

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Проектирование и разработка информационных систем

Практическая работа №2

Этапы проектирования и разработки ИС при структурном подходе к  
программированию. Стадия "Эскизный проект"

Выполнили:  
Студенты группы 051006

Шуляк Андрей  
Дранкевич Артём

Проверил:

Грибович А. А

Минск 2024

## CONTENTS

1	АНАЛИЗ ФУНКЦИОНАЛЬНЫХ И ЭКСПЛУАТАЦИОННЫХ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ .....	3
2	ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ, СТРУКТУРА ПРОГРАММНОГО ПРОДУКТА, СОСТАВ ФУНКЦИЙ, РЕЖИМЫ ФУНКЦИОНИРОВАНИЯ .....	4
3	ДИАГРАММЫ ПОТОКОВ ДАННЫХ .....	6
4	ОБЪЕКТЫ И ИХ АТТРИБУТЫ.....	10
5	СЛОВАРЬ ТЕРМИНОВ .....	13

# **1 АНАЛИЗ ФУНКЦИОНАЛЬНЫХ И ЭКСПЛУАТАЦИОННЫХ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ**

При анализе требований к функционированию и использованию информационной системы выявлено, что большинство ключевых операций, связанных с управлением клиентами, счетами, договорами и т. д., соответствуют принципам CRUD (Create, Read, Update, Delete). Суть в том, что система должна обеспечивать возможность создания новых записей (например, новых клиентов или договоров), чтения информации о существующих записях (таких как данные о счетах), обновления существующих записей (например, изменение персональных данных клиента) и удаления записей (например, закрытие счета).

Этот анализ позволяет определить основные требования к функциональности системы, включая необходимость разработки соответствующих интерфейсов для выполнения каждого из указанных CRUD-действий. Кроме того, требования к использованию подчеркивают важность обеспечения стабильной работы системы на различных платформах и устройствах, а также необходимость обеспечения безопасности передачи и хранения финансовых данных клиентов.

## **2 ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ, СТРУКТУРА ПРОГРАММНОГО ПРОДУКТА, СОСТАВ ФУНКЦИЙ, РЕЖИМЫ ФУНКЦИОНИРОВАНИЯ**

После анализа требований к функциональности и использованию программного продукта был выбран язык программирования Python, с использованием базы данных MariaDB. Этот выбор обоснован анализом требований информационной системы.

Python представляет собой гибкий и мощный инструмент для разработки приложений с разнообразной функциональностью. Он обладает превосходной производительностью и гибкостью, что позволяет эффективно реализовывать CRUD-интерфейсы и другие функциональные требования, необходимые для информационной системы.

MariaDB предоставляет надежную и эффективную базу данных, которая обеспечивает безопасность и управление данными. Использование MariaDB позволяет создавать масштабируемые и надежные приложения, что особенно важно для финансовых систем.

Выбор MariaDB в качестве системы управления базами данных обоснован ее надежностью, масштабируемостью и возможностями обработки транзакций. MariaDB представляет собой открытую и бесплатную СУБД с обширным набором функций, поддерживающую ACID (атомарность, согласованность, изолированность, долговечность) для обеспечения целостности данных в банковской среде.

Таким образом, использование Python, также клиент-серверной архитектуры совместно с MariaDB позволит эффективно реализовать требования функциональности и использования банковской системы, обеспечивая высокую производительность, надежность и безопасность.

Выбор функций CRUD (Create, Read, Update, Delete) обоснован необходимостью основных операций, которые должна предоставлять информационная система.

Клиент-серверная архитектура обеспечивает удобное взаимодействие между клиентскими и серверными компонентами приложения, что способствует эффективной работе системы и обработке данных.

Для клиент-серверной архитектуры характерны следующие особенности функционирования:

- Распределенная обработка данных: Клиент-серверная архитектура позволяет разделить обязанности между клиентской и серверной частями приложения. Это обеспечивает более эффективное управление и обработку данных в банковской системе, позволяя серверу выполнять вычисления и обработку, а клиенту — обеспечивать интерфейс и взаимодействие с пользователем;
- Централизованное хранение данных: Серверная часть приложения осуществляет централизованное хранение данных, что обеспечивает единый и надежный источник информации для всех клиентов. Это важно для банковской системы, так как обеспечивает консистентность данных и уменьшает риск их потери или повреждения;
- Управление доступом и безопасностью: Клиент-серверная архитектура обеспечивает более гибкий и эффективный контроль доступа к данным и ресурсам. Сервер может централизованно управлять правами доступа, обеспечивая высокий уровень безопасности для конфиденциальных финансовых данных в банковской системе;
- Сетевая гибкость и масштабируемость: Клиент-серверная архитектура позволяет легко масштабировать систему и обеспечивать доступ к данным через сеть. Это обеспечивает гибкость в управлении ресурсами и позволяет системе эффективно адаптироваться к изменяющимся потребностям банковского бизнеса.

### 3 ДИАГРАММЫ ПОТОКОВ ДАННЫХ

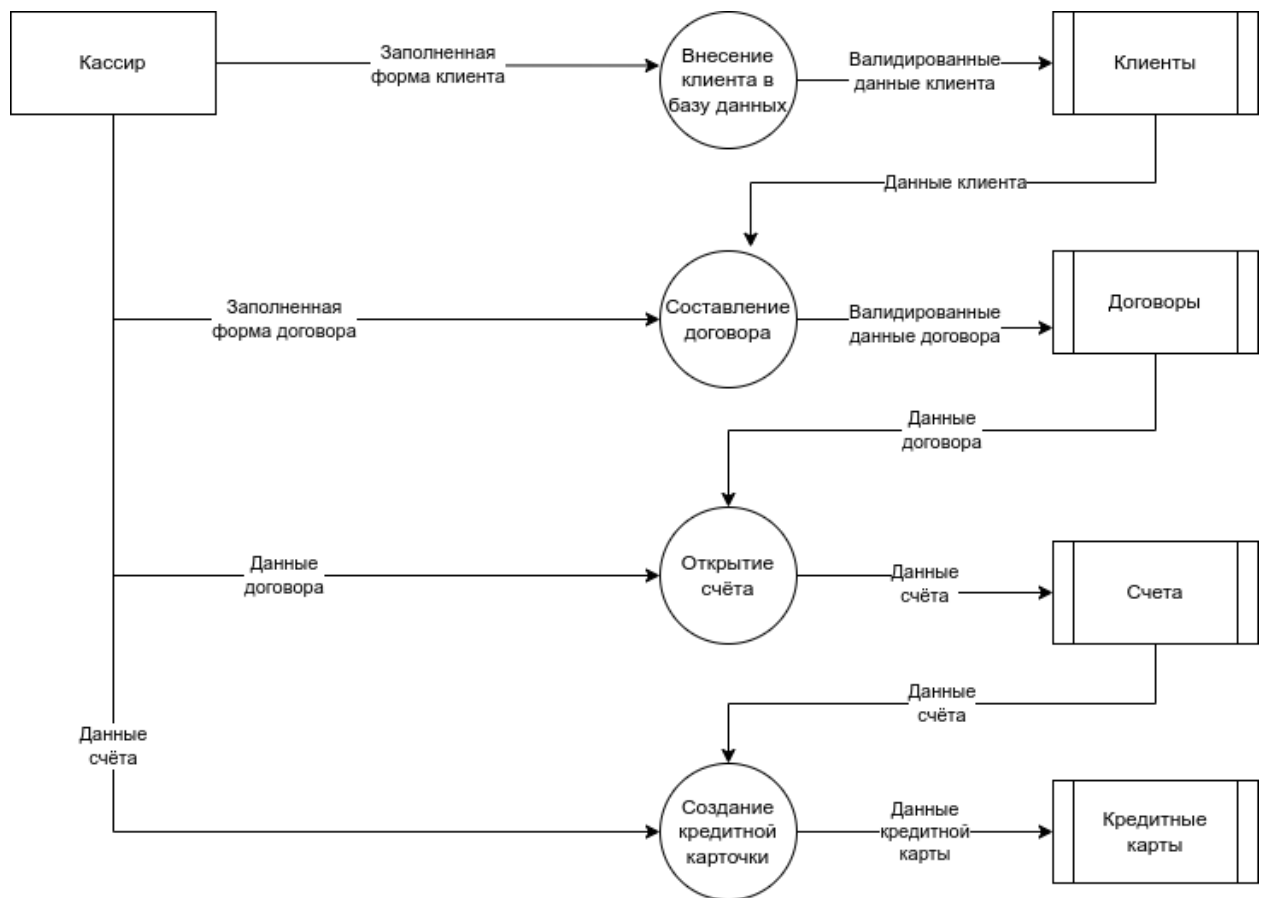


Рисунок 1 — Диаграмма потоков данных информационной системы

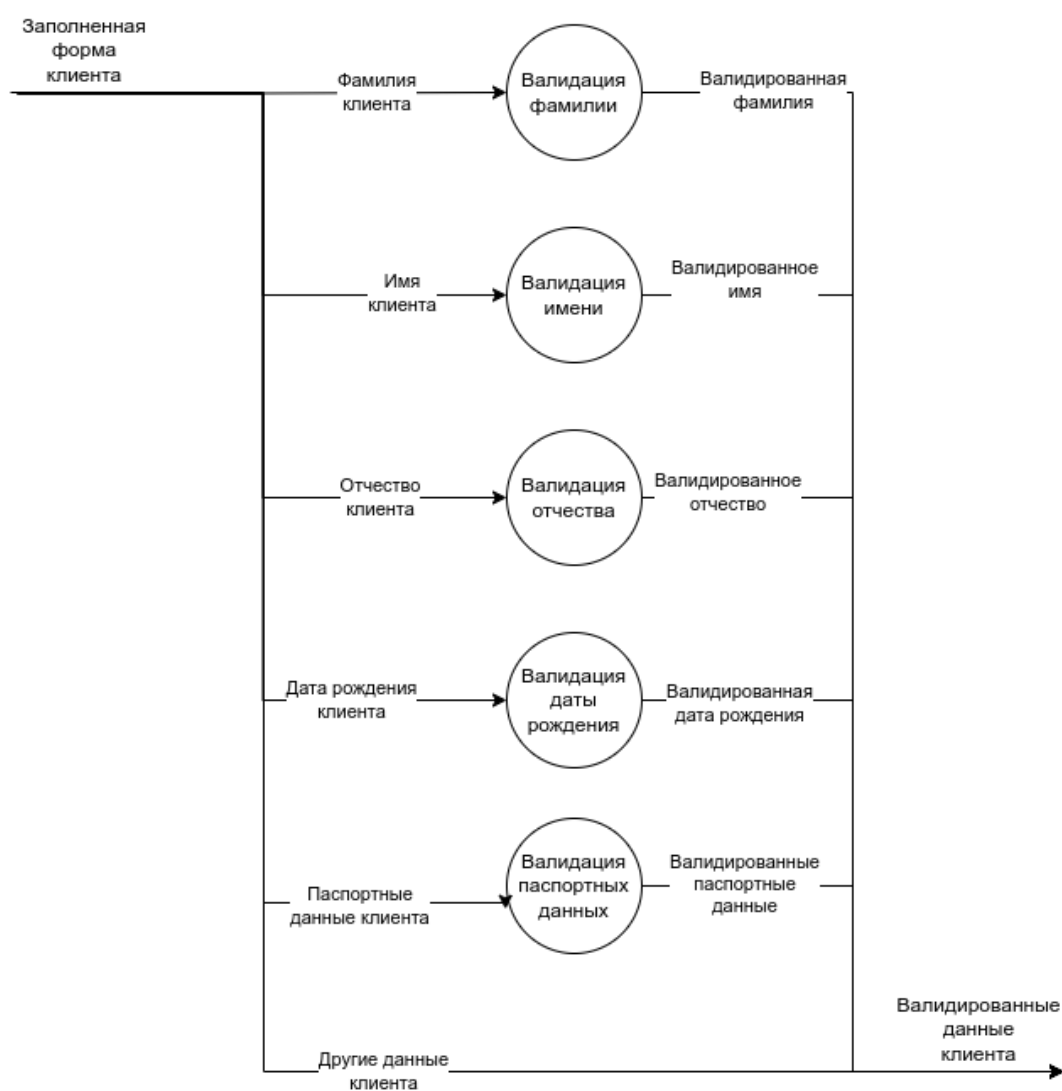


Рисунок 2 — Диаграмма потоков данных процесса ”Внесение клиента в базу данных”

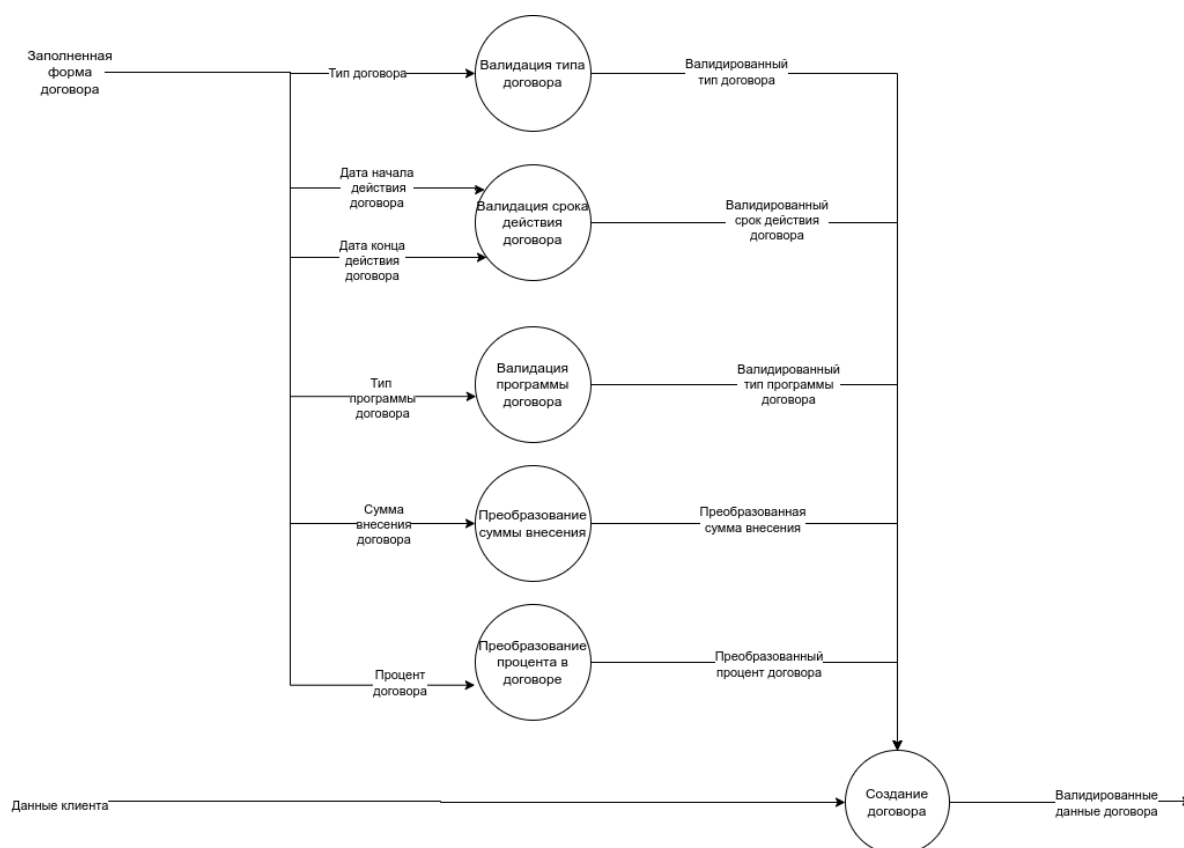


Рисунок 3 — Диаграмма потоков данных процесса "Составление договора"

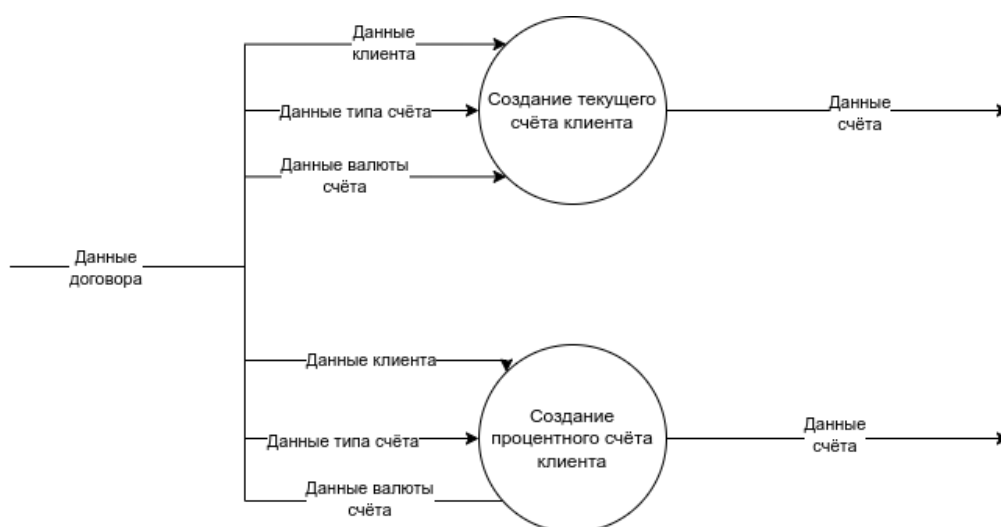


Рисунок 4 — Диаграмма потоков данных процесса "Открытие счёта"



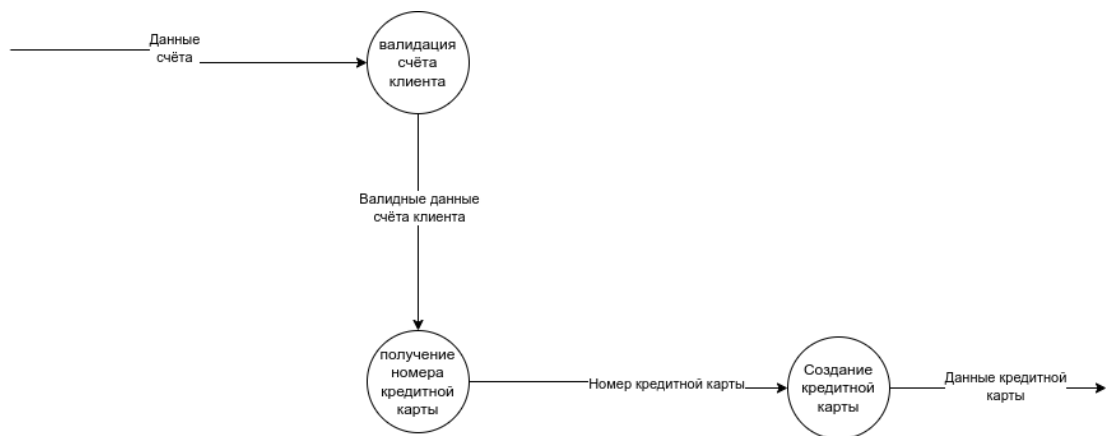


Рисунок 5 — Диаграмма потоков данных процесса "Создание кредитной карты"

## 4 ОБЪЕКТЫ И ИХ АТТРИБУТЫ

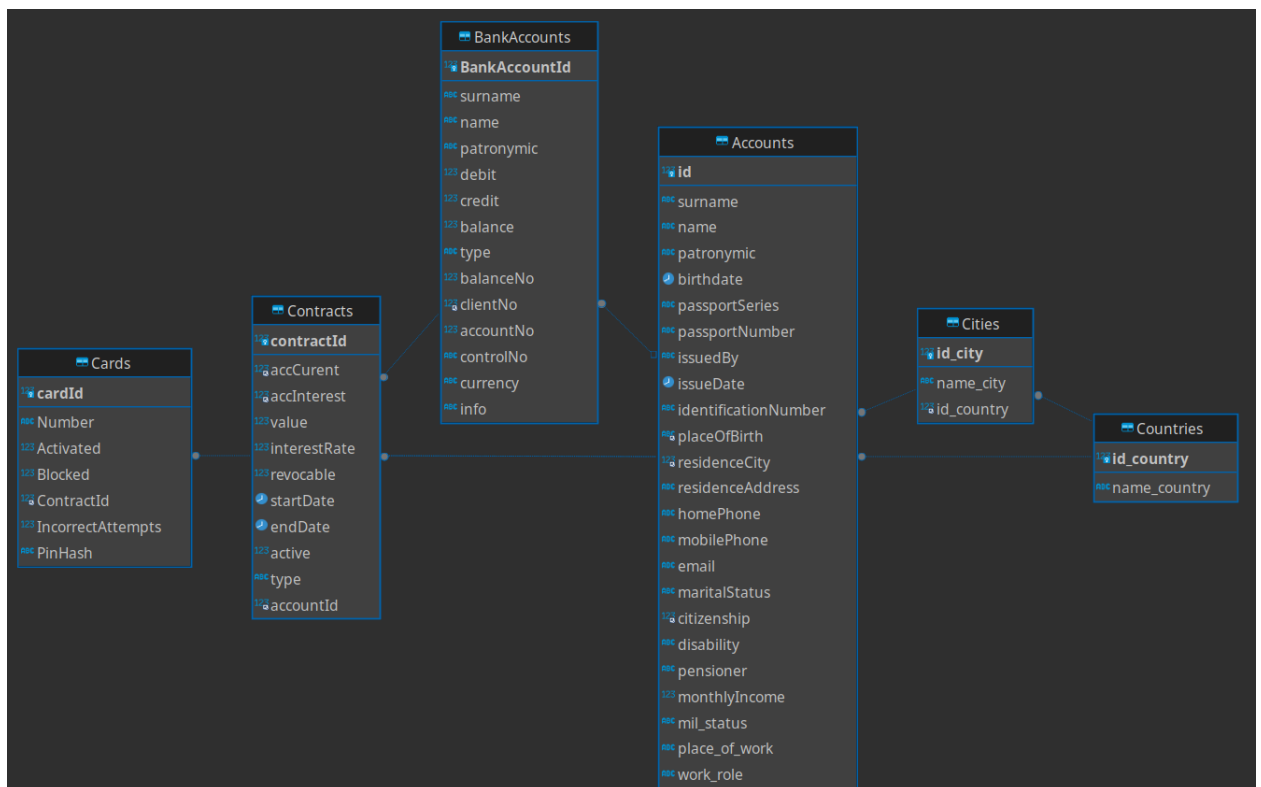


Рисунок 6 — Объекты и атрибуты информационной системы

Cards - Пластиковые карты:

- cardId - уникальный идентификатор;
- Number - номер карты;
- Activated - флаг активности карты;
- Blocked - флаг блокировки карты;
- ContractId - идентификатор привязанного контракта;
- IncorrectAttempts - число неверных попыток ввода пин-кода;
- PinHash - хеш пин-кода карты.

Contracts - Контракты:

- contractId - уникальный идентификатор;
- accCurent - идентификатор привязанного счёта;
- accInterest - идентификатор привязанного счёта;
- value - сумма, на которую заключён контракт;
- interestRate - процент, под который заключён контракт;

- revocable - флаг отзываемости;
- startDate - начало срока действия контракта;
- endDate - окончание срока действия контракта;
- active - флаг активности контракта;
- type - тип контракта;
- accountId - уникальный идентификатор привязанного аккаунта.

Bank Accounts - Банковские счета:

- BankAccountId - уникальный идентификатор;
- surname - фамилия клиента;
- name - имя клиента;
- patronymic - отчество клиента;
- debit - дебет данного счёта;
- credit - кредит данного счёта;
- balance - сальдо данного счёта;
- type - тип счёта;
- balanceNo - номер счёта;
- clientNo - идентификатор клиента;
- accountNo - номер клиента;
- controlNo - контрольный символ;
- currency - валюта счёта;
- info - информация о счёте.

Accounts - аккаунты пользователей:

- id - уникальный идентификатор;
- surname - фамилия пользователя;
- name - имя пользователя;
- patronymic - отчество пользователя;
- birthdate - дата рождения пользователя;
- passportSeries - серия паспорта пользователя;
- passportNumber - номер паспорта пользователя;
- issuedBy - орган выдачи паспорта;
- issueDate - срок действия паспорта;
- identificationNumber - идентификационный номер;
- placeOfBirth - место рождения пользователя;

- residenceCity - город проживания пользователя;
- residenceAddress - адрес проживания пользователя;
- homePhone - домашний телефон пользователя;
- mobilePhone - мобильный телефон пользователя;
- email - почта пользователя;
- maritalStatus - семейный статус пользователя;
- citizenship - гражданство пользователя;
- disability - флаг инвалидности пользователя;
- pensioner - флаг наличия пенсии у пользователя;
- monthlyIncome - доход пользователя за месяц;
- mil\_status - флаг военнообязанности пользователя;
- place\_of\_work - место работы пользователя;
- work\_role - должность пользователя.

#### Cities - Города:

- id\_city - уникальный идентификатор пользователя;
- name\_city - название города;
- id\_country - идентификатор страны города.

#### Countries - страны:

- id\_country - уникальный идентификатор страны;
- name\_country - название страны.

## 5 СЛОВАРЬ ТЕРМИНОВ

Клиент – лицо или организация, которая взаимодействует с банком в рамках финансовых услуг, таких как открытие счетов, получение кредитов и т.д. В информационной системе каждый клиент имеет уникальный идентификатор, а также хранятся его персональные данные и результаты финансовых операций.

Договор – электронный документ в информационной системе, отображающий правовое соглашение между клиентом и банком, определяющий условия финансовых услуг, таких как депозиты, кредиты и другие.

Кредит – сумма средств, которую банк выдает клиенту на определенных условиях, установленных в договоре, с обязательством возврата суммы с процентами.

Депозит – сумма средств, которую клиент размещает на своем счете в банке, за которую банк предоставляет проценты за хранение.

Счёт – электронная запись в информационной системе, отражающая финансовое состояние счета клиента.

Текущий счёт клиента – счет, используемый для проведения операций с текущими финансовыми средствами клиента, такие как оплата кредита, снятие наличных и прочее.

Процентный счёт клиента – счет, на котором начисляются проценты по депозиту или кредитной задолженности.

Кредитная карта – пластиковая карта, выпущенная банком, позволяющая клиенту осуществлять финансовые операции в кредит, снимать наличные средства в банкоматах и т.д. Кредитная карта связана с определенным кредитным счетом.