МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

Отчёт по обязательному заданию

**по дисциплине «Разработка приложений в интегрированных средах»**

Работу выполнил  
студент 35/2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.В. Фомин

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Проверил  
доцент кафедры, канд. техн. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Полупанов

Краснодар 2024

**Цель работы:** разработать клиентское приложение на языке С# для работы с базой данных формата PostgreSQL. Приложение должно иметь возможность добавлять, изменять, удалять данные из БД.

**Практическая часть.**

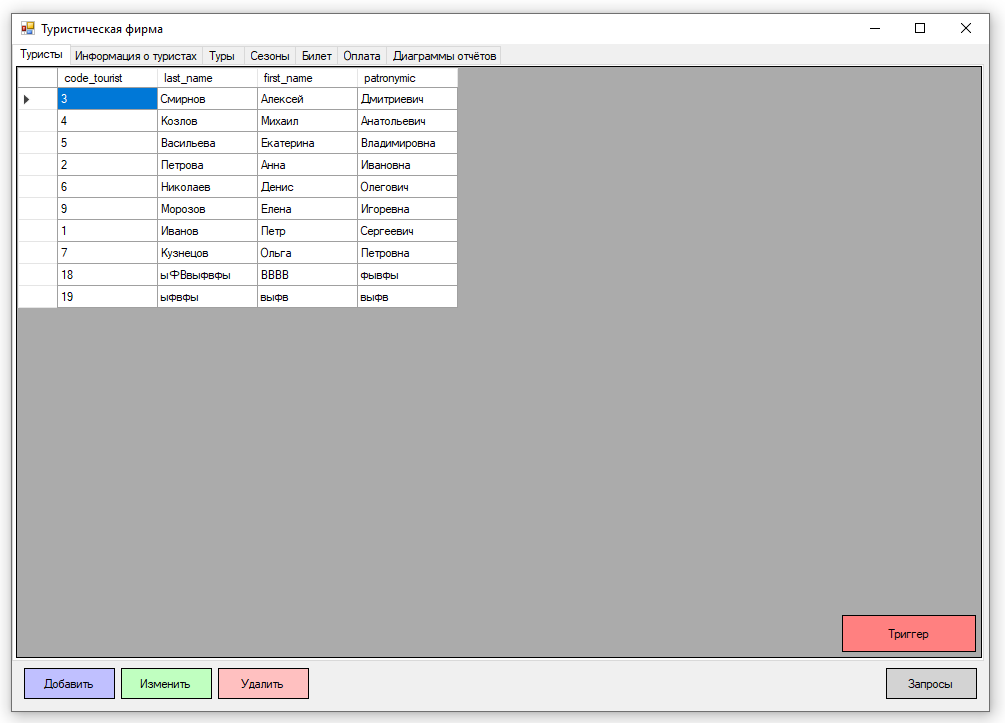


Рисунок 1 – Главное окно программы

На рисунке 1 представлено главное окно программы. Оно содержит вкладки для работы с различными аспектами данных, такими как туристы, информация о туристах, туры, сезоны, билеты и оплаты. Также есть вкладка для построения диаграмм отчетов. Внизу окна расположены кнопки для выполнения различных операций с данными.

Вкладка "Туристы" отображает таблицу с информацией о туристах, включая их ID, фамилию, имя и отчество. Здесь можно просматривать, добавлять, изменять и удалять записи о туристах.

Вкладка "Информация о туристах" предоставляет дополнительную информацию о туристах. В этой вкладке также представлена таблица с данными, которые можно редактировать и обновлять.

На вкладке "Туры" находится таблица, в которой отображается информация о турах, таких как название тура и цена. Здесь можно управлять данными о предлагаемых турах.

Вкладка "Сезоны" содержит таблицу с информацией о сезонах. В этой таблице представлены даты начала и окончания сезона, количество мест и статус доступности сезона.

На вкладке "Билет" отображается информация о билетах, включая ID билета и ID туриста. Эта вкладка предназначена для управления данными о билетах.

Вкладка "Оплата" показывает таблицу с информацией об оплате, включая дату оплаты и сумму. Здесь можно просматривать и управлять записями об оплатах.

Вкладка "Диаграммы отчетов" содержит две диаграммы (рисунок 2): круговую и столбчатую. Круговая диаграмма показывает процент выкупа туров клиентами, а столбчатая диаграмма отображает общую сумму выкупа всех туров по всем клиентам. Эти диаграммы помогают визуализировать данные о продажах туров.

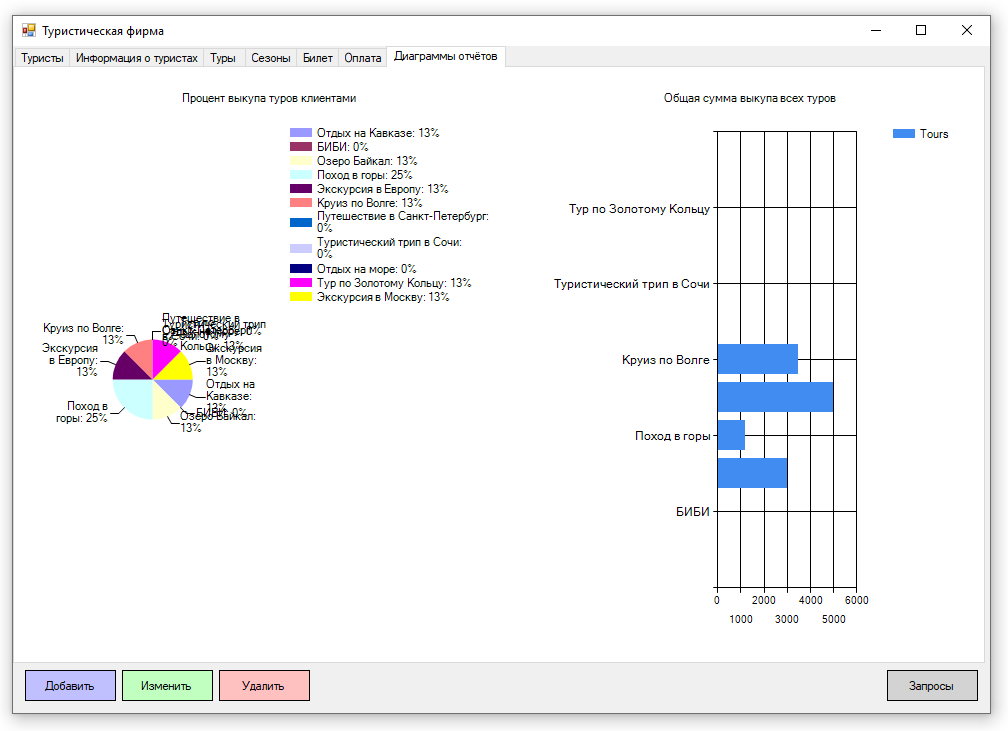


Рисунок 2 – Диаграммы на вкладке “Диаграммы отчетов”

Для построения графиков используются специальные функции создания DataTable объектов по запросу SQL (рисунок 3), а также функции для отрисовки полученной информации (рисунок 4).



Рисунок 3 – Функция создания DataTable для диаграммы

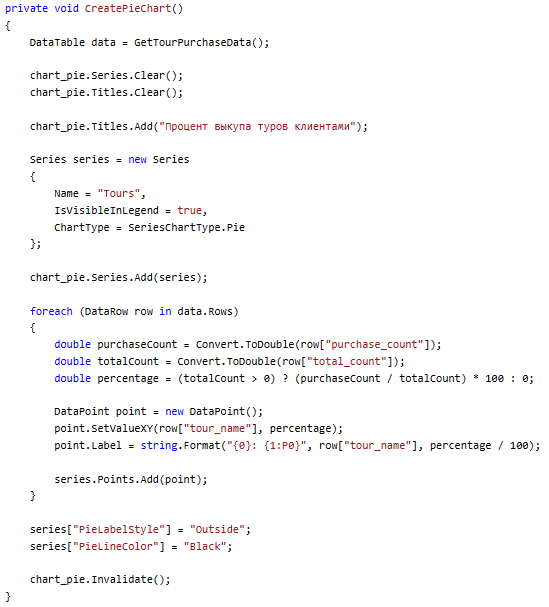


Рисунок 4 – Функция отрисовки диаграммы

Для загрузки данных в таблицы используются специальные функции (рисунок 5), которые заполняют DataGridView по SQL запросу.

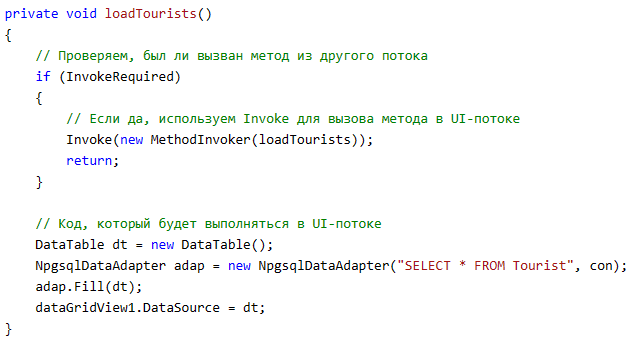


Рисунок 5 – Пример функции для загрузки данных в таблицу Туристы

По нажатию кнопки "Добавить" откроется новое окно для добавления записей (рисунок 7). В этом окне пользователь сможет ввести необходимые данные для создания новой записи в выбранной таблице. Окно представляет собой другую форму, которая вызывается в обработчике кнопки (рисунок 6). После ввода данных и подтверждения действия запись будет добавлена в базу данных, а таблица на главной форме обновится, чтобы отобразить новую запись.

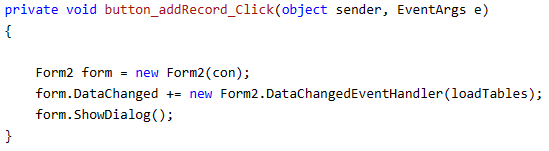


Рисунок 6 – Обработчик кнопки “Добавить”

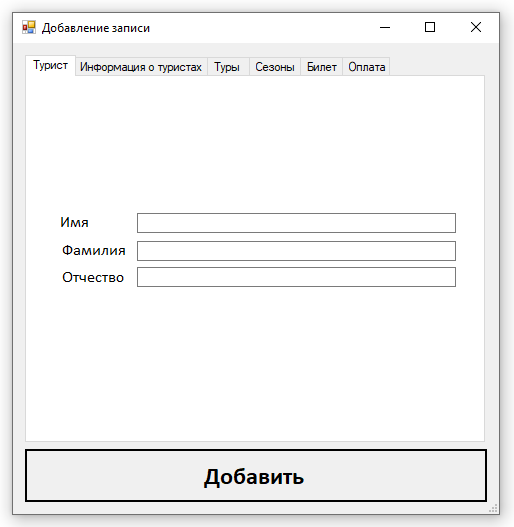


Рисунок 7 – Окно для добавления записей

Записи добавляются с помощью специальных функций добавления (рисунок 8). Внутри них программа получает данные из полей ввода формы и вставляет их в параметризированный запрос SQL. Для каждой таблицы своя функция.

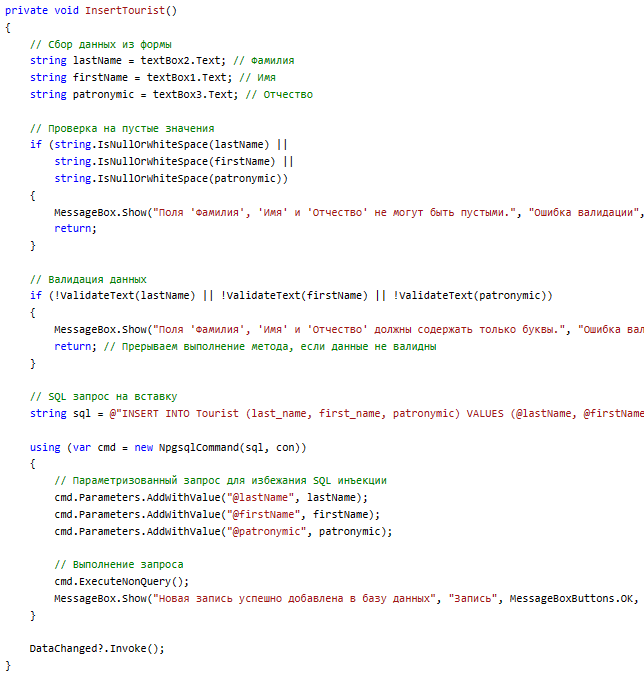


Рисунок 8 – Пример функции добавления записи

По нажатию кнопки "Изменить" пользователю предоставляется возможность редактировать существующую запись в выбранной таблице (рисунок 9). Для этого необходимо выбрать нужную строку в таблице и отредактировать данные непосредственно в ячейках таблицы. После внесения изменений данные автоматически обновятся в базе данных. Обновленная информация отобразится в таблице на главной форме.

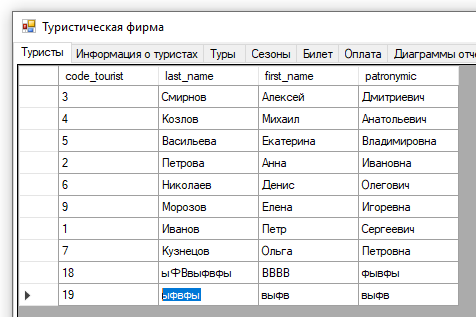


Рисунок 9 – Изменение записей по нажатию кнопки “Изменить”

Для изменения записей используются специальные функции для каждого DataGridView формы (рисунок 10). Внутри функций программа получает строчку и столбец ячейки, которую меняет пользователь, получает код этой ячейки, выполняет запрос SQL и обновляет таблицы.



Рисунок 10 – Пример функции изменения записи

По нажатию кнопки "Удалить" выбранная запись будет удалена из текущей таблицы. Пользователь должен выбрать строку, которую он хочет удалить, и нажать кнопку "Удалить". Программа запросит подтверждение удаления (рисунок 11), чтобы избежать случайных удалений. После подтверждения выбранная запись будет удалена из базы данных, и таблица на главной форме обновится, чтобы отобразить изменения.

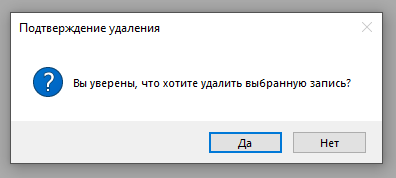


Рисунок 11 – Подтверждение удаления выбранной записи по нажатию кнопки “Удалить”

Для удаления используется специальная функция удаления (рисунок 12). Внутри нее программа проверяет в какой вкладке находится пользователь, берет код записи из таблицы в этой вкладке, выполняет запрос SQL на удаление и обновляет таблицы.

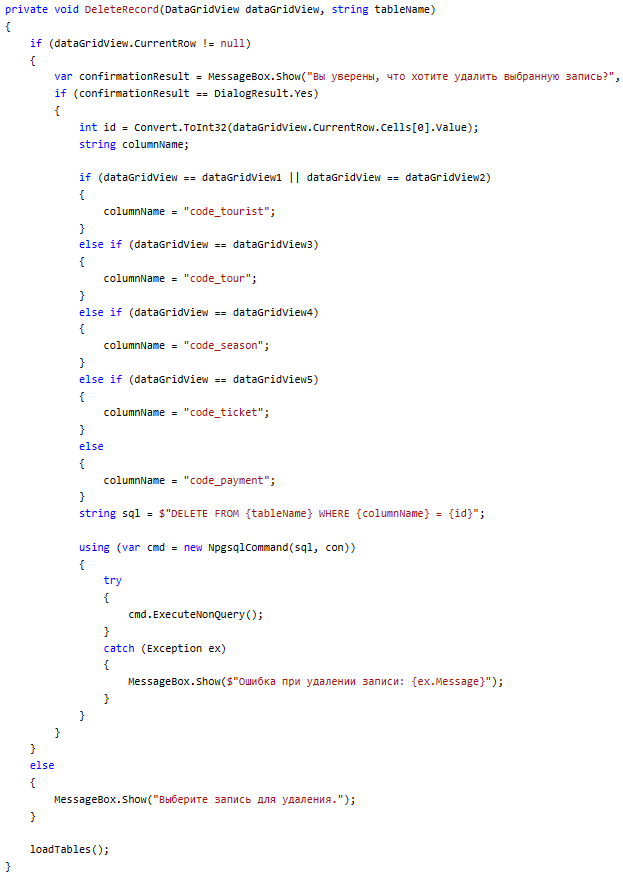


Рисунок 12 – Функция удаления записей

По нажатию кнопки "Запросы" откроется панель с инструментами для выполнения различных SQL-запросов (рисунок 13). В этой панели пользователь сможет вводить и выполнять агрегированные запросы, параметризованные запросы, а также экспортировать и импортировать данные в формате XML с помощью различных методов. Результаты запросов будут отображаться в соответствующих окнах или таблицах в зависимости от их типа.

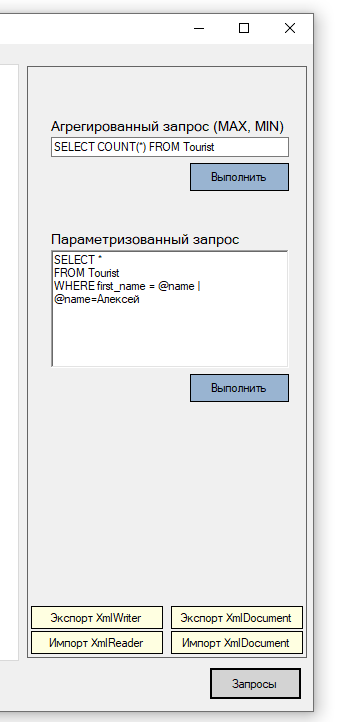


Рисунок 13 – Панель запросов

Для выполнения запросов используются специальный функции для двух видов запросов. Для агрегированных запросов (рисунок 14) функция просто выполняет запрос SQL, полученный из поля ввода. Для параметризированных запросов (рисунок 15) используется специальный шаблон запроса, по которому функция определяет, где параметр и куда его нужно вписать.

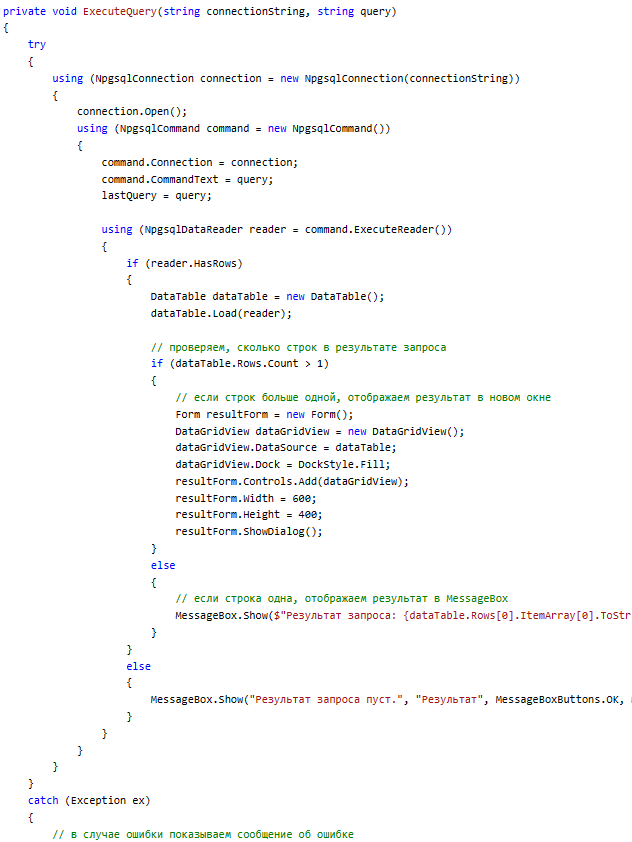


Рисунок 14 – Функция агрегированных запросов

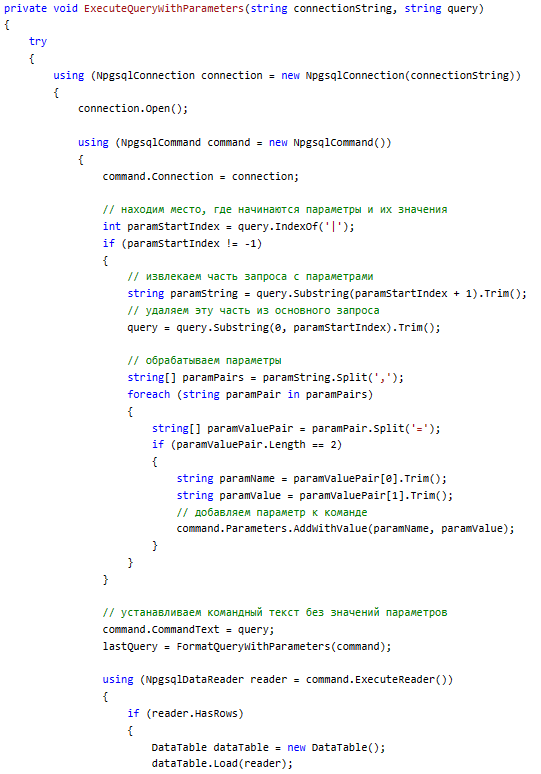


Рисунок 15 – Функция параметризированных запросов

Экспорт данных в формат XML начинается с проверки того, был ли выполнен хотя бы один запрос. Если нет, пользователю показывается сообщение об ошибке. Затем указывается путь для сохранения XML-файла, например, "result.xml". Создается объект XmlWriter, с помощью которого данные записываются в XML. Запись начинается с корневого элемента, после чего открывается соединение с базой данных и выполняется последний выполненный запрос. Для каждой строки результата создается элемент <Row>, содержащий данные в виде пар "имя колонки - значение". После завершения записи корневой элемент и документ закрываются, и пользователю показывается сообщение об успешной операции, подтверждающее, что данные успешно экспортированы в XML.

Импорт данных из XML начинается с проверки существования файла XML. Если файл не найден, пользователю показывается сообщение об ошибке. Затем создается объект XmlReader для чтения данных из XML-файла и пустая таблица DataTable. Для каждой строки <Row> из XML-файла добавляются соответствующие колонки и значения в таблицу DataTable. Если импортированные данные содержат несколько строк, они отображаются в новом окне с DataGridView. Если данных всего одна строка, результат показывается в виде сообщения пользователю.

Тот же принцип используется и для импорта и эспорта через XmlDocument. По заданию был написан триггер и триггерная функцию (рисунок 16), которая добавляет в таблицу «Билет» новую запись, при добавлении нового клиента в таблицу «Турист».

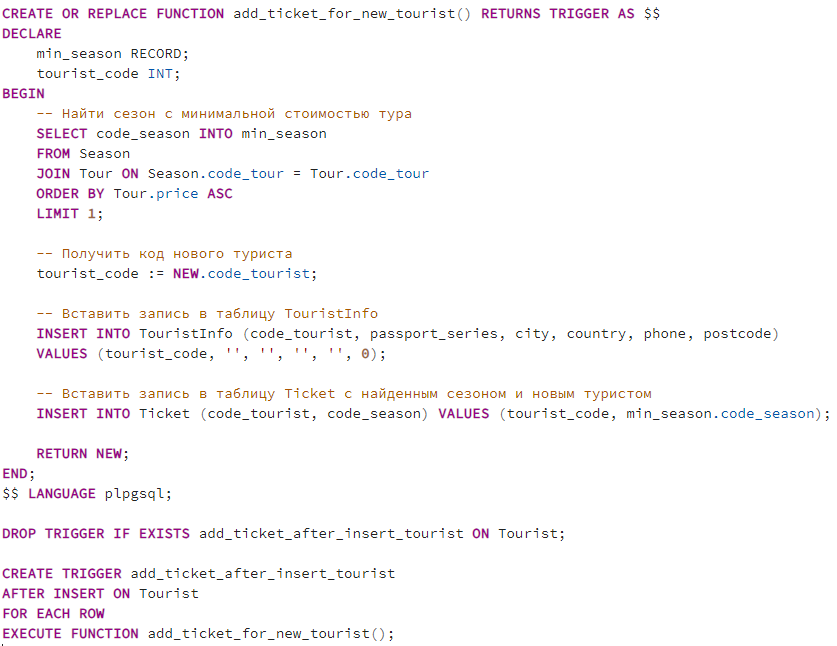


Рисунок 16 – Триггер и триггерная функция

**Вывод:** таким образом, в соответствии с заданием, мною разработано клиентское приложение на языке С# для работы с базой данных формата PostgreSQL.

Git репозиторий программы: https://github.com/Dadassasin/TourFirm