Item 106
git_comments:

git_commits:

1. **summary:** [SPARK-20195][SPARKR][SQL] add createTable catalog API and deprecate createExternalTable
message: [SPARK-20195][SPARKR][SQL] add createTable catalog API and deprecate createExternalTable ## What changes were proposed in this pull request? Following up on #17483, add createTable (which is new in 2.2.0) and deprecate createExternalTable, plus a number of minor fixes ## How was this patch tested? manual, unit tests Author: Felix Cheung <felixcheung_m@hotmail.com> Closes #17511 from felixcheung/rceatetable.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [SPARK-20159][SPARKR][SQL] Support all catalog API in R
body: ## What changes were proposed in this pull request? Add a set of catalog API in R ``` "currentDatabase", "listColumns", "listDatabases", "listFunctions", "listTables", "recoverPartitions", "refreshByPath", "refreshTable", "setCurrentDatabase", ``` https://github.com/apache/spark/pull/17483/files#diff-6929e6c5e59017ff954e110df20ed7ff ## How was this patch tested? manual tests, unit tests

github_pulls_comments:

1. **[Test build #75393 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75393/testReport)** for PR 17483 at commit [`e4808b6`](https://github.com/apache/spark/commit/e4808b656172e5d2994e0159ac4c0e326de1cb8a). * This patch **fails SparkR unit tests**. * This patch merges cleanly. * This patch adds no public classes.
2. cc @gatorsmile for any SQL specific inputs @felixcheung I will take a look at this later today. Meanwhile in the PR description can you note down what are the new functions being added in this change ?
3. updated PR description!
4. btw, @gatorsmile it looks like `listColumns` should throw `NoSuchTableException` and/or `NoSuchDatabaseException` instead of `AnalysisException` [here](https://github.com/apache/spark/blob/master/sql/core/src/main/scala/org/apache/spark/sql/internal/CatalogImpl.scala#L148)
5. **[Test build #75417 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75417/testReport)** for PR 17483 at commit [`5ab5834`](https://github.com/apache/spark/commit/5ab583443d60f6bf1d85608552962b46ac088633). * This patch **fails R style tests**. * This patch merges cleanly. * This patch adds no public classes.
6. **[Test build #75418 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75418/testReport)** for PR 17483 at commit [`28195b9`](https://github.com/apache/spark/commit/28195b98bf71c36b47e3e191b24715806f8aed6e). * This patch **fails SparkR unit tests**. * This patch merges cleanly. * This patch adds no public classes.
7. **[Test build #75420 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75420/testReport)** for PR 17483 at commit [`9c768ae`](https://github.com/apache/spark/commit/9c768ae983f8fbeed11b7c308ca7f5662f88d809). * This patch **fails SparkR unit tests**. * This patch merges cleanly. * This patch adds no public classes.
8. **[Test build #75422 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75422/testReport)** for PR 17483 at commit [`5093891`](https://github.com/apache/spark/commit/5093891e5a8fc0f299ebb4303ddb488e86f87221). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
9. **[Test build #75447 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75447/testReport)** for PR 17483 at commit [`3c66930`](https://github.com/apache/spark/commit/3c6693035b023d7c9c9e2caa3014247c130eb037). * This patch **fails SparkR unit tests**. * This patch merges cleanly. * This patch adds no public classes.
10. @gatorsmile I added a [line about recoverPartitions](https://github.com/apache/spark/pull/17483/files#diff-0921ddb2ea3cb3b1c329ae37ee829e9aR412), I think we should also be more clear in other language bindings? Also open https://issues.apache.org/jira/browse/SPARK-20188
11. **[Test build #75451 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75451/testReport)** for PR 17483 at commit [`2aea0cb`](https://github.com/apache/spark/commit/2aea0cb0f9b55acf741788aa72a573853560f3d9). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
12. **[Test build #75452 has finished](https://amplab.cs.berkeley.edu/jenkins/job/SparkPullRequestBuilder/75452/testReport)** for PR 17483 at commit [`aff13a8`](https://github.com/apache/spark/commit/aff13a860282375a650f6323987c73364ed439cd). * This patch passes all tests. * This patch merges cleanly. * This patch adds no public classes.
13. merged to master. thanks for the review!

github_pulls_reviews:

1. this is added to createExternalTable (won't work with json source otherwise) everything else is simply moved as-is from SQLContext.R
2. Just FYI, `createExternalTable ` is deprecated. See the PR: https://github.com/apache/spark/pull/16528 Let me make the corresponding changes in SQLContext too.
3. yes, I'm aware. I'm not sure if we need to make changes to SQLContext - schema is required for json source but it's ok in Scala to use createTable instead. It makes sense to add in R here though because - there is no createTable method (hmm, reviewing that PR you reference, maybe we should add it) - createTable just sounds too generic and too much like many existing R methods (in R, table is everywhere!), that I wasn't sure it's a good idea to add in R - createExternalTable since 2.0 is decoupled from SQLContext or SparkSession - it doesn't take either as parameter and it's calling on catalog
4. is `tables()` deprecated now ?
5. We dont have tests for `recoverPartitions` `refreshByPath` and `refreshTable` ?
6. I agree that `createTable` sounds very general, but I dont think its used by base R or any popular R package ?
7. right, there are some differences of the output (most notability catalog.listTables returns a `Dataset<Table>` - but I'm converting that into a DataFrame anyway), and I thought list* would be more consistent with other methods like `listColumn()`, `listDatabases()` ``` > head(tables("default")) database tableName isTemporary 1 default json FALSE Warning message: 'tables' is deprecated. Use 'listTables' instead. See help("Deprecated") > head(listTables("default")) name database description tableType isTemporary 1 json default <NA> EXTERNAL FALSE ``` If you think it makes sense, we could make `tables` an alias of `listTables` - it's going to call slightly different code on the Scala side and there are new columns and one different column name being returned.
8. `createExternalTable` is misleading now, because the table could be `managed` if users did not provide the value of `path`. Thus, we decided to rename it.
9. Yes. I knew it. See the JIRA: https://issues.apache.org/jira/browse/SPARK-19952. @hvanhovell plans to remove it in 2.3.0
10. right, I was just concerned that with `data.table`, `read.table` etc, table == data.frame in R as supposed to `hive table` or `managed table`, which could be fairly confusing. anyway, I think I'll follow up with a PR for `createTable` but as of now `path` is optional for `createExternalTable`, even though it's potentially misleading, it does work now.
11. changed.
12. ok, thanks
13. sharp eyes :) I was planning to add tests. I tested these manually, but the steps are more involved and these are only thin wrappers in R I think we should defer to scala tests.
14. > If you drop a managed table both data and meta data will be deleted if you drop an external table only metadata is deleted, external table is a way to protect data against accidental drop commands. Thus, it is a pretty important concept. It could be either Hive or Spark native one.

15. if I am not mistaken, the method `getTables()` is not used any more in R. Can we remove it from `r.SQLUtils`: https://github.com/apache/spark/blob/e8982ca7ad94e98d907babf2d6f1068b7cd064c6/sql/core/src/main/scala/org/apache/spark/sql/api/r/SQLUtils.scala#L219-L226 ? cc @HyukjinKwon
16. https://github.com/apache/spark/pull/30527
17. Yeah, I saw the PR. LGTM

jira_issues:

jira_issues_comments: