

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра комплексной информационной безопасности
электронно-вычислительных систем (КИБЭВС)

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА С БД ДЛЯ
ЗАГСА

Курсовая работа по дисциплине
«Безопасность систем баз данных»
Пояснительная записка

Студент гр. 712-1
_____ Благословиту К.М.
« ____ » _____ 2024г.

Руководитель
Старший преподаватель
кафедры КИБЭВС
_____ Новгородова Н.А.
Оценка « ____ » _____ 2024г.

РЕФЕРАТ

Курсовая работа содержит 59 страницы пояснительной записки, 70 рисунков, 6 источников, 3 приложения.

БАЗА ДАННЫХ, СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ, ИНФОРМАЦИОННАЯ СИСТЕМА ПЕРСОНАЛЬНЫХ ДАННЫХ, КОНФИДЕНЦИАЛЬНОСТЬ, ЦЕЛОСТНОСТЬ, ДОСТУПНОСТЬ, АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА.

Цель работы: необходимо предоставить пользовательский интерфейс автоматизированной информационной системы с подключением базы данных для ЗАГС.

Разработка программы проводилась на языке программирования C#, разработка базы данных проводилась в системе управления базами данных postgresSQL.

Курсовая работа выполнена в текстовом редакторе google docs.

Пояснительная записка оформлена согласно ОС ТУСУР 01-2021.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра комплексной информационной безопасности
электронно-вычислительных систем (КИБЭВС)

УТВЕРЖДАЮ

Заведующий кафедрой КИБЭВС

д-р техн. наук, профессор

_____ А.А. Шелупанов

« ____ » _____ 2024г.

ЗАДАНИЕ

на курсовую работу по дисциплине «**Безопасность систем баз данных**»
студенту Благословиту Кириллу Михайловичу группы 712-1 факультета
безопасности.

1 Тема работы: создание автоматизированной информационной системы
(АИС) с учебно-исследовательской базой данных (БД) для заданной предметной
области.

2 Исходные данные к работе:

2.1 Реляционная СУБД: PostgreSQL.

2.2 Данные по предметной области: в качестве предметной области был
выбран ЗАГС и выдача свидетельства о смерти.

3 Срок сдачи студентом законченной работы: « ____ » _____ 202_ г.

4 Содержание курсовой работы:

4.1 Проектирование инфологической модели данных:

- описание и структуризация предметной области (описание бизнес-процессов, диаграммы IDEF0/UML и др.);
- представление модели «Сущность-связь».

4.2 Проектирование логической модели данных:

- проектирование реляционной базы данных на основе принципов нормализации;
- графическое представление логической модели данных в IDEF1X;
- глоссарий модели.

4.3 Физическое проектирование БД:

- создание базы данных и ее необходимых элементов;
- ограничения на базу данных;
- сопоставление логических и физических имен.

4.4 Обеспечение безопасности данных.

4.5 Разработка программы по работе с данными БД для пользователей:

- меню, реализующее пользовательский интерфейс;
- просмотр данных с использованием экранных форм;
- добавление, редактирование, удаление данных;
- поиск и манипулирование данными (сортировки, фильтры и пр.);
- использование SQL операторов (операторы определения данных, операторы манипулирования данными и др.).

5 Содержание пояснительной записки:

- титульный лист;
- реферат на русском языке;
- задание;
- оглавление;
- введение;
- проектирование БД;
- вопросы безопасности данных;
- описание прикладной программы;
- заключение;

- список использованных источников;
- приложения (листинг программы и др.).

Пояснительная записка должна быть оформлена в соответствии со стандартом ТУСУР.

6 Дата выдачи задания:

« _____ » _____ 2024 г.

Задание согласовано:

Руководитель работы

Новгородова Н.А., старший преподаватель кафедры КИБЭВС

« _____ » _____ 2024 г. _____

Задание принято к исполнению

Благословиту К.М., студент группы 712-1

« _____ » _____ 2024 г. _____

Оглавление

Введение.....	7
1 ПРОЕКТИРОВАНИЕ ИНФОЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ.....	8
2 ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ.....	11
2.1 Логическое проектирование.....	12
2.2 Физическое проектирование БД.....	13
2.3 Управление доступом.....	23
3 РАЗРАБОТКА ПРОГРАММЫ ПО РАБОТЕ С ДАННЫМИ БД ДЛЯ ПОЛЬЗОВАТЕЛЕЙ.....	29
Заключение.....	38
Список использованных источников.....	39
Приложение А (справочное) Определение пунктов мер обеспечения безопасности персональных данных.....	40
Приложение Б (обязательное) Листинг программы.....	47
Приложение В (справочное) Как подключиться к базе данных, как источнику данных в Visual studio.....	48

Введение

Цель курсовой работы – создание автоматизированной информационной системы (АИС) с учебно-исследовательской базой данных (БД) для ЗАГС. В работе для выбранной предметной области создается автоматизированная информационная система в виде законченного программного продукта.

Информация в данной системе представлена в базе данных. Автоматизированная информационная система создается для работы с данными в БД конечных пользователей (согласно специфики предметной области).

1 ПРОЕКТИРОВАНИЕ ИНФОЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ

Предметная область ЗАГСа охватывает оформление актов гражданского состояния, таких как рождение, брак, расторжение брака и смерть. Окончательное получение соответствующего документа происходит после подачи необходимых документов, рассмотрения заявления и, при необходимости, прохождения дополнительной процедуры. После удовлетворительного рассмотрения заявления и оплаты государственной пошлины вам выдают соответствующий свидетельство или акт.

Для структуризации понимания бизнес-процессов и выделения взаимосвязей в системе была составлена диаграмма в нотации IDEF0, представленная на рисунке 1. Были выделены следующие входные данные:

- Данные паспорта;
- Заявление;
- Свидетельство о рождении;
- Свидетельство о браке.

В качестве механизмов управления были указаны:

- База данных «ЕГР ЗАГС»;
- База данных «Госуслуги»;
- НПА, регулирующие сферу обращения с ПДн;
- Федеральный закон от 15.11.1997 №143-ФЗ «Об актах гражданского состояния».

Также были выделены следующие механизмы управления:

- Сотрудник ЗАГС;
- Заявитель;
- Автоматизирования система «ЕГР ЗАГС»;
- Автоматизирования система «Госуслуги».

Были определены следующие результаты деятельности:

- Свидетельство о смерти

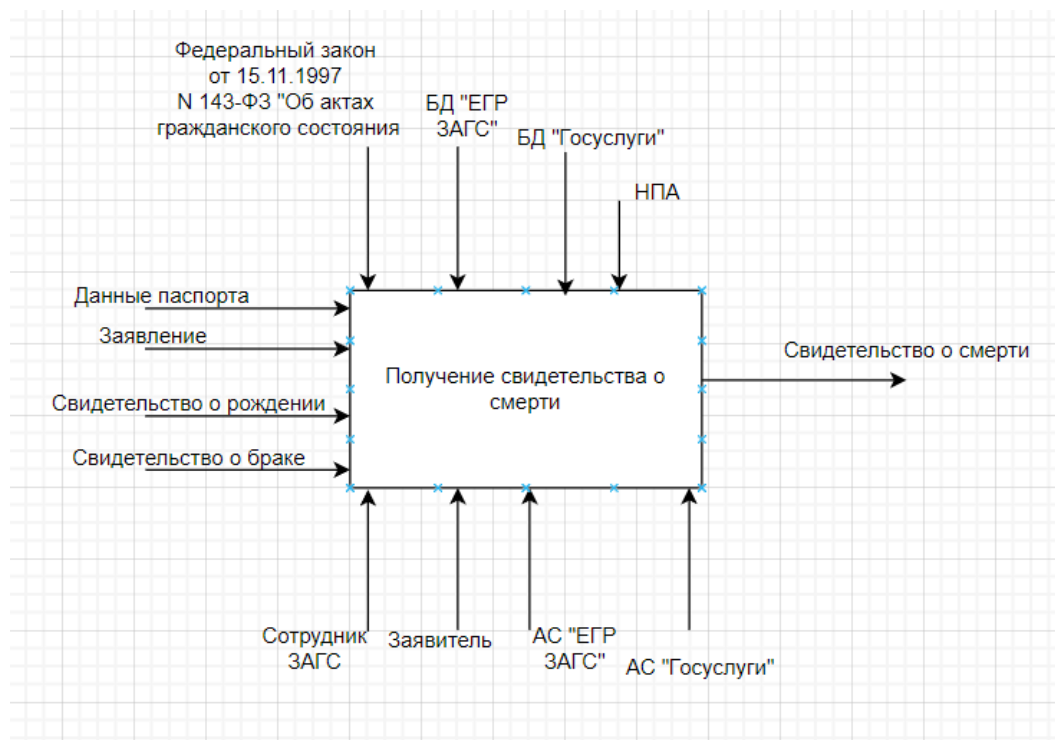


Рисунок 1 – «Черный ящик»

Была произведена декомпозиция построенной диаграммы, то есть разделение моделируемой функции на функции-компоненты, что представлено на рисунке 2. В качестве основных процессов были выделены:

- Принятие заявления;
- рассмотрение заявления;
- выдача свидетельства о смерти.

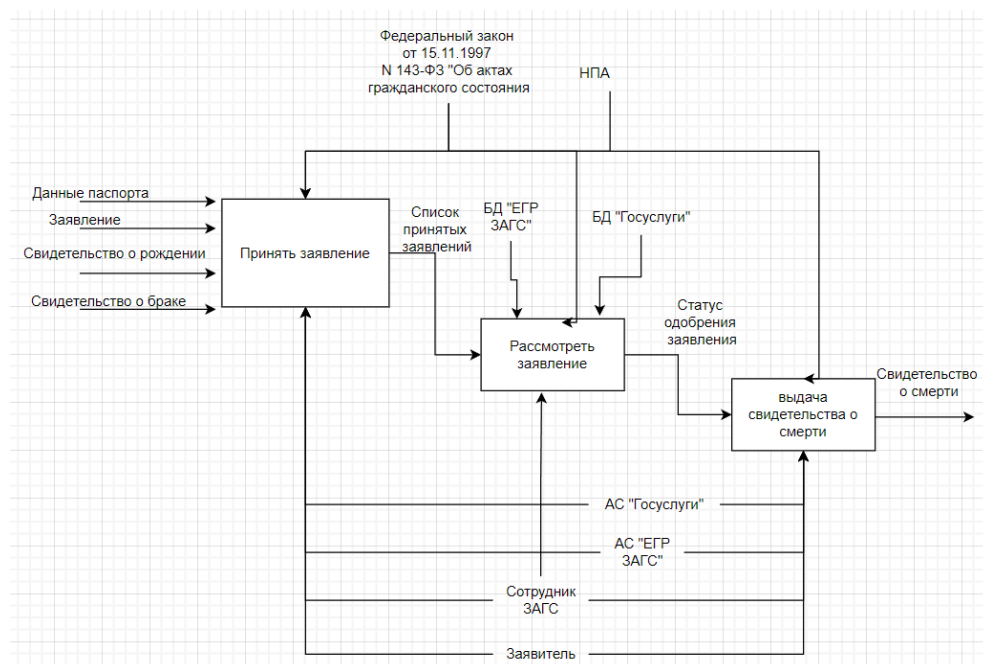


Рисунок 2 – Декомпозиция «Чёрного ящика»

Были выделены основные объекты предметной области, их характеристики и связи, на основе чего была построена концептуальная модель системы (рисунок 3).

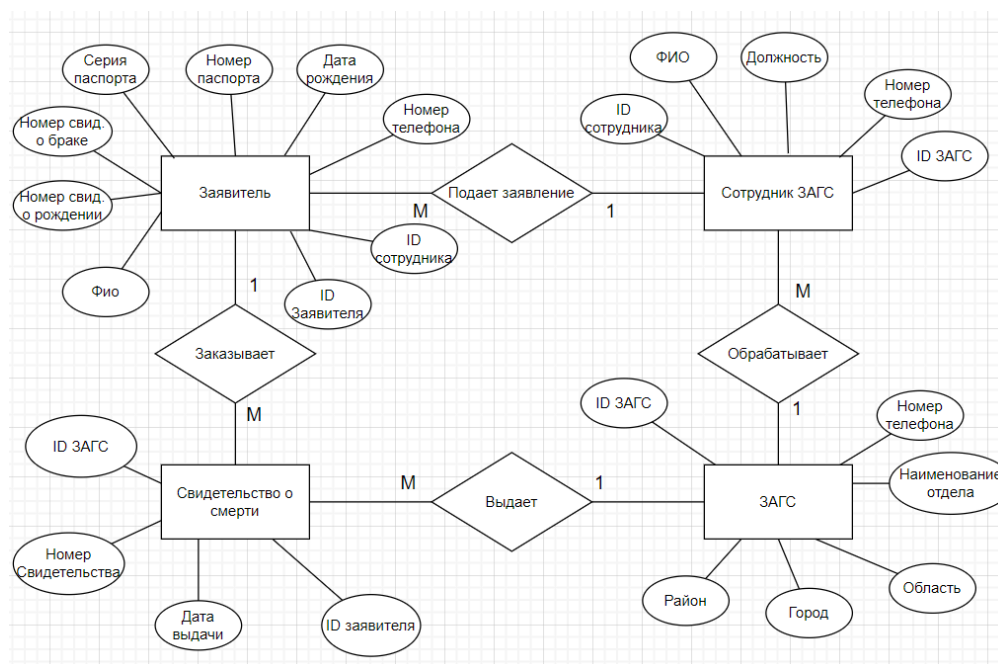


Рисунок 3 – Концептуальная модель системы

2 ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

В качестве основной системы управления базами данных, используемой в рамках выполнения данной курсовой работы, был выбран PostgreSQL. Во-первых, данная СУБД полностью поддерживает реляционную модель данных, что соответствует требованиям курсовой работы. Она позволяет создавать таблицы с четко определенными отношениями между ними, поддерживает использование первичных и внешних ключей для обеспечения целостности данных, а также поддерживает сложные запросы на языке SQL для взаимодействия с реляционными структурами [3]. Во-вторых, SQL Server обеспечивает высокий уровень производительности при работе с базами данных. Кроме того, данная СУБД обеспечивает многоуровневую защиту данных. В нее встроены такие функции, как шифрование данных, управление ролями и пользователями, а также динамическое маскирование данных, что позволяет контролировать доступ к информации на уровне отдельных строк и столбцов. Это важно для курсовой работы, где одной из задач является обеспечение безопасности построенной базы данных.

Для разработки и развертывания базы данных в рамках курсовой работы была выбрана операционная система семейства Windows. Данный выбор обусловлен полной поддержкой работы с SQL. А также сама операционная система удобна для использования во время разработки и развертывания базы данных.

2.1 Логическое проектирование

На этапе логического проектирования базы данных необходимо разработать ее «логическую» структуру в соответствии с инфологической моделью предметной области.

На основе концептуальной модели системы были определены отдельные сущности и их атрибуты, на основе выделенных сущностей и их атрибутов была построена логическая модель данных. Логическая модель была приведена к нормальной форме Бойса-Кодда и перестроена в нотации IDEF1X (рисунок 4).

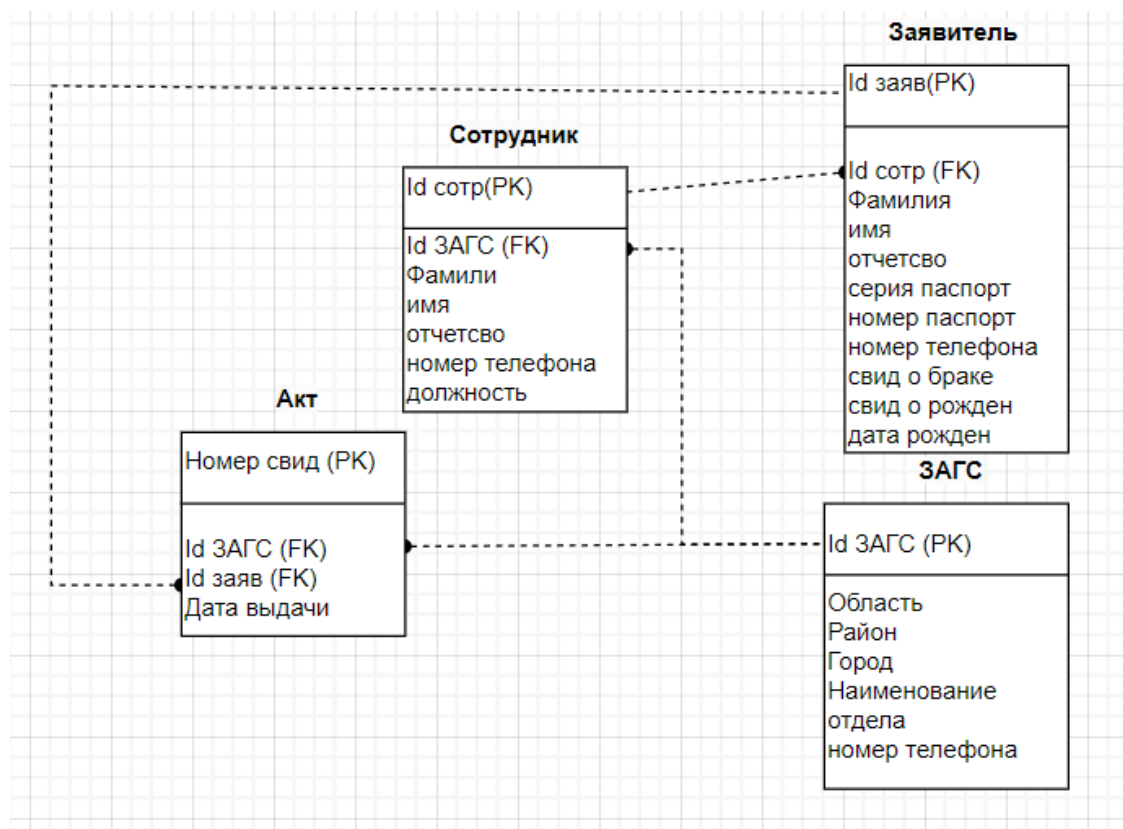


Рисунок 4 – Модель в нотации IDEF1X

2.2 Физическое проектирование БД

В приложении Г описано, как скачать и настроить PostgreSQL. В приложении Д описаны основные элементы управления в СУБД, а также как построить свою БД

На этапе физического проектирования необходимо работать в СУБД PostgreSQL. Были написаны скрипты для создания самой базы данных, скрипт на языке SQL представлен ниже:

```
CREATE DATABASE kurs
WITH
OWNER = postgres
ENCODING = 'UTF8'
LC_COLLATE = 'Russian_Kazakhstan.1251'
LC_CTYPE = 'Russian_Kazakhstan.1251'
LOCALE_PROVIDER = 'libc'
TABLESPACE = pg_default
CONNECTION LIMIT = -1
IS_TEMPLATE = False;
```

Для создания таблиц в базе данных необходимо понимать какие ограничения есть на каждый атрибут. Для этого был составлен глоссарий с указанием наложенных ограничений, представленный в таблице 1-4.

Таблица 1 – Глоссарий таблицы заявитель

имена атрибутов	тип	ограничения	null?
Фамилия	строковый	только буквы + тире, 30 символов	not
Имя	строковый	только буквы + тире, 30 символов	not
Отчество	строковый	только буквы, 30 символов	null
Id заяв	число	порядковый номер, не должно повторяться, только число, первичный ключ	not
Id сотр	число	порядковый номер, существует, не должно	not

		повторяться, только число, внешний ключ определяется из таблицы сотрудник первичного ключа Id сотрудника	
--	--	---	--

Продолжение таблицы 1

серия паспорт	число	4 символов, только число, уникально в связке с номером паспорта	not
номер паспорт	число	6 символов, только число, уникально в связке серией паспорта	not
номер телефона	число	формат +7XXXXXXXXXX, только числа	null
свид о браке	число	6 символов, только числа	null
свид о рожден	число	6 символов, только числа	not
дата рожден	дата	формат dd.mm.yyyy, только число + ., должно быть действительным	not

Таблица 2 – Глоссарий для таблицы сотрудник

имена атрибутов	тип	ограничения	null?
Фамилия	строковый	только буквы + тире, 30 символов	not
Имя	строковый	только буквы + тире, 30 символов	not
Отчество	строковый	только буквы, 30 символов	null
Id сотрудника	число	порядковый номер, не должно повторяться, только число, первичный ключ	not
Id ЗАГС	число	порядковый номер, существует, не должно повторяться, только число, внешний ключ на атрибут Id ЗАГС	not
номер телефона	число	формат +7XXXXXXXXXX, только числа	null

должность	строковый	только буквы + тире, 30 символов	not
-----------	-----------	----------------------------------	-----

Таблица 3 – Глоссарий таблицы ЗАГС

имена атрибутов	тип	ограничения	null?
Id ЗАГС	число	порядковый номер, не должно повторяться, только число, первичный ключ	not
Область	строковый	только буквы + тире, 30 символов	not
Район	строковый	только буквы + тире, 30 символов	not
Город	строковый	только буквы + тире, 30 символов	not
Наименование отдела	строковый	только буквы + пробел, 60 символов	not
номер телефона	число	формат +7XXXXXXXXXX, только числа	not

Таблица 4 – Глоссарий таблицы Акт

имена атрибутов	тип	ограничения	null?
Номер свид	число	7 символов, формат XX XXXXX, уникальный, первичный ключ	not
Id ЗАГС	число	порядковый номер, существует, не должно повторяться, только число, внешний ключ на атрибут таблицы ЗАГС Id ЗАГС	not
Дата выдачи	дата	формат dd.mm.yyyy, должно быть действительным	not
Id заявителя	число	порядковый номер, существует, не должно повторяться, только число, внешний ключ на атрибут таблицы Заявитель Id заяв	not

После этого необходимо создать таблицы. Пример создания таблиц через графический интерфейс представлен на рисунках 5-8.

ЗАГС

Общие Столбцы Дополнительно Ограничения Параметры Безопасность SQL

Имя: ЗАГС

Владелец: postgres

Схема: public

Табличное пространство: pg_default

Partitioned table?: ☐

Комментарий:

Закрыть Сбросить Сохранить

Рисунок 5 – Создание таблицы ЗАГС

ЗАГС

Общие **Столбцы** Дополнительно Ограничения Параметры Безопасность SQL

Наследуется из таблиц(ы): Выберите источник наследования...

Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный кл...	По умолч...
ID ЗАГС	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
область	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
город	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
район	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
наименование	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
телефон	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Закрыть Сбросить Сохранить

Рисунок 6 – Создание столбцов в таблице ЗАГС

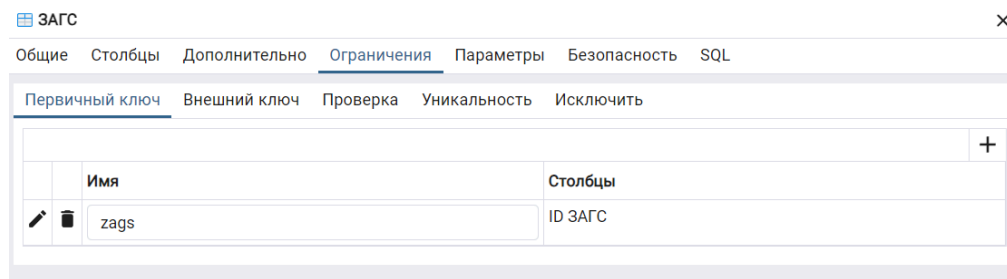


Рисунок 7 – Создание первичного ключа

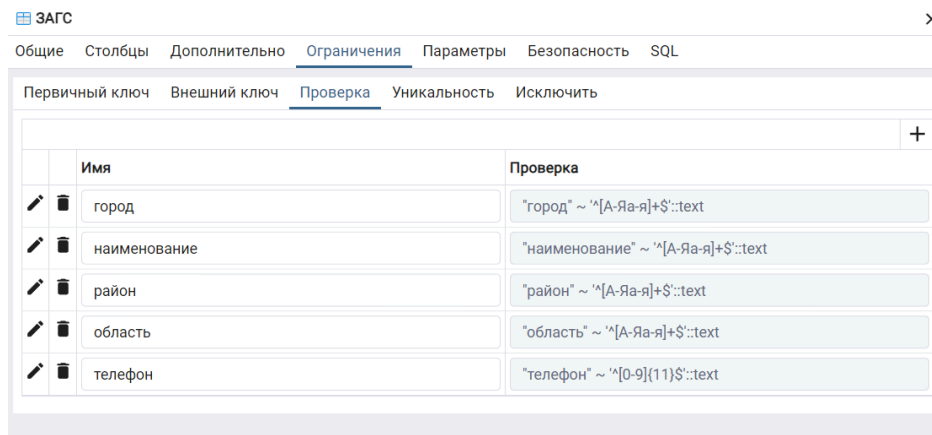


Рисунок 8 – Создание ограничений на столбцы

Далее были созданы таблицы с помощью скриптов. Скрипты представлены ниже.

Скрипт создания таблицы «Сотрудник»

```
-- Table: public.Сотрудники
```

```
-- DROP TABLE IF EXISTS public."Сотрудники";
```

```
CREATE TABLE IF NOT EXISTS public."Сотрудники"
```

```
(
```

```
  "ID сотрудника" bigint NOT NULL,
```

```
  "ID зарп" bigint NOT NULL,
```

```
  "Фамилия" text COLLATE pg_catalog."default" NOT NULL,
```

```
  "Имя" text COLLATE pg_catalog."default" NOT NULL,
```

```
  "Отчество" text COLLATE pg_catalog."default",
```

```
  "телефон" text COLLATE pg_catalog."default" NOT NULL,
```

```
  "Должность" text COLLATE pg_catalog."default" NOT NULL,
```

```
  CONSTRAINT sotr_pk PRIMARY KEY ("ID сотрудника"),
```

```

CONSTRAINT sotr_zags_fk FOREIGN KEY ("ID зaгc")
REFERENCES public."ЗАГC" ("ID ЗАГC") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT "Должность" CHECK ("Должность" ~ '^[А-Яа-я]+$':text),
CONSTRAINT "Имя" CHECK ("Имя" ~ '^[А-Яа-я]+$':text),
CONSTRAINT "Отчество" CHECK ("Отчество" ~ '^[А-Яа-я]+$':text),
CONSTRAINT "Фамилия" CHECK ("Фамилия" ~ '^[А-Яа-я]+[А-Яа-я]+-[А-Яа-я]+$':text),
CONSTRAINT "телефон" CHECK ("телефон" ~ '^[0-9]{11}$':text)
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS public."Сотрудники"
OWNER to postgres;

```

Скрипт создания таблицы «Заявитель»

```

-- Table: public.Заявитель

```

```

-- DROP TABLE IF EXISTS public."Заявитель";

```

```

CREATE TABLE IF NOT EXISTS public."Заявитель"
(
    "Фамилия" text COLLATE pg_catalog."default" NOT NULL,
    "Имя" text COLLATE pg_catalog."default" NOT NULL,
    "Отчество" text COLLATE pg_catalog."default",
    "ID заявителя" bigint NOT NULL,
    "ID сотрудника" bigint NOT NULL,
    "Серия паспорта" numeric(4,0) NOT NULL,
    "Номер паспорта" numeric(6,0) NOT NULL,
    "телефон" text COLLATE pg_catalog."default",
    "свид. о браке" text COLLATE pg_catalog."default",
    "Дата рождения" date NOT NULL,
    "свид. о рождении" text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT zayav_pk PRIMARY KEY ("ID заявителя"),
    CONSTRAINT "Брак" UNIQUE ("свид. о браке"),
    CONSTRAINT "Паспорт" UNIQUE ("Серия паспорта", "Номер паспорта"),
    CONSTRAINT "Рождение" UNIQUE ("свид. о рождении"),

```

```

CONSTRAINT zayav_sotr_fk FOREIGN KEY ("ID сотрудника")
REFERENCES public."Сотрудники" ("ID сотрудника") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT "Фамилия" CHECK ("Фамилия" ~ '^[А-Яа-я]+[А-Яа-я]+-[А-Яа-я]+$'::text),
CONSTRAINT "Имя" CHECK ("Имя" ~ '^[А-Яа-я]+$'::text),
CONSTRAINT "Отчество" CHECK ("Отчество" ~ '^[А-Яа-я]+$'::text),
CONSTRAINT "телефон" CHECK ("телефон" ~ '^[0-9]{11}$'::text),
CONSTRAINT "Дата рождения" CHECK ("Дата рождения" <= (CURRENT_DATE - '16
years'::interval))
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS public."Заявитель"
OWNER to postgres;

```

Скрипт создания таблицы «Акт»

```

-- Table: public.Акт

```

```

-- DROP TABLE IF EXISTS public."Акт";

```

```

CREATE TABLE IF NOT EXISTS public."Акт"
(
    "Номер" numeric(7,0) NOT NULL,
    "ID ЗАГС" bigint NOT NULL,
    "ID заявителя" bigint NOT NULL,
    "Дата выдачи" date NOT NULL,
    CONSTRAINT akt_nomer_pk PRIMARY KEY ("Номер"),
    CONSTRAINT akt_zags_fk FOREIGN KEY ("ID ЗАГС")
REFERENCES public."ЗАГС" ("ID ЗАГС") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID,
CONSTRAINT akt_zayav_fk FOREIGN KEY ("ID заявителя")
REFERENCES public."Заявитель" ("ID заявителя") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION

```

```

NOT VALID,
CONSTRAINT "Дата выдачи" CHECK ("Дата выдачи" <= CURRENT_DATE)
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS public."Акт"
OWNER to postgres;

```

После создания таблицы были заполнены 4 строками в каждой, также была проведена проверка ограничений. Было проверено работает ли ограничение, которое проверяет правильность заполнения столбца «район» (рисунок 9). Пример заполненной таблицы «ЗАГС» представлен на рисунке 10.

❗ Ошибочная строка содержит (1, Томская, Томск, 123456, ЗАГСТОМСК, 78945612378).новая строка в отношении "ЗАГС" нарушает ограничение-проверку "район" ❌

Рисунок 9 – Ошибка проверки ограничения

ID ЗАГС [PK] bigint	область text	город text	район text	наименование text	телефон text
4	Томская	Томск	Октябр...	ЗАГСТОМСК	34567891234
3	Томская	Томск	Ленинс...	ЗАГСТОМСК	23456789123
2	Томская	Томск	Кировс...	ЗАГСТОМСК	12345678912
1	Томская	Томск	Советс...	ЗАГСТОМСК	78945612378

Рисунок 10 – Заполненная таблица «ЗАГС»

Далее были заполнена таблица «Сотрудники». При заполнении таблицы «Сотрудники» были проверены следующие ограничение:

- внешний ключ столбца «ID загс» должен существовать (рисунок 11);
- столбец «Фамилия» обязательно должен быть вписан (рисунок 12);
- столбец «телефон» должен содержать только цифры и иметь только 11 символов, что показано на рисунках 13 и 14 соответственно;
- в столбце «Фамилия» может содержаться символ «-» (рисунок 15).

Ключ (ID загс)=(5) отсутствует в таблице "ЗАГС".INSERT или UPDATE в таблице "Сотрудники" нарушает ограничение внешнего ключа "sotr_zags_fk" ✖

Рисунок 11 – Пример ошибки при отсутствии записи в другой таблице

Ошибочная строка содержит (1, 1, null, Иван, null, 45678912345, менеджер).значение NULL в столбце "Фамилия" отношения "Сотрудники" нарушает ограничение NOT NULL ✖

Рисунок 12 – Пример ошибки незаполнения обязательного поля

Ошибочная строка содержит (1, 1, Иванов, Иван, null, 45678912д45, менеджер).новая строка в отношении "Сотрудники" нарушает ограничение-проверку "телефон" ✖

Рисунок 13 – Неправильный ввод номера телефона

Ошибочная строка содержит (1, 1, Иванов, Иван, null, 456789121145, менеджер).новая строка в отношении "Сотрудники" нарушает ограничение-проверку "телефон" ✖

Рисунок 14 – Пример заполнения телефона с избыточными цифрами

	ID сотрудника [PK] bigint	ID загс bigint	Фамилия text	Имя text	Отчество text	телефон text	Должность text
1	1	1	Иванов-Иванов	Иван	[null]	45678912145	менеджер

Рисунок 15 – Пример работающего ограничения на знак «-»

При заполнении таблицы «Заявитель» были проверены следующие ограничения:

- столбец «Дата рождения» должна быть действительной и ограничение по возрасту, то есть заявитель должен быть не моложе 16 лет, что представлено на рисунках 16 и 17 соответственно;
- столбец «свид. о рождении» должен быть уникален для каждого заявителя (рисунок 18);
- столбцы «серия паспорта» и «номер паспорта» должны быть уникальны в связке друг с другом (рисунок 19), пример того что столбцы по отдельности могут быть не уникальны (рисунок 20).

❗ Ошибочная строка содержит (Иванов , Иван, Иванович, 1, 1, 1234, 123456, 89123456789, С-Н123456, 2200-01-01, I-0123456).новая строка в отношении "Заявитель" нарушает ограничение-проверку "Дата рождения" ❌

Рисунок 16 – Дата не данный момент не может быть выставлена

❗ Ошибочная строка содержит (Иванов , Иван, Иванович, 1, 1, 1234, 123456, 89123456789, С-Н123456, 2020-01-01, I-0123456).новая строка в отношении "Заявитель" нарушает ограничение-проверку "Дата рождения" ❌

Рисунок 17 – Дата рождения меньше 16 лет

❗ Ключ ("свид. о рождении")=(I-0123456)" уже существует.повторяющееся значение ключа нарушает ограничение уникальности "Рождение" ❌

Рисунок 18 – Ошибка ограничения уникальности

❗ Ключ ("Серия паспорта", "Номер паспорта")=(1234, 123456)" уже существует.повторяющееся значение ключа нарушает ограничение уникальности "Паспорт" ❌

Рисунок 19 – Ключ не уникален в связке

	Фамилия text	Имя text	Отчество text	ID заявителя [PK] bigint	ID сотрудника bigint	Серия паспорта numeric (4)	Номер паспорта numeric (6)
1	Иванов	Петр	Иванович	2	1	2345	123456
2	Иванов	Иван	Иванович	1	1	1234	123456

Рисунок 20 – Пример неуникальности столбца «Серия паспорта»

Далее создается 3 роли для работы в базе данных (далее БД), а именно администратор БД (далее Адмн), пользователь БД, менеджер отдела ЗАГС. Пример создания пользователя на рисунках 21- 23.

Роль группы - Роли входа/группы

Общие | Определение | Права | Членство | Параметры | Безопасность | SQL

Имя: Адмн

Комментарии:

Заккрыть Сбросить Сохранить

Рисунок 21 – Создание группы пользователей под именем «Адмн»

Роль группы - Роли входа/группы

Общие | Определение | Права | Членство | Параметры | Безопасность | SQL

Пароль:

Роль активна до: No Expiry

Please note that if you leave this field blank, then password will never expire.

Макс. число подключений: -1

Заккрыть Сбросить Сохранить

Рисунок 22 – Создание пароля для пользователя

Роль группы - Роли входа/группы

Общие | Определение | Права | Членство | Параметры | Безопасность | SQL

Вход разрешён?

Superuser?

Создание ролей?

Создание баз?

Наследует права от родительских ролей?

Может создавать потоковую репликацию и резервные копии?

Bypass RLS?

Заккрыть Сбросить Сохранить

Рисунок 23 – Выдача прав для группы адмн

По такому же принципу были созданы еще две роли с именами «менедж» и «использователь».

2.3 Управление доступом

Для разграничения доступа к таблицам была создана матрица доступа, Матрица представлена в таблице 5. В таблице столбец «Заявитель» соответствует созданной в таблице группе ролей «использователь».

Таблица 5 – Матрица доступа

Таблица\Пользователь	Менеджер	Заявитель	Администратор	Администратор пользователей
Заявитель	Select, Update, Insert	Select	Select, Update, Insert	-
Акт	Select, Update, Insert	Select	Select, Update, Insert	-
ЗАГС	Select	-	Select, Update, Insert, Delete	-
Сотрудник	Select	-	Select, Update, Insert, Delete	-
Password	-	-	-	Insert, update

Чтобы созданные роли имели права соответствия матрице доступа необходимо в свойствах таблице во вкладке «Security» указать какая роль какие имеет возможности по отношению к таблице. В рамках данной курсовой необходимо реализовать защиту для 2 ролей, в качестве примера были выбраны роли «использователь» и «адмн». Настройки для таблицы «Акт» представлены на рисунке 24.

Privileges			+
Grantee	Privileges	Grantor	
адмн	arw	postgres	
пользователь	r	postgres	
	<input type="checkbox"/> ALL <input type="checkbox"/> INSERT <input checked="" type="checkbox"/> SELECT <input type="checkbox"/> UPDATE <input type="checkbox"/> DELETE <input type="checkbox"/> TRUNCATE <input type="checkbox"/> REFERENCES <input type="checkbox"/> TRIGGER	<input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> WITH GRANT OPTION <input type="checkbox"/> WITH GRANT OPTION	

Рисунок 24 – Настройки доступа для таблицы «Акт»

По примеру выше были созданы настройки и для таблицы «Заявитель». Для таблиц «ЗАГС» и «Сотрудник» были выбраны следующие роли «менедж» и «адмн», данный выбор обусловлен тем, что согласно матрице доступа у роли «исполнитель» не может быть доступа к данным таблицам (рисунок 25).

Privileges			+
Grantee	Privileges	Grantor	
goto	r	postgres	
postgres	axrtDdw	postgres	
адмн	arwd	postgres	
менедж	r	postgres	

Рисунок 25 – Настройки доступа для таблицы «ЗАГС»

На рисунке можно увидеть роли «goto», она была создана для проверки работоспособности ограничений доступа к таблицам. Также роль «postgres» являясь суперпользователем, автоматически имеет все права доступа к таблицам, данные права выдаются автоматически самой СУБД. Для проверки доступа к таблицам была произведена проверка, посредством входа в базу данных под пользователем «использователь», результаты проверки представлены на рисунках 26-28.

PostgreSQL 16

General **Connection** Parameters SSH Tunnel Advanced

Host name/address localhost

Port 5432

Maintenance database kurs

Username **использователь**

Kerberos authentication? ☐

Role

Service

Close Reset Save

Рисунок 26 – Вход под пользователем «использователь»

public.Акт/kurs/использователь@PostgreSQL 16

No limit

Data Output Messages Notifications

	Номер [PK] numeric (7)	ID ЗАГС bigint	ID заявителя bigint	Дата выдачи date
1	1	2	4	2024-01-01
2	2	2	3	2024-01-02
3	3	3	2	2024-01-03
4	4	1	1	2024-01-04

Рисунок 27 – Пользователь видит таблицу поскольку имеет доступ к ней

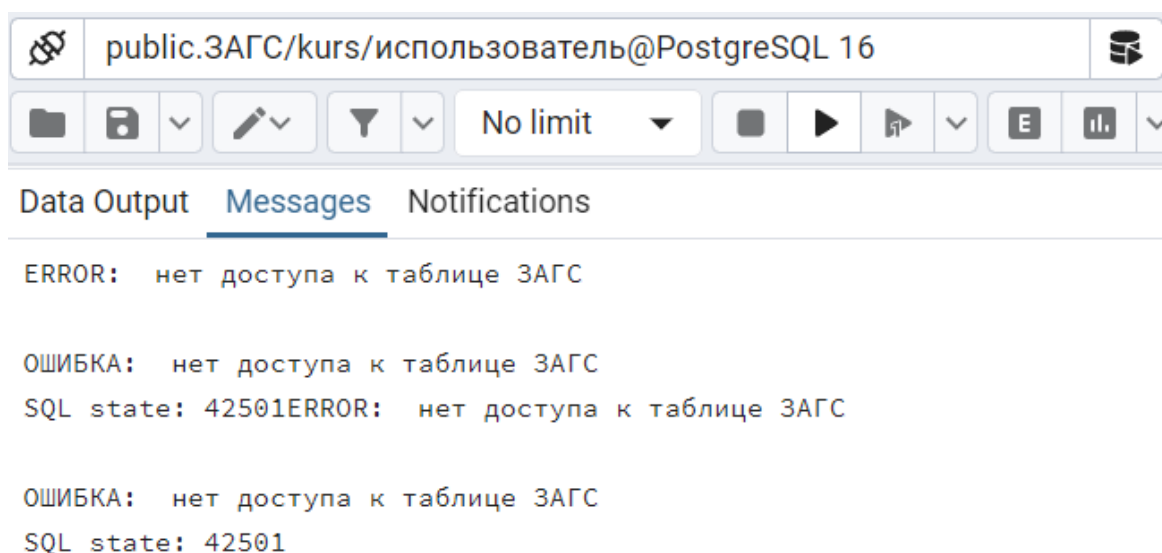


Рисунок 28 – Пользователь не видит данные таблицы, поскольку не имеет доступа к ней

Согласно федеральному закону №152 от 27 июля 2006 года «О персональных данных» [4], база данных (далее БД), в рамках курсовой работы, является информационной системой персональных данных (далее ИСПДн). Данные находящиеся в БД относятся к общим иным персональным данным. Далее необходимо обеспечить защиту конфиденциальности информации.

Так как в PostgreSQL, чтобы зашифровать данные необходимо дополнительно подключать библиотеки шифрования и шифровать во время заполнения таблицы, было принято решение шифровать данные на программном уровне.

Согласно постановлению правительства №1119 от 1 ноября 2012 года «Об утверждении требований к защите персональных данных при обработке в информационных системах персональных данных» [5] был установлен 3 уровень защищенности. Так как необходимо обеспечивать 3 уровень защищенности вследствие пункта 13 подпункта «д», для информационной системы актуальны угрозы 3-го типа и информационная система обрабатывает иные категории персональных данных более чем 100000 субъектов персональных данных, не являющихся сотрудниками оператора. Так как база данных, используемая ЗАГС, будет содержать информацию о более чем 100000

субъектов, которые не являются сотрудниками оператора базы данных. Также информация которая будет содержаться в базе данных относиться к иным персональным данным.

Согласно приказу ФСТЭК №21 18 февраля 2013 года «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных» [6], в рамках курсовой работы были выполнены следующие пункты:

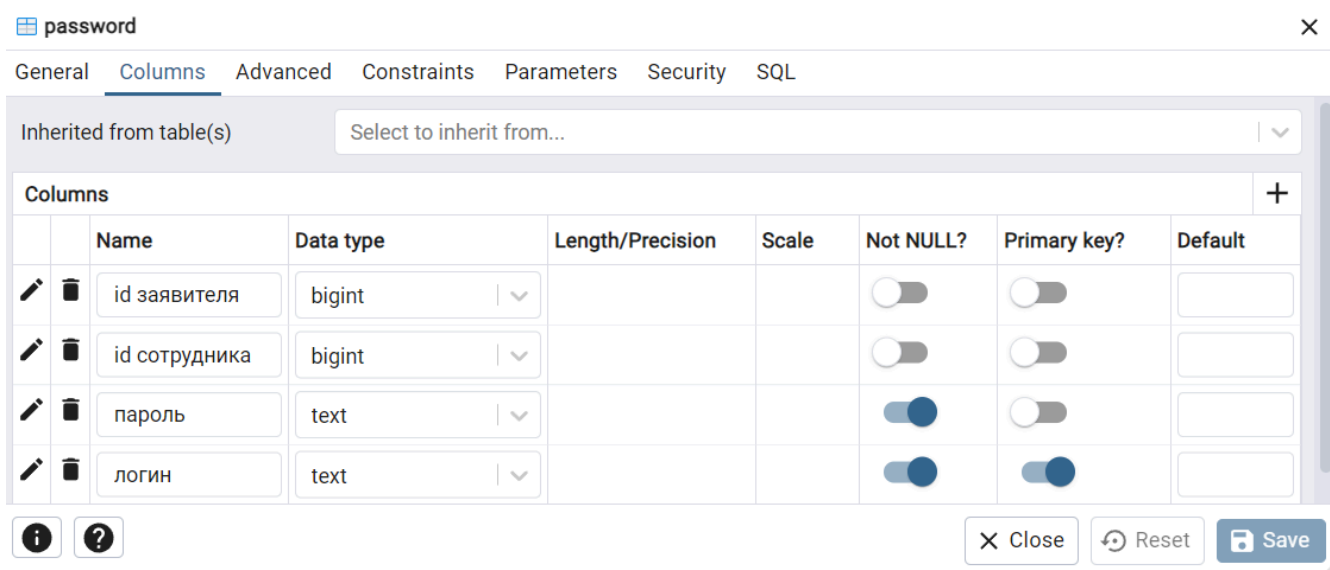
- мер идентификация и аутентификация субъектов доступа и объектов доступа: 1, 2, 3, 4, 5;
- мер управление доступом субъектов доступа к объектам доступа: 1, 2, 4, 5. Пункты 6 и 7 реализованы должны быть реализованы на программном уровне;
- меры защиты машинных носителей персональных данных также реализуются на программном уровне;
- для обеспечения целостности информационной системы и персональных данных были выполнены следующие пункты: 1, 2, 3, 4, 5, 6;
- мер по обеспечению доступности персональных данных: 2, 3;
- мер по управлению конфигурацией информационной системы и системы защиты персональных данных: 1.

Полное описание пунктов прописано в приложении А данного отчета. Так как в рамках данной курсовой работы были выполнены не все меры по обеспечению безопасности, их планируются выполнить в течении 3 лет после ввода проектируемой системы в эксплуатацию. Часть мер безопасности должны выполняться непосредственно на местах пользования проектируемой системы.

3 РАЗРАБОТКА ПРОГРАММЫ ПО РАБОТЕ С ДАННЫМИ БД ДЛЯ ПОЛЬЗОВАТЕЛЕЙ

Перед разработкой программы был выбран язык программирования C#. Данный выбор обусловлен тем что данный язык программирования был использован при разработке программы на курсовом проекте по языкам программирования.

Программа будет разработана для роли администратора (таблица 5). Данный пользователь не будет иметь доступ к СУБД, то есть работать только через интерфейс. Администратор может создавать логины и пароли для входа пользователей в приложение. В ходе создания формы «login», была создана дополнительная таблица с названием «password», данная таблица создана для хранения паролей входа в программу, и разграничение доступа на программном уровне. Столбцы таблицы представлены на рисунке 29.



password							
General Columns Advanced Constraints Parameters Security SQL							
Inherited from table(s) Select to inherit from...							
Columns							
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id заявителя	bigint			<input type="checkbox"/>	<input type="checkbox"/>	
	id сотрудника	bigint			<input type="checkbox"/>	<input type="checkbox"/>	
	пароль	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	логин	text			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Рисунок 29 – Столбцы таблицы «password»

Готовая программа выложена на сайте GitHub ссылка на проект представлена в приложении Б.

Для шифрования паролей с последующей передачей пароля в базу данных, используется следующий алгоритм шифрования:

- MD5 (Message-Digest algorithm) — алгоритм хеширования, разработанный профессором Р. Л. Ривестом в еще 1991 году. Алгоритм md5 шифрует любые данные в формате 128-bit hash (контрольную сумму), которую достаточно сложно подделать.

Для того чтобы была возможность создать остальных пользователей, в таблице «password» был добавлен пользователь с паролем, который был зашифрован, с помощью алгоритма хеширования описанный выше.

В ходе создания программы необходимо подключить базу данных, как источник данных. Как подключить БД показано в приложении В.

В рамках данной курсовой работы были созданы 4 формы, визуальная составляющая представлена на рисунках 30-33.

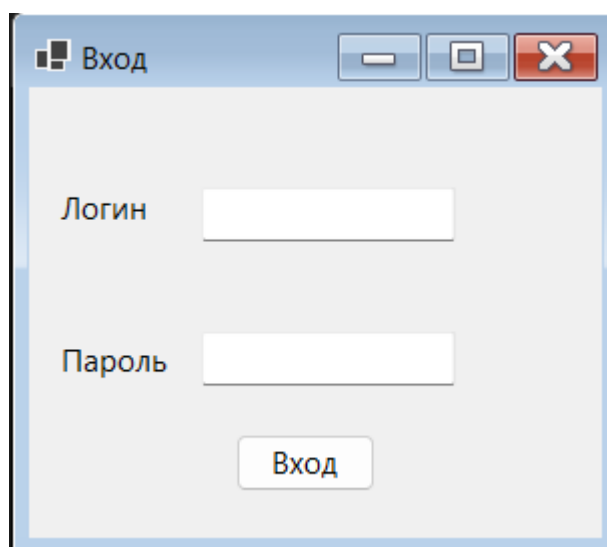
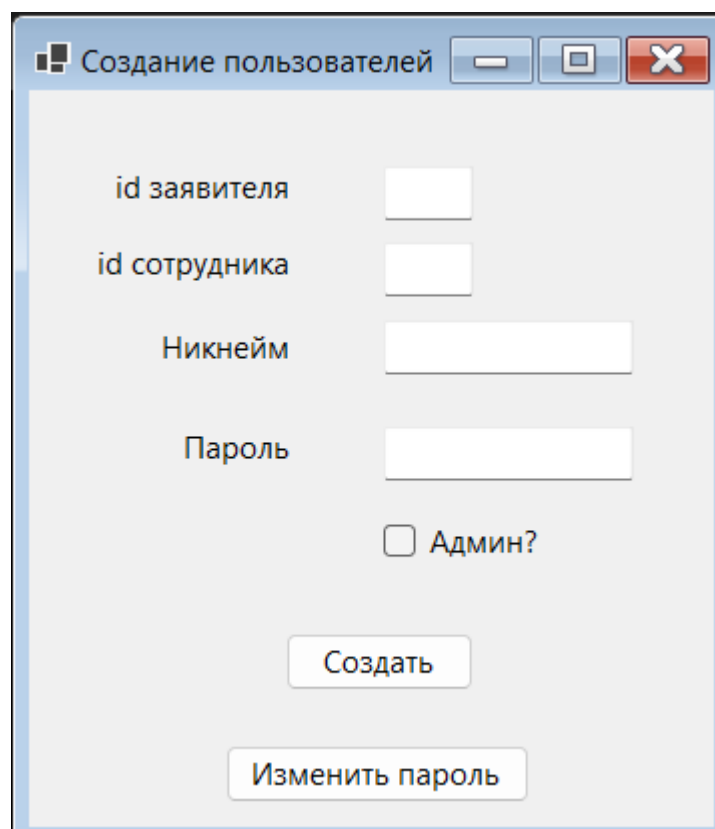


Рисунок 30 – Окно авторизации



Создание пользователей

id заявителя

id сотрудника

Никнейм

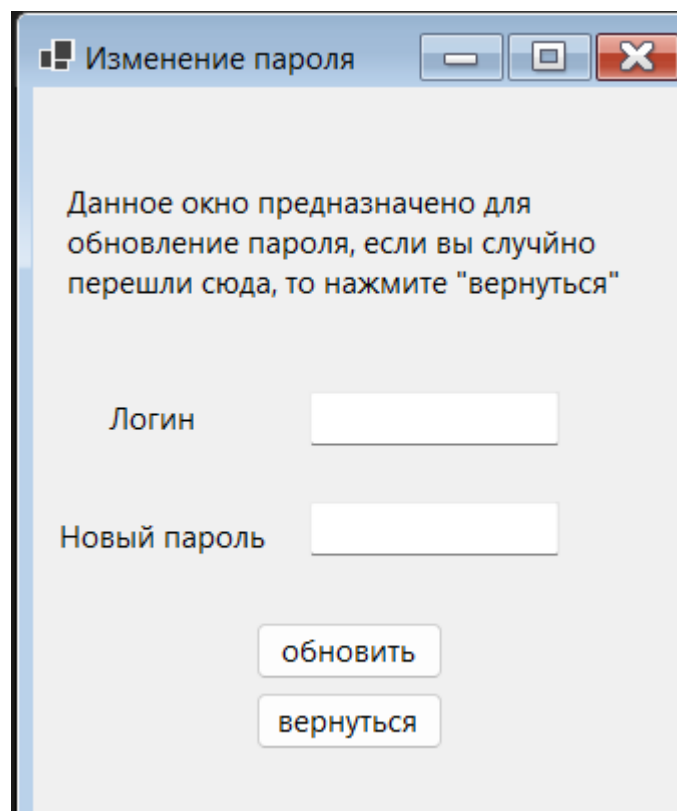
Пароль

☐ Админ?

Создать

Изменить пароль

Рисунок 31 – Окно создания пользователя



Изменение пароля

Данное окно предназначено для обновление пароля, если вы случайно перешли сюда, то нажмите "вернуться"

Логин

Новый пароль

обновить

вернуться

Рисунок 32 – Окно изменения пароля

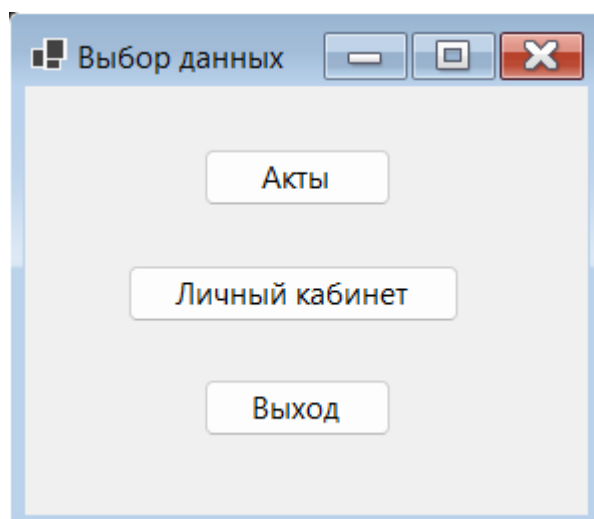


Рисунок 33 – Окно для пользователя

Окна создания пользователя и изменения пароля, доступны только администратору пользователей. Окно входа является начальным окном при запуске программы.

Для входа необходимо знать логин и пароль. Для проверки введенных данных происходит подключение к бд, через строку подключения. Строка подключения выглядит следующим образом:

```
string connectionString1 = "Server=localhost; port=5432; user id=postgres; password=1111; database=kurs";
```

При создании своей программы необходимо указывать свои данные в следующих пунктах: Server, port, user id, password, database. Данные параметры необходимо изменять, если при создании БД в postgresql, были указаны иные данные. Для защиты пароля при вводе в программе используется маскирование с заменой вводимых символов на символ «*», также для защиты пароля при передаче данных в БД и получения данных пароля из БД, создана функция с шифрования.

При передаче данных логина и пароля есть возможность ввода SQL-инъекции, для защиты от подобного типа атаки используется базовое экранирование. Данное экранирование переводит полученные данные в параметры после чего уже передает в БД, тем самым при попытке ввода SQL-инъекции, данная инъекция переводится в обычный текст и передается в

БД (рисунок 34). В ходе проверки работоспособности была экранирования была попытка ввода следующего запроса:

```
SELECT * FROM password WHERE логин = 'zaya' OR 1=1.
```

```
using (NpgsqlConnection connection = new NpgsqlConnection(connectionString1))
{
    connection.Open();
    // Проверяем наличие пользователя с указанным логином и паролем
    if (string.IsNullOrEmpty(login) || string.IsNullOrEmpty(password))
    {
        MessageBox.Show("Введите логин и пароль.");
        return;
    }
    using (NpgsqlCommand cmd = new NpgsqlCommand("SELECT \"id заявителя\", \"id сотрудника\", админ FROM password WHERE логин = @login AND пароль = @password", connection))
    {
        cmd.Parameters.AddWithValue("@login", login);
        cmd.Parameters.AddWithValue("@password", password);
        using (NpgsqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                // Обработка результатов
            }
        }
    }
}
```

Рисунок 34 – Экранирование

Результат попытки запроса, можно наблюдать на рисунке 35. Как видно в запросе имеется логин, который имеется в БД (рисунок 36).

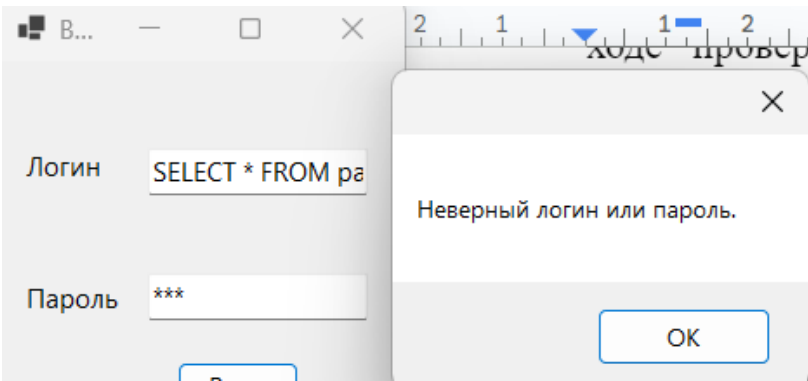


Рисунок 35 – Ошибка при вводе SQL-инъекции

	id заявителя bigint	id сотрудника bigint	пароль text	логин [PK] text	админ boolean
1	[null]	5	B59C67BF196A4758191E42F76670CEBA	admin	true
2	[null]	2	827CCB0EEA8A706C4C34A16891F84E7B	admin1	true
3	1	[null]	202CB962AC59075B964B07152D234B70	zaya	false

Рисунок 36 – Логин в БД

При нажатии кнопки «Вход» происходит переход в одно из следующих окон: окно создания пользователя или окно выбора данных. Код программы данной части представлен на рисунке 37.

```

if (idЗаявителя == DBNull.Value && idСотрудника == DBNull.Value)
{
    MessageBox.Show("Вас нет в базе.");
}
else if (idСотрудника != DBNull.Value && isAdmin)
{
    create_user create_user = new create_user();
    create_user.Show();
    this.Hide();
}
else if (idЗаявителя != DBNull.Value)
{
    choose_data choose_data = new choose_data();
    choose_data.Show();
    this.Hide();
}
// Здесь можно добавить обработку для других ролей

```

Рисунок 37 –Переход в последующие окна

Так как не удалось подключить БД, как источник данных, в каждой новой форме необходимо подключать вручную через строку подключения, которая была описана ранее. Также по причине того, что в формах создания пользователя и изменения пароля используется данные которые необходимо скрывать, в каждой форме прописана функция хэширования.

При создании нового пользователя, необходимо указать id заявителя или сотрудника в зависимости от того кем является пользователь, логин пользователя и пароль, который будет зашифрован. Также необходимо указать является ли пользователь администратором который может создавать пользователей. Для защиты от случайной выдачи прав администратора заявителю, при проверки заполненного поля «id заявителя», поставлена проверка на выдачу прав администратора (рисунок 38).

```

// Проверка на дублирование ID заявителя только если он указан
if (applicantId != null)
{
    using (var command = new NpgsqlCommand("SELECT COUNT(*) FROM password WHERE \"id заявителя\" = @applicantId", con
    {
        command.Parameters.AddWithValue("@applicantId", applicantId);
        int count = Convert.ToInt32(command.ExecuteScalar());
        if (count > 0)
        {
            MessageBox.Show("Пользователь с таким ID заявителя уже существует.");
            return;
        }
    }
    if (isAdmin)
    {
        MessageBox.Show("Пользователь с ID заявителя не может быть администратором.");
        return;
    }
}
}

```

Рисунок 38 – Часть кода с проверкой

Для проверки работоспособности была попытка выдачи прав администратора заявителю (рисунок 39).

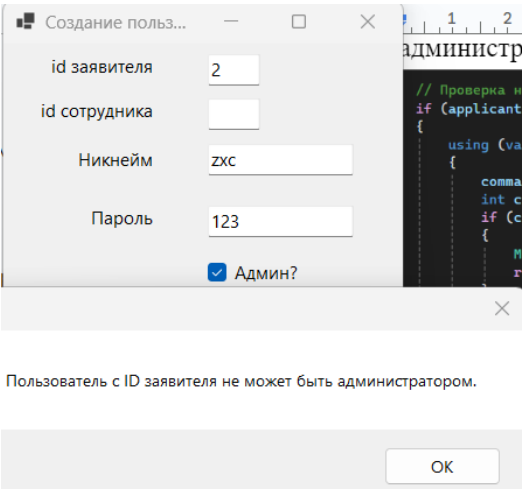


Рисунок 39 – Работоспособность проверки

Также была проверена БД на наличие данной записи (рисунок 40).

	id заявителя bigint	id сотрудника bigint	пароль text	логин [PK] text	админ boolean
1	[null]	5	B59C67BF196A4758191E42F76670CEBA	admin	true
2	[null]	2	827CCB0EEA8A706C4C34A16891F84E7B	admin1	true
3	1	[null]	202CB962AC59075B964B07152D234B70	zaya	false
4	2	[null]	202CB962AC59075B964B07152D234B70	zxc	false

Рисунок 40 – Отсутствие у записи «zxc» права админа

При создании пользователя его данные также переводятся в параметры, после чего направляются на запись в БД. Запись в БД реализовано через команду «INSERT INTO», экранирование и строка вноса данных показано на рисунке 41.

```
// Создаем нового пользователя в базе данных
using (var command = new NpgsqlCommand("INSERT INTO password (логин, пароль, \"id заявителя\", \"id сотрудника\", адм
{
    command.Parameters.AddWithValue("@login", login);
    command.Parameters.AddWithValue("@password", hashedPassword);
    command.Parameters.AddWithValue("@applicantId", applicantId == null ? DBNull.Value : (object)applicantId);
    command.Parameters.AddWithValue("@employeeId", employeeId == null ? DBNull.Value : (object)employeeId);
    command.Parameters.AddWithValue("@isAdmin", isAdmin);
    command.ExecuteNonQuery();
}
```

Рисунок 41 – Часть кода с внесением данных в БД

Для ускорения работоспособности формы которые были открыты ранее скрываются и переходят в фоновый режим работы. Но при переходе на форму авторизации, формы которые были открыты до этого закрываются.

При изменении пароля в следующей форме, необходимо указать существующий логин и новый пароль. При заполнении поля «логин» происходит проверка на наличие такого логина в БД, в то время как пароль так же шифруется и передается в БД в качестве параметра, что вновь защищает от одной SQL-инъекции. Обновление пароля происходит через команду «UPDATE» (рисунок 42).

```
using (var connection = new NpgsqlConnection(connectionString))
{
    connection.Open();
    using (var command = new NpgsqlCommand("UPDATE password SET пароль = @hashedNewPassword WHERE логин = @login", connection))
    {
        command.Parameters.AddWithValue("@hashedNewPassword", hashedNewPassword);
        command.Parameters.AddWithValue("@login", login);

        int rowsAffected = command.ExecuteNonQuery();

        if (rowsAffected > 0)
        {
            MessageBox.Show("Пароль успешно изменен.");
        }
        else
        {
            MessageBox.Show("Пользователь с таким логином не найден.");
        }
    }
}
```

Рисунок 42– Обновление пароля в коде

Также была прописана функция которая проверяет чтобы новый пароль не совпадал со старым (рисунок 43).

```
// Проверяем, не совпадает ли новый пароль со старым
using (var connection = new NpgsqlConnection(connectionString))
{
    connection.Open();
    using (var command = new NpgsqlCommand("SELECT пароль FROM password WHERE логин = @login", connection))
    {
        command.Parameters.AddWithValue("@login", login);
        string oldHashedPassword = command.ExecuteScalar()?.ToString();

        if (oldHashedPassword != null && hashedNewPassword == oldHashedPassword)
        {
            MessageBox.Show("Новый пароль не должен совпадать со старым.");
            return;
        }
    }
}
```

Рисунок 43 – Код проверки на новый пароль

Проверка на работоспособность кода, показана на рисунках 43-44. Также была проверена БД на изменение пароля. Хэш старого пароля пользователя «zxc» можно увидеть на рисунке 40, а нового пароля на рисунке 45.

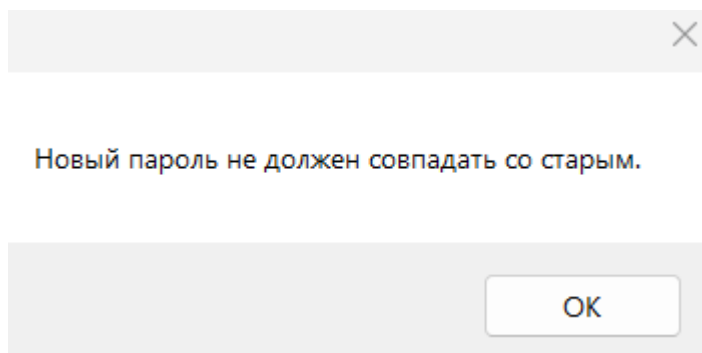


Рисунок 43 – Попытка написание нового пароля, который совпадает со старым.

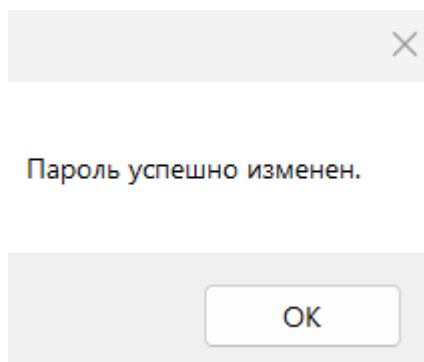


Рисунок 44 – Успешность смены пароля

	id заявителя bigint	id сотрудника bigint	пароль text	логин [PK] text	админ boolean
1	[null]	5	B59C67BF196A4758191E42F76670CEBA	admin	true
2	[null]	2	827CCB0EEA8A706C4C34A16891F84E7B	admin1	true
3	1	[null]	202CB962AC59075B964B07152D234B70	zaya	false
4	2	[null]	827CCB0EEA8A706C4C34A16891F84E7B	zxc	false

Рисунок 45 – Изменение пароля в БД

Заключение

В ходе выполнения данной курсовой работы была создана АИС с учебно-исследовательской базой данных для ЗАГС. Также спроектирована и создана соответствующая база данных, разграничение доступа на уровне СУБД и программном уровне. Были получены навыки работы со справочными системами, для определения федеральных законов и законодательных актов.

Для реализации данной курсовой работы были реализованы следующие пункты:

- Изучение работы с СУБД postgresSQL
- Способы подключения БД к программе
- Изучение законодательства РФ, в сфере обеспечения безопасности персональных данных

Список использованных источников

1. Образовательный стандарт вуза ОС ТУСУР 01-2021 [Электронный ресурс]: сайт ТУСУРа. URL: <https://regulations.tusur.ru/documents/70> (дата обращения: 01.10.2024).
2. Учебный курс безопасность систем баз данных [Электронный ресурс]: сайт системы дистанционного образования ТУСУРа. URL: <https://sdo.tusur.ru/course/view.php?id=2121> (дата обращения 03.09.2024).
3. Документация по работе с PostgreSQL [Электронный ресурс]: сайт PostgreSQL. URL: <https://postgrespro.ru/docs/postgresql/17/index> (дата обращения: 09.09.2024).
4. Федеральный закон №152 «О персональных данных» [Электронный ресурс]: сайт КонсультантПлюс. URL: https://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 29.10.2024).
5. Постановление правительства №1119 «Об утверждении требований к защите персональных данных при обработке в информационных системах персональных данных» [Электронный ресурс]: сайт КонсультантПлюс. URL: https://www.consultant.ru/document/cons_doc_LAW_137356/ (дата обращения: 01.11.2024).
6. Приказ Федеральной службы по техническому и экспортному контролю №21 «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных» [Электронный ресурс]: сайт ФСТЭК России. URL: <https://fstec.ru/dokumenty/vse-dokumenty/prikazy/prikaz-fstek-rossii-ot-18-fevralya-2013-g-n-21> (дата обращения: 11.11.2024).

Приложение А

(Справочное)

Определение пунктов мер обеспечения безопасности персональных данных

Согласно приказу ФСТЭК №21 18 февраля 2013 года «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных», для обеспечения 3 уровня защищенности персональных данных необходимо принять следующие меры:

- Идентификация и аутентификация субъектов доступа и объектов доступа:

1. Идентификация и аутентификация пользователей, являющихся работниками оператора;

2. Управление идентификаторами, в том числе создание, присвоение, уничтожение идентификаторов;

3. Управление средствами аутентификации, в том числе хранение, выдача, инициализация, блокирование средств аутентификации и принятие мер в случае утраты и (или) компрометации средств аутентификации;

4. Защита обратной связи при вводе аутентификационной информации;

5. Идентификация и аутентификация пользователей, не являющихся работниками оператора (внешних пользователей).

- Меры управления доступом субъектов доступа к объектам доступа:

1. Управление (заведение, активация, блокирование и уничтожение) учетными записями пользователей, в том числе внешних пользователей;

2. Реализация необходимых методов (дискреционный, мандатный, ролевой или иной метод), типов (чтение, запись, выполнение или иной тип) и правил разграничения доступа;

3. Управление (фильтрация, маршрутизация, контроль соединений, однонаправленная передача и иные способы управления) информационными потоками между устройствами, сегментами информационной системы, а также между информационными системами;

4. Разделение полномочий (ролей) пользователей, администраторов и лиц, обеспечивающих функционирование информационной системы;

5. Назначение минимально необходимых прав и привилегий пользователям, администраторам и лицам, обеспечивающим функционирование информационной системы;

6. Ограничение неуспешных попыток входа в информационную систему (доступа к информационной системе);

7. Блокирование сеанса доступа в информационную систему после установленного времени бездействия (неактивности) пользователя или по его запросу;

8. Разрешение (запрет) действий пользователей, разрешенных до идентификации и аутентификации;

9. Реализация защищенного удаленного доступа субъектов доступа к объектам доступа через внешние информационно-телекоммуникационные сети;

10. Регламентация и контроль использования в информационной системе технологий беспроводного доступа;

11. Регламентация и контроль использования в информационной системе мобильных технических средств;

12. Управление взаимодействием с информационными системами сторонних организаций (внешние информационные системы).

- В мерах ограничения программной среды нет необходимости. Меры защиты машинных носителей персональных данных:

1. Уничтожение (стирание) или обезличивание персональных данных на машинных носителях при их передаче между пользователями, в сторонние организации для ремонта или утилизации, а также контроль уничтожения (стирания) или обезличивания.

- Меры регистрация событий безопасности:

1. Определение событий безопасности, подлежащих регистрации, и сроков их хранения;

2. Определение состава и содержания информации о событиях безопасности, подлежащих регистрации;

3. Сбор, запись и хранение информации о событиях безопасности в течение установленного времени хранения;

4. Защита информации о событиях безопасности.

Меры антивирусной защиты:

1. Реализация антивирусной защиты;

2. Обновление базы данных признаков вредоносных компьютерных программ (вирусов).

Согласно данному приказу при 3 уровне защищенности персональных данных в мерах обнаружения вторжения нет необходимости. Меры по контролю (анализу) защищенности персональных данных:

1. Выявление, анализ уязвимостей информационной системы и оперативное устранение вновь выявленных уязвимостей;

2. Контроль установки обновлений программного обеспечения, включая обновление программного обеспечения средств защиты информации;

3. Контроль работоспособности, параметров настройки и правильности функционирования программного обеспечения и средств защиты информации;

4. Контроль состава технических средств, программного обеспечения и средств защиты информации.

- Для обеспечения целостности информационной системы и персональных данных, необходимы следующие меры:

1. Контроль целостности персональных данных, содержащихся в базах данных информационной системы;

2. Обеспечение возможности восстановления программного обеспечения, включая программное обеспечение средств защиты информации, при возникновении нештатных ситуаций;

3. Контроль содержания информации, передаваемой из информационной системы (контейнерный, основанный на свойствах объекта доступа, и (или) контентный, основанный на поиске запрещенной к передаче информации с

использованием сигнатур, масок и иных методов), и исключение неправомерной передачи информации из информационной системы;

4. Ограничение прав пользователей по вводу информации в информационную систему;

5. Контроль точности, полноты и правильности данных, вводимых в информационную систему;

6. Контроль ошибочных действий пользователей по вводу и (или) передаче персональных данных и предупреждение пользователей об ошибочных действиях.

- Меры по обеспечению доступности персональных данных:

1. Использование отказоустойчивых технических средств;

2. Периодическое резервное копирование персональных данных на резервные машинные носители персональных данных;

3. Обеспечение возможности восстановления персональных данных с резервных машинных носителей персональных данных (резервных копий) в течение установленного временного интервала.

- Меры по защите среды виртуализации:

1. Идентификация и аутентификация субъектов доступа и объектов доступа в виртуальной инфраструктуре, в том числе администраторов управления средствами виртуализации;

2. Управление доступом субъектов доступа к объектам доступа в виртуальной инфраструктуре, в том числе внутри виртуальных машин;

3. Регистрация событий безопасности в виртуальной инфраструктуре;

4. Доверенная загрузка серверов виртуализации, виртуальной машины (контейнера), серверов управления виртуализацией;

5. Реализация и управление антивирусной защитой в виртуальной инфраструктуре;

6. Разбиение виртуальной инфраструктуры на сегменты (сегментирование виртуальной инфраструктуры) для обработки персональных данных отдельным пользователем и (или) группой пользователей.

- Меры по защите технических средств:

1. Организация контролируемой зоны, в пределах которой постоянно размещаются стационарные технические средства, обрабатывающие информацию, и средства защиты информации, а также средства обеспечения функционирования;

2. Контроль и управление физическим доступом к техническим средствам, средствам защиты информации, средствам обеспечения функционирования, а также в помещения и сооружения, в которых они установлены, исключающие несанкционированный физический доступ к средствам обработки информации, средствам защиты информации и средствам обеспечения функционирования информационной системы, в помещения и сооружения, в которых они установлены;

3. Размещение устройств вывода (отображения) информации, исключающее ее несанкционированный просмотр;

4. Защита от внешних воздействий (воздействий окружающей среды, нестабильности электроснабжения, кондиционирования и иных внешних факторов).

- Меры по защите информационной системы, ее средств, систем связи и передачи данных:

1. Предотвращение задержки или прерывания выполнения процессов с высоким приоритетом со стороны процессов с низким приоритетом;

2. Обеспечение защиты персональных данных от раскрытия, модификации и навязывания (ввода ложной информации) при ее передаче (подготовке к передаче) по каналам связи, имеющим выход за пределы контролируемой зоны, в том числе беспроводным каналам связи;

3. Обеспечение доверенных канала, маршрута между администратором, пользователем и средствами защиты информации (функциями безопасности средств защиты информации);

4. Запрет несанкционированной удаленной активации видеокамер, микрофонов и иных периферийных устройств, которые могут активироваться удаленно, и оповещение пользователей об активации таких устройств;

5. Защита архивных файлов, параметров настройки средств защиты информации и программного обеспечения и иных данных, не подлежащих изменению в процессе обработки персональных данных;

6. Выявление, анализ и блокирование в информационной системе скрытых каналов передачи информации в обход реализованных мер или внутри разрешенных сетевых протоколов;

7. Защита беспроводных соединений, применяемых в информационной системе.

- Меры по выявлению инцидентов и реагированию на них:

1. Определение лиц, ответственных за выявление инцидентов и реагирование на них;

2. Обнаружение, идентификация и регистрация инцидентов;

3. Своевременное информирование лиц, ответственных за выявление инцидентов и реагирование на них, о возникновении инцидентов в информационной системе пользователями и администраторами;

4. Анализ инцидентов, в том числе определение источников и причин возникновения инцидентов, а также оценка их последствий;

5. Принятие мер по устранению последствий инцидентов;

6. Планирование и принятие мер по предотвращению повторного возникновения инцидентов.

- Меры по управлению конфигурацией информационной системы и системы защиты персональных данных;

1. Определение лиц, которым разрешены действия по внесению изменений в конфигурацию информационной системы и системы защиты персональных данных;

2. Управление изменениями конфигурации информационной системы и системы защиты персональных данных;

3. Анализ потенциального воздействия планируемых изменений в конфигурации информационной системы и системы защиты персональных данных на обеспечение защиты персональных данных и согласование изменений в конфигурации информационной системы с должностным лицом (работником), ответственным за обеспечение безопасности персональных данных;

4. Документирование информации (данных) об изменениях в конфигурации информационной системы и системы защиты персональных данных.

Приложение Б

(Обязательное)

Листинг программы

Программа представлена на сайте GitHub. Ссылка на проект:
<https://github.com/Tigrric/kursachacha>

Приложение В

(справочное)

Как подключиться к базе данных, как источнику данных в Visual studio

Для подключения к БД необходимо установить драйвера. Драйвера устанавливаются в программе «application stack builder» (рисунок В.1).

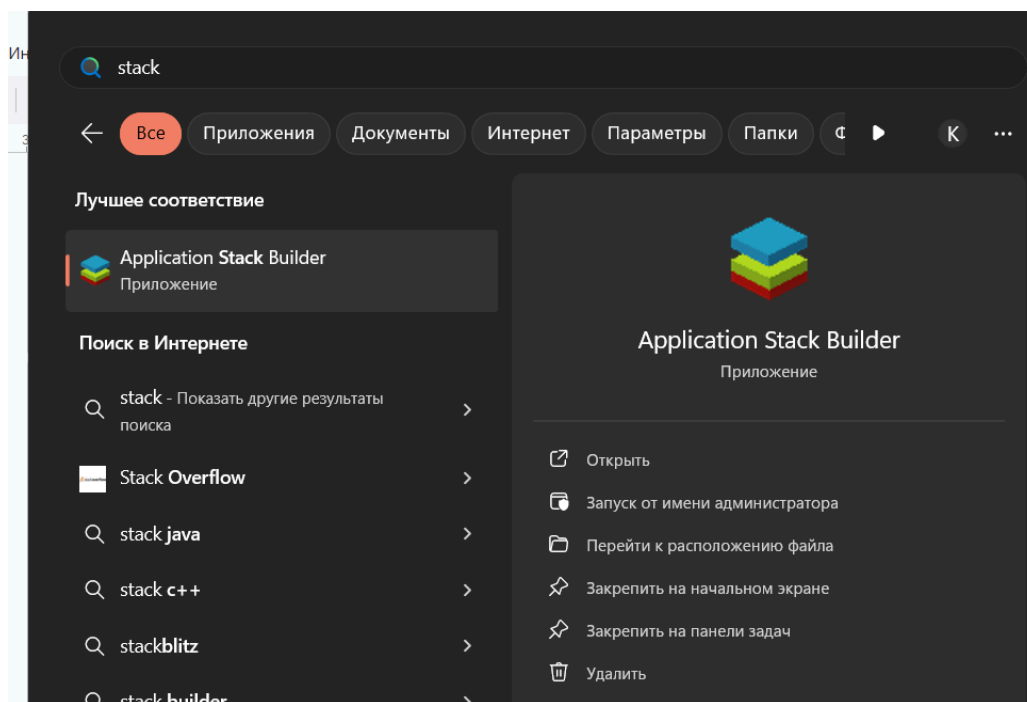


Рисунок В.1 – Поиск программы для установки драйверов

Далее необходимо выбрать подключение сервера (рисунок В.2). Далее нажать кнопку «next», если установилось на русском языке то нажимаем кнопку «далее».

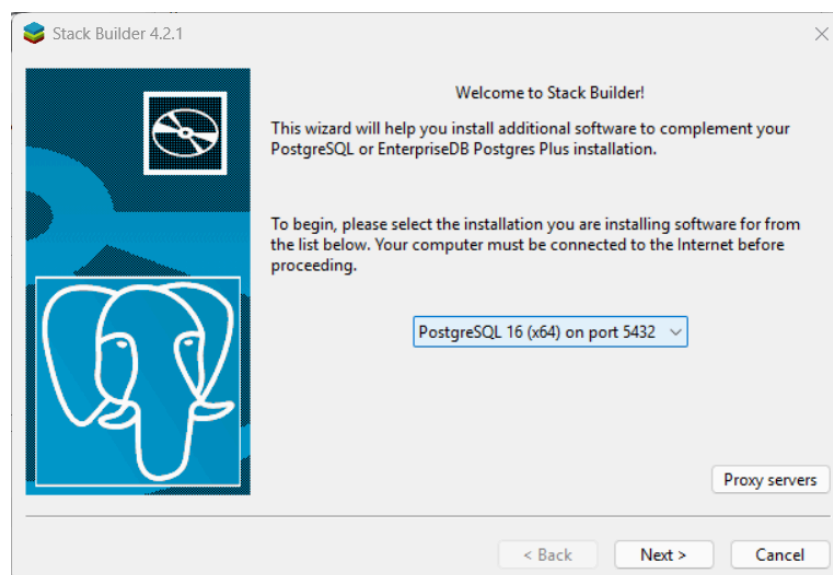


Рисунок В.2 – Выбор сервера подключения

После загрузки страницы, выбираем категорию «Database Drivers» (рисунок В.3). После раскрытия данной категории, ставим галочки напротив следующих пунктов: «Npgsql», «pgJDBC», «psqlODBC». В пункте «psqlODBC» выбираем нужную архитектуру 64 bit или 32 bit.

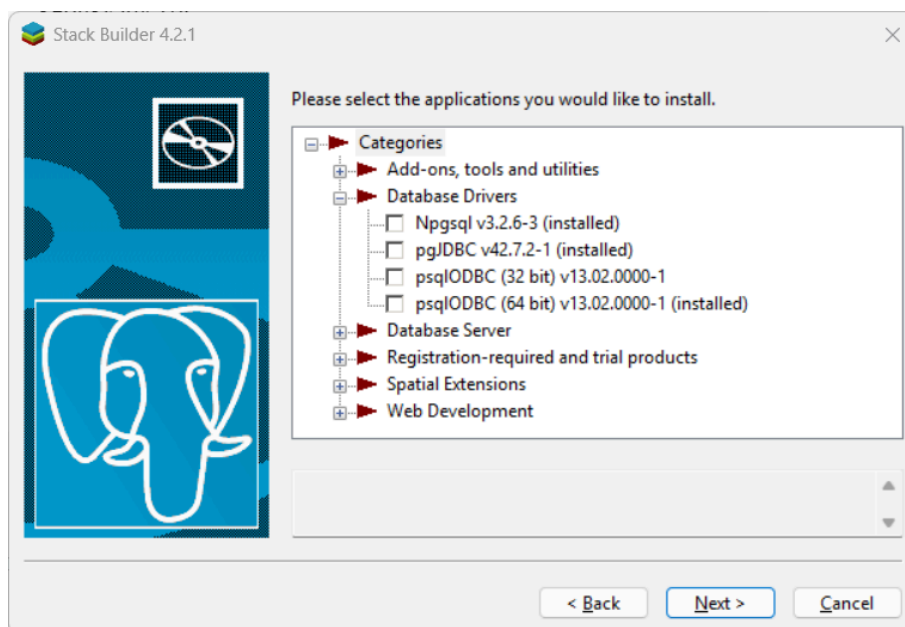


Рисунок В.3 – Раздел с драйверами БД

После этого несколько раз нажимаем кнопку «next», на рисунке В.4 выбираем директорию куда будет скачаны драйвера. В целях обеспечения

конфиденциальности, имя директорий будет скрыто, так как это является информацией относящиеся к персональным данным. В последующем в тексте также будет написано кнопка «next», но стоит помнить, что если у вас на русском языке то необходимо нажимать кнопку «далее».

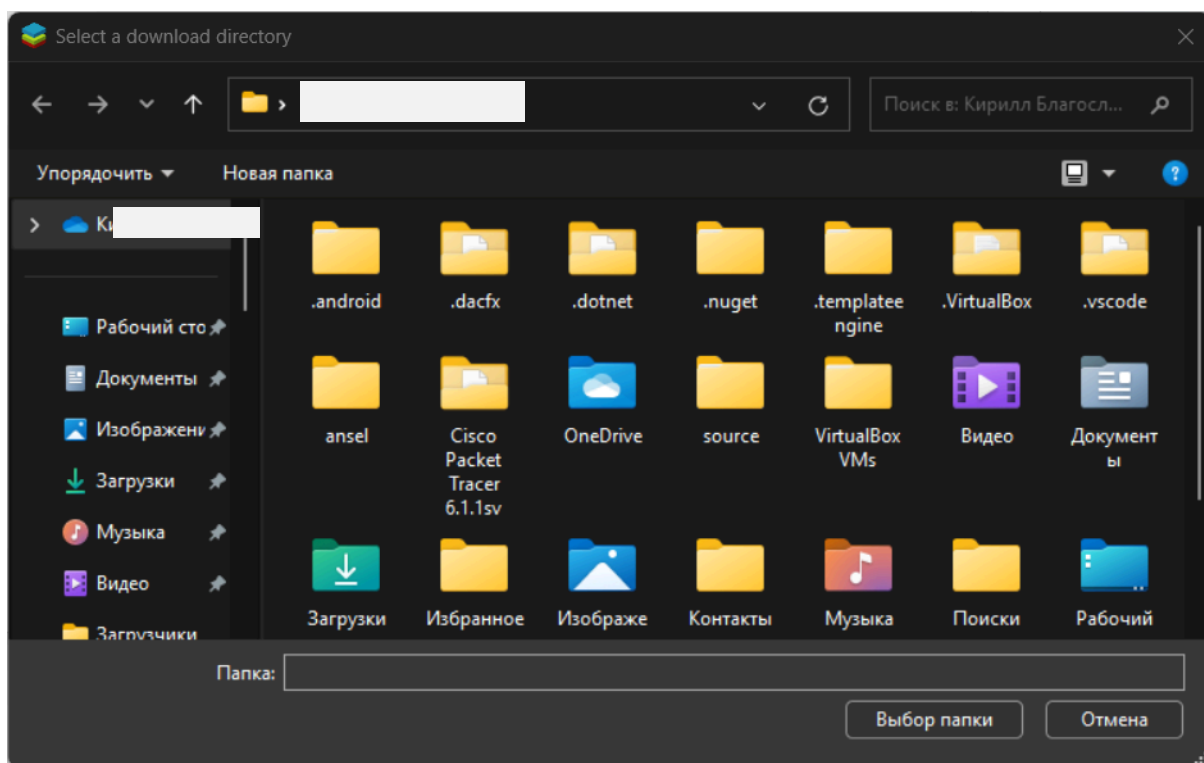


Рисунок В.4 – выбор директории установки

После этого происходит установка подключения к серверу и загрузка необходимых данных. После установки данных, необходимо их распаковать (рисунок В.5), необходимо просто нажать кнопку «next». Откроется окно распаковки (рисунок В.6). На рисунке В.7 также необходимо выбрать директорию установки, просто необходимо нажать «next».

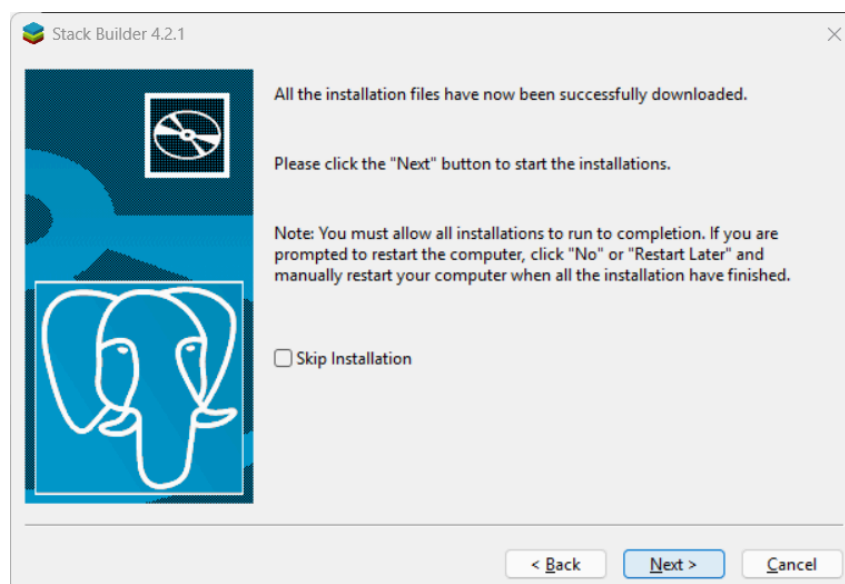


Рисунок В.5 – Окно распаковки драйверов

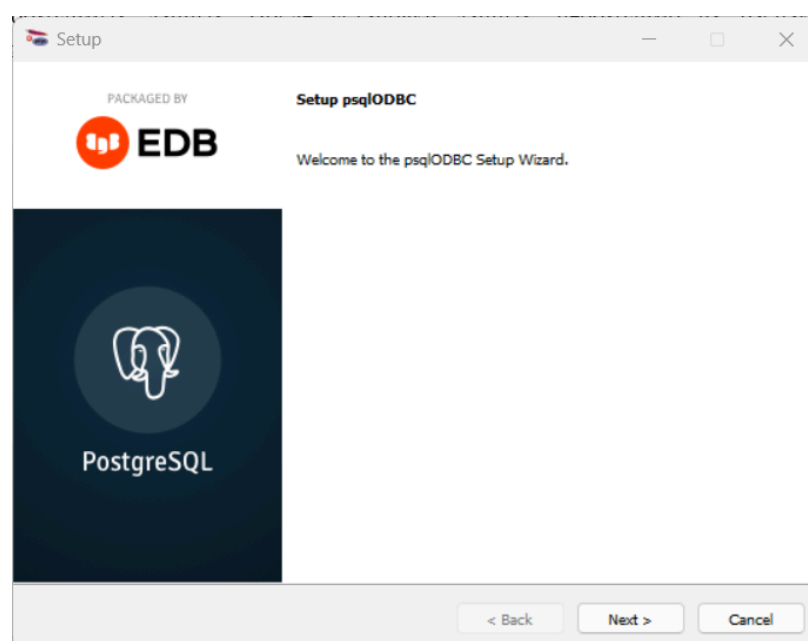


Рисунок В.6 – окно распаковки драйвера «postgresqlODBC»

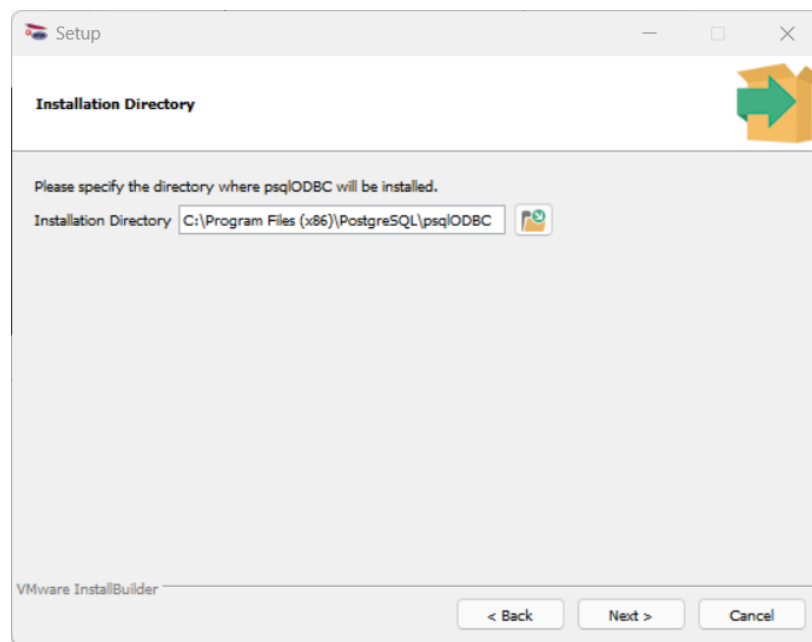


Рисунок В.7 – Выбор директории

После нажатия кнопки «next», начнется распаковка данных драйвера (рисунок В.8). После распаковки данных просто нажимаем «finish» или же «КОНЕЦ».

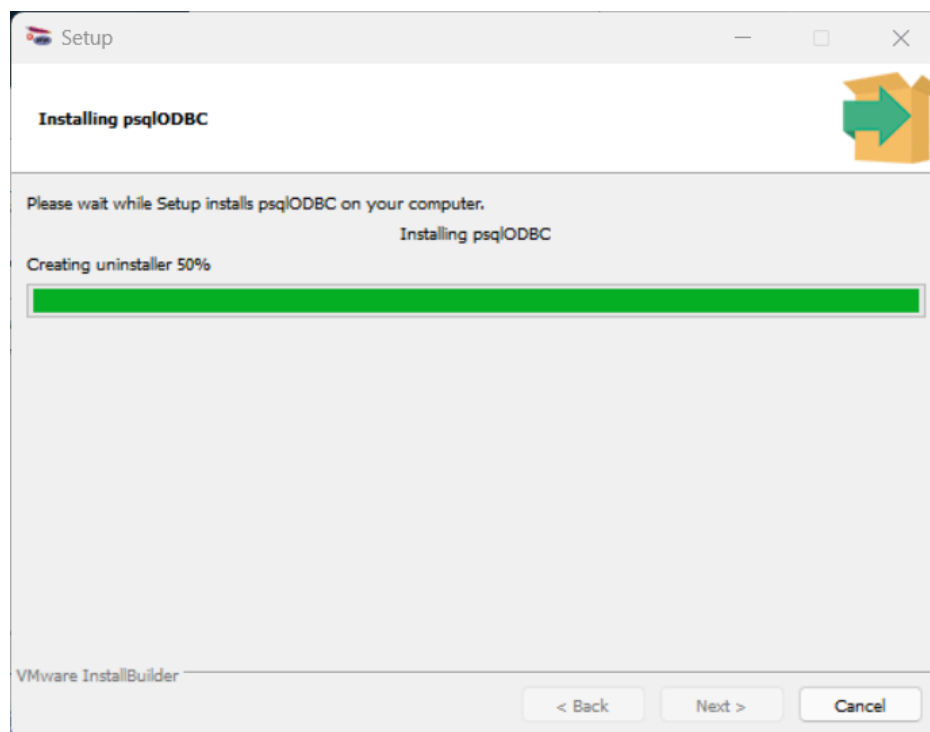


Рисунок В.7 – Процесс распаковки данных

Поздравляем! Вы установили драйвера для подключения в visual studio.

Далее необходимо в Visual studio в верхней панели (рисунок В.8) выбрать вкладку «Вид» (рисунок В.9). В данной вкладке перейти в раздел «другие окна – источник данных» (рисунок В.10).

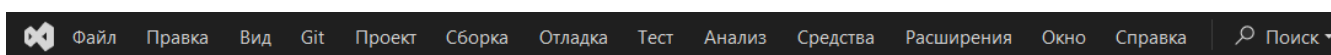


Рисунок В.8 – Панель с вкладками

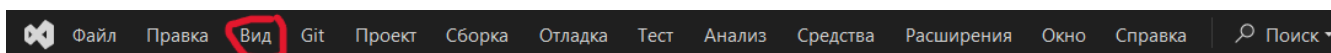


Рисунок В.9 – Вкладка «Вид»

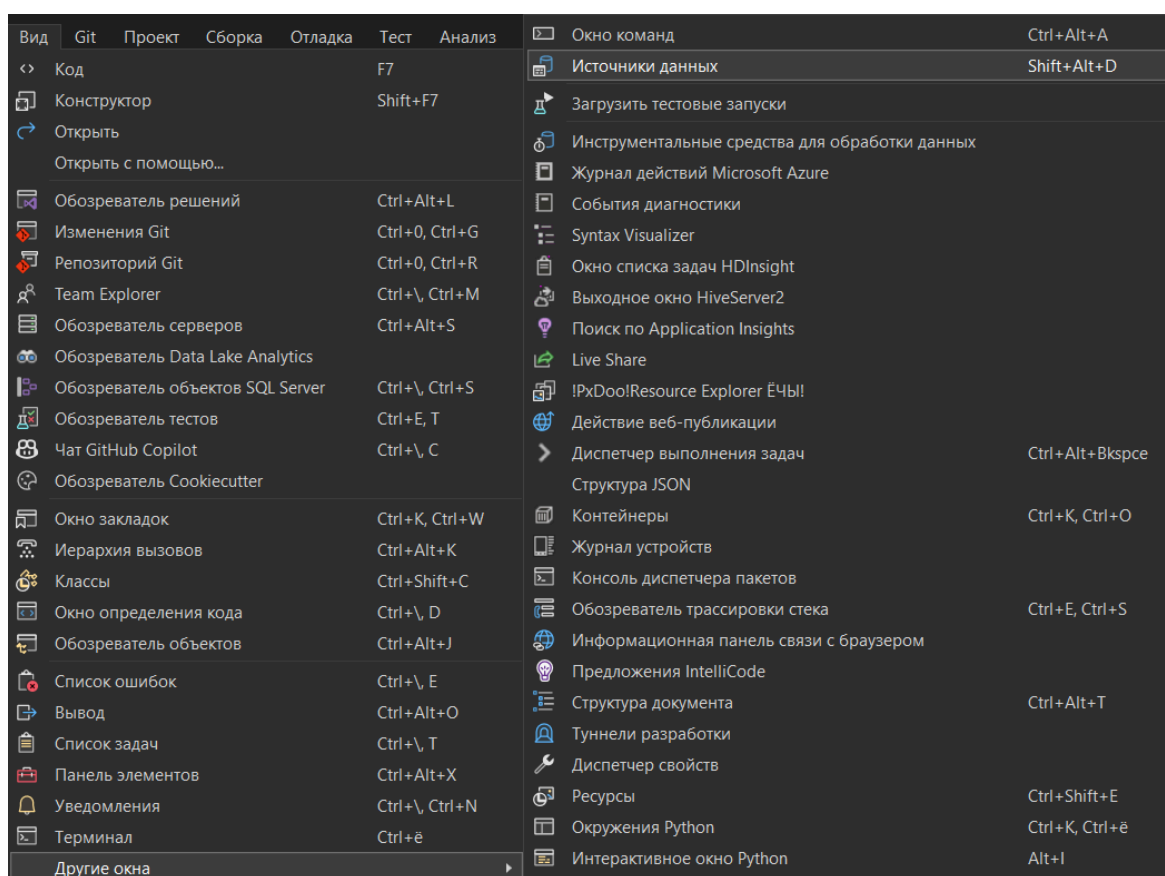


Рисунок В.10 – Источники данных

В случае возникновения ошибки такой же, как и на рисунке В.11, то необходимо будет либо подключать в каждой форме строку подключения, либо же создавать класс с подключением и подключаться через класс. как это реализовано в основном документе пояснительной записки.

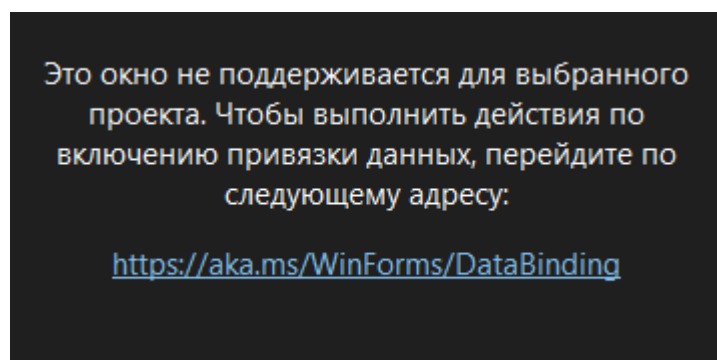


Рисунок В.11 – Невозможность подключения источника данных

Если у вас получилось подключиться, то переходим к следующему пункту. На рисунке В.12 необходимо нажать «Далее».

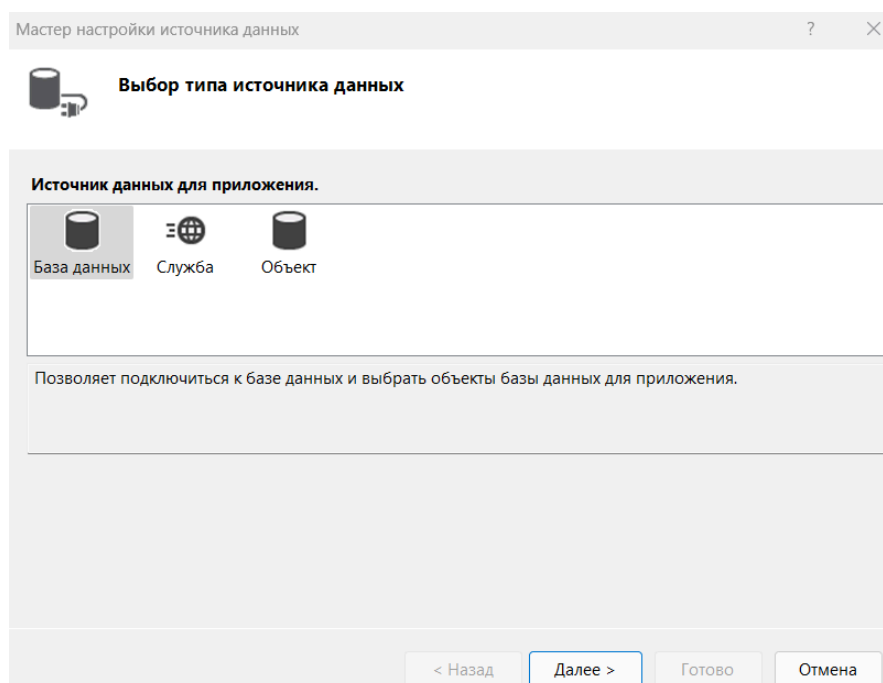
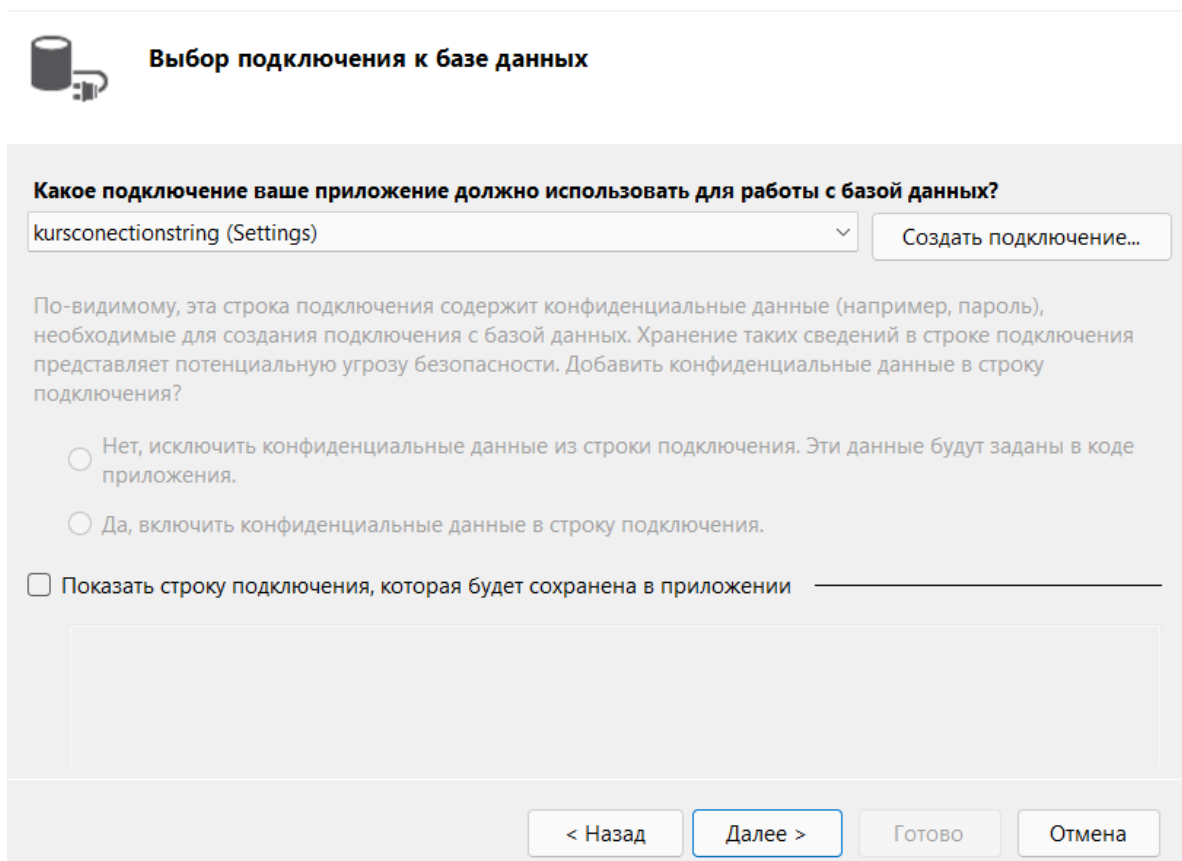


Рисунок В.12 – Выбор источника данных

После этого опять нажать далее. После этого пункта необходимо нажать «создать подключение» (рисунок В.13).



Выбор подключения к базе данных

Какое подключение ваше приложение должно использовать для работы с базой данных?

kursconnectionstring (Settings) Создать подключение...

По-видимому, эта строка подключения содержит конфиденциальные данные (например, пароль), необходимые для создания подключения с базой данных. Хранение таких сведений в строке подключения представляет потенциальную угрозу безопасности. Добавить конфиденциальные данные в строку подключения?

☐ Нет, исключить конфиденциальные данные из строки подключения. Эти данные будут заданы в коде приложения.

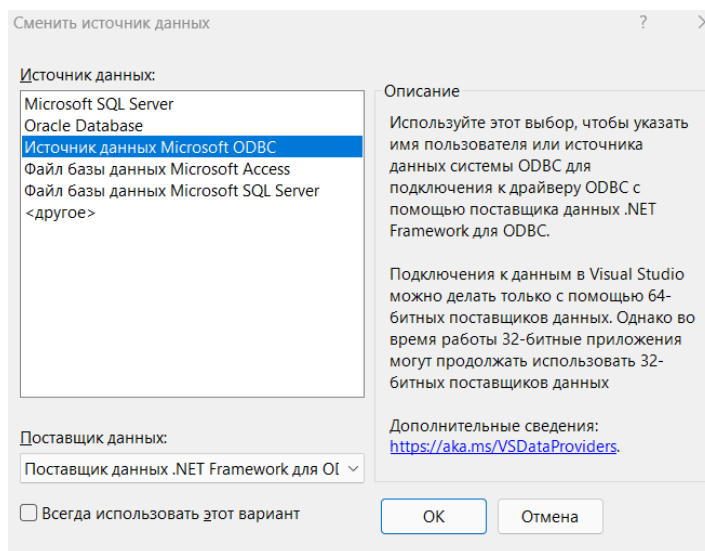
☐ Да, включить конфиденциальные данные в строку подключения.

☐ Показать строку подключения, которая будет сохранена в приложении

< Назад Далее > Готово Отмена

Рисунок В.13 – Создание подключения к БД

После этого необходимо выбрать источник данных ODBC (рисунок В.14)



Сменить источник данных

Источник данных:

- Microsoft SQL Server
- Oracle Database
- Источник данных Microsoft ODBC**
- Файл базы данных Microsoft Access
- Файл базы данных Microsoft SQL Server
- <другое>

Поставщик данных:

Поставщик данных .NET Framework для ODBC

☐ Всегда использовать этот вариант

Описание

Используйте этот выбор, чтобы указать имя пользователя или источника данных системы ODBC для подключения к драйверу ODBC с помощью поставщика данных .NET Framework для ODBC.

Подключения к данным в Visual Studio можно делать только с помощью 64-битных поставщиков данных. Однако во время работы 32-битные приложения могут продолжать использовать 32-битных поставщиков данных

Дополнительные сведения: <https://aka.ms/VSDDataProviders>

ОК Отмена

Рисунок В.14 – Источник данных ODBC

Чтобы продолжить работу необходимо прописать либо строку подключения, либо выбрать из списка системный источник (рисунок В.15). В

рамках данного приложения будет рассмотрен вариант подключения через строку подключения. Может появиться вопрос, чем отличается строка подключения здесь и то что прописано в основном документе? Ответ: в строке подключения, что используется здесь, дополнительно прописывается драйвер, который был установлен ранее.

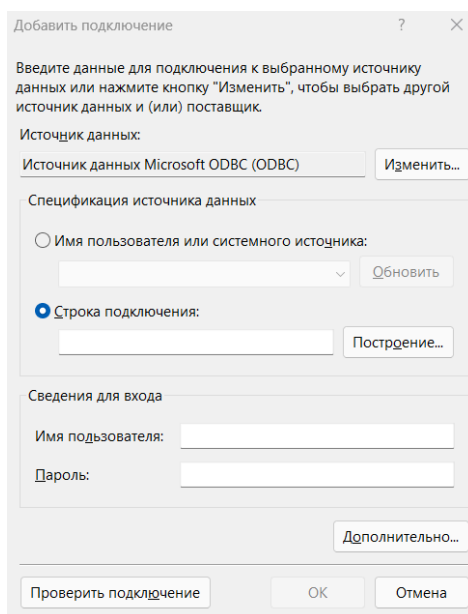


Рисунок В.15 – выбор спецификации

Далее необходимо вписать следующую строку:

```
Driver={PostgreSQL ODBC Driver(UNICODE)}; Server=<server>; Port=<port>;  
Database=<database>; UID=<user id>; PWD=<password>
```

В данной строке необходимо заменить все параметры что помечены как <server>, после этого необходимо нажать «Проверить подключение» (рисунок В.16).

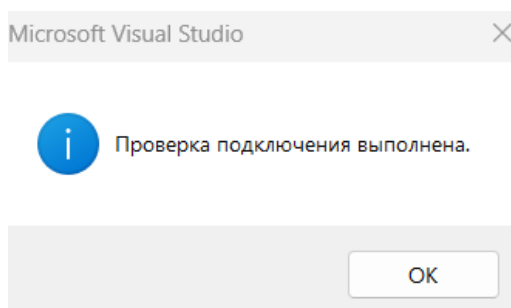


Рисунок В.16 – Успешная проверка подключения

После этого необходимо нажать «ок», как только вернетесь на окно, которое показано на рисунке В.13, необходимо нажать «далее». Откроется окно выбора объектов БД (рисунок В.17). Необходимо дать название набору данных или же оставить стандартное. Далее необходимо раскрыть список таблиц (рисунок В.18) и выбрать необходимые таблицы (рисунок В.19)

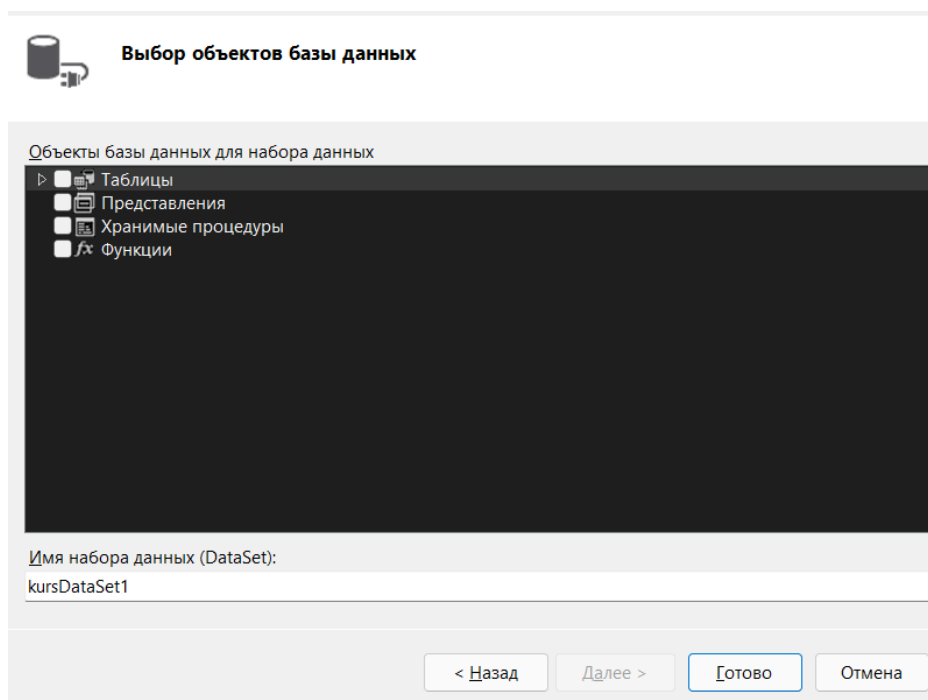


Рисунок В.17 – Выбор объектов БД

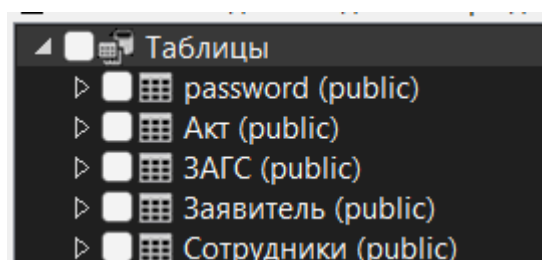


Рисунок В.18 – Таблицы

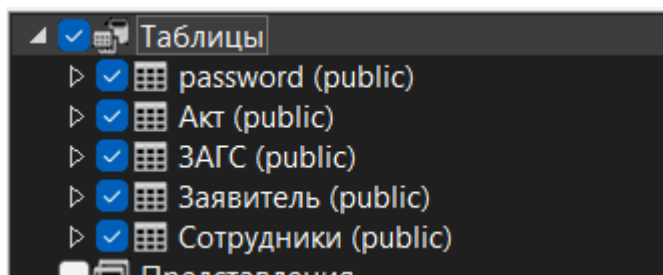


Рисунок В.19 – Выбор необходимых таблиц

Теперь во вкладке «Источник данных», можно видеть подключенную БД (рисунок В.20).

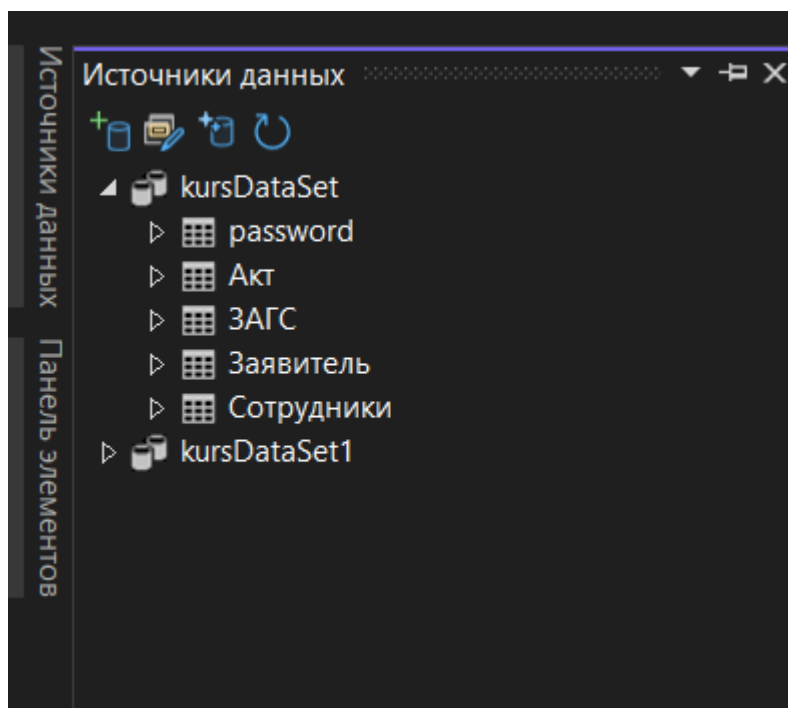


Рисунок В.20 – Источники данных

Теперь после подключения БД, для корректной работы команд подключения к БД, а также работы с БД, необходимо подгрузить библиотеку через установщик пакетов NuGet. Чтобы открыть его необходимо правой кнопкой мыши нажать на проект, после чего выбрать необходимый пункт (рисунок В.21).

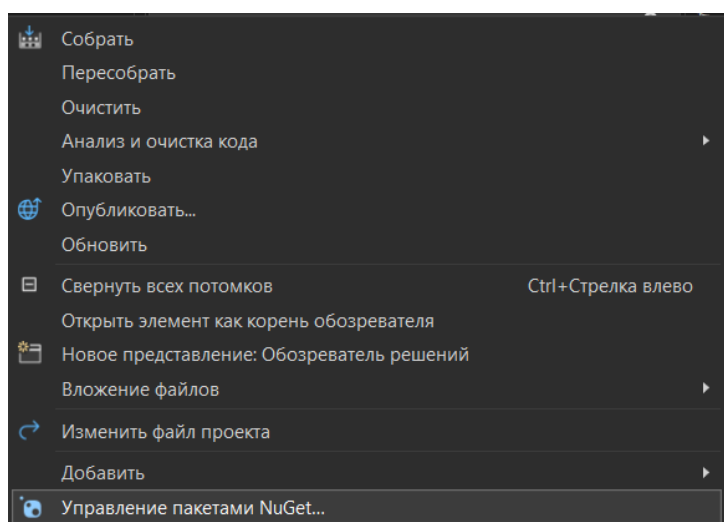


Рисунок В.21 – Установщик пакетов

После этого перейти в раздел «Обзор» и в поиске написать «npgsql» (рисунок В.22). После выбрать необходимый пакет, если работать через winform (Microsoft) необходимо выбрать первый пакет (рисунок В.23), если работа происходит через framework, то необходимо выбрать пакет с названием «EntityFrameworkCore» (рисунок В.24).

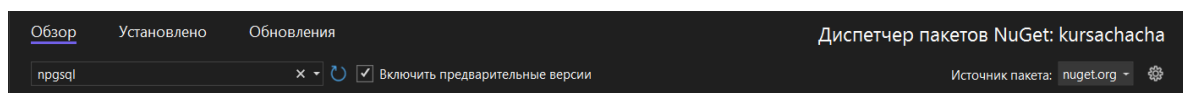


Рисунок В.22 – поиск пакета

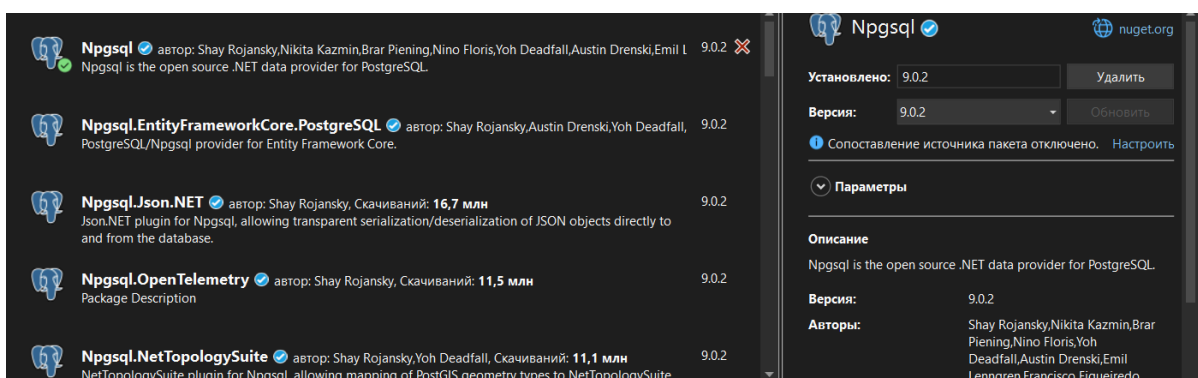


Рисунок В.23 – Пакет «Npgsql»

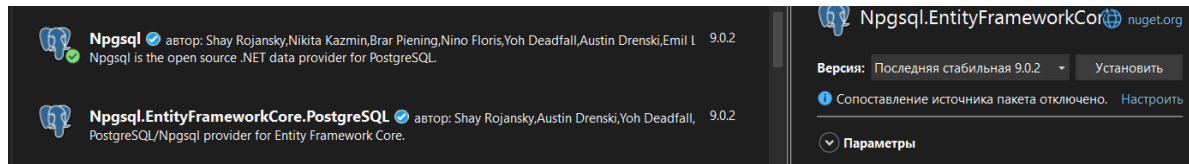


Рисунок В.24 – Пакет «EntityFrameworkCore»

После выбора необходимого пакета, нужно нажать «Установить», после чего начнется установка пакета. По окончании установки пакет можно посмотреть в разделе «Установлено» (рисунок В.25).

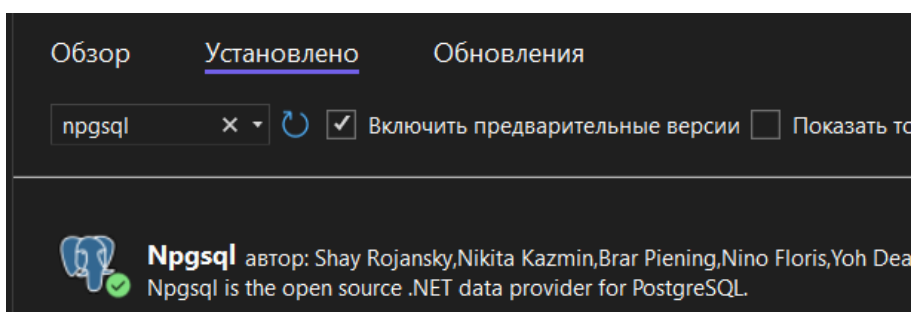


Рисунок В.25 – Установленный пакет

Приложение Г

(справочное)

Установка СУБД

Для установки СУБД необходимо перейти по следующей ссылке – <https://www.postgresql.org/download/windows/>. Вместе с самой СУБД будет установлен и графический элемент «pgAdmin4», но настоятельно рекомендуется установить данную программу отдельно по ссылке: <https://www.pgadmin.org/download/pgadmin-4-windows/>.

Установка проводилась на ОС Windows 10 на виртуальной машине, установка в ОС Windows 11 происходит аналогичным образом.

После установки загрузчика, открываем его. На рисунке Г.1 показано окно которое показывается после открытия, необходимо нажать «next». После этого необходимо выбрать директорию куда будет установлена программа (рисунок Г.2), в данном окне необходимо выбрать путь установки и нажать «next».

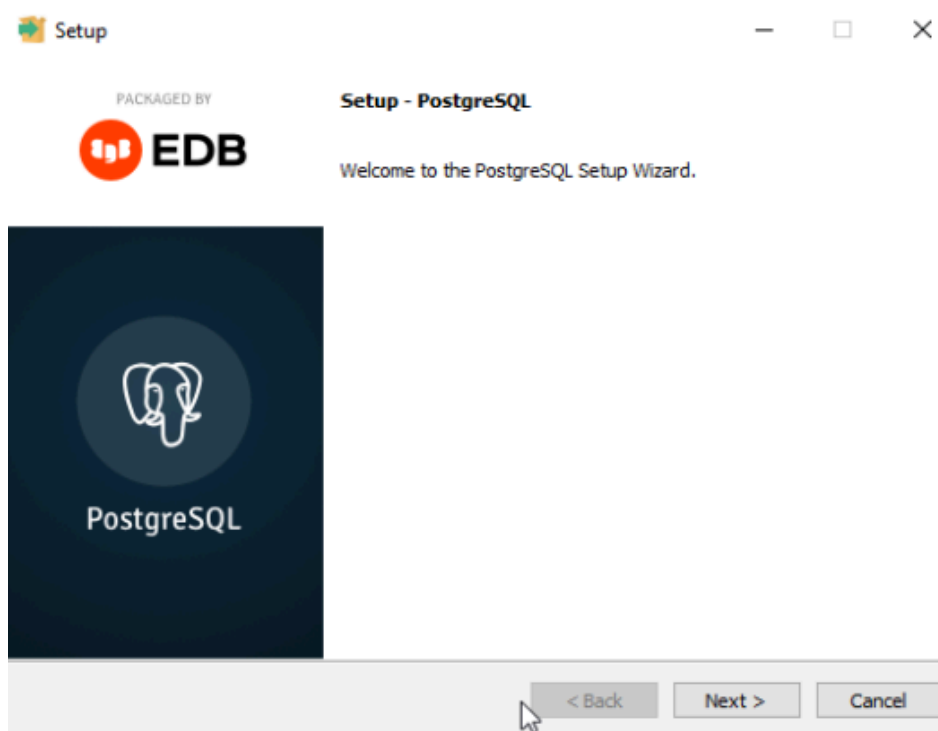


Рисунок Г.1 – Начальное окно установщика

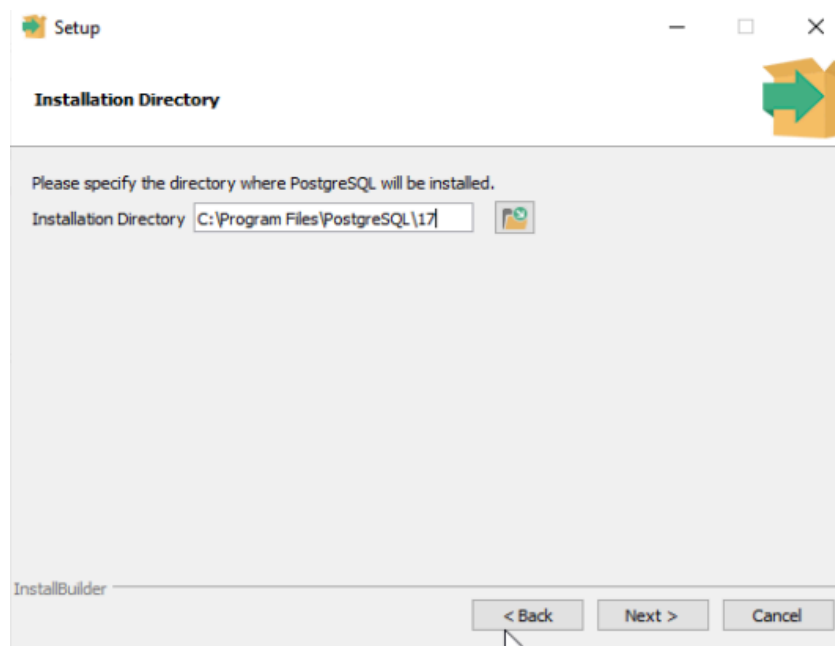


Рисунок Г.2 – Выбор директории

После нажатия кнопки появиться окно с выбором установочных программ (рисунок Г.3), если не выбраны программы, то необходимо выбрать все программы и нажать «next».

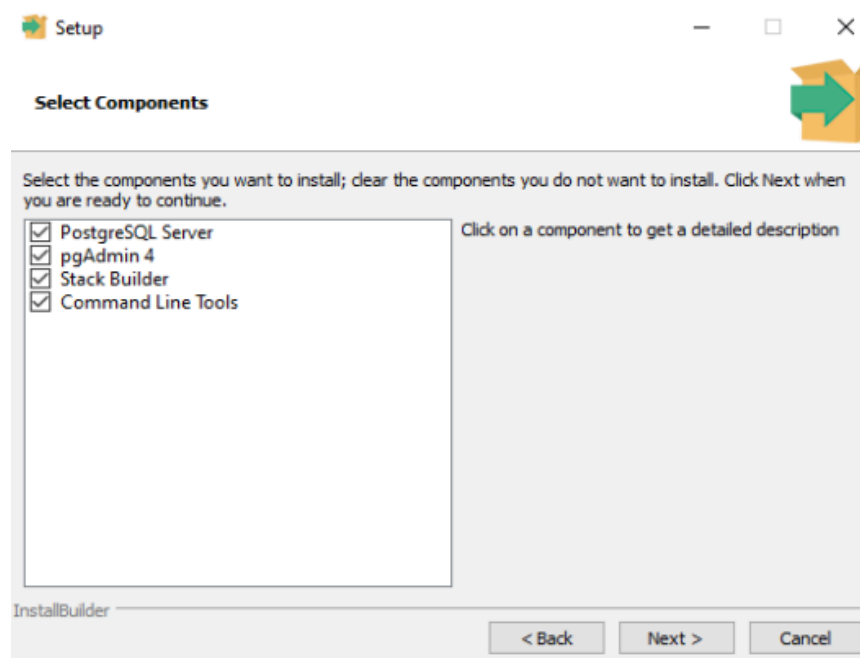


Рисунок Г.3 – Установка необходимых программ

Теоретическая часть. PostgreSQL Server – программа которая соответственно является СУБД postgres.

PgAdmin4 – программа являющаяся графической оболочкой.

Stack Builder – установщик дополнительных драйверов для postgres.

Command Line Tools — инструменты для работы с СУБД через командную строку.

После нажатия «next», необходимо будет выбрать путь по которому будут сохраняться данные, совет не менять путь сохранения (рисунок Г.4).

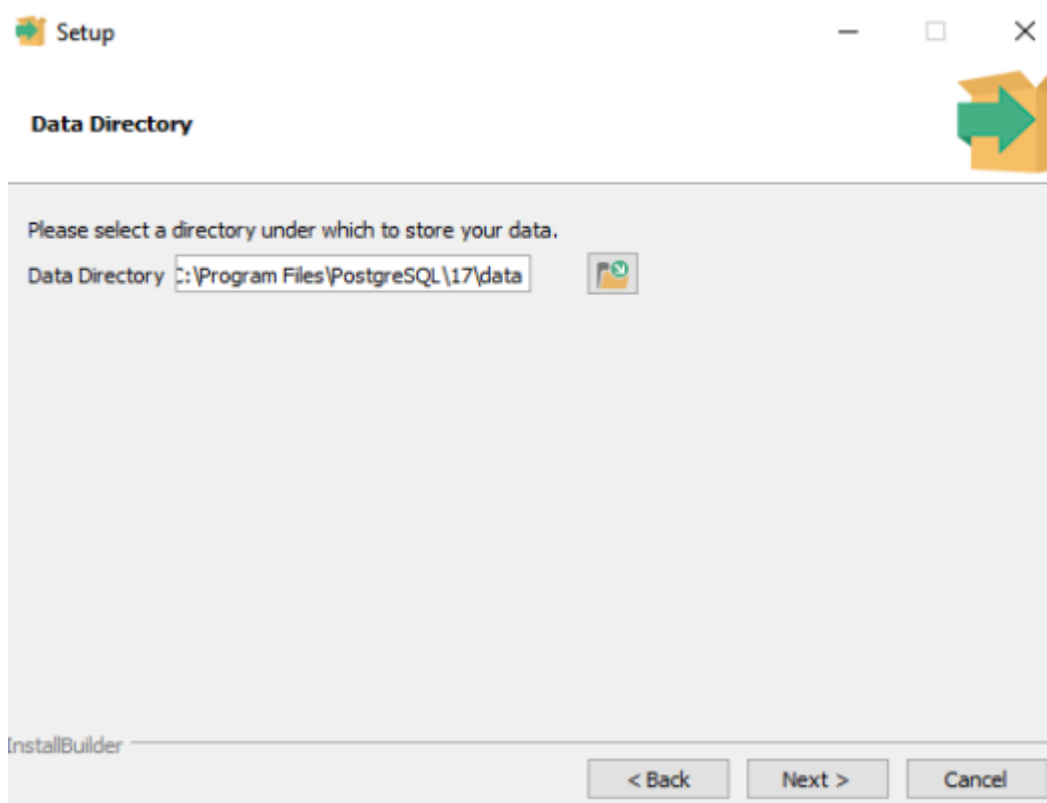


Рисунок Г.4 – Путь сохранение данных

После выбора пути необходимо ввести пароль входа для суперпользователя (рисунок Г.5). Совет: не используйте сложные пароли, в рамках лабораторных работ.

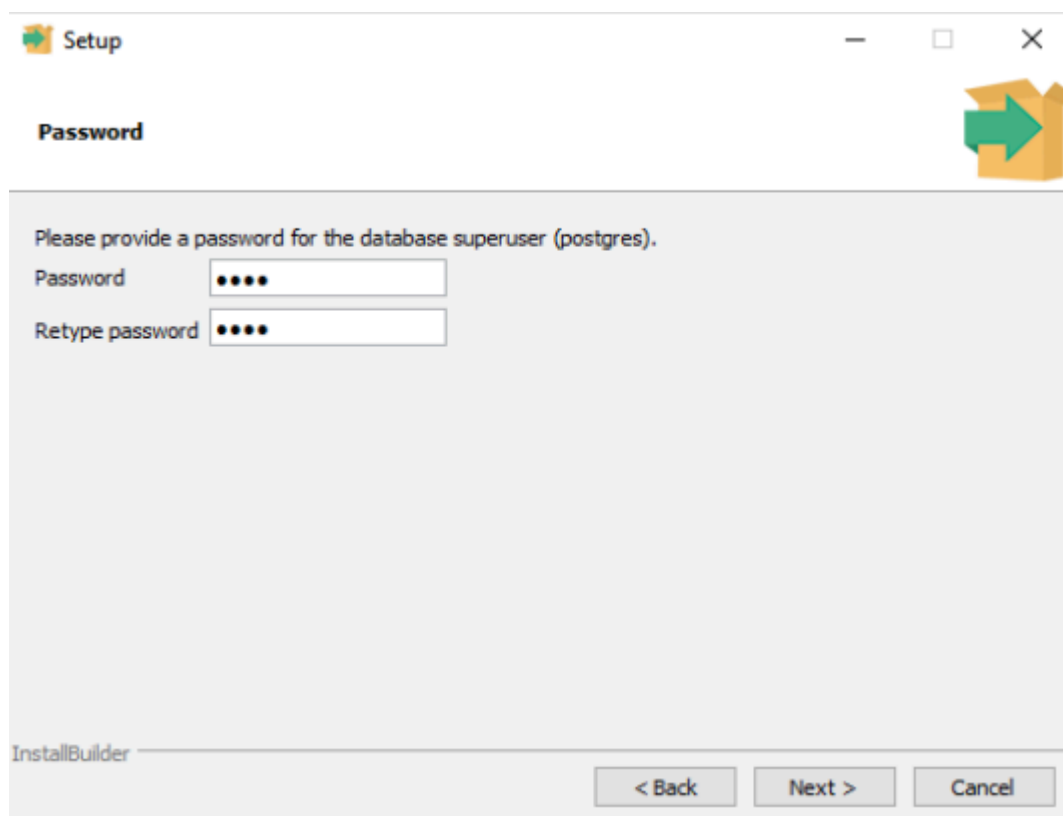


Рисунок Г.5 – Ввод пароля

В следующем окне необходимо указать порт, стандартный порт для СУБД «5432» (рисунок Г.6).

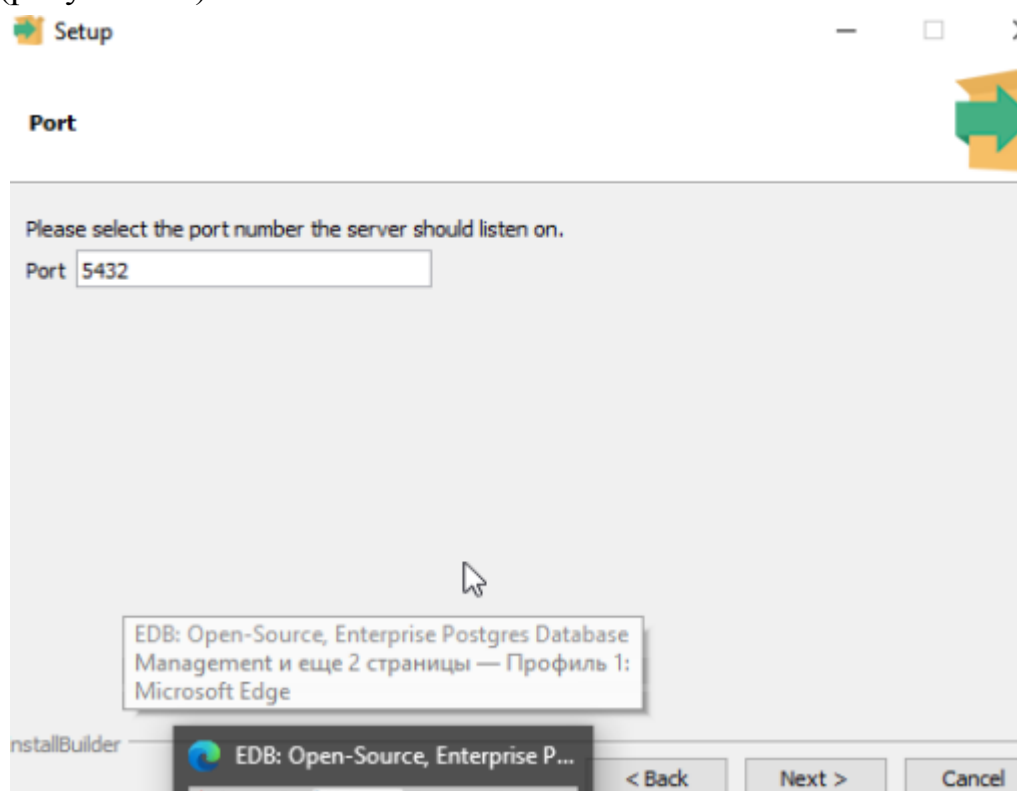


Рисунок Г.6 – Настройка порта

После настройки порта необходимо выбрать язык, можно оставить стандартным (рисунок Г.7).

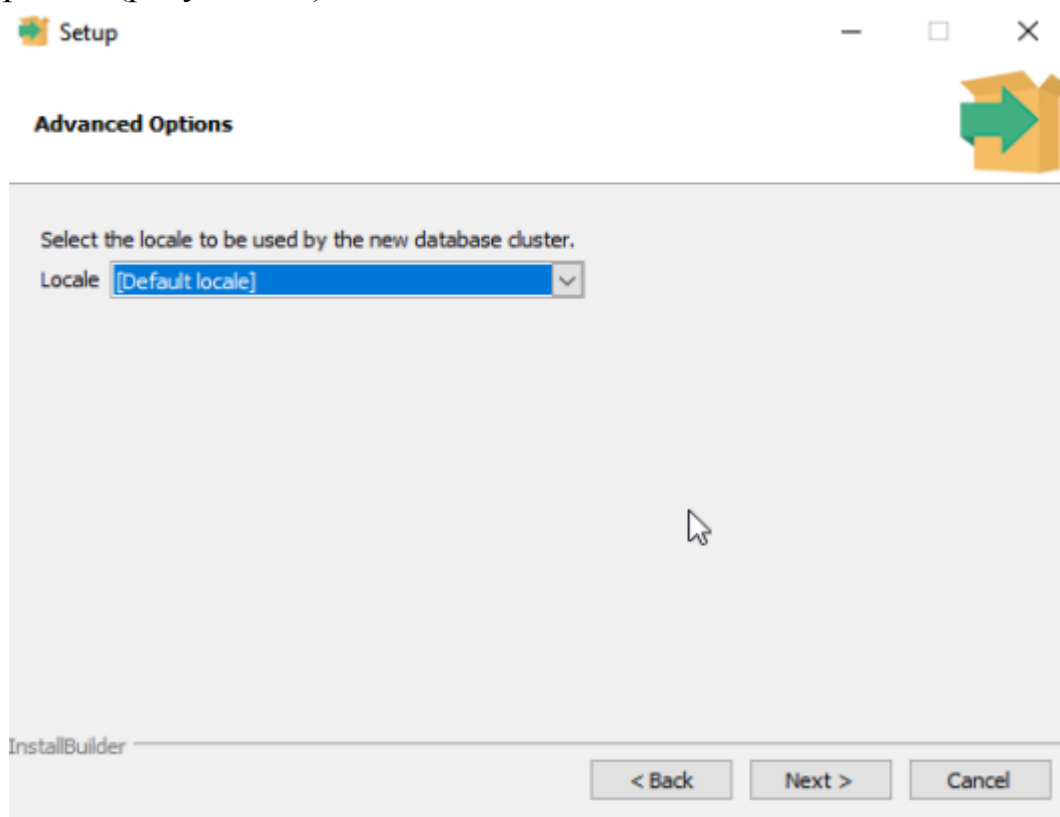


Рисунок Г.7 – Выбор языка

После нажатия «next» появится заключительное окно для проверки всех данных установки (рисунок Г.8).

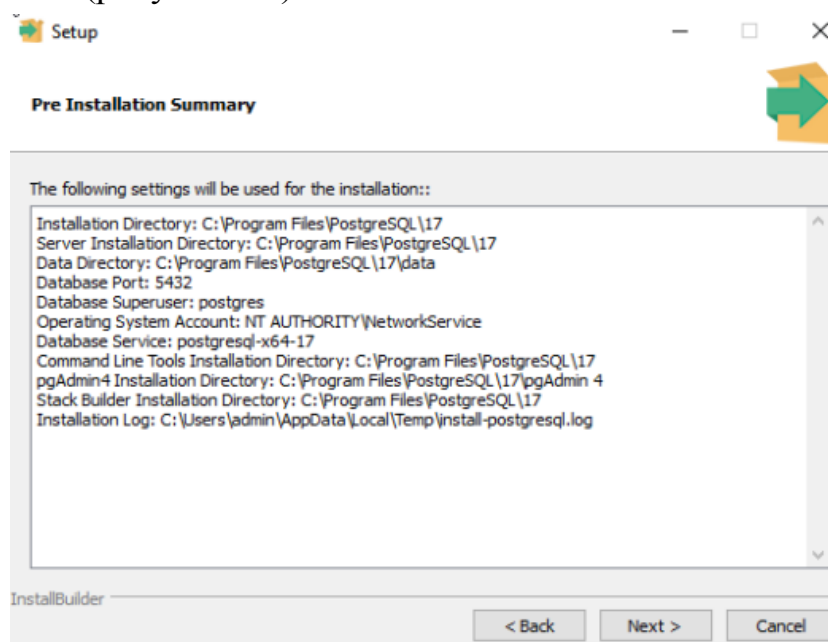


Рисунок Г.8 – Результирующее окно

После этого просто нажать «next» начнется установка postgres (рисунок Г.9).

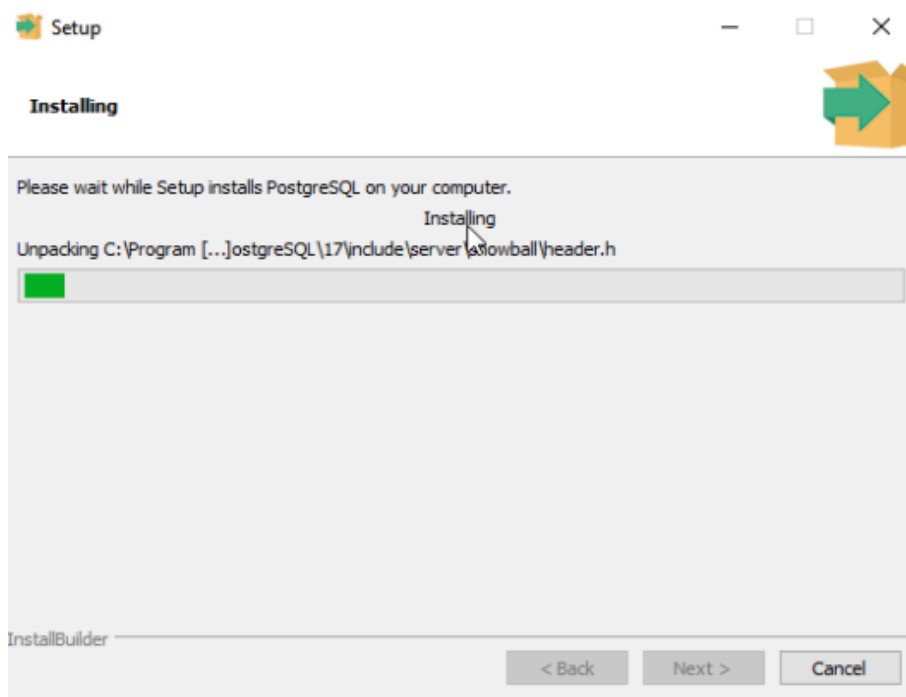


Рисунок Г.9 – Распаковка данных

После окончания установки нажать «finish». Откроется программа Stack builder, работа с которым описана в приложении В.

Далее необходимо установить pgAdmin4.

Для установки открываем установщик который скачали ранее. Нажимаем «next», принять политику и нажать «next». После этого откроется окно выбора пути установки программы (рисунок Г.10).

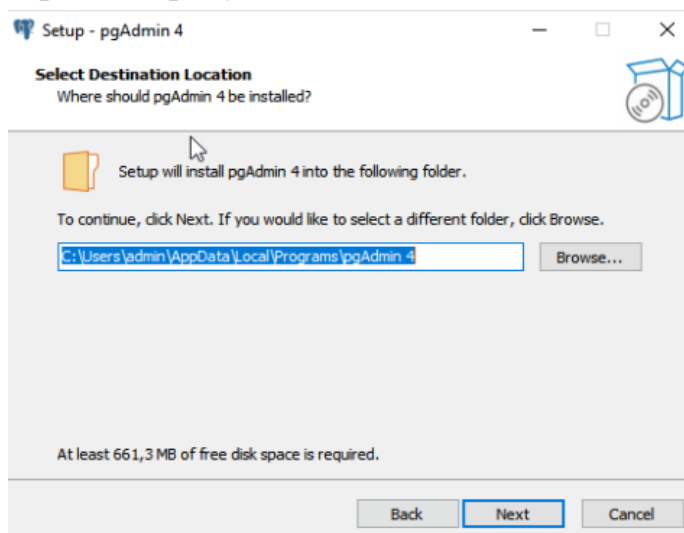


Рисунок Г.10 – Путь установки pgAdmin4

После этого нажимаем «next» – «install». Теперь ждем окончания установки программы. Нажимаем «finish».

Теперь у вас есть PostgreSQL и pgAdmin4.

Приложение Д

(справочное)

Работа в СУБД

При работе со старыми версиями postgresql стоит помнить что не будет возможности перенести данные через резервную копию на более новые версии.

После запуска программы может быть потребован пароль который был установлен ранее.

Для того чтобы начать работать необходимо открыть pgAdmin4. После этого на экране появятся списки ваших групп серверов (рисунок Д.1).



Рисунок Д.1 – Список групп серверов

Как можно наблюдать один сервер уже имеется, это базовый сервер, который создается автоматически. Необходимо раскрыть список и можно наблюдать все сервера что имеются (рисунок Д.2).

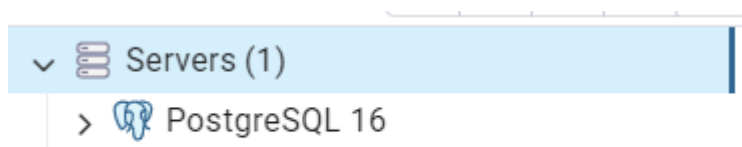


Рисунок Д.2 – Сервера

Далее работа будет происходить на базовом сервере. При раскрытии вкладки сервера, можно видеть следующие позиции «Database», «login/group Roles» и «Tablespace» (рисунок Д.3). В разделе «Database» можно наблюдать все базы данных что имеются на сервере. Чтобы создать новую базу данных необходимо сделать клик ЛКМ по соответствующей вкладке, далее ПКМ «Create - Database»(рисунок Д.4).

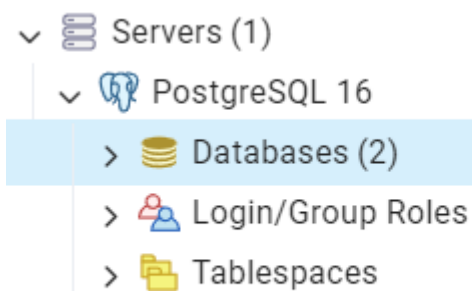


Рисунок Д.3 – Список пространств на сервере

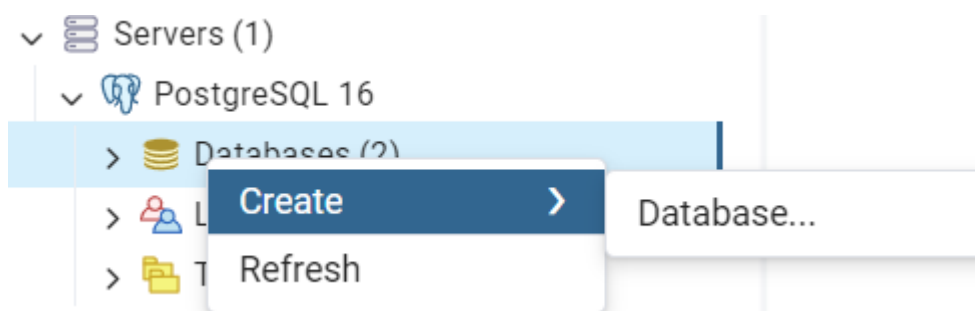


Рисунок Д.4 – Создание новой базы данных

В появившемся окне в первой строке пишем название базы данных (рисунок Д.5). Также в этой вкладке можно назначить пользователя владельцем бд, по умолчанию владельцем является суперпользователь. В следующем разделе можно поменять кодировку, которая используется в бд. В рамках лабораторных не нужно. Во вкладке «security» можно добавить пользователей которые могут пользоваться бд, по умолчанию все пользователи могут пользоваться. Также настраивается и доступ к бд, то есть возможности работы с бд. В разделе «SQL» можно видеть код создания бд на языке sql (рисунок Д.6).

The screenshot shows the 'Create - Database' dialog box with the 'General' tab selected. The 'Database' field contains the text 'Учебная'. The 'OID' field is empty. The 'Owner' field shows 'postgres' with a dropdown arrow. The 'Comment' field is a large empty text area. At the bottom, there are buttons for 'Close', 'Reset', and 'Save', along with information and help icons.

Field	Value
Database	Учебная
OID	
Owner	postgres
Comment	

Рисунок Д.5 – Название БД

The screenshot shows the 'Create - Database' dialog box with the 'SQL' tab selected. The SQL editor contains the following code:

```
1 CREATE DATABASE "Учебная"
2 WITH
3 OWNER = postgres
4 ENCODING = 'UTF8'
5 LOCALE_PROVIDER = 'libc'
6 CONNECTION LIMIT = -1
7 IS_TEMPLATE = False;
```

At the bottom, there are buttons for 'Close', 'Reset', and 'Save', along with information and help icons.

Рисунок Д.6 – SQL код создания

После нажатия «Save» бд появится в списке (рисунок Д.7). Чтобы создать таблицы в бд необходимо перейти «schemas–tables». После открытия левой кнопкой нажимаем на «Tables» , после правой на «create» (рисунок Д.8).

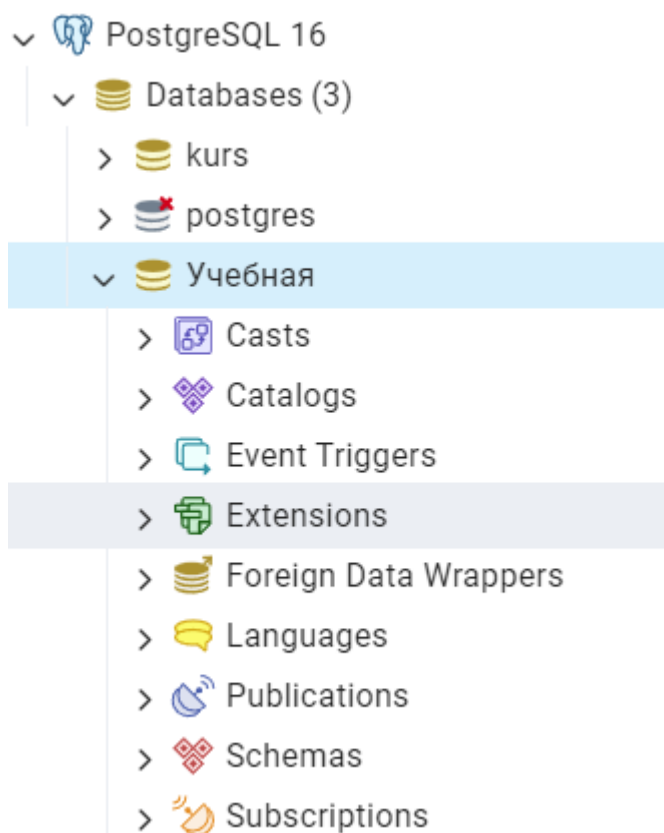


Рисунок Д.7 – Список Бд

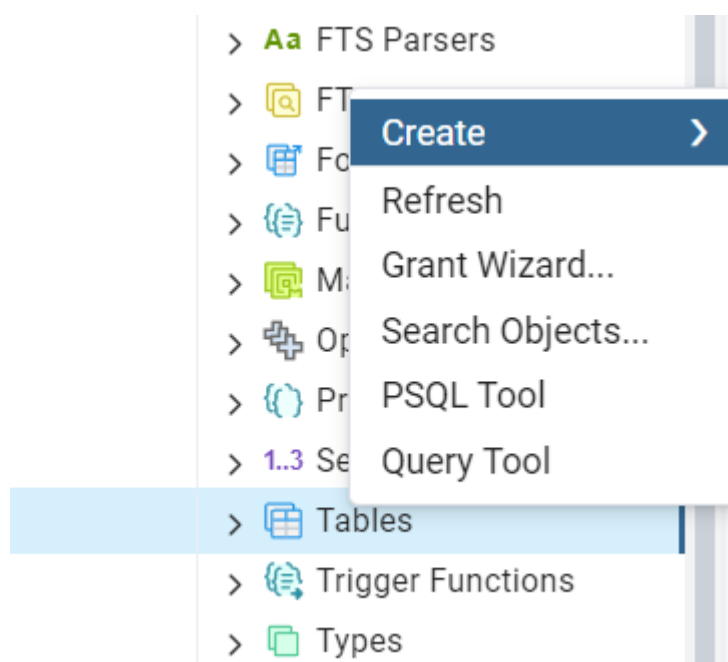


Рисунок Д.8 – Создание таблицы

В появившемся окне вписываем название таблицы (рисунок Д.9). В разделе «columns» создаем соответствующие столбцы бд (рисунок Д.10). При создании столбцов необходимо указать имя, тип данных которые будут храниться, при возможности максимальную длину. Также указываем те столбцы что обязательно должны быть заполнены, это делается переключением «Not NULL?» и указываем какой столбец является первичным ключом, переключая «Primary key?».

Рисунок Д.9 – Название таблицы

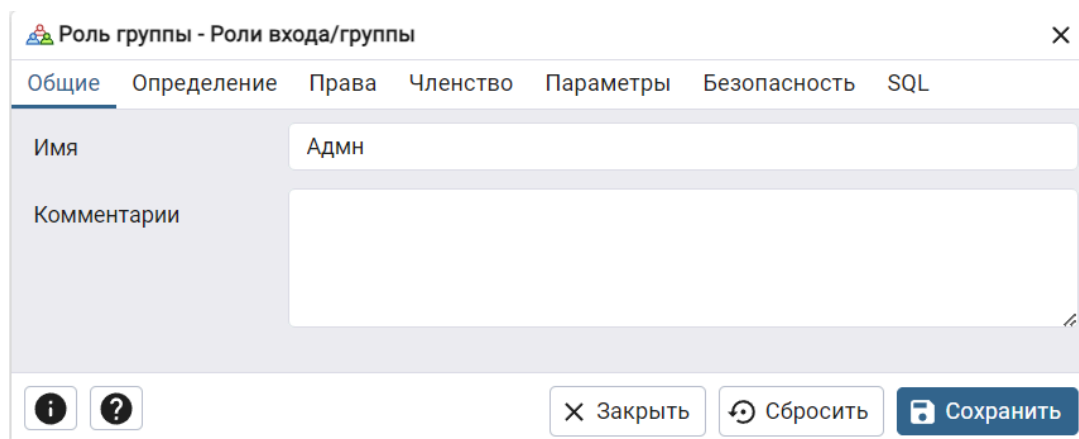
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
✱ ✎ 🗑	столбец1	text			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
✱ ✎ 🗑	столбец2	numeric	10		<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок Д.10 – Создание столбцов

В разделе «constrain» можно прописать различные ограничения. А именно, в подразделе «primary key» – ввести название первичного ключа (совет: называть ключ стоит по типу «название столбца_pk»), в «foreign key» – настроить внешний ключ (совет: название внешнего ключа лучше использовать по типу «название столбца_fk»), в «check» – соответственно прописать ограничения на столбцы, в «unique» – можно прописать ограничения

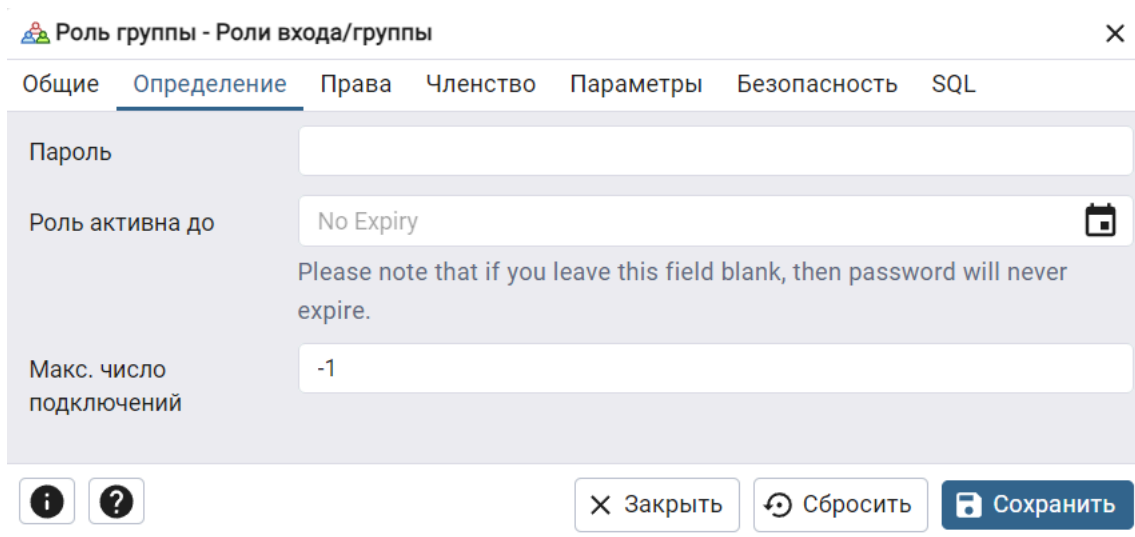
уникальности столбца, или нескольких столбцов в связке, как это отражено в основном тексте. В разделе «security» также настраивается доступ к таблице. Соответственно в «SQL» можно увидеть sql код для создания таблицы.

Для создания пользователя необходимо перейти в раздел «login/group roles». Далее следовать инструкции на рисунках Д.11 – 13.



The screenshot shows a web interface titled "Роль группы - Роли входа/группы" (Group Role - Login/group roles). The "Общие" (General) tab is selected. It contains a form with the following fields: "Имя" (Name) with the value "Адмн" (Admin), and "Комментарии" (Comments) which is an empty text area. At the bottom of the form are three buttons: "Заккрыть" (Close), "Сбросить" (Reset), and "Сохранить" (Save).

Рисунок Д.11 – Создание группы пользователей под именем «Адмн»



The screenshot shows the same web interface, but with the "Определение" (Definition) tab selected. It contains a form with the following fields: "Пароль" (Password) which is an empty field, "Роль активна до" (Role active until) with the value "No Expiry" and a calendar icon, and "Макс. число подключений" (Maximum number of connections) with the value "-1". A note below the "Role active until" field states: "Please note that if you leave this field blank, then password will never expire." At the bottom of the form are three buttons: "Заккрыть" (Close), "Сбросить" (Reset), and "Сохранить" (Save).

Рисунок Д.12 – Создание пароля для пользователя

На рисунке Д.12 показано где можно установить пароль для пользователей.

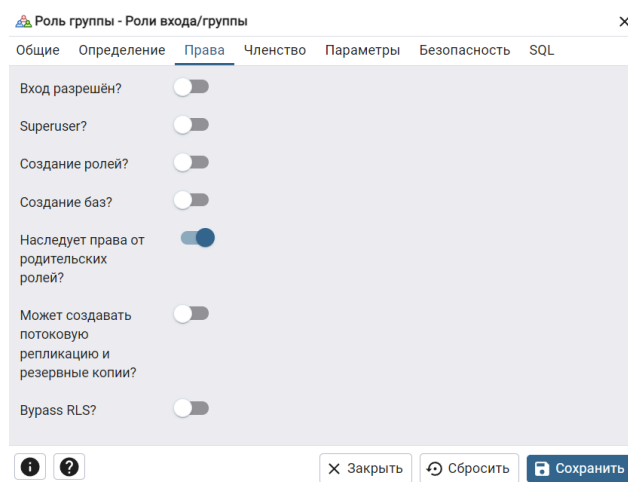


Рисунок Д.13 – Выдача прав для группы адмн

В случае если имеется код создания чего либо на языке sql, то его можно вписывать в соответствующую строку для этого можно нажать на знак (рисунок Д.14) или нажать комбинацию «alt+shift+q».

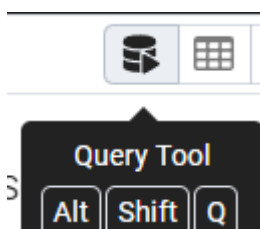


Рисунок Д.14 – Инструмент для вписания кода

После написание скрипта необходимо нажать «f5», если скрипт был на создание чего либо то необходимо перезагрузить бд, для этого нажав сначала ЛКМ по бд, надо нажать ПКМ по кнопке «refresh».