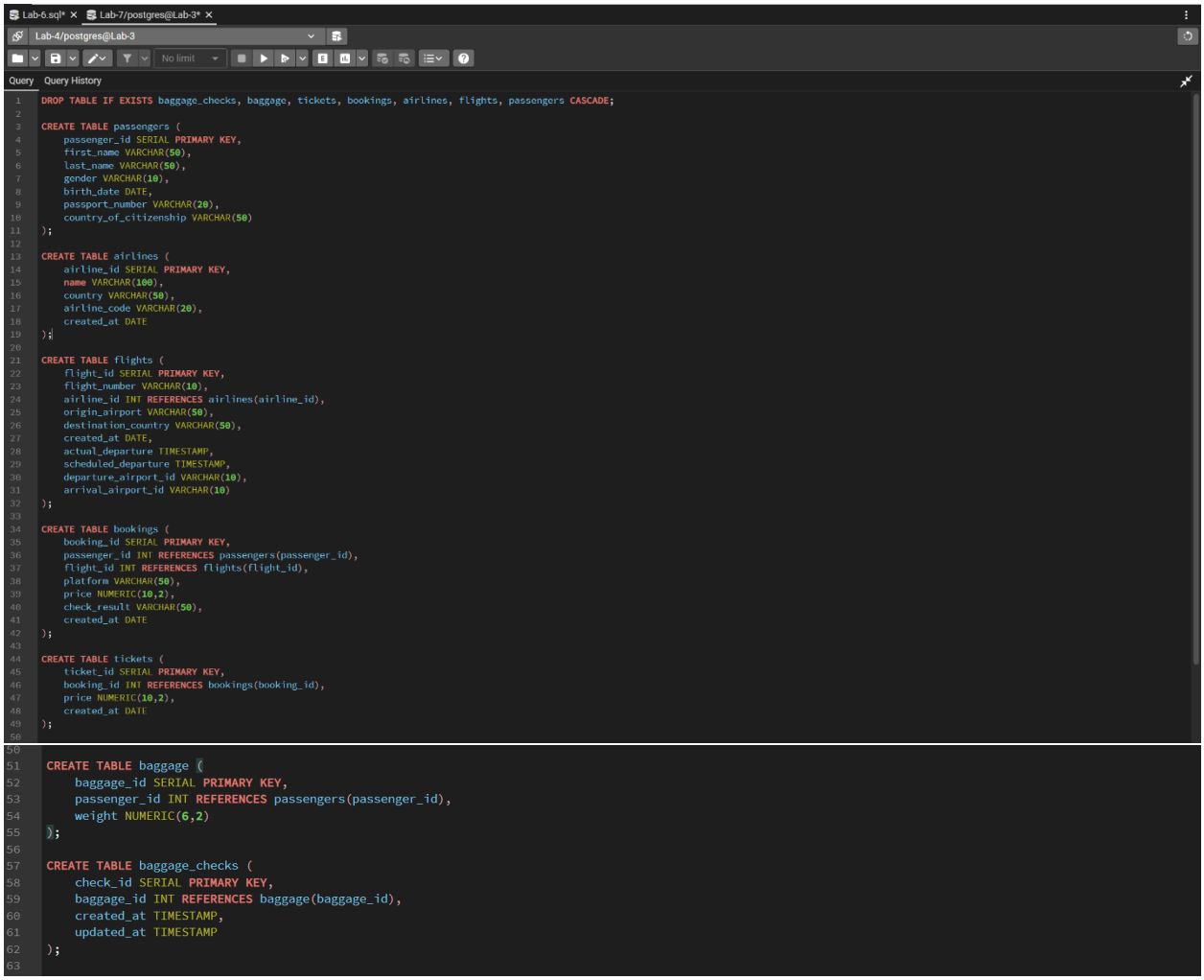
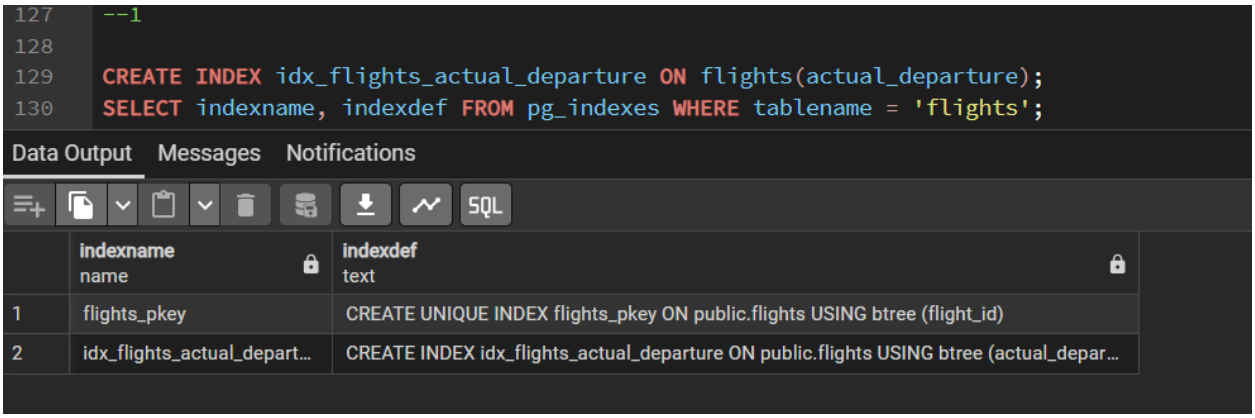
Lab-7



```sql
DROP TABLE IF EXISTS baggage_checks, baggage, tickets, bookings, airlines, flights, passengers CASCADE;

CREATE TABLE passengers (
    passenger_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    gender VARCHAR(10),
    birth_date DATE,
    passport_number VARCHAR(20),
    country_of_citizenship VARCHAR(50)
);

CREATE TABLE airlines (
    airline_id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    country VARCHAR(50),
    airline_code VARCHAR(20),
    created_at DATE
);

CREATE TABLE flights (
    flight_id SERIAL PRIMARY KEY,
    flight_number VARCHAR(10),
    airline_id INT REFERENCES airlines(airline_id),
    origin_airport VARCHAR(50),
    destination_country VARCHAR(50),
    created_at DATE,
    actual_departure TIMESTAMP,
    scheduled_departure TIMESTAMP,
    departure_airport_id VARCHAR(10),
    arrival_airport_id VARCHAR(10)
);

CREATE TABLE bookings (
    booking_id SERIAL PRIMARY KEY,
    passenger_id INT REFERENCES passengers(passenger_id),
    flight_id INT REFERENCES flights(flight_id),
    platform VARCHAR(50),
    price NUMERIC(10,2),
    check_result VARCHAR(50),
    created_at DATE
);

CREATE TABLE tickets (
    ticket_id SERIAL PRIMARY KEY,
    booking_id INT REFERENCES bookings(booking_id),
    price NUMERIC(10,2),
    created_at DATE
);

CREATE TABLE baggage (
    baggage_id SERIAL PRIMARY KEY,
    passenger_id INT REFERENCES passengers(passenger_id),
    weight NUMERIC(6,2)
);

CREATE TABLE baggage_checks (
    check_id SERIAL PRIMARY KEY,
    baggage_id INT REFERENCES baggage(baggage_id),
    created_at TIMESTAMP,
    updated_at TIMESTAMP
);
```

1. Create an index on the actual_departure column in the flights table.



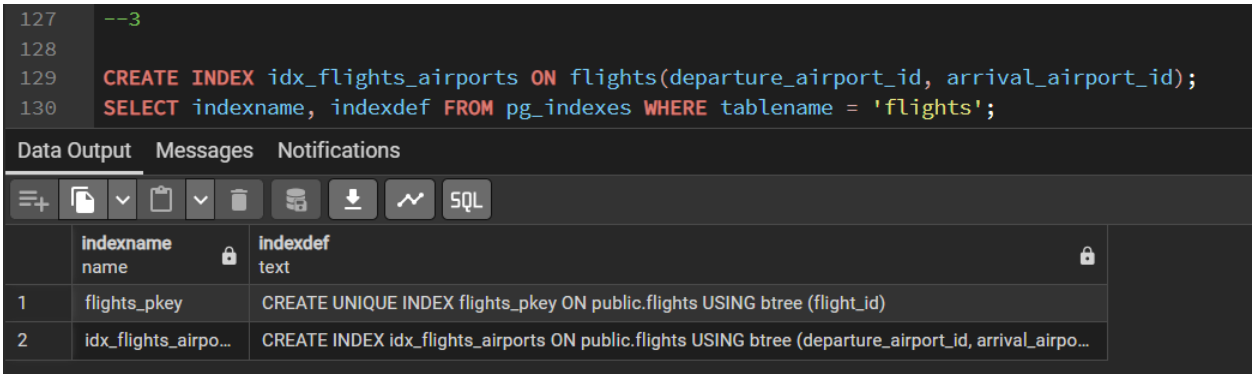
```sql
--1

CREATE INDEX idx_flights_actual_departure ON flights(actual_departure);
SELECT indexname, indexdef FROM pg_indexes WHERE tablename = 'flights';
```

Data Output | Messages | Notifications

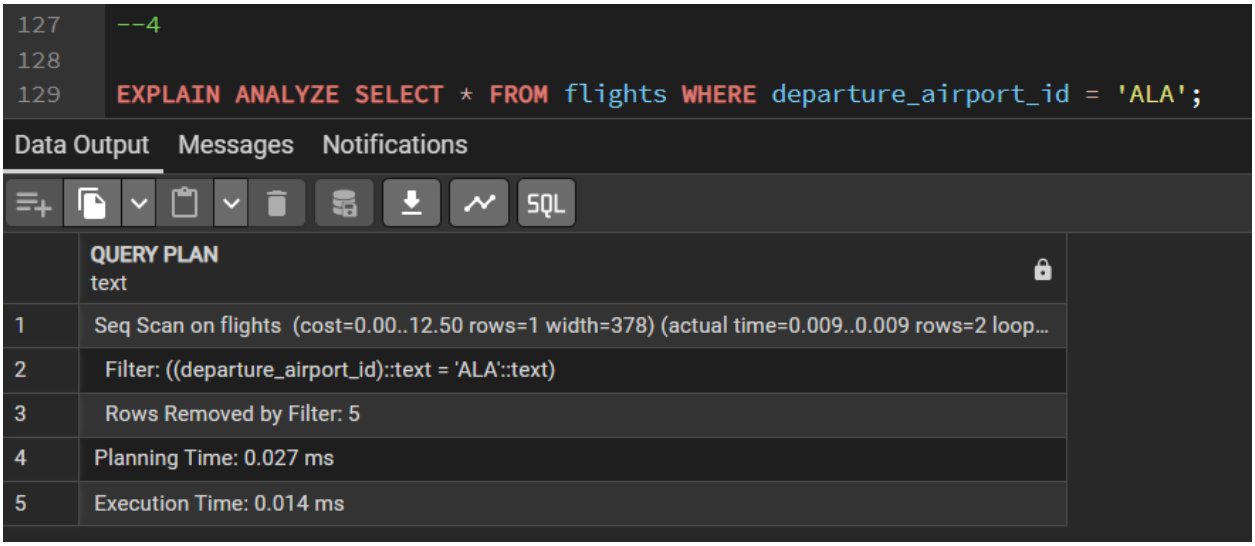
	indexname name	indexdef text
1	flights_pkey	CREATE UNIQUE INDEX flights_pkey ON public.flights USING btree (flight_id)
2	idx_flights_actual_depart...	CREATE INDEX idx_flights_actual_departure ON public.flights USING btree (actual_depar...

2. Create a unique index to ensure flight_no and scheduled_departure combinations are unique.

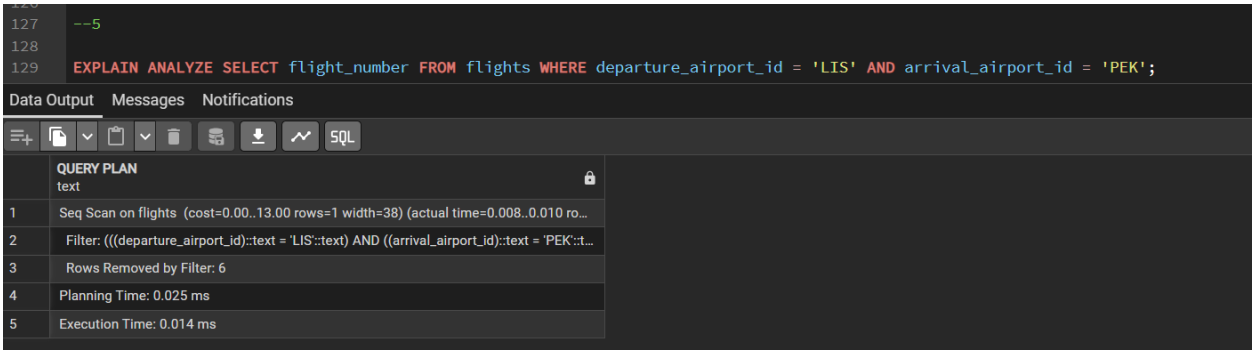
3. Create a composite index on the departure_airport_id and arrival_airport_id columns.



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.

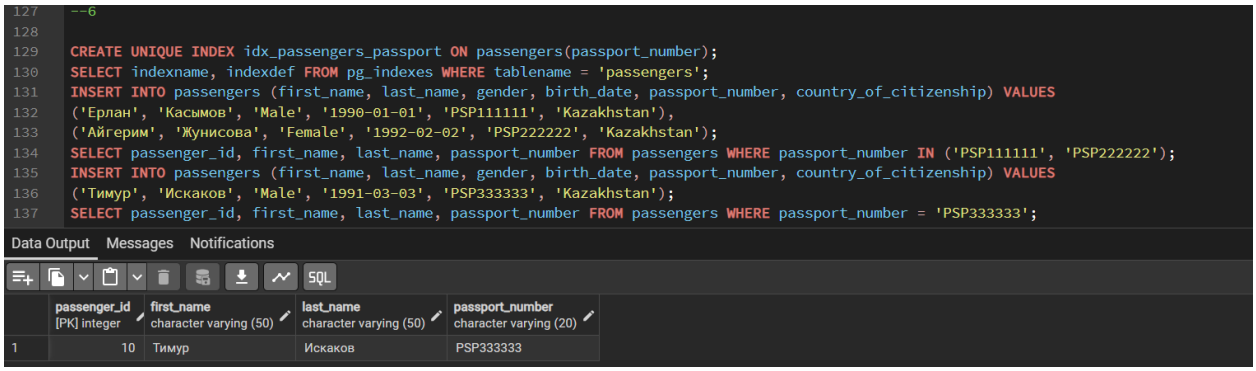


```
--5

EXPLAIN ANALYZE SELECT flight_number FROM flights WHERE departure_airport_id = 'LIS' AND arrival_airport_id = 'PEK';
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Seq Scan on flights (cost=0.00..13.00 rows=1 width=38) (actual time=0.008..0.010 ro...
2	Filter: (((departure_airport_id)::text = 'LIS'::text) AND ((arrival_airport_id)::text = 'PEK'::t...
3	Rows Removed by Filter: 6
4	Planning Time: 0.025 ms
5	Execution Time: 0.014 ms

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?

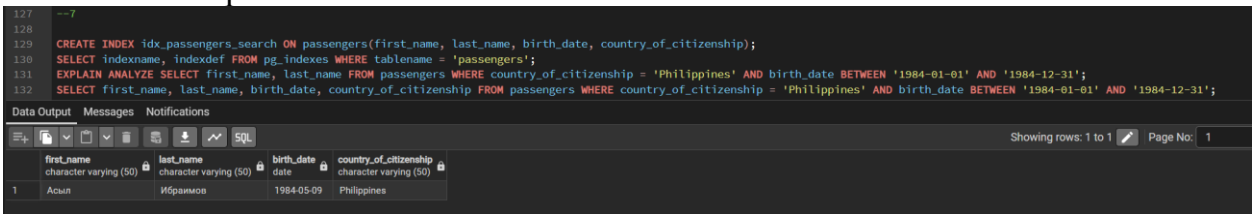


```
--6

CREATE UNIQUE INDEX idx_passengers_passport ON passengers(passport_number);
SELECT indexname, indexdef FROM pg_indexes WHERE tablename = 'passengers';
INSERT INTO passengers (first_name, last_name, gender, birth_date, passport_number, country_of_citizenship) VALUES
('Ерлан', 'Касымов', 'Male', '1990-01-01', 'PSP111111', 'Kazakhstan'),
('Айгерим', 'Жунисова', 'Female', '1992-02-02', 'PSP222222', 'Kazakhstan');
SELECT passenger_id, first_name, last_name, passport_number FROM passengers WHERE passport_number IN ('PSP111111', 'PSP222222');
INSERT INTO passengers (first_name, last_name, gender, birth_date, passport_number, country_of_citizenship) VALUES
('Тимур', 'Искаков', 'Male', '1991-03-03', 'PSP333333', 'Kazakhstan');
SELECT passenger_id, first_name, last_name, passport_number FROM passengers WHERE passport_number = 'PSP333333';
```

Data Output | Messages | Notifications

	passenger_id [PK] integer	first_name character varying (50)	last_name character varying (50)	passport_number character varying (20)
1	10	Тимур	Искаков	PSP333333

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.



```
--7

CREATE INDEX idx_passengers_search ON passengers(first_name, last_name, birth_date, country_of_citizenship);
SELECT indexname, indexdef FROM pg_indexes WHERE tablename = 'passengers';
EXPLAIN ANALYZE SELECT first_name, last_name FROM passengers WHERE country_of_citizenship = 'Philippines' AND birth_date BETWEEN '1984-01-01' AND '1984-12-31';
SELECT first_name, last_name, birth_date, country_of_citizenship FROM passengers WHERE country_of_citizenship = 'Philippines' AND birth_date BETWEEN '1984-01-01' AND '1984-12-31';
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1

	first_name character varying (50)	last_name character varying (50)	birth_date date	country_of_citizenship character varying (50)
1	Асыл	Ибраимов	1984-05-09	Philippines

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

```
--8

SELECT indexname, indexdef FROM pg_indexes WHERE tablename = 'passengers';
DROP INDEX IF EXISTS idx_passengers_passport, idx_passengers_search, idx_flights_actual_departure, idx_flights_unique, idx_flights_airports;
SELECT indexname FROM pg_indexes WHERE tablename IN ('passengers', 'flights');
```

Data Output | Messages | Notifications

	indexname name
1	passengers_pk...
2	flights_pkey