

THIRD PARTY LOGISTICS DATABASE

Milestone: Implementation in NoSQL

Group 05

Deepak Kumar Meena

Saravanan Arumugam

857 415 1106

646 251 2318

meena.d@northeastern.edu

arumugam.sa@northeastern.edu

Percentage of effort contributed by Student 1: 50

Percentage of effort contributed by Student 2: 50

Signature of Student 1: Deepak Kumar Meena

Signature of Student 2: Saravanan Arumugam

Submission Date: December 03, 2022

We have implemented the relational model into NoSQL in MongoDB using the online playground.

Queries to develop the collections:

```
db.createCollection("invoice");
```

```
db.createCollection("supplied");
```

Inserting Data:

```
db.invoice.insert({"invoice_id": 1903814011, "inv_date": new Date("2022-08-22"), "supplier_id": 29991, "customer_id": 328});
```

```
db.invoice.insert({"invoice_id": 1163439593, "inv_date": new Date("2021-06-19"), "supplier_id": 49115, "customer_id": 550});
```

```
db.invoice.insert({"invoice_id": 1172642985, "inv_date": new Date("2020-06-21"), "supplier_id": 11506, "customer_id": 120});
```

```
db.invoice.insert({"invoice_id": 1183037463, "inv_date": new Date("2020-05-04"), "supplier_id": 14453, "customer_id": 138});
```

```
db.invoice.insert({"invoice_id": 1233379593, "inv_date": new Date("2020-02-14"), "supplier_id": 29991, "customer_id": 327});
```

```
db.invoice.insert({"invoice_id": 1294545282, "inv_date": new Date("2021-10-17"), "supplier_id": 57675, "customer_id": 67});
```

```
db.invoice.insert({"invoice_id": 1333059457, "inv_date": new Date("2021-05-13"), "supplier_id": 44743, "customer_id": 5});
```

```
db.invoice.insert({"invoice_id": 1565562210, "inv_date": new Date("2020-12-23"), "supplier_id": 33740, "customer_id": 486});
```

```
db.invoice.insert({"invoice_id": 1578826120, "inv_date": new Date("2020-03-18"), "supplier_id": 71873, "customer_id": 806});
```

```
db.invoice.insert({"invoice_id": 1600543803, "inv_date": new Date("2021-01-07"), "supplier_id": 35018, "customer_id": 4});
```

```
db.supplied.insert({"part_nr": 10215229, "invoice_nr": 1093814011, "price": 2270212, "quantity": 18});
```

```
db.supplied.insert({"part_nr": 10311654, "invoice_nr": 1163439593, "price": 7549900, "quantity": 69});
```

```
db.supplied.insert({"part_nr": 10480956, "invoice_nr": 1172642985, "price": 6899619, "quantity": 871});
```

```
db.supplied.insert({"part_nr": 10592946, "invoice_nr": 1183037463, "price": 6718674, "quantity": 425});
```

IE 6700 Data Management for Analytics

```
db.supplied.insert({"part_nr": 10862492, "invoice_nr": 1233379593, "price": 5204500, "quantity": 359});  
db.supplied.insert({"part_nr": 10989528, "invoice_nr": 1294545282, "price": 427013, "quantity": 545});  
db.supplied.insert({"part_nr": 11004833, "invoice_nr": 1333059457, "price": 5071867, "quantity": 23});  
db.supplied.insert({"part_nr": 11174366, "invoice_nr": 1565562210, "price": 5824582, "quantity": 663});  
db.supplied.insert({"part_nr": 11478025, "invoice_nr": 1578826120, "price": 6858154, "quantity": 474});  
db.supplied.insert({"part_nr": 11494064, "invoice_nr": 1600543803, "price": 1678411, "quantity": 886});
```

```
db.vehicle.insert([  
  {"_id": 29, "registration_nr": 500371, "capacity": 6, "year": 2010, "age": 12, "vendor_id": 91},  
  {"_id": 64, "registration_nr": 49, "capacity": 6, "year": 2010, "age": 12, "vendor_id": 509},  
  {"_id": 71, "registration_nr": 56380228, "capacity": 3, "year": 2010, "age": 12, "vendor_id": 449},  
  {"_id": 33, "registration_nr": 149803032, "capacity": 6, "year": 2016, "age": 6, "vendor_id": 292},  
  {"_id": 43, "registration_nr": 91394167, "capacity": 12, "year": 2016, "age": 6, "vendor_id": 445},  
  {"_id": 54, "registration_nr": 3423519, "capacity": 1, "year": 2016, "age": 6, "vendor_id": 239},  
  {"_id": 98, "registration_nr": 912699374, "capacity": 8, "year": 2016, "age": 6, "vendor_id": 587},  
  {"_id": 75, "registration_nr": 3753043, "capacity": 6, "year": 2016, "age": 6, "vendor_id": 178},  
  {"_id": 89, "registration_nr": 821477, "capacity": 2, "year": 2020, "age": 2, "vendor_id": 32},  
  {"_id": 44, "registration_nr": 445, "capacity": 10, "year": 2020, "age": 2, "vendor_id": 882}])
```

The screenshot shows a web browser window with the URL https://www.pdbmbook.com/playground/mongo/wine/view/pgdb_1670105277_638bc8d339ff. The page title is "Principles of Database Management". The main content area is titled "Using: wine (mongo)" and contains a text input field with a JavaScript bulk insert statement. Below the input field is a "Run" button. To the right of the input field are links for "Back to Playground overview", "Reset", and "Tips". The "Reset" link has a tooltip that says "Click here to reset the database to its initial state (all your changes will be lost)." The "Tips" section contains instructions on how to use MongoDB shell commands. Below the "Run" button is a "Result" section showing the output of the bulk insert operation. The output is a JSON object indicating that all documents were successfully inserted.

```
db.vehicle.insert([  
  {"_id": 29, "registration_nr": 500371, "capacity": 6, "year": 2010, "age": 12, "vendor_id": 91},  
  {"_id": 64, "registration_nr": 49, "capacity": 6, "year": 2010, "age": 12, "vendor_id": 509},  
  {"_id": 71, "registration_nr": 56380228, "capacity": 3, "year": 2010, "age": 12, "vendor_id": 449},  
  {"_id": 33, "registration_nr": 149803032, "capacity": 6, "year": 2016, "age": 6, "vendor_id": 292},  
  {"_id": 43, "registration_nr": 91394167, "capacity": 12, "year": 2016, "age": 6, "vendor_id": 445},  
  {"_id": 54, "registration_nr": 3423519, "capacity": 1, "year": 2016, "age": 6, "vendor_id": 239},  
  {"_id": 98, "registration_nr": 912699374, "capacity": 8, "year": 2016, "age": 6, "vendor_id": 587},  
  {"_id": 75, "registration_nr": 3753043, "capacity": 6, "year": 2016, "age": 6, "vendor_id": 178},  
  {"_id": 89, "registration_nr": 821477, "capacity": 2, "year": 2020, "age": 2, "vendor_id": 32},  
  {"_id": 44, "registration_nr": 445, "capacity": 10, "year": 2020, "age": 2, "vendor_id": 882}])
```

Run

Result

```
BulkWriteResult({  
  "writeErrors": [],  
  "writeConcernErrors": [],  
  "nInserted": 10,  
  "nUpserted": 0,  
  "nMatched": 0,  
  "nModified": 0,  
  "nRemoved": 0,  
  "upserted": []  
})
```

© Principles of Database Management 2022.

Querying:

To find vehicles that are 6 years old or with the capacity of 3.

```
db.vehicle.find({$or: [{"age":6}, {"capacity":3} ] })
```

Run

Result

```
{ "_id" : 71, "registration_nr" : 56380228, "capacity" : 3, "year" : 2010, "age" : 12, "vendor_id" : 449 }
{ "_id" : 33, "registration_nr" : 149803032, "capacity" : 6, "year" : 2016, "age" : 6, "vendor_id" : 292 }
{ "_id" : 43, "registration_nr" : 91394167, "capacity" : 12, "year" : 2016, "age" : 6, "vendor_id" : 445 }
{ "_id" : 54, "registration_nr" : 3423519, "capacity" : 1, "year" : 2016, "age" : 6, "vendor_id" : 239 }
{ "_id" : 98, "registration_nr" : 912699374, "capacity" : 8, "year" : 2016, "age" : 6, "vendor_id" : 587 }
{ "_id" : 75, "registration_nr" : 3753043, "capacity" : 6, "year" : 2016, "age" : 6, "vendor_id" : 178 }
```

To find the vehicles with capacity more than 10

```
db.vehicle.find({"capacity": {$gt:10}})
```

Result

```
{ "_id" : 43, "registration_nr" : 91394167, "capacity" : 12, "year" : 2016, "age" : 6, "vendor_id" : 445 }
```

To find the total capacity of vehicles in each age

```
db.vehicle.aggregate(
[
{ $group: { _id: "$age", "total capacity": { $sum: "$capacity" } } },
{ $sort: { total: 1 } }
])
```

Run

Result

```
{ "_id" : 6, "total capacity" : 33 }
{ "_id" : 2, "total capacity" : 12 }
{ "_id" : 12, "total capacity" : 15 }
```