

Arquivo JSON: JSON - *Java Script Object Notation*: esse tipo de arquivo é muito utilizado em API de empresas. API's: são meios de comunicação entre os serviços oferecidos por empresas como Twitter, Facebook, Youtube, Embrapa e outros desenvolvedores.

1 - Trabalhando com JSON

```
import json

retorno = json.dumps(['Produtos', {'Notebook Dell': ('2TB', 'Novo', '110', 8450)}])
print(type(retorno))
print(retorno)
```

```
<class 'str'>
["Produtos", {"Notebook Dell": ["2TB", "Novo", "110", 8450]}]
```

A função. dump() formata os dados no **padrão JSON**, substituindo onde há aspas simples por aspas duplas.

2 - Trabalhando com JSON

```
import json

class Gato:

    def __init__(self, nome, raca):
        self.__nome = nome
        self.__raca = raca

    @property
    def nome(self):
        return self.__nome

    @property
    def raca(self):
        return self.__raca

ravena = Gato('Ravena', 'Ashera')

print(ravena.__dict__)

retorno = json.dumps(ravena.__dict__)

print(type(retorno))
print(retorno)
```

```
{'_Gato__nome': 'Ravena', '_Gato__raca': 'Ashera'}
<class 'str'>
{"_Gato__nome": "Ravena", "_Gato__raca": "Ashera"}
```

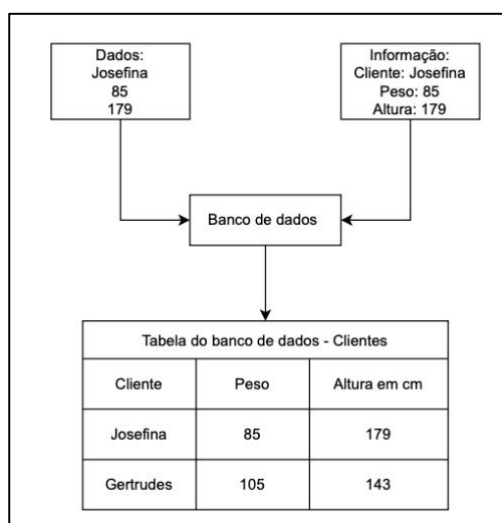
Observe que a variável retorno foi formatada com aspas duplas no padrão json.

Dados: São elementos bruto geralmente resultantes de uma entrada do usuário ou de outro sistema armazenado em um banco de dados.

Informação: São os elementos brutos tratados vindo de um banco de dados que acrescenta algum conhecimento da realidade analisada. Note que a informação é gerada a partir dos dados (Machado, 2012 / 2016)

Banco de dados: é uma coleção de dados organizados que podem ou não estarem inter-relacionados e que atendem a uma aplicação ou família de aplicações. (HEUSER, 2019).

Tabela representando dados, informação e possível estrutura de um banco de dados.



Fonte: o autor

Banco de dados e modelo de dados: um banco de dados contém dois componentes: os **dados** e o **modelo de dados**.

Um modelo de dados é uma definição formal da estrutura de um banco de dados que obedece a um padrão de armazenamento. Na tabela acima cada cliente possui um conjunto de dados (Cliente, Peso, Altura) que pode ser entendido como o **modelo de dados**. As informações específicas de cada um dos clientes (o nome, o peso, a altura) fazem parte do **componente dados** do banco de dados.

Dados inter-relacionados: esse conceito se refere ao suporte a uma aplicação ou a um conjunto de aplicações. Quando falamos em aplicação, estamos nos referindo a um conjunto de programas que colaboram para atender a um ou mais usuários como um *software* de reserva de passagens aéreas, ou um *software* que gerência o conteúdo de *blogs* na *Web*, onde o primeiro fora projetado para atender a múltiplos programas e o último projetado para atender a um programa específico (Heuser, 2019).

Existem diferente tipos de banco de dados, nessa disciplina o objetivo não é aprender sobre banco de dados de forma geral e sim focar nas duas principais

vertentes de banco de dados utilizados no momento que são: **banco de dados relacionais** e **banco de dados não-relacional**.

Banco de Dados Relacional x Não Relacional – SQL x NoSQL

Banco de dados não relacionais ou NoSQL, armazenam dados sem especificações prévias bem estabelecidas. Segundo F. V. C. (2020), a *big data* está abrindo portas para segmentos dos dados gerados de maneira não estruturada, tais como textos, fotos, vídeos, áudios etc. A sigla NoSQL significa *Not Only SQL*.

Bancos de dados relacionais apresentam uma estrutura de informações bem definidas como tabela contendo informações de um cliente. A sigla que representa esse tipo de banco de dados é SQL que significa (*Structured Query Language*).

SQL é uma linguagem de consulta estruturada de dados com ela podemos criar banco de dados, fazer consultas, inserir, atualizar, alterar e remover componentes de dados daqui para frente chamados de registros. Ela se baseia numa lógica relacional ou modelo relacional, sendo assim, a SQL é uma linguagem para trabalhar com banco de dados relacionais onde, a partir de um conjunto de dados estruturalmente organizados podemos acessar, gerenciar e atualizar os dados.

O Conceito de banco de dados relacional foi criado por Edgar Frank Codd em 1970 e publicado no artigo “*Relational Model of Data for Large Shared Data Banks*”, onde os dados são guardados em tabelas que por sua vez podem ou não estar relacionadas, o artigo pode ser acessado no endereço eletrônico: <https://dl.acm.org/doi/pdf/10.1145/362384.362685>. Nesse Modelo os dados são armazenados em tabelas que por sua vez podem ou não estar relacionadas.

Tipo_De_Produto		Produto	
CodTipoProd	DescrTipoProd	CodProd	DescrProd
1	Computador	1	Pc Desktop modelo AZ
2	Impressora	2	Impressora Lazer ABC

Tabela de Banco de dados Relacional - Adaptado de Heuser (2019)

Os bancos de dados relacionais são gerenciados por um RDBMS (*relational database management system*). Nesse tipo de sistema os dados são armazenados em um formato de tabela e o relacionamento entre elas é gerenciado pelo RDBMS.

As vantagens desses sistemas estão no uso de ferramentas conhecidas como SGDBR's ou Sistemas Gerenciadores de Bancos de Dados que oferecem aos usuários processos de validação, verificação, garantias de integridade dos dados, controle de concorrência, recuperação de falhas e segurança, controle de transações, otimização de consultas etc.

Alguns dos problemas estão na dificuldade em conciliar o modelo relacional com a demanda por escalabilidade, na dificuldade em organizar dados em um sistema distribuído quando trabalhamos com particionamento de dados.

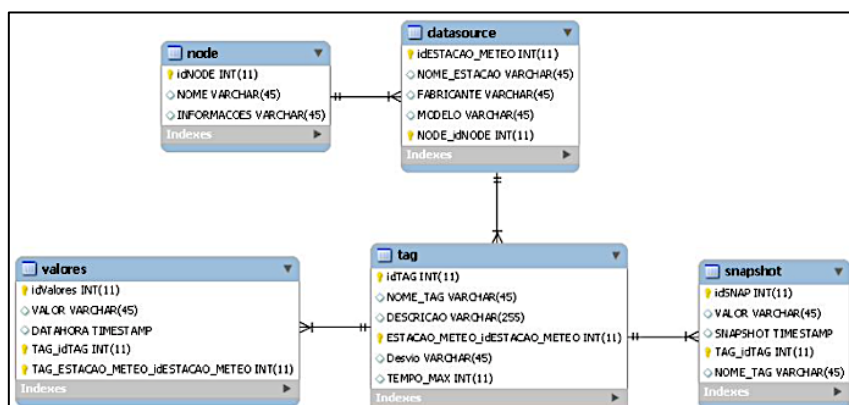


Tabela de Banco de dados Relacional – Fonte: o autor com bases exemplos da internet

Há diversos sistemas gerenciadores de bancos de dados relacionais e todos eles trabalham com a linguagem SQL em *back-ground*:



Fonte: exemplos da internet

Projeto de Banco de Dados: A partir do paradigma da programação orientada da a objetos diversas metodologias de análise de sistemas foram surgindo. Com isso, o trabalho de projetar banco de dados assumiu o papel de mixar um projeto orientado a objetos com um banco de dados relacional.

Isso ocorreu devido a ausência absoluta de projetos comerciais que utilizavam bancos de dados orientados a objetos que fossem confiáveis a um grande volume de dados. Nesse cenário surge uma nova técnica de projetar banco de dados, que não é a formatação pura de classes de dados, mas uma interação entre o ambiente de análise orientada a objetos e a modelagem de dados algo imprescindível para a administração de dados de uma organização.

Segundo F. V. C. (2020), o modelo de classes de objetos não se limita às informações e aos dados, sendo mais amplo quando integra as informações e os seus métodos de acesso, recuperação, processamento e outros em um único objeto. Para utilizá-lo devemos encontrar e separar os dados de seus processos, para obter domínio do negócio através da aplicação.

Por isso, é comum ter aplicações desenvolvidas com Análise Orientada a Objetos (AOO) mesclada com a tradicional modelagem de dados de modelos de

dados, que não são obtidos tão simplesmente quando da pura criação de diagramas de classes. Esse tipo de desenvolvimento é conhecido com *two fase commit* e envolve o diagrama de classes e modelo de dados Entidade-Relacionamento. Para aplicá-lo é preciso entender e dominar as técnicas de modelagem de dados e depois, aprender a lidar com sua equivalência em um modelo de classes de objetos.

Na atualidade tanto o Administrador de Dados (DA) quanto ao Administrador de Bancos de Dados (DBA) precisam entender e saber aplicar técnicas integradas à análise orientada a objetos, de modo a permitir que quem trabalha tanto com Orientação a Objetos quanto com análise estruturada tenha domínio de projetos de bancos de dados eficientes em qualquer ambiente de desenvolvimento com a garantia de coerência com os objetos de negócios de uma organização.

Modelagem de dados: Segundo Machado (2012 / 2016), a modelagem de dados minera as informações que representam um contexto, estruturando-as em um conjunto denominado modelo lógico de dados fornecendo diferentes níveis de abstração de dados.

O modelo de dados é um conjunto de conceitos que podem ser utilizados para descrever as estruturas lógicas e físicas de um banco de dados, servindo como parte importante da documentação dos sistemas que auxiliam os desenvolvedores e como documentação a ser entregue ao cliente que contratou o serviço.

É importante salientar, que um modelo de dados **não informa** quais dados estão armazenados em um banco de dados, **mas sim**, quais e que tipos de dados ele contém.

Modelos de banco de dados: é uma descrição dos tipos de informações que estão armazenadas em um banco de dados (HEUSER, 2019). Os primeiros modelos de bancos surgiram na década de 1960, junto com a ideia de criar modelos que organizem os dados de uma forma mais próxima do modo como são vistos e manipulados pelas pessoas no mundo real.

Aos definirmos abstrações que facilitem a compreensão da organização dos dados o processo se torna mais transparente e independente. Nesse contexto e corroborando com F.V.C (2020, página 30) podemos afirmar que “O objetivo de um modelo de dados é ter certeza de que todos os objetos de dados existentes em determinado contexto e requeridos pela aplicação e pelo banco de dados estão completamente representados e com precisão”.

É preciso levar em consideração se o nível de detalhamento do modelo de dados retrata com fidelidade o banco de dados físico, pois, toda a informação que está no modelo de dados lógico será utilizada para definir as tabelas, chaves primárias, estrangeiras, *stored procedures* e *triggers*.

Um banco de dados mal projetado requer mais tempo e retrabalho e trás resultados incorretos ou incompatíveis para a aplicação.

Abstração: é um processo mental, que usamos quando selecionamos as características e propriedades mais importantes de um conjunto de objetos ou fatos dentro de um contexto sob observação ou análise. Por exemplo: O analista, ao desenvolver a modelagem conceitual dos dados, deve se concentrar na observação dos objetos relevantes que existem em uma realidade ou contexto F.V.C (2020).

Ao abstrair não devemos pensar no momento de automatizar ou processar a informação da realidade estudada. Dessa maneira, podemos dizer que abstração é a visão, sem conceituações técnicas, que obtemos na mente a partir de uma realidade qualquer.



Figura bicicleta. Fonte F.V.C, 2020, página 34.

Ao observar a figura acima, “temos o resultado de um processo de abstração em que excluimos detalhes da estrutura de uma bicicleta, como os pedais, os freios, os mecanismos de tração e, inclusive, as possíveis diferenças entre os muitos modelos de bicicletas, e definimos o objeto como o que conceituamos e denominamos bicicleta” (F.V.C, 2020, página 34) e que para nós será um modelo conceitual.

Devemos sempre buscar no contexto da realidade do ambiente a abstração total do objeto para somente depois desenvolvermos a modelagem lógica. Por isso, o analista deve detalhar todos os elementos geradores de cada objeto, as leis que os regem, as operações que incidem sobre o objeto. **Atenção:** os documentos que registram esses objetos ou elementos de negócio só devem ser utilizados como apoio ao entendimento, e não como base para o desenvolvimento.

A preocupação nesta fase deve ser a de retratar as necessidades de informação que as pessoas precisam para alcançar os objetivos em seu ambiente de negócios. E para registrarmos essas necessidades precisamos fazer uso de um modelo de dados que nos mostre as informações existentes e como elas estão relacionadas às regras de negócio (F.V.C, 2020)

A partir desse modelo criado, será o momento de interagir com os usuários, validando seus objetivos e metas, permitindo o detalhamento do que

denominamos base de dados para construção de um sistema de informações aderente às necessidades dos usuários finais.

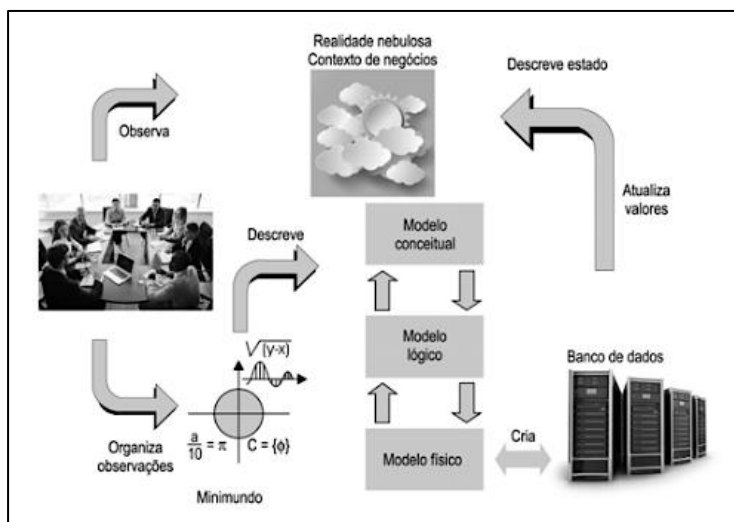


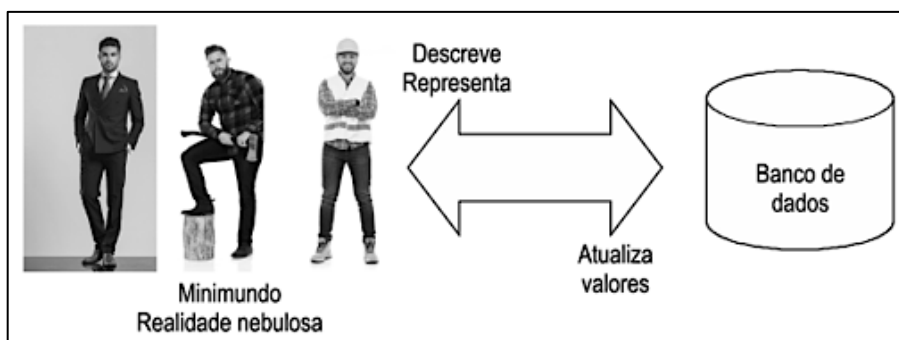
Figura: processo da realidade sendo estudada. Fonte: Mac, 2020, página 36

Os principais macro elementos construídos com a aplicação de técnicas de abstração são:

Minimundo: “é uma porção da realidade, captada pelo analista, que a gestão de negócios de uma organização tem interesse em observar, controlar e administrar. À complexidade existente no momento de analisar um minimundo pode levar o analista a subdividi-lo em partes menores, às quais damos o nome de visão de processo de negócio” (MACHADO, 2012 / 2016, página 36)

Banco de dados: Banco de dados pode ser definido como um conjunto de dados devidamente relacionados. Possuem as seguintes **características**:

- É uma coleção lógica coerente de dados com significado;
- É projetado, construído e preenchido com valores de dados para um propósito específico;
- Possui um conjunto predefinido de usuários e de aplicações.
- Representa algum aspecto do mundo real, o qual é chamado de minimundo; qualquer alteração efetuada no minimundo é automaticamente refletida no banco de dados.



Fonte: F. V. C., 2020, página 37

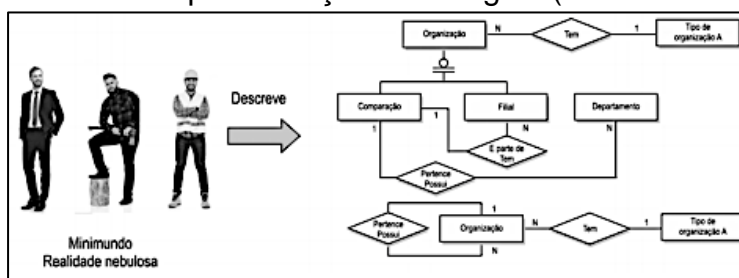
Normalmente um banco de dados bem projetado possui as seguintes propriedades:

- **Campo:** é a menor unidade de informação em um banco de dados;
- **Tipos de dados:** é o tipo de informação que o campo irá receber (int);
- **Registro:** é cada linha da tabela ou uma coleção de campos;
- **Tabela:** unidade de armazenamento do registro.

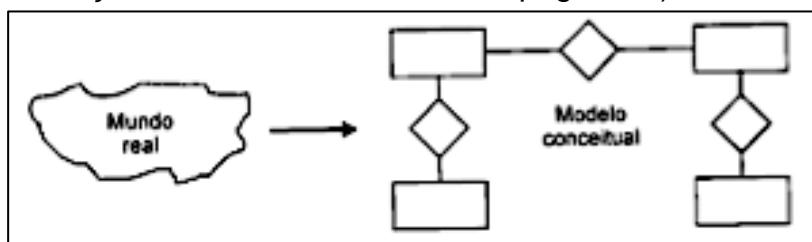
Produto	
CodProd	DescrProd
1	Pc Desktop modelo AZ
2	Impressora Lazer ABC

Fontes: (MACHADO, 2012 e HEUSER, 2019)

Modelo Conceitual: representa e descreve a realidade do ambiente do problema, constituindo-se em uma visão global dos principais dados e seus relacionamentos (estruturas de informação), completamente independente dos aspectos de sua implementação tecnológica (F. V. C. 2020, página: 37).



Modelo Conceitual: representa e/ou descreve a realidade do ambiente do problema, constituindo-se em uma visão global dos principais dados e relacionamentos (estruturas de informação), independente das restrições de implementação MACHADO, 2012 / 2016, página 48).

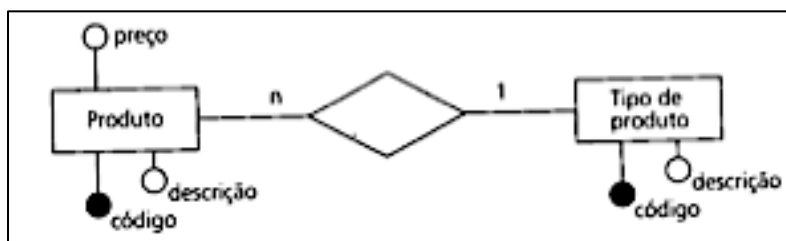


Exemplo de modelo conceitual. Fonte 2012, página 46

Segundo o autor, o objetivo do modelo conceitual é descrever as informações de uma realidade, as quais ficam armazenadas em um banco. É uma descrição de alto nível (macrodefinição), mas que tem a preocupação de captar e retratar toda a realidade de uma organização, setor, repartição, departamento etc.

Modelo Conceitual: é uma descrição do banco de dados de forma independente da implementação de um SGBD. O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD (HEUSER, 2019, página: 25).

Segundo o autor, a técnica de modelagem mais difundida é a abordagem entidade-relacionamento (ER). Nela o modelo conceitual é usualmente representado através de um diagrama, chamado de diagrama entidade relacionamento (DER).



Exemplo de modelo conceitual. Fonte: HEUSER, 2019, página: 26

Quando falamos em modelo conceitual, estamos nos referindo à primeira etapa de um projeto de banco de dados e que normalmente por é realizada por um analista de sistemas que tem por objetivo descrever as informações de um contexto de negócios em relatórios de requisitos do sistema e a partir desse relatório o **modelo conceitual** será desenvolvido no formato textual ou de diagramas de Entidade e Relacionamento e/ou Diagrama de Classes.

Exemplo: Em um sistema que está sendo desenvolvido existem apenas duas entidades (tabelas): Clientes e Pedidos, definidas pelo analista de sistemas que determinou quais os dados que serão necessários para cada entidade(tabela).

Exemplo de um modelo conceito textual:

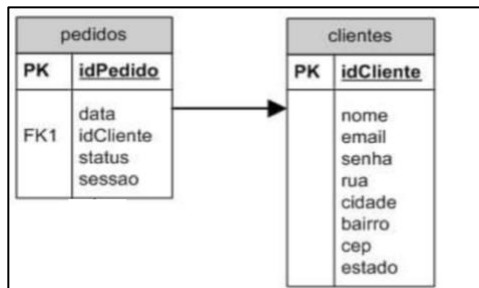
1. Clientes

Dados necessários: nome completo, tipo de pessoa (física ou jurídica), endereço, bairro, cidade, estado, telefone, e-mail, nome de contato.

2. Pedido

Dados necessários: número do pedido, código do produto, quantidade, código do cliente, código do vendedor.

O exemplo apresentado acima não é interessante nos dias de hoje.



Exemplo modelo conceitual: (DER) Diagrama Entidade e relacionamento ou (MER) Modelo de entidade e relacionamento. Fonte: o autor.

É importante estar atento que o modelo conceitual é sempre a primeira etapa de um projeto de banco de dados e que ele não retrata nem é vinculado aos aspectos ligados à abordagem do banco de dados que será utilizado, tampouco se preocupa com as formas de acesso ou estruturas físicas implementadas por um Sistema Gerenciador de Banco de Dados (SGBD) específico, dos dados nele apresentados.

O foco deve ser sempre direcionado ao entendimento e à representação de uma realidade, de um contexto. Sem um modelo conceitual não temos uma visão clara das regras do negócio e acabamos criando aplicações sem entender para que elas foram criadas (F. V. C. 2020)

Mas por onde começar?

Ao analisarmos as diferentes abordagens sobre requisitos Funcionais e Não funcionais de software. Nos requisitos funcionais temos a subdivisão de requisitos de informação, os quais serão levantados e detalhados por meio de modelos de dados. Por isso, nos exercícios sugeridos, vamos modelar os requisitos de informação de um projeto de software.

- As maiores dificuldades nos trabalhos de modelagem de dados encontram-se nos processos iniciais de identificação das entidades.
- O que são entidades em um sistema?
- Como podemos localizar as entidades e identificá-las sem que tenhamos uma margem de erros alta?
- Como conduzir o usuário a mostrar as entidades que tanto precisamos.

Como exemplo vamos analisar um sistema a ser desenvolvido com enfoque de análise de dados para atender a área de vendas de uma empresa.

Como não conhecemos nenhum detalhe da empresa e nem temos experiência com sistemas de vendas, teríamos primeiro a entidade **Vendas**.

O que caracteriza uma venda?

- Ao analisar a entidade abstrata vendas notaremos que ela tem uma diversos atributos, além de possuir outros objetos encapsulados, pois, venda é descrita como um vendedor que tira pedidos produtos para alguém que quer comprá-los.

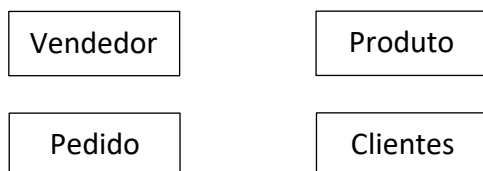
O parágrafo acima desmembra a abstração inicial denominada Venda em três outras entidades: Vendedor, Pedidos e Cliente.

Uma entidade é conceitualmente definida pelos dados que desejamos armazenar para o objeto em si.

Pedido indica a qualifica a existência de alguma quantidade de ocorrências do objeto Pedido; assim como os objetos Cliente e Vendedor. Sendo assim, temos três entidades que participam do contexto Vendas.

Esses objetos são qualificados como entidades porque podemos identificar suas ocorrências individualmente nos embasando para construir tabelas como informações sobre clientes, produtos, pedidos e vendedores da empresa (MACHADO, 2012 / 2016)

Dessa maneira teremos um diagrama ER com as seguintes entidades



- Exercícios – utilizando o draw.io disponível em:

<https://app.diagrams.net>

a) No papel de analista de sistemas desenvolva um relatório de requisitos para um banco de dados para gerenciar uma mercearia. Após usando o aplicativo draw.io desenvolva o diagrama de entidade e relacionamento simples sem se preocupar com os relacionamentos.

b) No papel de analista de sistemas desenvolva um relatório de requisitos para um banco de dados para gerenciar uma rede de trabalhadores na construção civil que realizam trabalhos de freelance, seu cliente é uma cooperativa. Após usando o aplicativo draw.io desenvolva o diagrama de entidade e relacionamento simples sem se preocupar com os relacionamentos.

c) No papel de analista de sistemas desenvolva um relatório de requisitos para um banco de dados para gerenciar uma loja de doces. Após usando o aplicativo draw.io desenvolva o diagrama de entidade e relacionamento simples sem se preocupar com os relacionamentos.

d) No papel de analista de sistemas desenvolva um relatório de requisitos para um banco de dados para gerenciar uma imobiliária. Após usando o aplicativo draw.io desenvolva o diagrama de entidade e relacionamento simples sem se preocupar com os relacionamentos.

f) No papel de analista de sistemas desenvolva um relatório de requisitos para um banco de dados para gerenciar um abrigo para cães. Após usando o aplicativo draw.io desenvolva o diagrama de entidade e relacionamento simples sem se preocupar com os relacionamentos.

g) No papel de analista de sistemas desenvolva um relatório de requisitos para um banco de dados para gerenciar um sistema processo seletivo. Após usando o aplicativo draw.io desenvolva o diagrama de entidade e relacionamento simples sem se preocupar com os relacionamentos.

h) No papel de analista de sistemas desenvolva um relatório de requisitos para um banco de dados para gerenciar um sistema de cursos. Após



usando o aplicativo draw.io desenvolva o diagrama de entidade e relacionamento simples sem se preocupar com os relacionamentos.

Poste os códigos .json e os exercícios resolvidos em um único arquivo no formato nome.zip no ambiente virtual.ifro.edu.br na tarefa “**Poste aqui a atividade avaliativa de 04-10-2024 – vale até 3 pontos até às 22:25**”

Referências

F. V. C. Santos, Rafael. Python: **Guia prático do básico ao avançado** (Cientista de dados Livro 2). 2020. 2ª Edição - Kindle.

Heuser, Carlos Alberto. **Banco de Dados Relacional: Conceitos, SQL e Administração**. 1ª Edição, Editora Porto Alegre, 2019).

Machado, Felipe Nery Rodrigues. **Projeto de banco de dados: uma visão prática**. Edição revisada. São Paulo. Editora Érica, 2012 / 2016. ISBN: 978-85-365-0252-6