



**Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής
ΠΜΣ “Πληροφορική”
Ακαδημαϊκό έτος 2021-22(εαρινό εξάμηνο)**

<<Εργασία στο μάθημα ΒΑΣΕΙΣ & ΑΠΟΘΗΚΕΣ ΔΕΔΟΜΕΝΩΝ>>

**Επιβλέποντες καθηγητές: Γιάννης Θεοδορίδης, Ρόζα Μαυροπόδη. Εργ. βοηθός: Ι. Κοντούλης
Φοιτητές: Φώτιος Τσιούμας (ΜΠΠΛ21079), Νικόλαος Ανδρέου (ΜΠΠΛ21005)**

**Αθήνα
Σεπτέμβριος 2022**

Περιεχόμενα

Εισαγωγή	3
Ανάλυση Απαιτήσεων	4
Εννοιολογικός Σχεδιασμός ΒΔ	5
Λογικός Σχεδιασμός ΒΔ	7
Υλοποίηση ΒΔ	13
Εφαρμογή Διεπαφής με τη ΒΔ	15

Εισαγωγή

Η παρούσα εργασία αποτελεί την υλοποίηση μιας βάσης δεδομένων και τη δημιουργία μιας διεπαφής με αυτή, για μια αεροπορική εταιρεία βάσει των προδιαγραφών της. Η εργασία περιλαμβάνει τις παρακάτω ενότητες.

Η πρώτη ενότητα αφορά την ανάλυση των δεδομένων που μας έχει δώσει η αεροπορική εταιρεία ως προδιαγραφές. Η συγκεκριμένη ανάλυση παίζει σημαντικό ρόλο για την επόμενη ενότητα που αφορά το ER Model.

Η δεύτερη ενότητα αφορά τον εννοιολογικό σχεδιασμό της βάσης δεδομένων (ER Model) το οποίο προκύπτει από την ανάλυση της προηγούμενης ενότητας. Στόχος αυτής της ενότητας είναι η εύρεση των οντοτήτων που προκύπτουν από τις προδιαγραφές, των χαρακτηριστικών αυτών, των συσχετίσεων μεταξύ τους και τα είδη των συσχετίσεων αυτών.

Η τρίτη ενότητα αποτελεί τον λογικό σχεδιασμό της βάσης δεδομένων (Relation Model). Κατα το στάδιο αυτό, αξιοποιούμε τα πορίσματα από το προηγούμενο στάδιο ώστε να σχηματίσουμε τους πίνακες, τα πεδία τους, τα primary keys και τα foreign keys τους. Η ενότητα περιλαμβάνει επίσης τα scripts για τη δημιουργία των πινάκων της βάσης, για το insert των δεδομένων και για τα παραγόμενα πεδία.

Η τέταρτη ενότητα αποτελεί την υλοποίηση της βάσης δεδομένων και περιλαμβάνει ενδεικτικά screenshots από τα instances των πινάκων.

Η πέμπτη και τελευταία ενότητα περιλαμβάνει την υλοποίηση μιας διεπαφής της βάσης δεδομένων. Αυτή η διεπαφή αποσκοπεί στην αλληλεπίδραση του χρήστη με τη βάση, μέσω συγκεκριμένων φορμών που θα του δίνουν τη δυνατότητα να κάνει αναζήτηση, επεξεργασία, εισαγωγή και διαγραφή δεδομένων.

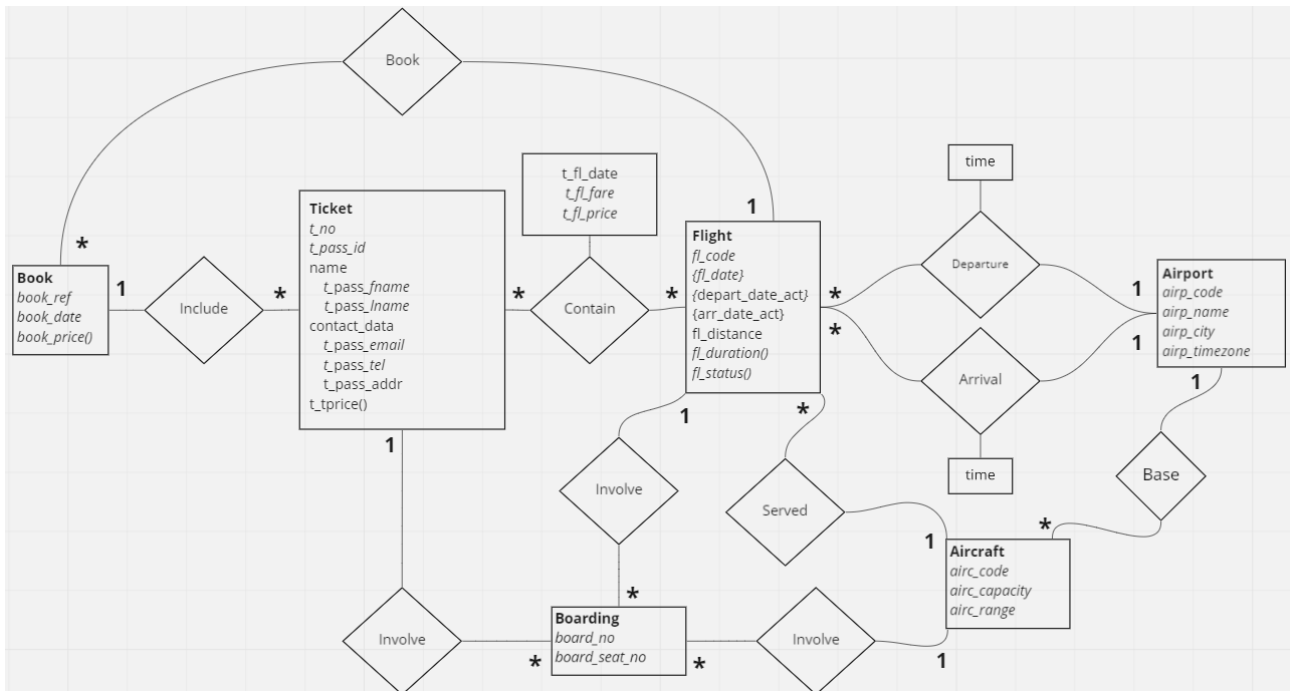
Ανάλυση Απαιτήσεων

Βάσει των προδιαγραφών της αεροπορικής εταιρείας που μας δόθηκαν, προκύπτουν 5 βασικές οντότητες, οι οποίες συσχετίζονται μεταξύ τους και περιέχουν διαφόρων ειδών χαρακτηριστικά και περιορισμούς. Ακολουθεί η ανάλυση των απαιτήσεων και στη συνέχεια ο ορισμός οντοτήτων με τις μεταξύ τους σχέσεις, τα χαρακτηριστικά και τους περιορισμούς τους:

1. Για να ταξιδέψει ένας επιβάτης με την αεροπορική εταιρεία πρέπει να κάνει **κράτηση** εισιτηρίου εκ των προτέρων. Η κράτηση περιλαμβάνει ένα **book_date** (δεν πρέπει να είναι νωρίτερα από ένα μήνα από την πτήση) και ένα αριθμό **book_ref** (ο οποίος περιλαμβάνει πολλούς επιβάτες με ξεχωριστό όμως εισιτήριο και αποτελεί συνδυασμό 16 χαρακτήρων και ψηφίων). Για κάθε κράτηση χρειαζόμαστε τη **συνολική αξία** της.
2. Ο κάθε επιβάτης έχει το δικό του **εισιτήριο**. Κάθε εισιτήριο περιλαμβάνει έναν μοναδικό αριθμό (**ticket_no**) και τα στοιχεία του επιβάτη (αριθμό ταυτότητας (**passenger_id**), ονοματεπώνυμο (**passenger_name**), στοιχεία επικοινωνίας (**contact_data**)). Επίσης ένα εισιτήριο περιέχει περισσότερα από ένα τμήματα πτήσεων, που το κάθε τμήμα έχει την τιμή του (**amount**) και την κατηγορία ταξιδιού του (**fare** - Economy/Business/FirstClass).
3. Η κάθε **πτήση** πηγαίνει από ένα αεροδρόμιο σε ένα άλλο. Η πτήση περιέχει έναν **αριθμό πτήσης** που αφορά το **ζεύγος αεροδρόμιο αναχώρησης - αεροδρόμιο άφιξης** (τα δύο αεροδρόμια δεν πρέπει να συμπίπτουν) και ταυτοποιείται με βάση την ημερομηνία αναχώρησης (**departure_date**). Επιπλέον περιλαμβάνει το **μοντέλο του αεροσκάφους**, την χιλιομετρική απόσταση (**distance**), την προγραμματισμένη ώρα αναχώρησης - άφιξης (**schedule_departure_time** - **schedule_arrival_time**), την πραγματική ώρα αναχώρησης - άφιξης (**actual_departure_time** - **actual_arrival_time**), την προγραμματισμένη διάρκεια πτήσης (**schedule_duration**) και την κατάσταση της πτήσης (**flight_status**) για την οποία ισχύουν τα εξής.
 - **Scheduled** (από 1 μήνα έως 24 ώρες πριν την προγραμματισμένη ώρα αναχώρησης): η πτήση είναι διαθέσιμη για κράτηση αλλά δεν έχει ανοίξει ακόμη το check-in. Πριν από αυτό το χρονικό διάστημα του 1 μηνός δεν μπορεί να γίνει κράτηση για την πτήση.
 - **OnTime** (από 24 ώρες πριν την προγραμματισμένη ώρα αναχώρησης έως την προγραμματισμένη ώρα αναχώρησης): η πτήση είναι ανοιχτή για check-in.
 - **Delayed** (μετά την προγραμματισμένη και πριν την πραγματική ώρα αναχώρησης, εφόσον η πτήση δεν έχει ακυρωθεί): η πτήση είναι ανοιχτή για check-in αλλά υπάρχει καθυστέρηση.
 - **Departed** (μετά την πραγματική ώρα αναχώρησης και πριν την πραγματική ώρα άφιξης): Το αεροσκάφος έχει αναχωρήσει και βρίσκεται εν πτήση.
 - **Arrived** (μετά την πραγματική ώρα άφιξης): Το αεροσκάφος έχει φτάσει στον προορισμό του.
 - **Cancelled**: Η πτήση ακυρώθηκε.
4. Κάθε **αεροδρόμιο** εξυπηρετεί διάφορες πτήσεις. Το κάθε αεροδρόμιο περιλαμβάνει ένα μοναδικό κωδικό (**code**), το όνομα του (**name**), την πόλη στην οποία εδρεύει (**city**) και την ζώνη ώρας του (**timezone**).
5. Κάθε **αεροσκάφος** ανήκει σε ένα αεροδρόμιο και ανατίθεται σε μια πτήση. Το κάθε αεροσκάφος περιλαμβάνει έναν κωδικό (**aircraft_code**), το **όνομα του μοντέλου**, την χωρητικότητα του (**capacity**) και την απόσταση που μπορεί να καλύψει (**range**). Ένα αεροσκάφος δεν μπορεί να ανατεθεί σε μια πτήση όπου η απόσταση της είναι μεγαλύτερη από την εμβέλεια του.
6. Κατά την διάρκεια του check in ο κάθε επιβάτης λαμβάνει μια **κάρτα επιβίβασης**. Κάθε κάρτα επιβίβασης λαμβάνει διαδοχικούς αριθμούς (**boarding_no** - είναι μοναδικοί μόνο σε επίπεδο πτήσης) και σε αυτή αναγράφονται το **όνομα του επιβάτη**, ο **αριθμός πτήσης** του και ο αριθμός θέσης του (**seat_no**). Οι επιπλέον περιορισμοί για την κάρτα επιβίβασης είναι οι παρακάτω:
 - Ο επιβάτης μπορεί να λάβει κάρτα επιβίβασης μόνο εάν η πτήση περιλαμβάνεται στο εισιτήριό του και μόνο εάν η πτήση είναι σε κατάσταση OnTime ή Delayed.
 - Ο συνδυασμός πτήσης-θέσης πρέπει να είναι μοναδικός για να αποφευχθεί η έκδοση δύο καρτών επιβίβασης για την ίδια θέση.

- Δεν επιτρέπεται να εκδοθούν περισσότερες κάρτες επιβίβασης από τις θέσεις που διαθέτει το μοντέλο αεροσκάφους που εκτελεί την πτήση.

Εννοιολογικός Σχεδιασμός ΒΔ



Με βάση την ανάλυση δεδομένων προκύπτουν οι παρακάτω 6 βασικές οντότητες. Παρακάτω καταγράφονται οι οντότητες, με τα χαρακτηριστικά τους και τις συσχετίσεις μεταξύ τους.

Οντότητες:

Κράτηση (**Book**) :

- **book_ref** (αριθμός κράτησης 6 θέσεων από γράμματα και ψηφία και μπορεί να περιλαμβάνει περισσότερους από έναν επιβάτες με διαφορετικό ticket).
- **book_date** (η ημερομηνία που έγινε η κράτηση, η οποία δεν μπορεί νωρίτερα από έναν μήνα από την πτήση).
- **book_price()** (αποτελεί παραγόμενο πεδίο και αναγράφει το σύνολο των τιμών των ticket της κράτησης).

Εισιτήριο (**Ticket**) :

- **t_no** (μοναδικός αριθμός για κάθε ticket που αποτελείται από 13 ψηφία).
- **t_pass_id** (αριθμός εγγράφου του επιβάτη)
- **name** (αποτελεί σύνθετο δεδομένο και περιλαμβάνει το **t_pass_fname** και το **t_pass_lname**)
- **contact_data** (αποτελεί σύνθετο δεδομένο και περιλαμβάνει το **t_pass_email**, το **t_pass_tel** και το **t_pass_addr**)
- **t_tprice()** (αποτελεί παραγόμενο πεδίο και αναγράφει το σύνολο των τιμών των επιμέρους τμημάτων πτήσεων του κάθε εισιτηρίου)

Πτήση (**Flight**) :

- **fl_code** (μοναδικός αριθμός για κάθε πτήση)
- **fl_date** (αποτελεί πλειότιμο καθώς η κάθε πτήση πραγματοποιείται πολλές φορές)
- **depart_date_act** (η ημερομηνία και ώρα αναχώρησης αποτελεί πλειότιμο καθώς η ώρα αναχώρησης της κάθε πτήσης διαφέρει κάθε φορά)
- **arr_date_act** (η ημερομηνία και ώρα άφιξης αποτελεί πλειότιμο καθώς η ώρα άφιξης της κάθε πτήσης διαφέρει κάθε φορά)
- **fl_distance** (η χιλιομετρική απόσταση της πτήσης)
- **fl_duration** (η προγραμματισμένη διάρκεια της πτήσης αποτελεί παραγόμενο και αποτελείται από τη διαφορά προγραμματισμένης ώρα αναχώρησης - άφιξης)
- **fl_status** (το status της πτήσης αποτελεί παραγόμενο και εξαρτάται από την τωρινή ώρα σε σχέση με την ώρα αναχώρησης)

Αεροδρόμιο (Airport):

- **airp_code** (ο μοναδικός κωδικός αεροδρομίου)
- **airp_name** (το όνομα του αεροδρομίου)
- **airp_city** (η πόλη που εδρεύει το αεροδρόμιο)
- **airp_timezone** (το timezone της πόλης που εδρεύει το αεροδρόμιο)

Αεροσκάφος (Aircraft):

- **airc_code** (ο μοναδικός κωδικός του αεροσκάφους)
- **airc_capacity** (ο αριθμός θέσεων του αεροσκάφους)
- **airc_range** (η χιλιομετρική απόσταση που μπορεί να καλύψει το αεροσκάφος)

Κάρτα επιβίβασης (Boarding_pass):

- **board_no** (ο αριθμός της κάρτας επιβίβασης που αποτελεί μοναδικό αριθμό μόνο κατά τη διάρκεια της ίδιας πτήσης)
- **board_seat_no** (ο αριθμός θέσης του επιβάτη που σε συνδυασμός με το board_no πρέπει να είναι μοναδικά στα πλαίσια μιας συγκεκριμένης πτήσης για να μην έχουμε δύο επιβάτες στην ίδια θέση)

Συσχετίσεις - Χαρακτηριστικά Συσχετίσεων - Περιορισμοί:

Κράτηση – Εισιτήριο: Μια κράτηση περιλαμβάνει πολλά εισιτήρια καθώς η κράτηση μπορεί να περιλαμβάνει πολλούς επιβάτες με διαφορετικό εισιτήριο.

Κράτηση - Πτήση: Μία πτήση περιλαμβάνει πολλές κρατήσεις.

Εισιτήριο – Πτήση: Ένα εισιτήριο περιλαμβάνει πολλά τμήματα πτήσεων και μια πτήση περιλαμβάνει πολλά εισιτήρια. Η σχέση έχει και δικά της χαρακτηριστικά όπως η ημερομηνία πτήσης, η τιμή πτήσης και η κατηγορία ταξιδιού της πτήσης.

Πτήση - Αεροδρόμιο: Πολλές πτήσεις αναχωρούν από ένα αεροδρόμιο και φτάνουν σε ένα άλλο αεροδρόμιο. Υπάρχει διπλή συσχέτιση που αφορά την αναχώρηση και την άφιξη και η κάθε σχέση έχει και από ένα χαρακτηριστικό που αφορά την προγραμματισμένη ώρα αναχώρησης - άφιξης.

Αεροδρόμιο - Αεροσκάφος: Σε ένα αεροδρόμιο εδρεύουν πολλά αεροσκάφη.

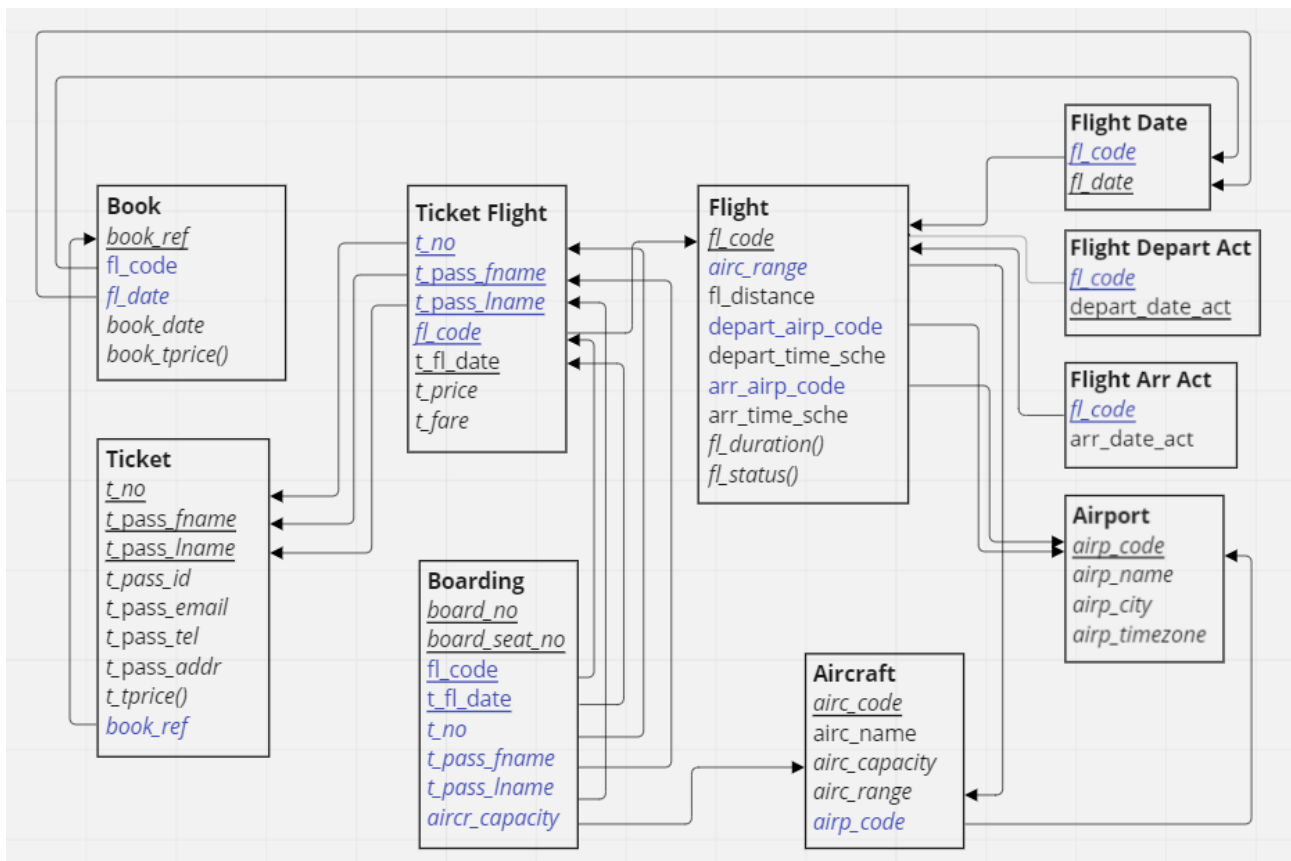
Αεροσκάφος - Πτήση: Ένα αεροσκάφος εκτελεί πολλές πτήσεις. Επίσης μια πτήση δεν μπορεί να ανατεθεί σε ένα αεροσκάφος με μικρότερο range απο το distance της.

Κάρτα Επιβίβασης - Αεροσκάφος: Για ένα αεροσκάφος εκδίδονται πολλές κάρτες επιβίβασης. Επίσης οι κάρτες επιβίβασης δεν μπορούν να είναι περισσότερες από τις θέσεις του αεροσκάφους.

Κάρτα Επιβίβασης - Πτήση: Μία πτήση περιλαμβάνει πολλές κάρτες επιβίβασης.

Κάρτα Επιβίβασης - Εισιτήριο: Ένα εισιτήριο περιλαμβάνει πολλές κάρτες επιβίβασης καθώς περιλαμβάνει και περισσότερα του ενός τμήματα πτήσεων.

Λογικός Σχεδιασμός ΒΔ



Από τις οντότητες και τις συσχετίσεις του ER Model προκύπτουν οι παρακάτω πίνακες, τα χαρακτηριστικά τους και τα primary και foreign key τους:

Book:

- Προέκυψε από την οντότητα Book. Συνδέεται με τον πίνακα Flight Date καθώς χρειαζόμαστε την ημερομηνία της πτήσης για να δημιουργήσουμε το constraint του ενός μήνα με τον book_date.
- **Primary keys:** book_ref.
- **Foreign keys:** fl_code, fl_date (θα αποτελέσει constraint για το book_date).

Ticket:

- Προέκυψε από την οντότητα Ticket. Συνδέεται με τον πίνακα Book.
- **Primary keys:** t_no, t_pass_fname, t_pass_lname.
- **Foreign keys:** book_ref.

Flight:

- Προέκυψε από την οντότητα Flight. Συνδέεται με τον πίνακα Airport καθώς χρειαζόμαστε τον κωδικό του αεροδρομίου αναχώρησης και άφιξης και με τον πίνακα Aircraft καθώς χρειαζόμαστε το airc_range που είναι το η εμβέλεια του αεροσκάφους, για να φτιάξουμε το constraint με το fl_distance ώστε να μην ανατεθεί μια πτήση σε αεροσκάφος μικρότερης εμβέλειας από την απόσταση της.
- **Primary keys:** fl_code.
- **Foreign keys:** airc_range, depart_airp_code, arr_airp_code.

Ticket Flight:

- Προέκυψε από την σχέση πολλά προς πολλά του πίνακα Ticket με τον πίνακα Flight. Ο συγκεκριμένος πίνακας εκτός από τα primary keys των δύο παραπάνω πινάκων έχει τρία ακόμα χαρακτηριστικά που ήρθαν από τη σχέση μεταξύ των πινάκων αυτών (t_fl_date, t_price, t_fare). Συνδέεται κατα συνέπεια με τους πίνακες Ticket και Flight.
- **Primary keys:** t_no, t_pass_fname, t_pass_lname, fl_code, t_fl_date.
- **Foreign keys:** t_no, t_pass_fname, t_pass_lname, fl_code.

Flight Date:

- Προέκυψε από το πλειότιμο fl_date του πίνακα flight και συνδέεται με τον πίνακα Flight.
- **Primary keys:** fl_code.
- **Foreign keys:** fl_date.

Flight Departure Actual:

- Προέκυψε από το πλειότιμο depart_date_act του πίνακα flight και συνδέεται με τον πίνακα Flight.
- **Primary keys:** fl_code.
- **Foreign keys:** depart_date_act.

Flight Arrival Actual:

- Προέκυψε από το πλειότιμο arr_date_act του πίνακα flight και συνδέεται με τον πίνακα Flight.
- **Primary keys:** fl_code.
- **Foreign keys:** arr_date_act.

Airport:

- Προέκυψε από την οντότητα Airport.
- **Primary keys:** airp_code.

Aircraft:

- Προέκυψε από την οντότητα Aircraft και συνδέεται με τον πίνακα Airport.
- **Primary keys:** airc_code.
- **Foreign keys:** airp_code.

Boarding:

- Προέκυψε από την οντότητα Boarding και συνδέεται με τον πίνακα Ticket Flight για να πάρει τις πληροφορίες που χρειάζεται τόσο από το Ticket όσο και από τον Flight. Επίσης, συνδέεται με τον πίνακα Aircraft για να πάρει το πεδίο airc_capacity και να δημιουργήσει το constraint για το πεδίο του board_no, καθώς ο αριθμός των καρτών επιβίβασης δεν θα πρέπει να υπερβαίνει τον πλήθος των θέσεων του αεροσκάφους.
- **Primary keys:** board_no, board_seat_no, fl_code, t_fl_date.
- **Foreign keys:** fl_code, t_fl_date, t_no, t_pass_fname, t_pass_lname, aircr_capacity.

Scripts δημιουργίας βάσης, εισαγωγής δεδομένων και παραγόμενων πεδίων:

Το script για το create των παραπάνω πινάκων βρίσκεται στο αρχείο **1_create_db.sql** και παρακάτω παρουσιάζονται κάποια ενδεικτικά screenshots:

```
1 CREATE TABLE airport(
2     airp_code VARCHAR(3) CHECK (char_length(airp_code) = 3),
3     airp_name VARCHAR(20) NOT NULL,
4     airp_city VARCHAR(20) NOT NULL,
5     airp_timezone TEXT NOT NULL,
6     PRIMARY KEY(airp_code)
7 );
8
9 CREATE TABLE aircraft(
10     airc_code VARCHAR(3) CHECK (airc_code ~ '^(\\d{3})?$'),
11     airc_name VARCHAR(20) NOT NULL,
12     airc_capacity INT UNIQUE,
13     airc_range INT UNIQUE,
14     airp_code VARCHAR(3),
15     PRIMARY KEY (airc_code),
16     FOREIGN KEY (airp_code) REFERENCES airport ON DELETE CASCADE
17 );
```

```
19 CREATE TABLE flight (
20     fl_code VARCHAR(10),
21     airc_range INT,
22     fl_distance INT NOT NULL CHECK (fl_distance <= airc_range),
23     depart_airp_code VARCHAR(20) CHECK (depart_airp_code <> arr_airp_code),
24     depart_time_sche TIME,
25     arr_airp_code VARCHAR(20) CHECK (depart_airp_code <> arr_airp_code),
26     arr_time_sche TIME,
27     fl_duration TIME GENERATED ALWAYS AS (arr_time_sche - depart_time_sche) STORED,
28     fl_status VARCHAR(15),
29     PRIMARY KEY(fl_code),
30     FOREIGN KEY(depart_airp_code) REFERENCES airport ON DELETE SET NULL,
31     FOREIGN KEY(arr_airp_code) REFERENCES airport,
32     FOREIGN KEY(airc_range) REFERENCES aircraft(airc_range) ON DELETE SET NULL
33 );
34
35 CREATE TABLE flight_date(
36     fl_code VARCHAR(10),
37     fl_date DATE,
38     PRIMARY KEY(fl_code, fl_date),
39     FOREIGN KEY(fl_code) REFERENCES flight ON DELETE CASCADE
40 );
41
42 CREATE TABLE flight_depart_act(
43     fl_code VARCHAR(10),
44     depart_date_act TIMESTAMP,
45     PRIMARY KEY(fl_code, depart_date_act),
46     FOREIGN KEY(fl_code) REFERENCES flight ON DELETE CASCADE
47 );
48
49 CREATE TABLE flight_arr_act(
50     fl_code VARCHAR(10),
51     arr_date_act TIMESTAMP,
52     PRIMARY KEY(fl_code, arr_date_act),
53     FOREIGN KEY(fl_code) REFERENCES flight ON DELETE CASCADE
54 );
55
```

```
56 CREATE TABLE book(
57     book_ref VARCHAR(6) UNIQUE CHECK (char_length(book_ref) = 6),
58     fl_code VARCHAR(10),
59     fl_date DATE,
60     book_date TIMESTAMP NOT NULL CHECK (fl_date - book_date <= '1 mon'),
61     book_tprice DOUBLE PRECISION,
62     PRIMARY KEY(book_ref),
63     FOREIGN KEY(fl_code, fl_date) REFERENCES flight_date(fl_code, fl_date) ON DELETE CASCADE
64 );
65
66 CREATE TABLE ticket(
67     t_no VARCHAR(13) CHECK (t_no ~ '^(\\d{13})?$'),
68     t_pass_fname VARCHAR(20),
69     t_pass_lname VARCHAR(20),
70     t_pass_id VARCHAR(15) NOT NULL,
71     t_pass_email VARCHAR(20) NOT NULL,
72     t_pass_tel VARCHAR(20) NOT NULL,
73     t_pass_addr VARCHAR(40) NOT NULL,
74     t_tprice DOUBLE PRECISION,
75     book_ref VARCHAR(6),
76     PRIMARY KEY(t_no, t_pass_fname, t_pass_lname),
77     FOREIGN KEY(book_ref) REFERENCES book(book_ref) ON DELETE CASCADE
78 );
79
80 CREATE TABLE ticket_flight (
81     t_no VARCHAR(13),
82     t_pass_fname VARCHAR(20),
83     t_pass_lname VARCHAR(20),
84     fl_code VARCHAR(10),
85     t_fl_date DATE NOT NULL,
86     t_price DOUBLE PRECISION NOT NULL,
87     t_fare VARCHAR(15) NOT NULL CHECK (t_fare IN ('Economy', 'Business', 'First class')),
88     PRIMARY KEY(t_no, t_pass_fname, t_pass_lname, fl_code, t_fl_date),
89     FOREIGN KEY(t_no, t_pass_fname, t_pass_lname) REFERENCES ticket(t_no, t_pass_fname, t_pass_lname) ON DELETE CASCADE,
90     FOREIGN KEY(fl_code) REFERENCES flight(fl_code) ON DELETE CASCADE
91 );
92
```

```
93 CREATE TABLE boarding(
94     airc_capacity INT,
95     board_no INT CHECK (board_no <= airc_capacity),
96     board_seat_no INT,
97     t_no VARCHAR(13),
98     t_pass_fname VARCHAR(20),
99     t_pass_lname VARCHAR(20),
100     fl_code VARCHAR(10),
101     t_fl_date DATE,
102     PRIMARY KEY (board_no, board_seat_no, fl_code, t_fl_date),
103     FOREIGN KEY(airc_capacity) REFERENCES aircraft(airc_capacity) ON DELETE SET NULL,
104     FOREIGN KEY(t_no, t_pass_fname, t_pass_lname, fl_code, t_fl_date) REFERENCES ticket_flight(t_no, t_pass_fname, t_pass_lname, fl_code, t_fl_date) ON DELETE CASCADE
105 );
```

Το script για το insert των data στους παραπάνω πινάκων βρίσκεται στο αρχείο **2_insert_data.sql** και παρακάτω παραθέτουμε κάποια ενδεικτικά screenshots του script:

```
2_insert_data.sql X
database > 2_insert_data.sql
1 INSERT INTO airport (airp_code, airp_name, airp_city, airp_timezone) VALUES
2 ('adf', 'pkybird0', 'Songjianghe', 'Asia/Harbin'),
3 ('xlv', 'gmaccurlye1', 'Landi Kotal', 'Asia/Karachi'),
4 ('azd', 'cbeeze2', 'Vidin', 'Europe/Sofia'),
5 ('vxn', 'ablondell3', 'Moscow', 'Europe/Moscow'),
6 ('zxc', 'ccolville4', 'Waldbillig', 'Europe/Luxembourg'),
7 ('das', 'arp_majal', 'Itapeva', 'Asia/Shanghai'),
8 ('ftg', 'el.ben', 'Koronadal', 'Asia/Karachi'),
9 ('zas', 'Marinna_air', 'Jiekeng', 'Europe/Sofia'),
10 ('zuy', 'Salomone', 'Wysoka', 'Asia/Kabul'),
11 ('xio', 'Alec', 'Skoroszyce', 'Europe/Luxembourg'),
12 ('dgh', 'Dagmar', 'Tatariv', 'Asia/Harbin'),
13 ('lvx', 'Ernestus', 'Iwaki', 'Asia/Karachi'),
14 ('dza', 'Sarah Frontman', 'Sofia', 'Europe/Sofia'),
15 ('xvn', 'Zorina', 'Jiekeng', 'Europe/Moscow'),
16 ('xcz', 'Rodney', 'Himaya', 'Europe/Luxembourg'),
17 ('fda', 'Reyna', 'Nangerang', 'Asia/Harbin'),
18 ('fds', 'Lela', 'Jiekeng', 'Asia/Karachi'),
19 ('zxs', 'Gardner', 'Athens', 'Europe/Instabul'),
20 ('hvx', 'Harley', 'Harbin', 'Asia/Harbin'),
21 ('wsa', 'Ode', 'Bromma', 'Europe/Luxembourg');
```

```
22
23 INSERT INTO aircraft (airc_code, airc_name, airc_capacity, airc_range, airp_code) VALUES
24 (345, 'rspringthorpe0', 35, 66, 'adf'),
25 (224, 'ggreiser1', 42, 24, 'xlv'),
26 (567, 'bokenden2', 101, 120, 'azd'),
27 (274, 'lfosse3', 52, 70, 'vxn'),
28 (264, 'jperrett4', 6, 96, 'zxc'),
29 (348, 'rspringthorpe0', 37, 67, 'das'),
30 (222, 'ggreiser1', 43, 15, 'ftg'),
31 (577, 'bokenden2', 100, 14, 'zas'),
32 (374, 'lfosse3', 53, 10, 'zuy'),
33 (364, 'jperrett4', 40, 100, 'xio');
34
35 INSERT INTO flight (fl_code, airc_range, fl_distance, depart_airp_code, depart_time_sche, arr_airp_code, arr_time_sche, fl
36 ('EDoazFvG2m', 66, 27, 'adf', '2:26 AM', 'xlv', '10:20 AM', 'Not Booked'),
37 ('16aghzXbPp', 96, 79, 'azd', '2:56 PM', 'vxn', '5:19 PM', 'Not Booked'),
38 ('1LeMJJzMjw', 96, 71, 'xlv', '11:12 AM', 'zxc', '1:56 AM', 'Not Booked'),
39 ('1DxfMSFQW8', 66, 31, 'adf', '7:09 PM', 'zxc', '8:04 PM', 'Not Booked'),
40 ('1J8PSkCuMZ', 14, 3, 'zxc', '4:49 PM', 'adf', '5:43 PM', 'Not Booked'),
41 ('HfdofJ86Jo', 66, 35, 'das', '5:26 AM', 'azd', '10:20 AM', 'Not Booked'),
42 ('OPljf24efj', 100, 79, 'azd', '4:56 PM', 'wsa', '5:19 PM', 'Not Booked'),
43 ('AV769djdvo', 96, 71, 'fda', '12:12 AM', 'wsa', '1:56 AM', 'Not Booked'),
44 ('hti86HUK97', 67, 31, 'zxs', '4:09 PM', 'hvx', '8:04 PM', 'Not Booked'),
45 ('1343ffvrsv', 14, 3, 'ftg', '8:49 PM', 'das', '5:43 AM', 'Not Booked');
46
```

```
47
48 INSERT INTO flight_date (fl_code, fl_date) VALUES
49 ('EDoazFvG2m', '6/9/2022'),
50 ('EDoazFvG2m', '9/9/2022'),
51 ('EDoazFvG2m', '12/9/2022'),
52 ('EDoazFvG2m', '15/9/2022'),
53 ('16aghzXbPp', '13/9/2022'),
54 ('16aghzXbPp', '14/9/2022'),
55 ('16aghzXbPp', '15/9/2022'),
56 ('16aghzXbPp', '25/9/2022'),
57 ('1LeMJJzMjw', '11/9/2022'),
58 ('1LeMJJzMjw', '20/9/2022'),
59 ('1LeMJJzMjw', '22/9/2022'),
60 ('1LeMJJzMjw', '24/9/2022'),
61 ('1DxfMSFQW8', '10/9/2022'),
62 ('1DxfMSFQW8', '12/9/2022'),
63 ('1DxfMSFQW8', '25/9/2022'),
64 ('1DxfMSFQW8', '26/9/2022'),
65 ('1J8PSkCuMZ', '2/9/2022'),
66 ('1J8PSkCuMZ', '3/9/2022'),
67 ('1J8PSkCuMZ', '9/9/2022'),
68 ('1J8PSkCuMZ', '13/9/2022'),
```

```
213 INSERT INTO ticket (t_no, t_pass_fname, t_pass_lname, t_pass_id, t_pass_email, t_pass_tel, t_pass_addr, t_tprice, book_ref) values
214 ('1265789542646', 'Celka', 'Nordass', '5556dfb', 'cnordass0@com.com', '+54-227-737-2986', '8 Union Center', null, 'gh3j42'),
215 ('5435851315564', 'Gwendolyn', 'Crosfeld', '352372347', 'gcrosfeld2@wikimedia.org', '+62-830-134-6538', '99 Summer Ridge Terrace', null, '4
216 ('7633435365673', 'Marthe', 'Breeze', '43574hgj3', 'mbreeze3@tripadvisor.com', '+48-710-372-1545', '5998 Briar Crest Hill', null, 'gh5hdr'),
217 ('3574345747346', 'Enid', 'Balsillie', '573363457', 'ebalsillie4@dagondesign.com', '+63-701-815-1748', '67807 Amoth Way', null, 'dd3j42'),
218 ('4257474534657', 'Melodee', 'Warby', 'jjj34453yj', 'mwarby5@timesonline.co.uk', '+976-365-348-1817', '83 Hudson Street', null, 'hhy54h'),
219 ('3457432346245', 'Vivienne', 'Staning', '346834hjwrh', 'vstaning6@taobao.com', '+51-603-631-8933', '9106 Rieder Park', null, 'dbbf3t'),
220 ('4257425724523', 'Clari', 'O' Ronan', 'yjtjr5', 'coronan7@upenn.edu', '+358-178-190-6404', '9 Brentwood Lane', null, 'gbgb63'),
221 ('2356236656522', 'Benny', 'Nuemann', '6437', 'bnuemann8@examiner.com', '+48-169-439-7624', '7 Mosinee Plaza', null, 'dbd23b'),
222 ('2457468545686', 'Cortney', 'Campling', 'jdn4gn', 'ccampling9@amazonaws.com', '+86-255-412-8752', '5211 Monica Road', null, '54gfd3'),
223 ('4568325768655', 'Carrissa', 'Carthy', 'hhct737', 'ccarthy@newyorker.com', '+255-137-636-2473', '5022 Dapin Trail', null, 'bdsdf2'),
```

Επιπλέον έχουμε δημιουργήσει τρία scripts, που αποτελούνται από functions και executions των functions αυτών για το γέμισμα των παραγόμενων πεδίων (**t_tprice**, **book_tprice**, **fl_status**).

Η πρώτη function έχει το ρόλο να γεμίσει το παραγόμενο πεδίο “**t_tprice**” του πίνακα “**ticket**” με το άθροισμα των διαφόρων τιμών του πεδίου “**t_price**” του πίνακα “**ticket_flight**” που έχουν την ίδια τιμή στο πεδίο “**t_no**”, δηλαδή το άθροισμα των τιμών των επιμέρους τμημάτων πτήσεων ενός εισιτηρίου (Αρχείο: 3_function_status.sql). Ακολουθεί ενδεικτικό screenshot της function και κάποιων ενδεικτικών εκτελέσεων της:

```
3_function_ticket.sql X
database > 3_function_ticket.sql
1 CREATE OR REPLACE FUNCTION sum_total_ticket(ticket_no VARCHAR(13))
2 RETURNS DOUBLE PRECISION
3 AS
4 $$
5 DECLARE
6     summs DOUBLE PRECISION;
7 BEGIN
8     SELECT sum(t_price) into summs FROM ticket_flight WHERE t_no = ticket_no;
9     UPDATE ticket SET t_tprice = summs WHERE t_no = ticket_no;
10    RETURN summs;
11 END
12 $$ LANGUAGE plpgsql;
13
14
15 SELECT sum_total_ticket('1265789542646');
16 SELECT sum_total_ticket('6541551135448');
17 SELECT sum_total_ticket('5435851315564');
18 SELECT sum_total_ticket('7633435365673');
19 SELECT sum_total_ticket('3574345747346');
20 SELECT sum_total_ticket('4257474534657');
21 SELECT sum_total_ticket('3457432346245');
22 SELECT sum_total_ticket('4257425724523');
23 SELECT sum_total_ticket('2356236656522');
24 SELECT sum_total_ticket('2457468545686');
25 SELECT sum_total_ticket('4568325768655');
26 SELECT sum_total_ticket('3592458564464');
27 SELECT sum_total_ticket('3686835634637');
28 SELECT sum_total_ticket('6785673457682');
29 SELECT sum_total_ticket('3568524573468');
```

Η δεύτερη function έχει το ρόλο να γεμίσει το παραγόμενο πεδίο “**book_tprice**” του πίνακα “**book**” με το άθροισμα των διαφόρων τιμών του πεδίου “**t_tprice**” του πίνακα “**ticket**” που έχουν την ίδια τιμή στο πεδίο “**book_ref**”, δηλαδή το άθροισμα των τιμών των tickets που ανήκουν στην ίδια κράτηση (Αρχείο: 4_function_book.sql). Ακολουθεί ενδεικτικό screenshot της function και κάποιων ενδεικτικών εκτελέσεων της:

```
4_function_book.sql X
database > 4_function_book.sql
1 CREATE OR REPLACE FUNCTION sum_total_book(book_r VARCHAR(6))
2 RETURNS DOUBLE PRECISION
3 AS
4 $$
5 DECLARE
6     summs DOUBLE PRECISION;
7 BEGIN
8     SELECT sum(t_tprice) into summs FROM ticket WHERE book_ref = book_r;
9     UPDATE book SET book_tprice = summs WHERE book_ref = book_r;
10    RETURN summs;
11 END
12 $$ LANGUAGE plpgsql;
13
14 SELECT sum_total_book('gh3j42');
15 SELECT sum_total_book('ukh1d5');
16 SELECT sum_total_book('44gh21');
17 SELECT sum_total_book('gh5hdr');
18 SELECT sum_total_book('dd3j42');
19 SELECT sum_total_book('hhy54h');
20 SELECT sum_total_book('dbbf3t');
21 SELECT sum_total_book('gbgb63');
22 SELECT sum_total_book('dbd23b');
23 SELECT sum_total_book('54gfd3');
24 SELECT sum_total_book('bdsdf2');
25 SELECT sum_total_book('sfb3sv');
26 SELECT sum_total_book('gnvx32');
27 SELECT sum_total_book('56657b');
28 SELECT sum_total_book('de324g');
```

Η τρίτη function έχει τέσσερα ορίσματα. Το πρώτο δέχεται έναν αριθμό ticket, το δεύτερο δέχεται έναν αριθμό πτήσης, το τρίτο δέχεται μια ημερομηνία και το τέταρτο μια ώρα. Με αυτόν τον τρόπο προσομοιώνουμε το status που μπορεί να πάρει η πτήση ανάλογα με τα δύο τελευταία ορίσματα. Αφού εκτελεστεί η function δίνοντας τα παραπάνω ορίσματα, γίνεται select του **t_fl_date** (πίνακας ticket_flight), του **depart_time_sche** (πίνακας flight), της ώρα του **depart_date_act** (πίνακας flight_depart_act), της ώρα του **arr_date_act** (πίνακας flight_arr_act) και του **fl_status** (πίνακας flight) και με βάση τα κριτήρια για το flight status, τα οποία έχουν οριστεί αναλυτικά στο κεφάλαιο της ανάλυσης απαιτήσεων, γεμίζει το παραγόμενο πεδίο “**fl_status**” του πίνακα “**flight**” με το αντίστοιχο status ([Αρχείο: 5_function_status.sql](#)). Ακολουθεί ενδεικτικό screenshot της function και κάποιων ενδεικτικών εκτελέσεων της:

```

5_function_status.sql X
database > 5_function_status.sql
1 CREATE OR REPLACE FUNCTION status_check(ticket_no VARCHAR(13), flight_code VARCHAR(10), i_date DATE, r_time TIME)
2 RETURNS VARCHAR
3 AS
4 $$
5 DECLARE
6     f_mounth DATE;
7     f_time TIME;
8     f_act_dep_time TIME;
9     f_act_arr_time TIME;
10    check_if_canceled VARCHAR;
11 BEGIN
12     SELECT t_fl_date INTO f_mounth FROM ticket_flight WHERE fl_code = flight_code AND t_no = ticket_no;
13     SELECT depart_time_sche INTO f_time FROM flight WHERE fl_code = flight_code;
14     SELECT cast(depart_date_act as time) INTO f_act_dep_time FROM flight_depart_act where cast(depart_date_act as date) = i_date
15     AND fl_code = flight_code;
16     SELECT cast(arr_date_act as time) INTO f_act_arr_time FROM flight_arr_act where cast(arr_date_act as date) = i_date AND fl_code
17     = flight_code;
18     SELECT fl_status INTO check_if_canceled FROM flight where fl_code = flight_code;
19     IF (check_if_canceled != 'Canceled') then
20         IF (EXTRACT(YEAR FROM f_mounth) - EXTRACT(YEAR FROM i_date) >= 1 OR EXTRACT(MONTH FROM f_mounth) - EXTRACT(MONTH FROM
21         i_date) >= 1) THEN
22             UPDATE flight SET fl_status = 'Not Booked' WHERE fl_code = flight_code;
23             RETURN 'Not Booked';
24         ELSEIF (EXTRACT(YEAR FROM f_mounth) - EXTRACT(YEAR FROM i_date) = 0 AND EXTRACT(MONTH FROM f_mounth) - EXTRACT(MONTH FROM
25         i_date) < 1 AND EXTRACT(DAY FROM f_mounth) - EXTRACT(DAY FROM i_date) >= 1) THEN
26             UPDATE flight SET fl_status = 'Scheduled' WHERE fl_code = flight_code;
27             RETURN 'Scheduled';
28         ELSEIF (EXTRACT(YEAR FROM f_mounth) - EXTRACT(YEAR FROM i_date) = 0 AND EXTRACT(MONTH FROM f_mounth) - EXTRACT(MONTH FROM
29         i_date) < 1 AND EXTRACT(DAY FROM f_mounth) - EXTRACT(DAY FROM i_date) >= 1) THEN
30             UPDATE flight SET fl_status = 'Scheduled' WHERE fl_code = flight_code;
31             RETURN 'Scheduled';
32         ELSEIF (f_time > r_time) then
33             UPDATE flight SET fl_status = 'Ontime' WHERE fl_code = flight_code;
34             RETURN 'Ontime';
35         ELSEIF (r_time < f_act_dep_time) then
36             UPDATE flight SET fl_status = 'Delayed' WHERE fl_code = flight_code;
37             RETURN 'Delayed';
38         ELSEIF (r_time > f_act_dep_time AND r_time < f_act_arr_time) then
39             UPDATE flight SET fl_status = 'Departed' WHERE fl_code = flight_code;
40             RETURN 'Departed';
41         ELSEIF (r_time > f_act_arr_time) then
42             UPDATE flight SET fl_status = 'Arrived' WHERE fl_code = flight_code;
43             RETURN 'Arrived';
44         END IF;
45     ELSE
46         RETURN 'Canceled';
47     END IF;
48 END
49 $$LANGUAGE plpgsql;
50
51 SELECT status_check('1265789542646','EDoazFvG2m','2022-09-05','11:00:00');
52 SELECT status_check('8234814684835','OP1jf24efj','2022-09-12','12:00:00');
53 SELECT status_check('2457468545686','16aghzXbPp','2022-09-15','17:57:00');
54 SELECT status_check('3686835634637','1LeMJzMjw','2022-09-11','10:57:00');
55 SELECT status_check('4568456346864','HfdofJ86Jo','2022-09-22','03:57:00');
56 SELECT status_check('6595567534773','1J8PSkCuMZ','2022-09-09','16:50:00');
57 SELECT status_check('4679346374835','ht186HUK97','2022-09-05','16:50:00');
58
59 UPDATE flight SET fl_status = 'Canceled' WHERE fl_code = 'AV769djvdo';
60 UPDATE flight SET fl_status = 'Canceled' WHERE fl_code = '1DxfMSFQW8';
61 UPDATE flight SET fl_status = 'Canceled' WHERE fl_code = '1343ffvrsv';
62
63 SELECT status_check('6794346347468','1343ffvrsv','2022-09-13','21:50:00');

```

Υλοποίηση ΒΔ

Αφού εκτελεστούν τα παραπάνω scripts με τη σειρά έχει ολοκληρωθεί η υλοποίηση της βάσης και έχουν συμπληρωθεί οι πίνακες με δεδομένα. Παρακάτω παρουσιάζουμε κάποια ενδεικτικά screenshots απο τους πίνακες της βάσης:

```
1 SELECT * FROM public.aircraft
2 ORDER BY airc_code ASC
```

Data Output Explain Messages Notifications

	airc_code [PK] character varying (3)	airc_name character varying (20)	airc_capacity integer	airc_range integer	airc_code character varying (3)
1	222	ggreiser1	43	15	ftg
2	224	ggreiser1	42	24	xlv
3	264	jperrett4	6	96	zxc
4	274	lfosse3	52	70	vxn
5	345	rspringthorpe0	35	66	adf
6	348	rspringthorpe0	37	67	das
7	364	jperrett4	40	100	xio
8	374	lfosse3	53	10	zuy
9	567	bokenden2	101	120	azd
10	577	bokenden2	100	14	zas

```
1 SELECT * FROM public.airport
2 ORDER BY airc_code ASC
```

Data Output Explain Messages Notifications

	airc_code [PK] character varying (3)	airp_name character varying (20)	airp_city character varying (20)	airp_timezone text
1	adf	plybird0	Songjianghe	Asia/Harbin
2	azd	cbeeze2	Vidin	Europe/Sofia
3	das	arp_majal	Itapeva	Asia/Shanghai
4	dgh	Dagmar	Tatariv	Asia/Harbin
5	dza	Sarah Frontman	Sofia	Europe/Sofia
6	fda	Reyna	Nangerang	Asia/Harbin
7	fds	Lela	Jiekeng	Asia/Karachi
8	ftg	el.ben	Koronadal	Asia/Karachi
9	hvx	Harley	Harbin	Asia/Harbin
10	lvx	Ernestus	Iwaki	Asia/Karachi
11	vxn	ablondell3	Moscow	Europe/Moscow
12	wsa	Ode	Bromma	Europe/Luxembourg
13	xcz	Rodney	Himaya	Europe/Luxembourg
14	xio	Alec	Skoroszyce	Europe/Luxembourg
15	xlv	gmaccurlye1	Landi Kotal	Asia/Karachi

```
1 SELECT * FROM public.boarding
2 ORDER BY board_no ASC, board_seat_no ASC, fl_code ASC, t_fl_date ASC
```

Data Output Explain Messages Notifications

	airc_capacity integer	board_no [PK] integer	board_seat_no [PK] integer	t_no character varying (13)	t_pass_fname character varying (20)	t_pass_lname character varying (20)	fl_code [PK] character varying (10)	t_fl_date [PK] date
1	100	1	1	6595567534773	Debbi	Eitone	1J8PSkCuMZ	2022-09-09
2	101	1	1	6541551135448	Valenka	Rosenau	1LeMJJzMjw	2022-09-11
3	35	1	1	4568456346864	Andy	Meyscham	HfdofJ86Jo	2022-09-22
4	100	2	2	5684356435685	Dix	Weedon	1J8PSkCuMZ	2022-09-09
5	101	2	2	3686835634637	Hildegarde	Hollingdale	1LeMJJzMjw	2022-09-11
6	35	2	2	2357686845758	Clareta	Greedier	HfdofJ86Jo	2022-09-22
7	35	3	3	3578456367583	Kailey	Yurivtsev	HfdofJ86Jo	2022-09-22
8	35	4	4	5248348524576	Diana	Bidewel	HfdofJ86Jo	2022-09-22
9	35	5	5	1245568376837	Nadean	Bouttell	HfdofJ86Jo	2022-09-22
10	35	6	6	8939321357357	Kakalina	Stoyell	HfdofJ86Jo	2022-09-22
11	35	7	7	6832462468245	Chrysa	Moncey	HfdofJ86Jo	2022-09-22
12	35	8	8	8353824682458	Zora	Kitney	HfdofJ86Jo	2022-09-22
13	35	9	9	5793556245882	Elberta	Wardall	HfdofJ86Jo	2022-09-22

```
1 SELECT * FROM public.book
2 ORDER BY book_ref ASC
```

Data Output Explain Messages Notifications

	book_ref [PK] character varying (6)	fl_code character varying (10)	fl_date date	book_date timestamp without time zone	book_tprice double precision
1	13vvrv	HfdofJ86Jo	2022-09-20	2022-08-29 00:00:00	77
2	235gsv	HfdofJ86Jo	2022-09-22	2022-08-23 00:00:00	157
3	23hbpd	OPJlf24efj	2022-09-13	2022-08-19 00:00:00	1167
4	36ebda	AV769djvdo	2022-09-07	2022-08-18 00:00:00	42
5	3ght67	1J8PSkCuMZ	2022-09-09	2022-08-27 00:00:00	214
6	3gtnrn	1J8PSkCuMZ	2022-09-02	2022-08-19 00:00:00	65
7	44gh21	EDoazFVG2m	2022-09-09	2022-08-25 00:00:00	55
8	454hsb	OPJlf24efj	2022-09-13	2022-08-14 00:00:00	57
9	46fb13	1DxfMSFQW8	2022-09-10	2022-08-29 00:00:00	80
10	47dbd2	AV769djvdo	2022-09-26	2022-08-30 00:00:00	504

```
1 SELECT * FROM public.flight_arr_act
2 ORDER BY fl_code ASC, arr_date_act ASC
```

Data Output Explain Messages Notifications

	fl_code [PK] character varying (10)	arr_date_act [PK] timestamp without time zone
1	1343ffrvsv	2022-09-14 05:43:00
2	16aghzXbPp	2022-09-13 17:19:00
3	16aghzXbPp	2022-09-14 17:19:00
4	16aghzXbPp	2022-09-15 17:19:00
5	16aghzXbPp	2022-09-25 17:19:00
6	1DxfMSFQW8	2022-09-10 20:04:00
7	1DxfMSFQW8	2022-09-12 20:10:00

```
1 SELECT * FROM public.flight
2 ORDER BY fl_code ASC
```

Data Output Explain Messages Notifications

	fl_code [PK] character varying (10)	airc_range integer	fl_distance integer	depart_airc_code character varying (20)	depart_time_sche time without time zone	arr_airc_code character varying (20)	arr_time_sche time without time zone	fl_duration time without time zone	fl_status character varying (15)
1	1343ffrvsv	14	3	ftg	20:49:00	das	05:43:00	08:54:00	Canceled
2	16aghzXbPp	96	79	azd	14:56:00	vxn	17:19:00	02:23:00	Arrived
3	1DxfMSFQW8	66	31	adf	19:09:00	zxc	20:04:00	00:55:00	Canceled
4	1J8PSkCuMZ	14	3	zxc	16:49:00	adf	17:43:00	00:54:00	Delayed
5	1LeMJJzMjw	96	71	xlv	11:12:00	zxc	01:56:00	14:44:00	Ontime
6	AV769djvdo	96	71	fda	00:12:00	wsa	01:56:00	01:44:00	Canceled

```
1 SELECT * FROM public.flight_date
2 ORDER BY fl_code ASC, fl_date ASC
```

Data Output Explain Messages Notifications

	fl_code [PK] character varying (10)	fl_date [PK] date
1	1343ffvrsv	2022-09-13
2	16aghzXbPp	2022-09-13
3	16aghzXbPp	2022-09-14
4	16aghzXbPp	2022-09-15
5	16aghzXbPp	2022-09-25
6	1DxfMSFQW8	2022-09-10
7	1DxfMSFQW8	2022-09-12
8	1DxfMSFQW8	2022-09-25
9	1DxfMSFQW8	2022-09-26
10	1J8PSkCuMZ	2022-09-02

```
1 SELECT * FROM public.flight_depart_act
2 ORDER BY fl_code ASC, depart_date_act ASC
```

Data Output Explain Messages Notifications

	fl_code [PK] character varying (10)	depart_date_act [PK] timestamp without time zone
1	1343ffvrsv	2022-09-13 20:49:00
2	16aghzXbPp	2022-09-13 14:56:54
3	16aghzXbPp	2022-09-14 14:58:54
4	16aghzXbPp	2022-09-15 14:56:54
5	16aghzXbPp	2022-09-25 14:57:54
6	1DxfMSFQW8	2022-09-10 19:12:54
7	1DxfMSFQW8	2022-09-12 19:16:54
8	1DxfMSFQW8	2022-09-25 19:09:00
9	1DxfMSFQW8	2022-09-26 19:09:00
10	1J8PSkCuMZ	2022-09-02 16:49:00

```
1 SELECT * FROM public.ticket
2 ORDER BY t_no ASC, t_pass_fname ASC, t_pass_lname ASC
```

Data Output Explain Messages Notifications

	t_no [PK] character varying (13)	t_pass_fname [PK] character varying (20)	t_pass_lname [PK] character varying (20)	t_pass_id character varying (15)	t_pass_email character varying (50)	t_pass_tel character varying (20)	t_pass_addr character varying (40)	t_price double precision	book_ref character varying (6)
1	1245568376837	Nadean	Bouttell	3579qhdy	nbouttell2c@posteros.com	+63-893-306-7817	9348 Hagan Park	97	sdvsv3
2	1265789542646	Celka	Nordass	5556dfb	cnordass0@com.com	+54-227-737-2986	8 Union Center	140	gh3j42
3	2354568567658	Darline	Olivello	berb5ge	dolivellok@vk.com	+20-335-764-3563	91942 Victoria Circle	77	7de3gd
4	2356236656522	Benny	Nuemann	6437	bnuemann8@examiner.com	+48-169-439-7624	7 Mosinee Plaza	55	dbd23b
5	2357686845758	Clareta	Greedier	54b76u8sbh	cgreedier9@opera.com	+31-841-156-7956	5478 Shasta Pass	157	235gsv
6	2358568325685	Sada	Tonge	3547nbtbhr	stongen@t-online.de	+55-915-761-8899	26 Northland Parkway	17	g5j4dd
7	2456793476834	Fay	Burel	tnnj5n6	fburel1l@slideshare.net	+55-175-735-7821	251 Buhler Trail	74	bsdv3s
8	2457468545686	Cortney	Campling	jdn4gn	ccampling9@amazonaws.com	+86-255-412-8752	5211 Monica Road	25	54gfd3
9	2457723448683	Carmelita	Kesper	6nnjfs6e5	ckesper1z@infoseek.co.jp	+380-559-223-0048	85487 Scott Lane	95	23nbgd
10	2458138357857	Lucila	Armour	dgtbh3zu	larmour2k@dlon.ne.jp	+86-627-167-8733	28916 Mandrake Drive	57	sfd14

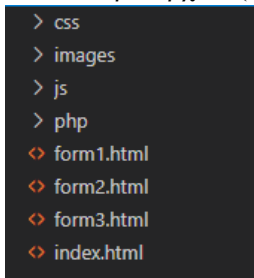
```
1 SELECT * FROM public.ticket_flight
2 ORDER BY t_no ASC, t_pass_fname ASC, t_pass_lname ASC, fl_code ASC, t_fl_date ASC
```

Data Output Explain Messages Notifications

	t_no [PK] character varying (13)	t_pass_fname [PK] character varying (20)	t_pass_lname [PK] character varying (20)	fl_code [PK] character varying (10)	t_fl_date [PK] date	t_price double precision	t_fare character varying (15)
1	1245568376837	Nadean	Bouttell	HfdofJ86Jo	2022-09-22	97	Economy
2	1265789542646	Celka	Nordass	16aghzXbPp	2022-09-13	102	First class
3	1265789542646	Celka	Nordass	EDoazFvG2m	2022-09-06	38	Economy
4	2354568567658	Darline	Olivello	1J8PSkCuMZ	2022-09-02	77	Economy
5	2356236656522	Benny	Nuemann	16aghzXbPp	2022-09-15	55	Economy
6	2357686845758	Clareta	Greedier	HfdofJ86Jo	2022-09-22	157	First class
7	2358568325685	Sada	Tonge	1J8PSkCuMZ	2022-09-13	17	Economy
8	2456793476834	Fay	Burel	1343ffvrsv	2022-09-13	74	Economy

Εφαρμογή Διεπαφής με τη ΒΔ

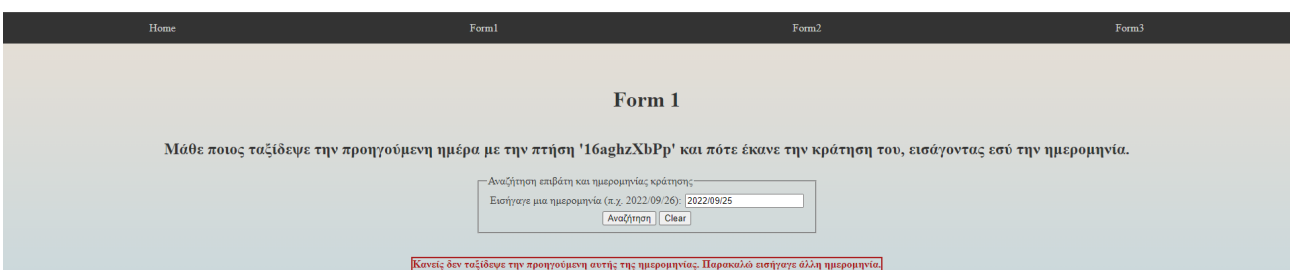
Η διεπαφή μας αποτελεί μια απλή ιστοσελίδα αλληλεπίδρασης τους πελάτη της αεροπορικής με τη βάση δεδομένων. Αποτελείται από τέσσερις σελίδες. Την **αρχική σελίδα** και τις **τρεις φόρμες** αλληλεπίδρασης του πελάτη με τη βάση. Ο κώδικας της σελίδας είναι δομημένος σε τέσσερις φακέλους (css, images, js, php) και τέσσερα αρχεία (html) εκτος φακέλων:



Η **αρχική σελίδα** είναι η παρακάτω, έχει ως σκοπό την ενημέρωση του πελάτη για τους τρεις δυνατούς τρόπους αλληλεπίδρασης του με την βάση και αποτελείται από html και css (αρχεία: **index.html**, **css\index.css**). Η σελίδα αυτή όπως βλέπουμε έχει ένα menu στην κορυφή της το οποίο αποτελεί μια λίστα () η οποία με την κατάλληλη css έχει πάρει την εικόνα που βλέπουμε παρακάτω και κάτω από το menu υπάρχουν δύο τίτλοι και μια λίστα τριών στοιχείων που το κάθε στοιχείο αντιπροσωπεύει μια ενέργεια και έχει ένα κουμπί το οποίο οδηγεί στην αντίστοιχη φόρμα.



Αν επιλέξουμε την πρώτη ενέργεια οδηγούμαστε στην **πρώτη φόρμα**. Η πρώτη φόρμα δίνει τη δυνατότητα στον πελάτη να μάθει ποιος ταξίδεψε την προηγούμενη μέρα από την ημερομηνία που θα εισαγάγει ο ίδιος, με την πτήση '16aghzXbPr'. Αν εισαγάγει μια ημερομηνία έγκυρη θα δει έναν πίνακα τριών στηλών με το όνομα, το επίθετο και την ημερομηνία κράτησης όλων των ατόμων που ταξίδεψαν την προηγούμενη μέρα από αυτή που έδωσε ο πελάτης. Αν εισαγάγει μη έγκυρη ημερομηνία του εμφανίζει μήνυμα που τον ειδοποιεί. Η σελίδα περιλαμβάνει επίσης menu πλοήγησης.



Ο κώδικας της παραπάνω φόρμας αποτελείται από τέσσερα αρχεία (**form1.html**, **css/forms.css**, **js/form1.js**, **php/form1.php**). Το πρώτο αρχείο **form1.html** αποτελεί την html φόρμα στο οποίο αλληλεπιδρά το δευτερο αρχείο **css/forms.css** για να του δώσει την παραπάνω μορφή. Μόλις πατήσει ο πελάτης το κουμπί αναζήτησης τα στοιχεία της φόρμας μεταφέρονται μέσω του τρίτου αρχείου **js/form1.js** στο τέταρτο αρχείο **php/form1.php** μέσω ενός ajax call. Μόλις μεταφερθούν τα δεδομένα στην php, γίνεται σύνδεση με τη βάση, πραγματοποιείται το υπολογισμός της προηγούμενης ημέρας από αυτή που έχει εισάγει ο πελάτης και πραγματοποιείται το query προς τη βάση. Μόλις επιστρέψει τα data η βάση, τοποθετούνται σε πίνακα html και στέλνονται στην js η οποία τα τα δώσει στην html για να τα παρουσιάσει στον πελάτη. Το **select query** αποτελεί ένα διπλό inner join ώστε να πάρουμε πληροφορίες από τους πίνακες ticket_flight και book, οι οποίοι δεν συνδέονται μεταξύ τους μέσω foreign key και σαν αποτέλεσμα χρειαζόμαστε το ενδιάμεσο inner join προς τον πίνακα ticket. Στο where του query τοποθετούμε δύο συνθήκες, μια για να πάρουμε πληροφορίες για την πτήση που θέλουμε μόνο και μια για να πάρουμε πληροφορίες μόνο για την προηγούμενη της ημερομηνίας που μας έχει δώσει ο πελάτης.

Αν επιλέξουμε τη δεύτερη ενέργεια οδηγούμαστε στη **δεύτερη φόρμα**. Η δεύτερη φόρμα δίνει τη δυνατότητα στον πελάτη να μάθει τους πέντε πιο δημοφιλής προορισμούς για το έτος που θα εισαγάγει ο ίδιος. Αν εισαγάγει ένα έτος έγκυρο θα δει έναν πίνακα με τις πέντε πιο δημοφιλής πόλεις. Αν εισάγει μη έγκυρο έτος του εμφανίζει μήνυμα που τον ειδοποιεί. Η σελίδα περιλαμβάνει επίσης menu πλοήγησης.

Ο κώδικας της παραπάνω φόρμας αποτελείται από τέσσερα αρχεία (**form2.html**, **css/forms.css**, **js/form2.js**, **php/form2.php**). Το πρώτο αρχείο **form2.html** αποτελεί την html φόρμα στο οποίο αλληλεπιδρά το δευτερο αρχείο **css/forms.css** για να του δώσει την παραπάνω μορφή. Μόλις πατήσει ο πελάτης το κουμπί αναζήτησης τα στοιχεία της φόρμας μεταφέρονται μέσω του τρίτου αρχείου **js/form2.js** στο τέταρτο αρχείο **php/form2.php** μέσω ενός ajax call. Μόλις μεταφερθούν τα δεδομένα στην php, εκτελούμε σύνδεση με τη βάση και το query προς τη βάση. Μόλις επιστρέψει τα data η βάση, τοποθετούνται σε πίνακα html και στέλνονται στην js, η οποία τα τα δώσει στην html για να τα παρουσιάσει στον πελάτη. Το **select query** προς τη βάση αποτελείται από δύο inner join για να μπορέσουμε να πάρουμε δεδομένα από τους πίνακες flight_date και airport μέσω του πίνακα flight. Το query περιλαμβάνει επίσης ένα where το οποίο μας δίνει πληροφορίες μόνο για το έτος που έχει εισάγει ο πελάτης, ένα group by για να ομαδοποιήσουμε τα αποτελέσματα, ένα order by για να ταξινομήσουμε τα δεδομένα με φθίνουσα σειρά με βάση το πόσες φορές επισκέφθηκαν αυτή τη πόλη και ένα limit για να πάρουμε τους πέντε πρώτους top προορισμούς.

Αν επιλέξουμε την τρίτη ενέργεια οδηγούμαστε στην τρίτη σελίδα. Η τρίτη σελίδα αποτελείται από τέσσερις φόρμες και δίνει τη δυνατότητα στον πελάτη να εισάγει, διαγράψει, δει και ενημερώσει αεροσκάφη του πίνακα aircraft. Παρακάτω φαίνονται οι τρόποι αλληλεπίδρασης με αυτές τις φόρμες.

Home

Form1

Form2

Form3

Form 3

Εισήγαγε, διέγραψε, δες, και ενημέρωσε αεροσκάφη.

Εισαγωγή Αεροσκάφους

Εισήγαγε τον κωδικό αεροσκάφους:

Εισήγαγε το όνομα αεροσκάφους:

Εισήγαγε την χωρητικότητα αεροσκάφους:

Εισήγαγε την εμβέλεια αεροσκάφους:

Επέλεξε το αεροδρόμιο του αεροσκάφους:

Songjianghe

Εισαγωγή Clear

Διαγραφή Αεροσκάφους

Εισήγαγε κωδικό αεροσκάφους:

Διαγραφή Clear

Αναζήτηση αεροσκάφους προς επεξεργασία

Εισήγαγε κωδικό αεροσκάφους:

Αναζήτηση Clear

Επεξεργασία Αεροσκάφους

Εισήγαγε τον κωδικό αεροσκάφους:

Εισήγαγε το όνομα αεροσκάφους:

Εισήγαγε την χωρητικότητα αεροσκάφους:

Εισήγαγε την εμβέλεια αεροσκάφους:

Επέλεξε το αεροδρόμιο του αεροσκάφους:

Songjianghe

Επεξεργασία Clear

Home

Form1

Form2

Form3

Form 3

Εισήγαγε, διέγραψε, δες, και ενημέρωσε αεροσκάφη.

Εισαγωγή Αεροσκάφους

Εισήγαγε τον κωδικό αεροσκάφους:

Εισήγαγε το όνομα αεροσκάφους:

Εισήγαγε την χωρητικότητα αεροσκάφους:

Εισήγαγε την εμβέλεια αεροσκάφους:

Επέλεξε το αεροδρόμιο του αεροσκάφους:

Athens

Εισαγωγή Clear

Διαγραφή Αεροσκάφους

Εισήγαγε κωδικό αεροσκάφους:

Διαγραφή Clear

Αναζήτηση αεροσκάφους προς επεξεργασία

Εισήγαγε κωδικό αεροσκάφους:

Αναζήτηση Clear

Home

Form1

Form2

Form3

Form 3

Εισήγαγε, διέγραψε, δες, και ενημέρωσε αεροσκάφη.

Εισαγωγή Αεροσκάφους

Εισήγαγε τον κωδικό αεροσκάφους:

Εισήγαγε το όνομα αεροσκάφους:

Εισήγαγε την χωρητικότητα αεροσκάφους:

Εισήγαγε την εμβέλεια αεροσκάφους:

Επέλεξε το αεροδρόμιο του αεροσκάφους:

Songjianghe

Εισαγωγή Clear

Διαγραφή Αεροσκάφους

Εισήγαγε κωδικό αεροσκάφους:

Διαγραφή Clear

Αναζήτηση αεροσκάφους προς επεξεργασία

Εισήγαγε κωδικό αεροσκάφους:

Αναζήτηση Clear

Aircraft Code	Aircraft Name	Aircraft Capacity	Aircraft Range	Airport Code
222	ggreiser1	43	15	ftg

Home

Form1

Form2

Form3

Form 3

Εισήγαγε, διέγραψε, δες, και ενημέρωσε αεροσκάφη.

Εισαγωγή Αεροσκάφους

Εισήγαγε τον κωδικό αεροσκάφους:

Εισήγαγε το όνομα αεροσκάφους:

Εισήγαγε την χωρητικότητα αεροσκάφους:

Εισήγαγε την εμβέλεια αεροσκάφους:

Επέλεξε το αεροδρόμιο του αεροσκάφους:

Songjianghe

Εισαγωγή Clear

Διαγραφή Αεροσκάφους

Εισήγαγε κωδικό αεροσκάφους:

Διαγραφή Clear

Αναζήτηση αεροσκάφους προς επεξεργασία

Εισήγαγε κωδικό αεροσκάφους:

Αναζήτηση Clear

Aircraft Code	Aircraft Name	Aircraft Capacity	Aircraft Range	Airport Code
222	ggreiser1	43	15	ftg

Επεξεργασία Αεροσκάφους

Εισήγαγε τον κωδικό αεροσκάφους:

Εισήγαγε το όνομα αεροσκάφους:

Εισήγαγε την χωρητικότητα αεροσκάφους:

Εισήγαγε την εμβέλεια αεροσκάφους:

Επέλεξε το αεροδρόμιο του αεροσκάφους:

Athens

Επεξεργασία Clear

Ο κώδικας της παραπάνω φόρμας αποτελείται από τέσσερα αρχεία (form3.html, css\forms.css, js\form3.js, php\form3.php). Το πρώτο αρχείο form3.html αποτελεί την html στο οποίο αλληλεπιδρά το δεύτερο αρχείο css\forms.css για να του δώσει την παραπάνω μορφή. Το μόνο ενδιαφέρον εδώ σε σχέση με τις προηγούμενες φόρμες, είναι ότι στις φόρμες εισαγωγής και ενημέρωσης αεροσκαφών, στο πεδίο “Επέλεξε το αεροδρόμιο του αεροσκάφους:” δίνεται στον πελάτη η επιλογή πόλης και η τιμή της πόλης είναι το airport code διότι αυτή η πληροφορία αποθηκεύεται στον πίνακα aircraft. Μόλις πατήσει ο πελάτης το ένα από τα κουμπιά υποβολής μιας φόρμας, τα στοιχεία της φόρμας μεταφέρονται μέσω του τρίτου αρχείου js\form3.js στο τέταρτο αρχείο php\form3.php μέσω ενός ajax call. Μόλις μεταφερθούν τα δεδομένα στην php, εκτελούμε σύνδεση με τη βάση και ανάλογα με την φόρμα που συμπληρώνει ο πελάτης εκτελείται το αντίστοιχο query προς τη βάση. Τα **queries** αποτελούν τα τέσσερα βασικά queries (**create/insert**, **delete**, **read/select** και **update**) προς μία βάση δεδομένων, λεγόμενα και ως **CRUD**.