

Part 1:

Connecting to the DB server

```
mongosh mongodb://localho × + ▾
Please enter a MongoDB connection string (Default: mongodb://localhost/): mo
ngodb://localhost:27017
mongodb://localhost:27017
Current Mongosh Log ID: 6631ad105f31c3bc0546b798
Connecting to:      mongodb://localhost:27017/?directConnection=true&ser
verSelectionTimeoutMS=2000&appName=mongosh+2.2.5
Using MongoDB:      7.0.9
Using Mongosh:      2.2.5
```

Loading of the .js file:

```
customerTransactions> load("C:/Users/johns/Desktop/GMU/Spring 2024/550/Project 3/customerTransactions.js")
true
customerTransactions> |
```

Product table count:

```
customerTransactions> db.product.count();
32
customerTransactions> |
```

Customer table count:

```
customerTransactions> db.customer.count();
10
customerTransactions> |
```

Transactions table count:

```
customerTransactions> db.transactions.count();
0
customerTransactions> |
```

Transactions_contains table count:

```
customerTransactions> db.transactions_contains.count();
0
customerTransactions> |
```

```
customerTransactions> db.customer.aggregate({$unwind: "$course"},{$group: {"_id":null,tcount: {$sum:1}}});
```

Part 2:

1.

```
customerTransactions> db.product.find({ category: 'Alcohol' }, {upc: 1, _id: 0});
[
  { upc: '10029' },
  { upc: '10030' },
  { upc: '10031' },
  { upc: '10032' }
]
customerTransactions> |
```

2.

```
customerTransactions> db.customer.find({ "transaction.transaction_ID": '104' }, { customer_ID: 1, _id: 0 });
[ { customer_ID: '10' } ]
customerTransactions> |
```

3.

```
customerTransactions> db.customer.find({ "transaction.transaction_ID": '104' }, { customer_ID: 1, _id: 0 });
[ { customer_ID: '10' } ]
customerTransactions> db.customer.aggregate([
...   { $match: { "transaction.0": { $exists: true } } }, // Filters customers with at least one transaction
...   { $project: { _id: 1, first_transaction_total: { $arrayElemAt: ["$transaction.total", 0] } } }, // Projects the total of the first transaction
...   { $match: { first_transaction_total: { $lt: 50 } } }, // Filters transactions with total less than $50
...   { $count: "customers_with_first_transaction_lt_50" } // Counts the number of customers
... ])
[ { customers_with_first_transaction_lt_50: 3 } ]
customerTransactions> |
```

4.

```
customerTransactions> db.product.find({ product_Description: { $exists: false } }, { product_name: 1, _id: 0 });
[ { product_name: 'Rice-A-Roni Cajun' } ]
customerTransactions> |
```

5.

```
customerTransactions> db.product.count({ brand: { $in: ['Pepsi', 'Coca-Cola'] } })
3
customerTransactions>
```

6.

```
customerTransactions> db.product.find({ marked_price: { $lt: 10 }, quantity:
  { $gte: 50 } }, { product_name: 1, _id: 0 });
[
  { product_name: 'Coca-Cola 2L' },
  { product_name: 'Pepsi 2L' },
  { product_name: 'Honed Glazed Ham' },
  { product_name: 'Hot Dog Buns' }
]
customerTransactions> |
```

7.

```
customerTransactions> db.customer.count({ transaction: { $exists: false } })
;
0
customerTransactions> |
```

8.

```
customerTransactions> db.customer.aggregate([
...   { $unwind: "$transaction" }, // Unwind the transaction array
...   { $lookup: { from: "transaction_contains", localField: "transaction.transaction_ID",
transaction_contains", foreignField: "_id", as: "contains" } }, // Look up the
transaction_contains collection
...   { $match: { "contains.upc": { $in: db.product.distinct("upc", { category: "Alcohol" }) } } }, // Match transactions containing alcoholic beverages
...   { $project: { "transaction.transaction_ID": 1, _id: 0 } } // Project only the transaction_ID
... ])
customerTransactions> |
```