Indexes were created to hasten the query speed for observation (1B rows).

```sql
-- Composite index (most powerful - covers most query patterns)
CREATE INDEX CONCURRENTLY idx_obs_user_metric_time
ON observation (user_id, metric_id, date_time);

-- Metric index (for queries filtering by metric type)
CREATE INDEX CONCURRENTLY idx_obs_metric
ON observation (metric_id);

-- DateTime index (for time-range queries)
CREATE INDEX CONCURRENTLY idx_obs_datetime
ON observation (date_time);
```

Before the script EXPLAIN ANALYZE of the sleepQuality script took 153691.761 ms to execute

```
                                                                    QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------------------------------------
 Gather Merge  (cost=18794806.20..18806538.32 rows=100554 width=31) (actual time=153678.960..153689.973 rows=88035 loops=1)
   Workers Planned: 2
   Workers Launched: 2
   ->  Sort  (cost=18793806.18..18793931.87 rows=50277 width=31) (actual time=153673.483..153674.112 rows=29345 loops=3)
         Sort Key: o.value
         Sort Method: quicksort  Memory: 2482kB
         Worker 0:  Sort Method: quicksort  Memory: 2383kB
         Worker 1:  Sort Method: quicksort  Memory: 2673kB
         ->  Parallel Hash Join  (cost=6139.72..18789880.15 rows=50277 width=31) (actual time=152446.014..153666.774 rows=29345 loops=3)
               Hash Cond: (o.user_id = u.user_id)
               ->  Parallel Seq Scan on observation o  (cost=0.00..18783342.40 rows=50277 width=18) (actual time=152405.962..153620.154 rows=29345 loops=3)
                     Filter: ((value < '60'::numeric) AND (metric_id = 43) AND (date_time >= (now() - '24:00:00'::interval)))
                     Rows Removed by Filter: 334417668
               ->  Parallel Hash  (cost=5618.88..5618.88 rows=41667 width=21) (actual time=39.721..39.722 rows=33333 loops=3)
                     Buckets: 131072  Batches: 1  Memory Usage: 6656kB
                     ->  Hash Join  (cost=3025.29..5618.88 rows=41667 width=21) (actual time=26.685..35.258 rows=33333 loops=3)
                           Hash Cond: (u.user_id = ui.user_id)
                           ->  Parallel Index Only Scan using "User_pkey" on "User" u  (cost=0.29..2020.96 rows=41667 width=4) (actual time=0.451..2.931 rows=33333 loops=3)
                                 Heap Fetches: 0
                           ->  Hash  (cost=1775.00..1775.00 rows=100000 width=17) (actual time=25.986..25.986 rows=100000 loops=3)
                                 Buckets: 131072  Batches: 1  Memory Usage: 6080kB
                                 ->  Seq Scan on user_info ui  (cost=0.00..1775.00 rows=100000 width=17) (actual time=0.039..10.363 rows=100000 loops=3)
 Planning Time: 9.119 ms
 Execution Time: 153691.761 ms
(24 rows)
```

After creating the indexes the same script executed in 200.695ms, a 99.87% reduction in time.

```
~/Projects/SQL/data management/Scenario 1 ❯ psql -d health_tracker -f "/Users/alectorres/Projects/SQL/data management/Scenario 1/sleepQuality.sql"
                                                                    QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------------------------------------
 Gather Merge  (cost=2791130.27..2802839.29 rows=100356 width=31) (actual time=197.037..206.880 rows=88035 loops=1)
   Workers Planned: 2
   Workers Launched: 2
   ->  Sort  (cost=2790130.25..2790255.69 rows=50178 width=31) (actual time=191.710..192.320 rows=29345 loops=3)
         Sort Key: o.value
         Sort Method: quicksort  Memory: 2542kB
         Worker 0:  Sort Method: quicksort  Memory: 2491kB
         Worker 1:  Sort Method: quicksort  Memory: 2505kB
         ->  Nested Loop  (cost=3025.87..2786212.66 rows=50178 width=31) (actual time=28.169..185.754 rows=29345 loops=3)
               ->  Hash Join  (cost=3025.29..5618.88 rows=41667 width=21) (actual time=27.052..39.145 rows=33333 loops=3)
                     Hash Cond: (u.user_id = ui.user_id)
                     ->  Parallel Index Only Scan using "User_pkey" on "User" u  (cost=0.29..2020.96 rows=41667 width=4) (actual time=0.178..1.840 rows=33333 loops=3)
                           Heap Fetches: 0
                     ->  Hash  (cost=1775.00..1775.00 rows=100000 width=17) (actual time=26.616..26.616 rows=100000 loops=3)
                           Buckets: 131072  Batches: 1  Memory Usage: 6080kB
                           ->  Seq Scan on user_info ui  (cost=0.00..1775.00 rows=100000 width=17) (actual time=0.145..11.153 rows=100000 loops=3)
               ->  Index Scan using idx_obs_user_metric_time on observation o  (cost=0.58..66.69 rows=4 width=18) (actual time=0.004..0.004 rows=1 loops=100000)
                     Index Cond: ((user_id = u.user_id) AND (metric_id = 43) AND (date_time >= (now() - '24:00:00'::interval)))
                     Filter: (value < '60'::numeric)
                     Rows Removed by Filter: 4
 Planning Time: 20.424 ms
 Execution Time: 208.695 ms
(22 rows)
```