

**Міністерство освіти і науки України**  
**Національний університет «Львівська політехніка»**  
**Інститут комп'ютерних наук та інформаційних технологій**  
**Кафедра «Системи штучного інтелекту»**



**Лабораторна робота №14**  
**з дисципліни «Організація баз даних та знань»**

***Виконав:***

студент

групи КН-211

Пучак Віталій

***Перевірила:***

Якимишин Х.М.

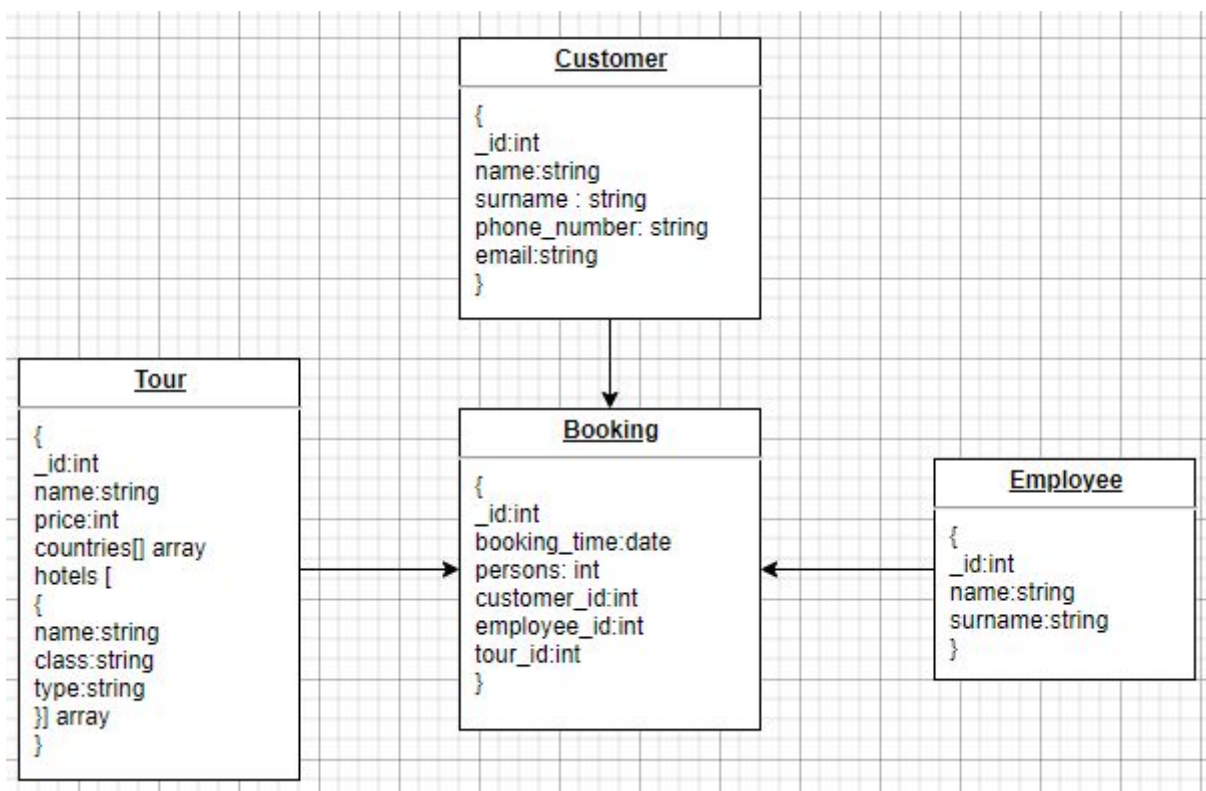
*Львів 2020 р.*

## “Розробка бази даних типу NoSQL”

**Мета роботи:** здобуття практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB.

### Хід роботи.

1. Розробити схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB..
2. Перетворити сутності діаграми БД, розробленої для лабораторної роботи №1, у структури, прийнятні для обробки в MongoDB.
3. Забезпечити реалізацію функцій редагування, додавання та видалення інформації в «сутність».



Створимо потрібні нам колекції

```
> db.createCollection("customer");
{ "ok" : 1 }
> db.aaa.drop()
true
> db.createCollection("employee");
{ "ok" : 1 }
> db.createCollection("booking");
{ "ok" : 1 }
> db.createCollection("tour");
{ "ok" : 1 }
```

Заповнимо деякі з них

```
db.customers.insertOne({ "name": "danylo", "surname": "mashtalir", "phone_number": "0673213244", "email": "danylo123@mail" })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5ec57dba97037e7d023adde4")
}
> db.customers.insertOne({ _id: 2, "name": "oleg", "surname": "vvd", "phone_number": "0673213244", "email": "oleg123@mail" })
{ "acknowledged" : true, "insertedId" : 2 }
> db.tour.insertOne({ name: "Bukovel", price: 1700, countries: ["Ukraine", "Poland", "Germany"], hotels: [{ name: "Hetman", class: "5 stars", type: "resort" }] })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5ec5812997037e7d023adde5")
}
```

Виведемо на екран добавлені дані

```
> db.customers.find().pretty()
{
  "_id" : ObjectId("5ec57dba97037e7d023adde4"),
  "name" : "danylo",
  "surname" : "mashtalir",
  "phone_number" : "0673213244",
  "email" : "danylo123@mail"
}
{
  "_id" : 2,
  "name" : "oleg",
  "surname" : "vvd",
  "phone_number" : "0673213244",
  "email" : "oleg123@mail"
}
```

```

> db.tour.find().pretty()
{
  "_id" : ObjectId("5ec5812997037e7d023adde5"),
  "name" : "Bukovel",
  "price" : 1700,
  "countries" : [
    "Ukraine",
    "Poland",
    "Germany"
  ],
  "hotels" : [
    {
      "name" : "Hetman",
      "class" : "5 stars",
      "type" : "resort"
    }
  ]
}
>

```

## Оновимо дані

```

> db.customers.update({_id:2},{ $set : {name:"roman",surname:"shevchenko"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find({_id:2}).pretty()
2020-05-20T22:30:38.697+0300 E QUERY [js] uncaught exception: SyntaxError: missing ) after argument list :
@ (shell):1:21
> db.customers.find({_id:2}).pretty()
{
  "_id" : 2,
  "name" : "roman",
  "surname" : "shevchenko",
  "phone_number" : "0673213244",
  "email" : "oleg123@mail"
}
>

```

Видалимо дані з id = 2

```
> db.customers.remove({_id:2})
WriteResult({ "nRemoved" : 1 })
> db.customers.find().pretty()
{
  "_id" : ObjectId("5ec57dba97037e7d023adde4"),
  "name" : "danylo",
  "surname" : "mashtalir",
  "phone_number" : "0673213244",
  "email" : "danylo123@mail"
}
```

Виконаємо ще декілька функцій:

```
> db.customers.count()
2

> db.customers.distinct("name")
[ "danylo", "roman" ]
```