

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

ОТЧЁТ ПО КУРСОВОЙ РАБОТЕ

по дисциплине «Базы данных»

Вариант 4

Симоненко И. С.



Выполнил

5130904/20102

22350270

Степина Н. О.

Группа

№ зач. Книжки

Преподаватель

Санкт-Петербург

2024

Оглавление

1. Условие задачи.....	3
2. Требования к приложению курсовой работы	4
3. Используемые технологии и архитектура проекта	5
4. Триггера	6
5. Структура проекта	7
6. Демонстрация работы приложения	7
7. Возможность изменения данных несколькими пользователями	12
8. Вывод.....	14
9. Список литературы	14

1. Условие задачи

Автоматизация домашнего бюджета

В рамках данного цикла лабораторных работ необходимо автоматизировать работу ведения домашнего бюджета. Для этого в рамках базы данных PostgreSQL необходимо создать объекты в схеме вашего пользователя (логин и пароль пользователя для доступа к базе данных студент должен получить у преподавателя) и написать клиентское приложение на базе компонентов ADO.NET или JDBC.

Под домашним бюджетом понимается ведение статистики финансовых операций с разделением их на группы для последующего анализа и планирования. Финансовая операция характеризуется расходной суммой, приходной суммой и датой проведения. Для разделения операций на группы вводится понятие статьи, в рамках которой производится финансовая операция. Статья характеризуется наименованием назначения платежа. Каждая операция должна принадлежать одной и только одной статье. Для построения динамики распределения финансовых потоков по статьям и анализа используется расчетный период времени равный одному месяцу. Каждый месяц в системе автоматически должен формироваться итоговый отчетный документ, будем называть его балансом. Баланс характеризуется расходной суммой, приходной суммой, суммой чистой прибыли и датой формирования. Каждая операция может попасть только в один баланс. После формирования баланса покрываемый им период времени считается закрытым и для коррекции или создания операций в рамках этого периода, необходимо расформирование баланса. На основании попадания операции в баланс на этапе анализа и планирования принимается решение об интервале времени, в рамках которого будет учитываться операция. При анализе в рамках одного баланса все операции агрегируются по статьям.

База данных должна удовлетворять следующим требованиям:

1. Контроль целостности данных, используя механизм связей
2. Операции модификации групп данных и данных в связанных таблицах должны быть выполнены в рамках транзакций.
3. Логика работы приложения должна контролироваться триггерами. В частности:
 - Триггер должен запрещать ввод операции с нулевыми суммами
 - Триггер должен контролировать, чтобы вводимая операция не попадала в закрытый период.
4. Все операции вычисления различных показателей (из требований к клиентскому приложению) должны реализовываться хранимыми процедурами.

Требования к клиентскому приложению:

1. Вход в приложение осуществляется после корректного введения пароля и логина с соответствующими ролями (пользователь не может вносить изменения в БД, администратор владеет всеми правами). Логин и пароль хранятся в виде хэшей.
2. Необходимо реализовать интерфейсы для ввода, модификации и удаления

- Операций
 - Статей
3. Должен быть реализован журнал анализа, отображающий сформированные балансы. В рамках этого журнала необходимо организовать формирование баланса и расформирование.
 4. Необходимо реализовать возможность просмотра следующих показателей (лучше графический вид):
 - Динамика изменения расходов и доходов за произвольный период времени и с произвольным набором анализируемых статей.
 - Процентное соотношение финансовых потоков по статьям.
 - Графическое отображение чистой прибыли бюджета от времени.

2. Требования к приложению курсовой работы

Приложение представляет собой клиентскую часть GUI для БД спроектированной в 1-ом семестре. Язык GUI C#, Java, Python, C++, PHP. Компилятор любой.

Приложение включает следующие элементы:

- отдельный файл параметров подключения к БД, который задаются файлом (формат plain text file или ini config file или xml file);
- авторизацию с хешированием паролей для пользователей (2 пользователя);
- меню приложения содержит группы элементов: СПРАВОЧНИКИ, ЖУРНАЛЫ, ОТЧЕТЫ;
- приложение должно обеспечивать ввод, редактирование, просмотр информации из БД;
- используется пользовательский элемент вида GridView для обзора таблиц (представлений) БД;
- GridView, который содержит внешние ключи, должен содержать нужные/необходимые атрибуты внешней таблицы.
- есть отдельная форма для работы с записью таблицы, содержащая эл-ты интерфейса ДОБАВИТЬ, ИЗМЕНИТЬ, УДАЛИТЬ, ЗАКРЫТЬ; открывается из GridView
- меню пункта ОТЧЕТЫ вызывает формы вывода отчетов (минимум 2-х) (выходной формат txt или xls или pdf).

3. Используемые технологии и архитектура проекта

Основные компоненты и технологии

1. Язык программирования: Python

- Python был выбран из-за его читаемости, простоты и богатого набора библиотек, которые ускоряют разработку приложений.

2. GUI: PyQt5

- PyQt5 — это набор привязок для инструментов разработки графических интерфейсов Qt5 на языке Python. Он предоставляет мощные возможности для создания кроссплатформенных GUI-приложений.
- В проекте PyQt5 используется для создания окон приложения, таких как главное окно, окно операций, окно статей и окно балансов.

3. База данных: PostgreSQL

- PostgreSQL — это мощная, открытая объектно-реляционная система управления базами данных. Она используется в проекте для хранения данных о пользователях, операциях, статьях и балансах.
- Взаимодействие с базой данных осуществляется с использованием библиотеки pg8000, которая предоставляет интерфейс для работы с PostgreSQL.

4. Управление данными: pg8000

- pg8000 — это чистая реализация PostgreSQL клиента на Python, которая используется для выполнения SQL-запросов и взаимодействия с базой данных.

5. Контекстный менеджер: contextlib

- Модуль contextlib используется для управления соединениями и курсорами базы данных, обеспечивая безопасное открытие и закрытие ресурсов.

6. Работа с конфигурационными файлами: configparser

- Модуль configparser используется для чтения конфигурационных файлов, которые содержат параметры подключения к базе данных.

7. Пакеты и модули проекта

- Проект структурирован по модульному принципу, разделяя функциональность на различные модули и пакеты:
 - gui: содержит файлы, отвечающие за графический интерфейс приложения.
 - services: содержит бизнес-логику и взаимодействие с базой данных.
 - db_manager.py: отвечает за управление подключением и выполнением запросов к базе данных.

Основные функции приложения

1. Авторизация и аутентификация пользователей

- Реализовано с помощью хранения хэшированных паролей в базе данных и проверки учетных данных при входе в систему.

2. Управление операциями

- Пользователи могут добавлять, редактировать и удалять операции. Введены триггеры для проверки корректности данных и предотвращения ошибок.

3. Управление статьями бюджета

- Пользователи могут добавлять и редактировать статьи бюджета, которые затем используются при вводе операций.

4. Формирование и расформирование балансов

- Реализованы функции для генерации и расформирования балансов. Генерация балансов агрегирует данные операций за определённый период, а расформирование позволяет редактировать операции, входящие в уже сформированные балансы.

5. Экспорт данных и графиков

- Реализована функция экспорта данных в Excel с использованием библиотеки `openpyxl`.

4. Триггера

Триггер для проверки нулевых сумм:

- Этот триггер предотвращает добавление операций с нулевыми суммами доходов или расходов. Он проверяет значения полей `income_amount` и `expense_amount` перед вставкой или обновлением записей в таблице `operations`, и если оба значения равны нулю, операция отклоняется.

Триггер для предотвращения изменений в закрытых периодах:

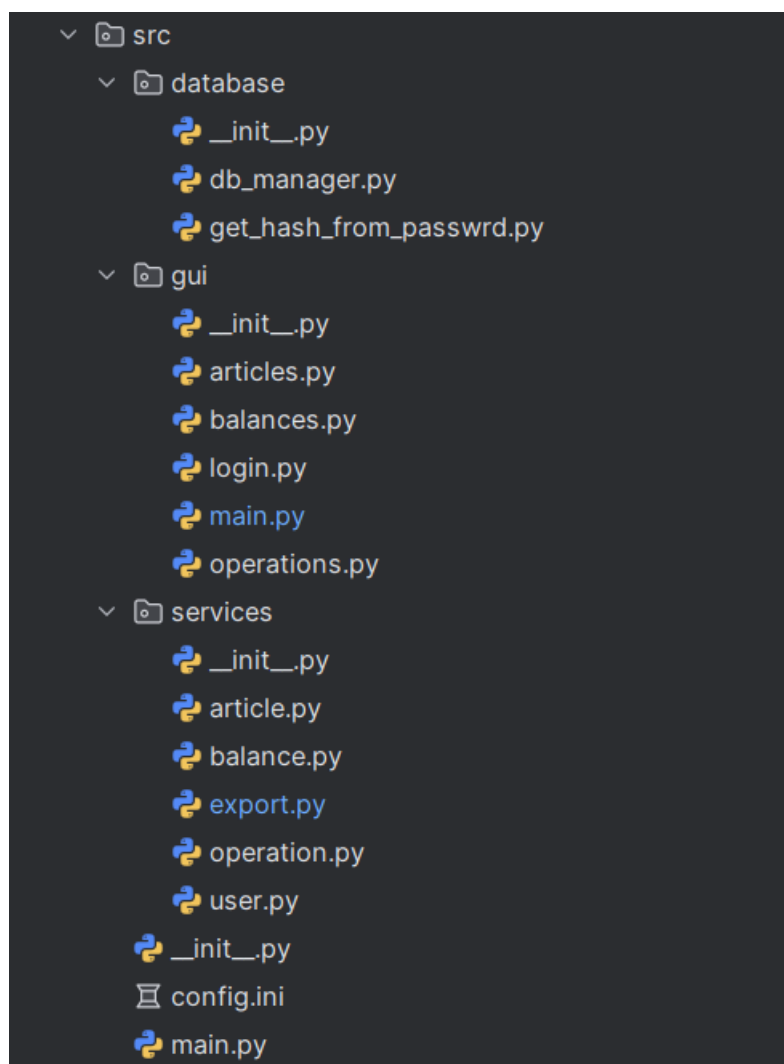
- Данный триггер контролирует, чтобы операции не могли быть добавлены или изменены в закрытых периодах, т.е. в периодах, для которых уже сформированы балансы. Он проверяет дату операции и сравнивает её с периодами, закрытыми в таблице `balances`. Если операция попадает в закрытый период, то выполнение вставки или обновления блокируется.

▼ Trigger Functions (2)

 `check_closed_period()`

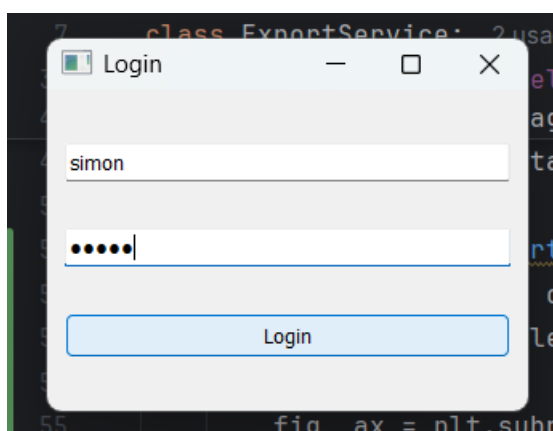
 `check_non_zero_amounts()`

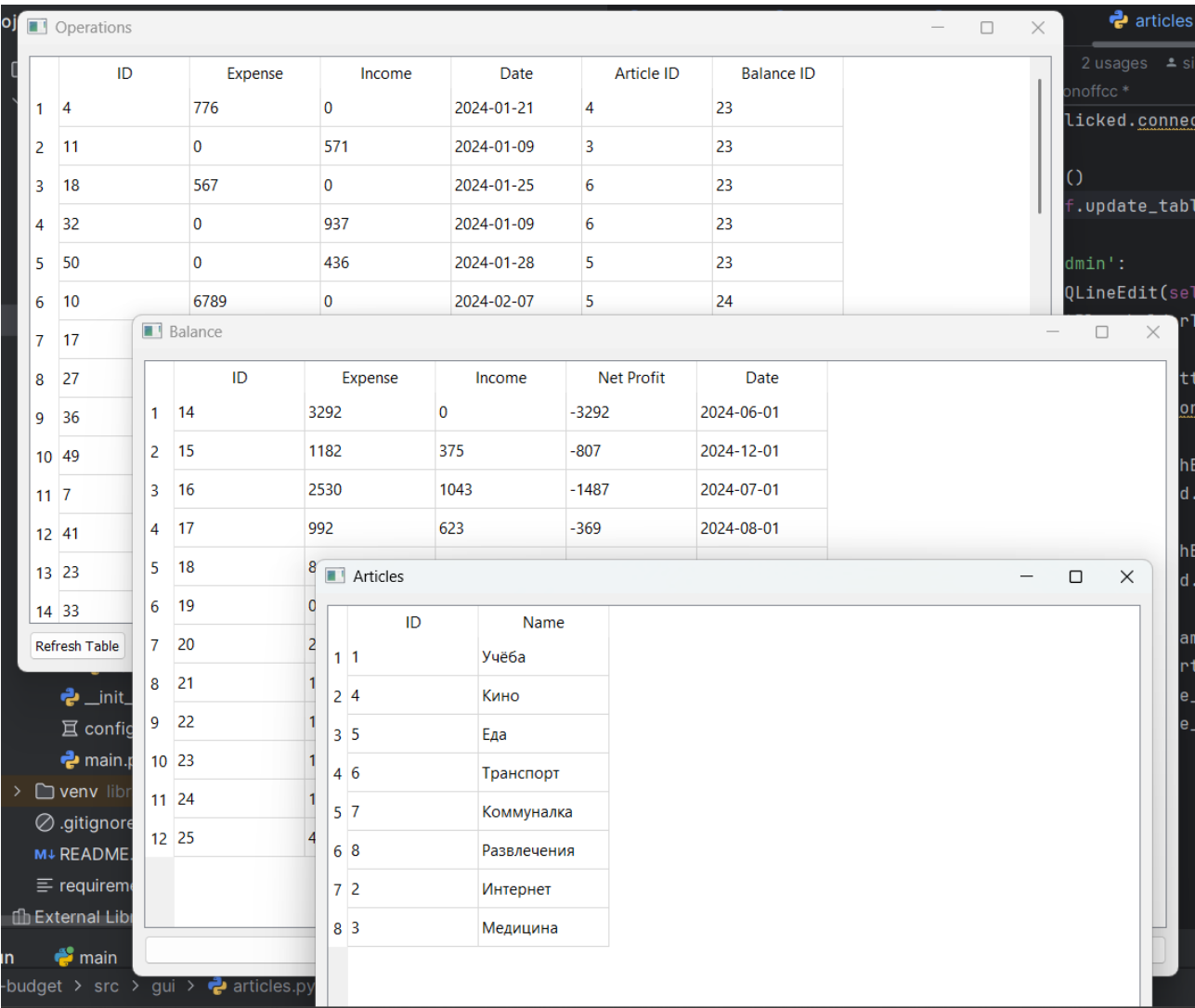
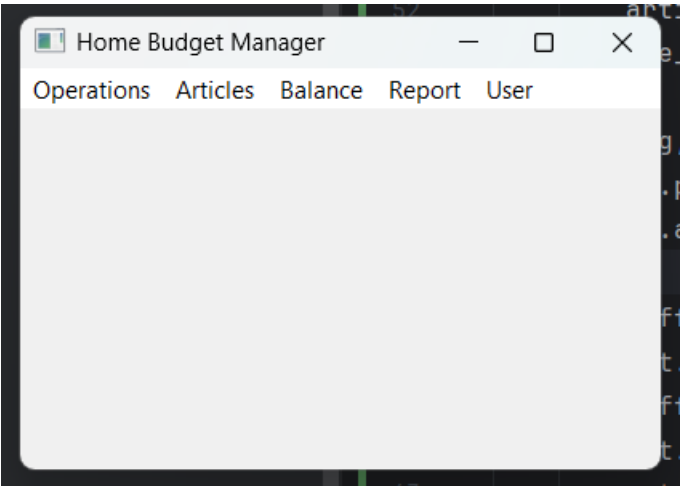
5. Структура проекта



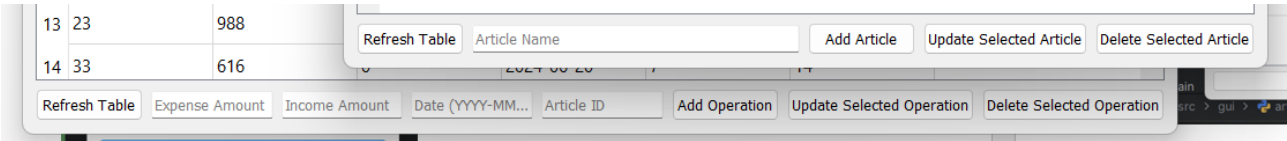
6. Демонстрация работы приложения

Вход за администратора.





Панель для админа выглядит так:



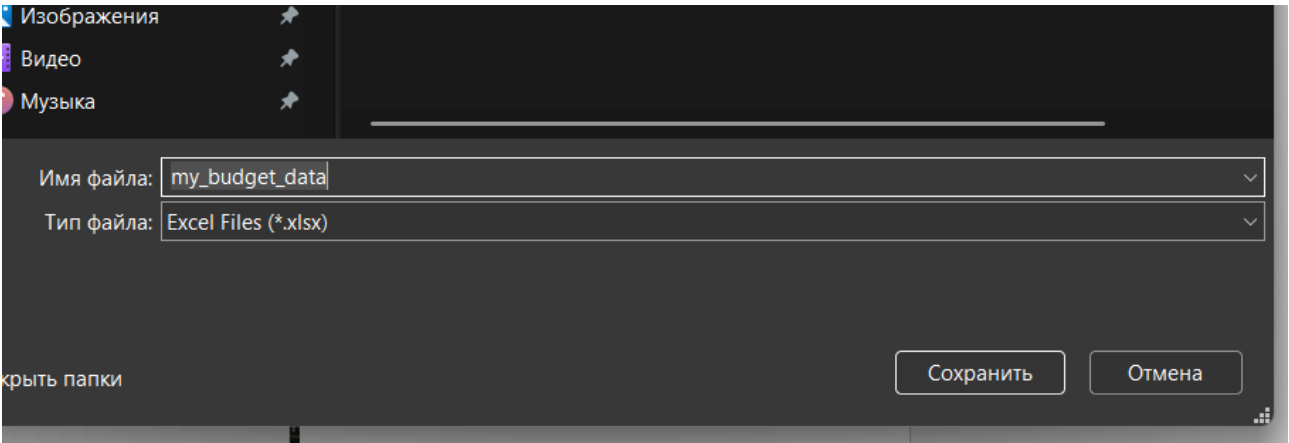
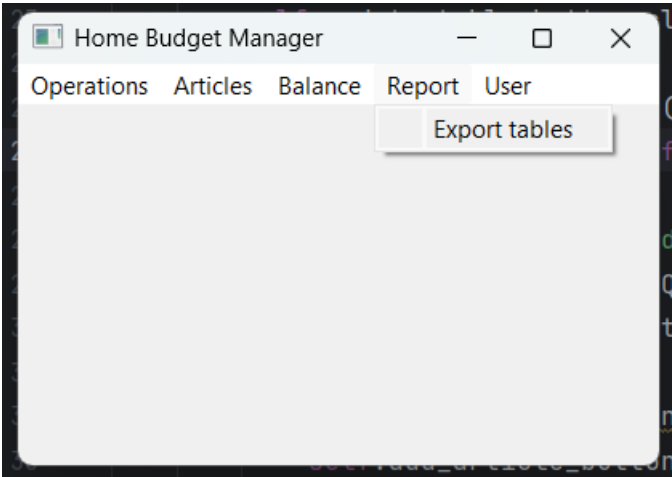
Панель для юзера выглядит так:

12	41	0	352	2
13	23	988	0	2
14	33	616	0	20

Refresh Table

Refresh Table

Экспорт данных:



	A	B	C	D	E	F
1	num	operation_date	income_amount	expense_amount	article_name	
2	1	2024-01-21	0	776	Кино	
3	2	2024-01-09	571	0	Медицина	
4	3	2024-01-25	0	567	Транспорт	
5	4	2024-01-09	937	0	Транспорт	
6	5	2024-01-28	436	0	Еда	
7	6	2024-02-07	0	6789	Еда	
8	7	2024-02-26	754	0	Коммуналка	
9	8	2024-02-01	0	7567	Еда	
10	9	2024-02-26	0	621	Еда	
11	10	2024-02-25	0	894	Медицина	
12	11	2024-09-29	0	401	Транспорт	
13	12	2024-09-24	352	0	Коммуналка	
14	13	2024-06-26	0	988	Кино	
15	14	2024-06-20	0	616	Коммуналка	
16	15	2024-06-22	0	362	Медицина	
17	16	2024-06-10	0	890	Кино	
18	17	2024-06-21	0	436	Еда	
19	18	2024-12-09	375	0	Транспорт	
20	19	2024-12-18	0	973	Интернет	
21	20	2024-12-19	0	209	Развлечения	
22	21	2024-07-30	198	0	Кино	
23	22	2024-07-02	0	678	Еда	
24	23	2024-07-01	845	0	Кино	
25	24	2024-07-12	0	506	Транспорт	
26	25	2024-07-11	0	429	Еда	
27	26	2024-07-31	0	917	Транспорт	
28	27	2024-08-18	0	992	Развлечения	

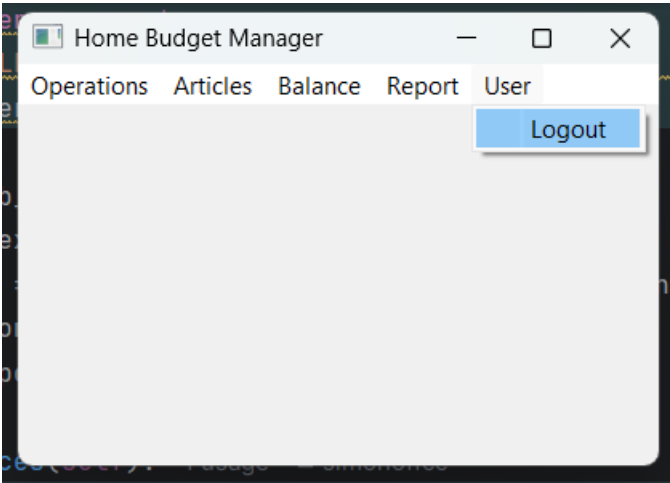
Статьи
Операции
Балансы
+

	A	B
1	article_name	
2	Учёба	
3	Кино	
4	Еда	
5	Транспорт	
6	Коммуналка	
7	Развлечения	
8	Интернет	
9	Медицина	
10		

	A	B	C	D	E	F
1	num	expense_sum	income_sum	net_profit	balance_date	
2	1	3292	0	-3292	2024-06-01	
3	2	1182	375	-807	2024-12-01	
4	3	2530	1043	-1487	2024-07-01	
5	4	992	623	-369	2024-08-01	
6	5	849	3587	2738	2024-11-01	
7	6	0	516	516	2024-10-01	
8	7	216	523	307	2024-05-01	
9	8	1066	2400	1334	2024-03-01	
10	9	1448	670	-778	2024-04-01	
11	10	1343	1944	601	2024-01-01	
12	11	15871	754	-15117	2024-02-01	
13	12	401	352	-49	2024-09-01	
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						

Статьи
Операции
Балансы

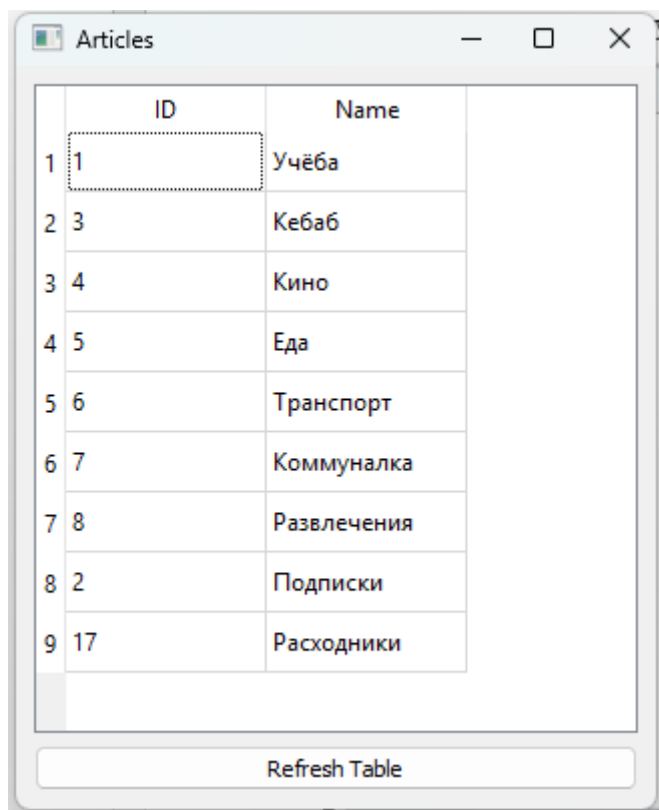
Выход и программы:



7. Возможность изменения данных несколькими пользователями

В графическом интерфейсе реализован функционал для обновления данных в таблице при открытых таблицах, поэтому если кто-либо изменит данных в базе, то эти данные можно будет сразу отобразить в графическом интерфейсе.

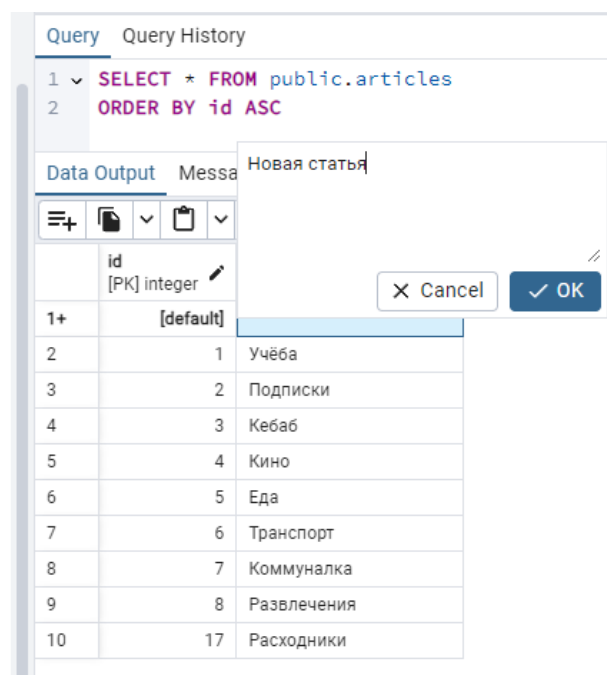
Таблица до изменения в другом источнике:



	ID	Name
1	1	Учёба
2	3	Кебаб
3	4	Кино
4	5	Еда
5	6	Транспорт
6	7	Коммуналка
7	8	Развлечения
8	2	Подписки
9	17	Расходники

Refresh Table

Изменение данных другим пользователем:



Query Query History

```
1 SELECT * FROM public.articles
2 ORDER BY id ASC
```

Data Output Message

Новая статья

id [PK] integer

1+ [default]

2	1	Учёба
3	2	Подписки
4	3	Кебаб
5	4	Кино
6	5	Еда
7	6	Транспорт
8	7	Коммуналка
9	8	Развлечения
10	17	Расходники

Cancel OK

Создали новую статью:

	id	name	
	[PK] integer	character varying (100)	
1	18	Новая статья	
2	1	Учёба	
3	2	Подписки	
4	3	Кебаб	
5	4	Кино	
6	5	Еда	
7	6	Транспорт	
8	7	Коммуналка	
9	8	Развлечения	
10	17	Расходники	

Обновляем данные в интерфейсе:

Articles			
	ID	Name	
1	1	Учёба	
2	3	Кебаб	
3	4	Кино	
4	5	Еда	
5	6	Транспорт	
6	7	Коммуналка	
7	8	Развлечения	
8	2	Подписки	
9	17	Расходники	
10	18	Новая статья	
Refresh Table			

8. Вывод

В ходе выполнения работы было разработано приложение представляющее собой клиентскую часть GUI для базы данных работы автопарка. Были изучены современные инструменты для работы с БД и реализации графического интерфейса. Значительных сложностей в ходе разработки не возникло.

9. Список литературы

1. **PostgreSQL Documentation:** Официальная документация PostgreSQL. Включает подробное описание использования SQL-запросов, создания триггеров и хранимых процедур. [электронный ресурс] <https://www.postgresql.org/docs/>
2. **Python Documentation:** Официальная документация по языку программирования Python, включая модули для работы с базами данных. [электронный ресурс] <https://docs.python.org/3/>
3. **pg8000 Documentation:** Документация по использованию библиотеки pg8000 для подключения и взаимодействия с PostgreSQL из Python. [электронный ресурс] <https://pg8000.readthedocs.io/>
4. **PyQt5 Documentation:** Официальная документация по библиотеке PyQt5, используемой для создания графического интерфейса. [электронный ресурс] <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
5. **ConfigParser Documentation:** Документация по модулю ConfigParser для работы с конфигурационными файлами в Python. [электронный ресурс] <https://docs.python.org/3/library/configparser.html>
6. **Matplotlib Documentation:** Официальная документация по библиотеке Matplotlib, используемой для создания графиков и диаграмм. [электронный ресурс] <https://matplotlib.org/stable/contents.html>
7. **Pandas Documentation:** Документация по библиотеке Pandas, используемой для работы с табличными данными и их экспорта в Excel. [электронный ресурс] <https://pandas.pydata.org/docs/>
8. **OpenPyXL Documentation:** Документация по библиотеке OpenPyXL для работы с Excel-файлами из Python. [электронный ресурс] <https://openpyxl.readthedocs.io/>
9. **Stack Overflow:** Онлайн-сообщество разработчиков для обсуждения вопросов и поиска решений по программированию. [электронный ресурс] <https://stackoverflow.com/>