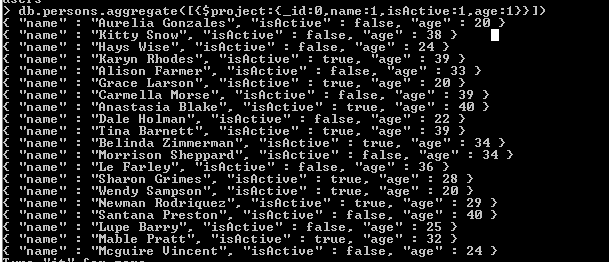
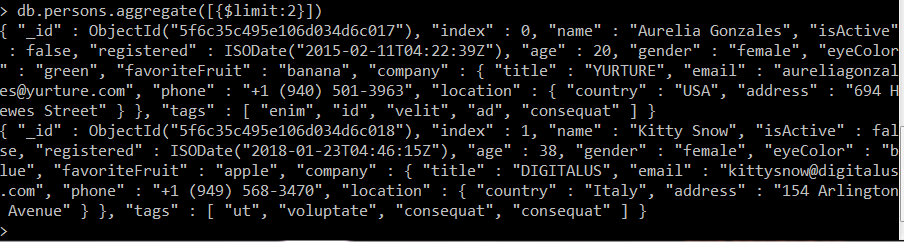
1. Pipelines
   1. $project - Selects specific fields



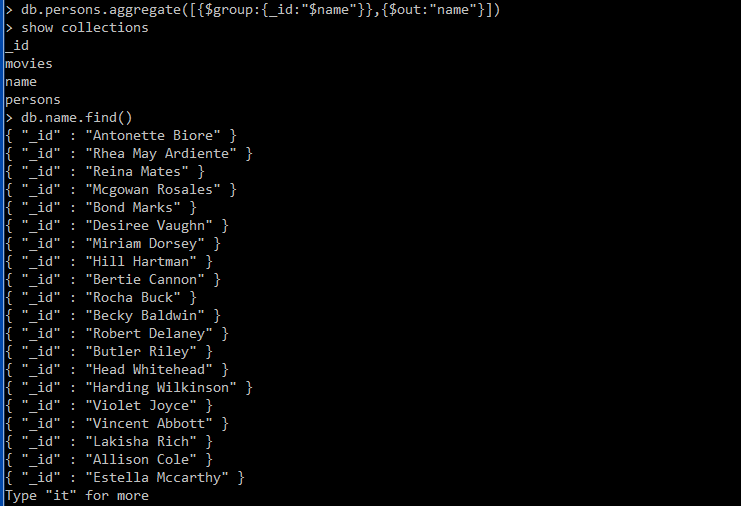
* 1. $limit - limits the amount of documents returned



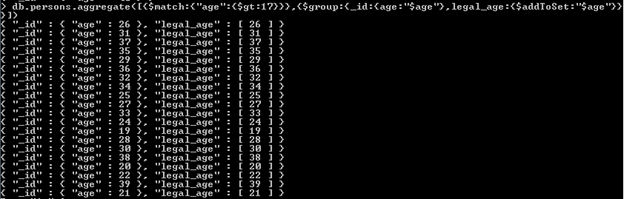
* 1. $skip - used to skip ahead in the array of documents



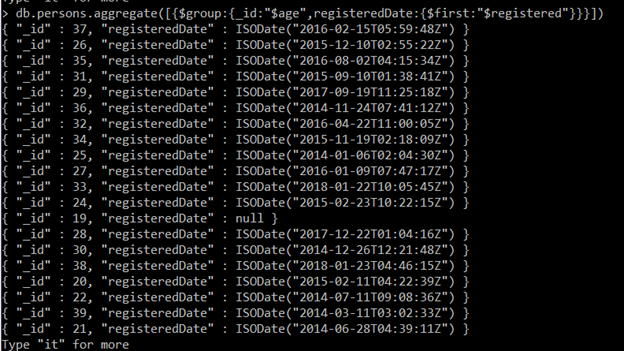
* 1. $out - writes new document into a specified collection



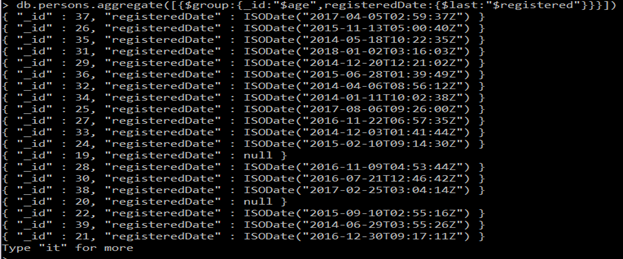
1. $Group Functions
   1. $addToSet - is an operator that adds or appends value into an array if the value does not exist in the array



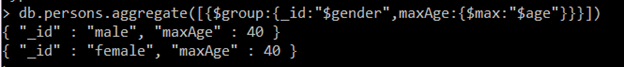
* 1. $first - Returns result from applying expression to the first document. It is only meaningful when the document is in the $group stage.



* 1. $last – returns result from applying expression to the first document. It is only meaningful when the document is in the $group stage.



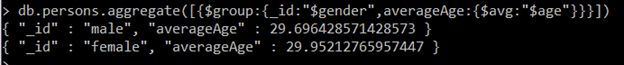
* 1. $max – returns the maximum value



* 1. $min – returns the minimum value. It has one or list of specified expression as its operand

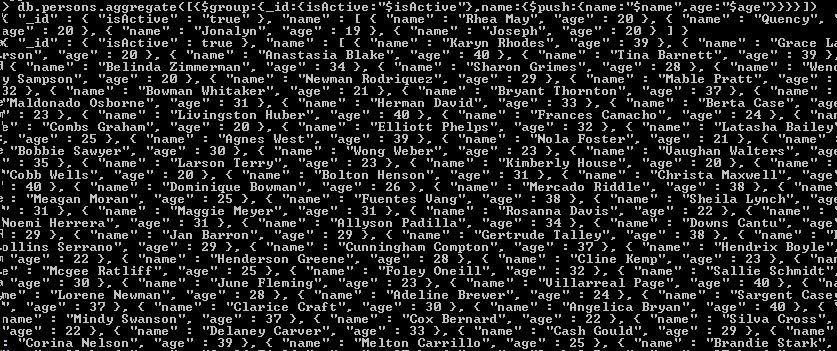


* 1. $avg-returns the collective average of all the numeric values

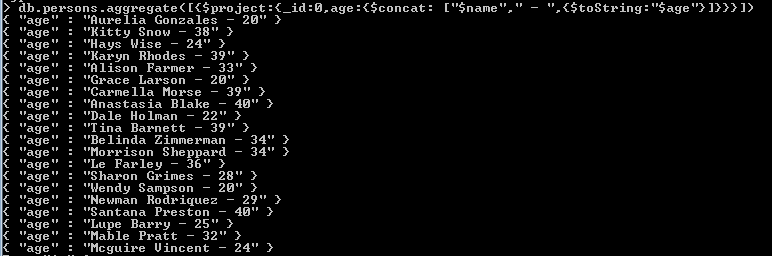


* 1. $push - the collective average of all the numeric values

db.persons.aggregate([{$group:{\_id:{isActive:"$isActive"},name:{$push:{name:"$name",age:"$age"}}}}])



1. String Functions
   1. $concat – joins two string together

**db.persons.aggregate([{$project:{\_id:0,age:{$concat: ["$name", “ - ",{$toString:"$age"}]}}}])**

* 1. $strcasecmp – compares two string with case-insensitive comparison

**db.per sons.aggregate([{$project:{\_id:0,name:1,Comparison:"Hill",compared:{$strcasecmp:["$name","Hill"]}}}])**

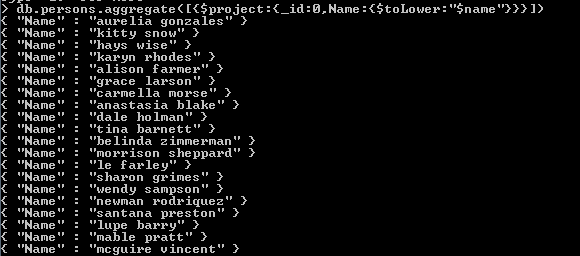
* 1. $substr – return substring with at a specified index

**db.persons.aggregate([{$project:{\_id:0,name:1,nicknames:{$substr:["$name", 0, 3]}}}])**

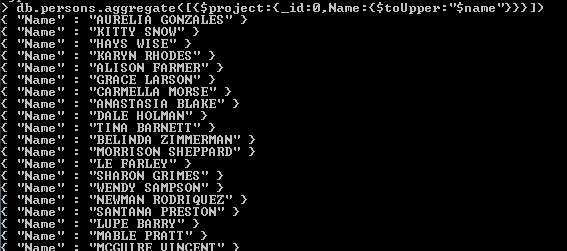


* 1. $toLower – convert string to lowercase

**db.persons.aggregate([{$project:{\_id:0,Name:{$toLower:"$name"}}}])**

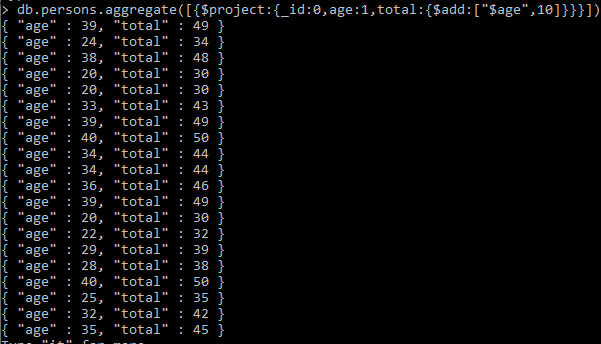


* 1. $toUpper – convert string to uppercase

db.persons.aggregate([{$project:{\_id:0,Name:{$toUpper:"$name"}}}])

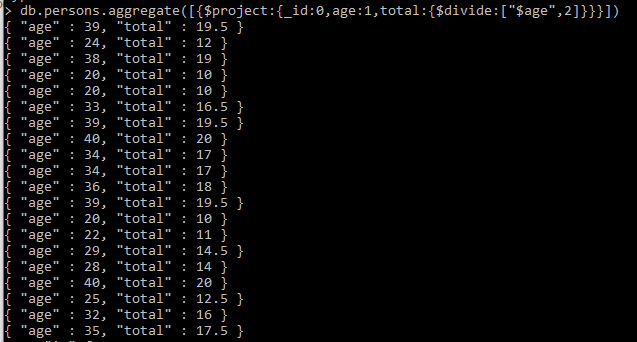
1. Arithmetic
   1. $add - adds both two numbers and return the total value

**db.persons.aggregate([{$project:{\_id:0,age:1,total:{$add:["$age",10]}}}])**



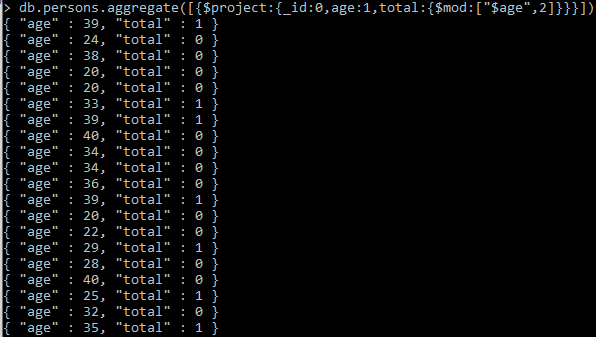
* 1. $divide - divide by one number to another and return the total value

**db.persons.aggregate([{$project:{\_id:0,age:1,total:{$divide:["$age",2]}}}])**

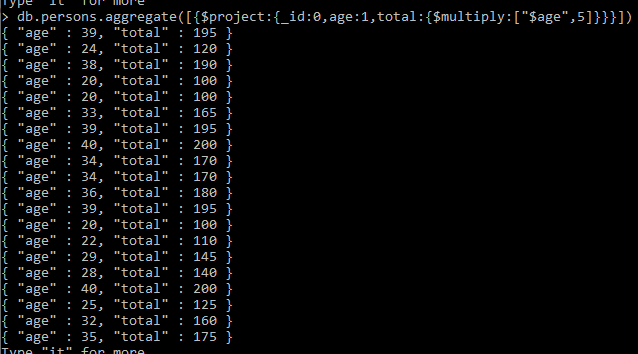


* 1. $mod - it find the number which will be divisible by or divide one number to another and return the remainder value

**db.persons.aggregate([{$project:{\_id:0,age:1,total:{$mod:["$age",2]}}}])**

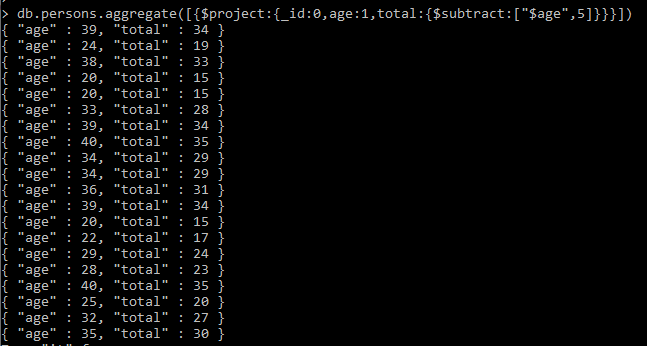


* 1. $multiply - it multiply both two numbers and return the total value

**db.persons.aggregate([{$project:{\_id:0,age:1,total:{$multiply:["$age",5]}}}]**

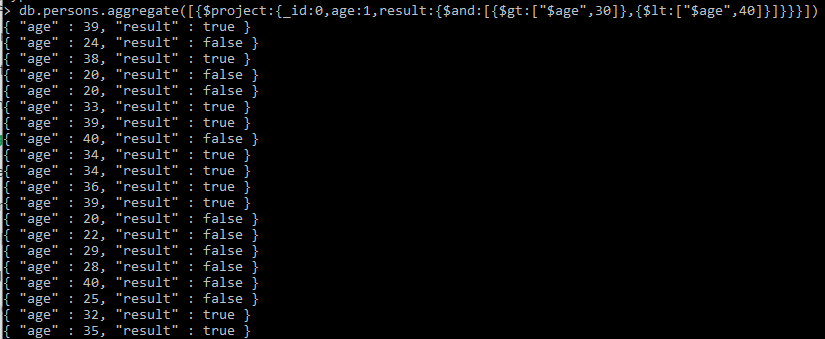
* 1. $subtract - find the difference between two numbers and return the difference

**db.persons.aggregate([{$project:{\_id:0,age:1,total:{$subtract:["$age",5]}}}])**



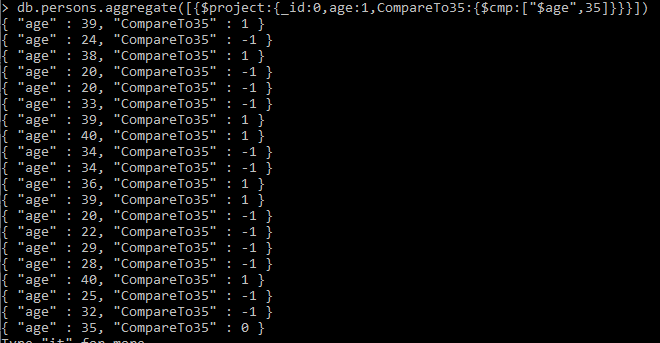
1. Date Functions (No need for sample query, or explanation just give what the functions will return)
   1. $dayOfYear - returns a number between 1 and 366 that refers to the number of days in a year.
   2. $dayOfMonth - returns number 1 to 31 that refers to the days of the month
   3. $dayOfWeek - returns the day of the week as a number between 1 (Sunday) and 7 (Saturday).
   4. $year - returns the year portion of a date.
   5. $month - returns the month of a date as a number between 1 and 12.
   6. $week - returns the week of the year as a number between 0 and 53.
2. Logical Functions
   1. $and - its evaluates if all the expression given is true

**db.persons.aggregate([{$project:{\_id:0,age:1,result:{$and:[{$gt:["$age",30]},{$lt:["$age",40]}]}}}])**



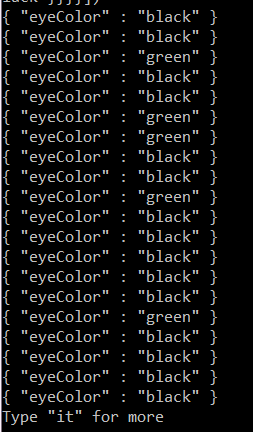
* 1. $cmp - it will compare two values

**db.persons.aggregate([{$project:{\_id:0,age:1,CompareTo35:{$cmp:["$age",35]}}}])**

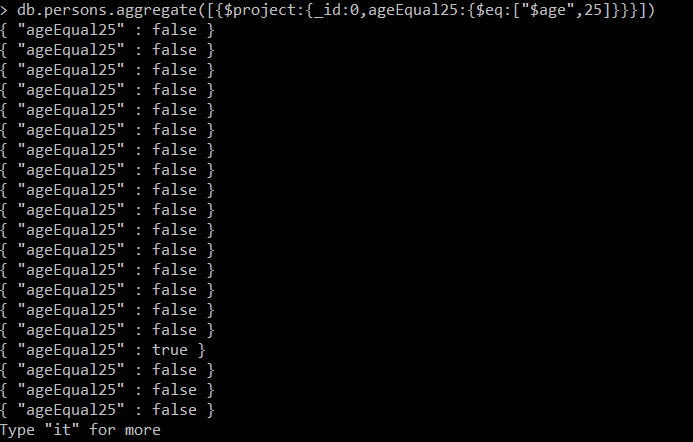


* 1. $cond - it evaluates the Boolean expression and also it required the three arguments which is if, if else and else.

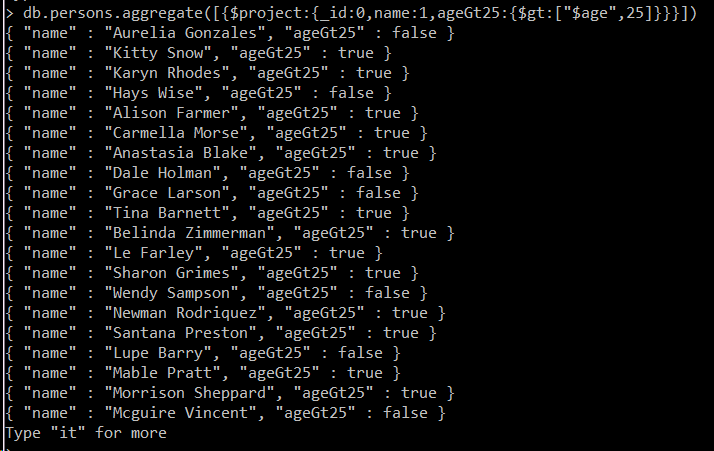




* 1. $eq - it evaluates if both values are equals



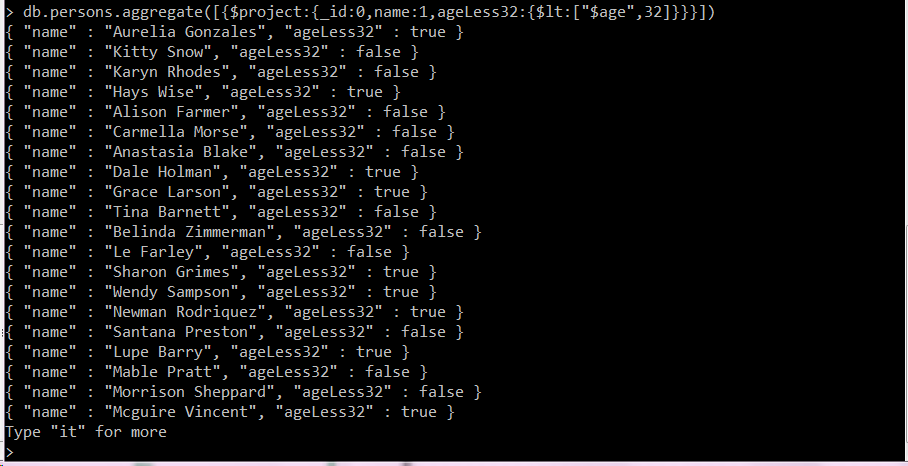
* 1. $gt - it determines both number, which is the greater value it is either the first value or second value and return the higher values



* 1. $gte - it determines each number is greater than or it is equal.



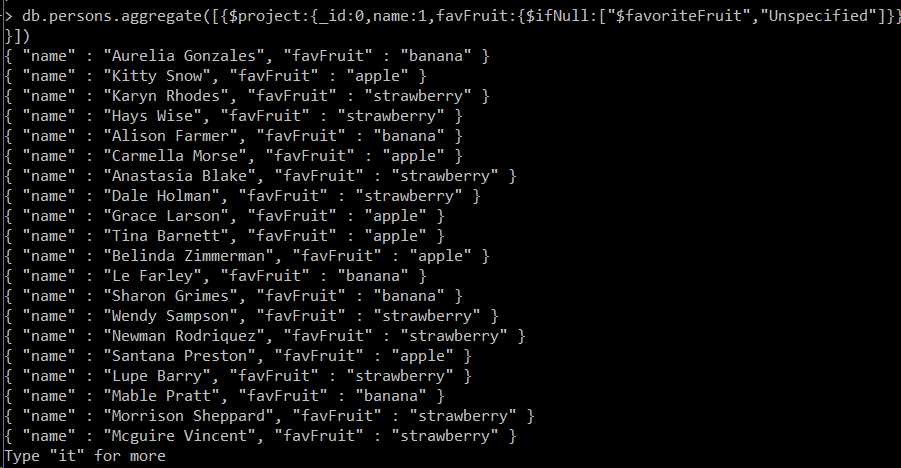
* 1. $lt - check if the field is lesser than the specified value.



* 1. $lte - check if the field is lesser than or equal the specified value.



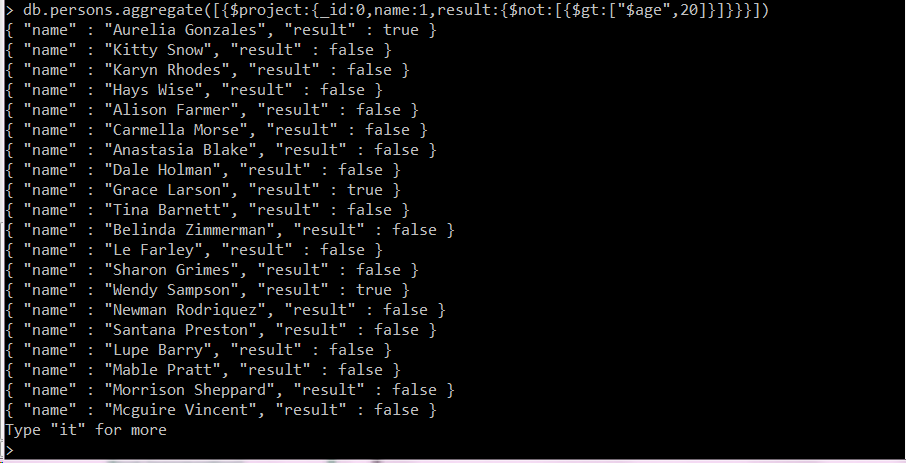
* 1. $ifNull - check if the field is null



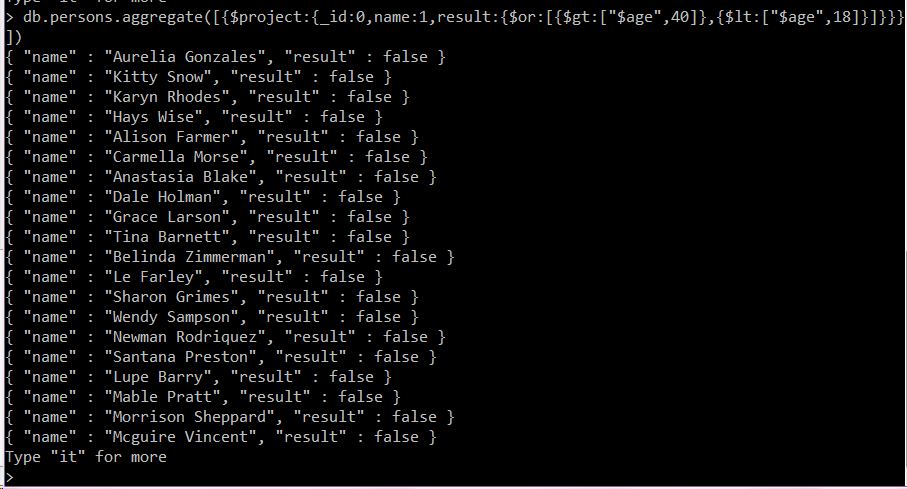
* 1. $ne - compare two value if its equal, compare both value and type



* 1. $not - selects the documents that do not match the expression



* 1. $or - evaluate one or more expressions and return true if one of them satisfies.



1. Set Operators
   1. $setEquals - compares two or more arrays and returns and returns true or false. True if they have the same distinct element.
   2. $setIntersection - takes two or more arrays and returns an array that contains the elements that appear in every input array.
   3. $setDifference - Takes two sets and returns an array containing the elements that only exist in the first set.
   4. $setUnion - takes two or more arrays and returns an array containing the elements that appear in both array and the remaining elements.
   5. $setIsSubset - takes two arrays and check if when the first array is a subset of the second, including if it is equal
   6. $anyElementTrue - evaluates an array and returns true if any of the elements are true and false if the other way around.
   7. $allElementsTrue - evaluates all the elements of an array and returns true if all elements are true and false otherwise.