Links

- https://spark.apache.org/docs/3.5.1/sql-data-sources.html

Required Downloads

Download Training Materials

- Put the directory into a an accessible folder for later

```
git clone https://github.com/databricks/LearningSparkV2.git
```

Download latest Jar for Avro

- https://repo1.maven.org/maven2/org/apache/spark/spark-avro_2.12/3.5.4/spark-avro_2.12-3.5.4.jar
- Put the directory into a an accessible folder for later

Installing Option 1: Apache Spark and Running Test Code

Download and Install Spark (Source)

1. Visit the Apache Spark download page.
2. Select the pre-built Spark package for Hadoop (e.g., "Pre-built for Apache Hadoop 2.7").
3. Download the compressed file and extract it:

```
tar -xf spark-<version>-bin-hadoop2.7.tgz
cd spark-<version>-bin-hadoop2.7
```

Spark-shell via binary

```
cd \folder\to\spark\spark-3.5.4-bin-hadoop3\bin\
```

```
.\spark-shell
```

- http://r748:4040/jobs/

Installing Option 2: Pyspark

```
pip install pyspark
```

Installing Option 3: Docker (used in this Demo)

Install Docker

- https://docs.docker.com/desktop/setup/install/windows-install/
- https://docs.docker.com/desktop/setup/install/linux/
- https://docs.docker.com/desktop/setup/install/mac-install/

Run Apache/spark Docker

- Pull Docker

```
docker pull apache/spark
```

- Run Docker Container with spark-shell

```
docker run -it apache/spark /opt/spark/bin/spark-shell
```

Run Apache/spark Docker with mounted Learning directory

- Run Docker with with external directory for the training materials

```
docker run -it -v path/to/LearningSparkV2-master:/data apache/spark
/opt/spark/bin/spark-shell
```

Run Apache/spark Docker with mounted Learning directory and Avro Support

Option 1: use packages

- In Docker

```
/opt/spark/bin/spark-shelll --packages org.apache.spark:spark-avro_2.12:3.5.4
```

- When running Docker

```
docker run -it -v path/to/LearningSparkV2-master:/data apache/spark
/opt/spark/bin/spark-shell --packages org.apache.spark:spark-avro_2.12:3.5.4
```

Option 2: Use External Jar

```
docker run -it -v path/to/LearningSparkV2-master:/data apache/spark
/opt/spark/bin/spark-shell --jars /data/spark-avro_2.12-3.5.4.jar
```

Start Spark-Shell directly from container

```
/opt/spark/bin/spark-shell --jars /data/spark-avro_2.12-3.5.4.jar
```

Demo

General Spark

```
spark
```

```
spark.version
```

```
spark.catalog.currentDatabase
```

```
spark.sql("SHOW DATABASES").show()
spark.sql("CREATE DATABASE IF NOT EXISTS my_new_database")
spark.sql("USE my_new_database")
```

```
spark.sql("SHOW TABLES").show()
```

```
:paste
import org.apache.spark.sql.SparkSession
```

```
val spark = SparkSession.builder()
  .appName("NewTestName")
  .master("local[*]")
  .getOrCreate();
```

DataFrameReader

Parquet Example

- Add filepath

```
val file = "/data/databricks-datasets/learning-spark-v2/flights/summary-data/parquet/2010-summary.parquet"
```

- read the parquet file and store it in a dataframe

```
val df = spark.read.format("parquet").load(file);
```

- show only the first 10 entries

```
df.show(10);
```

```
+-----------------+-------------------+-----+
|DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-----------------+-------------------+-----+
|    United States|            Romania|    1|
|    United States|            Ireland|  264|
|    United States|              India|   69|
|            Egypt|      United States|   24|
|Equatorial Guinea|      United States|    1|
|    United States|          Singapore|   25|
|    United States|            Grenada|   54|
|       Costa Rica|      United States|  477|
|          Senegal|      United States|   29|
|    United States|   Marshall Islands|   44|
+-----------------+-------------------+-----+
```

CSV Example

```
:paste
val df2 = spark.read.format("csv")
```

```
.option("inferSchema", "true")
.option("header", "true")
.option("mode", "PERMISSIVE")
.load("/data/databricks-datasets/learning-spark-v2/flights/summary-
data/csv/*")
```

- show class

```
df3.getClass()
```

- show entries

```
df3.show(10);
```

```
+-----------------+-------------------+-----+
|DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-----------------+-------------------+-----+
|    United States|            Romania|    1|
|    United States|            Ireland|  264|
|    United States|              India|   69|
|            Egypt|      United States|   24|
|Equatorial Guinea|      United States|    1|
|    United States|          Singapore|   25|
|    United States|            Grenada|   54|
|       Costa Rica|      United States|  477|
|          Senegal|      United States|   29|
|    United States|   Marshall Islands|   44|
+-----------------+-------------------+-----+
```

JSON Example

```
:paste
val df4 = spark.read.format("json")
 .load("/data/databricks-datasets/learning-spark-v2/flights/summary-
data/json/*")
```

```
df4.show(10);
```

```
+-----------------+-------------------+-----+
|DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-----------------+-------------------+-----+
```

```
|    United States|            Romania|   15|
|    United States|            Croatia|    1|
|    United States|            Ireland|  344|
|            Egypt|      United States|   15|
|    United States|              India|   62|
|    United States|          Singapore|    1|
|    United States|            Grenada|   62|
|       Costa Rica|      United States|  588|
|          Senegal|      United States|   40|
|          Moldova|      United States|    1|
+-----------------+-------------------+-----+
```

DataFrameWriter

- General Syntax

```
//DataFrameWrite,
DataFrameWriter.format(args)
 .option(args)
 .bucketBy(args)
 .partitionBy(args)
 .save(path)
//toGetAnInstanceHandle
DataFrameWriter.format(args).option(args).sortBy(args).saveAsTable(table)

DataFrame.write
// or
DataFrame.writeStream
```

- create sequence

```
:paste
val data = Seq(
  ("Alice", 30, "USA"),
  ("Bob", 35, "UK"),
  ("Cathy", 28, "Canada")
)
```

```
val df = spark.createDataFrame(data).toDF("Name", "Age", "Country")
```

```
df.createOrReplaceTempView("people_table")
```

```
:paste
df.write
  .mode("overwrite")
  .saveAsTable("people_table")
```

```
val resultDf = spark.sql("SELECT * FROM people_table")
```

```
df.show()
```

```
+----+---+-------+
|Name|Age|Country|
+----+---+-------+
| Bob| 35|     UK|
+----+---+-------+
```

- Save the result as a JSON file

```
:paste
resultDf.write
  .format("json")
  .mode("overwrite")
  .save("/tmp/data/json/people.json")
```

```
val resultDf = spark.read.format("json").load("/tmp/abd/dfr/json/people.json")
```

```
resultDf.show();
+---+-------+-----+
|Age|Country| Name|
+---+-------+-----+
| 28| Canada|Cathy|
| 30|    USA|Alice|
| 35|     UK|  Bob|
+---+-------+-----+
```

Parquet

Parquet - Reading Parquet files into a DataFrame

```
:paste
val file = """/data/databricks-datasets/learning-spark-v2/flights/summary-data/parquet/2010-summary.parquet/"""

val dfpr1 = spark.read.format("parquet").load(file)

dfpr1.show()
```

```
+-------------------+-------------------+-----+
|   DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-------------------+-------------------+-----+
|       United States|            Romania|    1|
|       United States|            Ireland|  264|
|       United States|              India|   69|
|               Egypt|      United States|   24|
|   Equatorial Guinea|      United States|    1|
|       United States|          Singapore|   25|
|       United States|            Grenada|   54|
|          Costa Rica|      United States|  477|
|             Senegal|      United States|   29|
|       United States|   Marshall Islands|   44|
|              Guyana|      United States|   17|
|       United States|       Sint Maarten|   53|
|               Malta|      United States|    1|
|             Bolivia|      United States|   46|
|            Anguilla|      United States|   21|
|Turks and Caicos ...|      United States|  136|
|       United States|        Afghanistan|    2|
|Saint Vincent and...|      United States|    1|
|               Italy|      United States|  390|
|       United States|             Russia|  156|
+-------------------+-------------------+-----+
```

Reading Parquet files into a Spark SQL table

```
:paste
spark.sql("""
CREATE OR REPLACE TEMPORARY VIEW us_delay_flights_tbl_p
USING parquet
OPTIONS (
  path "/data/databricks-datasets/learning-spark-v2/flights/summary-data/parquet/2010-summary.parquet/"
```

```
)
""")
```

```
val dfpr2 = spark.sql("SELECT * FROM us_delay_flights_tbl_p")
```

```
dfpr2.show()
```

```
+-------------------+-------------------+-----+
|   DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-------------------+-------------------+-----+
|       United States|            Romania|    1|
|       United States|            Ireland|  264|
|       United States|              India|   69|
|               Egypt|      United States|   24|
|   Equatorial Guinea|      United States|    1|
|       United States|          Singapore|   25|
|       United States|            Grenada|   54|
|          Costa Rica|      United States|  477|
|             Senegal|      United States|   29|
|       United States|   Marshall Islands|   44|
|              Guyana|      United States|   17|
|       United States|       Sint Maarten|   53|
|               Malta|      United States|    1|
|             Bolivia|      United States|   46|
|            Anguilla|      United States|   21|
|Turks and Caicos ...|      United States|  136|
|       United States|        Afghanistan|    2|
|Saint Vincent and...|      United States|    1|
|               Italy|      United States|  390|
|       United States|             Russia|  156|
+-------------------+-------------------+-----+
```

Writing DataFrames to Parquet files

- compression: none

```
:paste
dfpr1.write.format("parquet")
  .mode("overwrite")
  .option("compression", "none")
  .save("/tmp/data/parquet/df_parquet")
```

- snappy compression

```
:paste
dfpr1.write.format("parquet")
  .mode("overwrite")
  .option("compression", "snappy")
  .save("/tmp/data/parquet/df_parquet")
```

Writing DataFrames to Spark SQL tables

```
val file = "/data/databricks-datasets/learning-spark-v2/flights/summary-data/parquet/2010-summary.parquet"
```

```
val dfpr3 = spark.read.format("parquet").load(file);
```

```
:paste
dfpr3.write
 .mode("overwrite")
 .saveAsTable("us_delay_flights_tbl_p2")
```

```
val df = spark.sql("SELECT * FROM us_delay_flights_tbl_p2")
```

```
df.show()
```

JSON

Reading a JSON file into a DataFrame

- schema is inferred

```
:paste
val files = Seq( "/data/databricks-datasets/learning-spark-v2/flights/summary-data/json/2012-summary.json", "/data/databricks-datasets/learning-spark-v2/flights/summary-data/json/2013-summary.json" )

val dfjr = spark.read.format("json").option("multiline", "false").load(files: _*) // Spread operator to pass multiple files
```

```
dfjr.printSchema
dfjr.show()
```

```
root
 |-- DEST_COUNTRY_NAME: string (nullable = true)
 |-- ORIGIN_COUNTRY_NAME: string (nullable = true)
 |-- count: long (nullable = true)
```

```
+-------------------+-------------------+-----+
|   DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-------------------+-------------------+-----+
|       United States|            Romania|   12|
|       United States|            Croatia|    1|
|       United States|            Ireland|  266|
|               Egypt|      United States|   13|
|       United States|              India|   60|
|   Equatorial Guinea|      United States|    1|
|       United States|              Niger|    1|
|       United States|          Singapore|   22|
|       United States|            Grenada|   40|
|          Costa Rica|      United States|  509|
|             Senegal|      United States|   28|
|              Guyana|      United States|   34|
|       United States|       Sint Maarten|  260|
|       United States|   Marshall Islands|   33|
|             Bolivia|      United States|   33|
|            Anguilla|      United States|   22|
|       United States|           Paraguay|   15|
|             Algeria|      United States|    2|
|Turks and Caicos ...|      United States|  181|
|Saint Vincent and...|      United States|    4|
+-------------------+-------------------+-----+
```

```
:paste
val rowCount = dfjr.count()
println(s"Total rows in DataFrame: $rowCount")
```

Reading a JSON file into a Spark SQL table

```
:paste
spark.sql("""
  CREATE OR REPLACE TEMPORARY VIEW us_delay_flights_tbl_j
```

```
  USING json
  OPTIONS (
    path "/data/databricks-datasets/learning-spark-v2/flights/summary-
data/json/*"
  )
""")
```

```
spark.sql("SELECT * FROM us_delay_flights_tbl_j").show()
```

```
+-------------------+-------------------+-----+
|   DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-------------------+-------------------+-----+
|       United States|            Romania|   15|
|       United States|            Croatia|    1|
|       United States|            Ireland|  344|
|               Egypt|      United States|   15|
|       United States|              India|   62|
|       United States|          Singapore|    1|
|       United States|            Grenada|   62|
|          Costa Rica|      United States|  588|
|             Senegal|      United States|   40|
|             Moldova|      United States|    1|
|       United States|       Sint Maarten|  325|
|       United States|   Marshall Islands|   39|
|              Guyana|      United States|   64|
|               Malta|      United States|    1|
|            Anguilla|      United States|   41|
|             Bolivia|      United States|   30|
|       United States|           Paraguay|    6|
|             Algeria|      United States|    4|
|Turks and Caicos ...|      United States|  230|
|       United States|          Gibraltar|    1|
+-------------------+-------------------+-----+
```

Writing DataFrames to JSON files

```
:paste
dfjr.write.format("json")
 .mode("overwrite")
 .option("compression", "none")
 .option("multiline", "false")
 .save("/tmp/data/json/df_json_none")
```

- with snappy compression

```
:paste
dfjr.write.format("json")
 .mode("overwrite")
 .option("compression", "snappy")
 .save("/tmp/data/json/df_json_snappy")
```

CSV

Reading a CSV file into a DataFrame

```
:paste

val fileCSV = "/data/databricks-datasets/learning-spark-v2/flights/summary-data/csv/*"

val dfCSV1 = spark.read.format("csv")
 .option("nullValue", "")
 .load(fileCSV)

val dfCSV1Header = spark.read.format("csv")
 .option("header", "true")
 .option("nullValue", "")
 .load(fileCSV)

val dfCSV1Infer = spark.read.format("csv")
 .option("header", "true")
 .option("inferschema", "true")
 .option("nullValue", "")
 .load(fileCSV)

val schema = "DEST_COUNTRY_NAME STRING, ORIGIN_COUNTRY_NAME STRING, count INT"

val dfCSVSchema = spark.read.format("csv")
 .schema(schema)
 .option("header", "true")
 .option("mode", "FAILFAST")
 .option("nullValue", "")
 .load(fileCSV)
```

```
:paste
dfCSV1.printSchema()
```

```
dfCSV1.show
dfCSV1Header.printSchema()
dfCSV1Header.show
dfCSV1Infer.printSchema()
dfCSV1Infer.show
dfCSVSchema.printSchema()
dfCSVSchema.show
```

- without header, infer -> false

```
root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
```

```
+--------------------+--------------------+-----+
|                 _c0|                 _c1|  _c2|
+--------------------+--------------------+-----+
|   DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
|       United States|             Romania|    1|
|       United States|             Ireland|  264|
|       United States|               India|   69|
|               Egypt|       United States|   24|
|   Equatorial Guinea|       United States|    1|
|       United States|           Singapore|   25|
|       United States|             Grenada|   54|
|          Costa Rica|       United States|  477|
|             Senegal|       United States|   29|
|       United States|    Marshall Islands|   44|
|              Guyana|       United States|   17|
|       United States|        Sint Maarten|   53|
|               Malta|       United States|    1|
|             Bolivia|       United States|   46|
|            Anguilla|       United States|   21|
|Turks and Caicos ...|       United States|  136|
|       United States|         Afghanistan|    2|
|Saint Vincent and...|       United States|    1|
|               Italy|       United States|  390|
+--------------------+--------------------+-----+
```

- Schema defined

```
root
 |-- DEST_COUNTRY_NAME: string (nullable = true)
```

```
 |-- ORIGIN_COUNTRY_NAME: string (nullable = true)
 |-- count: integer (nullable = true)
```

```
+-------------------+-------------------+-----+
|   DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-------------------+-------------------+-----+
|       United States|            Romania|    1|
|       United States|            Ireland|  264|
|       United States|              India|   69|
|               Egypt|      United States|   24|
|   Equatorial Guinea|      United States|    1|
|       United States|          Singapore|   25|
|       United States|            Grenada|   54|
|          Costa Rica|      United States|  477|
|             Senegal|      United States|   29|
|       United States|   Marshall Islands|   44|
|              Guyana|      United States|   17|
|       United States|       Sint Maarten|   53|
|               Malta|      United States|    1|
|             Bolivia|      United States|   46|
|            Anguilla|      United States|   21|
|Turks and Caicos ...|      United States|  136|
|       United States|        Afghanistan|    2|
|Saint Vincent and...|      United States|    1|
|               Italy|      United States|  390|
|       United States|             Russia|  156|
+-------------------+-------------------+-----+
```

Reading a CSV file into a Spark SQL table

```
:paste
 spark.sql("""
  CREATE OR REPLACE TEMPORARY VIEW us_delay_flights_tbl_c
  USING csv
  OPTIONS (
    path "/data/databricks-datasets/learning-spark-v2/flights/summary-
data/csv/*",
    header "true",
    inferSchema "true",
    mode "FAILFAST"
  )
""");
```

```
spark.sql("SHOW TABLES").show()
```

```
val dfCSV2 = spark.sql("SELECT * FROM us_delay_flights_tbl_c")
```

```
dfCSV2.printSchema()
```

```
root
 |-- DEST_COUNTRY_NAME: string (nullable = true)
 |-- ORIGIN_COUNTRY_NAME: string (nullable = true)
 |-- count: integer (nullable = true)
```

```
df.show()
```

```
+-------------------+-------------------+-----+
|   DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-------------------+-------------------+-----+
|       United States|            Romania|    1|
|       United States|            Ireland|  264|
|       United States|              India|   69|
|               Egypt|      United States|   24|
|   Equatorial Guinea|      United States|    1|
|       United States|          Singapore|   25|
|       United States|            Grenada|   54|
|          Costa Rica|      United States|  477|
|             Senegal|      United States|   29|
|       United States|   Marshall Islands|   44|
|              Guyana|      United States|   17|
|       United States|       Sint Maarten|   53|
|               Malta|      United States|    1|
|             Bolivia|      United States|   46|
|            Anguilla|      United States|   21|
|Turks and Caicos ...|      United States|  136|
|       United States|        Afghanistan|    2|
|Saint Vincent and...|      United States|    1|
|               Italy|      United States|  390|
|       United States|             Russia|  156|
+-------------------+-------------------+-----+
```

Create a persistent SQL Table

```
:paste
spark.sql("""
  CREATE TABLE us_delay_flights_tbl_table
  USING CSV
  OPTIONS (
    path "/data/databricks-datasets/learning-spark-v2/flights/summary-
data/csv/*",
    header "true",
    inferSchema "true",
    mode "FAILFAST"
  )
""")
```

```
:paste
spark.sql("SHOW TABLES").show(false)
val dfCSV_table = spark.sql("SELECT * FROM us_delay_flights_tbl_table")
dfCSV_table.printSchema()
dfCSV_table.show()
```

Writing DataFrames to CSV files

```
:paste
dfCSV1Infer.write
  .format("csv")
  .mode("overwrite")
  .option("compression","uncompressed")
  .option("sep", ";")
  .option("escape", "\n")
  .option("header", "false")
  .partitionBy("DEST_COUNTRY_NAME")
  .save("/tmp/data/csv/df_part_csv")
```

Avro

Include Avro Support

- include Avro Support when using docker run

```
docker run -it -v C:/Users/snout/Downloads/abd/LearningSparkV2-master:/data
apache/spark /opt/spark/bin/spark-shell --jars /data/spark-avro_2.12-3.5.4.jar
```

- include avro support in container

```
/opt/spark/bin/spark-shell --jars /data/spark-avro_2.12-3.5.4.jar
```

Reading an Avro file into a DataFrame

```
:paste
val dfavro = spark.read.format("avro")
.load("/data/databricks-datasets/learning-spark-v2/flights/summary-data/avro/*");
```

```
dfavro.show(false)
```

```
+-----------------------------+-------------------+-----+
|DEST_COUNTRY_NAME            |ORIGIN_COUNTRY_NAME|count|
+-----------------------------+-------------------+-----+
|United States                |Romania            |1    |
|United States                |Ireland            |264  |
|United States                |India              |69   |
|Egypt                        |United States      |24   |
|Equatorial Guinea            |United States      |1    |
|United States                |Singapore          |25   |
|United States                |Grenada            |54   |
|Costa Rica                   |United States      |477  |
|Senegal                      |United States      |29   |
|United States                |Marshall Islands   |44   |
|Guyana                       |United States      |17   |
|United States                |Sint Maarten       |53   |
|Malta                        |United States      |1    |
|Bolivia                      |United States      |46   |
|Anguilla                     |United States      |21   |
|Turks and Caicos Islands     |United States      |136  |
|United States                |Afghanistan        |2    |
|Saint Vincent and the Grenadines|United States      |1    |
|Italy                        |United States      |390  |
|United States                |Russia             |156  |
+-----------------------------+-------------------+-----+
```

Reading an Avro file into a Spark SQL table

```
:paste
spark.sql("""
  CREATE OR REPLACE TEMPORARY VIEW episode_tbl_a
  USING avro
```

```
  OPTIONS (
    path "/data/databricks-datasets/learning-spark-v2/flights/summary-
data/avro/*"
  )
""")
```

```
:paste
val dfavro2 = spark.sql("SELECT * FROM episode_tbl_a order by count")
dfavro2.show(false)
```

```
+-----------------------------+-------------------+-----+
|DEST_COUNTRY_NAME            |ORIGIN_COUNTRY_NAME|count|
+-----------------------------+-------------------+-----+
|United States                |Romania            |1    |
|United States                |Ireland            |264  |
|United States                |India              |69   |
|Egypt                        |United States      |24   |
|Equatorial Guinea            |United States      |1    |
|United States                |Singapore          |25   |
|United States                |Grenada            |54   |
|Costa Rica                   |United States      |477  |
|Senegal                      |United States      |29   |
|United States                |Marshall Islands   |44   |
|Guyana                       |United States      |17   |
|United States                |Sint Maarten       |53   |
|Malta                        |United States      |1    |
|Bolivia                      |United States      |46   |
|Anguilla                     |United States      |21   |
|Turks and Caicos Islands     |United States      |136  |
|United States                |Afghanistan        |2    |
|Saint Vincent and the Grenadines|United States      |1    |
|Italy                        |United States      |390  |
|United States                |Russia             |156  |
+-----------------------------+-------------------+-----+
```

Writing DataFrames to Avro files

```
:paste
dfavro2.write
 .format("avro")
 .mode("overwrite")
 .option("recordName","TestDonnerstagRecord")
 .save("/tmp/data/avro/df_avro")
```

ORC

Reading an ORC file into a DataFrame

```
val fileORC = "/data/databricks-datasets/learning-spark-v2/flights/summary-data/orc/*";
```

```
val dfORC1 = spark.read.format("orc").load(fileORC);
```

```
dfORC1.show(10, false);
```

```
+-----------------+-------------------+-----+
|DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-----------------+-------------------+-----+
|United States    |Romania            |1    |
|United States    |Ireland            |264  |
|United States    |India              |69   |
|Egypt            |United States      |24   |
|Equatorial Guinea|United States      |1    |
|United States    |Singapore          |25   |
|United States    |Grenada            |54   |
|Costa Rica       |United States      |477  |
|Senegal          |United States      |29   |
|United States    |Marshall Islands   |44   |
+-----------------+-------------------+-----+
```

Reading an ORC file into a Spark SQL table

```
:paste
spark.sql("""
  CREATE OR REPLACE TEMPORARY VIEW us_delay_flights_tbl_o
  USING orc
  OPTIONS (
    path "/data/databricks-datasets/learning-spark-v2/flights/summary-data/orc/*"
  )
""")
```

```
val dfORC2 = spark.sql("SELECT * FROM us_delay_flights_tbl_o")
```

```
dfORC2.show()
```

```
+-----------------+-----------------+-----+
|  DEST_COUNTRY_NAME|ORIGIN_COUNTRY_NAME|count|
+-----------------+-----------------+-----+
|       United States|            Romania|    1|
|       United States|            Ireland|  264|
|       United States|              India|   69|
|               Egypt|      United States|   24|
|   Equatorial Guinea|      United States|    1|
|       United States|          Singapore|   25|
|       United States|            Grenada|   54|
|          Costa Rica|      United States|  477|
|             Senegal|      United States|   29|
|       United States|   Marshall Islands|   44|
|              Guyana|      United States|   17|
|       United States|       Sint Maarten|   53|
|               Malta|      United States|    1|
|             Bolivia|      United States|   46|
|            Anguilla|      United States|   21|
|Turks and Caicos ...|      United States|  136|
|       United States|        Afghanistan|    2|
|Saint Vincent and...|      United States|    1|
|               Italy|      United States|  390|
|       United States|             Russia|  156|
+-----------------+-----------------+-----+
```

Writing DataFrames to ORC files

```
:paste
dfORC2.write.format("orc")
 .mode("overwrite")
 .option("compression", "uncompressed")
 .save("/tmp/data/orc/df_orc")
```

```
:paste
dfORC2.write.format("orc")
 .mode("overwrite")
 .option("compression", "snappy")
 .save("/tmp/data/orc/df_orc")
```

- none, snappy, zlib, lzo, zstd and lz4)

Images

Reading an image file into a DataFrame

```
:paste
import org.apache.spark.ml.source.image
val imageDir = "/data/databricks-datasets/learning-spark-
v2/cctvVideos/train_images/"
val imagesDF = spark.read.format("image").load(imageDir)
imagesDF.printSchema
```

- Structure

```
root
 |-- image: struct (nullable = true)
 |    |-- origin: string (nullable = true)
 |    |-- height: integer (nullable = true)
 |    |-- width: integer (nullable = true)
 |    |-- nChannels: integer (nullable = true)
 |    |-- mode: integer (nullable = true)
 |    |-- data: binary (nullable = true)
 |-- label: integer (nullable = true)
```

```
imagesDF.select("image.height", "image.width", "image.nChannels",
"image.mode","label").show(5, false)
```

```
+------+-----+---------+----+-----+
|height|width|nChannels|mode|label|
+------+-----+---------+----+-----+
|288   |384  |3        |16  |0    |
|288   |384  |3        |16  |1    |
|288   |384  |3        |16  |0    |
|288   |384  |3        |16  |0    |
|288   |384  |3        |16  |0    |
+------+-----+---------+----+-----+
```

Binary

Reading a binary file into a DataFrame

```
:paste
val pathBinary = "/data/databricks-datasets/learning-spark-
v2/cctvVideos/train_images/"
val binaryFilesDF1 = spark.read.format("binaryFile")
 .option("pathGlobFilter", "*.jpg")
 .load(pathBinary)
binaryFilesDF1.printSchema();
binaryFilesDF1.show(5)
```

```
root
 |-- path: string (nullable = true)
 |-- modificationTime: timestamp (nullable = true)
 |-- length: long (nullable = true)
 |-- content: binary (nullable = true)
 |-- label: integer (nullable = true)
```

```
+--------------------+--------------------+------+--------------------+-----+
|                path|    modificationTime|length|             content|label|
+--------------------+--------------------+------+--------------------+-----+
|file:/data/databr...|2025-01-22 00:26:...| 55037|[FF D8 FF E0 00 1...|    0|
|file:/data/databr...|2025-01-22 00:26:...| 54634|[FF D8 FF E0 00 1...|    1|
|file:/data/databr...|2025-01-22 00:26:...| 54624|[FF D8 FF E0 00 1...|    0|
|file:/data/databr...|2025-01-22 00:26:...| 54505|[FF D8 FF E0 00 1...|    0|
|file:/data/databr...|2025-01-22 00:26:...| 54475|[FF D8 FF E0 00 1...|    0|
+--------------------+--------------------+------+--------------------+-----+
```

```
binaryFilesDF1.show(1,false)
```

Ignore partitioning data discvovery

```
:paste
val binaryFilesDF2 = spark.read.format("binaryFile")
 .option("pathGlobFilter", "*.jpg")
 .option("recursiveFileLookup", "true")
 .load(pathBinary)
```

```
binaryFilesDF2.printSchema();
```

```
root
 |-- path: string (nullable = true)
```

```
 |-- modificationTime: timestamp (nullable = true)
 |-- length: long (nullable = true)
 |-- content: binary (nullable = true)
```

```
binaryFilesDF2.show(5);
```

```
+--------------------+--------------------+------+--------------------+
|                path|    modificationTime|length|             content|
+--------------------+--------------------+------+--------------------+
|file:/data/databr...|2025-01-22 00:26:...| 54634|[FF D8 FF E0 00 1...|
|file:/data/databr...|2025-01-22 00:26:...| 54365|[FF D8 FF E0 00 1...|
|file:/data/databr...|2025-01-22 00:26:...| 54330|[FF D8 FF E0 00 1...|
|file:/data/databr...|2025-01-22 00:26:...| 54244|[FF D8 FF E0 00 1...|
|file:/data/databr...|2025-01-22 00:26:...| 54200|[FF D8 FF E0 00 1...|
+--------------------+--------------------+------+--------------------+
```

- writing binary data back to dataframe is not supported.

Demo Schema Evolution in Parquet

- https://spark.apache.org/docs/3.5.1/sql-data-sources-parquet.html

```
// This is used to implicitly convert an RDD to a DataFrame.
// Create a simple DataFrame, store into a partition directory
:paste
import spark.implicits._
val squaresDF = spark.sparkContext.makeRDD(1 to 5).map(i => (i, i *
i)).toDF("value", "square")
squaresDF.write.parquet("/tmp/data/parquet/test_table/key=1")

// Create another DataFrame in a new partition directory,
// adding a new column and dropping an existing column
:paste
val cubesDF = spark.sparkContext.makeRDD(6 to 10).map(i => (i, i * i *
i)).toDF("value", "cube")
cubesDF.write.parquet("/tmp/data/parquet/test_table/key=2")

:paste
squaresDF.show
cubesDF.show

+-----+------+
|value|square|
```

```
+-----+------+
|    1|     1|
|    2|     4|
|    3|     9|
|    4|    16|
|    5|    25|
+-----+------+
+-----+----+
|value|cube|
+-----+----+
|    6| 216|
|    7| 343|
|    8| 512|
|    9| 729|
|   10|1000|
+-----+----+

// Read the partitioned table
val mergedDF = spark.read.option("mergeSchema",
"true").parquet("/tmp/data/parquet/test_table")

mergedDF.show()
+-----+------+----+---+
|value|square|cube|key|
+-----+------+----+---+
|    1|     1|NULL|  1|
|    2|     4|NULL|  1|
|    4|    16|NULL|  1|
|    5|    25|NULL|  1|
|    3|     9|NULL|  1|
|    6|  NULL| 216|  2|
|    7|  NULL| 343|  2|
|    8|  NULL| 512|  2|
|    9|  NULL| 729|  2|
|   10|  NULL|1000|  2|
+-----+------+----+---+

mergedDF.printSchema()
// The final schema consists of all 3 columns in the Parquet files together
// with the partitioning column appeared in the partition directory paths
// root
//  |-- value: int (nullable = true)
//  |-- square: int (nullable = true)
//  |-- cube: int (nullable = true)
//  |-- key: int (nullable = true)
```