In [2]:

```
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.types import *

spark = SparkSession\
    .builder\
    .appName("chapter-09-data-src")\
    .getOrCreate()

import os
SPARK_BOOK_DATA_PATH = os.environ['SPARK_BOOK_DATA_PATH']
```

In [3]:

```
file_path = SPARK_BOOK_DATA_PATH + "/data/flight-data/csv/2010-summary.

csvFile = spark.read.format("csv")\
  .option("header", "true")\
  .option("mode", "FAILFAST")\
  .option("inferSchema", "true")\
  .load(file_path)
```

In [5]:

```
csvFile.show(5)
```

```
+-----------------+-------------------+-----+
|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|
+-----------------+-------------------+-----+
only showing top 5 rows
```

In [5]:

```
# COMMAND ----------

csvFile.write.format("csv").mode("overwrite").option("sep", "\t")\
  .save("/tmp/my-tsv-file.tsv")
```

In [6]:

```
# COMMAND ----------

file_path = SPARK_BOOK_DATA_PATH + "/data/flight-data/json/2010-summary
csvFile = spark.read.format("json").option("mode", "FAILFAST")\
  .option("inferSchema", "true")\
  .load(file_path)

csvFile.show(5)
```

```
+-----------------+-------------------+-----+
|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|
+-----------------+-------------------+-----+
only showing top 5 rows
```

In [7]:

```
# COMMAND ----------

csvFile.write.format("json").mode("overwrite").save("/tmp/my-json-file.
```

In [8]:

```
# COMMAND ----------
file_path = SPARK_BOOK_DATA_PATH + "/data/flight-data/parquet/2010-summa
csvFile = spark.read.format("parquet")\
  .load(file_path)

csvFile.show(5)
```

```
+-----------------+-------------------+-----+
|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|
+-----------------+-------------------+-----+
only showing top 5 rows
```

In [9]:

```
# COMMAND ----------

csvFile.write.format("parquet").mode("overwrite")\
  .save("/tmp/my-parquet-file.parquet")
```

In [10]:

```
# COMMAND ----------
file_path = SPARK_BOOK_DATA_PATH + "/data/flight-data/orc/2010-summary.
csvFile = spark.read.format("orc").load(file_path)

csvFile.show(5)
```

```
+-----------------+-------------------+-----+
|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|
+-----------------+-------------------+-----+
only showing top 5 rows
```

In [14]:

```
# COMMAND ----------

csvFile.write.format("orc").mode("overwrite").save("/tmp/my-json-file.o
```

In [6]:

```
# COMMAND ----------
file_path = SPARK_BOOK_DATA_PATH + "/data/flight-data/jdbc/my-sqlite.db
driver = "org.sqlite.JDBC"
path = file_path
url = "jdbc:sqlite:" + path
tablename = "flight_info"
```

In [7]:

```
# COMMAND ----------

dbDataFrame = spark.read.format("jdbc")\
    .option("url", url)\
    .option("dbtable", tablename)\
    .option("driver",  driver)\
    .load()
```

In [8]:

```
dbDataFrame.show(5)
```

```
+-----------------+-------------------+-----+
|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|
+-----------------+-------------------+-----+
only showing top 5 rows
```

In [ ]:

```
# COMMAND ----------

pgDF = spark.read.format("jdbc")\
  .option("driver", "org.postgresql.Driver")\
  .option("url", "jdbc:postgresql://database_server")\
  .option("dbtable", "schema.tablename")\
  .option("user", "username").option("password", "my-secret-password").


# COMMAND ----------

dbDataFrame.filter("DEST_COUNTRY_NAME in ('Anguilla', 'Sweden')").expla


# COMMAND ----------

pushdownQuery = """(SELECT DISTINCT(DEST_COUNTRY_NAME) FROM flight_info
  AS flight_info"""
dbDataFrame = spark.read.format("jdbc")\
  .option("url", url).option("dbtable", pushdownQuery).option("driver",
  .load()
```

In [ ]:

```
# COMMAND ----------

dbDataFrame = spark.read.format("jdbc")\
  .option("url", url).option("dbtable", tablename).option("driver",  dr
  .option("numPartitions", 10).load()


# COMMAND ----------

props = {"driver":"org.sqlite.JDBC"}
predicates = [
  "DEST_COUNTRY_NAME = 'Sweden' OR ORIGIN_COUNTRY_NAME = 'Sweden'",
  "DEST_COUNTRY_NAME = 'Anguilla' OR ORIGIN_COUNTRY_NAME = 'Anguilla'"]
spark.read.jdbc(url, tablename, predicates=predicates, properties=props
```

In [ ]:

```
spark.read.jdbc(url,tablename,predicates=predicates,properties=props)\
  .rdd.getNumPartitions() # 2
```

In [9]:

```
# COMMAND ----------

props = {"driver":"org.sqlite.JDBC"}
predicates = [
  "DEST_COUNTRY_NAME != 'Sweden' OR ORIGIN_COUNTRY_NAME != 'Sweden'",
  "DEST_COUNTRY_NAME != 'Anguilla' OR ORIGIN_COUNTRY_NAME != 'Anguilla'
spark.read.jdbc(url, tablename, predicates=predicates, properties=props
```

Out[9]:

```
510
```

In [ ]:

```
# COMMAND ----------

colName = "count"
lowerBound = 0L
upperBound = 348113L # this is the max count in our database
numPartitions = 10


# COMMAND ----------

spark.read.jdbc(url, tablename, column=colName, properties=props,
                lowerBound=lowerBound, upperBound=upperBound,
                numPartitions=numPartitions).count() # 255
```

In [ ]:

```
# COMMAND ----------

newPath = "jdbc:sqlite://tmp/my-sqlite.db"
csvFile.write.jdbc(newPath, tablename, mode="overwrite", properties=prop
```

In [ ]:

```
# COMMAND ----------

spark.read.jdbc(newPath, tablename, properties=props).count() # 255
```

In [ ]:

```
# COMMAND ----------

csvFile.write.jdbc(newPath, tablename, mode="append", properties=props)
```

In [ ]:

```
# COMMAND ----------

spark.read.jdbc(newPath, tablename, properties=props).count() # 765
```

In [ ]:

```
# COMMAND ----------

csvFile.limit(10).select("DEST_COUNTRY_NAME", "count")\
  .write.partitionBy("count").text("/tmp/five-csv-files2py.csv")
```

In [ ]:

```
# COMMAND ----------

csvFile.limit(10).write.mode("overwrite").partitionBy("DEST_COUNTRY_NAM
  .save("/tmp/partitioned-files.parquet")


# COMMAND ----------
```