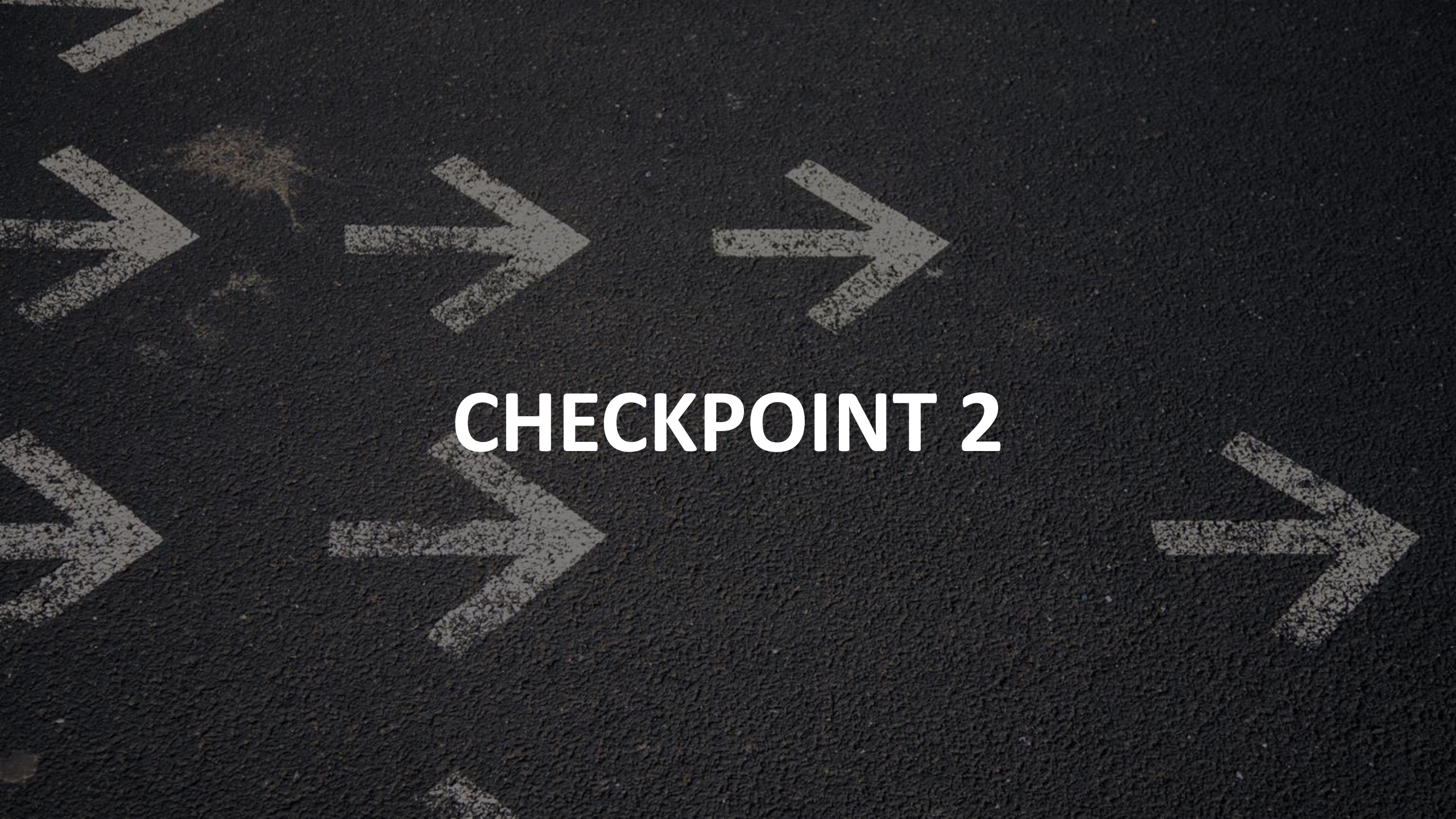
FIAP GRADUAÇÃO

FIAP

DATA SCIENCE

BIG DATA ARCHITECTURING & DATA INTEGRATION

Prof. Dr. Renê de Ávila Mendes


CHECKPOINT 2

Checkpoint 2

- **Data:** 24/04
- **Local:** em sala de aula

Objetivos da disciplina

DISCIPLINA: Big Data Architecturing & Data Integration

OBJETIVOS: Entenda as principais **arquiteturas** para ingestão, processamento e análise de grandes volumes de dados. Conheça as principais **ferramentas** open-source de Big Data como Hadoop, MapReduce, Spark, Sqoop, NiFi, Flume, Kafka, Zookeeper, HBase, Hive e as **integre** com as ferramentas de **extração, transformação e carga** de dados em modelos dimensionais. Entenda conceitos sobre **computação paralela e distribuída**, aplicação do Hadoop e bases Apache e arquiteturas *serverless* e desacopladas. Veja como **visualizar os dados** estruturados ou não estruturados com ferramentas de Self-Service Business Intelligence como PowerBI, utilizando as melhores práticas de **visualização de dados**.

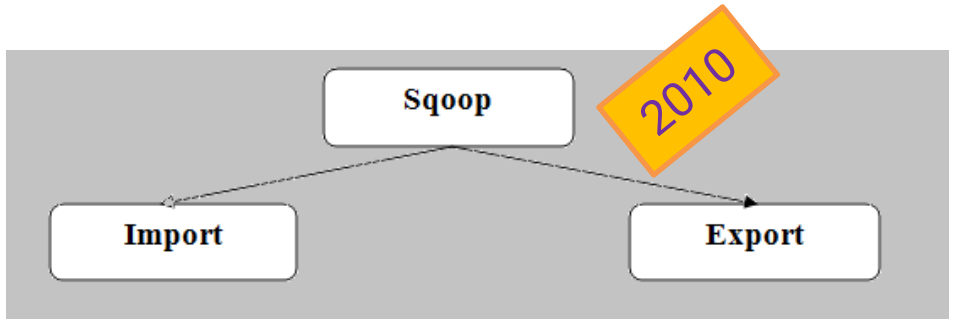
Assuntos – 1º Semestre

- Introdução a Big Data
- Conceitos Computação Paralela e Distribuída, Lei de Moore, Sistema HDFS
- Aplicação Hadoop. Usos e Administração de Ambientes Hadoop
- Introdução a MapReduce
- **Introdução à Integração de Dados**
- **Integração entre SQL e Hadoop - SQOOP**
- Bases Apache
- Bases Apache - PIG
- Introdução da Data Streaming - FLUME
- Introdução a análise de dados com SPARK

Sqoop

SQOOP

Apache Sqoop



- O Sqoop Apache Project é um utilitário de software livre de movimentação de dados de Hadoop para banco de dados baseado em JDBC.
- Sqoop foi criado em 2010 em um hackathon na Cloudera.
- Em 2011 foi disponibilizado como software livre.
- Em 2012 tornou-se um dos projetos top-level da Apache Software Foundation.

Apache Sqoop

- O nome Sqoop é uma contração da frase “SQl to hadOOP”.
- Tem como objetivo executar a transferência eficiente e bidirecional de dados entre o Hadoop e diversos serviços de armazenamento externo de dados estruturados.

FIAP

Apache Sqoop

- Apesar de o Hadoop ser amplamente conhecido por sua capacidade de armazenamento e processamento de grandes quantidades de dados, muita informação ainda hoje está armazenada em bancos de dados relacionais.
- Assim, surgiu o Apache Sqoop cujo objetivo é executar a transferência eficiente e bidirecional de dados entre o Hadoop e diversos serviços de armazenamento externo de dados estruturados.

FIAP

Apache Sqoop

- O Apache Sqoop pode ser útil
  - quando se deseja utilizar ferramentas de Big Data para o processamento de bases relacionais,
  - para a integração de bases relacionais e mainframes com dados já presentes no Hadoop
  - para o arquivamento dos dados no Hadoop.

FIAP

Banco de Dados Suportados

- Por padrão, o Sqoop utiliza JDBC (Java Database Connectivity) para se conectar aos bancos e, por esse motivo, acredita-se que ele é compatível com uma grande quantidade de bancos de dados, uma vez que os fornecedores implementam essa API.
- No entanto, o Sqoop não garante a compatibilidade e a performance com todos os bancos que possuem conectores JDBC devido às formas de implementação dessa API e a ligeiras diferenças que possam existir na sintaxe SQL de cada banco.

Banco de Dados Suportados

Banco de Dados	Versão	String de Conexão
HSQLDB	1.8.0+	jdbc:hsqldb:*//
MySQL	5.0+	jdbc:mysql://
Oracle	10.2.0+	jdbc:oracle:*//
PostgreSQL	8.3+	jdbc:postgresql://
CUBRID	9.2+	jdbc:cubrid:*
HSQLDB	1.8.0+	jdbc:hsqldb:*//

- Alguns bancos e respectivas versões que foram testadas com o Sqoop.
- Pode ser necessário instalar os drivers de conexão mesmo com o banco de dados estando presente na lista.

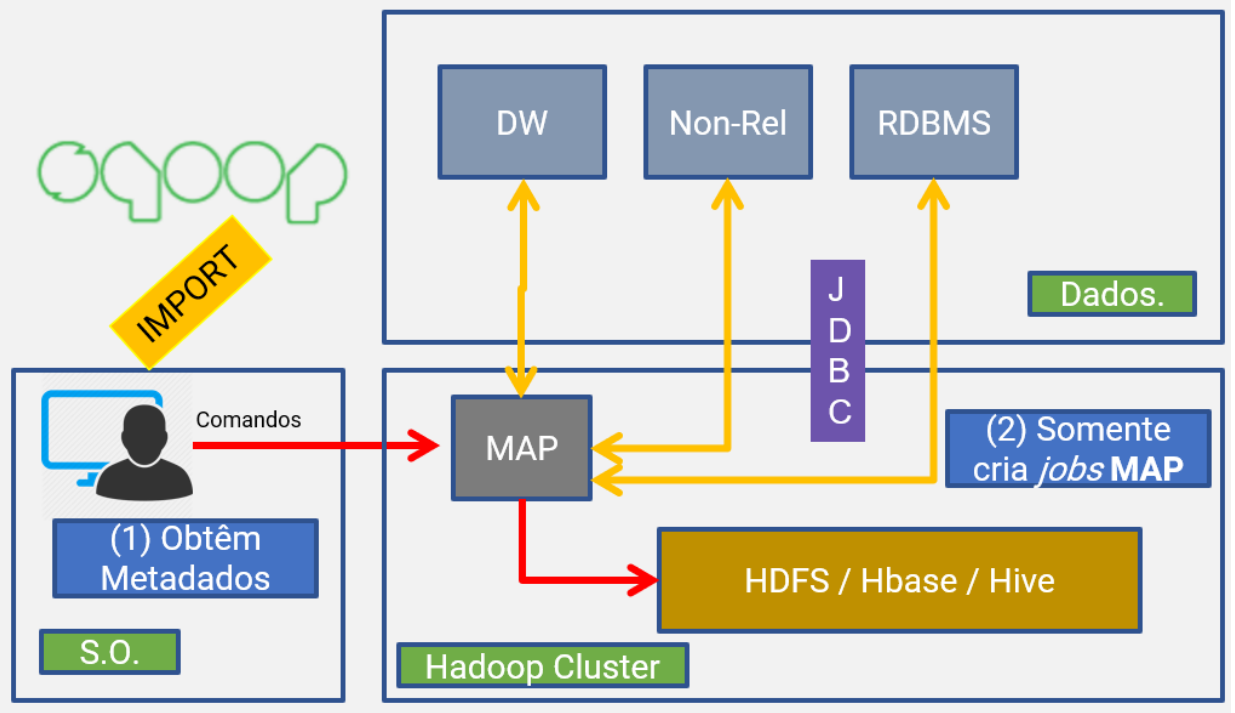
Fonte: http://sqoop.apache.org/docs/1.4.6/SqoopUserGuide.html

FIAP

Importação de Dados

- O Sqoop realiza a importação dos dados em
  - arquivos de texto
    - *–as-textfile*
  - arquivos Sequence File
    - *–as-sequencefile*
  - Avro
    - *–as-avrodatafile*
  - Parquet
    - *–as-parquetfile*

Arquitetura SQOOP



- A ferramenta **import** do SQOOP importa tabelas individuais do banco de dados para o HDFS. Cada linha em uma tabela é tratada como um registro HDFS.

Arquitetura SQOOP

- A importação é dividida em duas etapas principais. Na primeira fase:
- O Sqoop lê os metadados da tabela de origem para que sejam convertidos em um tipo de dados Java durante a criação de uma classe que encapsulará um registro dessa tabela.

Arquitetura SQOOP

```
INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-oracle/compile/afb1b49e95ac0c03b06cf9ce41b91e72/ACTIVITY.jar
INFO manager.OracleManager: Time zone has been set to GMT
INFO manager.OracleManager: Time zone has been set to GMT
INFO mapreduce.ImportJobBase: Beginning import of ACTIVITY
INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
```

- Essa classe também possui métodos para a serialização e desserialização desses dados.
- É um subproduto do processo de importação e pode ser utilizada por qualquer outra tarefa de MapReduce

Arquitetura SQOOP

```
19/10/27 19:15:45 INFO manager.OracleManager: Time zone has been set to GMT
19/10/27 19:15:45 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM ACTIVITY t WHERE 1=0
19/10/27 19:15:46 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-oracle/compile/afb1b49e95ac0c03b06cf9ce41b91e72/ACTIVITY.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

- Após a extração dos metadados e criação da classe, inícia a segunda etapa, o processo de importação dos dados, que nada mais é que uma simples consulta SQL.

Arquitetura SQOOP

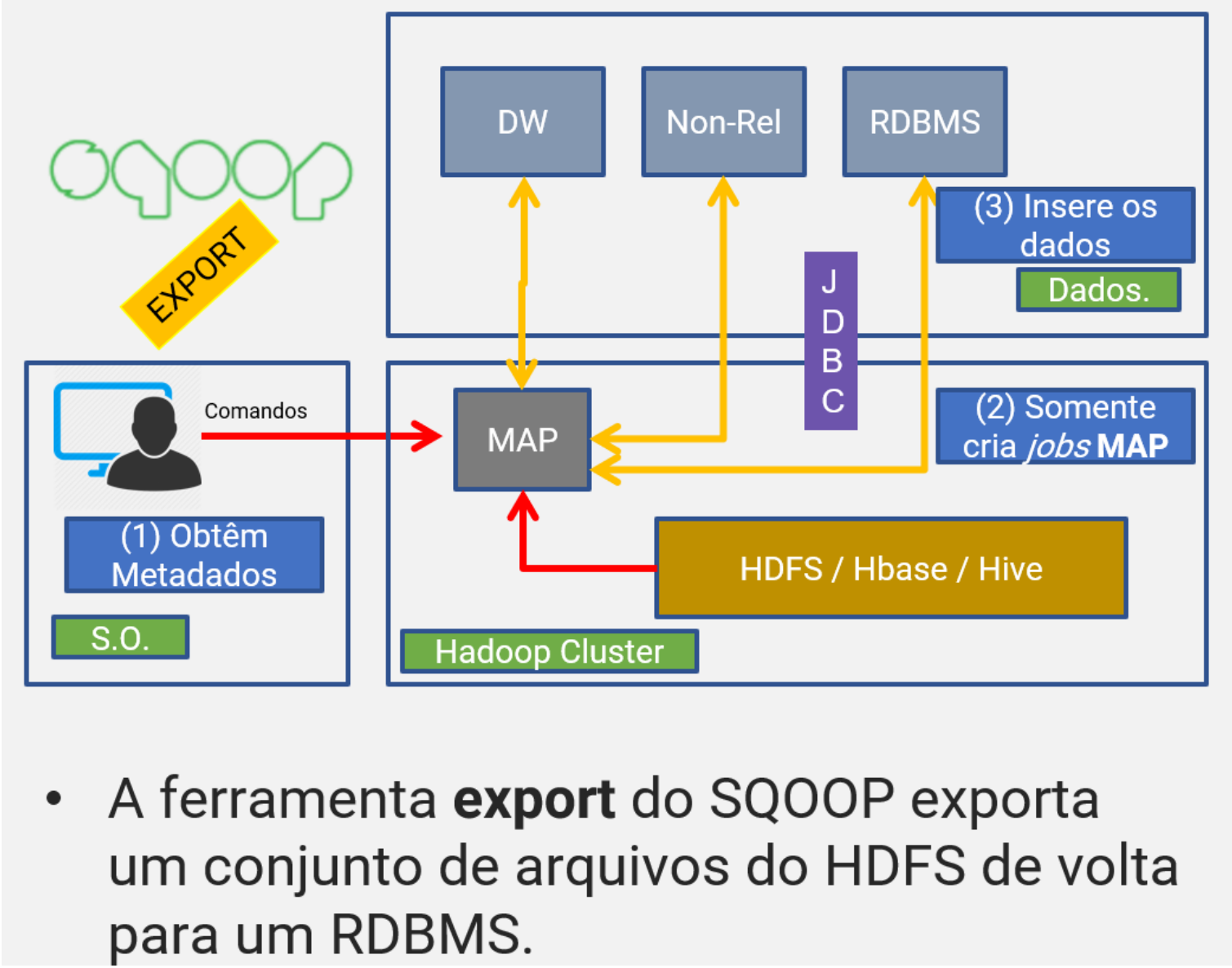
```
19/10/27 19:15:55 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, us
19/10/27 19:15:55 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/10/27 19:16:11 INFO db.DBInputFormat: Using read commited transaction isolation
19/10/27 19:16:11 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(ACTIVITY_ID)
19/10/27 19:16:14 INFO mapreduce.JobSubmitter: number of splits:4
```

- O Sqoop, por padrão, utiliza o nível de isolamento de transação chamado read committed, que assegura a não ocorrência de leituras sujas – ou seja, que uma transação TA leia dados modificados mas ainda não confirmados por uma transação TB.
- Enquanto o Sqoop faz a leitura dos registros, os dados podem ser modificados e confirmados por outras transações.

Arquitetura SQOOP

```
19/10/27 19:16:11 INFO db.DBInputFormat: Using read commited transaction isolation
19/10/27 19:16:11 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(ACTIVITY_ID), MAX(ACTIVITY_ID) FROM ACTIVITY
19/10/27 19:16:14 INFO mapreduce.JobSubmitter: number of splits:4
19/10/27 19:16:15 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1572203140633_0001
19/10/27 19:16:19 INFO impl.YarnClientImpl: Submitted application application_1572203140633_0001
```

- O Sqoop é capaz de paralelizar o processo de importação, distribuindo a consulta SQL por meio dos nós do seu cluster.
- Para isso, ele precisa saber qual coluna da tabela a ser importada deve ser utilizada para dividir os dados de origem e quantos processos deverão ser utilizados em paralelo.



- A ferramenta **export** do SQOOP exporta um conjunto de arquivos do HDFS de volta para um RDBMS.

Importação de Dados

- Na importação para arquivos textuais, cada registro do banco se tornará uma linha no arquivo de destino onde cada coluna é delimitada por padrão pela vírgula ou outro separador definido pelo usuário por meio do argumento *–fields-terminated-by*.

FIAP

Importação de Dados

Argumentos	Descrição
–enclosed-by <char>	Define o caractere que iniciará e terminará cada campo.
–escaped-by <char>	Define o caractere de escape.
–fields-terminated-by <char>	Define o caractere separador dos campos.
–lines-terminated-by <char>	Define o caractere para o fim de linha.
–mysql-delimiters	Utiliza os delimitadores MySQL padrão. Barra invertida mais o caractere n (\n) para linha, barra invertida (\ ) como caractere de escape e aspas simples (‘) como caractere opcional para início e fim de cada campo.
–optionally-enclosed-by <char>	Define o caractere opcional que iniciará e terminará cada campo. O caractere opcional será utilizado apenas quando o caractere delimitador aparecer no dado importado.

Importação de Dados

ID	Nome	Depart
1	nome1	d1
2	nome2	d2
3	nome3	d3

- O Sqoop realiza a importação dos dados em
  - arquivos de texto
    - -as-textfile

```
1;nome1;d1;2;nome2;d2;3;nome3;d3
```

  - arquivos Sequence File
    - -as-sequencefile

```
00110001 01101110 01101111 01101101
01100101 00110001 01100100 00110001
```

  - Avro
    - -as-avrodatafile

```
[{id:1,nome:nome1,depart:d1},
(id:2,nome:nome2:depart:d2},
(id:3,nome:nome3;depart:d3}]
```

  - Parquet
    - -as-parquetfile

```
1 2 3 nome1 nome2 nome3 d1 d2 d3
```

FIAP

Importação de Dados

- O formato de texto delimitado é ideal quando não estão sendo importados dados binários.
- Problema: O texto "As armas e os barões assinalados, que da ocidental praia Lusitana" pode apresentar problemas na importação caso o delimitador padrão (a vírgula) seja usado.
- Podemos resolver esse problema com o operador *--fields-terminated-by*.

Exemplo de Importação

```
SQL> select * from
moviedemo.activity;

ACTIVITY_ID NAME
----------- --------------------
          3 Pause
          6 List
          7 Search
          8 Login
          9 Logout
         10 Incomplete
         11 Purchase
          1 Rate
          2 Completed
          4 Start
          5 Browse
```

- Neste exemplo importaremos os dados de uma tabela Oracle denominada ACTIVITY.

Exemplo de Importação

```
sqoop import
--connect jdbc:oracle:thin:@localhost:1521/orcl
--username MOVIEDEMO --password welcome1
--table ACTIVITY
```

- import – indica uma operação de importação
- connect - recebe como parâmetro o endereço do servidor seguido pelo nome do banco de dados e o número da porta, quando necessário.
- username – indica o nome do *schema*
- password – indica a senha do usuário
- table – indica o nome da tabela

FIAP

Importação de Dados

- Por padrão, o Sqoop importará os dados para o HDFS em um diretório de mesmo nome da tabela de origem.
- Esse diretório de destino pode ser alterado por meio do argumento *--warehouse-dir*, tendo como parâmetro um diretório qualquer.

Importação de Dados

- Dentro do diretório especificado serão criados outros diretórios com o nome da tabela de origem contendo os arquivos com os respectivos dados importados.
- Uma outra opção é o argumento *--target-dir* seguido do caminho completo para o diretório de destino desejado

Exemplo de Importação

FIAP

```
sqoop import
--connect jdbc:oracle:thin:@localhost:1521/orcl
--username MOVIEDEMO --password welcome1
--table ACTIVITY --target-dir etl/input/dados
```

- import – indica uma operação de importação
- connect - recebe como parâmetro o endereço do servidor seguido pelo nome do banco de dados e o número da porta, quando necessário.
- username – indica o nome do *schema*
- password – indica a senha do usuário
- table – indica o nome da tabela
- target-dir – indica a localização dos novos arquivos importados

Importação de Dados

FIAP

- Nos exemplos, a senha é passada por meio do argumento --password.
- Apesar de essa ser uma forma muito simples de autenticação com o banco relacional, ela é também a maneira menos segura.
- Outra opção seria por meio do argumento --password-file, que recebe como parâmetro um arquivo que contém a senha.
- Pode-se ainda restringir as permissões de acesso a esse arquivo concedendo restrições de leitura apenas para o proprietário do arquivo (chmod 400)
- Uma terceira opção seria por meio do argumento *-P*, que solicitará a senha mediante o console a cada execução do Sqoop

FIAP

Importação de Dados

- O Sqoop permite a importação de toda uma tabela ou de algumas colunas específicas, ordenadas a critério do usuário.
- Para a importação de algumas colunas específicas, basta utilizar o argumento *--columns* separando o nome das colunas por vírgula – por exemplo, *--columns* "id, cidade, pais".
- Além disso, o Sqoop permite restringir as linhas a serem importadas através do argumento –*where*.

Exemplo de Importação

FIAP

```
sqoop import
--connect jdbc:oracle:thin:@localhost:1521/orcl
--username MOVIEDEMO --password welcome1
--table ACTIVITY --columns “name” --where
“ACTIVITY_ID > 7”
```

- columns – indica o nome das colunas que devem ser importadas
- where – indica o filtro que será aplicado durante a extração dos dados.

Importação de Dados

- Em tabelas que possuem colunas do tipo binário,
  - Se a coluna possui tamanho de até 16 MB, ou definido pelo argumento *--inline-lob-limit* <número de bytes>, então esse campo é inserido na mesma linha, junto às demais colunas desse registro.
  - Se o campo ultrapassar o valor padrão de 16 MB ou o valor definido pelo argumento *--inline-lob-limit*, o campo será armazenado em um novo arquivo – de até 263 bytes cada – em um subdiretório _lobs no diretório de destino.

FIAP

O processo de Importação

- A importação é dividida em duas etapas principais. Na primeira fase:
  - O Sqoop lê os metadados da tabela de origem para que sejam convertidos em um tipo de dados Java durante a criação de uma classe que encapsulará um registro dessa tabela.
  - Essa classe também possui métodos para a serialização e desserialização desses dados.
  - É um subproduto do processo de importação e pode ser utilizada por qualquer outra tarefa de MapReduce.

O processo de Importação

- (continuação)
  - A classe que possui o nome da tabela de origem é salva no diretório corrente de onde o Sqoop está sendo chamado.
  - Após a compilação dessa classe, o código compilado é salvo na pasta /tmp.
  - Tanto o nome da classe quanto os diretórios de destino da classe gerada e da classe compilada podem ser alterados por meio dos seguinte argumentos: *--class-name*, *--outdir* e *--bindir*, respectivamente.

O processo de Importação

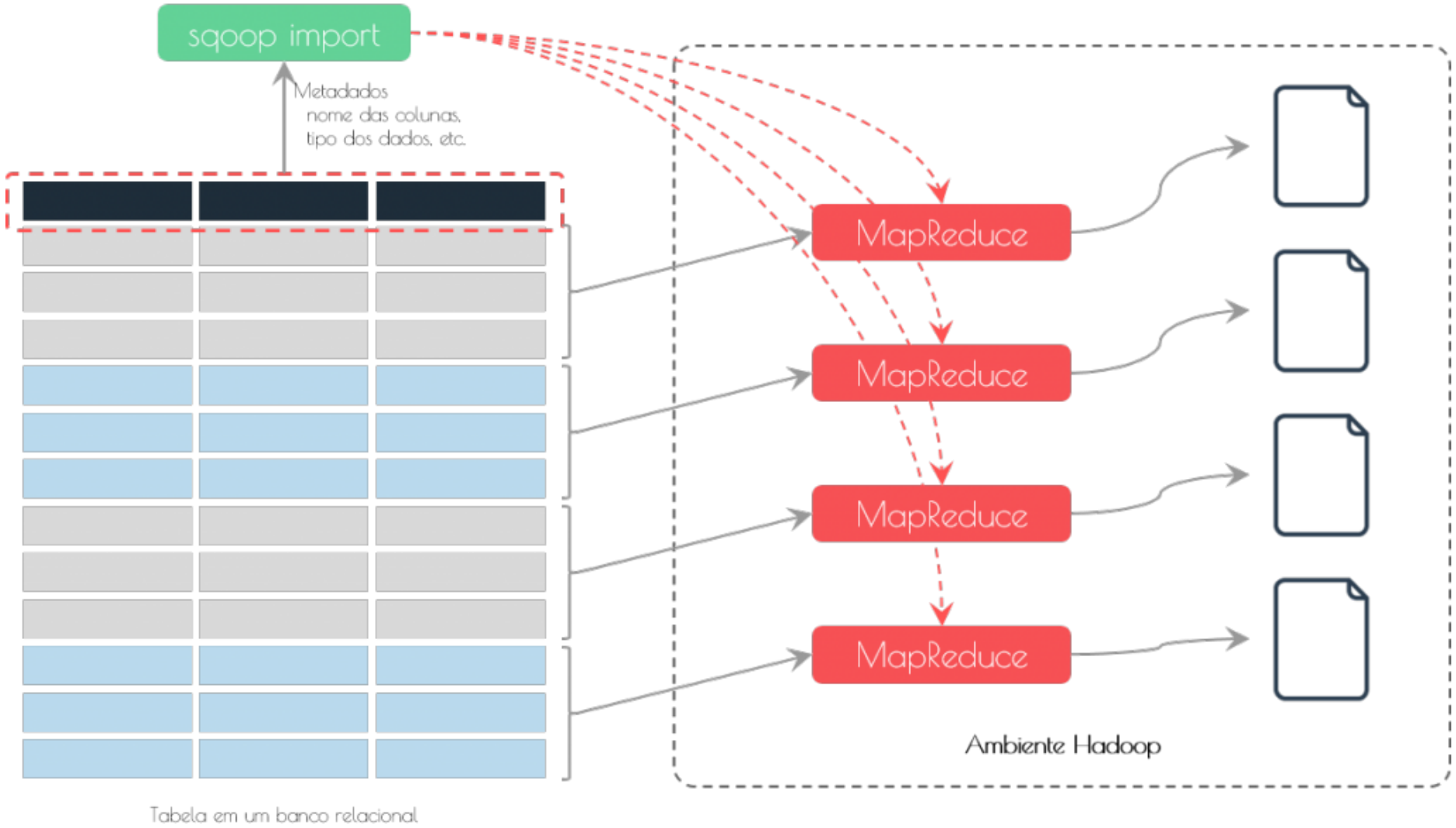
- (continuação)
  - Após a extração dos metadados e criação da classe, inícia a segunda etapa, o processo de importação dos dados, que nada mais é que uma simples consulta SQL.
  - O Sqoop, por padrão, utiliza o nível de isolamento de transação chamado *read committed*, que assegura a não ocorrência de leituras sujas – ou seja, que uma transação TA leia dados modificados mas ainda não confirmados por uma transação TB.
  - Enquanto o Sqoop faz a leitura dos registros, os dados podem ser modificados e confirmados por outras transações.

O processo de Importação

- O Sqoop é capaz de paralelizar o processo de importação, distribuindo a consulta SQL por meio dos nós do seu cluster.
- Para isso, ele precisa saber qual coluna da tabela a ser importada deve ser utilizada para dividir os dados de origem e quantos processos deverão ser utilizados em paralelo.

O processo de Importação

- O mais adequado é a utilização de uma coluna que possua valores uniformemente distribuídos e que possa ser passada para o Sqoop por meio do argumento *--split-by*.
- Quando não informada a coluna, o Sqoop tenta escolher por si só; e por padrão ele escolhe geralmente a chave primária da tabela


sqoop import
Metadados
nome das colunas,
tipo dos dados, etc.
MapReduce
MapReduce
MapReduce
MapReduce
Ambiente Hadoop
Tabela em um banco relacional

O processo de Importação

FIAP

- O número de tarefas de mapeamento utilizado no paralelismo pode ser parametrizado por meio do argumento *--m* ou *--num-mappers*. Por padrão, o Sqoop paraleliza o processo de importação dos dados em quatro tarefas.
- Ele obtém o maior e menor valor da coluna escolhida e faz uma divisão simples.

$$\frac{\text{MAX(Coluna escolhida)} - \text{MIN(Coluna escolhida)}}{\text{Total tarefas de importação}}$$

FIAP

O processo de Importação

- Em uma coluna id em que o menor valor é 0, o maior valor é 1000 e a importação dos dados será paralelizada em quatro tarefas, o Sqoop dividirá em partes iguais ((1000 – 0 ÷ 4) = 250) dando para cada tarefa de mapeamento uma consulta SQL com um intervalo diferente.

```
SELECT * FROM Tabela WHERE (id>= 0) AND (id < 250)
SELECT * FROM Tabela WHERE (id>= 250) AND (id < 500)
SELECT * FROM Tabela WHERE (id>= 500) AND (id < 750)
SELECT * FROM Tabela WHERE (id>= 750) AND (id <= 1000)
```

O processo de Importação

- Se a coluna utilizada na divisão não for uniformemente distribuída, isso pode prejudicar a performance do processo de importação, visto que algumas tarefas podem ficar sobrecarregadas e outras com pouco ou nenhum trabalho.

Exemplo de Importação

```
sqoop import --connect
jdbc:oracle:thin:@localhost:1521/orcl
--username MOVIEDEMO
--password welcome1
--query "SELECT activity_id, SYSDATE + activity_id,
name FROM activity WHERE \$CONDITIONS"
--target-dir etl/input/dados
--split-by "ACTIVITY_ID"
```

- Opcionalmente, o usuário pode substituir os três argumentos --table, --columns e --where pelo argumento --query, onde se deve definir toda a consulta SQL.

Exemplo de Importação

FIAP

```
sqoop import \
  --connect jdbc:mysql://mysql.example.com/sqoop \
  --username sqoop \
  --password sqoop \
  --query 'SELECT normcities.id, \
                  countries.country, \
                  normcities.city \
                  FROM normcities \
                  JOIN countries USING(country_id) \
                  WHERE $CONDITIONS' \
  --split-by id \
  --target-dir cities \
  --mapreduce-job-name normcities
```

- O uso do argumento --query permite extrair dados de diversas tabelas.

Fonte: https://www.oreilly.com/library/view/apache-sqoop-cookbook/9781449364618/ch04.html

FIAP

Importação Incremental

- O argumento *--incremental* importa apenas os dados novos da tabela.
- Possui dois modos (append e lastmodified).
  - append é adequado quando os seus dados de origem não sofrem atualização (UPDATE) e apenas novos dados são inseridos. Em conjunto com o argumento e parâmetro *–incremental* append, deve-se informar os argumentos *–check-column* e *–last-value*, em que o primeiro define qual coluna será utilizada no filtro e o último informa para essa coluna específica qual é o último valor já importado.

Exemplo de Importação append

```
sqoop import
–connect jdbc:mysql://mysql.example.com/BDTeste
–username eduardo
–password 123456
–table cidades
–incremental append
–check-column id
–last-value 3
```

Importação Incremental

- lastmodified é apropriado para os casos em que os dados de origem sofrem atualização.
- A tabela deve
  - possuir uma coluna do tipo data (DATE, TIME, DATETIME ou TIMESTAMP), que é atualizada sempre que um novo registro é adicionado ou um dado existente é modificado;
  - possuir uma coluna composta por valores únicos, como a chave primária.

FIAP

Exemplo de Importação lastmodified

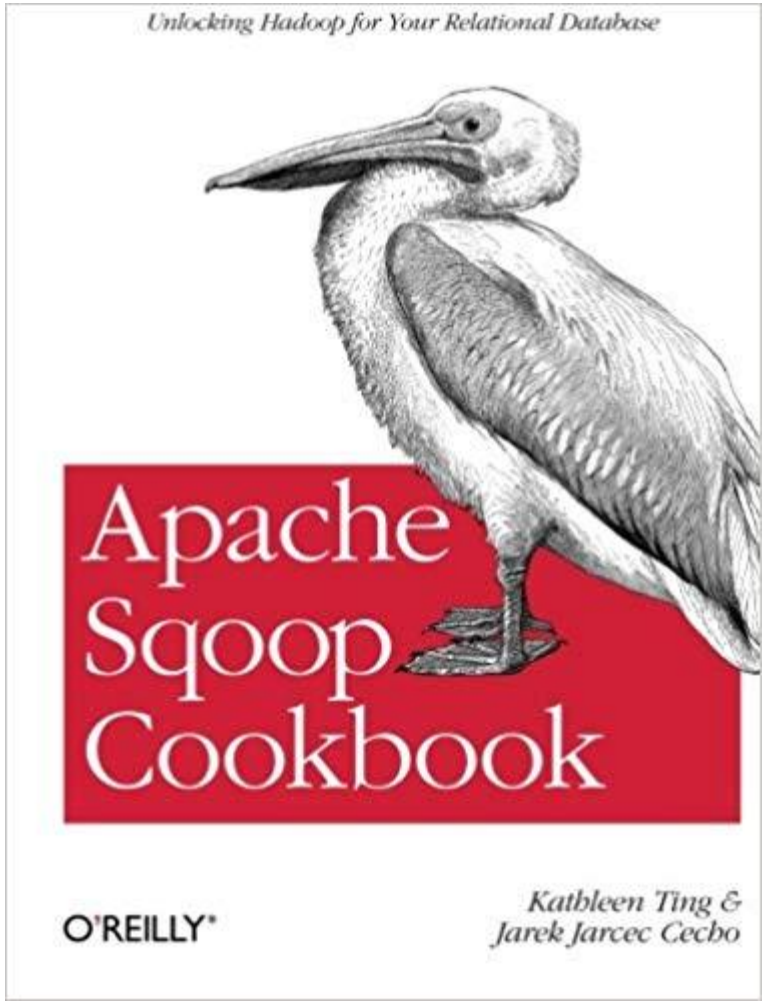
```
sqoop import
–connect jdbc:mysql://mysql.example.com/BDTeste
–username eduardo
–password 123456
–table visits
–incremental lastmodified
–check-column last_update_date
–last-value “1987-02-02”
–merge-key id
```

Importando dados para o Hive

```
sqoop import
--connect jdbc:oracle:thin:@localhost:1521/orcl
--username MOVIEDEMO
--password welcome1
--table CREW --hive-import
```

- É possível importar dados de um banco de dados relacional diretamente para o HIVE. O parâmetro --hive-import instrui o Sqoop a fazer isso.

Bibliografia



- Todos os exemplos de comandos Sqoop presentes nesta apresentação são adaptações dos exemplos do livro *Apache Sqoop Cookbook Unlocking Hadoop for Your Relational Database*, de 2013, escrito por Kathleen Ting e Jarek Jarcec Cecho.