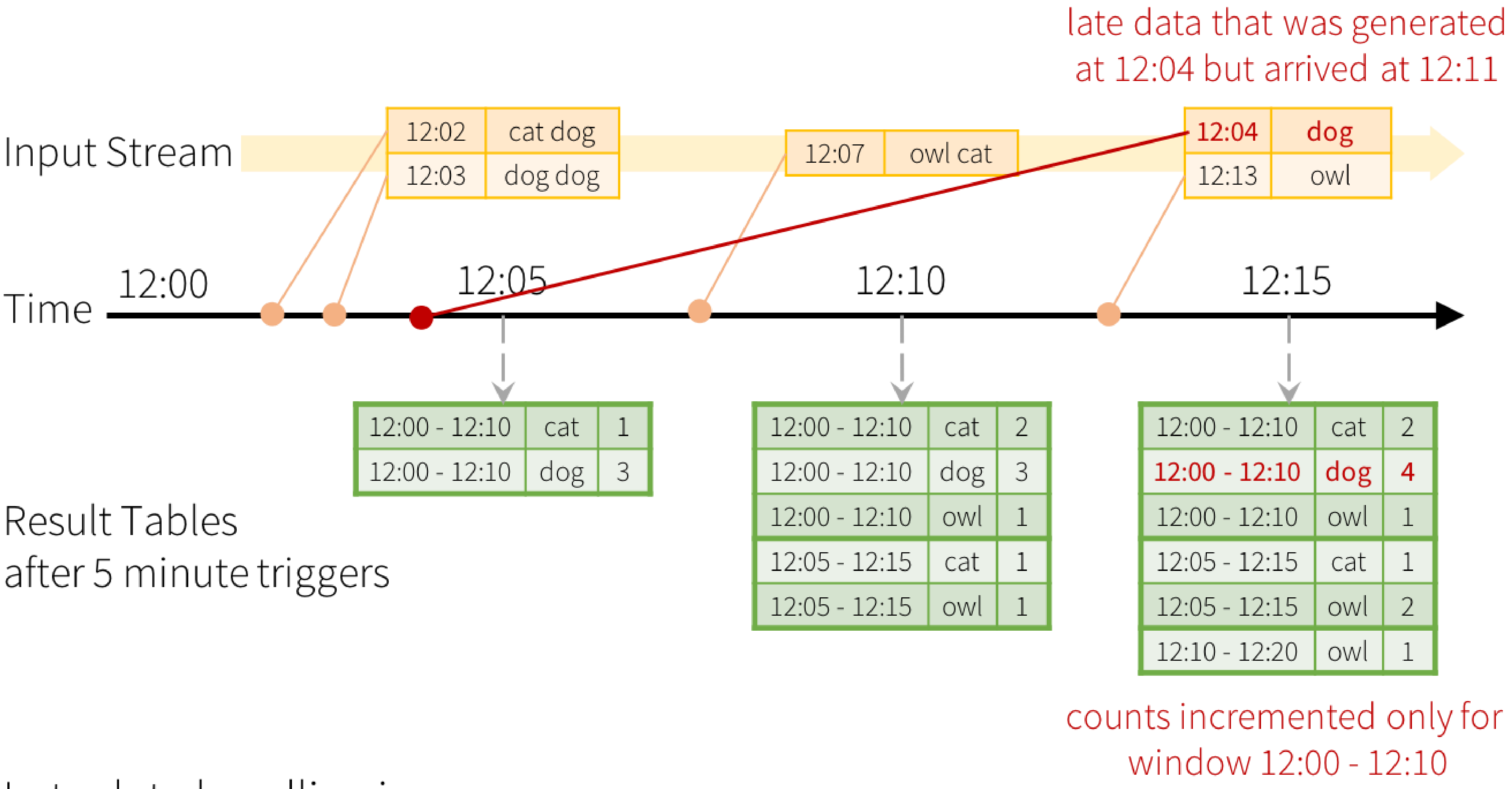
Handling Late Data and Watermarking



12:02	cat dog
12:03	dog dog

12:07	owl cat

12:04	dog
12:13	owl

Result Tables
after 5 minute triggers

12:00 - 12:10	cat	1
12:00 - 12:10	dog	3

12:00 - 12:10	cat	2
12:00 - 12:10	dog	3
12:00 - 12:10	owl	1
12:05 - 12:15	cat	1
12:05 - 12:15	owl	1

12:00 - 12:10	cat	2
12:00 - 12:10	dog	4
12:00 - 12:10	owl	1
12:05 - 12:15	cat	1
12:05 - 12:15	owl	2
12:10 - 12:20	owl	1

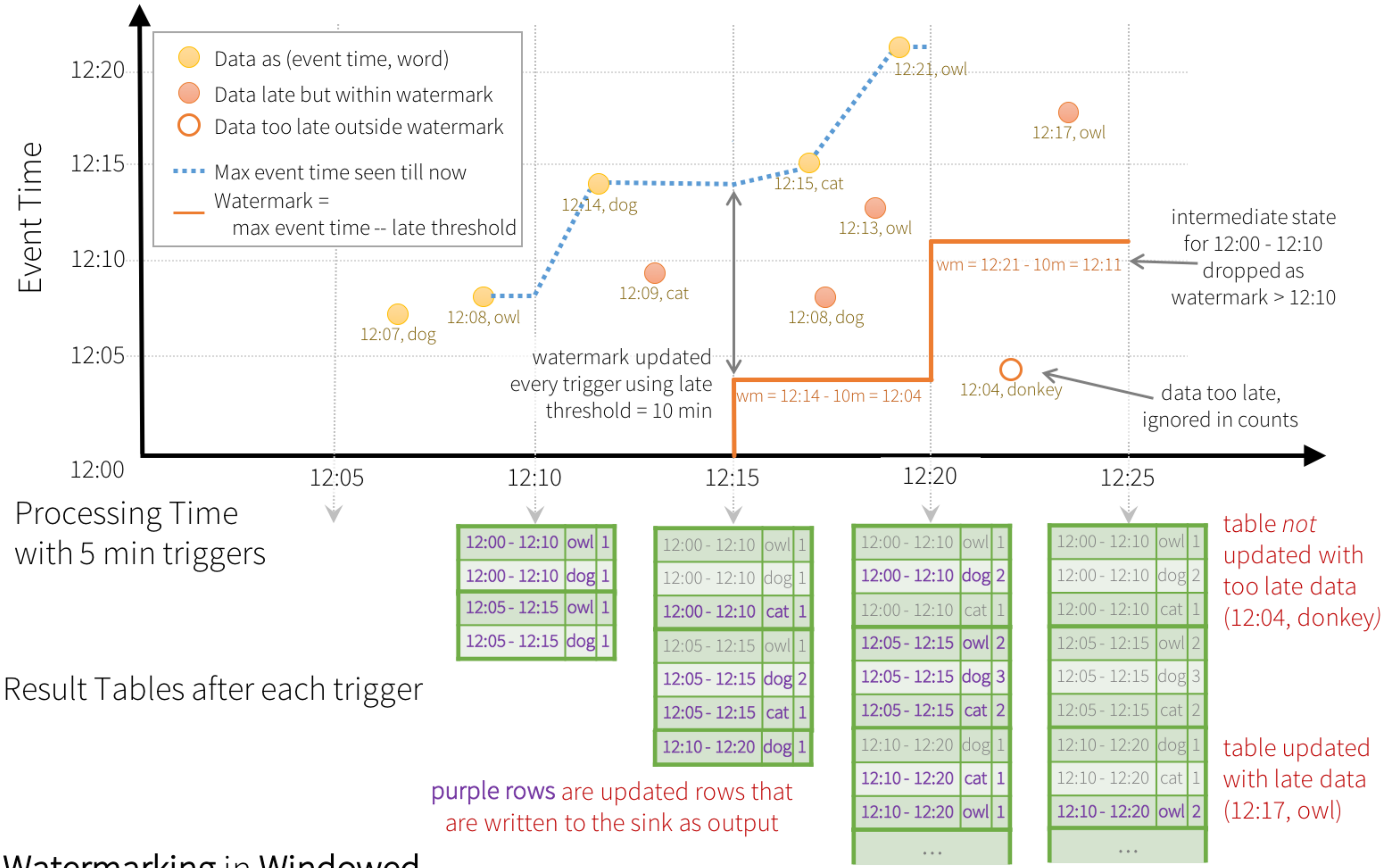
counts incremented only for window 12:00 - 12:10

Late data handling in Windowed Grouped Aggregation

Watermarking example code

```
words = ...  # streaming DataFrame of schema { timestamp: Timestamp, word: String }

# Group the data by window and word and compute the count of each group
windowedCounts = words \
    .withWatermark("timestamp", "10 minutes") \
    .groupBy(
        window(words.timestamp, "10 minutes", "5 minutes"),
        words.word) \
    .count()
```



watermark updated every trigger using late threshold = 10 min

intermediate state for 12:00 - 12:10 dropped as watermark > 12:10

data too late, ignored in counts

Processing Time with 5 min triggers

Result Tables after each trigger

12:00 - 12:10	owl	1
12:00 - 12:10	dog	1
12:05 - 12:15	owl	1
12:05 - 12:15	dog	1

12:00 - 12:10	owl	1
12:00 - 12:10	dog	1
12:00 - 12:10	cat	1
12:05 - 12:15	owl	1
12:05 - 12:15	dog	2
12:05 - 12:15	cat	1
12:10 - 12:20	dog	1

12:00 - 12:10	owl	1
12:00 - 12:10	dog	2
12:00 - 12:10	cat	1
12:05 - 12:15	owl	2
12:05 - 12:15	dog	3
12:05 - 12:15	cat	2
12:10 - 12:20	dog	1
12:10 - 12:20	cat	1
12:10 - 12:20	owl	1
...		

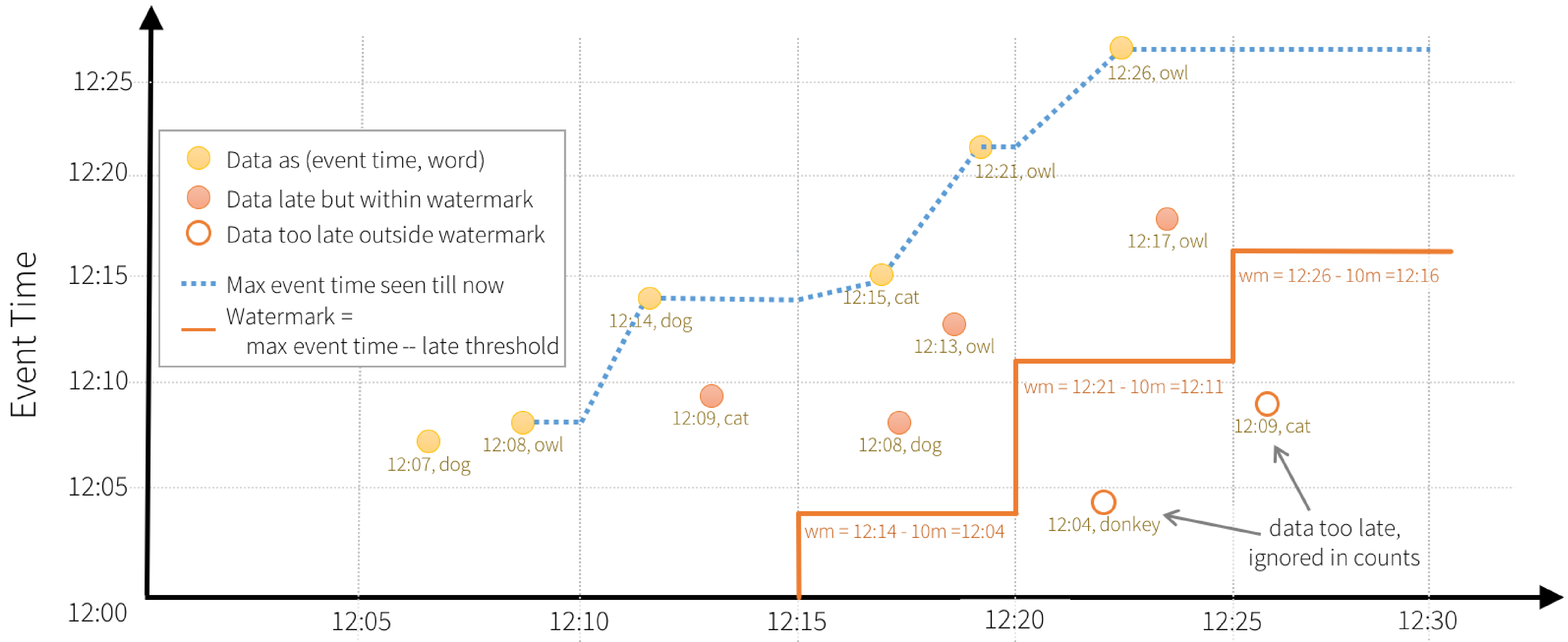
12:00 - 12:10	owl	1
12:00 - 12:10	dog	2
12:00 - 12:10	cat	1
12:05 - 12:15	owl	2
12:05 - 12:15	dog	3
12:05 - 12:15	cat	2
12:10 - 12:20	dog	1
12:10 - 12:20	cat	1
12:10 - 12:20	owl	2
...		

table *not* updated with too late data (12:04, donkey)

table updated with late data (12:17, owl)

purple rows are updated rows that are written to the sink as output

Watermarking in **Windowed Grouped Aggregation** with **Update Mode**



Processing Time
with 5 min triggers

partial counts for window 12:00 - 12:10 maintained as internal state while waiting for late data, so not yet added to result table

12:00 - 12:10	owl	1
12:00 - 12:10	cat	1
12:00 - 12:10	dog	2

12:00 - 12:10	owl	1
12:00 - 12:10	cat	1
12:00 - 12:10	dog	2
12:05 - 12:15	owl	2
12:05 - 12:15	cat	2
12:05 - 12:15	dog	3

final counts for 12:00 - 12:10 added to table when watermark > 12:10, late data counted, and intermediate state for window dropped

Watermarking in **Windowed Grouped Aggregation** with **Append Mode**

Result Tables after each trigger

Conditions for Watermarking

- Output mode must be set to *Append* or *Update*.
    - If it is set to *Complete* mode, it will requires all aggregate data to be preserved and will be incapable of using watermarking.
- Aggregation needs either an event-time column, or a window on the event-time column.
    - withWatermark must be called on the same column as the timestamp column used in the aggregate, `withWatermark` must be called before the aggregation for the watermark details to be used.