МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Пензенский государственный технологический университет»

(ПензГТУ)

Факультет информационных и образовательных технологий

Кафедра «Информационные технологии и системы»

Дисциплина «Управление информационными ресурсами»

ОТЧЕТ

по контрольной работе на тему:

«Учёт медицинских препаратов в ветеринарной клинике»

Выполнил: студент гр. 15ИС2м Баклушина Н.А.

Проверил: ст. преподаватель каф. ИТС Володин К.И.

Работа защищена с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пенза, 2016

Содержание

Введение…………………………………………………………………………...3

1. Краткая характеристика предметной области………………………………..5

2. Создание схемы хранения данных в рамках современной клиент-серверной СУБД с применением визуальных средств проектирования…………………..8

3. Разработка и тестирование запросов на языке *SQL*………………………...14

Заключение……………………………………………………………………….19

Список литературы………………………………………………………………20

Приложение 1…………………………………………………………………….21

Приложение 2…………………………………………………………………….22

Введение

Сейчас невозможно представить жизнь человека без новейших технологий. Они проникли уже во все сферы производства и торговли. Раньше, когда не было компьютеров, приходилось работать с бумажными документами, и это было малоэффективно, особенно в производственных сферах и бизнесе, так как поиск и обработка этих документов затрачивали довольно большое количество времени. Но прогресс не стоит на месте. Сейчас современные программные средства, такие как *Microsoft SQL Server , MySQL Workbench* обеспечивают быстрый ввод, обработку, хранение, изменение данных и позволяет сэкономить достаточно большое количество времени как у работников предприятия, так и у его клиентов.

Практически все организации используют компьютеры и выше перечисленные программные средства для хранения и обработки служебной информации. Эта информация содержится в базе данных. Базы данных играют особую роль в современном мире. Умение работать с ними является одним из важнейших навыков в работе с компьютером, а специалисты в этой области всегда востребованы.

База данных (БД) - это хранилище для большого количества систематизированных данных, с которыми можно производить определённые действия: добавления, удаления, изменения, копирования, упорядочивание. Эти базы данных образовываются и функционируют под управлением специальных программных комплексов (совокупностей языков программирования и программных средств), называемых системами управления базами данных (СУБД). По существу, система управления базами данных служит посредником между пользователями и базой данных [1].

За последние несколько лет наблюдается тенденция к усложнению структур данных. Простые виды информации, представимой в форме чисел и текстовых строк, дополняются многочисленными мультимедийными документами, графическими образами, хронологическими рядами и множеством прочих сложных информационных форм. В связи с этим появилось множество весьма изощренных СУБД, поддерживающих новые коллекции данных и способных реализовать преимущества современных аппаратных средств.

Целью контрольной работы является создание СУБД для работы ветеринарной клиники на основе программы *MySQL Workbench.*

В связи с поставленной целью были выявлены следующие задачи:

1. Выделение информационных объектов (отношения, таблицы) необходимых для работы ветеринарной клиники;
2. Создание схемы хранения данных в рамках современной клиент-серверной СУБД с применением визуальных средств проектирования, а именно программы *MySQL Workbench*;
3. Генерация кода *SQL* БД ветеринарной клиники;
4. Разработка и тестирование запросов на языке *SQL*.

Таким образом, можно сказать о том, что выбранная тема контрольной работы является актуальной, так как использование современных программных средств для автоматизация рабочего места служащего предприятия, в дальнейшем будет способствовать более качественной работе, а следовательно к повышению прибыли.

Аннотация:

1. В работе представлена краткая характеристика предметной области.
2. Разработана схема хранения СУБД учета медицинских препаратов ветеринарной клиники на в программе *MySQL Workbench.*
3. Были проведены основные формы запросов на языке *SQL.*
4. Сделаны выводы по проделанной работе.

1. Краткая характеристика предметной области

Предметной областью проекта является автоматизированное рабочее место по учёту медицинских препаратов в ветеринарной клинике.

Учитывая задачи создания БД, целесообразно выделить семь информационных объектов (отношения, таблицы): препарат, группы препаратов, поставщики, журнал прихода, журнал выдачи, МЛ, заказ препаратов.

Рассмотрим структуру таблиц, описывающих эти сущности, и разработаем модель данных «сущность-связь».

1. Сущность «Препарат» описывает данные мед.препаратов, ключевое поле – код препарата (*DrugsID*) ( (таблица 1)

Таблица 1 – Описание таблицы «Препарат» («*Drugs*»)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Код препарата (*DrugsID*) | Наименование препаратов (*DrugsName*) | Группа препаратов (*GroupsOfDrugsID*) | Производитель (*CreatorsID*) | Фото (*Photo*) |
| Тип | *INT* | *VARCHAR* | *INT* | *INT* | *VARCHAR* |
| Размер |  | 255 |  |  | 255 |

Сущность «Группы препарат» описывает данные групп мед.препаратов, ключевое поле – код группы (*GroupsOfDrugsID*)(таблица 2 )

Таблица 2 – Описание таблицы «Группы препаратов» («*GroupsOfDrugs*»)

|  |  |  |
| --- | --- | --- |
| Имя поля | Код группы (*GroupsOfDrugsID*) | Наименование группы (*GroupsOfDrugsName*) |
| Тип | *INT* | *VARCHAR* |
| Размер |  | 255 |

Сущность «ЖурналВыдачи» описывает данные о выдаче мед.препаратов МОЛ (таблица 3).

Таблица 3 – Описание таблицы «ЖурналВыдачи» («*BookDelivery*»)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Код записи (*RecordID*) | Кол-во (*Sum*) | Дата выдачи (*DateDelivery*) | Мол (*MOLID*) |
| Тип | *INT* | *INT* | *DATE* | *INT* |
| Размер |  |  |  |  |

Сущность «ЖурналПрихода» описывает данные о поступлении мед.препаратов, ключевое поле – код записи (таблица 4)

Таблица 4 – Описание таблицы «ЖурналПрихода» («*BookComing*»)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Код записи (*IDRecord)* | Код заказа (*CustomDrugsID*) | Препарат *(DrugsID*) | Цена (*Price*) | Приход (*Sum*) | Дата изготовления (*DateOfManufacture*) | Срок годности (*ExpirationDate*) |
| Тип | *INT* | *INT* | *INT* | *Numeric* | *INT* | *DATE* | *DATE* |
| Размер |  |  |  | (15,2) |  |  |  |

Сущность «Заказ препаратов» описывает данные о заказе мед.препаратов, ключевое поле – код заказа (*CustomDrugsID*) (таблица 5).

Таблица 5 – Описание таблицы «Заказ препаратов» («*CustomDrugs*»)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Код заказа (*CustomDrugsID*) | Дата заказа (*DataCustom*) | МОЛ (*MOLID*) |
| Тип | *INT* | *DATE* | *INT* |
| Размер |  |  |  |

Сущность «МОЛ» описывает данные о материально ответственных лицах, ключевое поле – код МОЛ (*MOLID*)(таблица 6).

Таблица 6 – Описание таблицы «МОЛ» («*MOL*»)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Код МОЛ (*MOLID*) | ФИО (*FIO*) | Должность (*Functions*) | Телефон (*Phone*) |
| Тип | *INT* | *VARCHAR* | *VARCHAR* | *VARCHAR* |
| Размер |  | 255 | 255 | 45 |

Сущность «Поставщики» описывает данные о поставщиках, ключевое поле – код поставщика (*CreatorsID*)(таблица 7).

Таблица 7 – Описание таблицы «Поставщики» (*Creators*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Код поставщика (*CreatorsID*) | Наименование организации (*CreatorsName*) | Город (*City*) | ИНН (*INN*) | Телефон (*Phone*) |
| Тип | *INT* | *VARCHAR* | *VARCHAR* | *INT* | *INT* |
| Размер |  | 255 | 255 | 12 | 45 |

Реальные объекты всегда каким-то образом взаимосвязаны. Схема данных представлена в Приложении А.

2. Создание схемы хранения данных в рамках современной клиент-серверной СУБД с применением визуальных средств проектирования

Для добавления модели необходимо кликнуть значок плюса рядом с заголовком "Models" или выбрать команду меню "File → New Model".

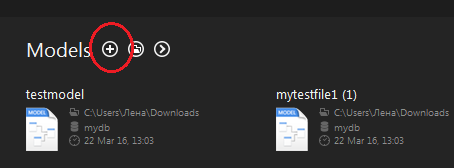


Рисунок 1 − Форма создания модели

В открывшемся окне необходимо ввести имя базы данных, выбрать кодировку и при необходимости заполнить поле комментария.

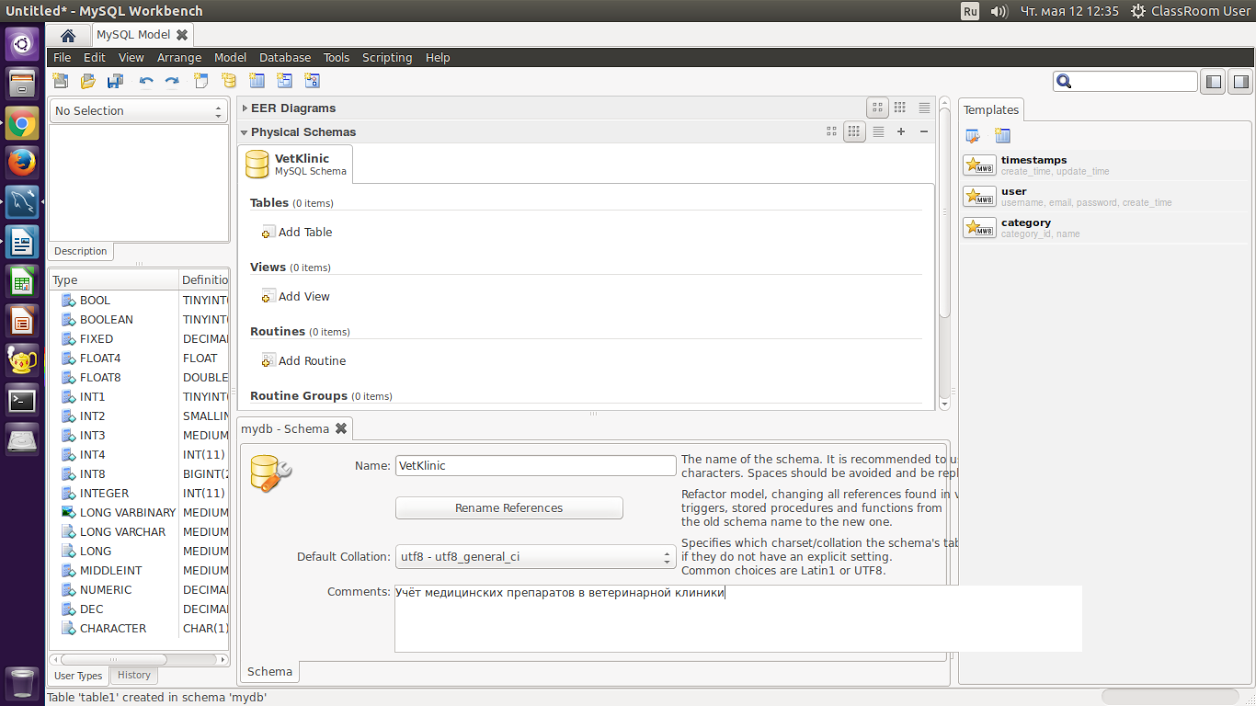


Рисунок 2 − Вкладка создания БД

Для создания новой таблицы на вкладке *«Physical Schemas»* необходимо кликнуть *«Add Table»* и заполнить поля — название таблицы, поля таблицы, их типы и длину.

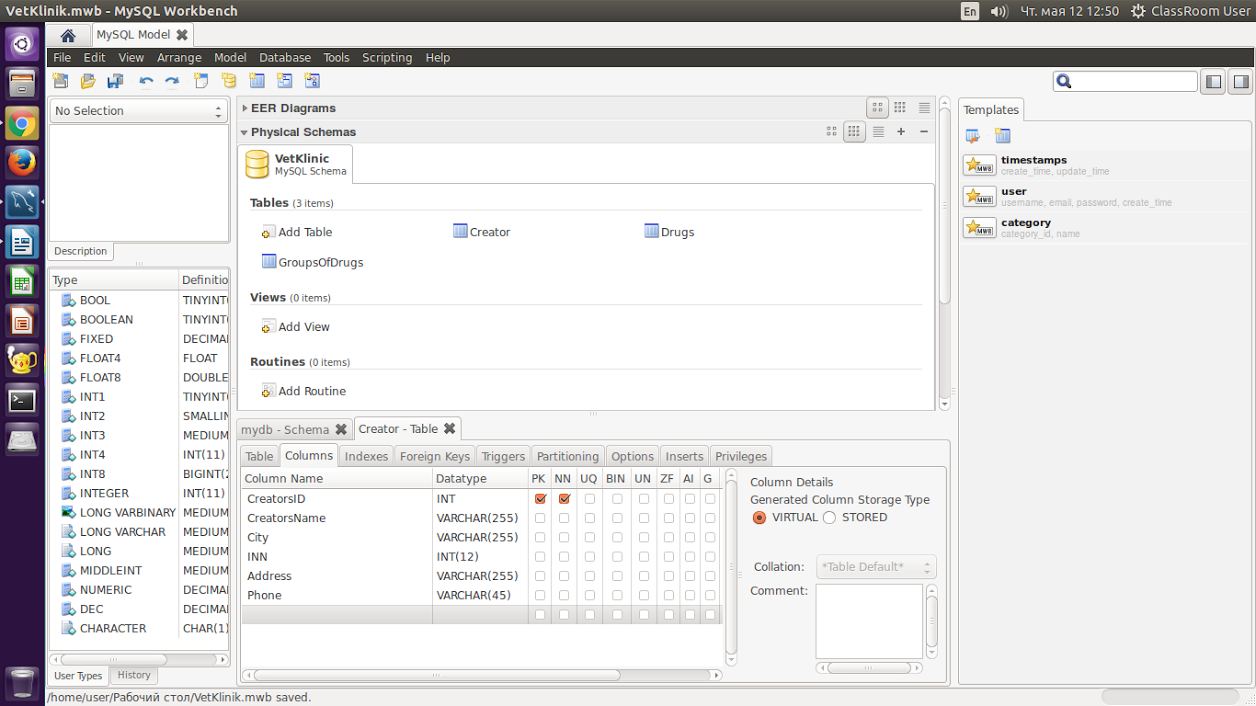


Рисунок 3 − Кнопка создания таблицы

После создания таблицы, необходимо заполнить все её поля.

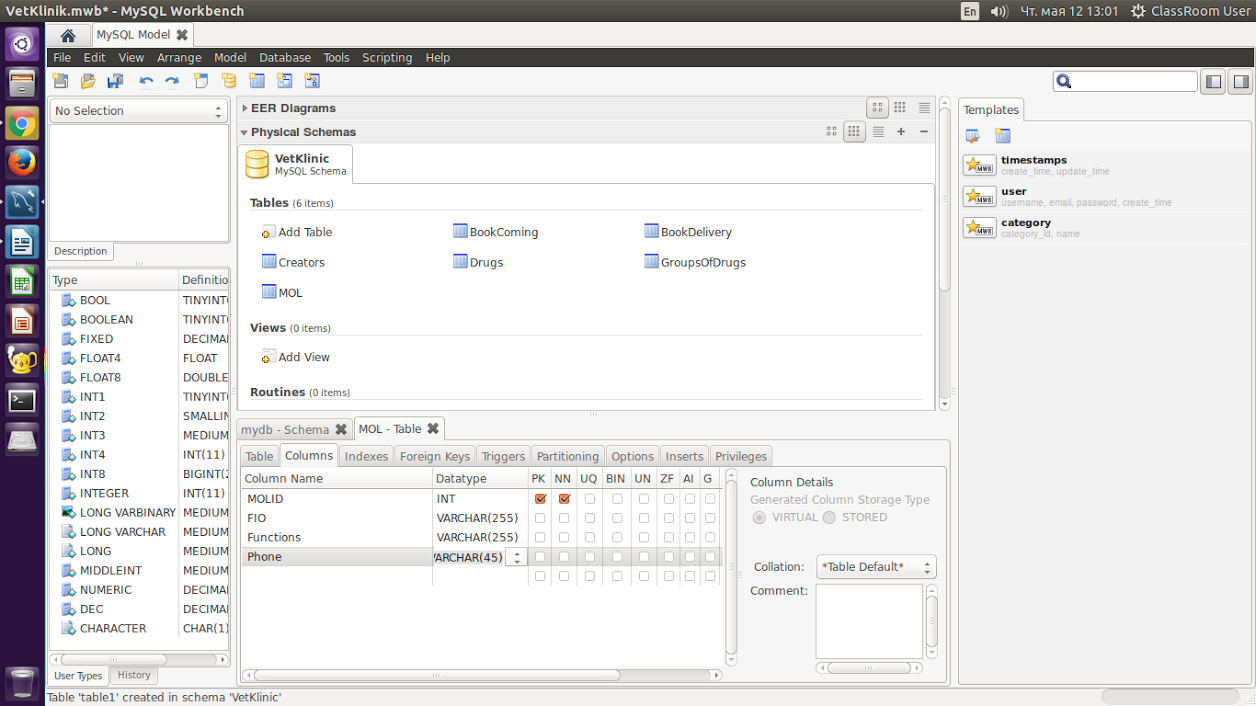


Рисунок 4 − Создание таблицы

Все созданные таблицы отображаются в разделе *«Tables»:*



Рисунок 5 − Созданные таблицы

После создания таблиц, необходимо создать связи между ними. Связи создаются через вкладку *«Foreign Keys»* у необходимой таблицы, либо непосредственно при рисовании диаграммы базы данных.

Для создания связи таблиц через вкладку *«Foreign Keys»* необходимо перейти на эту вкладку, ввести название ключа и выбрать таблицу для связи. Дальше в правой части формы, где указана таблица с полями у нужного поля выбирается поле указанной ранее таблицы, т.е. указывается связь между полями.

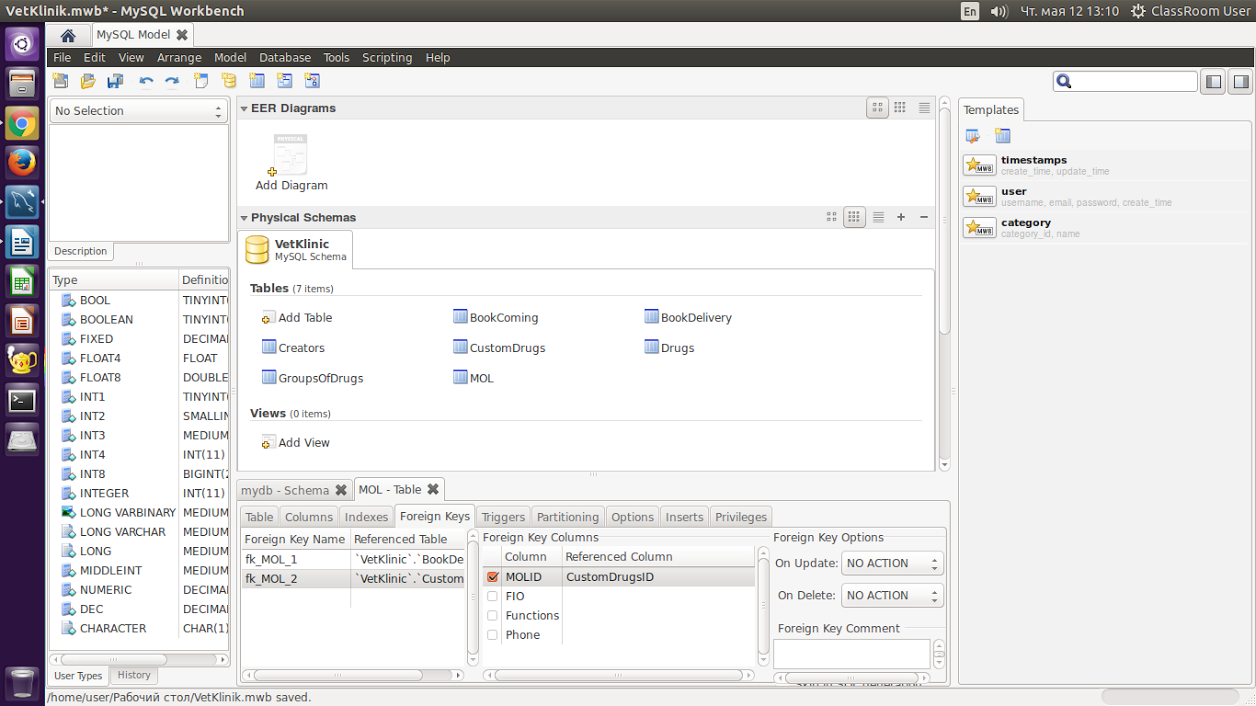


Рисунок 6 − Вкладка *«Foreign Keys»*

Для создания схемы данных, сущностей и связей между ними в *MySQL Workbench* существует редактор *EER*-диаграмм.

Для создания диаграммы необходимо в верхней части экрана управления базой данных дважды кликнуть манипулятором «Мышь» на иконку «Add Diagram»



Рисунок 7 − Иконка создания *EER*-диаграмм.

Откроется клетчатое поле для рисования:

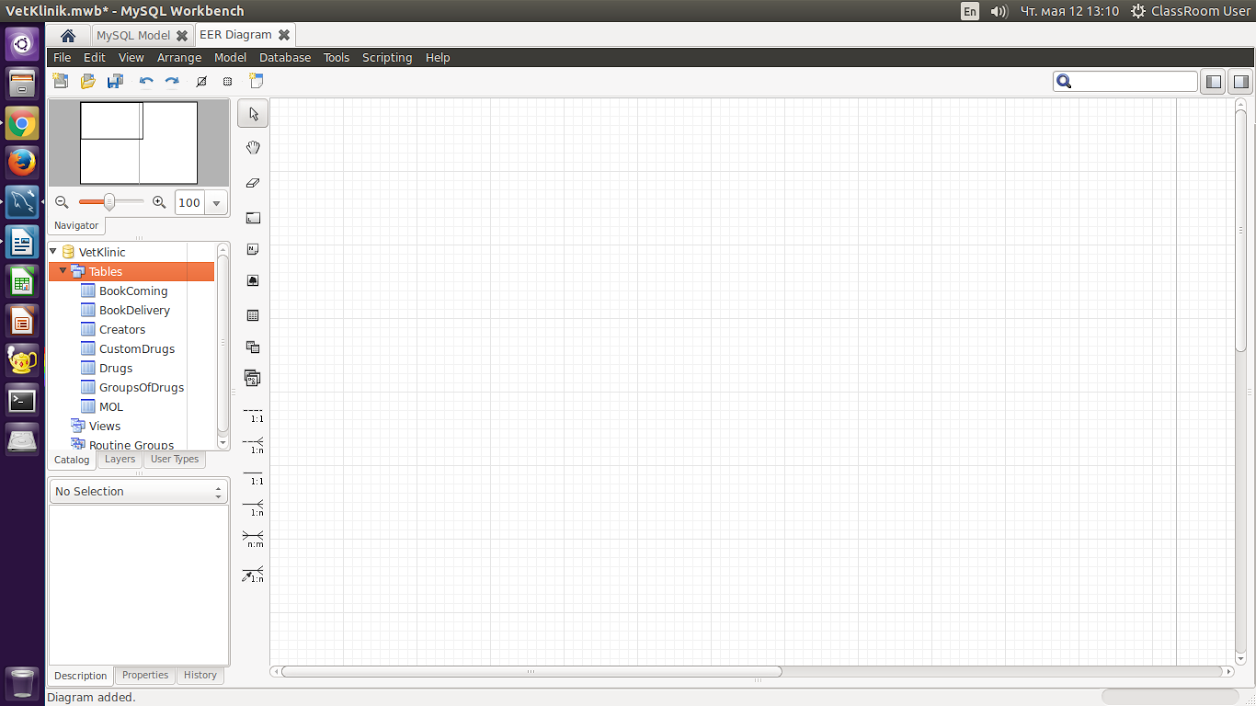


Рисунок 8 − Поле создания *EER*-диаграмм.

Необходимо выбирать слева в списке таблиц нужные таблицы и перетащить их мышью на эту клетчатую область. В результате сразу же будет построена полная диаграмма этих таблиц с учетом созданных ранее связей.

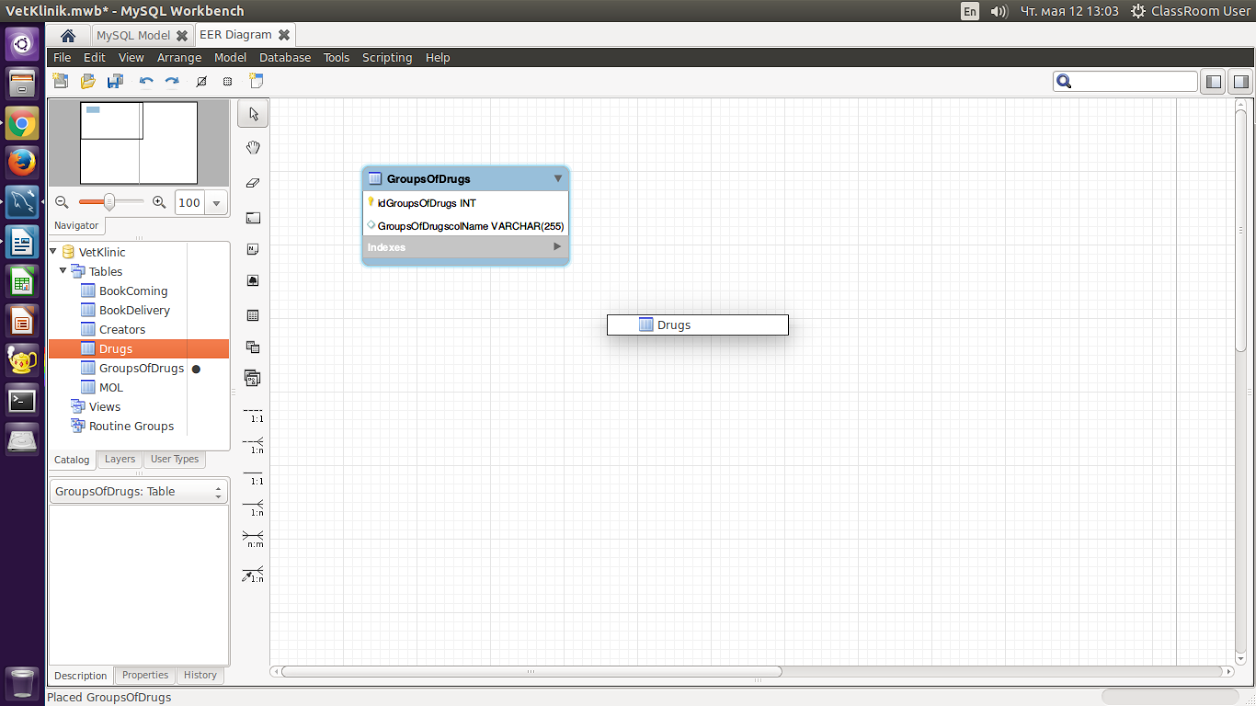


Рисунок 9 − Перенос таблиц для создания *EER*-диаграмм.

 Также здесь есть панель инструментов, в которой можно выбрать создание новой таблицы, изображения, блоки комментариев, блоки для объединения таблиц и различные варианты связей между таблицами (один к одному, один ко многим и т.п.).

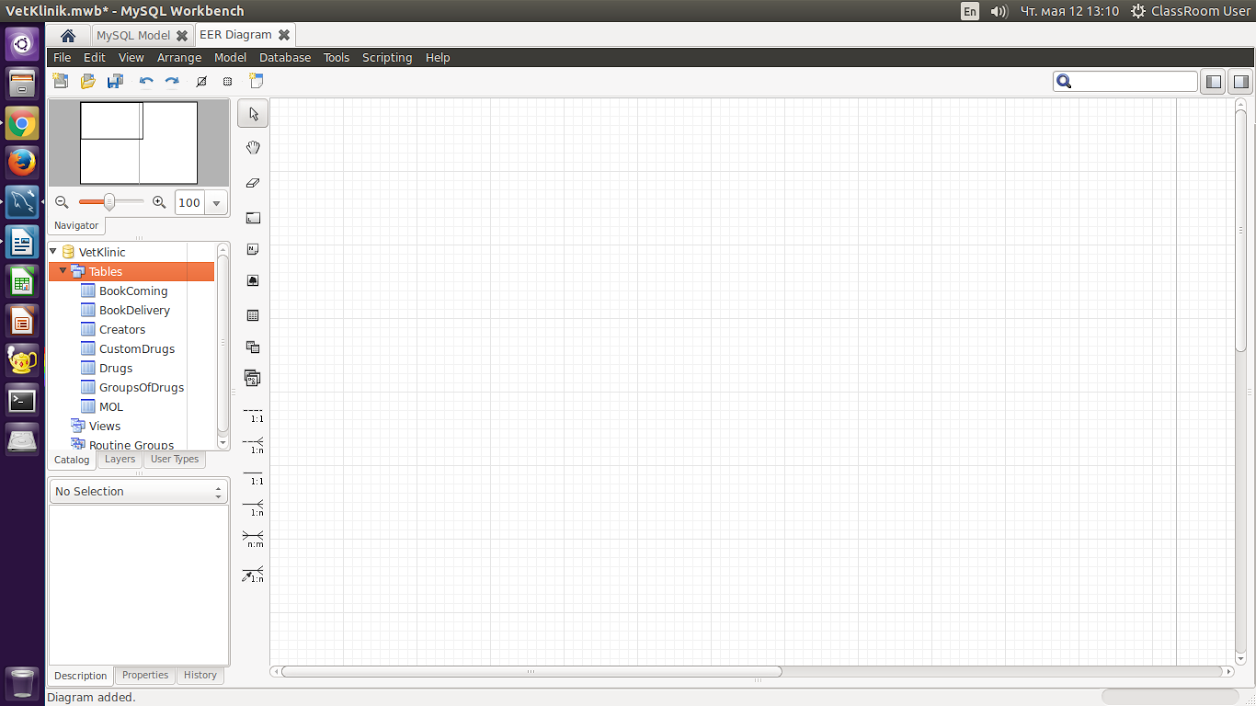


Рисунок 10 − Панель инструментов создания *EER*-диаграмм.

При создании новой таблицы визуально, она автоматически добавляется в саму модель.

Полная схема данных представлена в Приложении 1.

3. Разработка и тестирование запросов на языке *SQL*

Для того чтобы выполнять запросы необходимо сгенерировать код БД, для этого необходимо сделать следующие действия:

1. В строке *Database* выбирается функция *«Forward Engineer to Database »*

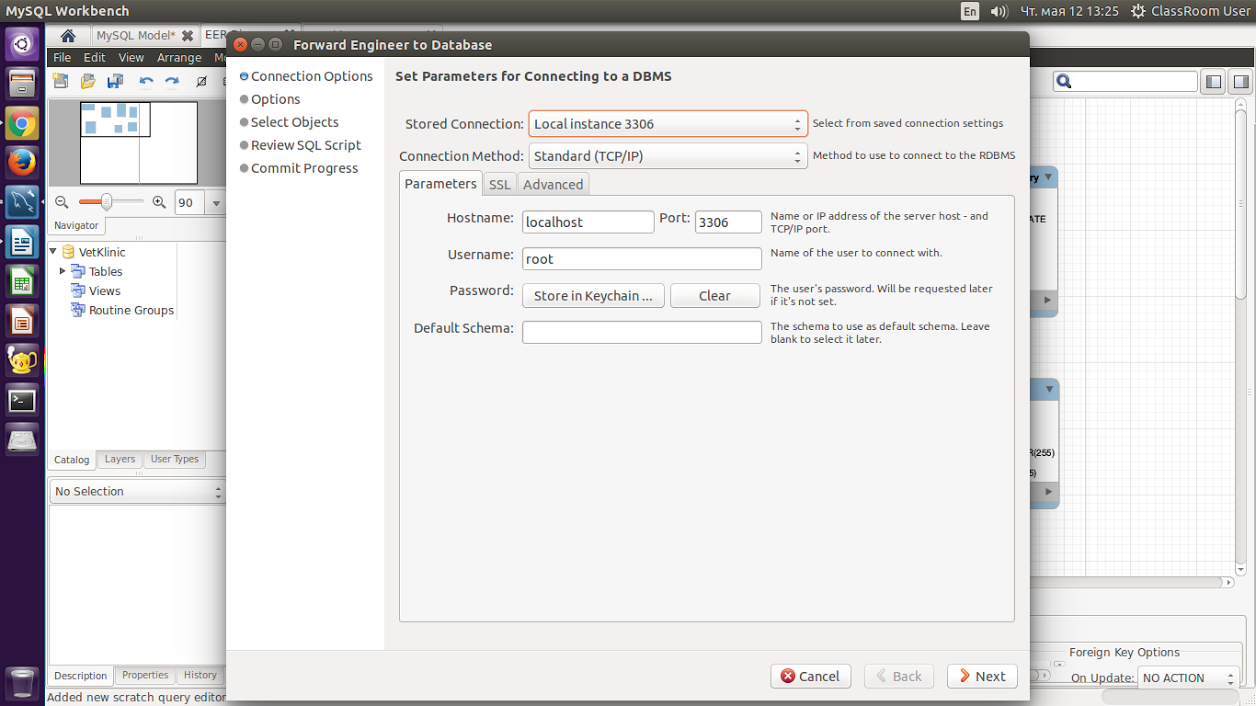


Рисунок 11 − Открытие кода *SQL* БД

2. Выбираются необходимые для открытия кода параметры

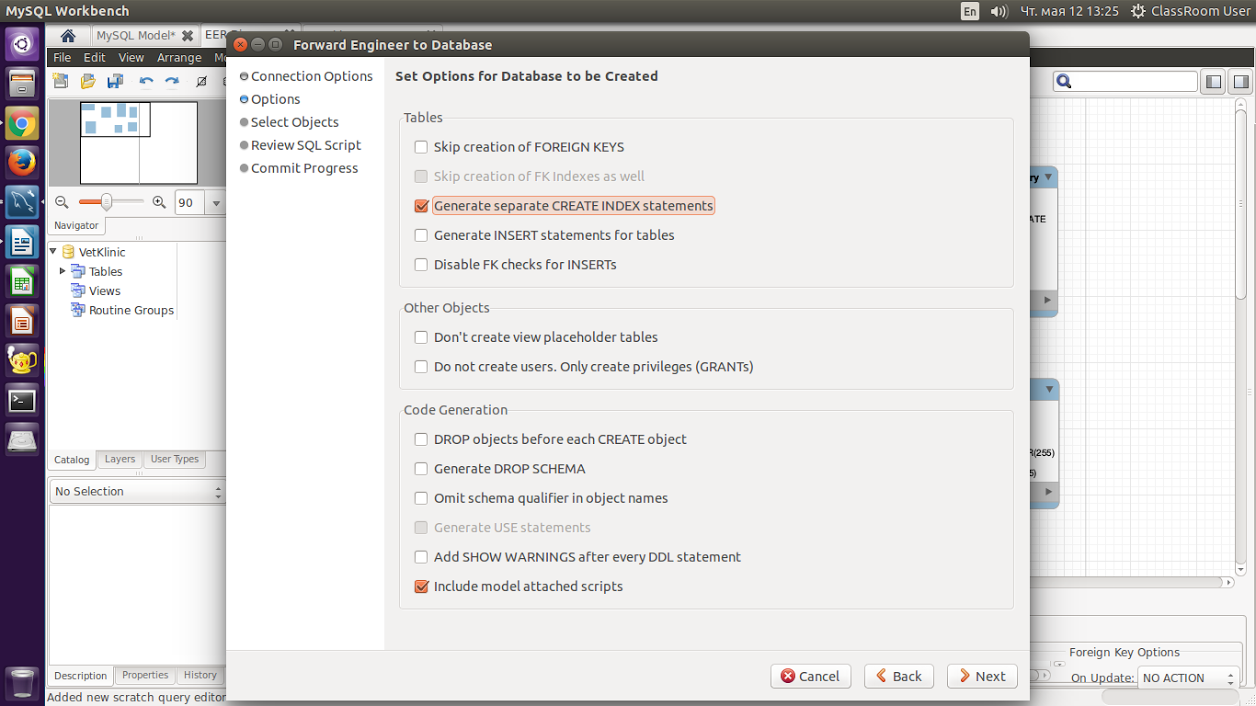


Рисунок 12 – Выбор параметров

3. Вводится пароль, открывающий доступ к серверу, где храниться код.

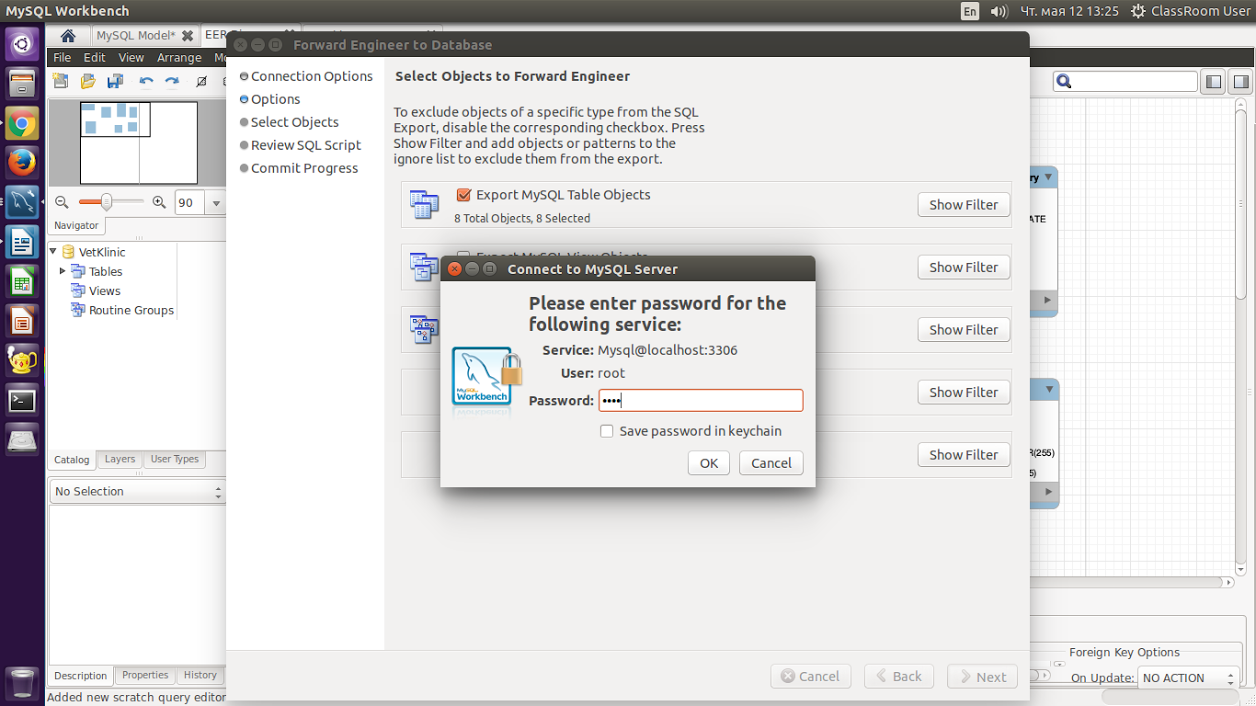


Рисунок 13 − Ввод пароля

4. После чего отображается необходимый код *SQL.* Полный код представлен в Приложении 2.

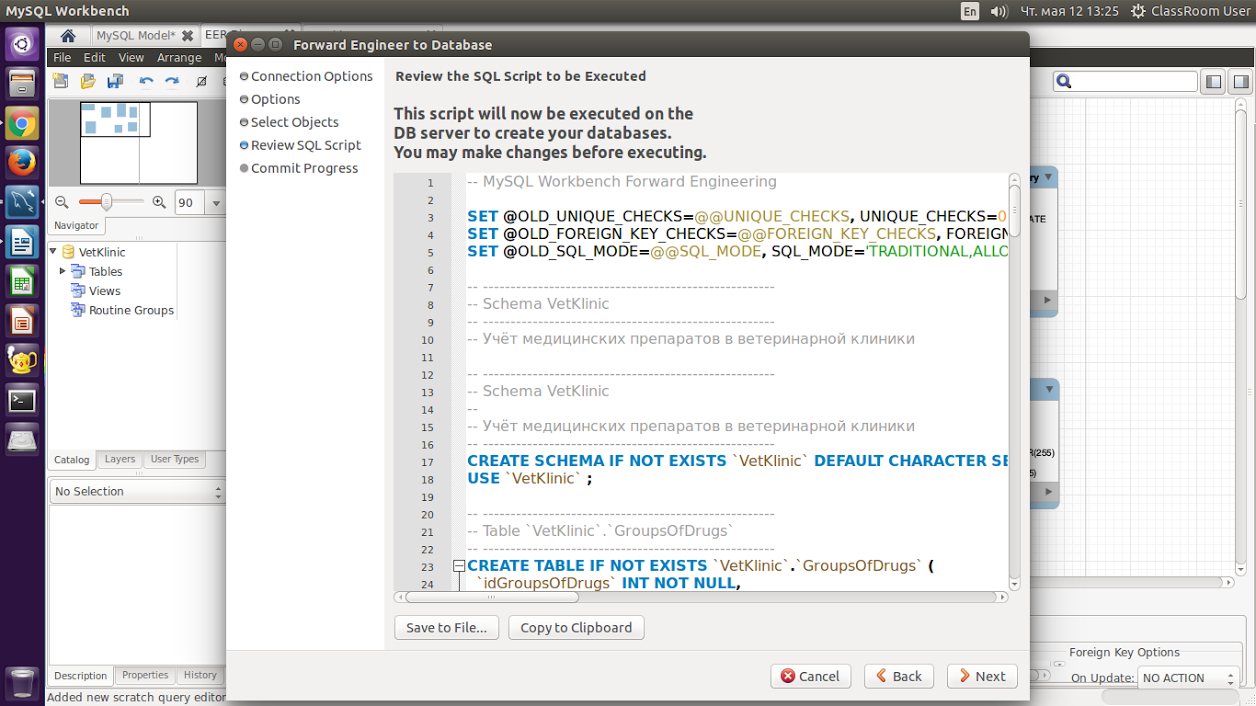


Рисунок 14 – Полученный код *SQL*

Проверим полученный ранее код на наличие ошибок. Если ошибок нет, то можно приступать к выполнению запросов.

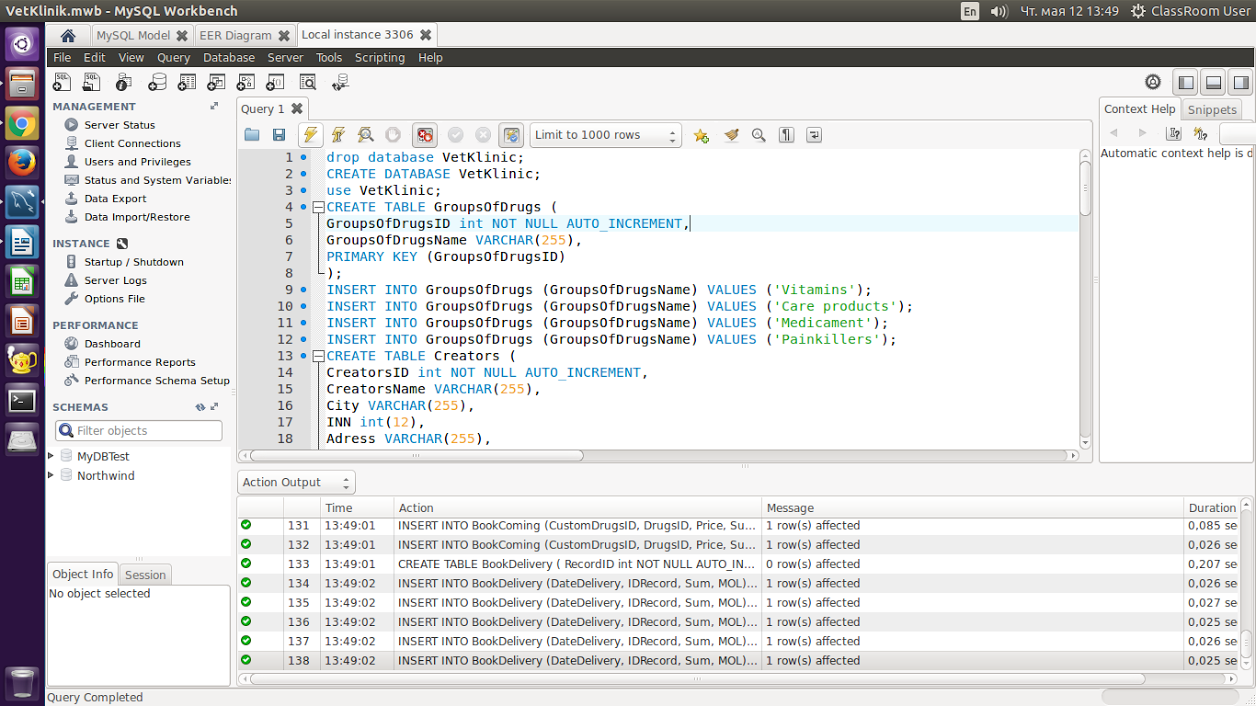


Рисунок 15 – Отладка и проверка кода *SQL*

Формирование запросов на языке *SQL* предполагает ввод команд с клавиатуры в строгом соответствии с их форматами.

Основой *SQL* является инструкция *SELECT*, используемая для создания запросов на выборку.

Синтаксис инструкции:

*SELECT [ ALL │ DISTINCT │DISTINCTROW ]* список\_выбора

*FROM* имена таблиц

*[WHERE* критерий поиска]

*[GROUP BY* имя столбца, имя столбца,…]

*[ HAVING* условие поиска]

Ниже приведены примеры выполнения *SQL* запросов:

1. Оператор (\*) - отбираются значения, находящиеся между указанными.

Пример:  
*use VetKlinic;*

*select \* from Creators*;

Результат:

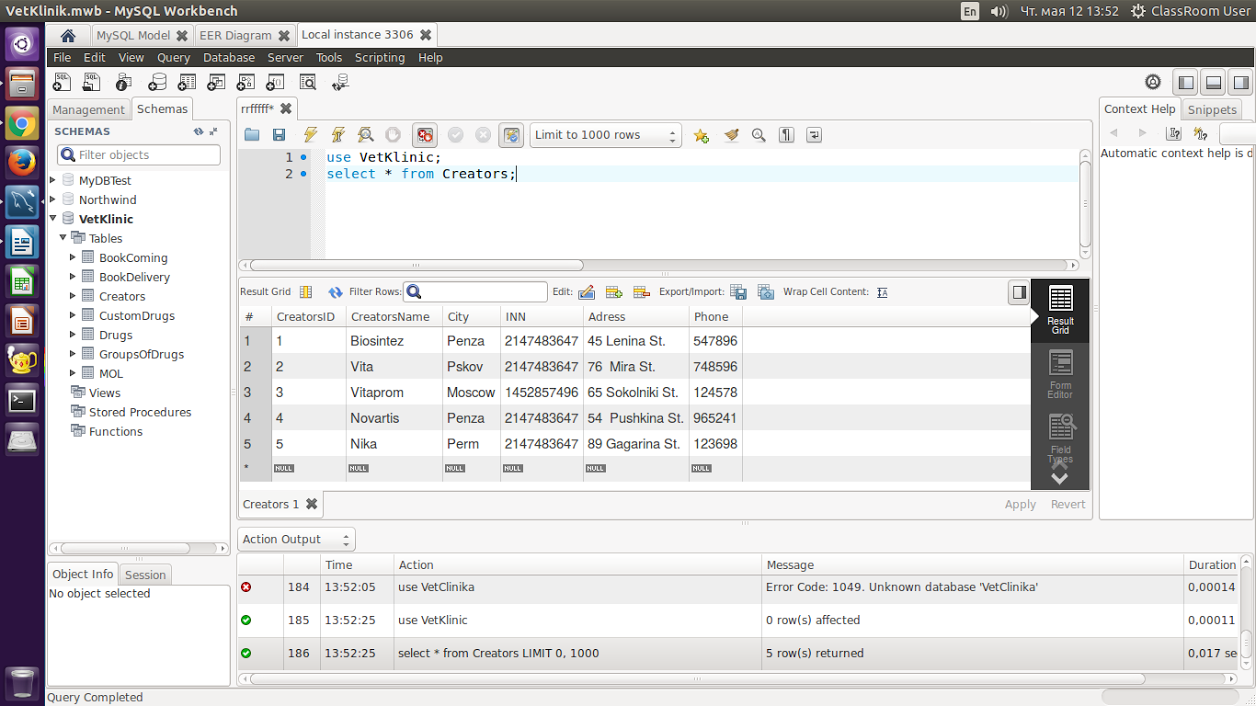


Рисунок 16 – Использование оператора «\*»

2. Оператор *«LIKE»* - отбираются значения, соответствующие образцу.

*use VetKlinic;*

*select CreatorsName FROM Cteators WHERE CreatorsName LIKE 'V%';*

Результат:

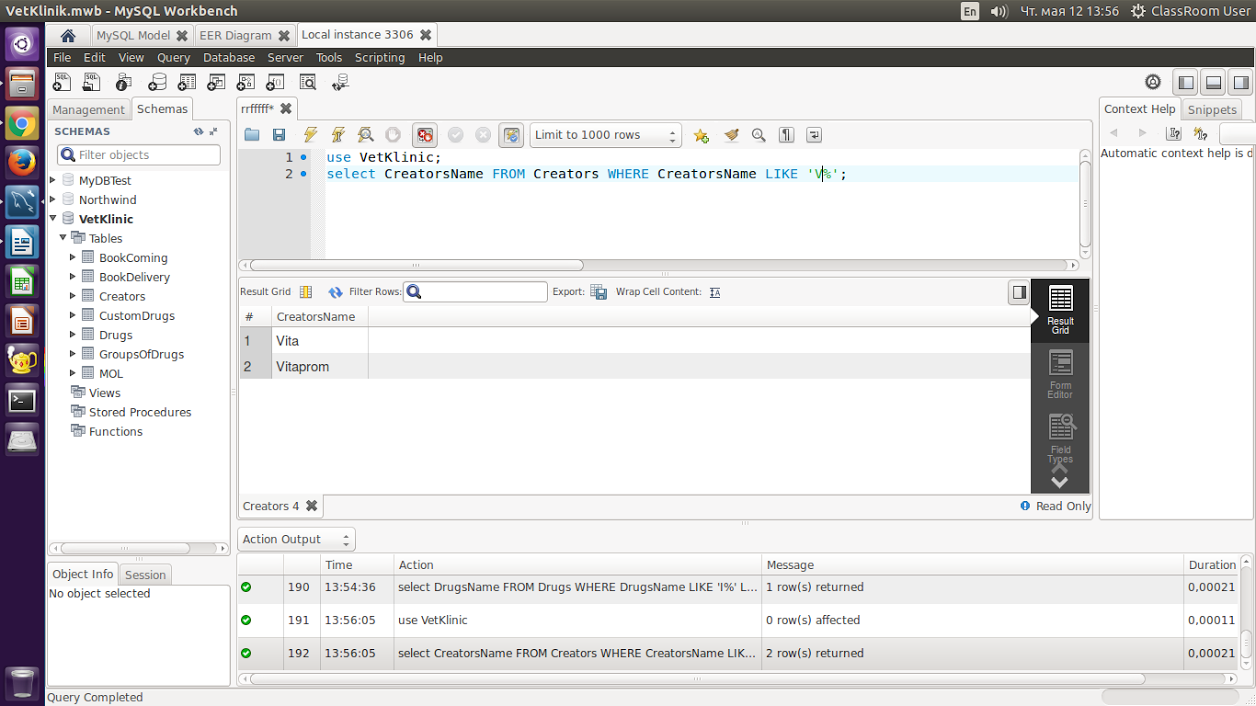


Рисунок 17 - Использование оператора «*LIKE*»

3. Оператор «in» - определяет, совпадает ли данное значение с каким-либо значением во вложенным запросе или списке.

*use VetKlinic;*

*select \* From Creators where City in (‘Moscow’);*

Результат:

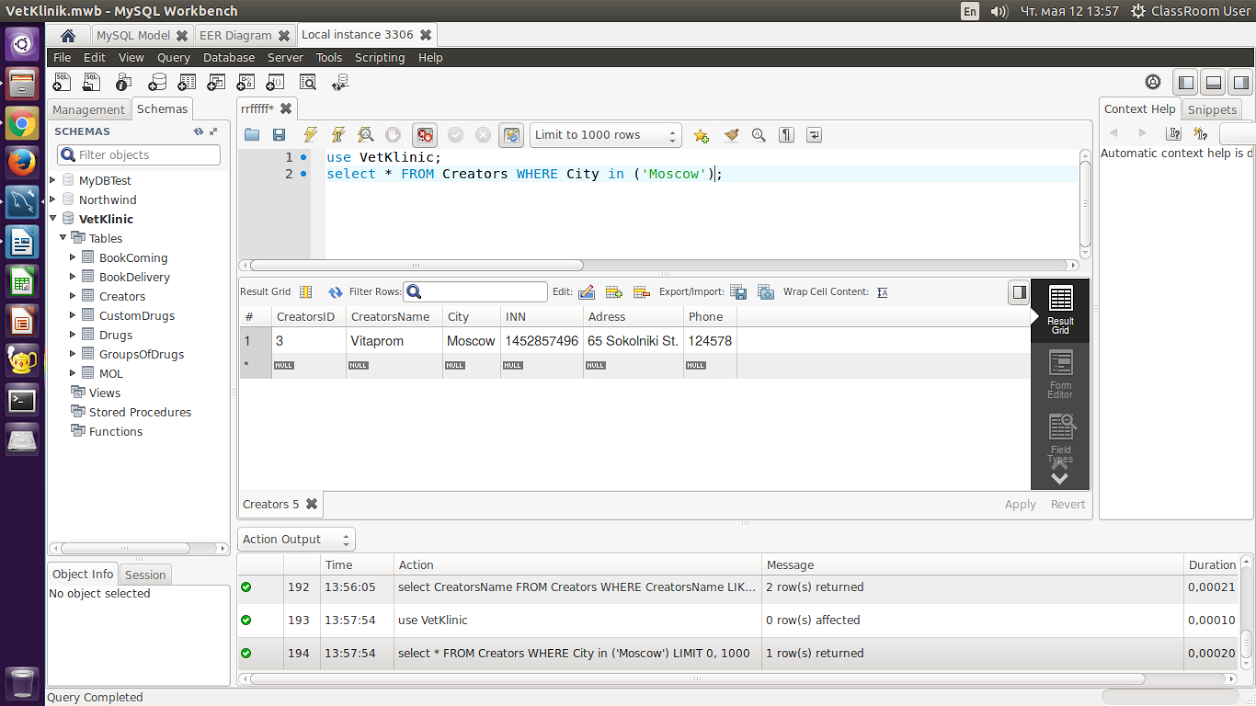


Рисунок 18 - Использование оператора «*In*»

4. Оператор «=» - отбираются значения равные указанному.

*use VetKlinic;*

*select \* From BookComing where IDRecord = 4;*

Результат:

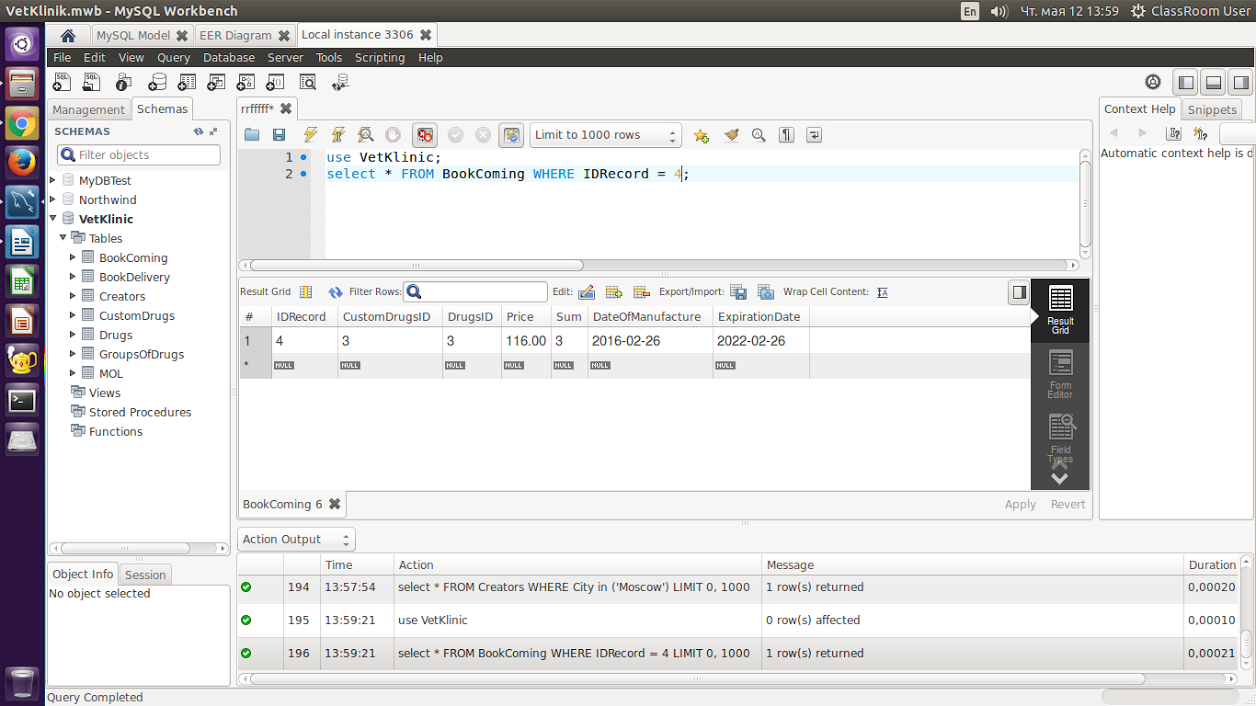


Рисунок 19 - Использование оператора «*=*»

5. Оператор «*DESC*» - Сортирует результаты запроса в порядке по убыванию.

*use VetKlinic;*

*select \* From BookComing order by DateDelivery DESC;*

Результат:

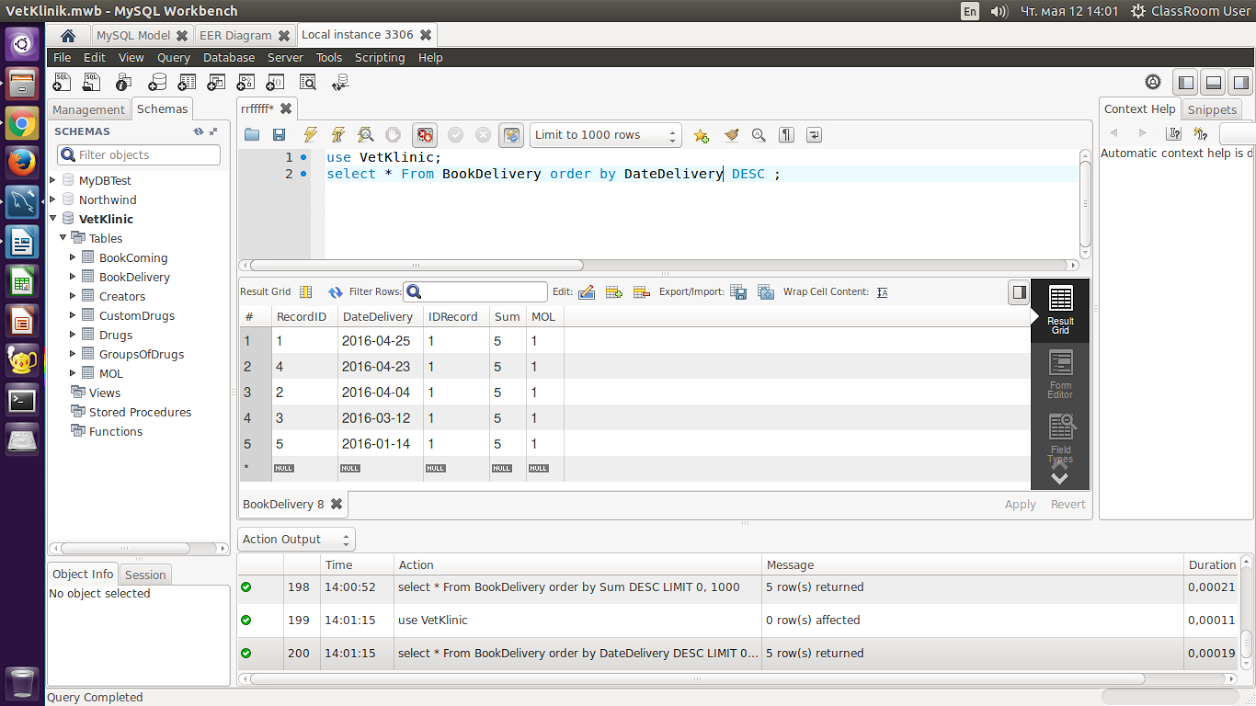


Рисунок 20 - Использование оператора «*DESC*»

6. Оператор *«INSERT INTO»* - вставляет строку в таблицу.

*use VetKlinic;*

*INSERT INTO Drugs (DrugsName, CreatorsID, GroupsOfDrudsID, Photo) Values (‘water’, 5, 4, ‘Pic7.pic’);*

Результат:

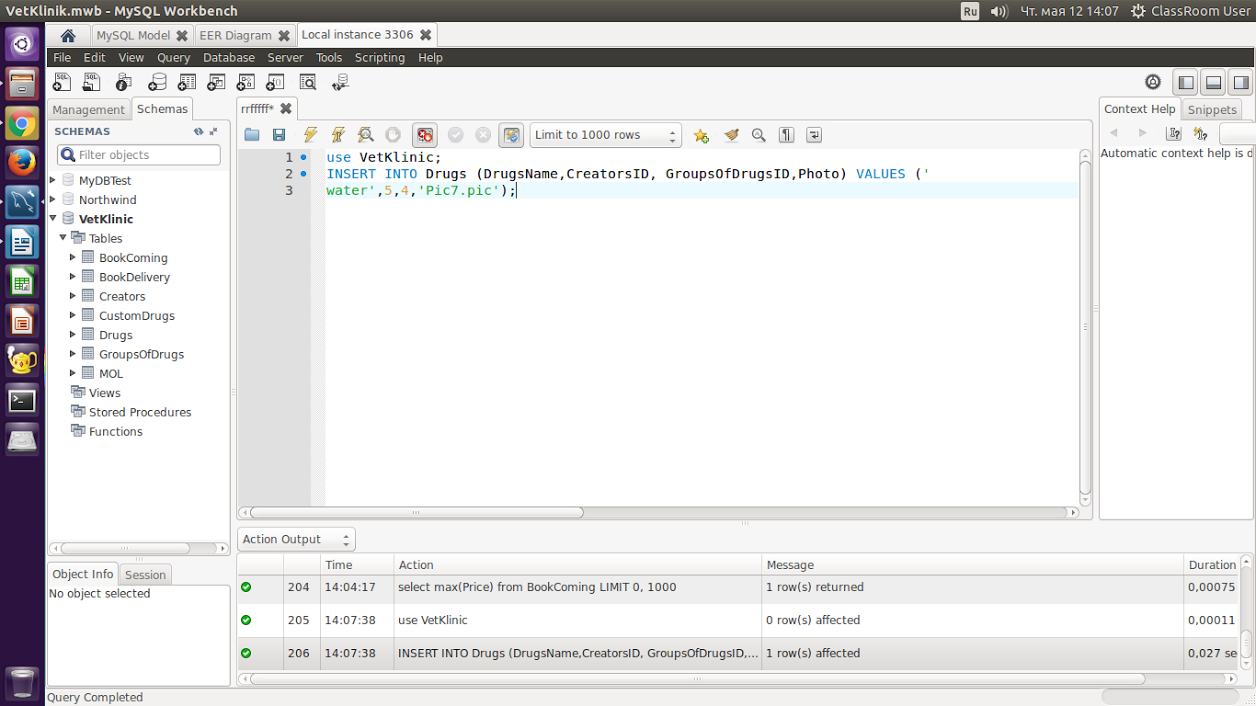


Рисунок 21 - Использование оператора *«INSERT INTO»*

Заключение

В данной контрольной работе разработана БД учета медицинских препаратов в ветеринарной клинике, которая позволяет автоматизировать обработку информации. В базе данных хранятся данные о препаратах, МОЛ и поставщиках, заказе препаратов и др.

В результате выполнения работы были выполнены следующие задачи:

1. Выделены информационные объекты (отношения, таблицы), которые были необходимы для работы ветеринарной клиники;
2. Создана схема хранения данных в рамках современной клиент-серверной СУБД с применением визуальных средств проектирования, а именно программы *MySQL Workbench*;
3. Сгенерирован код *SQL* БД ветеринарной клиники;
4. Разработаны и протестированы некоторые виды запросов на языке *SQL*.

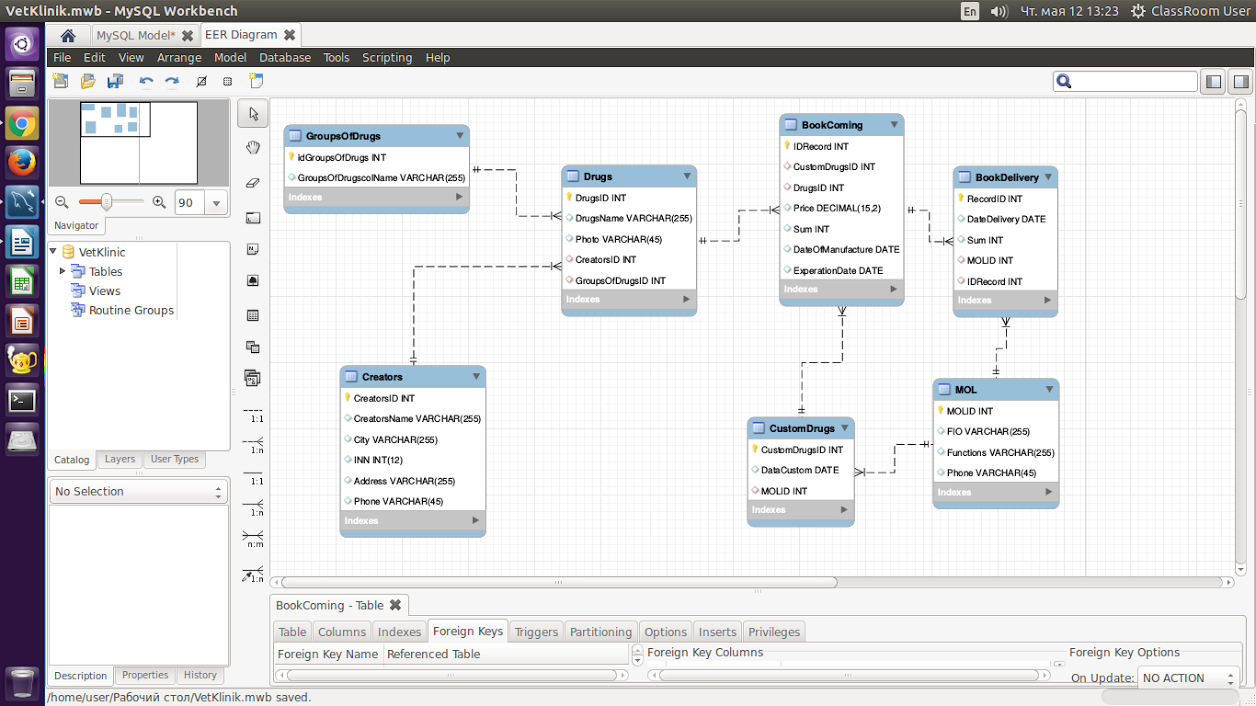
# Список литературы

1. Дейт, К. Дж. Введение в системы баз данных [Текст]: Пер. с англ. 7-е изд. / К. Дж. Дейт - М.: Издательский дом Вильямс, 2014. - 1072 с.
2. [www.mysql.com](http://www.mysql.com) [Электронный ресурс]
3. <https://github.com/volodink/irm-course> [Электронный ресурс]
4. [www.sql.ru](http://www.sql.ru) [Электронный ресурс]
5. <https://habrahabr.ru> [Электронный ресурс] – Форум. Сайт программистов

# 

Приложение 1

Схема БД



Приложение 2

Код *SQL*

drop database VetKlinic;

CREATE DATABASE VetKlinic;

use VetKlinik;

CREATE TABLE GroupsOfDrugs (

GroupsOfDrugsID int NOT NULL AUTO\_INCREMENT,

GroupsOfDrugsName VARCHAR(255),

PRIMARY KEY (GroupsOfDrugsID)

);

INSERT INTO GroupsOfDrugs (GroupsOfDrugsName) VALUES ('Vitamins');

INSERT INTO GroupsOfDrugs (GroupsOfDrugsName) VALUES ('Care products');

INSERT INTO GroupsOfDrugs (GroupsOfDrugsName) VALUES ('Medicament');

INSERT INTO GroupsOfDrugs (GroupsOfDrugsName) VALUES ('Painkillers');

CREATE TABLE Creators (

CreatorsID int NOT NULL AUTO\_INCREMENT,

CreatorsName VARCHAR(255),

City VARCHAR(255),

INN int(12),

Adress VARCHAR(255),

Phone VARCHAR(45),

PRIMARY KEY (CreatorsID)

);

INSERT INTO Creators (ProvidersName, City, INN, Adress, Phone) VALUES ('Biosintez','Penza','5874965487','45 Lenina St.', '547896');

INSERT INTO Creators (ProvidersName, City, INN, Adress, Phone) VALUES ('Vita','Pskov','7458961458','76 Mira St.', '748596');

INSERT INTO Creators(ProvidersName, City, INN, Adress, Phone) VALUES ('Vitaprom','Moscow','1452857496','65 Sokolniki St.', '124578');

INSERT INTO Creators (ProvidersName, City, INN, Adress, Phone) VALUES

('Novartis','Penza','5858965478','54 Pushkina St.', '965241');

INSERT INTO Creators (ProvidersName, City, INN, Adress, Phone) VALUES ('Nika','Perm','4785214593','89 Gagarina St.', '123698');

CREATE TABLE Drugs (

DrugsID int NOT NULL AUTO\_INCREMENT,

DrugsName VARCHAR(255),

CreatorsID int,

GroupsOfDrugsID int,

Photo VARCHAR(255),

PRIMARY KEY (DrugsID)

);

INSERT INTO Drugs (DrugsName,ProvidersID, GroupsOfDrugsID,Photo) VALUES ('Vata',2,2,'Pic1.pic');

INSERT INTO Drugs (DrugsName,ProvidersID, GroupsOfDrugsID,Photo) VALUES ('IOD',2,4,'Pic2.pic');

INSERT INTO Drugs (DrugsName,ProvidersID, GroupsOfDrugsID,Photo) VALUES ('Vizin',3,2,'Pic3.pic');

INSERT INTO Drugs (DrugsName,ProvidersID, GroupsOfDrugsID,Photo) VALUES ('Bint',2,5,'Pic4.pic');

CREATE TABLE MOL (

MOLID int NOT NULL AUTO\_INCREMENT,

FIO VARCHAR(255),

Functions VARCHAR(255),

Phone VARCHAR(45),

PRIMARY KEY (MOLID)

);

INSERT INTO MOL (FIO, Functions, Phone) VALUES ('Petrov Vitalii Yanovich', 'hirurg', '12547854789');

INSERT INTO MOL (FIO, Functions, Phone) VALUES ('Ivanov Ivan Ivanovich', 'hirurg', '1567567789');

INSERT INTO MOL (FIO, Functions, Phone) VALUES ('Ivanova Yana Viktorovna', 'medsestra', '6456546789');

CREATE TABLE CustomDrugs (

CustomDrugsID int NOT NULL AUTO\_INCREMENT,

DataCustom DATE,

MOLID int,

PRIMARY KEY (CustomDrugsID)

);

INSERT INTO CustomDrugs (DataCustom, MOL) VALUES ('2016-02-23', 1);

INSERT INTO CustomDrugs (DataCustom, MOL) VALUES ('2016-04-03', 3);

INSERT INTO CustomDrugs (DataCustom, MOL) VALUES ('2016-02-03', 3);

INSERT INTO CustomDrugs (DataCustom, MOL) VALUES ('2016-04-13', 2);

INSERT INTO CustomDrugs (DataCustom, MOL) VALUES ('2016-01-26', 1);

INSERT INTO CustomDrugs (DataCustom, MOL) VALUES ('2016-04-03', 2);

CREATE TABLE BookComing (

IDRecord int NOT NULL AUTO\_INCREMENT,

CustomDrugsID int,

DrugsID int,

Price numeric(15,2),

Sum int,

DateOfManufacture DATE,

ExpirationDate DATE,

PRIMARY KEY (IDRecord)

);

INSERT INTO BookComing (CustomDrugsID, DrugsID, Price, Sum, DateOfManufacture, ExpirationDate ) VALUES (2,2,10.00,3,'2015-01-26','2021-01-26');

INSERT INTO BookComing (CustomDrugsID, DrugsID, Price, Sum, DateOfManufacture, ExpirationDate ) VALUES (2,2,15.00,3,'2016-01-26','2022-01-26');

INSERT INTO BookComing (CustomDrugsID, DrugsID, Price, Sum, DateOfManufacture, ExpirationDate ) VALUES (3,3,150.00,3,'2015-01-26','2021-01-26');

INSERT INTO BookComing (CustomDrugsID, DrugsID, Price, Sum, DateOfManufacture, ExpirationDate ) VALUES (3,3,116.00,3,'2016-02-26','2022-02-26');

INSERT INTO BookComing (CustomDrugsID, DrugsID, Price, Sum, DateOfManufacture, ExpirationDate ) VALUES (2,2,18.00,3,'2016-02-24','2022-02-24');

CREATE TABLE BookDelivery (

RecordID int NOT NULL AUTO\_INCREMENT,

DateDelivery DATE,

IDRecord int,

Sum int,

MOL int,

PRIMARY KEY (RecordID)

);

INSERT INTO BookDelivery (DateDelivery, IDRecord, Sum, MOL) VALUES ('2016-04-25',1, 5, 1);

INSERT INTO BookDelivery (DateDelivery, IDRecord, Sum, MOL) VALUES ('2016-04-04',1, 5, 1);

INSERT INTO BookDelivery (DateDelivery, IDRecord, Sum, MOL) VALUES ('2016-03-12',1, 5, 1);

INSERT INTO BookDelivery (DateDelivery, IDRecord, Sum, MOL) VALUES ('2016-04-23',1, 5, 1);

INSERT INTO BookDelivery (DateDelivery, IDRecord, Sum, MOL) VALUES ('2016-01-14',1, 5, 1);