

Lab-10

1. Create a stored procedure to insert a new flight into the flights table.

```
209 --1
210
211 CREATE OR REPLACE PROCEDURE insert_flight(
212     p_flight_number VARCHAR,
213     p_airline_id INT,
214     p_origin_airport VARCHAR,
215     p_destination_country VARCHAR,
216     p_created_at DATE,
217     p_actual_departure TIMESTAMP,
218     p_scheduled_departure TIMESTAMP,
219     p_departure_airport_id VARCHAR,
220     p_arrival_airport_id VARCHAR,
221     p_status VARCHAR
222 )
223 LANGUAGE plpgsql
224 AS $$
225 BEGIN
226     INSERT INTO flights(
227         flight_number, airline_id, origin_airport, destination_country, created_at,
228         actual_departure, scheduled_departure, departure_airport_id, arrival_airport_id, status
229     )
230     VALUES (
231         p_flight_number, p_airline_id, p_origin_airport, p_destination_country, p_created_at,
232         p_actual_departure, p_scheduled_departure, p_departure_airport_id, p_arrival_airport_id, p_status
233     );
234 END;
235 $$;
236 CALL insert_flight('KC9999', 1, 'Almaty', 'France', '2024-12-02', '2024-12-02 08:00', '2024-12-02 08:30', 'ALA', 'CDG', 'Scheduled');
237 SELECT * FROM flights WHERE flight_number = 'KC9999';
Data Output Messages Notifications
Showing rows: 1 to 1 Page No: 1 of 1


| flight_id | flight_number | airline_id | origin_airport | destination_country | created_at | actual_departure    | scheduled_departure | departure_airport_id | arrival_airport_id | status    |
|-----------|---------------|------------|----------------|---------------------|------------|---------------------|---------------------|----------------------|--------------------|-----------|
| 1         | KC9999        | 1          | Almaty         | France              | 2024-12-02 | 2024-12-02 08:00:00 | 2024-12-02 08:30:00 | ALA                  | CDG                | Scheduled |


```

2. Create a stored procedure to update the status of a flight.

```
239 --2
240
241 CREATE OR REPLACE PROCEDURE update_flight_status(
242     p_flight_id INT,
243     p_status VARCHAR
244 )
245 LANGUAGE plpgsql
246 AS $$
247 BEGIN
248     UPDATE flights
249     SET status = p_status
250     WHERE flight_id = p_flight_id;
251 END;
252 $$;
253 CALL update_flight_status(1, 'Delayed');
254 SELECT flight_id, flight_number, status FROM flights WHERE flight_id = 1;
Data Output Messages Notifications
Showing rows: 1 to 1 Page No: 1 of 1


| flight_id | flight_number | status  |
|-----------|---------------|---------|
| 1         | KC2501        | Delayed |


```

3. Create a stored procedure that returns a list of flights departing from a specific airport.

```

256 --3
257
258 CREATE OR REPLACE PROCEDURE flights_from_airport(
259     p_departure_airport_id VARCHAR
260 )
261 LANGUAGE plpgsql
262 AS $$
263 BEGIN
264     RAISE NOTICE 'Flights from %:', p_departure_airport_id;
265     PERFORM * FROM flights WHERE departure_airport_id = p_departure_airport_id;
266 END;
267 $$;
268 SELECT * FROM flights WHERE departure_airport_id = 'ALA';
269

```

Data Output Messages Notifications

	flight_id [PK] integer	flight_number character varying (10)	airline_id integer	origin_airport character varying (50)	destination_country character varying (50)	created_at date	actual_departure timestamp without time zone	scheduled_departure timestamp without time zone	departure_airport_id character varying (10)	arrival_airport_id character varying (10)	status character varying (50)
1	12	DV4512	2	Almaty	Russia	2024-08-01	2024-08-01 15:55:00	2024-08-01 15:30:00	ALA	SVO	Delayed
2	21	KC9999	1	Almaty	France	2024-12-02	2024-12-02 08:00:00	2024-12-02 08:30:00	ALA	CDG	Scheduled
3	1	KC2501	1	Almaty	Turkey	2024-06-05	2024-06-05 08:00:00	2024-06-06 08:30:00	ALA	IST	Delayed

4. Create a function to calculate the average delay time of flights arriving at a specific airport.

```

270 --4
271
272 CREATE OR REPLACE FUNCTION avg_delay_at_airport_hm(p_arrival_airport VARCHAR)
273 RETURNS TABLE(hours INT, minutes INT)
274 LANGUAGE plpgsql
275 AS $$
276 BEGIN
277     RETURN QUERY
278     SELECT
279         TRUNC(avg_delay)::INT AS hours,
280         ROUND((avg_delay - TRUNC(avg_delay)) * 60)::INT AS minutes
281     FROM (
282         SELECT AVG(EXTRACT(EPOCH FROM (actual_departure - scheduled_departure))/3600) AS avg_delay
283         FROM flights
284         WHERE arrival_airport_id = p_arrival_airport
285     ) AS sub;
286 END;
287 $$;
288 SELECT * FROM avg_delay_at_airport_hm('IST');
289

```

Data Output Messages Notifications

	hours integer	minutes integer
1	-12	-3

5. Create a stored procedure that lists all passengers for a given flight number.

```

290 --5
291
292 CREATE OR REPLACE PROCEDURE passengers_by_flight(p_flight_number VARCHAR)
293 LANGUAGE plpgsql
294 AS $$
295 BEGIN
296     PERFORM *
297     FROM passengers p
298     JOIN bookings b ON p.passenger_id = b.passenger_id
299     JOIN flights f ON b.flight_id = f.flight_id
300     WHERE f.flight_number = p_flight_number;
301 END;
302 $$;
303 SELECT p.*
304 FROM passengers p
305 JOIN bookings b ON p.passenger_id = b.passenger_id
306 JOIN flights f ON b.flight_id = f.flight_id
307 WHERE f.flight_number = 'KC2501';

```

Data Output Messages Notifications

	passenger_id [PK] integer	first_name character varying (50)	last_name character varying (50)	gender character varying (10)	birth_date date	passport_number character varying (20)	country_of_citizenship character varying (50)
1	1	Бауыржан	Тлеубеков	Male	1997-03-22	KZ123456	Kazakhstan

6. Create a stored procedure to find the passenger who has taken the greatest number of flights.

```

309  --6
310
311  CREATE OR REPLACE PROCEDURE passenger_most_flights()
312  LANGUAGE plpgsql
313  AS $$
314  BEGIN
315      PERFORM passenger_id, COUNT(flight_id) AS flights_count
316      FROM bookings
317      GROUP BY passenger_id
318      ORDER BY flights_count DESC
319      LIMIT 1;
320
321  END;
322  $$;
323  SELECT passenger_id, COUNT(flight_id) AS flights_count
324  FROM bookings
325  GROUP BY passenger_id
326  ORDER BY flights_count DESC
327  LIMIT 1;

```

Data Output Messages Notifications

SQL

	passenger_id	flights_count
1	integer	bigint
1	11	1

7. Create a stored procedure to find all flights that are delayed by more than 24 hours.

```

328  --7
329
330  CREATE OR REPLACE PROCEDURE flights_delayed_24h()
331  LANGUAGE plpgsql
332  AS $$
333  BEGIN
334      PERFORM *
335      FROM flights
336      WHERE actual_departure - scheduled_departure > INTERVAL '24 hours';
337  END;
338  $$;
339  SELECT *
340  FROM flights
341  WHERE actual_departure - scheduled_departure > INTERVAL '24 hours';

```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

flight_id	flight_number	airline_id	origin_airport	destination_country	created_at	actual_departure	scheduled_departure	departure_airport_id	arrival_airport_id	status
[PK] integer	character varying (10)	integer	character varying (50)	character varying (50)	date	timestamp without time zone	timestamp without time zone	character varying (10)	character varying (10)	character varying (50)
1	9 AF1209	9	Paris	USA	2024-07-15	2024-07-16 09:10:00	2024-07-15 08:45:00	CDG	JFK	Delayed

8. Create a function that counts the number of flights for each airline.

```

342
343 --8
344
345 CREATE OR REPLACE FUNCTION count_flights_per_airline()
346 RETURNS TABLE(airline_id INT, flight_count INT)
347 AS $$ 
348 BEGIN
349   RETURN QUERY
350   SELECT f.airline_id, COUNT(*)::INT
351   FROM flights AS f
352   GROUP BY f.airline_id;
353 END;
354 $$ LANGUAGE plpgsql;
355 SELECT * FROM count_flights_per_airline();

```

Data Output Messages Notifications

	airline_id	flight_count
	integer	integer
1	9	2
2	3	2
3	5	2
4	4	2
5	10	2
6	6	2
7	2	2
8	7	2
9	1	3
10	8	2

9. Create a stored procedure to calculate the average ticket price for a specific flight.

```

356
357 --9
358
359 CREATE OR REPLACE PROCEDURE avg_ticket_price(p_flight_number VARCHAR)
360 LANGUAGE plpgsql
361 AS $$ 
362 BEGIN
363   PERFORM AVG(t.price)
364   FROM tickets t
365   JOIN bookings b ON t.booking_id = b.booking_id
366   JOIN flights f ON b.flight_id = f.flight_id
367   WHERE f.flight_number = p_flight_number;
368 END;
369 $$;
370 SELECT AVG(t.price) AS avg_price
371 FROM tickets t
372 JOIN bookings b ON t.booking_id = b.booking_id
373 JOIN flights f ON b.flight_id = f.flight_id
374 WHERE f.flight_number = 'KC2501';

```

Data Output Messages Notifications

	avg_price
	numeric
1	250.0000000000000000

10.Create a stored procedure to find the flight with the highest ticket price. The procedure should return the flight number, the departure and arrival airports, and the ticket price for the most expensive flight.

```
376 --10
377
378 CREATE OR REPLACE PROCEDURE most_expensive_flight()
379 LANGUAGE plpgsql
380 AS $$ 
381 BEGIN
382     PERFORM f.flight_number, f.departure_airport_id, f.arrival_airport_id, MAX(t.price)
383     FROM tickets t
384     JOIN bookings b ON t.booking_id = b.booking_id
385     JOIN flights f ON b.flight_id = f.flight_id
386     GROUP BY f.flight_number, f.departure_airport_id, f.arrival_airport_id
387     ORDER BY MAX(t.price) DESC
388     LIMIT 1;
389 
390 END;
391 $$;
392 SELECT f.flight_number, f.departure_airport_id, f.arrival_airport_id, MAX(t.price) AS max_price
393 FROM tickets t
394 JOIN bookings b ON t.booking_id = b.booking_id
395 JOIN flights f ON b.flight_id = f.flight_id
396 GROUP BY f.flight_number, f.departure_airport_id, f.arrival_airport_id
397 ORDER BY max_price DESC
398 LIMIT 1;
```

Data Output Messages Notifications

SQL

	flight_number character varying (10)	departure_airport_id character varying (10)	arrival_airport_id character varying (10)	max_price numeric
1	0F6714	KGF	TBS	420.00