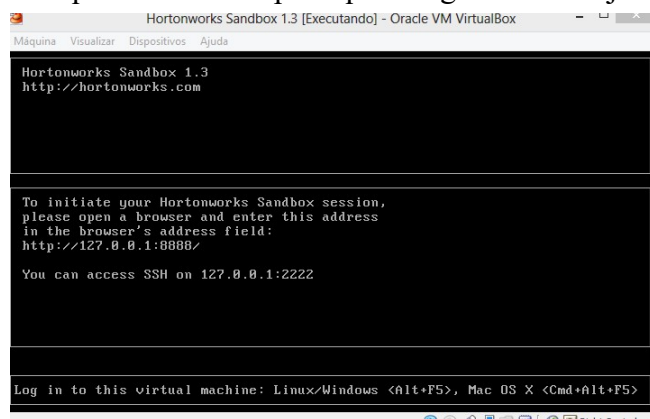


- 1) Para utilização das práticas com o Hadoop vamos usar uma sandbox (máquina virtual) fornecida pela HortonWorks. Esta máquina virtual possui os seguintes produtos instalados:

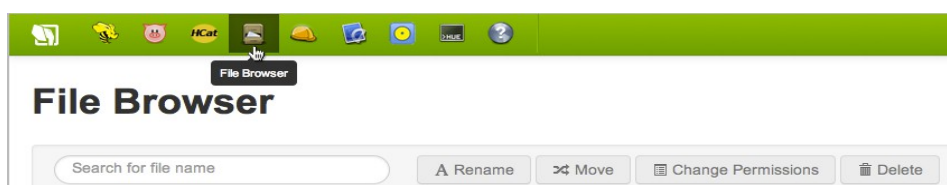
|                                  |          |
|----------------------------------|----------|
| Apache Hadoop                    | 1.2.0    |
| Apache Hive<br>includes HCatalog | 0.11.0   |
| Apache HBase                     | 0.94.6.1 |
| Apache ZooKeeper                 | 3.4.5    |
| Apache Pig                       | 0.11     |
| Apache Sqoop                     | 1.4.3    |
| Apache Oozie                     | 3.3.2    |
| Apache Ambari                    | 1.2.4    |
| Apache Flume                     | 1.3.1    |
| Apache Mahout                    | 0.7.0    |

Esta é a versão 2.3.2 do produto e pode ser obtida em:  
[https://hortonassets.s3.amazonaws.com/2.1/virtualbox/Hortonworks\\_Sandbox\\_2.1.ova](https://hortonassets.s3.amazonaws.com/2.1/virtualbox/Hortonworks_Sandbox_2.1.ova)

- 2) Inicialize a máquina virtual. Espere que a seguinte tela seja exibida:

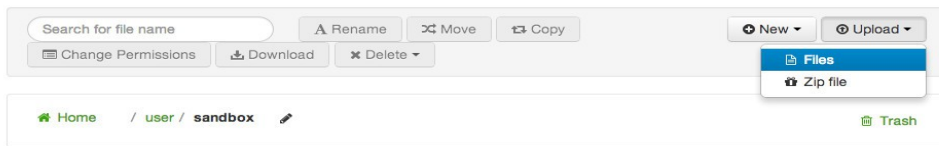


- 3) Abra um navegador e execute a url <http://localhost:8000> ;
- 4) Para este tutorial será utilizado uma base de dados com estatísticas de jogos de baseball entre os anos 1871 até 2011. Este arquivo possui 95.000 linhas. A base de dados pode ser acessada em: <http://seanlahman.com/files/database/lahman591-csv.zip> .
- 5) Apenas dois arquivos serão utilizados: master.csv e batting.csv
- 6) Fazendo Upload dos arquivos:  
Selecione a opção 'File Browser' no topo da página

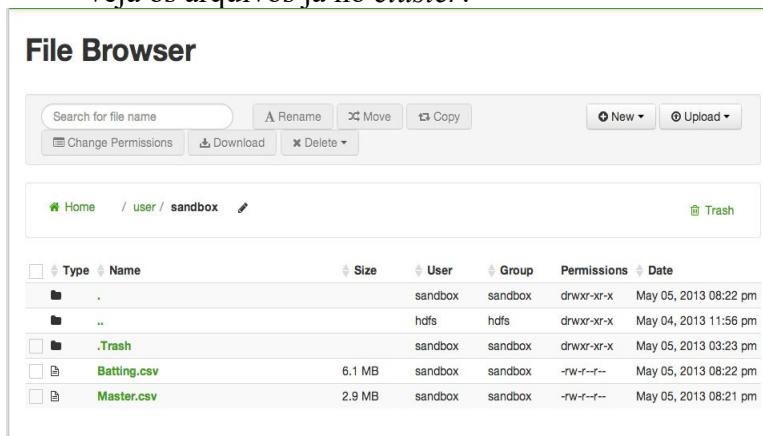


- Navegue na estrutura para o root e entre na pasta */data*;
- Clique no botão *upload* e selecione a opção *files*

## File Browser



- Selecione os dois arquivos: master.csv e batting.csv;
- Veja os arquivos já no *cluster*:



## 7) Criando as tabelas utilizando HCatalog

- Com arquivos já carregados no HDFS, agora serão criadas as tabelas utilizando o HCatalog/Hive (metastore). Selecione a opção HCat no menu:



- No menu 'Actions' selecione a opção 'Create a new table from a file';

### Create a new table from a file

**ACTIONS**  
 Create a new table from a file  
 Create a new table manually

**Table options**  
 Table Name:  Description:

**File options**  
 Input File:

- Criar uma tabela para o arquivo batting.csv:
  - nome da tabela: *batting\_data*;
  - O item 'optional' deixe em branco;
  - Selecione o arquivo;
  - Veja que algumas opções já estarão marcadas, por padrão. E o arquivo será pré exibido;

## Create a new table from a file

**ACTIONS**

Create a new table from a file

Create a new table manually

**Table options**

Table Name:  Description:

**File options**

Input File:  Choose a file

Encoding:  Read column headers: ☒ Import data: ☒

Delimiter:  Autodetect delimiter: ☒ Ignore whitespaces: ☐

Replace delimiter with:  Java-style comments: ☐ Ignore tabs: ☐

Single line comment:

**Table preview**

|        | Column name | Column name | Column name | Column name |
|--------|-------------|-------------|-------------|-------------|
|        | playerid    | yearid      | stint       | teamid      |
|        | Column type | Column type | Column type | Column type |
|        | string      | int         | int         | string      |
| Row #1 | aardsda01   | 2004        | 1           | SFN         |
| Row #2 | aardsda01   | 2006        | 1           | CHN         |
| Row #3 | aardsda01   | 2007        | 1           | CHA         |
| Row #4 | aardsda01   | 2008        | 1           | BOS         |
| Row #5 | aardsda01   | 2009        | 1           | SEA         |
| Row #6 | aardsda01   | 2010        | 1           | SEA         |
| Row #7 | aaronha01   | 1954        | 1           | ML1         |
| Row #8 | aaronha01   | 1955        | 1           | ML1         |
| Row #9 | aaronha01   | 1956        | 1           | ML1         |

Create table

- Altere o nome da coluna 'r' para 'runs' e altere se o tipo de dado para 'int';
- Clique no botão 'create table';

- Criar uma tabela para o arquivo master.csv:
  - nome da tabela: master\_data;
  - O item 'optional' deixe em branco;
  - Selecione o arquivo;

- Veja as duas tabelas criadas no HCatalog:

## HCatalog: Table List

**ACTIONS**

Create a new table from file

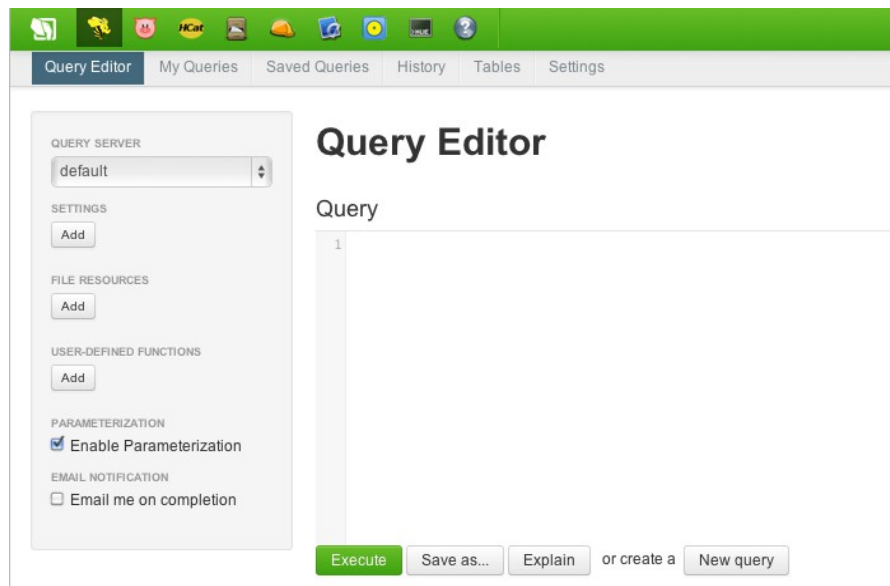
Create a new table manually

| Table Name   |  |
|--------------|--|
| batting_data | <span style="background-color: #ccc; padding: 2px 5px;">Browse Data</span> |
| master_data  | <span style="background-color: #ccc; padding: 2px 5px;">Browse Data</span> |
| sample_07    | <span style="background-color: #ccc; padding: 2px 5px;">Browse Data</span> |
| sample_08    | <span style="background-color: #ccc; padding: 2px 5px;">Browse Data</span> |

- Clique no botão 'Browse Data' para visualizar as tabelas criadas;

## 8) Fazendo algumas análise com o Hive

- Arquivos já carregados no HDFS;
- Metadados já criados para o Hive. Desta forma pode-se trabalhar emitindo-se comandos em HiveQL;
- Para emitir este comando será utilizado uma ferramentas chamada **Beeswax**: interface interativa para o Hive em que poderá ser digitado consultas e as mesmas serão avaliadas pelo Hive e transformadas em uma série de jobs *Map Reduce*;
- Para executar o Beeswax acesse o menu conforme figura abaixo:



- As consultas podem ser digitadas e para execução acione o botão 'Execute'. Não é possível digitar mais de uma consulta, separando-as por “;”, por exemplo.
- Como HCatalog e Hive são integrados, na verdade o mesmo produto, tudo que foi criado no HCatalog é válido para o Hive;
- Digite o comando para visualizar as tabelas:

```
show tables
```

| Query Results: Unsaved Query   |   |  |          |   |              |   |             |   |           |   |           |
|--|---|--|----------|---|--------------|---|-------------|---|-----------|---|-----------|
| <p>DOWNLOADS</p> <p>Download as CSV</p> <p>Download as XLS</p> <p>Save</p> <p>MR JOBS</p> <p>No Hadoop jobs were launched in running this query.</p> | <p>Results Query Log</p> <table> <thead> <tr> <th></th><th>tab_name</th></tr> </thead> <tbody> <tr> <td>0</td><td>batting_data</td></tr> <tr> <td>1</td><td>master_data</td></tr> <tr> <td>2</td><td>sample_07</td></tr> <tr> <td>3</td><td>sample_08</td></tr> </tbody> </table> |  | tab_name | 0 | batting_data | 1 | master_data | 2 | sample_07 | 3 | sample_08 |
|  | tab_name  |  |          |   |              |   |             |   |           |   |           |
| 0  | batting_data  |  |          |   |              |   |             |   |           |   |           |
| 1  | master_data   |  |          |   |              |   |             |   |           |   |           |
| 2  | sample_07   |  |          |   |              |   |             |   |           |   |           |
| 3  | sample_08   |  |          |   |              |   |             |   |           |   |           |

- Hive utiliza o esquema/banco de dados criado no HCatalog. Para estes caso,não é necessário utilizar o nome do esquema. Digite o comando;

```
Select * from batting_data
```

Query Editor

My Queries

Saved Queries

History

Tables

Settings

# Query Results: Unsaved Query

DOWNLOADS

Download as CSV

Download as XLS

Save

MR JOBS

No Hadoop jobs were launched in running this query.

Results

Query

Log

|   | playerid  | yearid | stint | teamid | lgid | g   | g_batting | ab  | runs | h   | 2b | 3b | hr | r   |
|---|-----------|--------|-------|--------|------|-----|-----------|-----|------|-----|----|----|----|-----|
| 0 | aardsda01 | 2004   | 1     | SFN    | NL   | 11  | 11        | 0   | 0    | 0   | 0  | 0  | 0  | 0   |
| 1 | aardsda01 | 2006   | 1     | CHN    | NL   | 45  | 43        | 2   | 0    | 0   | 0  | 0  | 0  | 0   |
| 2 | aardsda01 | 2007   | 1     | CHA    | AL   | 25  | 2         | 0   | 0    | 0   | 0  | 0  | 0  | 0   |
| 3 | aardsda01 | 2008   | 1     | BOS    | AL   | 47  | 5         | 1   | 0    | 0   | 0  | 0  | 0  | 0   |
| 4 | aardsda01 | 2009   | 1     | SEA    | AL   | 73  | 3         | 0   | 0    | 0   | 0  | 0  | 0  | 0   |
| 5 | aardsda01 | 2010   | 1     | SEA    | AL   | 53  | 4         | 0   | 0    | 0   | 0  | 0  | 0  | 0   |
| 6 | aaronha01 | 1954   | 1     | ML1    | NL   | 122 | 122       | 468 | 58   | 131 | 27 | 6  | 13 | 69  |
| 7 | aaronha01 | 1955   | 1     | ML1    | NL   | 153 | 153       | 602 | 105  | 189 | 37 | 9  | 27 | 106 |
| 8 | aaronha01 | 1956   | 1     | ML1    | NL   | 153 | 153       | 609 | 106  | 200 | 34 | 14 | 26 | 92  |
| 9 | aaronha01 | 1957   | 1     | ML1    | NL   | 151 | 151       | 615 | 118  | 198 | 27 | 6  | 44 | 132 |

Did you know? You can click on a row to select a column you want to jump to.

- É também possível visualizar as colunas de uma tabela com o comando;

```
describe from batting_data
```

**Query Editor** My Queries Saved Queries History Tables Settings

### Query Results: Unsaved Query

**DOWNLOADS**  
Download as CSV  
Download as XLS  
Save  
  
**MR JOBS**  
No Hadoop jobs were launched in running this query.

Results Query Log

|   | col_name  | data_type | comment |
|---|-----------|-----------|---------|
| 0 | playerid  | string    |         |
| 1 | yearid    | string    |         |
| 2 | stint     | string    |         |
| 3 | teamid    | string    |         |
| 4 | lgid      | string    |         |
| 5 | g         | string    |         |
| 6 | g_batting | string    |         |
| 7 | ab        | string    |         |
| 8 | runs      | int       |         |
| 9 | h         | string    |         |

Did you know? You can click on a row to select a column you want to jump to.

- Utilizando o Hive também é possível fazer junções dos dados. Digite o comando abaixo para testar uma junção;

```
select m.playerid, m.namefirst,m.namelast,  
       b.yearid,b.runs  
from master_data m  
join batting_data b on (m.playerid = b.playerid )
```

### Query Results: Unsaved Query

**DOWNLOADS**  
Download as CSV  
Download as XLS  
Save  
  
**MR JOB (1)**  
201302131533\_0001

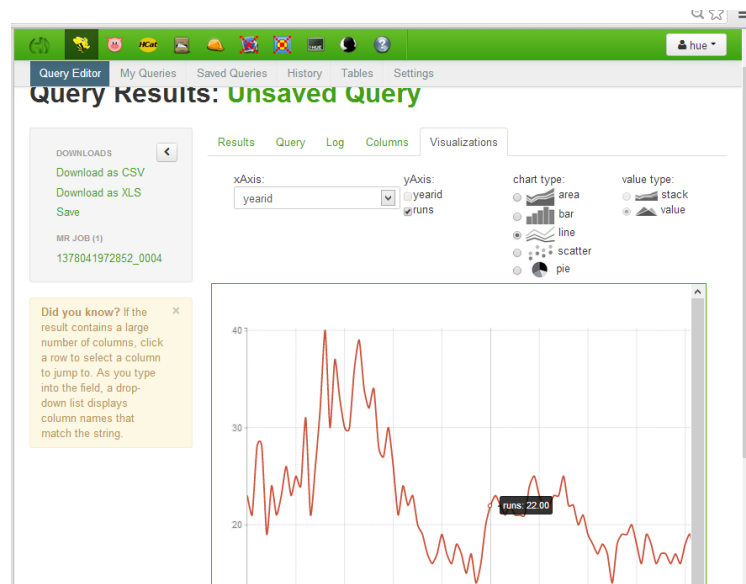
Results Query Log

|    | playerid  | namefirst | namelast | yearid | runs |
|----|-----------|-----------|----------|--------|------|
| 0  | aaronha01 | Hank      | Aaron    | 1976   | 22   |
| 1  | aaronha01 | Hank      | Aaron    | 1954   | 58   |
| 2  | aaronha01 | Hank      | Aaron    | 1955   | 105  |
| 3  | aaronha01 | Hank      | Aaron    | 1956   | 106  |
| 4  | aaronha01 | Hank      | Aaron    | 1957   | 118  |
| 5  | aaronha01 | Hank      | Aaron    | 1958   | 109  |
| 6  | aaronha01 | Hank      | Aaron    | 1959   | 116  |
| 7  | aaronha01 | Hank      | Aaron    | 1960   | 102  |
| 8  | aaronha01 | Hank      | Aaron    | 1961   | 115  |
| 9  | aaronha01 | Hank      | Aaron    | 1962   | 127  |
| 10 | aaronha01 | Hank      | Aaron    | 1963   | 121  |
| 11 | aaronha01 | Hank      | Aaron    | 1964   | 103  |
| 12 | aaronha01 | Hank      | Aaron    | 1965   | 109  |
| 13 | aaronha01 | Hank      | Aaron    | 1966   | 117  |

- Observe a geração dos códigos para o jobs Map Reduce. Veja o log. Observe também que é gerado um número para o processo job Map Reduce: 'MR JOB(1)';
- Outra funcionalidade é a visualização, Clique na aba 'Visualizations'. Não altere a visualização. Vamos criar um resultado mais adequado para ser apresentado:

```
select yearid,avg(runs) runs  
from batting_data  
group by yearid  
order by yearid
```

- Clique na aba 'Visualizations' e altere para um gráfico de linhas:



- Outras funcionalidades do Beeswax são armazenar ou visualizar as consultas já emitidas:

The screenshot shows the 'My Queries' interface. It has a search bar, 'View result', 'Edit', 'Clone', and 'Create New Query' buttons. Below, there are tabs for 'Recent Saved Queries' (0) and 'Recent Run Queries' (4). The table below lists the recent queries.

| Time              | Name    | Query   | State     |
|-------------------|---------|---|-----------|
| 09/03/13 06:00:01 | Unsaved | <code>select yearid, avg(runs) runs from batting_data group by yearid order by yearid</code>                  | available |
| 09/03/13 05:48:39 | Unsaved | <code>select m.playerid, m.namefirst, m.namelast, b.yearid, b.runs from master_data m join batting_...</code> | available |
| 09/03/13 05:42:52 | Unsaved | <code>describe batting_data</code>  | available |
| 09/03/13 05:37:13 | Unsaved | <code>show tables</code>  | available |

- Pode-se ainda consultar um histórico de consultas já emitidas (lembre-se que toda consulta foi convertida para um job *Map Reduce*);

The screenshot shows the 'History' interface. It has a table with columns: Time, Name, Query, User, State, and Result. The table lists the historical queries.

| Time              | Name      | Query   | User | State     | Result  |
|-------------------|-----------|---|------|-----------|---------|
| 09/03/13 06:00:01 | [Unsaved] | <code>select yearid, avg(runs) runs from batting_data group by yearid order by yearid</code>                  | hue  | available | Results |
| 09/03/13 05:48:39 | [Unsaved] | <code>select m.playerid, m.namefirst, m.namelast, b.yearid, b.runs from master_data m join batting_...</code> | hue  | available | Results |
| 09/03/13 05:42:52 | [Unsaved] | <code>describe batting_data</code>  | hue  | available | Results |
| 09/03/13 05:37:13 | [Unsaved] | <code>show tables</code>  | hue  | available | Results |

Showing 1 to 4 of 4 items, page 1 of 1