**Explain in brief:**

**● The complete structure and the working of “Oozie Workflow scheduler “**

Workflow in Oozie is a sequence of actions arranged in a control dependency DAG (Direct Acyclic Graph). The actions are in controlled dependency as the next action can only run as per the output of the current action.A workflow action can be a Hive action, Pig action, Java action, Shell action, etc.

One advantage of the Oozie framework is that it is fully integrated with the Apache Hadoop stack and supports Apache MapReduce, Pig, Hive, and Sqoop. In addition, it can be used to schedule jobs specific to a system, such as Java programs.

The directed acyclic graph (DAG) graph contains two types of nodes:

**1.Control nodes**

 Control nodes, which are used to define job chronology, provide the rules for beginning and ending a workflow and control the workflow execution path with possible decision points known as fork and join nodes.

Control node achieves this by using

1. Start Control Node-It is the entry point for a workflow job, it indicates the first workflow node the workflow job must transition to.

When a workflow is started, it automatically transitions to the node specified in the start .

A workflow definition must have one start node.

**2. Kill Control Node**

The kill node allows a workflow job to kill itself. When a workflow job reaches the kill it finishes in error (KILLED).If one or more actions started by the workflow job are executing when the kill node is reached, the actions will be killed. A workflow definition may have zero or more kill nodes.

**3.End control Node**- The end node is the end for a workflow job, it indicates that the workflow job has completed successfully.

When a workflow job reaches the end it finishes successfully (SUCCEEDED).

If one or more actions started by the workflow job are executing when the end node is reached, the actions will be killed. In this scenario the workflow job is still considered as successfully run.

A workflow definition must have one end node.

**4.Decision Control Node-**A decision node enables a workflow to make a selection on the execution path to follow. The behavior of a decision node can be seen as a switch-case statement.A decision node consists of a list of predicates-transition pairs plus a default transition. Predicates are evaluated in order or appearance until one of them evaluates to true and the

**5. Fork and Join Control Nodes-**A fork node splits one path of execution into multiple concurrent paths of execution.A join node waits until every concurrent execution path of a previous fork node arrives to it.The fork and join nodes must be used in pairs. The join node assumes concurrent execution paths are children of the same fork node.

**2.Action nodes:**

An action node represents a workflow task, e.g., moving files into HDFS, running a MapReduce, Pig or[Hive](http://www.guru99.com/hive-tutorials.html)jobs, importing data using Sqoop or running a shell script of a program written in Java.