Лабораторная работа 15

ПСКП

ПОИТ-3

**Задание 01**

1. Зарегистрируйтесь в облаке [https://www.mongodb.com](https://www.mongodb.com/)
2. Создайте базу данных(БД) с именем **BSTU**
3. Создайте в БД коллекции: **faculty**, **pulpit**.
4. **faculty**

****

1. **pulpit**

****

**Задание 02**

1. Разработайте приложение **15-01**, представляющее собой HTTP-сервер, прослушивающий порт ***3000***. Сервер должен обрабатывать запросы, описанные в следующих таблицах.

|  |  |  |
| --- | --- | --- |
| Метод  запроса | URI | Назначение |
| GET | api/faculties | получить список всех факультетов в json-формате |
| GET | api /pulpits | получить список всех кафедр в json-формате |
| POST | api/faculties | добавить новый факультет, данные в json-формате |
| POST | api/pulpits | добавить новую кафедру, данные в json-формате |
| PUT | api/faculties | корректировать информацию о факультете, данные в json-формате |
| PUT | api/pulpits | корректировать информацию о кафедре, данные в json-формате |
| DELETE | api/faculties/xyz | xyz - код факультета, удалить факультет с кодом xyz |
| DELETE | api/pulpits/xyz | xyz - код кафедры, удалить кафедру с кодом xyz |

|  |  |
| --- | --- |
| Метод  запроса | Назначение |
| GET | найденные в БД данные или сообщение об ошибке в json-формате |
| POST | принимает данные в json-формате,  возвращает добавленные в БД данные или сообщение об ошибке в json-формате |
| PUT | принимает данные в json-формате,  возвращает измененные в БД данные или сообщение об ошибке в json-формате |
| DELETE | возвращает удаленные в БД данные или сообщение об ошибке в json-формате |

1. Для проверки работоспособности приложения, используйте **POSTMAN**.

**Задание 04** Ответьте на следующие вопросы

1. Опишите структуру БД, поддерживаемую СУБД MangoDB.
2. Перечислите все функции API СУБД MangoDB с помощью которых можно извлечь данные из БД.
3. Перечислите все функции API СУБД MangoDB с помощью которых можно добавить данные в БД.
4. Перечислите все функции API СУБД MangoDB с помощью которых можно удалить данные в БД.
5. Перечислите все функции API СУБД MangoDB с помощью которых можно изменить данные в БД.
6. Опишите структуру БД, поддерживаемую СУБД MangoDB.

\*состоит из коллекций (реляц бд сост.из таблиц)

Кажд.коллекция имеет уник.проивольный id

\*является документно-ориент.сист, в кот.центральное понятие – документ

Документ – объект, хранящий инфу (сост из пар ключ-значений)

(реляц бд исп.табл.ус-во четким кол-вом строк и столбцов)

\*примен.бесхемная организ.неструктур.или слабоструктур д-х

\*хранит д-е любого типа

=> оперативная обработка д-х, запись и вывод рез-тов

=> отлично подходит для работы с нереляц.д-ми

1. Перечислите все функции API СУБД MangoDB с помощью которых можно извлечь данные из БД.

\*find() – как селект – выводит все

\*find ({..}) – селект по условию

\*find().limit(n) – установить макс.допустимое кол-во получаемых доков

\*find().skip(n) – вывести результаты пропустив первые n доков

\*find().sort({name: 1/-1}) – по возр/убыв

\*find().sort({name:1}).skip(3).limit(3);

1. Перечислите все функции API СУБД MangoDB с помощью которых можно добавить данные в БД.

\*insertOne({..}) – добавл.один документ в коллекцию

\*insertMany([{..}, {..}]) – добавл.неск.доков в колл.

\*insert() – м.добавл.как 1, так и неск.доков в колл.

1. Перечислите все функции API СУБД MangoDB с помощью которых можно удалить данные в БД.

\*remove({name: “Tom”}) – удалить из коллекции документ

\*findOneAndDelete(); - найти и удалить

\*deleteOne(), deleteMany();

\*drop() – удалить коллекцию

\*dropDatabase() – удалить БД

1. Перечислите все функции API СУБД MangoDB с помощью которых можно изменить данные в БД.

\*save(\_id) – если нах.док.с таким id – обновл; если нет – вставляется;

\*update(query, objNew, options)

\*query – что обновить

\*objNew – новая инфа, которая заменит старую при обновлении

\*options – доп.пар-ры при обновл. (upsert, multi)

\*upsert ( t – обнов., если найден + созд если нет; f – обнов, не созд)

\*multi (указ.должен ли обновл 1 эл-т(по умол); должны обн.все)

db.users.update({name : "Tom"}, {name: "Tom", age : 25}, {upsert: true})

\*updateOne() – обновить 1

db.users.updateOne({name : "Tom", age: 29}, {$set: {salary : 360}})

\*updateMany() – обновить все, кот.соотв.критерию

db.users.updateMany({name : "Tom"}, {$set: {salary : 560}})

-findOneAndUpdate();