

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

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

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 80 msec.

Total rows: Query complete 00:00:00.080 ✓ Query returned successfully in 80 msec. CRLF Ln 1, Col 73

The screenshot shows the pgAdmin 4 interface with a single query tab open. The query is:CREATE INDEX idx\_flights\_actual\_departure ON flights (act\_departure\_time);The 'Messages' tab is selected, displaying the message "Query returned successfully in 80 msec." A green success notification bar at the bottom right also says "✓ Query returned successfully in 80 msec." The status bar at the bottom shows "Total rows: Query complete 00:00:00.080" and "CRLF Ln 1, Col 73".

Create an index on the actual\_departure column in the flights table.

The screenshot shows the pgAdmin 4 interface. At the top, there's a menu bar with File, Object, Tools, Edit, View, Window, Help. Below the menu is a toolbar with various icons for database management. The main area has tabs for Welcome, postgres/postgres@PostgreSQL 17\*, and postgres/postgres@PostgreSQL 17\*. The current tab is 'postgres/postgres@PostgreSQL 17\*'. The 'Query' tab is selected, showing the following SQL code:

```
1 CREATE UNIQUE INDEX idx_flights_unique_schedule
2 ON flights (flight_id, sch_departure_time);
```

Below the code, there are tabs for Data Output, Messages, and Notifications. The 'Messages' tab is selected, displaying the output of the query:

CREATE INDEX

Query returned successfully in 115 msec.

In the bottom status bar, it says 'Total rows: 0' and 'Query complete 00:00:00.115'. There's also a green message box in the bottom right corner that says '✓ Query returned successfully in 115 msec. X'.

Create a unique index to ensure flight\_no and scheduled\_departure combinations are unique.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* × postgres/postgres@PostgreSQL 17 No limit Query History Scratch Pad ×

```

1 CREATE INDEX idx_flights_departure_arrival
2 ON flights (departing_airport_id, arriving_airport_id);
3
4 SELECT *
5 FROM flights
6 WHERE departing_airport_id = 228 AND arriving_airport_id = 203;

```

Data Output Messages Notifications

Showing rows: 1 to 200 Page No: 1 of 1

	flight_id [PK] integer	sch_departure_time timestamp without time zone	sch_arrival_time timestamp without time zone	departing_airport_id integer	arriving_airport_id integer	departing_gate text	arriving_gate character varying (50)	airline_id integer	act_departure_time timestamp without time zone	act_arrival_time timestamp without time zone	created_at timestamp without time zone	updated_at timestamp without time zone
1	1	2025-09-28 12:26:18.609531	2025-09-28 10:01:06.71777	228	203	G34	G22	150	2025-10-11 14:27:00.713001	2025-10-31 14:03:15.618924	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
2	2	2025-09-30 15:42:27.206758	2025-10-05 00:58:30.008069	228	203	G60	G84	150	2025-10-10 10:19:04.043316	2025-09-24 18:42:24.27333	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
3	3	2025-10-06 11:04:47.171199	2025-10-10 17:38:00.626649	228	203	G36	G56	150	2025-10-16 02:05:45.025777	2025-10-10 03:13:32.144707	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
4	4	2025-10-16 20:14:59.27743	2025-11-02 13:44:51.762457	228	203	G52	G91	150	2025-10-22 17:26:14.209645	2025-10-21 03:28:39.171613	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
5	5	2025-10-16 14:58:33.160989	2025-09-28 10:42:01.543047	228	203	G91	G86	150	2025-10-17 04:52:22.595113	2025-09-28 02:30:42.373822	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
6	6	2025-10-10 07:45:10.725845	2025-10-19 02:48:45.690694	228	203	G23	G54	150	2025-10-20 00:54:33.408785	2025-10-18 07:06:36.94968	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
7	7	2025-10-09 10:41:23.320238	2025-10-31 22:00:41.799403	228	203	G64	G20	150	2025-09-28 20:57:36.111865	2025-10-16 22:13:26.699055	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
8	8	2025-09-23 21:54:22.681495	2025-09-26 16:35:26.557003	228	203	G60	G15	150	2025-09-28 08:16:06.210953	2025-10-05 18:48:20.162353	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
9	9	2025-10-22 21:58:05.554618	2025-10-25 22:54:34.658662	228	203	G21	G81	150	2025-10-01 05:41:17.531548	2025-10-01 12:09:34.277285	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
10	10	2025-09-28 20:18:12.483113	2025-11-01 11:15:22.699518	228	203	G49	G84	150	2025-10-05 02:58:08.945964	2025-10-01 01:41:50.631377	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
11	11	2025-10-01 13:56:42.557443	2025-09-28 09:48:35.548966	228	203	G74	G89	150	2025-09-29 16:59:23.933231	2025-10-13 01:10:55.414749	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
12	12	2025-10-07 22:14:37.556397	2025-10-31 13:15:26.27224	228	203	G43	G25	150	2025-10-11 16:50:40.932978	2025-10-23 22:54:24.149017	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263
13	13	2025-10-11 09:43:37.111696	2025-10-03 16:54:29.610692	228	203	G53	G41	150	2025-09-28 05:51:18.724662	2025-10-26 21:18:04.106651	2025-09-23 21:33:52.752638	2025-09-23 21:33:52.75263

Total rows: 200 Query complete 00:00:00.209 CRLF Ln 6, Col 64

Create a composite index on the departure\_airport\_id and arriving\_airport\_id columns.

<b>Exercises</b>	<b>Difference</b>	<b>with index</b>	<b>without Index</b>
1	49	146 msec	195 msec
2	109	168 msec	277 msec
3	54	184 msec	238 msec

Evaluate the difference in query performance with and without indexes.  
 Measure performance differences.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* × postgres/postgres@PostgreSQL 17

No limit E

Query History

EXPLAIN ANALYZE  
SELECT \* FROM flights  
WHERE departing\_airport\_id = '228' AND arriving\_airport\_id = '203';

Data Output Messages Notifications

SQL

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

QUERY PLAN text

Seq Scan on flights (cost=0.00..6.00 rows=200 width=72) (actual time=0.051..0.197 rows=200 loops=...  
Filter: ((departing\_airport\_id = 228) AND (arriving\_airport\_id = 203))  
Planning Time: 0.333 ms  
Execution Time: 0.259 ms

Total rows: 4 Query complete 00:00:00.201 ✓ Successfully run. Total query runtime: 201 msec. 4 rows affected. CRLF Ln 3, Col 66

Use EXPLAIN ANALYZE to check index usage in a query filtering by departure\_airport and arrival\_airport.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

```
1 CREATE UNIQUE INDEX idx_passengers_passport_number
2 ON Passengers (passport_number);
3 SELECT indexname, indexdef FROM pg_indexes
4 WHERE tablename = 'passengers' AND indexname = 'idx_passengers_passport_number';
5 INSERT INTO Passengers (first_name, last_name, date_of_birth, gender, country_of_citizenship, country_of_residence, passport_number)
6 VALUES ('Nazar', 'Abishayev', '1979-08-09', 'male', 'Kazakhstan', 'Germany', 'ABC12345');
7 INSERT INTO Passengers (first_name, last_name, date_of_birth, gender, country_of_citizenship, country_of_residence, passport_number)
8 VALUES ('Lyazzat', 'Kabylkhan', '1979-10-04', 'female', 'Kazakhstan', 'Germany', 'ABC12345');
```

Data Output Messages Notifications

ERROR: повторяющееся значение ключа нарушает ограничение уникальности "uq\_passport\_number"  
Ключ "(passport\_number)=(ABC12345)" уже существует.

ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "uq\_passport\_number"  
SQL state: 23505  
Detail: Ключ "(passport\_number)=(ABC12345)" уже существует.

Total rows: Query complete 00:00:00.151 CRLF Ln 7, Col 1

Create a unique index for the passport\_number of the Passengers table. Check if the index was created or not. Insert into the table two new passengers. Explain in your own words what is going on in the output?

The screenshot shows the pgAdmin 4 interface. In the top-left corner, there's a logo and the text "pgAdmin 4". The menu bar includes "File", "Object", "Tools", "Edit", "View", "Window", and "Help". Below the menu is a toolbar with various icons for database management tasks. The main window has two tabs: "Query" (selected) and "Query History". The "Query" tab contains the following SQL code:

```
1 CREATE INDEX idx_passenger_details
2 ON Passengers (first_name, last_name, date_of_birth, country_of_citizenship);
3 explain analyze
4 SELECT first_name, last_name, date_of_birth
5 FROM Passengers
6 WHERE country_of_citizenship = 'Kazakhstan'
7 AND date_of_birth BETWEEN '1979-01-01' AND '1979-12-31';
8 drop index idx_passenger_details
```

The "Data Output" tab below shows the execution plan for the query:

QUERY PLAN
text
1 Seq Scan on passengers (cost=0.00..7.10 rows=1 width=24) (actual time=0.189..0.190 rows=1 loops=1)
2 Filter: ((date_of_birth >= '1979-01-01'::date) AND (date_of_birth <= '1979-12-31'::date) AND ((country_of_citizenship)::text = 'Kazakhstan'::text))
3 Rows Removed by Filter: 176
4 Planning Time: 1.567 ms
5 Execution Time: 0.223 ms

At the bottom of the pgAdmin window, there are status messages: "Total rows: 5" and "Query complete 00:00:00.132". On the right side, there are buttons for "CRLF" and "Ln 3, Col 1".

Create an index for the Passengers table. Use for that first name, last name, date of birth and country of citizenship. Then, write a SQL query to find a passenger who was born in Kazakhstan and was born in 1979 and check if the query uses indexes or not. Give the explanation of the results.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

```
1 SELECT tablename, indexname, indexdef
2 FROM pg_indexes
3 WHERE tablename = 'passengers'
```

Data Output Messages Notifications

SQL

	tablename name	indexname name	indexdef text
1	passengers	passengers_pkey	CREATE UNIQUE INDEX passengers_pkey ON public.passengers USING btree (passenger_id)
2	passengers	uq_passport_number	CREATE UNIQUE INDEX uq_passport_number ON public.passengers USING btree (passport_number)
3	passengers	idx_passengers_passport_number	CREATE UNIQUE INDEX idx_passengers_passport_number ON public.passengers USING btree (passport_number)

Showing rows: 1 to 3 Page No: 1 of 1

Total rows: 3 Query complete 00:00:00.156 ✓ Successfully run. Total query runtime: 156 msec. 3 rows affected. CRLF Ln 3, Col 31

The screenshot shows the pgAdmin 4 interface with the PostgreSQL 17 database selected. In the main query editor, a SQL query is run to list indexes for the 'passengers' table. The results are displayed in a table titled 'pg\_indexes'. The table has columns for index ID, table name, index name, and index definition. Three indexes are listed: a primary key index 'passengers\_pkey' on 'passenger\_id', a unique index 'uq\_passport\_number' on 'passport\_number', and another unique index 'idx\_passengers\_passport\_number' on 'passport\_number'. A success message at the bottom indicates the query was run successfully with a total runtime of 156 msec and 3 rows affected.

Write a SQL query to list indexes for table Passengers. After delete the created indexes.