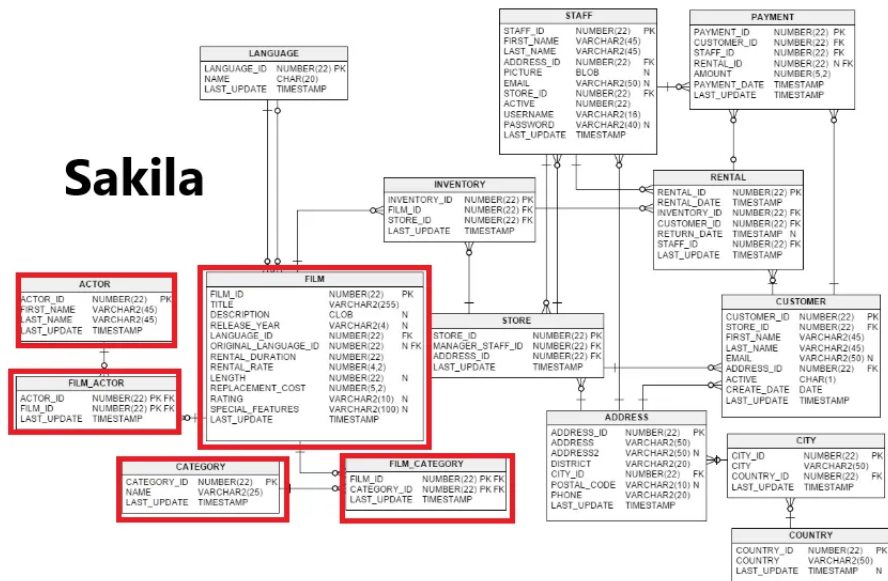


Консольное приложение поиска фильмов на Python

Sakila



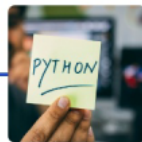
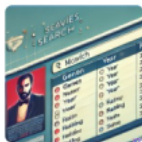
Обзор учебного проекта для поиска фильмов, реализованного с использованием языка программирования Python, включая его архитектуру, технологии и ключевые функции.

автор: Ольга Полищук

https://github.com/OlgaPolish/MyProject_Film

Консольное приложение поиска фильмов

Краткий обзор проекта и технологий



Описание проекта

Консольное приложение для поиска фильмов в базе данных, реализованное на Python.

Основные цели

Создание удобного интерфейса для поиска и анализа фильмов, регистрация популярных запросов.

Используемые технологии

Проект разработан на Python с базой данных MySQL и библиотеками pymysql, prettytable, dotenv.

Потенциал Python

Демонстрация возможностей языка Python для разработки приложений с взаимодействием с базами данных.

Этапы разработки технологий

Краткий обзор основных этапов разработки приложений

Этап 1: Подключение к базе данных

База данных Sakila подключена на локальный сервер; исследована структура таблиц.

Этап 2: Подключение приложения

Приложение осуществляет коннект к базе данных фильмов Sakila на локальном сервере.

Этап 3: Разработка SQL-запросов

Созданы и протестированы SQL-запросы для извлечения фильмов по различным фильтрам.

Интеграция SQL-запросов в Python

Настройка взаимодействия приложения с базой данных, выполнением SQL-запросов. Обработка и вывод результатов поиска по жанрам и годам выпуска.

Сохранение ключевых слов

Ключевые слова поиска сохраняются в отдельной таблице для оптимизации поиска.

Ключевые функции приложения для поиска фильмов

Функции для улучшения пользовательского опыта

```
| 1 | rating_and_year | 20 |
| 2 | category_and_year | 13 |
| 3 | keyword | 11 |
+---+-----+-----+
```

Enter the query number to get all records: 2

Total rows found: 9

```
+---+-----+-----+
| № | ID_Query | Query_Type | Keyword |
+---+-----+-----+
| 1 | 3 | category_and_year | Category ID: 1, Year: 2000 |
| 2 | 2 | category_and_year | Category ID: 9, Year: 2000 |
| 3 | 2 | category_and_year | Category ID: 2, Year: 2001 |
| 4 | 1 | category_and_year | Category ID: 3, Year: 2006 |
| 5 | 1 | category_and_year | Category ID: 10, Year: 2006 |
| 6 | 1 | category_and_year | Category ID: 11, Year: 2001 |
| 7 | 1 | category_and_year | Category ID: 13, Year: 2005 |
| 8 | 1 | category_and_year | Category ID: 16, Year: 2000 |
| 9 | 1 | category_and_year | Category ID: 15, Year: 12 |
+---+-----+-----+
```

Press any key to continue...

You can search movies by the following parameters:

1. Keyword in description.
2. Category and year.
3. Rating and year.
4. View popular queries.
5. Exit.

01 ПОИСК ФИЛЬМОВ

Позволяет пользователям находить фильмы по различным критериям с помощью SQL-запросов и фильтров.

02 РЕГИСТРАЦИЯ ПОИСКОВЫХ ТЕРМИНОВ

Автоматически регистрирует поисковые запросы в таблице для анализа популярных терминов.

03 РЕЙТИНГ ПОПУЛЯРНОСТИ

Отображает наиболее искомые термины, упрощая пользователям поиск актуальных фильмов.

Структура модулей приложения

Обзор ключевых модулей и их функциональности

Модуль db.py

Функции для подключения к базе данных и обработки ошибок подключения.

Модуль print_results.py

Отвечает за вывод результатов запросов в виде таблицы с пагинацией.

Модуль ui.py

Обрабатывает пользовательский интерфейс, включая отображение меню и обработку выбора пользователя.

Модуль main.py

Основной файл, запускающий приложение и обрабатывающий пользовательский ввод.

Модуль queries.py

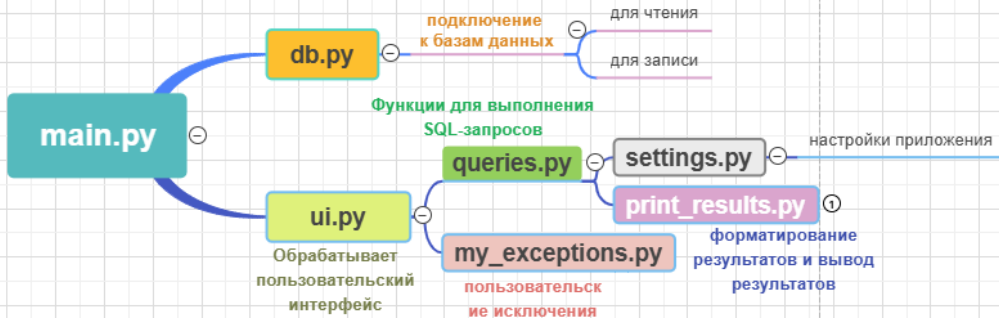
Содержит функции для выполнения различных SQL-запросов, таких как поиск фильмов.

Модуль my_exceptions.py

Определяет функции для обработки пользовательских запросов на продолжение или выход.

Модуль settings.py

Определяет настройки, например такие как цветовые коды ANSI для консольного вывода.



Заключение и выводы проекта

Анализ возможностей и выводы проекта

Эффективное использование SQL-запросов

Использование SQL-запросов позволило эффективно извлекать данные о фильмах из базы данных.

Возможности для дальнейшего расширения

Приложение может быть расширено за счет добавления графического интерфейса или интеграции с API для получения актуальных данных.



Важность структурирования кода

Структурирование кода значительно улучшает модульность и читаемость приложения, облегчая его поддержку.