ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΤΙΤΛΟΣ (π.χ. FlightsDB)

Υπότιτλος (π.χ. Βάση Δεδομένων Πτήσεων)

**Πρώτο Παραδοτέο**

Αριθμός Ομάδας (π.χ. Ομάδα 60)

Ονοματεπώνυμο1 ΑΕΜ1 email1@ece.auth.gr

Ονοματεπώνυμο2 ΑΕΜ2 email2@ece.auth.gr

Ονοματεπώνυμο3 ΑΕΜ3 email3@ece.auth.gr

ΗΜΕΡΟΜΗΝΙΑ

Περιεχόμενα

1. Εισαγωγή 4
   1. Σκοπός Εφαρμογής 4
   2. Περιγραφή Εφαρμογής 4
   3. Απαιτήσεις Εφαρμογής σε Δεδομένα 4
2. Κατηγορίες Χρηστών και Απαιτήσεις τους 5
3. Μοντέλο Οντοτήτων/Συσχετίσεων 6
   1. Γενική Περιγραφή: 6
   2. Καθορισμός Οντοτήτων 7
   3. Καθορισμός Συσχετίσεων 8
   4. Διάγραμμα Οντοτήτων/Συσχετίσεων 8
4. Σχεσιακό Μοντέλο 10
   1. Πεδία Ορισμού 10
   2. Σχέσεις 10
   3. Σχεσιακό Σχήμα 10
   4. Όψεις 11
5. Παραδείγματα 13
   1. Παραδείγματα Πινάκων 13
   2. Παραδείγματα Ερωτημάτων 13

# Εισαγωγή

## Σκοπός Εφαρμογής

{Αναφέρετε συνοπτικά ποιος είναι ο λόγος ύπαρξης της ΒΔ/εφαρμογής σας}

(π.χ. για τη FlightsDB, ο σκοπός είναι η κατασκευή μιας ΒΔ που θα περιέχει δεδομένα για πτήσεις. Πέρα από την καταγραφή των δρομολογίων, η εφαρμογή θα επιτρέπει την κράτηση εισιτηρίων, την προσπέλαση με χρήση ερωτημάτων για τη διαθεσιμότητα, κτλ.)

## Περιγραφή Εφαρμογής

{Περιγράψτε πως θα λειτουργεί η εφαρμογή σας, δηλαδή ποια είναι τα δεδομένα που θα αποθηκεύονται και πως θα τη χρησιμοποιούν οι χρήστες}

(π.χ. για τη FlightsDB, τα δεδομένα που αποθηκεύονται είναι πτήσεις, αεροδρόμια, κτλ., ενώ θα τη χρησιμοποιούν ελεγκτές εναέριας κυκλοφορίας, υπάλληλοι αεροπορικών εταιριών, καταναλωτές, κτλ.)

## Απαιτήσεις Εφαρμογής σε Δεδομένα

{Κάντε μια εκτίμηση για το μέγεθος της ΒΔ, εξηγώντας τους όγκους δεδομένων που αναμένεται να αποθηκεύσετε - μπορείτε να αναζητήσετε στοιχεία online}

(π.χ. για τη FlightsDB αναμένεται να έχουμε ~100000 κωδικούς πτήσεων – δηλαδή 100000 πτήσεις την ημέρα, επίσης αναμένονται 150 επιβάτες ανά πτήση κατά μέσο όρο, κτλ.)

# Κατηγορίες Χρηστών και Απαιτήσεις τους

{Αναφέρετε όλους τους πιθανούς χρήστες του συστήματός σας και καταγράψτε επιγραμματικά τις απαιτήσεις τους}

Παράδειγμα για τη FlightsDB:

Διαχειριστής:

Έχει ως ευθύνη την πλήρη διαχείριση της βάσης δεδομένων. Τα δικαιώματά του περιλαμβάνουν:

* Πρόσβαση σε όλο το πλήθος των δεδομένων της βάσης, συμπεριλαμβανομένων των στοιχείων επικοινωνίας όλων των χρηστών με σκοπό την επικοινωνία με τους τελευταίους εάν κρίνεται απαραίτητο.
* Δημιουργία νέων ρόλων χρηστών
* …

Υπάλληλος Αεροπορικής Εταιρείας:

Έχει ως ευθύνη τη διαχείριση των κρατήσεων. Τα δικαιώματά του περιλαμβάνουν:

* Πρόσβαση σε δεδομένα που αφορούν τις πτήσεις της αεροπορικής εταιρείας, συμπεριλαμβανομένων των αγορών εισιτηρίων.
* Πρόσβαση στο προφίλ της εταιρείας και δυνατότητα ενημέρωσής του
* …

# Μοντέλο Οντοτήτων/Συσχετίσεων

## Γενική Περιγραφή:

Entities & Relationships between them:

The entities of the E-R model are: User, Area, Route, Vehicle, Violations, Tolls, Parking Slot, and Traffic Problem. For every User there is at least one Vehicle, each Vehicle can only be located in one Area at a time and can follow only one Route at time, each Vehicle must have a starting and a destination Area, each Route is consisted of many different Areas and each Area can be a part of many different Routes. An Area can have Parking Slots, Tolls and Traffic Lights.

Attributes of Entities and Relationships:

* For every User there must be a *name,* a *gender,* a unique *id* which is the primary key, an *age* and a *mobile number.*
* EveryViolationhas a *unique id*, a *fee* and a *type* (red light, stop sign, speeding etc…)
* Every Area has a unique i*d* which is the primary key, a *name* and must have a *traffic* *metric* which shows the traffic situation in the Area.
* Neighboring Areas need to have an *area1\_id and area2\_id* which indicates the neighboring Areas ID’s.Their combination is the primary key.
* Every Route has a unique *id* which is the primary key and a *duration* which determines how long the Route is estimated to last (in minutes).
* Each Route-Area relationship needs a *route\_ID* and an *area\_ID* to indicate the relation between an Area and a Route. Their combination is the primary key.
* Every Vehicle must have a unique *license\_plate* and a *type* which defines the type of the car (SUV, small Car, large Car, Limo, etc). It must have also its ownUser and it follows a Route. It has a temporary location Area, has started its Route from a starting Area and is heading to a destinationArea.
* A Parking Slot belongs to a specific Area*,* has a unique *id*, a *location* and a *status* which shows if it is occupied or not.
* Traffic Light belongs to a specific Area*,* has a unique *id*, a *status* (red or green), a *location* and a *duration* (in minutes)*.*
* Tolls belongs to a specific Area, theyhave a *toll id*, a *location*which is the placement of the toll and a *toll* which defines the price of the toll.

\*Attributes written in *italics*

*\**Entities written in Underlines

Assumptions:

* Every Vehicle has a unique User. There can be no vehicle with two different owners.
* A Vehicle can have alternating locations depending on time. For example, if a Vehicle traverses through a Route it will change from two to multiple locations in order to arrive in the final destination.
* An Area can have multiple Vehicles simultaneously.
* The current location of each Vehicle is being updated by GPS in a fixed time interval schedule.
* The Route of each Vehicle is decided by the computing cloud that uses a Shortest Path Algorithm, taking into account the traffic metrics, the Traffic lights and the Tolls of the Areas. For example, the User with id = 1, starts from Area with id = 3, chooses the destination Area with id = 10, the computing cloud, taking into consideration the attributes that were mentioned previously, decides that the optimal Route is the one with id = 2, which passes through Areas with id = 1, 4, 6, 10 and each Area is neighboring to the previous and the next one.
* Two different Vehicles can be matched with the same Route.
* The traffic metric of an Area is computed as a fuzzy value after counting the Vehicles that are located in the area and is updated every a standard amount of minutes.
* The status of every Parking Slot is being updated by the embedded systems of the sensors that are placed in the Parking Slot.
* The violations are recorded by cameras and are automatically written to the database.

## Καθορισμός Οντοτήτων

{Αναφέρετε τις οντότητες της βάσης δεδομένων, καθώς και τα γνωρίσματά τους.}

Παράδειγμα για τη FlightsDB:

|  |  |  |
| --- | --- | --- |
| **Όνομα Οντότητας** | User | |
| **Περιγραφή** | Entity in whom we save every user in the system | |
| **Ιδιότητες** | Ισχυρή Οντότητα, … {αναφέρετε επίσης υπο/υπερκλάσεις} | |
| **Γνωρίσματα** | id | |
| Name | |
| Age | |
| Gender |  |

## Καθορισμός Συσχετίσεων

{Αναφέρετε τις συσχετίσεις της βάσης δεδομένων.}

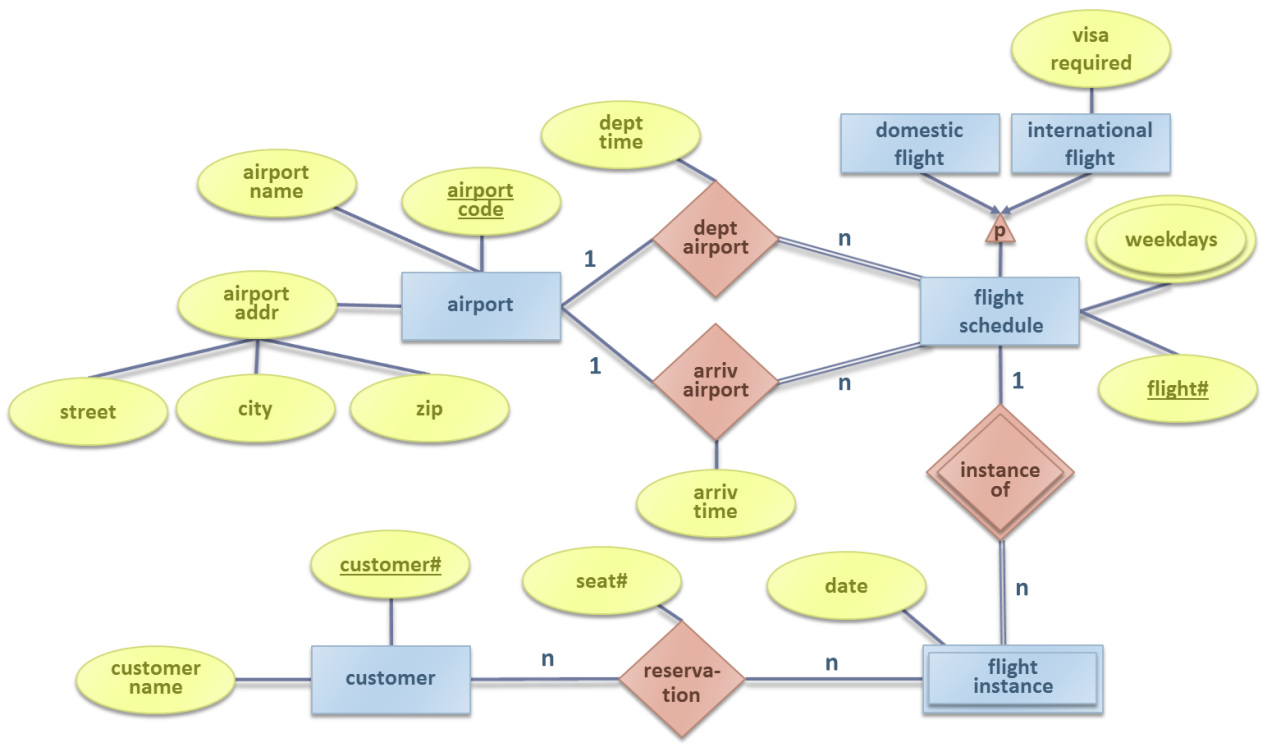
Παράδειγμα για τη FlightsDB:

|  |  |
| --- | --- |
| **Όνομα Συσχέτισης** | Flight\_Has\_Airport |
| **Περιγραφή** | Κάθε πτήση πρέπει να έχει ένα αεροδρόμιο αναχώρησης και ένα αεροδρόμιο προορισμού |
| **Ιδιότητες** | Has-A {αναφέρετε αν είναι Is-A και αν είναι Αναδρομική, Προσδιορίζουσα, Τριαδική} |
| **Λόγος πληθικότητας** | 1:2 |
| **Συμμετοχή** | Ολική Συμμετοχή του Flight |
| Μερική Συμμετοχή του Airport |
| **Γνωρίσματα** | - |

## Διάγραμμα Οντοτήτων/Συσχετίσεων

{Δείξτε το διάγραμμα Ο/Σ για τη βάση. Το διάγραμμα μπορείτε να το κατασκευάσετε σε πρόγραμμα της επιλογής σας, ωστόσο θα πρέπει να ακολουθεί το συμβολισμό Chen (δηλαδή οντότητες ως παραλληλόγραμμα, συσχετίσεις ως ρόμβοι, διπλή γραμμή για υποχρεωτική συμμετοχή, κτλ.)}

Παράδειγμα για τη FlightsDB:



# Σχεσιακό Μοντέλο

## Πεδία Ορισμού

{Προσδιορίστε τα πεδία ορισμού που θα χρησιμοποιήσετε για το σχεσιακό μοντέλο.}

Παράδειγμα για τη FlightsDB:

|  |  |
| --- | --- |
| **Πεδίο Ορισμού** | **Τύπος** |
| Ακέραιος | INT |
| Κωδ\_Αεροδρομίου | CHAR(3) |
| Απλό\_Αλφαριθμητικό | VARCHAR(25) |
| Διεύθυνση | VARCHAR(35) |
| … | … |

## Σχέσεις

{Προσδιορίστε τις σχέσεις του σχεσιακού μοντέλου.}

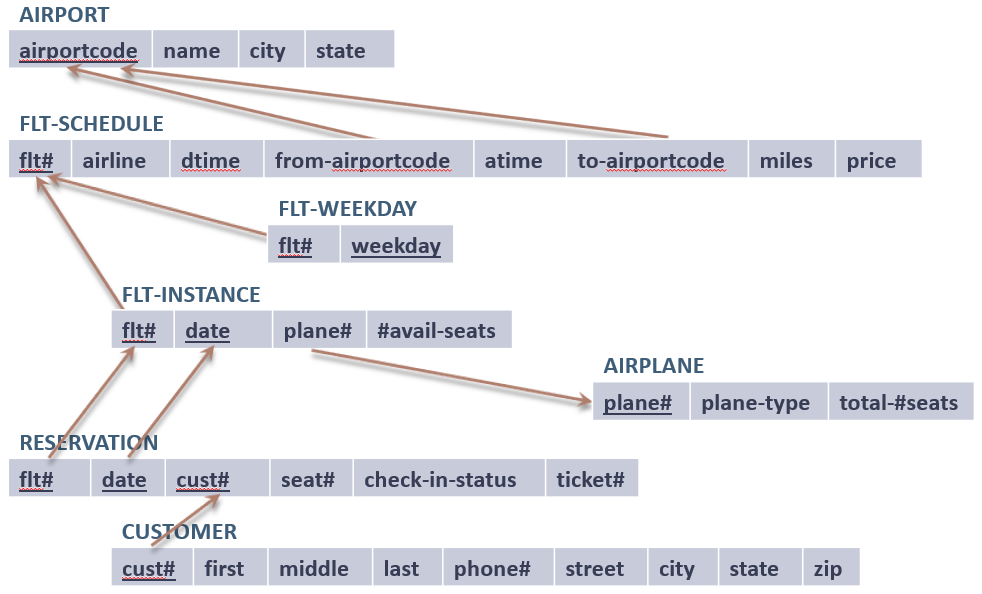
Παράδειγμα για τη FlightsDB:

|  |  |
| --- | --- |
| **Όνομα Σχέσης** | Airport |
| **Γνωρίσματα:** | |
| **Όνομα** | **Τύπος** |
| airport\_code | Κωδ\_Αεροδρομίου |
| name | Απλό\_Αλφαριθμητικό |
| city | Διεύθυνση |
| country | Διεύθυνση |
| **Περιορισμοί Ακεραιότητας:** | |
| **Πρωτεύον Κλειδί** | airport\_code |
| **Ξένα Κλειδιά** | - {αναφέρετε κλειδί και σχ. σχέση, π.χ. air\_code 🡪 Airport} |
|  |

## Σχεσιακό Σχήμα

{Δείξτε το σχεσιακό σχήμα για τη βάση. Το σχήμα μπορείτε να το κατασκευάσετε σε πρόγραμμα της επιλογής σας, ωστόσο θα πρέπει να ακολουθεί το συμβολισμό του μαθήματος (δηλαδή οι σχέσεις ως κεφαλίδες πινάκων, τα ξένα κλειδιά ως βέλη μιας κατεύθυνσης, κτλ.)}

Παράδειγμα για τη FlightsDB (προσοχή το παράδειγμα δεν είναι πλήρως αντίστοιχο με το διάγραμμα E/R που δόθηκε παραπάνω – για την εργασία θα πρέπει να είναι πλήρως αντίστοιχα):



## Όψεις

{Κατασκευάστε χρήσιμες όψεις για τη βάση. Κάθε όψη θα πρέπει να οριστεί με σχεσιακή άλγεβρα.}

Παράδειγμα για τη FlightsDB:

(έστω οι σχέσεις:

* FLIGHT(flight\_id, airline, fromairport, toairport, price, plane\_id)
* AIRPLANE(plane\_id, plane\_name)

)

Μια όψη που περιέχει όλες τις αεροπορικές εταιρίες που υπάρχουν στο σύστημα και τα ονόματα των αεροπλάνων που χρησιμοποιούν είναι η παρακάτω:

**ρ**AIRLINES(**π**airline, plane\_name(**π**airline, plane\_id(FLIGHT) Εικόνα 4 **π**plane\_id, plane\_name(AIRPLANE)))

# Παραδείγματα

## Παραδείγματα Πινάκων

{Δώστε ενδεικτικά παραδείγματα εγγραφών για κάθε πίνακα της βάσης.}

Παράδειγμα για τον πίνακα Airport της FlightsDB:

|  |  |  |  |
| --- | --- | --- | --- |
| **airport\_code** | **Name** | **city** | **country** |
| SKG | Makedonia | Thessaloniki | Greece |
| ATH | Eleftherios Venizelos | Athens | Greece |
| KVA | Megas Alexandros | Kavala | Greece |

Εκτίμηση για τον αριθμό των εγγραφών: ~40000

## Παραδείγματα Ερωτημάτων

{Δώστε ενδεικτικά παραδείγματα χρήσιμων ερωτημάτων.}

Παράδειγμα για τη FlightsDB:

(έστω οι σχέσεις:

* CUSTOMER(cust\_id, firstname, lastname, phone, street, city, zip)
* RESERVATION(flight\_id, date, cust\_id, ticket\_no, seat\_no)

)

Για μια πτήση (έστω την AA101) υποθέτουμε ότι ο/η αεροσυνοδός θα ήθελε να έχει τη λίστα των επιβατών μαζί με χρήσιμες πληροφορίες για το check in (id επιβάτη, αριθμός εισιτηρίου, θέση, όνομα και επώνυμο για κάθε επιβάτη). Εκτελούμε το παρακάτω ερώτημα:

**π**ticket\_no, seat\_no, cust\_id(**σ**flight\_id=AA101(RESERVATION)) Εικόνα 5 **π**cust\_id, firstname, lastname(CUSTOMER)