Advanced Databases

Phase 4- Indexing and query optimization

Project: Balkan geotourism

Bora Alili 231504

Esra Bekiri 231521

Marko Smilevski 231557

Fisnik Mamuti 231508

View 1: hotel booking stats view

1. Primary filter for the view hotel_booking_stats_view will be by the hotel_id

2. The initial time for execution for this view is this time 539ms. This time is acceptable so we have no need for indexing

```
[2026-05-17 20:10:21] advdb_202526l_prj_balk...o.public> SELECT *
                                                         FROM hotel_booking_stats_view
                                                         WHERE hotel_id = 1239
[2026-05-17 20:10:22] 1 row retrieved starting from 1 in 539 ms (execution: 203 ms, fetching: 336 ms)
```

3. No need for making an execution plan, because the time is acceptable. The time for the operations insert and update is

```
[2026-05-17 20:30:13] advdb_202526l_prj_balk...o.public> INSERT INTO hotels (name, star_rating, location_id)
                                                         VALUES ('Test Hotel5', 4, 1)
[2026-05-17 20:30:14] 1 row affected in 498 ms
```

```
                                                  update hotels set star_rating = 5 where id = 8374
 [2026-05-17 20:35:53] 5 rows retrieved starting from 1 in 1 s 302 ms (execution: 980 ms, fetching: 322 ms)
```

4. No need to change the query.
5. Execution time for the operations stays the same.

View 2: attraction popularity view

1. Primary filter for the view attraction_popularity_view will be by the attraction name
2. The initial time for execution for this view is 5s 41ms. The time is not acceptable, so we approach indexing.

```
                                                SELECT *
                                                FROM attraction_popularity_view
                                                WHERE name = '11 Oktomvri 46'
[2026-05-17 22:22:05] 16 rows retrieved starting from 1 in 5 s 41 ms (execution: 4 s 722 ms, fetching: 319 ms)
```

3. The slowest part is the parallel seq scan on attractions a

```
  ->  Nested Loop Left Join  (cost=0.43..231453.16 rows=944 width=43) (actual time=248.439..2694.369 rows=757 loops=5)
        ->  Parallel Seq Scan on attractions a  (cost=0.00..227326.00 rows=944 width=39) (actual time=29.459..331.285 rows=757 …
```

4.

```
[2026-05-17 23:47:02] advdb_202526l_prj_balk...o.public> create index idx_attraction_name on attractions(name)
[2026-05-17 23:48:29] completed in 1 m 27 s 81 ms
```

```
[2026-05-18 00:02:25] advdb_202526l_prj_balk...o.public> select *
                                                         from attraction_popularity_view where name ='144'
[2026-05-18 00:02:25] 1 row retrieved starting from 1 in 437 ms (execution: 107 ms, fetching: 330 ms)
```

```
[2026-05-18 00:05:14] advdb_202526l_prj_balk...o.public> INSERT INTO facultatives (name, description, location_id, trip_participant_id)
                                                         VALUES (
                                                                   'City Museum Tour',
                                                                   'Optional guided visit to the city museum',
                                                                   34032,
                                                                   4970644
                                                                 )
[2026-05-18 00:05:14] 1 row affected in 233 ms
```

```
[2026-05-18 00:06:43] advdb_202526l_prj_balk...o.public> UPDATE facultatives SET name = 'City Museum Tour', description = 'Optional guided visit to the city museum', location_id = 3, trip_partic
[2026-05-18 00:06:43] 1 row affected in 54 ms
```

5. After the indexing the times are acceptable.

View 3: flight revenue view

1. The primary filetr for this view is flight_routes_id

2. The initial time for execution is 439ms. This time is acceptable, so we have no need for indexing.

```
[2026-05-18 00:12:14] advdb_202526l_prj_balk...o.public> select * from flight_revenue_view where flight_route_id = 41
[2026-05-18 00:12:14] 1 row retrieved starting from 1 in 439 ms (execution: 89 ms, fetching: 350 ms)
```

3. No need for making an execution plan, because the time is acceptable. The time for the operations insert and update is

```
[2026-05-18 00:18:11] advdb_202526l_prj_balk...o.public> INSERT INTO flight_routes (from_airport_id, to_airport_id)
                                                         VALUES (1, 2)
[2026-05-18 00:18:12] 1 row affected in 91 ms
```

```
[2026-05-18 00:18:55] advdb_202526l_prj_balk...o.public> UPDATE flight_routes SET from_airport_id = 1, to_airport_id = 2 WHERE id = 30
[2026-05-18 00:18:55] 1 row affected in 32 ms
```

4. No need to change the query.
5. Execution time for the operations stays the same.

View 4: review photo engagement view

1. The primary filter for this view will be user_id, as well as review_id.
2. The initial time for execution for this view is this time 493. This time is acceptable so we have no need for indexing

```
[2026-05-18 00:22:41] advdb_202526l_prj_balk...o.public> select * from review_photo_engagement_view where review_id = 12 and user_id = 209447
[2026-05-18 00:22:42] 1 row retrieved starting from 1 in 493 ms (execution: 176 ms, fetching: 317 ms)
```

3. No need for making an execution plan, because the time is acceptable. The time for the operations insert and update is

```
[2026-05-18 00:26:40] advdb_202526l_prj_balk...o.public> INSERT INTO users (email, username, password_hash, created_at, name, discount_percentage)
                                                         VALUES ('test@email.com', 'testuser', 'hashed_password_here', NOW(), 'Test Name', 10)
[2026-05-18 00:26:40] 1 row affected in 180 ms
```

```
[2026-05-18 00:27:51] advdb_202526l_prj_balk...o.public> UPDATE users
                                                         SET email = 'testa@email.com',
                                                             username = 'testauser',
                                                             password_hash = 'hashed_password_here',
                                                             created_at = NOW(),
                                                             name = 'Test Name',
                                                             discount_percentage = 10
                                                         WHERE id = 1
[2026-05-18 00:27:51] 1 row affected in 29 ms
```

```
[2026-05-18 00:29:44] advdb_202526l_prj_balk...o.public> UPDATE reviews
                                                         SET comment = 'Great trip, everything was well organized!',
                                                             created_at = NOW(),
                                                             trip_participant_id = 1,
                                                             rating = 5
                                                         WHERE id = 1
[2026-05-18 00:29:44] 1 row affected in 28 ms
```

```
[2026-05-18 00:29:09] advdb_202526l_prj_balk...o.public> INSERT INTO reviews (comment, created_at, trip_participant_id, rating)
                                                         VALUES ('Great trip, everything was well organized!', NOW(), 1, 5)
[2026-05-18 00:29:09] 1 row affected in 66 ms
```

4. No need to change the query.
5. Execution time for the operations stays the same.

View 5: transport usage stats view

1. Primary filter for this view is transport_routes_id
2. The initial execution time for this view Is 1122 ms. This is not acceptable so we proceed with indexing

```
                                                         select *
                                                         from transport_usage_stats_view
[2026-05-18 00:45:31] 28 rows retrieved starting from 1 in 1 s 122 ms (execution: 806 ms, fetching: 316 ms)
```

3. The slowest part is the seq scan on flight_bookings, the time for insert/update are

```
    Hash Cond: (fb.flight_route_id = fr.id)
    ->  Seq Scan on flight_bookings fb  (cost=0.00..15310.00 rows=500000 width=22) (actual time=10.925..51.823 rows=500000 loops=1)
    ->  Hash  (cost=7.00..7.00 rows=400 width=8) (actual time=0.373..0.374 rows=401 loops=1)
```

```
[2026-05-18 00:59:57] advdb_202526l_prj_balk...o.public> UPDATE transport_routes
                                                         SET distance = 120,
                                                             estimated_time = 180,
                                                             from_station_id = 1,
                                                             to_station_id = 2
                                                         WHERE id = 1
[2026-05-18 00:59:57] 1 row affected in 29 ms
[2026-05-18 01:00:03] advdb_202526l_prj_balk...o.public> insert INTO transport_routes (distance, estimated_time, from_station_id, to_station_id)
                                                         VALUES (120, 180, 1, 2)
[2026-05-18 01:00:03] 1 row affected in 29 ms
```

4.

```
[2026-05-18 00:57:59] advdb_202526l_prj_balk...o.public> CREATE INDEX idx_flight_bookings_route_participant
                                                             ON flight_bookings (flight_route_id, trip_participant_id)
[2026-05-18 00:57:59] completed in 468 ms
```

```
[2026-05-18 00:58:12] advdb_202526l_prj_balk...o.public> CREATE INDEX idx_bus_bookings_route_participant
                                                             ON bus_bookings (route_id, trip_participant_id)
[2026-05-18 00:58:12] completed in 556 ms
```

5. The time needed for execution is now 367ms, which is acceptable.

```
[2026-05-18 01:01:43] advdb_202526l_prj_balk...o.public> select *
                                                         from transport_usage_stats_view where route_id = 12
[2026-05-18 01:01:44] 2 rows retrieved starting from 1 in 367 ms (execution: 34 ms, fetching: 333 ms)
```

6. The time needed for the operations insert and update after indexing is

```
[2026-05-18 01:03:48] advdb_202526l_prj_balk...o.public> insert INTO transport_routes (distance, estimated_time, from_station_id, to_station_id)
                                                         VALUES (120, 180, 1, 2)
[2026-05-18 01:03:48] 1 row affected in 291 ms
[2026-05-18 01:03:49] advdb_202526l_prj_balk...o.public> UPDATE transport_routes
                                                         SET distance = 120,
                                                             estimated_time = 180,
                                                             from_station_id = 1,
                                                             to_station_id = 2
                                                         WHERE id = 1
[2026-05-18 01:03:49] 1 row affected in 136 ms
```

View 6: trip duration stats view

1. The primary filter for this view is trip_id.
2. The initial execution time for this view Is 425 ms. This time is acceptable so we have no need for indexing.

```
[2026-05-18 01:09:16] advdb_202526l_prj_balk...o.public> select * from trip_duration_stats_view where trip_id = 67
[2026-05-18 01:09:17] 1 row retrieved starting from 1 in 425 ms (execution: 105 ms, fetching: 320 ms)
```

3. No need for making an execution plan, because the time is acceptable. The time for the operations insert and update is

```
[2026-05-18 01:12:04] advdb_202526l_prj_balk...o.public> INSERT INTO trips (name, description)
                                                         VALUES ('Summer Trip', 'A relaxing summer vacation trip')
[2026-05-18 01:12:04] 1 row affected in 72 ms
```

```
[2026-05-18 01:12:25] advdb_202526l_prj_balk...o.public> UPDATE trips
                                                             SET name = 'Summer Trip',
                                                                 description = 'A relaxing summer vacation trip'
                                                             WHERE id = 1
[2026-05-18 01:12:25] 1 row affected in 215 ms
```

4. No need to change the query.
5. Execution time for the operations stays the same.

View 7: trip execution pacing view

1. The primary filter for this view is trip_execution_id.
2. The initial execution time for this view Is 469 ms. This time is acceptable so we have no need for indexing.

```
2026-05-18 01:18:52] advdb_202526l_prj_balk...o.public> select * from trip_execution_pacing_view where trip_execution_id = 198642
2026-05-18 01:18:52] 1 row retrieved starting from 1 in 469 ms (execution: 138 ms, fetching: 331 ms)
```

3. No need for making an execution plan, because the time is acceptable. The time for the operations insert and update is

```
2026-05-18 01:23:37] advdb_202526l_prj_balk...o.public> INSERT INTO trip_executions (trip_id, start_date, end_date)
                                                        VALUES (1, '2026-06-01', '2026-06-10')
2026-05-18 01:23:37] 1 row affected in 157 ms
```

```
2026-05-18 01:25:39] advdb_202526l_prj_balk...o.public> UPDATE trip_executions
                                                        SET trip_id = 1,
                                                            start_date = '2026-06-01',
                                                            end_date = '2026-06-10'
                                                        WHERE id = 1
2026-05-18 01:25:39] 1 row affected in 147 ms
```

4. No need to change the query.
5. Execution time for the operations stays the same.

View 8: trip execution stats view

1. The primary filter for this view is trip_execution_id.
2. The initial execution time for this view Is 1530 ms. This time is not acceptable so we proceed with indexing.

```
                                select * from trip_execution_stats_view where trip_execution_id = 124121
ows retrieved starting from 1 in 1 s 530 ms (execution: 1 s 215 ms, fetching: 315 ms)
```

3. The slow part is the parallel seq scan on trip_participants. The time for operations insert/update

```
5       Sort Method: quicksort  Memory: 25kB
6       ->  Nested Loop Left Join  (cost=12.54..157.18 rows=12 width=56) (actual time=0.461..0.807 rows=11 loops=1)
7             ->  Nested Loop Left Join  (cost=12.11..55.75 rows=12 width=52) (actual time=0.376..0.553 rows=11 loops=1)
8                   ->  Hash Right Join  (cost=11.68..14.99 rows=1 width=44) (actual time=0.283..0.295 rows=1 loops=1)
9                         Hash Cond: (tl.trip_id = t.id)
10                        ->  Seq Scan on trip_locations tl  (cost=0.00..2.95 rows=95 width=8) (actual time=0.018..0.026 rows=98 loops=1)
11                        ->  Hash  (cost=11.67..11.67 rows=1 width=48) (actual time=0.220..0.222 rows=1 loops=1)
12                              Buckets: 1024  Batches: 1  Memory Usage: 9kB
13                              ->  Hash Join  (cost=8.46..11.67 rows=1 width=48) (actual time=0.211..0.215 rows=1 loops=1)
14                                    Hash Cond: (t.id = te.trip_id)
15                                    ->  Seq Scan on trips t  (cost=0.00..2.95 rows=95 width=32) (actual time=0.016..0.026 rows=98 loops=1)
16                                    ->  Hash  (cost=8.44..8.44 rows=1 width=20) (actual time=0.128..0.129 rows=1 loops=1)
17                                          Buckets: 1024  Batches: 1  Memory Usage: 9kB
18                                          ->  Index Scan using trip_executions_pkey on trip_executions te  (cost=0.42..8.44 rows=1 width=20) (actual t
19                                                Index Cond: (id = 123114)
20                  ->  Index Only Scan using idx_trip_participants_exec_id on trip_participants tp  (cost=0.43..40.64 rows=12 width=12) (actual time=0.
```

```
[2026-05-18 01:37:29] advdb_202526l_prj_balk...o.public> UPDATE trip_participants
                                                         SET trip_execution_id = 1,
                                                             user_id = 10,
                                                             passport_number = 'MK1234567',
                                                             status = 'completed'
                                                         WHERE id = 1
[2026-05-18 01:37:29] 1 row affected in 30 ms
[2026-05-18 01:37:33] advdb_202526l_prj_balk...o.public> INSERT INTO trip_participants (trip_execution_id, user_id, passport_number, status)
                                                         VALUES (1, 10, 'MK1234567', 'completed')
[2026-05-18 01:37:34] 1 row affected in 28 ms
```

4.

```
[2026-05-18 01:39:04] advdb_202526l_prj_balk...o.public> CREATE INDEX idx_trip_participants_execution
                                                           ON trip_participants (trip_execution_id)

[2026-05-18 01:39:19] completed in 15 s 80 ms
```

5. The time needed for execution is now 1133ms.

```
[2026-05-18 01:43:23] advdb_202526l_prj_balk...o.public> select * from trip_execution_stats_view where trip_execution_id = 12314
[2026-05-18 01:43:24] 1 row retrieved starting from 1 in 1 s 333 ms (execution: 1 s 19 ms, fetching: 314 ms)
```

6. The time needed for update and insert after indexing

```
[2026-05-18 01:45:24] advdb_202526l_prj_balk...o.public> INSERT INTO trip_participants (trip_execution_id, user_id, passport_number, status)
                                                         VALUES (1, 10, 'MK1234567', 'completed')
[2026-05-18 01:45:24] 1 row affected in 31 ms
[2026-05-18 01:45:27] advdb_202526l_prj_balk...o.public> UPDATE trip_participants
                                                         SET trip_execution_id = 1,
                                                             user_id = 10,
                                                             passport_number = 'MK1234567',
                                                             status = 'completed'
                                                         WHERE id = 1

[2026-05-18 01:45:27] 1 row affected in 153 ms
```

View 9: trip locations by day view

1. The primary filter for this table is trip_locations_id
2. The in initial execution time for this view Is 399 ms. This time is acceptable so we have no need for indexing

```
[2026-05-18 01:50:23] advdb_202526l_prj_balk...o.public> select  * from trip_locations_by_day_view where day_number = 4
[2026-05-18 01:50:23] 1 row retrieved starting from 1 in 399 ms (execution: 32 ms, fetching: 367 ms)
```

3. The slow part is the parallel seq scan on trip_participants. The time for operations insert/update

```
[2026-05-18 01:58:40] advdb_202526l_prj_balk...o.public> INSERT INTO trip_locations (trip_id, location_id, visit_order, day_number)
                                                         VALUES (1, 5, 112, 1)
[2026-05-18 01:58:40] 1 row affected in 582 ms
[2026-05-18 01:58:47] advdb_202526l_prj_balk...o.public> UPDATE trip_locations
                                                         SET trip_id = 1,
                                                             location_id = 5,
                                                             visit_order = 1,
                                                             day_number = 1
                                                         WHERE id = 1

[2026-05-18 01:58:48] 1 row affected in 753 ms
```

4. No need to change the query.
5. Execution time for the operations stays the same.

View 10: trip participant count view

1. The primary filter for this view is trip_id
2. The in initial execution time for this view Is 2352 ms. This time is not acceptable so we proceed with indexing.

```
[2026-05-18 02:04:52] advdb_202526l_prj_balk...o.public> select * from trip_participant_count_view where trip_id = 41
[2026-05-18 02:04:54] 1 row retrieved starting from 1 in 2 s 352 ms (execution: 2 s 34 ms, fetching: 318 ms)
```

3. The slowest part in the operation is Parallel Seq Scan on trip_participants. The time for insert/update is

```
->  Nested Loop Left Join  (cost=0.43..120974.51 rows=35956 width=12) (actual time=5210.245..5369.343 rows=28946 loops=4)
      ->  Parallel Seq Scan on trip_executions te  (cost=0.00..16771.26 rows=3269 width=12) (actual time=5210.110..5230.207 rows=2632 loop…
            Filter: (trip_id = 41)
```

```
[2026-05-18 02:09:05] advdb_202526l_prj_balk...o.public> INSERT INTO trips (name, description)
                                                         VALUES ('Summer Trip', 'A relaxing summer vacation trip')
[2026-05-18 02:09:05] 1 row affected in 44 ms
[2026-05-18 02:09:09] advdb_202526l_prj_balk...o.public> UPDATE trips
                                                         SET name = 'Summer Trip',
                                                             description = 'A relaxing summer vacation trip'
                                                         WHERE id = 1
[2026-05-18 02:09:10] 1 row affected in 256 ms
```

4.

```
[2026-05-18 02:10:05] advdb_202526l_prj_balk...o.public> CREATE INDEX idx_trip_executions_trip_id
                                                                 ON trip_executions (trip_id)
[2026-05-18 02:10:05] completed in 442 ms
[2026-05-18 02:10:08] advdb_202526l_prj_balk...o.public> CREATE INDEX idx_trip_participants_exec_id
                                                                 ON trip_participants (trip_execution_id, id)
[2026-05-18 02:10:51] completed in 43 s 421 ms
```

5. The time needed for execution is now 590ms.

```
[2026-05-19 01:33:53] advdb_202526l_prj_balk...o.public> explain  analyze  select * from trip_participant_count_view where trip_id = 41
[2026-05-19 01:33:54] 24 rows retrieved starting from 1 in 590 ms (execution: 273 ms, fetching: 317 ms)
```

6. The time needed for update and insert after indexing

```
[2026-05-18 02:23:28] advdb_202526l_prj_balk...o.public> INSERT INTO trips (name, description)
                                                         VALUES ('Summmaer Trip', 'A relaxing summer vacation trip')
[2026-05-18 02:23:28] 1 row affected in 30 ms
[2026-05-18 02:23:30] advdb_202526l_prj_balk...o.public> UPDATE trips
                                                         SET name = 'Summer Trgip',
                                                             description = 'A relaxing summer vacation trip'
                                                         WHERE id = 1
[2026-05-18 02:23:30] 1 row affected in 30 ms
```

View 11: trip participant status view

1. The primary filter for this view is trip_participant_id

2. The initial execution time for this view is 1868ms. This time is not acceptable so we proceed with indexing

```
[2026-05-18 02:24:46] advdb_202526l_prj_balk...o.public> select * from trip_participant_status_view where status = 'confirmed'
[2026-05-18 02:24:48] 1 row retrieved starting from 1 in 1 s 868 ms (execution: 1 s 552 ms, fetching: 316 ms)
```

3. The slowes part in this operation is Parallel Seq Scan. The time needed for insert/update is

```
->  Partial GroupAggregate  (cost=0.00..135334.34 rows=1 width=17) (actual time=388.801..388.802 rows=1 loops=5)
      ->  Parallel Seq Scan on trip_participants  (cost=0.00..134785.02 rows=219725 width=9) (actual time=26.408..377.741 rows=176000 loops=5)
            Filter: (status = 'cancelled'::text)
            Rows Removed by Filter: 2024001
```

```
[2026-05-18 02:30:46] advdb_202526l_prj_balk...o.public> INSERT INTO trip_participants (trip_execution_id, user_id, passport_number, status)
                                                         VALUES (1, 10, 'MK1234567', 'completed')
[2026-05-18 02:30:46] 1 row affected in 49 ms
[2026-05-18 02:30:47] advdb_202526l_prj_balk...o.public> UPDATE trip_participants
                                                         SET trip_execution_id = 1,
                                                             user_id = 10,
                                                             passport_number = 'MK1234567',
                                                             status = 'completed'
                                                         WHERE id = 1
[2026-05-18 02:30:48] 1 row affected in 27 ms
```

4.

```
[2026-05-18 02:32:10] advdb_202526l_prj_balk...o.public> CREATE INDEX idx_trip_participants_status
                                                                 ON trip_participants (status)
[2026-05-18 02:32:21] completed in 11 s 747 ms
```

5. The time needed for execution is now 528ms.

```
[2026-05-18 02:40:33] advdb_202526l_prj_balk...o.public> select * from trip_participant_status_view where status = 'confirmed'
[2026-05-18 02:40:34] 1 row retrieved starting from 1 in 528 ms (execution: 205 ms, fetching: 323 ms)
```

6. The time needed for insert/update after indexing is

```
[2026-05-18 02:41:23] advdb_202526l_prj_balk...o.public> INSERT INTO trip_participants (trip_execution_id, user_id, passport_number, status)
                                                         VALUES (1, 10, 'MK1234567', 'completed')
[2026-05-18 02:41:23] 1 row affected in 31 ms
[2026-05-18 02:41:24] advdb_202526l_prj_balk...o.public> UPDATE trip_participants
                                                         SET trip_execution_id = 1,
                                                             user_id = 10,
                                                             passport_number = 'MK1234567',
                                                             status = 'completed'
                                                         WHERE id = 1
[2026-05-18 02:41:24] 1 row affected in 33 ms
```

View 12: user trip review stats view

1. The primary filter for this view is user_id.
2. The initial execution time for this view is 1604ms. This time is not acceptable so we proceed with indexing

```
[2026-05-18 02:45:46] advdb_202526l_prj_balk...o.public> select * from user_trip_review_stats_view where user_id = 219232
[2026-05-18 02:45:48] 1 row retrieved starting from 1 in 1 s 604 ms (execution: 1 s 273 ms, fetching: 331 ms)
```

3. The slowest part in the operation is the Parallel Seq Scan on trip_participants. The time for insert/update

```
->  Nested Loop Left Join  (cost=0.42..134810.37 rows=3 width=20) (actual time=61.605..367.468 rows=2 loops=5)
      ->  Parallel Seq Scan on trip_participants tp  (cost=0.00..134785.02 rows=3 width=16) (actual time=35.351..289.691 rows=2 loop…
            Filter: (user_id = 21922)
            Rows Removed by Filter: 2199999
```

```
[2026-05-18 02:52:44] advdb_202526l_prj_balk...o.public> INSERT INTO users (email, username, password_hash, created_at, name, discount_percentage)
                                                         VALUES ('teasta@email.com', 'testuserasd', 'hashed_password_here', NOW(), 'Test Name', 10)
[2026-05-18 02:52:45] 1 row affected in 139 ms
[2026-05-18 02:52:50] advdb_202526l_prj_balk...o.public> UPDATE users
                                                         SET email = 'testa@email.com',
                                                             username = 'testauser',
                                                             password_hash = 'hashed_password_here',
                                                             created_at = NOW(),
                                                             name = 'Test Name',
                                                             discount_percentage = 10
                                                         WHERE id = 1
[2026-05-18 02:52:50] 1 row affected in 85 ms
```

4.

```
[2026-05-18 02:53:48] advdb_202526l_prj_balk...o.public> CREATE INDEX idx_tp_user_exec
                                                                 ON trip_participants (user_id, trip_execution_id)
[2026-05-18 02:54:17] completed in 29 s 294 ms
```

5. The time needed for execution is now 344ms.

```
[2026-05-18 02:56:09] advdb_202526l_prj_balk...o.public> select * from user_trip_review_stats_view where user_id = 219232
[2026-05-18 02:56:09] 1 row retrieved starting from 1 in 344 ms (execution: 31 ms, fetching: 313 ms)
```

6. The time needed for insert/update after indexing

```
[2026-05-18 02:57:30] advdb_202526l_prj_balk...o.public> INSERT INTO users (email, username, password_hash, created_at, name, discount_percentage)
                                                         VALUES ('teasta@emaail.com', 'testufserasd', 'hashed_password_here', NOW(), 'Test Name', 10)
[2026-05-18 02:57:30] 1 row affected in 28 ms
[2026-05-18 02:57:32] advdb_202526l_prj_balk...o.public> UPDATE users
                                                         SET email = 'testa@email.com',
                                                             username = 'testauser',
                                                             password_hash = 'hashed_password_here',
                                                             created_at = NOW(),
                                                             name = 'Test Name',
                                                             discount_percentage = 10
                                                         WHERE id = 1
[2026-05-18 02:57:32] 1 row affected in 30 ms
```