Basic OOZIE HAND on Activity Two

Scenario : Create two tables . One table will be external and 2nd table will be internal (normal), then insert data into internal table from external table using oozie .

1. Create file by the name of external.hive or you can give any name To do so open notepad or any text editor like "sublime" then put the following scrip inside the external.hive

```
drop table orc_table;
drop table external_table;
Create external table external_table (
  id string,
  age string,
  gender string,
  occupation_id string,
)
row format delimited fields terminated by ','
stored as textfile location  "/user/maria_dev/abc";
```

2. Create file by the name of orc.hive or You can give any name To do so open notepad or any text editor like "sublime" then put the following scrip inside the orc.hive

```
Create Table orc_table(
  id string,
  age string,
  gender string,
  occupation_id string,
)
STORED AS ORC;
```

3. Create file by the name of Copydata.hql or you can give any name To do so open notepad or any text editor like "sublime" then put the following scrip inside the Copydata.hql
   Following command insert data into orc table from external table

```
insert into table orc_table
select
  id
  age
  gender
  occupation_id
from external_table;
```

4. Create Ozzie workflow.xml To do so open notepad or any text editor like "sublime" then put the following xml script in it and save it by the name of workflow.xml or you can give any name.

```
<workflow-app xmlns = "uri:oozie:workflow:0.2" name = "copy data from external to orc">
  <start to = "Create_External_Table" />

  <action name = "Create_External_Table">
    <hive xmlns = "uri:oozie:hive-action:0.4">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <script>${nameNode}/user/maria_dev/oozie_handsonactivity/external.hive</script>
    </hive>

    <ok to = "Create_orc_Table" />
    <error to = "kill_job" />
  </action>

  <action name = "Create_orc_Table">
    <hive xmlns = "uri:oozie:hive-action:0.4">
     <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>

      <script>${nameNode}/user/maria_dev/oozie_handsonactivity/orc.hive</script>
    </hive>

    <ok to = "Insert_into_Table" />
    <error to = "kill_job" />
  </action>

  <action name = "Insert_into_Table">
    <hive xmlns = "uri:oozie:hive-action:0.4">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>

      <script>${nameNode}/user/maria_dev/oozie_handsonactivity/Copydata.hql</script>

    </hive>

    <ok to = "end" />
    <error to = "kill_job" />
  </action>

  <kill name = "kill_job">
    <message>Job failed</message>
  </kill>

  <end name = "end" />
```

```
</workflow-app>
```

5. Upload all the files in HDFS using Ambari (File view) under the following path inside

```
user/maria_dev/oozie_handsonactivity
```

6. Create configuration file by the name of job.properties To do so open notepad or any text editor like "sublime" and put the following scrip in it and save it by the name of "job.properties"

```
nameNode=hdfs://sandbox.hortonworks.com:8020
jobTracker=http://sandbox.hortonworks.com:8050
queueName=default
oozie.use.system.libpath=true
oozie.wf.application.path=${nameNode}/user/maria_dev/oozie_handsonactivity/workflow.xml
```

7. Upload job.properties file inside VMware local directory e.g Documents/foldername/job.properties

Run Oozie Job

8. We need to install oozie connector for mysql if you have.

   Type: hadoop fs -put /usr/share/java/mysql-connector-java.jar /user/oozie/share/lib/lib-20171110144231/sqoop

   // NOTE "lib_20171110144231" this folder must be different for every computer. You can get this folder name from ambare For that Goto ambari --> Click on File view --> then goto user/oozie/share/lib

9. Press Oozie button-> click on "service action" button then Press Restart all button.
10. Following command is responsible for run oozie job .Open you terminal or command prompt then type the following command

```
oozie job  -oozie http://localhost:11000/oozie  -config  .Documents/foldername/job.properties  -run
```

11. Type your sandbox IP on the browser : IP:110000/oozie (e.g 192.168.567.0:11000)/oozie . so you can see the oozie job or workflow status.

Oozie coordinator

We can create schedule by coordinator

Oozie Coordinator models the workflow execution triggers in the form of time, data or event predicates. The workflow job mentioned inside the **Coordinator** is started only after the given conditions are satisfied.

Definitions of the below given code is as follows −

- **start** − It means the start datetime for the job. Starting at this time the actions will be materialized.
- **end** − The end datetime for the job. When actions will stop being materialized.
- **timezone** − The timezone of the coordinator application.
- **frequency** − The frequency, in minutes, to materialize actions.

Control Information

- **timeout** − The maximum time, in minutes, that a materialized action will be waiting for the additional conditions to be satisfied before being discarded. A timeout of 0 indicates that at the time of materialization all the other conditions must be satisfied, else the action will be discarded. A timeout of 0 indicates that if all the input events are not satisfied at the time of action materialization, the action should timeout immediately. A timeout of -1 indicates no timeout, the materialized action will wait forever for the other conditions to be satisfied. The default value is -1.
- **concurrency** − The maximum number of actions for this job that can be running at the same time. This value allows to materialize and submit multiple instances of the coordinator app, and allows operations to catchup on delayed processing. The default value is 1.
- **execution** − Specifies the execution order if multiple instances of the coordinator job have satisfied their execution criteria. Valid values are −
    - FIFO (oldest first) default.
    - LIFO (newest first).
    - LAST_ONLY (discards all older materializations).

12. Create coordinator file by the name coordinator.xml you can give any name then put the following scrip inside the coordinator.xml

```
<coordinator-app xmlns = "uri:oozie:coordinator:0.2" name =
  "hcoord_copydata_from_external_orc" frequency = "15" start =   "2018-04-16T01:00Z" end = "2018-04-
17T00:45Z" timezone = "America/New_York" >


  <controls>
    <timeout>1</timeout>
    <concurrency>1</concurrency>
    <execution>FIFO</execution>
    <throttle>1</throttle>
  </controls>


  <action>
    <workflow>
      <app-path>${nameNode}/user/maria_dev/oozie_handsonactivity /workflow.xml</app-path>
    </workflow>
  </action>


</coordinator-app>
```

13. We need to make minor changes in job.properties file. Open job.properties file then delete the following line because we already define the workflow path inside the coordinator.xml

```
oozie.wf.application.path=${nameNode}/user/maria_dev/oozie_handsonactivity/workflow.xml
```

14. We have to add coordinator.xml file path inside the job.properties. Add following line in job.properties

```
oozie.coord.application.path=${nameNode}/user/maria_dev/oozie_handsonactivity /coordinator.xml
```

15. Upload job.properties file again inside VMware local directory e.g Documents/foldername/job.properties

16. Run oozie job again .Open you terminal or command prompt then type the following command

```
oozie job   -oozie http://localhost:11000/oozie  -config  .Documents /job.properties  -run
```