

PLAB 06/03/2025 TEMA 1 MongoDB

UAB

Universitat Autònoma de Barcelona

PROYECTO: Gestión de Restaurantes e Inspecciones

Oscar Pérez Yepes - 1565156

Roger Galdón Viladecans - 1565179

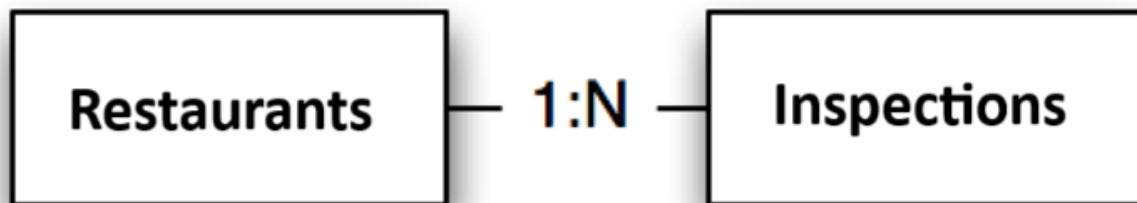
TAREAS OBLIGATORIAS (Hasta 4 puntos)	3
1. Diseño del esquema de la base de datos	3
2. Implementación de consultas en MongoDB	6
3. Uso de agregaciones	7
TAREAS AVANZADAS (Hasta 6 puntos)	10
1. Optimización del rendimiento	10
1.1 Consultas sin Indexar:	11
1.2 Implementación de index:	13
1.3 Consultas Indexadas:	13
2. Estrategias de escalabilidad	15
Outputs	18
[1] Comida de tipo "Chinese"	18
[2] Listar las inspecciones con violaciones, ordenadas por fecha.	23
[3] Encontrar restaurantes con una calificación superior a 4.	31
[4] Agrupar restaurantes por tipo de comida y calcular la calificación promedio.	37
[5] Contar el número de inspecciones por resultado y mostrar los porcentajes.	39
[6] Unir restaurantes con sus inspecciones utilizando \$lookup	40
[7] Restaurants sin index:	41
[8] consulta inspections sin index:	44
[9] consulta \$lookup sin index:	48
[10] consulta restaurants con index:	58
[11] consulta inspections con index:	61
[12] consulta \$lookup con index:	65

TAREAS OBLIGATORIAS (Hasta 4 puntos)

1. Diseño del esquema de la base de datos

- *Analizar la estructura de los datos y determinar el tipo de relación entre restaurantes e inspecciones (One-to-Few, One-to-Many, One-to-Millions).*
- *Justificar la elección de referencias (restaurant_id) en lugar de documentos embebidos. También puedes decidir crear una nueva colección que no utilice las referencias e incorpore los documentos embebidos. Justifica la decisión.*
- *Definir un esquema de validación para ambas colecciones.*

Analizando la estructura de los datos de las collections “restaurants” y “inspections” podemos determinar que se trata de una relación “One-to-Many.” Debido a que por cada restaurante tenemos varias inspecciones.



Se hace uso de las referencias (restaurant_id) en la “collection” de “inspections” por tal de poder referenciar el restaurante concreto cuando se realiza una “aggregation.” Sin embargo esto puede no ser lo mas optimo ya que necesitaremos hacer un “lookup” para juntar, como si se tratase de un “join”, diferentes “collections” a la hora de realizar “aggregations.” El uso de los lookup puede no ser lo más eficiente por lo que una opción alternativa sería generar una nueva “collection” con los documentos “embedded” de ambas “collections.”

```
{
  _id: <ObjectId>,
  username: "123xyz",
  contact: {
    phone: "123-456-7890",
    email: "xyz@example.com"
  },
  access: {
    level: 5,
    group: "dev"
  }
}
```

Embedded sub-document

Embedded sub-document

Hemos decidido mantener las referencias, en vez de hacer uso de documentos embebidos, ya que nos permite más flexibilidad a la hora de realizar modificaciones a los documentos. Además, este esquema se asimila más a una base de datos relacional, la cual estamos más familiarizados.

Sin embargo, el uso de este tipo de consultas nos acarrea algunos problemas, como puede ser el uso de "\$lookup" a la hora de realizar ciertas agregaciones, el cual dependiendo de la cantidad de información a buscar puede tener un impacto negativo en el rendimiento.

Definición JSON Schema. Como base para generar ambos esquemas, nos hemos basado en el tutorial de mongodb:

- [JSON Schema Examples Tutorial | MongoDB](#)

Inspections:

```
{
  $jsonSchema: {
    "type": "object",
    "properties": {
      "_id": {
        "type": "object",
        "properties": {
          "$oid": {
            "type": "string"
          }
        },
        "required": [
          "$oid"
        ]
      },
      "id": {
        "type": "string"
      },
      "certificate_number": {
        "type": "integer"
      },
      "business_name": {
        "type": "string"
      },
      "date": {
        "type": "string"
      },
      "result": {
        "type": "string"
      },
      "sector": {
        "type": "string"
      }
    }
  }
}
```

```
"address": {
  "type": "object",
  "properties": {
    "city": {
      "type": "string"
    },
    "zip": {
      "type": "string"
    },
    "street": {
      "type": "string"
    },
    "number": {
      "type": "string"
    }
  },
  "required": [
    "city",
    "zip",
    "street",
    "number"
  ],
  "restaurant_id": {
    "type": "string"
  }
},
"required": [
  "_id",
  "id",
  "certificate_number",
  "business_name",
  "date",
  "result",

```

```

"sector",
"address",
"restaurant_id"
    ],
    "additionalProperties": false
  }
}

```

Restaurants:

```

{
  $jsonSchema: {
    "type": "object",
    "properties": {
      "_id": {
        "type": "object",
        "properties": {
          "$oid": {
            "type": "string"
          }
        },
        "required": [
          "$oid"
        ]
      },
      "URL": {
        "type": "string"
      },
      "address": {
        "type": "string"
      },
      "address line 2": {
        "type": "string"
      },
      "name": {
        "type": "string"
      }
    },
    "outcode": {
      "type": "string"
    },
    "postcode": {
      "type": "string"
    },
    "rating": {
      "type": "integer"
    },
    "type_of_food": {
      "type": "string"
    },
    "required": [
      "_id",
      "URL",
      "address",
      "address line 2",
      "name",
      "outcode",
      "postcode",
      "rating",
      "type_of_food"
    ]
  }
}

```

Para ambos casos, hemos generado un “schema” que requiera que todos los campos estén especificados y con el tipo correcto a la hora de añadir un nuevo restaurante o inspección, especialmente en la colección de inspections, que debemos añadir el “restaurant_id” por tal de mantener las referencias.

Gracias al uso de los “schemas,” evitemos que se añadan registros con valores vacíos o de tipos erróneos.

2. Implementación de consultas en MongoDB

- Buscar todos los restaurantes de un tipo de comida específico (ej. "Chinese").
- Listar las inspecciones con violaciones, ordenadas por fecha.
- Encontrar restaurantes con una calificación superior a 4.

Consultas: Todos los restaurantes de tipo de comida "Chinese."

```
db.restaurants.find({"type_of_food": "Chinese"}) [1]
```

Consultas: Listar las inspecciones con violaciones, ordenadas por fecha.

```
db.inspections.find({ "result": "Violation Issued"
}).sort({"date": -1}) [2]
```

Consultas: Encontrar restaurantes con una calificación superior a 4.

```
db.restaurants.find({"rating": {$gt: 4}}) [3]
```

3. Uso de agregaciones

- Agrupar restaurantes por tipo de comida y calcular la calificación promedio.
- Contar el número de inspecciones por resultado y mostrar los porcentajes.
- Unir restaurantes con sus inspecciones utilizando \$lookup.

Consultas: Agrupar restaurantes por tipo de comida y calcular la calificación promedio.

```
db.restaurants.aggregate([
  {
    $group: {
      _id: "$type_of_food",
      promedio_calificacion: {$avg: "$rating"}
    }
  },
  {
    $project: {
      type_of_food: "$_id",
      promedio_calificacion: {$round:
["$promedio_calificacion", 2]},
      _id: 0
    }
  },
  {
    $sort: {promedio_calificacion: -1}
  }
]); [4]
```

Consultas: Contar el número de inspecciones por resultado y mostrar los porcentajes.

```
db.inspections.aggregate([
  {
    $group: {
      _id: "$result",
      count: {$sum: 1}
    }
  },
  {
    $group: {
      _id: null,
      total: {$sum: "$count"},
      results: {$addToSet: { result: "$_id", count: "$count"
}}}
  }
},
{
  $unwind: "$results"
},
{
  $project: {
    _id: 0,
    result: "$results.result",
    count: "$results.count",
    percentage: {
      $round: [{$multiply: [ {$divide:
["$results.count", "$total"]}, 100 ]}, 2]
    }
  }
},
{
  $sort: {count: -1}
}
]); [5]
```


Consultas: Unir restaurantes con sus inspecciones utilizando \$lookup

```
db.restaurants.aggregate([
  {
    $lookup: {
      from: "inspections",
      let: { restaurant_id_str: { $toString: "$_id" } },
      pipeline: [
        {
          $match: {
            $expr: { $eq: ["$restaurant_id",
"$${restaurant_id_str}"] }
          }
        }
      ],
      as: "inspection_history"
    }
  },
  {
    $project: {
      _id: 1,
      name: 1,
      type_of_food: 1,
      rating: 1,
      inspection_history: 1
    }
  }
]); [6]
```

TAREAS AVANZADAS (Hasta 6 puntos)

1. Optimización del rendimiento

- Identificar las posibles consultas más frecuentes.
- Implementar índices adecuados para esas consultas.
- Comparar el rendimiento antes y después de crear los índices utilizando explain().

Haciendo un análisis del tipo de información que disponemos hemos identificado posibles consultas más frecuentes que se pueden ocasionar.

Comenzando por la colección de restaurantes, los campos que pueden ser los más utilizados son el tipo de comida y la puntuación de los restaurantes. Pensando como un usuario, a la hora de realizar búsquedas estos dos campos son los más importantes, debido a que los usuarios buscarán el tipo de comida que prefieran y por norma general buscarán restaurantes con buenas valoraciones, por tal de tener una experiencia satisfactoria.

Por lo tanto, en esta colección deberemos indexar los campos "type_of_food" y "rating."

Siguiendo con la colección de inspecciones, una posibilidad es tener consultas que filtren los datos por un rango de fechas, como "inspecciones realizadas entre el 1 de enero y el 31 de diciembre de 2022".

Estas consultas probablemente utilicen campos relacionados con la fecha. Este tipo de consultas suelen ser muy frecuentes en aplicaciones que manejan datos históricos, como el sistema de gestión de inspecciones que se está analizando. Los usuarios a menudo necesitan acceder a los registros de inspecciones realizadas en un período de tiempo determinado, ya sea para fines de análisis, reportes o seguimiento.

Además, otro de los campos que sería útil indexar es el resultado de la inspección, esto nos permite buscar de manera más ágil el tipo de resultado de la inspección, por ejemplo si un restaurante tuvo una inspección negativa, es interesante saber en la próxima inspección el resultado de las inspecciones anteriores para comprobar si el dueño del restaurante ha solucionado los problemas que llevaron a una inspección negativa.

Debido a nuestro diseño de la base de datos, que mantenemos las referencias en la colección de inspecciones, deberemos también indexar el campo "restaurant_id" por tal de mejorar el rendimiento a la hora de usar lookups.

A continuación se muestran unas consultas sin los campos a indexar con el tiempo de ejecución, después indexaremos los campos mencionados previamente, y volveremos a ejecutar las mismas consultas por tal de realizar una comparativa de rendimiento.

1.1 Consultas sin Indexar:

Consulta restaurants type_of_food y rating:

```
db.restaurants.find({
  type_of_food: "Curry",
  rating: { $gt: 4 }
}).explain("executionStats"); [7]
```

Nos devuelve un COLSCAN, y un execution time de 1ms. El output completo puede visualizarse en la sección de outputs.

```
"indexFilterSet": false,
  "queryHash": "CB1D88FF",
  "planCacheShapeHash": "CB1D88FF",
  "planCacheKey": "4F073AD3",
  "optimizationTimeMillis": 0,
  "maxIndexedOrSolutionsReached": false,
  "maxIndexedAndSolutionsReached": false,
  "maxScansToExplodeReached": false,
  "prunedSimilarIndexes": false,
  "winningPlan": {
    "isCached": false,
    "stage": "COLLSCAN",
```

Consulta inspecciones con result y fechas:

```
db.inspections.find({
  result: "No Violation Issued",
  date: { $gte: "Jul 01 2023", $lte: "Jul 31 2023" }
}).explain("executionStats"); [8]
```

Nos devuelve un COLSCAN, y un execution time de 4ms. El output completo puede visualizarse en la sección de outputs.

```
"executionStats": {
  "executionSuccess": true,
  "nReturned": 109,
  "executionTimeMillis": 4,
  "totalKeysExamined": 0,
  "totalDocsExamined": 6370,
  "executionStages": {
    "isCached": false,
    "stage": "COLLSCAN",
```

Consulta union restaurants e inspections con lookup:

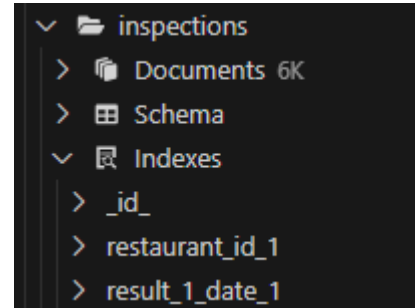
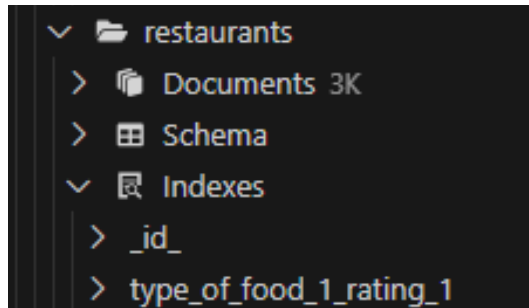
```
db.restaurants.aggregate([
  {
    $lookup: {
      from: "inspections",
      localField: "_id",
      foreignField: "restaurant_id",
      as: "inspection_history"
    }
  },
  {
    $match: {
      "inspection_history.result": "Violation Issued"
    }
  }
]).explain("executionStats");
```

En este caso, como el uso de \$lookup tiene un impacto en el rendimiento de la consulta, vemos que ha tardado un total de 9179 ms en ejecutarse.

```
"executionStats": {
  "executionSuccess": true,
  "nReturned": 2548,
  "executionTimeMillis": 9179,
  "totalKeysExamined": 0,
  "totalDocsExamined": 16233308,
  "executionStages": {
    "stage": "mkobj",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 9144,
    "opens": 1,
    "closes": 1,
    "saveState": 592,
    "restoreState": 592,
    "isEOF": 1,
    "objSlot": 16,
    "rootSlot": 1,
    "fieldBehavior": "drop",
    "fields": [],
    "projectFields": [
      "inspection_history"
    ],
  },
}
```

1.2 Implementación de index:

```
db.restaurants.createIndex({ type_of_food: 1, rating: 1 });  
db.inspections.createIndex({ result: 1, date: 1 });  
db.inspections.createIndex({ restaurant_id: 1 });
```



1.3 Consultas Indexadas:

Las consultas realizadas serán las mismas que las mencionadas previamente, por lo que sólo indicaremos el output del execution time. El output completo se puede visualizar en la sección de output.

Consulta restaurants type_of_food y rating [10]:

```
"executionStats": {  
  "executionSuccess": true,  
  "nReturned": 841,  
  "executionTimeMillis": 1,  
  "totalKeysExamined": 841,  
  "totalDocsExamined": 841,  
  "executionStages": {  
    ...  
  }  
},  
"inputStage": {  
  "stage": "IXSCAN",  
  "nReturned": 841,  
  "executionTimeMillisEstimate": 0,  
  "works": 842,
```

Observamos que ahora se ha realizado una búsqueda con IXSCAN, indicando que se han utilizado el índice. En este caso, la consulta nos ha tardado el mismo tiempo en realizarse que sin índices.

Consulta inspecciones con result y fechas [11]:

```
"executionStats": {  
  "executionSuccess": true,  
  "nReturned": 109,  
  "executionTimeMillis": 0,  
  "totalKeysExamined": 109,  
  "totalDocsExamined": 109,  
  "executionStages": {  
    "isCached": false,  
    "stage": "FETCH",  
    "nReturned": 109,  
    "executionTimeMillisEstimate": 0,  
    "works": 110,  
    "advanced": 109,  
    "needTime": 0,  
    "needYield": 0,  
    "saveState": 0,  
    "restoreState": 0,  
    "isEOF": 1,  
    "docsExamined": 109,  
    "alreadyHasObj": 0,  
    "inputStage": {  
      "stage": "IXSCAN",
```

En este caso, sí podemos ver una mejora, tal como antes había tardado 4ms ahora la búsqueda podemos decir que ha sido “instantánea” con un tiempo de ejecución de 0ms.

Consulta union restaurants e inspections con lookup [12]:

```
"executionStats": {
  "executionSuccess": true,
  "nReturned": 2548,
  "executionTimeMillis": 35,
  "totalKeysExamined": 0,
  "totalDocsExamined": 2548,
  "executionStages": {
    "stage": "mkobj",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 17,
    "opens": 1,
    "closes": 1,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "objSlot": 28,
    "rootSlot": 1,
    "fieldBehavior": "drop",
    "fields": [],
    "projectFields": [
      "inspection_history"
    ]
  }
}
```

En este caso podemos observar mejor como la implementación de índices mejora el rendimiento, mientras que antes tardaba una consulta más de 9000ms con la implementación de "restaurant_id" ahora solo tarda 35ms, lo que supone un gran incremento en el rendimiento, haciendo así más viable el uso de referencias.

2. Estrategias de escalabilidad

- Proponer una estrategia de sharding adecuada para este dataset.
- Diseñar un esquema de replicación para alta disponibilidad.
- Analizar posibles cuellos de botella y soluciones.

Para realizar este apartado se ha tenido que realizar de forma teórica porque en mongodb atlas disponemos de un cluster en el plan M0 que es el nivel gratuito. Es una buena opción para empezar con la introducción en mongodb pero el plan contiene limitaciones en las configuraciones de escalabilidad y alta disponibilidad.

<https://www.mongodb.com/docs/atlas/reference/free-shared-limitations/>

Estrategia de sharding

Una de las principales estrategias de escalabilidad para MongoDB es el sharding, que permite distribuir los datos entre múltiples servidores (shards) de manera transparente para

la aplicación. Para implementar una estrategia de sharding adecuada para este dataset, se deberá realizar un análisis exhaustivo de las características del mismo, como el volumen de datos, los patrones de acceso y las necesidades de escalabilidad. A continuación se detallan las siguientes estrategias que podríamos seguir:

1- Sharding basado en el campo restaurant_id

Para la colección de "inspections", se propone utilizar el campo restaurant_id como clave de sharding. Esta estrategia permitirá que todas las inspecciones de un mismo restaurante se almacenen en el mismo shard.

Al agrupar las inspecciones por restaurante en un mismo shard, se facilitarán las consultas y agregaciones orientadas a un restaurante en particular. Esto resulta adecuado cuando los patrones de uso de los datos suelen estar enfocados en operaciones y análisis a nivel de restaurante, como informes o estadísticas por establecimiento.

2- Sharding basado en el campo date

Para la colección de "inspections", se propone utilizar un campo de fecha/hora, como date, como clave de sharding. Esta estrategia resulta útil cuando las consultas y operaciones suelen estar orientadas a un rango de fechas, como informes mensuales o anuales de inspecciones.

Al utilizar un campo de fecha como clave de sharding, se logrará una distribución más uniforme de los datos en los shards, ya que las inspecciones se distribuirán en función de la fecha. Esto es beneficioso cuando se requieren consultas y análisis que abarcan un rango temporal, como informes periódicos sobre las inspecciones realizadas. Al tener las inspecciones distribuidas en los shards en función de la fecha, se facilitará la ejecución eficiente de consultas que requieran analizar las inspecciones dentro de un intervalo temporal específico.

Para hacer la activación del sharding se deberá de ejecutar el siguiente comando:

```
sh.enableSharding("nombre_de_la_base_de_datos")
```

Después se deberá de crear la shared collection con el campo que hemos definido y explicado anteriormente.

```
sh.shardCollection("nombre_de_la_base_de_datos", { restaurant_id: 1 })
```

```
sh.shardCollection("nombre_de_la_base_de_datos", { date: 1 })
```

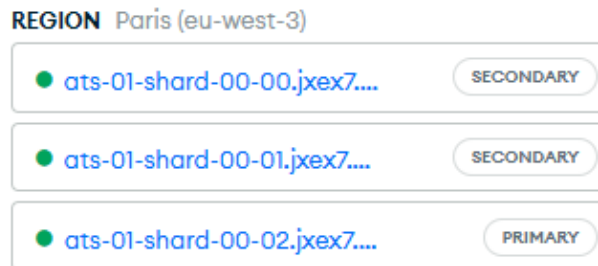
Replicación

Es crucial implementar un esquema de replicación que garantice la alta disponibilidad de los datos. Se ha diseñado una arquitectura de replicación que utiliza un conjunto de réplicas (Replica Set) compuesto por 3 nodos. Esta configuración ofrece importantes beneficios.

En primer lugar, permite la tolerancia a fallos, ya que el Replica Set puede seguir operativo incluso si uno de los nodos falla. Esto es crucial para mantener la disponibilidad del servicio. Además, con 3 nodos, el Replica Set puede elegir democráticamente un nuevo nodo primario en caso de que el actual falle, asegurando una transición suave y automática. Esto garantiza la continuidad de las operaciones.

Otro aspecto clave es el quórum de escritura. Con 3 nodos, se requiere que las escrituras sean confirmadas por al menos 2 nodos, lo que proporciona una mayor consistencia de los datos almacenados.

La configuración por defecto que tienes con el plan M0 y no puedes modificar la puedes visualizar en la siguiente pantalla que mostramos a continuación:



Para hacer la creación del conjunto de réplicas (Replica Set) podríamos realizar el siguiente código para su configuración:

```
rs.initiate({
  _id: "nombre_del_replica_set",
  members: [
    { _id: 0, host: "host1:27017" },
    { _id: 1, host: "host2:27017" },
    { _id: 2, host: "host3:27017" }
  ]
})
```

Cuello de botella y posibles soluciones

Se pueden ocasionar diferentes puntos de cuellos de botella y se deben analizar las siguientes métricas de rendimiento para controlar posibles problemas.

- 1- Consumo de CPU: : Monitorear el porcentaje de utilización de CPU en cada uno de los nodos para detectar si alguno está saturado.
- 2- Consumo de RAM: Revisar el uso de memoria RAM y detectar si algún nodo está llegando a sus límites.
- 3- Latencias en escritura/lectura de disco: Analizar la carga de lectura y escritura en disco para identificar si hay un nodo con mayor actividad.
- 4- Consumo de red: Monitorear el ancho de banda utilizado entre los nodos

Una vez se han analizado los datos comentamos en el punto anterior y identificados los posibles cuellos de botella, se pueden implementar algunas de las siguientes soluciones:

- Aumentar los recursos de hardware
- Aumentar con el número de nodos al clúster
- Optimizar consultas para mejorar el acceso en disco
- Configuración de cachés para minimizar las lecturas en el disco

Outputs

[1] Comida de tipo "Chinese"

```
[
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b35"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-113-fish-bar-wallasey/menu",
    "address": "113 Poulton Road",
    "address line 2": "Merseyside",
    "name": "113 Fish Bar",
    "outcode": "CH44",
    "postcode": "9DE",
    "rating": 5.5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b5d"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-4inlove-pe30/menu",
    "address": "10-12 Railway Road",
    "address line 2": "Norfolk",
    "name": "4 in Love",
    "outcode": "PE30",
    "postcode": "1NE",
    "rating": 5.5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b64"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-5-star-chinese-archway/menu",
    "address": "362 Hornsey Road",
    "address line 2": "London",
    "name": "5 Star Chinese",
    "outcode": "N19",
    "postcode": "4HD",
    "rating": 5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
```

```

    "$oid": "55f14312c7447c3da7051b75"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-7andone-bt53/menu",
  "address": "3 Charles Street",
  "address line 2": "County Antrim",
  "name": "7 & One",
  "outcode": "BT53",
  "postcode": "6DX",
  "rating": 5.5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b82"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-888tripleeight-bh9/menu",
  "address": "888 Wimborne Road",
  "address line 2": "Bournemouth",
  "name": "888 Triple Eight",
  "outcode": "BH9",
  "postcode": "2DR",
  "rating": 5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b85"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-9thkitchen-b26/menu",
  "address": "397 Sheldon Heath Road",
  "address line 2": "Sheldon",
  "name": "9th Kitchen",
  "outcode": "B26",
  "postcode": "2UB",
  "rating": 5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b30"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-1-2-3-chinese-rowlands-gill/menu",
  "address": "Unit 4 Spencer House",
  "address line 2": "Swalwell",
  "name": "1 2 3 Chinese",
  "outcode": "NE16",
  "postcode": "3DS",

```

```

    "rating": 4.5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b32"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-100menu-wn1/menu",
    "address": "50 Wallgate",
    "address line 2": "Wigan",
    "name": "100 Menu",
    "outcode": "WN1",
    "postcode": "1JU",
    "rating": 5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b55"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-328chineseonline-cf81/menu",
    "address": "2-4 Commercial Street",
    "address line 2": "Aberbargoed",
    "name": "328 Chinese Online",
    "outcode": "CF81",
    "postcode": "9BW",
    "rating": 5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b58"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-333Chinese-M11/menu",
    "address": "1276 Ashton Old Road",
    "address line 2": "Manchester",
    "name": "333 Chinese Takeaway1",
    "outcode": "M11",
    "postcode": "1JJ",
    "rating": 4.5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b59"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-350wok-ws10/menu",

```

```

"address": "136A Crankhall Lane",
"address line 2": "Wednesbury",
"name": "350 Wok",
"outcode": "WS10",
"postcode": "0ED",
"rating": 5.5,
"type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b5a"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-350wok-ws10/menu",
  "address": "136A Crankhall Lane",
  "address line 2": "Wednesbury",
  "name": "350 Wok",
  "outcode": "WS10",
  "postcode": "0ED",
  "rating": 5.5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b5b"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-350wok-ws10/menu",
  "address": "136A Crankhall Lane",
  "address line 2": "Wednesbury",
  "name": "350 Wok",
  "outcode": "WS10",
  "postcode": "0ED",
  "rating": 5.5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b63"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-5-star-chinese-archway/menu",
  "address": "362 Hornsey Road",
  "address line 2": "London",
  "name": "5 Star Chinese",
  "outcode": "N19",
  "postcode": "4HD",
  "rating": 5,
  "type_of_food": "Chinese"
},

```

```
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b72"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-57-orient-takeawayPR8/menu",
  "address": "81 Upper Aughton Road",
  "address line 2": "Southport",
  "name": "57 Orient Takeaway",
  "outcode": "PR8",
  "postcode": "5ND",
  "rating": 4.5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b7f"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-88-chinese-millbrook/menu",
  "address": "44 London Road",
  "address line 2": "Southampton",
  "name": "88 Chinese",
  "outcode": "SO15",
  "postcode": "2AH",
  "rating": 5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b81"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-888chinese-so23/menu",
  "address": "77 North Walls",
  "address line 2": "Winchester",
  "name": "888 Chinese Takeaway",
  "outcode": "SO23",
  "postcode": "8DA",
  "rating": 5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b86"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-A-G-Kitchen-M40/menu",
  "address": "267 Nuthurst Road",
  "address line 2": "Manchester",
  "name": "A & G Kitchen",

```

```

    "outcode": "M40",
    "postcode": "3PU",
    "rating": 5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051c5c"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-af88-stockport/menu",
    "address": "63 Middle Hill Gate",
    "address line 2": "Stockport",
    "name": "AF88",
    "outcode": "SK1",
    "postcode": "3EH",
    "rating": 4.5,
    "type_of_food": "Chinese"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051c79"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-aftertastesw6/menu",
    "address": "97 Newington Butts",
    "address line 2": "Elephant & Castle",
    "name": "After Taste",
    "outcode": "SE1",
    "postcode": "6SF",
    "rating": 5,
    "type_of_food": "Chinese"
  }
]

```

[2] Listar las inspecciones con violaciones, ordenadas por fecha.

```

{
  "_id": {
    "$oid": "56d61033a378eccde8a88a83"
  },
  "id": "26253-2015-ENFO",
  "certificate_number": 3021207,
  "business_name": "AMEERA'S",
  "date": "Sep 30 2024",
  "result": "Violation Issued",
  "sector": "Grocery-Retail - 808",

```

```

"address": {
  "city": "TYNE AND WEAR",
  "zip": "7AF",
  "street": "HYLTON ROAD",
  "number": "61"
},
"restaurant_id": "55f14312c7447c3da7051e48"
},
{
  "_id": {
    "$oid": "56d61035a378eccde8a96f6c"
  },
  "id": "5525-2016-ENFO",
  "certificate_number": 80021214,
  "business_name": "BEST KEBAB",
  "date": "Sep 30 2024",
  "result": "Violation Issued",
  "sector": "Cigarette Retail Dealer - 127",
  "address": {
    "city": "SUFFOLK",
    "zip": "8NP",
    "street": "BEECHES ROAD",
    "number": "37"
  },
  "restaurant_id": "55f14313c7447c3da70523e2"
},
{
  "_id": {
    "$oid": "56d61035a378eccde8a96024"
  },
  "id": "2284-2016-ENFO",
  "certificate_number": 5392337,
  "business_name": "BIDDICK HALL SPICE",
  "date": "Sep 30 2022",
  "result": "Violation Issued",
  "sector": "Grocery-Retail - 808",
  "address": {
    "city": "TYNE AND WEAR",
    "zip": "9NG",
    "street": "FIELDING COURT",
    "number": "14"
  },
  "restaurant_id": "55f14313c7447c3da7052454"
},
{
  "_id": {
    "$oid": "56d61034a378eccde8a9062b"
  },

```



```

},
"id": "10073-2015-ENFO",
"certificate_number": 9294197,
"business_name": "BIG MO'S",
"date": "Sep 29 2024",
"result": "Violation Issued",
"sector": "Secondhand Dealer [General] - 006",
"address": {
  "city": "ENFIELD",
  "zip": "4EZ",
  "street": "HIGH STREET",
  "number": "238"
},
"restaurant_id": "55f14313c7447c3da7052492"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a84b6e"
  },
  "id": "1391-2016-ENFO",
  "certificate_number": 50065856,
  "business_name": "AKASH TANDOORI",
  "date": "Sep 29 2024",
  "result": "Violation Issued",
  "sector": "Cigarette Retail Dealer - 127",
  "address": {
    "city": "ELGIN",
    "zip": "1JZ",
    "street": "SOUTH STREET",
    "number": "21"
  },
  "restaurant_id": "55f14312c7447c3da7051cc5"
},
{
  "_id": {
    "$oid": "56d61034a378eccde8a8ee03"
  },
  "id": "62211-2015-ENFO",
  "certificate_number": 9285859,
  "business_name": "ALBERT'S FISH & CHIPS",
  "date": "Sep 28 2024",
  "result": "Violation Issued",
  "sector": "Cigarette Retail Dealer - 127",
  "address": {
    "city": "SHEFFIELD",
    "zip": "1GN",
    "street": "CITY ROAD",

```

```

    "number": "746"
  },
  "restaurant_id": "55f14312c7447c3da7051d46"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a8751a"
  },
  "id": "25552-2015-ENFO",
  "certificate_number": 3019849,
  "business_name": "ASHMAAN TANDOORI",
  "date": "Sep 28 2023",
  "result": "Violation Issued",
  "sector": "Grocery-Retail - 808",
  "address": {
    "city": "WEST LOTHIAN",
    "zip": "7QJ",
    "street": "MILL ROAD",
    "number": "25"
  },
  "restaurant_id": "55f14312c7447c3da7051fea"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a86c06"
  },
  "id": "8394-2015-ENFO",
  "certificate_number": 3019095,
  "business_name": "042 RESTAURANT & BAR",
  "date": "Sep 27 2023",
  "result": "Violation Issued",
  "sector": "Grocery-Retail - 808",
  "address": {
    "city": "LONDON",
    "zip": "1HR",
    "street": "HIGH ROAD LEYTONSTONE",
    "number": "885"
  },
  "restaurant_id": "55f14312c7447c3da7051b2e"
},
{
  "_id": {
    "$oid": "56d61035a378eccde8a94b56"
  },
  "id": "59038-2015-ENFO",
  "certificate_number": 9304996,
  "business_name": "ATLANTIS FISH BAR & CAFE",

```

```

    "date": "Sep 27 2023",
    "result": "Violation Issued",
    "sector": "Funeral Homes - 888",
    "address": {
      "city": "BIRMINGHAM",
      "zip": "8ND",
      "street": "HIGH STREET",
      "number": "289"
    },
    "restaurant_id": "55f14312c7447c3da7052057"
  },
  {
    "_id": {
      "$oid": "56d61033a378eccde8a84fae"
    },
    "id": "3640-2015-ENFO",
    "certificate_number": 9301950,
    "business_name": "BIG FISH",
    "date": "Sep 27 2022",
    "result": "Violation Issued",
    "sector": "Drug Store Retail - 810",
    "address": {
      "city": "RUMNEY",
      "zip": "3EA",
      "street": "WENTLOOG ROAD",
      "number": "100"
    },
    "restaurant_id": "55f14313c7447c3da705247b"
  },
  {
    "_id": {
      "$oid": "56d61034a378eccde8a8c25f"
    },
    "id": "10291-2015-ENFO",
    "certificate_number": 5348420,
    "business_name": "BAMBOO SUSHI",
    "date": "Sep 27 2022",
    "result": "Violation Issued",
    "sector": "Cigarette Retail Dealer - 127",
    "address": {
      "city": "BANES",
      "zip": "1UQ",
      "street": "ST. JAMES'S PARADE",
      "number": "43"
    },
    "restaurant_id": "55f14313c7447c3da705219f"
  },

```

```
{
  "_id": {
    "$oid": "56d61033a378eccde8a8640a"
  },
  "id": "47627-2015-ENFO",
  "certificate_number": 3046170,
  "business_name": "BALTI VILLAGE",
  "date": "Sep 26 2024",
  "result": "Violation Issued",
  "sector": "Grocery-Retail - 808",
  "address": {
    "city": "HARPENDEN",
    "zip": "5AJ",
    "street": "LOWER LUTON ROAD",
    "number": "93"
  },
  "restaurant_id": "55f14313c7447c3da705217e"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a85d67"
  },
  "id": "68865-2015-ENFO",
  "certificate_number": 9317460,
  "business_name": "BEI JING",
  "date": "Sep 26 2024",
  "result": "Violation Issued",
  "sector": "Sidewalk Cafe - 013",
  "address": {
    "city": "STRETFORD",
    "zip": "8GR",
    "street": "GREAT STONE ROAD",
    "number": "79"
  },
  "restaurant_id": "55f14313c7447c3da70522b5"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a89e2b"
  },
  "id": "178-2015-UNIT",
  "certificate_number": 10005964,
  "business_name": "BIG FAT GOURMET",
  "date": "Sep 26 2024",
  "result": "Violation Issued",
  "sector": "Tow Truck Company - 124",
  "address": {
```

```

    "city": "LONDON",
    "zip": "3JD",
    "street": "WANDSWORTH ROAD",
    "number": "585"
  },
  "restaurant_id": "55f14313c7447c3da7052479"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a86a69"
  },
  "id": "727-2015-ENFO",
  "certificate_number": 70025272,
  "business_name": "BENGAL REGENCY",
  "date": "Sep 26 2024",
  "result": "Violation Issued",
  "sector": "Cigarette Retail Dealer - 127",
  "address": {
    "city": "DERBYSHIRE",
    "zip": "6JF",
    "street": "MAIN STREET",
    "number": "18"
  },
  "restaurant_id": "55f14313c7447c3da7052373"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a85cc0"
  },
  "id": "48936-2015-ENFO",
  "certificate_number": 50062629,
  "business_name": "BELLA DONNER PIZZA & GRILL",
  "date": "Sep 26 2023",
  "result": "Violation Issued",
  "sector": "Cigarette Retail Dealer - 127",
  "address": {
    "city": "SKELMERSDALE",
    "zip": "9PP",
    "street": "UNIT 3 PIKELAW PLACE",
    "number": ""
  },
  "restaurant_id": "55f14313c7447c3da70522ff"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a84439"
  },

```

```

    "id": "52640-2015-ENFO",
    "certificate_number": 9304757,
    "business_name": "57 ORIENT TAKEAWAY",
    "date": "Sep 26 2022",
    "result": "Violation Issued",
    "sector": "Cigarette Retail Dealer - 127",
    "address": {
      "city": "SOUTHPORT",
      "zip": "5ND",
      "street": "UPPER AUGHTON ROAD",
      "number": "81"
    },
    "restaurant_id": "55f14312c7447c3da7051b72"
  },
  {
    "_id": {
      "$oid": "56d61035a378eccde8a9400d"
    },
    "id": "60078-2015-ENFO",
    "certificate_number": 9310210,
    "business_name": "BAAN SIAM THAI CUISINE",
    "date": "Sep 25 2022",
    "result": "Violation Issued",
    "sector": "Cigarette Retail Dealer - 127",
    "address": {
      "city": "CUMBRIA",
      "zip": "1PQ",
      "street": "DALTON ROAD",
      "number": "231"
    },
    "restaurant_id": "55f14313c7447c3da70520ba"
  },
  {
    "_id": {
      "$oid": "56d61034a378eccde8a9003f"
    },
    "id": "33766-2015-ENFO",
    "certificate_number": 9293006,
    "business_name": "BASILO PIZZA",
    "date": "Sep 24 2024",
    "result": "Violation Issued",
    "sector": "Cigarette Retail Dealer - 127",
    "address": {
      "city": "WALTON ON THAMES",
      "zip": "2QS",
      "street": "CHURCH STREET",
      "number": "38"
    }
  }

```

```

    },
    "restaurant_id": "55f14313c7447c3da705221e"
  },
  {
    "_id": {
      "$oid": "56d61033a378eccde8a83dd1"
    },
    "id": "6562-2015-CMPL",
    "certificate_number": 5381300,
    "business_name": "BELLA PIZZA",
    "date": "Sep 24 2022",
    "result": "Violation Issued",
    "sector": "Home Improvement Contractor - 100",
    "address": {
      "city": "HULL",
      "zip": "0RE",
      "street": "PRESTONGATE",
      "number": "28"
    },
    "restaurant_id": "55f14313c7447c3da7052310"
  }
]

```

[3] Encontrar restaurantes con una calificación superior a 4.

```

[
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b2a"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-indiancom-ch4/menu",
    "address": "9 Broughton Hall Road",
    "address line 2": "Broughton",
    "name": "@Indian.com",
    "outcode": "CH4",
    "postcode": "0QR",
    "rating": 6,
    "type_of_food": "Curry"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b35"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-113-fish-bar-wallasey/menu",
    "address": "113 Poulton Road",
    "address line 2": "Merseyside",

```

```

"name": "113 Fish Bar",
"outcode": "CH44",
"postcode": "9DE",
"rating": 5.5,
"type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b3c"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-1stclasspizza-bs7/menu",
  "address": "388 Filton Avenue",
  "address line 2": "Bristol",
  "name": "1st Class Pizza",
  "outcode": "BS7",
  "postcode": "0BE",
  "rating": 4.5,
  "type_of_food": "Pizza"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b3d"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-1stclasspizza-bs7/menu",
  "address": "388 Filton Avenue",
  "address line 2": "Bristol",
  "name": "1st Class Pizza",
  "outcode": "BS7",
  "postcode": "0BE",
  "rating": 4.5,
  "type_of_food": "Pizza"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b3e"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-1stclasspizza-ng18/menu",
  "address": "70 Carter Lane",
  "address line 2": "Mansfield",
  "name": "1st Class Pizza",
  "outcode": "NG18",
  "postcode": "3DH",
  "rating": 4.5,
  "type_of_food": "Pizza"
},
{
  "_id": {

```



```

    "$oid": "55f14312c7447c3da7051b42"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-2-spice/menu",
  "address": "2 Station Approach",
  "address line 2": "Upminster",
  "name": "2 Spice",
  "outcode": "RM14",
  "postcode": "2TH",
  "rating": 5.5,
  "type_of_food": "Curry"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b45"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-23rdstreetpizza-m14/menu",
  "address": "317 Wilmslow Road",
  "address line 2": "Fallowfield",
  "name": "23rd Street Pizza",
  "outcode": "M14",
  "postcode": "6NW",
  "rating": 5,
  "type_of_food": "American"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b46"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-241-pizza/menu",
  "address": "107 King Richards Road",
  "address line 2": "Leicester",
  "name": "241 Pizza",
  "outcode": "LE3",
  "postcode": "5QG",
  "rating": 5,
  "type_of_food": "Pizza"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b4a"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-241piza-m34/menu",
  "address": "88 Manchester Road",
  "address line 2": "Manchester",
  "name": "2-4-1 Pizza",
  "outcode": "M34",
  "postcode": "3PR",

```

```

    "rating": 5,
    "type_of_food": "Pizza"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b4d"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-3chefs-fishbar-de21/menu",
    "address": "47 Roosevelt Avenue",
    "address line 2": "Derby",
    "name": "3 Chef Fish Bar",
    "outcode": "DE21",
    "postcode": "6JR",
    "rating": 4.5,
    "type_of_food": "Fish & Chips"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b4e"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-3chefs-de23/menu",
    "address": "558 Burton Road",
    "address line 2": "Littleover",
    "name": "3 Chefs",
    "outcode": "DE23",
    "postcode": "6FP",
    "rating": 4.5,
    "type_of_food": "American"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b5c"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-4in1-Express-CH41/menu",
    "address": "122 Duke Street",
    "address line 2": "Birkenhead",
    "name": "4 in 1 Express",
    "outcode": "CH41",
    "postcode": "8BT",
    "rating": 5,
    "type_of_food": "Pizza"
  },
  {
    "_id": {
      "$oid": "55f14312c7447c3da7051b5d"
    },
    "URL": "http://www.just-eat.co.uk/restaurants-4inlove-pe30/menu",

```

```

"address": "10-12 Railway Road",
"address line 2": "Norfolk",
"name": "4 in Love",
"outcode": "PE30",
"postcode": "1NE",
"rating": 5.5,
"type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b5f"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-4seasonspizzaandgrill-dy3/menu",
  "address": "12-14 High Street",
  "address line 2": "Dudley",
  "name": "4 Seasons Pizza & Grill",
  "outcode": "DY3",
  "postcode": "1RW",
  "rating": 5,
  "type_of_food": "Pizza"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b61"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-5in1-pr3/menu",
  "address": "23 Berry Lane",
  "address line 2": "Longridge",
  "name": "5 In 1 Takeaway",
  "outcode": "PR3",
  "postcode": "3JA",
  "rating": 5,
  "type_of_food": "Curry"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b64"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-5-star-chinese-archway/menu",
  "address": "362 Hornsey Road",
  "address line 2": "London",
  "name": "5 Star Chinese",
  "outcode": "N19",
  "postcode": "4HD",
  "rating": 5,
  "type_of_food": "Chinese"
},

```

```
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b71"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-555pizzaandkebab-ct12/menu",
  "address": "15 Northwood Road",
  "address line 2": "Kent",
  "name": "555 Pizza & Kebab",
  "outcode": "CT12",
  "postcode": "6RR",
  "rating": 4.5,
  "type_of_food": "Pizza"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b75"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-7andone-bt53/menu",
  "address": "3 Charles Street",
  "address line 2": "County Antrim",
  "name": "7 & One",
  "outcode": "BT53",
  "postcode": "6DX",
  "rating": 5.5,
  "type_of_food": "Chinese"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b76"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-7oaksgrilltn13/menu",
  "address": "127 St Johns Hill",
  "address line 2": "Sevenoaks",
  "name": "7 Oaks Grill",
  "outcode": "TN13",
  "postcode": "3PE",
  "rating": 4.5,
  "type_of_food": "English"
},
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b79"
  },
  "URL": "http://www.just-eat.co.uk/restaurants-7spices-g32/menu",
  "address": "975 Carntyne Road",
  "address line 2": "Glasgow",
  "name": "7 Spices Takeaway",

```

```
"outcode": "G32",  
"postcode": "6LY",  
"rating": 4.5,  
"type_of_food": "Curry"  
}  
]
```

[4] Agrupar restaurantes por tipo de comida y calcular la calificación promedio.

```
{  
  "_id": "Pasta",  
  "type_of_food": "Pasta",  
  "promedio_calificacion": 6  
},  
{  
  "_id": "Punjabi",  
  "type_of_food": "Punjabi",  
  "promedio_calificacion": 6  
},  
{  
  "_id": "Cakes",  
  "type_of_food": "Cakes",  
  "promedio_calificacion": 5.5  
},  
{  
  "_id": "Pick n Mix",  
  "type_of_food": "Pick n Mix",  
  "promedio_calificacion": 5.5  
},  
{  
  "_id": "Bagels",  
  "type_of_food": "Bagels",  
  "promedio_calificacion": 5.5  
},  
{  
  "_id": "Ice Cream",  
  "type_of_food": "Ice Cream",  
  "promedio_calificacion": 5.5  
},  
{  
  "_id": "Bangladeshi",  
  "type_of_food": "Bangladeshi",  
  "promedio_calificacion": 5.31  
},  
{
```

```
"_id": "Persian",
"type_of_food": "Persian",
"promedio_calificacion": 5.25
},
{
  "_id": "Mediterranean",
  "type_of_food": "Mediterranean",
  "promedio_calificacion": 5.17
},
{
  "_id": "Polish",
  "type_of_food": "Polish",
  "promedio_calificacion": 5.17
},
{
  "_id": "Greek",
  "type_of_food": "Greek",
  "promedio_calificacion": 5.14
},
{
  "_id": "Sushi",
  "type_of_food": "Sushi",
  "promedio_calificacion": 5.12
},
{
  "_id": "Curry",
  "type_of_food": "Curry",
  "promedio_calificacion": 5.04
},
{
  "_id": "Fish & Chips",
  "type_of_food": "Fish & Chips",
  "promedio_calificacion": 5.04
},
{
  "_id": "Ethiopian",
  "type_of_food": "Ethiopian",
  "promedio_calificacion": 5
},
{
  "_id": "Portuguese",
  "type_of_food": "Portuguese",
  "promedio_calificacion": 5
},
{
  "_id": "Breakfast",
  "type_of_food": "Breakfast",
```

```
"promedio_calificacion": 5
},
{
  "_id": "Korean",
  "type_of_food": "Korean",
  "promedio_calificacion": 5
},
{
  "_id": "Azerbaijan",
  "type_of_food": "Azerbaijan",
  "promedio_calificacion": 5
},
{
  "_id": "Afghan",
  "type_of_food": "Afghan",
  "promedio_calificacion": 4.97
}
]
```

[5] Contar el número de inspecciones por resultado y mostrar los porcentajes.

```
[
  {
    "result": "Violation Issued",
    "count": 1291,
    "percentage": 20.27
  },
  {
    "result": "Warning Issued",
    "count": 1280,
    "percentage": 20.09
  },
  {
    "result": "Fail",
    "count": 1280,
    "percentage": 20.09
  },
  {
    "result": "No Violation Issued",
    "count": 1260,
    "percentage": 19.78
  },
  {
    "result": "Pass",
    "count": 1259,
    "percentage": 19.76
  }
]
```

```
}
]
```

[6] Unir restaurantes con sus inspecciones utilizando \$lookup

```
[
{
  "_id": {
    "$oid": "55f14312c7447c3da7051b2a"
  },
  "name": "@Indian.com",
  "rating": 6,
  "type_of_food": "Curry",
  "inspection_history": [
    {
      "_id": {
        "$oid": "56d61035a378eccde8a9449f"
      },
      "id": "68109-2015-ENFO",
      "certificate_number": 3049371,
      "business_name": "@INDIAN.COM",
      "date": "Jul 03 2023",
      "result": "No Violation Issued",
      "sector": "Grocery-Retail - 808",
      "address": {
        "city": "BROUGHTON",
        "zip": "0QR",
        "street": "BROUGHTON HALL ROAD",
        "number": "9"
      },
    },
    "restaurant_id": "55f14312c7447c3da7051b2a"
  ],
  {
    "_id": {
      "$oid": "56d61035a378eccde8a94262"
    },
    "id": "8658-2015-ENFO",
    "certificate_number": 5351342,
    "business_name": "@INDIAN.COM",
    "date": "Apr 15 2022",
    "result": "Violation Issued",
    "sector": "Grocery-Retail - 808",
    "address": {
      "city": "BROUGHTON",
      "zip": "0QR",
      "street": "BROUGHTON HALL ROAD",
```



```

    "number": "9"
  },
  "restaurant_id": "55f14312c7447c3da7051b2a"
},
{
  "_id": {
    "$oid": "56d61033a378eccde8a84b30"
  },
  "id": "70586-2015-ENFO",
  "certificate_number": 3044697,
  "business_name": "@INDIAN.COM",
  "date": "Aug 13 2023",
  "result": "Fail",
  "sector": "Supermarket - 819",
  "address": {
    "city": "BROUGHTON",
    "zip": "0QR",
    "street": "BROUGHTON HALL ROAD",
    "number": "9"
  },
  "restaurant_id": "55f14312c7447c3da7051b2a"
}
]
},

```

[7] Restaurants sin index:

```

{
  "explainVersion": "1",
  "queryPlanner": {
    "namespace": "PLAB01.restaurants",
    "parsedQuery": {
      "$and": [
        {
          "type_of_food": {
            "$eq": "Curry"
          }
        },
        {
          "rating": {
            "$gt": 4
          }
        }
      ]
    }
  },
  "indexFilterSet": false,

```

```

"queryHash": "CB1D88FF",
"planCacheShapeHash": "CB1D88FF",
"planCacheKey": "4F073AD3",
"optimizationTimeMillis": 0,
"maxIndexedOrSolutionsReached": false,
"maxIndexedAndSolutionsReached": false,
"maxScansToExplodeReached": false,
"prunedSimilarIndexes": false,
"winningPlan": {
  "isCached": false,
  "stage": "COLLSCAN",
  "filter": {
    "$and": [
      {
        "type_of_food": {
          "$eq": "Curry"
        }
      },
      {
        "rating": {
          "$gt": 4
        }
      }
    ]
  },
  "direction": "forward"
},
"rejectedPlans": [],
},
"executionStats": {
  "executionSuccess": true,
  "nReturned": 841,
  "executionTimeMillis": 1,
  "totalKeysExamined": 0,
  "totalDocsExamined": 2548,
  "executionStages": {
    "isCached": false,
    "stage": "COLLSCAN",
    "filter": {
      "$and": [
        {
          "type_of_food": {
            "$eq": "Curry"
          }
        },
        {
          "rating": {

```

```

        "$gt": 4
      }
    }
  ]
},
"nReturned": 841,
"executionTimeMillisEstimate": 0,
"works": 2549,
"advanced": 841,
"needTime": 1707,
"needYield": 0,
"saveState": 0,
"restoreState": 0,
"isEOF": 1,
"direction": "forward",
"docsExamined": 2548
}
},
"queryShapeHash":
"7C27AC0A27F13B08B42C04B08F04752F10C2A27975547F25D1047C01B2ACE19B",
"command": {
  "find": "restaurants",
  "filter": {
    "type_of_food": "Curry",
    "rating": {
      "$gt": 4
    }
  }
},
"$db": "PLAB01"
},
"serverInfo": {
  "host": "atsuab2025-shard-00-01.icbhz.mongodb.net",
  "port": 27017,
  "version": "8.0.5",
  "gitVersion": "cb9e2e5e552ee39dea1e39d7859336456d0c9820"
},
"serverParameters": {
  "internalQueryFacetBufferSizeBytes": 104857600,
  "internalQueryFacetMaxOutputDocSizeBytes": 104857600,
  "internalLookupStageIntermediateDocumentMaxSizeBytes": 16793600,
  "internalDocumentSourceGroupMaxMemoryBytes": 104857600,
  "internalQueryMaxBlockingSortMemoryUsageBytes": 33554432,
  "internalQueryProhibitBlockingMergeOnMongoS": 0,
  "internalQueryMaxAddToSetBytes": 104857600,
  "internalDocumentSourceSetWindowFieldsMaxMemoryBytes": 104857600,
  "internalQueryFrameworkControl": "trySbeRestricted",
  "internalQueryPlannerIgnoreIndexWithCollationForRegex": 1
}

```

```

},
"ok": 1,
"$clusterTime": {
  "clusterTime": {
    "$timestamp": {
      "t": 1742376753,
      "i": 16
    }
  },
  "signature": {
    "hash": {
      "$binary": {
        "base64": "9XfEtTjpucDEzMBXmo6AUj6r+xo=",
        "subType": "00"
      }
    },
    "keyId": 7441318053373018000
  },
  "operationTime": {
    "$timestamp": {
      "t": 1742376753,
      "i": 16
    }
  }
}

```

[8] consulta inspections sin index:

```

{
  "explainVersion": "1",
  "queryPlanner": {
    "namespace": "PLAB01.inspections",
    "parsedQuery": {
      "$and": [
        {
          "result": {
            "$eq": "No Violation Issued"
          }
        },
        {
          "date": {
            "$lte": "Jul 31 2023"
          }
        }
      ]
    }
  }
}

```

```

    "date": {
      "$gte": "Jul 01 2023"
    }
  }
]
},
"indexFilterSet": false,
"queryHash": "27C505CE",
"planCacheShapeHash": "27C505CE",
"planCacheKey": "A0C71C4E",
"optimizationTimeMillis": 0,
"maxIndexedOrSolutionsReached": false,
"maxIndexedAndSolutionsReached": false,
"maxScansToExplodeReached": false,
"prunedSimilarIndexes": false,
"winningPlan": {
  "isCached": false,
  "stage": "COLLSCAN",
  "filter": {
    "$and": [
      {
        "result": {
          "$eq": "No Violation Issued"
        }
      },
      {
        "date": {
          "$lte": "Jul 31 2023"
        }
      },
      {
        "date": {
          "$gte": "Jul 01 2023"
        }
      }
    ]
  }
},
"direction": "forward"
},
"rejectedPlans": [],
},
"executionStats": {
  "executionSuccess": true,
  "nReturned": 109,
  "executionTimeMillis": 4,
  "totalKeysExamined": 0,
  "totalDocsExamined": 6370,

```

```

"executionStages": {
  "isCached": false,
  "stage": "COLLSCAN",
  "filter": {
    "$and": [
      {
        "result": {
          "$eq": "No Violation Issued"
        }
      },
      {
        "date": {
          "$lte": "Jul 31 2023"
        }
      },
      {
        "date": {
          "$gte": "Jul 01 2023"
        }
      }
    ]
  },
  "nReturned": 109,
  "executionTimeMillisEstimate": 0,
  "works": 6371,
  "advanced": 109,
  "needTime": 6261,
  "needYield": 0,
  "saveState": 0,
  "restoreState": 0,
  "isEOF": 1,
  "direction": "forward",
  "docsExamined": 6370
},
"queryShapeHash":
"70846EB72D4A855CB8B0C9F88F60E386AC4C2129BABAE59EA216613B10C213D9",
"command": {
  "find": "inspections",
  "filter": {
    "result": "No Violation Issued",
    "date": {
      "$gte": "Jul 01 2023",
      "$lte": "Jul 31 2023"
    }
  }
},
"$db": "PLAB01"

```

```

},
"serverInfo": {
  "host": "atsuab2025-shard-00-01.icbhz.mongodb.net",
  "port": 27017,
  "version": "8.0.5",
  "gitVersion": "cb9e2e5e552ee39dea1e39d7859336456d0c9820"
},
"serverParameters": {
  "internalQueryFacetBufferSizeBytes": 104857600,
  "internalQueryFacetMaxOutputDocSizeBytes": 104857600,
  "internalLookupStageIntermediateDocumentMaxSizeBytes": 16793600,
  "internalDocumentSourceGroupMaxMemoryBytes": 104857600,
  "internalQueryMaxBlockingSortMemoryUsageBytes": 33554432,
  "internalQueryProhibitBlockingMergeOnMongoS": 0,
  "internalQueryMaxAddToSetBytes": 104857600,
  "internalDocumentSourceSetWindowFieldsMaxMemoryBytes": 104857600,
  "internalQueryFrameworkControl": "trySbeRestricted",
  "internalQueryPlannerIgnoreIndexWithCollationForRegex": 1
},
"ok": 1,
"$clusterTime": {
  "clusterTime": {
    "$timestamp": {
      "t": 1742377651,
      "i": 3
    }
  },
  "signature": {
    "hash": {
      "$binary": {
        "base64": "F1n/03FYWunUAhOQvM6NnFQuZEY=",
        "subType": "00"
      }
    }
  },
  "keyId": 7441318053373018000
},
"operationTime": {
  "$timestamp": {
    "t": 1742377651,
    "i": 3
  }
}
}

```

[9] consulta \$lookup sin index:

```
{
  "explainVersion": "2",
  "stages": [
    {
      "$cursor": {
        "queryPlanner": {
          "namespace": "PLAB01.restaurants",
          "parsedQuery": {},
          "indexFilterSet": false,
          "queryHash": "B3956D95",
          "planCacheShapeHash": "B3956D95",
          "planCacheKey": "E66D7B71",
          "optimizationTimeMillis": 0,
          "maxIndexedOrSolutionsReached": false,
          "maxIndexedAndSolutionsReached": false,
          "maxScansToExplodeReached": false,
          "prunedSimilarIndexes": false,
          "winningPlan": {
            "isCached": false,
            "queryPlan": {
              "stage": "EQ_LOOKUP",
              "planNodeId": 2,
              "foreignCollection": "67b44757579f354f64a81043_PLAB01.inspections",
              "localField": "_id",
              "foreignField": "restaurant_id",
              "asField": "inspection_history",
              "strategy": "NestedLoopJoin",
              "scanDirection": "forward",
              "inputStage": {
                "stage": "COLLSCAN",
                "planNodeId": 1,
                "filter": {},
                "direction": "forward"
              }
            }
          },
          "slotBasedPlan": {
            "slots": "$$RESULT=s16 env: { }",
            "stages": "[2] mkobj s16 s1 [] drop [inspection_history = s15] true false \n[2] nlj inner
[s1] [s10] \n left \n [2] nlj inner [s1] [s1] \n left \n [1] scan s1 s2 none
none none none none none lowPriority [] @\"2b2f022d-07ee-45b7-8cce-9954c277ddd2\"
true false \n right \n [2] project [s10 = \n if isArrayEmpty(s8) \n
then [null] \n else s8 \n ] \n [2] group [] [s8 = addToSet(s6)]
spillSlots[s9] mergingExprs[aggSetUnion(s9)] \n [2] unwind s6 s7 s5 true \n
[2] project [s5 = getField(s1, \"_id\")] \n [2] limit 11 \n [2] coscan \n right
\n [2] limit 11 \n [2] union [s15] \n branch0 [s13] \n [2] project [s13 =
```



```

getElement(s11, 0)] \n          [2] group [] [s11 = addToArrayCapped(s3, 16793600)]
spillSlots[s12] mergingExprs[aggConcatArraysCapped(s12, 16793600)] \n          [2] filter
{traverseF((getField(s3, \"restaurant_id\") ?: null), lambda(l1.0) { isMember(l1.0, s10) }, true)}
\n          [2] scan s3 s4 none none none none none none []
@\"f80dcd8f-f8c6-4943-a918-a44a4b0fac4f\" true false \n          branch1 [s14] \n
[2] project [s14 = []] \n          [2] limit 1|| \n          [2] coscan \n\"
    }
    },
    \"rejectedPlans\": []
},
\"executionStats\": {
    \"executionSuccess\": true,
    \"nReturned\": 2548,
    \"executionTimeMillis\": 9179,
    \"totalKeysExamined\": 0,
    \"totalDocsExamined\": 16233308,
    \"executionStages\": {
        \"stage\": \"mkobj\",
        \"planNodeId\": 2,
        \"nReturned\": 2548,
        \"executionTimeMillisEstimate\": 9144,
        \"opens\": 1,
        \"closes\": 1,
        \"saveState\": 592,
        \"restoreState\": 592,
        \"isEOF\": 1,
        \"objSlot\": 16,
        \"rootSlot\": 1,
        \"fieldBehavior\": \"drop\",
        \"fields\": [],
        \"projectFields\": [
            \"inspection_history\"
        ],
        \"projectSlots\": [
            15
        ],
        \"forceNewObject\": true,
        \"returnOldObject\": false,
        \"inputStage\": {
            \"stage\": \"nlj\",
            \"planNodeId\": 2,
            \"nReturned\": 2548,
            \"executionTimeMillisEstimate\": 9133,
            \"opens\": 1,
            \"closes\": 1,
            \"saveState\": 592,
            \"restoreState\": 592,

```

```
"isEOF": 1,
"totalDocsExamined": 16233308,
"totalKeysExamined": 0,
"collectionScans": 2549,
"collectionSeeks": 0,
"indexScans": 0,
"indexSeeks": 0,
"indexesUsed": [],
"innerOpens": 2548,
"innerCloses": 1,
"outerProjects": [
  1
],
"outerCorrelated": [
  10
],
"outerStage": {
  "stage": "nlj",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 20,
  "opens": 1,
  "closes": 1,
  "saveState": 592,
  "restoreState": 592,
  "isEOF": 1,
  "totalDocsExamined": 2548,
  "totalKeysExamined": 0,
  "collectionScans": 1,
  "collectionSeeks": 0,
  "indexScans": 0,
  "indexSeeks": 0,
  "indexesUsed": [],
  "innerOpens": 2548,
  "innerCloses": 1,
  "outerProjects": [
    1
  ],
  "outerCorrelated": [
    1
  ],
  "outerStage": {
    "stage": "scan",
    "planNodeId": 1,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 2,
    "opens": 1,
```

```

    "closes": 1,
    "saveState": 592,
    "restoreState": 592,
    "isEOF": 1,
    "numReads": 2548,
    "recordSlot": 1,
    "recordIdSlot": 2,
    "scanFieldNames": [],
    "scanFieldSlots": []
  },
  "innerStage": {
    "stage": "project",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 18,
    "opens": 2548,
    "closes": 1,
    "saveState": 592,
    "restoreState": 592,
    "isEOF": 1,
    "projections": {
      "10": "\n  if isArrayEmpty(s8) \n    then [null] \n    else s8 \n"
    },
    "inputStage": {
      "stage": "group",
      "planNodeId": 2,
      "nReturned": 2548,
      "executionTimeMillisEstimate": 18,
      "opens": 2548,
      "closes": 1,
      "saveState": 592,
      "restoreState": 592,
      "isEOF": 1,
      "groupBySlots": [],
      "expressions": {
        "8": "addToSet(s6) ",
        "initExprs": {
          "8": null
        }
      },
      "mergingExprs": {
        "9": "aggSetUnion(s9) "
      },
      "usedDisk": false,
      "spills": 0,
      "spilledBytes": 0,
      "spilledRecords": 0,

```

```
"spilledDataStorageSize": 0,
"inputStage": {
  "stage": "unwind",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 0,
  "opens": 2548,
  "closes": 2548,
  "saveState": 592,
  "restoreState": 592,
  "isEOF": 1,
  "inputSlot": 5,
  "outSlot": 6,
  "outIndexSlot": 7,
  "preserveNullAndEmptyArrays": 1,
  "inputStage": {
    "stage": "project",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 592,
    "restoreState": 592,
    "isEOF": 1,
    "projections": {
      "5": "getField(s1, \"_id\") "
    },
  },
  "inputStage": {
    "stage": "limit",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 592,
    "restoreState": 592,
    "isEOF": 1,
    "limit": 1,
    "inputStage": {
      "stage": "coscan",
      "planNodeId": 2,
      "nReturned": 2548,
      "executionTimeMillisEstimate": 0,
      "opens": 2548,
      "closes": 2548,
      "saveState": 592,
```

```

        "restoreState": 592,
        "isEOF": 0
    }
}
}
}
}
},
"innerStage": {
    "stage": "limit",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 9113,
    "opens": 2548,
    "closes": 1,
    "saveState": 592,
    "restoreState": 592,
    "isEOF": 1,
    "limit": 1,
    "inputStage": {
        "stage": "union",
        "planNodeId": 2,
        "nReturned": 2548,
        "executionTimeMillisEstimate": 9112,
        "opens": 2548,
        "closes": 1,
        "saveState": 592,
        "restoreState": 592,
        "isEOF": 0,
        "inputSlots": [
            13,
            14
        ],
        "outputSlots": [
            15
        ],
        "inputStages": [
            {
                "stage": "project",
                "planNodeId": 2,
                "nReturned": 0,
                "executionTimeMillisEstimate": 9108,
                "opens": 2548,
                "closes": 2548,
                "saveState": 592,
                "restoreState": 592,
            }
        ]
    }
}

```

```

    "isEOF": 1,
    "projections": {
      "13": "getElement(s11, 0) "
    },
    "inputStage": {
      "stage": "group",
      "planNodeId": 2,
      "nReturned": 0,
      "executionTimeMillisEstimate": 9108,
      "opens": 2548,
      "closes": 2548,
      "saveState": 592,
      "restoreState": 592,
      "isEOF": 1,
      "groupBySlots": [],
      "expressions": {
        "11": "addToArrayCapped(s3, 16793600) ",
        "initExprs": {
          "11": null
        }
      }
    },
    "mergingExprs": {
      "12": "aggConcatArraysCapped(s12, 16793600) "
    },
    "usedDisk": false,
    "spills": 0,
    "spilledBytes": 0,
    "spilledRecords": 0,
    "spilledDataStorageSize": 0,
    "inputStage": {
      "stage": "filter",
      "planNodeId": 2,
      "nReturned": 0,
      "executionTimeMillisEstimate": 9105,
      "opens": 2548,
      "closes": 2548,
      "saveState": 592,
      "restoreState": 592,
      "isEOF": 1,
      "numTested": 16230760,
      "filter": "traverseF((getField(s3, \"restaurant_id\") ? : null), lambda(l1.0) {
isMember(l1.0, s10) }, true) ",
      "inputStage": {
        "stage": "scan",
        "planNodeId": 2,
        "nReturned": 16230760,
        "executionTimeMillisEstimate": 4255,

```

```

        "opens": 2548,
        "closes": 2548,
        "saveState": 592,
        "restoreState": 592,
        "isEOF": 1,
        "numReads": 16230760,
        "recordSlot": 3,
        "recordIdSlot": 4,
        "scanFieldNames": [],
        "scanFieldSlots": []
    }
}
},
{
    "stage": "project",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 3,
    "opens": 2548,
    "closes": 2548,
    "saveState": 592,
    "restoreState": 592,
    "isEOF": 0,
    "projections": {
        "14": "[ ]"
    },
    "inputStage": {
        "stage": "limit",
        "planNodeId": 2,
        "nReturned": 2548,
        "executionTimeMillisEstimate": 3,
        "opens": 2548,
        "closes": 2548,
        "saveState": 592,
        "restoreState": 592,
        "isEOF": 0,
        "limit": 1,
        "inputStage": {
            "stage": "coscan",
            "planNodeId": 2,
            "nReturned": 2548,
            "executionTimeMillisEstimate": 0,
            "opens": 2548,
            "closes": 2548,
            "saveState": 592,
            "restoreState": 592,

```

```

        "isEOF": 0
      }
    }
  ]
}
}
}
}
},
"nReturned": 2548,
"executionTimeMillisEstimate": 9173
},
{
  "$match": {
    "inspection_history.result": {
      "$eq": "Violation Issued"
    }
  },
  "nReturned": 0,
  "executionTimeMillisEstimate": 9179
}
],
"queryShapeHash":
"76F7192AE0A71E599FB1DAAEA8B4C6DA4F38E47A79210E509EE487E42C9921CD",
"serverInfo": {
  "host": "atsuab2025-shard-00-01.icbhz.mongodb.net",
  "port": 27017,
  "version": "8.0.5",
  "gitVersion": "cb9e2e5e552ee39dea1e39d7859336456d0c9820"
},
"serverParameters": {
  "internalQueryFacetBufferSizeBytes": 104857600,
  "internalQueryFacetMaxOutputDocSizeBytes": 104857600,
  "internalLookupStageIntermediateDocumentMaxSizeBytes": 16793600,
  "internalDocumentSourceGroupMaxMemoryBytes": 104857600,
  "internalQueryMaxBlockingSortMemoryUsageBytes": 33554432,
  "internalQueryProhibitBlockingMergeOnMongoS": 0,
  "internalQueryMaxAddToSetBytes": 104857600,
  "internalDocumentSourceSetWindowFieldsMaxMemoryBytes": 104857600,
  "internalQueryFrameworkControl": "trySbeRestricted",
  "internalQueryPlannerIgnoreIndexWithCollationForRegex": 1
},
"command": {
  "aggregate": "restaurants",
  "pipeline": [

```



```

{
  "$lookup": {
    "from": "inspections",
    "localField": "_id",
    "foreignField": "restaurant_id",
    "as": "inspection_history"
  }
},
{
  "$match": {
    "inspection_history.result": "Violation Issued"
  }
}
],
"cursor": {},
"$db": "PLAB01"
},
"ok": 1,
"$clusterTime": {
  "clusterTime": {
    "$timestamp": {
      "t": 1742377256,
      "i": 4
    }
  },
  "signature": {
    "hash": {
      "$binary": {
        "base64": "9PFAt+qLIT5cli3DnUJzjHrtitl=",
        "subType": "00"
      }
    }
  },
  "keyId": 7441318053373018000
}
},
"operationTime": {
  "$timestamp": {
    "t": 1742377256,
    "i": 4
  }
}
}
}

```

[10] consulta restaurants con index:

```
{
  "explainVersion": "1",
  "queryPlanner": {
    "namespace": "PLAB01.restaurants",
    "parsedQuery": {
      "$and": [
        {
          "type_of_food": {
            "$eq": "Curry"
          }
        },
        {
          "rating": {
            "$gt": 4
          }
        }
      ]
    },
    "indexFilterSet": false,
    "queryHash": "CB1D88FF",
    "planCacheShapeHash": "CB1D88FF",
    "planCacheKey": "3854CC62",
    "optimizationTimeMillis": 0,
    "maxIndexedOrSolutionsReached": false,
    "maxIndexedAndSolutionsReached": false,
    "maxScansToExplodeReached": false,
    "prunedSimilarIndexes": false,
    "winningPlan": {
      "isCached": false,
      "stage": "FETCH",
      "inputStage": {
        "stage": "IXSCAN",
        "keyPattern": {
          "type_of_food": 1,
          "rating": 1
        },
        "indexName": "type_of_food_1_rating_1",
        "isMultiKey": false,
        "multiKeyPaths": {
          "type_of_food": [],
          "rating": []
        },
        "isUnique": false,
        "isSparse": false,
        "isPartial": false,
```

```

        "indexVersion": 2,
        "direction": "forward",
        "indexBounds": {
            "type_of_food": [
                ["Curry\", \"Curry\"]
            ],
            "rating": [
                "(4, inf.0]"
            ]
        }
    },
    "rejectedPlans": [],
},
"executionStats": {
    "executionSuccess": true,
    "nReturned": 841,
    "executionTimeMillis": 1,
    "totalKeysExamined": 841,
    "totalDocsExamined": 841,
    "executionStages": {
        "isCached": false,
        "stage": "FETCH",
        "nReturned": 841,
        "executionTimeMillisEstimate": 1,
        "works": 842,
        "advanced": 841,
        "needTime": 0,
        "needYield": 0,
        "saveState": 0,
        "restoreState": 0,
        "isEOF": 1,
        "docsExamined": 841,
        "alreadyHasObj": 0,
        "inputStage": {
            "stage": "IXSCAN",
            "nReturned": 841,
            "executionTimeMillisEstimate": 0,
            "works": 842,
            "advanced": 841,
            "needTime": 0,
            "needYield": 0,
            "saveState": 0,
            "restoreState": 0,
            "isEOF": 1,
            "keyPattern": {
                "type_of_food": 1,
            }
        }
    }
}

```

```

    "rating": 1
  },
  "indexName": "type_of_food_1_rating_1",
  "isMultiKey": false,
  "multiKeyPaths": {
    "type_of_food": [],
    "rating": []
  },
  "isUnique": false,
  "isSparse": false,
  "isPartial": false,
  "indexVersion": 2,
  "direction": "forward",
  "indexBounds": {
    "type_of_food": [
      ["Curry", "Curry"]
    ],
    "rating": [
      "(4, inf.0]"
    ]
  },
  "keysExamined": 841,
  "seeks": 1,
  "dupsTested": 0,
  "dupsDropped": 0
}
},
"queryShapeHash":
"7C27AC0A27F13B08B42C04B08F04752F10C2A27975547F25D1047C01B2ACE19B",
"command": {
  "find": "restaurants",
  "filter": {
    "type_of_food": "Curry",
    "rating": {
      "$gt": 4
    }
  },
},
"$db": "PLAB01"
},
"serverInfo": {
  "host": "atsuab2025-shard-00-01.icbhz.mongodb.net",
  "port": 27017,
  "version": "8.0.5",
  "gitVersion": "cb9e2e5e552ee39dea1e39d7859336456d0c9820"
},
"serverParameters": {

```

```
"internalQueryFacetBufferSizeBytes": 104857600,
"internalQueryFacetMaxOutputDocSizeBytes": 104857600,
"internalLookupStageIntermediateDocumentMaxSizeBytes": 16793600,
"internalDocumentSourceGroupMaxMemoryBytes": 104857600,
"internalQueryMaxBlockingSortMemoryUsageBytes": 33554432,
"internalQueryProhibitBlockingMergeOnMongoS": 0,
"internalQueryMaxAddToSetBytes": 104857600,
"internalDocumentSourceSetWindowFieldsMaxMemoryBytes": 104857600,
"internalQueryFrameworkControl": "trySbeRestricted",
"internalQueryPlannerIgnoreIndexWithCollationForRegex": 1
},
"ok": 1,
"$clusterTime": {
  "clusterTime": {
    "$timestamp": {
      "t": 1742378659,
      "i": 5
    }
  },
  "signature": {
    "hash": {
      "$binary": {
        "base64": "EVYkswAJ5BD2/clrZtQqzLP1Uow=",
        "subType": "00"
      }
    },
    "keyId": 7441318053373018000
  },
  "operationTime": {
    "$timestamp": {
      "t": 1742378659,
      "i": 5
    }
  }
}
}
```

[11] consulta inspections con index:

```
{
  "explainVersion": "1",
  "queryPlanner": {
    "namespace": "PLAB01.inspections",
    "parsedQuery": {
      "$and": [
        {
          "result": {
```

```

    "$eq": "No Violation Issued"
  }
},
{
  "date": {
    "$lte": "Jul 31 2023"
  }
},
{
  "date": {
    "$gte": "Jul 01 2023"
  }
}
]
},
"indexFilterSet": false,
"queryHash": "27C505CE",
"planCacheShapeHash": "27C505CE",
"planCacheKey": "1D68F4A5",
"optimizationTimeMillis": 0,
"maxIndexedOrSolutionsReached": false,
"maxIndexedAndSolutionsReached": false,
"maxScansToExplodeReached": false,
"prunedSimilarIndexes": false,
"winningPlan": {
  "isCached": false,
  "stage": "FETCH",
  "inputStage": {
    "stage": "IXSCAN",
    "keyPattern": {
      "result": 1,
      "date": 1
    }
  },
  "indexName": "result_1_date_1",
  "isMultiKey": false,
  "multiKeyPaths": {
    "result": [],
    "date": []
  },
  "isUnique": false,
  "isSparse": false,
  "isPartial": false,
  "indexVersion": 2,
  "direction": "forward",
  "indexBounds": {
    "result": [
      "[\"No Violation Issued\", \"No Violation Issued\"]"
    ]
  }
}

```

```

    ],
    "date": [
      ["Jul 01 2023\","Jul 31 2023\"]
    ]
  }
}
},
"rejectedPlans": []
},
"executionStats": {
  "executionSuccess": true,
  "nReturned": 109,
  "executionTimeMillis": 0,
  "totalKeysExamined": 109,
  "totalDocsExamined": 109,
  "executionStages": {
    "isCached": false,
    "stage": "FETCH",
    "nReturned": 109,
    "executionTimeMillisEstimate": 0,
    "works": 110,
    "advanced": 109,
    "needTime": 0,
    "needYield": 0,
    "saveState": 0,
    "restoreState": 0,
    "isEOF": 1,
    "docsExamined": 109,
    "alreadyHasObj": 0,
    "inputStage": {
      "stage": "IXSCAN",
      "nReturned": 109,
      "executionTimeMillisEstimate": 0,
      "works": 110,
      "advanced": 109,
      "needTime": 0,
      "needYield": 0,
      "saveState": 0,
      "restoreState": 0,
      "isEOF": 1,
      "keyPattern": {
        "result": 1,
        "date": 1
      },
    },
  },
  "indexName": "result_1_date_1",
  "isMultiKey": false,
  "multiKeyPaths": {

```

```

    "result": [],
    "date": []
  },
  "isUnique": false,
  "isSparse": false,
  "isPartial": false,
  "indexVersion": 2,
  "direction": "forward",
  "indexBounds": {
    "result": [
      ["No Violation Issued", "No Violation Issued"]
    ],
    "date": [
      ["Jul 01 2023", "Jul 31 2023"]
    ]
  },
  "keysExamined": 109,
  "seeks": 1,
  "dupsTested": 0,
  "dupsDropped": 0
}
},
"queryShapeHash":
"70846EB72D4A855CB8B0C9F88F60E386AC4C2129BABAE59EA216613B10C213D9",
"command": {
  "find": "inspections",
  "filter": {
    "result": "No Violation Issued",
    "date": {
      "$gte": "Jul 01 2023",
      "$lte": "Jul 31 2023"
    }
  }
},
"$db": "PLAB01"
},
"serverInfo": {
  "host": "atsuab2025-shard-00-01.icbhz.mongodb.net",
  "port": 27017,
  "version": "8.0.5",
  "gitVersion": "cb9e2e5e552ee39dea1e39d7859336456d0c9820"
},
"serverParameters": {
  "internalQueryFacetBufferSizeBytes": 104857600,
  "internalQueryFacetMaxOutputDocSizeBytes": 104857600,
  "internalLookupStageIntermediateDocumentMaxSizeBytes": 16793600,
  "internalDocumentSourceGroupMaxMemoryBytes": 104857600,

```



```

"internalQueryMaxBlockingSortMemoryUsageBytes": 33554432,
"internalQueryProhibitBlockingMergeOnMongoS": 0,
"internalQueryMaxAddToSetBytes": 104857600,
"internalDocumentSourceSetWindowFieldsMaxMemoryBytes": 104857600,
"internalQueryFrameworkControl": "trySbeRestricted",
"internalQueryPlannerIgnoreIndexWithCollationForRegex": 1
},
"ok": 1,
"$clusterTime": {
  "clusterTime": {
    "$timestamp": {
      "t": 1742379369,
      "i": 12
    }
  },
  "signature": {
    "hash": {
      "$binary": {
        "base64": "6ZcAkysvDS0cGrCzNJoceOkOBqg=",
        "subType": "00"
      }
    },
    "keyId": 7441318053373018000
  },
  "operationTime": {
    "$timestamp": {
      "t": 1742379369,
      "i": 12
    }
  }
}
}

```

[12] consulta \$lookup con index:

```

{
  "explainVersion": "2",
  "stages": [
    {
      "$cursor": {
        "queryPlanner": {
          "namespace": "PLAB01.restaurants",
          "parsedQuery": {},
          "indexFilterSet": false,
          "queryHash": "B3956D95",
          "planCacheShapeHash": "B3956D95",
          "planCacheKey": "E66D7B71",

```

```

"optimizationTimeMillis": 0,
"maxIndexedOrSolutionsReached": false,
"maxIndexedAndSolutionsReached": false,
"maxScansToExplodeReached": false,
"prunedSimilarIndexes": false,
"winningPlan": {
  "isCached": false,
  "queryPlan": {
    "stage": "EQ_LOOKUP",
    "planNodeId": 2,
    "foreignCollection": "67b44757579f354f64a81043_PLAB01.inspections",
    "localField": "_id",
    "foreignField": "restaurant_id",
    "asField": "inspection_history",
    "strategy": "IndexedLoopJoin",
    "indexName": "restaurant_id_1",
    "indexKeyPattern": {
      "restaurant_id": 1
    },
    "scanDirection": "forward",
    "inputStage": {
      "stage": "COLLSCAN",
      "planNodeId": 1,
      "filter": {},
      "direction": "forward"
    }
  },
  "slotBasedPlan": {
    "slots": "$$RESULT=s28 env: { }",
    "stages": "[2] mkobj s28 s1 [] drop [inspection_history = s27] true false \n[2] nlj inner
[s1] [s8] \n left \n [2] nlj inner [s1] [s1] \n left \n [1] scan s1 s2 none
none none none none none lowPriority [] @"2b2f022d-07ee-45b7-8cce-9954c277ddd2\"
true false \n right \n [2] project [s8 = \n if isArrayEmpty(s6) \n
then [null] \n else s6 \n ] \n [2] group [] [s6 = addToSet(s4)]
spillSlots[s7] mergingExprs[aggSetUnion(s7)] \n [2] unwind s4 s5 s3 true \n
[2] project [s3 = getField(s1, \"_id\")] \n [2] limit 1ll \n [2] coscan \n right
\n [2] limit 1ll \n [2] union [s27] \n branch0 [s25] \n [2] project [s25 =
getElement(s23, 0)] \n [2] group [] [s23 = addToArrayCapped(s21, 16793600)]
spillSlots[s24] mergingExprs[aggConcatArraysCapped(s24, 16793600)] \n [2] filter
{traverseF((getField(s21, \"restaurant_id\") ?: null), lambda(l1.0) { isMember(l1.0, s8) }, true)}
\n [2] nlj inner [] [s17, s19, s20, s18, s16] \n left \n [2] nlj
inner [s16] [s14, s15] \n left \n [2] nlj inner [] [s9] \n
left \n [2] unwind s9 s10 s8 true \n [2] limit
1ll \n [2] coscan \n right \n
[2] project [s14 = ks(1ll, 0, s13, 1ll), s15 = ks(1ll, 0, s13, 2ll), s16 = {\"restaurant_id\" : 1}] \n
[2] union [s13] \n branch0 [s11] \n
[2] filter {(isArray(s9) && !(isMember(s11, s8)))} \n [2] project

```

```

[s11 = (getElement(s9, 0) ?: undefined)] \n
[2] limit 1ll \n
[2] coscan \n
branch1 [s12] \n
[2]
project [s12 = s9] \n
[2] limit 1ll \n
[2] coscan \n
right \n
[2] ixseek s14 s15 s18 s17 s19 s20
[] @\f80dcd8f-f8c6-4943-a918-a44a4b0fac4f" @\restaurant_id_1\" true \n
right
\n
[2] limit 1ll \n
[2] seek s17 s21 s22 s19 s20 s18 s16 none
none [] @\f80dcd8f-f8c6-4943-a918-a44a4b0fac4f" true false \n
branch1 [s26] \n
[2] project [s26 = []] \n
[2] limit 1ll \n
[2] coscan \n"
}
},
"rejectedPlans": []
},
"executionStats": {
"executionSuccess": true,
"nReturned": 2548,
"executionTimeMillis": 35,
"totalKeysExamined": 0,
"totalDocsExamined": 2548,
"executionStages": {
"stage": "mkobj",
"planNodeId": 2,
"nReturned": 2548,
"executionTimeMillisEstimate": 17,
"opens": 1,
"closes": 1,
"saveState": 9,
"restoreState": 9,
"isEOF": 1,
"objSlot": 28,
"rootSlot": 1,
"fieldBehavior": "drop",
"fields": [],
"projectFields": [
"inspection_history"
],
"projectSlots": [
27
],
"forceNewObject": true,
"returnOldObject": false,
"inputStage": {
"stage": "nlj",
"planNodeId": 2,
"nReturned": 2548,
"executionTimeMillisEstimate": 11,
"opens": 1,
"closes": 1,

```

```

"saveState": 9,
"restoreState": 9,
"isEOF": 1,
"totalDocsExamined": 2548,
"totalKeysExamined": 0,
"collectionScans": 1,
"collectionSeeks": 0,
"indexScans": 0,
"indexSeeks": 2548,
"indexesUsed": [
  "restaurant_id_1"
],
"innerOpens": 2548,
"innerCloses": 1,
"outerProjects": [
  1
],
"outerCorrelated": [
  8
],
"outerStage": {
  "stage": "nlj",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 0,
  "opens": 1,
  "closes": 1,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 1,
  "totalDocsExamined": 2548,
  "totalKeysExamined": 0,
  "collectionScans": 1,
  "collectionSeeks": 0,
  "indexScans": 0,
  "indexSeeks": 0,
  "indexesUsed": [],
  "innerOpens": 2548,
  "innerCloses": 1,
  "outerProjects": [
    1
  ],
  "outerCorrelated": [
    1
  ],
  "outerStage": {
    "stage": "scan",

```

```

    "planNodeId": 1,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 1,
    "closes": 1,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "numReads": 2548,
    "recordSlot": 1,
    "recordIdSlot": 2,
    "scanFieldNames": [],
    "scanFieldSlots": []
  },
  "innerStage": {
    "stage": "project",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 1,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "projections": {
      "8": "\n  if isArrayEmpty(s6) \n    then [null] \n    else s6 \n"
    },
    "inputStage": {
      "stage": "group",
      "planNodeId": 2,
      "nReturned": 2548,
      "executionTimeMillisEstimate": 0,
      "opens": 2548,
      "closes": 1,
      "saveState": 9,
      "restoreState": 9,
      "isEOF": 1,
      "groupBySlots": [],
      "expressions": {
        "6": "addToSet(s4) ",
        "initExprs": {
          "6": null
        }
      },
      "mergingExprs": {
        "7": "aggSetUnion(s7) "
      }
    }
  },

```

```

"usedDisk": false,
"spills": 0,
"spilledBytes": 0,
"spilledRecords": 0,
"spilledDataStorageSize": 0,
"inputStage": {
  "stage": "unwind",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 0,
  "opens": 2548,
  "closes": 2548,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 1,
  "inputSlot": 3,
  "outSlot": 4,
  "outIndexSlot": 5,
  "preserveNullAndEmptyArrays": 1,
  "inputStage": {
    "stage": "project",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "projections": {
      "3": "getField(s1, \"_id\") "
    },
    "inputStage": {
      "stage": "limit",
      "planNodeId": 2,
      "nReturned": 2548,
      "executionTimeMillisEstimate": 0,
      "opens": 2548,
      "closes": 2548,
      "saveState": 9,
      "restoreState": 9,
      "isEOF": 1,
      "limit": 1,
      "inputStage": {
        "stage": "coscan",
        "planNodeId": 2,
        "nReturned": 2548,

```

```

        "executionTimeMillisEstimate": 0,
        "opens": 2548,
        "closes": 2548,
        "saveState": 9,
        "restoreState": 9,
        "isEOF": 0
    }
}
}
}
}
},
"innerStage": {
    "stage": "limit",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 11,
    "opens": 2548,
    "closes": 1,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "limit": 1,
    "inputStage": {
        "stage": "union",
        "planNodeId": 2,
        "nReturned": 2548,
        "executionTimeMillisEstimate": 11,
        "opens": 2548,
        "closes": 1,
        "saveState": 9,
        "restoreState": 9,
        "isEOF": 0,
        "inputSlots": [
            25,
            26
        ],
        "outputSlots": [
            27
        ],
        "inputStages": [
            {
                "stage": "project",
                "planNodeId": 2,
                "nReturned": 0,
                "executionTimeMillisEstimate": 11,

```

```

    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "projections": {
      "25": "getElement(s23, 0) "
    },
    "inputStage": {
      "stage": "group",
      "planNodeId": 2,
      "nReturned": 0,
      "executionTimeMillisEstimate": 11,
      "opens": 2548,
      "closes": 2548,
      "saveState": 9,
      "restoreState": 9,
      "isEOF": 1,
      "groupBySlots": [],
      "expressions": {
        "23": "addToArrayCapped(s21, 16793600) ",
        "initExprs": {
          "23": null
        }
      },
      "mergingExprs": {
        "24": "aggConcatArraysCapped(s24, 16793600) "
      },
      "usedDisk": false,
      "spills": 0,
      "spilledBytes": 0,
      "spilledRecords": 0,
      "spilledDataStorageSize": 0,
      "inputStage": {
        "stage": "filter",
        "planNodeId": 2,
        "nReturned": 0,
        "executionTimeMillisEstimate": 11,
        "opens": 2548,
        "closes": 2548,
        "saveState": 9,
        "restoreState": 9,
        "isEOF": 1,
        "numTested": 0,
        "filter": "traverseF((getField(s21, \"restaurant_id\") ? : null), lambda(l1.0) {
isMember(l1.0, s8) }, true) ",
        "inputStage": {

```



```
"stage": "nlj",
"planNodeId": 2,
"nReturned": 0,
"executionTimeMillisEstimate": 10,
"opens": 2548,
"closes": 2548,
"saveState": 9,
"restoreState": 9,
"isEOF": 1,
"totalDocsExamined": 0,
"totalKeysExamined": 0,
"collectionScans": 0,
"collectionSeeks": 0,
"indexScans": 0,
"indexSeeks": 2548,
"indexesUsed": [
  "restaurant_id_1"
],
"innerOpens": 0,
"innerCloses": 0,
"outerProjects": [],
"outerCorrelated": [
  17,
  19,
  20,
  18,
  16
],
"outerStage": {
  "stage": "nlj",
  "planNodeId": 2,
  "nReturned": 0,
  "executionTimeMillisEstimate": 10,
  "opens": 2548,
  "closes": 2548,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 1,
  "totalDocsExamined": 0,
  "totalKeysExamined": 0,
  "collectionScans": 0,
  "collectionSeeks": 0,
  "indexScans": 0,
  "indexSeeks": 2548,
  "indexesUsed": [
    "restaurant_id_1"
  ],
```

```
"innerOpens": 2548,
"innerCloses": 2548,
"outerProjects": [
  16
],
"outerCorrelated": [
  14,
  15
],
"outerStage": {
  "stage": "nlj",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 10,
  "opens": 2548,
  "closes": 2548,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 1,
  "totalDocsExamined": 0,
  "totalKeysExamined": 0,
  "collectionScans": 0,
  "collectionSeeks": 0,
  "indexScans": 0,
  "indexSeeks": 0,
  "indexesUsed": [],
  "innerOpens": 2548,
  "innerCloses": 2548,
  "outerProjects": [],
  "outerCorrelated": [
    9
  ],
  "outerStage": {
    "stage": "unwind",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 10,
    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "inputSlot": 8,
    "outSlot": 9,
    "outIndexSlot": 10,
    "preserveNullAndEmptyArrays": 1,
    "inputStage": {
```

```
"stage": "limit",
"planNodeId": 2,
"nReturned": 2548,
"executionTimeMillisEstimate": 1,
"opens": 2548,
"closes": 2548,
"saveState": 9,
"restoreState": 9,
"isEOF": 1,
"limit": 1,
"inputStage": {
  "stage": "coscan",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 0,
  "opens": 2548,
  "closes": 2548,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 0
}
},
"innerStage": {
  "stage": "project",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 0,
  "opens": 2548,
  "closes": 2548,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 1,
  "projections": {
    "14": "ks(1ll, 0, s13, 1ll) ",
    "15": "ks(1ll, 0, s13, 2ll) ",
    "16": "{\"restaurant_id\" : 1} "
  },
  "inputStage": {
    "stage": "union",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
```

```
"isEOF": 1,
"inputSlots": [
  11,
  12
],
"outputSlots": [
  13
],
"inputStages": [
  {
    "stage": "filter",
    "planNodeId": 2,
    "nReturned": 0,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "numTested": 2548,
    "filter": "(isArray(s9) && !(isMember(s11, s8))) ",
    "inputStage": {
      "stage": "project",
      "planNodeId": 2,
      "nReturned": 2548,
      "executionTimeMillisEstimate": 0,
      "opens": 2548,
      "closes": 2548,
      "saveState": 9,
      "restoreState": 9,
      "isEOF": 1,
      "projections": {
        "11": "(getElement(s9, 0) ?: undefined) "
      },
    },
    "inputStage": {
      "stage": "limit",
      "planNodeId": 2,
      "nReturned": 2548,
      "executionTimeMillisEstimate": 0,
      "opens": 2548,
      "closes": 2548,
      "saveState": 9,
      "restoreState": 9,
      "isEOF": 1,
      "limit": 1,
      "inputStage": {
        "stage": "coscan",
```

```
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 0
  }
}
},
{
  "stage": "project",
  "planNodeId": 2,
  "nReturned": 2548,
  "executionTimeMillisEstimate": 0,
  "opens": 2548,
  "closes": 2548,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 1,
  "projections": {
    "12": "s9 "
  },
  "inputStage": {
    "stage": "limit",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 1,
    "limit": 1,
    "inputStage": {
      "stage": "coscan",
      "planNodeId": 2,
      "nReturned": 2548,
      "executionTimeMillisEstimate": 0,
      "opens": 2548,
      "closes": 2548,
      "saveState": 9,
      "restoreState": 9,
      "isEOF": 0
    }
  }
}
```

```

    }
  }
]
}
}
},
"innerStage": {
  "stage": "ixseek",
  "planNodeId": 2,
  "nReturned": 0,
  "executionTimeMillisEstimate": 0,
  "opens": 2548,
  "closes": 2548,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 1,
  "indexName": "restaurant_id_1",
  "keysExamined": 0,
  "seeks": 2548,
  "numReads": 2548,
  "indexKeySlot": 18,
  "recordIdSlot": 17,
  "snapshotIdSlot": 19,
  "indexIdSlot": 20,
  "outputSlots": [],
  "indexKeysToInclude": "00000000000000000000000000000000",
  "seekKeyLow": "s14 ",
  "seekKeyHigh": "s15 "
}
},
"innerStage": {
  "stage": "limit",
  "planNodeId": 2,
  "nReturned": 0,
  "executionTimeMillisEstimate": 0,
  "opens": 0,
  "closes": 0,
  "saveState": 9,
  "restoreState": 9,
  "isEOF": 0,
  "inputStage": {
    "stage": "seek",
    "planNodeId": 2,
    "nReturned": 0,
    "executionTimeMillisEstimate": 0,
    "opens": 0,
    "closes": 0,

```

```

        "saveState": 9,
        "restoreState": 9,
        "isEOF": 0,
        "numReads": 0,
        "recordSlot": 21,
        "recordIdSlot": 22,
        "seekRecordIdSlot": 17,
        "snapshotIdSlot": 19,
        "indexIdSlot": 20,
        "indexKeySlot": 18,
        "indexKeyPatternSlot": 16,
        "scanFieldNames": [],
        "scanFieldSlots": []
    }
}
}
}
},
{
    "stage": "project",
    "planNodeId": 2,
    "nReturned": 2548,
    "executionTimeMillisEstimate": 0,
    "opens": 2548,
    "closes": 2548,
    "saveState": 9,
    "restoreState": 9,
    "isEOF": 0,
    "projections": {
        "26": "[ ]"
    },
    "inputStage": {
        "stage": "limit",
        "planNodeId": 2,
        "nReturned": 2548,
        "executionTimeMillisEstimate": 0,
        "opens": 2548,
        "closes": 2548,
        "saveState": 9,
        "restoreState": 9,
        "isEOF": 0,
        "limit": 1,
        "inputStage": {
            "stage": "coscan",
            "planNodeId": 2,
            "nReturned": 2548,

```

```

        "executionTimeMillisEstimate": 0,
        "opens": 2548,
        "closes": 2548,
        "saveState": 9,
        "restoreState": 9,
        "isEOF": 0
      }
    }
  ]
}
},
"nReturned": 2548,
"executionTimeMillisEstimate": 31
},
{
  "$match": {
    "inspection_history.result": {
      "$eq": "Violation Issued"
    }
  },
  "nReturned": 0,
  "executionTimeMillisEstimate": 35
}
],
"queryShapeHash":
"76F7192AE0A71E599FB1DAAEA8B4C6DA4F38E47A79210E509EE487E42C9921CD",
"serverInfo": {
  "host": "atsuab2025-shard-00-01.icbhz.mongodb.net",
  "port": 27017,
  "version": "8.0.5",
  "gitVersion": "cb9e2e5e552ee39dea1e39d7859336456d0c9820"
},
"serverParameters": {
  "internalQueryFacetBufferSizeBytes": 104857600,
  "internalQueryFacetMaxOutputDocSizeBytes": 104857600,
  "internalLookupStageIntermediateDocumentMaxSizeBytes": 16793600,
  "internalDocumentSourceGroupMaxMemoryBytes": 104857600,
  "internalQueryMaxBlockingSortMemoryUsageBytes": 33554432,
  "internalQueryProhibitBlockingMergeOnMongoS": 0,
  "internalQueryMaxAddToSetBytes": 104857600,
  "internalDocumentSourceSetWindowFieldsMaxMemoryBytes": 104857600,
  "internalQueryFrameworkControl": "trySbeRestricted",

```



```
"internalQueryPlannerIgnoreIndexWithCollationForRegex": 1
},
"command": {
  "aggregate": "restaurants",
  "pipeline": [
    {
      "$lookup": {
        "from": "inspections",
        "localField": "_id",
        "foreignField": "restaurant_id",
        "as": "inspection_history"
      }
    },
    {
      "$match": {
        "inspection_history.result": "Violation Issued"
      }
    }
  ],
  "cursor": {},
  "$db": "PLAB01"
},
"ok": 1,
"$clusterTime": {
  "clusterTime": {
    "$timestamp": {
      "t": 1742379620,
      "i": 13
    }
  },
  "signature": {
    "hash": {
      "$binary": {
        "base64": "rOqTWrCvRJCStX2UQF5N2/GPjtE=",
        "subType": "00"
      }
    },
    "keyId": 7441318053373018000
  },
  "operationTime": {
    "$timestamp": {
      "t": 1742379620,
      "i": 13
    }
  }
}
```