

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Βάσεις Δεδομένων – ECE_ΓΚ703

ΕΦΑΡΜΟΓΗ ΑΝΑΖΗΤΗΣΗΣ/ΚΡΑΤΗΣΗΣ ΑΕΡΟΠΟΡΙΚΩΝ ΕΙΣΙΤΗΡΙΩΝ

Ομάδα 50

ΧΡΗΣΤΟΣ ΚΑΤΣΑΝΔΡΗΣ – 1072755

ΓΕΩΡΓΙΟΣ ΤΣΙΑΛΙΟΣ – 1072868

1 ΕΙΣΑΓΩΓΗ – ΣΤΟΧΟΣ ΟΜΑΔΙΚΗΣ ΕΡΓΑΣΙΑΣ

Το θέμα της ομαδικής εργασίας είναι η κατασκευή ενός προγράμματος με δύο κύριους στόχους: την αναζήτηση αεροπορικών εισιτηρίων και την κράτηση τους. Η επιδιωκόμενη λειτουργία της εφαρμογής είναι εύκολα κατανοητή: ο χρήστης αρχικά εισάγει ορισμένες πληροφορίες όπως η τοποθεσία στην οποία επιθυμεί να ταξιδέψει, το χρονικό διάστημα στο οποίο επιθυμεί να πραγματοποιηθεί το ταξίδι, το πλήθος των ταξιδιωτών κ.α. Στη συνέχεια, το πρόγραμμα πραγματοποιεί μία σχετική αναζήτηση στη βάση δεδομένων του και εμφανίζει στον χρήστη τις διαθέσιμες πτήσεις και πληροφορίες σχετικά με αυτές. Έπειτα, ο χρήστης μπορεί να επιλέξει μία από αυτές τις πτήσεις και να ακολουθήσει μία σειρά από βήματα, προκειμένου να πραγματοποιήσει την κράτηση των εισιτηρίων. Υπάρχουν πολλές εφαρμογές αυτού του τύπου στο διαδίκτυο, όπως για παράδειγμα η [Skyscanner](#) και η [airtickets](#). Συμβουλευτήκαμε σε μεγάλο βαθμό τέτοιες εφαρμογές για να πάρουμε ιδέες σχετικά με το τι θα πρέπει να παρέχει η εφαρμογή μας στον τελικό χρήστη και συνδυάσαμε χαρακτηριστικά από αρκετές από αυτές. Εν ολίγοις, προσπαθήσαμε να δημιουργήσουμε ένα περιβάλλον φιλικό προς τον χρήστη, δίνοντας του ταυτόχρονα πολλές επιλογές και δυνατότητες, τις οποίες θα δούμε αναλυτικά στη συνέχεια.

Βέβαια, η πολυπλοκότητα του προβλήματος του πραγματικού κόσμου μας οδήγησε σε ορισμένες παραδοχές στη λύση μας. Η πρώτη από αυτές σχετίζεται με την πληρωμή των αεροπορικών εισιτηρίων. Συγκεκριμένα, όπως προαναφέραμε, η εργασία μας στόχευσε και στην υλοποίηση της κράτησης των εισιτηρίων (κάτι το οποίο δεν πραγματοποιούν αρκετές από τις εφαρμογές αυτού του είδους, οι οποίες παραπέμπουν τον χρήστη στην σελίδα της αντίστοιχης αεροπορικής εταιρίας ή σε άλλη εφαρμογή). Ωστόσο, δεν συμπεριλάβαμε στην εφαρμογή μας το κομμάτι της πληρωμής των εισιτηρίων. Θεωρήσαμε δηλαδή, πως η πληρωμή του κόστους των εισιτηρίων θα γίνει μέσα από το περιβάλλον της τράπεζας και όχι από την εφαρμογή μας.

Παρόμοιες θεωρήσεις έγιναν καθ' όλη τη διάρκεια του σχεδιασμού και της κατασκευής της εφαρμογής, άλλοτε για λόγους απλοποίησης του προβλήματος και άλλοτε εξαιτίας της αδυναμίας εύρεσης κατάλληλων δεδομένων για τη βάση δεδομένων μας. Θα αναφέρουμε όλες τις αυτές τις παραδοχές στην επόμενη ενότητα, στην οποία θα ασχοληθούμε πιο αναλυτικά με τη διαδικασία ανάπτυξης της εφαρμογής μας.

Οι γλώσσες προγραμματισμού που χρησιμοποιήσαμε ήταν η SQL (σε περιβάλλον SQLite) για την κατασκευή της βάσης δεδομένων και για την εκτέλεση ερωτημάτων (Queries) προς αυτήν, καθώς και η Python για τη δημιουργία της εφαρμογής.

2 ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ

2.1 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Το πρώτο βήμα για την ανάπτυξη της εφαρμογής μας είναι η δημιουργία μίας βάσης δεδομένων (data base), η οποία θα περιέχει όλες τις χρήσιμες πληροφορίες των πτήσεων. Η δημιουργία της βάσης χωρίζεται σε 3 στάδια:

1. Σχεδιασμός Βάσης Δεδομένων
2. Κατασκευή Βάσης Δεδομένων
3. Εισαγωγή δεδομένων στη Βάση Δεδομένων

2.1.1 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Αρχικά, μοντελοποιήσαμε το πρόβλημα του φυσικού κόσμου, κατασκευάζοντας το διάγραμμα Οντοτήτων – Συσχετίσεων (Entity Relationship Diagram). Για την κατασκευή του διαγράμματος, χρησιμοποιήσαμε το εργαλείο [ERDMaker](#). Η τελική μορφή του διαγράμματος, η οποία προέκυψε μετά από πολλές αλλαγές σε όλη τη διάρκεια ανάπτυξης της εφαρμογής, φαίνεται στην *Εικόνα 1*.

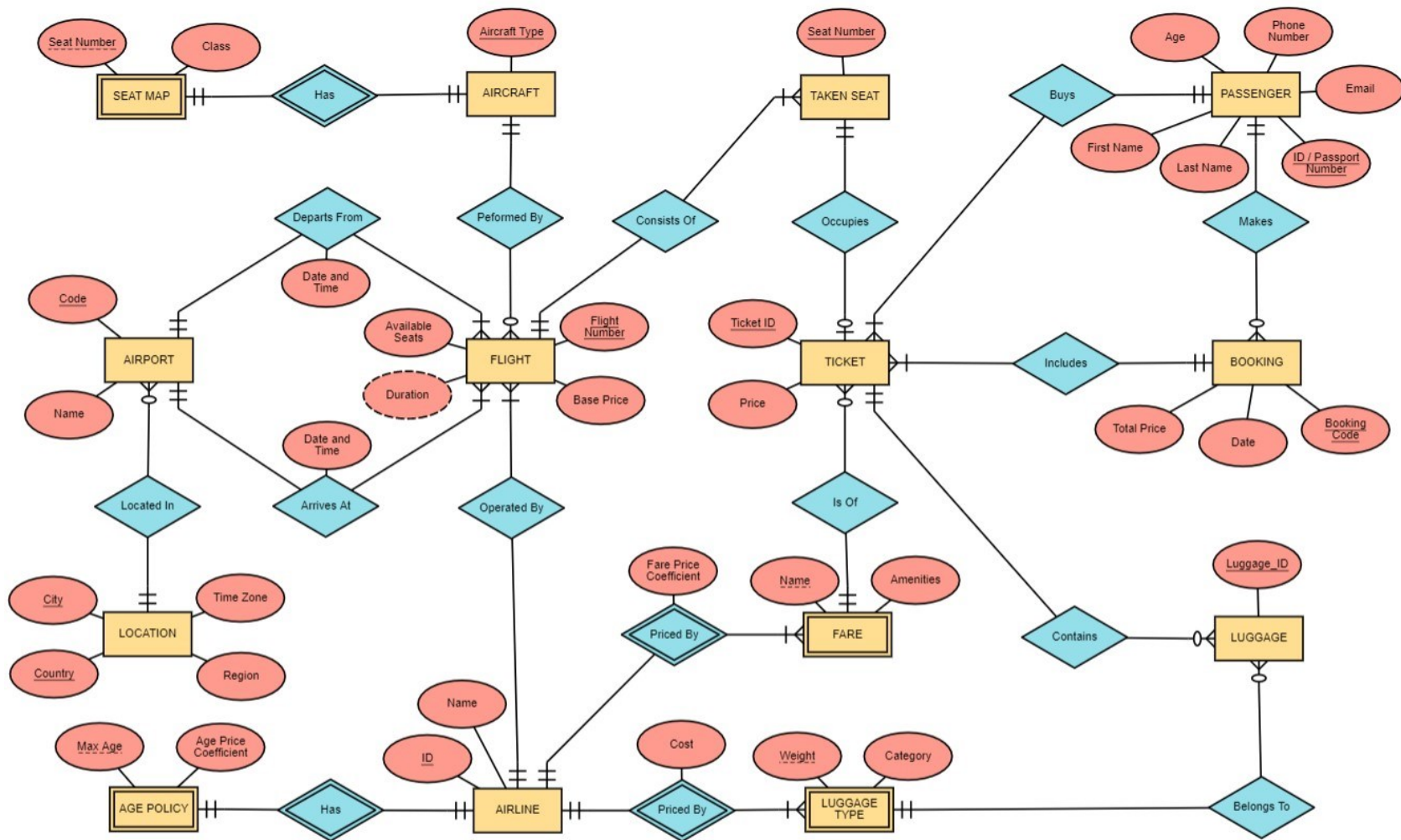
Περιγραφή διαγράμματος Οντοτήτων-Συσχετίσεων:

Το διάγραμμα διακρίνεται νοηματικά σε 2 τμήματα: αυτό που σχετίζεται με την αναζήτηση πτήσεων (αριστερό μέρος του διαγράμματος) με κεντρική οντότητα την *FLIGHT* και σε αυτό που σχετίζεται με την κράτηση εισιτηρίων (δεξιό μέρος του διαγράμματος) με κεντρικές οντότητες την *TICKET* και την *AIRLINE*. Ας αναλύσουμε τη μορφή του διαγράμματος και με τον τρόπο αυτό θα γίνει πιο σαφής η δομή της βάσης δεδομένων, καθώς και η λειτουργία της εφαρμογής μας.

Α) Αναζήτηση πτήσεων:

Ξεκινάμε από την οντότητα *AIRPORT*, η οποία έχει ως πρωτεύον κλειδί τον κωδικό του αεροδρομίου και ως γνώρισμα το όνομα του. Συσχετίζεται με την οντότητα *LOCATION*, η οποία περιλαμβάνει τις πληροφορίες σχετικά με την τοποθεσία του αεροδρομίου καθώς και με την ζώνη ώρας στην οποία αυτό βρίσκεται.

Η οντότητα *FLIGHT* αντιπροσωπεύει την κάθε πτήση και έχει ως γνωρίσματα τον κωδικό της πτήσης (*flight number*) και τη βασική τιμή της (*base price*). Ως βασική τιμή ορίσαμε το φθηνότερο εισιτήριο κάθε πτήσης, δηλ. την τιμή του εισιτηρίου για έναν ενήλικα, για τον φθηνότερο ναύλο (πχ Economy Light) και χωρίς πρόσθετη αποσκευή. Με άλλα λόγια, πρόκειται για την τιμή που συνοδεύει την πτήση και με την οποία έρχεται πρώτα σε επαφή ο χρήστης (πχ πτήσεις προς Παρίσι από 49€). Η τιμή αυτή, ανανεώνεται συνεχώς στον πραγματικό κόσμο, με βάση την ζήτηση. Ωστόσο, στην εφαρμογή μας θεωρήσαμε ότι η βασική τιμή κάθε πτήσης παραμένει σταθερή. Η οντότητα *FLIGHT* χαρακτηρίζεται επιπλέον από το πεδίο *available seats* (πλήθος διαθέσιμων θέσεων πτήσης) και το υπολογιζόμενο γνώρισμα *duration* (διάρκεια πτήσης). Θα περιγράψουμε τη διαδικασία υπολογισμού τους παρακάτω.



Εικόνα 1: Διάγραμμα Οντοτήτων-Συσχετίσεων

Υπάρχουν 5 συσχετίσεις μεταξύ της οντότητας *FLIGHT* και άλλων οντοτήτων. Συγκεκριμένα, η κάθε πτήση συνδέεται με την οντότητα *AIRPORT* με 2 τρόπους: μέσω της συσχέτισης *Departs From*, η οποία αφορά το αεροδρόμιο αναχώρησης και έχει ως γνώρισμα την ημερομηνία και ώρα αναχώρησης της πτήσης, και μέσω της συσχέτισης *Arrives At*, η οποία αφορά το αεροδρόμιο άφιξης και έχει ως γνώρισμα την ημερομηνία και ώρα άφιξης της πτήσης. Η διάρκεια κάθε πτήσης προκύπτει μέσα από αυτά τα 2 γνωρίσματα ως:

$$duration = arrival datetime - departure datetime$$

Ακόμα, η οντότητα *FLIGHT* συνδέεται με τις οντότητες *AIRLINE* και *AIRCRAFT*, ώστε να γνωρίζουμε για κάθε πτήση την αεροπορική εταιρία και τον τύπο του αεροσκάφους με το οποίο εκτελείται. Τέλος, η οντότητα *FLIGHT* συσχετίζεται με την οντότητα *TAKEN SEAT*. Για να εξηγήσουμε τη συγκεκριμένη οντότητα, πρέπει πρώτα να περιγράψουμε την προσέγγιση της εφαρμογής μας σχετικά με τις θέσεις κάθε πτήσης.

Είναι γνωστό, ότι δεν έχουν όλες οι πτήσεις το ίδιο πλάνο θέσεων (Seat Map), δηλ. πλήθος θέσεων και κατηγορία (class) της κάθε θέσης. Στην εφαρμογή μας, θεωρήσαμε ότι το πλάνο θέσεων εξαρτάται μόνο από τον τύπο του αεροσκάφους. Έτσι, ορίσαμε την οντότητα *SEAT MAP* ως ασθενή οντότητα, η οποία εξαρτάται από την οντότητα *AIRCRAFT*. Γνωρίζουμε ότι κάθε πτήση πραγματοποιείται από ένα αεροσκάφος, κι έτσι μπορούμε να βρούμε για κάθε πτήση το πλάνο θέσεων της. Αυτό είναι πιο αποδοτικό από άποψη χώρου από το να συσχετίζαμε απευθείας το πλάνο θέσεων με την κάθε πτήση, γιατί το πλήθος των πτήσεων στη βάση μας είναι πολύ μεγαλύτερο του πλήθους των τύπων αεροσκαφών. Η οντότητα *SEAT MAP* έχει γνωρίσματα τον αριθμό της θέσης (*seat number*) και την κατηγορία της, δηλ. Business ή Economy. Επομένως, η οντότητα *SEAT MAP* περιέχει θέσεις – αυτές του κάθε αεροσκάφους. Ωστόσο, δεν είναι η μόνη οντότητα του διαγράμματος μας που αφορά θέσεις. Υπάρχει και η οντότητα *TAKEN SEAT*, η οποία περιγράφει τις κατειλημμένες θέσεις μίας πτήσης. Για αυτό η οντότητα *TAKEN SEAT*, όπως αναφέραμε προηγουμένως, συνδέεται απευθείας με την οντότητα *FLIGHT*. Επομένως, καταλήγουμε στο ότι μπορούμε να βρούμε για κάθε πτήση όλες τις θέσεις της, ενώ κρατάμε αρχείο για τις κατειλημμένες θέσεις της κάθε πτήσης. Αφαιρώντας τα δύο αυτά σύνολα, προκύπτουν οι διαθέσιμες θέσεις κάθε πτήσης. Μέσω της καταμέτρησης αυτών των θέσεων υπολογίζεται το γνώρισμα *available seats*. Συγκεκριμένα, σε ψευδοκώδικα γράφεται ως εξής:

$$available\ seats = COUNT(seats - taken_seats)$$

Με τον τρόπο αυτόν αρχικοποιείται το γνώρισμα *available seats*. Κατά την διάρκεια εκτέλεσης του προγράμματος, κάθε φορά που ο χρήστης κάνει κράτηση ενός εισιτηρίου, τότε η αντίστοιχη θέση της πτήσης θα προστίθεται στον πίνακα *TAKEN_SEAT* και το γνώρισμα *available seats* της πτήσης θα μειώνεται κατά 1.

B) Κράτηση Εισιτηρίων:

Ας ξεκινήσουμε με την οντότητα *TICKET*, η οποία αναπαριστά το εισιτήριο μίας πτήσης. Η οντότητα αυτή συνδέεται με την *TAKEN SEAT*, ώστε να φαίνεται στο εισιτήριο η θέση της πτήσης την οποία καταλαμβάνει. Κάθε εισιτήριο αντιστοιχεί σε έναν επιβάτη (οντότητα *PASSENGER*) και συμπεριλαμβάνεται σε μία κράτηση (οντότητα *BOOKING*). Θα εξετάσουμε τις οντότητες αυτές πιο αναλυτικά παρακάτω. Η οντότητα *TICKET* έχει 2 γνωρίσματα: τον κωδικό του εισιτηρίου (*ticket number* – primary key) και την τιμή του εισιτηρίου (*price*). Αξίζει να δούμε πως υπολογίζεται η τιμή του εισιτηρίου. Για να το κάνουμε αυτό, πρέπει πρώτα να αναλύσουμε την οντότητα *AIRLINE*.

Η οντότητα *AIRLINE* έχει μόνο δύο γνωρίσματα: τον κωδικό της αεροπορικής (*airline ID* – primary key) και το όνομα της (*airline name*). Έχει όμως σημαντικό ρόλο στο διάγραμμα, χάρη στις συσχετίσεις της.

Η πρώτη από αυτές είναι με την ασθενή οντότητα *AGE POLICY*, η οποία περιγράφει το πως κοστολογεί η κάθε αεροπορική εταιρία τους επιβάτες της, με βάση την ηλικία τους. Έτσι, η οντότητα *AGE POLICY* έχει ως γνωρίσματα έναν συντελεστή τιμής (*age price coefficient*), ο οποίος εξαρτάται από την ηλικία του επιβάτη (*max age*) και την αεροπορική εταιρία. Για παράδειγμα, η αεροπορική εταιρία Aegean Airlines εφαρμόζει τον συντελεστή 0.66 επί της βασικής τιμής, για παιδιά κάτω των 12 ετών. Θα ασχοληθούμε με τον υπολογισμό αυτών, καθώς και άλλων συντελεστών, στην ενότητα ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ.

Μία άλλη σύνδεση της οντότητας *AIRLINE* γίνεται με την ασθενή οντότητα *FARE*. Η οντότητα αυτή, περιγράφει τον ναύλο του εισιτηρίου και για αυτό συνδέεται επίσης με την οντότητα *TICKET*. Έχει ως χαρακτηριστικά το όνομα του ναύλου (*name* – partial key) και το πεδίο *amenities*, το οποίο υλοποιείται ως string και περιγράφει τις υπηρεσίες που παρέχει ο κάθε ναύλος. Η κάθε αεροπορική παρέχει διαφορετικές υπηρεσίες στους ναύλους της και τους κοστολογεί διαφορετικά (γνώρισμα *fare price coefficient*). Για παράδειγμα, η αεροπορική εταιρία Aegean Airlines εφαρμόζει τον συντελεστή 1.06 επί της βασικής τιμής, για τον ναύλο Economy Flex.

Τέλος, η οντότητα *AIRLINE* συσχετίζεται με την ασθενή οντότητα *LUGGAGE TYPE*, η οποία περιέχει τους τύπους αποσκευών που επιτρέπει η κάθε αεροπορική εταιρία. Ο κάθε τύπος αποσκευής προσδιορίζεται από το βάρος του (*weight* – partial key) και ανήκει σε μία κατηγορία (*category*), ανάλογα με το αν η χιεραποσκευή είναι παραδοτέα ή όχι. Η κάθε αεροπορική κοστολογεί διαφορετικά τους τύπους αποσκευών της (γνώρισμα *cost*). Για παράδειγμα, η αεροπορική εταιρία Aegean Airlines κοστολογεί την αποσκευή των 23kg στα 27€. Η οντότητα *LUGGAGE TYPE* συνδέεται ακόμα με την οντότητα *LUGGAGE*, η οποία αναπαριστά τις αποσκευές που πληρώνουν οι επιβάτες. Έτσι, η κάθε αποσκευή ανήκει σε έναν τύπο αποσκευής. Επιπλέον, η οντότητα *LUGGAGE* συσχετίζεται με την οντότητα *TICKET*, ώστε να φαίνεται σε κάθε εισιτήριο, οι αποσκευές που συμπεριλαμβάνονται σε αυτό. Αξίζει να σημειωθεί, ότι κάποιες αεροπορικές προσφέρουν δωρεάν αποσκευές σε ορισμένους ναύλους τους. Αυτό φαίνεται στο πεδίο *amenities* του κάθε ναύλου και δεν αφορά την οντότητα *LUGGAGE*. Με άλλα λόγια, η οντότητα *LUGGAGE* περιλαμβάνει μόνο τις αποσκευές εκείνες, που έχει πληρώσει ξεχωριστά ο επιβάτης.

Πλέον, μπορούμε να εξηγήσουμε τον τρόπο με τον οποίο υπολογίζεται η τιμή του κάθε εισιτηρίου. Συγκεκριμένα, το πεδίο *price* του *TICKET* υπολογίζεται από τον τύπο:

$$\text{price} = \text{age price coefficient} \cdot \text{fare price coefficient} \cdot \text{base price} \\ + \text{LUGGAGE_TYPE.cost}$$

Δηλ. οι συντελεστές που είδαμε προηγουμένως πολλαπλασιάζονται επί την βασική τιμή της κάθε πτήσης, ενώ προστίθεται το κόστος πληρωτέας αποσκευής, εφόσον αυτή υπάρχει. Γίνεται φανερό, ότι ο υπολογισμός της τιμής του εισιτηρίου περιλαμβάνει κάποιες παραδοχές. Ειδικότερα, η επιβάρυνση της τιμής του εισιτηρίου που προκαλείται από τον ναύλο και τις αποσκευές είναι σταθερή για μία αεροπορική και δεν εξαρτάται από την συγκεκριμένη πτήση.

Το κομμάτι της κράτησης εισιτηρίων ολοκληρώνεται με τις οντότητες *PASSENGER* και *BOOKING*. Η οντότητα *PASSENGER* περιέχει τα στοιχεία των επιβατών και συνδέεται με την οντότητα *TICKET*, ώστε κάθε εισιτήριο να αντιστοιχεί σε έναν επιβάτη. Η οντότητα *BOOKING* εκφράζει την κράτηση, η οποία περιλαμβάνει ένα έως πολλά εισιτήρια. Έχει

γνωρίσματα τον κωδικό της κράτησης (*booking code* – primary key), την ημερομηνία της καθώς και την συνολική τιμή της κράτησης (*total price*), η οποία υπολογίζεται ως:

$$total\ price = \sum ticket\ price$$

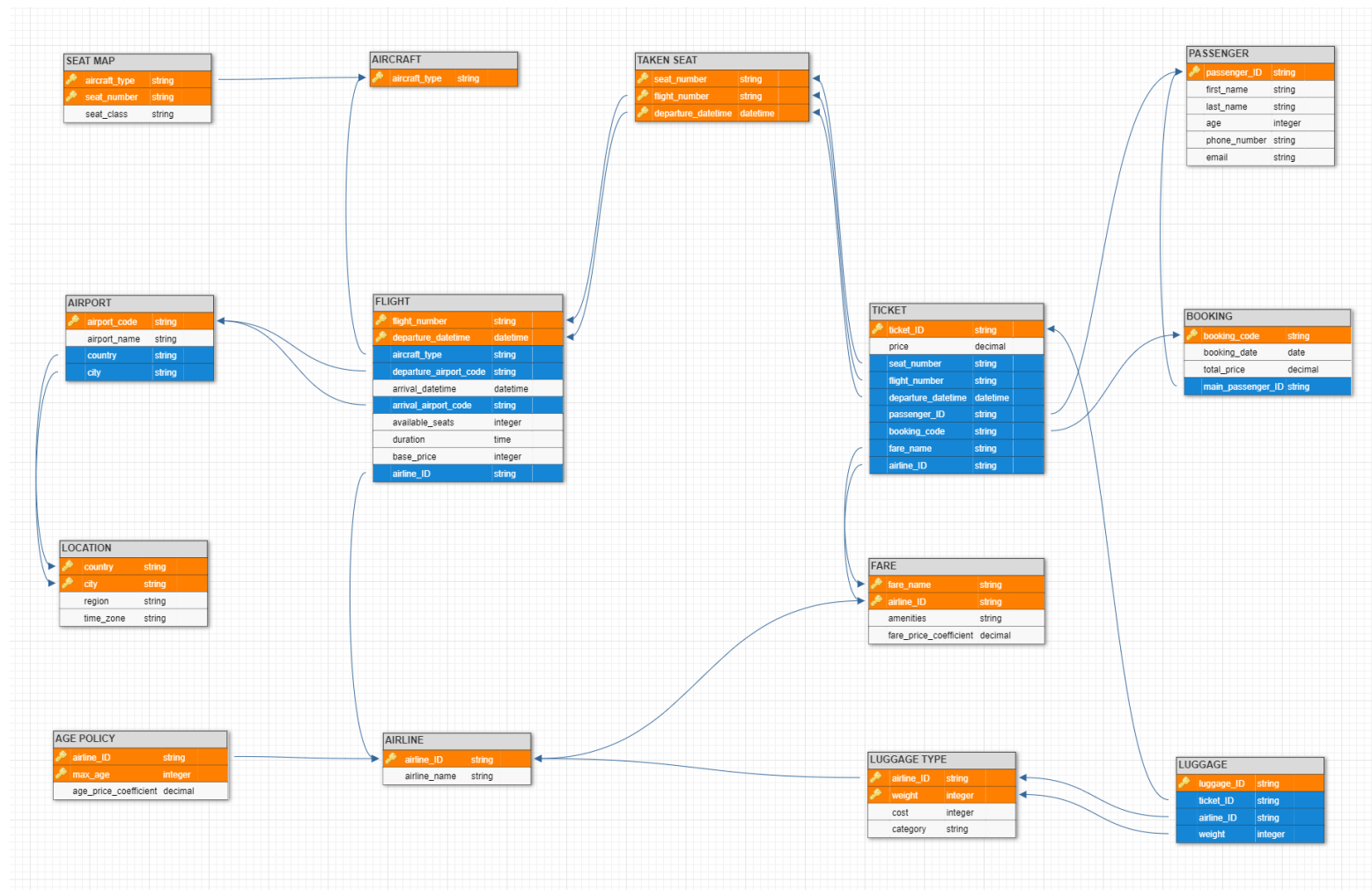
Ωστόσο, μία κράτηση πραγματοποιείται μόνο από έναν επιβάτη. Για να γίνουν καλύτερα κατανοητές οι συσχετίσεις μεταξύ αυτών των οντοτήτων και οι πληθικότητες τους, ας δούμε ένα παράδειγμα: Έστω μία τετραμελής οικογένεια, η οποία αποτελείται από τον πατέρα, τη μητέρα και δύο παιδιά. Η οικογένεια επιθυμεί να ταξιδέψει στο Παρίσι. Αφού επιλέξουν πτήση, θα κληθούν να ρυθμίσουν τα εισιτήρια τους. Συγκεκριμένα, για κάθε εισιτήριο, θα ζητηθεί να εισάγουν τα στοιχεία του επιβάτη, να επιλέξουν ναύλο και τυχόν πρόσθετη αποσκευή. Στη συνέχεια, θα οριστεί ένας από τους 2 ενήλικες ως master passenger, δηλ. αυτός ο οποίος θα πραγματοποιήσει την κράτηση. Αφού διεκπεραιωθεί και η κράτηση, πλέον η βάση μας θα περιέχει τα εξής δεδομένα: 1 κράτηση στην οποία φαίνεται ο master passenger, 4 εισιτήρια στα οποία φαίνεται ο επιβάτης τους και η κράτηση στην οποία ανήκουν.

Μετατροπή Διαγράμματος Οντοτήτων – Συσχετίσεων στο Σχεσιακό Μοντέλο

Το επόμενο βήμα της φάσης σχεδιασμού ήταν η μετατροπή του διαγράμματος Οντοτήτων – Συσχετίσεων στο Σχεσιακό μοντέλο (Schema). Για την κατασκευή του Σχεσιακού μοντέλου, χρησιμοποιήσαμε το εργαλείο [dbdesigner](#). Η τελική μορφή του μοντέλου φαίνεται στην *Εικόνα 2*.

Για τη μετάβαση από το διάγραμμα Οντοτήτων – Συσχετίσεων στο Σχεσιακό μοντέλο, χρησιμοποιήσαμε τους αντίστοιχους κανόνες που περιέχονται στο [1]. Έπειτα, ελέγξαμε ότι οι σχέσεις που προέκυψαν είναι στην 2^η και 3^η Κανονική Μορφή και όπου χρειάστηκε, κάναμε κανονικοποίηση σχέσεων, όπως περιγράφεται στο [2]. Έτσι προέκυψε για παράδειγμα ο πίνακας *LOCATION*, ξεχωριστός του *AIRPORT*, καθώς παρατηρήσαμε ότι το πεδίο *time_zone* δεν εξαρτάται από το ίδιο το αεροδρόμιο (*airport_code*), αλλά από την τοποθεσία του, και συγκεκριμένα από την πόλη του. Με τον τρόπο αυτό αποφύγαμε να επαναλαμβάνουμε το ίδιο *time_zone* για αεροδρόμια που ανήκουν στην ίδια πόλη.

Στο σχήμα μας, συμβολίσαμε με πορτοκαλί χρώμα τα πρωτεύοντα κλειδιά (primary keys) και με μπλε χρώμα τα ξένα κλειδιά (foreign keys). Παρατηρούμε ότι έχουν προστεθεί πολλά ξένα κλειδιά στους πίνακες μας, τα οποία όμως είναι απαραίτητα για τη σωστή λειτουργία της βάσης.



Εικόνα 2: Σχεσιακό μοντέλο

2.1.2 ΚΑΤΑΣΚΕΥΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Μετά τον θεωρητικό σχεδιασμό της βάσης δεδομένων της εφαρμογής μας, ακολουθεί η υλοποίηση της. Ο κώδικας SQL για την κατασκευή της βάσης βρίσκεται στο αρχείο FlyNow.sql, το οποίο επισυνάπτεται στο παραδοτέο αρχείο. Δεν θα αναφερθούμε σε ολόκληρο τον κώδικα, παρά μόνο σε ορισμένα αξιοσημείωτα σημεία.

Αρχικά, συμπεριλάβαμε όλον τον κώδικα για την κατασκευή της βάσης σε μία δοσοληψία (Transaction). Αυτό επιτυγχάνεται με την εντολή `BEGIN TRANSACTION;`, η οποία σηματοδοτεί την έναρξη της εκτέλεσης δοσοληψίας και την εντολή `END TRANSACTION;`, η οποία αποθηκεύει τις αλλαγές που έγιναν στη βάση και σηματοδοτεί το επιτυχές τέλος της δοσοληψίας. Με την χρήση δοσοληψίας, ομαδοποιήσαμε όλες τις εντολές σε μία ενιαία ενότητα εργασίας. Έτσι, σε περίπτωση σφάλματος κάποιου μπλοκ εντολών, θα αναιρεθεί το αντίκτυπο των προηγούμενων εντολών που ήδη εκτελέστηκαν (Rollback).

Εφαρμόσαμε διάφορους περιορισμούς (Constraints) στα γνωρίσματα των πινάκων. Αρχικά, προσδιορίσαμε τον τύπο δεδομένων κάθε γνωρίσματος (πχ `varchar`, `integer`, `date`), δηλ. κάθε γνώρισμα έχει ένα πεδίο ορισμού. Έπειτα, προσθέσαμε επιπλέον περιορισμούς με τις εξής εντολές:

- `NOT NULL` - ορίζει ότι το γνώρισμα δεν μπορεί να πάρει τιμή `NULL`.
- `GENERATED ALWAYS AS` - χρησιμοποιείται για τον προσδιορισμό της τιμής των υπολογιζόμενων γνωρισμάτων (generated columns). Έτσι, κατά την εισαγωγή δεδομένων, δεν εισάγουμε τιμή στην στήλη αυτή, αλλά υπολογίζεται από πράξεις σε άλλες στήλες.
- `PRIMARY KEY()` - έτσι ορίσαμε τα πρωτεύοντα κλειδιά του πίνακα. Οι περιορισμοί που εφαρμόζονται αυτόματα είναι `NOT NULL` για κάθε κλειδί, ενώ ο συνδυασμός τους πρέπει να είναι μοναδικός (`UNIQUE`).
- `FOREIGN KEY()` - έτσι ορίσαμε τα ξένα κλειδιά του πίνακα. Εφαρμόζεται αυτόματα ο περιορισμός της αναφορικής ακεραιότητας μεταξύ των σχέσεων, δηλ. δεν επιτρέπεται να υπάρχει αναφορά σε ανύπαρκτη πλειάδα της άλλης σχέσης. Μάλιστα, προσθέτοντας τον κώδικα: `ON DELETE ... ON UPDATE ...` σε κάθε δήλωση ξένου κλειδιού, ορίσαμε τι θα γίνει με τη συγκεκριμένη πλειάδα σε περίπτωση διαγραφής της πλειάδας της άλλης σχέσης ή σε περίπτωση ενημέρωσης του πρωτεύοντος κλειδιού της.
- `CHECK()` - με τον τρόπο αυτό εισάγαμε επιπλέον περιορισμούς, ώστε να ικανοποιούνται κάποιες λογικές απαιτήσεις. Για παράδειγμα ο εξής περιορισμός:

```
CHECK ("arrival_datetime" > "departure_datetime")
```

ελέγχει ότι η άφιξη μίας πτήσης έπεται χρονικά της αναχώρησης της.

Τέλος, χρησιμοποιήσαμε ευρετήρια (Indexes) για να αυξήσουμε την απόδοση του συστήματος μας, δηλ. να επιταχύνουμε την εκτέλεση των Queries στη βάση. Δημιουργήσαμε ευρετήρια με την εντολή `CREATE INDEX` και με βάση τις ανάγκες της εφαρμογής μας, δηλ. τα Queries τα οποία εκτελούμε. Εδώ να σημειωθεί ότι η SQLite δημιουργεί αυτόματα ευρετήρια για το πρωτεύον κλειδί κάθε πίνακα (`sqlite_autoindex`). Έτσι, αρχικά δημιουργήσαμε διάφορα ευρετήρια, πολλά από τα οποία παρατηρήσαμε ότι δεν χρησιμοποιούσε η βάση μας. Εξετάσαμε ποια ευρετήρια χρησιμοποιούνταν από την βάση, τοποθετώντας το μπλοκ εντολών: `EXPLAIN QUERY PLAN` πριν το Query που επιθυμούσαμε που να εκτελέσουμε. Επομένως, κρατήσαμε μόνο τα χρήσιμα ευρετήρια – αυτά τα οποία χρησιμοποιούσε η βάση και μείωναν τον χρόνο αναζήτησης. Στα παρακάτω παραδείγματα, θα δούμε πιο αναλυτικά την βελτίωση που επέφεραν τα ευρετήρια μας στον χρόνο εκτέλεσης τυπικών Queries της εφαρμογής μας.

detail		flight_1_number	departure_airport_code	departure_city	flight_1_departure_dat
SCAN TABLE FLIGHT AS FLIGHT_1		1	LH1753	ATH	Athens
SEARCH TABLE AIRPORT AS DEPARTURE_AIRPORT USING INDEX sqlite_autoindex_AIRPORT_1 (airport_code=?)		2	A3610	ATH	Athens
SEARCH TABLE AIRPORT AS INTERMEDIATE_AIRPORT USING INDEX sqlite_autoindex_AIRPORT_1 (airport_code=?)		3	A3850	ATH	Athens
SEARCH TABLE AIRLINE AS AIRLINE_1 USING INDEX sqlite_autoindex_AIRLINE_1 (airline_ID=?)					
SEARCH TABLE FLIGHT AS FLIGHT_2 USING AUTOMATIC COVERING INDEX (departure_airport_code=?)					
SEARCH TABLE AIRPORT AS ARRIVAL_AIRPORT USING INDEX sqlite_autoindex_AIRPORT_1 (airport_code=?)					
SEARCH TABLE AIRLINE AS AIRLINE_2 USING INDEX sqlite_autoindex_AIRLINE_1 (airline_ID=?)					
USE TEMP B-TREE FOR ORDER BY					
		Execution finished without errors. Result: 37 rows returned in 3498ms			

detail		flight_1_number	departure_airport_code	departure_city	flight_1_departure_dat
SEARCH TABLE AIRPORT AS DEPARTURE_AIRPORT USING INDEX idx_AIRPORT (city=?)		1	LH1753	ATH	Athens
SEARCH TABLE FLIGHT AS FLIGHT_1 USING INDEX idx_FLIGHT (departure_airport_code=?)		2	A3610	ATH	Athens
SEARCH TABLE AIRPORT AS INTERMEDIATE_AIRPORT USING INDEX sqlite_autoindex_AIRPORT_1 (airport_code=?)		3	A3850	ATH	Athens
SEARCH TABLE AIRLINE AS AIRLINE_1 USING INDEX sqlite_autoindex_AIRLINE_1 (airline_ID=?)					
SEARCH TABLE FLIGHT AS FLIGHT_2 USING INDEX idx_FLIGHT (departure_airport_code=?)					
SEARCH TABLE AIRPORT AS ARRIVAL_AIRPORT USING INDEX sqlite_autoindex_AIRPORT_1 (airport_code=?)					
SEARCH TABLE AIRLINE AS AIRLINE_2 USING INDEX sqlite_autoindex_AIRLINE_1 (airline_ID=?)					
USE TEMP B-TREE FOR ORDER BY					
		Execution finished without errors. Result: 37 rows returned in 1057ms			

Εικόνα 3: Εκτέλεση Query αναζήτησης πτήσεων με 1 στάση

Αρχικά, στην Εικόνα 3, μελετάμε την εκτέλεση ενός Query αναζήτησης πτήσεων με 1 στάση. Συγκεκριμένα, αναζητούμε πτήσεις με 1 στάση από Αθήνα προς Βαρκελώνη, την 20^η Μαρτίου του 2023 (θα αναλύσουμε τον κώδικα των Queries στην Ενότητα 2.2).

Πρώτα θα εκτελέσουμε το Query χωρίς να δημιουργήσουμε indexes. Στην Εικόνα 3.1, βλέπουμε τις ενέργειες που θα εκτελέσει η SQLite για τον υπολογισμό του Query. Παρατηρούμε ότι θα διατρέξει αρχικά όλο τον πίνακα FLIGHT. Αυτό είναι αρκετά χρονοβόρο, εξαιτίας του μεγέθους και των πολλών γνωρισμάτων του πίνακα αυτού. Για αυτό, στην Εικόνα 3.2 βλέπουμε πως το Query επέστρεψε 37 πλειάδες ως αποτελέσματα, σε χρόνο $\approx 3.5s$.

Στη συνέχεια, θα εκτελέσουμε το Query έχοντας πλέον δημιουργήσει τα εξής indexes:

- `idx_AIRPORT` με πεδίο ευρετηρίασης το γνώρισμα `city`
- `idx_FLIGHT` με πεδίο ευρετηρίασης το γνώρισμα `departure_airport_code`

Στην Εικόνα 3.3 βλέπουμε τις ενέργειες που θα εκτελέσει τώρα η SQLite για τον υπολογισμό του Query. Παρατηρούμε ότι χρησιμοποιεί τα indexes που δημιουργήσαμε (επιλεγμένες με μπλε γραμμές). Πλέον, στην Εικόνα 3.4 βλέπουμε πως το Query επέστρεψε τις ίδιες 37 πλειάδες με πριν σε χρόνο $\approx 1s$, δηλ. η αναζήτηση έγινε περίπου 3 φορές γρηγορότερη.

	flight_number	departure_airport_code	departure
1	VY8101	ATH	Athens
2	A3710	ATH	Athens
Execution finished without errors. Result: 2 rows returned in 165ms			

	flight_number	departure_airport_code	departure
1	VY8101	ATH	Athens
2	A3710	ATH	Athens
Execution finished without errors. Result: 2 rows returned in 15ms			

Εικόνα 4: Εκτέλεση Query αναζήτησης απευθείας πτήσεων

Ακόμα μεγαλύτερη βελτίωση παρατηρείται στην περίπτωση του Query αναζήτησης απευθείας πτήσεων, όπως φαίνεται στην *Εικόνα 4*. Συγκεκριμένα, τώρα αναζητούμε απευθείας πτήσεις από Αθήνα προς Βαρκελώνη, την 20^η Μαρτίου του 2023.

Εδώ, παρατηρούμε πως η αναζήτηση με χρήση indexes είναι περίπου 10 φορές γρηγορότερη.

2.1.3 ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΣΤΗ ΒΑΣΗ

Μετά την κατασκευή της βάσης δεδομένων, σειρά έχει η εισαγωγή δεδομένων σε αυτή. Προκειμένου να κάνουμε την εφαρμογή μας όσο το δυνατόν πιο αληθοφανή, χρησιμοποιήσαμε πραγματικά δεδομένα πτήσεων, τα οποία ανακτήσαμε από το διαδίκτυο με τη χρήση REST APIs.

Αρχικά, ανακτήσαμε δεδομένα αεροδρομίων, πόλεων και χωρών από τους ιστότοπους [FlightLabs](#), [Airlabs](#) και [RESTCountries](#) αντίστοιχα. Τα δεδομένα αεροδρομίων περιέχουν πληροφορίες, όπως ο κωδικός τριών γραμμάτων του αεροδρομίου, το όνομά του, η πόλη στην οποία ανήκει, η ζώνη ώρας του κ.λπ. Τα δεδομένα πόλεων περιέχουν το όνομα της πόλης, τον κωδικό δύο γραμμάτων της χώρας στην οποία βρίσκεται κ.λπ. Τα δεδομένα χωρών περιέχουν το όνομά της, την ήπειρο στην οποία βρίσκεται κ.λπ. Κάθε μία από αυτές τις πληροφορίες χρησιμοποιείται για την αναζήτηση πτήσεων από τον χρήστη.

Στη συνέχεια, ανακτήσαμε δεδομένα αεροπορικών εταιρειών από τον ιστότοπο [FlightLabs](#). Τα δεδομένα αεροπορικών εταιρειών περιέχουν πληροφορίες, όπως οι κωδικοί δύο και τριών γραμμάτων της αεροπορικής, το όνομά της κ.λπ.

Έπειτα, ανακτήσαμε από τον ιστότοπο [FlightLabs](#) όλες τις μελλοντικές πτήσεις που είναι προγραμματισμένες να αναχωρήσουν από 21 μεγάλα αεροδρόμια του κόσμου, κάθε μία μέρα του Μαρτίου 2023. Τα αεροδρόμια που χρησιμοποιήσαμε είναι: AMS, ATH, AUH, BCN, BER, CDG, DOH, DXB, HEL, HKG, HND, ICN, IST, JFK, LGW, LHR, MUC, MXP, SIN, VIE, ZRH. Τα δεδομένα αυτά περιέχουν πληροφορίες, όπως οι κωδικοί τριών γραμμάτων των αεροδρομίων αναχώρησης και άφιξης, οι προγραμματισμένες ώρες αναχώρησης και άφιξης, ο τύπος του αεροσκάφους με το οποίο θα εκτελεστεί η πτήση, ο αριθμός της πτήσης, η αεροπορική εταιρεία κ.λπ.

Όλα τα παραπάνω δεδομένα τα αποθηκεύσαμε τοπικά σε αρχεία JSON.

Για τα δεδομένα των πινάκων *AIRCRAFT* και *SEAT_MAP*, συγκεντρώσαμε όλους τους τύπους αεροσκαφών με τα οποία εκτελούνται οι παραπάνω πτήσεις και βρήκαμε το πλάνο θέσεων τους. Κατασκευάσαμε το αντίστοιχο αρχείο JSON, με τη βοήθεια ενός script στην Python.

Για τα δεδομένα των πινάκων *FARE*, *AGE_POLICY* και *LUGGAGE_TYPE*, κατασκευάσαμε τα αντίστοιχα αρχεία JSON με έναν αλγόριθμο παραγωγής τυχαίων αριθμών, οι οποίοι προσπαθήσαμε να ανταποκρίνονται όσο καλύτερα γίνεται στην πραγματικότητα.

Έχοντας πλέον έτοιμα όλα τα απαραίτητα δεδομένα για τη βάση μας σε αρχεία JSON, δημιουργούμε τους πίνακες με τα γνωρίσματα και τα constraints που περιεγράφηκαν παραπάνω και εκτελούμε διαδοχικά εντολές INSERT, με τη βοήθεια των κατάλληλων scripts στην Python. Με τον ίδιο τρόπο δημιουργούμε τα indexes που περιεγράφηκαν παραπάνω.

Οι υπόλοιποι πίνακες, οι οποίοι αφορούν στην κράτηση εισιτηρίων, θα γεμίσουν στη συνέχεια, όταν το πρόγραμμά μας χρησιμοποιηθεί από πελάτες. Ωστόσο, για να ελέγξουμε την ορθή λειτουργία της βάσης δεδομένων και του προγράμματος, εισαγάγαμε δεδομένα και σε αυτούς τους πίνακες, με τη βοήθεια των κατάλληλων scripts στην Python.

2.2 ΚΑΤΑΣΚΕΥΗ ΕΡΩΤΗΜΑΤΩΝ (QUERIES) ΠΡΟΣ ΤΗΝ ΒΑΣΗ ΣΕ SQLite

Πλέον, έχοντας την βάση μας έτοιμη προς χρήση, το επόμενο βήμα είναι η συγγραφή ερωτημάτων (queries) προς την βάση, ανάλογα με τις ανάγκες της εφαρμογής μας. Τα queries που κατασκευάσαμε ανήκουν σε 2 κατηγορίες:

- Queries σχετικά με την αναζήτηση πτήσεων
- Queries σχετικά με την κράτηση εισιτηρίων

Queries σχετικά με την αναζήτηση πτήσεων

Τα queries αυτά δέχονται ως είσοδο από τον χρήστη τα φίλτρα της αναζήτησης και επιστρέφουν ως αποτέλεσμα τις διαθέσιμες πτήσεις και πληροφορίες σχετικά με αυτές. Να σημειωθεί ότι η παραμετροποίηση των queries θα γίνει στη συνέχεια, κατά την ενσωμάτωση της SQLite στην Python. Επομένως εδώ, θα εξετάσουμε ένα συγκεκριμένο παράδειγμα, για να κατανοήσουμε τη λειτουργία των queries. Τα queries αναζήτησης πτήσεων χωρίζονται σε 3 βασικές κατηγορίες:

- Queries αναζήτησης απευθείας πτήσεων
- Queries αναζήτησης πτήσεων με 1 στάση
- Queries αναζήτησης πτήσεων με 2 στάσεις

Ο κώδικας για τα queries με στάσεις ακολουθεί την ίδια λογική, όμως είναι πιο μεγάλος. Για αυτό, χάριν ευκολίας, θα αναλύσουμε ένα query αναζήτησης απευθείας πτήσεων. Συγκεκριμένα, έστω το εξής παράδειγμα:

Επιθυμώ να ταξιδέψω από την Αθήνα στην Βαρκελώνη, στις 20 Μαρτίου του 2023, μέσω της Aegean Airlines και με κόστος λιγότερο από 100€. Δείξε μου τις διαθέσιμες πτήσεις και τις τιμές τους.

Το συγκεκριμένο Query υλοποιείται ως εξής:

```
SELECT

    flight_number,

    departure_airport_code,

    DEPARTURE_AIRPORT.city AS departure_city,

    departure_datetime,

    arrival_airport_code,

    ARRIVAL_AIRPORT.city AS arrival_city,

    arrival_datetime,

    available_seats,

    airline_name,

    duration AS duration_MIN,

    base_price

FROM
```

(

```

(
    FLIGHT JOIN AIRPORT AS
DEPARTURE_AIRPORT ON departure_airport_code =
DEPARTURE_AIRPORT.airport_code
)
JOIN AIRPORT AS ARRIVAL_AIRPORT ON
arrival_airport_code = ARRIVAL_AIRPORT.airport_code
)
NATURAL JOIN AIRLINE

WHERE
    DEPARTURE_AIRPORT.city = 'Athens'
AND
    ARRIVAL_AIRPORT.city = 'Barcelona' AND
    departure_datetime LIKE '2023-03-20 __:__:__' AND
    available_seats > 0 AND
    airline_name = 'Aegean Airlines' AND
    base_price < 100

ORDER BY base_price, departure_datetime;

```

Επεξήγηση Κώδικα:

Ξεκινάμε από το FROM, όπου παίρνουμε όλες τις πτήσεις με τα αεροδρόμια αναχώρησης και άφιξης τους και την αεροπορική εταιρία που εκτελεί την κάθε πτήση.

Έπειτα στο WHERE, κρατάμε μόνο τις πτήσεις που πληρούν τις εξής προϋποθέσεις: η πόλη αναχώρησης είναι η Αθήνα, η πόλη άφιξης είναι η Βαρκελώνη, η αναχώρηση της πτήσης γίνεται την 20^η Μαρτίου του 2023, η πτήση έχει διαθέσιμες θέσεις, εκτελείται από την Aegean Airlines και έχει βασική τιμή < 100€.

Στη συνέχεια στο SELECT, επιλέγουμε τις χρήσιμες πληροφορίες των πτήσεων που θα εμφανιστούν στον χρήστη. Αυτές είναι: ο κωδικός της πτήσης, ο κωδικός και η πόλη του αεροδρομίου αναχώρησης, η ημερομηνία και ώρα αναχώρησης, ο κωδικός και η πόλη του αεροδρομίου άφιξης, η ημερομηνία και ώρα άφιξης, το πλήθος διαθέσιμων θέσεων της πτήσης, το όνομα της αεροπορικής εταιρίας, η διάρκεια της πτήσης σε λεπτά και η βασική τιμή της πτήσης.

Τέλος, στο ORDER BY ταξινομούμε τα αποτελέσματα με βάση την βασική τιμή της πτήσης και την ώρα αναχώρησης της.

Αξίζει να σημειωθεί, ότι η αναζήτηση πτήσεων μπορεί να γίνει και εναλλακτικά, χρησιμοποιώντας διαφορετικά φίλτρα. Αυτά είναι τα εξής:

- Όσον αφορά την τοποθεσία: αναζήτηση με βάση την χώρα ή την ήπειρο άφιξης
- Όσον αφορά την ημερομηνία αναχώρησης: αναζήτηση με βάση τον μήνα αναχώρησης

Επίσης, η εφαρμογή μας υποστηρίζει και αναζήτηση πτήσεων μετ' επιστροφής. Αυτό υλοποιείται ως δύο διαφορετικά queries αναζήτησης πτήσης, ένα για να πάμε στον προορισμό μας και ένα για να γυρίσουμε από αυτόν. Θα δούμε πιο αναλυτικά τη λειτουργία αυτή στην Ενότητα 3: Αποτελέσματα από την εκτέλεση του προγράμματος.

Queries σχετικά με την κράτηση εισιτηρίων

Τα Queries κράτησης εισιτηρίων χωρίζονται σε 2 βασικές κατηγορίες:

- Queries εμφάνισης διαθέσιμων θέσεων πτήσης
- Queries υπολογισμού τιμής εισιτηρίου

Πρόκειται για 2 διαφορετικούς τύπους query, επομένως θα δούμε το κάθε ένα ξεχωριστά:

Ας ξεκινήσουμε με τα Queries εμφάνισης διαθέσιμων θέσεων πτήσης. Έστω το εξής παράδειγμα:

Δείξε μου τις διαθέσιμες θέσεις της πτήσης A3710 την 20^η Μαρτίου του 2023:

Το συγκεκριμένο Query υλοποιείται ως εξής:

```
SELECT seat_number, seat_class

FROM SEAT_MAP NATURAL JOIN (

    SELECT aircraft_type

    FROM FLIGHT

    WHERE

        flight_number = 'A3710'

AND

        departure_datetime LIKE '2023-03-

20 __:__:__'

)

WHERE seat_number NOT IN (

    SELECT seat_number

    FROM TAKEN_SEAT

    WHERE
```

```

                                flight_number = 'A3710'
AND
                                departure_datetime LIKE '2023-03-
20 __:__:__'
                                )

ORDER BY seat_number;

```

Επεξήγηση Κώδικα:

Το συγκεκριμένο query περιέχει 2 nested queries – ένα στο FROM και ένα στο WHERE. Η λογική του όμως είναι απλή:

Το nested query του FROM μας δίνει το *aircraft_type* της πτήσης A3710. Όπως αναφέραμε και στην *Ενότητα 2.1.1*, από αυτό εξαρτάται το πλάνο θέσεων της πτήσης. Έτσι, κάνοντας NATURAL JOIN με τον πίνακα *SEAT_MAP* παίρνουμε όλες τις θέσεις της πτήσης A3710. Από αυτές, θέλουμε να κρατήσουμε μόνο τις διαθέσιμες. Για αυτό στο WHERE, κρατάμε μόνο τις θέσεις της πτήσης που δεν βρίσκονται στον πίνακα *TAKEN_SEAT*. Έπειτα στο SELECT επιλέγουμε να εμφανίσουμε στον χρήστη για κάθε διαθέσιμη θέση, τον αριθμό της και την κατηγορία της (Economy / Business) και τέλος στο ORDER BY ταξινομούμε τις θέσεις με αύξουσα σειρά.

Στην συνέχεια, ας αναλύσουμε τα Queries υπολογισμού τιμής εισιτηρίου. Έστω το εξής παράδειγμα:

Δείξε μου την τιμή του εισιτηρίου για την πτήση A3710 την 20^η Μαρτίου του 2023, για παιδί ηλικίας 10 ετών, για ναύλο Economy Flex και με μία επιλεγμένη αποσκευή των 23 κιλών.

Το συγκεκριμένο Query υλοποιείται ως εξής:

```

SELECT
    flight_number,
    departure_datetime,
    airline_name,
    ROUND(fare_price_coefficient *
age_price_coefficient * base_price + LUGGAGE_1.cost,2) AS price
FROM
    (
        (
            (

```

```

                                FLIGHT NATURAL JOIN AIRLINE
                                )
                                NATURAL JOIN FARE
                                )
                                NATURAL JOIN AGE_POLICY
                                )
                                NATURAL JOIN LUGGAGE_TYPE AS LUGGAGE_1

WHERE

    flight_number = 'A3710'                                AND
    departure_datetime LIKE '2023-03-20 __:__:__' AND
    fare_name = 'Economy Flex'                                AND
    max_age = 12                                            AND
    LUGGAGE_1.weight = 23;

```

Επεξήγηση Κώδικα:

Ξεκινάμε από το FROM, όπου παίρνουμε όλες τις πτήσεις με τις αεροπορικές που τις εκτελούν, με τους ναύλους των αεροπορικών, την ηλικιακή κοστολόγηση τους και τις διαθέσιμες αποσκευές τους.

Έπειτα στο WHERE, επιλέγουμε την πτήση A3710 την 20^η Μαρτίου του 2023, με ναύλο Economy Flex, στην κατηγορία παιδί κάτω των 12 ετών, με αποσκευή των 23 κιλών.

Στη συνέχεια στο SELECT, επιλέγουμε να εμφανιστεί στον χρήστη ο κωδικός της πτήσης, η ημερομηνία και ώρα αναχώρησης της, η αεροπορική που την εκτελεί και η τιμή για το εισιτήριο που ζήτησε. Παρατηρούμε ότι για τον υπολογισμό της τιμής του εισιτηρίου, χρησιμοποιήσαμε τον τύπο που περιγράψαμε στην *Ενότητα 2.1.1*.

Αξίζει ακόμα να αναφερθεί, ότι το συγκεκριμένο query έχει αρκετές παραλλαγές, ανάλογα με την ηλικία του επιβάτη και το πλήθος των αποσκευών που έχει επιλέξει. Παρόλα αυτά, η φιλοσοφία της λύσης παραμένει η ίδια σε όλα τα queries αυτού του είδους.

2.3 ΚΑΤΑΣΚΕΥΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ PYTHON

Για την κατασκευή του προγράμματος *FlyNow*, έγινε χρήση της γλώσσας Python και του αντικειμενοστραφούς μοντέλου προγραμματισμού. Κατασκευάσαμε μία κλάση για κάθε μία οντότητα της βάσης δεδομένων και εισαγάγαμε τα αντίστοιχα γνωρίσματα. Κατασκευάσαμε μία βοηθητική κλάση `Trip`, η οποία απεικονίζει ένα ταξίδι (το οποίο μπορεί να αποτελείται από μία, δύο ή τρεις διακριτές πτήσεις).

Οι τρεις βασικές κλάσεις του προγράμματος είναι οι εξής:

- Η κλάση `DatabaseService`, η οποία πραγματοποιεί όλες τις επικοινωνίες με τη βάση δεδομένων, κάνοντας χρήση της βιβλιοθήκης `sqlite3`. Αξίζει να σημειωθεί πως κάθε φορά που επιτυγχάνεται σύνδεση με τη βάση, εκτελούμε την εντολή `PRAGMA foreign_keys=ON;` ώστε να επιβάλουμε τον έλεγχο αναφορικής ακεραιότητας, ο οποίος είναι από προεπιλογή απενεργοποιημένος.
- Η κλάση `NavigatorService`, η οποία υλοποιεί το Command Line Interface.
- Η κλάση `App`, η οποία περιέχει τις global μεταβλητές του προγράμματος, διατηρεί ένα αντικείμενο τύπου `DatabaseService` για την επικοινωνία του προγράμματος με τη βάση και δημιουργεί ένα αντικείμενο τύπου `NavigatorService` για την εκτέλεση του Command Line Interface.

Ιδιαίτερης μνείας χρήζουν οι κλάσεις `DatabaseService` και `NavigatorService`, τις οποίες θα εξετάσουμε αναλυτικά παρακάτω.

Η κλάση `DatabaseService`

Η κλάση `DatabaseService` υλοποιεί στις μεθόδους της όλα τα queries που κατασκευάστηκαν για την επικοινωνία με τη βάση δεδομένων. Οι μέθοδοι `searchTripsZeroStops`, `searchTripsOneStop`, `searchTripsTwoStops` υλοποιούν τα queries αναζήτησης πτήσεων με καμία, μία ή δύο στάσεις αντίστοιχα. Οι μέθοδοι αυτές είναι πλήρως παραμετροποιημένα και χτίζουν το query string σταδιακά, ανάλογα με τις επιλογές που κάνει ο χρήστης στο Command Line Interface.

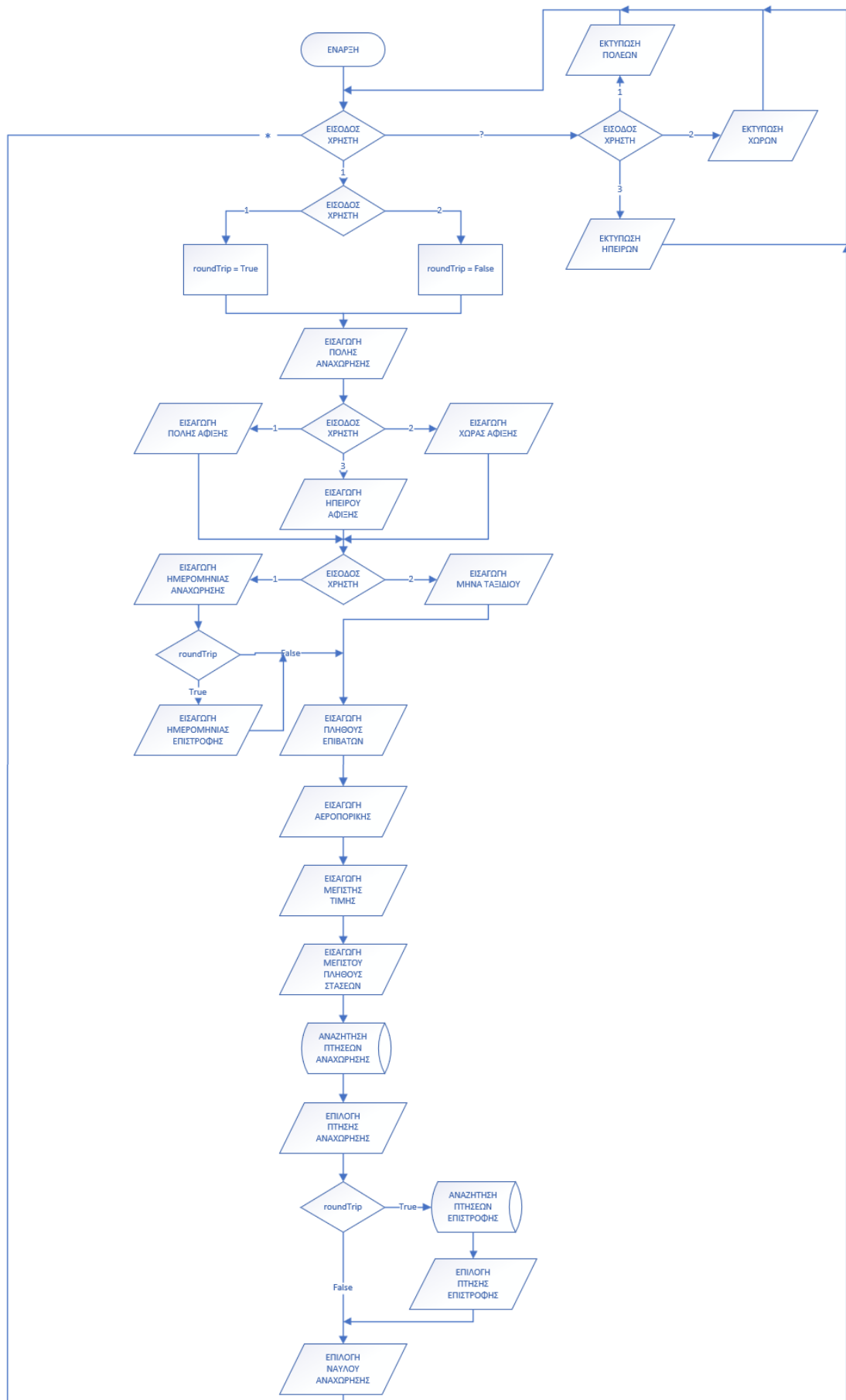
Η μέθοδος `getAvailableSeats` υλοποιεί το query εύρεσης των ελεύθερων θέσεων μίας συγκεκριμένης πτήσης, δοθέντος του αριθμού πτήσης και της ημερομηνίας αναχώρησης (του πρωτεύοντος κλειδιού του πίνακα *FLIGHT* δηλαδή), όπως αναλύθηκε στην προηγούμενη ενότητα.

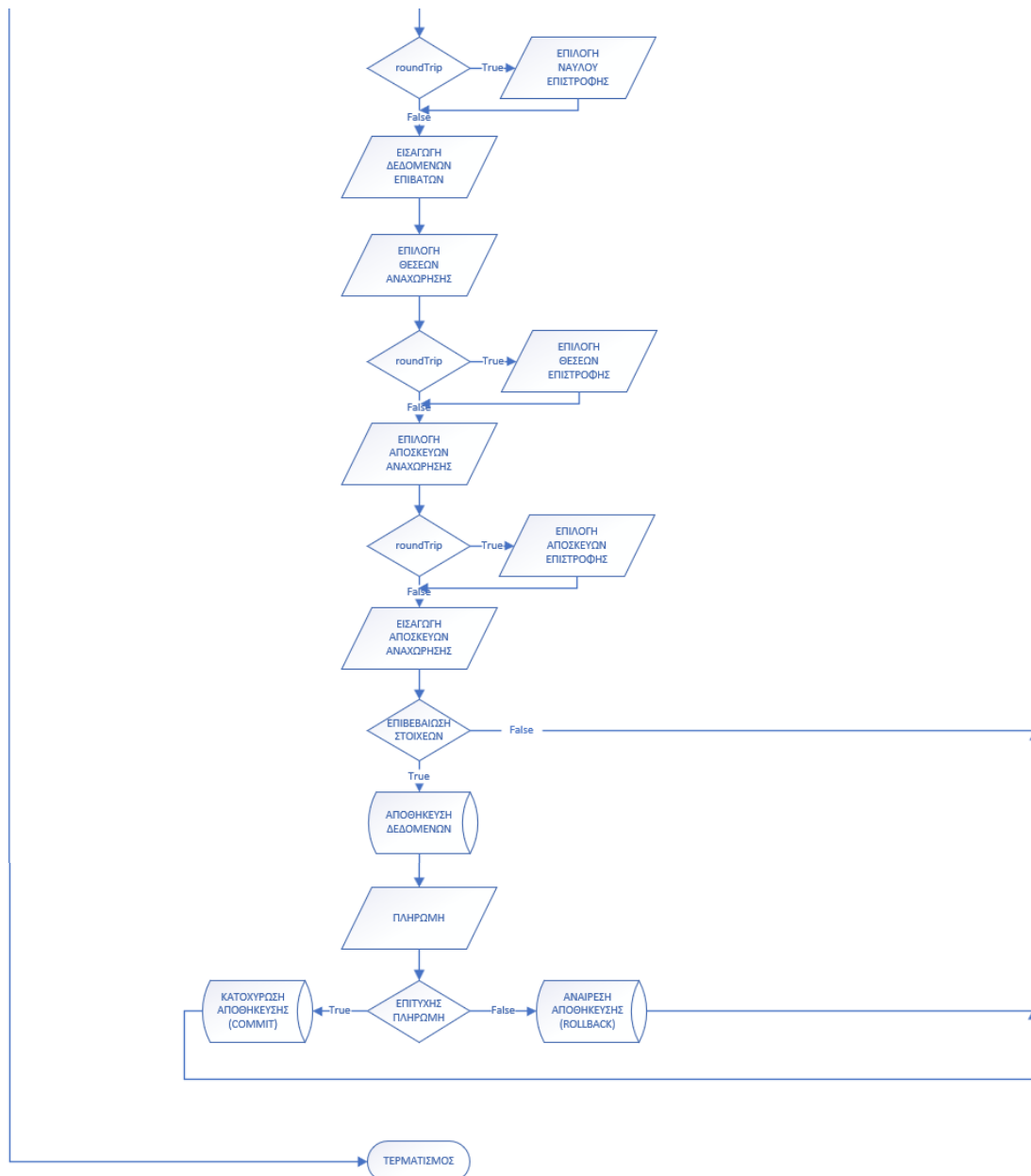
Η μέθοδος `calculateTicketPrice` υλοποιεί το query υπολογισμού της τιμής ενός εισιτηρίου με τον τρόπο που αναλύθηκε στην προηγούμενη ενότητα. Αξίζει να αναφερθεί πως το query string χτίζεται σταδιακά και πλήρως παραμετροποιημένα, ανάλογα με την ηλικία του χρήστη, τις αποσκευές που αυτός έχει επιλέξει, τον ναύλο κ.λπ.

Οι μέθοδοι `savePassenger`, `saveBooking`, `saveTicket`, `saveTakenSeat`, `saveLuggage` υλοποιούν τις εντολές `INSERT` στους αντίστοιχους πίνακες της βάσης δεδομένων και εκτελούνται στο τελικό στάδιο της κράτησης, όταν ο χρήστης έχει εκτελέσει όλα τα βήματα με επιτυχία.

Η κλάση `NavigatorService`

Η κλάση NavigatorService υλοποιεί όλες τις απαιτούμενες μεθόδους για την καθοδήγηση του χρήστη στη διαδικασία από την αναζήτηση πτήσεων έως και την ολοκλήρωση της κράτησης εισιτηρίων. Παρακάτω φαίνεται αφαιρετικά το διάγραμμα ροής του προγράμματος, ανάλογα με τις επιλογές του χρήστη.





Εικόνα 3. Το διάγραμμα ροής του προγράμματος.

Αξίζει να σημειωθεί πως καθ' όλη τη διαδικασία της αναζήτησης πτήσεων και κράτησης εισιτηρίων, έχουν εφαρμοστεί πρακτικές αμυντικού προγραμματισμού, προκειμένου να γίνει χειρισμός κάθε περίπτωσης εισόδων του χρήστη.

3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΠΟ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Παρακάτω φαίνεται ένα παράδειγμα εκτέλεσης της εφαρμογής, για μία τυπική διαδικασία αναζήτησης πτήσεων και κράτησης εισιτηρίων από έναν χρήστη, για το ακόλουθο σενάριο:

«Επιθυμώ να πραγματοποιήσω ένα ταξίδι μετ' επιστροφής από την Αθήνα στο Μόναχο, με ημερομηνία αναχώρησης την 9/3/2023 και ημερομηνία επιστροφής την 14/3/2023, για 2 ενήλικες, 1 παιδί και 1 βρέφος, κάνοντας έως 2 στάσεις, με ναύλο Economy ComfortFlex.»

```
-----
WELCOME TO FlyNow!
-----

Main menu options:
- Press 1 to start planning a new trip.
- Press ? for help.
- Press * to exit.
Your choice: █
```

Το κύριο μενού της εφαρμογής.

```
Your choice: 1

New trip menu options:
- Press 1 to begin a round trip.
- Press 2 to begin an one-way trip.
- Press * to return to main menu.
Your choice: █
```

Έναρξη σχεδιασμού ταξιδιού. Επιλογή ταξιδιού μετ' επιστροφής.

```
Your choice: 1

Where do you want to travel from?
Enter a city: █
```

Εισαγωγή πόλης αναχώρησης.

```
Enter a city: Athens
Athens selected.

Where do you want to travel to?
- Press 1 to select a city.
- Press 2 to select a country.
- Press 3 to select a region.
- Press * to return to main menu.
Your choice: █
```

Μενού επιλογής προορισμού. Επιλογή πόλης.

```
Your choice: 1
Enter a city: █
```

Εισαγωγή πόλης προορισμού.

```
Enter a city: Munich
Munich selected.

When do you want to travel?
- Press 1 to select specific dates.
- Press 2 to select a whole month.
- Press * to return to main menu.
Your choice: █
```

Μενού επιλογής ημερομηνιών ταξιδιού. Επιλογή συγκεκριμένων ημερομηνιών.

Your choice: 1
When do you want to depart?
Enter a date in the format YYYY/MM/DD:

Εισαγωγή ημερομηνίας αναχώρησης.

Enter a date in the format YYYY/MM/DD: 2023/03/09
When do you want to return?
Enter a date in the format YYYY/MM/DD:

Εισαγωγή ημερομηνίας επιστροφής.

Enter a date in the format YYYY/MM/DD: 2023/03/14
How many adults (12 y.o. or older) are going to travel? 2
How many children (2-11 y.o.) are going to travel? 1
How many infants (<2 y.o.) are going to travel? 1

Εισαγωγή πλήθους επιβατών.

Do you want to travel with a specific airline?
Enter the airline name, or leave the field blank to override this filter.
Your choice:

Καθορισμός φίλτρου αεροπορικής εταιρείας.

Do you want to set a maximum base price for your flight?
Enter the desired maximum price, or leave the field blank to override this filter.
Your choice:

Καθορισμός φίλτρου μέγιστης τιμής.

Do you want to set a maximum number of stops for your flight?
Enter the desired maximum number of stops, or leave the field blank to override this filter.
Your choice: 2






Καθορισμός φίλτρου μέγιστου αριθμού στάσεων.

Searching for flights that suit your preferences...
Searching for non-stop flights...
Found 3 non-stop flights.
Searching for one-stop flights...
Found 26 one-stop flights.
Searching for two-stops flights...
Found 221 two-stops flights.

Αναζήτηση πτήσεων αναχώρησης.

```
1.   ATH 2023-03-09 12:55 > 2023-03-09 14:30 MUC -- 270.88€  
    Direct (2h 35min) - Operated by Lufthansa  
    (LH1753)  
  
2.   ATH 2023-03-09 15:45 > 2023-03-09 17:25 MUC -- 286.87€  
    Direct (2h 40min) - Operated by Aegean Airlines  
    (A3806)  
  
3.   ATH 2023-03-09 08:30 > 2023-03-09 10:10 MUC -- 312.65€  
    Direct (2h 40min) - Operated by Aegean Airlines  
    (A3802)  
  
4.   ATH 2023-03-09 07:05 > 2023-03-09 11:20 MUC -- 342.75€  
    1 stop in ZRH (5h 15min) - Operated by Lufthansa and SWISS  
    (LX1843, LH2377)  
  
5.   ATH 2023-03-09 13:25 > 2023-03-09 18:30 MUC -- 387.32€  
    1 stop in VIE (6h 05min) - Operated by Austrian  
    (OS802, OS115)  
  
6.   ATH 2023-03-09 08:40 > 2023-03-09 13:55 MUC -- 382.12€  
    1 stop in VIE (6h 15min) - Operated by Aegean Airlines and Lufthansa  
    (A3860, LH2327)  
  
7.   ATH 2023-03-09 13:00 > 2023-03-09 18:30 MUC -- 449.59€  
    1 stop in MXP (6h 30min) - Operated by Lufthansa and Sky Express S.A.  
    (GQ830, LH1857)  
  
8.   ATH 2023-03-09 08:10 > 2023-03-09 13:40 MUC -- 516.95€  
    1 stop in IST (6h 30min) - Operated by THY - Turkish Airlines and Aegean Airlines  
    (A3990, TK1633)  
  
9.   ATH 2023-03-09 08:35 > 2023-03-09 14:40 MUC -- 594.78€  
    1 stop in CDG (7h 05min) - Operated by Aegean Airlines and Air France  
    (A3610, AF1622)  
  
10.  ATH 2023-03-09 08:40 > 2023-03-09 14:50 MUC -- 354.21€  
    1 stop in VIE (7h 10min) - Operated by Aegean Airlines and Austrian  
    (A3860, OS103)  
  
11.  ATH 2023-03-09 13:25 > 2023-03-09 19:35 MUC -- 683.90€  
    1 stop in LHR (7h 10min) - Operated by Aegean Airlines and Lufthansa  
    (A3602, LH2477)  
  
(Showing trips 1-11 out of 250)  
- Press a trip number to select it.  
- Press Enter to display next trips.  
- Press * to return to main menu.  
Your choice: █
```

Αποτελέσματα αναζήτησης πτήσεων αναχώρησης.

 <div> 3:45 PM ATH </div> <div> 2h 40m Non-stop </div> <div> 5:25 PM MUC </div>	11 deals from 47 € 185 € total Select →
<div>  Greener choice This flight emits 15% less CO₂ than the average for your search </div>	
 <div> 12:55 PM ATH </div> <div> 2h 35m Non-stop </div> <div> 2:30 PM MUC </div>	10 deals from 61 € 244 € total Select →
<div>  This flight emits 15% less CO₂ </div>	
 <div> 8:30 AM ATH </div> <div> 2h 40m Non-stop </div> <div> 10:10 AM MUC </div>	11 deals from 95 € 379 € total Select →

Επιβεβαίωση ορθότητας αποτελεσμάτων με σύγκριση με την εφαρμογή Skyscanner.

```
(Showing trips 1-11 out of 250)
- Press a trip number to select it.
- Press Enter to display next trips.
- Press * to return to main menu.
Your choice: 2
```

Επιλογή ταξιδιού αναχώρησης.

```
Searching for return flights that suit your preferences...
Searching for non-stop flights...
Found 2 non-stop flights.
Searching for one-stop flights...
Found 33 one-stop flights.
Searching for two-stops flights...
Found 259 two-stops flights.
```

Αναζήτηση πτήσεων επιστροφής.

1. MUC 2023-03-14 06:00 > 2023-03-14 09:25 ATH -- 253.02€
Direct (2h 25min) - Operated by Star Air
(DJ6118)
2. MUC 2023-03-14 08:40 > 2023-03-14 12:05 ATH -- 314.06€
Direct (2h 25min) - Operated by Lufthansa
(LH1752)
3. MUC 2023-03-14 15:35 > 2023-03-14 21:20 ATH -- 414.62€
1 stop in ZRH (4h 45min) - Operated by Lufthansa and SWISS
(LH2370, LX1842)
4. MUC 2023-03-14 08:55 > 2023-03-14 14:55 ATH -- 376.58€
1 stop in ZRH (5h 00min) - Operated by Aegean Airlines and Lufthansa
(LH2376, A3851)
5. MUC 2023-03-14 15:35 > 2023-03-14 21:35 ATH -- 405.06€
1 stop in VIE (5h 00min) - Operated by Aegean Airlines and Austrian
(OS104, A3863)
6. MUC 2023-03-14 09:00 > 2023-03-14 15:25 ATH -- 399.76€
1 stop in BER (5h 25min) - Operated by Aegean Airlines and Lufthansa
(LH1934, A3821)
7. MUC 2023-03-14 07:50 > 2023-03-14 14:55 ATH -- 387.34€
1 stop in ZRH (6h 05min) - Operated by AIR DOLOMITI S.p.A. LINEE AEREE REGIONALI EUROPEE and Aegean Airlines
(EN2366, A3851)
8. MUC 2023-03-14 06:50 > 2023-03-14 14:00 ATH -- 523.45€
1 stop in CDG (6h 10min) - Operated by Lufthansa and Air France
(LH2226, AF1532)
9. MUC 2023-03-14 08:00 > 2023-03-14 15:25 ATH -- 438.34€
1 stop in BER (6h 25min) - Operated by Aegean Airlines and Lufthansa
(LH1930, A3821)
10. MUC 2023-03-14 08:55 > 2023-03-14 16:25 ATH -- 603.76€
1 stop in CDG (6h 30min) - Operated by Aegean Airlines and Lufthansa
(LH2228, A3611)
11. MUC 2023-03-14 12:20 > 2023-03-14 19:50 ATH -- 609.59€
1 stop in AMS (6h 30min) - Operated by Lufthansa and Transavia Airlines
(LH2304, HV6867)

Αποτελέσματα αναζήτησης πτήσεων επιστροφής.

(Showing trips 1-11 out of 294)
- Press a trip number to select it.
- Press Enter to display next trips.
- Press * to return to main menu.
Your choice: 3

Επιλογή ταξιδιού επιστροφής.

Your choice: 3
Trip 3 selected.

Let's make your booking!

Step 1.1: Select a fare for your trip from ATH to MUC.
1. Economy Light -- 286.87€
2. Economy Flex -- 304.08€
3. Economy ComfortFlex -- 341.38€
4. Business -- 458.99€
- Press a fare number to select it.
- Press ? to view the amenities of each fare.
- Press * to return to main menu.
Your choice:

Επιλογή ναύλου για την πτήση αναχώρησης.

Your choice: 3

Step 1.2: Select a fare for your trip from MUC to ATH.
1. Economy Light -- 414.62€
2. Economy Flex -- 443.09€
3. Economy ComfortFlex -- 504.60€
4. Business -- 713.83€
- Press a fare number to select it.
- Press ? to view the amenities of each fare.
- Press * to return to main menu.
Your choice:

Επιλογή ναύλου για τις πτήσεις επιστροφής.

```
Your choice: 3
Step 2: Enter passenger information.
- Adult 1: First name: 
```

Εισαγωγή στοιχείων επιβατών.

```
Step 2: Enter passenger information.
- Adult 1: First name: Ιωάννης
- Adult 1: Last name: Παπαδόπουλος
- Adult 1: Identity card or Passport number: AK123456
- Adult 1: Age: 35
- Adult 1: Phone number: +306944596942
- Adult 1: E-mail: i.papadopoulos@gmail.com

- Adult 2: First name: Μαρία
- Adult 2: Last name: Αναστασίου
- Adult 2: Identity card or Passport number: AK234567
- Adult 2: Age: 33
- Adult 2: Phone number: +306973916940
- Adult 2: E-mail: maria.an@outlook.com

- Child 1: First name: Χαράλαμπος
- Child 1: Last name: Παπαδόπουλος
- Child 1: Identity card or Passport number: PI3456789
- Child 1: Age: 4

- Infant 1: First name: Νεφέλη
- Infant 1: Last name: Παπαδοπούλου
- Infant 1: Age: 1
```

Στοιχεία επιβατών.

```
Step 3.1:      Select seats for your trip from ATH to MUC.

Available seats for flight ATH-MUC:
Seat   Class
----   -
1A     Economy Class
1B     Economy Class
1C     Economy Class
1D     Economy Class
1E     Economy Class
1F     Economy Class
2A     Economy Class
2B     Economy Class
2C     Economy Class
2D     Economy Class
2E     Economy Class
2F     Economy Class
3A     Economy Class
3B     Economy Class
3C     Economy Class
3D     Economy Class
```

Εμφάνιση διαθέσιμων θέσεων για την πτήση αναχώρησης.

```
28F     Economy Class
29A     Economy Class
29B     Economy Class
29C     Economy Class
29D     Economy Class
29E     Economy Class
29F     Economy Class
Enter seat number for Ιωάννης Παπαδόπουλος for flight ATH-MUC: 
```

Επιλογή θέσεων επιβατών για την πτήση αναχώρησης.

```
Enter seat number for Ιωάννης Παπαδόπουλος for flight ATH-MUC: 8D
Enter seat number for Μαρία Αναστασίου for flight ATH-MUC: 8E
Enter seat number for Χαράλαμπος Παπαδόπουλος for flight ATH-MUC: 8F
Νεφέλη Παπαδοπούλου is infant, so they will sit with an adult.
```

Τα βρέφη κάθονται με κάποιον ενήλικα.

Step 3.2: Select seats for your trip from MUC to ATH.

Available seats for flight MUC-ZRH:

Seat	Class
1C	Business Class
1D	Business Class
2A	Business Class
2F	Business Class
3C	Business Class
3D	Business Class
4A	Business Class
4F	Business Class
5A	Economy Class
5C	Economy Class
5D	Economy Class
5F	Economy Class
6A	Economy Class
6C	Economy Class

Εμφάνιση διαθέσιμων θέσεων για την 1η πτήση επιστροφής.

Enter seat number for Ιωάννης Παπαδόπουλος for flight MUC-ZRH: 28A
Enter seat number for Μαρία Αναστασίου for flight MUC-ZRH: 28C
Enter seat number for Χαράλαμπος Παπαδόπουλος for flight MUC-ZRH: 28D
Νεφέλη Παπαδοπούλου is infant, so they will sit with an adult.

Επιλογή θέσεων επιβατών για την 1η πτήση επιστροφής.

Enter seat number for Ιωάννης Παπαδόπουλος for flight ZRH-ATH: 21A
Enter seat number for Μαρία Αναστασίου for flight ZRH-ATH: 21B
Enter seat number for Χαράλαμπος Παπαδόπουλος for flight ZRH-ATH: 21C
Νεφέλη Παπαδοπούλου is infant, so they will sit with an adult.

Εμφάνιση διαθέσιμων θέσεων και επιλογή θέσεων επιβατών για την 2η πτήση επιστροφής.

Step 4.1: Add luggage for your trip from ATH to MUC.

Luggage policy for flight ATH-MUC, for fare Economy ComfortFlex:
You have one free carry-on bag up to 8kg.
You have one free checked baggage up to 23kg.

Extra luggage costs for flight ATH-MUC:

Carry-on baggage up to 8kg:	14€
Checked baggage up to 23kg:	27€
Checked baggage up to 32kg:	46€

Does Ιωάννης Παπαδόπουλος want extra luggage for flight ATH-MUC?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

Δωρεάν αποσκευές για τον επιλεγμένο ναύλο και κόστη επιπλέον αποσκευών για την πτήση αναχώρησης.

Does Ιωάννης Παπαδόπουλος want extra luggage for flight ATH-MUC?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

Does Μαρία Αναστασίου want extra luggage for flight ATH-MUC?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

Does Χαράλαμπος Παπαδόπουλος want extra luggage for flight ATH-MUC?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

Προσθήκη επιπλέον, επί πληρωμή αποσκευών για την πτήση αναχώρησης.

```

Step 4.2:      Add luggage for your trip from MUC to ATH.

Luggage policy for flight MUC-ZRH, for fare Economy ComfortFlex:
You have one free carry-on bag up to 8kg.

Extra luggage costs for flight MUC-ZRH:
-----
Carry-on baggage up to 8kg:      13€
Checked baggage up to 23kg:      22€
Checked baggage up to 32kg:      38€

Luggage policy for flight ZRH-ATH, for fare Economy ComfortFlex:
You have one free carry-on bag up to 8kg.

Extra luggage costs for flight ZRH-ATH:
-----
Carry-on baggage up to 8kg:      13€
Checked baggage up to 23kg:      24€
Checked baggage up to 32kg:      41€

Does Ιωάννης Παπαδόπουλος want extra luggage for flight MUC-ZRH?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice: 

```

Δωρεάν αποσκευές για τον επιλεγμένο ναύλο και κόστη επιπλέον αποσκευών για την πτήση επιστροφής.

```

Does Ιωάννης Παπαδόπουλος want extra luggage for flight MUC-ZRH?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

Does Ιωάννης Παπαδόπουλος want extra luggage for flight ZRH-ATH?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

Does Μαρία Αναστασίου want extra luggage for flight MUC-ZRH?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice: 23
A 23kg luggage piece was added for Μαρία Αναστασίου for 22€.
Your choice:

Does Μαρία Αναστασίου want extra luggage for flight ZRH-ATH?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice: 23
A 23kg luggage piece was added for Μαρία Αναστασίου for 24€.
Your choice:

Does Χαράλαμπος Παπαδόπουλος want extra luggage for flight MUC-ZRH?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

Does Χαράλαμπος Παπαδόπουλος want extra luggage for flight ZRH-ATH?
Enter the desired luggage weight to add, or leave the field blank to continue.
Your choice:

```

Προσθήκη επιπλέον, επί πληρωμή αποσκευών για την πτήση επιστροφής.

Almost done! Here is a summary of your booking:

Your trip from ATH to MUC:

Direct flight

- Departs at: Thursday, 09 March 2023 03:45 PM local time
- Arrives at: Thursday, 09 March 2023 05:25 PM local time
- Duration: 2h 40min

A3806 by Aegean Airlines

Fare: Economy ComfortFlex

Passengers:

- Ιωάννης Παπαδόπουλος (Adult)
Identity card or Passport number: AK123456
Age: 35
Phone number: +306944596942
E-mail: i.papadopoulos@gmail.com
- Μαρία Αναστασίου (Adult)
Identity card or Passport number: AK234567
Age: 33
Phone number: +306973916940
E-mail: maria.an@outlook.com
- Χαράλαμπος Παπαδόπουλος (Child)
Identity card or Passport number: PI3456789
Age: 4
- Νεφέλη Παπαδοπούλου (Infant)
Age: 1

Selected seats:

- Ιωάννης Παπαδόπουλος: Seat 8D
- Μαρία Αναστασίου: Seat 8E
- Χαράλαμπος Παπαδόπουλος: Seat 8F
- Νεφέλη Παπαδοπούλου is infant, so they will sit with an adult.

Selected luggage pieces:

- Ιωάννης Παπαδόπουλος:
 - 1 Carry-on luggage piece up to 8kg - 0€
 - 1 Checked luggage piece up to 23kg - 0€
- Μαρία Αναστασίου:
 - 1 Carry-on luggage piece up to 8kg - 0€
 - 1 Checked luggage piece up to 23kg - 0€
- Χαράλαμπος Παπαδόπουλος:
 - 1 Carry-on luggage piece up to 8kg - 0€
 - 1 Checked luggage piece up to 23kg - 0€

Επιβεβαίωση στοιχείων για την πτήση αναχώρησης.

```

Your trip from MUC to ATH:
One stop in ZRH
- Departs at: Tuesday, 14 March 2023 03:35 PM local time
- Arrives at: Tuesday, 14 March 2023 09:20 PM local time
- Duration: 4h 45min
LH2370 by Lufthansa, LX1842 by SWISS

Fare: Economy ComfortFlex

Selected seats for flight MUC-ZRH:
- Ιωάννης Παπαδόπουλος: Seat 28A
- Μαρία Αναστασίου: Seat 28C
- Χαράλαμπος Παπαδόπουλος: Seat 28D
- Νεφέλη Παπαδοπούλου is infant, so they will sit with an adult.

Selected seats for flight ZRH-ATH:
- Ιωάννης Παπαδόπουλος: Seat 21A
- Μαρία Αναστασίου: Seat 21B
- Χαράλαμπος Παπαδόπουλος: Seat 21C
- Νεφέλη Παπαδοπούλου is infant, so they will sit with an adult.

Selected luggage pieces for flight MUC-ZRH:
- Ιωάννης Παπαδόπουλος:
  - 1 Carry-on luggage piece up to 8kg - 0€
- Μαρία Αναστασίου:
  - 1 Carry-on luggage piece up to 8kg - 0€
  - 1 Checked luggage piece up to 23kg - 22€
- Χαράλαμπος Παπαδόπουλος:
  - 1 Carry-on luggage piece up to 8kg - 0€

Selected luggage pieces for flight ZRH-ATH:
- Ιωάννης Παπαδόπουλος:
  - 1 Carry-on luggage piece up to 8kg - 0€
- Μαρία Αναστασίου:
  - 1 Carry-on luggage piece up to 8kg - 0€
  - 1 Checked luggage piece up to 23kg - 24€
- Χαράλαμπος Παπαδόπουλος:
  - 1 Carry-on luggage piece up to 8kg - 0€

Total price: 891.97€

Is this information correct? Enter Yes to confirm, or No to cancel your booking and start over.
Your choice: █

```

Επιβεβαίωση στοιχείων για τις πτήσεις επιστροφής.

```

Is this information correct? Enter Yes to confirm, or No to cancel your booking and start over.
Your choice: Yes

Redirecting to the bank environment...

Payment complete!

You have successfully made your booking! Opening the booking confirmation...

Returning to main menu...

```

Ολοκλήρωση πληρωμής και επιτυχής κράτηση εισιτηρίων.

17	AK123456	Ιωάννης	Παπαδόπουλος	35	+306944596942	i.papadopoulos@gmail.com
18	AK234567	Μαρία	Αναστασίου	33	+306973916940	maria.an@outlook.com
19	PI3456789	Χαράλαμπος	Παπαδόπουλος	4	NULL	NULL
20	AK123456_INF	Νεφέλη	Παπαδοπούλου	1	NULL	NULL

Επιβεβαίωση εισαγωγής δεδομένων στον πίνακα PASSENGER.

25	08D	A3806	2023-03-09 13:45:00
26	08E	A3806	2023-03-09 13:45:00
27	08F	A3806	2023-03-09 13:45:00
28	28A	LH2370	2023-03-14 14:35:00
29	28C	LH2370	2023-03-14 14:35:00
30	28D	LH2370	2023-03-14 14:35:00
31	21A	LX1842	2023-03-14 16:45:00
32	21B	LX1842	2023-03-14 16:45:00
33	21C	LX1842	2023-03-14 16:45:00

Επιβεβαίωση εισαγωγής δεδομένων στον πίνακα TAKEN SEAT.

32	2023010820062254528401	123.69	08D	A3806	2023-03-09 13:45:00	AK123456	20230108200622545284	Economy ComfortFlex	AEE
33	2023010820062254528402	123.69	08E	A3806	2023-03-09 13:45:00	AK234567	20230108200622545284	Economy ComfortFlex	AEE
34	2023010820062254528403	81.63	08F	A3806	2023-03-09 13:45:00	PI3456789	20230108200622545284	Economy ComfortFlex	AEE
35	2023010820062254528404	12.37	08D	A3806	2023-03-09 13:45:00	AK123456_INF	20230108200622545284	Economy ComfortFlex	AEE
36	2023010820062254528405	38.48	28A	LH2370	2023-03-14 14:35:00	AK123456	20230108200622545284	Economy ComfortFlex	DLH
37	2023010820062254528406	60.48	28C	LH2370	2023-03-14 14:35:00	AK234567	20230108200622545284	Economy ComfortFlex	DLH
38	2023010820062254528407	21.93	28D	LH2370	2023-03-14 14:35:00	PI3456789	20230108200622545284	Economy ComfortFlex	DLH
39	2023010820062254528408	6.16	28A	LH2370	2023-03-14 14:35:00	AK123456_INF	20230108200622545284	Economy ComfortFlex	DLH
40	2023010820062254528409	152.5	21A	LX1842	2023-03-14 16:45:00	AK123456	20230108200622545284	Economy ComfortFlex	SWR
41	2023010820062254528410	176.5	21B	LX1842	2023-03-14 16:45:00	AK234567	20230108200622545284	Economy ComfortFlex	SWR
42	2023010820062254528411	80.82	21C	LX1842	2023-03-14 16:45:00	PI3456789	20230108200622545284	Economy ComfortFlex	SWR
43	2023010820062254528412	13.72	21C	LX1842	2023-03-14 16:45:00	AK123456_INF	20230108200622545284	Economy ComfortFlex	SWR

Επιβεβαίωση εισαγωγής δεδομένων στον πίνακα TICKET.

7	20230108200622545284	2023-01-08 20:06:22	891.97	AK123456
---	----------------------	---------------------	--------	----------

Επιβεβαίωση εισαγωγής δεδομένων στον πίνακα BOOKING.

4 ΑΝΑΦΟΡΕΣ

[1] Ν. Αβούρης. Ενότητα 2: Εισαγωγή στο σχεσιακό μοντέλο, μετάβαση από ERD σε relational schema

[2] Ν. Αβούρης. Ενότητα 7: Κανονικοποίηση, Δοσοληψίες, Ασφάλεια