

Laboratory work №3

1. Select all the data of passengers whose last name is same as first name.

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, the 'passengers' table under the 'Tables' section is selected. The 'Columns' tab is open, showing columns: passenger_id, first_name, last_name, date_of_birth, gender, country_of_citizenship, country_of_residence, passport_number, created_at, and update_at. In the main query editor, the following SQL code is written:

```
1 SELECT * FROM passengers WHERE first_name = last_name
```

The Data Output tab shows the schema for the result:

passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship	country_of_residence
[PK] integer	character varying (50)	character varying (50)	date	character varying (50)	character varying (50)	character varying (50)

Total rows: 0 Query complete 00:00:00.102 CRLF Ln 1, Col 52

2. Select the last name of all passengers without duplicates.

The screenshot shows the pgAdmin 4 interface. The 'passengers' table is selected in the Object Explorer. The 'Columns' tab is open, showing the same set of columns as the previous screenshot. In the main query editor, the following SQL code is written:

```
1 SELECT DISTINCT last_name FROM passengers
```

The Data Output tab shows the results:

last_name
Hardman
Payne
Mixer
Gheorghe
Golsby
McGennis
Danihelka
Musla
Lynett
Roman
Rosling

Showing rows: 1 to 199 Page No: 1 of 1 Total rows: 199 Query complete 00:00:00.083 CRLF Ln 1, Col 31

3. Find all male passengers born between 1990 and 2000.

```
SELECT* FROM passengers WHERE gender= 'male' AND date_of_birth BETWEEN '1990-01-01' AND '2000-12-31';
```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with the 'Tables' section expanded, showing tables like 'airline', 'airport', 'baggage', etc., and the 'passenger' table with 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 window contains the SQL query: 'SELECT* FROM passengers WHERE gender= 'male' AND date_of_birth BETWEEN '1990-01-01' AND '2000-12-31';'. Below the query is a Data Output tab showing the results:

passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship	country_of_residence

Total rows: 0 Query complete 00:00:00.104 CRLF Ln 1, Col 101

4. Find price of tickets sold for each month in sorted way.

```
SELECT EXTRACT(YEAR FROM created_at) AS year, EXTRACT(MONTH FROM created_at) AS month, SUM(price) AS total_price FROM booking GROUP BY year, month ORDER BY year, month;
```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with the 'Tables' section expanded, showing tables like 'airline', 'airport', 'baggage', etc., and the 'booking' table with its columns: booking_id, passenger_id, booking_platform, created_at, update_at, status, and price. The main window contains the SQL query: 'SELECT EXTRACT(YEAR FROM created_at) AS year, EXTRACT(MONTH FROM created_at) AS month, SUM(price) AS total_price FROM booking GROUP BY year, month ORDER BY year, month;'. Below the query is a Data Output tab showing the results:

year	month	total_price
2023	3	155294.81
2023	4	240649.84
2023	5	207361.58
2023	6	177058.76
2023	7	158703.58
2023	8	213237.05

Total rows: 13 Query complete 00:00:00.073 CRLF Ln 4, Col 3

5. Create a query that shows all flights flying to 'China'.

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with tables like `flights`, `airport`, and `country`.
- Query Editor:** Contains the following SQL query:

```
1 SELECT f.* FROM flights f
2 JOIN airport a ON f.arrival_airport_id = a.airport_id
3 WHERE a.country = 'China'
```
- Data Output:** Displays the results of the query, showing 238 rows of flight information, including columns like `flight_id`, `flight_no`, `scheduled_departure`, and `arriving_airport_id`.
- System Bar:** Shows the date as 30.09.2025 and the time as 22:44.

6. Show airlines from any of: ('France','Portugal','Poland') created between '2023-11-01' and '2024-03-31'.

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with tables like `airline`.
- Query Editor:** Contains the following SQL query:

```
1 SELECT * FROM airline
2 WHERE airline_country IN ('France', 'Portugal', 'Poland') AND created_at
3 BETWEEN '2023-11-01' AND '2024-03-31';
```
- Data Output:** Displays the results of the query, showing 7 rows of airline information, including columns like `airline_id`, `airline_code`, and `airline_name`.
- System Bar:** Shows the date as 30.09.2025 and the time as 22:44.

7. Find all airline names based in Kazakhstan.

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'Tables' section, the 'airline' table is selected. In the main pane, a query is run:

```
1 SELECT* FROM airline
2 WHERE airline_country = 'Kazakhstan';
3
```

The results show one row:

airline_id	airline_code	airline_name	airline_country	created_at	update_at
5	CYLQ	YLQ	Kazakhstan	2023-03-23	2023-07-26

Total rows: 1 Query complete 00:00:00.073

8. Reduce the cost of booking price by 10% created before '11-01-2023'.

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'booking' table, the 'Columns' section is expanded. A query is run:

```
1 UPDATE booking SET price = price*0.9 WHERE created_at < '2023-01-11';
```

The results show the message:

UPDATE 0
Query returned successfully in 57 msec.

Total rows: 0 Query complete 00:00:00.057

9. Find top3 overweighted baggage with more than 25kg.

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, a database named 'Lab_3' is selected, and a table named 'baggage' is expanded to show its columns: baggage_id, weight_in_kg, created_date, update_date, and booking_id. A SQL query is run in the main pane:

```
1 SELECT * FROM baggage WHERE weight_in_kg > 25 ORDER BY weight_in_kg DESC LIMIT 3;
```

The Data Output pane displays the results:

baggage_id	weight_in_kg	created_date	update_date	booking_id
321	49.80	2023-07-07	2023-11-10	349
61	49.57	2023-04-21	2023-07-31	464
102	49.44	2024-01-24	2023-11-01	53

Total rows: 3 Query complete 00:00:00.084

10. Find the youngest passengers' full name.

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, a database named 'Lab_3' is selected, and a table named 'passengers' is expanded to show 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. A SQL query is run in the main pane:

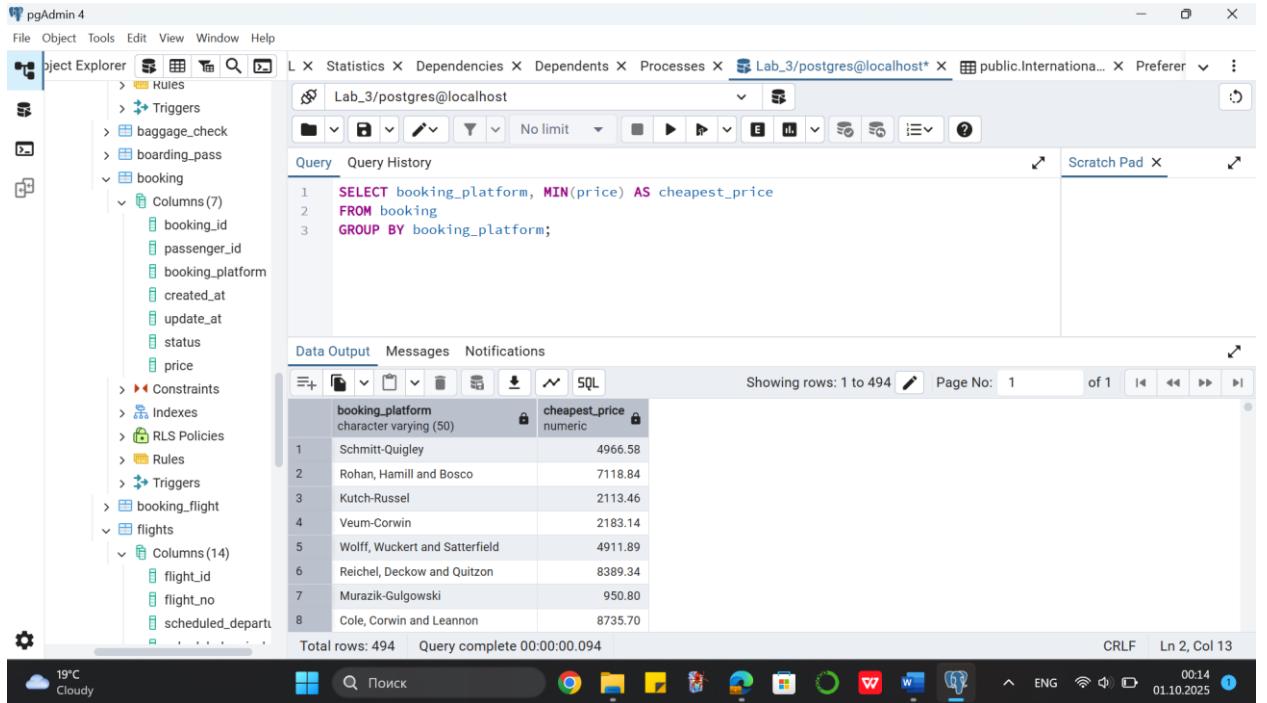
```
1 SELECT first_name || ' ' || last_name AS full_name FROM passengers
2 ORDER BY date_of_birth DESC
3 LIMIT 1;
```

The Data Output pane displays the results:

full_name
Bernie Michal

Total rows: 1 Query complete 00:00:00.084

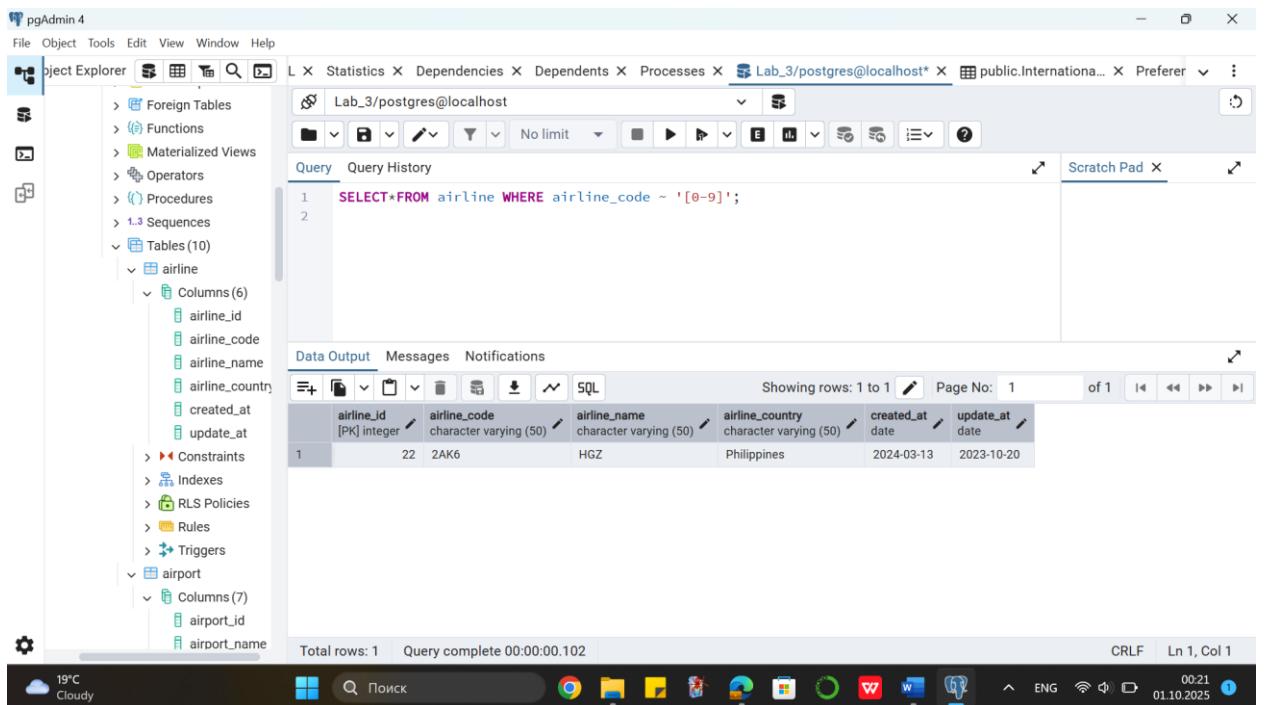
11. Find the cheapest booking price on each booking platform.



```
SELECT booking_platform, MIN(price) AS cheapest_price
FROM booking
GROUP BY booking_platform;
```

booking_platform	cheapest_price
Schmitt-Quigley	4966.58
Rohan, Hamill and Bosco	7118.84
Kutch-Russel	2113.46
Veum-Corwin	2183.14
Wolff, Wuckert and Satterfield	4911.89
Reichel, Deckow and Quitzon	8389.34
Murazik-Gulgowski	950.80
Cole, Corwin and Leannon	8735.70

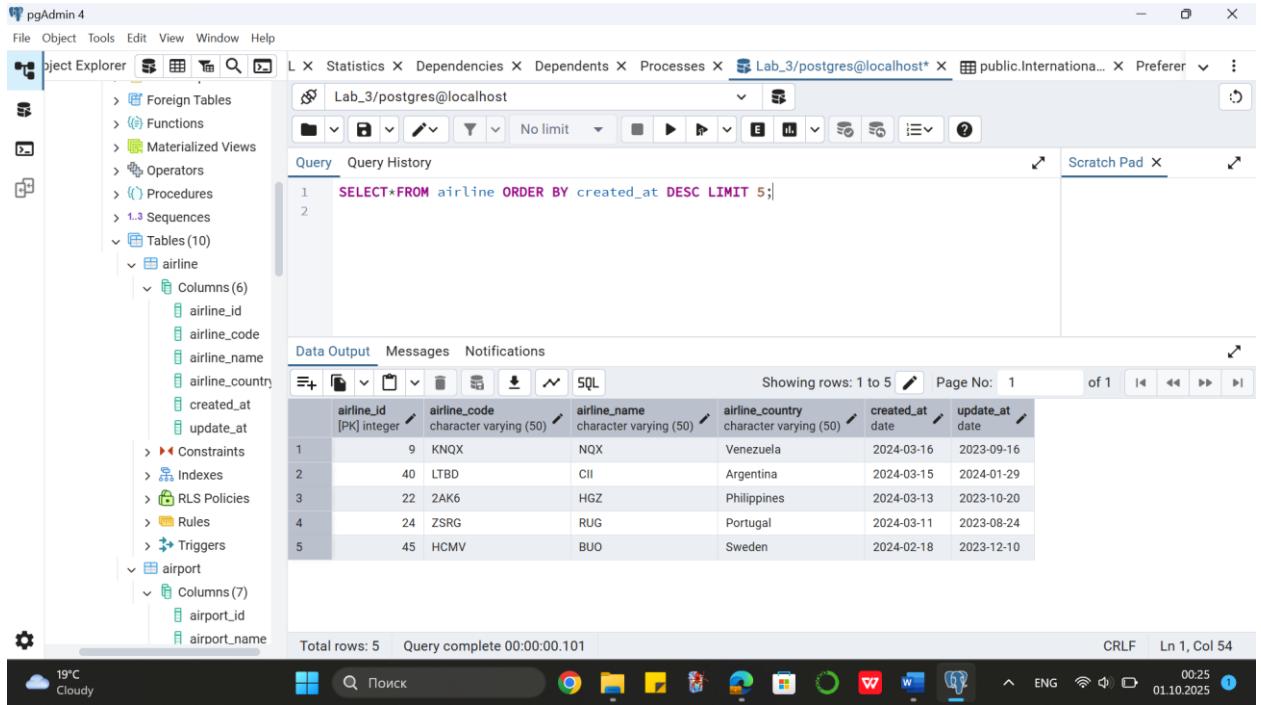
12. Return airlines whose airline_code contains a digit.



```
SELECT * FROM airline WHERE airline_code ~ '[0-9]';
```

airline_id	airline_code	airline_name	airline_country	created_at	update_at
22	2AK6	HGZ	Philippines	2024-03-13	2023-10-20

13. List the top5 most recently created airlines.



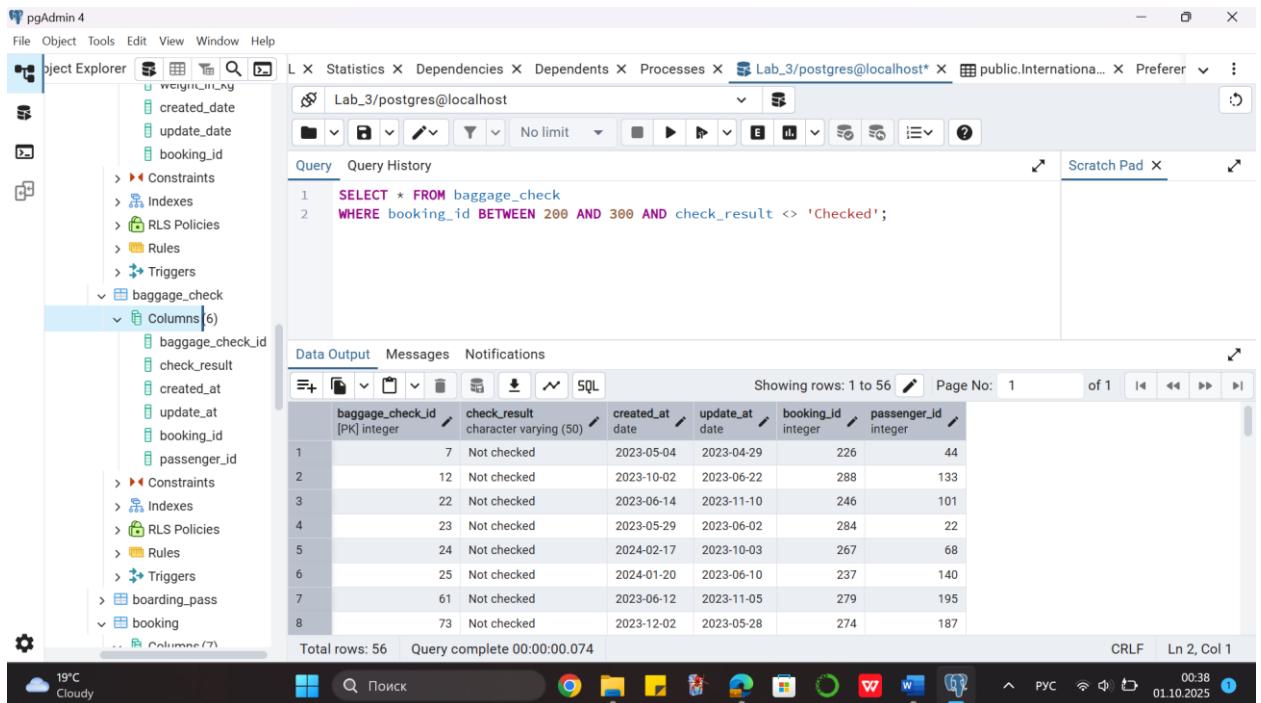
The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with the **airline** table selected. The **airline** table has 6 columns: **airline_id**, **airline_code**, **airline_name**, **airline_country**, **created_at**, and **update_at**.
- Query Editor:** Contains the SQL query: `SELECT * FROM airline ORDER BY created_at DESC LIMIT 5;`
- Data Output:** Displays the results of the query, showing 5 rows of airline data. The columns are **airline_id**, **airline_code**, **airline_name**, **airline_country**, **created_at**, and **update_at**. The data is as follows:

	airline_id	airline_code	airline_name	airline_country	created_at	update_at
1	9	KNQX	NQX	Venezuela	2024-03-16	2023-09-16
2	40	LTBD	CII	Argentina	2024-03-15	2024-01-29
3	22	2AK6	HGZ	Philippines	2024-03-13	2023-10-20
4	24	ZSRG	RUG	Portugal	2024-03-11	2023-08-24
5	45	HCMV	BUO	Sweden	2024-02-18	2023-12-10

Total rows: 5 Query complete 00:00:00.101 CRLF Ln 1, Col 54

14. Return all rows where booking_id is between 200 and 300 inclusive and check_result <> 'Checked'.



The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with the **baggage_check** table selected. The **baggage_check** table has 6 columns: **baggage_check_id**, **check_result**, **created_at**, **update_at**, **booking_id**, and **passenger_id**.
- Query Editor:** Contains the SQL query: `SELECT * FROM baggage_check WHERE booking_id BETWEEN 200 AND 300 AND check_result <> 'Checked';`
- Data Output:** Displays the results of the query, showing 8 rows of baggage check data. The columns are **baggage_check_id**, **check_result**, **created_at**, **update_at**, **booking_id**, and **passenger_id**. The data is as follows:

	baggage_check_id	check_result	created_at	update_at	booking_id	passenger_id
1	7	Not checked	2023-05-04	2023-04-29	226	44
2	12	Not checked	2023-10-02	2023-06-22	288	133
3	22	Not checked	2023-06-14	2023-11-10	246	101
4	23	Not checked	2023-05-29	2023-06-02	284	22
5	24	Not checked	2024-02-17	2023-10-03	267	68
6	25	Not checked	2024-01-20	2023-06-10	237	140
7	61	Not checked	2023-06-12	2023-11-05	279	195
8	73	Not checked	2023-12-02	2023-05-28	274	187

Total rows: 56 Query complete 00:00:00.074 CRLF Ln 2, Col 1

15. Baggage checks where update_at is in the same month as created_at but occurs earlier than created_at.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer pane, which lists various database objects like 'baggage_check' with its columns: baggage_check_id, check_result, created_at, update_at, booking_id, and passenger_id. The 'SQL' tab in the center contains the following SQL code:

```
1 SELECT * FROM baggage_check
2 WHERE EXTRACT(YEAR FROM created_at) = EXTRACT(YEAR FROM update_at)
3 AND EXTRACT(MONTH FROM created_at) = EXTRACT(MONTH FROM update_at)
4 AND update_at < created_at;
```

The results pane below shows the output of the query, displaying 18 rows of data:

baggage_check_id	check_result	created_at	update_at	booking_id	passenger_id	
1	34	Checked	2023-04-25	2023-04-08	28	57
2	86	Checked	2023-12-25	2023-12-07	312	22
3	104	Checked	2024-03-17	2024-03-14	41	83
4	109	Not checked	2023-08-30	2023-08-09	469	91
5	119	Not checked	2023-12-02	2023-12-01	135	55
6	148	Checked	2024-03-17	2024-03-16	86	5
7	219	Checked	2023-04-17	2023-04-14	233	200
8	248	Not checked	2023-06-22	2023-06-09	449	60
Total rows: 18	Query complete 00:00:00.056	CRLF	Ln 4, Col 28			