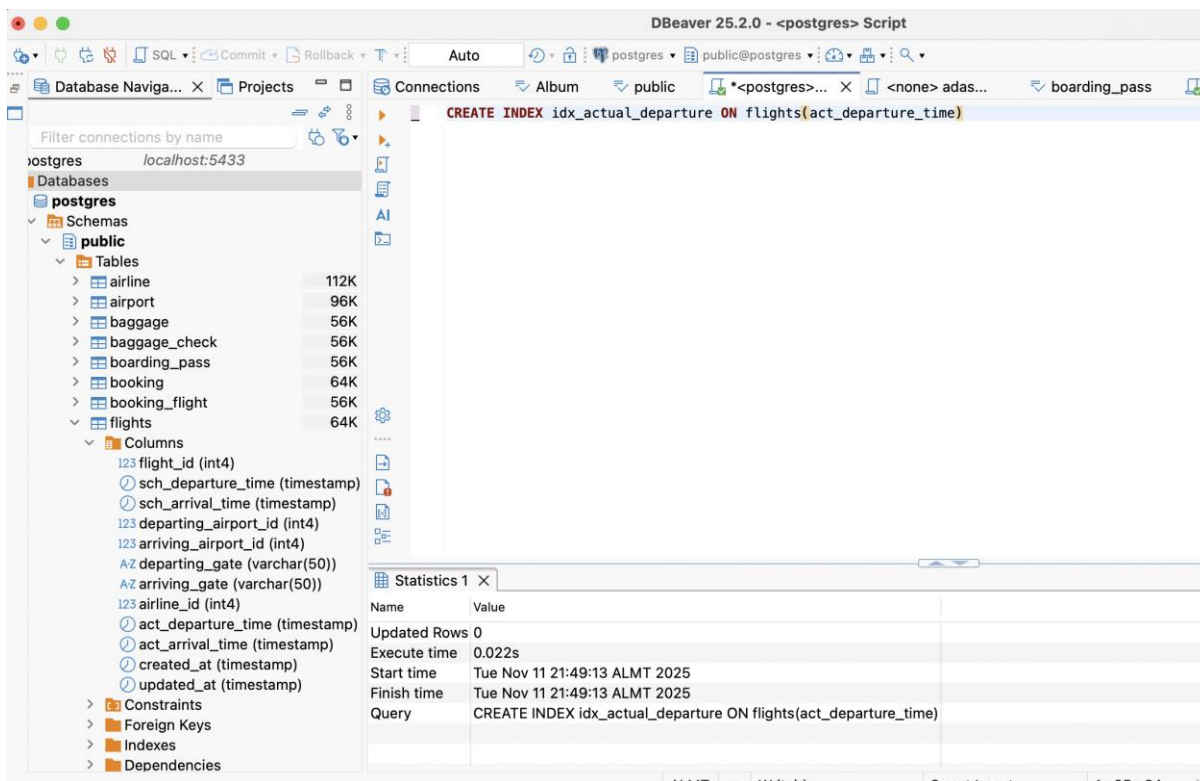
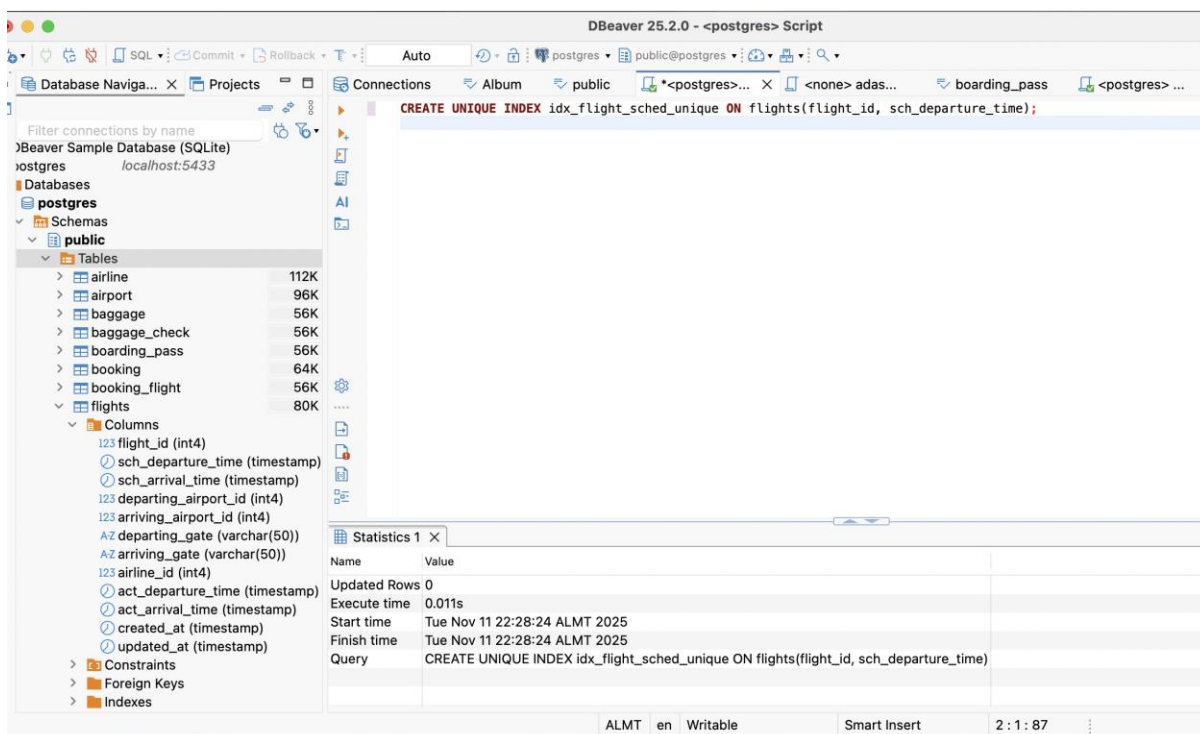


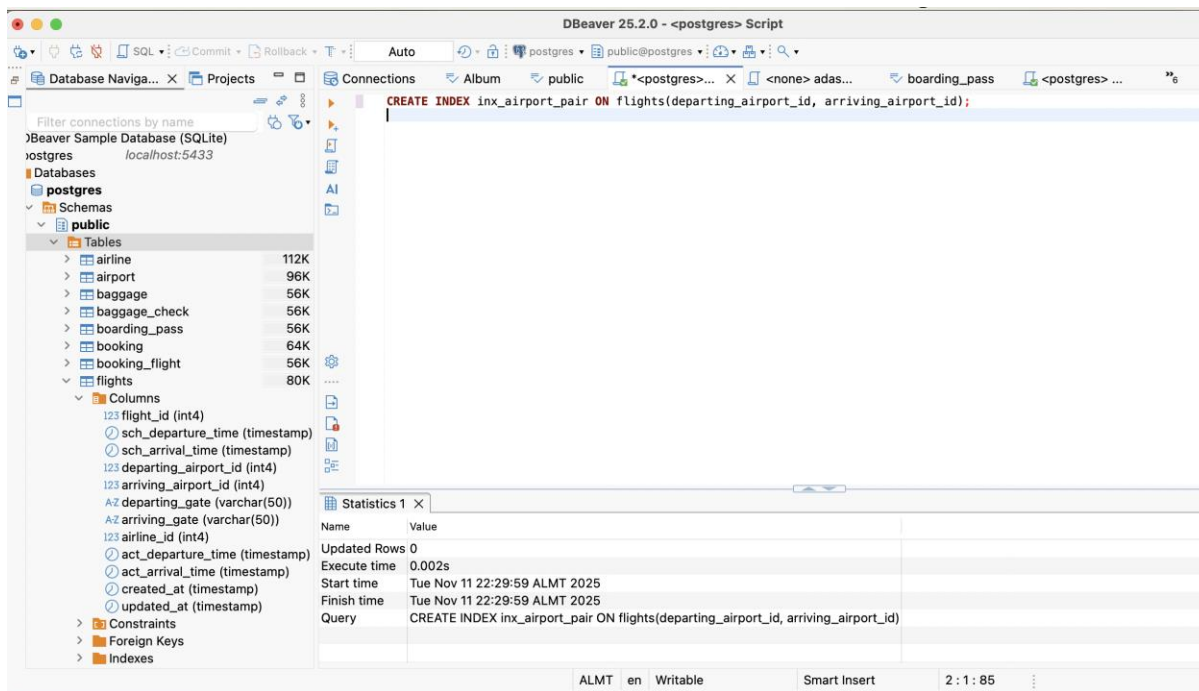
1. Create an index on the actual\_departure column in the flights table.



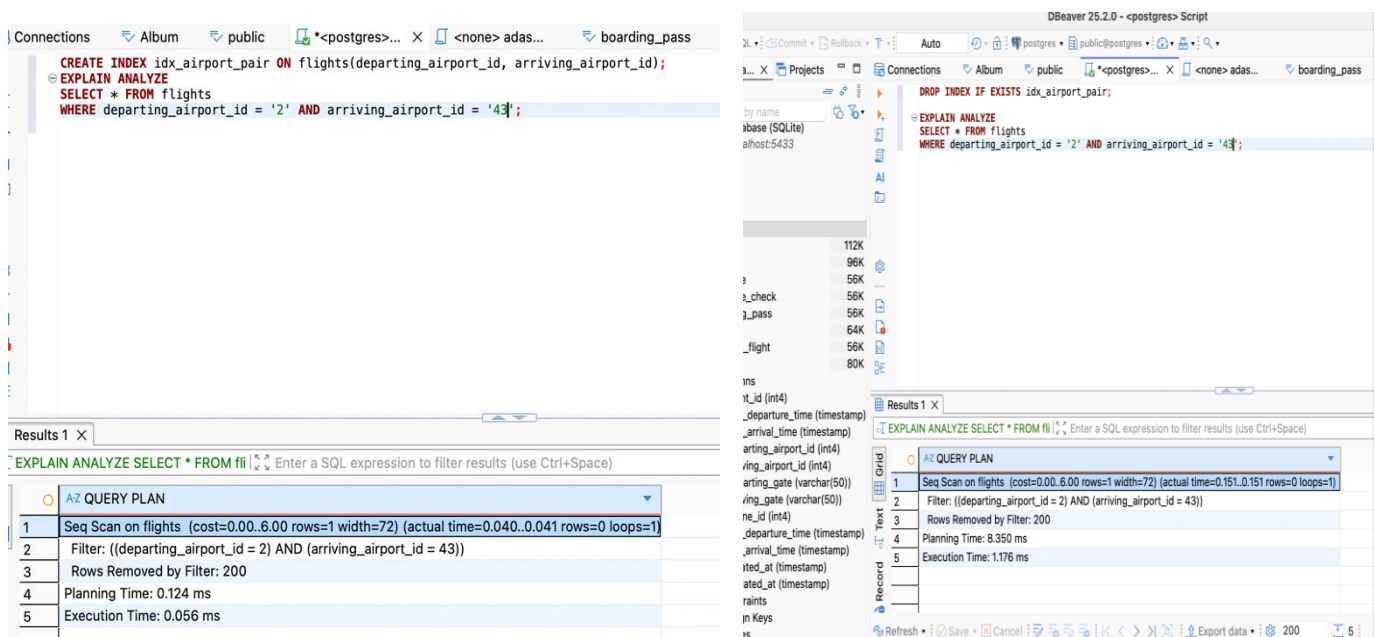
2. Create a unique index to ensure flight\_no and scheduled\_departure combinations are unique.



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



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



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

The screenshot shows the DBeaver 25.2.0 interface with a PostgreSQL database connection. The left sidebar displays the database structure, including the 'public' schema and the 'flights' table (128K rows). The main window shows the SQL editor with the following query:

```
EXPLAIN ANALYZE
SELECT * FROM flights
WHERE departing_airport_id = '2' AND arriving_airport_id = '43';
```

The query results are displayed in the 'Results 1' tab, showing the 'AZ QUERY PLAN' and the execution details:

Step	Operation	Cost	Rows	Width	Actual Time	Actual Rows	Actual Loops
1	Seq Scan on flights	(cost=0.00..6.00 rows=1 width=72)	1	72	(actual time=0.052..0.053 rows=0 loops=1)	0	1
2	Filter: ((departing_airport_id = 2) AND (arriving_airport_id = 43))						
3	Rows Removed by Filter: 200						
4	Planning Time: 1.092 ms						
5	Execution Time: 0.104 ms						

The status bar at the bottom indicates that 5 row(s) were fetched in 0.007s (0.001s fetch) on 2025-11-12 at 00:12:04.

6. 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?

DBEaver 25.2.0 - <postgres> Script

SQL Commit Rollback Auto postgres public@postgres public@postgres

Database Naviga... Projects Connections Album public \*<postgres>... <none> adas... <postgres>

Filter connections by name

Databases

postgres

Schemas

public

Tables

- airline 112K
- airport 96K
- baggage 56K
- baggage\_check 56K
- boarding\_pass 56K
- booking 64K
- booking\_flight 56K
- flights 128K

Columns

- flight\_id (int4)
- sch\_departure\_time (timestamp)
- sch\_arrival\_time (timestamp)
- departing\_airport\_id (int4)
- arriving\_airport\_id (int4)
- departing\_gate (varchar(50))
- arriving\_gate (varchar(50))
- airline\_id (int4)
- act\_departure\_time (timestamp)
- act\_arrival\_time (timestamp)
- created\_at (timestamp)
- updated\_at (timestamp)

Constraints

Foreign Keys

Indexes

CREATE UNIQUE INDEX idx\_passport\_unique ON passengers(passport\_number);

Statistics 1

Name	Value
Updated Rows	0
Execute time	0.013s
Start time	Wed Nov 12 00:12:25 ALMT 2025
Finish time	Wed Nov 12 00:12:25 ALMT 2025
Query	CREATE UNIQUE INDEX idx_passport_unique ON passengers(passport_number)

7. 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 Philippines and was born in 1984 and check if the query uses indexes or not. Give the explanation of the results.

The image consists of two screenshots of a database management tool interface, likely DBeaver, showing SQL queries and their execution results.

**Top Screenshot:**

- SQL Editor:** Contains the following SQL query:
 

```
CREATE INDEX idx_passenger_data ON passengers(first_name, last_name, date_of_birth, country_of_citizenship);
EXPLAIN ANALYZE
SELECT * FROM passengers
WHERE country_of_citizenship = 'Philippines' AND date_of_birth = '1984-01-01';
```
- Results 1 X:** Shows the query plan for the SELECT statement.
 

Step	Operation	Cost	Time	Rows	Width	Loops
1	Seq Scan on passengers	cost=0.00..6.00	rows=1 width=72	actual time=0.059..0.060	rows=0	loops=1
2	Filter: (((country_of_citizenship)::text = 'Philippines'::text) AND (date_of_birth = '1984-01-01'::date))					
3	Rows Removed by Filter:			200		
4	Planning Time:		0.121 ms			
5	Execution Time:		0.075 ms			

**Bottom Screenshot:**

- SQL Editor:** Contains the same SQL query as the top screenshot.
- Statistics 1 X:** Shows the execution statistics for the query.
 

Name	Value
Updated Rows	0
Execute time	0.008s
Start time	Wed Nov 12 00:13:41 ALMT 2025
Finish time	Wed Nov 12 00:13:41 ALMT 2025
Query	CREATE INDEX idx_passenger_data ON passengers(first_name, last_name, date_of_birth, country_of_citizenship)

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



SELECT indexname, indexdef  
FROM pg\_indexes  
WHERE tablename = 'passengers';

pg\_indexes 1 X

SELECT indexname, indexdef FROM pg\_indexes

	AZ indexname	AZ indexdef
1	pk_passengers	CREATE UNIQUE INDEX pk_passengers ON public.passengers USING btree (passenger_id)
2	idx_passport_unique	CREATE UNIQUE INDEX idx_passport_unique ON public.passengers USING btree (passport_number)
3	idx_passenger_data	CREATE INDEX idx_passenger_data ON public.passengers USING btree (first_name, last_name, date_of_birth, count)

Refresh Save Cancel Export data 200 3

3 row(s) fetched - 0.013s, on 2025-11-12 at 00:15:33

ALMT en Writable Smart Insert 4 : 1 : 75

мне нужно

DBeaver 25.2.0 - <postgres> Script

SQL Commit Rollback Auto postgres public@postgres

base Naviga... X Projects

Connections by name

- act\_arrival\_time (timestamp)
- created\_at (timestamp)
- updated\_at (timestamp)
- Constraints
- Foreign Keys
- Indexes
- Dependencies
- References
- Partitions
- Triggers
- Rules
- Policies
- passengers 72K
- Columns
- passenger\_id (int4)
- first\_name (varchar(50))
- last\_name (varchar(50))
- date\_of\_birth (date)
- gender (varchar(50))
- country\_of\_citizenship (varchar(50))
- country\_of\_residence (varchar(50))
- passport\_number (varchar(20))
- created\_at (timestamp)
- updated\_at (timestamp)
- Constraints
- Foreign Keys
- Indexes
- Dependencies
- References
- Partitions
- Triggers

SELECT indexname, indexdef  
FROM pg\_indexes  
WHERE tablename = 'passengers';  
DROP INDEX IF EXISTS idx\_passport\_unique;

Statistics 1 X

Name	Value
Updated Rows	0
Execute time	0.008s
Start time	Wed Nov 12 00:16:43 ALMT 2025
Finish time	Wed Nov 12 00:16:43 ALMT 2025
Query	DROP INDEX IF EXISTS idx_passport_unique