

Name: HRISAV BHOWMICK

UID: A20011

Use collection "transactions" created by importing "transactions.json" on day 2

```
mongoimport --db transac --jsonArray --collection transactions --file  
E:\Praxis\Classroom\BDSN\NoSQL\1\transactions.json
```

```
show dbs
```

```
use transac
```

```
show collections
```

```
db.transactions.find().pretty()
```

```
{  
  "_id" : ObjectId("60029324723e4ed3746bf615"),  
  "Id" : 103,  
  "Name" : "Dylan",  
  "TransactionId" : "tran4",  
  "Transaction" : [  
    {  
      "ItemId" : "a100",  
      "price" : 200  
    },  
    {  
      "ItemId" : "a110",  
      "price" : 200  
    }  
  ],  
  "Subscriber" : true,  
  "Payment" : null,  
  "Note" : "Payment is Null"  
},  
  
{  
  "_id" : ObjectId("60029324723e4ed3746bf616"),  
  "Id" : 100,  
  "Name" : "John",  
  "TransactionId" : "tran1",  
  "Transaction" : [  
    {  
      "ItemId" : "a100",  
      "price" : 200  
    },  
    {  
      "ItemId" : "a110",  
      "price" : 200  
    }  
  ],  
  "Subscriber" : true,  
  "Payment" : {  
    "Type" : "Credit-Card",  
    "Total" : 400,  
    "Success" : true  
  },  
  "Note" : "1st Complete Record"  
}
```

# 1. Calculate the total transaction amount by aggregating Payment. Total in all records

```
db.transactions.aggregate(  
  {$match: {}},  
  {$group: {_id: "Id", total_pay: {$sum: "$Payment.Total"}}},  
  {$project: {_id: 0, total_pay: 1}}
```

)

```
> db.transactions.aggregate(  
... {$match: {}},  
... {$group: {_id: "Id", total_pay: {$sum: "$Payment.Total"}}},  
... {$project: {_id: 0, total_pay: 1}}  
... )  
< {"total_pay" : 2400 }  
\
```

## # 2. Aggregate per record by aggregating Transaction.price

```
db.transactions.aggregate(  
    {$project: {transaction_price: {$sum: "$Transaction.price"}}}  
)
```

```
> db.transactions.aggregate(  
... {$project: {transaction_price: {$sum: "$Transaction.price"}}}  
... )  
< {"_id" : ObjectId("60009c83c30b8190938cee2a"), "transaction_price" : 400 }  
< {"_id" : ObjectId("60009c83c30b8190938cee2b"), "transaction_price" : 400 }  
< {"_id" : ObjectId("60009c83c30b8190938cee2c"), "transaction_price" : 400 }  
< {"_id" : ObjectId("60009c83c30b8190938cee2d"), "transaction_price" : 400 }  
< {"_id" : ObjectId("60029324723e4ed3746bf613"), "transaction_price" : 400 }  
< {"_id" : ObjectId("60029324723e4ed3746bf614"), "transaction_price" : 400 }  
< {"_id" : ObjectId("60029324723e4ed3746bf615"), "transaction_price" : 400 }  
< {"_id" : ObjectId("60029324723e4ed3746bf616"), "transaction_price" : 400 }  
\
```

## # 3. Aggregate per payment type by adding up Payment.Total

```
db.transactions.aggregate(  
    {$group: {_id: "$Payment.Type", total_pay_by_type: {$sum: "$Payment.Total"}}}  
)
```

```
> db.transactions.aggregate(  
... {$group: {_id: "$Payment.Type", total_pay_by_type: {$sum: "$Payment.Total"}}}  
... )  
< {"_id" : "Credit-Card", "total_pay_by_type" : 1600 }  
< {"_id" : "Debit-Card", "total_pay_by_type" : 800 }  
< {"_id" : null, "total_pay_by_type" : 0 }  
\
```

## # 4. Find the max id

```
db.transactions.aggregate(  
    {$group: {_id: "null", max_id: {$max: "$Id"}}},  
    {$project: {_id: 0, max_id: 1}}  
)
```

```
> db.transactions.aggregate(
... {$group: {_id: "null", max_id: {$max: "$Id"}}},
... {$project: {_id: 0, max_id: 1}}
... )
{ "max_id" : 103 }
```

#### # 5. Find the max Transaction.price

```
db.transactions.aggregate(
    {$group: {_id: "Transaction", max_transac: {$max: {$max: "$Transaction.price"}}},
    {$project: {_id: 0, max_transac: 1}}
)
```

```
> db.transactions.aggregate(
... {$group: {_id: "Transaction", max_transac: {$max: {$max: "$Transaction.price"}}},
... },
... {$project: {_id: 0, max_transac: 1}}
... )
{ "max_transac" : 200 }
```