this chapter is best to use [databricks community cluster (https://community.cloud.databricks.com)](https://community.cloud.databricks.com) to practise

see `../SparkSQL.sql` and `../SparkSQL.html`

In [2]:
```
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.types import *

spark = SparkSession\
    .builder\
    .appName("chapter-10-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/json/2015-summary.json"

spark.read.json(file_path)\
  .createOrReplaceTempView("some_sql_view") # DF => SQL
```

In [5]:
```
df = spark.sql("""
SELECT DEST_COUNTRY_NAME, sum(count)
FROM some_sql_view GROUP BY DEST_COUNTRY_NAME
""")\
  .where("DEST_COUNTRY_NAME like 'S%'").where("`sum(count)` > 10")
# SQL => DF


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

In [6]:
```
df.show(5)
```

```
+--------------------+----------+
|   DEST_COUNTRY_NAME|sum(count)|
+--------------------+----------+
|             Senegal|        40|
|              Sweden|       118|
|               Spain|       420|
|    Saint Barthelemy|        39|
|Saint Kitts and N...|       139|
+--------------------+----------+
only showing top 5 rows
```

In [ ]:
```
CREATE TABLE flights (
  DEST_COUNTRY_NAME STRING, ORIGIN_COUNTRY_NAME STRING, count LONG)
USING JSON OPTIONS (path '/data/flight-data/json/2015-summary.json')


-- COMMAND ----------

CREATE TABLE flights_csv (
  DEST_COUNTRY_NAME STRING,
  ORIGIN_COUNTRY_NAME STRING COMMENT "remember, the US will be most prevalent",
  count LONG)
USING csv OPTIONS (header true, path '/data/flight-data/csv/2015-summary.csv')


-- COMMAND ----------

CREATE TABLE flights_from_select USING parquet AS SELECT * FROM flights


-- COMMAND ----------

CREATE TABLE IF NOT EXISTS flights_from_select
  AS SELECT * FROM flights


-- COMMAND ----------

CREATE TABLE partitioned_flights USING parquet PARTITIONED BY (DEST_COUNTRY_NAME)
AS SELECT DEST_COUNTRY_NAME, ORIGIN_COUNTRY_NAME, count FROM flights LIMIT 5


-- COMMAND ----------

CREATE EXTERNAL TABLE hive_flights (
  DEST_COUNTRY_NAME STRING, ORIGIN_COUNTRY_NAME STRING, count LONG)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/data/flight-data-hive/'


-- COMMAND ----------

CREATE EXTERNAL TABLE hive_flights_2
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
```

```
LOCATION '/data/flight-data-hive/' AS SELECT * FROM flights


-- COMMAND ----------

INSERT INTO flights_from_select
  SELECT DEST_COUNTRY_NAME, ORIGIN_COUNTRY_NAME, count FROM flights LIMIT 20


-- COMMAND ----------

INSERT INTO partitioned_flights
  PARTITION (DEST_COUNTRY_NAME="UNITED STATES")
  SELECT count, ORIGIN_COUNTRY_NAME FROM flights
  WHERE DEST_COUNTRY_NAME='UNITED STATES' LIMIT 12


-- COMMAND ----------

DESCRIBE TABLE flights_csv


-- COMMAND ----------

SHOW PARTITIONS partitioned_flights


-- COMMAND ----------

REFRESH table partitioned_flights


-- COMMAND ----------

MSCK REPAIR TABLE partitioned_flights


-- COMMAND ----------

DROP TABLE flights_csv;


-- COMMAND ----------
```

```
DROP TABLE IF EXISTS flights_csv;


-- COMMAND ----------

CACHE TABLE flights


-- COMMAND ----------

UNCACHE TABLE FLIGHTS


-- COMMAND ----------

CREATE VIEW just_usa_view AS
  SELECT * FROM flights WHERE dest_country_name = 'United States'


-- COMMAND ----------

CREATE TEMP VIEW just_usa_view_temp AS
  SELECT * FROM flights WHERE dest_country_name = 'United States'


-- COMMAND ----------

CREATE GLOBAL TEMP VIEW just_usa_global_view_temp AS
  SELECT * FROM flights WHERE dest_country_name = 'United States'


-- COMMAND ----------

SHOW TABLES


-- COMMAND ----------

CREATE OR REPLACE TEMP VIEW just_usa_view_temp AS
  SELECT * FROM flights WHERE dest_country_name = 'United States'
```

```
-- COMMAND ----------

SELECT * FROM just_usa_view_temp

-- COMMAND ----------

EXPLAIN SELECT * FROM just_usa_view

-- COMMAND ----------

EXPLAIN SELECT * FROM flights WHERE dest_country_name = 'United States'

-- COMMAND ----------

DROP VIEW IF EXISTS just_usa_view;

-- COMMAND ----------

SHOW DATABASES

-- COMMAND ----------

CREATE DATABASE some_db

-- COMMAND ----------

USE some_db

-- COMMAND ----------

SHOW tables

SELECT * FROM flights -- fails with table/view not found

-- COMMAND ----------
```

```
SELECT * FROM default.flights
```

```
-- COMMAND ----------

SELECT current_database()
```

```
-- COMMAND ----------

USE default;
```

```
-- COMMAND ----------

DROP DATABASE IF EXISTS some_db;
```

```
-- COMMAND ----------

SELECT [ALL|DISTINCT] named_expression[, named_expression, ...]
    FROM relation[, relation, ...]
    [lateral_view[, lateral_view, ...]]
    [WHERE boolean_expression]
    [aggregation [HAVING boolean_expression]]
    [ORDER BY sort_expressions]
    [CLUSTER BY expressions]
    [DISTRIBUTE BY expressions]
    [SORT BY sort_expressions]
    [WINDOW named_window[, WINDOW named_window, ...]]
    [LIMIT num_rows]

named_expression:
    : expression [AS alias]

relation:
    | join_relation
    | (table_name|query|relation) [sample] [AS alias]
    : VALUES (expressions)[, (expressions), ...]
          [AS (column_name[, column_name, ...])]

expressions:
```

```
    : expression[, expression, ...]

sort_expressions:
    : expression [ASC|DESC][, expression [ASC|DESC], ...]


-- COMMAND ----------

SELECT
  CASE WHEN DEST_COUNTRY_NAME = 'UNITED STATES' THEN 1
       WHEN DEST_COUNTRY_NAME = 'Egypt' THEN 0
       ELSE -1 END
FROM partitioned_flights


-- COMMAND ----------

CREATE VIEW IF NOT EXISTS nested_data AS
  SELECT (DEST_COUNTRY_NAME, ORIGIN_COUNTRY_NAME) as country, count FROM flights


-- COMMAND ----------

SELECT * FROM nested_data


-- COMMAND ----------

SELECT country.DEST_COUNTRY_NAME, count FROM nested_data


-- COMMAND ----------

SELECT country.*, count FROM nested_data


-- COMMAND ----------

SELECT DEST_COUNTRY_NAME as new_name, collect_list(count) as flight_counts,
  collect_set(ORIGIN_COUNTRY_NAME) as origin_set
FROM flights GROUP BY DEST_COUNTRY_NAME
```

```
-- COMMAND ----------

SELECT DEST_COUNTRY_NAME, ARRAY(1, 2, 3) FROM flights


-- COMMAND ----------

SELECT DEST_COUNTRY_NAME as new_name, collect_list(count)[0]
FROM flights GROUP BY DEST_COUNTRY_NAME


-- COMMAND ----------

CREATE OR REPLACE TEMP VIEW flights_agg AS
  SELECT DEST_COUNTRY_NAME, collect_list(count) as collected_counts
  FROM flights GROUP BY DEST_COUNTRY_NAME


-- COMMAND ----------

SELECT explode(collected_counts), DEST_COUNTRY_NAME FROM flights_agg


-- COMMAND ----------

SHOW FUNCTIONS


-- COMMAND ----------

SHOW SYSTEM FUNCTIONS


-- COMMAND ----------

SHOW USER FUNCTIONS


-- COMMAND ----------

SHOW FUNCTIONS "s*";
```

```
-- COMMAND ----------

SHOW FUNCTIONS LIKE "collect*";


-- COMMAND ----------

SELECT count, power3(count) FROM flights


-- COMMAND ----------

SELECT dest_country_name FROM flights
GROUP BY dest_country_name ORDER BY sum(count) DESC LIMIT 5


-- COMMAND ----------

SELECT * FROM flights
WHERE origin_country_name IN (SELECT dest_country_name FROM flights
      GROUP BY dest_country_name ORDER BY sum(count) DESC LIMIT 5)


-- COMMAND ----------

SELECT * FROM flights f1
WHERE EXISTS (SELECT 1 FROM flights f2
            WHERE f1.dest_country_name = f2.origin_country_name)
AND EXISTS (SELECT 1 FROM flights f2
            WHERE f2.dest_country_name = f1.origin_country_name)


-- COMMAND ----------

SELECT *, (SELECT max(count) FROM flights) AS maximum FROM flights


-- COMMAND ----------

SET spark.sql.shuffle.partitions=20


-- COMMAND ----------
```