**ИНСТИТУТ ТРАНСПОРТА И СВЯЗИ**



ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕЛЕКОММУНИКАЦИЙ

КАФЕДРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ

**КУРСОВАЯ РАБОТА**

по дисциплине

„Базы и Банки Данных”

Тема: «РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ ЭКСКУРСИОННОГО БЮРО»

Выполнил: Фролов Георгий

Студ. код: 70905, Группа: 4801BD

Проверил:

Рига

2020

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc57891103)

[ВВЕДЕНИЕ 3](#_Toc57891104)

[1. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ 4](#_Toc57891105)

[2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 4](#_Toc57891106)

[2.1. Анализ предметной области 4](#_Toc57891107)

[2.2. Определение требований и функций системы 4](#_Toc57891108)

[2.3. Разработка модели базы данных 5](#_Toc57891109)

[3. РЕАЛИЗАЦИЯ БД В СУБД 9](#_Toc57891110)

[3.1. Создание таблиц и схемы базы данных 9](#_Toc57891111)

[3.2. Разработка и создание форм для заполнения таблиц базы данных 10](#_Toc57891112)

[3.3. Разработка и создание запросов 14](#_Toc57891113)

[3.4. Разработка и создание отчётов 15](#_Toc57891114)

[3.5. Разработка и создание приложения пользователя 17](#_Toc57891115)

[4. ВЫВОДЫ 22](#_Toc57891116)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 23](#_Toc57891117)

# ВВЕДЕНИЕ

Данная курсовая работа написана в рамках дисциплины «Базы и Банки Данных» и посвящена разработке базы данных для хранения информации в экскурсионном бюро. В работе подробно описаны все этапы разработки. Первый раздел посвящен проектированию Базы Данных: анализу предметной области, определению требований и функций системы, разработке модели БД. Второй раздел посвящён реализации БД: созданию таблиц и схемы базы данных, разработке и созданию форм для заполнения таблиц, разработке и созданию запросов, отчётов и приложения пользователя.

База данных создана в Системе Управления Баз Данных *PostgreSQL.* Программа написана в среде программирования *Visual Studio* на языке *C#.*

## ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Для предметной области «Экскурсионное Бюро», спроектировать и реализовать базу данных в СУБД *Microsoft Access*, которая включает не менее 5 сущностей, имеет удобный пользовательский интерфейс, обеспечивающий ввод и корректировку данных в таблицы БД, получение с помощью запросов необходимой информации и вывод документов на печать в виде отчетов.

## ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

### Анализ предметной области

База данных создаётся для организации экскурсий и подсчёта прибыли для экскурсионного бюро, и для хранения информации о текущих и прошедших экскурсиях. Базой данных будут пользоваться работники экскурсионного бюро. При покупке или заказе билета на экскурсию в базу будет заноситься персональная информация о клиенте (имя, фамилия и др.). Билет должен оформляться только на одного клиента. Клиенты связаны с конкретными экскурсиями через купленный билет. Экскурсии связаны с определённым типом экскурсии (цена билетов, длительность, максимальное допустимое кол-во клиентов), который в свою очередь связан со списком достопримечательностей, которые будут посещены во время экскурсии данного типа. Тип экскурсии – это шаблон, по которому организуются конкретные экскурсии. С экскурсиями также связаны экскурсоводы, которые проводят экскурсии. У каждой экскурсии могут быть один или более экскурсоводов.

### Определение требований и функций системы

Система должна обеспечивать удобный и корректный ввод информации об:

* Экскурсанте: имя, фамилия, статус(VIP-лицо / пенсионер или др.);
* Экскурсоводе: имя, фамилия, персональный код, дата рождения, дата принятия на работу;
* Экскурсии: дата и время начала, статус(запланирована, пройдена, отменена);
* Типе Экскурсии: продолжительность, цена участия, макс. число экскурсантов;
* Достопримечательности: название, адрес, оценка качества(популярности).

Корректный ввод информации будет обеспечиваться ограничениями при вводе персонального кода, численных полей, даты и времени.

Система должна давать возможность регистрации экскурсантов на конкретные экскурсии, а также должна позволять организовывать новые экскурсии. При регистрации экскурсантов на экскурсии нужно учесть максимально-возможное количество экскурсантов на экскурсию.

Система на запрос должна обеспечивать:

* Получение информации об экскурсии;
* Получение списка экскурсий, назначенных в определённый интервал времени;
* Подсчёт прибыли с экскурсии (или экскурсий за указанный период времени);
* Получение списка экскурсантов и экскурсоводов на конкретной экскурсии;
* Назначать экскурсоводов на экскурсии;
* Регистрировать клиентов на экскурсии;
* Выводить список экскурсий без назначенных экскурсоводов;
* Добавление информации в таблицы экскурсий, экскурсоводов, клиентов и другие таблицы.

Система в качестве выходных документов должна:

* Распечатывать план экскурсий на конкретный период времени с назначенными на них экскурсоводами;
* Распечатывать список экскурсантов на конкретной экскурсии;
* Распечатывать прибыль с экскурсий за указанный период времени.

### Разработка модели базы данных

На основе анализа предметной области разрабатывается реляционная модель базы данных. Логическая модель базы данных будет представлена в виде графической модели «сущность-связь», в которой используются основные элементы: сущность, атрибут и связь.

На основе анализа предметной области были определены следующие основные сущности: экскурсант (Client), экскурсия (Tour), тип экскурсии (Tour Type), достопримечательность (Site), экскурсовод (Guide). Была построена модель «сущность-связь» для предметной области «Экскурсионное Бюро».

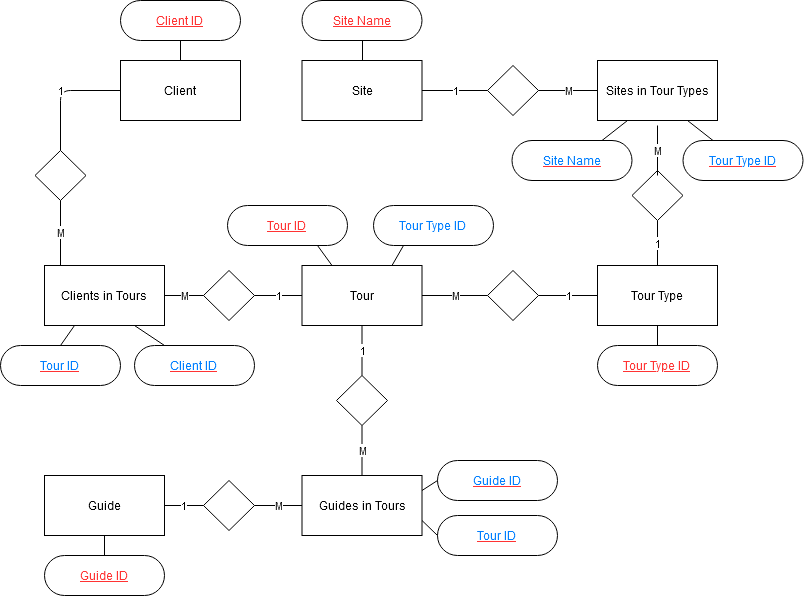


Рис.2.1. **Модель базы данных для предметной области «Экскурсионное Бюро»**

Типы связей между сущностями получены, исходя из действующих бизнес-правил в данной предметной области.

**Экскурсант-Экскурсия (М:N)**

1 экскурсант может посетить **М** экскурсий;

1 экскурсию могут посетить **N** экскурсантов.

**Экскурсовод-Экскурсия (M:N)**

1 экскурсовод может проводить **М** экскурсий;

1 экскурсию могут проводить **N** экскурсоводов.

**Тип Экскурсии-Экскурсия (1:M)**

1 типу экскурсии принадлежит **М** экскурсий;

1 экскурсия принадлежит 1 типу экскурсии.

**Достопримечательность-Тип Экскурсии (M:N)**

1 достопримечательность посещается в **М** типах экскурсии;

В 1 типе экскурсии посещается **N** достопримечательностей.

·

Таблица 2.1

**Сущности и атрибуты**

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Свойства атрибута | | |
| Имя атрибута | Тип атрибута | Тип данных |
| Client | Client ID | Ключ первичный | Число целочисленное |
| CName | Описательный | Текст |
| CSurname | Описательный | Текст |
| CStatus | Описательный | Текст |
| ID Code | Описательный | Текст |
| Tour | Tour ID | Ключ первичный | Число целочисленное |
| Tour Type ID | Ключ внешний | Текст |
| TStatus | Описательный | Текст |
| Start Datetime | Описательный | Дата/Время |
| Guide | Guide ID | Ключ первичный | Число целочисленное |
| GName | Описательный | Текст |
| GSurname | Описательный | Текст |
| GStatus | Описательный | Текст |
| Birth Date | Описательный | Дата |
| Employment Date | Описательный | Дата |
| ID Code | Описательный | Текст |
| Clients in Tours | Client ID | Ключ первичный /внешний | Число целочисленное |
| Tour ID | Ключ первичный /внешний | Число целочисленное |
| Guides in Tours | Guide ID | Ключ первичный /внешний | Число целочисленное |
| Tour ID | Ключ первичный /внешний | Число целочисленное |
| Tour Type | Tour Type ID | Ключ первичный | Текст |
| Price | Описательный | Число вещественное |
| Max Participants | Описательный | Число целочисленное |
| Duration | Описательный | Время |
| Tour Type Name | Описательный | Текст |
| Site | SName | Ключ первичный | Текст |
| Address | Описательный | Адрес |
| Quality Mark | Описательный | Число целочисленное |
| Sites in Tour Types | SName | Ключ первичный /внешний | Текст |
| Tour Type Name | Ключ первичный /внешний | Текст |

## РЕАЛИЗАЦИЯ БД В СУБД

### Создание таблиц и схемы базы данных

На основе таблицы 2.1, где представлены все сущности и их атрибуты создаются таблицы базы данных. Ниже показаны SQL-запросы, с помощью которых будут созданы структуры таблиц:

Таблица 3.1. **SQL запросы, описывающие структуры таблиц.**

|  |  |
| --- | --- |
| Таблица | Запрос |
| Clients | CREATE TABLE clients(  id INTEGER,  name VARCHAR(64) NOT NULL,  surname VARCHAR(64) NOT NULL,  idcode VARCHAR(16) NOT NULL,  status VARCHAR(48) NOT NULL DEFAULT 'Common',  CONSTRAINT pk\_id PRIMARY KEY(id)); |
| Guides | CREATE TABLE guides(  id INTEGER PRIMARY KEY,  name VARCHAR(64) NOT NULL,  surname VARCHAR(64) NOT NULL,  idcode VARCHAR(16) NOT NULL,  available BOOLEAN DEFAULT true,  employ\_date DATE DEFAULT CURRENT\_DATE,  birth\_date DATE); |
| Sites | CREATE TABLE sites(  name VARCHAR(64) PRIMARY KEY,  address VARCHAR(96) NOT NULL,  quality INTEGER CHECK (quality >= 0 AND quality <= 10)); |
| Tour Types | CREATE TABLE ttypes(  id INTEGER PRIMARY KEY,  name VARCHAR(64) NOT NULL,  price INTEGER NOT NULL CHECK (price >= 0),  max\_participants INTEGER NOT NULL CHECK (max\_participants > 0),  duration BIGINT NOT NULL CHECK (duration > 0)); |
| Tours | CREATE TABLE tours(  id INTEGER PRIMARY KEY,  ttype\_id INTEGER,  start\_datetime TIMESTAMP,  CONSTRAINT fk\_ttype FOREIGN KEY(ttype\_id) REFERENCES ttypes(id)); |
| Tour-Client | CREATE TABLE tour\_client(  tour\_id INTEGER NOT NULL,  client\_id INTEGER NOT NULL,  PRIMARY KEY(tour\_id, client\_id),  CONSTRAINT fk\_tour\_id FOREIGN KEY(tour\_id) REFERENCES tours(id),  CONSTRAINT fk\_client\_id FOREIGN KEY(client\_id) REFERENCES clients(id)); |
| Tour-Guide | CREATE TABLE tour\_guide(  tour\_id INTEGER NOT NULL,  guide\_id INTEGER NOT NULL,  PRIMARY KEY(tour\_id, guide\_id),  CONSTRAINT fk\_tour\_id FOREIGN KEY(tour\_id) REFERENCES tours(id),  CONSTRAINT fk\_guide\_id FOREIGN KEY(guide\_id) REFERENCES guides(id)); |
| TourType-Site | CREATE TABLE ttype\_site(  ttype\_id INTEGER NOT NULL,  site\_name VARCHAR(64) NOT NULL,  PRIMARY KEY(ttype\_id, site\_name),  CONSTRAINT fk\_ttype\_id FOREIGN KEY(ttype\_id) REFERENCES ttypes(id),  CONSTRAINT fk\_site\_name FOREIGN KEY(site\_name) REFERENCES sites(name)); |

PostgreSQL не поддерживает вывод отношений таблиц, поэтому связи между таблицами представлены с помощью среды MS Access:

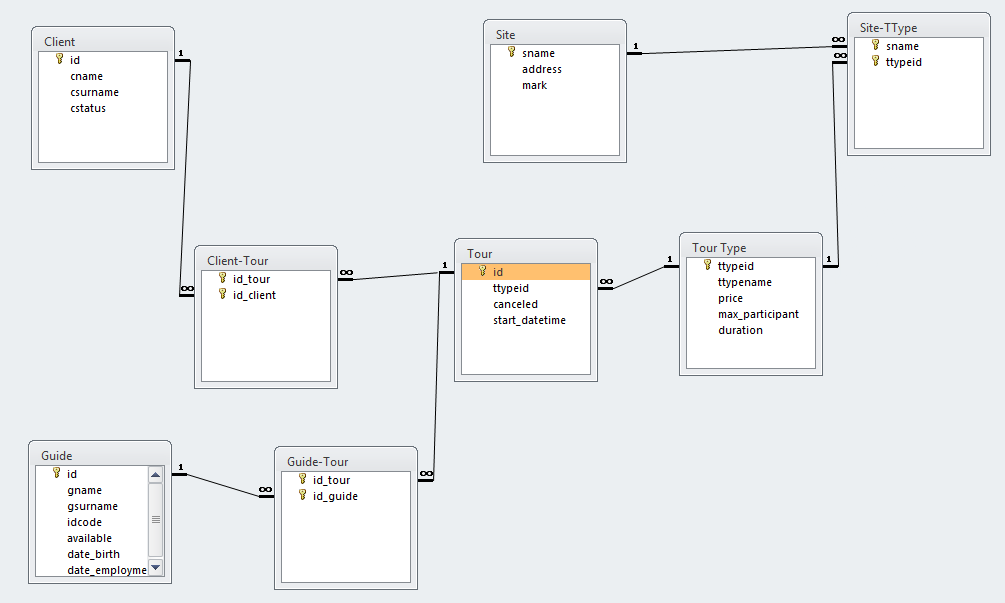
****

Рис. 3.1. **Схема базы данных.**

### Разработка и создание форм для заполнения таблиц базы данных

Для заполнения 5 основных таблиц формы будут созданы в среде *Visual Studio*. Для таблиц Clients, Guides, Tours будут созданы отдельные формы. Таблицы TourTypes и Sites будут иметь "общую" таблицу, т.е. их формы будут доступны из одного окна.

Таблица 3.3. **Формы заполнения таблиц.**

|  |  |  |  |
| --- | --- | --- | --- |
| Форма | Форма включает | | |
| Таблицы | Поля | Свойства |
| Заполнение таблицы "Clients" | Clients | ID  Name  Surname  IDCode  Status | Высчитывается автоматически  Маска персонального кода  ComboBox |
| Заполнение таблицы "Guides" | Guides | ID  Name  Surname  IDCode  Available  Birth Date  Employment Date | Высчитывается автоматически  Маска персонального кода  CheckBox, Авт. "Yes"  Эл-т DatePicker  Эл-т DatePicker, автоматически выбирается текущий день |
| Заполнение таблиц "Sites", "TourTypes", "TourType-Site" | Sites  TourTypes  TourType-Site | Все  ID  TourTypeName  Price  Max Participants  Duration(hours)  Все | Высчитывается автоматически  >= 0  >= 0  >= 0  Для нового типа экскурсии выбираются из таблицы "Sites" достоприм., которые будут связаны с новым типом экскурсии. |
| Заполнение таблицы "Tours" | Tours | ID  StartDateTime  TourType ID | Высчитывается автоматически  Эл-т DatePicker, позже текущего дня.  Выбирается из таблицы "TourTypes" |
| Заполнение таблицы "Tour-Client" | Tour-Client | Все | Из таблицы "Tours" выбирается экскурсия, к которой нужно присвоить экскурсантов, выбранных в таблице "Clients" |
| Заполнение таблицы "Tour-Guide" | Tour-Guide | Все | Из таблицы "Guides" выбирается экскурсовод, которого нужно назначить на экскурсии, выбранные в таблице "Tours" |

Ниже приведены формы, реализованные на языке *C#* с помощью графического интерфейса *WPF* в среде *Visual Studio:*

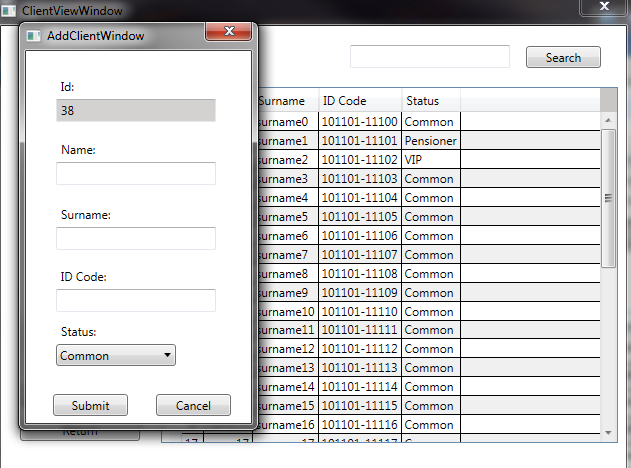


Рис. 3.2. **Форма нового клиента.**

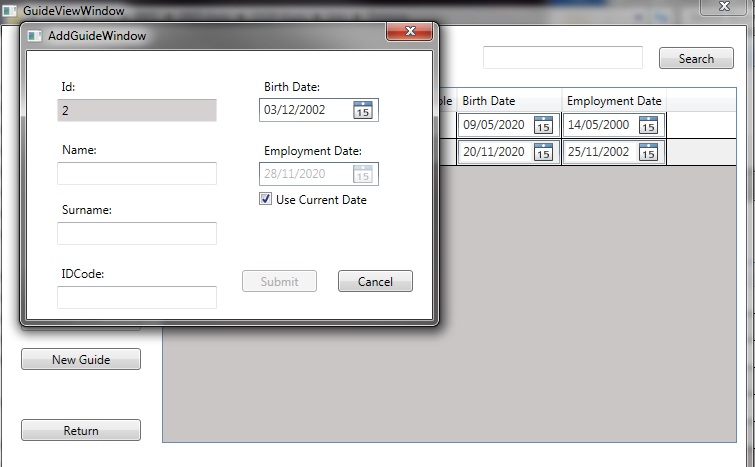


Рис. 3.3. **Форма нового экскурсовода.**

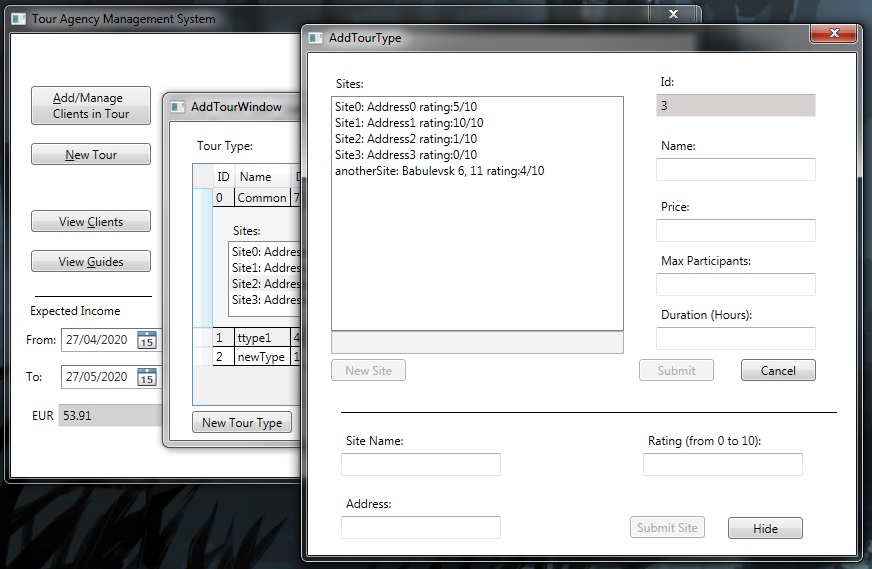


Рис. 3.4. **Форма новой достопримечательности, нового типа экскурсии.**

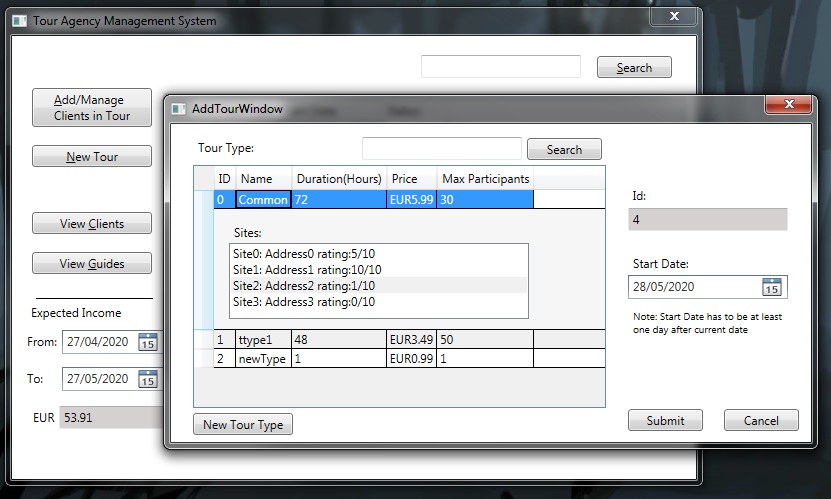


Рис. 3.5. **Форма новой экскурсии.**

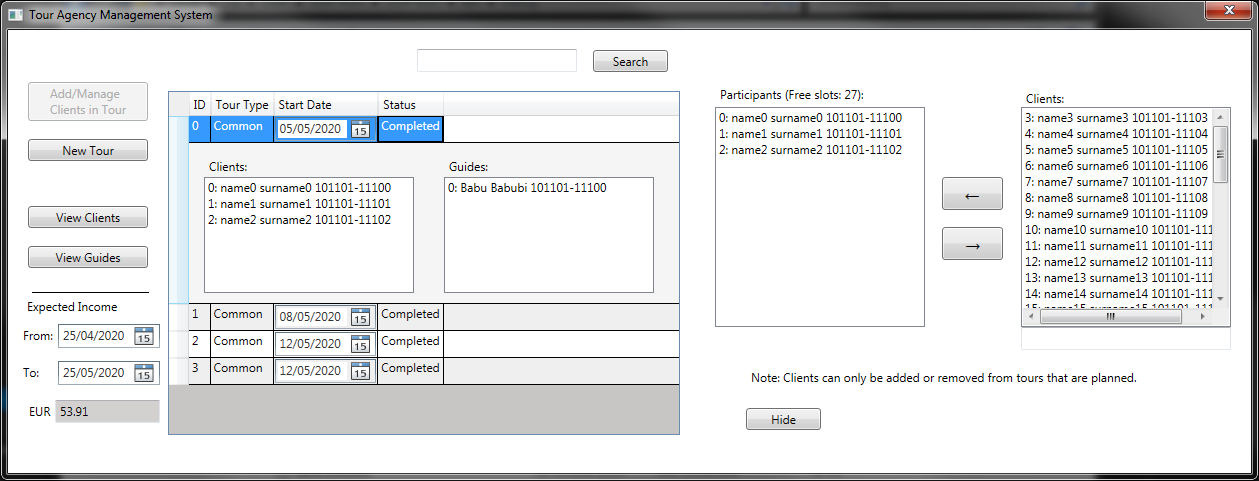


Рис. 3.6. **Форма добавления экскурсантов на экскурсию.**

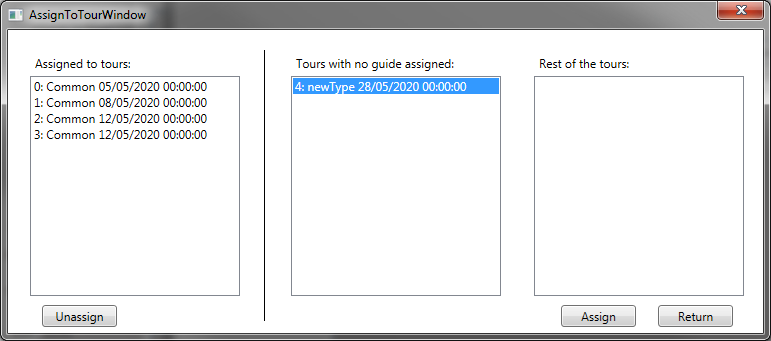


Рис. 3.7. **Форма назначения экскурсовода на экскурсии.**

### Разработка и создание запросов

Согласно требованиям, которые были определены на этапе проектирования разработаны следующие запросы:

Таблица 3.3. **Запросы к базе данных.**

|  |  |  |
| --- | --- | --- |
| Имя запроса | Назначение | Тип запроса |
| GetTourList | Получение полного списка экскурсий и назначенных на них экскурсоводов и клиентов | Выборка |
| GetClientList\_id | Получение списка клиентов на конкретной экскурсии | Параметрический, выборка |
| GetGuideList\_id | Получение списка экскурсоводов на конкретной экскурсии | Параметрический, выборка |
| GetTourType\_Id | Получение записи о типе экскурсии с конкретным ID. | Параметрический, выборка |
| GetTourList\_FromTo | Получение списка экскурсий, назначенных в диапазон даты и времени | Параметрический, выборка |
| GetRevenue | Расчёт прибыли за экскурсии, назначенные в диапазоне даты и времени. | Параметрический, с вычисляемым полем |
| GetRevenueByTourType | Расчёт прибыли за экскурсии по каждому типу экскурсии, назначенные в диапазоне даты и времени. | Параметрический, с вычисляемым полем |
| GetTourListWithoutGuide | Получение списка экскурсий без назначенных экскурсоводов | Выборка |

### Разработка и создание отчётов

На этапе проектирования были определены выходные документы, которые необходимо реализовать в виде отчётов. Структуры разрабатываемых отчётов представлены в таблице.

Таблица 3.4. **Структуры отчётов.**

|  |  |  |  |
| --- | --- | --- | --- |
| Название отчёта | Назначение | Источники отчета | |
| Таблица/Запрос | Поля из таблицы/запроса |
| RevenueReport | Формирует отчёт о прибыли с экскурсий на заданном интервале времени. | GetRevenue GetTourList\_FromTo GetRevenueByTourType | Все  Все  Все |

Примеры готовых отчётов изображены на следующих рисунках:

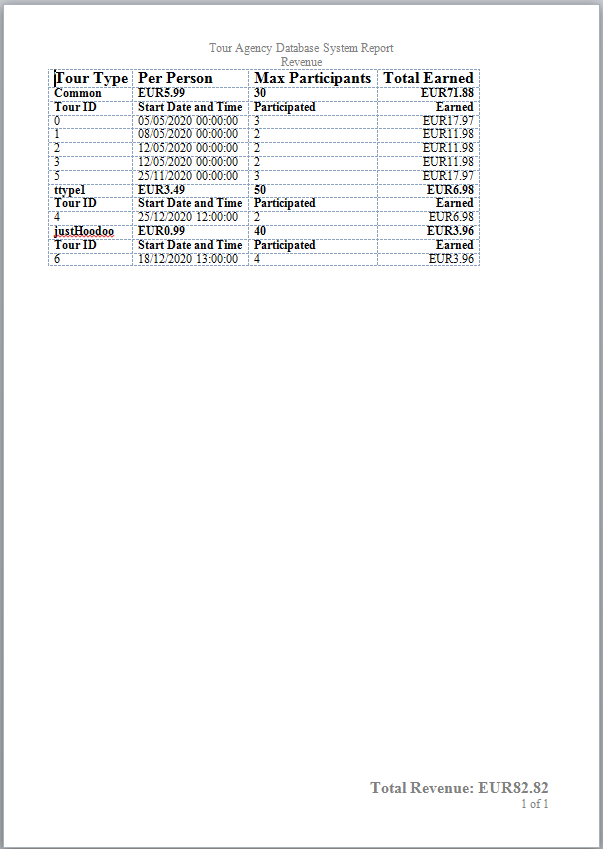
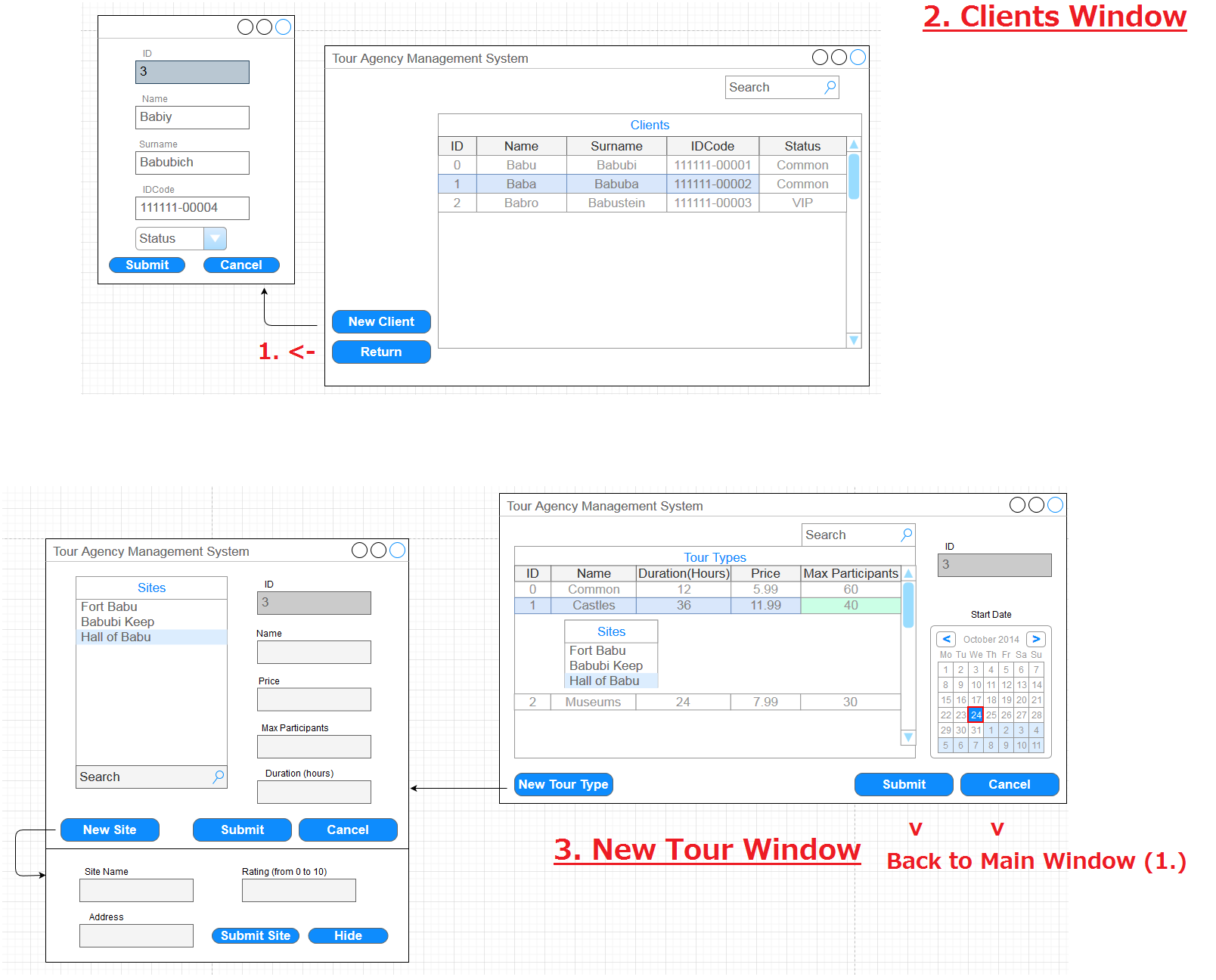


Рис. 3.8. **Отчёт по прибыли (Revenue).**

### Разработка и создание приложения пользователя

Приложение пользователя будет разработано на языке *C#* с помощью графического интерфейса *WPF* в среде разработки *Visual Studio*. Ниже приведены макеты окон программы, связи между ними:





В соответствии с разработанной схемой был создан интерфейс пользователя. Вид реализованного интерфейса представлен на рисунках ниже.

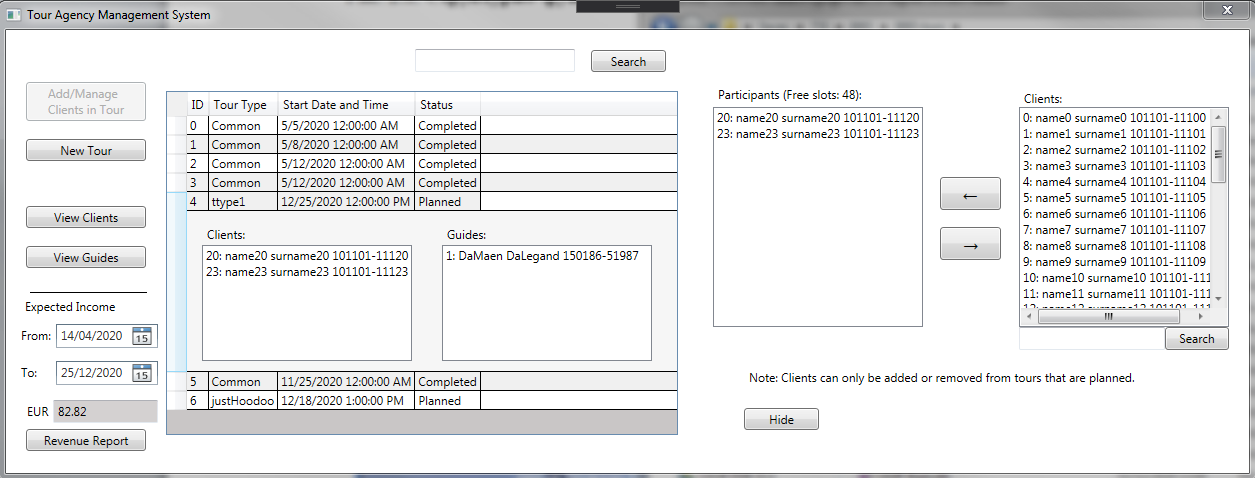
****

Рис. 3.9. **Главное окно, список экскурсий, назначение клиентов на экскурсии.**

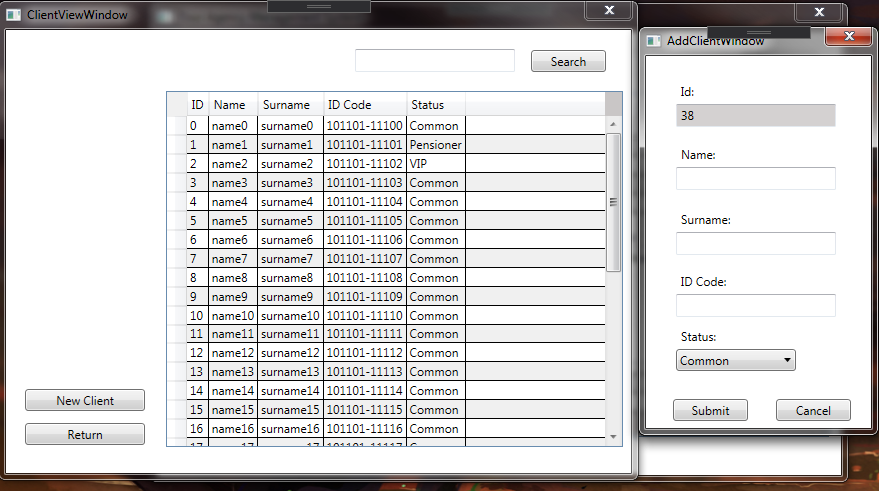
****

Рис. 3.10. **Список клиентов, форма нового клиента.**

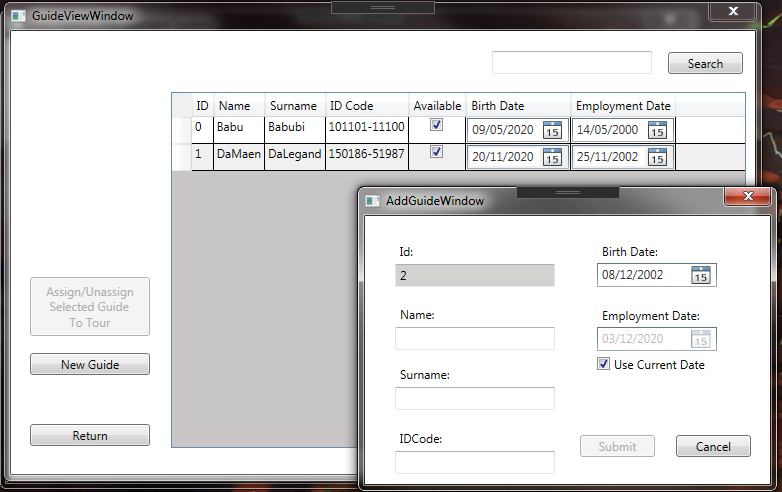
****

Рис. 3.11. **Список экскурсоводов, форма нового экскурсовода.**

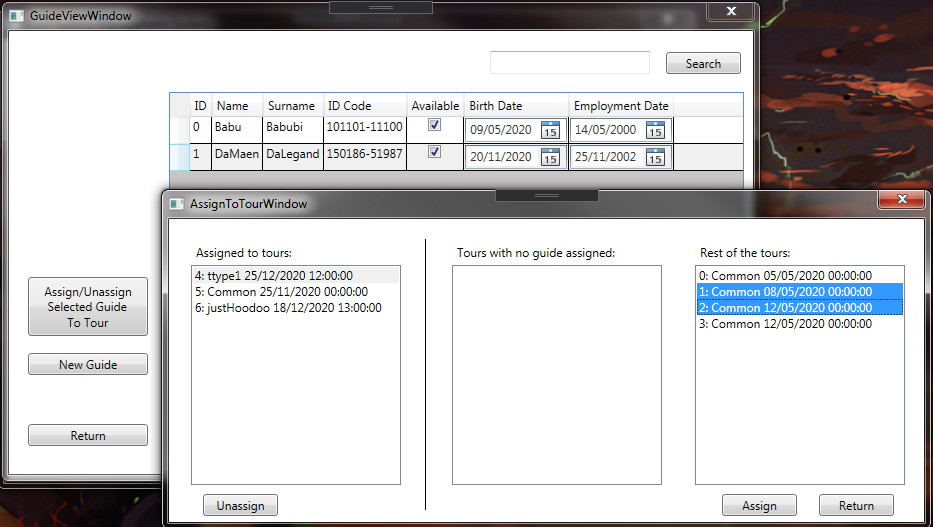
****

Рис. 3.12. **Окно назначения экскурсовода на экскурсии.**

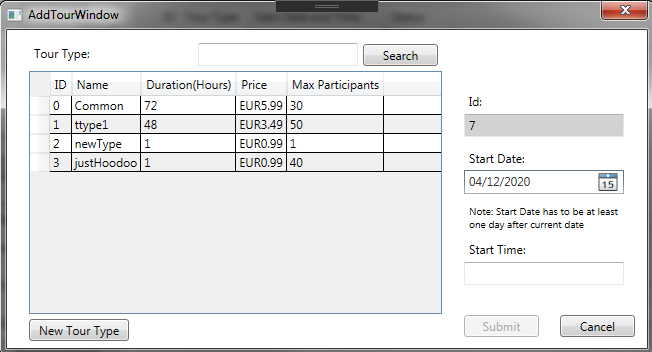
****

Рис. 3.13. **Форма новой экскурсии, список типов экскурсий.**

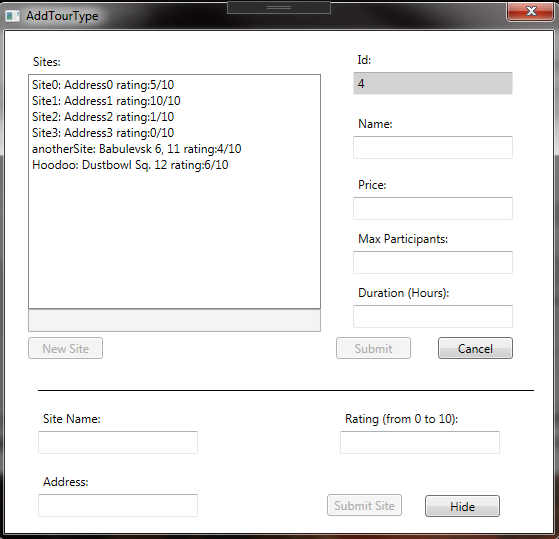
****

Рис. 3.14. **Форма нового типа экскурсии и форма новой достопримечательности, список достопримечательностей.**

## ВЫВОДЫ

В ходе выполнения курсовой работы по дисциплине "Базы и Банки Данных", была реализована база данных для экскурсионного бюро. Был проведён анализ предметной области: выявлены требуемые сущности и функции программы. Была построена модель "сущность-связь". На основе модели были созданы таблицы базы данных в СУБД *PostgreSQL*. Были созданы запросы для обращения к базе данных. Был создан отчёт, который выводит прибыль с экскурсий за заданный интервал времени. На языке программирования *C#* были реализованы формы заполнения таблиц, их отображение. Было создано приложение пользователя.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

* Документация Npgsql: https://www.npgsql.org/doc/index.html;
* Документация PostgreSQL: https://www.postgresql.org/docs/13/index.html;