

**git\_comments:****git\_commits:**

1. **summary:** PHOENIX-1420 DatabaseMetaData.getSchemas returns incorrect column order  
**message:** PHOENIX-1420 DatabaseMetaData.getSchemas returns incorrect column order

**github\_issues:****github\_issues\_comments:****github\_pulls:****github\_pulls\_comments:****github\_pulls\_reviews:****jira\_issues:**

1. **summary:** DatabaseMetaData.getSchemas returns incorrect column order  
**description:** DatabaseMetaData.getSchemas returns incorrect column order. According to java 6 api doc, getSchemas should return answer set columns in the following order: 1.TABLE\_SCHEM String => schema name 2.TABLE\_CATALOG String => catalog name The Phoenix JDBC driver returns these in reverse column order, i.e. 1. TABLE\_CATALOG 2. TABLE\_SCHEM Our experience with other JDBC drivers is that they return the data w/ TABLE\_SCHEM being the first column.
2. **summary:** DatabaseMetaData.getSchemas returns incorrect column order  
**description:** DatabaseMetaData.getSchemas returns incorrect column order. According to java 6 api doc, getSchemas should return answer set columns in the following order: 1.TABLE\_SCHEM String => schema name 2.TABLE\_CATALOG String => catalog name The Phoenix JDBC driver returns these in reverse column order, i.e. 1. TABLE\_CATALOG 2. TABLE\_SCHEM Our experience with other JDBC drivers is that they return the data w/ TABLE\_SCHEM being the first column.  
**label:** documentation
3. **summary:** DatabaseMetaData.getSchemas returns incorrect column order  
**description:** DatabaseMetaData.getSchemas returns incorrect column order. According to java 6 api doc, getSchemas should return answer set columns in the following order: 1.TABLE\_SCHEM String => schema name 2.TABLE\_CATALOG String => catalog name The Phoenix JDBC driver returns these in reverse column order, i.e. 1. TABLE\_CATALOG 2. TABLE\_SCHEM Our experience with other JDBC drivers is that they return the data w/ TABLE\_SCHEM being the first column.
4. **summary:** DatabaseMetaData.getSchemas returns incorrect column order  
**description:** DatabaseMetaData.getSchemas returns incorrect column order. According to java 6 api doc, getSchemas should return answer set columns in the following order: 1.TABLE\_SCHEM String => schema name 2.TABLE\_CATALOG String => catalog name The Phoenix JDBC driver returns these in reverse column order, i.e. 1. TABLE\_CATALOG 2. TABLE\_SCHEM Our experience with other JDBC drivers is that they return the data w/ TABLE\_SCHEM being the first column.
5. **summary:** DatabaseMetaData.getSchemas returns incorrect column order  
**description:** DatabaseMetaData.getSchemas returns incorrect column order. According to java 6 api doc, getSchemas should return answer set columns in the following order: 1.TABLE\_SCHEM String => schema name 2.TABLE\_CATALOG String => catalog name The Phoenix JDBC driver returns these in reverse column order, i.e. 1. TABLE\_CATALOG 2. TABLE\_SCHEM Our experience with other JDBC drivers is that they return the data w/ TABLE\_SCHEM being the first column.
6. **summary:** DatabaseMetaData.getSchemas returns incorrect column order  
**description:** DatabaseMetaData.getSchemas returns incorrect column order. According to java 6 api doc, getSchemas should return answer set columns in the following order: 1.TABLE\_SCHEM String => schema name 2.TABLE\_CATALOG String => catalog name The Phoenix JDBC driver returns these in reverse column order, i.e. 1. TABLE\_CATALOG 2. TABLE\_SCHEM Our experience with other JDBC drivers is that they return the data w/ TABLE\_SCHEM being the first column.
7. **summary:** DatabaseMetaData.getSchemas returns incorrect column order  
**description:** DatabaseMetaData.getSchemas returns incorrect column order. According to java 6 api doc, getSchemas should return answer set columns in the following order: 1.TABLE\_SCHEM String => schema name 2.TABLE\_CATALOG String => catalog name The Phoenix JDBC driver returns these in reverse column order, i.e. 1. TABLE\_CATALOG 2. TABLE\_SCHEM Our experience with other JDBC drivers is that they return the data w/ TABLE\_SCHEM being the first column.

**jira\_issues\_comments:**

1. We based the column order on the javadoc:  
[http://docs.oracle.com/javase/7/docs/api/java/sql/DatabaseMetaData.html#getTables\(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String\)](http://docs.oracle.com/javase/7/docs/api/java/sql/DatabaseMetaData.html#getTables(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))
2. **body:** getSchema() answer set description in java doc does seem inconsistent with getTables , getColumnns, I grant you that. I would also have thought that catalog should go before schema. The description for getSchema seems to contradict this, however. As the java api doc is the 'standard', though, it should not be left to the reader to reinterpret the documentation. Are you aware of any other JDBC driver that sets the column-ordering this way?  
**label:** documentation
3. Strange - for getSchemas() it should be schema name first followed by table cat, but for getColumnns and getTables table cat is first. We can fix this in our next patch release. In the meantime, perhaps you can access via the name instead of positionally?
4. OK, no problem.
5. SUCCESS: Integrated in Phoenix-master #482 (See [https://builds.apache.org/job/Phoenix-master/482/]) PHOENIX-1420 DatabaseMetaData.getSchemas returns incorrect column order (jtaylor: rev ad525844fbbdb1fe1ca84938fc8eeff51ab49979) \* phoenix-core/src/it/java/org/apache/phoenix/end2end/QueryDatabaseMetaDataIT.java \* phoenix-core/src/main/java/org/apache/phoenix/jdbc/PhoenixDatabaseMetaData.java
6. SUCCESS: Integrated in Phoenix-3.0-hadoop1 #310 (See [https://builds.apache.org/job/Phoenix-3.0-hadoop1/310/]) PHOENIX-1420 DatabaseMetaData.getSchemas returns incorrect column order (jtaylor: rev b5012d994278982c287b18b9051616cd76ac8b80) \* phoenix-core/src/main/java/org/apache/phoenix/jdbc/PhoenixDatabaseMetaData.java \* phoenix-core/src/it/java/org/apache/phoenix/end2end/QueryDatabaseMetaDataIT.java
7. Bulk close of all issues that has been resolved in a released version.