

Object Explorer

Servers (1)

PostgreSQL 17

Databases (1)

postgres

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications

Schemas (1)

public

- > Aggregates
- > Collations
- > Domains
- > FTS Configurations
- > FTS Dictionaries
- > FTS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences

Tables (11)

- > airline
- > airport
- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > booking_flight
- > flights
- > passengers

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

Query

```
1 CREATE INDEX actual_departure_index ON flights(actual_departure);
```

Scratch Pad X

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 167 msec.

- Servers (1)
 - PostgreSQL 17
- Databases (1)
 - postgres
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (11)
 - airline
 - airport
 - baggage
 - baggage_check
 - boarding_pass
 - booking
 - booking_flight
 - flights
 - passengers
 - security_check

postgres/postgres@PostgreSQL 17

No limit

Query History

Scratch Pad

CREATE UNIQUE INDEX name ON flights (flight_no, scheduled_departure);

Data Output Messages Notifications

ERROR: создать уникальный индекс "name" не удалось
Ключ (flight_no, scheduled_departure)=(US-KS, 2023-09-04) дублируется.

ОШИБКА: создать уникальный индекс "name" не удалось
SQL state: 23505
Detail: Ключ (flight_no, scheduled_departure)=(US-KS, 2023-09-04) дублируется.

Total rows: 0 Query complete 00:00:00.150

Copy To

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@PostgreSQL 17*

Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (11) Columns (14)

CREATE INDEX numb ON flights(departure_airport_id, arrival_airport_id);

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 66 msec.

✓ Query returned successfully in 66 msec. ✎

Object Explorer

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X postgres/postgres@PostgreSQL 17*

Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (11)
airline
airport
baggage
baggage_check
boarding_pass
booking
booking_flight
flights
Columns (14)
flight_id
flight_no
scheduled_departure
scheduled_arrival
departure_airport_id
arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at

postres/postgres@PostgreSQL 17

No limit

Query History

EXPLAIN ANALYZE
SELECT * FROM flights WHERE flight_no = '10';

Data Output Messages Notifications

Showing rows: 1 to 5 Page No: 1 of 1

QUERY PLAN
text

- Seq Scan on flights (cost=0.00..25.50 rows=1 width=61) (actual time=0.073..0.073 rows=0 loops=...
Filter: ((flight_no)::text = '10'::text)
Rows Removed by Filter: 1000
Planning Time: 3.129 ms
Execution Time: 0.088 ms

```
1 CREATE INDEX num ON flights (flight_no);
2
3 EXPLAIN ANALYZE
4 SELECT * FROM flights WHERE flight_no = '10';
```

Data Output Messages Notifications



Showing rows: 1 to 4 Page No: 1 of 1

QUERY PLAN text	
1	Index Scan using num on flights (cost=0.28..8.29 rows=1 width=61) (actual time=0.026..0.026 rows=0 loops=...
2	Index Cond: ((flight_no)::text = '10'::text)
3	Planning Time: 1.069 ms
4	Execution Time: 0.039 ms

File Object Tools Edit View Window Help

Object Explorer

- booking_flight_id
 - booking_id
 - flight_id
 - created_at
 - update_at
- ▶ Constraints
- ▶ Indexes
- ▶ RLS Policies
- ▶ Rules
- ▶ Triggers
- flights
 - Columns (14)
 - flight_id
 - flight_no
 - scheduled_departure
 - scheduled_arrival
 - departure_airport_id
 - arrival_airport_id
 - departing_gate
 - arriving_gate
 - airline_id
 - status
 - actual_departure
 - actual_arrival
 - created_at
 - update_at
 - ▶ Constraints
 - ▶ Indexes
 - ▶ RLS Policies
 - ▶ Rules
 - ▶ Triggers
- passengers
- security_check
- student
- Trigger Functions
- Types

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

```
1 EXPLAIN ANALYZE
2
3 SELECT DISTINCT departure_airport_id, arrival_airport_id FROM flights;
4
```

Data Output Messages Notifications

Showing rows: 1 to 6 Page No: 1 of 1

QUERY PLAN

text

- 1 HashAggregate (cost=28.00..29.00 rows=100 width=8) (actual time=0.393..0.417 rows=375 loops=1)
 - 2 Group Key: departure_airport_id, arrival_airport_id
 - 3 Batches: 1 Memory Usage: 61kB
 - 4 -> Seq Scan on flights (cost=0.00..23.00 rows=1000 width=8) (actual time=0.030..0.109 rows=1000 loops=1)
 - 5 Planning Time: 5.733 ms
 - 6 Execution Time: 0.519 ms

Successfully run. Total query runtime: 53 msec. 6 rows affected.

Object Explorer

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X postgres/postgres@PostgreSQL 17 X

booking_flight_id

- booking_id
- flight_id
- created_at
- update_at

> Constraints

> Indexes

> RLS Policies

> Rules

> Triggers

flights

Columns(14)

- flight_id
- flight_no
- scheduled_departure
- scheduled_arrival
- departure_airport_id
- arrival_airport_id
- departing_gate
- arriving_gate
- airline_id
- status
- actual_departure
- actual_arrival
- created_at
- update_at

> Constraints

> Indexes

> RLS Policies

> Rules

> Triggers

passengers

security_check

student

Trigger Functions

Types

Views

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X postgres/postgres@PostgreSQL 17 X

Query History

Query

CREATE UNIQUE INDEX pass_num_index ON passengers(passport_number);

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 54 msec.

Total Rows: 0 Query Duration: 00:00:00.054 CPU: 0.1% OS: 0.0%

✓ Query

Dashboard Properties SQL Statistics Dependencies Dependencies Procedures PostgreSQL/PostgreSQL@PostgreSQL 17

arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at
▶ Constraints
▶ Indexes
▶ RLS Policies
▶ Rules
▶ Triggers
▼ passengers
 Columns (10)
 passenger_id
 first_name
 last_name
 date_of_birth
 gender
 country_of_citizenship
 country_of_residence
 passport_number
 created_at
 update_at
 ▶ Constraints
 ▶ Indexes
 ▶ RLS Policies
 ▶ Rules
 ▶ Triggers
 security_check
 student
 Trigger Functions
 Types
 Views

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 INSERT INTO passengers (
2     passenger_id,
3     first_name,
4     last_name,
5     date_of_birth,
6     gender,
7     country_of_citizenship,
8     country_of_residence,
9     passport_number,
10    created_at,
11    update_at
12 )
13 VALUES (
14     203,
15     'Mrina',
16     'Gedjj',
17     '2001-07-17',
18     'female',
19     'Kazakhstan',
20     'Kazakhstan',
21     'KZ123477',
22     NOW(),
23     NOW()
24 );
25
26
27
```

Scratch Pad

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 39 msec.

arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at

> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers

passengers

> Columns (10)

- passenger_id
- first_name
- last_name
- date_of_birth
- gender
- country_of_citizenship
- country_of_residence
- passport_number
- created_at
- update_at

> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers

security_check

student

Trigger Functions

Types

Views

postgres/postgres@PostgreSQL 17

No limit

```
1 INSERT INTO passengers (
2     passenger_id,
3     first_name,
4     last_name,
5     date_of_birth,
6     gender,
7     country_of_citizenship,
8     country_of_residence,
9     passport_number,
10    created_at,
11    update_at
12 )
13 VALUES (
14     204,
15     'Mencwka',
16     'Pkcnbj',
17     '2000-07-17',
18     'female',
19     'Kazakhstan',
20     'Kazakhstan',
21     'KZ123477',
22     NOW(),
23     NOW()
24 );
25
26
27
```

Data Output Messages Notifications

ERROR: повторяющееся значение ключа нарушает ограничение уникальности "pass_num_index"
Ключ "(passport_number)=(KZ123477)" уже существует.

ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "pass_num_index"
SQL state: 23505
Detail: Ключ "(passport_number)=(KZ123477)" уже существует.

6 When you create a unique index, PostgreSQL automatically ensures that the values in that column are all unique.
If you try to insert a row with a passport number that already exists, the database rejects it to keep data integrity.

arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at

> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers

passengers
 Columns(10)
 passenger_id
 first_name
 last_name
 date_of_birth
 gender
 country_of_citizenship
 country_of_residence
 passport_number
 created_at
 update_at

> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers

> security_check
> student

> Trigger Functions
> Types

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 CREATE INDEX pass_ind ON passengers(first_name, last_name, date_of_birth, country_of_citizenship);
```

Scratch Pad X

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 39 msec.

Object Explorer

- departing_gate
- arriving_gate
- airline_id
- status
- actual_departure
- actual_arrival
- created_at
- update_at
- > Constraints
- > Indexes
- > RLS Policies
- > Rules
- > Triggers
- passengers
 - Columns (10)
 - passenger_id
 - first_name
 - last_name
 - date_of_birth
 - gender
 - country_of_citizenship
 - country_of_residence
 - passport_number
 - created_at
 - update_at
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - > Triggers
- security_check
- student
- Trigger Functions
- Types
- Views
- Subscriptions
- Login/Group Roles

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

```
1 country_of_citizenship='Philippines' AND date_of_birth BETWEEN '1984-01-01' AND '1984-12-31';
2
```

Scratch Pad X

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship	country_of_residence	passport_number
1	8	Howard	Igo	1984-03-29	Male	Philippines	China	408429911-1

7 The query used the index because the conditions matched the indexed columns directly. Using BETWEEN allowed the database to efficiently look up only the rows within the 1984 date range instead of scanning the whole table.

This makes the query faster and shows that the index is working as intended

Departing_date
Arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at

> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers

v passengers

v Columns (10)

- passenger_id
- first_name
- last_name
- date_of_birth
- gender
- country_of_citizenship
- country_of_residence
- passport_number
- created_at
- update_at

> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers

> security_check

> student

> Trigger Functions

> Types

> Views

> Subscriptions

postgres/postgres@PostgreSQL 17

No limit

Query Query History

```
1 SELECT
2   indexname AS index_name,
3   indexdef AS definition
4   FROM
5   pg_indexes
6   WHERE
7   tablename='passengers'
```

Data Output Messages Notifications

Showing rows: 1 to 5

Page No: 1 of 1

SQL

	index_name name	definition text
1	passengers_pkey	CREATE UNIQUE INDEX passengers_pkey ON public.passengers USING btree (passenger_id)
2	unique_id	CREATE UNIQUE INDEX unique_id ON public.passengers USING btree (passenger_id)
3	pass_num_index	CREATE UNIQUE INDEX pass_num_index ON public.passengers USING btree (passport_number)
4	pass_ind	CREATE INDEX pass_ind ON public.passengers USING btree (first_name, last_name, date_of_birth, country_of_citizenship)
5	gender_ind	CREATE INDEX gender_ind ON public.passengers USING btree (gender)

Object Explorer Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X postgres/postgres@PostgreSQL 17* X

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer pane, which lists database objects like `departing_gate`, `arriving_gate`, `airline_id`, `status`, `actual_departure`, `actual_arrival`, `created_at`, `update_at`, and several sections under `passenger` (Constraints, Indexes, RLS Policies, Rules, Triggers). Below these are `Columns (10)` for `passenger_id` (first_name, last_name, date_of_birth, gender, country_of_citizenship, country_of_residence, passport_number, created_at, update_at) and other tables like `security_check` and `student`. At the bottom of the Object Explorer are sections for Trigger Functions, Types, Views, Subscriptions, Login/Group Roles, and Tablespaces.

postgres/postgres@PostgreSQL 17

Query History

```
1 DROP INDEX pass_ind;
2 DROP INDEX gender_ind;
3 DROP INDEX pass_num_index;
```

Data Output Messages Notifications

```
DROP INDEX
```

Query returned successfully in 163 msec.

Total rows: Query complete 00:00:00.163 CRLF Ln 1, Col 1