Колледж ТОО “Astana IT University”

Отчёт по

производственному обучению

Приложение для учёта расходных материалов

в учебном заведении

ПМ 01 - Администрирование база данных

ПМ 02 - Составление алгоритма и создание блок-схемы на основе спецификации программного обеспечения

Группа: ПО2511

ФИО студента:

Гамиль Диас Данiярулы

ФИО преподавателей:

Аманова Ф.Ж.

Баймаканов А.Т..

Астана 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном функционировании организаций любого масштаба — будь то офисы, образовательные учреждения, производственные предприятия или склады — ежедневно используются различные расходные материалы: канцелярские принадлежности, инструменты, запчасти и хозяйственные товары. Их своевременный и точный учёт играет ключевую роль в обеспечении бесперебойной работы, рациональном расходовании бюджета и предотвращении ненужных затрат.

Традиционные методы учёта, основанные на ведении журналов или таблиц, зачастую оказываются недостаточно эффективными. Они подвержены человеческим ошибкам, риску потери данных и не обеспечивают актуальной информации об остатках материалов. В таких условиях принятие управленческих решений становится менее точным, а контроль над расходами — затруднённым.

Автоматизация учёта расходных материалов позволяет существенно повысить эффективность работы организации. Применение специализированных программных решений снижает влияние человеческого фактора, упрощает контроль над остатками, ускоряет обработку данных и предоставляет возможность оперативно получать актуальную информацию о наличии ресурсов.

Цель проекта — разработать приложение для учёта расходных материалов, обеспечивающее полный контроль над движением ресурсов. Система поддерживает добавление новых материалов, редактирование записей, списание и контроль остатков, что позволяет всегда иметь актуальные данные. Хранение информации будет реализован на базе PostgreSQL, а графический интерфейс будет создан на Python с использованием Tkinter.

Разработанное приложение обеспечивает надёжный учёт материалов, снижает вероятность ошибок и способствует более рациональному использованию ресурсов, повышая эффективность работы организации.

Задачи проекта - В рамках проекта необходимо проанализировать требования к системе учёта расходных материалов и определить перечень функций, которые должны быть реализованы в приложении.

- Следует спроектировать структуру базы данных, включающую таблицы материалов, категорий, транзакций, пользователей и логов, а также продумать связи между ними.
- Требуется создать базу данных PostgreSQL и настроить ограничения, обеспечивающие корректность и целостность хранимых данных.
- Необходимо разработать удобный графический интерфейс пользователя на языке Python с использованием библиотеки Tkinter.
- Важно реализовать функционал добавления категорий и материалов, выполнения операций прихода и расхода, удаления записей и просмотра истории операций.
- Далее необходимо провести тестирование разработанного приложения и убедиться в корректной работе всех функций и взаимодействия с базой данных.
- Завершающим шагом является подготовка отчёта о проделанной работе и презентации, отражающих результаты разработки.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Выбор инструментов по базы данных и python

В основе проекта лежит продуманная модель данных, состоящая из **пяти ключевых таблиц**, каждая из которых отвечает за свою область логики:

- **materials** — карточки материалов с количеством;
- **categories** — типы и группы материалов;
- **users** — данные пользователей системы;
- **roles** — уровни доступа и функции;
- **transactions** — журнал операций поступления и расхода.

Такая структура обеспечивает:

- чёткое разнесение логики;
- удобство фильтрации;
- предотвращение ошибок данных;

```sql
CREATE TABLE categories (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);
CREATE TABLE materials (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    unit VARCHAR(50) NOT NULL,
    quantity INTEGER DEFAULT 0,
    min_quantity INTEGER DEFAULT 0,
    category_id INTEGER REFERENCES categories(id) ON DELETE SET NULL
);

CREATE TABLE transactions (
    id SERIAL PRIMARY KEY,
    material_id INTEGER REFERENCES materials(id) ON DELETE CASCADE,
    type VARCHAR(10) CHECK (type IN ('приход', 'расход')),
    amount INTEGER NOT NULL,
    date DATE DEFAULT CURRENT_DATE,
    comment TEXT
);

CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    role VARCHAR(20) NOT NULL DEFAULT 'user',
    created_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE logs (
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(id),
    action VARCHAR(255) NOT NULL,
    timestamp TIMESTAMP DEFAULT NOW(),
    details TEXT
);
```

Рисунок 1.1

1.2 Таблицы и ключи

- Каждая таблица содержит продуманную структуру данных:
- **Первичные ключи (PRIMARY KEY)** обеспечивают уникальность каждой записи.
- **Уникальные поля (UNIQUE)** препятствуют появлению дублирующихся категорий и других данных.
- **NOT NULL** гарантирует обязательность важной информации.
- **CHECK-ограничения** контролируют корректность значений, например, количество материала не может быть отрицательным.
- **Именованные столбцы** отражают реальную структуру материальных объектов.
- Данная архитектура повышает надёжность хранения и облегчает дальнейшую обработку данных.

Пример:

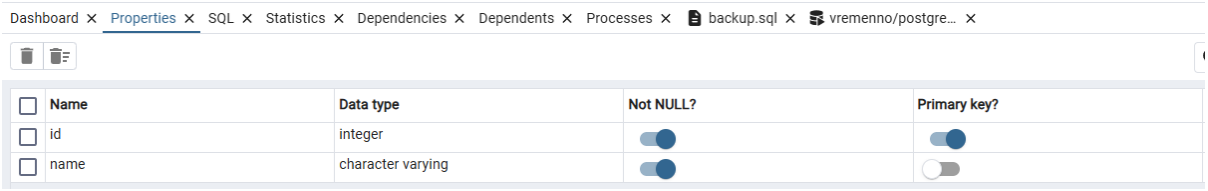

Рисунок 1.2.1

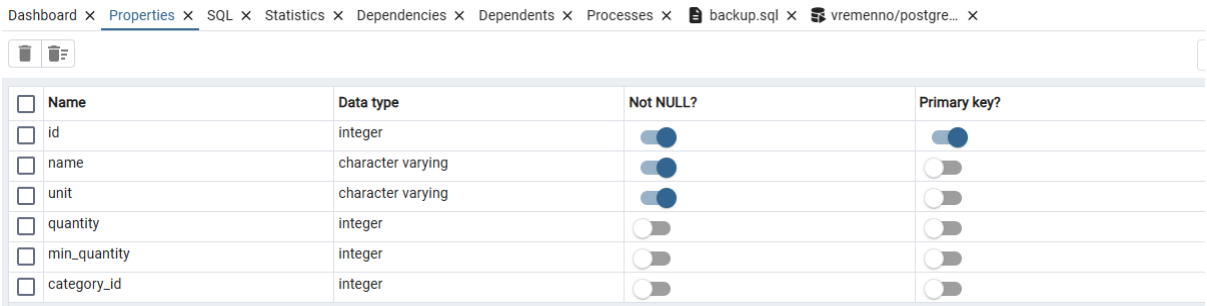

Рисунок 1.2.2

1.3 Логические связи

Модель данных построена на основе трёх типов связей:

1.3.1 1 → Many

- категория может содержать множество материалов;
- один материал может иметь множество операций.

1.3.2 Many ↔ Many

- пользователи и роли связаны через промежуточную таблицу.

1.3 3 → 1 (частично)

- дополнительные данные пользователей.

Такая структура обеспечивает:

- целостность данных,
- строгую связь объектов,
- предотвращение ошибок при операциях,
- масштабируемость системы.

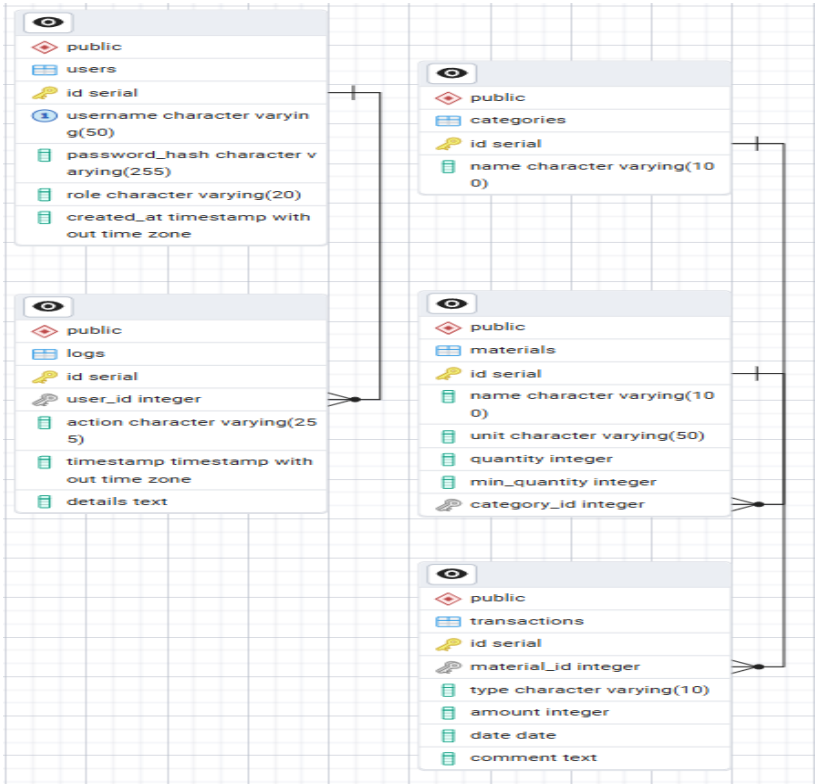

Рисунок 1.3

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 SQL-запросы: простые

В приложении активно используются базовые SQL-команды, которые формируют основу взаимодействия с базой:

- **SELECT** — получение данных материалов, категорий и операций;
- **INSERT** — добавление новых материалов или категорий;
- **UPDATE** — изменение количества или комментариев;
- **DELETE** — удаление записей с проверкой связей.

Эти запросы вызываются автоматически через интерфейс программы и являются основой всей её функциональности.

```
96  SELECT * FROM materials;
```

Data Output | Messages | Notifications

SQL

	id [PK] integer	name character varying (100)	unit character varying (50)	quantity integer	min_quantity integer	category_id integer
1	1	Бумага А4	пачка	20	5	1
2	2	Ручка синяя	шт	150	30	1
3	3	Маркер перманентный	шт	40	10	1
4	5	Мыло жидкое	литр	10	3	2
5	6	Кабель HDMI	шт	12	3	3
6	7	Мышь компьютерная	шт	18	5	3
7	8	Тетрадь 12 листов	шт	200	50	4
8	9	Обложка для тетрадей	шт	300	100	4
9	10	Краска акриловая	банка	35	10	5
10	11	Кисть малярная	шт	50	20	5
11	4	Салфетки влажные	упаковка	25	5	2

Рисунок 2.1

2.2 SQL-запросы: сложные

- Для формирования отчётов, фильтрации и аналитики используются более продвинутые SQL-конструкции:
- **JOIN** — объединение таблиц для получения расширенной информации;
- **GROUP BY** — группировка данных для отчётов;
- **Подзапросы** — выборка значений на основе внутренних запросов;
- **ORDER BY** — сортировка для удобства пользователя;
- **WHERE с несколькими условиями** — точная фильтрация.
- Благодаря таким запросам отчёты становятся точными и информативными

```
101  SELECT m.id,
102         m.name AS material_name,
103         c.name AS category_name,
104         m.quantity
105  FROM materials m
106  JOIN categories c ON c.id = m.category_id;
107
```

Data Output | Messages | Notifications

	id integer	material_name character varying (100)	category_name character varying (100)	quantity integer
1	1	Бумага А4	Канцелярия	20
2	2	Ручка синяя	Канцелярия	150
3	3	Маркер перманентный	Канцелярия	40
4	5	Мыло жидкое	Хозяйственные товары	10
5	6	Кабель HDMI	Компьютерные комплектующие	12
6	7	Мышь компьютерная	Компьютерные комплектующие	18
7	8	Тетрадь 12 листов	Учебные материалы	200
8	9	Обложка для тетрадей	Учебные материалы	300
9	10	Краска акриловая	Расходные материалы для мастерск...	35
10	11	Кисть малярная	Расходные материалы для мастерск...	50
11	4	Салфетки влажные	Хозяйственные товары	25

Рисунок 2.2

3. Оптимизация и производительность

Для повышения производительности реализован комплекс мер:

3.1.1 Индексация:

- индексы на первичных и внешних ключах;
- индексы на часто используемых полях (category_id).

3.2.2 Оптимизированные запросы:

- исключение дублирующихся операций;
- сокращение повторных обращений к БД;
- логически оптимизированные конструкции.

3.3.3 Поведенческие ограничения:

- строгие типы данных;
- автоматическая обработка некоторых операций.

Все эти меры делают работу системы стабильной даже при больших объёмах данных.

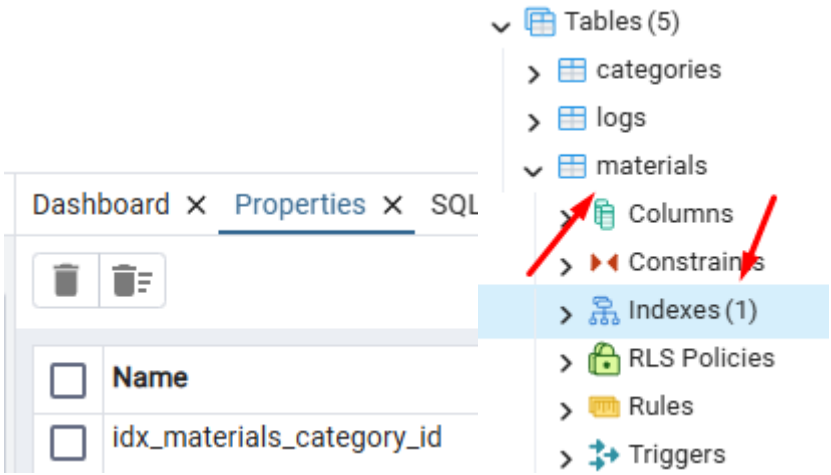

Рисунок 3.3

4. Автоматизация

Приложение автоматизирует множество рутинных действий:

- автоматическое обновление остатков при любой операции;
- автоматический выбор категории материала;
- автоматическая запись всех действий в журнал операций;
- автоматическое оповещение пользователя об ошибках;
- автоматическая проверка обязательных полей.

Такая автоматизация снижает нагрузку на пользователя и исключает человеческий фактор.

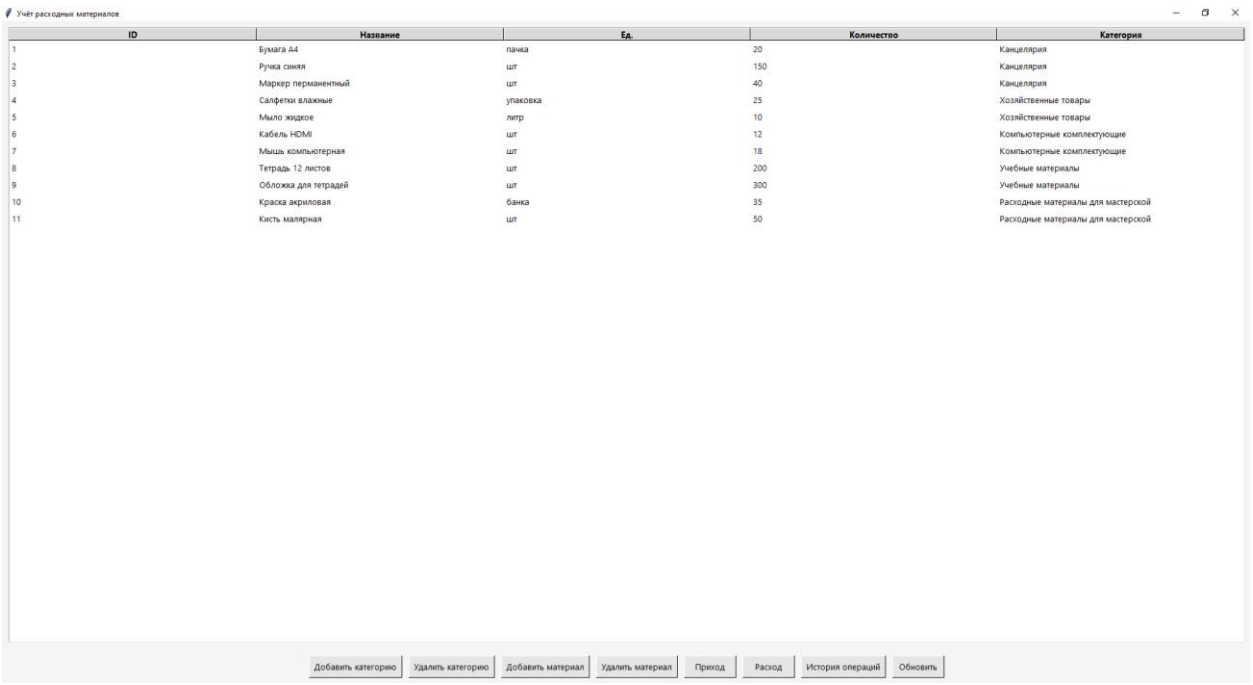

Рисунок 4

5. Логика обработки операций (приход и расход)

В системе реализована полноценная логика учёта движения материалов:

5.1 Как работает приход

- Пользователь выбирает материал
- Вводит количество
- Остаток автоматически **увеличивается**
- Операция записывается в таблицу *transactions*
- В интерфейсе сразу обновляется текущий остаток

5.2 Как работает расход

- Проверяется достаточное количество на складе
- Если материала меньше — отображается ошибка
- Если достаточно — остаток автоматически **уменьшается**
- Все данные сохраняются в историю операций

5.3 Преимущества такой логики

- Исключает человеческие ошибки
- Позволяет отслеживать всю историю движения материалов

  Обеспечивает корректный учёт в реальном времени

```
INSERT INTO transactions (material_id, type, amount, comment) VALUES
-- Бумага А4
(1, 'приход', 10, 'Пополнение склада'),
(1, 'расход', 3, 'Выдано учителю'),
(1, 'расход', 2, 'Использовано на печать документов'),

-- Ручки
(2, 'расход', 20, 'Выдача студентам'),
(2, 'приход', 50, 'Закуп новых ручек'),

-- Маркеры
(3, 'расход', 5, 'Исписались'),
(3, 'приход', 10, 'Новая поставка'),

-- Хозтовары
(4, 'расход', 5, 'Для уборки'),
(5, 'приход', 5, 'Закупка расходников'),

-- Комплектующие
(6, 'приход', 5, 'Поставка HDMI-кабелей'),
(6, 'расход', 2, 'Использовано в аудитории'),
(7, 'расход', 3, 'Выдано в компьютерный класс'),

-- Учебные материалы
(8, 'расход', 40, 'Раздано студентам'),
(9, 'приход', 100, 'Закупка перед семестром'),

-- Мастерская
(10, 'приход', 10, 'Поставка акриловой краски'),
(10, 'расход', 3, 'Использовано на занятии'),
(11, 'расход', 10, 'Использовано в мастерской');
```

Рисунок 5

6. Отчёты и аналитика

Система формирует информативные отчёты:

- отчёт по остаткам материалов;
- отчёт по категориям;
- отчёт по операциям расхода/прихода;
- список наиболее часто используемых материалов;
- фильтры по датам и категориям.

Отчёты позволяют оценивать активность использования материалов и принимать управленческие решения.

```
101  SELECT m.id,
102         m.name AS material_name,
103         c.name AS category_name,
104         m.quantity
105  FROM materials m
106  JOIN categories c ON c.id = m.category_id;
```

Data Output | Messages | Notifications

SQL

	id integer	material_name character varying (100)	category_name character varying (100)	quantity integer
1	3	Маркер перманентный	Канцелярия	40
2	2	Ручка синяя	Канцелярия	150
3	1	Бумага А4	Канцелярия	20
4	4	Салфетки влажные	Хозяйственные товары	25
5	5	Мыло жидкое	Хозяйственные товары	10
6	7	Мышь компьютерная	Компьютерные комплектующие	18
7	6	Кабель HDMI	Компьютерные комплектующие	12
8	9	Обложка для тетрадей	Учебные материалы	300
9	8	Тетрадь 12 листов	Учебные материалы	200
10	11	Кисть малярная	Расходные материалы для мастерск...	50
11	10	Краска акриловая	Расходные материалы для мастерск...	35

Рисунок 6

7 Тестирование

Проект проходил комплексное тестирование:

7.1 Функциональное тестирование:

- операции добавления расхода/прихода;
- корректное обновление данных;
- создание категорий и материалов.

7.2 Тестирование интерфейса:

- проверка всех кнопок;
- проверка форм ввода;
- тестирование сценариев пользователя.

7.3 Тестирование БД:

- проверка связей;
- проверка ограничений;
- проверка поведения при ошибках.

Система успешно прошла проверку на корректность и стабильность.

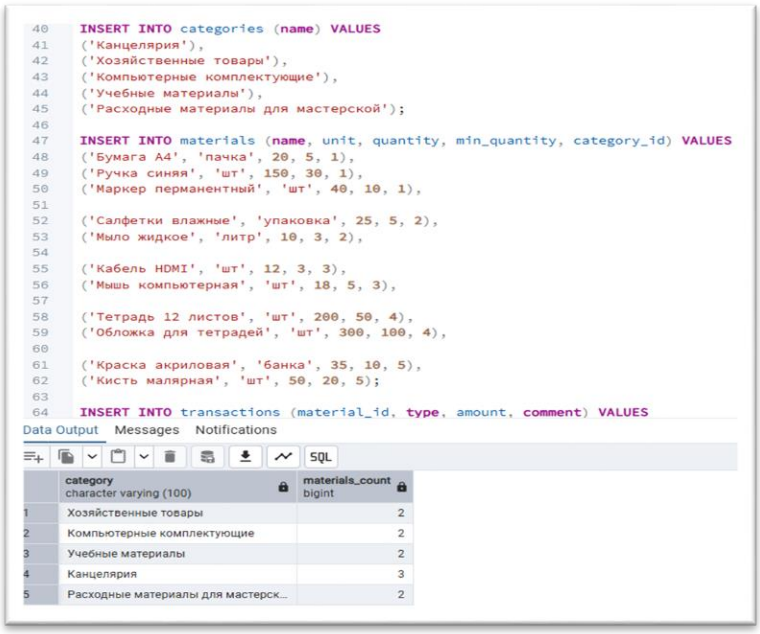

Рисунок 7

ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта было создано полноценное программное решение, позволяющее автоматизировать учёт расходных материалов и значительно упростить работу сотрудников. Система включает в себя структурированную базу данных на PostgreSQL, обеспечивающую надёжное хранение информации, а также удобный графический интерфейс на Python (Tkinter), который делает взаимодействие с данными интуитивно понятным даже для неподготовленного пользователя. Реализованный функционал охватывает операции добавления, редактирования, удаления, распределения и анализа материалов, что делает систему универсальной и практичной в реальном использовании.

По итогу разработка полностью соответствует требованиям модуля ПМ01 и ПМ02. Проект демонстрирует умение грамотно проектировать структуру БД, создавать и оптимизировать SQL-запросы, организовывать логику приложения, а также обеспечивать надёжность, целостность и безопасность данных. Особое внимание уделено устойчивой работе программы, обработке ошибок и корректной реализации всех предусмотренных функций.

Готовое решение обладает широкими возможностями для дальнейшего расширения: добавление новых модулей, интеграция с другими системами, улучшение интерфейса или подключение дополнительных инструментов аналитики. На текущем этапе проект полностью готов к внедрению и может использоваться в реальной организации как рабочая система учёта материалов, способная повысить эффективность и прозрачность внутренних процессов.

Литература

1.PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*. Retrieved from https://www.postgresql.org/docs/

2.Tkinter.ru. (2023). *Руководство по Tkinter*. Retrieved from https://tkinter.ru

3.Моргунов, Е. П. (2022). *PostgreSQL: основы языка SQL* (Учебное пособие). Санкт-Петербург: БХВ-Питер.

4.Cuantum Technologies LLC. (2023). *Python and SQL Bible*. O'Reilly Media.