The translator should only support the following mongodb operators:

$or ,

$and(remember $and & comma separated values on an object are the same)

$lt

$lte

$gt

$gte

$ne

$in

db.user.find({age:{$gte:21}},{name:1,\_id:1});

db.user.find({\_id:23113},{name:1,age:1});

db.user.find({name:'julio'});

db.users.find( { $or: [ { status: "A" } ,{ age: 50 } ] })

db.inventory.find( { type: { $in: [ 'food', 'snacks' ] } } )

db.users.find( { age: { $gt: 25, $lte: 50 } })

db.inventory.find( { $or: [ { qty: { $gt: 100 }, {$lt:500} }, { price: { $lt: 9.95 } } ] } )

The query :

Select SELECT\_CLAUSE from TABLE\_NAME where WHERE\_CLAUSE

Algorithm:

1. First get the first part of substring db.user.find to substitute the TABLE\_NAME ex. user
2. From the second part of the substring ({age:{$gte:21}},{name:1,\_id:1}), populate a hashmap. The Hashmap will contain values like below.

|  |  |
| --- | --- |
| **KEY** | **VALUE** |
| Julio | Name |
| 23113 | \_id |
| 1 | Age, name , \_id |
| {$gte:21} | age |
| [ { status: "A" } ,{ age: 50 } ] | $or |
| { $in: [ 'food', 'snacks' ] } | Type |
| [ { qty: { $gt: 100 } }, { price: { $lt: 9.95 } } ] | $or |

1. Select Clause substitution:
   1. If there is any entry in the map for the key “1”, substitute the associated values in the SELECT\_CLAUSE.
   2. If there is no entry found with key “1”, substitute \* in the SELECT\_CLAUSE.
2. Remove the key “1” and its associated values.
3. Remove the key “0” and its associated values.
4. Where Clause substitution:
   1. Iterate over the map.
   2. Keys not starting with “{“ and “[“ like name=’julio’ should be figured out without any extra parsing.
   3. If the key starts with “{“, the following word will be an operator. Parse that operator.
      1. Possible values are $lt, $lte, $gt, $gte, $ne, $in . The SQL equivalent operator will be available as mapping.
      2. Except $in , all the operators will be operated on a single value. There the condition will be made like age >=21.
      3. For $in, the condition will be made type in (‘food’,’snacks’). No extra parsing is required except change of brackets.
   4. If the key starts with “[“, the condition operator can be $or or $and.
      1. In the case of $or or $and, the condition should go inbetween each {}.
      2. If again individual {} has condition like $gt, $lt, it should go the point c.