Senac

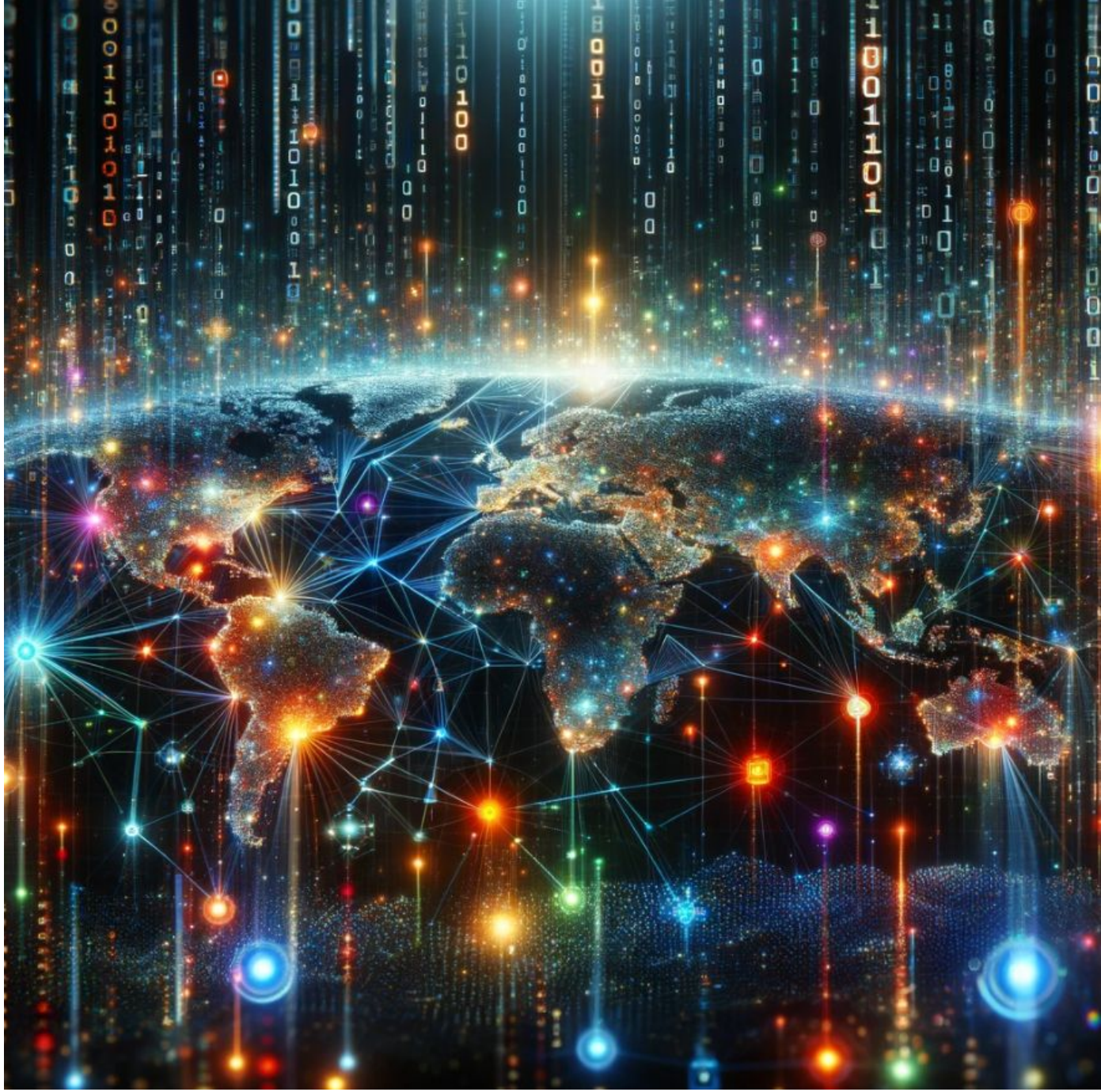
Fecomércio

Sesc

Big Data

Agosto

2025



O conjunto de Dados da aula passada

Big Data

O engenheiro de dados da sua empresa forneceu acesso a dois conjuntos de dados pré-processados. Agora, cabe a você analisá-los cuidadosamente para identificar quais escolas utilizaram seus recursos de forma mais eficiente na preparação de estudantes que se destacaram nas Olimpíadas de Redação e de Matemática.

Big Data

Acessando o Arquivo no drive

```
import pandas as pd
import requests
from io import StringIO

# Criação do dataFrame dos alunos
# ID do arquivo no Google Drive
file_id = '15aOJIGAyLMSY1gecjiCgu2ko_riIcKQy'
file_id2 = '1Jgto7psHaMRTAVzcFt7D6SgJiHMB7uGT'

# URL modificada para forçar o download do arquivo
url = f"https://drive.google.com/uc?id={file_id}"
url2 = f"https://drive.google.com/uc?id={file_id2}"
```

Big Data

```
# Tentando obter o arquivo com requests
try:
    response = requests.get(url)
    response.raise_for_status()  # Lança um erro para respostas não-sucedidas
    # Usando StringIO para converter o texto em um arquivo em memória e, então, lendo com o Pandas
    csv_raw = StringIO(response.text)
    estudantes = pd.read_csv(csv_raw)

    response2 = requests.get(url2)
    response2.raise_for_status()
    csv_raw = StringIO(response.text)
    escolas = pd.read_csv(csv_raw)
except requests.RequestException as e:
    print(f"Erro ao acessar o arquivo: {e}")
```

Big Data

```
escolas
```

	ID_Escola	Nome_Escola	Tipo_Escola	Numero_Alunos	Orcamento_Anual
0	0	Escola A	Publica	2917	1910635
1	1	Escola B	Publica	2949	1884411
2	2	Escola C	Particular	1761	1056600
3	3	Escola D	Publica	4635	3022020
4	4	Escola E	Particular	1468	917500
5	5	Escola F	Particular	2283	1319574
6	6	Escola G	Particular	1858	1081356

Big Data

```
estudantes
```

	ID_Estudante	Nome_Estudante	Genero	Serie	Nome_Escola	Nota_Redacao	Nota_Matematica
0	0	Kevin Bradley	M	6	Escola A	66	79
1	1	Paul Smith	M	9	Escola A	94	61
2	2	John Rodriguez	M	9	Escola A	90	60
3	3	Oliver Scott	M	9	Escola A	67	58
4	4	William Ray	F	6	Escola A	97	84

Big Data

Perguntas:

Existem dados faltantes nas tabelas?

Existe algo em comum nas duas tabelas?

Big Data

Verificação de dados faltantes:

```
escolas.isnull().sum()
```

```
escolas.isna().sum()
```

Repetir o processo para escolas.

Big Data

Combinando os datasets:

```
data = pd.merge(estudantes,
                escolas,
                how='left',
                on=["Nome_Escola", "Nome_Escola"])
```

✓ 0.0s

Big Data

Entendendo as variáveis categóricas:

```
data["Genero"].unique()

data["Serie"].unique()

data["Tipo_Escola"].unique()
```

Big Data

Algumas perguntas:

- Qual o orçamento total das escolas?

- Qual a nota média dos alunos nas disciplinas analisadas?

Big Data

Algumas perguntas:

- Qual o orçamento total das escolas?

```
escolas["Orcamento_Anual"].sum()
```

- Qual a nota média dos alunos nas disciplinas analisadas?

```
mediaRedacao = data["Nota_Redacao"].mean()
mediaMatematica = data["Nota_Matematica"].mean()
```

Big Data

Personalizando os resultados obtidos:

```
# Criando um novo DataFrame com os resultados
resultados = pd.DataFrame({
    "Operação": ["Média Redação", "Média Matemática"],
    "Resultado": [mediaRedacao, mediaMatematica]
})

# Exibindo a tabela
display(resultados)
```

Big Data

Quantos alunos ficaram com nota superior a 90 em redação?

```
highRed = data[data["Nota_Redacao"]>90]
len(highRed)
```

E qual o percentual?

```
len(highRed)/len(estudantes) * 100
```

Big Data

Quantos alunos ficaram com nota superior a 90 em matemática?

```
highMat = data[data["Nota_Matematica"]>90]
len(highMat)
```

Qual o percentual?

```
len(highMat)/len(estudantes) * 100
```

Big Data

Quantos Alunos tiraram nota maior do que 90 nas duas disciplinas?

```
highBoth = data[(data["Nota_Redacao"]>90) & (data["Nota_Matematica"]>90)]
len(highBoth)
```

Qual o percentual?

```
len(data[(data["Nota_Redacao"]>90) & (data["Nota_Matematica"]>90)])/len(estudantes) * 100
```

Big Data

Quantos alunos que obtiveram alto desempenho em ambas as disciplinas são de escolas públicas?

```
highBoth["Tipo_Escola"].value_counts()
```

✓ 0.0s

```
Tipo_Escola
Publica       1321
Particular    1002
Name: count, dtype: int64
```

Big Data

Quantas alunas obtiveram alto desempenho em ambas as disciplinas?

```
highBoth["Genero"].value_counts()
```

✓ 0.0s

```
Genero
M    1167
F    1156
Name: count, dtype: int64
```

Big Data

Como ficou a distribuição dos alunos de alto desempenho pelas série?

```
highBoth["Serie"].value_counts()
```

✓ 0.0s

```
Serie
6    663
8    604
7    599
9    457
Name: count, dtype: int64
```

Big Data

Qual o total por escolas?

```
highBoth["Nome_Escola"].value_counts()
```

✓ 0.0s

Big Data

```
# Agrupando por 'Nome_Escola' e 'Tipo_Escola', e contando o número de registros em cada grupo
sumario = highBoth.groupby(['Nome_Escola', 'Tipo_Escola']).size().sort_values(ascending=False)

# Exibindo o sumário
print(sumario)
```

Big Data

Qual o orçamento per capita de cada escola?

```
perCapita = escolas["Orcamento_Anual"]/escolas["Numero_Alunos"]
escolas["Per_Capita"] = perCapita
```

✓ 0.0s

O cenário atual e as oportunidades

Big Data

Em 2016 a IBM publicou um estudo mostrando que aproximadamente 2,5 quintilhões de bytes (2,5 exabytes) de dados são criados diariamente, e que naquela época 90% dos dados do mundo foram criados nos anos de 2014 e 2015.

Segundo a International Data Corporation (IDC), o fornecimento global de dados atingirá 175 zettabytes (equivalente a 175 trilhões de gigabytes ou 175 bilhões de terabytes) anualmente até 2025.

Big Data

- Um **megabyte** é cerca de um milhão (na verdade, 2^20 de bytes). Arquivos de áudio MP3 de alta qualidade variam de 1 a 2,4 MB por minuto.

- Um **gigabyte** é cerca de 1000 megabytes (na verdade, 2^30 bytes). Equivale a aproximadamente 141 horas de áudio MP3.

- Um **terabyte** é cerca de 1000 gigabytes (na verdade, 2^40 bytes). Equivale a aproximadamente 28 anos de áudio MP3.

- Um **petabyte** é cerca de 1000 terabytes, o que equivale a aproximadamente 141 milhões de horas de áudio MP3.

- Um **exabyte** é cerca de 1000 petabytes, o que equivale a aproximadamente 141 bilhões de horas de áudio MP3.

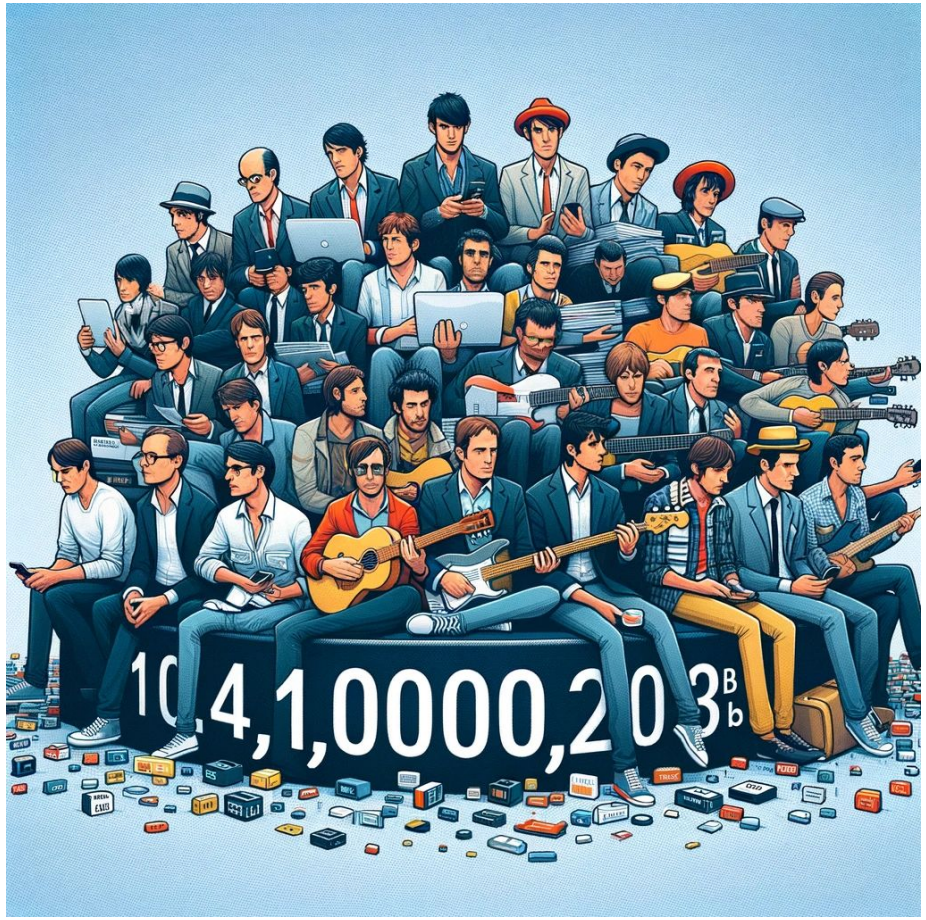
Big Data

Hoje existem mais dispositivos IoT (Internet das coisas), do que aparelhos que não possuem essa tecnologia. O número de dispositivos conectados em Internet das Coisas (IoT) no mundo deve atingir o volume de 41,7 bilhões até o final de 2023.

Observação: Dispositivos IoT são todas e quaisquer tecnologias que possibilitam que os mais diferentes objetos se conectem à internet e interajam com ela de maneira autônoma.

Big Data

Em 2023, estima-se que cerca de 328 milhões de terabytes de dados foram gerados. Em dados mais corretos, são 330 Exabytes de dados diariamente.



Big Data

A explosão de big data provavelmente continuará exponencialmente nos próximos anos. Com 50 bilhões de dispositivos computacionais no horizonte, só podemos imaginar quantos mais haverá nas próximas décadas. É crucial para empresas, governos, militares e até indivíduos conseguirem lidar com todos esses dados.

Big Data

O apelo do big data para o grande empresariado é inegável, dada as realizações que estão acelerando rapidamente. Muitas empresas estão fazendo investimentos significativos e obtendo resultados valiosos. Isso está forçando os concorrentes a investir também, aumentando rapidamente a necessidade de profissionais de computação com experiência em ciência de dados e ciência da computação.

Big Data

Referências:

https://www.networkworld.com/article/966746/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html

https://www.linkedin.com/pulse/o-n%C3%BAmero-de-dispositivos-conectados-em-iot-mundo-deve-atingir/?originalSubdomain=pt

O que é big data?

Big Data

Big Data é um conjunto de dados maior e mais complexo, especialmente de novas fontes de dados. Esses conjuntos de dados são tão volumosos que o software tradicional de processamento de dados simplesmente não consegue gerenciá-los. No entanto, esses grandes volumes de dados podem ser usados para resolver problemas de negócios que você não conseguiria resolver antes.

Análise de Big Data

Big Data

A análise de dados é uma disciplina acadêmica e profissional madura e bem desenvolvida. O termo "análise de dados" foi cunhado em 1962, embora as pessoas já analisassem dados usando estatísticas há milhares de anos, remontando aos antigos egípcios. A análise de big data é um fenômeno mais recente — o termo "big data" foi cunhado por volta de 2000.

Big Data

Os cinco V's do big data:

1. Volume — a quantidade de dados que o mundo está produzindo está crescendo exponencialmente.

2. Velocidade — a rapidez com que esses dados estão sendo produzidos, a velocidade com que se movem pelas organizações e a rapidez com que as alterações de dados estão crescendo rapidamente.

Big Data

3. Variedade — os dados costumavam ser alfanuméricos (ou seja, consistindo de caracteres alfabéticos, dígitos, pontuação e alguns caracteres especiais) — hoje também incluem imagens, áudios, vídeos e dados de um número explosivo de sensores da Internet das Coisas em nossas casas, empresas, veículos, cidades e mais.

Big Data

4. Veracidade — a validade dos dados — eles são completos e precisos? Podemos confiar nesses dados ao tomar decisões cruciais? Eles são reais?

5. Valor – capacidade de extrair insights significativos a partir dos dados. Se refere à capacidade de transformar dados em benefícios concretos. É o processo de identificar partes de dados que são mais úteis e, assim, têm mais valor para ajudar organizações a tomar decisões mais informadas e eficazes.

Big Data

Graças ao avanço tecnológico, especialmente refletido na Lei de Moore, a capacidade de armazenar, processar e transferir esses dados se tornou econômica e eficiente, com capacidades que aumentam exponencialmente. O armazenamento digital evoluiu a ponto de ser possível manter de forma prática e acessível a vasta quantidade de dados que produzimos, fenômeno conhecido como big data.

Infraestruturas de Big Data

Big Data

Vamos discutir as infraestruturas de hardware e software populares para trabalhar com big data e desenvolvimento de aplicações de big data, tanto em desktops quanto baseadas na nuvem.

Big Data

Bancos de dados

- Bancos de dados são infraestruturas críticas para armazenar e manipular grandes volumes de dados que criamos.

- Eles são essenciais para manter esses dados de maneira segura e confidencial, especialmente com leis de privacidade rigorosas, como LGPD no Brasil, HIPAA nos EUA e GDPR na UE.

Big Data

- A maioria dos dados produzidos hoje é não estruturada, como posts do Facebook ou tweets, ou semi-estruturada, como documentos JSON e XML.

- Bancos de dados relacionais não são adequados para dados não estruturados ou semi-estruturados usados em aplicações de big data.

Big Data

- Com a evolução do big data, novos tipos de bancos de dados foram criados para lidar eficientemente com esses dados, incluindo NoSQL e NewSQL.

- Os NewSQL combinam benefícios dos bancos de dados relacionais e NoSQL.

Big Data

Apache Hadoop

- Muitos dos dados atuais são tão grandes que não cabem em um único sistema.

- Com o crescimento do big data, surgiram necessidades de armazenamento de dados distribuídos e capacidades de processamento paralelo para processar os dados mais eficientemente.

Big Data

Apache Hadoop

- Isso levou ao desenvolvimento de tecnologias complexas, como o Apache Hadoop, para processamento de dados distribuídos com paralelismo massivo em clusters de computadores, onde os detalhes intrincados são automaticamente e corretamente gerenciados.

Big Data

Apache Spark

- Apache Spark foi desenvolvido como uma solução para melhorar o desempenho do processamento de big data, executando tarefas em memória, ao contrário do Hadoop, que realiza muitas operações de I/O em disco em vários computadores.

Big Data

Apache Spark

- O Spark streaming é usado para processar dados em fluxo contínuo em mini-lotes. O Spark streaming coleta dados durante um intervalo de tempo especificado e, em seguida, fornece esse lote de dados para processamento.

Big Data

Big Data na Nuvem

Os fornecedores de nuvem focam em tecnologia de arquitetura orientada a serviços (SOA), na qual eles fornecem capacidades "como um Serviço" que as aplicações se conectam e usam na nuvem. Serviços comuns fornecidos por fornecedores de nuvem incluem:

Big data as a Service (BDaaS)	Platform as a Service (PaaS)
Hadoop as a Service (HaaS)	Software as a Service (SaaS)
Hardware as a Service (HaaS)	Storage as a Service (SaaS)
Infrastructure as a Service (IaaS)	Spark as a Service (SaaS)

Big Data

Referências:

https://www.purestorage.com/br/knowledge/big-data/big-data-vs-traditional-data.html

https://www.purestorage.com/br/knowledge/big-data.html

Banco de dados Relacionais no Python

Big Data

Uma **banco de dados** é uma coleção integrada de dados. Um **sistema de gerenciamento de banco de dados** (SGBD) fornece mecanismos para armazenar e organizar dados de maneira consistente com o formato da base de dados.

Os SGBD permitem um acesso conveniente e armazenamento de dados sem preocupação com a representação interna dos banco de dados.

Big Data

Alguns SGBD's open-source, são: **SQLite, PostgreSQL, MariaDB and MySQL.**

Na aula de hoje vamos utilizar o SQLite por sua integração com o python.

Big Data

Vamos acessar a URL do arquivo no Google Drive

url = https://drive.google.com/uc?id=1QrKwpVdrCflRxPJSHhT2sffqIHiqj-mS

Big Data

No python vamos precisar da biblioteca gdown

```
import gdown

# URL do Google Drive convertida para download direto
url = 'https://drive.google.com/uc?id=1kcH4xji_A1_FCgHoDOf6eu9EYcX0blyP'

# Caminho local para salvar o arquivo .db
output_path = 'chinook.db'

# Baixar o arquivo
gdown.download(url, output_path, quiet=False)
```

Big Data

Executando o código anterior será feito o download de um arquivo chamado “chinook.db” que será o banco de dados do primeiro momento da nossa aula.

Para trabalhar com o banco de dados em Python, primeiro use a função `connect` do `sqlite3` para se conectar ao banco de dados e obter um objeto `Connection`:

Big Data

Chinook é um banco de dados de exemplo disponível para SQL Server, Oracle, MySQL, entre outros. Ele pode ser criado executando um único script SQL. O banco de dados Chinook é uma alternativa ao banco de dados Northwind, sendo ideal para demonstrações e testes de ferramentas ORM que visam servidores de banco de dados únicos e múltiplos.

SQLite + Pandas

Big Data

Para trabalhar com o banco de dados em Python, primeiro use a função `connect` do `sqlite3` para se conectar ao banco de dados e obter um objeto `Connection`:

```
import sqlite3

connection = sqlite3.connect('chinook.db')
```

Big Data

O pandas oferece um método eficiente para carregar dados de bancos de dados SQL. Geralmente, utilizamos esse método para executar uma consulta SQL, usando uma conexão já estabelecida com o banco de dados. Assim, para visualizar todas as tabelas do banco podemos utilizar a consulta:

```
import pandas as pd

pd.read_sql_query("SELECT * FROM sqlite_master WHERE type='table'", connection)
```

Big Data

O banco de dados possui onze tabelas. Analisando a tabela de ‘Album’, temos:

```
pd.read_sql("select * from Album", connection)
```

Big Data

Perceba que temos duas colunas de índice. Para resolver isso:

```
pd.read_sql("select * from Album", connection, index_col=["AlbumId"])
```

Big Data

Analisando outra tabela do banco:

```
pd.read_sql("select * from Invoice", connection)
```

Big Data

Vamos criar uma função chamada sq para simplificar o nosso trabalho. Com ela não vamos precisar ficar escrevendo pd.read_sql a todo momento...

```
def sq(str,con=connection):
    return pd.read_sql('''{}'''.format(str), con)
```

Big Data

Frequentemente selecionaremos linhas em um banco de dados que atendem a certos critérios de seleção, especialmente em grandes volumes de dados, onde um banco de dados pode conter muitas linhas. Apenas as linhas que satisfazem os critérios de seleção (formalmente chamados de predicados) são selecionadas. A cláusula WHERE do SQL especifica os critérios de seleção de uma consulta. Valores de string em consultas SQL são delimitados por aspas simples (').

Big Data

```
sq(''' select * from Invoice where total  > '10' ''')
```

Big Data

Complicando um pouco:

```
sq('''select *
from invoice
where total < (select avg(total) from invoice)
''')
```

Big Data

A cláusula WHERE pode conter os operadores <, >, <=, >=, =, <> (diferente) e LIKE. O operador LIKE é usado para correspondência de padrões—procurando por strings que combinam com um padrão dado. Um padrão que contém o caractere curinga de porcentagem (%) procura por strings que tenham zero ou mais caracteres na posição do caractere de porcentagem no padrão. Por exemplo, vamos localizar todos os artistas cujo nome começa com a letra D:

Big Data

```
sq('''select * from Artist where name like 'D%' ''')
```

Big Data

A cláusula ORDER BY ordena os resultados de uma consulta em ordem ascendente (do menor para o maior) ou descendente (do maior para o menor), especificados com ASC e DESC, respectivamente. A ordem de classificação padrão é ascendente, portanto, ASC é opcional. Vamos ordenar os títulos dos álbuns em ordem ascendente:

Big Data

```
sq('''select * from Album order by title asc''')
```

Big Data

Você pode mesclar dados de várias tabelas, o que é referido como juntar as tabelas, com o JOIN.

```
sq(''' SELECT * FROM album
JOIN artist ON artist.artistid = album.artistid ''')
```

Big Data

Para evitar a duplicação de colunas no seu resultado, você deve especificar explicitamente quais colunas deseja retornar na sua consulta e pode usar aliases para renomeá-las conforme necessário:

```
sq(''' select album.albumid,
       album.title,
       artist.artistid AS artist_id,
       artist.name
from album
JOIN artist ON artist.artistid = album.artistid ''')
```

Big Data

Exercício 1: Qual funcionário tem o maior número total de clientes?

```
sq(''' SELECT
        e.FirstName || ' ' || e.LastName AS Employee,
        COUNT(c.customerid) AS Total_Customer
FROM Employee AS e
INNER JOIN Customer AS c
ON e.EmployeeId = c.SupportRepId
GROUP BY 1
ORDER BY 1 DESC''')
```

ETL

Big Data

Extrair, Transformar e Carregar (ETL) é um processo utilizado por organizações orientadas a dados para coletar informações de diversas fontes, processá-las e reuní-las em um formato útil. Este processo suporta atividades como descoberta de dados, geração de relatórios, análise e tomada de decisões.

Big Data

As fontes de dados variam em tipo, formato, volume e confiabilidade, necessitando de adequação para uso efetivo. Os destinos desses dados processados incluem bancos de dados, data warehouses ou data lakes, variando conforme as necessidades e implementações técnicas específicas.

Big Data

Segundo a Oracle:

Extrair

Durante a extração, o ETL identifica os dados e os copia de suas origens, de forma que possa transportar os dados para o armazenamento de dados de destino. Os dados podem vir de fontes estruturadas e não estruturadas, incluindo documentos, emails, aplicações de negócios, bancos de dados, equipamentos, sensores, terceiros e muito mais.

Big Data

Transformar

Como os dados extraídos são brutos em sua forma original, eles precisam ser mapeados e transformados para prepará-los para o armazenamento de dados eventual. No processo de transformação, o ETL valida, autentica, desduplica e/ou agrega os dados de formas que tornam os dados resultantes confiáveis e consultáveis.

Big Data

Carregar

O ETL move os dados transformados para o armazenamento de dados de destino. Esta etapa pode implicar o carregamento inicial de todos os dados de origem ou pode ser o carregamento de alterações incrementais nos dados de origem. Você pode carregar os dados em tempo real ou em lotes programados.

Big Data

Mais detalhes:

https://www.oracle.com/br/integration/what-is-etl/

Nosso Primeiro ETL

Big Data

No endereço http://universities.hipolabs.com/, você encontra uma API que facilita a listagem de universidades por país. Podemos utilizar os dados em formato JSON fornecidos por essa API para criar um banco de dados. Este processo envolve extrair os dados, transformá-los conforme necessário e carregá-los em um sistema de banco de dados adequado.

Big Data


```
[{"web_pages": ["https://www.uniceub.br"], "domains": ["sempreceub.com", "uniceub.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Centro
Universit\u00e1rio de Bras\u00edlia, UNICEUB"}, {"web_pages": ["http://www.baraodemaua.br/"], "domains": ["baraodemaua.br"], "alpha_two_code": "BR", "state-province": null,
"country": "Brazil", "name": "Centro Universit\u00e1rio Barao de Maua"}, {"web_pages": ["http://www.brazcubas.br/"], "domains": ["brazcubas.br"], "alpha_two_code": "BR",
"state-province": null, "country": "Brazil", "name": "Universidade Braz Cubas"}, {"web_pages": ["http://www.candidomendes.br/"], "domains": ["candidomendes.br"],
"alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Universidade Candido Mendes"}, {"web_pages": ["http://www.castelobranco.br/"], "domains":
["castelobranco.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Universidade Castelo Branco"}, {"web_pages":
["http://www.claretiano.edu.br/"], "domains": ["claretiano.edu.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Centro Universit\u00e1rio
Claretiano"}, {"web_pages": ["http://www.creupi.br/"], "domains": ["creupi.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Centro Regional
Universit\u00e1rio de Espir\u00edto Santo do Pinhal"}, {"web_pages": ["http://www.emescam.br/"], "domains": ["emescam.br"], "alpha_two_code": "BR", "state-province": null,
"country": "Brazil", "name": "EMESCAM - Escola Superior de Ci\u00eancias da Santa Casa de Miseric\u00f3rdia de Vit\u00f3ria"}, {"web_pages": ["http://www.epm.br/"], "domains"
["epm.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Universidade Federal de S\u00e3o Paulo"}, {"web_pages": ["http://www.estacio.br/"],
"domains": ["estacio.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Universidade Est\u00e1cio de S\u00e1"}, {"web_pages":
["http://www.faap.br/"], "domains": ["faap.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "FAAP - Funda\u00e7\u00e3o Armando Alvares
Penteado"}, {"web_pages": ["http://www.faculdadescuritiba.br/"], "domains": ["faculdadescuritiba.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil",
"name": "Faculdades Integradas Curitiba"}, {"web_pages": ["http://www.fae.edu/"], "domains": ["fae.edu"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil",
"name": "FAE Business School - Faculdade de Administra\u00e7\u00e3o e Economia"}, {"web_pages": ["http://www.feituverava.com.br/"], "domains": ["feituverava.com.br"],
"alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Funda\u00e7\u00e3o Educacional de Ituverava"}, {"web_pages": ["http://www.fic.br/"], "domains":
["fic.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Faculdade Integradas do Cear\u00e1"}, {"web_pages": ["http://www.fua.br/"], "domains
["fua.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Universidade do Amazonas"}, {"web_pages": ["http://www.furb.rct-sc.br/"], "domains":
["furb.rct-sc.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Universidade Regional de Blumenau"}, {"web_pages": ["http://www.furg.br/"],
"domains": ["furg.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Universidade Federal do Rio Grande"}, {"web_pages":
["http://www.impa.br/"], "domains": ["impa.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name": "Instituto Nacional de Matem\u00e1tica Pura e
Aplicada - IMPA"}, {"web_pages": ["http://www.ime.eb.mil.br"], "domains": ["ime.eb.mil.br"], "alpha_two_code": "BR", "state-province": null, "country": "Brazil", "name":
```

Big Data

Primeiro precisamos acessar o JSON da API

```
import sqlite3
import requests


url = "http://universities.hipolabs.com/search?country=Brazil"

response = requests.get(url)
response.raise_for_status()
universities = response.json()
```

Big Data

Aqui estamos transformando o resultado do get em uma variável que pode ser usada no python

```
universities[0]
```

✓ 0.0s

```
{'domains': ['unochapeco.edu.br'],
 'web_pages': ['https://unochapeco.edu.br'],
 'name': 'Universidade Comunitária da Região de Chapecó - Unochapecó',
 'country': 'Brazil',
 'alpha_two_code': 'BR',
 'state-province': None}
```

Big Data

Vamos nos conectar a um banco de dados SQLite chamado `universities.db` e usar um cursor para executar comandos SQL. Vamos criar uma tabela com várias colunas para armazenar informações sobre universidades, como ID, nome, país, estado/província, páginas da web e domínios.

Big Data

```
conn = sqlite3.connect('universidades.db')

c = conn.cursor()
```

Big Data

```
c.execute(''' CREATE TABLE IF NOT EXISTS universities
              (id INTEGER PRIMARY KEY AUTOINCREMENT,
              name TEXT,
              country TEXT,
              state_province TEXT,
              web_pages TEXT,
              domains TEXT);
              ''')
```

Big Data

Vamos iterar sobre uma lista de universidades, onde cada universidade é representada como um dicionário. Para cada universidade, o código executa uma instrução SQL INSERT para adicionar uma nova linha à tabela. Os valores inseridos são o nome da universidade, o país, o estado ou província, as páginas web e os domínios associados.

Big Data

```
for university in universities:
    c.execute(''' INSERT INTO universities (name, country, state_province, web_pages, domains) VALUES (?,?,?,?,?);''',
              (university.get('name'),
              university.get('country'),
              university.get('state-province'),
              ', '.join(university.get('web_pages', [])),
              ', '.join(university.get('domains', []))
              ))
```

Big Data

Os dados precisam ser salvos no banco que criamos

```
conn.commit()
conn.close()
```

Big Data

Conectando ao novo banco de dados:

```
connection = sqlite3.connect('universities.db')
```

Big Data

Verificando as tabelas que foram geradas:

```
pd.read_sql("SELECT * FROM sqlite_master WHERE type='table'", connection)
```

Big Data

Acessando o banco das universidades:

```
pd.read_sql("select * from universities", connection)
```

Big Data

Procurando todas as universidades de Pernambuco.

```
pd.read_sql("select * from universities where name like '%Pernambuco%' ", connection)
```

Dúvidas?

Marco Mialaret, MSc

Telefone:
81 98160 7018

E-mail:
marcomialaret@gmail.com