

snowflake®

SUMMIT 2022

LIVE IN VEGAS





snowflake®



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



1

MIX THE INGREDIENTS

2

DRIZZLE BATTER INTO OIL



1

MIX THE INGREDIENTS

2

DRIZZLE BATTER INTO OIL

3

COOK IN OIL



1

MIX THE INGREDIENTS

2

DRIZZLE BATTER INTO OIL

3

COOK IN OIL

5 MINUTES



1

MIX THE INGREDIENTS

2

DRIZZLE BATTER INTO OIL

3

COOK IN OIL

4

REMOVE AND DRY

5 MINUTES



The background image shows a complex industrial structure from an aerial perspective. It features several large, dark, cylindrical funnels or pipes arranged in a grid-like pattern. These structures have a metallic, ribbed texture and are illuminated by sunlight, creating bright highlights and deep shadows. In the lower right foreground, there's a smaller, rectangular building with a corrugated metal roof and some red markings. A bright green arrow points upwards and to the left in the upper right corner of the image.

PRELUDE FUNNELS

A SET OF STEPS

MIX THE INGREDIENTS

DRIZZLE BATTER INTO OIL

COOK IN OIL

REMOVE AND DRY



IN ORDER

1

MIX THE INGREDIENTS

2

DRIZZLE BATTER INTO OIL

3

COOK IN OIL

4

REMOVE AND DRY



1

MIX THE INGREDIENTS

2

DRIZZLE BATTER INTO OIL

3

COOK IN OIL

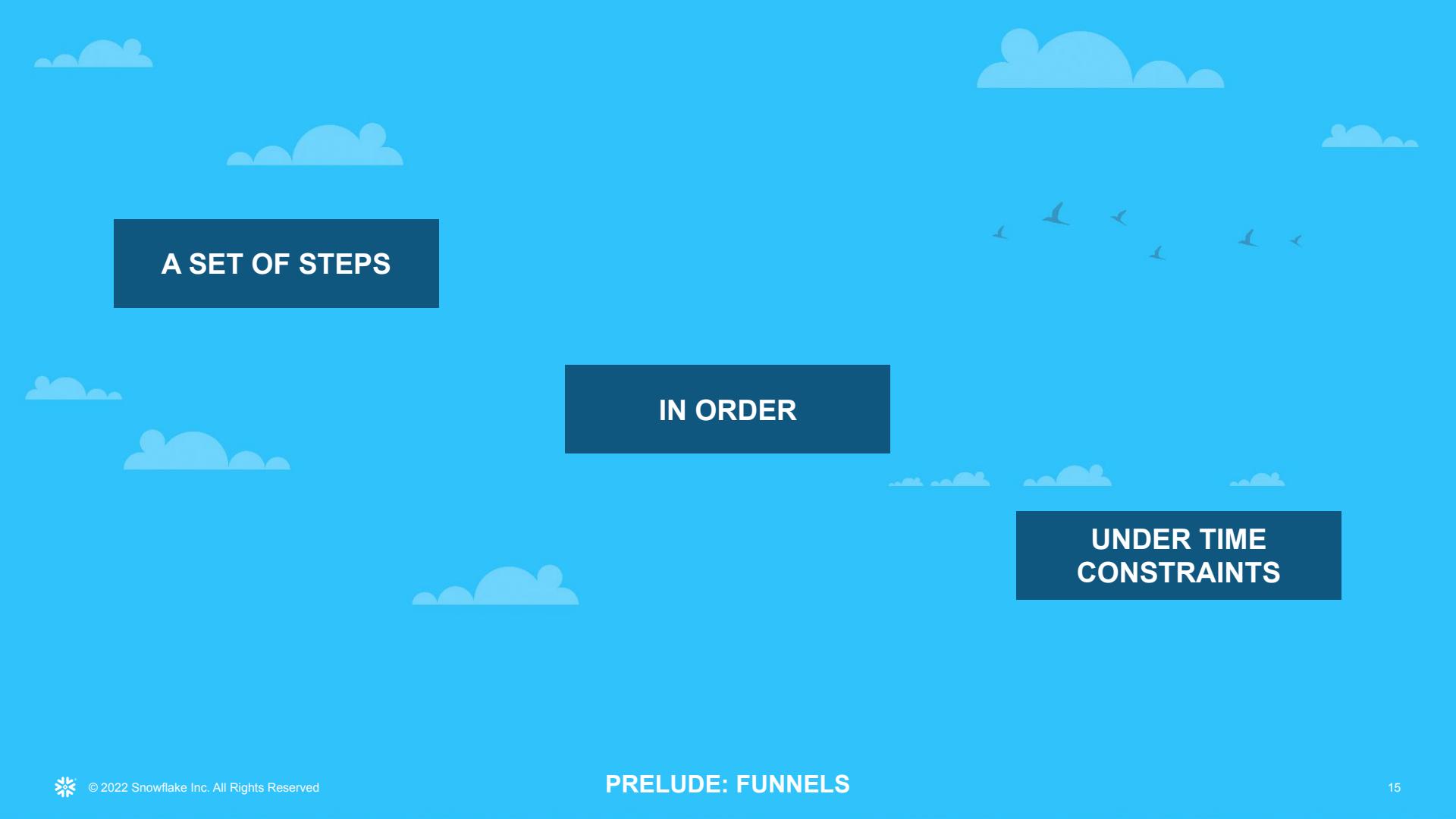
4

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



A > B





JOIN

WINDOW

REGEX



A close-up photograph of two hands clasped together. One hand, appearing darker-skinned, is gripping the fingers of a lighter-skinned hand. A green silicone wristband with a small white logo is visible on the darker-skinned wrist. The background is a plain, light grey.

PART ONE

THE JOIN



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'
```

```
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

```
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_id = step_A.event_id  
    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

```
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 | 22 s |  - |

NET: THE J

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

JOIN

PATTERN PERFORMANCE

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



The background image shows a window frame with four panes. The top-left pane shows a range of snow-capped mountains under a cloudy sky. The other three panes show a large body of water, likely a lake, with distant mountains across it. The overall scene is a scenic view from inside a building.

PART TWO

WINDOWS

```
WITH prep AS (...)  
    , scan AS (  
SELECT user_id  
    , is_step_A AS step_A_match  
    , LAG(IFNULL(step_A_match, event_at), 1) OVER (PARTITION BY user_id ORDER BY event_at)  
        IGNORE NULLS  
    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 IFNULL(step_A_match, user_id)) AS users_A  
    , COUNT(DISTINCT IFNULL(step_B_match, user_id)) AS users_B  
FROM scan
```

```
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

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

```
WITH prep AS (...  
    , scan AS (  
SELECT user_id  
    , is_step_A AS step_A_match  
  
    , LAG(IFNULL(step_A_match, event_at, NULL))  
        IGNORE NULLS  
        OVER (PARTITION BY u  
              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 |  - |
| WINDOW | A > B > C > D |  - |
| | A > B |  - |
| | A > B > C |  - |
| | A > B > C > D |  - |



TAKEAWAYS



WINDOW

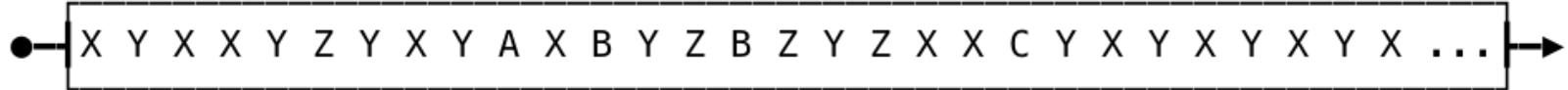
Performance beats JOINs

Scales to 1B+ rows

**Unwieldy to write for
complex funnels**



THE TROUBLE WITH SCANS



[USER 111 EVENT STREAM]

THE TROUBLE WITH SCANS

A > B

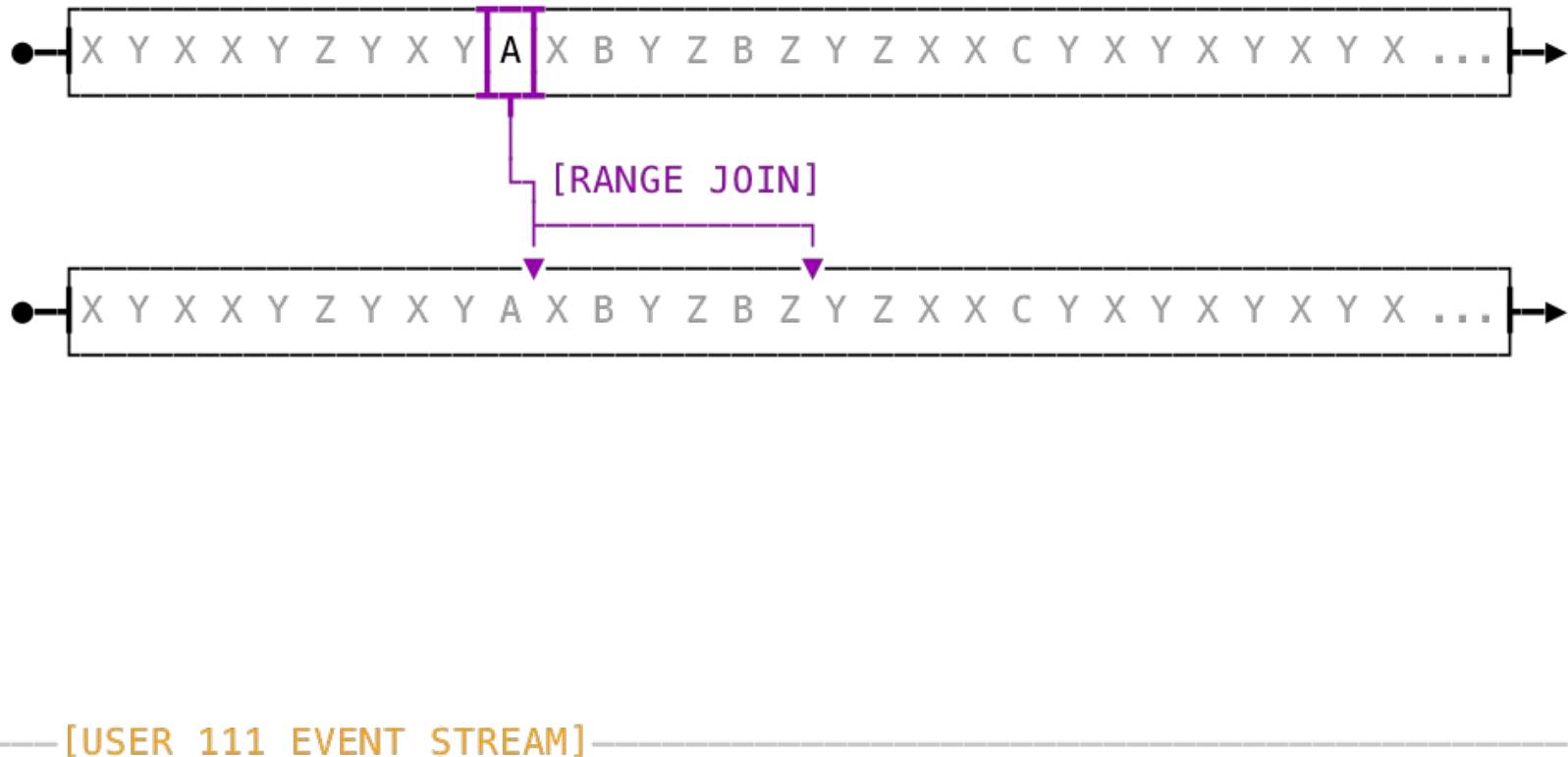
— [SCAN] —————→



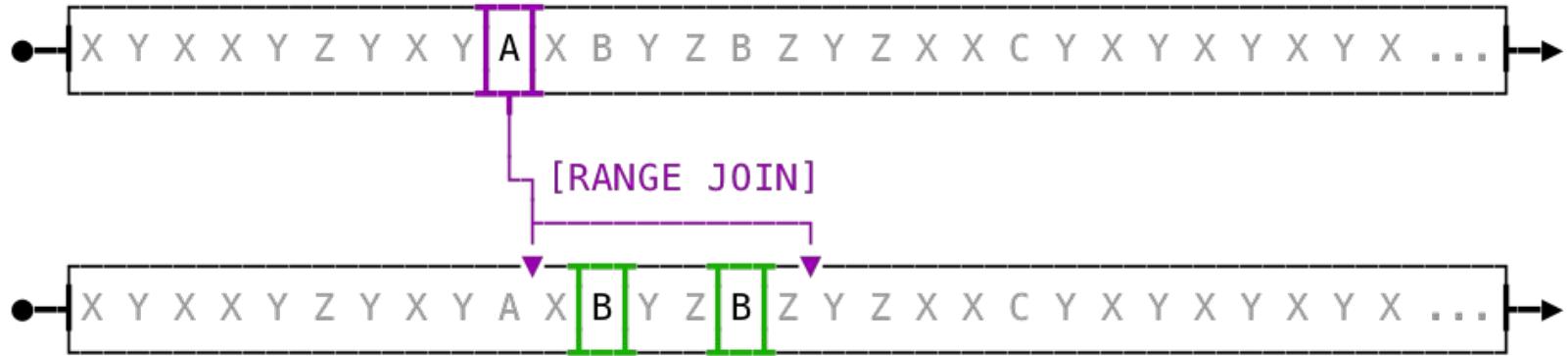
[USER 111 EVENT STREAM]

THE TROUBLE WITH SCANS





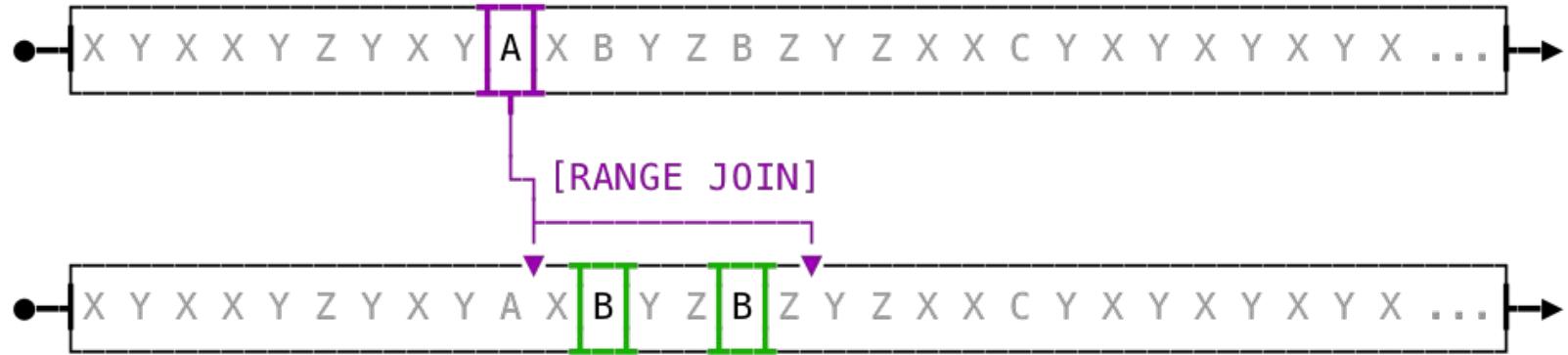
A > B



[USER 111 EVENT STREAM]

THE TROUBLE WITH SCANS

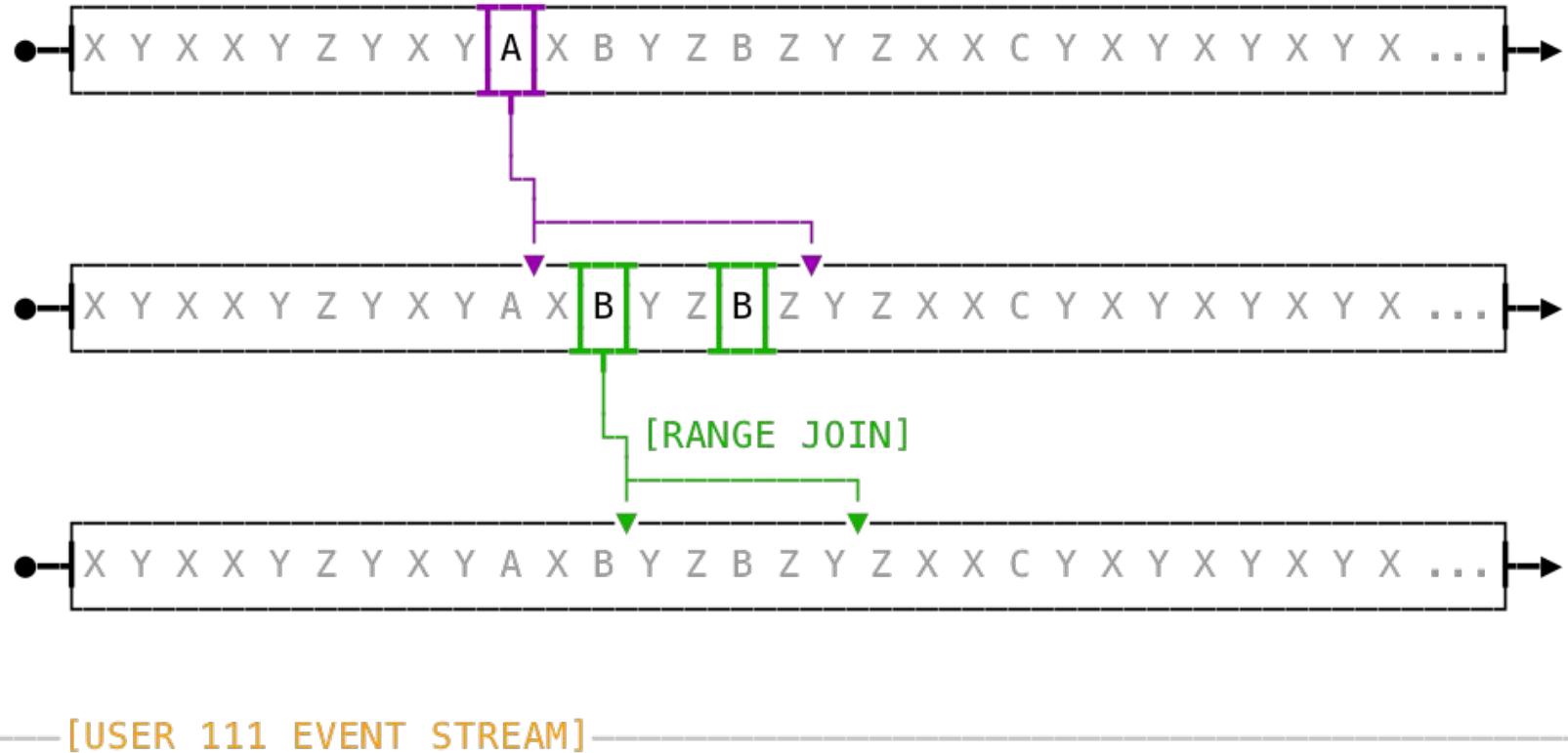




[USER 111 EVENT STREAM]

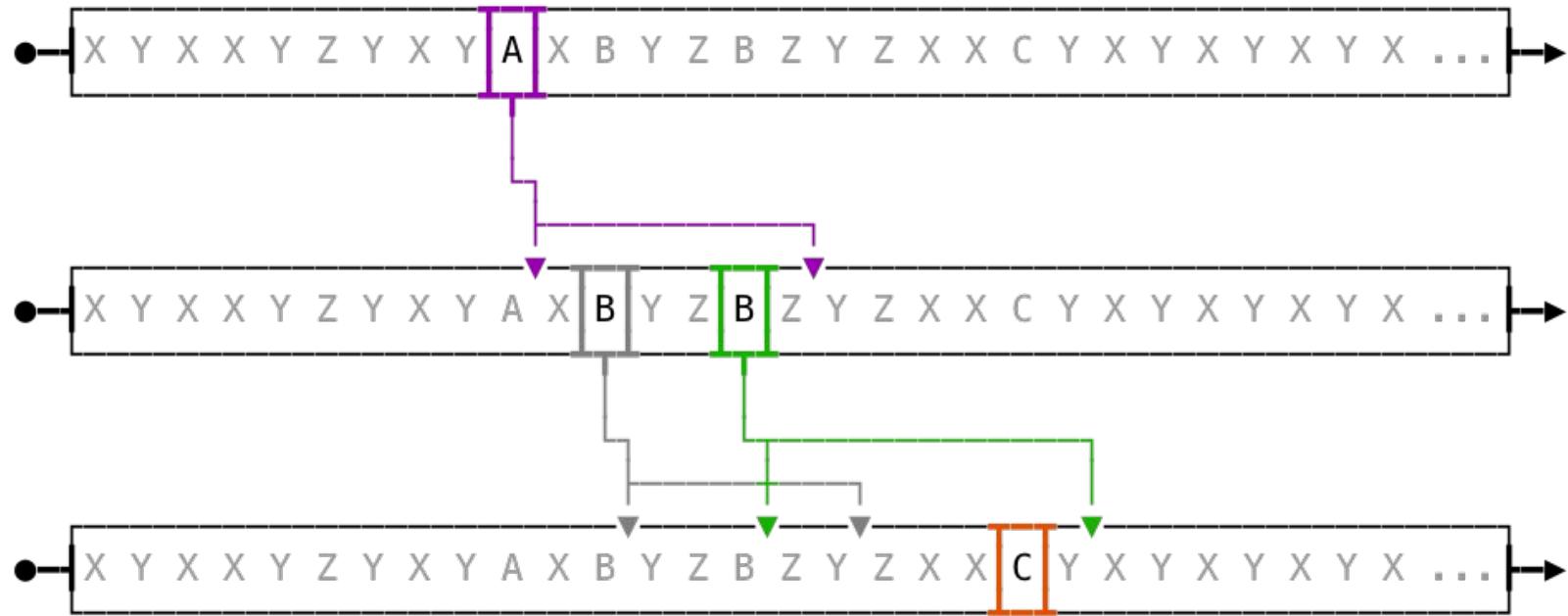
THE TROUBLE WITH SCANS





[USER 111 EVENT STREAM]

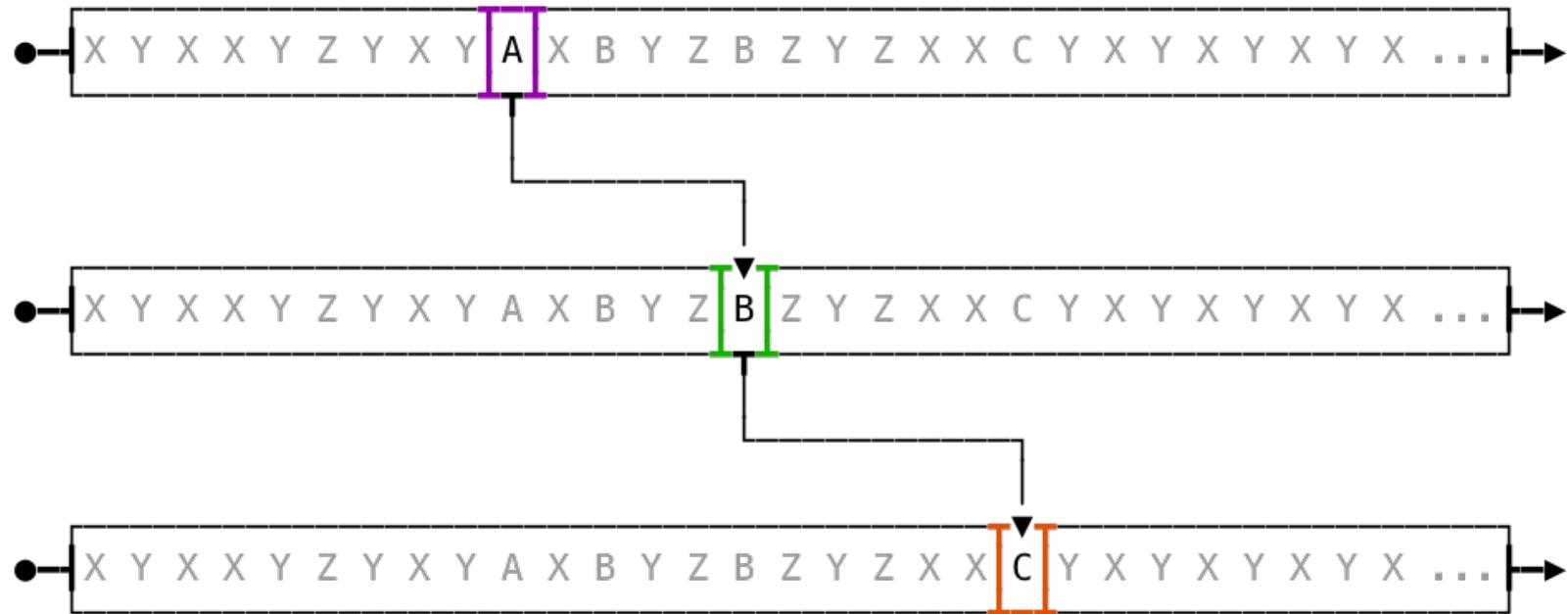
A > B > C



[USER 111 EVENT STREAM]



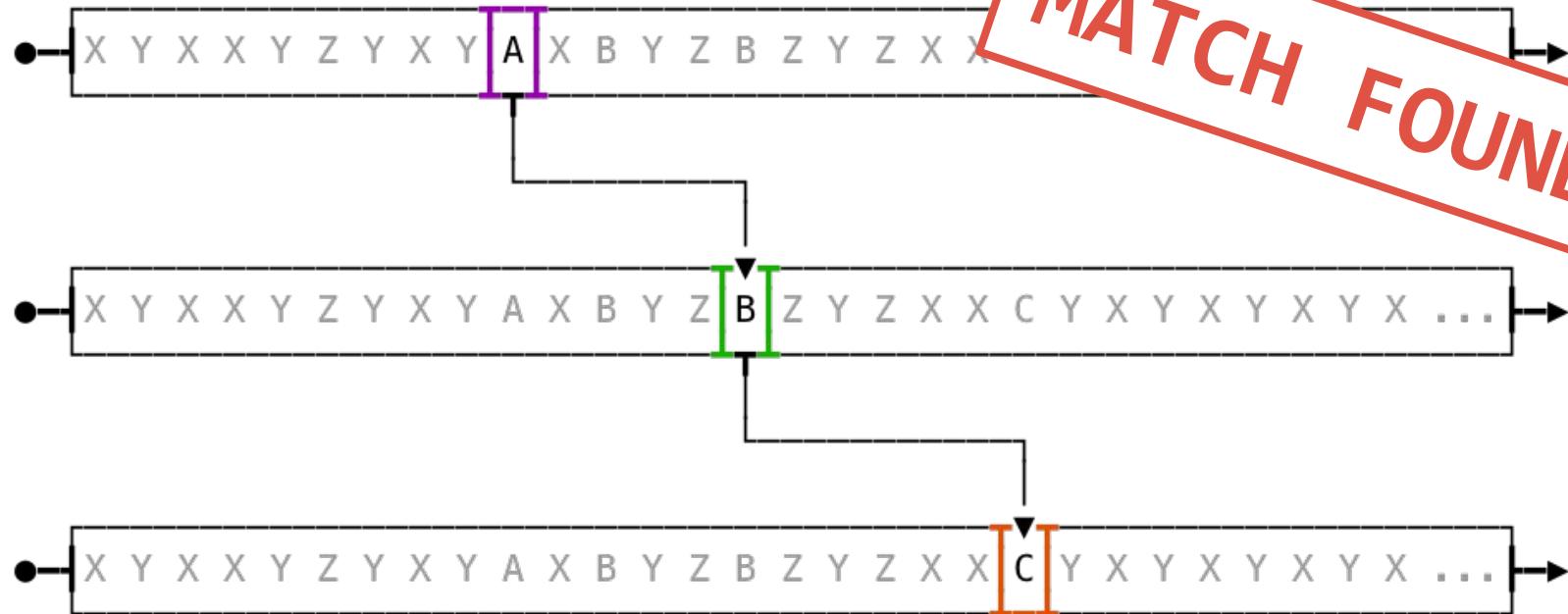
A > B > C



[USER 111 EVENT STREAM]



A > B > C



[USER 111 EVENT STREAM]



© 2022 Snowflake Inc. All Rights Reserved

THE TROUBLE WITH SCANS



PART THREE

REGEX



```
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
```

```
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

```
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

```
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 (...)
```

THE MAGIC

```
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
```



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 | <div style="width: 25%;"><div style="background-color: orange;"></div></div> | ★★★ |
| WINDOW | 27 s | 3 | <div style="width: 75%;"><div style="background-color: orange;"></div></div> | ★ |
| REGEX | 16 s | 5 | <div style="width: 100%;"><div style="background-color: orange;"></div></div> | ★★ |





ONE MORE THING...





FINALE
UDTF

```
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))
```

```
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))
```

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))
```

```
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))
```

IN ORDER

```
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))
```

UNDER TIME
CONSTRAINTS

PERFORMANCE IN REVIEW

A>B>C>D

| | SPEED | MAX K | PERFORMANCE | EASE OF USE |
|--------|-------|-------|---|-------------|
| JOIN | 44 s | 3 | <div style="width: 25%;"><div style="background-color: orange;"></div></div> | ★★★ |
| WINDOW | 27 s | 3 | <div style="width: 75%;"><div style="background-color: orange;"></div></div> | ★ |
| REGEX | 16 s | 5 | <div style="width: 100%;"><div style="background-color: orange;"></div></div> | ★★ |

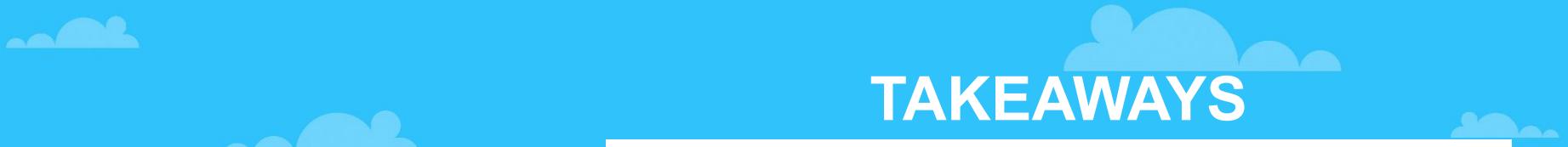


PERFORMANCE IN REVIEW

A>B>C>D

| | SPEED | MAX K | PERFORMANCE | EASE OF USE |
|--------|-------|-------|--|---|
| JOIN | 44 s | 3 | <div style="width: 20px; height: 10px; background-color: #f0a040;"></div> |  |
| WINDOW | 27 s | 3 | <div style="width: 45px; height: 10px; background-color: #f0a040;"></div> |  |
| REGEX | 16 s | 5 | <div style="width: 80px; height: 10px; background-color: #f0a040;"></div> |  |
| UDTF | 12 s | 8+ | <div style="width: 100px; height: 10px; background-color: #ff9900;"></div> |  |





TAKEAWAYS

UDTF

20% faster than any others

Single scan approach

Scales to 100B+ rows

Easy query language



A photograph of a two-lane asphalt road curving through a dense forest. The sky is a warm orange and yellow at sunset. The word "EPILOGUE" is centered in large, white, sans-serif capital letters.

EPILOGUE



github.com/teej/sf-funnels



github.com/teej/sf-funnels

JOIN[BIN]

JOIN[K-PASS]

WINDOW[BIN]

REGEX[NESTED]

ARRAY UDF

MATCH_RECOGNIZE

CONNECT BY

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**





THANK YOU.