Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ В.Ф.УТКИНА»

Рязанский станкостроительный колледж РГРТУ

|  |  |
| --- | --- |
|  | Проект принят с оценкой:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.В. Новицкий  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |

КУРСОВОЙ ПРОЕКТ

Тема   
ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ПРЕДМЕТНОЙ ОБЛАСТИ

«Гостиница»

Специальность: 09.02.07 Информационные системы и программирование(квалификация - программист)

МДК.02.02 Технология разработки и защиты баз данных

Группа: ИСП-31

Пояснительная записка

КП.59/09.02.07

Студент: Филяк Илья Игоревич \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_

(подпись) (дата)

Рязань 2021

**Введение**

**Наименование программы**

База данных «Гостиница».

**Краткая характеристика области применения**

Содержит сведения о должностях, персонале, номерах и клиентах гостиницы. В итоговом отчете должны содержаться сведения о номере операции, номере комнаты, коде клиента, дате заселения, дате выселения, а также итоговой стоимости проживания. Область применения программы — администрирование гостиницей.

**1.Постановка задачи и анализ предметной области**

Спроектировать базу данных для предметной области **Гостиница.**

Создать информационную систему для заданной предметной области. Она должна включать в себя связанные таблицы базы данных, а также набор входных форм для их заполнения, запросы различных типов, обеспечивающие поиск и обработку хранимых данных. Система должна обеспечивать возможность добавления, изменения и удаления данных в базе и иметь удобный интерфейс для работы.

Оператором данной базы данных будет администратор гостиницы. Для выполнения своих обязанностей может быть интересна следующая информация:

* Информация о клиентах;
* Информация о персонале;
* Информация о номерах;
* Информация о должностях;
* Информация о учете работы гостиницы.

Для достижения цели курсового проекта необходимо реализовать все пункты.

Таким образом, необходимо реализовать ввод, хранение и изменение информации в базе данных:

* Информация о персонале гостиницы (Код сотрудника, ФИО сотрудника, серия и номер паспорта, адрес фактического проживания, дата рождения, должность, время начала рабочего дня, время окончания рабочего дня, телефон, статус (работает или уволен));
* Информация о клиентах (Код клиента (счетчик), ФИО, Страна, телефон клиента, серия и номер паспорта, адрес фактического проживания, дата рождения, цель приезда (не обязательно), телефон);
* Информация о номерах гостиницы (Код номера, состояние (забронировано или нет, заселен или свободен), суточная стоимость проживания, кол-во мест, оплачено/не оплачено, код сотрудника)
* Информация о учете работы гостиницы (Код операции, код номера, код клиента, дата заселения, дата выселения, стоимость проживания (суточная стоимость номера \* кол-во дней)).

Входные формы:

* Главная

Конечный продукт должен выполнять следующие функции:

* Ввод, изменение и удаление данных в таблицах (клиенты, сотрудники);
* Открытие запросов;
* Открытие отчетов;
* Вызов формы редактирования данных, из форм отображающих эти данные.

Для реализации вышеперечисленных функций конечный продукт должен содержать формы:

* Главную (начальную) форму выбора действий, из которых должны запускаться остальные формы, реализующие все функции;
* Формы редактирования информации о сотрудниках;
* Формы редактирования персональных данных клиента;
* Форма вывода отчета о работе гостиницы.

**2.Назначение проектируемой базы данных**

Спроектированная база данных предназначена для автоматизированного учета работы **Гостиницы**, которая содержит следующие данные:

* Информация о персонале гостиницы (Код сотрудника, ФИО сотрудника, серия и номер паспорта, адрес фактического проживания, дата рождения, должность, время начала рабочего дня, время окончания рабочего дня, телефон, статус (работает или уволен));
* Информация о клиентах (Код клиента (счетчик), ФИО, Страна, телефон клиента, серия и номер паспорта, адрес фактического проживания, дата рождения, цель приезда (не обязательно), телефон);
* Информация о номерах гостиницы (Код номера, состояние (забронировано или нет, заселен или свободен), суточная стоимость проживания, оплачено/не оплачено, код сотрудника);
* Информация о учете работы гостиницы (Код операции, код номера, код клиента, дата заселения, дата выселения, стоимость проживания (суточная стоимость номера \* кол-во дней))
* Информация о должностях.

Данная разработка должна использоваться для программного обеспечения ОС Microsoft Windows 8.1/10, СУБД Microsoft Access 2013/2016. Для персональных компьютеров не ниже AMD A8 - 7410.

**3.Проектирование инфологической модели данных**

**3.1 Информационные потребности пользователя (анализ запросов)**

При разработке данного курсового проекта была выбрана следующая предметная область: «**Гостиница**».

В ней необходимо отразить:

* Персональные данные клиента (Код клиента (счетчик), ФИО, Страна, телефон клиента, серия и номер паспорта, адрес фактического проживания, дата рождения, цель приезда (не обязательно), телефон);
* Доход за день;
* Доход за месяц;
* Все свободные номера;
* Все забронированные или заселенные номера;
* Номера, которые освободятся сегодня;
* Персональные данные сотрудников (Код сотрудника, ФИО сотрудника, серия и номер паспорта, адрес фактического проживания, дата рождения, должность, время начала рабочего дня, время окончания рабочего дня).

**3.2 Определение сущностей и связей**

**Сущность** - это объект, информация о котором должна быть представлена в базе данных.

**Экземпляр сущности** - это информация о конкретном представителе объекта.

**Связь** - соединение между двумя и более сущностями.

**Экземпляр связи** - конкретная связь между конкретными представителями объектов.

Сущности, представленные в данном курсовом проекте:

* **Клиенты** (содержит информацию о клиентах);
* **Персонал** (содержит информацию о персонале гостиницы);
* **Должности** (содержит информацию о должностях);
* **Номера** (содержит информацию о номерах гостиницы);
* **УчетРаботы** (учет клиентов гостиницы).

**3.3 Определение функций пользователя, атрибутов, ключей**

**Атрибут** - свойство сущности или связи.

**Ключ сущности** - атрибут или набор атрибутов, используемый для однозначной идентификации экземпляра сущности.

**Ключи и атрибуты, в данном курсовом проекте:**

Сущность **Клиенты** содержит следующие атрибуты: КодКлиента – является ключом, Фамилия, Имя, Отчество, Страна, СерияНомерПаспорта, Адрес, ДатаРождения, ЦельПриезда, Телефон.

Сущность **Персонал** содержит следующие атрибуты: КодСотрудника – является ключом, Фамилия, Имя, Отчество, СерияНомерПаспорта, Адрес, ДатаРождения, НачалоРабДня, КонецРабДня, Должность, Телефон, Статус.

Сущность **Должности** содержит следующие атрибуты: КодДолжности – является ключом, Название, Зарплата.

Сущность **Номера** содержит следующие атрибуты: КодНомера – является ключом, Состояние, СуточнаяСтоимость, КоличествоМест, СостояниеОплаты, КодСотрудника.

Сущность **УчетРаботы** содержит следующие атрибуты: КодОперации – является ключом, КодНомера, КодКлиента, ДатаЗаселения, ДатаВыселения, ОбщСтоимостьПроживания.

**3.4 Выявление и описание ограничений целостности**

Под целостностью данных понимаются ссылочные ограничения, т.е. те ограничения, которые нужно соблюдать для сохранения целостности связи между таблицами, в случае если в них будут изменяться или удаляться записи.

Для обеспечения целостности данных в Access есть 4 варианта:

1. Если не указано каскадное обновление связей, то предотвращается изменение значений первичного ключа в главной таблице, если существуют связанные записи в подчиненной таблице.

2. Если указано каскадное обновление связей, то при изменении значений первичного ключа будут изменяться соответствующие значения в связанной таблице.

3. Если не указано каскадное удаление связанных записей, то предотвращается удаление связанных записей из главной таблицы, если имеются связанные с ней записи в подчиненной.

4. Если указано каскадное удаление, то связанные записи подчиненной таблицы удаляются автоматически.

Также к ограничениям целостности можно отнести ограничения на столбец и на таблицу, а точнее на значения данных в них. К таким ограничениям можно отнести следующие:

* Запрещение null значения: данные, заносимые в столбец или таблицу, не должны равняться нулю.
* Ограничения на допустимые значения полей: условие, которому должны удовлетворять данные, вносимые в таблицу.
* Ограничение первичного ключа: на практике рекомендуется для каждой таблицы создавать первичный ключ, особенностью которого является не допуск null значения.
* Ограничение уникальных ключей: необходимость ввода различных (уникальных) данных.

В данном курсовом проекте используются следующие ограничения данных в таблицах:

1)Формат даты

1.Таблица **Клиенты**

В поле ***ДатаРождения*** используется маска ввода 00.00.0000; т. е. краткий формат даты.

2.Таблица **Персонал**

В поле ***ДатаРождения*** используется маска ввода 00.00.0000; т. е. краткий формат даты.

В поле ***НачалоРабДня*** используется маска ввода 00:00; т. е. краткий формат времени.

В поле ***КонецРабДня*** используется маска ввода 00:00; т. е. краткий формат времени.

2)Телефон

1.Таблица **Клиенты**

В поле ***Телефон*** используется маска ввода 0\-000\-000\-00\-00;

2.Таблица **Персонал**

В поле ***Телефон*** используется маска ввода 0\-000\-000\-00\-00.

3)Подстановка

1.Таблица **Персонал**

В поле ***Статус*** используется подстановка значений: работает, уволен; т. е. в это поле можно занести только эти значения.

В поле ***Должность*** используется подстановка значений из таблицы **Должности**.

2.Таблица **Номера**

В поле ***Состояние*** используется подстановка значений: забронирован, заселен, свободен; т. е. в это поле можно занести только эти значения.

В поле ***КодСотрудника*** используется подстановка значений из таблицы **Персонал**.

3.Таблица **УчетРаботы**

В поле ***КодНомера*** используется подстановка значений из таблицы **Номера**;

В поле ***КодКлиента*** используется подстановка значений из таблицы **Клиенты**.

**3.5 Разработка инфологической модели предметной области**

**Инфологическая модель** описывает предметную область на содержательном уровне. Результатом этого анализа являются списки объектов предметной области, перечни свойств, или атрибутов, определение связей между объектами и описание структуры предметной области в виде диаграммы.

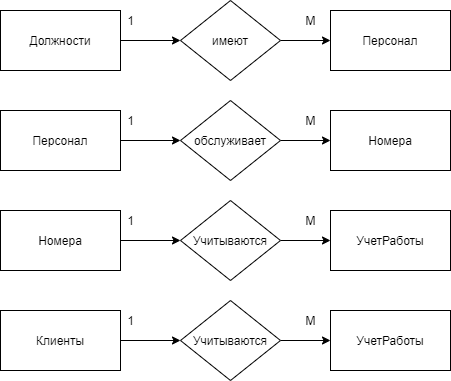
Определим связи данной предметной области на этапе разработки инфологической модели.

Связь между сущностями можно охарактеризовать степенью связи и классом принадлежности сущности к связи. Где степень связи показывает, сколько экземпляров одной сущности могут быть связано с каждым экземпляром другой сущности, и может иметь три значения:

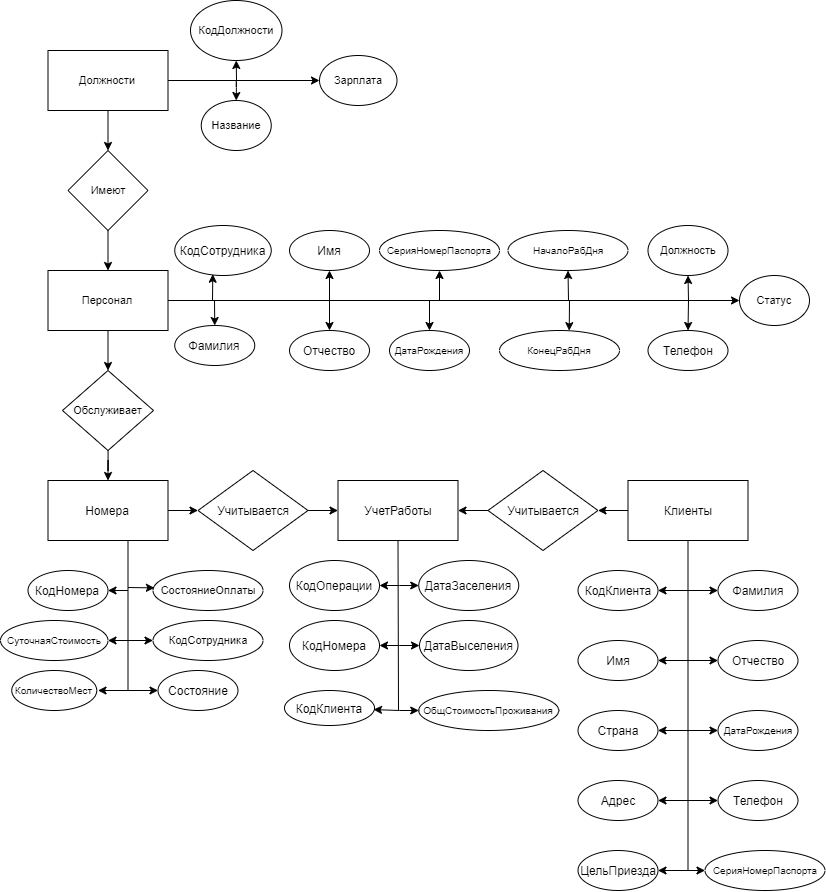
* Один к одному (1:1) – означает. Что каждый экземпляр первой сущности может быть связан только с одним экземпляром второй сущности и наоборот.
* Один ко многим (1:М или М:1) – означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, а каждый экземпляр второй сущности может быть связан только с одним экземпляром первой сущности.
* Многие ко многим (М:N) – означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности и наоборот.

Класс принадлежности сущности к связи может быть обязательным (каждый экземпляр сущности обязательно должен быть связан с другой сущностью) и необязательным (каждый экземпляр сущности не требует связи с экземпляром другой сущности).

**В данном курсовом проекте используются односторонние связи.**



**Расширенная ER-диаграмма.**



**3.6 Доказательство того, что все отношения (таблицы) находятся в 1-ой,2-ой,3-ей нормальных формах.**

* Для того чтобы таблица считалась нормализованной к первой нормальной форме, каждое из ее полей должно быть неделимым (атомарным) и таблица не должна содержать никаких повторяющихся групп полей.
* Для того чтобы привести таблицу ко второй нормальной форме, нужно, чтобы она удовлетворяла первой нормальной форме и, чтобы все не ключевые поля полностью зависели от первичного ключа таблицы и от каждого поля в первичном ключе, если последний состоит из нескольких полей. Это значит, что каждое не ключевое поле должно уникально определяться первичным ключом и полями, его составляющими. Второй нормальной форме удовлетворяют все таблицы.
* Для того чтобы таблица была приведена к третьей нормальной форме, нужно, чтобы она удовлетворяла второй нормальной форме и все не ключевые поля полностью зависели от первичного ключа таблицы и не зависели друг от друга. Таким образом, к квалификации второй нормальной формы добавляется требование независимости каждого не ключевого поля таблицы от других не ключевых полей. Нужно исключить из таблицы также поля, которые можно вычислить по другим не ключевым полям таблицы. В нашем случае третьей нормальной форме удовлетворяют все таблицы.

**4. Проектирование дата логической модели БД**

**4.1 Выбор СУБД.**

СУБД представляет собой совокупность языковых и программных средств, с помощью которых база данных создается и поддерживается. На данный момент существует множество языков, с помощью которых можно создавать различные структуры и вводить в них необходимые элементы управления. При выборе модели данных мы остановились на реляционной модели из-за ее математической определенности и наличия большого количества СУБД, которые поддерживают реляционную модель данных. Из всего множества СУБД была выбрана Microsoft Access2016 плюс ко всему разработка интерфейса проводится на Visual Studio на языке C#, так как я предпочитаю более гибкие инструменты.

**4.2 Отображение инфологической модели на даталогическую модель**

**Даталогическая** модель описывает объекты и связи предметной области на формальном уровне. Ее разработка основывается на инфологической модели. В процессе разработки осуществляется выбор модели данных, и определяются ее элементы.

Учитывая выбранную СУБД и разработанную инфологическую модель предметной области, была разработана следующая даталогическая модель:

-**Клиенты**: КодКлиента, Фамилия, Имя, Отчество, Страна, СерияНомерПаспорта, Адрес, ДатаРождения, ЦельПриезда, Телефон.

-**Персонал**: КодСотрудника, Фамилия, Имя, Отчество, СерияНомерПаспорта, Адрес, ДатаРождения, НачалоРабДня, КонецРабДня, Должность, Телефон, Статус.

-**Должности**: КодДолжности, Название, Зарплата.

-**Номера**: КодНомера, Состояние, СуточнаяСтоимость, КоличествоМест, СостояниеОплаты, КодСотрудника.

**-УчетРаботы**: КодОперации, КодНомера, КодКлиента, ДатаЗаселения, ДатаВыселения, ОбщСтоимостьПроживания.

**Таблица 1 Клиенты**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Обязательное поле** | **Индексированное поле** | **Условие на значение** |
| КодКлиента | Счетчик(int) | да | да |  |
| Фамилия | Текстовый | да | нет |  |
| Имя | Текстовый | да | нет |  |
| Отчество | Текстовый | да | нет |  |
| Страна | Текстовый | да | нет |  |
| СерияНомерПаспорта | Текстовый | да | нет |  |
| Адрес | Текстовый | да | нет |  |
| ДатаРождения | Дата и время | да | нет |  |
| ЦельПриезда | Текстовый | нет | нет |  |
| Телефон | Текстовый | да | нет |  |

**Таблица 2 Персонал**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Обязательное поле** | **Индексированное поле** | **Условие на значение** |
| КодСотрудника | Счетчик(int) | да | да |  |
| Фамилия | Текстовый | да | нет |  |
| Имя | Текстовый | да | нет |  |
| Отчество | Текстовый | да | нет |  |
| СерияНомерПаспорта | Текстовый | да | нет |  |
| Адрес | Текстовый | да | нет |  |
| ДатаРождения | Дата и время | да | нет |  |
| НачалоРабДня | Дата и время | да | нет |  |
| КонецРабДня | Дата и время | да | нет |  |
| Должность | Числовой(баг) | да | нет |  |
| Телефон | Текстовый | да | нет |  |
| Статус | Текстовый | да | нет |  |

**Таблица 3 Дожности**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Обязательное поле** | **Индексированное поле** | **Условие на значение** |
| КодДолжности | Числовой | да | да |  |
| Название | Текстовый | да | нет |  |
| Зарплата | Денежный | да | нет |  |

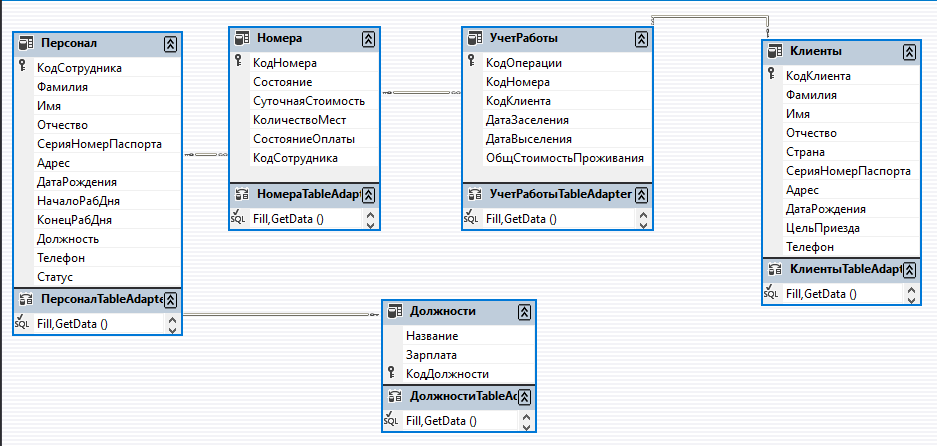
**Таблица 4 Номера**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Обязательное поле** | **Индексированное поле** | **Условие на значение** |
| КодНомера | Числовой | да | да |  |
| Состояние | Текстовый | да | нет |  |
| СуточнаяСтоимость | Денежный | да | нет |  |
| КоличествоМест | Числовой | да | нет |  |
| СостояниеОплаты | Логический | нет | нет |  |
| КодСотрудника | Числовой | да | нет |  |

**Таблица 5 УчетРаботы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Обязательное поле** | **Индексированное поле** | **Условие на значение** |
| КодОперации | Счетчик | да | да |  |
| КодНомера | Числовой | да | нет |  |
| КодКлиента | Числовой | да | да |  |
| ДатаЗаселения | Дата и время | да | нет |  |
| ДатаВыселения | Дата и время | да | нет |  |
| ОбщСтоимостьПроживания | Денежный | нет | нет |  |

**4.3 Схема дата логической модели базы данных (схема данных)**



**5. Реализация БД**

**5.1 Разработка средств реализации ограничений целостности**

В любой момент времени БД имеет некоторую определенную конфигурацию значений данных, которые отражают действительность, т.е. являются частью реального мира. Просто определить конфигурацию значений не имеет смысла без связи с внешним миром. Поэтому требуется уточнить определение БД, включив в него правила целостности, которые необходимы для информирования СУБД о различного рода ограничениях реального мира с целью не допустить “ абсурдных ” значений данных.

Для любого отношения можно создать ряд правил - ограничений. Каждая конкретная БД должна иметь свои ограничения, связанные с предметной областью, которые накладываются на хранящиеся в ней данные. К таким ограничениям целостности относятся:

1. Ограничения на атрибуты (тип атрибута, диапазон допустимых значений).

2. Число кортежей отношения должно быть равно числу первичных ключей (наличие кортежей –дубликатов не допускается).

Первое ограничение накладывается на атрибуты всех отношений на этапе определения типа атрибута.

Второе ограничение накладывается на отношения на этапе заполнения таблиц данными о БД.

Существует также два общих правила целостности. Они касаются потенциальных и внешних ключей:

1. Первичный ключ является уникальным идентификатором отношения. Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе, принимал неопределенное значение. В отношении не может быть несколько кортежей с одинаковыми значениями первичного ключа.

Потенциальный ключ отношения не может иметь пустого значения (NULL). Так как объект, не имеющий идентичности, не существует.

2. Если r2 – некоторое отношение с внешним ключом X , то должно существовать такое базовое отношение r1 с первичным ключом K , что каждое значение X в r2 совпадает со значением К в каком-либо кортеже отношения r1.

В процессе создания БД сначала осуществляется конструирование таблиц, далее создается схема данных, в которой фиксируются связи между таблицами. В этой схеме могут быть заданы параметры обеспечения целостности базы данных, если модель была разработана в соответствии с требованиями нормализации. Целостность данных означает, что в БД установлены и корректно поддерживаются взаимосвязи между записями разных таблиц при их загрузке, добавлении и удалении в связанных таблицах, а также при изменении значений ключевых полей.

**5.2 Разработка процедур ведения БД (Запросы)**

Для получения нужной информации необходимо объединять исходные отношения. Связывание исходных отношений осуществляется при помощи запросов, создаваемых мастером и дорабатываемых вручную до получения требуемого результата. Запросы описаны при помощи встроенного языка запросов SQL.

В данном проекте созданы следующие запросы:

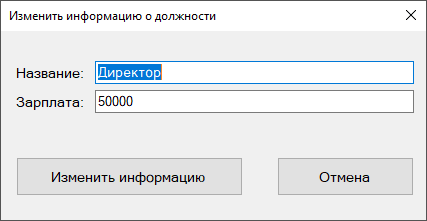
**Запрос на выборку по коду должности**

SELECT Название, Зарплата, КодДолжности

FROM Должности

WHERE КодДолжности = ?

Позволяет заполнять форму изменения информации о должности



**Запрос на обновление информации о должности**

UPDATE Должности

SET Название = ?, Зарплата = ?

WHERE (КодДолжности = ?)

**Запрос на присвоение статуса «уволен» сотруднику**

UPDATE Персонал

SET Статус = 'Уволен'

WHERE (КодСотрудника = ?)

**Запрос на отображение списка уволенных сотрудников(нужен скрин)**

SELECT КодСотрудника, Фамилия, Имя, Отчество, СерияНомерПаспорта, Адрес, ДатаРождения, НачалоРабДня, КонецРабДня, Должность, Телефон, Статус

FROM Персонал

WHERE Статус = 'Уволен'

**Запрос на отображение списка работающих сотрудников**

SELECT КодСотрудника, Фамилия, Имя, Отчество, СерияНомерПаспорта, Адрес, ДатаРождения, НачалоРабДня, КонецРабДня, Должность, Телефон, Статус

FROM Персонал

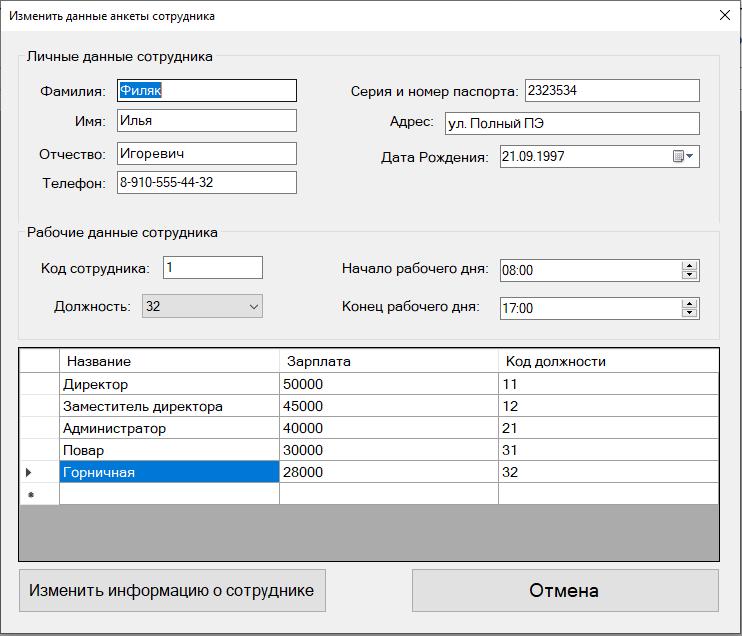
WHERE Статус = 'Работает'

**Запрос на отображение информации о конкретном сотруднике**

SELECT КодСотрудника, Фамилия, Имя, Отчество, СерияНомерПаспорта, Адрес, ДатаРождения, НачалоРабДня, КонецРабДня, Должность, Телефон, Статус

FROM Персонал

WHERE КодСотрудника = ?



**Запрос на изменение информации о сотруднике**

UPDATE Персонал

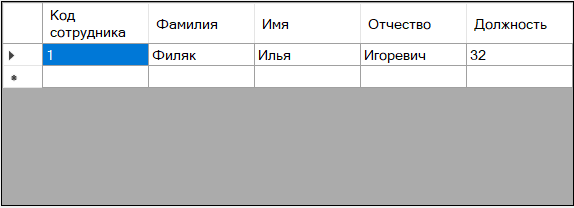
SET КодСотрудника = ?, Фамилия = ?, Имя = ?, Отчество = ?, СерияНомерПаспорта = ?, Адрес = ?, ДатаРождения = ?, НачалоРабДня = ?, КонецРабДня = ?, Должность = ?, Телефон = ?, Статус = ?

WHERE (КодСотрудника = ?)

**Запрос на формирование сокращенной таблицы «Персонал»**

SELECT КодСотрудника, Фамилия, Имя, Отчество, Должность

FROM Персонал



**Запрос на удаление комнаты из таблицы «Номера»**

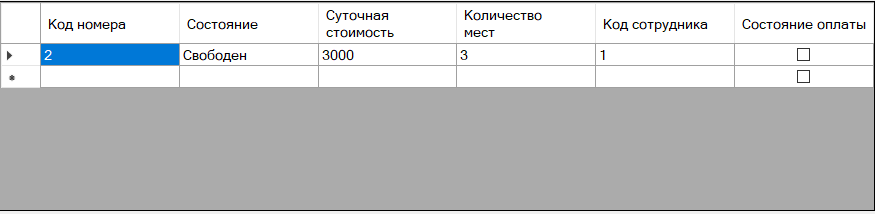
DELETE FROM Номера

WHERE (КодНомера = ?)

**Запрос на отображение только свободных номеров**

SELECT КодНомера, Состояние, СуточнаяСтоимость, КоличествоМест, СостояниеОплаты, КодСотрудника FROM Номера

WHERE Состояние = 'Свободен'

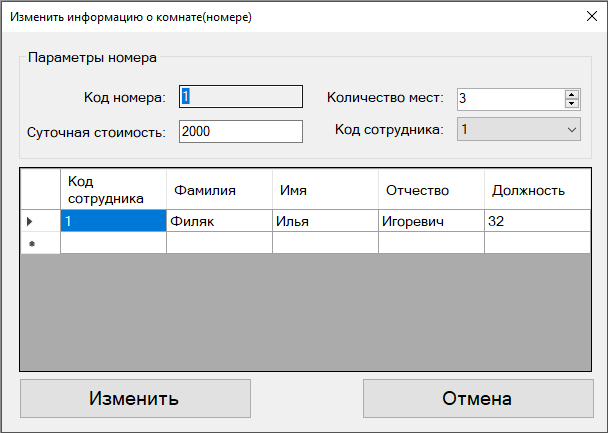


**Запрос на отображение информации о конкретном номере**

SELECT КодНомера, Состояние, СуточнаяСтоимость, КоличествоМест, СостояниеОплаты, КодСотрудника

FROM Номера

WHERE КодНомера = ?



**Запрос на изменения информации о комнате**

UPDATE Номера

SET Состояние = ?, СуточнаяСтоимость = ?, КоличествоМест = ?, СостояниеОплаты = ?, КодСотрудника = ?

WHERE (КодНомера = ?)

Запрос на изменение состояния номера(свободен, заселен, забронирован)

UPDATE Номера

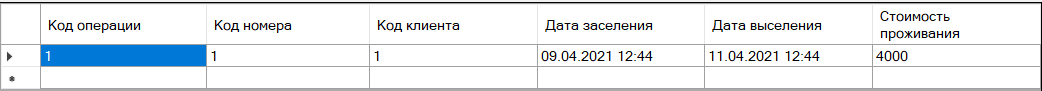
SET Состояние = ?

WHERE (КодНомера = ?)

**Запрос на отображение операций за текущий месяц**

SELECT КодОперации, КодНомера, КодКлиента, ДатаЗаселения, ДатаВыселения, ОбщСтоимостьПроживания FROM УчетРаботы

WHERE YEAR(ДатаЗаселения) = YEAR(Now()) AND MONTH(ДатаЗаселения) = MONTH(Now())



**Запрос на изменение информации о операции по коду клиента**

UPDATE УчетРаботы

SET КодНомера = ?, ДатаЗаселения = ?, ДатаВыселения = ?, ОбщСтоимостьПроживания = ?

WHERE КодКлиента = ?

**Запрос на изменение информации о клиенте**

UPDATE Клиенты

SET Фамилия = ?, Имя = ?, Отчество = ?, Страна = ?, СерияНомерПаспорта = ?, Адрес = ?, ДатаРождения = ?, ЦельПриезда = ?, Телефон = ?

WHERE (КодКлиента = ?)

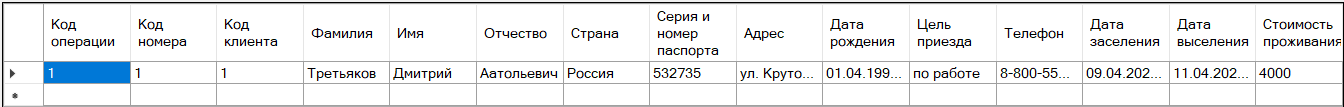
**Составной запрос на отображение полной информации о учете работы**

SELECT УчетРаботы.КодОперации, УчетРаботы.КодНомера, УчетРаботы.КодКлиента, Клиенты.Фамилия, Клиенты.Имя, Клиенты.Отчество, Клиенты.Страна, Клиенты.СерияНомерПаспорта, Клиенты.Адрес,

Клиенты.ДатаРождения, Клиенты.ЦельПриезда, Клиенты.Телефон, УчетРаботы.ДатаЗаселения, УчетРаботы.ДатаВыселения, УчетРаботы.ОбщСтоимостьПроживания

FROM (Клиенты INNER JOIN

УчетРаботы ON Клиенты.КодКлиента = УчетРаботы.КодКлиента)



**Составной запрос на отображение информации о конкретной операции**

SELECT УчетРаботы.КодОперации, УчетРаботы.КодНомера, УчетРаботы.КодКлиента, Клиенты.Фамилия, Клиенты.Имя, Клиенты.Отчество, Клиенты.Страна,

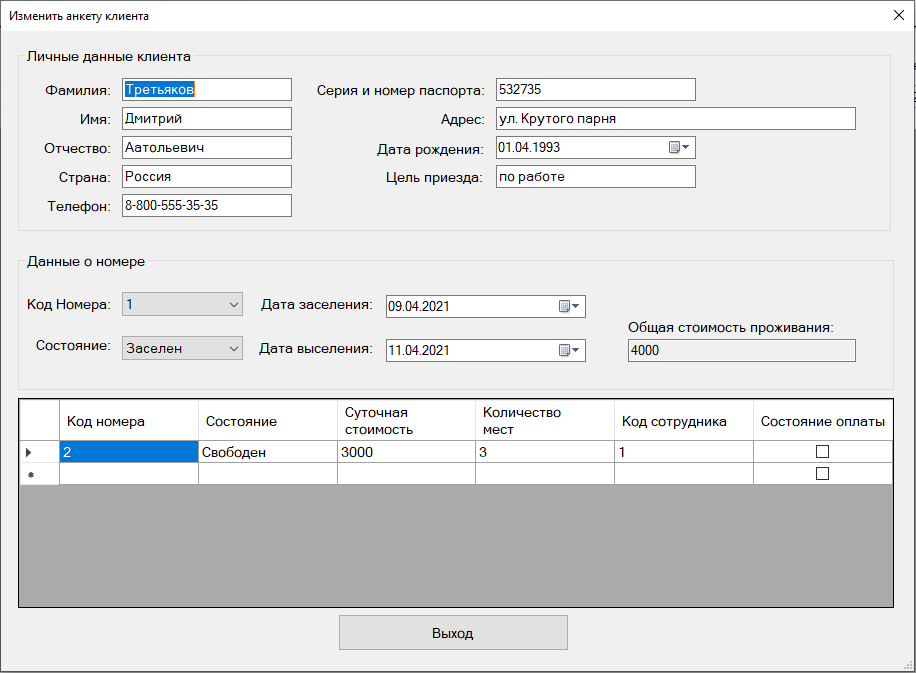
Клиенты.СерияНомерПаспорта, Клиенты.Адрес, Клиенты.ДатаРождения, Клиенты.ЦельПриезда, Клиенты.Телефон, УчетРаботы.ДатаЗаселения,

УчетРаботы.ДатаВыселения, УчетРаботы.ОбщСтоимостьПроживания

FROM (Клиенты INNER JOIN

УчетРаботы ON Клиенты.КодКлиента = УчетРаботы.КодКлиента)

WHERE (Клиенты.КодКлиента = ?)



**Составной запрос на поиск клиента по ФИО**

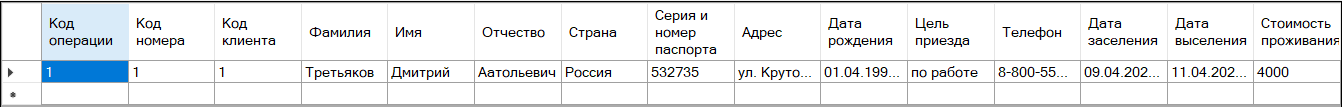
SELECT УчетРаботы.КодОперации, УчетРаботы.КодНомера, УчетРаботы.КодКлиента, Клиенты.Фамилия, Клиенты.Имя, Клиенты.Отчество, Клиенты.Страна, Клиенты.СерияНомерПаспорта, Клиенты.Адрес, Клиенты.ДатаРождения,

Клиенты.ЦельПриезда, Клиенты.Телефон, УчетРаботы.ДатаЗаселения, УчетРаботы.ДатаВыселения, УчетРаботы.ОбщСтоимостьПроживания

FROM (Клиенты INNER JOIN

УчетРаботы ON Клиенты.КодКлиента = УчетРаботы.КодКлиента)

WHERE Клиенты.Фамилия = ? AND Клиенты.Имя = ? AND Клиенты.Отчество = ?



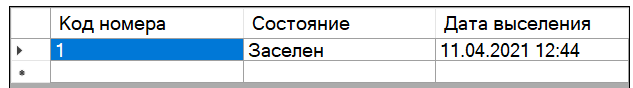
**Запрос на отображение только заселенных номеров**

SELECT УчетРаботы.КодНомера, Номера.Состояние, УчетРаботы.ДатаВыселения

FROM (Номера INNER JOIN

УчетРаботы ON Номера.КодНомера = УчетРаботы.КодНомера)

WHERE (Номера.Состояние = 'Заселен')



**Составной запрос на показ информации о зарплате каждого сотрудника**

SELECT Персонал.КодСотрудника, Персонал.Должность, Должности.Зарплата

FROM (Должности INNER JOIN

Персонал ON Должности.КодДолжности = Персонал.Должность)

(Нигде не отображается)

**6. Интерфейс и руководство пользователя**

Для запуска данного проекта достаточно сделать двойной щелчок мышью по файлу “Hotel.exe”. Открывается главная кнопочная форма (Рис.1). Из нее можно попасть, нажимая соответствующие кнопки, в формы просмотра и работы с БД.

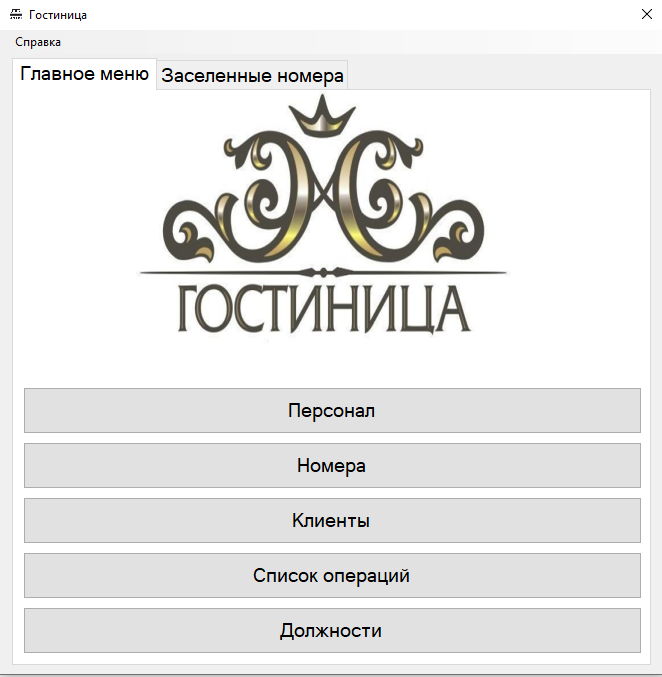


Рис. 1. Иллюстрация «Главной формы»

При нажатии на кнопку «Должности» открывается форма со списком должностей – рисунок 2.

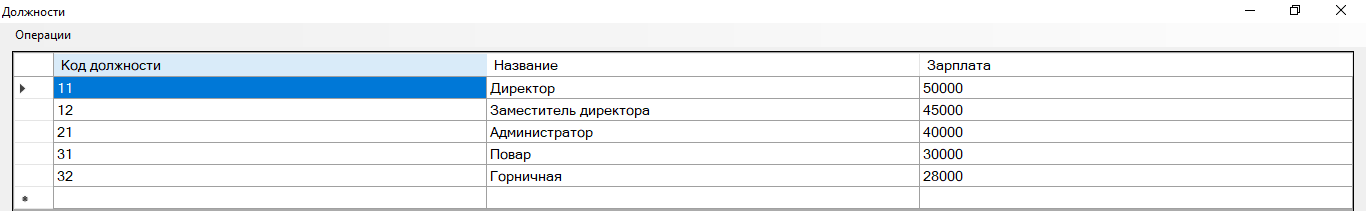


Рис. 2. Форма «Должности»

При нажатии на кнопку «Операции» на форме «Должности» открывается список операций, состоящий из одного элемента – «Изменить информацию о должности». При нажатии на эту кнопку открывается форма «Изменить информацию о должности», где мы можем изменить название и зарплату на выбранной должности - рисунок 3.

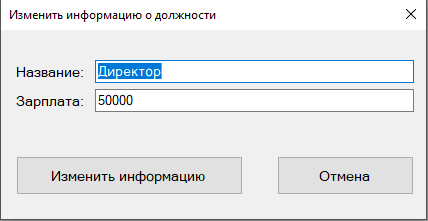


Рис. 3. Форма «Изменить информацию о должности»

При нажатии на кнопку «Персонал» на главной форме, открывается форма со списком сотрудников – рисунок 4.

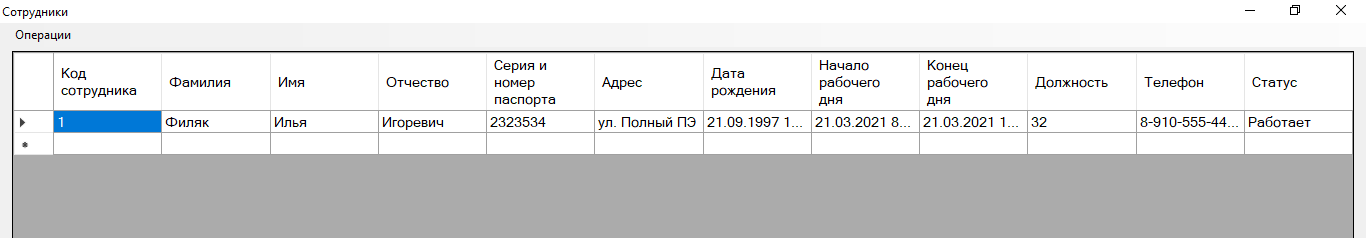


Рис. 4. Форма «Персонал»

При нажатии на кнопку «Операции» на форме «Персонал» открывается список операций – рисунок 5.

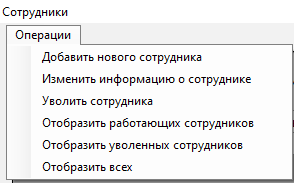


Рис. 5. Список доступных операций.

При нажатии на кнопку «Добавить нового сотрудника» откроется форма «Анкета сотрудника» - рисунок 6.

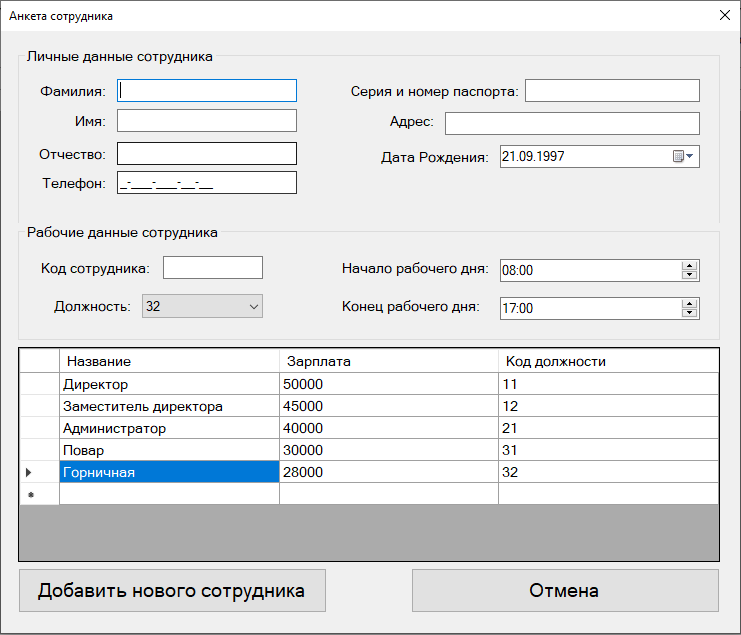


Рис. 6. Форма «Анкета сотрудника»

При нажатии на кнопку «Изменить информацию о сотруднике» откроется форма «Изменить данные анкеты сотрудника» - рисунок 7.

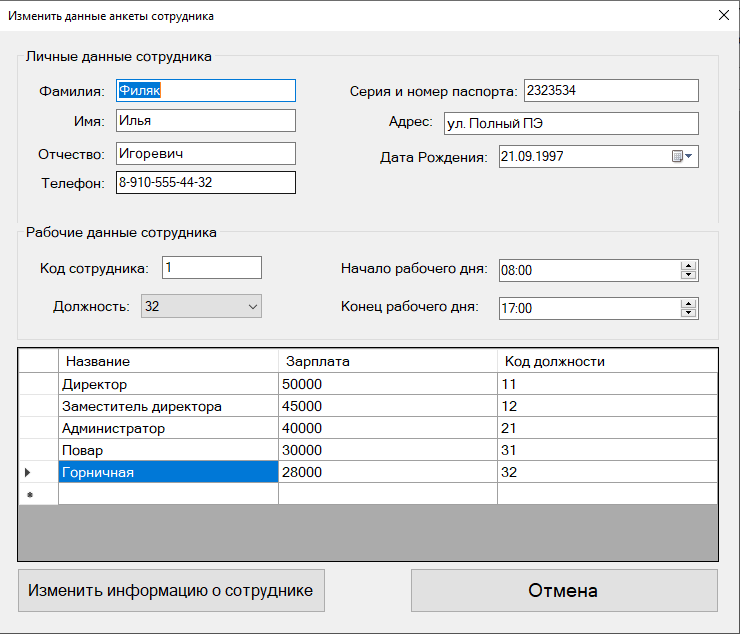


Рис. 7. Форма «Изменить данные анкеты сотрудника».

При нажатии на кнопку «Уволить сотрудника» мы изменяем статус выбранного сотрудника на «Уволен».

При нажатии на кнопку «Отобразить работающих сотрудников» на списке формы «Персонал» отображаются только работающие сотрудники - рисунок 8.

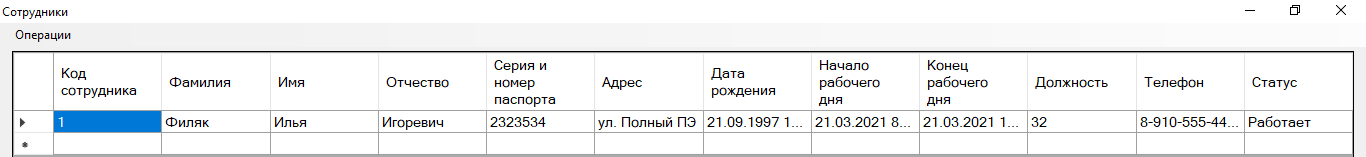


Рис. 8. Список работающих сотрудников.

При нажатии на кнопку «Отобразить уволенных сотрудников» на списке формы «Персонал» отображаются только уволенные сотрудники – рисунок 9.

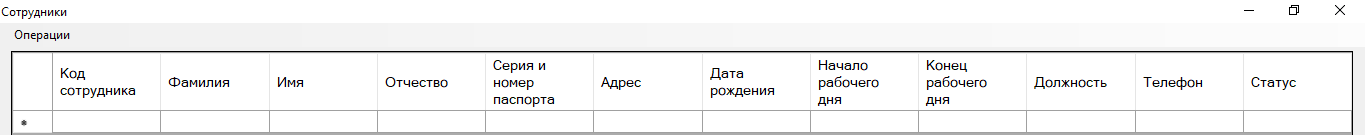


Рис. 9. Список уволенных сотрудников.

При нажатии на кнопку «Отобразить всех» на списке формы «Персонал» отображаются все сотрудники(даже уволенные) – рисунок 10.

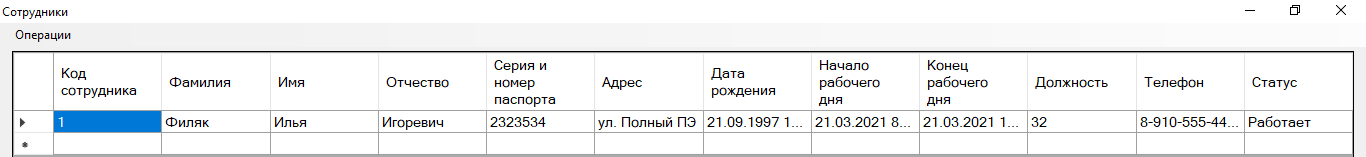


Рис. 10. Список всех сотрудников.

При нажатии на кнопку «Номера» на главной форме, открывается форма со списком комнат в гостинице – рисунок 11.

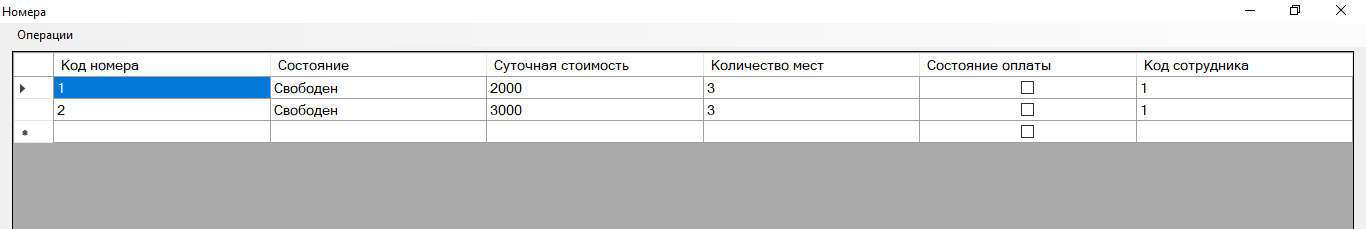


Рис. 11. Форма «Номера».

При нажатии на кнопку «Операции» на форме «Номера» открывается список операций – рисунок 12.

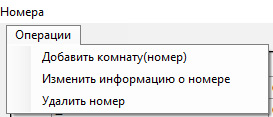


Рис. 12. Список операций на форме «Номера».

При нажатии на кнопку «Добавить комнату(номер)», откроется форма «Добавить комнату(номер) – рисунок 13. (Внимание невозможно добавить комнату, если список сотрудников пуст)

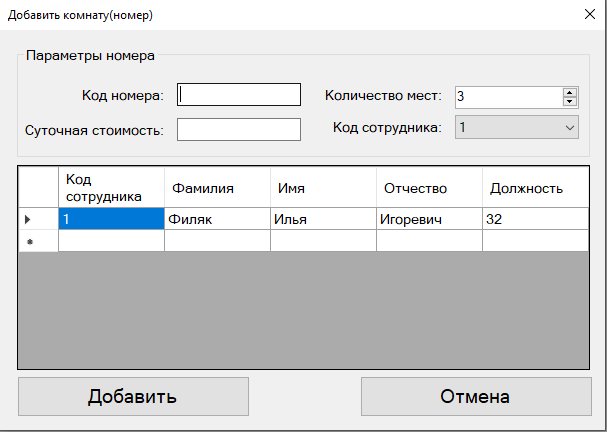


Рис. 13. Форма «Добавить комнату(номер)»

При нажатии на кнопку «Изменить информацию о номере», откроется форма «Изменить информацию о комнате(номере)» – рисунок 14. (Внимание невозможно добавить комнату, если список сотрудников пуст)

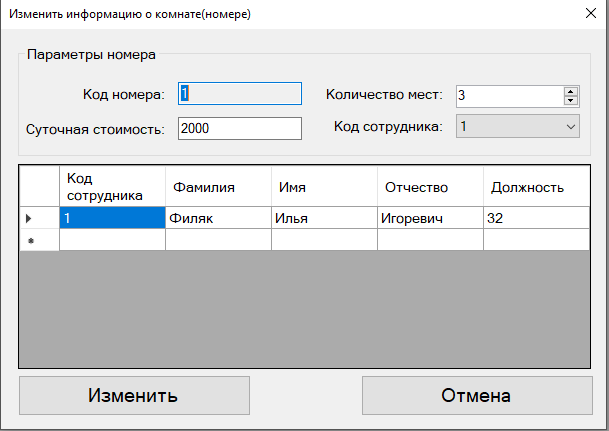


Рис 14. Форма «Изменить информацию о комнате(номере)»

При нажатии на кнопку «Удалить номер», выбранный номер из списка будет удален(Внимание, невозможно удалить номер, который уже фигурировал в таблице «Учет работы») – рисунок 15.

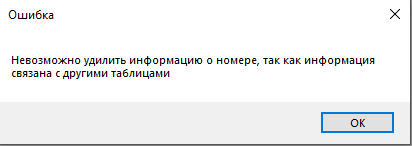


Рис. 15. Сообщение об ошибке.

При нажатии «Клиенты» на главной форме, открывается форма со списком клиентов и операций – рисунок 16.

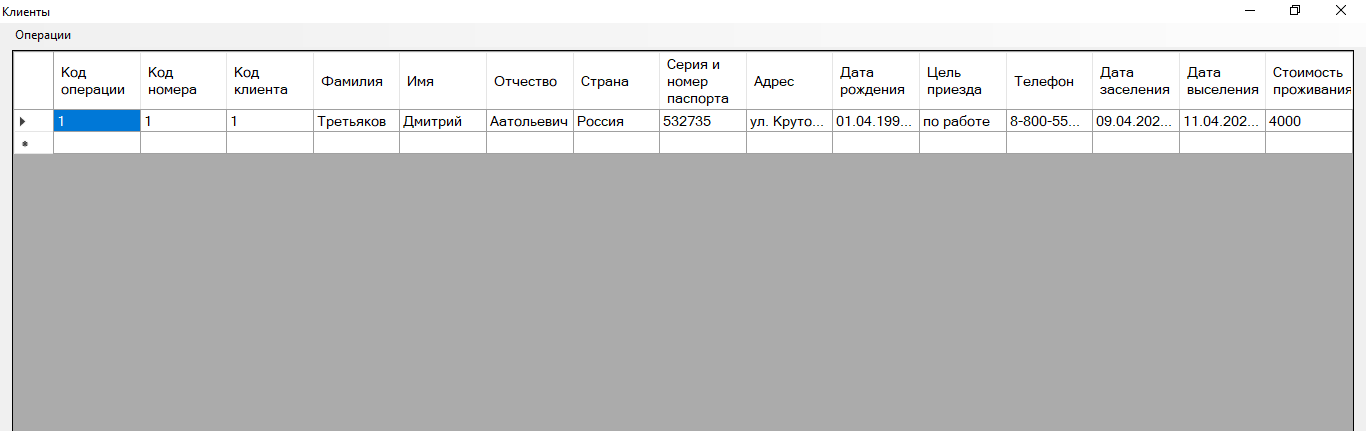


Рис. 16. Форма «Клиенты»

При нажатии на кнопку «Операции» на форме «Клиенты» открывается список операций – рисунок 17.

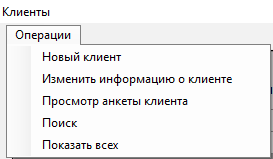


Рис. 17. Список операций на форме «Клиенты».

При нажатии на кнопку «Новый клиент», открывается форма «Анкета клиента» - рисунок 18.

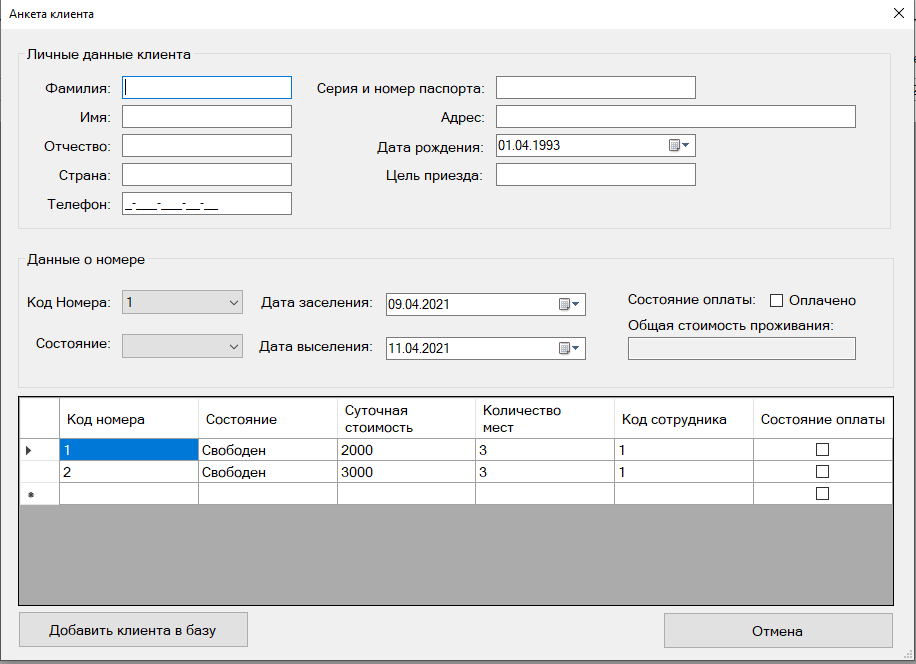


Рис. 18. Форма «Анкета клиента»

При нажатии на кнопку «Изменить информацию о клиенте», открывается форма «Изменить анкету клиента» - рисунок 19.

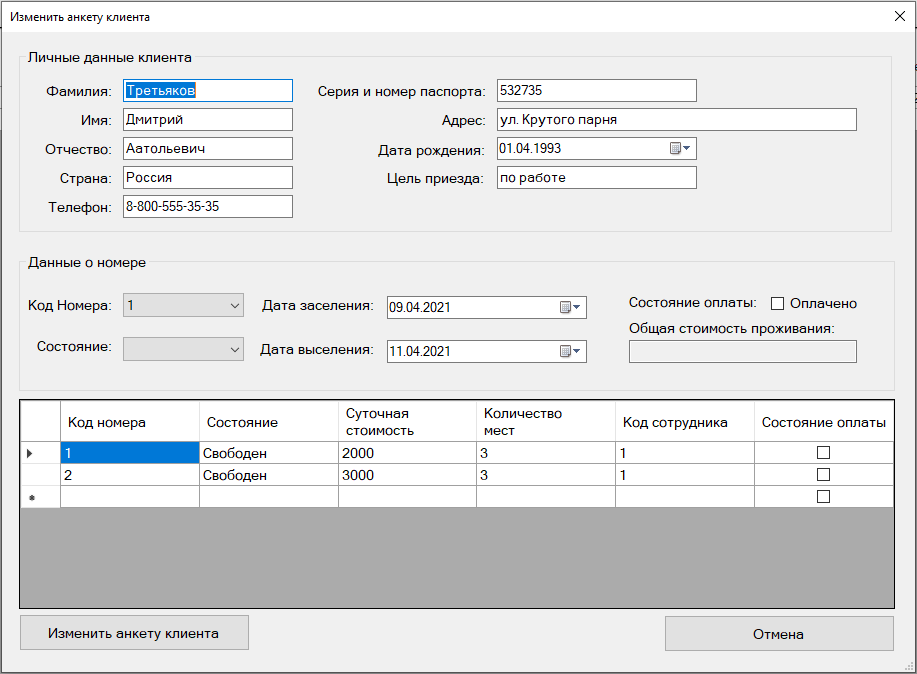


Рис. 19. Форма «Изменить анкету клиента».

При нажатии на кнопку «Просмотр анкеты клиента», открывается форма «Изменить анкету клиента» - рисунок 20.(Нужно изменить имя формы)

При нажатии на кнопку «Поиск», открывается форма «Поиск» - рисунок 21.

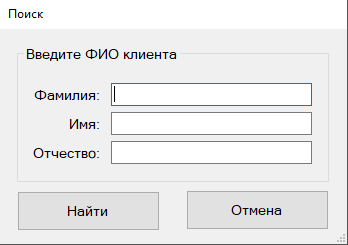


Рис 21. Форма «Поиск»

При нажатии на кнопку «Показать всех», показывается полный список всех клиентов гостиницы – рисунок 22.

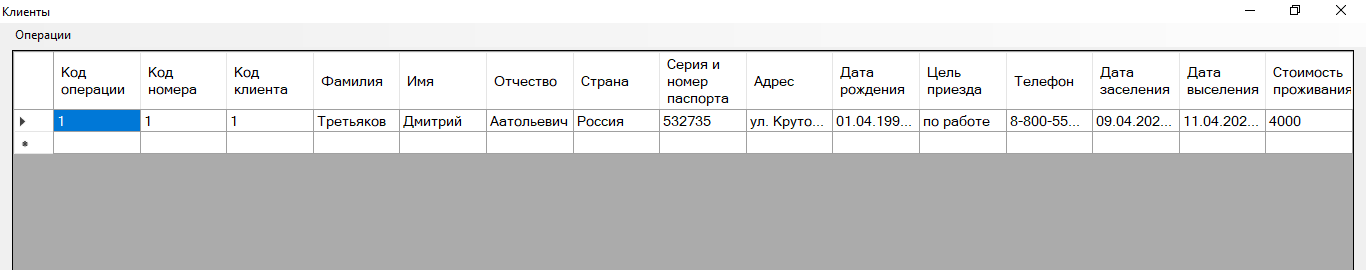


Рис. 22. Показ всех клиентов.

При нажатии на кнопку «Список операций» на главной форме, открывается форма «Учет работы»– рисунок 23.

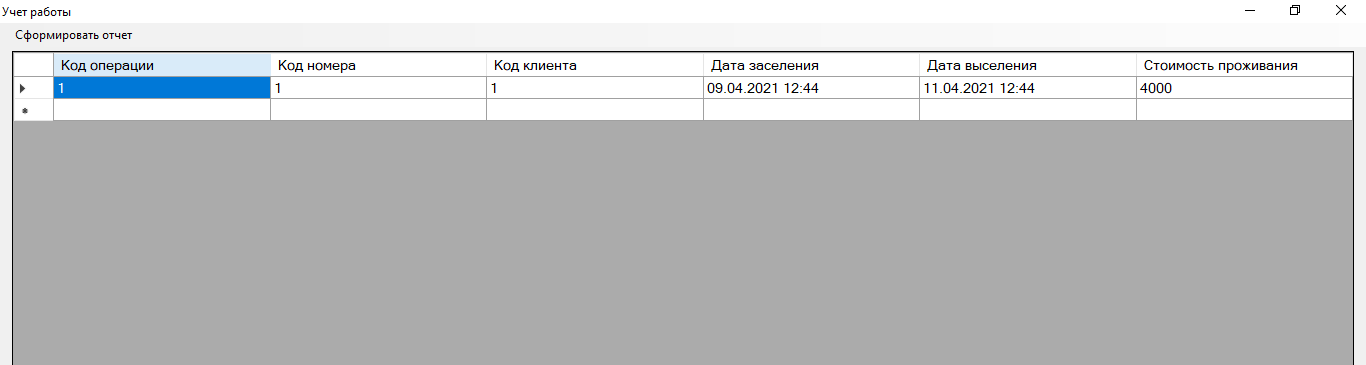


Рис. 23. Форма «Учет работы»

При нажатии на кнопку «Сформировать отчет» откроется форма «Отчет за текущий месяц» - рисунок 24.

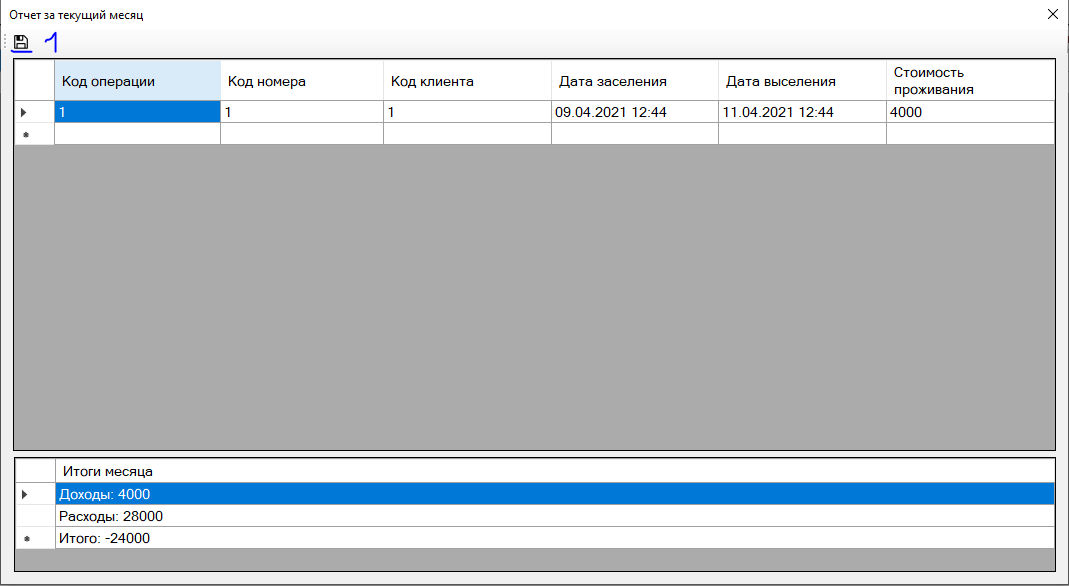


Рис. 24. Форма «Отчет за текущий месяц»

При нажатии кнопки (1), открывается диалоговое окно с выбором имени файла отчета и адреса его расположения. Отчет сохраняется в формат «xlsx».

При нажатии на кнопку «Справка», «О программе» на главной форме, открывается форма «О программе»– рисунок 25.

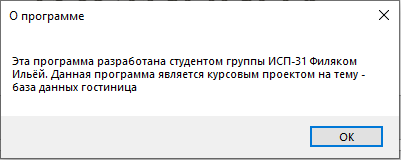


Рис. 25. Форма «О программе».

При нажатии на кнопку «Заселенные номера», открывается список заселенных номеров – рисунок 26.

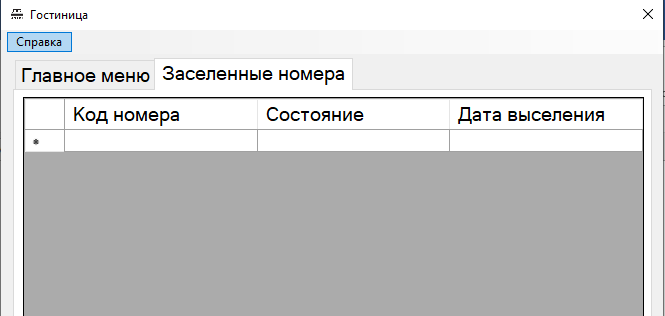


Рис 26. Заселенные номера.

**7. Спецификация входных и выходных запросов**

В данной программе содержаться следующие модули:

Модуль: Accounting.cs;

Модуль: AddRoom.cs;

Модуль: ChangeClientProfile.cs;

Модуль: ChangeInfJobPosition.cs;

Модуль: ChangeRoom;

Модуль: ChangeWorkerProfile.cs;

Модуль: ClientProfile.cs;

Модуль: Clients.cs;

Модуль: JobPositions.cs;

Модуль: MainForm.cs;

Модуль: Report.cs;

Модуль: Rooms.cs;

Модуль: Search.cs;

Модуль: ShowClientProfile.cs;

Модуль: WorkerProfile.cs;

Модуль: Workers.cs.

Для ввода и вывода информации, хранящейся в таблицах базы данных, используются формы и отчеты, которые находятся в соответствующих разделах форм. Формы используют запросы для спецификации отражаемых данных.

Форма: Главная форма;

Форма: Должности;

Форма: Изменить информацию о должности;

Форма: Сотрудники;

Форма: Анкета сотрудника;

Форма: Изменить данные анкеты сотрудника;

Форма: Номера;

Форма: Добавить комнату(номер);

Форма: Изменить информацию о комнате(номере);

Форма: Клиенты;

Форма: Анкета клиента;

Форма: Изменить анкету клиента;

Форма: Поиск;

Форма: Просмотр анкеты клиента;

Форма: Учет работы;

Форма: Отчет за текущий месяц.

**8. Заключение**

В данном курсовом проекте была создана база данных по предметной области Гостиница. Спроектированная база данных «Hotel» дает возможность удобного ввода, редактирования, удаления и хранения данных. В ней полностью реализованы входные и выходные формы. Это позволяет облегчить работу при сборе и накоплении необходимых сведений о сотрудниках и клиентах. Так же автоматизировано создание статистических отчётов.

Данный курсовой проект может применяться как самостоятельная программа в небольших гостиницах (хотя нуждается в доработке).

**9. Список использованных источников**

ГОСТ 19.106-78 - ТРЕБОВАНИЯ К ПРОГРАММНЫМ ДОКУМЕНТАМ, ВЫПОЛНЕННЫМ ПЕЧАТНЫМ СПОСОБОМ.