Big Data

Aggregations with MongoDB

von
Benjamin Ellmer (S2210455012)

Mobile Computing Master
FH Hagenberg

April 26, 2023

Step 1

Install and start MongoDB using docker compose:

```
git clone https://github.com/Digital-Media/fhooe-mongo-dock
   .git
docker compose -f fhooe-mongo-dock/docker-compose.yml up -d
```

Connect to the MongoDB shell in the container:

```
docker exec -it mongodb /bin/bash -c "mongosh"
```

Show and create database:

```
show databases
use onlineshop
```

Insert Data from exercise description:

```
db.orders.insertMany([
    ...
]);
```

Find all rows of collection orders:

```
db.orders.find({})
```

Step 2

Step 3

Get sum of total sums:

```
onlineshop> db.orders.aggregate([
    {
        $group: {
            _id: null,
            totalSum: { $sum: "$total_sum" }
        }
    }
])
```

Output:

```
[ { _id: null, totalSum: 435 } ]
```

Get sum of qty * price per user:

```
db.orders.aggregate([
    { $unwind: "$items" },
    { $group: {
        _id: "$user_id",
        sum:{ $sum: { $multiply:
            [ "$items.price", "$items.qty"]
        }}
    }}
])
```

Output:

```
[
  { _id: 2, sum: 125 },
  { _id: 4, sum: 155 },
  { _id: 1, sum: 95 },
  { _id: 3, sum: 60 }
]
```

Get sum of qty * price per user using mapReduce:

```
db.orders.mapReduce(
    function() { this.items.forEach(
        item => emit(this.user_id, item.qty * item.price)
    );},
    function(key, values) { return Array.sum(values); },
    { out: { inline: 1 },
      finalize: function(key, reducedValue) {
        return { _id: key, sum: reducedValue };
      }
    }
)
```

Output:

```
{
  results: [
    { _id: 1, value: { _id: 1, sum: 95 } },
    { _id: 4, value: { _id: 4, sum: 155 } },
    { _id: 2, value: { _id: 2, sum: 125 } },
    { _id: 3, value: { _id: 3, sum: 60 } }
  ],
  ok: 1
}
```

Step 4

Delete all data in the orders collection:

```
db.orders.deleteMany({})
```

Drop the onlineshop database

```
db.dropDatabase()
```

Appendix