

Row Pattern Recognition

Deep Dive!



ISO/IEC 19075-5 · SQL Feature R020 Implementation in Window Functions

Tatsuo Ishii — SRA OSS K.K. Henson Choi — INZENT CO.,LTD.

ishii@postgresql.org

assam@inzent.com

PGConf.dev
MAY 19-22, 2026 · VANCOUVER

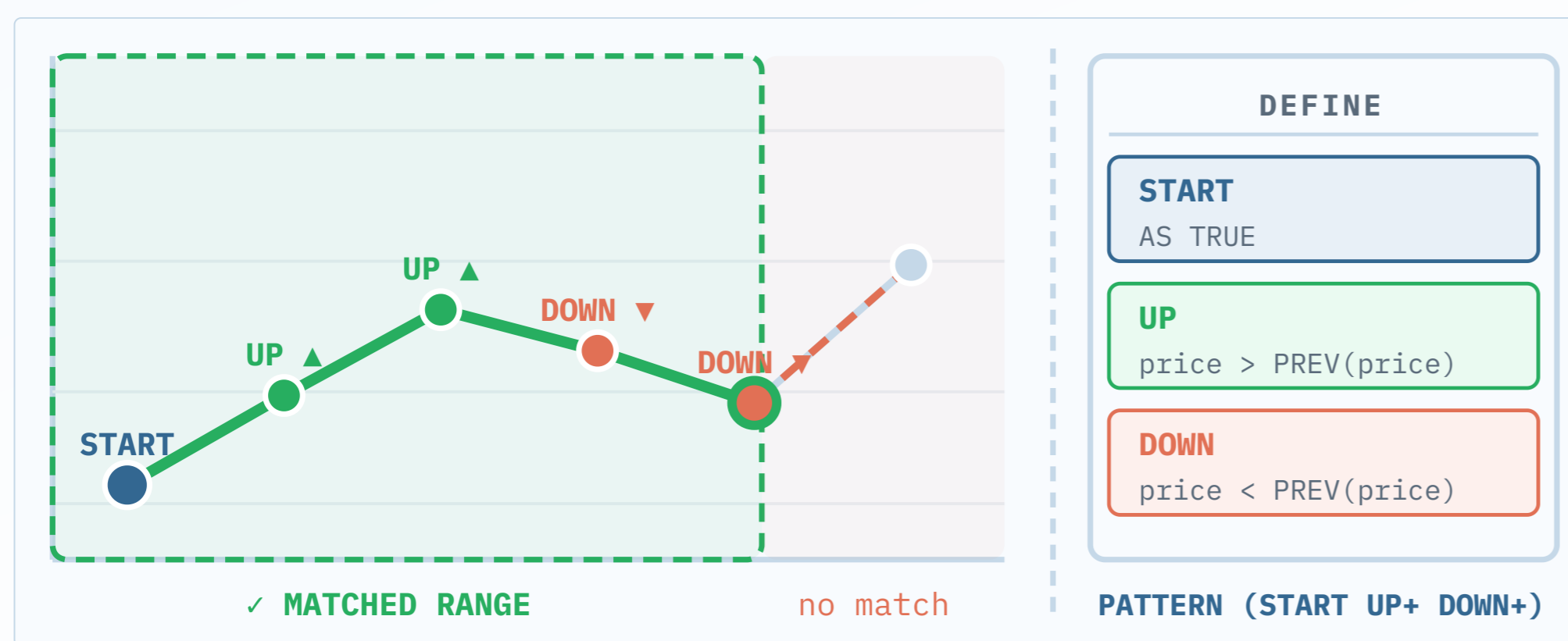
TARGET RELEASE

PostgreSQL 20

ROW PATTERN RECOGNITION · PGSQL-HACKERS

<https://www.postgresql.org/message-id/flat/20230625.210509.1276733411677577841.t-ishii%40sranhm.sra.co.jp>

WHAT IS ROW PATTERN RECOGNITION?



CORE INNOVATION – CONTEXT ABSORPTION

$O(n^2) \rightarrow O(n)$

LONGER THREADS EAT SHORTER ONES – $O(N^2)$ BECOMES $O(N)$.



NO PRIOR RDBMS HAS EVER EXECUTED NFA THIS WAY.

EXAMPLE – PATTERN (START UP+ DOWN+)

```
SELECT company, tdate, price,
       first_value(price) OVER w AS start_price,
       last_value(price) OVER w AS end_price
FROM stock
WINDOW w AS (
  PARTITION BY company ORDER BY tdate
  ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING
  AFTER MATCH SKIP PAST LAST ROW
  PATTERN (START UP+ DOWN+)
  DEFINE
    START AS TRUE,
    UP AS price > PREV(price),
    DOWN AS price < PREV(price)
);
```

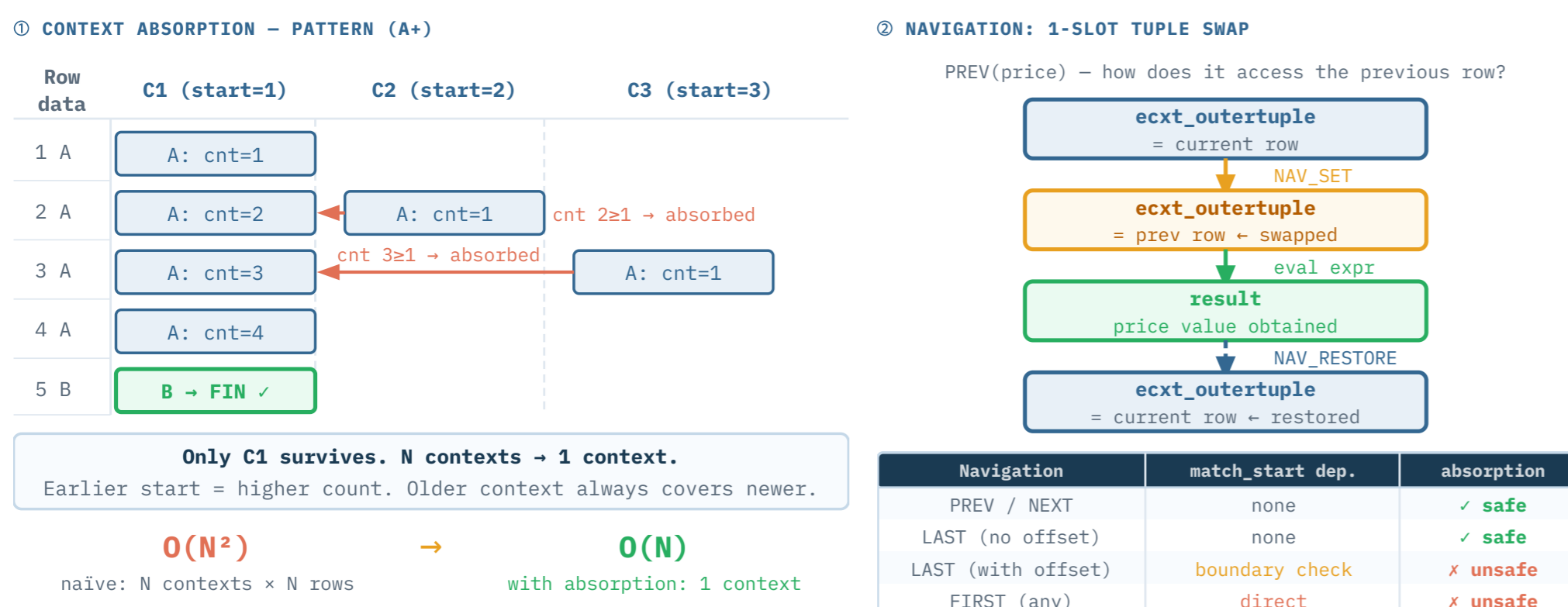
RESULT – QUERY OUTPUT

tdate	price	start	end	pattern
2024-01-01	100	100	108	START
2024-01-02	110	-	-	UP ▲
2024-01-03	120	-	-	UP ▲
2024-01-04	115	-	-	DOWN ▼
2024-01-05	108	-	-	DOWN ▼ ✓
2024-01-06	130	-	-	no match

DEFINE EVALUATES CONDITIONS ACROSS ROWS.

READ UP CLOSE

DESIGN & PERFORMANCE – WHAT MAKES THIS IMPLEMENTATION DIFFERENT



MATCH SUCCESS – PATTERN (A+ B+ C+ D)				MATCH FAILURE – PATTERN (A+ B+ C+ E)			
Rows	PostgreSQL	Trino	Speedup	Rows	PostgreSQL	Trino	Speedup
20,000	19 ms	~0.3 s	~16x	20,000	17 ms	358 s	21,000x
60,000	57 ms	~1.5 s	~26x	60,000	51 ms	4,424 s	86,700x
100,000	99 ms	~3.3 s	~33x	100,000	92 ms	20,014 s	217,000x

Both scale $O(n)$ – single-pass after match confirmed
PG: $O(n)$ via absorption – Trino: $O(n^2)$, no absorption

Test: PATTERN (A+ B+ C+ D/E) – match fails at the last row of 100K partition.
Success gap: Trino's JVM overhead. A non-PostgreSQL C-based DBMS would outperform on success, but degrade to $O(n^2)$ on failure – no context absorption.
Context absorption, unique to PostgreSQL, keeps it linear in both cases.

FOR FULL DESIGN DETAILS AND PATCH, SCAN THE QR CODE ABOVE.

WANTED

Think you know NFA theory beyond Ken Thompson?
RDBMS experience outlasting Michael Stonebraker?

→ We need you.

Review the first commit.
Become the author of the next patch.

REWARD

Your name as a contributor to one of the
largest commits in PostgreSQL history.

THE RPR GOLD MINE

80%
is still unwritten.