Screenshot 1 (top):

```
270    select * from flights;
271
272    --3
273    CREATE INDEX idx_haha3 ON flights(departing_airport_id, arriving_airport_id);
274
275    --4
276    DROP INDEX IF EXISTS idx_haha3;
277
278    EXPLAIN ANALYZE SELECT *
279    FROM flights
280    WHERE departing_airport_id=50 AND arriving_airport_id=100;
281
282
283
284
```

QUERY PLAN

```
1   Seq Scan on flights   (cost=0.00..6.00 rows=1 width=74) (actual time=0.059..0.059 rows=0 loops=1)
2     Filter: ((departing_airport_id = 50) AND (arriving_airport_id = 100))
3     Rows Removed by Filter: 200
4   Planning Time: 2.006 ms
5   Execution Time: 0.088 ms
```

5 rows

Screenshot 2 (bottom):

```
270    select * from flights;
271
272    --3
273    CREATE INDEX idx_haha3 ON flights(departing_airport_id, arriving_airport_id);
274
275    --4
276    DROP INDEX IF EXISTS idx_haha3;
277
278    EXPLAIN ANALYZE SELECT *
279    FROM flights
280    WHERE departing_airport_id=50 AND arriving_airport_id=100;
281
282
283
284
```

QUERY PLAN

```
1   Seq Scan on flights   (cost=0.00..6.00 rows=1 width=74) (actual time=0.065..0.065 rows=0 loops=1)
2     Filter: ((departing_airport_id = 50) AND (arriving_airport_id = 100))
3     Rows Removed by Filter: 200
4   Planning Time: 1.849 ms
5   Execution Time: 0.088 ms
```

5 rows

```sql
228    --8
229    SELECT
230        tablename,
231        indexname,
232        indexdef
233    FROM
234        pg_indexes
235    WHERE
236        tablename = 'passengers';
237
238    DROP INDEX IF EXISTS idx_passport_number;
239    DROP INDEX IF EXISTS idx_passengers_composite_search;
```

Services

Database
  postgres@localhost
    console_4
    console_7 442 ms
    console_6
    console_5
    console_2
    console

Output --8 --8 2

tablename	indexname	indexdef
passengers	passengers_pkey	CREATE UNIQUE INDEX passengers_pkey ON public.passengers USING btree (pass...

1 row

Database Consoles > postgres@localhost > console_7   233:5 CRLF UTF-8 4 spaces

---

```sql
201    WHERE departing_airport_id=50 AND arriving_airport_id=100;
202
203
204    CREATE UNIQUE INDEX idx_passport_number ON passengers(passport_number);
205
206    SELECT pg_indexes.tablename, pg_indexes.indexname, pg_indexes.indexdef
207    FROM pg_indexes
208    WHERE tablename= 'passengers';
209
210    INSERT INTO passengers(first_name, last_name, date_of_birth, gender, country_of_citizenship, country_of_residence, passport_number)
211    VALUES
212    ( first_name 'Dmitriy', last_name 'Hzhz', date_of_birth '1985-11-01', gender 'Male', country_of_citizenship 'Russia', country_of_residence 'Russia', passport_number 'P999999'),
213    ( first_name 'Nazerke', last_name 'Hzhz', date_of_birth '1985-11-01', gender 'Female', country_of_citizenship 'Kazakhstan', country_of_residence 'Kazakhstan', passport_number 'P999993'
214
215
```

Services

Database
  postgres@localhost
    console_4
    console_7 442 ms
    console_6
    console_5
    console_2
    console

Output --8 --8 2

tablename	indexname	indexdef
passengers	passengers_pkey	CREATE UNIQUE INDEX passengers_pkey ON public.passengers USING btree (pass...

1 row

Database Consoles > postgres@localhost > console_7   204:56 CRLF UTF-8 4 spaces

```
214
215
216    --7
217    CREATE INDEX idx_passengers_composite_search
218    ON passengers(first_name, last_name, date_of_birth, gender, country_of_citizenship);
219
220  ✓ EXPLAIN ANALYZE SELECT *
221    FROM passengers
222    WHERE country_of_citizenship='Philippines'
223    AND date_of_birth BETWEEN '19984-01-01' AND '1984-12-31';
224    --When you run EXPLAIN ANALYZE on your query, you will see the following plan:
225    --Seq Scan on passengers (Sequential Scan)
226    --This result means that PostgreSQL did not use your new index (idx_passengers_composite_search). Instead, it chose to read the entire passengers table row by row (all 200
227
```

Services

- Database
  - postgres@localhost
    - console_4
    - console_7 459 ms
    - console_6
    - console_5
    - console_2
    - console

Output | postgres.pg_catalog.pg_indexes | Result 27

	QUERY PLAN
1	Seq Scan on passengers (cost=0.00..6.50 rows=1 wi…
2	Filter: ((date_of_birth >= '19984-01-01'::date) …
3	Rows Removed by Filter: 200
4	Planning Time: 3.373 ms
5	Execution Time: 0.048 ms

5 rows

Database Consoles > postgres@localhost > console_7   227:1 CRLF UTF-8 4 spaces