



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Βάσεις Δεδομένων

Εξαμηνιαίο Project

Ακαδημαϊκό Έτος: 2016-2017

Μπάτζιου Ελένη (03111138)
Σιγλίδης Ιωάννης (03112069)
Τερζιδάκη Αλεξάνδρα (03112513)

Λεπτομέρειες Υλοποίησης

Τα εργαλεία που χρησιμοποιήσαμε είναι τα παρακάτω:

- Σύστημα Διαχείρισης Βάσεων Δεδομένων : MySQL
- Δημιουργία User Interface: PHP
- Διαχείριση User Interface : HTML, CSS
- Web Server : Apache

Χρησιμοποιήθηκε το πρόγραμμα XAMPP, ένα open-source εργαλείο που περιλαμβάνει τον Apache web server, τη βάση δεδομένων MySQL, υποστηρίζει τη γλώσσα PHP μέσω του συστήματος βάσης δεδομένων phpMyAdmin. Ως DataBase Storage Engine χρησιμοποιείται από το XAMPP by default η InnoDB.

Οι επιλογές μας έχουν τα εξής σημαντικά πλεονεκτήματα:

XAMPP: (+) Ένα σημαντικό πλεονέκτημα του XAMPP είναι η ευκολία στην εγκατάσταση, και το καθαρό περιβάλλον του, που το καθιστά εύχρηστο ακόμη και για αρχάριους. Επιπλέον, είναι cross-platform, που σημαίνει ότι μπορεί να χρησιμοποιηθεί με τον ίδιο τρόπο ανεξαρτήτως του λειτουργικού συστήματος του μηχανήματος. Σημαντικό πλεονέκτημα ακόμη είναι ότι είναι δωρεάν και open-source, και περιέχει όλα τα απαραίτητα εργαλεία για την δημιουργία ενός λειτουργικού web server.

(-) Ένα μειονέκτημα, το οποίο βεβαίως δεν επηρεάζει την εφαρμογή μας για τις απαιτήσεις της άσκησης, είναι η μη καταλληλότητά του για server παραγωγής, καθώς χαρακτηρίζεται από έλλειψη χαρακτηριστικών ασφαλείας.

MySQL Server : (+) Επιλέξαμε τον mySQL Server, καθώς σε αντίθεση με τον SQL Server, είναι συμβατός και με Unix συστήματα. Επιπλέον, η mySQL είναι RDBMS ανοιχτού κώδικα με ευρεία χρήση, γνωστή για την αξιοπιστία της και με ιδιαίτερα ενεργή κοινότητα ανάπτυξης και συντήρησης. Μπορούμε δηλαδή με μια γρήγορη αναζήτηση στο διαδίκτυο να διορθώσουμε πιθανά προβλήματα, βασιζόμενοι στο υπάρχον documentation, εφόσον υπάρχει πλούσιο υλικό από παραδείγματα και υποδείξεις.

(-) Η mySQL δεν υποστηρίζει τον έλεγχο για constraints. Έτσι χρειάστηκε να υλοποιηθεί ο έλεγχος μέσω της php. Η InnoDB Storage είναι κάπως αργή σε απόδοση, όταν πρόκειται να εκτελεστούν πολλές αναγνώσεις δεδομένων. Χαρακτηρίζεται από μικρή κλιμάκωση της απόδοσης, ένα πρόβλημα που εμείς δεν αντιμετωπίσαμε στην εφαρμογή μας λόγω του μικρού όγκου δεδομένων.

InnoDB Storage Engine: Η InnoDB είναι αξιόπιστη και υψηλής απόδοσης. Είναι σχεδιασμένη σύμφωνα με το μοντέλο ACID, επομένως υποστηρίζονται οι λειτουργίες commit, roll-back και crash-recovery και έτσι προστατεύονται τα δεδομένα. Οργανώνει τα δεδομένα του δίσκου σε clustered indexes, που έχει ως αποτέλεσμα να βελτιστοποιεί συχνά queries σε πρωτεύοντα κλειδιά, και ελαχιστοποιείται ο απαιτούμενος αριθμός I/Os για την προσκόμιση κλειδιών. Τέλος, υποστηρίζει Αναφορική Ακεραιότητα Κλειδιού.

phpMyAdmin: (+) Όσον αφορά στο περιβάλλον του phpMyAdmin, είναι ιδιαίτερα βολικό για την εκτέλεση εργασιών συντήρησης σε πίνακες, τη δημιουργία αντιγραφών ασφαλείας πληροφοριών και την άμεση επεξεργασία της Βάσης Δεδομένων ενώ έχει επίσης ένα πολύ μεγάλο documentation στο διαδίκτυο. Η δυνατότητα της άμεσης επεξεργασίας της Βάσης μπορεί να λειτουργήσει βέβαια και αρνητικά καθώς δεν υπάρχει η δυνατότητα του undo ή του undelete με αποτέλεσμα να μπορεί να χαλάσει μέρος της βάσης μας. Η php είναι συμβατή με την mySQL, οπότε μπορούμε να καλούμε μέσω της php, απευθείας ερωτήματα στη βάση μας. Το περιβάλλον phpMyAdmin, είναι χρήσιμο για εργασίες συντήρησης σε πίνακες, δημιουργία αντιγράφων ασφαλείας πληροφοριών και άμεση επεξεργασία της βάσης δεδομένων. Επίσης, έχει πολύ μεγάλο documentation στο διαδίκτυο.

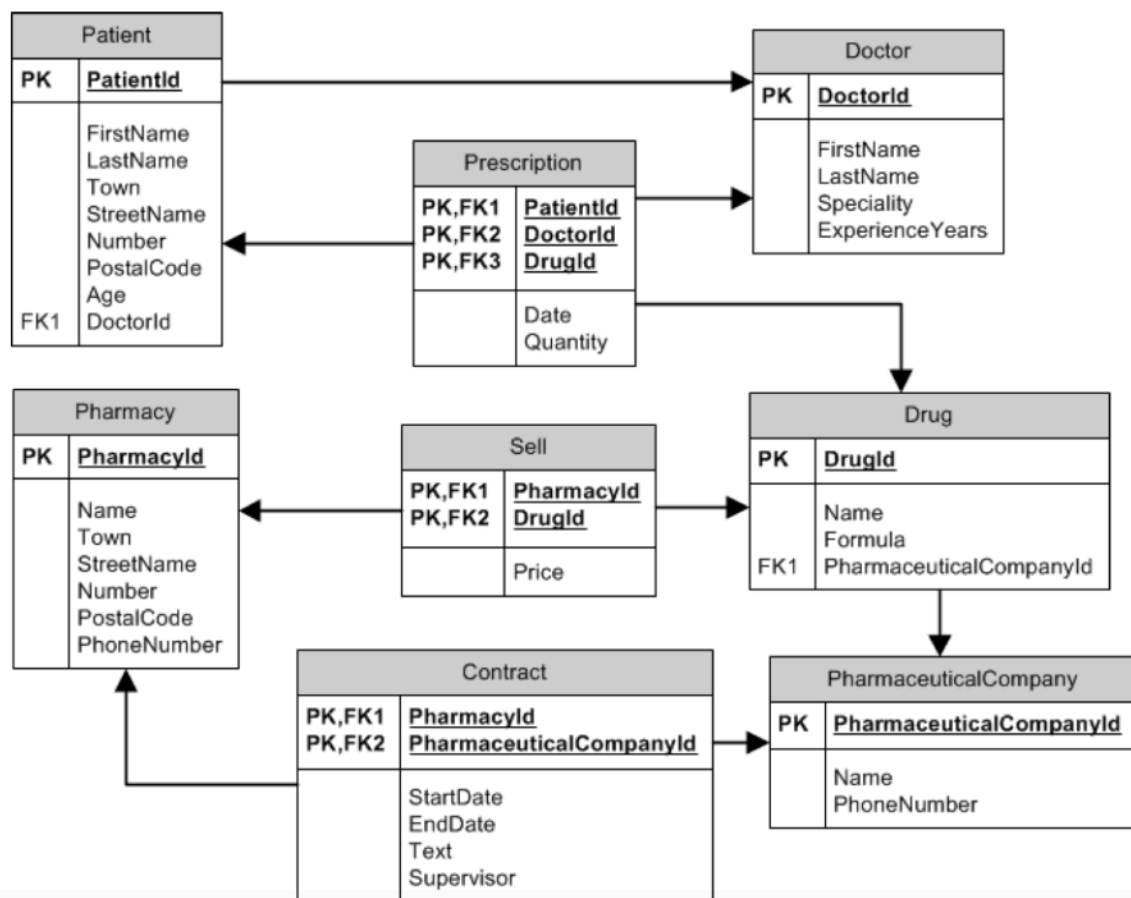
(-) Δεδομένου ότι επεξεργαζόμαστε άμεσα τη βάση, και δεν υποστηρίζεται η δυνατότητα undo ή undelete, κινδυνεύουμε να χαλάσουμε κάποιο κομμάτι της βάσης.

HTML, CSS:

Βασικές για τη δημιουργία ιστοσελίδων και τη μορφοποίηση του User Interface, με πλήρες υλικό στο διαδίκτυο και εύκολη εκμάθηση και χρήση. Τα html αρχεία έχουν κοινό css, όπου προσδιορίζεται το style καθολικά, για να υπάρχει μεγαλύτερη ανεξαρτησία και ευελιξία.

Σχεσιακό Μοντέλο

Το σχεσιακό μοντέλο της φαρμακευτικής εταιρία Perscription R-X, ακολουθήθηκε όπως έχει δοθεί και από την πρώτη άσκηση, φαίνεται παρακάτω:



Λειτουργικές Απαιτήσεις - Προδιαγραφές

Οι οντότητες και οι σχέσεις της βάσης δεδομένων υπόκεινται στους εξής περιορισμούς:

1. Έχουν τεθεί περιορισμοί ακεραιότητας, καθώς όλα τα πεδία των σχέσεων έχουν τεθεί ως NOT NULL.
2. Τα πεδία ID στις DOCTOR, DRUG, PATIENT, PHARMACY, PHARMACEUTICAL COMPANY αποτελούν τα πρωτεύοντα κλειδιά και έχουν τεθεί στη βάση ως autoincrement.
3. Για κάθε ξένο κλειδί, υπάρχει ο περιορισμός ξένου κλειδιού, που επιβάλλει να συμπληρωθεί το πεδίο ξένου κλειδιού υποχρεωτικά με υπαρκτή τιμή του πίνακα στον οποίο αποτελεί πρωτεύον κλειδί.
4. Με τη διαγραφή ενός μέλους του DOCTOR ελέγχουμε αν υπάρχει κάποιο μέλος των PRESCRIPTION και PATIENT το οποίο να είναι συνδεδεμένο με τον γιατρό με το ίδιο doctorID πεδίου. Αν υπάρχει, τότε εμφανίζεται αντίστοιχο προειδοποιητικό μήνυμα στο χρήστη, και αν επιλέξει SUBMIT τότε θα διαγραφούν και τα αντίστοιχα στοιχεία του PRESCRIPTION και του PATIENT .
5. Με τη διαγραφή ενός μέλους του DRUG, ελέγχουμε αν υπάρχει κάποιο μέλος του PRESCRIPTION και του SELL το οποίο να είναι συνδεδεμένο με το φάρμακο το ίδιο ID πεδίου. Αν υπάρχει, τότε εμφανίζεται αντίστοιχο προειδοποιητικό μήνυμα στο χρήστη, και αν επιλέξει SUBMIT τότε θα διαγραφούν και τα αντίστοιχα στοιχεία του PRESCRIPTION και SELL.
6. Με τη διαγραφή ενός μέλους του PATIENT, ελέγχουμε αν υπάρχει κάποιο μέλος του PRESCRIPTION το οποίο να είναι συνδεδεμένο με το φάρμακο το ίδιο ID πεδίου. Αν υπάρχει, τότε εμφανίζεται αντίστοιχο προειδοποιητικό μήνυμα στο χρήστη, και αν επιλέξει SUBMIT τότε θα διαγραφούν και τα αντίστοιχα στοιχεία του PRESCRIPTION.
7. Με τη διαγραφή ενός μέλους του PHARMACEUTICAL COMPANY, ελέγχουμε αν υπάρχει κάποιο μέλος του CONTRACT και του DRUG (και των SELL και PRESCRIPTION κατ'επέκταση) το οποίο να είναι συνδεδεμένο με την εταιρία με το ίδιο ID πεδίου. Αν υπάρχει, τότε εμφανίζεται αντίστοιχο προειδοποιητικό μήνυμα στο χρήστη, και αν επιλέξει SUBMIT τότε θα διαγραφούν και τα αντίστοιχα στοιχεία του CONTRACT και DRUG ,SELL,PRESRIPTION.
8. Με τη διαγραφή ενός μέλους του PHARMACY, ελέγχουμε αν υπάρχει κάποιο μέλος του CONTRACT και του SELL το οποίο να είναι συνδεδεμένο με την εταιρία με το ίδιο ID πεδίου. Αν υπάρχει, τότε εμφανίζεται αντίστοιχο προειδοποιητικό μήνυμα στο χρήστη, και αν επιλέξει SUBMIT τότε θα διαγραφούν και τα αντίστοιχα στοιχεία του CONTRACT και SELL.

DLLs

```
--  
-- Table structure for table `contract`  
--
```

```
CREATE TABLE `contract` (  
  `PharmacyId` int(11) NOT NULL,  
  `PharmaceuticalCompanyId` int(11) NOT NULL,  
  `StartDate` date NOT NULL,  
  `EndDate` date NOT NULL,  
  `Text` text NOT NULL,  
  `Supervisor` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```

--
-- Table structure for table `doctor`
--

CREATE TABLE `doctor` (
  `DoctorId` int(11) NOT NULL,
  `FirstName` varchar(255) NOT NULL,
  `LastName` varchar(255) NOT NULL,
  `Speciality` varchar(255) NOT NULL,
  `ExperienceYears` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `drug`
--

CREATE TABLE `drug` (
  `DrugId` int(11) NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Formula` varchar(255) NOT NULL,
  `PharmaceuticalCompanyId` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `patient`
--

CREATE TABLE `patient` (
  `PatientId` int(11) NOT NULL,
  `FirstName` varchar(255) NOT NULL,
  `LastName` varchar(255) NOT NULL,
  `Town` varchar(255) NOT NULL,
  `StreetName` varchar(255) NOT NULL,
  `Number` int(11) NOT NULL,
  `PostalCode` int(11) NOT NULL,
  `Age` int(11) NOT NULL,
  `DoctorId` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `pharmaceuticalcompany`
--

CREATE TABLE `pharmaceuticalcompany` (
  `PharmaceuticalCompanyId` int(11) NOT NULL,
  `Name` varchar(255) NOT NULL,
  `PhoneNumber` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```
--
-- Table structure for table `pharmacy`
--

CREATE TABLE `pharmacy` (
  `PharmacyId` int(11) NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Town` varchar(255) NOT NULL,
  `StreetName` varchar(255) NOT NULL,
  `StreetNumber` int(11) NOT NULL,
  `PostalCode` int(11) NOT NULL,
  `PhoneNumber` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `prescription`
--

CREATE TABLE `prescription` (
  `PatientId` int(11) NOT NULL,
  `DoctorId` int(11) NOT NULL,
  `DrugId` int(11) NOT NULL,
  `Date` date NOT NULL,
  `Quantity` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `sell`
--

CREATE TABLE `sell` (
  `PharmacyId` int(11) NOT NULL,
  `DrugId` int(11) NOT NULL,
  `Price` float NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Ευρετήρια

Κατασκευάσαμε πρωτεύοντα ευρετήρια με βάση το primary key της κάθε σχέσης, εφόσον αυτό είναι το πεδίο που χρησιμοποιείται περισσότερο για πράξεις update και delete.

```
--
-- Indexes for table `contract`
--

ALTER TABLE `contract`
  ADD PRIMARY KEY (`PharmacyId`,`PharmaceuticalCompanyId`),
  ADD KEY `PharmacyId` (`PharmacyId`),
  ADD KEY `PharmaceuticalCompanyId` (`PharmaceuticalCompanyId`);

--
-- Indexes for table `doctor`
--

ALTER TABLE `doctor`
  ADD PRIMARY KEY (`DoctorId`);
```

```

--
-- Indexes for table `drug`
--
ALTER TABLE `drug`
  ADD PRIMARY KEY (`DrugId`),
  ADD KEY `PharmaceuticalCompanyId` (`PharmaceuticalCompanyId`);

--
-- Indexes for table `patient`
--
ALTER TABLE `patient`
  ADD PRIMARY KEY (`PatientId`),
  ADD KEY `DoctorId` (`DoctorId`);

--
-- Indexes for table `pharmaceuticalcompany`
--
ALTER TABLE `pharmaceuticalcompany`
  ADD PRIMARY KEY (`PharmaceuticalCompanyId`);

--
-- Indexes for table `pharmacy`
--
ALTER TABLE `pharmacy`
  ADD PRIMARY KEY (`PharmacyId`);

--
-- Indexes for table `prescription`
--
ALTER TABLE `prescription`
  ADD PRIMARY KEY (`PatientId`,`DoctorId`,`DrugId`),
  ADD KEY `PatientId` (`PatientId`),
  ADD KEY `DoctorId` (`DoctorId`),
  ADD KEY `DrugId` (`DrugId`),
  ADD KEY `DoctorId_2` (`DoctorId`),
  ADD KEY `DoctorId_3` (`DoctorId`,`DrugId`);

--
-- Indexes for table `sell`
--
ALTER TABLE `sell`
  ADD PRIMARY KEY (`PharmacyId`,`DrugId`),
  ADD KEY `sell_restriction_drug` (`DrugId`),
  ADD KEY `PharmacyId` (`PharmacyId`);

```

Autoincrements

```

--
-- AUTO_INCREMENT for table `doctor`
--
ALTER TABLE `doctor`
  MODIFY `DoctorId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;

```

```

--
-- AUTO_INCREMENT for table `drug`
--
ALTER TABLE `drug`
  MODIFY `DrugId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;

--
-- AUTO_INCREMENT for table `patient`
--
ALTER TABLE `patient`
  MODIFY `PatientId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;

--
-- AUTO_INCREMENT for table `pharmaceuticalcompany`
--
ALTER TABLE `pharmaceuticalcompany`
  MODIFY `PharmaceuticalCompanyId` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `pharmacy`
--
ALTER TABLE `pharmacy`
  MODIFY `PharmacyId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

```

Constraints

```

--
-- Constraints for table `contract`
--
ALTER TABLE `contract`
  ADD CONSTRAINT `fk_contract_pharmaceutical_comp_restr` FOREIGN KEY
  (`PharmaceuticalCompanyId`) REFERENCES `pharmaceuticalcompany`
  (`PharmaceuticalCompanyId`),
  ADD CONSTRAINT `fk_contract_pharmacy_restriction` FOREIGN KEY (`PharmacyId`)
  REFERENCES `pharmacy` (`PharmacyId`);

--
-- Constraints for table `drug`
--
ALTER TABLE `drug`
  ADD CONSTRAINT `fk_restriction_dpc` FOREIGN KEY (`PharmaceuticalCompanyId`)
  REFERENCES `pharmaceuticalcompany` (`PharmaceuticalCompanyId`);

--
-- Constraints for table `patient`
--
ALTER TABLE `patient`
  ADD CONSTRAINT `fk_restriction_dp` FOREIGN KEY (`DoctorId`) REFERENCES
  `doctor` (`DoctorId`);

```



```
--
-- Constraints for table `prescription`
--
ALTER TABLE `prescription`
  ADD CONSTRAINT `perscription_doct_restriction` FOREIGN KEY (`DoctorId`)
REFERENCES `doctor` (`DoctorId`),
  ADD CONSTRAINT `perscription_drug_restriction` FOREIGN KEY (`DrugId`)
REFERENCES `drug` (`DrugId`),
  ADD CONSTRAINT `perscription_patient_restriction` FOREIGN KEY (`PatientId`)
REFERENCES `patient` (`PatientId`);

--
-- Constraints for table `sell`
--
ALTER TABLE `sell`
  ADD CONSTRAINT `sell_restriction_drug` FOREIGN KEY (`DrugId`) REFERENCES
`drug` (`DrugId`),
  ADD CONSTRAINT `sell_restriction_pharmacy` FOREIGN KEY (`PharmacyId`)
REFERENCES `pharmacy` (`PharmacyId`);
```

Insertions

```
--
-- Dumping data for table `contract`
--

INSERT INTO `contract` (`PharmacyId`, `PharmaceuticalCompanyId`, `StartDate`,
`EndDate`, `Text`, `Supervisor`) VALUES
(2, 2, '2016-10-07', '2019-08-12', 'Lorem Ipsum', 'Olga Dymov'),
(2, 3, '2016-03-14', '2017-11-09', 'Lorem Ipsum', 'Konstantin Treplyov'),
(3, 1, '2016-09-10', '2019-06-13', 'Lorem Ipsum', 'Charlotta Ivanovna'),
(3, 2, '2016-12-11', '2020-09-30', 'Lorem Ipsum', 'Misail Poloznev'),
(3, 4, '2016-09-08', '2018-07-01', 'Lorem Ipsum', 'Semyon Yepikhodov'),
(4, 1, '2016-02-18', '2019-01-02', 'Lorem Ipsum', 'Andrei Kovrin'),
(4, 2, '2016-10-31', '2019-06-20', 'Lorem Ipsum', 'Yelena Andreyevna'),
(4, 3, '2016-12-06', '2018-12-21', 'Lorem Ipsum', 'Polina Andreyevna'),
(4, 4, '2016-11-28', '2018-11-02', 'Lorem Ipsum', 'Yermolai Lopakhin'),
(5, 2, '2016-05-09', '2020-05-04', 'Lorem Ipsum', 'Marya Yefimovna Grekova'),
(5, 3, '2016-08-25', '2020-02-26', 'Lorem Ipsum', 'Dmitri Gurov'),
(5, 4, '2016-01-27', '2018-11-25', 'Lorem Ipsum', 'Fyodor Kulygin'),
(6, 1, '2016-12-07', '2018-03-29', 'Lorem Ipsum', 'Ilya Ilyich Telegin'),
(6, 2, '2016-07-23', '2019-09-29', 'Lorem Ipsum', 'Olga Plemlyannikov'),
(6, 4, '2016-06-11', '2018-10-10', 'Lorem Ipsum', 'Irina Sergeyevna');

--
-- Dumping data for table `doctor`
--

INSERT INTO `doctor` (`DoctorId`, `FirstName`, `LastName`, `Speciality`,
`ExperienceYears`) VALUES
(1, 'Katerina', 'Verkhovtsev', 'Cardiologist', 15),
(2, 'Pavel', 'Smerdyakov', 'Endocrinologist', 7),
```

```
(3, 'Liza', 'Khokhlakov', 'Gastroenterologist', 24),
(4, 'Agrafena', 'Svetlov', 'Pediatrician', 15),
(5, 'Mikhail', 'Rakitin', 'Oncologist', 12),
(6, 'Pyotr', 'Miusov', 'Radiologist', 19),
(7, 'Kuzma', 'Samsonov', 'Pulmonologist', 26),
(8, 'Grifoty', 'Vasilievich', 'Surgeon', 9);
```

```
--
```

```
-- Dumping data for table `drug`
```

```
--
```

```
INSERT INTO `drug` (`DrugId`, `Name`, `Formula`, `PharmaceuticalCompanyId`)
VALUES
(1, 'SANDOSTATIN', 'Octreotide', 1),
(2, 'DUODART', 'Tamsulosin, Dutasteride', 3),
(3, 'HYCAMTIN', 'Topotecan', 3),
(4, 'LIPITOR', 'Atorvastatin', 2),
(5, 'AMOXIL', 'Amoxicillin', 3),
(6, 'YASMIN', 'Drospirenone, Ethinyl estradiol ', 4),
(7, 'GALVUS', 'Vildagliptin', 1),
(8, 'RELENZA', 'Zanamivir', 3),
(9, 'TASIGNA', 'Nilotinib', 1),
(10, 'LYRICA', 'Pregabalin', 2),
(11, 'DIOVAN', 'Valsartan', 1),
(12, 'FLIXOTIDE', 'Fluticasone', 3),
(13, 'XARELTO', 'Rivaroxaban', 4),
(14, 'EXJADE', 'Deferasirox', 1),
(15, 'CELEBREX', 'Celecoxib', 2),
(16, 'SUTENT', 'Sunitib', 2),
(17, 'CIPROXIN', 'Ciprofloxacin', 4),
(18, 'NEXAVAR', 'Sorafenib', 4);
```

```
--
```

```
-- Dumping data for table `patient`
```

```
--
```

```
INSERT INTO `patient` (`PatientId`, `FirstName`, `LastName`, `Town`,
`StreetName`, `Number`, `PostalCode`, `Age`, `DoctorId`) VALUES
(1, 'Stepan', 'Stepanovich', 'Athens', 'Pesmatzoglou', 4, 10559, 33, 3),
(2, 'Elisabeth', 'Alexeievna', 'Athens', 'Koumpari', 1, 10674, 59, 4),
(3, 'Yakov', 'Alpatych', 'Moschato', 'Konstantinoupoleos', 91, 18345, 42, 2),
(4, 'Maria', 'Dolokhova', 'Kalambaka', 'Xatzipetrou', 24, 42200, 34, 4),
(5, 'Tikhon', 'Shtcherbatov', 'Peristeri', 'Ethnikis Antistaseos', 24, 12134,
58, 7),
(6, 'Nikolai', 'Rostov', 'Ioannina', 'Kougkiou', 35, 45221, 41, 5),
(7, 'Maria', 'Hendrihovna', 'Agios Dimitrios', 'Dilou', 43, 17342, 29, 2),
(8, 'Mikhail', 'Ivanovich', 'Polichni', 'Rodou', 21, 56532, 37, 4),
(9, 'Anna', 'Malvintsev', 'Moschato ', 'Leoforos Poseidonos', 3, 18344, 31, 3),
(10, 'Pyotr', 'Shinshin', 'Agioli Anargyroi', 'Kefallinias', 56, 13561, 53, 5),
(11, 'Boris', 'Drubetskoy', 'Neapoli', 'Kallikratias', 32, 56728, 62, 1),
(12, 'Nikolay', 'Bolkonsky', 'Kilkis', 'Aristotelous', 24, 61100, 31, 6),
(13, 'Helene', 'Kuragina', 'Argiroupoli', 'Pelloponisou', 110, 16451, 44, 1),
(14, 'Makar', 'Bazdeyev', 'Larisa', 'Gr. Lambraki', 14, 41447, 18, 7),
(15, 'Nastasya', 'Ivanovna', 'Nafplio', 'Messologiou', 5, 21100, 67, 4);
```

```
--
-- Dumping data for table `pharmacy`
--

INSERT INTO `pharmacy` (`PharmacyId`, `Name`, `Town`, `StreetName`,
`StreetNumber`, `PostalCode`, `PhoneNumber`) VALUES
(1, 'Osip Vikhrov Apoteka', 'Larisa', 'Thoukididou', 5, 41222, 2147483647),
(2, 'Agrafiya Yolkinova Apoteka', 'Ambelokipoi', 'Fillipoupoleos', 74, 56121,
2147483647),
(3, 'Anastasia Aslanova Apoteka', 'Athens', 'Xarilaou Trikoupi', 50, 10680,
2103665000),
(4, 'Petya Ismaylova Apoteka', 'Tripoli', 'Taxiarchon', 82, 22100, 2147483647),
(5, 'Rasputin Kuzubov Apoteka', 'Xanthi', 'Sarantaporou', 63, 67100,
2147483647),
(6, 'Rada Zykov Apoteka', 'Metsovo', 'Tositsa', 26, 44200, 2147483647);
```

```
--
-- Dumping data for table `prescription`
--

INSERT INTO `prescription` (`PatientId`, `DoctorId`, `DrugId`, `Date`,
`Quantity`) VALUES
(1, 4, 8, '2016-01-27', 2),
(2, 6, 15, '2016-09-12', 4),
(4, 1, 4, '2016-06-22', 2),
(5, 4, 10, '2016-10-21', 4),
(6, 4, 10, '2016-09-02', 1),
(7, 5, 12, '2016-06-20', 3),
(10, 1, 8, '2016-05-18', 3),
(11, 5, 14, '2016-09-08', 2),
(12, 6, 4, '2016-12-27', 3),
(13, 4, 3, '2016-02-11', 3),
(13, 5, 1, '2016-10-14', 4),
(15, 6, 1, '2016-06-29', 4);
```

```
--
-- Dumping data for table `sell`
--

INSERT INTO `sell` (`PharmacyId`, `DrugId`, `Price`) VALUES
(1, 1, 50),
(1, 2, 34),
(1, 4, 20),
(2, 1, 44),
(2, 7, 7),
(2, 8, 24),
(2, 11, 1),
(3, 2, 13),
(3, 3, 33),
(3, 5, 1),
(3, 6, 25),
(3, 7, 38),
(3, 9, 5),
(3, 11, 13),
(4, 3, 29);
```

Triggers

Τα triggers σε μια βάση δεδομένων εξασφαλίζουν την εσωτερική συνέπειά της. Κατασκευάσαμε τα ακόλουθα triggers για την περίπτωσή μας:

- Το παρακάτω trigger ελέγχει πριν από κάθε εισαγωγή στις συνταγές αν ένας ασθενής συνταγογραφείται από τον παρόντα γιατρό. Αν όχι, επιστρέφει μήνυμα λάθους και δεν επιτρέπει την πραγματοποίηση της εισαγωγής. Το συγκεκριμένο trigger προέκυψε από την περιγραφή της βάσης στην Άσκηση 1, όπου σε δύο σημεία αναγράφεται το εξής: "Ένας ασθενής μπορεί να συνταγογραφείται μόνο από τον παρόντα γιατρό του"

```
CREATE TRIGGER `current_prescription_valid_doctor_trigger` BEFORE INSERT ON
`prescription`
FOR EACH ROW
BEGIN
IF ((SELECT DoctorId FROM `patient` WHERE PatientId=NEW.PatientId) <>
NEW.DoctorId) THEN
    SIGNAL SQLSTATE '45000' set message_text = 'Error: A patient can be
perscripted only by his current doctor!';
END IF;
END
```

- Το επόμενο trigger διατηρεί μία ομοιόμορφη κατανομή μεταξύ ασθενών και γιατρών. Αν γίνει μία εισαγωγή στη βάση ενός ασθενή σε ένα γιατρό ο οποίος έχει το μέγιστο για τη βάση πλήθος ασθενών και η διαφορά μεγίστου και ελαχίστου είναι μεγαλύτερη από 2 (επιλέχθηκε για να προσομοιώνει μία δυνατότητα σχετικής προτίμησης), τότε μεταβάλλουμε τον γιατρό ο οποίος ορίζεται για να συνταγογραφεί αυτόν τον ασθενή σε αυτόν με το ελάχιστο πλήθος ασθενών. Μετά από πολλές εισαγωγές (αν θεωρήσουμε την αρχικοποίηση της βάσης πριν το trigger) όλοι οι γιατροί θα έχουν το ίδιο πλήθος ασθενών, με απόσταση το πολύ δύο (2) ασθενείς στο πλήθος.

```
CREATE TRIGGER `doctors_patient_distribution_trigger` BEFORE INSERT ON `patient`
FOR EACH ROW
BEGIN
DECLARE max integer;
DECLARE min integer;
set @max := (SELECT COUNT(DoctorId) AS NumOfPatients FROM `patient` GROUP BY
DoctorId ORDER BY COUNT(DoctorId) DESC LIMIT 1);
set @min := (SELECT COUNT(DoctorId) AS NumOfPatients FROM `patient` GROUP BY
DoctorId ORDER BY COUNT(DoctorId) ASC LIMIT 1);
IF( (@max - @min >= 3) AND ((SELECT DoctorId AS NumOfPatients FROM `patient`
GROUP BY DoctorId ORDER BY COUNT(DoctorId) DESC LIMIT 1) = NEW.DoctorId )) THEN
    set NEW.DoctorId = (SELECT DoctorId AS NumOfPatients FROM `patient` GROUP
BY DoctorId ORDER BY COUNT(DoctorId) ASC LIMIT 1);
END IF;
END
```

Views

Κατασκευάσαμε δύο όψεις: μια ενημερώσιμη και μια μη ενημερώσιμη.

Ενημερώσιμη Όψη

Κατασκευάσαμε μια όψη που περιέχει τους γιατρούς που είναι καταχωρημένοι στη βάση με περισσότερα από 15 χρόνια εμπειρίας, με τα αντίστοιχα πεδία ονόματος, ID, ειδικότητα και έτη εμπειρίας. Η ενημερώσιμη όψη θα πρέπει να περιλαμβάνει στοιχεία από έναν μόνο πίνακα, ώστε να μπορούν να υλοποιηθούν οι ενέργειες insert, update και delete δίχως πρόβλημα.

```
CREATE VIEW updateable AS
SELECT FirstName, LastName, DoctorId, Speciality, ExperienceYears
FROM doctor
WHERE ExperienceYears > 15
```

Μη ενημερώσιμη Όψη

Η μη ενημερώσιμη όψη που κατασκευάσαμε παρουσιάζει την μέση τιμή κάθε φαρμάκου. Η χρήση των aggregate δεν επιτρέπει την ενημέρωση της βάσης μέσω της όψης.

```
CREATE
ALGORITHM = UNDEFINED
VIEW `avgPrice`
(DrugId, DrugName, AvgPrice)
AS SELECT Dr.DrugId, Dr.Name, ROUND(AVG(S.Price), 2)
FROM sell S, drug Dr
WHERE S.DrugId=Dr.DrugId
GROUP BY S.DrugId
```

SQL QUERIES

Δίνεται στον χρήστη η δυνατότητα να επιλέξει ανάμεσα στα παρακάτω ερωτήματα:

- Φαρμακείο και συμβόλαιο με ημερομηνία λήξης τον μήνα Μάρτιο

```
SELECT P.Name, C.EndDate, C.Supervisor
FROM pharmacy P JOIN contract C ON P.PharmacyId=C.PharmacyId
WHERE C.EndDate LIKE '2017-03-%'
```

- Όνομα γιατρού και φαρμακευτικής εταιρίας με την οποία συνδέεται

```
SELECT DISTINCT D.FirstName, D.LastName, C.Name
FROM doctor D, prescription P, drug Dr, pharmaceuticalcompany C
WHERE D.DoctorId=P.DoctorId AND P.DrugId=Dr.DrugId AND
Dr.PharmaceuticalCompanyId=C.PharmaceuticalCompanyId
```

- Ταξινόμηση των φαρμάκων κατά τιμή

```
SELECT D.Name,MIN(S.Price),P.Name
FROM pharmacy P, drug D, sell S
WHERE P.PharmacyId=S.PharmacyId AND D.DrugId=S.DrugId
GROUP BY D.Name
ORDER BY S.Price
```

- Ταξινόμηση των φαρμακείων κατά αριθμό πωλήσεων

```
SELECT S.PharmacyId,P.Name, SUM(S.Price) as price
FROM sell S JOIN Pharmacy P ON S.PharmacyId=P.PharmacyId
GROUP BY S.PharmacyId
ORDER BY price DESC,S.PharmacyId
```

- Αριθμός ασθενών που αντιστοιχούν σε κάθε γιατρό, ταξινομημένα κατά ID γιατρού

```
SELECT D.FirstName, D.LastName, Count(P.DoctorId) as Count
FROM patient P, doctor D
WHERE (P.DoctorId=D.DoctorId)
GROUP BY D.DoctorId
```

- Ονόματα φαρμάκων με τιμή μεγαλύτερη του μέσου όρου

```
SELECT DISTINCT d.Name, s.Price
FROM sell s, drug d
WHERE s.Price > (select avg(Price) from sell) and (s.DrugId=d.DrugId)
GROUP BY d.Name
```

- Ονόματα γιατρών με πάνω από 5 έτη εμπειρίας και ασθενείς από την πόλη Αθήνα

```
SELECT DISTINCT d.FirstName, d.LastName
FROM doctor d
JOIN patient p
ON p.DoctorId=d.DoctorId
WHERE d.ExperienceYears > 5 and p.Town='Athens'
```

- Γιατροί με 2 ή περισσότερους ασθενείς

```
SELECT doctor.FirstName, doctor.LastName, COUNT(*) AS \"NumberofPatients\"
FROM (patient INNER JOIN doctor ON patient.DoctorId=doctor.DoctorId)
GROUP BY doctor.FirstName, doctor.LastName having COUNT(*)>=2;
```

- Ονόματα φαρμάκων που δεν διατίθενται προς πώληση

```
SELECT Name AS \"Drug Name\"
FROM drug
WHERE (NOT EXISTS (SELECT * FROM sell WHERE drug.DrugId=sell.DrugId))
```

- Ονόματα γιατρών και ασθενών των οποίων το όνομα περιέχει τη συλλαβή “nov”

```
(SELECT LastName AS \"Last Name\" FROM patient WHERE LastName LIKE '%nov%')
UNION
(SELECT LastName AS \"Last Name\" FROM doctor WHERE LastName LIKE '%nov%')
```