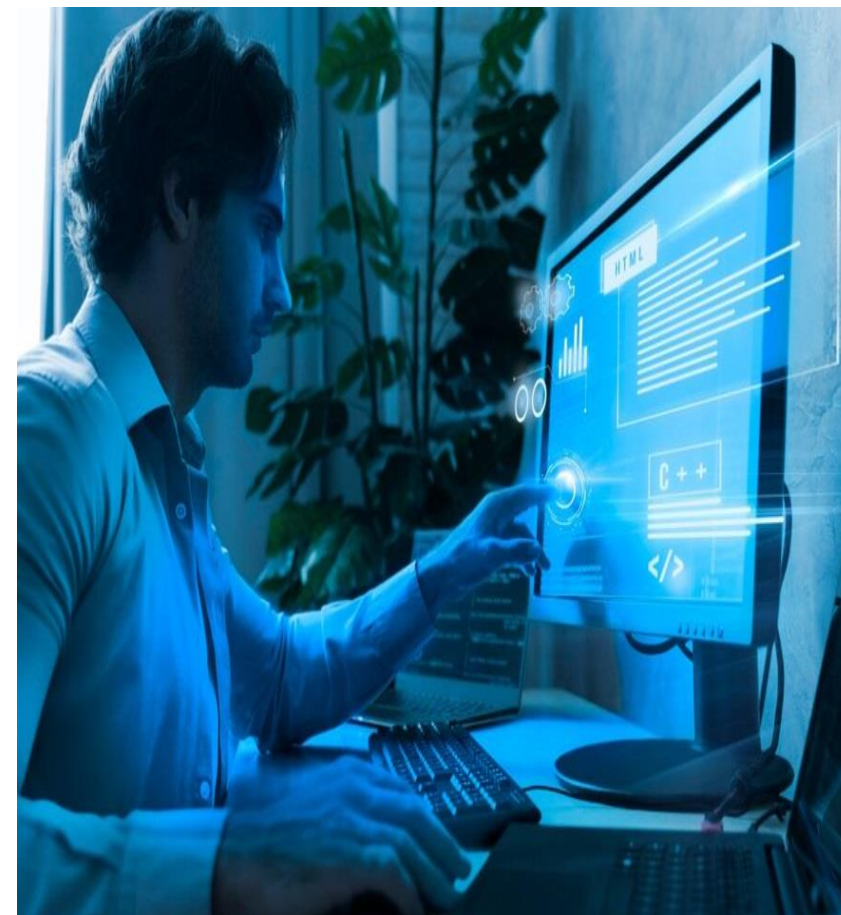
Vantagens ao Utilizar um SGBD

As principais vantagens da utilização de um SGBD são:

- Controle de Redundância;
- Restrição ao Acesso Não-Autorizado;
- Prover Armazenamento Persistente para Objetos de Programas;
- Prover Estruturas de Armazenamento e Técnicas de Busca;
- Prover Backup e Recuperação;
- Representar Relacionamentos Complexos entre Dados;

Representar Relacionamentos Complexos entre Dados

- Um banco de dados deve incluir numerosas variedades de dados que são inter-relacionados de muitas formas. Por exemplo, considere o banco de dados da figura do próximo slide.
- O estudante 'Pedro' está relacionado com três registros na tabela OFERTA. Similarmente, cada registro em OFERTA é relacionada com uma disciplina e um professor.



Representar Relacionamentos Complexos entre Dados

O estudante 'Pedro' está relacionado com três registros na tabela OFERTA. Similarmente, cada registro em OFERTA é relacionada com uma disciplina e um professor.

PROFESSOR			ALUNO			
CPF	Nome	Salario	Matricula	Nome	DataNascimento	Curso
10120214450	Lucas Menezes	5420.00	1918023	Camila Braga	12/03/1999	COM
11040540330	Bruno Santos	4200.00	1918142	João Lage	20/11/2001	PRO
10560930210	Marcos Lima	7150.00	1828201	Pedro Santos	02/09/2000	COM
10010220470	Elton Cardoso	6500.00				
10190267390	Rafael Silva	2500.00				
10432029020	Fernando Vaz	3880.00				

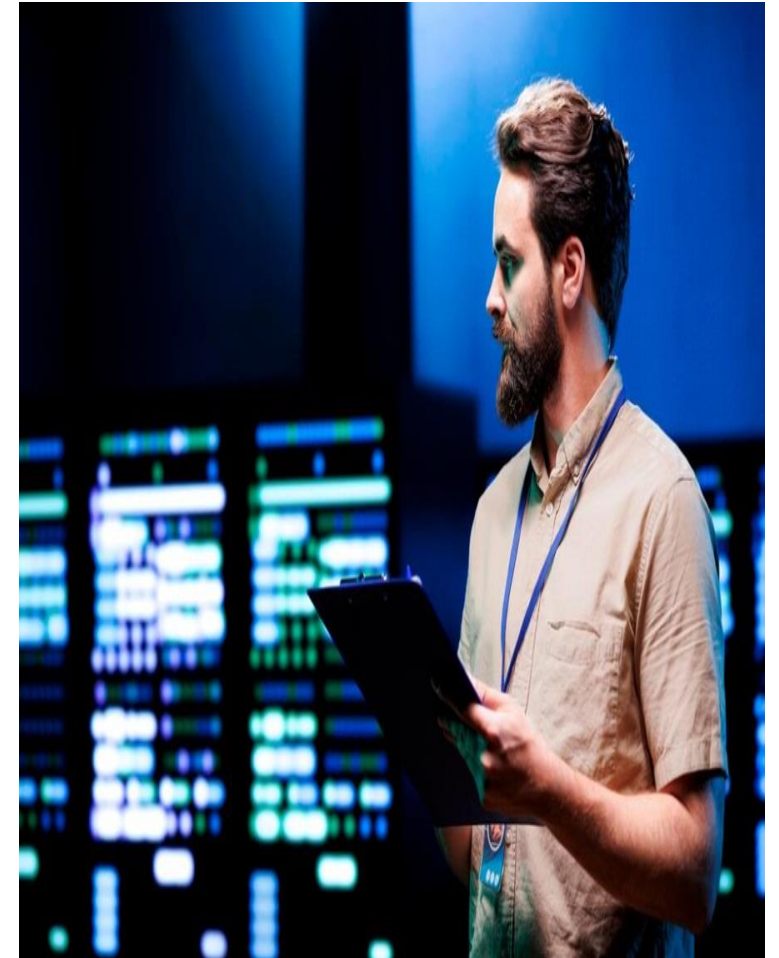
DISCIPLINA			PREREQUISITO	
Codigo	Nome	Horas	CodigoDisciplina	CodigoPrerequisito
CEA160	Cálculo Diferencial e Integral I	60	CSI440	CSI443
CSI030	Programação de Computadores I	60	CSI440	CSI488
CSI440	Banco de Dados I	60	CSI488	CSI030
CSI443	Matemática Discreta	60		
CSI488	Algoritmos e Estruturas de Dados I	60		

OFERTA							
Codigo	CodigoDisciplina	Semestre	Ano	Nota	Frequencia	Aluno	Professor
1	CSI030	1	2019	8.6	90%	1918023	10120214450
2	CSI030	1	2019	7.4	95%	1918142	10120214450
3	CSI030	1	2019	2.5	50%	1828201	10120214450
4	CSI440	1	2019	4.7	75%	1918142	11040540330
5	CSI160	1	2019	9.5	100%	1828201	10560930210
6	CSI443	1	2019	7.7	82%	1918023	10010220470
7	CSI030	2	2019	6.0	90%	1828201	10432029020
8	CSI440	2	2019	6.7	85%	1918142	10190267390
9	CSI160	2	2019	8.2	92%	1918023	10560930210
10	CSI443	2	2019	9.0	100%	1828201	10010220470

Structured Query Language (SQL) é a linguagem padrão para manipular bancos de dados relacionais. A SQL foi desenvolvida nos anos 70 pela IBM e foi adotada por várias outras companhias que desenvolvem SGBDs

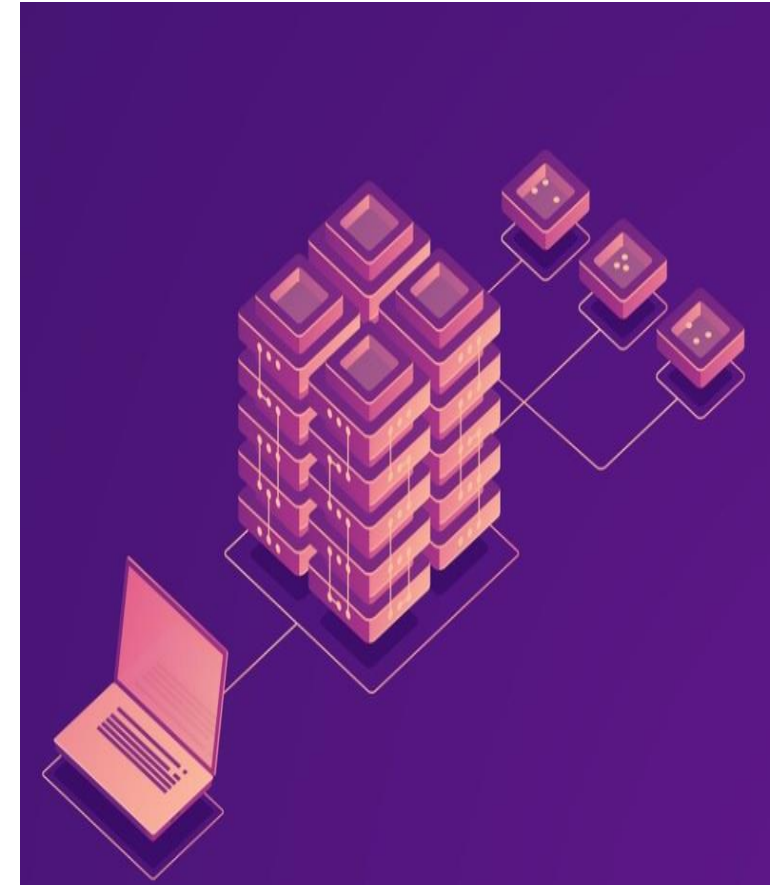
Definição e Tipos de Dados em SQL:

SQL usa os termos **tabela**, **linha** e **coluna** para os termos relacionais **relação**, **tupla** e **atributo**, respectivamente.



Criando um Banco de dados:

```
CREATE DATABASE databasename;
```



CREATE

Criando uma tabela:

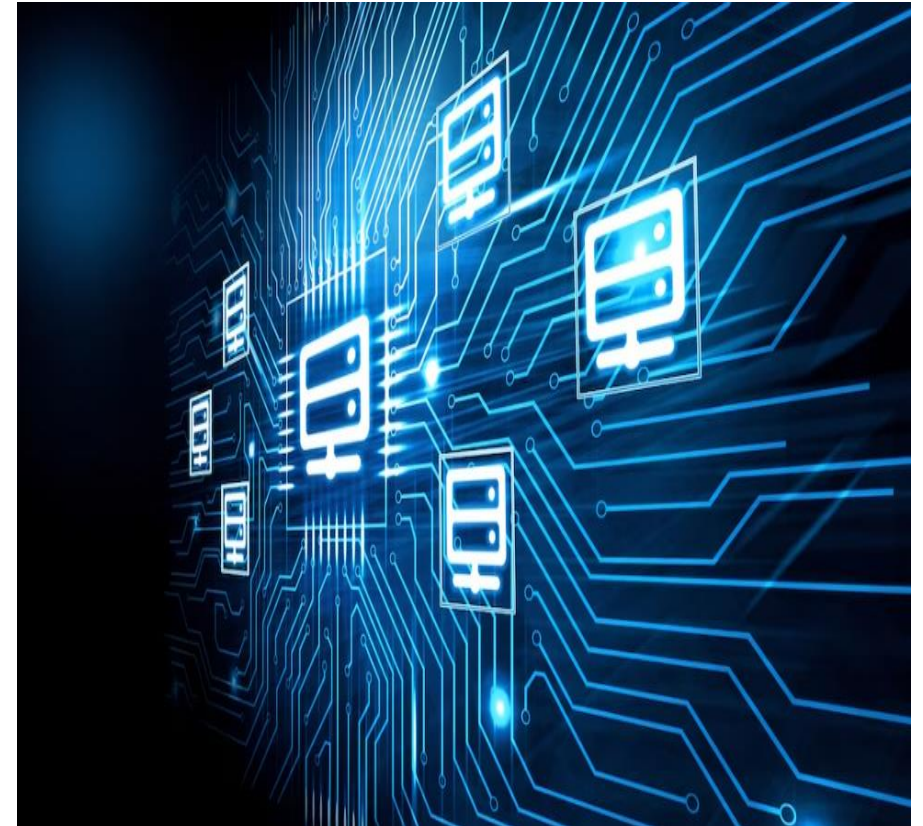
```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```



DROP

Apagando um Banco de Dados:

```
DROP DATABASE databasename;
```



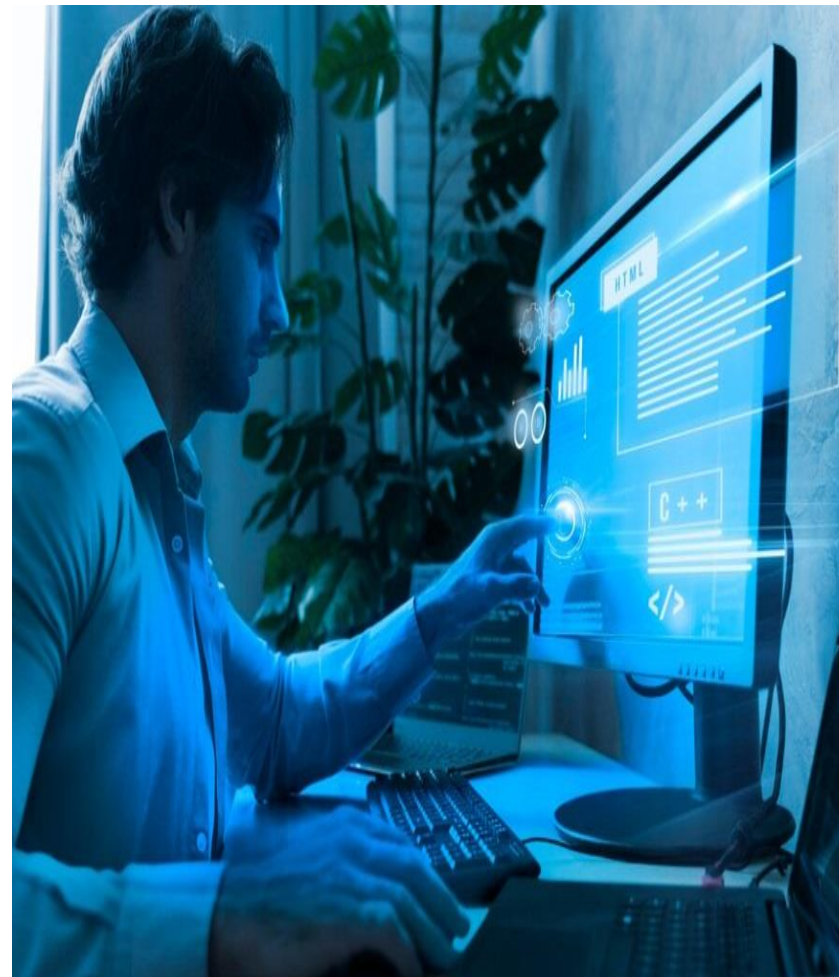
DROP

Apagando um Banco de Dados:

```
DROP DATABASE databasename;
```

Apagando uma Tabela:

```
DROP TABLE table_name;
```



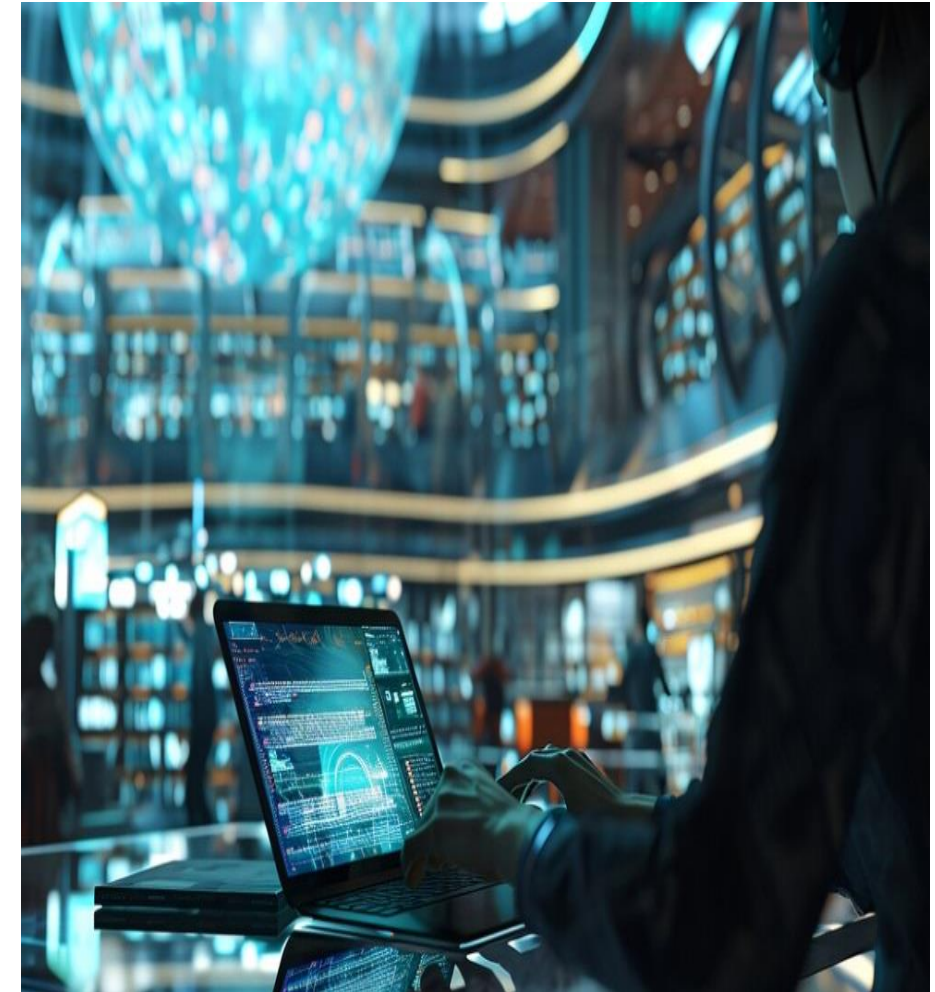
ALTER

Alterando uma tabela - Adição de uma Coluna

```
ALTER TABLE table_name  
ADD column_name datatype;
```

Alterando uma tabela - Remoção de uma coluna:

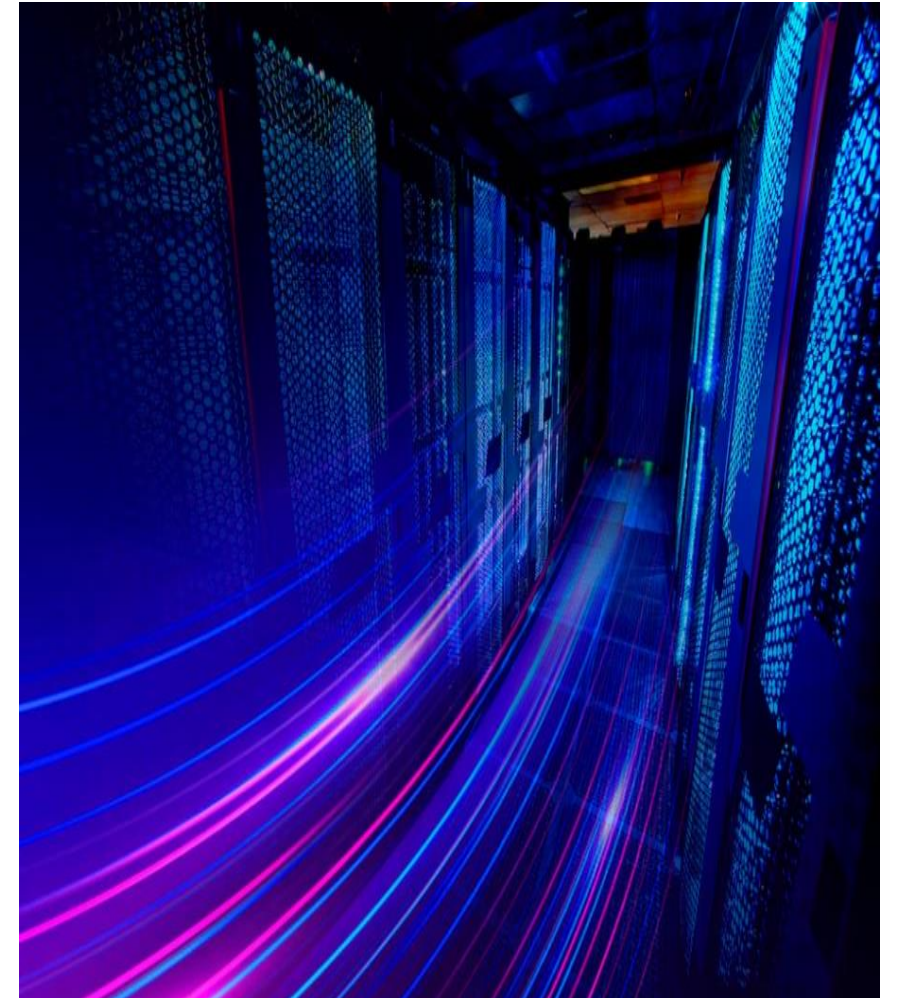
```
ALTER TABLE table_name  
DROP COLUMN column_name;
```



ALTER

Alterando uma tabela Modificando o tipo de atributo de uma coluna

```
ALTER TABLE table_name  
ALTER COLUMN column_name datatype;
```

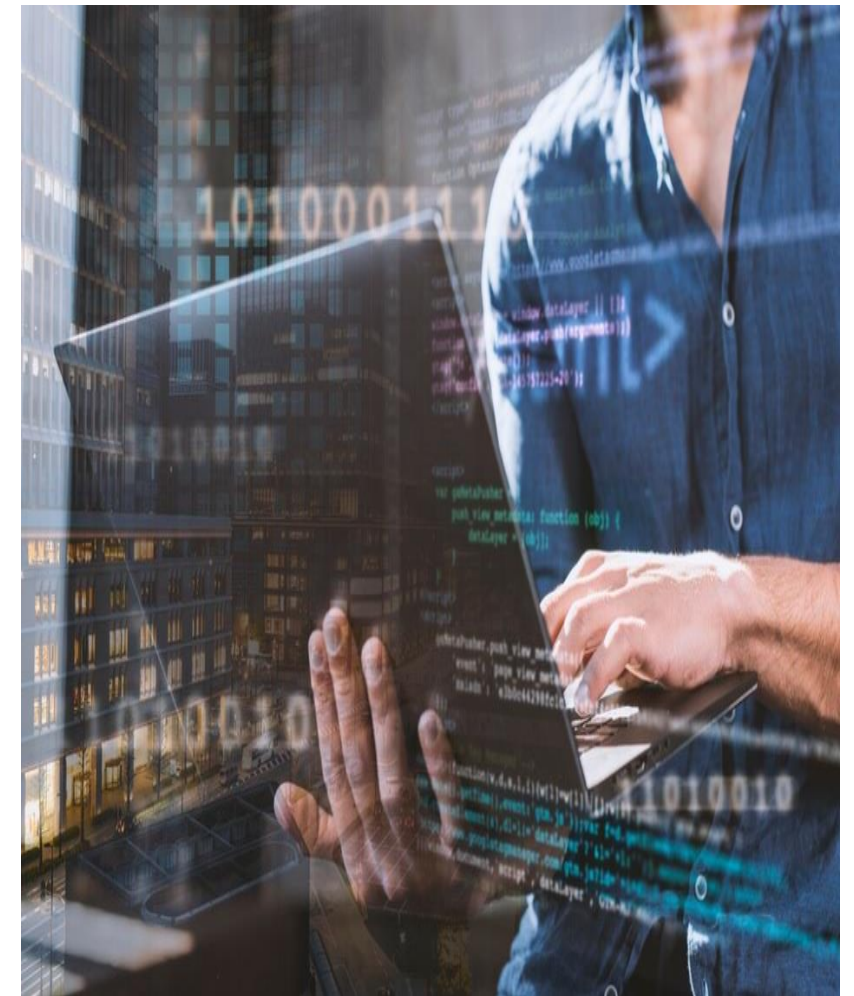


SELECT

A instrução SELECT é usada para selecionar dados de um banco de dados.

```
SELECT column1, column2, ...  
FROM table_name;
```

```
SELECT * FROM table_name;
```



Exemplo de questão

8 – Um grupo de cientistas deseja organizar melhor as suas informações sobre dinossauros descobertos e lhe enviou os dados abaixo como exemplo:

Dinossauro	Grupo	Toneladas	Ano Descoberta	Descobridor	Era	Início (milhões)	Fim (milhões)	País
Saichania	Anquilossauros	4	1977	Maryanska	Cretáceo	145	66	Mongólia
Tricerátops	Ceratopsídeos	6	1887	John Bell Hatcher	Cretáceo	70	66	Canadá
Kentrossauro	Estegossauros	2	1909	Cientistas Alemães	Jurássico	200	145	Tanzânia
Pinacossauro	Anquilossauros	6	1999	Museu Americano de História Natural	Triássico	85	75	China

Utilizando os seus conhecimentos de banco de dados e SQL, faça uma consulta que retorne os dinossauros descobertos com mais de 5 toneladas.

Resposta da questão

8 - Utilizando os seus conhecimentos de banco de dados e SQL, faça uma consulta que retorne os dinossauros descobertos com mais de 5 toneladas.

Obs:. Colocamos o nome da tabela de `dinossauros_descobertos` e a base de dados já selecionada `dinossauros`.

Resposta:

```
SELECT * FROM `dinossauros_descobertos` WHERE dinossauros_descobertos.Toneladas>5
```

A instrução INSERT é usada para inserir novos registros em uma tabela.

Especifique os nomes das colunas e os valores a serem inseridos:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO `contatos` (`id`, `nome`, `idade`, `dataCadastro`) VALUES ('1', 'Maria', '54', '2022-09-19');
```

Se você estiver adicionando valores para todas as colunas da tabela, não precisará especificar os nomes das colunas na consulta SQL. No entanto, certifique-se de que a ordem dos valores esteja na mesma ordem das colunas na tabela. Aqui, a sintaxe INSERT INTO seria a seguinte:

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO `contatos` VALUES ('1', 'Maria', '54', '2022-09-19');
```

A instrução UPDATE é usada para modificar os registros existentes em uma tabela.

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Exemplo:

```
UPDATE `contatos` SET `idade` = '37' WHERE `contatos`.`id` = 1;
```

DELETE

A instrução DELETE é usada para excluir registros existentes em uma tabela.

```
DELETE FROM table_name WHERE condition;
```

Exemplo:

```
"DELETE FROM `contatos` WHERE `contatos`.`id` = 2";
```

Exercício-01:

Instalar o SGDB's MariaDB (Documentação: <https://mariadb.com/kb/en/documentation/>)

que vem junto com o pacote Xampp: https://www.apachefriends.org/pt_br/download.html

1) Tentar resolver o exercício dos dinossauros conforme foi realizado em aula. Criar a base de dados, tabela, inserir dados, e finalmente fazer as consultas.

Xampp versão antiga Professor Andre (mais estável):

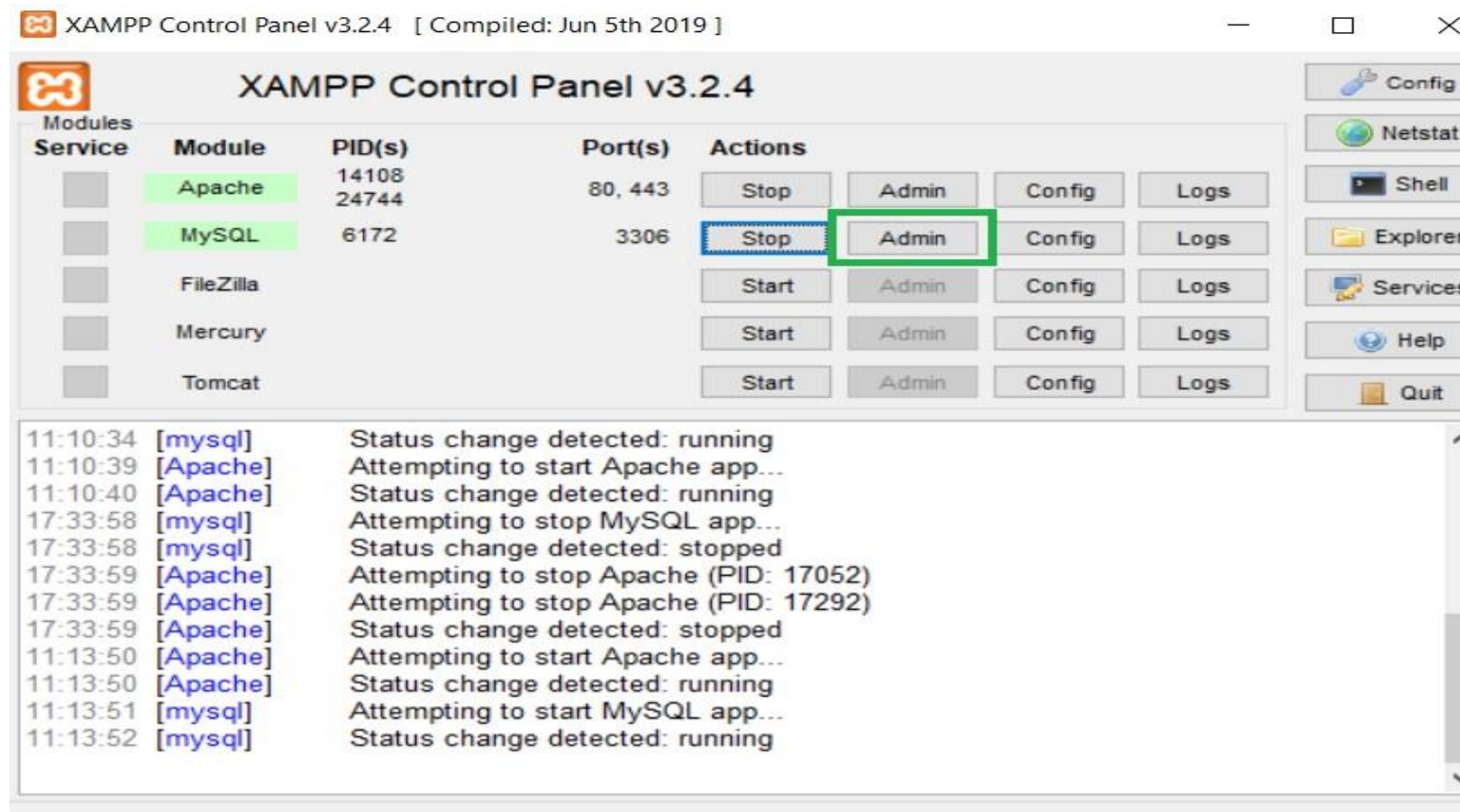
<https://drive.google.com/drive/folders/1QKlwp68AqhnyVBbSM2Jk7HdX57Lqa3hS?usp=sharing>

Existe uma forma de exportar toda a estrutura do BD em formato '*.sql' através da ferramenta 'phpmyadmin' que vem junto do pacote Xampp.

Entregar o arquivo 'nome.SQL' exportado com toda a estrutura do BD e os dados inseridos no mysql.

Dicas - Criando o banco de dados

- Após instalar o Xampp, vamos utilizar o **SGBD MariaDB**: <https://mariadb.com/kb/en/documentation/> que vem instalado no pacote para criar o banco de dados.



Criando o banco de dados

- Criando uma nova BASE DE DADOS no phpmyadmin que administra e exhibe graficamente o SGBD MariaDB ou mysql.

The screenshot displays the phpMyAdmin web interface. On the left sidebar, the 'Novo' (New) button is highlighted with a red rectangle. The main content area shows the 'Definições gerais' (General) and 'Configurações de aspecto' (Appearance) panels. The 'Definições gerais' panel includes a dropdown for 'Collation de conexão do servidor MySQL' set to 'utf8mb4_unicode_ci'. The 'Configurações de aspecto' panel shows the 'Língua - Language' set to 'Português - Portuguese' and the 'Tema' (Theme) set to 'pmahomme'. On the right, there are three summary panels: 'Servidor de base de dados' (Database server), 'Servidor web' (Web server), and 'phpMyAdmin'. The 'Servidor de base de dados' panel lists details such as 'Servidor: 127.0.0.1 via TCP/IP', 'Tipo de servidor: MariaDB', and 'Versão do servidor: 10.4.14-MariaDB'. The 'Servidor web' panel lists 'Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.10'. The 'phpMyAdmin' panel lists version information and links to documentation and support.

Criando o banco de dados

- Nome da base de dados: **Dinossauros**
- Agrupamento: **utf8mb4_general_ci**



Criando o banco de dados

Preencher os dados para criar a primeira tabela na base de dados e clicar em 'guarda' ou 'executar' dependendo da versão.

The screenshot shows the phpMyAdmin interface for creating a new table. The browser address bar indicates the server is 127.0.0.1, the database is 'dinossauros', and the table being created is 'Dinossauros_descobertos'. The interface includes a sidebar with a database tree and a main content area with various tabs like 'Procurar', 'Estrutura', 'SQL', etc. The 'Add new table' form is active, showing the table name and a button to 'Executar'. Below this, there are fields for defining columns: 'Inicio' (INT, 15), 'Fim' (INT, 15), and 'Pais' (VARCHAR, 50). Each column has a dropdown to 'Escolher a partir das Colunas Centrais'. At the bottom, there are sections for 'Comentários da tabela', 'Colaço', 'Motor de armazenamento' (set to InnoDB), and 'Definição da PARTIÇÃO'. A red box highlights the 'Guarda' button at the bottom right.

Criando o banco de dados

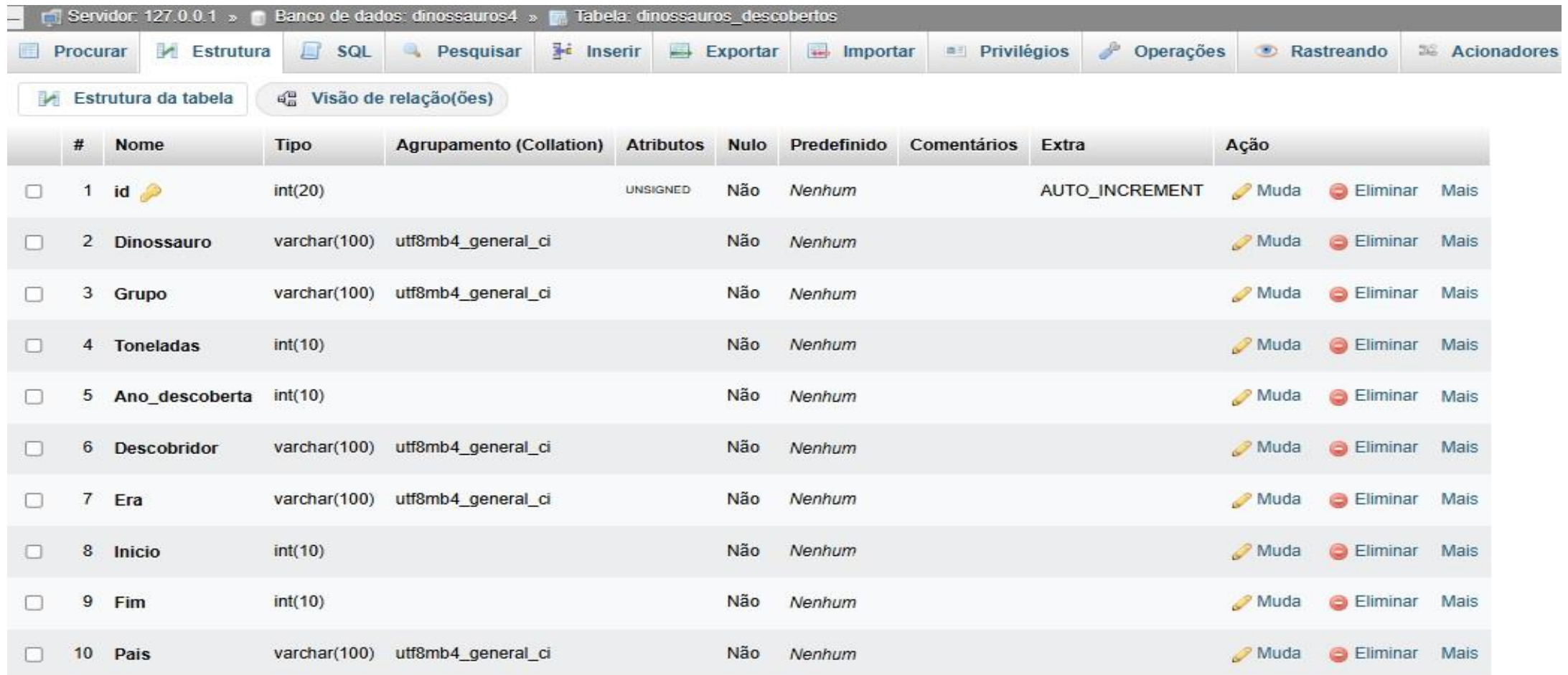
A tabela padronizada com um 'Id' para cada Dinossauro, está logo abaixo. Usaremos nesse formato no Banco de Dados.

id	Dinossauro	Grupo	Toneladas	Ano_descoberta	Descobridor	Era	Inicio	Fim	País
1	Saichania	Anquilossauros	4	1977	Maruanska	Cretáceo	145	66	Mongólia
2	Tricerátops	Ceratopsídeos	6	1887	John Bell Hatcher	Cretáceo	70	66	Canadá
3	Kentrossaur o	Estegossauros	2	1909	Cientistas alemães	Jurássico	200	145	Tanzânia
4	Pinacossaur o	Anquilossauros	6	1999	Museu Americano de História Natural	Triássico	85	75	China

Comentário: Usaremos a coluna `id` como **chave primária (primary key)** da tabela — um identificador **único e não nulo** para cada registro, que **não se repete** e permite relacionar esta tabela com outras de forma segura.

Criando o banco de dados

- Se tudo ocorreu certo a tabela **dinossauros_descobertos** foi criada na base de dados **dinossauros**



The screenshot shows a database management interface with a toolbar at the top containing buttons for 'Procurar', 'Estrutura', 'SQL', 'Pesquisar', 'Inserir', 'Exportar', 'Importar', 'Privilegios', 'Operações', 'Rastreado', and 'Acionadores'. Below the toolbar, there are two tabs: 'Estrutura da tabela' (selected) and 'Visão de relação(ões)'. The main area displays a table structure for 'Tabela: dinossauros_descobertos'.

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id	int(20)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT	Muda Eliminar Mais
<input type="checkbox"/>	2	Dinossauro	varchar(100)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	3	Grupo	varchar(100)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	4	Toneladas	int(10)			Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	5	Ano_descoberta	int(10)			Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	6	Descobridor	varchar(100)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	7	Era	varchar(100)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	8	Inicio	int(10)			Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	9	Fim	int(10)			Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	10	Pais	varchar(100)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais

Inserindo dados no banco de dados

- Agora vamos inserir os dados na tabela `dinossauros_descobertos` e adicionar a 'primary key'. Como o 'id' foi configurado como Auto Incremento, não precisamos passar o valor dele, o banco de dados faz isso automático. Passamos os parâmetros apenas para os outros dados.

Comandos SQL para inserir no banco de dados:

1)

```
INSERT INTO dinossauros_descobertos
```

```
(Dinossauro, Grupo, Toneladas, Ano_descoberta, Descobridor, Era, Inicio, Fim, Pais)
```

```
VALUES
```

```
('Saichania', 'Anquilossauros', 4, 1977, 'Maruanska', 'Cretáceo', 145, 66, 'Mongólia');
```

2)

```
INSERT INTO dinossauros_descobertos
```

```
(Dinossauro, Grupo, Toneladas, Ano_descoberta, Descobridor, Era, Inicio, Fim, Pais)
```

```
VALUES
```

```
('Tricerátops', 'Ceratopsídeos', 6, 1887, 'John Bell Hatcher', 'Cretáceo', 70, 66, 'Canadá');
```

3)

```
INSERT INTO dinossauros_descobertos
```

```
(Dinossauro, Grupo, Toneladas, Ano_descoberta, Descobridor, Era, Inicio, Fim, Pais)
```

```
VALUES
```

```
('Kentrossauro', 'Estegossauros', 2, 1909, 'Cientistas alemães', 'Jurássico', 200, 145, 'Tanzânia');
```

Inserindo dados no banco de dados

- Continuação

4)

```
INSERT INTO dinossauros_descobertos
```

```
    (Dinossauro, Grupo, Toneladas, Ano_descoberta, Descobridor, Era, Inicio, Fim, Pais)
```

```
VALUES
```

```
    ('Pinacossauro', 'Anquilossauros', 6, 1999, 'Museu Americano de História Natural', 'Triássico', 85, 75, 'China');
```

```
-- Conferir
```

```
SELECT * FROM dinossauros_descobertos;
```

Exportar o arquivo '.sql' para entregar com as outras atividades de aula.

Colocar num arquivo zip ou rar para entregar na atividade da aula de hoje.

Referências

HOGAN, Brian P. HTML 5 e CSS3: Desenvolva hoje com o padrão de amanhã . Rio de Janeiro (RJ): Ciência Moderna Ltda., 2012 282 p. ISBN 978-85-399-0260-6

Fábio Flatschart. HTML 5 - Embarque Imediato, 2011. (Biblioteca Digital)

Deitel, Paul J.; Deitel, Harvey M.. Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores, 2008. (Biblioteca Digital)

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. 2. ed. São Paulo, SP: Novatec, 2014. 335 p. ISBN 978-85-7522-403-8.

Denilson Bonatti. Desenvolvimento de Jogos em HTML5, 2014. (Biblioteca Digital)

W3SCHOOL. The world's largest web development site. Acessado em 2018. Disponível em: <http://www.w3schools.com/>.

W3C. World Wide Web Consortium. Acessado em 2018. Disponível em: <http://www.w3.org>

MORRISON, Michael. Use a cabeça JavaScript. Rio de Janeiro (RJ): Alta Books, 2008. 606 p. <https://br.freepik.com/fotos-vetores-gratis/desenvolvimento-web>

Material do Professor Fabrício Tonetto Londero, 2023.

Obrigado pela atenção!!



Email: andre.flores@ufn.edu.br