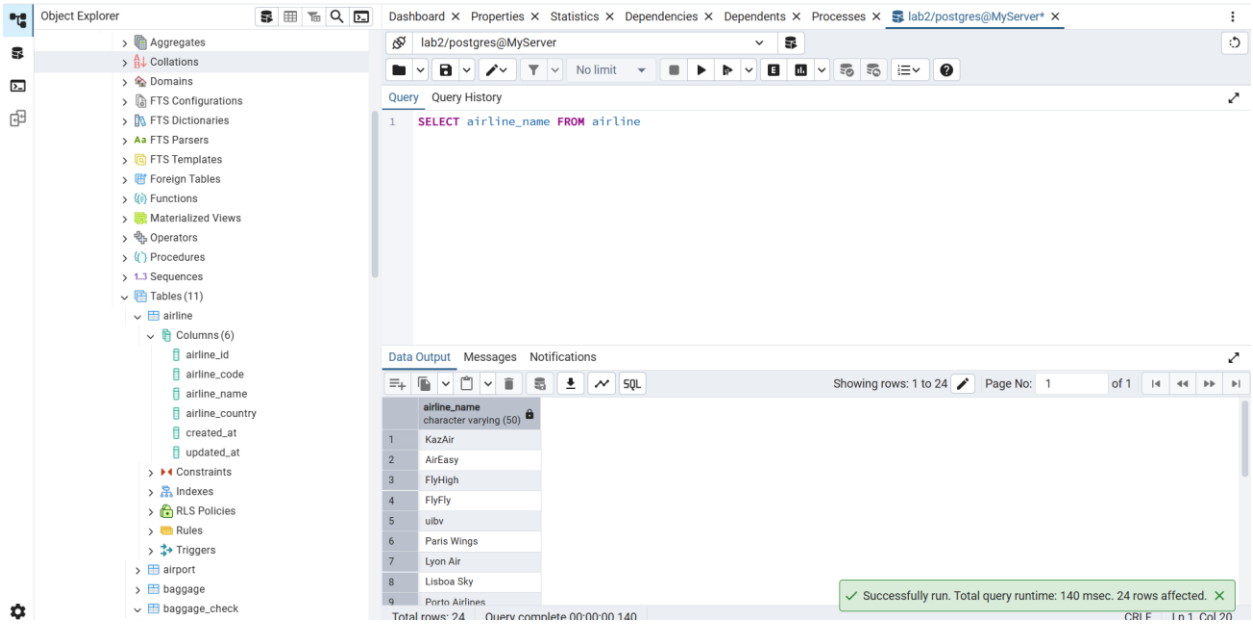
LAB 4 Abdykamat Adilet.

1) Retrieve all airline names in uppercase.

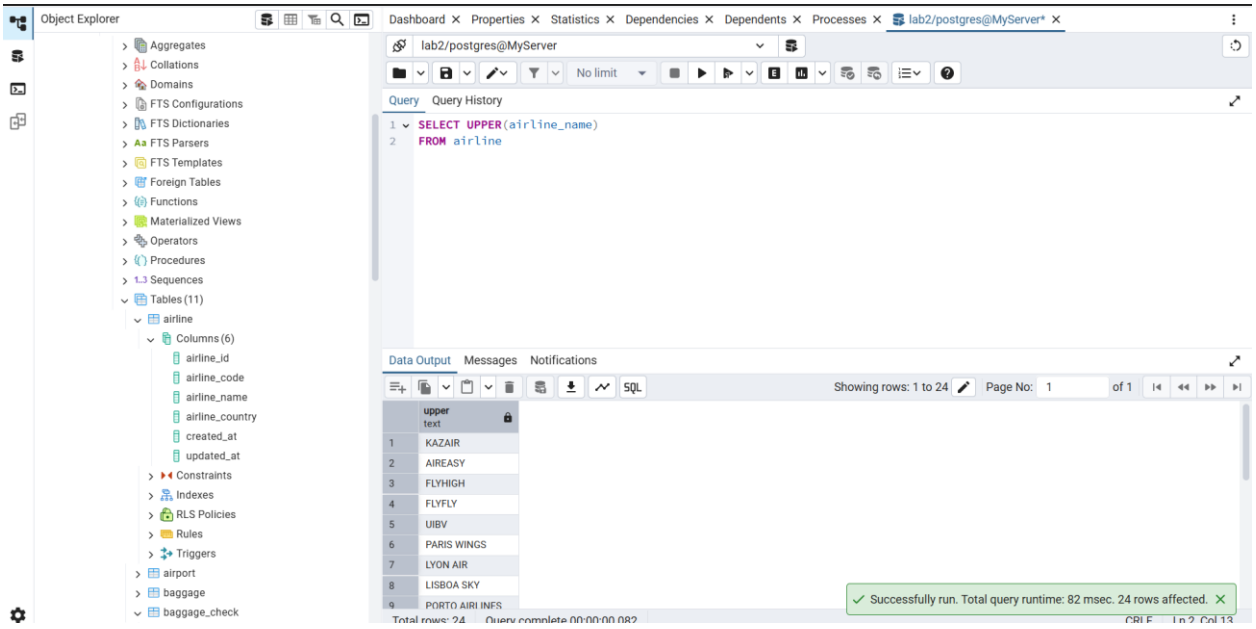
До:



После:

SELECT UPPER(airline_name)

FROM airline



2. Replace any occurrence of the word "Air" in airline names with "Aero".

SELECT REPLACE(airline_name, ‘Air’, ‘Aero’)

FROM airline

Query Query History

```
1 SELECT REPLACE(airline_name, 'Air', 'Aero')
2 FROM airline
```

Data Output Messages Notifications

Showing rows: 1 to 24 Page No: 1 of 1

	replace text
1	KazAero
2	AeroEasy
3	FlyHigh
4	FlyFly
5	uibv
6	Paris Wings
7	Lyon Aero
8	Lisboa Sky
9	Porto Aerolines

Successfully run. Total query runtime: 82 msec. 24 rows affected.

3. Fing all flight numbers that coordinates with both airline 1 and airline 2.

Query Query History

```
1 SELECT flight_id
2 FROM flights
3 WHERE airline_id IN (1, 2)
4 GROUP BY flight_id
5 HAVING COUNT(DISTINCT airline_id) = 2;
```

Data Output Messages Notifications

flight_id [PK] integer

4. Retrieve airports that contain the word "Reginal" and "Air" in their names.

SELECT airport_name FROM airport
WHERE airport_name LIKE '%Regional%' AND airport_name LIKE '%Air%'

Query | Query History

```
1  SELECT airport_name FROM airport
2  WHERE airport_name LIKE '%Regional%' AND airport_name LIKE '%Air%'
```

Data Output | Messages | Notifications

SQL | Showing rows: 1 to 3

	airport_name character varying (50)
1	Astana Regional Air Center
2	Shymkent Regional Airport
3	Taraz Regional Air Base
4	Pavlodar Regional Air Port
5	Kyzylorda Regional Air Station
6	Ekibastuz Regional Air Terminal
7	Temirtau Regional Air Hub
8	Baikonur Regional Air Field
9	Balkhash Regional Air Base

Total rows: 39 | Query complete 00:00:00.109

5. Retrieve passenger names and format their birth dates as 'Month DD, YYYY'..o

SELECT first_name,last_name,
TO_CHAR(date_of_birth, 'Month DD, YYYY') AS formatted_birth_date
FROM passengers;

Query | Query History

```
1  SELECT first_name,last_name,
2         TO_CHAR(date_of_birth, 'Month DD, YYYY') AS formatted_birth_date
3  FROM passengers;
```

Data Output | Messages | Notifications

SQL | Showing rows: 1 to 30

	first_name character varying (50)	last_name character varying (50)	formatted_birth_date text
1	Иван	Иван	May 15, 1990
2	Мария	Петрова	December 03, 1988
3	Алексей	Сидоров	August 22, 1979
4	Елена	Козлова	February 14, 1995
5	Дмитрий	Федоров	November 30, 1982
6	Ольга	Николаева	July 19, 1992
7	Сергей	Васильев	April 08, 1975
8	Анна	Смирнова	September 25, 1987
9	Михаил	Попов	March 17, 1991

6. Find flight numbers that have been delayed based on the actual arrival time.

SELECT arriving_airport_id FROM flights
WHERE sch_arrival_time < act_arrival_time;

Query Query History

```
1 SELECT arriving_airport_id FROM flights
2 WHERE sch_arrival_time < act_arrival_time;
```

Data Output Messages Notifications

SQL

	arriving_airport_id integer
1	5
2	6
3	7
4	8
5	9
6	10
7	1
8	2
9	3

7. Create a query that divides passengers into age groups like 'Young' and 'Adult' based on their birth date. Young passengers age between 18 and 35, Adult passengers age between 36 and 55.

```
SELECT first_name,last_name,
    DATE_PART('year',AGE(date_of_birth)) as Age,
    CASE
    WHEN DATE_PART('year',AGE(date_of_birth)) BETWEEN 18 and 35 THEN 'Young'
    WHEN DATE_PART('year',AGE(date_of_birth)) BETWEEN 36 and 55 THEN 'Adult'
    ELSE 'Other'
    END AS age_group
FROM passengers
```

Query | Query History

```sql
SELECT first_name,last_name,
    DATE_PART('year',AGE(date_of_birth)) as Age,
    CASE
        WHEN DATE_PART('year',AGE(date_of_birth)) BETWEEN 18 and 35 THEN 'Young'
        WHEN DATE_PART('year',AGE(date_of_birth)) BETWEEN 36 and 55 THEN 'Adult'
        ELSE 'Other'
    END AS age_group
FROM passengers
```

Data Output | Messages | Notifications

Showing rows: 1 to 30 | Page No

	first_name character varying (50)	last_name character varying (50)	age double precision	age_group text
1	Иван	Иван	35	Young
2	Мария	Петрова	36	Adult
3	Алексей	Сидоров	46	Adult
4	Елена	Козлова	30	Young
5	Дмитрий	Федоров	42	Adult
6	Ольга	Николаева	33	Young
7	Сергей	Васильев	50	Adult
8	Анна	Смирнова	38	Adult
9	Михаил	Попов	34	Young

Successfully run. Total query ru

Total rows: 30 | Query complete 00:00:00.201

8. Create a query that categorizes ticket prices based on their price as "Cheap," "Medium" or "Expensive."

```sql
SELECT ticket_id, price,
    CASE
    WHEN price < 10000 THEN 'Cheap'
    WHEN price BETWEEN 10000 and 25000 THEN 'Medium'
    ELSE 'Expensive'
     END AS price_category
FROM tickets;
```

Query Query History

```
1  SELECT ticket_id, price,
2      CASE
3          WHEN price < 10000 THEN 'Cheap'
4          WHEN price BETWEEN 10000 and 25000 THEN 'Medium'
5          ELSE 'Expensive'
6       END AS price_category
7  FROM tickets;
```

Data Output Messages Notifications

Showing rows: 1 to 30

	ticket_id [PK] integer	price numeric (10,2)	price_category text
1	1	15000.00	Medium
2	2	25000.00	Medium
3	3	18000.00	Medium
4	4	22000.00	Medium
5	5	30000.00	Expensive
6	6	12000.00	Medium
7	7	28000.00	Expensive
8	8	16000.00	Medium
9	9	19000.00	Medium

Successfully run. To

Total rows: 30 Query complete 00:00:00.215

9. Find number of airline names in each airline country.

SELECT airline_country,
COUNT(airline_name) as number_of_airlines
FROM airline
GROUP BY airline_country;

Query Query History

```
1 SELECT airline_country,
2 COUNT(airline_name) as number_of_airlines
3 FROM airline
4 GROUP BY airline_country;
```

Data Output Messages Notifications

Showing rows: 1 to

	airline_country character varying (50)	number_of_airlines bigint
1	Portugal	5
2	France	5
3	Turkey	1
4	China	1
5	UK	1
6	Germany	1
7	Kazakhstan	2
8	Japan	1
9	Brazil	1

Total rows: 10 Query complete 00:00:00.213

10. Find flights that arrived late according to their actual arrival time compared to the scheduled arrival time.

```
SELECT flight_id,sch_arrival_time,act_arrival_time,
   (act_arrival_time - sch_arrival_time) AS delay_duration
FROM flights
WHERE act_arrival_time > sch_arrival_time;
```

Query | Query History

```sql
SELECT flight_id,sch_arrival_time,act_arrival_time,
    (act_arrival_time - sch_arrival_time) AS delay_duration
FROM flights
WHERE act_arrival_time > sch_arrival_time;
```

Data Output | Messages | Notifications

Showing rows: 1 to 20 | Page No: 1

	flight_id [PK] integer	sch_arrival_time timestamp without time zone	act_arrival_time timestamp without time zone	delay_duration interval
1	1	2024-01-15 12:30:00	2024-01-15 12:35:00	00:05:00
2	2	2024-01-20 14:00:00	2024-01-20 14:10:00	00:10:00
3	3	2024-02-05 12:20:00	2024-02-05 12:25:00	00:05:00
4	4	2024-02-18 18:45:00	2024-02-18 18:55:00	00:10:00
5	5	2024-03-01 15:20:00	2024-03-01 15:30:00	00:10:00
6	6	2024-03-15 11:40:00	2024-03-15 11:45:00	00:05:00
7	7	2024-04-02 19:10:00	2024-04-02 19:20:00	00:10:00
8	8	2024-04-20 13:50:00	2024-04-20 13:55:00	00:05:00
9	9	2024-05-05 17:30:00	2024-05-05 17:35:00	00:05:

Successfully run. Total query runtime: 197

Total rows: 20 | Query complete 00:00:00.197