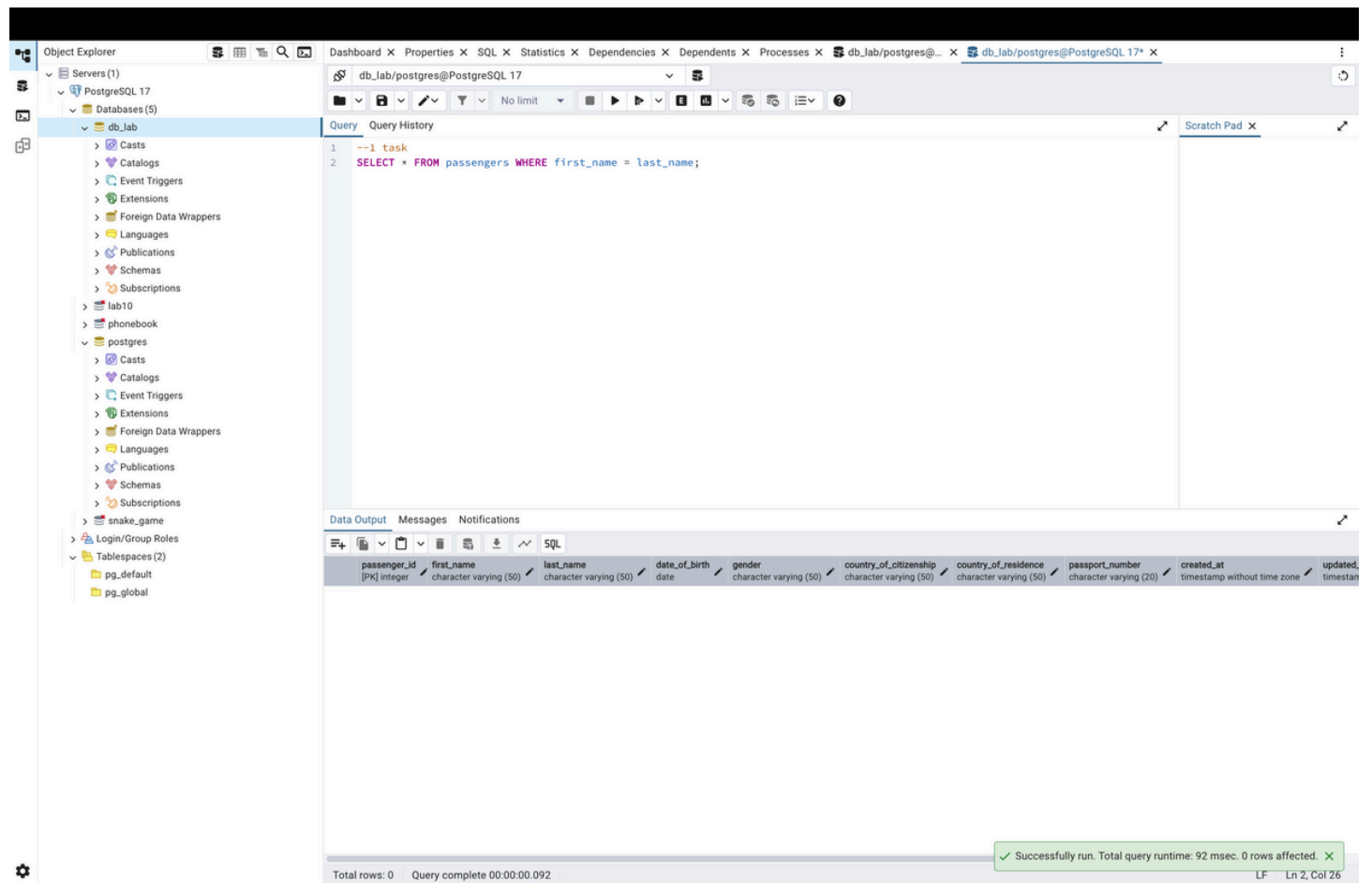


## Laboratory work 3

### Tasks:

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



The screenshot shows a PostgreSQL IDE interface. On the left, the 'Object Explorer' pane displays a tree view of the database structure, including 'Servers (1)', 'PostgreSQL 17', 'Databases (5)', and 'db\_lab'. The 'db\_lab' database is expanded, showing various objects like 'Casts', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas', 'Subscriptions', 'lab10', 'phonebook', 'postgres', 'snake\_game', 'Login/Group Roles', and 'Tablespaces (2)'. The 'Query' pane in the center contains the following SQL query:

```
--1 task
SELECT * FROM passengers WHERE first_name = last_name;
```

Below the query pane, the 'Data Output' pane shows the schema of the 'passengers' table:

passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship	country_of_residence	passport_number	created_at	updated_at
[PK] integer	character varying (50)	character varying (50)	date	character varying (50)	character varying (50)	character varying (50)	character varying (20)	timestamp without time zone	timestamp

At the bottom of the IDE, a status bar indicates 'Total rows: 0' and 'Query complete 00:00:00.092'. A green notification box at the bottom right states: '✓ Successfully run. Total query runtime: 92 msec. 0 rows affected. ✕'.

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

The screenshot shows the PostgreSQL IDE interface. The Object Explorer on the left displays the database structure, including the 'db\_lab' database and its tables. The main query editor contains the following SQL query:

```
--2 task
SELECT DISTINCT last_name FROM passengers;
```

The Data Output pane at the bottom shows the results of the query, displaying 15 rows of distinct last names. A status message at the bottom right indicates: "Successfully run. Total query runtime: 98 msec. 200 rows affected."

last_name
Chippindall
Sydney
Nabbs
Scutter
Runge
Berthon
Carass
Kittelman
Buttwell
Zottoli
Cassy
Sayton
Maffia
Panyer
Trayhorn

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

The screenshot shows the PostgreSQL IDE interface. The main query editor contains the following SQL query:

```
--3 task
SELECT * FROM passengers WHERE gender = 'Male' AND date_of_birth BETWEEN '1990-01-01' AND '2000-12-31';
```

The Data Output pane at the bottom shows the results of the query, displaying 0 rows. A status message at the bottom right indicates: "Successfully run. Total query runtime: 94 msec. 0 rows affected."

passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship	country_of_residence	passport_number	created_at	updated_at
--------------	------------	-----------	---------------	--------	------------------------	----------------------	-----------------	------------	------------

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

The screenshot shows the DBeaver interface with a PostgreSQL database named 'db\_lab/postgres@PostgreSQL 17'. The 'Object Explorer' on the left shows the database structure. The 'Query' editor in the center contains the following SQL query:

```
--4 task
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 'Data Output' tab at the bottom shows the results of the query, which are currently empty. A status bar at the bottom indicates 'Total rows: 0' and 'Query complete 00:00:00.084'. A green message box at the bottom right states 'Successfully run. Total query runtime: 84 msec. 0 rows affected.'

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

The screenshot shows the DBeaver interface with a PostgreSQL database named 'db\_lab/postgres@PostgreSQL 17'. The 'Query' editor in the center contains the following SQL query:

```
--5 task
SELECT * FROM flights WHERE arriving_airport_id IN (SELECT airport_id FROM Airport WHERE country = 'China');
```

The 'Data Output' tab at the bottom shows the results of the query, which are displayed in a table with 14 columns: flight\_id, sch\_departure\_time, sch\_arrival\_time, departing\_airport\_id, arriving\_airport\_id, departing\_gate, arriving\_gate, airline\_id, and act\_departure\_time. The table shows 21 rows of flight data. A status bar at the bottom indicates 'Total rows: 21' and 'Query complete 00:00:00.154'. A green message box at the bottom right states 'Successfully run. Total query runtime: 145 msec. 21 rows affected.'

flight_id	sch_departure_time	sch_arrival_time	departing_airport_id	arriving_airport_id	departing_gate	arriving_gate	airline_id	act_departure_time
1	6 2025-03-20 00:00:00	2025-05-20 00:00:00	6	6	NH4005	NH	6	2025-03-12 00:00:00
2	27 2025-04-30 00:00:00	2025-05-17 00:00:00	27	27	AV3764	AV	27	2024-09-27 00:00:00
3	29 2024-10-04 00:00:00	2025-03-08 00:00:00	29	29	ET3606	ET	29	2024-11-14 00:00:00
4	35 2025-01-23 00:00:00	2025-03-07 00:00:00	35	35	AM3012	AM	35	2024-12-07 00:00:00
5	39 2025-07-18 00:00:00	2025-05-03 00:00:00	39	39	AI9592	AI	39	2025-04-28 00:00:00
6	51 2025-09-09 00:00:00	2025-04-17 00:00:00	51	51	NZ1348	NZ	51	2024-12-04 00:00:00
7	55 2024-11-30 00:00:00	2025-09-05 00:00:00	55	55	EK9900	EK	55	2024-10-12 00:00:00
8	74 2025-03-09 00:00:00	2025-08-05 00:00:00	74	74	AC8416	AC	74	2024-10-20 00:00:00
9	89 2024-10-29 00:00:00	2025-03-20 00:00:00	89	89	AI7280	AI	89	2024-10-21 00:00:00
10	111 2025-01-05 00:00:00	2025-09-18 00:00:00	111	111	AM9465	AM	111	2024-12-06 00:00:00
11	117 2024-12-05 00:00:00	2025-04-10 00:00:00	117	117	QR6944	QR	117	2024-11-01 00:00:00
12	129 2025-09-21 00:00:00	2025-05-04 00:00:00	129	129	AI4444	AI	129	2024-12-26 00:00:00
13	135 2024-12-04 00:00:00	2025-01-31 00:00:00	135	135	NH6679	NH	135	2024-10-25 00:00:00
14	154 2024-12-20 00:00:00	2025-03-20 00:00:00	154	154	LA8052	LA	154	2025-09-08 00:00:00

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

The screenshot shows the PostgreSQL IDE interface. On the left, the Object Explorer displays the database structure, including the 'db\_lab' database. The main query editor contains the following SQL query:

```
--6 task
SELECT * FROM airline_info WHERE airline_country IN ('France','Portugal','Poland')
AND created_at BETWEEN '2023-11-01' AND '2024-03-31';
```

The Data Output pane at the bottom shows the schema of the 'airline\_info' table:

airline_id	airline_code	airline_name	airline_country	created_at	updated_at	info
[PK] integer	character varying (30)	character varying (300)	character varying (50)	timestamp without time zone	timestamp without time zone	character varying (50)

The status bar at the bottom indicates: "Total rows: 0 Query complete 00:00:00.110". A green message box at the bottom right states: "Successfully run. Total query runtime: 110 msec. 0 rows affected."

7. Find all airline names based in Kazakhstan.

The screenshot shows the PostgreSQL IDE interface. The query editor contains the following SQL query:

```
--7 task
SELECT airline_name FROM airline_info WHERE airline_country = 'Kazakhstan';
```

The Data Output pane at the bottom shows the schema of the 'airline\_info' table:

airline_name
character varying (300)

The status bar at the bottom indicates: "Total rows: 0 Query complete 00:00:00.074". A green message box at the bottom right states: "Successfully run. Total query runtime: 74 msec. 0 rows affected."

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

Object Explorer: Servers (1) > PostgreSQL 17 > Databases (5) > db\_lab

Query:

```
--8 task
UPDATE booking SET price = price * 0.9 WHERE created_at < '2023-11-01';
```

Data Output: Messages

UPDATE 0

Query returned successfully in 61 msec.

Total rows: Query complete 00:00:00.061

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

Object Explorer: Servers (1) > PostgreSQL 17 > Databases (5) > db\_lab

Query:

```
--9 task
SELECT * FROM baggage WHERE weight_in_kg > 25 ORDER BY weight_in_kg DESC LIMIT 3;
```

Data Output: Messages

baggage_id	weight_in_kg	created_at	updated_at	booking_id
[PK] integer	numeric (4,2)	timestamp without time zone	timestamp without time zone	integer

Successfully run. Total query runtime: 87 msec. 0 rows affected.

Total rows: 0 Query complete 00:00:00.087

10. Find the youngest passengers' full name.

Object Explorer: Servers (1) > PostgreSQL 17 > Databases (5) > db\_lab

Query: `--10 task`  
`2 SELECT first_name, last_name FROM passengers ORDER BY date_of_birth DESC LIMIT 1;`

Data Output:

	first_name character varying (50)	last_name character varying (50)
1	Panchito	Callam

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

Total rows: 1 Query complete 00:00:00.149

Successfully run. Total query runtime: 149 msec. 1 rows affected.

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

Object Explorer: Servers (1) > PostgreSQL 17 > Databases (5) > db\_lab

Query: `--11 task`  
`2 SELECT booking_platform, MIN(price) AS cheapest_price FROM booking GROUP BY booking_platform;`

Data Output:

	booking_platform character varying (50)	cheapest_price numeric
--	--	---------------------------

Showing rows: 0 to 0 Page No: 1 of 1

Total rows: 0 Query complete 00:00:00.147

Successfully run. Total query runtime: 147 msec. 0 rows affected.

12. Return airlines whose airline\_code contains a digit.

Object Explorer | Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Processes | db\_lab/postgres@... | db\_lab/postgres@PostgreSQL 17\*

db\_lab/postgres@PostgreSQL 17

Query Query History

```
--12 task
2 SELECT * FROM airline_info WHERE airline_code ~ '[0-9]';
```

Data Output Messages Notifications

Showing rows: 1 to 4 Page No: 1 of 1

	airline_id [PK] integer	airline_code character varying (30)	airline_name character varying (300)	airline_country character varying (50)	created_at timestamp without time zone	updated_at timestamp without time zone	info character varying (50)
1	203	FR001	AirEasy	France	2025-09-23 23:24:53.188879	2025-09-23 23:24:53.188879	French airline
2	204	BR001	FlyHigh	Brazil	2025-09-23 23:24:53.188879	2025-09-23 23:24:53.188879	Brazilian airline
3	205	PL001	FlyFly	Poland	2025-09-23 23:24:53.188879	2025-09-23 23:24:53.188879	Polish airline
4	201	KZ001	KazAir	Turkey	2025-09-23 23:21:28.044211	2025-09-23 23:21:28.044211	National airline

Total rows: 4 Query complete 00:00:00.101

Successfully run. Total query runtime: 101 msec. 4 rows affected.

13. List the top5 most recently created airlines.

Object Explorer | Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Processes | db\_lab/postgres@... | db\_lab/postgres@PostgreSQL 17\*

db\_lab/postgres@PostgreSQL 17

Query Query History

```
--13 task
2 SELECT * FROM airline_info ORDER BY created_at DESC LIMIT 5;
```

Data Output Messages Notifications

Showing rows: 1 to 5 Page No: 1 of 1

	airline_id [PK] integer	airline_code character varying (30)	airline_name character varying (300)	airline_country character varying (50)	created_at timestamp without time zone	updated_at timestamp without time zone	info character varying (50)
1	205	PL001	FlyFly	Poland	2025-09-23 23:24:53.188879	2025-09-23 23:24:53.188879	Polish airline
2	204	BR001	FlyHigh	Brazil	2025-09-23 23:24:53.188879	2025-09-23 23:24:53.188879	Brazilian airline
3	203	FR001	AirEasy	France	2025-09-23 23:24:53.188879	2025-09-23 23:24:53.188879	French airline
4	201	KZ001	KazAir	Turkey	2025-09-23 23:21:28.044211	2025-09-23 23:21:28.044211	National airline
5	36	QR	Qatar Airways	France	2025-09-22 00:00:00	2025-03-18 00:00:00	phasellus id

Total rows: 5 Query complete 00:00:00.086

Successfully run. Total query runtime: 86 msec. 5 rows affected.

14. Return all rows where booking\_id is between 200 and 300 inclusive and check\_result <> 'Checked'.

Object Explorer | Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Processes | db\_lab/postgres@... | db\_lab/postgres@PostgreSQL 17\*

db\_lab/postgres@PostgreSQL 17

Query Query History

```
--14 task
2 SELECT * FROM baggage_check WHERE booking_id BETWEEN 200 AND 300 AND check_result <> 'Checked';
```

Data Output Messages Notifications

	baggage_check_id [PK] integer	check_result character varying (50)	created_at timestamp without time zone	updated_at timestamp without time zone	booking_id integer	passenger_id integer
1	46	Gallard	2024-11-04 00:00:00	2024-12-01 00:00:00	269	46
2	92	Ingaberg	2024-10-10 00:00:00	2024-10-29 00:00:00	267	92
3	116	Richmound	2024-11-14 00:00:00	2025-09-01 00:00:00	200	116
4	180	Layla	2025-03-14 00:00:00	2025-04-25 00:00:00	297	180

Showing rows: 1 to 4 Page No: 1 of 1

Total rows: 4 Query complete 00:00:00.117

Successfully run. Total query runtime: 117 msec. 4 rows affected.

15. Baggage checks where update\_at is in the same month as created\_at but occurs earlier than created\_at.

Object Explorer | Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Processes | db\_lab/postgres@... | db\_lab/postgres@PostgreSQL 17\*

db\_lab/postgres@PostgreSQL 17

Query Query History

```
--15 task
2 SELECT * FROM baggage_check WHERE DATE_TRUNC('month', updated_at) = DATE_TRUNC('month', created_at)
3 AND updated_at < created_at;
```

Data Output Messages Notifications

	baggage_check_id [PK] integer	check_result character varying (50)	created_at timestamp without time zone	updated_at timestamp without time zone	booking_id integer	passenger_id integer
1	3	Yettie	2025-03-26 00:00:00	2025-03-16 00:00:00	3815	3
2	20	Tynan	2025-04-23 00:00:00	2025-04-07 00:00:00	37	20
3	73	Shelia	2025-04-08 00:00:00	2025-04-05 00:00:00	83882	73
4	97	North	2025-09-17 00:00:00	2025-09-03 00:00:00	62	97
5	110	Adina	2024-10-19 00:00:00	2024-10-10 00:00:00	454	110
6	166	Jacinthe	2025-07-23 00:00:00	2025-07-01 00:00:00	457	166
7	193	Hatti	2024-10-18 00:00:00	2024-10-11 00:00:00	2	193

Showing rows: 1 to 7 Page No: 1 of 1

Total rows: 7 Query complete 00:00:00.101

Successfully run. Total query runtime: 101 msec. 7 rows affected.