FIAP GRADUAÇÃO

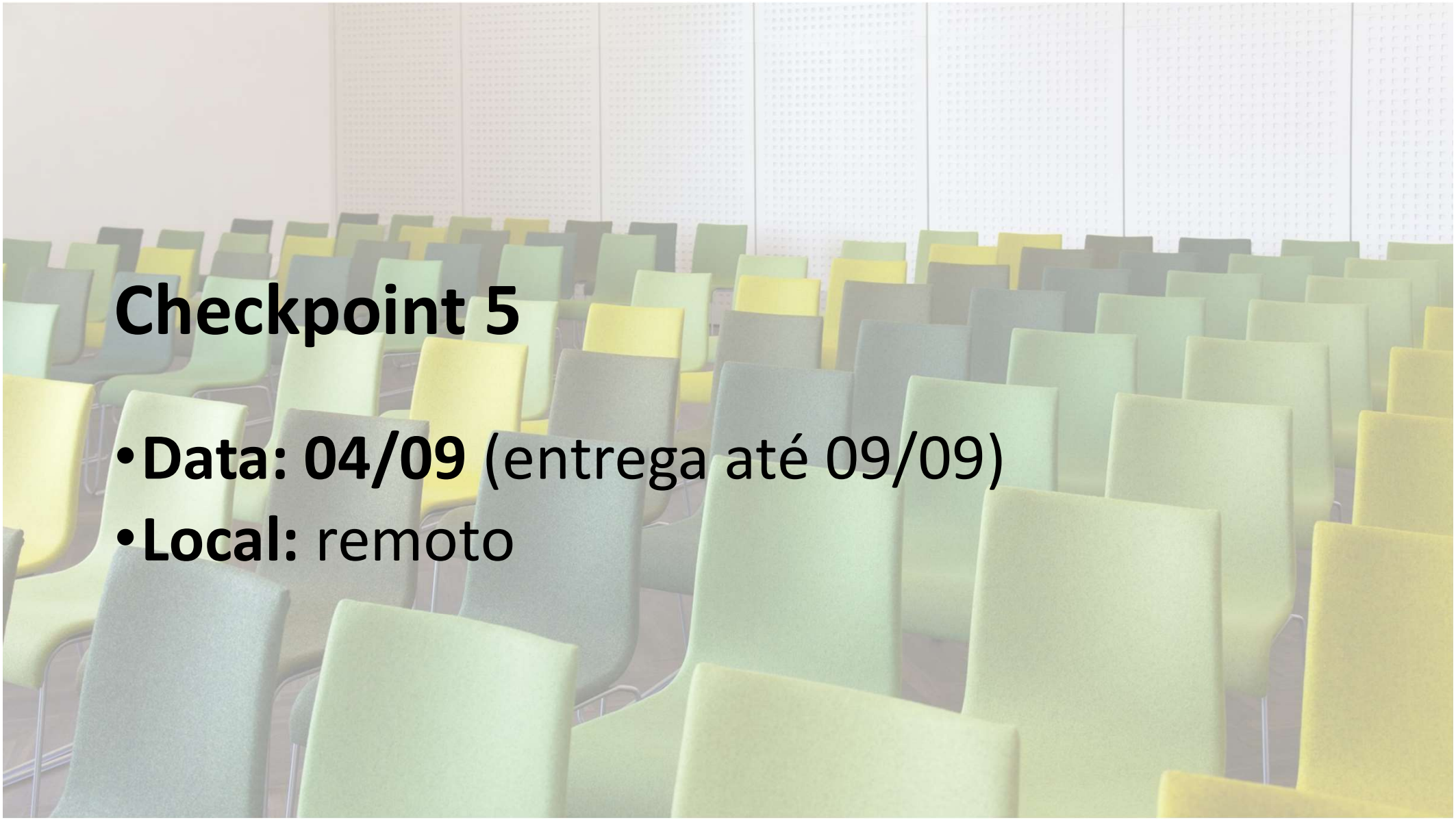
FIAP

DATA SCIENCE

BIG DATA ARCHITECTURING & DATA INTEGRATION

Prof. Dr. Renê de Ávila Mendes

Material cedido pelo Prof. Milton Goya



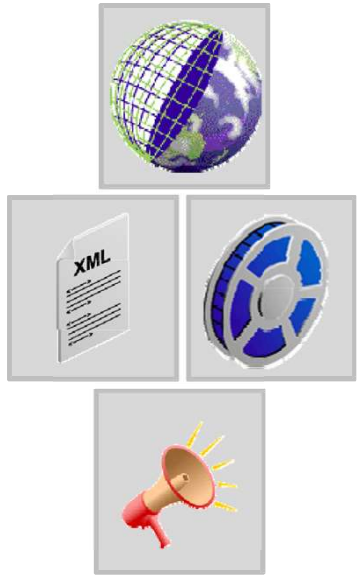
Checkpoint 5

- **Data: 04/09** (entrega até 09/09)
- **Local:** remoto

Hive

Aplicações Baseadas em Hive

- Processamento de Logs
- Mineração de texto
- Indexação de documentos
- Business analytics
- Modelos Preditivos



O que é Hive

- Apache Hive foi inicialmente desenvolvido pelo Facebook em 2007 para ajudar a empresa a gerir a grande quantidade de dados gerada.
- Naquela época, o Facebook tinha 15TB.
- Poucos anos depois este voluma já tinha crescido para 700TB.

O que é Hive

- O seu Data Warehouse estava demorando demais para processar as consultas diárias e por isso eles decidiram mover os dados para o Hadoop.
- No entanto, criar tarefas MapReduce não era fácil e consumia muito tempo dos seus funcionários.
- Assim, eles decidiram criar o Hive com os conceitos já familiares dos bancos de dados.

O que é Hive

- É um data warehouse escalável e tolerante a falhas construído em cima do Apache Hadoop.
- Suporta um grande volume de dados que cresce rapidamente.
- É um sistema para processamento em batch.
- Possui uma linguagem de consulta chamada HiveQL (Hive query language) que é similar ao SQL

O que NÃO é Hive

- Não é:
  - Um banco de dados relacional
  - Um projeto para Online Transaction Processing (OLTP)
  - Uma linguagem para consultas em tempo real e atualizações de linhas

FIAP

Quando utilizar o Hive

- Quando já se sabe SQL
  - É fácil utilizar o Hive quando já se tem conhecimento da linguagem SQL.

- Compatível com outras ferramentas do ecossistema Hadoop
  - O Hive é fácil de utilizar e aprender e já vem sendo utilizado há um bom tempo. Isso faz com que outras ferramentas do ecossistema Hadoop permitam a utlização do Hive como umas das opções de armazenamento

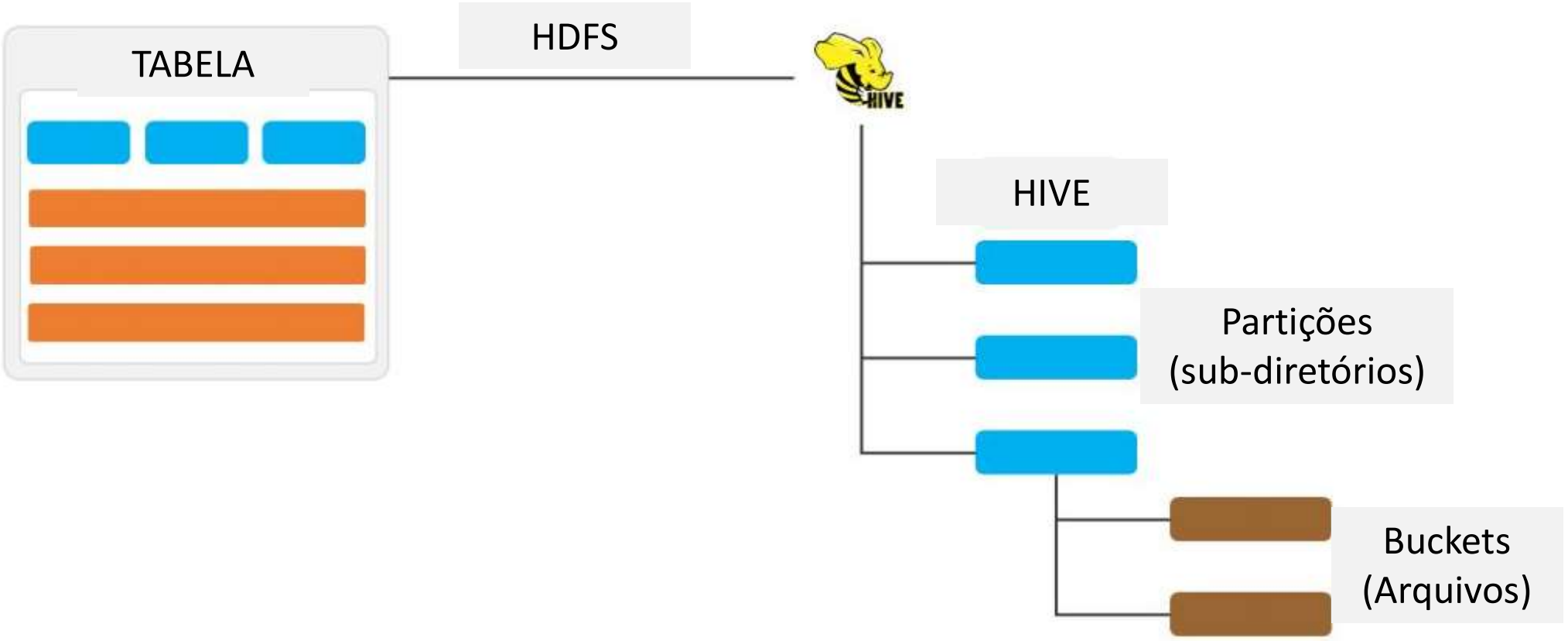
FIAP

Quando utilizar o Hive

- Dados estruturados
  - O Hive é uma boa opção para os dados estruturados em forma de tabelas. Dados semiestruturados como XML precisam ser transformados para utilização no Hive.

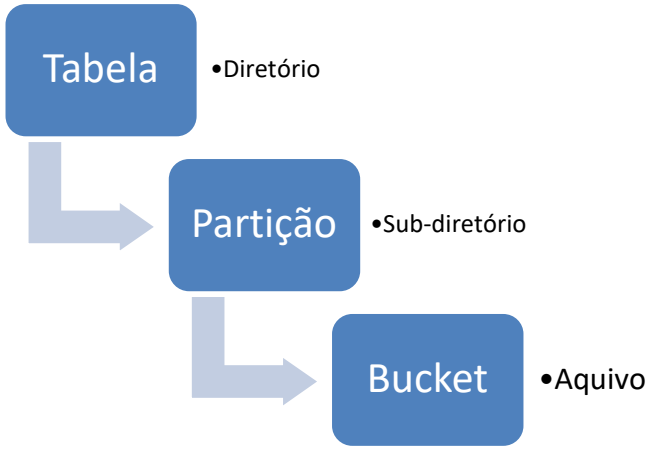
- Processamento em batch
  - Hive é um sistema para processamento em batch e pode fazer consultas sobre um volume de dados na ordem de petabytres. O Hive não foi projetado para atualização de registros únicos e também não é capaz de responder rapidamente a consultas assim como fazem muitos bancos de dados relacionais.

Modelo de Dados do Hive



Modelo de Dados do Hive

- Os dados no Hive são organizados em:
  - Tabelas
    - Partições
      - Buckets

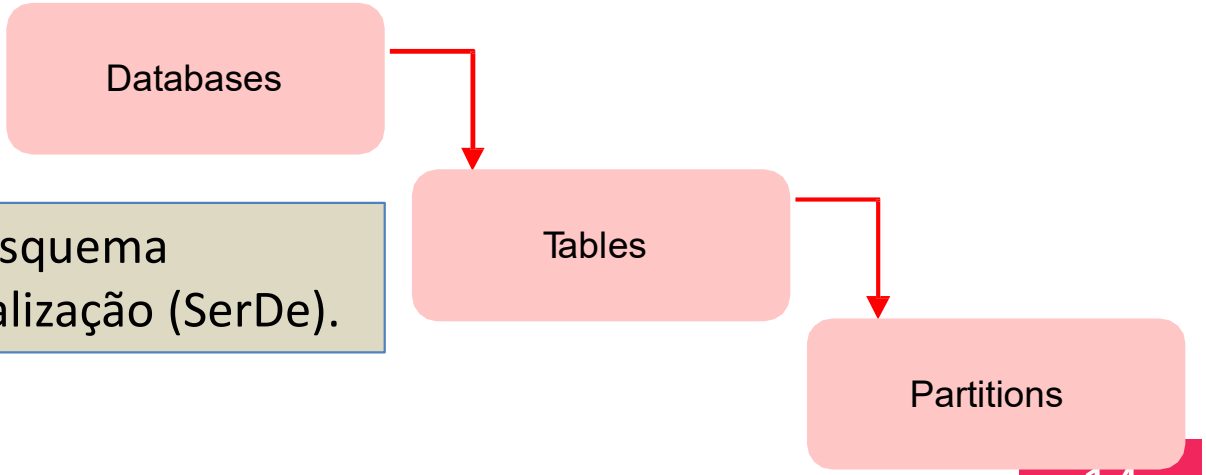


- O layout físico do Hive é um diretório do warehouse no HDFS que é dividido em tabelas e partições.

Hive: unidades de dados

FIAP

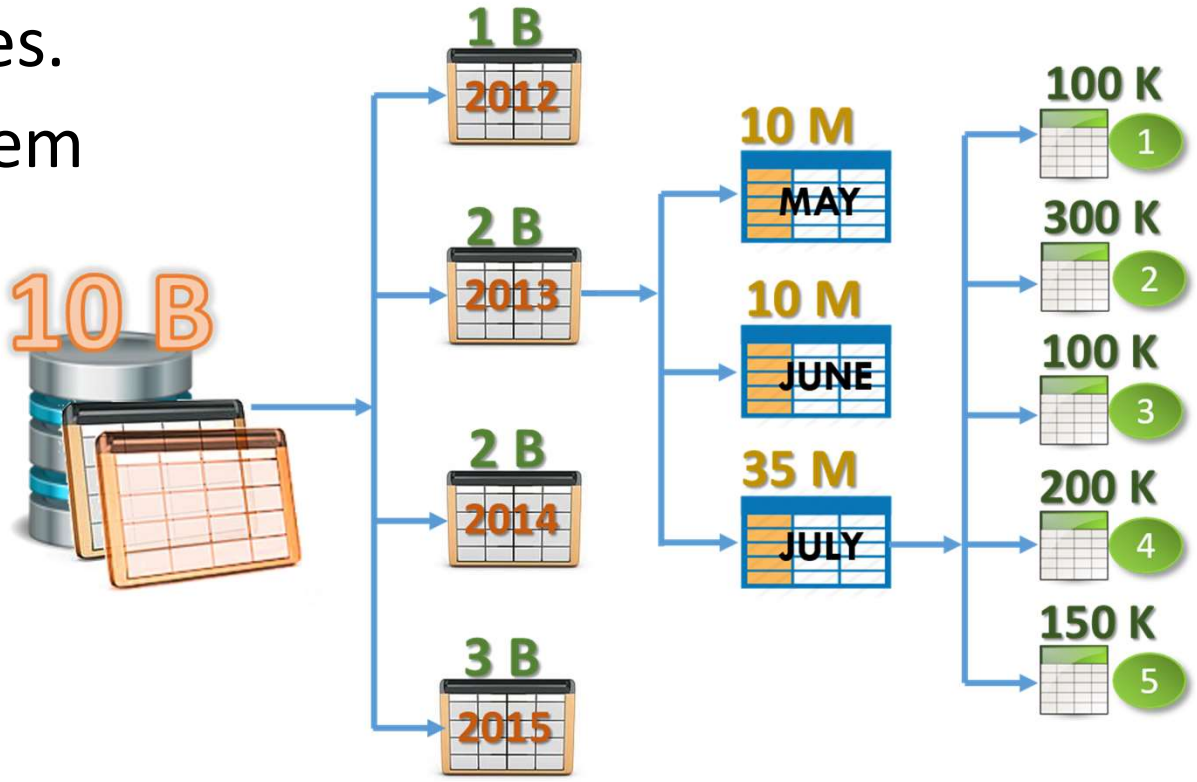
- As tabelas do Hive: o mesmo de um RDBMS
  - O Hive fica no topo do HDFS, as tabelas são mapeadas para diretórios no sistema de arquivos HDFS.
- Cada tabela tem um diretório no HDFS.
- Os dados são serializados e armazenados como arquivos nesse diretório.
- Possui um mecanismo próprio para serialização e desserilização.



Os usuários podem especificar um esquema específico de SERialização – Desserialização (SerDe).

Unidade de Dados

- Partições
  - Cada tabela pode ser dividida em partições.
  - As partições permitem a distribuição de dados dentro de subdiretórios
  - As partições são mapeadas para subdiretórios no sistema de arquivos subjacente.



Unidade de Dados

- Partições

```
CREATE_TABLE Vendas
             (vendas_id INT,
              quantidade FLOAT)
PARTITIONED BY (UF STRING,
                ano INT,
                mes INT)
```

  - Nesse caso, cada partição será dividida em subpartições denominadas de **buckets**.
    - Vendas / UF=MG / ano=2020 / mes = 12

Unidade de Dados

- Buckets
  - Os dados de cada partição são divididos em **buckets**.
  - A divisão dos dados é feita por meio de uma função *hash* da coluna particionada.
    - $MOD(Hash(coluna), NumBuckets) = Número\ do\ Bucket$
  - Cada **bucket** é armazenado em um arquivo no diretório particionado.

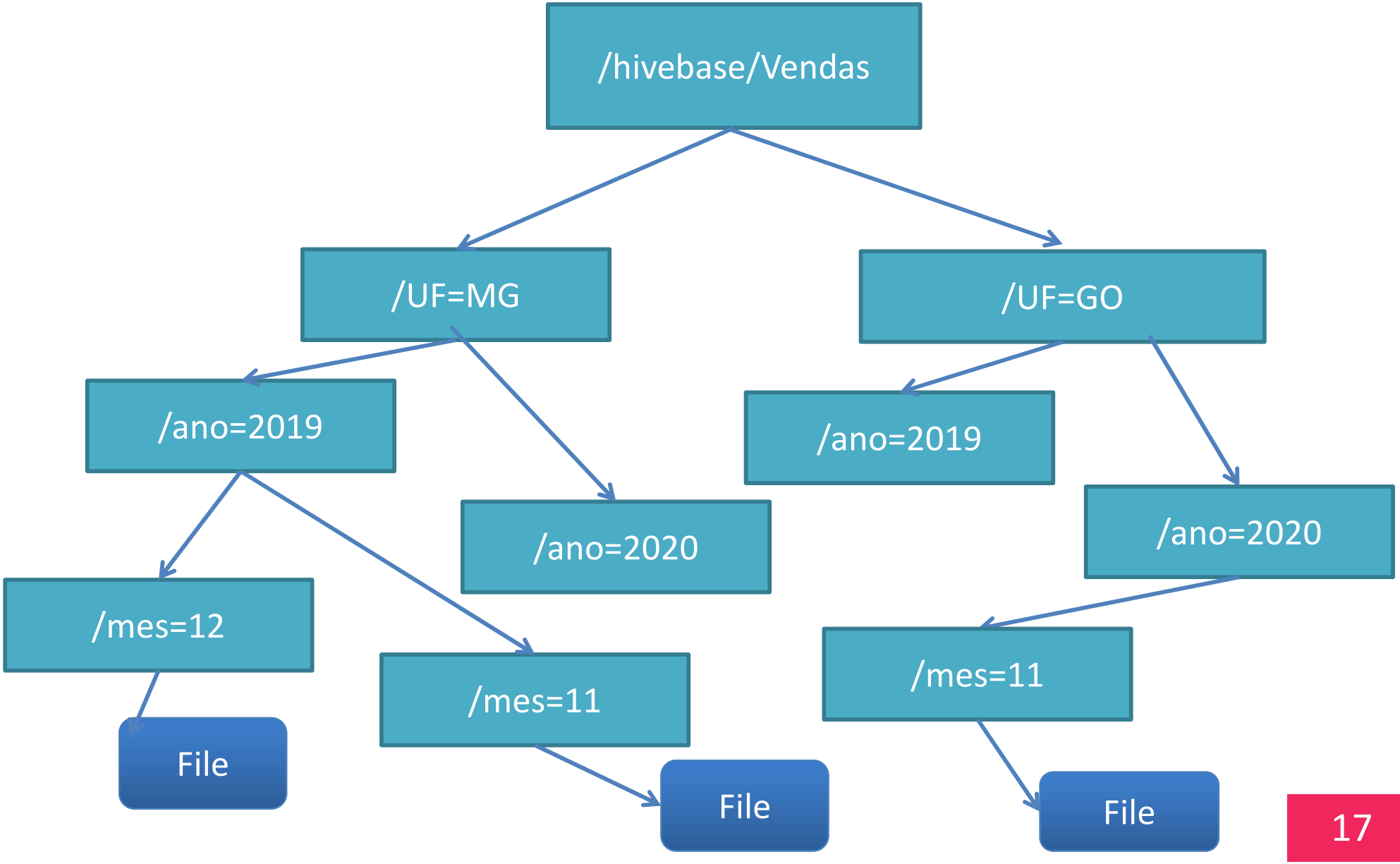
Compactação de Partição

Compactação de partição: apenas as partições relevantes são acessadas.

```
SQL> SELECT SUM(sales_amount)
  2  FROM sales
  3  WHERE sales_date BETWEEN
  4  TO_DATE('01-MAR-1999',
  5          'DD-MON-YYYY') AND
  6  TO_DATE('31-MAY-1999',
  7          'DD-MON-YYYY');
```

Exemplo de particionamento em Oracle SQL.

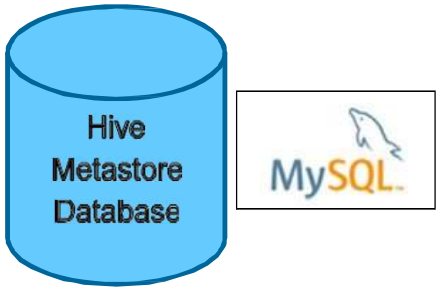
Hierarquia das Partições HIVE



FIAP

Hive Metastore

- Contém metadados sobre bancos de dados, tabelas e partições
- Contém informações sobre como as linhas e colunas são delimitadas nos arquivos HDFS usados nas consultas
- Age como um banco de dados RDBMS em que o Hive persiste esquemas de tabelas e outros metadados do sistema



DataTypes

Numéricos
Tinyint
Smallint
Int/Integer
BigInt
Float
Double
Decimal/Numeric

Date/Time
Timestamp
Date
Interval

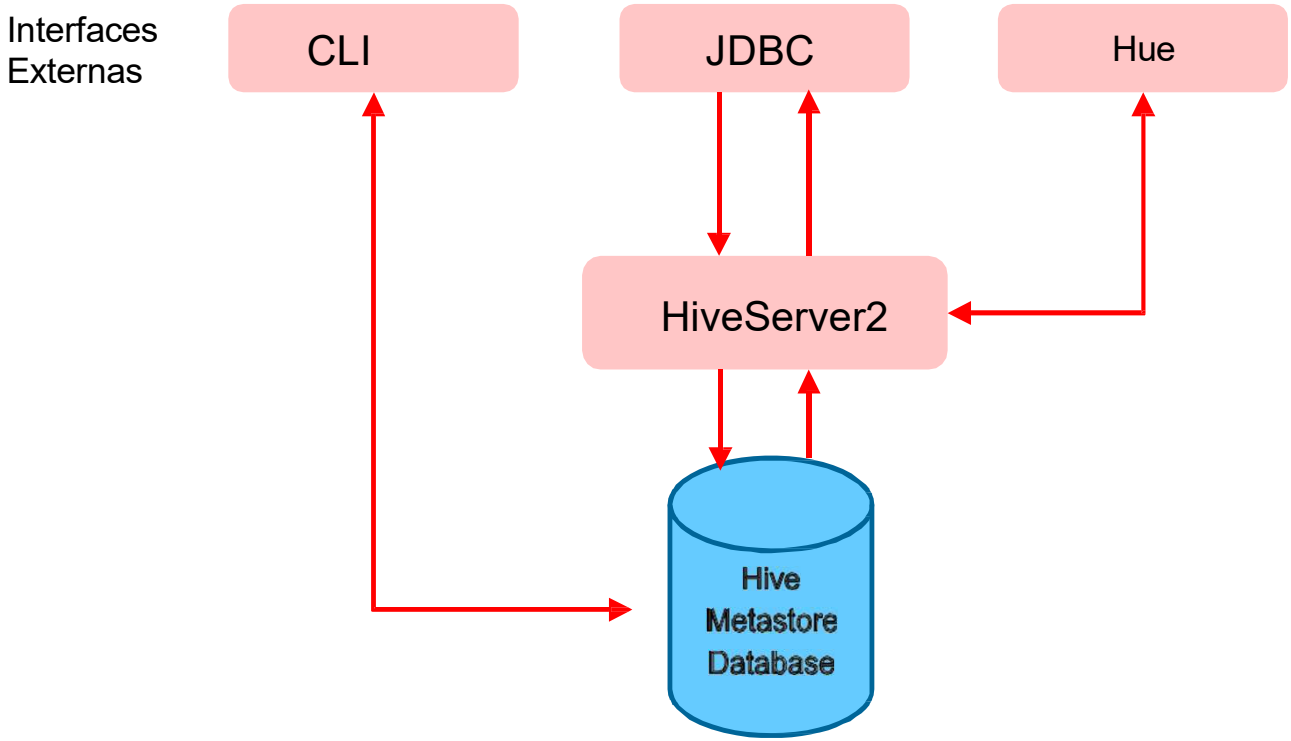
Texto
String
Varchar
Char

Miscelânia
Boolean
Binary
NULL

Complexos
Arrays
Maps
Structs
Union

Fonte: IBM

Framework do Hive



https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL#LanguageManual%20%20DDL-RowFormat,StorageFormat,andSerDe

Criando um Banco de Dados Hive

1. Inicie o Hive

```
[oracle@localhost mapreduce]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
Hive history file=/tmp/oracle/hive_job_log_oracle_201302071749_169058549.txt
hive>
```

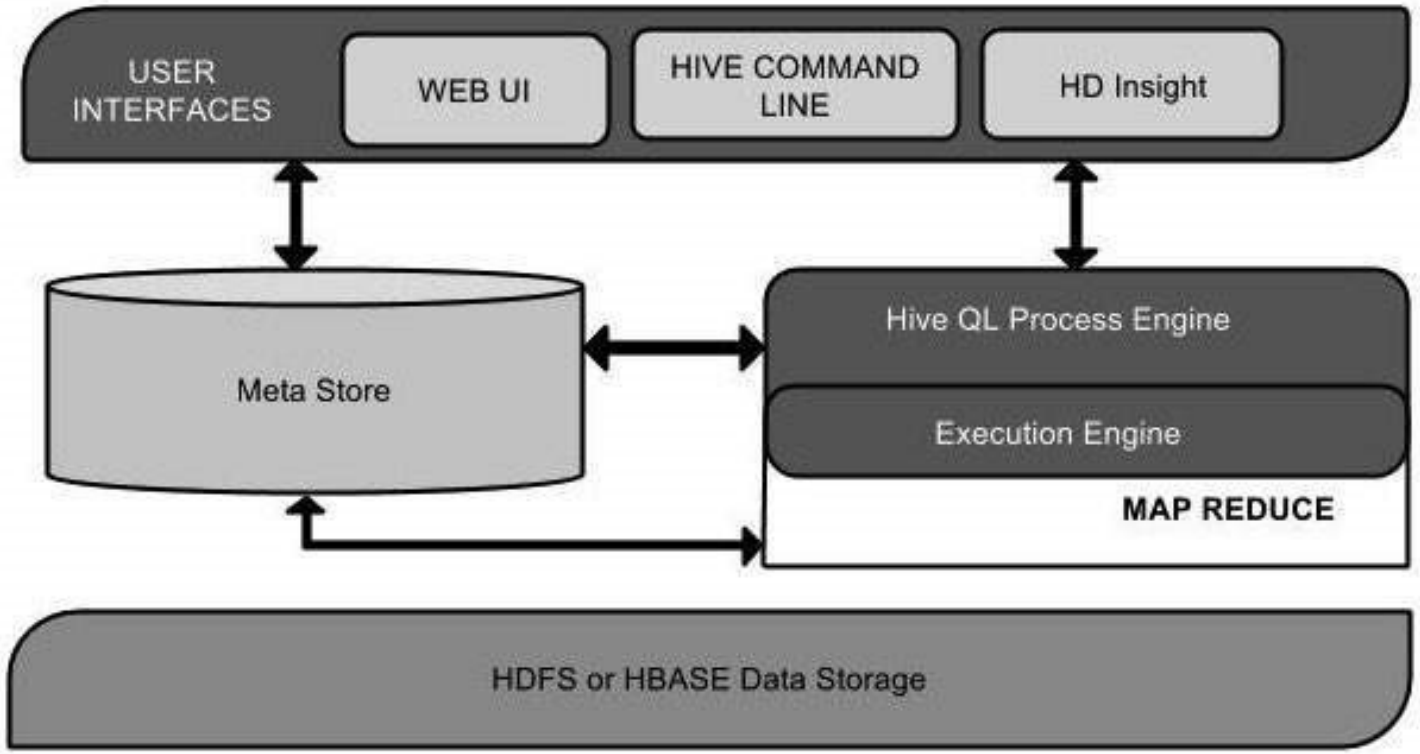
```
hive> create database moviework;
OK
Time taken: 4.288 seconds
hive>
```

2. Crie o banco de Dados

```
hive> show databases;
OK
default
moviedemo
moviework
Time taken: 1.281 seconds
hive>
```

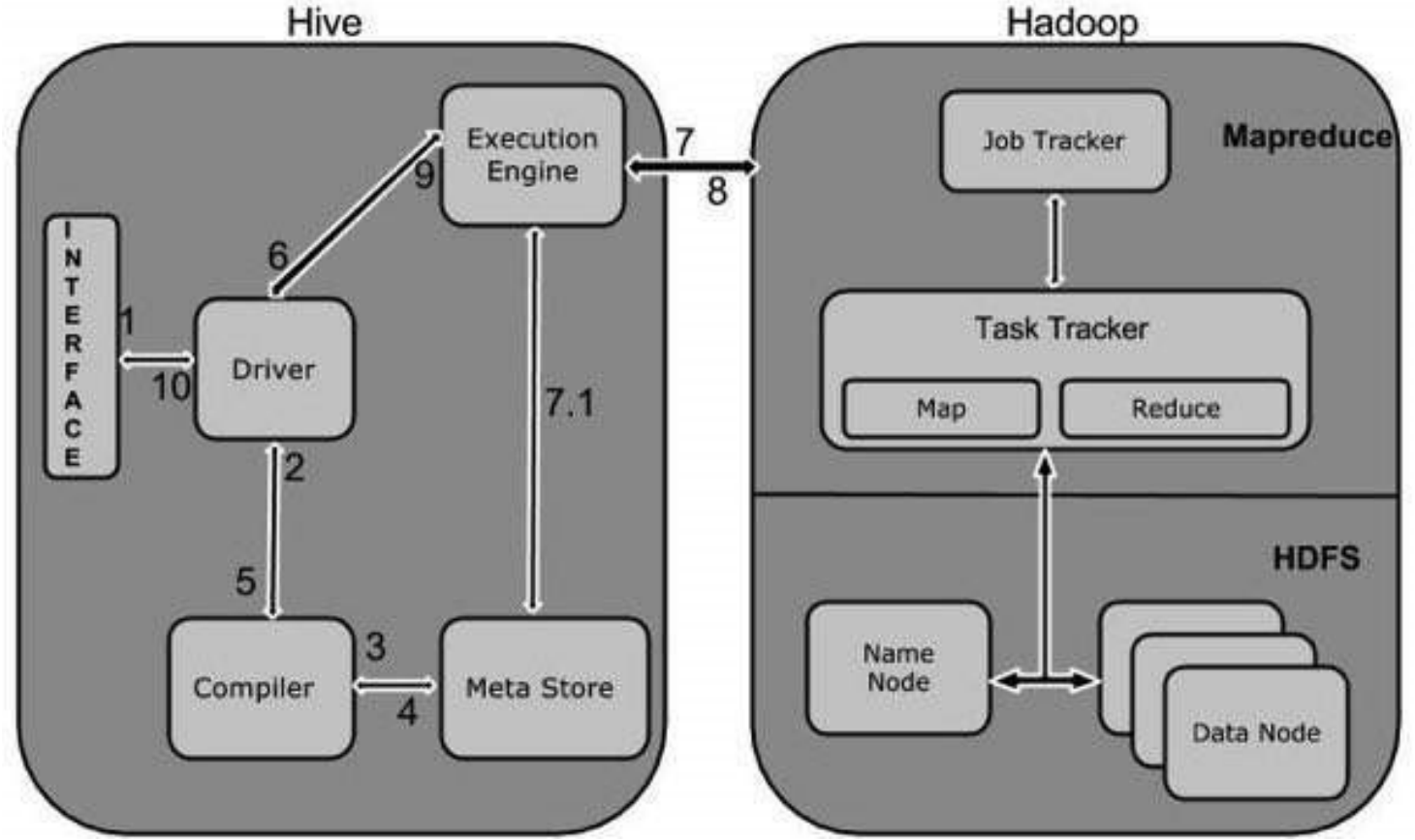
3. Verifique a criação do banco de dados

Arquitetura



Fonte: IBM

Fluxo de Trabalho entre Hive e Hadoop



Fonte: IBM

Create Table - Sintaxe

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name

[(col_name data_type [COMMENT col_comment], ...)]
[COMMENT table_comment]
[ROW FORMAT row_format]
[STORED AS file_format]
```

Fonte: IBM

Create Table - Exemplo

```
hive> CREATE TABLE IF NOT EXISTS employee ( eid int, name String,
salary String, destination String)
COMMENT 'Employee details'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

- A tabela não será criada se ela já existir e opção IF NOT EXISTS for usada.
- COMMENT indica um comentário para a tabela
- ROW FORMAT indica o formato da linha.
- No exemplo, cada campo e linha tem seu terminador e a serão armazenados como um arquivo de texto

Fonte: IBM

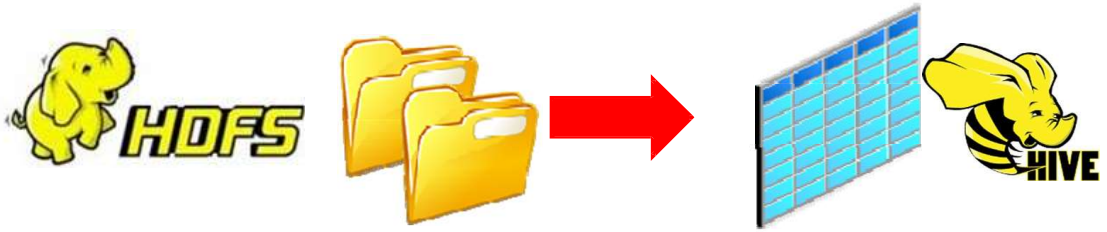
Criando uma tabela com dados JSON

```
22
23 -- Create table over source JSON
24 CREATE EXTERNAL TABLE IF NOT EXISTS movieapp_log_json (
25   custId INT,
26   movieId INT,
27   genreId INT,
28   time STRING,
29   recommended STRING,
30   activity INT,
31   rating INT,
32   price FLOAT
33 )
34 ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.JsonSerde'
35 LOCATION '/user/oracle/moviedemo/applog/';
```

Sintaxe SQL

Opções SerDe

- Uma tabela HIVE é mapeada para um diretório HDFS.



Exemplo

```
CREATE TABLE
       IF NOT EXISTS population
(ano INT,
 idade INT,
 sexo INT,
 qtd  INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

Insert

FIAP

- No HIVE os dados são inseridos massivamente através da instrução LOAD DATA.
- Os dados podem ser inseridos a partir de um arquivo armazenados localmente ou no Hadoop.
- Sintaxe:

```
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename
[PARTITION (partcol1=val1, partcol2=val2 ...)]
```

  - LOCAL indica o caminho do arquivo local (opcional)
  - OVERWRITE indica que os dados serão substituídos na tabela (opcional)
  - PARTITION indica as partições que serão usadas (opcional)

Fonte: IBM

Insert

FIAP

- Exemplo:
  - Os dados estão em um arquivo denominado **`sample.txt`** no diretório **`/home/user`**:

```
hive> LOAD DATA LOCAL INPATH '/home/user/sample.txt'
OVERWRITE INTO TABLE employee;
```

Fonte: IBM

Insert

FIAP

- Exemplo:

```
INSERT INTO TABLE students
  VALUES ('fred flintstone', 35, 1.28),
         ('barney rubble', 32, 2.32);
```

Fonte: IBM

Exemplo

```
LOAD DATA LOCAL INPATH
'/home/oracle/Downloads/population.csv'
OVERWRITE INTO TABLE population;
```

ALTER TABLE

```
ALTER TABLE name RENAME TO new_name
ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])
ALTER TABLE name DROP [COLUMN] column_name
ALTER TABLE name CHANGE column_name new_name new_type
ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])
```

```
hive> ALTER TABLE employee RENAME TO emp;
```

Fonte: IBM

DROP TABLE

FIAP

```
DROP TABLE [IF EXISTS] table_name;
```

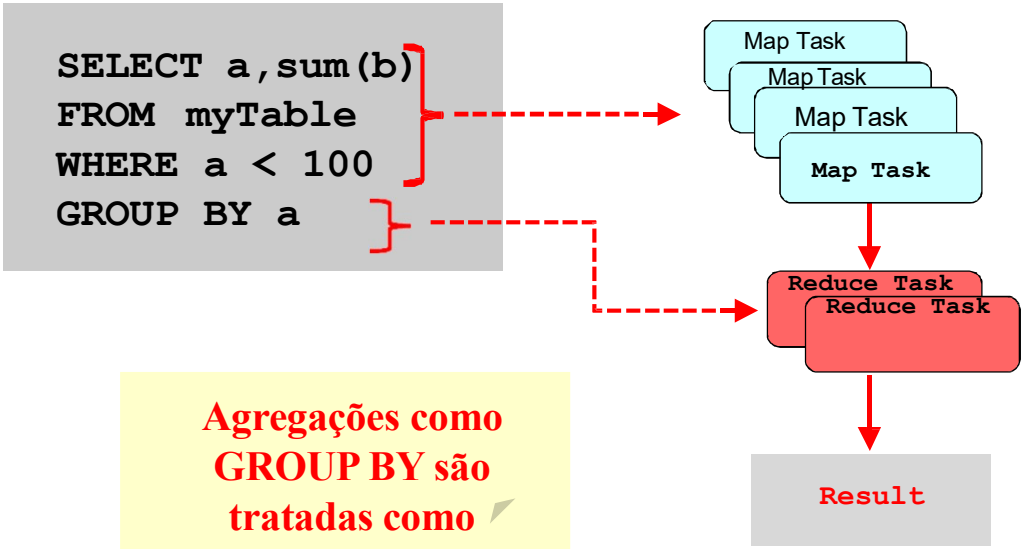
```
hive> DROP TABLE IF EXISTS employee;
```

- Elimina os dados e metadados da tabela.

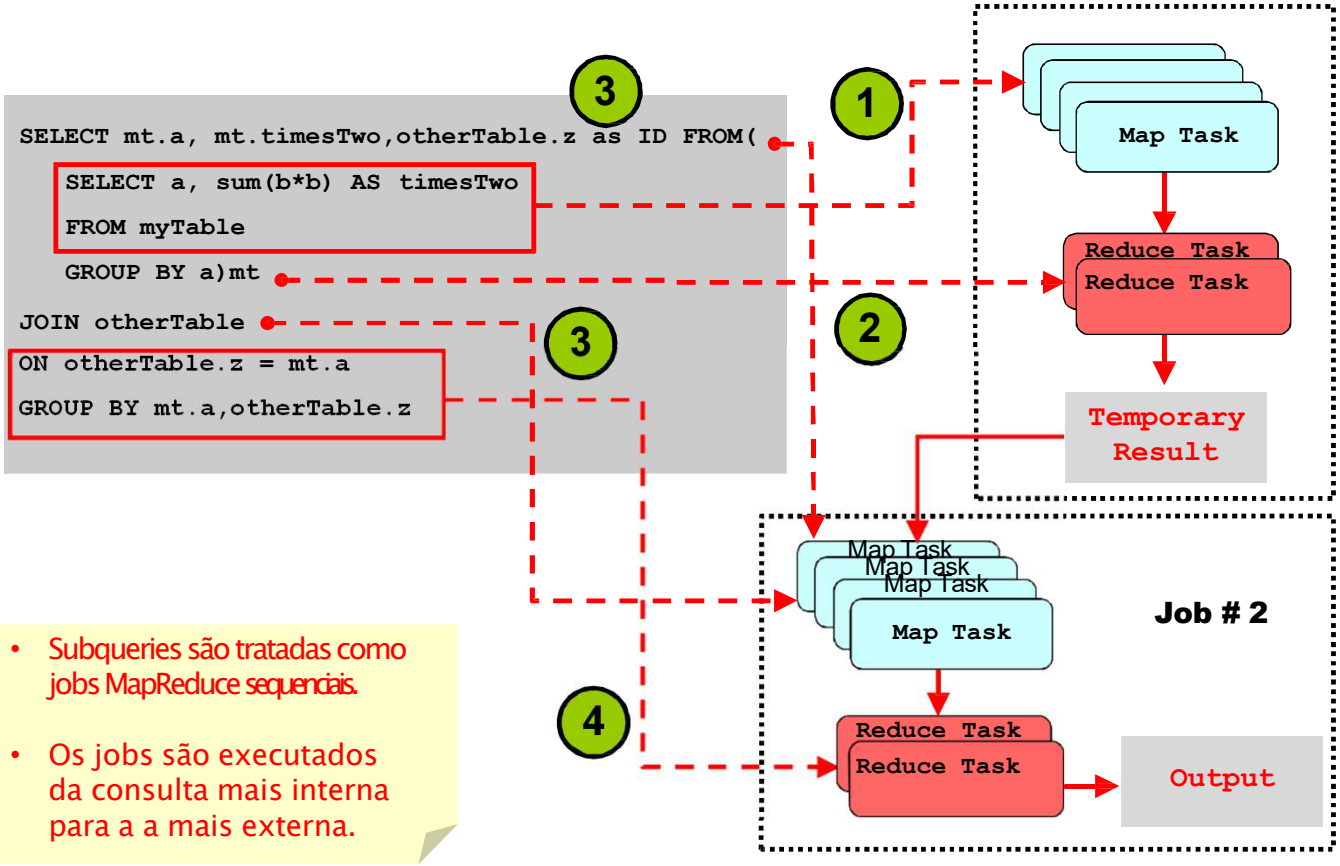
Fonte: IBM

Manipulação de dados no Hive

`SELECT` Hive com a cláusula `WHERE`:



Manipulação de dados no Hive: consultas aninhadas



- Subqueries são tratadas como jobs MapReduce sequenciais.
- Os jobs são executados da consulta mais interna para a a mais externa.

SELECT

FIAP

```
SELECT [ALL | DISTINCT] select_expr, select_expr, ...
FROM table_reference
[WHERE where_condition]
[GROUP BY col_list]
[HAVING having_condition]
[CLUSTER BY col_list | [DISTRIBUTE BY col_list] [SORT BY col_list]]
[LIMIT number];
```

```
hive> SELECT * FROM employee WHERE salary>30000;
```

```
hive> SELECT Id, Name, Dept FROM employee ORDER BY DEPT;
```

```
hive> SELECT Dept,count(*) FROM employee GROUP BY DEPT;
```

```
hive> SELECT c.ID, c.NAME, c.AGE, o.AMOUNT
FROM CUSTOMERS c JOIN ORDERS o
ON (c.ID = o.CUSTOMER_ID);
```

Fonte: IBM

Exemplo

```
SELECT *
  FROM population;

SELECT *
  FROM population
 WHERE ano > 1990;

SELECT ano, SUM(qtd)
  FROM population
 GROUP BY ano;
```

HIVE e PIG

- Semelhanças
  - Ambas são linguagens de alto nível que trabalham no topo de uma estrutura MapReduce.
  - Ambas usam a estrutura do HDFS e podem coexistir.

HIVE e PIG

- Diferenças
  - Language
    - Pig — é procedural ; (A = load ‘mydata’; dump A)
    - Hive — é declarative; (select * from A)
  - Atividade
    - Pig – melhor utilizado em pesquisas *ad hoc* (análises sob demanda)
    - Hive – melhor utilizado na geração de relatórios (relatório semana de BI).

HIVE e PIG

- Diferenças
  - Usuários
    - Pig – Programadores e desenvolvedores
    - Hive – Business Analytics
  - Necessidade do usuário
    - Pig – ambiente de desenvolvimento e debug mais desenvolvido
    - Hive – integração com outras ferramentas via JDBC e ODBC

Limitações do Hive

- Não permite materialized views.
- Não suporta transações.
- Não foi criado para pesquisas ad hoc.
- Suporte limitado a subconsultas.
- Usa um subconjunto do SQL-92.
- Otimizador embrionário.