

create database

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The top bar includes icons for file, database, query, and help, along with tabs for 'test_DB' and 'Version control'. The main area is divided into several panes:

- Database Explorer** pane on the left lists databases: 'postgres@localhost' (selected, 2 items), 'lab2_db' (0 of 3), 'postgres' (1 of 3), and 'Server Objects'.
- Object Explorer** pane at the top right shows a tree structure for 'postgres.public' and 'postgres@localhost'.
- SQL Editor** pane in the center contains a query window titled 't.sql'. The query is: 'CREATE DATABASE lab2_db;'. The status bar indicates '1 ✓' (one successful execution).
- Files** pane on the right shows the file structure: 'test_DB' (C:\Users\amina\...), 't.sql[postgres@localhost]', and 'Scratches and Consoles'.

The overall theme is dark, and the UI elements are clearly legible.

create tables

The image shows a code editor interface with four tabs, each containing a portion of an SQL script for creating tables. The tabs are labeled 't.sql' and are part of a 'Version control' system.

Table Definitions:

- airline_info:** Contains columns for airline_id (SERIAL PRIMARY KEY), airline_code (VARCHAR(30) NOT NULL), airline_name (VARCHAR(50) NOT NULL), airline_country (VARCHAR(50) NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), and updated_at (TIMESTAMP NOT NULL DEFAULT now()).
- airport:** Contains columns for airport_id (SERIAL PRIMARY KEY), airport_name (VARCHAR(50) NOT NULL), country (VARCHAR(50) NOT NULL), state (VARCHAR(50) NOT NULL), city (VARCHAR(50) NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), and updated_at (TIMESTAMP NOT NULL DEFAULT now()).
- baggage_check:** Contains columns for baggage_check_id (SERIAL PRIMARY KEY), check_result (VARCHAR(50) NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), updated_at (TIMESTAMP NOT NULL DEFAULT now()), booking_id (INT NOT NULL), and passenger_id (INT NOT NULL).
- baggage:** Contains columns for baggage_id (SERIAL PRIMARY KEY), weight_in_kg (DECIMAL(4,2) NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), updated_at (TIMESTAMP NOT NULL DEFAULT now()), and booking_id (INT NOT NULL).
- boarding_pass:** Contains columns for boarding_pass_id (SERIAL PRIMARY KEY), booking_id (INT NOT NULL), seat (VARCHAR(50) NOT NULL), boarding_time (TIMESTAMP NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), and updated_at (TIMESTAMP NOT NULL DEFAULT now()).
- booking:** Contains columns for booking_id (SERIAL PRIMARY KEY), flight_id (INT NOT NULL), passenger_id (INT NOT NULL), booking_platform (VARCHAR(50) NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), updated_at (TIMESTAMP NOT NULL DEFAULT now()), status (VARCHAR(50) NOT NULL), and price (DECIMAL(7,2) NOT NULL).
- flights:** Contains columns for flight_id (SERIAL PRIMARY KEY), sch_departure_time (TIMESTAMP NOT NULL), sch_arrival_time (TIMESTAMP NOT NULL), departing_airport_id (INT NOT NULL), arriving_airport_id (INT NOT NULL), departing_gate (VARCHAR(50) NOT NULL), arriving_gate (VARCHAR(50) NOT NULL), airline_id (INT NOT NULL), act_departure_time (TIMESTAMP NOT NULL), act_arrival_time (TIMESTAMP NOT NULL), and created_at (TIMESTAMP NOT NULL DEFAULT now()).
- passengers:** Contains columns for passenger_id (SERIAL PRIMARY KEY), first_name (VARCHAR(50) NOT NULL), last_name (VARCHAR(50) NOT NULL), date_of_birth (DATE NOT NULL), gender (VARCHAR(50) NOT NULL), country_of_citizenship (VARCHAR(50) NOT NULL), country_of_residence (VARCHAR(50) NOT NULL), passport_number (VARCHAR(20) NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), and updated_at (TIMESTAMP NOT NULL DEFAULT now()).
- security_check:** Contains columns for security_check_id (SERIAL PRIMARY KEY), check_result (VARCHAR(20) NOT NULL), created_at (TIMESTAMP NOT NULL DEFAULT now()), updated_at (TIMESTAMP NOT NULL DEFAULT now()), and passenger_id (INT NOT NULL).

Logs:

- [2025-09-23 16:25:38] Connected to lab2_d
- [2025-09-23 16:25:38] completed in 2 s 70
- [2025-09-23 16:25:38] Connected to lab2_d
- [2025-09-23 16:25:38] completed in 2 s 70
- [2025-09-23 16:25:38] Connected to lab2_d
- [2025-09-23 16:25:38] completed in 2 s 70

alter table

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

- Top Bar:** Includes icons for file operations (New, Open, Save, etc.), a search bar, and tabs for "test_DB" and "Version control".
- Database Explorer:** Shows connections to "postgres@localhost" (2 databases: "lab2_db" and "postgres") and "Server Objects".
- Script Editor:** A tab titled "t.sql" containing the following SQL code:

```
1 ✓ ALTER TABLE airline_info RENAME TO airline;
2
3 ✓ ALTER TABLE booking RENAME COLUMN price TO ticket_price;
4
5 ✓ ALTER TABLE flights
6     ALTER COLUMN departing_gate TYPE TEXT;
7
8 ✓ ALTER TABLE airline
9     DROP COLUMN info;
```
- Services:** A terminal window showing the execution history of the commands in "t.sql":

```
[2025-09-23 18:05:31] lab2_db> ALTER TABLE airline_info RENAME TO airline
[2025-09-23 18:05:31] completed in 13 ms
[2025-09-23 18:05:31] lab2_db> ALTER TABLE booking RENAME COLUMN price TO ticket_pr
[2025-09-23 18:05:31] completed in 4 ms
[2025-09-23 18:05:31] lab2_db> ALTER TABLE flights
[2025-09-23 18:05:31]     ALTER COLUMN departing_gate TYPE TEXT
[2025-09-23 18:05:31] completed in 20 ms
[2025-09-23 18:05:31] lab2_db> ALTER TABLE airline
[2025-09-23 18:05:31]     DROP COLUMN info
[2025-09-23 18:05:31] completed in 11 ms
```
- Bottom Status:** Shows the current connection is "test_DB > t.sql" and the file is saved with "10:1 CRLF".

foreign key

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

- Database Explorer:** Shows the schema structure of the `lab2_db` database, which contains the `public` schema and various tables like `airline`, `airport`, `baggage`, etc.
- Current Connection:** `postgres@localhost` (1 of 3)
- Current File:** `t.sql`
- Code Content:** The SQL script defines multiple foreign key constraints and their corresponding indexes across several tables. The constraints are named `fk_booking_flight_booking`, `fk_booking_flight_flight`, `fk_flights_departing_airport`, `fk_flights_arriving_airport`, `fk_bookings_airline`, and `fk_booking_pass`. The indexes are named `idx_booking_flight_booking`, `idx_booking_flight_flight`, `idx_flights_departing_airport`, `idx_flights_arriving_airport`, `idx_flights_airline`, and `idx_boarding_pass_booking`.

```
1 ALTER TABLE security_check
2   ADD CONSTRAINT fk_security_check_passenger
3     FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id) ON DELETE CASCADE;
4 CREATE INDEX idx_security_check_passenger ON security_check(passenger_id);
5
6 ALTER TABLE booking
7   ADD CONSTRAINT fk_booking_passenger
8     FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id) ON DELETE CASCADE;
9 CREATE INDEX idx_booking_passenger ON booking(passenger_id);
10
11 ALTER TABLE baggage_check
12   ADD CONSTRAINT fk_baggage_check_passenger
13     FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id) ON DELETE CASCADE;
14 CREATE INDEX idx_baggage_check_passenger ON baggage_check(passenger_id);
15
16 ALTER TABLE baggage_check
17   ADD CONSTRAINT fk_baggage_check_booking
18     FOREIGN KEY (booking_id) REFERENCES booking(booking_id) ON DELETE CASCADE;
19 CREATE INDEX idx_baggage_check_booking ON baggage_check(booking_id);
20
21 ALTER TABLE baggage
22   ADD CONSTRAINT fk_baggage_booking
23     FOREIGN KEY (booking_id) REFERENCES booking(booking_id) ON DELETE CASCADE;
24 CREATE INDEX idx_baggage_booking ON baggage(booking_id);
25
26 ALTER TABLE boarding_pass
27   ADD CONSTRAINT fk_boarding_pass_booking
28     FOREIGN KEY (booking_id) REFERENCES booking(booking_id) ON DELETE CASCADE;
29 CREATE INDEX idx_boarding_pass_booking ON boarding_pass(booking_id);
30
31 ALTER TABLE booking_flight
32   ADD CONSTRAINT fk_booking_flight_booking
33     FOREIGN KEY (booking_id) REFERENCES booking(booking_id) ON DELETE CASCADE;
34 CREATE INDEX idx_booking_flight_booking ON booking_flight(booking_id);
35
36 ALTER TABLE booking_flight
37   ADD CONSTRAINT fk_booking_flight_flight
38     FOREIGN KEY (flight_id) REFERENCES flights(flight_id) ON DELETE CASCADE;
39 CREATE INDEX idx_booking_flight_flight ON booking_flight(flight_id);
40
41 ALTER TABLE flights
42   ADD CONSTRAINT fk_flights_departing_airport
43     FOREIGN KEY (departing_airport_id) REFERENCES airport(airport_id) ON DELETE RESTRICT;
44 CREATE INDEX idx_flights_departing_airport ON flights(departing_airport_id);
45
46 ALTER TABLE flights
47   ADD CONSTRAINT fk_flights_arriving_airport
48     FOREIGN KEY (arriving_airport_id) REFERENCES airport(airport_id) ON DELETE RESTRICT;
49 CREATE INDEX idx_flights_arriving_airport ON flights(arriving_airport_id);
50
51 ALTER TABLE flights
52   ADD CONSTRAINT fk_flights_airline
53     FOREIGN KEY (airline_id) REFERENCES airline(airline_id) ON DELETE RESTRICT;
54 CREATE INDEX idx_flights_airline ON flights(airline_id);
55
56 ALTER TABLE booking
57   ADD CONSTRAINT fk_booking_flight
58     FOREIGN KEY (flight_id) REFERENCES flights(flight_id) ON DELETE RESTRICT;
59 CREATE INDEX idx_booking_flight ON booking(flight_id);
60
61 |
```

This image shows a screenshot of a multi-terminal code editor interface, likely from a tool like Redgate SQL Source Control or a similar database management environment. The interface is divided into several panes, each displaying a different part of a large SQL script.

The main script (in.sql) contains numerous `INSERT INTO` statements for various tables, including:

- airline**: Adds records for Aeroméxico, American Airlines, South African Airways, Ethiopian Airlines, and others.
- airport**: Adds records for Kobe Airport, Xuzhou Guanyin Airport, Shahri Ashrafi Esfahani Airport, Svamsudin Noor Airport, Enontekiö Airport, Contadora Airport, Fria Airport, West Angeles Airport, Balikesir Merkez Airport, Yotvata Airfield, and Salah Airport.
- passengers**: Adds records for passengers with names like Sile, Aaron, Denna, Wackett, Dagmar, Eunice, Hollian, Dido, Oren, Roalfe, Kora, Shimony, Tildie, Clea, Delilah, Ginnie, Jacklin, Wendie, Cordi, Arlyene, Teressa, Brose, Linus, Vanesch, and Oralynne.
- flights**: Adds records for flights between various airports, including Schiphol, Amsterdam, and other international destinations.
- boarding_pass**: Adds records for boarding passes with seat numbers like SVE-V\$, S\$U\$-ORS, SKE-708\$, S\$CN-353\$, S\$US-RTS\$, S\$BP-KAD\$, S\$CH-L55\$, S\$US-WY\$, S\$KI-KSS\$, S\$IN-MHS\$, and S\$HR-13\$\$. These entries include complex dollar sign (\$) placeholder strings for seat numbers.
- booking**: Adds records for bookings with flight IDs, passenger IDs, and various status codes and ticket prices.
- baggage**: Adds records for luggage items with weight_in_kg values ranging from 3 to 50+.

The code editor features syntax highlighting, code completion, and navigation tools typical of modern IDEs. The bottom of the screen shows the current file path, encoding (CRLF), character set (UTF-8), and other standard file metadata.

Database Explorer

t.sql × in.sql

postgres@localhost 2

lab2_db 1 of 3

postgres 1 of 3

Server Objects

1 ✓ INSERT INTO airline (airline_code, airline_name, airline_country, created_at, updated_at)
2 VALUES (airline_code 'KZ', airline_name 'KazAir', airline_country 'Kazakhstan', created_at now(), updated_at now());

Services

Tx + ⏪ ⏴ ⏵ ⏳ ×

Database

postgres@localhost

t.sql 322 ms

in.sql

[2025-09-24 07:51:30] Connected to lab2_db
[2025-09-24 07:51:30] lab2_db> INSERT INTO airline (airline_code, airline_name, airline_country, created_at, updated_at)
VALUES ('KZ', 'KazAir', 'Kazakhstan', now(), now())
[2025-09-24 07:51:30] 1 row affected in 19 ms

test_DB > t.sql

2:53 CRLF UTF-8 4 spaces ⏴ Trial

TD test_DB Version control

Database Explorer t.sql x in.sql

postgres@localhost 2
lab2_db 1 of 3
postgres 1 of 3
Server Objects

Playground Tx: Auto

```
1 ✓ UPDATE airline
2 SET airline_country = 'Turkey',
3     updated_at = now()
4 WHERE airline_name = 'KazAir';
5
```

Services

Tx + ⊖ ⊕ | ⌂ ×

Database postgres@localhost
t.sql 105 ms
in.sql

Output lab2_db.public.airline

```
[2025-09-24 07:52:49] 1 row affected in 15 ms
[2025-09-24 07:52:54] lab2_db.public> UPDATE airline
          SET airline_country = 'Turkey',
              updated_at = now()
        WHERE airline_name = 'KazAir'
[2025-09-24 07:52:54] 1 row affected in 17 ms
```

test_DB > t.sql

5:1 CRLF UTF-8 4 spaces Trial

14°C Cloudy 07:53 ENG 24.09.2025

This screenshot shows a PostgreSQL database management interface. The top section is the Database Explorer, displaying a connection to 'postgres@localhost' with two databases selected: 'lab2_db' (1 of 3) and 'postgres' (1 of 3). A query editor window titled 't.sql' is open, containing the following SQL code:

```
1 ✓ UPDATE airline
2 SET airline_country = 'Turkey',
3     updated_at = now()
4 WHERE airline_name = 'KazAir';
5
```

The bottom section is the Services panel, specifically the Transaction (Tx) tab. It shows a transaction history with the following entries:

- [2025-09-24 07:52:49] 1 row affected in 15 ms
- [2025-09-24 07:52:54] lab2_db.public> UPDATE airline
SET airline_country = 'Turkey',
 updated_at = now()
WHERE airline_name = 'KazAir'
- [2025-09-24 07:52:54] 1 row affected in 17 ms

The transaction for 't.sql' is highlighted with a blue selection bar.

The screenshot shows a PostgreSQL database management interface within a code editor. The top bar includes tabs for 'test_DB' and 'Version control', and icons for file operations like save, open, and copy. The main area is titled 'Database Explorer' and contains two tabs: 't.sql' (selected) and 'in.sql'. The 't.sql' tab displays the following SQL code:

```
1 ✓ INSERT INTO airline (airline_code, airline_name, airline_country, created_at, updated_at)
VALUES
  (airline_code 'AE', airline_name 'AirEasy', airline_country 'France', created_at now(), updated_at now()),
  (airline_code 'FH', airline_name 'FlyHigh', airline_country 'Brazil', created_at now(), updated_at now()),
  (airline_code 'FF', airline_name 'FlyFly', airline_country 'Poland', created_at now(), updated_at now());
```

The 'Services' panel at the bottom shows a transaction history with 't.sql' highlighted. The 'Output' panel shows the execution log:

```
[2025-09-24 07:54:28] lab2_db.public> INSERT INTO airline (airline_code, airline_name, airline_country, created_at, updated_at)
VALUES
  ('AE', 'AirEasy', 'France', now(), now()),
  ('FH', 'FlyHigh', 'Brazil', now(), now()),
  ('FF', 'FlyFly', 'Poland', now(), now())
[2025-09-24 07:54:28] 3 rows affected in 12 ms
```

The bottom status bar shows the file path 'test_DB > t.sql', encoding '6:1 CRLF', character set 'UTF-8', and other system information.

Services

The screenshot shows a PostgreSQL client interface. At the top, there are icons for creating a new transaction (Tx), adding a new connection (plus sign), viewing connections (eye), and closing the window (X). Below this is a toolbar with a square icon and a dropdown menu labeled "Database". Under the "Database" menu, "postgres@localhost" is selected, indicated by a blue elephant icon. Two queries are currently running:

- A green query icon followed by "t.sql" and "87 ms".
- A purple query icon followed by "in.sql".

At the bottom, the current connection is shown as "test_DB > t.sql".

TD test_DB Version control

Database Explorer

t.sql x in.sql

postgres@localhost 2

lab2_db 1 of 3

postgres 1 of 3

Server Objects

SELECT COUNT(*) AS cheap_bookings
FROM booking
WHERE ticket_price < 10000;

DELETE FROM booking
WHERE ticket_price < 10000;

lab2_db.public postgres@localhost

```
1 ✓ SELECT COUNT(*) AS cheap_bookings
2   FROM booking
3   WHERE ticket_price < 10000;
4
5 ✓ DELETE FROM booking
6   WHERE ticket_price < 10000;
```

Services

Tx + ⚡ ⚡ | ⚡ ⚡

Database

postgres@localhost

t.sql 128 ms

in.sql

Output cheap_bookings:bigint

cheap_bookings

1	29

1 row

test_DB > t.sql

6:27 CRLF UTF-8 4 spaces Trial

14°C Cloudy

08:03

ENG

24.09.2025

1	29