Structured Streaming

Learning Objectives

- Process streaming data
- DataStreamReader
- DataStreamWriter

Data Stream

- Any data source that grows over time

- New files landing in cloud storage
- Updates to a database captured in a CDC feed
- Events queued in a pub/sub messaging feed

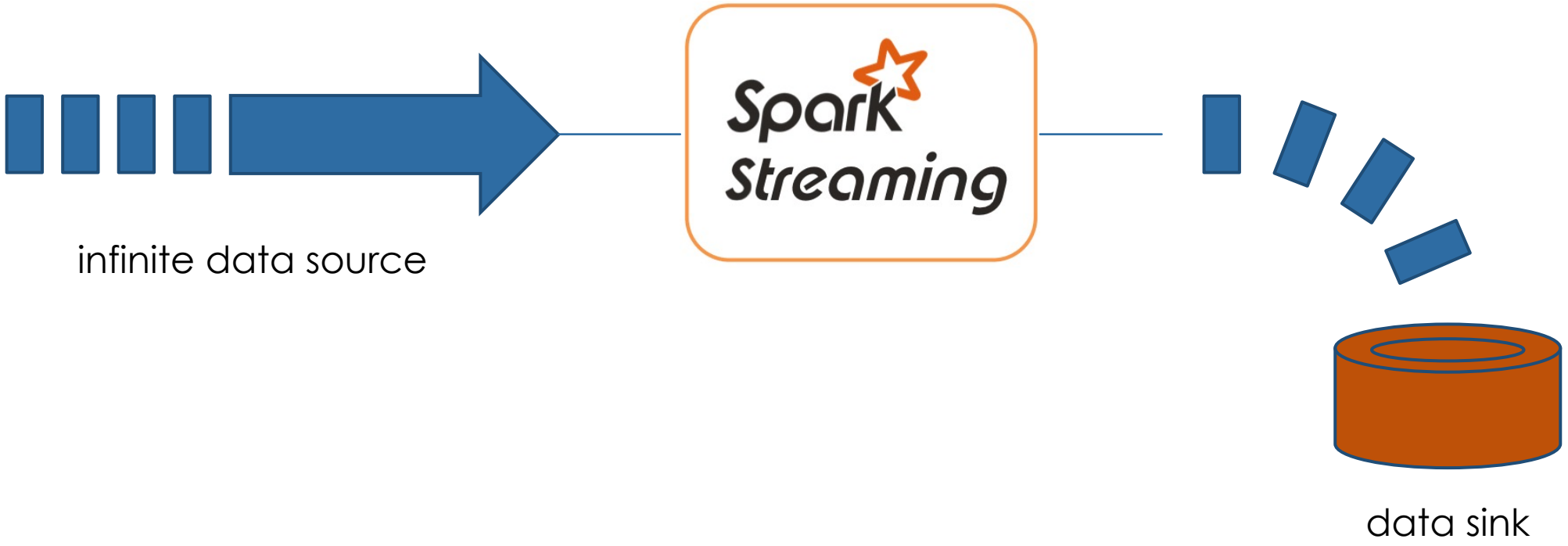
Processing Data Stream

- 2 approaches:

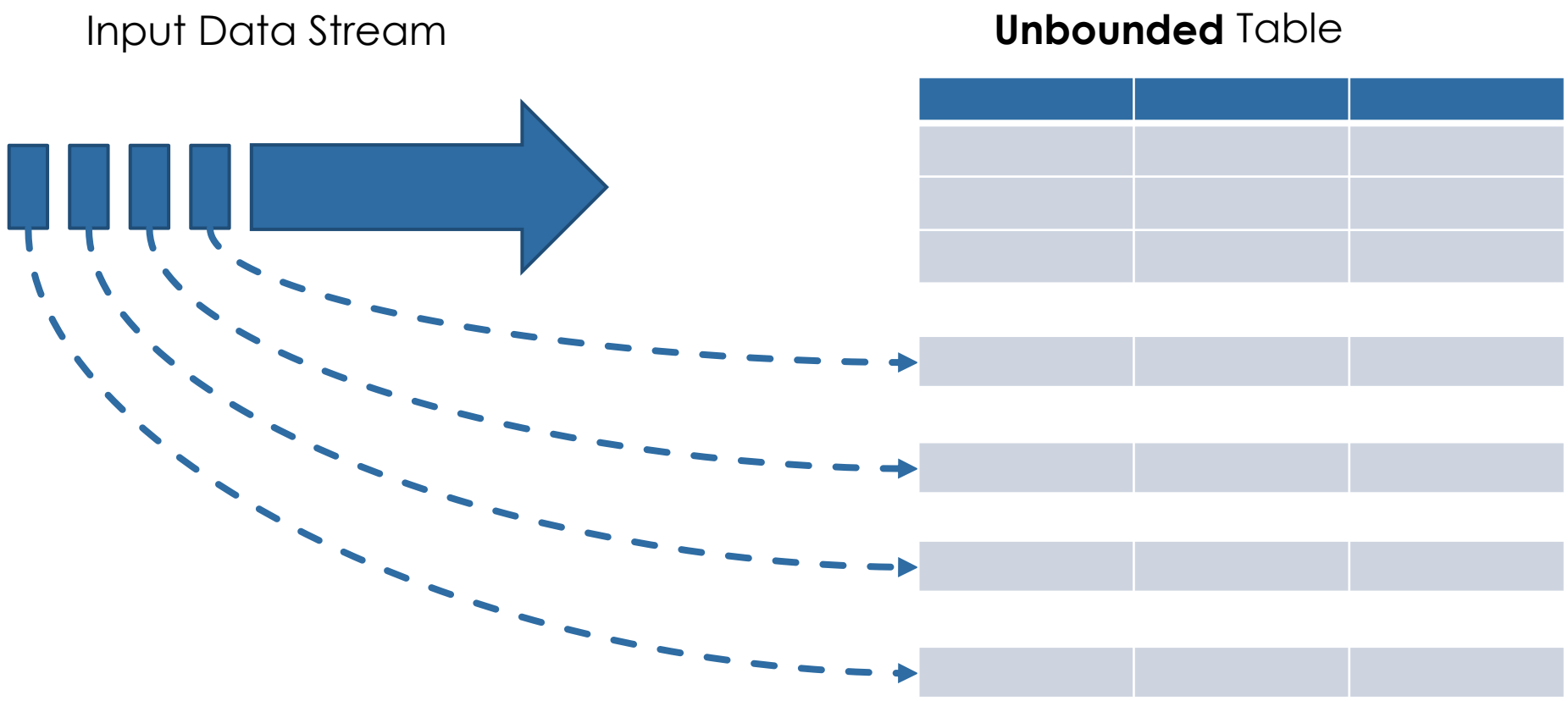
1. Reprocess the entire source dataset each time

2. Only process those new data added since last update
   - Structured Streaming

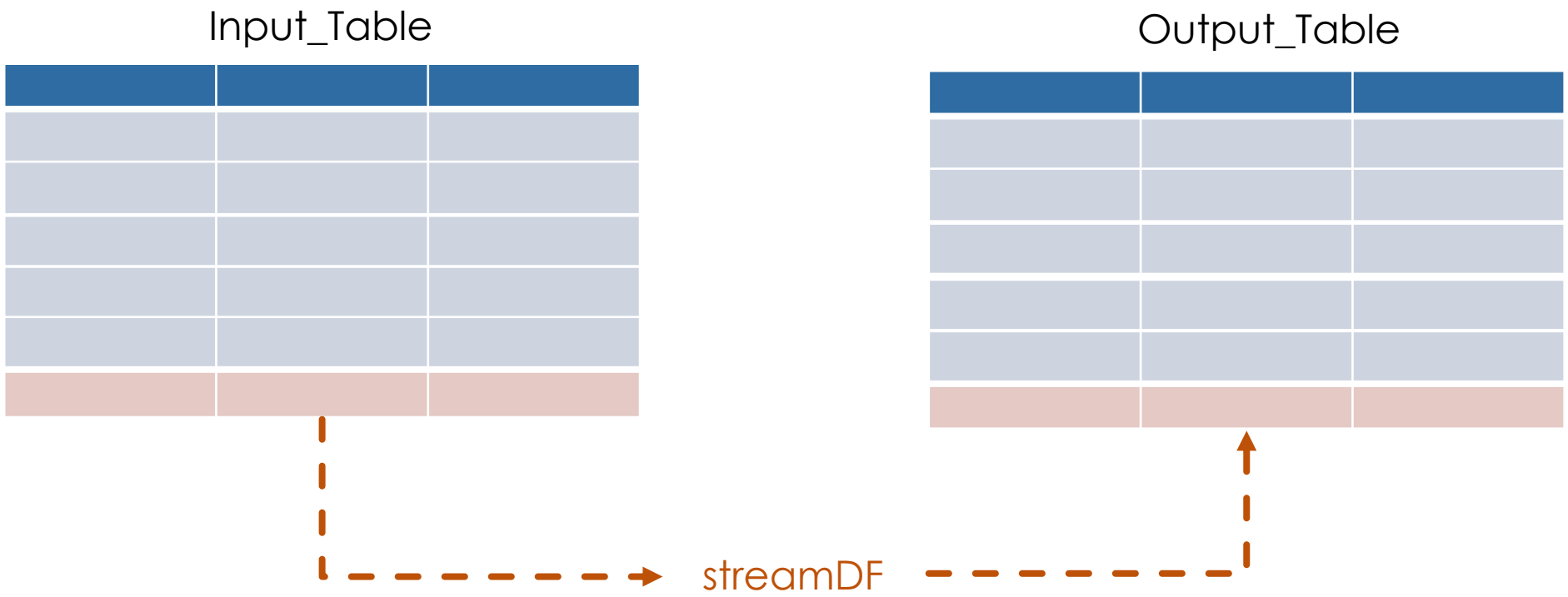
Spark Structured Streaming



Treating Infinite Data as a Table



Input Streaming Table




```
streamDF = spark.readStream
                .table("Input_Table")
```

```
streamDF.writeStream
        .trigger(processingTime="2 minutes")
        .outputMode("append")
        .option("checkpointLocation", "/path")
        .table("Output_Table")
```

Trigger Intervals

```
streamDF.writeStream
        .trigger(processingTime="2 minutes")
        .outputMode("append")
        .option("checkpointLocation", "/path")
        .table("Output_Table")
```

Trigger	Method call	Behavior
Unspecified		Default: processingTime="500ms"
Fixed interval	`.trigger(processingTime="5 minutes")`	Process data in micro-batches at the user-specified intervals
Triggered batch	`.trigger(once=True)`	Process all available data in a single batch, then stop
Triggered micro-batches	`.trigger(availableNow=True)`	Process all available data in multiple micro-batches, then stop

Output Modes

```
streamDF.writeStream
        .trigger(processingTime="2 minutes")
        .outputMode("append")
        .option("checkpointLocation", "/path")
        .table("Output_Table")
```

Mode	Method call	Behavior
Append (Default)	`.outputMode("append")`	Only newly appended rows are incrementally appended to the target table with each batch
Complete	`.outputMode("complete")`	The target table is overwritten with each batch

Checkpointing

```
streamDF.writeStream
        .trigger(processingTime="2 minutes")
        .outputMode("append")
        .option("checkpointLocation", "/path")
        .table(”Output_Table")
```

- Store stream state
- Track the progress of your stream processing

- Can **Not** be shared between separate streams

Guarantees

1. Fault Tolerance
   - Checkpointing + Write-ahead logs
     - record the offset range of data being processed during each trigger interval.
2. Exactly-once guarantee
   - Idempotent sinks

Unsupported Operations

- Some operations are not supported by streaming DataFrame
  - Sorting
  - Deduplication
- Advanced methods
  - Windowing
  - Watermarking