МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт комплексной безопасности и цифровых технологий (ИКБ)
Кафедра КБ-14 «Цифровые технологии обработки данных»

Администрирование баз данных

Практическая работа №1

Выполнили студенты 3 курса

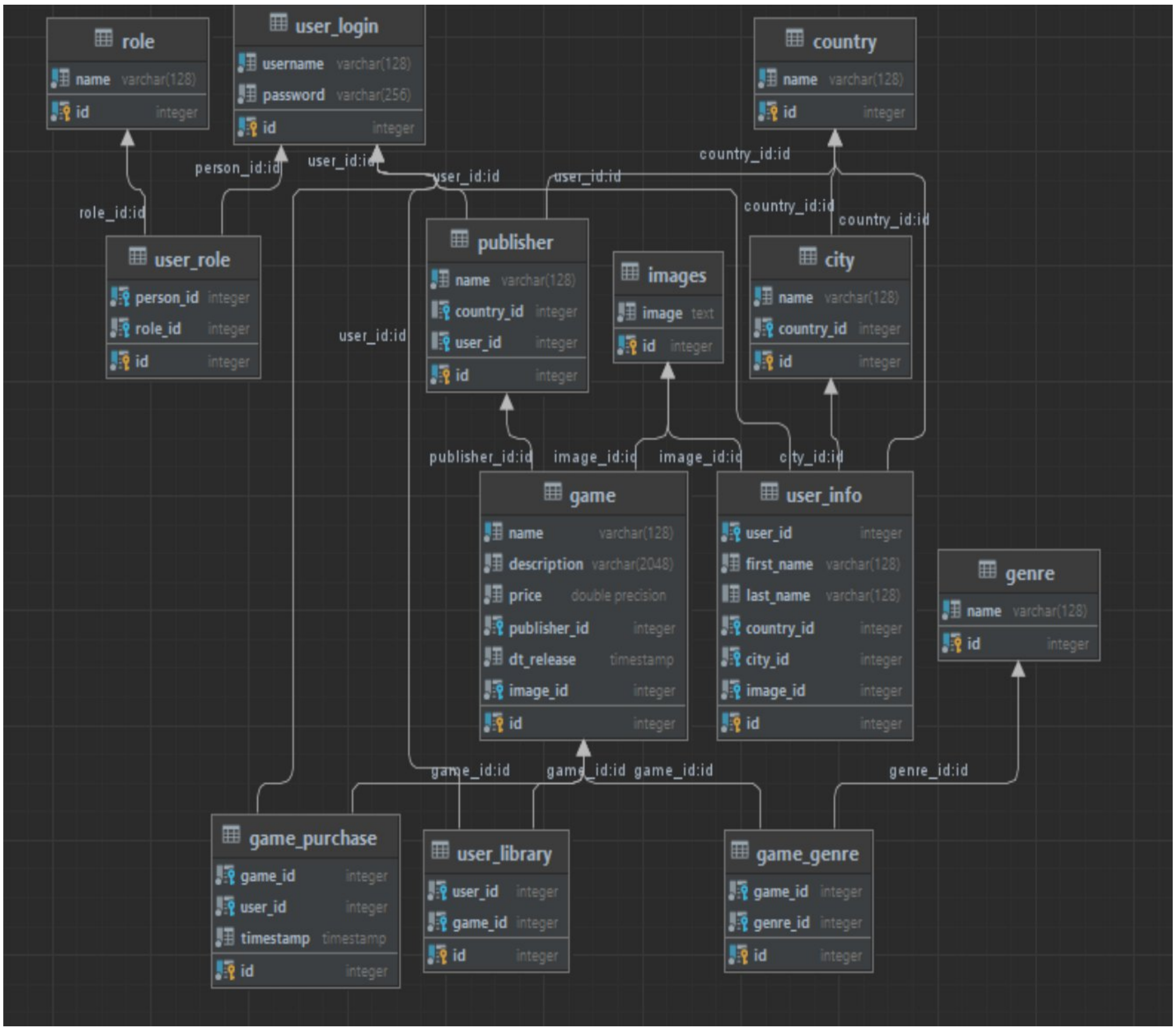
Зарин Н.Н.

Луговой И.И.

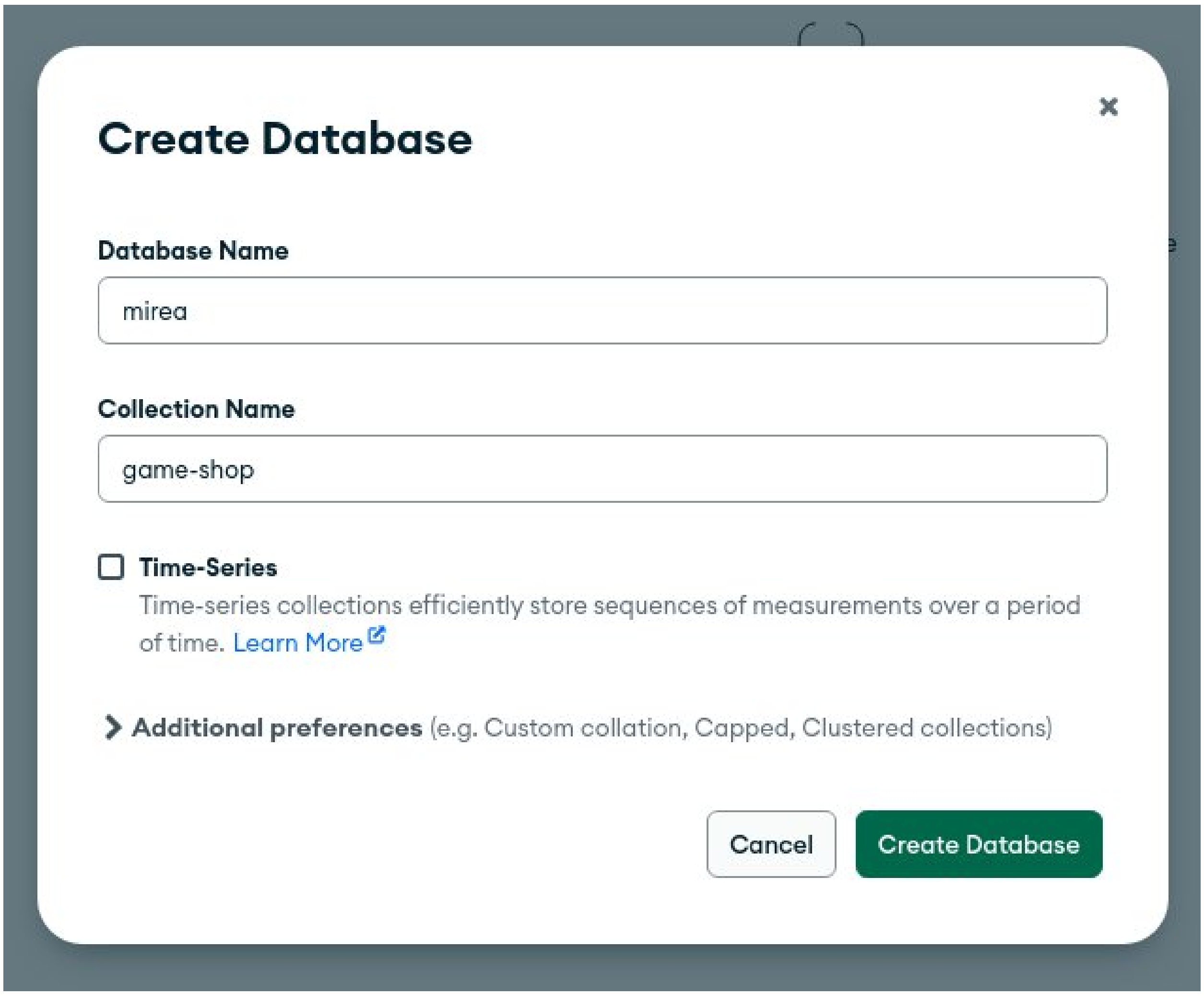
БСБО-05-20

Москва 2023

В качестве темы для БД был взят интернет-магазин игр, который использовался на прошлом курсе для CRM-системы.



Создание БД в MongoDB Compass


Create Database

Database Name: mirea

Collection Name: game-shop

Time-Series [ ]
Time-series collections efficiently store sequences of measurements over a period of time. Learn More

×
Additional preferences (e.g. Custom collation, Capped, Clustered collections)
Cancel
Create Database

Создание коллекций

Категории

```
db.createCollection(
    "categories", {
        validator: {
            $jsonSchema: {
                bsonType: "object",
                title: "Category validation",
                required: [
                    "name"
                ],
                properties: {
                    name: {
                        bsonType: "string"
                    }
                }
            }
        }
    }
)
```

Игры

```
db.createCollection(
    "games", {
        validator: {
            $jsonSchema: {
                bsonType:"object",
                title: "Game validation",
                required: [
                    "name",
                    "amount",
                    "price",
                    "publisher",
                    "category"
                ],
                properties: {
                    name: {
                        bsonType:"string"
                    },
                    amount: {
                        bsonType: "int"
                    },
                    price: {
                        bsonType: "int"
                    },
                    publisher: {
```

```
                    bsonType: "string"
                },
                category: {
                    bsonType: "objectId"
                }
            }
        }
    }
}
)
```

Работники

```
db.createCollection(
    "workers", {
        validator: {
            $jsonSchema: {
                bsonType: "object",
                title: "Worker validation",
                required: [
                    "name",
                    "phone",
                    "email",
                    "position",
                ],
                properties: {
                    name: {
                        bsonType: "string"
                    },
                    position: {
                        bsonType: "string"
                    },
                    phone: {
                        bsonType: "string"
                    },
                    email: {
                        bsonType: "string"
                    }
                }
            }
        }
    }
)
```

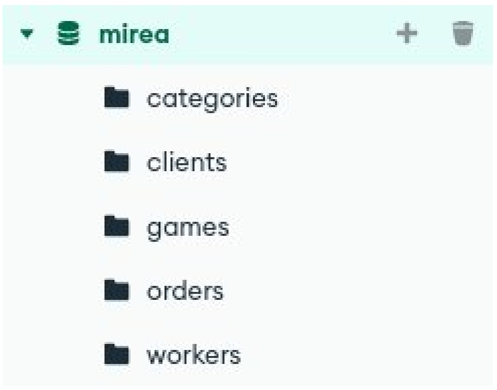
Клиенты

```
db.createCollection(
    "clients", {
        validator: {
            $jsonSchema: {
                bsonType: "object",
                title: "Client validation",
                required: [
                    "nickname",
                    "email"
                ],
                properties: {
                    nickname: {
                        bsonType: "string"
                    },
                    email: {
                        bsonType: "string"
                    }
                }
            }
        }
    }
)
```

Заказы

```
db.createCollection(
    "orders", {
        validator: {
            $jsonSchema: {
                bsonType: "object",
                title: "Order validation",
                required: [
                    "client",
                    "games",
                    "date",
                    "status"
                ],
                properties: {
                    client: {
                        bsonType: "objectId"
                    },
                    games: {
                        bsonType: "array",
                        items: {
                            required: [
                                "game"
                            ],
                            properties: {
                                game: {
                                    bsonType: "objectId"
                                }
                            }
                        }
                    },
                    date: {
                        bsonType: "date"
                    },
                    status: {
                        bsonType: "bool"
                    }
                }
            }
        }
    }
)
```

БД


mirea

categories

clients

games

orders

workers

Заполнение данными

```
db.categories.insertMany([
   {
      "name": "Шутер"
   },
   {
      "name": "Приключение"
   },
   {
      "name": "Хоррор"
   },
   {
      "name": "Файтинг"
   },
   {
      "name": "Фентези"
   },
   {
      "name": "Гонки"
   },
   {
      "name": "Головоломки"
   },
   {
      "name": "Детектив"
   },
   {
      "name": "Аниме"
   },
   {
      "name": "Симулятор"
   },
   {
      "name": "Экшн"
   }
])
db.clients.insertMany([
   {
      "nickname": "tawer228",
      "email": "tawer228@ya.ru"
   },
   {
      "nickname": "keka564",
      "email": "fgh3@gmail.com"
   },
   {
      "nickname": "gus",
      "email": "gusein22@bk.com"
```

```
    },
    {
        "nickname": "walter1",
        "email": "ww563@gmail.com"
    },
    {
        "nickname": "baltika7",
        "email": "pivolublu@mail.ru"
    }
])
db.workers.insertMany([
    {
        "name": "Жыксамонов Акылбек",
        "phone": "+79037484847",
        "email": "zhick_a@gmail.com",
        "position": "Старший модератор"
    },
    {
        "name": "Иванов Иван",
        "phone": "+79154862915",
        "email": "ivaaaaaaaaaaan@ya.ru",
        "position": "Модератор"
    },
    {
        "name": "Зауров Владислав",
        "phone": "+79037583916",
        "email": "vlad1ck_z@gmail.com",
        "position": "Разработчик"
    },
])
db.games.insertMany([
    {
        "name": "Rocket League",
        "price": 0,
        "publisher": "Epic Games",
        "categories": [
            {
                "category": db.categories.findOne({"name": "Гонки"})["_id"],
            }
        ]
    },
    {
        "name": "Battlefield 5",
        "price": 2499,
        "publisher": "EA",
        "categories": [
            {
                "category": db.categories.findOne({"name": "Шутер"})["_id"],
            }
        ]
    },
    {
        "name": "Uncharted 4",
        "price": 2999,
        "publisher": "Sony",
        "categories": [
            {
                "category": db.categories.findOne({"name": "Экшн"})["_id"],
            },
            {
                "category": db.categories.findOne({"name": "Шутер"})["_id"],
            },
            {
                "category": db.categories.findOne({"name": "Приключение"})["_id"],
            }
        ]
    },
    {
        "name": "Resident Evil Village",
        "price": 1499,
        "publisher": "Capcom",
        "categories": [
            {
                "category": db.categories.findOne({"name": "Хоррор"})["_id"],
            },
            {
                "category": db.categories.findOne({"name": "Приключение"})["_id"],
            }
        ]
    },
    {
        "name": "Assetto Corsa",
        "price": 799,
        "publisher": "Kunos Simulazioni",
        "categories": [
            {
                "category": db.categories.findOne({"name": "Гонки"})["_id"]
            },
            {
                "category": db.categories.findOne({"name": "Симулятор"})["_id"]
            }
        ]
    }
])
db.orders.insertMany([
    {
```

```
    "client": db.clients.findOne({"email": "gusein22@bk.com"})["_id"],
    "games": [
        {
            "game": db.games.findOne({"name": "Resident Evil Village"})["_id"]
        }
    ],
    "date": new ISODate(),
    "status": true,
},
{
    "client": db.clients.findOne({"email": "tawer228@ya.ru"})["_id"],
    "games": [
        {
            "game": db.games.findOne({"name": "Rocket League"})["_id"]
        }
    ],
    "date": new ISODate(),
    "status": true,
},
{
    "client": db.clients.findOne({"email": "pivolublu@mail.ru"})["_id"],
    "games": [
        {
            "game": db.games.findOne({"name": "Battlefield 5"})["_id"]
        }
    ],
    "date": new ISODate(),
    "status": false,
},
{
    "client": db.clients.findOne({"email": "ww563@gmail.com"})["_id"],
    "games": [
        {
            "game": db.games.findOne({"name": "Assetto Corsa"})["_id"]
        }
    ],
    "date": new ISODate(),
    "status": true,
},
{
    "client": db.clients.findOne({"email": "fgh3@gmail.com"})["_id"],
    "games": [
        {
            "game": db.games.findOne({"name": "Uncharted 4"})["_id"]
        }
    ],
    "date": new ISODate(),
    "status": true,
    },
])
```

Реализовать следующие запросы к созданной БД:

1. Получение списка всех категорий

```
db.categories.find()
```

2. Получение списка всех продуктов в категории

```
db.games.find({"categories.category": db.categories.findOne({name: "Гонки"})["_id"]})
```

3. Поиск продукта по названию

```
db.games.findOne({"name": "Rocket League"})
```

4. Добавление продукта в корзину клиента

```
db.orders.insertOne(
{
        "client": db.clients.findOne({"email": "pivolublu@mail.ru"})["_id"],
        "games": [
            {
                "game": db.games.findOne({"name": "Assetto Corsa"})["_id"]
            }
        ],
        "date": new ISODate(),
        "status": false,
})
```

5. Получение списка всех заказов клиента

```
db.orders.find({"client": db.clients.findOne({"email": "pivolublu@mail.ru"})["_id"]})
```

6. Обновление статуса заказа

```
db.orders.updateOne({"_id": ObjectId("647b8163d9ca604b34de0474")}, {"$set":
{"status": true}})
```

7. Получение списка топ-продаж за последние месяцы с учетом цены и количества проданных товаров

```
db.orders.aggregate([
  {
    $match: {
      date: {
        $gte: new ISODate('2023-06-03')
      }
    }
  },
  {
    $unwind: "$games"
  },
  {
    $lookup:
      {
        from: "games",
        localField: "games.game",
        foreignField: "_id",
        as: "games"
      }
  },
  {
    $unwind: "$games"
  },
  {
    $group:
     {
       _id: "$games.name",
       total_sum: {
          $sum: "$games.price"
       },
     }
  },
  {
    $sort: {
     total_sum: -1
    }
  }
])
```

8. Получение списка клиентов, которые сделали более чем N покупок в последнее время

```
db.orders.aggregate([
  {
    $match: {
     date: {
        $gte: new ISODate('2023-06-01')
     }
    }
  },
  {
    $lookup:
      {
        from: "clients",
        localField: "client",
        foreignField: "_id",
        as: "clients"
      }
  },
  {
    $group:
     {
       _id: "$clients._id",
       orders_count: {
          $count: {}
       }
```

```
        }
    },
    {
     $match: {
        orders_count: {
           $gt: 1
        }
     }
    },
    {
      $sort: {
        orders_count: -1
      }
    }
])
```

9. Получите какие категории товаров пользовались спросом в заданный срок

```
db.orders.aggregate(
    {
      $match: {
        date: {
           $gte: new ISODate('2023-06-01'),
           $lte: new ISODate('2023-06-05')
        }
      }
    },
    {
      $unwind: "$games"
    },
    {
        $lookup: {
           from: "games",
           localField: "games.game",
           foreignField: "_id",
           as: "game"
        }
    },
    {
      $unwind: "$games"
    },
    {
      $lookup:
      {
        from: "categories",
        localField: "game.category",
        foreignField: "_id",
        as: "categories"
      }
    },
  {
    $group:
     {
        _id: "$categories.name"
     }
  }
)
```

10. Какие товары не были проданы в какую-то дату

```
db.orders.aggregate(
   {
    $match: {
     date: {
        $gte: new ISODate('2023-06-01'),
        $lte: new ISODate('2023-06-05')
     }
    }
  },
  {
    $unwind: "$games"
  },
  {
      $lookup: {
         from: "games",
         localField: "game",
         foreignField: "_id",
         as: "games"
      }
  },
  {
    $unwind: "$games"
  },
  {
    $unset: "games"
  }
)
```

Создание пользователей для БД:

Роль администратора

```
db.createRole({
   role: "admin",
   privileges: [
      {
         resource: {db: "mirea", collection: "categories"},
         actions: ["find", "insert", "update", "remove"]
      },
      {
         resource: {db: "mirea", collection: "games"},
         actions: ["find", "insert", "update", "remove"]
      },
      {
         resource: {db: "mirea", collection: "workers"},
         actions: ["find", "insert", "update", "remove"]
      },
      {
         resource: {db: "mirea", collection: "clients"},
         actions: ["find", "insert", "update", "remove"]
      },
      {
         resource: {db: "mirea", collection: "orders"},
         actions: ["find", "insert", "update", "remove"]
      }
   ],
   roles: []
})
```

Роль менеджера

```
db.createRole({
   role: "manager",
   privileges: [
      {
         resource: {db: "mirea", collection: "categories"},
         actions: ["find", "insert", "update", "remove"]
      },
      {
         resource: {db: "mirea", collection: "games"},
         actions: ["find", "insert", "update", "remove"]
      },
      {
         resource: {db: "mirea", collection: "orders"},
         actions: ["find", "insert", "update", "remove"]
      }
   ],
   roles: []
})
```

Роль пользователя

```
db.createRole({
    role: "user",
    privileges: [
        {
            resource: {db: "mirea", collection: "games"},
            actions: ["find"]
        },
        {
            resource: {db: "mirea", collection: "orders"},
            actions: ["find", "insert"]
        }
    ],
    roles: []
})
```

Роль гостя

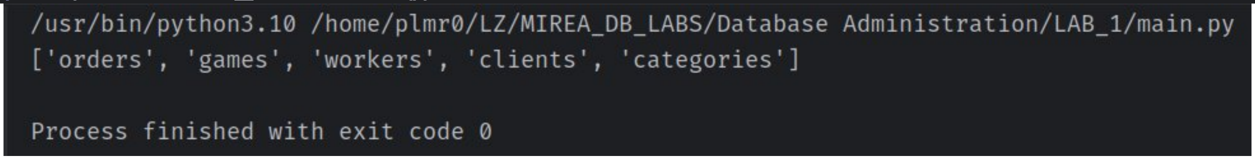
```
db.createRole({
    role: "guest",
    privileges: [
        {
            resource: {db: "mirea", collection: "games"},
            actions: ["find"]
        },
    ],
    roles: []
})
```

Реализация программы на языке Python:

```python
from pymongo import MongoClient
from datetime import datetime


class Client:
    def __init__(self):
        self.client = MongoClient('localhost', 27017)
        self.db = self.client['mirea']

    # Все коллекции
    def show_collections(self):
        return self.db.list_collection_names()

    # Все категории
    def show_categories(self):
        categories_collection = self.db['categories']
        return [category['name'] for category in categories_collection.find()]

    # Поиск игры по названию
    def find_games_by_name(self, name):
        games_collection = self.db['games']
        cursor = games_collection.find({'name': {'$regex': name, '$options': 'i'}})
        games = [game['name'] for game in cursor]
        return games

    # Поиск игр по категории
    def find_games_by_category(self, category):
        category_collection = self.db['categories']
        game_collection = self.db['games']
        category_games = game_collection.find({'categories.category':
category_collection.find_one({'name': category})["_id"]})
        games = [game['name'] for game in category_games]
        return games

    # Поиск игр по издателю
    def find_games_by_publisher(self, publisher):
        games_collection = self.db['games']
        cursor = games_collection.find({'publisher': {'$regex': publisher, '$options': 'i'}})
        games = [game['name'] for game in cursor]
        return games

    # Поиск игр в заданном ценовом диапазоне
    def find_games_by_price(self, min_price, max_price):
        games_collection = self.db['games']
        cursor = games_collection.find({'price': {'$gte': min_price, '$lte': max_price}})
        games = [game['name'] for game in cursor]
        return games
    # Добавить игры в корзину
    def add_games_to_order(self, client_email, *args):
        _id = self.db.orders.insert_one(
            {
                "client": self.db.clients.find_one({"email": client_email})["_id"],
                "games": [
                    {
                        "game": self.db.games.find_one({"name": game})["_id"]
                    }
                    for game in args
                ],
                "date": datetime.now(),
                "status": False,
            })
        return _id.inserted_id

    # Общая стоимость корзины
    def calculate_cart_total(self, *args):
        total = 0
        games_collection = self.db['games']
        for name in args:
            game = games_collection.find_one({'name': name})
            total += game['price']
        return total
```

Вывод всех коллекций:

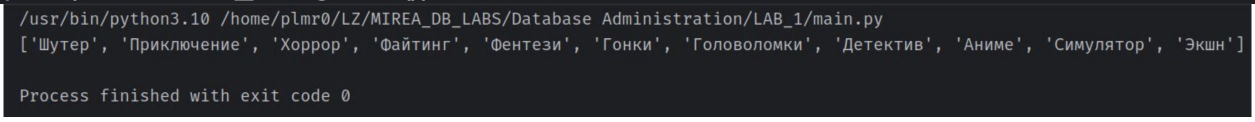
```python
client = Client()
print(client.show_collections())
```



```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
['orders', 'games', 'workers', 'clients', 'categories']

Process finished with exit code 0
```

Вывод всех категорий:

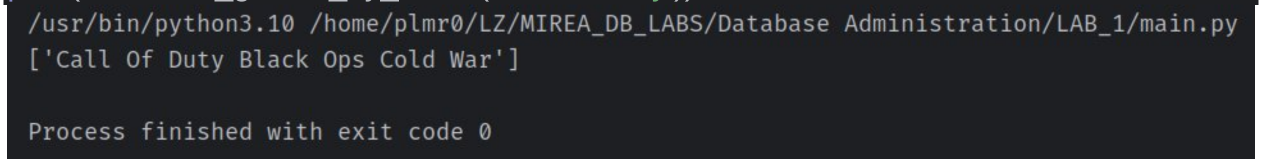
```python
client = Client()
print(client.show_categories())
```



```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
['Шутер', 'Приключение', 'Хоррор', 'Файтинг', 'Фентези', 'Гонки', 'Головоломки', 'Детектив', 'Аниме', 'Симулятор', 'Экшн']

Process finished with exit code 0
```

Поиск игр по названию:

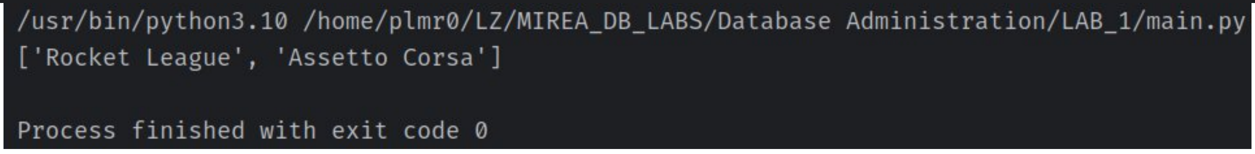
```python
client = Client()
print(client.find_games_by_name('Call Of Duty'))
```



```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
['Call Of Duty Black Ops Cold War']

Process finished with exit code 0
```

Поиск игр по категории:

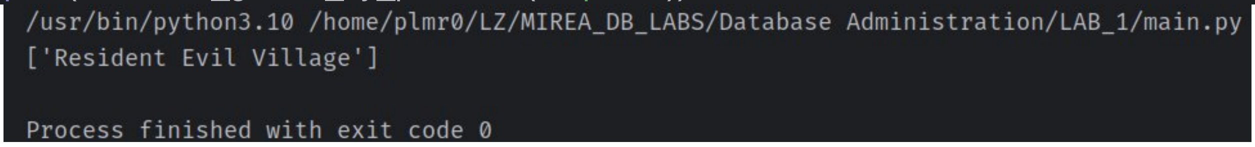
```
client = Client()
print(client.find_games_by_category('Гонки'))
```



```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
['Rocket League', 'Assetto Corsa']

Process finished with exit code 0
```

Поиск игр по издателю:

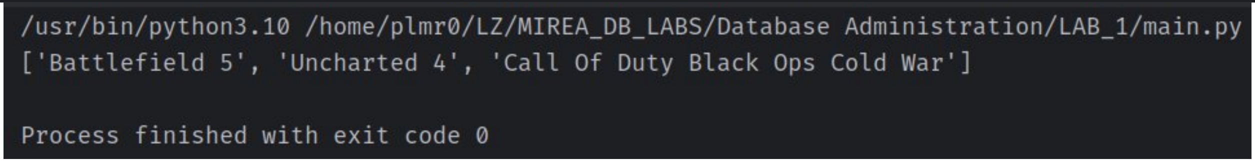
```
client = Client()
print(client.find_games_by_publisher('Capcom'))
```



```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
['Resident Evil Village']

Process finished with exit code 0
```

Поиск игр в заданном ценовом диапазоне:

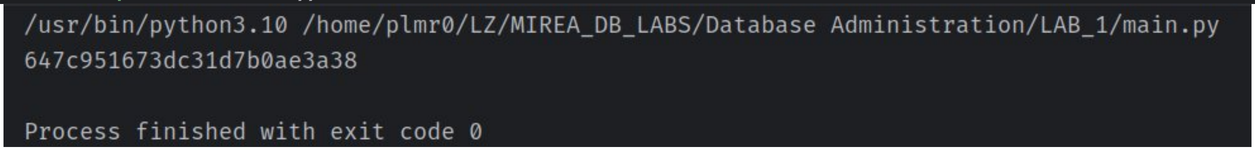
```
client = Client()
print(client.find_games_by_price(1999, 4999))
```



```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
['Battlefield 5', 'Uncharted 4', 'Call Of Duty Black Ops Cold War']

Process finished with exit code 0
```

Добавление игр в корзину:

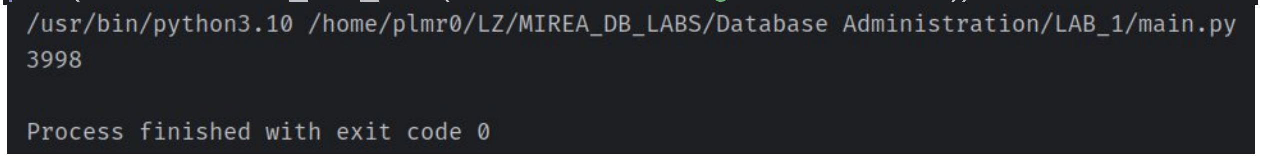
```
client = Client()
print(client.add_games_to_order('pivolublu@mail.ru', 'Rocket League', 'Call Of Duty
Black Ops Cold War'))
```



```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
647c951673dc31d7b0ae3a38

Process finished with exit code 0
```

Подсчет общей стоимости корзины:

```
client = Client()
print(client.calculate_cart_total('Resident Evil Village', 'Battlefield 5'))
```

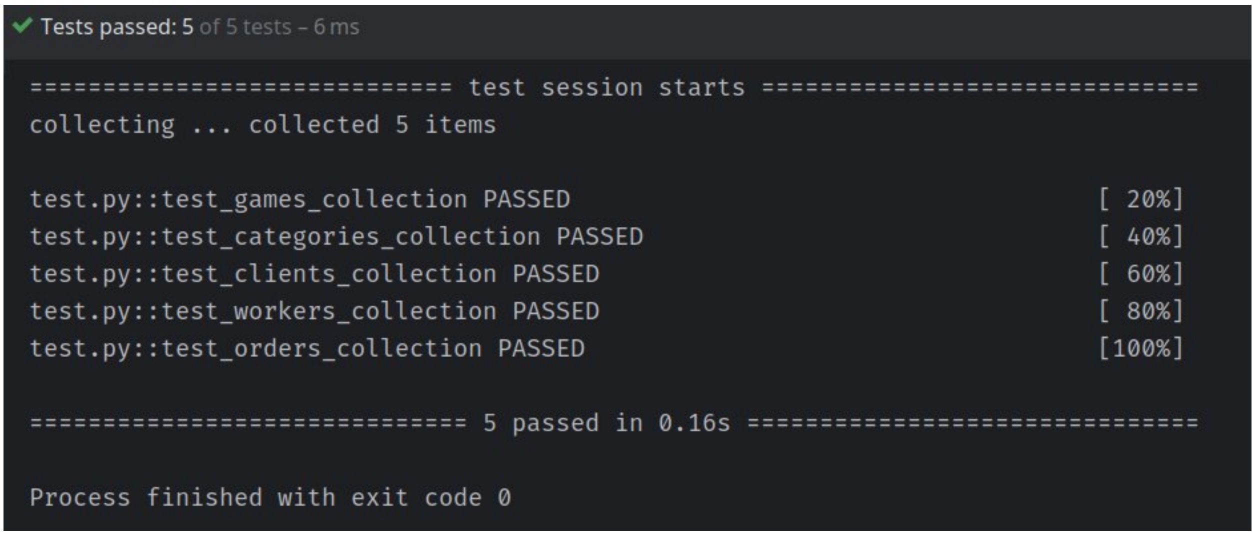


```
/usr/bin/python3.10 /home/plmr0/LZ/MIREA_DB_LABS/Database Administration/LAB_1/main.py
3998

Process finished with exit code 0
```

Тесты для проверки схемы базы данных:

```python
import pymongo

client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["mirea"]


def test_games_collection():
    games = db["games"]

    assert games is not None
    assert "name" in games.find_one()
    assert "price" in games.find_one()
    assert "publisher" in games.find_one()
    assert "categories" in games.find_one()


def test_categories_collection():
    categories = db["categories"]

    assert categories is not None
    assert "name" in categories.find_one()


def test_clients_collection():
    clients = db["clients"]

    assert clients is not None
    assert "nickname" in clients.find_one()
    assert "email" in clients.find_one()


def test_workers_collection():
    workers = db["workers"]

    assert workers is not None
    assert "name" in workers.find_one()
    assert "email" in workers.find_one()
    assert "phone" in workers.find_one()
    assert "position" in workers.find_one()


def test_orders_collection():
    orders = db["orders"]

    assert orders is not None
    assert "client" in orders.find_one()
    assert "games" in orders.find_one()
    assert "date" in orders.find_one()
    assert "status" in orders.find_one()
```

Результат тестов:



✓ Tests passed: 5 of 5 tests – 6 ms

```
============================ test session starts ==============================
collecting ... collected 5 items

test.py::test_games_collection PASSED                                    [ 20%]
test.py::test_categories_collection PASSED                               [ 40%]
test.py::test_clients_collection PASSED                                  [ 60%]
test.py::test_workers_collection PASSED                                  [ 80%]
test.py::test_orders_collection PASSED                                   [100%]

============================== 5 passed in 0.16s ===============================

Process finished with exit code 0
```