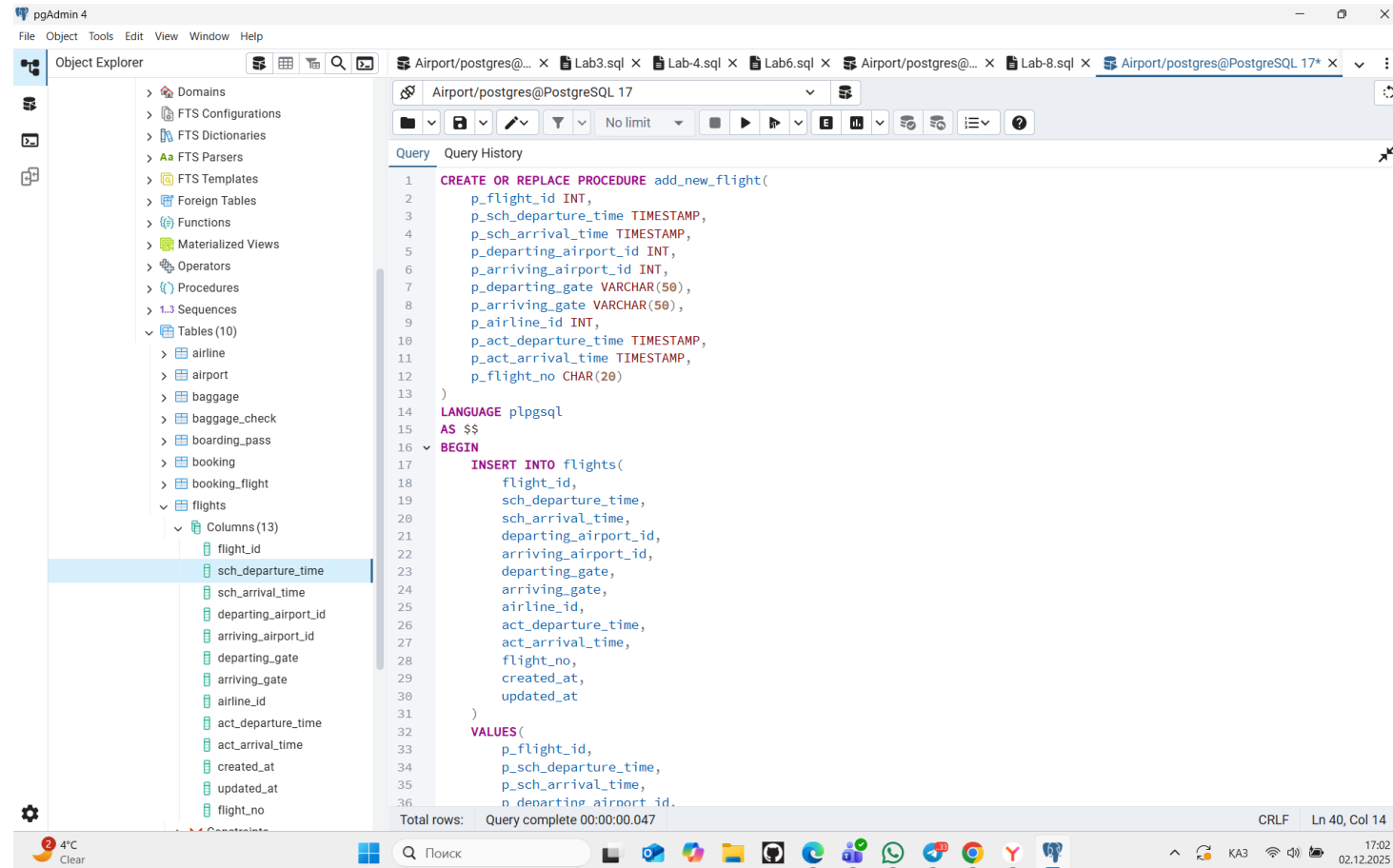


task1



TASK 2

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows a tree structure of database objects. The 'flights' table is selected, and its columns are listed: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, departing_gate, arriving_gate, airline_id, act_departure_time, act_arrival_time, created_at, updated_at, and flight_no. The 'sch_departure_time' column is highlighted.

The main query editor on the right contains the following SQL code:

```
1 CREATE OR REPLACE PROCEDURE update_flight_arrival(  
2   p_flight_id INT,  
3   p_act_arrival_time TIMESTAMP  
4 )  
5  
6 LANGUAGE plpgsql  
7 AS $$  
8 BEGIN  
9     UPDATE flights  
10    SET  
11      act_arrival_time = p_act_arrival_time,  
12      updated_at = CURRENT_TIMESTAMP  
13    WHERE flight_id = p_flight_id;  
14 END;  
15 $$;
```

Below the query editor, the 'Messages' tab shows the execution result: 'Query returned successfully in 47 msec.'

The bottom status bar indicates 'Total rows: Query complete 00:00:00.047' and 'Ln 15, Col 4'.

TASK #3

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows a tree structure of database objects. The 'flights' table is selected, and its columns are listed: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, departing_gate, arriving_gate, airline_id, act_departure_time, act_arrival_time, created_at, updated_at, and flight_no. The 'sch_departure_time' column is highlighted.

The main query editor window shows a SQL query that creates or replaces a procedure named 'get_departing_flights'. The procedure takes 'p_departing_airport_id' as an integer parameter and returns a list of flight details from the 'flights' table where the 'departing_airport_id' matches the parameter.

```
CREATE OR REPLACE PROCEDURE get_departing_flights(  
    p_departing_airport_id INT  
)  
LANGUAGE plpgsql  
AS $$  
BEGIN  
    SELECT  
        flight_id,  
        sch_departure_time,  
        sch_arrival_time,  
        departing_airport_id,  
        arriving_airport_id,  
        departing_gate,  
        arriving_gate,  
        airline_id,  
        act_departure_time,  
        act_arrival_time,  
        flight_no,  
        created_at,  
        updated_at  
    FROM flights  
    WHERE departing_airport_id = p_departing_airport_id;  
END;  
$$;
```

Below the query editor, the 'Messages' tab shows the execution results: 'CREATE PROCEDURE' and 'Query returned successfully in 49 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.049' and 'Ln 67, Col 20'.

TASK #4

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows a tree view of the database schema. The 'flights' table is selected, and its columns are listed: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, departing_gate, arriving_gate, airline_id, act_departure_time, act_arrival_time, created_at, updated_at, and flight_no. The 'sch_arrival_time' column is highlighted.

The main query editor on the right contains the following SQL code:

```
89  
90  
91 CREATE OR REPLACE FUNCTION calculate_average_delay(  
92   p_arrival_airport_id INT  
93 )  
94 RETURNS INTERVAL  
95 LANGUAGE plpgsql  
96 AS $$  
97 DECLARE  
98   avg_delay INTERVAL;  
99 BEGIN  
100   SELECT AVG(act_arrival_time - sch_arrival_time)  
101   INTO avg_delay  
102   FROM flights  
103   WHERE arriving_airport_id = p_arrival_airport_id  
104         AND act_arrival_time IS NOT NULL  
105         AND sch_arrival_time IS NOT NULL;  
106   RETURN avg_delay;  
107 END;  
108 $$;
```

Below the query editor, the 'Data Output' tab shows the message: 'Query returned successfully in 53 msec.'

The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.053' and 'Ln 108, Col 4'.

TASK #5

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows the database structure, with the 'passengers' table under the 'public' schema selected. The main query editor on the right contains a PL/pgSQL procedure named 'get_passengers_by_flight'. The procedure takes a flight number as input and returns a list of passengers for that flight, including their ID, name, last name, passport number, booking ID, seat, and status. The query is executed successfully, as indicated by the 'Data Output' pane at the bottom, which shows a message: 'Query returned successfully in 45 msec.' The status bar at the bottom indicates the query is complete and provides the execution time.

```
109
110 CREATE OR REPLACE PROCEDURE get_passengers_by_flight(
111     p_flight_no CHAR(20)
112 )
113 LANGUAGE plpgsql
114 AS $$
115 BEGIN
116     SELECT
117         p.passenger_id,
118         p.first_name,
119         p.last_name,
120         p.passport_number,
121         b.booking_id,
122         b.seat,
123         b.status
124     FROM passengers p
125     JOIN booking b ON p.passenger_id = b.passenger_id
126     JOIN flights f ON b.flight_id = f.flight_id
127     WHERE f.flight_no = p_flight_no;
128 END;
129 $$;
```

CREATE PROCEDURE

Query returned successfully in 45 msec.

Total rows: Query complete 00:00:00.045 CRLF Ln 129, Col 4

TASK #6

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows the database structure, with the 'passengers' table under the 'public' schema selected. The main query editor on the right contains a PL/pgSQL function definition. The query is as follows:

```
128 END;  
129 $$;  
130  
131 CREATE OR REPLACE PROCEDURE find_most_frequent_flyer()  
132 LANGUAGE plpgsql  
133 AS $$  
134 BEGIN  
135     SELECT  
136         p.passenger_id,  
137         p.first_name,  
138         p.last_name,  
139         p.passport_number,  
140         COUNT(b.booking_id) AS total_flights  
141     FROM passengers p  
142     JOIN booking b ON p.passenger_id = b.passenger_id  
143     GROUP BY p.passenger_id, p.first_name, p.last_name, p.passport_number  
144     ORDER BY total_flights DESC  
145     LIMIT 1;  
146 END;  
147 $$;
```

Below the query editor, the 'Data Output' tab shows the message: 'CREATE PROCEDURE' and 'Query returned successfully in 76 msec.' A green notification bubble at the bottom right also states: 'Query returned successfully in 76 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.076' and 'CRLF Ln 147, Col 4'.

TASK #7

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' shows a tree structure of the database schema, with the 'public' schema expanded and the 'passengers' table selected. The main pane shows a SQL query being executed in the 'Query' tab. The query is a PL/pgSQL procedure named 'find_flights_delayed_over()' that iterates through flight records and raises a notice for flights delayed by more than 24 hours. The 'Messages' tab at the bottom shows the successful execution of the procedure. The status bar at the bottom indicates the query completed in 56 milliseconds.

```
149 CREATE OR REPLACE PROCEDURE find_flights_delayed_over()
150 LANGUAGE plpgsql
151 AS $$
152 DECLARE
153     flight_record RECORD;
154 BEGIN
155     FOR flight_record IN
156     SELECT
157         f.flight_id,
158         f.flight_no,
159         f.sch_departure_time,
160         f.act_departure_time,
161         (f.act_departure_time - f.sch_departure_time) AS delay_interval
162     FROM flights f
163     WHERE (f.act_departure_time - f.sch_departure_time) > INTERVAL '24 hours'
164     LOOP
165         RAISE NOTICE 'Flight ID: %, Number: %, Scheduled: %, Actual: %, Delay: %',
166             flight_record.flight_id,
167             flight_record.flight_no,
168             flight_record.sch_departure_time,
169             flight_record.act_departure_time,
170             flight_record.delay_interval;
171     END LOOP;
172 END;
173 $$;
```

Query returned successfully in 56 msec.

Total rows: Query complete 00:00:00.056 CRLF Ln 173, Col 4

TASK #8

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' shows a tree structure of database objects. The 'public' schema is expanded, and the 'passengers' table is selected. The main pane shows a SQL query editor with the following code:

```
167 flight_record.flight_no,  
168 flight_record.sch_departure_time,  
169 flight_record.act_departure_time,  
170 flight_record.delay_interval;  
171  
172 END LOOP;  
173  
174 END;  
175 $$;  
176  
177 CREATE OR REPLACE FUNCTION count_flights()  
178 RETURNS TABLE (  
179     airline_id INT,  
180     flight_count BIGINT  
181 )  
182 LANGUAGE plpgsql  
183 AS $$  
184 BEGIN  
185     RETURN QUERY  
186     SELECT  
187         f.airline_id,  
188         COUNT(f.flight_id) AS flight_count  
189     FROM flights f  
190     GROUP BY f.airline_id  
191     ORDER BY flight_count DESC;  
192 END;  
193 $$;
```

Below the query editor, the 'Data Output' tab shows the message: 'Query returned successfully in 52 msec.' A green notification bar at the bottom right also displays: '✓ Query returned successfully in 52 msec. ✕'. The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.052' and 'CRLF Ln 191, Col 4'.

TASK #9

pgAdmin 4

Object Explorer

- 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 (10)
 - airline
 - airport
 - baggage
 - baggage_check
 - boarding_pass
 - booking
 - booking_flight
 - flights
 - passengers
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - security_check
 - Trigger Functions

Query

```
192 CREATE OR REPLACE PROCEDURE avg_ticket(  
193   p_flight_id INT,  
194   OUT p_avg_price DECIMAL,  
195   OUT p_flight_no CHAR(20)  
196 )  
197 LANGUAGE plpgsql  
198 AS $$  
199 BEGIN  
200   SELECT  
201     AVG(b.ticket_price),  
202     f.flight_no  
203   INTO  
204     p_avg_price,  
205     p_flight_no  
206   FROM booking b  
207   JOIN flights f ON b.flight_id = f.flight_id  
208   WHERE b.flight_id = p_flight_id  
209   GROUP BY f.flight_no;  
210 END;  
211 $$;  
212  
213
```

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 51 msec.

Total rows: Query complete 00:00:00.051

✓ Query returned successfully in 51 msec. ✕

CRLF Ln 213, Col 1

3°C Clear

Поиск

18:10 02.12.2025

TASK #10

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows a tree structure of database objects. The 'public' schema is expanded, and the 'passengers' table is selected. The main query editor on the right contains a PL/pgSQL procedure named 'find_flight'. The procedure has four output parameters: 'p_flight_no' (CHAR(20)), 'p_departure_airport' (VARCHAR(50)), 'p_arrival_airport' (VARCHAR(50)), and 'p_ticket_price' (DECIMAL). The procedure body uses a SELECT statement with JOINs to retrieve flight information from the 'booking', 'flights', and 'airport' tables, ordered by ticket price and limited to one result. A toolbar above the editor includes an 'Execute query' button with an 'Alt | F5' shortcut. Below the editor, the 'Data Output' tab shows the execution status: 'Query returned successfully in 62 msec.' and 'Query complete 00:00:00.062'. A green notification bubble at the bottom right confirms the successful execution. The Windows taskbar at the bottom shows the system clock as 18:19 on 02.12.2025.

```
213
214
215 CREATE OR REPLACE PROCEDURE find_flight(
216   OUT p_flight_no CHAR(20),
217   OUT p_departure_airport VARCHAR(50),
218   OUT p_arrival_airport VARCHAR(50),
219   OUT p_ticket_price DECIMAL
220 )
221 LANGUAGE plpgsql
222 AS $$
223 BEGIN
224   SELECT
225     f.flight_no,
226     dep.airport_name,
227     arr.airport_name,
228     b.ticket_price
229   INTO
230     p_flight_no,
231     p_departure_airport,
232     p_arrival_airport,
233     p_ticket_price
234   FROM booking b
235   JOIN flights f ON b.flight_id = f.flight_id
236   JOIN airport dep ON f.departing_airport_id = dep.airport_id
237   JOIN airport arr ON f.arriving_airport_id = arr.airport_id
238   ORDER BY b.ticket_price DESC
239   LIMIT 1;
240 END;
241 $$;
```

Query returned successfully in 62 msec.

Query complete 00:00:00.062

✓ Query returned successfully in 62 msec. ✕

CRLF Ln 241, Col 4