Połączenie z bazą danych przy pomocy sqlAlchemy

```
1  from sqlalchemy import create_engine

   db_string = "postgresql://wbauer_adb:adb2020@pgsql-196447.vipserv.org:5432/wbauer_adb"

   db = create_engine(db_string)
   connection_sqlalchemy = db.connect()
   result_set = db.execute("SELECT * FROM city")
   print(result_set)
```

```
<sqlalchemy.engine.cursor.LegacyCursorResult object at 0x0000023E30C80D90>
```

1.Ile kategorii filmów mamy w wypożyczalni?

Kod:

```
2  categories = db.execute("SELECT  * FROM film_category")
   cat_list = []
   for c in categories:
       if c[1] not in cat_list:
           cat_list.append(c[1])

   print('Wynik:')
   print('='* 90)
   print("liczba kategorii: ", len(cat_list))
```

```
Wynik:
==========================================================================================
liczba kategorii:  16
```

2.Wyświetl listę kategorii w kolejności alfabetycznej.

Kod:

```
3  categories_names = db.execute("SELECT  * FROM category")
   names_list = []
   for name in categories_names:
       names_list.append(name[1])
   names_list.sort()
   print('Wynik:')
   print('='* 90)
   print("\nLista w kategorii w kolejności alfabetycznej")
   for n in range (len(names_list)):
       print(n+1, ":",  names_list[n])
```

```
Wynik:
==========================================================================================

Lista w kategorii w kolejności alfabetycznej
1 : Action
2 : Animation
3 : Children
4 : Classics
5 : Comedy
6 : Documentary
7 : Drama
```

```
8 : Family
9 : Foreign
10 : Games
11 : Horror
12 : Music
13 : New
14 : Sci-Fi
15 : Sports
16 : Travel
```

3.Znajdź najstarszy i najmłodszy film do wypożyczenia.

Kod:

```
year = db.execute("SELECT * FROM film")
year_list = []
for d in year:
    if d[3] not in year_list:
        year_list.append(d[3])
print('Wynik:')
print('='* 90)
print('lista z wszytkimi latami premier:',year_list, '\nWszystkie filmy z 2006 roku')
```

```
Wynik:
==========================================================================================
lista z wszytkimi latami premier: [2006]
Wszystkie filmy z 2006 roku
```

4.Ile wypożyczeń odbyło się między 2005-07-01 a 2005-08-01?

Kod:

```
rental_data = db.execute("SELECT * FROM rental")
rental_data_list = []
for r in rental_data:
    rental_data_list.append(list(r[1].timetuple()))
rental_in_time = []
for n in rental_data_list:
    if n[0] == 2005:
        if n[1] == 7 or (n[1]==8 and n[2]==1):
            rental_in_time.append(n)
print('Wynik:')
print('='* 90)
print("Liczba wszytkich wypożyczeń między podanymi datami wynosi:",len(rental_in_time))
```

```
Wynik:
==========================================================================================
Liczba wszytkich wypożyczeń między podanymi datami wynosi: 7380
```

5.Ile wypożyczeń odbyło się między 2010-01-01 a 2011-02-01?

Kod:

```
rental_data = db.execute("SELECT * FROM rental")
rental_data_list = []
for r in rental_data:
    rental_data_list.append(list(r[1].timetuple()))
rental_in_time = []
for n in rental_data_list:
    if n[0] == 2010 or (n[0]==2011 and n[1]==1) or (n[0]==2011 and n[1]==2 and n[2]==1):
            rental_in_time.append(n)
print('Wynik:')
```

```
print('='* 90)
print("Liczba wszytkich wypożyczeń między podanymi datami wynosi:",len(rental_in_time))
```

```
Wynik:
==========================================================================================
Liczba wszytkich wypożyczeń między podanymi datami wynosi: 0
```

6.Znajdź największą płatność wypożyczenia.

Kod:

```
rental_amount = db.execute("SELECT amount FROM payment ")
max = -100
for r in rental_amount:
    if r[0] > max:
         max = r[0]
print('Wynik:')
print('='* 90)
print("Największa opłata za film wyniosła",max)
```

```
Wynik:
==========================================================================================
Największa opłata za film wyniosła 11.99
```

7.Znajdź wszystkich klientów z Polski, Nigerii lub Bangladeszu.

Kod:

```
customers_ = db.execute("SELECT customer.first_name,customer.last_name ,country.country "
                        "FROM (((customer "
                        "INNER JOIN address ON customer.address_id = address.address_id)"
                        "INNER JOIN city ON address.city_id = city.city_id)"
                        "INNER JOIN country ON city.country_id = country.country_id)"
                        "WHERE country IN ('Poland', 'Nigeria', 'Bangladesh') "
                        "ORDER BY country ASC")
print('Wynik:')
print('='* 90)
print('Lista klientów z Polski, Bangladeszu i Nigerii: \n')
print("{:15}".format('First Name')+'| '+"{:15}".format('Last Name') +'| '+ "{:15}".format('Country'))
print('========================================')
for r in customers_:
    print("{:15}".format(r[0])+ '| '+"{:15}".format(r[1]) + '| '+ "{:15}".format(r[2]))
```

```
Wynik:
==========================================================================================
Lista klientów z Polski, Bangladeszu i Nigerii:

First Name     | Last Name      | Country
========================================
Michelle       | Clark          | Bangladesh
Stephen        | Qualls         | Bangladesh
Frank          | Waggoner       | Bangladesh
Marilyn        | Ross           | Nigeria
Elsie          | Kelley         | Nigeria
Gladys         | Hamilton       | Nigeria
Sonia          | Gregory        | Nigeria
Rodney         | Moeller        | Nigeria
Velma          | Lucas          | Nigeria
Carol          | Garcia         | Nigeria
Olga           | Jimenez        | Nigeria
Bertha         | Ferguson       | Nigeria
Tracey         | Barrett        | Nigeria
Jo             | Fowler         | Nigeria
```

```
Wallace         | Slone           | Nigeria
Constance       | Reid            | Nigeria
Brian           | Wyman           | Poland
Sidney          | Burleson        | Poland
Marjorie        | Tucker          | Poland
Russell         | Brinson         | Poland
Leah            | Curtis          | Poland
Ruben           | Geary           | Poland
Johnnie         | Chisholm        | Poland
Jimmie          | Eggleston       | Poland
```

8.Gdzie mieszkają członkowie personelu?

Kod:

```
15
staff_ =  db.execute("SELECT staff.first_name,staff.last_name,country.country ,city.city , address.ad
                        "FROM (((staff "
                        "INNER JOIN address ON staff.address_id = address.address_id)"
                        "INNER JOIN city ON address.city_id = city.city_id)"
                        "INNER JOIN country ON city.country_id = country.country_id)"
                        "ORDER BY country ASC")

print('Wynik:')
print('='* 90)
print('Adresy Pracowników: \n')
print("{:15}".format('First Name')+'| '+"{:15}".format('Last Name') +'| '+ "{:15}".format('Country') +
print('='*17*5)
for r in staff_:
    print("{:15}".format(r[0])+ '| '+"{:15}".format(r[1]) + '| '+ "{:15}".format(r[2])+ '| '+ "{:15}"

Wynik:
==========================================================================================
Adresy Pracowników:

First Name     | Last Name      | Country        | City           | Address
==========================================================================================
Jon            | Stephens       | Australia      | Woodridge      | 1411 Lillydale Drive
Mike           | Hillyer        | Canada         | Lethbridge     | 23 Workhaven Lane
```

9.Ilu pracowników mieszka w Argentynie lub Hiszpanii?

Kod:

```
10  staff_in_spain =  db.execute("SELECT staff.first_name,staff.last_name,country.country ,city.city , ad
                        "FROM (((staff "
                        "INNER JOIN address ON staff.address_id = address.address_id)"
                        "INNER JOIN city ON address.city_id = city.city_id)"
                        "INNER JOIN country ON city.country_id = country.country_id)"
                        "WHERE country IN ('Spain')")
    staff_in_argentina =  db.execute("SELECT staff.first_name,staff.last_name,country.country ,city.city
                        "FROM (((staff "
                        "INNER JOIN address ON staff.address_id = address.address_id)"
                        "INNER JOIN city ON address.city_id = city.city_id)"
                        "INNER JOIN country ON city.country_id = country.country_id)"
                        "WHERE country IN ('Argentina')")
    spain_list = []
    argentina_list = []
    for s in staff_in_spain:
        spain_list.append(s)
    for a in staff_in_argentina:
        argentina_list.append(a)
    print('Wynik:')
    print('='* 90)
```

```
print("Liczba osób z personelu mieszkających w Hiszpanii: ", len(spain_list))
print("Liczba osób z personelu mieszkających w Argentynie: ", len(argentina_list))
```

```
Wynik:
==========================================================================================
Liczba osób z personelu mieszkających w Hiszpanii:  0
Liczba osób z personelu mieszkających w Argentynie:  0
```

10.Jakie kategorie filmów zostały wypożyczone przez klientów?

Kod:

```
11 rentaled_categories =  db.execute("SELECT  DISTINCT category.name, category.category_id "
                          "FROM (((rental "
                          "INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id)"
                          "INNER JOIN film_category ON inventory.film_id = film_category.film_id)"
                          "INNER JOIN category ON category.category_id = film_category.category_id)"
                                    "ORDER BY category_id ASC")
print('Wynik:')
print('='* 90)
print('Lista wypożyczonych kategorii filmowych: \n')
print("{:^15}".format('Category ID')+'| '+"{:^15}".format('Category name'))
print('========================================')
for r in rentaled_categories:
    print("{:^15}".format(r[1])+ '| '+"{:^15}".format(r[0]))
```

```
Wynik:
==========================================================================================
Lista wypożyczonych kategorii filmowych:

  Category ID  |  Category name
========================================
       1       |     Action
       2       |    Animation
       3       |    Children
       4       |    Classics
       5       |     Comedy
       6       |   Documentary
       7       |      Drama
       8       |     Family
       9       |     Foreign
      10       |      Games
      11       |     Horror
      12       |      Music
      13       |       New
      14       |     Sci-Fi
      15       |     Sports
      16       |     Travel
```

11.Znajdź wszystkie kategorie filmów wypożyczonych w Ameryce.

Kod:

```
12 rentaled_categories_america =  db.execute("SELECT DISTINCT category.name, category.category_id, count
                          "FROM (((((((rental "
                          "INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id)"
                          "INNER JOIN store ON inventory.store_id = store.store_id)"
                          "INNER JOIN address ON address.address_id = store.address_id)"
                          "INNER JOIN city ON address.city_id = city.city_id)"
                          "INNER JOIN country ON city.country_id = country.country_id)"
                          "INNER JOIN film_category ON inventory.film_id = film_category.film_id)"
                          "INNER JOIN category ON category.category_id = film_category.category_id)"
                          "WHERE country.country IN ('Canada','United States','Mexico')")
```

```
print('Wynik:')
print('='* 90)
print("Kategorie filmowe wypożyczone w północnej Ameryce:\n")
print("{:15}".format('Category ID')+'| '+"{:15}".format('Category Name') +'| '+ "{:15}".format('Count
print('='*17*3)
for r in rentaled_categories_america:
    print("{:15}".format(r[1])+ '| '+"{:^15}".format(r[0]) + '| '+ "{:15}".format(r[2]))

Wynik:
==========================================================================================
Kategorie filmowe wypożyczone w północnej Ameryce:

Category ID    | Category Name  | Country
================================================
             16|     Travel     | Canada
              6|   Documentary  | Canada
              8|     Family     | Canada
              5|     Comedy     | Canada
              3|    Children    | Canada
              2|    Animation   | Canada
             15|     Sports     | Canada
             12|      Music     | Canada
              9|     Foreign    | Canada
             14|     Sci-Fi     | Canada
             13|       New      | Canada
             10|      Games     | Canada
              4|    Classics    | Canada
              1|     Action     | Canada
             11|     Horror     | Canada
              7|      Drama     | Canada
```

12.Znajdź wszystkie tytuły filmów, w których grał: Olympia Pfeiffer lub Julia Zellweger lub Ellen Presley

Kod:

```
13 actors =  db.execute("SELECT actor.first_name, actor.last_name, film.title "
                        "FROM ((actor "
                        "INNER JOIN film_actor ON film_actor.actor_id = actor.actor_id)"
                        "INNER JOIN film ON film.film_id = film_actor.film_id)"
                        "WHERE ((actor.first_name = 'Olympia' and actor.last_name = 'Pfeiffer') or"
                        "(actor.first_name = 'Julia' and actor.last_name = 'Zellweger') or "
                        "(actor.first_name = 'Ellen' and actor.last_name = 'Presley'))")
print('Wynik:')
print('='* 90)
print("Kategorie filmowe wypożyczone w północnej Ameryce:\n")
print("{:15}".format('First Name')+'| '+"{:15}".format('Last Name') +'| '+ "{:15}".format('Film title
print('='*17*3)
for a in actors:
    print("{:15}".format(a[1])+ '| '+"{:15}".format(a[0]) + '| '+ "{:15}".format(a[2]))

Wynik:
==========================================================================================
Kategorie filmowe wypożyczone w północnej Ameryce:

First Name     | Last Name      | Film title
================================================
Presley        | Ellen          | Bilko Anonymous
Presley        | Ellen          | Caribbean Liberty
Presley        | Ellen          | Casper Dragonfly
Presley        | Ellen          | Empire Malkovich
Presley        | Ellen          | Floats Garden
Presley        | Ellen          | Frogmen Breaking
Presley        | Ellen          | Homeward Cider
Presley        | Ellen          | Hyde Doctor
```

Presley	Ellen	Image Princess
Presley	Ellen	Jacket Frisco
Presley	Ellen	Microcosmos Paradise
Presley	Ellen	Network Peak
Presley	Ellen	Oscar Gold
Presley	Ellen	Pickup Driving
Presley	Ellen	Pinocchio Simon
Presley	Ellen	Private Drop
Presley	Ellen	Roots Remember
Presley	Ellen	Scarface Bang
Presley	Ellen	Secretary Rouge
Presley	Ellen	Spy Mile
Presley	Ellen	Streetcar Intentions
Presley	Ellen	Tadpole Park
Presley	Ellen	Treasure Command
Presley	Ellen	Turn Star
Presley	Ellen	Women Dorado
Pfeiffer	Olympia	Badman Dawn
Pfeiffer	Olympia	Chitty Lock
Pfeiffer	Olympia	Color Philadelphia
Pfeiffer	Olympia	Contact Anonymous
Pfeiffer	Olympia	Deep Crusade
Pfeiffer	Olympia	Effect Gladiator
Pfeiffer	Olympia	Express Lonely
Pfeiffer	Olympia	Firehouse Vietnam
Pfeiffer	Olympia	Fugitive Maguire
Pfeiffer	Olympia	Hanky October
Pfeiffer	Olympia	Ice Crossing
Pfeiffer	Olympia	Idols Snatchers
Pfeiffer	Olympia	Intolerable Intentions
Pfeiffer	Olympia	Magnolia Forrester
Pfeiffer	Olympia	Mars Roman
Pfeiffer	Olympia	Maude Mod
Pfeiffer	Olympia	Murder Antitrust
Pfeiffer	Olympia	None Spiking
Pfeiffer	Olympia	Others Soup
Pfeiffer	Olympia	Psycho Shrunk
Pfeiffer	Olympia	Santa Paris
Pfeiffer	Olympia	Sense Greek
Pfeiffer	Olympia	Storm Happiness
Pfeiffer	Olympia	Sweet Brotherhood
Pfeiffer	Olympia	Titanic Boondock
Pfeiffer	Olympia	Tourist Pelican
Pfeiffer	Olympia	Traffic Hobbit
Pfeiffer	Olympia	Wait Cider
Zellweger	Julia	Breakfast Goldfinger
Zellweger	Julia	Cranes Reservoir
Zellweger	Julia	Dares Pluto
Zellweger	Julia	Detective Vision
Zellweger	Julia	Divorce Shining
Zellweger	Julia	Hollow Jeopardy
Zellweger	Julia	Jeopardy Encino
Zellweger	Julia	Lambs Cincinatti
Zellweger	Julia	Majestic Floats
Zellweger	Julia	Minds Truman
Zellweger	Julia	Open African
Zellweger	Julia	Outlaw Hanky
Zellweger	Julia	Panky Submarine
Zellweger	Julia	Rider Caddyshack
Zellweger	Julia	Won Dares
Zellweger	Julia	Wyoming Storm