

**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

|  |  |
| --- | --- |
| **Направление подготовки:** | Информационные системы и программирование |
|  | (наименование направления) |
| **Форма обучения*:*** | очная |
|  | (очная, очно-заочная, заочная) |

**Курсовая Работа**

|  |
| --- |
| **Курсовая Работа** |
| (Вид работы) |
| Управление и автоматизация баз данных |
| (Дисциплина) |
| Разработка информационной системы учета договоров страхования |
| (Тема) |

|  |
| --- |
| **Работу выполнил(а)** |
| VДКИП-111 |
| (группа) |
| Ощепков Даниил Алексеевич |
| (ФИО) |
| **Проверил(а)** |
|  |
| (ФИО) |

**Москва 2025**

**Тема курсовой работы:**

**Разработка информационной системы учета договоров страхования**

**Структура курсовой работы**

**Введение**

* Обоснование выбора темы.
* Актуальность автоматизации учета страховых договоров.
* Цель и задачи курсовой работы.
* Объект и предмет исследования.

**Глава 1. Анализ предметной области**

* Характеристика деятельности страховой компании.
* Проблемы текущей системы учета (Excel, дублирование данных, задержки).
* Требования регуляторов (ЦБ РФ).
* Обоснование необходимости автоматизации.

**Глава 2. Проектирование системы учета договоров страхования (на основе ЛР2)**

* Назначение и цели СУДС.
* Описание архитектуры системы.
* Функциональные модули:
  + Регистрация договоров.
  + Управление филиалами.
  + Управление видами страхования.
  + Формирование отчетов.
* Нефункциональные требования:
  + Производительность, безопасность, совместимость.

**Глава 3. Реализация компонентов системы (на основе ЛР3–ЛР9)**

* Структура базы данных (таблицы: договоры, филиалы, виды).
* Основные SQL-запросы:
  + Добавление/обновление записей.
  + Расчёт страховой премии.
  + Отчёты по филиалам и видам страхования.
* Фрагменты программного кода (если реализовано).
* Макеты пользовательского интерфейса (из приложений).

**Глава 4. Тестирование и внедрение (из ЛР6–ЛР8)**

* Виды тестирования: модульное, интеграционное, приёмочное.
* Критерии успешности.
* Результаты тестирования.
* Этапы внедрения и обучения пользователей.

**Заключение**

* Результаты выполнения курсовой.
* Достижение цели и задач.
* Рекомендации по дальнейшему развитию системы.

**Приложения**

* Схема базы данных.
* SQL-запросы.
* Интерфейсы.
* Глоссарий.

**Введение**

Автоматизация процессов учета договоров страхования является важной задачей для современных страховых компаний. В условиях роста количества клиентов, требований к отчетности и необходимости снижения ошибок ручного ввода, переход к информационным системам учета становится логичным этапом развития бизнеса.

Цель данной курсовой работы — разработка информационной системы учета договоров страхования, обеспечивающей регистрацию договоров, централизованное управление справочниками филиалов и видов страхования, формирование отчетности, а также интеграцию с бухгалтерскими системами.

Задачи:

1. Провести анализ предметной области страховой компании.
2. Разработать инфологическую, логическую и физическую модели базы данных.
3. Реализовать базу данных в PostgreSQL, разработать SQL-запросы.
4. Создать хранимые процедуры, триггеры, функции.
5. Разработать диаграммы процессов и компонентов системы.
6. Обосновать архитектуру и подходы к безопасности.
7. Провести моделирование бизнес-процессов в нотациях DFD и IDEF0.

Объект исследования: бизнес-процесс учета договоров страхования.  
Предмет исследования: информационная система СУДС (система учета договоров страхования).  
Методы исследования: системный анализ, структурное и объектно-ориентированное проектирование, SQL, нотации UML, DFD, IDEF0.

**Глава 1. Анализ предметной области**

### ****1.1. Анализ предметной области****

Объектом автоматизации является страховая компания ООО «СтрахГарант», осуществляющая свою деятельность через сеть филиалов, заключающих договоры страхования с физическими и юридическими лицами. Ведение учета в настоящий момент осуществляется вручную с помощью таблиц Excel, что приводит к дублированию данных, ошибкам в расчетах страховых премий и затрудняет формирование регламентированной отчетности.

**Проблемы текущей системы:**

* отсутствие централизованной базы данных;
* ручной ввод данных, высокая вероятность ошибок;
* замедление процесса формирования отчетов;
* нарушение сроков подачи данных в контролирующие органы (ЦБ РФ).

**Цель автоматизации** — создать информационную систему учета договоров страхования (СУДС), позволяющую:

* централизованно вести базу договоров, филиалов и видов страхования;
* автоматизировать расчёт страховой премии;
* формировать сводные отчеты и экспортировать данные в бухгалтерские системы (1С, SAP);
* обеспечить безопасность и разграничение доступа к данным.

### ****1.2. Инфологическое проектирование****

В системе выделяются следующие сущности:

| **Сущность** | **Атрибуты** |
| --- | --- |
| **Филиал** | Код\_филиала (PK), Наименование\_филиала, Адрес, Телефон |
| **Вид страхования** | Код\_вида (PK), Наименование |
| **Договор** | Номер\_договора (PK), Дата\_заключения, Страховая\_сумма, Тарифная\_ставка, Код\_филиала (FK), Код\_вида (FK) |

#### ****Описание атрибутов сущностей****

**Филиал:**

* Код\_филиала – уникальный идентификатор филиала.
* Наименование\_филиала – наименование подразделения.
* Адрес – местоположение филиала.
* Телефон – контактный номер.

**Вид страхования:**

* Код\_вида – уникальный код вида.
* Наименование – наименование страхового продукта (например, «КАСКО», «ДМС»).

**Договор:**

* Номер\_договора – уникальный номер договора.
* Дата\_заключения – дата оформления.
* Страховая\_сумма – сумма покрытия.
* Тарифная\_ставка – процент по договору.
* Код\_филиала – ссылка на филиал.
* Код\_вида – ссылка на вид страхования.

#### ****Ключи и связи****

* Филиал (1) ← Договор (много) — связь "один ко многим"
* Вид страхования (1) ← Договор (много) — связь "один ко многим"

Модель реализуется в СУБД PostgreSQL, связи обеспечиваются внешними ключами.



### ****1.3. Построение логической модели базы данных****

На этапе логического проектирования осуществляется переход от инфологической модели к реляционной, представленной в виде нормализованных таблиц. База данных соответствует третьей нормальной форме (3НФ), что обеспечивает:

* устранение избыточности,
* логическую целостность данных,
* удобство масштабирования.

**Реляционные таблицы:**

CREATE TABLE Филиал (

Код\_филиала SERIAL PRIMARY KEY,

Наименование\_филиала VARCHAR(100) NOT NULL,

Адрес VARCHAR(255) NOT NULL,

Телефон VARCHAR(20) NOT NULL

);

CREATE TABLE Вид\_страхования (

Код\_вида SERIAL PRIMARY KEY,

Наименование VARCHAR(100) NOT NULL UNIQUE

);

CREATE TABLE Договоры (

Номер\_договора SERIAL PRIMARY KEY,

Дата\_заключения DATE NOT NULL,

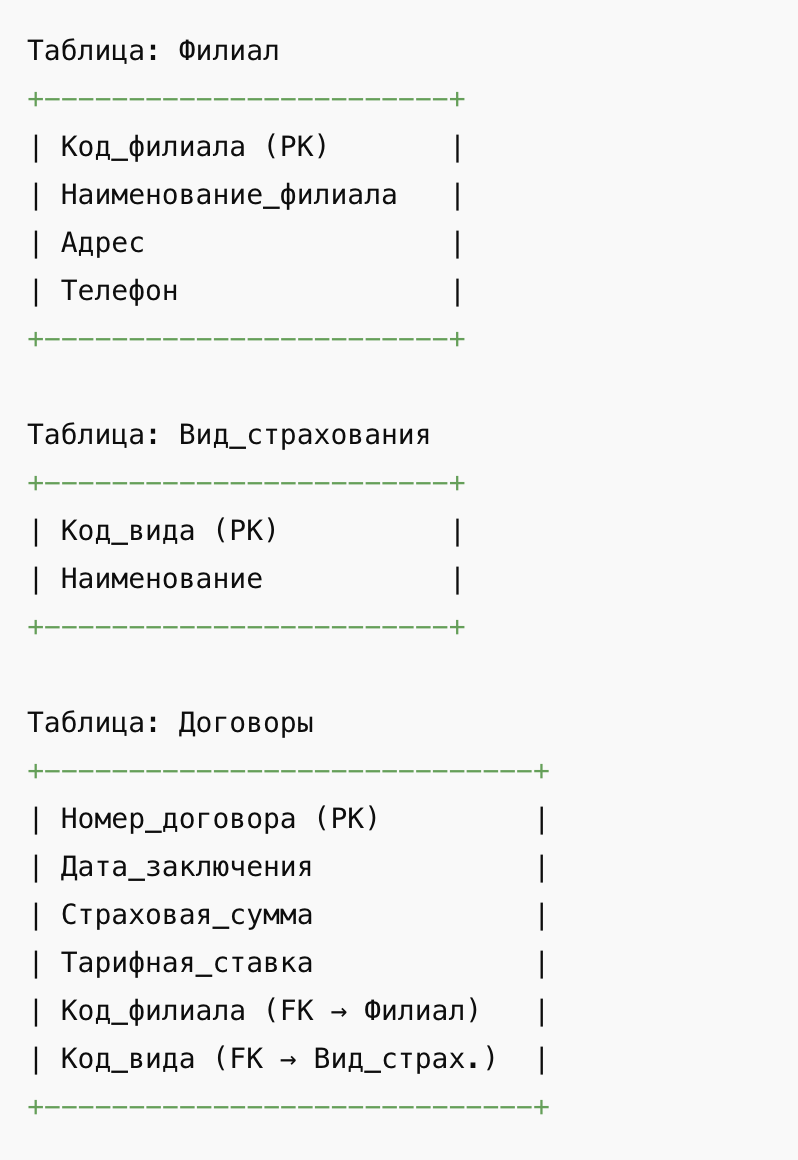
Страховая\_сумма DECIMAL(15, 2) NOT NULL CHECK (Страховая\_сумма >= 0),

Тарифная\_ставка DECIMAL(5, 2) NOT NULL CHECK (Тарифная\_ставка BETWEEN 0 AND 100),

Код\_филиала INT NOT NULL REFERENCES Филиал(Код\_филиала),

Код\_вида INT NOT NULL REFERENCES Вид\_страхования(Код\_вида)

);



**Описание структуры таблиц:**

| **Таблица** | **Поле** | **Тип данных** | **Ограничения** |
| --- | --- | --- | --- |
| Филиал | Код\_филиала | SERIAL | Первичный ключ |
|  | Наименование\_филиала | VARCHAR(100) | NOT NULL |
|  | Адрес | VARCHAR(255) | NOT NULL |
|  | Телефон | VARCHAR(20) | NOT NULL |
| Вид страхования | Код\_вида | SERIAL | Первичный ключ |
|  | Наименование | VARCHAR(100) | NOT NULL, UNIQUE |
| Договоры | Номер\_договора | SERIAL | Первичный ключ |
|  | Дата\_заключения | DATE | NOT NULL |
|  | Страховая\_сумма | DECIMAL(15,2) | NOT NULL, >= 0 |
|  | Тарифная\_ставка | DECIMAL(5,2) | NOT NULL, от 0 до 100 |
|  | Код\_филиала | INT | NOT NULL, внешний ключ на Филиал |
|  | Код\_вида | INT | NOT NULL, внешний ключ на Вид страхования |

### ****1.4. Построение физической модели базы данных****

На данном этапе описываются параметры хранения данных и создаются технические объекты в выбранной СУБД PostgreSQL. Основной упор делается на:

* обеспечение целостности данных,
* автоматическую генерацию ключей (SERIAL),
* ограничение по значениям (например, CHECK для суммы и ставки),
* индексацию по внешним ключам (в PostgreSQL это создаётся автоматически при создании внешних связей).

**Примечания:**

* Таблицы оптимизированы по производительности.
* Поля Тарифная\_ставка и Страховая\_сумма рассчитаны для высокоточной финансовой обработки.
* Все внешние связи реализованы через REFERENCES.

**Глава 2. Проектирование системы учета договоров страхования**

### ****2.1. Общие сведения о системе****

Информационная система учета договоров страхования (СУДС) разрабатывается по заказу страховой компании ООО «СтрахГарант» в соответствии с приказом генерального директора № 45-Т от 15.03.2025. Разработку выполняет компания ООО «IT-Стратегия». Срок выполнения — 6 месяцев с момента утверждения технического задания.

**Наименование системы:**  
Система учета договоров страхования (СУДС)

**Назначение системы:**  
Автоматизация процессов регистрации и учета договоров, централизованное управление справочниками филиалов и видов страхования, формирование отчетности, интеграция с бухгалтерскими системами и обеспечение информационной безопасности.

### ****2.2. Цели и задачи создания системы****

**Цель разработки:**  
Создание надежного цифрового инструмента для ведения учета страховых договоров, обеспечивающего достоверность, оперативность и безопасность данных.

**Основные задачи:**

* Исключение дублирования и несогласованности данных.
* Снижение влияния человеческого фактора при расчетах.
* Повышение скорости и точности формирования отчетов.
* Интеграция с системами 1С и SAP.
* Обеспечение ролифицированного доступа к функционалу.

### ****2.3. Характеристика объекта автоматизации****

До внедрения системы:

* Учет договоров осуществляется вручную в Excel.
* Частые ошибки в расчетах премий.
* Задержки в отчетности.
* Нет централизованного хранилища данных.

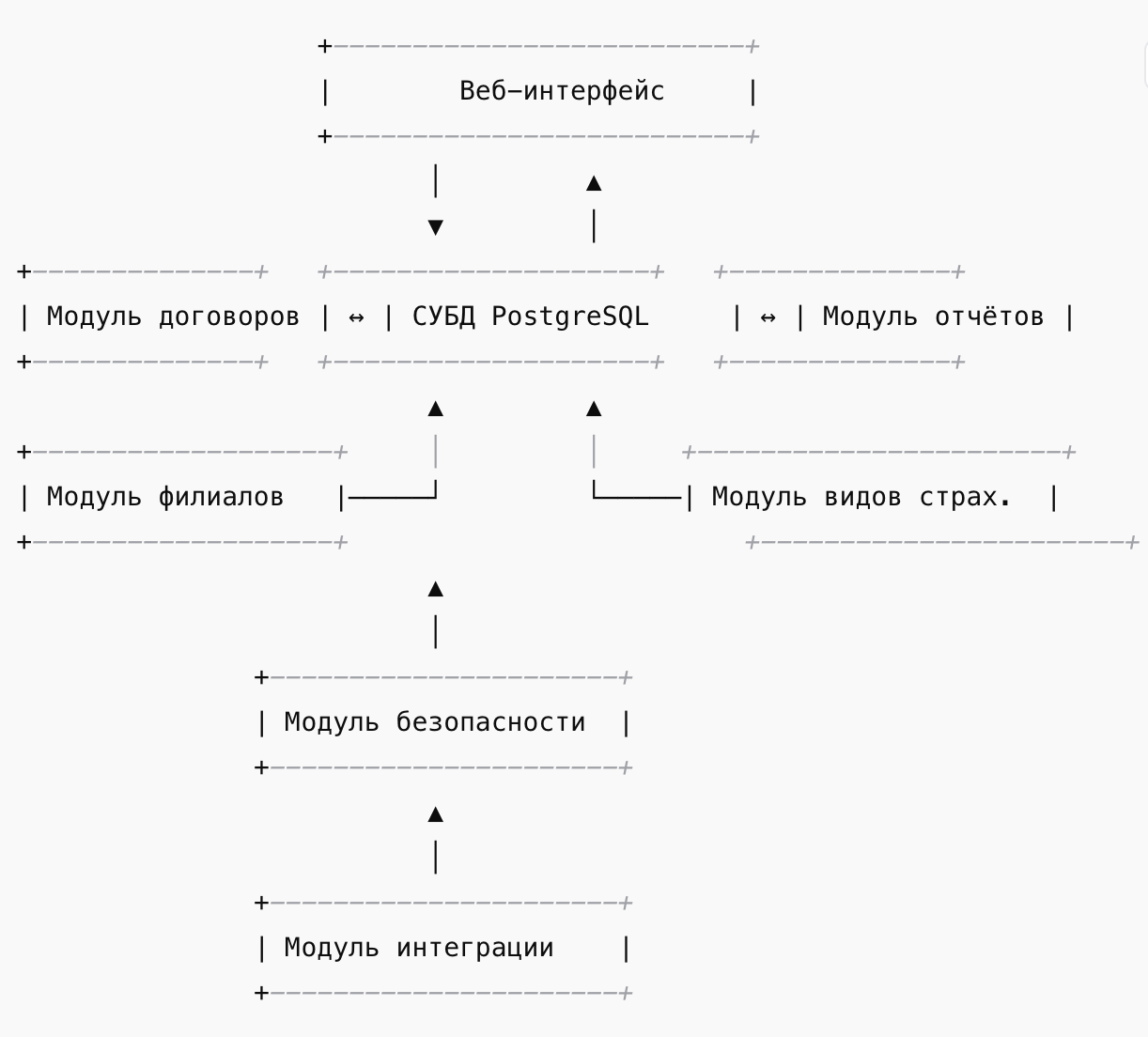
**Обоснование необходимости автоматизации:**

* Более 5000 новых договоров ежегодно.
* Ужесточение требований к отчетности со стороны регуляторов (ЦБ РФ).
* Рост числа филиалов компании.

### ****2.4. Структура и модули системы****

Система будет состоять из следующих модулей:

| **Модуль** | **Описание** |
| --- | --- |
| **Модуль договоров** | Регистрация и хранение договоров. Расчет страховой премии. |
| **Модуль филиалов** | Управление справочником филиалов (добавление, редактирование, удаление). |
| **Модуль видов страхования** | Работа со справочником страховых продуктов. |
| **Модуль отчетности** | Формирование отчетов: сводный по филиалам, топ-5 популярных видов. |
| **Модуль безопасности** | Аутентификация пользователей, разграничение прав доступа. |
| **Модуль интеграции** | Экспорт данных в бухгалтерские системы (1С, SAP). |



### ****2.5. Функциональные и нефункциональные требования****

#### ****Функциональные требования:****

* Регистрация договоров с параметрами: дата, сумма, тариф, филиал, вид.
* Расчет страховой премии:  
  Премия = Страховая сумма × Тарифная ставка
* Ведение справочников филиалов и видов страхования.
* Формирование двух типов отчетов:
  + Сводный отчет по филиалам.
  + Топ-5 популярных видов страхования.
* Авторизация пользователей и контроль доступа по ролям.

#### ****Нефункциональные требования:****

* **Производительность:**
  + Обработка более 100 параллельных запросов.
  + Время отклика системы не более 2 секунд.
* **Безопасность:**
  + Реализация ролевой модели доступа: администратор, менеджер, аудитор.
  + Шифрование персональных данных в соответствии с ГОСТ Р 34.10–2012.
* **Совместимость:**
  + Операционные системы: Windows 10+, Linux.
  + СУБД: PostgreSQL 12+, MySQL 8.0+.

### ****2.6. Этапы разработки и внедрения****

| **Этап** | **Содержание** | **Срок** |
| --- | --- | --- |
| 1. Проектирование | Архитектура, схемы, модели | 1 месяц |
| 2. Разработка интерфейса | Формы для работы с договорами и справочниками | 2 месяца |
| 3. Реализация и интеграция БД | Построение таблиц, связи, загрузка | 1.5 месяца |
| 4. Тестирование | Модульное, интеграционное, приемочное | 1 месяц |
| 5. Внедрение | Развертывание, обучение персонала | 0.5 месяца |

### ****2.7. Контроль качества и приемка****

**Формы контроля:**

* Модульное тестирование с покрытием кода не менее 90%.
* Интеграционные тесты — на стыке модулей.
* Приемочные испытания — при участии заказчика.

**Критерии приемки:**

* Соответствие всем требованиям ТЗ.
* Отсутствие критических ошибок (Severity 1).
* Успешное прохождение тестирования.

**Глава 3. Реализация компонентов системы**

### ****.1. Диаграмма вариантов использования (Use Case)****

В системе участвуют три ключевые роли (актора):

* **Администратор** — отвечает за безопасность и управление пользователями.
* **Менеджер** — регистрирует и редактирует договоры, управляет справочниками.
* **Аудитор** — генерирует аналитические отчёты.

#### Основные варианты использования:

| **Актор** | **Варианты использования** |
| --- | --- |
| **Администратор** | - Управление пользователями - Настройка безопасности - Управление справочниками |
| **Менеджер** | - Регистрация договора - Расчёт страховой премии - Просмотр/редактирование договоров - Управление филиалами и видами страхования |
| **Аудитор** | - Генерация сводного отчета - Формирование топ-5 популярных видов страхования |
| **Общие действия** | - Авторизация в системе (для всех акторов) |

Регистрация договора включает автоматический расчет страховой премии:  
**Страховая премия = Страховая сумма × Тарифная ставка**

****

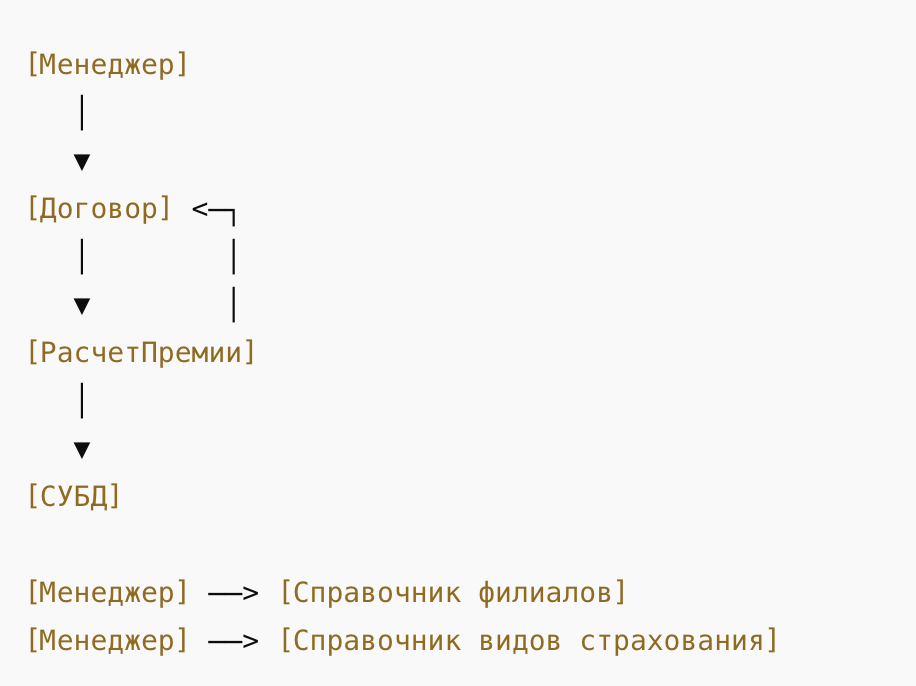
### ****3.2. Диаграмма кооперации и диаграмма последовательности****

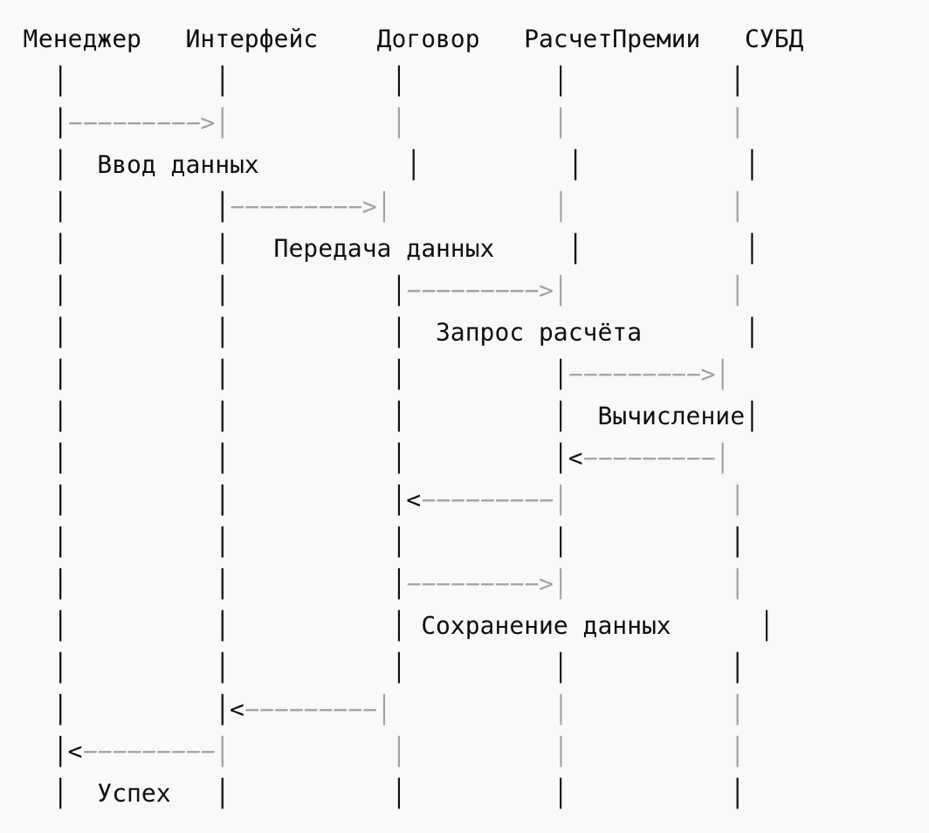
#### ****Кооперация компонентов:****

* **Менеджер** взаимодействует с:
  + объектом **Договор** — для регистрации, редактирования;
  + **СправочникФилиалов** и **СправочникВидовСтрахования** — при выборе параметров договора;
  + модулем **РасчетПремии** — для вычислений;
  + **СУБД** — для сохранения данных.
* **Администратор** управляет объектами **Пользователь** и **Безопасность**.
* **Аудитор** вызывает модуль **Отчет**, который обращается к БД.

#### ****Сценарий "Регистрация договора" (последовательность):****

1. Менеджер открывает форму.
2. Вводит данные договора (сумма, ставка, филиал, вид).
3. Интерфейс передаёт данные в объект **Договор**.
4. **Договор** запрашивает расчёт у **РасчетПремии**.
5. Полученный результат сохраняется в **СУБД**.
6. Интерфейс уведомляет менеджера об успехе.





### ****3.3. Диаграмма классов****

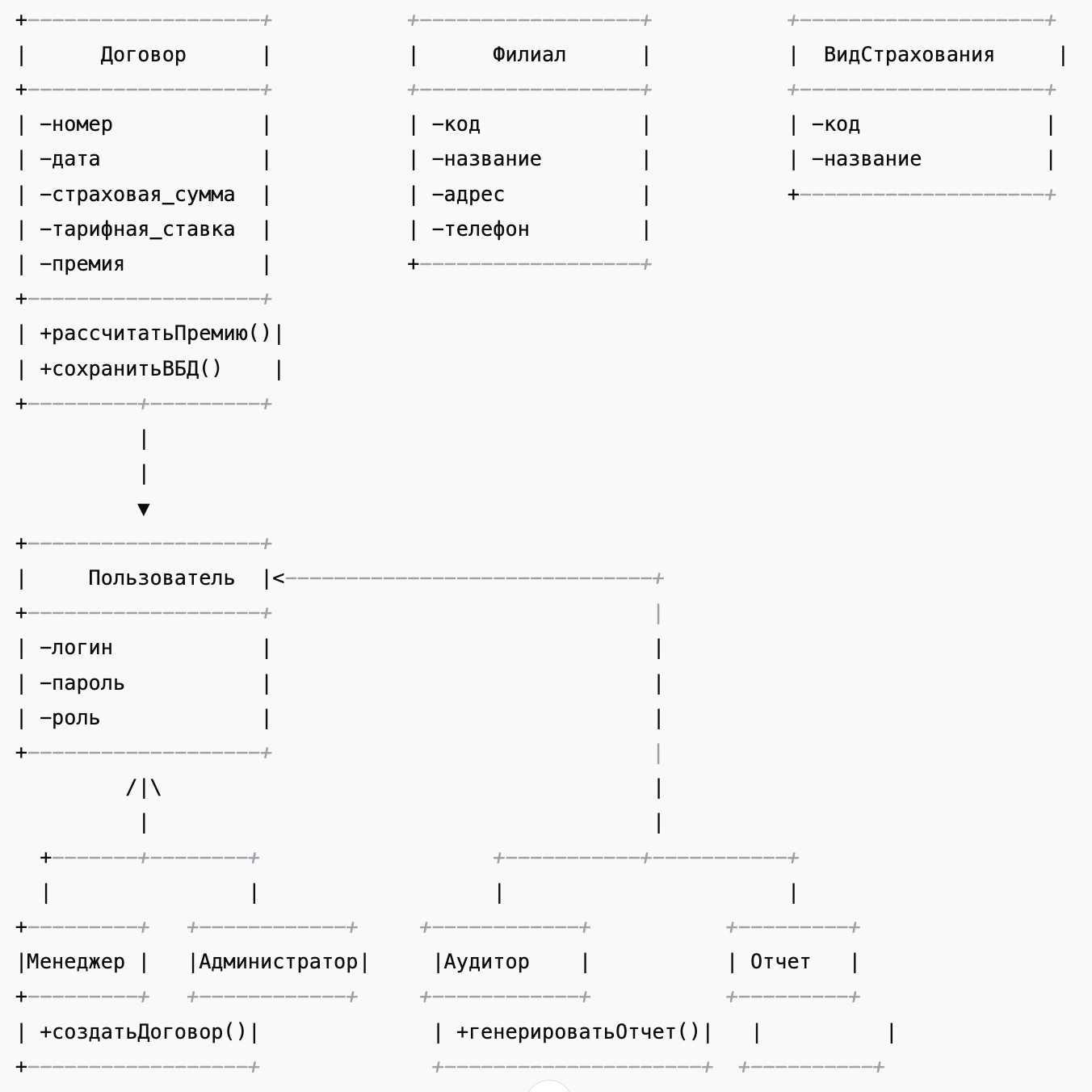
На этапе проектирования классов выделяются следующие сущности:

#### ****Классы:****

| **Класс** | **Атрибуты / Методы** |
| --- | --- |
| **Договор** | Номер, Дата, Сумма, Тариф, Премия рассчитатьПремию(), сохранитьВБД() |
| **Филиал** | Название, Адрес, Телефон |
| **ВидСтрахования** | Название, Код |
| **Пользователь** (абстрактный) | Логин, Пароль, Роль |
| **Менеджер** | создатьДоговор(), редактироватьДоговор() |
| **Администратор** | добавитьПользователя(), настроитьБезопасность() |
| **Аудитор** | сгенерироватьОтчет() |
| **Отчет** | Подклассы: СводныйОтчет, ТопВидовСтрахования Методы: загрузитьДанныеИзБД(), экспортВPDF() |
| **РасчетПремии** | вычислить(сумма, тариф) |
| **СУБД** | сохранитьДанные(), получитьДанные() |

#### ****Связи между классами:****

* Договор агрегирует Филиал и ВидСтрахования.
* Менеджер создает Договор.
* Аудитор вызывает Отчет.
* Все взаимодействия с данными идут через СУБД.



### ****3.4. Диаграммы деятельности и состояний****

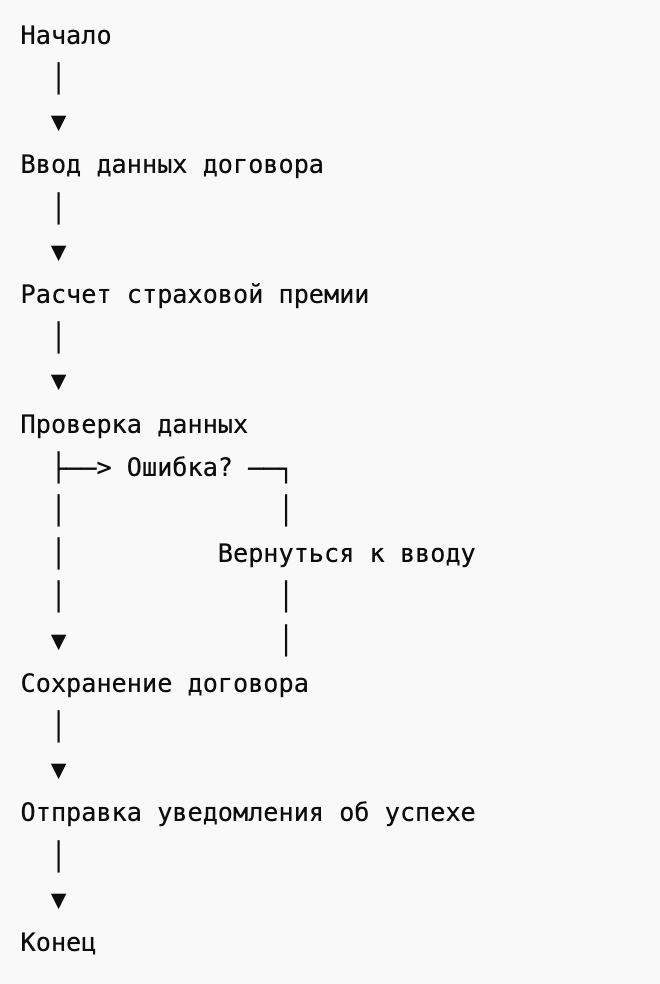
#### ****Диаграмма деятельности "Регистрация договора"****

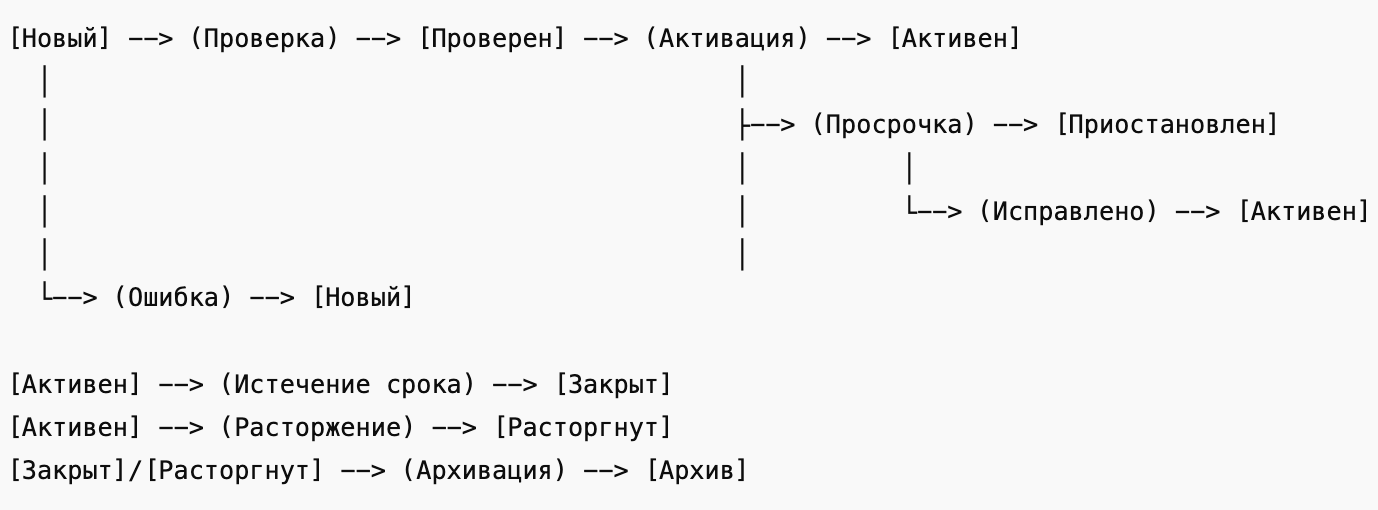
1. Менеджер открывает форму.
2. Вводит данные.
3. Система рассчитывает премию:  
   Премия = Сумма × Тариф
4. Проверка данных:
   * если ошибки — возврат;
   * если всё верно — сохранение в БД.
5. Уведомление об успешной регистрации.

#### ****Состояния договора:****

| **Состояние** | **Пояснение** |
| --- | --- |
| Новый | Только создан |
| Проверен | Данные верифицированы |
| Активен | Вступил в силу |
| Приостановлен | Временная блокировка |
| Закрыт | Завершён по сроку |
| Расторгнут | Прекращён досрочно |
| Архив | Перемещён в архив |

Переходы происходят на основе действий менеджера или условий (например, просрочки платежа).

****

****

**Глава 4. Администрирование базы данных**

### ****4.1. Создание пользовательских ролей и управление доступом****

Для обеспечения безопасности и разграничения прав доступа в СУБД PostgreSQL создаются роли с определенными правами.

**Основные роли:**

* **Администратор** — полный доступ к базе данных, управление пользователями, резервное копирование.
* **Менеджер** — права на добавление, редактирование и просмотр данных в таблицах договоров, филиалов, видов страхования.
* **Аудитор** — права только на чтение данных и формирование отчетов.

**Пример создания ролей и назначения прав:**

**-- Создание роли администратора**

**CREATE ROLE admin LOGIN PASSWORD 'admin\_password' SUPERUSER;**

**-- Создание роли менеджера**

**CREATE ROLE manager LOGIN PASSWORD 'manager\_password';**

**GRANT SELECT, INSERT, UPDATE ON ALL TABLES IN SCHEMA public TO manager;**

**-- Создание роли аудитора**

**CREATE ROLE auditor LOGIN PASSWORD 'auditor\_password';**

**GRANT SELECT ON ALL TABLES IN SCHEMA public TO auditor;**

### ****4.2. Настройка авторизации и безопасности****

* Использование SSL-соединений для защиты данных при передаче.
* Шифрование персональных данных согласно ГОСТ Р 34.10-2012.
* Аудит действий пользователей — ведение логов операций с базой.

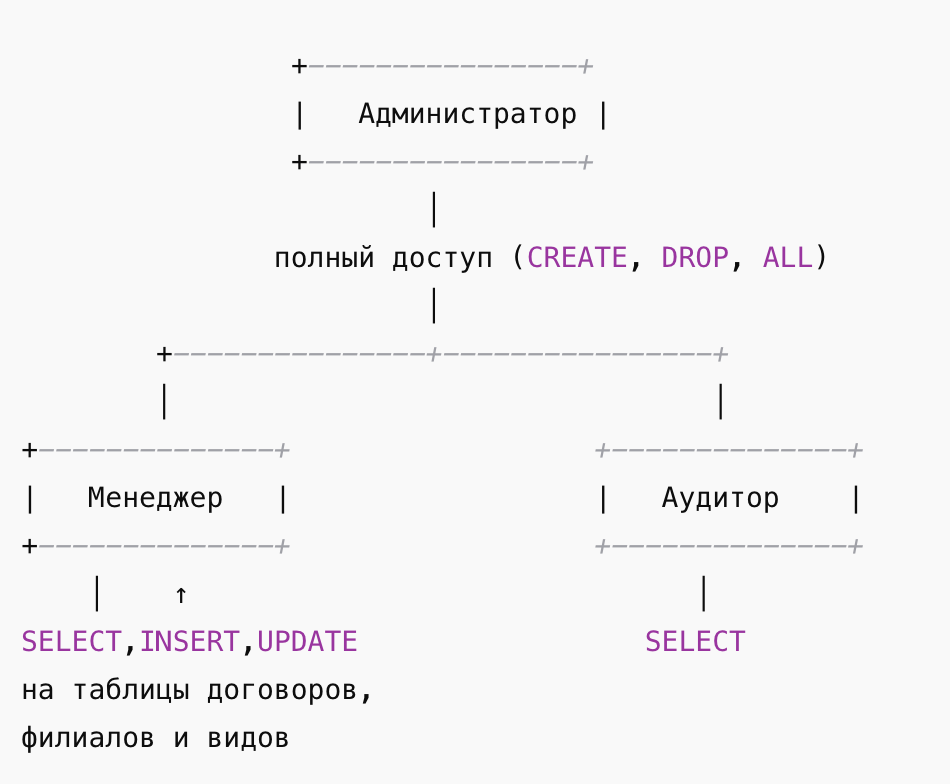
### ****4.3. Модель восстановления и стратегия резервного копирования****

* Регулярное создание резервных копий (bkp) всей базы.
* Использование инструментов PostgreSQL: pg\_dump, pg\_restore.
* Стратегия бэкапов:
  + Ежедневный полный бэкап.
  + Еженедельный инкрементальный бэкап.
* Тестирование восстановления — ключевая часть процедуры.

### ****4.4. Настройка параметров безопасности****

* Ограничение доступа по IP.
* Использование сильных паролей.
* Регулярное обновление СУБД для устранения уязвимостей.

### ****4.5. Схемы и диаграммы****

****

Стратегия резервного копирования

****

**Глава 5. Заключение**

В ходе выполнения курсовой работы была разработана информационная система учета договоров страхования (СУДС), отвечающая современным требованиям автоматизации страховой деятельности.

Реализованы основные модули:

* централизованное хранение данных по договорам, филиалам и видам страхования;
* автоматический расчет страховой премии;
* формирование сводных отчетов и топ-5 популярных видов страхования;
* разграничение доступа на уровне ролей пользователей;
* обеспечение безопасности данных и резервного копирования.

### Основные выводы:

* Автоматизация учетных процессов позволяет существенно снизить риск ошибок, ускорить обработку данных и повысить качество отчетности.
* Использование реляционной базы данных PostgreSQL обеспечивает надежность и масштабируемость системы.
* Внедрение системы облегчит интеграцию с бухгалтерскими системами и повысит прозрачность бизнес-процессов.

### Рекомендации на перспективу:

* Реализовать веб-интерфейс для удобной работы менеджеров и аудиторов.
* Добавить модуль мониторинга и аналитики на базе машинного обучения.
* Автоматизировать процессы резервного копирования и восстановления.
* Перенести часть бизнес-логики из серверных процедур в клиентское приложение для повышения отзывчивости.

## **Список использованной литературы**

1. Федеральный закон от 27.07.2006 №149-ФЗ «Об информации, информационных технологиях и о защите информации». URL: <https://docs.cntd.ru/document/9005388>
2. Стружкин, Н. П. Базы данных: проектирование : учебник для СПО / Н. П. Стружкин, В. В. Годин. — Москва : Юрайт, 2023.
3. Стасышин, В. М. Базы данных: технологии доступа : учебное пособие для СПО / В. М. Стасышин, Т. Л. Стасышина. — Москва : Юрайт, 2023.
4. ГОСТ Р 34.10-2012 — Криптографическая защита информации.
5. Документация PostgreSQL: <https://www.postgresql.org/docs/>
6. Практические руководства по SQL и проектированию БД.
7.  Федеральный закон от 27.07.2006 №149-ФЗ «Об информации, информационных технологиях и о защите информации». URL: <https://docs.cntd.ru/document/9005388>
8.  Стружкин, Н. П., Годин, В. В. Базы данных: проектирование : учебник для СПО. — Москва : Юрайт, 2023. — 477 с.
9.  Стасышин, В. М., Стасышина, Т. Л. Базы данных: технологии доступа : учебное пособие для СПО. — Москва : Юрайт, 2023. — 164 с.
10.  ГОСТ Р 34.10-2012. Криптографическая защита информации.
11.  Китов, В. А., Музычкин, П. А., Неделькин, А. А. От кибернетики и АСУ до цифровой экономики. — Москва : РЭУ им. Г. В. Плеханова, 2020. — 64 с.
12.  Документация PostgreSQL. URL: <https://www.postgresql.org/docs/>
13.  Барышников, А. А. Проектирование баз данных. — Москва : Инфра-М, 2021. — 320 с.
14.  Кирьянов, В. М. Современные системы управления базами данных. — СПб : Питер, 2022. — 415 с.
15.  Парамонов, А. В., Коннов, И. А. Глоссарий официальных дефиниций в сфере ИТ и защиты информации. — Н. Новгород: Изд-во «Дятловы горы», 2021. — 232 с.
16.  Разработка и проектирование баз данных — этапы и методики. URL: <https://www.artwell.ru/services/razrabotka-baz-dannykh/>
17.  Метанит. Руководство по PostgreSQL. URL: <https://metanit.com/sql/postgresql/>
18.  Проблемы передачи информации / гл. ред. Бассалыго Л. А. — Москва: Академкнига, 2021. — 450 с.

## **Приложения**

* Текст Технического задания (ЛР2)
* Код SQL создания таблиц и индексов
* Примеры SQL-запросов (выборки, группировки, обновления)
* Код хранимых процедур, триггеров и функций
* Диаграммы и схемы (в текстовом виде, как обсуждали)
* Макеты интерфейсов (если есть)