1. Анализ технического задания.

1.1. Описание предметной области.

Зачастую частные организации или юридические лица с целью безопасности обращаются в охранные организации для того, что бы обезопасить себя от угроз и рисков, связанных с преступной деятельностью, вандализмом, кражами и другими правонарушениями. Охранные организации предоставляют широкий спектр услуг, включая физическую охрану объектов, установку систем видеонаблюдения, сигнализации и контроля доступа.

В данной курсовой работе поставлена задача создания приложения для работы с базой данных Охранного агенства. Для разработки такой системы необходимы СУБД для создания базы данных и среда разработки для создания интерфейса и функций по работе с БД.

Согласно заданию, в программе нужно учесть следующие особенности:

1. Мы можем вести полный учёт о клиентах, добавлять, удалять, и редактировать по необходимости.

2. Добавлять новые сервисы по мере их появления в организации.

3. Вести учёт сотрудников, добавлять, удалять, и редактировать данные о них.

4. Вести учёт данных о заказах, а так же подкреплять необходимые документы при внесении любых изменений в заказ.

1.2. Анализ технического задания

Исходя из данных требований к приложению, база данных должна содержать несколько таблиц и данные таблицы должны быть связаны для целостности системы. Для разработки базы данных в рамках поставленной задачи можно использовать SQLite, которая как раз относится к СУБД реляционного типа.

Достоинством SQLite является то, что она имеет очень простой графический интерфейс и подходит для начального уровня знакомства с базами данных.

Для разработки программы, работающей с базой данных, нужно использовать объектно – ориентированный подход. Разработка приложения будет вестись на языке C#. В основе системы быстрой разработки (Windows Forms(.NET Framework)) лежит технология визуального проектирования и событийного программирования, суть которой заключается в том, что среда разработки берёт на себя большую часть рутиной работы, оставляя программисту работу по конструированию диалоговых окон и созданию процедур обработки событий. Такая среда разработки позволяет создавать самые различные программы: от простейших однооконных приложений до программ работы с распределёнными базами.

Для того чтобы создать программу, необходимо учесть то, что она создаётся, прежде всего, для пользователя, и поэтому немаловажным требованием к программе должен стать удобный и интуитивно понятный интерфейс. Необходимо предусмотреть все возможности управления приложением, чтобы упростить работу пользователя и максимально обеспечить эффективность работы.

Программа должна правильно работать с данными, т.е всегда должен выводится нужный результат, требуемый пользователю. Приложение должно мгновенно реагировать на действия пользователя и в зависимости от запроса с его стороны формировать выходной результат.

2 Разработка моделей данных

Этот этап играет ключевую роль в разработке приложения для работы с базой данных. На этом этапе выделяются сущности, их атрибуты и взаимосвязи между ними. Создание и настройка базы данных в выбранной системе управления осуществляется на основе логической модели, которая затем преобразуется в физическую. Этот процесс включает в себя определение структуры таблиц, индексов, ключей и других объектов базы данных.

2.1 Концептуальная модель**.**

Создание концептуальной модели начинается с анализа предметной области и выделения сущностей.

Концептуальная схема или концептуальная модель данных представляет собой карту понятий и их взаимосвязей, используемых в базах данных. Она описывает семантику организации и включает в себя набор утверждений о ее сущности. В частности, модель определяет ключевые элементы, представляющие интерес для организации (классы сущностей), информацию о которых она собирает, а также их характеристики (атрибуты) и связи между парами этих значимых элементов (отношения). (Рис. 1).

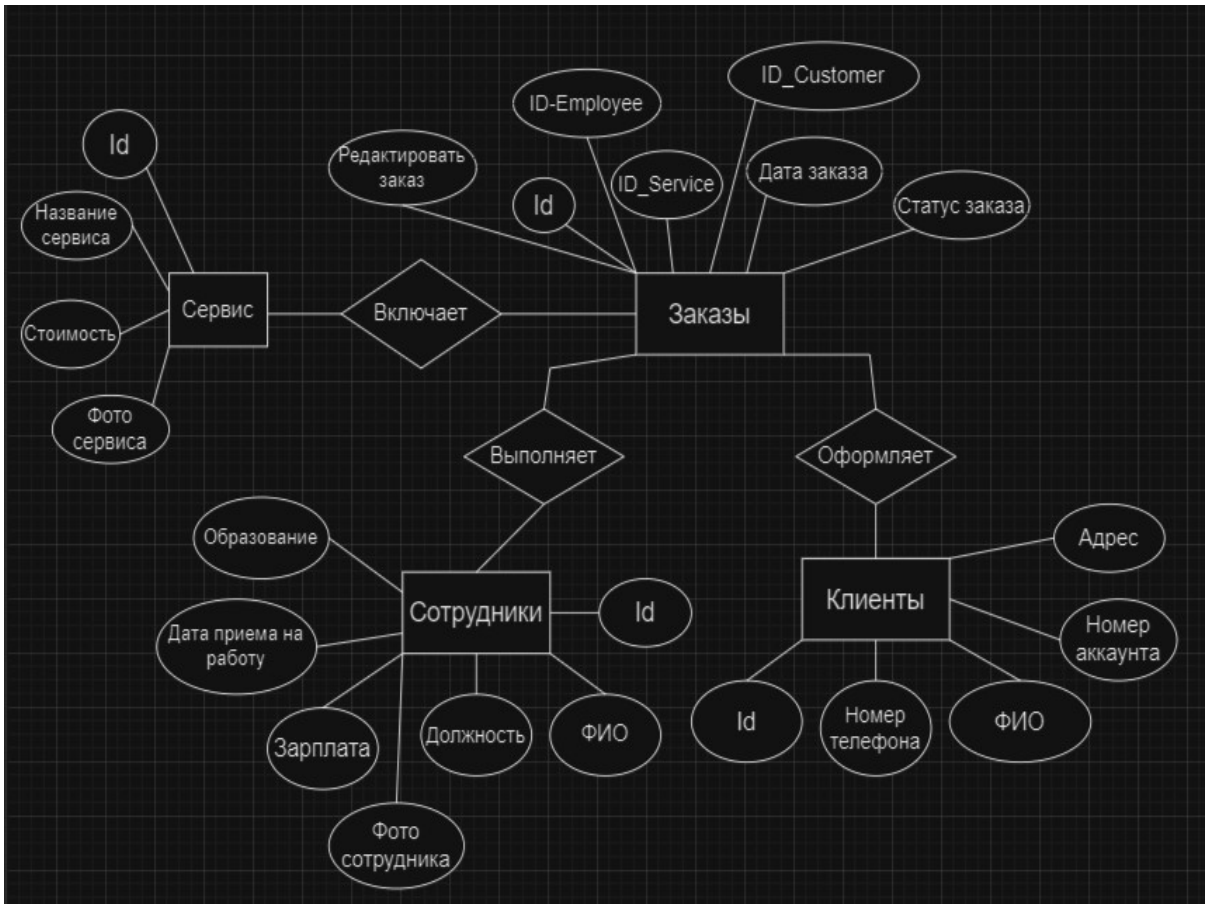

Рис. 1 - Концептуальная модель данных.

2.2. Разработка логической модели

Разработка логической модели базы данных - это этап проектирования, на котором определяется структура и организация данных без учёта конкретной реализации в виде таблиц и столбцов.

Логическая модель базы данных описывает структуру данных, их атрибуты и взаимосвязи, но не затрагивает вопросы хранения информации. На этом этапе проектирования определяются сущности (таблицы), их атрибуты (столбцы) и связи между ними. Логическая модель позволяет понять, какие данные будут храниться, как они связаны и как будут использоваться. Этот этап предшествует разработке физической модели базы данных, которая представляет собой реальную реализацию в виде таблиц, ключей и связей.

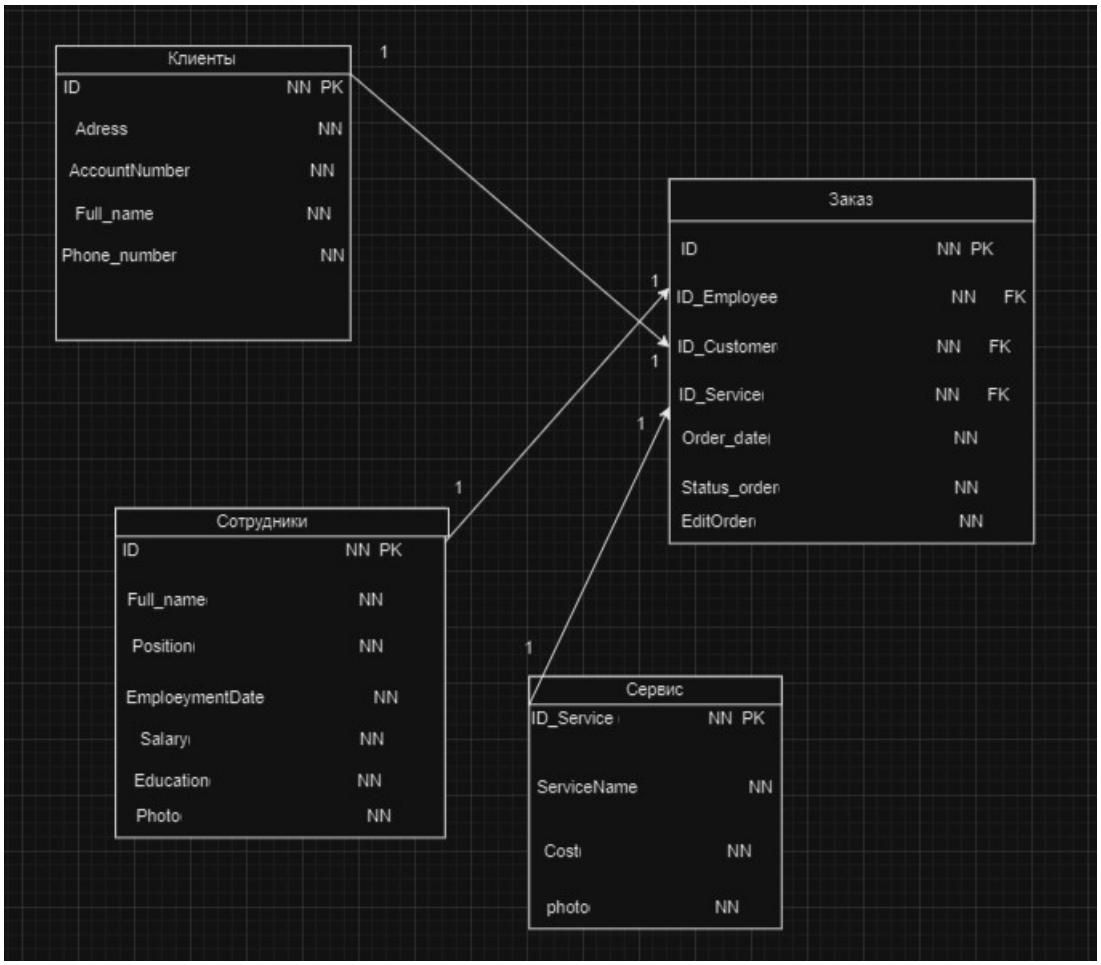

Рис. 2 - Логическая модель данных.

2.3. Разработка физической модели

Процесс разработки физической модели базы данных включает в себя несколько последовательных шагов, направленных на создание структуры для хранения и организации данных. Основные этапы этого процесса следующие:

1. Анализ требований: На этом этапе осуществляется сбор и анализ требований к базе данных, что включает определение сущностей и их атрибутов,

а также выявление взаимосвязей между данными.

2. Физическая реализация: Здесь создается физическая структура базы данных, включая выбор типов данных, определение размеров полей и индексов, а также оптимизацию производительности.

3. Тестирование и оптимизация: После реализации физической модели базы данных проводится тестирование её функционирования и оптимизация для обеспечения эффективной работы.

Эти основные этапы помогают разработать физическую модель базы данных, которая будет эффективно хранить и обрабатывать данные.

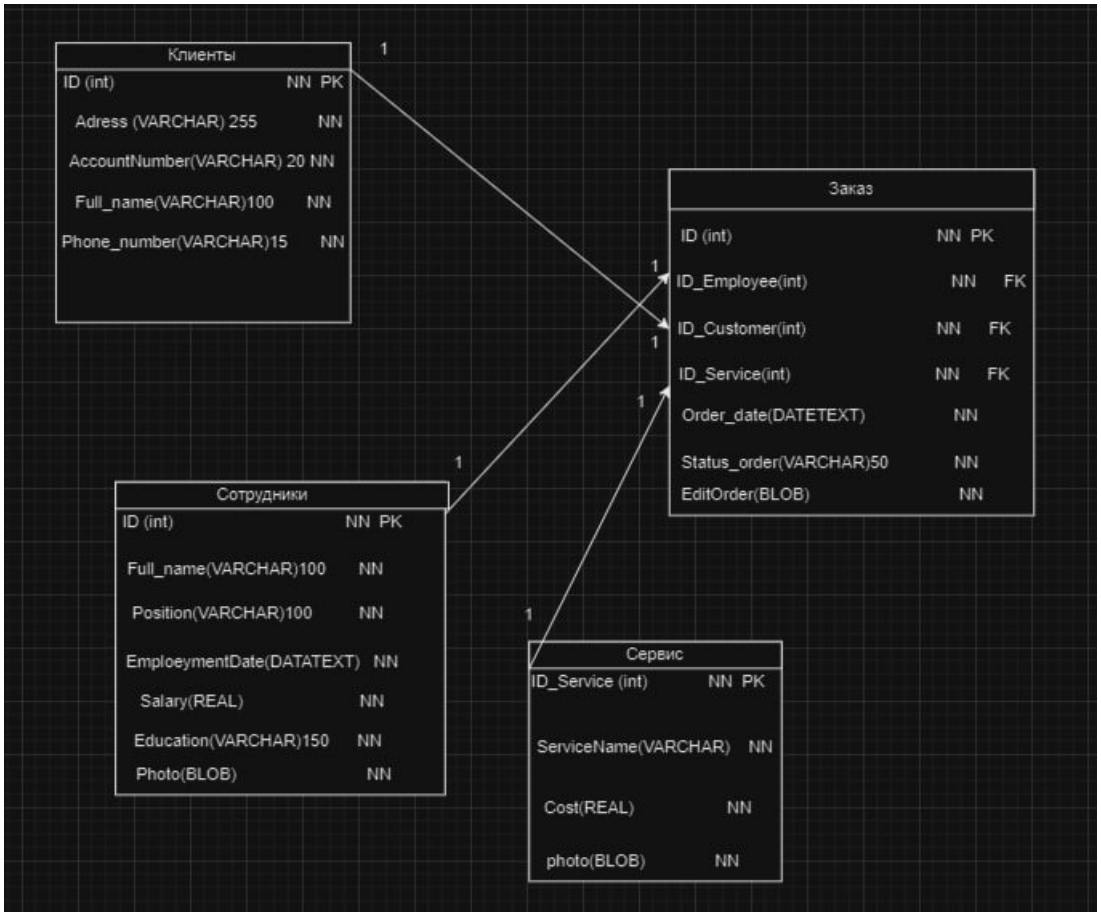

Рис.3 - Физическая модель.

В результате разработки этих моделей достигается понимание оптимизации структуры данных и обеспечение эффективного хранения и обработки информации. Каждая модель играет свою роль в процессе создания базы данных, начиная с общего представления данных и заканчивая их конкретной реализацией.

3. Разработка и реализация АИС

3.1. Разработка средств защиты данных и программ.

Авторизация представляет собой процесс подтверждения личности пользователя и предоставления ему доступа к определённым ресурсам или функциям. В веб-разработке авторизация обычно осуществляется через ввод имени пользователя и пароля, которые используются для доступа к защищённым данным или выполнения определённых действий.

Аутентификация играет ключевую роль в обеспечении информационной безопасности и предотвращении несанкционированного доступа. Также возможно управлять правами доступа пользователей, определяя, какие ресурсы они могут использовать и какие действия выполнять в приложении.

Существует несколько методов авторизации, таких как базовая аутентификация, токены доступа и OAuth. Каждый из этих методов имеет свои преимущества и недостатки, и выбор конкретного метода зависит от потребностей приложения и необходимого уровня безопасности для защиты данных.

Тем не менее, внедрение авторизации может быть сложным процессом, требующим дополнительных ресурсов и усилий. Кроме того, некоторые приложения могут потребовать одобрения или противоречить своему основному назначению.

В целом, решение о внедрении авторизации в приложении должно приниматься с учётом потребностей пользователей, безопасности данных и целей проекта.

Преимущества авторизации:

1. Безопасность данных: Авторизация помогает предотвратить несанкционированный доступ к конфиденциальной информации.

2. Контроль доступа: Администраторы могут управлять правами пользователей и ограничивать функциональность приложения.

3. Улучшенный пользовательский опыт: Авторизация может обеспечить персонализированный опыт для пользователей, позволяя им сохранять свои настройки.

4. Предотвращение мошенничества: Она помогает избежать мошенничества, такого как взлом учётных записей и кража личных данных.

Недостатки авторизации:

1. Усложнение процесса входа: Добавление авторизации может сделать процесс входа более сложным для пользователей.

2. Проблемы с забытыми паролями: Пользователи могут забыть свои пароли и столкнуться с трудностями при доступе к приложению.

3. Расходы на поддержку: Предприятиям необходимо выделять дополнительные ресурсы на поддержку пользователей, испытывающих трудности с входом в систему.

4. Ограничения доступа: Авторизация может ограничить доступ некоторых пользователей к приложению, особенно если они сталкиваются с техническими проблемами или забытыми учётными данными.

Отказ от авторизации в моём приложении объясняется тем, что оно ориентировано на специалистов в определённых областях, имеющих доступ к конкретным данным и ресурсам. В этом случае пользователи приложения принадлежат к узкому профессиональному кругу, и необходимость в разрешениях сводится к минимуму, так как нет нужды делиться доступом или защищать конфиденциальную информацию от посторонних лиц. Поэтому отсутствие авторизации упрощает использование приложения для целевой аудитории и улучшает общий пользовательский опыт. В конечном счёте, добавление авторизации может привести к функциональной избыточности и усложнению, что не соответствует целям и задачам приложения.