

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

“Санкт-Петербургский национальный исследовательский университет
информационных технологий механики и оптики”

Мегафакультет: трансляционных информационных технологий

Факультет: информационных технологий и программирования

Лабораторная работа №6

По дисциплине: “Проектирование баз данных”

Тема: “Создание базы данных в MongoDB”

Выполнила студент группы №М3216:

Шевцов Роман Сергеевич

САНКТ-ПЕТЕРБУРГ

2025

Задача:

Создать базу данных в MongoDB на примере схемы, сделанной в первой лабораторной работе.

Требования:

1. Необходимо взять за основу схему БД из первой лабораторной работы.
2. В основных коллекциях должно быть не менее 10 записей.

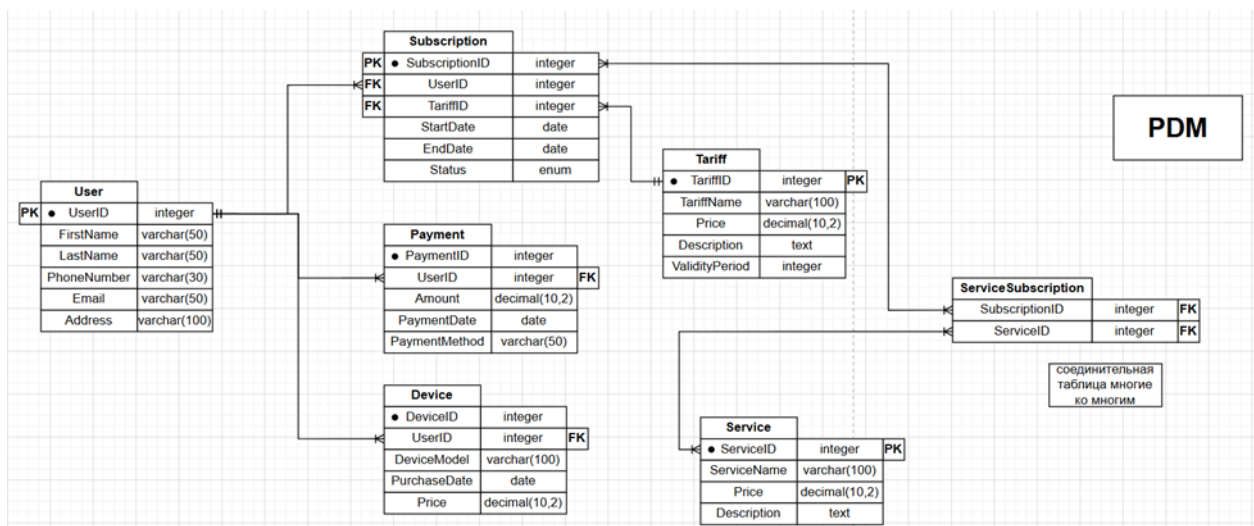
Порядок выполнения работы:

1. Проанализируйте, как будет выглядеть NoSQL база данных, содержащая те же сущности, что и реляционная модель в первой лабораторной работе.
2. Модифицируйте структуры для хранения данных, если необходимо.
3. Создайте коллекции.
4. Создайте индекс, содержащий как минимум два поля для одной из коллекций.
5. Заполните БД данными.
6. Создайте представление, которое будет в себя включать данные как минимум из двух таблиц.

На защите лабораторной необходимо будет продемонстрировать базу данных, выборки данных, индекс, представление, запросы.

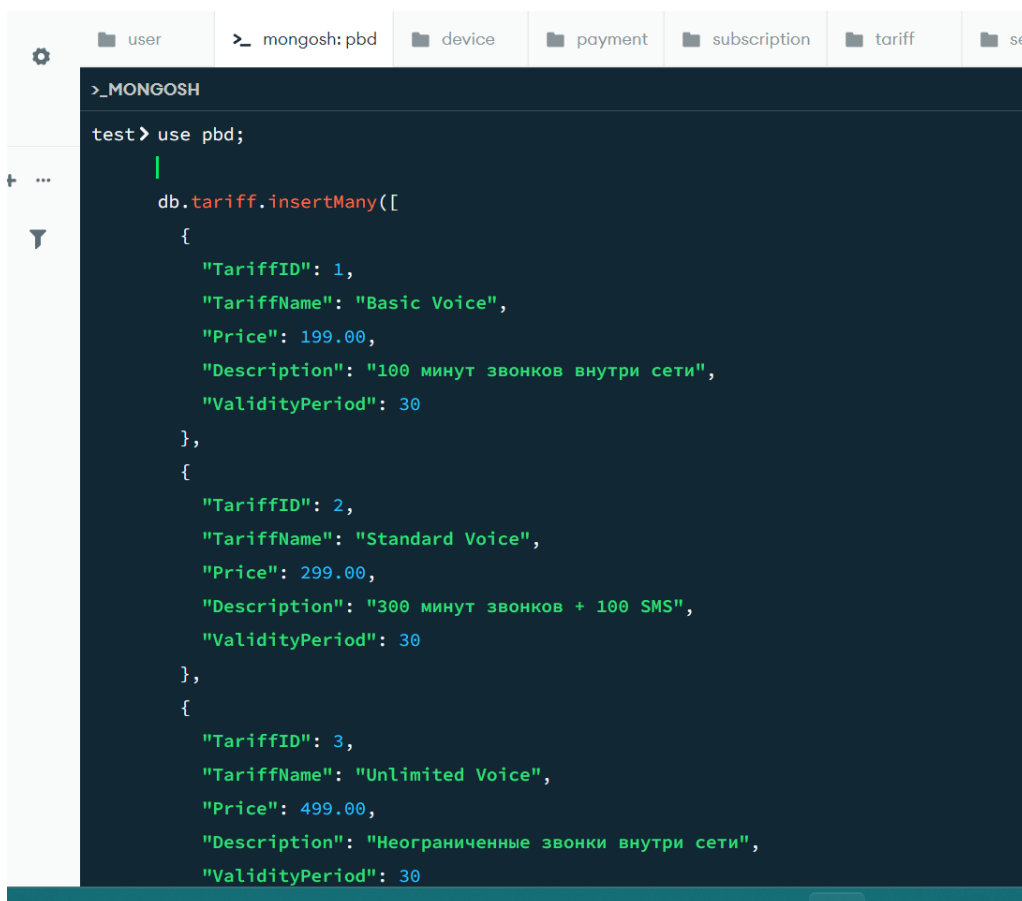
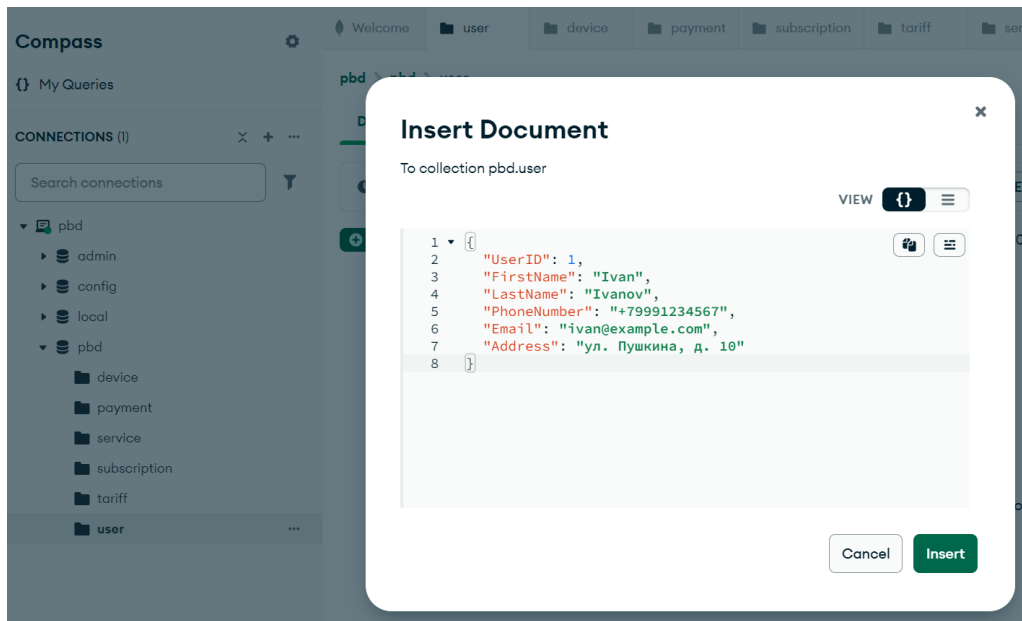
Решение:

Наша изначальная схема:



На этапе планирования решим что ServiceSubscription будет вложена в Subscription для удобства.

В MongoDB вместо таблиц используются коллекции, а вместо строк – документы. Создадим нужные коллекции и начнем их заполнять.



По такому примеру заполняем все коллекции, любым удобным способом.

После того как заполнили все коллекции, обновим subscription чтобы учесть в нем связь многие ко многим ServiceSubscription из схемы:

```
> db.subscription.updateOne(
  { SubscriptionID: 1 },
  { $set: { Services: [1, 3, 5] } }
);

db.subscription.updateOne(
  { SubscriptionID: 2 },
  { $set: { Services: [2, 4] } }
);

db.subscription.updateOne(
  { SubscriptionID: 3 },
  { $set: { Services: [1, 2] } }
);

db.subscription.updateOne(
  { SubscriptionID: 4 },
  { $set: { Services: [5, 6] } }
);

db.subscription.updateOne(
  { SubscriptionID: 5 },
  { $set: { Services: [2, 3, 4] } }
);
```

Как видим данные обновились:

The screenshot shows the MongoDB Compass interface. At the top, the connection is 'mongodb://localhost:27020/pbd'. The 'subscription' collection is selected. The 'Documents' tab is active, showing 10 documents. The first document is expanded, displaying the following fields:

- _id: ObjectId('68238e15a9104ea6d68eca38')
- SubscriptionID: 1
- UserID: 1
- TariffID: 3
- StartDate: 2024-12-01T00:00:00.000+00:00
- EndDate: 2025-01-01T00:00:00.000+00:00
- Status: "active"
- Services: Array (3)
 - 0: 1
 - 1: 3
 - 2: 5

Создадим индекс в коллекции subscription по двум полям UserID и TariffID.

```
> db.subscription.createIndex(
  { UserID: 1, TariffID: 1 },
  { name: "user_tariff_index" }
);
< user_tariff_index
pbd>
```

Проверим что индекс появился:

pbd > pbd > subscription Open Mongol

Documents 10 Aggregations Schema **Indexes 2** Validation

Create Index Refresh VIEWING INDEXES SEARCH I

Name & Definition	Type	Size	Usage	Properties	Status
> _id_	REGULAR	20.5 KB	4 (since Tue May 13 2025)	UNIQUE	READY
> user_tariff_index	REGULAR	20.5 KB	0 (since Tue May 13 2025)	COMPOUND	READY

Проверим индекс:

```
> db.subscription.find({ UserID: 3, TariffID: 1 });
< {
  _id: ObjectId('68238e15a9104ea6d68eca3a'),
  SubscriptionID: 3,
  UserID: 3,
  TariffID: 1,
  StartDate: 2024-12-10T00:00:00.000Z,
  EndDate: 2025-01-10T00:00:00.000Z,
  Status: 'active',
  Services: [
    1,
    2
  ]
}
```

Теперь создадим представление, которое покажет:

- имя пользователя
- email
- ID подписки
- статус подписки
- название тарифа
- цена подписки

```
pbdb> db.createView(  
  "UserSubscriptionsView",  
  "subscription",  
  [  
    {  
      $lookup: {  
        from: "user",  
        localField: "UserID",  
        foreignField: "UserID",  
        as: "user_info"  
      }  
    },  
    {  
      $unwind: "$user_info"  
    },  
    {  
      $lookup: {  
        from: "tariff",  
        localField: "TariffID",  
        foreignField: "TariffID",  
        as: "tariff_info"  
      }  
    },  
    {  
      $unwind: "$tariff_info"  
    },  
    {  
      $project: {  
        _id: 0,  
        SubscriptionID: 1,  
        Status: 1,  
        UserName: {  
          $concat: ["$user_info.FirstName", " ", "$user_info.LastName"]  
        },  
        Email: "$user_info.Email",  
        TariffName: "$tariff_info.TariffName",  
        Price: "$tariff_info.Price"  
      }  
    }  
  ]  
);
```

Проверка:

```
> db.UserSubscriptionsView.find().pretty();
< {
  SubscriptionID: 1,
  Status: 'active',
  UserName: 'Иван Иванов',
  Email: 'ivan@example.com',
  TariffName: 'Unlimited Voice',
  Price: 499
}
{
  SubscriptionID: 2,
  Status: 'active',
  UserName: 'Мария Петрова',
  Email: 'maria@example.com',
  TariffName: 'Standard Internet',
  Price: 599
}
{
  SubscriptionID: 3,
  Status: 'active',
  UserName: 'Алексей Смирнов',
  Email: 'aleksei@example.com',
  TariffName: 'Basic Voice',
  Price: 199
}
{
  SubscriptionID: 4,
  Status: 'expired',
  UserName: 'Ольга Кузнецова',
  Email: 'olga@example.com',
  TariffName: 'Standard Voice',
  Price: 299
}
{
  SubscriptionID: 5,
  Status: 'active',
  UserName: 'Дмитрий Попов',
  Email: 'dmitry@example.com',
  TariffName: 'Unlimited Internet',
  Price: 999
}
```

Итоговая база данных и коллекции:

Compass

My Queries

CONNECTIONS (1)

Search connections

mongodb: pbd

admin

config

local

pbd

UserSubscriptionsView

device

payment

service

subscription

tariff

user

mongodb: pbd

pbd

pbd

Sort byCollection Namejr

device

Storage size:20.48 kB

Documents:10

Avg. document size:112.00 B

Indexes:1

Total index size:20.48 kB

payment

Storage size:20.48 kB

Documents:10

Avg. document size:108.00 B

Indexes:1

Total index size:20.48 kB

service

Storage size:20.48 kB

Documents:10

Avg. document size:135.00 B

Indexes:1

Total index size:20.48 kB

subscription

Storage size:20.48 kB

Documents:10

Avg. document size:153.00 B

Indexes:2

Total index size:40.96 kB

tariff

Storage size:20.48 kB

Documents:10

Avg. document size:161.00 B

Indexes:1

Total index size:36.84 kB

user

Storage size:20.48 kB

Documents:10

Avg. document size:192.00 B

Indexes:1

Total index size:36.84 kB

UserSubscriptionsView

VIEW

READ-ONLY

View on:

subscription

Примеры запросов:

```
> db.user.find({ LastName: "Иванов" })
< {
  _id: ObjectId('68238171acf854ed70893acd'),
  UserID: 1,
  FirstName: 'Иван',
  LastName: 'Иванов',
  PhoneNumber: '+79991234567',
  Email: 'ivan@example.com',
  Address: 'ул. Пушкина, д. 10'
}
> db.tariff.find({ Price: { $gt: 300 } })
< [
  {
    _id: ObjectId('68238aa8acf854ed70893b05'),
    TariffID: 3,
    TariffName: 'Unlimited Voice',
    Price: 499,
    Description: 'Неограниченные звонки внутри сети',
    ValidityPeriod: 30
  },
  {
    _id: ObjectId('68238ab4acf854ed70893b07'),
    TariffID: 4,
    TariffName: 'Basic Internet',
    Price: 399,
    Description: '5 ГБ трафика',
    ValidityPeriod: 30
  },
  {
    _id: ObjectId('68238ac2acf854ed70893b09'),
    TariffID: 5,
    TariffName: 'Standard Internet',
    Price: 599,
    Description: '15 ГБ трафика',
    ValidityPeriod: 30
  }
]
```

Проверка индекса:

```
> db.subscription.getIndexes()
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { UserID: 1, TariffID: 1 }, name: 'user_tariff_index' }
]
pbd> |
```