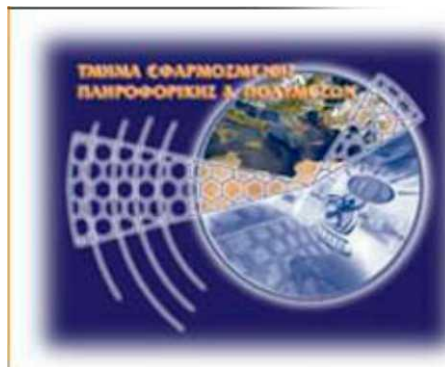




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής Τ.Ε



Πτυχιακή Εργασία Τίτλος: Μικροβιολογικό Εργαστήριο
Graduation Project Title : Microbiological Laboratory

Γουβιανάκης Ιδομενέας - Άλεν Α.Μ. 1989

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

Ημερομηνία παρουσίασης: 07 Σεπτεμβρίου 2015

Ευχαριστίες

Η πτυχιακή μου εργασία δε θα είχε ολοκληρωθεί αν δεν υπήρχε η αμέριστη υποστήριξη και εμπιστοσύνη ορισμένων ανθρώπων τους οποίους θέλω να ευχαριστήσω.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου Δρ. Παπαδάκη Νικόλαο που με εμπιστεύτηκε και μου ανάθεσε αυτήν την εργασία.

Πολλά ευχαριστώ σε όλους μου τους καθηγητές για τις γνώσεις που μου προσέφεραν και για τις σκέψεις που μου δημιούργησαν για τις επιλογές μου στην μετέπειτα πορεία μου στη ζωή.

Τέλος ιδιαίτερα ευχαριστώ στα αδέρφια μου Μαρία και Μιχάλη που είναι πάντα δίπλα μου και με στηρίζουν με το δικό τους τρόπο.

Περίληψη

Σε αυτή την πτυχιακή εργασία θα φτιάξουμε μια βάση δεδομένων για ένα μικροβιολογικό εργαστήριο που θα αναλύει τα στοιχεία κάποιων ασθενών και θα δίνει τις απαραίτητες πληροφορίες.

Summary

In this Graduation project we will make a database for a Microbiological Laboratory which will analyze the elements of some patients and give the necessary information.

Πίνακας Περιεχομένων

Links

Κεφάλαιο 1

Περιγραφή οντοτήτων.....4

Κεφάλαιο 2

Περιγραφή σχέσεων οντοτήτων.....6

Κεφάλαιο 3

Σχεσιακό μοντέλο..... 8

Κεφάλαιο 4

Περιορισμοί Ακεραιότητας.....11

Κεφάλαιο 5

Συναρτησιακές Εξαρτήσεις – Κλειδιά.....12

Κεφάλαιο 6

Τρίτη Κανονική Μορφή(3NF).....14

Κεφάλαιο 7

SQL Queries.....16

Ορισμοί σχέσεων σε Γλώσσα Ορισμού Δεδομένων

Κεφάλαιο 8

Εγχειρίδιο Χρήσης.....23

Κεφάλαιο 9

Ανάπτυξη της διαδικτυακής εφαρμογής.....27

Κεφάλαιο 10

Εγκατάσταση - Εκτέλεση εφαρμογής.....28

Links.....30

Περιγραφή οντοτήτων

Patient (Ασθενής / Πελάτης)

- patientId (PK, Long)
- firstName (String) - Όνομα
- lastName (String) - Επίθετο
- phoneNumber (String) – Τηλεφωνικός αριθμός
- address (String) – Διεύθυνση (Οδός, Αριθμός, ΤΚ, Περιοχη, Πόλη)
- socialSecurityNumber (String) – Μοναδικός κωδικός ασφαλισμένου (π.χ. ΑΜΚΑ)

Examination (Εξέταση/Τύπος Εξέτασης)

- examinationId (PK, Long)
- name (String) - Όνομα τύπου εξέτασης
- code (String) - Κωδικός αναφοράς εξέτασης
- cause (String) – Αίτιο εξέτασης (πχ ενδείξεις / συμπτώματα)

ExaminationResult (Διενέργεια Εξέτασης Ασθενή / Αποτελέσματα / Παραλαβή)

- examinationResultId (PK, Long)
- patientId (FK->Patient.patientId, Long) Ταυτοποίηση ασθενή για τον οποίο έγινε η εξέταση
- examinationId (FK->Examination.examinationId, Long) Ταυτοποίηση εξέτασης την οποία έκανε ο ασθενής
- examinationDate (Timestamp) Ημερομηνία διενέργειας της εξέτασης από τον ασθενή. Μελλοντική ημερομηνία υποδηλώνει τον προγραμματισμό αυτής
- results (String) Κείμενο που περιέχει τα αποτελέσματα της εξέτασης. Μη ύπαρξη τιμής υποδηλώνει την μη έκδοση αποτελέσματος μέχρι την δεδομένη στιγμή.
- resultDeliveryDate (Timestamp) Ημερομηνία παραλαβής των αποτελεσμάτων της εξέτασης. Μη ύπαρξη τιμής υποδηλώνει ότι ο ασθενής δεν έχει παραλάβει ακόμη τα αποτελέσματα.

Reagent (Αντιδραστήριο εξέτασης)

- reagentId (PK, Long)
- name (String) - Όνομα αντιδραστηρίου
- code (String) - Κωδικός αντιδραστηρίου
- minSupply (Double) – Ελάχιστη ποσότητα αντιδραστηρίου που θα πρέπει να έχει διαθέσιμη το μικροβιολογικό εργαστήριο, κάτω από την οποία είναι αναγκαία η παραγγελία επιπλέον ποσότητας

- `currSupply (Double)` – Η τρέχουσα ποσότητα του αντιδραστηρίου που έχει διαθέσιμη το μικροβιολογικό εργαστήριο για να την χρησιμοποιήσει σε εξετάσεις.

ReagentOrder (Παραγγελία [ενός ή περισσότερων] αντιδραστηρίων)

- `reagentOrderId (PK, Long)`
- `code (String)` - Κωδικός αναφοράς παραγγελίας
- `orderDate (Timestamp)` – Ημερομηνία παραγγελίας. Υποχρεωτικό πεδίο. Τιμή που τοποθετείται χρονικά στο μέλλον υποδηλώνει τον χρονοπρογραμματισμό μια παραγγελίας
- `deliveryDate (Timestamp)` – Ημερομηνία παραλαβής της παραγγελίας. Ενημερώνεται κατά την παραλαβή της παραγγελίας.

ReagentOrderEntry (Επιμέρους στοιχείο παραγγελίας, για τις λεπτομέρειες παραγγελίας ενός συγκεκριμένου αντιδραστηρίου)

- `reagentEntryOrderId (PK, Long)`
- `reagentOrderId (FK->ReagentOrder.reagentOrderId, Long)` - Ταυτοποίηση παραγγελίας, μέρος της οποίας ανήκουν τα στοιχεία/λεπτομέρειες παραγγελίας για το υποκείμενο αντιδραστήριο
- `reagentId (FK->Reagent.reagentId, Long)` - Ταυτοποίηση του αντιδραστηρίου για το οποίο καταχωρούνται επι μέρους στοιχεία/λεπτομέρειες παραγγελίας (εδώ η ποσότητα παραγγελίας)
- `quantity (Double)` – Η ποσότητα παραγγελίας για το υποκείμενο αντιδραστήριο

Περιγραφή σχέσεων οντοτήτων

(1..1) Patient – ExaminationResult (0..N)

Ένας ασθενής μπορεί να μην έχει κάνει καμία ή να έχει κάνει πολλαπλές εξετάσεις στο μικροβιολογικό εργαστήριο, συνεπώς μια οντότητα τύπου Patient μπορεί να αντιστοιχεί σε καμία ή περισσότερες οντότητες τύπου ExaminationResult. Μία οντότητα τύπου ExaminationResult αντιστοιχεί σε μια και μόνο μία οντότητα τύπου Patient

(0..N) ExaminationResult – Examination (1..1)

Μια διενέργεια εξέτασης (ExaminationResult) αφορά ένα συγκεκριμένο τύπο εξέτασης (Examination). Για έναν τύπο εξέτασης μπορεί να έχουν διενεργηθεί από καμία έως περισσότερες εξετάσεις για έναν ασθενή

(1..N) Examination – Reagent (1..N)

Για τη διενέργεια ενός συγκεκριμένου τύπου εξέτασης (Examination) απαιτείται η χρήση ενός ή περισσότερων αντιδραστηρίων (Reagent). Ένα συγκεκριμέν αντιδραστήριο μπορεί να συμμετέχει στη δημιουργία ενός ή περισσότερων τύπων εξέτασης. Η συμμετοχή αυτή του αντιδραστηρίου για την διενέργεια μιας εξέτασης, χαρακτηρίζεται από την ποσότητα που πρέπει να χρησιμοποιηθεί για την διενέργεια της εξέτασης. Συνεπώς, έχουμε να κάνουμε με μια N-N σχέση με χαρακτηριστικό που εννοιολογικά ανήκει στη σχέση αυτή.

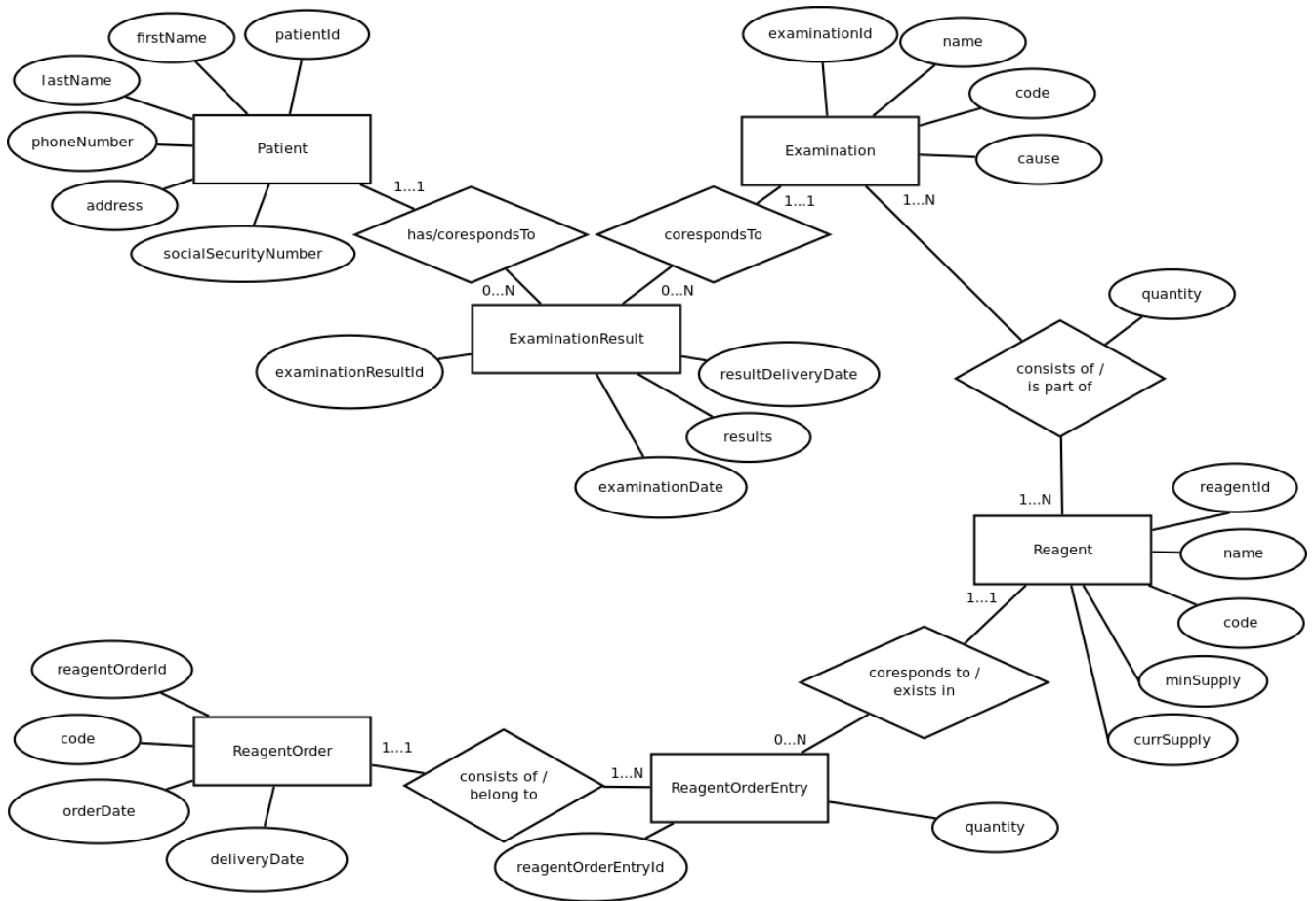
(1..1) ReagentOrder – ReagentOrderEntry (1..N)

Μια παραγγελία αντιδραστηρίων αποτελείται από τα επιμέρους στοιχεία της παραγγελίας για ένα ή περισσότερα αντιδραστήρια. Συνεπώς σε μια οντότητα τύπου ReagentOrder ομαδοποιούνται μία ή περισσότερες οντότητες τύπου ReagentOrderEntry. Αντιστρόφως μια τέτοια εγγραφή στοιχείων παραγγελίας ενός αντιδραστηρίου, αντιστοιχεί σε μία και μόνο παραγγελία.

(1..N) ReagentOrderEntry – Reagent (1..1)

Μια εγγραφή επιμέρους στοιχείων παραγγελίας (ReagentOrderEntry) αναφέρεται σε ένα και μόνο αντιδραστήριο. Για ένα αντιδραστήριο μπορεί να υπάρχουν πολλές εγγραφές στοιχείων παραγγελίας αυτού, σε διαφορετικές παραγγελίες.

Μικροβιολογικό Εργαστήριο



Σχεσιακό μοντέλο

Το σχεσιακό μοντέλο ακολουθεί στο μεγαλύτερο μέρος του το μοντέλο οντοτήτων σχέσεων με τις οντότητες να αντιστοιχούνται σε πίνακες και τις σχέσεις ($x \dots 1 - x \dots N$, $x \text{ in } [0,1]$) να υλοποιούνται με ξένο κλειδί στην πλευρά του N . Εξαιρέση αποτελούν οι σχέσεις ($x \dots N - x \dots N$, $x \text{ in } [0,1]$) οι οποίες υλοποιούνται σε ξεχωριστό πίνακα με ξένο κλειδί το πρωτεύον κλειδί κάθε μέρους μαζί με τα όποια επιπλέον γνωρίσματα της σχέσης.

Σημειώσεις:

- Επιλέχθηκε η χρήση μη φυσικών κλειδιών, ως πρωτεύοντα κλειδιά των πινάκων, δλδ κλειδιών – στηλών που η πληροφορία που φέρουν δεν έχει κάποια σημασία στο πλαίσιο της εφαρμογής.
- Η χρήση φυσικών κλειδιών, δλδ στηλών των οποίων το περιεχόμενο είναι μοναδικό από τη φύση του ή/και μες στο πλαίσιο-λειτουργίας/πεδίο-ορισμού της εφαρμογής, μπορεί να αποτελέσει πρόβλημα σε μελλοντικές αλλαγές στα δεδομένα αυτά, που μέχρι στιγμής μπορεί να θεωρούνται αμετάβλητα και μοναδικά. Σε μεγάλα συστήματα με πολλές εγγραφές τέτοιες αλλαγές μπορεί να επιφέρουν επιπλοκές των οποίων η ανάλυση βρίσκεται εκτός του θεωρητικού πεδίου και έχει να κάνει με τις πρακτικές συνέπειες των υφιστάμενων λειτουργικών υποδομών.
- Στη συγκεκριμένη υλοποίηση για την εξυπηρέτηση της ανάγκης υλοποίησης στον κώδικα μιας σχέσης (N Examination – Reagent N), η οποία φέρει γνώρισμα, η σχέση αναπαρίσταται ως μια ξεχωριστή οντότητα, την ExaminationComposition. Η οντότητα έχει δικό της, ξεχωριστό, πρωτεύον κλειδί (σε αντίθεση με ένα composite primary key που θα απαρτιζόταν από τα πρωτεύοντα κλειδιά των οντοτήτων/πινάκων των οποίων τη σχέση υλοποιεί), ακολουθώντας την ίδια μορφή με τις υπόλοιπες οντότητες της εφαρμογής. Επιπροσθέτως τα σύνθετα κλειδιά συνδυαζόμενα με άλλα σύνθετα κλειδιά δημιουργούν επιπλέον πρακτικά προβλήματα εκτός της θεωρητικής ανάλυσης, πχ, μπορεί να δημιουργήσουν πίνακες δυσνόητους, και πολύ πιο σημαντικό, (στο επίπεδο ανάπτυξης λογισμικού) πίνακες δύσχρηστους αναφορικά με την πληροφορία που πρέπει να δομείται προγραμματιστικά για να μπορέσει κάποιος αναφερθεί σε ένα τέτοιο σύνθετο κλειδί που απαρτίζεται από άλλα σύνθετα κλειδιά.

Πίνακες

Patient

- patientId (PK, BIGINT)
- firstName (NVARCHAR)
- lastName (NVARCHAR)
- phoneNumber (NVARCHAR)
- address (NVARCHAR)
- socialSecurityNumber (NVARCHAR)

Examination

- examinationId (PK, BIGINT)
- name (NVARCHAR)
- code (NVARCHAR)
- cause (NVARCHAR)

ExaminationResult

- examinationResultId (PK, BIGINT)
- patientId (FK->Patient.patientId, BIGINT)
- examinationId (FK->Examination.examinationId, BIGINT)
- examinationDate (TIMESTAMP)
- results (NVARCHAR)
- resultDeliveryDate (TIMESTAMP)

Reagent

- reagentId (PK, BIGINT)
- name (NVARCHAR)
- code (NVARCHAR)
- minSupply (FLOAT)
- currSupply (FLOAT)

ExaminationComposition (1..N Examination – Reagent 1..N)

- examinationCompositionId (PK, BIGINT)
- examinationId (FK->Examination.examinationId, BIGINT)
- reagentId (FK->Reagent.reagentId, BIGINT)
- quantity (FLOAT)

ReagentOrder

- reagentOrderId (PK, BIGINT)
- code (NVARCHAR)
- orderDate (TIMESTAMP)
- deliveryDate (TIMESTAMP)

ReagentOrderEntry

- reagentEntryOrderId (PK, BIGINT)
- reagentOrderId (FK->ReagentOrder.reagentOrderId, BIGINT)
- reagentId (FK->Reagent.reagentId, BIGINT)
- quantity (FLOAT)

Περιορισμοί Ακεραιότητας

Γενικοί κανόνες:

Όλα τα πρωτεύοντα κλειδιά (PK) πρέπει να έχουν τιμή (NOT NULL) και να είναι μοναδική στον πίνακα.

Όλα τα ξένα κλειδιά (FK->Table.PK) αναφέρονται σε ένα πρωτεύον κλειδί κάποιου άλλου πίνακα συνεπώς πρέπει να φέρουν μια τιμή (NOT NULL) η οποία να υπάρχει στην στήλη-πρωτεύον κλειδί του πίνακα στον οποίο αναφέρονται.

Όλα τα NVARCHAR πεδία μπορούν να οριστούν με ένα ενδεικτικό πλήθος 64 χαρακτήρων, με εξαίρεση το ExaminationResult.result, το οποίο πρέπει να έχει το μέγιστο επιτρεπτό (πχ NVARCHAR(MAX))

Όλα τα NVARCHAR πεδία αναφοράς (όλα τα “*.code” πεδία και Patient.socialSecurityNumber) πρέπει να είναι μοναδικα (UNIQUE CONSTRAINT)

Όλα τα FLOAT πεδία πρέπει να έχουν τιμή (NOT NULL), μη αρνητική (π.χ. check(quantity >= 0))

Το ExaminationResult.examinationDate, εφ όσον έχει τιμή, πρέπει να έπεται του ExaminationResult.resultDeliveryDate (ή οριακά να ισούται με αυτό).

Το ReagentOrder.deliveryDate, εφ όσον έχει τιμή, πρέπει να έπεται του ReagentOrder.orderDate (ή οριακά να ισούται με αυτό).

Ειδικοί περιορισμοί

Στον πίνακα ExaminationResult δεν μπορεί να υπάρχει πάνω από μια πλειάδα που να αναφέρεται στον ίδιο ασθενή, στον ίδιο τύπο εξέτασης, την ίδια χρονική στιγμή, καθώς δεν έχει και πρακτικό νόημα να κάνει ένας ασθενής την ίδια εξέταση πάνω από μία φορά την ίδια χρονική στιγμή. Συνεπώς, η τριπλέτα (patientId, examinationId, examinationDate) υπόκειται σε unique constraint

Ο πίνακας ExaminationComposition αντιπροσωπεύει μια σχέση μεταξύ δύο πινάκων οπότε εκ φύσεως φέρει κλειδί που απαρτίζεται από δύο ξένα κλειδιά, για τα οποία ισχύει ο περιορισμός ακεραιότητας της μοναδικότητας. Έχουμε λοιπόν UNIQUE CONSTRAINT στο ζεύγος (examinationId, reagentId)

Ο πίνακας ReagentOrderEntry αντιπροσωπεύει τις, ανα αντιδραστήριο, επι μέρους λεπτομέρεις παραγγελίας (στη παρούσα φάση, η ποσότητα παραγγελίας για το εν λόγω αντιδραστήριο). Επειδή μια παραγγελία μπορεί να αφορά περισσότερα από έν αντιδραστήρια αλλά κι επειδή κάθε αντιδραστήριο μπορεί να παραγγελθεί πολλές φορές (όποτε υπάρχει έλλειψη) ο συνδυασμός του πρωτεύοντος κλειδιού της παραγγελίας και του αντιδραστηρίου πρέπει να είναι μοναδικός (καθώς δεν έχει νόημα σε μια παραγγελία να υπάρχουν δύο ξεχωριστές καταχωρήσεις για το ίδιο αντιδραστήριο – θα υπάρχει μιά καταχώρηση με την απαιτούμενη ποσότητα). Συνεπώς έχουμε UNIQUE CONSTRAINT στις στήλες (reagentOrderId, reagentId)

Συναρτησιακές Εξαρτήσεις – Κλειδιά

Γενικές παρατηρήσεις

Πρέπει αρχικά να σημειωθεί ότι η απόφαση χρήσης μη φυσικών κλειδιών (surrogate keys) ως πρωτεύοντων κλειδιών για κάθε πίνακα επηρεάζει σε μεγάλο βαθμό το αποτέλεσμα της ανάλυσης των συναρτησιακών εξαρτήσεων. Εξ ορισμού όλα τα υπόλοιπα πεδία σε κάθε πίνακα εξαρτώνται συναρτησιακά από το αντίστοιχο μη φυσικό πρωτεύον κλειδί που έχει επιλεγθεί.

Τα φυσικά κλειδιά σε κάθε πίνακα προσδιορίζουν συναρτησιακά τα υπόλοιπα πεδία (μαζί και τα surrogate keys)

Θα μπορούσαμε να παραλείψουμε τα surrogate keys από την ανάλυση, αλλά θα τα συμπεριλάβουμε τόσο για λόγους ορθότητας - επειδή συμμετέχουν σε πίνακες που υλοποιούν σχέσεις οντοτήτων (έμμεσα όπως το ExaminationResult ή άμεσα όπως το ExaminationComposition) όσο και για λόγους πληρότητας.

Patient:

socialSecurityNumber → firstName, lastName, phoneNumber, address[, patientId]

patientId → firstName, lastName, phoneNumber, address, socialSecurityNumber

Examination:

code → name, cause[, examinationId]

examinationId → code, name, cause

ExaminationResult:

patientId, examinationId, examinationDate → results, resultDeliveryDate[, examinationResultId]

examinationResultId → patientId, examinationId, examinationDate, results, resultDeliveryDate

Reagent:

code → name, minSupply, currSupply[, reagentId]

reagentId → code, name, minSupply, currSupply

ExaminationComposition:

examinationId, reagentId → quantity[, examinationCompositionId]

examinationCompositionId → examinationId, reagentId, quantity

ReagentOrder:

code → orderDate, deliveryDate[, reagentOrderId]

reagentOrderId → code, orderDate, deliveryDate

ReagentOrderEntry:

reagentOrderId, reagentId → quantity[, reagentOrderEntryId]

reagentOrderEntryId → reagentOrderId, reagentId, quantity

Σημειώσεις:

- με κίτρινη σκίαση παρουσιάζονται τόσο τα surrogate keys όσο και οι συναρτησιακές εξαρτήσεις που προκύπτουν από την ύπαρξη αυτών.
- Με πορτοκαλί σκίαση σηματοδοτούνται τα surrogate keys που συμμετέχουν ως σύνθετα κλειδιά σε συναρτησιακές εξαρτήσεις, στη θέση των οποίων, αν δεν χρησιμοποιούσαμε surrogate keys, θα μπαίναν τα αντίστοιχα πρωτεύοντα κλειδιά που προκύπτουν για τον αντίστοιχο πίνακα. Για παράδειγμα, στον πίνακα ExaminationResult, τα πεδία (patientId, examinationId, examinationDate) προσδιορίζουν συναρτησιακά τα results, resultDeliveryDate[, examinationResultId]. Αν δεν χρησιμοποιούσαμε surrogate keys, στη θέση τους, ως κλειδιά των πινάκων Patient και Examination θα χρησιμοποιούσαμε τα socialSecurityNumber και code αντίστοιχα. Συνεπώς στον πίνακα ExaminationResult η συναρτησιακή εξάρτηση θα οριζόταν ως socialSecurityNumber, code, examinationDate → results, resultDeliveryDate
- Η επιλογή χρήσης κλειδιών χωρίς φυσική σημασία στο πεδίο του μοντέλου οντοτήτων μας έχει τεκμηριωθεί παραπάνω.
- Έχουν παραλειφθεί οι τετριμμένες συναρτησιακές εξαρτήσεις.

Τρίτη Κανονική Μορφή (3NF)

Για να μετατραπεί το σχήμα σε 3NF θα πρέπει να γίνουν κάποιες μετατροπές στον πίνακα Patient καθώς είναι ο μόνος που δεν πληρεί καμιά κανονική μορφή καθώς περιέχει το σύνθετο γνώρισμα address. Για να μετατραπεί σε 1NF σπάμε το σύνθετο γνώρισμα σε πολλά απλά. Υποθέτοντας ότι η διεύθυνση απαρτίζεται από τα city (πόλη), street (οδό), streetNumber (αριθμό), postalCode (TK), ο πίνακας γίνεται:

Patient

- patientId (PK, BIGINT)
- firstName (NVARCHAR)
- lastName (NVARCHAR)
- phoneNumber (NVARCHAR)
- socialSecurityNumber (NVARCHAR)
- city (NVARCHAR)
- street (NVARCHAR)
- streetNumber (NVARCHAR) (υποστήριξη μορφών τύπου “35A”)
- postalCode (NVARCHAR)

για μετατροπή σε 2η κανονική μορφή δημιουργούμε επιπλέον πίνακα

PatientAddress

- patientId (PK, BIGINT)
- city (NVARCHAR)
- street (NVARCHAR)
- streetNumber (INTEGER)
- postalCode (NVARCHAR)

όπου έχουμε

patientId → city, street, streetNumber, postalCode

Σε αυτόν τον πίνακα (δεδομένου ότι ένας T.K. Αντιστοιχεί σε μια οδό μιας πόλης) ισχύει επιπλέον

postalCode → city, street

πρέπει να εξαλείψουμε την μεταβατική συναρτησιακή εξάρτηση patientId → city, street
πρέπει να μεταφέρουμε την πληροφορία της διεύθυνσης σε ξεχωριστό πίνακα με πρωτεύον κλειδί τον T.K. Με τον οποίο θα συνδέεται με την πληροφορία:

PostalCode

- postalCodeId (PK, BIGINT)
- postalCode (NVARCHAR)
- city (NVARCHAR)
- street (NVARCHAR)

ενώ ο **PatientAddress** θα γίνει

- patientId (PK, BIGINT)
- streetNumber (INTEGER)
- postalCodeId (FK->PostalCode.postalCodeId, BIGINT)

και **Patient**

- patientId (PK, BIGINT)
- firstName (NVARCHAR)
- lastName (NVARCHAR)
- phoneNumber (NVARCHAR)
- socialSecurityNumber (NVARCHAR)

οι οποίοι πίνακες βρίσκονται πλέον σε 3NF. Δεν υπάρχει απώλεια πληροφορίας καθώς το εσωτερικό γινόμενο (inner join) αυτών μας δίνει την πληροφορία που είχαμε στην αρχή χωρίς επιπλέον λανθασμένες πλειάδες.

SQL Queries

Οι αναζητήσεις δεδομένων στη καθορίζονται εν πολλοίς και από το τι αναζητήσεις επιτρέπει το web interface της εφαρμογής. Σε γενικές γραμμές

- υπάρχει μια αντιστοιχία μίας διαχειριστικής οθόνης για κάθε μία οντότητα στο ER-Model του Μικροβιολογικού εργαστηρίου.
- παρέχεται η δυνατότητα αναζήτησης κάθε τέτοιας κεντρικής οντότητας με βάση ένα ή περισσότερα από τα πεδία αυτής της οντότητας. Για κάθε πεδίο μιας οντότητας αντιστοιχεί ένα πεδίο εισόδου τιμής αναζήτησης η οποία τιμή συγκρίνεται είτε με τον τελεστή ισότητας '=' για αριθμητικές τιμές είτε με τον τελεστή 'LIKE' για αλφαριθμητικές (επιτρέποντας τη χρήση '%' ως wildcard). Εξαιρέση αποτελούν τα πεδία ημερομηνιών για τα οποία δεν έχει νόημα η σύγκριση ισότητας με μια δεδομένη τιμή, συνεπώς για κάθε πεδίο ημερομηνίας μιας οντότητας αντιστοιχούνται δύο πεδία εισόδου τιμής αναζήτησης που έχουν την έννοια του “από” και “έως” και η αναζήτηση ελέγχει εφόσον το πεδίο της οντότητας βρίσκεται μες στο διάστημα [“από”, “έως”].
- η αναζήτηση με βάση οποιοδήποτε πεδίο επιτυγχάνεται με τη χρήση ενός query το οποίο είναι δομημένο έτσι ώστε να παίρνει υπόψιν του την παρουσία ή απουσία τιμής αναζήτησης για οποιοδήποτε από τα πεδία αναζήτησης
- συγκεκριμένα για το πρωτεύον κλειδί κάθε οντότητας, εφόσον έχει δοθεί τιμή αναζήτησης, δεν λογαριάζεται καμιά άλλη τιμή καθώς αντιστοιχεί σε μία και μόνο καταχώρηση, οπότε δεν έχει νόημα η αναζήτηση σε συνδυασμό με κάποιο άλλο πεδίο

Με βάση τα παραπάνω τα queries αναζήτησης για κάθε κεντρική οντότητα δομούνται ως εξής (Σημαντική Σημείωση: Οι επερωτήσεις είναι γραμμένες σε JPA QL [1], [2]. Για τις παρακάτω επερωτήσεις οι διαφορές είναι κυρίως συντακτικές. Όπου χρειάζεται παρέχεται και η αντίστοιχη επερώτηση σε standard SQL)

Patient

```
select patient from Patient patient
```

```
where
```

```
(patient.patientId = :patientId)
```

```
or
```

```
(
```

```
(null = :patientId)
```

```
and ((null = :firstName) or (patient.firstName like :firstName))
```

```
and ((null = :lastName) or (patient.lastName like :lastName))
```

```
and ((null = :phoneNumber) or (patient.phoneNumber like :phoneNumber))
```

```
and ((null = :address) or (patient.address like :address))
```

```
and ((null = :socialSecurityNumber) or (patient.socialSecurityNumber
```



```
like :socialSecurityNumber))  
    )
```

Επιπλέον για τους ασθενείς υπάρχει και η αναζήτηση αυτών που δεν έχουν λάβει τα αποτελέσματά τους. Συνεπώς υπάρχει ακόμα μια επερώτηση στη βάση για τις εγγραφές Patient για τις οποίες υπάρχουν εγγραφές στον ExaminationResult όπου το πεδίο resultNotes έχει τιμή (σηματοδοτεί το ότι έχουν βγεί τα αποτελέσματα) ενώ το πεδίο resultDeliveryDate δεν έχει τιμή (η ύπαρξη τιμής σηματοδοτεί την παράδοση των αποτελεσμάτων):

```
select patient from Patient patient where patient.patientId in  
    (select examinationResult.patient.patientId from ExaminationResult  
examinationResult  
    where (examinationResult.resultNotes is not null)  
    and (examinationResult.resultDeliveryDate is null)  
    )
```

Η επερώτηση σε standard SQL (χωρίς τη χρήση inner join):

```
select patient.* from Patient patient where patient.patientId in  
    (select examinationResult.patientId from ExaminationResult examinationResult  
    where (examinationResult.resultNotes is not null)  
    and (examinationResult.resultDeliveryDate is null)  
    )
```

Examination

```
select examination  
from Examination examination  
where (examination.examinationId = :examinationId)  
or  
(  
    (null = :examinationId)  
    and ((null = :name) or (examination.name like :name))  
    and ((null = :code) or (examination.code like :code))  
    and ((null = :cause) or (examination.cause like :cause))  
)
```

ExaminationResult

```
select examinationResult from ExaminationResult examinationResult
where
(examinationResult.examinationResultId = :examinationResultId)
or
(
    (null = :examinationResultId)
    and ((null = :patientId) or (examinationResult.patientId = :patientId))
    and ((null = :examinationId) or (examinationResult.examinationId = :examinationId))

    and ((null = :examinationDateFrom) or
(examinationResult.examinationDate >= :examinationDateFrom))
    and ((null = :examinationDateTo) or (examinationResult.examinationDate
< :examinationDateTo))
    and ((null = :resultNotes) or (examinationResult.resultNotes like :resultNotes))
    and ((null = :resultDeliveryDateFrom) or
(examinationResult.resultDeliveryDate >= :resultDeliveryDateFrom))
    and ((null = :resultDeliveryDateTo) or (examinationResult.resultDeliveryDate
< :resultDeliveryDateTo))
)
```

Reagent

```
select reagent
from Reagent reagent
where
(reagent.reagentId = :reagentId)
or
(
    (null = :reagentId)
    and ((null = :name) or (reagent.name like :name))
    and ((null = :code) or (reagent.code like :code))
)
```

ExaminationComposition

```
select examinationComposition
from ExaminationComposition examinationComposition
where
(examinationComposition.examinationCompositionId = :examinationCompositionId)
or
(
  (null = :examinationCompositionId)
  and ((null = :examinationId) or (examinationComposition.examinationId
= :examinationId))
  and ((null = :reagentId) or (examinationComposition.reagentId = :reagentId))
)
```

ReagentOrder

```
select reagentOrder from ReagentOrder reagentOrder
where
(reagentOrder.reagentOrderId = :reagentOrderId)
or
(
  (null = :reagentOrderId)
  and ((null = :code) or (reagentOrder.code = :code))
  and ((null = :orderDateFrom) or (reagentOrder.orderDate >= :orderDateFrom))
  and ((null = :orderDateTo) or (reagentOrder.orderDate < :orderDateTo))
)
```

Ορισμοί σχέσεων σε Γλώσσα Ορισμού Δεδομένων

Για την οντότητα **Patient**

```
create table microbiologyLab.patient (  
  patient_id bigint not null,  
  address varchar(255),  
  first_name varchar(255) not null,  
  last_name varchar(255) not null,  
  phone_number varchar(255),  
  social_security_number varchar(255) not null,  
  primary key (patient_id)  
);  
  
create unique index UK_9docwt49y5gmrqxoe05swnas on microbiologyLab.patient  
(social_security_number);
```

Για την οντότητα **Examination**

```
create table microbiologyLab.examination (  
  examination_id bigint not null,  
  cause varchar(255),  
  code varchar(255) not null,  
  name varchar(255),  
  primary key (examination_id)  
);  
  
create unique index UK_k6a296hgxtjbpfg3dtwdwhgnh on microbiologyLab.examination  
(code);
```

Για την οντότητα **ExaminationResult**

```
create table microbiologyLab.examination_result (  
  examination_result_id bigint not null,  
  examination_date datetime not null,  
  examination_id bigint not null,  
  patient_id bigint not null,  
  result_delivery_date datetime,  
  result_notes varchar(255),  
  primary key (examination_result_id)  
);
```

Για την οντότητα **Reagent**

```
create table microbiologyLab.reagent (  
    reagent_id bigint not null,  
    code varchar(255) not null,  
    curr_supply double(22, 0) not null,  
    min_supply double(22, 0) not null,  
    name varchar(255),  
    primary key (reagent_id)  
);  
create unique index UK_jokrgderi0g4h7gtgrelcgu6l on microbiologyLab.reagent (code);
```

Για την οντότητα **ExaminationComposition**

```
create table microbiologyLab.examination_composition (  
    examination_composition_id bigint not null,  
    examination_id bigint not null,  
    quantity double(22, 0) not null,  
    reagent_id bigint not null,  
    primary key (examination_composition_id)  
);  
create unique index UK_bdrsp6mxswb3kfw86u5ea9hx2 on  
microbiologyLab.examination_composition (examination_id, reagent_id);
```

Για την οντότητα **ReagentOrder**

```
create table microbiologyLab.reagent_order (  
    reagent_order_id bigint not null,  
    code varchar(255) not null,  
    delivery_date datetime,  
    order_date datetime not null,  
    primary key (reagent_order_id)  
);  
create unique index UK_9481lpiy9xe6gb3eowiqbt934 on microbiologyLab.reagent_order  
(code);
```

Για την οντότητα **ReagentOrderEntry**

```
create table microbiologyLab.reagent_order_entry (  
  reagent_order_entry_id bigint not null,  
  quantity double(22, 0) not null,  
  reagent_id bigint not null,  
  reagent_order_id bigint not null,  
  primary key (reagent_order_entry_id),  
  foreign key FK_c2uarxy4xagxmwtkbsa8abwkx (reagent_order_id) references  
  reagent_order(reagent_order_id),  
  foreign key FK_3xb1761j60t507j65iudv9ao3 (reagent_id) references reagent(reagent_id)  
);  
  
create unique index UK_d5gb3oiwomlv8slt8txa7udss on  
microbiologyLab.reagent_order_entry (reagent_order_id, reagent_id);
```

Εγχειρίδιο Χρήσης

Ακολουθεί μια σύντομη περιγραφή χρήσης του συστήματος.

Η εφαρμογή έχει ένα αρκετά απλό και άμεσο user interface. Η εισαγωγή της πληροφορίας γίνεται τμηματικά για κάθε εννοιολογικό κομμάτι του συστήματος (Ασθενείς, Αντιδραστήρια, Τύποι εξετάσεων, Σύνθεση τύπων εξετάσεων, Εξετάσεις Ασθενών & Αποτελέσματα, Παραγγελίες Αντιδραστηρίων), σε ξεχωριστή σελίδα. Η πλοήγηση από την μία σελίδα στην άλλη γίνεται με συνδέσμους που βρίσκονται, πάντα ορατοί, στην κορυφή της οθόνης.

Illustration

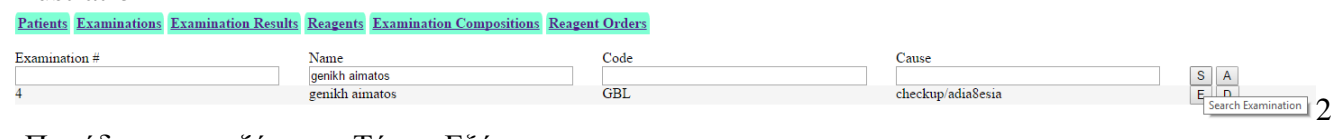


Σε γενικές γραμμές κάθε τέτοια σελίδα δίνει τις παρακάτω βασικές δυνατότητες στον χρήστη:

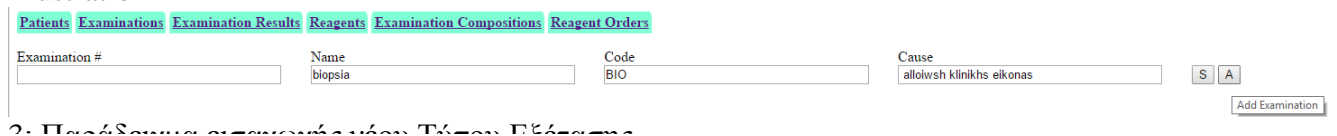
1. Αναζήτηση εγγραφών του τύπου πληροφορίας που προβάλλει (πχ Ασθενών).
2. Καταχώρηση νέων εγγραφών του τύπου πληροφορίας που προβάλλει (πχ Ασθενών)

Για την παρουσίαση μιας λειτουργικής αλλά και εύχρηστης διεπαφής στο χρήστη, όπου ήταν δυνατόν, τα πεδία εισόδου πληροφορίας ήταν κοινά για τις δύο αυτές λειτουργίες και οι διαφορετικές λειτουργίες προσφέρονταν σε διαφορετικά κουμπιά. Επομένως, η πληροφορία που εισάγεται σε κάθε πεδίο εισόδου, είτε αποτελεί τιμή αναζήτησης αν επιλεγεί αναζήτηση, ή τιμή νέας εγγραφής αν επιλεγεί προσθήκη εγγραφής. Τα κουμπιά αναζήτησης και εισαγωγής φέρουν τα γράμματα 'S' και 'A'.

Illustration



Illustration



Σε άλλες περιπτώσεις όπου η αναζήτηση περιέχει διαφορετικά πεδία από αυτά της προσθήκης εγγραφής, τότε προσφέρονται διαφορετικά πεδία εισόδου μαζί με το αντίστοιχο κουμπί λειτουργίας για την αναζήτηση και διαφορετικά για την εισαγωγή εγγραφής.

Illustration

Examination Result #	Patient	Examination	Examination Date From	Examination Date To	Result Notes	Result Delivery Date From	Result Delivery Date To
1	1 1	1	Wed Jun 10 2015		Please Redo Check	Wed Jun 17 2015	

4. Παράδειγμα αναζήτησης πεδίου εισόδου. Αναζήτηση και εισαγωγή

Κάτω από αυτή τη νοητή περιοχή της οθόνης εμφανίζονται σε λίστα οι υπάρχουσες εγγραφές για τον προβαλλόμενο τύπο πληροφορίας. Η λίστα αυτή περιέχει τις εγγραφές ως αποτέλεσμα της αναζήτησης που έχει γίνει (αρχικά, κατά την είσοδο σε μια σελίδα, γίνεται αναζήτηση χωρίς περιορισμούς οπότε προβάλλονται όλες οι εγγραφές) Σε κάθε εγγραφή προσφέρονται οι υπόλοιπες λειτουργίες που μπορεί να εφαρμόσει κάποιος. Για να είναι ευδιάκριτες οπτικά σε κάθε εγγραφή εναλλάσσεται το χρώμα παρασκήνιου

Illustration

Examination #	Name	Code	Cause	S	A	E	D
1	1	1	1				
2	2	2	2				
3	3	3	3				
4	genikh aimatos	GBL	checkup/adiaesia				
5	genikh ourwn	GUR	check				

5. Παράδειγμα αναζήτησης

3. Τροποποίηση της εγγραφής (κουμπί 'E'):

Illustration

14	Gouvianaki Maria	genikh ourwn	Daylight Time Thu Jul 23 2015 00:00:00 GMT+0300 (GTB Daylight Time)	E	D
----	------------------	--------------	--	---	---

6. Κουμπί Τροποποίηση (Οθόνη Είσοδος και Αποβολής του Αρθρώ)

κατά την επιλογή της τα πεδία της εγγραφής μετατρέπονται σε τροποποιήσιμα πεδία εισόδου (για όσα έχει νόημα). Σε αυτή τη κατάσταση προσφέρονται στο χρήστη οι δυνατότητες

- i. Ενημέρωσης της εγγραφής με τις όποιες αλλαγές (αν) έχουν γίνει (κουμπί 'U').

Illustration

14	Gouvianaki Maria	genikh ourwn	07 / 23 / 2015	epaneksetash	mm / dd / yyyy	U	C
----	------------------	--------------	----------------	--------------	----------------	---	---

7. Κουμπί Ενημέρωση εγγραφής (Οθόνη Είσοδος & Αποβολής του Αρθρώ)

- ii. Ακύρωσης των τροποποιήσεων που έχει κάνει (κουμπί 'C').

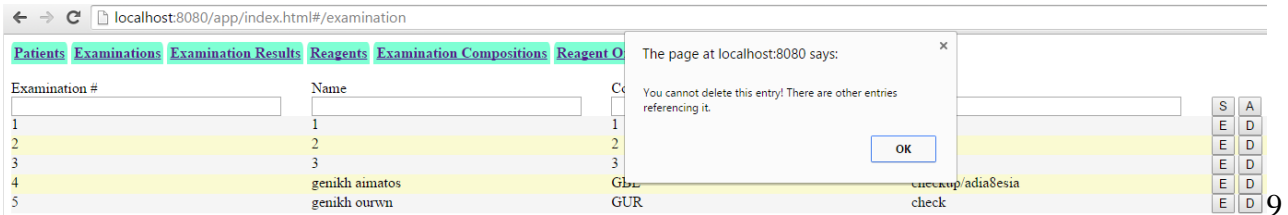
Illustration

Reagent #	Name	Code	Minimum Supply	Current Supply	S	A	U	C	E	D
1	1	1	2000	8						
2	2	2	3	2						

- 4. Διαγραφή της εγγραφής (κουμπί 'D'): Επιτρέπει στο χρήστη να διαγράψει την εν λόγω εγγραφή. Προσοχή! Για λόγους ακεραιότητας των δεδομένων δεν είναι δυνατή η διαγραφή κάποιας εγγραφής (πχ Αντιδραστήριο) όταν αυτή συμμετέχει στην δόμηση της πληροφορίας ενός άλλου τύπου πληροφορίας (πχ, όταν το αντιδραστήριο

συμμετέχει στον ορισμό της σύνθεσης ενός τύπου εξέτασης). Το σχετικό μήνυμα παρουσιάζεται στον χρήστη εφόσον επιχειρήσει μια τέτοια διαγραφή.

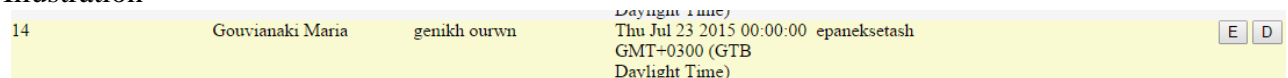
Illustration



9 : Απο τέλεσμα προσπάθειας Διαγραφής Τύπου Εξέταση για τον οποίο υπάρχουν δεδομένα

Όποια επιλογή κι αν επιλεγεί από το χρήστη, η διεπαφή έπειτα επιστρέφει στην αρχική μορφή παρουσίασης των αποτελεσμάτων αναζήτησης με το αποτέλεσμα της ενέργειας που έγινε (πχ διαγραφή)

Illustration

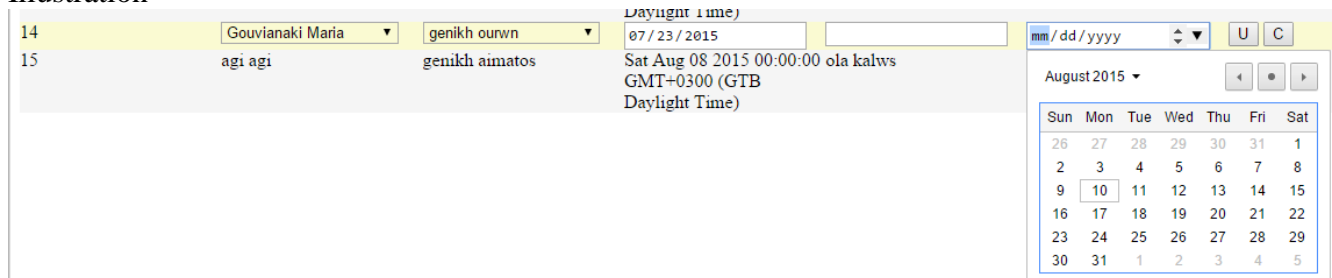


10: Επιστροφή Εμφάνισής Εμφανισής

Όπως φαίνεται στις προεπισκοπίσεις, σε όλα τα κουμπιά λειτουργιών, αν αφηθεί ο δείκτης πάνω από αυτά εμφανίζεται μια σύντομη περιγραφή της λειτουργία που προσφέρουν.

Εδώ πρέπει να σημειωθεί ότι για την εισαγωγή δεδομένων τύπου ημερομηνίας, χρησιμοποιήθηκε η δυνατότητα που προσφέρει το html v.5 όπου μπορεί κανείς να ορίσει πεδία εισόδου (input) με τύπο date. Αυτή τη στιγμή μόνο ο περιηγητής Chrome υλοποιεί αυτό το κομμάτι των προδιαγραφών του html5 και συνεπώς μόνο με αυτόν μπορεί να χρησιμοποιηθεί σωστά η εφαρμογή (Στους υπόλοιπους περιηγητές η ημερομηνία πρέπει να εισαχθεί ως κείμενο σε μορφή yy-MM-dd)

Illustration



11: Πεδίο εισόδου τύπου ημερομηνίας html5

Πέρα από τις λειτουργίες που περιγράφηκαν παραπάνω, το σύστημα προσφέρει μια επιπλέον λειτουργία αναζήτησης των Ασθενών για τους οποίους υπάρχει τουλάχιστον μια εγγραφή μιας χρονοπρογραμματισμένης εξέτασης, στην οποία έχει συμπληρωθεί το πεδίο του αποτελέσματος της εξέτασης (δλδ έχουν βγει τα αποτελέσματα της εξέτασης), αλλά το πεδίο ημερομηνίας παραλαβής της εξέτασης δεν έχει τιμή, (δλδ ο ασθενής δεν έχει παραλάβει τα αποτελέσματα)

Patient #	First Name	Last Name	Phone Number	Address	Social Security Number	Operations
1	1	1	1	1	1	S A P E D Patients who haven't received their examination results

Τέλος, ειδική μνεία πρέπει να γίνει για την οθόνη Παραγγελιών Αντιδραστηρίων, όπου η πληροφορία παρουσιάζεται σε 2 επίπεδα. Στο πρώτο επίπεδο έχουμε την εισαγωγή/αναζήτηση Παραγγελιών και από κάτω εμφανίζονται οι εγγραφές της παραγγελίας. Κάθε εγγραφή Παραγγελίας όμως αποτελεί με τη σειρά της μια παρόμοια δομή πεδίων εισόδου όπου δίνεται η δυνατότητα εισαγωγής των επι μέρους εγγραφών παραγγελίας Αντιδραστηρίων (για την εν λόγω παραγγελία). Η αναζήτηση δεν έχει νόημα και δεν υπάρχει σχετικό κουμπί. Έτσι στην εγγραφή μιας παραγγελίας παρουσιάζεται μια ομάδα πεδίων εισόδου επι μέρους Εγγραφών Παραγγελίας Αντιδραστηρίων και από κάτω οι υπάρχουσες τέτοιες εγγραφές για την εν λόγω παραγγελία.

Patients **Examinations** **Examination Results** **Reagents** **Examination Compositions** **Reagent Orders**

Order #	Code	Order Date From	Order Date To	Result Delivery Date From	Result Delivery Date To
Order #	Code	mm / dd / yyyy	mm / dd / yyyy	mm / dd / yyyy	mm / dd / yyyy
Order #	Code	mm / dd / yyyy	mm / dd / yyyy		
2	TEST	Thu Jul 23 2015 03:00:00 GMT+0300 (GTB Daylight Time)			
Reagent Order Entry #	Reagent Order	Reagent	Reagent Quantity		
1	TEST	1	10		
2	TEST	2	20		
4	Paragellia Olwn apo 10	Mon Aug 10 2015 00:00:00 GMT+0300 (GTB Daylight Time)			
Reagent Order Entry #	Reagent Order	Reagent	Reagent Quantity		
5	Paragellia Olwn apo 10	1	10		
6	Paragellia Olwn apo 10	2	10		
7	Paragellia Olwn apo 10	3	10		
10	Paragellia Olwn apo 10	4	10		
7	test2	Sat Aug 22 2015 00:00:00 GMT+0300 (GTB Daylight Time)			
Reagent Order Entry #	Reagent Order	Reagent	Reagent Quantity		
12	test2	1	10		

Ανάπτυξη της διαδικτυακής εφαρμογής

Για την δημιουργία της εφαρμογής χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες

- **Maven** επιτρέπει την οργάνωση και αυτοματοποίησης μεταφρασης του κώδικα και δημιουργίας των εκτελέσιμων. Με την κατάλληλη καθοδήγηση προσφέρει αυτόματη εύρεση και ανάκτηση όλων των προγραμματιστικών εξαρτήσεων σε τρίτες βιβλιοθήκες και τεχνολογίες, γενικά. [3]
- **Springboot** δέσμη τεχνολογιών (framework) το οποίο επιτρέπει την γρήγορη ανάπτυξη εφαρμογών, προσφέροντας τόσο επιλογές ρυθμίσεων Maven για την οργάνωση ενός Project με τις επιθυμητές τεχνολογίες όσο και τεχνολογίες άμεσης δημιουργίας διαφόρων χαρακτηριστικών και προσφερόμενων λειτουργιών, όπως REST web services, Persistence layer για επικοινωνία με τη βάση, ακόμα και την δημιουργία εκτελέσιμων που εμπεριέχουν και τον Application Server όπου θα τρέξει η εφαρμογή για γρήγορη δημιουργία ενός application σε ένα μόνο εκτελέσιμο!
- **Java EE** οι επερωτήσεις στη βάση γίνονται με τη γλώσσα JPA QL. Οι σχέσεις των οντοτήτων ακολουθούν το JPA στανταρντ.
- **Angular JS** τεχνολογία javascript που επιτρέπει την δημιουργία single-page applications προσφέροντας μηχανισμούς υποστήριξης της διεπαφής του χρήστη και της πλοήγησης σε αυτή, που μέχρι τώρα πρόσφεραν server-side τεχνολογίες. Εγκατάσταση – Εκτέλεση εφαρμογής

Εγκατάσταση – Εκτέλεση εφαρμογής

Για την εγκατάσταση της εφαρμογής απαιτείται η ύπαρξη μιας βάσης δεδομένων. Η βάση που θα χρησιμοποιηθεί (MySQL, in memory db, κλπ) μπορεί να καθοριστεί στο σετάρισμα των αρχείων pom.xml (αρχεία maven) με τα οποία το spring-boot θα αναλάβει να συνδέσει την εφαρμογή. Οι όποιες ρυθμίσεις σύνδεσης καθορίζονται στο αρχείο application.properties . Έστω ότι το project της εφαρμογής βρίσκεται στο φάκελο MicrobiologyLab, το αρχείο αυτό βρίσκεται στο path MicrobiologyLab\src\main\resources\application.properties

Το project έχει σεταριστεί έτσι ώστε μπορεί να τρέξει είτε **κατευθείαν** από την ιεραρχία φακέλων που βρίσκεται ο κώδικας με τη χρήση του maven είτε τρέχοντας το **εκτελέσιμο Jar** που παράγεται από το maven για το project.

Για να τρέξει κανείς εντολές του maven πρέπει να εγκαταστήσει το maven (κατεβάζουμε το τελευταίο version και βάζουμε το φάκελο του εκτελέσιμου – πχ \maven\apache-maven-3.3.3\bin – στο path του συστήματος. (Υπόψιν ότι χρειάζεται να υπάρχει συνδεσιμότητα με το Internet για την ληψη όλων των εξαρτήσεων)

Για να τρέξουμε την εφαρμογή από το Project εκτελούμε στο φάκελο του project

```
mvn spring-boot:run
```

Για να εκτελέσουμε την εφαρμογή από το jar, αρχικά τρέχουμε την εντολή mvn install, για να δημιουργηθεί. Το εκτελέσιμο θα δημιουργηθεί στο

```
MicrobiologyLab\target\microbiologyLab-0.0.1-SNAPSHOT.jar
```

Έπειτα εκτελώντας

```
java -jar microbiologyLab-0.0.1-SNAPSHOT.jar
```

ξεκινάει η εφαρμογή

Ανοίγουμε ένα περιηγητή (γι αυτό το version της εφαρμογής τον Chrome browser) και επισκεπτόμαστε το url

<http://localhost:8080/app/index.html>

Σημείωση: εφ' όσον δεν υπάρχει διαθέσιμη βάση πχ MySQL το πρόγραμμα μπορεί να τρέξει με on-memory βάση. Αρκεί να μπει στο MicrobiologyLab\pom.xml η ρύθμιση για τη χρήση βάσης “h2” (βρίσκεται σε σχόλια) και να βγει η ρύθμιση για τη MySQL.

Περιορισμοί Υλοποίησης – Δυνατότητες βελτίωσης

Ένας περιορισμός που αναφέρθηκε νωρίτερο στο παρόν κείμενο σχετίζεται με την χρήση του `<input>` element με βάση το html5 specification. Αυτό έχει σαν αποτέλεσμα για τους υπόλοιπους περιηγητές τα πεδία εισόδου τύπου ημερομηνίας να φαίνονται σαν πεδία εισόδου κειμένου. Σε αυτά για να μπορέσει να εισάγει τιμή ο χρήστης θα πρέπει να την εισάγει στη μορφή `yyyy-MM-dd`. Για την βελτίωση αυτού του προβλήματος θα έπρεπε να χρησιμοποιηθεί κάποια έτοιμη λύση μιας βιβλιοθήκης javascript.

Τέλος, ο πιο σημαντικός περιορισμός αυτή τη στιγμή είναι ότι η εφαρμογή δεν διαχειρίζεται σωστά τα ελληνικά, οπότε τα ονόματα για να είναι αναγνώσιμα πρέπει να εισάγονται με λατινικούς χαρακτήρες. Για να διορθωθεί πρέπει να σεταριστεί σωστά το encoding των requests να ερμηνεύεται ως UTF-8 και όχι ως ISO-8859-1.

Αναφορικά με τις βελτιώσεις που θα μπορούσαν να γίνουν, η πιο σημαντική είναι η επιλογή των ασθενών στην σελίδα χρονοπρογραμματισμού εξετάσεων.

Για την περίπτωση προσθήκης εξέτασης, η επιλογή ασθενή θα μπορούσε να γίνεται με κάποιας μορφής αναζήτηση, αντί για επιλογή από το `<select>` element, καθώς για πάρα πολύ μεγάλο πλήθος εγγεγραμμένων ασθενών η χρήση του θα γίνεται πιο δύσκολη για το χρήστη. Αυτό θα μπορούσε να παρακαμφθεί είτε αν στη θέση του drop down list υπήρχε ένα παρόμοιο πεδίο εισόδου όπου επέτρεπε να πληκτρολογεί κανείς και με βάση την πληκτρολόγηση να φίλτραρε και τα προσφερόμενα αποτελέσματα. Διαφορετικά, θα μπορούσε να χρησιμοποιηθεί μια επιπλέον ομάδα πεδίων αναζήτησης, σκοπός της οποίας θα ήταν να διαλέξει το χρήστη που θέλουμε να επιλεγεί.

Για τα πεδία εισόδου αναζήτησης η λύση θα ήταν πιο απλή, αντί να διαλέγουμε κατευθείαν τον Ασθενή θα μπορούσαμε να ψάχνουμε με βάση, πχ, το επίθετό του.

Links

1. https://en.wikipedia.org/wiki/Java_Persistence_Query_Language
2. http://docs.oracle.com/cd/E15523_01/apirefs.1111/e13946/ejb3_langref.html
3. <https://maven.apache.org/>
4. <http://projects.spring.io/spring-boot/>

http://docs.oracle.com/cd/E15523_01/apirefs.1111/e13946/ejb3_langref.html

