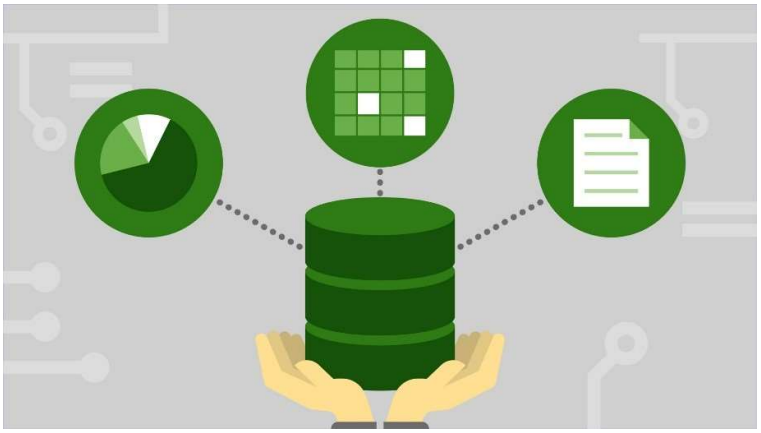
به نام خدا

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

پیش‌گزارش شماره ۵

پرنیان فاضل
۸۱۰۱۹۸۵۱۶

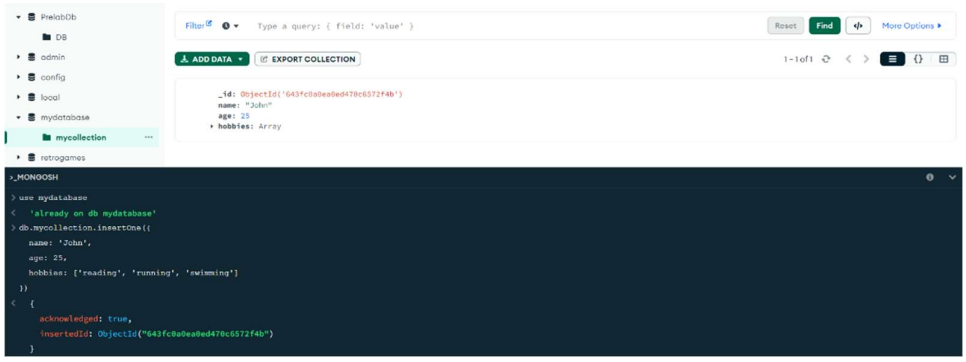
بهار ۱۴۰۲

آشنایی با مانگودی‌بی

دستورات به همراه نمونه استفاده از دستور:
(تغییرات در mydatabase انجام می‌شود)

- **insert**

در اینجا یک سندگرا inert میکنیم به این صورت که name و age و hobbies را مشخص می‌کنیم.



همانطور که دیده می‌شود پس از refresh کردن mydatabase هم طبق تغییر ما به روز شده است.

- **find**

حال با استفاده از دستور find، سندگرا قبل که اضافه کردیم را پیدا می‌کنیم. خروجی به صورت زیر است:

```
> db.mycollection.find({ age: { $gt: 20 } })
< {
    _id: ObjectId("643fc0a0ea0ed470c6572f4b"),
    name: 'John',
    age: 25,
    hobbies: [
      'reading',
      'running',
      'swimming'
    ]
  }
```

- **replace**

در این قسمت با استفاده از دستور replace، سندگرا قبل را به این صورت تغییر می‌دهیم که اسم John را به Jane تغییر دهیم. میبینیم که دیتابیس هم به درستی تغییر پیدا می‌کند:

```
> db.mycollection.replaceOne({ name: 'John' }, { name: 'Jane', age: 30 })
<   {
      acknowledged: true,
      insertedId: null,
      matchedCount: 1,
      modifiedCount: 1,
      upsertedCount: 0
    }
> db.mycollection.find()
<   {
      _id: ObjectId("643fc0a0ea0ed470c6572f4b"),
      name: 'Jane',
      age: 30
    }
```

- **delete**

حال رکورد مذکور را با استفاده از ()deleteOne پاک می‌کنیم. میبینیم که دیتابیس هم به درستی تغییر پیدا می‌کند:

```
> db.mycollection.find()
<   {
      _id: ObjectId("643fc0a0ea0ed470c6572f4b"),
      name: 'Jane',
      age: 30
    }
> db.mycollection.deleteOne({ name: 'Jane' })
<   {
      acknowledged: true,
      deletedCount: 1
    }
> db.mycollection.find()
<
mydatabase>
```

- **aggregation**

در اینجا من ۲ سند دیگر را وارد دیتابیس کرد. حال آن‌ها را با استفاده از aggregate و $sort، بر اساس فیلد سن و به صورت صعودی مرتب می‌کنم. میبینیم که خروجی درست است:

```
> db.mycollection.aggregate([
    { $sort: { totalAge: 1 } }
  ])
< {
    _id: ObjectId("643fc4c1ea0ed470c6572f4c"),
    name: 'Parnian',
    age: 25,
    hobbies: [
      'reading',
      'running',
      'swimming'
    ]
  }
  {
    _id: ObjectId("643fc4d2ea0ed470c6572f4d"),
    name: 'Andrew',
    age: 39,
    hobbies: [
      'reading',
      'running',
      'swimming'
    ]
  }
```

* قابل توجه است که در صورت استفاده از 1- به جای ۱، داده‌ها به صورت نزولی مرتب می‌شدند:

```
> db.mycollection.aggregate([
    { $sort: { totalAge: -1 } }
  ])
< {
    _id: ObjectId("643fc4c1ea0ed470c6572f4c"),
    name: 'Parnian',
    age: 25,
    hobbies: [
      'reading',
      'running',
      'swimming'
    ]
  }
  {
    _id: ObjectId("643fc4d2ea0ed470c6572f4d"),
    name: 'Andrew',
    age: 39,
    hobbies: [
      'reading',
      'running',
      'swimming'
    ]
  }
```

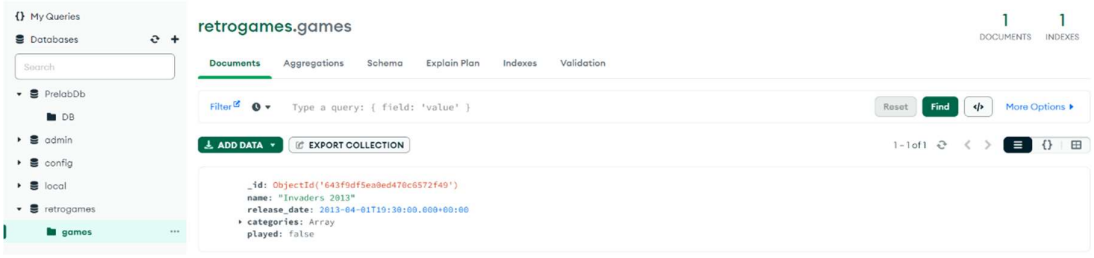
نصب و راه اندازی مانگودی‌بی

مانگودی‌بی را مطابق راهنمای داده شده نصب کردم.

به بخش retrogames رفته و دستورات را انجام می‌دهیم. ابتدا به دیتابیس retrogames متصل می‌شویم و یک رکورد به آن اضافه می‌کنیم:

```
> use retrogames
<   'switched to db retrogames'
> game1 =
      { name: "Invaders 2013",
      release_date: new Date(2013, 03, 02),
      categories: ["space", "shooter", "remake"],
      played: false
      }
<   {
      name: 'Invaders 2013',
      release_date: 2013-04-01T19:30:00.000Z,
      categories: [ 'space', 'shooter', 'remake' ],
      played: false
    }
> db.games.insertOne(game1)
<   {
      acknowledged: true,
      insertedId: ObjectId("643f9df5ea0ed470c6572f49")
    }
retrogames >
```

همانطورکه دیده می‌شود دیتابیس اضافه شده است:



حال با توجه به اینکه دستور use retrogames که در بالا استفاده کرده ایم، دیتابیس پیش فرض را به retrogames تغییر داده است، بنابراین db به این دیتابیس اشاره دارد. حال سندی که در بخش قبل اضافه کردیم را با استفاده از دستور find میبینیم:

```
> db.games.find()
<   {
      _id: ObjectId("643f9df5ea0ed470c6572f49"),
      name: 'Invaders 2013',
      release_date: 2013-04-01T19:30:00.000Z,
      categories: [
        'space',
        'shooter',
        'remake'
      ],
      played: false
    }
retrogames >
```

دقت شود که در اینجا چون پایگاه داده تنها یک سندگرا دارد، خروجی تنها یک سندگرا را باز می‌گرداند. اما ممکن است مجموعه‌ای شامل هزاران بازی باشد که برای بازیابی آن‌ها، نیاز به ابزاری برای محدود کردن سندگراهای واکشی شده خواهد بود. در پوسته MongoDBمیتوان از تابع limit محدودیت اعمال کرد:

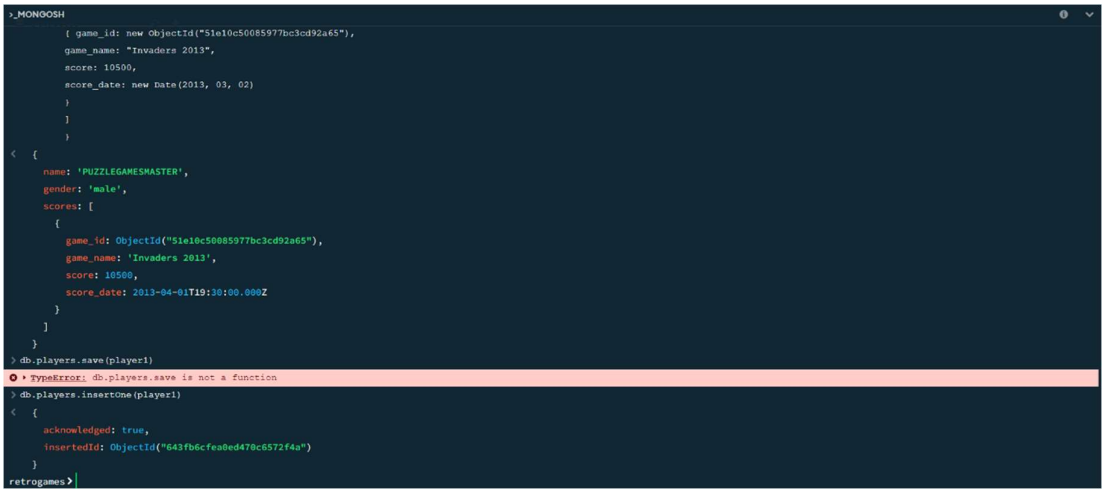
```
> db.games.find().limit(100)
<   {
      _id: ObjectId("643f9df5ea0ed470c6572f49"),
      name: 'Invaders 2013',
      release_date: 2013-04-01T19:30:00.000Z,
      categories: [
        'space',
        'shooter',
        'remake'
      ],
      played: false
    }
retrogames >
```

*توجه شود که در اینجا چون تنها یک بازی در دیتابیس موجود است خروجی این بخش و بخش قبل یکسان شد.

حال برای جست‌وجو و اکشی براساس یک فیلد اطلاعاتی خاص، میتوان از دستور زیر استفاده کرد: (در اینجا بر اساس نام محدود شده است)

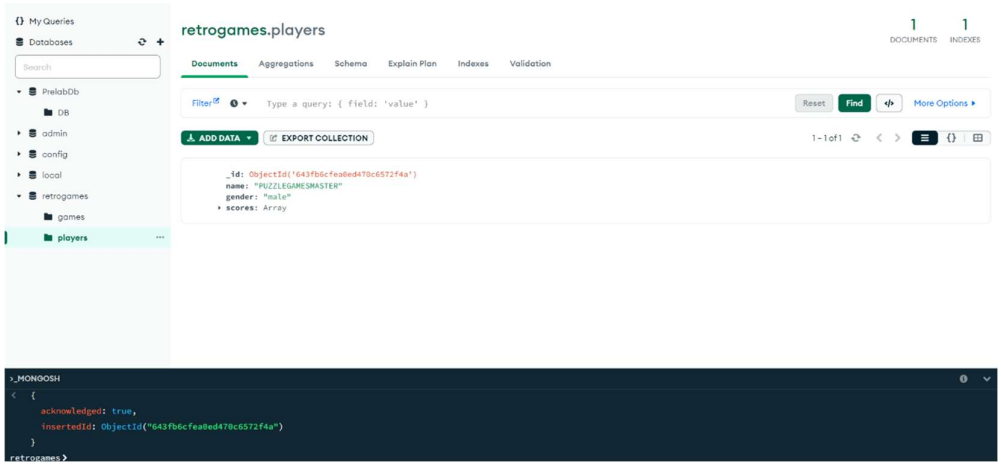
```
> db.games.findOne({ name: "Invaders 2013"})
<   {
      _id: ObjectId("643f9df5ea0ed470c6572f49"),
      name: 'Invaders 2013',
      release_date: 2013-04-01T19:30:00.000Z,
      categories: [
        'space',
        'shooter',
        'remake'
      ],
      played: false
    }
```

حال یک بازیکن به دیتابیس اضافه می‌کنیم:



در دستورات داده شده از save استفاده شده که نادرست است و از insertOne استفاده میکنیم.

حال پس از refresh کردن، میتوانیم player را در دیتابیس ببنیم:



اکنون با توجه به اینکه بازیکنی وجود دارد که امتیازی را برای بازی Invaders 2013 به ثبت رسانده، دستور زیر مقدار فیلد اطلاعاتی playedرا به مقدار true به‌روزرسانی می‌کند. در MongoDB برای به‌روزرسانی هر سندگرا از دستور db.games.update استفاده می‌شود که در آن، پارامتر نخست شرط جست‌وجو را مشخص کرده و پارامتر دوم که با اپراتور $set مشخص می‌شود، مقدار مورد نظر برای به‌روزرسانی را مشخص می‌کند.

```
>_MONGOSH
> db.games.update(
    { _id: new ObjectId("51e10c50085977bc3cd92a65") },
    { $set: { played: true } }
    )
< 'DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.'
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0
  }
```

در دستور بالا، مدیر پایگاه داده،بازی با شناسه منحصربه‌فرد را که همان شناسه سندگرا game1 است را پیدا کرده و مقدار فیلد اطلاعاتی played آن را با مقدار true به‌روزرسانی می‌کند.

حال اگر همین player1 دوباره به انجام همین بازی پرداخت، اطلاعات جدید باید به مجموعه players افزوده شود. به این منظور، فیلد score در سندگرا player1، آرایه‌ای از امتیازات خواهد بود و درنتیجه در این طراحی، افزودن امتیاز جدید به معنای به‌روزرسانی، فیلد اطلاعات score برای player1 است.

```
> db.players.update(
    { _id: new ObjectId("51e1108908597 7bc3cd92a67") },
    { $push: { scores: {
        game_id: new ObjectId("51e10c50085977bc3cd92a65"),
        game_name: "Invaders 2013",
        score: 30250,
        score_date: new Date(2013, 2, 3)
      }
    }}
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0
  }
```

این یک دستور به‌روزرسانی است که پارامتر اول، شناسه سندگرا مورد نظر برای به‌روزرسانی را مشخص می‌کند که شناسه تولید شده توسط دیتابیس برای player1 بوده و گزینه دوم، اپراتور $push است که برای افزودن یک المان جدید به آرایه score مورد استفاده قرار گرفته است. به‌زبان ساده‌تر این دستور بیان می‌کند که باید المان جدیدی به فیلد اطلاعاتی score که خود ساختار آرایه‌ای دارد متعلق به شناسه player1 در مجموعه players افزوده شود.

من چند بار متوالی از دستور update مانند دستور بالا استفاده کردم و پس از ()find خروجی به صورت زیر شد:

```
> db.players.find()
<   {
      _id: ObjectId("643fb6cfea0ed470c6572f4a"),
      name: 'PUZZLEGAMESMASTER',
      gender: 'male',
      scores: [
        {
          game_id: ObjectId("51e10c50085977bc3cd92a65"),
          game_name: 'Invaders 2013',
          score: 10500,
          score_date: 2013-04-01T19:30:00.000Z
        },
        {
          game_id: ObjectId("51e10c50085977bc3cd92a65"),
          game_name: 'Invaders 2013',
          score: 30250,
          score_date: 2013-04-02T19:30:00.000Z
        },
        {
          game_id: ObjectId("643f9df5ea0ed470c6572f49"),
          game_name: 'Invaders 2013',
          score: 30250,
          score_date: 2013-04-02T19:30:00.000Z
        }
      ]
    }
retrogames>
```

همانطور که مشاهده می‌شود به scores به درستی اضافه شده اند.

مشکلات و توضیحات تکمیلی

در بعضی قسمت‌های کد داده شده در سایت ارور وجود داشت که با کمی جستجو برطرف شد. برای مثال تغییر »« به "" و یا تغییر save به insertOne().

آنچه آموختم / پیشنهادات

با مونگودی‌بی آشنا شدم و دستورات اولیه آن را یادگرفتم.