HCS Challenge

Daniel Becerra – Software Engineer

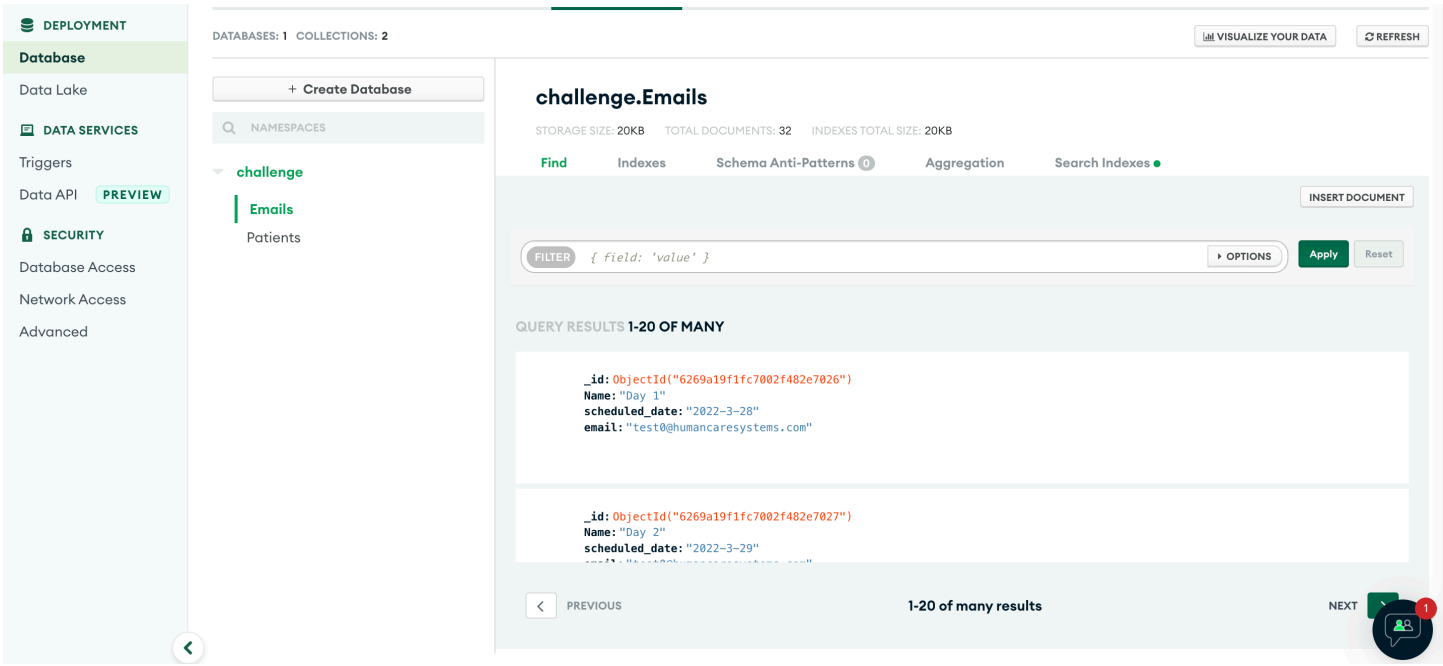
Populating the data

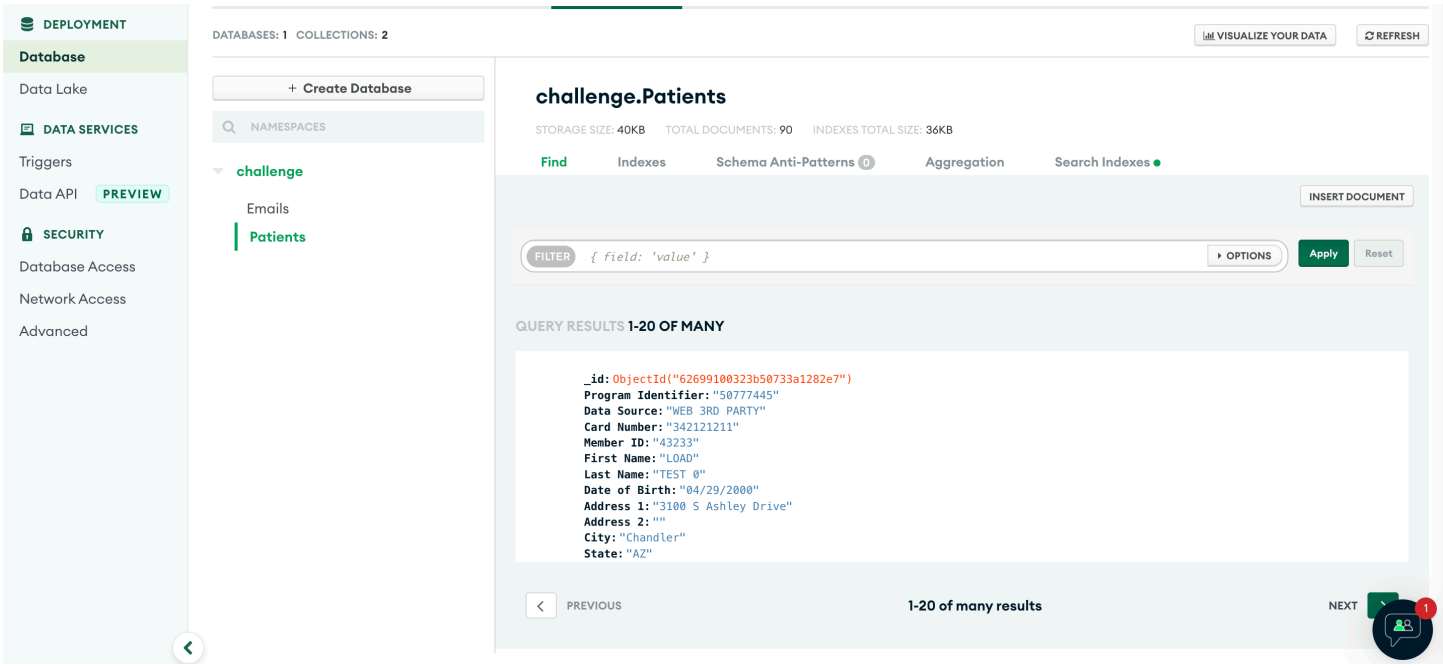
The info is filled out using a file reader plus a csv parser called csv-parser

```
const mongodb = require('mongodb');
const MongoClient = mongodb.MongoClient;
const url = 'mongodb+srv://dajobeuz:eiB3bZqI2bclH1Pd@hcs.tcgbp.mongodb.net/myFirstDatabase?retryWrites=true&w=majority';
const csv = require('csv-parser');
const fs = require('fs');

let patientsData = [];
let emails = [];

fs.createReadStream('patients.csv')
    .pipe(csv({ separator: '|' }))
    .on('data', (data) => patientsData.push(data))
    .on('end', () => {
        let curDate = new Date();
        patientsData.filter(patient => { return patient["CONSENT"] === "Y"}).forEach(filteredPatient => {
            emails.push({
                'Name': "Day 1",
                'scheduled_date': curDate.getFullYear() + '-' + curDate.getMonth() + '-' + (curDate.getDate() + 1),
                'email': filteredPatient['Email Address']
            },
            {
                'Name': "Day 2",
                'scheduled_date': curDate.getFullYear() + '-' + curDate.getMonth() + '-' + (curDate.getDate() + 2),
                'email': filteredPatient['Email Address']
            },
            {
                'Name': "Day 3",
                'scheduled_date': curDate.getFullYear() + '-' + curDate.getMonth() + '-' + (curDate.getDate() + 3),
                'email': filteredPatient['Email Address']
            },
            {
                    'Name': "Day 4",
                    'scheduled_date': curDate.getFullYear() + '-' + curDate.getMonth() + '-' + (curDate.getDate() + 4),
                    'email': filteredPatient['Email Address']
            });
        });

        //loading patient data
```

```
//loading patient data
MongoClient.connect(url).then(result => {
    var dbo = result.db("challenge");
    dbo.collection("Patients").insertMany(patientsData, (err, res) => {
      if (err) throw err;
      console.log(res.insertedCount);
      result.close();
    });
}).catch(err => {
    console.log(err);
});

//loading email data
MongoClient.connect(url).then(result => {
    var dbObject = result.db("challenge");
    dbObject.collection("Emails").insertMany(emails, (err, res) => {
        if (err) throw err;
        console.log(res.insertedCount);
        result.close();
    });

}).catch(err => {
    console.log(err)
});
```

Here’s the mongodb view





And here's the automation process

```
danielbecerra@MacBook-Pro hcs_challenge % npm run automation

> hcs_challenge@1.0.0 automation
> node printing.js && npm run test


Priting ID with no email but consent is Y:  62699100323b50733a1282f2, 62699151fcbf3f3f9c0b6f3e, 626991af15b70935affd1f6f, 626992e596b514b5873cc1ce, 6269933c5c570536e89b9251
Print ID were first name is empty 62699100323b50733a1282ee, 62699100323b50733a1282f1, 62699151fcbf3f3f9c0b6f3a, 62699151fcbf3f3f9c0b6f3d, 626991af15b70935affd1f6b, 626991af15b70935affd1f6e, 626992e596b514b5873cc1ca, 626992e596b514b5873cc1cd, 6269933c5c570536e89b924d, 6269933c5c570536e89b9250

> hcs_challenge@1.0.0 test
> jest

 PASS  ./test.spec.js
  Automaton and Test Suite
    ✓ Verify the data in flat file matches the data in Patients collection. (2 ms)
    ✓ Verify Emails were created in Emails Collection for patients who have CONSENT as Y. (10 ms)
    ✓ Verify emails for each patient are scheduled correctly. (3 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        0.673 s, estimated 1 s
Ran all test suites.
```