

Минобрнауки России
Юго-Западный государственный университет

Кафедра информационной безопасности

КУРСОВОЙ ПРОЕКТ

по дисциплине «Безопасность систем баз данных»
(наименование учебной дисциплины)
на тему «Разработка базы данных информационной системы дата майнинга
для интернет площадки ЕВАН»

Направление подготовки 10.03.01 Информационная безопасность
(код, наименование)

Автор проекта

Д. Хемраев
(инициалы, фамилия)

24.12.19г
(подпись, дата)

Группа ИБ 715

Руководитель проекта

А.Г. Спеваков
(инициалы, фамилия)

24.12.19г
(подпись, дата)

Проект защищен

24.12.19
(дата)

Оценка

отлично

Члены комиссии

24.12.19
(подпись, дата)

24.12.19
(подпись, дата)

24.12.19
(подпись, дата)

А.Г. Спеваков
(инициалы, фамилия)

М.В. Жуковская
(инициалы, фамилия)

А.В. Воронцов
(инициалы, фамилия)

Курск, 2019 г.

Минобрнауки России
Юго-Западный государственный университет

Кафедра информационной безопасности

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Студент Хемраев Д. шифр 17-06-0266 группа ИБ- 716
(фамилия, инициалы)

1. Тема: Разработка базы данных информационной системы дата майнинга для интернет площадки ЕВАУ

2. Срок представления проекта к защите « 29 » 12 2019 г.

3. Исходные данные для проектирования:

Цель: исследовать методы проектирования, создания, управления защиты баз данных с помощью СУБД

Задачи: – произвести поиск и сбор информации по методам проектирования, создания и управления базами данных при помощи СУБД;

– разработать структуру базы и заполнить таблицы;

– произвести поиск и сбор информации по методам защиты баз данных в СУБД;

– разработать и реализовать клиентское приложение для управления базами данных.

4. Содержание пояснительной записки курсового проекта:

4.1. Введение

4.2. Теоретическая часть

4.3. Разработка структуры базы данных

4.4. Защита от несанкционированного доступа

4.5. Клиентское приложение

4.6. Заключение

4.7. Список использованной литературы

4.8. Приложение А


4.9. Приложение Б

Руководитель проекта


(подпись, дата)

А.Г. Слеваков
(инициалы, фамилия)

Задание принял к исполнению

 24.09.19
(подпись, дата)

Д.Хемраев
(инициалы, фамилия)

Реферат

Данный текстовый документ имеет объём 39 страниц, количество иллюстраций составляет 20, приложение – 1. При работе над данной курсовой работой было использовано 11 источников литературы.

Перечень ключевых слов: базы данных, СУБД, триггеры, ключи, атрибуты.

Объектом разработки является программная реализация клиентской части базы данных информационной системы дата майнинга для интернет площадки EBAY

Целью проекта является разработка базы данных информационной системы дата майнинга для интернет площадки EBAY, изучить методы подключения к API таких сервисов, правильно отправить туда запросов, получение данных из этих сервисов. Правильная проектирование базы данных под такие сервисы, чтобы они были гибкими, надежными и безопасными.

Программный продукт позволяет пользователю просматривать всю информацию о товаре и следить за динамикой изменение цены товара.

Простота интерфейса создает комфортные условия для эксплуатации.

Содержание

Введение	5
1 Анализ систем управления баз данных	6
1.1 Проектирование базы данных	6
1.2 Представление базы данных в среде MySQL Workbench	9
1.3 Администрирование базы данных	10
2 Разработка структуры базы данных	14
2.1 Описание предметной области БД	14
2.2 Концептуальная модель (ER-диаграмма)	16
2.3 Заполнение БД в среде MySQL Workbench	17
2.4 Создание пользователей и наделение их соответствующими правами на управление базой	20
3 Защита от несанкционированного доступа	26
3.1. Настройка защиты RDS Amazon	27
4 Клиентское приложение	27
4.1 Тестирование пользовательского интерфейса	29
4.2 Тестирование интерфейса администратора	30
Заключение	31
Список использованных источников	32
Приложения	33
Приложения А	33
Приложения Б	37

Введение

Настоящее время большинство организаций используют различные базы данных для автоматизации процессов автоматизации обработки информации, удобства её эксплуатации, повышения надёжности хранения данных, сокращения числа возможных ошибок в работе, предотвращения избыточности данных, а также целях повышения производительности. Состав баз данных и их размер определяются направлением деятельности конкретного предприятия и его мощностью.

Связи с тем, что современные информационные системы оперируют большими объемами и сложными структурами данных, средства управления базами данных можно выделить в отдельную группу информационных систем, отвечающих за управление сложными структурированными данными.

Сегодня можно с уверенностью утверждать, что решение широкого круга задач в любой сфере деятельности человека сегодня практически невозможно без использования оперативно управляемых баз данных.

Цель курсового проекта: исследовать методы проектирования, создания, управления и защиты баз данных с помощью СУБД.

Для достижения цели были поставлены следующие задачи:

- 1) произвести поиск и сбор информации по методам проектирования,
- 2) создания и управления базами данных при помощи СУБД;
- 3) разработать структуру базы и заполнить таблицы;
- 4) произвести поиск и сбор информации по методам защиты баз данных в СУБД;
- 5) разработать и реализовать клиентское приложение для управления базами данных.

1 Анализ систем управления баз данных

1.1 Проектирование базы данных

Базы данных характеризует довольно неоднозначная и широкая терминология. По этой причине термины и определения вводятся последовательно, так же, как и описываются особенности существующих технологий.

Предметная область (ПО) – часть реального мира (например: вокзал, студенческая группа, страна, деканат, кинотеатр, бухгалтерия, порт и т.д.). Конкретное состояние ПО фиксируется в данных. ПО описывается с какой-то степенью точности, следовательно, она является моделью, т.е. создается модель предметной области (МПО). ПО моделируется при помощи информационных объектов и связей между ними, следовательно, МПО – информационная модель .

Данные – информация, подготовленная в определенном формате, для определенных целей.

Информационный объект:

- однозначно выделяемый объект реального мира;
- некоторое понятие или процесс, относящийся к предметной области;
- абстракция множества предметов реального мира такая, что:
 - а) все предметы (экземпляры) в этом множестве имеют одни и те же характеристики (свойства);
 - б) все экземпляры подчинены и согласовываются с одним и тем же набором правил и линий поведения.

Объект – это нечто существующее и различное. Иногда вместо слова «объект» употребляют слово «сущность». Каждый объект имеет уникальное имя.

При идентификации (выделении) ПО теория рекомендует составить описание объекта, т.е. краткое информативное утверждение, которое позволяет установить: является ли реальный предмет экземпляром объекта.

Каждый объект может быть задан при помощи наборов свойств. Свойства, которые справедливы для всех возможных экземпляров объекта, т.е. являются общими, называются атрибутами объекта. Т.к. объект может иметь несколько свойств, то он задается набором своих атрибутов.

Элемент данных (ЭД) – то, при помощи чего определяется или задается свойство данных объекта. ЭД имеет уникальное имя и совокупность значений, связанных с этим именем. Это множество называется доменом.

Типичные домены – это множество целых или рациональных чисел, множество символьных строк.

Элементы данных делятся на ключи и атрибуты. Если речь идет о конкретном объекте, то среди множества его атрибутов существует некоторое его подмножество, значения атрибутов которого однозначно определяют каждый экземпляр данного объекта. Минимальное множество элементов данных, однозначно определяющих запись об объекте, называется ключом. Множество атрибутов, этот ключ включающих – супер ключом. В то же время элементы, которые не являются ключами, называют атрибутами.

В большинстве задач, использующих базы данных, записи определяются номером строки (или записи). Такой атрибут является уникальным, т.е. в таблице он не повторяется. Если рассматривать множество атрибутов из одного элемента, то такой атрибут можно использовать как первичный ключ – ключ, используемый для идентификации данных. Ключи, не являющиеся первичными, называются вторичными. Ключи, содержащие несколько элементов, могут быть составными, сложными, сцепленными.

Запись–совокупность значений элементов данных, которые описывают конкретный экземпляр объекта.

Набор записей – совокупность записей об объекте для всех экземпляров объектов одного типа (это и есть объект).

Если два элемента данных двух конкретных информационных объектов могут использоваться вместе, тогда говорят, что между такими объектами есть связь.

Связь – отношение между элементами данных или объектами, которые могут использоваться совместно в информационной модели. Связь можно выявить в предметной области на основе анализа взаимодействий элементов данных или 6 объектов. Также она возникает, когда есть соотношение «часть – целое», «класс – подкласс», «вид – подвид».

Файл данных – поименованный, упорядоченный набор записей об объектах одного типа (или об объектах разного типа).

Выделяют несколько свойств баз данных, некоторые из которых представлены и описаны ниже:

- 1) избыточность;
- 2) независимость данных и программ;
- 3) взаимосвязь данных и их структурируемость;
- 4) ответ на непредвиденные запросы;
- 5) изменение предметной области.

Свойство избыточности исключает ситуацию, приводящую к противоречивости данных. Например, если в нескольких объектах хранится информация об одном и том же объекте или связи. Такая ситуация называется избыточностью, и, если при удалении какого-нибудь объекта сведения о нем и его связях теряются, то избыточность минимальна. В базах данных такие объекты интегрируют в одну систему и выделяют в них одинаковые части.

Свойство независимости определяет независимость реализации от входных данных.

Взаимосвязь данных определяет порядок следования записей и их структуру, при этом записи должны быть формализованы.

На непредвиденные запросы система данных и их набор должна отвечать. Предметная область должна быть доступна для изменения без изменения существующих файлов.

Базой данных называют некоторую совокупность данных и связей между ними, удовлетворяющую вышеперечисленным условиям.

Системой управления базами данных (СУБД) называют программно-технические средства для добавления новых или модификации существующих данных, а также для их поиска (управляющий способ), и поддерживающие при этом все свойства баз данных.

Программные средства, СУБД, заполненную базу данных (а иногда и прикладные программы) называют банком данных.

1.2 Представление базы данных в среде MySQL Workbench

При создании структуры базы данных составляется структура входящих в нее таблиц и связей между ними. В СУБД за связи между таблицами отвечают индексы, формируемые из значений одной или нескольких записей таблицы и создаваемые с целью повышения производительности запросов. Также индексы используются при поиске записей в базе данных, так как строки в таблице хранятся в произвольном порядке, а поиск путем последовательного перебора в случае таблиц с большим количеством строк может занимать огромное время. Так как индекс имеет структуру сбалансированного дерева, с его использованием происходит ускорение работы. Индексы, как и аргументы таблицы, могут быть уникальными (именно за счет них реализуются связи между таблицами), т.е. исключающими возможность повторяющихся значений в таблице.

Индексы также разделяют на кластерные и не кластерные. Кластерный индекс у таблицы может быть только один, и, при его наличии, строки таблицы строго связаны с элементами индекса, т.е. хранятся в заданном индексом порядке, что значительно ускоряет доступ к данным при выполнении запросов к таким таблицам.

Отсутствие кластерного индекса делает таблицу неупорядоченной, поэтому индекс, созданный для этой таблицы, содержит только указатели на

строки таблицы. Такой индекс называется некластерным. При выборке из таблицы с некластерным индексом необходимо по крайней мере еще одно обращение к диску для получения самой записи.

Чтобы улучшить производительность базы индексы создают на тех полях таблицы, которые чаще используются в запросах, при этом индексы могут включать несколько полей, при этом для одной таблицы можно создать несколько индексов.

Чем больше индексов, тем медленнее выполняются операции добавления, удаления и обновления, поскольку сами индексы необходимо обновлять. Кроме того, индексы занимают дополнительный объем памяти.

Модель Сущность-Связь (ER-модель) – это модель данных, позволяющая описывать концептуальные схемы. Она предоставляет графическую нотацию, основанную на блоках и соединяющих их линиях, с помощью которых можно описывать объекты и отношения между ними какой-либо другой модели данных. ER-модель удобна при проектировании информационных систем, баз данных, архитектур компьютерных приложений, и других систем. С её помощью можно выделить ключевые сущности, присутствующие в модели, и обозначить отношения, которые могут устанавливаться между этими сущностями. Важно отметить, что сами отношения также являются сущностями, что позволяет устанавливать отношения на множестве самих отношений.

На этапе перехода к реализации данной ER-диаграммы в виде реальной информационной системы или программы, происходит отображение ER-модели в более детальную модель данных реляционной (объектной, сетевой, логической, или др.) базы данных, которая называется физической моделью данных по отношению к исходной ER-диаграмме.

1.3 Администрирование базы данных

Для администрирования базы данных используется механизм определения прав пользователя СУБД Microsoft SQL Server 2017.

В этой СУБД используется два уровня защиты. Первый – проверка подлинности пользователя, второй – авторизация, или проверка прав доступа.

Во время проверки подлинности пользователя определяется, может ли пользователь подключиться к серверу. После прохождения проверки подлинности определяется, какие действия в базе данных он сможет выполнять (проверка прав доступа).

Проверку подлинности также называют проверкой идентификатора учетной записи (ИУЗ). ИУЗ используется при подключении к серверу, он вводится пользователем и проверяется сервером. Именно ИУЗ обеспечивает пользователю право подключаться к серверу. В СУБД SQL Server 2017 существуют два вида проверки подлинности: средствами самого сервера и средствами системы Windows. Выбор вида для текущего соединения происходит при подключении к серверу.

В СУБД SQL Server 2017 пользователем, имеющим право на доступ ко всем элементам и функциям базы данных, является системный администратор (sa). При ИУЗ средствами системы выполняется вход от имени системы, при этом пользователь будет обладать правами администратора.

Администратор может предоставить пользователям и группам пользователей право устанавливать соединение с сервером. Если проверка подлинности выполняется средствами системы, то для проверки авторизации пользователя используются средства Windows. Сервер же идентифицирует пользователя по имени его сетевой учетной записи, и разрешает или запрещает соединение, не требуя его отдельной регистрации в SQL Server 2017. Такой тип соединения называется доверенным.

При проверке подлинности средствами сервера администратор БД может создавать учетные записи пользователей SQL Server, создавая учетные записи, не связанные с учетными записями Windows NT 4.0/2000. В этом случае сервер сам проверяет пользователя, производя поиск такой учетной записи в SQL Server 2017.

При запуске Microsoft SQL Server 2017 по умолчанию используется режим проверки подлинности средствами системы. Также СУБД позволяет работать в смешанном режиме, т.е. пользователь должен пройти проверку системы и сервера одновременно.

Далее СУБД определяет права авторизованного пользователя выполнять различные действия. Сам по себе ИУЗ не дает прав доступа к различным объектам базы, он позволяет перейти к этапу авторизации. Такой механизм защиты не дает пользователю автоматический доступ ко всем базам СУБД.

Как правило, администратор БД должен сопоставить идентификатор учетной записи пользователя идентификатору пользователя в БД, прежде чем пользователь, подключившийся с использованием этого идентификатора учетной записи, получит доступ или сможет выполнить какие-либо действия в этой БД. Администратор БД определяет права доступа к объектам (таким как таблицы, представления и хранимые процедуры) в БД для всех учетных записей пользователей.

Если учетная запись пользователя на сервере не связана ни с одной учетной записью пользователя в БД, она связывается с идентификатором учетной записи гостя в этой БД (если такой идентификатор существует). Если в БД присутствует учетная запись пользователя гостя, права подключающегося пользователя ограничиваются правами пользователя гостя. Если в БД отсутствует учетная запись пользователя гостя, подключающийся пользователь не получит доступ к БД до тех пор, пока его учетная запись на сервере не будет связана с учетной записью БД. По умолчанию учетная запись гостя отключена.

Механизм ролей позволяет объединять пользователей по группам с определенными правами. СУБД Microsoft SQL Server 2017 имеет встроенные роли для базы данных и для самого сервера. Права для них заранее настроены. При этом системный администратор может создавать новые роли. В рассматриваемой СУБД присутствует роль public, предоставляющая начальные права. Дополнительные права предоставляют в явном виде.

Фиксированные роли базы данных задаются на уровне базы данных и предусмотрены в каждой базе данных. Элементы ролей базы данных db_owner и db_securityadmin могут управлять членством в фиксированных ролях базы данных, однако только члены роли базы данных db_owner могут добавлять членов в фиксированную роль базы данных db_owner. Имеются следующие фиксированные роли базы данных: db_accessadmin, db_backupoperator, db_datareader, db_datawriter, db_ddladmin, db_denydatareader, db_denydatawriter, db_owner, db_securityadmin.

2 Разработка структуры базы данных

2.1 Описание предметной области БД

База данных будет разрабатываться на архитектуре сервера – MySQL Server.

Опишем предметную область базы данных:

1. Продукт – информация о товаре хранящейся на сервере
2. Продавец – человек, который продает товар.
3. Категория – у товара есть категория.
4. Цена – у товара есть цена.
5. Изображение – у товара есть изображение,

В таблице 2.1 представлено описание объекта «product» и его полей.

Таблица 2.1 – Объект «product»

Meaning	Designation	Example	Type
id	id	4265468464	INT, PRIMARY KEY, NOT NULL
Название продукта	title	mice, trackballs & touchpads	VARCHAR(75), NOT NULL
Цена продукта	price	8	FLOAT, NOT NULL
Ссылка продукта	url	https://www.ebay.com/itm/new-genuine-hp-usb-optical-mouse-672652-001-black-wired-2-button-scroll-/323981514951	VARCHAR(100), NOT NULL

В таблице 2.2 представлено описание объекта «seller_product» и его полей.

Таблица 2.2 – Объект «seller_product»

Meaning	Designation	Example	Type
id		3	INT, PRIMARY KEY, NOT NULL
Имя продавца	name_seller	peneywise	VARCHAR(45), NOT NULL
По уникальному идентификатору имеем возможность найти объект.	id_product	4265468464	INT, NOT NULL

В таблице 2.3 представлено описание объекта «category_product» и его полей.

Таблица 2.3– Объект «category _product »

Meaning	Designation	Example	Type
id	id	3	PRIMARY KEY, NOT NULL, INT
Наименование категории	Name_categ	mice, trackballs & touchpads	VARCHAR(45), NOT NULL

В таблице 2.4 представлено описание объекта «image _product» и его полей.

Таблица 2.4 – Объект «image_product »

Meaning	Designation	Example	Type
id	id	2	PRIMARY KEY, NOT NULL, INT(5)
URL изображения	url_product	https://i.ebayimg.com/images/g/eeoAAOSwbqddzvn6/s-11600.jpg	VARCHAR(30), NOT NULL
По уникальному идентификатору имеем возможность найти объект.	id_product	4265468464	INT, NOT NULL

В таблице 2.4 представлено описание объекта «price _product» и его полей.

Таблица 2.4 – Объект «price_product »

Meaning	Designation	Example	Type
id	id	2	PRIMARY KEY, NOT NULL, INT(5)
Цена	Price	2500	VARCHAR(30), NOT NULL
Дата Время	date	23.12.2019 00:23:12	DATETIME
По уникальному идентификатору имеем возможность найти объект.	id_product	4265468464	INT, NOT NULL

Система функциональных зависимостей.

«product»

(1) У каждого продукта должен быть свой индивидуальный номер

(2) F1: id-> title, about, price,url

(3) K1= id

« seller_product»

(1) У каждого продавца должен быть свой индивидуальный номер

(2) F1: id-> name_seller, id_product

(3) K1= id

« category_product»

У каждой категории должен быть свой индивидуальный номер

(1) F1: id -> name_categ, id_product

(2) K1= id

« image_product»

(1) У каждого изображения индивидуальный номер

(2) F1: id -> url_product, id_product

(3) K1= id

« price_product»

(1) У каждой цены продукта индивидуальный номер

(3) F1: id -> price, date , id_product

(2) K1= id

2.2 Концептуальная модель (ER-диаграмма)

С помощью программы Mysql Workbench составляем ER-диаграмму нашей базы данных.

Модель сущность-связь (ER-модель) позволяет описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы

данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма сущность-связь (ER-диаграмма) (англ. Entity-relationship diagram, ERD).

На рисунке 2.1 представлена ER-диаграмма базы данных удаленного онлайн-тестирования. В приложении А представлен sql – скрипт создания таблиц.

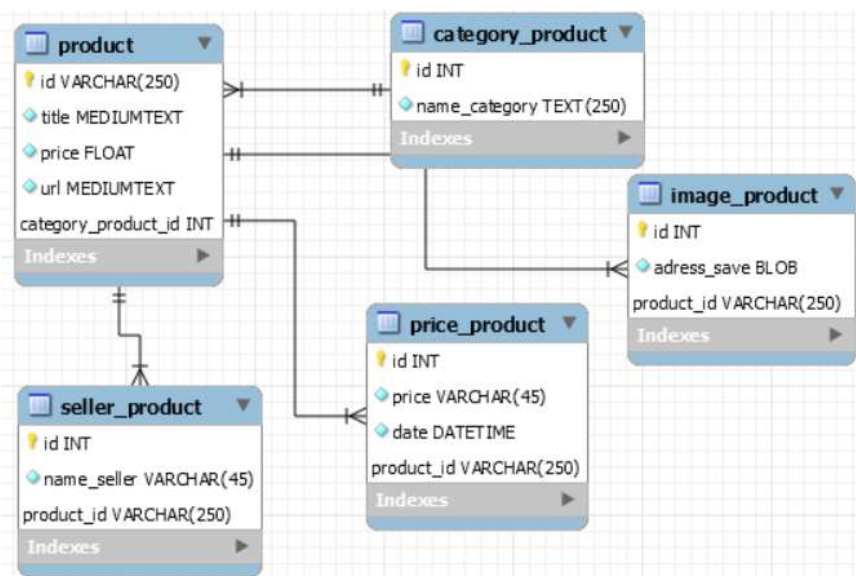


Рисунок 2.1 – Концептуальная модель (ER-диаграмма)

2.3 Заполнение БД в среде MySQL Workbench

Если после полученная диаграмма соответствует требованиям нашей БД, то переходим к заполнению таблиц конкретными значениями. На рисунке 2.1 представлено открытие таблиц в режиме просмотра значений/редактирования.

Скрипт для заполнения столбцов таблицы category_product:

```
INSERT INTO `api_ebay`.`category_product`
(`id`,`name_category`) VALUES(<{id: }>,<{name_category: }>);
```



id	name_category
1	Apple Laptops
2	Cell Phones & Smartphones
3	Video Game Consoles
4	Mice, Trackballs & Touchpads
5	Tablets & eBook Readers
6	Cases, Covers, Keyboard Folios
7	Other Tablet & eBook Accs
8	Harry Potter
9	Necklaces & Pendants
10	Video Games
11	DVDs & Blu-ray Discs

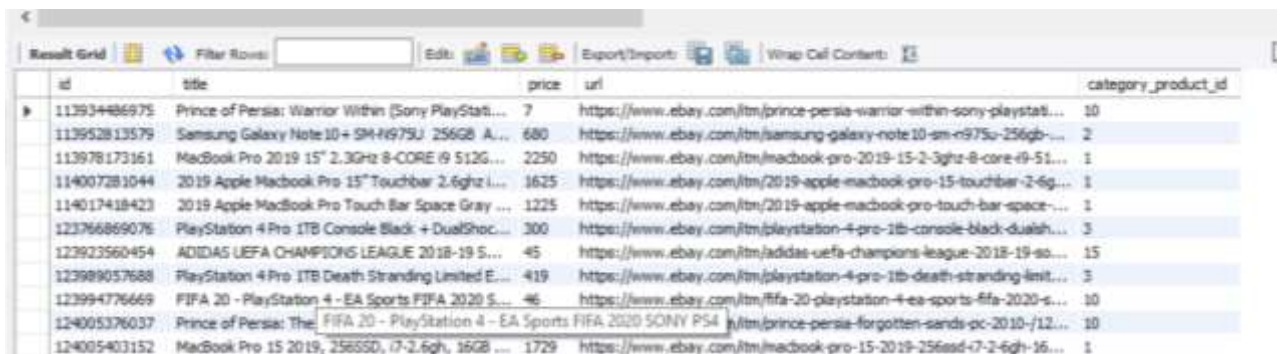
Рисунок 2.2 – Заполненная таблица category_product

Для выборки данных:

```
SELECT `category_product`.`id`, `category_product`.`name_category`
FROM `api_ebay`.`category_product`;
```

Скрипт для заполнения столбцов таблицы product:

```
INSERT INTO `api_ebay`.`product` (`id`,`title`,`price`,`url`,
`category_product_id`) VALUES (<{id: }>,<{title: }>,<{price: }>,<{url:
}>,<{category_product_id: }>);
```



id	title	price	url	category_product_id
113934486975	Prince of Persia: Warrior Within (Sony PlaySta...	7	https://www.ebay.com/itm/prince-persia-warrior-within-sony-playsta...	10
113952813579	Samsung Galaxy Note 10+ SM-N975U 256GB A...	680	https://www.ebay.com/itm/samsung-galaxy-note-10-sm-n975u-256gb-...	2
113978173161	MacBook Pro 2019 15" 2.3GHz 8-CORE i9 512G...	2250	https://www.ebay.com/itm/macbook-pro-2019-15-2-3ghz-8-core-i9-51...	1
114007281044	2019 Apple MacBook Pro 15" Touchbar 2.6ghz i...	1625	https://www.ebay.com/itm/2019-apple-macbook-pro-15-touchbar-2-6g...	1
114017418423	2019 Apple MacBook Pro Touch Bar Space Gray ...	1225	https://www.ebay.com/itm/2019-apple-macbook-pro-touch-bar-space-...	1
123766869076	PlayStation 4 Pro 1TB Console Black + DualSho...	300	https://www.ebay.com/itm/playstation-4-pro-1tb-console-black-dualsh...	3
123923560454	ADIDAS UEFA CHAMPIONS LEAGUE 2018-19 S...	45	https://www.ebay.com/itm/adidas-uefa-champions-league-2018-19-so...	15
123989057688	PlayStation 4 Pro 1TB Death Stranding Limited E...	419	https://www.ebay.com/itm/playstation-4-pro-1tb-death-stranding-lim...	3
123994776669	FIFA 20 - PlayStation 4 - EA Sports FIFA 2020 S...	46	https://www.ebay.com/itm/fifa-20-playstation-4-ea-sports-fifa-2020-s...	10
124005376037	Prince of Persia: The Sands of Time (Sony PS4 ...)	1729	https://www.ebay.com/itm/prince-persia-forbidden-sands-ps4-2010-12...	10
124005403152	MacBook Pro 15 2019, 256SSD, i7-2.6gh, 16GB ...	1729	https://www.ebay.com/itm/macbook-pro-15-2019-256ssd-i7-2-6gh-16...	1

Рисунок 2.3 – Заполненная таблица product

Для выборки данных:

```
SELECT `product`.`id`,`product`.`title`,`product`.`price`,
`product`.`url`,
`product`.`category_product_id` FROM `api_ebay`.`product`;
```

Скрипт для заполнения столбцов таблицы image_product:

```
INSERT INTO `api_ebay`.`image_product` (`id`,`adress_save`,`product_id`)
VALUES (<{id: }>,<{adress_save: }>,<{product_id: }>);
```

id	adress_save	product_id
1	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\143469151754\143469151754\143469151754-4300.jpg	143469151754
2	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\392591686224\392591686224\392591686224-4300.jpg	392591686224
3	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\143399429306\143399429306\143399429306-4300.jpg	143399429306
4	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\143468093745\143468093745\143468093745-4300.jpg	143468093745
5	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\143389160099\143389160099\143389160099-4300.jpg	143389160099
6	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\143449925313\143449925313\143449925313-4300.jpg	143449925313
7	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\133154284340\133154284340\133154284340-4300.jpg	133154284340
8	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\113978173161\113978173161\113978173161-4300.jpg	113978173161
9	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\202851272854\202851272854\202851272854-4300.jpg	202851272854
10	D:\projects\курсовые\базы_данных\client_EBAY\client_EBAY\bin\Debug\images\Apple Laptops\202851272854\202851272854\202851272854-4300.jpg	202851272854

Рисунок 2.4 – Заполненная таблица product

Для выборки данных:

`SELECT 'id', 'adress_save', 'product_id' FROM`

``api_ebay`.`image_product`;`

Таким же образом заполняем остальные таблицы.

```

1 * SELECT "price_product"."id",
2       "price_product"."price",
3       "price_product"."date",
4       "price_product"."product_id"
5 FROM "api_ebay"."price_product";
6

```

id	price	date	product_id
1	1149	2019-12-17 00:00:00	143469151754
2	2499	2019-12-17 00:00:00	392591686224
3	989	2019-12-17 00:00:00	143399429306
4	1129	2019-12-17 00:00:00	143468093745
5	999	2019-12-17 00:00:00	143448701704
6	749	2019-12-17 00:00:00	143389160099
7	1099	2019-12-17 00:00:00	143449925313
8	1540	2019-12-17 00:00:00	133154284340
9	2250	2019-12-17 00:00:00	113978173161
10	2100	2019-12-17 00:00:00	202851272854
11	849	2019-12-17 00:00:00	143469151751

Рисунок 2.5 – Заполненная таблица price_product

```

1 * INSERT INTO "api_ebay"."seller_product"
2   ('id',
3    'name_seller',
4    'product_id')
5  VALUES
6    ({{id: }},
7     {{name_seller: }},
8     {{product_id: }});

```

id	price	date	product_id
1	1149	2019-12-17 00:00:00	143469151754
2	2499	2019-12-17 00:00:00	392591686224
3	989	2019-12-17 00:00:00	143399429306
4	1129	2019-12-17 00:00:00	143468093745

Рисунок 2.6 – Заполненная таблица seller_product

```
SELECT `product`.`id`, `product`.`title`, `product`.`price`,
`product`.`url`,`product`.`category_product_id`FROM
`api_ebay`.`product`, `api_ebay`.`price_product`where `product`.`id`=
price_product.`product id`and `price_product`.`date` >'2019-12-01';
```

Этот скрипт выполняет запрос на выборку о товарах которые были добавлены после 1-го декабря.

id	title	price	url	category_product_id
133246745282	2019 2020 Topps UEFA Champions League Mat...	\$5	https://www.ebay.com/itm/2019-2020-topps-uefa-champions-league-...	13
283366817143	Jedon Sancho 2019 Topps Chrome Uefa Champ...	1	https://www.ebay.com/itm/jedon-sancho-2019-topps-chrome-uefa-ch...	13
333230674681	UEFA Champions League patch kit- FC Barcel...	13	https://www.ebay.com/itm/uefa-champions-league-patch-kit-fc-barcel...	14
254408583804	2019-20 TOPPS UEFA CHAMPIONS LEAGUE STL...	18	https://www.ebay.com/itm/2019-20-topps-uefa-champions-league-stl...	13
223774579198	FIFA 20 Xbox One Brand New Sealed FIFA 2020	40	https://www.ebay.com/itm/fifa-20-xbox-one-brand-new-sealed-fifa-20...	30
223742901401	FIFA 20 - Sony PlayStation 4 - EA Sports FIFA 2...	41	https://www.ebay.com/itm/fifa-20-sony-playstation-4-ea-sports-fifa-2...	30
323994776669	FIFA 20 - PlayStation 4 - EA Sports FIFA 2020 S...	46	https://www.ebay.com/itm/fifa-20-playstation-4-ea-sports-fifa-2020-s...	30
294451772471	FIFA 20 PS4 2020 Standard Edition Brand New ...	38	https://www.ebay.com/itm/fifa-20-ps4-2020-standard-edition-brand-n...	30
202853675191	FIFA 20 PS4 for Sony PlayStation 4 Soccer Gam...	40	https://www.ebay.com/itm/fifa-20-ps4-sony-playstation-4-soccer-gam...	30
254361753548	Nintendo Switch FIFA 2020 Brand New Factory ...	48	https://www.ebay.com/itm/nintendo-switch-fifa-2020-brand-new-facto...	30
202852853641	FIFA 20 Xbox One for Microsoft XB1 2019 2020 ...	40	https://www.ebay.com/itm/fifa-20-xbox-one-microsoft-xb1-2019-2020...	30
303361347929	FIFA 20 - PlayStation 4 - EA Sports FIFA 2020 S...	44	https://www.ebay.com/itm/fifa-20-playstation-4-ea-sports-fifa-2020-s...	30

Рисунок 2.7– Полученные данные

2.4 Создание пользователей и наделение их соответствующими правами на управление базой

Пользователь «Superuser» может добавить данные в базу данных. Создаем пользователя Superuser (рисунок 2.8).

1)Переходим Administration->Users and Privileges

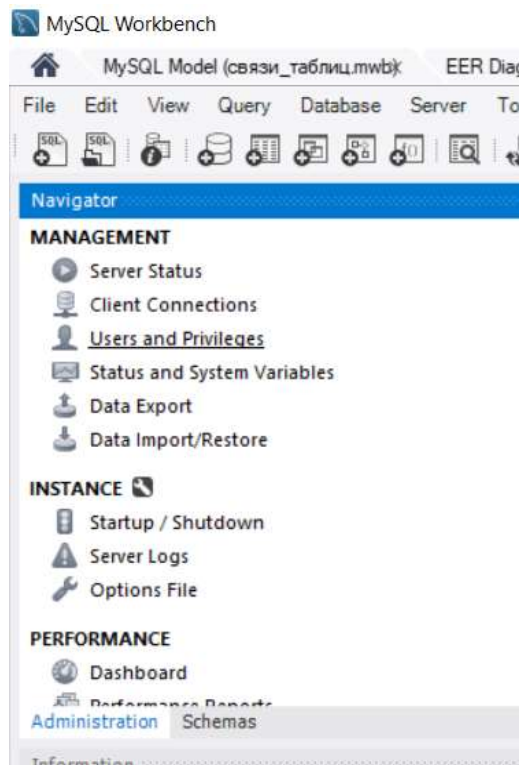


Рисунок 2.8-Панель управление

2) Вводим логин и пароль нового пользователя (рисунок 2.9).

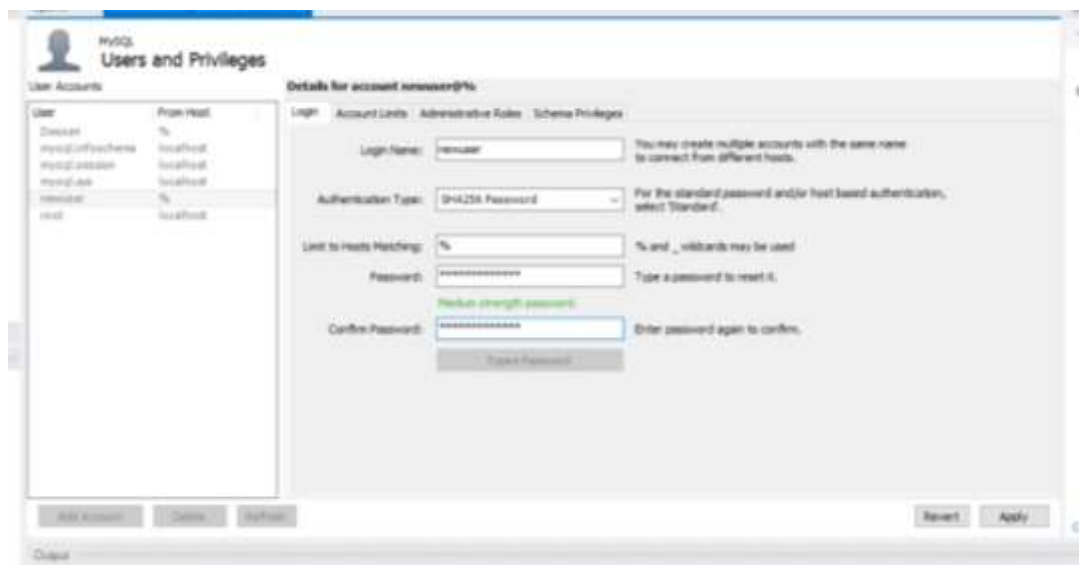


Рисунок 2.9 – Создание пользователя

3) Определяем для пользователя количество запросов за час (рисунок 2.10).

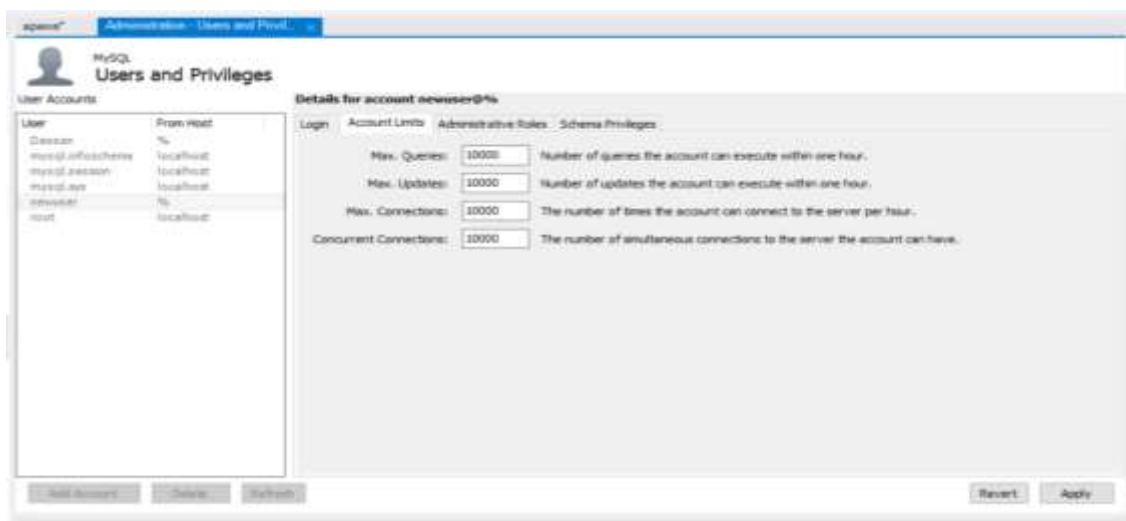


Рисунок 2.10 -Количество запросов на одного пользователя

4) Определяем роли пользователя

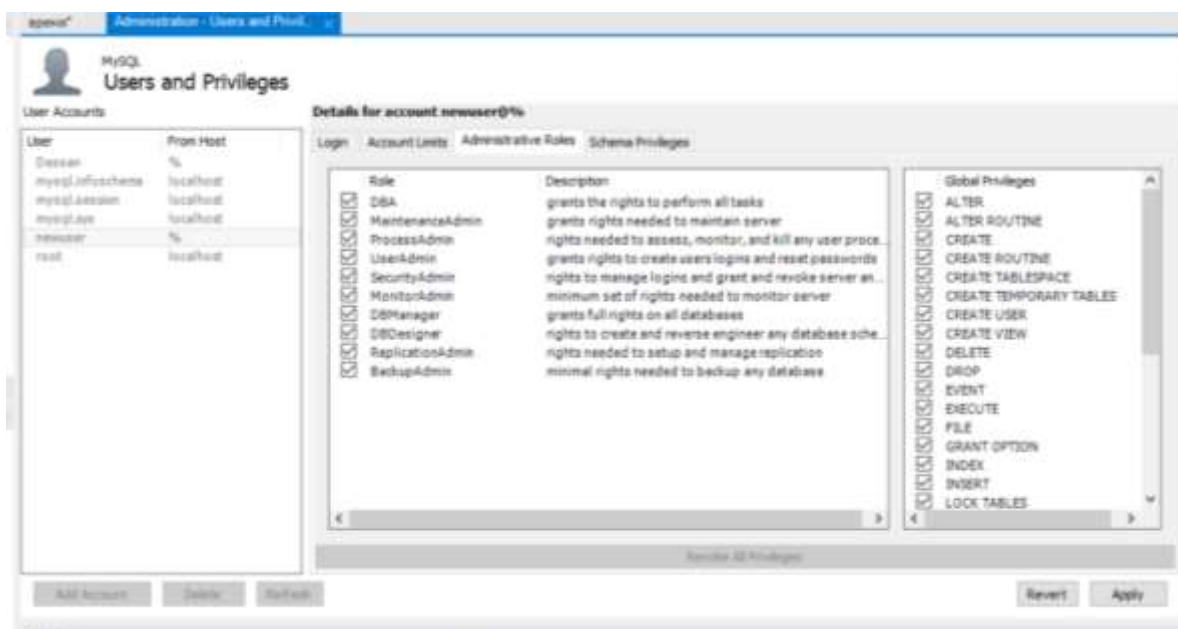


Рисунок 2.11- привилегии Superuser

5) Устанавливаем какие таблицы может использовать пользователь (рисунок 2.12).

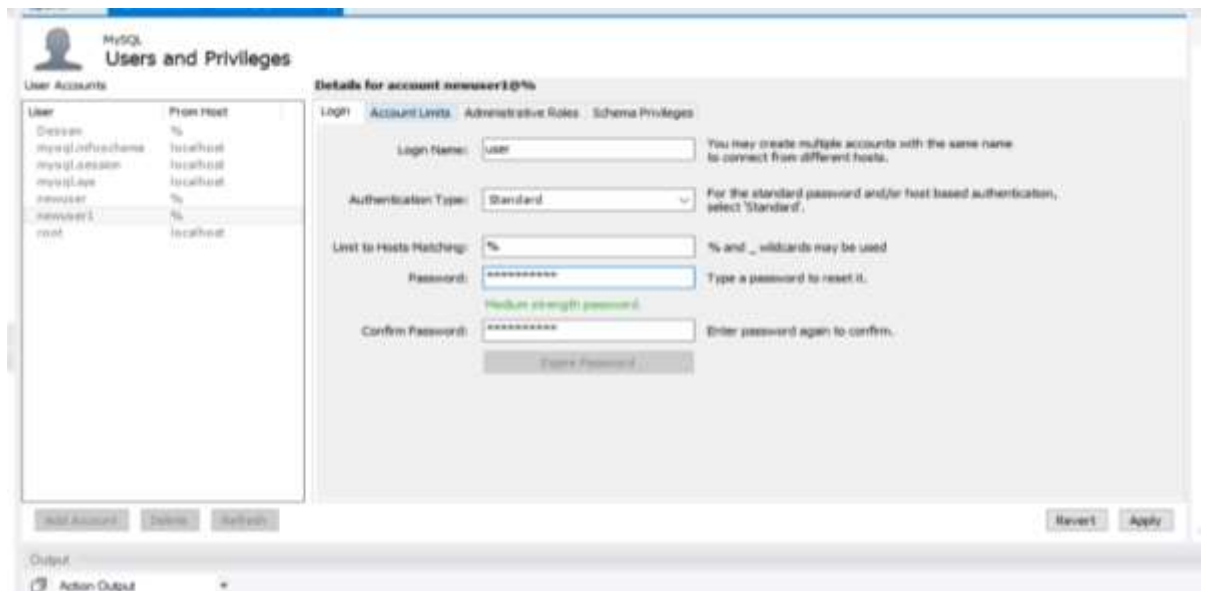


Рисунок 2.14-Создание пользователя

3) Определяем для пользователя количество запросов за час(рисунок 2.15).

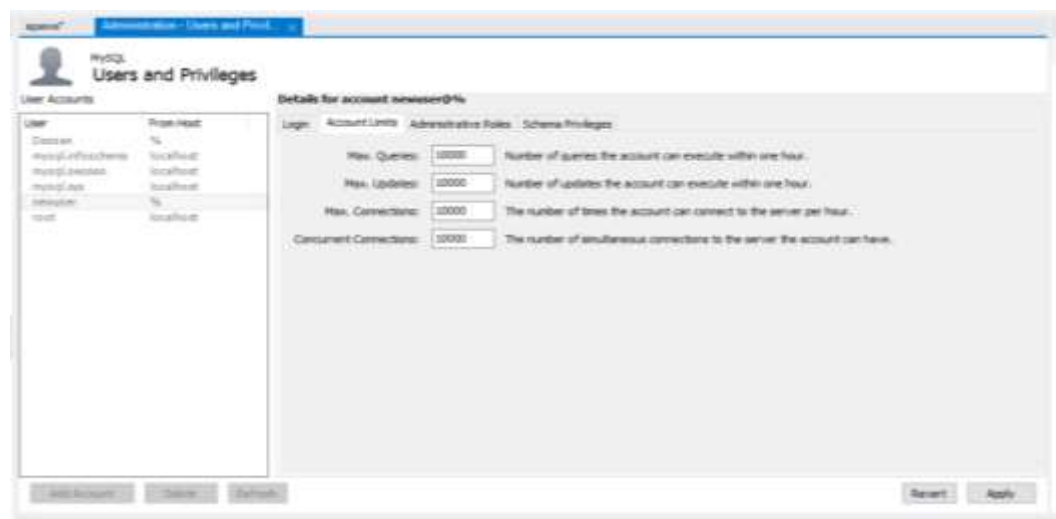


Рисунок 2.15- количество запросов

4) Определяем роли пользователя(рисунок 2.16).

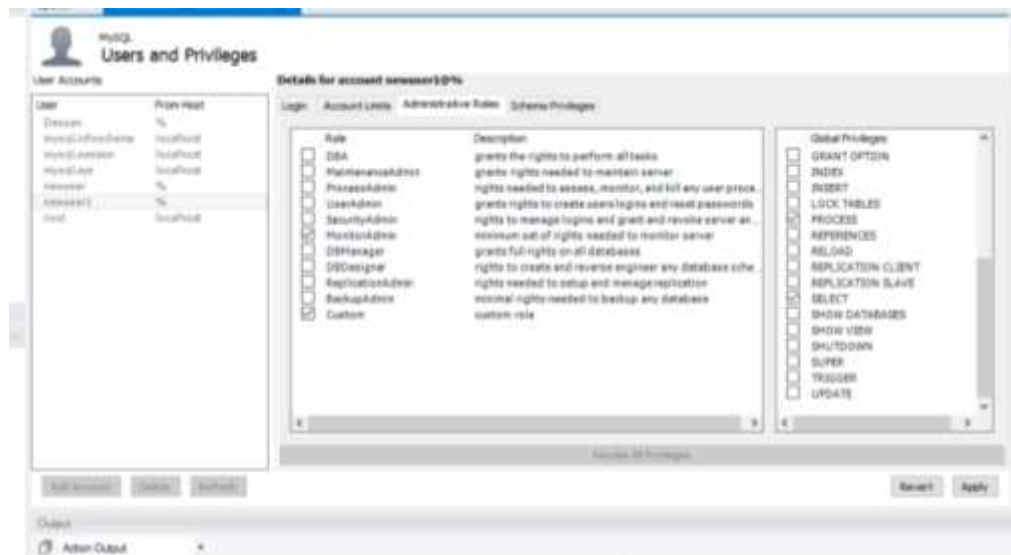


Рисунок 2.16-привилегии пользователя

5) Устанавливаем какие таблицы может использовать пользователь (рисунок 2.17).

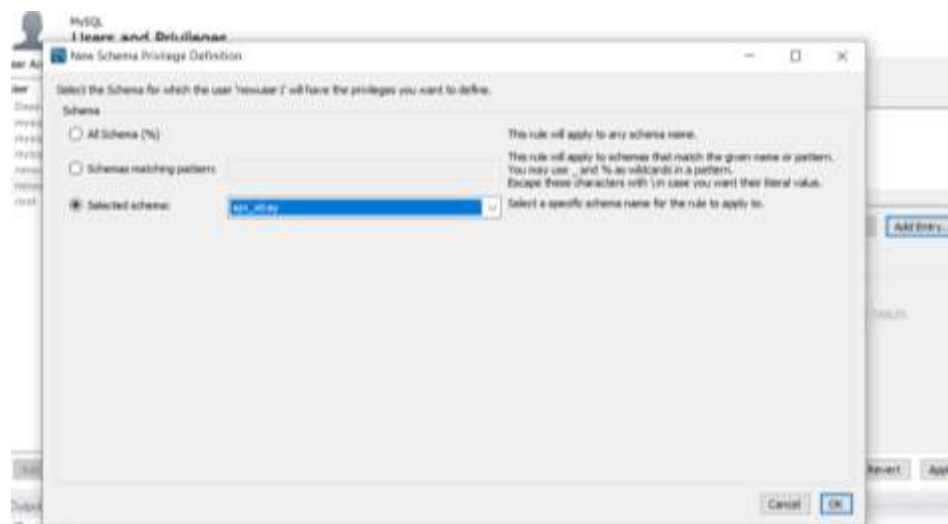


Рисунок 2.17 – доступные таблицы

3 Защита от несанкционированного доступа

Для защиты от несанкционированного доступа к данным, хранящимся на компьютере, наиболее часто используются пароли. Компьютер разрешает доступ к своим ресурсам только тем пользователям, которые зарегистрированы и ввели правильный пароль. Каждому конкретному пользователю может быть разрешен доступ только к определенным информационным ресурсам. При этом может производиться регистрация всех попыток несанкционированного доступа. Это наиболее простой способ защиты информации.

Элементом защиты в данном курсовом проекте является ip адрес устройств, так как моя база данных хранится в RDS Amazon (рисунок 3.1).

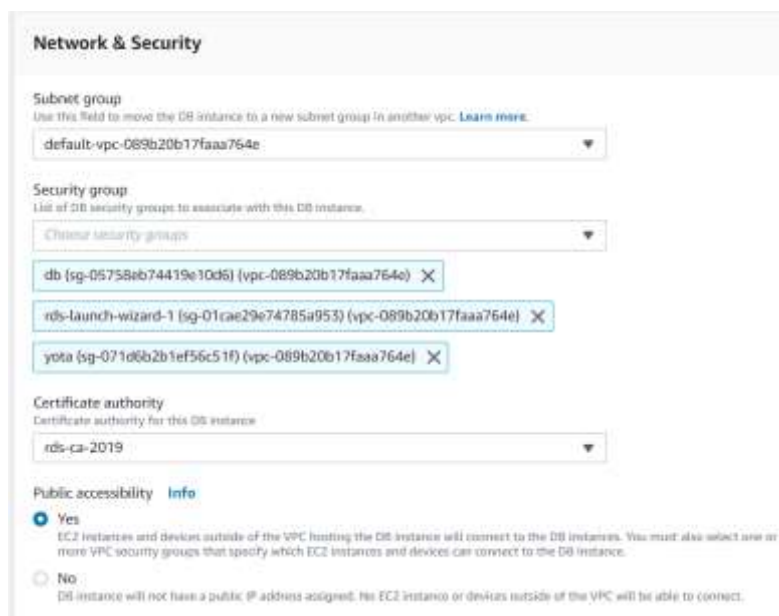


Рисунок 3.1 – Защитная система бд

Похожий алгоритм используется в большинстве проектах, содержащих различные роли доступа. В таких приложениях наименьший функционал предоставляется учетной записи, которую могут использовать все пользователи приложения. Пароль для учетных записей такого типа делают либо общедоступным, либо пустым.

3.1. Настройка защиты RDS Amazon



Рисунок 3.2 – Добавление security group

Нажимаем кнопку “Add Rule”, выходит вторая окно.



Рисунок 3.3– Add Rule

Тут определяем какому порту и какой тип подключение, и по какому ip адресу будет подключаться пользователь. После этого переходим к нашей базе и указываем ему этого пользователя.

4 Клиентское приложение

Для создания клиентского интерфейса используется следующее ПО: Microsoft Visual Studio 2019 и MySQL Workbench.

На рисунке 4.1 изображено окно работы пользователя «Пользователя». В выпадающем списке показывается категория товаров. В окне 2 отображаются товары под выбранную категорию. После нажатие на товар показывается вся информация про товар.

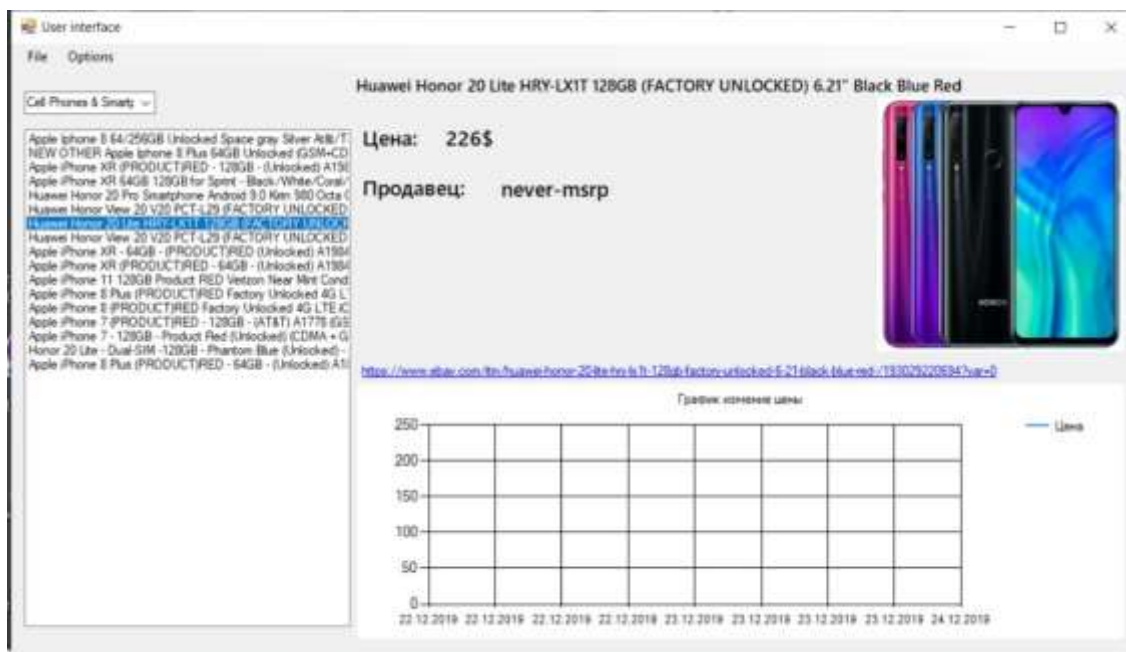


Рисунок 4.1 – Пользовательский интерфейс

На рисунке 4.2 изображено окно работы пользователя «Администратор». Форма 3 содержит кнопки для взаимодействия с базой данных. Кнопка “Добавить товар” добавляет товар на базу данных. Кнопка “Обновить цены” обновляет цены товаров принудительно. Это для администратора. Цены товаров обновляется на сервере. Кнопка “Удалить” Удаляет товары, которые сняты с продажи. А на листе показывается товары которые были сняты с продажи.



Рисунок 4.2 –Интерфейс администратора

В приложении Б приведен код программы для пользователя.

4.1 Тестирование пользовательского интерфейса



Рисунок 4.3 –Выбор категории

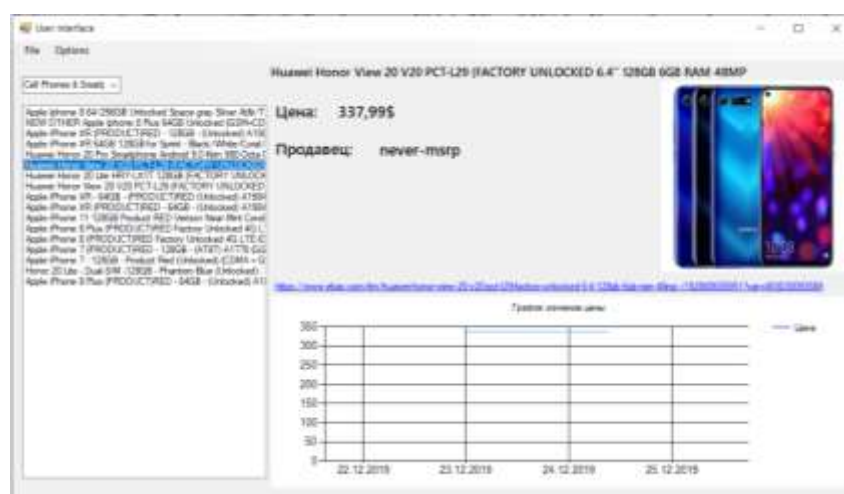


Рисунок 4.4 –Выбор категории

4.2 Тестирование интерфейса администратора

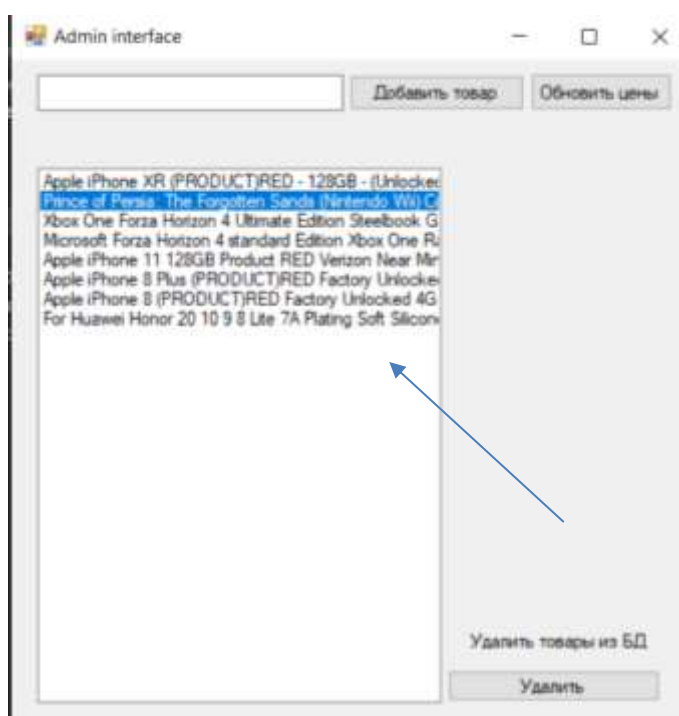


Рисунок 4.5 – Поле показывает товары

Рисунке 4.5 показывает товары которые были сняты с продажи.
При нажатии кнопки “Удалить” будет удалены товары с базы данных.

Заключение

В результате выполнения курсового проекта была реализована БД «Разработка базы данных информационной системы дата майнинга для интернет площадки EBAY», которая содержит таблицы и запросы. Запросы используются для быстрого поиска информации в базе данных и получения ответов на разнообразные вопросы. Для защиты от несанкционированного доступа была организована защита в MySQL Workbench, запрашивающая пароль пользователя для допуска к БД.

А так же, в результате выполнения курсового проекта, были изучены основные понятия, определения, свойства баз данных и систем управления ими, получены практические знания, умения и навыки использования основ теории баз данных в одной из современных сред программирования. (MySQL Workbench).

Были решены задачи, поставленные в начале работы. Была разработана структура базы данных; разработан интерфейс программы; в программу были включены функции поиска, выполнения различных запросов и отчетов. При этом были учтены все требования, выдвинутые в начале выполнения данного проекта.

Созданная информационная система отвечает всем требованиям, заявленным в начале работы, и содержит удобный интерфейс, обеспечивающий работу с ней.

Список использованных источников

1. Фленов М.Е. Linux глазами хакера [Текст] / М.Е. Фленов. - СПб.: БХВПетербург, 2019. — 416 с
2. Иванова, Г.С. Технология программирования [Текст] / Г.С. Иванова. – М.: “КНОРУС”, 2016. – 334 с.
3. Нархид, Н. и др. Apache Kafka потоковая обработка и анализ данных [Текст] / Н. Нархид, Г.Шапира – СПб.: Питер , 2019. — 320 с
4. Бабенко, Л. К. и др. Параллельные алгоритмы для решения задач защиты информации [Текст] / Л. К. Бабенко. – М.: “И.Д. Горячая линия-Телеком”, 2014. -304с.
5. Тарасов, С. СУБД для программиста. Базы данных изнутри [Текст] / С.Тарасов. – М.: “И.Д. СОЛОН-Пресс”, 2015. – 320 с.
6. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Гамма — СПб.: “ И.Д. Питер”, 2015. — 368 с.
7. Бабаш, А.В. Информационная безопасность. Лабораторный практикум: учебное пособие [Текст] / А.В. Бабаш. – М.: “И.Д. КНОРУС”, 2012 – 132 с.
8. Скиена, С. Алгоритмы. Руководство по разработке [Текст] / С. Скиена, — СПб.: “ И.Д. БХВ-Петербург”, 2011. — 720 с.
9. Фленов М.Е. Web-сервер глазами хакера [Текст]/ М.Е.Фленов - СПб.: БХВПетербург, 2009. — 336 с
10. Python 3 для начинающих [Электронный ресурс]. – Режим доступа: <https://pythonworld.ru/> (дата обращения 18.12.2019г.).
11. Полное руководство по языку программирования C# 8.0 и платформе .NET Core 3 [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/tutorial/> (дата обращения 22.12.2019г.).

Приложения

Приложения А

-- MySQL Workbench Forward Engineering

```
SET                @OLD_UNIQUE_CHECKS=@ @UNIQUE_CHECKS,  
UNIQUE_CHECKS=0;
```

```
SET      @OLD_FOREIGN_KEY_CHECKS=@ @FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0;
```

```
SET                @OLD_SQL_MODE=@ @SQL_MODE,  
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZE  
RO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_E  
NGINE_SUBSTITUTION';
```

-- -----

-- Schema api_ebay

-- -----

-- -----

-- Schema api_ebay

-- -----

```
CREATE SCHEMA IF NOT EXISTS `api_ebay` DEFAULT CHARACTER SET  
utf8 ;
```

```
USE `api_ebay` ;
```

-- -----

-- Table `api_ebay`.`category_product`

-- -----

```
CREATE TABLE IF NOT EXISTS `api_ebay`.`category_product` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `name_category` TEXT(250) NOT NULL,  
  PRIMARY KEY (`id`))
```

```

ENGINE = InnoDB;

-----

-- Table `api_eyay`.`product`
-----

CREATE TABLE IF NOT EXISTS `api_eyay`.`product` (
  `id` VARCHAR(250) NOT NULL,
  `title` MEDIUMTEXT NOT NULL,
  `price` FLOAT NOT NULL,
  `url` MEDIUMTEXT NOT NULL,
  `category_product_id` INT NOT NULL,
  PRIMARY KEY (`id`, `category_product_id`),
  INDEX `fk_product_category_product1_idx` (`category_product_id` ASC)
  VISIBLE,
  CONSTRAINT `fk_product_category_product1`
    FOREIGN KEY (`category_product_id`)
    REFERENCES `api_eyay`.`category_product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `api_eyay`.`seller_product`
-----

CREATE TABLE IF NOT EXISTS `api_eyay`.`seller_product` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name_seller` VARCHAR(45) NOT NULL,
  `product_id` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`id`, `product_id`),
  INDEX `fk_seller_product_product_idx` (`product_id` ASC) VISIBLE,
  CONSTRAINT `fk_seller_product_product`
    FOREIGN KEY (`product_id`)

```

```

REFERENCES `api_ebay`.`product` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `api_ebay`.`image_product`
-----

CREATE TABLE IF NOT EXISTS `api_ebay`.`image_product` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `adress_save` BLOB NOT NULL,
  `product_id` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`id`, `product_id`),
  INDEX `fk_image_product_product1_idx` (`product_id` ASC) VISIBLE,
  CONSTRAINT `fk_image_product_product1`
    FOREIGN KEY (`product_id`)
      REFERENCES `api_ebay`.`product` (`id`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `api_ebay`.`price_product`
-----

CREATE TABLE IF NOT EXISTS `api_ebay`.`price_product` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `price` VARCHAR(45) NOT NULL,
  `date` DATETIME NOT NULL,
  `product_id` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`id`, `product_id`),
  INDEX `fk_price_product_product1_idx` (`product_id` ASC) VISIBLE,
  CONSTRAINT `fk_price_product_product1`

```

```
FOREIGN KEY (`product_id`)
REFERENCES `api_ebay`.`product` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Приложения Б

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using IronPython.Hosting;
using MySql.Data.MySqlClient;

using System.Windows.Forms.DataVisualization.Charting;
using System.IO;

namespace client_EBAY
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            listBox1.Items.Clear();

            this.chart1.Series.Clear();

            this.chart1.Titles.Add("График изменение цены");

            DB db = new DB();

            DataTable table = new DataTable();

            MySqlDataAdapter adapter = new MySqlDataAdapter();

            MySqlCommand command = new MySqlCommand("SELECT `name_category` FROM
`category_product`", db.GetConnection());
            adapter.SelectCommand = command;
            adapter.Fill(table);
            var myData = table.Select();
            for (int i = 0; i < myData.Length; i++)
            {

                comboBox1.Items.Add(myData[i].ItemArray[0]);

            }
        }

        DataTable table1 = new DataTable();

        public void grafik(string arg)
        {
            DB db = new DB();
            MySqlDataAdapter adapter2 = new MySqlDataAdapter();
            MySqlCommand command2 = new MySqlCommand("SELECT
`price_product`.`price`,`price_product`.`date`FROM `api_ebay`.`price_product` where
`price_product`.`product_id` = @ucat_name ", db.GetConnection());
            command2.Parameters.Add("@ucat_name", MySqlDbType.VarChar).Value = arg;
```

```

        adapter2.SelectCommand = command2;
        DataTable table3 = new DataTable();
        _ = adapter2.Fill(table3);

        var myData3 = table3.Select();

        chart1.Series.Clear();

        if (chart1.Series.Count==0)
        {
            Series series = chart1.Series.Add("Цена");
            series.ChartType = SeriesChartType.Spline;
            int k1 = myData3.Length;

            for (int i = 0; i < myData3.Length; i++)
            {
                series.Points.AddXY(myData3[i].ItemArray[1],
myData3[i].ItemArray[0]);

            }
        }

        }

private void listBox1_SelectedIndexChanged_1(object sender, EventArgs e)
{
    string str = listBox1.SelectedItem.ToString();
    int l = listBox1.SelectedIndex;
    var myData1 = table1.Select();

    label5.Text = myData1[l].ItemArray[1].ToString();

    if (Convert.ToDouble(myData1[l].ItemArray[2].ToString())==0)
    {
        label3.Text = " Товар снять с продажи ";
    }
    else
    {
        label3.Text = myData1[l].ItemArray[2].ToString() + "$";
    }

    linkLabel1.Text= myData1[l].ItemArray[3].ToString();
    string arg = myData1[l].ItemArray[0].ToString();

    DB db = new DB();
    MySqlDataAdapter adapter = new MySqlDataAdapter();
    MySqlCommand command = new MySqlCommand("select
`seller_product`.`name_seller`,`image_product`.`adress_save` from `seller_product`,
`image_product` where `seller_product`.`product_id`=@ucat_name and
`image_product`.`product_id`=@ucat_name ", db.GetConnection());
    command.Parameters.Add("@ucat_name", MySqlDbType.VarChar).Value = arg;
    adapter.SelectCommand = command;
    DataTable table2 = new DataTable();
    _ = adapter.Fill(table2);

    var myData2 = table2.Select();

```

```

        label14.Text = myData2[0].ItemArray[0].ToString();

        byte[] img = (byte[])table2.Rows[0][1];

        MemoryStream ms = new MemoryStream(img);

        pictureBox1.Image=Image.FromStream(ms);
        grafik(myData1[1].ItemArray[0].ToString());

    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        listBox1.Items.Clear();
        string str1 = comboBox1.SelectedItem.ToString();
        int l = comboBox1.SelectedIndex + 1;
        DB db = new DB();
        MySqlDataAdapter adapter = new MySqlDataAdapter();
        table1.Rows.Clear();
        MySqlCommand command = new MySqlCommand("SELECT
`product`.`id`,`product`.`title`,`product`.`price`,`product`.`url`,`product`.`category_pr
oduct_id` FROM `api_eyay`.`product` where `product`.`category_product_id`= @ucat_name
", db.GetConnection());
        command.Parameters.Add("@ucat_name", MySqlDbType.VarChar).Value =
1.ToString();
        adapter.SelectCommand = command;
        _ = adapter.Fill(table1);
        var myData1 = table1.Select();

        for (int i = 0; i < myData1.Length; i++)
        {
            listBox1.Items.Add(myData1[i].ItemArray[1]);
        }
    }

    private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
    {
        System.Diagnostics.Process.Start(linkLabel1.Text);
    }

    private void pictureBox1_Click(object sender, EventArgs e)
    {
        System.Diagnostics.Process.Start(linkLabel1.Text);
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        DB db = new DB();
        db.closeConnection();
    }

    private void fileToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        DB db = new DB();
        db.closeConnection();
        Close();
    }
}
}

```