

## Отчет по практической работе №15

**Тема:** составление программ с использованием регулярных выражений в IDE PyCharm Community.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ для работы с базами данных в IDE PyCharm Community.

Вариант 5

**Тип алгоритма:** базами данных

**Текст программы:**

```
import sqlite3

# Создаем подключение к базе данных
conn = sqlite3.connect('library.db')
cursor = conn.cursor()

# Создаем таблицы
cursor.execute("""
    CREATE TABLE IF NOT EXISTS Авторы (
        КодАвтора INTEGER PRIMARY KEY,
        Фамилия TEXT,
        Имя TEXT
    )
""")

cursor.execute("""
    CREATE TABLE IF NOT EXISTS Книги (
        КодКниги INTEGER PRIMARY KEY,
        Название TEXT,
        Раздел TEXT,
        Издательство TEXT,
        ГодИздания INTEGER,
        МестоХранения TEXT,
        FOREIGN KEY (Раздел) REFERENCES Разделы(Раздел),
        FOREIGN KEY (Издательство) REFERENCES Издательства(Издательство)
    )
""")

cursor.execute("""
    CREATE TABLE IF NOT EXISTS Разделы (
        Раздел TEXT PRIMARY KEY
    )
""")

cursor.execute("""
    CREATE TABLE IF NOT EXISTS Издательства (
        Издательство TEXT PRIMARY KEY,
        Город TEXT
    )
""")

cursor.execute("""
    CREATE TABLE IF NOT EXISTS АвторКниги (
```

```

КодАвтораКниги INTEGER PRIMARY KEY,
КодКниги INTEGER,
КодАвтора INTEGER,
FOREIGN KEY (КодКниги) REFERENCES Книги(КодКниги),
FOREIGN KEY (КодАвтора) REFERENCES Авторы(КодАвтора)
)
'''

```

# Заполняем таблицы произвольными данными

```

cursor.execute("INSERT INTO Авторы (Фамилия, Имя) VALUES ('Иванов', 'Иван')")
cursor.execute("INSERT INTO Авторы (Фамилия, Имя) VALUES ('Петров', 'Петр')")
cursor.execute("INSERT INTO Авторы (Фамилия, Имя) VALUES ('Сидоров', 'Сидор')")

```

```

cursor.execute("INSERT INTO Книги (Название, Раздел, Издательство, ГодИздания,
МестоХранения) VALUES ('Книга 1', 'Раздел 1', 'Издательство 1', 2020, 'Место 1')")
cursor.execute("INSERT INTO Книги (Название, Раздел, Издательство, ГодИздания,
МестоХранения) VALUES ('Книга 2', 'Раздел 2', 'Издательство 2', 2018, 'Место 2')")
cursor.execute("INSERT INTO Книги (Название, Раздел, Издательство, ГодИздания,
МестоХранения) VALUES ('Книга 3', 'Раздел 1', 'Издательство 1', 2021, 'Место 3')")

```

```

cursor.execute("SELECT COUNT(*) FROM Разделы WHERE Раздел = 'Раздел 1'")
result = cursor.fetchone()

```

```

if result[0] == 0:

```

```

    cursor.execute("INSERT INTO Разделы (Раздел) VALUES ('Раздел 1')")
    conn.commit()

```

```

else:

```

```

    print("Запись уже существует")

```

```

cursor.execute("SELECT COUNT(*) FROM Разделы WHERE Раздел = 'Раздел 2'")

```

```

result = cursor.fetchone()

```

```

if result[0] == 0:

```

```

    # Записи не существует, выполняем вставку

```

```

    cursor.execute("INSERT INTO Разделы (Раздел) VALUES ('Раздел 2')")
    conn.commit()

```

```

else:

```

```

    print("Запись уже существует")

```

```

cursor.execute("SELECT COUNT(*) FROM Разделы WHERE Раздел = 'Раздел 3'")

```

```

result = cursor.fetchone()

```

```

if result[0] == 0:

```

```

    # Записи не существует, выполняем вставку

```

```

    cursor.execute("INSERT INTO Разделы (Раздел) VALUES ('Раздел 3')")
    conn.commit()

```

```

else:

```

```

    print("Запись уже существует")

```

# Проверка наличия записи

```

cursor.execute("SELECT COUNT(*) FROM Издательства WHERE Издательство =
'Издательство 1'")

```

```

result = cursor.fetchone()

```

```

if result[0] == 0:

```

```

    # Записи не существует, выполняем вставку

```

```

    cursor.execute("INSERT INTO Издательства (Издательство, Город) VALUES
('Издательство 1', 'Город 1')")

```

```

    conn.commit()

```

```

else:

```

```

    print("Запись уже существует")

```

```
cursor.execute("SELECT * FROM Издательства")
result = cursor.fetchall()
for row in result:
    print(row)
```

```
cursor.execute("INSERT INTO АвторКниги (КодКниги, КодАвтора) VALUES (1, 1)")
cursor.execute("INSERT INTO АвторКниги (КодКниги, КодАвтора) VALUES (2, 2)")
cursor.execute("INSERT INTO АвторКниги (КодКниги, КодАвтора) VALUES (3, 3)")
```

```
# SQL-запросы на выборку данных из БД
print("1. Получить список всех книг, отсортированных по году издания:")
cursor.execute("SELECT * FROM Книги ORDER BY ГодИздания")
print(cursor.fetchall())
```

```
print("\n2. Получить список всех книг заданного автора:")
cursor.execute("SELECT * FROM Книги INNER JOIN АвторКниги ON Книги.КодКниги =
АвторКниги.КодКниги INNER JOIN Авторы ON АвторКниги.КодАвтора =
Авторы.КодАвтора WHERE Авторы.Фамилия = 'Иванов'")
print(cursor.fetchall())
```

```
print("\n3. Получить список всех книг из заданного раздела:")
cursor.execute("SELECT * FROM Книги WHERE Раздел = 'Раздел 1'")
print(cursor.fetchall())
```

```
print("\n4. Получить список всех книг, изданных заданным издательством:")
cursor.execute("SELECT * FROM Книги WHERE Издательство = 'Издательство 1'")
print(cursor.fetchall())
```

```
print("\n5. Получить список всех авторов в алфавитном порядке:")
cursor.execute("SELECT * FROM Авторы ORDER BY Фамилия, Имя")
print(cursor.fetchall())
```

```
print("\n6. Получить список всех книг, отсортированных по названию и году издания:")
cursor.execute("SELECT * FROM Книги ORDER BY Название, ГодИздания")
print(cursor.fetchall())
```

```
print("\n7. Получить список всех книг заданного автора, отсортированных по году
издания:")
cursor.execute("SELECT * FROM Книги INNER JOIN АвторКниги ON Книги.КодКниги =
АвторКниги.КодКниги INNER JOIN Авторы ON АвторКниги.КодАвтора =
Авторы.КодАвтора WHERE Авторы.Фамилия = 'Иванов' ORDER BY Книги.ГодИздания")
print(cursor.fetchall())
```

```
print("\n8. Получить список всех книг, опубликованных в заданном году:")
cursor.execute("SELECT * FROM Книги WHERE ГодИздания = 2021")
print(cursor.fetchall())
```

```
print("\n9. Получить список всех авторов, написавших книги для заданного
издательства:")
cursor.execute("SELECT DISTINCT Авторы.* FROM Авторы INNER JOIN АвторКниги ON
Авторы.КодАвтора = АвторКниги.КодАвтора INNER JOIN Книги ON
АвторКниги.КодКниги = Книги.КодКниги WHERE Книги.Издательство = 'Издательство
1'")
```

```
print(cursor.fetchall())
```

```
print("\n10. Получить список всех книг, в названии которых есть заданное слово:")  
cursor.execute("SELECT * FROM Книги WHERE Название LIKE '%книга%'")  
print(cursor.fetchall())
```

```
# SQL-запросы на обновление данных в БД  
print("\nSQL-запросы на обновление данных в БД:")
```

```
print("\n1. Обновить год издания всех книг, написанных автором с фамилией 'Иванов',  
установив год издания равным 2022:")  
cursor.execute("UPDATE Книги SET ГодИздания = 2022 WHERE КодКниги IN (SELECT  
КодКниги FROM АвторКниги INNER JOIN Авторы ON АвторКниги.КодАвтора =  
Авторы.КодАвтора WHERE Авторы.Фамилия = 'Иванов')")  
conn.commit()
```

```
print("\n2. Обновить название и год издания книги, хранящейся в городе 'Москва',  
установив название 'Новая книга' и год издания равным 2023:")  
cursor.execute("UPDATE Книги SET Название = 'Новая книга', ГодИздания = 2023 WHERE  
КодКниги IN (SELECT КодКниги FROM Издательства INNER JOIN Книги ON  
Издательства.Издательство = Книги.Издательство WHERE Издательства.Город =  
'Москва')")  
conn.commit()
```

```
print("\n3. Обновить название и раздел всех книг, написанных автором с именем  
'Александр' и фамилией 'Петров', установив название 'Новое название' и раздел  
'Фантастика':")  
cursor.execute("UPDATE Книги SET Название = 'Новое название', Раздел = 'Фантастика'  
WHERE КодКниги IN (SELECT КодКниги FROM АвторКниги INNER JOIN Авторы ON  
АвторКниги.КодАвтора = Авторы.КодАвтора WHERE Авторы.Фамилия = 'Петров' AND  
Авторы.Имя = 'Александр')")  
conn.commit()
```

```
# SQL-запросы на удаление данных из БД  
print("\nSQL-запросы на удаление данных из БД:")
```

```
print("\n1. Удалить все книги, опубликованные в заданном году:")  
cursor.execute("DELETE FROM Книги WHERE ГодИздания = 2021")  
conn.commit()
```

```
print("\n2. Удалить всех авторов, написавших книги для заданного издательства:")  
cursor.execute("DELETE FROM Авторы WHERE КодАвтора IN (SELECT КодАвтора FROM  
АвторКниги INNER JOIN Книги ON АвторКниги.КодКниги = Книги.КодКниги WHERE  
Книги.Издательство = 'Издательство 1')")  
conn.commit()
```

```
# SQL-запросы на обновление данных в БД
```

```
print("\n4. Обновить название всех книг, которые были опубликованы в годы с 2010 по  
2015 включительно, установив название 'Старое название':")  
cursor.execute("UPDATE Книги SET Название = 'Старое название' WHERE ГодИздания  
BETWEEN 2010 AND 2015")  
conn.commit()
```

```
print("\n5. Обновить место хранения всех книг, написанных автором с кодом 7,  
установив место хранения 'Библиотека №2':")
```

```
cursor.execute("UPDATE Книги SET МестоХранения = 'Библиотека №2' WHERE  
КодКниги IN (SELECT КодКниги FROM АвторКниги WHERE КодАвтора = 7)")  
conn.commit()
```

```
print("\n6. Обновление города из таблицы Издательства по коду города в таблице  
Книги:")  
cursor.execute("UPDATE Издательства SET Город = (SELECT DISTINCT Город FROM Книги  
WHERE Книги.Издательство = Издательства.Издательство)")  
conn.commit()
```

```
print("\n7. Обновление кода автора в таблице АвторКниги по коду автора в таблице  
Авторы:")  
cursor.execute("UPDATE АвторКниги SET КодАвтора = (SELECT DISTINCT КодАвтора  
FROM Авторы WHERE Авторы.КодАвтора = АвторКниги.КодАвтора)")  
conn.commit()
```

```
print("\n8. Обновление названия раздела в таблице Книги по названию раздела в  
таблице Разделы:")  
cursor.execute("UPDATE Книги SET Раздел = (SELECT DISTINCT Раздел FROM Разделы  
WHERE Разделы.Раздел = Книги.Раздел)")  
conn.commit()
```

```
print("\n9. Обновление года издания в таблице Книги по году издания в таблице  
АвторКниги:")  
cursor.execute("UPDATE Книги SET ГодИздания = (SELECT DISTINCT ГодИздания FROM  
АвторКниги WHERE АвторКниги.КодКниги = Книги.КодКниги)")  
conn.commit()
```

```
print("\n10. Обновление места хранения в таблице Книги по названию издательства в  
таблице Издательства:")  
cursor.execute("UPDATE Книги SET МестоХранения = (SELECT DISTINCT  
Издательства.Город FROM Издательства WHERE Издательства.Издательство =  
Книги.Издательство)")  
conn.commit()
```

```
print("\n11. Обновление фамилии автора в таблице Авторы по коду автора в таблице  
АвторКниги:")  
cursor.execute("UPDATE Авторы SET Фамилия = (SELECT DISTINCT Фамилия FROM  
АвторКниги WHERE АвторКниги.КодАвтора = Авторы.КодАвтора)")  
conn.commit()
```

```
print("\n12. Обновить год издания всех книг, изданных в городе 'Москва', на 2022 год.")  
cursor.execute("UPDATE Книги SET ГодИздания = 2022 WHERE МестоХранения =  
'Москва'")  
conn.commit()
```

```
print("\n13. Обновить место хранения всех книг, написанных автором с фамилией  
'Иванов', на 'Книжный шкаф 1'.")  
cursor.execute("UPDATE Книги SET МестоХранения = 'Книжный шкаф 1' WHERE  
КодКниги IN (SELECT КодКниги FROM АвторКниги INNER JOIN Авторы ON  
АвторКниги.КодАвтора = Авторы.КодАвтора WHERE Авторы.Фамилия = 'Иванов')")  
conn.commit()
```

```
print("\n14. Обновить год издания всех книг, написанных автором с именем 'Анна', на  
2023 год.")
```

```
cursor.execute("UPDATE Книги SET ГодИздания = 2023 WHERE КодКниги IN (SELECT  
КодКниги FROM АвторКниги INNER JOIN Авторы ON АвторКниги.КодАвтора =  
Авторы.КодАвтора WHERE Авторы.Имя = 'Анна')")  
conn.commit()
```

```
print("\n15. Обновить название раздела всех книг, изданных в городе 'Санкт-  
Петербург', на 'Классика'.")
```

```
cursor.execute("UPDATE Книги SET Раздел = 'Классика' WHERE КодКниги IN (SELECT  
КодКниги FROM Издательства INNER JOIN Книги ON Издательства.Издательство =  
Книги.Издательство WHERE Издательства.Город = 'Санкт-Петербург')")  
conn.commit()
```

```
print("\n16. Обновить год издания всех книг, написанных автором с фамилией  
'Петров', на 2024 год.")
```

```
cursor.execute("UPDATE Книги SET ГодИздания = 2024 WHERE КодКниги IN (SELECT  
КодКниги FROM АвторКниги INNER JOIN Авторы ON АвторКниги.КодАвтора =  
Авторы.КодАвтора WHERE Авторы.Фамилия = 'Петров')")  
conn.commit()
```

```
# Закрываем соединение с базой данных  
cursor.close()  
conn.close()
```

**Протокол работы программы:**

**1. Получить список всех книг, отсортированных по году издания:**

```
[(4, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (5, 'Книга 1', 'Раздел 1',  
'Издательство 1', None, 'Город 1'), (6, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город  
2'), (7, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (8, 'Книга 2', 'Раздел 2',  
'Издательство 2', None, 'Город 2'), (9, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город  
1'), (10, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (11, 'Книга 1', 'Раздел 1',  
'Издательство 1', None, 'Город 1'), (12, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город  
2'), (13, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (14, 'Книга 2', 'Раздел 2',  
'Издательство 2', None, 'Город 2'), (15, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город  
1'), (16, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (17, 'Книга 1', 'Раздел 1',  
'Издательство 1', None, 'Город 1'), (18, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город  
2'), (19, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (20, 'Книга 2', 'Раздел 2',  
'Издательство 2', None, 'Город 2'), (22, 'Книга 2', 'Раздел 2', 'Издательство 2', 2018, 'Место  
2'), (3, 'Книга 1', 'Раздел 1', 'Издательство 1', 2020, 'Город 1'), (21, 'Книга 1', 'Раздел 1',  
'Издательство 1', 2020, 'Место 1'), (23, 'Книга 3', 'Раздел 1', 'Издательство 1', 2021, 'Место  
3'), (1, 'Книга 1', 'Раздел 1', 'Издательство 1', 2022, 'Город 1'), (2, 'Книга 2', 'Раздел 2',  
'Издательство 2', 2024, 'Город 2')]
```

**2. Получить список всех книг заданного автора:**

```
[]
```

**3. Получить список всех книг из заданного раздела:**

```
[(1, 'Книга 1', 'Раздел 1', 'Издательство 1', 2022, 'Город 1'), (3, 'Книга 1', 'Раздел 1',  
'Издательство 1', 2020, 'Город 1'), (5, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город  
1'), (7, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (9, 'Книга 1', 'Раздел 1',  
'Издательство 1', None, 'Город 1'), (11, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город  
1'), (13, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (15, 'Книга 1', 'Раздел 1',
```

'Издательство 1', None, 'Город 1'), (17, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (19, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (21, 'Книга 1', 'Раздел 1', 'Издательство 1', 2020, 'Место 1'), (23, 'Книга 3', 'Раздел 1', 'Издательство 1', 2021, 'Место 3')]]

4. Получить список всех книг, изданных заданным издательством:

[(1, 'Книга 1', 'Раздел 1', 'Издательство 1', 2022, 'Город 1'), (3, 'Книга 1', 'Раздел 1', 'Издательство 1', 2020, 'Город 1'), (5, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (7, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (9, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (11, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (13, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (15, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (17, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (19, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (21, 'Книга 1', 'Раздел 1', 'Издательство 1', 2020, 'Место 1'), (23, 'Книга 3', 'Раздел 1', 'Издательство 1', 2021, 'Место 3')]]

5. Получить список всех авторов в алфавитном порядке:

[(4, None, 'Иван'), (7, None, 'Иван'), (10, None, 'Иван'), (13, None, 'Иван'), (16, None, 'Иван'), (19, None, 'Иван'), (22, None, 'Иван'), (25, None, 'Иван'), (28, None, 'Иван'), (5, None, 'Петр'), (8, None, 'Петр'), (11, None, 'Петр'), (14, None, 'Петр'), (17, None, 'Петр'), (20, None, 'Петр'), (23, None, 'Петр'), (26, None, 'Петр'), (29, None, 'Петр'), (6, None, 'Сидор'), (9, None, 'Сидор'), (12, None, 'Сидор'), (15, None, 'Сидор'), (18, None, 'Сидор'), (21, None, 'Сидор'), (24, None, 'Сидор'), (27, None, 'Сидор'), (30, None, 'Сидор'), (31, 'Иванов', 'Иван'), (2, 'Петров', 'Петр'), (32, 'Петров', 'Петр'), (33, 'Сидоров', 'Сидор')]

6. Получить список всех книг, отсортированных по названию и году издания:

[(5, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (7, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (9, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (11, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (13, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (15, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (17, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (19, 'Книга 1', 'Раздел 1', 'Издательство 1', None, 'Город 1'), (3, 'Книга 1', 'Раздел 1', 'Издательство 1', 2020, 'Город 1'), (21, 'Книга 1', 'Раздел 1', 'Издательство 1', 2020, 'Место 1'), (1, 'Книга 1', 'Раздел 1', 'Издательство 1', 2022, 'Город 1'), (4, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (6, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (8, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (10, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (12, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (14, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (16, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (18, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (20, 'Книга 2', 'Раздел 2', 'Издательство 2', None, 'Город 2'), (22, 'Книга 2', 'Раздел 2', 'Издательство 2', 2018, 'Место 2'), (2, 'Книга 2', 'Раздел 2', 'Издательство 2', 2024, 'Город 2'), (23, 'Книга 3', 'Раздел 1', 'Издательство 1', 2021, 'Место 3')]]

7. Получить список всех книг заданного автора, отсортированных по году издания:

[]

8. Получить список всех книг, опубликованных в заданном году:

[(23, 'Книга 3', 'Раздел 1', 'Издательство 1', 2021, 'Место 3')]

**9. Получить список всех авторов, написавших книги для заданного издательства:**

**□**

**10. Получить список всех книг, в названии которых есть заданное слово:**

**□**

**SQL-запросы на обновление данных в БД:**

**1. Обновить год издания всех книг, написанных автором с фамилией 'Иванов', установив год издания равным 2022:**

**2. Обновить название и год издания книги, хранящейся в городе 'Москва', установив название 'Новая книга' и год издания равным 2023:**

**3. Обновить название и раздел всех книг, написанных автором с именем 'Александр' и фамилией 'Петров', установив название 'Новое название' и раздел 'Фантастика':**

**SQL-запросы на удаление данных из БД:**

**1. Удалить все книги, опубликованные в заданном году:**

**2. Удалить всех авторов, написавших книги для заданного издательства:**

**4. Обновить название всех книг, которые были опубликованы в годы с 2010 по 2015 включительно, установив название 'Старое название':**

**5. Обновить место хранения всех книг, написанных автором с кодом 7, установив место хранения 'Библиотека №2':**

**6. Обновление города из таблицы Издательства по коду города в таблице Книги:**

**7. Обновление кода автора в таблице АвторКниги по коду автора в таблице Авторы:**

**8. Обновление названия раздела в таблице Книги по названию раздела в таблице Разделы:**

**9. Обновление года издания в таблице Книги по году издания в таблице АвторКниги:**

**10. Обновление места хранения в таблице Книги по названию издательства в таблице Издательства:**

**11. Обновление фамилии автора в таблице Авторы по коду автора в таблице АвторКниги:**



**12. Обновить год издания всех книг, изданных в городе 'Москва', на 2022 год.**

**13. Обновить место хранения всех книг, написанных автором с фамилией 'Иванов', на 'Книжный шкаф 1'.**

**14. Обновить год издания всех книг, написанных автором с именем 'Анна', на 2023 год.**

**15. Обновить название раздела всех книг, изданных в городе 'Санкт-Петербург', на 'Классика'.**

**16. Обновить год издания всех книг, написанных автором с фамилией 'Петров', на 2024 год.**

**Вывод:** В процессе выполнения практического занятия выработал приобрести навыки составление программ для работы с базами данных в IDE PyCharm Community.