**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение Московской области «Люберецкий техникум имени Героя Советского Союза, летчика-космонавта Ю.А.Гагарина»**

**КУРСОВАЯ РАБОТА**

Назиров Юрик Казбекович

по МДК.11.01 **Разработка, администрирование и защита баз данных**

Курс 2 Группа № ИС-21

Тема: Спроектировать базу данных организации автоматизации процесса обработки результатов маркетинговых исследований продукции, производимой или реализуемой предприятием.

Выполнил студент \_\_\_\_\_\_\_\_\_\_\_\_\_ Назиров Юрик Казбекович (подпись) (ФИО полность)

Руководитель Тарджиманян Лия Николаевна

(подпись) (ФИО полностью)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дзержинский 2023

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ...………………………………………………………….…………….….3

ГЛАВА 1 ОСНОВЫ БАЗ ДАННЫХ..............................................................................4

1.1 Системы управления базами данных………………………………………4

1.2 Модели данных. ООП……………………………………………………….4

1.3 Проблемы проектирования баз данных…………………………………….5

1.4 Реляционная алгебра и реляционное исчисление………………………....7

1.5 SQL89 и SQL92………………………………………………………………8

1.6 Распределенные системы баз данных, хранение данных…………….….10

1.7 Администрирование базы данных………………………………………...11

ГЛАВА 2 БАЗА ДАННЫХ АВТОМАТИЗИРОВАННОЙ РЕГИСТРАЦИИ ДОКУМЕНТОВ, НА ПРЕДПРИЯТИИ ПО ПРОИЗВОДСТВУ ОФИСНОЙ ТЕХНИКИ ......................,..............................................................................................14

2.1 Техническое задание……………………………………………………….14

2.2 Построение инфологической (концептуальной) модели

предметной области...……………………………………………………..…..15

2.3 Проектирование логической структуры базы данных.…………………..15

2.4 Организация ввода и корректировки данных в БД………………………18

2.5 Описание информационных потребностей пользователей

и их реализации………………………………………………………………..19

2.6 Разработка интерфейса……………………………………………………20

2.7 Разбор кода………………………………………………………………….22

ЗАКЛЮЧЕНИЕ.…………………………………………………………………….…28 СПИСОК ИСТОЧНИКОВ……………………………………….……………...……29

**ВВЕДЕНИЕ**

**Цель курсового проектирования:**

Спроектировать базу данных (БД) организации автоматизации процесса обработки результатов маркетинговых исследований продукции, производимой или реализуемой предприятием

Для для достижения поставленной цели необходимо решить следующие **задачи**:

* создать ER-диаграмму по техническому заданию (в Draw SQL );
* создать базу данных по ER-диаграмме (в Sqlite);
* заполнить базу данных тестовыми данными;
* создать авторизацию (в Qt5 дизайнер);
* разработать формы для автоматизированного заполнения данными БД (в Qt5 дизайнер);
* подключить формы к базе данных (в PyCharm)
* тестирование созданных форм на работоспособность.

Для создания базы данных в данной курсовой работе была выбрана СУБД SQLite основываясь на ее экономичности, надежности и портативности. Разработка базы данных будет включать в себя анализ требований, проектирование структуры, создание схемы и написание SQL запросов. Конечный результат будет представлять готовую базу данных, соответствующую требованиям предприятия, чтобы эффективно обрабатывать результаты маркетинговых исследований продукции.

**ГЛАВА 1. ОСНОВЫ БАЗ ДАННЫХ**

**1.1 Системы управления базами данных**

Традиционные файловые системы, которые используются для хранения небольших объемов данных, более просты в использовании, но при увеличении объемов данных они становятся неэффективными и не могут обеспечить достаточный уровень безопасности и устойчивости к сбоям. Именно поэтому Системы управления базами данных (СУБД) становятся все более популярными среди компаний и организаций. СУБД, в отличие от традиционных файловых систем, позволяют более эффективно управлять большими объемами данных и сложными приложениями. Но, стоит отметить, что работа с БД может быть более сложной и затратной. На сегодняшний день существует множество СУБД, каждая из которых имеет свои преимущества и недостатки.

Архитектура СУБД представляет собой организацию, хранение и обработку данных в БД через три уровня: внешние схемы данных, концептуальные схемы и физические схемы данных. Эти три уровня предназначены для определения логической модели данных, структуры хранения данных и управлению транзакциями. Это позволяет не только управлять данными, но и обеспечить безопасность и оптимизацию запросов к БД.

Цель архитектуры СУБД - обеспечить эффективное управление данными, а также обеспечить конфиденциальность, целостность и доступность данных. Составление схемы данных и написание запросов на языке SQL требуют определенных знаний и навыков. Однако, современные СУБД обычно имеют ряд удобных инструментов для работы с данными.

Вместе с развитием технологий баз данных появились новые возможности и перспективы для их развития. Например, использование облачных СУБД для масштабирования и уменьшения нагрузки на локальные серверы, использование искусственного интеллекта и машинного обучения для улучшения производительности и управления данными. Все это показывает, что СУБД остаются важным элементом для эффективного управления данными в современном мире.

**1.2 Модели данных. ООП**

Существуют два разных подхода к хранению и управлению данными: реляционная модель данных и объектно-ориентированная модель данных. В реляционной модели данные представляются в виде таблиц, а единицы связи между данными указываются отношениями между таблицами. В то же время, объектно-ориентированная модель данных работает с объектами, а связь между данными устанавливается на уровне ссылок между объектами. Объектно-ориентированный подход использует концепции объектно-ориентированного программирования (ООП) для разработки баз данных, где приложения могут быть более гибкими и расширяемыми, что делает его популярным для обработки больших объемов данных.

ООП также широко используется в разработке программного обеспечения, где объекты инкапсулируют данные и функции, а механизмы наследования и полиморфизма управляют данными и моделями. Инкапсуляция скрывает детали реализации от пользователей, что делает код более безопасным и легко поддерживаемым. ООП является популярным подходом для разработки ПО в различных областях, включая создание приложений для работы с данными.

Сетевая модель является еще одной моделью хранения данных, где данные представляются в виде графа. Она обладает рядом преимуществ, таких как гибкость, быстрый доступ и эффективность. Недостатки: сложность, неоднородность и ограниченность.

**1.3 Проблемы проектирования баз данных**

Существуют различные проблемы, возникающие при проектировании баз данных. Одной из них является проблема слабых объектов, которая возникает, когда объекты не могут существовать без родительских объектов и не имеют своего уникального идентификатора. Существуют два способа решения этой проблемы, в зависимости от конкретной ситуации. Также могут возникнуть методологические, технологические и другие проблемы при проектировании баз данных.

Функциональная зависимость - это связь между атрибутами в отношении, где значение одного определяет значение другого. Ее необходимо учитывать при проектировании баз данных, чтобы избежать дублирования данных и улучшить эффективность запросов. Для удаления избыточных функциональных зависимостей необходимо использовать нормальные формы.

Многозначные зависимости и нормальные формы высокого порядка - это концепции, которые также связаны с проектированием баз данных и их нормализацией. Они используются для устранения многозначных зависимостей, но могут привести к усложнению структуры базы данных.

Декомпозиция без потерь - это разбиение отношения в базе данных на более мелкие отношения, сохраняя все функциональные зависимости. Однако, это может увеличить количество таблиц и усложнить структуру базы данных, что может повлиять на производительность запросов. Необходимо выбирать подходящие функциональные зависимости и декомпозицию в зависимости от требований и задач базы данных.

**1.4 Реляционная алгебра и реляционное исчисление**

Реляционная алгебра и реляционное исчисление - это две основные модели языков запросов для работы с реляционными базами данных. Обе модели предназначены для извлечения данных из таблиц и выполнения сложных операций, включая объединение, пересечение, выборку, проекцию и соединение. Каждая модель имеет свои преимущества и может использоваться в различных сценариях.

Реляционная алгебра является набором математических операций, которые позволяют выбирать и обрабатывать данные в таблицах. Эти операции включают в себя выборку, проекцию, объединение и пересечение таблиц. Реляционная алгебра может использоваться для написания простых запросов и осуществления быстрого доступа к данным. Более сложные операции могут включать соединение таблиц и использование вложенных запросов.

Реляционное исчисление, в свою очередь, является более формальным языком, который используется для формулирования запросов на выборку данных. Он также включает в себя два варианта - исчисление кортежей и исчисление доменов. В исчислении кортежей запросы формулируются в терминах строк таблицы, а в исчислении доменов - в терминах значений столбцов таблицы. Реляционное исчисление может использоваться для написания более сложных запросов, в том числе запросов, которые не могут быть отображены с помощью реляционной алгебры.

Обе модели имеют свои преимущества и недостатки, и выбор между ними зависит от конкретных требований и задач. Некоторые операции, такие как выборка и проекция, легче сделать с помощью реляционной алгебры, тогда как более сложные операции, такие как соединение, могут использоваться с помощью реляционного исчисления. Кроме того, операции реляционной алгебры могут использоваться для реализации запросов в реляционном исчислении.

Как правило, реляционная алгебра используется для формулирования запросов в строгой математической терминологии, которая базируется на теории множеств, тогда как реляционное исчисление используется для оперирования более сложными запросами. Обе модели являются основой для многих СУБД и используются в различных приложениях от управления экономической и бизнес-информацией до научных исследований.

**1.5 SQL89 и SQL92**

Стандарты языка SQL для работы с реляционными базами данных включают в себя SQL89 и более расширенную SQL92. SQL92 ввел новые функции и операторы, такие как JOIN для объединения данных из нескольких таблиц и внешние ключи для связывания данных из разных таблиц, и позволяет использовать подзапросы и новые строковые функции, такие как SUBSTRING и TRIM. В отличие от SQL89, SQL92 ввел системные таблицы и новые типы данных, такие как BOOLEAN и INTERVAL. Однако SQL89 все еще может использоваться, так как он проще в использовании.

MS SQLite - это компактная встраиваемая библиотека, реализующая SQL-диалект, совместимый со стандартом SQL-92, и поддерживающая операторы SQL, функции и триггеры. Она не имеет отдельного сервера баз данных и может использоваться для работы с локальными базами данных в различных приложениях.

При выборе базы данных для проекта необходимо учитывать такие критерии, как производительность, масштабируемость, надежность и стоимость. Мэйнфреймы могут обеспечить высокую производительность, но могут быть дорогими в обслуживании и разработке, в то время как файловые серверы более доступны, но могут потребовать дополнительных затрат на оборудование и сетевые ресурсы. Базы данных клиент/сервер могут быть более доступными, но могут потребовать дополнительных затрат на сетевое оборудование и программное обеспечение.

Логика приложения определяет его функциональность и способность взаимодействовать с пользователем. Разделение приложения на клиентский, бизнес-логический и слой данных позволяет легко модифицировать и изменять каждый слой без изменения всего приложения. Принципы систем клиент/сервер, такие как разделение обязанностей, прозрачность распределения, масштабируемость, надежность и безопасность, должны учитываться при разработке приложения. Критерии производительности, надежности, масштабируемости, безопасности и удобства использования помогают оценить системы клиент/сервер на их эффективность и пригодность для конкретных условий.

**1.6 Распределенные системы баз данных, хранение данных**

Распределенные системы баз данных (РСБД) — это сложные компьютерные системы, которые управляют хранением и обработкой данных в распределенной среде. Их главная задача - обеспечение высокой производительности и надежности. Однако у централизованных РСБД есть проблемы с масштабируемостью и доступностью, что стало причиной создания более сложных и масштабируемых распределенных РСБД.

Для создания РСБД необходимо учитывать принципы, которые обеспечат высокую доступность, масштабируемость, отказоустойчивость и безопасность данных в распределенной среде. Это разделение данных, репликация данных, согласованность данных, выполнение транзакций в соответствии с принципами ACID, масштабируемость, безопасность данных, отказоустойчивость и управление конфликтами.

Также сложной задачей является управление распределенной информацией, которое может вызвать проблемы согласованности данных, безопасности, управления версиями, сложности интеграции и масштабируемости.

Хранилище данных — это система, которая организует и хранит большие объемы данных, доступные для аналитики и бизнес-пользователей. Хранилище данных должно соответствовать принципам целостности, единообразия, интеграции, гибкости, производительности, безопасности, управления метаданными и поддержки различных запросов.

Для обработки и хранения больших объемов данных может потребоваться использование распределенных хранилищ данных. Они объединяют несколько узлов на разных компьютерах и обеспечивают быстрый доступ к данным, высокую отказоустойчивость и балансировку нагрузки.

Проектирование РСБД и хранилищ данных является сложным процессом, который требует правильного понимания бизнес-требований и данных, правильной модели данных, производительности и безопасности. При проектировании этих систем необходимо учитывать все эти факторы, чтобы создать эффективную и надежную систему хранения и обработки данных.

**1.7 Администрирование базы данных**

Администрирование баз данных является жизненно важной функцией для обеспечения эффективности, безопасности и целостности информационных систем. Эта функция включает в себя установку, конфигурацию, мониторинг производительности, создание резервных копий, управление безопасностью и схемой. Необходимо соблюдать стандарты безопасности, производительности, резервного копирования и управления схемой, чтобы эффективно администрировать базы данных.

Для безопасности и сохранения целостности данных в базах данных (БД) необходимо предотвращать несанкционированный доступ, ошибки проектирования, SQL-инъекции и другие виды атак. Это достигается путем использования различных методов, таких как аутентификация, авторизация, регулярное резервное копирование, шифрование данных и программы для защиты от уязвимостей, а также установления обновлений и предохранительных мер на компонентах системы.

Сжатие данных - это эффективный способ улучшить производительность и экономить место на диске. Для достижения данной цели используются различные методы сжатия, включая алгоритмы со сжатием без потерь и с потерями. Но если работа выполняется с большими объемами информации, создание и сжатие таких массивов данных может создать сложности с их доступом, управлением и безопасностью. Поэтому для решения этих проблем необходимо использовать современные методы и средства.

Среда WEB идеально подходит для создания приложений баз данных, которые могут быть использованы через Интернет при помощи веб-браузера. Для разработки таких приложений программисты используют различные языки программирования, такие как PHP, JavaScript, Ruby, Python и другие, а также специализированные фреймворки и библиотеки, которые значительно упрощают процесс разработки.

Для создания структуры базы данных используются ER-диаграммы, которые помогают определить связи между таблицами и полями. Для проектирования, анализа, тестирования и документирования БД используются CASE-средства.

Тестирование новых приложений и изменений в существующих осуществляется с использованием специальных баз данных для тестирования. СУБД используются для управления и обработки данных внутри базы данных, что позволяет эффективно управлять и обрабатывать данные.

Оптимизация производительности баз данных - это важный аспект их работы. Для ее достижения используются инструменты мониторинга производительности, которые помогают оптимизировать работу БД.

Резервное копирование и восстановление данных - важные аспекты безопасности баз данных. Их осуществление осуществляется при помощи специальных инструментов, которые создают резервные копии и восстанавливают данные в случае необходимости.

Анализ данных - еще один важный аспект использования баз данных. Для этого используются инструменты анализа данных, которые позволяют извлекать информацию из БД и ее анализировать.

Безопасность баз данных - это комплекс мер, направленных на защиту данных от несанкционированного доступа, кражи, изменений и уничтожения. Их основные мероприятия включают аутентификацию и авторизацию пользователей, аудит данных, шифрование, правильную настройку системы и обучение пользователей, а также резервное копирование и восстановление данных.

Серверы баз данных также нуждаются в специальных мерах безопасности, таких как резервирование, кластеризация, мониторинг и защита от несанкционированного доступа или злоупотребления, регулярное обновление и обслуживание, а также физическая защита от пожаров, наводнений и краж.

**ГЛАВА 2. БАЗА ДАННЫХ ОРГАНИЗАЦИИ АВТОМАТИЦАЗИИ ПРОЦЕССА ОБРАБОТКИ РЕЗУЛЬТАТОВ МАРКЕТИНГОВЫХ ИССЛЕДОВАНИЙ ПРОДУКЦИИ, ПРОИЗВОДИМОЙ ИЛИ РЕАЛИЗУЕМОЙ ПРЕДПРИЯТИЕМ.**

**2.1 Техническое задание**

Общее описание проекта: на предприятии необходима система автоматизации процесса обработки результатов маркетинговых исследований продукций, производимой или реализуемой предприятием. В рамках проекта необходимо разработать базу данных для хранения информации о продукции, клиентах, заказах и т.д.

Требования к базе данных:

* База данных должна быть реляционной и хранить информацию о продукции, клиентах, заказах и т.д.
* База данных должна обеспечивать возможность добавления, удаления записей в таблицы.
* Должна быть предусмотрена возможность связывания записей в разных таблицах.
* Для каждой таблицы должна быть определена первичная ключевая колонка.
* База данных должна обеспечивать защиту данных от несанкционированного доступа.

Описание функциональности:

В интерфейсе должно быть реализовано добавление новых данных в базу данных. Также пользователь может удалять ненужные записи из базы данных. Должна быть реализована возможность просмотра полных таблиц. Интерфейс должен быть защищен от стороннего доступа к изменению целостности данных в БД.

Для проектирования базы данных автоматизированной регистрации документов и кадрового учета на предприятии можно использовать методологию Entity-Relationship (ER). Эта методология позволяет описать сущности, атрибуты и связи между ними.

**2.2 Построение инфологической (концептуальной) модели предметной области**

На основе технического задания в базе данных должны присутствовать следующие сущности: заказы, категории, клиенты, детали заказов, продукты и маркетинговые исследования. У которых должны присутствовать атрибуты и связи, точная схема базы данных находится ниже см. рис. 2.1

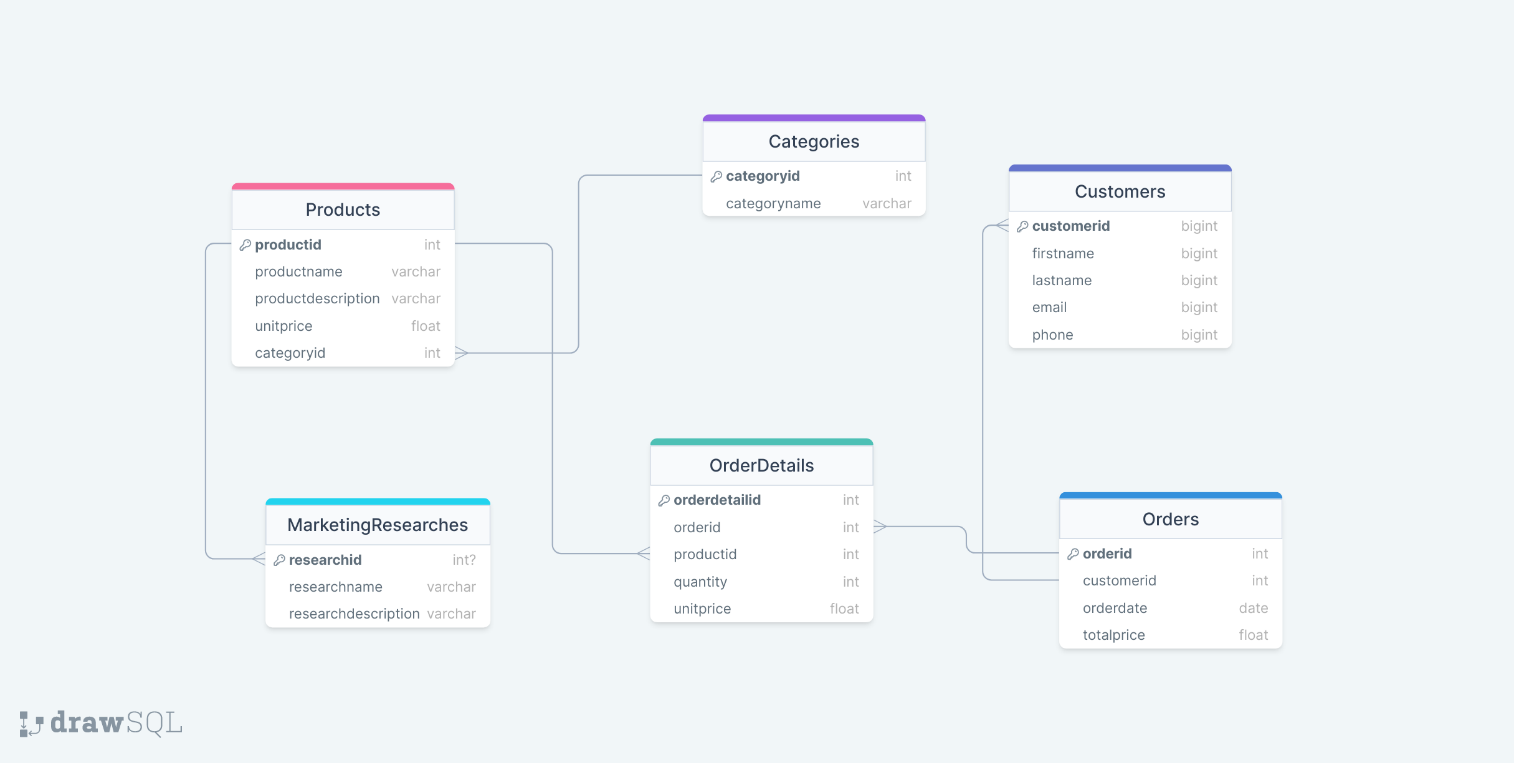


Рисунок 2.1 ER-диаграмма базы данных.

**2.3 Проектирование логической структуры базы данных**

На базе ER-диаграммы создаём базу данных в Sqlite. У нас есть шесть сущностей: заказы, категории, клиенты, детали заказов, продукты и маркетинговые исследования.

Таблица “Заказы ” имеет следующие атрибуты:

IdЗаказа

IdКлиента

Дата заказа

Итоговая цена

Таблица “Категории” имеет следующие атрибуты:

IdКатегории

Название категории

Таблица “Детали заказов” имеет следующие атрибуты:

IdДеталей заказа

IdЗаказа

IdПродукта

Количество

Цена за единицу

Таблица “Продукты” имеет следующие атрибуты:

IdПродукта

IdКатегории

Наименование товара

Описание товара

Цена за единицу

Таблица “Маркетинговые исследования” имеет следующие атрибуты:

IdИсследования

Название исследования

Описание исследования

Таблица “Клиенты” имеет следующие атрибуты:

IdКлиента

Имя

Фамилия

Email

Номер телефона

Для реализации контроля целостности были использованы встроенные средства СУБД SQLite, такие как ограничения NOT NULL, UNIQUE и FOREIGN KEY.

В качестве типов данных были выбраны наиболее подходящие для каждого поля: INTEGER для идентификаторов, TEXT для строковых значений, REAL для дат и времени.

**2.4 Организация ввода и корректировки данных в БД**

Первичный ввод записей в базу данных будет производится при помощи импортирования данных из Excel см. (рис. 2.2) . Далее ввод данных будет производится непосредственно с формы.

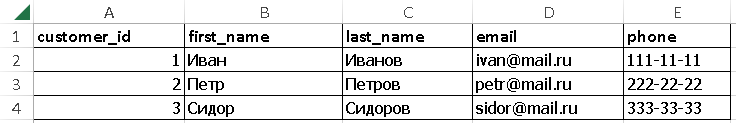


Рисунок 2.2 Ввод данных в Excel

Корректироваться данных производится с формы, или непосредственно в Sqlite см. (рис. 2.3)

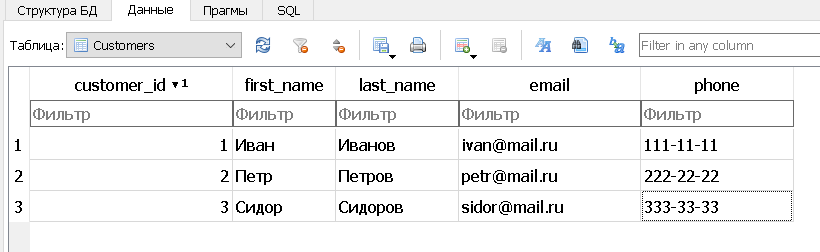


Рисунок 2.3 Корректировка данных в SQLite

**2.5 Описание информационных потребностей пользователей и их реализации**

У пользователя могут возникнуть следующие потребности при пользовании интерфейсом:

* авторизация на форме;
* добавление данных в базу данных;
* удаление данных из базы данных;

При запуске интерфейса все пользователи попадают на первую страницу

страницу авторизации, чтобы продолжить работу с формами все пользователи должны зарегистрироваться.

Нажав на кнопку зарегистрироваться, пользователь переходит на страницу регистрации. У пользователи просят ввести в ячейку email(email пользователя) в ячейку пароль (пароль чтобы в дальнейшем с помощью него заходить в базу данных), после ввода данных пользователь нажимает на кнопку зарегистрироваться вследствие чего его данные сохраняются в базе данных и пользователь возвращается в окно авторизации он вводит свой email и пароль и успешно переходит в на след форму для выбора таблиц для дальнейшей работы с ними.

После авторизации пользователь попадает на форму “Выбор таблицы”, с которой пользователь может выбрать определенную таблицу для дальнейшей работы с ним.

Работа с формами и редактирование данных в БД.

Выбрав таблицу для дальнейшей работы с ней у пользователя открывается форма таблицы например таблицы “Клиенты”, чтобы добавить новую запись в базу данных необходимо заполнить все необходимые ячейки, после чего нажать на кнопку добавить, данные будут успешно добавлены в таблицу.

Чтобы удалить ненужную информацию из БД необходимо в окне, в котором отображается таблица, выбрать запись и нажать на нее, после чего кликнуть на кнопку “Удалить”.

Все остальные формы работают аналогично форме - “Клиенты”.

**2.6 Разработка интерфейса**

Для начала в QT 5 дизайнере необходимо создать внешний вид форм, перенести на окно необходимые кнопки, поля для ввода данных, текстовые значения см. (рис. 2.4).

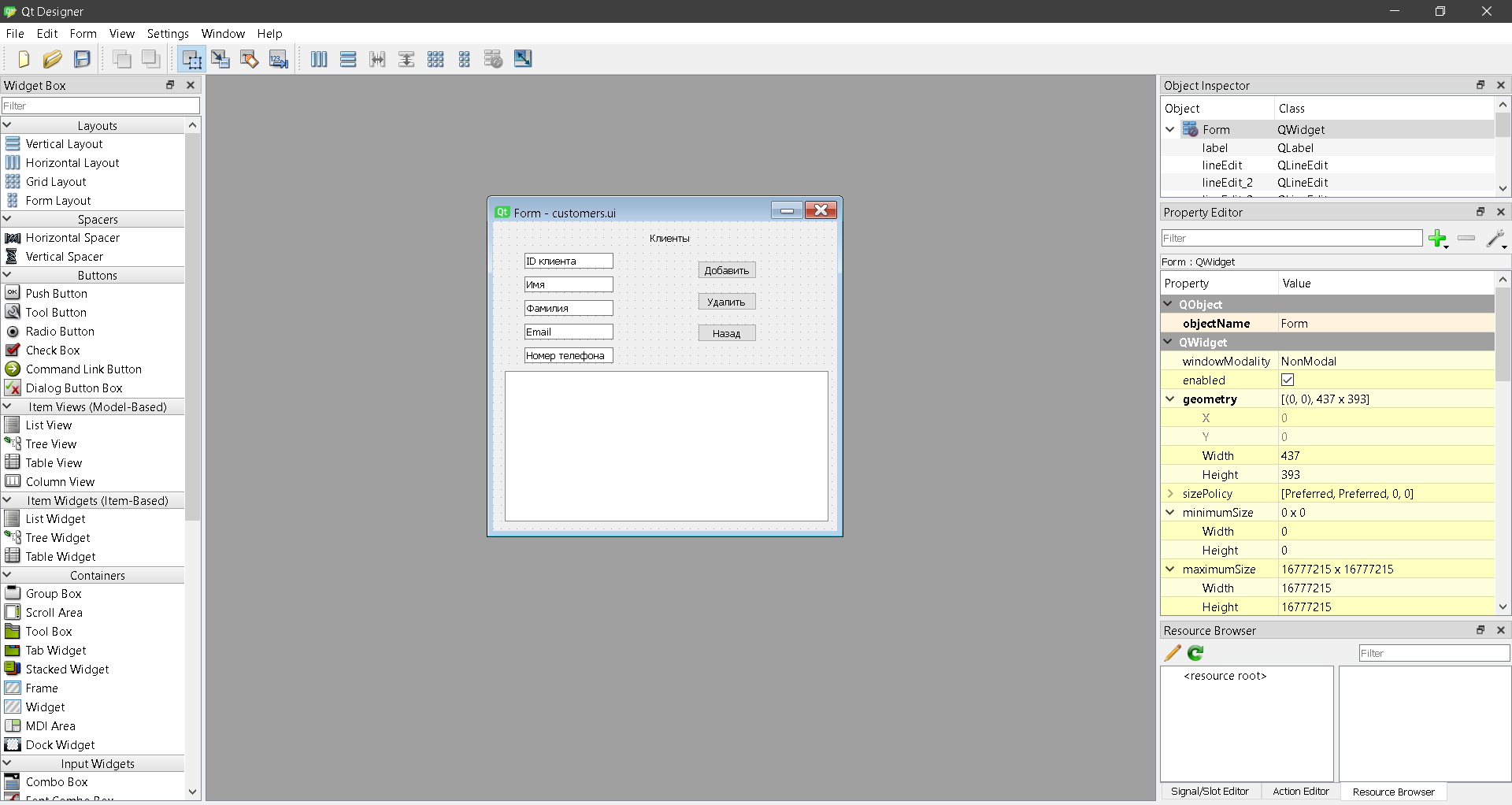


Рисунок 2.4 Создание формы в QT дизайнере

После создания внешнего вида формы нам необходимо добавить функции на кнопки и поля ввода, чтобы наш интерфейс исправно функционировал (рис. 2.5).

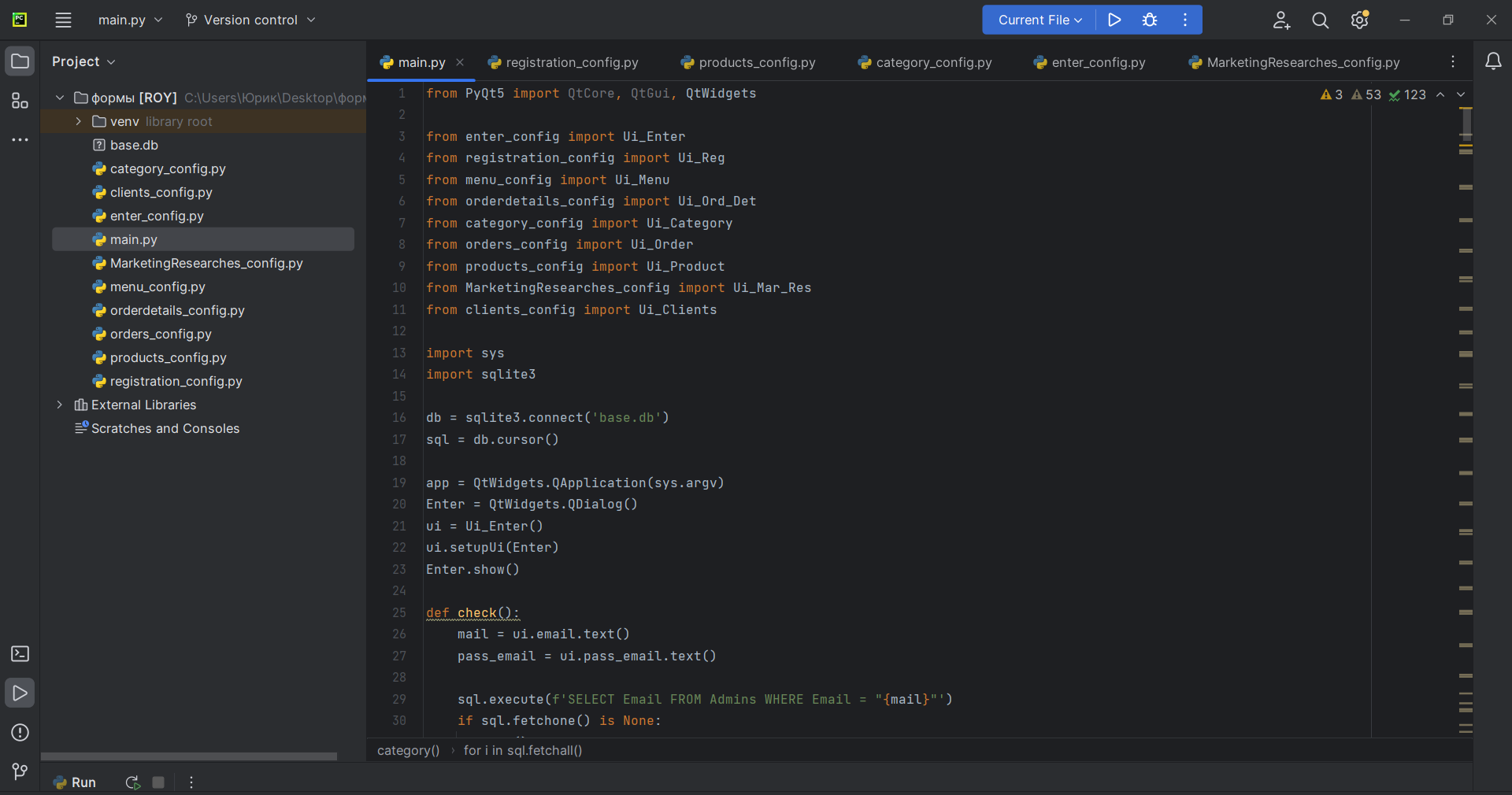


Рисунок 2.5 Написание кода в PyCharm

**2.7 Разбор кода**

Создание класса и вызов необходимых функций (рис. 2.6).

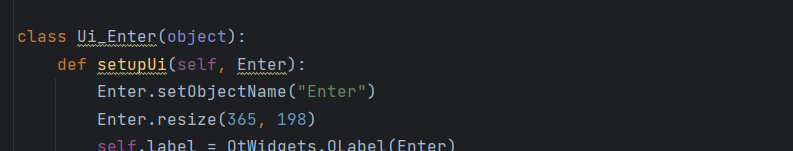


Рисунок 2.6 Создание класса.

Авторизация пользователя, проверка введенных данных (рис. 2.7).

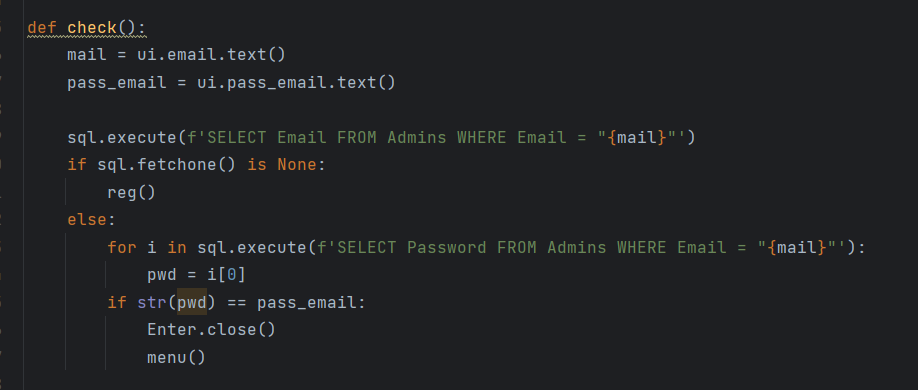


Рисунок 2.7 Авторизация

Открытие Данных на форме, SQL запрос на отображение таблицы на форме, подключение к базе данных (рис. 2.8).

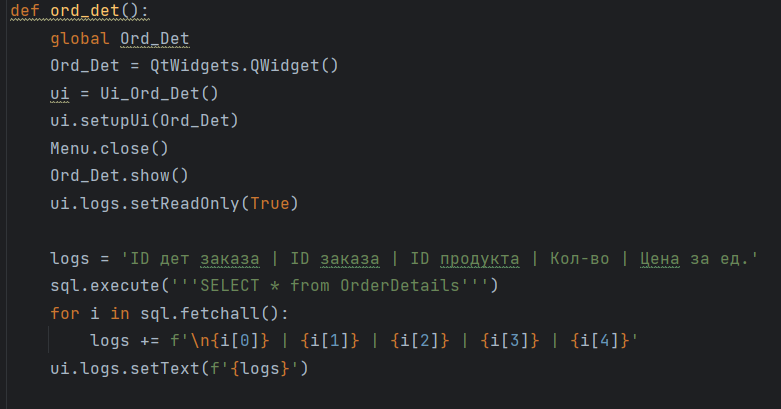


Рисунок 2.8 Запуск формы, вывод данных

Удаление данных из таблицы (рис. 2.9).



Рисунок 2.9. Удаление данных

Добавление данных в таблицы (рис. 2.10).



Рисунок 2.10. Добавление данных

Функция формы для выбора таблицы (рис. 2.11).



Рисунок 2.11 Выбор таблицы

Функция для возвращения в меню (рис. 2.12).

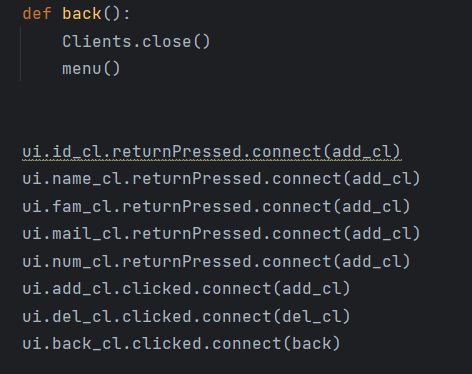


Рисунок 2.12 Функции кнопки “назад”

Функция для завершения работы (рис. 2.12).

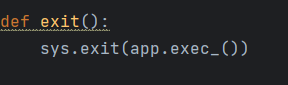


Рисунок 2.12 Функции кнопки “выход”

Все формы работают на основе кода, который показан выше, изменения происходят только в SQL запросах.

**ЗАКЛЮЧЕНИЕ**

**Целью** курсового проектированияявляласьcпроектировать базу данных организации автоматизации процесса обработки результатов маркетинговых исследований продукции, производимой или реализуемой предприятием. .

Для для достижения поставленной цели были решены следующие **задачи**:

* была создана ER-диаграмма по техническому заданию (в Draw SQL );
* была создана база данных по ER-диаграмме (в Sqlite);
* БД была заполнена тестовыми данными;
* была создана авторизация (в Qt5 дизайнер);
* разработаны формы для автоматизированного заполнения данными БД (в Qt5 дизайнер);
* подключены формы к базе данных (в PyCharm)
* были протестированы на работоспособность созданные формы.

**СПИСОК ИСТОЧНИКОВ**

1. "Базы данных: проектирование, реализация, использование" К. Джеймс, Г. Хеннеси;

[Т.Коннолли, К.Бегг - Базы данных. Проектирование, реализация и сопровождение. Теория и практика. (3-е издание) (vk.com)](https://vk.com/doc-76428449_418087682?hash=7nEOjM82RzlG7uoRjnXmMQFgsyI4A9fqe0piaZDtq58&dl=a55Zg9Df9Lu35lQMR498O7kuntEbQCInAhVB6itpErX)

- 14 апреля 2023

1. "Системы управления базами данных" К. Дейт;

[Введение в системы баз данных - Дейт К.Дж. (djvu.online)](https://djvu.online/file/xvtymREmD4CKu?ysclid=lhbl0vsqb2884376138)

- 20 апреля 2023

1. "Базы данных. Концептуальное проектирование и моделирование" А. Николаев, А. Николаева;

<https://edu.tltsu.ru/sites/sites_content/site216/html/media67140/lec3-4_is1.pdf>

- 29 апреля 2023

1. "Реляционные базы данных: теория и практика" М. Элмасри, Ш. Наватхе.

<https://books.4nmv.ru/books/vvedenie_v_relyatsionnye_bazy_dannykh_3642766.pdf>

- 4 мая 2023