CS306 Database Management

Project Pharmacy

Authors:

Azra Selimović 160302176

Hana Hadžo 140302057

Melika Hodžić 150302105

Professor:

Gürsoy Osman

Contents

CS306 Project – Pharmacy

Gathering Data Items

When starting a pharmacy company, it is necessary to have some program with a database where all the products will be held. Pharmacy has a lot of drugs that should be sorted by some characteristics they share or should be totally separated so they don't mix up and cause some problems. Also it is important to be up to date with all that information easily, mostly because of the documentations and invoices.

We had an interview with an owner of Čoga pharmacy who stated his requirements as follows:

- Database should store all our products
- Each product should also have its own id or barcode and expiry date
- Database has to have the current number of each drug in our repository
- Drugs should be separated by their features
- Pills, syrups, medical supplies, etc. should also be separated
- Each drug has its price by which it will be sold
- There is also a tax for each drug
- Database should contain the vendors from whom the product is bought and the price of that sell
- Prescription drugs should be separated
- Database should contain the doctors who prescribe the drugs
- Database should contain all employees by their names, id and profession
- Database should store which employee sold which drug

Entities, attributes and restrictions:

1. Items (itemName, typeName **(FK)**, expiryDate, **barcode** **(PK)**, sellingPrice **(FK)**, availability, description **(FK)**, supplierName **(FK)**)
2. Bill (**billID** **(PK),** itemName **(FK)**, quantity, customerID **(FK)**, sellingPrice **(FK)**, frequency, employeeID **(FK)**)
3. Employees (fName, lName, DOB, position, **employeeID** **(PK)**, salary, sex)
4. Stock (**stockID(PK)**,barcode **(FK)**,expiryDate, noOfItems, availability)
5. Prescription (**prescriptionID** **(PK),** itemName **(FK)**, doctorID **(FK)**, frequency, quantity, customerID **(FK)**)
6. Doctor (**doctorID** **(PK)**, docFName, docLName)
7. Drug (itemName **(FK)**, constraints, illness **(FK)**)
8. Prices (purchasePrice **(FK)**, sellingPrice**(FK)**, profit, tax, barecode **(FK)**)
9. Supplier (**supplierName** **(PK)**, supplierID)
10. Type (typeCode, **typeName** (**PK**), barcode)
11. Customer (**customerID** **(PK)**, custFNAme, custLName, email **(FK)**, illness **(FK)**, sex)
12. Purchase (supplierName**(FK)**,barcode**(FK)**, noOfItems, **purchasePrice** **(PK)**, tax, totalPaid)
13. Disorder (**illness** **(PK)**, doctorID (**FK**))

Relationships:

Stock HOLDS Items

Items HAVE Price

Items HAVE Type

Items ARE ON Purchase

Supplier PROVIDES Items

Doctor PRESCRIBES Prescription

Doctor DIAGNOSE Disorder

Customer GETS Prescription

Customer PAYS Bill

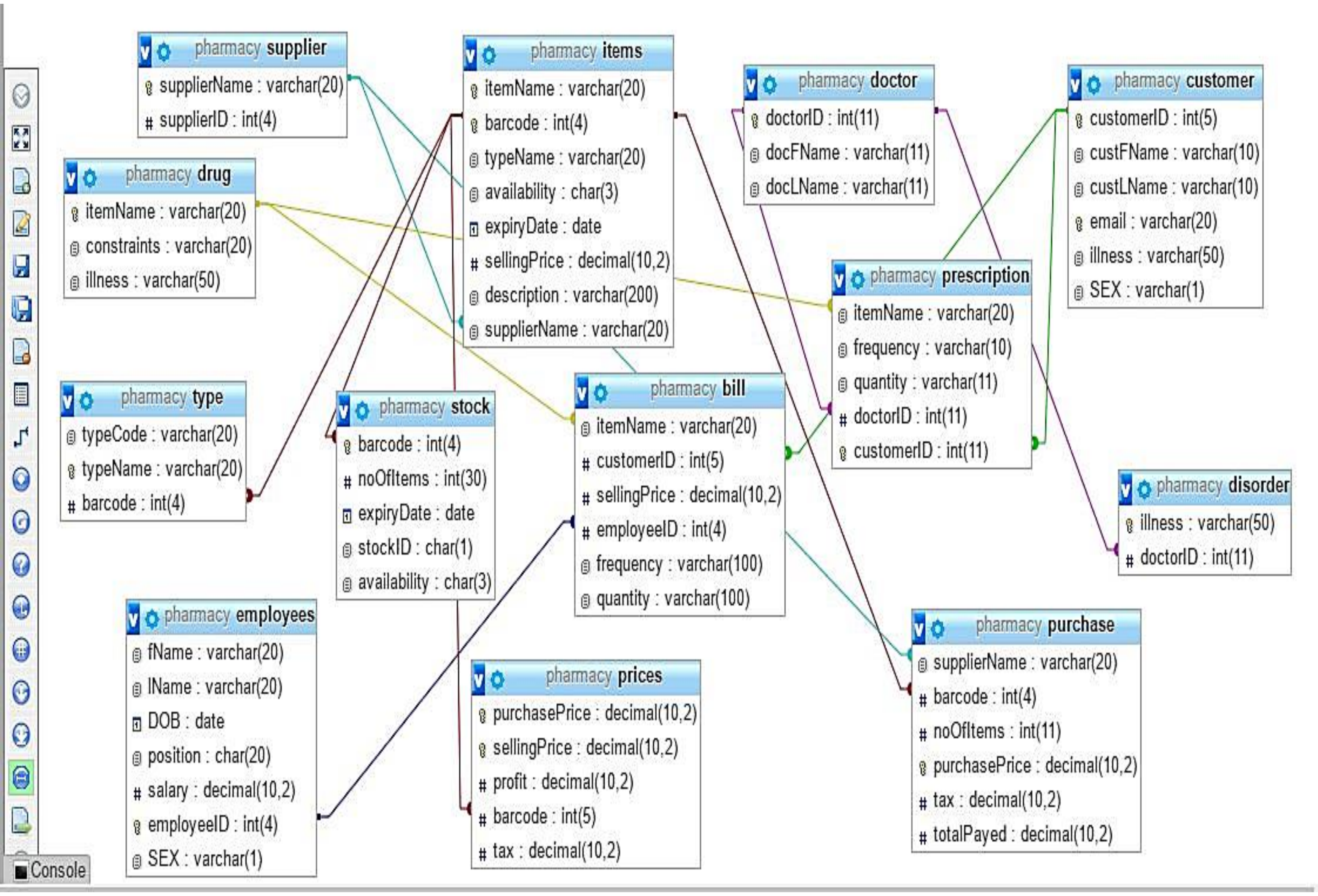
Prescription CONTAINS Drug / Drug IS PRESCRIBED BY Prescription

Bill CONTAINS Drug

Employees ENSURE Bill

Purchase IS NOTIFIED BY Bill

ER model



pharmacy supplier
supplierName : varchar(20)
supplierID : int(4)

pharmacy items
itemName : varchar(20)
barcode : int(4)
typeName : varchar(20)
availability : char(3)
expiryDate : date
sellingPrice : decimal(10,2)
description : varchar(200)
supplierName : varchar(20)

pharmacy doctor
doctorID : int(11)
docFName : varchar(11)
docLName : varchar(11)

pharmacy customer
customerID : int(5)
custFName : varchar(10)
custLName : varchar(10)
email : varchar(20)
illness : varchar(50)
SEX : varchar(1)

pharmacy drug
itemName : varchar(20)
constraints : varchar(20)
illness : varchar(50)

pharmacy prescription
itemName : varchar(20)
frequency : varchar(10)
quantity : varchar(11)
doctorID : int(11)
customerID : int(11)

pharmacy type
typeCode : varchar(20)
typeName : varchar(20)
barcode : int(4)

pharmacy stock
barcode : int(4)
noOfItems : int(30)
expiryDate : date
stockID : char(1)
availability : char(3)

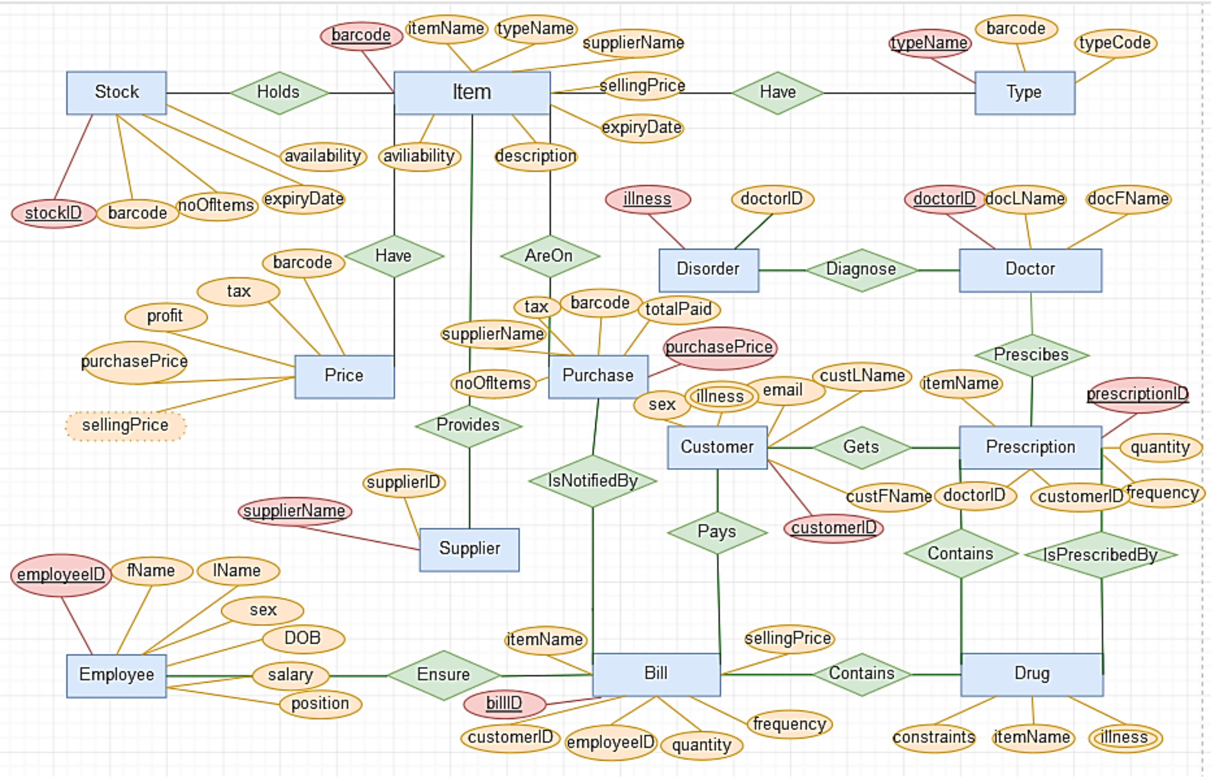
pharmacy bill
itemName : varchar(20)
customerID : int(5)
sellingPrice : decimal(10,2)
employeeID : int(4)
frequency : varchar(100)
quantity : varchar(100)

pharmacy disorder
illness : varchar(50)
doctorID : int(11)

pharmacy employees
fName : varchar(20)
lName : varchar(20)
DOB : date
position : char(20)
salary : decimal(10,2)
employeeID : int(4)
SEX : varchar(1)

pharmacy prices
purchasePrice : decimal(10,2)
sellingPrice : decimal(10,2)
profit : decimal(10,2)
barcode : int(5)
tax : decimal(10,2)

pharmacy purchase
supplierName : varchar(20)
barcode : int(4)
noOfItems : int(11)
purchasePrice : decimal(10,2)
tax : decimal(10,2)
totalPayed : decimal(10,2)



ER diagram redrawn using the Chen notation

We have redrawn UML model into CHEN notation. Because of easier recognition, we pointed relationship names, entities and attributes in different colors. We made an effort to make it as clear as possible. Relationships are usually put from right to left or from top to bottom. There is shown a distinct relationship : ‘Prescription Contains Drug’ but also ‘Drug IsPrescribedBy Prescription’. We also recognized different kinds of attributes. Primary key attributed is painted into red color, underlined, recording to rule. Multivalued attributed is illness since there are many kind of that. Also, we found derived attribute. It is sellingPrice since it can be calculate by purchasePrice, profit and tax.

Normalization

1NF

Bill table: We can distinguish itemName with sellingPrice from whole bill table and the previous table will have customerID, employeeID, frequency and quantity. But that would not be bill table anymore. The first one we can call item table and the second one undefined. But we can define dependency.

itemName → sellingPrice (sellingPrice depends on itemName)

Customer table:

1NF

customerID → custFName, custLName, email

We can define Person table here by separating custFName, custLName, email, SEX and customerID from the whole customer table. That means we are leaving illness column, but what is pharmacy customer without something that will help its health?

Disorder table:

It is already normalized, we cannot normalize it further.

doctorID → illness

Doctor table:

It cannot be normalized further.

doctorID → docFName, docLName

Drug table:

It cannot be normalized further.

itemName → constraints, illness

Employees table:

1NF

We can create two relations from this table. Person table: fName, lName, DOB, SEX and Employee table: employeeID, salary and position.

employeeID → fName, lName, DOB, SEX, salary, position

salary → position

position → salary

Items table:

We can create two tables from here: First one is item relation: itemName and barcode; and the second one is details relation: typeName, availability, expiryDate, sellingPrice, description and supplierName.

barcode → itemName, description

itemName → barcode, description

itemName, barcode → sellingPrice

Prescription table:

Every prescription needs a drug, customer name, doctor signature and the way of consummation.

customerID → frequency, quantity

doctorID → itemName

Prices table:

barcode → tax

barcode → sellingPrice, purchasePrice

tax → profit

sellingPrice, purchasePrice,tax → profit

Purchase table:

supplierName, noOfItems, purchasePrice → totalPaid

tax → purchasePrice, totalPaid

Stock table:

It is already normalized, we cannot normalize it further.

barecode → noOfItems, expiryDate, stockID, availability

Supplier table:

It is already normalized, we cannot normalize it further.

supplierID → supplierName

supplierName → supplierID

Type table:

2NF:

typeCode, typeName → barecode

typeCode → typeName

typeName → typeCode

Implementation

```
CREATE DATABASE IF NOT EXISTS `pharmacy` DEFAULT CHARACTER SET
utf8 COLLATE utf8_unicode_ci;
USE `pharmacy`;
```

Table structure for table `items`

```
DROP TABLE IF EXISTS `items`;
CREATE TABLE `items` (
  `itemName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
  `barcode` int(4) NOT NULL,
  `typeName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
  `availability` char(3) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL
DEFAULT 'no'
  `expiryDate` date NOT NULL,
  `sellingPrice` decimal(10,2) NOT NULL,
  `description` varchar(200) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
  `supplierName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `items`

```
INSERT INTO `items` (`itemName`, `barcode`, `typeName`, `availability`, `expiryDate`,
`sellingPrice`, `description`, `supplierName`)
VALUES ('Andol', 1, 'pills', 'yes', '2020-03-02', '3.60', 'pain killer', 'pharmamed'),
('Aspirin', 4, 'pills', 'yes', '2013-03-04', '2.00', 'reduce fever', 'pharmamed'),
('Brufen', 5, 'pills', 'yes', '2038-06-01', '6.80', 'pain killer', 'bosnalijek'),
('Green Slim', 6, 'syrup', 'yes', '2011-01-01', '6.00', 'dietetic product', 'bosnalijek'),
('Natal Complex', 7, 'pills', 'yes', '2019-01-05', '7.40', 'Multivitamin formulation for pregnant
women, breastfeeding women and women planning pregnancy ', 'games'),
('Alpenkraft ', 8, 'syrup', 'yes', '2023-09-08', '7.00', 'For throat, bronchi and coughning',
'games'),
('Argan oil cream', 9, 'cream', 'yes', '2029-05-05', '11.00', 'cream for hands and nails for every
skin type', 'vita'),
('Kondrovit complex ', 10, 'gel', 'yes', '2024-05-03', '13.00', 'Help with artheritis, joint and
muscle pain', 'vita'),
('Baby tea', 11, 'tea', 'yes', '2019-11-12', '5.00', 'It smoothes the cramps in the digestive tract,
for babies', 'pharmamed'),
('Tea 19', 12, 'tea', 'no', '2017-09-08', '5.00', 'helps regulate blood sugar levels', 'vita'),
('AleriX', 13, 'pills', 'no', '2021-03-02', '7.00', 'To suppress allergies ', 'bosnalijek'),
('first aid kid ', 14, 'medical supply', 'yes', '2023-06-07', '25.00', 'first aid kid', 'games'),
('Oton blood pressure ', 15, 'medical suply', 'yes', '2018-05-02', '7.00', 'for pressure
measurement', 'vita'),
('Nittlive gloves', 16, 'medical supply', 'yes', '2028-01-12', '0.50', 'wide use', 'games'),
('Digital Thermometer', 17, 'medical supply', 'yes', '2020-12-12', '9.00', 'for body temperature
measurement', 'games'),
('promensil', 18, 'pills', 'yes', '2013-06-08', '4.00', 'For women during and after menopause',
'bosnalijek'),
('Breath right', 19, 'strips', 'no', '2020-03-02', '13.00', 'sleeping aid', 'vita'),
('Alli ', 20, 'pills', 'yes', '2019-12-09', '6.75', 'dietetic product, dietary supplements for weight
loss', 'games'),
('Yokebe', 21, 'tea', 'no', '2024-07-02', '5.00', 'dietetic product, dietary supplements for weight
loss', 'vita'),
('Crowes Cremine', 22, 'cream', 'no', '2018-04-04', '8.00', 'make up remover, cleanser and
moistourize', 'games'),
('Vichy Lifactiv', 23, 'cream', 'yes', '2018-04-04', '4.50', 'day cream for dry skin', 'games'),
('Lamberts Vitamin C', 24, 'pills', 'no', '2018-04-04', '3.00', 'oral intake due to lack of vitamin
C ', 'vita');
```

Table structure for table `bill`

```
DROP TABLE IF EXISTS `bill`;
CREATE TABLE `bill` (
  `itemName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
  `customerID` int(5) NOT NULL,
  `sellingPrice` decimal(10,2) NOT NULL,
  `employeeID` int(4) NOT NULL,
  `frequency` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
  `quantity` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL
) ENGINE=InnoDB DEFAULT CHARSET= utf8;
```

Dumping data for table `bill`

```
INSERT INTO `bill` (`itemName`, `customerID`, `sellingPrice`, `employeeID`, `frequency`,
`quantity`) VALUES ('Kondrovit complex ', 3, '13.00', 1, '2', '2'),
 ('Baby tea', 2, '5.00', 3, '1', '0'),
 ('Digital Thermometer', 1, '9.00', 4, '1', '1');
```

Table structure for table `customer`

```
DROP TABLE IF EXISTS `customer`;
CREATE TABLE `customer` (
  `customerID` int(5) NOT NULL,
  `custFName` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `custLName` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `illness` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `SEX` varchar(1) NOT NULL DEFAULT 'F'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `customer`

```
INSERT INTO `customer` (`customerID`, `custFName`, `custLName`, `email`, `illness`, `SEX`)
VALUES (1, 'Ahmed', 'Hodžić', 'ahmed.h@outook.com', 'stomachache', 'M'),
 (2, 'Faris', 'Kahrić', 'faris_k@gmail.com', 'high\\blood p', 'M'),
 (3, 'Dženan', 'Kasumović', 'dy.kasum@gmail.com', 'obesity', 'M'),
 (4, 'Alma', 'Mehić', 'a.mehic@live.com', 'allergies', 'F'),
 (5, 'Sara', 'Latić', 'sara11@outlook.com', 'diabetes', 'F');
```

Table structure for table `disorder

```
DROP TABLE IF EXISTS `disorder`;
CREATE TABLE `disorder` (
  `illness` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `doctorID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `disorder`

```
INSERT INTO `disorder` (`illness`, `doctorID`)
 VALUES ('acne', 1234),
('allergies', 1234),
('bronchitis', 1234),
('cold', 1234),
('headache', 1234),
('high\\low blood p', 1234),
('insomnia', 1234),
('menopause', 1234),
('stomachache', 1234),
('toothache', 1234),
('artheritis', 5678),
('diabetes', 5678),
('obesity', 5678);
```

Table structure for table `doctor`

```
DROP TABLE IF EXISTS `doctor`;
CREATE TABLE `doctor` (
  `doctorID` int(11) NOT NULL,
  `docFName` varchar(11) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `docLName` varchar(11) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `doctor`

```
INSERT INTO `doctor` (`doctorID`, `docFName`, `docLName`)
VALUES (1234, 'john', 'wayne'),
        (5678, 'britney', 'spears');
```

Table structure for table `drug`

```
DROP TABLE IF EXISTS `drug`;
CREATE TABLE `drug` (
  `itemName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `constraints` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `illness` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `drug`

```
INSERT INTO `drug` (`itemName`, `constraints`, `illness`) VALUES
('AleriX', '', 'allergies'),
('Alli', 'not for kids', 'obesity'),
('Alpenkraft', '', 'bronchitis'),
('Argan oil cream', '', ''),
('Aspirin', '', 'cold'),
('Baby tea', 'for babies only', ''),
('Breath right', 'not for kids', 'insomnia'),
('Brufen', 'not for kids', 'stomachache'),
('Crowes Cremine', '', ''),
('Digital Thermometer', '', ''),
('first aid kit', '', ''),
('Green Slim', 'not for kids', 'obesity'),
('Kondrovit complex', 'for older than 40', 'artheritis'),
('Lamberts Vitamin C', '', ''),
('Natal Complex', 'for pregnant women', ''),
('Nittlive gloves', '', ''),
('Oton blood preasure', 'not for kids', 'high\\low blood p'),
('promensil', 'for women after 40', 'menopause'),
('Tea 19', 'not for kids', 'high\\low blood p'),
('Vichy Lifactive', '', 'acne'),
('Yokebe', 'not for kids', 'obesity');
```

Table structure for table `employees`

```
DROP TABLE IF EXISTS `employees`;
CREATE TABLE `employees` (
  `fName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `lName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `DOB` date NOT NULL,
  `position` char(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `salary` decimal(10,2) NOT NULL,
  `employeeID` int(4) NOT NULL,
  `SEX` varchar(1) NOT NULL DEFAULT 'F'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `employees`

```
INSERT INTO `employees` (`fName`, `lName`, `DOB`, `position`, `salary`, `employeeID`,
`SEX`)
 VALUES ('Nirmela', 'Hasić', '1993-02-05', 'seller', '1000.00', 1, 'F'),
            ('Adela ', 'Hadzic', '1990-07-02', 'magister', '1400.00', 2, 'F'),
  ('Amra ', 'Muminovic', '1989-06-03', 'seller', '1000.00', 3, 'F'),
  ('Ajla', 'Čogić', '1987-08-01', 'pharmasist', '2000.00', 4, 'F');
```

Table structure for table `prescription`

```
DROP TABLE IF EXISTS `prescription`;
CREATE TABLE `prescription` (
   `itemName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
   `frequency` varchar(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
   `quantity` varchar(11) CHARACTER SET utf32 COLLATE utf32_unicode_ci NOT
NULL,
  `doctorID` int(11) NOT NULL,
  `customerID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `prescription`

```
INSERT INTO `prescription` (`itemName`, `frequency`, `quantity`, `doctorID`, `customerID`)
VALUES ('Digital Thermometer', '1', '1', 5678, 2),
 ('Kondrovit complex', '2', '2', 1234, 3),
 ('Baby tea', '1', '0', 5678, 5);
```

Table structure for table `prices`

```
DROP TABLE IF EXISTS `prices`;
CREATE TABLE `prices` (
  `purchasePrice` decimal(10,2) NOT NULL,
  `sellingPrice` decimal(10,2) NOT NULL,
  `profit` decimal(10,2) NOT NULL,
  `barcode` int(5) NOT NULL,
  `tax` decimal(10,2) NOT NULL DEFAULT '0.17'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `prices`

```
INSERT INTO `prices` (`purchasePrice`, `sellingPrice`, `profit`, `barcode`, `tax`)
VALUES ('2.00', '3.20', '0.54', 4, '0.17'),
 ('2.50', '3.60', '0.61', 1, '0.17'),
            ('4.90', '6.00', '1.02', 6, '0.17');
```

Table structure for table `purchase`

```
DROP TABLE IF EXISTS `purchase`;
CREATE TABLE `purchase` (
  `supplierName` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `barcode` int(4) NOT NULL,
  `noOfItems` int(11) NOT NULL,
  `purchasePrice` decimal(10,2) NOT NULL,
  `tax` decimal(10,2) NOT NULL DEFAULT '0.17',
  `totalPaid` decimal(10,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ROW_FORMAT=COMPACT;
```

Dumping data for table `purchase`

```
INSERT INTO `purchase` (`supplierName`, `barcode`, `noOfItems`, `purchasePrice`, `tax`, `totalPaid`) VALUES ('pharmamed', 4, 45, '2.00', '0.17', '90.00'),
('pharmamed', 1, 40, '2.50', '0.17', '120.00');
```

Table structure for table `stock`

```
DROP TABLE IF EXISTS `stock`;
CREATE TABLE `stock` (
  `barcode` int(4) NOT NULL,
  `noOfItems` int(30) NOT NULL,
  `expiryDate` date NOT NULL,
  `stockID` char(1) NOT NULL DEFAULT 'A',
  `availability` char(3) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL
DEFAULT 'no'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `stock`

```
INSERT INTO `stock` (`barcode`, `noOfItems`, `expiryDate`, `stockID`, `availability`)
 VALUES (1, 29, '2020-03-02', 'A', 'yes'),
(4, 36, '2013-03-04', 'A', 'yes'),
(5, 40, '2038-06-01', 'A', 'yes'),
(6, 50, '2011-01-01', 'A', 'yes'),
(7, 36, '2019-01-05', 'A', 'yes'),
(8, 45, '2023-09-08', 'A', 'yes'),
(9, 55, '2029-05-05', 'A', 'yes'),
(10, 28, '2024-05-03', 'A', 'yes'),
(11, 36, '2019-11-12', 'A', 'yes'),
(12, 40, '2017-09-08', 'A', 'yes'),
(13, 55, '2021-03-02', 'A', 'yes'),
(14, 20, '2023-06-07', 'A', 'yes'),
(15, 25, '2018-05-02', 'A', 'yes'),
(16, 100, '2028-01-12', 'A', 'yes'),
(17, 28, '2020-12-12', 'A', 'yes'),
(18, 65, '2013-06-08', 'A', 'yes'),
(19, 55, '2020-03-02', 'A', 'yes'),
(20, 60, '2019-12-09', 'A', 'yes'),
(21, 45, '2024-07-02', 'A', 'yes'),
(22, 30, '2018-04-04', 'A', 'yes'),
(23, 25, '2018-04-04', 'A', 'yes'),
(24, 50, '2018-04-04', 'A', 'yes');
```

Table structure for table `supplier`

```
DROP TABLE IF EXISTS `supplier`;
CREATE TABLE `supplier` (
  `supplierName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
  `supplierID` int(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `supplier`

```
INSERT INTO `supplier` (`supplierName`, `supplierID`)
VALUES ('bosnalijek', 3333),
('games', 4444),
('pharmamed', 1111),
('vita', 2222);
```

Table structure for table `type`

```
DROP TABLE IF EXISTS `type`;
CREATE TABLE `type` (
  `typeCode` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `typeName` varchar(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `barcode` int(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Dumping data for table `type`

```
INSERT INTO `type` (`typeCode`, `typeName`, `barcode`)
VALUES ('724', 'cream', 23),
('415', 'gel', 10),
('321', 'medical supply', 15),
('100', 'pill', 13),
('234', 'strips', 19),
('121', 'syrup', 19),
('213', 'tea', 21);
```

Indexes for table `bill`

```
 ALTER TABLE `bill`
 ADD KEY `itemName` (`itemName`,`customerID`,`sellingPrice`,`employeeID`),
 ADD KEY `customerID` (`customerID`),
 ADD KEY `employeeID` (`employeeID`);
```

Indexes for table `customer`

```
ALTER TABLE `customer`
```

```
  ADD PRIMARY KEY (`customerID`),
  ADD UNIQUE KEY `customerID_2` (`customerID`),
  ADD UNIQUE KEY `email` (`email`),
  ADD KEY `customerID` (`customerID`),
  ADD KEY `illness` (`illness`);
```

Indexes for table `disorder`

```
ALTER TABLE `disorder`
  ADD PRIMARY KEY (`illness`),
  ADD KEY `doctorID` (`doctorID`);
```

Indexes for table `doctor`

```
ALTER TABLE `doctor`
  ADD PRIMARY KEY (`doctorID`),
  ADD UNIQUE KEY `doctorID` (`doctorID`);
```

Indexes for table `drug`

```
ALTER TABLE `drug`
  ADD PRIMARY KEY (`itemName`),
  ADD UNIQUE KEY `itemName` (`itemName`),
  ADD KEY `illness` (`illness`);
```

Indexes for table `employees`

```
ALTER TABLE `employees`
  ADD PRIMARY KEY (`employeeID`),
  ADD UNIQUE KEY `employeeID` (`employeeID`);
```

Indexes for table `items`

```
ALTER TABLE `items`
  ADD PRIMARY KEY (`barcode`),
  ADD UNIQUE KEY `itemName` (`itemName`),
  ADD UNIQUE KEY `barcode` (`barcode`),
  ADD KEY `typeCode` (`typeName`,`sellingPrice`,`description`,`supplierName`),
  ADD KEY `supplierName` (`supplierName`),
  ADD KEY `itemName_2` (`itemName`);
```

Indexes for table `prescription`

```
ALTER TABLE `prescription`
```

```
  ADD UNIQUE KEY `customerID` (`customerID`),
  ADD KEY `itemName` (`itemName`),
  ADD KEY `doctorID` (`doctorID`,`customerID`);
```

Indexes for table `prices`

```
ALTER TABLE `prices`
  ADD UNIQUE KEY `purchasePrice` (`purchasePrice`),
  ADD UNIQUE KEY `sellingPrice` (`sellingPrice`),
  ADD KEY `barcode` (`barcode`);
```

Indexes for table `purchase`

```
ALTER TABLE `purchase`
  ADD PRIMARY KEY (`purchasePrice`),
  ADD KEY `barcode` (`barcode`,`purchasePrice`),
  ADD KEY `supplierName` (`supplierName`);
```

Indexes for table `stock`

```
ALTER TABLE `stock`
  ADD UNIQUE KEY `barcode` (`barcode`),
  ADD KEY `stockID` (`stockID`);
```

Indexes for table `supplier`

```
ALTER TABLE `supplier`
  ADD PRIMARY KEY (`supplierName`);
```

Indexes for table `type`

```
ALTER TABLE `type`
  ADD PRIMARY KEY (`typeName`),
  ADD KEY `barcode` (`barcode`);
```

AUTO_INCREMENT for dumped tables

AUTO_INCREMENT for table `customer`

```
ALTER TABLE `customer`
  MODIFY `customerID` int(5) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6;
```

AUTO_INCREMENT for table `doctor`

```
ALTER TABLE `doctor`
  MODIFY `doctorID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=5679;
```

AUTO_INCREMENT for table `employees`

```
ALTER TABLE `employees`
  MODIFY `employeeID` int(4) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6;
```

AUTO_INCREMENT for table `items`

```
ALTER TABLE `items`
  MODIFY `barcode` int(4) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=25;
```

Constraints for dumped tables

Constraints for table `bill`

```
ALTER TABLE `bill`
  ADD CONSTRAINT `bill_ibfk_1` FOREIGN KEY (`customerID`) REFERENCES
`customer` (`customerID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `bill_ibfk_2` FOREIGN KEY (`employeeID`) REFERENCES
`employees` (`employeeID`) ON UPDATE CASCADE,
  ADD CONSTRAINT `bill_ibfk_3` FOREIGN KEY (`itemName`) REFERENCES `drug`
(`itemName`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Constraints for table `disorder`

```
ALTER TABLE `disorder`
  ADD CONSTRAINT `disorder_ibfk_1` FOREIGN KEY (`doctorID`) REFERENCES
`doctor` (`doctorID`);
```

Constraints for table `items`

```
ALTER TABLE `items`
  ADD CONSTRAINT `items_ibfk_1` FOREIGN KEY (`supplierName`) REFERENCES
`supplier` (`supplierName`);
```

Constraints for table `prescription`

```
ALTER TABLE `prescription`
    ADD CONSTRAINT `prescription_ibfk_1` FOREIGN KEY (`customerID`)
REFERENCES `customer` (`customerID`) ON DELETE CASCADE ON UPDATE
CASCADE,
     ADD CONSTRAINT `prescription_ibfk_2` FOREIGN KEY (`doctorID`)
REFERENCES `doctor` (`doctorID`) ON DELETE CASCADE ON UPDATE
CASCADE,
     ADD CONSTRAINT `prescription_ibfk_3` FOREIGN KEY (`itemName`)
REFERENCES `drug` (`itemName`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

Constraints for table `prices`

```
ALTER TABLE `prices`
   ADD CONSTRAINT `prices_ibfk_1` FOREIGN KEY (`barcode`) REFERENCES
`items` (`barcode`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Constraints for table `purchase`

```
ALTER TABLE `purchase`
  ADD CONSTRAINT `purchase_ibfk_1` FOREIGN KEY (`barcode`) REFERENCES
`items` (`barcode`) ON UPDATE CASCADE,
    ADD CONSTRAINT `purchase_ibfk_2` FOREIGN KEY (`supplierName`)
REFERENCES `supplier` (`supplierName`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

Constraints for table `stock`

```
ALTER TABLE `stock`
   ADD CONSTRAINT `stock_ibfk_1` FOREIGN KEY (`barcode`) REFERENCES
`items` (`barcode`) ON UPDATE CASCADE;
```

Constraints for table `type`

```
ALTER TABLE `type`
  ADD CONSTRAINT `type_ibfk_1` FOREIGN KEY (`barcode`) REFERENCES `items`
(`barcode`) ON UPDATE CASCADE;
```

Defining profit column:

```
UPDATE `price` SET `profit`=`tax`*`sellingPrice`
```

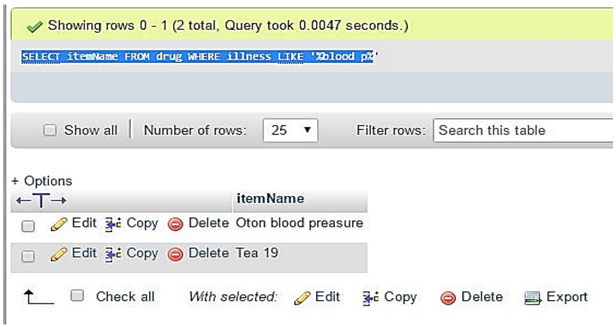
Defining availability column

```
UPDATE  `stock`  SET `availability`=`yes` WHEN `noOfItems`>0
```

Queries

To prove that our database works as it should, we wrote several queries and attached their outputs. They make things simplified, so we use this kind of queries while we were building our database, too. It is easier for us to relate events that could or can happen.

```
SELECT itemName FROM drug WHERE illness LIKE '%blood p%
```



✓ Showing rows 0 - 1 (2 total, Query took 0.0047 seconds.)

```
SELECT itemName FROM drug WHERE illness LIKE '%blood p%'
```

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table

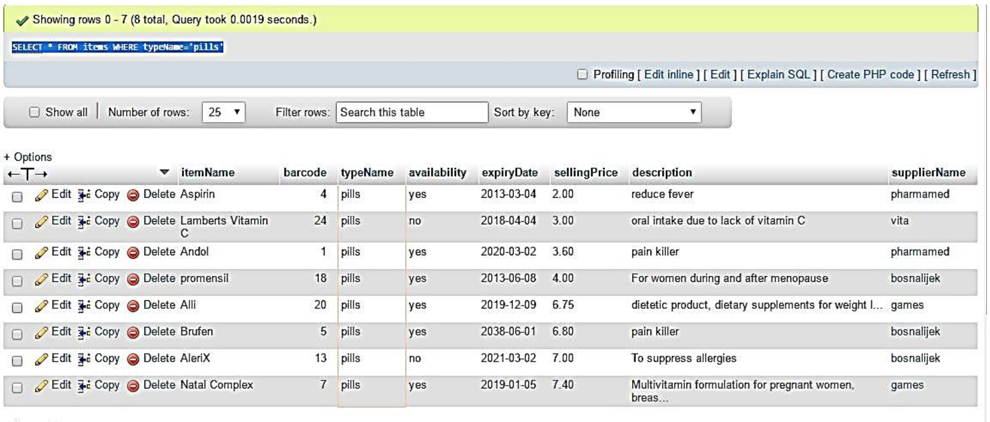
+ Options

←T→	itemName
□ Edit Copy Delete	Oton blood preasure
□ Edit Copy Delete	Tea 19

□ Check all *With selected:* Edit Copy Delete Export

Here we searched for all drugs that are for high or low blood pressure.

```
SELECT * FROM items WHERE typeName='pills'
```



✓ Showing rows 0 - 7 (8 total, Query took 0.0019 seconds.)

```
SELECT * FROM items WHERE typeName='pills'
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

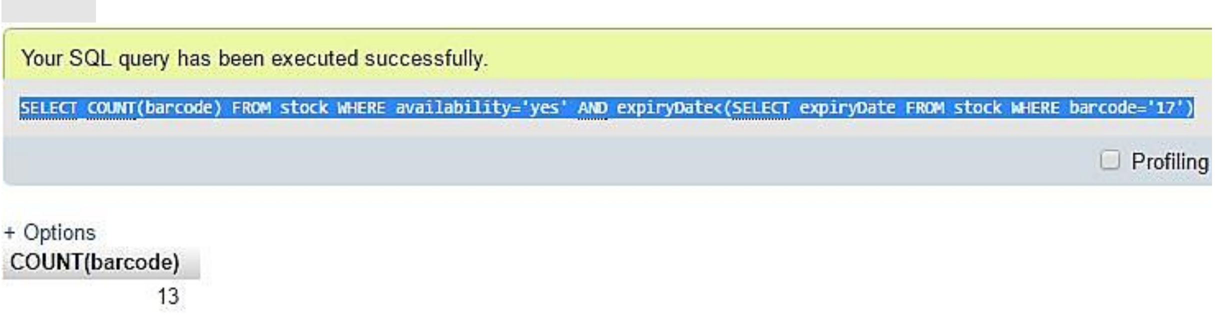
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options

←T→	itemName	barcode	typeName	availability	expiryDate	sellingPrice	description	supplierName
□ Edit Copy Delete	Aspirin	4	pills	yes	2013-03-04	2.00	reduce fever	pharmamed
□ Edit Copy Delete	Lamberts Vitamin C	24	pills	no	2018-04-04	3.00	oral intake due to lack of vitamin C	vita
□ Edit Copy Delete	Andol	1	pills	yes	2020-03-02	3.60	pain killer	pharmamed
□ Edit Copy Delete	promensil	18	pills	yes	2013-06-08	4.00	For women during and after menopause	bosnalijek
□ Edit Copy Delete	Alli	20	pills	yes	2019-12-09	6.75	dietetic product, dietary supplements for weight l...	games
□ Edit Copy Delete	Brufen	5	pills	yes	2038-06-01	6.80	pain killer	bosnalijek
□ Edit Copy Delete	AleriX	13	pills	no	2021-03-02	7.00	To suppress allergies	bosnalijek
□ Edit Copy Delete	Natal Complex	7	pills	yes	2019-01-05	7.40	Multivitamin formulation for pregnant women, breas...	games

Here we searched for all pills and their characteristics.

```
SELECT COUNT(barcode) FROM stock WHERE availability='yes' AND expiryDate<(SELECT expiryDate FROM stock WHERE barcode='17')
```



Your SQL query has been executed successfully.

```
SELECT COUNT(barcode) FROM stock WHERE availability='yes' AND expiryDate<(SELECT expiryDate FROM stock WHERE barcode='17')
```

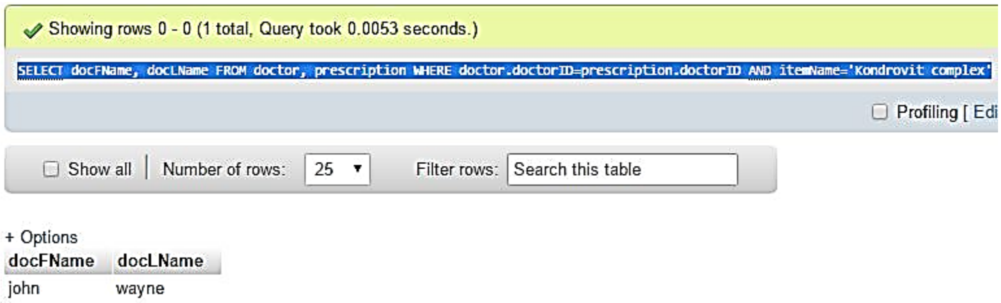
□ Profiling

+ Options

COUNT(barcode)
13

Here we counted how many drugs have expiry date which is smaller than the expiry date of item barcode 17.

```
SELECT docFName, docLName FROM doctor, prescription WHERE doctor.doctorID=prescription.doctorID AND itemName='Kondrovit complex'
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0053 seconds.)

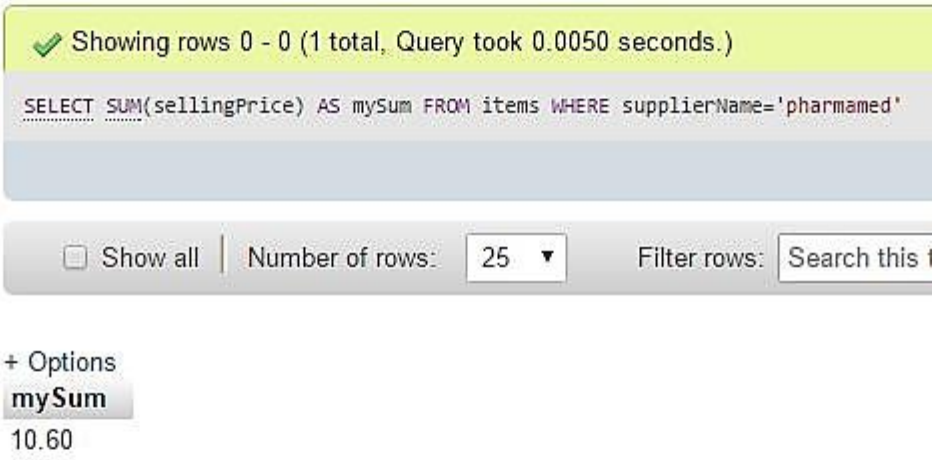
```
SELECT docFName, docLName FROM doctor, prescription WHERE doctor.doctorID=prescription.doctorID AND itemName='Kondrovit complex'
```

□ Profiling [Edi
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table
+ Options

docFName	docLName
john	wayne

Here we searched for a doctor who prescribed the Kondrovit complex.

```
SELECT SUM(sellingPrice) AS mySum FROM items WHERE supplierName='pharmamed'
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0050 seconds.)

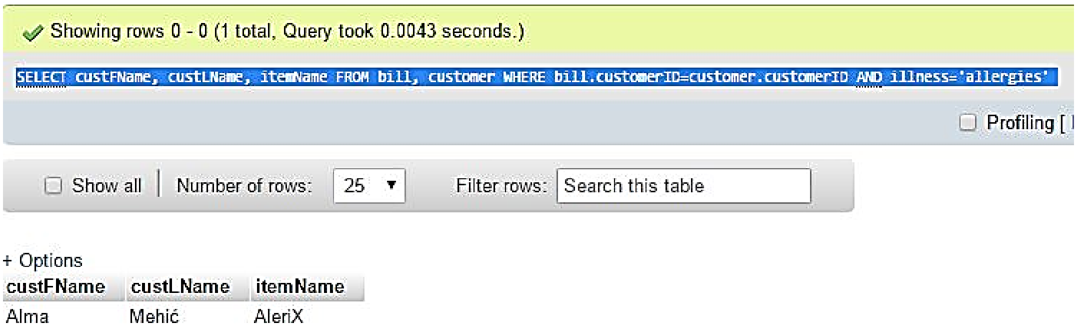
```
SELECT SUM(sellingPrice) AS mySum FROM items WHERE supplierName='pharmamed'
```

□ Show all | Number of rows: 25 ▾ Filter rows: Search this t
+ Options

mySum
10.60

Here we’ve searched about the overall selling price for supplier Pharmamed

```
SELECT custFName, custLName, itemName FROM bill, customer WHERE bill.
customerID=customer.customerID AND illness='allergies'
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0043 seconds.)

```
SELECT custFName, custLName, itemName FROM bill, customer WHERE bill.customerID=customer.customerID AND illness='allergies'
```

custFName	custLName	itemName
Alma	Mehić	AleriX

Here we searched for a customer who is suffering from allergies.

Report

For our task, we had to choose some real life problem to make an adequate database. We found that pharmacy database will be and excellent choice and consulting with our professor prove that. Implementation of such database requires a lot of work and much knowledge. We chose phpMyAdmin as our tool. for such a task. This experience teaches us a lot, and now we can freely say that we are able to create a good, usable database.

Our first task was to gather some data about pharmacies, so we went to the one existing pharmacy in our town to talk with pharmacists who gave us a good base and gave us the requirements to which we had to satisfy. Right away we went to analyze those requirements and define entities.

First entity we defined is entity items, where we add all characteristics for each item as itemName, barcode, its type, description, expiryDate and so on. After that we just tried to connect that table with every other we created, so we put reasonable attributes for each table, such as in bill we add itemName, which means its connected with items table, and bill also has a customerID attribute which means its connected with customer table and so on.

After defining 13 entities, each of them had the appropriate connection with some other table. It took a lot of time and effort to find the best possible schedule. To make sure that our database works right, we needed to fill it with appropriate values, define primary and foreign keys so we could connect them correctly. Then we realized that for each type of values we should create a special table where we will put an exact vlue for each attribute without repeating it and how it deals with other values from other attributes in that table. That is a point where we started our normalization process. We tried hard to make every table normalized as much as possible. Many of them are already in 1FN but also there are relations in 2NF, as you can see.

There are some columns which were set manually so we didn’t have populate every row and every column every time. Profit attribute from price table is an example. There we wrote an SQL statement where we said that profit table will be updated (it will not be blank) in such a way that for every row which has a sellingPrice we will multiply that value with default value of tax in our country, and get the value of profit for each item in our database. Every query we checked in SQL part to prove that it works as it should. In this paper you can see our altering for tables and updates so you can be up to date of our work.