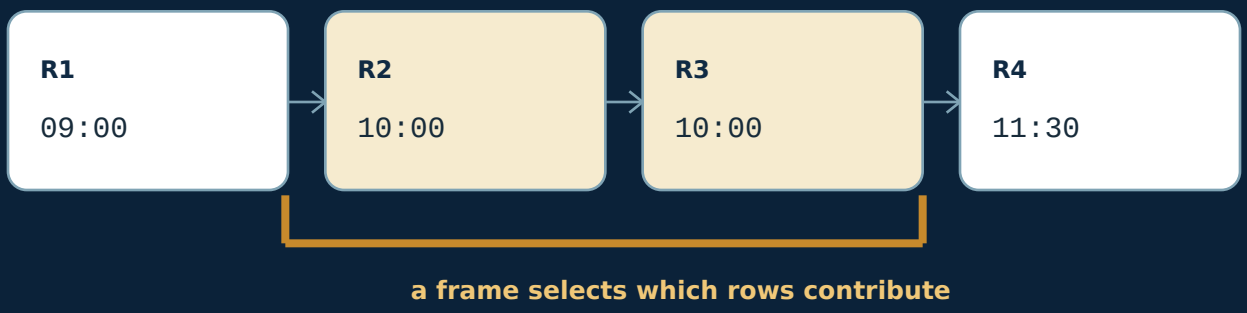
WindowFrame

Three SQL Window Function Cases

See how ORDER BY shapes the rows a calculation can see.

ORDERED ROWS



PRACTICE LOOP

01 READ
the rows and business rule

02 PREDICT
which rows enter the frame

03 VERIFY
with complete PostgreSQL SQL

Worked practice for readers comfortable with basic SQL

QUESTION 01

Peer timestamps and running balances

SCENARIO

Finance needs a timestamp-based running balance and a stable event-by-event balance for account A. Two entries share 10:00 UTC.

RAW INPUT / wf_s1_base.ledger_entries

`entry_id integer PK | account_id text | posted_at timestamptz | amount numeric`

entry_id	account	posted_at (UTC)	amount
1	A	2026-01-05 09:00	50.00
2	A	2026-01-05 10:00	-20.00
3	A	2026-01-05 10:00	15.00
4	A	2026-01-05 11:30	10.00
5	B	2026-01-05 10:00	999.00

YOUR TASK

Write SQL for account A returning entry_id, posted_at, amount, default_time_balance, and stable_row_balance. The first balance uses PostgreSQL's peer-aware ordered-aggregate default. The second uses stable (posted_at, entry_id) order and an explicit row frame.

CONVENTIONS

Filter account A. Display by posted_at, entry_id. Keep the tied 10:00 rows distinct.

Prediction prompt: do the tied rows share a default balance, and do their stable balances differ?

PREDICT BEFORE PAGE 4

entry 2 / default	entry 2 / stable

QUESTION 02

Stable inventory changes

SCENARIO

Starting stock is 100. Operations needs stock after every individual W1 change, including changes with tied timestamps.

RAW INPUT / wf_s1_transfer.inventory_changes

`change_id integer PK | sku text | effective_at timestamptz | delta integer`

change_id	sku	effective_at (UTC)	delta
11	W1	2026-02-01 08:00	-4
12	W1	2026-02-01 08:00	7
13	W1	2026-02-01 09:15	-3
14	W1	2026-02-01 09:15	-2
15	W2	2026-02-01 08:00	50

YOUR TASK

Write SQL for SKU W1 returning change_id, effective_at, delta, and stock_after_change. Use stable (effective_at, change_id) order with an explicit row frame. Add each delta to starting stock 100.

CONVENTIONS

Filter W1. Display by effective_at, change_id. W2 is a distractor and must not contribute.

Prediction prompt: write four stock values; tied rows still advance one change at a time.

PREDICT BEFORE PAGE 5

08:00 / id 11	08:00 / id 12	09:15 / ids 13-14

ANSWER 01

Peers and stable rows are different frames

COMPLETE EXECUTABLE SCRIPT

POSTGRESQL 17.11 / copy-selectable text

```
\set ON_ERROR_STOP on
SET TIME ZONE 'UTC';
\pset null '[NULL]'
\echo 'SOLUTION 1 - Peer timestamps and running balances'
SELECT entry_id, posted_at, amount,
       sum(amount) OVER (
           PARTITION BY account_id ORDER BY posted_at
       ) AS default_time_balance,
       sum(amount) OVER (
           PARTITION BY account_id ORDER BY posted_at, entry_id
           ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
       ) AS stable_row_balance
FROM wf_s1_base.ledger_entries
WHERE account_id = 'A'
ORDER BY posted_at, entry_id;
```

EXPECTED OUTPUT

id	time	amount	default	stable
1	09:00	50.00	50.00	50.00
2	10:00	-20.00	45.00	30.00
3	10:00	15.00	45.00	45.00
4	11:30	10.00	55.00	55.00

WHY ENTRY 2 DIFFERS

Calculation order is set inside OVER(...). Final display order is the last ORDER BY.

DEFAULT FRAME / ORDER BY posted_at

entry 1 IN +50

entry 2 IN -20

entry 3 IN +15

Includes entries 1, 2, 3 -> 45

The 10:00 peers share one default frame end.

STABLE ROW FRAME / posted_at, entry_id

entry 1 IN +50

entry 2 IN -20

Includes entries 1, 2 -> 30

ROWS advances through one stable row at a time.

ANSWER 02

A stable key makes tied changes deterministic

COMPLETE EXECUTABLE SCRIPT

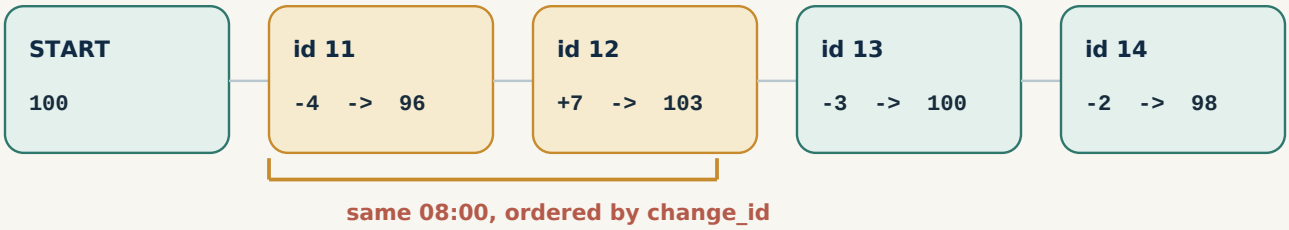
POSTGRESQL 17.11 / copy-selectable text

```
\set ON_ERROR_STOP on
SET TIME ZONE 'UTC';
\echo 'SOLUTION 2 - Stable inventory changes'
SELECT change_id, effective_at, delta,
       100 + sum(delta) OVER (
           PARTITION BY sku ORDER BY effective_at, change_id
           ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
       ) AS stock_after_change
FROM wf_s1_transfer.inventory_changes
WHERE sku = 'W1'
ORDER BY effective_at, change_id;
```

EXPECTED OUTPUT

change_id	time	delta	stock_after
11	08:00	-4	96
12	08:00	7	103
13	09:15	-3	100
14	09:15	-2	98

MECHANISM / START AT 100, THEN ADVANCE ONE ROW



WHY The W2 row is filtered out. For W1, the explicit ROWS frame uses the stable tie-breaker, so each tied change updates stock separately.