

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

КУРСОВОЙ ПРОЕКТ

по курсу "Базы данных"

Работу

выполнила:

Карпушкина О.А.

Группа:

3530901/70202

Преподаватель:

Мяснов А.В.

Санкт-Петербург
2021

Содержание

1. Техническое задание	3
2. Ход выполнения	3
2.1. План разработки	3
2.2. Структура приложения	3
2.3. Функциональность администратора	4
2.4. Функциональность пользователя	8
Выводы	14
3. Приложения	15
3.1. Листинг main.py	15
3.2. Листинг utils.py	18

1. Техническое задание

Разработать консольное приложение, позволяющее пользователем получать определенные данные из базы. Организовать различные способы фильтрации при получении данных. Добавить роль администратора и назначить ей расширенный список функций. Приложение должно быть основано на базе данных библиотеки, созданной в течение семестра.

2. Ход выполнения

2.1. План разработки

1. Составление описания функциональности для администратора и пользователя
2. Написание соответствующих функций/структур в виде библиотеки
3. Создание скрипта для консольного взаимодействия
4. Ручное тестирование всей функциональности

2.2. Структура приложения

Приложение состоит из двух скриптов: *main.py*, *utils.py*

main.py Файл содержит код для организации консольного взаимодействия с пользователем приложения. Выполняет регистрацию, авторизацию, вывод меню (рис. 2.1), получение аргументов для выполняемых функций.

```
Welcome to musical library database client!
Menu
0 : Register
1 : Log in

Enter the action: 1
Username: db_admin
Password:
Welcome, db_admin (admin: True).
Menu
0 : Add new records (admin)
1 : Update records and custom queries (admin)
2 : Search records by name
3 : Find albums by the name and release date
4 : Find albums by the release date
5 : Artists of the label
6 : Artists of the country
7 : Find artist-album-track
100 : Help
400 : Exit
```

Рис. 2.1. Меню приложения

utils.py Файл содержит код функций, которые предлагаются для выполнения. Выполняет проверку введенных аргументов, выполнение объявленных запросов, печать результатов. Содержит в себе класс *User*, который используется для идентификации пользователя во время сессии при работе с различными функциями. Например, пользователю с привилегиями администратора можно выполнять расширенный набор функций.

2.3. Функциональность администратора

Администратор может выполнять следующие действия:

1. Добавление записей
2. Действия с записями (select, update, delete)
3. Выполнение явных запросов

Добавление записей Функция **adding_records_to_tables** позволяет добавить в таблицу запись. Она сначала выводит список всех таблиц, потом список редактируемых таблиц. Администратор вводит название таблицы, потом ему поочередно предлагается ввести значения соответствующих полей таблицы. После этого выполняется запрос на добавление.

Добавление альбома:

```
Enter the action: 0
Adding records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: dict_keys(['albums', 'genres', 'record_labels', 'singers', 'tracks'])
Table: albums
Album name: TEST NAME FROM APP
Release date (YYYY-MM-DD): 2021-01-20
```

Рис. 2.2. Вид консоли

	id	name	cover	release_date
11145	11145	ALICE GOLD	<null>	2020-08-22
11146	11146	ROBIN REPRESENTATIVES	https://api.adorable.io/avatars/256...	2020-09-02
11147	11147	Test album 2021	<null>	2020-12-30
11148	11148	Test album 2021	<null>	2020-12-30
11149	11149	TEST NAME FROM APP	<null>	2021-01-20

Рис. 2.3. Состояние таблицы с альбомами

Добавление жанра:

```
Enter the action: 0
Adding records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: dict_keys(['albums', 'genres', 'record_labels', 'singers', 'tracks'])
Table: genres
Genre name: TEST GENRE NAME 2021
```

Рис. 2.4. Вид консоли

1677	1677	christmas music
1678	1678	synthwave
1679	1679	kliningan
1680	1680	TEST GENRE NAME 2021

Рис. 2.5. Состояние таблицы с жанрами

Добавление трека:

```

Enter the action: 0
Adding records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: dict_keys(['albums', 'genres', 'record_labels', 'singers', 'tracks'])
Table: tracks
Track name: TEST TRACK 2021
Track text: BLABLA...
Text duration: 1000
Text genre: TEST GENRE NAME 2021

```

Рис. 2.6. Вид консоли

22291	22291 Test track 2021	empty...	1001	2
22292	22292 TEST TRACK 2021	BLABLA...	1000	1680

Рис. 2.7. Состояние таблицы с треками

Добавление исполнителя:

```

Enter the action: 0
Adding records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: dict_keys(['albums', 'genres', 'record_labels', 'singers', 'tracks'])
Table: singers
Singers name: TEST SINGER 2021
Singer bio: BIO.....
Singer country: Russia

```

Рис. 2.8. Вид консоли

11143	4 Test update	In 1989 the building was heavily damaged by ...	195	<null>
11144	11147 TEST SINGER 2021	BIO.....	3	<null>

Рис. 2.9. Состояние таблицы с исполнителями

Добавление лейбла:

```

Enter the action: 0
Adding records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: dict_keys(['albums', 'genres', 'record_labels', 'singers', 'tracks'])
Table: record_labels
Record label name: TEST LABEL 2021
Record label bio: BIO LABEL

```

Рис. 2.10. Вид консоли

1148	1148 TEST LABEL 2021	BIO LABEL
------	----------------------	-----------

Рис. 2.11. Состояние таблицы с лейблами

Действия с записями (select, update, delete) Функция `action_with_records` позволяет совершать различные действия с записями в таблицах (выборка, удаление, обновление). Если у записи есть какие-то связи с другими таблицами (например, ссылка на эту

запись из другой таблицы по идентификатору), то действие будет ошибочным и программа его не выполнит.

Выбор альбомов:

```
Enter the action: 1
Updating records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: ['albums', 'genres', 'record_labels', 'singers', 'tracks']
Available actions: ['select', 'update', 'delete', 'custom query']
Note: delete and update actions are available if you want to affect on unconnected records
Action: select
Table: albums
Fields: ['name', 'release_date', 'cover']
Filter field: name
Filter value:
Selected fields (using comma): name, release_date
Limit (int number or all):5
name      release_date
Visions 2012-01-31
A New Place 2 Drown      2015-12-10
6 Feet Beneath the Moon 2013-08-24
Art Angels      2015-11-06
PRODUCE JACK    2019-05-09
```

Рис. 2.12. Вид консоли

Обновление альбома:

```
Enter the action: 1
Updating records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: ['albums', 'genres', 'record_labels', 'singers', 'tracks']
Available actions: ['select', 'update', 'delete', 'custom query']
Note: delete and update actions are available if you want to affect on unconnected records
Action: update
Table: albums
Fields: ['name', 'release_date', 'cover']
Filter field: name
Filter value: TEST NAME FROM APP
Update field: name
Update value: TEST NAME FROM APP updated
Records updated
```

Рис. 2.13. Вид консоли

id	name	release_date	cover
11148	Test album 2021	<null>	2020-12-30
11149	TEST NAME FROM APP updated	<null>	2021-01-20

Рис. 2.14. Состояние таблицы альбомов

Удаление альбома:

id	name	release_date	cover
11147	Test album 2021	<null>	2020-12-30
11148	Test album 2021	<null>	2020-12-30
11149	TEST NAME FROM APP updated	<null>	2021-01-20

Рис. 2.15. Состояние таблицы до удаления

```

Enter the action: 1
Updating records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: ['albums', 'genres', 'record_labels', 'singers', 'tracks']
Available actions: ['select', 'update', 'delete', 'custom query']
Note: delete and update actions are available if you want to affect on unconnected records
Action: delete
Table: albums
Fields: ['name', 'release_date', 'cover']
Filter field: name
Filter value: TEST NAME FROM APP updated
Records deleted

```

Рис. 2.16. Вид консоли

11148	11148 Test album 2021	<null>	2020-12-30
-------	-----------------------	--------	------------

Рис. 2.17. Состояние таблицы после удаления

Выбор всех жанров:

```

Enter the action: 1
Updating records
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: ['albums', 'genres', 'record_labels', 'singers', 'tracks']
Available actions: ['select', 'update', 'delete', 'custom query']
Note: delete and update actions are available if you want to affect on unconnected records
Action: select
Table: genres
Fields: ['name']
Filter field: name
Filter value:
Selected fields (using comma): *
Limit (int number or all):100
1 Synth-pop
2 Indie-rock
3 Postpunk
4 Ambient
5 Trip hop
6 Hip-hop
7 Ethereal
8 broken beat
9 brukdown
10 crunkcore
11 lied
12 festejo
13 house music
14 romani music

```

Рис. 2.18. Вид консоли 1

```

99 outlaw country
100 tejano music
Menu
0 : Add new records (admin)
1 : Actions with records (select, update, delete) and custom queries (admin)
2 : Search records by name
3 : Find albums by the name and release date
4 : Find albums by the release date
5 : Artists of the label
6 : Artists of the country
7 : Find artist-album-track
100 : Help
400 : Exit

```

Рис. 2.19. Вид консоли 2

Выполнение явных запросов Функция `custom_query_mode` позволяет выполнить любой введенный запрос напрямую без каких-либо проверок. Это самое уязвимое место

программы, если говорить о внесении непоправимых изменений в БД в негативном смысле.

Выполнение запроса на выборку:

```
Enter the action: 1
Actions with records (select, delete, update)
List of all tables: ['albums', 'countries', 'genres', 'label_singer', 'record_labels', 'singers', 'track_list', 'tracks', 'tracks_singers', 'users']
List of editable tables: ['albums', 'genres', 'record_labels', 'singers', 'tracks']
Available actions: ['select', 'update', 'delete', 'custom query']
Note: delete and update actions are available if you want to affect on unconnected records
Action: custom query
Query (enter exit to return to the menu): select name from countries
[('Canada',), ('UK',), ('Russia',), ('Germany',), ('France',), ('Spain',), ('Italy',), ('USA',), ('Dominican Republic',), ('Liberia',), ('Andorra',),
 ('Norway',), ('Anguilla',), ('Panama',), ('Pitcairn',), ('French Polynesia',), ('Saint Vincent and the Grenadines',), ('Algeria',), ('Benin',), ('Seychelles',), ('Eswatini',), ('Cook Islands',), ('Cuba',), ('Djibouti',), ('Aruba',), ('Ukraine',), ('Northern Mariana Islands',), ('Christmas Island',), ('Leone',), ('Fiji',), ('Seychelles',), ('New Caledonia',), ('Thailand',), ('Lesotho',), ('Lithuania',), ('Nepal',), ('Korea (Democratic People's Republic',), ('Cabo Verde',), ('Ghana',), ('Malta',), ('Saint Pierre and Miquelon',), ('Uganda',), ('Maldives',), ('Faroe Islands',), ('Bulgaria',), ('Bahamas',), ('Kazakhstan',), ('Tonga',), ('Guyana',), ('Guinea-Bissau',), ('New Zealand',), ('Egypt',), ('Bosnia and Herzegovina',), ('Antigua and Barbuda',), ('Martinique',), ('Montenegro',), ('Turks and Caicos Islands',), ('Jordan',), ('Eritrea',), ('Gibraltar',), ('Libya',), ('Mauritius',), ('Costa Rica',), ('Iraq',), ('Turkey',), ('Indonesia',), ('Marshall Islands',), ('Iceland',), ('Israel',), ('Portugal',), ('Cameroon',), ('Svalbard and Jan Mayen',), ('Turkmenistan',), ('Venezuela (Bolivarian Republic of)',), ('Zimbabwe',), ('Heard Island and McDonald Islands',), ('Mayotte',), ('Mexico',), ('Denmark',), ('Mongolia',), ('South Africa',), ('Finland',), ('Cocos (Keeling) Islands',), ('Mozambique',), ('Jamaica',), ('Niger',), ('Ukraine',), ('Micronesia (Federated States of)',), ('El Salvador',), ('Belarus',), ('Côte d'Ivoire',), ('Solomon Islands',), ('Central African Republic',), ('Great Britain and Northern Ireland',), ('Holy See',), ('Australia',), ('Bahrain',), ('Kyrgyzstan',), ('Netherlands',), ('Liechtenstein',), ('Malawi',)
```

Рис. 2.20. Вид консоли

2.4. Функциональность пользователя

Пользователь может выполнять следующие действия:

1. Поиск любых записей по имени
2. Поиск альбома по имени и дате релиза
3. Поиск альбомов за определенный промежуток времени
4. Поиск информации о сотрудничестве типа артист-лейбл
5. Вывод артистов заданной страны
6. Поиск по артисту/альбому/треку

Поиск любых записей по имени Эта функция позволяет в заданной таблице найти соответствие по имени. Для этого используется функция `find_records_by_name`. В нее передается объект курсора, имя таблицы для поиска и строка соответствия. Для выделения соответствия используется конструкция *like*. Благодаря ей можно искать совпадения в имени вне зависимости от регистра введенных данных. Поиск соответствия ищется во всех строках (разрешено любое количество символов до и после совпадения).

Поиск альбомов:

```
Enter the action: 2
Who or what do you want to find?
Available tables: ['album', 'singer', 'genre', 'record label', 'track']
albums
Please, try again
Available tables: ['album', 'singer', 'genre', 'record label', 'track']
album
Name: cool
Founded items:
COMMENT COOL          2020-08-15
SPENT COOL             2020-11-16
SHADOW COOL            2020-07-11
```

Рис. 2.21. Вид консоли

Поиск жанров:

```
Enter the action: 2
Who or what do you want to find?
Available tables: ['album', 'singer', 'genre', 'record label', 'track']
genre
Name: bel
Founded items:
below
irish rebel music
bel canto
```

Рис. 2.22. Вид консоли

Поиск альбома по имени и дате релиза Эта функция позволяет найти альбом по году выпуска и именному совпадению. Для этого пользователь вводит строку, которая будет искаться в имени и год выпуска. В ответ на это функция `find_record_by_name_release_date` выдает сначала запись об альбоме, потом список исполнителей альбома и его трек лист.

```
Enter the action: 3
What album do you want to find?
Name: cool
Release date (YYYY-MM-DD): 2020-08-15
Founded albums:
COMMENT COOL      No cover      2020-08-15
Artists:
Donn Mayo
Track list (name, duration in secs):
exercises passing      5268
..
```

Рис. 2.23. Пример поиска сингла

```

Enter the action: 3
What album do you want to find?
Name: Miss Anthropocene
Release date (YYYY-MM-DD): 2020-02-21
Founded albums:
Miss Anthropocene + test          No cover          2020-02-21
Artists:
Vs
Dance
Devel
Evelynn Mcdowell
Grimes
Msn Header
Substances Founded
Wine Lawrence
Grants Popular
Exciting Mystery
Silent Lincoln
Contrast
Am
Horses Di
Track list (name, duration in secs):
roulette si                        1
4ÆM                               195
construction saved                1
So Heavy I Fell Through The Earth 368
New Gods                         195
You'll miss me when I'm not around 161
manage represents                 1
IDORU                            432
My Name Is Dark                   356
Before the fever                  217
Violence                         220
Darkseid                         224
Delete Forever                    237

```

Рис. 2.24. Пример поиска сборника

Поиск альбомов за определенный промежуток времени Данная функция выполняет ровно то, что указано в названии. Пользователь вводит дату в указанном формате и она далее передается в функцию `find_albums_by_release_date`, где происходит запрос и печать требуемой информации. Также эта функция подразумевает возможность ввода пустой конечной даты. Это означает, что нужно найти все альбомы, которые выпущены с указанной даты по настоящее время.

Примеры на рис. 2.25 и на рис. 2.26.

```

Enter the action: 4
What from-release-date of album do you want to find (YYYY-MM-DD)?
2020-12-30
What to-release-date of album do you want to find (YYYY-MM-DD or nothing)?

Albums (name, release date):
ASSOCIATE TOWN          2020-12-31
BLOCKS PROBLEM          2020-12-31
COVERED DON             2020-12-31
DURATION UNIVERSITY     2020-12-31
EXPERIENCES AUCTIONS    2020-12-31
JACKSONVILLE WINGS     2020-12-31
JOURNEY CHARGER         2020-12-31
KILL AGAINST            2020-12-31
MICHAEL PP              2020-12-31
NOBLE STATIONS          2020-12-31
ORDERED PREFER          2020-12-31
PANTYHOSE PERSPECTIVE   2020-12-31
RELIGIOUS ELIMINATE     2020-12-31
SUBSEQUENT JACKET       2020-12-31
TRIPLE ORACLE           2020-12-31

```

Рис. 2.25. Поиск с заданной даты по настоящее время

```

Enter the action: 4
What from-release-date of album do you want to find (YYYY-MM-DD)?
2012-01-01
What to-release-date of album do you want to find (YYYY-MM-DD or nothing)?
2015-01-01
Albums (name, release date):
IP WORKING                2012-01-04
Visions                   2012-01-31
COLLECTIONS RAM           2012-07-28
GOURMET SUBMISSION        2013-04-09
TURNED IMAGES             2013-07-31
UNIVERSAL RECEPTION       2013-08-23
6 Feet Beneath the Moon   2013-08-24
BEDS INGREDIENTS         2013-09-12
UKRAINE TAUGHT           2014-06-17
WAIT DAD                  2014-07-06
ABC INVESTIGATION         2014-09-18

```

Рис. 2.26. Поиск с заданных временных рамок

Поиск информации о сотрудничестве типа артист-лейбл Указанная функция позволяет получить информацию о всех контрактах, которые были у лейбла в указанный промежуток времени. Данная функция также позволяет ввод пустой даты окончания.

Примеры на рис. 2.27 и на рис. 2.28.

```

Enter the action: 5
What record label are you looking for?
4AD
What from-collab-date of collaboration with record label do you want to find (YYYY-MM-DD)?
2000-01-01
What to-collab-date of collaboration with record label do you want to find (YYYY-MM-DD)?

Founded records (record label, artists names, date from):
4AD Records      Grimes          2015-11-06
4AD Records      Sun Victoria    2017-02-03
4AD Records      Participants     2017-03-03
4AD Records      Moves Labels    2017-03-07
4AD Records      Amsterdam Section 2017-03-20
4AD Records      Port            2017-04-27
4AD Records      Detroit         2017-05-15
4AD Records      Pulse           2017-06-02
4AD Records      Chemical        2017-12-26
4AD Records      Sessions        2018-03-27
4AD Records      Misc Traffic    2018-08-04
4AD Records      Creation Senator 2018-11-07
4AD Records      Minister        2018-12-12
4AD Records      Millard Sparks  2019-10-04
4AD Records      Restaurant No   2019-12-17

```

Рис. 2.27. Поиск с заданной даты по настоящее время

```

Enter the action: 5
What record label are you looking for?
Arbutus
What from-collab-date of collaboration with record label do you want to find (YYYY-MM-DD)?
2000-01-01
What to-collab-date of collaboration with record label do you want to find (YYYY-MM-DD)?
2020-01-01
Founded records (record label, artists names, date from, date_to):
Arbutus      Grimes          2010-01-10      2015-11-06
Arbutus      Finally Injection 2017-07-10      2018-09-15
Arbutus      Youlanda Blanchard 2018-04-07      2018-06-11
Arbutus      Generally       2018-08-27      2019-01-14
..

```

Рис. 2.28. Поиск с заданных временных рамок

Вывод артистов заданной страны Данная функция дает ввести пользователю название страны, а после выводит на экран всех артистов, у которых указана введенная страна.

Примеры на рис. 2.29.

```
Enter the action: 6
What country are interested in?
Canada
Founded artists (name, bio, country):
Grimes      None      Canada
Appeared in  Messages can be sent to and received from ports, but these messages must obey the so-called "port protocol." Erlang is a general-purpose
programming language.      Canada
Clothes Views  Its main implementation is the Glasgow Haskell Compiler.      Canada
Thread  Make me a sandwich. Ports are used to communicate with the external world. Ports are used to communicate with the external world.
Confirmation  He looked inquisitively at his keyboard and wrote another sentence. It is also a garbage-collected runtime system.      Canada
Hans Garcia   Type classes first appeared in the Haskell programming language.      Canada
```

Рис. 2.29. Поиск исполнителей заданной страны

Поиск по артисту/альбому/треку Данная функция позволяет выполнять следующие запросы по поиску:

1. Поиск по артисту: позволяет найти все треки заданного артиста или артиста с частичным совпадением имени, также указываются название альбома и дата выпуска.
2. Поиск по альбому: позволяет найти все альбомы совпадающие с введенным названием, все треки входящие в альбом, а также предоставить информацию об исполнителях
3. Поиск по треку: позволяет найти всех исполнителей и все альбомы, куда входит композиция.

Примеры на рис. 2.30, 2.31, 2.32, 2.33.

```
Enter the action: 7
What do you want to find? [singer, album, track]
singer
Enter the name: grimes
Founded records (artist's name, album, track, release date):
Grimes      ADVANCE PROTEINS      inquiry mixed      2020-12-07
Grimes      ALPHA HEARTS      My Name Is Dark      2019-11-28
Grimes      ALTERNATIVES MOOD      customs usd      2019-08-29
Grimes      BENEFIT UPON      slot registration      2020-10-15
Grimes      BIBLE INSTALLING      job chronic      2017-09-24
Grimes      BID ILL      job chronic      2018-07-29
Grimes      BIRDS PRESENTATION      replaced circuit      2020-11-20
Grimes      BRIEF HAPPENS      Before the fever      2018-04-01
Grimes      CABLE CAT      customs usd      2017-11-16
Grimes      CHANGES CITYSEARCH      ian probably      2017-09-14
Grimes      CHOOSING ISSUES      inquiry mixed      2020-05-10
Grimes      CHRIST ND      journal writing      2020-04-16
Grimes      CONDITION DOG      inquiry mixed      2020-02-17
Grimes      CORPORATE EXAMINATION      customs usd      2019-01-07
Grimes      DELHI PROVINCE      replaced circuit      2020-02-17
Grimes      DUST RISKS      New Gods      2017-03-23
Grimes      EXPECTED RAISING      inquiry mixed      2020-10-04
Grimes      FASTER APPAREL      customs usd      2017-06-11
Grimes      GAMBLING WHOSE      collectables marshall      2017-03-16
```

Рис. 2.30. Поиск для имени исполнителя

```

Enter the action: 7
What do you want to find? [singer, album, track]
album
Enter the name: Visions
Founded records (album's name, artist, track, release date):
ATTORNEY PROVISIONS      Affect Literature      bankruptcy graph      2018-03-24
ATTORNEY PROVISIONS      Anjelica O'neal      constantly directly      2018-03-24
ATTORNEY PROVISIONS      Ar      sit estimated      2018-03-24
ATTORNEY PROVISIONS      Diesel      bankruptcy graph      2018-03-24
ATTORNEY PROVISIONS      Estimate Yoga      bankruptcy graph      2018-03-24
ATTORNEY PROVISIONS      Garnett Ramos      bankruptcy graph      2018-03-24
ATTORNEY PROVISIONS      Hampshire Bars      cpu positive      2018-03-24
ATTORNEY PROVISIONS      Hong Covering      constantly directly      2018-03-24
ATTORNEY PROVISIONS      Ins Evening      sit estimated      2018-03-24
ATTORNEY PROVISIONS      Ratio      bankruptcy graph      2018-03-24
ATTORNEY PROVISIONS      Richmond      bankruptcy graph      2018-03-24
ATTORNEY PROVISIONS      Signal      constantly directly      2018-03-24
ATTORNEY PROVISIONS      Val Flower      bankruptcy graph      2018-03-24
EAST PROVISIONS      About Britney      delivery beautiful      2020-09-01
EAST PROVISIONS      Const      delivery beautiful      2020-09-01
EAST PROVISIONS      Me      ag parts      2020-09-01
EAST PROVISIONS      Quickly      dell ab      2020-09-01
EAST PROVISIONS      Yael Bolton      fucked semester      2020-09-01
PROVISIONS FAMILIAR      Charis Marks      plugin update      2020-08-08
PROVISIONS FAMILIAR      Editor      plugin update      2020-08-08
PROVISIONS FAMILIAR      Erik Burris      eyes carry      2020-08-08
PROVISIONS FAMILIAR      Initially      website automobile      2020-08-08
PROVISIONS FAMILIAR      Kill Terry      website automobile      2020-08-08
PROVISIONS FAMILIAR      Monitoring      seattle machine      2020-08-08
PROVISIONS FAMILIAR      Needs      responsibilities act      2020-08-08
PROVISIONS FAMILIAR      Sunset Somewhat      plugin update      2020-08-08
PROVISIONS FAMILIAR      Tennis      sh notebooks      2020-08-08
PROVISIONS FAMILIAR      Trouble      eyes carry      2020-08-08
PROVISIONS HIV      Concentration Smart      ideal oldest      2020-05-14
PROVISIONS HIV      Conservation Flexible      ideal oldest      2020-05-14
PROVISIONS HIV      Definition      vacuum architecture      2020-05-14

```

Рис. 2.31. Поиск для названия альбома

```

Enter the action: 7
What do you want to find? [singer, album, track]
track
Enter the name: Infinite ♡ Without Fulfillment
Founded records (track's name, artist, album, release date):
Infinite ♡ Without Fulfillment      Analyst      Visions      2012-01-31
Infinite ♡ Without Fulfillment      Grimes      Visions      2012-01-31

```

Рис. 2.32. Поиск названия трека

```

Enter the action: 7
What do you want to find? [singer, album, track]
album
Enter the name: Miss
Founded records (album's name, artist, track, release date):
ADMISSION BALANCE      Accept Had      interactive thus      2020-02-15
ADMISSION BALANCE      Bureau Cultures  interactive thus      2020-02-15
ADMISSION BALANCE      Jonas Harrison  interactive thus      2020-02-15
ADMISSION BALANCE      Paris Simon     interactive thus      2020-02-15
ADMISSION BALANCE      Raise Derived   interactive thus      2020-02-15
ADMISSION IRISH         Bookmark        create chinese        2020-08-18
ADMISSION IRISH         Constitutes Twiki estimate student      2020-08-18
ADMISSION IRISH         Deeann Ratliff  create chinese        2020-08-18
ADMISSION IRISH         Values Platforms estimate student      2020-08-18
ARTWORK MISSION         Assistance      arrangement happy      2020-01-05
ARTWORK MISSION         Devices         at valentine          2020-01-05
ARTWORK MISSION         Donn Mejia      careers master         2020-01-05
ARTWORK MISSION         Gbp             arrangement happy      2020-01-05
ARTWORK MISSION         Relative        arrangement happy      2020-01-05
ARTWORK MISSION         Wall Moscow     stack python           2020-01-05
ASSUMES COMMISSION      Keeley Harrell  amateur join           2020-02-10
BLOGGING ADMISSION      Both Cultures   awarded blogs          2020-07-20
BLOGGING ADMISSION      Cool Alternate  dispatched total       2020-07-20
BLOGGING ADMISSION      Harlan Velazquez dispatched total       2020-07-20
BLOGGING ADMISSION      Header Receive  awarded blogs          2020-07-20
BLOGGING ADMISSION      Isreal Wiggins  babies stories         2020-07-20
BLOGGING ADMISSION      Jacksonville Carefully roads mention          2020-07-20
BLOGGING ADMISSION      Matt            dispatched total       2020-07-20
BLOGGING ADMISSION      Moreover        dispatched total       2020-07-20
BLOGGING ADMISSION      Obtain          roads mention          2020-07-20
BLOGGING ADMISSION      Opened Physicians awarded blogs          2020-07-20
BLOGGING ADMISSION      Pink Partners   awarded blogs          2020-07-20
BRISTOL MISSED          Alphonse Stephens prepare thesis         2018-03-11

```

Рис. 2.33. Поиск для частичного названия

Выводы

В ходе выполнения курсового проекта было создано консольное приложение для работы с БД музыкальной библиотеки. Данное приложение организует взаимодействие с библиотекой для двух видов ролей: администратор и пользователь. Администратор имеет расширенную функциональность: он имеет доступ к самим записям в БД, может их добавлять, изменять, удалять. Также есть функция выполнения любых введенных запросов, что позволит администратору производить любые действия, но изначально закладывалась идея добавления/удаления индексов и выполнения запросов **EXPLAIN ANALYSE**.

Функциональность пользователя библиотеки позволяет выполнять ему различные действия по получению данных. Были добавлены функции, которые показали наиболее интересными потенциальному пользователю программы. Созданная архитектура позволяет достаточно быстро в случае необходимости добавить новые функции.

Для заявленных функций было проведено ручное тестирование, результаты которого были представлены в отчете. Это оказалось довольно трудоемкой и однообразной задачей, но альтернативного способа решения вопроса тестирования не было найдено.

Файл *utils.py* подразумевался как библиотечный файл. На данный момент он может использоваться при написании другого консольного приложения. При добавлении небольших изменений в методы, которые исполняют определенный запрос, а если конкретно, то возвращать полученные из БД данные в каком-либо формате, то библиотеку можно будет использовать при написании других приложений, где уместно использование явного обращения к базе данных с помощью `psycopg2`.

3. Приложения

3.1. Листинг main.py

```
1 import utils
2 import sys
3 from getpass import getpass
4
5 con, cursor = utils.create_connection()
6
7 first_level_functions = {
8     0: 'Register',
9     1: 'Log_in'
10 }
11
12
13 all_functions = {
14     0: 'Add_new_records_(admin)',
15     1: 'Actions_with_records_(select, _update, _delete) _and _custom _queries_(admin)
16     ↪ ',
17     2: 'Search_records_by_name',
18     3: 'Find_albums_by_the_name_and_release_date',
19     4: 'Find_albums_by_the_release_date',
20     5: 'Artists_of_the_label',
21     6: 'Artists_of_the_country',
22     7: 'Find_artist-album-track',
23     100: 'Help',
24     400: 'Exit'
25 }
26
27 print_help = {
28     'Search_records_by_name': "This_option_helps_you_to_find_any_record_in_the_
29     ↪ music_library_\n",
30     'Find_albums_by_the_name_and_release_date': "This_option_helps_you_to_find_
31     ↪ THE_ALBUM_that_you_are_looking_for_using"
32     " _name_and_release_date_to_make_
33     ↪ no_mistakes_\n",
34     'Find_albums_by_the_release_date': "This_option_helps_you_to_find_all_albums
35     ↪ _between_entered_dates_\n",
36     'Artists_of_the_label': "This_option_helps_you_to_find_everything_about_
37     ↪ artist-label_collaborations_using_record_"
38     "label_name_and_dates_of_partnership_\n",
39     'Artists_of_the_country': "This_option_helps_you_to_find_every_artist_of_the
40     ↪ _country_that_you_entered_\n",
41     'Find_artist-album-track': "This_option_helps_you_to_find_every_artist/album
42     ↪ /track_and_everything_connected_with_"
43     "it_\n",
44     'Exit': "This_option_helps_you_to_get_out_of_this_little_library-program",
45 }
46
47
48 def session(user):
49     while True:
50         print('Menu')
51         for action_fun in all_functions:
52             print('{}:{}'.format(action_fun, all_functions[action_fun]))
53         session_action = None
54         while session_action is None:
```

```

48         try:
49             print('\nEnter the action: ', end='')
50             session_action = int(input())
51         except ValueError:
52             session_action = None
53             print('You entered wrong value, try again')
54
55     try:
56         if session_action == 0:
57             if user.admin is True:
58                 utils.adding_records_to_tables(cursor)
59             else:
60                 print('You have got no permission to perform this action')
61         elif session_action == 1:
62             if user.admin is True:
63                 utils.action_with_records(cursor)
64             else:
65                 print('You have got no permission to perform this action')
66         elif session_action == 2:
67             print('Who or what do you want to find?')
68             find_actions = ['album', 'singer', 'genre', 'record_label', '
↪ track']
69
70             print('Available tables: ', find_actions)
71             find_action = input()
72             while find_action not in find_actions:
73                 print('Please, try again')
74                 print('Available tables: ', find_actions)
75                 find_action = input()
76
77             print('Name: ', end='')
78             find_name = input()
79             utils.find_records_by_name(cursor, find_name, find_action)
80         elif session_action == 3:
81             print('What album do you want to find?')
82             print('Name: ', end='')
83             album_name = input()
84             print('Release date (YYYY-MM-DD): ', end='')
85             album_release_date = input()
86             utils.find_record_by_name_release_date(cursor, album_name,
↪ album_release_date)
87         elif session_action == 4:
88             print('What from-release-date of album do you want to find (YYYY-
↪ -MM-DD)?')
89             from_date = input()
90             print('What to-release-date of album do you want to find (YYYY-
↪ MM-DD or nothing)?')
91             to_date = input()
92             utils.find_albums_by_release_date(cursor, from_date, to_date)
93         elif session_action == 5:
94             print('What record_label are you looking for?')
95             label_name = input()
96             print('What from-collab-date of collaboration with record_label
↪ do you want to find (YYYY-MM-DD)?')
97             from_date = input()
98             print('What to-collab-date of collaboration with record_label do
↪ you want to find (YYYY-MM-DD)?')
99             to_date = input()
100            utils.record_label_partnership(cursor, label_name, from_date,
↪ to_date)

```



```

101         elif session_action == 6:
102             print('What_country_are_interested_in?')
103             country_name = input()
104             utils.find_singers_by_country(cursor, country_name)
105         elif session_action == 7:
106             print('What_do_you_want_to_find?_[singer,album,track]')
107             option = input()
108             print('Enter_the_name:', end='')
109             name = input()
110             utils.find_singer_album_track(cursor, name, option[0])
111         elif session_action == 100:
112             for item in print_help:
113                 print(item, end='\t-->\t')
114                 print(print_help[item])
115         elif session_action == 400:
116             print('Goodbye!')
117             sys.exit(0)
118         else:
119             print('Wrong_menu_item._Try_again')
120
121     except (ValueError, KeyError) as err:
122         print('Please,_try_again:{}`.format(err))
123
124
125 def login():
126     result = False
127     while not result:
128         print('Username:', end='')
129         username = input()
130         password = getpass()
131
132         try:
133             user_obj = utils.check_user(cursor, username, password)
134             result = True
135         except (ValueError, KeyError):
136             print('Log_in_failed._Try_to_log_in_again')
137             result = False
138     session(user_obj)
139
140
141 def registration():
142     print('Welcome_to_the_registration_form!_Please,_enter_your_username')
143     print('Username:', end='')
144
145     username = input()
146     while len(username) > 100:
147         print('Your_username_is_too_long,_try_to_use_less_characters')
148         print('Username:', end='')
149
150         username = input()
151
152     print('Password:', end='')
153     password = getpass()
154     try:
155         utils.registration(con, cursor, (username, password, False))
156     except EnvironmentError:
157         print('Registration_failed.')
158
159
160 if __name__ == '__main__':

```

```

161     print('Welcome_to_musical_library_database_client!')
162     print('Menu')
163     for a in first_level_functions:
164         print('{{}_:_{}}'.format(a, first_level_functions[a]))
165     action = None
166     while action is None:
167         try:
168             print('\nEnter_the_action:_', end='')
169             action = int(input())
170         except ValueError:
171             action = None
172             print('You_entered_wrong_value,_try_again')
173
174     if action == 0:
175         registration()
176     elif action == 1:
177         login()

```

3.2. Листинг utils.py

```

1 import psycopg2
2
3 tables = ['albums', 'countries', 'genres', 'label_singer', 'record_labels', '
    ↪ singers', 'track_list', 'tracks',
4         'tracks_singers', 'users']
5
6 editable_tables = {'albums': ['name', 'release_date', 'cover'],
7                     'genres': ['name'],
8                     'record_labels': ['name', 'bio'],
9                     'singers': ['name', 'bio', 'countries_id', 'photo'],
10                    'tracks': ['name', 'text', 'duration', 'genres_id']}
11
12 actions = ['select', 'update', 'delete', 'custom_query']
13
14
15 class User:
16
17     def __init__(self, username, admin=False):
18         self.username = username
19         self.admin = admin
20
21     def __str__(self):
22         return '{{}_({admin:_{}})'.format(self.username, self.admin)
23
24
25 def create_connection():
26     f = open('data/db_info.txt', 'r')
27     conn_params = dict()
28
29     for r in f:
30         pair_dict = r.split('\n')[0]
31         conn_params[pair_dict.split(':')[0]] = pair_dict.split(':')[1]
32
33     schema = conn_params['schema']
34
35     con = psycopg2.connect(
36         host=conn_params['host'],
37         database=conn_params['database'],
38         user=conn_params['user'],

```

```

39         password=conn_params['password'],
40         options=f'-c_search_path={schema}'
41     )
42     con.autocommit = True
43
44     cursor = con.cursor()
45     return con, cursor
46
47
48 def registration(con, cursor, data):
49     try:
50         cursor.execute("insert_into_users_(username,_password,_admin_privileges)
↪ _values_(%s,%s,%s)",
51                        data)
52
53     except psycopg2.IntegrityError as e:
54         print('Error_in_db._Maybe_user_with_entered_username_is_already_exists.'
↪ , e.pgerror)
55         con.commit()
56
57         cursor.execute("select_username_from_users_where_username_=_%s", (data[0],))
58
59         respond = cursor.fetchall()
60         if respond[0][0] == data[0]:
61             return True
62         else:
63             raise EnvironmentError('Your_account_was_not_added._Try_again_later')
64
65
66 def check_user(cursor, username, password):
67     cursor.execute('select_username,_admin_privileges,_password_from_users_
↪ where_username_=_%s_', (username,))
68
69     user = cursor.fetchall()
70
71     if len(user) != 1:
72         raise KeyError('You_are_not_registered')
73     if password != user[0][-1]:
74         print('Wrong_password,_try_again')
75         raise ValueError('Your_password_was_incorrect')
76     else:
77         user_obj = User(username, admin=user[0][1])
78         print('Welcome,_{}'.format(user_obj))
79         return user_obj
80
81
82 def adding_records_to_tables(cursor):
83     print('Adding_records')
84     print('List_of_all_tables:{}'.format.tables))
85     print('List_of_editable_tables:{}'.format(editable_tables.keys()))
86
87     print('Table:', end='')
88     table = input()
89     try:
90         if table == list(editable_tables.keys())[0]:
91             print('Album_name:', end='')
92             album_name = input()
93
94             print('Release_date_(YYYY-MM-DD):', end='')
95             release_date = input()

```

```

96
97     cursor.execute("insert_into_albums_(name,_release_date)_values_(%s,_
↪ %s)", (album_name, release_date))
98
99     elif table == list(editable_tables.keys())[1]:
100         print('Genre_name:_', end='')
101         genre_name = input()
102
103     cursor.execute("insert_into_genres_(name)_values_(%s)", (genre_name
↪ ,))
104
105     elif table == list(editable_tables.keys())[2]:
106         print('Record_label_name:_', end='')
107         record_label_name = input()
108
109         print('Record_label_bio:_', end='')
110         record_label_bio = input()
111
112     cursor.execute("insert_into_record_labels_(name,_bio)_values_(%s,_%s
↪ ),",
113                    (record_label_name, record_label_bio))
114
115     elif table == list(editable_tables.keys())[3]:
116         print('Singers_name:_', end='')
117         singer_name = input()
118
119         print('Singer_bio:_', end='')
120         singer_bio = input()
121
122         print('Singer_country:_', end='')
123         singer_country = input()
124
125     cursor.execute("select_id_from_countries_where_name_=_%s", (
↪ singer_country,))
126     respond = cursor.fetchall()
127     singer_country_id = respond[0][0]
128
129     cursor.execute("insert_into_singers_(name,_bio,_countries_id)_values
↪ _(%s,_%s,_%s)",
130                    (singer_name, singer_bio, int(singer_country_id)))
131
132     elif table == list(editable_tables.keys())[4]:
133         print('Track_name:_', end='')
134         track_name = input()
135
136         print('Track_text:_', end='')
137         track_text = input()
138
139         print('Text_duration:_', end='')
140         track_duration = int(input())
141
142         print('Text_genre:_', end='')
143         track_genre = input()
144
145     cursor.execute("select_id_from_genres_where_name_=_%s", (track_genre
↪ ,))
146     respond = cursor.fetchall()
147     track_genre_id = respond[0][0]
148

```

```

149         cursor.execute("insert_into_tracks_(name,_text,_duration,_genres_id)
↪ _values_(%s,%s,%s,%s)",
150                        (track_name, track_text, track_duration,
↪ track_genre_id))
151     else:
152         raise ValueError('You_entered_wrong_table')
153     except psycopg2.DataError:
154         raise ValueError('You_entered_wrong_value!!!!!!')
155
156
157 def custom_query_mode(cursor):
158     while True:
159         try:
160             print('Query_(enter_exit_to_return_to_the_menu):_', end='')
161             query = input()
162             if query == 'exit':
163                 break
164             cursor.execute(query)
165             respond = cursor.fetchall()
166             print(respond)
167         except psycopg2.DataError:
168             print('You_entered_wrong_value!_Try_again_')
169         except psycopg2.DatabaseError:
170             print('There_was_database_error._You_have_entered_something_wrong. ')
171
172
173 def check_selected(selected, editable_fields):
174     if selected == '*':
175         return True
176     selected = selected.split(',')
177     selected_field_list = list(map(str.strip, selected))
178     bool_fields = [x in editable_fields for x in selected_field_list]
179     return len(selected_field_list) == sum(bool_fields)
180
181
182 def action_with_records(cursor):
183     print('Actions_with_records_(select,_delete,_update)')
184     print('List_of_all_tables:_{ }'.format(tables))
185     print('List_of_editable_tables:_{ }'.format(list(editable_tables.keys())))
186     print('Available_actions:_{ }'.format(actions))
187     print('Note:_delete_and_update_actions_are_available_if_you_want_to_affect_
↪ on_unconnected_records')
188
189     print('Action:_', end='')
190     action = input()
191     if action not in actions:
192         raise KeyError('You_entered_wrong_action')
193     elif action == actions[3]:
194         custom_query_mode(cursor)
195     else:
196         print('Table:_', end='')
197         table = input()
198         while table not in tables:
199             print('Please,_try_again_')
200             print('Table:_', end='')
201             table = input()
202
203         print('Fields:_', editable_tables[table])
204
205         print('Filter_field:_', end='')

```

```

206     filter_field = input()
207     while filter_field not in editable_tables[table]:
208         print('Please, _try_again')
209         print('Filter_field:_', end='')
210         filter_field = input()
211
212     print('Filter_value:_', end='')
213     filter_value = input()
214     try:
215         if action == actions[1]:
216             print('Update_field:_', end='')
217             update_field = input()
218             while update_field not in editable_tables[table]:
219                 print('Please, _try_again')
220                 print('Update_field:_', end='')
221                 update_field = input()
222
223             print('Update_value:_', end='')
224             update_value = input()
225
226             request = 'update_{}_set_{}_=_{}_where_{}_=_{}'.format(table,
↪ update_field, '%s', filter_field, '%s')
227             cursor.execute(request, (update_value, filter_value))
228             print('Records_updated')
229
230             elif action == actions[2]:
231                 request = 'delete_from_{}_where_{}_=_{}'.format(table,
↪ filter_field, '%s')
232                 cursor.execute(request, (filter_value,))
233                 print('Records_deleted')
234
235             elif action == actions[0]:
236                 print('Selected_fields_(using_comma):_', end='')
237                 selected_field = input()
238
239                 while check_selected(selected_field, editable_tables[table]) is
↪ not True:
240                     print('Please, _try_again')
241                     print('Selected_fields_(using_comma):_', end='')
242                     selected_field = input()
243
244                 print('Limit_(int_number_or_all):', end='')
245                 limit = int(input())
246                 limit_str = '_limit_' + str(limit)
247                 if filter_value == '':
248                     request = 'select_{}_from_{}'.format(selected_field, table)
249                     if limit:
250                         request += limit_str
251
252                     cursor.execute(request)
253
254                 else:
255                     request = 'select_{}_from_{}_where_{}_=_%s_limit_%s'.format(
↪ selected_field, table, filter_field,
256
↪ filter_value)
257                     if limit:
258                         request += limit_str
259
260                     cursor.execute(request, (filter_value,))

```

```

261         respond = cursor.fetchall()
262         if selected_field != '*':
263             selected_fields = selected_field.split(',')
264             for s in selected_fields:
265                 print(s, end='\t')
266             print()
267         for r in respond:
268             for field in r:
269                 print(field, end='\t')
270             print()
271
272     except psycopg2.DataError:
273         print('You entered wrong value! Try again')
274     except psycopg2.DatabaseError:
275         print('There was database error. You have entered something wrong.')
276
277
278 def print_respond(respond):
279     for r in respond:
280         if len(r) == 1:
281             print(r[0])
282         elif len(r) == 2:
283             print(r[0], r[1], sep='\t\t')
284         elif len(r) == 3:
285             print(r[0], r[1], r[2], sep='\t\t')
286         elif len(r) == 4:
287             print(r[0], r[1], r[2], r[3], sep='\t\t')
288
289
290 def find_record_by_name_release_date(cursor, album_name, album_release_date):
291     album_name = '%' + album_name + '%'
292     sql_album = "select name, cover, release_date from albums where name ilike %s"
293     ↪ s_and_release_date = %s"
294     try:
295         cursor.execute(sql_album, (album_name, album_release_date))
296     except psycopg2.DataError:
297         raise ValueError('You entered wrong data value')
298     except psycopg2.DatabaseError:
299         raise ValueError('There was database error. You have entered something')
300     ↪ wrong.')
301     album_respond = cursor.fetchall()
302     albums = []
303     album = dict()
304     print('Founded albums:')
305     for x in album_respond:
306         album['name'] = x[0]
307         print(album['name'], end='\t')
308
309         if x[1] is None:
310             album['cover'] = 'No cover'
311         else:
312             album['cover'] = x[1]
313             print(album['cover'], end='\t')
314
315         album['release_date'] = x[2]
316         print(album['release_date'], end='\t')
317         albums.append(album)
318
319     sql_tracks = "select t.name, t.duration, s.name from track_list " \

```

```

319         "join_albums_a_on_track_list.album_id=_a.id_" \
320         "join_tracks_t_on_track_list.tracks_id=_t.id_" \
321         "join_tracks_singers_ts_on_t.id=_ts.tracks_id_" \
322         "join_singers_s_on_ts.singers_id=_s.id_" \
323         "where_a.name_ilike_%s_and_a.release_date=_%s_" \
324         "order_by_a.release_date;"
325
326     cursor.execute(sql_tracks, (album_name, album_release_date))
327     track_list_respond = cursor.fetchall()
328     artists = set()
329     tracks = set()
330     for x in track_list_respond:
331         artists.add((x[2],))
332         tracks.add((x[0], x[1]))
333
334     print('\nArtists:_')
335     print_respond(artists)
336     print('Track_list_(name,_duration_in_secs):_')
337     print_respond(tracks)
338
339
340 def find_records_by_name(cursor, name, find_action):
341     name = '%' + name + '%'
342
343     sql_albums = "select_name,_release_date_from_albums_" \
344                 "where_name_ilike_%s_"
345
346     sql_singers_name = "select_singers.name,_singers.bio,_singers.photo,_c.name_"
347     ↪ from_singers_" \
348                 "join_countries_c_on_c.id=_singers.countries_id_" \
349                 "where_singers.name_ilike_%s_"
350
351     sql_genres = "select_name_from_genres_" \
352                 "where_name_ilike_%s_"
353
354     sql_record_labels = "select_name_from_record_labels_" \
355                         "where_name_ilike_%s_"
356
357     sql_tracks = "select_name,_duration_from_tracks_" \
358                 "where_name_ilike_%s_"
359
360     action_table = {'album': sql_albums, 'singer': sql_singers_name,
361                    'genre': sql_genres, 'record_label': sql_record_labels,
362                    'track': sql_tracks}
363
364     try:
365         cursor.execute(action_table[find_action], (name,))
366     except psycopg2.DataError:
367         raise ValueError('You_entered_wrong_value!_Try_again')
368     except psycopg2.DatabaseError:
369         raise ValueError('There_was_database_error._You_have_entered_something_'
370 ↪ wrong.')
371     respond = cursor.fetchall()
372
373     print('Founded_items:_')
374     print_respond(respond)
375
376
377 def find_albums_by_release_date(cursor, from_date, to_date):
378     if len(to_date) == 0:

```



```

377         sql_find = "select _name, _release_date _from _albums _where _release_date _>_%
↪ s_order_by_release_date, _name"
378         data = (from_date,)
379
380     else:
381         sql_find = "select _name, _release_date _from _albums _where _release_date _
↪ between_%s_and_%s_" \
382             "order_by_release_date, _name"
383         data = (from_date, to_date)
384     try:
385         cursor.execute(sql_find, data)
386
387     except psycopg2.DataError:
388         raise ValueError('You _entered _wrong _value! _Try _again ')
389     except psycopg2.DatabaseError:
390         raise ValueError('There _was _database _error . _You _have _entered _something _
↪ wrong. ')
391     respond = cursor.fetchall()
392     print('Albums_(name, _release_date):_')
393     print_respond(respond)
394
395
396 def record_label_partnership(cursor, label_name, from_date, to_date):
397     label_name = '%' + label_name + '%'
398     if len(to_date) == 0:
399         sql_record_label_collabs = "select _rl.name, _s.name, _date_from _from _
↪ label_singer_" \
400             "join _record_labels _rl_on _rl.id=_
↪ label_singer.record_label_id_" \
401             "join _singers _s_on _s.id=_label_singer.
↪ singer_id_" \
402             "where _rl.name _like _%s _and _date_from _>_%s_
↪ and _date_to _is _null_" \
403             "order_by _rl.name, _date_from, _s.name;"
404         data = (label_name, from_date)
405         print('Founded_records_(record_label, _artists_names, _date_from): ')
406     else:
407         sql_record_label_collabs = "select _rl.name, _s.name, _date_from, _date_to _
↪ from _label_singer_" \
408             "join _record_labels _rl_on _rl.id=_
↪ label_singer.record_label_id_" \
409             "join _singers _s_on _s.id=_label_singer.
↪ singer_id_" \
410             "where _rl.name _like _%s _and _date_from _>_%s_
↪ and _date_to _<_%s_" \
411             "order_by _rl.name, _date_from, _s.name;"
412         data = (label_name, from_date, to_date)
413         print('Founded_records_(record_label, _artists_names, _date_from, _date_to)
↪ : ')
414     try:
415         cursor.execute(sql_record_label_collabs, data)
416     except psycopg2.DataError:
417         raise ValueError('You _entered _wrong _value! _Try _again ')
418     except psycopg2.DatabaseError:
419         raise ValueError('There _was _database _error . _You _have _entered _something _
↪ wrong. ')
420     respond = cursor.fetchall()
421     print_respond(respond)
422
423

```

```

424 def find_singers_by_country(cursor, country_name):
425     country_name = '%' + country_name + '%'
426
427     sql_country = "select_singers.name, singers.bio, c.name_from_singers \
428                   "join_countries_c_on_c.id=singers.countries_id" \
429                   "where_c.name_ilike_%s"
430
431     data = (country_name,)
432
433     try:
434         cursor.execute(sql_country, data)
435     except psycopg2.DataError:
436         raise ValueError('You_entered_wrong_value!_Try_again')
437     except psycopg2.DatabaseError:
438         raise ValueError('There_was_database_error._You_have_entered_something_
↪ wrong. ')
439     respond = cursor.fetchall()
440     print('Founded_artists_(name, bio, country): ')
441     print_respond(respond)
442
443
444 def find_singer_album_track(cursor, name, option):
445     name = '%' + name + '%'
446     all_options = {'s': "Founded_records_(artist's_name, album, track, release_
↪ date):",
447                   'a': "Founded_records_(album's_name, artist, track, release_
↪ date):",
448                   't': "Founded_records_(track's_name, artist, album, release_
↪ date):"}
449     print_str = all_options[option]
450     all_options.__delitem__(option)
451     sql_find = "select_{ }.name, { }.name, { }.name, a.release_date_from_
↪ tracks_singers" \
452               "join_singers_s_on_s.id=tracks_singers.singers_id" \
453               "join_tracks_t_on_t.id=tracks_singers.tracks_id" \
454               "join_track_list_tl_on_t.id=tl.tracks_id" \
455               "join_albums_a_on_tl.album_id=a.id" \
456               "where_{ }.name_ilike_%s" \
457               "order_by_{ }.name, { }.name, { }.name, a.release_date;" .format(
↪ option, list(all_options.keys())[0],
458                                                       list
↪ (all_options.keys())[1], option, option,
459                                                       list
↪ (all_options.keys())[0],
460                                                       list
↪ (all_options.keys())[1])
461     try:
462         cursor.execute(sql_find, (name,))
463     except psycopg2.DataError:
464         raise ValueError('You_entered_wrong_value!_Try_again')
465     except psycopg2.DatabaseError:
466         raise ValueError('There_was_database_error._You_have_entered_something_
↪ wrong. ')
467     respond = cursor.fetchall()
468     print(print_str)
469     print_respond(respond)

```