Лабораторная работа № 13

Тема: Проектирование моделей данных в MongoDB.

Цель: Закрепить знания о моделях данных в MongoDB и научиться применять встроенные, нормализованные и гибридные модели данных на практике.

Задание:

1. Встроенная модель данных.
Создайте коллекцию для хранения информации о пользователях и их адресах.
Структура документа пользователя:
Поля: user_id, name, email.
Поле addresses должно быть массивом вложенных объектов с полями: street, city, zip.
Добавьте в коллекцию несколько документов.
Выполните запрос для получения всех пользователей, которые живут в городе "New York".
Пример документа:

```
{
   "user_id": 1,
   "name": "John Doe",
   "email": "john.doe@example.com",
   "addresses": [
      { "street": "123 Main St", "city": "New York", "zip": "10001" },
      { "street": "456 Oak St", "city": "Los Angeles", "zip": "90001" }
   ]
}
```

2. Нормализованная модель данных.
Создайте две коллекции: users и orders.
В коллекции users храните данные о пользователях: user_id, name, email.
В коллекции orders храните данные о заказах: order_id, user_id (ссылка на users), items (список товаров с полями name, price).
Добавьте в коллекцию orders несколько документов.
Выполните запрос, который использует $lookup, чтобы вывести все заказы определённого пользователя по его user_id.

Пример документа в коллекции users:

```
{
   "user_id": 1,
   "name": "John Doe",
   "email": "john.doe@example.com",
   "age": 30
}
```

Пример документа заказа:

```
{
   "order_id": 101,
   "user_id": 1,
   "items": [
      { "name": "Laptop", "price": 1200 },
      { "name": "Mouse", "price": 50 }
   ]
}
```

Поля:
order_id — уникальный идентификатор заказа.
user_id — идентификатор пользователя, связанный с заказом. Ссылается на пользователя из коллекции users.
order_date — дата заказа.
items — массив товаров, включающий поля:
item_id — уникальный идентификатор товара.
name — название товара.
price — цена товара.
quantity — количество единиц товара в заказе.

3. Гибридная модель данных.
Создайте коллекцию orders, где:
Информация о товарах встроена в документ заказа.
Информация о пользователе хранится в отдельной коллекции и связана через user_id.
Выполните запрос на получение заказов с включением данных о пользователе, используя $lookup.

Например:
Коллекция users:

```
{
   "user_id": 1,
   "name": "John Doe",
   "email": "john.doe@example.com",
   "age": 30
}
```

Поля:
user_id — уникальный идентификатор пользователя.
name — имя пользователя.
email — электронная почта пользователя.
age — возраст пользователя.

Коллекция orders:

```
{
   "order_id": 101,
   "user_id": 1,
   "order_date": "2024-10-01",
   "items": [
      { "item_id": 1, "name": "Laptop", "price": 1200, "quantity": 1 },
      { "item_id": 2, "name": "Mouse", "price": 50, "quantity": 2 }
   ]
}
```

Поля:
order_id — уникальный идентификатор заказа.
user_id — идентификатор пользователя, связанный с заказом (ссылается на пользователя в коллекции users).
order_date — дата заказа.
items — встроенная информация о товарах в заказе, включая поля:
item_id — уникальный идентификатор товара.
name — название товара.
price — цена товара.
quantity — количество единиц товара в заказе.
Пример запроса с объединением коллекций orders и users по полю user_id, с использованием оператора $lookup:

```
db.orders.aggregate([
   {
     $lookup: {
        from: "users",           // Коллекция, с которой мы объединяем
        localField: "user_id",    // Поле в коллекции "orders"
        foreignField: "user_id",  // Поле в коллекции "users"
        as: "user_info"           // Как назвать результирующее поле с данными пользователя
     }
   },
   {
     $unwind: "$user_info"        // Разворачиваем массив user_info для удобного отображения
   }
])
```

Описание запроса:
$lookup — объединяет коллекцию orders с коллекцией users по полю user_id.
from: "users" — коллекция, из которой будут извлечены данные (в нашем случае, это users).
localField: "user_id" — поле в коллекции orders, по которому будет выполняться соединение.
foreignField: "user_id" — поле в коллекции users, которое должно совпадать с полем user_id в коллекции orders.
as: "user_info" — результирующее поле, в котором будет храниться информация о пользователе.
$unwind — оператор, который разворачивает массив user_info в единичный объект для более удобного отображения (так как по умолчанию результат $lookup — это массив).

4. Валидация схемы.
Создайте коллекцию products с валидацией схемы:

Поле name должно быть строкой.
Поле price должно быть числом и не может быть меньше 0.
Поле category должно быть обязательным.
Попробуйте вставить корректный и некорректный документ, чтобы убедиться, что валидация работает.
Пример создания коллекции с валидацией:

```
db.createCollection("products", {
   validator: {
      $jsonSchema: {
         bsonType: "object",
         required: ["name", "price", "category"],
         properties: {
            name: {
               bsonType: "string"
            },
            price: {
               bsonType: "double",
               minimum: 0
            },
            category: {
               bsonType: "string"
            }
         }
      }
   }
})
```

Отчет должен содержать (см. образец):
- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты подтверждающие выполнение заданий.

Отчеты в формате **pdf** отправлять на email: **colledge20education23@gmail.com**