ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО КАТЕДРА “КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ”

Дата на задаване: 24.02.2022 Оценка:..............................................

Краен срок за предаване и защита: Преподавател......................(подпис)

Преди изпита Дата:..................................................

З А Д А Н И Е

*за курсов проект по дисциплината*: Нерелационни бази данни

1. Тема: Създайте MongoDB база данни, която описва посетителите на обществени сгради (име, фамилия, номер на мобилния телефон, локация – x, y, z). Напишете програмния код, който автоматично населява базата данни с информация за 1000 посетители.

1. Изисквания по съдържанието: ..............................................................................................
   1. Теоретична част:
      * Проектиране на базата данни. Избор и обосновка на подходяща схема.
      * Разработка на необходимото програмно осигуряване за населяване на базата данни с информация за 1000 посетители.
      * Експерименти. Изводи.
      * Използвана литература.

.....................................................................................................

* 1. Практическо изпълнение:
  2. Обем на чертежите: .............................................................................................................

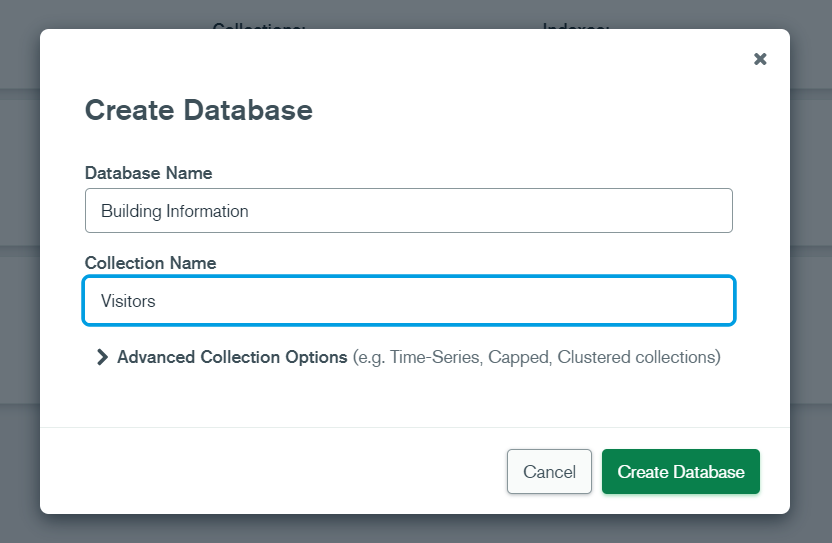
1. Задължителни консултации: ........................... .......................... ..............................

*Студент*: Ренета Георгиева Христова *Преподавател*:

(име, презиме, фамилия)  (фамилия и подпис) *курс*: 3, *група*: 1, *фак. №*: доц. Р. Иванов / ......................

***Проектиране на базата данни. Избор и обосновка на подходяща схема.***

За изпълнението на задачата използвам MongoDB Compass и Node .js. Първо създадох база данни на име BuildingInformation, в която добавих и колекция Visitors.



Фигура 1 Създаване на база данни

В последствие програмно се добавя още една колекция Buildings, в която са описани типа на обществената сграда и id. Връзката между обектите избрах да е 1:1, като за всеки потребител има записана една сграда. В случай, че един и същи потребител посети една и съща сграда няколко пъти, ще има толкова резултати, колкото пъти я е посетил. Това дава по-голяма гъвкавост при агрегацията на изходната информация, а при желание да се получи по- задълбочена информация относно посещенията или посетителите може да се използва сортитане.

***Разработка на необходимото програмно осигуряване за населяване на базата данни с информация за 1000 посетители.***

За да свържа Mongo DB и Node .js, използвам следния програмен код:

const MongoClient = **require**("mongodb").MongoClient;

const uri = "mongodb://localhost:27017";

Следващата стъпка е създаването на втора колекция Buildings, в която са описани четири типа обществени сгради.

let dbo = db.**db**("BuildingInformation");

  let buildingsObj = [

    { buldingType: "Museum" },

    { buldingType: "Library" },

    { buldingType: "Mall" },

    { buldingType: "Bussines center" },

  ];

След като са създадени записите за сградите, се добавят в базата и ги селектираме, за да може в последствие да боравим с данните за тях.

dbo

    .**collection**("Buildings")

    .**insertMany**(buildingsObj)

    .**then**((r) => {

      dbo

        .**collection**("Buildings")

        .**find**({})

        .**toArray**((err, result)

Декларирам празен масив obj, в които по-късно, чрез метод push, ще се добавят обектите за посетители.

let obj = [];

За да се създадат 1000 записа използвам for цикъл. При всяка итерация, на случаен принцип, се избира по едно id на сграда, което Mongo DB автоматично е добавил. Това id ще служи като връзка между Visitors и Buildings, като чрез него ще можем да проверим всеки посетител коя сграда е посетил.

for (let i = 0; i <= 1000; i++) {

            let randomBuilding =

              result[Math.**floor**(Math.**random**() \* result.length)];

*// Генериране на произволен телефонен номер*

            let randomPhone = Math.**floor**(Math.**random**() \* 1000000000);

Към вече декларирания масив obj се добавят обектите, които описваат посетителите.

obj.**push**({

              first\_name: ` First Name ${i}`,

              last\_name: `Last Name ${i}`,

              phone: `${randomPhone}`,

              location: ["11", "22", "33"],

              visitedBuilding: randomBuilding.\_id,

            });

Накрая, към колекцията Visitors, с метод insertMany добавяме записите.

dbo

            .**collection**("Visitors")

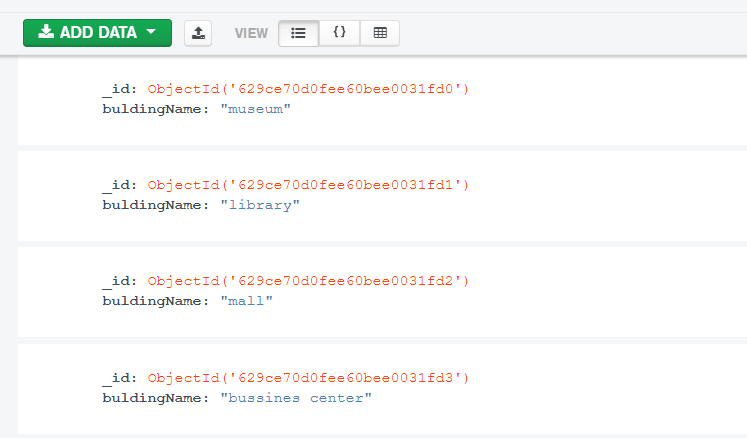
            .**insertMany**(obj)

            .**then**((r) => db.**close**());

        });

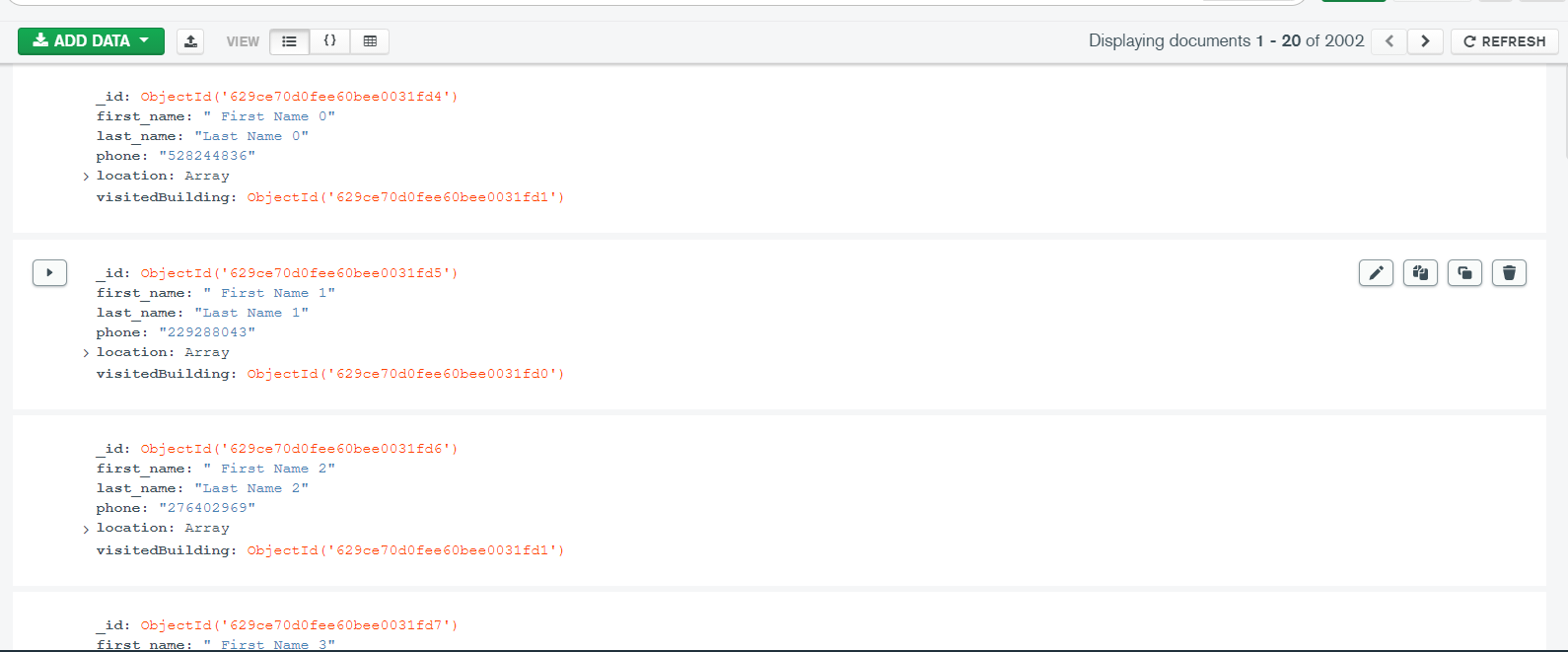
***Експерименти. Изводи.***

След изпълнението на кода, в колекцията Buildings се добавят четирите записа, които описват типа- buldingName и id на сградата.



Фигура 2 Колекция Buildings

В колекцията Visitors се побавят 1000 записа на посетители, които са описани с id, име и фамилия, телефонен номер, локация (която в случая е една и съща за всички, понеже не съм сигурна дали тя също трябва да се генерира на случаен принцип) и visitedBuilding, което съответсва на id на сграда от колекцията Buildings.



Фигура 3 колекция Visitors

***Използвана литература***

<https://www.w3schools.com/nodejs/nodejs_mongodb.asp>

<http://kst.tugab.bg/nosql/>

<https://www.mongodb.com/docs/guides/crud/insert/>

***Практическо изпълнение***

const MongoClient = **require**("mongodb").MongoClient;

const uri = "mongodb://localhost:27017";

MongoClient.**connect**(uri, function (err, db) {

  if (err) throw err;

  let dbo = db.**db**("BuildingInformation");

  let buildingsObj = [

    { buldingType: "Museum" },

    { buldingType: "Library" },

    { buldingType: "Mall" },

    { buldingType: "Bussines center" },

  ];

*// Insert 5 Buidings documents.*

  dbo

    .**collection**("Buildings")

    .**insertMany**(buildingsObj)

    .**then**((r) => {

*// След успешно вмъкване на заиса, селектиаме колекцията*

      dbo

        .**collection**("Buildings")

        .**find**({})

        .**toArray**((err, result) => {

          let obj = [];

          for (let i = 0; i <= 1000; i++) {

*// Получаваме Building Objeсt Id на случаен принцип и го добавяме*

            let randomBuilding =

              result[Math.**floor**(Math.**random**() \* result.length)];

*// Генериране на произволен телефонен номер*

            let randomPhone = Math.**floor**(Math.**random**() \* 1000000000);

*// Създаваме запис за посетител*

            obj.**push**({

              first\_name: ` First Name ${i}`,

              last\_name: `Last Name ${i}`,

              phone: `${randomPhone}`,

              location: ["11", "22", "33"],

              visitedBuilding: randomBuilding.\_id,

            });

          }

*// 1000 посетителя се добавят в колекция visitors*

          dbo

            .**collection**("Visitors")

            .**insertMany**(obj)

            .**then**((r) => db.**close**());

        });

    });

});