

Εργασία στο μάθημα Βάσης Δεδομένων 2024/2025

Δημιουργία και διαχείριση μέσω GUI βάσης δεδομένων εκδοτικού οίκου

Μαρία - Νίκη Ζωγράφου

Προπτυχιακή φοιτήτρια τμήματος ΗΜΤΥ, Πανεπιστημίου Πατρών, up1096060@upnet.gr

Νικόλαος Γέροντας

Προπτυχιακός φοιτητής τμήματος ΗΜΤΥ, Πανεπιστημίου Πατρών, up1092813@upnet.gr

Σε αυτήν την εργασία περιγράφουμε τη σχεδίαση και υλοποίηση μιας εφαρμογής διαχείρισης και ανάλυσης δεδομένων για έναν εκδοτικό οίκο, αξιοποιώντας μια σχεσιακή βάση δεδομένων, στα πλαίσια της διδασκαλίας του μαθήματος «Βάσεις Δεδομένων» για το ακαδημαϊκό έτος 2024-2025.

2.2 Schema:

Στη συνέχεια, βάση του παραπάνω ERD καταλήξαμε στο σχεσιακό μοντέλο που ακολουθεί. Η σχεδίαση του ήταν εφικτή μέσω του web εργαλείου [Schema Builder](#) του τμήματός μας:

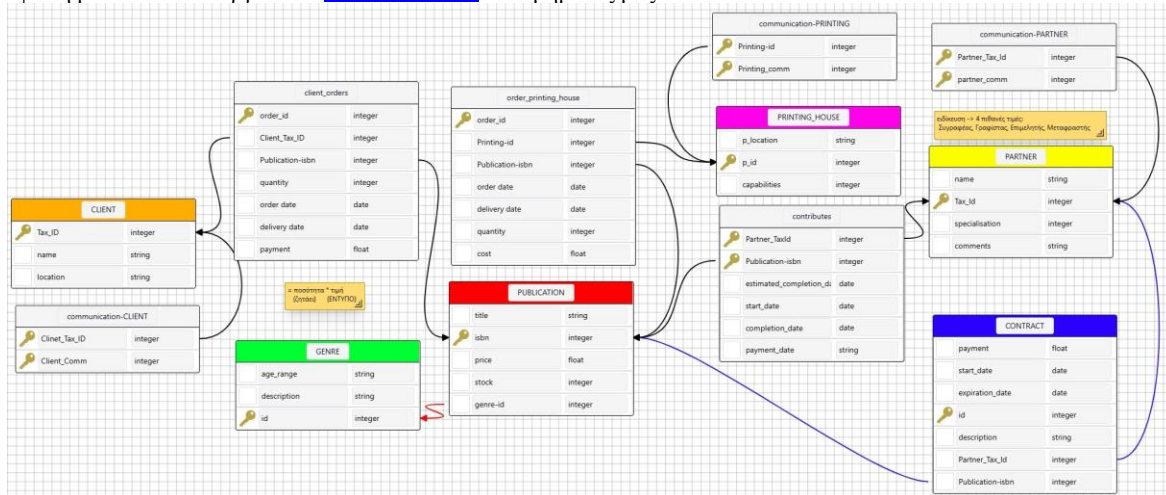


Figure 2: Schema της βάσης δεδομένων μας.

2.3 Δημιουργία Βάσης & Data Generation:

Αποφασίσαμε να χρησιμοποιήσουμε την SQLite για την βάση μας, καθώς αποτελεί βασική βιβλιοθήκη της Python. Έτσι, μετατρέψαμε το παραπάνω Schema σε εντολές δημιουργίας της βάσης δεδομένων μας. Δώσαμε ιδιαίτερη έμφαση στην διασφάλιση της αναφορικής ακεραιότητας της βάσης, καθώς και στην δημιουργία **ρεαλιστικών** δεδομένων για αυτή μέσω δικού μας κώδικα και όχι κάποιας έτοιμης βιβλιοθήκης όπως τη Faker.

2.4 Δημιουργία CRUD Queries (Insert, Delete, Update):

Για την υλοποίηση των βασικών λειτουργιών CRUD, αναπτύχθηκαν αντίστοιχες SQL λειτουργίες στην Python. Η εφαρμογή αυτών πάνω στην βάση δεδομένων συμμορφώνεται στους κανόνες που ορίστηκαν στο προηγούμενο βήμα, με αποτέλεσμα την σωστή διαχείριση της βάσης από τον χρήστη μέσω του GUI.

2.5 Δημιουργία Εφαρμογής με GUI:

Το GUI της κύριας εφαρμογής μας αναπτύχθηκε με τη χρήση της βιβλιοθήκης Tkinter, προσφέροντας έναν φιλικό και εύχρηστο τρόπο διαχείρισης της βάσης δεδομένων. Παράλληλα, δόθηκε ιδιαίτερη έμφαση στη σωστή οργάνωση του κώδικα, με την κατάτμηση των διαφόρων λειτουργιών της γραφικής διεπαφής σε ξεχωριστά αρχεία Python για καλύτερη δομή και ανάπτυξη.

2.6 Δημιουργία SQL Queries Στατιστικών:

Για την εξαγωγή στατιστικών στοιχείων του εκδοτικού οίκου, αναπτύχθηκαν εξειδικευμένα SQL queries που ανταποκρίνονται στις βασικές ανάγκες ανάλυσης δεδομένων, όπως ο υπολογισμός του συνολικού κόστους συνεργασιών και εκτυπώσεων. Τα αποτελέσματα εμφανίζονται τόσο μέσω γραφικών (με χρήση Matplotlib) όσο και στο terminal.

2.7 Scalable Βάση - Μεγαλύτερο Data Generation:

Για την υποστήριξη μεγάλης κλίμακας δεδομένων, το εργαλείο Data Generation ενσωματώθηκε με δυνατότητα ρύθμισης του πλήθους των παραγόμενων δεδομένων μέσω μεταβλητής (scale factor). Επιπλέον, η διαχείριση των παραγόμενων δεδομένων δοκιμάστηκε με χρήση της SQLite.

2.8 Βελτίωση GUI για Statistics:

Το τμήμα της εφαρμογής που είναι υπεύθυνο για την παρουσίαση των στατιστικών στοιχείων, σχεδιάστηκε με τέτοιο τρόπο ώστε η εισαγωγή νέων εξειδικευμένων SQL queries να γίνεται με μεγάλη ευκολία από τους σχεδιαστές της βάσης.

Υπήρχε υστέρια βελτίωση των παραπάνω με επαναληπτική διαδικασία εκτός του ERD και του Schema, δηλαδή νέα queries, βελτίωση γραφικών, βελτίωση Data generation.

3 ΑΞΙΟΛΟΓΗΣΗ

Η αξιολόγηση του project βασίστηκε σε συγκεκριμένα κριτήρια επιτυχίας που διασφαλίζουν την αξιοπιστία, την ευχρηστία και τη λειτουργικότητα της εφαρμογής:

3.1 Ασφάλεια Δεδομένων:

Η διαγραφή συνεργατών, πελατών ή τυπογραφείων των οποίων τα κλειδιά χρησιμοποιούνται ως ξένα κλειδιά σε άλλους πίνακες αποτρέπεται μέσω των περιορισμών της SQL (π.χ., ON DELETE RESTRICT). Οι λειτουργίες INSERT, DELETE και UPDATE πραγματοποιούνται μόνο όταν επιτρέπονται από τη λογική της εφαρμογής, διασφαλίζοντας ότι δεν γίνονται αλλαγές που θα μπορούσαν να οδηγήσουν σε ασυνέπειες. Παραδείγματος χάριν δεν επιτρέπεται η αλλαγή Tax_ID στη βάση, γεγονός που αποτρέπει τα λάθη των χρηστών και εξασφαλίζει την αξιοπιστία της εφαρμογής.

3.2 Ορθή Αποθήκευση και Παρουσίαση των Δεδομένων:

Όταν επιτρέπεται η τροποποίηση δεδομένων, οι αλλαγές ενημερώνονται αμέσως στο GUI και καταγράφονται σωστά στη βάση.

3.3 Λογικό και Γρήγορο Data Generation:

Κατά τη δημιουργία δεδομένων (Data generation), οι ημερομηνίες ακολουθούν λογική σειρά: ένα βιβλίο δεν μπορεί να τυπωθεί πριν ολοκληρωθεί η συγγραφή του, ούτε να παραδοθεί πριν εκτυπωθεί. Αντίστοιχα δημιουργούνται στοιχεία ISBN, tax id, τίτλοι, ονόματα και αριθμοί ποσοτήτων που βρίσκονται μέσα σε λογικά πλαίσια, δημιουργώντας έτσι μια φαινομενικά ρεαλιστική βάση δεδομένων.

Επιπλέον χρησιμοποιείται βελτιστοποίηση με dictionaries για την αποθήκευση κρίσιμων ημερομηνιών, μειώνοντας τον υπολογιστικό φόρτο και επιταχύνοντας την δημιουργία δεδομένων.

3.4 Ευελιξία στη Δημιουργία Δεδομένων:

Το Data generation είναι προσαρμοσμένο ώστε να υποστηρίζει διάφορες τιμές scale factor, επιτρέποντας τη δημιουργία δεδομένων διαφορετικής κλίμακας. Η διαδικασία δημιουργίας ολοκληρώνεται εντός λογικών χρονικών ορίων, με ~25,000,000 συνολικές εγγραφές (δηλαδή, scale factor = 100,000) να παράγονται σε λιγότερο από 10 λεπτά.

3.5 Οπτικοποίηση και Γραφικά:

Η εφαρμογή διαθέτει απλά και καλοσχεδιασμένα γραφικά, με μικρά animations και πολύχρωμα charts στατιστικών που κάνουν την ανάλυση δεδομένων πιο εύκολη και ευνόητη.

Παρέχει ευανάγνωστη εκτύπωση δεδομένων στο τερματικό, διευκολύνοντας την παρακολούθηση των ενεργειών του συστήματος.

3.6 Ευχρηστία:

Οι λειτουργίες αναζήτησης (search functions) και στατιστικών (statistics) είναι εύκολα προσβάσιμες και παρέχουν χρήσιμες πληροφορίες με φιλικό τρόπο προς τον χρήστη. Στην υλοποίηση περιλαμβάνεται ο τελεστής LIKE της SQL για πιο ευέλικτα κριτήρια αναζήτησης. Επιπλέον, έχει ενσωματωθεί λειτουργία που βασίζεται στον αλγόριθμο Levenshtein, η οποία σε περιπτώσεις τυπογραφικών λαθών προτείνει στον χρήστη τη σωστή λέξη, εφόσον αυτή απέχει μόνο ένα γράμμα από την αρχική.

Επίσης, η εφαρμογή υλοποιεί dynamic path calculation, διευκολύνοντας την εγκατάσταση και τη χρήση της σε διαφορετικά περιβάλλοντα και υπολογιστές, χωρίς την ανάγκη χειροκίνητων ρυθμίσεων διαδρομών. Αυτό αυξάνει τη φορητότητα και τη φιλικότητα προς το χρήστη.

4 ΔΕΔΟΜΕΝΑ

Προκειμένου να αξιολογήσουμε τη λειτουργικότητα του **Data Generation** και της **GUI εφαρμογής**, δημιουργήσαμε μια γιγαντιαία βάση δεδομένων με **scale factor = 100,000**. Αυτή η βάση μας επέτρεψε να δοκιμάσουμε το σύστημα στα όριά του, εξασφαλίζοντας την αντοχή και την αξιοπιστία του.

Table 1: Πίνακας που δείχνει το τελικό πλήθος των εγγραφών για scale factor = 100,000.

| Οντότητες | PARTNER | CONTRACT | CLIENT | PRINTING_HOUSE | GENRE | PUB. |
|-----------|-----------|-----------|-----------|----------------|-------|------|
| Πλήθος | 1,000,000 | 3,000,000 | 1,500,000 | 500,000 | 20 | 2 M |

| Συσχετίσεις | contributes | | client_orders | order_printing_house | | |
|-------------|-------------|--|---------------|----------------------|--|--|
| Πλήθος | 2,999,238 | | 5,000,000 | 3,834,593 | | |

| Multivalued | communication-PARTNER | | communication-CLIENT | communication-PRINTING | | |
|-------------|-----------------------|--|----------------------|------------------------|--|--|
| Πλήθος | 2,000,032 | | 2,999,079 | 999,892 | | |

```
PS C:\Users\Cobra\Downloads\publishing_house_db-main\ourDB> python generate_db_records.py
SQLite version: 3.45.3
Old database file deleted successfully!

Database created successfully!

Partner generation took 10.27341 seconds.
Client generation took 17.13347 seconds.
Printing house generation took 0.60302 seconds.
Genre generation took 0.00012 seconds.
Publication generation took 27.27438 seconds.
Contract generation took 71.89660 seconds.
Client order generation took 133.09443 seconds.
Printing order generation took 70.89965 seconds.
Contribution generation took 177.41725 seconds.
Communication generation took 17.98097 seconds.

Database populated successfully!
Total time: 550.33647 seconds - Scale factor: 100000.
PS C:\Users\Cobra\Downloads\publishing_house_db-main\ourDB> |
```

Figure 3:
Αναλυτικός
χρόνος εκτέλεσης
της γεννήτριας.

[Σύνδεσμος για την GIGA βάση](#)

5 ΔΙΑΜΟΙΡΑΣΜΟΣ:

Η εργασία αποτελείται από πολλά μέρη, απαιτώντας στενή συνεργασία για τη δημιουργία και τη σύνδεση των επιμέρους τμημάτων της. Για την καλύτερη οργάνωση, χρησιμοποιήθηκε ένα κοινό έγγραφο Google Docs, όπου καταγράφονταν τα tasks, το άτομο στο οποίο ανατέθηκαν, η κατάσταση ολοκλήρωσής τους και η προτεραιότητά τους. Αυτή η μεθοδολογία διευκόλυνε τον συντονισμό και τη διασφάλιση της προόδου του έργου. Αν και χρειάστηκε στενή συνεργασία για την δημιουργία όλων των τμημάτων του project, ο καθένας επικεντρώθηκε περισσότερο σε συγκεκριμένα μέρη του έργου, με τον εξής διαμοιρασμό:

- ERD: από κοινού
- Schema: από κοινού
- Δημιουργία Βάσης - Διασφάλιση Αναφορικής Ακεραιότητας: κ. Ζωγράφου
- CRUD queries (database_manager.py): κ. Γέροντας
- Αξιολόγηση λειτουργιών CRUD: κ. Ζωγράφου
- Data Generation: από κοινού
- SQL queries: από κοινού
- Scalability της Βάσης: κ. Γέροντας
- GUI Εφαρμογής: από κοινού
- GUI Στατιστικών: κ. Γέροντας

6 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Όπως και προηγουμένως, οι προαναφερθέντες στόχοι θα αποτελέσουν τη βάση του χρονοδιαγράμματος που ακολουθεί:

- 2.1 ERD: Τέλη Οκτωβρίου
- 2.2 Schema: Τέλη Οκτωβρίου
- 2.3 Δημιουργία Βάσης & Data Generation: Αρχές Δεκεμβρίου
- 2.4 Δημιουργία CRUD Queries (Insert, Delete, Update): Αρχές Δεκεμβρίου
- 2.5 Δημιουργία Εφαρμογής με GUI: Μέσα Δεκεμβρίου
- 2.6 Δημιουργία SQL Queries Στατιστικών: Μέσα Δεκεμβρίου
- 2.7 Scalable Βάση - Μεγαλύτερο Data Generation: Τέλη Δεκεμβρίου
- 2.8 Βελτίωση GUI για Statistics: Διακοπές Χριστουγέννων
- 2.9 Βελτίωση εφαρμογής, data generation, προσθήκη SQL queries: Διακοπές Χριστουγέννων

7 ΟΔΗΓΙΕΣ

7.1 Οδηγίες Εγκατάστασης

Για να εκτελεστεί σωστά η εφαρμογή ο χρήστης χρειάζεται να ακολουθήσει τα εξής βήματα:

Απαιτήσεις:

Python: Εγκαταστήστε την Python από τον σύνδεσμο <https://www.python.org/downloads/>.

Git Bash (προαιρετικό): Για εύκολη λήψη του repository.

Βήματα Εγκατάστασης:

Αν το Git Bash είναι εγκαταστημένο:

```
git clone https://github.com/ManvaZ1/publishing\_house\_db
```

Αλλιώς:

Ακολουθήστε τον σύνδεσμο: github.com/ManvaZ1/publishing_house_db.

Κάντε κλικ στο Code (πράσινο κουμπί) → Download ZIP.

Αποσυμπίεστε το αρχείο και ανοίξτε ένα τερματικό στο φάκελο.

Τέλος, εκτελέστε τις εξής εντολές στο τερματικό:

```
cd publishing_house_db  
  
pip install -r requirements.txt  
  
python ourDB\generate_db_records.py  
  
py
```

Σημείωση: Σε περιβάλλοντα macOS ή Linux, χρησιμοποιήστε / αντί για \ στις διαδρομές.

7.2 Παραδείγματα Χρήσης

Παράδειγμα Insert – Partner Tab:

Ο χρήστης χρειάζεται να επιλέξει από το αρχικό μενού την καρτέλα partners και ύστερα να πατήσει Insert νέου συνεργάτη, αφού συμπληρώσει τα επιθυμητά στοιχεία. Ο νέος συνεργάτης προστίθεται και εμφανίζεται στην καρτέλα partners. Σε περίπτωση που ο χρήστης προσπαθήσει συνεργάτη με Tax_Id που υπάρχει ήδη στην βάση, δεν θα του επιτραπεί και θα υπάρξει κατάλληλο error message:

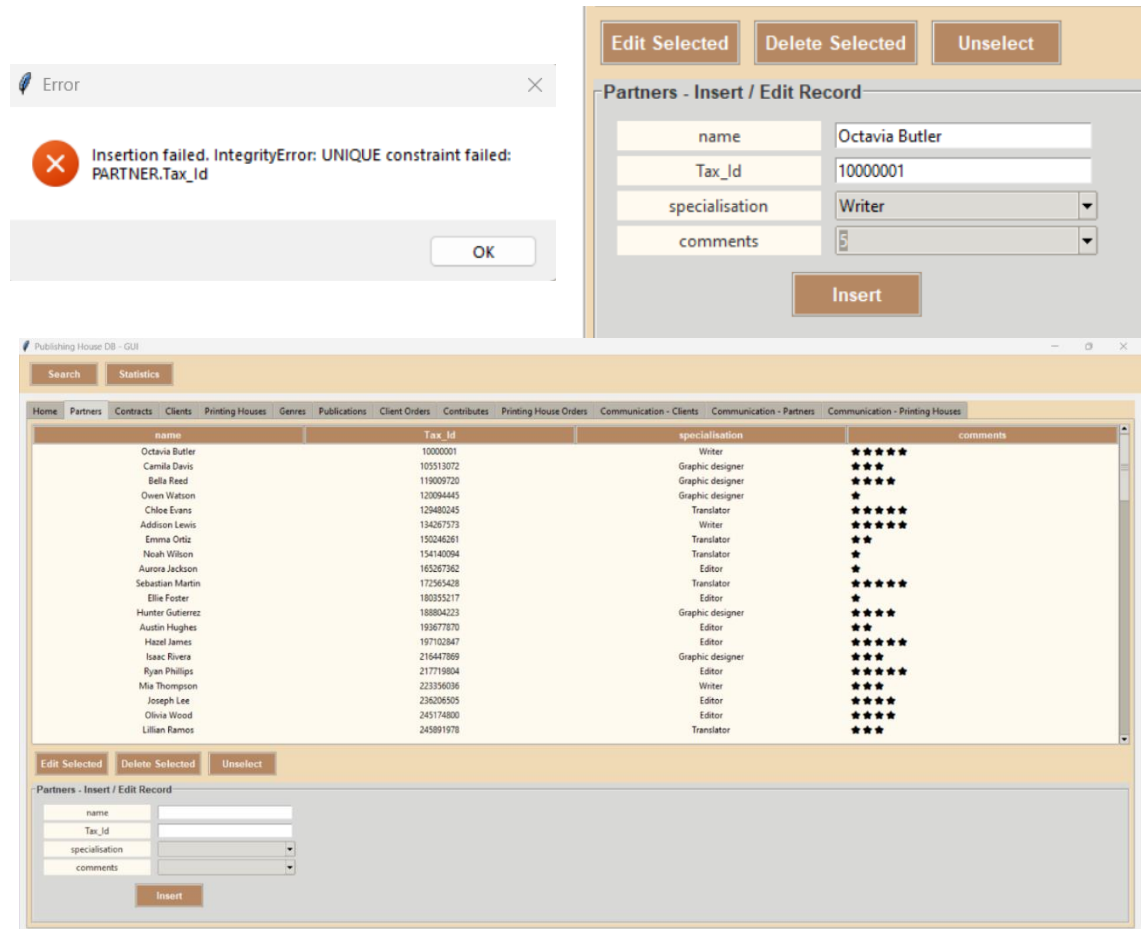


Figure 4: Παράδειγμα χρήσης Insert.

Αντίστοιχα λειτουργούν και οι υπόλοιπες επιλογές insert στα διαφορετικά tabs.

Παράδειγμα update και delete – Client Tab:

Επιλογή του Client Tab και ύστερα επιλογή με αριστερό κλικ του πελάτη που θέλουμε να επεξεργαστούμε. Με **Edit Selected** μας δίνεται η δυνατότητα να αλλάξουμε στοιχεία και με το κουμπί **Update** αποθηκεύονται. Με **Delete Selected** το στοιχείο διαγράφεται, αν δεν παραβιάζεται η αναφορική ακεραιότητα (ελέγχεται από την εφαρμογή).

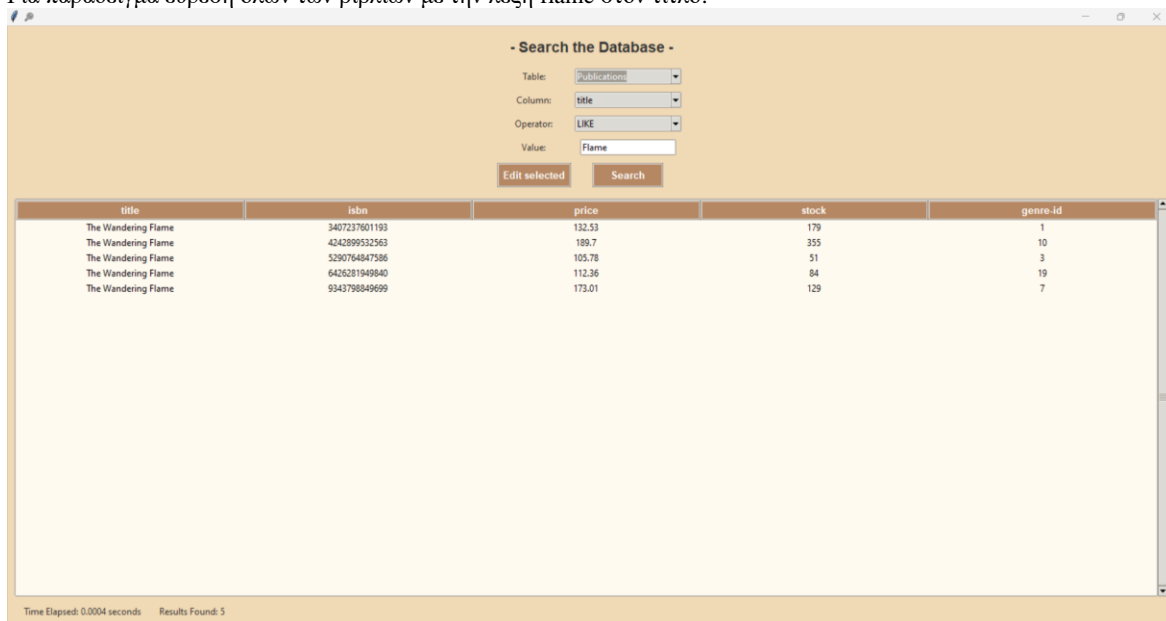


Figure 5: Παρουσίαση του GUI και χρήση Edit/Delete.

Παράδειγμα Search:

Το κουμπί Search πάνω αριστερά ανοίγει μια νέα καρτέλα. Ο χρήστης επιλέγει σε ποιον πίνακα θέλει να ψάξει, για ποιο στοιχείο του και τι τελεστή να χρησιμοποιήσει. Υποστηρίζονται οι: =, <, >, <=, >=, LIKE, δηλαδή οι τελεστές της SQL. Εμφανίζεται επίσης ο αριθμός των αποτελεσμάτων και ο χρόνος αναζήτησης κάτω αριστερά.

Για παράδειγμα εύρεση όλων των βιβλίων με την λέξη flame στον τίτλο:



| title | isbn | price | stock | genre-id |
|---------------------|---------------|--------|-------|----------|
| The Wandering Flame | 3407237601193 | 132.53 | 179 | 1 |
| The Wandering Flame | 4242899532563 | 189.7 | 355 | 10 |
| The Wandering Flame | 5290764847386 | 105.78 | 51 | 3 |
| The Wandering Flame | 6426281949840 | 112.36 | 84 | 19 |
| The Wandering Flame | 9343798849699 | 173.01 | 129 | 7 |

Figure 6: Το παράθυρο του Search. Παράδειγμα αναζήτησης του τίτλου Flame σε έντυπα.

Παράδειγμα Statistics:

Το κουμπί statistics πάνω αριστερά ανοίγει μια νέα καρτέλα με επιλογές για εμφάνιση στατιστικών:

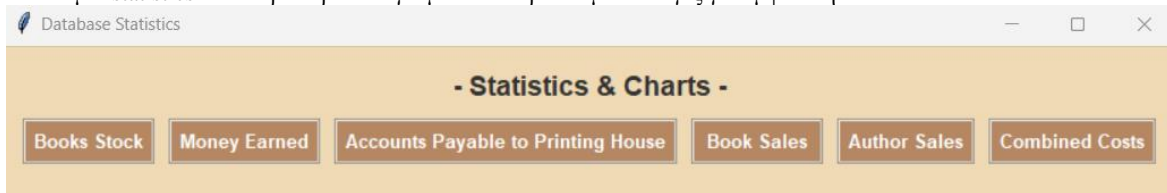


Figure 7: Οι διαθέσιμες επιλογές στατιστικών για εμφάνιση.

Κάθε κουμπί εμφανίζει το αντίστοιχο chart:



Figure 8: Γράφημα για τα χρήματα που εισέπραξε ο εκδοτικός οίκος ανά έτος.

Τα στατιστικά αποτελέσματα τυπώνονται επίσης στο terminal:

```
=====
Money Earned by Year
=====
+-----+-----+
| Year | Money Earned |
+-----+-----+
| 2010 | 88029.57 |
| 2011 | 118618.58999999998 |
| 2012 | 109931.41 |
| 2013 | 146937.63000000003 |
| 2014 | 118947.95000000001 |
| 2015 | 87809.06999999999 |
| 2016 | 102928.79999999999 |
| 2017 | 151913.12999999998 |
| 2018 | 77207.54 |
| 2019 | 121182.51999999999 |
| 2020 | 73991.29999999997 |
| 2021 | 93972.81000000003 |
| 2022 | 98121.19000000002 |
| 2023 | 109479.92000000001 |
| 2024 | 77133.92 |
+-----+-----+
```

Figure 9: Αναλυτικότερα αποτελέσματα των στατιστικών για τα έσοδα ανά έτος.