

# Βάση Δεδομένων για Αεροδρόμιο

ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ (ECE\_ΓΚ703) - ΟΜΑΔΑ 20



ΕΥΘΥΜΙΟΣ, ΠΑΝΙΔΗΣ

[themispan2002@gmail.com](mailto:themispan2002@gmail.com), [up1084012@ac.upatras.gr](mailto:up1084012@ac.upatras.gr)

ΑΛΕΞΑΝΔΡΟΣ ΜΑΡΙΟΣ, ΤΣΑΚΙΡΙΔΗΣ

[alexandros.tsakiridis@gmail.com](mailto:alexandros.tsakiridis@gmail.com), [up1083879@ac.upatras.gr](mailto:up1083879@ac.upatras.gr)

ΛΕΞΕΙΣ - ΚΛΕΙΔΙΑ: database, sqlite, python, python311, aeronautics, airport, pysimplegui, flet, flutter

## ΠΕΡΙΕΧΟΜΕΝΑ

1	ΠΕΡΙΛΗΨΗ .....	3
2	ΜΕΘΟΔΟΛΟΓΙΑ .....	3
2.1	Entity Relationship Diagram .....	3
2.2	Σχήμα της βάσης .....	4
2.3	Περιεχόμενο Πινάκων .....	5
2.4	Στήλες Πινάκων.....	6
2.5	Scripts .....	7
3	ΑΞΙΟΛΟΓΗΣΗ.....	7
4	ΔΕΔΟΜΕΝΑ.....	8
5	ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ .....	9
6	ΠΑΡΑΔΕΙΓΜΑΤΑ.....	10
6.1	Employee View .....	10
6.2	Flight Occurrences.....	10
6.3	Schedule Creation.....	11
6.4	Airport Distances.....	11
6.5	Schedules Printing .....	12
6.6	Insert New Employee .....	12
6.7	Flights by Terminal .....	13
6.8	Airport Map.....	13
7	ΠΑΡΑΡΤΗΜΑ .....	14
8	ΑΠΟΘΕΤΗΡΙΟ ΚΩΔΙΚΑ.....	15
9	ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ .....	15

## 1 ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία αφορά τον σχεδιασμό βάσης δεδομένων που εξυπηρετεί της ανάγκες ενός αεροδρομίου. Η υλοποίηση έγινε με τη βάση δεδομένων SQLite και την Python 3. Περιλαμβάνονται δύο απλές γραφικές επαφές, υποδείγματα για το πώς θα μπορούσαν να υλοποιηθούν οι επόμενες, η μία επιτρέπει την εισαγωγή στοιχείων και η άλλη προβάλλει τα ήδη υπάρχοντα.

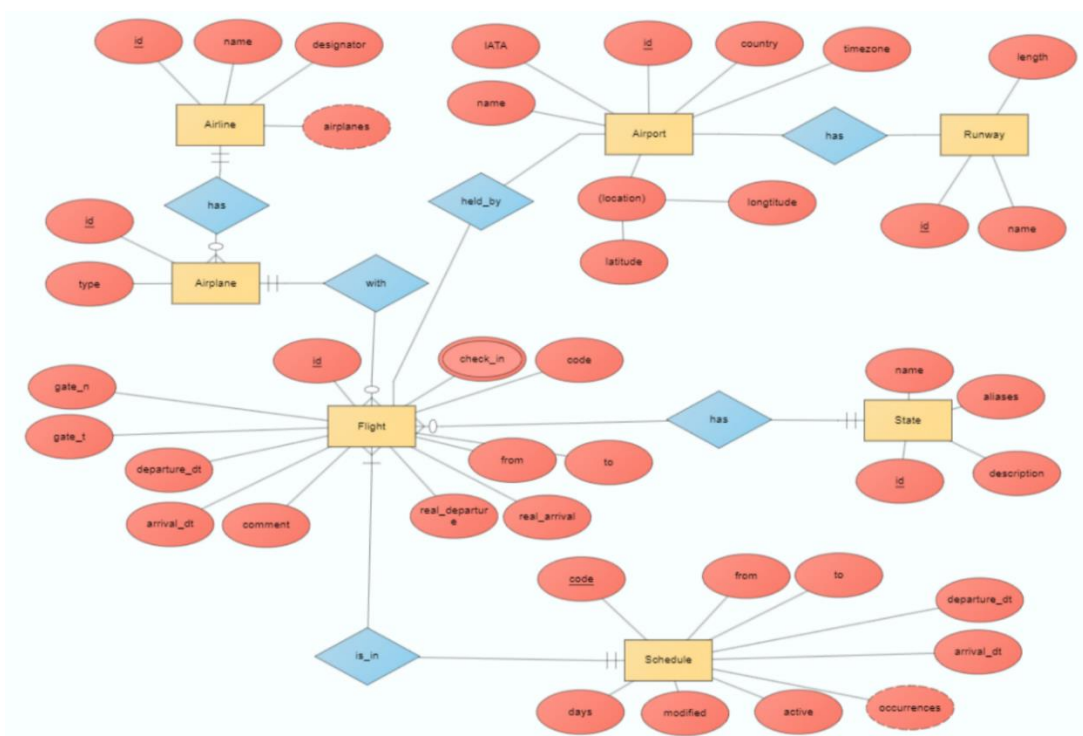
Η βάση αφορά το αεροδρόμιο αυτό καθ' αυτό και όχι τις αεροπορικές εταιρείες που διενεργούν πτήσεις σε αυτό. Επομένως, δεν περιλαμβάνει δεδομένα τα οποία διαχειρίζονται εταιρίες, όπως οι κρατήσεις των επιβατών, τα προσωπικά τους στοιχεία, οι τιμές των εισιτηρίων ή η πολιτική μεταφοράς αποσκευών για κάθε πτήση. Περιλαμβάνει δεδομένα που αφορούν τη διαχείριση του προγράμματος αφίξεων και αναχωρήσεων, την ενημέρωση και την κατεύθυνση των επιβατών για την εκάστοτε πτήση όσον αφορά την ώρα, την κατάσταση της πτήσης καθώς και το πού πρέπει να βρίσκονται οι επιβάτες προκειμένου να επιβιβαστούν στο αεροπλάνο. Τέλος, διαχειρίζεται δεδομένα που αφορούν τους εργαζόμενους και τα παραρτήματα στα οποία δουλεύουν.

## 2 ΜΕΘΟΔΟΛΟΓΙΑ

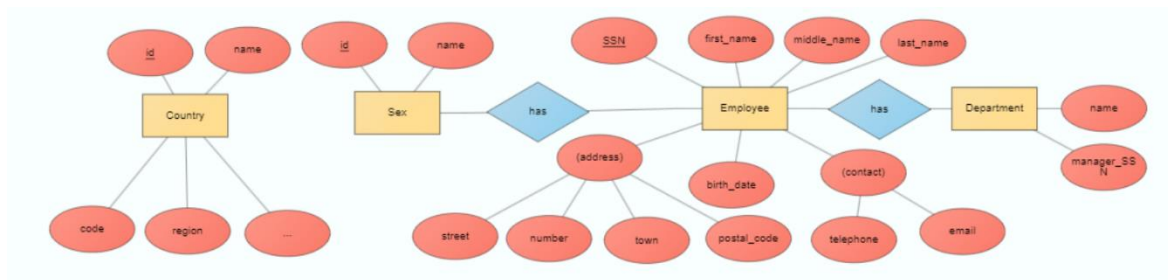
Το πρώτο βήμα για την υλοποίηση της βάσης ήταν να αποφασιστούν οι πίνακες που θα περιέχει και οι στήλες του καθενός. Στη συνέχεια, εισήχθησαν δεδομένα στους πίνακες και παράλληλα, γράφτηκε ο κώδικας Python. Το αεροδρόμιο που επιλέχτηκε είναι το Ελευθέριος Βενιζέλος της Αθήνας. Πληροφορίες για το πότε έγινε η κάθε εργασία υπάρχουν στην ενότητα του χρονοδιαγράμματος. Πληροφορίες για το τι κάνουν οι κλάσεις και οι μέθοδοι του κώδικα, υπάρχουν στα docstrings τους.

### 2.1 Entity Relationship Diagram

Ακολουθεί το διάγραμμα οντοτήτων – συσχετίσεων που δημιουργήθηκε με το online tool ERD Maker [9]:

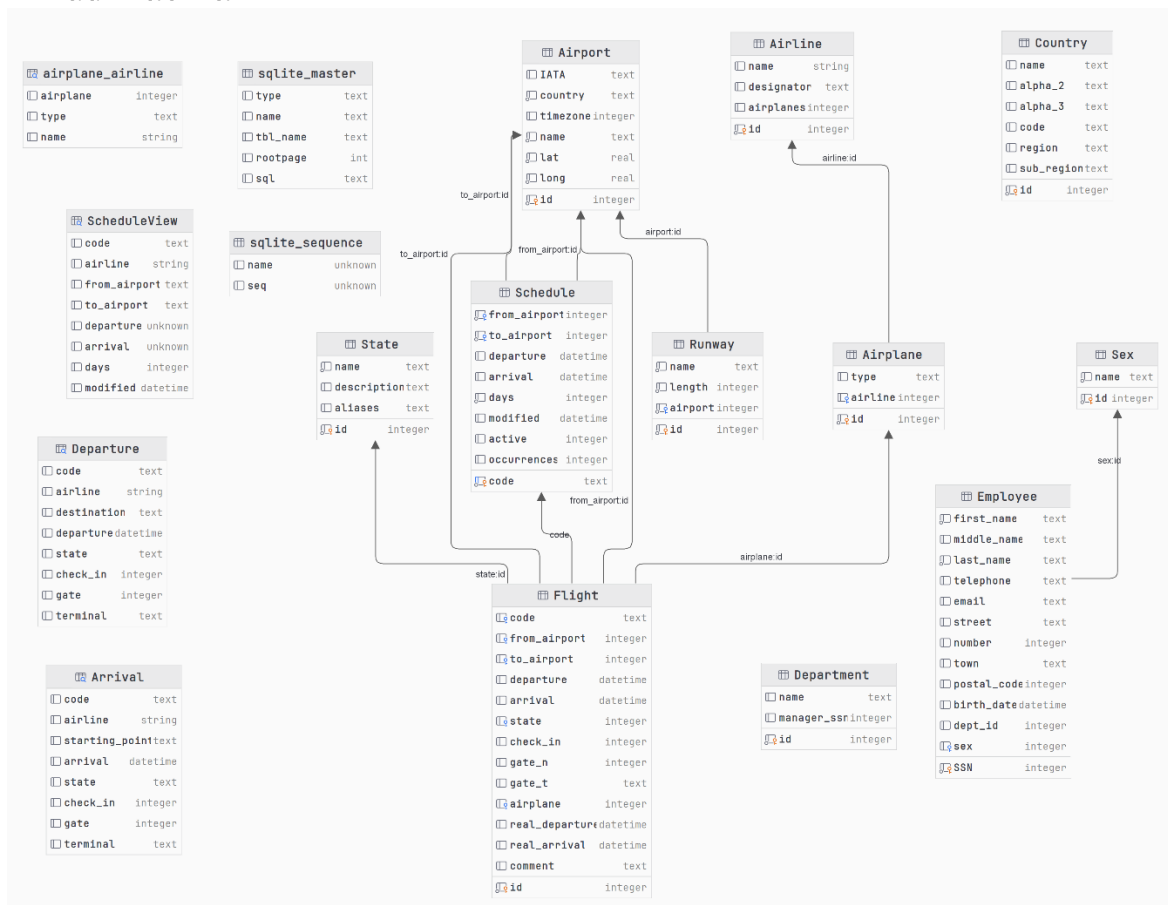


Εικόνα 1: Οι βασικές οντότητες της βάσης.



Εικόνα 2: Οι υπόλοιπες οντότητες της βάσης, σχετικές με τους εργαζόμενους.

## 2.2 Σχήμα της βάσης



Εικόνα 3: Σχήμα της βάσης φτιαγμένο με τη βιβλιοθήκη yFiles, περιλαμβάνει τους πίνακες, τα Views και τους πίνακες που δημιουργούνται αυτόματα από την SQLite, τον sqlite\_master για τη δομή των υπολοίπων και τον sqlite\_sequence για τα auto increments.

## 2.3 Περιεχόμενο Πινάκων

Ο πίνακας *Airline* περιλαμβάνει το όνομα της αεροπορικής εταιρείας και το designator της που αποτελείται από δύο *alphanumeric* χαρακτήρες.

Ο πίνακας *Airplane* περιλαμβάνει τον τύπο του αεροπλάνου και την αεροπορική εταιρεία στην οποία ανήκει. Δεν καταγράφονται δεδομένα όπως κινητήρες, αριθμός θέσεων, χωρητικότητα για αποσκευές, καθώς είναι θέματα που απασχολούν μόνο την εταιρεία.

Ο πίνακας *Airport* περιλαμβάνει τον κωδικό τριών χαρακτήρων που έχει δώσει η IATA στο αεροδρόμιο, τη χώρα στην οποία βρίσκεται, τη ζώνη ώρας, το πλήρες όνομά του και τις γεωγραφικές του συντεταγμένες σε δεκαδική μορφή.

Ο πίνακας *Country* περιέχει κάποιες πληροφορίες για την κάθε χώρα του πλανήτη, δηλαδή το όνομά της, έναν αριθμητικό κωδικό, κωδικό με δύο και τρεις χαρακτήρες, περιοχή και υποπεριοχή στην οποία ανήκει. Ο *Country* δημιουργήθηκε για να περιέχει χώρες των οποίων οι κωδικοί θα γράφονταν στον *Airport*, τελικά κάτι τέτοιο δεν έγινε και στον *Airport* γράφονται απευθείας τα ονόματα των χωρών.

Βασικός λόγος για τον οποίο δεν χρησιμοποιήθηκε τελικά ο *Country*, είναι ότι δεν περιλαμβάνει τις ζώνες ώρας, οι οποίες έτσι και αλλιώς μπορεί να μην είναι ίδιες σε όλες τις περιοχές μιας χώρας. Γι' αυτό, θα ήταν λάθος η ζώνη ώρας του αεροδρομίου να είναι η ζώνη ώρας της χώρας στην οποία ανήκει, καθώς αυτό δεν ισχύει πάντα.

Ο πίνακας *Department* περιλαμβάνει το όνομα του κάθε παραρτήματος του αεροδρομίου και μια στήλη για να αποθηκεύεται ο κωδικός του υπεύθυνου εργαζομένου (*manager*).

Ο πίνακας *Employee* αποθηκεύει τα στοιχεία των εργαζομένων, ΑΦΜ, κύριο και μεσαίο όνομα, επώνυμο, τηλέφωνο επικοινωνίας, διεύθυνση email, διεύθυνση που αποτελείται από οδό, αριθμό, πόλη κατοικίας και ταχυδρομικό κώδικα, ημερομηνία γέννησης, κωδικό παραρτήματος στο οποίο εργάζεται και φύλο.

Ο πίνακας *Flight* περιέχει δεδομένα για μία πτήση που θα γίνει συγκεκριμένη μέρα, ώρα, με προκαθορισμένη αφετηρία και προορισμό, αεροπλάνο, όπως και διάρκεια. Όλες οι πτήσεις που γίνονται βάσει της ίδιας προγραμματισμένης πτήσης (*Scheduled Flight*), θεωρούμε πως μοιράζονται τον ίδιο κωδικό πτήσης, ο οποίος περιέχει πληροφορία και για την εταιρεία που τη διενεργεί.

Ο πίνακας *Runway* περιλαμβάνει πληροφορίες για έναν διάδρομο αεροδρομίου, το μήκος του, το όνομά του που δείχνει προσανατολισμό σε μοίρες και τέλος, το αεροδρόμιο στο οποίο βρίσκεται [7].

Ο πίνακας *Schedule* περιέχει πληροφορίες για τις πτήσεις που είναι προγραμματισμένες να γίνονται κάθε εβδομάδα, τις ίδιες μέρες και την ίδια ώρα, κωδικό πτήσης, αεροδρόμιο αφετηρίας και προορισμού, ώρα αναχώρησης, ώρα άφιξης, τις μέρες που θα γίνεται η πτήση, ημερομηνία και ώρα τελευταίας τροποποίησης και ένδειξη αν η εγγραφή είναι ενεργή. Στο πεδίο *occurrences* καταμετράται τακτικά και καταγράφεται ο αριθμός των εγγεγραμμένων πτήσεων στον πίνακα *Flight* με τον κωδικό της προγραμματισμένης.

Για παράδειγμα, μια προγραμματισμένη πτήση από την Αθήνα στη Θεσσαλονίκη που γίνεται όλο τον χρόνο στις 9 το πρωί σε ώρα *Greenwich*, δηλαδή στις 7 το πρωί σε τοπική ώρα τους χειμερινούς μήνες, από Δευτέρα έως Παρασκευή. Κάθε οντότητα *Schedule* έχει έναν κωδικό πτήσης.

Ο πίνακας *Sex* χρησιμοποιείται για να προσδιορίζεται το φύλο των εργαζομένων: αρσενικό, θηλυκό, άλλο.

Ο πίνακας *State* περιλαμβάνει όνομα και περιγραφή για την κάθε κατάσταση που μπορεί να βρίσκεται κάποια εγγραφή του πίνακα *Flight*.

## 2.4 Στήλες Πινάκων

Πίνακας 1: Πίνακες της βάσης

Table Name	Table Columns
Airline	integer <b>id</b> , text <b>name</b> , text <b>designator</b> , integer <b>airplanes</b>
Airplane	integer <b>id</b> , text <b>type</b> , integer <b>airline</b>
Airport	integer <b>id</b> , text <b>IATA</b> , text <b>country</b> , integer <b>timezone</b> , text <b>name</b> , real <b>lat</b> , real <b>long</b>
Country	integer <b>id</b> , text <b>name</b> , text <b>alpha_2</b> , text <b>alpha_3</b> , text <b>code</b> , text <b>region</b> , text <b>sub_region</b>
Department	integer <b>id</b> , text <b>name</b> , integer <b>manager_ssn</b>
Employee	integer <b>SSN</b> , text <b>first_name</b> , text <b>middle_name</b> , text <b>last_name</b> , text <b>telephone</b> , text <b>email</b> , text <b>street</b> , text <b>number</b> , text <b>town</b> , text <b>postal_code</b> , datetime <b>birth_date</b> , integer <b>dept_id</b> , integer <b>sex</b>
Flight	integer <b>id</b> , text <b>code</b> , integer <b>from_airport</b> , integer <b>to_airport</b> , datetime <b>departure</b> , datetime <b>arrival</b> , integer <b>state</b> , integer <b>check_in</b> , integer <b>gate_n</b> , text <b>gate_t</b> , integer <b>airplane</b> , datetime <b>real_departure</b> , datetime <b>real_arrival</b> , text <b>comment</b>
Runway	integer <b>id</b> , text <b>name</b> , integer <b>length</b> , integer <b>airport</b>
Schedule	text <b>code</b> , integer <b>from_airport</b> , integer <b>to_airport</b> , datetime <b>departure</b> , datetime <b>arrival</b> , integer <b>days</b> , datetime <b>modified</b> , integer <b>active</b> , integer <b>occurrences</b>
Sex	integer <b>id</b> , text <b>name</b>
State	integer <b>id</b> , text <b>name</b> , text <b>description</b> , text <b>aliases</b>

Στην βάση υλοποιήθηκαν μερικά Views, για queries που χρειάστηκε να γίνονται συχνά.

Το *AirplaneAirline* με join των δύο πινάκων τυπώνει όλα τα αεροπλάνα μαζί με το όνομα της αεροπορικής εταιρείας στην οποία ανήκουν.

Το *Arrival* φιλτράρει των πίνακα *Flight* και τυπώνει μόνο όσες πτήσεις αποτελούν άφιξη.

Το *Departure* λειτουργεί ομοίως.

Το *EmployeeView* τυπώνει όλες τις εγγραφές του *Employee*, αντικαθιστώντας τα **sex** και τα **dept\_id** με τα κανονικά ονόματά τους.

Τέλος, το *ScheduleView* τυπώνει τις ενεργές εγγραφές του *Schedule*, αντικαθιστώντας τα ids με όνομα αεροπορικής και κωδικούς IATA αεροδρομίων.

Στην κλάση Database υλοποιήθηκαν ενδεικτικά κάποιες μέθοδοι για *Create, read, update and delete* tasks.

Η μέθοδος *schedules()* εκτελεί ένα query για όλες τις εγγραφές του πίνακα *Schedule* με δυο join που αντικαθιστούν τα ids των αεροδρομίων με τους κωδικούς IATA και το πεδίο *days* με τα ονόματα των ημερών αντί για την κωδικοποίηση. Η συγκεκριμένη εργασία δεν θα μπορούσε να γίνει με view της βάσης γιατί η SQLite φαίνεται πως δεν δίνει τη δυνατότητα μετατροπής δεκαδικού σε δυαδικό καθώς και την υπόλοιπη διαδικασία του εκτελεί η Python για να μετατρέψει τους δεκαδικούς σε μέρη στην υλοποίησή μας.

## 2.5 Scripts

Το *add\_schedule.py* χρησιμοποιήθηκε για τη δημιουργία τυχαίων εγγραφών *Schedule*, με επιλογή αεροπορικής εταιρείας και κάποιου αεροδρομίου. Η μέθοδος *random\_schedule* αποφασίζει τυχαία αν η προγραμματισμένη πτήση θα είναι αναχώρηση ή άφιξη, διαλέγει μια τυχαία ώρα άφιξης ή αναχώρησης και τέλος επιλέγει τυχαία κάποιες μέρες της εβδομάδας για να διενεργείται η πτήση.

Το *airplane\_insert.py* είναι παράδειγμα script που χρησιμοποιήθηκε για την εισαγωγή δεδομένων από αρχείο csv, εν προκειμένω για τον πίνακα *Airplane*.

Το *country\_insert.py* κάνει αντίστοιχη διαδικασία, αυτή τη φορά για τον πίνακα *Country*.

**ΣΗΜΕΙΩΣΗ:** Αντί για τη χρήση του command *.import* της SQLite για τα csv αρχεία, προτιμήθηκε η εισαγωγή των δεδομένων να γίνει χειροκίνητα για να υπάρχει δυνατότητα επιλογής μόνο των επιθυμητών στηλών.

Το *distances.py* ανακτά από τον *Airport* όλα τα αεροδρόμια του εξωτερικού, τυπώνει όλες τις πληροφορίες τους και μετρά την απόστασή τους από το Ελ. Βενιζέλος σε γεωγραφικές μοίρες, χιλιόμετρα και τέλος, την εκτιμώμενη διάρκεια μιας πτήσης από το Ελ. Βενιζέλος προς αυτά. Για τον υπολογισμό της γωνίας πάνω στη Γη, χρησιμοποιήθηκε το Haversine formula [8]. Η απόσταση σε χιλιόμετρα υπολογίστηκε πολλαπλασιάζοντας τη γωνία με την ακτίνα της Γης, η οποία δεν είναι σταθερή, οπότε κάθε φορά υπολογιζόταν συναρτήσει των γεωγραφικών πλατών των δύο αεροδρομίων. Η χρονική διάρκεια υπολογίστηκε χρησιμοποιώντας μια σταθερή ταχύτητα πτήσης, κάτι που φυσικά δεν είναι καθόλου ακριβές για πτήσεις μικρής απόστασης.

Το *draw.py* ζωγραφίζει έναν απλό χάρτη που δείχνει ποιοτικά τις θέσεις δύο αεροδρομίων σε σχέση με το σημείο 0, 0 της Γης, δηλαδή την τομή Ισημερινού και Πρώτου Μεσημβρινού. Για να γίνει αυτό, χρησιμοποιείται η μέθοδος *map* της κλάσης *Airport*, η οποία παίρνει σαν παράμετρο ένα δεύτερο τέτοιο αντικείμενο και φτιάχνει τον χάρτη μεταξύ τους. Πάνω από τον χάρτη, τυπώνονται οι πληροφορίες των δύο αεροδρομίων για ευκολότερη εξαγωγή συμπερασμάτων.

Το *employee\_generator.py* καλεί την συνάρτηση *random\_employee()* της κλάσης *Database* η οποία δημιουργεί τυχαίους εργαζόμενους για το αεροδρόμιο. Συγκεκριμένα καλεί την συνάρτηση *load\_files()* της κλάσης *Employee* του *models.py* η οποία φορτώνει κάποια αρχεία txt με αγγλικά ονόματα, επίθετα και ελληνικές πόλεις, επιλέγει τυχαία κάποια δεδομένα και επιστρέφει ένα tuple με τυχαίους υπαλλήλους και τα αντίστοιχα στοιχεία τους.

Έχουν γίνει δύο υλοποιήσεις γραφικής διεπαφής με περιορισμένη λειτουργικότητα

Ο χρήστης έχει τη δυνατότητα να προσθέσει δεδομένα στον πίνακα κάνοντας νέα εγγραφή εργαζομένου από το script *GUI\_insert\_Employees.py*, το οποίο χρησιμοποιεί την εξωτερική βιβλιοθήκη *PySimpleGUI*.

Το *flet\_flights.py* χρησιμοποιεί τη βιβλιοθήκη *flet* (Flutter for Python) για την προβολή των αφίξεων και των αναχωρήσεων σε καρτέλα browser, κάτι που θα μπορούσε να επεκταθεί σε mobile application. Με την χρήση των κουμπιών στην πάνω δεξιά γωνία του παραθύρου, μπορεί να γίνεται εναλλαγή ανάμεσα στα Terminals του αεροδρομίου ή ανάμεσα σε Αναχωρήσεις και Αφίξεις.

## 3 ΑΞΙΟΛΟΓΗΣΗ

Η εφαρμογή στην τρέχουσα κατάστασή της παραμένει ημιτελής, για την ώρα μπορεί να αποθηκεύσει και να διαχειριστεί διάφορες πληροφορίες που χρειάζεται ένα αεροδρόμιο όσον αφορά τις πτήσεις που διενεργούνται, χωρίς να επεκτείνεται σε λογιστικά ζητήματα, όπως δεδομένα που αφορούν μισθοδοσία, ασφαλίσεις ή άδειες των εργαζομένων.

Ένα πιθανό επόμενο στάδιο της υλοποίησης θα ήταν η κατασκευή φιλικής προς τον χρήστη γραφικής διεπαφής για την προβολή όλων των πινάκων μέσω παραθύρων που θα επέτρεπαν την επεξεργασία των υφιστάμενων εγγραφών ή την προσθήκη νέων.

Για αυτό, ενδεικτικά υλοποιήθηκε το παράθυρο εισαγωγής νέου εργαζομένου με την PySimpleGUI [3]. Το πρόγραμμα, φυσικά, ελέγχει τα στοιχεία που θα εισαχθούν, για παράδειγμα την εγκυρότητα της διεύθυνσης email και του τηλεφώνου ή του ΑΦΜ.

Η υλοποίηση στο Flutter [4] που προβάλλει σε καρτέλα φυλλομετρητή τον πίνακα αφίξεων και αναχωρήσεων θα μπορούσε να είναι τμήμα της ιστοσελίδας στην οποία οι επιβάτες θα μπορούσαν να συνδέονται και να παίρνουν όλες τις απαραίτητες πληροφορίες για να επιβιβαστούν στην πτήση τους ή να δουν την εκτιμώμενη ώρα άφιξης κάποιας πτήσης που περιμένουν. Ένα άλλο βήμα θα ήταν η ανάπτυξη mobile εφαρμογής που θα παρουσίαζε τα ίδια δεδομένα, κάτι τέτοιο όμως θα απαιτούσε υπηρεσία hosting και πιθανότατα κάποια άλλη βάση δεδομένων, καταλληλότερη από την SQLite για τέτοιο σκοπό.

Μια άλλη λειτουργία θα μπορούσε να είναι η χρήση εξωτερικής βιβλιοθήκης για την εξαγωγή σε αρχείο pdf ημερήσιων αναφορών για παράδειγμα, τμημάτων του πίνακα των εργαζομένων ή των πτήσεων που διενεργήθηκαν καθυστερημένα. Περαιτέρω βελτίωση της εφαρμογής θα περιλάμβανε και περισσότερα δεδομένα, λόγου χάριν όλα τα αεροδρόμια της Ευρώπης και όλους τους διαδρόμους του καθενός, σε συνδυασμό με real-time δεδομένα εναέριας κυκλοφορίας για τους αεροδιαδρόμους που χρησιμοποιούν τα αεροπλάνα στα ταξίδια τους.

Ταυτόχρονα, η ύπαρξη περισσότερων εγγραφών στον πίνακα *Schedule* αυτομάτως θα σήμαινε και περισσότερες εγγραφές στον *Flight*, κάτι που θα βελτίωνε την εικόνα των πτήσεων, που για την ώρα είναι συνήθως λιγότερες από πέντε την ημέρα. Για να είναι πιο αληθοφανείς οι πίνακες, θα έπρεπε να έχουν δημιουργηθεί πτήσεις και για τον μήνα Ιανουάριο, καθώς τώρα οι πρώτες πτήσεις στον πίνακα είναι για την 1<sup>η</sup> Φεβρουαρίου, οπότε δεν μπορούν να φανούν τα διαφορετικά States που μπορεί να πάρει μια πτήση.

Ο υπολογισμός της διάρκειας πτήσης μεταξύ δύο αεροδρομίων θα μπορούσε να γίνει ακριβέστερος με την αξιοποίηση μετεωρολογικών δεδομένων, διευθύνσεων διαδρόμων και την συμπερίληψη στον πίνακα των αεροπλάνων στοιχείων για τις ταχύτητες που πετάει το κάθε μοντέλο αεροπλάνου.

Όπως και να 'χει, η σωστή παράδοση της εφαρμογής θα ήταν σε εκτελέσιμο αρχείο (για παράδειγμα exe) το οποίο θα εγκαθίσταται σε κάθε συσκευή χωρίς αρχείο της βάσης δεδομένων, καθώς αυτή δεν μπορεί να είναι διαφορετική για κάθε εγκατάσταση. Και πάλι, κάτι τέτοιο θα απαιτούσε διαφορετική βάση δεδομένων, κάποια υπηρεσία hosting και σύστημα διαχείρισης χρηστών. Ο διαχειριστής θα έχει δικαιώματα επεξεργασίας, διαφορετικά από αυτά ενός απλού χρήστη που απλά θα συνδέεται στην ιστοσελίδα ή θα χρησιμοποιεί την εφαρμογή στο κινητό του για να βλέπει τις επόμενες πτήσεις.

#### 4 ΔΕΔΟΜΕΝΑ

Τα δεδομένα της βάσης εισήχθησαν είτε με script που δημιούργησε τυχαία δεδομένα, είτε από αρχεία csv. Συγκεκριμένα για τον κάθε πίνακα έγιναν τα εξής:

Για τον πίνακα *Airline* αναζητήθηκαν κάποιες ενδεικτικές αεροπορικές εταιρείες με τις οποίες συνεργάζεται το “Ελευθέριος Βενιζέλος”, αποθηκεύτηκαν σε ένα αρχείο .csv και τις περάσαμε στη βάση.

Για τον πίνακα *Airplane* αναζητήθηκε ο στόλος της κάθε αεροπορικής εταιρείας στο site της, καταγράφηκαν τα μοντέλα σε csv και δημιουργήθηκαν τυχαία αεροπλάνα.

Για τον πίνακα *Airport*, για τα αεροδρόμια του εσωτερικού χρησιμοποιήθηκε λίστα της Wikipedia [6] με τα αεροδρόμια της Ελλάδας για το πλήρες όνομα και τον κωδικό IATA τους, στη συνέχεια βρέθηκαν από το Google Earth οι συντεταγμένες τους. Ενδεικτικά, περάστηκαν και κάποια επιλεγμένα αεροδρόμια του εξωτερικού.

Για τον πίνακα *Country* τα δεδομένα εισήχθησαν από ελεύθερο αρχείο csv που βρέθηκε στο διαδίκτυο.

Για τον πίνακα *Department* βρέθηκαν τα ονόματα παραρτημάτων που υπάρχουν σε ένα τυπικό αεροδρόμιο. Χρησιμοποιήθηκαν τα παραρτήματα που λειτουργούν στο Billings-Logan International Airport [5] μιας και για το Ελευθέριος Βενιζέλος δεν βρέθηκε κάποια επίσημη πληροφορία.



Για τον πίνακα *Employee* υπάρχει το script της Python *employee\_generator.py* το οποίο δημιούργησε τυχαίο όνομα, επίθετο, τηλέφωνο, email, ημερομηνία γέννησης και στοιχεία διεύθυνσης για τον κάθε εργαζόμενο.

Στο πίνακα *Flight* τα δεδομένα εισήχθησαν αυτόματα, με χρήση μιας μεθόδου. Η μέθοδος αυτή λέγεται *generate\_scheduled\_flights*, δέχεται ως παράμετρο έναν κωδικό πτήσης που υπάρχει στον πίνακα *Schedule* και για καθορισμένο διάστημα ημερομηνιών προσθέτει μια εγγραφή *Flight* με τον κωδικό αυτό στις μέρες που πρέπει. Το διάστημα ημερομηνιών καθορίζεται σε δύο *datetime.date* σταθερές της κλάσης *Database*, την *SCHEDULE\_START* και την *SCHEDULE\_END*, οι οποίες υποδεικνύουν τη δημιουργία πτήσεων από την 1<sup>η</sup> Φεβρουαρίου μέχρι τις 30 Απριλίου. Η εταιρεία που διενεργεί τις πτήσεις και η ώρα τους θα είναι πάντα η ίδια, αυτό που θα αλλάζει θα είναι η ημερομηνία, το αεροπλάνο που χρησιμοποιείται και οι πληροφορίες για το gate.

Στον πίνακα *Runway* εισήχθησαν οι δύο διάδρομοι του Ελ. Βενιζέλος, όπως έχουν καταγραφεί στη Wikipedia [7].

Για τον πίνακα *Schedule* τα δεδομένα εισήχθησαν με την μέθοδο *random\_schedule()* της κλάσης *Database* η οποία χρησιμοποιώντας ένα designator αεροπορικής και ένα airport id, δημιουργεί μια τυχαία εγγραφή, επιλέγοντας αν θα πρόκειται για άφιξη ή αναχώρηση, σίγουρα όμως θα είναι κάποια πτήση ανάμεσα στο δοσμένο αεροδρόμιο και το Ελ. Βενιζέλος. Η εγγραφή έχει ημερομηνία άφιξης ή αναχώρησης, προφανώς το ένα από τα δύο πεδία θα είναι null, αυτό εξαρτάται από το αν το αεροδρόμιο προέλευσης της Αθήνας βρίσκεται στο πεδίο *from\_airport* ή *to\_airport*. Αυτή η ημερομηνία ορίζεται στη μεταβλητή της κλάσης *MONTH*, η ώρα δημιουργείται τυχαία όπως και οι μέρες για τις οποίες εκτελείται η εκάστοτε πτήση.

Οι πίνακες *Sex* και *State* δεν είχαν ανάγκη για πολλά δεδομένα, οπότε συμπληρώθηκαν χειροκίνητα.

## 5 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Οκτώβριος 2023: Δημιουργία βασικών κλάσεων για τη λειτουργία της εφαρμογής, συζήτηση για τη μορφή της βάσης. Η βάση SQLite επιλέχθηκε για πρακτικούς λόγους, δεν υπάρχει ανάγκη για server, αποτελείται από ένα αρχείο και είναι ευκολότερα διαχειρίσιμη).

Χωρισμός του κώδικα σε δύο python packages, το assets με modules για τις λειτουργίες της εφαρμογής και την διαχείριση της βάσης και το scripts με scripts που ελέγχουν τη λειτουργία των κλάσεων στο assets.

Νοέμβριος 2023: Σταδιακή εισαγωγή δεδομένων στους πίνακες της βάσης με τους τρόπους που περιγράφηκαν παραπάνω. Μετά την ενδιάμεση παρουσίαση και την ανατροφοδότηση για τη δομή της βάσης, αφαίρεση των περιττών πινάκων, βελτιώσεις και αλλαγές.

Υλοποίηση διαφόρων λειτουργιών στην κλάση *Coordinates*, δημιουργία *Enumerations* και κλάσεων που αναπαριστούν τα δεδομένα των πινάκων της κλάσης. Αντικατάσταση των assert statements με περιγραφικά *Exceptions*.

Δεκέμβριος 2023: Τροποποιήσεις στις κλάσεις του assets. Χρήση του Git για Version Control System και αποθετηρίου στο GitHub για την παράλληλη εργασία στο project. Επέκταση της κλάσης *Database* με πολλές μεθόδους για αναζητήσεις στη βάση και μηχανισμούς για ανίχνευση λογικών σφαλμάτων. Διορθώσεις και συμπλήρωση των docstrings στα assets. Δημιουργία γραφικής διεπαφής για εισαγωγή νέου εργαζομένου στη βάση, με πληκτρολόγηση των προσωπικών του στοιχείων σε φόρμα.

Ιανουάριος 2024: Προσθήκη μιας δεύτερης γραφικής διεπαφής για την προβολή των αφίξεων ή των αναχωρήσεων, ξεχωριστά για τον κάθε τερματικό σταθμό. Συγγραφή της αναφοράς, δημιουργία της παρουσίασης.

## 6 ΠΑΡΑΔΕΙΓΜΑΤΑ

Ακολουθούν μερικά παραδείγματα και screenshots από τη χρήση της εφαρμογής. Περισσότερες πληροφορίες στο αποθετήριο του [GitHub](#).

### 6.1 Employee View

Το View αυτό προβάλλει τα στοιχεία των εργαζομένων, κάνοντας join τους κατάλληλους πίνακες.

ID	SSN	first	middle	last	sex	telephone	email	street	number	town	birth_date	department
1	121943494	Britta	Millery	Hudson	male	+30 694 387 7815	britt.huds3@aia.gr	Zaimi	69	Megara	1900-01-10	Aircraft Rescue Fire Fighting (ARFF)
2	133569548	Terron	Ioa	Rosales	male	+30 694 932 9766	te-rosal275@gmail.com	Othonos - Amalias	13	Xanthi	1988-05-25	Aircraft Rescue Fire Fighting (ARFF)
3	144993897	Jonerik	Jermario	Contreras	male	+30 694 568 6701	joneri-contrer@proton.me	Kanakari	31	Ierissos	1938-04-23	Administration
4	147986492	Sarde	Antonio	Henderson	male	+30 694 029 5005	color-sard-he@aol.com	Zaimi	18	Rythira	1991-08-11	Airport Police
5	153646331	Kayna	Trenton	Glenn	male	+30 694 436 8032	kayn.gle1979@yahoo.com	Ellinos Stratistou	41	Sparti	1979-05-21	Airport Police
6	155644455	Karlisha	Cleve	Velazquez	other	+30 694 973 5412	karli.velazq1993@yahoo.com	Alexandrou Ipsilantou	60	Nafpaktos	1993-01-04	Building Maintenance
7	163643264	Ellen	Keagan	Mueller	female	+30 694 326 2808	commonell-mu@hotmail.com	Egnatias	47	Nafplio	1975-02-13	Aircraft Rescue Fire Fighting (ARFF)
8	165987329	Onayra	Laporscha	Maldonado	other	+30 694 909 8422	onay_maldo1961@outlook.com	Agiou Nikolaou	50	Alexandria	1981-05-28	Airport Police
9	169425837	Adlizi	Agal	Shields	female	+30 694 863 8959	move.adl.shiel@aol.com	Othonos - Amalias	12	Ermooupolis	1937-06-19	Aircraft Rescue Fire Fighting (ARFF)
10	172633798	Lateasha	Maribeth	Woodward	female	+30 694 559 5522	lat_woodwar1968@proton.me	Serokostopoulou	72	Marousi	1968-09-11	Aircraft Rescue Fire Fighting (ARFF)
11	173271122	Karlie	Lorina	Yoder	female	+30 694 245 6073	karli_yo1962@proton.me	Penepistimiou	68	Larissa	1962-12-02	Airport Police
12	178666288	Charles	Shyann	McClain	male	+30 694 858 6797	chrle-mcc1990@proton.me	Maizonos	47	Rhodes	1990-12-22	Airport Police
13	178842326	Lula	Jaron	Bishop	female	+30 694 388 9376	particularlul-bi@yahoo.com	Ellinos Stratistou	18	Gytheio	1969-11-11	Administration
14	187261371	Linwood	Susi	Lloyd	male	+30 694 527 9329	linw-lloy1977@gmail.com	Zaimi	3	Gelaxidi	1977-04-26	Aircraft Rescue Fire Fighting (ARFF)
15	187995873	Trenecia	Charlina	Stone	female	+30 694 683 2762	treneci-ston1969@gmail.com	Akti Dimeon	61	Malia	1969-03-06	Building Maintenance
16	188654537	Dawnmarie	Askia	Durham	female	+30 694 907 1929	dawnmari-durh1967@gmail.com	Maizonos	59	Kastoria	1969-03-13	Building Maintenance
17	211949214	Jean	Riquel	Lin	other	+30 694 278 2532	seed.jea.1@aia.gr	Kalavriton	32	Nafplio	1986-10-27	Airfield Maintenance / Operations
18	212866916	Joshua	Bashan	Cox	female	+30 694 612 6170	winter-jos-cox@proton.me	Papaflessa	52	Zahara	1963-11-11	Building Maintenance
19	215271394	Lakiya	Willena	Hurley	male	+30 694 926 7857	strong_laki_hur@yahoo.com	Egnatias	53	Livadia	1967-06-11	Airport Police
20	215728355	Alvie	Elva	Coleman	female	+30 694 286 4529	stone_alv_cole@yahoo.com	Astiggos	23	Koropi	1956-10-27	Aircraft Rescue Fire Fighting (ARFF)
21	221355556	Jeanine	Faron	Benton	male	+30 694 733 9435	jeanbento1993@outlook.com	Lenessou	75	Aigio	1993-10-19	Airfield Maintenance / Operations
22	223233804	Alberto	Laddie	Meadows	male	+30 694 026 3508	albeseador1968@proton.me	Papaflessa	26	Komotini	1966-05-17	Aircraft Rescue Fire Fighting (ARFF)
23	224853418	Antonnette	Mariano	Hanna	female	+30 694 099 3945	requireanhan@yahoo.com	Kalavriton	42	Distomo	1985-04-13	Airfield Maintenance / Operations
24	235872997	Yahayra	Adina	Garner	male	+30 694 065 6467	yaha.gai1988@aia.gr	Zaimi	5	Karpenisi	1988-05-09	Administration
25	239178209	Shareena	Courtenay	Mullen	female	+30 694 583 0451	sharenmull11964@aia.gr	Korinthou	23	Vytina	1964-10-28	Administration
26	242617249	Zakiyah	Keleigh	Simon	male	+30 694 176 2926	exact.za-si@aol.com	Kalavriton	62	Krestena	1961-07-27	Administration
27	244735622	Maura	Starlette	Schwartz	female	+30 694 266 0767	mau_schwa153@aia.gr	Eleutheriou Venizelou	70	Megara	1968-02-17	Aircraft Rescue Fire Fighting (ARFF)
28	246286157	Rina	Joette	Marshall	female	+30 694 343 3687	gune.rimarsh@proton.me	Ellinos Stratistou	32	Koropi	1984-09-26	Aircraft Rescue Fire Fighting (ARFF)

### 6.2 Flight Occurrences

Παρακάτω φαίνεται το μήνυμα στο τέλος της μεθόδου `update_schedule_occurrences`, η οποία για τον κωδικό πτήσης κάθε ενεργής εγγραφής προγραμματισμένης πτήσης, μετρά στον πίνακα *Flight* εγγραφές με τον ίδιο κωδικό και ενημερώνει το υπολογιζόμενο πεδίο *occurrences* της συγκεκριμένης εγγραφής.

```
FR-348
FR-589
FR-418
FR-849
2067 NEW FLIGHT OCCURRENCES DETECTED.
121 QUERIES EXECUTED          CHANGES_COMMITTED          DATABASE_CLOSED

Process finished with exit code 0
```

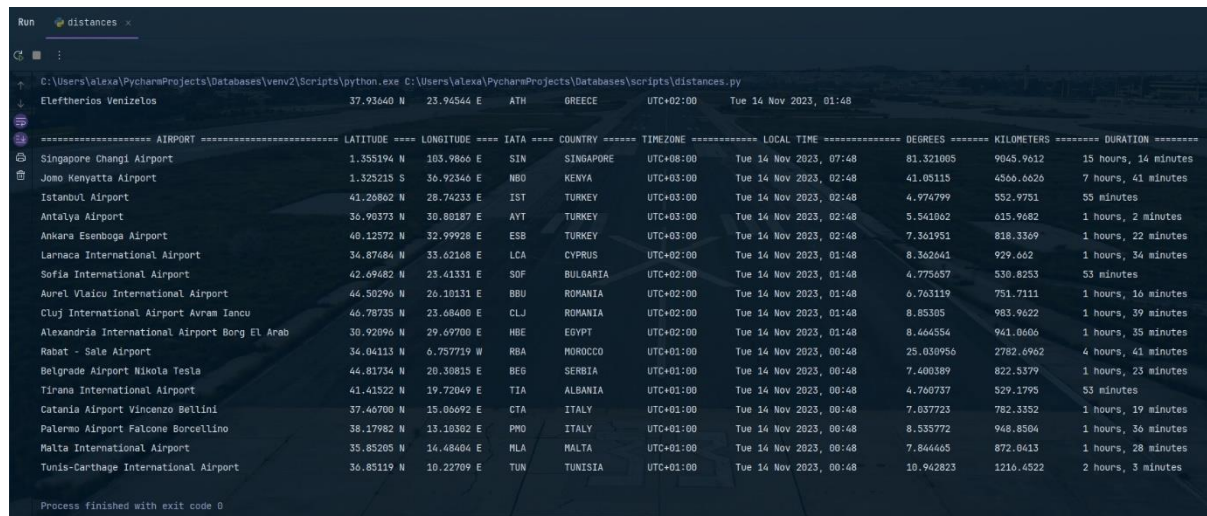
### 6.3 Schedule Creation

Εδώ φαίνονται όσα τυπώνονται κατά τη διάρκεια δημιουργίας νέων εγγραφών στον πίνακα *Schedule*. Αν το αντικείμενο της κλάσης Database δημιουργηθεί με το setting *DEBUG*, στο τέλος του script τυπώνεται ο αριθμός των queries που πραγματοποιήθηκαν. Όπως πάντα, το αντικείμενο κάνει από μόνο του commit τις αλλαγές στη βάση και κλείνει το connection.

```
['FR-849', 0, 48, datetime.datetime(2024, 4, 20, 13, 40), None, 36, 18, 'C', 2485]
2024-04-23 Day.TUESDAY True
['FR-849', 0, 48, datetime.datetime(2024, 4, 23, 13, 40), None, 37, 26, 'C', 270]
2024-04-25 Day.THURSDAY True
['FR-849', 0, 48, datetime.datetime(2024, 4, 25, 13, 40), None, 11, 9, 'B', 2527]
2024-04-27 Day.SATURDAY True
['FR-849', 0, 48, datetime.datetime(2024, 4, 27, 13, 40), None, 0, 17, 'B', 350]
2024-04-30 Day.TUESDAY True
['FR-849', 0, 48, datetime.datetime(2024, 4, 30, 13, 40), None, 22, 18, 'A', 639]
39 SCHEDULED FLIGHTS CREATED.
146 QUERIES EXECUTED      CHANGES_COMMITTED      DATABASE_CLOSED
```

### 6.4 Airport Distances

Το παρακάτω screenshot είναι από την εκτέλεση του *distances.py*, κατά τη διάρκεια του οποίου υπολογίζεται η απόσταση διαφόρων προορισμών από ένα αεροδρόμιο αναφοράς και μια εκτιμώμενη διάρκεια πτήσης για το καθένα [8].



AIRPORT	LATITUDE	LONGITUDE	IATA	COUNTRY	TIMEZONE	LOCAL TIME	DEGREES	KILOMETERS	DURATION
Singapore Changi Airport	1.355194 N	103.9866 E	SIN	SINGAPORE	UTC+08:00	Tue 14 Nov 2023, 07:48	81.321005	9045.9612	15 hours, 14 minutes
Jomo Kenyatta Airport	1.525215 S	36.92346 E	NBO	KENYA	UTC+03:00	Tue 14 Nov 2023, 02:48	41.05115	4566.6626	7 hours, 41 minutes
Istanbul Airport	41.28862 N	28.74233 E	IST	TURKEY	UTC+03:00	Tue 14 Nov 2023, 02:48	4.974799	552.9751	55 minutes
Antalya Airport	36.98373 N	30.88187 E	AYT	TURKEY	UTC+03:00	Tue 14 Nov 2023, 02:48	5.541862	615.9682	1 hours, 2 minutes
Ankara Esenboga Airport	40.12572 N	32.99928 E	ESB	TURKEY	UTC+03:00	Tue 14 Nov 2023, 02:48	7.361951	818.3369	1 hours, 22 minutes
Larnaca International Airport	34.87484 N	33.62168 E	LCA	CYPRUS	UTC+02:00	Tue 14 Nov 2023, 01:48	8.362841	929.662	1 hours, 34 minutes
Sofia International Airport	42.69482 N	23.61331 E	SOF	BULGARIA	UTC+02:00	Tue 14 Nov 2023, 01:48	4.775657	530.8253	53 minutes
Aurel Vlaicu International Airport	44.50296 N	26.10131 E	BBU	ROMANIA	UTC+02:00	Tue 14 Nov 2023, 01:48	6.763119	751.7111	1 hours, 16 minutes
Cluj International Airport Avram Iancu	46.78735 N	23.68408 E	CLJ	ROMANIA	UTC+02:00	Tue 14 Nov 2023, 01:48	8.85305	983.9622	1 hours, 39 minutes
Alexandria International Airport Borg El Arab	30.92896 N	29.69700 E	HBE	EGYPT	UTC+02:00	Tue 14 Nov 2023, 01:48	8.464354	941.0466	1 hours, 35 minutes
Rabat - Sale Airport	34.04113 N	6.757719 W	RBA	MOROCCO	UTC+01:00	Tue 14 Nov 2023, 00:48	25.030956	2782.6962	4 hours, 41 minutes
Belgrade Airport Nikola Tesla	44.81734 N	20.30815 E	BEG	SERBIA	UTC+01:00	Tue 14 Nov 2023, 00:48	7.400389	822.5379	1 hours, 23 minutes
Tirana International Airport	41.41522 N	19.72049 E	TIA	ALBANIA	UTC+01:00	Tue 14 Nov 2023, 00:48	4.760737	529.1795	53 minutes
Catania Airport Vincenzo Bellini	37.46700 N	15.06692 E	CTA	ITALY	UTC+01:00	Tue 14 Nov 2023, 00:48	7.637723	782.3352	1 hours, 19 minutes
Palermo Airport Falcone Borsellino	38.17982 N	13.10302 E	PMO	ITALY	UTC+01:00	Tue 14 Nov 2023, 00:48	8.535772	948.8504	1 hours, 36 minutes
Malta International Airport	35.85265 N	14.48484 E	MLA	MALTA	UTC+01:00	Tue 14 Nov 2023, 00:48	7.844465	872.0413	1 hours, 28 minutes
Tunis-Carthage International Airport	36.85119 N	10.22709 E	TUN	TUNISIA	UTC+01:00	Tue 14 Nov 2023, 00:48	10.942823	1216.4522	2 hours, 3 minutes

Φυσικά, ο υπολογισμός της διάρκειας πτήσης δεν είναι ακριβής και όπως αναφέρθηκε και νωρίτερα, θα μπορούσε να βελτιωθεί με διάφορους τρόπους. Ένας από αυτούς θα ήταν να λαμβάνεται υπ' όψιν η διεύθυνση των διαδρόμων στο κάθε αεροδρόμιο. Στα δεξιά φαίνεται ο υπολογισμός της διεύθυνσης ενός διαδρόμου του Ελ. Βενιζέλος, μέσω του property *direction*.

```
126
127 if __name__ == "__main__":
128     runway = Airport.Runway(0, "03L/21R", 3800, 0)
129     print(f"RUNWAY {runway.name} DIRECTION:", runway.direction)
130
```

Run airport ~

C:\Users\alexa\PycharmProjects\Databases\assets\airport.py

DATABASE AIRPORT CONNECTED

RUNWAY 03L/21R DIRECTION: 3

CHANGES\_COMMITTED DB\_CLOSED

## 6.5 Schedules Printing

Παρακάτω φαίνονται τα περιεχόμενα του πίνακα Schedule, επεξεργασμένα για να τυπώνονται τα ονόματα των αεροπορικών εταιρειών, οι κωδικοί των αεροδρομίων και οι μέρες που θα εκτελείται η κάθε προγραμματισμένη πτήση σε μορφή πιο φιλική από τον ακέραιο που είναι αποθηκευμένος στη βάση.

FLIGHT	AIRLINE	FROM	TO	DEPARTURE	ARRIVAL	WEEK DAYS	LAST MODIFICATION
A3-101	Aegean Airlines	CFU	ATH	14:50		TUE WED SAT	2024-01-13 19:52:09
A3-143	Aegean Airlines	ATH	MLA	17:35		TUE THU FRI	2024-01-13 19:52:09
A3-155	Aegean Airlines	ATH	CFU	23:15		SUN TUE THU	2024-01-13 19:52:09
A3-225	Aegean Airlines	KIT	ATH	12:55		MON THU FRI	2024-01-13 19:52:09
A3-240	Aegean Airlines	ATH	CTA	01:20		MON WED FRI	2023-12-25 12:56:43
A3-256	Aegean Airlines	ATH	CFU	18:45		SUN TUE FRI	2024-01-13 19:52:09
A3-342	Aegean Airlines	IST	ATH	09:50		TUE WED FRI SAT	2024-01-13 19:52:09
A3-348	Aegean Airlines	KLX	ATH	02:30		SUN TUE THU FRI	2024-01-13 19:52:09
A3-417	Aegean Airlines	ATH	IOA	04:00		MON FRI	2024-01-13 19:52:09
A3-425	Aegean Airlines	ATH	CHQ	10:10		SUN THU SAT	2023-12-25 09:54:23
A3-481	Aegean Airlines	CTA	ATH	20:15		SUN MON THU SAT	2024-01-13 19:52:09
A3-492	Aegean Airlines	ATH	IST	15:50		SUN MON SAT	2024-01-13 19:52:09
A3-551	Aegean Airlines	ATH	JTR	16:50		SUN MON TUE SAT	2024-01-13 19:52:09
A3-553	Aegean Airlines	ATH	CTA	21:20		WED SAT	2024-01-13 19:52:09
A3-584	Aegean Airlines	ATH	LRA	17:05		SUN WED FRI	2024-01-13 19:52:09
A3-592	Aegean Airlines	IST	ATH	12:50		SUN MON TUE SAT	2024-01-13 19:52:09
A3-612	Aegean Airlines	ATH	ZTH	22:45		TUE	2024-01-13 19:52:09
A3-634	Aegean Airlines	ATH	CTA	12:20		MON TUE WED FRI	2024-01-13 19:52:09

## 6.6 Insert New Employee

Στα δεξιά φαίνεται το παράθυρο διαλόγου που δημιουργείται μέσω του PySimpleGUI για την εισαγωγή στοιχείων νέου εργαζομένου.

Enter the following information for a new Employee record:

Ssn: 123456123

First: MyFirstName

Middle: Middle

Last: MyLastName

Telephone: 6900000000

Email:

Street:

Number:

Town:

Postal\_code:

Birth Date: 1980-01-18

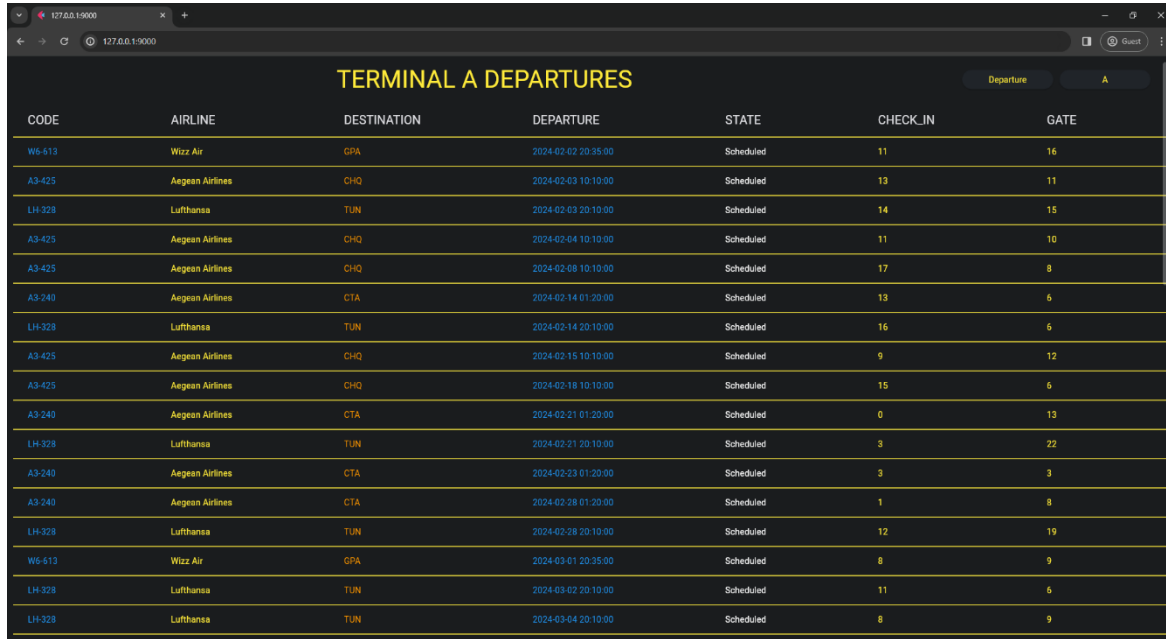
ent ID: Administration

Sex: male

Submit

## 6.7 Flights by Terminal

Παρακάτω φαίνεται δείγμα από υποθετική ιστοσελίδα στο localhost που προβάλλει πτήσεις που επίκειται να διενεργηθούν. Με το αριστερό κουμπί πάνω δεξιά ή το πλήκτρο **C** ο επισκέπτης μπορεί να μεταβεί από τις Αναχωρήσεις στις Αφίξεις, με το δεξί κουμπί ή το πλήκτρο **T** μπορεί να αλλάξει τερματικό σταθμό, ενώ με το πλήκτρο **R** μπορεί να κάνει ανανέωση της σελίδας.



CODE	AIRLINE	DESTINATION	DEPARTURE	STATE	CHECK_IN	GATE
W6-613	Wizz Air	GPA	2024-02-02 20:35:00	Scheduled	11	16
A3-425	Aegean Airlines	CHQ	2024-02-03 10:10:00	Scheduled	13	11
LH-328	Lufthansa	TUN	2024-02-03 20:10:00	Scheduled	14	15
A3-425	Aegean Airlines	CHQ	2024-02-04 10:10:00	Scheduled	11	10
A3-425	Aegean Airlines	CHQ	2024-02-08 10:10:00	Scheduled	17	8
A3-240	Aegean Airlines	CTA	2024-02-14 01:20:00	Scheduled	13	6
LH-328	Lufthansa	TUN	2024-02-14 20:10:00	Scheduled	16	6
A3-425	Aegean Airlines	CHQ	2024-02-15 10:10:00	Scheduled	9	12
A3-425	Aegean Airlines	CHQ	2024-02-16 10:10:00	Scheduled	15	6
A3-240	Aegean Airlines	CTA	2024-02-21 01:20:00	Scheduled	0	13
LH-328	Lufthansa	TUN	2024-02-21 20:10:00	Scheduled	3	22
A3-240	Aegean Airlines	CTA	2024-02-23 01:20:00	Scheduled	3	3
A3-240	Aegean Airlines	CTA	2024-02-28 01:20:00	Scheduled	1	8
LH-328	Lufthansa	TUN	2024-02-28 20:10:00	Scheduled	12	19
W6-613	Wizz Air	GPA	2024-03-01 20:35:00	Scheduled	8	9
LH-328	Lufthansa	TUN	2024-03-02 20:10:00	Scheduled	11	6
LH-328	Lufthansa	TUN	2024-03-04 20:10:00	Scheduled	8	9

## 6.8 Airport Map

```
===== AIRPORT ===== LATITUDE ==== LONGITUDE ==== IATA ==== COUNTRY ===== TIMEZONE ===== LOCAL TIME =====
Belgrade Airport Nikola Tesla 44.81734 N 20.30815 E BEG SERBIA UTC+01:00 Sun 14 Jan 2024, 15:59
Malta International Airport 35.85205 N 14.48404 E MLA MALTA UTC+01:00 Sun 14 Jan 2024, 15:59
```

```
#####
##
##
## [BEG]
##
##
## [MLA]
##
##
##
##
## |
## |
## -----O-----
## |
## |
#####
```

## 7 ΠΑΡΑΡΤΗΜΑ

Ένα από τα πρώτα ζητήματα που έπρεπε να λυθεί ήταν το πώς θα περιγράφονταν οι επαναλαμβανόμενες πτήσεις στη βάση, τελικά αποφασίστηκε να υπάρχει ξεχωριστός πίνακας με υποδείγματα, τα οποία θα χρησιμοποιούνταν για να φτιάχνουν πτήσεις σε διαφορετικές ημερομηνίες, πάντα με ίδια αεροπορική εταιρεία, κωδικό, ώρα.

Στο σημείο αυτό προέκυψε πρόβλημα με την αναπαράσταση δεδομένων ημερομηνίας και / ή ώρας στη βάση, καθώς η πιο ικανοποιητική επιλογή που προσφέρει η SQLite φαίνεται πως είναι το string representation. Για αυτόν τον λόγο χρησιμοποιείται το enumeration DateFormat, το οποίο αποθηκεύει τα formats που χρησιμοποιούνται για την ημερομηνία, την ώρα και τον συνδυασμό τους. Όπως φαίνεται στον πίνακα Schedule, το μόνο που ενδιαφέρει για μια προγραμματισμένη πτήση είναι η ώρα απογείωσης ή προσγείωσης, οι εγγραφές όμως έχουν αποθηκευτεί με datetime format, το οποίο έγινε κατά λάθος και δεν διορθώθηκε ποτέ, καθώς, αν και κακή πρακτική δεν δημιουργήσε πρόβλημα.

Άλλο θέμα που εμφανίστηκε στον πίνακα Schedule ήταν το πώς θα αποθηκεύονταν οι μέρες της εβδομάδας που θα διενεργείται η συγκεκριμένη προγραμματισμένη πτήση. Αποφασίστηκε να κωδικοποιούνται μέσω του Enumeration Day σε έναν ακέραιο αριθμό. Οι πιθανοί συνδυασμοί των επτά ημερών της εβδομάδας είναι  $2^7$ , οπότε ο αριθμός είναι από το 0 μέχρι το 127. Η μέρα 0 είναι η Κυριακή, το Σάββατο είναι η μέρα 6. Αν το πεδίο days του Schedule γράφει 1, σημαίνει ότι η πτήση θα γίνεται μόνο τα Σάββατα. Αν γράφει 2, θα γίνεται μόνο τις Παρασκευές, αν γράφει 8, θα γίνεται τις Τετάρτες, αν γράφει 64, θα γίνεται μόνο τις Κυριακές. Σε άλλο παράδειγμα, αν μια πτήση γίνεται τις καθημερινές (Δε, Τρ, Τε, Πε, Πα: 1, 2, 3, 4, 5:  $2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 32 + 16 + 8 + 4 + 2 = 62$ ), αναπαρίσταται με τον αριθμό 62. Η διαδικασία μετατροπής λίστας με ημέρες, δηλαδή λίστα με members του Day enumeration σε αριθμό και το αντίστροφο, γίνονται με συναρτήσεις της Day. Ακολουθεί παράδειγμα χρήσης τους:

```
from assets import Day

first_example: list[Day] = [Day.MONDAY, Day.WEDNESDAY, Day.FRIDAY]
print(Day.days_to_code(first_example)) # prints 42

second_example: int = 100
print(", ".join([str(day) for day in Day.code_to_days(second_example)]))
# prints Day.SUNDAY, Day.MONDAY, Day.THURSDAY
```

Ένα άλλο πρόβλημα που εξακολουθεί να υπάρχει, είναι οι ανακρίβειες στον υπολογισμό απόστασης σημείων, δηλαδή στην απόσταση δύο αεροδρομίων. Για να λυθεί αυτό, δοκιμάστηκαν διάφορες λύσεις που έχουν διατηρηθεί στην κλάση Coordinates. Η συγκεκριμένη κλάση δοκιμάστηκε με πολλούς τρόπους εκτός της χρήσης της στα αεροδρόμια της βάσης, για αυτό έχουν υλοποιηθεί πολλές dunder μέθοδοι για τα αντικείμενά της.

Όσον αφορά την κλάση Database που διαχειρίζεται τη βάση, τα σημαντικότερα προβλήματα που παρατηρήθηκαν ήταν τα εξής:

Προέκυπταν Exceptions όταν χρησιμοποιούνταν πάνω από μία συνδέσεις ή cursors, για να λυθεί αυτό επεξεργάστηκε η κλάση ώστε να μην επιτρέπει τη δημιουργία περισσότερων από ενός αντικειμένων της. Αργότερα, έγινε και μια τροποποίηση στη δημιουργία του sqlite3.Connection στην προσπάθεια να μπορεί να λειτουργήσει η βιβλιοθήκη παράλληλα με αυτές των γραφικών.

Η κλάση Database έχει τα sqlite3.Connection και Cursor αποθηκευμένα σε ομώνυμα protected instance attributes, αυτό έχει γίνει για να μην μπορεί ο χρήστης να εκτελέσει query χρησιμοποιώντας άμεσα το cursor. Αντί για αυτό, η κλήση του αντικειμένου καλεί εσωτερικά την Cursor.execute() για να πραγματοποιηθεί το query. Με την πρακτική αυτή, η βάση δεν επιτρέπει στον χρήστη της τη διαγραφή πίνακα και μετρά τα queries που εκτελούνται.

Όταν ολοκληρώνεται η εκτέλεση ενός script και ο garbage collector σβήνει τα αντικείμενα του προγράμματος, το αντικείμενο της κλάσης Database πριν διαγραφεί, κάνει αυτόματα commit και μετά κλείνει το connection με τη βάση, οπότε αυτά δεν είναι κάτι που πρέπει να απασχολεί τον χρήστη της βάσης.



Έγινε προσπάθεια το αντικείμενο Database να μπορεί να χρησιμοποιηθεί και σαν context manager, αυτό έμεινε ανολοκλήρωτο.

Στα πρώτα στάδια χρήσης της βάσης, τα queries γράφονταν με χρήση f-string, μέχρι που όλα τα f-strings αντικαταστάθηκαν με την προτεινόμενη στο internet σύνταξη, με τα arguments του query ξεχωριστά.

Κάποια μέρα, το script που πέρναγε εγγραφές στον πίνακα *Schedule*, δεν λειτούργησε σωστά λόγω παραπανίσιων quotes και έγραψε *None* στη στήλη του πίνακα που προοριζόταν να μείνει κενή. Γενικότερα, η χρήση των quotes, του question mark για placeholder έγινε συνήθεια σε διάστημα εβδομάδων.

## 8 ΑΠΟΘΕΤΗΡΙΟ ΚΩΔΙΚΑ

[ΣΥΝΔΕΣΜΟΣ GITHUB](#)

## 9 ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ

Εγκατάσταση του βιβλιοθήκης PySimpleGUI [3] από [εδώ](#).

Εγκατάσταση του βιβλιοθήκης Flet [4] από [εδώ](#).

Συγκεκριμένα, για το project απαιτείται η εγκατάσταση των παραπάνω βιβλιοθηκών στη συσκευή μας μέσω PIP:

```
pip install PySimpleGUI
pip install flet
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Python 3.11 Documentation. Retrieved October 9, 2023 from <https://www.python.org/doc/>
- [2] SQLite Documentation. Retrieved November 6, 2023 from <https://www.sqlite.org/docs.html>
- [3] Πακέτο γραφικής διεπαφής PySimpleGUI. Retrieved December 23, 2023 from <https://www.pysimplegui.org/en/latest/>
- [4] Πακέτο γραφικής διεπαφής Flet (Flutter for Python). Retrieved January 10, 2024 from <https://flet.dev/docs/>
- [5] Airport Departments. Retrieved December 20, 2023 from <https://www.flybillings.com/1748/Airport-Division>
- [6] Wikipedia. 2023. Wikipedia: the Free Encyclopedia. Κατάλογος αεροδρομίων της Ελλάδας. Retrieved November 2023 from [https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%AC%CE%BB%CE%BF%CE%B3%CE%BF%CF%82\\_%CE%B1%CE%B5%CF%81%CE%BF%CE%B4%CF%81%CE%BF%CE%BC%CE%AF%CF%89%CE%BD\\_%CF%84%CE%B7%CF%82\\_%CE%95%CE%BB%CE%B%CE%AC%CE%B4%CE%B1%CF%82](https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%AC%CE%BB%CE%BF%CE%B3%CE%BF%CF%82_%CE%B1%CE%B5%CF%81%CE%BF%CE%B4%CF%81%CE%BF%CE%BC%CE%AF%CF%89%CE%BD_%CF%84%CE%B7%CF%82_%CE%95%CE%BB%CE%B%CE%AC%CE%B4%CE%B1%CF%82)
- [7] Wikipedia. 2023. Wikipedia: the Free Encyclopedia. Διάδρομος αεροδρομίου. Retrieved October 2023 from [https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%AC%CE%B4%CF%81%CE%BF%CE%BC%CE%BF%CF%82\\_%CE%B1%CE%B5%CF%81%CE%BF%CE%B4%CF%81%CE%BF%CE%BC%CE%AF%CE%BF%CF%85%CE%9F%CE%BD%CE%BF%CE%BC%CE%B1%CF%83%CE%AF%CE%B1\\_%CE%B4%CE%B9%CE%B1%CE%B4%CF%81%CF%8C%CE%BC%CF%89%CE%BD](https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%AC%CE%B4%CF%81%CE%BF%CE%BC%CE%BF%CF%82_%CE%B1%CE%B5%CF%81%CE%BF%CE%B4%CF%81%CE%BF%CE%BC%CE%AF%CE%BF%CF%85%CE%9F%CE%BD%CE%BF%CE%BC%CE%B1%CF%83%CE%AF%CE%B1_%CE%B4%CE%B9%CE%B1%CE%B4%CF%81%CF%8C%CE%BC%CF%89%CE%BD)
- [8] Geeks for Geeks. 2023. Program for distance between two points on earth. Retrieved October 2023 from <https://www.geeksforgeeks.org/program-distance-two-points-earth/>
- [9] ERD Maker. 2021. Website: <https://hci.ece.upatras.gr/erdmaker/>