Hive job submitted with non-default name node (fs.default.name) doesn't process locations properly

# Summary

Create table/index/database DDL doesn't work properly if all following conditions are true:

* Metastore service is used
* fs.default.name is specified and it differs from the default one
* Location is not specified or specified as a not full URI

Specifying non-default fs.default.name makes a lot of sense in Azure environment, because it allows to submit the same job to different Hadoop clusters. See <https://issues.apache.org/jira/browse/OOZIE-1685> for more information.

# Issue in details

DDLTask class indirectly issues metastore service calls (if corresponded configuration present) passing locations as-is. If those locations are not fully qualified then they will be resolved in metastore service using default fs.default.name.

If fs.default.name passed to the hive client is not the default one, then it will lead to inconsistency: data locations will not be resolved as expected.

# Description of the fix

In the following methods of the DDL task locations will be resolved before passing further:

* createTableLike
* createTable
* addPartition
* createIndex

In order to preserve backward compatibility while fix this scenario, locations will be resolved only if fs.default.name is not the default one. It will be done as follows:

**public** **boolean** isDefaultNameNode() {

**return** !conf.getChangedProperties().containsKey(HiveConf.ConfVars.*HADOOPFS*.varname);

}

Locations will be resolved through the following new private method:

**private** String getQualifiedPath(Path path) **throws** HiveException {

FileSystem fs;

**try** {

fs = path.getFileSystem(conf);

} **catch** (IOException e) {

**throw** **new** HiveException(e);

}

**return** (**new** Path(fs.getUri().getScheme(), fs.getUri().getAuthority(), path

.toUri().getPath())).toString();

}