

Track: B20-AI-01	Week 3
Name: Artem Chernitsa	Introduction to Big Data
Email: a.chernitsa@innopolis.university	04.04.2023

Report

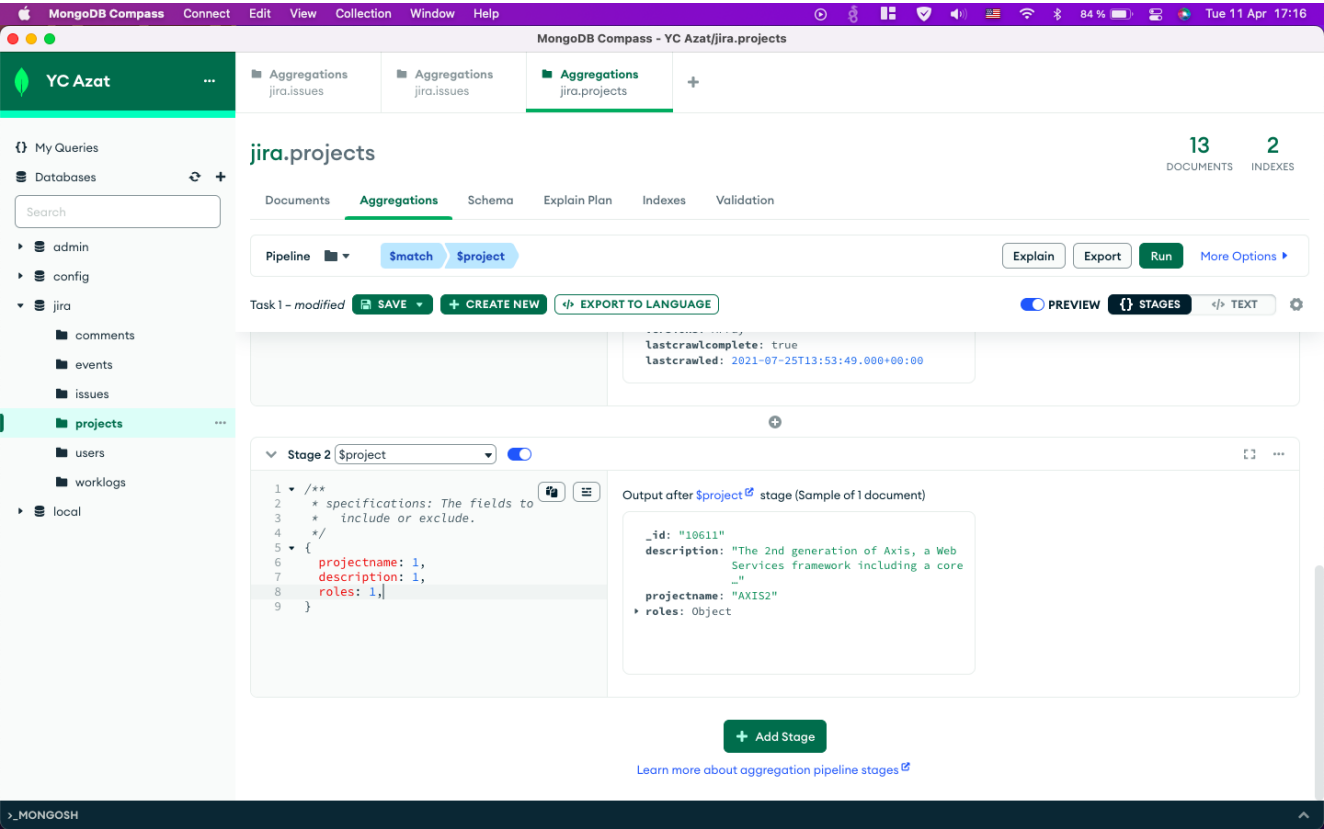
```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
kot_mapku3@afrodita:~/Developer/s23/bd/assignment03

./assignment03 (-zsh)
2023-04-10T11:50:44.160+0300 preparing collections to restore from
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/comments.bson.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/comments.metadata.json.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/events.bson.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/events.metadata.json.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/issues.bson.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/issues.metadata.json.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/projects.bson.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/projects.metadata.json.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/users.bson.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/users.metadata.json.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/worklogs.bson.gz", skipping...
2023-04-10T11:50:44.161+0300 don't know what to do with file "JIRA/worklogs.metadata.json.gz", skipping...
2023-04-10T11:50:44.161+0300 0 document(s) restored successfully. 0 document(s) failed to restore.
> mongorestore --gzip --db jira JIRA
2023-04-10T11:51:01.132+0300 The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclude=${DATABASE}.${COLLECTION}
2023-04-10T11:51:01.132+0300 building a list of collections to restore from JIRA dir
2023-04-10T11:51:01.135+0300 reading metadata for jira.worklogs from JIRA/worklogs.metadata.json.gz
2023-04-10T11:51:01.139+0300 reading metadata for jira.comments from JIRA/comments.metadata.json.gz
2023-04-10T11:51:01.142+0300 reading metadata for jira.events from JIRA/events.metadata.json.gz
2023-04-10T11:51:01.143+0300 reading metadata for jira.issues from JIRA/issues.metadata.json.gz
2023-04-10T11:51:01.144+0300 reading metadata for jira.projects from JIRA/projects.metadata.json.gz
2023-04-10T11:51:01.144+0300 reading metadata for jira.users from JIRA/users.metadata.json.gz
2023-04-10T11:51:01.146+0300 finished restoring jira.comments (0 documents, 0 failures)
2023-04-10T11:51:01.146+0300 Failed: jira.comments: error reading database: (Unauthorized) command listCollections requires authentication
2023-04-10T11:51:01.146+0300 0 document(s) restored successfully. 0 document(s) failed to restore.
> mongorestore --host localhost:27017 --u artem --p [REDACTED] --gzip --db jira JIRA
2023-04-10T11:53:53.887+0300 error connecting to host: could not connect to server: connection() error occurred during connection handshake: auth error: sasl conversation error: unable to authenticate
2023-04-10T11:53:53.887+0300 using mechanism "SCRAM-SHA-1": (AuthenticationFailed) Authentication failed.
> mongorestore --host localhost:27017 --u artem --p [REDACTED] --authenticationDatabase=admin --gzip --db jira JIRA
2023-04-10T11:54:50.857+0300 The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclude=${DATABASE}.${COLLECTION}
2023-04-10T11:54:50.857+0300 building a list of collections to restore from JIRA dir
2023-04-10T11:54:50.859+0300 reading metadata for jira.projects from JIRA/projects.metadata.json.gz
2023-04-10T11:54:50.866+0300 reading metadata for jira.users from JIRA/users.metadata.json.gz
2023-04-10T11:54:50.870+0300 reading metadata for jira.worklogs from JIRA/worklogs.metadata.json.gz
2023-04-10T11:54:50.873+0300 reading metadata for jira.comments from JIRA/comments.metadata.json.gz
2023-04-10T11:54:50.877+0300 reading metadata for jira.events from JIRA/events.metadata.json.gz
2023-04-10T11:54:50.877+0300 reading metadata for jira.issues from JIRA/issues.metadata.json.gz
2023-04-10T11:54:50.984+0300 restoring jira.worklogs from JIRA/worklogs.bson.gz
2023-04-10T11:54:51.028+0300 restoring jira.events from JIRA/events.bson.gz
2023-04-10T11:54:51.047+0300 restoring jira.issues from JIRA/issues.bson.gz
2023-04-10T11:54:51.142+0300 restoring jira.comments from JIRA/comments.bson.gz
2023-04-10T11:54:53.852+0300 [.....] jira.worklogs 1.79MB/115MB (1.6%)
2023-04-10T11:54:53.852+0300 [.....] jira.events 1.11MB/324MB (0.3%)
2023-04-10T11:54:53.852+0300 [.....] jira.issues 2.34MB/729MB (0.3%)
2023-04-10T11:54:53.852+0300 [.....] jira.comments 2.31MB/765MB (0.3%)
2023-04-10T11:54:53.852+0300 [.....] jira.worklogs 4.73MB/115MB (4.1%)
2023-04-10T11:54:56.852+0300 [.....] jira.events 2.07MB/324MB (0.6%)
2023-04-10T11:54:56.852+0300 [.....] jira.issues 4.67MB/729MB (0.6%)
2023-04-10T11:54:56.852+0300 [.....] jira.comments 4.97MB/765MB (0.6%)

./assignment03 (-zsh)
2023-04-10T12:20:11.745+0300 [#####] jira.comments 758MB/765MB (99.1%)
2023-04-10T12:20:14.745+0300 [#####] jira.events 303MB/324MB (93.5%)
2023-04-10T12:20:14.745+0300 [#####] jira.comments 760MB/765MB (99.4%)
2023-04-10T12:20:14.745+0300 [#####] jira.events 303MB/324MB (93.7%)
2023-04-10T12:20:17.745+0300 [#####] jira.comments 762MB/765MB (99.6%)
2023-04-10T12:20:17.745+0300 [#####] jira.events 304MB/324MB (93.9%)
2023-04-10T12:20:20.745+0300 [#####] jira.comments 763MB/765MB (99.7%)
2023-04-10T12:20:20.746+0300 [#####] jira.comments 765MB/765MB (100.0%)
2023-04-10T12:20:23.276+0300 finished restoring jira.comments (4639882 documents, 0 failures)
2023-04-10T12:20:23.276+0300 [#####] jira.events 305MB/324MB (94.2%)
2023-04-10T12:20:26.745+0300 [#####] jira.events 306MB/324MB (94.7%)
2023-04-10T12:20:29.745+0300 [#####] jira.events 308MB/324MB (95.0%)
2023-04-10T12:20:32.745+0300 [#####] jira.events 309MB/324MB (95.4%)
2023-04-10T12:20:35.745+0300 [#####] jira.events 311MB/324MB (96.0%)
2023-04-10T12:20:38.745+0300 [#####] jira.events 312MB/324MB (96.5%)
2023-04-10T12:20:41.745+0300 [#####] jira.events 313MB/324MB (96.8%)
2023-04-10T12:20:44.745+0300 [#####] jira.events 315MB/324MB (97.4%)
2023-04-10T12:20:47.747+0300 [#####] jira.events 316MB/324MB (97.8%)
2023-04-10T12:20:50.745+0300 [#####] jira.events 318MB/324MB (98.2%)
2023-04-10T12:20:53.745+0300 [#####] jira.events 319MB/324MB (98.6%)
2023-04-10T12:20:56.745+0300 [#####] jira.events 321MB/324MB (99.1%)
2023-04-10T12:20:59.745+0300 [#####] jira.events 322MB/324MB (99.6%)
2023-04-10T12:21:02.744+0300 [#####] jira.events 323MB/324MB (99.9%)
2023-04-10T12:21:03.224+0300 [#####] jira.events 324MB/324MB (100.0%)
2023-04-10T12:21:03.224+0300 finished restoring jira.events (7583864 documents, 0 failures)
2023-04-10T12:21:03.229+0300 restoring indexes for collection jira.users from metadata
2023-04-10T12:21:03.233+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"projectname_1", "v":2, Key:primitive.D(primitive.E(Key:"projectname", Value:1)), PartialFilterExpression:primitive.D(
nil)})
2023-04-10T12:21:03.233+0300 restoring indexes for collection jira.issues from metadata
2023-04-10T12:21:03.233+0300 restoring indexes for collection jira.events from metadata
2023-04-10T12:21:03.233+0300 no indexes to restore for collection jira.projects
2023-04-10T12:21:03.233+0300 restoring indexes for collection jira.worklogs from metadata
2023-04-10T12:21:03.233+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"issue_1", "v":2, Key:primitive.D(primitive.E(Key:"issue", Value:1)), PartialFilterExpression:primitive.D(nil)})
2023-04-10T12:21:03.233+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"projectname_1", "v":2, Key:primitive.D(primitive.E(Key:"projectname", Value:1)), PartialFilterExpression:primitive.D(
nil)})
2023-04-10T12:21:03.234+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"issue_1", "v":2, Key:primitive.D(primitive.E(Key:"issue", Value:1)), PartialFilterExpression:primitive.D(nil)})
2023-04-10T12:21:03.234+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"projectname_1", "v":2, Key:primitive.D(primitive.E(Key:"projectname", Value:1)), PartialFilterExpression:primitive.D(
nil)})
2023-04-10T12:21:03.234+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"projectname_1", "v":2, Key:primitive.D(primitive.E(Key:"projectname", Value:1)), PartialFilterExpression:primitive.D(
nil)})
2023-04-10T12:21:06.270+0300 restoring indexes for collection jira.comments from metadata
2023-04-10T12:21:06.270+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"issue_1", "v":2, Key:primitive.D(primitive.E(Key:"issue", Value:1)), PartialFilterExpression:primitive.D(nil)})
2023-04-10T12:21:06.270+0300 index: 6idx.IndexDocument(Options:primitive.M{"name":"projectname_1", "v":2, Key:primitive.D(primitive.E(Key:"projectname", Value:1)), PartialFilterExpression:primitive.D(
nil)})
2023-04-10T12:23:16.744+0300 13720869 document(s) restored successfully. 0 document(s) failed to restore.
~/Developer/s23/bd/assignment03
28m 26s 12:23:16 PM
```

Task 1

Your query as a text	Collection name
<pre>[{ \$match: /** * query: The query in MQL. */ { projectname: { \$regex: "^A", }, }, }, { \$project: /** * specifications: The fields to * include or exclude. */ { projectname: 1, description: 1, roles: 1, }, },]</pre>	projects

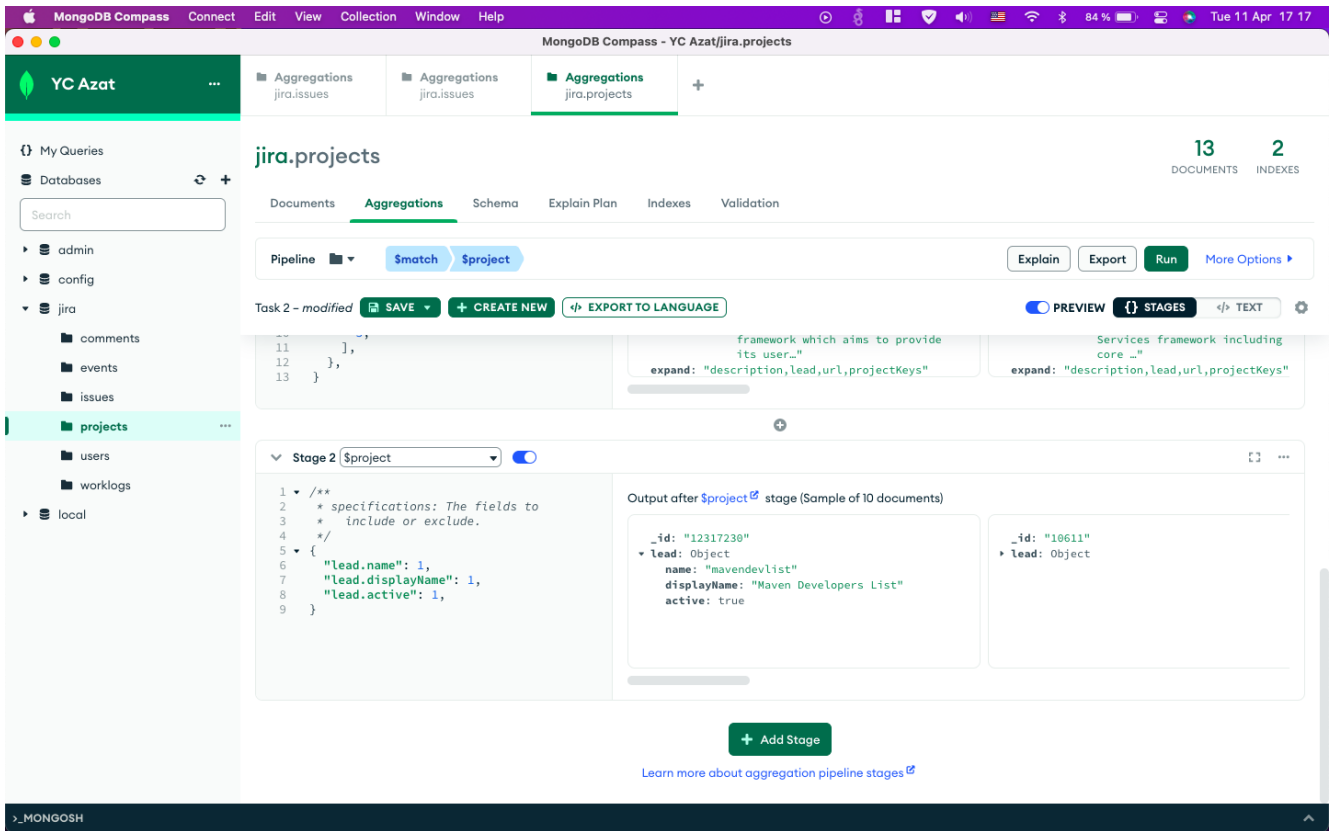


Your comments (optional)

Task 2

Your query as a text	Collection name
<pre>[{ \$match: /** * query: The query in MQL. */ { \$expr: { \$gt: [{ \$size: "\$issueTypes", }, 5,], }, }, },], }</pre>	projects

<pre>\$project: /** * specifications: The fields to * include or exclude. */ { "lead.name": 1, "lead.displayName": 1, "lead.active": 1, }, },]</pre>	
--	--

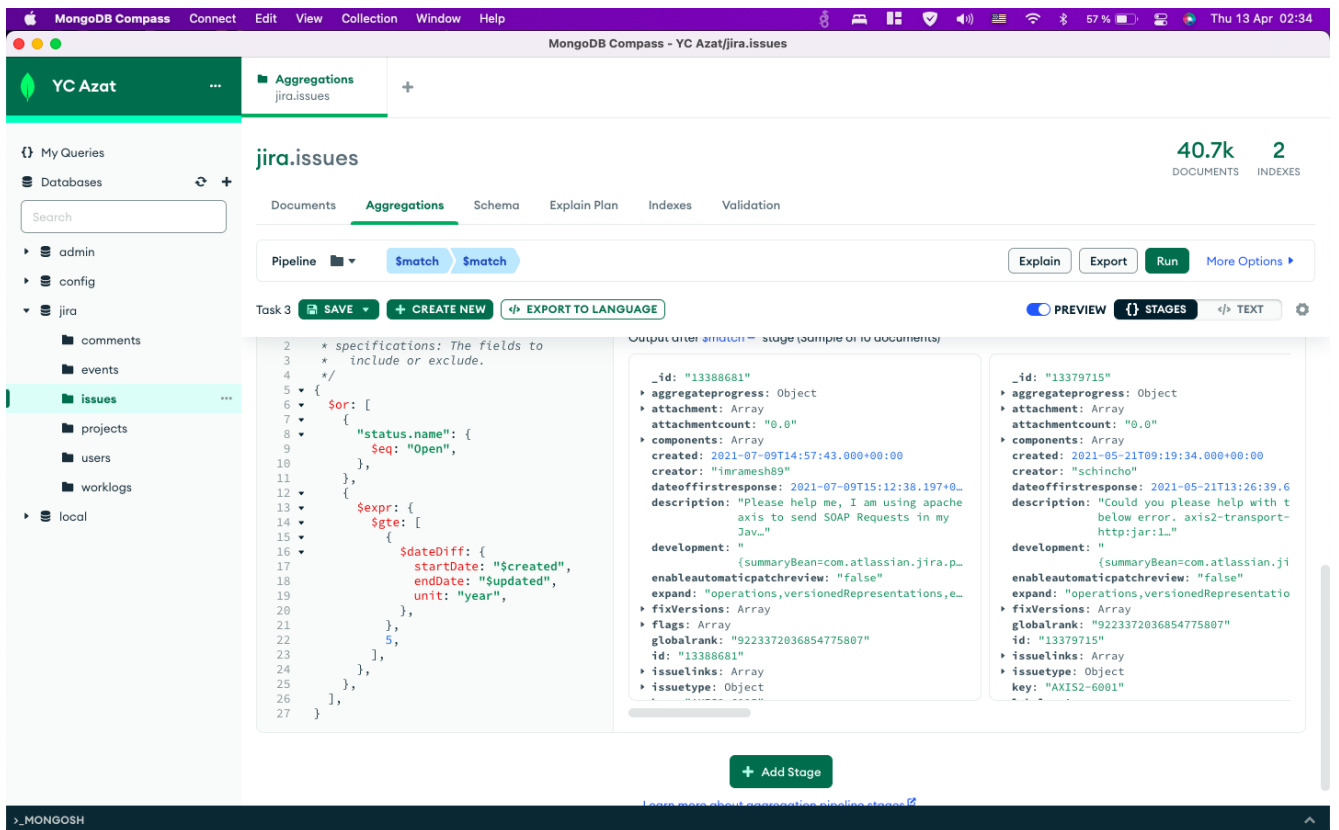


Your comments (optional)

Task 3

Your query as a text	Collection name
<pre>[{ \$match:</pre>	issues

```
/**
 * query: The query in MQL.
 */
{
  "issuetype.name": {
    $eq: "Bug",
  },
},
{
  $match:
  /**
   * specifications: The fields to
   * include or exclude.
   */
  {
    $or: [
      {
        "status.name": {
          $eq: "Open",
        },
      },
      {
        $expr: {
          $gte: [
            {
              $dateDiff: {
                startDate: "$created",
                endDate: "$updated",
                unit: "year",
              },
            },
            5,
          ],
        },
      },
    ],
  },
},
],
},
]
```



Your comments (optional)

Task 4

Your query as a text

Collection name

< put here the screenshot of the output>

Your comments (optional)

Task 5

Your query as a text

Collection name

```
[
{
```

events

```

$group: {
  _id: {
    issue: "$issue",
    projectname: "$projectname",
  },
  counter: {
    $sum: 1,
  },
},
},
{
  $addFields:
  /**
   * newField: The new field name.
   * expression: The new field expression.
   */
  {
    issue: "$_id.issue",
    projectname: "$_id.projectname",
  },
},
}
]

```

The screenshot shows the MongoDB Compass interface for the `jira.events` collection. The aggregation pipeline is defined as follows:

```

1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   issue: "$_id.issue",
7   projectname: "$_id.projectname",
8 }

```

The output after the `$addFields` stage (Sample of 10 documents) is shown on the right:

```

{ "_id": { "issue": "12769767", "projectname": "DRILL" }, "counter": 4 }
{ "_id": { "issue": "12985725", "projectname": "DISPATCH" }, "counter": 6 }

```

Your comments (optional)

Task 6

Your query as a text	Collection name
<pre>[{ \$group: /** * _id: The id of the group. * fieldN: The first field name. */ { _id: { author: "\$author", projectname: "\$projectname", }, timeSpentSeconds: { \$avg: "\$timeSpentSeconds", }, }, },]</pre>	worklogs

The screenshot displays the MongoDB Compass interface for the 'jira.worklogs' collection. The 'Aggregations' tab is active, showing a single stage named '\$group'. The stage configuration is as follows:

```
1  /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5  {
6    _id: {
7      author: "$author",
8      projectname: "$projectname",
9    },
10   timeSpentSeconds: {
11     $avg: "$timeSpentSeconds",
12   },
13 }
```

The output after the '\$group' stage (Sample of 10 documents) is shown as:

```
{
  "_id": Object,
  "author": "jdcasey",
  "projectname": "HPH",
  "timeSpentSeconds": 3000
}
```

The interface also shows a sidebar with databases and collections, and a top bar with the MongoDB Compass logo and navigation options.

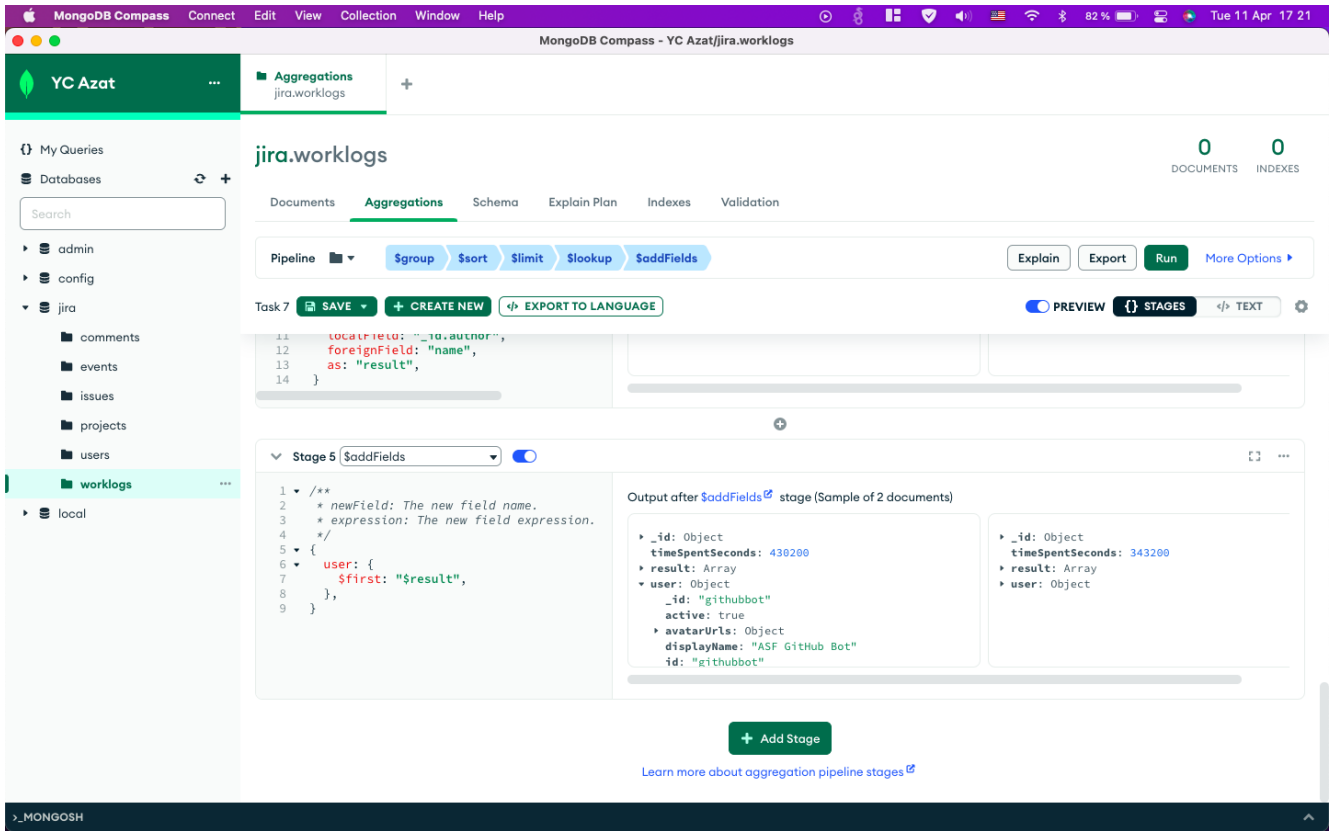
Your comments (optional)

There's a bug in showing the number of documents in the collection.

Task 7

Your query as a text	Collection name
<pre>[{ \$group: /** * _id: The id of the group. * fieldN: The first field name. */ { _id: { author: "\$author", }, timeSpentSeconds: { \$sum: "\$timeSpentSeconds", }, }, }, { \$sort: /** * Provide any number of field/order pairs. */ { timeSpentSeconds: -1, }, }, { \$limit: /** * Provide the number of documents to limit. */ 2, }, { \$lookup: /** * from: The target collection. * localField: The local join field. * foreignField: The target join field. * as: The name for the results. * pipeline: Optional pipeline to run on the foreign collection. * let: Optional variables to use in the pipeline field stages. */ { from: "users", localField: "_id.author", foreignField: "name",</pre>	worklogs

```
    as: "result",
  },
},
{
  $addFields:
  /**
   * newField: The new field name.
   * expression: The new field expression.
   */
  {
    user: {
      $first: "$result",
    },
  },
},
},
]
```



Your comments (optional)

Task 8

Your query as a text

Collection name

--	--

< put here the screenshot of the output>

Your comments (optional)

Task 9

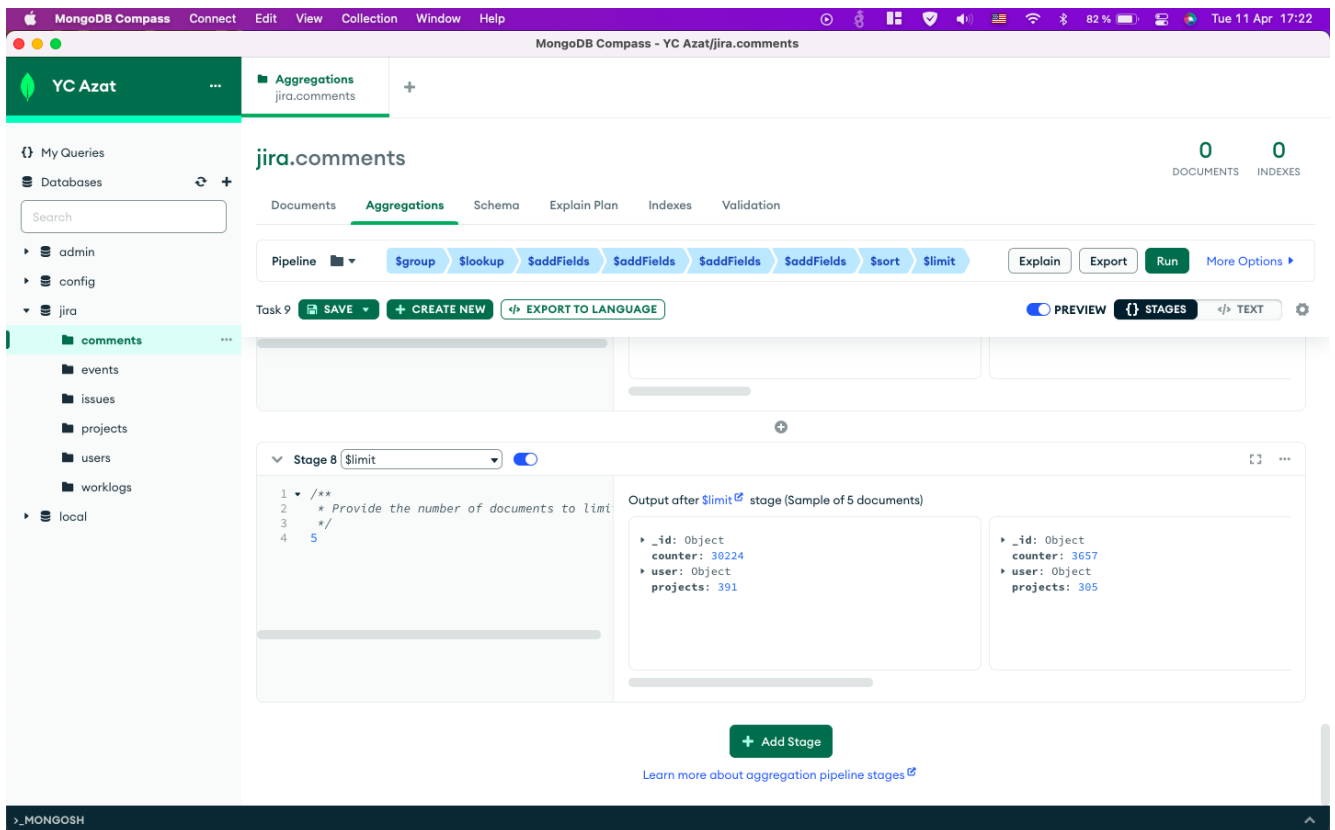
Your query as a text	Collection name
<pre>[{ \$group: /** * _id: The id of the group. * fieldN: The first field name. */ { _id: { user: "\$author", }, counter: { \$sum: 1, }, }, }, { \$lookup: /** * from: The target collection. * localField: The local join field. * foreignField: The target join field. * as: The name for the results. * pipeline: Optional pipeline to run on the foreign collection. * let: Optional variables to use in the pipeline field stages. */ { from: "users", localField: "_id.user", foreignField: "id", as: "user", }, }, { \$addFields: /** * newField: The new field name.</pre>	comments

```

    * expression: The new field expression.
    */
    {
      projects: {
        $first: "$user",
      },
    },
  },
  {
    $addFields:
    /**
     * newField: The new field name.
     * expression: The new field expression.
     */
    {
      projects: "$projects.projectname",
    },
  },
  {
    $addFields:
    /**
     * newField: The new field name.
     * expression: The new field expression.
     */
    {
      projects: {
        $size: "$projects",
      },
    },
  },
  {
    $addFields:
    /**
     * newField: The new field name.
     * expression: The new field expression.
     */
    {
      user: {
        $first: "$user",
      },
    },
  },
  {
    $sort:
    /**
     * Provide any number of field/order pairs.
     */
    {
      counter: -1,
      projects: -1,
    },
  },
}

```

<pre>{ \$limit: /** * Provide the number of documents to limit. */ 5, },]</pre>	
--	--



Your comments (optional)

Task 10

Your query as a text	Collection name
<pre>[{ \$group: /** * _id: The id of the group. * fieldN: The first field name. */ {</pre>	comments

```

    _id: {
      user: "$author",
    },
    counter: {
      $sum: 1,
    },
  },
},
{
  $lookup:
  /**
   * from: The target collection.
   * localField: The local join field.
   * foreignField: The target join field.
   * as: The name for the results.
   * pipeline: Optional pipeline to run on the foreign collection.
   * let: Optional variables to use in the pipeline field stages.
   */
  {
    from: "users",
    localField: "_id.user",
    foreignField: "id",
    as: "user",
  },
},
{
  $addFields:
  /**
   * newField: The new field name.
   * expression: The new field expression.
   */
  {
    projects: {
      $first: "$user",
    },
  },
},
{
  $addFields:
  /**
   * newField: The new field name.
   * expression: The new field expression.
   */
  {
    projects: "$projects.projectname",
  },
},
{
  $addFields:
  /**
   * newField: The new field name.
   * expression: The new field expression.

```

```

    */
    {
      projects: {
        $size: "$projects",
      },
    },
  },
  {
    $addFields:
    /**
     * newField: The new field name.
     * expression: The new field expression.
     */
    {
      user: {
        $first: "$user",
      },
    },
  },
  {
    $sort:
    /**
     * Provide any number of field/order pairs.
     */
    {
      counter: -1,
      projects: -1,
    },
  },
  {
    $limit:
    /**
     * Provide the number of documents to limit.
     */
    5,
  },
  {
    $lookup:
    /**
     * from: The target collection.
     * localField: The local join field.
     * foreignField: The target join field.
     * as: The name for the results.
     * pipeline: Optional pipeline to run on the foreign collection.
     * let: Optional variables to use in the pipeline field stages.
     */
    {
      from: "events",
      localField: "user.id",
      foreignField: "author",
      as: "event",
    },
  },

```

},

]

MongoDB Compass

Connect

Edit

View

Collection

Window

Help

MongoDB Compass - YC Azat/jira.comments

YC Azat

Aggregations

jira.comments

My Queries

Databases

admin

config

jira

comments

events

issues

projects

users

worklogs

local

jira.comments

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

Pipeline

\$group

\$lookup

\$addFields

\$addFields

\$addFields

\$addFields

\$sort

+2

Explain

Export

Run

More Options

Untitled - modified

SAVE

CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

Stage 9

\$lookup

Output after \$lookup stage (Sample of 5 documents)

1 /**

2 * from: The target collection.

3 * localField: The local join field.

4 * foreignField: The target join field.

5 * as: The name for the results.

6 * pipeline: Optional pipeline to run on the

7 * let: Optional variables to use in the p

8 */

9 {

10 from: "events",

11 localField: "user.id",

12 foreignField: "author",

13 as: "event",

14 }

_id: Object

counter: 30224

user: Object

projects: 391

event: Array

0: Object

_id: "15566640"

author: "githubbot"

created: 2016-08-16T06:57:14.437+00:00

id: HYPERREALM

_id: Object

counter: 3657

user: Object

projects: 305

event: Array

+ Add Stage

Learn more about aggregation pipeline stages

MongoDB Compass

Connect

Edit

View

Collection

Window

Help

MongoDB Compass - YC Azat/jira.comments

Stage 9: \$lookup

ENABLED

ADD STAGE

STAGE INPUT

Sample of 5 documents

_id: Object

counter: 30225

user: Object

projects: 391

_id: Object

counter: 3660

user: Object

projects: 305

_id: Object

counter: 3316

user: Object

projects: 83

_id: Object

counter: 2873

user: Object

projects: 20

_id: Object

counter: 1973

user: Object

projects: 38

STAGE OUTPUT

Sample of 5 documents

_id: Object

counter: 30225

user: Object

_id: "githubbot"

active: true

avatarUrls: Object

displayName: "ASF GitHub Bot"

id: "githubbot"

key: "githubbot"

name: "githubbot"

projectname: Array

self: "https://issues.apache.org/jir_

username=githubbot"

timeZone: "Etc/UTC"

projects: 391

event: Array

0: Object

_id: "15566640"

author: "githubbot"

created: 2016-08-16T06:57:14.437+...

id: "15566640"

issue: "12997467"

items: Array

projectname: "AXIS2"

1: Object

2: Object

3: Object

4: Object

5: Object

6: Object

7: Object

8: Object

9: Object

10: Object

11: Object

12: Object

13: Object

14: Object

Your comments (optional)

Task 11

Your query as a text	Collection name
<pre>[{ \$addFields: /** * newField: The new field name. * expression: The new field expression. */ { date: { \$dateDiff: { startDate: "\$created", endDate: new ISODate(), unit: "year", }, }, }, }, { \$group: /** * _id: The id of the group. * fieldN: The first field name. */ { _id: "\$projectname", last_year: { \$min: "\$date", }, }, }, { \$match: /** * query: The query in MQL. */ { \$expr: { \$gte: ["\$last_year", 5], }, }, },]</pre>	issues

The screenshot shows the MongoDB Compass interface. On the left, the 'jira' database is selected, and the 'issues' collection is highlighted. The main panel shows the 'Aggregations' tab for the 'jira.issues' collection. The aggregation pipeline is defined as follows:

```

1  /**
2   * query: The query in MQL.
3   */
4  {
5    $expr: {
6      $gte: ["$last_year", 5],
7    },
8  }

```

The output after the \$match stage (Sample of 2 documents) is shown:

```

_id: "JCRMI"
last_year: 8

_id: "TRANSACTION"
last_year: 10

```

Buttons for 'Pipeline', 'AddFields', 'Group', 'Match', 'Explain', 'Export', 'Run', and 'More Options' are visible. The status bar at the bottom indicates 'MONGODB'.

Your comments (optional)

Task 12

Your query as a text	Collection name
<pre>[{ \$match: /** * query: The query in MQL. */ { \$text: { \$search: "Google Apache", \$caseSensitive: true, }, }, },]</pre>	projects

The screenshot shows the MongoDB Compass interface for the 'jira.projects' collection. The left sidebar shows the database structure with 'projects' selected. The main area displays an aggregation pipeline with one stage named '\$match'. The query for the '\$match' stage is: `{ $text: { $search: 'Google Apache', $caseSensitive: true } }`. The output of the pipeline shows a sample document: `{ _id: '12313820', archived: false, assigneeType: 'UNASSIGNED', avatarUrls: Object, components: Array, description: 'Drill is a distributed system for interactive analysis of large-scale _', expand: 'description,lead,url,projectKeys' }`.

Your comments (optional)

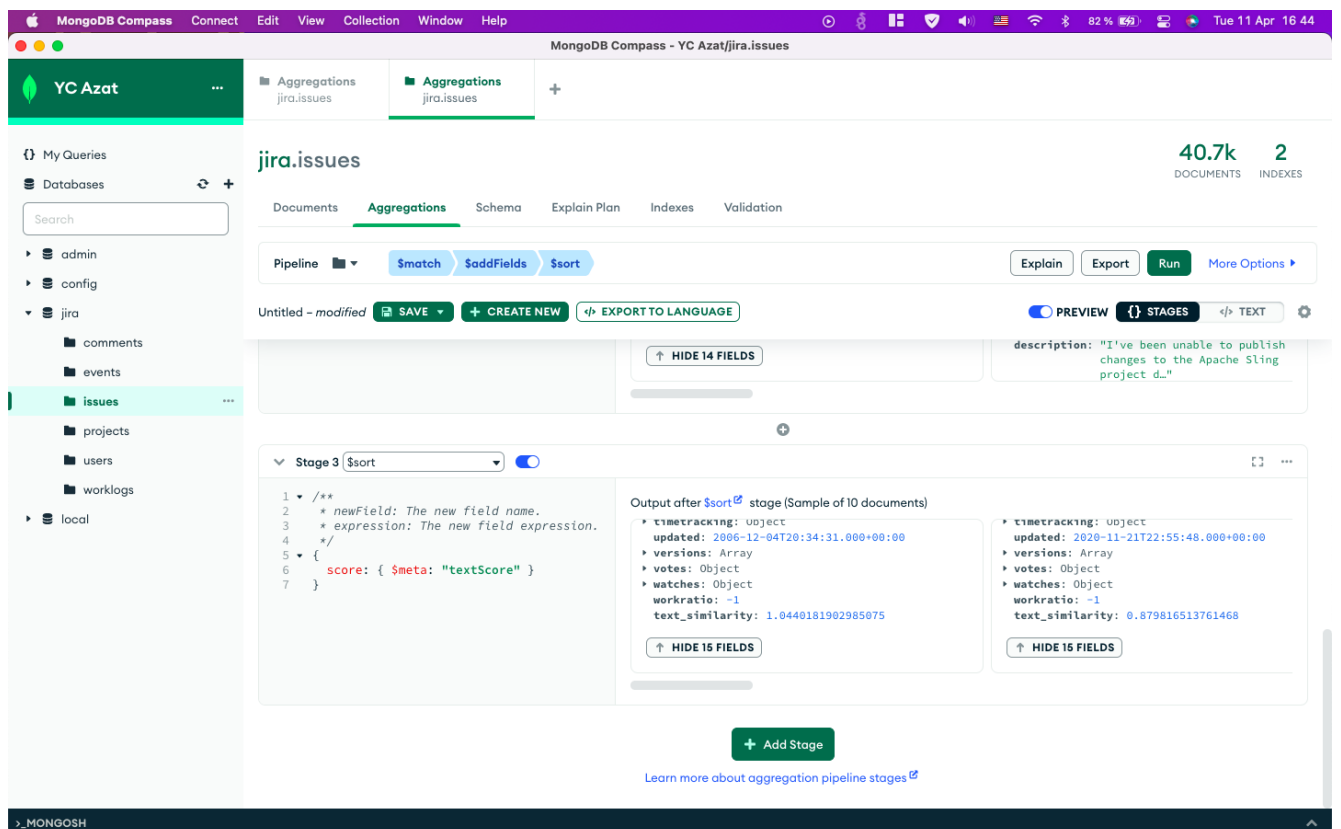
Task 13

Your query as a text	Collection name
<pre>[{ \$match: /** * query: The query in MQL. */ { \$text: { \$search: "game", \$caseSensitive: false, \$diacriticSensitive: false, }, }, }, { \$addFields: /** * newField: The new field name. * expression: The new field expression. */ } }</pre>	issues

```

    */
    {
      text_similarity: {
        $meta: "textScore",
      },
    },
  },
  {
    $sort:
    /**
     * newField: The new field name.
     * expression: The new field expression.
     */
    {
      score: {
        $meta: "textScore",
      },
    },
  },
}
]

```



The screenshot displays the MongoDB Compass web interface. On the left, a sidebar shows the database structure with 'jira' selected. The main panel shows the 'jira.issues' collection with 40.7k documents and 2 indexes. An aggregation pipeline is configured with three stages: '\$match', '\$addFields', and '\$sort'. The '\$sort' stage is active, showing a sample output of 10 documents. The output shows documents sorted by a 'score' field, which is calculated using the '\$meta: "textScore"' operator. The interface includes buttons for 'Run', 'Export', and 'More Options' for the pipeline. A bottom status bar shows the command prompt with '>_MONGOSH'.

Your comments (optional)