

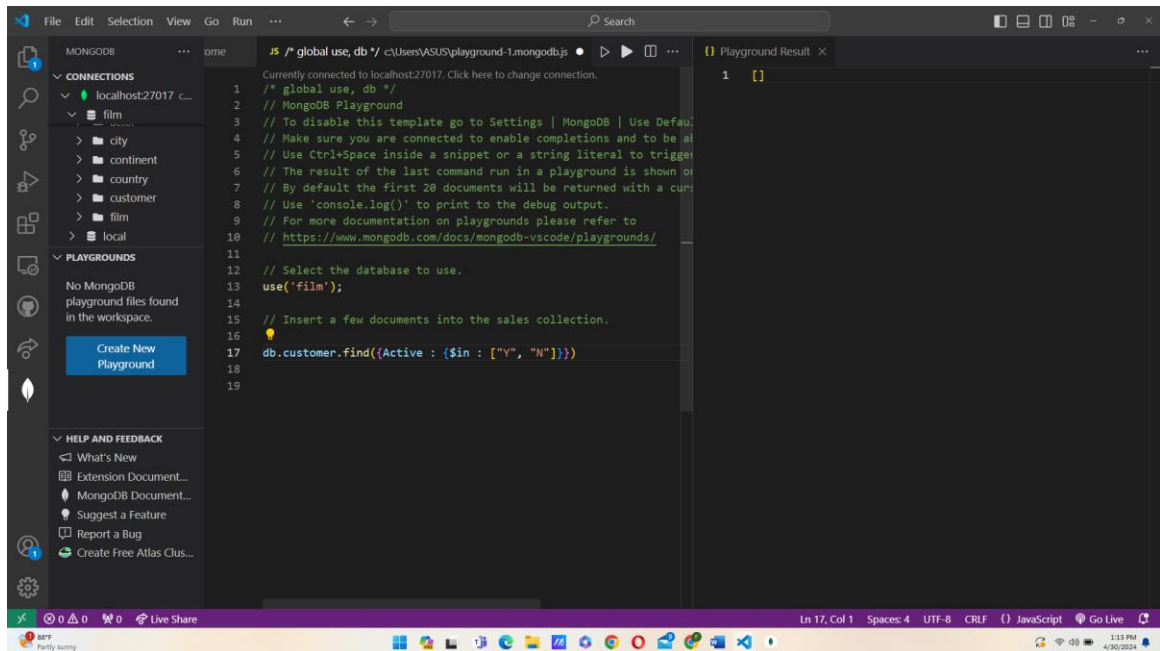
Nama : Frila Cahya Wardani

Kelas : TI – 21 – PA

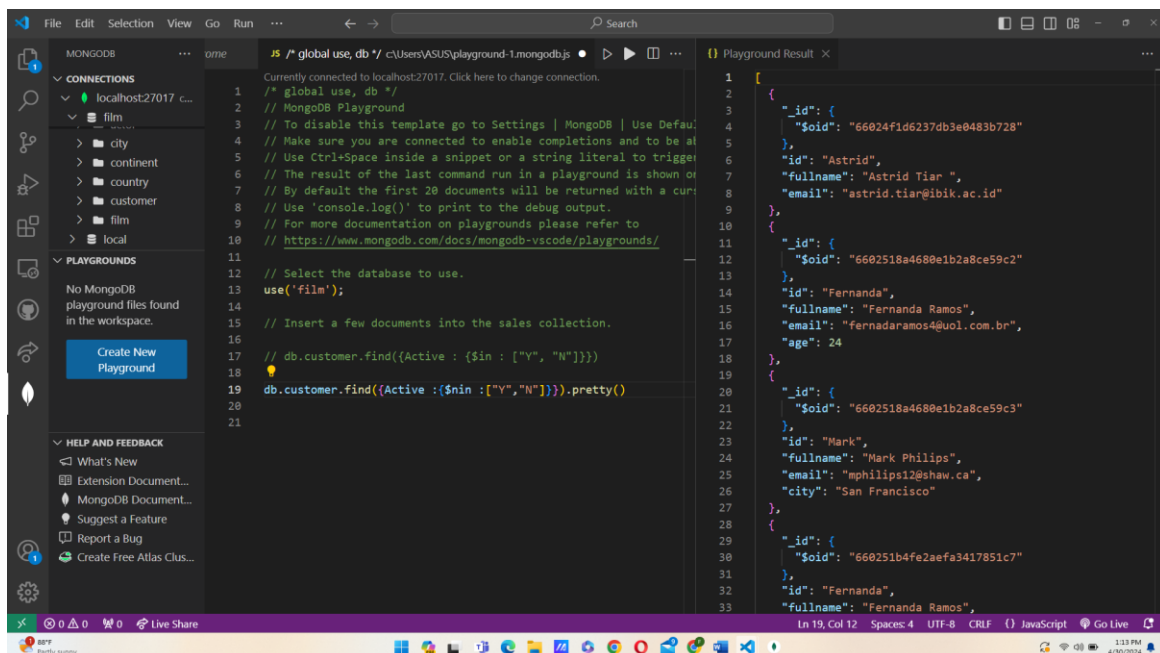
NPM : 212310014

Lab. Basis Data Lanjut Modul 5

1. Buat dan jalankan perintah untuk mencari data dari collection Customer menggunakan minimal 2 buah variasi dari perintah \$in dan \$nin



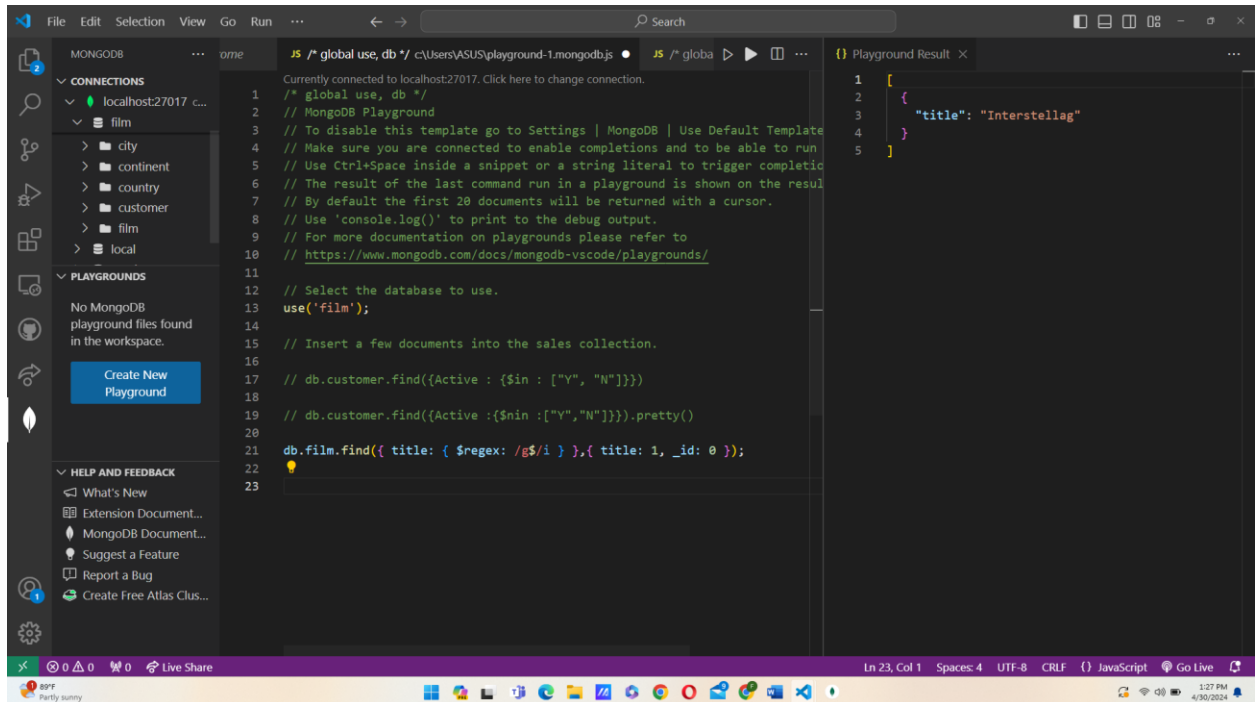
The screenshot shows the MongoDB Playground interface in VS Code. The left sidebar displays the 'CONNECTIONS' list with 'localhost:27017 c...' selected, and the 'PLAYGROUNDS' section with a 'Create New Playground' button. The main editor area contains a JavaScript script for the MongoDB Playground. The script includes comments about disabling the template, connecting to the database, and inserting documents. The query being executed is `db.customer.find({Active: {$in: ["Y", "N"]}})`. The 'Playground Result' panel on the right shows the output as an empty array `[]`.



The screenshot shows the MongoDB Playground interface in VS Code. The left sidebar displays the 'CONNECTIONS' list with 'localhost:27017 c...' selected, and the 'PLAYGROUNDS' section with a 'Create New Playground' button. The main editor area contains a JavaScript script for the MongoDB Playground. The script includes comments about disabling the template, connecting to the database, and inserting documents. The query being executed is `db.customer.find({Active: {$nin: ["Y", "N"]}}).pretty()`. The 'Playground Result' panel on the right shows the output as a JSON array of three documents:

```
1 {
2   "_id": {
3     "$oid": "66024fd6237db3e0483b728"
4   },
5   "id": "Astrid",
6   "fullname": "Astrid Tiar ",
7   "email": "astrid.tiar@ibik.ac.id"
8 },
9 {
10  "_id": {
11    "$oid": "6602518a4680e1b2a8ce59c2"
12  },
13  "id": "Fernanda",
14  "fullname": "Fernanda Ramos",
15  "email": "fernadaramos4@uol.com.br",
16  "age": 24
17 },
18 {
19  "_id": {
20    "$oid": "6602518a4680e1b2a8ce59c3"
21  },
22  "id": "Mark",
23  "fullname": "Mark Phillips",
24  "email": "mphilips12@shaw.ca",
25  "city": "San Francisco"
26 },
27 {
28  "_id": {
29    "$oid": "660251b4fe2aefa3417851c7"
30  },
31  "id": "Fernanda",
32  "fullname": "Fernanda Ramos",
33  "email": "fernadaramos4@uol.com.br",
34  "age": 24
35 }
```

2. Buatlah dan Jalankan perintah untuk mencari data dari collection Film untuk menampilkan data Title yang diakhiri dengan huruf G.

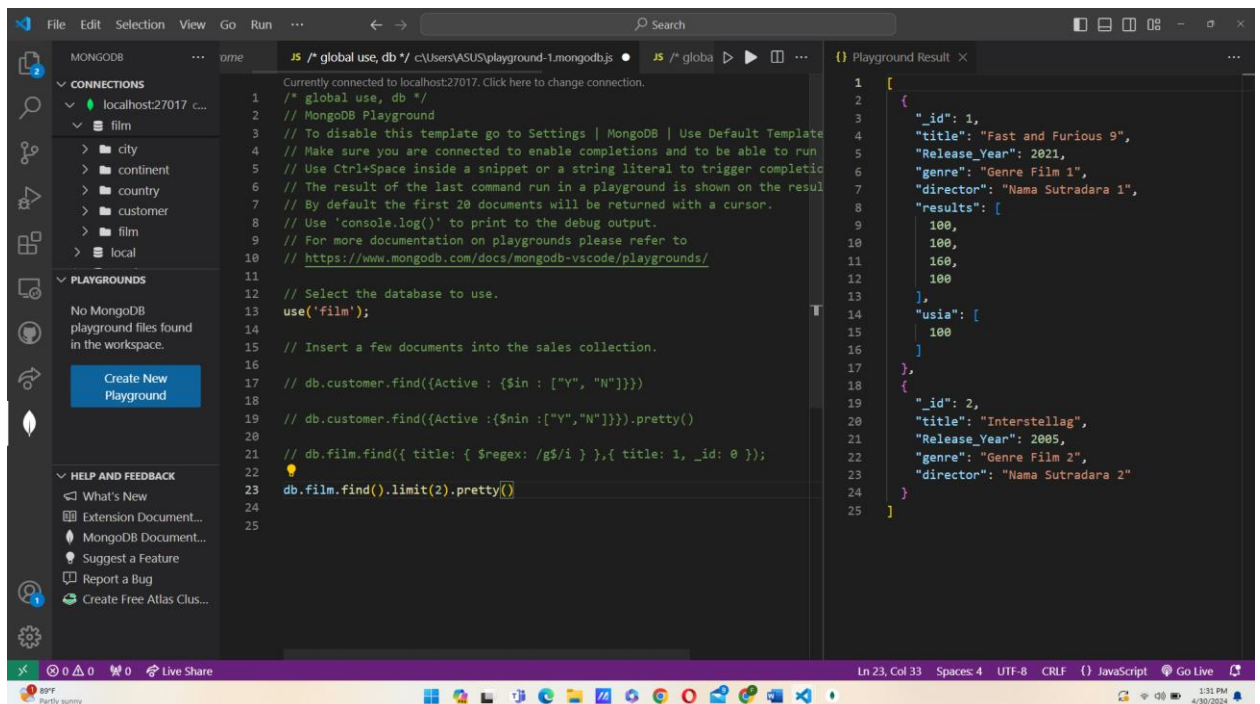


The screenshot shows the MongoDB Playground interface in VS Code. The left sidebar displays the 'CONNECTIONS' and 'PLAYGROUNDS' sections. The main editor area contains a JavaScript script for connecting to the database and querying the 'film' collection. The 'PLAYGROUND RESULT' pane on the right shows the output of the query, which is a single document with the title 'Interstellag'.

```
1 /* global use, db */
2 // MongoDB Playground
3 // To disable this template go to Settings | MongoDB | Use Default Template
4 // Make sure you are connected to enable completions and to be able to run
5 // Use Ctrl+Space inside a snippet or a string literal to trigger completions
6 // The result of the last command run in a playground is shown on the results pane
7 // By default the first 20 documents will be returned with a cursor.
8 // Use 'console.log()' to print to the debug output.
9 // For more documentation on playgrounds please refer to
10 // https://www.mongodb.com/docs/mongodb-vscode/playgrounds/
11
12 // Select the database to use.
13 use('film');
14
15 // Insert a few documents into the sales collection.
16
17 // db.customer.find({Active : {$in : ["Y", "N"]}})
18
19 // db.customer.find({Active : {$in : ["Y", "N"]}}).pretty()
20
21 db.film.find({ title: { $regex: /g$/i } }, { title: 1, _id: 0 });
22
23
```

```
1 [
2   {
3     "title": "Interstellag"
4   }
5 ]
```

3. Buatlah dan jalankan perintah untuk menampilkan 2 buah data dari collection Customer.

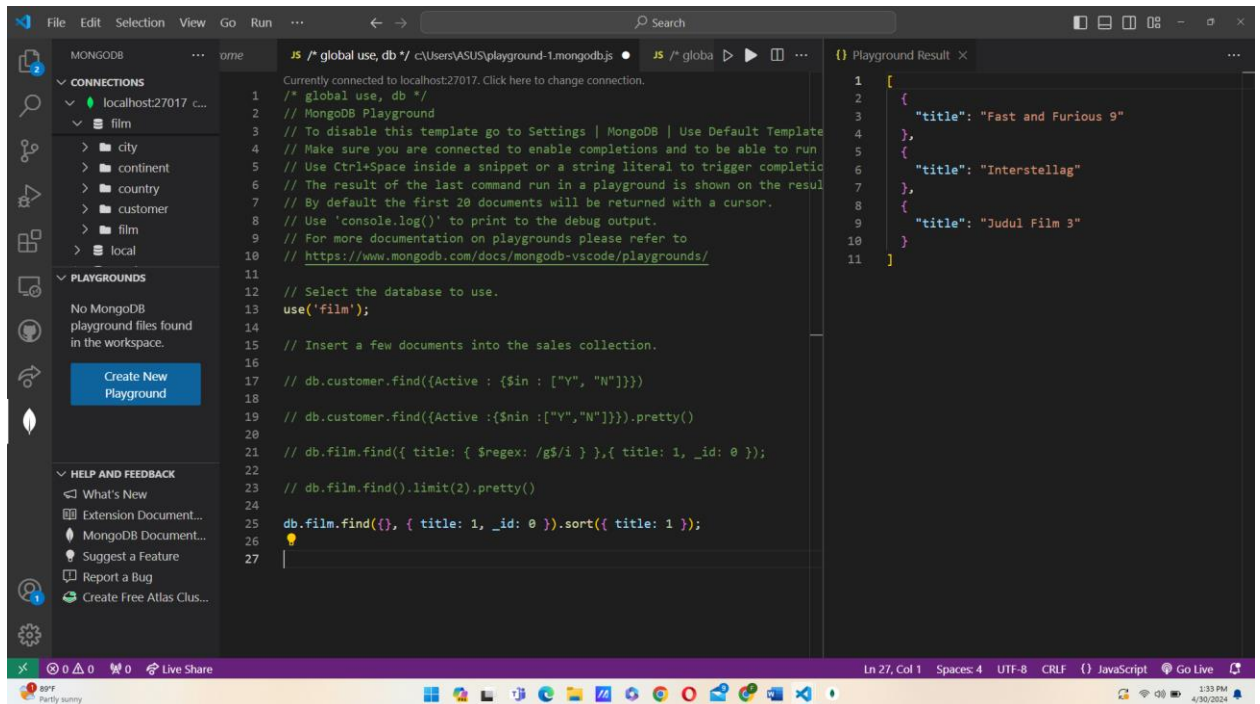


The screenshot shows the MongoDB Playground interface in VS Code. The left sidebar displays the 'CONNECTIONS' and 'PLAYGROUNDS' sections. The main editor area contains a JavaScript script for connecting to the database and querying the 'film' collection. The 'PLAYGROUND RESULT' pane on the right shows the output of the query, which is a single document with the title 'Interstellag'.

```
1 /* global use, db */
2 // MongoDB Playground
3 // To disable this template go to Settings | MongoDB | Use Default Template
4 // Make sure you are connected to enable completions and to be able to run
5 // Use Ctrl+Space inside a snippet or a string literal to trigger completions
6 // The result of the last command run in a playground is shown on the results pane
7 // By default the first 20 documents will be returned with a cursor.
8 // Use 'console.log()' to print to the debug output.
9 // For more documentation on playgrounds please refer to
10 // https://www.mongodb.com/docs/mongodb-vscode/playgrounds/
11
12 // Select the database to use.
13 use('film');
14
15 // Insert a few documents into the sales collection.
16
17 // db.customer.find({Active : {$in : ["Y", "N"]}})
18
19 // db.customer.find({Active : {$in : ["Y", "N"]}}).pretty()
20
21 // db.film.find({ title: { $regex: /g$/i } }, { title: 1, _id: 0 });
22
23 db.film.find().limit(2).pretty()
24
25
```

```
1 [
2   {
3     "_id": 1,
4     "title": "Fast and Furious 9",
5     "Release_Year": 2021,
6     "genre": "Genre Film 1",
7     "director": "Nama Sutradara 1",
8     "results": [
9       100,
10      100,
11      160,
12      100
13     ],
14     "usia": [
15       100
16     ]
17   },
18   {
19     "_id": 2,
20     "title": "Interstellag",
21     "Release_Year": 2005,
22     "genre": "Genre Film 2",
23     "director": "Nama Sutradara 2"
24   }
25 ]
```

4. Buatlah dan jalankan perintah untuk menampilkan semua data di film berdasarkan urutan alfabet.



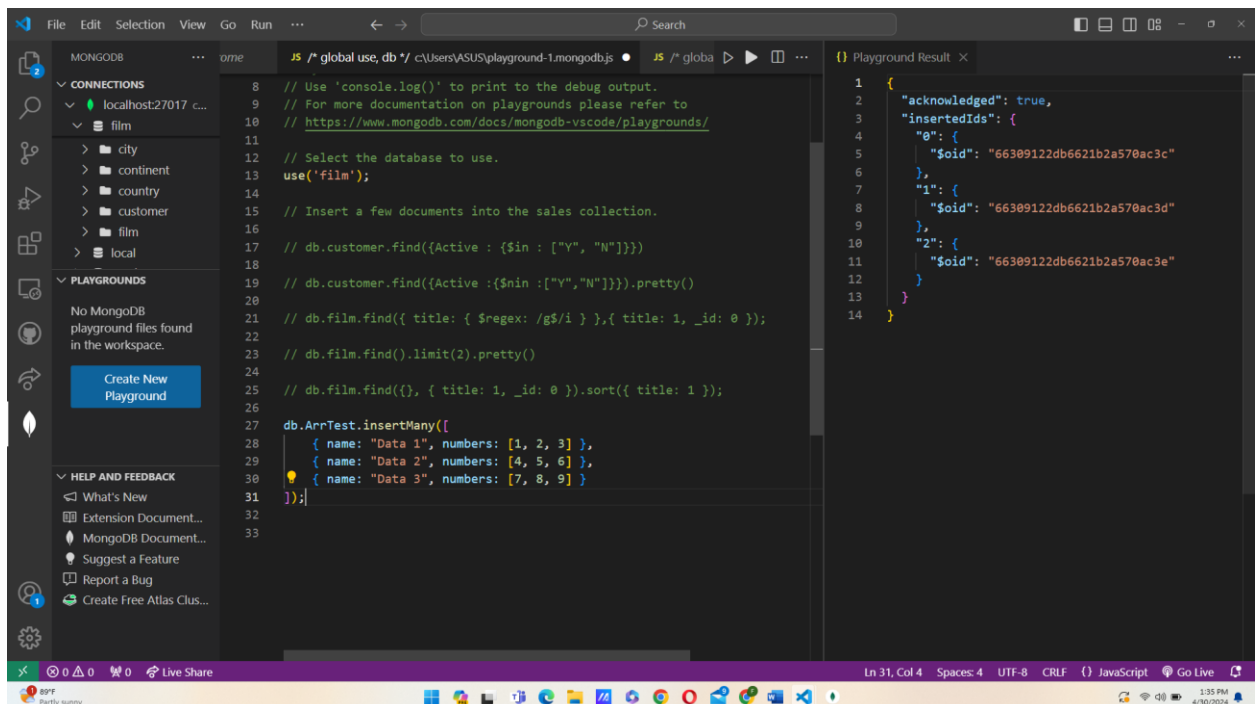
The screenshot shows the MongoDB Playground interface in VS Code. The left sidebar displays the 'film' collection under the 'localhost:27017' database. The main editor contains a JavaScript script with the following code:

```
1 /* global use, db */
2 // MongoDB Playground
3 // To disable this template go to Settings | MongoDB | Use Default Template
4 // Make sure you are connected to enable completions and to be able to run
5 // Use Ctrl+Space inside a snippet or a string literal to trigger completio
6 // The result of the last command run in a playground is shown on the resul
7 // By default the first 20 documents will be returned with a cursor.
8 // Use 'console.log()' to print to the debug output.
9 // For more documentation on playgrounds please refer to
10 // https://www.mongodb.com/docs/mongodb-vscode/playgrounds/
11
12 // Select the database to use.
13 use('film');
14
15 // Insert a few documents into the sales collection.
16
17 // db.customer.find({Active : {$in : ["Y", "N"]}})
18
19 // db.customer.find({Active : {$nin : ["Y", "N"]}}).pretty()
20
21 // db.film.find({ title: { $regex: /g$/i } }, { title: 1, _id: 0 });
22
23 // db.film.find().limit(2).pretty()
24
25 db.film.find({}, { title: 1, _id: 0 }).sort({ title: 1 });
26
27
```

The 'Playground Result' pane on the right shows the output of the query:

```
1 [
2   {
3     "title": "Fast and Furious 9"
4   },
5   {
6     "title": "Interstellar"
7   },
8   {
9     "title": "Judul Film 3"
10  }
11 ]
```

5. Buatlah collection baru dengan nama ArrTest masukan beberapa data yang memiliki setidaknya 1 array kemudian buat dan jalankan perintah Push, Pull dan addToSet secara bergiliran.



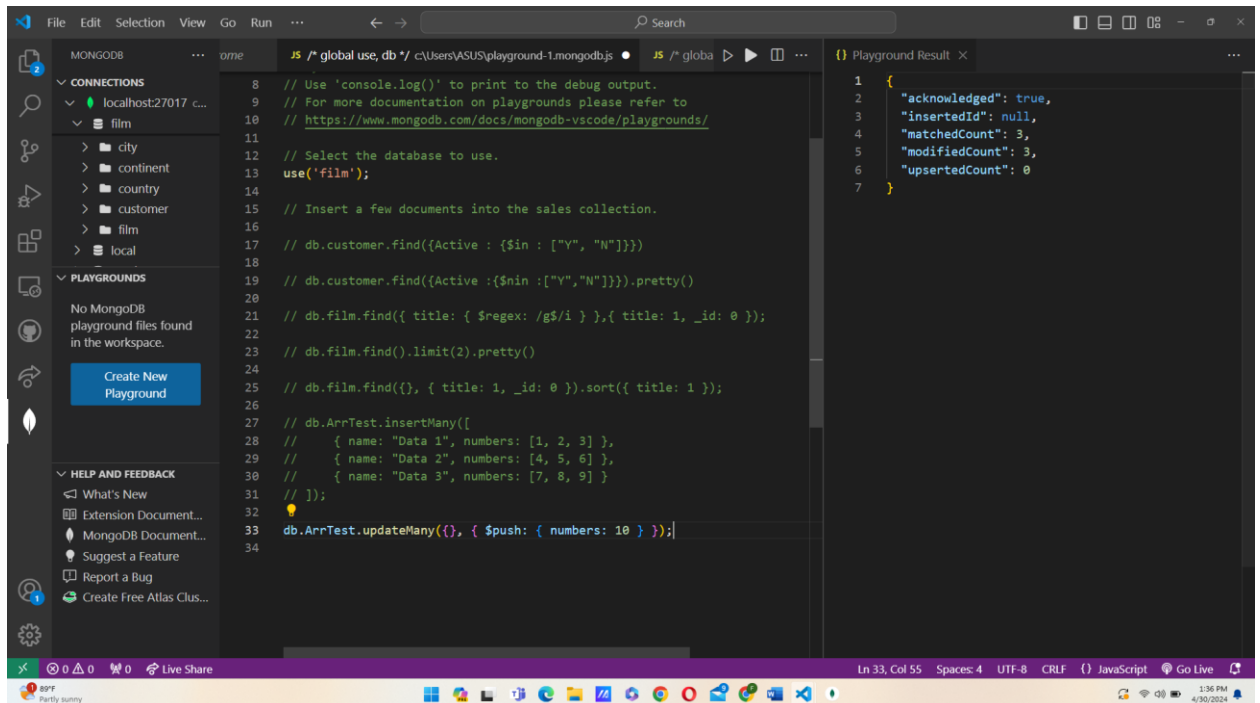
The screenshot shows the MongoDB Playground interface in VS Code. The left sidebar displays the 'ArrTest' collection under the 'localhost:27017' database. The main editor contains a JavaScript script with the following code:

```
8 // Use 'console.log()' to print to the debug output.
9 // For more documentation on playgrounds please refer to
10 // https://www.mongodb.com/docs/mongodb-vscode/playgrounds/
11
12 // Select the database to use.
13 use('film');
14
15 // Insert a few documents into the sales collection.
16
17 // db.customer.find({Active : {$in : ["Y", "N"]}})
18
19 // db.customer.find({Active : {$nin : ["Y", "N"]}}).pretty()
20
21 // db.film.find({ title: { $regex: /g$/i } }, { title: 1, _id: 0 });
22
23 // db.film.find().limit(2).pretty()
24
25 // db.film.find({}, { title: 1, _id: 0 }).sort({ title: 1 });
26
27
28 db.ArrTest.insertMany([
29   { name: "Data 1", numbers: [1, 2, 3] },
30   { name: "Data 2", numbers: [4, 5, 6] },
31   { name: "Data 3", numbers: [7, 8, 9] }
32 ]);
33
```

The 'Playground Result' pane on the right shows the output of the insertMany command:

```
1 {
2   "acknowledged": true,
3   "insertedIds": {
4     "0": {
5       "$oid": "66309122db6621b2a570ac3c"
6     },
7     "1": {
8       "$oid": "66309122db6621b2a570ac3d"
9     },
10    "2": {
11      "$oid": "66309122db6621b2a570ac3e"
12    }
13  }
14 }
```

```
db.ArrTest.updateMany({}, { $push: { numbers: 10 } });
```



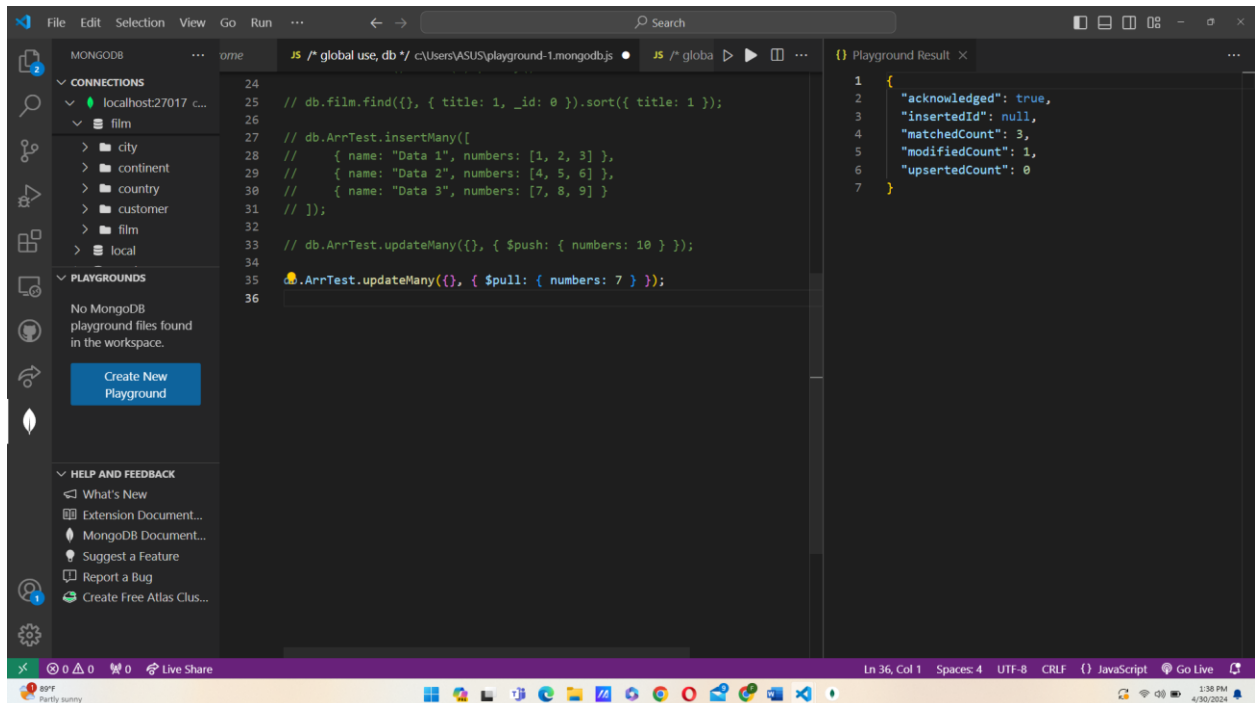
The screenshot shows the VS Code interface with a MongoDB playground script. The left sidebar shows the 'CONNECTIONS' panel with 'localhost:27017 c...' and a 'film' collection. The main editor shows a JavaScript script with the following code:

```
8 // Use 'console.log()' to print to the debug output.
9 // For more documentation on playgrounds please refer to
10 // https://www.mongodb.com/docs/mongodb-vscode/playgrounds/
11
12 // Select the database to use.
13 use('film');
14
15 // Insert a few documents into the sales collection.
16 // db.customer.find({Active : {$in : ["Y", "N"]}})
17 // db.customer.find({Active : {$in : ["Y", "N"]}}).pretty()
18
19 // db.film.find({ title: {$regex: /g$/i }},{ title: 1, _id: 0 });
20 // db.film.find().limit(2).pretty()
21
22 // db.film.find({}, { title: 1, _id: 0 }).sort({ title: 1 });
23
24 // db.ArrTest.insertMany([
25 //   { name: "Data 1", numbers: [1, 2, 3] },
26 //   { name: "Data 2", numbers: [4, 5, 6] },
27 //   { name: "Data 3", numbers: [7, 8, 9] }
28 // ]);
29
30 db.ArrTest.updateMany({}, { $push: { numbers: 10 } });
```

The right sidebar shows the 'Playground Result' panel with the following JSON output:

```
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 3,
5   "modifiedCount": 3,
6   "upsertedCount": 0
7 }
```

```
db.ArrTest.updateMany({}, { $pull: { numbers: 7 } });
```



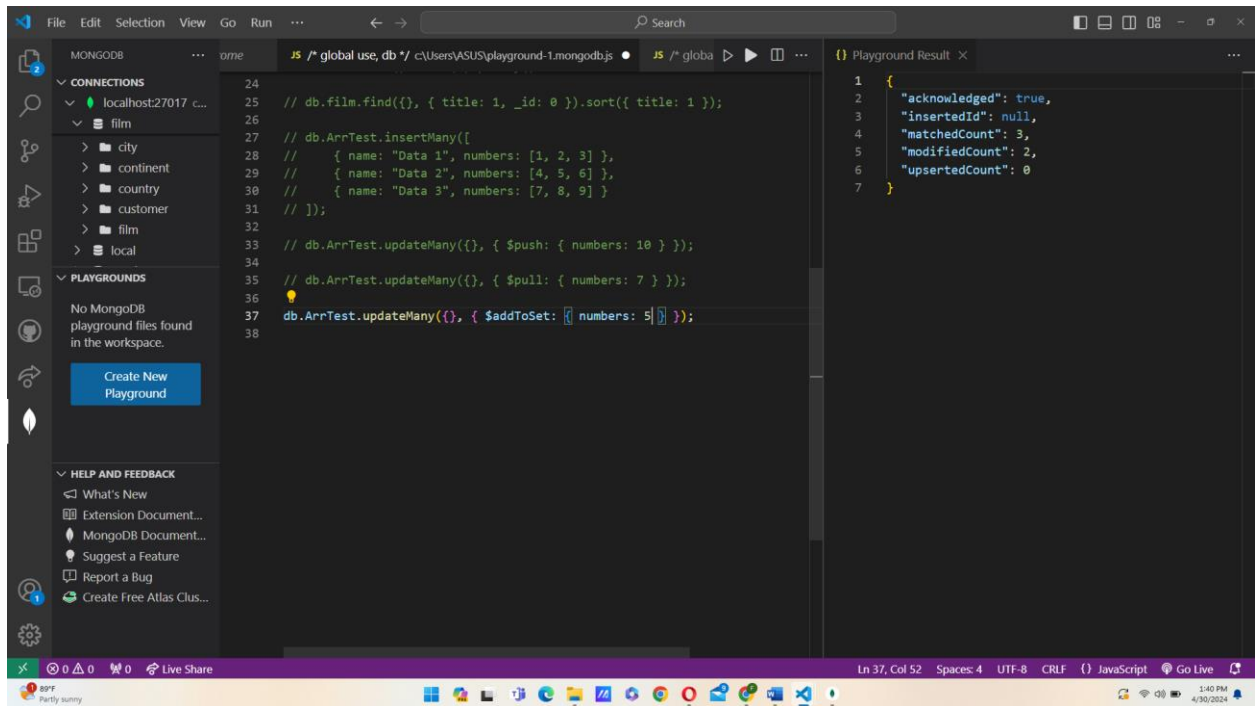
The screenshot shows the VS Code interface with a MongoDB playground script. The left sidebar shows the 'CONNECTIONS' panel with 'localhost:27017 c...' and a 'film' collection. The main editor shows a JavaScript script with the following code:

```
24 // db.film.find({}, { title: 1, _id: 0 }).sort({ title: 1 });
25
26 // db.ArrTest.insertMany([
27 //   { name: "Data 1", numbers: [1, 2, 3] },
28 //   { name: "Data 2", numbers: [4, 5, 6] },
29 //   { name: "Data 3", numbers: [7, 8, 9] }
30 // ]);
31
32 // db.ArrTest.updateMany({}, { $push: { numbers: 10 } });
33
34 // db.ArrTest.updateMany({}, { $pull: { numbers: 7 } });
```

The right sidebar shows the 'Playground Result' panel with the following JSON output:

```
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 3,
5   "modifiedCount": 1,
6   "upsertedCount": 0
7 }
```

```
db.ArrTest.updateMany({}, { $addToSet: { numbers: 5 } });
```



Playground Result

```
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 3,
5   "modifiedCount": 2,
6   "upsertedCount": 0
7 }
```

Ln 37, Col 52 Spaces: 4 UTF-8 CRLF () JavaScript Go Live