



Dos Dados ao Lakehouse

*Blue Lake - Nelson Lakes National Park - Nova Zelândia
(O lago com a água mais clara do mundo)*

Quem sou?

- Alex Souza
 - Mestre em IA
 - Analista de Dados
 - Professor de Pós-Graduação



Dicas iniciais...

- Inglês
- Estudioso
- Curioso
- Resolvedor de problemas
- Monte um Portfólio
- Não seja um Pato



Roteiro

- Introdução
- Gerenciamento de Dados
 - Bancos de Dados
 - Data Warehouses
 - Data Lakes
- Lakehouses
- Montando um Data Mart dentro de um Lakehouse



Introdução

- Grande volume de dados
- Múltiplas fontes de dados
- Centralização
- Disponibilização
 - Engenheiros, Analistas, Cientistas de dados



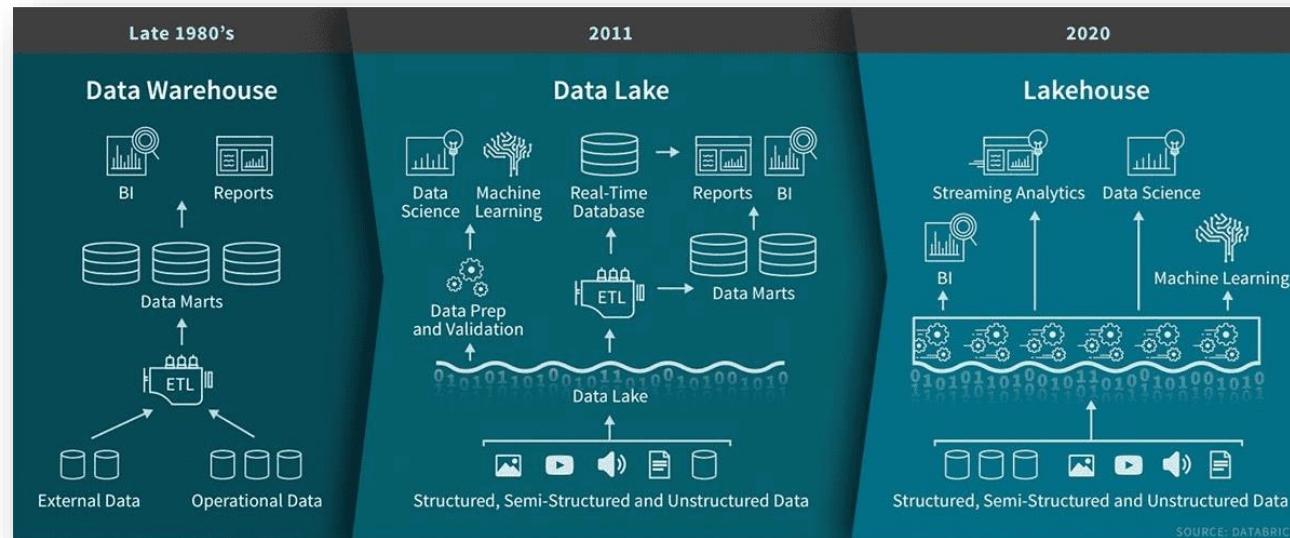
Introdução

- Historicamente, muitas soluções foram criadas e utilizadas para atender a essas necessidades
 - Banco de Dados
 - Data Warehouse
 - Datalake
- Limitações
- Nuvem
- Lakehouse



Gerenciamento de Dados

- O gerenciamento de dados vem mudando com o passar dos anos conforme a **necessidade de acesso mais rápido** aos dados vem crescendo, dados estes **estruturados, semiestruturados e não estruturados**, com grande volumetria e múltiplas origens.



Banco de Dados

- Bancos de Dados relacionais
 - Coletar, armazenar e analisar dados
 - Forma simples e confiável
 - Baixa quantidade de dados
 - Internet
 - Aumento volume de dados | vários bancos de dados
 - Silos de dados
 - integração



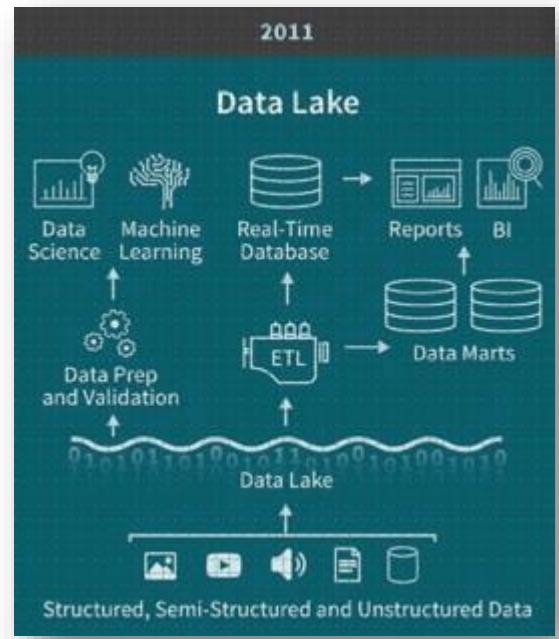
Data Warehouses

- Armazenar diversas atividades (diversos bd's)
 - Centralização e Apoio a tomada de decisão
- Big Data
 - Novas necessidades
 - Consultar Dados não estruturados
 - Tempo real, *Machine Learning*
- Formato proprietário (dependência de fornecedor)
- Custo



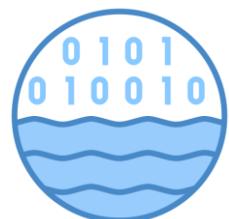
Data Lakes

- Análise à variados e grandes volumes de dados à um baixo custo (processamento | armazenamento)
 - Nuvem (flexibilidade, praticidade e custo)
- Apache Hadoop | Apache Spark
 - Cluster de computadores
 - Mecanismo de processamento e análise unificado



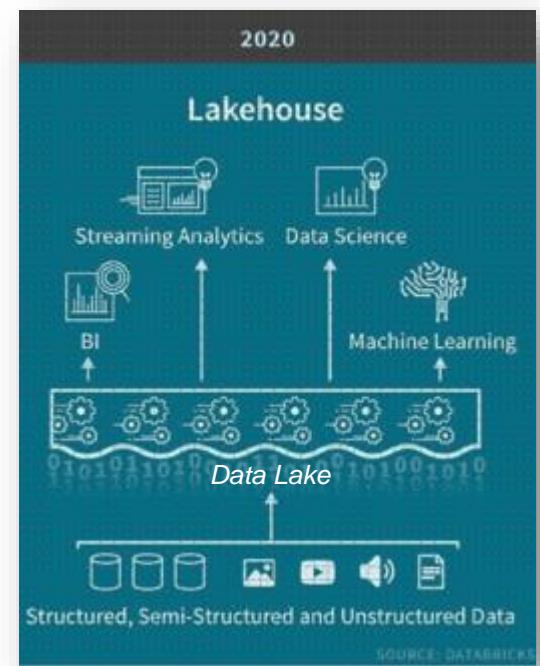
Data Lakes

- Limitações
 - Arquitetura complexa para usuários
 - Não suportam transações ACID
 - Falta de consistência e isolamento torna-o quase impossível para **mesclar inserções** e **consultas** de dados, assim como processamentos em Lote (*batch*) e *streaming*.
 - Não tem um controle de versão
 - Não impõem qualidade de dados

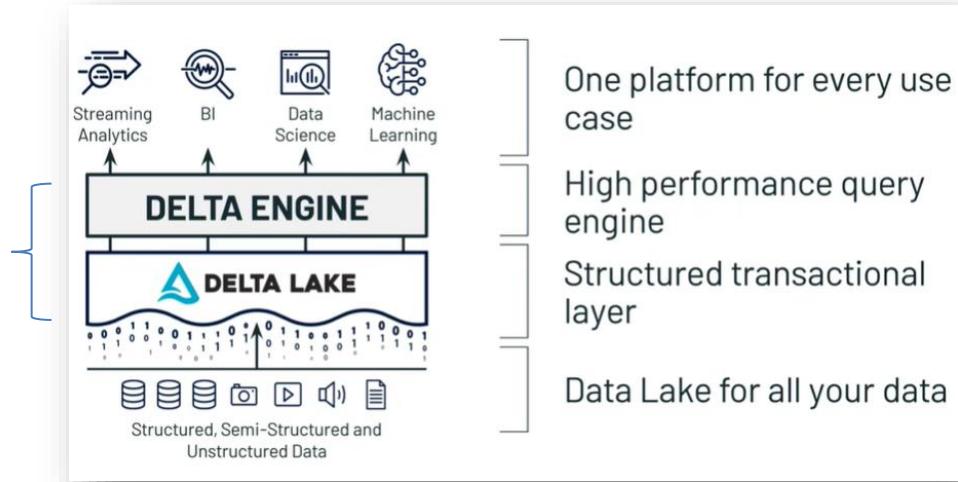
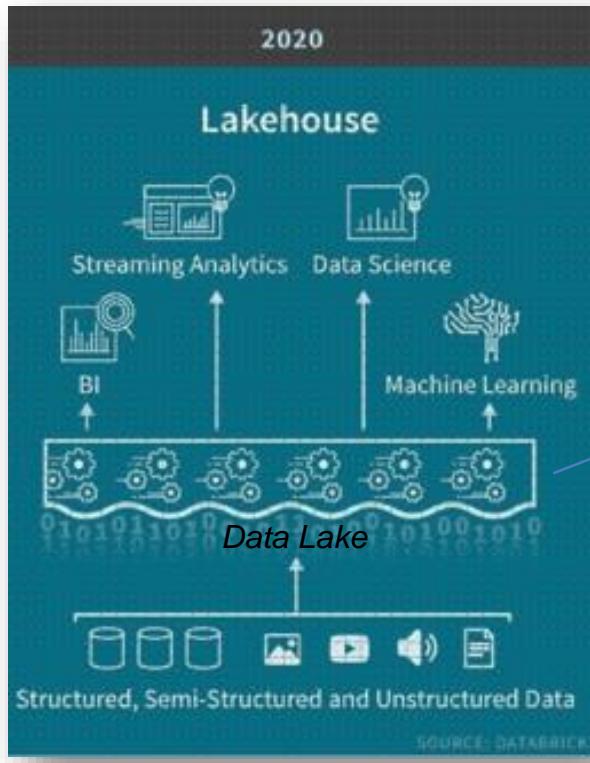


Lakehouses

- Um *Lakehouse* é uma novo **conceito** que permite aos usuários fazer tudo, desde **BI**, **análise SQL**, **ciência de dados** e **ML** em uma única plataforma.
- O *Lakehouse* tem uma abordagem opinativa para construir **Data Lakes**, adicionando atributos de **Data Warehouses** - confiabilidade, desempenho e qualidade, mantendo a abertura e **escala** de **Data Lakes**.



Lakehouses (Camadas)



One platform for every use case

High performance query engine

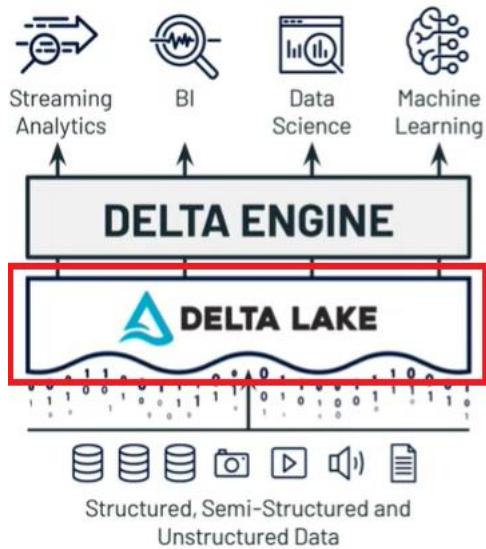
Structured transactional layer

Data Lake for all your data



databricks

Lakehouses (*Delta Lake*)



- Transações ACID
- Validação de esquema
- Indexação
- Desempenho de SQL



databricks

Lakehouses (*Delta Engine*)



- Transações ACID
- Validação de esquema
- Indexação
- Desempenho de SQL



databricks

Lakehouses (Vantagens)

› Vantagens e diferenciais

- › Dados estruturados, semiestruturados e não estruturados
- › Alta escalabilidade e baixo custo (Nuvem)
- › Desempenho de SQL
- › Suporte a BI, como os DW, só que também dá suporte a ML
- › Suporta controle de versão dos dados



Lakehouses (Desafios superados)

- Desafios que podem ser superados...
 - Unificação da equipe de dados
 - Silos de dados
 - Dados obsoletos
 - Aprisionamento à fornecedores

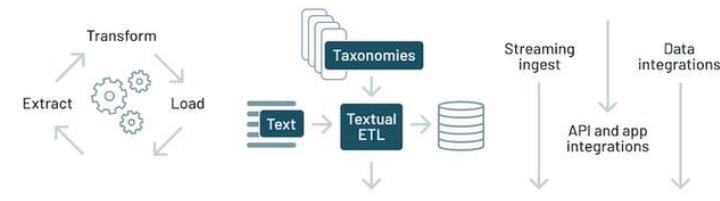


Data Lakehouse

Structured

Textual

Other unstructured



Raw data in open file formats

Curated data with
governance

Open API's with direct file access using SQL, R, Python and other languages



Lakehouses (Ferramentas)



Lakehouses (Comparativo de Funcionalidades)

Feature Comparison ~ Hudi, Iceberg & Delta



Features for Apache Spark 3.0	Delta Lake	Apache Iceberg	Apache Hudi
Transaction	Y	Y	Y
MVCC	Y	Y	Y
Time Travel	Y	Y	Y
Schema Evolution	Y	Y	Y
Data Mutation	Y (Insert/Update/Delete/Merge)	Y (Insert/Delete/Merge)	Y (Upsert/Insert/Bulk Insert)
Streaming	Source & Sink for Structured Streaming	Writes using Structured Streaming	DeltaStreamer & HiveIncrementsPuller
File Format	Parquet	Parquet, ORC & AVRO	Parquet
Compaction Cleanup	Manual	Manual	Manual & Auto
Language Support	Scala/Java/Python	Java/Python	Java/Python
Storage Abstraction	Y	Y	N
API Dependency	Apache Spark	Native & Apache Spark	Apache Spark
Data Ingestion	Spark, Presto, Hive	Spark, Hive & Trino	DeltaStreamer

Lakehouses (demonstração)



Lakehouses (case)

CASE

A Z-Educa é uma empresa da área de educação que trabalha com venda de sistemas de ensino para escolas da educação básica em todo território nacional.

- Problema
 - Descentralização de dados
 - Servidor de Banco de dados (SQL Server) do ERP da empresa está aloorado em Datacenter (em uma sala exclusiva e monitoramento de temperatura 24hs por dia);
 - Servidor de Backup aloorado em Datacenter (em uma sala exclusiva e monitoramento de temperatura 24hs por dia);
 - Servidor de Data Warehouse (SQL Server) aloorado em Datacenter;
 - Custo
 - Servidor de Data Warehouse;
 - Licença do SQL Server Standard do Servidor de Data Warehouse;
 - Necessidade de análises avançadas
 - Atualmente realizadas no computador do Cientista de Dados;
 - dados de fontes públicas - INEP, redes sociais, documentos, imagens e etc...
- Projeto
 - Fase 1 (entregue)
 - Data Lake (grande repositório de dados brutos)
 - Com dados para análises avançadas (cientista de dados)
 - Censo Escolar - INEP (exemplo)
 - Backups
 - Com dados do monitoramento do datacenter (utilizado pela equipe de infraestrutura)
 - Datacenter envia direto para o Data Lake (de forma automática) e não mais por e-mail diariamente
 - Fase 2 (em andamento...)
 - Migração do Data Warehouse
 - começando pelo **Data Mart** de Vendas (iremos detalhar aqui...)

Lakehouses (recursos)

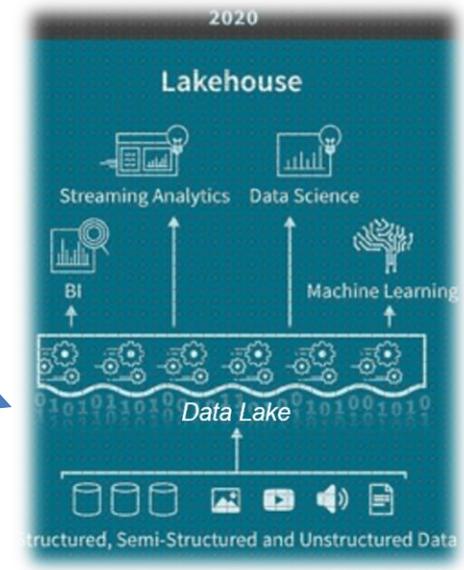
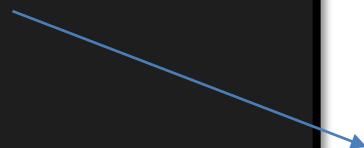
Criando um Lakehouse com o Delta Lake (Databricks)

- Ferramenta - *Databricks community*
- Usaremos *Spark SQL* e **Delta Lake** para ingerir dados e criar as tabelas do ERP para serem nossa única fonte de verdade (*Single Source of Truth - SSOT*).
- A partir daí, usaremos as tabelas *Silver* para construir nosso **Data Mart** de Vendas.
- Também usaremos **Delta Lake** para realizar manutenções nas tabelas (inserts, updates e deletes) e tornar nossos dados mais em "tempo real".
- Possíveis integrações do nosso **Data Mart** com outras fontes de dados.
- Ler os dados pelo Power BI.

Lakehouses (nossa data lake)

Visualizando nosso Data Lake

- no nosso caso, esta no **DBFS - File Store** da própria Databricks, mas normalmente é alocado no AWS S3, Azure, HDFS, Google Cloud...
 - Visão geral do que tem no Data Lake da Z-Educa...
 - Tables
 - Censo 2020 (utilizado para análise de mercado, clusterização de escolas, perfil escolar e etc...)
 - ERP (tabelas replicadas do ERP)
 - Estudos (dados dos sensores do datacenter, arquivos texto, imagens, vídeos que estão sendo utilizados para estudos internos)
 - Saúde (dados de um sistema que monitora dados de saúde dos funcionários Z-Educa)
 - Temp (arquivos temporários)
 - lake
 - o banco de dados que criaremos logo na sequência...
 - user
 - /user/hive/warehouse/censo2020
 - tabela tratada do censo escolar 2020



Lakehouses (nossa data lake)

Data source 

Upload File S3 DBFS Other Data Sources Partner Integrations

Select a file from DBFS 

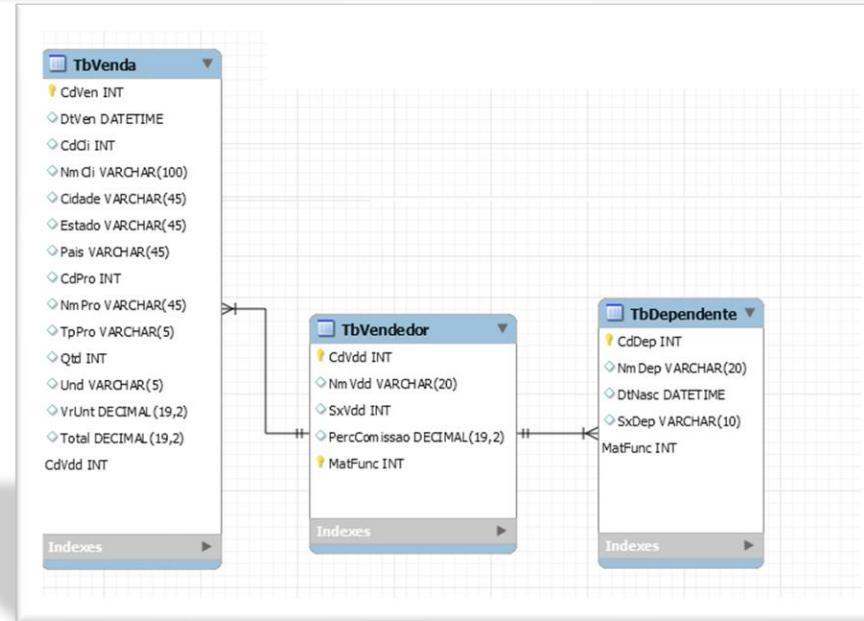
<ul style="list-style-type: none"><input type="checkbox"/> FileStore<input type="checkbox"/> dbacademy<input type="checkbox"/> lake<input type="checkbox"/> tmp<input type="checkbox"/> user	<ul style="list-style-type: none"><input type="checkbox"/> import-stage<input type="checkbox"/> plots<input type="checkbox"/> shared_uploads<input type="checkbox"/> tables	<ul style="list-style-type: none"><input type="checkbox"/> censo2020<input type="checkbox"/> erp<input type="checkbox"/> estudos<input type="checkbox"/> saude<input type="checkbox"/> temp	<ul style="list-style-type: none"> tbdependente.csv tbvendas.csv tbvendedor.csv
--	--	---	--

[/FileStore/tables/erp](#)

Lakehouses (erp)

- No que se diz respeito a Tabelas do ERP (tables/erp)

- contem a replicação de 3 tabelas que são origens para os **Data Marts** de Vendas
- simulamos a replicação dos dados históricos (existem diversas técnicas - CDC do Data Migration Services da Amazon e etc...)
- e replicação de inclusões, alterações e deletes... (simulando o near real time...)



Lakehouses (lake)

Criando o banco de dados (chamaremos de: Lake)

- que usaremos para análises de BI
 - aparecerá uma pasta no nosso Data Lake, mas a estrutura estará vazia...

Cmd 7

```
CREATE DATABASE IF NOT EXISTS lake;
USE lake;
--drop database lake;
```

OK

Command took 0.21 seconds -- by aasouzaconsult@gmail.com at 19/06/2021 08:42:12 on ZouzaCluster

Data source 

Upload File

S3

DBFS

Select a file from DBFS 

 FileStore

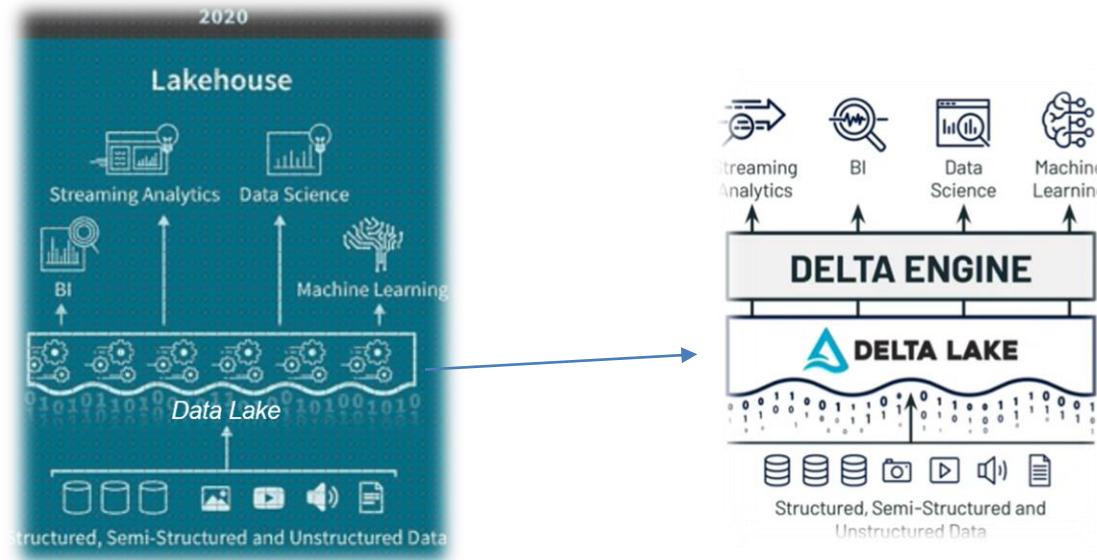
 dbacademy

 lake

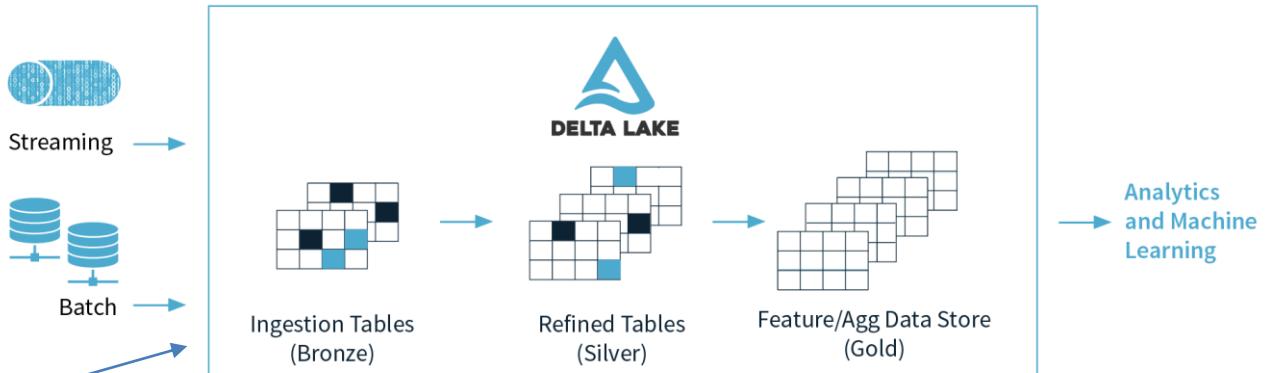
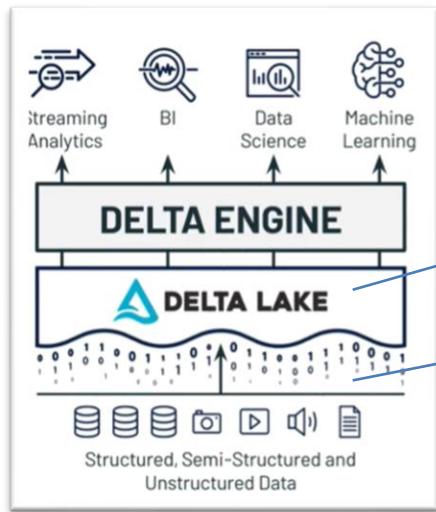
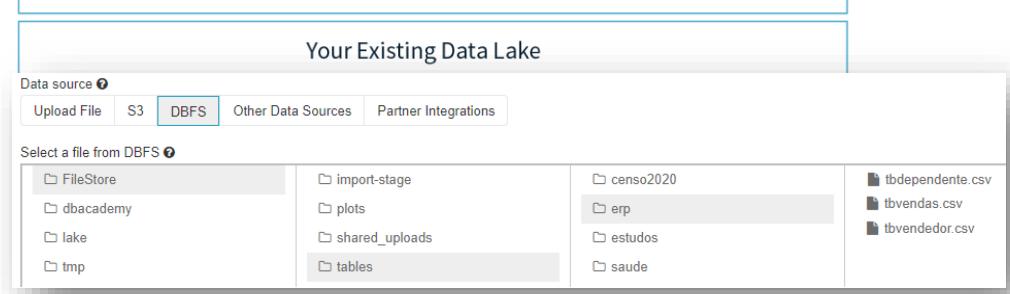
 tmp

 user

Lakehouses (delta lake)



Lakehouses (delta lake - camadas)

The screenshot shows a Databricks interface for "Your Existing Data Lake". The top navigation bar includes "Data source" (Upload File, S3, DBFS, Other Data Sources, Partner Integrations), with "DBFS" selected. Below is a file browser titled "Select a file from DBFS". The left sidebar lists "FileStore", "dbacademy", "lake", and "tmp". The main area shows a grid of files under "import-stage", "erp", "estudos", and "saude". Specific files like "censo2020", "tbdependente.csv", "tbvendas.csv", and "tbvendedor.csv" are visible.

Lakehouses (camada bronze...)

Criando a camada Bronze (Ingestion Tables) - Tabelas de Ingestão

- criando as tabelas de ingestão, conforme arquivos históricos recebidos do ERP (contidos no nosso Data lake)

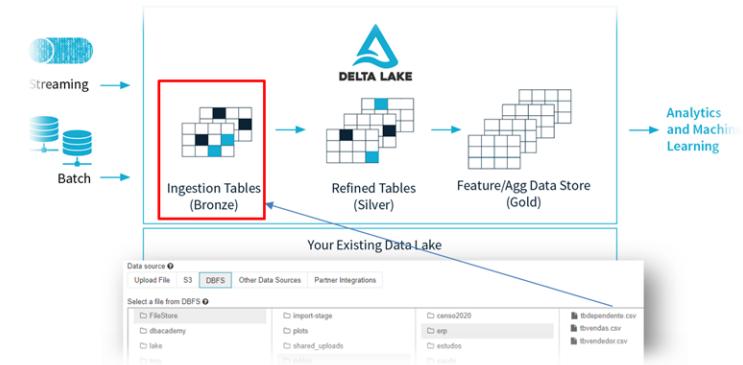
Cmd 9

```
-- Aqui pegaremos informações históricas de 3 tabelas do ERP
DROP TABLE IF EXISTS tbdependente_hist;
DROP TABLE IF EXISTS tbvendas_hist;
DROP TABLE IF EXISTS tbvendedor_hist;

-- Carga Batch
-- Uma carga com dados históricos de dependentes
CREATE TABLE tbdependente_hist
USING csv
OPTIONS (
  PATH "/FileStore/tables/erp/tbdependente.csv", header "true", sep ';'
);

-- Uma carga com dados históricos de vendas
CREATE TABLE tbvendas_hist
USING csv
OPTIONS (
  PATH "/FileStore/tables/erp/tbvendas.csv", header "true", sep ';'
);

-- Uma carga com dados históricos de vendedores
CREATE TABLE tbvendedor_hist
USING csv
OPTIONS (
  PATH "/FileStore/tables/erp/tbvendedor.csv", header "true", sep ';'
);
```



Lakehouses (ex. camada bronze...)

Visualizando as tabelas da camada Bronze

- lendo estas tabelas utilizando SQL :)

Cmd 11

```
-- Visualizando a carga de dependentes
select * from tbdependente_hist;

-- vale lembrar:
-- - nomenclatura dos campos
-- - padronização (campo sexo)
```

▶ (1) Spark Jobs

	CdDep	NmDep	DtNasc	SxDep	MatFunc	delete
1	1	Dependente 1	02/02/2010 00:00	Masc	1	0
2	2	Dependente 2	05/04/2012 00:00	Masc	3	0
3	3	Dependente 3	04/03/2013 00:00	Fem	3	0
4	4	Dependente 4	05/05/2010 00:00	Fem	4	0
5	5	Dependente 5	06/07/2019 00:00	Masc	4	0
6	6	Dependente 6	02/03/2018 00:00	Fem	9	0

Showing all 6 rows.



Command took 1.99 seconds -- by aasouzaconsult@gmail.com at 07/06/2021 08:00:54 on ZouzaCluster



Ingestion Tables
(Bronze)

Lakehouses (ex. camada prata...)

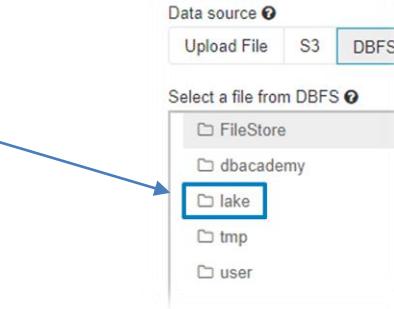
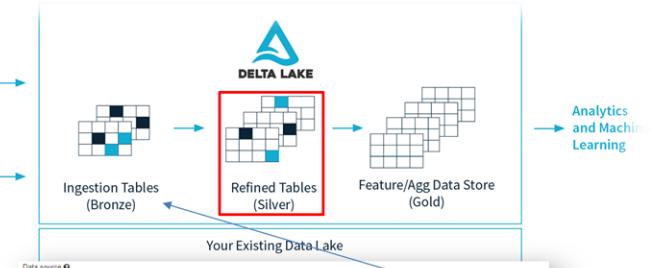
Criando a tabela silver de Vendas

- salvando em formato colunar e otimizado (.PARQUET)
- semanticamente amigável
- conversões de campos numéricos

Cmd 22

```
DROP TABLE IF EXISTS vendas_silver;

CREATE TABLE vendas_silver
USING PARQUET
PARTITIONED BY (Estado)
LOCATION "/lake/vendas/silver"
AS (
    SELECT CdVen as CódigoVenda
        , DtVen as DataVenda
        , CdCli as CódigoCliente
        , NmCli as NomeCliente
        , Cidade
        , Estado
        , País
        , CdPro as CódigoProduto
        , NmPro as NomeProduto
        , TpPro as TipoProduto
        , cast(Qtd as int)           as Quantidade
        , Und
        , cast(VrUnit as decimal(18,2)) as ValorUnitario
        , CdVdd                         as CódigoVendedor
        , delete
    FROM tbvendas_hist
)
```

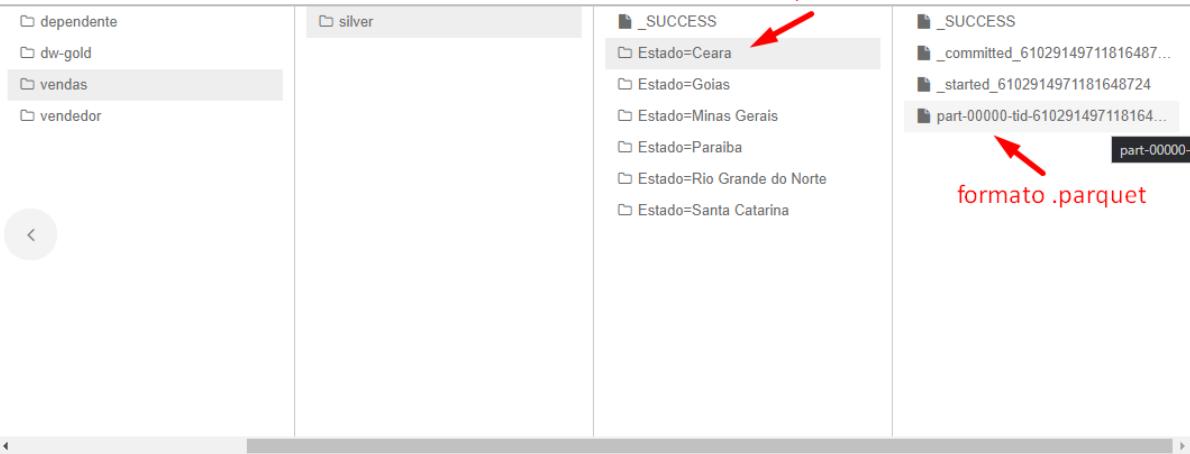


Lakehouses (ex. estrutura .parquet)

Data source 

Upload File S3 DBFS Other Data Sources Partner Integrations

Select a file from DBFS 



particionamento

formato .parquet

caminho

`/Lake/vendas/silver/Estado=Ceara`


 Refined Tables
 (Silver)

Lakehouses (converter para Delta)

Criando uma "Delta Table" a partir de uma tabela PARQUET do Data Lake tradicional

- Primeiro, converteremos os arquivos locais (Parquet) em formato Delta.
- A conversão cria um log de transações Delta Lake que rastreia os arquivos.
- Agora, o diretório é um diretório de arquivos Delta e dá suporte aos recursos do Lakehouse (ACID, Controle de Versão e etc...)

Cmd 35

```
-- Tabela de Dependentes
CONVERT TO DELTA
parquet.`./lake/dependente/silver`
PARTITIONED BY (SexoDependente int)
```

► (10) Spark Jobs

OK

Command took 15.60 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 14:34:01 on ZouzaCluster

Cmd 36

```
-- Tabela de Vendas
CONVERT TO DELTA
parquet.`./lake/vendas/silver`
PARTITIONED BY (Estado string)
```

► (10) Spark Jobs

OK

Command took 12.13 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 14:34:39 on ZouzaCluster

Cmd 37

```
-- Tabela de Vendedor
CONVERT TO DELTA
parquet.`./lake/vendedor/silver`
PARTITIONED BY (SexoVendedor int)
```

► (10) Spark Jobs

OK



- Transações ACID
- Validação de esquema
- **Indexação**
- Desempenho de SQL



Refined Tables
(Silver)

Lakehouses (Registrar para Delta)

Registrar as tabelas no Metastore

- cria as tabelas agora em formato DELTA

Cmd 39

```
-- Etapa 2: Registrar a Tabela Delta
-- A seguir, vamos registrar a tabela no Metastore. O comando Spark SQL inferirá automaticamente o esquema de dados lendo os rodapés dos arquivos Delta.
```

```
-- Dependente
```

```
DROP TABLE IF EXISTS dependente_silver;
```

```
CREATE TABLE dependente_silver
USING DELTA
LOCATION "/lake/dependente/silver"
```

► (4) Spark Jobs

OK

Command took 3.83 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 14:36:36 on ZouzaCluster

Cmd 40

```
-- Vendas
```

```
DROP TABLE IF EXISTS vendas_silver;
```

```
CREATE TABLE vendas_silver
USING DELTA
LOCATION "/lake/vendas/silver"
```

► (4) Spark Jobs

OK

Command took 4.05 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 14:36:42 on ZouzaCluster

Cmd 41

```
-- Vendedor
```

```
DROP TABLE IF EXISTS vendedor_silver;
```

```
CREATE TABLE vendedor_silver
USING DELTA
LOCATION "/lake/vendedor/silver"
```

► (4) Spark Jobs

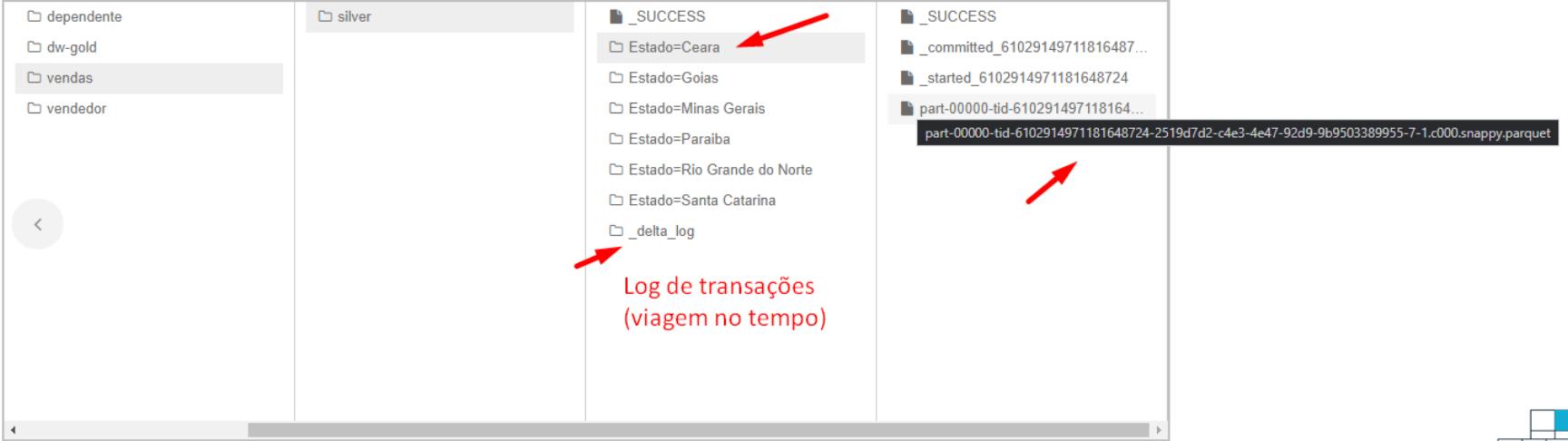


Refined Tables
(Silver)

Lakehouses (ex. estrutura delta)

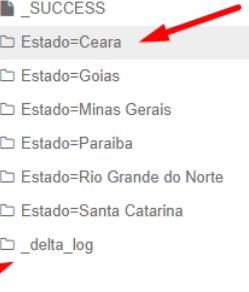
Data source  DBFS Upload File S3 Other Data Sources Partner Integrations

Select a file from DBFS 



`/lake/vendas/silver/Estado=Ceara`

`_SUCCESS`

`Estado=Ceara` 

`Estado=Goias`

`Estado=Minas Gerais`

`Estado=Paraiba`

`Estado=Rio Grande do Norte`

`Estado=Santa Catarina`

`_delta_log`

`_SUCCESS`

`_committed_61029149711816487...`

`_started_6102914971181648724`

`part-00000-tid-6102914971181648724-2519d7d2-c4e3-4e47-92d9-9b9503389955-7-1.c000.snappy.parquet`

Log de transações
(viagem no tempo)



Refined Tables
(Silver)

Lakehouses (ex. select na prata)

```
-- Consultando em formato Delta (Vendas)
select * from vendas_silver
```

▶ (4) Spark Jobs

	CodigoVenda	DataVenda	CodigoCliente	NomeCliente	Cidade	Pais	CodigoProduto	NomeProduto	TipoProduto	Quantidade	Und	ValorUnitario	CodigoVendedor	delete	Estado
1	3	01/02/2010 00:00	5	Cliente BB	Belo Horizonte	Brasil	1	Produto A	A	4200	KG	0.34	4	0	Minas Ge
2	10	08/05/2010 00:00	5	Cliente BB	Belo Horizonte	Brasil	1	Produto A	A	24000	KG	0.34	1	0	Minas Ge
3	2	01/02/2010 00:00	2	Cliente AB	Fortaleza	Brasil	2	Produto B	A	730	KG	2.00	5	0	Ceara
4	4	02/02/2010 00:00	3	Cliente BC	Baturite	Brasil	1	Produto A	A	250	KG	7.00	2	0	Ceara
5	5	03/02/2010 00:00	4	Cliente CC	Fortaleza	Brasil	1	Produto A	A	4500	KG	0.34	1	0	Ceara
6	6	04/03/2010 00:00	5	Cliente CD	Eusebio	Brasil	1	Produto A	A	11200	KG	0.34	3	0	Ceara
7	1	01/02/2010 00:00	1	Cliente AA	Florianopolis	Brasil	1	Produto A	A	4000	KG	0.34	2	0	Santa Ca
8	8	06/03/2010 00:00	7	Cliente DE	Joao Pessoa	Brasil	1	Produto A	A	12000	KG	0.34	2	0	Paraiba
9	7	05/03/2010 00:00	6	Cliente DD	Goiania	Brasil	1	Produto A	A	12500	KG	0.34	2	0	Goias
10	9	07/04/2010 00:00	8	Cliente EE	Natal	Brasil	1	Produto A	A	17500	KG	0.34	3	0	Rio Gran

Showing all 10 rows.

Command took 2.89 seconds -- by aasouzaconsult@gmail.com at 24/06/2021 10:21:00 on ZouzaCluster


 Refined Tables
 (Silver)

Lakehouses (camada ouro...)

Criando a camada Gold (refined) - Análise

- A seguir, criaremos uma tabela Delta - Gold. Faremos isso criando uma tabela agregada a partir dos dados na tabela delta: vendas_silver que acabamos de criar.

```
Cmd 48
%fs
rm -r /lake/dw-gold/totalvendas/

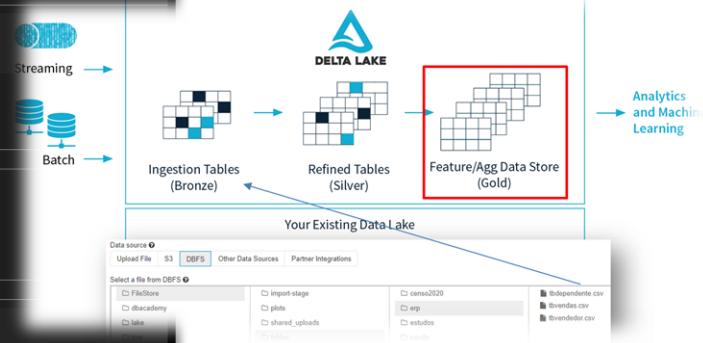
res7: Boolean = true
Command took 1.42 seconds -- by asouzaconsult@gmail.com at 16/06/2021 14:42:00 on ZouzaCluster
Cmd 49
```

Criando uma tabela de agregação

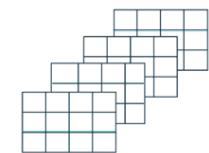
- Valor total de vendas por Estado

```
Cmd 50
-- Criamos a tabela usando o padrão Spark SQL Create Table As Select (CTAS).
-- A subconsulta usada para definir a tabela é normalmente um consulta agregada
-- Agregação Total vendas --
DROP TABLE IF EXISTS totalvendas;

CREATE TABLE totalvendas
USING DELTA
LOCATION '/lake/dw-gold/totalvendas/'
AS (
  SELECT Estado
    , SUM(Quantidade * ValorUnitario) ValorTotal
   FROM vendas_silver
  WHERE delete = 0
 GROUP BY Estado
  )
```

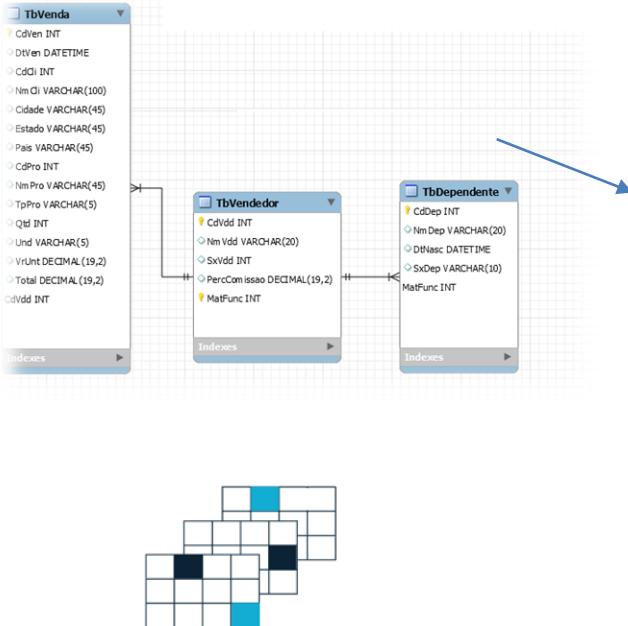


Lakehouses (ex. select na ouro...)

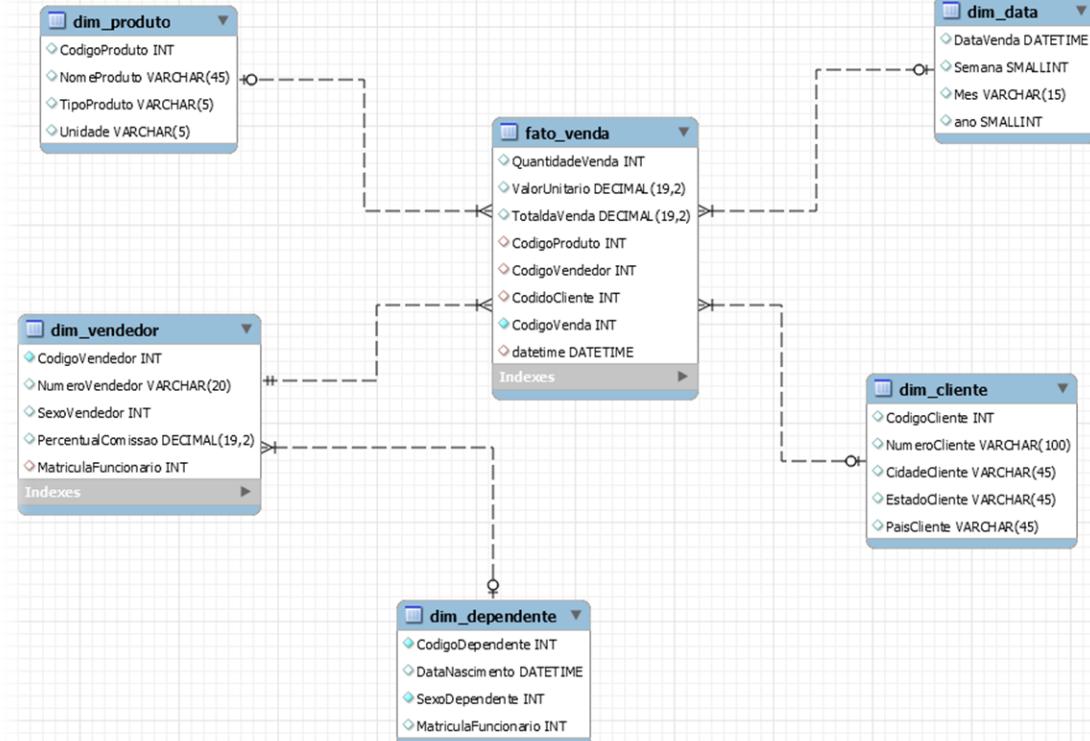


Feature/Agg Data Store
(Gold)

Lakehouses (data mart)



Refined Tables
(Silver)



Lakehouses (dimensões - data mart)

Criando um DW aqui no Lakehouse :)

- Das tabelas Silver (DELTA), já podemos montar nosso **Data Marts** de Vendas, salvando na camada superior gold ou criando *views* (no caso aqui, criamos *views*)

Cmd 53

Criando as Dimensões e Fatos

Cmd 54

```
-----  
-- Dimensão Dependente --  
  
CREATE OR REPLACE VIEW dim_dependente  
AS (  
  SELECT *  
    , '11000082' as INEP_ESCOLA -- todos os filhos dos vendedores estudam na mesma escola (benefício)  
   FROM dependente_silver  
 WHERE delete = 0  
)
```

OK

Command took 0.90 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 15:03:23 on ZouzaCluster

Cmd 55

```
-----  
-- Dimensão Vendedores --  
  
CREATE OR REPLACE VIEW dim_vendedor  
AS (  
  SELECT *  
   FROM vendedor_silver  
 WHERE delete = 0  
)
```

OK

Command took 0.59 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 15:03:28 on ZouzaCluster



Refined Tables
(Silver)

Lakehouses (fato - data mart)

Tabela de Vendas - Silver

- Além do FATO Venda em si, esta tabela contem:
 - informações de clientes (que iremos transformar em uma *dimensão cliente*)
 - informações de produto (que iremos transformar em uma *dimensão produto*)

Cmd 57

```
SELECT * FROM vendas_silver
```

▶ (3) Spark Jobs

	CodigoVenda	DataVenda	CodigoCliente	NomeCliente	Cidade	Pais	CodigoProduto	NomeProduto	TipoProduto	Quantidade	Und	ValorUnitario	CodigoVendedor	delete	Estado
1	3	01/02/2010 00:00	5	Cliente BB	Belo Horizonte	Brasil	1	Produto A	A	4200	KG	0.34	4	0	Minas Gerais
2	10	08/05/2010 00:00	5	Cliente BB	Belo Horizonte	Brasil	1	Produto A	A	24000	KG	0.34	1	0	Minas Gerais
3	2	01/02/2010 00:00	2	Cliente AB	Fortaleza	Brasil	2	Produto B	A	730	KG	2.00	5	0	Ceará
4	4	02/02/2010 00:00	3	Cliente BC	Baturité	Brasil	1	Produto A	A	250	KG	7.00	2	0	Ceará
5	5	03/02/2010 00:00	4	Cliente CC	Fortaleza	Brasil	1	Produto A	A	4500	KG	0.34	1	0	Ceará
6	6	04/03/2010 00:00	5	Cliente CD	Eusebio	Brasil	1	Produto A	A	11200	KG	0.34	3	0	Ceará
7	1	01/02/2010 00:00	1	Cliente AA	Florianópolis	Brasil	1	Produto A	A	4000	KG	0.34	2	0	Santa Catarina

Showing all 10 rows.



Command took 1.49 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 15:08:59 on ZouzaCluster

Cmd 58

-- Fato Vendas --

```
CREATE OR REPLACE VIEW fato_vendas
AS (
  SELECT CódigoVenda
    , DataVenda
    , Quantidade
    , ValorUnitario
    , CódigoCliente
    , CódigoProduto
    , CódigoVendedor
   FROM vendas_silver
  WHERE delete = 0
)
```



Refined Tables
(Silver)

Lakehouses (dimensões - data mart)

```
-----  
-- dim Cliente --  
-----  
  
CREATE OR REPLACE VIEW dim_cliente  
AS (  
    SELECT DISTINCT  
       CodigoCliente  
, NomeCliente  
, Cidade  
, Estado  
, País  
    FROM vendas_silver  
    WHERE delete = 0  
)  
  
OK  
  
Command took 0.69 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 15:09:30 on ZouzaCluster
```

Cmd 60

```
-----  
-- dim Produto --  
-----  
  
CREATE OR REPLACE VIEW dim_produto  
AS (  
    SELECT DISTINCT  
       CodigoProduto  
, NomeProduto  
, TipoProduto  
, Und  
    FROM vendas_silver  
    WHERE delete = 0  
)  
  
OK
```

Command took 0.55 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 15:09:44 on ZouzaCluster



Refined Tables
(Silver)

Lakehouses (consultando data mart)

Consultando algumas views do Data Mart

```
Cmd 62
select * from fato_vendas
```

► (3) Spark Jobs

	CodigoVenda	DataVenda	Quantidade	ValorUnitario	CodigoCliente	CodigoProduto	CodigoVendedor
1	3	01/02/2010 00:00	4200	0.34	5	1	4
2	10	08/05/2010 00:00	24000	0.34	5	1	1
3	2	01/02/2010 00:00	730	2.00	2	2	5
4	4	02/02/2010 00:00	250	7.00	3	1	2
5	5	03/02/2010 00:00	4500	0.34	4	1	1
6	6	04/03/2010 00:00	11200	0.34	5	1	3
7	1	01/02/2010 00:00	4000	0.34	1	1	2

Showing all 10 rows.

[grid] [refresh] [down]

Command took 2.26 seconds -- by assouzaconsult@gmail.com at 16/06/2021 15:09:51 on ZouzaCluster

```
Cmd 63
select * from dim_produto
```

► (1) Spark Jobs

	CodigoProduto	NomeProduto	TipoProduto	Und
1	1	Produto A	A	KG
2	2	Produto B	A	KG

Showing all 2 rows.

[grid] [refresh] [down]

Command took 2.03 seconds -- by assouzaconsult@gmail.com at 07/06/2021 08:07:13 on ZouzaCluster


 Refined Tables
 (Silver)

Lakehouses (consultando data mart)

Exemplo de relacionamento entre Tabelas

- Fato_Vendas com a Dim_Produto

Cmd 67

```
select ven.*  
      , pro.nomeproduto  
     from fato_vendas ven  
join dim_produto pro on pro.codigoproduto = ven.codigoproduto
```

▶ (4) Spark Jobs

	CodigoVenda	DataVenda	Quantidade	ValorUnitario	CodigoCliente	CodigoProduto	CodigoVendedor	nomeproduto
1	3	01/02/2010 00:00	4200	0.34	5	1	4	Produto A
2	10	08/05/2010 00:00	24000	0.34	5	1	1	Produto A
3	2	01/02/2010 00:00	730	2.00	2	2	5	Produto B
4	4	02/02/2010 00:00	250	7.00	3	1	2	Produto A
5	5	03/02/2010 00:00	4500	0.34	4	1	1	Produto A
6	6	04/03/2010 00:00	11200	0.34	5	1	3	Produto A
7	1	01/02/2010 00:00	4000	0.34	1	1	2	Produto A

Showing all 10 rows.



Command took 4.06 seconds -- by aasouzaconsult@gmail.com at 24/06/2021 11:39:09 on ZouzaCluster

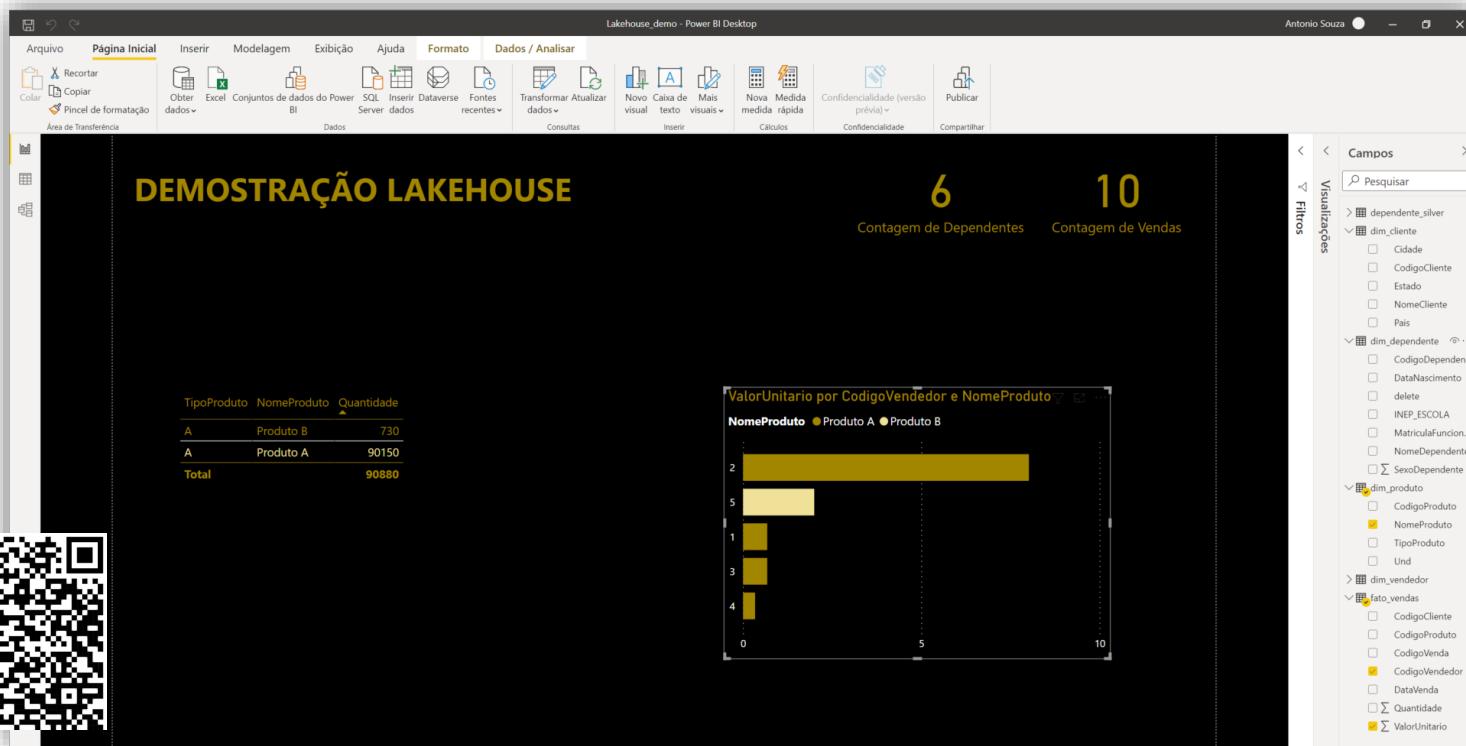


- Transações ACID
- Validação de esquema
- Indexação
- Desempenho de SQL



Refined Tables
(Silver)

Lakehouses (power bi)

 Power BI

The screenshot shows a Power BI Desktop interface with the following elements:

- Top Bar:** Arquivo, Página Inicial, Inserir, Modelagem, Exibição, Ajuda, Formato, Dados / Analisar.
- Left Sidebar:** Área de Transferência, Recortar, Copiar, Pincel de formatação, Obter dados (with options for Excel, Conjuntos de dados do Power BI, SQL Server, Dataverse, Fontes recentes), Consultas, Inserir (with options for Novo, Caixa de texto, Mais visuais, Nova medida, Atualizar dados, Confidencialidade (versão prévia), Compartilhar).
- Main Area:**
 - Section 1:** DEMOSTRAÇÃO LAKEHOUSE. Contains a table:

TipoProduto	NomeProduto	Quantidade
A	Produto B	730
A	Produto A	90150
Total		90880
 - Section 2:** Contagem de Dependentes (Value: 6) and Contagem de Vendas (Value: 10).
 - Section 3:** ValorUnitario por CódigoVendedor e NomeProduto. A horizontal bar chart with categories 2, 5, 1, 3, 4. The legend indicates Produto A (light blue) and Produto B (dark blue). The chart has a Y-axis from 0 to 10 and an X-axis from 0 to 10.
- Right Sidebar:** Campos (Fields) and Filtros (Filters). The Fields pane lists various dimensions and fact tables with their respective columns, many of which are collapsed. The Filters pane shows filters applied to the fields.



Lakehouses (simulando incremento...)

Anexando novos registros e atualizações a uma tabela delta existente

Cmd 69

Inserções na Tabela de Dependentes (forma I - Direta - Apenas INSERTS)

- teve alterações no ERP e a replicação mandou novas adições...

Cmd 70

```
-- Existem dois padrões para modificar as tabelas Delta existentes:  
-- - anexar arquivos a um diretório existente de arquivos Delta  
-- - mesclando um conjunto de atualizações e inserções  
  
-- Nesta lição, exploramos o primeiro.  
-- No contexto de nosso pipeline de ingestão de dados, esta é a adição de novos arquivos brutos à nossa única fonte da Verdade  
-- primeiramente trazemos os arquivos para a RAW - Bronze (aqui é só uma das formas e podem ser feitas de diversas outras maneiras)  
DROP TABLE IF EXISTS tbdependente_20210607_0620;  
  
-- Carga Batch  
-- Uma carga com dados atualizados de dependentes  
CREATE TABLE tbdependente_20210607_0620  
USING csv  
OPTIONS (  
    PATH "/FileStore/tables/erp/tbdependente_20210607_0620.csv", header "true", sep ";" -- arquivos com inserções na tabela dep  
);
```

> (1) Spark Jobs

OK

Command took 19.32 seconds -- by asouzaconsult@gmail.com at 16/06/2021 15:13:23 on ZouzeCluster

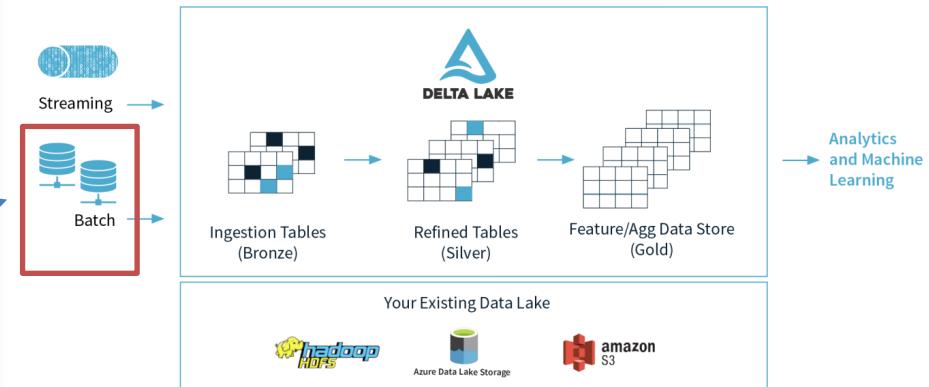
Cmd 71

```
-- O que veio no arquivo acima... (esta adição...)  
SELECT CdDep as CódigoDependente  
    , NmDep as NomeDependente  
    , DtNasc as DataNascimento  
    , MatFunc as MatrículaFuncionário  
    , delete -- 0 ativo e 1 apagado  
    , case when SxDep = "Mas" then 1 else 0 end as SexoDependente  
FROM tbdependente_20210607_0620
```

> (1) Spark Jobs

	CódigoDependente	NomeDependente	DataNascimento	MatrículaFuncionario	delete	SexoDependente
1	10	Dependente 10	02/03/2021 00:00	2	0	0

Showing all 1 rows.



Ingestion Tables
(Bronze)

Lakehouses (simulando incremento...)

Adicionando de forma direta os dados que chegaram à Silver

- pois eram apenas *inserts*

Cmd 73

```
-- colocar na ordem de ordenação das colunas da silver
INSERT INTO dependente_silver
SELECT CdDep    as CódigoDependente
      , NmDep    as NomeDependente
      , DtNasc   as DataNascimento
      , MatFunc  as MatrículaFuncionário
      , delete -- 0 ativo e 1 apagado
      , case when SxDep = "Masc" then 1 else 0 end as SexoDependente
FROM tbdependente_20210607_0820
```

▶ (5) Spark Jobs

	num_affected_rows	num_inserted_rows
1	1	1

Showing all 1 rows.



Command took 7.40 seconds -- by sasouzaconsult@gmail.com at 16/06/2021 15:14:02 on ZouzaCluster



Refined Tables
(Silver)

Lakehouses (power bi...)



DEMOSTRAÇÃO LAKEHOUSE

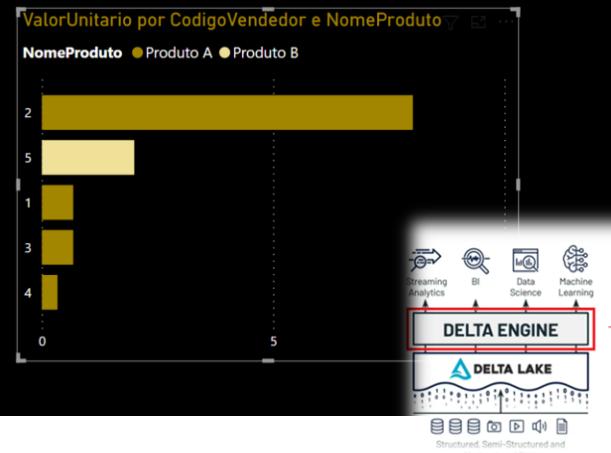
TipoProduto	NomeProduto	Quantidade
A	Produto B	730
A	Produto A	90150
Total		90880

7

Contagem de Dependentes

10

Contagem de Vendas



- Transações ACID
- Validação de esquema
- Indexação
- Desempenho de SQL

Lakehouses (viagem no tempo...)

Viagem no Tempo

- Versão 0

Cod 75

```
-- Delta Lake pode consultar uma versão anterior de uma tabela Delta usando um recurso conhecido como viagem no tempo. Aqui --> https://docs.databricks.com/delta/quick-start.html#query-an-earlier-version-of-the-table-time-travel
select * from dependente_silver VERSION AS OF 0
```

(4) Spark Jobs

Código Dependente	Nome Dependente	Data Nascimento	Matrícula Funcionário	delete	Sexo Dependente
1	Dependente 1	02/02/2010 00:00	1	0	1
2	Dependente 2	05/04/2012 00:00	3	0	1
3	Dependente 5	06/07/2019 00:00	4	0	1
4	Dependente 3	04/03/2013 00:00	3	0	0
5	Dependente 4	05/05/2010 00:00	4	0	0
6	Dependente 6	02/03/2018 00:00	9	0	0

Showing all 6 rows.

Command took 2.67 seconds -- by ssouzaconsult@gmail.com at 16/06/2021 15:15:33 on ZouzeCluster

Cod 76

Viagem no Tempo

- Versão recente

Cod 77

```
select * from dependente_silver
```

(3) Spark Jobs

Código Dependente	Nome Dependente	Data Nascimento	Matrícula Funcionário	delete	Sexo Dependente
1	Dependente 10	02/03/2021 00:00	2	0	0
2	Dependente 1	02/02/2010 00:00	1	0	1
3	Dependente 2	05/04/2012 00:00	3	0	1
4	Dependente 5	06/07/2019 00:00	4	0	1
5	Dependente 3	04/03/2013 00:00	3	0	0
6	Dependente 4	05/05/2010 00:00	4	0	0
7	Dependente 6	02/03/2018 00:00	9	0	0

Showing all 7 rows.

Select a file from DBFS

Upload File S3 DBFS Other Data Sources Partner Integrations

_SUCCESS

dependente dw-gold vendas silver

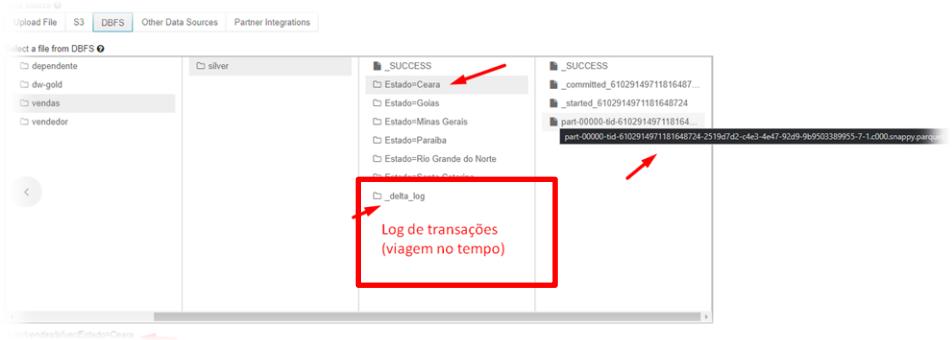
Estadão:Ceará

Log de transações
(viagem no tempo)

dw-gold vendas silver

_SUCCESS

committed_6102914971181648724... started_6102914971181648724... part-00000-fid_6102914971181648724-2519d7d2-c4e3-4e47-92c5-9b79503389955-7-c000.unappy.parquet




Refined Tables
(Silver)

Lakehouses (simulando incrementos)

Inserts, Updates, Deletes na Tabela de Vendas (Upsert)

- Aqui usaremos de forma didática as movimentações do dia da tabela vindo em um .csv
- Mas na vida real, por exemplo, poderá fazer replicação do ERP para o **Amazon S3** usando o **DMS (Data Migration Services)** via **CDC (Change Data Capture)** e montar algum jobs para realizar a leitura destes arquivos que chegam de forma automática.

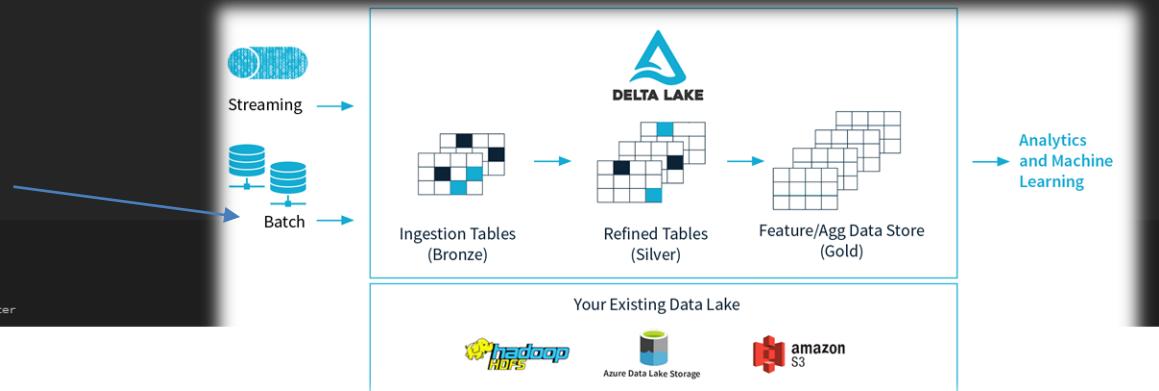
Cmd 100

```
-- Arquivo só com as alterações
DROP TABLE IF EXISTS tbvendas_20210607;

-- Carga Batch
-- Uma carga com dados d-1
CREATE TABLE tbvendas_20210607
USING csv
OPTIONS (
  PATH "/FileStore/tables/erp/tbvendas_20210607.csv", header "true", sep ','
);

▶ (1) Spark Jobs
OK

Command took 1.53 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 15:55:33 on ZouzaCluster
```



Ingestion Tables
(Bronze)

Lakehouses (simulando incrementos)

Criando a View de Upserts da Tabela de Vendas

- todas movimentações do dia (até o momento)...
- Criando view com os inserts, updates e deletes lógicos feitos no ERP (vendas) para adicionarmos na Silver (já com os dados tratados)

Cmd 102

```
-- Criando view com os inserts, updates e deletes lógicos feitos no ERP (vendas) para adicionarmos na Silver (já com os dados tratados)
CREATE OR REPLACE TEMPORARY VIEW upserts
```

```
AS (
    SELECT
        CdVen as CódigoVenda
        , DtVen as DataVenda
        , CdCli as CódigoCliente
        , NmCli as NomeCliente
        , Cidade
        , País
        , CdPro as CódigoProduto
        , NmPro as NomeProduto
        , TpPro as TipoProduto
        , cast(Qtd as int)           as Quantidade
        , Und
        , cast(VrUnit as decimal(18,2)) as ValorUnitario
        , CdVdd                         as CódigoVendedor
        , delete
        , Estado
    FROM tbvendas_20210607
)
```



Refined Tables
(Silver)

Lakehouses (simulando incrementos)

Analisando os schemas das tabelas aqui envolvidas

```
Cmd 104
DESCRIBE vendas_silver
```

	col_name	data_type	comment
1	CodigoVenda	string	
2	DataVenda	string	
3	CodigoCliente	string	
4	NomeCliente	string	
5	Cidade	string	
6	Pais	string	
7	CodigoProduto	string	

Showing all 18 rows.

```
Cmd 105
DESCRIBE upserts
```

	col_name	data_type	comment
1	CodigoVenda	string	null
2	DataVenda	string	null
3	CodigoCliente	string	null
4	NomeCliente	string	null
5	Cidade	string	null
6	Pais	string	null
7	CodigoProduto	string	null

Showing all 15 rows.



- Transações ACID
- Validação de esquema
- Indexação
- Desempenho de SQL

Lakehouses (simulando incrementos)

Verificando os registros que serão movimentados

```
Cmd 107
-- Verificando...
SELECT * FROM upserts
▶ (1) Spark Jobs
```

	CodigoVenda	DataVenda	CodigoCliente	NomeCliente	Cidade	Pais	CodigoProduto	NomeProduto	TipoProduto	Quantidade	Und	ValorUnitario	CodigoVendedor	delete	Estado
1	2	01/02/2010 00:00	2	Cliente AB	Fortaleza	Brasil	2	Produto B	A	730	KG	2.00	5	1	Ceara
2	10	08/05/2010 00:00	5	Cliente BB	Belo Horizonte	Brasil	1	Produto A	A	30000	KG	0.34	2	0	Minas Ge
3	11	01/06/2021 00:00	1	Cliente AA	Florianopolis	Brasil	1	Produto A	A	4000	KG	0.34	2	0	Santa Ca
4	12	02/06/2021 00:00	2	Cliente AB	Fortaleza	Brasil	2	Produto B	A	730	KG	2.00	5	0	Ceara
5	13	04/06/2021 00:00	5	Cliente BB	Belo Horizonte	Brasil	1	Produto A	A	4200	KG	0.34	4	0	Minas Ge
6	14	04/06/2021 00:00	3	Cliente BC	Baturite	Brasil	1	Produto A	A	250	KG	7.00	2	0	Ceara
7	15	05/06/2021 00:00	4	Cliente CC	Fortaleza	Brasil	1	Produto A	A	4500	KG	0.34	1	0	Ceara
8	16	05/06/2021 00:00	5	Cliente CD	Eusebio	Brasil	1	Produto A	A	11200	KG	0.34	3	0	Ceara
9	17	06/06/2021 00:00	6	Cliente DD	Goiania	Brasil	1	Produto A	A	12500	KG	0.34	2	0	Goias
10	18	07/06/2021 00:00	7	Cliente DE	Joao Pessoa	Brasil	1	Produto A	A	12000	KG	0.34	2	0	Paraiba
11	19	07/06/2021 00:00	8	Cliente EE	Natal	Brasil	1	Produto A	A	17500	KG	0.34	3	0	Rio Gran

Showing all 11 rows.

Lakehouses (merge)

Aplicando o MERGE

Cmd 109

```
-- A palavra "upsert" é uma mala de viagem das palavras "atualizar" e "inserir", e é isso que ela faz. Um upsert irá
-- Ao fazer o upsert em uma tabela Delta existente, use o Spark SQL para realizar a mesclagem de outra tabela ou visu
alterações.
-- A mesclagem anexa os arquivos novos / inseridos e os arquivos contendo as atualizações ao diretório de arquivos De
MERGE INTO vendas_silver
USING upserts
ON vendas_silver.CodigoVenda = upserts.CodigoVenda
WHEN MATCHED THEN
    UPDATE SET
        vendas_silver.Quantidade      = upserts.Quantidade,
        vendas_silver.ValorUnitario   = upserts.ValorUnitario,
        vendas_silver.CodigoVendedor = upserts.CodigoVendedor,
        vendas_silver.delete          = upserts.delete
WHEN NOT MATCHED THEN
    INSERT (CodigoVenda, DataVenda, CódigoCliente, NomeCliente, Cidade, País, CódigoProduto, NomeProduto, TipoProduto, Quantidade, Unid, ValorUnitario, CódigoVendedor, delete, Estado)
    VALUES (CodigoVenda, DataVenda, CódigoCliente, NomeCliente, Cidade, País, CódigoProduto, NomeProduto, TipoProduto, Quantidade, Unid, ValorUnitario, CódigoVendedor, delete, Estado)
```

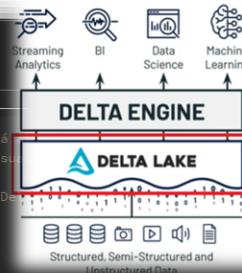
▶ (10) Spark Jobs

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserted_rows
1	11	2	0	9

Showing all 1 rows.



Command took 11.64 seconds -- by assouzaconsult@gmail.com at 16/06/2021 16:03:54 on ZouzaCluster



- ▶ Transações ACID
- ▶ Validação de esquema
- ▶ Indexação
- ▶ Desempenho de SQL

Lakehouses (power bi...)

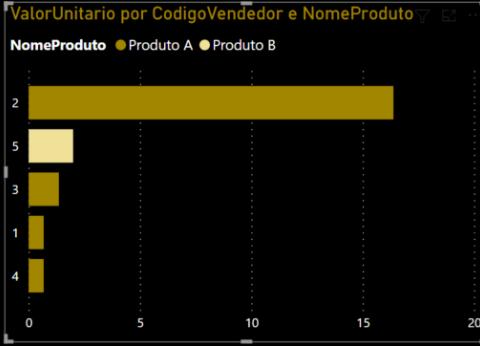


DEMOSTRAÇÃO LAKEHOUSE

TipoProduto	NomeProduto	Quantidade
A	Produto B	730
A	Produto A	162300
Total		163030

7 Contagem de Dependentes

18 Contagem de Vendas



CódigoVendedor	NomeProduto	ValorUnitario
2	Produto A	~16
5	Produto B	~2
3	Produto A	~1
1	Produto A	~0.5
4	Produto A	~0.5

Lakehouses (outras fontes...)

Integrando BI tradicional com outras fontes de dados, ciência de dados e etc...

- Fonte unica da verdade
- Possíveis integrações

Cmd 119

Relacionamento entre Bancos de Dados dentro do Lakehouse

- Pode ler e relacionar tabelas de bancos de dados distintos (lake x default)
 - exemplo abaixo, relacionando do nosso *Data Mart* (lake) com uma base Pública do Censo Escolar já tratada em outra análise de mercado realizada pelo Data Scientist (*default*)

Cmd 120

-- Delta Location (criado anteriormente pela Cientista de Dados)

-- /user/hive/warehouse/censo2020

```
SELECT * FROM default.censo2020 limit 10;
```

-- Aqui pode relacionar com os bd's do ERP por exemplo, ou do Datacenter

▶ (1) Spark Jobs

	ANO	INEP	ESCOLA	CO_ORGAO REGIONAL	TP_SITUACAO_FUNCIONAMENTO	DT_ANO LETIVO_INICIO	DT_ANO LETIVO TERMINO	CO_REGIAO	CO_MESORREGIAO
1	2020	11000023	EEEABNAEL MACHADO DE LIMA - CENE	00009	1	06/02/2020	15/12/2020	1	1101
2	2020	11000040	EMEIEF PEQUENOS TALENTOS	00009	1	06/02/2020	23/12/2020	1	1101
3	2020	11000058	CENTRO DE ENSINO CLASSE A	00009	1	03/02/2020	11/12/2020	1	1101
4	2020	11000082	CENTRO EDUCACIONAL PRESBITERIANO 15 DE NOVEMBRO	00009	1	03/02/2020	04/12/2020	1	1101
5	2020	11000104	CENTRO EDUC CORA CORALINA	00009	1	03/02/2020	15/12/2020	1	1101
6	2020	11000171	CENTRO EDUCACIONAL MOJUCA	00009	1	04/02/2020	29/12/2020	1	1101
7	2020	11000180	INTERACAO - CURSOS E COLEGIO	00009	2	null	null	1	1101

Showing all 10 rows.



Lakehouses (outras fontes...)

```
-- E pode relacionar os 2 bancos... aqui pegando dados do ERP e cruzando com a base pública do Censo Escolar
select *
  from     lake.dim_dependente dep          -- erp
  left join default.censo2020  cen on cen.inep = dep.inep_escola -- base pública (enriquecer seus dados)
```

▶ (2) Spark Jobs

	CodigoDependente	NomeDependente	DataNascimento	MatriculaFuncionario	delete	SexoDependente	INEP_ESCOLA	ANO	INEP	ESCOLA	CO_ORGAO_F
1	10	Dependente 10	02/03/2021 00:00	2	0	0	11000082	2020	11000082	CENTRO EDUCACIONAL PRESBITERIANO 15 DE NOVEMBRO	00009
2	1	Dependente 1	02/02/2010 00:00	1	0	1	11000082	2020	11000082	CENTRO EDUCACIONAL PRESBITERIANO 15 DE NOVEMBRO	00009
3	2	Dependente 2	05/04/2012 00:00	3	0	1	11000082	2020	11000082	CENTRO EDUCACIONAL PRESBITERIANO 15 DE NOVEMBRO	00009
4	5	Dependente 5	06/07/2019 00:00	4	0	1	11000082	2020	11000082	CENTRO EDUCACIONAL PRESBITERIANO 15 DE NOVEMBRO	00009
5	3	Dependente 3	04/03/2013 00:00	3	0	0	11000082	2020	11000082	CENTRO EDUCACIONAL PRESBITERIANO 15 DE NOVEMBRO	00009
6	4	Dependente 4	05/05/2010 00:00	4	0	0	11000082	2020	11000082	CENTRO EDUCACIONAL PRESBITERIANO 15 DE NOVEMBRO	00009

Showing all 6 rows.



Command took 3.38 seconds -- by assouzaconsult@gmail.com at 13/06/2021 16:55:22 on ZouzaCluster

Cmd 122



BI Público - Elaborado pela Z-Educa com Dados do Censo Escolar

- Censo Escolar 2020



Lakehouses (outras fontes...)

Ciência de dados...

- Analisando a mesma base do Censo Escolar que utilizamos acima, só que aqui utilizando Python...

Cmd 124

Montar uma classificador para saber perfil de escola (Pública ou Privada) para investir

Cmd 125

```
%python
#####
# Importando bibliotecas #
#####

import seaborn as sns
from matplotlib import pyplot as plt
import pandas as pd

#####
# Lendo dados direto do DeltaLake #
#####

censo2020_ia = spark.read.format("delta").load("/user/hive/warehouse/censo2020")
censo2020_ia = censo2020_ia.toPandas()

#####
# Filtrado apenas as Escolas que estão em funcionamento #
#####

df = censo2020_ia[censo2020_ia['TP_SITUACAO_FUNCIONAMENTO']=='1']

#####
# Tratando alguns Outliers #
#####

df = df[df['QtdAlunos'] > 25]
df = df[df['QtdAlunos'] < 2000] # Existem alguns outliers, alias, por mim classificados como outliers

# Particular 1 | Públicas 0
df["TP_DEPENDENCIA"] = pd.to_numeric(df["TP_DEPENDENCIA"])

df.loc[ df['TP_DEPENDENCIA'] < 4, 'target' ] = 0
df.loc[ df['TP_DEPENDENCIA'] >= 4, 'target' ] = 1

▶ (1) Spark Jobs
```



```
%python
# Separar as classes
X = df.drop(['target'],axis=1)
y = df['target']

#Abaixo importamos as bibliotecas do algoritmo mencionado acima e também importamos uma biblioteca
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

#Agora, vamos dividir o conjunto em Treinamento e Teste, utilizando o train_test_split importado
X_treino, X_teste, y_treino, y_teste = train_test_split(X, y,test_size=0.30,random_state=42)

#Treinando o Modelo - Na etapa abaixo o algoritmo aprenderá os padrões de dados de treinamento
#então, ser usado para obter previsões dos novos dados cuja resposta de destino você não conhece
#O modelo é o clf_rf.
clf_rf = RandomForestClassifier()

clf_rf.fit(X_treino,y_treino)

#Exibindo as métricas
from sklearn import metrics
print (metrics.classification_report(y_teste,clf_rf.predict(X_teste)))


```

	precision	recall	f1-score	support
0.0	0.95	0.98	0.96	36956
1.0	0.91	0.82	0.87	11399
accuracy			0.94	48355
macro avg	0.93	0.90	0.91	48355
weighted avg	0.94	0.94	0.94	48355

Command took 18.14 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 17:00:16 on ZouzaCluster

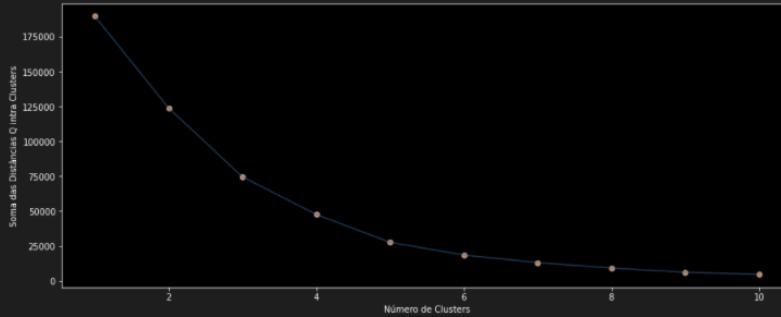
Lakehouses (outras fontes...)

```
%python
# Importando Bibliotecas.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
import plotly.graph_objs as go
from sklearn.cluster import KMeans
import warnings
import os

# Variáveis para criação do modelo
X = censo2020_ia[['TP_SITUACAO_FUNCIONAMENTO', 'TP_DEPENDENCIA', 'TP_LOCALIZACAO']]

# Escolhendo o melhor K (número de clusters)
inertia = []
for n in range(1 , 11):
    algorithm = (KMeans(n_clusters = n))
    algorithm.fit(X)
    inertia.append(algorithm.inertia_)

plt.figure(1 , figsize = (15 , 6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-.' , alpha = 0.5)
plt.xlabel('Número de Clusters') , plt.ylabel('Soma das Distâncias Q intra Clusters')
plt.show()
```



Command took 18.42 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 17:01:55 on ZouzaCluster

```
%python
import pandasql as ps

query = """select clusters, count(clusters) from censo2020_ia group by clusters"""
ps.sqldf(query, locals())
```

Out[54]:

	clusters	count(clusters)
0	0	31059
1	1	60178
2	2	53696
3	3	41763
4	4	27472
5	5	10061

Command took 19.52 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 19:44:36 on ZouzaCluster

Lakehouses (outras fontes...)

Lendo arquivos...

Cmd 139

Lendo arquivos .parquet

- acompanhamento de temperatura de Datacenter
- DW tradicional não permite arquivos como .parquet por padrão, aqui fica tudo integrado
 - e a equipe de infraestrutura hoje analisa estes dados via Power BI e lendo aqui do Lakehouse (não mais recebendo por e-mail...)

Cmd 140

```
--https://databricks.com/glossary/what-is-parquet
DROP TABLE IF EXISTS datacenter_raw;

-- Informações de temperatura do data center da empresa do ERP acima
CREATE TABLE datacenter_raw -- camada bronze
USING parquet
OPTIONS (
  PATH "/FileStore/tables/estudos/data_centers_q2_q3_snappy.parquet"
);
```

▶ (1) Spark Jobs

OK

Command took 1.90 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 16:20:25 on ZouzaCluster

Lakehouses (outras fontes...)



```
-- Lendo primeiro registro  
-- Observem que vem todo o registro basicamente no objeto source  
SELECT * FROM datacenter_raw LIMIT 1;
```

	dc_id	date	source
1	dc-101	2019-04-01	► ["sensor-i-gauge": {"description": "Sensor attached to the container ceilings", "ip": "34.232.238.223", "id": 17, "temps": [16, 31, 22, 23, 25, 24, 20, 18, 20, 23, 16, 18], "co2_level": [1041, 1317, 1122, 864, 1005, 1048]}, "sensor-ipad": {"description": "Sensor ipad attached to carbon cylinders", "ip": "107.208.111.63", "id": 22, "temps": [7, 10, 15, 12, 10, 13, 9, 11, 15, 13, 10, 14], "co2_level": [1000, 1009, 999, 1249, 905, 1088]}, "sensor-inest": {"description": "Sensor attached to the factory ceilings", "ip": "231.24.15.160", "id": 30, "temps": [26, 20, 21, 21, 15, 11, 12, 22, 17, 22, 16, 19], "co2_level": [1000, 1180, 1187, 1136, 1175, 1059]}, "sensor-istick": {"description": "Sensor embedded in exhaust pipes in the ceilings", "ip": "272.112.280.252", "id": 33, "temps": [7, 8, 17, 14, 14, 2, 11, 19, 11, 6, 14, 16], "co2_level": [1232, 1246, 1237, 1142, 1249, 1218]}]

Showing all 1 rows.

Command took 0.62 seconds -- by assouzaconsult@gmail.com at 16/06/2021 16:20:37 on ZouzaCluster

```
Cmd 143

-- explodindo o source
-- observamos que esta dividido em key e value
SELECT EXPLODE (source)
FROM datacenter raw;
```

	key	value
1	sensor-igauge	► { "description": "Sensor attached to the container ceilings", "ip": "34.232.238.223", "id": 17, "temps": [16, 31, 22, 23, 25, 24, 20, 18, 20, 23, 16, 18], "co2_level": [1041, 1317, 1122, 864, 1005, 1048] }
2	sensor-ipad	► { "description": "Sensor ipad attached to carbon cylinders", "ip": "107.208.111.63", "id": 22, "temps": [7, 10, 15, 12, 10, 13, 9, 1, 15, 13, 10, 14], "co2_level": [1000, 1009, 999, 1249, 905, 1088] }
3	sensor-inest	► { "description": "Sensor attached to the factory ceilings", "ip": "231.24.15.160", "id": 30, "temps": [26, 20, 21, 21, 15, 11, 12, 22, 17, 22, 16, 19], "co2_level": [1000, 1180, 1187, 1136, 1175, 1059] }
4	sensor-istick	► { "description": "Sensor embedded in exhaust pipes in the ceilings", "ip": "272.112.280.252", "id": 33, "temps": [7, 8, 17, 14, 14, 2, 11, 19, 11, 6, 14, 16], "co2_level": [1232, 1246, 1237, 1142, 1249, 1218] }
5	sensor-igauge	► { "description": "Sensor attached to the container ceilings", "ip": "47.132.238.88", "id": 17, "temps": [12, 16, 9, 10, 14, 13, 15, 12, 16, 16, 16, 16, 16, 16], "co2_level": [1026, 1020, 1025, 1023, 1024, 1027] }

```
-- Explodindo e criando o select --
-----
CREATE OR REPLACE VIEW tabelao_datacenter AS
WITH explode_source
AS
(
SELECT
dc_id,
to_date(date) AS date,
EXplode (source)
FROM datacenter_raw
)
SELECT
dc_id,
date,
key,
value.description,
value.ip,
value.temps,
value.co2_level
FROM explode_source;
```

Lakehouses (outras fontes...)

Equipe de Infraestrutura pode analisar seus dados!

:)

Cmd 146

```
select * from tabelao_datacenter
```

► (1) Spark Jobs

	dc_id	date	key	description	ip	temps	co2_level
1	dc-101	2019-04-01	sensor-igauge	Sensor attached to the container ceilings	34.232.238.223	► [16, 31, 22, 23, 25, 24, 20, 18, 20, 23, 16, 18]	► [1041, 1317, 1122, 864, 1005, 1048]
2	dc-101	2019-04-01	sensor-ipad	Sensor ipad attached to carbon cylinders	107.208.111.63	► [7, 10, 15, 12, 10, 13, 9, 11, 15, 13, 10, 14]	► [1000, 1009, 999, 1249, 905, 1088]
3	dc-101	2019-04-01	sensor-inest	Sensor attached to the factory ceilings	231.24.15.160	► [26, 20, 21, 21, 15, 11, 12, 22, 17, 22, 16, 19]	► [1000, 1180, 1187, 1136, 1175, 1059]
4	dc-101	2019-04-01	sensor-istick	Sensor embedded in exhaust pipes in the ceilings	272.112.280.252	► [7, 8, 17, 14, 14, 2, 11, 19, 11, 6, 14, 16]	► [1232, 1246, 1237, 1142, 1249, 1218]

Truncated results, showing first 1000 rows.



Command took 0.82 seconds -- by sasouzaconsult@gmail.com at 16/06/2021 16:22:23 on ZouzeCluster

Lakehouses (outras fontes...)

Projetos iniciados para leitura de arquivos (.png, .txt, .pdf e etc...)

- <https://docs.databricks.com/data/data-sources/image.html>

```
Cmd 148
```

```
%python
```

```
df = spark.read.format("binaryFile").load("/FileStore/tables/estudos/Delta_Lake.png")
df.printSchema()
```

```
df.show()
```

```
▶ (1) Spark Jobs
```

```
▶ └─ df: pyspark.sql.dataframe.DataFrame = [path: string, modificationTime: timestamp ... 2 more fields]
```

```
root
```

```
|-- path: string (nullable = true)
|-- modificationTime: timestamp (nullable = true)
|-- length: long (nullable = true)
|-- content: binary (nullable = true)
```

```
+-----+-----+-----+
|       path|modificationTime|length|           content|
+-----+-----+-----+
|dbfs:/FileStore/t...|2021-06-07 18:56:07|174449|[89 50 4E 47 0D 0...|
+-----+-----+-----+
```

Lakehouses (outras fontes...)

```
%python
df = spark.read.text("/FileStore/tables/estudos/Arquivotexto.txt")
#df = spark.read.text("/FileStore/tables/estudos/Lakehouse_Uma_saída_para_as_limitações_dos_Data_Lakes.pdf")

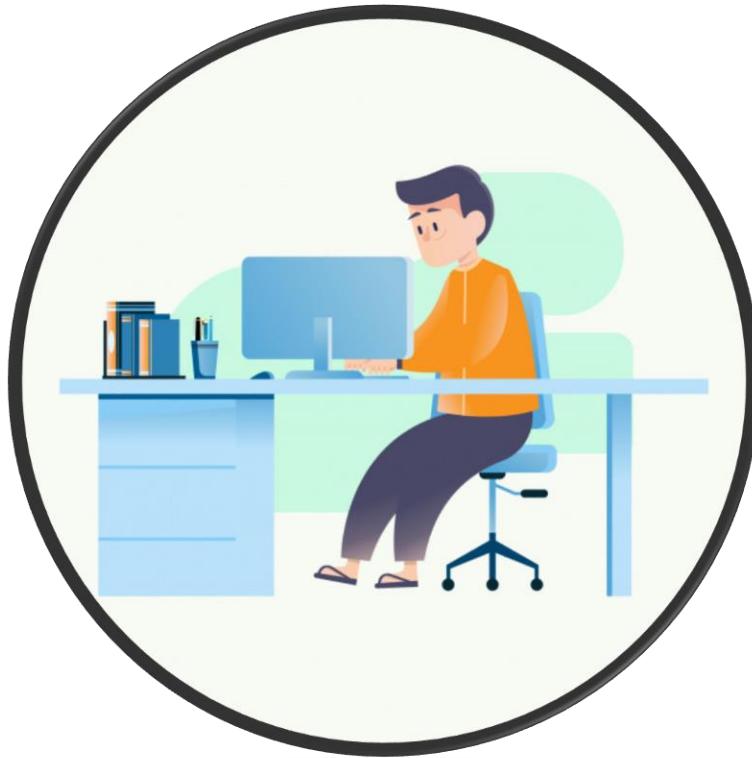
df.collect()

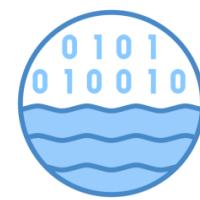
▶ (1) Spark Jobs
▶   df: pyspark.sql.dataframe.DataFrame = [value: string]

Out[5]: [Row(value='Gerenciamento de Dados: dos Dados ao Lakehouse'),
 Row(value=''),
 Row(value='Introdução'),
 Row(value=''),
 Row(value='Nos últimos anos empresas vem coletando uma grande quantidade de dados de múltiplas fontes distintas (ERP, CRM, E-Commerce, IoT...), analisar de forma centralizada este grande volume de dados e disponibilizar para e quipes de Negócios, Analistas e Cientistas de Dados é, no mínimo, desafiador.'),
 Row(value=''),
 Row(value='Historicamente, muitas soluções diferentes foram criadas e utilizadas para atender a essa necessidade: um Banco de Dados, um Data Warehouse e, durante a última década, o conceito de Data Lake. Todas essas soluções tiveram seus benefícios óbvios na época, mas também diferentes tipos de limitações que se tornaram aparentes conforme as necessidades de gerenciamento de dados mudaram ao longo dos anos. O surgimento da nuvem está criando uma oportunidade para as equipes de dados repensarem suas abordagens. Dados nas plataformas de cloud (nuvem) moderna estão seguindo uma nova arquitetura - Lakehouse.'),
 Row(value=''),
 Row(value='O Lakehouse simplifica radicalmente a infraestrutura de dados e acelera a inovação em uma era em que a Análise de Dados e o Machine Learning (Aprendizagem de Máquina) vem tomado conta das empresas. Essa nova arquitetura mescla as melhores funcionalidades dos Data Lakes com as melhores dos Data Warehouse. Portanto, os mais tradicionais casos de Data Warehouse são suportados na abordagem Lakehouse.'),
 Row(value=''),
 Row(value='Banco de Dados, Data Warehouse e Data Lakes (Necessidades e Limitações)'),
 Row(value=''),
 Row(value='O gerenciamento de dados vem mudando com o passar dos anos conforme a necessidade de acesso mais rápido aos dados vem crescendo, dados estes estruturados e não estruturados, com grande volumetria e múltiplas origens.'),
 Row(value=''),
 Row(value='Primeiramente vieram os bancos de dados relacionais, que as empresas utilizavam para coletar, armazenar e analisar dados de forma simples e confiável. Por muitos anos eles foram suficientes pela relativamente "baixa" quantidade de dados.')
]

Command took 0.55 seconds -- by aasouzaconsult@gmail.com at 16/06/2021 16:24:56 on ZouzaCluster
```

Lakehouses (vendo na prática...)





Obrigado e fico a disposição!!!

