

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»

Индивидуальный проект

по дисциплине «Хранение и обработка данных»

На тему:

База Данных с данными о клавиатурах

Выполнил:

Казумьян Максим Евгеньевич R3180

Санкт-Петербург, 2024г.

1. Введение / Описание предметной области

В сообществе кастомных клавиатур есть проблема неорганизованности данных. Проектов много, но нет единого места где можно отфильтровать клавиатуры по параметрам и интересующим характеристикам (Количество кнопок, открытый исходный код, эргономика и тп). В таблице keyboards хранятся характеристики клавиатуры, а в остальных таблицах значения к некоторым ключам в характеристиках.

2. Цель

Разработка консольного приложения, которое будет выдавать список клавиатур по необходимым параметрам

3. Ожидаемые запросы к предметной области:

1. Поиск по кол-ву кнопок. Запрос должен возвращать названия клавиатур с количеством кнопок больше N
2. Поиск по стране производителя. Запрос должен возвращать клавиатуры в формате: {«название клавиатуры», «производитель»}
3. Поиск по раскладке и году основания компании-производителя, для поиска винтажных клавиатур. Запрос должен возвращать клавиатуры в формате {«название клавиатуры», «название производителя»}
4. Поиск по конструкции корпуса. Запрос должен возвращать названия клавиатур с конструкцией корпуса, заданной пользователем.
5. Поиск клавиатур с открытым исходным кодом. Запрос должен возвращать клавиатуры с открытым исходным кодом.

4. Архитектура будущей базы данных

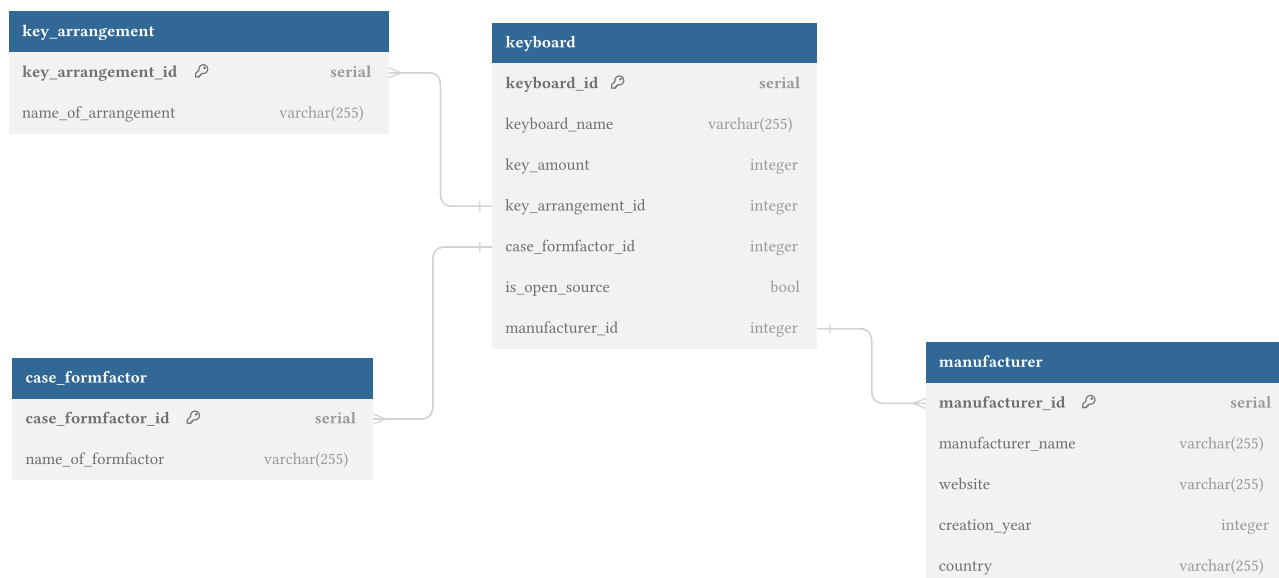


Таблица 1. key_arrangement - раскладки

Название	Тип	Описание
key_arrangement_id	SERIAL	Идентификатор раскладки
name_of_arrangement	VARCHAR(255)	Название раскладки

Есть разные стандарты раскладок, Европейская ISO, Американская ANSI, Японская JIS и т.д. Помимо этого существуют и другие вариации, например, ортолинейная раскладка, где клавиши идут по ровной сетке, без сдвига.

Таблица 2. case_formfactor - форм-факторы корпуса

Название	Тип	Описание
case_formfactor_id	SERIAL	Идентификатор форм-фактора
name_of_formfactor	VARCHAR(255)	Название форм-фактора

Некоторые клавиатуры цельные, некоторые разделены на две половинки (для эргономики), некоторые разделены на две половинки, но все равно соединены одним корпусом, и так далее. Это и покрывает таблица выше.

Таблица 3. manufacturer - производители

Название	Тип	Описание
manufacturer_id	SERIAL	Идентификатор производителя
manufacturer_name	VARCHAR(255)	Название производителя
website	VARCHAR(255)	Веб-сайт производителя
creation_year	INTEGER	Дата основания компании-производителя

Таблица с производителями, говорит сама за себя.

Таблица 4. keyboard - клавиатуры

Название	Тип	Описание
keyboard_id	SERIAL	Идентификатор клавиатуры
keyboard_name	VARCHAR(255)	Название клавиатуры
key_amount	VARCHAR(255)	Количество кнопок
key_arrangement_id	INTEGER	Идентификатор раскладки
case_formfactor_id	INTEGER	Идентификатор форм-фактора
is_open_source	INTEGER	Наличие открытого исходного кода
manufacturer_id	INTEGER	Идентификатор производителя

И итоговая таблица с ссылками на клавиатуры.

5. Получение данных

1. Сначала с помощью скрипта `create_db.py` создается база данных и таблицы. В самом начале файла задаются данные для работающего postgres сервера.
2. Данные заполняются рандомом, с помощью python-скрипта `create_db.py`. Для названий компаний используются случайные сущ + прил, а для названий клавиатур ещё добавляется кол-во кнопок.

При ошибках (допустим, если создаются две компании с одинаковым названием, или клавиатуре рандомно выпал референс к компании, которой не существует) пропускается сгенерированная строка и скрипт идет дальше. Сути не меняет, все равно получается база на 100+ записей.

6. Результат

Для обращения к базе надо использовать скрипт `cli_app.py`. Обращение к элементам происходит через аргументы командной строки, реализация сделана через библиотеку `argparse`.

Доступные команды выводятся через флаг `-h`. Используя данный флаг с командой можно получить информацию о том, какие аргументы надо ввести.

Доступные команды:

1. `search_key_amount`
2. `search_country`
3. `search_vintage`
4. `search_case_formfactor`
5. `search_open_source`

На след. странице несколько примеров

Вывод скрипта с флагом -h

```
sat@mbp-sat dsap % python3 cli_app.py -h
usage: cli_app.py [-h] {search_key_amount,search_country,search_vintage,search_case_formfactor,search_open_source} ...

best keyboard search engine!!

positional arguments:
  {search_key_amount,search_country,search_vintage,search_case_formfactor,search_open_source}

options:
  -h, --help            show this help message and exit
sat@mbp-sat dsap %
```

Пример помощи по доступным аргументам для команды

```
sat@mbp-sat dsap % python3 cli_app.py search_vintage -h
usage: cli_app.py search_vintage [-h] key_arrangement creation_year

positional arguments:
  key_arrangement  Key layout. Available layouts: ANSI, ISO, ERGO, JIS, HHKB, STAGGERED, ORTHOLINEAR
  creation_year    Creation year

options:
  -h, --help            show this help message and exit
sat@mbp-sat dsap %
```

Пример использования с командой `search_country` и `search_key_amount`.

```
{ 'название клавиатуры': 'Valor Supreme 6', 'производитель': 'nice tiger' }
sat@mbp-sat dsap % python3 cli_app.py search_country USA
Keyboards from USA:
{'название клавиатуры': 'Quantum Thunderous 57', 'производитель': 'shy tiger'}
{'название клавиатуры': 'Vortex Mighty 76', 'производитель': 'smart desert'}
{'название клавиатуры': 'Echo Dynamic 150', 'производитель': 'warm tiger'}
{'название клавиатуры': 'Havoc Relentless 5', 'производитель': 'happy bunny'}
{'название клавиатуры': 'Apex Fierce 120', 'производитель': 'shy tiger'}
{'название клавиатуры': 'Valor Agile 7', 'производитель': 'lively sun'}
{'название клавиатуры': 'Havoc Dynamic 105', 'производитель': 'warm tiger'}
{'название клавиатуры': 'Rift Resilient 128', 'производитель': 'nice tiger'}
{'название клавиатуры': 'Valor Thunderous 113', 'производитель': 'calm mountain'}
{'название клавиатуры': 'Valor Intrepid 90', 'производитель': 'warm tiger'}
{'название клавиатуры': 'Rift Dynamic 94', 'производитель': 'proud eagle'}
{'название клавиатуры': 'Havoc Intrepid 52', 'производитель': 'proud bird'}
{'название клавиатуры': 'Surge Dynamic 18', 'производитель': 'happy bunny'}
{'название клавиатуры': 'Surge Supreme 90', 'производитель': 'quick sun'}
{'название клавиатуры': 'Phantom Savage 1', 'производитель': 'happy bunny'}
{'название клавиатуры': 'Blitz Supreme 129', 'производитель': 'proud bird'}
{'название клавиатуры': 'Phantom Mighty 48', 'производитель': 'happy bunny'}
{'название клавиатуры': 'Apex Relentless 94', 'производитель': 'smart desert'}
{'название клавиатуры': 'Phantom Resilient 100', 'производитель': 'proud eagle'}
{'название клавиатуры': 'Zenith Invincible 110', 'производитель': 'lively forest'}
{'название клавиатуры': 'Apex Invincible 31', 'производитель': 'lively sun'}
{'название клавиатуры': 'Blitz Fierce 95', 'производитель': 'smart desert'}
{'название клавиатуры': 'Titan Resilient 66', 'производитель': 'warm tiger'}
{'название клавиатуры': 'Apex Thunderous 1', 'производитель': 'tall forest'}
{'название клавиатуры': 'Blitz Mighty 79', 'производитель': 'lively forest'}
{'название клавиатуры': 'Havoc Supreme 98', 'производитель': 'proud bird'}
{'название клавиатуры': 'Nexus Dynamic 85', 'производитель': 'warm tiger'}
{'название клавиатуры': 'Rift Fearless 88', 'производитель': 'quick sun'}
{'название клавиатуры': 'Valor Supreme 6', 'производитель': 'nice tiger'}
sat@mbp-sat dsap % python3 cli_app.py search_key_amount 20
Keyboards with more than 20 keys:
Quantum Thunderous 57
Vortex Mighty 76
Echo Agile 125
Valor Epic 142
Vortex Intrepid 57
Echo Supreme 100
Echo Invincible 148
Echo Dynamic 150
Rift Supreme 34
Blitz Legendary 59
```

Весь код и этот отчет доступны в репозитории:

<https://github.com/Sat1l/dsap-project>