

# TASK 1

The screenshot shows the pgAdmin 4 interface for PostgreSQL 17. The left sidebar is the Object Explorer, displaying the database structure. The 'Tables' section under 'flights' is expanded, showing 12 columns: 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, and updated\_at. The main pane shows a query window with the following command:

```
CREATE INDEX act_departure_time_index ON flights(act_departure_time)
```

The 'Messages' tab in the bottom right shows the execution results:

```
CREATE INDEX
Query returned successfully in 50 msec.
```

The status bar at the bottom indicates "Query complete 00:00:00.050".

# TASK 2

The screenshot shows the pgAdmin 4 interface. The title bar reads "pgAdmin 4" and "Airport/postgres@PostgreSQL 17\*". The left sidebar is the "Object Explorer" showing a database structure with 10 tables: airline, airport, baggage, baggage\_check, boarding\_pass, booking, booking\_flight, flights, passengers, and security\_check. The "flights" table is currently selected. The main pane contains a query editor with the following SQL command:

```
CREATE UNIQUE INDEX unique_schdep_flightno_index ON flights(flight_no, sch_departure_time)
```

The "Messages" tab in the bottom right shows the results of the query:

```
CREATE INDEX
Query returned successfully in 53 msec.
```

At the bottom, the status bar indicates "Total rows: 0" and "Query complete 00:00:00.053". The system tray at the very bottom shows various icons and the date/time "07.11.2025 16:31".

# TASK 3

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Airport/postgres@PostgreSQL 17\*

Query History

Query

```
CREATE INDEX flights_departure_arrival_index ON flights(departing_airport_id, arriving_airport_id)
```

Scratch Pad

Tables (10)

- airline
- airport
- baggage
- baggage\_check
- boarding\_pass
- booking
- booking\_flight
- flights

Columns (13)

- 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
- flight\_no

Constraints

Indexes

RLS Policies

Rules

Triggers

passengers

security\_check

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 55 msec.

Total rows: Query complete 00:00:00.055 CRLF Ln 1, Col 99

Поиск

16:55 07.11.2025

# TASK 4

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Airport/postgres@PostgreSQL 17\*

Servers (1) PostgreSQL 17 Databases (2) Airport

Query Query History

```
1 SELECT * FROM flights
2 WHERE act_departure_time BETWEEN '2020-01-01' AND '2025-02-01';
```

Data Output Messages Notifications

flight_Id	sch_departure_time	sch_arrival_time	departing_airport_id	arriving_airport_id	departing_gate	arriving_gate	airline_Id	act_dep
1	2022-06-07 09:11:49	2022-06-07 18:11:49	180	185	G10	H1	2	2022-0
2	2021-06-29 14:30:59	2021-06-30 02:30:59	118	15	G16	H19	1	2021-0
3	2023-08-24 22:10:45	2023-08-25 09:10:45	118	15	G21	H20	2	2023-0
4	2022-10-16 14:46:12	2022-10-16 20:46:12	20	117	G22	H13	150	2022-1
5	2022-10-05 10:35:32	2022-10-05 19:35:32	171	35	G19	H18	4	2022-1
6	2021-06-17 05:06:08	2021-06-17 07:06:08	164	198	G14	H3	193	2021-0
...	...	...	...	...	...	...	...	...
Total rows: 173	Query complete 00:00:00.095							

CRLF Ln 2, Col 39

07.11.2025 17:13

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Airport/postgres@PostgreSQL 17\*

Servers (1) PostgreSQL 17 Databases (2) Airport

Query Query History

```
1 SELECT * FROM flights
2 WHERE act_departure_time BETWEEN '2020-01-01' AND '2025-02-01';
```

Data Output Messages Notifications

flight_Id	sch_departure_time	sch_arrival_time	departing_airport_id	arriving_airport_id	departing_gate	arriving_gate	airline_Id	act_dep
1	2022-06-07 09:11:49	2022-06-07 18:11:49	180	185	G10	H1	2	2022-0
2	2021-06-29 14:30:59	2021-06-30 02:30:59	118	15	G16	H19	1	2021-0
3	2023-08-24 22:10:45	2023-08-25 09:10:45	118	15	G21	H20	2	2023-0
4	2022-10-16 14:46:12	2022-10-16 20:46:12	20	117	G22	H13	150	2022-1
5	2022-10-05 10:35:32	2022-10-05 19:35:32	171	35	G19	H18	4	2022-1
6	2021-06-17 05:06:08	2021-06-17 07:06:08	164	198	G14	H3	193	2021-0
...	...	...	...	...	...	...	...	...
Total rows: 173	Query complete 00:00:00.106							

CRLF Ln 2, Col 39

07.11.2025 17:12

WITH INDEX

WITHOUT INDEX

# TASK 5

The screenshot shows the pgAdmin 4 interface for PostgreSQL 17. The left sidebar is the Object Explorer, displaying the database structure under the 'Airport' schema. The main area is a query editor titled 'Airport/postgres@PostgreSQL 17\*' containing the following SQL code:

```
1 EXPLAIN ANALYZE
2 SELECT * FROM flights
3 WHERE departing_airport_id = 1 AND arriving_airport_id = 3;
```

The 'Data Output' tab is selected, showing the query plan and execution details:

QUERY PLAN  
text

1	Index Scan using flights_departure_arrival_index on flights (cost=0.14..8.16 rows=1 width=76) (actual time=0.038..0.038 rows=0 loops=1)
2	Index Cond: ((departing_airport_id = 1) AND (arriving_airport_id = 3))
3	Planning Time: 1.529 ms
4	Execution Time: 0.055 ms

Total rows: 4 Query complete 00:00:00.053

CRLF Ln 4, Col 1

At the bottom, the taskbar shows various system icons and the date/time: 17:18 07.11.2025.

# TASK 6

The screenshot displays two pgAdmin 4 sessions side-by-side.

**Left Session (Airport/postgres@PostgreSQL 17):**

- Query Editor:

```
CREATE UNIQUE INDEX passenger_passport_number_index ON passengers(passport_number)
```
- Data Output:

CREATE INDEX  
Query returned successfully in 54 msec.

**Right Session (public.passengers/Airport/postgres@PostgreSQL 17):**

- Query Editor:

```
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)  
VALUES (201, 'Asel', 'Nurpeis', '1999-09-09', 'Female', 'Kazakhstan', 'Kazakhstan', 'P00001273', NOW(), NOW())
```
- Data Output:

ERROR: duplicate key value violates unique constraint "passenger\_passport\_number\_index"  
Key (passport\_number)=(P00001273) already exists.  
SQL state: 23505  
Detail: Key (passport\_number)=(P00001273) already exists.

CHECKNG

# TASK 7

The screenshot shows the pgAdmin 4 interface with two query panes and an object explorer.

**Object Explorer:**

- Airport/postgres@PostgreSQL 17\*
- public.passengers...
- passenger
- 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
- flight\_no
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- passenger
- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- security\_check
- Trigger Functions
- Types
- Views
- Subscriptions
- postgres

**Query History (Left):**

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

**Data Output (Left):**

CREATE INDEX

Query returned successfully in 55 msec.

**Query History (Right):**

```
EXPLAIN ANALYZE  
SELECT *  
FROM passengers  
WHERE country_of_citizenship = 'Philippines'  
AND EXTRACT(YEAR FROM date_of_birth) = 1984;
```

**Data Output (Right):**

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

text
1 Seq Scan on passengers (cost=0.00..6.50 rows=1 width=64) (actual time=0.021..0.036 rows=1 loops=1)
2 Filter: ((country_of_citizenship)::text = 'Philippines'::text) AND (EXTRACT(year FROM date_of_birth) = 1984::num...
3 Rows Removed by Filter: 199
4 Planning Time: 0.100 ms
5 Execution Time: 0.044 ms

Total rows: 5 Query complete 00:00:00.062

Successfully run. Total query runtime: 62 msec. 5 rows

PostgreSQL used a sequential scan, not the index. The EXTRACT(YEAR FROM date\_of\_birth) function stops the index from working, and the table is small, so a full scan is faster.

# TASK 8

pgAdmin 4

Airport/postgres@PostgreSQL 17\* | public.passengers... | Airport/postgres@PostgreSQL 17 | public.passengers... | Scratch Pad

File Object Tools Edit View Window Help

Object Explorer

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  
flight\_no  
nstraints  
lexes  
S Policies  
les  
ggers  
engers  
lums  
nstraints  
lexes  
S Policies  
les  
ggers  
city\_check  
Functions

Query Query History

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

Data Output Messages Notifications

SQL

Showing rows: 1 to 5 | Page No: 1 of 1 | < > << >>

indexname	indexdef
passenger_pkey	CREATE UNIQUE INDEX passengers_pkey ON public.passengers USING btree (passenger_id)
passenger_passport_number_key	CREATE UNIQUE INDEX passengers_passport_number_key ON public.passengers USING btree (passport_number)
unique_passport	CREATE UNIQUE INDEX unique_passport ON public.passengers USING btree (passport_number)
passenger_passport_number_index	CREATE INDEX passenger_passport_number_index ON public.passengers USING btree (passport_number)
passenger_name_birth_country_index	CREATE INDEX passengers_name_birth_country_index ON public.passengers USING btree (first_name, last_name, date_of_birth, country_of_citiz...

Total rows: 5 | Query complete 00:00:00.077 | CRLF | Ln 8, Col 1

Airport/postgres@PostgreSQL 17\* | public.passengers... | Airport/postgres@PostgreSQL 17 | public.passengers... | Scratch Pad

File Object Tools Edit View Window Help

Object Explorer

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  
flight\_no  
> Constraints  
> Indexes  
> RLS Policies  
> Rules  
> Triggers  
> passengers  
> Columns  
> Constraints  
> Indexes  
> RLS Policies  
> Rules  
> Triggers  
> security\_check  
> Trigger Functions  
> Types  
> Views  
> Subscriptions  
> postgres

Query Query History

```
1 DROP INDEX IF EXISTS passenger_passport_number_index;
2 DROP INDEX IF EXISTS passengers_name_birth_country_index;
```

Data Output Messages Notifications

DROP INDEX

Query returned successfully in 56 msec.

Total rows: | Query complete 00:00:00.056 | CRLF | ENG