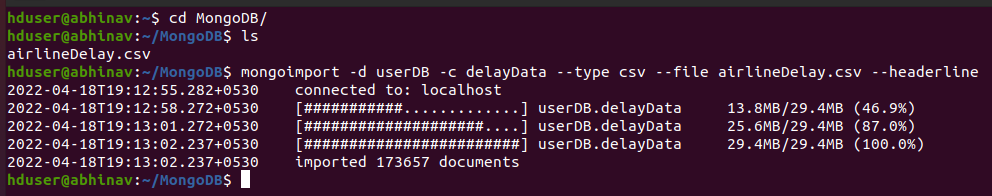
**MongoDB:**

**Queries:**

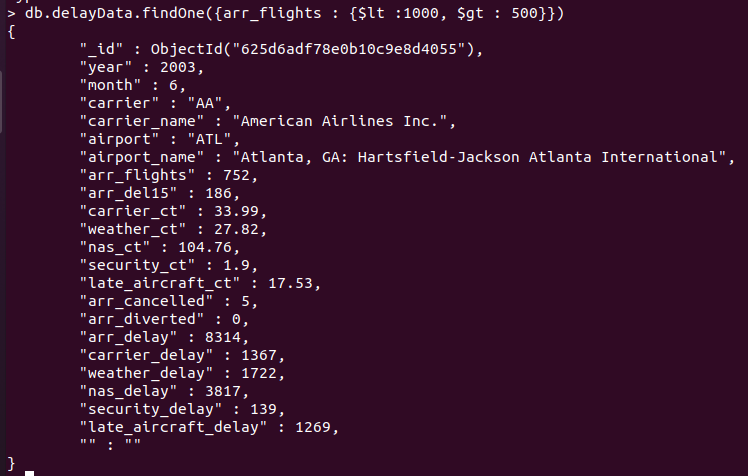
Dataset loading

* + 
  + 

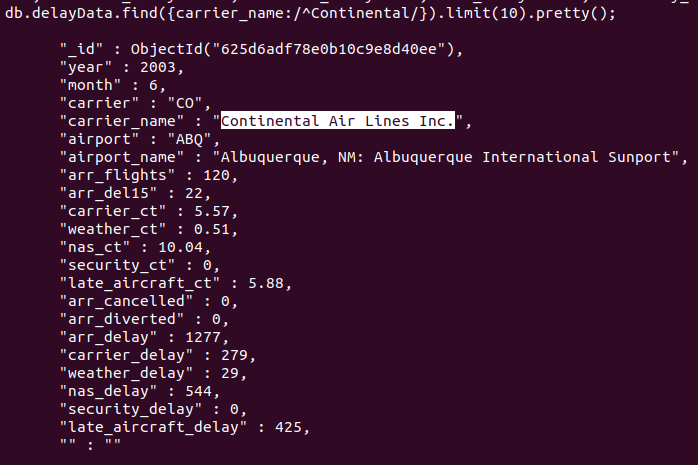
1. **Find the airports that which flights count in between 500 and 1000**

* **db.delayData.find({arr\_flights : {$lt :1000, $gt : 500}})**
* **To find one document:**

db.delayData.findOne({arr\_flights : {$lt :1000, $gt : 500}})

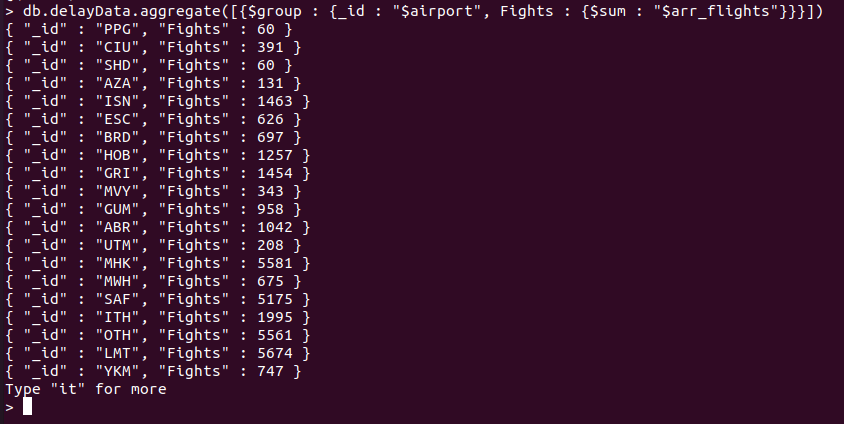
* 

1. **Find the airline name that starts with “Continental”.**

* db.delayData.find({carrier\_name:/^Continental/}).limit(10).pretty();
* 
* 

1. **Find the count of flights that are arriving for airports**

* db.delayData.aggregate([{$group : {\_id : "$airport", Fights : {$sum : "$arr\_flights"}}},{"$sort": {Flights: 1}}])



1. **Find the carrier names that belongs to ANC and IND airports**

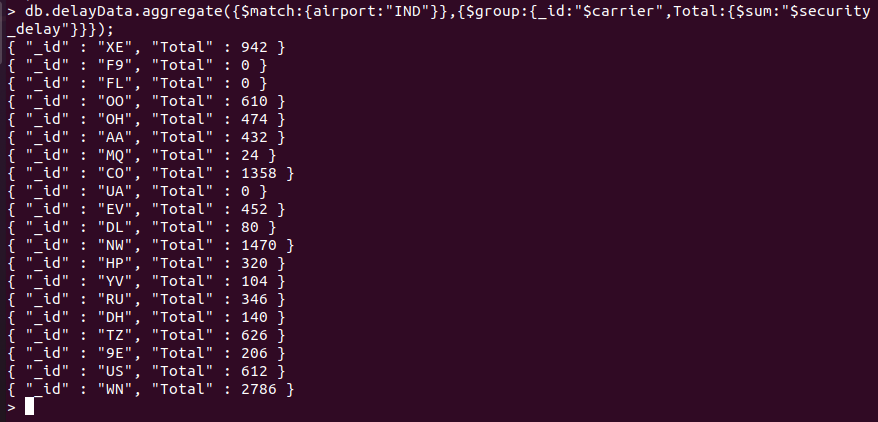
* **db.delayData.find({airport:{$in:['ANC','IND']}}).limit(10).pretty();**

**Output:**



1. **Find the total security delay of IND airport with airlines having security delays**

* **db.delayData.aggregate({$match:{airport:"IND"}},{$group:{\_id:"$carrier",Total:{$sum:"$security\_delay"}}});**



1. **Find the number of flight diversions that occur in St Louis**

**Query:**

* + var m1 = function(){  
     var rval = { airport: this.airport\_name,  
     year: this.year };  
     if( this.airport=="STL") {  
     emit( rval, this.arr\_diverted );  
     }  
    };
  + var r1 = function( key, valArr ) {  
     return Array.sum( valArr );  
    }
  + db.delayData.mapReduce( m1, r1, {out:'tmp'});
  + db.tmp.find().sort( {'\_id.year':-1} )

**Result:**

