**Part 1: Выбор Сценария**

Для данной работы выбран сценарий: **Управление арендной недвижимостью:** Отслеживание арендной недвижимости, арендаторов, договоров аренды и запросов на обслуживание.

**Part 2: Проектирование Базы Данных и Документация**

Идентификация Сущностей и Атрибутов:

1. **Lessor**
2. **Lessee**
3. **Leased Property**
4. **Lease Аgreement**
5. **Service request**

**Проектирование Таблиц:**

1. **Название таблицы: Lessor** 
   * **Описание**: Хранит информацию об арендодателях. Также предпологается, что **идентификационного номера паспорта может быть недостаточно** для **уникальности**, поэтому **уникальность проверяется по гражданству и идентификационному номеру**. Также проверяется возраст арендодателя, чтоб он был совершеннолетний(рассматривается **совершеннолетие с 18-лет**) и чтобы не вводились слишком большие значения(**>125 лет люди ещё не жили**)
   * **Атрибуты**
     + *Lessor\_ID: INTEGER, PRIMARY KEY, NOT NULL, UNIQUE;*
     + *First\_Name: VARCHAR(20), NOT NULL;*
     + *Last\_Name: VARCHAR(25), NOT NULL;*
     + *Middle\_Name: VARCHAR(25), NULL;*
     + *Gender: CHAR(1), NOT NULL;*
     + *Date\_Of\_Birth: DATE, NOT NULL;*
     + *Identification\_Number: VARCHAR(15), NOT NULL;*
     + *Citizenship: VARCHAR(20), NOT NULL;*
   * **Ограничения**
     + *PK\_Lessor: PRIMARY KEY (Lessor\_ID);*
     + *UQ\_Lessor\_Data: UNIQUE (Identification\_Number, Citizenship);*
     + *CHK\_Lessor\_Age: CHECK (EXTRACT(YEAR FROM AGE(CURRENT\_DATE, Date\_Of\_Birthday)) >= 18 AND EXTRACT(YEAR FROM AGE(CURRENT\_DATE, Date\_Of\_Birthday)) < 125);*
     + *CHK\_Gender: CHECK (Gender IN (‘M’, ‘F’))*
2. **Название таблицы: Lessee** 
   * **Описание**: Хранит информацию об арендаторах(аналогичные ограничения с таблицей “**Lessor**”)
   * **Атрибуты**
     + *Lessee\_ID: INTEGER, PRIMARY KEY, NOT NULL, UNIQUE;*
     + *First\_Name: VARCHAR(20), NOT NULL;*
     + *Last\_Name: VARCHAR(25), NOT NULL;*
     + *Middle\_Name: VARCHAR(25), NULL;*
     + *Gender: CHAR(1), NOT NULL;*
     + *Date\_Of\_Birth: DATE, NOT NULL;*
     + *Identification\_Number: VARCHAR(15), NOT NULL;*
     + *Citizenship: VARCHAR(20), NOT NULL;*
   * **Ограничения**
     + *PK\_Lessee: PRIMARY KEY (Lessee\_ID);*
     + *UQ\_Lessee\_Data: UNIQUE (Identification\_Number, Citizenship);*
     + *CHK\_Lessee\_Age: CHECK (EXTRACT(YEAR FROM AGE(CURRENT\_DATE, Date\_Of\_Birthday)) >= 18 AND EXTRACT(YEAR FROM AGE(CURRENT\_DATE, Date\_Of\_Birthday)) < 125);*
     + *CHK\_Gender: CHECK (Gender IN (‘M’, ‘F’))*
3. **Название таблицы:** **Leased\_Property**
   * **Описание**: Хранит информацию о недвижимости. Предпологается, что **адрес** будет **введён полностью**(область, район, населённый пункт, улица, номер дома или квартиры)). В поле **Apartment\_Number** значение по умолчанию 0 создаётся для частных домов, которые не имеют нумерацию по квартирам.
   * **Атрибуты**
     + *Property\_ID: INTEGER, PRIMARY KEY, NOT NULL, UNIQUE;*
     + *Region: VARCHAR(25), NOT NULL;*
     + *District: VARCHAR(25), NOT NULL;*
     + *Locality: VARCHAR(30) NOT NULL;*
     + *Street: VARCHAR(30), NOT NULL;*
     + *Building\_Number: INTEGER, NOT NULL;*
     + *Apartment\_Number: INTEGER, NOT NULL, DEFAULT 0;*
     + *Area: NUMERIC(5,2), NOT NULL;*
     + *Room\_Count: SMALLINT, NOT NULL;*
     + *Lessor\_ID: INTEGER, FOREIGN KEY (Lessor), NOT NULL;*
   * **Ограничения**
     + *PK\_Property: PRIMARY KEY (Property\_ID);*
     + *CHK\_Area\_And\_Rooms: CHECK (Room\_Count > 0 AND Area > 0);*
     + *FK\_Property\_Lessor: FOREIGN KEY (Lessor\_ID) REFERENCES Lessor(Lessor\_ID);*
     + *CHK\_Building\_Number: CHECK (Building\_Number > 0);*
     + *CHK\_Apartment\_Number: CHECK(Apartment\_Number >= 0);*
4. **Название таблицы:** **Lease\_Аgreement**
   * **Описание**: Хранит информацию о договоре. Предпологается, что **номер договора – это не только цифры, но и символы**.
   * **Атрибуты**
     + *Lease\_Agreement\_ID: INTEGER, PRIMARY KEY, NOT NULL, UNIQUE;*
     + *Agreement\_Number: VARCHAR(20), NOT NULL, UNIQUE;*
     + *Signing\_Date: DATE, NOT NULL;*
     + *Lease\_Start\_Date: DATE, NOT NULL;*
     + *Lease\_End\_Date: DATE, NOT NULL;*
     + *Price: NUMERIC, NOT NULL;*
     + *Additional\_Conditions: TEXT, NULL;*
     + *Lessee\_ID: INTEGER, FOREIGN KEY (Lessee), NOT NULL;*
     + *Property\_ID: INTEGER, FOREIGN KEY (Leased\_Property), NOT NULL;*
   * **Ограничения**
     + *PK\_Lease\_Agreement: PRIMARY KEY (Lease\_Agreement\_ID);*
     + *CHK\_Price: CHECK (Price > 0);*
     + *CHK\_Lease\_Start\_End: CHECK(Lease\_End\_Date > Lease\_Start\_Date)*
     + *FK\_Lease\_Lessee: FOREIGN KEY (Lessee\_ID) REFERENCES Lessee(Lessee\_ID);*
     + *FK\_Lease\_Property: FOREIGN KEY (Property\_ID) REFERENCES Leased\_Property(Property\_ID);*
5. **Название таблицы: Service\_Request**
   * **Описание**: Хранит информацию о **запросах на обслуживание(по причине ремонта, и т.д.)**
   * **Атрибуты**
     + *Request\_ID: INTEGER, PRIMARY KEY, NOT NULL, UNIQUE;*
     + *Request\_Date: DATE, NOT NULL;*
     + *Request\_Description: TEXT, NOT NULL;*
     + *Request\_Status: VARCHAR(15), NOT NULL, DEFAULT ‘NEW’;*
     + *Lease\_Agreement\_ID: INTEGER, FOREIGN KEY (Lease\_Agreement), NOT NULL;*
   * **Ограничения**
     + *PK\_Request: PRIMARY KEY (Request\_ID);*
     + *FK\_Request\_Lease: FOREIGN KEY (Lease\_Agreement\_ID) REFERENCES Lease\_Agreement(Lease\_Agreement\_ID);*

**Взаимосвязи:**

* **Lessor** и **Leased\_Property** (Один-ко-Многим): Одному арендодателю может принадлежать множество арендной недвижимости, но каждая арендная недвижимость в этом варианте имеет одного владельца.
  + **Leased\_Property.Property\_Owner** является внешним ключом, ссылающимся на **Lessor.Lessor\_ID**.
* **Lessor** и **Lease\_Аgreement** (Один-ко-Многим): Одному арендодателю может принадлежать множество договоров об аренде, но каждый договор имеет одного владельца.
  + **Lease\_Аgreement.Lessor\_ID** является внешним ключом, ссылающимся на **Lessor.Lessor\_ID**.
* **Lessee** и **Lease\_Аgreement** (Один-ко-Многим): Одному арендатору может принадлежать множество договоров об аренде, но каждый договор имеет одного арендатора.
  + **Lease\_Аgreement.Lessee\_ID** является внешним ключом, ссылающимся на **Lessee.Lessee\_ID**.
* **Leased\_Property** и **Lease\_Аgreement** (Один-ко-Многим): Одной недвижимости может принадлежать множество договоров об аренде, но в данном варианте в каждом договоре прописывается лишь одна недвижимость.
  + **Lease\_Аgreement.Property\_ID** является внешним ключом, ссылающимся на **Leased\_Property.Property\_ID**.
* **Lease\_Аgreement** и **Service\_Request** (Один-ко-Многим): Одному договору может принадлежать множество запросов, но каждый запрос прикреплён к одному договору.
  + **Service\_Request.Lease\_Agreement\_ID** является внешним ключом, ссылающимся на **Lease\_Аgreement.Lease\_Agreement\_ID**.
* **Lessee** и **Leased\_Property** (Многие-ко-Многим): Один арендатор может снимать несколько помещений. Один объект недвижимости может быть в аренде у нескольких людей(пример: квартира, сдающаяся по комнатам)
  + **Lease\_Agreement.Lessee\_ID** является внешним ключом, ссылающимся на **Lessee.Lessee\_ID**;
  + **Lease\_Agreement.Property\_ID** является внешним ключом, ссылающимся на **Leased\_Property.Property\_ID.**

**Part 3: ER-Диаграмма**

