

INTRODUÇÃO AO HIVE

//Banco de Dados

//Criar um banco de dados no Hive

```
hive> CREATE DATABASE financials;  
hive> CREATE DATABASE IF NOT EXISTS financials;
```

//Mostra todos os bancos de dados

```
hive> SHOW DATABASES;
```

//Mostra Todos os bancos que começam com h

```
hive> SHOW DATABASES LIKE 'h.*';
```

// Cria um banco de dados em uma localizacao especifica

```
hive> CREATE DATABASE financials LOCATION '/my/preferred/directory';
```

// Mostra informacoes da base de dados

```
hive> DESCRIBE DATABASE financials;  
hive> DESCRIBE DATABASE default;
```

// Escolhe o banco de dados

```
hive> USE financials;
```

// Apaga o banco de dados

```
hive> DROP DATABASE IF EXISTS financials;
```

//Tabelas

//Criar uma tabela

//Verificar o conteúdo do arquivo NASDAQ_daily_prices_A.csv. Fonte: historical NASDAQ data from InfoChimps

```
head NASDAQ_daily_prices_A.csv
```

```
exchange,stock_symbol,date,stock_price_open,stock_price_high,stock_
price_low,stock_price_close,stock_volume,stock_price_adj_close
NASDAQ,ABXA,2009-12-09,2.55,2.77,2.50,2.67,158500,2.67
NASDAQ,ABXA,2009-12-08,2.71,2.74,2.52,2.55,131700,2.55
NASDAQ,ABXA,2009-12-07,2.65,2.76,2.65,2.71,174200,2.71
NASDAQ,ABXA,2009-12-04,2.63,2.66,2.53,2.65,230900,2.65
NASDAQ,ABXA,2009-12-03,2.55,2.62,2.51,2.60,360900,2.60
NASDAQ,ABXA,2009-12-02,2.41,2.59,2.40,2.53,287700,2.53
NASDAQ,ABXA,2009-12-01,2.35,2.44,2.27,2.40,302000,2.40
NASDAQ,ABXA,2009-11-30,2.36,2.36,2.11,2.25,446100,2.25
NASDAQ,ABXA,2009-11-27,2.35,2.42,2.30,2.35,135200,2.35
```

//Criar um script denominado stocks.sql

```
CREATE EXTERNAL TABLE IF NOT EXISTS stocks (
`exchange` STRING,
`symbol` STRING,
`ymd` STRING,
`price_open` FLOAT,
`price_high` FLOAT,
`price_low` FLOAT,
`price_close` FLOAT,
`volume` INT,
`price_adj_close` FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

DESCRIBE stocks;

LOAD DATA LOCAL INPATH '/home/bitnami/NASDAQ_daily_prices_A.csv' IN
TO TABLE stocks;

SELECT * FROM stocks;
```

//Executar o comando abaixo

```
sudo hive -f stocks.sql
```

NASDAQ	ARRS	2001-07-19	9.02	9.44	8.9	9.31	3321600	9.31
NASDAQ	ARRS	2001-07-18	8.96	9.1	8.73	8.77	852900	8.77
NASDAQ	ARRS	2001-07-17	9.8	9.88	8.78	8.97	682700	8.97
NASDAQ	ARRS	2001-07-16	10.34	10.38	9.86	9.86	187500	9.86
NASDAQ	ARRS	2001-07-13	10.0	10.5	10.0	10.37	346000	10.37
NASDAQ	ARRS	2001-07-12	10.3	10.54	9.9	10.25	532700	10.25
NASDAQ	ARRS	2001-07-11	9.99	10.31	9.8	10.01	354800	10.01
NASDAQ	ARRS	2001-07-10	10.74	10.9	9.92	9.99	818000	9.99
NASDAQ	ARRS	2001-07-09	11.8	11.86	10.3	10.7	2356800	10.7
NASDAQ	ARRS	2001-07-06	12.5	12.62	11.82	11.82	229200	11.82
NASDAQ	ARRS	2001-07-05	13.04	13.13	12.42	12.7	240100	12.7
NASDAQ	ARRS	2001-07-03	13.01	13.59	13.01	13.17	299500	13.17
NASDAQ	ARRS	2001-07-02	12.24	13.15	12.23	13.01	517500	13.01
NASDAQ	ARRS	2001-06-29	12.1	13.0	11.2	12.4	641300	12.4
NASDAQ	ARRS	2001-06-28	12.0	12.45	12.0	12.23	305000	12.23
NASDAQ	ARRS	2001-06-27	11.58	12.53	11.52	11.93	347800	11.93

//Criar a tabela dividends

//Criar um script denominado dividends.sql

```
CREATE EXTERNAL TABLE IF NOT EXISTS dividends (  
  `exchange` STRING,  
  `symbol` STRING,  
  `ymd` STRING,  
  `dividends` FLOAT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';  
  
DESCRIBE dividends;  
  
LOAD DATA LOCAL INPATH '/home/bitnami/NASDAQ_dividends_A.csv' INTO  
TABLE dividends;
```

```
SELECT * FROM dividends;
```

//Executar o comando abaixo

```
sudo hive -f dividends.sql
```

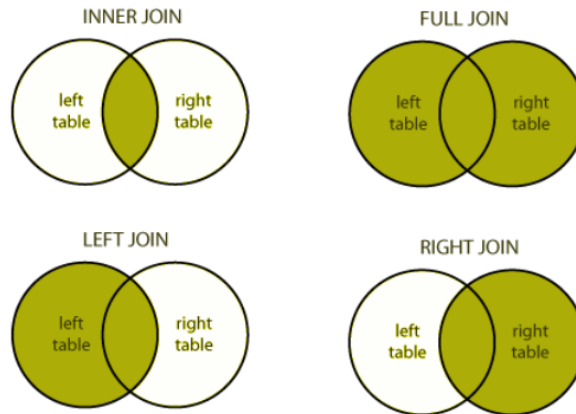
NASDAQ	AROW	2003-08-27	0.17943
NASDAQ	AROW	2003-05-28	0.17943
NASDAQ	AROW	2003-02-26	0.17251
NASDAQ	AROW	2002-11-26	0.17251
NASDAQ	AROW	2002-08-28	0.1643
NASDAQ	AROW	2002-05-29	0.1643
NASDAQ	AROW	2002-02-27	0.15117
NASDAQ	AROW	2001-11-28	0.15117
NASDAQ	AROW	2001-08-29	0.14397
NASDAQ	AROW	2001-05-30	0.14397
NASDAQ	AROW	2001-02-27	0.13145
NASDAQ	AROW	2000-11-29	0.13145
NASDAQ	AROW	2000-08-30	0.12519
NASDAQ	AROW	2000-05-30	0.12519
NASDAQ	AROW	2000-02-28	0.11892

// Consultas

```
hive> SELECT * FROM stocks WHERE symbol = 'AROW';
```

```
hive> SELECT * FROM dividends WHERE symbol = 'AROW';
```

// Joins



// Inner Join

```
hive> SELECT s.symbol, s.ymd, s.price_high , d.dividends  
FROM stocks s INNER JOIN dividends d ON (s.symbol = d.symbol AND s.  
ymd = d.ymd) WHERE s.symbol = 'AROW';
```

// Full Join

```
hive> SELECT s.symbol, s.ymd, s.price_high , d.dividends  
FROM stocks s FULL OUTER JOIN dividends d ON (s.symbol = d.symbol A  
ND s.ymd = d.ymd) WHERE s.symbol = 'ARRS';
```

// Tabelas com particionamento

// Criando a tabela bovespa

Banco de dados da série histórica das ações da Bovespa de 2001 a 2018 disponível em <https://www.kaggle.com/dcampeao/bovespa>.

```
CREATE TABLE bovespa (  
  `typereg` TINYINT,  
  `data` VARCHAR(8),  
  `bdicode` SMALLINT,  
  `codneg` VARCHAR(10),  
  `markettype` SMALLINT,  
  `company` VARCHAR(50),  
  `spec` VARCHAR(20),  
  `prazot` VARCHAR(3),  
  `currency` VARCHAR(3),  
  `open` FLOAT,  
  `max` FLOAT,  
  `min` FLOAT,  
  `med` INT,  
  `close` FLOAT,  
  `preofc` FLOAT,  
  `preofv` FLOAT,  
  `totneg` INT,  
  `quatot` BIGINT  
)  
  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ','  
STORED AS TEXTFILE  
LOCATION '/user/dados';
```

// Consultando os dados na tabela particionada torna a consulta mais rápida.

```
SELECT * FROM bovespa WHERE data = '20021007';
```

// Cria uma nova tabela particionada pelo campo data

```
CREATE TABLE bovespa_part (  
  `typereg` TINYINT,
```

```

`bdicode` SMALLINT,
`codneg` VARCHAR(10),
`marketype` SMALLINT,
`company` VARCHAR(50),
`spec` VARCHAR(20),
`prazot` VARCHAR(3),
`currency` VARCHAR(3),
`open` FLOAT,
`max` FLOAT,
`min` FLOAT,
`med` INT,
`close` FLOAT,
`preofc` FLOAT,
`preofv` FLOAT,
`totneg` INT,
`quatot` BIGINT
)
PARTITIONED BY (`data` VARCHAR(8))
STORED AS TEXTFILE;

```

// Migra os dados da tabela bovespa para a tabela particionada

```

SET hive.exec.dynamic.partition = true;
SET hive.exec.dynamic.partition.mode = nonstrict;
SET hive.exec.mode.local.auto=false
SET hive.auto.convert.join=false;

```

```

INSERT OVERWRITE TABLE bovespa_part PARTITION(data)
SELECT typereg,
bdicode,
codneg,
marketype,
company,
spec,
prazot,
currency,
open,
max,

```

```
min,  
med,  
close,  
preofc,  
preofv,  
totneg,  
quatot,  
data  
FROM bovespa;
```

// Consultando os dados na tabela particionada torna a consulta mais rápida.

```
SELECT * FROM bovespa_part WHERE data = '20021007';
```