Spark SQL

INTRODUCTION TO SPARK SQL IN PYTHON

Mark Plutowski Phd
Data Scientist

Create SQL table and query it

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/

Using Python version 3.6
SparkSession available as 'spark'.
>>> |
```

Load a dataframe from file

```
df = spark.read.csv(filename)
```

```
df = spark.read.csv(filename, header=True)
```

Create SQL table and query it

```
df.createOrReplaceTempView("schedule")
spark.sql("SELECT * FROM schedule WHERE station = 'San Jose'")
    .show()
```

```
+--------+--------+-----+
|train_id| station| time|
+--------+--------+-----+
|     324|San Jose|9:05a|
|     217|San Jose|6:59a|
+--------+--------+-----+
```
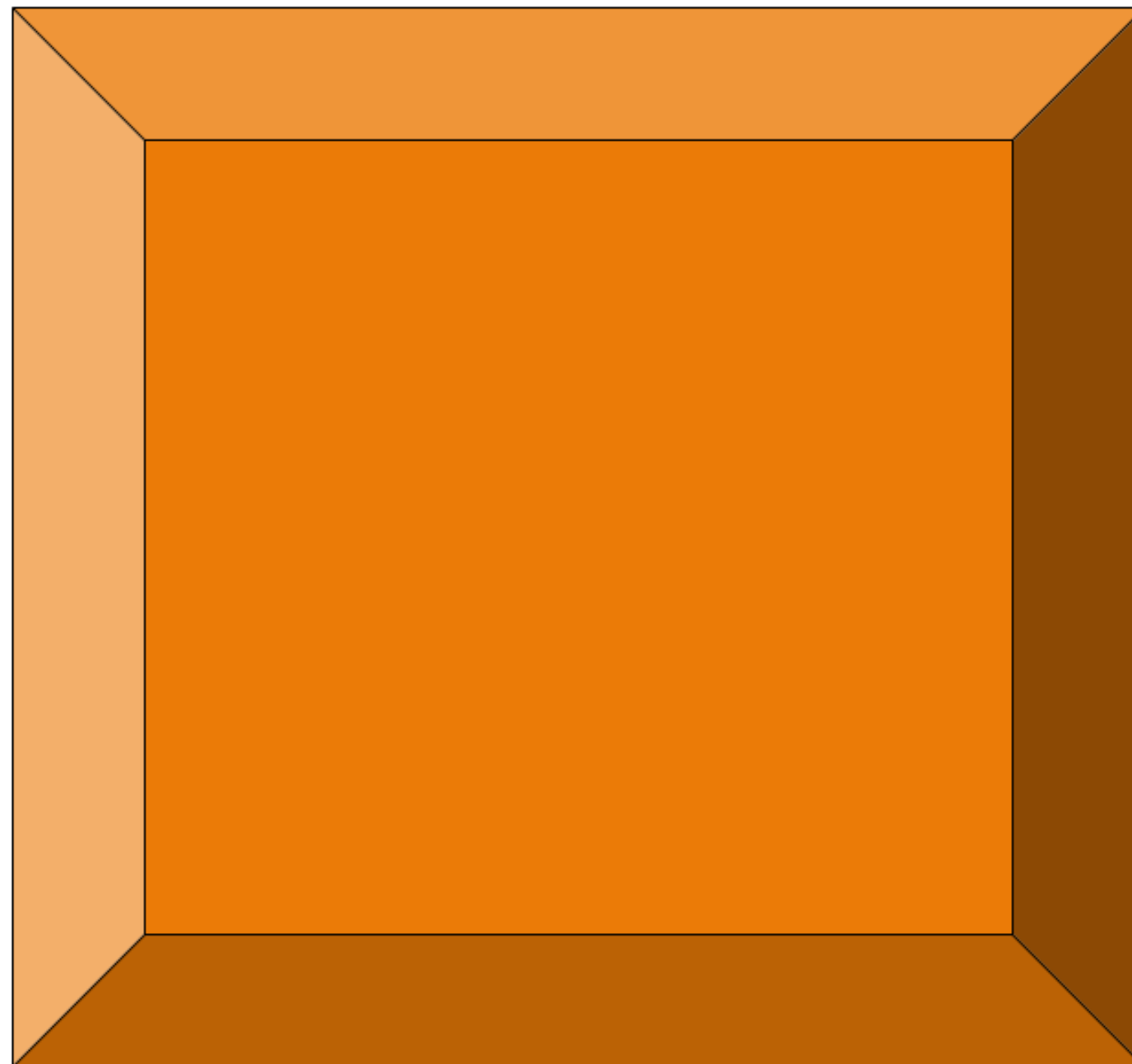
Inspecting table schema

```
result = spark.sql("SHOW COLUMNS FROM tablename")
```
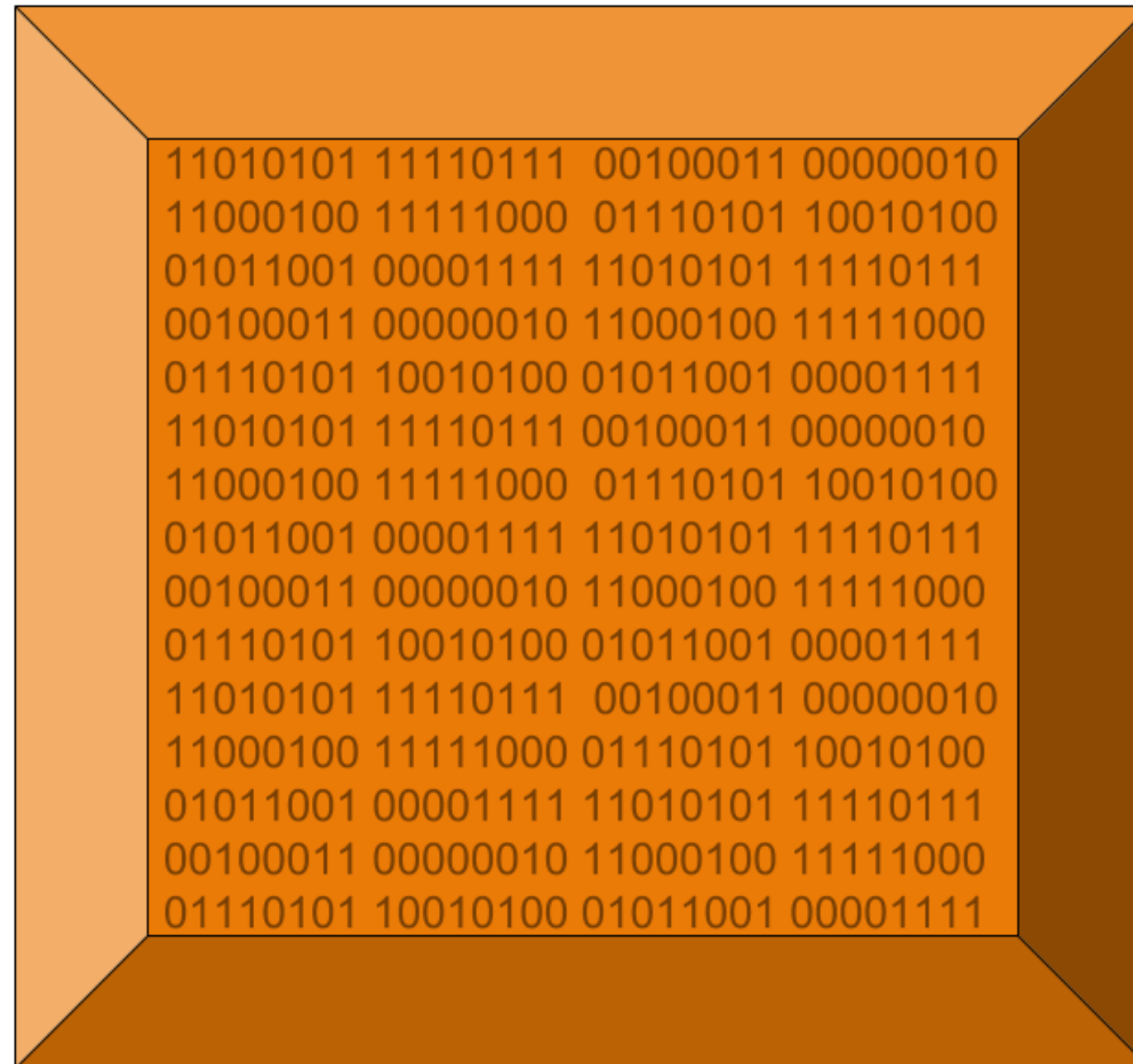
```
result = spark.sql("SELECT * FROM tablename LIMIT 0")
```

```
result = spark.sql("DESCRIBE tablename")
```

```
result.show()
```
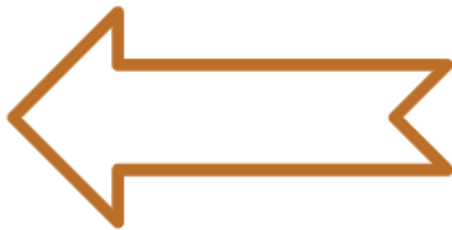
```
print(result.columns)
```

Tabular data

```
+--------+-------------+-----+
|train_id|      station| time|
+--------+-------------+-----+
|     324|San Francisco|7:59a|
|     324|   22nd Street|8:03a|
|     324|      Millbrae|8:16a|
|     324|     Hillsdale|8:24a|
|     324|  Redwood City|8:31a|
|     324|     Palo Alto|8:37a|
|     324|      San Jose|9:05a|
|     217|        Gilroy|6:06a|
|     217|    San Martin|6:15a|
|     217|   Morgan Hill|6:21a|
|     217|  Blossom Hill|6:36a|
|     217|       Capitol|6:42a|
|     217|        Tamien|6:50a|
|     217|      San Jose|6:59a|
+--------+-------------+-----+
```

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

train_id	station	time
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

train_id	station	time
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

train_id	station	time
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

train_id	station	time
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

train_id	station	time
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

train_id	station	time
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

train_id	station	time
324	San Francisco	7:59a
324	22nd Street	8:03a
324	Millbrae	8:16a
324	Hillsdale	8:24a
324	Redwood City	8:31a
324	Palo Alto	8:37a
324	San Jose	9:05a

Server 1

Server 2

train_id	station	time
217	Gilroy	6:06a
217	San Martin	6:15a
217	Morgan Hill	6:21a
217	Blossom Hill	6:36a
217	Capitol	6:42a
217	Tamien	6:50a
217	San Jose	6:59a

S
Q
L

Structured Query Language
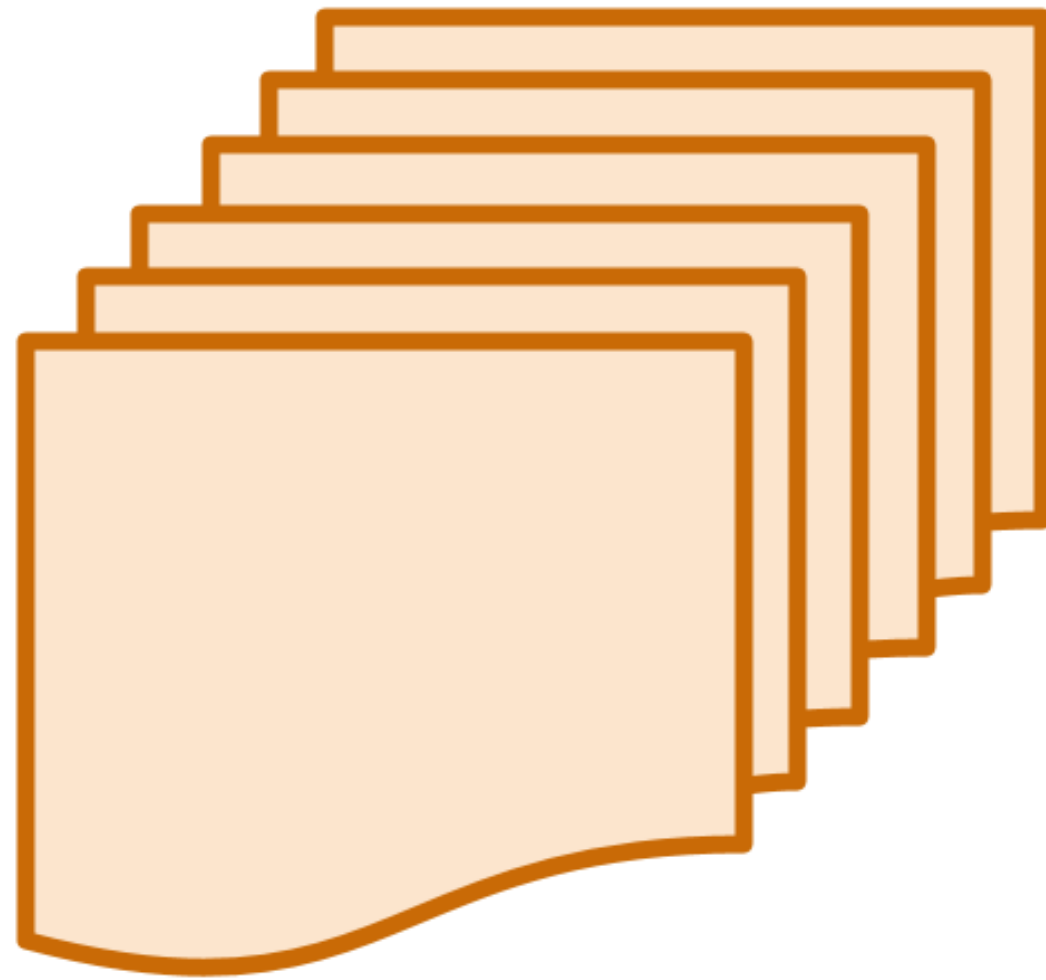
Loading delimited text

Loads a comma-delimited file `trainsched.txt` into a dataframe called `df` :

```
df = spark.read.csv("trainsched.txt", header=True)
```

Loading delimited text

```
df = spark.read.csv("trainsched.txt", header=True)
df.show()
```

```
+--------+------------+-----+
|train_id|     station| time|
+--------+------------+-----+
|     324|San Francisco|7:59a|
|     324|  22nd Street|8:03a|
|     324|     Millbrae|8:16a|
|     324|    Hillsdale|8:24a|
|     324| Redwood City|8:31a|
|     ...|          ...|  ...|
|     217| Blossom Hill|6:36a|
|     217|      Capitol|6:42a|
|     217|       Tamien|6:50a|
|     217|     San Jose|6:59a|
+--------+------------+-----+
```

Welcome to

```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/
```

Using Python version 3.6
SparkSession available as 'spark'.
>>>

Welcome to

```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/
```

Using Python version 3.6
SparkSession available as 'spark'.
>>> |

Welcome to

```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/
```

Using Python version 3.6
SparkSession available as 'spark'.
>>> |

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/

Using Python version 3.6
SparkSession available as 'spark'.
>>>
```

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/

Using Python version 3.6
SparkSession available as 'spark'.
>>> |
```

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/

Using Python version 3.6
SparkSession available as 'spark'.
>>> |
```

Welcome to

```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/
```

Using Python version 3.6
SparkSession available as 'spark'.
>>>

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\
      /_/

Using Python version 3.6
SparkSession available as 'spark'.
>>> |
```

Let's practice

datacamp

Window Function SQL

INTRODUCTION TO SPARK SQL IN PYTHON

Mark Plutowski
Data Scientist

What is a Window Function SQL?

- Express operations more simply than dot notation or queries

- Each row uses the values of other rows to calculate its value

A train schedule

train_id	station	time
324	San Francisco	7:59
324	22nd Street	8:03
324	Millbrae	8:16
324	Hillsdale	8:24
324	Redwood City	8:31
324	Palo Alto	8:37
324	San Jose	9:05

Column with time until next stop added

train_id	station	time	time_to_next_stop
324	San Francisco	7:59	4 min
324	22nd Street	8:03	13 min
324	Millbrae	8:16	8 min
324	Hillsdale	8:24	7 min
324	Redwood City	8:31	6 min
324	Palo Alto	8:37	28 min
324	San Jose	9:05	null

Column with time of next stop

train_id	station	time	time (following row)
324	San Francisco	7:59	8:03
324	22nd Street	8:03	8:16
324	Millbrae	8:16	8:24
324	Hillsdale	8:24	8:31
324	Redwood City	8:31	8:37
324	Palo Alto	8:37	9:05
324	San Jose	9:05	null

OVER clause and ORDER BY clause

```
query = """
SELECT train_id, station, time,
LEAD(time, 1) OVER (ORDER BY time) AS time_next
FROM sched
WHERE train_id=324 """

spark.sql(query).show()
```

```
+--------+-----------+-----+---------+
|train_id|    station| time|time_next|
+--------+-----------+-----+---------+

|     324|San Francisco|7:59 |    8:03 |
|     324|  22nd Street|8:03 |    8:16 |
|     324|     Millbrae|8:16 |    8:24 |
|     324|    Hillsdale|8:24 |    8:31 |
|     324| Redwood City|8:31 |    8:37 |
|     324|    Palo Alto|8:37 |    9:05 |
|     324|     San Jose|9:05 |    null |
+--------+-----------+-----+---------+
```

PARTITION BY clause

```
SELECT
train_id,
station,
time,
LEAD(time,1) OVER (PARTITION BY train_id ORDER BY time) AS time_next
FROM sched
```

Result of adding PARTITION BY clause

```
+--------+-------------+-----+---------+
|train_id|      station| time|time_next|
+--------+-------------+-----+---------+
|     217|       Gilroy|6:06 |    6:15 |
|     217|   San Martin|6:15 |    6:21 |
|     217|  Morgan Hill|6:21 |    6:36 |
|     217| Blossom Hill|6:36 |    6:42 |
|     217|      Capitol|6:42 |    6:50 |
|     217|       Tamien|6:50 |    6:59 |
|     217|     San Jose|6:59 |    null |
|     324|San Francisco|7:59 |    8:03 |
|     324|   22nd Street|8:03 |    8:16 |
|     324|     Millbrae|8:16 |    8:24 |
|     324|    Hillsdale|8:24 |    8:31 |
|     324| Redwood City|8:31 |    8:37 |
|     324|    Palo Alto|8:37 |    9:05 |
|     324|     San Jose|9:05 |    null |
+--------+-------------+-----+---------+
```

train_id	station	time	time_to_next_stop
324	San Francisco	7:59	4 min
324	22nd Street	8:03	13 min
324	Millbrae	8:16	8 min
324	Hillsdale	8:24	7 min
324	Redwood City	8:31	6 min
324	Palo Alto	8:37	28 min
324	San Jose	9:05	null

Let's practice

Dot notation and SQL

INTRODUCTION TO SPARK SQL IN PYTHON

Mark Plutowski
Data Scientist

Our table has 3 columns

```
df.columns
```

```
['train_id', 'station', 'time']
```

```
df.show(5)
```

```
+--------+-------------+-----+
|train_id|      station| time|
+--------+-------------+-----+
|     324|San Francisco|7:59 |
|     324|  22nd Street|8:03 |
|     324|     Millbrae|8:16 |
|     324|    Hillsdale|8:24 |
|     324| Redwood City|8:31 |
+--------+-------------+-----+
```

We only need 2

```
df.select('train_id','station')
  .show(5)
```

```
+--------+-------------+
|train_id|      station|
+--------+-------------+
|     324|San Francisco|
|     324|  22nd Street|
|     324|     Millbrae|
|     324|    Hillsdale|
|     324| Redwood City|
+--------+-------------+
```

Three ways to select 2 columns

- `df.select('train_id', 'station')`

- `df.select(df.train_id, df.station)`

- `from pyspark.sql.functions import col`

- `df.select(col('train_id'), col('station'))`

Two ways to rename a column

```
df.select('train_id','station')
    .withColumnRenamed('train_id','train')
    .show(5)
```

```
+-----+------------+
|train|     station|
+-----+------------+
|  324|San Francisco|
|  324|  22nd Street|
|  324|     Millbrae|
|  324|    Hillsdale|
|  324| Redwood City|
+-----+------------+
```

```
df.select(col('train_id').alias('train'), 'station')
```

Don't do this!

```
df.select('train_id', df.station, col('time'))
```

SQL queries using dot notation

```
spark.sql('SELECT train_id AS train, station FROM schedule LIMIT 5')
      .show()
```

```
+-----+------------+
|train|     station|
+-----+------------+
|  324|San Francisco|
|  324|  22nd Street|
|  324|     Millbrae|
|  324|    Hillsdale|
|  324| Redwood City|
+-----+------------+
```

```
df.select(col('train_id').alias('train'), 'station')
  .limit(5)
  .show()
```

Window function SQL

```python
query = """
SELECT *,
ROW_NUMBER() OVER(PARTITION BY train_id ORDER BY time) AS id
FROM schedule
"""

spark.sql(query)
    .show(11)
```

Window function SQL

```
+-------+-----------+-----+---+
|train_id|    station| time| id|
+-------+-----------+-----+---+
|    217|     Gilroy|6:06 |  1|
|    217| San Martin|6:15 |  2|
|    217|Morgan Hill|6:21 |  3|
|    217|Blossom Hill|6:36 |  4|
|    217|    Capitol|6:42 |  5|
|    217|     Tamien|6:50 |  6|
|    217|   San Jose|6:59 |  7|
|    324|San Francisco|7:59 |  1|
|    324| 22nd Street|8:03 |  2|
|    324|   Millbrae|8:16 |  3|
|    324|  Hillsdale|8:24 |  4|
+-------+-----------+-----+---+
```

Window function using dot notation

```python
from pyspark.sql import Window,
from pyspark.sql.functions import row_number
df.withColumn("id", row_number()
                .over(
                    Window.partitionBy('train_id')
                        .orderBy('time')
                )
    )
```

- ROW_NUMBER in SQL : `pyspark.sql.functions.row_number`

- The inside of the OVER clause : `pyspark.sql.Window`

- PARTITION BY : `pyspark.sql.Window.partitionBy`

- ORDER BY : `pyspark.sql.Window.orderBy`

Using a WindowSpec

- The `over` function in Spark SQL corresponds to a `OVER` clause in SQL.

- The class `pyspark.sql.window.Window` represents the inside of an `OVER` clause.

```
window = Window.partitionBy('train_id').orderBy('time')
dfx = df.withColumn('next', lead('time',1).over(window))
```

- Above, type(window) is pyspark.sql.window.WindowSpec

Let's practice

INTRODUCTION TO SPARK SQL IN PYTHON