

Εργασία μαθήματος Βάσεις Δεδομένων 2023-2024

Δημιουργία εφαρμογής αναζήτησης/κράτησης αεροπορικών εισιτηρίων.

Ηλίας Ουζούνης

Προπτυχιακός φοιτητής τμήματος HMTY, up1083749@ac.upatras.gr

Κωνσταντίνος Τσάμπρας

Προπτυχιακός φοιτητής τμήματος HMTY, up1083749@ac.upatras.gr

Σε αυτήν την έκθεση περιγράφουμε τη σχεδίαση και υλοποίηση μιας εφαρμογής κράτησης και αναζήτησης αεροπορικών εισιτηρίων με χρήση μιας βάσης δεδομένων στα πλαίσια της διδασκαλίας του μαθήματος Βάσεις Δεδομένων 2023-2024

1 ΠΕΡΙΛΗΨΗ

Η εφαρμογή που σχεδιάσαμε είχε ως σκοπό να πληροί της προϋποθέσεις μιας σύγχρονης εφαρμογής διαχείρισης αεροπορικών εισιτηρίων, και να ανταποκρίνεται σε μεγάλο όγκο δεδομένων, όπως επιβάλλει η ευρεία διάδοση των αεροπορικών μέσων ως μέσα μεταφοράς στην σύγχρονη κοινωνία. Πιο συγκεκριμένα, σκοπός μας ήταν να ενσωματώσουμε λειτουργίες όπως

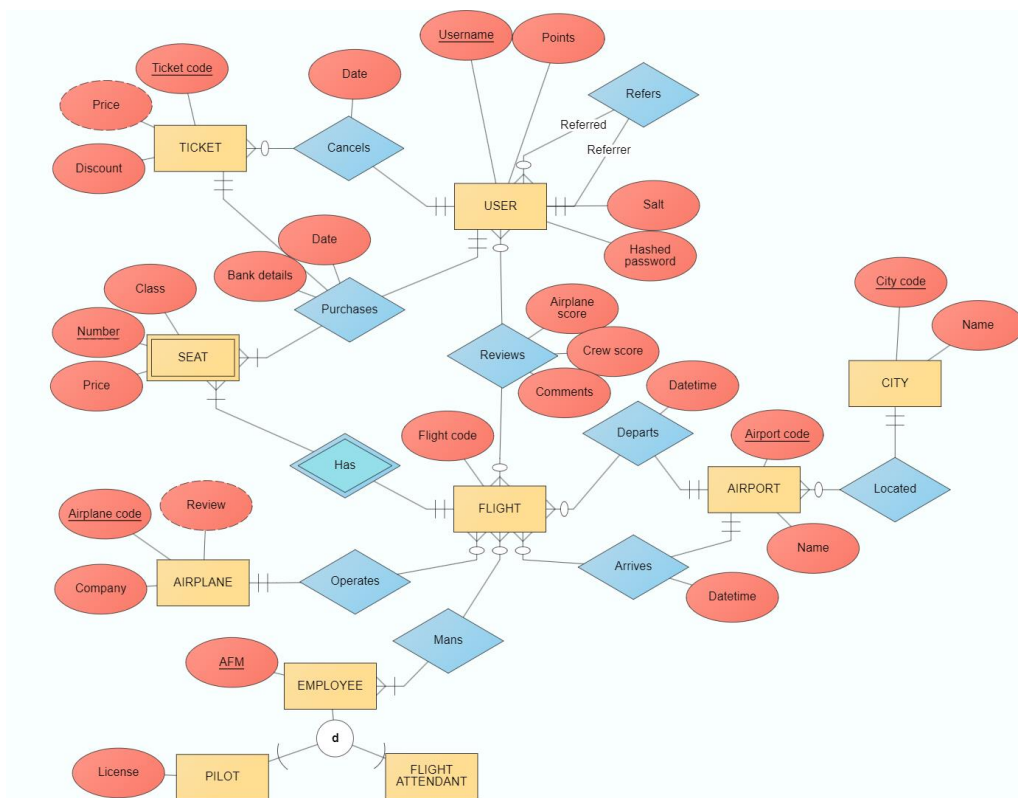
- Κράτηση εισιτηρίων (με ανταπόκριση αν χρειάζεται)
- Ακύρωση εισιτηρίων
- Προβολή επόμενης προγραμματισμένης πτήσης
- Αξιολόγηση εμπειρίας πτήσης (πληρώματος και αεροπλάνου)
- Δυνατότητα στους admins να βλέπουν πλήθος στατιστικών στοιχείων για την βάση.

2 ΜΕΘΟΔΟΛΟΓΙΑ

Ακολουθούν τα βήματα που ακολουθήσαμε για να υλοποιήσουμε την εφαρμογή μας.

2.1 ERD

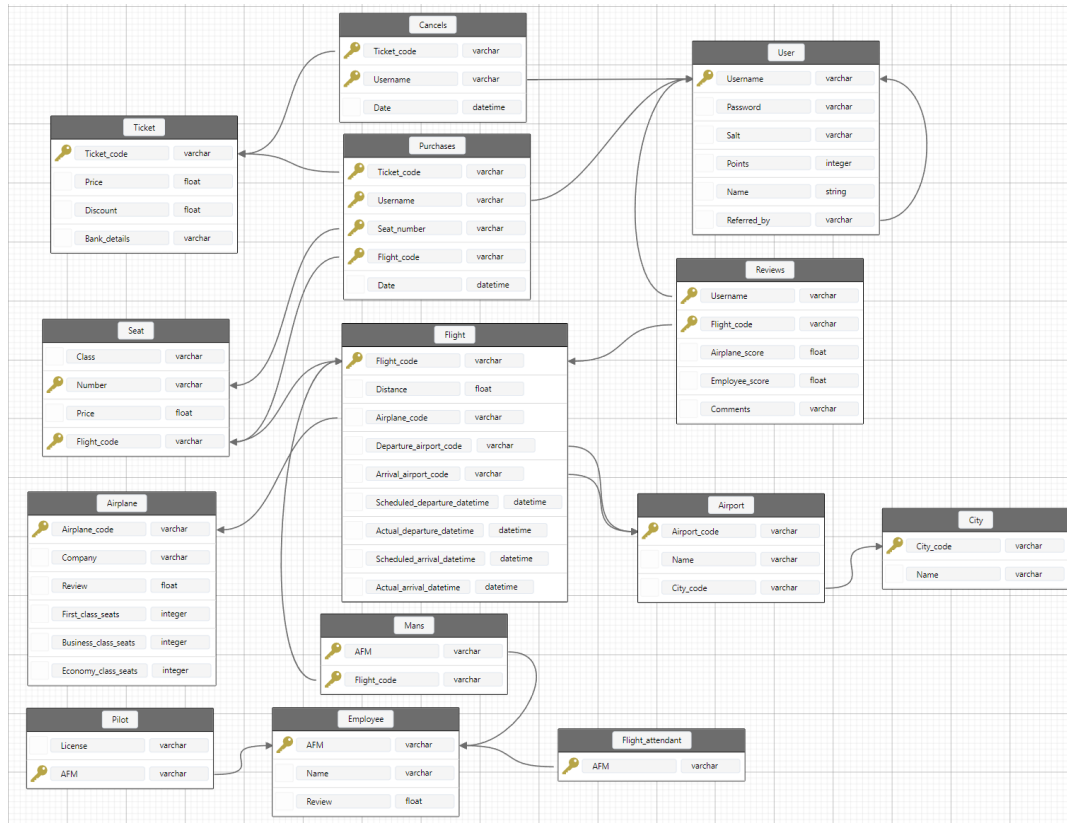
Αφού συλλέξαμε τις λειτουργίες που θέλουμε να υπηρετεί η εφαρμογή μας έπρεπε να σχεδιάσουμε την μορφή της βάσης δεδομένων μας. Η σχεδίαση αυτή έγινε με το εργαλείο erdmaker του τμήματος HMTY (<https://hci.ece.upatras.gr/erdmaker>):



Εικόνα 1. Το ERD Διάγραμμα της βάσης δεδομένων μας

2.2 SCHEMA

Στην συνέχεια έπρεπε να μεταφράσουμε την μορφή αυτή σε σχήμα βάσης δεδομένων, το οποίο έγινε με την βοήθεια του εργαλείου schemamaker του τμήματος HMTY (https://schemamaker.fly.dev/schema_builder):



Εικόνα 2. Το σχήμα της βάσης δεδομένων μας

2.3 SQL (Δημιουργία βάσης)

Επιλέξαμε να χρησιμοποιήσουμε την SQLite για την βάση μας, οπότε το επόμενο βήμα ήταν να μετατρέψουμε το σχήμα της βάσης σε εντολές δημιουργίας της βάσης δεδομένων της εφαρμογής μας. Το εργαλείο schemamaker αυτοματοποιεί αυτήν την διαδικασία.

2.4 Δημιουργία Mockup βάσης

Το επόμενο λογικό βήμα είναι να γεμίσουμε την βάση με μικρό πλήθος δεδομένων ώστε να μπορούμε να ελέγξουμε αν τα queries μας συμπεριφέρονται όπως πρέπει.

2.5 Δημιουργία Queries

Στην συνέχεια δημιουργήσαμε queries μέσω των οποίων θα επικοινωνεί η εφαρμογή μας με την βάση δεδομένων. Κάθε query πρέπει να ελεγχθεί για να διασφαλιστεί ότι λειτουργεί σωστά, τόσο σε αυτό το βήμα όσο και σε επόμενα όπου θα έχουμε μια μεγαλύτερη βάση δεδομένων. Για την εφαρμογή μας επιλέξαμε την χρήση της γλώσσας Python και της βιβλιοθήκης sqlite3. Άρα τα queries γίνονται μέσω συναρτήσεων της γλώσσας Python.

2.6 Δημιουργία εφαρμογής

Επόμενο βήμα είναι η δημιουργία της ίδιας της εφαρμογής, όπου υλοποιείται η διεπαφή με τον χρήστη και η επικοινωνία με την βάση μέσω των παραπάνω συναρτήσεων.

2.7 Δημιουργία βάσης μεγάλου μεγέθους

Όπως αναφέραμε έχουμε ως προαπαιτούμενο η βάση μας να χειρίζεται μεγάλο πλήθος δεδομένων, οπότε για να ελέγξουμε αυτό, καθώς και την σωστή λειτουργία της βάσης σε τυχαία δεδομένα, φτιάξαμε ένα πρόγραμμα δημιουργίας ψευδοτυχαίων δεδομένων με τα οποία γεμίσαμε την βάση μας.

2.8 Δημιουργία βασικής γραφικής διεπαφής

Αποφασίσαμε να δημιουργήσουμε έστω μια βασική γραφική διεπαφή για απλές λειτουργίες όπως η αγορά απλών εισιτηρίων, η ακύρωση εισιτηρίων και η αξιολόγηση της τελευταίας πτήσης. Πιο σύνθετες λειτουργίες όπως η αγορά εισιτηρίων με ανταπόκριση (όπου δεν γίνεται απευθείας) παραμένει ως λειτουργία μόνο μέσω γραμμής εντολών.

3 ΑΞΙΟΛΟΓΗΣΗ

Τα κριτήρια με βάση τα οποία κρίναμε την εφαρμογή μας.

3.1 Ορθότητα

Το πρώτο και μη διαπραγματεύσιμο κριτήριο είναι να μην υπάρχουν λάθη στην λειτουργία της εφαρμογής. Κάθε λειτουργία πρέπει να λειτουργεί σωστά και να εκτελεί ακριβώς τον σκοπό της, κάτι το οποίο έχουμε διαπιστώσει για τις λειτουργίες της εφαρμογής, ελέγχοντας τα δεδομένα στην βάση μετά από εκτελέσεις των λειτουργιών (πχ αν αποθηκεύτηκαν σωστά τα δεδομένα ενός χρήστη/μια αγοράς).

3.2 Ασφάλεια

Μπορεί η εφαρμογή μας να τρέχει τοπικά και μην έχει πολλούς πραγματικούς χρήστες αλλά, παρόλα αυτά πρέπει να διασφαλίσουμε ότι τα credentials του κάθε χρήστη παραμένουν ασφαλή σε περίπτωση διάρρευσης των δεδομένων της εφαρμογής. Πιο συγκεκριμένα ακόμα και αν διαρρεύσουν τα περιεχόμενα της βάσης δεδομένων δεν πρέπει να επιτρέψουμε σε ένα κακόβουλο πρόσωπο να κλέψει τους κωδικούς άλλου χρήστη και να συνδεθεί ως αυτός προκαλώντας χάος.

Για να το πετύχουμε αυτό χρησιμοποιήσαμε hashing των κωδικών πρόσβασης των χρηστών ως αποτέλεσμα η βάση να μην περιέχει αυτούσιους τους κωδικούς των χρηστών, αλλά μόνο τα hashes αυτών. Επιπροσθέτως, για να αποφύγουμε και την περίπτωση χρήσης προϋπολογισμένων hashes γνωστών κωδικών από κακόβουλα τρίτα πρόσωπα, χρησιμοποιούμε hash με χρήση τυχαίου προθέματος (salt) ρυθμιζόμενης ταχύτητας (αργές ταχύτητες hashing περιορίζουν τις επιθέσεις brute force) το οποίο αποθηκεύεται και αυτό στην βάση δεδομένων, όπως ο hashed κωδικός. Για τον σκοπό αυτό χρησιμοποιήθηκε η βιβλιοθήκη bcrypt της Python.

3.3 Ταχύτητα

Όπως αναφέρθηκε προηγουμένως, μια βάση δεδομένων με τόσα πολλά δεδομένα είναι απαραίτητο να παράγει αποτελέσματα σε λογικά χρονικά διαστήματα. Έτσι είναι απαραίτητο να χρησιμοποιήσουμε indexes για την οργάνωση των δεδομένων της βάσης και επιτάχυνση των queries, ειδικά σε πίνακες μεγάλου μεγέθους που συμμετέχουν σε σύνθετα queries.

Ενδεικτικά παρατίθενται οι χρόνοι εκτέλεσης των στατιστικών για την μεγάλη βάση δεδομένων (δηλαδή σε πίνακες με έως και 9 εκατομμύρια εγγραφές):

1. Popular airports	: 32.0 milliseconds
2. Well reviewed airplanes	: 972.4 milliseconds
3. Well reviewed crew	: 15.1 seconds
4. Popular days	: 48.0 milliseconds
5. Users with most purchases	: 22.9 seconds
6. Users with most referrals	: 7.1 milliseconds
7. Users with most points	: 1.5 milliseconds
8. Most cancelled flights	: 23.7 seconds
9. Example flight between cities	: 21.0 milliseconds

Εικόνα 3. Χρόνος εκτέλεσης στατιστικών Queries

4 ΔΕΔΟΜΕΝΑ

Όπως αναφέρθηκε και προηγουμένως, εκτός από μια μικρή δοκιμαστική βάση στην οποία τα δεδομένα εισήχθησαν χειροκίνητα, χρειάστηκε και η δημιουργία μια μεγάλης βάσεις για να δοκιμάσουμε την ταχύτητα και την ορθότητα των λειτουργιών.

Για τον σκοπό αυτόν έγινε χρήση της βιβλιοθήκης της Python Faker καθώς και της βιβλιοθήκης numpy.random. Η δημιουργία των τέτοιου πλήθους δεδομένων είναι μια σύνθετη διαδικασία αφού πρέπει να τηρούνται όλοι οι κανόνες της βάσης δεδομένων, καθώς και να δημιουργούνται σε ένα εύλογο χρονικό διάστημα. (Η δημιουργία των ψευδοτυχαίων δεδομένων γίνεται στο αρχείο random_data_generator.py). Ενδεικτικά δημιουργήθηκαν περίπου:

Οντότητα	Χρήστες	Υπάλληλοι	Αεροπλάνα	Πτήσεις	Θέσεις	Κρατήσεις
Πλήθος	10.000	10.000	1.000	60.000	9.000.000	1.500.000

5 ΔΙΑΜΟΙΡΑΣΜΟΣ ΕΡΓΑΣΙΩΝ

Όπως είδαμε στην ενότητα 2, η εργασία αποτελείται από πολλά μέρη. Μερικά από αυτά έγιναν από κοινού από τους δύο συμμετέχοντες ενώ άλλα από κυρίως τον έναν από τους δύο. Πάντα βέβαια υπήρχε συνεννόηση και συνεργασία όπου απαιτείτο. Πιο συγκεκριμένα:

- 2.1 (ERD) δημιουργήθηκε από κοινού
- 2.2 (Schema) δημιουργήθηκε από κοινού
- 2.3 (SQL-Δημιουργία βάσης) αυτόματη διαδικασία
- 2.4 (Δημιουργία Mockup βάσης) δημιουργήθηκε από τον κ. Τσάμπρα
- 2.5 (Δημιουργία Queries) γράφτηκαν και από τους δύο συμμετέχοντες)
- 2.6 (Δημιουργία εφαρμογής) υλοποιήθηκε από τον κ. Τσάμπρα
- 2.7 (Δημιουργία βάσης μεγάλου μεγέθους) το αρχείο δημιουργήθηκε από τον κ. Ουζούνη με μικρή βοήθεια από τον κ. Τσάμπρα
- 2.8 (δημιουργία βασικής γραφικής διεπαφής) υλοποιήθηκε από τον κ. Ουζούνη)

6 ΧΡΟΝΟΔΙΑΓΓΡΑΜΜΑ

Όπως και παραπάνω, θα χρησιμοποιήσουμε τους προαναφερθέντες στόχους για τη δημιουργία του χρονοδιαγράμματος.

Πιο συγκεκριμένα:

- 2.1 (ERD) Αρχές Νοεμβρίου
- 2.2 (Schema) Αρχές Νοεμβρίου
- 2.3 (SQL-Δημιουργία βάσης) Τέλη Δεκεμβρίου
- 2.4 (Δημιουργία Mockup βάσης) Τέλη Δεκεμβρίου
- 2.5 (Δημιουργία Queries) Διακοπές Χριστουγέννων
- 2.6 (Δημιουργία εφαρμογής) Αρχές Ιανουαρίου
- 2.7 (Δημιουργία βάσης μεγάλου μεγέθους) 7-10 Ιανουαρίου
- 2.8 (δημιουργία βασικής γραφικής διεπαφής) Αρχές Ιανουαρίου
- Η συγγραφή της έκθεσης και της παρουσίασης έγιναν μεταξύ 10 και 14 Ιανουαρίου

Βεβαίως έγιναν αλλαγές στα αρχικά στάδια (ERD, schema, Queries) ανάλογα με τις ανάγκες που συναντούσαμε στην διάρκεια του υλοποίησης της εφαρμογής.

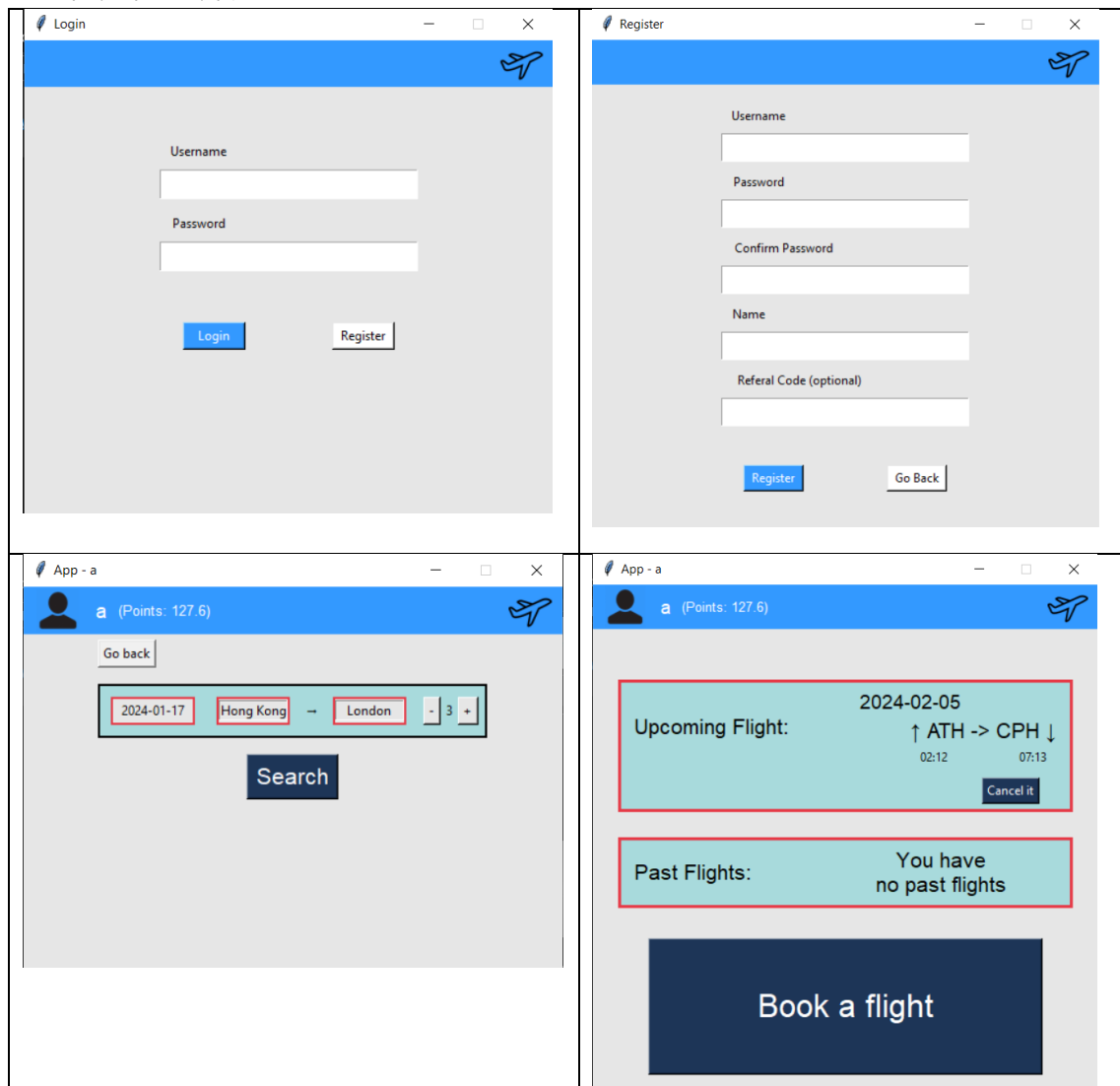
7 ΠΑΡΑΡΤΗΜΑΤΑ

7.1 Οδηγίες εγκατάστασης, αποθετήριο κώδικα

Οδηγίες εγκατάστασης υπάρχουν στο αρχείο [README](#), όπου βρίσκεται και ο κώδικας της εφαρμογής.

7.2 Παραδείγματα χρήσης

7.2.1 Γραφική διεπαφή (main_gui.py)



7.2.2 Διαπαθή κονσόλας (user.py)

<pre>Welcome to the app, select an option: 1. Login 2. Register Enter your choice: 2 Enter your username and password Username: username1 Password: password1 Confirm password: password1 Registration successful Enter your name Name: Nikos Were you referred by another user? (y/n): n You have 0 points! Select an option: 1. Book a flight 2. Leave a review for your last flight 3. Exit Enter your choice: 1 Enter your bank details Bank details: 2145-2145 2145-2145 Enter the date you want to fly Date: 2024-01-19 enter the city of departure City of Departure: /nagv enter the city of arrival City of Arrival: /lhw No flights available, checking for connecting flights Available connecting flights: Option 0: (intermediate city: Cape Town)) Flight Code: 2291 Crew Score: 1.17 Plane Score: 1.66 Number of Seats: First Class: 7 Business Class: 24 Economy Class: 69</pre>	<pre>Available connecting flights: Option 0: (intermediate city: Cape Town)) Flight Code: 2291 Crew Score: 1.17 Plane Score: 1.66 Number of Seats: First Class: 7 Business Class: 24 Economy Class: 69 Flight Code: 36309 Crew Score: 1.42 Plane Score: 1.85 Number of Seats: First Class: 8 Business Class: 19 Economy Class: 68 ----- Option 1: (intermediate city: Abu Dhabi)) Flight Code: 2570 Crew Score: 1.33 Plane Score: 1.92 Number of Seats: First Class: 8 Business Class: 21 Economy Class: 72 Flight Code: 26106 Crew Score: 1.42 Plane Score: 2.02 Number of Seats: First Class: 7 Business Class: 27 Economy Class: 85 -----</pre>
--	---

```
Enter the flight you want to book
Flight: 0
Enter the class you want to book
1. First Class
2. Business Class
3. Economy Class
Class: 3
How many seats would you like to book? Must be less than 68.
Number of seats: 3
Purchase successful!
Total price: 1464.0
Total points awarded: 146.0
```

7.2.3 Διεπαφή διαχειριστή (admin.py)

```
You are now connected to the database.
Type 'help' for a list of commands.
Type 'exit' to exit the program.

> help
Commands:
help - print this help message
exit - exit the program
create - create a new flight
stats - print statistics about the database

> stats
Stats Menu:
1. Popular airports
2. Well reviewed airplanes
3. Well reviewed crew
4. Popular days
5. Users with most purchases
6. Users with most referrals
7. Users with most points
8. Most cancelled flights
Enter the number of the stat you want to view (or 'exit' to go back): 4
Days with most flights:
Thursday: 9697 flights
Wednesday: 9091 flights
Friday: 8826 flights
Saturday: 8475 flights
Tuesday: 8366 flights
Monday: 7928 flights
Sunday: 7617 flights
Execution time: 43.0 milliseconds

Enter the number of the stat you want to view (or 'exit' to go back):
```