Εφαρμογή λιμανιού στις βάσεις δεδομένων

Ομαδική εργασία βάσεων δεδομένων διδασκαλίας 2021-2022, ομάδα 42

Άγγελος Νικόλαος Β. Αναγνωστόπουλος
University of Patras, up1066593@upnet.gr

Μιχάλης Α. Δροσιάδης
University of Patras, up1066594@upnet.gr

Η παρούσα αποτελεί αναφορά για την ομαδική εργασία των βάσεων δεδομένων (2021-2022). Η εφαρμογή που ανατέθηκε στην ομάδα μας αφορά την δημιουργία ενός λειτουργικού λιμανιού. Αποφασίσαμε το λιμάνι να είναι εμπορικό αντί για επιβατικό, με το κύριο μέρος της εργασίας να ασχολείται στον τρόπο πραγματοποίησης συναλλαγών, καθώς και την ιχνηλάτιση των πλοίων που αφορούν το λιμάνι μας, είτε αυτά βρίσκονται σε κάποια θέση του, είτε είναι προγραμματισμένο να έρθουν με εμπορεύματα σε κάποια μελλοντική ημερομηνία Επίσης υλοποιήθηκαν οι κατάλληλες δομές για περεταίρω χρήσιμες πληροφορίες σχετικά με διαθέσιμες θέσεις, βάρδιες προσωπικού, περιβαλλοντικές πληροφορίες κ.α..

Λέξεις Κλειδιά: Βάσεις δεδομένων, λιμάνι, κοντέινερ

1 ΕΙΣΑΓΩΓΗ

Το πρώτο μέρος της εργασίας αφορά τον ορισμό του εύρους της εργασίας καθώς και σχεδιασμό του μικροκόσμου του προβλήματος μας. Για αυτό το σκοπό ήταν απαραίτητη η έρευνα πανομοιότυπων εργασιών και εφαρμογών και η αντίστροφη μηχάνευση τους. Η μοντελοποίηση απεδείχθη πρόκληση ως προς την βέλτιστη υλοποίηση της, παρόλα αυτά εκτιμούμε τη λύση κάτι παραπάνω από ικανοποιητική.

Πρώτο βήμα ήταν η δημιουργία του διαγράμματος οντοτήτων συσχετίσεων (ERD). Για τον σκοπό αυτό διαλέξαμε να μην χρησιμοποιήσουμε το ERDMaker, διότι αν και λειτουργικό, το βρίσκουμε ελλιπές. Αντ' αυτού, προτιμήσαμε το ERD Plus, για τις λειτουργίες αυτόματης δημιουργίας σχεσιακού μοντέλου και σύνταξης εντολών SQL. Δυστυχώς ανακαλύψαμε ότι η αυτοματοποιημένη δουλειά που το εργαλείο πραγματοποιεί δεν ήταν ικανοποιητική και χρειαζόταν πολλαπλές διορθώσεις, τα διαγράμματα έφτασαν στην τελική τους μορφή.

Έπειτα ασχοληθήκαμε με την σύνταξη των εντολών CRUD στην SQL. Για αυτή τη δουλειά προτιμήσαμε την MySQL, ως την πιο δοκιμασμένη και εύχρηστη γλώσσα SQL. Η σύνταξη των CRUD commands καθώς και των ερωτήσεων (queries) έγινε με χρήση του MySQL Workbench και του MySQL connector. Αυτές οι εντολές έπειτα χρησιμοποιήθηκαν για την δημιουργία του Python application, το οποίο και χρησιμοποιούμε για την διεπαφή με τη βάση δεδομένων μας. Φυσικά δεν περιοριστήκαμε σε απλά CRUD commands όμως. Έπειτα συντάξαμε εντολές για κάποια views, έτσι ώστε να μπορούμε να έχουμε τις πιο χρήσιμες πληροφορίες από τους πίνακες μας συγκεντρωμένες, αλλά και για να μπορούμε να δώσουμε πρόσβαση σε συγκεκριμένα views σε συγκεκριμένους χρήστες για λόγους ασφαλείας. Επίσης δημιουργήσαμε ένα ευρετήριο (index) για τον πίνακα των πλοίων, μιας και ο όγκος δεδομένων που είχαμε ήταν δύσκολο να διατρεχθεί γραμμικά.

Η εφαρμογή αναπτύχθηκε με χρήση του MySQL connector για Python3 και η γραφική διεπαφή της με την βιβλιοθήκη TkInter. Αυτά προτιμήθηκαν για ευκολία στη χρήση και την κατανόηση του προκύπτοντος κώδικα.

1.1 Παραδοχές μικρόκοσμου

- Τα πλοία έρχονται φέρνοντας εμπορεύματα είτε σε container είτε χύμα.
- Τα πλοία που μας απασχολούν καταλαμβάνουν μία θέση με βάση το μήκος τους.
- Τα πλοία μπορούν να φτάσουν ή να αναχωρήσουν από το λιμάνι σε ώρες διαφορετικές από τις καθορισμένες.
- Το λιμάνι στεγάζει αποθήκη για τα εμπορεύματα της οποίας η υλοποίηση αποφεύχθηκε μετά από συνεννόηση με τους διδάσκοντες, λόγω της έκτασης της εργασίας.
- Το κάθε πλοίο μπορεί να έχει διαφορετικά container με τα ίδια ή και διαφορετικά εμπορεύματα το κάθε ένα (ή και χύμα).
- Το λιμάνι έχει εγκαταστάσεις για προσωπικό που εργάζεται σε αυτό και αναλαμβάνει τα πλοία που καταφθάνουν.

1.2 Σημειώσεις επί του Relational Schema

Υλοποιούνται οι πίνακες Arrival, Departure, ώστε να μπορούν να ανακτηθούν δεδομένα για την κίνηση του λιμανιού και για τα πλοία που εξυπηρετούνται (πχ. Ετήσιοι απολογισμοί, υπολογισμός κινητικότητας κ.α.). Αν ακολουθούσε κανείς αυστηρά το ERD Θα υλοποιούσε απλώς δύο Foreign Keys, όμως προτιμήθηκαν οι πίνακες για τις προαναφερθείσες λειτουργίες. Επίσης για να γίνει ξεκάθαρος ο τύπος δεδομένων κάθε πίνακα, αποφασίσαμε το πλοίο να μην κρατάει record για την θέση στην οποία βρίσκεται και να υλοποιήσουμε το ανάποδο, η θέση δηλαδή να γνωρίζει αν έχει η όχι πλοίο αυτή τη στιγμή.

2 ΛΕΠΤΟΜΕΡΙΕΣ ΣΧΕΔΙΑΣΜΟΥ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Ας δούμε τώρα λεπτομερώς τα διαράμματα, τους κώδικες αλλά και τα διάφορα εραλεία που χρησιμοποιήσαμε για την εργασία.

2.1 ERD-RelationalSchema

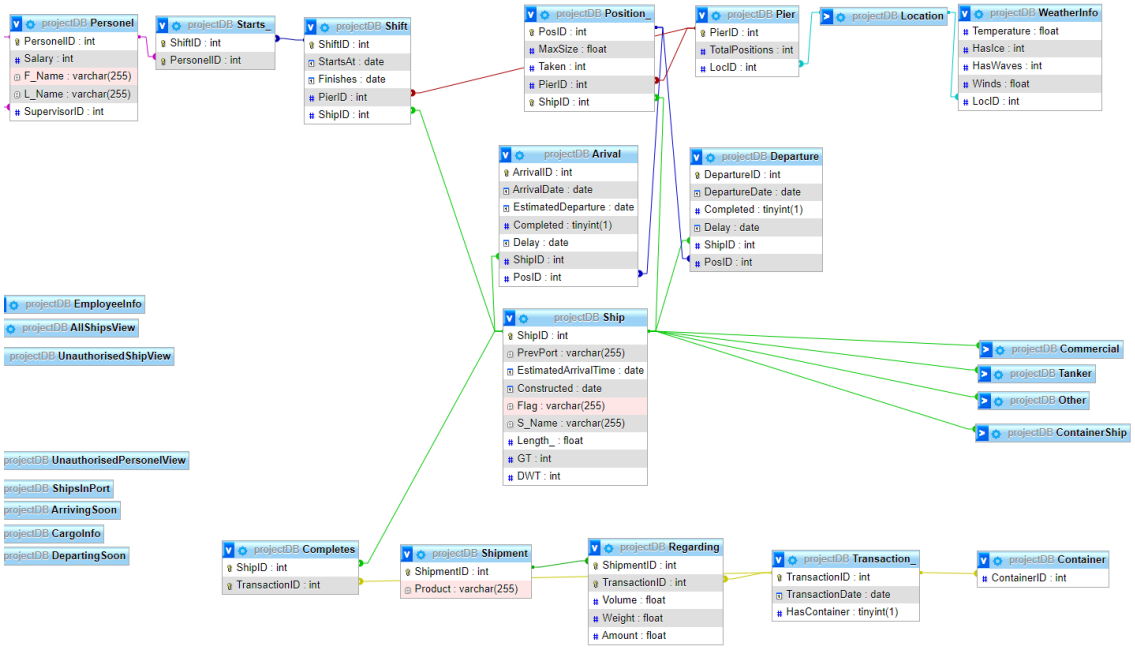

Figure 1: Relational schema as shown in PhpMyAdmin

Δεδομένης της πολυπλοκότητας που έχει μία τέτοια εφαρμογή, επιλέξαμε να εστιάσουμε σε συγκεκριμένους τομείς για την παρούσα εργασία. Όπως και θα περιμέναμε, ο πίνακας του πλοίου έχει τις περισσότερες πληροφορίες αλλά και τις

περισσότερες συνδέσεις με άλλους πίνακες. Επιλέχθηκε η υλοποίηση με disjoint αντί για ένα ακόμη πεδίο για λόγους ελέγχου των τύπων των πλοίων που εξυπηρετεί το λιμάνι μας. Έπειτα, ιδιαίτερη βάση δώθηκε στις δοσοληψίες εμπορευμάτων (κάτω τμήμα εικόνας) αλλά και στην καταγραφή των αφίξεων/αναχωρήσεων. Προσπαθήσαμε να έχουμε κανονικοποιημένες μορφές στους πίνακές μας και οι σχέσεις να φαίνονται ξεκάθαρα μέσω των κλειδιών. Τα ονόματα που δώθηκαν στα πεδία επίσης εξυπηρετούν τον ίδιο σκοπό ευαναγνωσιμότητας και ευκολίας στην πλοήγηση του διαγράμματος.

2.2 MySQL database creation

Όπως αναφέρθηκε και παραπάνω, η πλειοψηφία των εντολών συντάχθηκε στο MySQL Workbench. Η MySQL, ως η πλέον standard επιλογή για σχεσιακές βάσεις δεδομένων, υλοποιεί εύκολα και γρήγορα τις λειτουργίες που θα χρειαζόμασταν. Το πρώτο κομμάτι αφορούσε την μετάφραση του σχεσιακού σχήματος σε κώδικα MySQL. Αυτό έγινε με τη δημιουργία των αρχείων db_creation.sql και updatesDeletes.sql. Έτσι είχαμε τον σκελετό της βάσης στα χέρια μας.

Το επόμενο βήμα αφορούσε το γέμισμα της βάσης με εγγραφές. Αυτό αρχικά έγινε με τη χρήση του insertions.sql, όπου ενδεικτικά υπάρχουν μερικές εγγραφές για να φανούν ξεκάθαρα τα ερωτήματα που θα συντάσαμε. Επιλέξαμε όμως να φτιάξουμε και ένα python script, το οποίο δημιουργεί αυτόματα ένα αρχείο .sql με πλοία (διαλέξαμε πλοία και όχι υπαλλήλους, βάρδιες ή κάτι άλλο διότι εκεί δίνουμε έμφαση στην εργασία, θα μπορούσε με μικρές τροποποιήσεις ο ίδιος κώδικας κάλλιστα να κάνει την ίδια δουλειά για τα παραπάνω), τα οποία χρησιμοποιεί η βάση δεδομένων μας. Αυτό έγινε για να δείξουμε την χρησιμότητα των ευρετηρίων (indexes), με τα οποία μειώσαμε τον χρόνο των αναζητήσεων σημαντικά!

Τέλος, μας ενδιέφεραν τα θέματα ασφάλειας. Δεν είναι δυνατόν ο μέσος χρήστης να έχει πληροφορίες για όλους τους πίνακες που έχει το λιμάνι μας! Οπότε δημιουργήσαμε όψεις (views) και μέσω της εντολής GRANT δίνουμε πρόσβαση μόνο στις πληροφορίες που εμείς θέλουμε ο χρήστης να έχει. Εξαίρεση αποτελεί φυσικά ο root user ο οποίος έχει πρόσβαση στο σύνολο της βάσης δεδομένων και των εγγραφών της.

2.3 Python application and GUI

Βέβαια, η SQL δεν αρκεί για να φτιάξουμε μία εφαρμογή. Ένας μέσος χρήστης δε ξέρει πώς να χειριστέι μία βάση δεδομένων από το command line. Για αυτό το σκοπό δημιουργήσαμε μία γραφική διεπαφή σε Python 3.x η οποία με χρήση του module Tkinter. Φτιάξαμε λοιπόν μία εφαρμογή με το python mysql connector, η οποία εκμεταλλεύεται τα προαναφερθέντα views και τα εμφανίζει στον χρήστη. Υπάρχει επίσης παράθυρο σύνταξης SQL για τους σκοπούς της παρουσίασης (στο οποίο φυσικά πρόσβαση έχει μόνο ο root). Για τις ανάγκες του πρότζεκτ, δημιουργήσαμε χωριστά τα αρχεία που δημιουργούν, γεμίζουν και εμφανίζουν την βάση δεδομένων μας. Το db_init.py αναλαμβάνει τις δύο πρώτες διεργασίες, ενώ το client.py αναλαμβάνει τη σύνδεση σε μία υπάρχουσα ΒΔ και την εμφάνιση των περιεχομένων της σε ένα GUI.

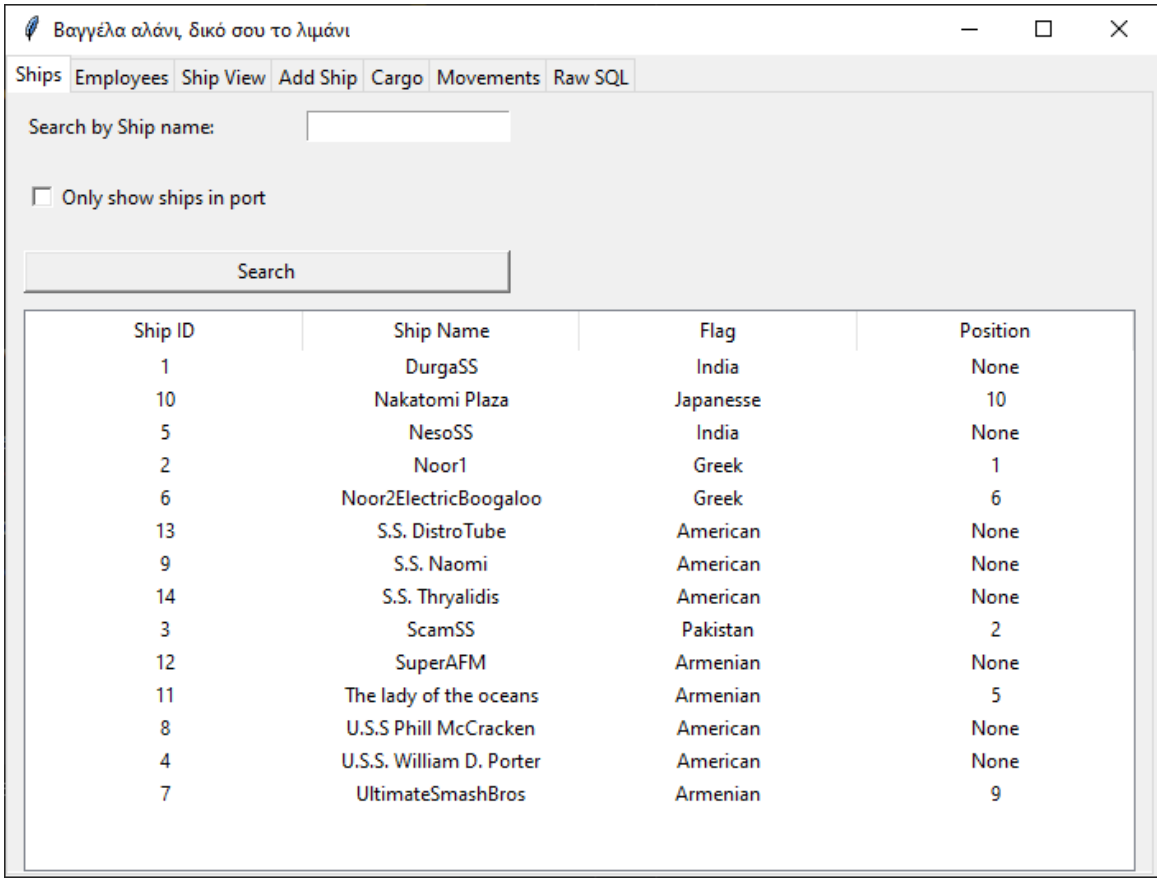

Figure 2: Python GUI Sample

2.4 Αλγόριθμοι

Αλγόριθμος 1: Δημιουργία τυχαίων πλοίων

```
def make_ship():
    with open("../../sqlStuff/randomShips.sql", "a") as all_ships:
        port = str(random.choice(ports))
        con = "DATE '" + random_date(datetime.datetime.strptime("1940-1-1",
"%Y-%m-%d"),datetime.datetime.strptime("2020-1-1", "%Y-%m-%d")) + "'"
        EAT = "DATE '" + random_date(datetime.datetime.strptime("1940-1-1",
"%Y-%m-%d"),datetime.datetime.strptime("2020-1-1", "%Y-%m-%d")) + "'"
        flag = str(random.choice(flags))
        name = str(random.choice(ship_names))
        length = random.randint(0,100)
        gt = random.randint(0,10)
```

```
        dwt = random.randint(0,100)

        ship_data = (name,port,EAT,con,flag,length,gt,dwt)
        sql = (str(insert_str.format(*ship_data)))
        all_ships.write(sql)
        all_ships.write("\n")
```

Αλγόριθμος 2: Χρήση SQL commands με Python

```
try:
        mydb = mysql.connector.connect(**config)
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Wrong username/password combination")
            logger.info("Wrong username/password combination")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            logger.info("Database does not exist")
        else:
            logger.critical(err)
    else:
        cursor = mydb.cursor(buffered=True)

        try:
            cursor.execute("DROP DATABASE projectDB")
        except:
            pass

        os.chdir(os.path.dirname(__file__))
        dbutils.use_database(mydb, cursor)
        print("Creating tables:")
        dbutils.execute_sql_file(mydb, cursor, "../sqlStuff/dbcreation.sql")
        print("Inserting data:")
```

```
dbutils.execute_sql_file(mydb, cursor, "../sqlStuff/insertions.sql")
dbutils.execute_sql_file(mydb, cursor, "../sqlStuff/randomShips.sql")
print("Creating views:")
dbutils.execute_sql_file(mydb, cursor, "../sqlStuff/views.sql")
print("Creating ships index:")
dbutils.execute_sql_file(mydb, cursor, "../sqlStuff/indexes.sql")
mydb.commit()
```

ΕΡΓΑΣΙΑ

Μιχάλης Δροσιάδης: Σχεδιασμός ΒΔ, Δημιουργία GUI, διορθώσεις στην SQL. Dockerisation της εφαρμογής.

Αναγνωστόπουλος Άγγελος: Αρχικά drafts SQL, python mysql connector scripts, db population script, Σχεδιασμός ΒΔ, python packaging, σύνταξη αναφοράς.

REFERENCES

Ramez Elmasri & Shamkant B. Naathe (2021). Fundamentals of Database Systems (7th Edition). PEARSON.

Jesper Wisbor Krogh (2018) MySQL Connector/Python Revealed:SQL and NoSQL Data Storage Using MySQL for Python Programmers (1st Edition) Apress.