



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря
Сікорського»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування та спеціалізованих
комп'ютерних систем

Лабораторна робота №2
з дисципліни
«Бази даних і засоби управління»

Тема: «Створення додатку бази даних,
орієнтованого на взаємодію з СУБД
PostgreSQL»

Виконав: студент III курсу

ФПМ групи КВ-81

Рожко Д.В.

Перевірив:

Київ – 2020

Завдання

Вимоги до пункту №1 деталізованого завдання:

- ілюстрації обробки виняткових ситуацій (помилки) при введенні/вилученні даних;
- ілюстрації валідації даних при введенні користувачем.

Вимоги до пункту №2 деталізованого завдання:

- копії екрану (ілюстрації) з фрагментами згенерованих даних таблиць;
- копії SQL-запитів, що ілюструють генерацію при визначених вхідних параметрах.

Вимоги до пункту №3 деталізованого завдання:

- ілюстрації введення пошукового запиту та результатів виконання запитів;
- копії SQL-запитів, що ілюструють пошук з зазначеними початковими параметрами

Вимоги до пункту №4 деталізованого завдання:

- ілюстрації програмного коду з репозиторію Git.

Пункт №1

*ілюстрації обробки виняткових ситуацій (помилки) при
уведенні/вилучення даних;*

Обробка результатів при **коректному** введенні даних в таблицю TYPE

```
TABLE "type":
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {}
TABLE if date use format dd/mm/yyyy 'type'
Input 'type_id':
15
INT
Input 'spoil_quick':
False
Input 'type_name':
acid
type
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {}
5. {'type_id': 15, 'spoil_quick': False, 'type_name': 'acid'}

Process finished with exit code 0
```

Обробка результатів при **коректному** видаленні даних з таблиці TYPE

```
TABLE "type":
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {'type_id': 15, 'spoil_quick': False, 'type_name': 'acid'}
5. {}
Date input in format d/m/y
Do you want to delete row input [Y]:
y
Input attribute ['type_id', 'spoil_quick', 'type_name'] to delete:
type_id
Input value of attribute:
15
-----
We have just removed <built-in function id> from our list
-----
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}

Process finished with exit code 0
```

Обробка помилок з таблиці TYPE (не існуючий атрибут таблиці)

```
TABLE "type":
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {}
Do you want to test task 1 (Yes/No)[Y/N]:
y
Date input in format d/m/y
Do you want to delete row input [Y]:
y
Input attribute ['type_id', 'spoil_quick', 'type_name'] to delete:
person_id
WARNING: Incorrect attribute
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {}

Process finished with exit code 0
```

Обробка помилок з таблиці TYPE (не коректний тип, *type_id(integer)*, а не (*string*))

```
TABLE "type":
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {}
Do you want to test task 1 (Yes/No)[Y/N]:
y
Date input in format d/m/y
Do you want to delete row input [Y]:
y
Input attribute ['type_id', 'spoil_quick', 'type_name'] to delete:
type_id
Input value of attribute:
False

AttributeError: 'bool' object has no attribute 'upper'
```

Обробка **помилки** з таблиці TYPE (не коректний тип, *spoil_quick(boolean)*, а не *(string)*)

```
TABLE "type":
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {}
Do you want to test task 1 (Yes/No)[Y/N]:
y
Date input in format d/m/y
Do you want to delete row input [Y]:
n
TABLE if date use format dd/mm/yyyy 'type'
Input 'type_id':
54
INT
Input 'spoil_quick':
abracadabra
Input 'type_name':
magic
Traceback (most recent call last):
  File "E:\КПИ\5_semester\PostgreSQL\lab2-mvc\MVC.py", line 21, in <module>
    c.insert_item(list_input)
  File "E:\КПИ\5_semester\PostgreSQL\lab2-mvc\processing\entity\type\ControllerType.py",
    spoil_quick = eval(list_atr[1])
  File "<string>", line 1, in <module>
NameError: name 'abracadabra' is not defined
```

Обробка помилок з таблиці TYPE (не коректний тип, *type_name* (*string*), а не (*integer*))

```
TABLE "type":
--- TYPE LIST ---
1. {'type_id': 1, 'spoil_quick': True, 'type_name': 'animal'}
2. {'type_id': 2, 'spoil_quick': False, 'type_name': 'electronics'}
3. {'type_id': 3, 'spoil_quick': True, 'type_name': 'plants'}
4. {}
Do you want to test task 1 (Yes/No)[Y/N]:
y
Date input in format d/m/y
Do you want to delete row input [Y]:
n
TABLE if date use format dd/mm/yyyy 'type'
Input 'type_id':
20
INT
Input 'spoil_quick':
False
Input 'type_name':
20
Traceback (most recent call last):
  File "E:\КПИ\5 semester\PostgreSQL\lab2-mvc\MVC.py", line 21, in <m
    c.insert_item(list_input)
  File "E:\КПИ\5 semester\PostgreSQL\lab2-mvc\processing\entity\type\
    self.validation(type_id, spoil_quick, type_name)
  File "E:\КПИ\5 semester\PostgreSQL\lab2-mvc\processing\entity\type\
    assert isinstance(type_name, str), '\type_name\' must be \'strin
AssertionError: 'type_name' must be 'string' type
TYPE
```

Обробка помилок з таблиці THING (не коректний тип, *expiration_date*(*date*), а не (*integer*))

```
TABLE "thing":
--- THING LIST ---
1. {'thing_id': 1, 'quantity': 150, 'expiration_date': datetime.date(2001, 3, 25), 'breakable': True, 'type_id': 2, 'order_id': 1, 'name': 'computer M-1'}
2. {'thing_id': 2, 'quantity': 10000, 'expiration_date': datetime.date(2001, 2, 24), 'breakable': True, 'type_id': 1, 'order_id': 2, 'name': 'egg'}
3. {'thing_id': 3, 'quantity': 1000, 'expiration_date': datetime.date(2002, 2, 10), 'breakable': False, 'type_id': 3, 'order_id': 2, 'name': 'corn'}
4. {'thing_id': 4, 'quantity': 5025, 'expiration_date': datetime.date(2000, 6, 5), 'breakable': False, 'type_id': 3, 'order_id': 3, 'name': 'tomatoes'}
5. {'thing_id': 5, 'quantity': 6578, 'expiration_date': datetime.date(2003, 6, 20), 'breakable': True, 'type_id': 2, 'order_id': 4, 'name': 'phone F1'}
6. {'thing_id': 7, 'quantity': 5, 'expiration_date': datetime.date(2000, 12, 21), 'breakable': False, 'type_id': 3, 'order_id': 2, 'name': 'pears'}
7. {}
```

```

Do you want to test task 1 (Yes/No)[Y/N]:
Y
Date input in format d/m/y
Do you want to delete row input [Y]:
n
TABLE if date use format dd/mm/yyyy 'thing'
Input 'thing_id':
1
INT
Input 'quantity':
5
INT
Input 'expiration_date':
6
INT
Input 'breakable':
False
Input 'type_id':
5
INT
Input 'order_id':
15
INT
Input 'name':
hahhaha
[1, 5, 6, 'False', 5, 15, 'hahhaha']

```

Traceback (most recent call last):

```

File "E:\KPII\5_semester\PostgreSQL\lab2-mvc\MVC.py", line 21, in <module>
    c.insert_item(list_input)

```

```

File "E:\KPII\5_semester\PostgreSQL\lab2-mvc\processing\entity\thing\Controlle
    expiration_date = date.strptime(list_atr[2], '%d/%m/%Y')

```

TypeError: strptime() argument 1 must be str, not int

Пункт №2

Занум:

```
SELECT a.name, b.type_name, a.quantity
FROM thing a INNER JOIN type b
      ON(a.type_id=b.type_id)
WHERE type_name LIKE 'plants';
```

```
QUERY :
      SELECT a.name, b.type_name, a.quantity
      FROM thing a INNER JOIN type b ON(a.type_id=b.type_id)
      WHERE type_name LIKE 'plants';
-----
['corn', 'plants', 1000]
['tomatoes', 'plants', 5025]
['pears', 'plants', 5]
Current Time = 02:32:15.749539
-----
```

Занум:

```
SELECT a.name, b.type_name, a.quantity, o.date
FROM thing a INNER JOIN public.type b
      ON(a.type_id=b.type_id)
      INNER JOIN public.order o ON(o.order_id=a.order_id)
WHERE o.date BETWEEN TO_DATE('01.01.2001', 'DD.MM.YYYY')AND
      TO_DATE('31.12.2001', 'DD.MM.YYYY');
```

```
QUERY :
      SELECT a.name, b.type_name, a.quantity, o.date
      FROM thing a INNER JOIN public.type b ON(a.type_id=b.type_id)
      INNER JOIN public.order o ON(o.order_id=a.order_id)
      WHERE o.date BETWEEN TO_DATE('01.01.2001', 'DD.MM.YYYY')
      AND TO_DATE('31.12.2001', 'DD.MM.YYYY');
-----
['tomatoes', 'plants', 5025, datetime.date(2001, 11, 4)]
['phone F1', 'electronics', 6578, datetime.date(2001, 3, 25)]
Current Time = 02:32:15.749539
-----
```

Занум:

```
SELECT a.name, b.type_name, a.quantity, o.quantity,
      p.name, b.spoil_quick
FROM thing a INNER JOIN public.type b
      ON(a.type_id=b.type_id)
      INNER JOIN public.order o ON(o.order_id=a.order_id)
```



```
INNER JOIN person p ON(o.person_id=p.person_id)
WHERE b.spoil_quick IS true;
```

```
QUERY :
  SELECT a.name, b.type_name, a.quantity, o.quantity, p.name, b.spoil_quick
FROM thing a INNER JOIN public.type b ON(a.type_id=b.type_id)
      INNER JOIN public.order o ON(o.order_id=a.order_id)
      INNER JOIN person p ON(o.person_id=p.person_id)
WHERE b.spoil_quick IS true;
-----
['egg', 'animal', 10000, 7, 'Paul McCartney', True]
['corn', 'plants', 1000, 7, 'Paul McCartney', True]
['tomatoes', 'plants', 5025, 4, 'Steve Martin', True]
['pears', 'plants', 5, 7, 'Paul McCartney', True]
Current Time = 02:32:15.749539
-----
```

Допустиме також консольне введення даних для констант

```
-----
CHANGE DATE IN QUERY:
SELECT a.name, b.type_name, a.quantity, o.date
FROM thing a INNER JOIN public.type b ON(a.type_id=b.type_id)
      INNER JOIN public.order o ON(o.order_id=a.order_id)
WHERE o.date BETWEEN TO_DATE('01.01.2001', 'DD.MM.YYYY')
      AND TO_DATE('31.12.2001', 'DD.MM.YYYY');
-----
INPUT FIRST DATE IN FORMAT 'DD.MM.YYYY':
15.06.2001
INPUT SECOND DATE IN FORMAT 'DD.MM.YYYY':
31.12.2001
['tomatoes', 'plants', 5025, datetime.date(2001, 11, 4)]
Do you want to test task 3 (Yes/No)[Y/N]:
```

Пункт №3

- ілюстрації уведення пошукового запиту та результатів виконання запитів;

```
Do you want to test task 3 (Yes/No)[Y/N]:
Y
1 | {'num': 75.0, 'random string': 'SU'}
2 | {'num': 78.0, 'random string': 'LW'}
3 | {'num': 78.0, 'random string': 'NV'}
4 | {'num': 81.0, 'random string': 'JW'}
5 | {'num': 75.0, 'random string': 'TN'}
6 | {'num': 77.0, 'random string': 'JO'}
7 | {'num': 76.0, 'random string': 'IO'}
8 | {'num': 78.0, 'random string': 'VR'}
9 | {'num': 80.0, 'random string': 'KQ'}
10 | {'num': 74.0, 'random string': 'PO'}
11 | {'num': 74.0, 'random string': 'QK'}
12 | {'num': 75.0, 'random string': 'IK'}
13 | {'num': 80.0, 'random string': 'JQ'}
14 | {'num': 78.0, 'random string': 'UL'}
15 | {'num': 79.0, 'random string': 'NI'}
16 | {'num': 73.0, 'random string': 'JV'}
17 | {'num': 78.0, 'random string': 'KP'}
18 | {'num': 77.0, 'random string': 'WP'}
19 | {'num': 77.0, 'random string': 'OO'}
20 | {'num': 79.0, 'random string': 'RV'}
21 | {'num': 81.0, 'random string': 'IS'}
22 | {'num': 85.0, 'random string': 'PO'}
23 | {'num': 73.0, 'random string': 'WK'}
24 | {'num': 87.0, 'random string': 'QV'}
25 | {'num': 85.0, 'random string': 'IL'}
26 | {'num': 74.0, 'random string': 'QT'}
27 | {'num': 81.0, 'random string': 'JN'}
28 | {'num': 74.0, 'random string': 'OS'}
```

```
29 | {'num': 79.0, 'random string': 'QK'}
30 | {'num': 84.0, 'random string': 'VQ'}
31 | {'num': 85.0, 'random string': 'IR'}
32 | {'num': 83.0, 'random string': 'JV'}
33 | {'num': 85.0, 'random string': 'PQ'}
34 | {'num': 87.0, 'random string': 'RV'}
35 | {'num': 78.0, 'random string': 'IN'}
36 | {'num': 81.0, 'random string': 'JN'}
37 | {'num': 83.0, 'random string': 'JT'}
38 | {'num': 85.0, 'random string': 'IR'}
39 | {'num': 75.0, 'random string': 'SW'}
40 | {'num': 74.0, 'random string': 'QT'}
41 | {'num': 84.0, 'random string': 'OM'}
42 | {'num': 86.0, 'random string': 'OL'}
43 | {'num': 85.0, 'random string': 'LW'}
44 | {'num': 74.0, 'random string': 'QJ'}
45 | {'num': 74.0, 'random string': 'OQ'}
46 | {'num': 86.0, 'random string': 'OQ'}
47 | {'num': 75.0, 'random string': 'II'}
48 | {'num': 78.0, 'random string': 'KP'}
49 | {'num': 81.0, 'random string': 'VT'}
50 | {'num': 85.0, 'random string': 'PW'}
51 | {'num': 76.0, 'random string': 'UR'}
52 | {'num': 82.0, 'random string': 'JK'}
53 | {'num': 75.0, 'random string': 'UQ'}
54 | {'num': 76.0, 'random string': 'KP'}
55 | {'num': 75.0, 'random string': 'LW'}
56 | {'num': 83.0, 'random string': 'JI'}
57 | {'num': 82.0, 'random string': 'RS'}
58 | {'num': 81.0, 'random string': 'QI'}
59 | {'num': 76.0, 'random string': 'PU'}
```

```

60 | {'num': 84.0, 'random string': 'OS'}
61 | {'num': 73.0, 'random string': 'WP'}
62 | {'num': 78.0, 'random string': 'MW'}
63 | {'num': 82.0, 'random string': 'QW'}
64 | {'num': 81.0, 'random string': 'PM'}
65 | {'num': 82.0, 'random string': 'PT'}
66 | {'num': 85.0, 'random string': 'IJ'}
67 | {'num': 84.0, 'random string': 'WT'}
68 | {'num': 77.0, 'random string': 'WL'}
69 | {'num': 74.0, 'random string': 'TV'}
70 | {'num': 74.0, 'random string': 'VW'}
71 | {'num': 81.0, 'random string': 'OK'}
72 | {'num': 83.0, 'random string': 'MU'}
73 | {'num': 85.0, 'random string': 'IT'}
74 | {'num': 73.0, 'random string': 'NT'}
75 | {'num': 84.0, 'random string': 'MQ'}
76 | {'num': 79.0, 'random string': 'SP'}
77 | {'num': 80.0, 'random string': 'NP'}
78 | {'num': 87.0, 'random string': 'OS'}
79 | {'num': 83.0, 'random string': 'SK'}
80 | {'num': 76.0, 'random string': 'US'}
81 | {'num': 77.0, 'random string': 'NJ'}
82 | {'num': 84.0, 'random string': 'PV'}
83 | {'num': 77.0, 'random string': 'RP'}
84 | {'num': 76.0, 'random string': 'LS'}
85 | {'num': 81.0, 'random string': 'OS'}
86 | {'num': 86.0, 'random string': 'QI'}
87 | {'num': 83.0, 'random string': 'KI'}
88 | {'num': 84.0, 'random string': 'VT'}
89 | {'num': 77.0, 'random string': 'OS'}
90 | {'num': 84.0, 'random string': 'NS'}

```

```

91 | {'num': 79.0, 'random string': 'IR'}
92 | {'num': 73.0, 'random string': 'LU'}
93 | {'num': 83.0, 'random string': 'SR'}
94 | {'num': 76.0, 'random string': 'NM'}
95 | {'num': 82.0, 'random string': 'QR'}
96 | {'num': 84.0, 'random string': 'ML'}
97 | {'num': 74.0, 'random string': 'MU'}
98 | {'num': 83.0, 'random string': 'VV'}
99 | {'num': 73.0, 'random string': 'LT'}
100 | {'num': 85.0, 'random string': 'UO'}

```

- *копії SQL-запитів, що ілюструють генерацію при визначених вхідних параметрах.*

```

SELECT trunc(73 + random()*15),
       chr(trunc(73 + random()*15)::int) || chr(trunc(73 +
random()*15)::int)
FROM generate_series(1,100);

```