```
query = df.writeStream \
    .trigger(processingTime="10sec")
```

→ Checkpointing Interval

① Global aggregation

```
grouped_data = df.groupby('Country').agg(sum(Sales))
```

Country, Sales

IND, 50

USA, 20
IND, 30

States after 1st sec:

```
{
  "IND": 50
}
```

State after Batch 2 ← Prev Computed

```
{
  "IND": 70,
  "USA": 30
}
```

② Window Based

```
window_grouped = df.groupBy(timestamp_col, window_size)
```

IND, 30 } 10:00:00
USA, 20 } 10:00:15
IND, 40 } 10:00:24
USA, 20] 10:00:35

Window1 = 10:00:00 to 10:00:30
State after first Batch

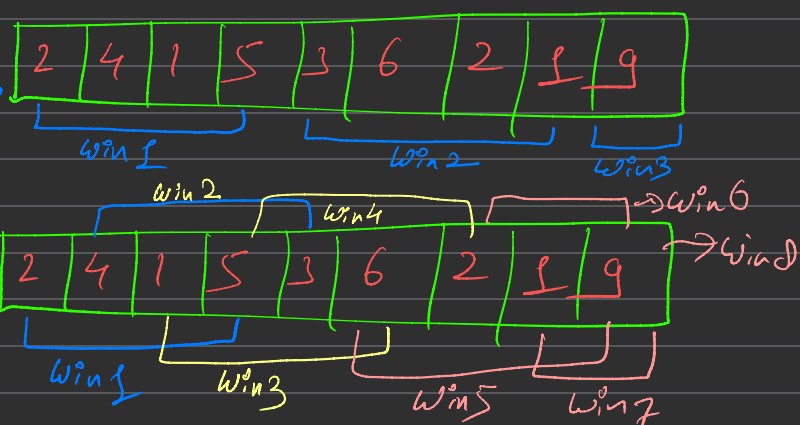
Window	Country	Value
00:00-00:30	IND	40
00:00-00:30	USA	20

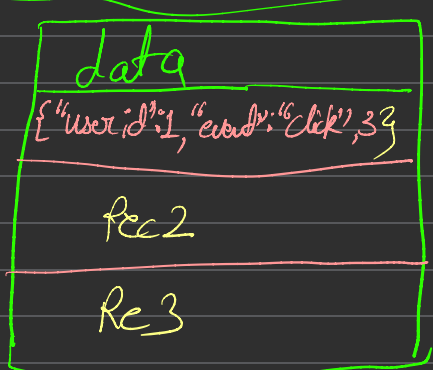
State after Batch 2

Window 2 ⇒ 00:30 to 01:00

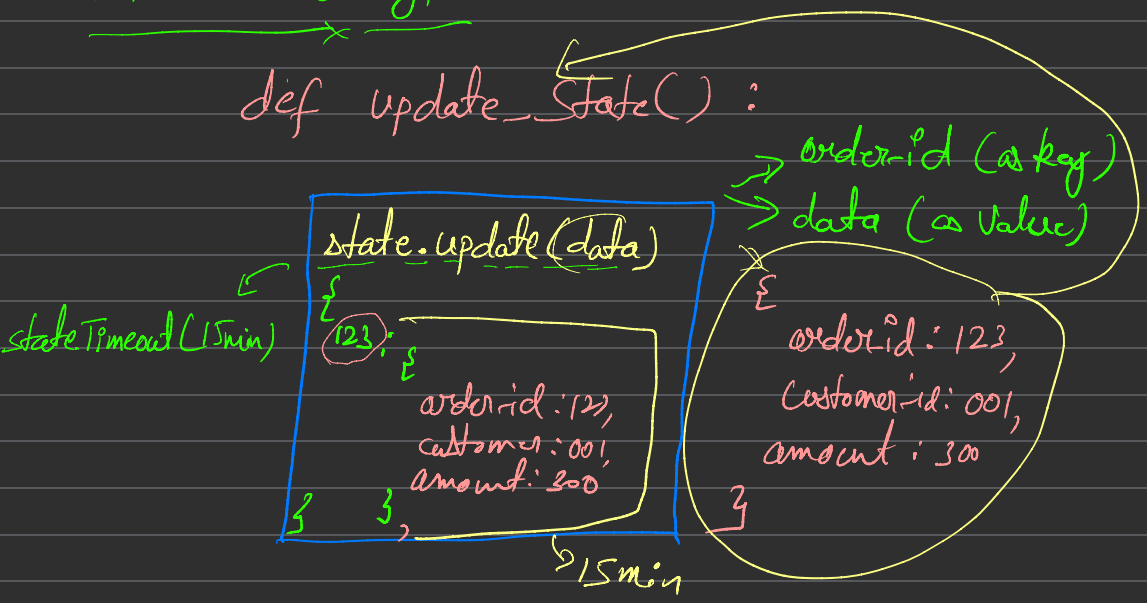
Window	Country	Value
00:00 - 00:30	IND	40
00:00 - 00:30	USA	20
00:30 - 01:00	USA	20

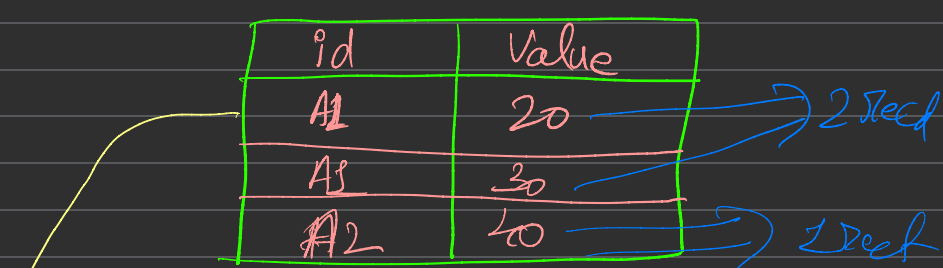
window of 4 element





stateStruct Type





update_state([A1], [(A1, 20)], state);

current_state = state.get if state.exists else

{ current_state = { sum: 0, count: 0 } }

{ sum: 0, count: 0 }

total_sum = current_state["sum"]

total_count = current_state["count"]

total_sum = 0

total_count = 0

[{"id": "A1", "Value": 20}]

pdf

for pdf in pdf_iter:

total_sum = total_sum + pdf["Value"].sum()

total_count = total_count + len(pdf)

id	Value
A1	20
A1	10

id	Value
A1	30

30 + 50

total_sum = 0 + 50

total_count = 0 + 2

[{"A1": [{"id": A1, "Value": 20}, {"id": A1, "Value": 10}]

},

{"A2": [{"id": A2, "Value": 30}]

}]

A1

Before Start

State = Empty

A1

Current_state = {

Sum: 0,

Count: 0

}

A1

After Break

current_state {

sum: 50

count: 2

}

update_state

state.update(new)

A1

state = {

sum: 50,

count = 2

}