

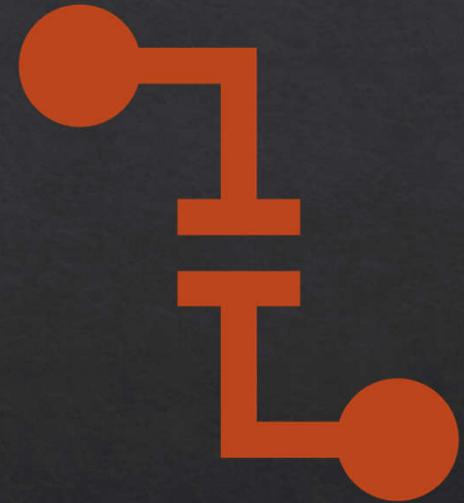


# Big Data com Hive e Impala

## SQOOP

# Criar Banco no Hive

```
beeline  
!connect jdbc:hive2://  
create database retail_db;
```



# Listar Banco de Dados e Tabelas

```
sqoop list-databases --connect  
jdbc:mysql://localhost/ --username root  
--password cloudera
```

```
sqoop list-tables --connect  
jdbc:mysql://localhost/retail_db --  
username root --password cloudera
```

# Opção 1 - Importar cada Tabela

```
sqoop import --connect  
jdbc:mysql://localhost/retail_db --  
table customers --username root --  
password cloudera --hive-import --  
create-hive-table --hive-table  
retail_db.customers;
```

- ❖ categories
- ❖ customers
- ❖ departments
- ❖ order\_items
- ❖ orders
- ❖ products

## Opção 2 - Importar Todas as Tabelas

```
sqoop import-all-tables --connect  
jdbc:mysql://localhost/retail_db --  
username root --password cloudera --  
hive-import --hive-overwrite --hive-database  
retail_db --create-hive-table --m 1;
```

# Checando Resultados no Hive

```
mysql -u root -pcloudera
use retail_db;
show tables;

+-----+-----+
| tab_name |
+-----+-----+
| categories
| customers
| departments
| order_items
| orders
| products
+-----+-----+

select count(*) from order_items;

+-----+-----+
| _c0  |
+-----+-----+
| 172198 |
+-----+-----+
```



# Importação Incremental

- ❖ Importa apenas novos registros
- ❖ No mysql

```
select * from categories;
```

```
|      58 |                      8 | NFL Players          |
+-----+-----+
58 rows in set (0.00 sec)
```

```
insert into categories values(59,8,"Teste");
```



# Importação Incremental

```
sqoop import --connect
jdbc:mysql://localhost/retail_db --
table categories --username root --
password cloudera --hive-import --hive-
database retail_db --check-column
category_id --incremental append --
last-value 58
```

# Checar Hive

```
select * from categories;
```

59	8	Teste	
+	-----+	-----+	-----+



# Atualização

- ❖ Suportado pelo SQOOP
- ❖ Até o momento não suportado para Hive
- ❖ Necessário campo data

# Atualização

- ❖ Necessário campo data
- ❖ Usando tabela orders, campo order\_date
- ❖ Maior valor > 2014-07-24

order_id	order_date	order_customer_id	order_status
57752	2014-07-24 00:00:00	1618	COMPLETE

- ❖ update orders set order\_date = '2014-07-25', order\_customer\_id=1167 where order\_id = 57752;
- ❖ select \* from orders where order\_id = 57752;

order_id	order_date	order_customer_id	order_status
57752	2014-07-25 00:00:00	1167	COMPLETE



# Atualização

- ❖ Necessário campo data
- ❖ Não suportado pelo Hive
- ❖ Pode ser feito diretamente ao arquivo no HDFS
- ❖ `describe formatted orders;`

col_name	data_type	comment
# col_name	data_type	comment
order_id	NULL	NULL
order_date	int	
order_customer_id	string	
order_status	int	
	string	
	NULL	
# Detailed Table Information		
Database:	NULL	NULL
Owner:	retail_db	NULL
CreateTime:	cloudera	NULL
LastAccessTime:	Mon Aug 05 17:36:12 PDT 2019	NULL
Protect Mode:	UNKNOWN	NULL
Retention:	None	NULL
Location:	0	NULL
Table Type:	hdfs://quickstart.cloudera:8020/user/hive/warehouse/retail_db.db/orders	NULL
	UNPARTED TABLE	NULL

# Atualização

- ❖ sqoop import --connect jdbc:mysql://localhost/retail\_db --table orders --username root --password cloudera --check-column order\_date --incremental lastmodified --last-value 2014-07-24 --target-dir /user/hive/warehouse/retail\_db.db/orders
- ❖ sqoop import --connect jdbc:mysql://localhost:3306/ydb --table yloc --username root -P --check-column rDate --incremental lastmodified --last-value 2014-01-25 --target-dir yloc/loc

