

Χριστονάσης Αντώνιος Μάριος

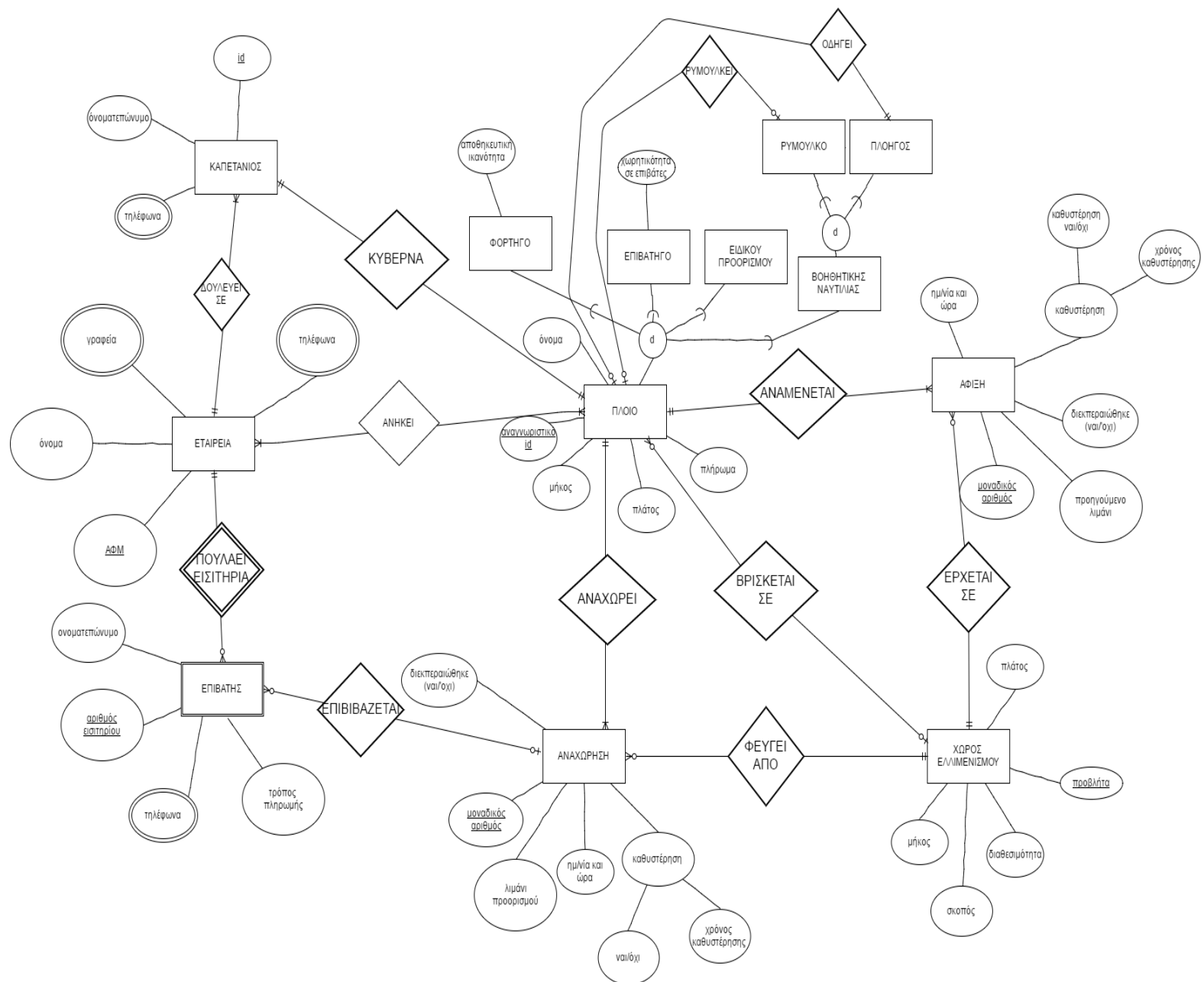
7661

2016-17

Σχεδιασμός μιας Βάσης Δεδομένων ενός Λιμανιού

Σκοπός μου στο συγκεκριμένο project ήταν ο σχεδιασμός της Βάσης Δεδομένων ενός Λιμανιού, με κύριες λειτουργίες τον χειρισμό αφίξεων, αναχωρήσεων και χώρων ελλιμενισμού. Ξεκινήσαμε τη διαδικασία δημιουργώντας ένα εκτεταμένο διάγραμμα οντοτήτων συσχετίσεων, που περιγράφει αναλυτικά τον μικρόκοσμό μας και εξυπηρετεί τους βασικούς στόχους της βάσης μας.

EERD



Ένα πολύ βασικό κομμάτι του σχεδιασμού μίας βάσης δεδομένων είναι **το διάγραμμα οντοτήτων συσχετίσεων**. Ένα καλό διάγραμμα οντοτήτων συσχετίσεων οδηγεί σωστά την υπόλοιπη διαδικασία υλοποίησης της βάσης. Αντίστοιχα, ένα κακό *eerd* οδηγεί σε μια κακή υλοποίηση. Η διαδικασία αυτή αποτελεί περισσότερο τέχνη παρά μεθοδολογία, γι' αυτό και περάσαμε από διάφορα στάδια και αρκετό brainstorming για να καταλήξουμε στο τελικό διάγραμμα. Η εμπειρία που αποκτήσαμε μέσα στο εξάμηνο με σχεδιασμό πολλών διαγραμμάτων μας βοήθησε αρκετά. Χρησιμοποιήσαμε το online εργαλείο draw.io.

Το πρώτο βήμα στη διαδικασία αποτελεί ο ορισμός των **οντοτήτων** του διαγράμματός μας. Επιλέξαμε να υλοποιήσουμε το πλοίο και τα είδη πλοίων σαν σχέσεις τύπου είναι. Έπειτα, ορίσαμε σαν οντότητα τον χώρο ελλιμενισμού. Στη συνέχεια, προσθέσαμε την άφιξη, την αναχώρηση, τον καπετάνιο του πλοίου και την εταιρεία που ανήκει το πλοίο σαν βασικές οντότητες. Αξίζει να σημειωθεί η αμφιβολία μας κατά τη διαδικασία της σχεδίασης για το ποιες από αυτές τις οντότητες αποτελούν ασθενείς οντότητες. Η λογική ήταν πως χωρίς τη σχέση τους με το πλοίο, δεν έχουν λόγο ύπαρξης. Αντίθετα με τον χώρο ελλιμενισμού που υπάρχει ανεξάρτητα των πλοίων. Παρόλαυτα, επιλέξαμε να τις ορίσουμε σαν ισχυρές οντότητες καθώς δεν θεωρήσαμε πολύ πιθανό η σχέση τους με το πλοίο να μην υπάρχει. Άλλωστε κάθε πλοίο οφείλει να έχει κάποιον καπετάνιο και να ανήκει σε κάποια εταιρεία. Επίσης, θεωρήσαμε πως το λιμάνι είναι αρκετά ενεργό(έχει αφίξεις και αναχωρήσεις συνέχεια) και πως αν ένα πλοίο δεν έχει αφίξεις και αναχωρήσεις θα φύγει από το λιμάνι και θα πάει σε κάποιο ναυπηγείο. Γι' αυτό και τις ορίσαμε σαν ισχυρές οντότητες, αντίθετα με τον επιβάτη. Ο επιβάτης αν δεν αγοράσει εισιτήριο(που είναι και η προσδιορίζουσα σχέση του με την εταιρεία) δεν υφίσταται στη βάση μας σαν επιβάτης. Αντίστοιχη λογική ισχύει και αν η εταιρεία δεν πουλάει εισιτήρια, κάτι που συμβαίνει αν τις έχουν τελειώσει ή αν είναι εταιρεία που δεν διαθέτει επιβατηγά πλοία. Γι' αυτό λοιπόν ο επιβάτης ορίστηκε σαν ασθενής οντότητα.

Πάμε στα **γνωρίσματα** που επιλέξαμε για τις οντότητές μας. Όσον αφορά το πλοίο, το όνομα, το μήκος και το πλάτος του είναι αρκετά προφανή. Επίσης επιλέξαμε να υπάρχει ένα αναγνωριστικό id μοναδικό για κάθε πλοίο, που θα πάρει το ρόλο του πρωτεύοντος κλειδιού. Στο γνώρισμα πλήρωμα θα αποθηκεύουμε το πλήθος των ατόμων που αποτελούν το πλήρωμα του κάθε πλοίου. Όπως είναι φυσικό, όλα αυτά τα γνωρίσματα κληρονομούνται στις υποκλάσεις των σχέσεων τύπου είναι. Προσθέσαμε απλά στο φορτηγό την αποθηκευτική ικανότητα σε τόνους και στο επιβατηγό την χωρητικότητα σε επιβάτες, δύο γνωρίσματα που δεν χρειαζόμαστε στα άλλα είδη πλοίων.

Για τις οντότητες άφιξη και αναχώρηση τα γνωρίσματα που επιλέξαμε είναι πανομοιότυπα. Η ημερομηνία και ώρα που είναι προγραμματισμένη να γίνει η κάθε άφιξη/αναχώρηση. Ένας μοναδικός αριθμός που δρα σαν πρωτεύων κλειδί. Ένα σύνθετο γνώρισμα καθυστέρηση που μας ενημερώνει αν υπήρξε καθυστέρηση και πόσος ήταν ο χρόνος καθυστέρησης σε λεπτά από την προγραμματισμένη ώρα. Μία λογική μεταβλητή για το αν έχει διεκπεραιωθεί ή όχι. Τέλος, για την άφιξη το όνομα του προηγούμενου λιμανιού και αντίστοιχα για την αναχώρηση το όνομα του λιμανιού προορισμού.

Για τους χώρους ελλιμενισμού κάναμε τις εξής παραδοχές. Ο κάθε χώρος ελλιμενισμού βρίσκεται σε διαφορετική προβλήτα. Γι' αυτό και η προβλήτα αποτελεί πρωτεύων κλειδί. Επίσης ο κάθε χώρος ελλιμενισμού διαθέτει πολλές θέσεις ιδίου μήκους και πλάτος. Γι' αυτό λοιπόν κρατάμε το μέγιστο μήκος και πλάτος της κάθε θέσης, αλλά και την διαθεσιμότητα σε αριθμό θέσεων του εκάστοτε χώρου ελλιμενισμού. Τέλος, στο γνώρισμα σκοπός κρατάμε ένα κειμενάκι το οποίο περιγράφει το είδος των πλοίων που δέχεται ο

συγκεκριμένος χώρος ελλιμενισμού. Θεωρήσαμε λογικό να υπάρχουν για παράδειγμα άλλοι χώροι ελλιμενισμού που ταιριάζουν σε φορτηγά, άλλοι χώροι ελλιμενισμού για επιβατηγά και σκάφη και άλλοι για ειδικά σκάφη όπως τα ρυμουλκά. Χρησιμοποιώντας όλες αυτές τις πληροφορίες μαζί με τα γνωρίσματα των πλοίων μας, μπορούν να γίνουν οι κατάλληλες επιλογές για το που θα αράξει κάθε πλοίο και αντίστοιχα κατάλληλες επιλογές για τις αναχωρήσεις και τις αφίξεις μας.

Στον καπετάνιο κρατάμε επίσης ένα id που δρα σαν πρωτεύων κλειδί, το ονοματεπώνυμό του, αλλά και σαν πλειότιμο γνώρισμα τα τηλέφωνα του(κινητό/οικίας κλπά). Το ίδιο κάναμε για τα τηλέφωνα της εταιρείας αλλά και του επιβάτη. Επίσης πλειότιμο γνώρισμα επιλέξαμε και για τα γραφεία της εταιρείας, αφού πιθανότατα διαθέτει γραφεία σε περισσότερες από μία τοποθεσίες. Για την εταιρεία επίσης κρατάμε το όνομά της αλλά και το ΑΦΜ της, που εδώ είναι πρωτεύων κλειδί.

Τέλος, για τον επιβάτη κρατάμε επίσης τον αριθμό εισιτηρίου του, που εδώ είναι πρωτεύων κλειδί. Να σημειωθεί πως κάναμε την παραδοχή ότι το κάθε εισιτήριο έχει διαφορετικό αριθμό για τον κάθε επιβάτη. Κρατάμε επίσης το ονοματεπώνυμό του, αλλά και τον τρόπο με τον οποίο πλήρωσε, σε περίπτωση που θέλουμε μελλοντικά να του προσφέρουμε σχετικές διευκολύνσεις ή προσφορές ανάλογα με τις προτιμήσεις πληρωμής του.

Ας τελειώσουμε με τις **συσχετίσεις** μεταξύ των οντοτήτων, τις οποίες επιλέξαμε να δείξουμε στο διάγραμμά μας με το λεγόμενο «πόδι της χήνας». Το πλοίο έχει συσχέτιση ένα προς ένα με τον καπετάνιο, καθώς ένας και μόνο ένας καπετάνιος κυβερνάει ένα και μόνο ένα πλοίο. Παρόλο που η συσχέτιση ένα προς ένα συνήθως υποδηλώνει πιθανό λάθος στη σχεδίασή μας, δεν επιλέξαμε εδώ να κάνουμε τον καπετάνιο γνώρισμα στο πλοίο, αφού έχει δικά του γνωρίσματα που τον χαρακτηρίζουν, τα οποία δεν έχει πολύ νόημα να βάλουμε σαν γνωρίσματα στο πλοίο.

Το πλοίο τώρα έχει συσχέτιση ένα με υποχρεωτική συμμετοχή προς πολλά με υποχρεωτική συμμετοχή με την άφιξη και την αναχώρηση. Αυτό συμβαίνει γιατί θεωρήσαμε πως κρατάμε ένα ιστορικό αφίξεων και αναχωρήσεων, αλλά και γιατί κάναμε την παραδοχή πως κάθε πλοίο που βρίσκεται στο λιμάνι θα σχετίζεται υποχρεωτικά με μία ή πολλές αφίξεις/αναχωρήσεις. Αλλιώς, αν δεν σχετίζεται με αναχώρηση/άφιξη, θεωρούμε ότι πρέπει να πάει σε κάποιο ναυπηγείο και άρα να μην κρατιέται πλέον στη βάση μας. Επίσης, θεωρήσαμε ότι γνωρίζουμε και το μελλοντικό πρόγραμμα αφίξεων/αναχωρήσεων των πλοίων μας σε βάθος κάποιων μηνών. Τέλος, η κάθε άφιξη/αναχώρηση πρέπει να σχετίζεται με ένα και μόνο ένα πλοίο.

Συνεχίζοντας, ένα πλοίο μπορεί να μην βρίσκεται σε κανέναν χώρο ελλιμενισμού (αν είναι εν πλω) ή σε μόνο έναν αν βρίσκεται στο λιμάνι. Επίσης, κάποιος χώρος ελλιμενισμού ενδέχεται να φιλοξενεί πολλά ή και κανένα πλοίο. Εξ ου και η σχέση ένα μη υποχρεωτική προς πολλά μη υποχρεωτική, μεταξύ πλοίου και χώρου ελλιμενισμού.

Η σχέση άφιξης/αναχώρησης με τον χώρο ελλιμενισμού είναι πολλά μη υποχρεωτική προς ένα με υποχρεωτική συμμετοχή. Αυτό γιατί ένας χώρος ελλιμενισμού μπορεί να φιλοξενεί πολλές αφίξεις/αναχωρήσεις ή και καμία, όμως κάθε άφιξη/αναχώρηση πρέπει να σχετίζεται με ένα και μόνο ένα χώρο ελλιμενισμού.

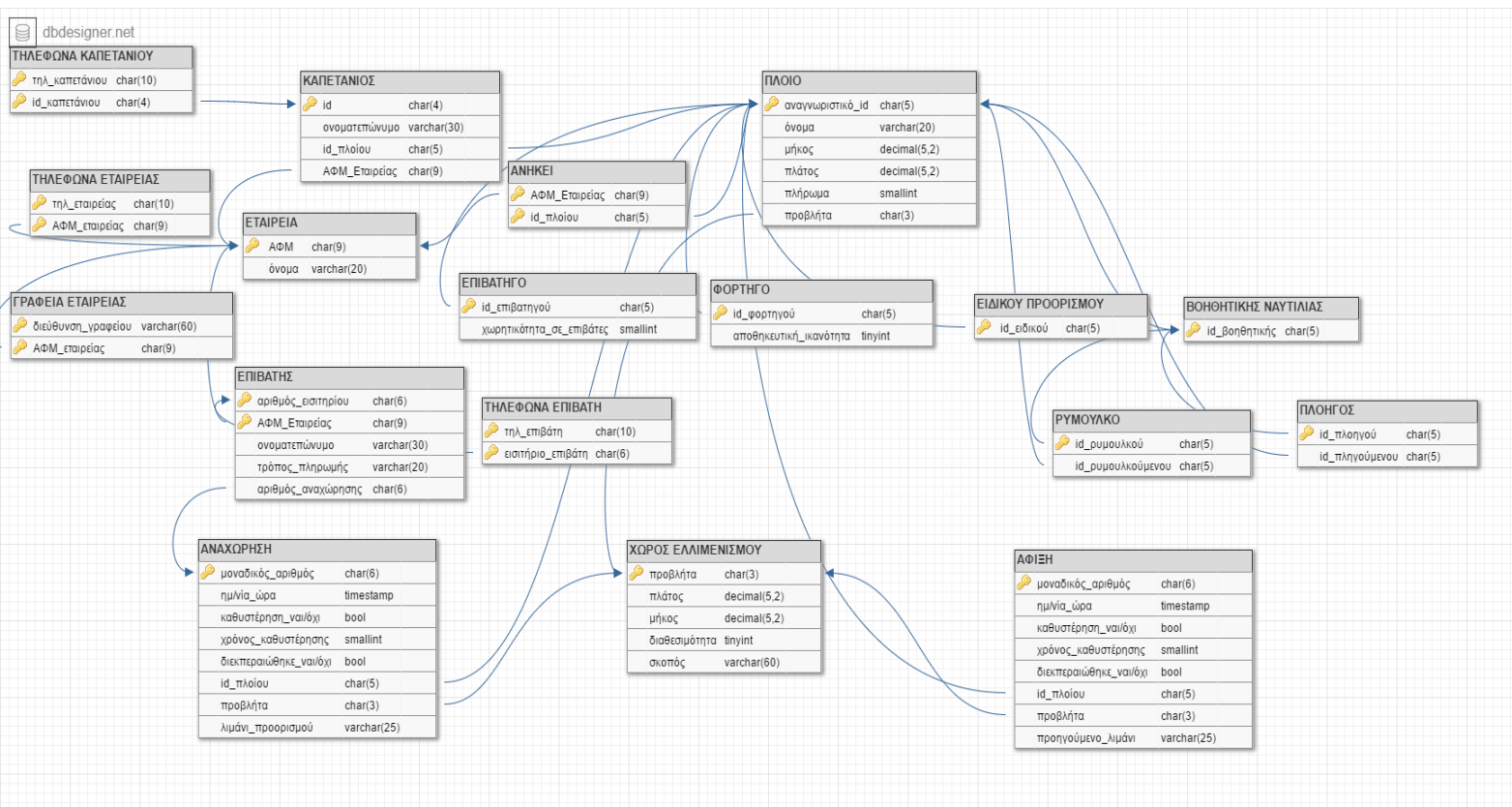
Ένα ρυμουλκό μπορεί να ρυμουλκεί ένα ή κανένα πλοίο και αντίστοιχα ένα πλοίο μπορεί να ρυμουλκείται από ένα και μόνο ένα ρυμουλκό. Γι' αυτό και η επιλογή της σχέσης ένα προς ένα με μη υποχρεωτικές συμμετοχές. Εδώ πρέπει να εξηγήσουμε τι είναι ο πλοηγός. Σε

πολλά λιμάνια υπάρχουν ειδικοί καπετάνιοι που πηγαίνουν με ένα μικρό πλοιάκι στα μεγάλα πλοία που έρχονται στο λιμάνι. Ο καπετάνιος αυτός ανεβαίνει στο μεγάλο πλοίο και αναλαμβάνει να το «παρκάρει», καθώς δεν οφείλει ο καπετάνιος του κάθε πλοίου να γνωρίζει όλα τα λιμάνια. Εμείς λοιπόν εδώ θεωρήσαμε τον πλοηγό σαν μία σχέση εξειδίκευσης στα πλοία που υπάρχουν στο λιμάνι μας. Είναι το μικρό πλοιάκι το οποίο πηγαίνει τον καπετάνιο στο κάθε πλοίο που πρέπει να «παρκάρει». Επίσης θεωρήσαμε πως το λιμάνι μας απαιτεί από τα πλοία που έρχονται να χρησιμοποιήσουν τον πλοηγό, για αποφυγή ατυχημάτων. Έτσι η σχέση πλοηγού και πλοίου είναι ένα με υποχρεωτική προς ένα με μη υποχρεωτική συμμετοχή. Αυτό γιατί κάθε πλοίο πρέπει να πλοηγείται από ένα και μόνο ένα πλοηγό, αλλά κάθε πλοηγός μπορεί να πλοηγεί μέχρι ένα πλοίο, όμως μπορεί και κανένα αν δεν υπάρχει εκείνη τη στιγμή πλοίο που να χρειάζεται πλοήγηση.

Για την εταιρεία θεωρήσαμε πως πρέπει να διαθέτει ένα ή περισσότερα πλοία για να έχει νόημα να υπάρχει στη βάση μας. Επίσης, κάθε πλοίο πρέπει να ανήκει σε κάποια εταιρεία, αλλά μπορεί να ανήκει και σε περισσότερες σε περίπτωση συνιδιοκτησίας. Ακόμα, η εταιρεία πρέπει να διαθέτει έναν ή περισσότερους καπετάνιους, αλλά κάθε καπετάνιος μπορεί να ανήκει σε μία και μόνο μία εταιρεία. Τέλος, η εταιρεία μπορεί να πουλάει πολλά εισιτήρια ή και κανένα (αν τις έχουν τελειώσει ή αν δεν είναι εταιρεία που διαθέτει επιβατηγά). Αυτή είναι και η προσδιορίζουσα σχέση της με την ασθενή οντότητα επιβάτης, ο οποίος κάναμε την παραδοχή ότι μπορεί να αγοράσει ένα και μόνο ένα εισιτήριο για τον εαυτό του(απαιτείται για παράδειγμα κάποια διαδικασία ταυτοποίησης του κάθε επιβάτη).

Σχεσιακό Μοντέλο

Το επόμενο βήμα στο σχεδιασμό της βάσης μας ήταν η μετατροπή του eerd σε σχεσιακό μοντέλο. Η διαδικασία που ακολουθήσαμε περιγράφεται παρακάτω.



Χρησιμοποιήσαμε κι εδώ ένα online εργαλείο, το dbdesigner.net. Ξεκινήσαμε δημιουργώντας ένα πίνακα για κάθε ισχυρή μας οντότητα με όλα τα απλά της γνωρίσματα. Έπειτα, όπου υπήρχε σύνθετο γνώρισμα το σπάσαμε κρατώντας μόνο τα απλά του γνωρίσματα. Τέλος, ορίσαμε το primary key της κάθε ισχυρής οντότητας.

Συνεχίσαμε με την μη ισχυρή μας οντότητα ΕΠΙΒΑΤΗΣ, για την οποία φτιάξαμε επίσης έναν πίνακα με τα γνωρίσματά της. Έπειτα, προσθέσαμε το primary key της ΕΤΑΙΡΕΙΑΣ, το ΑΦΜ, σαν foreign key στον πίνακα ΕΠΙΒΑΤΗΣ (στα foreign keys αλλάξαμε λίγο τα ονόματά τους ώστε να φαίνεται εύκολα η σημασία τους στον καινούριο πίνακα). Τέλος, το πρωτεύων κλειδί μας εδώ είναι ο συνδυασμός του foreign key και του προσδιορίζοντος γνωρίσματός μου, είναι δηλαδή ένα σύνθετο πρωτεύων κλειδί.

Μετά απ' αυτό, υλοποιήσαμε τις σχέσεις τύπου είναι του πλοίου. Επιλέξαμε να κάνουμε ένα πίνακα για την υπερκλάση κι έναν διαφορετικό πίνακα για την κάθε υποκλάση μας. Στην υπερκλάση απλά προσθέσαμε τα γνωρίσματά της. Στις υποκλάσεις προσθέσαμε τα γνωρίσματά τους, αλλά και το πρωτεύων της υπερκλάσης σαν ξένο κλειδί.

Πάμε στις συσχετίσεις. Στην 1-1 το μόνο που κάνουμε είναι πως πηγαίνουμε στον πίνακα της μίας οντότητας που υπάρχει στη σχέση (εμείς επιλέξαμε τον καπετάνιο) και της προσθέσαμε σαν ξένο κλειδί το πρωτεύων της άλλης οντότητας που παίρνει μέρος στη σχέση (εδώ το πλοίο).

Για τις συσχετίσεις 1-N βρίσουμε τη μεριά της σχέσης που αφορά το N και της προσθέτουμε σαν ξένο κλειδί το πρωτεύων της άλλης οντότητας της σχέσης. Προσθέτουμε επίσης τα πιθανά γνωρίσματα που χαρακτηρίζουν τη σχέση (εδώ δεν είχαμε κάπου τέτοια γνωρίσματα).

Τέλος, στις συσχετίσεις M-N δημιουργούμε έναν καινούριο πίνακα για τη σχέση (εδώ ο πίνακας ανηκει). Παίρνουμε στη συνέχεια τα πρωτεύοντα κλειδιά των σχετιζόμενων οντοτήτων και τα προσθέτουμε σαν ξένα κλειδιά στον πίνακα της σχέσης. Το πρωτεύων κλειδί της σχέσης είναι σύνθετο και απαρτίζεται απ' το συνδυασμό των 2 ξένων κλειδιών.

Κλείνοντας, φτιάξαμε καινούριο πίνακα για τα πλειότιμα γνωρίσματά μας, προσθέτοντας σαν ξένο κλειδί το πρωτεύων της αντίστοιχης οντότητας που το πλειότιμο χαρακτηρίζει. Κι εδώ έχουμε σύνθετο πρωτεύων κλειδί με το συνδυασμό του γνωρίσματος και του ξένου κλειδιού.

Κατά τη διάρκεια της διαδικασίας επιλέγαμε και τους τύπους που θα χρησιμοποιούσαμε στο κάθε γνώρισμα. Για γνωρίσματα των οποίων γνωρίζαμε το μήκος, όπως τα ids, τα ΑΦΜ και τα τηλέφωνα, επιλέξαμε char συγκεκριμένου μήκους. Για γνωρίσματα με άγνωστο μήκος, όπως κείμενα περιγραφής, ονόματα λιμανιών, ονόματα ατόμων/πλοίων και διευθύνσεις επιλέξαμε varchar με ένα εύλογο μέγιστο μήκος. Για ημερομηνία και ώρα διαλέξαμε timestamp, αν και εικάζουμε πως εξίσου καλά θα δούλευε και ο τύπος datetime. Έγινε χρήση ακόμα boolean τύπων για γνωρίσματα τύπου ναι/όχι (όπως το αν έχει διεκπεραιωθεί ή αργήσει μια αναχώρηση/άφιξη). Για μήκος και πλάτος χρησιμοποιήσαμε decimal(5,2), δηλαδή 3 ψηφίων αριστερά και 2 δεξιά της υποδιαστολής, ώστε να έχουμε την ακρίβεια που θέλαμε. Τέλος, για τα διάφορα αριθμητικά δεδομένα χρησιμοποιήθηκαν παραλλαγές του τύπου int (smallint, tinyint κλπά). Διαλέξαμε το μικρότερο δυνατό τύπο για το κάθε γνώρισμα, ώστε να μην δεσμεύουμε περιττό χώρο στη βάση μας.

Δημιουργία Βάσης

Χρησιμοποιήσαμε το εργαλείο dbdesigner στο οποίο σχεδιάσαμε το σχεσιακό μας, ώστε να πάρουμε ένα αρχικό κώδικα sql που να δημιουργεί τους πίνακες της βάσης μας. Ένα πρόβλημα που αντιμετωπίσαμε, ήταν με τα ονόματα των πινάκων, τα οποία αντί για κεφαλαία δημιουργήθηκαν με πεζά. Κάναμε τις απαραίτητες αλλαγές στα create table μας ώστε να φύγουν όλα τα errors και να δημιουργηθεί η βάση μας. Η δημιουργία έγινε στην rhrtmyadmin αφού πρώτα είχαμε ενεργοποιήσει apache και mysql στο pc μας μέσω του XAMPP. Επίσης να σημειωθεί πως όταν δημιουργήσαμε τη βάση μας στην rhrtmyadmin επιλέξαμε collation utf8 general ci, για να μην έχουν πρόβλημα με τους χαρακτήρες του ελληνικού αλφαβήτου. Οι εντολές φαίνονται παρακάτω.

```
CREATE TABLE `ΑΦΙΞΗ` (  
    `μοναδικός_αριθμός` char(6) NOT NULL,  
    `ημ/νία_ώρα` TIMESTAMP NOT NULL,  
    `καθυστέρηση_ναι/όχι` boolean NOT NULL,  
    `χρόνος_καθυστέρησης` smallint,  
    `διεκπεραιώθηκε_ναι/όχι` boolean NOT NULL,  
    `id_πλοίου` char(5) NOT NULL,  
    `προβλήτα` char(3) NOT NULL,  
    `προηγούμενο_λιμάνι` varchar(25) NOT NULL,  
    PRIMARY KEY (`μοναδικός_αριθμός`)  
);
```

```
CREATE TABLE `ΧΩΡΟΣ ΕΛΛΙΜΕΝΙΣΜΟΥ` (  
    `προβλήτα` char(3) NOT NULL,  
    `πλάτος` DECIMAL(5,2) NOT NULL,  
    `μήκος` DECIMAL(5,2) NOT NULL,  
    `διαθεσιμότητα` tinyint NOT NULL,  
    `σκοπός` varchar(60) NOT NULL,  
    PRIMARY KEY (`προβλήτα`)  
);
```

```
CREATE TABLE `ΑΝΑΧΩΡΗΣΗ` (  
    `μοναδικός_αριθμός` char(6) NOT NULL,  
    `ημ/νία_ώρα` TIMESTAMP NOT NULL,
```

```
`καθυστέρηση_ναι/όχι` boolean NOT NULL,  
`χρόνος_καθυστέρησης` smallint,  
`διεκπεραιώθηκε_ναι/όχι` boolean NOT NULL,  
`id_πλοίου` char(5) NOT NULL,  
`προβλήτα` char(3) NOT NULL,  
`λιμάνι_προορισμού` varchar(25) NOT NULL,  
PRIMARY KEY (`μοναδικός_αριθμός`)  
);
```

```
CREATE TABLE `ΚΑΠΕΤΑΝΙΟΣ` (  
    `id` char(4) NOT NULL,  
    `ονοματεπώνυμο` varchar(30) NOT NULL,  
    `id_πλοίου` char(5) NOT NULL,  
    `ΑΦΜ_Εταιρείας` char(9) NOT NULL,  
    PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE `ΕΤΑΙΡΕΙΑ` (  
    `ΑΦΜ` char(9) NOT NULL,  
    `όνομα` varchar(20) NOT NULL,  
    PRIMARY KEY (`ΑΦΜ`)  
);
```

```
CREATE TABLE `ΕΠΙΒΑΤΗΣ` (  
    `αριθμός_εισιτηρίου` char(6) NOT NULL,  
    `ΑΦΜ_Εταιρείας` char(9) NOT NULL,  
    `ονοματεπώνυμο` varchar(30) NOT NULL,  
    `τρόπος_πληρωμής` varchar(20) NOT NULL,  
    `αριθμός_αναχώρησης` char(6) NOT NULL,  
    PRIMARY KEY (`αριθμός_εισιτηρίου`,`ΑΦΜ_Εταιρείας`)  
);
```

```
CREATE TABLE `ΠΛΟΙΟ` (  
    `αναγνωριστικό_id` char(5) NOT NULL,  
    `όνομα` varchar(20) NOT NULL,  
    `μήκος` DECIMAL(5,2) NOT NULL ,  
    `πλάτος` DECIMAL(5,2) NOT NULL ,  
    `πλήρωμα` smallint NOT NULL ,  
    `προβλήτα` char(3),  
    PRIMARY KEY (`αναγνωριστικό_id`)  
);
```

```
CREATE TABLE `ΦΟΡΤΗΓΟ` (  
    `id_φορτηγού` char(5) NOT NULL,  
    `αποθηκευτική_ικανότητα` tinyint NOT NULL,  
    PRIMARY KEY (`id_φορτηγού`)  
);
```

```
CREATE TABLE `ΕΠΙΒΑΤΗΓΟ` (  
    `id_επιβατηγού` char(5) NOT NULL,  
    `χωρητικότητα_σε_επιβάτες` smallint NOT NULL,  
    PRIMARY KEY (`id_επιβατηγού`)  
);
```

```
CREATE TABLE `ΕΙΔΙΚΟΥ ΠΡΟΟΡΙΣΜΟΥ` (  
    `id_ειδικού` char(5) NOT NULL,  
    PRIMARY KEY (`id_ειδικού`)  
);
```

```
CREATE TABLE `ΒΟΗΘΗΤΙΚΗΣ ΝΑΥΤΙΛΙΑΣ` (  
    `id_βοηθητικής` char(5) NOT NULL,  
    PRIMARY KEY (`id_βοηθητικής`)
```


);

```
CREATE TABLE `ΡΥΜΟΥΛΚΟ` (  
    `id_ρυμουλκού` char(5) NOT NULL,  
    `id_ρυμουλκούμενου` char(5) NOT NULL,  
    PRIMARY KEY (`id_ρυμουλκού`)  
);
```

```
CREATE TABLE `ΠΛΟΗΓΟΣ` (  
    `id_πλοηγού` char(5) NOT NULL,  
    `id_πληγούμενου` char(5) NOT NULL,  
    PRIMARY KEY (`id_πλοηγού`)  
);
```

```
CREATE TABLE `ΑΝΗΚΕΙ` (  
    `ΑΦΜ_Εταιρείας` char(9) NOT NULL,  
    `id_πλοίου` char(5) NOT NULL,  
    PRIMARY KEY (`ΑΦΜ_Εταιρείας`,`id_πλοίου`)  
);
```

```
CREATE TABLE `ΤΗΛΕΦΩΝΑ ΚΑΠΕΤΑΝΙΟΥ` (  
    `τηλ_καπετάνιου` char(10) NOT NULL,  
    `id_καπετάνιου` char(4) NOT NULL,  
    PRIMARY KEY (`τηλ_καπετάνιου`,`id_καπετάνιου`)  
);
```

```
CREATE TABLE `ΤΗΛΕΦΩΝΑ ΕΤΑΙΡΕΙΑΣ` (  
    `τηλ_εταιρείας` char(10) NOT NULL,  
    `ΑΦΜ_εταιρείας` char(9) NOT NULL,  
    PRIMARY KEY (`τηλ_εταιρείας`,`ΑΦΜ_εταιρείας`)  
);
```

```
CREATE TABLE `ΓΡΑΦΕΙΑ ΕΤΑΙΡΕΙΑΣ` (  
    `διεύθυνση_γραφείου` varchar(60) NOT NULL,  
    `ΑΦΜ_εταιρείας` char(9) NOT NULL,  
    PRIMARY KEY (`διεύθυνση_γραφείου`,`ΑΦΜ_εταιρείας`)  
);
```

```
CREATE TABLE `ΤΗΛΕΦΩΝΑ ΕΠΙΒΑΤΗ` (  
    `τηλ_επιβάτη` char(10) NOT NULL,  
    `εισιτήριο_επιβάτη` char(6) NOT NULL,  
    PRIMARY KEY (`τηλ_επιβάτη`,`εισιτήριο_επιβάτη`)  
);
```

```
ALTER TABLE `αφιξη` ADD CONSTRAINT `ΑΦΙΞΗ_fk0` FOREIGN KEY (`id_πλοίου`)  
REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `αφιξη` ADD CONSTRAINT `ΑΦΙΞΗ_fk1` FOREIGN KEY (`προβλήτα`)  
REFERENCES `χωρος ελλιμενισμού`(`προβλήτα`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `αναχωρηση` ADD CONSTRAINT `ΑΝΑΧΩΡΗΣΗ_fk0` FOREIGN KEY (`id_πλοίου`)  
REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `αναχωρηση` ADD CONSTRAINT `ΑΝΑΧΩΡΗΣΗ_fk1` FOREIGN KEY (`προβλήτα`)  
REFERENCES `χωρος ελλιμενισμού`(`προβλήτα`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `καπετανιος` ADD CONSTRAINT `ΚΑΠΕΤΑΝΙΟΣ_fk0` FOREIGN KEY (`id_πλοίου`)  
REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `καπετανιος` ADD CONSTRAINT `ΚΑΠΕΤΑΝΙΟΣ_fk1` FOREIGN KEY  
(`ΑΦΜ_Εταιρείας`) REFERENCES `εταιρεια`(`ΑΦΜ`) on DELETE CASCADE on UPDATE  
CASCADE;
```

```
ALTER TABLE `επιβατης` ADD CONSTRAINT `ΕΠΙΒΑΤΗΣ_fk0` FOREIGN KEY (`ΑΦΜ_Εταιρείας`)  
REFERENCES `εταιρεια`(`ΑΦΜ`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `επιβατης` ADD CONSTRAINT `ΕΠΙΒΑΤΗΣ_fk1` FOREIGN KEY  
(`αριθμός_αναχώρησης`) REFERENCES `αναχωρηση`(`μοναδικός_αριθμός`) on DELETE  
CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `πλοιο` ADD CONSTRAINT `ΠΛΟΙΟ_fk0` FOREIGN KEY (`προβλήτα`)  
REFERENCES `χωρος ελλιμενισμού`(`προβλήτα`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `φορτηγο` ADD CONSTRAINT `ΦΟΡΤΗΓΑ_fk0` FOREIGN KEY (`id_φορτηγού`)  
REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `επιβατηγο` ADD CONSTRAINT `ΕΠΙΒΑΤΗΓΑ_fk0` FOREIGN KEY  
(`id_επιβατηγού`) REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on UPDATE  
CASCADE;
```

```
ALTER TABLE `ειδικου προορισμου` ADD CONSTRAINT `ΕΙΔΙΚΟΥ ΠΡΟΟΡΙΣΜΟΥ_fk0` FOREIGN  
KEY (`id_ειδικού`) REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on UPDATE  
CASCADE;
```

```
ALTER TABLE `βοηθητικης ναυτιλιας` ADD CONSTRAINT `ΒΟΗΘΗΤΙΚΗΣ ΝΑΥΤΙΛΙΑΣ_fk0`  
FOREIGN KEY (`id_βοηθητικής`) REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE  
CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `ρυμουλκο` ADD CONSTRAINT `ΡΥΜΟΥΛΚΟ_fk0` FOREIGN KEY  
(`id_ρυμουλκού`) REFERENCES `βοηθητικης ναυτιλιας`(`id_βοηθητικής`) on DELETE  
CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `ρυμουλκο` ADD CONSTRAINT `ΡΥΜΟΥΛΚΟ_fk1` FOREIGN KEY  
(`id_ρυμουλκούμενου`) REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on  
UPDATE CASCADE;
```

```
ALTER TABLE `πλοηγος` ADD CONSTRAINT `ΠΛΟΗΓΟΣ_fk0` FOREIGN KEY (`id_πλοηγού`)  
REFERENCES `βοηθητικης ναυτιλιας`(`id_βοηθητικής`) on DELETE CASCADE on UPDATE  
CASCADE;
```

```
ALTER TABLE `πλοηγος` ADD CONSTRAINT `ΠΛΟΗΓΟΣ_fk1` FOREIGN KEY (`id_πληγούμενου`)  
REFERENCES `πλοιο`(`αναγνωριστικό_id`) on DELETE CASCADE on UPDATE CASCADE;
```

```
ALTER TABLE `ανηκει` ADD CONSTRAINT `ΑΝΗΚΕΙ_fk0` FOREIGN KEY (`ΑΦΜ_Εταιρείας`) REFERENCES `εταιρεια`(`ΑΦΜ`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `ανηκει` ADD CONSTRAINT `ΑΝΗΚΕΙ_fk1` FOREIGN KEY (`id_πλοίου`) REFERENCES `πλοιο`(`αναγνωριστικό_id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `τηλεφωνα καπετανιου` ADD CONSTRAINT `ΤΗΛΕΦΩΝΑ ΚΑΠΕΤΑΝΙΟΥ_fk0` FOREIGN KEY (`id_καπετάνιου`) REFERENCES `καπετανιος`(`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `τηλεφωνα εταιρειας` ADD CONSTRAINT `ΤΗΛΕΦΩΝΑ ΕΤΑΙΡΕΙΑΣ_fk0` FOREIGN KEY (`ΑΦΜ_εταιρείας`) REFERENCES `εταιρεια`(`ΑΦΜ`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `γραφεια εταιρειας` ADD CONSTRAINT `ΓΡΑΦΕΙΑ ΕΤΑΙΡΕΙΑΣ_fk0` FOREIGN KEY (`ΑΦΜ_εταιρείας`) REFERENCES `εταιρεια`(`ΑΦΜ`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `τηλεφωνα επιβατη` ADD CONSTRAINT `ΤΗΛΕΦΩΝΑ ΕΠΙΒΑΤΗ_fk0` FOREIGN KEY (`εισιτήριο_επιβάτη`) REFERENCES `επιβατης`(`αριθμός_εισιτηρίου`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Να σημειώσουμε πως όλα μας τα γνωρίσματα ορίστηκαν σαν NOT NULL, εκτός από την προβλήτα στο πλοίο, η οποία μπορεί να είναι κενή αν το πλοίο είναι εν πλω και τον χρόνο καθυστέρησης στις αφίξεις/αναχωρήσεις, αφού μπορεί ένα πλοίο να μην έχει καθυστερήσει. Επίσης, ορίσαμε τα πρωτεύοντα κλειδιά όπως φαίνεται στα create tables και τα ξένα κλειδιά μετά την δημιουργία των πινάκων με τη χρήση constraints και alter tables.

Γέμισμα βάσης

Γεμίσαμε τη βάση μας με κάποια fake δεδομένα, κάνοντας χρήση πολλαπλών insert. Στη διαδικασία αυτή μας βοήθησε ιδιαίτερα το website generatedata.com. Μας βοήθησε στη δημιουργία των δεδομένων αλλά και στη δημιουργία των εντολών, τις οποίες βέβαια χρειάστηκε να «πειράξουμε» και μόνοι μας, ώστε να δουλέψουν με τον επιθυμητό τρόπο. Να αναφερθεί ότι χρειάστηκε ιδιαίτερη προσοχή στη σειρά με την οποία τρέξαμε τα insert αλλά και στα δεδομένα που εισάγαμε, ώστε να ικανοποιούνται περιορισμοί αναφορικής ακεραιότητας. Κύριο θέμα ήταν η ύπαρξη των δεδομένων που αναφέρονται σε ξένα κλειδιά αλλά και η μοναδικότητα των πρωτευόντων κλειδιών. Οι εντολές που χρησιμοποιήσαμε τελικά φαίνονται παρακάτω.

```
INSERT INTO `χωρος ελλιμενισμού` (`προβλήτα`,`πλάτος`,`μήκος`,`διαθεσιμότητα`,`σκοπός`) VALUES
```

```
("B24","431.93","033.93",80,"amet"),("G96","272.88","054.48",172,"vulputate"),("K41","137.70","027.30",153,"Nunc mauris. Morbi non sapien molestie"),("S95","185.14","029.42",125,"Integer in"),("C14","439.42","115.03",77,"enim, gravida sit amet, dapibus id, blandit at,"),("J31","077.08","015.05",19,"molestie arcu. Sed"),("X37","202.53","062.52",125,"Morbi quis urna."),("D77","185.07","034.94",63,"tincidunt,"),("B37","197.56","081.65",100,"eget, ipsum. Donec"),("V32","109.26","018.79",191,"at pretium aliquet,");
```

```
INSERT INTO `πλοιο` (`αναγνωριστικό_id`,`όνομα`,`μήκος`,`πλάτος`,`πλήρωμα`,`προβλήτα`) VALUES ("29763","Risus","417.84","110.26",1405,"C14"),("05445","Tellus","220.10","39.59",1726, NULL),("34881","Et Ultrices","120.52","020.50",507,"K41"),("87616","Arcu Sed","183.78","029.38",1728,NULL),("50340","Ut","381.08","106.95",42,"C14"),("06501","Litora","061.19","014.66",1266,"J31"),("35914","Donec","200.39","052.28",107,NULL),("92488","Uti","148.80","026.69",990,"D77"),("78652","Dolor Quisque Tincidunt","070.22","017.56",1101,NULL),("69292","Suspendisse Tristique Neque","090.02","015.36",209,"V32");
```

```
INSERT INTO `αφιξη` (`μοναδικός_αριθμός`,`ημ/νία_ώρα`,`καθυστέρηση_ναι/όχι`,`χρόνος_καθυστέρησης`,`διεκπεραιώθηκε_ναι/όχι`,`id_πλοίου`,`προβλήτα`,`προηγούμενο_λιμάνι`) VALUES ("416877","2016-06-01 15:46:13",TRUE,30,TRUE,"29763","C14","dapibus"),("526070","2016-09-01 04:06:54",FALSE,0,FALSE,"05445","B24","dui, nec"),("887619","2016-06-18 17:19:46",TRUE,73,TRUE,"34881","K41","magna nec"),("437847","2016-06-22 14:51:27",TRUE,3616,FALSE,"87616","X37","Duis at"),("320136","2016-06-11 08:05:49",TRUE,3802,TRUE,"50340","C14","erat semper rutrum."),("377652","2016-09-10 17:43:12",FALSE,0,TRUE,"06501","J31","dictum eleifend"),("382909","2016-06-17 00:52:52",TRUE,2490,FALSE,"35914","C14","nascetur"),("112134","2016-09-10 12:57:45",TRUE,9937,TRUE,"69292","V32","placerat velit."),("792633","2016-06-10 12:31:04",TRUE,10404,FALSE,"78652","V32","mauris. Suspendisse aliquet"),("777862","2016-06-08 09:15:56",TRUE,29,TRUE,"92488","D77","natoque");
```

```
INSERT INTO `φορτηγο` (`id_φορτηγού`,`αποθηκευτική_ικανότητα`) VALUES ("29763",60),("05445",72);
```

```
INSERT INTO `επιβατηγο` (`id_επιβατηγού`,`χωρητικότητα_σε_επιβάτες`) VALUES ("50340",1139),("87616",2423),("34881",1970);
```

```
INSERT INTO `ειδικου_προορισμου` (`id_ειδικού`) VALUES ("06501");
```

```
INSERT INTO `βοηθητικής ναυτιλίας` (`id_βοηθητικής`) VALUES  
("35914"),("92488"),("78652"),("69292");
```

```
INSERT INTO `ρυμουλκο` (`id_ρυμουλκού`,`id_ρυμουλκούμενου`) VALUES  
("35914","06501"),("92488","29763");
```

```
INSERT INTO `πλοηγος` (`id_πλοηγού`,`id_πλοηγούμενου`) VALUES  
("78652","05445"),("69292","87616");
```

```
INSERT INTO `αναχωρηση`  
(`μοναδικός_αριθμός`,`ημ/νία_ώρα`,`καθυστέρηση_ναι/όχι`,`χρόνος_καθυστέρησης`,`διεκπ  
εραιώθηκε_ναι/όχι`,`id_πλοίου`,`προβλήτα`,`λιμάνι_προορισμού`) VALUES  
("316877","2016-06-07  
15:46:13",TRUE,30,TRUE,"29763","C14","dapibus"),("426070","2016-09-02  
04:06:54",FALSE,0,FALSE,"05445","B24","dui, nec"),("787619","2016-06-19  
17:19:46",TRUE,73,TRUE,"34881","K41","magna nec"),("337847","2016-06-24  
14:51:27",TRUE,3616,FALSE,"87616","X37","Duis at"),("320137","2016-06-12  
08:05:49",TRUE,3802,TRUE,"50340","C14","erat semper rutrum."),("377653","2016-09-13  
17:43:12",FALSE,0,TRUE,"06501","J31","dictum eleifend"),("382908","2016-06-17  
00:52:52",TRUE,2490,FALSE,"35914","C14","nascetur"),("112133","2016-09-10  
12:57:45",TRUE,9937,TRUE,"69292","V32","placerat velit."),("792632","2016-06-11  
12:31:04",TRUE,10404,FALSE,"78652","V32","mauris. Suspendisse  
aliquet"),("777863","2016-06-09 09:15:56",TRUE,29,TRUE,"92488","D77","natoque");
```

```
INSERT INTO `εταιρεία` (`ΑΦΜ`,`όνομα`) VALUES  
("533608625","Iacus."),("957673218","Class"),("400272687","tincidunt.  
Donec"),("226188975","facilis"),("745449502","enim"),("943148339","nibh."),("027303693  
","Iectus"),("877731921","ante"),("194189787","nec ante."),("403677111","tellus.");
```

```
INSERT INTO `επιβατης`  
(`αριθμός_εισιτηρίου`,`ΑΦΜ_Εταιρείας`,`ονοματεπώνυμο`,`τρόπος_πληρωμής`,`αριθμός_α  
ναχώρησης`) VALUES ("866711","533608625","velit. Quisque varius.", "Vivamus non lorem  
vitae odio", "316877"),("939136","957673218","Donec tempus,","pharetra sed, hendrerit a,  
arcu.", "316877"),("378612","877731921","felis orci, adipiscing non,","accumsan sed, facilis  
vitae,","426070"),("891650","194189787","tempor diam dictum  
sapien.", "Fusce", "787619"),("523293","194189787","eget laoreet posuere, enim", "euismod  
in, dolor. Fusce", "320137"),("532322","745449502","amet luctus vulputate,  
nisi", "vel", "377653"),("090312","745449502","id enim. Curabitur massa.", "erat semper  
rutrum.", "382908"),("922935","745449502","Cras vehicula aliquet  
libero.", "quis", "112133"),("357357","745449502","id ante", "massa lobortis ultrices.  
Vivamus", "112133"),("664308","745449502","id, mollis nec,","id nunc", "792632");
```

```
INSERT INTO `ανηκει` (`ΑΦΜ_Εταιρείας`,`id_πλοίου`) VALUES
("533608625","29763"),("957673218","05445"),("400272687","50340"),("226188975","876
16"),("745449502","34881"),("943148339","06501"),("027303693","35914"),("877731921","
92488"),("194189787","78652"),("403677111","69292");
```

```
INSERT INTO `καπετανιος` (`id`,`ονοματεπωνυμο`,`id_πλοίου`,`ΑΦΜ_Εταιρείας`) VALUES
("5283","Marny Guerra","29763","533608625"),("9611","Grady
Hines","05445","957673218"),("1255","Serina
Holland","50340","400272687"),("7117","Josiah
Adams","87616","226188975"),("5166","Scarlet
Gilmore","34881","745449502"),("5769","Skyler
Rogers","06501","943148339"),("6349","Allen
Guerra","35914","027303693"),("0634","Porter
Blackwell","92488","877731921"),("8086","Sierra
Luna","78652","194189787"),("8626","Lavinia Becker","69292","403677111");
```

```
INSERT INTO `τηλεφωνα καπετανιου` (`τηλ_καπετάνιου`,`id_καπετάνιου`) VALUES
("6912593226","5283"),("2610593226","5283"),("6984312121","9611"),("6972377462","12
55"),("2683377462","1255"),("6968479692","7117"),("6991658828","5166"),("6987299468"
,"5769"),("6978573193","6349"),("6983771725","0634"),("6983895875","8086"),("6969392
527","8626");
```

```
INSERT INTO `τηλεφωνα επιβατη` (`τηλ_επιβάτη`,`εισιτήριο_επιβάτη`) VALUES
("6932593226","866711"),("2610583226","866711"),("6994312121","939136"),("697237746
3","378612"),("2684377462","378612"),("6968479692","891650"),("6991658828","523293"
),("6987299468","532322"),("6978573193","090312"),("6983771725","922935"),("6983895
875","357357"),("6969392527","664308");
```

```
INSERT INTO `τηλεφωνα εταιρειας` (`τηλ_εταιρείας`,`ΑΦΜ_Εταιρείας`) VALUES
("2610222345","533608625"),("2610222346","533608625"),("2613098768","957673218"),("
2310521010","400272687"),("2430234567","226188975"),("2314567890","745449502"),("2
109900900","943148339"),("2345678987","027303693"),("2314232232","877731921"),("22
13456765","194189787"),("2345678888","403677111");
```

```
INSERT INTO `γραφεια εταιρειας` (`διεύθυνση_γραφείου`,`ΑΦΜ_Εταιρείας`) VALUES
("Πατρέως 12, Πάτρα, Ελλάδα","533608625"),("Γεωργίου 10, Αθήνα","533608625"),("Rivers
Port, Miami","957673218"),("Γούναρη 42, Πάτρα","400272687"),("San Jose Street, Madrid,
Spain","226188975"),("Vallejo Street, Spain, Barcelona","745449502"),("Wunder Street,
Berlin, Germany","943148339"),("Λιμάνι
Θεσσαλονίκης","027303693"),("Πειραιάς","877731921"),("Ραφήνα","194189787"),("Λιμάνι
Κέρκυρας","403677111");
```

Δημιουργία Εφαρμογής

Χρησιμοποιήσαμε τη γλώσσα python για να συνδεθούμε στη βάση δεδομένων μας μέσω μιας εφαρμογής. Να σημειωθεί ότι το rhmyadmin πρέπει να παραμείνει ανοιχτό για να συνδεθούμε στη βάση μας. Πιο συγκεκριμένα, το scriptaki που χρησιμοποιήσαμε φαίνεται παρακάτω. Κατά κύριο λόγο είναι το scriptaki που μας είχε δωθεί μέσα στο εξάμηνο για να συνδεθούμε σε βάσεις δεδομένων μέσω της python. Άξιο αναφοράς είναι πως για να δουλέψει το scriptaki έπρεπε το socket να είναι κενό, καθώς δουλέψαμε σε windows και όχι σε unix μηχανή.

```
# A minimal Mysql shell for experiments

import pymysql

# σημείωση να αντικατασταθεί με το socket file που ορίζεται στο config
sock = ""

h = 'localhost'

while True:

    try:

        dbname = input("Βάση δεδομένων: ")

        con = pymysql.connect(host = h, unix_socket = sock, \
                               user='root', passwd=None, db=dbname, charset='utf8')

        except pymysql.Error as e:

            print ('Σφάλμα', e)

        else:

            con.isolation_level = None

            cur = con.cursor()

            cur.execute('select version()')

            print('Έκδοση βάσης δεδομένων: {}'.format(cur.fetchone()))

            break

buffer = ""

print ("ΒΔ:", dbname, " Δώστε εντολές SQL (enter για έξοδο)")

while True:

    line = input('>>>')

    if line == "":

        print ("byeeeeeeeeeeeeeee")
```



```

        break

buffer += line

print (buffer)

if True:

    try:

        buffer = buffer.strip()

        if buffer.lstrip().upper().startswith("SELECT"):

            count=0

            cur.execute(buffer)

            desc = [x[0] for x in cur.description]

            print(*desc, sep='\t')

            for row in cur.fetchall():

                for i in row :

                    print (i, end = '\t')

                print()

                count +=1

            if count%30==0:

                reply = input ( "....more ? (y/n)")

                if reply != "y" :

                    break

                else:

                    print ()

            else:

                cur.execute(buffer)

                cur.execute('commit')

                print ("σύνολο :", cur.rowcount )

        except pymysql.Error as e:

            print ("An error occurred:", e)

            buffer = ""

cur.execute('commit')

con.close()

```

Τυπικά Queries προς τη Βάση Δεδομένων μαζί με τα αποτελέσματά τους

Όταν τρέχουμε το scriptaki η διαδικασία είναι ιδιαίτερα απλή. Δίνουμε το όνομα της βάσης μας όταν μας ζητηθεί και είμαστε έτοιμοι να κάνουμε queries προς αυτή. Βλέπουμε κάποια τυπικά queries που τρέξαμε εμείς παρακάτω, τα οποία δούλεψαν κατά τον επιθυμητό τρόπο.

- 1) Τα πλοία που αυτή τη στιγμή είναι αραγμένα σε κάποια προβλήτα, μαζί με την προβλήτα τους.

```
select όνομα, προβλήτα from πλοιο where προβλήτα is not null
```

| όνομα | προβλήτα |
|-------|----------|
|-------|----------|

| | |
|--------|-----|
| Litora | J31 |
|--------|-----|

| | |
|-------|-----|
| Risus | C14 |
|-------|-----|

| | |
|-------------|-----|
| Et Ultrices | K41 |
|-------------|-----|

| | |
|-----|-----|
| Ut, | C14 |
|-----|-----|

| | |
|----------------------|-----|
| Suspendisse Tristiqu | V32 |
|----------------------|-----|

| | |
|------|-----|
| Uti, | D77 |
|------|-----|

σύνολο : 6

- 2) Τα ονόματα των πλοίων που είναι φορτηγά

```
select όνομα from πλοιο where αναγνωριστικό_id in (select id_φορτηγού from φορτηγο)
```

| όνομα |
|-------|
|-------|

| |
|--------|
| Tellus |
|--------|

| |
|-------|
| Risus |
|-------|

σύνολο : 2

- 3) Μια κατάσταση με τα ονόματα των πλοίων και των αντίστοιχων καπετάνιων τους.

```
select καπετανιος.ονοματεπώνυμο as όνομα_καπετάνιου,πλοιο.όνομα as  
όνομα_πλοίου from καπετανιος join πλοιο on  
καπετανιος.id_πλοίου=πλοιο.αναγνωριστικό_id
```

| όνομα_καπετάνιου | όνομα_πλοίου |
|------------------|--------------|
|------------------|--------------|

| | |
|------------------|------|
| Porter Blackwell | Uti, |
|------------------|------|

| | |
|-----------------|-------------------------|
| Serina Holland | Ut, |
| Scarlet Gilmore | Et Ultrices |
| Marny Guerra | Risus |
| Skyler Rogers | Litora |
| Allen Guerra | Donec |
| Josiah Adams | Arcu Sed |
| Sierra Luna | Dolor Quisque Tincidunt |
| Lavinia Becker | Suspendisse Tristique |
| Grady Hines | Tellus |

σύνολο : 10

- 4) Τις προβλήτες στις οποίες αναμένουμε άφιξη επιβατηγού πλοίου και τα ονόματα των αντίστοιχων πλοίων.

```
select πλοιο.όνομα as όνομα_πλοίου, αφιξη.προβλήτα as προβλήτα_άφιξης from
(αφιξη join πλοιο on αφιξη.id_πλοίου=πλοιο.αναγνωριστικό_id) where αφιξη.id_πλοίου
in (select id_επιβατηγού from επιβατηγο)
```

όνομα_πλοίου προβλήτα_άφιξης

Ut, C14

Arcu Sed X37

Et Ultrices K41

σύνολο : 3

- 5) Τα ονόματα των εταιρειών που διαθέτουν ρυμουλκά.

```
select εταιρεια.όνομα as όνομα_εταιρείας from εταιρεια where εταιρεια.ΑΦΜ in (select
ανηκει.ΑΦΜ_Εταιρείας from ανηκει where ανηκει.id_πλοίου in (select
ρυμουλκο.id_ρυμουλκού from ρυμουλκο))
```

όνομα_εταιρείας

lectus

ante

σύνολο : 2

Σημείωση: Το διάγραμμα eerd και το relational model, καθώς επίσης οι εντολές sql για τη δημιουργία και το γέμισμα της βάσης μας υπάρχουν και σε ξεχωριστά αρχεία. Το ίδιο ισχύει και για το Python σκριπτάκι, αλλά και για την παρουσίαση για την τελική εξέταση.