

Item 331

git_comments:

git_commits:

1. **summary:** SOLR-8819: Implement DatabaseMetaDataImpl getTables() and fix getSchemas(). (Trey Cahill, Joel Bernstein, Kevin Risdén)
message: SOLR-8819: Implement DatabaseMetaDataImpl getTables() and fix getSchemas(). (Trey Cahill, Joel Bernstein, Kevin Risdén)

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Implement DatabaseMetaDataImpl getTables() and fix getSchemas()
description: DbVisualizer NPE when clicking on DB References tab. After connecting, NPE if double click on "DB" under connection name then click on References tab.

jira_issues_comments:

1. Added a patch that fixes the NPE caused by clicking on the DB References tab by implementing the getTables() method in DatabaseMetaDataImpl. Removed restriction of requiring the field "TABLE_CAT" when querying _CATALOGS_ by wrapping CatalogsStream in SelectStream. Removed restriction of requiring the both the fields "TABLE_SCHEM" and "TABLE_CATALOG" when querying _SCHEMAS_ by wrapping SchemasStream in SelectStream. Added tests as needed.
2. Since DatabaseMetaDataImpl.getTables isn't fully implemented, it looks like it causes the following errors in DbVisualizer when clicking on the table looking icon under the table name. We should probably implement all of the return columns for DatabaseMetaDataImpl.getTables instead of just the two currently in the patch. {code} 2016-03-11 08:24:34.821 FINE 870 [pool-3-thread-2 - E.ǂǂ] RootConnection: DatabaseMetaDataImpl.getTables("localhost:9983", "test", "%", null) 2016-03-11 08:24:34.843 FINE 870 [ExecutorRunner-pool-2-thread-1 - E.ǂǂ] getting column 1 (java.lang.String) 'TABLE_SCHEM' using getString() 2016-03-11 08:24:34.843 FINE 870 [ExecutorRunner-pool-2-thread-1 - E.ǂǂ] getting column 2 (java.lang.String) 'TABLE_CATALOG' using getString() 2016-03-11 08:24:34.844 FINE 870 [ExecutorRunner-pool-2-thread-1 - Z.processResultSet] Fetched Rows: 1 Columns: 2 Exec: 0.022 Fetch: 0.000 sec 2016-03-11 08:24:34.847 WARN 870 [ExecutorRunner-pool-2-thread-1 - C.getValueAt] IndexOutOfBoundsException: row=0 column=2 rowCount=1 columnCount=2 java.lang.IndexOutOfBoundsException: Index: 2, Size: 2 at java.util.ArrayList.rangeCheck(ArrayList.java:653) at java.util.ArrayList.get(ArrayList.java:429) at com.onseven.dbvis.K.A.C.getValueAt(Z:2606) at com.onseven.dbvis.K.B.B.ǂ(Z:61) at com.onseven.dbvis.n.I.K.ǂ(Z:2413) at com.onseven.dbvis.n.I.K.ǂ(Z:240) at com.onseven.dbvis.n.I.K.ǂ(Z:2852) at com.onseven.dbvis.n.I.K.execute(Z:659) at com.onseven.dbvis.K.B.Z.ǂ(Z:2285) at com.onseven.dbvis.K.B.L.ǂ(Z:1374) at com.onseven.dbvis.K.B.L.doInBackground(Z:1521) at javax.swing.SwingWorker\$1.call(SwingWorker.java:295) at java.util.concurrent.FutureTask.run(FutureTask.java:266) at javax.swing.SwingWorker.run(SwingWorker.java:334) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) at java.lang.Thread.run(Thread.java:745) 2016-03-11 08:24:34.850 WARN 870 [ExecutorRunner-pool-2-thread-1 - C.getValueAt] IndexOutOfBoundsException: row=0 column=3 rowCount=1 columnCount=2

java.lang.IndexOutOfBoundsException: Index: 3, Size: 2 at
java.util.ArrayList.rangeCheck(ArrayList.java:653) at java.util.ArrayList.get(ArrayList.java:429) at
com.onseven.dbvis.K.A.C.getValueAt(Z:2606) at com.onseven.dbvis.K.B.B.†‡(Z:61) at
com.onseven.dbvis.n.I.K.†‡(Z:2413) at com.onseven.dbvis.n.I.K.†‡(Z:240) at com.onseven.dbvis.n.I.K.†‡(Z:2852) at com.onseven.dbvis.n.I.K.execute(Z:659) at com.onseven.dbvis.K.B.Z.†‡(Z:2285) at
com.onseven.dbvis.K.B.L.†‡(Z:1374) at com.onseven.dbvis.K.B.L.doInBackground(Z:1521) at
javax.swing.SwingWorker\$1.call(SwingWorker.java:295) at
java.util.concurrent.FutureTask.run(FutureTask.java:266) at
javax.swing.SwingWorker.run(SwingWorker.java:334) at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at
java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) at
java.lang.Thread.run(Thread.java:745) 2016-03-11 08:24:34.853 WARN 870 [ExecutorRunner-pool-2-thread-1 - C.getValueAt] IndexOutOfBoundsException: row=0 column=4 rowCount=1 columnCount=2
java.lang.IndexOutOfBoundsException: Index: 4, Size: 2 at
java.util.ArrayList.rangeCheck(ArrayList.java:653) at java.util.ArrayList.get(ArrayList.java:429) at
com.onseven.dbvis.K.A.C.getValueAt(Z:2606) at com.onseven.dbvis.K.B.B.†‡(Z:61) at
com.onseven.dbvis.n.I.K.†‡(Z:2413) at com.onseven.dbvis.n.I.K.†‡(Z:240) at com.onseven.dbvis.n.I.K.†‡(Z:2852) at com.onseven.dbvis.n.I.K.execute(Z:659) at com.onseven.dbvis.K.B.Z.†‡(Z:2285) at
com.onseven.dbvis.K.B.L.†‡(Z:1374) at com.onseven.dbvis.K.B.L.doInBackground(Z:1521) at
javax.swing.SwingWorker\$1.call(SwingWorker.java:295) at
java.util.concurrent.FutureTask.run(FutureTask.java:266) at
javax.swing.SwingWorker.run(SwingWorker.java:334) at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at
java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) at
java.lang.Thread.run(Thread.java:745) 2016-03-11 08:24:34.855 WARN 870 [ExecutorRunner-pool-2-thread-1 - C.getValueAt] IndexOutOfBoundsException: row=0 column=3 rowCount=1 columnCount=3
java.lang.IndexOutOfBoundsException: Index: 3, Size: 3 at
java.util.ArrayList.rangeCheck(ArrayList.java:653) at java.util.ArrayList.get(ArrayList.java:429) at
com.onseven.dbvis.K.A.C.getValueAt(Z:2606) at
com.jidesoft.grid.TableModelWrapperImpl.getValueAt(Unknown Source) at
com.jidesoft.grid.DefaultTableModelWrapper.getValueAt(Unknown Source) at com.onseven.dbvis.K.B.†‡(Z:292) at com.onseven.dbvis.n.I.K.†‡(Z:679) at com.onseven.dbvis.n.I.K.†‡(Z:240) at
com.onseven.dbvis.n.I.K.†‡(Z:2852) at com.onseven.dbvis.n.I.K.execute(Z:659) at
com.onseven.dbvis.K.B.Z.†‡(Z:2285) at com.onseven.dbvis.K.B.L.†‡(Z:1374) at
com.onseven.dbvis.K.B.L.doInBackground(Z:1521) at
javax.swing.SwingWorker\$1.call(SwingWorker.java:295) at
java.util.concurrent.FutureTask.run(FutureTask.java:266) at
javax.swing.SwingWorker.run(SwingWorker.java:334) at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at
java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) at
java.lang.Thread.run(Thread.java:745) 2016-03-11 08:24:34.856 WARN 870 [ExecutorRunner-pool-2-thread-1 - C.getValueAt] IndexOutOfBoundsException: row=0 column=4 rowCount=1 columnCount=3
java.lang.IndexOutOfBoundsException: Index: 4, Size: 3 at
java.util.ArrayList.rangeCheck(ArrayList.java:653) at java.util.ArrayList.get(ArrayList.java:429) at
com.onseven.dbvis.K.A.C.getValueAt(Z:2606) at
com.jidesoft.grid.TableModelWrapperImpl.getValueAt(Unknown Source) at
com.jidesoft.grid.DefaultTableModelWrapper.getValueAt(Unknown Source) at com.onseven.dbvis.K.B.†‡(Z:292) at com.onseven.dbvis.n.I.K.†‡(Z:679) at com.onseven.dbvis.n.I.K.†‡(Z:240) at
com.onseven.dbvis.n.I.K.†‡(Z:2852) at com.onseven.dbvis.n.I.K.execute(Z:659) at
com.onseven.dbvis.K.B.Z.†‡(Z:2285) at com.onseven.dbvis.K.B.L.†‡(Z:1374) at
com.onseven.dbvis.K.B.L.doInBackground(Z:1521) at
javax.swing.SwingWorker\$1.call(SwingWorker.java:295) at
java.util.concurrent.FutureTask.run(FutureTask.java:266) at
javax.swing.SwingWorker.run(SwingWorker.java:334) at

java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142) at
java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617) at
java.lang.Thread.run(Thread.java:745) {code}

3. Improves on [~cahilltr]'s patch. His patch and subsequent manual testing with DbVisualizer uncovered an issue with getSchemas. We shouldn't be returning any schema information since we don't have the concept of schemas. getTables is really what we want and this is now implemented in a TablesStream. This is a change to what is returned is returned by the JDBC driver. This should be put into Solr 6 before the release so we don't have to deal with backwards compatibility issues.
4. Fix usage of get/set Schema in ConnectionImpl.
5. Add test file that was missed on previous patch.
6. Added the ability for Collection Aliases to be considered their own databases and added tests as needed.
7. Was seeing NPE's at times when using DBVisualizer made calls to getTables(). Was able to trace the issue back to a null ZKStateReader in the cloudSolrClient. Calling cloudSolrClient.connect() prevents this from happening.
8. Just read the patch but haven't applied it yet. I think it looks pretty good I'll try to make some time to apply it today and test it out. We should probably hold off on collection alias support because the streaming API doesn't yet support aliases. So I don't believe SolrCloud aliases are queryable in the SQLHandler. One other very small thing is the TablesStream might make more sense as the TableStream, where each Tuple returns a single Table.
9. [~jbernste] I removed alias support and renamed TablesStreams to TableStream. Also updated the get on cloudSolrClient.getZKStateReader() to only get once instead of twice.
10. Ok things are looking pretty good to me.
11. Commit db6e6591741ab5aecf3a75326a9db126aa8cd0c4 in lucene-solr's branch refs/heads/master from [~risdenk] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=db6e659>] SOLR-8819: Implement DatabaseMetaDataImpl getTables() and fix getSchemas(). (Trey Cahill, Joel Bernstein, Kevin Riden)
12. Commit 39270d9df8959067a18edfe7c1e3cdb24adce543 in lucene-solr's branch refs/heads/branch_6x from [~risdenk] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=39270d9>] SOLR-8819: Implement DatabaseMetaDataImpl getTables() and fix getSchemas(). (Trey Cahill, Joel Bernstein, Kevin Riden)
13. Commit 658b2a4a925ed694a2c11add8c1bb81de1525294 in lucene-solr's branch refs/heads/branch_6_0 from [~risdenk] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=658b2a4>] SOLR-8819: Implement DatabaseMetaDataImpl getTables() and fix getSchemas(). (Trey Cahill, Joel Bernstein, Kevin Riden)