

Laboratory Work 4

1. Retrieve all airline names in uppercase.

The screenshot shows a PostgreSQL IDE interface. On the left, the 'Object Explorer' pane displays a database schema with tables like 'airline', 'airline_info', 'airport', 'baggage', 'booking', 'flights', 'passengers', and 'security_check'. The 'airline_info' table is selected. The main query editor shows the following SQL query:

```
--1 task
SELECT UPPER(airline_name) AS airline_name_upper FROM airline_info;
```

The 'Data Output' pane at the bottom displays the results of the query. It shows a table with one column, 'airline_name_upper', and four rows of data:

airline_name_upper
SOUTH AFRICAN AIRWAYS
SINGAPORE AIRLINES
SAS SCANDINAVIAN
AIR INDIA

A status bar at the bottom indicates 'Total rows: 204' and 'Query complete 00:00:00.113'. A green message box at the bottom right states: 'Successfully run. Total query runtime: 113 msec. 204 rows affected.'

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

The screenshot shows a database management tool interface. On the left, the 'Object Explorer' pane displays a tree view of database objects, including tables, columns, constraints, and triggers. The 'booking' table is selected, showing its columns: booking_id, flight_id, passenger_id, booking_platform, created_at, updated_at, status, and price. The main pane displays a SQL query:

```
--2 task
SELECT REPLACE(airline_name, 'Air', 'Aero') AS modified_name FROM airline_info;
```

The 'Data Output' pane shows the results of the query, displaying a table with one column, 'modified_name', and five rows of data:

modified_name
South African Airways
Singapore Airlines
SAS Scandinavian
Aero India
Sinnapore Airlines

The status bar at the bottom indicates 'Total rows: 204' and 'Query complete 00:00:00.090'. A green message box at the bottom right states: 'Successfully run. Total query runtime: 90 msec. 204 rows affected.'

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

The screenshot shows the same database management tool interface. The 'Object Explorer' pane is the same. The main pane displays a SQL query:

```
--3 task
SELECT flight_id FROM flights WHERE airline_id IN (1, 2);
```

The 'Data Output' pane shows the results of the query, displaying a table with one column, 'flight_id', and two rows of data:

flight_id
1
2

The status bar at the bottom indicates 'Total rows: 1' and 'Query complete 00:00:00.195'. A green message box at the bottom right states: 'Successfully run. Total query runtime: 195 msec. 1 rows affected.'

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

The screenshot shows a database management tool interface. On the left, the 'Object Explorer' pane displays a tree view of database objects. The 'airports' table is selected, showing its columns: baggage_id, weight_in_kg, created_at, updated_at, and booking_id. The main query editor displays the following SQL query:

```
--4 task
SELECT airport_name FROM airport WHERE airport_name LIKE '%Reginald%' AND airport_name LIKE '%Air%';
```

The 'Data Output' pane at the bottom shows the results of the query, which is empty. A status bar at the bottom indicates 'Total rows: 0' and 'Query complete 00:00:00.105'. A green message box at the bottom right states 'Successfully run. Total query runtime: 105 msec. 0 rows affected.'

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

The screenshot shows the same database management tool interface. The 'Object Explorer' pane displays the 'passengers' table, showing its columns: baggage_id, weight_in_kg, created_at, updated_at, and booking_id. The main query editor displays the following SQL query:

```
--5 task
SELECT
  CONCAT(first_name, ' ', last_name) AS passenger_name,
  TO_CHAR(date_of_birth, 'Month DD, YYYY') AS formatted_birthdate
FROM passengers;
```

The 'Data Output' pane at the bottom shows the results of the query, which are displayed in a table with two columns: 'passenger_name' and 'formatted_birthdate'. The table contains 200 rows of data. A status bar at the bottom indicates 'Total rows: 200' and 'Query complete 00:00:00.325'. A green message box at the bottom right states 'Successfully run. Total query runtime: 325 msec. 200 rows affected.'

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

The screenshot shows the SQL Enterprise Manager interface. On the left, the Object Explorer displays the database structure, with the 'booking' table selected. The main pane shows a query window with the following SQL code:

```
--6 task
SELECT flight_id FROM flights WHERE act_arrival_time > sch_arrival_time;
```

The Data Output pane at the bottom shows the results of the query:

flight_id [PK] integer
1
2
3
4
5
6
7
8

The status bar indicates: Total rows: 53, Query complete 00:00:00.265. A green message box at the bottom right says: "Successfully run. Total query runtime: 00:00:00.265".

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.

The screenshot shows the SQL Enterprise Manager interface. On the left, the Object Explorer displays the database structure, with the 'passengers' table selected. The main pane shows a query window with the following SQL code:

```
--7 task
SELECT first_name, last_name,
CASE
WHEN EXTRACT(YEAR FROM AGE(date_of_birth)) BETWEEN 18 AND 35 THEN 'Young'
WHEN EXTRACT(YEAR FROM AGE(date_of_birth)) BETWEEN 36 AND 55 THEN 'Adult'
ELSE 'Other'
END AS age_group
FROM passengers;
```

The Data Output pane at the bottom shows the results of the query:

first_name character varying (50)	last_name character varying (50)	age_group text	
1	Jess	Strongman	Other
2	Edgardo	Hawyes	Other
3	Hester	Wooler	Other
4	Jayson	Mogg	Other
5	Xulius	Rinewit	Other

The status bar indicates: Total rows: 200, Query complete 00:00:00.183. A green message box at the bottom right says: "Successfully run. Total query runtime: 183 msec. 200 rows affected."

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

The screenshot shows a PostgreSQL IDE interface. On the left, the 'Object Explorer' pane displays the database structure, with 'Tables (11)' expanded under the 'public' schema. The 'booking' table is selected. The main editor shows a SQL query:

```
1 --8 task
2 SELECT booking_id, price,
3 CASE
4     WHEN price < 100 THEN 'Cheap'
5     WHEN price BETWEEN 100 AND 500 THEN 'Medium'
6     ELSE 'Expensive'
7 END AS price_category
8 FROM booking;
```

Below the query editor, the 'Data Output' pane shows the schema of the result set:

booking_id	price	price_category
[PK] integer	numeric (7,2)	text

At the bottom, a status bar indicates 'Total rows: 0' and 'Query complete 00:00:00.105'. A green notification box at the bottom right states: 'Successfully run. Total query runtime: 105 msec. 0 rows affected.'

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

Object Explorer

- Servers (1)
 - PostgreSQL 17
 - Databases (5)
 - db_lab
 - Castes
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (11)
 - airline
 - airline_info
 - airport
 - baggage
 - Columns (5)
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - baggage_check
 - boarding_pass
 - booking
 - Columns (8)
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers

db_lab/postgres@PostgreSQL 17

Query

```
--9 task
SELECT airline_country,
       COUNT(airline_name) AS total_airlines
FROM airline_info
GROUP BY airline_country;
```

Data Output

airline_country	total_airlines
Turkey	1
Spain	1
Argentina	6
Switzerland	2
New Zealand	1

Showing rows: 1 to 28 | Page No: 1 | of 1

Successfully run. Total query runtime: 75 msec. 28 rows affected.

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

Object Explorer

- Tables (11)
 - airline
 - airline_info
 - airport
 - baggage
 - Columns (5)
 - baggage_id
 - weight_in_kg
 - created_at
 - updated_at
 - booking_id
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - baggage_check
 - boarding_pass
 - booking
 - Columns (8)
 - booking_id
 - flight_id
 - passenger_id
 - booking_platform
 - created_at
 - updated_at
 - status
 - price
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - booking_flight
 - flights
 - passengers
 - security_check
 - Trigger Functions
 - Types
 - Views
 - Subscriptions
- lab10
- phonebook
- Castes
- Catalogs
- Event Triggers

db_lab/postgres@PostgreSQL 17

Query

```
--10 task
SELECT flight_id FROM flights WHERE act_arrival_time > sch_arrival_time;
```

Data Output

flight_id
2
4
6
7
8

Showing rows: 1 to 53 | Page No: 1 | of 1

Successfully run. Total query runtime: 85 msec. 53 rows affected.