NOTE while running hadoop job, you have to pass output folder path Which Doesn't exist on HDFS.

Step 1:

First of all we need to parse the data for both OSM as well as PokemonGo KML.

Step 2:

First, we will parse the data for Pokemongo.kml file.
In order to parse an KML file use KMLParser java file that is provided with the project.
Just provide input File as first argument and Output File Path as second argument for parsing files.

Step 3:
Once the data is parsed, you will get the parsed data file, you need to put that parsed data file in HDFS.
In my case I placed in to /input/pokemon/parsed-data.txt

Step 4: (Please see hbase.sql)
Now create a table in Hbase,

I have created the table using the Script which is present in the hbase.sql file(I have written step number above that line).

Step 5:
Export the Jar file of the project and specify Main class `PokemonGoBulkLoadDriver` while generating MANIFEST.MF as main class in eclipse.

Once the Jar file is exported, you need to export your hbase class path as HADOOP_CLASSPATH so that HADOOP can find the necessary files from HBase libs.

I have used below command in terminal for that purpose.

```
export HADOOP_CLASSPATH=`hbase classpath`
```

Now you can run Jar file using below command,

```
hadoop jar <InputFilePath> <OutputFilePath>
```

I have used, below command.

```
hadoop jar pokemonDataLoader.jar /input/fi/pokemongo/parsed-pokemon.txt
/output/pokemongo
```

Now the Process will start and if it completes without any errors then the Data is successfully loaded in the Hbase.

Here in this case we have used Hbase Bulk Load Feature, which is Extremely faster than Ordinary API.

Now the Data is present in Hbase.

Here you can fire simple queries to analyze the Hbase data.

Step 6:

Once the data is present in the Hbase we can also access that data in Hive using Hbase Hive integreation.

Step 7: (Please see hbase-hive-integration.sql)
Perform Step 7 from hbase-hive-integration.sql file as I have mentioned the steps to perform.
Please update the PATHS accordingly as per your installation just follow my instruction and your paths.

Step 8: (Please see hive.sql)

perform STEP 8 from hive.sql file and create a hive table which maps to Hbase table.

Step 9:(Please see hive.sql)
Step 9 is all the required queries for PokemonGo as well as Gyms perform them in Hive Shell and get the outputs.

I will also send the outputs in other file.

Step 10:(Please see hbase.sql)
Create Hbase tables for OpenStreetMap nodes and ways.

Step 11:
Export the Jar file of the project and specify Main class `OpenStreetMapNodesBulkLoadDriver` while generating MANIFEST.MF as main class in eclipse.
Now you can run Jar file using below command,

```
hadoop jar <InputFilePath> <OutputFilePath>
```

Now the Process will start and if it completes without any errors then the Data is successfully loaded in the Hbase.

Step 12:
Export the Jar file of the project and specify Main class `OpenStreetMapWaysBulkLoadDriver` while generating MANIFEST.MF as main class in eclipse.
Now you can run Jar file using below command,

```
hadoop jar <InputFilePath> <OutputFilePath>
```

Now the Process will start and if it completes without any errors then the Data is successfully loaded in the Hbase.

Once the steps 11 & 12 are completed the data is loaded successfully in the Hbase tables
now we want to analyze it using hive.

Step 13:(Please see hive.sql) and run both the query in Hive shell.

Step 14:(hive.sql)
We can confirm the data availability by firing the queries in step 14(hive.sql).

Step 15 (hive.sql)
we can find all the way's nodes which are passing Gymnasium.