使用mongodb語法計算每個欄位出現的次數

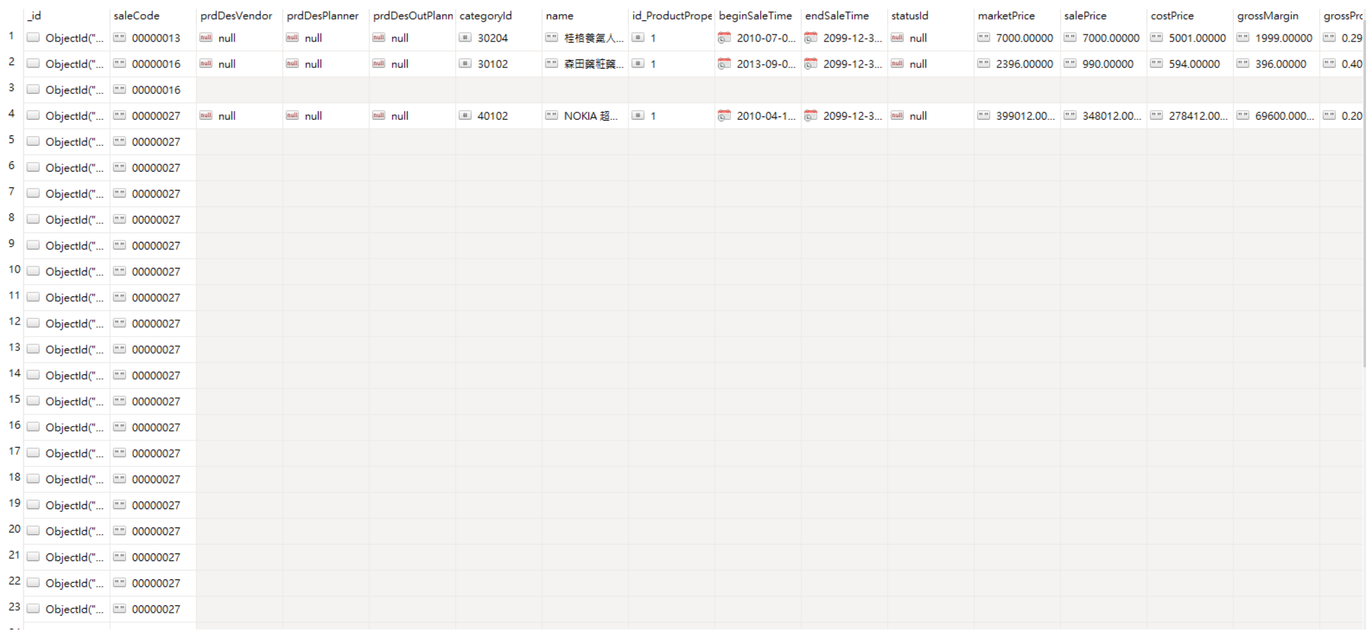
背景

mongodb中有一個collection用來記錄每次有sql欄位變動的歷程，每一次變動都會寫入一筆資料，

需求為希望統計此collection中每個欄位出現的次數，以便後續做資料分析，因此需要計算每個欄位出現的次數。

Log_salesmix_lite collection

用來記錄sql欄位變動的collection，每次sql欄位變動都會寫入一筆資料，因此可以用來計算每個欄位出現的次數。



	_id	saleCode	prdDesVendor	prdDesPlanner	prdDesOutPlann	categoryId	name	id_ProductPrope	beginSaleTime	endSaleTime	statusId	marketPrice	salePrice	costPrice	grossMargin	grossPrc
1	ObjectId("...	00000013	null	null	null	30204	桂格養氣人...	1	2010-07-0...	2099-12-3...	null	7000.00000	7000.00000	5001.00000	1999.00000	0.29
2	ObjectId("...	00000016	null	null	null	30102	森田藥粧藥...	1	2013-09-0...	2099-12-3...	null	2396.00000	990.00000	594.00000	396.00000	0.40
3	ObjectId("...	00000016														
4	ObjectId("...	00000027	null	null	null	40102	NOKIA 超...	1	2010-04-1...	2099-12-3...	null	399012.00...	348012.00...	278412.00...	69600.000...	0.20
5	ObjectId("...	00000027														
6	ObjectId("...	00000027														
7	ObjectId("...	00000027														
8	ObjectId("...	00000027														
9	ObjectId("...	00000027														
10	ObjectId("...	00000027														
11	ObjectId("...	00000027														
12	ObjectId("...	00000027														
13	ObjectId("...	00000027														
14	ObjectId("...	00000027														
15	ObjectId("...	00000027														
16	ObjectId("...	00000027														
17	ObjectId("...	00000027														
18	ObjectId("...	00000027														
19	ObjectId("...	00000027														
20	ObjectId("...	00000027														
21	ObjectId("...	00000027														
22	ObjectId("...	00000027														
23	ObjectId("...	00000027														

嘗試用nodejs程式計算

一開始想法是用nodejs程式連結mongodb撈出資料後進行計算，但後來發現效能太差，因此改用mongodb語法進行計算。

以下語法以及說明

note : 2023-11-05T00:00:00+0800 代表+8時區

目標

原有的資料結構 array，每個key代表當次有更改的欄位以及對應的數值

- 每次修改內容不同，所以每個json的key以及數量也會不同

```
[{
    "_id" : ObjectId("58c8c066275ae7289cad62ce"),
    "saleCode" : "00000013",
    "prdDesVendor" : null,
    "prdDesPlanner" : null,
    "prdDesOutPlanner" : null,
    "categoryId" : NumberLong(30204),
    "name" : "桂格養氣人蔘aa=",
    "id_ProductProperty" : NumberLong(1),
    "beginSaleTime" : ISODate("2010-07-01T16:00:00.000Z"),
    "endSaleTime" : ISODate("2099-12-30T16:00:00.000Z"),
    "statusId" : null,
    "marketPrice" : "7000.00000",
    "salePrice" : "7000.00000",
    "costPrice" : "5001.00000",
    "grossMargin" : "1999.00000",
    "grossProfit" : "0.29000",
    "presentDescription" : null,
    "shelveReservation" : 1,
    "canReturn" : 1,
    "canChange" : 1,
    "paymentPeriod" : 60,
    "installmentSetting" : 1,
    "shipmentTaxType" : null,
    "isReceipt" : null,
    "employeePrice" : "6990.00",
    "saleBonus" : null,
    "canBuyByProfit" : 1,
    "canBuyByAccount" : null,
    ...
    }
    ,...
    ]
```

目標

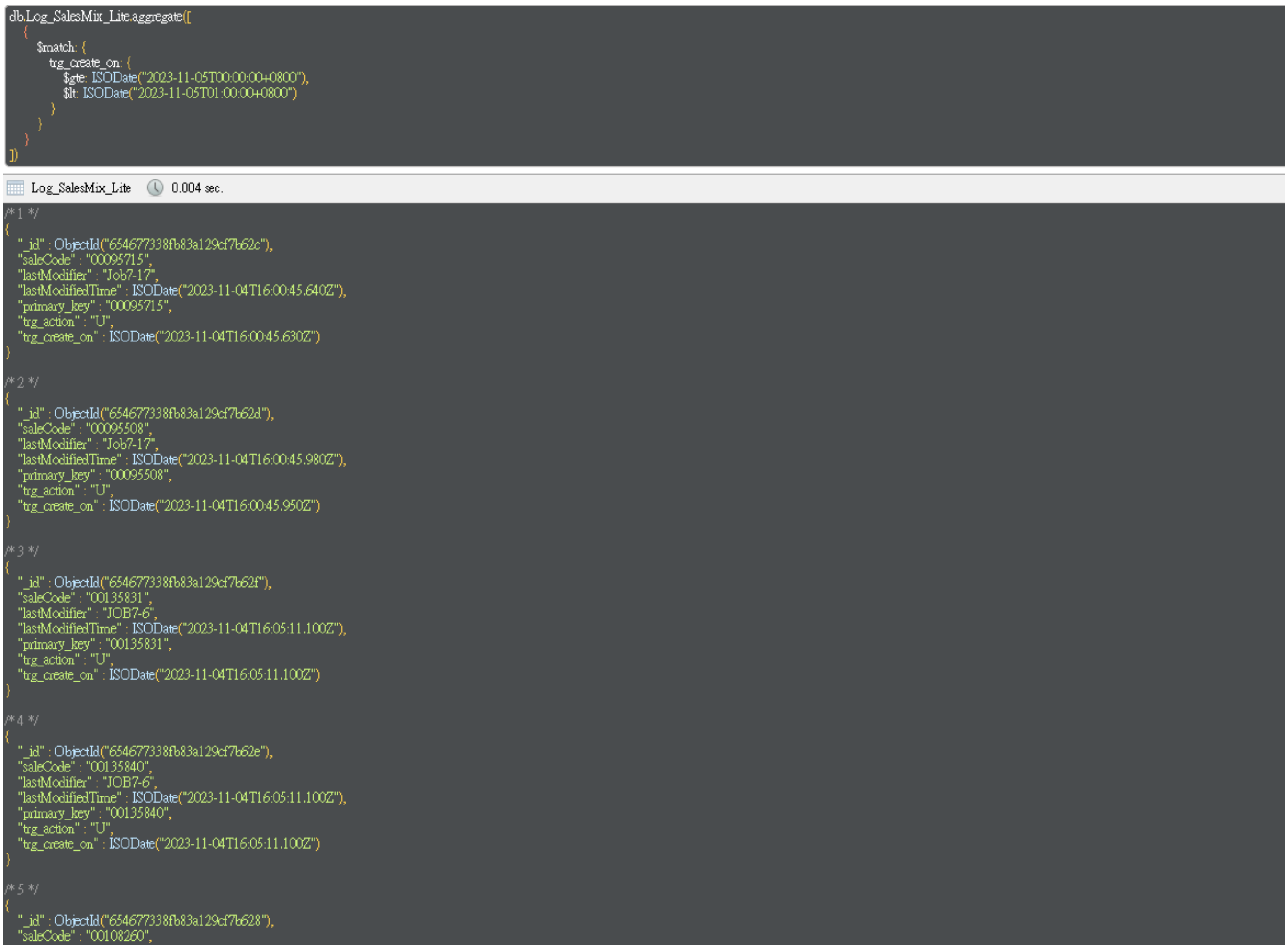
```
{
    "lastModifier" : 9.0,
    "lastModifiedTime" : 8.0,
    "primary_key" : 9.0,
    "isExpressProduct2" : 2.0,
    "fugoContractNo" : 1.0,
    "isExpressProduct" : 2.0,
    "trg_action" : 9.0,
    "_id" : 9.0,
    "trg_create_on" : 9.0,
    "saleCode" : 9.0,
    ...
}
```

使用aggregate 方法做pipeline計算，需設計各種stage達到我們想要的結果

Step.1 撈出需要計算的資料

要撈出`Log_SalesMix_Lite`此collection 中`trg_create_on`欄位在`2023-11-05T00:00:00+0800`到`2023-11-05T01:00:00+0800`之間的資料

```
    {
            $match: {
                trg_create_on: {
                    $gte: ISODate("2023-11-05T00:00:00+0800"),
                    $lt: ISODate("2023-11-05T01:00:00+0800")
                }
            }
        },
```



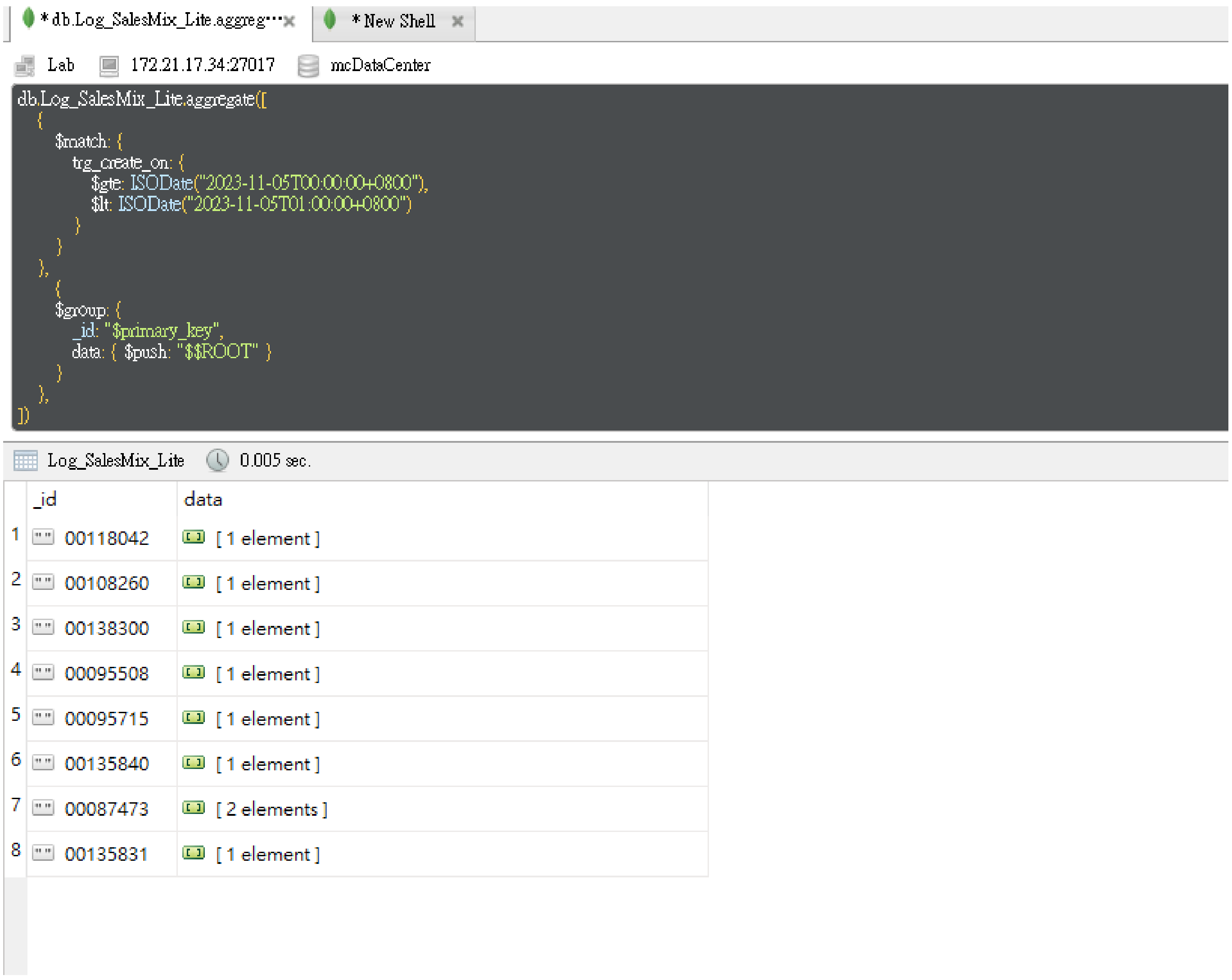
```
db.Log_SalesMix_Lite.aggregate([
  {
    $match: {
      trg_create_on: {
        $gte: ISODate("2023-11-05T00:00:00+0800"),
        $lt: ISODate("2023-11-05T01:00:00+0800")
      }
    }
  }
])
```

Log_SalesMix_Lite 0.004 sec.

```
/* 1 */
{
  "_id" : ObjectId("654677338fb83a129cf7b62c"),
  "saleCode" : "00095715",
  "lastModifier" : "Job7-17",
  "lastModifiedTime" : ISODate("2023-11-04T16:00:45.640Z"),
  "primary_key" : "00095715",
  "trg_action" : "U",
  "trg_create_on" : ISODate("2023-11-04T16:00:45.630Z")
}

/* 2 */
{
  "_id" : ObjectId("654677338fb83a129cf7b62d"),
  "saleCode" : "00095508",
  "lastModifier" : "Job7-17",
  "lastModifiedTime" : ISODate("2023-11-04T16:00:45.980Z"),
  "primary_key" : "00095508",
  "trg_action" : "U",
  "trg_create_on" : ISODate("2023-11-04T16:00:45.950Z")
}

/* 3 */
{
  "_id" : ObjectId("654677338fb83a129cf7b62f"),
  "saleCode" : "00135831",
  "lastModifier" : "JOB7-6",
  "lastModifiedTime" : ISODate("2023-11-04T16:05:11.100Z"),
  "primary_key" : "00135831",
  "trg_action" : "U",
  "trg_create_on" : ISODate("2023-11-04T16:05:11.100Z")
}

/* 4 */
{
  "_id" : ObjectId("654677338fb83a129cf7b62e"),
  "saleCode" : "00135840",
  "lastModifier" : "JOB7-6",
  "lastModifiedTime" : ISODate("2023-11-04T16:05:11.100Z"),
  "primary_key" : "00135840",
  "trg_action" : "U",
  "trg_create_on" : ISODate("2023-11-04T16:05:11.100Z")
}

/* 5 */
{
  "_id" : ObjectId("654677338fb83a129cf7b628"),
  "saleCode" : "00108260",
```

Step.2 根據primary_key分組，並把同個key的data放進一個array中

```
   {
            $group: {
                _id: "$primary_key",
                data: { $push: "$$ROOT" }
            }
        },
```


* db.Log_SalesMix_Lite.aggreg··· × | * New Shell ×

Lab | 172.21.17.34:27017 | mcDataCenter

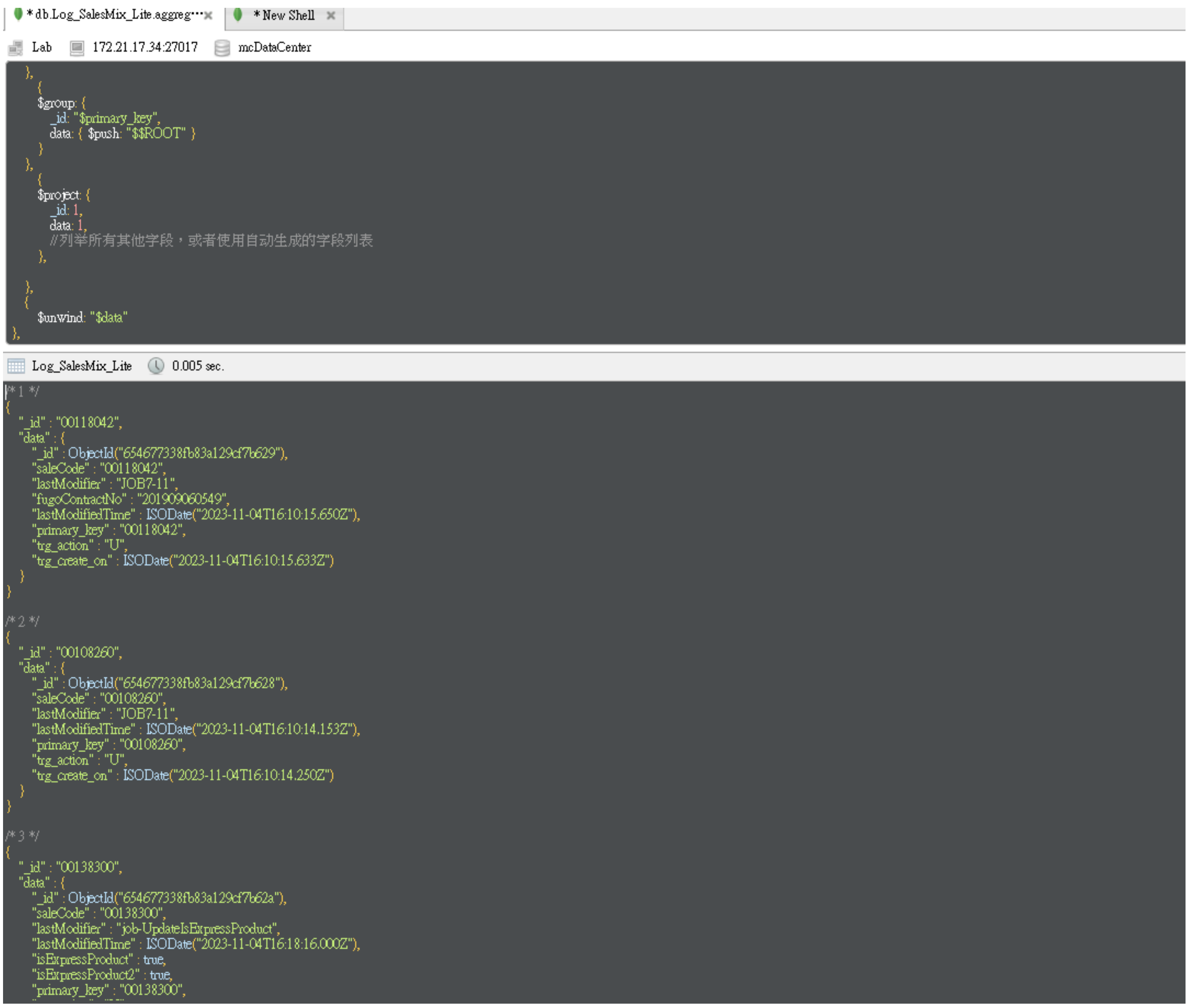
```
db.Log_SalesMix_Lite.aggregate([
   {
      $match: {
         trg_create_on: {
            $gte: ISODate("2023-11-05T00:00:00+0800"),
            $lt: ISODate("2023-11-05T01:00:00+0800")
         }
      }
   },
      {
      $group: {
         _id: "$primary_key",
         data: { $push: "$$ROOT" }
      }
   },
])
```

Log_SalesMix_Lite | 0.005 sec.

	_id	data
1	00118042	[1 element]
2	00108260	[1 element]
3	00138300	[1 element]
4	00095508	[1 element]
5	00095715	[1 element]
6	00135840	[1 element]
7	00087473	[2 elements]
8	00135831	[1 element]

Step.3 將data array轉成object

```
{
        $unwind: "$data"
},
```

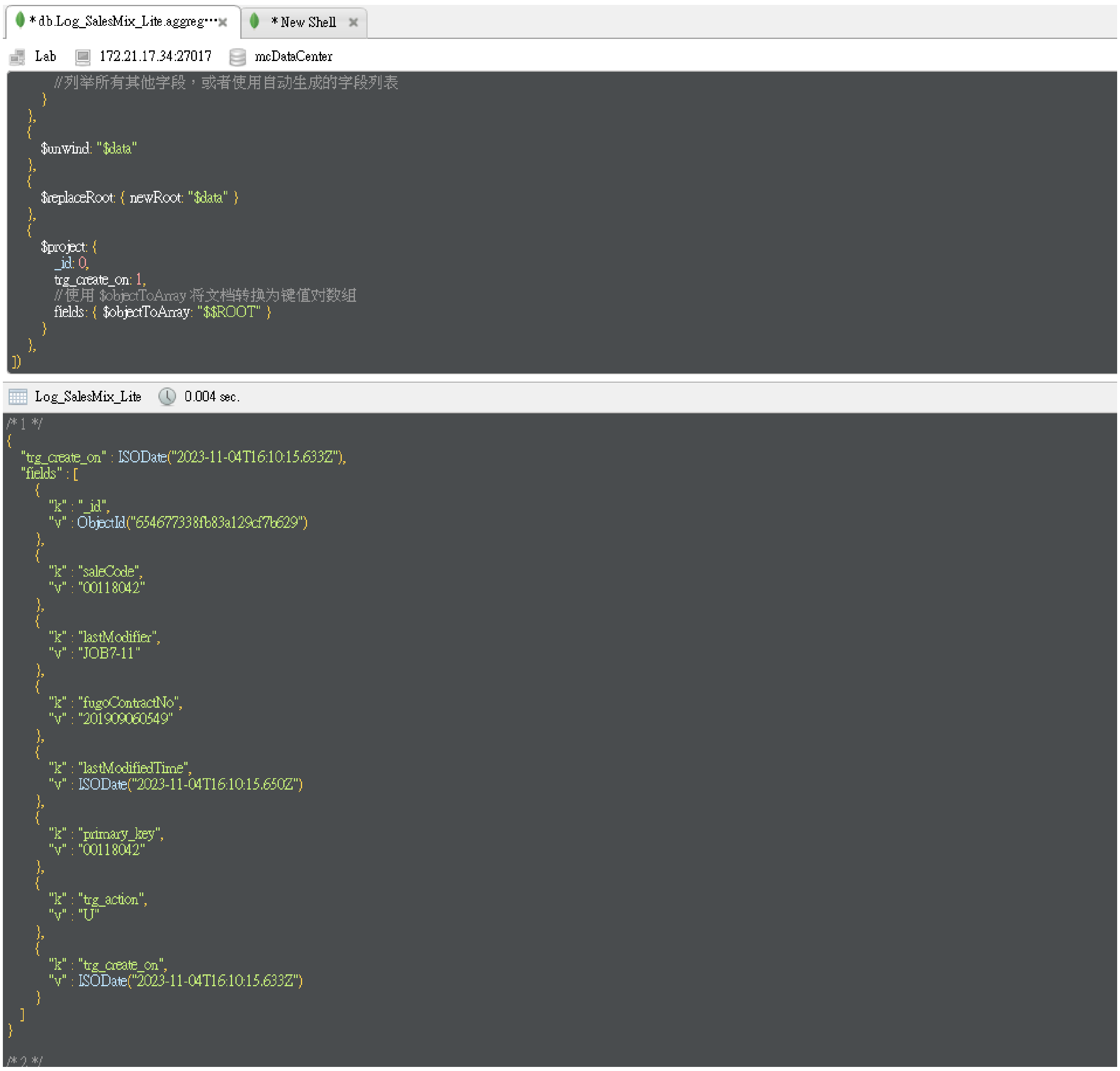


```
    },
      {
      $group: {
        _id: "$primary_key",
        data: { $push: "$$ROOT" }
      }
    },
      {
      $project: {
        _id: 1,
        data: 1,
        //列举所有其他字段，或者使用自动生成的字段列表
      },

    },
    {
      $unwind: "$data"
 },
```

```
/* 1 */
{
  "_id" : "00118042",
  "data" : {
    "_id" : ObjectId("654677338fb83a129cf7b629"),
    "saleCode" : "00118042",
    "lastModifier" : "JOB7-11",
    "fugoContractNo" : "201909060549",
    "lastModifiedTime" : ISODate("2023-11-04T16:10:15.650Z"),
    "primary_key" : "00118042",
    "trg_action" : "U",
    "trg_create_on" : ISODate("2023-11-04T16:10:15.633Z")
  }
}

/* 2 */
{
  "_id" : "00108260",
  "data" : {
    "_id" : ObjectId("654677338fb83a129cf7b628"),
    "saleCode" : "00108260",
    "lastModifier" : "JOB7-11",
    "lastModifiedTime" : ISODate("2023-11-04T16:10:14.153Z"),
    "primary_key" : "00108260",
    "trg_action" : "U",
    "trg_create_on" : ISODate("2023-11-04T16:10:14.250Z")
  }
}

/* 3 */
{
  "_id" : "00138300",
  "data" : {
    "_id" : ObjectId("654677338fb83a129cf7b62a"),
    "saleCode" : "00138300",
    "lastModifier" : "job-UpdateIsExpressProduct",
    "lastModifiedTime" : ISODate("2023-11-04T16:18:16.000Z"),
    "isExpressProduct" : true,
    "isExpressProduct2" : true,
    "primary_key" : "00138300",
```

Step.4 重新組合object，並且將每個欄位轉成key-value的形式

```
    {
        $replaceRoot: { newRoot: "$data" }
    },
    {
        $project: {
            _id: 1,
            trg_create_on: 1,
            // 使用 $objectToArray 将文档转换为键值对数组
            fields: { $objectToArray: "$$ROOT" }
        }
    },
    {
        $unwind: "$fields"
    },
```



* db.Log_SalesMix_Lite.aggreg··· | * New Shell

Lab | 172.21.17.34:27017 | mcDataCenter

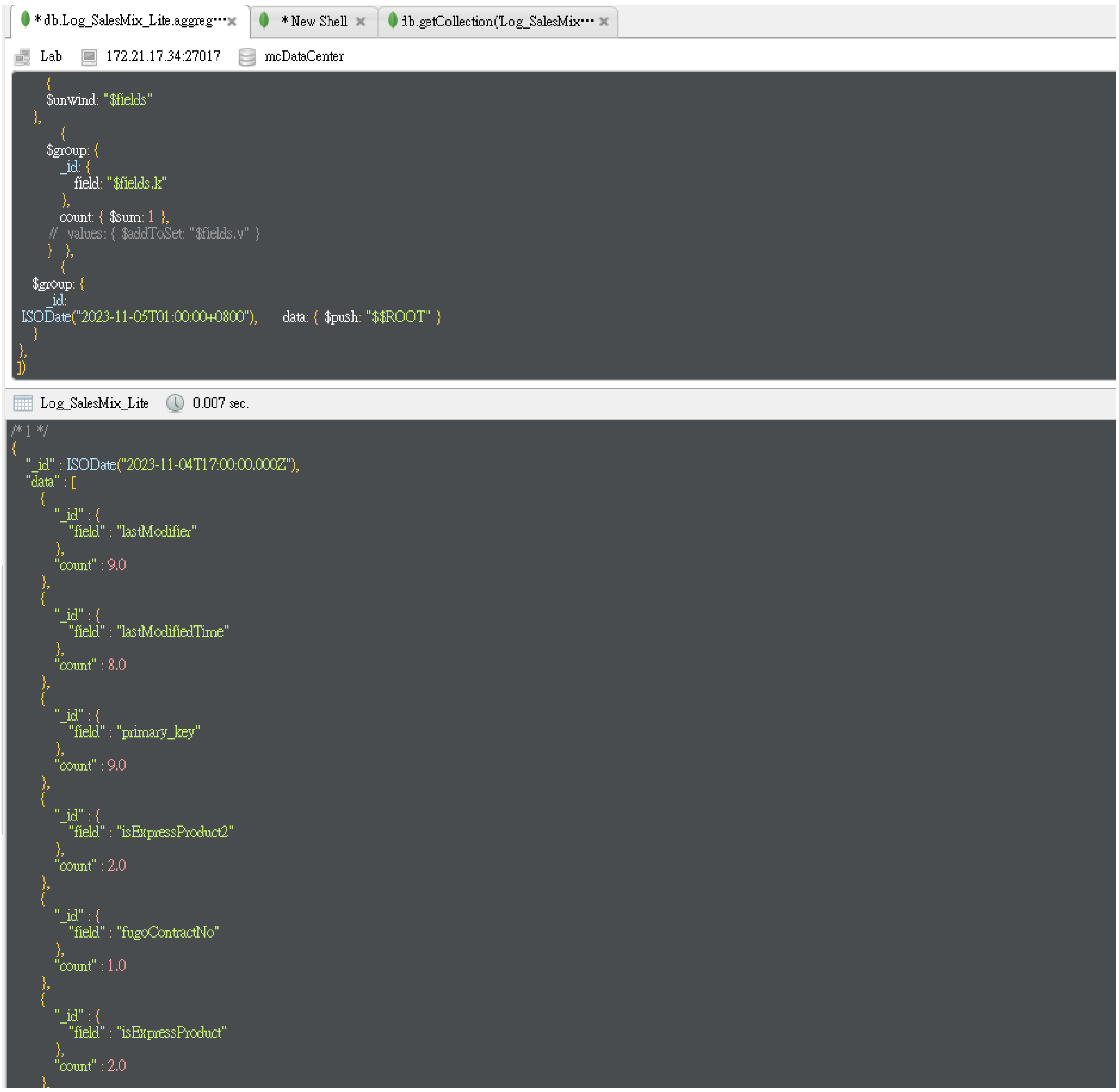
```
        //列举所有其他字段，或者使用自动生成的字段列表
      }
    },
    {
      $unwind: "$data"
    },
    {
      $replaceRoot: { newRoot: "$data" }
    },
    {
      $project: {
        _id: 0,
        trg_create_on: 1,
        //使用 $objectToArray 将文档转换为键值对数组
        fields: { $objectToArray: "$$ROOT" }
      }
    },
])
```

Log_SalesMix_Lite | 0.004 sec.

```
/* 1 */
{
  "trg_create_on" : ISODate("2023-11-04T16:10:15.633Z"),
  "fields" : [
    {
      "k" : "_id",
      "v" : ObjectId("654677338fb83a129cf7b629")
    },
    {
      "k" : "saleCode",
      "v" : "00118042"
    },
    {
      "k" : "lastModifier",
      "v" : "JOB7-11"
    },
    {
      "k" : "fugoContractNo",
      "v" : "201909060549"
    },
    {
      "k" : "lastModifiedTime",
      "v" : ISODate("2023-11-04T16:10:15.650Z")
    },
    {
      "k" : "primary_key",
      "v" : "00118042"
    },
    {
      "k" : "trg_action",
      "v" : "U"
    },
    {
      "k" : "trg_create_on",
      "v" : ISODate("2023-11-04T16:10:15.633Z")
    }
  ]
}

/* 2 */
```

Step.5 根據key做group並計算次數

```
{
  $group: {
     _id: {
        field: "$fields.k"
            },
            count: { $sum: 1 },
         //   values: { $addToSet: "$fields.v" }
        }   },
    {
    $group: {
        _id:
    ISODate("2023-11-05T00:00:00+0800"),
    data: { $push: "$$ROOT" }
    }
},
```

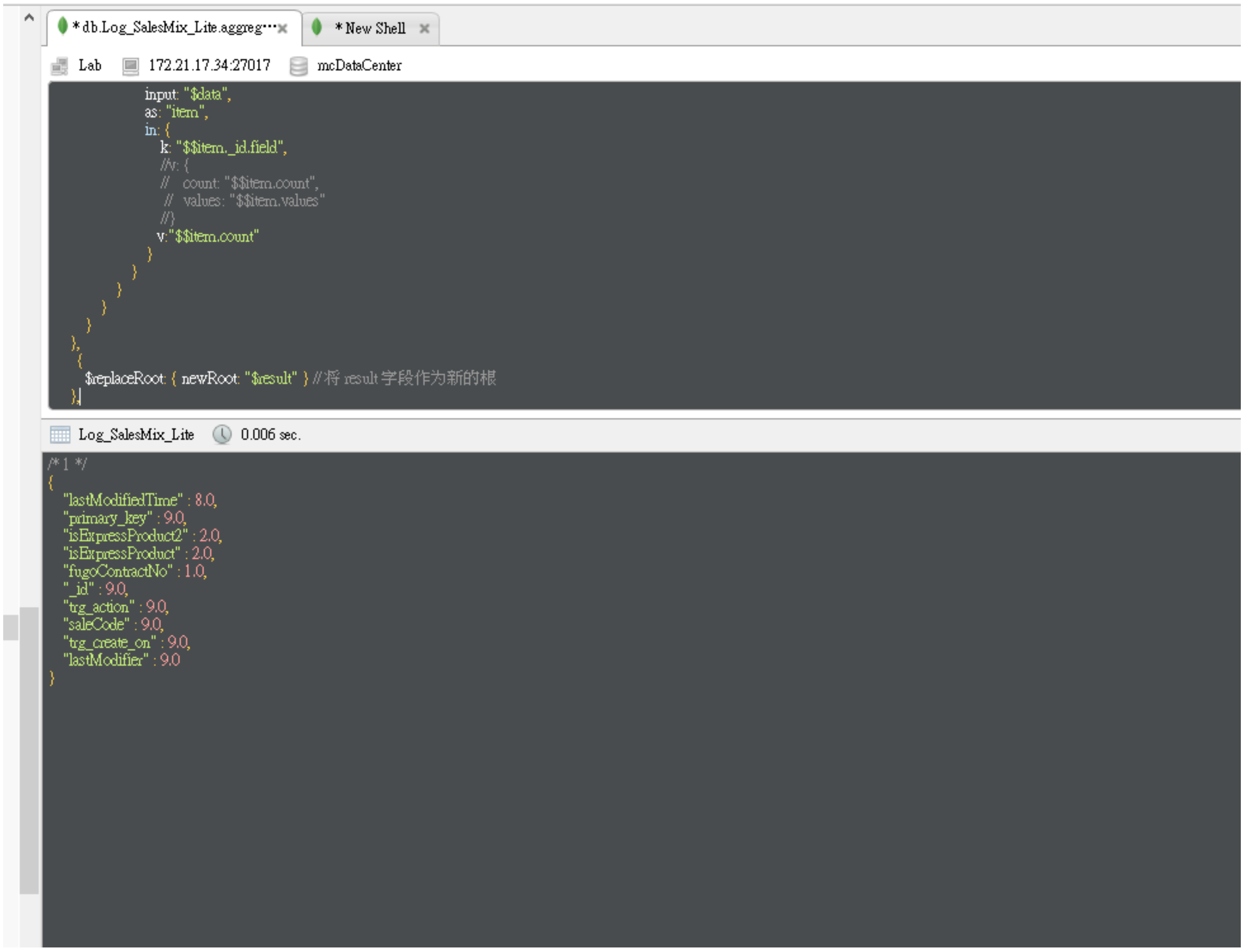


```
* db.Log_SalesMix_Lite.aggreg···    * New Shell    db.getCollection('Log_SalesMix···
Lab    172.21.17.34:27017    mcDataCenter
    {
    $unwind: "$fields"
  },
      {
    $group: {
       _id: {
          field: "$fields.k"
       },
       count: { $sum: 1 },
    //   values: { $addToSet: "$fields.v" }
    }  },
       {
  $group: {
     _id:
 ISODate("2023-11-05T01:00:00+0800"),      data: { $push: "$$ROOT" }
  }
},
])
```

```
Log_SalesMix_Lite    0.007 sec.
/* 1 */
{
    "_id" : ISODate("2023-11-04T17:00:00.000Z"),
    "data" : [
        {
            "_id" : {
                "field" : "lastModifier"
            },
            "count" : 9.0
        },
        {
            "_id" : {
                "field" : "lastModifiedTime"
            },
            "count" : 8.0
        },
        {
            "_id" : {
                "field" : "primary_key"
            },
            "count" : 9.0
        },
        {
            "_id" : {
                "field" : "isExpressProduct2"
            },
            "count" : 2.0
        },
        {
            "_id" : {
                "field" : "fugoContractNo"
            },
            "count" : 1.0
        },
        {
            "_id" : {
                "field" : "isExpressProduct"
            },
            "count" : 2.0
        },
```

Step6. 重新組合資料並取出所需要的data

```
{
        $project: {
            _id: 1,
            result: {
                $arrayToObject: {
                    $map: {
                        input: "$data",
                        as: "item",
                        in: {
                            k: "$$item._id.field",
                            //v: {
                            //    count: "$$item.count",
                             //   values: "$$item.values"
                    //}
                   v:"$$item.count"
                 }
               }
             }
           }
         }
     },
     {
         $replaceRoot: { newRoot: "$result" } // 将 result 字段作为新的根
     },
```



```
               input: "$data",
               as: "item",
               in: {
                 k: "$$item._id.field",
                 //v: {
                 //   count: "$$item.count",
                  //  values: "$$item.values"
                 //}
                v:"$$item.count"
              }
            }
          }
        }
      }
   },
    {
     $replaceRoot: { newRoot: "$result" } //将 result 字段作为新的根
   },
```

```
/* 1 */
{
    "lastModifiedTime" : 8.0,
    "primary_key" : 9.0,
    "isExpressProduct2" : 2.0,
    "isExpressProduct" : 2.0,
    "fugoContractNo" : 1.0,
    "_id" : 9.0,
    "trg_action" : 9.0,
    "saleCode" : 9.0,
    "trg_create_on" : 9.0,
    "lastModifier" : 9.0
}
```

Step7. 加入新的欄位並將結果寫入新的collection

```
 {
   $addFields: { _id: ISODate("2023-11-05T00:00:00+0800")  } },
  {
        $merge: {
             into: "Log_SalesMix_Summary" // 将结果添加到现有集合
       }
   }
```

以下為全部語法

```
db.Log_SalesMix_Lite.aggregate([
    {
        $match: {
            trg_create_on: {
                $gte: ISODate("2023-11-05T00:00:00+0800"),
                $lt: ISODate("2023-11-05T01:00:00+0800")
            }
        }
    },
    {
        $group: {
            _id: "$primary_key",
            data: { $push: "$$ROOT" }
        }
    },
    {
        $project: {
            _id: 1,
            data: 1,
            // 列举所有其他字段，或者使用自动生成的字段列表
        }
    },
    {
        $unwind: "$data"
    },
    {
        $replaceRoot: { newRoot: "$data" }
    },
    {
        $project: {
            _id: 0,
            trg_create_on: 1,
            // 使用 $objectToArray 将文档转换为键值对数组
            fields: { $objectToArray: "$$ROOT" }
        }
    },
    {
        $unwind: "$fields"
    },
        {
        $group: {
            _id: {
                field: "$fields.k"
            },
            count: { $sum: 1 },
       //   values: { $addToSet: "$fields.v" }
        }   },
    {
    $group: {
        _id:
 ISODate("2023-11-05T01:00:00+0800"),       data: { $push: "$$ROOT" }
```

```
            }
        },
        {
                $project: {
                    _id: 1,
                    result: {
                        $arrayToObject: {
                            $map: {
                                input: "$data",
                                as: "item",
                                in: {
                                    k: "$$item._id.field",
                                    //v: {
                                    //     count: "$$item.count",
                                     //    values: "$$item.values"
                                    //}
                                   v:"$$item.count"
                                }
                            }
                        }
                    }
                }
            },
            {
                $replaceRoot: { newRoot: "$result" } // 将 result 字段作为新的根
            },
            { $addFields: { _id: ISODate("2023-11-05T00:00:00+0800")  } },
         {
                $merge: {
                    into: "Log_SalesMix_Summary" // 将结果添加到现有集合
              }
          }
        ])
```

後來用C#語法改寫以便可以加入排程

```
using System;
using System.Linq;
using MongoDB.Bson;
using MongoDB.Driver;


namespace runMongoDailyJob
{
    internal class Program
    {
        static void Main(string[] args)
        {

            string mongodbHost =
Environment.GetEnvironmentVariable("MONGODB_HOST") ??
```

```
"admin:Aa123456@172.21.17.34"; // 默认值为 "localhost"
            string mongodbPort =
Environment.GetEnvironmentVariable("MONGODB_PORT") ?? "27017"; // 默认值为 "27017"
            string databaseName = "mcDataCenter"; // 你的数据库名称

            string connectionString = $"mongodb://{mongodbHost}:{mongodbPort}";

            MongoClient client = new MongoClient(connectionString);
            IMongoDatabase database = client.GetDatabase(databaseName);
            IMongoCollection<BsonDocument> collection =
database.GetCollection<BsonDocument>("Log_SalesMix_Lite");

            DateTime currentTime = DateTime.Now;
            DateTime currentHour = currentTime.Date.AddHours(currentTime.Hour);

            // 计算前一个小时的时间
            DateTime oneHourAgo = currentHour.AddHours(-1);

            var matchStage = new BsonDocument("$match", new BsonDocument
            {
    { "trg_create_on", new BsonDocument
        {
            { "$gte", oneHourAgo },
            { "$lt", currentHour }
        }
    }
});

            var sortStage = new BsonDocument("$sort", new BsonDocument
        {
            { "trg_create_on", 1 }
        });

            var groupStage = new BsonDocument("$group", new BsonDocument
        {
            { "_id", "$primary_key" },
            { "data", new BsonDocument("$push", "$$ROOT") }
        });

            var unwindStage1 = new BsonDocument("$unwind", "$data");

            var replaceRootStage = new BsonDocument("$replaceRoot", new
BsonDocument
        {
            { "newRoot", "$data" }
        });

            var projectStage = new BsonDocument("$project", new BsonDocument
        {
            { "_id", 0 },
            { "trg_create_on", 1 },
            { "fields", new BsonDocument("$objectToArray", "$$ROOT") }
        });
```

```
            var unwindStage2 = new BsonDocument("$unwind", "$fields");

            var groupStage2 = new BsonDocument("$group", new BsonDocument
        {
            { "_id", new BsonDocument("field", "$fields.k") },
            { "count", new BsonDocument("$sum", 1) }
        });

            var groupStage3 = new BsonDocument("$group", new BsonDocument
        {
            { "_id", oneHourAgo },
            { "data", new BsonDocument("$push", "$$ROOT") }
        });

            var projectStage2 = new BsonDocument("$project", new BsonDocument
        {
            { "_id", 1 },
            { "result", new BsonDocument("$arrayToObject", new BsonDocument
                {
                    { "$map", new BsonDocument
                        {
                            { "input", "$data" },
                            { "as", "item" },
                            { "in", new BsonDocument
                                {
                                    { "k", "$$item._id.field" },
                                    { "v", "$$item.count" }
                                }
                            }
                        }
                    }
                })
            }
        });

            var replaceRootStage2 = new BsonDocument("$replaceRoot", new
BsonDocument
        {
            { "newRoot", "$result" }
        });

            var addFieldsStage = new BsonDocument("$addFields", new BsonDocument
        {
            { "_id", oneHourAgo }
        });
            var mergeStage = new BsonDocument("$merge", new BsonDocument
        {
            { "into", "Log_SalesMix_Summary" }
        });
            var pipeline = new[]
            {
            matchStage, sortStage, groupStage, unwindStage1, replaceRootStage,
            projectStage, unwindStage2, groupStage2, groupStage3,
            projectStage2, replaceRootStage2, addFieldsStage,mergeStage
            // 添加其他阶段...
        };
            collection.Aggregate<BsonDocument>(pipeline)
        }
    }
}
```

在正式區執行一次約為3秒，可將約10萬筆的record所有欄位計算出來並寫入新的collection中

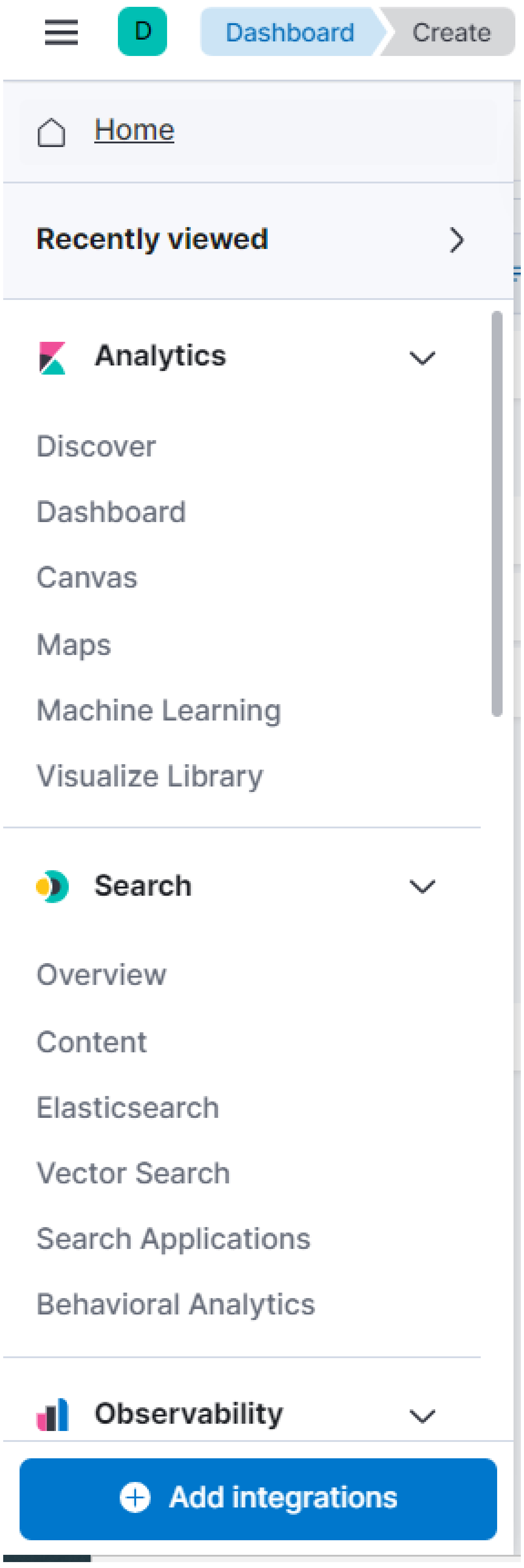
透過ELK視覺化

1. 修改C#程式碼如下，可以將結果寫入ELK中

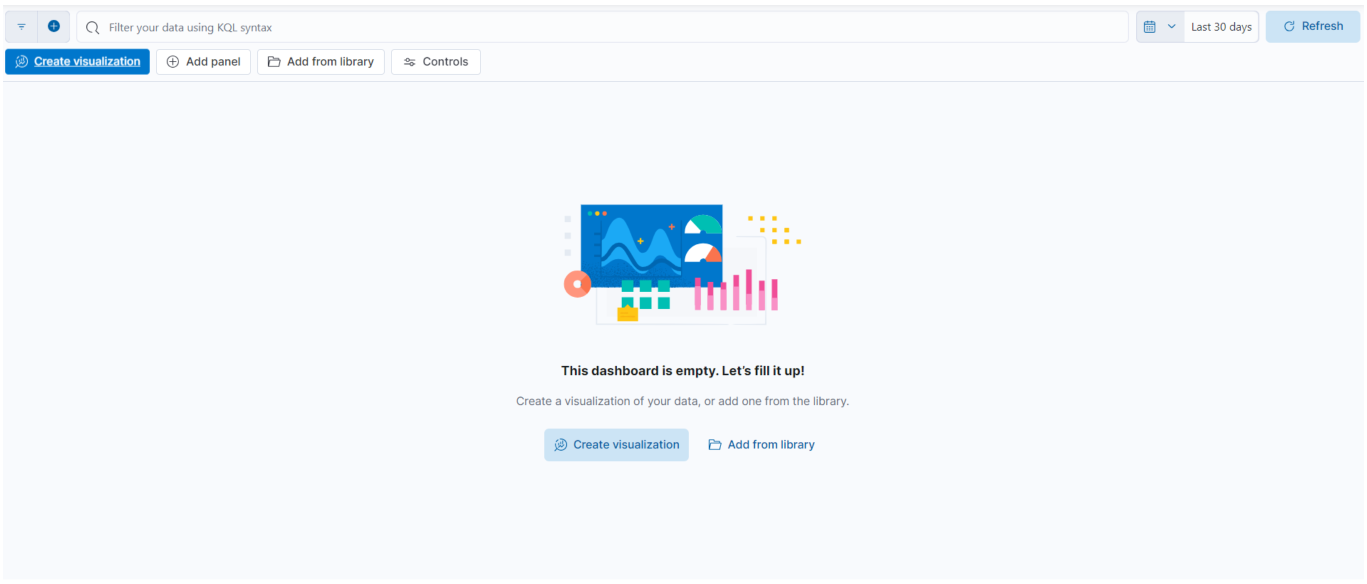
   參考

2. 建立dashboard 把結果視覺化，可依此觀察各欄位次數的變化

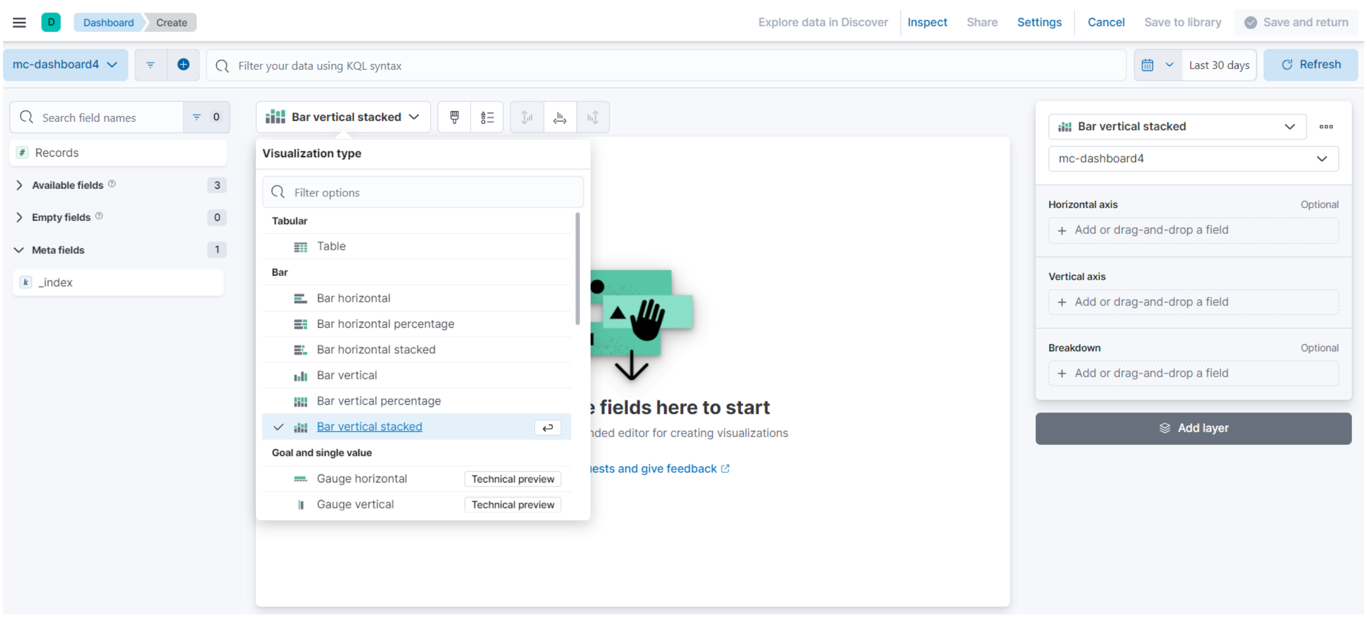
- 選取左方的dashboard



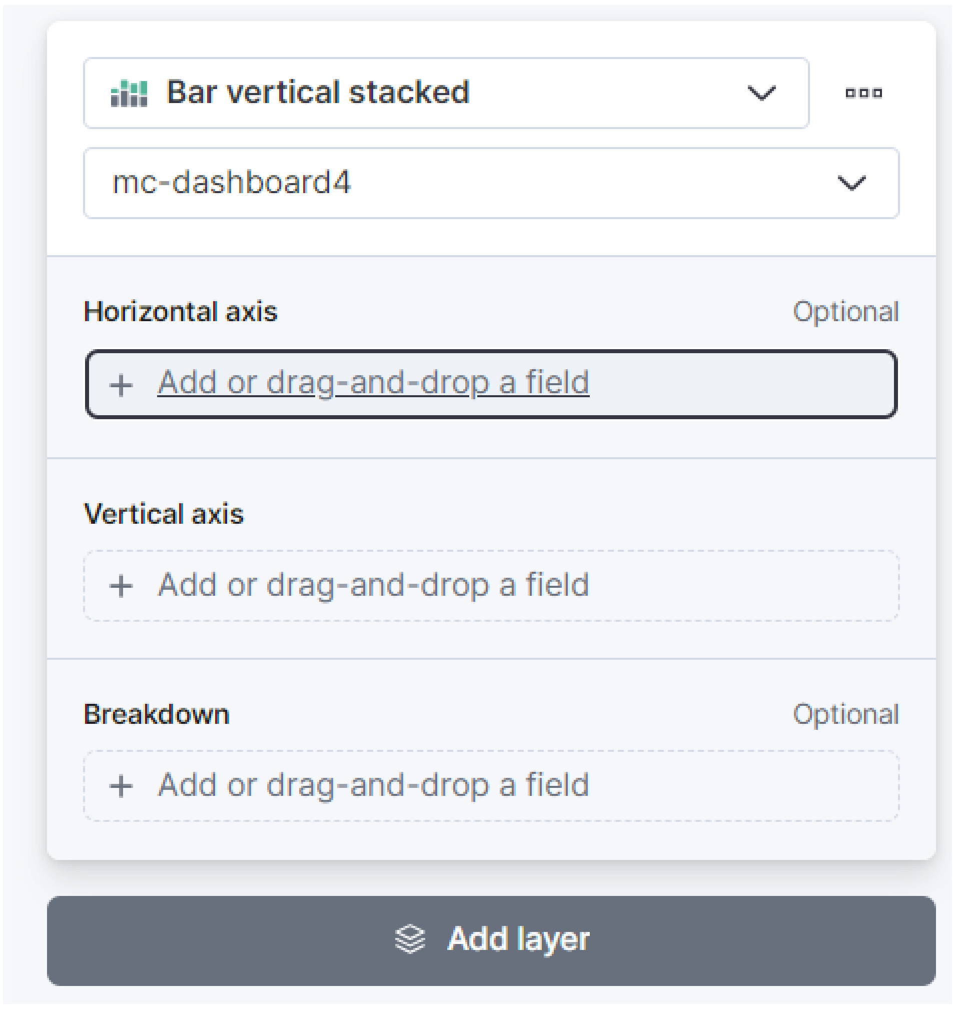
- Create Visualization


Filter your data using KQL syntax
Last 30 days
Refresh
Create visualization
Add panel
Add from library
Controls
This dashboard is empty. Let's fill it up!
Create a visualization of your data, or add one from the library.
Create visualization
Add from library

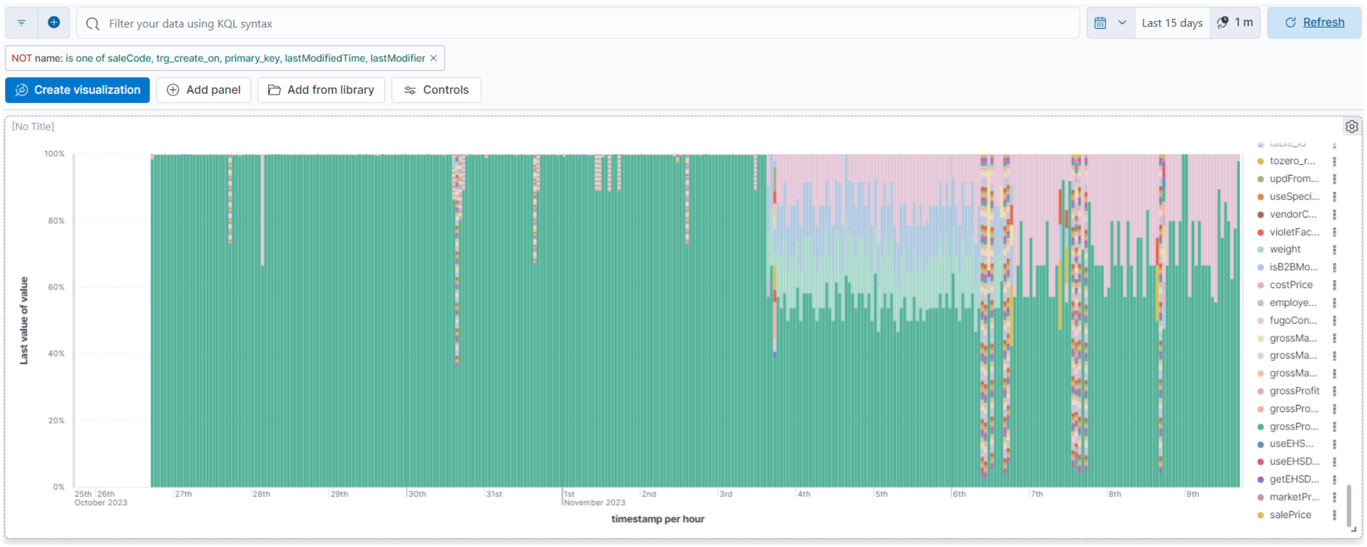
- 圖表類型選取 `Bar vertical percentage`



- 設定右方x-axis以及y-axis，breakdown的部分記得選`name`



- 結果


NOT name: is one of saleCode, trg_create_on, primary_key, lastModifiedTime, lastModifier