

Βάσεις Δεδομένων

Εξαμηνιαία Εργασία

Ονόματα: Δραγώνας Σωτήρης 03114187, Ζαρόγιαννης Γιάννης 03114186

Α ερώτημα:

Διαλέξαμε την MySQL ως βάση δεδομένων και την JavaFX ως περιβάλλον ανάπτυξης της εφαρμογής γραφικού περιβάλλοντος.

Πλεονεκτήματα βάσης:

- Υπάρχει αρκετά καλό documentation στο διαδίκτυο σχετικά με την MySQL και κοινότητα για ερωτήσεις
- Το MySQL workbench είχε εργαλεία που σου επέτρεπαν την εύκολη κατασκευή της βάσης δεδομένων
- Ήταν εύκολη η εισαγωγή δεδομένων στη βάση εφόσον είχαμε στα χέρια μας κατάλληλα διαμορφωμένα .csv αρχεία
- Εύκολη διεπαφή με την Java

Μειονεκτήματα βάσης:

- Παρόλο που σε λανθασμένα queries, εμφανιζόταν error message, τις περισσότερες φορές δε βοηθούσε πολύ στην αναγνώριση του λάθους
- Κάποιες αλλαγές που κάναμε σε πεδία των πινάκων απαιτούσαν προκειμένου να λειτουργήσουν να αφαιρέσουμε τα αντίστοιχα foreign keys και στο τέλος να τα ξαναπροσθέσουμε, κάτι που ίσως θα μπορούσε να γίνεται αυτόματα
- Το γραφικό περιβάλλον δεν ήταν ιδιαίτερα σύγχρονο και χρηστικό (πολλές φορές απαιτούσε να πατάμε τα κουμπιά ανανέωσης με το χέρι προκειμένου να δούμε ορισμένες αλλαγές, επίσης είχε κακή αυτοσυμπλήρωση)
- Έλειπαν ορισμένοι τελεστές π.χ.: Intersect
- Δεν αναγνώριζε εκ των προτέρων πολλές φορές το λάθος syntax σε ένα query

Πλεονεκτήματα JavaFX:

- Εύκολη σχετικά δημιουργία του γραφικού περιβάλλοντος χρησιμοποιώντας το Scene Builder
- Καλή αντιστοιχία ανάμεσα σε τύπους δεδομένων SQL και Java
- Η αντικειμενοστρέφεια της γλώσσας βοήθησε αρκετά στην αναπαράσταση του γραφικού περιβάλλοντος
- Πολλά χρήσιμα APIs (π.χ.: για την διαχείριση ημερομηνιών, αριθμών, συμβολοσειρών κλπ)
- Πολύ εύκολη διαμόρφωση του γραφικού περιβάλλοντος στις ανάγκες μας με χρήση css
- Το IntelliJ IDEA βοήθησε πάρα πολύ την διαδικασία προγραμματισμού

Μειονεκτήματα JavaFX:

- Αρκετά επαναλαμβανόμενος κώδικας σε ορισμένες περιπτώσεις
- Τα connections και τα statements τα σχετικά με την βάση δεν έχουν αυτόματο τρόπο αποδέσμευσης

Έρώτημα:

Για την υλοποίηση της βάσης χρησιμοποιήσαμε το σχεσιακό μοντέλο που δίνεται στην ενδεικτική λύση που δόθηκε. Προσθέσαμε ωστόσο στον πίνακα των emails και των phone numbers δικό τους νέο id ως κύριο κλειδί προκειμένου να μπορεί κάθε κατάσταση να έχει πολλές εγγραφές σε αυτούς τους πίνακες δηλαδή πολλά emails και τηλέφωνα.

Κάθε γνώρισμα που είναι foreign key ή primary key έχει το δικό του ευρετήριο. Προσθέσαμε ένα ευρετήριο για το start_date των ενοικιάσεων για να μπορέσουμε να χρησιμοποιήσουμε το πεδίο ως foreign key σε άλλο πίνακα.

Για να εξασφαλίσουμε την ακεραιότητα οντότητας χρησιμοποιήσαμε auto increment σε όσους πίνακες είχαν id και στους υπόλοιπους τρέχαμε queries για να διαπιστώσουμε αν υπήρχαν ίδια primary keys. Για την εξασφάλιση αναφορικής ακεραιότητας χρησιμοποιήσαμε την επιλογή cascade on delete και cascade on update ώστε να μην μένουν «ξεκρέμαστες» εγγραφές. Επίσης όταν εισάγουμε νέες εγγραφές το κάνουμε με τη σωστή σειρά και σε όλους τους πίνακες που χρειάζεται και αναζητούμε με queries τις εξαρτήσεις τους σε άλλους πίνακες. Για την ακεραιότητα πεδίου τιμών έχουμε επιλέξει σωστούς τύπους σε κάθε περίπτωση και φροντίζουμε μέσω ελέγχων στη Java τα δεδομένα που εισάγονται στη βάση να έχουν τον κατάλληλο τύπο. Για την ακεραιότητα στηλών έχουμε θέσει ως unique τα πεδία που εξ ορισμού λαμβάνουν μοναδικές τιμές ανά εγγραφή και προτού εισάγουμε δεδομένα κάνουμε τους κατάλληλους ελέγχους στη Java. Έχουμε επιλέξει κατάλληλους περιορισμούς στα δεδομένα και έχουμε φτιάξει triggers τα οποία διασφαλίζουν ότι αυτοί τηρούνται.

Λεπτομέρειες της υλοποίησης υπάρχουν σε σχόλια στα δύο αρχεία .sql που δίνονται και στον κώδικα της εφαρμογής.