

The screenshot shows the DBeaver application interface. On the left is a tree view of 'Servers' containing two entries: 'PostgreSQL 17' and 'PostgreSQL 18'. The 'PostgreSQL 17' node is expanded, showing 'Databases (4)', 'Login/Group Roles', and 'Tablespaces'. The 'PostgreSQL 18' node is collapsed. The main area is a query editor titled 'airport_database/postgres@PostgreSQL 17*'. The query window contains the following SQL code:

```
1 CREATE INDEX idx_flights_actual_departure
2 ON flights (actual_departure);
```

The 'Messages' tab in the bottom panel displays the execution results:

```
CREATE INDEX
Query returned successfully in 2 secs 45 msec.
```

At the bottom, status information includes 'Total rows:' and 'Query complete 00:00:02.045'.

The screenshot shows the DBeaver interface with the following details:

- Servers:** Servers (2) - PostgreSQL 17 (selected), PostgreSQL 18.
- Databases:** Databases (4) - airport_database (selected).
- Query Bar:** Shows the connection information: airport_database/postgres@PostgreSQL 17. Includes a toolbar with various icons for file operations, search, and navigation.
- Query Tab:** Contains the SQL code for creating a unique index:

```
1 CREATE UNIQUE INDEX idx_flights_flight_no_scheduled_departure
2 ON flights (flight_no, scheduled_departure);
3
```
- Data Output Tab:** Displays the results of the query:

```
CREATE INDEX
Query returned successfully in 879 msec.
```
- Messages Tab:** Shows a green notification bar at the bottom right: ✓ Query returned successfully in 879 msec. X
- Bottom Status:** Total rows: 0 | Query complete 00:00:00.879 | LF | Ln 3, Col 1

The screenshot shows the DBeaver database management tool interface. The left sidebar displays the 'Servers' section with two entries: 'PostgreSQL 17' (selected) and 'PostgreSQL 18'. The main workspace is a query editor titled 'airport_database/postgres@PostgreSQL 17'. The query window contains the following SQL code:

```
1 CREATE INDEX idx_flights_departure_arrival
2 ON flights (departure_airport_id, arrival_airport_id);
```

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

```
CREATE INDEX
Query returned successfully in 1 secs 715 msec.
```

A green status bar at the bottom right indicates: ✓ Query returned successfully in 1 secs 715 msec. X. The bottom navigation bar shows 'Total rows: 0' and 'Query complete 00:00:01.715'.

airport_database/postgres@PostgreSQL 17* X

Servers (2) airport_database/postgres@PostgreSQL 17

Databases (4) No limit

Login/Group Roles Tablespaces

PostgreSQL 18

Query History

```
1 -- 4.1 Query by actual_departure
2 EXPLAIN ANALYZE
3 SELECT *
4 FROM flights
5 WHERE actual_departure = '2024-01-01';
6
7 -- 4.2 Query by flight_no and scheduled_departure
8 EXPLAIN ANALYZE
9 SELECT *
10 FROM flights
11 WHERE flight_no = 'KC123'
12 AND scheduled_departure = '2024-02-10';
13
14 -- 4.3 Query by departure_airport_id and arrival_airport_id
15 EXPLAIN ANALYZE
16 SELECT *
17 FROM flights
18 WHERE departure_airport_id = 1
```

Data Output Messages Notifications

SQL

Showing rows: 1 to 7 | Page No: 1 of 1 | < << > >> | ↻

	QUERY PLAN
1	Bitmap Heap Scan on flights (cost=4.30..9.97 rows=2 width=61) (actual time=0.032..0.058 rows=2 loops=1)
2	Recheck Cond: ((departure_airport_id = 1) AND (arrival_airport_id = 2))
3	Heap Blocks: exact=2
4	-> Bitmap Index Scan on idx_flights_departure_arrival (cost=0.00..4.29 rows=2 width=0) (actual time=0.027..0.027 rows=2 loops=1)
5	Index Cond: ((departure_airport_id = 1) AND (arrival_airport_id = 2))
6	Planning Time: 0.061 ms
7	Execution Time: 0.070 ms

Total rows: 7 Query complete 00:00:00.330

✓ Successfully run. Total query runtime: 330 msec. 7 rows affected. ✘

LF | Ln 12, Col 42

airport_database/postgres@PostgreSQL 17*

Servers (2) PostgreSQL 17 Databases (4) Login/Group Roles Tablespaces PostgreSQL 18

EXPLAIN ANALYZE
SELECT *
FROM flights
WHERE departure_airport_id = 1
AND arrival_airport_id = 2;

Data Output Messages Notifications

Showing rows: 1 to 7 Page No: 1 of 1

	QUERY PLAN
1	Bitmap Heap Scan on flights (cost=4.30..9.97 rows=2 width=61) (actual time=0.028..0.030 rows=2 loops=1)
2	Recheck Cond: ((departure_airport_id = 1) AND (arrival_airport_id = 2))
3	Heap Blocks: exact=2
4	-> Bitmap Index Scan on idx_flights_departure_arrival (cost=0.00..4.29 rows=2 width=0) (actual time=0.018..0.018 rows=2 loops=1)
5	Index Cond: ((departure_airport_id = 1) AND (arrival_airport_id = 2))
6	Planning Time: 0.117 ms
7	Execution Time: 0.053 ms

Total rows: 7 Query complete 00:00:02.036 ✓ Successfully run. Total query runtime: 2 secs 36 msec. 7 rows affected. LF Ln 5, Col 32

airport_database/postgres@PostgreSQL 17* X

Servers (2) airport_database/postgres@PostgreSQL 17

Databases (4) No limit

Login/Group Roles Tablespaces

PostgreSQL 18

Query History

```
1 CREATE UNIQUE INDEX idx_passengers_passport_number
2 ON passengers (passport_number);
3
4 -- check
5 SELECT indexname, indexdef
6 FROM pg_indexes
7 WHERE tablename = 'passengers'
8 AND indexname = 'idx_passengers_passport_number';
9
```

Data Output Messages Notifications

SQL

	indexname name	indexdef text
1	idx_passengers_passport_num...	CREATE UNIQUE INDEX idx_passengers_passport_number ON public.passengers USING btree (passport_nu...

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

Total rows: 1 Query complete 00:00:00.835

Successfully run. Total query runtime: 835 msec. 1 rows affected. ✓

LF | Ln 9, Col 1

airport_database/postgres@PostgreSQL 17*

Servers (2) PostgreSQL 17 Databases (4) Login/Group Roles Tablespaces PostgreSQL 18

Query History

```
1 CREATE INDEX idx_passengers_name_dob_citizenship
2 ON passengers (first_name, last_name, date_of_birth, country_of_citizenship);
3
4 --ANALYZE
5 EXPLAIN ANALYZE
6 SELECT
7     passenger_id,
8     first_name,
9     last_name,
10    date_of_birth,
11    country_of_citizenship
12 FROM passengers
13 WHERE country_of_citizenship = 'Philippines'
14   AND date_of_birth BETWEEN '1984-01-01' AND '1984-12-31';
15
```

Data Output Messages Notifications

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

	QUERY PLAN	text
1	Seq Scan on passengers	(cost=0.00..6.50 rows=1 width=31) (actual time=0.014..0.056 rows=1 loops=1)
2	Filter:	((date_of_birth >= '1984-01-01'::date) AND (date_of_birth <= '1984-12-31'::date) AND ((country_of_citizenship)::text = 'Philippines'::text))
3	Rows Removed by Filter:	199
4	Planning Time:	0.462 ms
5	Execution Time:	0.069 ms

Total rows: 5 | Query complete 00:00:00.751 | LF | Ln 4, Col 10

Successfully run. Total query runtime: 751 msec. 5 rows affected.

The screenshot shows the DBeaver PostgreSQL client interface. The left sidebar displays the 'Servers' section with two entries: 'PostgreSQL 17' (selected) and 'PostgreSQL 18'. The main area is a query editor titled 'airport_database/postgres@PostgreSQL 17'. It contains the following SQL code:

```
1 SELECT indexname, indexdef
2 FROM pg_indexes
3 WHERE tablename = 'passengers';
4
5 DROP INDEX IF EXISTS idx_passengers_passport_number;
6 DROP INDEX IF EXISTS idx_passengers_name_dob_citizenship;
7
```

The 'Messages' tab at the bottom shows the execution results:

```
DROP INDEX
Query returned successfully in 1 secs 746 msec.
```

At the bottom status bar, it says 'Total rows: 0' and 'Query complete 00:00:01.746'.