

Data Science  
Academy

Data Science Academy marxv49@gmail.com 5a686b02be32f0447a0e403b

# Big Data Real-Time Analytics com Python e Spark





Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Big Data Real-Time Analytics com Python e Spark

**Seja muito bem-vindo(a)!**



Data Science  
Academy

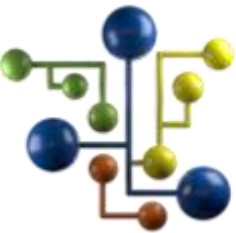
Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Big Data Real-Time Analytics com Python e Spark

## Apache Spark SQL



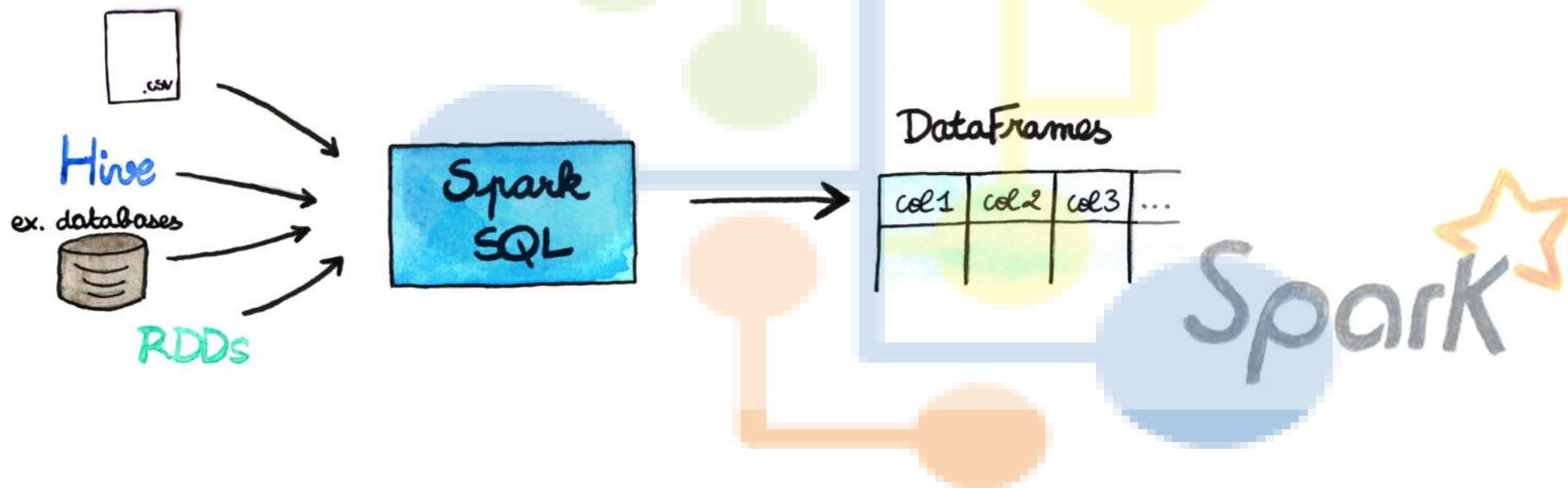




Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Apache Spark SQL





# Apache Spark SQL

## O que vamos estudar neste capítulo?

- Computação em Nuvem e em Cluster
- Linguagem SQL
- Spark SQL com RDD's
- Spark SQL com Dataframes
- Spark SQL com Arquivos CSV
- Spark SQL com Arquivos JSON
- Spark SQL com Bancos de Dados Relacionais
- Spark SQL com Bancos de Dados Não-Relacionais
- Tabelas Temporárias com Spark SQL
- Como Construir um Cluster Spark em Nuvem



Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Big Data Real-Time Analytics com Python e Spark

## Computação em Nuvem e em Cluster





# Computação em Nuvem e em Cluster





Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Computação em Nuvem e em Cluster

Não temos como processar e armazenar  
Big Data em apenas uma máquina.

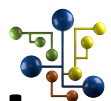
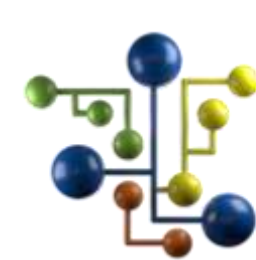




# Computação em Nuvem e em Cluster



E a solução criada para poder armazenar e processar todo esse volume de dados, é a Cloud Computing. O conceito de Cloud Computing ou Computação em Nuvem é relativamente simples.



Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

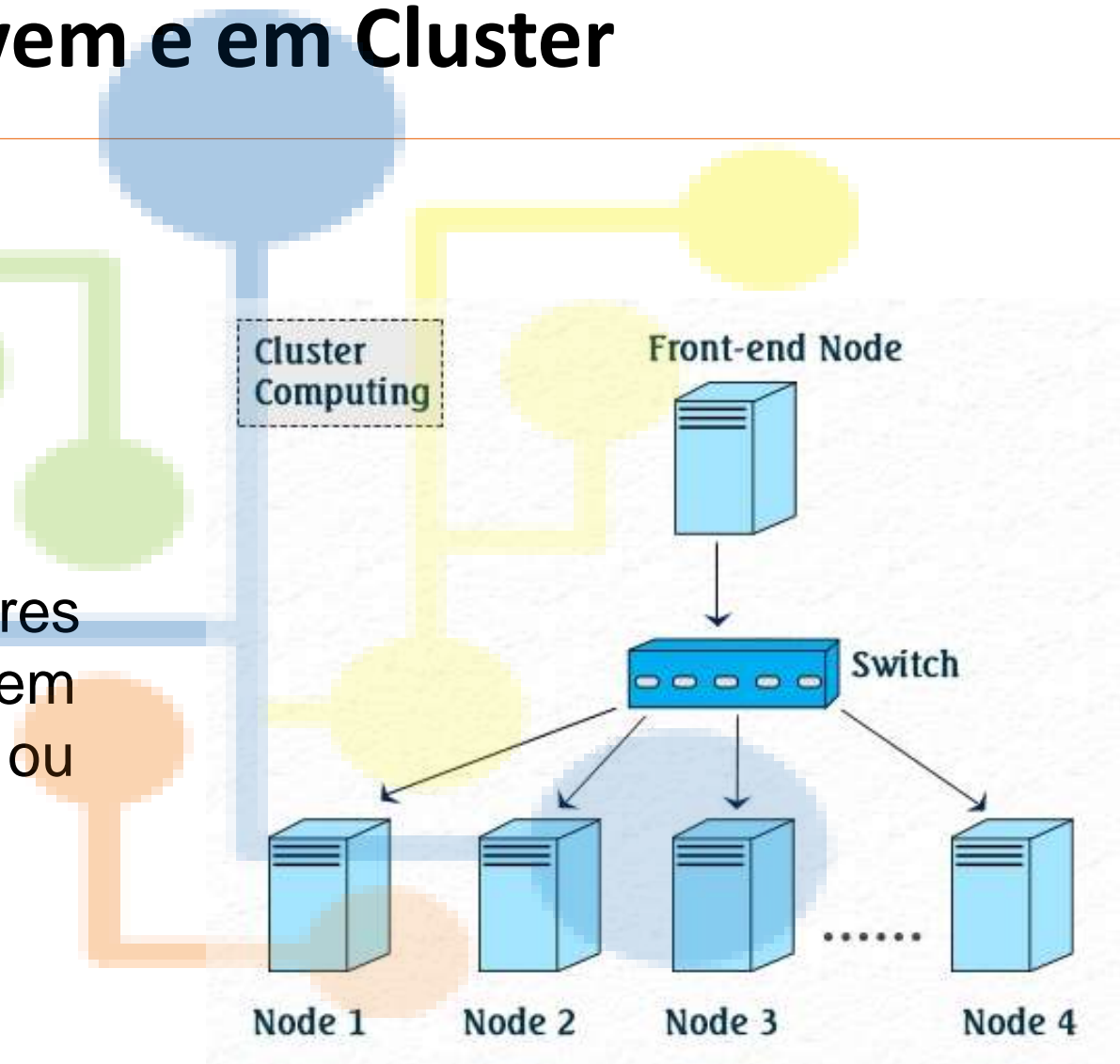
# Computação em Nuvem e em Cluster





# Computação em Nuvem e em Cluster

Um cluster é um sistema que compreende 2 ou mais computadores (chamados nodes) que trabalham em conjunto para executar aplicações ou realizar outras tarefas.







Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Computação em Nuvem e em Cluster



Node



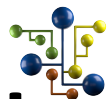
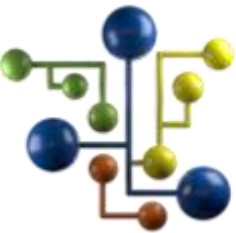
Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Computação em Nuvem e em Cluster







# Computação em Nuvem e em Cluster

Por que não usar máquinas de baixo custo, ao invés de super computadores, e depois unir essas máquinas de baixo custo em clusters, de modo que pudéssemos criar um único sistema com milhares de máquinas, milhares de CPU's e muita, muita memória RAM?





Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Computação em Nuvem e em Cluster

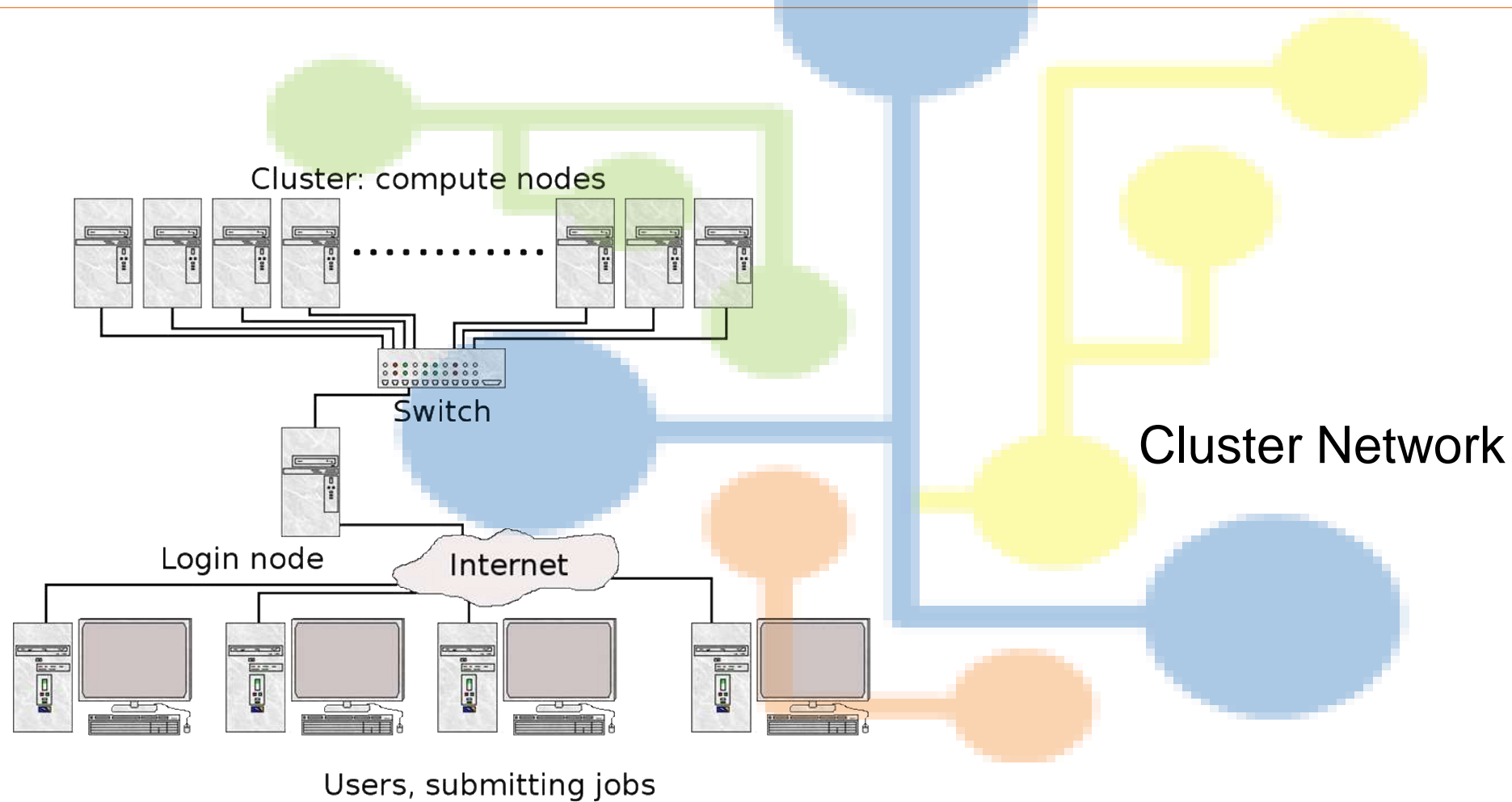


Google:

1 a 5% dos HD's vão falhar por ano e cerca de 0,2% de falhas com memória.



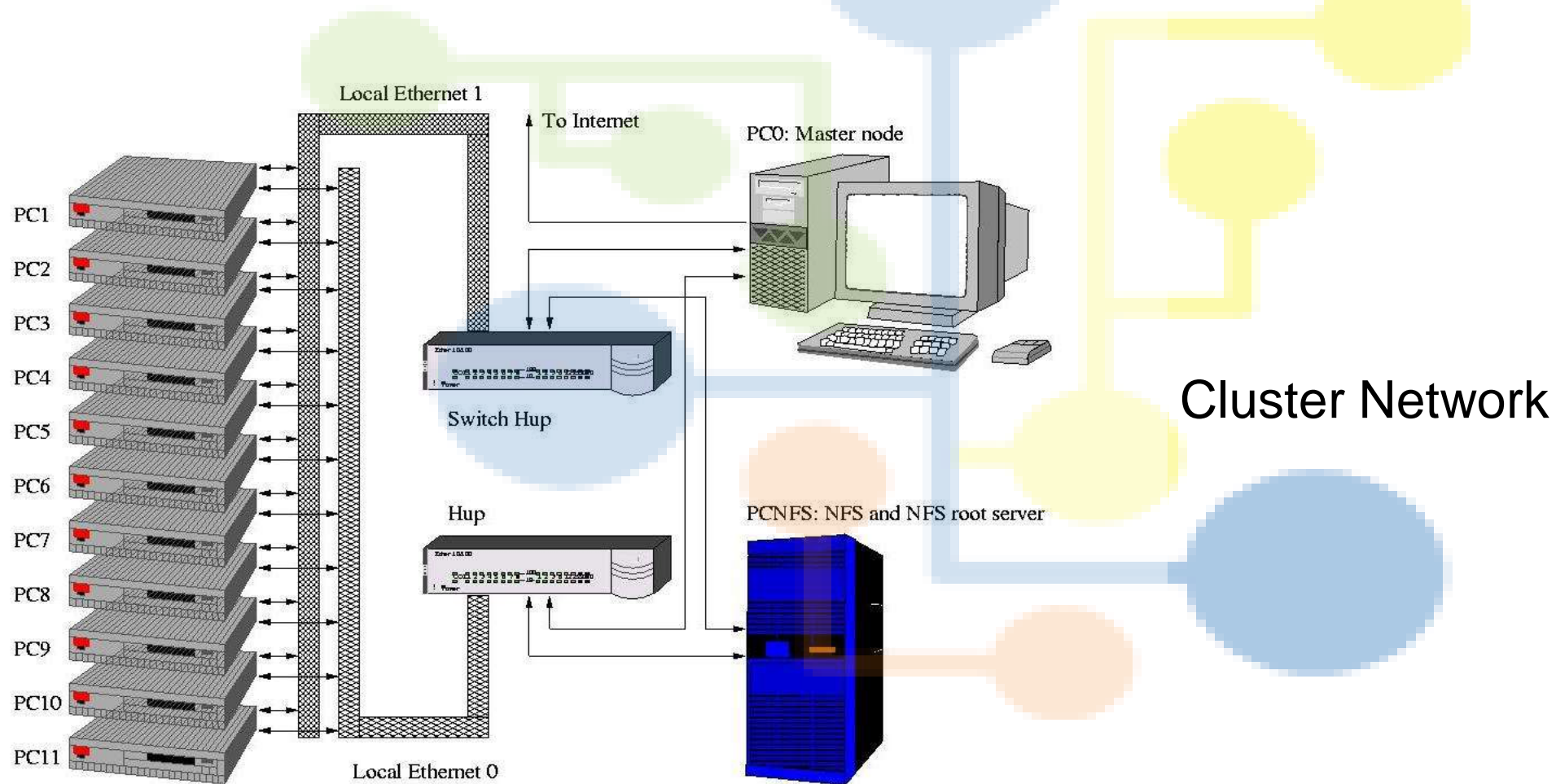
# Computação em Nuvem e em Cluster







# Computação em Nuvem e em Cluster

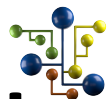




# Computação em Nuvem e em Cluster

Ok, resolvemos o problema de armazenamento e processamento, distribuindo os dados por clusters (conjuntos de computadores) espalhados por datacenters em todo mundo e oferecendo este serviço pela internet!



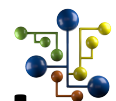
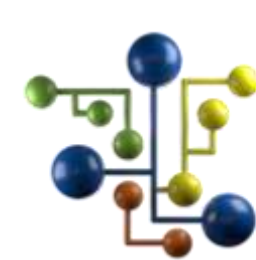


Data Science  
Academy

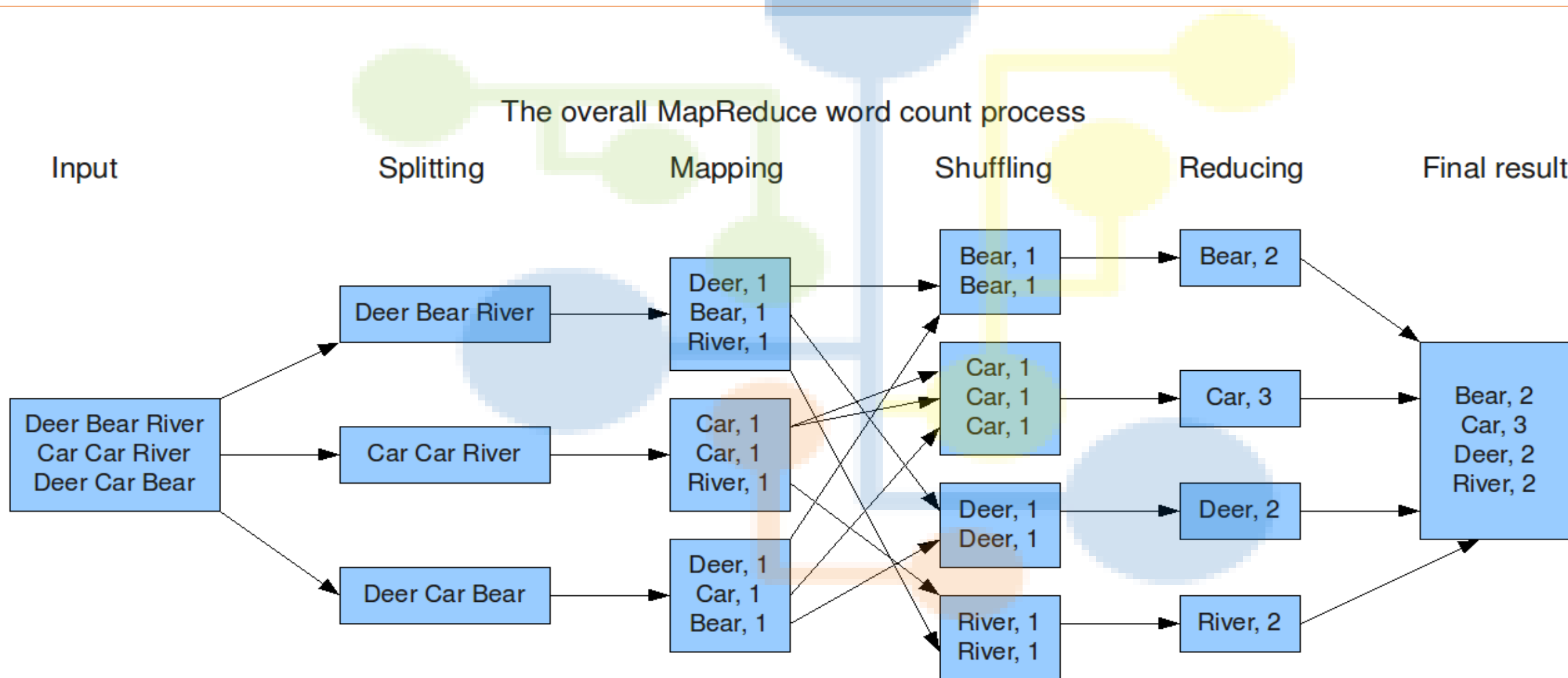
Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Computação em Nuvem e em Cluster

Mas como vamos processar esses dados distribuídos?

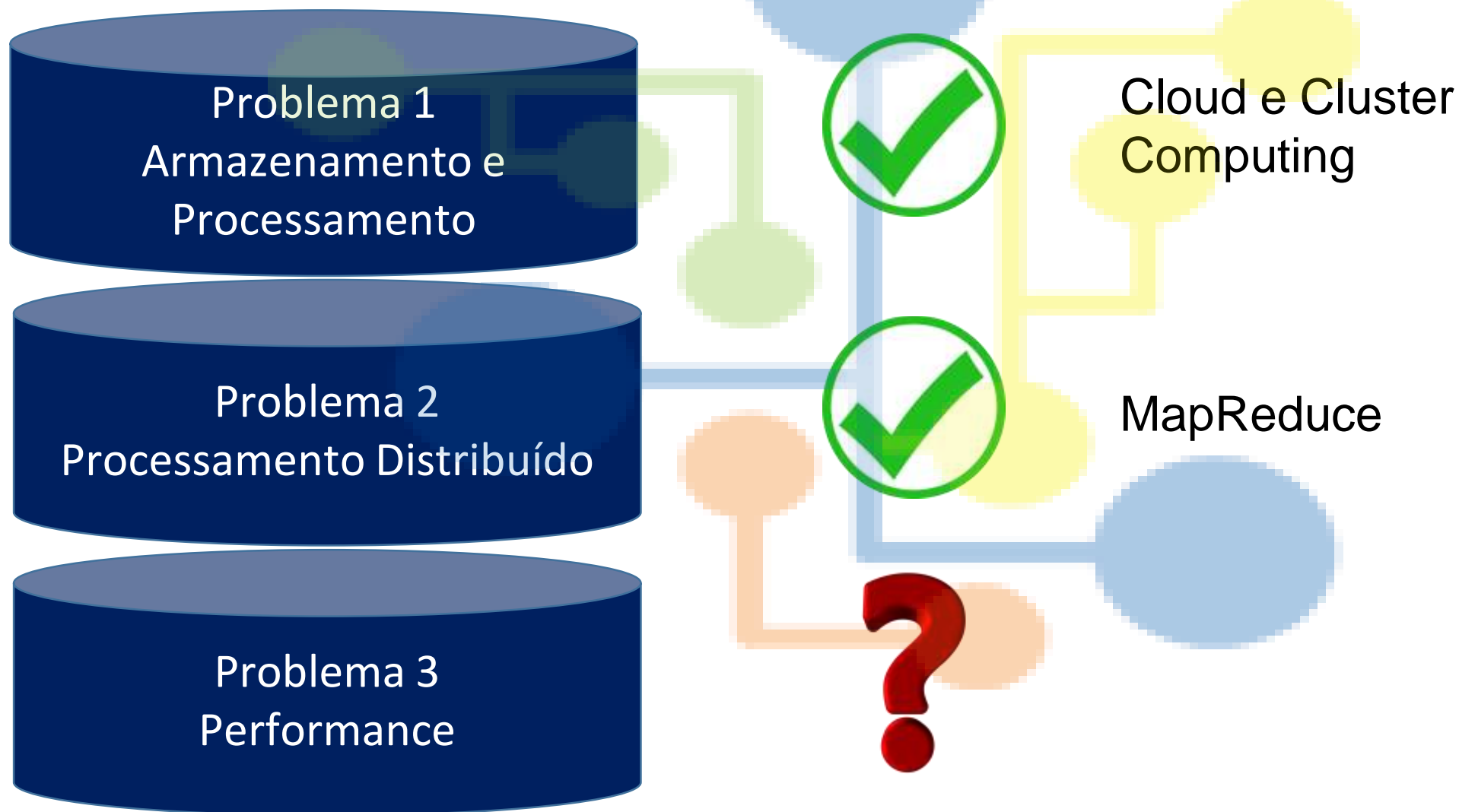


# Computação em Nuvem e em Cluster





# Computação em Nuvem e em Cluster



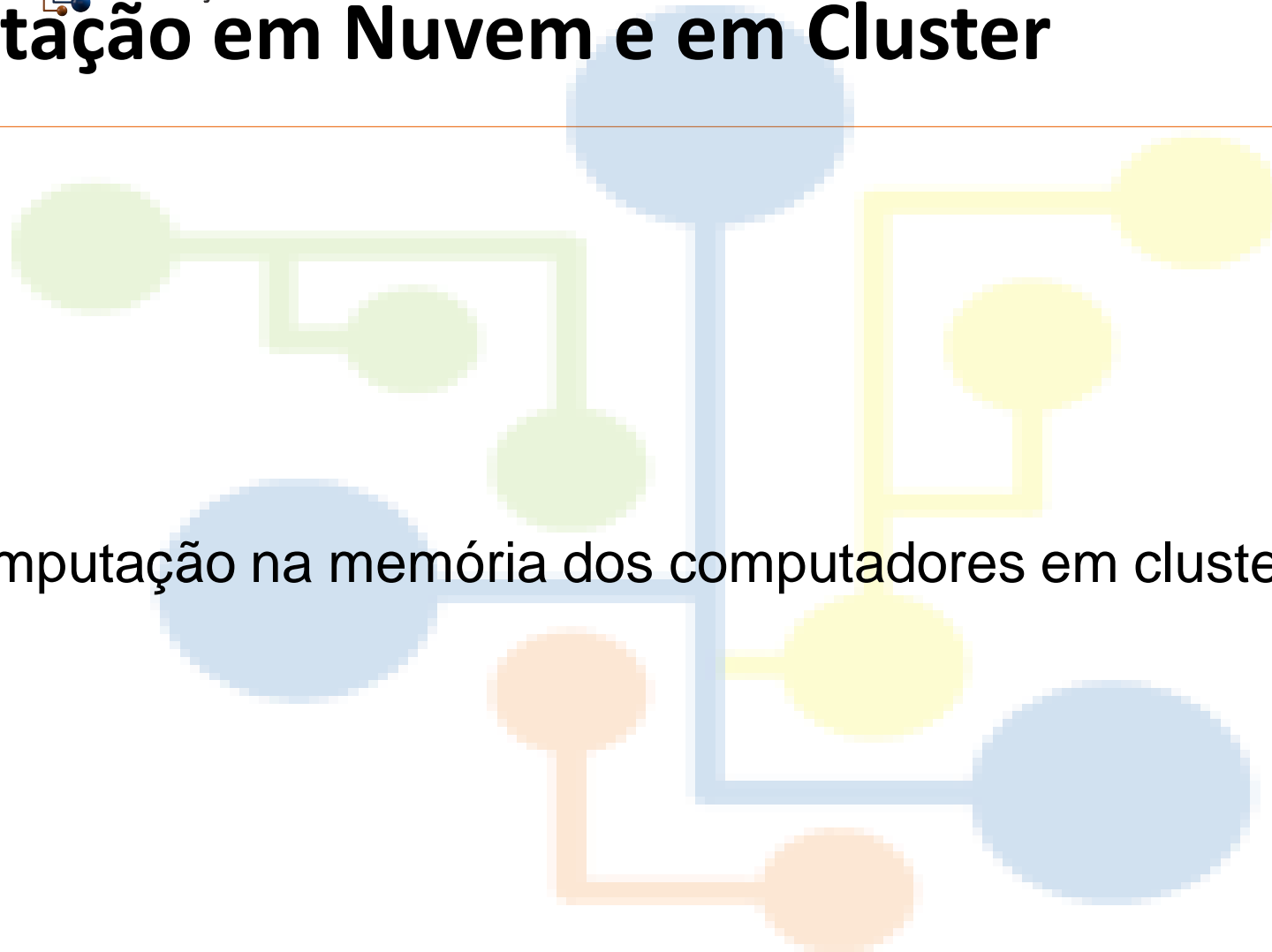


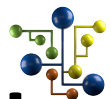
Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

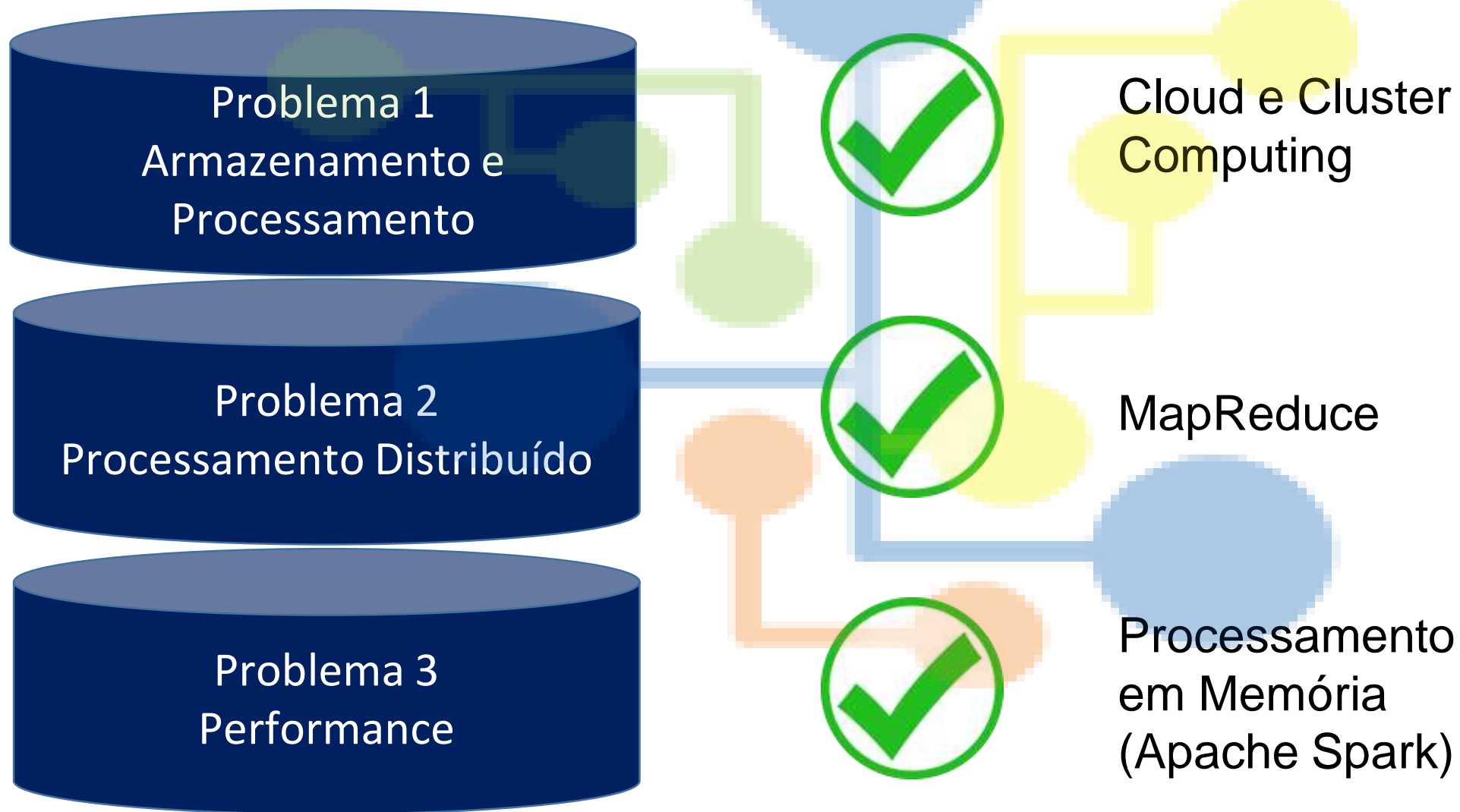
# Computação em Nuvem e em Cluster

Computação na memória dos computadores em cluster!

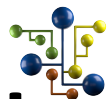
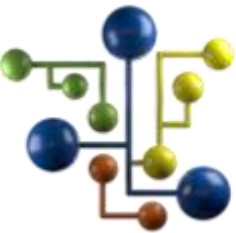




# Computação em Nuvem e em Cluster

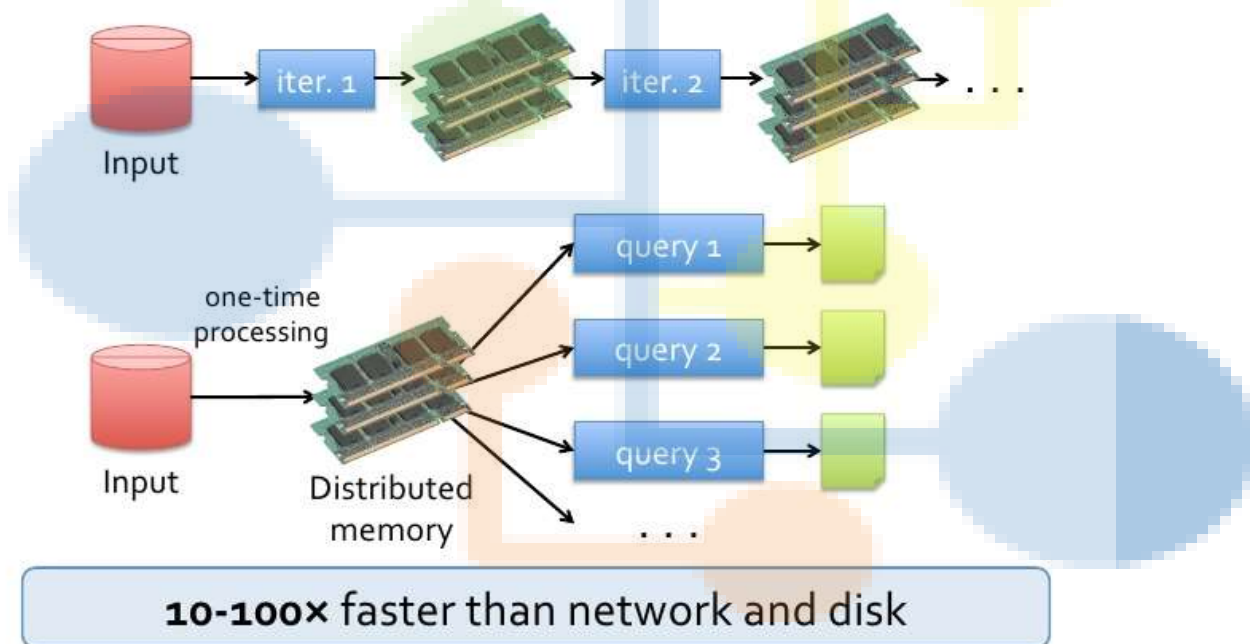


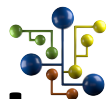




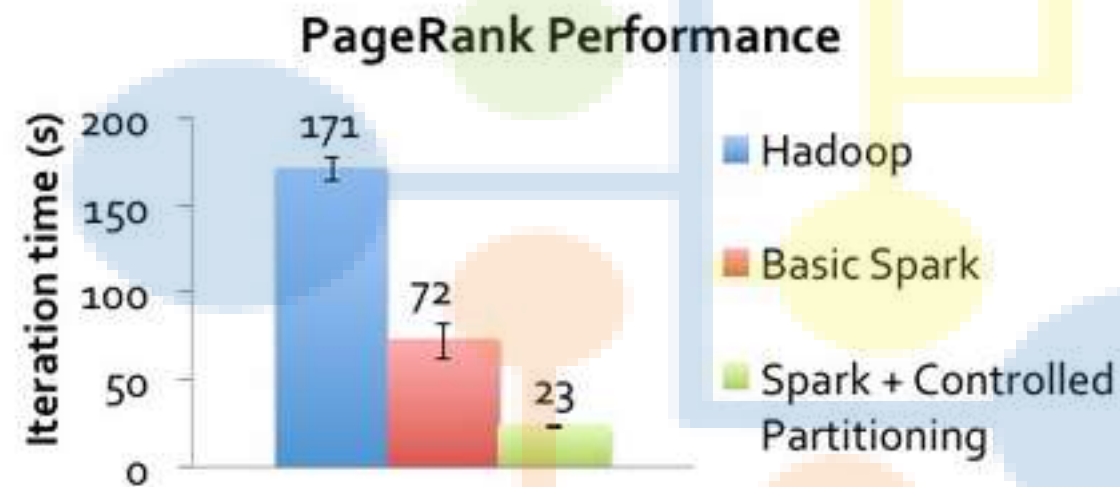
# Computação em Nuvem e em Cluster

## Data Sharing in Spark





# Computação em Nuvem e em Cluster

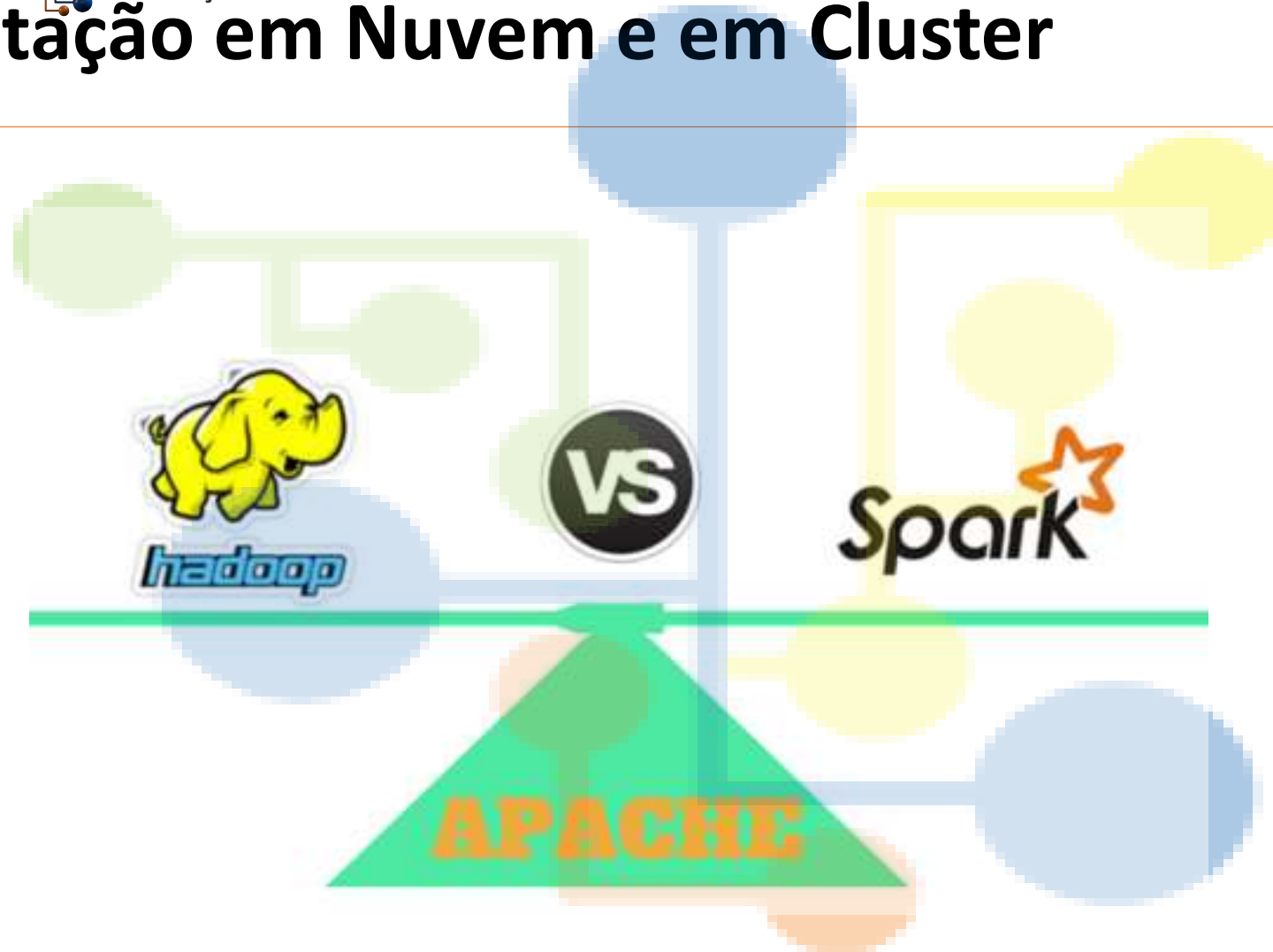




Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Computação em Nuvem e em Cluster





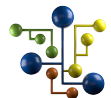
Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

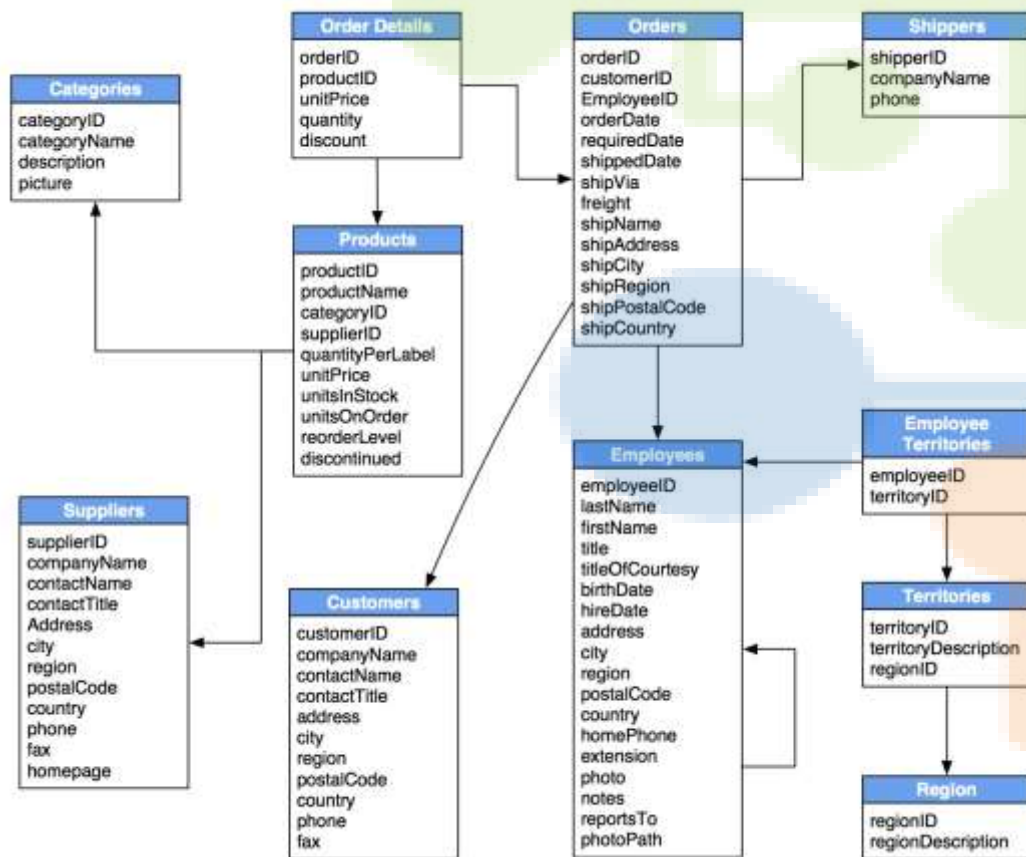
# Big Data Real-Time Analytics com Python e Spark

## Linguagem SQL



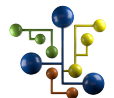


# Linguagem SQL



Usamos Linguagem SQL para consultar e manipular dados em bancos de dados.





Data Science  
Academy

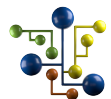
Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Linguagem SQL



# SQL

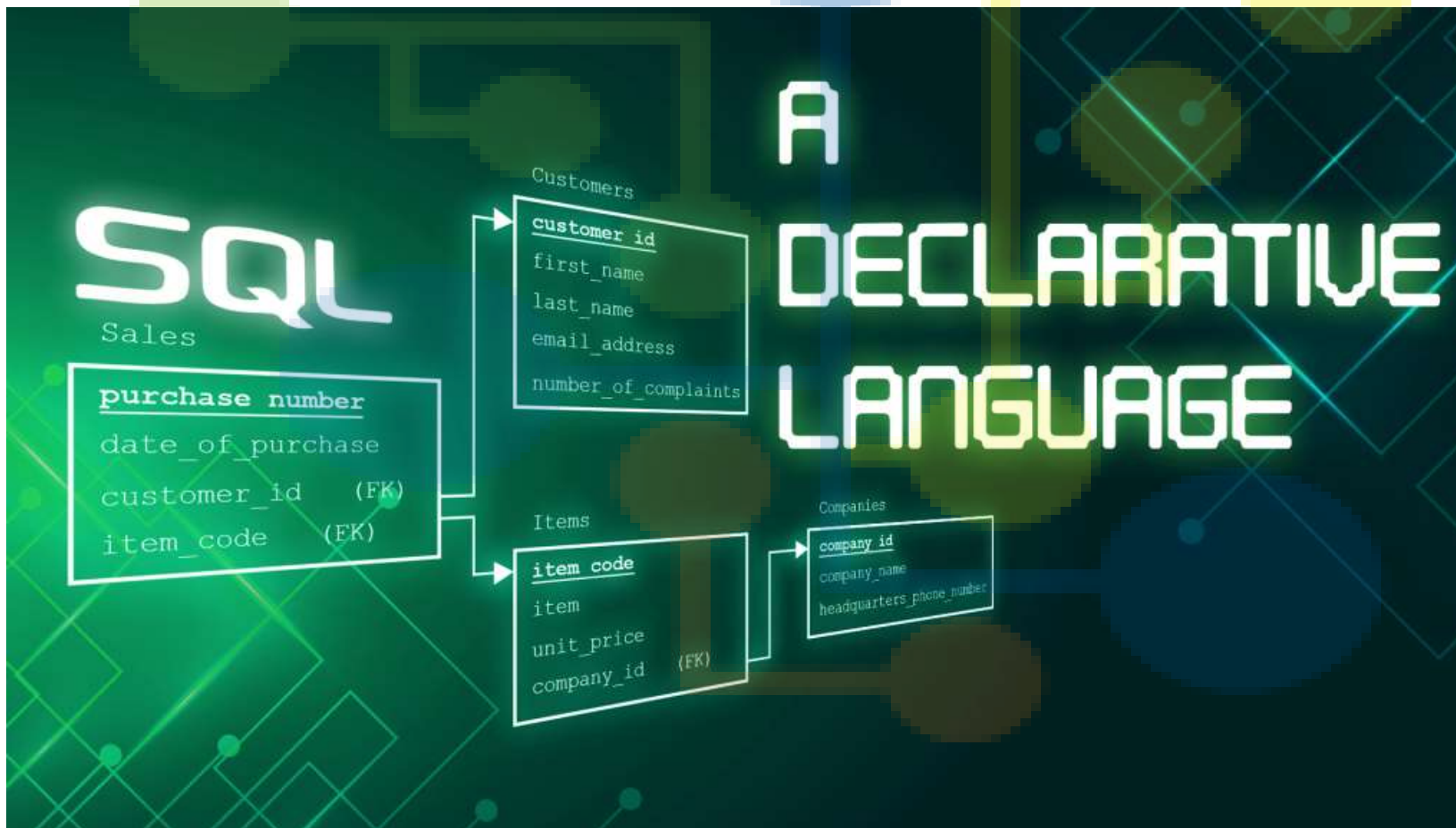
ANSI (American National Standards Institute)



Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Linguagem SQL

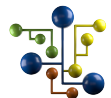




# Linguagem SQL

## Os comandos SQL se dividem em 3 tipos principais:

- Linguagem de Manipulação de Dados ou DML (Data Manipulation Language)
- Linguagem de Definição de Dados ou DDL (Data Definition Language)
- Linguagem de Controle de Dados ou DCL (Data Control Language)

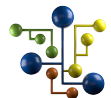


Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Linguagem SQL

**Linguagem de Consultas de Dados ou DQL (Data Query Language)**



# Linguagem SQL

## Instruções SELECT:

- **Cláusulas** (FROM, WHERE, GROUP BY, HAVING, ORDER BY, DISTINCT)
- **Operadores Lógicos** (AND, OR, NOT)
- **Operadores de Comparação** (<, >, <>, <=, =, >=, BETWEEN, LIKE)
- **Funções de Soma** (AVG, COUNT, SUM, MIN, MAX)



# Linguagem SQL

## Instruções DML:

**INSERT:** utilizado para inserir registros em uma tabela.

Exemplo: INSERT into CLIENTE(ID, NOME) values(1000,'Obama');

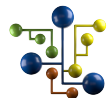
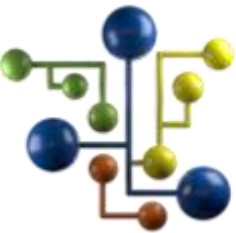
**UPDATE:** utilizado para alterar valores de uma ou mais linhas de uma tabela.

Exemplo: UPDATE CLIENTE set NOME = 'Angela' WHERE ID = 1000;

**DELETE:** utilizado para excluir um ou mais registros de uma tabela.

Exemplo: DELETE FROM CLIENTE WHERE ID = 1000;





# Linguagem SQL

## Instruções DDL:

**CREATE:** utilizado para criar objetos no banco de dados.

Exemplo (criar uma tabela):

CREATE TABLE FORNECEDOR ( ID INT PRIMARY KEY, NOME VARCHAR(50));

**ALTER:** utilizado para alterar a estrutura de um objeto.

Exemplo (adicionar uma coluna em uma tabela existente):

ALTER TABLE FORNECEDOR ADD TELEFONE CHAR(10);

**DROP:** utilizado para remover um objeto do banco de dados.

Exemplo (remover uma tabela):

DROP TABLE FORNECEDOR;



# Linguagem SQL

## Instruções DCL:

**GRANT:** autoriza um usuário a executar alguma operação.

Exemplo (dar permissão de consulta na tabela fornecedor para o usuário Obama):

GRANT select ON fornecedor TO obama;

**REVOKE:** restringe ou remove a permissão de um usuário executar alguma operação.

Exemplo (não permitir que o usuário obama crie tabelas no banco de dados):

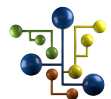
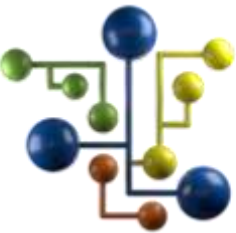
REVOKE CREATE TABLE FROM obama;



# Linguagem SQL

## Instruções DTL (Linguagem de controle de transações):

- BEGIN TRANSACTION
- COMMIT
- ROLLBACK



# Linguagem SQL

## Terminologia

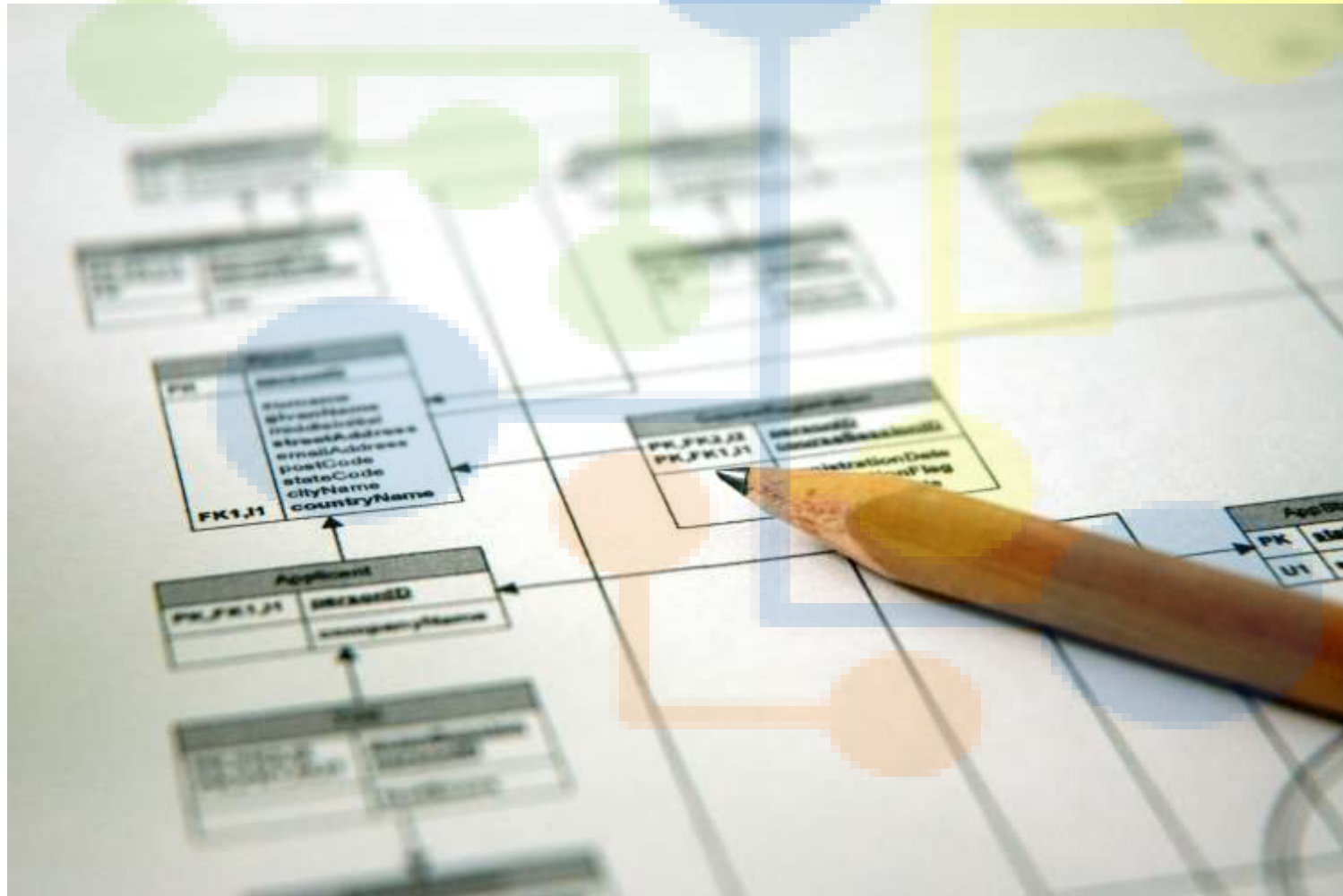
- Relação
- Tabela
- Coluna (Atributo)
- Linhas (registros)
- Chave Primária (Primary Key)



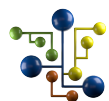
Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

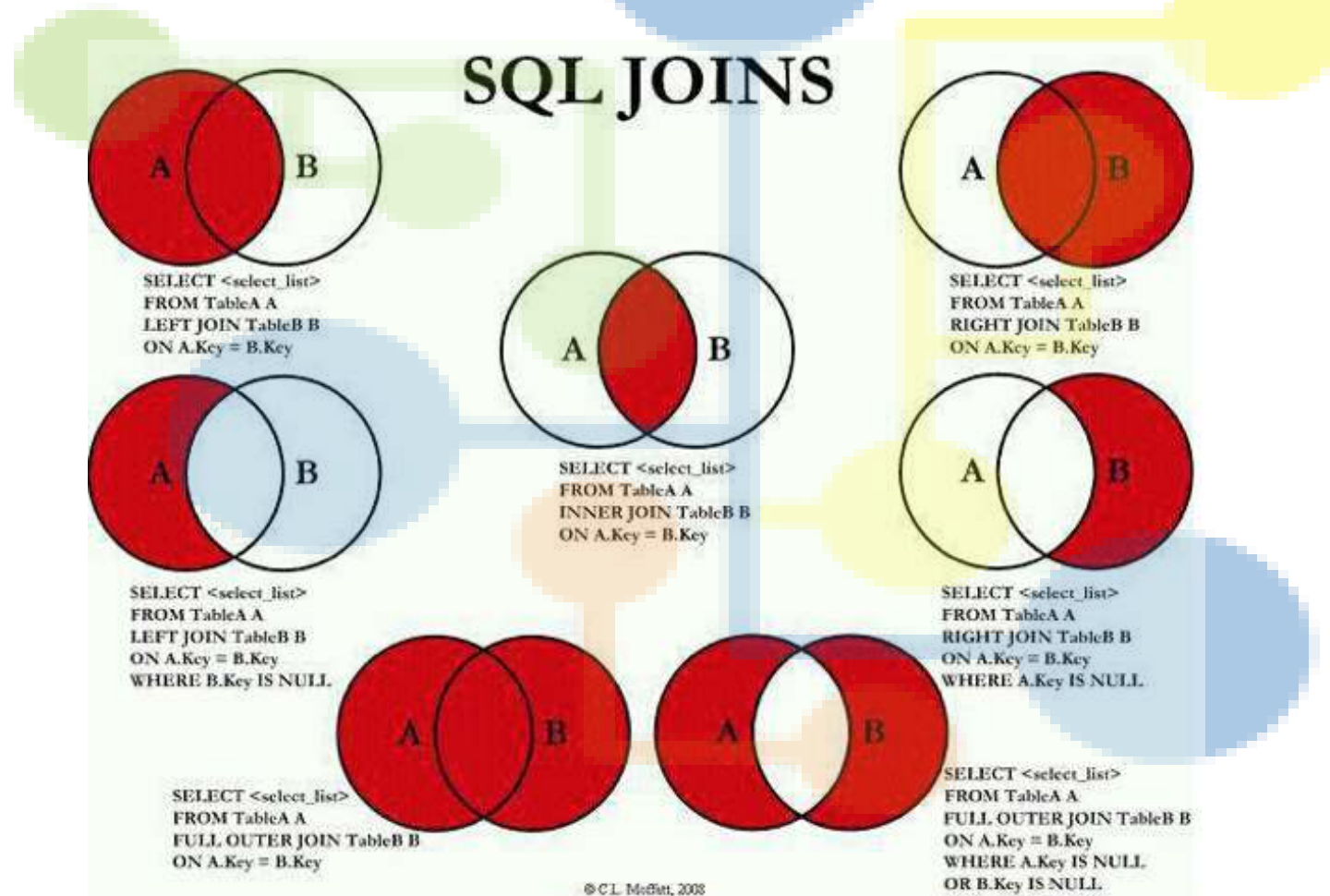
# Linguagem SQL







# Linguagem SQL





Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Big Data Real-Time Analytics com Python e Spark

## Apache Spark SQL

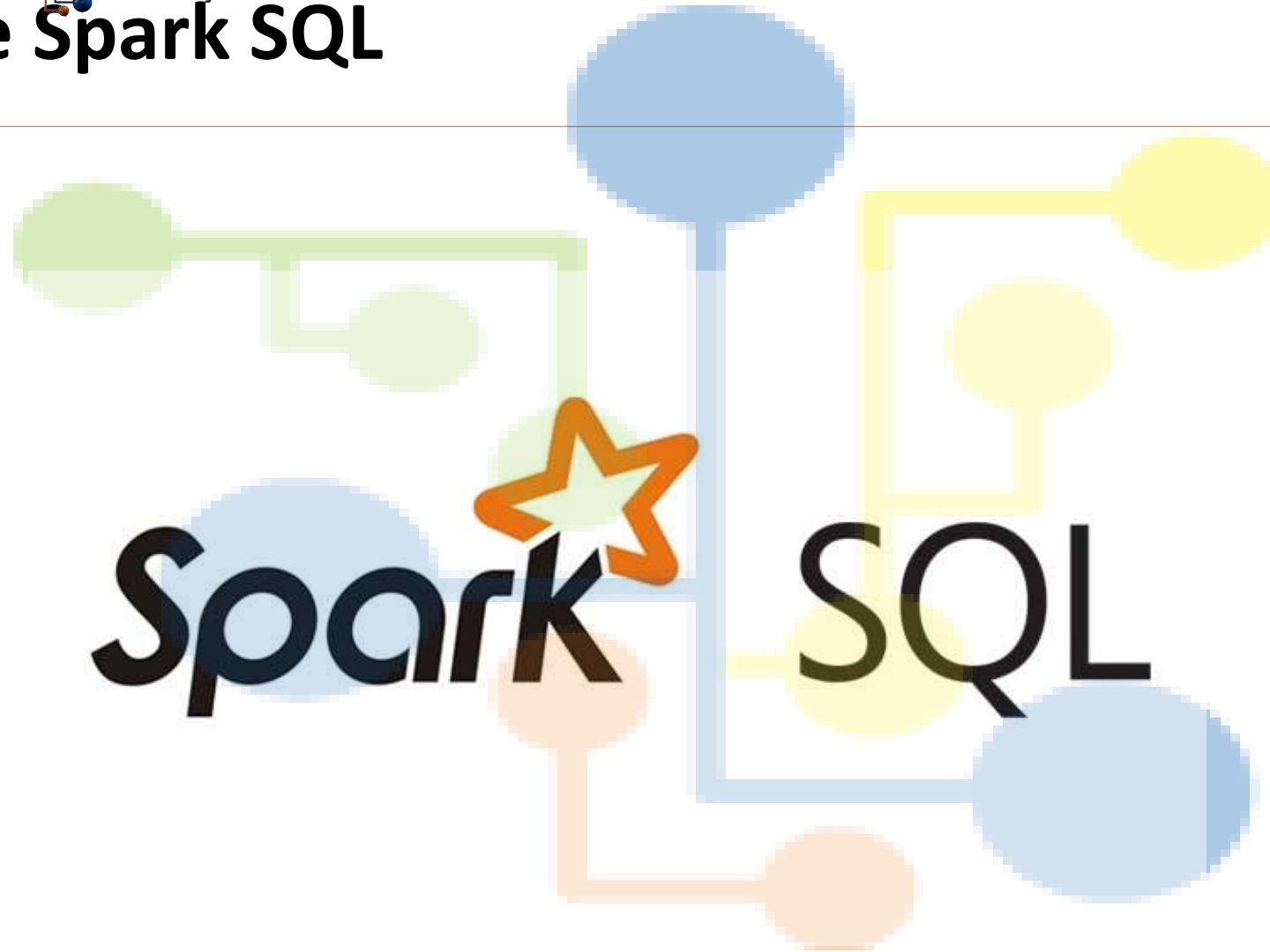




Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Apache Spark SQL





# Apache Spark SQL

## Principais Funcionalidades do Spark SQL:

- Dataframes
- API Data Source
- Servidor JDBC Interno
- Funcionalidades para Data Science



# Apache Spark SQL

## Componentes do Spark SQL

O Spark SQL possui 3 componentes principais:  
DataFrame, Spark Session e SQL Context.





# Apache Spark SQL

## Dataframe

- RDDs já existentes
- Arquivos de dados estruturados
- Conjunto de dados JSON
- Tabelas Hive
- Banco de dados externos
- Mais eficientes que os RDD's

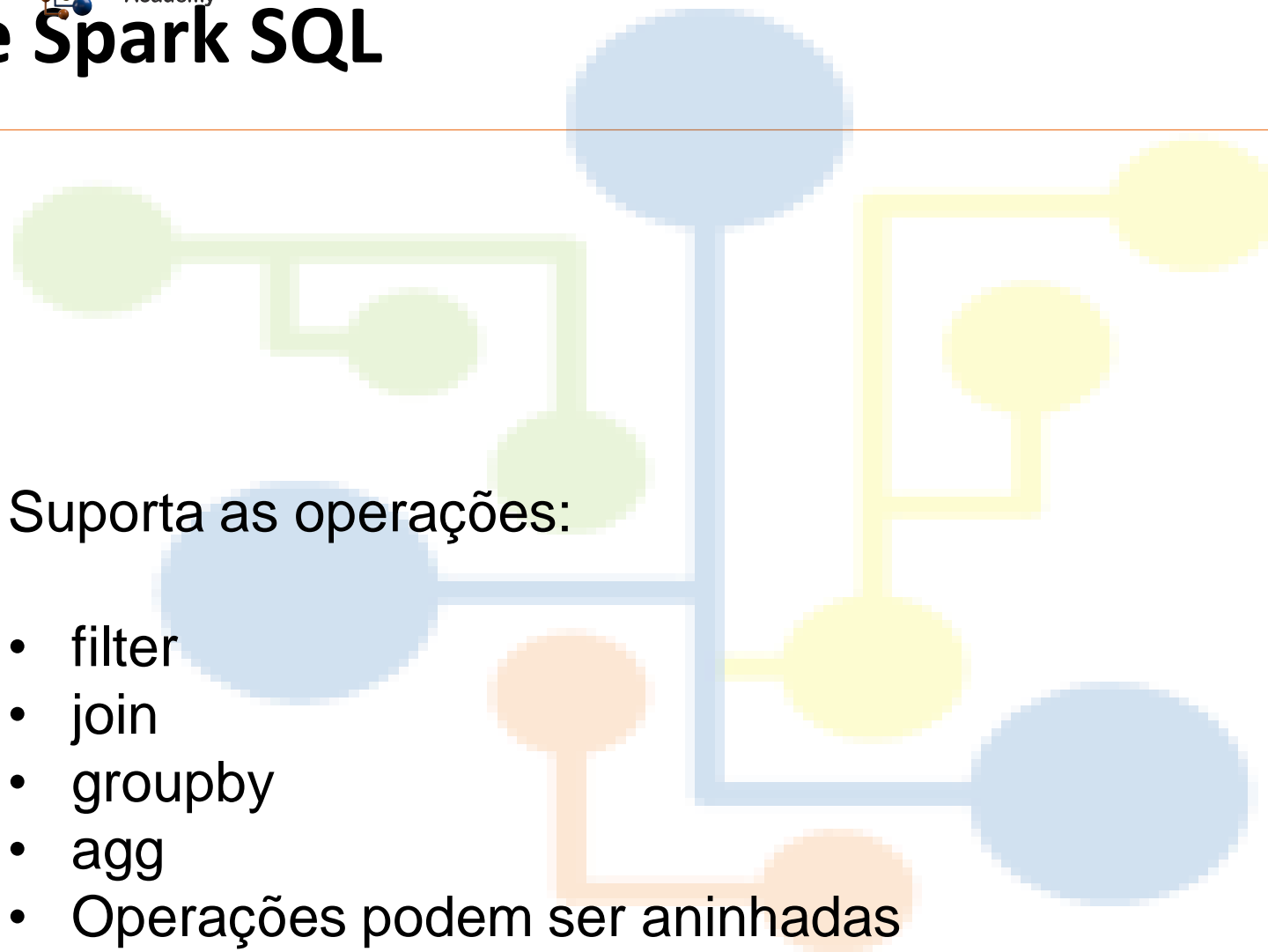


# Apache Spark SQL

## Dataframe

Suporta as operações:

- filter
- join
- groupby
- agg
- Operações podem ser aninhadas

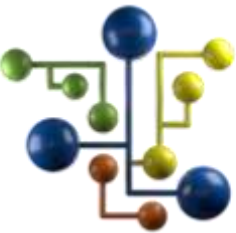




# Apache Spark SQL

## Spark Session

- Criamos um Spark Session para acessar as funcionalidades do Spark SQL.
- Dataframes são criados a partir de Spark Sessions.
- Permite registrar um dataframe como uma tabela temporária e executar queries SQL sobre ele (muito útil no processamento de streams de dados).



# Apache Spark SQL

## SQL Context

```
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
```



# Apache Spark SQL

## Fontes de Dados JDBC

**jdbc:mysql://localhost:3306/test**

API

Database

Server name  
on which  
MySQL is  
running

Port  
Number

database  
name





Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b

# Apache Spark SQL

## Tabelas Temporárias





# Apache Spark SQL

Uma das principais vantagens do Spark SQL é que você não precisa reaprender nada. Podemos usar os mesmos conceitos de SQL que usamos em bancos de dados relacionais, extrair dados para o Spark e então nos beneficiarmos do processamento paralelo e distribuído fornecido pelo framework Spark.



Data Science  
Academy

Data Science Academy marxv49@gmail.com 5e686b2be32fc3447a0e403b



Tenha uma Excelente Jornada de Aprendizagem.

Muito Obrigado por Participar!

Equipe Data Science Academy