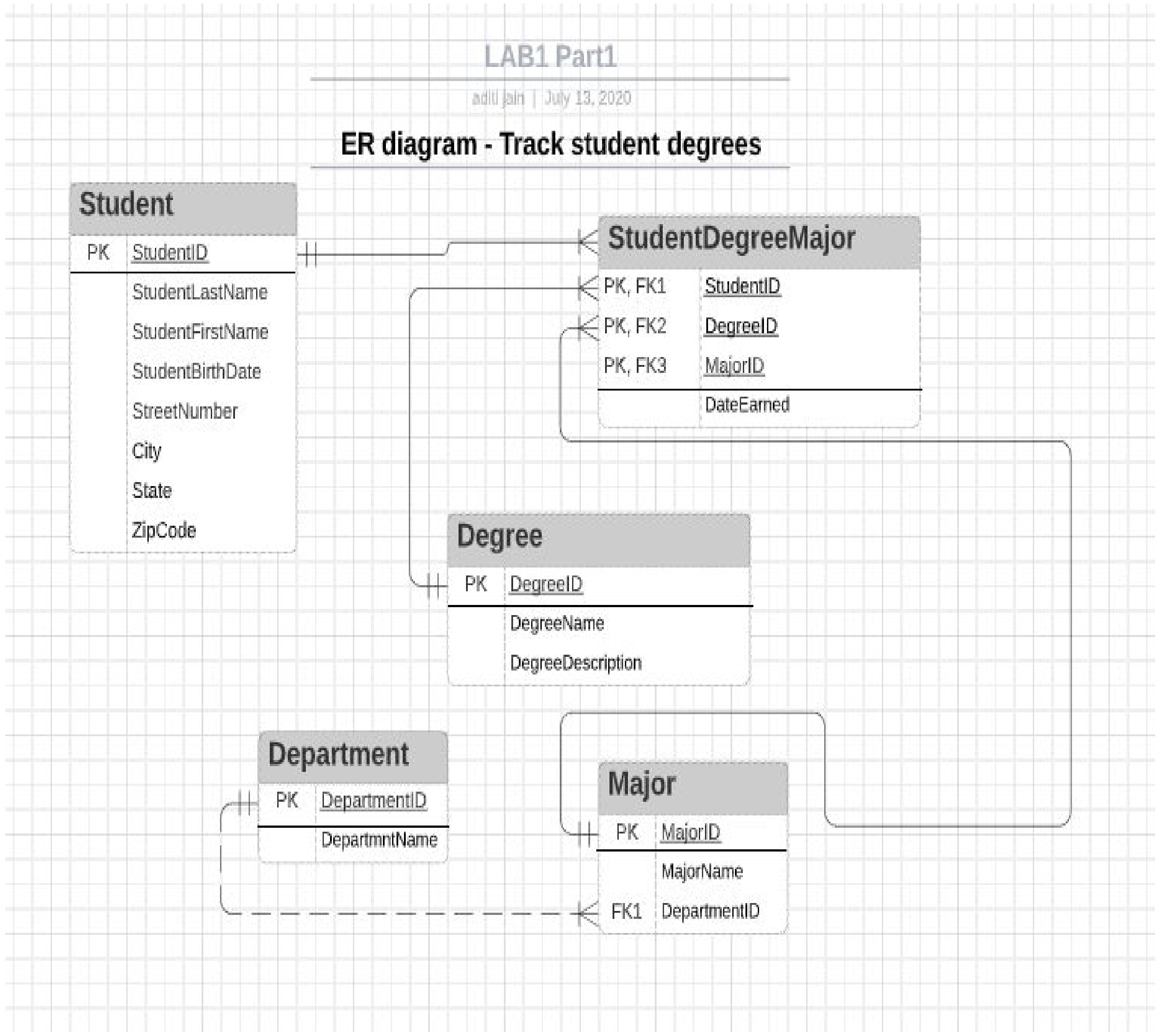
LAB 1 Solutions

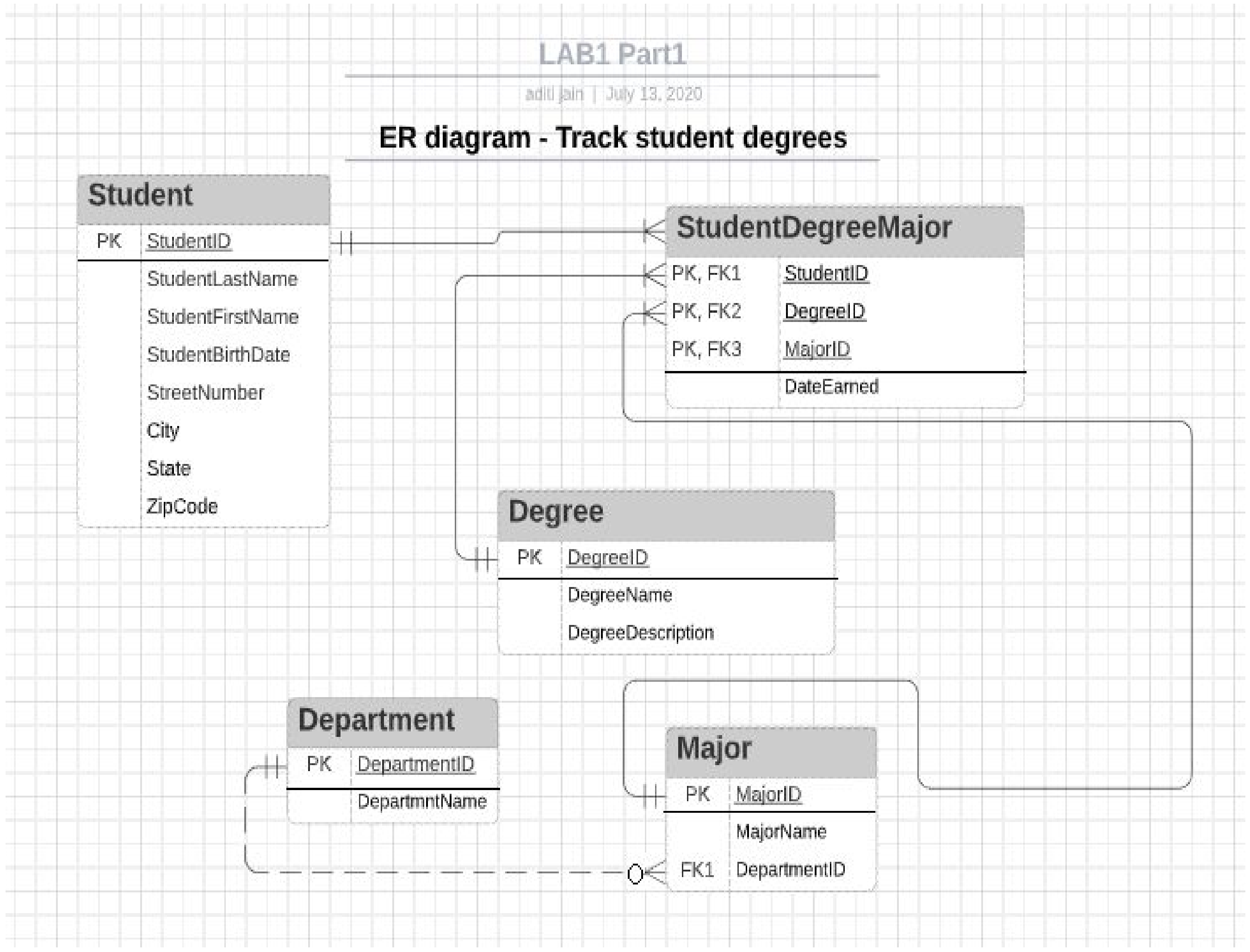
Lab 1 Part 1

ER Diagram is uploaded as Lab1 Part 1 ER diagram.png.

Assumption- Department has to provide a major and department cannot be any administrative or student body department. A major is provided by 1 department. A department can provide one or many major. Student can have one or more degree in one or more major.



Assumption- Department can be any administrative or student body department or provide a major.



Here, we have optional participation for the department when the department is administrative .

Lab1 - Part 2 - Data Integrity

In the new design, the surrogate key ResultID is used instead of the composite primary key (PatientID, TestID) for entity Result. Original key attributes are non-key attributes in the new design. Also, the relationship is non-identifying. To make sure that the business rules are still maintained, following things are to be ensured:

1. In the Result entity, original key attributes i.e. PatientID and TestID should be made **mandatory** fields. This means whenever a new row is inserted in the Result entity, the values for the attributes PatientID and TestID should be entered and values cannot be null.

2. We set up the **referential integrity** for the original key attributes (PatientID, TestID) present in Result such that PatientID is a **foreign key** in Result entity pointing to PatientID in Patient entity and TestID is **foreign key** in Result entity pointing to ResultID in parent entity - Test. So, when a new row is added in the Result entity, the values of attributes PatientID and TestID are matched from the values present in the parent entities - Patient and Test respectively to ensure that the value added is a good value. Good value means only the values that are present in the parent entity can be added in the child entity.

3. Combination of original key attributes (PatientID, TestID) should be **UNIQUE** in the Result entity by enforcing the unique index constraint in the Result entity.

Lab 1 Part 3 MongoDB - Calculate Totals

The code for calculation is:

{

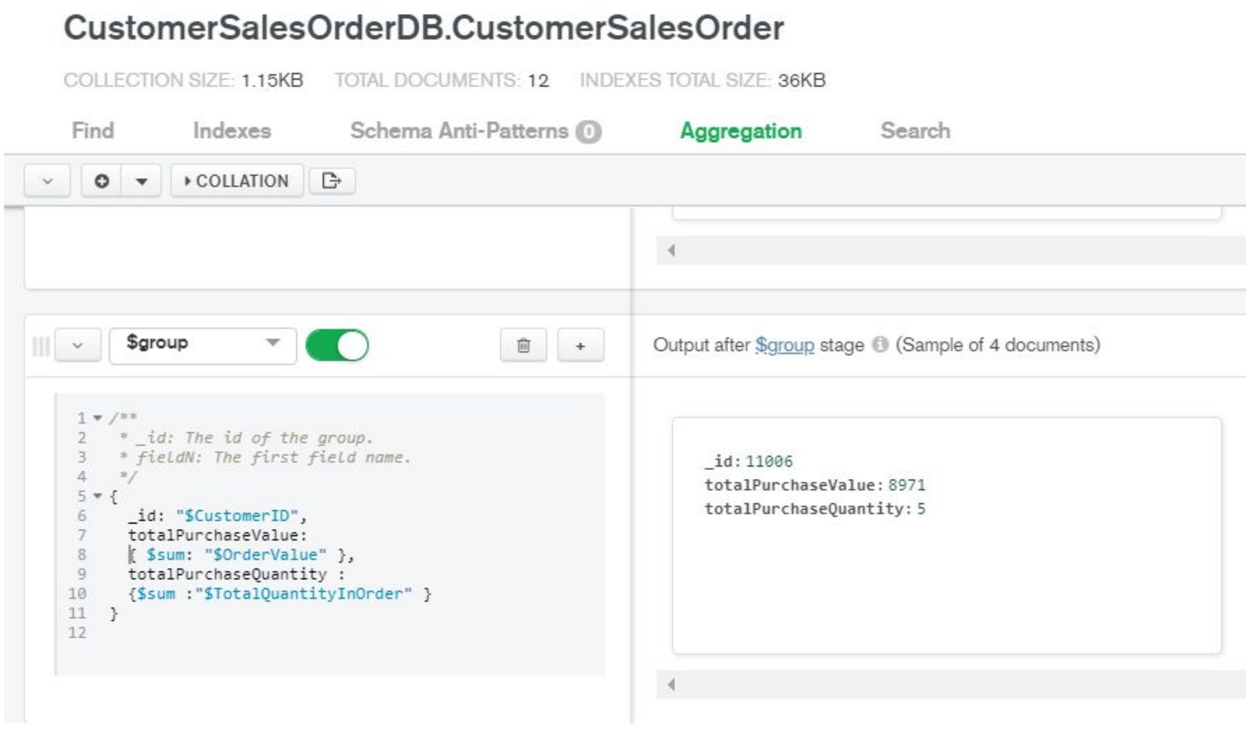
_id: "$CustomerID",

totalPurchaseValue: { $sum: "$OrderValue" },

totalPurchaseQuantity : {$sum :"$TotalQuantityInOrder" }

}

We write the code under the Aggregation tab and choose the $group function to get results..


CustomerSalesOrderDB.CustomerSalesOrder

COLLECTION SIZE: 1.15KB TOTAL DOCUMENTS: 12 INDEXES TOTAL SIZE: 36KB

Output after $group stage (Sample of 4 documents)

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
  _id: "$CustomerID",
  totalPurchaseValue:
  { $sum: "$OrderValue" },
  totalPurchaseQuantity :
  {$sum :"$TotalQuantityInOrder" }
}
```

```
_id: 11006
totalPurchaseValue: 8971
totalPurchaseQuantity: 5
```

Results-

CustomerID- 11005, total purchase value - 8973 , total purchase quantity - 6.

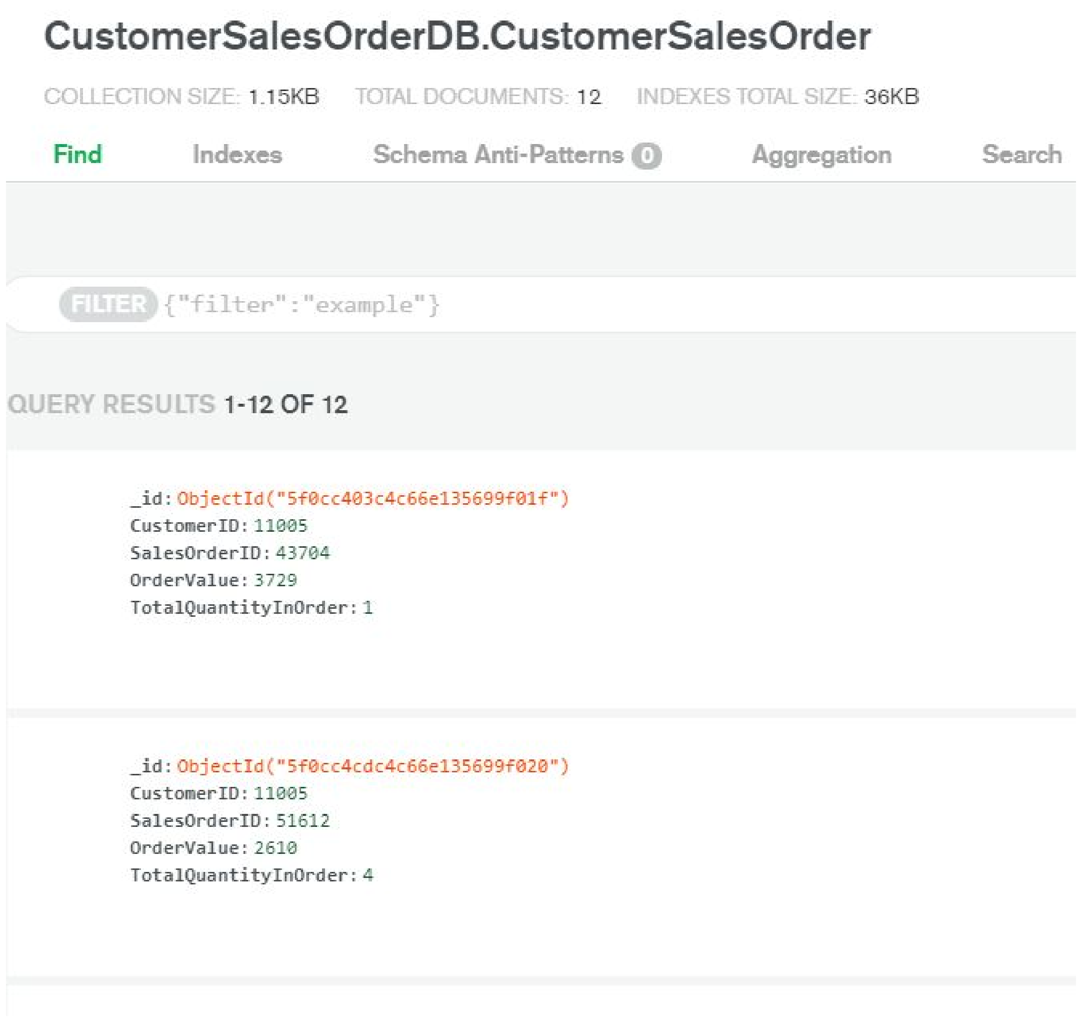
CustomerID- 11006, total purchase value - 8971, total purchase quantity - 5 .

CustomerID- 11007, total purchase value - 9073 , total purchase quantity - 8.

CustomerID- 11008, total purchase value - 8957 , total purchase quantity - 7.

Screenshots

Created document database with 12 documents in the collection, please find below the screenshots-



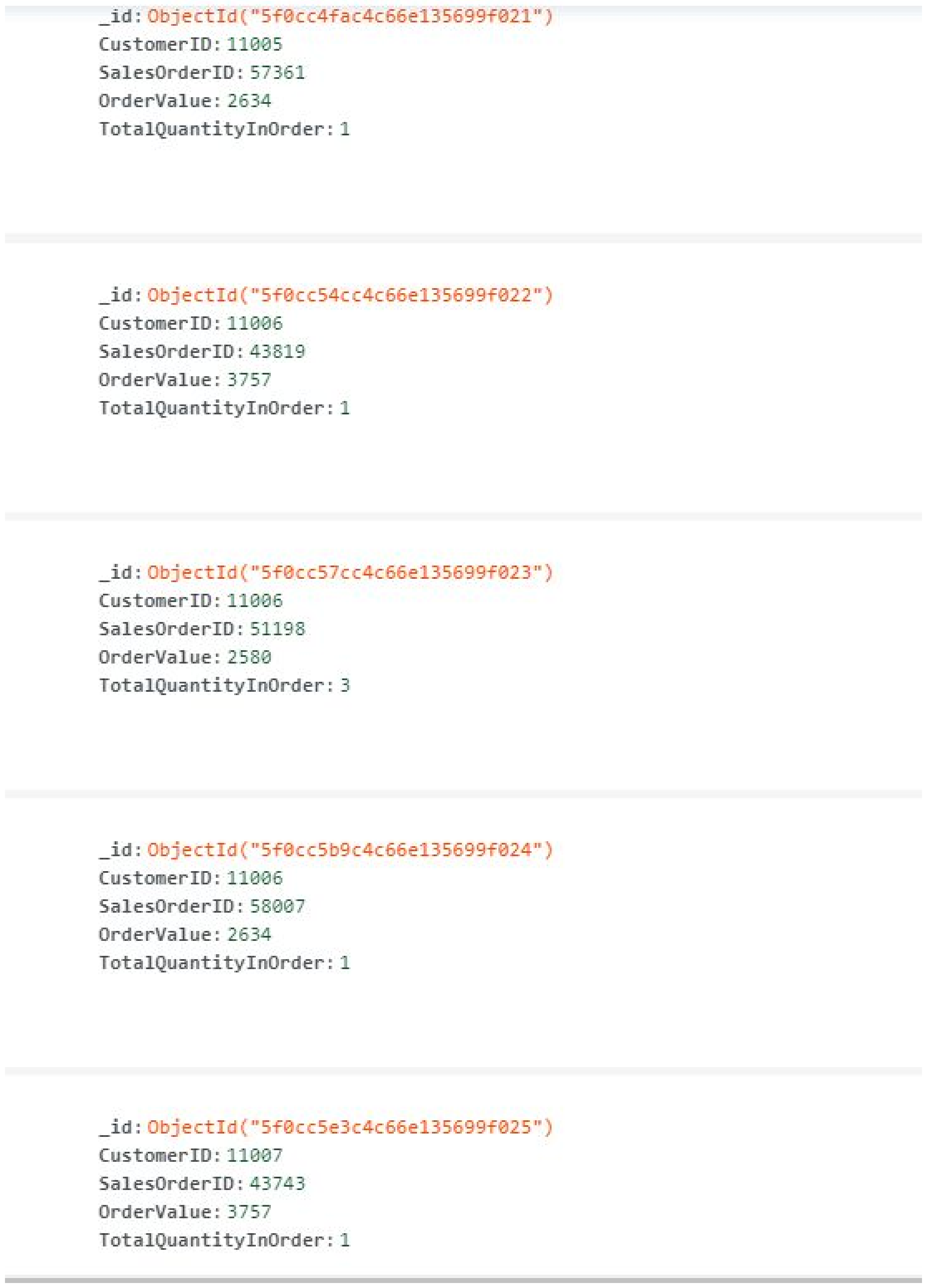
CustomerSalesOrderDB.CustomerSalesOrder

COLLECTION SIZE: 1.15KB   TOTAL DOCUMENTS: 12   INDEXES TOTAL SIZE: 36KB

QUERY RESULTS 1-12 OF 12

```
_id: ObjectId("5f0cc403c4c66e135699f01f")
CustomerID: 11005
SalesOrderID: 43704
OrderValue: 3729
TotalQuantityInOrder: 1
```

```
_id: ObjectId("5f0cc4cdc4c66e135699f020")
CustomerID: 11005
SalesOrderID: 51612
OrderValue: 2610
TotalQuantityInOrder: 4
```



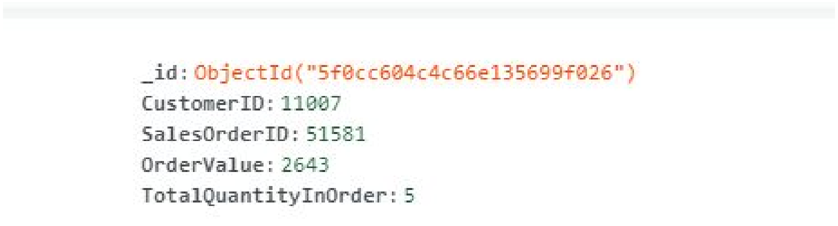
```
_id: ObjectId("5f0cc4fac4c66e135699f021")
CustomerID: 11005
SalesOrderID: 57361
OrderValue: 2634
TotalQuantityInOrder: 1
```

```
_id: ObjectId("5f0cc54cc4c66e135699f022")
CustomerID: 11006
SalesOrderID: 43819
OrderValue: 3757
TotalQuantityInOrder: 1
```

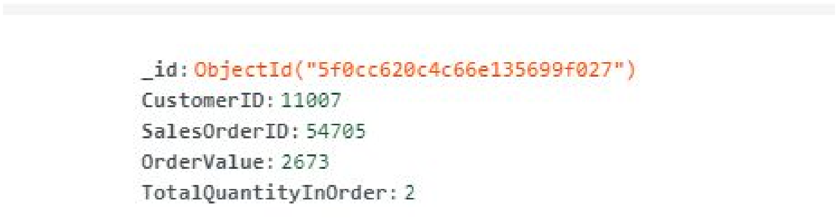
```
_id: ObjectId("5f0cc57cc4c66e135699f023")
CustomerID: 11006
SalesOrderID: 51198
OrderValue: 2580
TotalQuantityInOrder: 3
```

```
_id: ObjectId("5f0cc5b9c4c66e135699f024")
CustomerID: 11006
SalesOrderID: 58007
OrderValue: 2634
TotalQuantityInOrder: 1
```

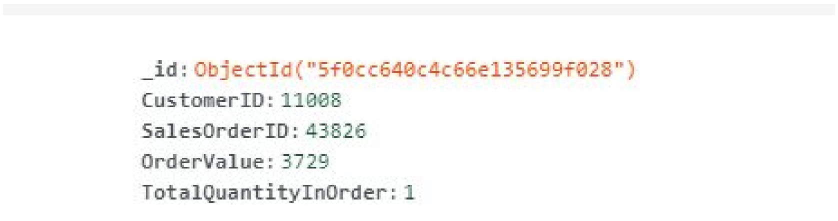
```
_id: ObjectId("5f0cc5e3c4c66e135699f025")
CustomerID: 11007
SalesOrderID: 43743
OrderValue: 3757
TotalQuantityInOrder: 1
```



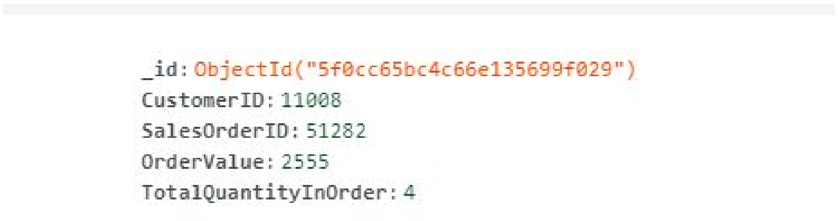
```
_id: ObjectId("5f0cc604c4c66e135699f026")
CustomerID: 11007
SalesOrderID: 51581
OrderValue: 2643
TotalQuantityInOrder: 5
```



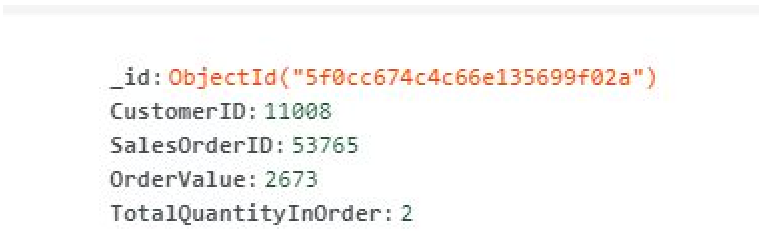
```
_id: ObjectId("5f0cc620c4c66e135699f027")
CustomerID: 11007
SalesOrderID: 54705
OrderValue: 2673
TotalQuantityInOrder: 2
```



```
_id: ObjectId("5f0cc640c4c66e135699f028")
CustomerID: 11008
SalesOrderID: 43826
OrderValue: 3729
TotalQuantityInOrder: 1
```



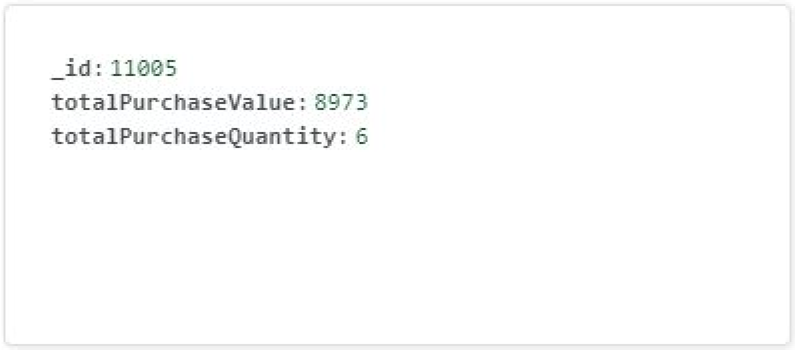
```
_id: ObjectId("5f0cc65bc4c66e135699f029")
CustomerID: 11008
SalesOrderID: 51282
OrderValue: 2555
TotalQuantityInOrder: 4
```



```
_id: ObjectId("5f0cc674c4c66e135699f02a")
CustomerID: 11008
SalesOrderID: 53765
OrderValue: 2673
TotalQuantityInOrder: 2
```

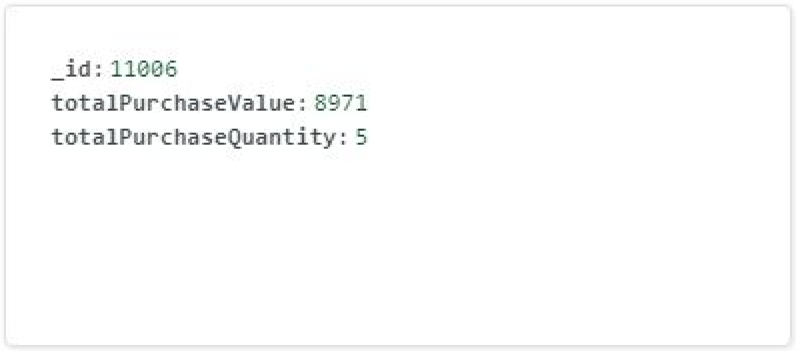
Output of Totals after executing code-

For customer ID- 11005-



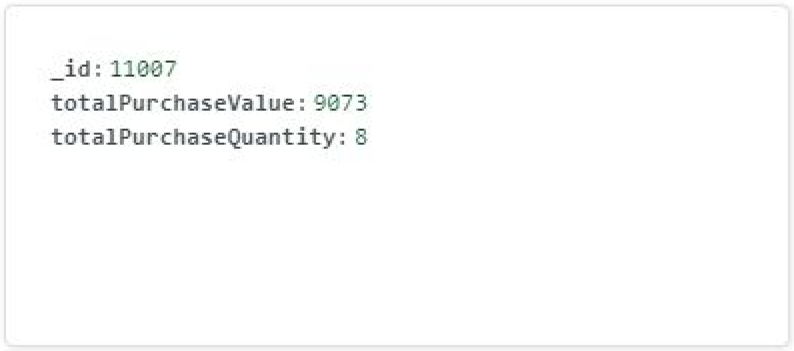
```
_id: 11005
totalPurchaseValue: 8973
totalPurchaseQuantity: 6
```

For customer ID- 11006-



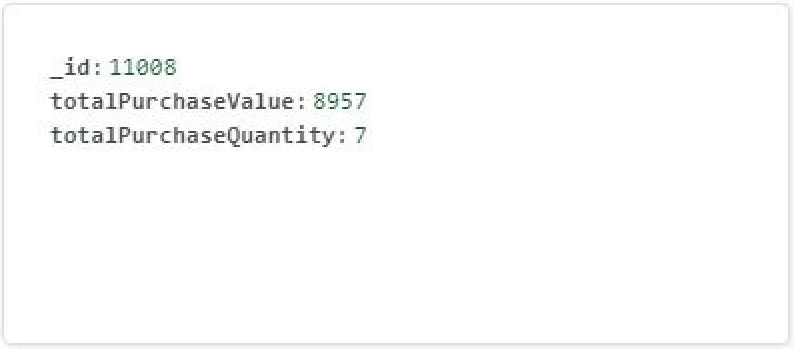
```
_id: 11006
totalPurchaseValue: 8971
totalPurchaseQuantity: 5
```

For customer ID- 11007-



```
_id: 11007
totalPurchaseValue: 9073
totalPurchaseQuantity: 8
```

For customer ID- 11008-



```
_id: 11008
totalPurchaseValue: 8957
totalPurchaseQuantity: 7
```

Code execution screenshot is attached as Part 3 - mongoDB Atlas.png, Results.png.