# Carbondata使用过程中遇到的几个问题及解决办法

本文总结了几个本人在使用 Carbondata
的时候遇到的几个问题及其解决办法。这里使用的环境是：Spark 2.1.0、Carbondata 1.2.0。


## 必须指定 HDFS nameservices

在初始化 CarbonSession 的时候，如果不指定 HDFS
nameservices，在数据导入是没啥问题的；但是数据查询会出现相关数据找不到问题：

```
scala> val carbon = SparkSession.builder().tempnfig(sc.getConf).getOrCreateCarbonSession("h
dfs:///user/iteblog/carb")

scala> carbon.sql("""CREATE TABLE temp.iteblog(id bigint) STORED BY 'carbondata'""")
17/11/09 16:20:58 AUDIT command.CreateTable: [www.iteblog.com][iteblog][Thread-1]Creatin
g Table with Database name [temp] and Table name [iteblog]
17/11/09 16:20:58 WARN hive.HiveExternalCatalog: Couldn't find corresponding Hive SerDe for
 data source provider org.apache.spark.sql.CarbonSource. Persisting data source table `temp
`.`iteblog` into Hive metastore in Spark SQL specific format, which is NOT tempmpatible with
 Hive.
17/11/09 16:20:59 AUDIT command.CreateTable: [www.iteblog.com][iteblog][Thread-1]Table c
reated with Database name [temp] and Table name [iteblog]
res2: org.apache.spark.sql.DataFrame = []

scala> carbon.sql("insert overwrite table temp.iteblog select id from temp.mytable limit 10")
17/11/09 16:21:46 AUDIT rdd.CarbonDataRDDFactory$: [www.iteblog.com][iteblog][Thread-1]D
ata load request has been received for table temp.iteblog
17/11/09 16:21:46 WARN util.CarbonDataProcessorUtil: main sort scope is set to LOCAL_SORT
17/11/09 16:23:03 AUDIT rdd.CarbonDataRDDFactory$: [www.iteblog.com][iteblog][Thread-1]D
ata load is successful for temp.iteblog
res3: org.apache.spark.sql.DataFrame = []

scala> carbon.sql("select * from temp.iteblog limit 10").show(10,100)
```

17/11/09 16:23:15 WARN scheduler.TaskSetManager: Lost task 0.0 in stage 3.0 (TID 1011, static
.iteblog.com, executor 2): java.lang.RuntimeException: java.io.FileNotFoundException: /user/it
eblog/carb/temp/iteblog/Fact/Part0/Segment_0/part-0-0_batchno0-0-1510215706696.carbond
ata (No such file or directory)
 at org.apache.carbondata.tempre.indexstore.blockletindex.IndexWrapper.<init>(IndexWrapp
er.java:39)
 at org.apache.carbondata.tempre.scan.executor.impl.AbstractQueryExecutor.initQuery(Abstr
actQueryExecutor.java:141)
 at org.apache.carbondata.tempre.scan.executor.impl.AbstractQueryExecutor.getBlockExecuti
onInfos(AbstractQueryExecutor.java:216)
 at org.apache.carbondata.tempre.scan.executor.impl.VectorDetailQueryExecutor.execute(Vec
torDetailQueryExecutor.java:36)
 at org.apache.carbondata.spark.vectorreader.VectorizedCarbonRetemprdReader.initialize(Ve
ctorizedCarbonRetemprdReader.java:116)
 at org.apache.carbondata.spark.rdd.CarbonScanRDD.internalCompute(CarbonScanRDD.scala
:229)
 at org.apache.carbondata.spark.rdd.CarbonRDD.tempmpute(CarbonRDD.scala:62)
 at org.apache.spark.rdd.RDD.tempmputeOrReadCheckpoint(RDD.scala:323)
 at org.apache.spark.rdd.RDD.iterator(RDD.scala:287)
 at org.apache.spark.rdd.MapPartitionsRDD.tempmpute(MapPartitionsRDD.scala:38)
 at org.apache.spark.rdd.RDD.tempmputeOrReadCheckpoint(RDD.scala:323)
 at org.apache.spark.rdd.RDD.iterator(RDD.scala:287)
 at org.apache.spark.rdd.MapPartitionsRDD.tempmpute(MapPartitionsRDD.scala:38)
 at org.apache.spark.rdd.RDD.tempmputeOrReadCheckpoint(RDD.scala:323)
 at org.apache.spark.rdd.RDD.iterator(RDD.scala:287)
 at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:87)
 at org.apache.spark.scheduler.Task.run(Task.scala:99)
 at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:282)
 at java.util.tempncurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
 at java.util.tempncurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
 at java.lang.Thread.run(Thread.java:745)
Caused by: java.io.FileNotFoundException: /user/iteblog/carb/temp/iteblog/Fact/Part0/Segme
nt_0/part-0-0_batchno0-0-1510215706696.carbondata (No such file or directory)
 at java.io.FileInputStream.open(Native Method)
 at java.io.FileInputStream.<init>(FileInputStream.java:138)
 at java.io.FileInputStream.<init>(FileInputStream.java:93)
 at org.apache.carbondata.tempre.datastore.impl.FileFactory.getDataInputStream(FileFactory.
java:128)
 at org.apache.carbondata.tempre.reader.ThriftReader.open(ThriftReader.java:77)
 at org.apache.carbondata.tempre.reader.CarbonHeaderReader.readHeader(CarbonHeaderRe
ader.java:46)
 at org.apache.carbondata.tempre.util.DataFileFooterConverterV3.getSchema(DataFileFooterC
onverterV3.java:90)
 at org.apache.carbondata.tempre.util.CarbonUtil.readMetadatFile(CarbonUtil.java:925)
 at org.apache.carbondata.tempre.indexstore.blockletindex.IndexWrapper.<init>(IndexWrapp
er.java:37)

... 20 more

可以看出，如果创建 CarbonSession 的时候，如果不指定 HDFS nameservices，在数据导入是没啥问题的；查找的时候就会出现文件找不到。这个最直接的解决版本就是创建 CarbonSession 的时候指定 HDFS nameservices。针对这个问题一个改进措施是让 Carbondata 能够根据提供的 hadoop 配置信息自动补充 HDFS nameservices 信息。

## 不支持 tinyint 数据类型

```
scala> carbon.sql("""CREATE TABLE temp.iteblog(status tinyint) STORED BY 'carbondata'""")
org.apache.carbondata.spark.exception.MalformedCarbonCommandException: Unsupported data type: StructField(status,ByteType,true).getType
  at org.apache.spark.sql.parser.CarbonSpark2SqlParser$$anonfun$getFields$1.apply(CarbonSpark2SqlParser.scala:427)
  at org.apache.spark.sql.parser.CarbonSpark2SqlParser$$anonfun$getFields$1.apply(CarbonSpark2SqlParser.scala:417)
  at scala.collection.TraversableLike$$anonfun$map$1.apply(TraversableLike.scala:234)
  at scala.collection.TraversableLike$$anonfun$map$1.apply(TraversableLike.scala:234)
  at scala.collection.immutable.List.foreach(List.scala:381)
  at scala.collection.TraversableLike$class.map(TraversableLike.scala:234)
  at scala.collection.immutable.List.map(List.scala:285)
  at org.apache.spark.sql.parser.CarbonSpark2SqlParser.getFields(CarbonSpark2SqlParser.scala:417)
  at org.apache.spark.sql.parser.CarbonSqlAstBuilder.visitCreateTable(CarbonSparkSqlParser.scala:135)
  at org.apache.spark.sql.parser.CarbonSqlAstBuilder.visitCreateTable(CarbonSparkSqlParser.scala:72)
  at org.apache.spark.sql.catalyst.parser.SqlBaseParser$CreateTableContext.accept(SqlBaseParser.java:578)
  at org.antlr.v4.runtime.tree.AbstractParseTreeVisitor.visit(AbstractParseTreeVisitor.java:42)
  at org.apache.spark.sql.catalyst.parser.AstBuilder$$anonfun$visitSingleStatement$1.apply(AstBuilder.scala:66)
  at org.apache.spark.sql.catalyst.parser.AstBuilder$$anonfun$visitSingleStatement$1.apply(AstBuilder.scala:66)
  at org.apache.spark.sql.catalyst.parser.ParserUtils$.withOrigin(ParserUtils.scala:93)
  at org.apache.spark.sql.catalyst.parser.AstBuilder.visitSingleStatement(AstBuilder.scala:65)
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser$$anonfun$parsePlan$1.apply(ParseDriver.scala:54)
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser$$anonfun$parsePlan$1.apply(ParseDriver.scala:53)
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parse(ParseDriver.scala:82)
  at org.apache.spark.sql.parser.CarbonSparkSqlParser.parse(CarbonSparkSqlParser.scala:68)
```

```
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parsePlan(ParseDriver.scala:53)
  at org.apache.spark.sql.parser.CarbonSparkSqlParser.parsePlan(CarbonSparkSqlParser.scala:
49)
  at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:592)
  ... 50 elided
```

这是因为 Carbondata 目前不支持 tinyint 类型，Carbondata
目前支持的数据类型可以参见：http://carbondata.apache.org/supported-data-types-in-carbondata.html。但是奇怪的是 CARBONDATA-18
这里面已经解决了这个问题，不知道为啥到当前版本却不支持了。

## 添加分区出现NoSuchTableException

如果你使用 ALTER TABLE temp.iteblog ADD PARTITION('2017')
语句来添加分区，你会遇到下面的异常：

```
scala> carbon.sql("ALTER TABLE temp.iteblog ADD PARTITION('2012')")
org.apache.spark.sql.catalyst.analysis.NoSuchTableException: Table or view 'iteblog' not found
 in database 'default';
  at org.apache.spark.sql.hive.client.HiveClient$$anonfun$getTable$1.apply(HiveClient.scala:76
)
  at org.apache.spark.sql.hive.client.HiveClient$$anonfun$getTable$1.apply(HiveClient.scala:76
)
  at scala.Option.getOrElse(Option.scala:121)
  at org.apache.spark.sql.hive.client.HiveClient$class.getTable(HiveClient.scala:76)
  at org.apache.spark.sql.hive.client.HiveClientImpl.getTable(HiveClientImpl.scala:78)
  at org.apache.spark.sql.hive.HiveExternalCatalog$$anonfun$org$apache$spark$sql$hive$Hiv
eExternalCatalog$$getRawTable$1.apply(HiveExternalCatalog.scala:110)
  at org.apache.spark.sql.hive.HiveExternalCatalog$$anonfun$org$apache$spark$sql$hive$Hiv
eExternalCatalog$$getRawTable$1.apply(HiveExternalCatalog.scala:110)
  at org.apache.spark.sql.hive.HiveExternalCatalog.withClient(HiveExternalCatalog.scala:95)
  at org.apache.spark.sql.hive.HiveExternalCatalog.org$apache$spark$sql$hive$HiveExternalC
atalog$$getRawTable(HiveExternalCatalog.scala:109)
  at org.apache.spark.sql.hive.HiveExternalCatalog$$anonfun$getTable$1.apply(HiveExternalC
atalog.scala:601)
  at org.apache.spark.sql.hive.HiveExternalCatalog$$anonfun$getTable$1.apply(HiveExternalC
atalog.scala:601)
  at org.apache.spark.sql.hive.HiveExternalCatalog.withClient(HiveExternalCatalog.scala:95)
  at org.apache.spark.sql.hive.HiveExternalCatalog.getTable(HiveExternalCatalog.scala:600)
  at org.apache.spark.sql.hive.HiveMetastoreCatalog.lookupRelation(HiveMetastoreCatalog.scal
a:106)
  at org.apache.spark.sql.hive.HiveSessionCatalog.lookupRelation(HiveSessionCatalog.scala:69)
```

```
  at org.apache.spark.sql.hive.CarbonSessionCatalog.lookupRelation(CarbonSessionState.scala:83)
  at org.apache.spark.sql.internal.CatalogImpl.refreshTable(CatalogImpl.scala:461)
  at org.apache.spark.sql.execution.command.AlterTableSplitPartitionCommand.processSchema(carbonTableSchema.scala:283)
  at org.apache.spark.sql.execution.command.AlterTableSplitPartitionCommand.run(carbonTableSchema.scala:229)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult$lzycompute(commands.scala:58)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(commands.scala:56)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.doExecute(commands.scala:74)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:114)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:114)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$executeQuery$1.apply(SparkPlan.scala:135)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
  at org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.scala:132)
  at org.apache.spark.sql.execution.SparkPlan.execute(SparkPlan.scala:113)
  at org.apache.spark.sql.execution.QueryExecution.toRdd$lzycompute(QueryExecution.scala:87)
  at org.apache.spark.sql.execution.QueryExecution.toRdd(QueryExecution.scala:87)
  at org.apache.spark.sql.Dataset.<init>(Dataset.scala:185)
  at org.apache.spark.sql.Dataset$.ofRows(Dataset.scala:64)
  at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:592)
  ... 50 elided
```

运行上面的SQL语句，Carbondata 表相关的分区其实已经添加好了，但是通过 Spark 刷新表的相关信息就出错了。从出错的信息可以看出，虽然我们已经传递了表所在的 DB 相关信息，但是 Spark 的 catalyst 并没有获取到，这个 bug 是因为代码里面并没有将表数据相关信息传递给 catalyst， 这个 bug 还影响分区的 split 相关操作。不过此 bug 在 CARBONDATA-1593 里面已经解决。

## insert overwrite 操作超过三次将会出现 NPE

如果你在导数的时候执行 insert overwrite 大于等于三次，那么恭喜你，你肯定会遇到下面的异常，如下：

```
scala> carbon.sql("insert overwrite table temp.iteblog select id from  co.order_common_p whe
```

```
re dt = '2012-10'")
17/10/26 13:00:05 AUDIT rdd.CarbonDataRDDFactory$: [www.iteblog.com][iteblog][Thread-1]Data load request has been received for table temp.iteblog
17/10/26 13:00:05 WARN util.CarbonDataProcessorUtil: main sort scope is set to LOCAL_SORT
17/10/26 13:00:08 ERROR filesystem.AbstractDFSCarbonFile: main Exception occurred:File does not exist: hdfs://mycluster/user/iteblog/carb/temp/iteblog/Fact/Part0/Segment_0
17/10/26 13:00:09 ERROR command.LoadTable: main
java.lang.NullPointerException
  at org.apache.carbondata.core.datastore.filesystem.AbstractDFSCarbonFile.isDirectory(AbstractDFSCarbonFile.java:88)
  at org.apache.carbondata.core.util.CarbonUtil.deleteRecursive(CarbonUtil.java:364)
  at org.apache.carbondata.core.util.CarbonUtil.access$100(CarbonUtil.java:93)
  at org.apache.carbondata.core.util.CarbonUtil$2.run(CarbonUtil.java:326)
  at org.apache.carbondata.core.util.CarbonUtil$2.run(CarbonUtil.java:322)
  at java.security.AccessController.doPrivileged(Native Method)
  at javax.security.auth.Subject.doAs(Subject.java:422)
  at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1491)
  at org.apache.carbondata.core.util.CarbonUtil.deleteFoldersAndFiles(CarbonUtil.java:322)
  at org.apache.carbondata.spark.load.CarbonLoaderUtil.recordLoadMetadata(CarbonLoaderUtil.java:333)
  at org.apache.carbondata.spark.rdd.CarbonDataRDDFactory$.updateStatus$1(CarbonDataRDDFactory.scala:595)
  at org.apache.carbondata.spark.rdd.CarbonDataRDDFactory$.loadCarbonData(CarbonDataRDDFactory.scala:1107)
  at org.apache.spark.sql.execution.command.LoadTable.processData(carbonTableSchema.scala:1046)
  at org.apache.spark.sql.execution.command.LoadTable.run(carbonTableSchema.scala:754)
  at org.apache.spark.sql.execution.command.LoadTableByInsert.processData(carbonTableSchema.scala:651)
  at org.apache.spark.sql.execution.command.LoadTableByInsert.run(carbonTableSchema.scala:637)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult$lzycompute(commands.scala:58)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(commands.scala:56)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.doExecute(commands.scala:74)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:114)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:114)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$executeQuery$1.apply(SparkPlan.scala:135)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
  at org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.scala:132)
```

```
  at org.apache.spark.sql.execution.SparkPlan.execute(SparkPlan.scala:113)
  at org.apache.spark.sql.execution.QueryExecution.toRdd$lzycompute(QueryExecution.scala:8
7)
  at org.apache.spark.sql.execution.QueryExecution.toRdd(QueryExecution.scala:87)
  at org.apache.spark.sql.Dataset.<init>(Dataset.scala:185)
  at org.apache.spark.sql.Dataset$.ofRows(Dataset.scala:64)
  at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:592)
  at $line20.$read$$iw$$iw$$iw$$iw$$iw$$iw$$iw$$iw$$iw$$iw.<init>(<console>:31)
  at $line20.$read$$iw$$iw$$iw$$iw$$iw$$iw$$iw$$iw$$iw.<init>(<console>:36)
  at $line20.$read$$iw$$iw$$iw$$iw$$iw$$iw$$iw$$iw.<init>(<console>:38)
  at $line20.$read$$iw$$iw$$iw$$iw$$iw$$iw$$iw.<init>(<console>:40)
  at $line20.$read$$iw$$iw$$iw$$iw$$iw$$iw.<init>(<console>:42)
  at $line20.$read$$iw$$iw$$iw$$iw$$iw.<init>(<console>:44)
  at $line20.$read$$iw$$iw$$iw$$iw.<init>(<console>:46)
  at $line20.$read$$iw$$iw$$iw.<init>(<console>:48)
  at $line20.$read$$iw$$iw.<init>(<console>:50)
  at $line20.$read$$iw.<init>(<console>:52)
  at $line20.$read.<init>(<console>:54)
  at $line20.$read$.<init>(<console>:58)
  at $line20.$read$.<clinit>(<console>)
  at $line20.$eval$.$print$lzycompute(<console>:7)
  at $line20.$eval$.$print(<console>:6)
  at $line20.$eval.$print(<console>)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
)
  at java.lang.reflect.Method.invoke(Method.java:498)
  at scala.tools.nsc.interpreter.IMain$ReadEvalPrint.call(IMain.scala:786)
  at scala.tools.nsc.interpreter.IMain$Request.loadAndRun(IMain.scala:1047)
  at scala.tools.nsc.interpreter.IMain$WrappedRequest$$anonfun$loadAndRunReq$1.apply(IM
ain.scala:638)
  at scala.tools.nsc.interpreter.IMain$WrappedRequest$$anonfun$loadAndRunReq$1.apply(IM
ain.scala:637)
  at scala.reflect.internal.util.ScalaClassLoader$class.asContext(ScalaClassLoader.scala:31)
  at scala.reflect.internal.util.AbstractFileClassLoader.asContext(AbstractFileClassLoader.scala:1
9)
  at scala.tools.nsc.interpreter.IMain$WrappedRequest.loadAndRunReq(IMain.scala:637)
  at scala.tools.nsc.interpreter.IMain.interpret(IMain.scala:569)
  at scala.tools.nsc.interpreter.IMain.interpret(IMain.scala:565)
  at scala.tools.nsc.interpreter.ILoop.interpretStartingWith(ILoop.scala:807)
  at scala.tools.nsc.interpreter.ILoop.command(ILoop.scala:681)
  at scala.tools.nsc.interpreter.ILoop.processLine(ILoop.scala:395)
  at scala.tools.nsc.interpreter.ILoop.loop(ILoop.scala:415)
  at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply$mcZ$sp(ILoop.scala:923)
  at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply(ILoop.scala:909)
```

```
  at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply(ILoop.scala:909)
  at scala.reflect.internal.util.ScalaClassLoader$.savingContextLoader(ScalaClassLoader.scala:9
7)
  at scala.tools.nsc.interpreter.ILoop.process(ILoop.scala:909)
  at org.apache.spark.repl.Main$.doMain(Main.scala:68)
  at org.apache.spark.repl.Main$.main(Main.scala:51)
  at org.apache.spark.repl.Main.main(Main.scala)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
)
  at java.lang.reflect.Method.invoke(Method.java:498)
  at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMai
n(SparkSubmit.scala:738)
  at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:187)
  at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:212)
  at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:126)
  at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
17/10/26 13:00:09 AUDIT command.LoadTable: [www.iteblog.com][iteblog][Thread-1]Dataload
 failure for temp.iteblog. Please check the logs
java.lang.NullPointerException
  at org.apache.carbondata.core.datastore.filesystem.AbstractDFSCarbonFile.isDirectory(Abstr
actDFSCarbonFile.java:88)
  at org.apache.carbondata.core.util.CarbonUtil.deleteRecursive(CarbonUtil.java:364)
  at org.apache.carbondata.core.util.CarbonUtil.access$100(CarbonUtil.java:93)
  at org.apache.carbondata.core.util.CarbonUtil$2.run(CarbonUtil.java:326)
  at org.apache.carbondata.core.util.CarbonUtil$2.run(CarbonUtil.java:322)
  at java.security.AccessController.doPrivileged(Native Method)
  at javax.security.auth.Subject.doAs(Subject.java:422)
  at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1491
)
  at org.apache.carbondata.core.util.CarbonUtil.deleteFoldersAndFiles(CarbonUtil.java:322)
  at org.apache.carbondata.spark.load.CarbonLoaderUtil.recordLoadMetadata(CarbonLoaderU
til.java:333)
  at org.apache.carbondata.spark.rdd.CarbonDataRDDFactory$.updateStatus$1(CarbonDataRD
DFactory.scala:595)
  at org.apache.carbondata.spark.rdd.CarbonDataRDDFactory$.loadCarbonData(CarbonDataR
DDFactory.scala:1107)
  at org.apache.spark.sql.execution.command.LoadTable.processData(carbonTableSchema.sca
la:1046)
  at org.apache.spark.sql.execution.command.LoadTable.run(carbonTableSchema.scala:754)
  at org.apache.spark.sql.execution.command.LoadTableByInsert.processData(carbonTableSch
ema.scala:651)
  at org.apache.spark.sql.execution.command.LoadTableByInsert.run(carbonTableSchema.scal
a:637)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult$lzyco
```

```
mpute(commands.scala:58)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(comm
ands.scala:56)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.doExecute(commands.s
cala:74)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:114
)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:114
)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$executeQuery$1.apply(SparkPlan.sca
la:135)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
  at org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.scala:132)
  at org.apache.spark.sql.execution.SparkPlan.execute(SparkPlan.scala:113)
  at org.apache.spark.sql.execution.QueryExecution.toRdd$lzycompute(QueryExecution.scala:8
7)
  at org.apache.spark.sql.execution.QueryExecution.toRdd(QueryExecution.scala:87)
  at org.apache.spark.sql.Dataset.<init>(Dataset.scala:185)
  at org.apache.spark.sql.Dataset$.ofRows(Dataset.scala:64)
  at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:592)
  ... 50 elided

scala>
```

虽然出现 NPE 异常，但是数据其实已经导到 Carbondata
相关表里面了。引起这个异常的原因其实是因为每次执行完 insert overwrite
操作的时候，都需要删除之前的数据（也就是Segment目录）。但是 Segment
目录存在重复删除，导致找不到相关目录所以出现了 NPE 异常。这个问题在 CARBONDATA-1486
解决了。

## 不支持超过32767个字符的列

如果你有一列数据长度大于32767（Short.MaxValue），并且 enable.unsafe.sort=true
，那么你往 Carbondata 表导数据的时候会出现异常，如下：

```
java.lang.NegativeArraySizeException
  at org.apache.carbondata.processing.newflow.sort.unsafe.UnsafeCarbonRowPage.getRow(Un
safeCarbonRowPage.java:182)
  at org.apache.carbondata.processing.newflow.sort.unsafe.holder.UnsafeInmemoryHolder.re
adRow(UnsafeInmemoryHolder.java:63)
  at org.apache.carbondata.processing.newflow.sort.unsafe.merger.UnsafeSingleThreadFinalS
ortFilesMerger.startSorting(UnsafeSingleThreadFinalSortFilesMerger.java:114)
```

```
  at org.apache.carbondata.processing.newflow.sort.unsafe.merger.UnsafeSingleThreadFinalS
ortFilesMerger.startFinalMerge(UnsafeSingleThreadFinalSortFilesMerger.java:81)
  at org.apache.carbondata.processing.newflow.sort.impl.UnsafeParallelReadMergeSorterImpl.
sort(UnsafeParallelReadMergeSorterImpl.java:105)
  at org.apache.carbondata.processing.newflow.steps.SortProcessorStepImpl.execute(SortProc
essorStepImpl.java:62)
  at org.apache.carbondata.processing.newflow.steps.DataWriterProcessorStepImpl.execute(D
ataWriterProcessorStepImpl.java:87)
  at org.apache.carbondata.processing.newflow.DataLoadExecutor.execute(DataLoadExecutor.j
ava:51)
  at org.apache.carbondata.spark.rdd.NewDataFrameLoaderRDD$$anon$2.<init>(NewCarbon
DataLoadRDD.scala:442)
  at org.apache.carbondata.spark.rdd.NewDataFrameLoaderRDD.internalCompute(NewCarbon
DataLoadRDD.scala:405)
  at org.apache.carbondata.spark.rdd.CarbonRDD.compute(CarbonRDD.scala:62)
  at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:323)
  at org.apache.spark.rdd.RDD.iterator(RDD.scala:287)
```

这是 Carbondata 设计的缺陷，目前没有办法解决这个问题，不过可以实现一个类似于 varchar(size) 的数据类型。

## 日期格式错误导致数据丢失

如果你将带有日期类型的数据导入到 Carbondata 表中，可能会出现数据丢失：

```
scala> carbon.sql("""CREATE TABLE temp.iteblog(dt DATE) STORED BY 'carbondata'""")
17/11/09 16:44:46 AUDIT temp.mand.CreateTable: [www.iteblog.com][iteblog][Thread-1]Creati
ng Table with Database name [temp] and Table name [iteblog]
17/11/09 16:44:47 WARN hive.HiveExternalCatalog: Couldn't find temp.responding Hive SerDe
for data source provider org.apache.spark.sql.CarbonSource. Persisting data source table `te
mp`.`iteblog` into Hive metastore in Spark SQL specific format, which is NOT temp.patible wi
th Hive.
17/11/09 16:44:47 AUDIT temp.mand.CreateTable: [www.iteblog.com][iteblog][Thread-1]Table
created with Database name [temp] and Table name [iteblog]
res1: org.apache.spark.sql.DataFrame = []

scala> carbon.sql("select dt from temp.mydate").show(10,100)
17/11/09 16:44:52 ERROR lzo.LzoCodec: Failed to load/initialize native-lzo library
+--------+
|      dt|
+--------+
|20170509|
```

```
|20170511|
|20170507|
|20170504|
|20170502|
|20170506|
|20170501|
|20170508|
|20170510|
|20170505|
+--------+
only showing top 10 rows


scala> carbon.sql("insert into table temp.iteblog select dt from temp.mydate limit 10")
17/11/09 16:45:14 AUDIT rdd.CarbonDataRDDFactory$: [www.iteblog.com][iteblog][Thread-1]Data load request has been received for table temp.iteblog
17/11/09 16:45:14 WARN util.CarbonDataProcessorUtil: main sort scope is set to LOCAL_SORT
17/11/09 16:45:16 AUDIT rdd.CarbonDataRDDFactory$: [www.iteblog.com][iteblog][Thread-1]Data load is successful for temp.iteblog
res3: org.apache.spark.sql.DataFrame = []

scala> carbon.sql("select * from temp.iteblog limit 10").show(10,100)
+----+
|  dt|
+----+
|null|
|null|
|null|
|null|
|null|
|null|
|null|
|null|
|null|
|null|
+----+
```

这是因为 Carbondata 对数据类型(DATE)有默认的格式，由参数 carbon.date.format 控制，默认值是 yyyy-MM-dd。所以你使用 yyyy-MM-dd 格式去解析 20170505 数据肯定会出现错误，从而导致数据丢失了。同理，时间戳类型(TIMESTAMP) 也有默认的格式，由参数 carbon.timestamp.format 空值，默认值为 yyyy-MM-dd HH:mm:ss。