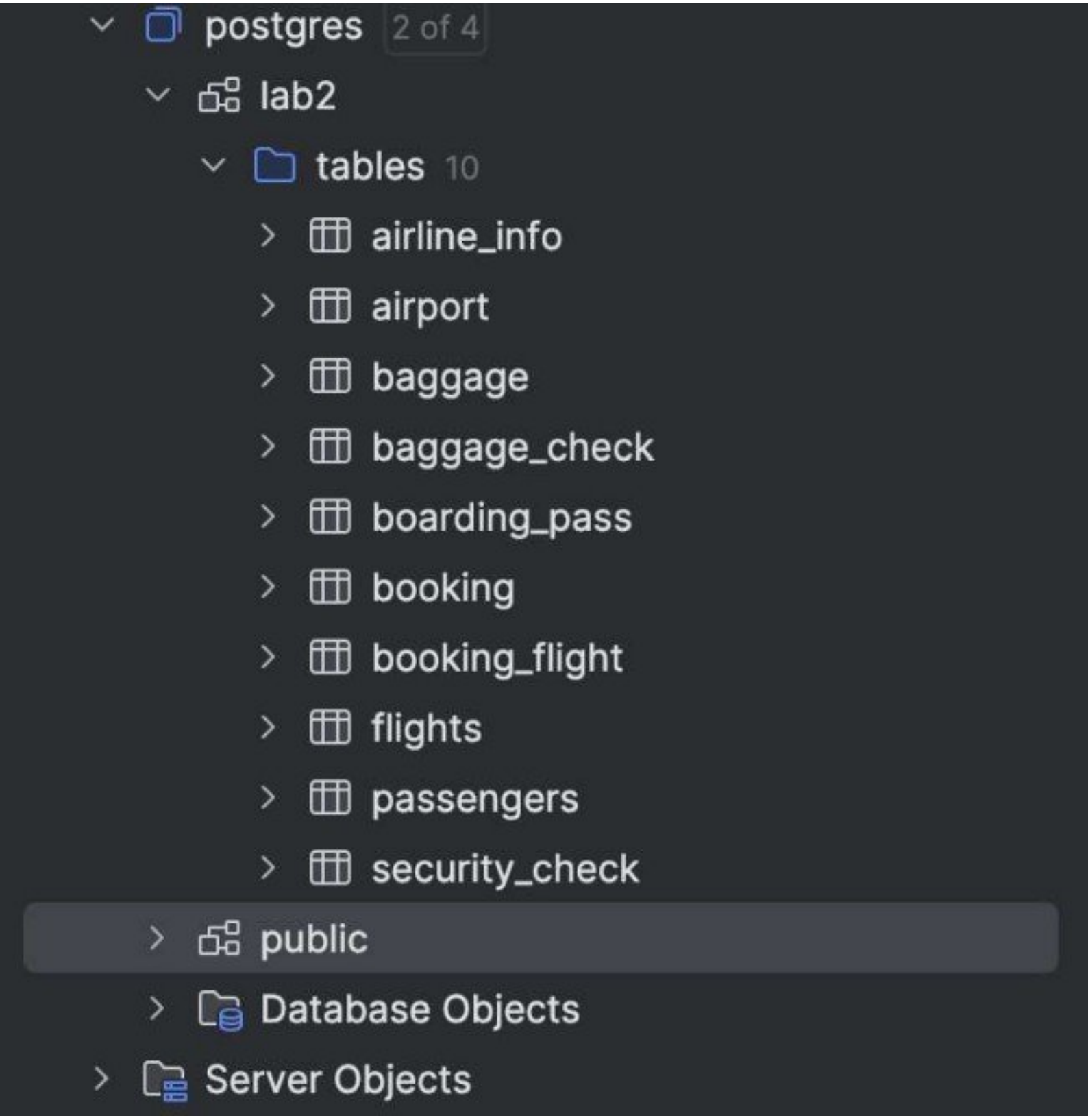

postgres 2 of 4
lab2
tables 10
airline_info
airport
baggage
baggage_check
boarding_pass
booking
booking_flight
flights
passengers
security_check
public
Database Objects
Server Objects

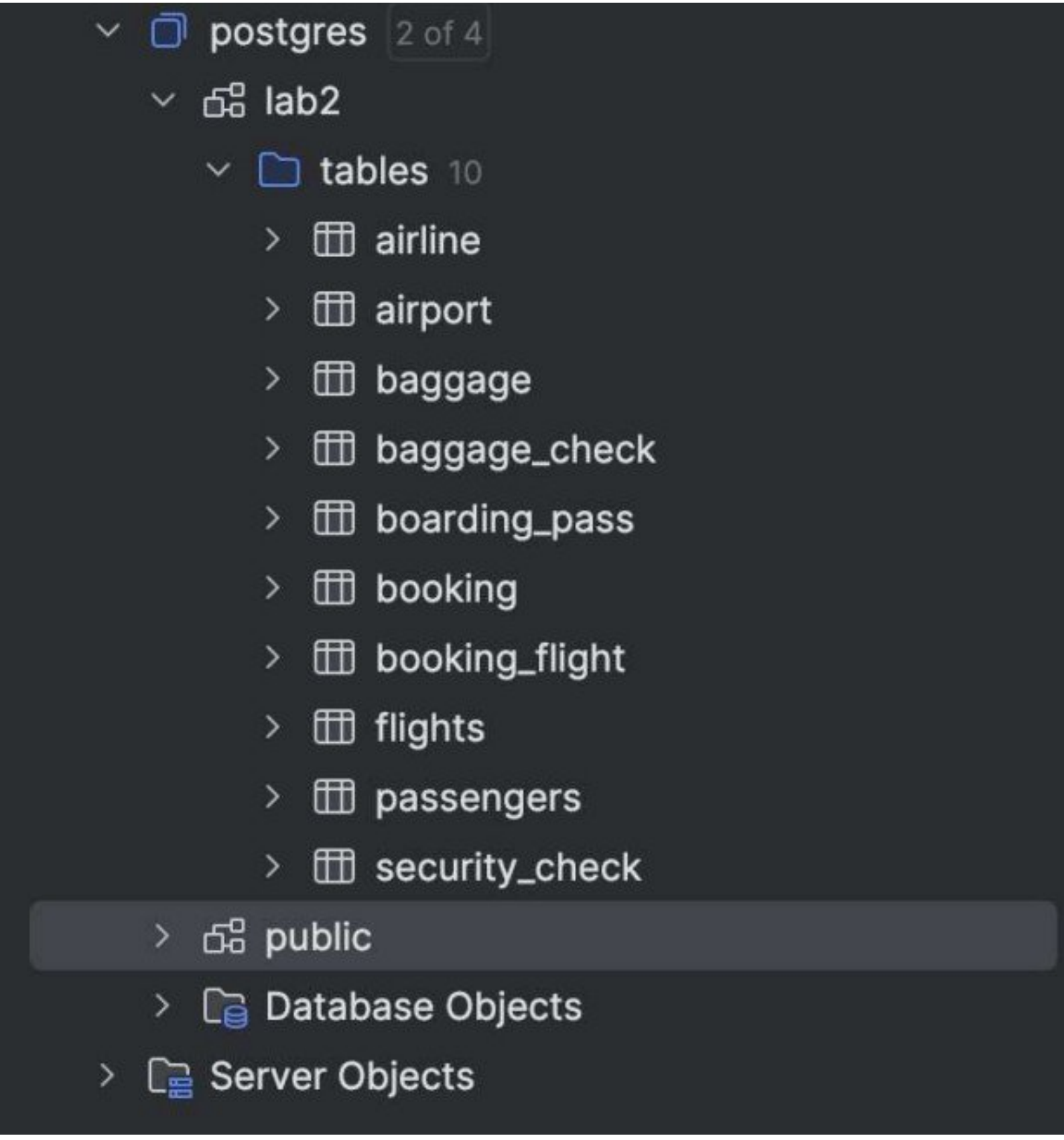
```sql
CREATE TABLE baggage_check (
  baggage_check_id INT PRIMARY KEY,
  check_result     VARCHAR(50) NOT NULL,
  created_at       TIMESTAMP   NOT NULL,
  updated_at       TIMESTAMP   NOT NULL,
  booking_id       INT         NOT NULL,
  passenger_id     INT         NOT NULL
);


CREATE TABLE boarding_pass (
  boarding_pass_id INT PRIMARY KEY,
  booking_id       INT         NOT NULL,
  seat             VARCHAR(50) NOT NULL,
  boarding_time    TIMESTAMP   NOT NULL,
  created_at       TIMESTAMP   NOT NULL,
  updated_at       TIMESTAMP   NOT NULL
);

CREATE TABLE security_check (
  security_check_id INT PRIMARY KEY,
  check_result      VARCHAR(20) NOT NULL,
  created_at        TIMESTAMP   NOT NULL,
  updated_at        TIMESTAMP   NOT NULL,
  passenger_id      INT         NOT NULL
);
```

```sql
CREATE TABLE booking (
  booking_id       INT PRIMARY KEY,
  flight_id        INT          NOT NULL,
  passenger_id     INT          NOT NULL,
  booking_platform VARCHAR(50) NOT NULL,
  created_at       TIMESTAMP   NOT NULL,
  updated_at       TIMESTAMP   NOT NULL,
  status           VARCHAR(50) NOT NULL,
  price            DECIMAL(7,2) NOT NULL
);


CREATE TABLE booking_flight (
  booking_flight_id INT PRIMARY KEY,
  booking_id        INT NOT NULL,
  flight_id         INT NOT NULL,
  created_at        TIMESTAMP NOT NULL,
  updated_at        TIMESTAMP NOT NULL
);


CREATE TABLE baggage (
  baggage_id   INT PRIMARY KEY,
  weight_in_kg DECIMAL(4,2) NOT NULL,
  created_at   TIMESTAMP    NOT NULL,
  updated_at   TIMESTAMP    NOT NULL,
  booking_id   INT          NOT NULL
);
```

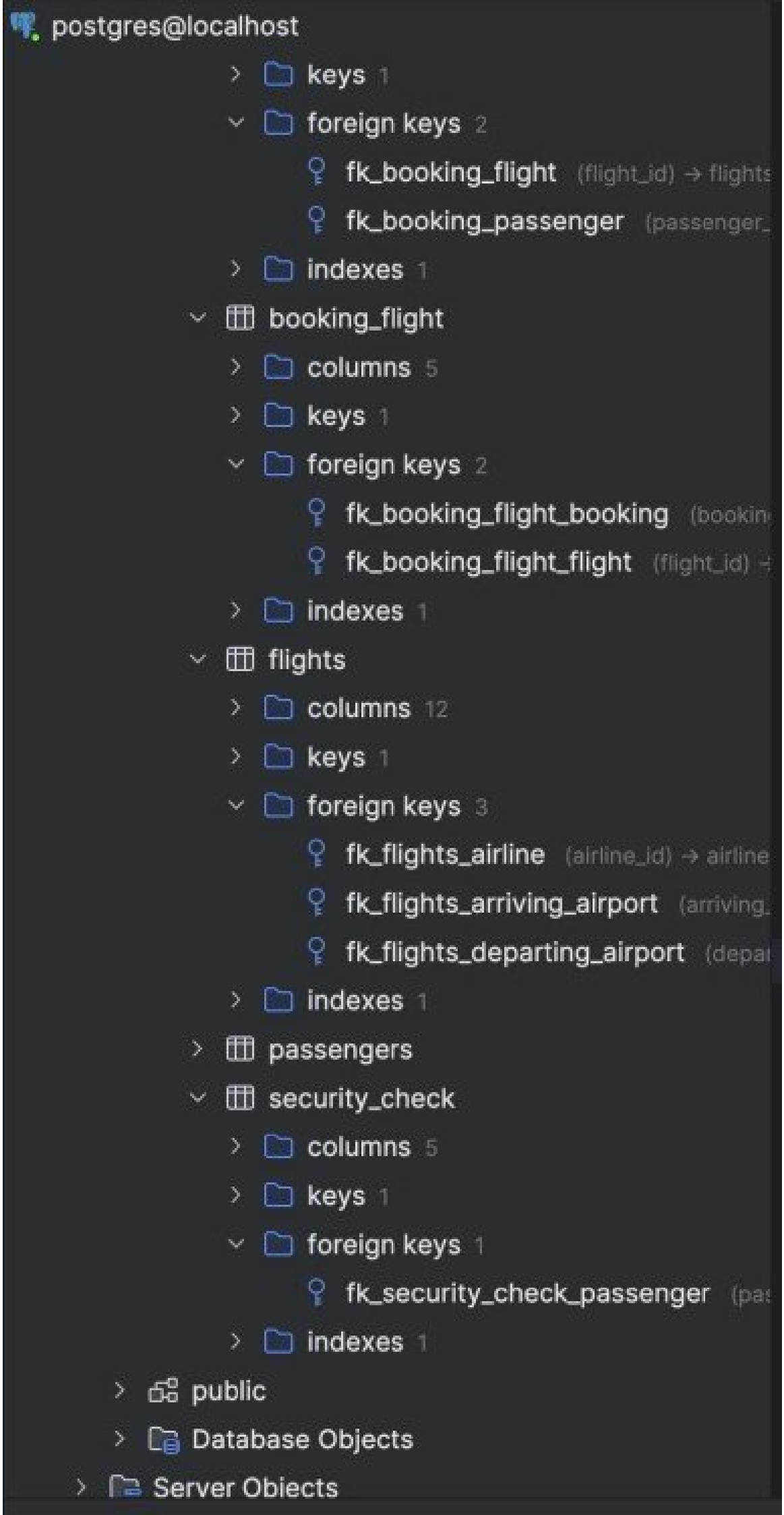
```sql
CREATE TABLE passengers (
  passenger_id              INT PRIMARY KEY,
  first_name                VARCHAR(50) NOT NULL,
  last_name                 VARCHAR(50) NOT NULL,
  date_of_birth             DATE        NOT NULL,
  gender                    VARCHAR(50) NOT NULL,
  country_of_citizenship    VARCHAR(50) NOT NULL,
  country_of_residence      VARCHAR(50) NOT NULL,
  passport_number           VARCHAR(20) NOT NULL,
  created_at                TIMESTAMP   NOT NULL,
  updated_at                TIMESTAMP   NOT NULL
);
```

```sql
CREATE TABLE flights (
  flight_id              INT 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,
  created_at             TIMESTAMP   NOT NULL,
  updated_at             TIMESTAMP   NOT NULL
);
```

```sql
Create TABLE airline_info (
  airline_id INT     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,
  updated_at TIMESTAMP    NOT NULL,
  info VARCHAR(50) NOT    NULL
);

CREATE TABLE airport (
  airport_id   INT 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,
  updated_at   TIMESTAMP   NOT NULL
);
```


postgres 2 of 4
lab2
tables 10
airline
airport
baggage
baggage_check
boarding_pass
booking
booking_flight
flights
passengers
security_check
public
Database Objects
Server Objects

```sql
ALTER TABLE airline_info RENAME TO airline;

ALTER TABLE booking RENAME COLUMN price TO ticket_price;

ALTER TABLE flights ALTER COLUMN departing_gate TYPE TEXT;

ALTER TABLE airline DROP COLUMN info;

```


postgres@localhost
keys 1
foreign keys 2
fk_booking_flight (flight_id) → flights
fk_booking_passenger (passenger_
indexes 1
booking_flight
columns 5
keys 1
foreign keys 2
fk_booking_flight_booking (bookin
fk_booking_flight_flight (flight_id) →
indexes 1
flights
columns 12
keys 1
foreign keys 3
fk_flights_airline (airline_id) → airline
fk_flights_arriving_airport (arriving_
fk_flights_departing_airport (depa
indexes 1
passengers
security_check
columns 5
keys 1
foreign keys 1
fk_security_check_passenger (pa
indexes 1
public
Database Objects
Server Objects

```sql
ALTER TABLE booking
  ADD CONSTRAINT fk_booking_passenger
  FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id);


ALTER TABLE security_check
  ADD CONSTRAINT fk_security_passenger
  FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id);


ALTER TABLE baggage_check
  ADD CONSTRAINT fk_baggage_check_passenger
  FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id);

ALTER TABLE flights
  ADD CONSTRAINT fk_flights_airline
  FOREIGN KEY (airline_id) REFERENCES airline(airline_id);


ALTER TABLE flights
  ADD CONSTRAINT fk_flights_departing_airport
  FOREIGN KEY (departing_airport_id) REFERENCES airport(airport_id);

ALTER TABLE flights
  ADD CONSTRAINT fk_flights_arriving_airport
  FOREIGN KEY (arriving_airport_id) REFERENCES airport(airport_id);


```

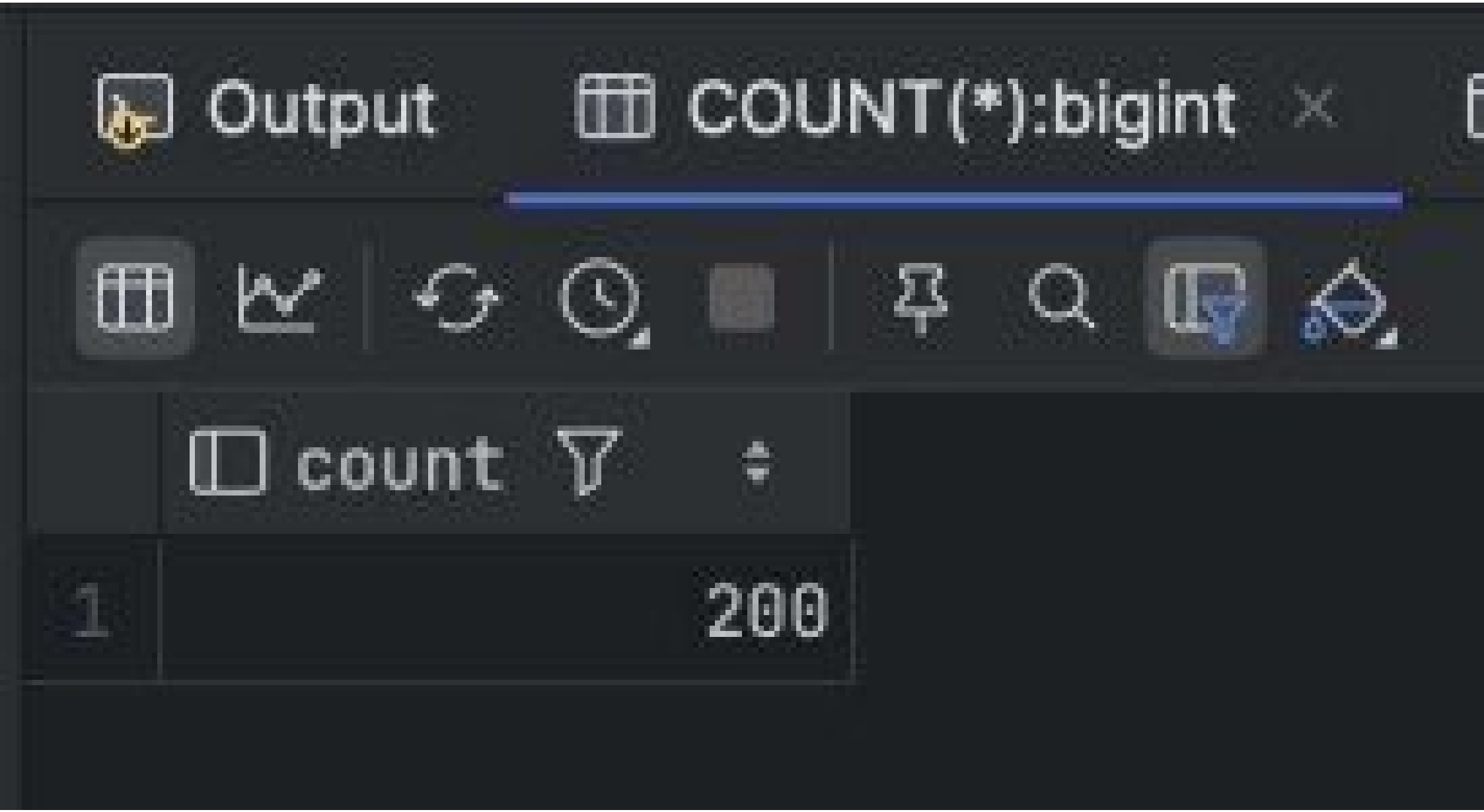
```sql

ALTER TABLE baggage_check
  ADD CONSTRAINT fk_baggage_check_booking
  FOREIGN KEY (booking_id) REFERENCES booking(booking_id);

ALTER TABLE baggage
  ADD CONSTRAINT fk_baggage_booking
  FOREIGN KEY (booking_id) REFERENCES booking(booking_id);

ALTER TABLE boarding_pass
  ADD CONSTRAINT fk_boarding_pass_booking
  FOREIGN KEY (booking_id) REFERENCES booking(booking_id);

ALTER TABLE booking_flight
  ADD CONSTRAINT fk_booking_flight_booking
  FOREIGN KEY (booking_id) REFERENCES booking(booking_id);


ALTER TABLE booking_flight
  ADD CONSTRAINT fk_booking_flight_flight
  FOREIGN KEY (flight_id) REFERENCES flights(flight_id);
```

```sql
-- 1) airline
INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at, updated_at)
SELECT gs,
       'AL' || gs::text,
       'Airline_' || gs::text,
       'Country_' || ((gs % 10) + 1)::text,
       now(), now()
FROM generate_series(1,200) AS gs
ON CONFLICT (airline_id) DO NOTHING;

-- 2) airport
INSERT INTO airport (airport_id, airport_name, country, state, city, created_at, updated_at)
SELECT gs,
       'Airport_' || gs::text,
       'Country_' || ((gs % 10) + 1)::text,
       'State_'   || ((gs % 5)  + 1)::text,
       'City_'    || ((gs % 20) + 1)::text,
       now(), now()
FROM generate_series(1,200) AS gs
ON CONFLICT (airport_id) DO NOTHING;

-- 3) passengers
INSERT INTO passengers (passenger_id, first_name, last_name, date_of_birth, gender,
                        country_of_citizenship, country_of_residence, passport_number, created_at, updated_at)
SELECT gs,
       'Name_' || gs::text,
       'Surname_' || gs::text,
       date '1980-01-01' + (gs::text || ' days')::interval,
       CASE WHEN gs % 2 = 0 THEN 'male' ELSE 'female' END,
       'Country_' || ((gs % 10) + 1)::text,
       'Country_' || ((gs % 10) + 1)::text,
       'P' || gs::text,
       now(), now()
FROM generate_series(1,200) AS gs
ON CONFLICT (passenger_id) DO NOTHING;
```

```sql
-- 4) flights
INSERT INTO flights (flight_id, sch_departure_time, sch_arrival_time,
                     departing_airport_id, arriving_airport_id,
                     departing_gate, arriving_gate, airline_id,
                     act_departure_time, act_arrival_time, created_at, updated_at)
SELECT flight_id gs,
       sch_departure_time now() + (gs::text || ' hours')::interval,
       sch_arrival_time now() + (gs::text || ' hours')::interval + interval '2 hours',
       ((gs % 200) + 1),
       ((gs % 200) + 1),
       departing_gate 'DG_' || gs::text,
       arriving_gate 'AG_' || gs::text,
       ((gs % 200) + 1),
       act_departure_time now() + (gs::text || ' hours')::interval,
       act_arrival_time now() + (gs::text || ' hours')::interval + interval '2 hours',
       created_at now(), updated_at now()
FROM generate_series(1,200) AS gs
ON CONFLICT (flight_id) DO NOTHING;

-- 5) booking  (поле уже переименовано на ticket_price)
INSERT INTO booking (booking_id, flight_id, passenger_id, booking_platform,
                     created_at, updated_at, status, ticket_price)
SELECT booking_id gs,
       ((gs % 200) + 1),
       ((gs % 200) + 1),
       booking_platform 'platform_' || ((gs % 3) + 1)::text,
       created_at now(), updated_at now(),
       status 'status_' || ((gs % 3) + 1)::text,
       (10000 + gs)::numeric(7,2)
FROM generate_series(1,200) AS gs
ON CONFLICT (booking_id) DO NOTHING;
```

```sql
-- 6) booking_flight
INSERT INTO booking_flight (booking_flight_id, booking_id, flight_id, created_at, updated_at)
SELECT booking_flight_id gs, booking_id gs, ((gs % 200) + 1), created_at now(), updated_at now()
FROM generate_series(1,200) AS gs
ON CONFLICT (booking_flight_id) DO NOTHING;

-- 7) baggage
INSERT INTO baggage (baggage_id, weight_in_kg, created_at, updated_at, booking_id)
SELECT baggage_id gs,
       (10 + (gs % 20))::numeric(4,2),
       created_at now(), updated_at now(),
       booking_id gs
FROM generate_series(1,200) AS gs
ON CONFLICT (baggage_id) DO NOTHING;

-- 8) baggage_check
INSERT INTO baggage_check (baggage_check_id, check_result, created_at, updated_at, booking_id, passenger_id)
SELECT baggage_check_id gs, check_result 'OK', created_at now(), updated_at now(), booking_id gs, passenger_id gs
FROM generate_series(1,200) AS gs
ON CONFLICT (baggage_check_id) DO NOTHING;

-- 9) boarding_pass
INSERT INTO boarding_pass (boarding_pass_id, booking_id, seat, boarding_time, created_at, updated_at)
SELECT boarding_pass_id gs, booking_id gs, seat 'Seat_' || gs::text, boarding_time now(), created_at now(), updated_at now()
FROM generate_series(1,200) AS gs
ON CONFLICT (boarding_pass_id) DO NOTHING;

-- 10) security_check
INSERT INTO security_check (security_check_id, check_result, created_at, updated_at, passenger_id)
SELECT security_check_id gs, check_result 'OK', created_at now(), updated_at now(), passenger_id gs
FROM generate_series(1,200) AS gs
ON CONFLICT (security_check_id) DO NOTHING;
```

```sql
SELECT COUNT(*) FROM airport;
SELECT COUNT(*) FROM airline;
SELECT COUNT(*) FROM passengers;
SELECT COUNT(*) FROM flights;
SELECT COUNT(*) FROM booking;
```


Output
COUNT(*):bigint
count
1
200

```sql
INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at, updated_at)
VALUES ( airline_id 201, airline_code 'KZ', airline_name 'KazAir', airline_country 'Kazakhstan', created_at now(), updated_at now());

-- Проверка
SELECT * FROM airline WHERE airline_name='KazAir';
```

	airline_id	airline_code	airline_name	airline_country	created_at	upd
1	201	KZ	KazAir	Kazakhstan	2025-09-23 20:54:26.378997	2025-0

```sql
UPDATE airline
SET airline_country='Turkey', updated_at=now()
WHERE airline_name='KazAir';

-- Проверка
SELECT * FROM airline WHERE airline_name='KazAir';
```

	airline_id	airline_code	airline_name	airline_country	created_at	upd
1	201	KZ	KazAir	Turkey	2025-09-23 20:54:26.378997	2025-

```sql
INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at, updated_at)
VALUES
( airline_id 202, airline_code 'AE', airline_name 'AirEasy', airline_country 'France', created_at now(), updated_at now()),
( airline_id 203, airline_code 'FH', airline_name 'FlyHigh', airline_country 'Brazil', created_at now(), updated_at now()),
( airline_id 204, airline_code 'FF', airline_name 'FlyFly', airline_country 'Poland', created_at now(), updated_at now());

-- Проверка
SELECT airline_id, airline_code, airline_name, airline_country
FROM airline
WHERE airline_id BETWEEN 202 AND 204;
```

	airline_id	airline_code	airline_name	airline_country
1	202	AE	AirEasy	France
2	203	FH	FlyHigh	Brazil
3	204	FF	FlyFly	Poland

```sql
-- До удаления
SELECT flight_id, sch_arrival_time, act_arrival_time
FROM flights
WHERE (sch_arrival_time BETWEEN '2024-01-01' AND '2024-12-31')
   OR (act_arrival_time BETWEEN '2024-01-01' AND '2024-12-31');

-- Удаление
DELETE FROM flights
WHERE (sch_arrival_time BETWEEN '2024-01-01' AND '2024-12-31')
   OR (act_arrival_time BETWEEN '2024-01-01' AND '2024-12-31');

-- После удаления
SELECT flight_id, sch_arrival_time, act_arrival_time FROM flights LIMIT 10;
```

	flight_id	sch_arrival_time	act_arrival_time
1	1	2025-09-23 23:46:32.067402	2025-09-23 23:46:32.067402
2	2	2025-09-24 00:46:32.067402	2025-09-24 00:46:32.067402
3	3	2025-09-24 01:46:32.067402	2025-09-24 01:46:32.067402
4	4	2025-09-24 02:46:32.067402	2025-09-24 02:46:32.067402
5	5	2025-09-24 03:46:32.067402	2025-09-24 03:46:32.067402
6	6	2025-09-24 04:46:32.067402	2025-09-24 04:46:32.067402
7	7	2025-09-24 05:46:32.067402	2025-09-24 05:46:32.067402
8	8	2025-09-24 06:46:32.067402	2025-09-24 06:46:32.067402
9	9	2025-09-24 07:46:32.067402	2025-09-24 07:46:32.067402
10	10	2025-09-24 08:46:32.067402	2025-09-24 08:46:32.067402

10 rows

```sql
-- До
SELECT booking_id, ticket_price FROM booking ORDER BY booking_id LIMIT 10;

-- Обновление
UPDATE booking
SET ticket_price = ROUND(ticket_price * 1.15, 2),
    updated_at = now();

-- После
SELECT booking_id, ticket_price FROM booking ORDER BY booking_id LIMIT 10;
```

	booking_id	ticket_price
1	1	11501.15
2	2	11502.30
3	3	11503.45
4	4	11504.60
5	5	11505.75
6	6	11506.90
7	7	11508.05
8	8	11509.20
9	9	11510.35
10	10	11511.50

10 rows

```sql
-- До
SELECT booking_id, ticket_price FROM booking WHERE ticket_price < 10000 LIMIT 10;

-- Удаление
DELETE FROM booking WHERE ticket_price < 10000;

-- После
SELECT booking_id, ticket_price FROM booking ORDER BY ticket_price LIMIT 10;
```

	booking_id	ticket_price
1	1	11501.15
2	2	11502.30
3	3	11503.45
4	4	11504.60
5	5	11505.75
6	6	11506.90
7	7	11508.05
8	8	11509.20
9	9	11510.35
10	10	11511.50

10 rows