

Etapa Desafio Técnico - Data Engineer

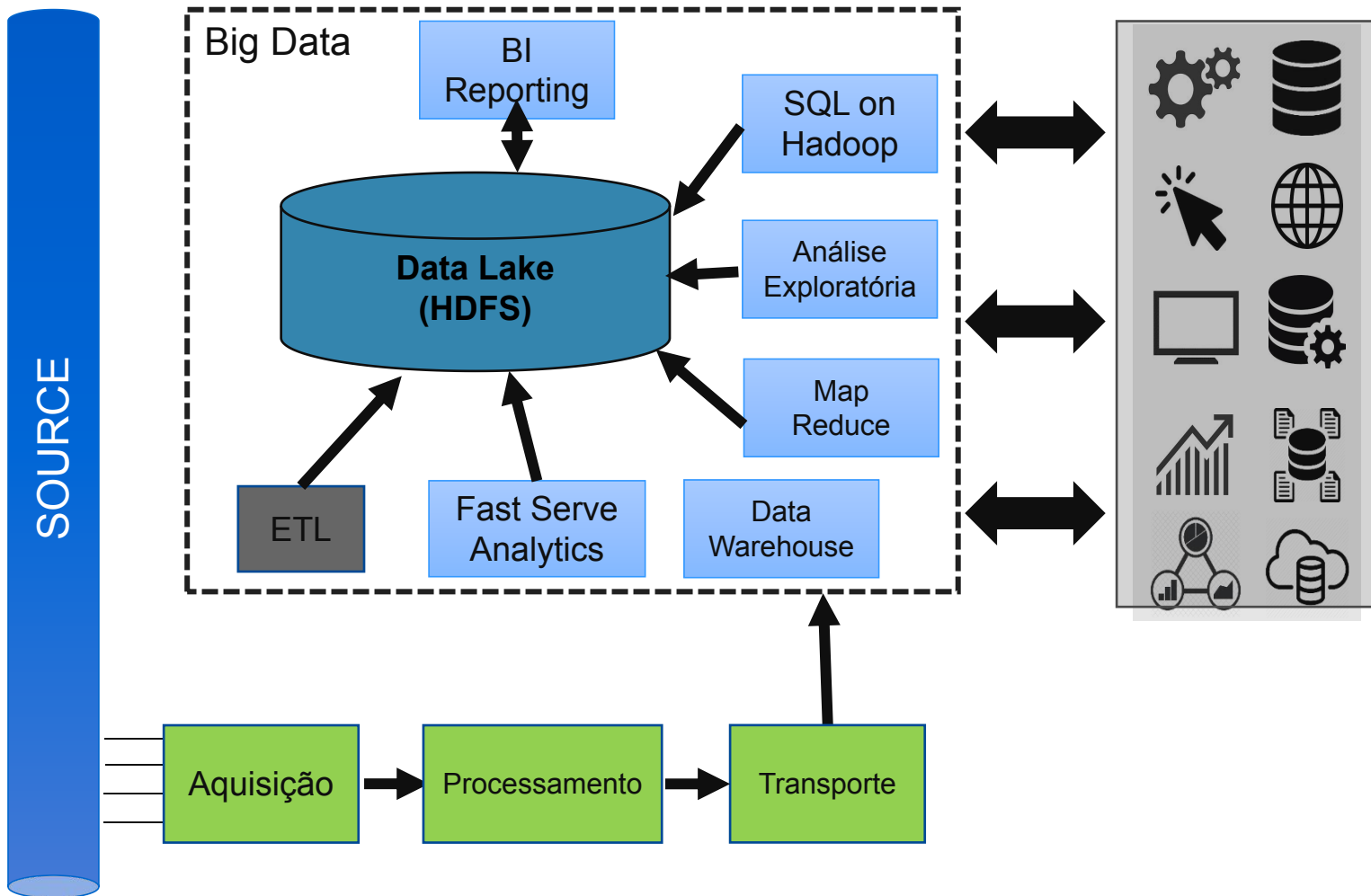
Rodrigo Romanzini

Agenda

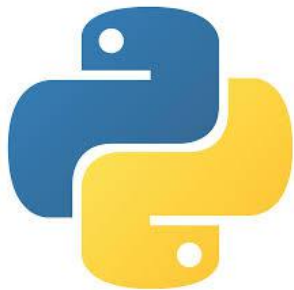
1. Objetivo
2. Arquitetura
3. Stack
4. Macro Fluxo Dados – *Data Lake*
5. Abordagem
6. Considerações

- Desenhar e implantar um novo DataWarehouse baseado em BigData.
- O Foco será na definição, implementação e implantação.
- A proposta inicial é demonstrar minhas habilidades em diversos cenários.

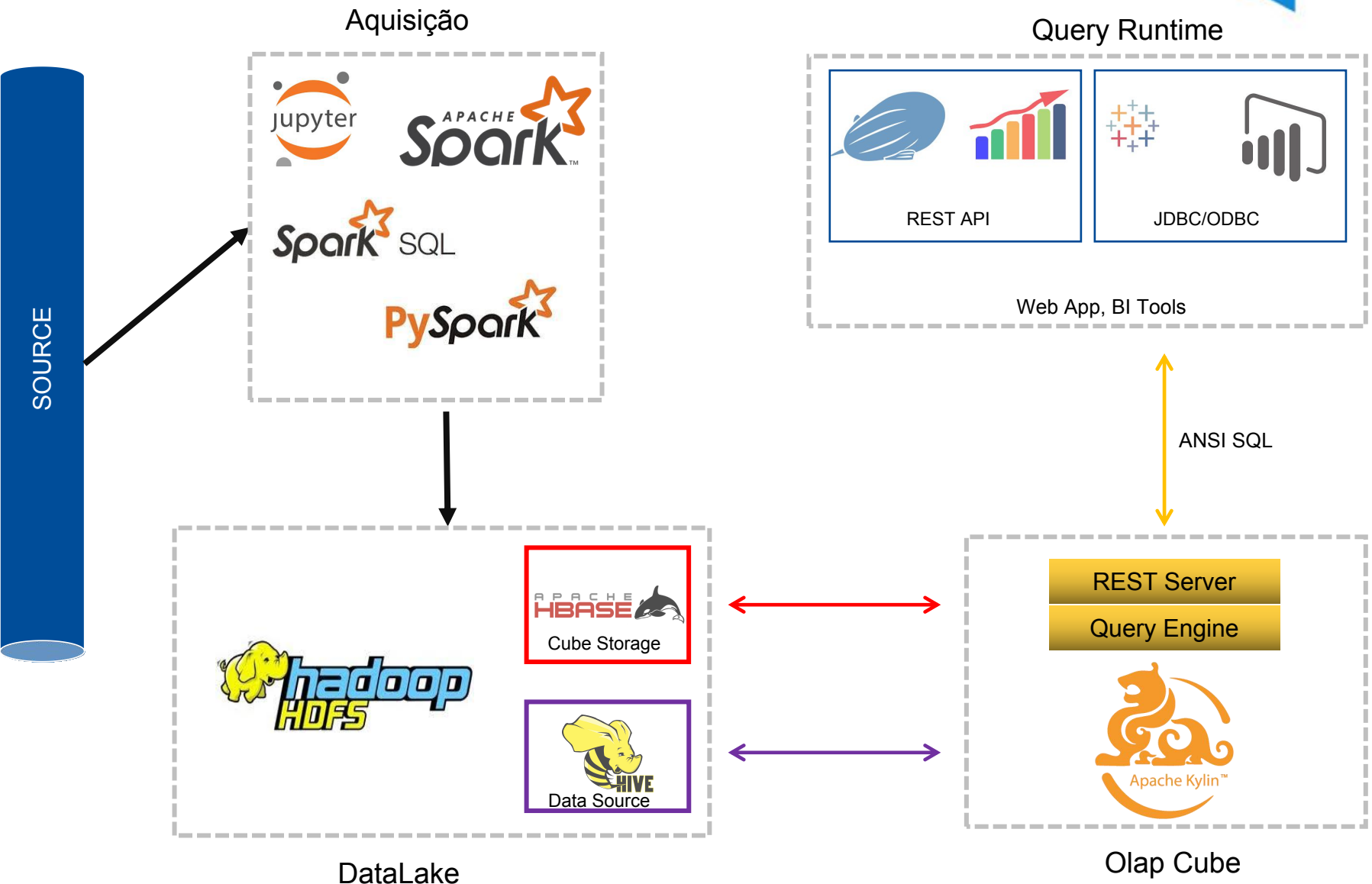
Arquitetura



Stack

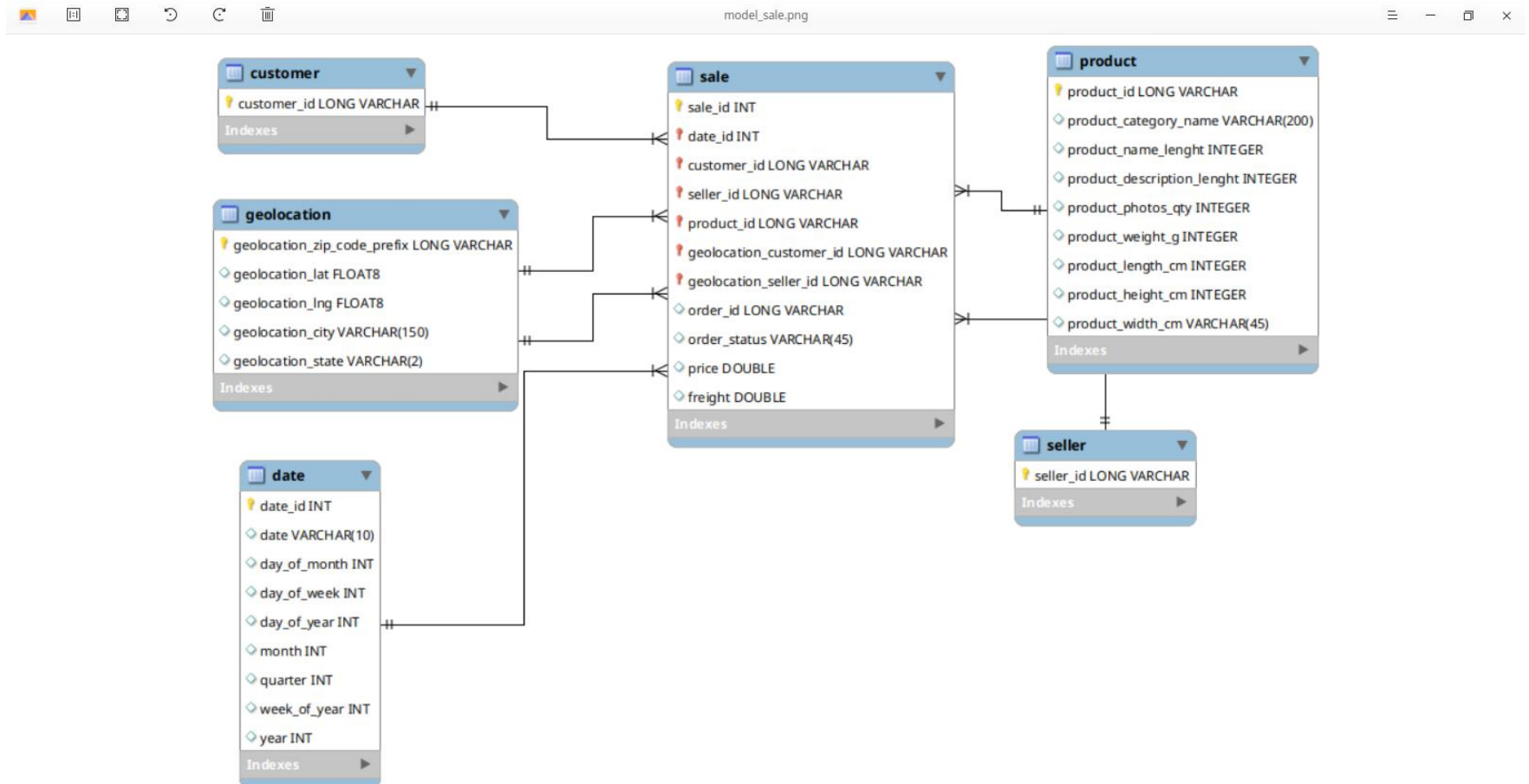


Macro Fluxo de Dados - DataLake



Abordagem

MODELO DE DADOS



Abordagem

AQUISIÇÃO DE DADOS

jupyter work-at-olist-data Last Checkpoint: Última Quarta-feira às 12:34 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3

Run

Spark SQL

O Spark SQL é usado para acessar dados estruturados com Spark.

```
In [1]: from pyspark.sql import SparkSession
        from pyspark.sql import SQLContext
        from pyspark.sql import Row
```

```
In [2]: # Spark Session - usada quando se trabalha com Dataframes no Spark
        spSession = SparkSession.builder.master("local").appName("Work-At-Olist-Data").config("spark.some.config.option", "some-value").getOrCreate()
```

```
In [3]: # Criando o SQL Context para trabalhar com Spark SQL
        sqlContext = SQLContext(sc)
```

```
In [4]: # Importando o arquivo e criando um DF
        customerDF = spSession.read.csv('datasets/olist_customers_dataset.csv',inferSchema=True, header = True)
        geolocationDF = spSession.read.csv('datasets/olist_geolocation_dataset.csv',inferSchema=True, header = True)
        orderItemDF = spSession.read.csv('datasets/olist_order_items_dataset.csv',inferSchema=True, header = True)
        orderDF = spSession.read.csv('datasets/olist_orders_dataset.csv',inferSchema=True, header = True)
        productDF = spSession.read.csv('datasets/olist_products_dataset.csv',inferSchema=True, header = True)
        sellerDF = spSession.read.csv('datasets/olist_sellers_dataset.csv',inferSchema=True, header = True)
```

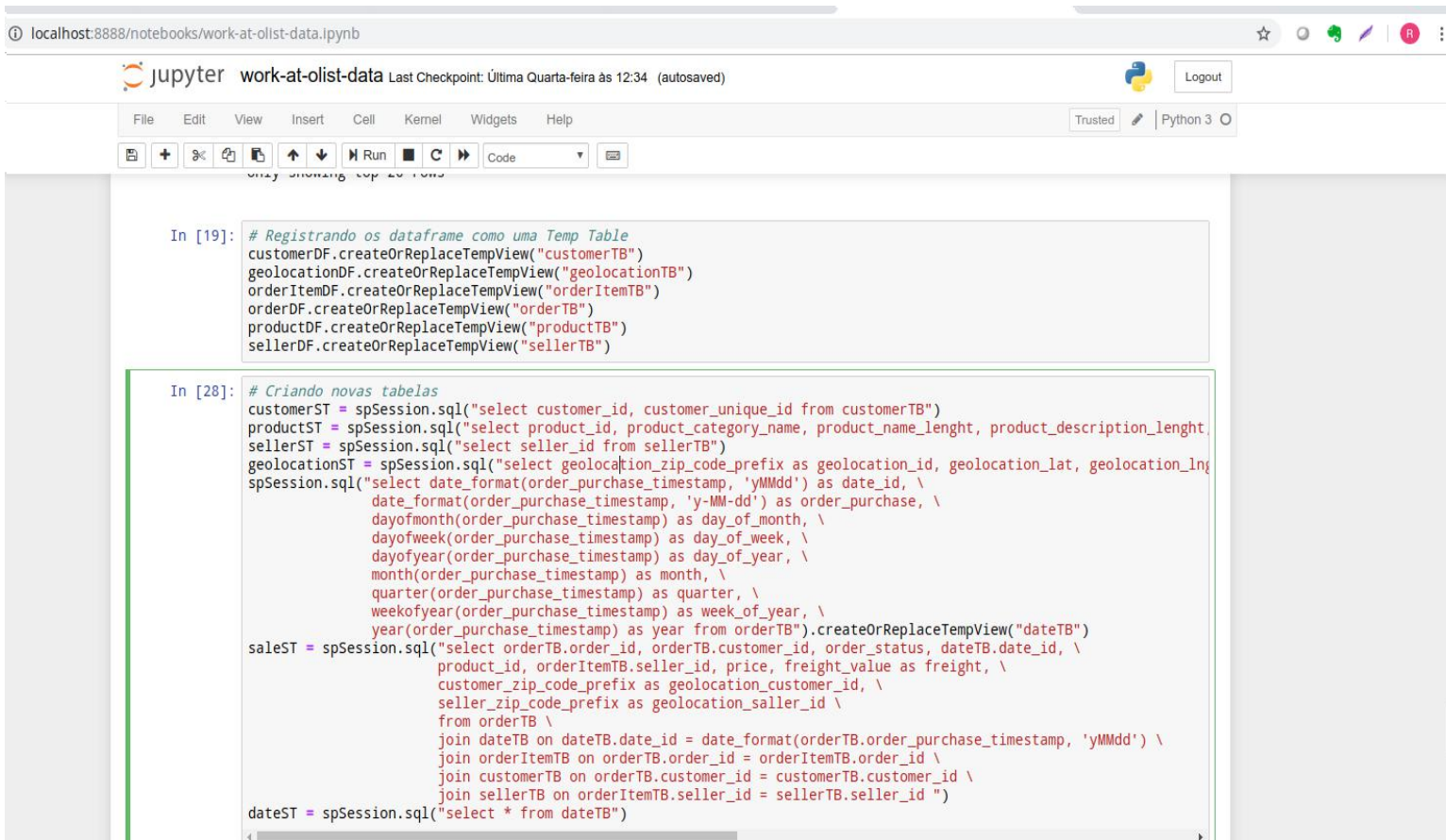
```
In [ ]: # Verificando o schema
```

```
In [5]: customerDF.dtypes
```

```
Out[5]: [('customer_id', 'string'),
         ('customer_unique_id', 'string'),
         ('customer_zip_code_prefix', 'int'),
         ('customer_city', 'string'),
         ('customer_state', 'string')]
```


Abordagem

PROCESSAMENTO DE DADOS

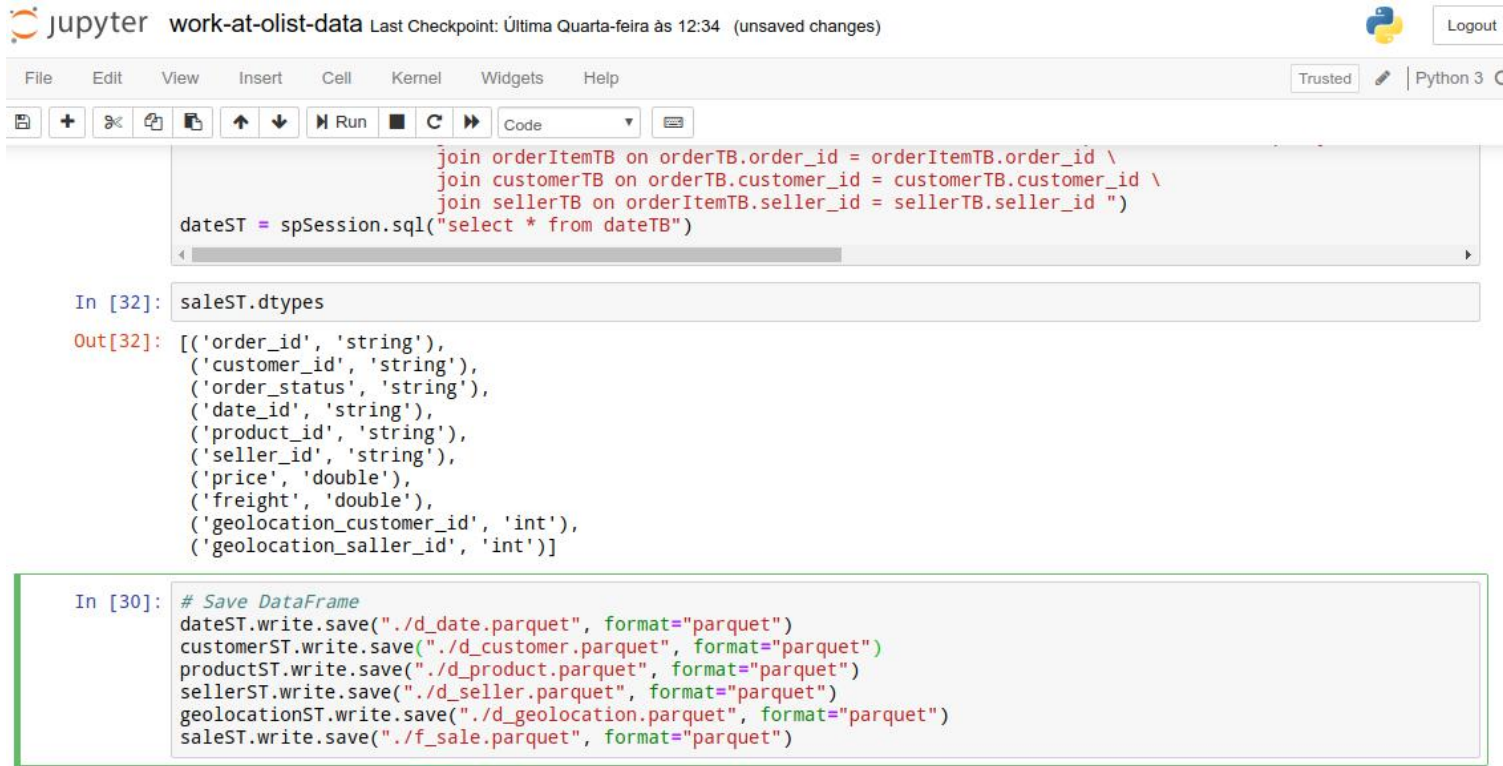


```
In [19]: # Registrando os dataframe como uma Temp Table
customerDF.createOrReplaceTempView("customerTB")
geolocationDF.createOrReplaceTempView("geolocationTB")
orderItemDF.createOrReplaceTempView("orderItemTB")
orderDF.createOrReplaceTempView("orderTB")
productDF.createOrReplaceTempView("productTB")
sellerDF.createOrReplaceTempView("sellerTB")

In [28]: # Criando novas tabelas
customerST = spSession.sql("select customer_id, customer_unique_id from customerTB")
productST = spSession.sql("select product_id, product_category_name, product_name_lenght, product_description_lenght,
sellerST = spSession.sql("select seller_id from sellerTB")
geolocationST = spSession.sql("select geolocation_zip_code_prefix as geolocation_id, geolocation_lat, geolocation_lng
spSession.sql("select date_format(order_purchase_timestamp, 'yMMdd') as date_id, \
    date_format(order_purchase_timestamp, 'y-MM-dd') as order_purchase, \
    dayofmonth(order_purchase_timestamp) as day_of_month, \
    dayofweek(order_purchase_timestamp) as day_of_week, \
    dayofyear(order_purchase_timestamp) as day_of_year, \
    month(order_purchase_timestamp) as month, \
    quarter(order_purchase_timestamp) as quarter, \
    weekofyear(order_purchase_timestamp) as week_of_year, \
    year(order_purchase_timestamp) as year from orderTB").createOrReplaceTempView("dateTB")
saleST = spSession.sql("select orderTB.order_id, orderTB.customer_id, orderTB.order_status, dateTB.date_id, \
    product_id, orderItemTB.seller_id, price, freight_value as freight, \
    customer_zip_code_prefix as geolocation_customer_id, \
    seller_zip_code_prefix as geolocation_saller_id \
    from orderTB \
    join dateTB on dateTB.date_id = date_format(orderTB.order_purchase_timestamp, 'yMMdd') \
    join orderItemTB on orderTB.order_id = orderItemTB.order_id \
    join customerTB on orderTB.customer_id = customerTB.customer_id \
    join sellerTB on orderItemTB.seller_id = sellerTB.seller_id ")
dateST = spSession.sql("select * from dateTB")
```

Abordagem

TRANSPORTE DE DADOS



The image shows a Jupyter Notebook interface with the title "work-at-olist-data". The top bar indicates the last checkpoint was on Thursday at 12:34 with unsaved changes. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook contains three code cells. The first cell has a SQL query with joins and a select statement. The second cell shows the output of a dtype check for a DataFrame. The third cell contains code to save multiple DataFrames to parquet files.

```
join orderItemTB on orderTB.order_id = orderItemTB.order_id \
join customerTB on orderTB.customer_id = customerTB.customer_id \
join sellerTB on orderItemTB.seller_id = sellerTB.seller_id ")
dateST = spSession.sql("select * from dateTB")

In [32]: saleST.dtypes
Out[32]: [('order_id', 'string'),
 ('customer_id', 'string'),
 ('order_status', 'string'),
 ('date_id', 'string'),
 ('product_id', 'string'),
 ('seller_id', 'string'),
 ('price', 'double'),
 ('freight', 'double'),
 ('geolocation_customer_id', 'int'),
 ('geolocation_saller_id', 'int')]

In [30]: # Save DataFrame
dateST.write.save("./d_date.parquet", format="parquet")
customerST.write.save("./d_customer.parquet", format="parquet")
productST.write.save("./d_product.parquet", format="parquet")
sellerST.write.save("./d_seller.parquet", format="parquet")
geolocationST.write.save("./d_geolocation.parquet", format="parquet")
saleST.write.save("./f_sale.parquet", format="parquet")
```

Abordagem

HIVE CREATE TABLES

cloudera-quickstart-vm-S.13.0-0-virtualbox [Executando] - Oracle VM VirtualBox

```
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
Applications  Places  System

cloudera@quickstart:~/Downloads

File Edit View Search Terminal Help

2019-10-16 23:17:33,074 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.89 sec
MapReduce Total cumulative CPU time: 7 seconds 890 msec
Ended Job = job_1571291603760_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.89 sec HDFS Read: 690200 HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 890 msec
OK
99441
Time taken: 37.848 seconds, Fetched: 1 row(s)
hive> create external table d_customer (customer_id STRING, customer_unique_id STRING)
> ROW FORMAT SERDE 'parquet.hive.serde.ParquetHiveSerDe'
> STORED AS
> INPUTFORMAT "parquet.hive.DeprecatedParquetInputFormat"
> OUTPUTFORMAT "parquet.hive.DeprecatedParquetOutputFormat"
> LOCATION '/user/hive/warehouse/d_customer';
OK
Time taken: 0.067 seconds
hive> create external table d_product (product_id STRING, product_category_name STRING, product_name_lenght int, product_description_lenght int, product_photos_qty int, product_weight_g int, product_length_cm int, product_height_cm int, product_width_cm int)
> ROW FORMAT SERDE 'parquet.hive.serde.ParquetHiveSerDe'
> STORED AS
> INPUTFORMAT "parquet.hive.DeprecatedParquetInputFormat"
> OUTPUTFORMAT "parquet.hive.DeprecatedParquetOutputFormat"
> LOCATION '/user/hive/warehouse/d_product';
OK
Time taken: 0.07 seconds
hive> create external table d_seller (seller_id STRING)
> ROW FORMAT SERDE 'parquet.hive.serde.ParquetHiveSerDe'
> STORED AS
> INPUTFORMAT "parquet.hive.DeprecatedParquetInputFormat"
> OUTPUTFORMAT "parquet.hive.DeprecatedParquetOutputFormat"
> LOCATION '/user/hive/warehouse/d_seller';
OK
Time taken: 0.059 seconds
hive> create external table d_geolocation (geolocation_id int, geolocation_lat double, geolocation_lng double, geolocation_city STRING, geolocation_state STRING)
> ROW FORMAT SERDE 'parquet.hive.serde.ParquetHiveSerDe'
> STORED AS
> INPUTFORMAT "parquet.hive.DeprecatedParquetInputFormat"
> OUTPUTFORMAT "parquet.hive.DeprecatedParquetOutputFormat"
> LOCATION '/user/hive/warehouse/d_geolocation';
OK
Time taken: 0.065 seconds
hive> create external table f_sale (order_id string, customer_id string, order_status string, date_id string, product_id string, seller_id string, price double, freight double, geolocation_customer_id int, geolocation_seller_id int)
> ROW FORMAT SERDE 'parquet.hive.serde.ParquetHiveSerDe'
> STORED AS
> INPUTFORMAT "parquet.hive.DeprecatedParquetInputFormat"
> OUTPUTFORMAT "parquet.hive.DeprecatedParquetOutputFormat"
> LOCATION '/user/hive/warehouse/f_sale';
OK
Time taken: 0.12 seconds
```

cloudera@quickstart:~

Abordagem

HIVE QUERY SQL

```
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Applications Places System

cloudera@quickstart:~/Downloads

File Edit View Search Terminal Help

1e9e8ef04dbcf4541ed26657ea517e5      perfumaria      40      287      1      225      16      10      14
3aa071139cb16b67ca9e5dea641aaa2f      artes      44      276      1      1000      30      18      20
96bd76ec8810374ed1b65e291975717f      esporte_lazer      46      250      1      154      18      9      15
cef67bcfe19066a932b7673e239eb23d      bebes      27      261      1      371      26      4      26
9dc1a7de27444849c219cffi95d0b71      utilidades_domesticas      37      402      4      625      20      17      13
41d3672d4792049fa1779bb35283ed13      instrumentos_musicais      60      745      1      200      38      5      11
732bd381ad09e530fe0a5f457d81becb      cool_stuff      56      1272      4      18350      70      24      44
2548af3e6e77a690cf3eb6368e9ab61e      moveis_decoracao      56      184      2      900      40      8      40
37cc742be07708b53a98702e77a21a02      eletrodomesticos      57      163      1      400      27      13      17
8c92109888e8cdf9d66dc7e463025574      brinquedos      36      1156      1      600      17      10      12
Time taken: 0.079 seconds, Fetched: 10 row(s)
hive> select * from d_seller limit 10;
OK
3442f8959a84dea7ee197c632cb2df15
dlb65fc7debc3361ea86b5f14c68d2e2
ce3ad9de960102d0677a81f5d0bb7b2d
c0f3eea2e14555b6faea3dd58c1blc3
51a04a8a6bdc23decc82b0880742cf
c240c4061717ac1806ae6ee72be3533b
e49c26c3edfa46d227d5121a6b6e4d37
1b938a7ec6ac5061a66a3766e0e75f90
768a86e36ad6aee3d03ee3c6433d61df
ccc4bbb5f32a6ab2b7066a4130f114e3
Time taken: 0.07 seconds, Fetched: 10 row(s)
hive> select * from d_geolocation limit 10;
OK
1037      -23.54562128115268      -46.63929204800168      sao paulo      SP
1046      -23.546081127035535      -46.64482029837157      sao paulo      SP
1046      -23.54612896641469      -46.64295148361138      sao paulo      SP
1041      -23.5443921648681      -46.63949930627844      sao paulo      SP
1035      -23.541577961711493      -46.64160722329613      sao paulo      SP
1012      -23.547762303364266      -46.63536053788448      sao paulo      SP
1047      -23.546273112412678      -46.64122516971552      sao paulo      SP
1013      -23.546923208436723      -46.6342636964915      sao paulo      SP
1029      -23.543769055769133      -46.63427784085132      sao paulo      SP
1011      -23.547639550320632      -46.63603162315495      sao paulo      SP
Time taken: 0.067 seconds, Fetched: 10 row(s)
hive> select * from f_sale limit 10;
OK
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      6782d593f63105318f46bbf7633279bf      325f3178fb58e2a9778334621eecdcbf9      27.9      3.81      26551      6790
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      e95ee6822b66ac6058e2e4afff656071a      a17f621c590ea0fab3d5d883e1630ec6      21.33      25.39      26551      18055
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      6782d593f63105318f46bbf7633279bf      325f3178fb58e2a9778334621eecdcbf9      27.9      3.81      26551      6790
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      e95ee6822b66ac6058e2e4afff656071a      a17f621c590ea0fab3d5d883e1630ec6      21.33      25.39      26551      18055
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      6782d593f63105318f46bbf7633279bf      325f3178fb58e2a9778334621eecdcbf9      27.9      3.81      26551      6790
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      e95ee6822b66ac6058e2e4afff656071a      a17f621c590ea0fab3d5d883e1630ec6      21.33      25.39      26551      18055
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      6782d593f63105318f46bbf7633279bf      325f3178fb58e2a9778334621eecdcbf9      27.9      3.81      26551      6790
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      e95ee6822b66ac6058e2e4afff656071a      a17f621c590ea0fab3d5d883e1630ec6      21.33      25.39      26551      18055
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      6782d593f63105318f46bbf7633279bf      325f3178fb58e2a9778334621eecdcbf9      27.9      3.81      26551      6790
014405982914c2cde2796ddcf0b8703d      2de342d6e5905a5a8bb3a991c855f3e2      delivered      20170726      e95ee6822b66ac6058e2e4afff656071a      a17f621c590ea0fab3d5d883e1630ec6      21.33      25.39      26551      18055
Time taken: 0.086 seconds, Fetched: 10 row(s)
hive>
```

Abordagem

JOBS RUNNING

Apache Kylin : Implementing OLAP

Kylin

RUNNING Applications

localhost:8088/cluster/apps/RUNNING

hadoop

RUNNING Applications

Logged in as: dr.who

Cluster

About Nodes Applications

NEW SAVING SUBMITTED ACCEPTED RUNNING FINISHED FAILED KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
13	6	1	6	7	8 GB	8 GB	0 B	7	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCores Reserved
0	0	0	0	0	0	0	0 B	0 B	0 B	0	0	0

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Reserved CPU VCoers	Reserved Memory MB	Progress	Tracking UI
application_1571291603760_0003	cloudera	Kylin Hive Column Cardinality Job table=DEFAULT.F_SALE output=hdfs://quickstart.cloudera:8020/kylin/kylin_metadata/cardinality/e9531551-a1eb-49c9-9f30-92d65340604d/DEFAULT.F_SALE project=SalesModel	MAPREDUCE	root.cloudera	Thu Oct 17 03:24:46 -0300 2019	N/A	RUNNING	UNDEFINED	7	7	8192	0	0		ApplicationMaster

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Abordagem

CRIAÇÃO DO CUBO

The screenshot displays the Apache Kylin web interface in a browser. The address bar shows 'localhost:7070/kylin/models'. The top navigation bar includes 'Kylin', 'SalesProject', 'Insight', 'Model' (selected), 'Monitor', and 'System'. On the right, there are links for 'Help' and 'Welcome, ADMIN'. Below the navigation bar, there are search and add icons. The main content area is divided into two panels. The left panel, titled 'Tables', shows a tree view of data sources under 'DEFAULT', including 'D_CUSTOMER', 'D_DATE', 'D_GEOLOCATION', 'D_PRODUCT', 'D_SELLER', and 'F_SALE' (selected). The right panel, titled 'Table Schema:F_SALE', shows the table's columns and their data types. It includes tabs for 'Columns', 'Extend Information', and 'Access'. There are buttons for 'Reload Table' and 'Unload Table'. A search filter is also present. The table lists 10 columns with their IDs, names, data types, cardinalities, and comments.

ID	Name	Data Type	Cardinality	Comment
1	ORDER_ID	varchar(256)		
2	CUSTOMER_ID	varchar(256)		
3	ORDER_STATUS	varchar(256)		
4	DATE_ID	varchar(256)		
5	PRODUCT_ID	varchar(256)		
6	SELLER_ID	varchar(256)		
7	PRICE	double		
8	FREIGHT	double		
9	GEOLOCATION_CUSTOMER_ID	integer		
10	GEOLOCATION_SALLER_ID	integer		

Apache Kylin | Apache Kylin Community

Abordagem

CRIAÇÃO DO CUBO

The screenshot displays the Apache Kylin web interface for creating a new model. The browser address bar shows 'localhost:7070/kylin/models/add'. The top navigation bar includes the Kylin logo, a dropdown menu set to 'SalesProject', and tabs for 'Insight', 'Model' (which is the active tab), 'Monitor', and 'System'. On the right of the navigation bar are links for 'Help' and 'Welcome, ADMIN'.

The main content area is titled 'Model Designer' and features a five-step progress bar:

- Model Info**: The current step, indicated by a green checkmark in a circle.
- Data Model**: Indicated by a circle with the number 2.
- Dimensions**: Indicated by a circle with the number 3.
- Measures**: Indicated by a circle with the number 4.
- Settings**: Indicated by a circle with the number 5.

Below the progress bar, there are two input fields:

- Model Name**: A text input field containing the value 'SalesModel'. A red asterisk indicates this field is required.
- Description**: A larger text area for providing a description of the model.

A green 'Next' button with a right-pointing arrow is located at the bottom right of the form.

Abordagem

CRIAÇÃO DO CUBO

Kylin RUNNING Applications localhost:7070/kylin/models/add











Kylin SalesProject Insight Model Monitor System Help Welcome, ADMIN

Model Designer

Model Info 2 Data Model 3 Dimensions 4 Measures 5 Settings

Fact Table * DEFAULT.F_SALE

+ Add Lookup Table Filter ...

ID	Table Alias	Table Name	Table Kind	Join Type	Join Condition	Actions
1	D_PRODUCT	DEFAULT.D_PRODUCT	Normal	inner	F_SALE.PRODUCT_ID = D_PRODUCT.PRODUCT_ID	 
2	D_DATE	DEFAULT.D_DATE	Normal	inner	F_SALE.DATE_ID = D_DATE.DATE_ID	 
3	D_CUSTOMER	DEFAULT.D_CUSTOMER	Normal	inner	F_SALE.CUSTOMER_ID = D_CUSTOMER.CUSTOMER_ID	 
4	D_SELLER	DEFAULT.D_SELLER	Normal	inner	F_SALE.SELLER_ID = D_SELLER.SELLER_ID	 
5	D_GEOLOCATION	DEFAULT.D_GEOLOCATION	Normal	inner	F_SALE.GEOLOCATION_CUSTOMER_ID = D_GEOLOCATION.GEOLOCATION_ID	 

← Prev Next →

Apache Kylin | Apache Kylin Community

Abordagem

CRIAÇÃO DO CUBO

The screenshot displays the Apache Kylin web interface for creating a cube. The browser address bar shows `localhost:7070/kylin/models/add`. The top navigation bar includes tabs for **Insight**, **Model** (active), **Monitor**, and **System**. A progress bar at the top indicates the current step is **3 Dimensions**, with previous steps **Model Info** and **Data Model** completed, and subsequent steps **Measures** and **Settings** pending.

The main content area is titled "Select dimension columns" and contains a table with 6 rows. Each row represents a dimension table with its ID, table alias, and a list of selected columns.

ID	Table Alias	Columns
1	F_SALE	Select Column...
2	D_PRODUCT	PRODUCT_CATEGORY_NAME
3	D_DATE	ORDER_PURCHASE, DAY_OF_WEEK, MONTH, QUARTER, YEAR
4	D_CUSTOMER	Select Column...
5	D_SELLER	Select Column...
6	D_GEOLOCATION	GEOLOCATION_CITY, GEOLOCATION_STATE

At the bottom right of the table, there are navigation buttons: **← Prev** and **Next →**.

The footer of the page displays the Apache Kylin logo and the text "Apache Kylin | Apache Kylin Community".

Abordagem

CRIAÇÃO DO CUBO

The screenshot displays the Apache Kylin web interface for creating a cube. The browser address bar shows `localhost:7070/kylin/cubes/add/`. The top navigation bar includes the Kylin logo, a project dropdown set to 'SalesProject', and tabs for 'Insight', 'Model' (active), 'Monitor', and 'System'. On the right of the navigation bar are links for 'Help' and 'Welcome, ADMIN'.

On the left sidebar, there is a '+ New' button and a 'Models' section containing a box labeled 'SalesModel'.

The main area is titled 'Cube Designer' and features a progress bar with seven steps: 1. Cube Info (active), 2. Dimensions, 3. Measures, 4. Refresh Setting, 5. Advanced Setting, 6. Configuration Overwrites, and 7. Overview.

The 'Cube Info' step contains the following form fields:

- Model Name ***: A dropdown menu currently showing 'SalesModel'.
- Cube Name ***: A text input field containing 'SalesCube'.
- Notification Email List**: A text input field containing 'Comma Separated'.
- Notification Events ***: A selection field with three options: 'ERROR', 'DISCARDED', and 'SUCCEED'.
- Description**: A large text area for additional information.

A green 'Next' button with a right arrow is located at the bottom right of the form.

Abordagem

CRIAÇÃO DO CUBO

The screenshot displays the Apache Kylin web interface for creating a cube. The browser address bar shows `localhost:7070/kylin/cubes/add/`. The navigation bar includes tabs for **Kylin**, **SalesProject**, **Insight**, **Model** (active), **Monitor**, and **System**. On the left, there is a **+ New** button and a **Models** section containing **SalesModel**.

The main area is the **Cube Designer**, which features a progress bar with seven steps: **Cube Info**, **Dimensions**, **Measures** (current step, indicated by a blue circle with the number 3), **Refresh Setting**, **Advanced Setting**, **Configuration Overwrites**, and **Overview**.

Below the progress bar is a table of measures:

Name	Expression	Parameters	Return Type	Actions
<code>_COUNT_</code>	COUNT	<code>__Value:1, Type:constant</code>	bigint	
TOTAL_SALES	SUM	<code>__Value:F_SALE.PRICE, Type:column</code>	double	
TOTAL_FREIGHT	SUM	<code>__Value:F_SALE.FREIGHT, Type:column</code>	double	

At the bottom left of the table is a **+ Measure** button. At the bottom right are **← Prev** and **Next →** navigation buttons.

Abordagem

CRIAÇÃO DO CUBO

The screenshot displays the Apache Kylin Monitor web interface. The top navigation bar includes tabs for 'SalesProject', 'Insight', 'Model', 'Monitor' (active), and 'System'. The main content area is divided into two sections: 'Jobs' and 'Slow Queries'. The 'Jobs' section shows a table of jobs with filters for 'Cube Name' and 'Jobs in' (LAST ONE WEEK, ALL, NEW, PENDING, RUNNING, STOPPED, FINISHED, ERROR, DISCARDED). A single job is listed: 'BUILD CUBE - SalesCube - 20160901000000_20161231000000 - GMT+08:00 2019-10-18 04:40:26'. The job's progress is 100%, and it took 22.17 minutes. The 'Detail Information' panel on the right provides more details about the job, including its ID, status (FINISHED), duration, and MapReduce waiting time. The bottom of the interface shows the start time of the job and the first step name: 'Create Intermediate Flat Hive Table' with a data size of 75.14 MB.

Jobs Slow Queries

Cube Name: Filter ...

Jobs in: LAST ONE WEEK ALL NEW PENDING RUNNING STOPPED FINISHED ERROR DISCARDED

Job Name	Cube	Progress	Last Modified Time	Duration	Actions
BUILD CUBE - SalesCube - 20160901000000_20161231000000 - GMT+08:00 2019-10-18 04:40:26	SalesCube	100%	2019-10-18 05:02:45 GMT+8	22.17 mins	Action

Total: 1

Detail Information

Job Name	BUILD CUBE - SalesCube - 20160901000000_20161231000000 - GMT+08:00 2019-10-18 04:40:26
Job ID	aea635ec-c733-4a9a-a71c-f32b6f35cfdc
Status	FINISHED
Duration	22.17 mins
MapReduce Waiting	4.15 mins

Start 2019-10-18 04:40:34 GMT+8

2019-10-18 04:40:34 GMT+8

#1 Step Name: Create Intermediate Flat Hive Table
Data Size: 75.14 MB

Apache Kylin | Apache Kylin Community

Abordagem

EXECUÇÃO DE QUERY

The screenshot displays the Apache Kylin web interface in a browser. The address bar shows `localhost:7070/kylin/query#query_content_results`. The interface includes a top navigation bar with tabs for 'SalesProject', 'Insight', 'Model', 'Monitor', and 'System'. A left sidebar lists 'Tables' under 'DEFAULT', including 'D_DATE' and 'F_SALE'. The main area is divided into 'New Query', 'Saved Queries', and 'Query History' sections. The 'New Query' section contains a SQL query:

```
1 select date_id, avg(price) as Avg_Price
2 from F_SALE
3 group by date_id
4 order by date_id
5
```

Below the query editor, a tip suggests using keyboard shortcuts to list keywords. The 'Project' is set to 'SalesProject', and a 'LIMIT' of 50000 is specified. The 'Results' section shows a grid of 23 rows, each with a status icon (green checkmark for success, yellow triangle for warning, or red X for error). The 'Query String' section displays the start time (2019-10-18 21:29:04 GMT+8), duration (0.48s), and buttons for 'Rerun' and 'Save'. The bottom status bar indicates 'Status: Success', 'Project: SalesProject', and 'Cubes: CUBE(name=SalesCube)'.

Tables

- DEFAULT
 - D_DATE
 - F_SALE

New Query Saved Queries Query History

```
1 select date_id, avg(price) as Avg_Price
2 from F_SALE
3 group by date_id
4 order by date_id
5
```

Tips: Ctrl+Shift+Space or Alt+Space(Windows), Command+Option+Space(Mac) to list keywords in query box.

Project: SalesProject LIMIT 50000 Submit

Results

1 ⚠	2 ⚠	3 ⚠	4 ✓	5 ⚠	6 ✓	7 ⚠	8 ⚠	9 ✓	10 ⚠	11 ⚠
12 ⚠	13 ✓	14 ✓	15 ✓	16 ⚠	17 ⚠	18 ⚠	19 ⚠	20 ✓	21 ✓	22 ⚠
23 ✓										

Status: All

Query String Start Time: 2019-10-18 21:29:04 GMT+8 Duration: 0.48s Rerun Save

Status: Success Project: SalesProject Cubes: CUBE(name=SalesCube)

Apache Kylin | Apache Kylin Community

Abordagem

REQUEST API - CUBO

The screenshot displays the ARC (Apache Request Console) interface. The top bar is blue with the 'ARC' logo on the left and an information icon on the right. The main area is divided into a left sidebar and a right main panel.

Left Sidebar:

- HTTP request:** A list of recent requests. The selected request is a GET request to `http://localhost:7070/kylin/api/cubes`.
- Socket:** A section for socket connections.
- History:** A list of historical requests.
- Saved:** A list of saved requests, including 'Get SalesCube' and 'Get Avg Price'.
- Projects:** A dropdown menu for selecting projects.

Main Panel:

The main panel is titled 'Request' and shows the details of the selected request. It includes a 'Method' dropdown set to 'POST' and a 'Request URL' field containing `http://localhost:7070/kylin/api/query`. A 'SEND' button is located to the right of the URL.

Below the URL, there are tabs for 'Parameters', 'Headers', 'Body', and 'Variables'. The 'Body' tab is currently selected. The 'Body content type' is set to 'application/json', and the 'Editor view' is set to 'Raw input'.

The 'Body' tab shows a JSON payload:

```
{  "sql": "select date_id, avg(price) as Avg_Price from F_SALE group by date_id order by date_id ",  "offset": 0,  "limit": 50000,  "acceptPartial": false,  "project": "SalesProject"}
```

Below the JSON payload, the status bar shows '200 OK' and '813.76 ms'. A 'DETAILS' link is available to the right.

At the bottom of the interface, there is a green banner that reads 'Install new ARC with new features!' and a status bar indicating 'Selected environment: Default'.

Abordagem

REQUEST API - CUBO RESPONSE

The screenshot displays the ARC (API Request Client) interface. The top bar is blue with the ARC logo and the title "Request". On the left, there is a sidebar with sections: "HTTP request", "Socket", "History", "Saved", and "Projects". The "History" section is expanded, showing a list of requests. The selected request is a GET request to "http://localhost:7070/kylin/api/cubes". The main area shows the request details, including the method (GET), URL, and the response body. The response is a JSON object with a status of "200 OK" and a response time of "813.76 ms". The response body contains a list of results, each with a date_id and an average price.

Request Details:

- Method: GET
- URL: http://localhost:7070/kylin/api/cubes
- Status: 200 OK
- Response Time: 813.76 ms

Request Body (JSON):

```
{
  "sql": "select date_id, avg(price) as Avg_Price from F_SALE group by date_id order by date_id ",
  "offset": 0,
  "limit": 50000,
  "acceptPartial": false,
  "project": "SalesProject"
}
```

Response Body (JSON):

```
{
  "columnMetas": Array[2] ...
  "results": [Array[13]]
  -0: [Array[2]]
    0: "20160904",
    1: "36.445"
  -1: [Array[2]]
    0: "20160905",
    1: "59.5"
  -2: [Array[2]]
    0: "20160915",
    1: "59.5"
}
```

Selected environment: Default

Considerações

- Gostaríamos de analisar suas habilidades com SQL, modelagem dimensional e integração de dados. Mostre seu conhecimento em processos de ETL e conceitos de Data Warehouse? Que tal replicar nossos datasets, remodelar em um banco de dados e apresentar as melhorias realizadas em sua criação?
 - Conforme demonstrado acima, minha proposta foi a construção de uma Data Warehouse em ambiente de BigData.
- É possível utilizar o modelo proposto em um ambiente cloud? Quais plataformas ou serviços você utilizaria? Quais as vantagens do modelo escolhido em questões de performance?
 - Sim seria possível a utilização do modelo proposto em cloud, poderia ser implementado tanto na Amazon utilizando os serviços do EMR quando a Azure através do HDINSIGHT
- Alguns membros do time dizem que a atual modelagem do banco de dados é adequada para o uso dos cientistas de dados e analistas de BI, porém, outros dizem que existem formas de modelar bancos de dados que trarão mais eficiência. Qual é a sua opinião sobre isso?
 - Através da arquitetura proposta acima, seria possível atender os times de cientistas de dados e analistas de BI.
- Estamos preocupados com o vertiginoso aumento do volume em nosso banco de dados atual? Você consideraria uma opção mais escalável ou devemos manter a estrutura existente?
 - Consideraria uma opção mais escalável. A sugestão seria a utilização proposta acima de um ambiente em BigData que poderá dar mais escalabilidade horizontal. A utilização de uma plataforma como a Amazon EMR poderia ser uma alternativa.
- Nossa ferramenta de visualização de dashboards está lenta e o nosso time detectou que o problema está na infraestrutura de dados. Como você abordaria esta situação do ponto de vista de arquitetura de dados?
 - Sugiro a utilização de uma ferramenta como o Kylin que poderia ser a solução do problema de infraestrutura de dados, pois os cubos são processados e armazenados no HBASE e então disponibilizados através de chamadas REST API ou jdbc/odbc.
- Nosso banco de dados está hospedado na nuvem e nossas ferramentas de análise de dados são "on premisses". Você manteria este arranjo ou faria mudanças visando mais performance?
 - Utilizaria a ferramenta Kylin implementada na nuvem e as ferramentas de análise de dados passariam a fazer chamadas de REST API.

Considerações

- Nossa área operacional necessita de informações em tempo real, porém os diretores da empresa, que acompanham somente informações de KPIs mensais, alegam que isso é desnecessário e acarretaria custos. Qual é o seu posicionamento sobre isso?
 - Partindo do pressuposto que o modelo implementado acima estaria em produção, poderia ser implantado um Near Real Time com uma janela de 60 minutos por exemplo, bastaria para isso implementar um broker kafka onde os sistemas passariam a gerar eventos das transações em tópicos e ajustar a camada de aquisição, os demais fluxos já estariam preparados para trabalhar com Real Time. Com essa abordagem, conseguiríamos atender tanto a área operacional com informações em Near Real Time e também deixariamos os diretores satisfeitos, pois os custos não seriam altos.
- Nosso time que está focado em Governança de Dados alega que documentar os processos é mais importante do que refatorar os mais de 500 scripts que estão funcionando com lentidão. Como você atuaria neste impasse, se tivesse que priorizar o trabalho?
 - Focaria em refatorar os scripts baseado na arquitetura proposta já com Near Real Time para dar mais agilidade ao negócio e em conjunto trabalhar com o time de Governança de Dados para documentar os processos.
- Aqui no olist, somos muito mão na massa! Como Engenheiro(a) de dados, mostre pra gente o que você consegue fazer na prática com esse nosso banco de dados. (Sabemos que é uma amostra, mas imagine que o todo pode ser petabytes de dados)
 - Neste documento foquei em mostrar as atividades realizadas durante os trabalhos, caso julguem necessário, teria imenso prazer em ir pessoalmente até a Olist e levar meu notebook para mostrar “in loco” todo o trabalho realizado.

A blue speech bubble with a white border and a grey vertical bar on the left. The word "Thanks!" is written in white inside the bubble.

Thanks!