MCSD1123 - MONGO_DB LAB ASSIGNMENT

Name: Siti Nurainna binti Wahid,

Matric ID: MCS221018

Question 1

1.Write a query statement to display only address, cuisine and name

In [ ]:

```
[
    {
        '$project': {
            'address': 1,
            'cuisine': 1,
            'name': 1
        }
    }
]
```

2.Write a query to display the first 5 restaurant which is in the borough Manhattan.

In [ ]:

```
[
    {
        '$match': {
            'borough': 'Manhattan'
        }
    }, {
        '$limit': 5
    }
]
```

3.Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Manhattan and serve cuisine American.

In [ ]:

```
[
    {
        '$match': {
        'borough': 'Manhattan',
        'cuisine': 'American '
        }
    }, {
        '$skip': 5
    }, {
        '$limit': 5
    }
]
```

4.Write a query to find the restaurants that achieved a score, more than 80 but less than 100.

In [ ]:

```
[
    {
        "grades": {
```

```
        "$elemMatch": {
          "score": {
            "$gt": 80,
            "$lt": 100
      }}}
    }
]
```

5.Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 3rd element of grades array contains a grade of "A" and score 12 on an ISODate "2013-04-30T00:00:00Z"

In [ ]:

```
[
    {
        '$match': {
            'grades.2.grade': 'A',
            'grades.2.score': 12,
            'grades.2.date': datetime(2013, 4, 30, 0, 0, 0, tzinfo=timezone.utc)
        }
    }, {
        '$project': {
            '_id': 1,
            'name': 1,
            'grades': {
                '$filter': {
                    'input': '$grades',
                    'cond': {
                        '$and': [
                            {
                                '$eq': [
                                    '$$this.grade', 'A'
                                ]
                            }, {
                                '$eq': [
                                    '$$this.score', 12
                                ]
                            }, {
                                '$eq': [
                                    '$$this.date', datetime(2013, 4, 30, 0, 0, 0, tzinf
o=timezone.utc)
                                ]
                            }
                        ]
                    }
                }
            }
        }
    }
]
```

6.Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

In [ ]:

```
[
    {
        '$match': {
            'cuisine': {
                '$ne': 'American '
            },
            'grades.score': {
                '$gt': 70
            },
            'address.coord.0': {
                '$lt': -65.754168
            }
        }
```

```
    }
]
```

7.Write a query to find the restaurants which do not prepare any Italian cuisine and achieved a grade point 'A' not belongs to the borough Manhattan. The document must be displayed according to the cuisine in descending order.

In [ ]:

```
[
    {
        '$match': {
            'cuisine': {
                '$ne': 'Italian'
            },
            'grades.grade': 'A',
            'borough': {
                '$ne': 'Manhattan'
            }
        }
    }, {
        '$sort': {
            'cuisine': -1
        }
    }
]
```

8.Write a query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'Jewish/Kosher' and 'Caribbean' or restaurant's name begins with letter 'Wil'

In [ ]:

```
[
    {
        '$match': {
            '$or': [
                {
                    'cuisine': {
                        '$nin': [
                            'Jewish/Kosher', 'Caribbean'
                        ]
                    }
                }, {
                    'name': {
                        '$regex': re.compile(r"^Wil(?i)")
                    }
                }
            ]
        }
    }, {
        '$project': {
            'restaurant_id': 1,
            'name': 1,
            'borough': 1,
            'cuisine': 1
        }
    }
]
```

9.Write an aggregate query to find average score obtained by each of restaurant. Sort the score in ascending order and only view the first 5 restaurant. (Hint : use $unwind to reconstruct the grades array)

In [ ]:

```
[
    {
        '$unwind': {
            'path': '$grades',
```

```
            'includeArrayIndex': 'string',
            'preserveNullAndEmptyArrays': False
        }
    }, {
        '$group': {
            '_id': '$_id',
            'name': {
                '$first': '$name'
            },
            'borough': {
                '$first': '$borough'
            },
            'cuisine': {
                '$first': '$cuisine'
            },
            'avgScore': {
                '$avg': '$grades.score'
            }
        }
    }, {
        '$sort': {
            'avgScore': 1
        }
    }, {
        '$limit': 5
    }
]
```

10.Write an aggregate query to find average score obtained by each of restaurant. Sort the score in ascending order and only view the first 5 restaurant. (Hint : use $unwind to reconstruct the grades array)

In [ ]:

```
[
    {
        '$unwind': {
            'path': '$grades',
            'includeArrayIndex': 'string',
            'preserveNullAndEmptyArrays': False
        }
    }, {
        '$group': {
            '_id': '$_id',
            'name': {
                '$first': '$name'
            },
            'borough': {
                '$first': '$borough'
            },
            'cuisine': {
                '$first': '$cuisine'
            },
            'avgScore': {
                '$avg': '$grades.score'
            }
        }
    }, {
        '$sort': {
            'avgScore': 1
        }
    }, {
        '$limit': 5
    }
]
```

11.Write a query to count the number of restaurants at the Morris Park Avenue.

In [ ]:

```
[
```

```
    {
        '$match': {
            'address.street': 'Morris Park Ave'
        }
    }, {
        '$count': 'restaurantCount'
    }
]
```

Question 2

1.Write query to display invoice number, invoice date for StockCode 85123A that have quantity order more than 6.

In [ ]:

```
[
    {
        '$match': {
            'StockCode': '85123A',
            'Quantity': {
                '$gt': 6
            }
        }
    }, {
        '$project': {
            'InvoiceNo': 1,
            'InvoiceDate': 1
        }
    }
]
```

2.Write a query to display only StockCode and UnitPrice (in ascending order)

In [ ]:

```
[
    {
        '$project': {
            'StockCode': 1,
            'UnitPrice': 1
        }
    }, {
        '$sort': {
            'UnitPrice': 1
        }
    }
]
```

3.Find the total quantity order for StockCode 22941 for customer in United Kingdom

In [ ]:

```
[
    {
        '$match': {
            'StockCode': 22941,
            'Country': 'United Kingdom'
        }
    }, {
        '$group': {
            '_id': None,
            'totalQuantity': {
                '$sum': '$Quantity'
            }
        }
    }
```

```
]
```

4.Find the total quantity item purchase in invoice no 536367

In [ ]:

```
[
    {
        '$match': {
            'InvoiceNo': 536367
        }
    }, {
        '$group': {
            '_id': None,
            'totalQuantity': {
                '$sum': '$Quantity'
            }
        }
    }
]
```

5.Find the total quantity order for each StockCode.

In [ ]:

```
[
    {
        '$group': {
            '_id': '$StockCode',
            'totalQuantity': {
            '$sum': '$Quantity'
            }
        }
    }
]
```

6.Find the maximum quantity order of each StockCode.

In [ ]:

```
[
    {
        '$group': {
            '_id': '$StockCode',
            'maxQuantity': {
            '$max': '$Quantity'
            }
        }
    }
]
```

7.Find the StockCode and Description of maximum quantity in each order

In [ ]:

```
[
    {
        '$group': {
            '_id': '$InvoiceNo',
            'maxQuantity': {
                '$max': '$Quantity'
            },
            'StockCode': {
                '$first': '$StockCode'
            },
            'Description': {
                '$first': '$Description'
            }
```

```
            }
        }
]
```

Question 1(1)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 10 documents


```
_id: ObjectId('648d40addf6d5f55cd83a673')
address: Object
  building: "1007"
  coord: Array
    0: -73.856077
    1: 40.848447
  street: "Morris Park Ave"
  zipcode: "10462"
cuisine: "Bakery"
name: "Morris Park Bake Shop"
```


Question 1(2)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 5 documents


```
_id: ObjectId('648d40addf6d5f55cd83a675')
address: Object
  building: "351"
  coord: Array
    0: -73.98513559999999
    1: 40.7676919
  street: "West   57 Street"
  zipcode: "10019"
borough: "Manhattan"
cuisine: "Irish"
grades: Array
  0: Object
    date: 2014-09-06T00:00:00.000+00:00
    grade: "A"
    score: 2
  1: Object
    date: 2013-07-22T00:00:00.000+00:00
    grade: "A"
    score: 11
  2: Object
    date: 2012-07-31T00:00:00.000+00:00
    grade: "A"
    score: 12
  3: Object
    date: 2011-12-29T00:00:00.000+00:00
    grade: "A"
    score: 12
name: "Dj Reynolds Pub And Restaurant"
restaurant_id: "30191841"
```

Question 1(3)

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 5 documents

```
▸ address: Object
  borough: "Manhattan"
  cuisine: "American "
▸ grades: Array
  name: "Cafe Metro"
  restaurant_id: "40363298"
```

```
  _id: ObjectId('648d40addf6d5f55cd83a69f')
▸ address: Object
  borough: "Manhattan"
  cuisine: "American "
▸ grades: Array
  name: "Berkely"
  restaurant_id: "40363685"
```

Question 1(4)

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 3 documents

```
  _id: ObjectId('648d40addf6d5f55cd83a872')
▸ address: Object
  borough: "Manhattan"
  cuisine: "Indian"
▾ grades: Array
  ▸ 0: Object
  ▸ 1: Object
  ▸ 2: Object
  ▸ 3: Object
  ▸ 4: Object
  ▾ 5: Object
      date: 2012-04-06T00:00:00.000+00:00
      grade: "C"
      score: 92
  ▸ 6: Object
```

Question 1(5)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 6 documents

```
_id: ObjectId('648d40addf6d5f55cd83a674')
name: "Wendy'S"
▾ grades: Array
  ▾ 0: Object
      date: 2013-04-30T00:00:00.000+00:00
      grade: "A"
      score: 12
```

Question 1(6)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 5 documents

```
_id: ObjectId('648d40addf6d5f55cd83a872')
▸ address: Object
borough: "Manhattan"
cuisine: "Indian"
▾ grades: Array
  ▸ 0: Object
  ▸ 1: Object
  ▸ 2: Object
  ▸ 3: Object
  ▸ 4: Object
  ▾ 5: Object
      date: 2012-04-06T00:00:00.000+00:00
      grade: "C"
      score: 92
  ▸ 6: Object
```

Question 1(7)

STAGE OUTPUT OPTIONS

Sample of 10 documents


```
_id: ObjectId('648d40addf6d5f55cd83ae38')
▸ address: Object
borough: "Queens"
cuisine: "Vietnamese/Cambodian/Malaysia"
▸ grades: Array
name: "Pho Bac Vietnamese Seafood Cuisine"
restaurant_id: "40578058"
```

```
_id: ObjectId('648d40addf6d5f55cd83b26d')
▸ address: Object
borough: "Brooklyn"
cuisine: "Vegetarian"
▸ grades: Array
name: "Bliss Bakery & Cafe"
restaurant_id: "40763388"
```


Question 1(8)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 10 documents


```
_id: ObjectId('648d40addf6d5f55cd83a673')
borough: "Bronx"
cuisine: "Bakery"
name: "Morris Park Bake Shop"
restaurant_id: "30075445"
```

```
_id: ObjectId('648d40addf6d5f55cd83a674')
borough: "Brooklyn"
cuisine: "Hamburgers"
name: "Wendy'S"
restaurant_id: "30112340"
```

Question 1(9)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 5 documents

```
_id: ObjectId('648d40addf6d5f55cd83b25e')
name: "Circle In The Square Theatre"
borough: "Manhattan"
cuisine: "American "
avgScore: 0.6666666666666666
```

```
_id: ObjectId('648d40addf6d5f55cd83b112')
name: "Gold Bar B"
borough: "Manhattan"
cuisine: "American "
avgScore: 1
```

Question 1(1)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 5 documents

```
_id: ObjectId('648d40addf6d5f55cd83b25e')
name: "Circle In The Square Theatre"
borough: "Manhattan"
cuisine: "American "
avgScore: 0.6666666666666666
```

```
_id: ObjectId('648d40addf6d5f55cd83b112')
name: "Gold Bar B"
borough: "Manhattan"
cuisine: "American "
avgScore: 1
```

Question 1(11)

PIPELINE OUTPUT

Sample of 1 document

OUTPUT OPTIONS

```
restaurantCount: 3
```

Question 2(1)

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 2 documents

```
_id: ObjectId('648d9e90df6d5f55cd83b605')
InvoiceNo: 536394
InvoiceDate: "1/12/2010 10:39"
```

```
_id: ObjectId('648d9e90df6d5f55cd83b69f')
InvoiceNo: 536406
InvoiceDate: "1/12/2010 11:33"
```

Question 2(2)

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 10 documents

```
_id: ObjectId('648d9e90df6d5f55cd83b76d')
StockCode: 22139
UnitPrice: 0
```

```
_id: ObjectId('648d9e90df6d5f55cd83b700')
StockCode: 20668
UnitPrice: 0.12
```

Question 2(3)

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 1 document

```
_id: null
totalQuantity: 6
```

Question 2(4)

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 1 document

```
_id: null
totalQuantity: 57
```

Question 2(5)

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 10 documents

```
_id: 22910
totalQuantity: 87
```

```
_id: 84880
totalQuantity: 36
```

```
_id: 22217
totalQuantity: 12
```

Question 2(6)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 10 documents

```
_id: 17021
maxQuantity: 600
```

```
_id: "84563A"
maxQuantity: 1
```

```
_id: "84596E"
maxQuantity: 1
```

```
_id: "35004C"
maxQuantity: 48
```

Question 2(7)

PIPELINE OUTPUT OUTPUT OPTIONS

Sample of 10 documents

```
_id: 536366
maxQuantity: 6
StockCode: 22633
Description: "HAND WARMER UNION JACK"
```

```
_id: 536397
maxQuantity: 48
StockCode: "35004B"
Description: "SET OF 3 BLACK FLYING DUCKS"
```