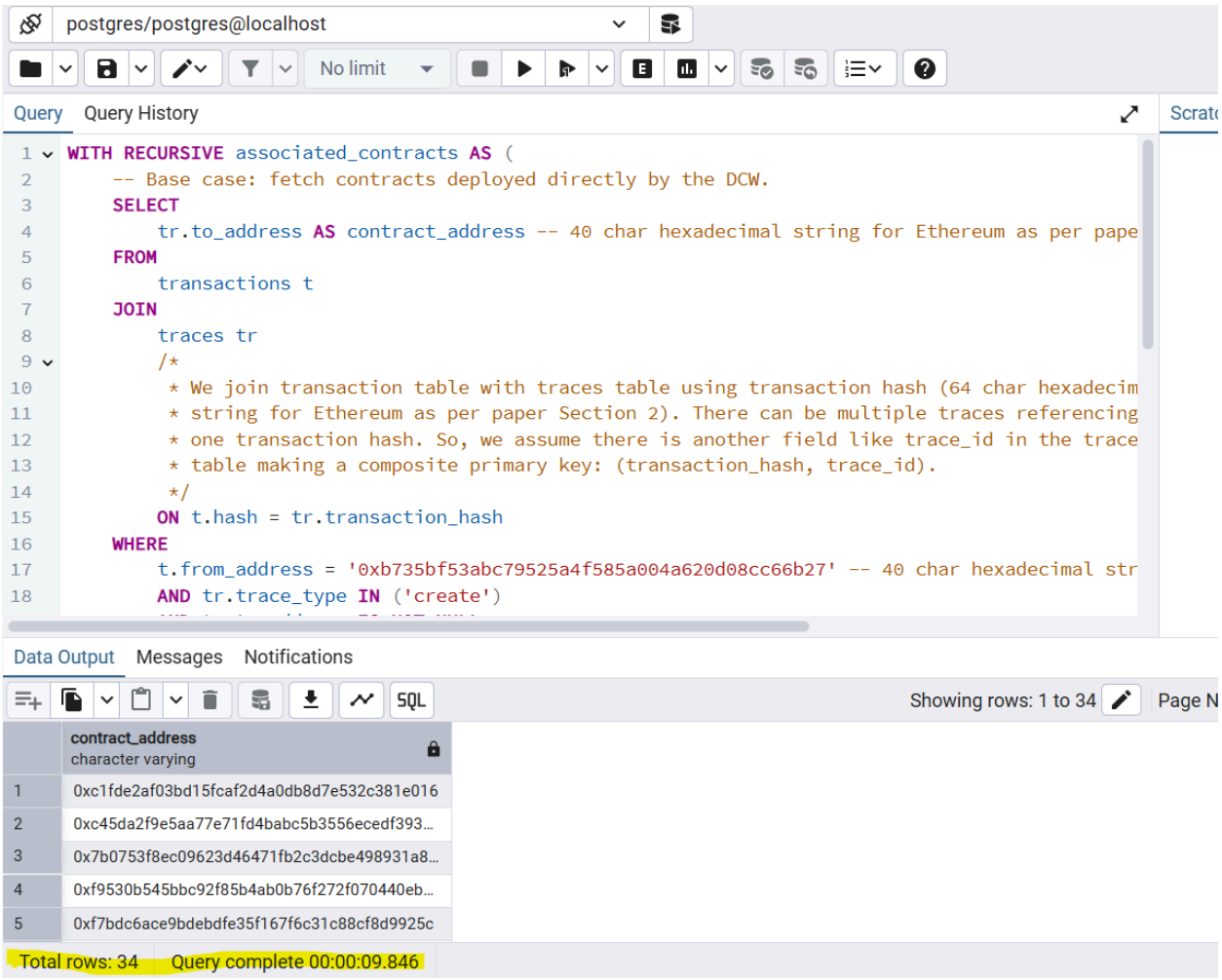
Problem statement

Fetch all associated contracts for the Ethereum wallet address: 0xb735bf53abc79525a4f585a004a620d08cc66b27.

Steps

- **Before creating an index on the transaction_hash column of the traces table:**

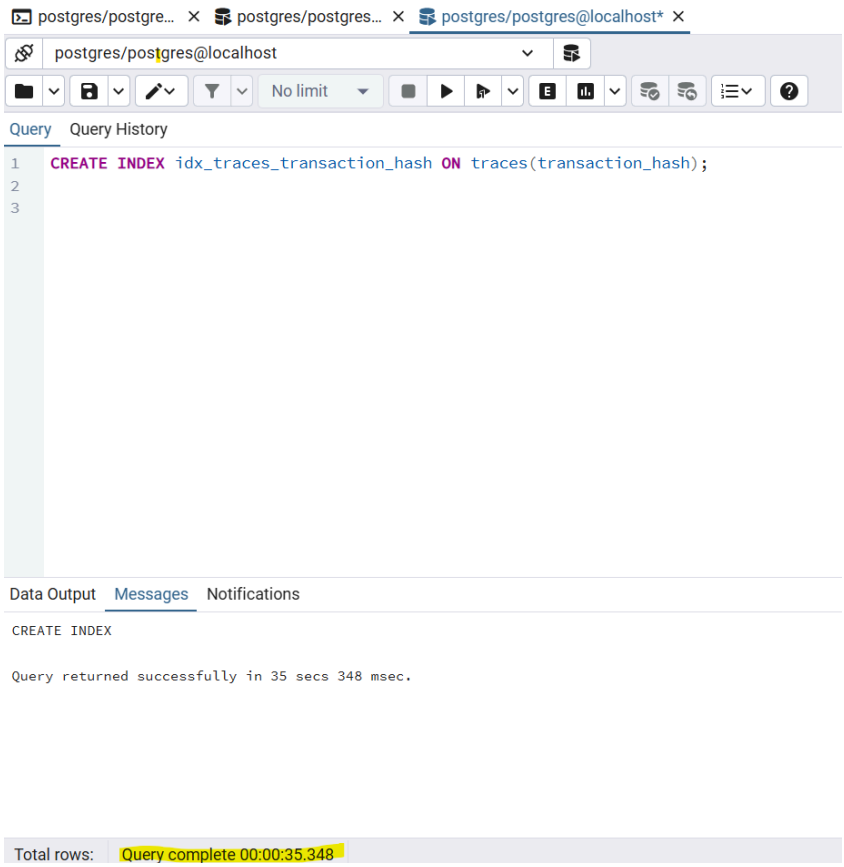
Output



Query Plan using EXPLAIN

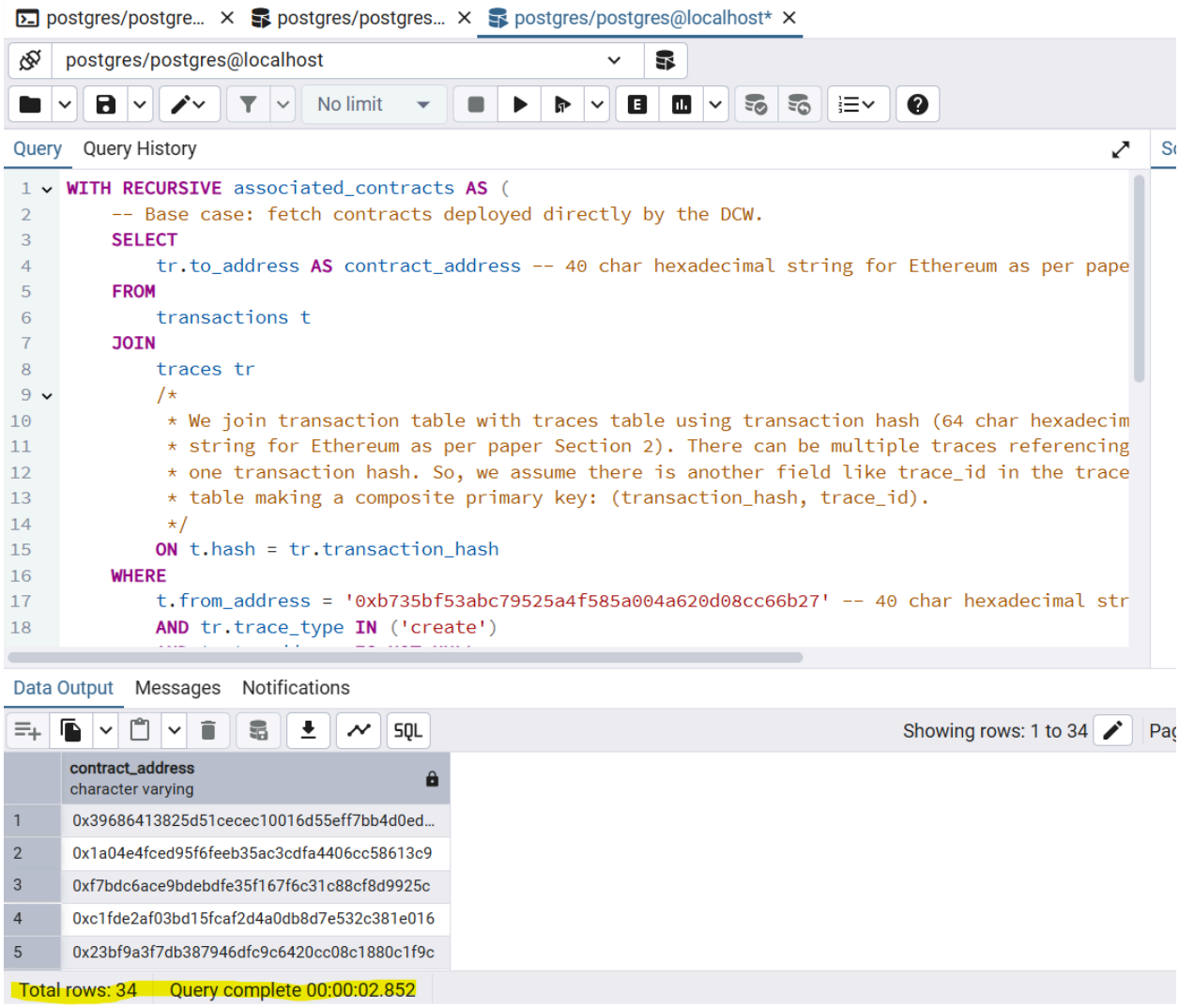
```
QUERY PLAN
CTE Scan on associated_contracts  (cost=3752262.34..3752265.76 rows=171 width=32)
  CTE associated_contracts
    ->  Recursive Union  (cost=121885.22..3752262.34 rows=171 width=43)
          ->  Gather  (cost=121885.22..338811.18 rows=1 width=43)
                Workers Planned: 2
                ->  Parallel Hash Join  (cost=120885.22..337811.08 rows=1 width=43)
                      Hash Cond: ((tr.transaction_hash)::text = (t.hash)::text)
                      ->  Parallel Seq Scan on traces tr  (cost=0.00..216910.14 rows=4190 width=110)
                            Filter: ((to_address IS NOT NULL) AND ((trace_type)::text = 'create'::text))
                      ->  Parallel Hash  (cost=120884.47..120884.47 rows=60 width=67)
                            ->  Parallel Seq Scan on transactions t  (cost=0.00..120884.47 rows=60 width=67)
                                  Filter: ((from_address)::text = '0xb735bf53abc79525a4f585a004a620d08cc66b27'::text)
          ->  Hash Join  (cost=217962.84..341344.94 rows=17 width=43)
                Hash Cond: ((t_1.from_address)::text = (ac.contract_address)::text)
                ->  Gather  (cost=217962.51..341317.81 rows=7104 width=86)
                      Workers Planned: 2
                      ->  Parallel Hash Join  (cost=216962.51..339607.41 rows=2960 width=86)
                            Hash Cond: ((t_1.hash)::text = (tr_1.transaction_hash)::text)
                            ->  Parallel Seq Scan on transactions t_1  (cost=0.00..119136.38 rows=699238 width=110)
                            ->  Parallel Hash  (cost=216910.14..216910.14 rows=4190 width=110)
                                  ->  Parallel Seq Scan on traces tr_1  (cost=0.00..216910.14 rows=4190 width=110)
                                        Filter: ((to_address IS NOT NULL) AND ((trace_type)::text = 'create'::text))
                ->  Hash  (cost=0.20..0.20 rows=10 width=32)
                      ->  WorkTable Scan on associated_contracts ac  (cost=0.00..0.20 rows=10 width=32)
```

- **Index creation**



- **After creating an index on the transaction_hash column of the traces table:**

Output



Query Plan using EXPLAIN

```
QUERY PLAN
CTE Scan on associated_contracts  (cost=1521392.04..1521395.46 rows=171 width=32)
  CTE associated_contracts
    ->  Recursive Union  (cost=1000.55..1521392.04 rows=171 width=43)
          ->  Gather  (cost=1000.55..123123.26 rows=1 width=43)
                Workers Planned: 2
                ->  Nested Loop  (cost=0.56..122123.16 rows=1 width=43)
                      ->  Parallel Seq Scan on transactions t  (cost=0.00..120884.47 rows=60 width=67)
                            Filter: ((from_address)::text = '0xb735bf53abc79525a4f585a004a620d08cc66b27'::text)
                      ->  Index Scan using idx_traces_transaction_hash on traces tr  (cost=0.56..20.63 rows=1 width=110)
                            Index Cond: ((transaction_hash)::text = (t.hash)::text)
                            Filter: ((to_address IS NOT NULL) AND ((trace_type)::text = 'create'::text))
          ->  Nested Loop  (cost=0.88..139826.71 rows=17 width=43)
                ->  Hash Join  (cost=0.33..135258.32 rows=3914 width=67)
                      Hash Cond: ((t_1.from_address)::text = (ac.contract_address)::text)
                      ->  Seq Scan on transactions t_1  (cost=0.00..128925.71 rows=1678171 width=110)
                      ->  Hash  (cost=0.20..0.20 rows=10 width=32)
                            ->  WorkTable Scan on associated_contracts ac  (cost=0.00..0.20 rows=10 width=32)
                ->  Index Scan using idx_traces_transaction_hash on traces tr_1  (cost=0.56..1.16 rows=1 width=110)
                      Index Cond: ((transaction_hash)::text = (t_1.hash)::text)
                      Filter: ((to_address IS NOT NULL) AND ((trace_type)::text = 'create'::text))
```

Observation

We can see that the query time **decreased** from **9.85s to 2.85s** after creating an index on the transaction_hash column in the traces table. The query plans generated using the EXPLAIN clause confirm the fact that creating the index avoided full table lookups for traces data and led to a **~71% drop** in the query execution time.