Содержание

[ВВЕДЕНИЕ 5](#_Toc69161779)

[1 Описание предметной области 7](#_Toc69161780)

[2 Проектирование системы управления базами данных 9](#_Toc69161781)

[2.1 Входные данные 12](#_Toc69161782)

[2.2 ER-диаграмма 14](#_Toc69161783)

[2.3 Нормализация таблиц 15](#_Toc69161784)

[2.3.1 Первая нормальная форма (1NF) 16](#_Toc69161785)

[2.3.2 Вторая нормальная форма (2NF) 17](#_Toc69161786)

[2.3.3 Третья нормальная форма (3NF) 18](#_Toc69161787)

[2.4 Выбор модели 20](#_Toc69161788)

[2.6 Структура таблиц 22](#_Toc69161789)

[2.7 Схема данных, описание связей 26](#_Toc69161790)

[2.8 Описание запросов 27](#_Toc69161791)

[2.8.1 Запрос «Информация по владельцам» 28](#_Toc69161792)

[2.8.2 Запрос «Поиск по ГОСномеру» 28](#_Toc69161793)

[2.8.3 Запрос «Количество услуг по маркам машин» 29](#_Toc69161794)

[2.8.4 Запрос «К оплате» 30](#_Toc69161795)

[2.8.5 Запрос «Ремонты весной» 31](#_Toc69161796)

[2.8.6 Запрос «Количество выполненных заказов без задержек» 33](#_Toc69161797)

[2.9 Описание форм 34](#_Toc69161798)

[2.9.1 Главная кнопочная форма 34](#_Toc69161799)

[2.9.2 Форма «Диаграмма прибыли по месяцам» 35](#_Toc69161800)

[2.9.3 Форма «Информация о заказах» 36](#_Toc69161801)

[2.9.4 Форма «Информация об автомобилях» 38](#_Toc69161802)

[2.9.5 Форма «Формирование заказа» 39](#_Toc69161803)

[2.9.6 Форма «Информация о владельце» 40](#_Toc69161804)

[2.10 Описание отчетов 41](#_Toc69161805)

[2.10.1 Отчет «Квитанция об оплате» 41](#_Toc69161806)

[2.10.2 Отчет «Мастер» 42](#_Toc69161807)

[2.10.3 Отчет «Полный отчет по владельцам» 43](#_Toc69161808)

[2.11 Администрирование и защита СУБД 43](#_Toc69161809)

[ЗАКЛЮЧЕНИЕ 47](#_Toc69161810)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 48](#_Toc69161811)

# **ВВЕДЕНИЕ**

В последние годы потребность в ремонте и сервисе техники стала одной из самых насущных в жизни современного цивилизованного человечества. С тех пор как появились первые механические устройства возникла необходимость в их обслуживании и ремонте. Появляется рост предложений со стороны автосервисов, что приводит к серьезной конкуренции на рынке услуг. Для того чтобы организации, предоставляющие услуги, были успешными сегодня, требуется огромная гибкость, высокое качество предоставляемых услуг и удовлетворенность потребностей клиента.

Организация потребностей клиентов в автосервисе является нестандартной задачей: большинство заказов выполняется в течение нескольких дней или даже часов, что затрудняет отслеживание всей работы, проделанной с сервисом, автомобилями, клиентами и сроками. Для решения проблемы бухгалтерского учета и повышения качества обслуживания необходимо использование современных информационных технологий, основанных на использовании автоматизированных информационных систем.

В этом случае удобное решение - база данных, в которой хранится вся информация о клиентах компании и их заказах.

В данной курсовой работе будут рассмотрены разновидности услуг, предоставляемые на СТО, приведены примеры клиентской базы, прайс-листы с ценой и наименованием услуг, фамилиями мастеров - принимающих автомобили на приём и автослесарей, производящих работу – что подчеркивает актуальность темы.

Таким образом, была сформулирована цель данной работы – закрепление навыков создания и управления базами данных, анализ и изучение соответствующей предметной области, практического применения, оптимизации и упрощения работы автосервиса, а также ведение учета заказов, работ мастеров и другой информации, связанной с работой предприятия.

Для достижения цели поставлены задачи:

* изучить теоретические сведенья, связанные с работой автосервиса;
* выбрать и описать инструментальные средства реализации базы данных;
* разработать структуру будущей базы данных;
* построить интерфейс пользователей;
* обеспечить защиту базы данных;
* составить ER-диаграмму предметной области;
* создать формы для заполнения таблиц и формы для отражения информации из базы.

# **1 Описание предметной области**

Мир движется вперёд каждый день недели, каждую минуту и любой час. Сегодня путешествия - это не то что было пару лет назад. Люди переезжают с одного места в другое по личным или профессиональным причинам. Некоторые из них даже путешествуют неделями или месяцами. Гонка, цель которой — добраться до одного места, а затем перейти к другому пункту назначения. Важность автосервиса в такие времена имеет первостепенное значение. Он с лёгкостью может как поспособствовать своевременному выполнению задач, так и разрушить Ваши планы на день.

Автомобильный транспорт получил широкое распространение в начале XX века. Тогда же стали появляться первые авторемонтные предприятия, ставшие прообразам современных автосервисов. До этого владельцы «самоходных повозок» обслуживали свой транспорт самостоятельно. Первые автомобильные мастерские организовывались на основе мастерских, занимающихся ремонтом экипажей или велосипедов.

Начальный этап существования автосервисов носил полукриминальный характер. Автовладельцы в основном были богатыми людьми, и механики нисколько не стеснялись пользоваться их некомпетентностью ради собственной наживы. Существуют свидетельства, подтверждающие, что служащие таких автомастерских продавали клиентам ненужные детали, воровали на смазочном масле и бензине, как в своё время кучера воровали на овсе и соломе.

Развитием отечественных автосервисов занималось Императорское российское автомобильное общество, основанное в Санкт-Петербурге в 1903 году. Общество объединило различные автомобильные клубы, которые на тот момент существовали в обеих столицах, а также в Одессе, Киеве, Днепре и Риге. Общество владело небольшим гаражом с мастерскими. В них и производился ремонт автомобилей.

Накануне Первой мировой войны в России существовало около 100 предприятий, так или иначе связанных с технической поддержкой автомобилей. Все необходимые для ремонта детали владельцы автосервисов либо выписывали из-за границы прямо от производителей, либо же приобретали в крупных торговых домах, занимающихся поставками автозапчастей.

Эпоха индустриализации сделала автомобильный транспорт более доступным для широких слоёв населения, и авторемонтные предприятия стали открываться во всех крупных городах. Тогда это были крупные промышленные единицы, специализирующиеся на комплексном обслуживании автомобилей. Сегодня подобные организации сохранились в основном в военной отрасли.

Сегодня автосервис (или станция технического обслуживания) - это организация, предоставляющая услуги населению и/или организациям по плановому техническому обслуживанию, текущему и капитальному ремонтам, устранению авто-поломок, установке дополнительного оборудования (тюнингу), восстановительному (кузовному) ремонту автотранспорта.

Автосервис представляет собой комплекс сооружений и механизмов (подъёмники, шиномонтаж, балансировка, стенд развал-схождения, установка для замены масла, промывки топливной системы, рихтовочное и покрасочно-сушильное оборудование, стенды и тестеры для диагностики электрической цепи автомобиля), а также ручной и пневматический инструмент, собранные в одном месте для полноценного комплексного ремонта и обслуживания автомобилей.

# **2 Проектирование системы управления базами данных**

*База данных (БД)* — именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области, или иначе БД — это совокупность взаимосвязанных данных при такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений в определенной предметной области. БД состоит из множества связанных файлов.

*Система управления базами данных (СУБД)* — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

По степени универсальности различаются два класса СУБД — системы общего назначения и специализированные системы.

СУБД общего назначения не ориентированы на какую-либо конкретную предметную область или на информационные потребности конкретной группы пользователей. Каждая система такого рода реализуется как программный продукт, способный функционировать на некоторой модели ЭВМ в определенной операционной обстановке. СУБД общего назначения обладает средствами настройки на работу с конкретной БД в условиях конкретного применения.

В некоторых ситуациях СУБД общего назначения не позволяют добиться требуемых проектных и эксплуатационных характеристик (производительность, занимаемый объем памяти и прочее). Тем не менее создание специализированных СУБД весьма трудоемкий процесс и для того, чтобы его реализовать, нужны очень веские основания.

В процессе реализации своих функций СУБД постоянно взаимодействует с базой данных и с другими прикладными программными продуктами пользователя, предназначенными для работы с данной БД и называемыми приложениями.

Для того чтобы СУБД успешно справлялась со своими задачами, она должна обладать определенными функциями:

*Управление данными во внешней памяти*

Данная функция предоставляет пользователям возможности выполнения самых основных операций, которые осуществляются с данными, — это сохранение, извлечение и обновление информации. Она включает в себя обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для ускорения доступа к данным.

*Управление транзакциями*

Транзакция — это последовательность операций над БД, рассматриваемых СУБД как единое целое. Транзакция представляет собой набор действий, выполняемых с целью доступа или изменения содержимого базы данных. Примерами простых транзакций может служить добавление, обновление или удаление в базе данных сведений о некоем объекте. Сложная же транзакция образуется в том случае, когда в базу данных требуется внести сразу несколько изменений. Инициализация транзакции может быть вызвана отдельным пользователем или прикладной программой.

*Восстановление базы данных*

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев:

* мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания);
* жесткие сбои, характеризуемые потерей информации на носителях внешней памяти.

Поддержание надежности хранения данных в БД требует избыточности хранения данных, причем та часть данных, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД.

*Поддержка языков БД*

Для работы с базами данных используются специальные языки, называемые языками баз данных. В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language — язык структурированных запросов). Язык SQL позволяет определять схему реляционной БД и манипулировать данными.

*Словарь данных*

Одной из основополагающих идей рассмотренной выше трехуровневой архитектуры является наличие интегрированного системного каталога с данными о схемах, пользователях, приложениях и т. д. Системный каталог, который еще называют словарем данных, является, таким образом, хранилищем информации, описывающей данные в базе данных. Предполагается, что каталог доступен как пользователям, так и функциям СУБД. Обычно в словаре данных: содержится следующая информация:

* имена, типы и размеры элементов данных;
* имена связей;
* накладываемые на данные ограничения поддержки целостности;
* имена пользователей, которым предоставлено право доступа к данным;
* внешняя, концептуальная и внутренняя схемы и отображения между ними;
* статистические данные, например, частота транзакций и счетчики обращений к объектам базы данных.

*Управление параллельным доступом*

Одна из основных целей создания и использования СУБД заключается в том, чтобы множество пользователей могло осуществлять параллельный доступ к совместно обрабатываемым данным, если система является многопользовательской. Параллельный доступ сравнительно просто организовать, если все пользователи выполняют только чтение данных, поскольку в этом случае они не могут помешать друг другу. Однако, когда два или больше пользователей одновременно получают доступ к базе данных, конфликт с нежелательными последствиями легко может возникнуть, например, если хотя бы один из них попытается обновить данные.

СУБД должна гарантировать, что при одновременном доступе к базе данных многих пользователей подобных конфликтов не произойдет.

*Контроль доступа к данным*

СУБД должна иметь механизм, гарантирующий возможность доступа к базе данных только санкционированных пользователей и защищающий ее от любого несанкционированного доступа.

В современных СУБД поддерживается один из двух широко распространенных подходов к вопросу обеспечения безопасности данных: избирательный подход или обязательный подход.

В большинстве современных систем предусматривается избирательный подход, при котором некий пользователь обладает различными правами при работе с разными объектами. Значительно реже применяется альтернативный, обязательный подход, где каждому объекту данных присваивается некоторый классификационный уровень, а каждый пользователь обладает некоторым уровнем допуска.

*Поддержка целостности данных*

Термин целостность используется для описания корректности и непротиворечивости, хранимых в БД данных. Реализация поддержки целостности данных предполагает, что СУБД должна содержать сведения о тех правилах, которые нельзя нарушать при работе с данными, и обладать инструментами контроля за тем, чтобы данные и их изменения соответствовали заданным правилам.

## **2.1 Входные данные**

Входные данные – это данные, которые размещаются при создании базы данных администратором с клавиатуры или переносятся из других систем.

Входными данными в системе управления базами данных являются данные о автосервисе – информация о владельце, мастере, машине, заказе и вида ремонта.

Данные в базе распределяются на статическую и динамическую часть. Автомобили и владельцы автомобилей, все атрибуты этих сущностей каждый раз пополняют базу данных, а мастера, заказы, услуги являются динамическими данными.

Сущности базы данных «Автошанс»: владелец, заказы, мастер, виды обслуживания, дисконтная карта, автомобиль.

Атрибутами объекта «Автомобиль» являются:

* Техпаспорт
* Марка машины
* ГОСномер
* Цвет автомобиля
* Год выпуска
* Объем двигателя
* Коробка передач
* Тип машины

Атрибутами объекта «Владелец» являются:

* ФИО Владельца
* Номер телефона
* Адрес
* Номер дисконтной карты

Объект «Мастер» представлен такими атрибутами, как:

* ФИО Мастера
* Дата поступления на работу
* Квалификация
* Специализация
* Адрес мастера
* Номер телефона мастера

Объект «Заказы» имеет следующие атрибуты:

* № заказа
* Дата заказа
* Срок исполнения
* Задержка
* Причина задержки

Объект «Виды обслуживания», атрибутами которой будут:

* Наименование работы
* Стоимость
* Срок выполнения
* Гарантия качества

Объект «Дисконтная карта» содержит в себе следующие атрибуты:

* Номер дисконтной карты
* Дата выдачи
* Количество бонусов

## **2.2 ER-диаграмма**

*Диаграмма Entity-Relationship (Erd)* отображает отношения набора сущностей, хранящиеся в базе данных. Другими словами, ER-диаграммы формируют логическую структуру баз данных.

ER-диаграммы была предложена Питером Ченом в 1971 году, как визуальное представление концептуального подхода к моделированию для реляционной базы данных и сети.

Для данной системы управления базами данных «Автошанс» ER – диаграмма выглядит следующим образом.

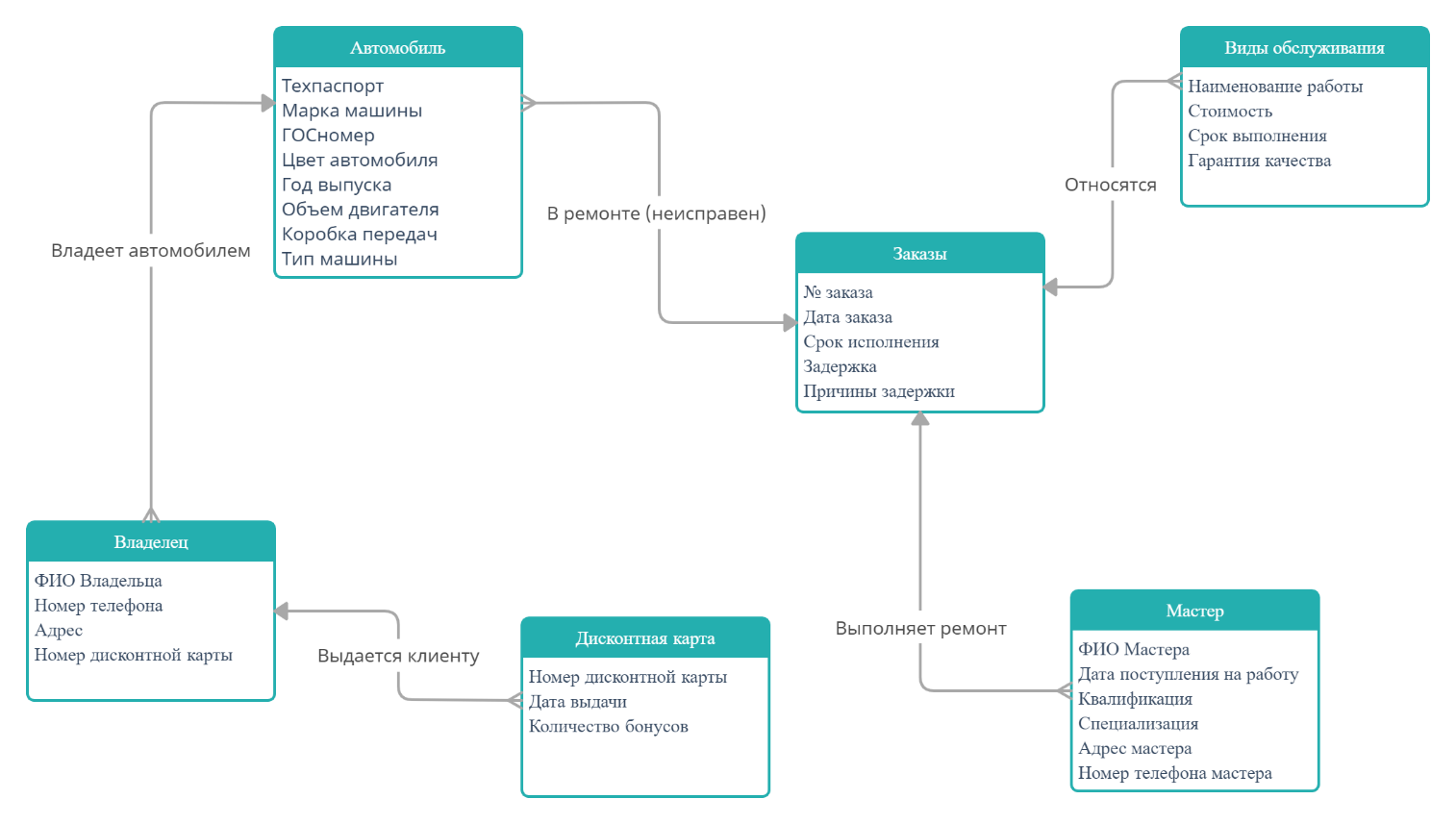


Рисунок 1 – ER - диаграмма системы управления базами данных «Автошанс»

## **2.3 Нормализация таблиц**

Нормальная форма (*Database normalization*, *Нормализация базы данных*) — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется, как совокупность требований, которым должно удовлетворять отношение.

Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, называется нормализацией. Нормализация делит большие таблицы на меньшие таблицы и связывает их, используя отношения. Конечной целью нормализации является уменьшение потенциальной противоречивости, хранимой в базе данных информации. По мнению К. Дейта, общее назначение процесса нормализации заключается в следующем:

* исключение некоторых типов избыточности;
* устранение некоторых аномалий обновления;
* разработка проекта базы данных, который является достаточно «качественным» представлением реального мира, интуитивно понятен и может служить хорошей основой для последующего расширения;
* упрощение процедуры применения необходимых ограничений целостности.

Устранение избыточности производится, как правило, за счёт декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты (то есть факты, не выводимые из других хранимых фактов).

Создании и развитии теории нормализации принимали участие многие учёные. Однако первые три нормальные формы и концепцию функциональной зависимости предложил Э. Кодд.

### 2.3.1 Первая нормальная форма (1NF)

Таблица находится в первой нормальной форме **1НФ** тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

В реляционной модели отношение всегда находится в первой нормальной форме по определению понятия отношение. Что же касается различных таблиц, то они могут не быть правильными представлениями отношений и, соответственно, могут не находиться в 1НФ.

Таблица 1. Поля первой нормальной формы

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ФИО Мастера | Дата поступления на работу | Квалификация | Специализация | | Адрес мастера | Номер телефона мастера | Наименование работы |
|  |  |  |  | |  |  |  |
| Стоимость | Срок выполнения | Гарантия качества | № заказа | | Дата заказа | Срок исполнения | Задержка |
|  |  |  |  | |  |  |  |
| Причина задержки | Техпаспорт | Марка машины | ГОСномер | | Цвет автомобиля | Год выпуска | Объем двигателя |
|  |  |  |  | |  |  |  |
| Коробка передач | Тип машины | ФИО Владельца | Номер телефона | Адрес | Номер дисконтной карты | Дата выдачи | Количество бонусов |
|  |  |  |  | |  |  |  |

### 2.3.2 [Вторая нормальная форма](https://codewiki.imagetube.xyz/w/index.php?title=%D0%92%D1%82%D0%BE%D1%80%D0%B0%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0&action=edit&redlink=1) (2NF)

Таблица находится во второй нормальной форме тогда и только тогда, когда все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом. Функционально полная зависимость означает, что если потенциальный ключ является составным, то атрибут зависит от всего ключа и не зависит от его частей.

Таблица 2. Мастер, вторая нормальная форма

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код Мастера | ФИО Мастера | Дата поступления на работу | Квалификация | Специализация | Адрес мастера | Номер телефона матсера |

Таблица 3. Виды обслуживания, вторая нормальная форма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код работы | Наименование работы | Стоимость | Срок выполнения | Гарантия качества |

Таблица 4. Заказы ремонта, вторая нормальная форма

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № Заказа | Дата заказа | Техпаспорт | Код работы | Код Мастера | Срок исполнения | Задержка | Причина задержки |

Таблица 5. Автомобиль, вторая нормальная форма

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Техпаспорт | Марка машины | ГОСномер | Цвет автомобилия | Год выпуска | Объем двигателя | Коробка передач | Тип машины |

Таблица 6. Владелец, вторая нормальная форма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ГОСномер | ФИО Владельца | Номер телефона | Адрес | Номер дисконтной карты |

Таблица 7. Дисконтная карта, вторая нормальная форма

|  |  |  |
| --- | --- | --- |
| Номер дисконтной карты | Дата выдачи | Количество бонусов |

### 2.3.3 [Третья нормальная форма](https://codewiki.imagetube.xyz/w/index.php?title=%D0%A2%D1%80%D0%B5%D1%82%D1%8C%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0&action=edit&redlink=1) (3NF)

Таблица находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме, и отсутствуют транзитивные функциональные зависимости не ключевых атрибутов от ключевых.

Таблица 8. Мастер, третья нормальная форма

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код Мастера | ФИО Мастера | Дата поступления на работу | Код квалификации | Адрес мастера | Номер телефона мастера |

В таблице 8 «Мастер» назначается ключевое поле «Код мастера», которое будет связано с таблицей 11 «Заказы». Так же добавлено поле «Код квалификации» для устранения избыточности данных.

Таблица 9. Специализация, третья нормальная форма

|  |  |  |
| --- | --- | --- |
| Код квалификации | Квалификация | Специализация |

В качестве ключевого добавлено поле «Код квалификации» в таблице 9 «Специализация», для установки связи с таблицей 8 «Мастер».

Таблица 10. Виды обслуживания, третья нормальная форма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код работы | Наименование работы | Стоимость | Срок выполнения | Гарантия качества |

В таблице 10 «Виды обслуживания» назначается ключевое поле «Код работы», которое будет связано с таблицей 11 «Заказы».

Таблица 11. Заказы, третья нормальная форма

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № Заказа | Дата заказа | Техпаспорт | Код собственника | Код работы | Код Мастера | Код исполнения |

Для устранения избыточности и транзитивных зависимости были добавлены искусственные поля «Код работы», «Код Исполнения», которые в свою очередь служат для соединения с таблицами 10 «Виды обслуживания», 12 «Готовность заказа».

Таблица 12. Готовность заказа, третья нормальная форма

|  |  |  |  |
| --- | --- | --- | --- |
| Код исполнения | Срок исполнения | Задержка | Причины задержки |

В качестве ключевого добавлено поле «Код исполнения» в таблице 12 «Готовность заказа», для установки связи с таблицей 11 «Заказы».

Таблица 13. Автомобиль, третья нормальная форма

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Техпаспорт | Марка машины | ГОСномер | Цвет автомобиля | Год выпуска | Объем двигателя | Коробка передач | Тип машины |

В таблице 13 «Автомобиль» назначается ключевое поле «Техпаспорт», которое в свою очередь будет связано с таблицей 11 «Заказы».

Таблица 14. Собственник, третья нормальная форма

|  |  |
| --- | --- |
| Код собственника | ГОСномер |

Для третьей нормальной формы формируется и выноситься поле «Код собственника» в таблицу 14 «Собственник», для устранения ошибок, при наличии более одного транспортного средства у владельца. В качестве ключа, назначается поле «ГОСномер» для связи с таблицей 13 «Автомобиль».

Таблица 15. Владелец, третья нормальная форма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код собственника | ФИО Владельца | Номер телефона | Адрес | Номер дисконтной карты |

В качестве ключа, добавлено поле «Код собственника» в таблице 15 «Владелец», для установки связи с таблицей 14 «Собственник».

Таблица 16. Дисконтная карта, третья нормальная форма

|  |  |  |
| --- | --- | --- |
| Номер дисконтной карты | Дата выдачи | Количество бонусов |

В таблице 16 «Дисконтная карта» назначается ключевое поле «Номер дисконтной карты», которое будет служить для связано с таблицей 15 «Владелец».

## **2.4 Выбор модели**

Основы современной информационной технологии составляют базы данных и системы управления базами данных (СУБД), роль которых как единого средства хранения, обработки и доступа к большим объемам информации постоянно возрастает. Большое количество разработанных к настоящему времени разнообразных СУБД связано с существованием различных моделей данных. При проектировании БД мы сталкиваемся с задачей выбора наиболее подходящей модели данных для конкретной предметной области.

Из приведенной схемы (рисунок 1) видно, что между объектами существуют связи, имеющие тип как «один ко многим». Это позволяет осуществить проектирование БД с использованием как реляционной, так и сетевой модели данных. Предпочтение было отдано реляционной модели данных, которая реализуется в пакете Office, Microsoft Access.

БД может быть основана на одной модели или на совокупности нескольких моделей. Любую модель данных можно рассматривать как объект, который характеризуется своими свойствами (параметрами), и над ней, как над объектом, можно производить какие-либо действия.

Любая модель должна обеспечивать такие операции над БД:

* поиск указанного элемента базы;
* переход от одних данных к другим;
* движение по записям;
* поиск записи;
* удаление записи;

Различают три основных типа моделей данных – реляционная, иерархическая и сетевая.

Иерархическая структура представляет собой совокупность элементов, в которой данные одного уровня подчинены данным другого уровня, а связи между элементами образуют древовидную структуру. В такой структуре исходные элементы порождают другие элементы, причем эти элементы в свою очередь порождают следующие элементы и т.д. Существенно то, что каждый порожденный элемент имеет только одного «родителя».

Существуют и более сложные – сетевые структуры, в которых каждый порожденный элемент может иметь более одного порождающего элемента. Сетевая модель данных отличается от иерархической тем, что каждый элемент сетевой структуры данных связан с любым другим элементом.

Реляционная база данных — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Строка таблицы содержит данные об одном объекте (например, товаре, клиенте), а столбцы таблицы описывают различные характеристики этих объектов – атрибутов (например, наименование, код товара, сведения о клиенте). Записи, т. е. строки таблицы, имеют одинаковую структуру – они состоят из полей, хранящих атрибуты объекта. Каждое поле, т. е. столбец, описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них отображаются различные информационные свойства объекта.

Быстрое развитие потребностей применений БД выдвигает новые требования к СУБД: естественные и эффективные представления в БД разнообразных отношений между объектами предметных областей; СУБД должна обеспечивать поиск, модификацию и сохранность данных, а также оперативный доступ (время отклика), защиту целостности данных от аппаратных сбоев и программных ошибок, разграничение прав и защита от несанкционированного доступа, поддержка совместной работы нескольких пользователей с данными.

Этим требованиям отвечают многие современные СУБД, в том числе и Access. Microsoft Access включает в себя традиционные технологии и возможности реляционных СУБД, предоставляет средства создания базы нормализованных данных и форм для диалоговой работы с ней и удобным графическим интерфейсом. С построением базы нормализованных данных тесно связана разработка и эффективная реализация задач пользователя. Для рения многих задач достаточно использовать такие объекты Access, как формы, запросы, отчеты. Эти объекты легко создаются в диалоговом режиме. Механизм обработки событий, возникающих в процессе диалоговой работы с данными, позволяет объединять в приложении пользователя отдельные запросы, формы и отчеты и получать нестандартные решения в практических приложениях пользователя.

## **2.6 Структура таблиц**

Таблица – это совокупность связанных данных, хранящихся в структурированном виде в базе данных. Она состоит из столбцов и строк.

В реляционных базах данных и плоских файлах баз данных, таблица – это набор элементов данных (значений), использующий модель вертикальных столбцов (имеющих уникальное имя) и горизонтальных строк. Ячейка – место, где строка и столбец пересекаются. Таблица содержит определенное число столбцов, но может иметь любое количество строк. Каждая строка однозначно определяется одним или несколькими уникальными значениями, которые принимают ее ячейки из определенного подмножества столбцов. Подмножество столбцов, которое уникально идентифицирует строку, называется первичным ключом.

«Таблица» – это еще один термин для «отношения»; разница между ними в том, что таблица обычно представляет собой мультимножество (набор) строк, а отношение представляет собой множество и не допускает дубликатов. Помимо обычных данных, таблицы, как правило, имеют связанные с ними метаданные, такие как ограничения, относящиеся к таблицам в целом или к значениям определенных столбцов.

Данные в таблице не обязательно физически хранятся в базе данных. Представления также функционируют, как реляционные таблицы, но их данные вычисляются во время выполнения запроса. Внешние таблицы (в СУБД informix или Oracle) также можно рассматривать как представления. Таблицы составляют основу всей информационной системы. Данные СУБД представлены в 9 таблицах.

Структура таблицы «Мастер» представлена на рисунке 2. В таблице присутствует мастер подстановки который подставляет данные «Код квалификации» из таблицы «Специализация». Ключевым полем является «Код Мастера».

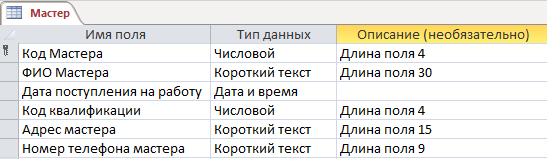


Рисунок 2 – Структура таблицы «Мастер»

Структура таблицы «Специализация» представлена на рисунке 3. Таблица кодирует поле «Квалификация» для более удобного дальнейшего использования этих данных. Ключевым полем является «Код квалификации».

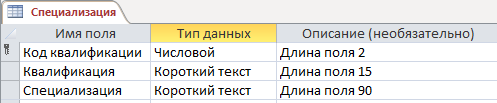


Рисунок 3 – Структура таблицы «Специализация»

Структура таблицы «Виды обслуживания» представлена на рисунке 4. Таблица содержит числовое денежное поле в котором указана стоимость выполняемой работы,а также логическое поле для отметки гарантии качества . Ключевым полем является «Код работы».

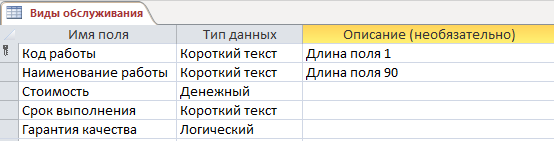


Рисунок 4 – Структура таблицы «Виды обслуживания»

Структура таблицы «Заказы» представлена на рисунке 5. Таблица содержит коды тематик и событий для оформления заказа, которые добавлены мастером подстановки из таблиц: «Собственник» , «Виды обслуживания», «Мастер», «Готовность».

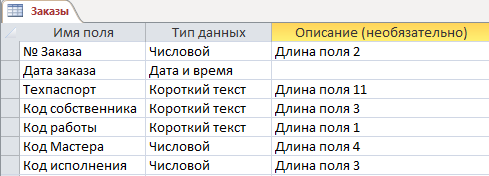


Рисунок 5 – Структура таблицы «Заказы»

Структура таблицы «Готовность» представлена на рисунке 6. Таблица содержит логическое поле задержки и информацию готовности заказа. Ключевым полем является «Код исполнения».

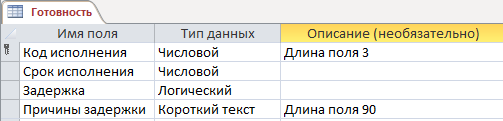


Рисунок 6 – Структура таблицы «Готовность»

Структура таблицы «Автомобиль» » представлена на рисунке 7. Ключевым полем является «Техпаспорт».

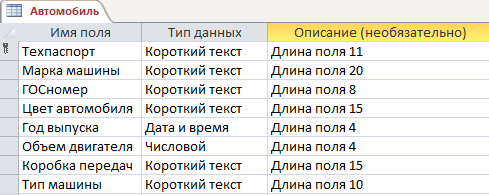


Рисунок 7 – Структура таблицы «Автомобиль»

Структура таблицы «Собственник» представлена на рисунке 8. В таблице содержится ключевое поле «ГОСномер» и внешний ключ «Код собственника» по отношению к таблице «Владелец».

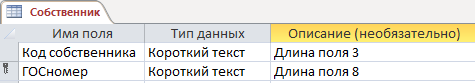


Рисунок 8– Структура таблицы «Собственник»

Структура таблицы «Владелец» представлена на рисунке 9. В таблице содержится ключевое поле «Код собственника».

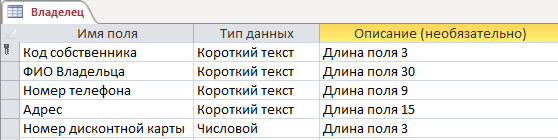


Рисунок 9 – Структура таблицы «Владелец»

Структура таблицы «Дисконтная карта» представлена на рисунке 10. Таблица содержит информацию о скидочных картах, с помощью которых можно сократить цену при оплате услуги. Ключевым полем является «Номер дисконтной карты».

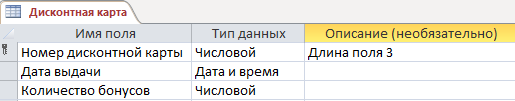


Рисунок 10 – Структура таблицы «Дисконтная карта»

## **2.7 Схема данных, описание связей**

Схема данных является графическим образом БД. Она используется различными объектами Access для определения связей между несколькими таблицами.

Схема данных обеспечивает целостность данных, единое информационное пространство. В реляционной базе данных в таблицах устонавливаются ключевые поля и внешние ключи, и связи таблиц выполняются через эти поля.

Существует несколько типов связей между сущностями: "один к одному", "один ко многим" и "многие ко многим".

* связь «один ко многим» самая распространенная;
* в связи «многие ко многим» строке таблицы А может сопоставляться несколько строк таблицы Б, и наоборот. Такие связи создаются определением третьей таблицы, которая называется таблицей соединения, чей первичный ключ состоит из внешних ключей А и Б;
* в связи «многие к одному» строке таблицы А может сопоставляться только одна строка таблицы Б, и наоборот. Связь «один к одному» создается, если для обоих связанных ключей определены ограничения первичного ключа или уникальности.

Главными таблицами (с ключевыми полями) в данной СУБД являются:

* «Мастер»;
* «Специализация»;
* «Вид обслуживания»;
* «Готовность»;
* «Автомобиль»;
* «Собственник»;
* «Владелец»;
* «Дисконтная карта».

Таблицы «Мастер», «Виды обслуживания», «Готовность», «Автомобиль» по ключевым полям «Код Мастера», «Код работы», «Код исполнения», «Техпаспорт» связаны с таблицей «Заказы» по типу связи «от одного ко многим». Таблица «Специализация» по ключевому полю «Код квалификации» связана с таблицей «Мастер» по типу связи «от одного ко многим». Таблица «Дисконтная карта» по ключевому полю «Номер дисконтной карты» связана с таблицей «Владелец» по типу связи «от одного ко многим». Таблица «Владелец» по ключевому полю «Код собственника» связана с таблицей «Собственник» по типу связи «от одного ко многим». Таблица «Собственник» по ключевому полю «ГОСномер» связана с таблицей «Автомобиль» по типу связи «от одного ко многим». Все эти связи отображены на рисунке 11.

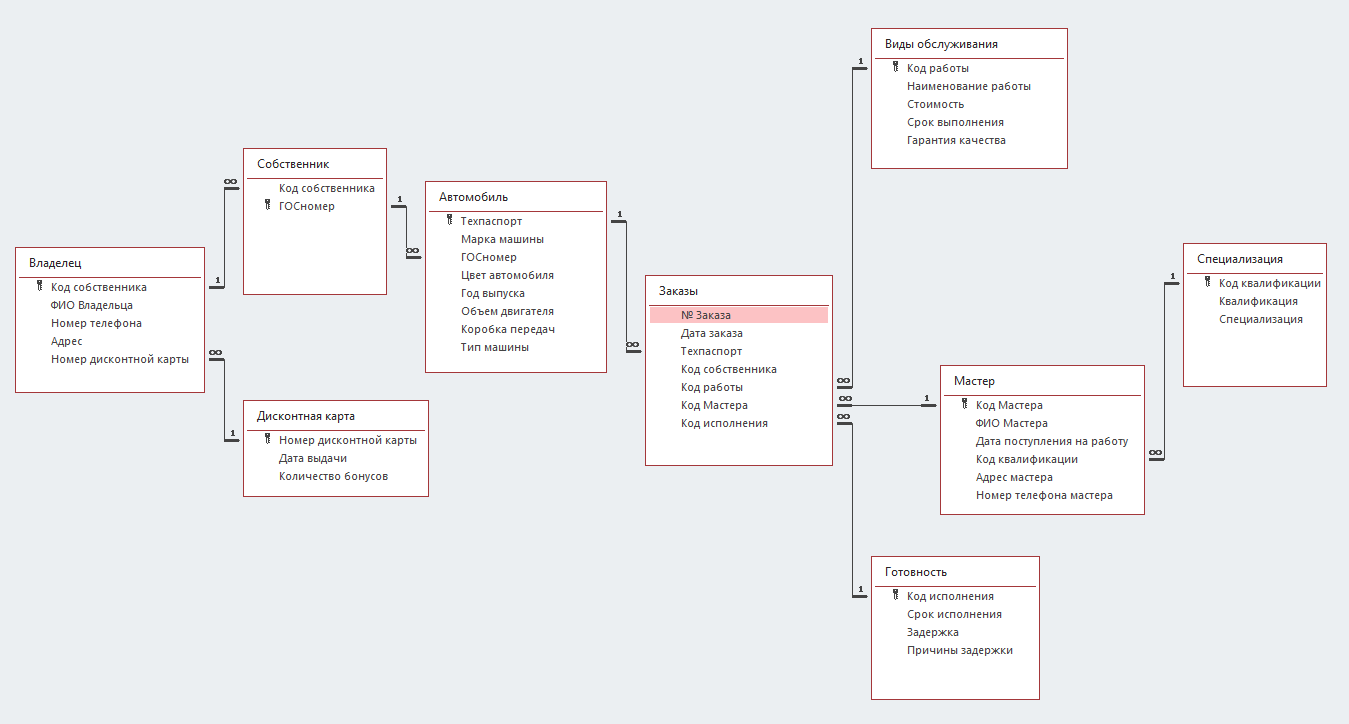


Рисунок 11 – Схема данных СУБД «Автошанс»

## **2.8 Описание запросов**

Запросы – это объект базы данных, который служит для извлечения данных из таблиц и предоставления их пользователю в удобном виде. Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную таблицу. Применение запросов позволяет избежать дублирования данных в таблицах и обеспечивает максимальную гибкость при поиске и отображении данных в базе данных.

### 2.8.1 Запрос «Информация по владельцам»

Запрос «Информация по владельцам» (рисунок 12). Запрос на языке SQL для общей сводки информации о владельцах.

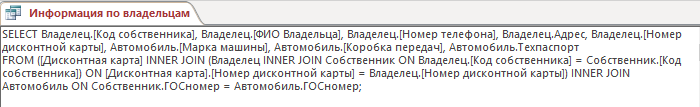


Рисунок 12 – Запрос «Информация по владельцам» в режиме SQL



Рисунок 13 – Запрос «Информация по владельцам» в режиме «Таблица»

### 2.8.2 Запрос «Поиск по ГОСномеру»

Запрос «Поиск по ГОСномеру» (рисунок 14). Запрос с параметром, служит для быстрого поиска необходимой информации о машине.

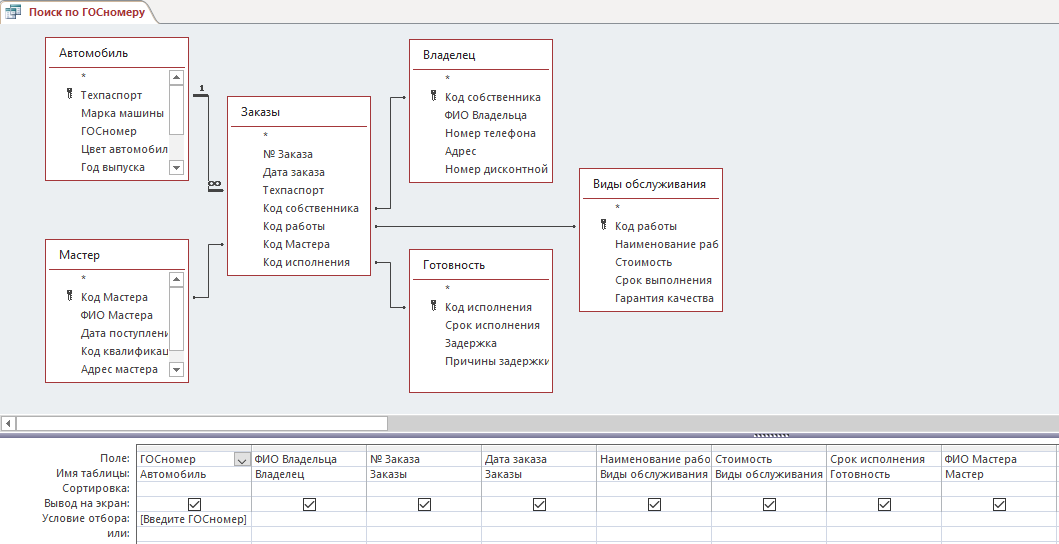


Рисунок 14 – Запрос «Поиск по ГОСномеру» в режиме «Конструктор»

На рисунке 15 изображено всплывающее окно для ввода значения параметра.

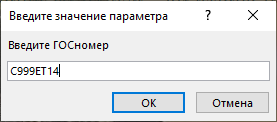


Рисунок 15 – Запрос «Поиск по ГОСномеру», ввод параметра «ГОСномер» для поиска

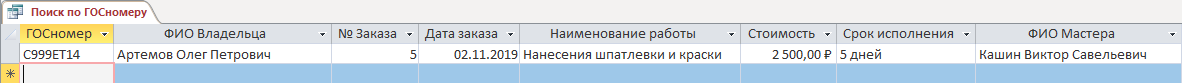
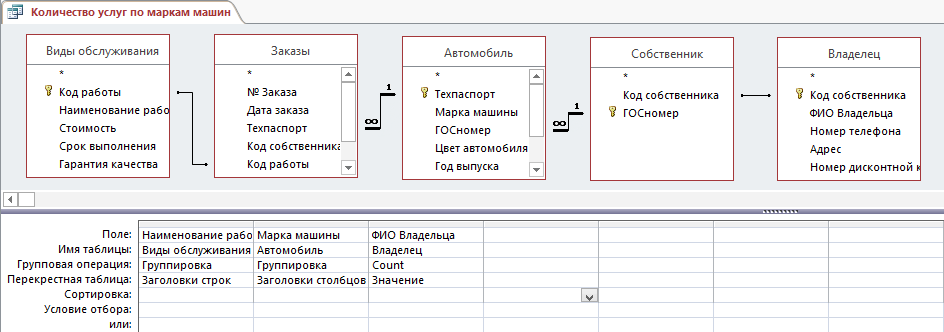


Рисунок 16 – Запрос «Поиск по ГОСномеру» в режиме «Таблица»

### 2.8.3 Запрос «Количество услуг по маркам машин»

Запрос «Количество услуг по маркам машин» (рисунок 14). Перекрестный запрос, представляет информацию по услугам, количество которых, были оказаны определенным маркам машин.

Рисунок 16 – Запрос «Количество услуг по маркам машин» в режиме конструктор



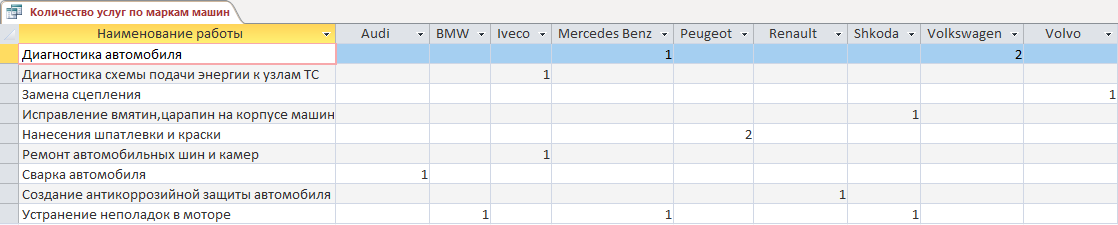
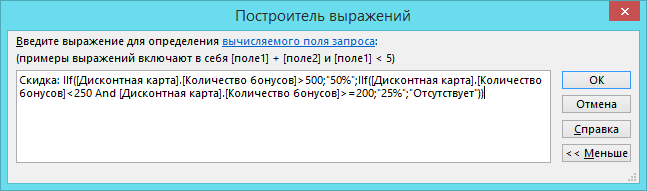


Рисунок 17 – Запрос «Количество услуг по маркам машин» в режиме таблицы

### 2.8.4 Запрос «К оплате»

Запрос «К оплате» (рисунок 22). Запрос представляет полную информацию об оплате ремонта и при необходимости высчитывание скидки клиента.

Рисунок 18 – Запрос «К оплате», поле «Скидка»



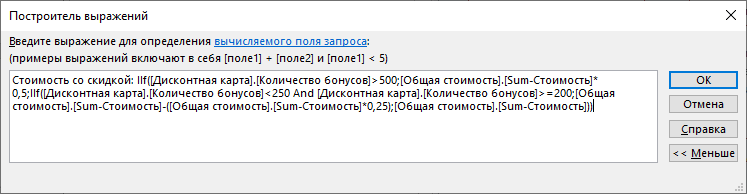


Рисунок 19 – Запрос «К оплате», поле «Стоимость со скидкой»

На рисунке 21 изображена схема запроса «К оплате». В структуре запроса присутствует поле из запроса «Общая стоимость» (рисунок 20).

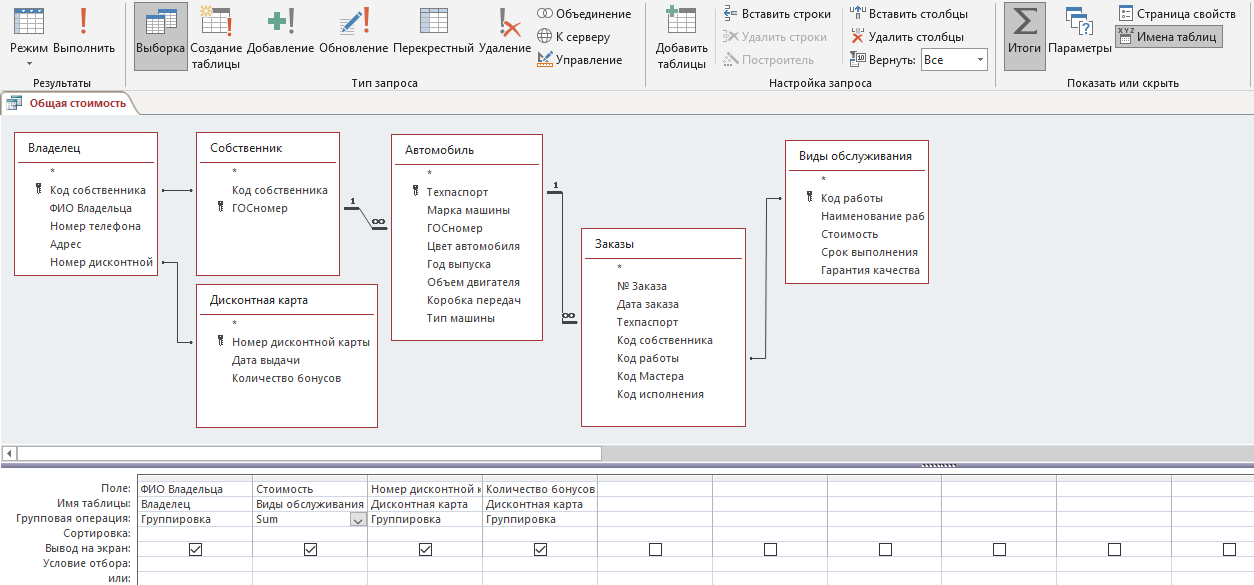
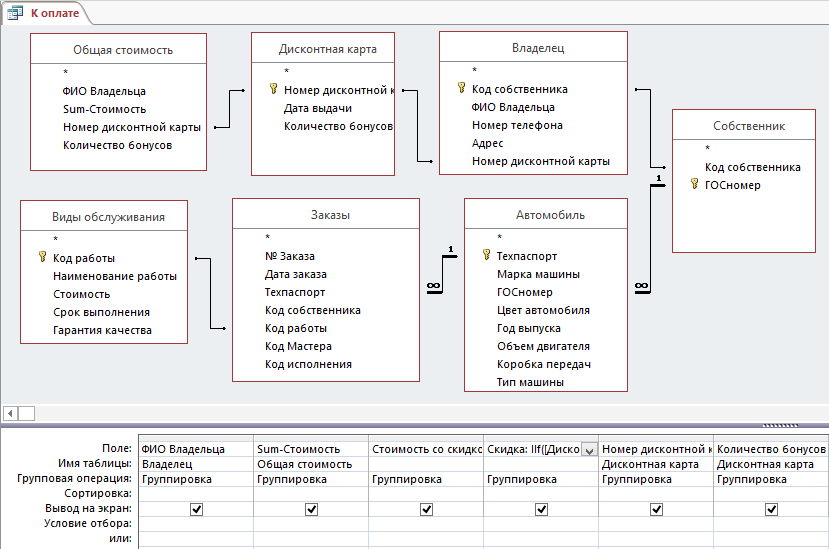


Рисунок 20 – Запрос «Общая стоимость», режим «Конструктор»

Итоговый запрос «Общая стоимость», основан на групповой операции «Sum» в поле «Стоимость».

Рисунок 21 – Запрос «К оплате», режим «Конструктор»



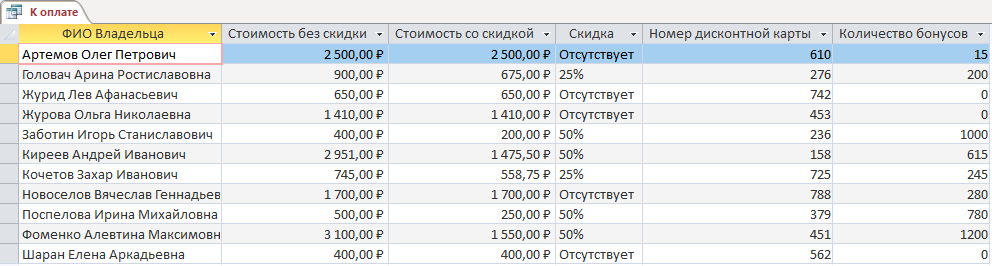


Рисунок 22 – Запрос «К оплате», режим «Таблица»

### 2.8.5 Запрос «Ремонты весной»

Запрос «Ремонты весной» (рисунок 18). Отображает информацию о количестве и месяце выполнения заказов весной.

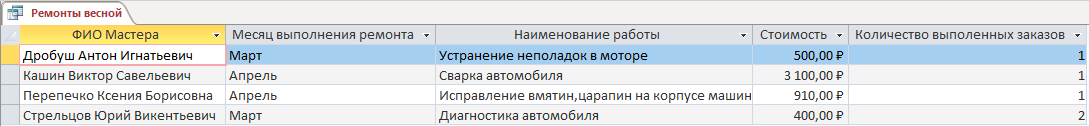


Рисунок 23 – Запрос «Ремонты весной», режим «Таблица»

На рисунке 24, показан построитель выражений для поля «Месяц выполнения ремонта». Чтобы вычислить месяц из поля «Дата заказа», используются функции из категории выражений «Дата и время»:

* Month (Дата): возвращает значение типа Variant of Integer, целое число от 1 до 12 включительно, соответствующее месяцу года;
* MonthName (month; [abbreviate]): возвращает значение типа String (строка), указывающее месяц.

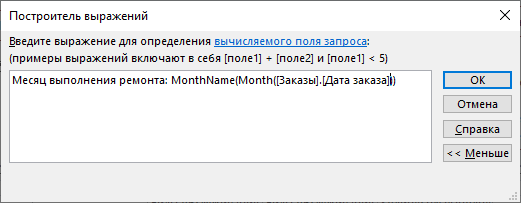


Рисунок 24 – Запрос «Ремонты весной», поле «Месяц выполнения ремонта»

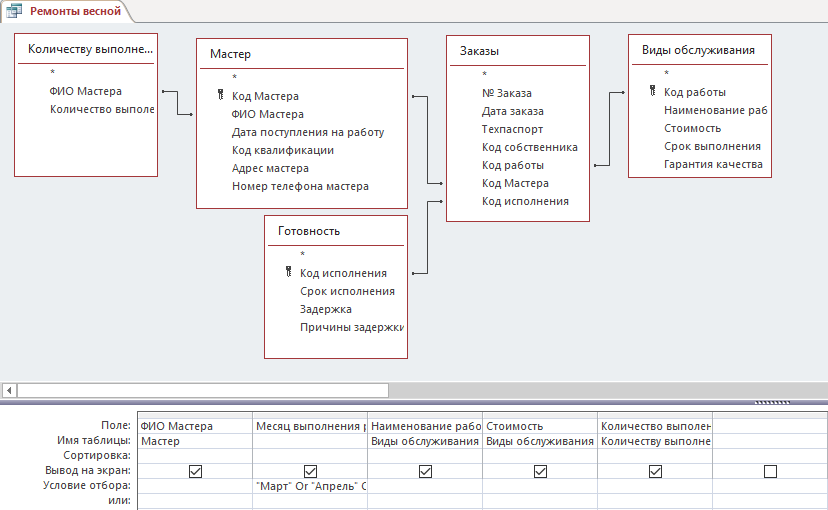


Рисунок 25 – Запрос «Ремонты весной», режим «Конструктор»

Организация отбора весенних месяцев изображена на рисунке 26, путем добавления условия в поле «Условие отбора».

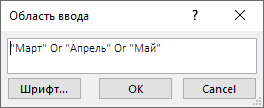


Рисунок 26 – Запрос «Ремонты весной», условие отбора поля «Месяц выполнения ремонта»

### 2.8.6 Запрос «Количество выполненных заказов без задержек»

Запрос «Количество выполненных заказов без задержек» (рисунок 28). Итоговый запрос с вычисление, предоставляет информацию о количестве выполненных заказов без задержек.

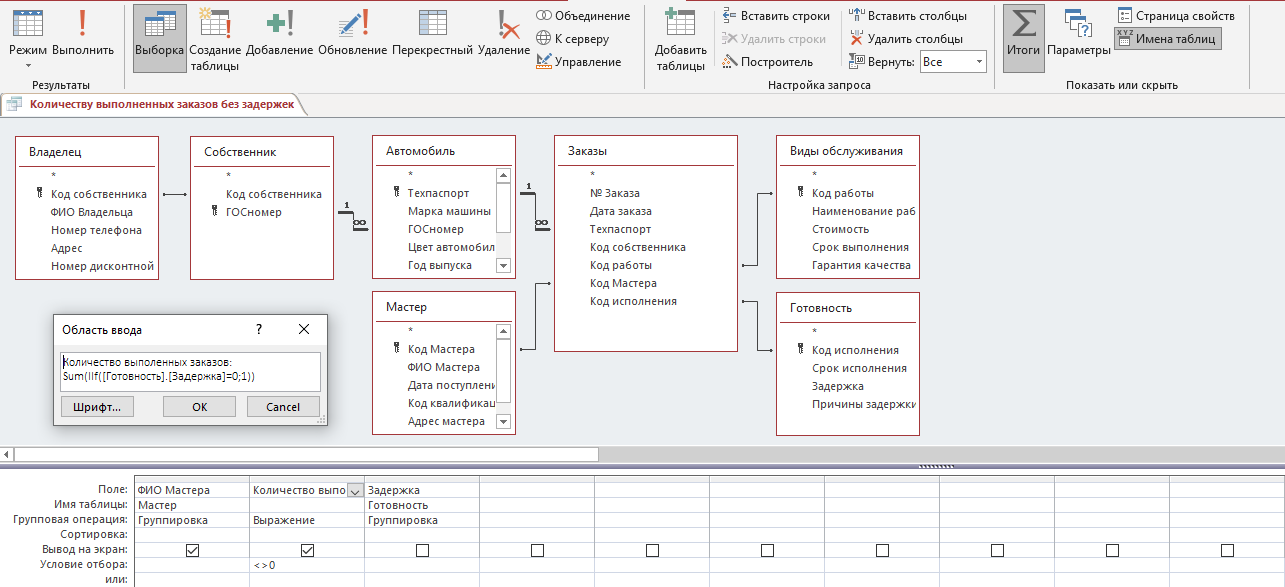


Рисунок 27 – Запрос «Количество выполненных заказов без задержек», режим «Конструктор»

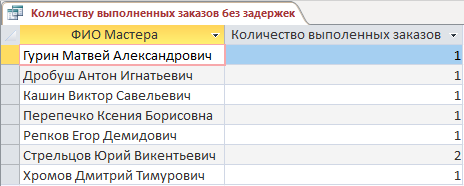


Рисунок 28 – Запрос «Количество выполненных заказов без задержек», режим «Таблицы»

На рисунке 27, изображена схема запроса «Количество выполненных заказов без задержек». Для вычисления количества выполненных заказов, используется функция «Iif» из категории выражения «Управление».

* Iif (expression; truepart; falsepart): возвращает одну из двух частей, в зависимости от результата вычисления выражения.

## **2.9 Описание форм**

В форме Access, как на витрине магазина, удобно просматривать и открывать нужные элементы. Так как форма — это объект, с помощью которого пользователи могут добавлять, редактировать и отображать данные, хранящиеся в базе данных классического приложения Access, ее внешний вид играет важную роль. Если база данных классического приложения Access используется несколькими пользователями, хорошо продуманные формы — залог точности данных и эффективности работы с ними. Основной формой при запуске базы данных является главная кнопочная форма которая содержит набор тематических страниц для получения информации по конкретным запросам пользователя.

### 2.9.1 Главная кнопочная форма

Главная кнопочная форма создается при помощи «Пустая форма» и нужна для навигации по СУБД. Главная страница разделена на категории «Запросы», «Формы», «Отчеты» и содержит следующие кнопки: «Анализ ремонтов по месяцам», «Поиск по ГОСномеру», «Сводка по выполненным заказам», «Ремонты весной», «Количество выполненных заказов без задержек», «Количество услуг по маркам машин», «Диаграмма прибыли по месяцам», «Выбор диапазона дат заказов», «Информация об автомобиле», «Информация о владельце», «Формирование заказа», «К оплате», «Поиск по мастеру», «Полный отчет по владельцам», «Выход из приложения». Каждый из разделов предоставляет сотруднику, перейти в необходимую вкладку с помощью макрокоманд:

* ОткрытьЗапрос - используется для открытия запроса на выборку или перекрестного запроса в Режим таблицы, Конструктор или предварительном просмотре. Это действие выполняет запрос на изменение, с последующим предоставлением выбора в режиме ввода данных;
* ОткрытьФорму - применяется для открытия форма в представление формы, Конструктор, предварительного просмотра или Режим таблицы. Предоставляет выбор режима ввода данных и окна для формы и ограничения отображаемой в форме записи;
* ОткрытьОтчет – позволяет открыть отчет в Конструктор или предварительного просмотра, либо отправить отчет непосредственно на принтер, где при необходимости можно ограничить записи, которые печатаются в отчете.

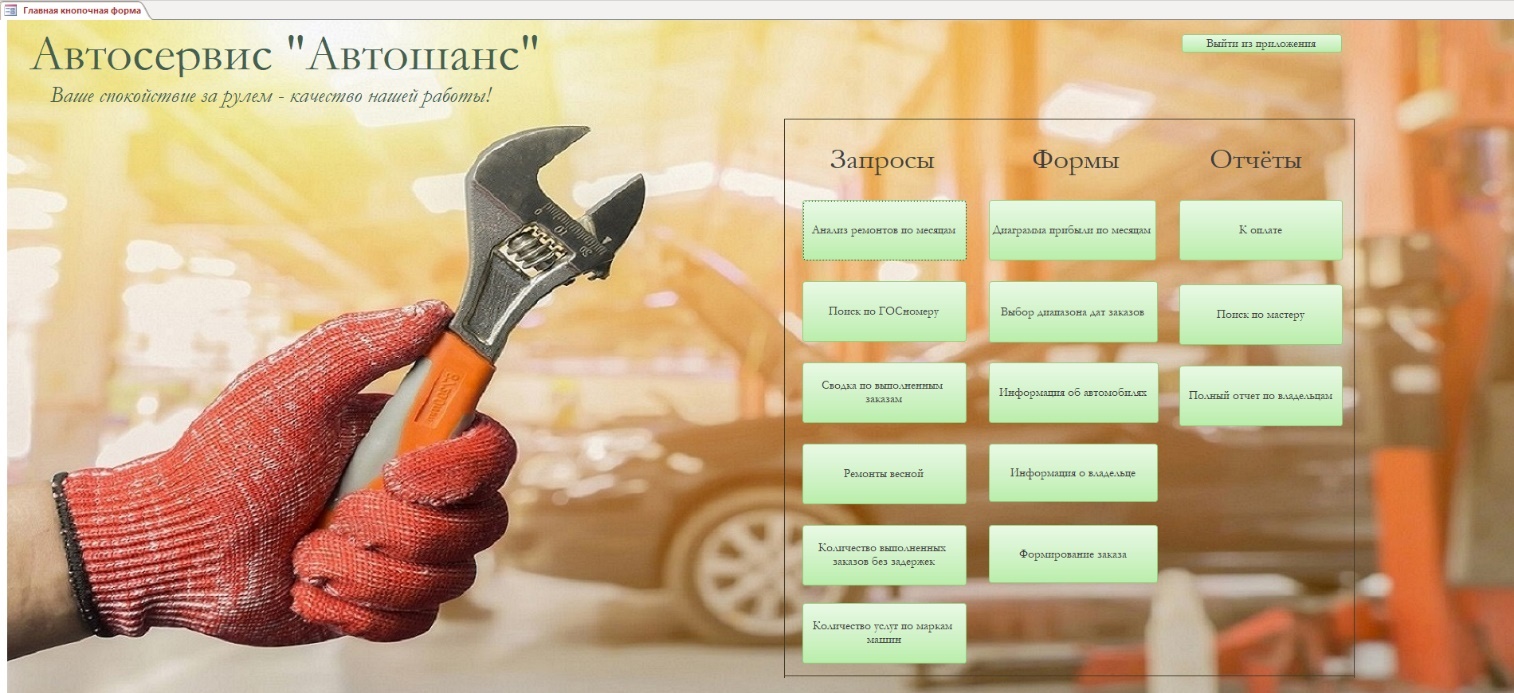


Рисунок 29 – Форма «Главная кнопочная форма», режим «Режим форма»

### 2.9.2 Форма «Диаграмма прибыли по месяцам»

Форма «Диаграмма прибыли по месяцам» (рисунок 30). Представляет из себя графическое представление, визуализации данных прибыли по месяцам.

С помощью формы, можно проанализировать статистику оборота прибыли денежных средств, за определенные промежутки времени.

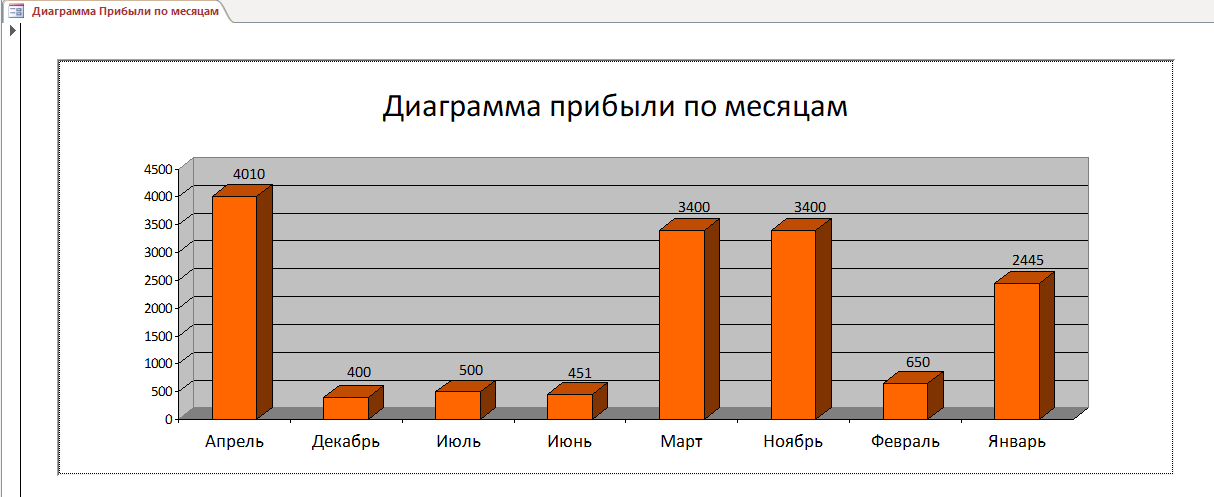


Рисунок 30 – Форма «Диаграмма прибыли по месяцам», режим «Режим форма»

### 2.9.3 Форма «Информация о заказах»

Форма «Информация о заказах» (рисунок 31). В данной форме имеется два текстовых поля «С», «По», в которых вписываются необходимые даты для фильтрации заказов.

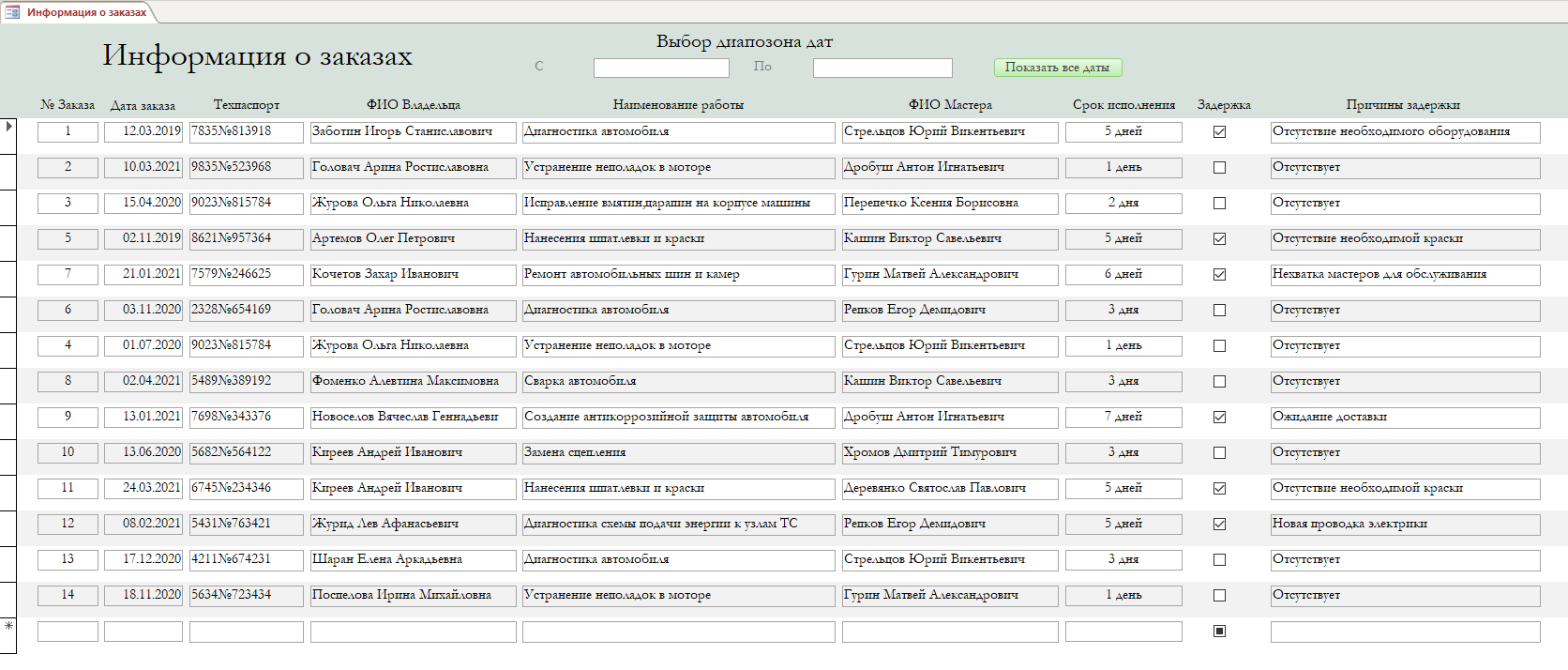


Рисунок 31 – Форма «Информация о заказах», режим «Режим формы»

Ниже представленны макросы с помощью которых функционирует данная форма.

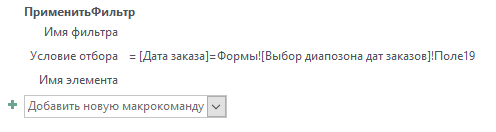


Рисунок 32 – Макрос для формы «Информация о заказах», поле «С»

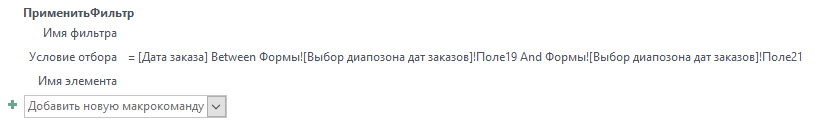


Рисунок 33 – Макрос для формы «Информация о заказах», поле «ПО»

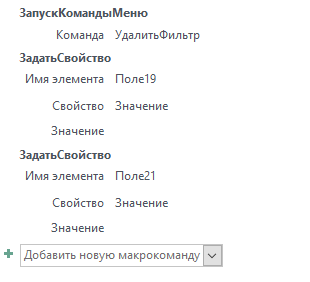


Рисунок 34 – Макрос для формы «Информация о заказах», кнопка «Показать все даты»

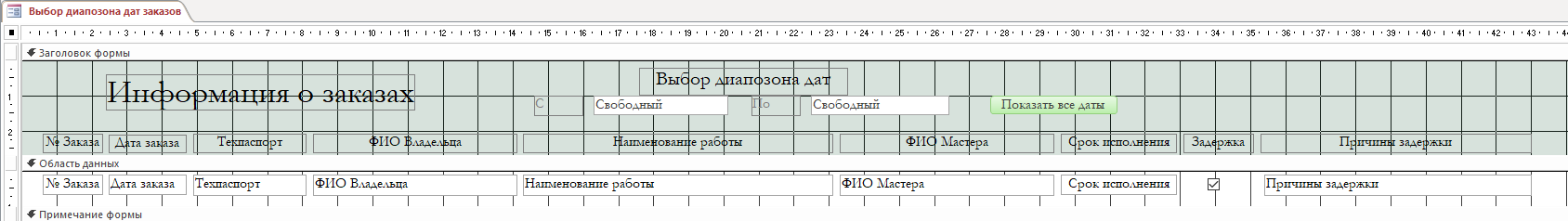


Рисунок 35 – Форма «Информация о заказах», режим «Конструктор»

### 2.9.4 Форма «Информация об автомобилях»

Форма добавления «Информация об автомобилях» (рисунок 36). Отображает и позволят сотруднику ввести данные об автомобиле, поступившем на ремонт.

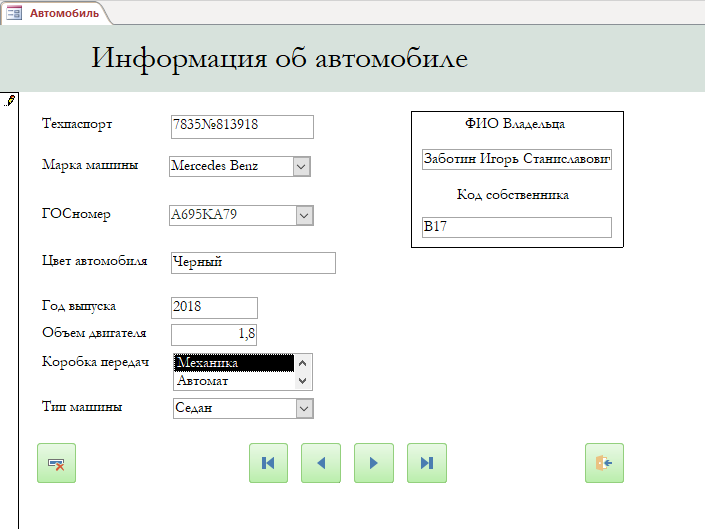


Рисунок 36 – Форма «Информация об автомобилях», режим «Режим формы»

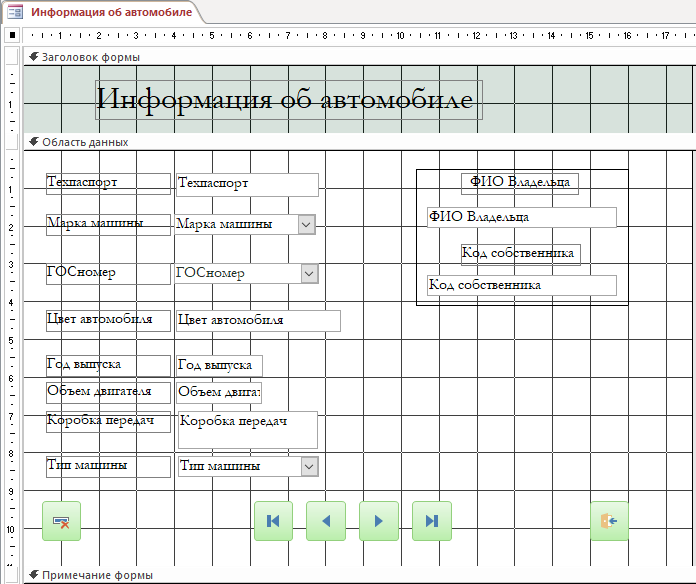


Рисунок 37 – Форма «Информация об автомобилях», режим «Конструктор»

На форме присутствуют компонеты кнопок, для организации перехода по записям клиентов и кнопка «Выход из формы».

### 2.9.5 Форма «Формирование заказа»

Форма добавления «Формирование заказа» (рисунок 38). Форма предоставляет сотруднику автосервиса «Автошанс», просматривать и формировать заказы на основе уже введенных данных. На странице формы находятся компонеты кнопок, для организации перехода по записям клиентов и кнопка «Выход из формы».

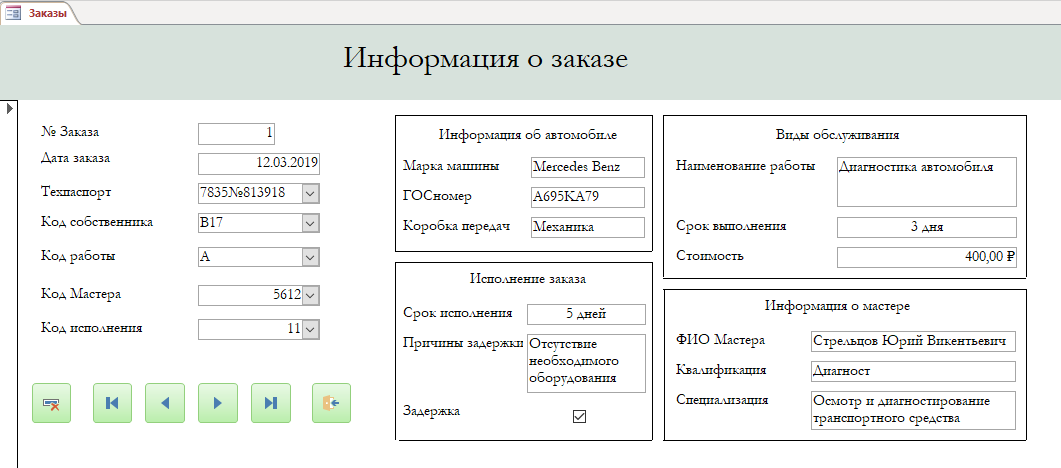


Рисунок 38 – Форма «Формирование заказа», режим «Режим формы»

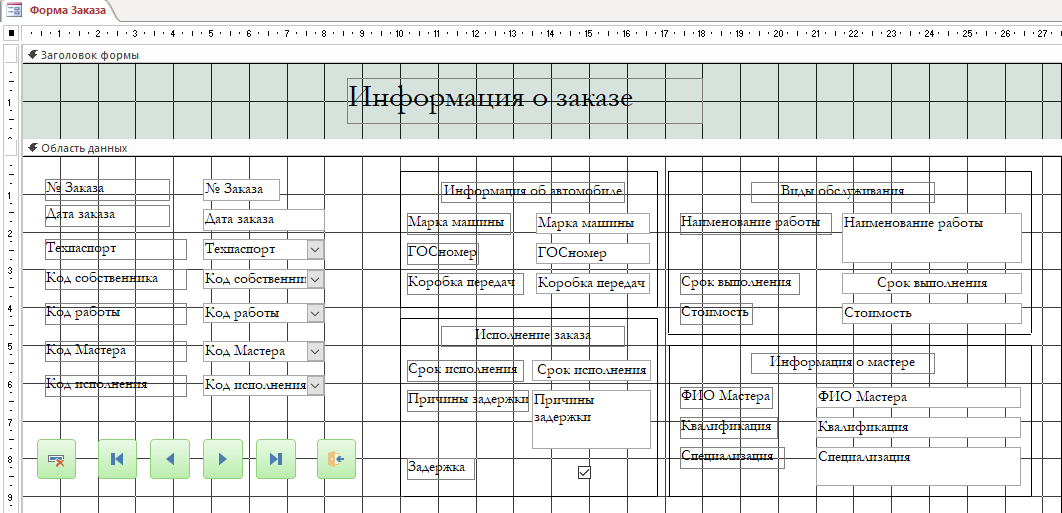


Рисунок 39 – Форма «Формирование заказа», режим «Конструктор»

### 2.9.6 Форма «Информация о владельце»

Форма «Информация о владельце» (рисунок 40). В данной форме присутствуют нескольких подчиненных форм, для расширенной информации о владельце, его транспорте и активных заказов.

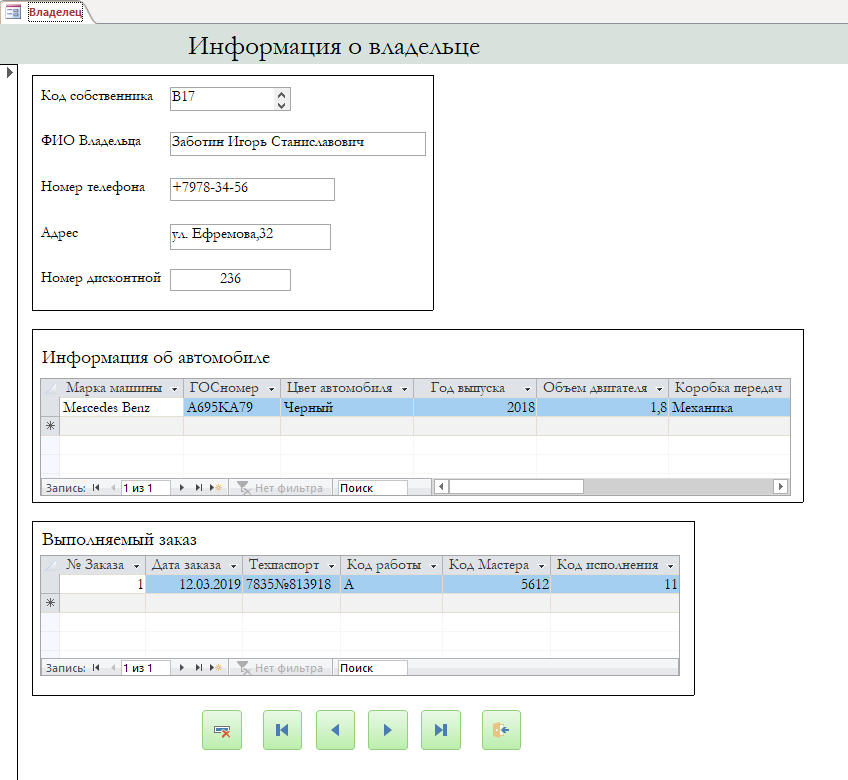


Рисунок 40 – Форма «Информация о владельце», режим «Режим формы»

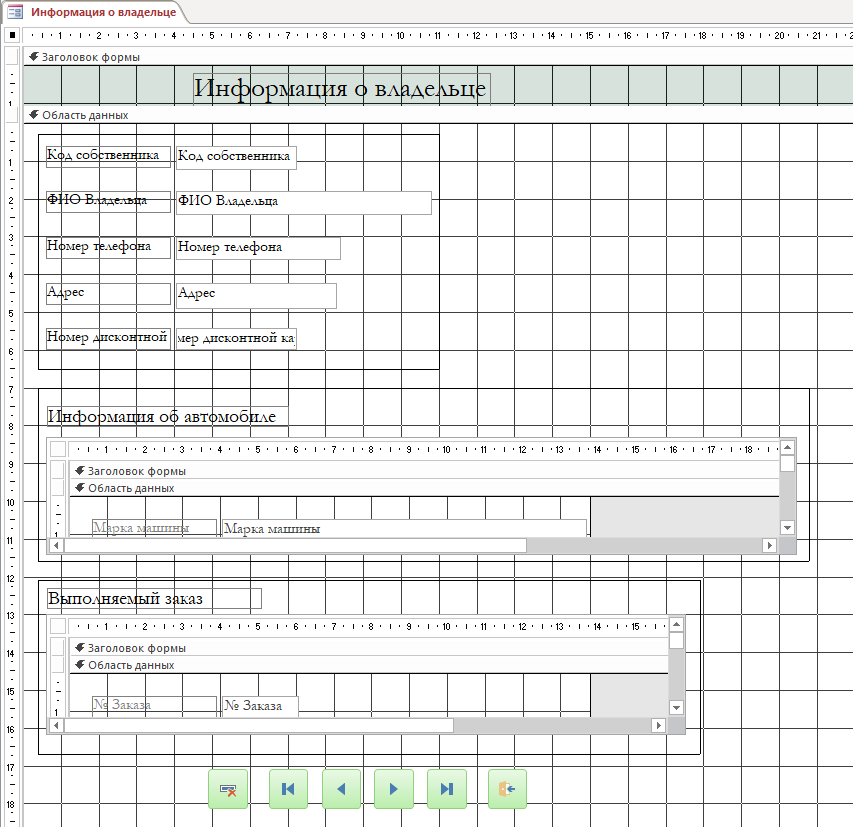


Рисунок 41 – Форма «Информация о владельце», режим «Конструктор»

Для организации переходя по записям,используются кнопки «Следующая запись», «Предыдущая запись», «Последняя запись», «Первая запись», «Последняя запись», «Выйти из приложения», «Удалить запись».

## **2.10 Описание отчетов**

Отчет – это объект базы данных, который предназначен для вывода информации из баз данных, прежде всего на принтер. Отчеты позволяют выбрать из баз данных нужную пользователю информацию, оформить ее в виде документа, перед выводом на печать просмотреть на экране. Источником данных для отчета может служить таблица или запрос. Кроме данных, полученных из таблиц, в отчете могут отображаться вычисляемые поля, например, итоговые суммы.

### 2.10.1 Отчет «Квитанция об оплате»

Отчет «Квитанция об оплате» (рисунок 42). Отчет создан для итогового просмотра стоимости заказов каждого владельца, в случае наличия необходимых бонусов, хранящихся на «Дисконтной карте», клиенты выписывается скидка в соотношении накопленных бонусов.



Рисунок 42 – Отчет «Квитанция об оплате», режим «Представление отчета»

Отчет создан на основе запроса «К оплате» (рисунок 22).



Рисунок 43 – Отчет «Квитанция об оплате», режим «Конструктор»

### 2.10.2 Отчет «Мастер»

Отчет «Мастер» (рисунок 45). При открытие запроса появляется окно (рисунок 44), в котором необходимо написать фамилию мастера, после чего покажется вся информация по данному запросу.

Рисунок 44 – Отчет «Квитанция об оплате», ввод параметра для поиска

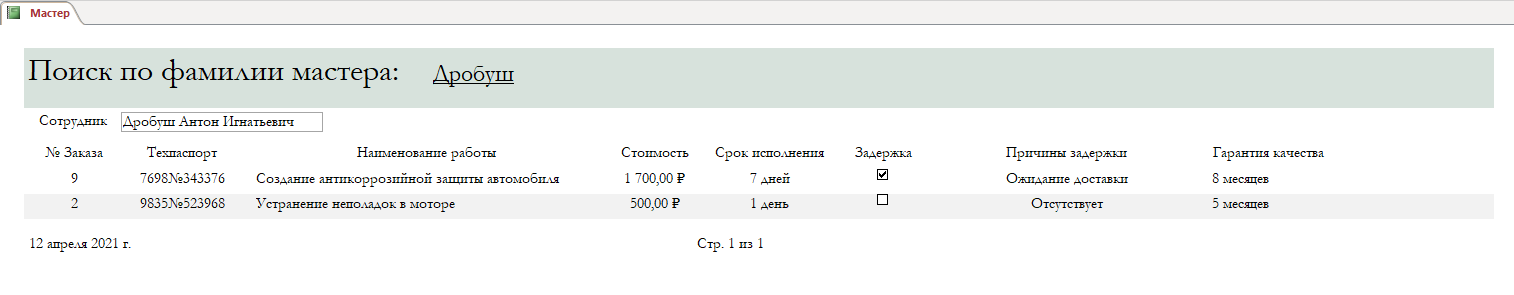


Рисунок 45 – Отчет «Мастер», режим «Представление отчета»



Рисунок 46 – Отчет «Мастер», режим «Конструктор»

С помощью отчета «Мастер», можно найти всю информацию о выполнении заказа определенного сотрудника автосервиса «Автошанс».

### 2.10.3 Отчет «Полный отчет по владельцам»

Отчет «Полный отчет по владельцам» (рисунок 47). Отчет создан для просмотра информации по каждого владельцу, его автотранспорте, заказе и стоимости работы.

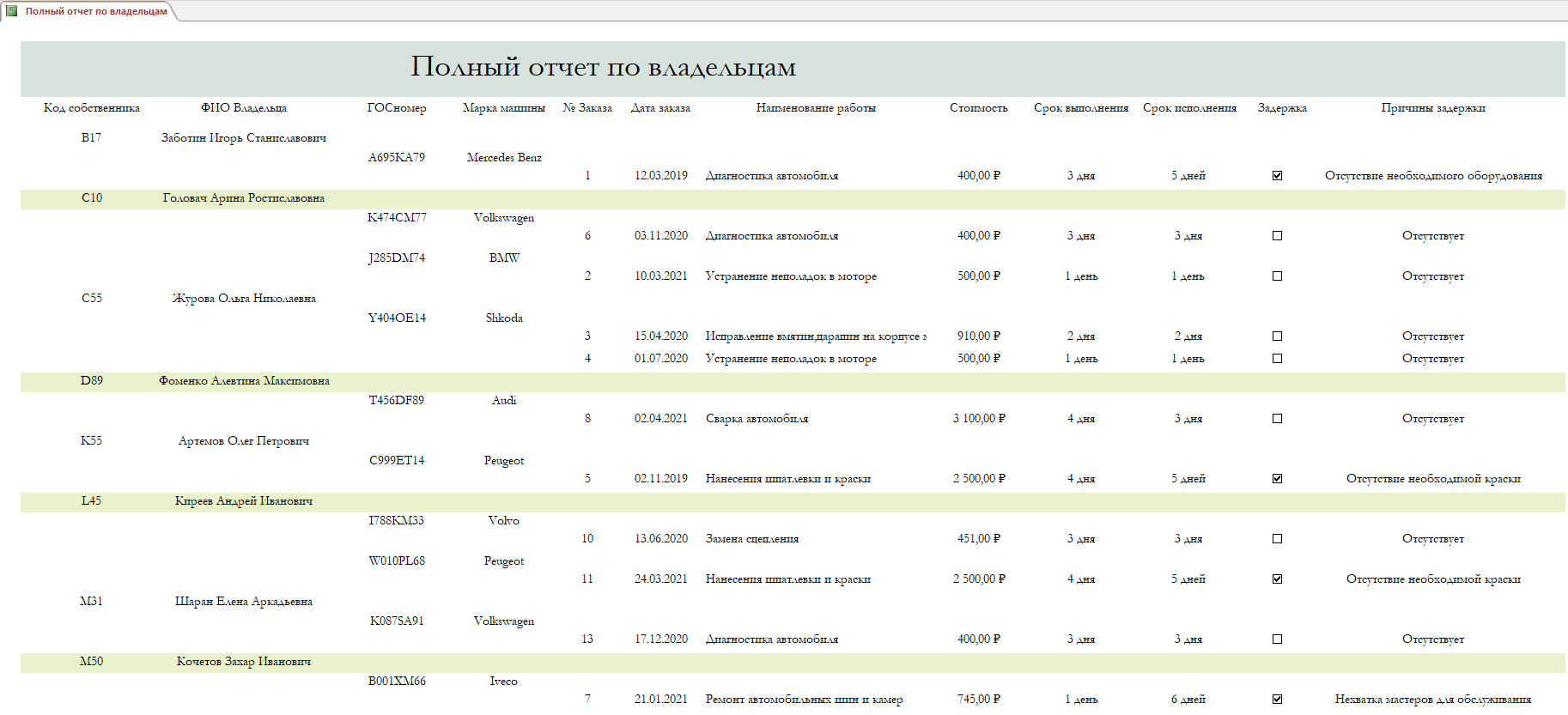


Рисунок 47 – Отчет «Полный отчет по владельцам», режим «Представление отчета»

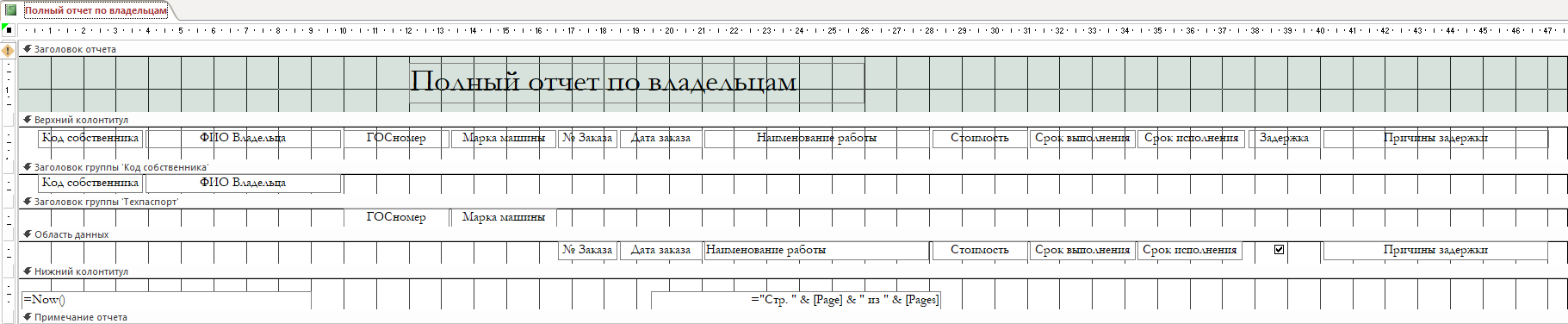


Рисунок 48 – Отчет «Полный отчет по владельцам», режим «Конструктор»

## **2.11 Администрирование и защита СУБД**

Администрирование базы данных – это функция управления базой данных (БД). Лицо ответственное за администрирование БД называется “Администратор базы данных” (АБД) или “Database Administrator” (DBA).

Необходимость персонала, обеспечивающего администрирование данными в системе БД в процессе функционирования, является следствием централизованного характера управления данными в таких системах, постоянно требующего поиска компромисса между противоречивыми требованиями к системе в социальной пользовательской среде. Хотя такая необходимость и признавалась на ранних стадиях развития технологии баз данных, четкое понимание и структуризация функций персонала, занятого администрированием, сложилось только вместе с признанием многоуровневой архитектуры СУБД.

Поскольку система баз данных может быть весьма большой и может иметь много пользователей, должно существовать лицо или группа лиц, управляющих этой системой. Такое лицо называется администратором базы данных (АБД).

В любой базе данных должен быть хотя бы один человек, выполняющий административные обязанности; если база данных большая, эти обязанности могут быть распределены между несколькими администраторами.

Средства защиты БД в различных СУБД несколько отличаются друг от друга. На основе анализа современных СУБД Borland и Microsoft можно утверждать, что средства защиты БД условно делятся на две группы, основные и дополнительные.

К основным средствам защиты информации можно отнести следующие средства:

* парольная защита;
* шифрование данных и программ;
* установление прав доступа к объектам БД;
* защита полей и записей таблиц БД.

Парольная защита представляет простой и эффективный способ защиты БД от несанкционированного доступа. Пароли устанавливаются конечными пользователями или администраторами БД. Учет и хранение паролей производится самой СУБД. Обычно пароли хранятся в определенных системных файлах СУБД в зашифрованном виде. Поэтому просто найти и определить пароль невозможно. После ввода пароля пользователю СУБД предоставляются все возможности по работе с защищенной БД.

Шифрование данных (всей базы или отдельных таблиц) применяется для того, чтобы другие программы не могли прочитать данные. Шифрование исходных текстов программ позволяет скрыть от несанкционированного пользователя описание соответствующих алгоритмов.

В целях контроля использования основных ресурсов СУБД во многих системах имеются средства установления прав доступа к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта (пользователь, создавший объект), а также администратор БД имеют все права. Остальные пользователи к разным объектам могут иметь различные уровни доступа.

По отношению к таблицам в общем случае могут предусматриваться следующие права доступа.

* просмотр (чтение) данных;
* изменение (редактирование) данных;
* добавление новых записей;
* добавление и удаление данных;
* все операции, в том числе изменение структуры таблицы.

К данным, имеющимся в таблице, могут применяться меры защиты по отношению к отдельным полям и отдельным записям. В реляционных СУБД отдельные записи специально не защищаются.

Применительно к защите данных в полях таблиц можно выделить следующие уровни прав доступа:

* полный запрет доступа;
* только чтение;
* разрешение всех операций (просмотр, ввод новых значений, удаление и изменение).

По отношению к формам могут предусматриваться две основные операции: вызов для работы и разработка (вызов Конструктора). Запрет вызова Конструктора целесообразно делать для экранных форм готовых приложений, чтобы конечный пользователь случайно не испортил приложение. В самих экранных формах отдельные элементы могут быть тоже защищены. Например, некоторые поля исходной таблицы вообще могут отсутствовать или скрыты от пользователя, а некоторые поля – доступны только для просмотра.

Отчеты во многом похожи на экранные формы, за исключением следующего. Во-первых, они не позволяют изменять данные в таблицах, а во-вторых, основное их назначение – вывод информации на печать. На отчеты, так же, как и на экранные формы, может накладываться запрет на вызов средств их разработки.

Для обеспечения защиты от несанкционированного доступа к СУБД «Автосервис», был установлен пароль при запуске базы данных (рисунок 49), а также заблокирован доступ ко всем объектам Access, предоставляя взаимодействие только с «Главная кнопочная форма» (рисунок 50).

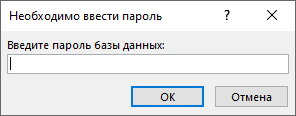


Рисунок 49 – Диалоговое окно для ввода пароля от СУБД «Автосервис»

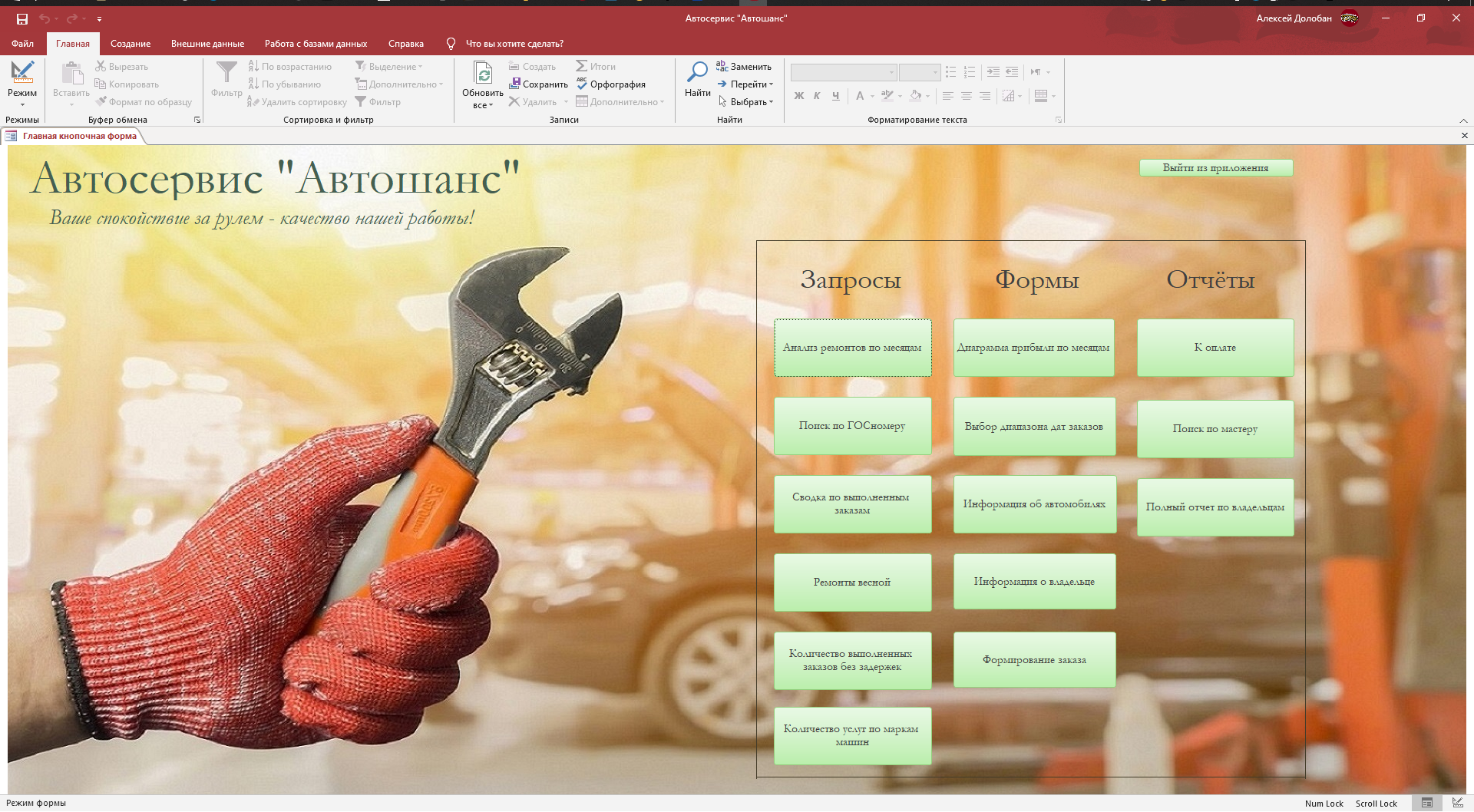


Рисунок 50 – База данных автосервис «Автошанс» после ввода пароля

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения данной работы, была создана система управления базами данных в программе Access «Автосервис», в которой представлена информация о клиентах автосервиса, мастерах и выполняемых заказах.

Сформирован эргономический пользовательский интерфейс. Приложение позволяет решать все задачи, сформулированные в задании на курсовую работу.

База данных «Автосервис» содержит полную информацию об объектах в 9 таблицах, формы «Информация о владельце», «Формирование заказа», «Диаграмма прибыли по месяцам», «Информация об автомобиле», «Главная кнопочная форма», запросы «К оплате», «Сводка по выполненным заказам», «Анализ ремонта по месяцам», «Поиск по ГОСномеру» и отчеты «Работы мастера», «Квитанция об оплате», которые отражают систему организации работы автосервиса, направленную на удовлетворение потребностей потребителя (сотрудников) посредством оказания индивидуальных услуг.

Данная работа может быть рассмотрена сотрудниками автосервиса для практического применения.

## **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Давыдов, Н.А. Производственно-техническая инфраструктура предприятий автосервиса / Н.А. Давыдов. - М.: Academia, 2018. - 154 c.
2. Голицына, О.Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2004. - 352 c.
3. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2012. - 400 c.
4. Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова. - СПб.: Питер, 2013. - 240 c.
5. Кириллов, В.В. Введение в реляционные базы данных.Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. - СПб.: БХВ-Петербург, 2012. - 464 c.
6. Кошелев, В.Е. Базы данных в ACCESS 2007: Эффективное использование / В.Е. Кошелев. - М.: Бином-Пресс, 2009. - 592 c.
7. Кузин, А.В. Базы данных: Учебное пособие для студ. высш. учеб. заведений / А.В. Кузин, С.В. Левонисова. - М.: ИЦ Академия, 2012. - 320 c.
8. Ливена, С.В. Практика увольнений за прогул. По материалам базы данных "Пакет кадровика" / С.В. Ливена. - М.: ИНФРА-М, 2008. - 51 c.
9. Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В.Ю. Пирогов. - СПб.: БХВ-Петербург, 2009. - 528 c.
10. Фуфаев, Э.В. Базы данных: Учебное пособие для студентов учреждений среднего профессионального образования / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: ИЦ Академия, 2012. - 320 c.