Class 5

Projection Operators:

Understand Projection:

1. Create and demonstrate how projection operators ($, $elematch and $slice) would be used in the MongoDB.

Name	Description
$	Projects the first element in an array that matches the query condition.
$elemMatch	Projects the first element in an array that matches the specified $elemMatch condition.
$meta	Projects the available per-document metadata.
$slice	Limits the number of elements projected from an array. Supports skip and limit slices.

Create new collection called candidates

To see whether the collection is added or not use operation

>show collections

From this operation we can see whether the collection is added or not.

Now the new collection "candidates" is displayed.

db.candidates.find()

This above operation display all the candidates details.

```
{
  _id: ObjectId('668330b9acc6041a15186d29'),
  name: 'Alice Smith',
  age: 20,
  courses: [ 'English', 'Biology', 'Chemistry' ],
  gpa: 3.4,
  home_city: 'New York City',
  blood_group: 'A+',
  is_hotel_resident: true
},
{
  _id: ObjectId('668330b9acc6041a15186d2a'),
  name: 'Bob Johnson',
  age: 22,
  courses: [ 'Computer Science', 'Mathematics', 'Physics' ],
  gpa: 3.8,
  home_city: 'Los Angeles',
  blood_group: 'O-',
  is_hotel_resident: false
},
{
  _id: ObjectId('668330b9acc6041a15186d2b'),
  name: 'Charlie Lee',
  age: 19,
```

db.candidates.find({},{name:1,age:1,gpa:1});

```
db> db.candidates.find({},{name:1,age:1,gpa:1});
[
  {
    _id: ObjectId('668330b9acc6041a15186d29'),
    name: 'Alice Smith',
    age: 20,
    gpa: 3.4
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    age: 22,
    gpa: 3.8
  },
```

db.candidates.find({},{_id:0,courses:0});

```
db> db.candidates.find({},{_id:0,courses:0});
[
  {
    name: 'Alice Smith',
    age: 20,
    gpa: 3.4,
    home_city: 'New York City',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    name: 'Bob Johnson',
    age: 22,
    gpa: 3.8,
    home_city: 'Los Angeles',
    blood_group: 'O-',
    is_hotel_resident: false
  },
```

Finding the candidates Enrolled in "computer science" with specific projection.

db.candidates.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1});

```
db> db.candidates.find({courses:{$elemMatch:{$eq:"Computer Science"}}},{name:1,"courses.$":1})
;
[
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2f'),
    name: 'Gabriel Miller',
    courses: [ 'Computer Science' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d33'),

    courses: [ 'Computer Science' ]
  }
]
```

Slice Operators

db.candidates.find({},{name:1,courses:{$slice:2}})

```
db> db.candidates.find({},{name:1,courses:{$slice:2}});
[
  {
    _id: ObjectId('668330b9acc6041a15186d29'),
    name: 'Alice Smith',
    courses: [ 'English', 'Biology' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2a'),
    name: 'Bob Johnson',
    courses: [ 'Computer Science', 'Mathematics' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2b'),
    name: 'Charlie Lee',
    courses: [ 'History', 'English' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2c'),
    name: 'Emily Jones',
    courses: [ 'Mathematics', 'Physics' ]
  },
  {
    _id: ObjectId('668330b9acc6041a15186d2d'),
    name: 'David Williams',
    courses: [ 'English', 'Literature' ]
```