

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н. П. ОГАРЁВА»

Институт электроники и светотехники  
Кафедра автоматизированных систем обработки информации и управления

## КУРСОВОЙ ПРОЕКТ

«Разработка системы управления хранилищем данных для  
систематизации информации о студентах и абитуриентах  
университета»

Автор курсового проекта \_\_\_\_\_ Васькин В.А.  
(подпись) (дата)

Направление подготовки 090301 Информатика и вычислительная техника

Обозначение курсового проекта КП-02069964-090301-02-20

Руководитель работы

К. Т. Н., \_\_\_\_\_ С.А. Ямашкин  
(подпись) (дата)

Оценка \_\_\_\_\_

Саранск  
2020

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н. П. ОГАРЁВА»

Институт электроники и светотехники  
Кафедра автоматизированных систем обработки информации и управления

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

Студент: Васькин Вадим Александрович

1 Тема: Разработка системы управления хранилищем данных для систематизации информации о студентах и абитуриентах университета.

2 Срок представления проекта к защите: \_\_\_\_\_

3 Исходные данные для научного исследования: учебные пособия,  
Интернет-источники

4 Изучение возможностей программы PostgreSQL для их заполнения из Telegram бота, наделение правами пользователей в зависимости от положения пользователя в университете.

Руководитель работы \_\_\_\_\_

подпись, дата

\_\_\_\_\_

инициалы, фамилия

Задание принял к исполнению \_\_\_\_\_

дата, подпись

## Реферат

Курсовой проект содержит 54 листа, 24 рисунка, 8 источников.

### ПРОГРАММИРОВАНИЕ, БАЗЫ ДАННЫХ, СУБД, TELEGRAM-BOT, PYTHON

Объектом разработки является программа PostgreSQL - свободная объектно-реляционная система управления базами данных (СУБД).

Цель проекта – изучение возможностей программы PostgreSQL для администрирования баз данных, их заполнение из Telegram бота, наделение правами пользователей в зависимости от положения пользователя в университете.

*Предмет исследования* – методы работы с базами данных.

*Методология и методы исследования.* При выполнении курсового проекта были использованы методы построения и администрирования баз данных.

*Практическая значимость.* В ходе выполнения работы были созданы базы данных, прописаны права пользователя и связаны с Telegram ботом.

Степень внедрения – будет востребована в заведениях, которым требуется заполнение баз данных, в частности Высшие учебные заведения для сбора информации о студентах.

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

|                                                  |         |          |       |      |                                                                                                                       |  |  |
|--------------------------------------------------|---------|----------|-------|------|-----------------------------------------------------------------------------------------------------------------------|--|--|
| Изм.                                             | Лист    | № докум. | Подп. | Дата | КР-02069964-090301-02-20                                                                                              |  |  |
| Разраб.                                          | Васькин |          |       |      | Разработка системы управления хранилищем данных для систематизации информации о студентах и абитуриентах университета |  |  |
| Проверил                                         | Ямашкин |          |       |      |                                                                                                                       |  |  |
| Реценз.                                          |         |          |       |      |                                                                                                                       |  |  |
| Н.контр.                                         |         |          |       |      |                                                                                                                       |  |  |
| Чт.Автора                                        |         |          |       |      |                                                                                                                       |  |  |
| Лит.                                             | Лист    | Листов   |       |      |                                                                                                                       |  |  |
|                                                  | 4       | 54       |       |      |                                                                                                                       |  |  |
| ФГБОУ ВО "МГУ им. Н. П. Огарёва" ИЭС, АСОИУ, 341 |         |          |       |      |                                                                                                                       |  |  |

## СОДЕРЖАНИЕ

|                                                                                                                               |    |
|-------------------------------------------------------------------------------------------------------------------------------|----|
| ВВЕДЕНИЕ .....                                                                                                                | 6  |
| 1 Анализ предметной области.....                                                                                              | 9  |
| 1.1 Системы управления базами данных .....                                                                                    | 9  |
| 1.2 Классификация СУБД.....                                                                                                   | 11 |
| 1.2.1 Классификация СУБД по модели данных. ....                                                                               | 12 |
| 1.2.2 Классификация СУБД по степени распределённости .....                                                                    | 12 |
| 1.2.3 Классификация СУБД по способу доступа к базам данным .....                                                              | 13 |
| 1.3 Администрирование СУБД .....                                                                                              | 13 |
| 1.3.1 Основные функции администратора .....                                                                                   | 13 |
| 2 СУБД PostgreSQL .....                                                                                                       | 16 |
| 2.1 Что такое PostgreSQL .....                                                                                                | 16 |
| 2.2 Преимущества PostgreSQL .....                                                                                             | 17 |
| 2.3 Основы работы с PostgreSQL .....                                                                                          | 19 |
| 3 Подготовка базы данных для хранения и редактирования информации о них, предназначенных для кураторов и преподавателей ..... | 24 |
| 3.1 Основная теория .....                                                                                                     | 24 |
| 3.1.1 Определение таблиц, которые должна содержать база данных .....                                                          | 24 |
| 3.1.2 Присвоение ключевых полей .....                                                                                         | 25 |
| 3.2 Структура базы данных, подготовленной для заполнения из Телеграмм – Бота .....                                            | 25 |
| 4 Связь системы управления баз данных с Telegram ботом .....                                                                  | 31 |
| 4.1 Python, как язык программирования .....                                                                                   | 31 |
| 4.2 Совместная работа Python и PostgreSQL .....                                                                               | 32 |
| 4.2.1 Подключение к базе данных .....                                                                                         | 32 |
| 4.2.2 Формирование запросов .....                                                                                             | 34 |
| 4.2.3 Передача данных с Телеграмм – бота в базу данных .....                                                                  | 35 |
| 5 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА. ....                                                                                      | 38 |

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 4    |



## ВВЕДЕНИЕ

Ежегодно кураторы первокурсников сталкиваются с следующей проблемой: сбор информации о первокурсниках, их данные, ссылки на социальные сети. Этой проблемы можно избежать посредством связки Telegram бота, из которого будут поступать данные и Баз данных, в которых эта информация будет доступна для хранения и редактирования.

**Кого затрагивает данная проблема:** абитуриенты высших учебных заведений и кураторов.

**Решение:** связка Telegram бота, из которого будут поступать данные и Баз данных, в которых эта информация будет доступна для хранения и редактирования.

**Актуальность** выбранной темы заключается в том, что на данный момент в современных системах управления базами данных имеется возможность не только хранить данные в своих структурах, но и хранить программный код, т.е. методы, с помощью которых происходит взаимодействие с потребителем или другим программно-аппаратным комплексом, а также проводить их администрирование, наделять правами пользователей, а на примере такой мощной системы как PostgreSQL можно показать актуальность и важность работы с базами данных.

Эти системы предоставляются для свободного использования в сфере образования, что делает их привлекательными и доступными для пользователей.

**Цель курсового проекта** – изучение возможностей программы PostgreSQL для администрирования баз данных, их заполнение из Telegram бота, наделение правами пользователей в зависимости от положения пользователя в университете.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | КР-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 6    |

**Для достижения поставленной цели необходимо решить ряд задач, а именно:**

- 1) Анализ предметной области.
- 2) Анализ существующих научных и практических решений в выбранной области, изучить методы, принципы и технологии работы с базами данных в программе PostgreSQL.
- 3) Подготовка базы данных абитуриентов для хранения и редактирования информации о них, предназначенных для кураторов и преподавателей.
- 4) Интеграция получившейся системы управления баз данных с Telegram ботом, с которого будет происходить прием информации, вывод информации в PDF или Word file.
- 5) Разработка программного продукта и тестирование с большим количеством пользователей.

Рассматриваемая тема отнюдь не новая, ибо администрирование баз данных сейчас очень популярно. Ценность же данного курсового проекта в том, что в итоге, соединив СУБД с интернет – ботом получится система, которая будет предназначена для сбора данных первокурсников для кураторов и преподавателей.

Практическая и теоретическая ценность данного исследования в том, что данное программное обеспечение может быть использовано ВУЗом для работы с первокурсниками и не только.

***Решение поставленных задач определило структуру курсового проекта.***

Структура курсовой работы состоит из введения, пяти глав, заключения и библиографического списка.

Во введении объясняется актуальность исследуемой темы, обозначаются объект, предмет, цели и задачи исследования.

Первая глава посвящена рассмотрению теоретических аспектов области, а именно СУБД, их классификация, права пользователя, администрирование и т.д.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата | Подп. и дата |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 7    |

Вторая глава посвящена рассмотрению существующих научных и практических решений в выбранной области, изучить методы, принципы и технологии работы с базами данных в программе PostgreSQL.

Третья глава посвящена практическому исследованию, а именно подготовке баз данных для хранения и редактирования информации.

Четвертая глава посвящена практическому исследованию, а именно связь получившейся системы управления баз данных с Telegram ботом, с которого будет происходить прием информации, вывод информации в PDF или Word file..

Пятая глава посвящена разработке программного продукта, а также проведению тестирования его с большим количеством пользователей.

В заключении отражены основные аспекты исследования и сделаны обобщающие выводы.

**Объект исследования** – программа PostgreSQL.

**Предмет исследования** – методы работы с базами данных.

**Методология и методы исследования.** При выполнении курсового проекта были использованы методы построения и администрирования баз данных.

**Практическая значимость.** В ходе выполнения работы были созданы базы данных, прописаны права пользователя и связаны с Telegram ботом.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата | Подп. и дата |
| Инв. № подл. | Подп. и дата |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | КР-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 8    |



# 1 Анализ предметной области

## 1.1 Системы управления базами данных

В современном мире информационные системы, поддерживающие концепцию интеграции данных, занимают огромные объёмы информации, сложно организованы, и при этом пытающиеся полностью удовлетворить потребности огромного количества пользователей. Данная глава посвящена теоретической части моей курсовой работы с целью сформировать представление о базах данных. В ней описано о том, что же такое Система управления базами данных, и то, что они «могут», об их возможностях и вариантах использования.

Обработка данных об объектах реального мира – вот основная задача любой информационной системы. В общепринятом понимании база данных – совокупность сведений о конкретных объектах реального мира, в какой либо предметной области.

Само же определение базы данных в нужном нам контексте звучит следующим образом. База данных – информационная модель, состоящая из данных, которые хранятся в памяти компьютера и связанных между собой по определенным признакам, которые определяют их взаимосвязь друг с другом, общие принципы манипулирования, хранения и описания. Сама по себе информационная модель – информация об объекте, которая определила его структуру в соответствии с заданной целью.

Если заглянуть в прошлое, то первые базы данных были созданы на основе файловых систем, и основным там было прикладное программное обеспечение, ибо вся ответственность возлагалась именно на них. Но время шло, и сейчас файловые базы данных практически не применяются, ибо сейчас его целиком и полностью вытеснило СУБД – система управления базами данных. Для ясности дадим определение данному термину.

|              |      |              |       |              |                          |              |  |              |      |
|--------------|------|--------------|-------|--------------|--------------------------|--------------|--|--------------|------|
| Подп. и дата |      | Инв. № дубл. |       | Взам. инв. № |                          | Подп. и дата |  | Инв. № подл. |      |
|              |      |              |       |              |                          |              |  |              |      |
| Изм.         | Лист | № докум.     | Подп. | Дата         | КР-02069964-090301-02-20 |              |  |              | Лист |
|              |      |              |       |              |                          |              |  |              | 9    |

Система управления базами данных (СУБД) – это языковые и программные средства, которые служат для того, чтобы создавать, вести и применять базы данных современных пользователей.

Сегодня саму структуру СУБД составляют не столько базы данных и языки программирования (ПО), сколько средства разработки и механизмы взаимодействия с пользователем на высоком уровне. Средства разработки, о которых идет речь, позволяют проектировать приложения, которые работают как одна из составляющих СУБД. На Рисунке 1 представлены основные определения, которые напрямую связаны с понятием систем управления базами данных.

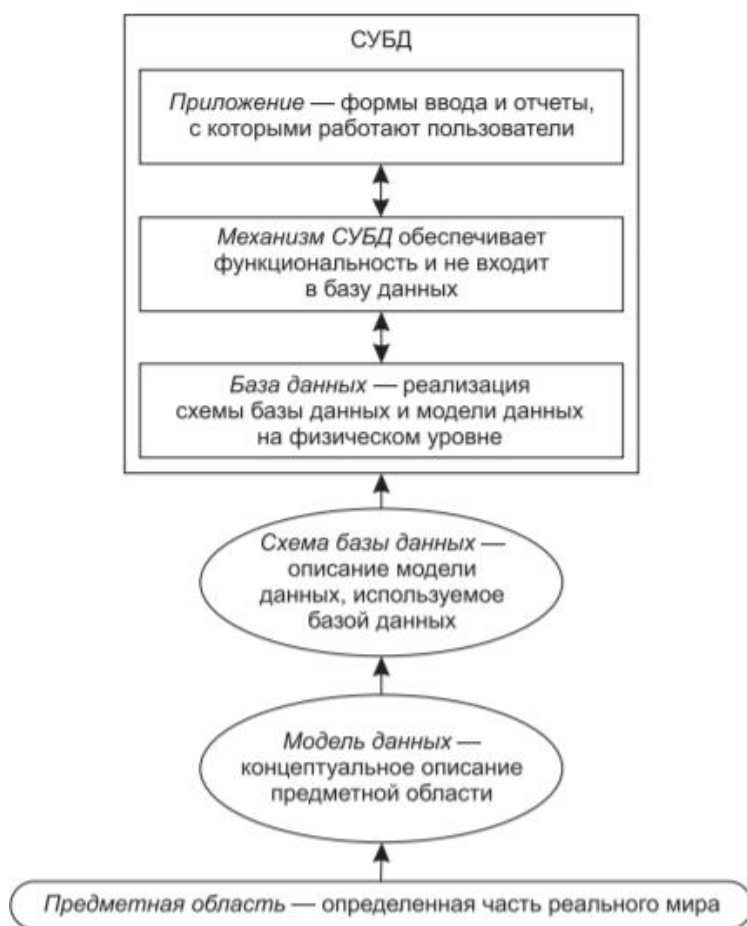


Рисунок 1 – Основные понятия СУБД

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата | Подп. и дата |

|      |      |          |       |      |
|------|------|----------|-------|------|
| Изм. | Лист | № докум. | Подп. | Дата |
|------|------|----------|-------|------|

KP-02069964-090301-02-20

Я посчитал необходимым вставить в этот раздел данные определения, потому что, на мой взгляд, они являются ключевыми при раскрытии понятия СУБД и его основ.

## 1.2 Классификация СУБД

С понятием баз данных и СУБД мы ознакомились, следующим подпунктом будет классификация баз данных. СУБД классифицируются по 4 основным направлениям:

- 1) по модели данных;
- 2) по степени распределённости;
- 3) по способу доступа к базам данных;
- 4) по степени универсальности;

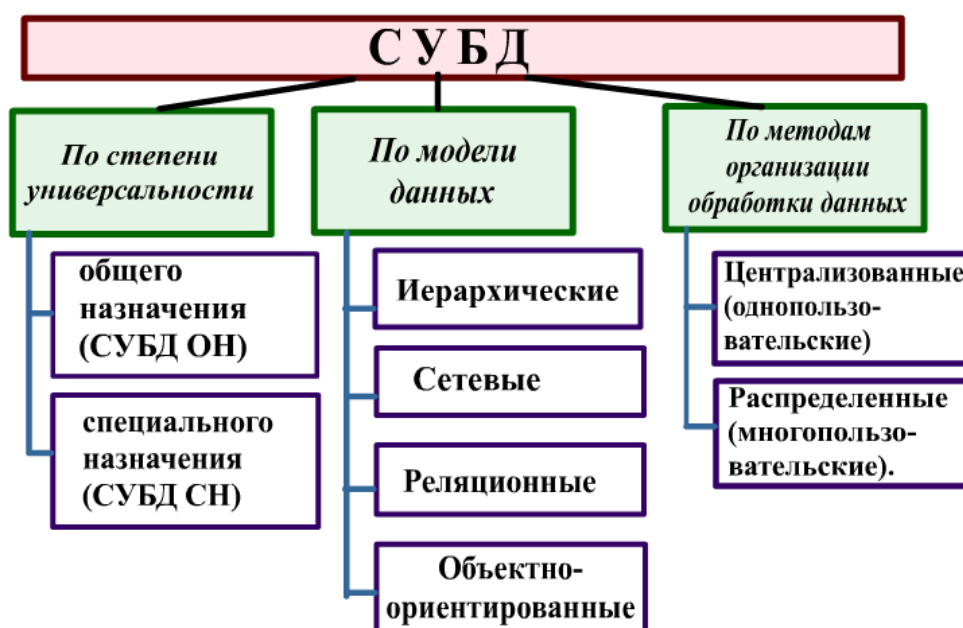


Рисунок 2 – Классификация СУБД

В следующих разделах будет подробно расписано каждое из направлений классификаций СУБД.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата | Инв. № дубл. |
| Изм.         | Лист         |
| № докум.     | Подп.        |
| Дата         |              |

### 1.2.1 Классификация СУБД по модели данных.

- 1) Иерархическая СУБД – это модель данных, в которой представление объектов имеет вид дерева, которые (объекты) находятся на различных уровнях.
- 2) Сетевая СУБД – система управления, при которой построение идёт на основе сетевой модели данных.
- 3) Реляционная СУБД – система управления реляционными базами данных, ориентированная на организацию данных в виде двумерных таблиц.
- 4) Объектно-ориентированная СУБД – система управления базами данных, которая основана на объектной модели данных, обрабатывающая данные как абстрактные объекты, наделённые свойствами.
- 5) Объектно-реляционная СУБД – система управления, которая поддерживает технологии, реализующие объектно-ориентированный подход: объекты, классы, наследование и т.д.

### 1.2.2 Классификация СУБД по степени распределённости

Локальные СУБД – система управления, при которой все части СУБД расположена на локальном сервере пользователя. Локальные СУБД вместе с базой данных располагаются на компьютере пользователя. Такие системы не зависят от сетевых процессов, поскольку управление базой данных выполняется автономно. Определённые достоинства, связанные в частности с безопасностью превращаются в очевидные недостатки: в случае работы с базой данных нескольких клиентов, ее редакцию приходится выполнять на каждом компьютере, при этом невозможно «стыковать» изменения, вносимые различными пользователем.

|              |              |              |              |              |      |      |          |       |      |                          |      |
|--------------|--------------|--------------|--------------|--------------|------|------|----------|-------|------|--------------------------|------|
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата | Изм. | Лист | № докум. | Подп. | Дата | КР-02069964-090301-02-20 | Лист |
|              |              |              |              |              |      |      |          |       |      |                          | 12   |

| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дудл. | Подп. и дата |
|--------------|--------------|--------------|--------------|--------------|
|              |              |              |              |              |



- сбор информации о первокурсниках путем сбора их в аудитории.  
Работа с раздаточным материалом, бланками.

На мой взгляд, эти решения достаточно устарели, и то, что предложено мной в курсовом проекте достаточно ново и инновационно. Для достижения поставленной задачи требуется изучить методы, принципы и технологии работы с базами данных в программе PostgreSQL.

|              |              |          |       |      |                          |              |              |              |
|--------------|--------------|----------|-------|------|--------------------------|--------------|--------------|--------------|
| Инв. № подл. | Подп. и дата |          |       |      | Инв. № дудл.             | Взам. инв. № | Подп. и дата | Инв. № подл. |
|              |              |          |       |      |                          |              |              |              |
| Изм.         | Лист         | № докум. | Подп. | Дата | КР-02069964-090301-02-20 |              |              |              |
|              |              |          |       |      |                          |              |              |              |
|              |              |          |       |      | Лист 15                  |              |              |              |

## 2 СУБД PostgreSQL

### 2.1 Что такое PostgreSQL

PostgreSQL – это объектно-реляционная система управления базами данных (ОРСУБД, ORDBMS), основанная на POSTGRES, Version 4.2 – программе, разработанной на факультете компьютерных наук Калифорнийского университета в Беркли. В POSTGRES появилось множество новшеств, которые были реализованы в некоторых коммерческих СУБД гораздо позднее.

PostgreSQL – система управления базами данных с открытым исходным кодом, которая была разработана в Беркли. Данная СУБД полностью поддерживает стандарт SQL. PostgreSQL, как СУБД, обладает следующими качествами:

- 1) Надежность – полное соответствие принципам ACID – атомарность, непротиворечивость, изолированность, сохранность данных.
- 2) Производительность – данное СУБД основывается на использование индексов, блокировке системы, отличной масштабируемости при конкурентной работе.
- 3) Расширяемость – настройка системы пользователем путём определения новых функций, типов языка, индексов и операторов.

Она поддерживает большую часть стандарта SQL и предлагает множество современных функций:

- сложные запросы;
- внешние ключи;
- триггеры;
- изменяемые представления;
- транзакционная целостность;
- многоверсионность.

Кроме того, пользователи могут всячески расширять возможности PostgreSQL, например, создавая свои:

|              |              |              |              |              |      |      |          |       |      |                          |      |
|--------------|--------------|--------------|--------------|--------------|------|------|----------|-------|------|--------------------------|------|
| Инв. № подл. | Подл. и дата | Взам. инв. № | Инв. № дубл. | Подл. и дата | Изм. | Лист | № докум. | Подп. | Дата | КР-02069964-090301-02-20 | Лист |
|              |              |              |              |              |      |      |          |       |      |                          | 16   |
|              |              |              |              |              |      |      |          |       |      |                          |      |



- типы данных;
- функции;
- операторы;
- агрегатные функции;
- методы индексирования;
- процедурные языки.

А благодаря свободной лицензии, PostgreSQL разрешается бесплатно использовать, изменять и распространять всем и для любых целей – личных, коммерческих или учебных.

## 2.2 Преимущества PostgreSQL

Но почему PostgreSQL? Ведь существует большинство реляционных данных с открытым кодом (Например: MySQL, MariaDB и Firebird). Что же данная СУБД может предложить, чего нет в остальных? Представлю список преимуществ PostgreSQL, относительно других СУБД, которые лично мне показались решающими и определяющими в выборе программы для разработки базы данных.

### 1) Модель данных

PostgreSQL – объектно-реляционная СУБД, что дает ей преимущество над остальными SQL базами данных с открытым исходным кодом. Преимущества объектно – реляционных СУБД – поддержка пользовательских объектов и их поведения, типы данных, функции, операции, домены и индексы. Эти качества показывают Postgres со стороны, как надёжное и гибкое СУБД, которое дает пользователям большую свободу. Кроме того в его функциях есть создание, хранение и извлечение сложных структур данных.

### 2) Структуры и типы данных

|              |              |          |       |      |                          |      |
|--------------|--------------|----------|-------|------|--------------------------|------|
| Инв. № подл. | Подп. и дата |          |       |      | КР-02069964-090301-02-20 | Лист |
|              | Взам. инв. № |          |       |      |                          | 17   |
|              | Инв. № дубл. |          |       |      |                          |      |
|              | Подп. и дата |          |       |      |                          |      |
| Изм.         | Лист         | № докум. | Подп. | Дата |                          |      |

Огромный список типов данных, которые поддерживает PostgreSQL. Кроме числовых, с плавающей точкой, текстовых, булевых и других ожидаемых типов данных (а также множества их вариаций), PostgreSQL может похвастаться поддержкой uuid, денежного, перечисляемого, геометрического, бинарного типов, сетевых адресов, битовых строк, текстового поиска, xml, json, массивов, композитных типов и диапазонов, а также некоторых внутренних типов для идентификации объектов и местоположения логов. PostgreSQL обеспечивает хранение разных типов сетевых адресов. Тип данных CIDR (бесклассовая маршрутизация интернет домена, Classless Internet Domain Routing) следует соглашению для сетевых адресов IPv4.

### 3) Многомерные массивы

Поскольку Постгрес – это объектно-реляционная база данных, массивы значений могут храниться для большинства существующих типов данных. Сделать это можно путём добавления квадратных скобок к спецификации типа данных для столбца или с помощью выражения ARRAY. Размер массива может быть задан, но это необязательно.

### 4) Геометрические данные

Геоданные быстро становятся основным требованием для многих приложений. PostgreSQL уже давно поддерживает множество геометрических типов данных, таких как точки, линии, круги и многоугольники. Один из этих типов – PATH, он состоит из множества последовательно расположенных точек и может быть открытым (начальная и конечная точки не связаны) или закрытым (начальная и конечная точки связаны). Давайте рассмотрим в качестве примера туристическую тропу. В данном случае туристическая тропа – это петля, поэтому начальная и конечная точки связаны, и, значит, мой путь является закрытым. Круглые скобки вокруг набора координат указывают на закрытый путь, а квадратные – на открытый.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | КР-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 18   |

| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дудл. | Подп. и дата |
|--------------|--------------|--------------|--------------|--------------|
|              |              |              |              |              |

SQL-функции выполняют произвольный список операторов SQL и возвращают результат последнего запроса в списке. В простом случае (не с множеством) будет возвращена первая строка результата последнего запроса. (Помните, что понятие "первая строка" в наборе результатов с несколькими строками определено точно, только если присутствует ORDER BY). Если последний запрос вообще не вернёт строки, будет возвращено значение NULL.

Тело SQL-функции должно представлять собой список SQL-операторов, разделённых точкой с запятой. Точка с запятой после последнего оператора может отсутствовать. Если только функция не объявлена как возвращающая void, последним оператором должен быть SELECT, либо INSERT, UPDATE или DELETE с предложением RETURNING.

KP-02069964-090301-02-20

`\connect db_name` – подключиться к базе с именем `db_name`  
`\du` – список пользователей  
`\dp` (или `\z`) – список таблиц, представлений, последовательностей, прав доступа к ним  
`\di` – индексы  
`\ds` – последовательности  
`\dt` – список таблиц  
`\dt+` – список всех таблиц с описанием  
`\dt *s*` – список всех таблиц, содержащих `s` в имени  
`\dv` – представления  
`\dS` – системные таблицы  
`\d+` – описание таблицы  
`\o` – пересылка результатов запроса в файл  
`\l` – список баз данных  
`\i` – читать входящие данные из файла  
`\e` – открывает текущее содержимое буфера запроса в редакторе (если иное не указано в окружении переменной `EDITOR`, то будет использоваться по умолчанию `vi`)  
`\d "table_name"` – описание таблицы  
`\i` запуск команды из внешнего файла, например `\i /my/directory/my.sql`  
`\pset` – команда настройки параметров форматирования  
`\echo` – выводит сообщение  
`\set` – устанавливает значение переменной среды. Без параметров выводит список текущих переменных (`\unset` – удаляет)  
`\?` – справочник `psql`  
`\help` – справочник `SQL`  
`\q` (или `Ctrl+D`) – выход с программы

На следующих скриншотах экрана представлены примеры реализации

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

|      |      |          |       |      |
|------|------|----------|-------|------|
|      |      |          |       |      |
| Изм. | Лист | № докум. | Подп. | Дата |

KP-02069964-090301-02-20

некоторых функций консоли Postgres – SQL SHALL. На рисунке 3 изображено подключение к локальному серверу, которое происходит путём проверки пользователя на подлинность.

Рисунок 3 – Подключение к локальному серверу

На рисунке 4 изображено подключение к базе данных, находящейся на сервере, к которому мы подключились на предыдущем рисунке, данное действие выполняет команда \с.

Рисунок 4 – Подключение к базе данных с помощью команды \с

На рисунке 5 изображен вывод информации о таблице students, обо всех переменных которые находятся в его структуре, а также информация о ключах, которыми таблицы связаны между собой. Данное действие выполняет команда \с.

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

```

Вы подключены к базе данных "kurs" как пользователь "postgres".
kurs=# \d students
                                Таблица "public.students"
Столбец |          Тип          | Правило сортировки | Допустимость NULL |      По умолчанию
-----+-----+-----+-----+-----
id       | integer               |                     | not null           | nextval('students_id_seq'::regclass)
name     | character varying(255)|                     |                     |
phone    | character varying(20) |                     |                     |
email    | character varying(255)|                     |                     |
vk_acc   | character varying(255)|                     |                     |
groop_id | integer               |                     |                     |
Индексы:
    "students_pkey" PRIMARY KEY, btree (id)
Ограничения внешнего ключа:
    "students_groop_id_fkey" FOREIGN KEY (groop_id) REFERENCES groop(id)

```

Рисунок 5 – Вывод информации о таблице

На рисунке 6 изображен вывод заполненной таблицы students. Данное действие выполняет команда select. После данной команды пишется идентификатор формата, который определяет тип выводимой информации. В данном случае “\*” после команды select выводит всю информацию, которая там содержится.

```

kurs=# select * from students;
 id |      name      | phone |      email      | vk_acc | groop_id
----+-----+-----+-----+-----+-----
  1 | Vaskin Vadim Aleksandrovic | 01    | vaskin@mail.ru  | id4353245 | 1
  2 | Likanina Dasha          | 02    | vaasdf@mail.ru  | id124154345 | 2
  3 | Грек Андрей Михайлович   |       |                  |          |
(3 строки)

```

Рисунок 6 – Вывод информации таблицы

На рисунке 7 изображено заполнение таблицы. Это происходит с помощью запроса insert, после которой вводится название таблицы, имена переменных, в которые необходимо записать информацию, после чего вводится информация, записывающаяся в переменные.

```

kurs=# insert into students (name) values ('Сусленкова Анастасия Михайловна');
INSERT 0 1

```

Рисунок 7 – Заполнение таблицы

Это лишь малая часть тех функций, которыми обладает данная объектно-реляционная система управления базами данных, но они не могли быть не представлены здесь в качестве примера.

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

| Инб. № подл. | Подп. и дата | Взам. инб. № | Инб. № дубл. | Подп. и дата |
|--------------|--------------|--------------|--------------|--------------|
|              |              |              |              |              |

### 3 Подготовка базы данных для хранения и редактирования информации о них, предназначенных для кураторов и преподавателей

#### 3.1 Основная теория

На первом этапе проектирования базы данных необходимо определить цель создания базы данных, основные ее функции и информацию, которую она должна содержать. Моя база данных разработана для кураторов учебных заведений с целью систематизации информации о студентах и абитуриентах университета. Схема работы моей базы данных очень проста, у нас имеется таблица со студентами, каждый из которых в свою очередь относится к таблице с группами, каждая из групп которых относится к институту. То есть фактически будут использоваться в основном 3 таблицы – Студенты, группы и институты. После заполнения базы данных, проводимая с телеграмм Бота, будет возможность вывести табличку в CSV файл, с которым впоследствии будут работать кураторы.

##### 3.1.1 Определение таблиц, которые должна содержать база данных

Один из наиболее сложных этапов в процессе создания базы данных – разработка таблиц, так как результаты, которые должна выдавать база данных не всегда дают полное представление о структуре таблицы. Таблицы должны содержать всю информацию разрабатываемой базы. В моем случае это Студенты, с определенной группой и институтом и информация о них (ФИО, номер телефона, электронная почта и т.д.). Все таблицы хранят максимально полную характеристику, информацию и описание для дальнейшей успешной работы с базой данных.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | КР-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 24   |



### 3.1.2 Присвоение ключевых полей

Для связи данных из разных таблиц, например, данные о студенте и группе, в которой он учится, каждая таблица должна содержать набор полей или поле, где будет задаваться индивидуальное значение каждой записи в таблице. Такое поле или набор полей называют основным ключом. Именно благодаря ключам будет функционировать база данных, сопоставляя, связывая и формируя информацию из разных таблиц. Количество ключей варьируется от одного до нескольких. Вообще, ключ – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

### 3.2 Структура базы данных, подготовленной для заполнения из Телеграмм – Бота

База Данных «kurs», предназначенная для заполнения содержит 3 таблицы:

- студенты;
- институты;
- группы студентов.

Во всех таблицах указываются первичные или внешние ключи.

**Таблица Students:** предназначена для хранения полной информации о всех первокурсниках, заполнивших базу данных. Например, номер телефона, электронная почта и т.д.

Для получения информации о конкретном объекте таблицы или представления служит команда \d. Она показывает следующие характеристики:

- список столбцов вместе с их типами и значениями по умолчанию;
- индексы;
- ограничения;
- внешние ключи;

|              |      |              |       |              |                          |              |  |              |      |
|--------------|------|--------------|-------|--------------|--------------------------|--------------|--|--------------|------|
| Подп. и дата |      | Инв. № дубл. |       | Взам. инв. № |                          | Подп. и дата |  | Инв. № подл. |      |
|              |      |              |       |              |                          |              |  |              |      |
| Изм.         | Лист | № докум.     | Подп. | Дата         | КР-02069964-090301-02-20 |              |  |              | Лист |
|              |      |              |       |              |                          |              |  |              | 25   |

– триггеры;

С помощью данной команды, выведем все объекты таблицы Students.  
Результат выполнения этого запроса представлен на Рисунке 8:

| Таблица "public.students"                                            |                        |                    |                   |                                      |
|----------------------------------------------------------------------|------------------------|--------------------|-------------------|--------------------------------------|
| Столбец                                                              | Тип                    | Правило сортировки | Допустимость NULL | По умолчанию                         |
| id                                                                   | integer                |                    | not null          | nextval('students_id_seq'::regclass) |
| name                                                                 | character varying(255) |                    |                   |                                      |
| phone                                                                | character varying(20)  |                    |                   |                                      |
| email                                                                | character varying(255) |                    |                   |                                      |
| vk_acc                                                               | character varying(255) |                    |                   |                                      |
| groop_id                                                             | integer                |                    |                   |                                      |
| Индексы:                                                             |                        |                    |                   |                                      |
| "students_pkey" PRIMARY KEY, btree (id)                              |                        |                    |                   |                                      |
| Ограничения внешнего ключа:                                          |                        |                    |                   |                                      |
| "students_groop_id_fkey" FOREIGN KEY (groop_id) REFERENCES groop(id) |                        |                    |                   |                                      |

Рисунок 8 – объекты таблицы Students

Далее распишем каждое поле таблицы, и за что оно отвечает.

*id* – порядковый номер студента. Поле является счетчиком. Данное поле заполняется автоматически.

*name* – ФИО студента. Поле заполняется из Телеграмм Бота.

*phone* – телефонный номер студента. Поле заполняется из Телеграмм Бота.

*email* – адрес электронной почты студента. Поле заполняется из Телеграмм Бота.

*vk\_acc*– аккаунт социальной сети ВК. Поле заполняется из Телеграмм Бота.

*groop\_id* – внешний ключ, соединяющий *id* каждого студента с номером определенной группы (каждый студент принадлежит определенной группе). Поле заполняется из Телеграмм Бота.

Вывод всех элементов заполненной таблицы в SQL Shall выглядит следующим образом. Мы используем команду Select \* (выбрать и вывести все элементы таблички) для вывода их в окно консоли. Результат выполнения запроса, а именно, заполненная данными пользователя табличка представлена на Рисунке 9:

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
|      |      |          |       |      | КР-02069964-090301-02-20 | Лист |
| Изм. | Лист | № докум. | Подп. | Дата |                          | 26   |

```
kurs=# select * from students;
```

| id | name                            | phone  | email          | vk_acc      | groop_id |
|----|---------------------------------|--------|----------------|-------------|----------|
| 1  | Vaskin Vadim Aleksandrovic      | 01     | vaskin@mail.ru | id4353245   | 1        |
| 2  | Likanina Dasha                  | 02     | vaasdf@mail.ru | id124154345 | 2        |
| 3  | Грек Андрей Михайлович          |        |                |             |          |
| 4  | Сусленкова Анастасия Михайловна | 890223 |                |             |          |

(4 строки)

Рисунок 9 – вывод всех элементов таблицы Students

**Таблица Groop:** предназначена для хранения полной информации о всех группах, которые имеются в Высшем учебном заведении. А также все группы с помощью первичных ключей привязаны к институту.

Для получения информации о конкретном объекте таблицы или представления служит команда \d.

С помощью данной команды, выведем все объекты таблицы Groops. Результат запроса представлен на Рисунке 10:

```
kurs=# \d groop
```

| Столбец            | Тип                    | Правило сортировки | Допустимость NULL | По умолчанию                      |
|--------------------|------------------------|--------------------|-------------------|-----------------------------------|
| id                 | integer                |                    | not null          | nextval('groop_id_seq'::regclass) |
| name               | character varying(255) |                    |                   |                                   |
| number_of_students | integer                |                    |                   |                                   |
| institute_id       | integer                |                    |                   |                                   |

Индексы:  
 "groop\_pkey" PRIMARY KEY, btree (id)  
 Ограничения внешнего ключа:  
 "groop\_institute\_id\_fkey" FOREIGN KEY (institute\_id) REFERENCES institute(id)  
 Ссылки извне:  
 TABLE "students" CONSTRAINT "students\_groop\_id\_fkey" FOREIGN KEY (groop\_id) REFERENCES groop(id)

Рисунок 10 – объекты таблицы Groop

Далее распишем каждое поле таблицы и за что оно отвечает.

*id* – порядковый номер группы. Поле является счетчиком. Данное поле заполняется автоматически.

*name* – наименование группы. Поле заполняется сразу, ибо количество групп фиксировано.

*number\_of\_students* – количество студентов той или иной группы, которое подсчитывается путём id\_students.

*institute\_id* – внешний ключ, соединяющий id каждой группы с номером определенного института (каждая группа принадлежит определенному институту). Поле заполняется из Телеграмм Бота.

Вывод всех элементов заполненной таблицы в SQL Shall выглядит следующим образом. Мы используем команду Select \* (выбрать и вывести все элементы таблички) для вывода их в окно консоли. Результат выполнения запроса представлен на Рисунке 11.

```
kurs=# select * from groop;
 id | name | number_of_students | institute_id
-----+-----+-----+-----
  1 | 141  |                    |           1
  2 | 111  |                    |           1
  3 | 121  |                    |           1
  4 | 131  |                    |           1
  5 | 161  |                    |           1
  6 | 171  |                    |           1
  7 | 181  |                    |           1
  8 | 191  |                    |           1
  9 | 112  |                    |           1
 10 | 151  |                    |           1
(10 строк)
```

Рисунок 11 – вывод всех элементов таблицы Groop

**Таблица Institute:** предназначена для хранения полной информации о всех институтах, которые имеются в Высшем учебном заведении. Таблицы связаны таким образом, что каждый институт имеет группы, в которых, в свою очередь, входят студенты. Всё это выполнено с помощью внешнего ключа.

Для получения информации о конкретном объекте таблицы или представления служит команда \d.

С помощью данной команды, выведем все объекты таблицы institute. (Рисунок 12).

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

```
kurs=# \d institute
```

| Столбец        | Тип                    | Правило сортировки | Допустимость NULL | По умолчанию                          |
|----------------|------------------------|--------------------|-------------------|---------------------------------------|
| id             | integer                |                    | not null          | nextval('institute_id_seq'::regclass) |
| name           | character varying(255) |                    |                   |                                       |
| number_of_spec | integer                |                    |                   |                                       |

Индексы:  
 "institute\_pkey" PRIMARY KEY, btree (id)

Ссылки извне:  
 TABLE "groop" CONSTRAINT "groop\_institute\_id\_fkey" FOREIGN KEY (institute\_id) REFERENCES institute(id)

Рисунок 12 – объекты таблицы Institute

Далее распишу каждое поле таблицы и за что оно отвечает.

*id* – порядковый номер института. Поле является счетчиком. Данное поле заполняется автоматически.

*name* – наименование группы. Поле заполняется сразу, ибо количество групп фиксировано и известно заранее.

*number\_of spec* – количество групп в том или ином институте, которое подсчитывается путём id\_groop.

Вывод всех элементов заполненной таблицы в SQL Shall выглядит следующим образом. Мы используем команду Select \* (выбрать и вывести все элементы таблички) для вывода их в окно консоли.

| id         | name | number_of_spec |
|------------|------|----------------|
| 1          | IES  | 9              |
| 2          | IFH  | 9              |
| 3          | ASF  | 9              |
| (3 строки) |      |                |

Рисунок 13 – вывод всех элементов таблицы Institute

Для того, чтобы в PostgreSQL сделать отчёт о проделанной работе, воспользуемся SQL – запросом COPY\_FROM, с помощью которого, есть возможность вывести заполненную с Телеграмм бота базу данных в CSV файл.

Формирование запроса для заполнения CSV – файла данными таблицы представлено на следующем фрагменте программного кода:

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
|      |      |          |       |      | КР-02069964-090301-02-20 | Лист |
| Изм. | Лист | № докум. | Подп. | Дата |                          | 29   |

## Листинг программы

```
kurs=# COPY students TO 'C:/FFOutput/kursach.csv' DELIMITER ',' CSV
HEADER;
```

Как можно заметить, происходит копирование данных из таблицы students в файл по пути 'C:/FFOutput/kursach.csv' с запятой в виде разделителя. Содержимое CSV – файла kursach.csv представлено на Рисунке 14

The image shows the Microsoft Excel interface. The ribbon at the top is set to 'Главная' (Home). The spreadsheet data is as follows:

|   | A  | B                               | C      | D              | E           | F        |
|---|----|---------------------------------|--------|----------------|-------------|----------|
| 1 | id | name                            | phone  | email          | vk_acc      | groop_id |
| 2 | 1  | Vaskin Vadim Aleksandrovic      |        | vaskin@mail.ru | id4353245   | 1        |
| 3 | 2  | Likanina Dasha                  |        | vaasdf@mail.ru | id124154345 | 2        |
| 4 | 3  | Грек Андрей Михайлович          |        |                |             |          |
| 5 | 4  | Сусленкова Анастасия Михайловна | 890223 |                |             |          |
| 6 |    |                                 |        |                |             |          |

Рисунок 14 – таблица в CSV формате

## 4 Связь системы управления баз данных с Telegram ботом

### 4.1 Python, как язык программирования

Все приложения взаимодействуют с данными, чаще всего через СУБД (систему управления базами данных). В одних языках программирования существуют модули для работы с СУБД, для других же существуют сторонние пакеты или же библиотеки. Подобная библиотека имеется и в языке программирования Python. Кратко расскажу об этом языке программирования.

Python – высокоуровневый язык программирования. Он поддерживает несколько парадигм программирования: структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Программный код организовывается в функции и классы, объединяющиеся в модули. Python – язык очень универсальный. Он подходит для многих задач, для которых используются библиотеки. Вот несколько примеров популярных библиотек для Python:

- **Pygame** – Библиотека для создания небольших игр и мультимедийных приложений.
- **NumPy** – Библиотека для работы с искусственным интеллектом и машинным обучением. Используется для сложных математических вычислений.
- **Pandas** – Библиотека для работы с большими данными.
- **SQLAlchemy** – Библиотека для работы с базами данных.
- **Django, Flask** – Библиотеки для разработки серверной части приложений.

Наличие готовых библиотек обозначает, что под каждую задачу программиста есть уже готовое решение, и что – то придумывать с нуля не придётся.

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

|      |      |          |       |      |
|------|------|----------|-------|------|
|      |      |          |       |      |
| Изм. | Лист | № докум. | Подп. | Дата |

KP-02069964-090301-02-20

Но нашей задачей является связать нашу базу данных и Телеграмм – бот на Python. Данная связка будет происходить в несколько этапов, которые будут расписаны мной далее.

## 4.2 Совместная работа Python и PostgreSQL

### 4.2.1 Подключение к базе данных

Прежде чем взаимодействовать с любой базой данных через SQL-библиотеку, с ней необходимо связаться. В этом разделе будет расписано, как подключиться из приложения Python к базам данных PostgreSQL. Первым делом нужно самостоятельно запустить сервер, на котором будет храниться наша база данных. В отличие от SQLite, в Python по умолчанию нет модуля, который можно использовать для подключения к базе данных PostgreSQL. Само же формирование базы данных состоит из двух этапов:

- 1) Установка соединения с сервером PostgreSQL.
- 2) Выполнение запроса для создания БД.

Для того чтоб взаимодействовать с бд есть решение – модуль psycopg2. Он появился в Python версии 2.7, его сопровождал модуль sql. Его цель — упростить и обезопасить работу при формировании динамических запросов. На следующем фрагменте кода представлено вариант подключения базы данных к Python:

#### Листинг программы.

```
def create_connection(db_name, db_user, db_password, db_host, db_port):
    connection = None
    try:
        connection = psycopg2.connect(
            database=db_name,
            user=db_user,
            password=db_password,
            host=db_host,
```

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 32   |



| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дудл. | Подп. и дата |
|--------------|--------------|--------------|--------------|--------------|
|              |              |              |              |              |

- 1) db\_name – название базы данных
- 2) db\_user – имя пользователя
- 3) db\_password – пароль пользователя
- 4) db\_host – хост
- 5) db\_port – порт

### Листинг Программы:

В квадратных кавычках прописываются переменные, соответствующие параметрам базы данных, необходимых для подключения и, соответственно, работы с ней.

### 4.2.2 Формирование запросов

Для выполнения запросов в Postgres используется метод `cursor.execute()`. В этом разделе мы определим функцию `execute_query()`, которая использует этот метод. Функция будет принимать объект `connection` и строку запроса. Далее строка запроса будет передаваться методу `execute()`. В этом разделе он будет использоваться для формирования таблиц, а в следующих – мы применим его для выполнения запросов на обновление и удаление.

Итак, начнем с определения функции `execute_query()`. Ее реализация написана в Листинге программы, представленной ниже:

#### Листинг программы.

```
def execute_query(connection, query):
    connection.autocommit = True
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        print("Query executed successfully")
    except OperationalError as e:
        print(f"The error '{e}' occurred")
```

Мы можем использовать эту функцию для организации таблиц, вставки, изменения и удаления записей в нашей базе данных PostgreSQL. Это можно показать на примере запроса на создание таблицы в базе данных (создадим таблицу `users`). На следующем фрагменте кода представлена её реализация.

#### Листинг программы.

```
create_users_table = """
CREATE TABLE IF NOT EXISTS users (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL,
    age INTEGER,
```

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 34   |

```

gender TEXT,
nationality TEXT
)
"""
execute_query(connection, create_users_table)

```

Другим немаловажным аспектом является указание столбцов с автоматическим инкрементом. Для введения их в программу используется ключевое слово **SERIAL**. Пример указания ссылок на внешние ключи представлен в следующем листинге программного кода:

#### Листинг программы.

```

create_posts_table = """
CREATE TABLE IF NOT EXISTS posts (
    id SERIAL PRIMARY KEY,
    title TEXT NOT NULL,
    description TEXT NOT NULL,
    user_id INTEGER REFERENCES users(id)
)
"""
execute_query(connection, create_posts_table)

```

Фрагмент кода, представленный выше иллюстрирует создание таблицы базы данных, в которой один из элементов – столбец с автоматическим инкрементом.

### 4.2.3 Передача данных с Телеграмм – бота в базу данных

Для того, чтобы реализовать передачу данных из языка программирования в базу **данных** используется подход, использующий метод `cursor.executemany()`. Он принимает два параметра:

- строка query, содержащая заполнители для вставляемых записей;

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 35   |

- список записей, которые мы хотим вставить.

Передается SQL-запрос с заполнителями и списком записей методу `execute()`. Каждая запись в списке должна являться кортежем, значения которого соответствуют значениям столбца в таблице БД. На следующем фрагменте кода представлена пример вставки пользовательских записей в таблицу `users`:

#### Листинг программы.

```
users = [
    ("James", 25, "male", "USA"),
    ("Leila", 32, "female", "France"),
    ("Brigitte", 35, "female", "England"),
    ("Mike", 40, "male", "Denmark"),
    ("Elizabeth", 21, "female", "Canada"),
]

user_records = ", ".join(["%s"] * len(users))

insert_query = (
    f"INSERT INTO users (name, age, gender, nationality) VALUES
    {user_records}"
)

connection.autocommit = True
cursor = connection.cursor()
cursor.execute(insert_query, users)
```

Список `users` содержит пять пользовательских записей в виде кортежей. Затем мы создаём строку с пятью элементами-заполнителями (`%s`), соответствующими пяти пользовательским записям. Строка-заполнитель объединяется с запросом, который вставляет записи в таблицу `users`. Наконец, строка запроса и пользовательские записи передаются в метод `execute()`. Наглядно мы можем это наблюдать на следующем фрагменте программного кода:

|              |              |
|--------------|--------------|
| Инв. № подл. | Подл. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подл. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 36   |

## Листинг программы.

```
posts = [
    ("Happy", "I am feeling very happy today", 1),
    ("Hot Weather", "The weather is very hot today", 2),
    ("Help", "I need some help with my work", 2),
    ("Great News", "I am getting married", 1),
    ("Interesting Game", "It was a fantastic game of tennis", 5),
    ("Party", "Anyone up for a late-night party today?", 3),
]

post_records = ", ".join(["%s"] * len(posts))

insert_query = (
    f"INSERT INTO posts (title, description, user_id) VALUES
    {post_records}"
)

connection.autocommit = True
cursor = connection.cursor()
cursor.execute(insert_query, posts)
```

Выполнение данного кода приведет к выполнению запроса, который отвечает за вставку введенной информации в базу данных.

Изучив весь данный материал можно сделать вывод, что Python, как язык программирования очень удобен для работы с базами данных, в нашем случае, с PostgreSQL. Удобство же проявляется в том, что в языке есть специальные модули, позволяющие достаточно просто подключаться к базам данных и выполнять запросы, связанные с ней. Из этого можно сделать вывод, что при разработке данного курсового проекта язык для объединения с базами данных был выбран правильно.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 37   |

## 5 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА.

### 5.1 Запросы и функции, формирующие структуру продукта

Программирование ведется целиком на языке программирования Python. Далее приведены фрагменты основного кода программного продукта с их описанием. Во фрагменте кода, приведенном ниже, происходит получение id пользователя, который в данный момент пользуется Telegram ботом.

#### Листинг программы

```
cur = con.cursor()
cur.execute(
    "INSERT INTO students (id) VALUES ({}).format(user_id)
)
con.commit()
```

После присвоения пользователю id необходимым условием является подключение к базе данных. Реализация данного действия представлена на следующем листинге:

#### Листинг программы

```
con = psycopg2.connect(
    database="kursach",
    user="admin",
    password=" ",
    host="127.0.0.1",
    port="5433"
)
```

В данном фрагменте кода `con = psycopg2.connect` – операнд для взаимодействия с базой данных.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |
|------|------|----------|-------|------|
| Изм. | Лист | № докум. | Подп. | Дата |
|      |      |          |       |      |

KP-02069964-090301-02-20

Лист  
38

Далее следуют характеристические переменные подключения.

- database=" " – имя базы данных.
- user=" " – имя пользователя.
- password=" " – пароль.
- host=" " – хост.
- port=" " – порт.

После чего происходит присвоение пользователю его личного id и занесение его в базу.

В данном фрагменте кода используются следующие методы.

- cur = con.cursor() – Метод для выполнения запросов PostgreSQL.
- con.commit() – Метод для сохранения изменений в базе данных.

Далее пользователь заполняет данные, предлагаемые Telegram ботом, о себе. Для начала бот запрашивает ввести у пользователя его инициалы. Для этого вновь происходит заход в базу данных, после чего происходит запрос в базу данных, вследствие которого происходит запись. Осуществляется она посредством SQL – запроса UPDATE. В следующем фрагменте кода представлено заполнение имени пользователя путём SQL запроса.

#### Листинг программы

```
cur = con.cursor()
cur.execute(
    "update students set {}='{}' where id =
    {}".format("name",str(message.text), message.from_user.id)
)
con.commit()
```

Далее следуют SQL запросы, которые заполняют следующие данные: номер телефона, адрес электронной почты и идентификатор аккаунта социальной сети ВКонтакте. Они аналогичны предыдущему, с единственным изменением имени переменной, в которую записывается информация о студенте.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 39   |





На фрагменте кода, представленном выше, реализована проверка на корректный ввод инициалов пользователя. Он состоит из двух блоков программы. В блоке try прописывается основной код. При получении системной ошибки запускается код, прописанный в блоке except.

Проверка реализована на всех этапах заполнения информации и для наглядности приведем еще один пример – проверка на заполнение пользователем id социальной сети ВКонтакте. Данная проверка представлена на следующем листинге программного кода:

#### Листинг программы

```
try:
    index = 1
    index = message.text.find("https://vk.com")
    if index == 0:
        ...
    else:
        send = bot.send_message(message.chat.id, "Ты допустил
ошибку! вид ствоей ссылки должен быть такой:\http://vk.com/id00000000")
        bot.register_next_step_handler(send, vkID)

except Exception:
    send = bot.send_message(message.chat.id, "Мы допустил ошибку!
введи ещё раз свой vk:")
    bot.register_next_step_handler(send, vkID)
```

На фрагменте кода, представленном выше, реализована проверка на корректный ввод id социально сети ВК. Данная проверка была реализована путём поиска подстроки в строке, введенной пользователем, а именно ищет основную часть ссылки и ее местоположение.

Последним шагом будет вывод данных определенной группы в отдельный CSV файл, который будет удобен в использовании кураторам. Данный операция

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 41   |

| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |
|--------------|--------------|--------------|--------------|--------------|
|              |              |              |              |              |

```

try:
    "COPY (SELECT * FROM students where group_id='{ }') TO
'Q:/Work/env/first/course/core/group.pdf' CSV ".format(message.text)
)
con.commit()
send = bot.send_message(message.chat.id, "Файл сформирован")
with open("Q:/Work/env/first/course/core/group.pdf", "rb") as
misc:
    file=misc.read()
    bot.send_document(message.chat.id, file)
except Exception:
    send = bot.send_message(message.chat.id, "Ошибка в номере
группы")

bot.register_next_step_handler(send, CSV)

```

Все вышеперечисленные команды, функции, запросы – немаловажная часть всего того, что войдет в финальную реализацию. Все они составляют оплот программной реализации, тестирование которой будет проведено в следующем разделе.

### 5.2.1 Студенческий бот

Это студенческий бот, предназначенный для заполнения, редактирования и вывода данных.

На выбор пользователю представлено два варианта, которые он может выбрать. Они реализованы с помощью кнопок. С помощью первой выполняется заполнение данных пользователем, с помощью второй вывести табличку с информацией, которая была заполнена ранее. Процесс заполнения представлен на Рисунке 16:

|              |              |              |              |              |                                                                                                                                                                                                                                                                                                                                        |
|--------------|--------------|--------------|--------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата | <p>и вывода данных.</p> <p>На выбор пользователю представлено два варианта, которые он может выбрать. Они реализованы с помощью кнопок. С помощью первой выполняется заполнение данных пользователем, с помощью второй вывести табличку с информацией, которая была заполнена ранее. Процесс заполнения представлен на Рисунке 16:</p> |
|              |              |              |              |              |                                                                                                                                                                                                                                                                                                                                        |
|              |              |              |              |              |                                                                                                                                                                                                                                                                                                                                        |
| Изм.         | Лист         | № докум      | Подп         | Дата         | <p>КР-02069964-090301-02-20</p>                                                                                                                                                                                                                                                                                                        |
|              |              |              |              |              | <p>Лист</p> <p>43</p>                                                                                                                                                                                                                                                                                                                  |





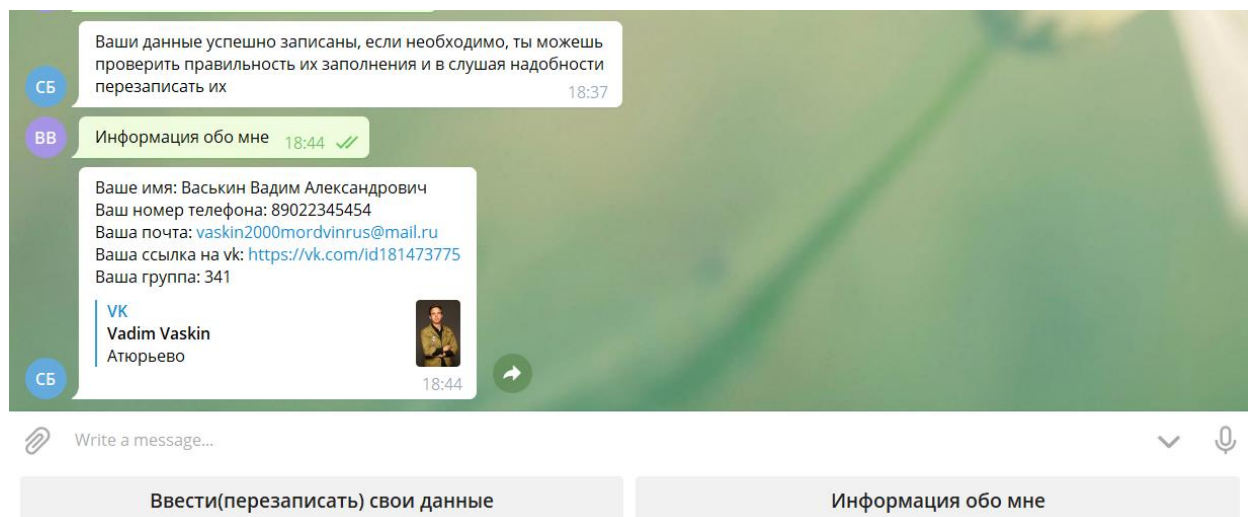


Рисунок 19 – табличка с данными студента

Теперь проверим правильность данных, которые были записаны в БД. Для этого с помощью запроса `Select * from students` выведем на экран список всех студентов, которые были занесены в базу данных. Результат приведен на Рисунке 20:

| kursach=# select * from students; |                               |             |                              |                                                                     |          |
|-----------------------------------|-------------------------------|-------------|------------------------------|---------------------------------------------------------------------|----------|
| id                                | name                          | phone       | email                        | vk_acc                                                              | groop_id |
| 1359755865                        | Зевайкин Андрей Евгеньевич    | 89513438128 | mxxskorpion@gmail.com        | <a href="https://vk.com/thevaykin">https://vk.com/thevaykin</a>     | 303      |
| 859263274                         | Камаева Анастасия Андреевна   | 89271893737 | aakamaeva@mail.ru            | <a href="https://vk.com/id155401145">https://vk.com/id155401145</a> | 341      |
| 531554436                         | Юртаев Данила Вячеславович    | 89063795515 | kniloslavskiy@gmail.com      | <a href="https://vk.com/id194937316">https://vk.com/id194937316</a> | 241      |
| 495734480                         | Кемаев Владислав Викторович   | 89876844568 | Kemaev-v@mail.ru             | <a href="https://vk.com/vvvlad_ka">https://vk.com/vvvlad_ka</a>     | 341      |
| 49021857                          | Тепалева Юлия Евгеньевна      | 89969603121 | tepaevayuliya@yandex.ru      | <a href="https://vk.com/id134400786">https://vk.com/id134400786</a> | 341      |
| 296749257                         | Дугушкин Антон Сергеевич      | 89271784820 | dugushkin.anton@mail.ru      | <a href="https://vk.com/anton_trou">vk.com/anton_trou</a>           | 341      |
| 122073030                         | Ямашкин Станислав Анатольевич | 79271821717 | Yamashkinsa@mail.ru          | yamashkinsa                                                         | 941      |
| 269548081                         | Андронов Евгений Алексеевич   | 89101235498 | gafter921@yandex.ru          | <a href="https://vk.com/gedkotlet">https://vk.com/gedkotlet</a>     | 341      |
| 1342634302                        | Васькин Вадим Александрович   | 89022345454 | vaskin2000mordvinrus@mail.ru | <a href="https://vk.com/id181473775">https://vk.com/id181473775</a> | 341      |
| (9 строк)                         |                               |             |                              |                                                                     |          |

Рисунок 20 – таблица PostgreSQL

Сверив данные, убеждаемся, что запись в БД из бота была реализована правильно.

|              |              |              |              |              |                          |  |  |  |  |      |
|--------------|--------------|--------------|--------------|--------------|--------------------------|--|--|--|--|------|
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |                          |  |  |  |  | Лист |
|              |              |              |              |              |                          |  |  |  |  | 46   |
| Изм.         | Лист         | № докум.     | Подп.        | Дата         | KP-02069964-090301-02-20 |  |  |  |  |      |

| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дудл. | Подп. и дата |
|--------------|--------------|--------------|--------------|--------------|
|              |              |              |              |              |

Перейдя по ссылке @TestAllowRightLeftBot в поиске Telegram, перед нами открывается диалог с ботом, представленный на Рисунке 21:

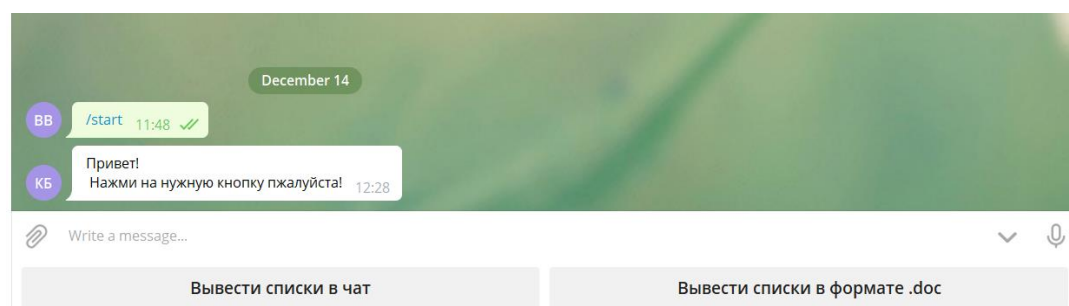


Рисунок 21 – чат, в кураторском боте

На рисунке видно, что кураторам на выбор представлено два варианта для дальнейшей работы. Первая кнопка выводит информацию о студентах группы в чат Telegram бота, в виде табличек с информацией. Пример вывода представлен на следующем скриншоте, сделанном с приложения Telegram:







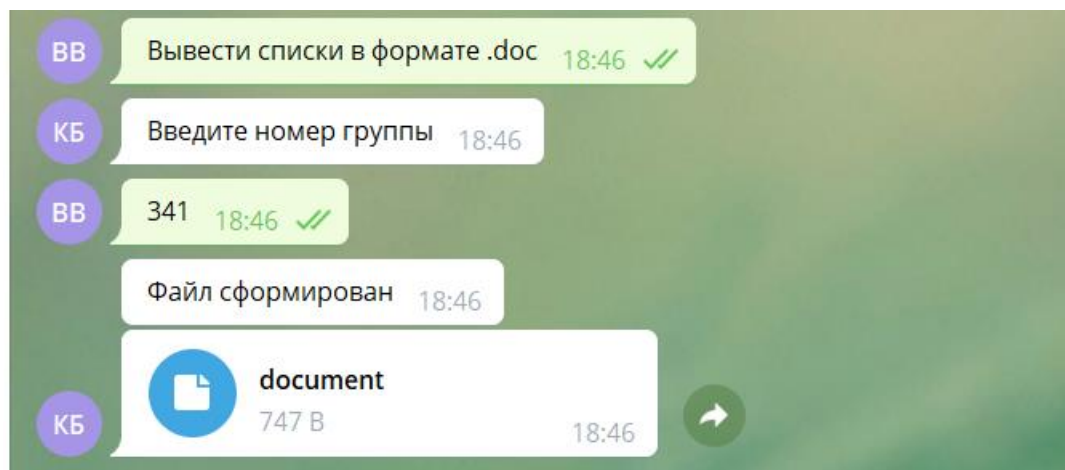


Рисунок 23 – процесс формирования файла

После скачивания файла и открытия его в редакторе, наблюдаем CSV файл, заполненный данными студентов 341 группы. На мой взгляд, это наиболее удобный вариант для визуализации студентов и дальнейшей работы с ними, предназначенный для кураторов. Пример готового отчета по 341 группе представлен на Рисунке 24:

|                                                                                                                 |           |                             |             |                         |                            |          |
|-----------------------------------------------------------------------------------------------------------------|-----------|-----------------------------|-------------|-------------------------|----------------------------|----------|
| group.csv [C:\Users\ACER\Downloads\] - Cell                                                                     |           |                             |             |                         |                            |          |
| Файл(F) Главная(H) Вставить(I) Разметка страницы(P) Формулы(M) Данные(A) Рецензирование(R) Просмотреть(W)       |           |                             |             |                         |                            |          |
| Вставить Формат ячеек Изменить текст Узор Затенение Общий Условное форматирование Стили Вставить Удалить Формат |           |                             |             |                         |                            |          |
| E11                                                                                                             |           |                             |             |                         |                            |          |
|                                                                                                                 | A         | B                           | C           | D                       | E                          | F        |
| 1                                                                                                               | id        | name                        | phone       | email                   | vk_acc                     | group_id |
| 2                                                                                                               | 859263274 | Камаева Анастасия Андреевна | 89271893737 | aakamaeva@mail.ru       | https://vk.com/id155401145 | 341      |
| 3                                                                                                               | 269548081 | Андронов Евгений Алексеевич | 89101235498 | gafter921@yandex.ru     | https://vk.com/gedkotlet   | 341      |
| 4                                                                                                               | 495734480 | Кемаев Владислав Викторович | 89876844568 | Kemaev-v@mail.ru        | https://vk.com/vvlad_ka    | 341      |
| 5                                                                                                               | 296749257 | Дугушкин Антон Сергеевич    | 89271784820 | dugushkin.anton@mail.ru | vk.com/antony_trou         | 341      |

Рисунок 24 – таблица в CSV формате

После всего вышесказанного, можно сделать вывод, что данный продукт, разработанный для кураторов, для сбора информации об абитуриентах, работает исправно и соответствует всем требованиям, поэтому тестирование можно признать успешным.

### 5.3 Выявление вариантов использования

Ежегодно кураторы первокурсников сталкиваются со следующей проблемой: сбор информации о первокурсниках, их данные, ссылки на социальные сети. Поэтому, продукт, разработанный в ходе данной курсовой работы должен стать отличной альтернативой тому, чем сейчас привыкли пользоваться кураторы. В таблице 1, приведенной ниже, расписаны варианты использования продукта.

Таблица 1 – Выявление вариантов использования

| <i>Основной фактор</i> | <i>Наименование</i>                  | <i>Формулировка</i>                                         |
|------------------------|--------------------------------------|-------------------------------------------------------------|
| Студент                | Заполнение баз данных                | Заполнение студентом своих личных данных                    |
| Куратор                | Просмотр и редактирование Баз данных | Просмотр и редактирование данных студентов кураторами.      |
| Преподаватель          | Просмотр и редактирование Баз данных | Просмотр и редактирование данных студентов преподавателями. |

Исходя из таблицы, представленной выше, делаем вывод, что данный проект может облегчить жизнь не только кураторам, но и преподавателям, которые смогут использовать Бот для распечатки актуальных списков студентов, и редактирования информации о них.

В таблице 2, представленной ниже, представлены альтернативные пути решения и недостатки методов, которыми кураторы пользуются сейчас для сбора данных о студентах.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата | Подп. и дата |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | KP-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 50   |

Таблица 2 – Альтернативные пути решения и их недостатки

| Пути решения                                                                                                                 | Недостатки                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Сбор информации о первокурсниках путем общения с каждым из них в социальных сетях, с целью получить от них нужную информацию | Большое количество времени и сил, для достижения данной задачи. Составление баз данных первокурсников вручную.     |
| Сбор информации о первокурсниках путем сбора их в аудитории. Работа с раздаточным материалом, бланками.                      | Большое количество времени и сил, которые уходят на сбор, анализ бланков, а также перевод их в электронный формат. |

На основе таблиц, приведенных выше, я делаю вывод, что тема курсового проекта актуальна. Поэтому, есть надежда, что реализация данной темы будет действительно востребована и найдет свое применение хотя бы в стенах нашего института.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата |              |
| Инв. № подл. |              |

|      |      |          |       |      |
|------|------|----------|-------|------|
| Изм. | Лист | № докум. | Подп. | Дата |
|      |      |          |       |      |

КР-02069964-090301-02-20

Лист

51

## ЗАКЛЮЧЕНИЕ

В результате написания курсового проекта была разработана и реализована система управления хранилищем данных для систематизации информации о студентах и абитуриентах университета.

Изучены возможности программы PostgreSQL для создания и работы с базами данных. Была подготовлена база данных с таблицами студентов, направлений подготовки и институтами, которые содержат данные направления. Между ними были прописаны связи.

Реализована связь между подготовленной на Posgres базой данных и Телеграмм ботом, написанным на языке программирования Python, целью которого было заполнение данных студентов и отправление этих данных на сервер, с которого уже шла запись в базу данных и дальнейшее формирование её в виде CSV файла.

Данная реализация сможет выполнить цель, которая была преследована мной изначально – помощь кураторам для сбора информации о абитуриентах и увидит свет, хотя бы в рамках нашего Университета.

Работа над данным проектом не закончена, еще многое предстоит сделать, прежде чем можно будет представить готовый продукт, поэтому данный проект можно будет развивать в качестве дипломной работы.

|              |              |
|--------------|--------------|
| Инв. № подл. | Подп. и дата |
| Взам. инв. № | Инв. № дубл. |
| Подп. и дата | Подп. и дата |
| Инв. № подл. | Подп. и дата |

|      |      |          |       |      |                          |      |
|------|------|----------|-------|------|--------------------------|------|
| Изм. | Лист | № докум. | Подп. | Дата | КР-02069964-090301-02-20 | Лист |
|      |      |          |       |      |                          | 52   |

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) В.В. Кириллов Основы проектирования реляционных баз данных. Учебное пособие. - СПб.: ИТМО, 1994. - 90 с.
- 2) М. Мейер Теория реляционных баз данных. - М.: Мир, 1987. - 608 с.
- 3) PostgreSQL Reference Manual - Volume 1: SQL Language Reference – The PostgreSQL Global Development Group, 2007.
- 4) Уорсли Дж., Дрейк Дж. PostgreSQL. Для профессионалов. - СПб.: Питер, 2003. - 496с.
- 5) Википедия. Свободная энциклопедия [Электронный ресурс].URL: <https://ru.wikipedia.org/wiki>
- 6) Классификация систем управления баз данных [Электронный ресурс]. URL: <https://scienceforum.ru/2016/article/2016019197>
- 7) Администрирование баз данных. [Электронный ресурс]. URL: <https://studfile.net/preview/2949235/page:7/>
- 8) Клиентские приложения PostgreSQL [Электронный ресурс]. URL:<https://postgrespro.ru>
- 9) Удаление данных. Команда DELETE [Электронный ресурс]. URL: <https://metanit.com/sql/postgresql/3.5.php>
- 10) СУБД Postgre SQL. История, современные возможности [Электронный ресурс]. URL: [https://www.yaneuch.ru/cat\\_22/subd-postgre-sql-istoriya-sovremennye/536062.3393248.page1.html](https://www.yaneuch.ru/cat_22/subd-postgre-sql-istoriya-sovremennye/536062.3393248.page1.html)
- 11) Пакет: python-psycopg2 [Электронный ресурс]. URL: <https://packages.debian.org/ru/jessie/python-psycopg2>
- 12) Экспорт данных из PostgreSQL в Excel [Электронный ресурс]. URL: [https://habr.com/ru/company/crystal\\_service/blog/245975/](https://habr.com/ru/company/crystal_service/blog/245975/)
- 13) Кодировка символов базы данных PostgreSQL [Электронный ресурс]. URL: <https://coderoad.ru>

|              |              |              |              |              |                                                                          |  |  |  |  |      |
|--------------|--------------|--------------|--------------|--------------|--------------------------------------------------------------------------|--|--|--|--|------|
| Инв. № подл. | Подл. и дата | Взам. инв. № | Инв. № дубл. | Подл. и дата | <div style="text-align: center;"> <i>KP-02069964-090301-02-20</i> </div> |  |  |  |  | Лист |
|              |              |              |              |              |                                                                          |  |  |  |  | 53   |
|              |              |              |              |              |                                                                          |  |  |  |  |      |
| Изм.         | Лист         | № докум.     | Подп.        | Дата         |                                                                          |  |  |  |  |      |

16) Положение столбца в таблице базы данных PostgreSQL [Электронный ресурс]. URL: <https://qastack.ru/programming/285733/how-do-i-alter-the-position-of-a-column-in-a-postgresql-database-table>

[illegible]