

Όνομα: Χαριτούδης Απόστολος-Ανδρέας

A.M: Π17178

Μάθημα: Βάσεις Δεδομένων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ (ΠΑΠΕΙ)

## ΕΡΩΤΗΜΑ 1

### Database: Airline Database

Η βάση δεδομένων περιέχει 8 πίνακες, οι οποίοι είναι: Bookings, Tickets, Tickets\_flights, Flights, Boarding\_passes, Seats, Aircrafts\_data και Airports\_data.

Το Bookings περιλαμβάνει τα: book\_ref, book\_date και total\_amount. Το book\_ref είναι τύπου character, μήκους 6 και είναι το primary key του πίνακα αυτού, καθώς χρησιμοποιείται για να κάνουμε μοναδικές συνδέσεις στα εισιτήρια και τους επιβάτες, καθώς και στην πτήση τους. Το book\_date είναι τύπου timestamp με timezone και οι ημερομηνίες που αναγράφονται σε αυτή την στήλη δεν είναι νωρίτερα από ένα μήνα από την πτήση τους. Τέλος το total\_amount είναι τύπου numeric (10,2) και χρησιμοποιείται για να δηλώσει το συνολικό ποσό που χρεώθηκε αυτός που έκανε την κράτηση. Να σημειωθεί πως όπως ζητήθηκε στην εργασία μια κράτηση μπορεί να περιλαμβάνει περισσότερα από ένα εισιτήρια και το total\_amount περιγράφει την συνολική τους αξία.

Το Tickets περιλαμβάνει τα: ticket\_no, book\_ref, passenger\_id, passenger\_name και contact\_data. Το ticket\_no είναι το νούμερο του εισιτηρίου, τύπου character, μήκους 13 και λειτουργεί ως primary key, καθώς είναι unique (εδώ επειδή έβαλα καταλάθος διπλότυπες τιμές δεν μου επέτρεπε να το κάνω primary key). Εδώ το book\_ref αναφέρεται για να γίνει συσχέτιση με τα στοιχεία των επιβατών και το εισιτήριό τους. Λειτουργεί επίσης σαν ξένο κλειδί. Το passenger\_id είναι τύπου character varying, μήκους 20. Το passenger\_name είναι τύπου text και το contact\_data είναι jsonb.

Το Tickets\_flights περιλαμβάνει τα: ticket\_no, flight\_id, fare\_conditions και amount. Το ticket\_no που σε συνδυασμό με το flight\_id λειτουργεί ως primary key. Το flight\_id είναι τύπου integer και δεν επιτρέπεται να πάρει τιμή  $\leq$  του 0. Επιπλέον, το flight\_id είναι και ξένο κλειδί. Το fare\_conditions είναι character varying, μήκους 10 και παίρνει τιμές Economy, First για first class και Business. Το amount περιγράφει την τιμή του εισιτηρίου. Σε αυτήν την περίπτωση εδώ περιλαμβάνεται μόνο η τιμή του ενός εισιτηρίου.

Το Flights περιλαμβάνει τα: flight\_id, flight\_no, scheduled\_departure, departure\_airport, arrival\_airport, status, aircraft\_code, actual\_departure, distance και actual\_arrival. Το flight\_id είναι το ίδιο με αυτό που αναφέρθηκε στο tickets\_flights. Το flight\_no είναι τύπου character, μήκους 6 και έχει δηλωθεί ως unique σε συνδυασμό με το scheduled\_departure, το οποίο είναι τύπου timestamp με timezone. Αυτός ο συνδυασμός χαρακτηρίστηκε unique, καθώς δεν γίνεται μια πτήση να φεύγει από διαφορετικά σημεία ταυτόχρονα. Το departure\_airport και arrival\_airport είναι τύπου character 3 και είναι η κωδική ονομασία των αεροδρομίων. Το status είναι character varying 20 και περιγράφει την κατάσταση της πτήσης, η οποία εξαρτάται από την function timesnap, η οποία έχει ένα time snapshot με ημερομηνία 30/06/2022 23:59 (αρχική ημερομηνία παράδοσης εργασίας), αλλά δεν χρησιμοποιείται κάπου, παρά σαν οδηγός για να καθοριστούν οι καταστάσεις των πτήσεων: On time ή delayed για πτήσεις που είναι να γίνουν την επόμενη μέρα, departed για πτήσεις που έχουν φύγει πριν από αυτήν την ημερομηνία, αλλά δεν έχουν φτάσει (δηλαδή η ημερομηνία άφιξης είναι μεταγενέστερη αυτής), arrived για όσες πτήσεις έχουν φτάσει στον προορισμό τους πριν από αυτήν την ημερομηνία, scheduled για όσες πτήσεις είναι να γίνουν εντός 1 μήνα με βάση αυτή την ημερομηνία και cancelled, η κατάσταση στην οποία η πτήση ακυρώνεται. Όμως, αυτή η κατάσταση δεν έχει κάποιο χρονικό περιθώριο καθώς μπορεί να γίνει για πτήσεις που είναι να πραγματοποιηθούν και για πτήσεις που ήταν να πραγματοποιηθούν, αλλά δεν πραγματοποιήθηκαν. Το aircraft\_code είναι ο 3ψήφιος κωδικός των αεροπλάνων και είναι μοναδικός και είναι τύπου character, μήκους 3 και είναι ξένο κλειδί. Το , actual\_departure και actual\_arrival είναι τύπου timestamp και περιγράφουν την ημερομηνία που τελικά πραγματοποιήθηκε κάποια πτήση. Τέλος το distance είναι τύπου integer και μας δίνει πληροφορίες για την απόσταση των 2 αεροδρομίων (αεροδρόμιο αναχώρησης, αεροδρόμιο άφιξης) σε km.

To Boarding\_passes περιλαμβάνει τα: ticket\_no, flight\_id, boarding\_no και seat\_no. Το ticket\_no είναι όπως αναφέρθηκε πριν ο αριθμός εισιτηρίου που είναι unique. Seat\_no είναι τυπου char var μήκους 4 και δηλώνει σε ποια σειρά και νούμερο καθεται ο επιβατης. Το boarding\_no είναι τυπου int και δείχνει προτεραιότητα επιβίβασης.

To Seats περιλαμβάνει τα: aircraft\_code, seat\_no και fare\_conditions. Το seats δείχνει όλες τις υπαρκτες θέσεις κάθε αεροπλανου που υπαρχεις την βάση, καθώς και το κόστος και κατηγορία κάθε θέσης.

To Aircrafts\_data περιλαμβάνει τα: aircraft\_code, model και range. Περιλαμβάνει τον κωδικό του αεροπλανου, το μοντελο και την χιλιομετρικη αποσταση που μπορεί να διανυσει.

To Airports\_data περιλαμβάνει τα: airport\_code, airport\_name, city, coordinates και timezone. Περιλαμβάνει τους κωδικους των αεροδρομιων, την γεωμετρικη τους τοποθεσια, το timezone στο οποιο ανηκουν και την πολη στην οποια ανηκουν.

## BCNF

Όλοι οι πίνακες είναι σε μορφή BCNF καθώς κανενας πίνακας και πιο σωστα κανένα στοιχείο του πίνακα δεν εξαρτάται από μη κλειδί attribute και δεν έχουν διπλοτυπες στήλες.

## Flights\_View και Routes\_View

### Flights\_View

Το Flights\_View πραγματοποιείται έτσι:

```
CREATE OR REPLACE VIEW public.routes_view
```

```
AS
```

```
WITH f3 AS (
```

```
    SELECT f2.flight_no,
```

```
        f2.departure_airport,
```

```
        f2.arrival_airport,
```

```
        f2.aircraft_code,
```

```
        f2.duration,
```

```
        array_agg(f2.days_of_week) AS days_of_week
```

```
FROM ( SELECT f1.flight_no,
```

```
        f1.departure_airport,
```

```
        f1.arrival_airport,
```

```
        f1.aircraft_code,
```

```
        f1.duration,
```

```

        f1.days_of_week
    FROM ( SELECT flights.flight_no,
        flights.departure_airport,
        flights.arrival_airport,
        flights.aircraft_code,
        flights.scheduled_arrival - flights.scheduled_departure AS duration,
        to_char(flights.scheduled_departure, 'ID':text)::integer AS days_of_week
        FROM flights) f1

    GROUP BY f1.flight_no, f1.departure_airport, f1.arrival_airport, f1.aircraft_code, f1.duration,
f1.days_of_week

    ORDER BY f1.flight_no, f1.departure_airport, f1.arrival_airport, f1.aircraft_code, f1.duration,
f1.days_of_week) f2

    GROUP BY f2.flight_no, f2.departure_airport, f2.arrival_airport, f2.aircraft_code, f2.duration
)

SELECT f3.flight_no,
    f3.departure_airport,
    dep.airport_name AS departure_airport_name,
    dep.city AS departure_city,
    f3.arrival_airport,
    arr.airport_name AS arrival_airport_name,
    arr.city AS arrival_city,
    f3.aircraft_code,
    f3.duration,
    f3.days_of_week
FROM f3,
    airports dep,
    airports arr
WHERE f3.departure_airport = dep.airport_code AND f3.arrival_airport = arr.airport_code;

ALTER TABLE public.routes_view
    OWNER TO postgres;

```

Routes\_View

To Routes\_View

πραγματοποιείται έτσι:

```
CREATE OR REPLACE VIEW public.flights_view
AS
SELECT f.flight_id,
       f.flight_no,
       f.scheduled_departure,
       timezone(dep.timezone, f.scheduled_departure) AS scheduled_departure_local,
       f.scheduled_arrival,
       timezone(arr.timezone, f.scheduled_arrival) AS scheduled_arrival_local,
       f.scheduled_arrival - f.scheduled_departure AS scheduled_duration,
       f.departure_airport,
       dep.airport_name AS departure_airport_name,
       dep.city AS departure_city,
       f.arrival_airport,
       arr.airport_name AS arrival_airport_name,
       arr.city AS arrival_city,
       f.status,
       f.aircraft_code,
       f.actual_departure,
       timezone(dep.timezone, f.actual_departure) AS actual_departure_local,
       f.actual_arrival,
       timezone(arr.timezone, f.actual_arrival) AS actual_arrival_local,
       f.actual_arrival - f.actual_departure AS actual_duration
FROM flights f,
     airports dep,
     airports arr
WHERE f.departure_airport = dep.airport_code AND f.arrival_airport = arr.airport_code;
```

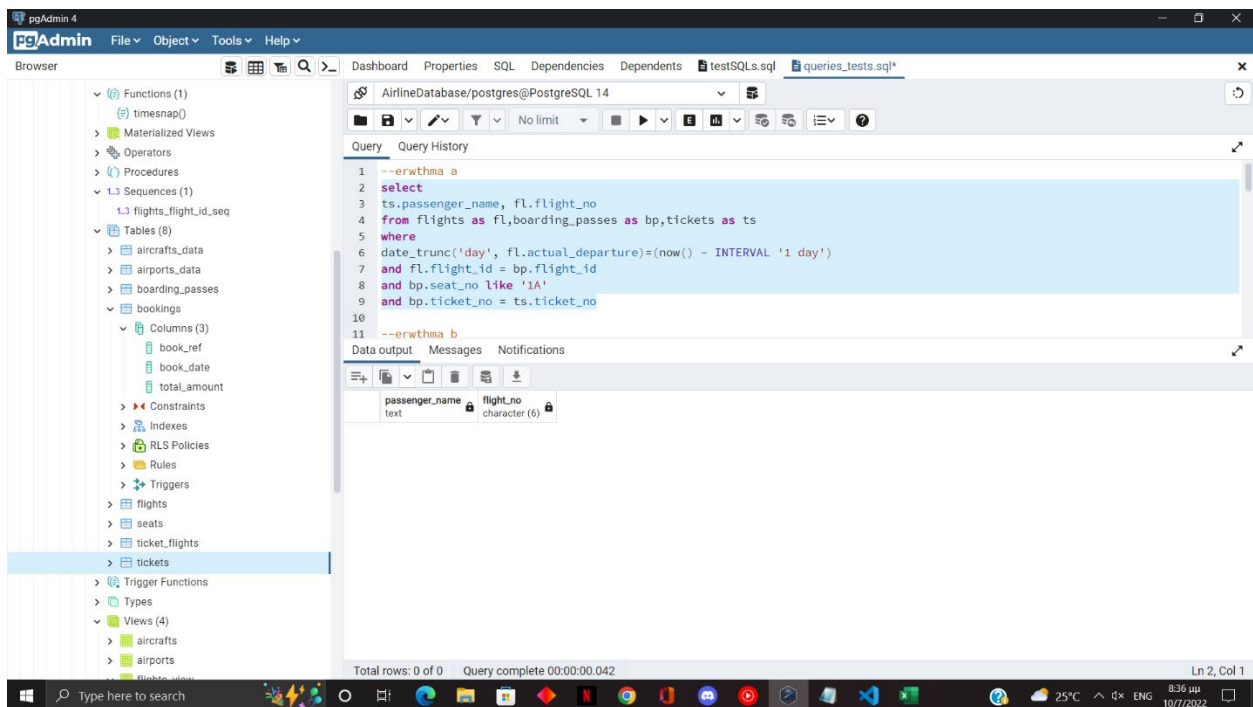
ALTER TABLE public.flights\_view  
OWNER TO postgres;

## ΕΡΩΤΗΜΑ 2

### Queries

Όλα τα queries του ερωτήματος 2 βρίσκονται στο .sql αρχείο.

α. Για να επιτευχθεί το ζητούμενο της ερώτησης αυτής έγραψα το εξής query (δίνεται παρακάτω) αποτελέσματα



β. Για να επιτευχθεί το ζητούμενο της ερώτησης αυτής έγραψα το εξής query (δίνεται παρακάτω) αποτελέσματα. Επειδή στο πρώτο ερωτημα δεν εμφανίζεται καποια πτηση εβαλα μια τυχαια για να εμφανιστει

The screenshot shows the pgAdmin 4 interface. On the left, the 'Tables' browser is expanded, showing the 'flights' table. The main pane displays a SQL query in the 'Query' editor:

```

11 --erwthma b
12 select count(sts.seat_no) as total_free
13 from seats as sts
14 where not exists
15 (
16 select bp.seat_no
17 from flights as fl, boarding_passes as bp
18 where fl.flight_no='PG0124'
19 and bp.flight_id=fl.flight_id
20 and sts.seat_no=bp.seat_no
21 )

```

The 'Data output' pane shows the result of the query:

total_free
130

At the bottom, a status bar indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.036'.

γ. Για να επιτευχθεί το ζητούμενο της ερώτησης αυτής έγραψα το εξής query (δίνεται παρακάτω) αποτελέσματα

The screenshot shows the pgAdmin 4 interface. The 'Query' editor contains two queries:

```

21 )
22
23
24 --erwthma c
25 select fl.actual_departure - fl.scheduled_departure as f_delay, fl.flight_no
26 from flights as fl
27 order by f_delay desc
28 limit 5;
29
30 --erwthma d
31 select tks.passenger_name, sum(fl.distance) as total_distance_km

```

The 'Data output' pane shows the results of the first query (lines 25-28):

f_delay	flight_no
[null]	PG0035
[null]	PG0042
[null]	PG0019
[null]	PG0056
[null]	PG0043

A green status bar at the bottom indicates 'Successfully run. Total query runtime: 46 msec. 5 rows affected.' and 'Total rows: 5 of 5'.

δ. Για να επιτευχθεί το ζητούμενο της ερώτησης αυτής έγραψα το εξής query (δίνεται παρακάτω)

pgAdmin 4

File Object Tools Help

Browser

- Functions (1)
  - timesnap()
- Materialized Views
- Operators
- Procedures
- Sequences (1)
  - flights\_flight\_id\_seq
- Tables (8)
  - aircrafts\_data
  - airports\_data
  - boarding\_passes
  - bookings
  - flights**
  - seats
  - ticket\_flights
  - tickets
- Trigger Functions
- Types
- Views (4)
  - aircrafts
  - airports
  - flights\_view
  - routes\_view
- Columns (10)
  - flight\_no
  - departure\_airport
  - departure\_airport\_name
  - departure\_city

AirlineDatabase/postgres@PostgreSQL 14

Query Query History

```
31 select tks.passenger_name, sum(fl.distance) as total_distance_km
32 from flights as fl, ticket_flights as tf, tickets as tks
33 where fl.flight_id=tf.flight_id
34 and tf.ticket_no= tks.ticket_no
35 and date_trunc('year', fl.scheduled_departure ) = date_trunc('year', now())
36 group by tks.passenger_name
37 order by total_distance_km desc
38 limit 5;
39
40 --erwthma e
41 select ct.city, count(tf.ticket_no) as total_visits
```

Data output Messages Notifications

	passenger_name text	total_distance_km bigint
1	Manfred Hamst...	10020
2	Ashlen Lorimer	3835
3	Tony Peet	3835
4	Avictor Philipps	3835
5	Charleen Caddies	3835

Total rows: 5 of 5 Query complete 00:00:00.049 Ln 31, Col 1

ε. Για να επιτευχθεί το ζητούμενο της ερώτησης αυτής έγραψα το εξής query (δίνεται παρακάτω)



pgAdmin 4

File Object Tools Help

Browser

- Functions (1)
  - timesnap()
- Materialized Views
- Operators
- Procedures
- Sequences (1)
  - flights\_flight\_id\_seq
- Tables (8)
  - aircrafts\_data
  - airports\_data
  - boarding\_passes
  - bookings
  - flights
  - seats
  - ticket\_flights
  - tickets
- Trigger Functions
- Types
- Views (4)
  - aircrafts
  - airports
  - flights\_view
    - Columns (20)
    - Rules
    - Triggers
  - routes\_view
    - Columns (10)
      - flight\_no
      - departure\_airport
      - departure\_airport\_name
      - departure\_city

Dashboard Properties SQL Dependencies Dependents testSQLs.sql queries\_tests.sql\* public.flights/...

AirlineDatabase/postgres@PostgreSQL 14

Query Query History

```
31 select tks.passenger_name, sum(fl.distance) as total_distance_km
32 from flights as fl, ticket_flights as tf, tickets as tks
33 where fl.flight_id=tf.flight_id
34 and tf.ticket_no= tks.ticket_no
35 and date_trunc('year', fl.scheduled_departure ) = date_trunc('year', now())
36 group by tks.passenger_name
37 order by total_distance_km desc
38 limit 5;
39
--erwthma e
41 select ct.city, count(tf.ticket_no) as total_visits
42 from airports_data as ct, flights as fl, ticket_flights as tf
43 where ct.airport_code=fl.actual_airport
```

Data output Messages Notifications

	passenger_name	total_distance_km
	text	bigint
1	Manfred Harmst...	10020
2	Ashlen Lorimer	3835
3	Tony Peet	3835
4	Avictor Philippe	3835
5	Charleen Caddies	3835

Total rows: 5 of 5 Query complete 00:00:00.042

Successfully run. Total query runtime: 42 msec. 5 rows affected.

pgAdmin 4

File Object Tools Help

Browser

- Functions (1)
  - timesnap()
- Materialized Views
- Operators
- Procedures
- Sequences (1)
  - flights\_flight\_id\_seq
- Tables (8)
  - aircrafts\_data
  - airports\_data
  - boarding\_passes
  - bookings
  - flights
  - seats
  - ticket\_flights
  - tickets
- Trigger Functions
- Types
- Views (4)
  - aircrafts
  - airports
  - flights\_view
    - Columns (20)
    - Rules
    - Triggers
  - routes\_view
    - Columns (10)
      - flight\_no
      - departure\_airport
      - departure\_airport\_name
      - departure\_city

Dashboard Properties SQL Dependencies Dependents testSQLs.sql queries\_tests.sql\* public.flights/...

AirlineDatabase/postgres@PostgreSQL 14

Query Query History

```
47 order by total_visits desc
48 limit 5;
49
--erwthma f
52 select tks.passenger_name,min(bp.boarding_no) as faster_boarding
53 from ticket_flights as tf, boarding_passes as bp, tickets as tks
54 where bp.ticket_no=tf.ticket_no
55 and tf.ticket_no=tks.ticket_no
56 group by tks.passenger_name
57 HAVING count(tks.passenger_name)>2
58 order by faster_boarding asc
```

Data output Messages Notifications

	passenger_name	faster_boarding
	text	integer

Total rows: 0 of 0 Query complete 00:00:00.042

Successfully run. Total query runtime: 42 msec. 0 rows affected.

pgAdmin 4

File Object Tools Help

Browser

Functions (1)  
timesnap()

Materialized Views

Operators

Procedures

Sequences (1)  
flights\_flight\_id\_seq

Tables (8)  
aircrafts\_data  
airports\_data  
boarding\_passes  
bookings  
flights  
seats  
ticket\_flights  
tickets

Trigger Functions

Types

Views (4)  
aircrafts  
airports  
flights\_view  
Columns (20)  
Rules  
Triggers  
routes\_view  
Columns (10)  
flight\_no  
departure\_airport  
departure\_airport\_name  
departure\_city

AirlineDatabase/postgres@PostgreSQL 14

Query Query History

```
36 group by tks.passenger_name
37 order by total_distance_km desc
38 limit 5;
39
40 --erwthma e
41 select ct.city, count(tf.ticket_no) as total_visits
42 from airports_data as ct, flights as fl, ticket_flights as tf
43 where ct.airport_code=fl.arrival_airport
44 and fl.flight_id=tf.flight_id
45 and date_trunc('year', fl.scheduled_departure ) = date_trunc('year', now())
46 group by ct.city
```

Data output Messages Notifications

	city	total_visits
	character varying (32)	bigint
1	Frankfurt	335
2	London	136
3	Athens	90
4	Rome	88
5	Madrid	79

Total rows: 5 of 5 Query complete 00:00:00.041 Ln 41, Col 1

pgAdmin 4

File Object Tools Help

Browser

Functions (1)  
timesnap()

Materialized Views

Operators

Procedures

Sequences (1)  
flights\_flight\_id\_seq

Tables (8)  
aircrafts\_data  
airports\_data  
boarding\_passes  
bookings  
flights  
seats  
ticket\_flights  
tickets

Trigger Functions

Types

Views (4)  
aircrafts  
airports  
flights\_view  
Columns (20)  
Rules  
Triggers  
routes\_view  
Columns (10)  
flight\_no  
departure\_airport  
departure\_airport\_name  
departure\_city

AirlineDatabase/postgres@PostgreSQL 14

Query Query History

```
41 select ct.city, count(tf.ticket_no) as total_visits
42 from airports_data as ct, flights as fl, ticket_flights as tf
43 where ct.airport_code=fl.arrival_airport
44 and fl.flight_id=tf.flight_id
45 and date_trunc('year', fl.scheduled_departure ) = date_trunc('year', now())
46 group by ct.city
47 order by total_visits desc
48 limit 5;
49
50
51
```

Data output Messages Notifications

	city	total_visits
	character varying (32)	bigint
1	Frankfurt	335
2	London	136
3	Athens	90
4	Rome	88
5	Madrid	79

Total rows: 5 of 5 Query complete 00:00:00.041 Ln 41, Col 1

Successfully run. Total query runtime: 41 msec. 5 rows affected.

f. Για να επιτευχθεί το ζητούμενο της ερώτησης αυτής έγραψα το εξής query (δίνεται παρακάτω)