

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. On the left, the 'Object Explorer' pane displays the database structure, including tables like 'airline', 'airport', 'baggage', 'boarding_pass', 'booking', 'booking_flight', 'flights', and 'passengers'. The 'passengers' table is selected, showing 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 query editor displays the following SQL query:

```
1 SELECT * FROM passengers WHERE first_name = last_name
2
```

The 'Data Output' pane at the bottom shows the query results with columns: 'passenger_id [PK] integer', 'first_name character varying (50)', 'last_name character varying (50)', 'date_of_birth date', 'gender character varying (50)', 'country_of_citizenship character varying (50)', and 'country_of_residence character varying (50)'. The status bar indicates 'Total rows: 0' and 'Query complete 00:00:00.102'.

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

The screenshot shows the pgAdmin 4 interface. The 'Object Explorer' pane shows the 'passengers' table selected. The main query editor displays the following SQL query:

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

The 'Data Output' pane shows the query results with the column 'last_name character varying (50)'. The results are displayed as a list of 11 distinct last names:

last_name
1 Hardman
2 Payne
3 Mixter
4 Gheorghe
5 Golsby
6 McGennis
7 Danihelka
8 Musla
9 Lynett
10 Romain
11 Rosling

The status bar indicates 'Total rows: 199' and 'Query complete 00:00:00.083'.

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

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'passengers' table selected under the 'Tables' folder. The main query editor contains the following SQL query:

```
1 SELECT* FROM passengers WHERE gender= 'male' AND date_of_birth BETWEEN '1990-01-01' A
```

The 'Data Output' tab at the bottom shows the schema of the 'passengers' table:

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)

The status bar at the bottom indicates 'Total rows: 0' and 'Query complete 00:00:00.104'.

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

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'booking' table selected under the 'Tables' folder. The main query editor contains the following SQL query:

```
1 SELECT
2     EXTRACT(YEAR FROM created_at) AS year,
3     EXTRACT(MONTH FROM created_at) AS month,
4     SUM(price) AS total_price
5 FROM booking
6 GROUP BY year, month
7 ORDER BY year, month;
```

The 'Data Output' tab at the bottom shows the results of the query:

	year	month	total_price
	numeric	numeric	numeric
1	2023	3	155294.81
2	2023	4	240649.84
3	2023	5	207361.58
4	2023	6	177058.76
5	2023	7	158703.58
6	2023	8	213237.05

The status bar at the bottom indicates 'Total rows: 13' and 'Query complete 00:00:00.073'.

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

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database structure, with 'airport' selected under 'Tables'. The 'Query' pane 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'
```

The 'Data Output' pane shows the results of the query, displaying 7 rows of flight data. The columns are: flight_id (PK) integer, flight_no character varying (50), scheduled_departure date, scheduled_arrival date, departure_airport_id integer, arrival_airport_id integer, and departing_gate character varying (50).

flight_id (PK) integer	flight_no character varying (50)	scheduled_departure date	scheduled_arrival date	departure_airport_id integer	arrival_airport_id integer	departing_gate character varying (50)
1	3 FI-OL	2023-03-29	2023-08-01	18	12	47
2	9 IN-OR	2023-05-18	2023-09-19	6	9	659
3	13 BR-PE	2024-01-16	2023-06-02	13	20	1891
4	14 US-IN	2023-07-24	2024-02-27	18	3	973
5	23 CN-65	2023-03-21	2024-01-08	2	3	934
6	24 BR-TO	2024-03-09	2023-11-22	3	9	561
7	25 MZ-P	2024-01-22	2023-09-05	15	12	27

Total rows: 238 Query complete 00:00:00.117

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. On the left, the 'Object Explorer' pane displays the database structure, with 'airline' selected under 'Tables'. The 'Query' pane 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';
4
```

The 'Data Output' pane shows the results of the query, displaying 7 rows of airline data. The columns are: airline_id (PK) integer, airline_code character varying (50), airline_name character varying (50), airline_country character varying (50), created_at date, and update_at date.

airline_id (PK) integer	airline_code character varying (50)	airline_name character varying (50)	airline_country character varying (50)	created_at date	update_at date
1	4 OOKB	KHS	France	2023-11-18	2024-02-18
2	13 OIFE	IFH	Portugal	2024-01-09	2023-11-09
3	15 SYKZ	KKG	Poland	2023-12-24	2023-10-30
4	24 ZSRG	RUG	Portugal	2024-03-11	2023-08-24
5	28 RKPD	JDG	Poland	2023-11-03	2023-12-06
6	30 CYST	YST	Poland	2023-11-24	2023-10-07
7	43 AYKM	KMA	France	2023-11-21	2023-05-30

Total rows: 7 Query complete 00:00:00.069

7. Find all airline names based in Kazakhstan.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'airline' table selected under 'Tables (10)'. The main pane shows a SQL query:

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

The 'Data Output' tab shows the results of the query:

airline_id	airline_code	airline_name	airline_country	created_at	update_at
1	5	CYLQ	YLQ	Kazakhstan	2023-03-23

The status bar at the bottom indicates 'Total rows: 1' and '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. The left sidebar displays the database structure, with the 'booking' table selected under 'Tables (14)'. The main pane shows an SQL update query:

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

The 'Messages' tab shows the execution status:

```
UPDATE 0
Query returned successfully in 57 msec.
```

The status bar at the bottom indicates 'Total rows: 0' and 'Query complete 00:00:00.057'.

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

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'baggage' table selected under the 'Columns' tab. The main query editor contains the following SQL query:

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

The 'Data Output' tab shows the results of the query, displaying 3 rows. The columns are: baggage_id [PK] integer, weight_in_kg numeric (4,2), created_date date, update_date date, and booking_id integer.

baggage_id [PK] integer	weight_in_kg numeric (4,2)	created_date date	update_date date	booking_id integer	
1	321	49.80	2023-07-07	2023-11-10	349
2	61	49.57	2023-04-21	2023-07-31	464
3	102	49.44	2024-01-24	2023-11-01	53

The status bar at the bottom indicates 'Total rows: 3' and 'Query complete 00:00:00.084'.

10. Find the youngest passengers' full name.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'passengers' table selected under the 'Columns' tab. The main query editor contains the following SQL query:

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

The 'Data Output' tab shows the results of the query, displaying 1 row. The column is: full_name text.

full_name text
Bernie Michal

The status bar at the bottom indicates 'Total rows: 1' and 'Query complete 00:00:00.084'.

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

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'booking' table. The main query editor contains the following SQL query:

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

The 'Data Output' tab shows the results of the query, displaying two columns: 'booking_platform' and 'cheapest_price'. The results are as follows:

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

The status bar at the bottom indicates 'Total rows: 494' and 'Query complete 00:00:00.094'.

12. Return airlines whose airline_code contains a digit.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'airline' table. The main query editor contains the following SQL query:

```
1 SELECT * FROM airline WHERE airline_code ~ '[0-9]';
2
```

The 'Data Output' tab shows the results of the query, displaying columns: 'airline_id', 'airline_code', 'airline_name', 'airline_country', 'created_at', and 'update_at'. The results are as follows:

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

The status bar at the bottom indicates 'Total rows: 1' and 'Query complete 00:00:00.102'.

13. List the top5 most recently created airlines.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'airline' table. The main query editor contains the following SQL query:

```
1 SELECT * FROM airline ORDER BY created_at DESC LIMIT 5;
```

The 'Data Output' tab shows the results of the query, displaying 5 rows. The status bar indicates 'Total rows: 5' and 'Query complete 00:00:00.101'.

airline_id [PK] integer	airline_code character varying (50)	airline_name character varying (50)	airline_country character varying (50)	created_at date	update_at date
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

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

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'baggage_check' table. The main query editor contains the following SQL query:

```
1 SELECT * FROM baggage_check
2 WHERE booking_id BETWEEN 200 AND 300 AND check_result <> 'Checked';
```

The 'Data Output' tab shows the results of the query, displaying 8 rows. The status bar indicates 'Total rows: 56' and 'Query complete 00:00:00.074'.

baggage_check_id [PK] integer	check_result character varying (50)	created_at date	update_at date	booking_id integer	passenger_id integer
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

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

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- weight_kg
- created_date
- update_date
- booking_id
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- baggage_check
 - Columns (6)
 - baggage_check_id
 - check_result
 - created_at
 - update_at
 - booking_id
 - passenger_id
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- boarding_pass
- booking
 - Columns (7)

Lab_3/postgres@localhost

Query

```
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;
```

Data Output

	baggage_check_id [PK] integer	check_result character varying (50)	created_at date	update_at date	booking_id integer	passenger_id integer
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

19°C Cloudy

Поиск

00:44 01.10.2025