

The screenshot shows a PostgreSQL database management interface with the following components:

- Database Explorer**: Shows the database structure for `postgres@localhost`. It includes tables `booking`, `booking_flight`, `flights` (with 12 columns), `passengers`, and `security_check`. It also lists `keys` (1), `foreign keys` (3), and `indexes` (5).
- Editor**: A code editor window titled `1.sql` containing the following SQL command:

```
CREATE INDEX idx_flights_actual_departure
ON flights (act_departure_time);
```
- Files**: A file explorer showing the file `1.sql` located at `C:\Users\amina\Desktop\lab 7`.
- Services**: A log of database transactions. The log shows the execution of the `CREATE INDEX` command and its completion details.
- System Tray**: Shows the date and time as `11.11.2025 23:37`, along with icons for battery, signal, and other system status.

Editor Content (1.sql):

```
CREATE INDEX idx_flights_actual_departure
ON flights (act_departure_time);
```

Services Log:

```
[2025-11-11 23:36:25] Connected to lab2_db
[2025-11-11 23:36:25] completed in 2 s 109 ms
[2025-11-11 23:37:16] lab2_db> CREATE INDEX idx_flights_actual_departure
[2025-11-11 23:37:16] ON flights (act_departure_time)
[2025-11-11 23:37:16] completed in 61 ms
```

Bottom Status Bar:

lab 7 > 1.sql 2:33 CRLF UTF-8 4 spaces ⚡ 23:37
1 5°C Clear ENG WiFi 11.11.2025

The screenshot shows a PostgreSQL database management interface with several panes:

- Database Explorer**: Shows the schema of the `flights` table, including columns like `flight_id`, `sch_departure_time`, and `airline_id`.
- Editor**: A code editor window titled `1.sql` containing two `CREATE INDEX` statements. The first statement has been executed successfully (indicated by a green checkmark). The second statement is currently being run, indicated by a yellow progress bar.
- Services**: A log pane showing the execution of the SQL statements. The first statement completed in 2 ms, and the second completed in 13 ms.
- Files**: A file explorer pane showing the file `1.sql` is located in the directory `C:\Users\amina\Desktop\lab 7`.

```
1 CREATE INDEX idx_flights_actual_departure
ON flights (act_departure_time);
2
3
4
5
6 ✓ CREATE UNIQUE INDEX ux_flights_airline_sched
ON flights (airline_id, sch_departure_time);
7
8
```

```
[2025-11-11 23:36:25] completed in 2 s 109 ms
[2025-11-11 23:37:16] lab2_db> CREATE INDEX idx_flights_actual_departure
ON flights (act_departure_time)
[2025-11-11 23:37:16] completed in 61 ms
[2025-11-11 23:38:06] lab2_db.public> CREATE UNIQUE INDEX ux_flights_airline_sched
ON flights (airline_id, sch_departure_time)
[2025-11-11 23:38:06] completed in 13 ms
```

Bottom status bar: 7:45 CRLF UTF-8 4 spaces 23:38 11.11.2025

The screenshot shows a PostgreSQL database management interface with several panels:

- Database Explorer**: Shows the schema of the `flights` table, including columns like `flight_id`, `sch_departure_time`, and `arriving_airport_id`.
- Editor**: A code editor window titled `1.sql` containing the following SQL code:

```
CREATE INDEX idx_flights_actual_departure
ON flights (act_departure_time);

CREATE UNIQUE INDEX ux_flights_airline_sched
ON flights (airline_id, sch_departure_time);

CREATE INDEX idx_flights_dep_arr
ON flights (departing_airport_id, arriving_airport_id);
```
- Files**: A file browser showing the file `1.sql` under the path `Lab3\lab 7`.
- Services**: A log of database transactions, showing the execution of the three index creation statements.

Editor Content (1.sql):

```
1 CREATE INDEX idx_flights_actual_departure
2 ON flights (act_departure_time);
3
4
5
6 CREATE UNIQUE INDEX ux_flights_airline_sched
7 ON flights (airline_id, sch_departure_time);
8
9
10
11 CREATE INDEX idx_flights_dep_arr
12 ON flights (departing_airport_id, arriving_airport_id);
```

Services Log:

```
[2025-11-11 23:37:16] completed in 61 ms
[2025-11-11 23:38:06] lab2_db.public> CREATE UNIQUE INDEX ux_flights_airline_sched
[2025-11-11 23:38:06] ON flights (airline_id, sch_departure_time)
[2025-11-11 23:38:06] completed in 13 ms
[2025-11-11 23:38:55] lab2_db.public> CREATE INDEX idx_flights_dep_arr
[2025-11-11 23:38:55] ON flights (departing_airport_id, arriving_airport_id)
[2025-11-11 23:38:55] completed in 18 ms
```

Bottom Status Bar:

lab 7 > 1.sql 12:56 CRLF UTF-8 4 spaces ⚡ 🔔

The screenshot shows a PostgreSQL database management interface with the following components:

- Database Explorer**: Shows the schema of the `flights` table, including columns like `flight_id`, `sch_departure_time`, and `act_departure_time`.
- Query Editor**: An open SQL file (`1.sql`) containing the following code:

```
EXPLAIN ANALYZE
SELECT f.flight_id
FROM flights f
WHERE f.act_departure_time >= now() - INTERVAL '30 days';

SET enable_indexscan = off;
EXPLAIN ANALYZE
SELECT f.flight_id
FROM flights f
WHERE f.act_departure_time >= now() - INTERVAL '30 days';

SET enable_indexscan = on;
```
- File Explorer**: Shows the file structure under `Lab3` and `lab 7`, including the `1.sql` file.
- Services**: Shows a transaction history with one entry: `1.sql 1 s 279 ms`.
- Output**: Displays the results of the EXPLAIN ANALYZE statements, including the query plan and execution statistics:1 FILTER: (act_departure_time >= (now() - 30 days)::interval)
2 Rows Removed by Filter: 20
3 Planning Time: 8.368 ms
4 Execution Time: 0.093 ms

The screenshot shows a PostgreSQL database management interface with the following components:

- Database Explorer**: Shows the schema of the `flights` table, including columns like `flight_id`, `sch_departure_time`, and `act_departure_time`.
- Query Editor**: An open SQL file (`1.sql`) containing the following code:

```
13
14
15
16
17 EXPLAIN ANALYZE
18 SELECT f.flight_id
19 FROM flights f
20 WHERE f.act_departure_time >= now() - INTERVAL '30 days';
21
22 SET enable_indexscan = off;
23 EXPLAIN ANALYZE
24 SELECT f.flight_id
25 FROM flights f
26 WHERE f.act_departure_time >= now() - INTERVAL '30 days';
27
28 SET enable_indexscan = on;
```
- Files**: A file browser showing the file `1.sql` under the path `Lab3\lab 7`.
- Services**: A panel showing the execution of the query from the file, with a duration of `465 ms`.
- Output**: A results panel showing the query plan and execution statistics:

1	FILTER: (act_departure_time >= now() - 30 days ::interval)
2	Rows Removed by Filter: 20
3	Planning Time: 0.243 ms
4	Execution Time: 0.061 ms

The screenshot shows a PostgreSQL database management interface with the following components:

- Database Explorer**: Shows the database schema with tables `booking`, `booking_flight`, and `flights`. The `flights` table has 12 columns: `flight_id`, `sch_departure_time`, `sch_arrival_time`, `departing_airport_id`, `arriving_airport_id`, `departing_gate`, and `arriving_gate`.
- Query Editor (1.sql)**: Contains the following SQL code:

```
SET enable_indexscan = on;
EXPLAIN (ANALYZE, BUFFERS)
SELECT f.flight_id
FROM flights f
WHERE f.departing_airport_id = :dep_id
    AND f.arriving_airport_id = :arr_id;
```
- Files**: Shows the file structure under `Lab3` and `lab 7`, including the `1.sql` file.
- Services**: Shows a transaction list with one entry: `1.sql 509 ms`.
- Output**: Displays the `QUERY PLAN` for the executed query:

```
1 Seq Scan on flights f  (cost=0.00..1.30 rows=1 width=4) (actual time=0.023..0.024 rows=0 loops=1)
2   Filter: ((departing_airport_id = 1) AND (arriving_airport_id = 2))
3   Rows Removed by Filter: 20
4   Buffers: shared hit=1
5 Planning:
6   Buffers: shared hit=5
7 Planning Time: 0.286 ms
8 Execution Time: 0.047 ms
```
- Bottom Status Bar**: Shows the file name `1.sql`, line count `36:40`, encoding `CRLF`, character set `UTF-8`, and indentation `4 spaces`.

Lab3 Version control

Database Explorer

1.sql

postgres@localhost

- > booking
- > booking_flight
- < flights
 - columns 12
 - flight_id integer = nextval('f...
 - sch_departure_time time
 - sch_arrival_time timestamp
 - departing_airport_id integer
 - arriving_airport_id integer
 - departing_gate text
 - arriving_gate varchar(50)
 - airline_id integer
 - act_departure_time timestamp
 - act_arrival_time timestamp
 - created_at timestamp = now()
 - updated_at timestamp = now()

keys 1

CREATE INDEX idx_passengers_name_dob_citizenship
ON passengers (first_name, last_name, date_of_birth, country_of_citizenship);

EXPLAIN (ANALYZE, BUFFERS)
SELECT passenger_id,
 first_name,
 last_name,
 date_of_birth
FROM passengers
WHERE country_of_citizenship = 'Philippines'
 AND date_part('year', date_of_birth) = 1984
 AND first_name ILIKE 'A%'
 AND last_name ILIKE 'B%';

Services

Tx

+ Database

postgres@localhost

1.sql 464 ms

Output Result 14

QUERY PLAN

1 Seq Scan on passengers (cost=0.00..1.88 rows=1 width=158) (actual time=0.770..0.771 rows=0 loops=1)
Filter: ((first_name ~~* 'A%'::text) AND ((last_name)::text ~~* 'B%'::text) AND ((country_of_citizenship)::text ~~* 'Philippines'::text))
Rows Removed by Filter: 35
Buffers: shared hit=1
Planning:
Buffers: shared hit=61 read=1
Planning Time: 5.060 ms

8 rows

lab 7 > 1.sql

70:29 CRLF UTF-8 4 spaces

5°C Clear

23:45 11.11.2025

Files

Lab3 C:\Users\amina\Desktop\KBTU\DB\Lab3

lab 7 C:\Users\amina\Desktop\lab 7

1.sql[postgres@localhost]

Scratches and Consoles

23:45 11.11.2025

The screenshot displays a PostgreSQL database management interface with several panels:

- Database Explorer**: Shows the schema of the `flights` table, including columns like `flight_id`, `sch_departure_time`, and `arriving_airport_id`.
- File Explorer**: Shows the file structure under `Lab3` and `lab 7`, including the `1.sql` file.
- Code Editor**: The `1.sql` file contains SQL code for creating an index and inserting data into the `passengers` table. An error message is shown at line 49 indicating a unique constraint violation for the passport number.
- Output**: Shows the results of the transaction, including the insertion of a new passenger record and the subsequent error message.

```
1.sql
1. WHERE f.departing_airport_id = :dep_id
2. AND f.arriving_airport_id = :arr_id;
3.
4. CREATE UNIQUE INDEX ux_passengers_passport
5. ON passengers (passport_number);
6.
7. INSERT INTO passengers (first_name, last_name, date_of_birth, gender,
8.                         country_of_citizenship, country_of_residence, passport_number,
9.                         created_at, updated_at)
10.                        VALUES ( first_name 'Test', last_name 'User1', date_of_birth '1990-05-05', gender 'Male', col
11.                               created_at now(), updated_at now());
12.
13. ! INSERT INTO passengers (first_name, last_name, date_of_birth, gender,
14.                         country_of_citizenship, country_of_residence, passport_number,
15.                         created_at, updated_at)
16.                        VALUES ( first_name 'Test', last_name 'User2', date_of_birth '1992-06-06', gender 'Female', col
17.                               created_at now(), updated_at now());
18.
19. [23505] ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "unique_passport"
20. Подробности: Ключ "(passport_number)=(KZ123456)" уже существует.
```

Services: Shows the transaction history, with the most recent entry being the execution of `1.sql`.

Output pane content:

```
Output Result 8
now(), now()
[2025-11-11 23:42:47] 1 row affected in 35 ms
[2025-11-11 23:43:04] lab2_db.public> INSERT INTO passengers (first_name, last_name, date_of_birth, gender,
[2025-11-11 23:43:04]                                         country_of_citizenship, country_of_residence, passport_number,
[2025-11-11 23:43:04]                                         created_at, updated_at)
[2025-11-11 23:43:04]                                         VALUES ('Test', 'User2', '1992-06-06', 'Female', 'Kazakhstan', 'Kazakhstan', 'I
[2025-11-11 23:43:04]                               now(), now())
[2025-11-11 23:43:04] [23505] ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "unique_passport"
[2025-11-11 23:43:04] Подробности: Ключ "(passport_number)=(KZ123456)" уже существует.
```

Bottom Status Bar: Shows the file name `1.sql`, line count `49:1`, encoding `CRLF`, character set `UTF-8`, and other status indicators.

```
DROP INDEX IF EXISTS ux_passengers_passport;  
DROP INDEX IF EXISTS idx_passengers_name_dob_citizenship;
```

Output Result 15

```
          AND first_name ILIKE 'A%'  
          AND last_name ILIKE 'B%'  
[2025-11-11 23:45:46] 8 rows retrieved starting from 1 in 429 ms (execution: 19 ms, fetching: 410 ms)  
[2025-11-11 23:46:19] lab2_db.public> SELECT * FROM pg_indexes WHERE tablename = 'passengers'  
[2025-11-11 23:46:19] 4 rows retrieved starting from 1 in 458 ms (execution: 37 ms, fetching: 421 ms)  
[2025-11-11 23:46:34] lab2_db.public> DROP INDEX IF EXISTS ux_passengers_passport  
[2025-11-11 23:46:34] completed in 17 ms  
[2025-11-11 23:46:39] lab2_db.public> DROP INDEX IF EXISTS idx_passengers_name_dob_citizenship  
[2025-11-11 23:46:39] completed in 10 ms
```