NC STATE UNIVERSITY
DEPARTMENT OF
STATISTICS

Streaming Joins

Justin Post

Streaming: Joins

Justin Post

Recap

- Create a spark session

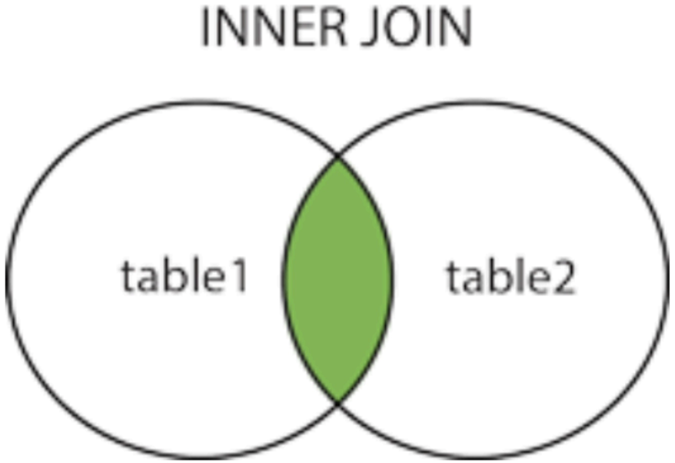
1. Read in a stream
    - Stream from a file, terminal, or use something like kafka
2. Set up transformations/aggregations to do (mostly using SQL type functions)
    - Perhaps over windows
    - Use a watermark to allow for late data
3. Set up writing of the query to an output source
    - Console (for debugging)
    - File (say .csv)
    - Database
4. `query.start()` the query!
    - Continues listening until terminated (`query.stop()`)

Can combine two streams or a stream and a static data source

Recall Our Common Joins

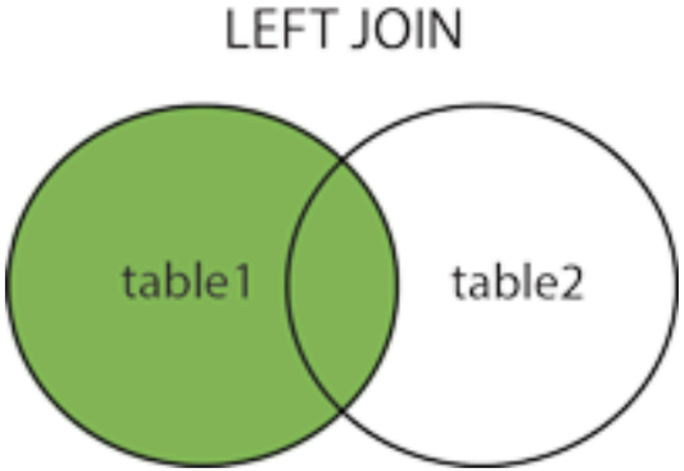
Combining two (or more) tables in SQL is called doing a **join**

- Inner Join: Returns records with matching keys in both tables



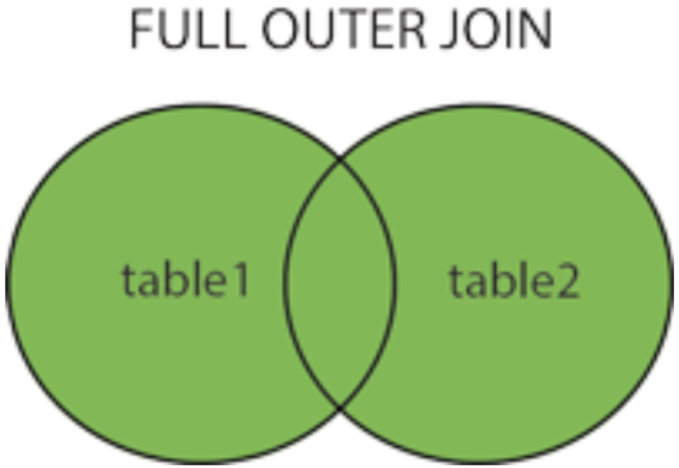
Recall Our Common Joins

- Left (Outer) Join: Returns all records from the 'left' table and any matching records from the 'right' table



Recall Our Common Joins

- (Full) Outer Join: Returns all records when there is a match from the 'left' or 'right' table



Common Joins

- Can do the following **stream to stream** joins in Spark Structured Streaming:
    - Inner
    - Left (must specify watermark on right and time constraints)
        - Right works similarly
    - Full Outer (must specify watermark and time constraints on at least one side)
    - Left Semi (return any rows from the left dataset that were matched with the right table)

Common Joins

- Can do the following **stream to stream** joins in Spark Structured Streaming:
  - Inner
  - Left (must specify watermark on right and time constraints)
    - Right works similarly
  - Full Outer (must specify watermark and time constraints on at least one side)
  - Left Semi (return any rows from the left dataset that were matched with the right table)
- Must use **append** output mode
- Cannot do aggregations before joins

Example Inner Join Syntax

- Suppose you have `streamDF1` and `streamDF2`

```
streamDF1.join(streamDF2, "col_id")    # inner join on common column col_id
```

Example Left (Outer) Join Syntax

- Suppose you have `streamDF1` and `streamDF2`
- Each has some watermarks (code modified from here)

```
# Define watermarks
impressionsWithWatermark = impressions \
  .selectExpr("adId AS impressionAdId", "impressionTime") \
  .withWatermark("impressionTime", "10 seconds ")   # max 10 seconds late

clicksWithWatermark = clicks \
  .selectExpr("adId AS clickAdId", "clickTime") \
  .withWatermark("clickTime", "20 seconds")        # max 20 seconds late
```

Example Left (Outer) Join Syntax

- Suppose you have `streamDF1` and `streamDF2`
- Each has some watermarks (code modified from here)

```
from pyspark.sql.functions import expr

# Left outer join with time range conditions
impressionsWithWatermark.join(
  clicksWithWatermark,
  expr("""
    clickAdId = impressionAdId AND
    clickTime >= impressionTime AND
    clickTime <= impressionTime + interval 1 hour
    """),
  "leftOuter"
)
```

Stream-static Joins

We can also do joins of a stream with a static spark Data Frame

- Suppose streaming DF is on the left and static on the right
    - Inner, left outer, and left semi are supported

Stream-static Joins

We can also do joins of a stream with a static spark Data Frame

- Suppose streaming DF is on the left and static on the right
    - Inner, left outer, and left semi are supported

Example syntax:

```
streamingDF.join(staticDF, "column", "inner")
```

- For all of these, we then need to write the query!

Example

- Let's jump into pyspark and do a few joins!

Recap

- Can do some stream-to-stream joins and stream-to-static joins
- For stream to stream joins
  - Must use **append** output mode
  - Cannot do aggregations before joins

Course Recap

- 5 V's of Big Data
    - Volume
    - Variety
    - Velocity
    - Veracity (Variability)
    - Value
- Understanding of the Big Data pipeline and basics of handling Big Data
    - Databases/Data Lakes/Data Warehouses/etc.
    - Hadoop
    - Spark
- Modeling data
    - Machine learning algorithms
    - Tuning and testing models
- Common issues seen on data with velocity and Spark Structured Streaming