
snowflake®
SUMMIT 2022
LIVE IN VEGAS
LAS VEGAS

OS303 FUNNEL ANALYSIS IN SQL
FROM START TO FINISH

TJ Murphy (@teej_m) | Head of Data, Multi Media LLC

TJ Murphy

Head of Data

MULTI MEDIA LLC

PART ONE

JOIN

PART TWO

WINDOW

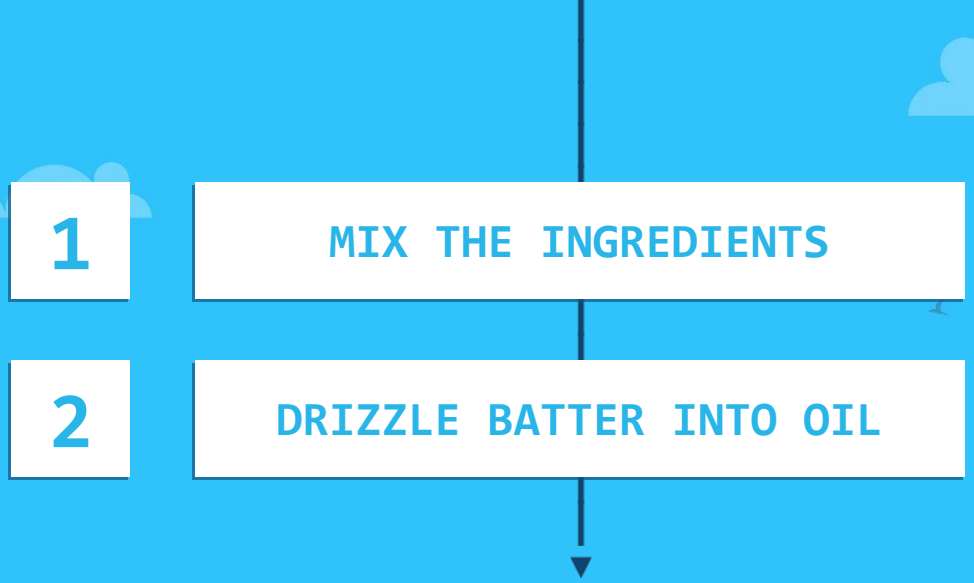
PART THREE

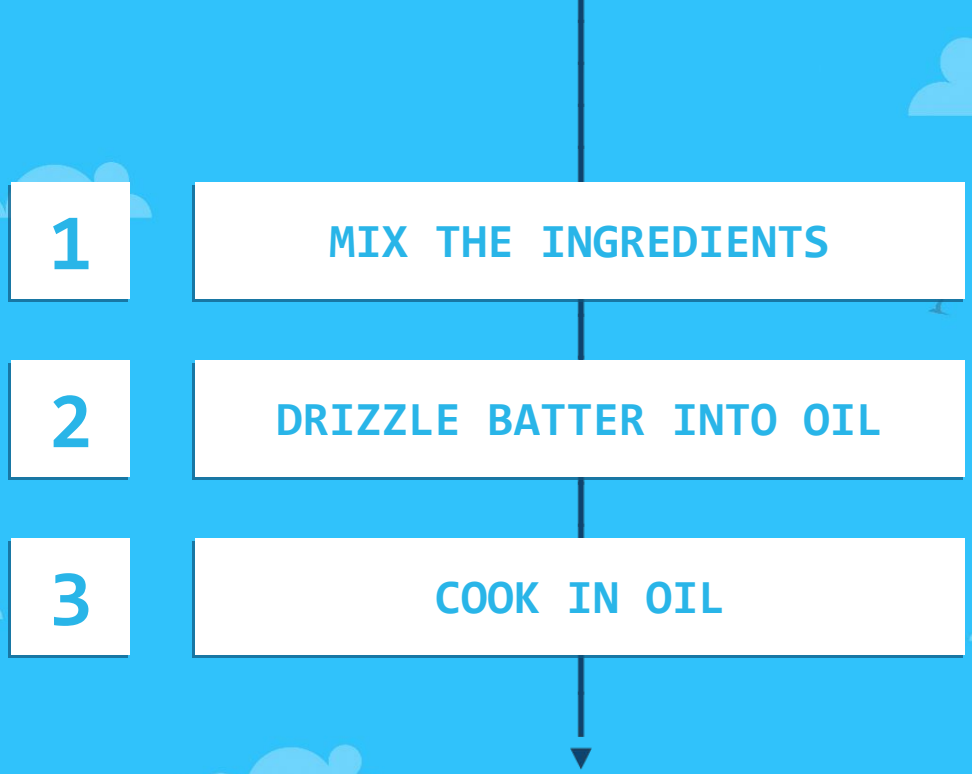
REGEX


Funnel
Cake

1

MIX THE INGREDIENTS





FUNNEL CAKE


PRELUDE
FUNNELS

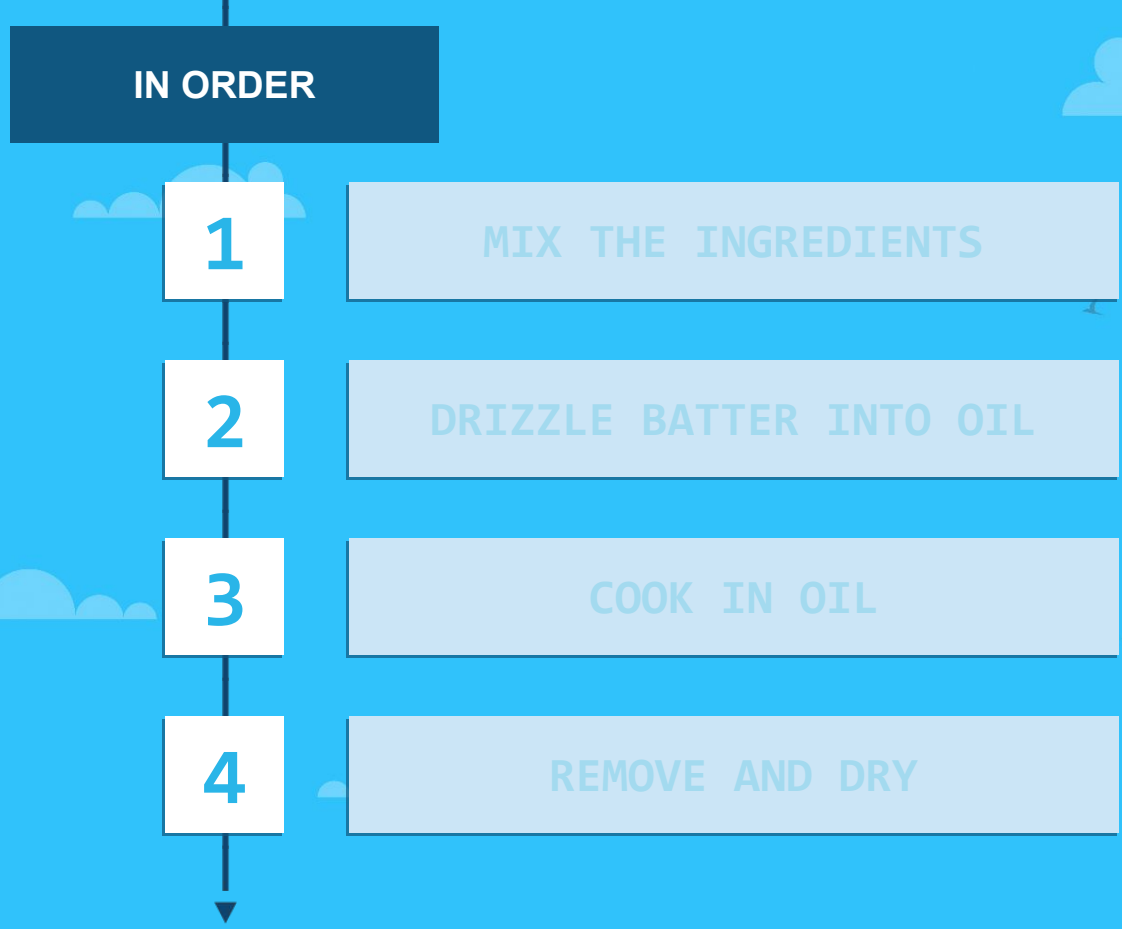
A SET OF STEPS

MIX THE INGREDIENTS

DRIZZLE BATTER INTO OIL

COOK IN OIL

REMOVE AND DRY



UNDER TIME CONSTRAINTS

5 MINUTES

A SET OF STEPS

IN ORDER

UNDER TIME CONSTRAINTS

- DNA SEQUENCING
- MARKETING ATTRIBUTION
- A/B TESTING
- COHORT ANALYSIS
- USER PATHING
- AGGRESSIVE DRIVING DETECTION
- STATE-TRANSITION GRAPHS
- RANDOMIZED CONTROLLED TRIALS
- TIMED AUTOMATA
- MODEL CHECKING
- COMPLEX EVENT PROCESSING

MIX › DRIZZLE › COOK › REMOVE

A›B›C›D

JOIN

WINDOW

REGEX


PART ONE
THE JOIN

github.com/teej/sf-funnels

github.com/teej/sf-funnels

```
SELECT COUNT(DISTINCT step_A.user_id) AS users_A
     , COUNT(DISTINCT step_B.user_id) AS users_B

  FROM events AS step_A

  LEFT JOIN events AS step_B
    ON step_B.event_name = 'B'

   AND step_B.user_id = step_A.user_id
   AND step_B.event_at > step_A.event_at
   AND step_B.event_at < step_A.event_at + INTERVAL '30 days'

 WHERE step_A.event_name = 'A'
```

github.com/teej/sf-funnels

```
SELECT COUNT(DISTINCT step_A.user_id) AS users_A
     , COUNT(DISTINCT step_B.user_id) AS users_B

  FROM events AS step_A

  LEFT JOIN events AS step_B
    ON step_B.event_name = 'B'

   AND step_B.user_id = step_A.user_id
   AND step_B.event_at > step_A.event_at
   AND step_B.event_at < step_A.event_at + INTERVAL '30 days'

 WHERE step_A.event_name = 'A'
```

DEFINE STEPS

github.com/teej/sf-funnels

```
SELECT COUNT(DISTINCT step_A.user_id) AS users_A
     , COUNT(DISTINCT step_B.user_id) AS users_B

  FROM events AS step_A

  LEFT JOIN events AS
    ON step_B.event_

   AND step_B.user_id = step_A.user_id
   AND step_B.event_at > step_A.event_at
   AND step_B.event_at < step_A.event_at + INTERVAL '30 days'

 WHERE step_A.event_name = 'A'
```

IN ORDER

github.com/teej/sf-funnels

```
SELECT COUNT(DISTINCT step_A.user_id) AS users_A
     , COUNT(DISTINCT step_B.user_id) AS users_B

  FROM events AS step_A

  LEFT JOIN events AS step_B
    ON step_B.event_name = 'B'

   AND step_B.user_id = step_A.user_id
   AND step_B.event_at > step_A.event_at
   AND step_B.event_at < step_A.event_at + INTERVAL '30 days'

 WHERE step_A.event_name = 'A'
```

UNDER TIME CONSTRAINTS

A>B

	SCALE FACTOR	SPEED	PERFORMANCE
JOIN	50	1.1s	
	100	2.3s	
	200	3.4s	
	400	7.5s	
	800	13 s	
	1,600	33 s	

A›B›C

	SCALE FACTOR	SPEED	PERFORMANCE
JOIN	50	1.5s	
	100	3.4s	
	200	5.3s	
	400	11 s	
	800	19 s	
	1 600	35 s	

	PATTERN	PERFORMANCE
JOIN	A›B	
	A›B›C	
	A›B›C›D	
	A›B›C›D›E	

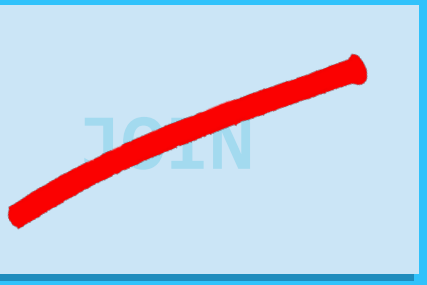
TAKEAWAYS

JOIN

Straightforward to implement

Easy to add constraints

Doesn't scale well



WINDOW

REGEX


PART TWO
WINDOWS

A>B

github.com/teej/sf-funnels

```
    WITH prep AS (...)
       , scan AS (
  SELECT user_id
       , is_step_A AS step_A_match
       , LAG(IFF(step_A_match, event_at, NULL))
           IGNORE NULLS
           OVER (PARTITION BY user_id ORDER BY event_at)
         AS prior_step_A_at
       , is_step_B
           AND event_at < prior_step_A_at + INTERVAL '30 days'
         AS step_B_match
    FROM prep
)
  SELECT COUNT(DISTINCT IFF(step_A_match, user_id, NULL)) AS users_A
       , COUNT(DISTINCT IFF(step_B_match, user_id, NULL)) AS users_B
    FROM scan
```

github.com/teej/sf-funnels

```
  WITH prep AS (

SELECT user_id
     , event_at
     , event_name = 'A' AS is_step_A
     , event_name = 'B' AS is_step_B
  FROM events
 WHERE event_name IN ('A', 'B')

)
     , scan AS (...)

SELECT ...
  FROM scan
```

DEFINE STEPS

github.com/teej/sf-funnels

```
    WITH prep AS (...)
       , scan AS (
  SELECT user_id
       , is_step_A AS step_A_match

       , LAG(IFF(step_A_match, event_at, NULL))
           IGNORE NULLS
           OVER (PARTITION BY user_id ORDER BY event_at)
         AS prior_step_A_at

       , is_step_B
           AND event_at < prior_step_A_at + INTERVAL '30 days'
         AS step_B_match
    FROM prep
)
  SELECT ...
    FROM scan
```

IN ORDER

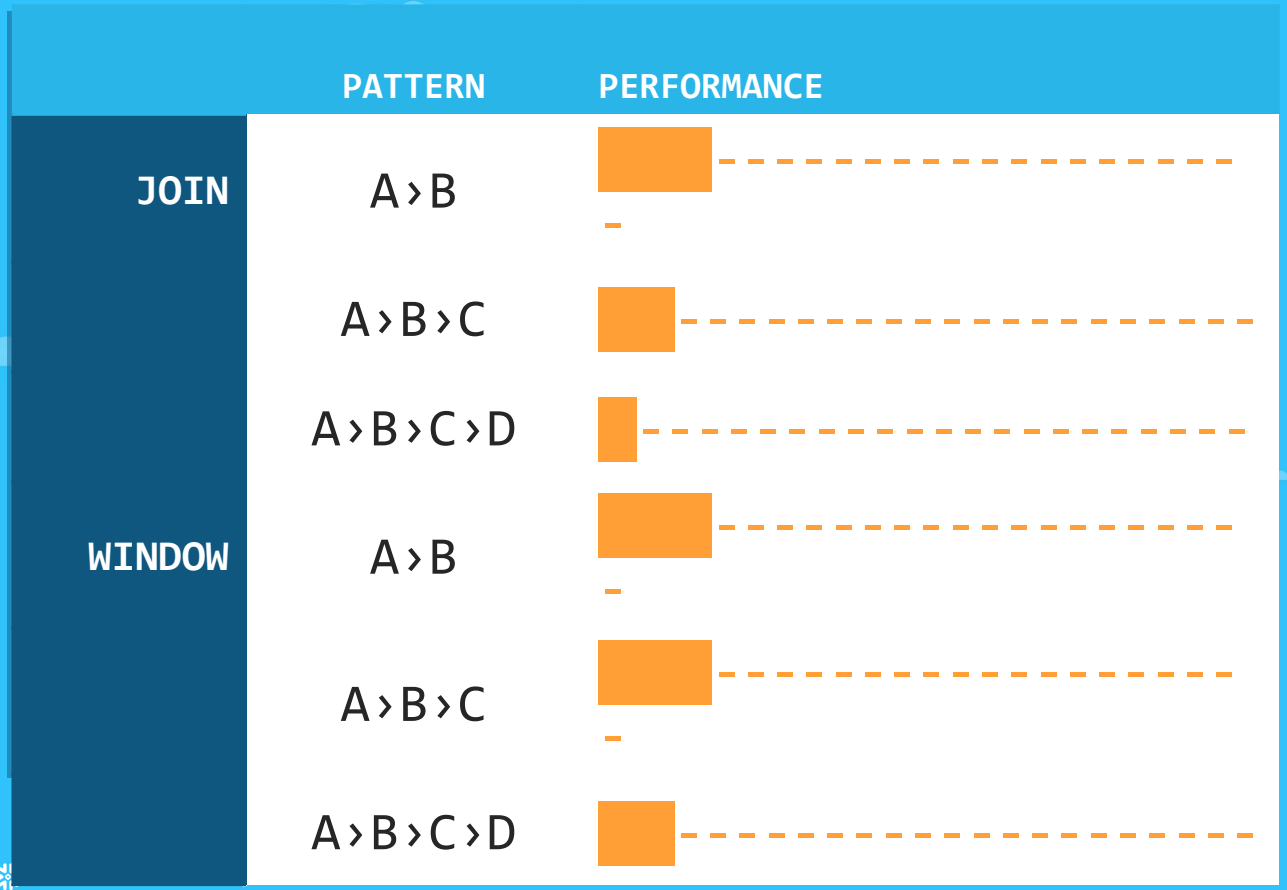
github.com/teej/sf-funnels

```
    WITH prep AS (...)
       , scan AS (
  SELECT user_id
       , is_step_A AS step_A_match

       , LAG(IFF(step_A_match, event_at, NULL))
           IGNORE NULLS
           OVER (PARTITION BY u                _at)
         AS prior_step_A_at

       , is_step_B
           AND event_at < prior_step_A_at + INTERVAL '30 days'
         AS step_B_match
    FROM prep
)
  SELECT ...
    FROM scan
```

UNDER TIME CONSTRAINTS


PATTERN
PERFORMANCE
JOIN
A›B
A›B›C
A›B›C›D
WINDOW
A›B
A›B›C
A›B›C›D

TAKEAWAYS

WINDOW

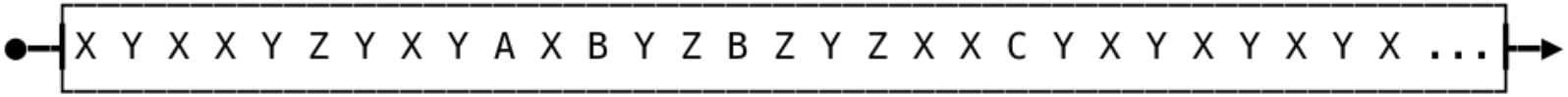
Performance beats JOINs

Scales to 1B+ rows

Unwieldy to write for complex funnels

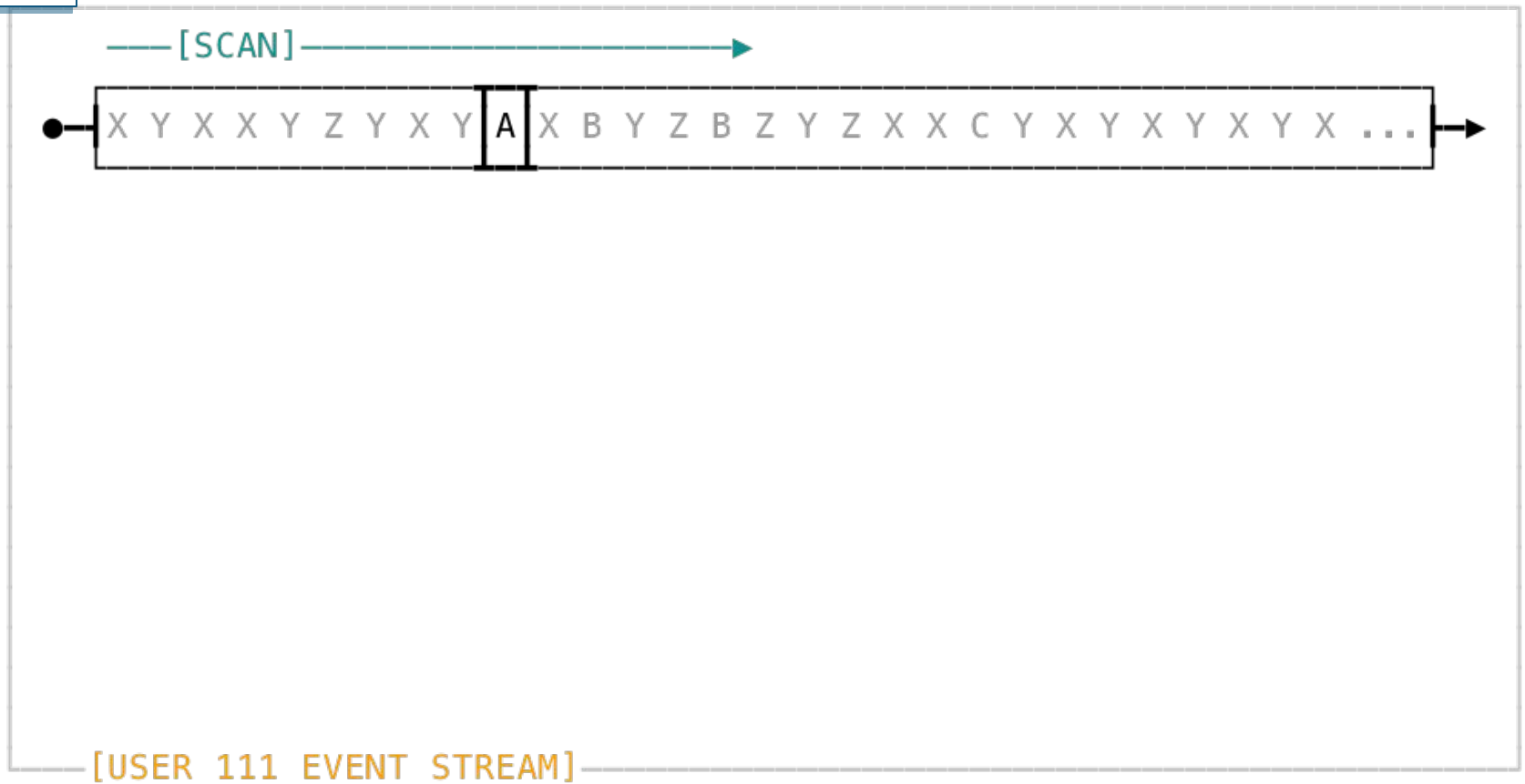
THE TROUBLE WITH SCANS

A›B



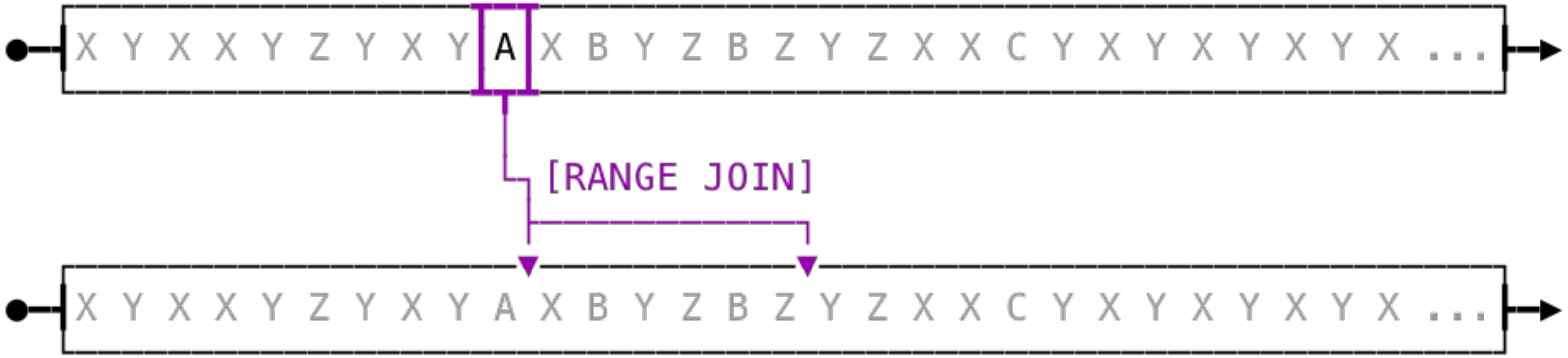
[USER 111 EVENT STREAM]

A›B



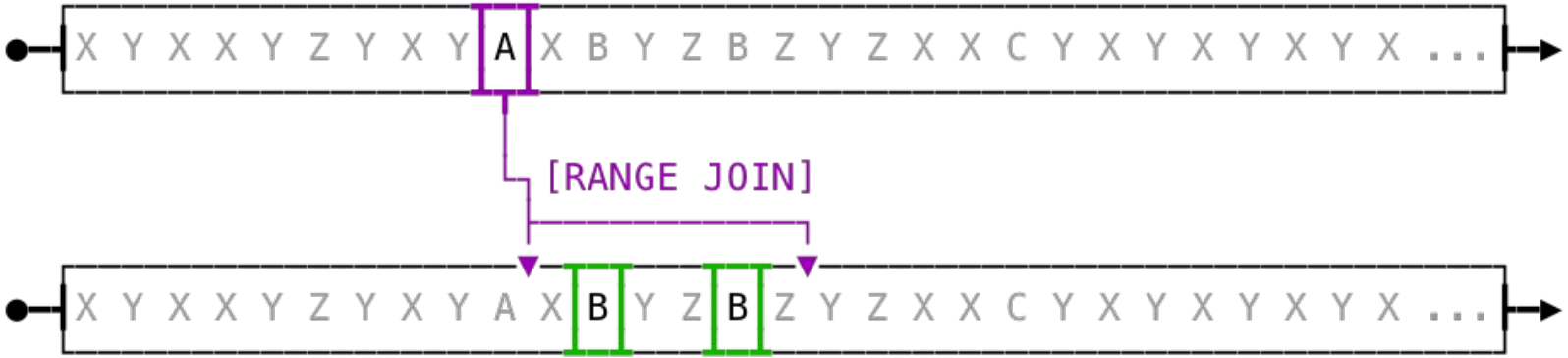
THE TROUBLE WITH SCANS

A>B



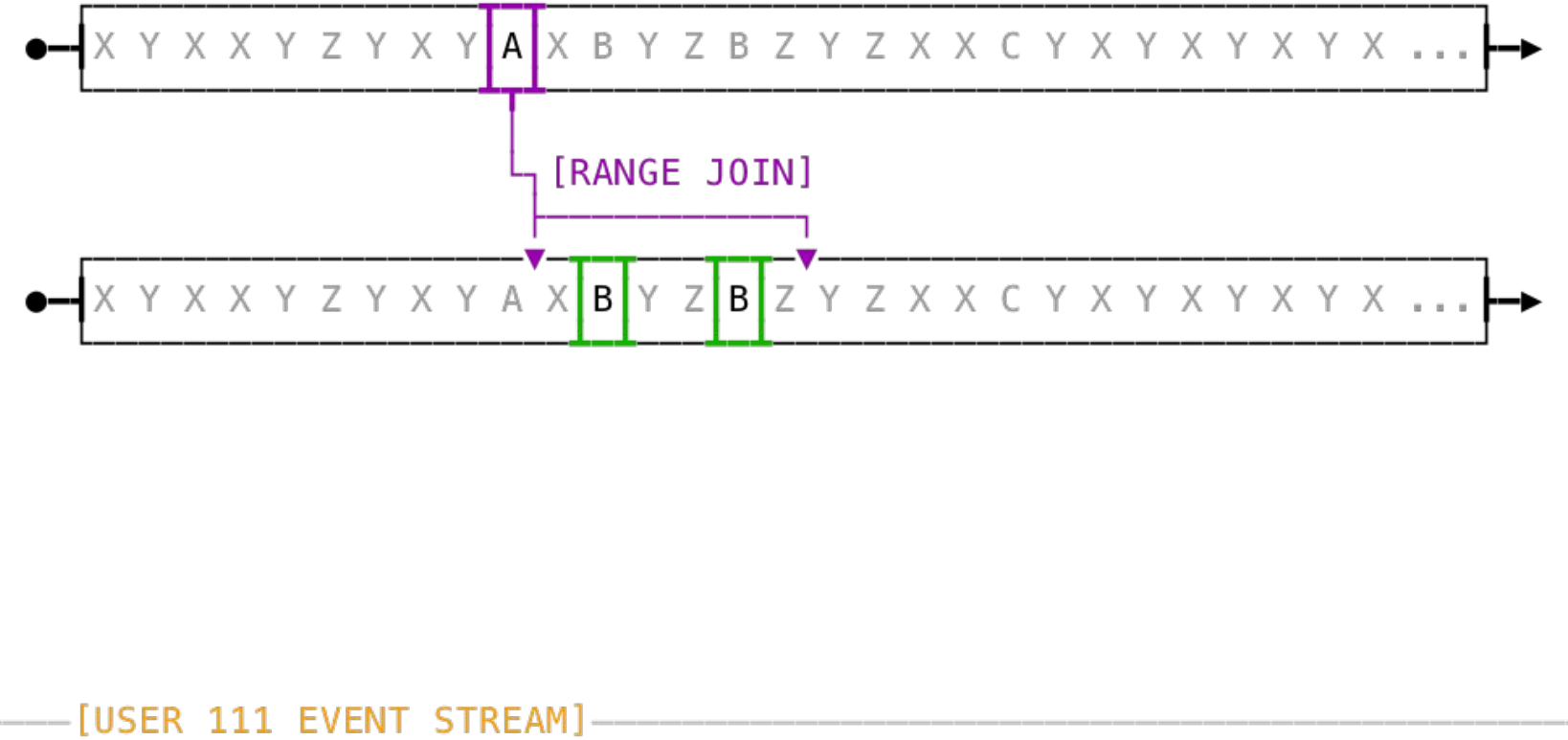
[USER 111 EVENT STREAM]

A›B

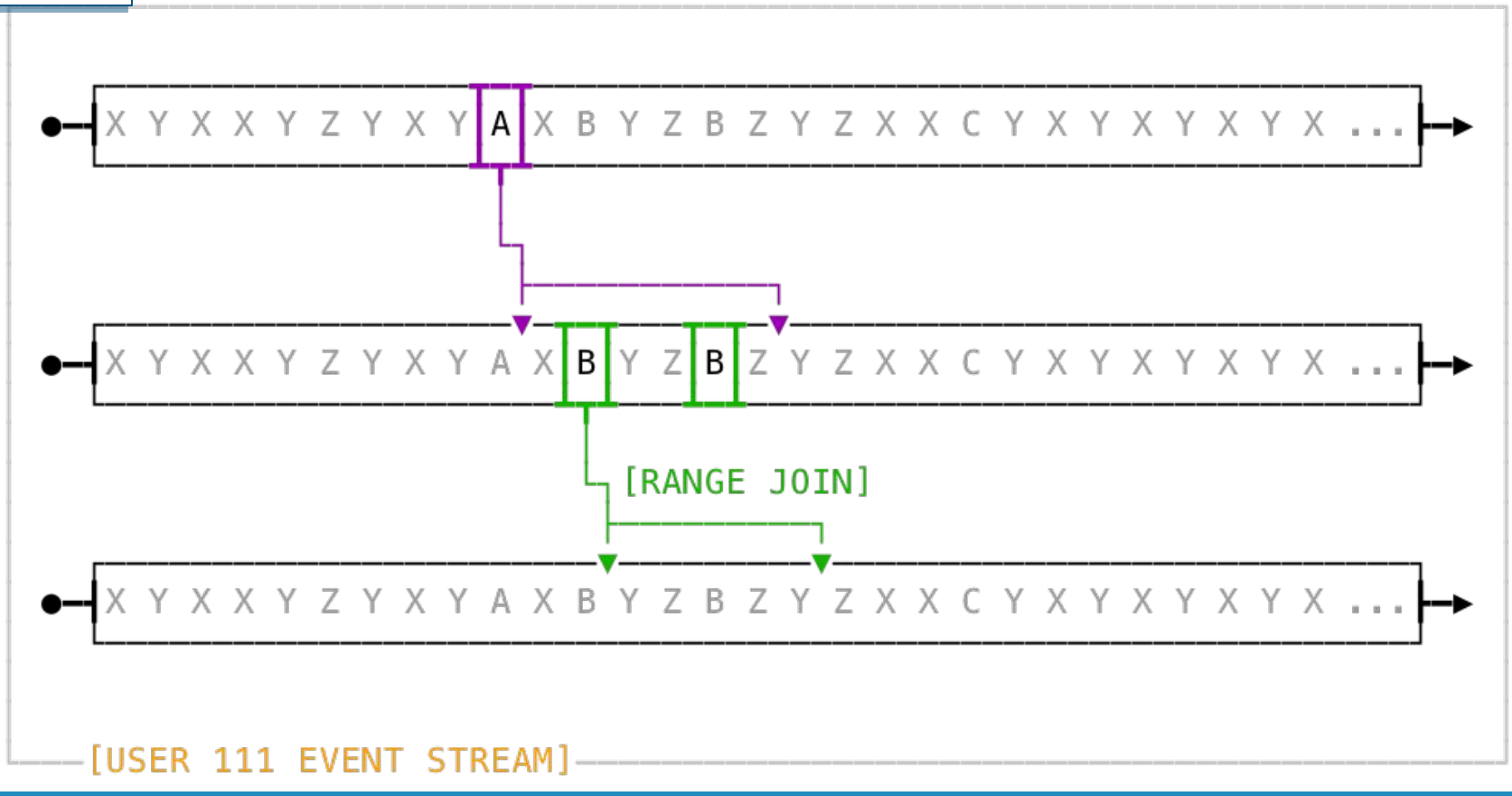


[USER 111 EVENT STREAM]

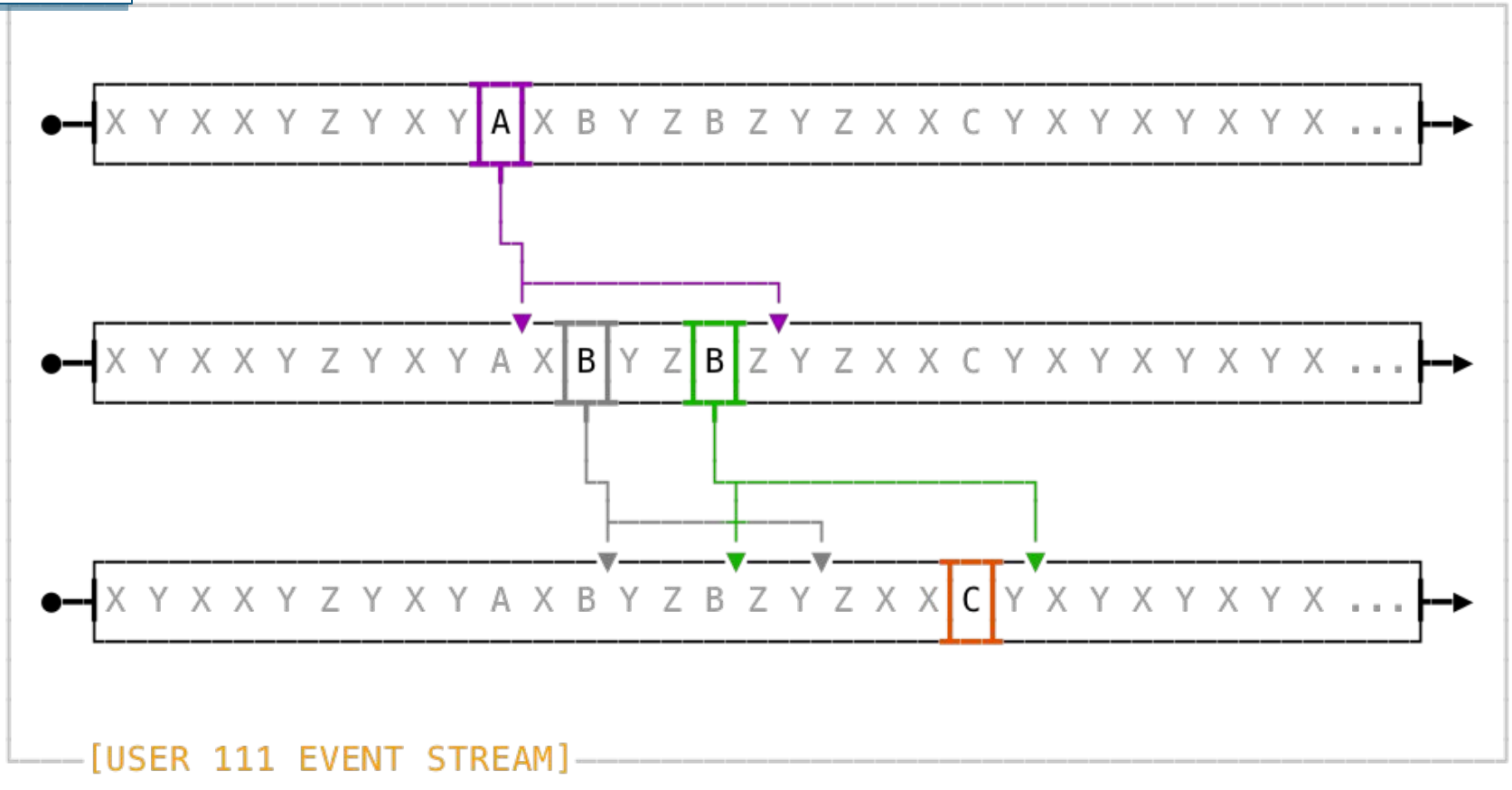
A›B›C



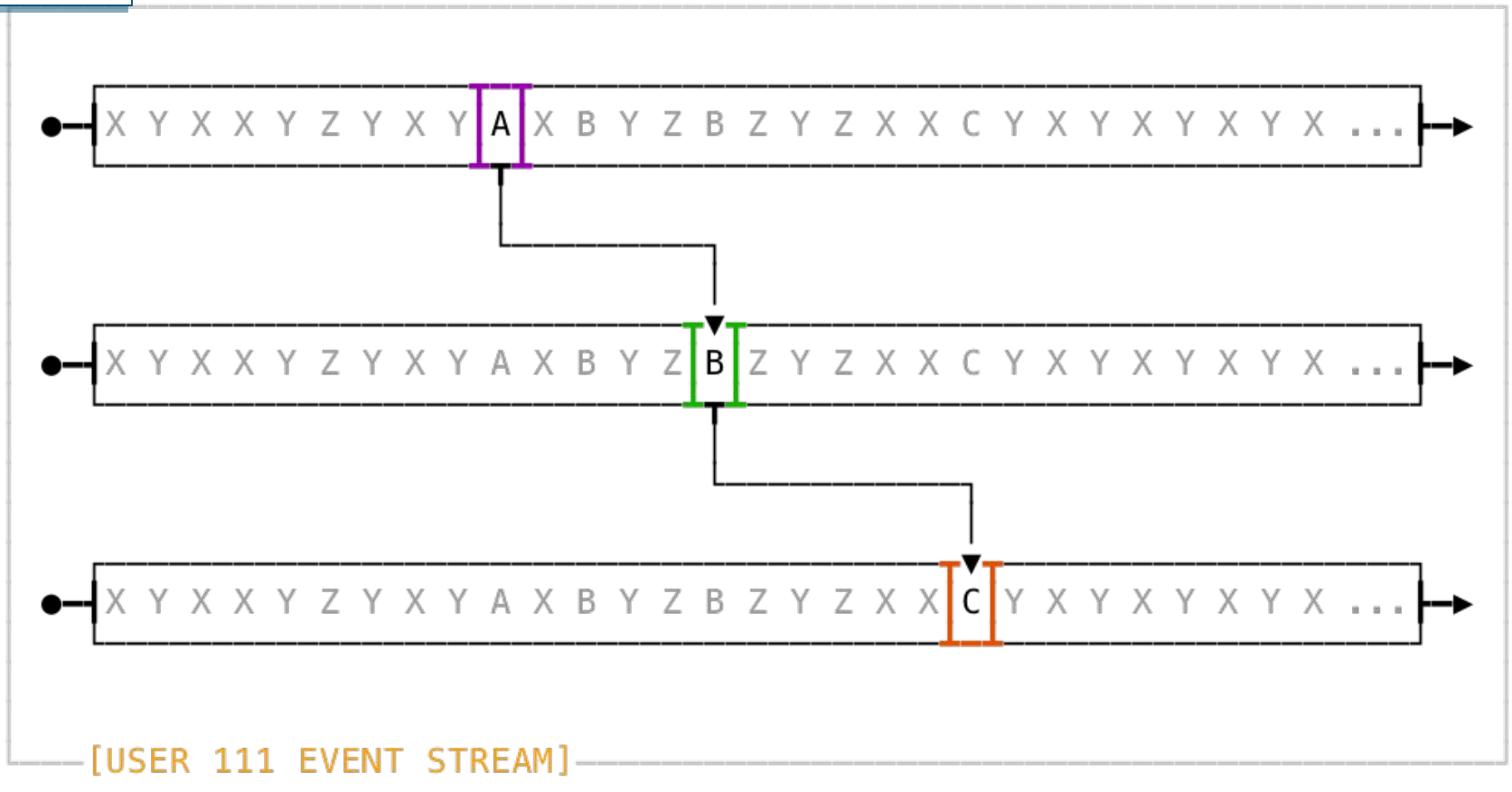
A›B›C



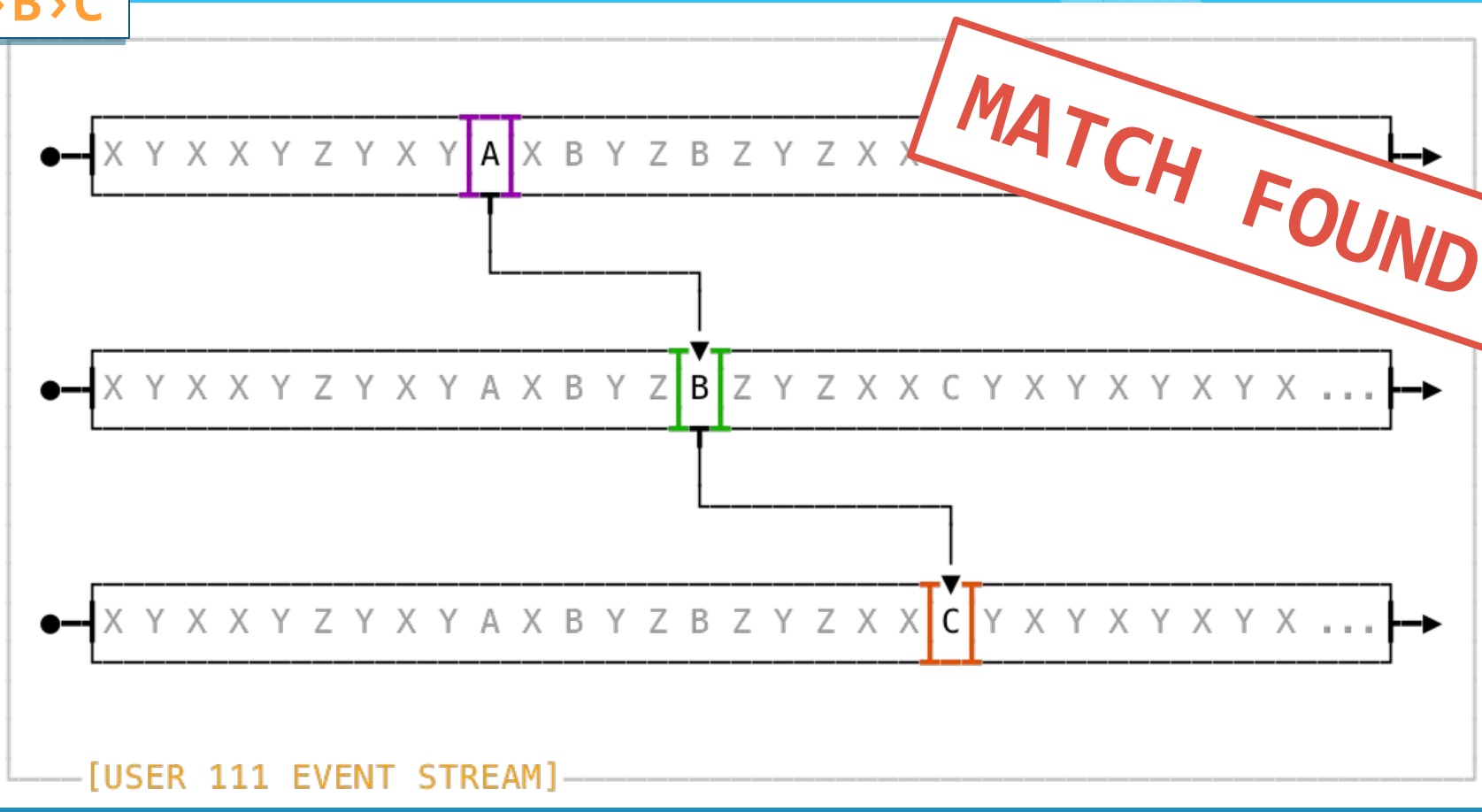
A›B›C



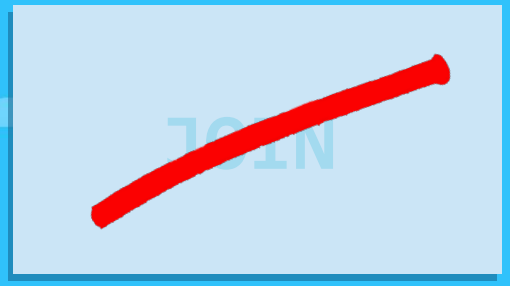
A›B›C



A›B›C



MATCH FOUND!



REGEX

PART THREE

REGEX

A›B›C

github.com/teej/sf-funnels

```
    WITH prep AS (...)
       , annotated AS (
  SELECT user_id
       , event_at
       , LAG(event_at) OVER (PARTITION BY user_id ORDER BY event_at) AS prior_at
       , CASE
           WHEN event_at < prior_at + INTERVAL '30 days'
           THEN '>' ELSE '#'
         END as spacer,
         spacer || event_name as event_token
    FROM prep
)
       , aggregated AS (
  SELECT user_id
       , LISTAGG(event_token) WITHIN GROUP (ORDER BY event_at) AS seq
    FROM annotated
   GROUP BY 1
)

 SELECT ...
   FROM aggregated
```

github.com/teej/sf-funnels

```
   WITH prep AS (

 SELECT user_id
      , event_at
      , event_name
   FROM events
  WHERE event_name IN ('A', 'B', 'C')

)
      , annotated AS (...)
      , aggregated AS (...)
SELECT ...
  FROM aggregated
```

DEFINE STEPS

github.com/teej/sf-funnels

```
    WITH prep AS (...)
       , annotated AS (

  SELECT user_id
       , event_at
       , LAG(event_at) OVER (PARTITION BY user_id ORDER BY event_at) AS prior_at
       , CASE
           WHEN event_at < prior_at + INTERVAL '30 days'
           THEN '>' ELSE '#'
         END AS spacer,
         spacer || event_name as event_token
    FROM prep

)
       , aggregated AS (...)
 SELECT ...
   FROM aggregated
```

UNDER TIME CONSTRAINTS

github.com/teej/sf-funnels

```
   WITH prep AS (...)
      , annotated AS (...)
      , aggregated AS (

  SELECT user_id
       , LISTAGG(event_token) WITHIN GROUP (ORDER BY event_at) AS seq
    FROM annotated
   GROUP BY 1

)
 SELECT ...
   FROM aggregated
```

IN ORDER

```
WITH prep AS (...)
   , annotated AS (...)
   , aggregated AS (...)

SELECT COUNT(DISTINCT IFF(seq REGEXP '.*A.*',     user_id, NULL)) AS users_A
     , COUNT(DISTINCT IFF(seq REGEXP '.*A>B.*',   user_id, NULL)) AS users_B
     , COUNT(DISTINCT IFF(seq REGEXP '.*A>B>C.*', user_id, NULL)) AS users_C
  FROM aggregated
```

THE MAGIC

TAKEAWAYS

REGEX

Blazing fast – scales to 10B+ rows and high K

Supports variety of patterns

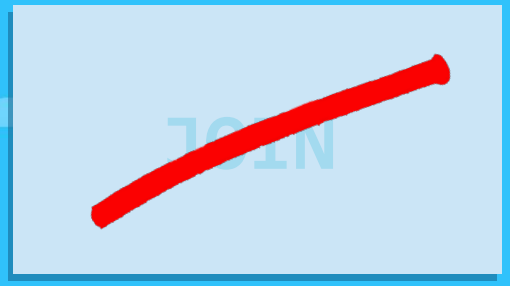
Limited time constraints

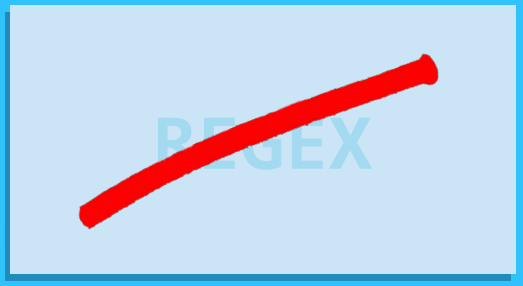
PERFORMANCE IN REVIEW

PERFORMANCE IN REVIEW

A›B›C›D

	SPEED	MAX K	PERFORMANCE	EASE OF USE
JOIN	44 s	3		★★★
WINDOW	27 s	3		★
REGEX	16 s	5		★★





ONE MORE THING...


FINALE
UDTF

github.com/teej/sf-funnels

```
  WITH prep AS (
SELECT user_id
     , DATE_PART(epoch_millisecond, event_at)::FLOAT AS event_epoch
     , event_name
  FROM events
 WHERE event_name IN (...)
)

SELECT COUNT(DISTINCT IFF(matches[0]::BOOLEAN, user_id, NULL)) AS users_A
     , ...
     , ...
     , ...
  FROM prep
     , TABLE(funnel_match(event_name, event_epoch, '...')
             OVER (PARTITION BY user_id ORDER BY event_epoch))
```

A›B›C›D›E›F›G

github.com/teej/sf-funnels

```
  WITH prep AS (
SELECT user_id
     , DATE_PART(epoch_millisecond, event_at)::FLOAT AS event_epoch
     , event_name
  FROM events
 WHERE event_name IN ('A', 'B', 'C', 'D', 'E', 'F', 'G')
)

SELECT COUNT(DISTINCT IFF(matches[0]::BOOLEAN, user_id, NULL)) AS users_A
     , COUNT(DISTINCT IFF(matches[1]::BOOLEAN, user_id, NULL)) AS users_B
     , COUNT(DISTINCT IFF(matches[2]::BOOLEAN, user_id, NULL)) AS users_C
     , ...
  FROM prep
     , TABLE(funnel_match(event_name, event_epoch, 'A>B>C>D>E>F>G')
             OVER (PARTITION BY user_id ORDER BY event_epoch))
```

A›B›C›D›E›F›G

github.com/teej/sf-funnels

DEFINE STEPS

```
  WITH prep AS (
SELECT user_id
     , DATE_PART(epoch_millisecond, event_at)::FLOAT AS event_epoch
     , event_name
  FROM events
 WHERE event_name IN ('A', 'B', 'C', 'D', 'E', 'F', 'G')
)

SELECT COUNT(DISTINCT IFF(matches[0]::BOOLEAN, user_id, NULL)) AS users_A
     , COUNT(DISTINCT IFF(matches[1]::BOOLEAN, user_id, NULL)) AS users_B
     , COUNT(DISTINCT IFF(matches[2]::BOOLEAN, user_id, NULL)) AS users_C
     , ...
  FROM prep
     , TABLE(funnel_match(event_name, event_epoch, 'A>B>C>D>E>F>G')
             OVER (PARTITION BY user_id ORDER BY event_epoch))
```

A›B›C›D›E›F›G

github.com/teej/sf-funnels

```
  WITH prep AS (
SELECT user_id
     , DATE_PART(epoch_millisecond, event_at)::FLOAT AS event_epoch
     , event_name
  FROM events
 WHERE event_name IN ('A', 'B', 'C', 'D', 'E', 'F', 'G')
)

SELECT COUNT(DISTINCT IFF(matches[0]::BOOLEAN, user_id, NULL)) AS users_A
     , COUNT(DISTINCT IFF(matches[1]::BOO                    ) AS users_B
     , COUNT(DISTINCT IFF(matches[2]::BOO                    ) AS users_C
     , ...
  FROM prep
     , TABLE(funnel_match(event_name, event_epoch, 'A>B>C>D>E>F>G')
             OVER (PARTITION BY user_id ORDER BY event_epoch))
```

IN ORDER

A›B›C›D›E›F›G

github.com/teej/sf-funnels

```
  WITH prep AS (
SELECT user_id
     , DATE_PART(epoch_millisecond, event_at)::FLOAT AS event_epoch
     , event_name
  FROM events
 WHERE event_name IN ('A', 'B', 'C', 'D', 'E', 'F', 'G')
)

SELECT COUNT(DISTINCT IFF(matches[0]::BOOLEAN, user_id, NULL)) AS users_A
     , COUNT(DISTINCT IFF(matches[1]::BOOLEAN, user_id,
     , COUNT(DISTINCT IFF(matches[2]::BOOLEAN, user_id,
     , ...
  FROM prep
     , TABLE(funnel_match(event_name, event_epoch, 'A>B>C>D>E>F>G')
             OVER (PARTITION BY user_id ORDER BY event_epoch))
```

UNDER TIME CONSTRAINTS

PERFORMANCE IN REVIEW

A›B›C›D

	SPEED	MAX K	PERFORMANCE	EASE OF USE
JOIN	44 s	3		★★★
WINDOW	27 s	3		★
REGEX	16 s	5		★★

PERFORMANCE IN REVIEW

A›B›C›D

	SPEED	MAX K	PERFORMANCE	EASE OF USE
JOIN	44 s	3		★★★
WINDOW	27 s	3		★
REGEX	16 s	5		★★
UDTF	**12 s**	**8+**		★★★

TAKEAWAYS

UDTF

20% faster than any others

Single scan approach

Scales to 100B+ rows

Easy query language


EPILOGUE

github.com/teej/sf-funnels

github.com/teej/sf-funnels

JOIN[BIN]	ARRAY UDF
JOIN[K-PASS]	MATCH_RECOGNIZE
WINDOW[BIN]	CONNECT BY
REGEX[NESTED]	RECURSIVE CTE

TJ Murphy

HEAD OF DATA

MULTI MEDIA LLC

`TWITTER › teej_m`
`GITHUB  › teej`

Please take a moment to rate your experience during this session.

INCLUDE THIS SESSION'S ID:

OS303

snowflake®

THANK YOU.