

1 task

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure for 'MY DB' > 'Servers (1)' > 'PostgreSQL 17' > 'Databases (6)' > 'airport_2'. The 'public' schema is expanded, showing various objects like Aggregates, Collations, Domains, etc. The main query editor window is titled 'airport_2/postgres@PostgreSQL 17' and contains the following SQL query:

```
--lab7
--1
CREATE INDEX idx_act_depp on flights (actual_departure);
```

The 'Data Output' tab is selected, showing the message: 'Query returned successfully in 64 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.064'. A notification box on the right states: 'You are currently running version 9.6 of pgAdmin 4, however the current version is 9.9. Please click [here](#) for more information.'

2 task

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure for 'MY DB' > 'Servers (1)' > 'PostgreSQL 17' > 'Databases (6)' > 'airport_2'. The 'public' schema is expanded, showing various objects like Aggregates, Collations, Domains, etc. The main query editor window is titled 'airport_2/postgres@PostgreSQL 17' and contains the following SQL query:

```
--2
CREATE UNIQUE INDEX idx_fsck on flights (flight_no, scheduled_departure)
```

The 'Data Output' tab is selected, showing an error message: 'ERROR: создать уникальный индекс "idx_fsck" не удалось. Ключ (flight_no, scheduled_departure)=(US-KS, 2023-09-04) дублируется. ОШИБКА: создать уникальный индекс "idx_fsck" не удалось. SQL state: 23505. Detail: Ключ (flight_no, scheduled_departure)=(US-KS, 2023-09-04) дублируется.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.066'. A notification box on the right states: 'You are currently running version 9.6 of pgAdmin 4, however the current version is 9.9. Please click [here](#) for more information.'

3 task

The screenshot shows the pgAdmin 4 interface with the 'airports/postgres@PostgreSQL 17' connection selected. The 'airports' database is expanded, showing the 'flights' table with 14 columns. The 'Query' tab is active, displaying the following SQL commands:

```
--2
CREATE UNIQUE INDEX idx_fsch ON flights (flight_no, scheduled_departure);

--3
CREATE INDEX idx_dep_arr ON flights (departure_airport_id, arrival_airport_id);
```

The 'Data Output' tab shows the message: 'CREATE INDEX' and 'Query returned successfully in 60 msec.' A notification box on the right states: 'Query returned successfully in 60 msec.' and 'You are currently running version 9.6 of pgAdmin 4, however the current version is 9.9. Please click [here](#) for more information.'

Total rows: Query complete 00:00:00.060 CRLF Ln 5713, Col 1

4 task

The screenshot shows the pgAdmin 4 interface with the 'airports/postgres@PostgreSQL 17' connection selected. The 'airports' database is expanded, showing the 'flights' table. The 'Query' tab is active, displaying the following SQL commands:

```
--4
CREATE INDEX idx_dep_arr ON flights (departure_airport_id, arrival_airport_id);

--5
SELECT * FROM flights WHERE actual_departure = '2023-09-04';
```

The 'Data Output' tab shows the results of the query:

flight_id	flight_no	scheduled_departure	scheduled_arrival	departure_airport_id	arrival_airport_id	departing_gate	arriving_gate	airline_id	status
1	78 US-VT	2023-08-28	2023-08-02	20	11	335	515	1	Delayed
2	454 US-WV	2023-06-10	2023-07-18	1	7	1987	45	42	Delayed
3	920 KR-11	2023-05-15	2023-06-23	16	13	59	244	5	Delayed

Total rows: 3 Query complete 00:00:00.123 CRLF Ln 5712, Col 1

Successfully run. Total query runtime: 123 msec. 3 rows affected.

pgAdmin 4

Object Explorer: Servers (1) > PostgreSQL 17 > Databases (6) > airport_2 > Schemas (1) > public > Tables (10)

Query:

```
--3
CREATE INDEX idx_dep_arr ON flights (departure_airport_id, arrival_airport_id);
--4
CREATE INDEX idx_ac_dep ON flights(actual_departure);
SELECT * FROM flights WHERE actual_departure = '2023-09-04';
```

Data Output:

flight_id	flight_no	flight_no	scheduled_departure	scheduled_arrival	departure_airport_id	arrival_airport_id	departing_gate	arriving_gate	airline_id	status
[PK] integer	character varying (50)	character varying (50)	date	date	integer	integer	character varying (50)	character varying (50)	integer	character varying (10)
1	78	US-VT	2023-08-28	2023-08-02	20	11	335	515	1	Delayed
2	454	US-WV	2023-06-10	2023-07-18	1	7	1987	45	42	Delayed
3	920	KR-11	2023-05-15	2023-06-23	16	13	59	244	5	Delayed

Successfully run. Total query runtime: 118 msec. 3 rows affected.

5 task

pgAdmin 4

Object Explorer: Tables (10) > flights > Columns (14)

Query:

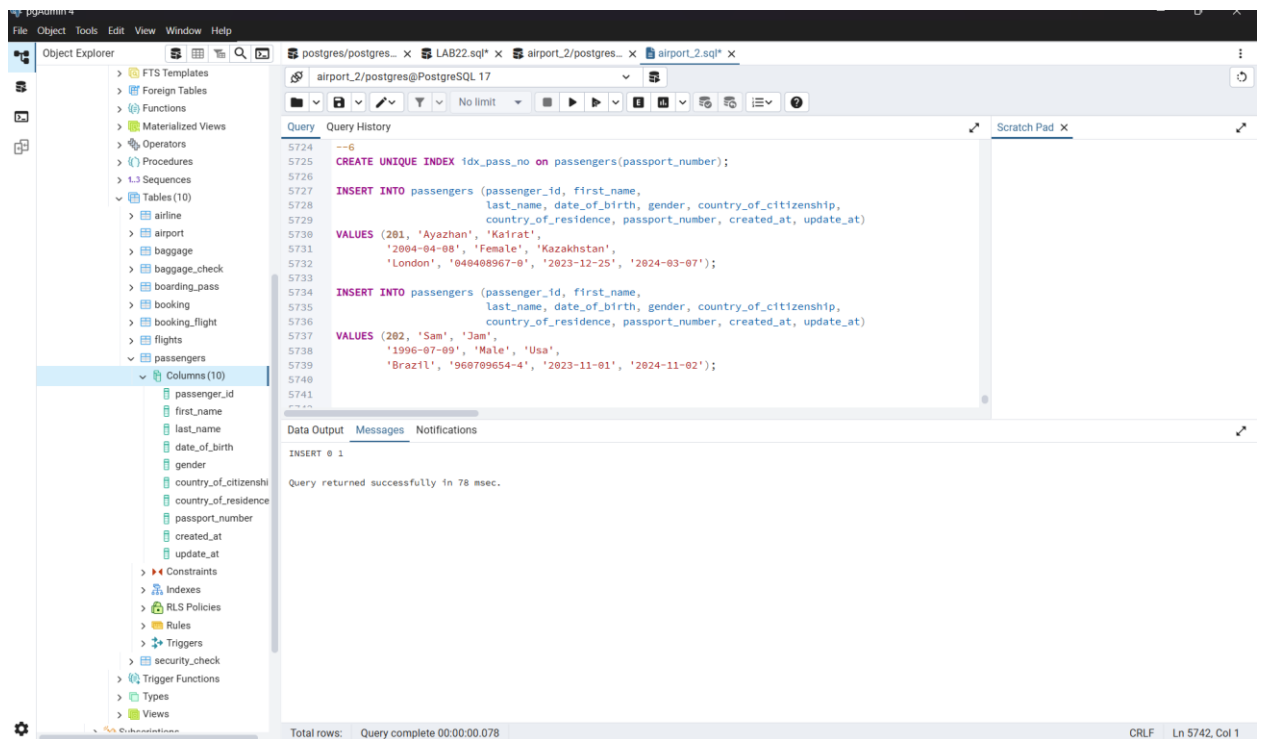
```
WHERE actual_departure = '2023-09-04';
--5
EXPLAIN ANALYZE
SELECT * FROM flights
WHERE departure_airport_id = 15
AND arrival_airport_id = 18
```

QUERY PLAN

```
test
1  Bitmap Heap Scan on flights  (cost=4.30..9.97 rows=2 width=61) (actual time=0.084..0.089 rows=5 loops=1)
2    Recheck Cond: ((departure_airport_id = 15) AND (arrival_airport_id = 18))
3    Heap Blocks: exact=5
4    -> Bitmap Index Scan on idx_dep_arr  (cost=0.00..4.29 rows=2 width=0) (actual time=0.073..0.073 rows=5 loops=1)
5          Index Cond: ((departure_airport_id = 15) AND (arrival_airport_id = 18))
6  Planning Time: 0.122 ms
7  Execution Time: 0.112 ms
```

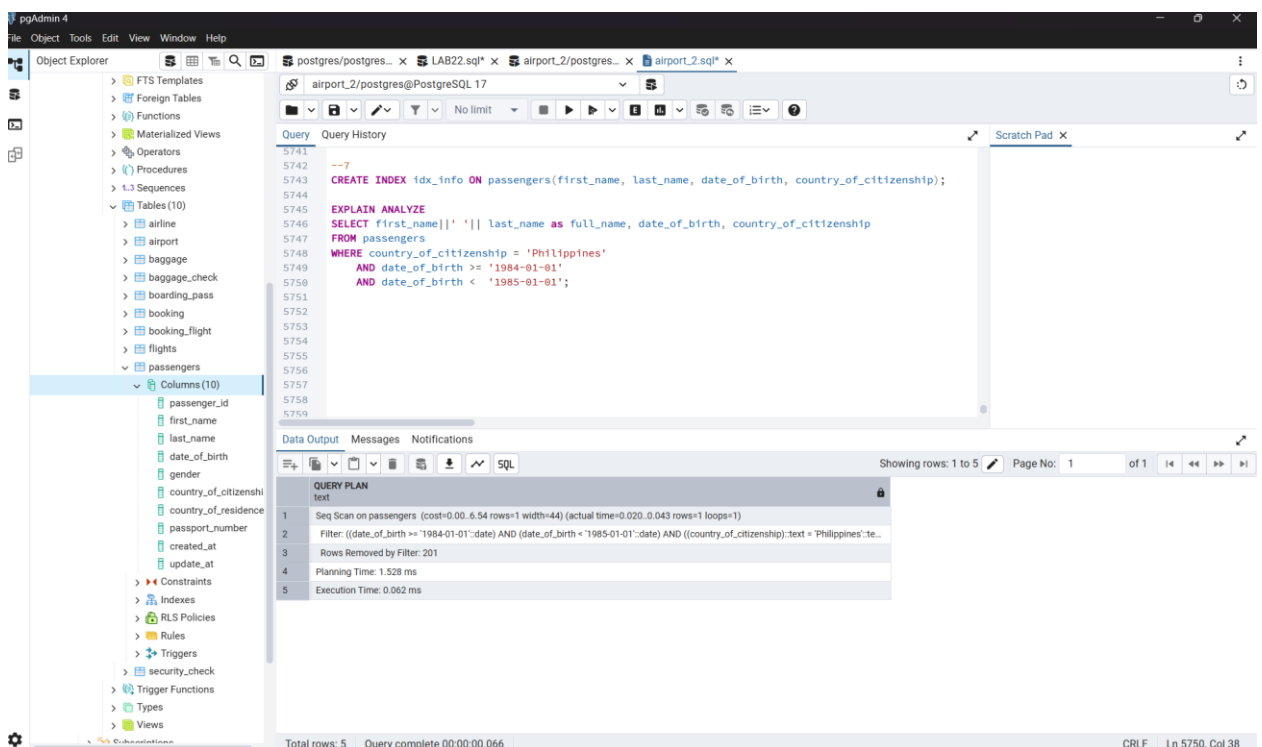
Total rows: 7 Query complete 00:00:00.142

6 task



When I add a UNIQUE index to the `passport_number` column in the `Passengers` table, the database guarantees that no two rows can have the same passport number. This means every passport number must be different. After creating the index, I check whether it was successfully created. Then I insert two new rows with different passport numbers, and the operation works without any problems. However, if I try to insert rows with the same passport number, the database returns an error. This error indicates that the uniqueness constraint has been violated because the passport number already exists in the table.

7 task



First, we created a composite index on the columns `country_of_citizenship`, `date_of_birth`, `first_name`, and `last_name`. These columns were chosen because the query filters by `country_of_citizenship` and `date_of_birth`, so placing them first in the index improves performance. The SQL query searches for passengers who are citizens of the Philippines and were born in 1984. To allow PostgreSQL to use the index efficiently, the date condition was written as a range:

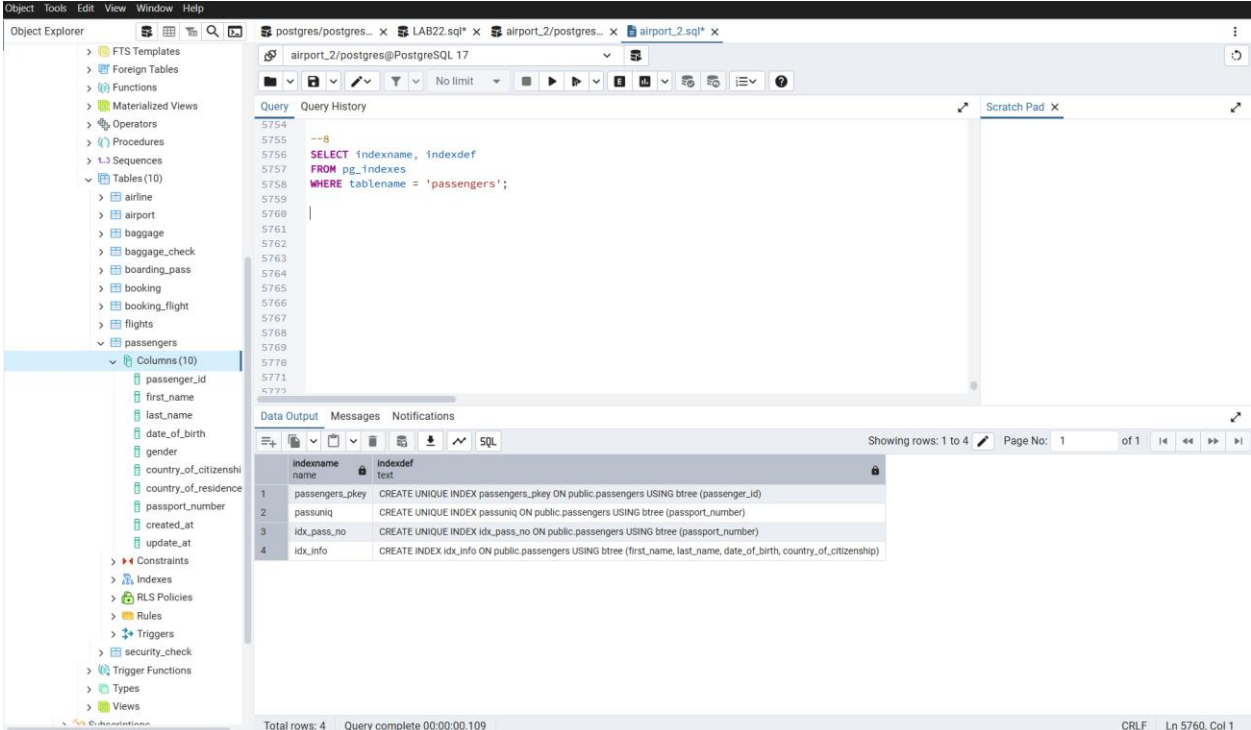
```
date_of_birth >= '1984-01-01'
```

```
AND date_of_birth < '1985-01-01'
```

When running the query with `EXPLAIN ANALYZE`, PostgreSQL performs an **Index Scan**, which confirms that the newly created index is actually used. If the date had been written using a function such as `EXTRACT(YEAR FROM date_of_birth)`, PostgreSQL would instead perform a **Sequential Scan**, because functions applied to indexed columns prevent index usage.

Therefore, the query with the date range and the composite index provides the most efficient execution plan.

8 task



The screenshot shows a PostgreSQL database management tool interface. On the left, the 'Object Explorer' pane displays the database structure, including tables, functions, and views. The 'passengers' table is selected, showing its columns: `passenger_id`, `first_name`, `last_name`, `date_of_birth`, `gender`, `country_of_citizenship`, `country_of_residence`, `passport_number`, `created_at`, and `update_at`.

The main query editor displays the following SQL query:

```
--8
SELECT indexname, indexdef
FROM pg_indexes
WHERE tablename = 'passengers';
```

The 'Data Output' pane shows the results of the query, listing the index names and their definitions for the 'passengers' table:

indexname	indexdef
passengers_pkey	CREATE UNIQUE INDEX passengers_pkey ON public.passengers USING btree (passenger_id)
passuniq	CREATE UNIQUE INDEX passuniq ON public.passengers USING btree (passport_number)
idx_pass_no	CREATE UNIQUE INDEX idx_pass_no ON public.passengers USING btree (passport_number)
idx_info	CREATE INDEX idx_info ON public.passengers USING btree (first_name, last_name, date_of_birth, country_of_citizenship)

The bottom status bar indicates 'Total rows: 4' and 'Query complete 00:00:00.109'.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- > Tables (10)
 - airline
 - airport
 - baggage
 - baggage_check
 - boarding_pass
 - booking
 - booking_flight
 - flights
 - 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
 - Trigger Functions
 - Types
 - Views

Query

```
--8
SELECT indexname, indexdef
FROM pg_indexes
WHERE tablename = 'passengers';

DROP INDEX idx_info, idx_pass_no;
```

Query History

Scratch Pad

Data Output Messages Notifications

DROP INDEX

Query returned successfully in 92 msec.

Total rows: Query complete 00:00:00.092

CRLF Ln 5767, Col 1