

Author: Nursultan Yerbatyr

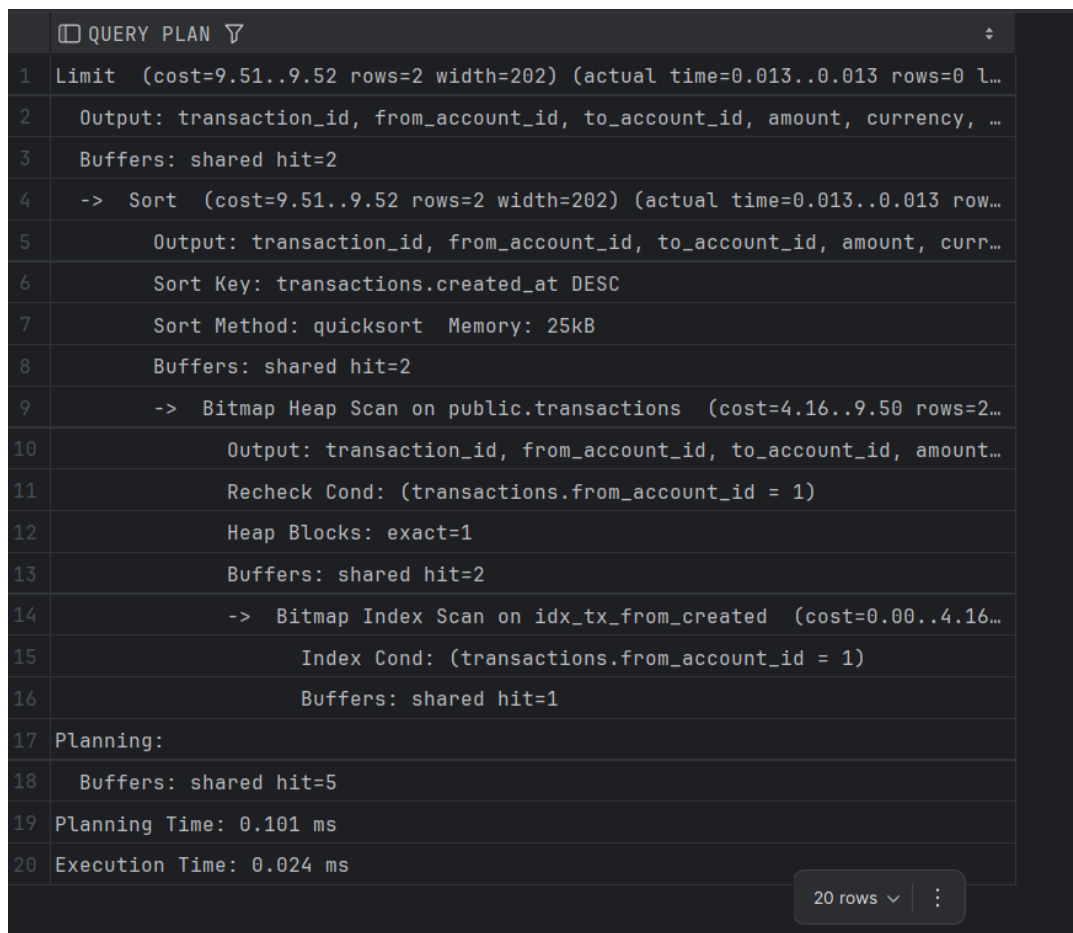
- Schema: customers, accounts, exchange_rates, transactions, audit_log
- Sample data (≥ 10 customers/accounts/transactions/rates)
- Procedures: process_transfer(...) and process_salary_batch(...)
- Reporting views: customer_balance_summary, daily_transaction_report, suspicious_activity_view (WITH security_barrier)
- Indexes: B-tree, composite, hash, GIN (JSONB), partial, expression, covering index
- Materialized view: mv_salary_summary
- Test queries and EXPLAIN suggestions at file end

Important notes

- process_transfer uses SELECT ... FOR UPDATE, SAVEPOINT and logs failures/successes to audit_log. It enforces sender status, sufficient balance, and daily limit (in KZT). Returns JSON with status and code.
- process_salary_batch uses pg_advisory_lock, stages payments in a temp table, uses per-item savepoints, applies updates atomically and bypasses daily limits (salary exception). Returns counts + failed details.
- Views convert balances to KZT using latest exchange_rates. Ensure exchange_rates contains needed direct X - > KZT rates for conversions.

Test & grading artifacts to include

- Capture EXPLAIN (ANALYZE, BUFFERS) outputs for:
 - SELECT * FROM accounts WHERE account_number = '...';
 - SELECT * FROM transactions WHERE from_account_id = 1 ORDER BY created_at DESC LIMIT 50;



The screenshot shows the query plan for the query: `SELECT * FROM transactions WHERE from_account_id = 1 ORDER BY created_at DESC LIMIT 50;`. The plan is as follows:

Line	Plan
1	Limit (cost=9.51..9.52 rows=2 width=202) (actual time=0.013..0.013 rows=0 l...
2	Output: transaction_id, from_account_id, to_account_id, amount, currency, ...
3	Buffers: shared hit=2
4	-> Sort (cost=9.51..9.52 rows=2 width=202) (actual time=0.013..0.013 row...
5	Output: transaction_id, from_account_id, to_account_id, amount, curr...
6	Sort Key: transactions.created_at DESC
7	Sort Method: quicksort Memory: 25kB
8	Buffers: shared hit=2
9	-> Bitmap Heap Scan on public.transactions (cost=4.16..9.50 rows=2...
10	Output: transaction_id, from_account_id, to_account_id, amount...
11	Recheck Cond: (transactions.from_account_id = 1)
12	Heap Blocks: exact=1
13	Buffers: shared hit=2
14	-> Bitmap Index Scan on idx_tx_from_created (cost=0.00..4.16...
15	Index Cond: (transactions.from_account_id = 1)
16	Buffers: shared hit=1
17	Planning:
18	Buffers: shared hit=5
19	Planning Time: 0.101 ms
20	Execution Time: 0.024 ms

At the bottom right of the screenshot, there is a control bar showing "20 rows" and a vertical ellipsis icon.

- Demonstrate concurrency:

- Session A: BEGIN; SELECT * FROM accounts WHERE account_number = '...' FOR UPDATE; (hold)
- Session B: run process_transfer that touches same account (it will block until A commits/rolls back).

Limitations / assumptions

- Exchange-rate fallback is basic; prefer direct p_currency->KZT rates.
- Audit logging is inserted by procedures (no triggers).
- Sample data should be verified for typos (account ids / balances)
- .