

KYLIN-5426 Exporting the table structure API has regressed after the spark3 upgrade

RC

The problem here is mainly caused by the behavior changes of Spark 2.x and 3.x.

Here is an example to analyze the difference between Spark 2.x and 3.x in detail.

example

prepare data

Create a table with Hive - Hive Source Table

```
1 CREATE TABLE hjl_test2.hive_source_table1(  
2     age bigint,  
3     id bigint,  
4     name string,  
5     sex string  
6 ) stored as orc;
```

Creating Tables with Spark SQL - Spark Source Table

```
1 CREATE TABLE hjl_test2.spark_source_table1(  
2     age bigint,  
3     id bigint,  
4     name string,  
5     sex string  
6 ) USING orc;
```

Spark 2.4.3

For the `show create table {tableIdentity}` statement, there is only one

`ShowCreateTableCommand`

And you can identify the source of the table, whether it is a Hive table (`HiveSourceTable`) or a non-Hive table (`SparkDataSourceTable`).

- If it is `HiveSourceTable` , identify Hive's Serde and other information, and output `Hive DDL` format.
- If `SparkDataSourceTable` , convert to `Spark SQL DDL` format.

```
case class ShowCreateTableCommand(table: TableIdentifier) extends RunnableCommand {
  override val output: Seq[Attribute] = Seq(
    AttributeReference("createtab_stmt", StringType, nullable = false)()
  )

  override def run(sparkSession: SparkSession): Seq[Row] = {
    val catalog = sparkSession.sessionState.catalog
    val tableMetadata = catalog.getTableMetadata(table)

    // TODO: unify this after we unify the CREATE TABLE syntax for hive serde and data source table.
    val stmt = if (DDLUtils.isDataSourceTable(tableMetadata)) {
      showCreateDataSourceTable(tableMetadata)
    } else {
      showCreateHiveTable(tableMetadata)
    }

    Seq(Row(stmt))
  }
}
```

Using Spark SQL Reading Tables DDL

```
1 spark-sql> show create table hjl_test2.spark_source_table1;
2 CREATE TABLE `hjl_test2`.`spark_source_table1` (
3   `age` BIGINT,
4   `id` BIGINT,
5   `name` STRING,
6   `sex` STRING
7 ) USING orc OPTIONS (`serialization.format` '1');
8
9
10 spark-sql> show create table hjl_test2.hive_source_table1;
11 CREATE TABLE `hjl_test2`.`hive_source_table1` (
12   `age` BIGINT,
13   `id` BIGINT,
14   `name` STRING,
15   `sex` STRING
16 ) ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
17 WITH SERDEPROPERTIES ('serialization.format' = '1')
```

```

18 STORED AS
19     INPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
20     OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
21 TBLPROPERTIES (
22     'numFilesErasureCoded' = '0',
23     'transient_lastDdlTime' = '1670926751'
24 );

```

Spark 3.2.0

Added `show create table {tableIdentity} as serde` to return DDL of `HiveSourceTable`, corresponding to `ShowCreateTableAsSerdeCommand`

- If the table is a `SparkSourceTable`, an error will be reported
- If the table is a `HiveSourceTable`, identify Hive's Serde and other information, and output `Hive DDL` format.

```

case class ShowCreateTableAsSerdeCommand(
  table: TableIdentifier,
  override val output: Seq[Attribute])
  extends LeafRunnableCommand with ShowCreateTableCommandBase {

  Cheng Lian +3

  override def run(sparkSession: SparkSession): Seq[Row] = {
    val catalog = sparkSession.sessionState.catalog
    val tableMetadata = catalog.getTableRawMetadata(table)

    val stmt = if (DDLUtils.isDataSourceTable(tableMetadata)) {
      throw QueryCompilationErrors.showCreateTableAsSerdeNotAllowedOnSparkDataSourceTableError(
        table)
    } else {
      showCreateHiveTable(tableMetadata)
    }

    Seq(Row(stmt))
  }

```

The original `show create table {tableIdentity}` corresponds to `ShowCreateTableCommand`

- Both `HiveSourceTable` and `SparkSourceTable` return to `Spark SQL DDL` format.

```

case class ShowCreateTableCommand(
  table: TableIdentifier,
  override val output: Seq[Attribute])
  extends LeafRunnableCommand with ShowCreateTableCommandBase {

  Liang-Chi Hsieh +5

  override def run(sparkSession: SparkSession): Seq[Row] = {
    val catalog = sparkSession.sessionState.catalog
    if (catalog.isTempView(table)) {
      throw QueryCompilationErrors.showCreateTableNotSupportedOnTempView(table.identifier)
    } else {
      val tableMetadata = catalog.getTableRawMetadata(table)

      // TODO: [SPARK-28692] unify this after we unify the
      // CREATE TABLE syntax for hive serde and data source table.
      val metadata = if (DDLUtils.isDatasourceTable(tableMetadata)) {
        tableMetadata
      } else {
        // For a Hive serde table, we try to convert it to Spark DDL.
        if (tableMetadata.unsupportedFeatures.nonEmpty) {
          throw QueryCompilationErrors.showCreateTableFailToExecuteUnsupportedFeatureError(
            tableMetadata)
        }

        if ("true".equalsIgnoreCase(tableMetadata.properties.getOrElse("transactional", "false"))) {
          throw QueryCompilationErrors.showCreateTableNotSupportTransactionalHiveTableError(
            tableMetadata)
        }
      }
    }
  }
}

```

ShowCreateTableCommand

```

1 spark-sql> show create table hjl_test2.spark_source_table1;
2 CREATE TABLE `hjl_test2`.`spark_source_table1` (
3   `age` BIGINT,
4   `id` BIGINT,
5   `name` STRING,
6   `sex` STRING)
7 USING orc;
8
9
10 spark-sql> show create table hjl_test2.hive_source_table1;
11 CREATE TABLE `hjl_test2`.`hive_source_table1` (
12   `age` BIGINT,
13   `id` BIGINT,
14   `name` STRING,
15   `sex` STRING)

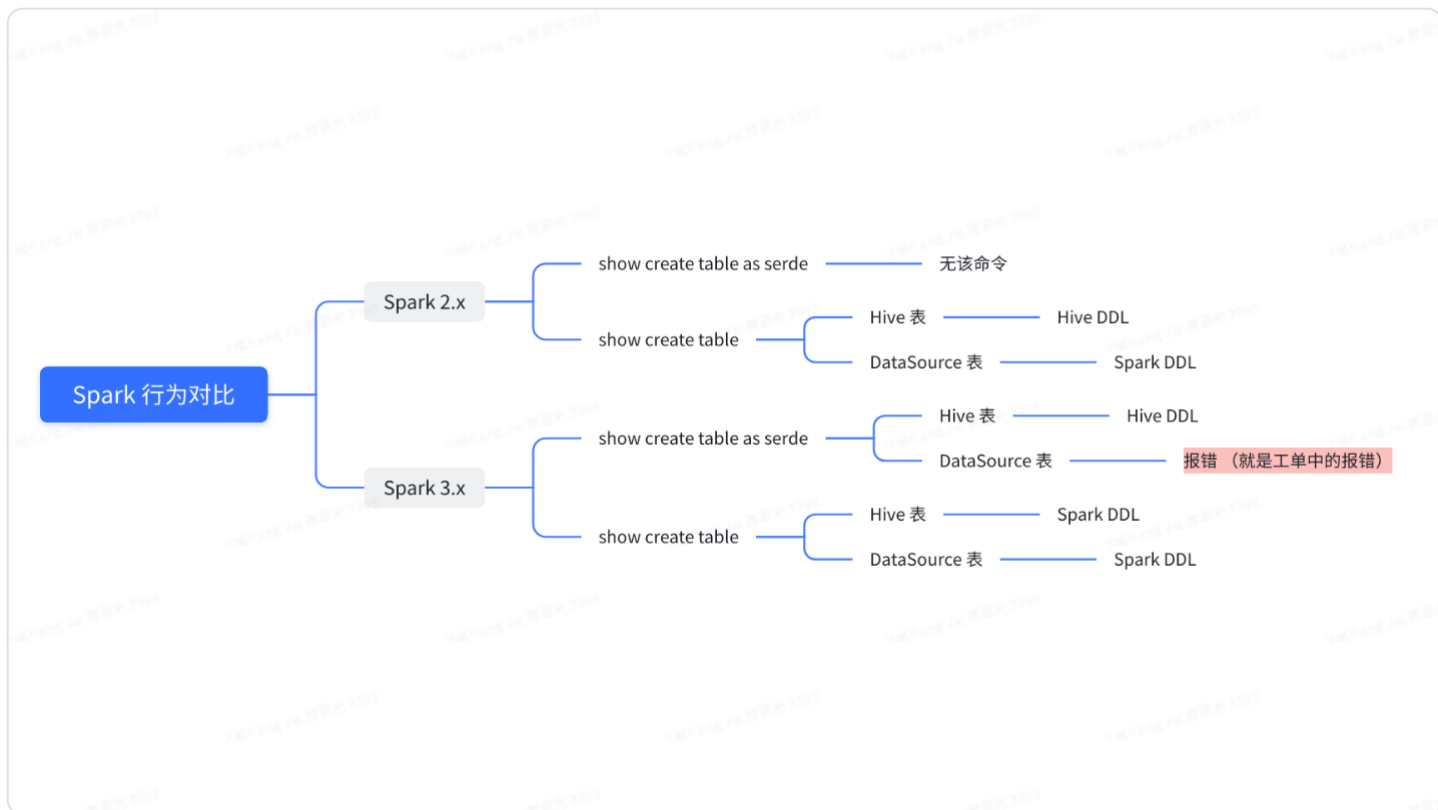
```

```
16 USING orc
17 TBLPROPERTIES (
18   'numFilesErasureCoded' = '0',
19   'transient_lastDdlTime' = '1670926751');
```

ShowCreateTableAsSerdeCommand

```
1 -- 报错
2 spark-sql> show create table hjl_test2.spark_source_table1 as serde;
3 Error in query: `hjl_test2`.`spark_source_table1` is a Spark data source table.
4
5
6 -- 正常, 并返回Hive DDL
7 spark-sql> show create table hjl_test2.hive_source_table1 as serde;
8 CREATE TABLE `hjl_test2`.`hive_source_table1` (
9   `age` BIGINT,
10  `id` BIGINT,
11  `name` STRING,
12  `sex` STRING)
13 ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
14 WITH SERDEPROPERTIES (
15   'serialization.format' = '1')
16 STORED AS
17  INPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
18  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
19 TBLPROPERTIES (
20   'numFilesErasureCoded' = '0',
21   'transient_lastDdlTime' = '1670926751');
```

conclusion



Fix Design

This problem is mainly caused by the Spark 2 liter 3.

Location code: `org.apache.spark.sql. DdlOperation #getTableDesc`

Use the `show create table as serde` command

Since the customer is using Spark DDL to build the table, as shown below

As RC above analysis conclusion, Spark 3.x uses `show create table as serde` to obtain Spark DataSource Table, an error will be reported.

```
CREATE EXTERNAL TABLE MONIKER_READ2 (
  id LONG COMMENT '',
  name STRING COMMENT '')
USING org.apache.spark.sql.execution.datasources.moniker.MonikerReader
OPTIONS (
  'CALC_ID'='4e30a07c-2de3-11ed-a261-0242ac120002',
  'CALC_NAME'='CALC_NAME',
  'CALC_UUID'='572c2a70-2de3-11ed-a261-0242ac120002',
  'DATASET_ID'='623fb206-2de3-11ed-a261-0242ac120002',
  'DATASET_UUID'='6a9a0c1c-2de3-11ed-a261-0242ac120002',
  'DATASET_NAME'='DATASET_NAME',
  'DATASET_CUT_KEYS'='LATEST'
)
LOCATION ' ';
```

```

109 def getTableDesc(database: String, table: String): String = {
110   val sql = s"show create table ${database}.${table} as serde"
111   var ddl = ""
112   val logicalPlan: LogicalPlan = SparderEnv.getSparkSession.sessionState.sqlParser.parsePlan(sql)
113   val queryExecution: QueryExecution = SparderEnv.getSparkSession.sessionState.executePlan(logicalPlan,
114     CommandExecutionMode.SKIP)
115   stripRootCommandResult(queryExecution.executedPlan) match {
116     case ExecutedCommandExec(show: ShowCreateTableAsSerdeCommand) =>
117       val catalog: SessionCatalog = SparderEnv.getSparkSession.sessionState.catalog
118       val metadata: CatalogTable = catalog.getTableMetadata(show.table)
119       metadata.tableType match {
120         case CatalogTableType.VIEW =>
121           val builder = new StringBuilder
122           builder += s"CREATE VIEW ${show.table.quotedString}"
123           if (metadata.schema.nonEmpty) {
124             builder += metadata.schema.map(_.toViewDDL).mkString("(", ", ", ", ")")
125           }
126           builder += metadata.viewText.mkString(" AS\n", " ", "\n")
127           ddl = builder.toString()
128         case CatalogTableType.MANAGED => ddl = ""
129         case CatalogTableType.EXTERNAL => ddl = SparderEnv.getSparkSession.sql(sql).takeAsList(1).get(0).getString(0)

```

The fix is simple

1. Get Meta information passed to Table in Spark Catalog
2. Determine whether the current table is HiveSourceTable or SparkDataSourceTable
 - a. HiveSourceTable -> show create table as serde
 - b. SparkDataSourceTable -> show create table

RC

这里的问题主要是Spark 2.x 与 3.x 的行为变更导致的。

下面举例子来详细分析一下Spark 2.x 与 3.x 的区别。

举例

准备数据

使用 Hive 创建表 - Hive Source Table

```
1 CREATE TABLE hjl_test2.hive_source_table1(
```



```
2    age bigint,  
3    id  bigint,  
4    name string,  
5    sex string  
6 ) stored as orc;
```

使用Spark SQL 创建表 - Spark Source Table

```
1 CREATE TABLE hjl_test2.spark_source_table1(  
2    age bigint,  
3    id  bigint,  
4    name string,  
5    sex string  
6 ) USING orc;
```

Spark 2.4.3

对于 `show create table {tableIdentity}` 语句，只有一个

`ShowCreateTableCommand`

并且可以识别Table的来源，是否是Hive表(`HiveSourceTable`)或者非Hive表(统一认为`SparkDataSourceTable`)。

- 如果是 `HiveSourceTable`，则识别Hive的Serde等信息，并输出 `Hive DDL` 格式。
- 如果是 `SparkDataSourceTable`，则转换成 `Spark SQL DDL` 格式。


```

case class ShowCreateTableCommand(table: TableIdentifier) extends RunnableCommand {
  override val output: Seq[Attribute] = Seq(
    AttributeReference("createtab_stmt", StringType, nullable = false)()
  )

  override def run(sparkSession: SparkSession): Seq[Row] = {
    val catalog = sparkSession.sessionState.catalog
    val tableMetadata = catalog.getTableMetadata(table)

    // TODO: unify this after we unify the CREATE TABLE syntax for hive serde and data source table.
    val stmt = if (DDLUtils.isDataSourceTable(tableMetadata)) {
      showCreateDataSourceTable(tableMetadata)
    } else {
      showCreateHiveTable(tableMetadata)
    }

    Seq(Row(stmt))
  }
}

```

使用Spark SQL 读取表DDL

```

1 spark-sql> show create table hjl_test2.spark_source_table1;
2 CREATE TABLE `hjl_test2`.`spark_source_table1` (
3   `age` BIGINT,
4   `id` BIGINT,
5   `name` STRING,
6   `sex` STRING
7 ) USING orc OPTIONS (`serialization.format` '1');
8
9
10 spark-sql> show create table hjl_test2.hive_source_table1;
11 CREATE TABLE `hjl_test2`.`hive_source_table1` (
12   `age` BIGINT,
13   `id` BIGINT,
14   `name` STRING,
15   `sex` STRING
16 ) ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
17 WITH SERDEPROPERTIES ('serialization.format' = '1')
18 STORED AS
19   INPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
20   OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
21 TBLPROPERTIES (
22   'numFilesErasureCoded' = '0',
23   'transient_lastDdlTime' = '1670926751'

```

Spark 3.2.0

新增了 `show create table {tableIdentity} as serde` 专门用于返回 `HiveSourceTable` 的 DDL，对应 `ShowCreateTableAsSerdeCommand`

- 如果 table 是 `SparkSourceTable`，则会报错
- 如果 table 是 `HiveSourceTable`，则识别Hive的Serde等信息，并输出 `Hive DDL` 格式。

```
case class ShowCreateTableAsSerdeCommand(
  table: TableIdentifier,
  override val output: Seq[Attribute])
  extends LeafRunnableCommand with ShowCreateTableCommandBase {

  Cheng Lian +3

  override def run(sparkSession: SparkSession): Seq[Row] = {
    val catalog = sparkSession.sessionState.catalog
    val tableMetadata = catalog.getTableRawMetadata(table)

    val stmt = if (DDLUtils.isDataSourceTable(tableMetadata)) {
      throw QueryCompilationErrors.showCreateTableAsSerdeNotAllowedOnSparkDataSourceTableError(
        table)
    } else {
      showCreateHiveTable(tableMetadata)
    }

    Seq(Row(stmt))
  }
}
```

原有的 `show create table {tableIdentity}` 对应 `ShowCreateTableCommand`

- 不论是 `HiveSourceTable` 还是 `SparkSourceTable`，均返回 `Spark SQL DDL` 格式。

```

case class ShowCreateTableCommand(
  table: TableIdentifier,
  override val output: Seq[Attribute])
  extends LeafRunnableCommand with ShowCreateTableCommandBase {

  // Liang-Chi Hsieh +5

  override def run(sparkSession: SparkSession): Seq[Row] = {
    val catalog = sparkSession.sessionState.catalog
    if (catalog.isTempView(table)) {
      throw QueryCompilationErrors.showCreateTableNotSupportedOnTempView(table.identifier)
    } else {
      val tableMetadata = catalog.getTableRawMetadata(table)

      // TODO: [SPARK-28692] unify this after we unify the
      // CREATE TABLE syntax for hive serde and data source table.
      val metadata = if (DDLUtils.isDatasourceTable(tableMetadata)) {
        tableMetadata
      } else {
        // For a Hive serde table, we try to convert it to Spark DDL.
        if (tableMetadata.unsupportedFeatures.nonEmpty) {
          throw QueryCompilationErrors.showCreateTableFailToExecuteUnsupportedFeatureError(
            tableMetadata)
        }

        if ("true".equalsIgnoreCase(tableMetadata.properties.getOrElse("transactional", "false"))) {
          throw QueryCompilationErrors.showCreateTableNotSupportTransactionalHiveTableError(
            tableMetadata)
        }
      }
    }
  }
}

```

ShowCreateTableCommand

```

1 spark-sql> show create table hjl_test2.spark_source_table1;
2 CREATE TABLE `hjl_test2`.`spark_source_table1` (
3   `age` BIGINT,
4   `id` BIGINT,
5   `name` STRING,
6   `sex` STRING)
7 USING orc;
8
9
10 spark-sql> show create table hjl_test2.hive_source_table1;
11 CREATE TABLE `hjl_test2`.`hive_source_table1` (
12   `age` BIGINT,
13   `id` BIGINT,

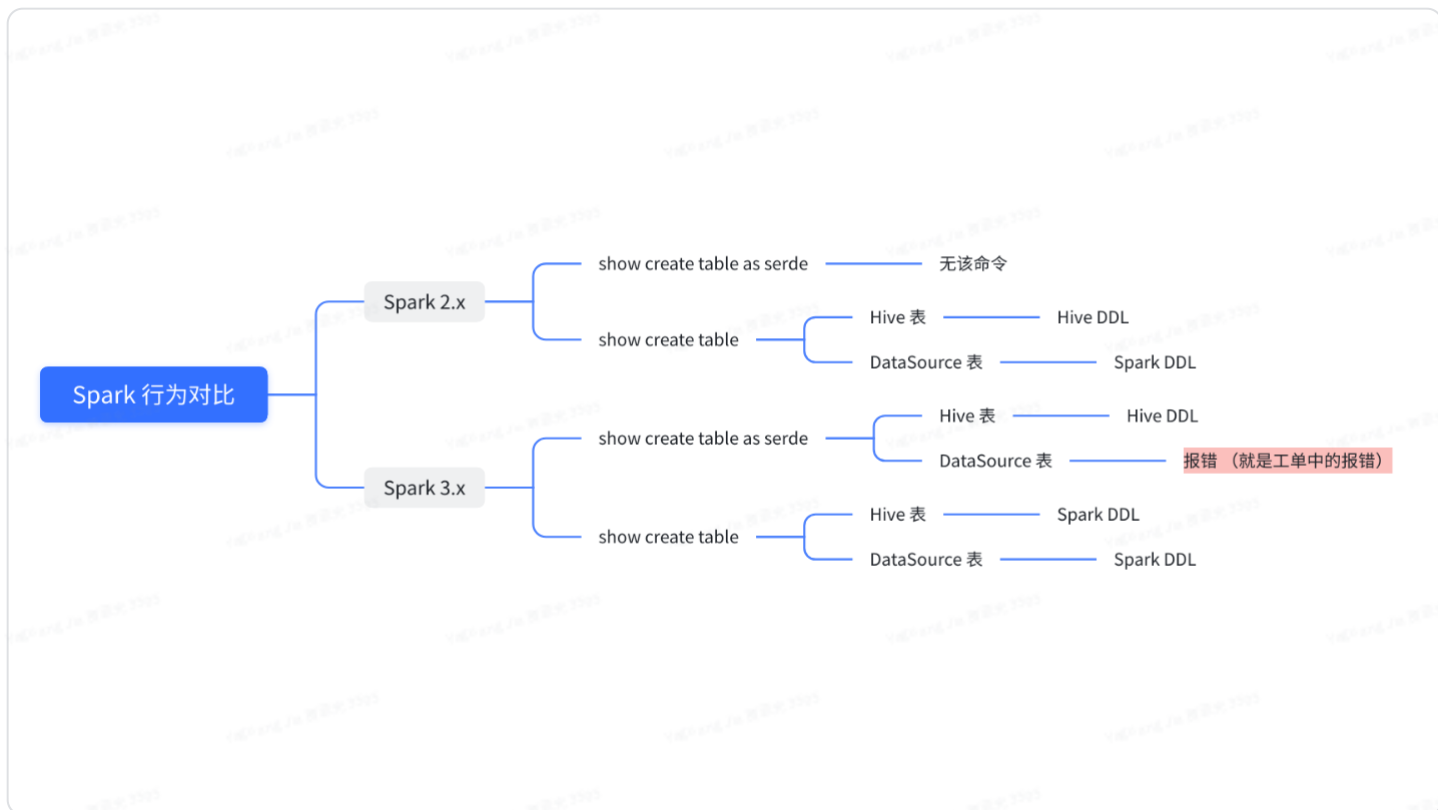
```

```
14 `name` STRING,  
15 `sex` STRING)  
16 USING orc  
17 TBLPROPERTIES (  
18 'numFilesErasureCoded' = '0',  
19 'transient_lastDdlTime' = '1670926751');
```

ShowCreateTableAsSerdeCommand

```
1 -- 报错  
2 spark-sql> show create table hjl_test2.spark_source_table1 as serde;  
3 Error in query: `hjl_test2`.`spark_source_table1` is a Spark data source table.  
4  
5  
6 -- 正常, 并返回Hive DDL  
7 spark-sql> show create table hjl_test2.hive_source_table1 as serde;  
8 CREATE TABLE `hjl_test2`.`hive_source_table1`(  
9   `age` BIGINT,  
10  `id` BIGINT,  
11  `name` STRING,  
12  `sex` STRING)  
13 ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'  
14 WITH SERDEPROPERTIES (  
15   'serialization.format' = '1')  
16 STORED AS  
17 INPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'  
18 OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'  
19 TBLPROPERTIES (  
20   'numFilesErasureCoded' = '0',  
21   'transient_lastDdlTime' = '1670926751');
```

结论



Fix Design

这个问题主要是Spark 2 升 3导致的。

定位代码：`org.apache.spark.sql.DdlOperation#getTableDesc`

代码中写死了使用 `show create table as serde` 命令

由于客户使用的是Spark DDL进行建表，如下图

如上面RC分析结论，Spark 3.x 使用 `show create table as serde` 获取Spark DataSource Table时，会报错。

```
CREATE EXTERNAL TABLE MONIKER_READ2(
  id LONG COMMENT '',
  name STRING COMMENT '')
USING org.apache.spark.sql.execution.datasources.moniker.MonikerReader
OPTIONS (
  'CALC_ID'='4e30a07c-2de3-11ed-a261-0242ac120002',
  'CALC_NAME'='CALC NAME',
  'CALC_UUID'='572c2a70-2de3-11ed-a261-0242ac120002',
  'DATASET_ID'='623fb206-2de3-11ed-a261-0242ac120002',
  'DATASET_UUID'='6a9a0c1c-2de3-11ed-a261-0242ac120002',
  'DATASET_NAME'='DATASET NAME',
  'DATASET_CUT_KEYS'='LATEST'
)
LOCATION ' ';
```

```

109 def getTableDesc(database: String, table: String): String = {
110     val sql = s"show create table ${database}.${table} as serde"
111     var ddl = ""
112     val logicalPlan: LogicalPlan = SparderEnv.getSparkSession.sessionState.sqlParser.parsePlan(sql)
113     val queryExecution: QueryExecution = SparderEnv.getSparkSession.sessionState.executePlan(logicalPlan,
114         CommandExecutionMode.SKIP)
115     stripRootCommandResult(queryExecution.executedPlan) match {
116         case ExecutedCommandExec(show: ShowCreateTableAsSerdeCommand) =>
117             val catalog: SessionCatalog = SparderEnv.getSparkSession.sessionState.catalog
118             val metadata: CatalogTable = catalog.getTableMetadata(show.table)
119             metadata.tableType match {
120                 case CatalogTableType.VIEW =>
121                     val builder = new StringBuilder
122                     builder += s"CREATE VIEW ${show.table.quotedString}"
123                     if (metadata.schema.nonEmpty) {
124                         builder += metadata.schema.map(_.toViewDDL).mkString("(", ", ", ", ")")
125                     }
126                     builder += metadata.viewText.mkString(" AS\n", "", "\n")
127                     ddl = builder.toString()
128                 case CatalogTableType.MANAGED => ddl = ""
129                 case CatalogTableType.EXTERNAL => ddl = SparderEnv.getSparkSession.sql(sql).takeAsList(1).get(0).getString(0)

```

修法很简单

1. 在Spark Catalog中获取传入Table的Meta信息
2. 判断当前表是HiveSourceTable还是SparkDataSourceTable
 - a. HiveSourceTable -> show create table as serde
 - b. SparkDataSourceTable -> show create table