Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №1**

По дисциплине: «Анализ больших данных»

Тема: «**Реляционные данные. Исследовательский анализ данных.**

**Построение визуализаций данных OLAP**»

|  |
| --- |
| Выполнил:  студент группы  Б.ПИН.РИС - 21.06  Иванов А.В. |
| Проверила:  старший преподаватель  кафедры ПО  Корнеева Е.И. |

Тверь 2025

**Формулировка задач, описание условий**

**Задание на лабораторную работу**

1. Создать базу из файлов или скриптов согласно варианту. Сложность программы выбрать из представленных в тексте лабораторной работы.

2.  Подключиться к базе данных из python.

3.  Сделать описание данных. Из каких таблиц и полей состоят данные таблиц? Какие из данных являются признаками? К какому типу данных и к какой шкале относятся признаки?

4. Одномерный анализ. Построить гистограммы распределения количественных признаков, которые важны для задачи. Сделать вывод после построения. Какое распределение для каждого из признаков? Почему, по вашему мнению, признаки важны для задачи?

5. Многомерный анализ. Построить графики из 3-4 признаков. Выбрать категориальные (номинальные, порядковые или бинарные) признаки и количественные. Что получилось на каждом графике? Почему, по вашему мнению, признаки важны для задачи?

**Вариант и условия задач. Указать какой сложности выполняется задание.**

Вариант 5.

**Сложность: Rare**

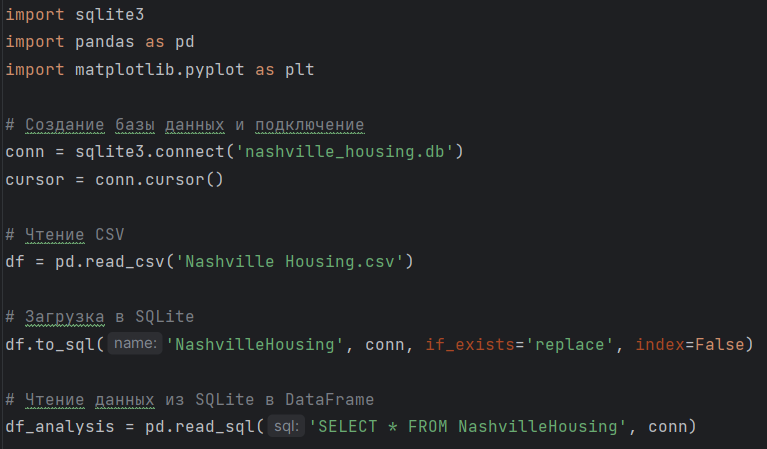
* Реализовать Jupiter Notebook или консольное приложение для выполнения задания.
* В качестве базы данных выбрать sqlite3. Создать базу из файлов или скриптов согласно варианту.
* Подключиться к базе данных из python.
* Сделать описание данных и выводы по заданию.
* Соединить признаки в 1 таблицу pandas для анализа
* Одномерный анализ. Построить 2 гистограммы распределения количественных признаков, которые важны для задачи и сделать их описание по заданию.
* Многомерный анализ. Построить хотя бы 1 график из 3-4 признаков и сделать его описание по заданию.

**Ссылка на репозиторий с программной реализацией**

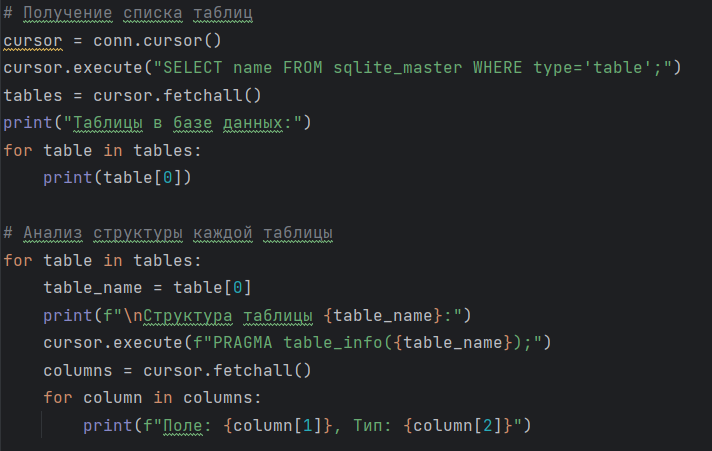
https://github.com/ArtyomI69/data\_science

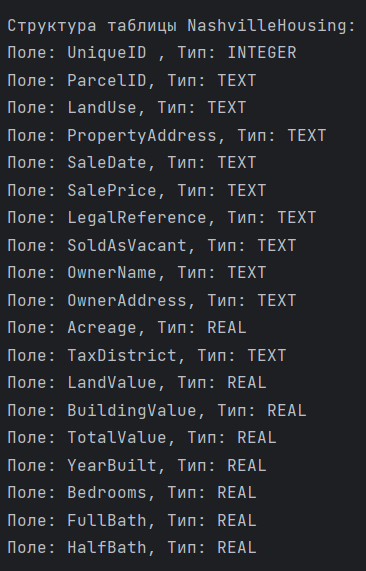
# Описание проделанной работы

## 1. Создание базы данных SQLite3 и подключение из Python



## 2. Описание данных



**Схема БД**:  


**Описание столбцов в таблице NashvilleHousing с указанием их признаков**:

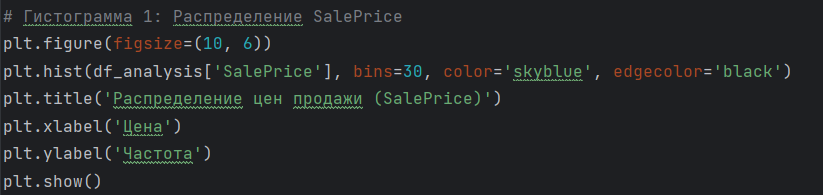
1. **UniqueID** (INTEGER) – уникальный идентификатор записи (категориальный, идентификатор).
2. **ParcelID** (TEXT) – идентификатор земельного участка (категориальный, идентификатор).
3. **LandUse** (TEXT) – тип использования земли (категориальный).
4. **PropertyAddress** (TEXT) – адрес недвижимости (категориальный).
5. **SaleDate** (TEXT) – дата продажи (временной признак).
6. **SalePrice** (TEXT) – цена продажи (числовой, но записан как текст, возможно, требует преобразования).
7. **LegalReference** (TEXT) – юридическая ссылка (категориальный).
8. **SoldAsVacant** (TEXT) – была ли недвижимость продана как пустой участок (категориальный, бинарный).
9. **OwnerName** (TEXT) – имя владельца (категориальный).
10. **OwnerAddress** (TEXT) – адрес владельца (категориальный).
11. **Acreage** (REAL) – площадь участка в акрах (числовой, непрерывный).
12. **TaxDistrict** (TEXT) – налоговый округ (категориальный).
13. **LandValue** (REAL) – стоимость земли (числовой, непрерывный).
14. **BuildingValue** (REAL) – стоимость здания (числовой, непрерывный).
15. **TotalValue** (REAL) – общая стоимость недвижимости (числовой, непрерывный).
16. **YearBuilt** (REAL) – год постройки (числовой, дискретный).
17. **Bedrooms** (REAL) – количество спален (числовой, дискретный).
18. **FullBath** (REAL) – количество полных ванных комнат (числовой, дискретный).
19. **HalfBath** (REAL) – количество половинных ванных комнат (числовой, дискретный).

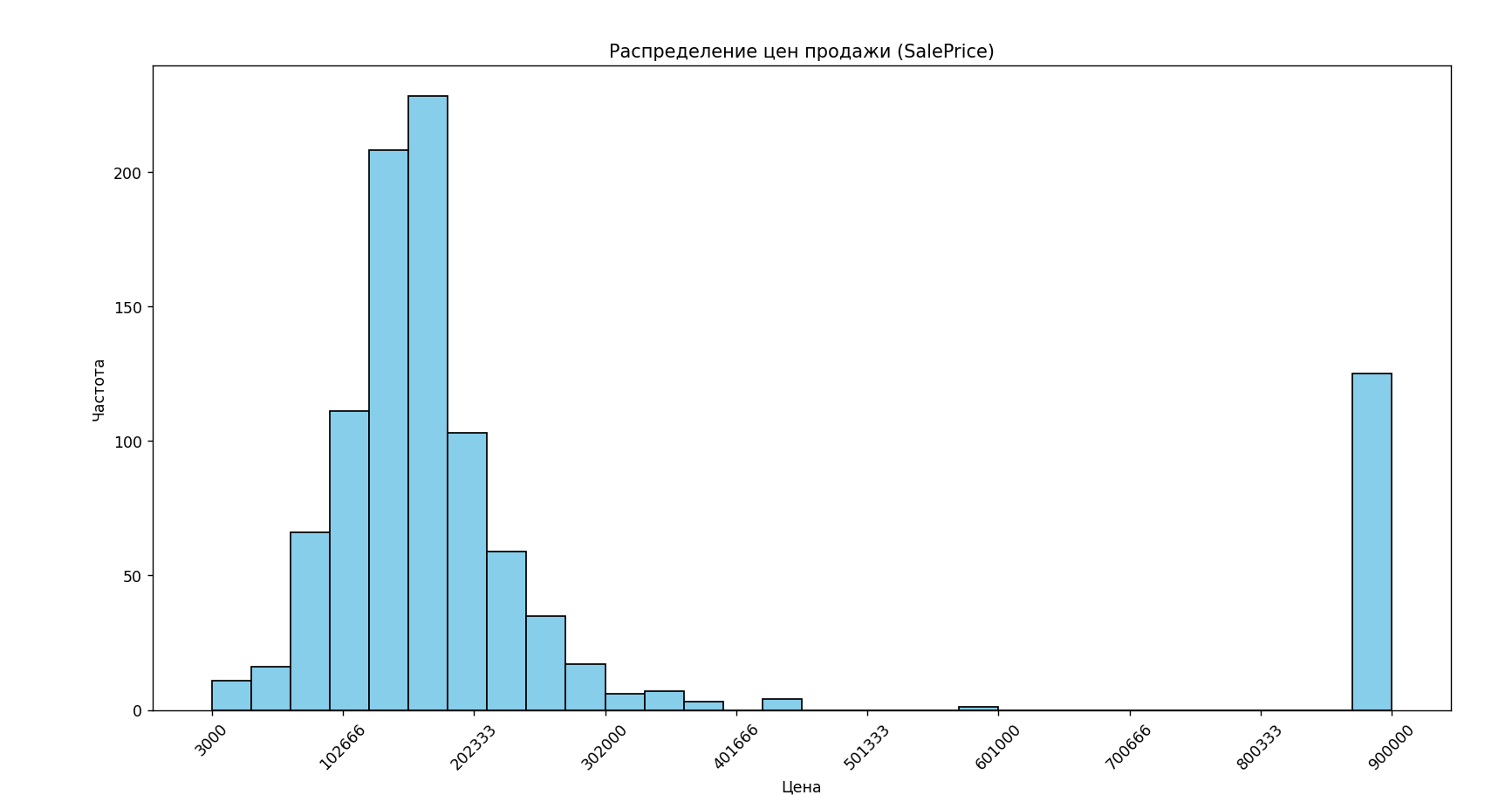
**Признаки, важные для задачи прогнозирования стоимости:**

* SalePrice (целевая переменная).
* YearBuilt, Acreage, Bedrooms, LandValue, BuildingValue.

## 3. Одномерный анализ

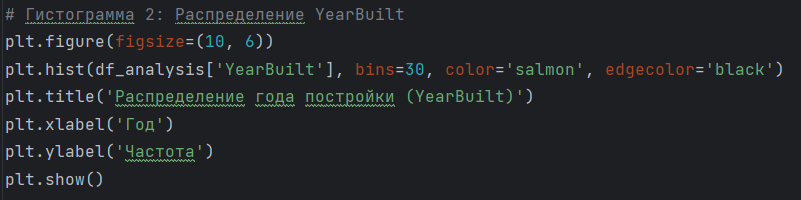
Распределение SalePrice

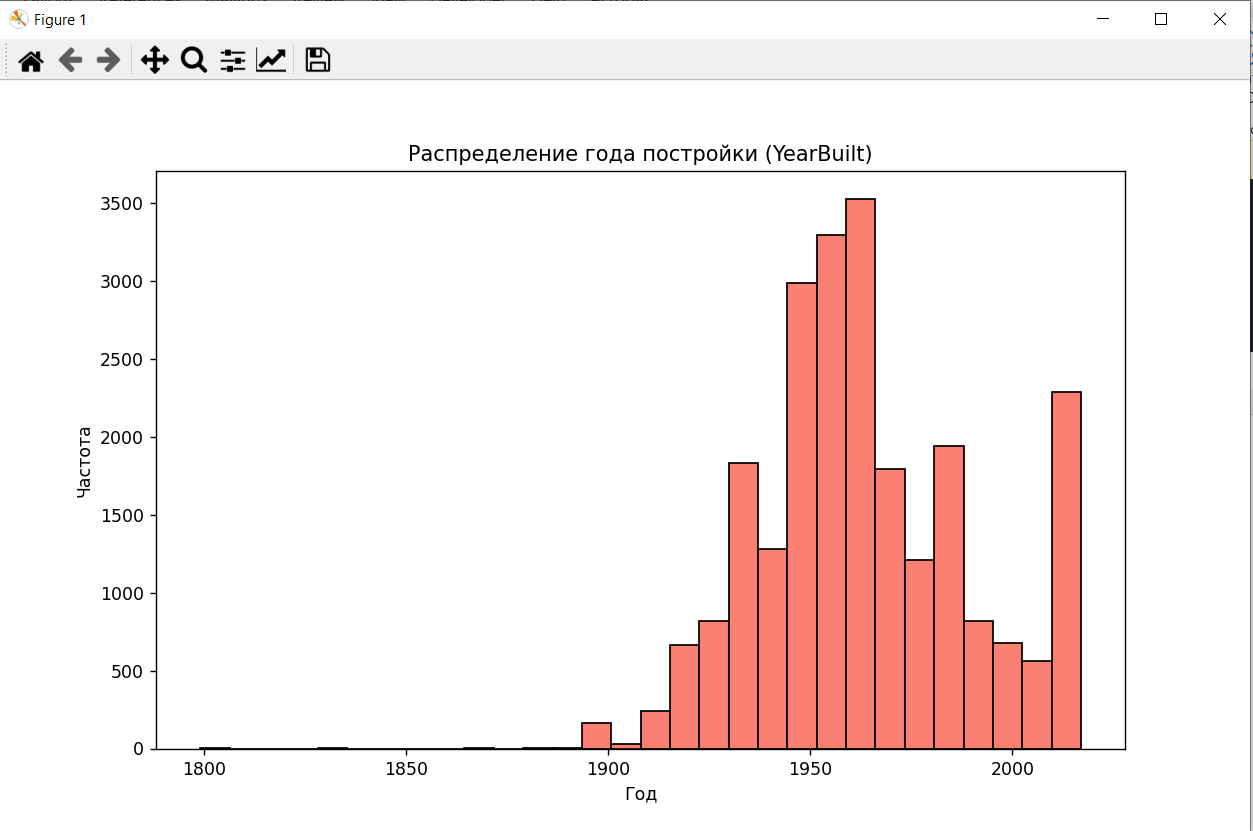




1. **Форма распределения**: Гистограмма показывает, что большинство цен на недвижимость сосредоточено в диапазоне от **100 000 до 250 000**. Распределение имеет **правостороннюю асимметрию**, что означает, что есть небольшое количество объектов с очень высокими ценами.
2. **Выбросы**: Видно несколько значений, сильно отличающихся от основной массы данных. Например, есть отдельное значение около **800 000**, что можно считать выбросом.
3. **Мода**: Наиболее частые цены сосредоточены в диапазоне **150 000 – 200 000**, что указывает на наиболее популярные ценовые категории.

Распределение YearBuilt

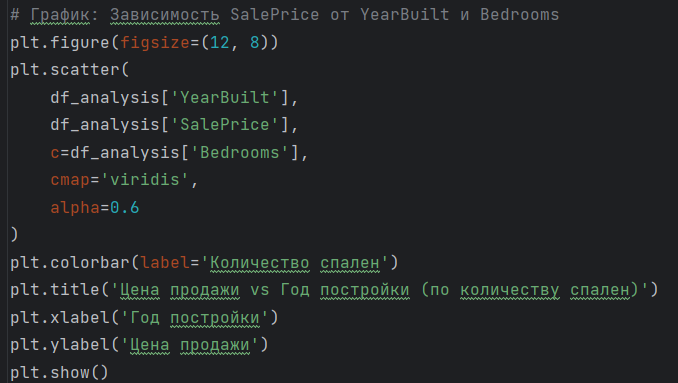


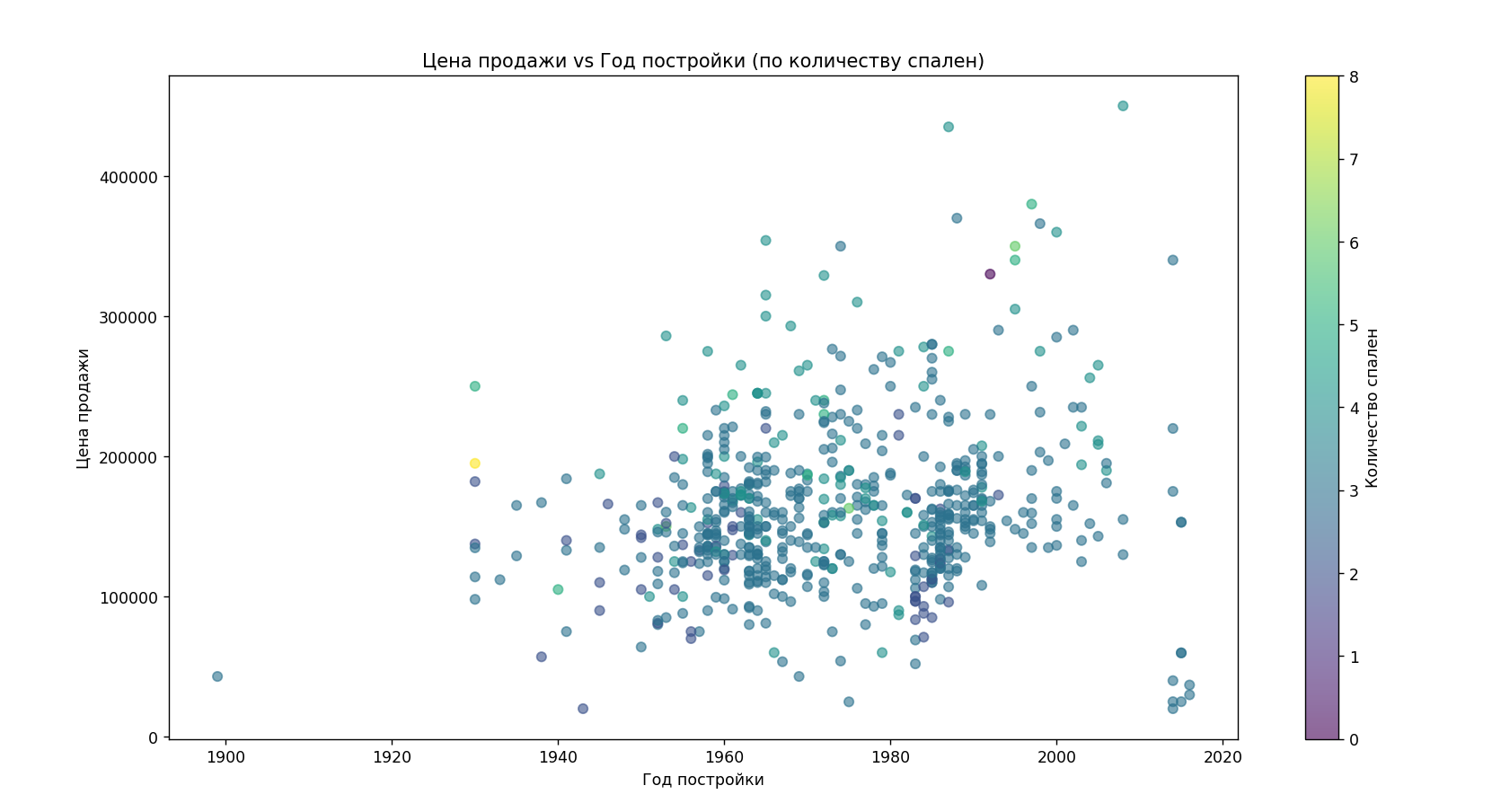


1. **Форма распределения**: Основное количество домов построено после **1900 года**, с заметным ростом после **1950 года**.
2. **Пик строительства**: Максимальное количество построенных домов приходится на **1950–1980 годы**, затем идет спад, а после **2000 года** наблюдается новый рост.
3. **Аномалии**: Можно заметить небольшие значения до **1900 года**, но они являются редкими. Также резкий скачок после **2000 года** может указывать на активное строительство в последние годы.

## 4. Многомерный анализ

Зависимость SalePrice от YearBuilt и Bedrooms





**Вывод:**

* Более новые дома (после 2000 года) имеют более высокую стоимость.
* Дома с большим количеством спален (3-5) концентрируются в верхнем ценовом диапазоне.
* Видна положительная корреляция между годом постройки и ценой.

**Краткий вывод по работе. Описание реализованной программы и её функций.**

В данной лабораторной работе мы перенесли таблицу из csv в sqlite3.

Нашли признаки.

И сделали одномерный и многомерный анализ.

* **SalePrice** и **YearBuilt** — ключевые признаки для анализа стоимости недвижимости.
* Многомерный анализ показывает, что новые дома с большим количеством спален имеют более высокую стоимость.

**Ссылки на используемые материалы. Документация**

Документация Python sqlite3: [**https://docs.python.org/3/library/sqlite3.html**](https://docs.python.org/3/library/sqlite3.html)

Pandas: [**https://pandas.pydata.org/docs**](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html)

Matplotlib : [**https://matplotlib.org**](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html)

**Листинг кода**

import sqlite3  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
  
def load\_data\_to\_db(csv\_file, db\_file):  
 df = pd.read\_csv(csv\_file)  
 with sqlite3.connect(db\_file) as conn:  
 df.to\_sql('NashvilleHousing', conn, if\_exists='replace', index=False)  
 return conn  
  
  
def analyze\_db\_structure(conn):  
 cursor = conn.cursor()  
 cursor.execute("SELECT name FROM sqlite\_master WHERE type='table';")  
 tables = cursor.fetchall()  
 print("Таблицы в базе данных:")  
 for table in tables:  
 print(table[0])  
  
 for table in tables:  
 table\_name = table[0]  
 print(f"\nСтруктура таблицы {table\_name}:")  
 cursor.execute(f"PRAGMA table\_info({table\_name});")  
 columns = cursor.fetchall()  
 for column in columns:  
 print(f"Поле: {column[1]}, Тип: {column[2]}")  
  
  
def fetch\_data(conn, query):  
 return pd.read\_sql(query, conn)  
  
  
def clean\_data(df):  
 try:  
 df['SalePrice'] = df['SalePrice'].replace('[\$,]', '', regex=True).astype(float)  
 except Exception as e:  
 print(f"Ошибка обработки данных: {e}")  
 return df  
  
  
def plot\_histogram(data, column, bins, color, title, xlabel, ylabel):  
 plt.figure(figsize=(10, 6))  
 plt.hist(data[column], bins=bins, color=color, edgecolor='black')  
 plt.title(title)  
 plt.xlabel(xlabel)  
 plt.ylabel(ylabel)  
 plt.xticks(rotation=45)  
 plt.show()  
  
  
def plot\_scatter(data, x\_col, y\_col, c\_col, cmap, title, xlabel, ylabel):  
 plt.figure(figsize=(12, 8))  
 plt.scatter(data[x\_col], data[y\_col], c=data[c\_col], cmap=cmap, alpha=0.6)  
 plt.colorbar(label='Количество спален')  
 plt.title(title)  
 plt.xlabel(xlabel)  
 plt.ylabel(ylabel)  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 db\_connection = load\_data\_to\_db('Nashville Housing.csv', 'nashville\_housing.db')  
 analyze\_db\_structure(db\_connection)  
  
 df\_analysis = fetch\_data(db\_connection, 'SELECT \* FROM NashvilleHousing LIMIT 1000')  
 df\_analysis1 = fetch\_data(db\_connection, 'SELECT \* FROM NashvilleHousing')  
  
 df\_analysis = clean\_data(df\_analysis)  
  
 plot\_histogram(df\_analysis, 'SalePrice', bins=30, color='skyblue', title='Распределение цен продажи (SalePrice)',  
 xlabel='Цена', ylabel='Частота')  
 plot\_histogram(df\_analysis1, 'YearBuilt', bins=30, color='salmon', title='Распределение года постройки (YearBuilt)',  
 xlabel='Год', ylabel='Частота')  
 plot\_scatter(df\_analysis, 'YearBuilt', 'SalePrice', 'Bedrooms', cmap='viridis',  
 title='Цена продажи vs Год постройки (по количеству спален)', xlabel='Год постройки',  
 ylabel='Цена продажи')