Chapter 3: Slow Queries

Lab: Analyse Profiler Data

Back to the Question

This lab asks 2 different questions:

1. Which index was created?
2. When was it created?

To determine after which operation we can see that a change on our query execution, we will have to look for a change on the `planSummary` of our profile data, for our queries.

COPY

```
db.profiler_data.find({planSummary: {$ne:
"COLLSCAN"}}).sort({ts:1}).limit(1)
{
  "planSummary": "IXSCAN { address.zip: 1 }",
  "execStats": {
    "stage": "FETCH",
    "nReturned": 41113,
    "executionTimeMillisEstimate": 20,
    "works": 41114,
    "advanced": 41113,
    "needTime": 0,
    "needYield": 0,
    "saveState": 321,
    "restoreState": 321,
    "isEOF": 1,
    "invalidates": 0,
    "docsExamined": 41113,
```

```
    "alreadyHasObj": 0,
    "inputStage": {
      "stage": "IXSCAN",
      "nReturned": 41113,
      "executionTimeMillisEstimate": 10,
      "works": 41114,
      "advanced": 41113,
      "needTime": 0,
      "needYield": 0,
      "saveState": 321,
      "restoreState": 321,
      "isEOF": 1,
      "invalidates": 0,
      "keyPattern": {
        "address.zip": 1
      },
      "indexName": "address.zip_1",
      "isMultiKey": false,
      "multiKeyPaths": {
        "address.zip": [ ]
      },
      "isUnique": false,
      "isSparse": false,
      "isPartial": false,
      "indexVersion": 2,
      "direction": "forward",
      "indexBounds": {
        "address.zip": [
          "[\"18648-1078\", \"18648-1078\"]"
        ]
      },
      "keysExamined": 41113,
      "seeks": 1,
      "dupsTested": 0,
      "dupsDropped": 0,
      "seenInvalidated": 0
    }
  },
  "ts": ISODate("2017-03-06T22:56:30.407Z")
}
```


The creation of an index implies that we should no longer use **COLLSCAN** to support our queries. Which means that, if we look for the first occurrence of a `planSummary` that is not

equals to `COLLSCAN`, we will find the a point where the query is already running, supported by an index.

To avoid being overloaded by the profiler document we should project the most relevant fields for our task in hands:

- From `originatingCommand` or `execStats`, we can tell which command was optimized, in our case a query.
- `ts` will determine the timestamp on which the **COLLSCAN** is no longer present
- `planSummary` will tell us which new query plan summary is there. So we can see which index we are now using. In our case `{ address.zip: 1 }`

This answers the first part of our problem, which index was created:

```
db.employees.createIndex({'address.zip': 1})
```

COPY

The second part requires a bit more of digging.

If we know that after a certain point, our query become optimized, we need go back and see when was the last `ts` where the `planSummary` was still a **COLLSCAN**

To get this we would need the following query:

```
db.profiler_data.find({"ts": {$lt: ISODate("2017-03-0
```

COPY

This will give us all the operations, that happened before the detected **IXSCAN**. Then, we will need to find the latest `db.employees.find({"address.zip": XXX })` **COLLSCAN** query :

COPY

```
{
"op": "getmore",
"ns": "m312.employees",
"query": {
  "getMore": NumberLong("30684393245"),
  "collection": "employees"
},
"originatingCommand": {
  "find": "employees",
  "filter": {
    "address.zip": "03686-4697"
  }
```

```
},
"cursorid": 30684393245,
"keysExamined": 0,
"docsExamined": 617053,
"cursorExhausted": true,
"numYield": 4821,
"locks": {
  "Global": {
    "acquireCount": {
      "r": NumberLong("9644")
    }
  },
  "Database": {
    "acquireCount": {
      "r": NumberLong("4822")
    }
  },
  "Collection": {
    "acquireCount": {
      "r": NumberLong("4822")
    }
  }
},
"nreturned": 20491,
"responseLength": 8222727,
"protocol": "op_query",
"millis": 405,
"planSummary": "COLLSCAN",
"execStats": {
  "stage": "COLLSCAN",
  "filter": {
    "address.zip": {
      "$eq": "03686-4697"
    }
  },
  "nReturned": 62390,
  "executionTimeMillisEstimate": 276,
  "works": 2888270,
  "advanced": 62390,
  "needTime": 2825879,
  "needYield": 0,
  "saveState": 22567,
  "restoreState": 22567,
```

```
  "isEOF": 1,
  "invalidates": 0,
  "direction": "forward",
  "docsExamined": 2888268
},
"ts": ISODate("2017-03-06T22:55:55.065Z"),
"client": "192.168.14.1",
"allUsers": [ ],
"user": ""
}
```

And therefore the `ts` value is:

```
ISODate("2017-03-06T22:55:55.065Z")
```

COPY

The correct answer would be

COPY

```
db.employees.createIndex({'address.zip': 1})

After 2017-03-06T22:55:55.065Z
```

Proceed to next section