

Έκθεση ομαδικής εργασίας ομάδας 18 – Εφαρμογή Πινακοθήκης

Βάσεις Δεδομένων, 2021-2022

Σιτήστας Κωνσταντίνος Κυριάκος, Α.Μ.: 1066577 – up1066577@upnet.gr

Χαλαντζούκας Φοίβος, Α.Μ.: 1066579 - up1066579@upnet.gr

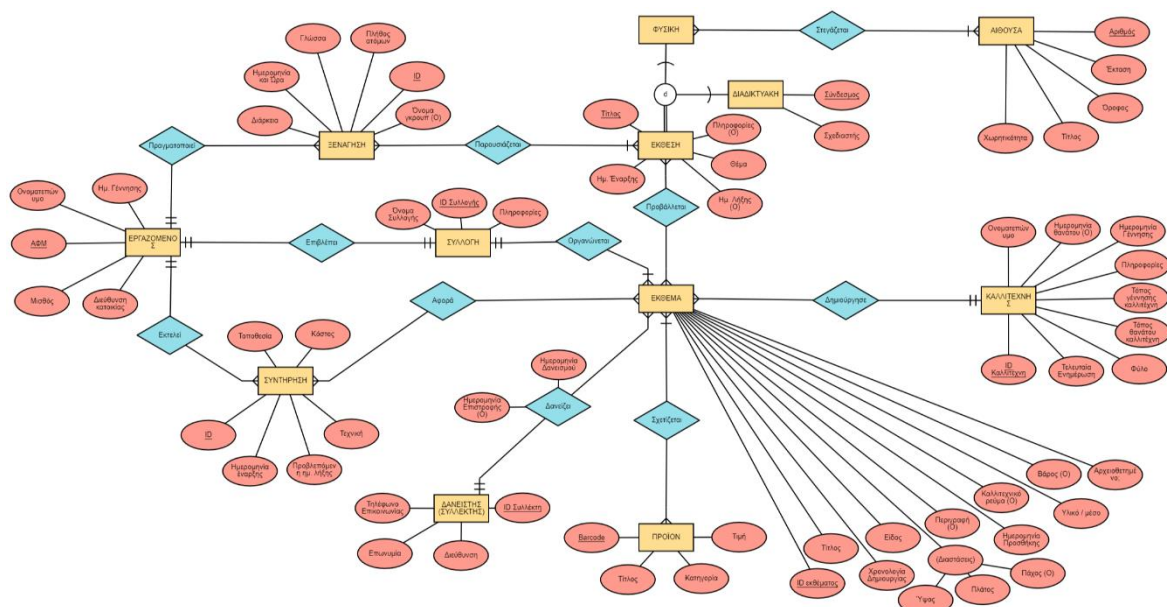
1 ΠΕΡΙΛΗΨΗ

Ζητούμενο της ομαδικής εργασίας που μας ανατέθηκε είναι η ανάπτυξη εφαρμογής που θα καλύπτει τις ανάγκες μιας πινακοθήκης. Για την δημιουργία της εφαρμογής αυτής, κάναμε έρευνα σε διάφορες πινακοθήκες του κόσμου, για να καταλάβουμε τις βασικές ιδέες οργάνωσης και λειτουργίας μιας πινακοθήκης, μιας και το θέμα ήταν αρκετά πρωτόγνωρο και για τους δυο μας. Μετά από αρκετή έρευνα και αρκετές ώρες αναζήτησης, τελικά καταλήξαμε σε έναν τρόπο οργάνωσης, αρκετά παρόμοιο με αυτόν της Εθνικής Πινακοθήκης, δημιουργώντας ωστόσο μια αρκετά βασική δομή και το αντίστοιχο εννοιολογικό μοντέλο ERD. Το διάγραμμά μας περιλαμβάνει, σύμφωνα με την κρίση μας, τις πιο βασικές πληροφορίες που αφορούν την λειτουργία μιας πινακοθήκης, δίνοντας έμφαση στα εκθέματα, καθώς αποτελούν τον βασικό λόγο ύπαρξης μιας πινακοθήκης. Γύρω από τα εκθέματα, χτίσαμε το μοντέλο μας, ενσωματώνοντας τις διάφορες λειτουργίες που λαμβάνουν χώρα στην πινακοθήκη μας. Μια βασική δυσκολία που αντιμετωπίσαμε ήταν η πολύ περιορισμένη εμπειρία μας με τον τρόπο λειτουργίας μιας πινακοθήκης, καθώς κανείς μας δεν είχε επισκεφθεί τέτοιο χώρο στο παρελθόν.

2 ΜΕΘΟΔΟΛΟΓΙΑ

Όπως αναφέρθηκε και στην περίληψη, πριν ξεκινήσουμε την δημιουργία του ERD μοντέλου για την εφαρμογή μας, αφιερώσαμε σε πρώτο στάδιο αρκετό χρόνο για να εξετάσουμε πώς λειτουργούν διάφορες πινακοθήκες ανά τον κόσμο, και πώς οργανώνουν τα έργα τους, τις συλλογές τους, τις δυνατότητες που παρέχουν στο ευρύ κοινό. Η κύρια πηγή αναζήτησης φυσικά ήταν το διαδίκτυο, μέσα από τις επίσημες ιστοσελίδες φορέων τέχνης, ωστόσο ήρθαμε και σε επαφή με άτομα στο περιβάλλον μας που μας έδωσαν κάποιες παραπάνω ιδέες για το τι θα μπορούσε να περιλαμβάνει η βάση δεδομένων μας. Όλη αυτή η διαδικασία γινόταν και από τους 2 μας, και κάναμε εβδομαδιαίες συναντήσεις για να συζητήσουμε τα συμπεράσματα που βγάλαμε και να διατηρούμε επαφή με το αντικείμενο.

Έπειτα, προχωρήσαμε στην ενσωμάτωση κάποιων βασικών οντοτήτων και συσχετίσεων που θα μπορούσαν να απαρτίζουν την βάση μας, μη δίνοντας έμφαση στα γνωρίσματα σε αρχικό στάδιο. Έπειτα, εμπλουτίσαμε το ERD με διάφορα γνωρίσματα, με έμφαση όμως στα πιο βασικά, ώστε να υπάρχει λειτουργικότητα στην εφαρμογή μας και όχι περιττή πληροφορία η οποία ενδεχομένως θα μας προκαλούσε δυσκολίες στην συνέχεια. Πρωταρχικός μας στόχος ήταν να διατηρήσουμε έναν σχετικά μικρό αριθμό οντοτήτων & γνωρισμάτων, διατηρώντας όμως την βάση μας όσο πιο λειτουργική γίνεται. Για την δημιουργία του εννοιολογικού μοντέλου, κάναμε χρήση του εργαλείου ERD Maker (<https://erdmaker.com/>). Ακολουθεί στιγμιότυπο από το τελικό εννοιολογικό μοντέλο της βάσης μας:



Με την ολοκλήρωση του εννοιολογικού διαγράμματος, συνεχίσαμε με τον μετασχηματισμό του διαγράμματος αυτού σε σχεσιακό μοντέλο, κάτι που πραγματοποιήσαμε με χρήση του εργαλείου DB Designer (<https://www.dbdesigner.net/>).

Κατά τον μετασχηματισμό εφαρμόσαμε τα βήματα που παρουσιάστηκαν κατά την διάρκεια των διαλέξεων, όσον αφορά τις διάφορες συσχετίσεις, ειδικεύσεις, κ.λπ.. Ιδιαίτερη προσοχή δώσαμε στην πολλαπλότητα των σχέσεων μεταξύ των οντοτήτων, από την οποία προέκυψαν κάποιοι βοηθητικοί πίνακες στο σχεσιακό μοντέλο. Στην παραπάνω εικόνα (Εικόνα 2) φαίνεται η δομή του σχεσιακού μοντέλου, καθώς και τα Foreign Keys τα οποία είχαμε δηλώσει ήδη από την δημιουργία του.

Αν και η πλατφόρμα του DB Designer προσφέρει την δυνατότητα εξαγωγής του σχεσιακού μοντέλου σε γλώσσα SQLite, υπάρχουν κάποιες ατέλειες στον μετασχηματισμό του σε SQL και πολλές επιλογές δεν περνούσαν στην SQL. Για αυτό τον λόγο, με εντολές SQL δημιουργήσαμε τους διάφορους πίνακες της βάσης δεδομένων μας (το πρώτο μέρος του αρχείου (art-gallery.db.sql που περιλαμβάνεται στα παραδοτέα). Στο αρχείο αυτό, περιλαμβάνεται στην δημιουργία των πινάκων και ο ορισμός των κλειδιών (πρωτεύοντων και ξένων), ο τύπος των στοιχείων, αν υπάρχει κάποια προεπιλεγμένη τιμή για την δημιουργία των δεδομένων και αν κάποιο χαρακτηριστικό μπορεί να παραμείνει κενό κατά την εισαγωγή. Αφού καταχωρήσαμε έναν μικρό αριθμό εγγραφών στην βάση μας, εκτελέσαμε κάποια βασικά queries στο περιβάλλον του DB Browser for SQLite για να δοκιμάσουμε τα δεδομένα μας. Τέλος, χρησιμοποιήσαμε την ίδια εφαρμογή και για την προσθήκη όλων των υπόλοιπων δεδομένων της βάσης, καθώς ήταν πολύ πιο γρήγορο από το να εισάγουμε τα δεδομένα με χρήση εντολών SQL. Με την ολοκλήρωση της εισαγωγής δεδομένων στην βάση μας, χρησιμοποιήσαμε την δυνατότητα του της εφαρμογής DB Browser για εξαγωγή των δεδομένων σε μορφή SQL, και έτσι λοιπόν προέκυψε το τελικό αρχείο art-gallery.db.sql που μπορεί να χρησιμοποιηθεί για ορισμό των πινάκων αλλά και εισαγωγή των δεδομένων της βάσης.

Έχοντας ολοκληρώσει τη σχεδίαση της βάσης και έχοντας εισάγει ένα αρχικό πλήθος δεδομένων σε αυτή, προχωρήσαμε στη σχεδίαση μιας εφαρμογής σε γλώσσα Python, έχοντας ως στόχο να μπορούμε να προσπελάσουμε και να επεξεργαστούμε τη βάση μας μέσω της εφαρμογής αυτής. Κύριο μέλημα της σχεδίασής μας ήταν να δίνεται στο χρήστη η δυνατότητα να εκτελεί πλήθος διαφορετικών λειτουργιών - άλλες πιο απλές και άλλες πιο σύνθετες - μέσα από μία ενιαία πλατφόρμα. Η εφαρμογή εν τέλει διαρθρώθηκε γύρω από δύο βασικούς άξονες: την αναζήτηση και ανάκτηση δεδομένων, και την εισαγωγή ή επεξεργασία αυτών.

Το πρώτο βήμα στη σχεδίαση μας ήταν η δημιουργία μίας απλής διεπαφής με τη χρήση της βιβλιοθήκης tkinter, μέσω της οποίας ο χρήστης θα μπορούσε να πραγματοποιήσει αναζήτηση και ανάκτηση δεδομένων σχετικά με μία οντότητα, καθώς και να εμφανίσει τα δεδομένα αυτά. Ακολούθως, εμπλουτίσαμε την διαδικασία της αναζήτησης, αυξάνοντας τα κριτήριά της και δίνοντας τη δυνατότητα στο χρήστη να παίρνει πιο εξειδικευμένα αποτελέσματα για μία οντότητα. Για να ολοκληρώσουμε το κομμάτι της ανάκτησης δεδομένων, προσθέσαμε ένα πλήθος κουμπιών που υλοποιούν κάποιες καθορισμένες αναζητήσεις, οι οποίες αφορούν τυπικά σενάρια χρήσης της βάσης.

Έχοντας ολοκληρώσει το κομμάτι της αναζήτησης, περάσαμε στο κομμάτι της διαχείρισης των δεδομένων. Στο πλαίσιο αυτής της διαδικασίας προσθέσαμε μια σειρά από κουμπιά, που επιτρέπουν στον χρήστη να εισάγει νέα δεδομένα στη βάση, να επεξεργάζεται τα υπάρχοντα δεδομένα, ή ακόμα και να διαγράφει δεδομένα από τη βάση όπου αυτό απαιτείται και επιτρέπεται. Καθόλη τη διάρκεια της υλοποίησης αυτών των επιλογών, δώσαμε την απαραίτητη προσοχή στο να τηρούνται οι περιορισμοί και οι κανόνες ακεραιότητας που είχαμε θέσει κατά τη σχεδίαση της βάσης μας. Τέλος, προσθέσαμε ένα πλαίσιο μέσω του οποίου είναι δυνατή η εκτέλεση οποιασδήποτε ακολουθίας εντολών σε γλώσσα SQL, όπως γίνεται στα προγράμματα διαχείρισης βάσεων δεδομένων.

Η τελική έκδοση της διεπαφής μας έχει την ακόλουθη μορφή:

Εικόνα 3: Στιγμιότυπο της εφαρμογής υλοποιημένης με γλώσσα Python

3 ΑΞΙΟΛΟΓΗΣΗ

Σαν πρώτο στάδιο αξιολόγησης είχαμε αρχικά, όσον αφορά την δημιουργία του εννοιολογικού μοντέλου, την παρουσίαση που πραγματοποιήσαμε στα μέσα του εξαμήνου. Αφότου λάβαμε σχόλια σχετικά με το μοντέλο μας, προσπαθήσαμε να βελτιώσουμε τα σημεία που μας επισημάνθηκαν και να δουλέψουμε περαιτέρω πάνω σε αυτό, ώστε να απαιτούνται όσο το δυνατόν λιγότερες αλλαγές στην συνέχεια. Το επόμενο στάδιο αξιολόγησης αφορούσε τη μετατροπή του εννοιολογικού μοντέλου στο αντίστοιχο σχεσιακό διάγραμμα. Καθώς πραγματοποιούσαμε τη μετατροπή αυτή, δώσαμε πρωτίστως έμφαση στη σωστή αποτύπωση των συσχετίσεων του εννοιολογικού διαγράμματος στο αντίστοιχο σχεσιακό μοντέλο, ώστε να αποτυπώνονται με τον κατάλληλο τρόπο – είτε με τη δημιουργία πινάκων συσχέτισης, είτε με την δημιουργία ξένων κλειδιών – οι σχέσεις πολλαπλότητας ανάμεσα στις οντότητες του εννοιολογικού διαγράμματός μας. Όσον αφορά την αξιολόγηση της βάσης, ως θεμελιώδη στόχο είχαμε τον αυστηρό καθορισμό όλων των κανόνων που είχαμε θεσπίσει κατά τη σχεδίαση του εννοιολογικού και του σχεσιακού διαγράμματος, και στους οποίους όφειλαν να υπόκεινται οι πίνακες των δεδομένων μας. Τέλος, σχετικά με το τελευταίο κομμάτι του πρότζεκτ μας, την εφαρμογή που αναπτύξαμε σε γλώσσα python, είχαμε ως στόχο την δημιουργία μιας «ολοκληρωμένης» εφαρμογής, η οποία θα έχει ως κύρια χαρακτηριστικά 1) μικρό (έως ιδανικά μηδαμινό) πλήθος bugs/δυσλειτουργιών, και 2) ικανοποιητικά μεγάλο εύρος δυνατοτήτων που παρέχονται στο χρήστη. Ως εκ τούτου θεωρήσαμε σκόπιμο να μην δώσουμε τόση έμφαση στο κομμάτι της ωραιοποίησης και των γραφικών, δίνοντας προτεραιότητα στη λειτουργικότητα και τη σταθερότητα της εφαρμογής μας.

4 ΔΕΔΟΜΕΝΑ

Σαν κύρια πηγή δεδομένων επιλέξαμε την Εθνική Πινακοθήκη, καθώς περιλαμβάνει μεγάλο αριθμό έργων, καλλιτεχνών και άλλων δεδομένων που μας διευκόλυναν την διαδικασία όσον αφορά την συλλογή έργων, καλλιτεχνών, συλλογών και εκθέσεων. Δυστυχώς, δεν υπήρχε κάποιος τρόπος για μαζική λήψη δεδομένων από την πινακοθήκη, ή και από άλλες πηγές στο διαδίκτυο, καθώς είτε δεν υπήρχε τρόπος για την άντλησή τους, είτε τα δεδομένα δεν ικανοποιούσαν τις προδιαγραφές της βάσης μας και θα χρειαζόταν πολύς κόπος για την εύρεση των απαραίτητων δεδομένων. Συνεπώς, η εισαγωγή των δεδομένων πραγματοποιήθηκε χειροκίνητα από την Εθνική Πινακοθήκη, κάτι που σημαίνει ότι περιορίστηκε σημαντικά το πλήθος τους σε σύγκριση με τα πραγματικά δεδομένα της πινακοθήκης, ωστόσο η λειτουργικότητα μπορεί να επεκταθεί εύκολα και για πολύ μεγαλύτερο πλήθος δεδομένων.

5 ΚΑΤΑΜΕΡΙΣΜΟΣ ΕΡΓΑΣΙΩΝ

Σε γενικές γραμμές, και οι 2 μας ασχοληθήκαμε από κοινού σε όλες τις φάσεις δημιουργίας της βάσης μας, καθώς και της ανάπτυξης του κώδικα και βρήκαμε κάποιες κοινές ώρες που δουλεύαμε παράλληλα, ώστε να μπορούμε εύκολα να συνεννοηθούμε για σημεία στα οποία χρειάζονταν κοινές αποφάσεις. Ωστόσο, ένα σημείο που χωρίσαμε κάπως ρόλους ήταν το στάδιο μετά την αρχική δημιουργία της βάσης δεδομένων, στο οποίο ο Κωνσταντίνος συνέχισε με τον εμπλουτισμό της βάσης δεδομένων με παραπάνω στοιχεία, ενώ ο Φοίβος ξεκίνησε να αναπτύσσει τον κορμό της εφαρμογής Python και της γραφικής διεπαφής της, με κάποια βασικά χαρακτηριστικά και λειτουργικότητα. Μετά την ολοκλήρωση της βάσης, και οι 2 συνεργαστήκαμε στην προσθήκη νέων λειτουργιών της εφαρμογής και την βελτίωση της γραφικής διεπαφής της. Η δημιουργία του ERD, του λογικού σχήματος, της αναφοράς και της παρουσίασης πραγματοποιήθηκε με συνεισφορά και από τα δύο μέλη της ομάδας.

6 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Για την δημιουργία της εφαρμογής ακολουθήσαμε τις 3 φάσεις που προτείνονταν στο εγχειρίδιο του μαθήματος, δηλαδή την δημιουργία του εννοιολογικού μοντέλου, και τον μετασχηματισμό του σε αντίστοιχο λογικό μοντέλο, έπειτα ασχοληθήκαμε με την σύνταξη των εντολών SQL για την βάση μας και τον εμπλουτισμό της με δεδομένα, ενώ σε τελευταίο στάδιο δημιουργήσαμε την εφαρμογή μας με χρήση Python ώστε ο χρήστης να μπορεί να αλληλεπιδράσει με την βάση. Αναλυτικά το χρονοδιάγραμμα που ακολουθήσαμε φαίνεται παρακάτω:

- Νοέμβριος – μέχρι την ενδιάμεση παρουσίαση του Project: Έρευνα για την δομή και λειτουργία διαφόρων πινακοθηκών στην Ελλάδα και τον κόσμο – δημιουργία του εννοιολογικού διαγράμματος της βάσης δεδομένων
- Τέλη Νοεμβρίου – Μέσα Δεκεμβρίου: Βελτίωση του εννοιολογικού διαγράμματος μετά την ενδιάμεση παρουσίαση
- Μέσα Δεκεμβρίου: Μετασχηματισμός του εκτεταμένου διαγράμματος οντοτήτων συσχετίσεων στο αντίστοιχο λογικό σχεσιακό μοντέλο
- Τέλη Δεκεμβρίου: Δημιουργία της αρχικής βάσης δεδομένων με χρήση εντολών SQL (SQLite), προσθήκη στοιχείων στην βάση δεδομένων και εκκίνηση ανάπτυξης εφαρμογής σε περιβάλλον Python, ενσωμάτωση βασικών queries για αναζήτηση
- Αρχές Ιανουαρίου: Προσθήκη δυνατοτήτων στην εφαρμογή Python (σύνθετα queries αναζήτησης, εντολές CRUD), βελτίωση γραφικών
- Τελευταία εβδομάδα: Προετοιμασία αναφοράς και παρουσίασης της ομαδικής εργασίας

7 ΠΑΡΑΡΤΗΜΑ

7.1 ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ

- 1) Κατεβάζουμε το αρχείο με κατάληξη .db και ένα από τα εκτελέσιμα αρχεία με κατάληξη .exe, .py, από τον σύνδεσμο: https://github.com/sitistas/art-gallery-ECE_CK703-team18
- 2) Τοποθετούμε το αρχείο .db στον ίδιο φάκελο με το εκτελέσιμο αρχείο που έχουμε κατεβάσει
- 3) Τρέχουμε το εκτελέσιμο αρχείο

7.2 ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ ΕΦΑΡΜΟΓΗΣ

The screenshot shows a web application interface with a light blue background. At the top, there is a header bar with the text 'Εφαρμογή Πανεπιστημίου - Ομάδα 18'. Below the header, there is a search section with a 'Choose entity:' dropdown, a 'Search term:' input field, a 'Choose criterion:' dropdown, and 'Search' and 'Show all values' buttons. Below this is an 'Advanced Search' section with four criteria dropdowns (Criterion 1 to 4) and several buttons: 'Ενέργηση σε οθίσκους', 'Έργο καλλιτεχνή', 'Έργο ανά ύψος', 'Διατεταμένα έργα', '#Έργων ανά συλλογή', and 'Έργα που συστηρεύονται'. At the bottom of the interface, there is a row of buttons: 'Insert', 'Edit selection', 'Delete selection', 'Manage exhibition', 'Run a query', and 'Close'.

- **Choose entity:** Δίπλα σε αυτό το Label έχουμε ένα μενού επιλογών τύπου **Combobox**, το οποίο αποτελεί το πιο σημαντικό ίσως αντικείμενο της εφαρμογής μας, καθώς η επιλογή του επηρεάζει τη λειτουργία των υπόλοιπων γραφικών στοιχείων (κουμπιά και comboboxes).
- **Search term:** Στα δεξιά αυτού του label έχουμε τη δομή της απλής αναζήτησης

This close-up shows the search section of the application interface. It includes a 'Choose entity:' dropdown menu, a 'Search term:' input field, a 'Choose criterion:' dropdown menu, and 'Search' and 'Show all values' buttons.

Για να πραγματοποιήσουμε απλή αναζήτηση:

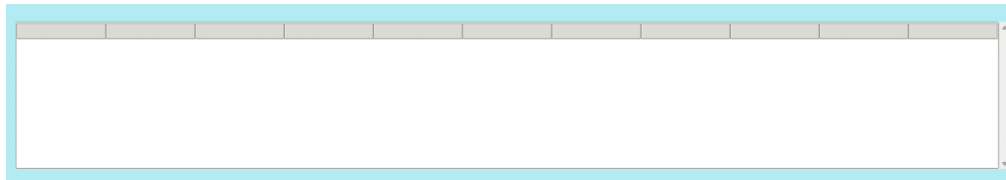
- 1) Επιλέγουμε από το combobox δίπλα στο choose entity την οντότητα για την οποία θέλουμε να κάνουμε αναζήτηση.

2) Αν θέλουμε να δούμε όλες τις εγγραφές του σχετικού πίνακα, πατάμε το κουμπί **Show all Values**. Εναλλακτικά, εισάγουμε στο πλαίσιο κειμένου δίπλα στο search term τον όρο αναζήτησης που επιθυμούμε. Ακολουθώντας, επιλέγουμε από το combobox δεξιά του **‘Choose criterion’** το γνώρισμα στο οποίο θέλουμε να αντιστοιχίσουμε την τιμή που εισάγουμε, και τέλος πατάμε στα δεξιά το κουμπί **Search** για να δούμε τα αποτελέσματα που προκύπτουν. Η αναζήτηση δεν είναι αυστηρή, και οι τιμές γνωρισμάτων που γίνονται δεκτές μπορούν είτε να είναι ίδιες με την τιμή που έχουμε εισάγει, είτε μπορεί να περιλαμβάνουν αυτή.

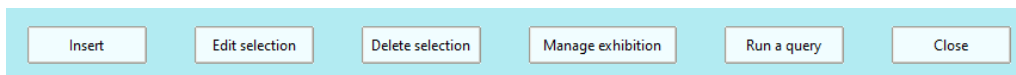
- Κάτω από το Choose entity έχουμε το μενού της **σύνθετης αναζήτησης Advanced search**. Η αναζήτηση αυτή αφορά πάλι μία μόνο οντότητα, αυτή που έχει επιλεγεί στο κεντρικό combobox.

- Στο μενού αυτό έχουμε **4 πλαίσια εισόδου και 4 comboboxes**. Κάθε ένα σέτ πλαισίου-combobox λειτουργεί όπως το πλαίσιο απλής αναζήτησης, με τη διαφορά ότι τα 4 αυτά σέτ συνδυάζονται μεταξύ τους για να μας δώσουν πιο εξειδικευμένα αποτελέσματα απ' ό,τι η απλή αναζήτηση. Αφού συμπληρώσουμε όσα από τα πλαίσια κειμένου θέλουμε και αφού επιλέξουμε τα κατάλληλα κριτήρια δεξιά, πατώντας το κουμπί Advanced search πραγματοποιούμε τη σύνθετη αναζήτηση. Όπως και η απλή αναζήτηση, η σύνθετη αναζήτηση δεν είναι αυστηρή.
- Δεξιά από το τμήμα της σύνθετης αναζήτησης έχουμε **6 κουμπιά**, 3 από τα οποία συνοδεύονται από ένα πλαίσιο κειμένου.

- **Ξενάγηση σε εκθέσεις:** Εμφανίζει τις εκθέσεις τις οποίες πρόκειται να επισκεφθεί κάθε γκρουπ που έχει ID ή όνομα όμοιο ή ίδιο με την τιμή που έχουμε εισάγει στο αντίστοιχο πλαίσιο εισόδου.
 - **Έργα καλλιτέχνη:** Εμφανίζει όλα τα έργα όπου το όνομα ή το ID του καλλιτέχνη που τα έχει σχεδιάσει είναι ίδιο ή όμοιο με την τιμή που έχουμε εισάγει στο αντίστοιχο πλαίσιο εισόδου.
 - **Έργα ανά έκθεση:** Εμφανίζει όλα τα έργα όπου εμφανίζονται σε κάποια έκθεση όπου το όνομα ή το ID της έκθεσης είναι ίδιο ή όμοιο με την τιμή που έχουμε εισάγει στο αντίστοιχο πλαίσιο εισόδου.
 - **Δανεισμένα έργα:** Εμφανίζει όλα τα έργα που βρίσκονται στην κατοχή της πινακοθήκης αλλά δεν ανήκουν σε αυτή, και τα οποία βρίσκονται εκεί με δανεισμό καθορισμένης ή αορίστου διάρκειας.
 - **#Εργων ανά συλλογή:** Εμφανίζει τα ονόματα και το πλήθος όλων των συλλογών που έχουν πάνω από ένα έργο
 - **Έργα που συντηρούνται:** Εμφανίζει τα έργα που απουσιάζουν από την πινακοθήκη για λόγους συντήρησης.
- Κάτω από όλα αυτά υπάρχει μια δομή τύπου **treeview**, στην οποία τυπώνονται τα αποτελέσματα κάθε αναζήτησης.



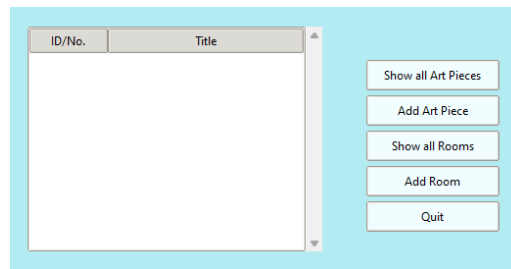
- Κάτω από το treeview βρίσκονται 6 επιπλέον πλήκτρα, 4 από τα οποία χρησιμοποιούνται για τη διαχείριση των δεδομένων της βάσης.



- **Insert:** Δημιουργεί ένα παράθυρο όπου μπορούμε να δημιουργήσουμε μία νέα εγγραφή στον πίνακα της οντότητας που έχουμε επιλέξει στο κεντρικό Combobox 'Choose entity'. Στο παράθυρο αυτό υπάρχουν 2 κουμπιά **Submit** (υποβολής των δεδομένων που εισάγει ο χρήστης) και **Quit** (εξόδου από το παράθυρο εισόδου).
Σε περίπτωση που η επιλεγμένη οντότητα είναι μία από τις 'Maintenance', 'Guided Tour', 'Product', αν ολοκληρωθεί επιτυχώς η εισαγωγή των νέων δεδομένων, θα εμφανιστεί ένα νέο παράθυρο για την εισαγωγή κάποιας τιμής για τη δημιουργία των αντίστοιχων πινάκων συσχετίσεων
- **Edit selection:** Δημιουργεί ένα παράθυρο επεξεργασίας, όπου μπορούμε να επεξεργαστούμε τα δεδομένα μιας εγγραφής. Για να λειτουργήσει σωστά το κουμπί οφείλουμε να έχουμε επιλέξει μια εγγραφή από το treeview, και να μην έχουμε αλλάξει την τιμή του κεντρικού Combobox μετά την τελευταία αναζήτηση.

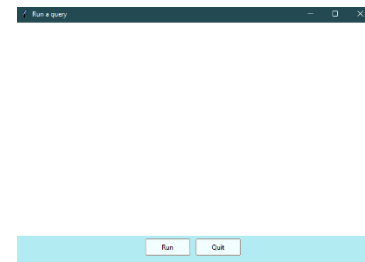
Αν π.χ. ένας χρήστης αναζητήσει και εμφανίσει όλα τα έργα τέχνης στο treeview, επιλέξει ένα από αυτά, και πατήσει 'Edit selection' αφότου έχει αλλάξει την τιμή του κεντρικού Combobox, η λειτουργία επεξεργασίας δε θα λειτουργήσει σωστά.

- **Delete selection:** Διαγράφει το αντικείμενο που έχουμε επιλέξει στο treeview. Αρχικά προειδοποιητικό παράθυρο στο χρήστη, όπου ο χρήστης μπορεί να εγκρίνει ή να απορρίψει τη διαγραφή του αντικειμένου που έχει επιλέξει πριν το πάτημα του κουμπιού 'Delete'. Όπως και προηγουμένως, ο χρήστης οφείλει να μην έχει αλλάξει την τιμή του κεντρικού Combobox αφότου έχει πραγματοποιήσει την τελευταία αναζήτησή του. Δηλαδή η οντότητα του αντικειμένου που έχουμε επιλέξει στο treeview για να διαγράψουμε, πρέπει να είναι ίδια με αυτή που εμφανίζεται στο κεντρικό Combobox.
- Μετά την εισαγωγή, επεξεργασία, ή διαγραφή δεδομένων, ο χρήστης πρέπει να πραγματοποιήσει εκ νέου αναζήτηση για να δει τις αλλαγές.
- **Manage exhibition:** Δημιουργεί παράθυρο επεξεργασίας ειδικά για μια έκθεση



- 1) Ο χρήστης επιλέγει μια έκθεση από το treeview και βεβαιώνεται ότι το 'Choose entity' έχει επιλεγμένη τη λέξη 'Exhibition'.
- 2) Ο χρήστης πατάει manage exhibition και του εμφανίζεται ένα παράθυρο διαχείρισης της έκθεσης που έχει επιλέξει. Το παράθυρο περιλαμβάνει 5 κουμπιά και ένα treeview όπου εμφανίζει όλα τα αποτελέσματα.
 - a. **Show all art pieces:** Εμφανίζει όλα τα έργα που εμφανίζονται στη συγκεκριμένη έκθεση
 - b. **Add art piece:** Εμφανίζει πλαίσιο εισόδου που ο χρήστης μπορεί να εισάγει τον τίτλο ενός έργου τέχνης που θέλει να προσθέσει στην επιλεγμένη έκθεση
 - c. **Show all rooms:** Εμφανίζει όλες τις αίθουσες στις οποίες λαμβάνει χώρα η συγκεκριμένη έκθεση
 - d. **Add room:** Εμφανίζει πλαίσιο εισόδου που ο χρήστης μπορεί να εισάγει τον αριθμό μιας αίθουσας που θέλει να προσθέσει στην επιλεγμένη έκθεση
 - e. **Quit:** Κλείνει το παράθυρο διαχείρισης

- **Run a query:** Εμφανίζει ένα νέο παράθυρο στο οποίο ο χρήστης μπορεί να γράψει εντολές SQL και να τις εκτελέσει πατώντας το κουμπί Run. Τα αποτελέσματα τυπώνονται σε ένα νέο παράθυρο σε μορφή treeview. Το κουμπί Quit κλείνει το παράθυρο εντολών.



- **Close:** Το κουμπί αυτό κλείνει το κεντρικό παράθυρο της εφαρμογής και τερματίζει τη σύνδεση με τη βάση δεδομένων.

Αν το παράθυρο που προκύπτει από το κουμπί 'Run a query' είναι ακόμα ανοιχτό, ο χρήστης μπορεί να συνεχίσει να δουλεύει κανονικά σε εκείνο.

7.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΑΝΤΙΣΤΟΙΧΟΣ ΚΩΔΙΚΑΣ

7.3.1 Απλή αναζήτηση

Choose entity: Piece_of_art Search term: 68 Choose criterion: ID Search Show all values

Advanced Search

Criterion 1: Full_Name Criterion 2: Birth_date Criterion 3: Date_of_death Criterion 4: Sex

Εισαγωγή σε εκθέσεις Έργα κολλιτέχνη Έργα ανά έκθεση Δανεισμένα έργα #Έργων ανά συλλογή Έργα που συντηρούνται

Advanced Search

ID	Title	Artist_ID	Collection_ID	Year_of_creation	Acquisition_Date	Date_added	Type	Material	Height
368	Αρκάδιω Τονισ	1	2	~1600	2020-01-01	2020-01-01	Πένα, πινέλο σε χαρτί	Πένα, πινέλο	21.2

```
#Γενική Αναζήτηση
def search_by():
    try:
        entity = cbox1.get() #Ανάκτηση οντότητας
        criter = cbox2.get() #Ανάκτηση κριτηρίου αναζήτησης
        term = artist_search_entry.get() #Ανάκτηση εισόδου χρήστη
        query_str="SELECT * FROM "+entity+" WHERE "+criter+" LIKE '"+str.title(term)+"%" #Query
        find(query_str) #Εκτέλεση query και εκτύπωση αποτελεσμάτων
    except: pass
```

```
#Συνάρτηση εκτέλεσης query, ανάκτησης και εμφάνισης αποτελεσμάτων
def find(query):
    clear_treeview() #Εκκαθάριση του treeview
    c.execute(query) #Εκτέλεση query
    data1 = c.fetchall() #Ανάκτηση αποτελεσμάτων
    tree_view_insert(data1) #Εισαγωγή αποτελεσμάτων στο treeview
    printcolumns(data1) #Ρύθμιση στηλών του treeview
```

```
#Εισαγωγή των αποτελεσμάτων ενός query στο Treeview
def tree_view_insert(data):
    for row in data:
        tree_view.insert('', 'end', values=row)
```

7.3.2 Σύνθετη αναζήτηση

Choose entity:

Artist

Search term:

Choose criterion:

ID

Search

Show all values

Advanced Search

r

Criterion 1

Full_Name

17

Criterion 2

Birth_date

18

Criterion 3

Date_of_death

A

Criterion 4

Sex

Ξανάγηση σε ελθέιους

Έργα καλλιτέχνη

Έργα ανά έλθέιη

Δαντελμένα έργα

#Έργων ανά σελόγη

Έργα που σντηντρούνται

Advanced Search

ID	Full_Name	Last_updated	Birth_date	Date_of_deat	Information	Place_of_birth	Place_of_death	Sex
6	Goya y Lucientes Francisco (Friday t.	2021-12-23 15:30	1746	1828	Ισπανός ζωγράφος και χαράκτης. Α	Φουεντεντενός	Μπαρόν	A

```
#20v0teta anazήτηση
def adv_search():
    try:
        entity = cbox1.get() #Ανάκτηση οντότητας
        critr1 = crit1_box.get() #Ανάκτηση 1ου κριτηρίου αναζήτησης
        critr2 = crit2_box.get() #Ανάκτηση 2ου κριτηρίου κ.ο.κ.
        critr3 = crit3_box.get()
        critr4 = crit4_box.get()
        term1 = crit1_search_entry.get() #Ανάκτηση εισόδου για το πρώτο κριτήριο
        term2 = crit2_search_entry.get() #Ανάκτηση για το 2ο κ.ο.κ.
        term3 = crit3_search_entry.get()
        term4 = crit4_search_entry.get()
        temp1=[critr1, critr2, critr3, critr4] #Λίστα με τις τιμές όλων των comboboxes των κριτηρίων
        temp2=[term1, term2, term3, term4] #Λίστα με τις τιμές όλων των σχετικών entries
        critlist=[] #Λίστα με τα κριτήρια που έχουμε επιλέξει
        termlist=[] #Λίστα με τις εισόδους όπου έχουμε δώσει κάποια τιμή
        for i in range(len(temp1)):
            if temp1[i]!='':
                critlist.append(temp1[i])
                termlist.append(temp2[i])
        if termlist==[]: return
        #Εκτέλεση query
        query_str="SELECT * FROM "+entity+" WHERE "+critlist[0]+" LIKE '%"+str.title(termlist[0])+"%'"
        for i in range(1, len(termlist)):
            query_str+=" AND "+critlist[i]+" LIKE '%"+str.title(termlist[i])+"%'"
        find(query_str)
    except: pass
```

7.3.3 Έργα καλλιτέχνη «Γύζης»

Criterion 1
 Criterion 2
 Criterion 3
 Criterion 4

Painting_ID	Title	Full_Name	Collection_Name	Year_of_creat	Type	Height	Width	Depth
119	Γάντες	Γύζης Νικόλαος	Νεοελληνική Ζωγραφική	1878	Λάδι σε μουσαμά	46.0	37.0	None
602	Οι κόρες του καλλιτέχνη, Πηνελόπη	Γύζης Νικόλαος	Νεοελληνική Ζωγραφική	1895	Κάρβουνο και κιμωλία σε σκούρο χρ	34.0	27.0	None
607	Παποθήκη	Γύζης Νικόλαος	Νεοελληνική Ζωγραφική	1885	Λάδι σε μουσαμά	35.0	27.0	None
618	Νέος Αιών	Γύζης Νικόλαος	Νεοελληνική Ζωγραφική	1895	Λάδι σε μουσαμά	48.0	39.0	None
635	Εσπρινή Συμφωνία	Γύζης Νικόλαος	Νεοελληνική Ζωγραφική	1886	Λάδι σε μουσαμά	102.0	139.0	None
1858	Η Χαρτομάντισσα	Γύζης Νικόλαος	Νεοελληνική Ζωγραφική	1858	Λάδι σε μουσαμά	72.0	92.0	None

```
#Αναζήτηση έργων τέχνης ενός καλλιτέχνη
def pieces_of_artist():
    artist_selection = str.title(pieces_of_artist_entry.get()) #Ανάκτηση εισόδου για τον καλλιτέχνη
    query = """SELECT p.ID as Painting_ID, Title, t.Full_name, c.name as Collection_Name, Year_of_creation, Type, Height, Width, Depth
    FROM (Artist as a JOIN Piece_of_Art as p ON a.ID = p.Artist_ID) as t
    JOIN Collection as c ON c.ID = t.Collection_ID WHERE a.Full_Name LIKE '%'+ artist_selection + '%'"""
    find(query)
```

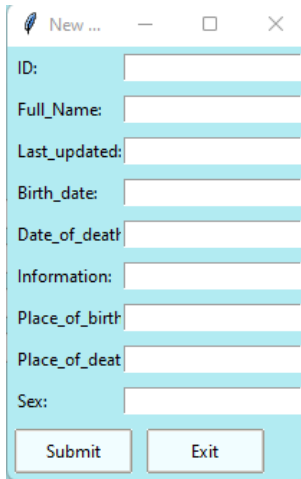
7.3.4 Έργα που συντηρούνται

Criterion 1
 Criterion 2
 Criterion 3
 Criterion 4

Title	Starting_date	Expected_return
Η συνουσία των Αγγέλων	2021-12-31	2022-7-5
Φύλλο Με Δώδεκα Αρχαία Νομίσμα	2021-12-31	2022-8-25
Γυμνό	2021-12-31	2022-2-20
Ανατολήτης	2021-12-31	2022-9-21
Αραγμένα Καράβια	2021-12-31	2022-5-5
Αυτοπροσωπογραφία	2021-12-31	2022-9-7
Αυτοπροσωπογραφία	2021-12-31	2022-6-9
Κεφάλι Νέου	2021-12-31	2022-4-11

```
#Συνάρτηση για εύρεση των έργων που απουσιάζουν λόγω συντήρησης
def maint_poa():
    query="""SELECT Title, Starting_date, Expected_return
    FROM (Piece_of_Art as p join Under_maint as u on u.AP_ID = p.ID)
    JOIN Maintenance as m on m.ID = u.Maint_ID WHERE Starting_date < datetime('now')
    AND Expected_return > datetime('now')"""
    find(query)
```

7.3.5 Insert Artist



New ...

ID:

Full_Name:

Last_updated:

Birth_date:

Date_of_death:

Information:

Place_of_birth:

Place_of_death:

Sex:

```
#Συνάρτηση insert
def new():
    try:
        attr_list=[]
        ont=cbox1.get() #Ανάκτηση κεντρικής οντότητας
        c.row_factory = lambda cursor, row: row[0] #STRs αντί για tuples
        c.execute("SELECT name FROM PRAGMA_table_info('"+ont+"')") #Ανάκτηση ονομάτων attributes αυτής της οντότητας
        data=c.fetchall()
        for i in data:
            attr_list.append(i)
        c.row_factory = None #Αναίρεση της προηγούμενης αλλαγής για τα tuples

        new_win=Tk() #Δημιουργία παραθύρου εισαγωγής δεδομένων
        add_style(new_win)
        new_win.title('New data')
        ents=makeform(new_win, attr_list, 'new') #Διαμόρφωση παραθύρου εισαγωγής δεδομένων
        #Κουμπιά αποθήκευσης και εξόδου
        sub_btn = ttk.Button(new_win, text = 'Submit',command=(lambda:insert_data(new_win, ents, ont)))
        sub_btn.pack(side = LEFT, padx = 5, pady = 5)
        quit_btn = ttk.Button(new_win, text = 'Exit', command=new_win.destroy)
        quit_btn.pack(side = LEFT, padx = 5,    pady = 5)
        new_win.mainloop()
    except: pass
```

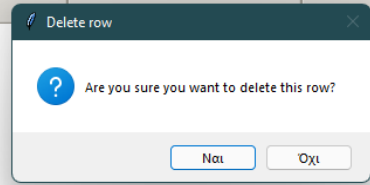
```

#Εισαγωγή δεδομένων με το πάτημα του Submit στο παράθυρο insert
def insert_data(new_win, ents, ont): #ont η οντότητα που μας αφορά
    #ents οι εισδοχοι στο παράθυρο εισαγωγής new_win
    try:
        data=() #Οι τιμές των entries από το παράθυρο επεξεργασίας
        q="" #Τα ερωτηματικά στο query για το insert
        for i in ents:
            temp=i.get()
            if temp=="":
                temp=None
            data+=(temp,)
            q+=", "
        q=q[:-2]+");"
        sql_query="INSERT INTO "+ont+" VALUES("+q
        c.execute(sql_query, data)
        con.commit()
        new_win.destroy()
        di=[]
        di['Maintenance']='Under_maint'
        di['Guided_Tour']='Is_Presented'
        di['Product']='Inspired_by'
        if ont in di: #Όταν δημιουργούμε μία από τις 3 οντότητες του λεξικού di, εμφανίζεται αυτόματα παράθυρο για τη δημιουργία εγγραφής στους αντίστοιχους πίνακες συσχέτισης
            newont=di[ont] #Τίτλος του πίνακα συσχέτισης
            new_ex(newont, data[0]) #Δημιουργία πίνακα συσχέτισης
    except sqlite3.IntegrityError as err:
        t=str(err)
        x=t.partition(".")
        messagebox.showerror('Error', ""+x[-1]+"" value missing/not acceptable") #Εμφάνιση errorbox με τη μη έγκυρη τιμή
        new_win.lift()

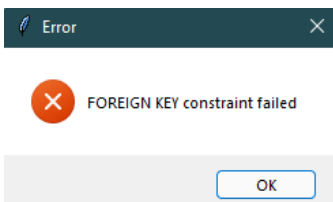
```

7.3.6 Delete

Title	Starting_date	Expected_return
Η συνανλία Των Αγγέλων	2021-12-31	2022-7-5
Φύλλο Με Δώδεκα Αρχαία Νομίσμα	2021-12-31	2022-8-25
Γυμνό	2021-12-31	2022-2-20
Ανατολίτης	2021-12-31	2022-9-21
Αραγμένα Καράβια	2021-12-31	2022-5-5
Αυτοπροσωπογραφία	2021-12-31	2022-9-7
Αυτοπροσωπογραφία	2021-12-31	2022-6-9
Κεφάλι Νέου	2021-12-31	2022-4-11



Error σε περίπτωση παραβίασης των κανόνων της βάσης



```
#Συνάρτηση διαγραφής (delete selection)
def del_sel():
    try:
        ont=cbox1.get() #Ανάκτηση οντότητας
        i = tree_view.selection()[0] #Ανάκτηση επιλεγμένου αντικειμένου
        sel = tree_view.item(i)['values'][0] #Η τιμή κλειδιού του επιλεγμένου αντικειμένου

        c.row_factory = lambda cursor, row: row[0] #STRs αντί για tuples
        c.execute("SELECT name FROM PRAGMA_table_info('"+ont+"')") #Attributes αντικειμένου
        data=c.fetchone() #Το όνομα του πρωτεύοντος κλειδιού (που είναι και το attribute)
        c.row_factory = None
        ans=askyesno(title='Delete row', message='Are you sure you want to delete this row?') #Προειδοποιητικό παράθυρο
        if ans: #Αν ο χρήστης αποδεχτεί την προειδοποίηση
            try:
                sql_query="DELETE FROM "+ont+" WHERE "+data+"=?" #query διαγραφής
                c.execute(sql_query, (sel,))
                con.commit()
            except sqlite3.IntegrityError as err: #Έλεγχος για περιορισμούς και εμφάνιση σχετικού μηνύματος
                t=str(err)
                messagebox.showerror('Error', t)
    except:pass
```

7.3.7 Edit προϊόντος

Barcode	Category	Title	Price
1	Εικόνες	Άγιος Πέτρος (El Greco) - Μικρογραφία	25.0
2	Εικόνες	Ελπίς	13.0
3	Αξεσουάρ	Μπρελόκ Άγιος Πέτρος	8.0
4	Φυλλάδιο	Έργα του Νικολάου Γύζη	0.0
5	Φυλλάδιο	Έργα του Γιάννη Τσαρούχη	0.0
6	Μινιατούρα	Επιτύμβια σύνθεση	10.0
7	Μινιατούρα	Μini κεραμικό αγγείο	15.0

Edit S...

Barcode: 1

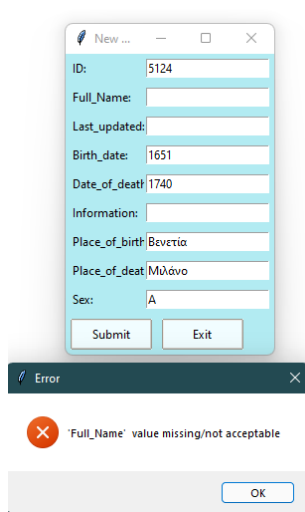
Category: Εικόνες

Title: Άγιος Πέτρος (El Greco)

Price: 25.0

Submit Exit

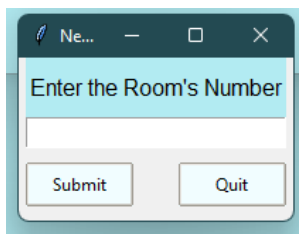
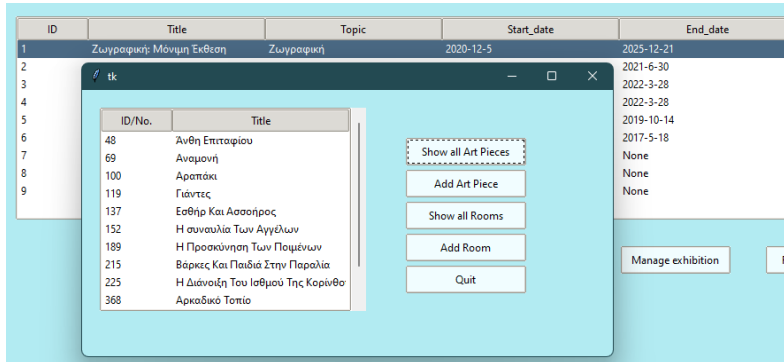
7.3.8 Error σε περίπτωση παραβίασης κανόνων της βάσης



```
#Συνάρτηση edit
def edit_sel():
    try:
        i = tree_view.selection()[0] #δείκτης αντικειμένου που έχει επιλεγεί από το treeview
        sel = tree_view.item(i)['values'] #Οι τιμές του επιλεγμένου αντικειμένου
        attr_list=[] #Λίστα attributes του αντικειμένου
        ont=cbox1.get() #Ανάκτηση οντότητας του αντικειμένου
        c.row_factory = lambda cursor, row: row[0] #STRs αντί για tuples
        c.execute("SELECT name FROM PRAGMA_table_info('"+ont+"')") #Ανάκτηση της λίστας attributes του αντικειμένου
        data=c.fetchall()
        for i in data:
            attr_list.append(i)
        c.row_factory = None
        edit_win=Tk() #Δημιουργία παραθύρου επεξεργασίας
        add_style(edit_win)
        edit_win.title('Edit Selection')
        ents=makeform(edit_win, attr_list,'edit', sel) #Δημιουργία entryboxes, labels στο παράθυρο επεξεργασίας
        #Κουμπιά αποθήκευσης και εξόδου
        sub_btn = ttk.Button(edit_win, text = 'Submit',command=(lambda:savechanges(edit_win, ents, attr_list, sel, ont)))
        sub_btn.pack(side = LEFT, padx = 5, pady = 5)
        quit_btn = ttk.Button(edit_win, text = 'Exit', command=edit_win.destroy)
        quit_btn.pack(side = LEFT, padx = 5, pady = 5)

        edit_win.mainloop()
    except: #Σε περίπτωση μη επιλογής αντικειμένου από το treeview
        pass
```

7.3.9 Manage exhibition «Ζωγραφική μόνιμη έκθεση» – Show all art pieces – Add room



```
#Συνάρτηση για προσθήκη στοιχείου σε μια έκθεση (αίθουσα ή έκθεμα) ή προσθήκη εγγραφής σε άλλο πίνακα συσχέτισης
def new_ex(str, id):
    new_ex_win=Tk() #δημιουργία νέου παραθύρου για εισαγωγή δεδομένων
    add_style(new_ex_win)
    new_ex_win.title('New data')
    ent_frame=Frame(new_ex_win)
    ent_frame.grid(row=0)

    if str=='Piece_of_Art': #Αν το παράθυρο εμφανίστηκε επειδή πατήσαμε Add Art Piece στο μενού διαχείρισης έκθεσης
        txt='Enter the title of the Art Piece'

    elif str=='Takes_Place': #Αν το παράθυρο εμφανίστηκε επειδή πατήσαμε Add Room στο μενού διαχείρισης έκθεσης
        txt='Enter the Room's Number'

    elif str=='Inspired_by': #Αν το παράθυρο εμφανίστηκε επειδή δημιουργήσαμε προϊόν
        txt='Enter the title of an art piece by which this product is inspired'

    elif str=='Under_maint': #Αν το παράθυρο εμφανίστηκε επειδή δημιουργήσαμε συντήρηση
        txt='Enter the ID of the art piece to be maintained'

    elif str=='Is_Presented': #Αν το παράθυρο εμφανίστηκε επειδή δημιουργήσαμε ξενάγηση
        txt='Enter the title of an exhibition this tour is going to visit'

    #Διαμόρφωση παραθύρου εισαγωγής δεδομένων (ID ή τίτλου)
    ap_label=Label(ent_frame, text=txt, font=('bold', 12), pady=10, background=bg_color)
    ap_label.pack()
    data_entry=Entry(ent_frame, width=20, font=12)
    data_entry.pack(padx=5)
    sb_btn=ttk.Button(ent_frame, text='Submit', width=10, command=lambda:addval(str, data_entry, id, new_ex_win))
    qt_btn=ttk.Button(ent_frame, text='Quit', command=new_ex_win.destroy, width=10)
    sb_btn.pack(side='left', pady=10, padx=5)
    qt_btn.pack(side='right', pady=10, padx=5)
```

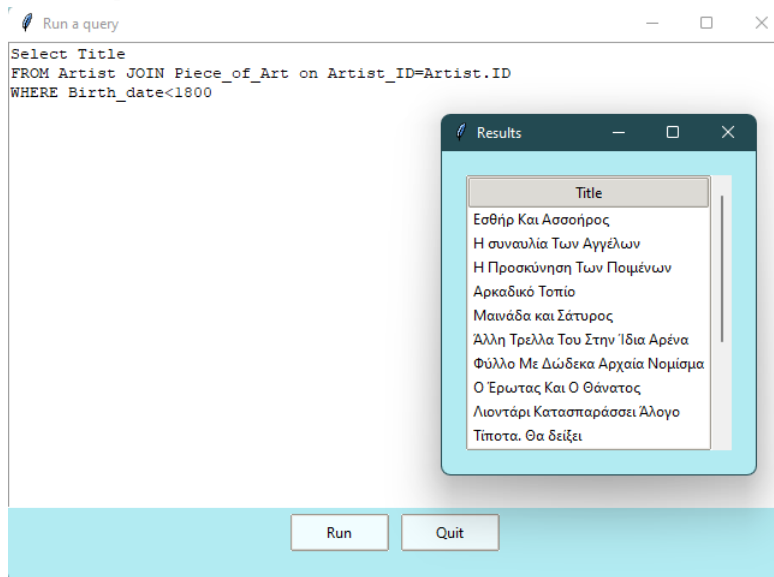
```

#Προσθήκη εγγραφής στους πίνακες συσχέτισης
def addval(str, data_entry, id, new_ex_win):
    key=data_entry.get() #Αν η τιμή που λάβαμε στο παράθυρο εισόδου δεν ήταν κενή
    if key!=None and key!='':
        if str=='Piece_of_Art' or str=='Inspired_by' or str=='Is_Presented':
            #Αν το str εισόδου είναι ένα από τα παραπάνω, τότε από την είσοδο που έχει δοθεί (η οποία είναι όνομα/τίτλος)
            #πρέπει να γίνει ανάκτηση ενός σχετικού ID για την δημιουργία της εγγραφής στον πίνακα που θέλουμε
            dict={'Piece_of_Art':'Piece_of_Art', 'Inspired_by':'Piece_of_Art', 'Is_Presented':'Exhibition'}
            try:
                c.execute("SELECT ID FROM "+dict[str]+" WHERE Title=?", (key,))
                data=c.fetchone()
                if str=='Inspired_by': #Στον πίνακα inspired_by η τιμή που εισάγουμε στο προηγούμενο παράθυρο τοποθετείται πρώτη
                    val=((data[0]),(id),)
                else:
                    val=((id),(data[0]),)
                sql_query="INSERT INTO "+str+" VALUES (?,?)"
                c.execute(sql_query, val)
                con.commit()
                if str=='Piece_of_Art': #Σε περίπτωση που προσθέτουμε πίνακα σε έκθεση, το παράθυρο καταστρέφεται
                    new_ex_win.destroy()
                else: #Σε περίπτωση που προσθέτουμε εγγραφή στους πίνακες Inspired_by, Is_Presented, το παράθυρο καθαρίζεται και μένει
                    #ανοιχτό για να μπορεί ο χρήστης να συμπληρώσει περισσότερες εγγραφές
                    messagebox.showinfo('Success', 'Insertion Successful') #Ειδοποίηση ότι ολοκληρώθηκε επιτυχώς η εγγραφή
                    data_entry.delete(0, END)
                    data_entry.insert(0,'')
                    new_ex_win.lift()
            except:
                messagebox.showerror('Error', "Invalid Title") #Errorbox σε περίπτωση εισόδου μη έγκυρου τίτλου
                new_ex_win.lift()

        elif str=='Takes_Place' or str=='Under_maint':
            #Αν το str εισόδου είναι ένα από τα παραπάνω, τότε η είσοδος που έχει δοθεί είναι το σχετικού ID που χρειαζόμαστε
            #για την δημιουργία της εγγραφής στον πίνακα που θέλουμε
            try:
                val=((id),(key),)
                sql_query="INSERT INTO "+str+" VALUES (?,?)"
                c.execute(sql_query, val)
                con.commit()
                data_entry.delete(0, END)
                data_entry.insert(0,'')
                new_ex_win.destroy()
            except:
                messagebox.showerror('Error', "Invalid Number") #Errorbox σε περίπτωση εισόδου μη έγκυρου ID
                new_ex_win.lift()
    else: pass

```

7.3.10 Run a query -> Run



```
#Τρέξιμο εντολής sql με το κουμπί run
def runsql(txt_edit):
    query = txt_edit.get("1.0", END) #Ανάκτηση της εισόδου από το πλαίσιο κειμένου και αποθήκευσή της σε str για εκτέλεση του query
    try:
        c.execute(query)
    except: #Σε περίπτωση κλεισίματος του βασικού παραθύρου ενώ είναι ακόμα ανοιχτό το παράθυρο 'Run query',
        #η σύνδεση με τη βάση χάνεται, οπότε εδώ την επαναφέρουμε
        con = sqlite3.connect('art-gallery.db')
        c = con.cursor()
        c.execute(query)
    data1 = c.fetchall()
    columns= [description[0] for description in c.description] #τίτλοι των στηλών των αποτελεσμάτων που πήραμε από το query
    results_win = Tk() #Παράθυρο αποτελεσμάτων
    add_style(results_win)
    results_win.title("Results")
    res_frame=Frame(results_win)
    res_frame.grid(row=4, column=0, columnspan=4, rowspan=6, pady=20, padx=20)
    tree_view = ttk.Treeview(res_frame, columns=columns, show="headings") #Treeview για την εκτύπωση των αποτελεσμάτων
    len_count=0 #Counter για την ανάγνωση του πρώτου tuple που επιστρέφει το query για ρύθμιση του πλάτους των στηλών
    for col in columns: #Ρύθμιση του πλάτους των στηλών
        wid=len(str(data1[0][len_count])) #Μήκος της πρώτης τιμής που επιστρέφεται για τη στήλη col
        if wid>4 or wid==0:
            tree_view.column(col, width=200)
        else:
            tree_view.column(col, width=80)
            tree_view.heading(col, text=col)
        len_count+=1
    tree_view.pack(side="left", fill="y")
    scrollbar = Scrollbar(res_frame, orient='vertical') #Δημιουργία scrollbar για το tree
    scrollbar.configure(command=tree_view.yview)
    scrollbar.pack(side="right", fill="y")
    tree_view.config(yscrollcommand=scrollbar.set)
    for row in data1:
        tree_view.insert('', 'end', values=row)
    results_win.mainloop()
```