Σύστημα Διαχείρισης Αντιπροσωπείας Αυτοκινήτων

Των φοιτητών:

* Αθανάσιος Γεωργαλής 2021023
* Κωνσταντίνος Δεσποινίδης 2021035

1. **Πρόλογος**

Το Car Manager είναι μια desktop εφαρμογή, η οποία λειτουργεί ως πληροφοριακό σύστημα μηχανογράφησης αντιπροσωπείας αυτοκινήτων. Μέσω ενός εύχρηστου και απλού UI δίνει την δυνατότητα στον χρήστη να κατανοήσει το περιβάλλον της εφαρμογής ώστε να πλοηγηθεί εύκολα σε αυτήν και να αξιοποιήσει πολλές απο τις λειτουργίες της.

Πιο συγκεκριμένα, οι λειτουργίες που έχουν υλοποιηθεί στην εφαρμογή είναι οι εξής:

1. Προσθήκη/επεξεργασία/διαγραφή αυτοκινήτων
2. Προσθήκη/επεξεργασία/διαγραφή υπαλλήλων
3. Προσθήκη/επεξεργασία/διαγραφή πελατών
4. Προσθήκη/επεξεργασία/διαγραφή κατασκευαστών αυτοκινήτων
5. Προσθήκη/επεξεργασία/διαγραφή μοντέλων αυτοκινήτων
6. Προσθήκη/επεξεργασία/διαγραφή πωλήσεων
7. Εκτέλεση 10 διαφορετικών αναζητήσεων
8. **Ανάπτυξη και λειτουργίες εφαρμογής**

* **Εργαλεία και βιβλιοθήκες**

Τα βασικά εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι η γλώσσα προγραμματισμού **Java**, το **API Java Swing** , η βάση δεδομένων **Postgresql**, και το **IDE NetBeans**.

Χρησιμοποιήσαμε την βιβλιοθήκη [DataBridge](https://github.com/KDesp73/DataBridge) , δική μας κατασκευής, η οποία βοηθάει στην σύνδεση και επικοινωνία με την βάση, αλλά και στην κλήση αποθηκευμένων διεργασιών και συναρτήσεων. Ενώ υποστηρίζει αυτόματη φόρτωση του ResultSet σε λιστα αντικειμένων κάνοντας χρήση reflection, δεν χρησιμοποιήσαμε αυτή την λειτουργία, έτσι ωστε να γίνεται χειροκίνητη εξαγωγή των δεδομένων και δημιουργία αντικειμένων.

* **Δομή**

Η εφαρμογή αποτελείται από 15 παράθυρα στα οποία ο χρήστης μπορεί να πλοηγηθεί και να αλληλεπιδράσει. Πιο συγκεκριμένα:

1. παράθυρο πληροφοριών αυτοκινήτων
2. παράθυρο προσθήκης αυτοκινήτου
3. παράθυρο επεξεργασίας αυτοκινήτου
4. παράθυρο φύλλου καταγραφής αυτοκινήτων
5. παράθυρο πληροφοριών πελατών
6. παράθυρο προσθήκης πελάτη
7. παράθυρο επεξεργασίας πελάτη
8. παράθυρο πληροφοριών υπαλλήλων
9. παράθυρο προσθήκης υπαλλήλου
10. παράθυρο φύλλου καταγραφής υπαλλήλων
11. παράθυρο πληροφοριών πωλήσεων
12. παράθυρο προσθήκη νέας πώλησης
13. παράθυρο επεξεργασίας πώλησης
14. παράθυρο αναζητήσεων
15. παράθυρο σχετικά με την εφαρμογή και τους δημιουργούς

* **Ανάπτυξη**

Για την ανάπτυξη του GUI της εφαρμογής χρησιμοποιήθηκαν τα εξής components:

1. JFrame
2. JPanel
3. JLabel
4. JFormattedTextField
5. JButton
6. JList
7. JSpinner
8. JMenu
9. JMenuItem
10. JComboBox
11. JEditorPane
12. JScrollPane
13. JMenuBar
14. JTable
15. JTextField
16. JTextArea
17. JCheckBox

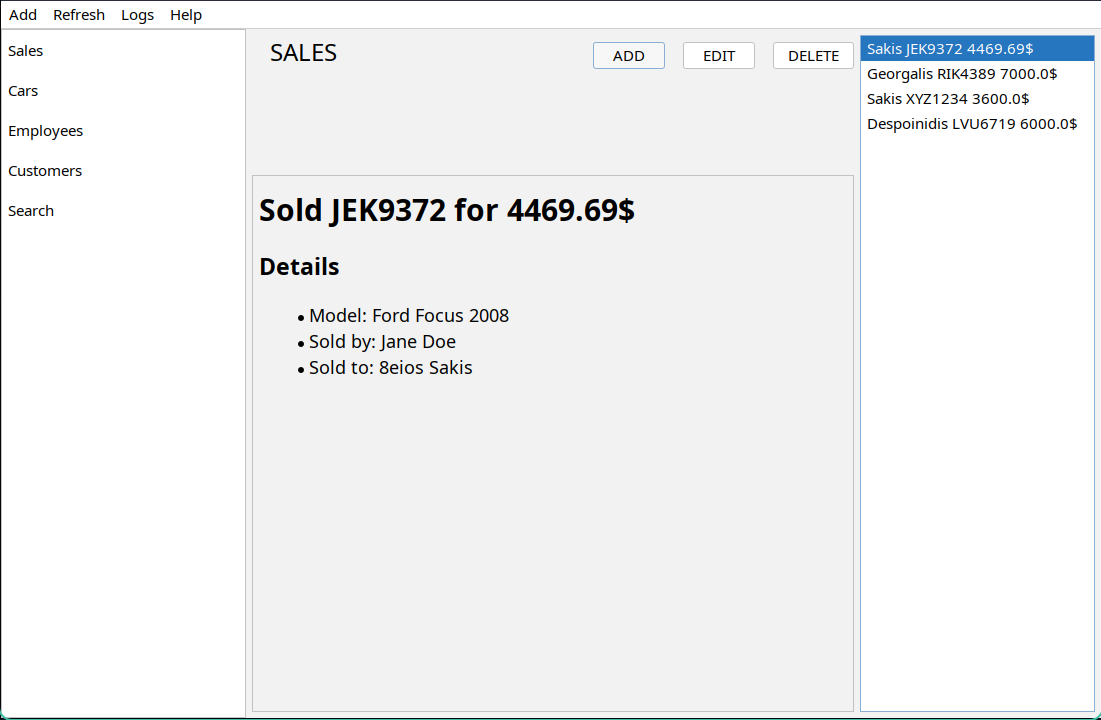
* **Panels και λειτουργικότητα**

Τα βασικά panels της εφαρμογής είναι:

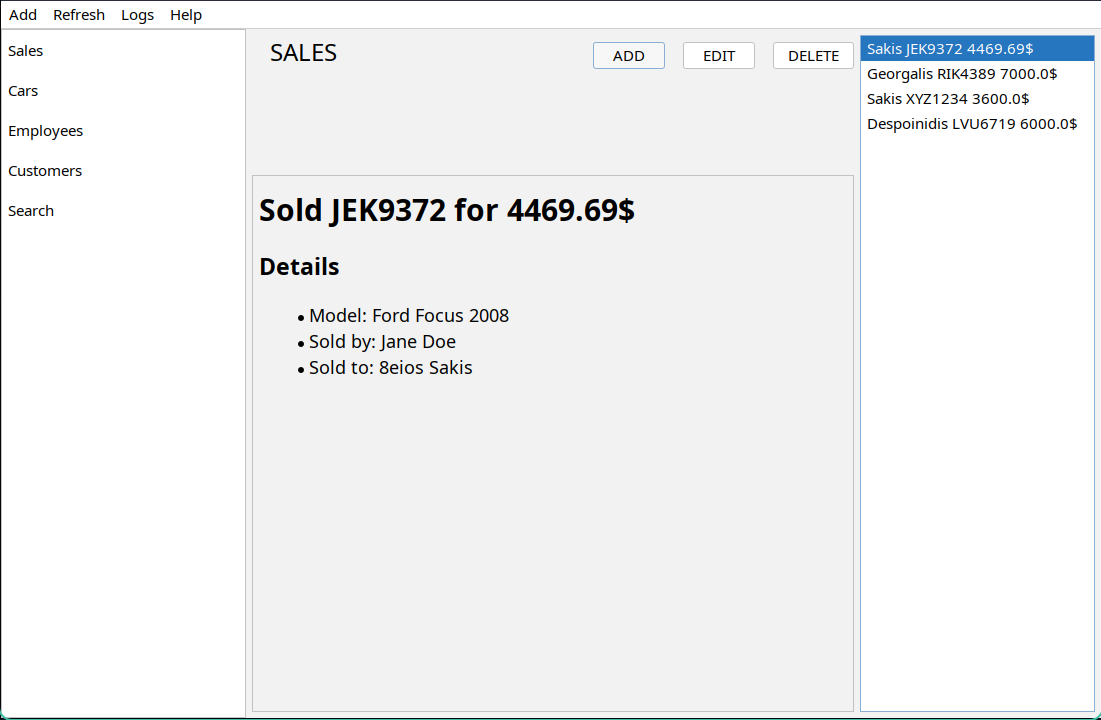
1. Sales
2. Cars
3. Employees
4. Customers
5. Search

στα οποία ο χρήστης μπορεί να πλοηγηθεί κάνοντας κλικ σε μια απο τις επιλογές στα αριστερά του παραθυρου

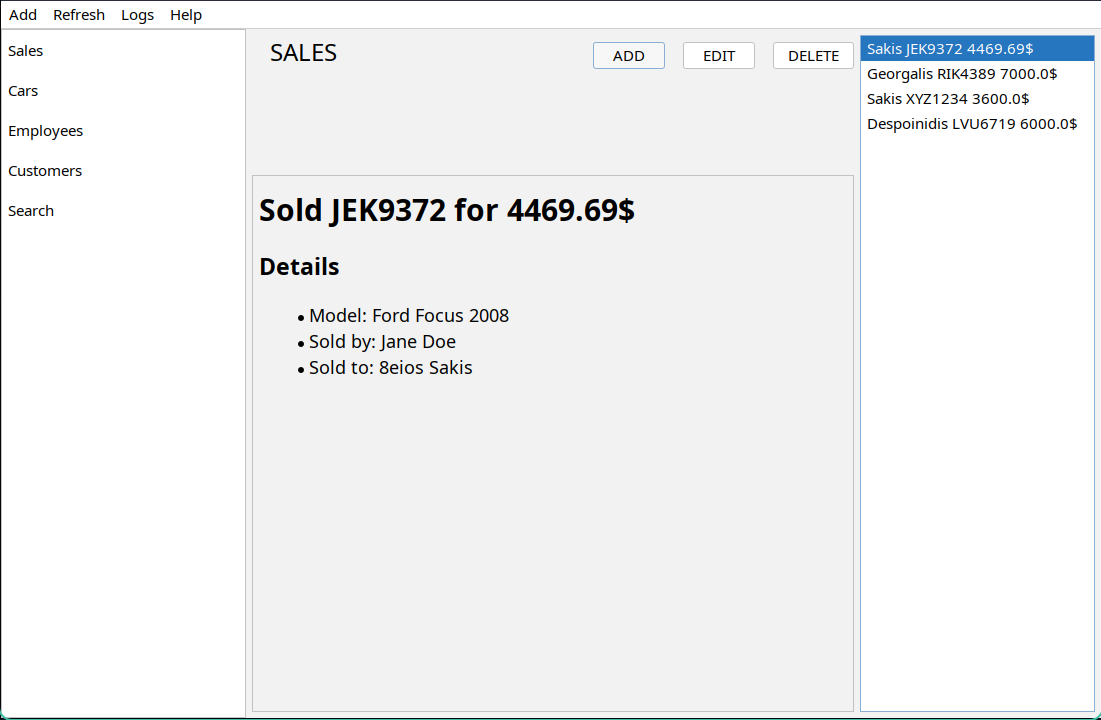






Αφού ο χρήστης επιλέξει μια απο τις παραπάνω επιλογές, στον διπλανό χώρο εμφανίζονται τα αντίστοιχα στοιχεία και components του κάθε panel

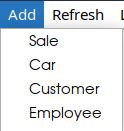
Στο επάνω μερός του παραθύρου υπάρχει μια μπάρα με 4 εργαλεία

1. Add
2. Refresh
3. Logs
4. Help

Καθένα απο τα εργαλεία αποτελεί ένα dropdown menu με επιλογές.

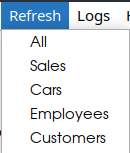
* **Add**

Προσθήκη νέας καταγραφής σε μια απο τις επιλογές sale, cars, employees και customers



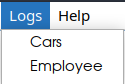
* **Refresh**

Ανανέωση είτε όλων των πληροφοριών είτε ξεχωριστά για καθεμιά απο τις επιλογες sales, cars, employees και customers



* **Logs**

Επιλογή προβολής πίνακα καταγραφών είτε των αυτοκινήτων είτε των υπαλλήλων



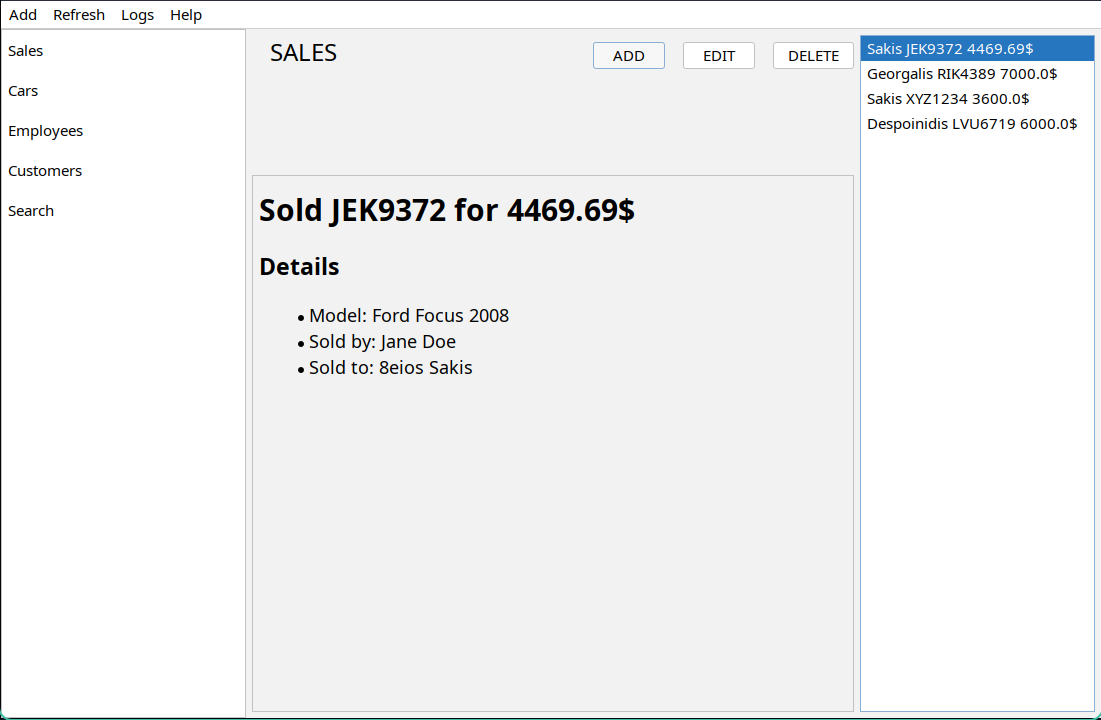
* **Help**

Επιλογές βοηθού της εφαρμογής και πληροφορίες σχετικά με την εφαρμογή και τους δημιουργούς της

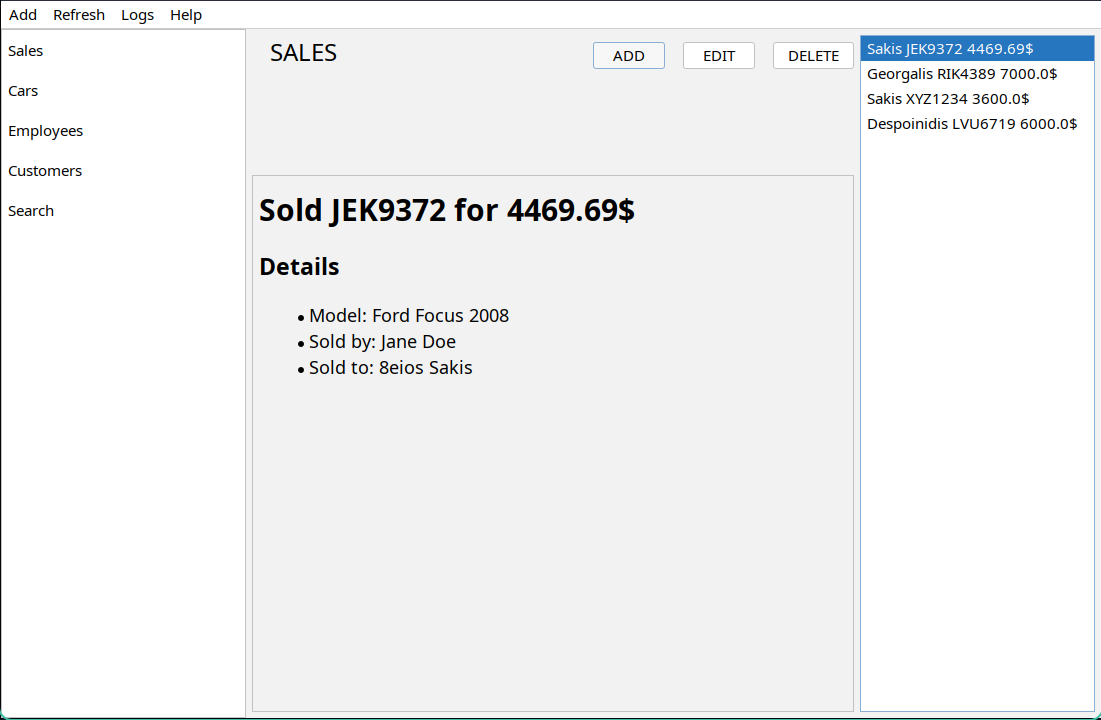


Κάθε panel εκτός του search, χωρίζεται σε 3 ίδια μέρη:

1. Πλάσιο ονομασίας panel και κουμπίων ενεργειών

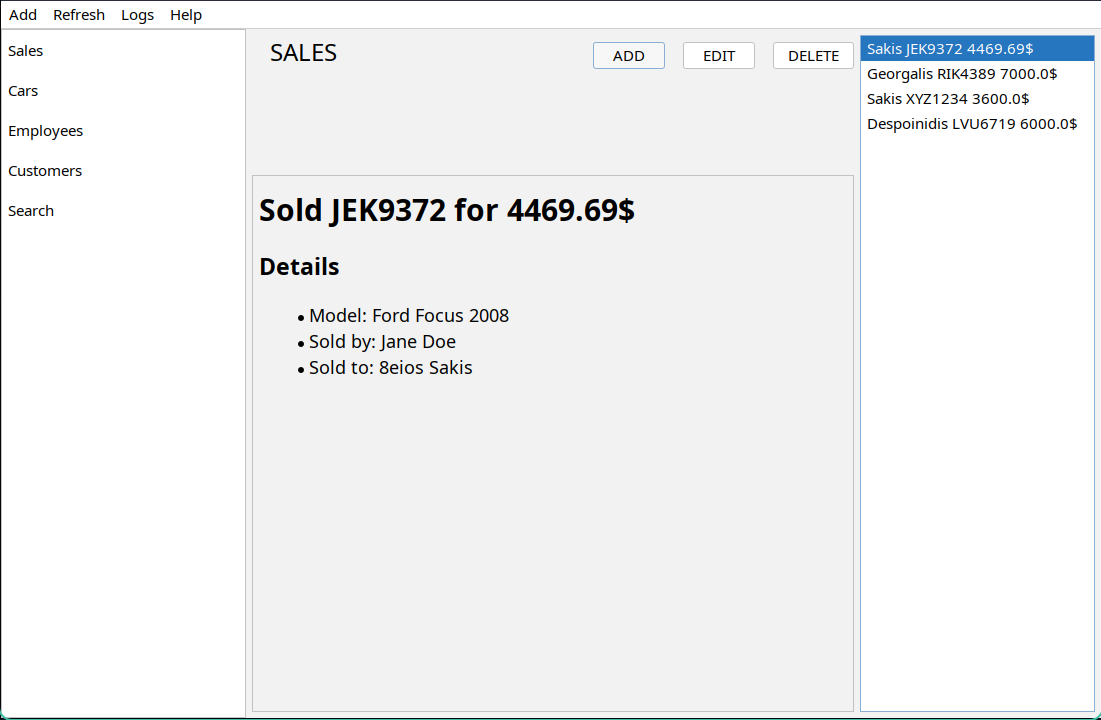


1. Πλαίσιο λίστας καταγραφών





1. Πλαίσιο πληροφοριών μιας επιλεγμένης καταγραφής



* **Panel πωλήσεων**

Στο panel πωλήσεων ο χρήστης μπορεί να βρεί όλες τις πωλήσεις αυτοκινήτων που γίνανε για όλους τους πελάτες.

Κάνοντας κλικ σε μια καταγραφή απο την λίστα με τις πωλήσεις, εμφανίζονται σε ένα πλαίσιο οι παρακάτω πληροφορίες της συγκεκριμένης πώλησης:

- Πινακίδα αυτοκινήτου που πουλήθηκε

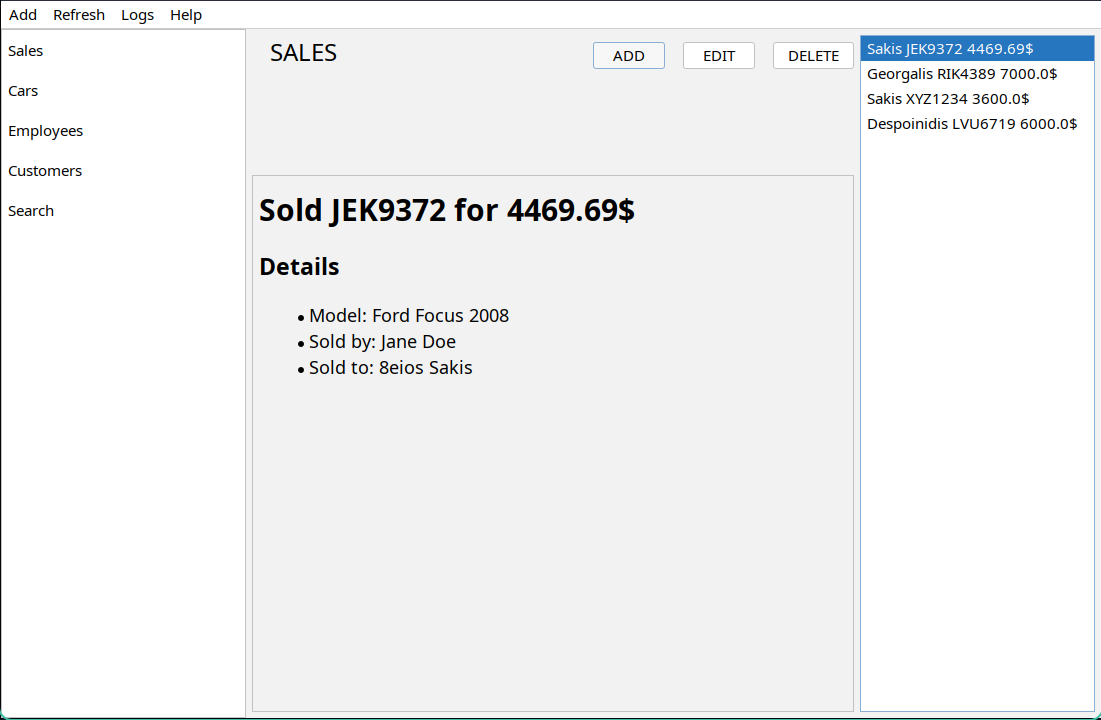
- Τιμή πώλησης

- Μοντέλου αυτοκινήτου

- Ονοματεπώνυμο υπαλλήλου που πραγματοποίησε την πώληση

- Ονοματεπώνυμο πελάτη που αγόρασε το αυτοκίνητο

Επιπρόσθετα, ο χρήστης μπορει μέσω τριών κουμπιών να πραγματοποιήσει νέα πώληση ενός απο τα εναπομείναντα αυτοκίνητα τα οποία έχουν περάσει το απαιτούμενο service και να επεξεργαστεί ή να διαγράψει μια ήδη υπάρχουσα πώληση.



* **Panel υπαλλήλων**

Στο panel υπαλλήλων ο χρήστης μπορεί να βρεί όλους τους υπαλλήλους της αντιπροσωπείας.

Κάνοντας κλικ σε μια καταγραφή απο την λίστα με τους υπαλλήλους, εμφανίζονται σε ένα πλαίσιο οι παρακάτω πληροφορίες του συγκεκριμένου υπαλλήλου:

- Όνομα

- Επίθετο

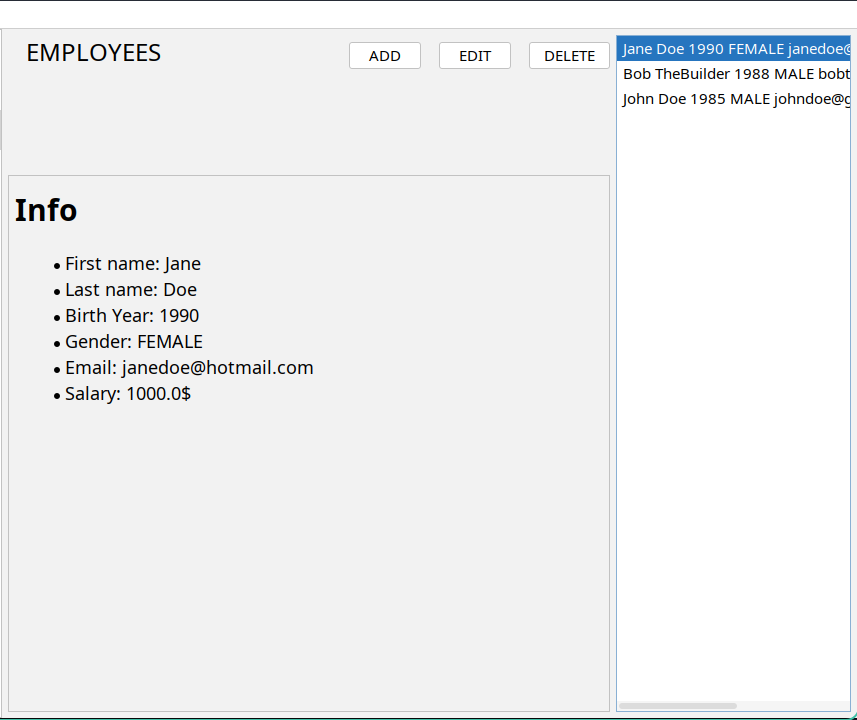
- Ημερομηνία γέννησης

- Φύλο

- Email επικοινωνίας

- Μισθός

Επιπρόσθετα, ο χρήστης μπορει μέσω τριών κουμπιών να προσθέσει νέο πελάτη και να επεξεργαστεί ή να διαγράψει έναν ήδη υπάρχων υπάλληλο.



* **Panel πελατών**

Στο panel πελατών ο χρήστης μπορεί να βρεί όλους τους καταγεγραμμένους πελάτες.

Κάνοντας κλικ σε μια καταγραφή απο την λίστα με τους πελάτες, εμφανίζονται σε ένα πλαίσιο οι παρακάτω πληροφορίες του συγκεκριμένου πελάτη:

- Όνομα

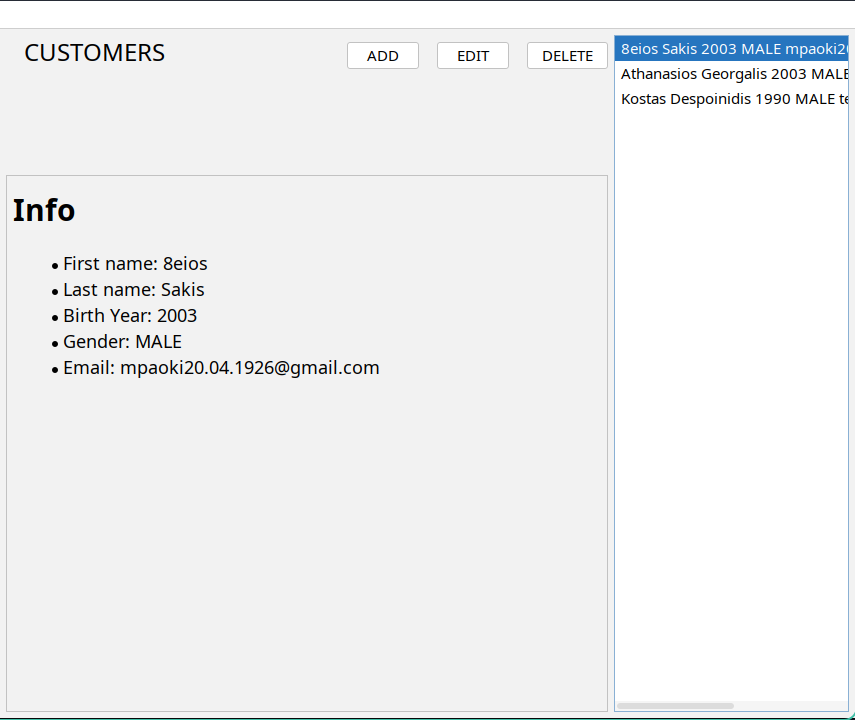
- Επίθετο

- Ημερομηνία γέννησης

- Φύλο

- Email επικοινωνίας

Επιπρόσθετα, ο χρήστης μπορει μέσω τριών κουμπιών να προσθέσει νέο πελάτη και να επεξεργαστεί ή να διαγράψει έναν ήδη υπάρχων πελάτη.



* **Panel αυτοκινήτων**

Στο panel αυτοκινήτων ο χρήστης μπορεί να βρεί όλα τα καταγεγραμμένα αυτοκίνητα.

Κάνοντας κλικ σε μια καταγραφή απο την λίστα με τους αυτοκίνητα, εμφανίζονται σε ένα πλαίσιο οι παρακάτω πληροφορίες του συγκεκριμένου αυτοκινήτου:

- Αριθμός πινακίδας

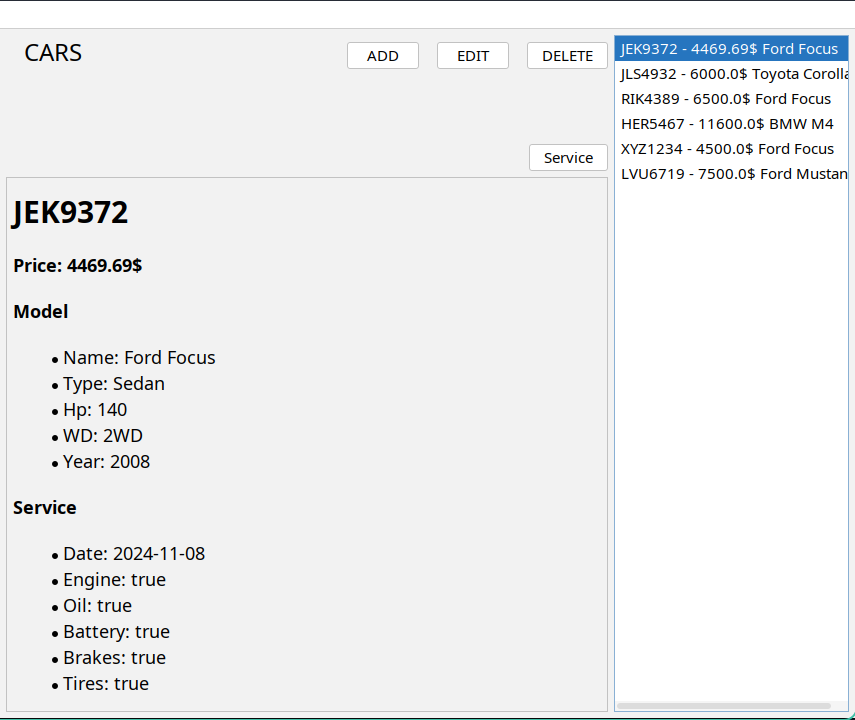
- Τιμή

- Πληροφορίες μοντέλου:

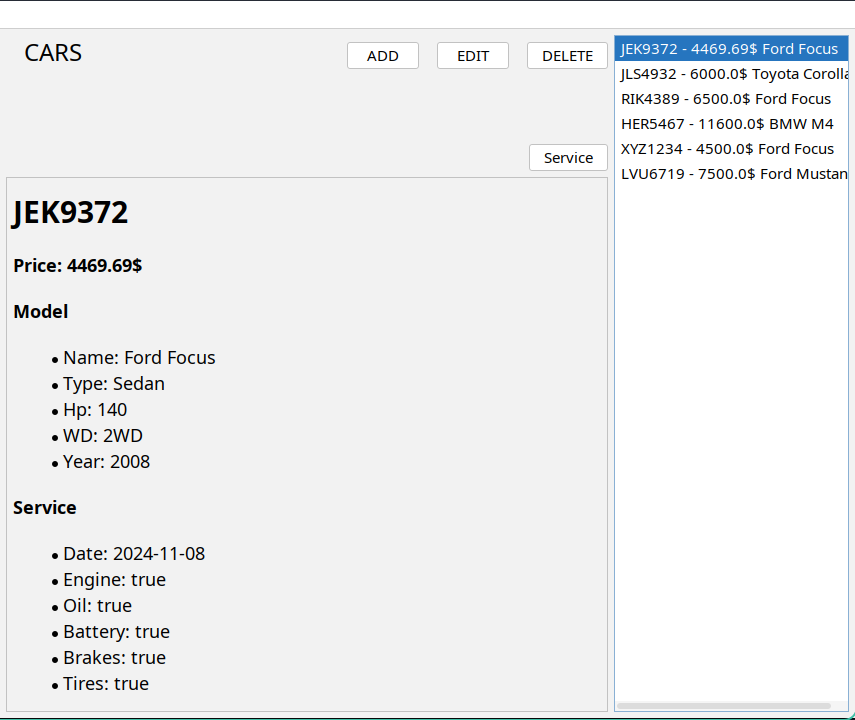
1. Όνομα
2. Τύπος
3. Ίπποι
4. Wheel Drive
5. Έτος κυκλοφορίας

- Service:

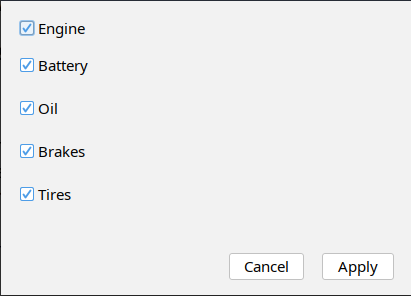
1. Ημερομηνία
2. Κατάσταση μηχανής
3. Κατάσταση λαδιών
4. Κατάσταση φρένων
5. Κατάσταση λάστιχων

Επιπρόσθετα, ο χρήστης μπορει μέσω τριών κουμπιών να προσθέσει νέο πελάτη και να επεξεργαστεί ή να διαγράψει έναν ήδη υπάρχων πελάτη.

Είναι σημαντικό να σημειωθεί πως στο συγκεκριμένο panel υπάρχει και ένα τέταρτο κουμπί το οποίο ονομάζεται Service και ανακατευθύνει τον χρήστη σε ένα νέο παράθυρο στο οποίο φαίνονται οι καταστάσεις του service του επιλεγμένου αυτοκινήτου.



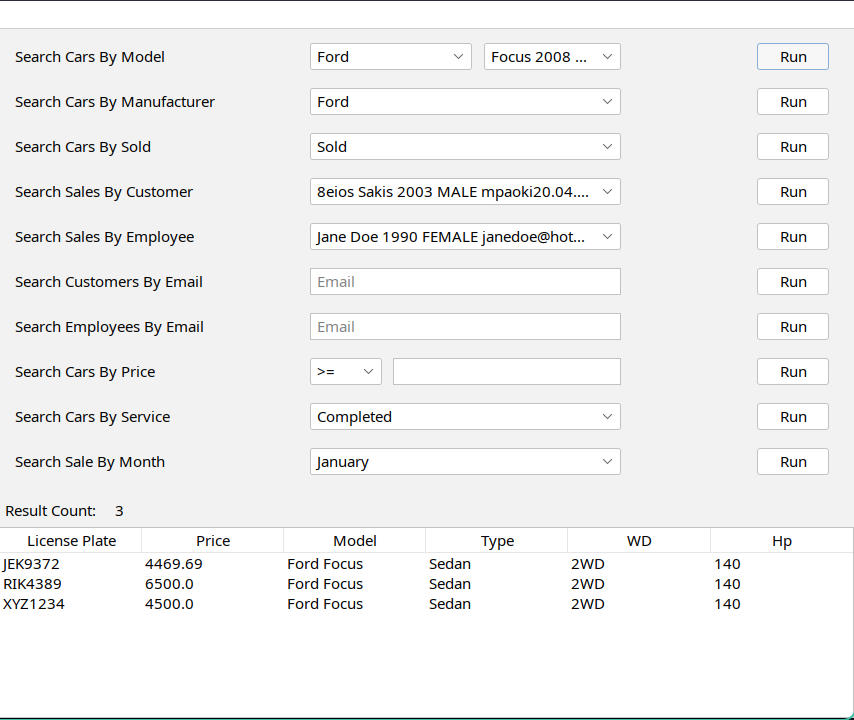




Παράθυρο Service

* **Panel αναζήτησης**

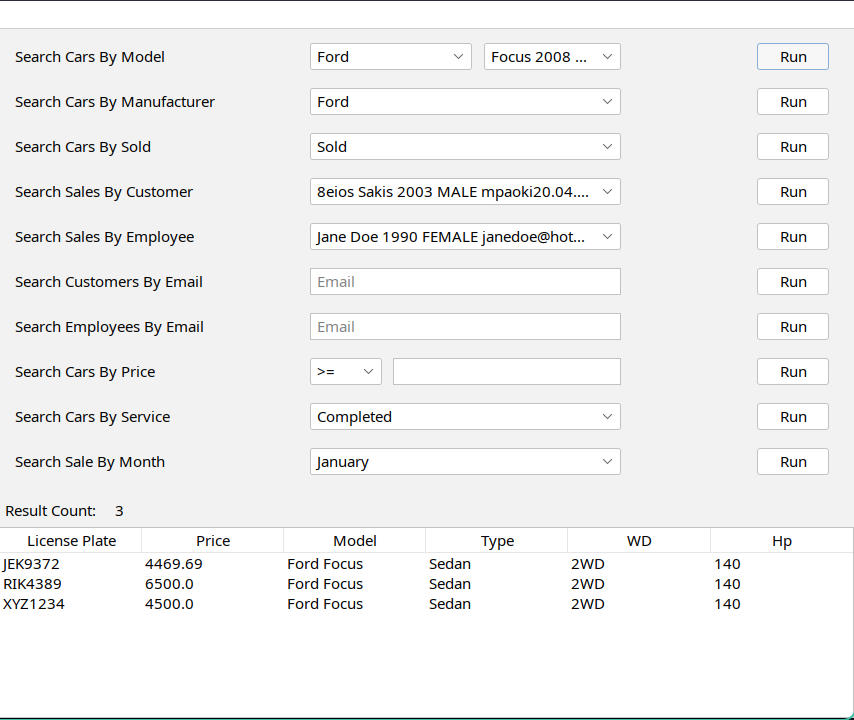
Στο panel αναζητήσεων υπάρχουν 10 διαφορετικές αναζητήσεις που μπορεί να εκτελέσει ο χρήστης προκειμένου να λάβει τις ανάλογες πληροφορίες.





Κάθε αναζήτηση αποτελείται απο 3 μέρη:

1. τον τίτλο της αναζήτησης
2. τα φίλτρα
3. το κουμπί της εκτέλεσης της αναζήτησης





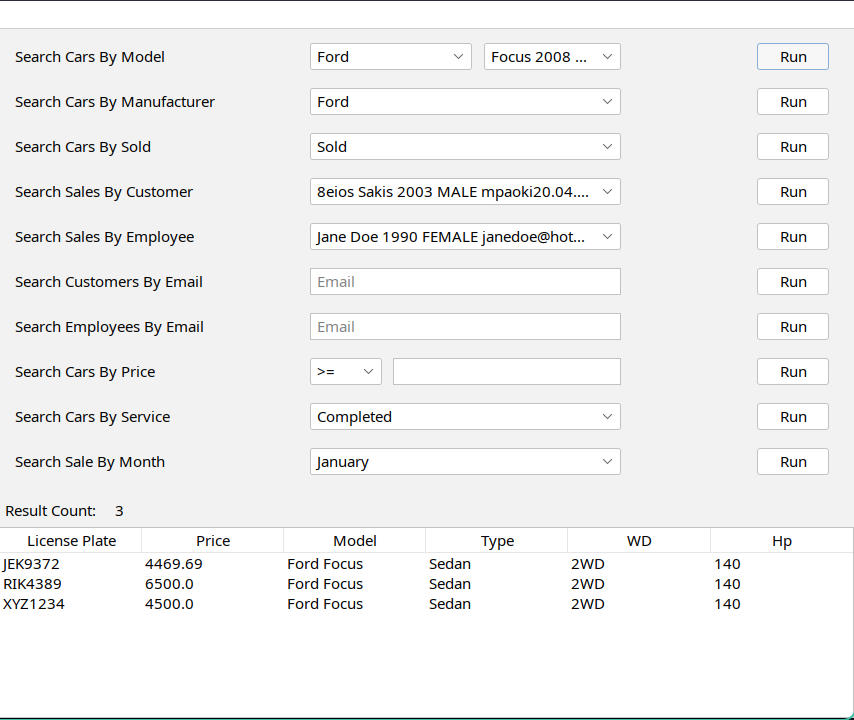
κουμπί εκτέλεσης

φίλτρα

τίτλος αναζήτησης

Στο κάτω μέρος του παραθύρου υπάρχει ένας μετρητής καταγραφών, όπου μετράει τις γραμμές του πίνακα και ένας πίνακας ο οποίος γεμίζει με τα δεδομένα του αποτελέσματος μιας αναζήτησης. Πιο συγκεκριμένα, βάζοντας τα κατάλληλα φίλτρα στην επιθυμητή αναζήτηση και κάνοντας κλικ στο κουμπί Run, ο χρήστης μπορεί να δεί τα αποτελέσματα της αναζήτησής του στον εν λόγω πίνακα.

Κάθε φορά που εκτελείται μια αναζήτηση, για να γεμίσει ο πίνακας με δεδομένα, επικοινωνεί η βάση δεδομένων με την εφαρμογή, εκτελείται το αντίστοιχο query μέσω functions και procedures της Postgres και επιστρέφεται το αποτέλεσμα στον πίνακα, όπου και πλέον είναι διαθέσιμο στον χρήστη.



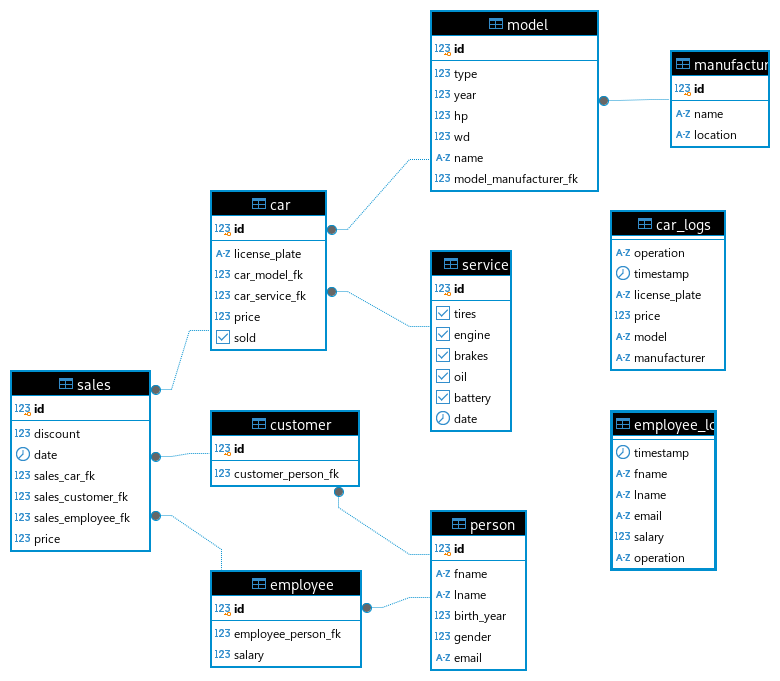


**ΣΗΜΑΝΤΙΚΗ ΣΗΜΕΙΩΣΗ:**

Μετά απο κάθε ενέργεια που εκτελείται στην εφαρμογή και σχετίζεται με την βάση δεδομένων, όπως λήψη, επεξεργασία, διαγραφή και νέα καταγραφή δεδομένων, η βάση δεδομένων Postgres επικοινωνεί με την εφαρμογή και έπειτα εκτελούνται οι ανάλογες ενέργειες.

1. **Δομή Βάσης Δεδομένων**
   1. **Πινακες**
      1. **Διάγραμμα ER**

Παρακάτω παρουσιάζεται το διάγραμμα ER της βάσης δεδομένων



* + 1. **Ανάλυση Πινάκων**

**Person**

Ο πίνακας Person περιέχει βασικές πληροφορίες οποιουδήποτε ανθρώπου συμπεριλαμβάνεται στο σύστημα διαχειρισης. Περιέχει πληροφορίες, χρήσιμες στους πίνακες Employee και Customer. Οι στήλες του πίνακα είναι οι εξής:

* Όνομα (fname)
* Επίθετο (lname)
* Έτος Γέννησης (birth\_year)
* Φύλο (gender)
* Email (email)
* ID (id)

**Employee**

Ο πίνακας Employee, κάνοντας χρήση ξένου κλειδιού αποθηκεύει τις γενικές πληροφορίες του υπαλλήλου στον πίνακα Person και περιέχει μόνο τα ειδικά για την δουλειά, στοιχεία. Οι στήλες του πίνακα είναι οι εξής:

* Μισθός (salary)
* Ξένο κλειδί προς τον πίνακα Person (employee\_person\_fk)
* ID (id)

**Customer**

Όπως ο πίνακας Employee έτσι και ο Customer περιέχει ξένο κλειδί που δείχνει στον πίνακα Person με σκοπό ο ίδιος να περιέχει μόνο τις ειδικές πληροφορίες ενός πελάτη. Οι στήλες του πίνακα είναι οι εξής:

* Ξένο κλειδί προς τον πίνακα Person (customer\_person\_fk)
* ID (id)

Στην προκειμένη περίπτωση δεν υπάρχουν επιπλέον ειδικές πληροφορίες, αλλά είναι εύκολο να προστεθούν εαν είναι αναγκαίο.

Τέλος, στην περίπτωση που ένας υπάλληλος της αντιπροσωπείας αποφασίσει να αγοράσει αυτοκίνητο από την αντιπροσωπεία, προστείθεται στον πίνακα Customer αλλά οι πληροφορίες του ήδη υπάρχουν στον πίνακα Person και έτσι δεν χρειάζεται να επαναληφθούν.

**Manufacturer**

O πίνακας Manufacturer περιέχει πληροφορίες για τους κατασκευαστές αυτοκινήτων που υπάρχουν στην αγορά. Οι στήλες του πίνακα είναι οι εξής:

* Όνομα (name)
* Τοποθεσία (location)
* ID (id)

Για παράδειγμα Ford, USA ή Fiat, Italy

**Model**

Ο πίνακας Model αναφέρεται σε συγκεκριμένα μοντέλα αυτοκινήτων και τις σχετικές πληροφορίες τους. Οι στήλες του πίνακα είναι οι εξής:

* Όνομα (name)
* Τύπος (type)
* Χρονιά Κυκλοφορίας (year)
* Ιπποδύναμη (hp)
* Κίνηση τροχών (wd)
* Ξένο Κλειδί προς τον πίνακα Manufacturer (model\_manufacturer\_fk)
* ID (id)

**Car**

Ο πίνακας Car περιέχει καταγραφές που αναφέρονται σε πραγματικά αυτοκίνητα που βρίσκονται στην αντιπροσωπεία ή έχουν πωληθεί απο αυτή. Οι στήλες του πίνακα είναι οι εξής:

* Πινακίδα (license\_plate)
* Τιμή (price)
* Έχει πραγματοποιηθεί πώληση (sold)
* Ξένο κλειδί προς τον πίνακα Model (car\_model\_fk)
* Ξένο κλειδί προς τον πίνακα Service (car\_service\_fk)
* ID (id)

**Service**

Ο πίνακας Service περιέχει καταγραφές, μία για κάθε αυτοκίνητο, με πληροφορίες για τις επισκευές που έχουν γίνει στο κάθε αυτοκίνητο. Οι στήλες του πίνακα είναι οι εξής:

* Λάστιχα (tires)
* Μηχανή (engine)
* Φρένα (brakes)
* Λάδια (oil)
* Μπαταρία (battery)
* Ημερομηνία πλήρους ολοκλήρωσης (date)
* ID (id)

To Service θεωρείται ολοκληρωμένο όταν όλα τα boolean πεδία είναι true.

**Sales**

O πίνακας Sales περιέχει κάθε πώληση που έχει διεξαχθεί απο την αντιπροσωπεία. Οι στήλες του πίνακα είναι οι εξής:

* Τελική τιμή (price). Υπολογίζεται απο την τιμή πώλησης του αυτοκινήτου και την πιθανή έκπτωση
* Έκπτωση (discount)
* Ξένο κλειδί προς τον πίνακα Car (sales\_car\_fk)
* Ξένο κλειδί προς τον πίνακα Customer (sales\_customer\_fk)
* Ξένο κλειδί προς τον πίνακα Employee (sales\_employee\_fk)
* Ημερομηνία πώλησης (date)
* ID (id)

**Car Logs**

Ο πίνακας car\_logs κάνοντας χρήση των triggers αποθηκεύει οποιαδήποτε ενέργεια πραγματοποιηθεί στον πίνακα Car συμπεριλαμβανομένου των πληροφοριών της σχετικής καταγραφής. Οι στήλες του πίνακα είναι οι εξής:

* Ενέργεια (operation)
* Πινακίδα (license\_plate)
* Τιμή (price)
* Μοντέλο (model)
* Κατασκευαστής (manufacturer)
* Χρονική στιγμή (timestamp)

**Employee Logs**

Ο πίνακας employee\_logs κάνοντας χρήση των triggers αποθηκεύει οποιαδήποτε ενέργεια πραγματοποιηθεί στον πίνακα Employee συμπεριλαμβανομένου των πληροφοριών της σχετικής καταγραφής. Οι στήλες του πίνακα είναι οι εξής:

* Ενέργεια (operation)
* Όνομα (fname)
* Επίθετο (lname)
* Email (email)
* Μισθός (salary)
* Χρονική στιγμή (timestamp)
  1. **Αποθηκευμένες Διεργασίες**

Κατα την διάρκεια της ανάπτυξης του λογισμικού, έγινε χρήση αποθηκευμένων διεργασιών και συναρτήσεων. H δομή των ονομάτων αλλά και των παραμέτρων αυτών των διεργασιών κρατήθηκε σταθερή για λόγους ευχρηστίας και προβλεψιμότητας. Σε κάθε πίνακα ανήκουν οι διεργασίες

* + insert\_<table>(insert\_params…)
  + update\_<table>(id, insert\_params…)
  + delete\_<table>(id | <unique-field>)

Για παράδειγμα για τον πίνακα car εχουμε τις διεργασίες:

* insert\_car(varchar, float4, int4)
* update\_car(int4, varchar, float4, int4)
* delete\_car(varchar)

όπου στην περίπτωση της delete αντι για id, η διεργασία δέχεται τις πινακίδες του αυτοκινήτου.

Επίσης για κάθε πίνακα υπάρχουν συναρτήσεις ανάκτησης δεδομένων σε μορφή πινάκων της μορφής

select\_all\_<table>() ή select\_<table>s\_by\_<field>(param\_<field>).

Η χρήση αποθηκευμένων διεργασιών έχει ένα πλήθος πλεονεκτημάτων. Κύρια πλεονεκτήματα είναι:

* η ενθυλάκωση της δομής της βάσης και των διαδικασιών που προκύπτουν κατα την εισαγωγη, επεξεργασία, διαγραφή και ανάκτηση δεδομένων απο τον client
* η βελτίωση της απόδοσης αφού διεργασίες που απαιτούν μεγάλη επεξεργαστική ισχύ εκτελούνται απο τον server.
* η μείωση της πιθανότητας για SQL Injection
* η ανεξαρτητοποίηση των ερωτημάτων απο την εφαρμογή

**Car**

*Διεργασίες*

* insert\_car(varchar, float4, int4) Κάνει εισαγωγή αυτοκινήτου στη βάση. Οι παράμετροι της διεργασίας είναι οι:
* Πινακίδα
* Τιμή
* Ξένο κλειδί προς τον πίνακα model
* update\_car(int4, varchar, float4, int4)

Αναλαμβάνει την ανανέωση τιμών ενός αυτοκινήτου. Οι παράμετροι της είναι ίδιες με αυτές της insert\_car με μόνη διαφορά την προσθήκη του id

* delete\_car(varchar)

Διαγράφει το αυτοκίνητο που αντιστοιχεί στην παράμετρο πινακίδα

*Συναρτήσεις*

* select\_all\_cars()
* select\_cars\_by\_model(varchar)
* select\_cars\_by\_manufacturer(varchar)
* select\_cars\_by\_sold(bool)

Η συνάρτηση αυτή επιστρέφει τα αυτοκίνητα που έχουν πωληθεί εάν η παράμετρος έχει την τιμή true και αντίστοιχα επιστρέφει τα αυτοκίνητα που δεν έχουν πωληθεί εάν η παράμετρος έχει την τιμή false

* select\_cars\_by\_price\_greater(float4)
* select\_cars\_by\_price\_less(float4)
* select\_cars\_by\_price(float4)

Οι παραπάνω συναρτήσεις επιστρέφουν αυτοκίνητα βάσει της τιμής τους. Η πρώτη επιστρέφει αυτοκίνητα που έχουν πωληθεί με τιμή μεγαλύτερη ή ίση της παραμέτρου. Η δεύτερη επιστρέφει αυτά που έχουν πωληθεί με τιμή μικρότερη ή ίση της παραμέτρου και τέλος η τρίτη επιστρέφει αυτά που πωλήθηκαν ακριβώς στην τιμή της παραμέτρου

* select\_cars\_by\_service\_completed()

Η παραπάνω συνάρτηση επιστρέφει τα αυτοκίνητα τα οποία έχουν ολοκληρώσει όλους τους απαραίτητους ελέγχους, δηλαδή εάν όλα τα boolean πεδία της καταγραφής service που αντιστοιχεί στο κάθε αυτοκίνητο είναι true

* select\_cars\_by\_service\_not\_completed()

Η συνάρτηση αυτή επιστρέφει τα αυτοκίνητα τα οποία δεν έχουν ολοκληρώσει όλους τους απαραίτητους ελέγχους, δηλαδή εάν έστω και ένα απο τα boolean πεδία της καταγραφής service που αντιστοιχεί στο κάθε αυτοκίνητο είναι false

*Εναύσματα*

* car\_logs\_trigger

Το έναυσμα car\_logs\_trigger ενεργοποιείται σε οποιαδήποτε ενέργεια πραγματοποιηθεί στον πίνακα car (INSERT, UPDATE, DELETE). Έπειτα καλεί την διεργασία create\_car\_log() η οποία εισάγει τις πληροφορίες της ενέργειας στον πίνακα car\_logs

**Model**

*Διεργασίες*

* insert\_model(varchar, int4, int4, int4, int4, int4)
* update\_model(int4, varchar, int4, int4, int4, int4, int4)
* delete\_model(int4)
* populate\_model()

Η διεργασία populate\_model εισάγει ένα πλήθος μοντέλων για ένα μέρος των κατασκευαστών που βρίσκονται στη βάση. Ιδανικά, κάνοντας χρήση API Endopoint θα εισαγάγαμε όλα τα μοντέλα για κάθε κατασκευστή.

*Συναρήσεις*

* select\_all\_models()
* select\_models\_by\_manufacturer(varchar)

Η συνάρτηση αυτή επιστρέφει όλα τα αποθηκευμένα μοντλελα για τον κατασκευαστή της παραμέτρου. Για παράδειγμα εάν η παράμετρος είναι το string “Ford”, Θα επιστραφούν τα μοντέλα “Focus”, “Fiesta”, κλπ.

**Manufacturer**

*Διεργασίες*

* insert\_manufacturer(varchar, varchar)
* update\_manufacturer(int4, varchar, varchar)
* delete\_manufacturer(varchar)
* populate\_manufacturer()

Η διεργασία populate\_manufacturer εισάγει ένα ικανοποιητικό πλήθος κατασκευαστών αυτοκινήτων που αποτελούν ένα μεγάλο ποσοστό της αγοράς

*Συνρτήσεις*

* select\_all\_manufacturers()
* select\_manufacturers\_by\_location(varchar)

Επιστρέφει τους κατασκευαστές βάσει της τοποθεσίας τους. Για παράδειγμα για την παράμετρο “Italy”, θα επιστρέψει τους κατασκευαστές “Fiat”, “Ferrari”, “Lamborghini” και “Alpha Romeo”

**Service**

*Διεργασίες*

Απουσιάζουν οι διεργασίες insert\_service και delete\_service αφου κάθε service, δημιουργείται και διαγράφεται μαζί με το αυτοκίνητο που αντιστοιχίζεται

* update\_service(int4, bool, bool, bool, bool, bool)

*Συναρτήσεις*

Δεν χρειάζονται συναρτήσεις ανάκτησης δεδομένων αφου μαζί με το αυτοκίνητο γίνεται και η ανάκτηση του service που του αντιστοιχεί

**Employee**

*Διεργασίες*

* insert\_employee(float, varchar, varchar, int4, int4, varchar)
* update\_employee(int4, float, varchar, varchar, int4, int4, varchar)
* delete\_employee(int4)

*Συναρτήσεις*

* select\_all\_employees()
* select\_employees\_by\_email(varchar)

*Εναύσματα*

* employee\_logs\_trigger

Το έναυσμα employee\_logs\_trigger ενεργοποιείται σε οποιαδήποτε ενέργεια πραγματοποιηθεί στον πίνακα employee (INSERT, UPDATE, DELETE). Έπειτα καλεί την διεργασία create\_employee\_log() η οποία εισάγει τις πληροφορίες της ενέργειας στον πίνακα employee\_logs

**Customer**

*Διεργασίες*

* insert\_customer(varchar, varchar, int4, int4, varchar)
* update\_customer(int4, varchar, varchar, int4, int4, varchar)
* delete\_customer(int4)

*Συναρτήσεις*

* select\_all\_customers()
* select\_customers\_by\_email(varchar)

**Sales**

*Διεργασίες*

* insert\_sale(float4, float4, int4, int4, int4)
* update\_sale(int4, float4, float4, int4, int4, int4)
* delete\_sale(int4)

*Συναρτήσεις*

* select\_all\_sales()
* select\_sales\_by\_customer(int4)

Η συνάρτηση αυτή δέχεται το ξένο κλειδί του πελάτη, το οποίο είναι διαθέσιμο απο το JComboBox του UI, και επιστρέφει ένα πίνακα με τις αγορές του.

* select\_sales\_by\_employee(int4)
* select\_sale\_by\_id(int4)
* select\_sales\_by\_month(int4)

Η συνάρτηση select\_sales\_by\_month επιστρέφει όλες τις πωλήσεις που πραγματοποιήθηκαν ένα συγκεκριμένο μήνα του χρόνου (1-12)

**Car Log**

* create\_car\_log()
* select\_all\_car\_logs()

**Employee Log**

* create\_employee\_log()
* select\_all\_employee\_logs()

1. **Επίλογος**

Το σύστημα διαχείρισης αντιπροσωπείας αυτοκινήτων που αναπτύξαμε είναι αρκετά κοντά στην ολοκλήρωση. Μερικές στοχευμένες βελτιώσεις μπορουν να το κανουν χρήσιμο ακόμα και σε υπάρχουσες αντιπροσωπείες. Κάνοντας χρήση απομακρυσμένης βάσης δεδομένων μια αντιπροσωπεία μπορεί να χρησιμοποιεί την εφαρμογή, ταυτόχρονα σε περισσότερους απο έναν υπολογιστές και είναι ασφαλή τα πιθανώς ευαίσθητα δεδομένα.