Advanced Data Management - Assignment N°04

Name: Manasa Reddy Bandari

E-Mail: manasa-reddy.bandari@stud.uni-bamberg.de

Matriculation number: 1942692

Name: Oormila Ramanandan Kottayi Pilapprathodi

E-Mail: oormila-ramanandan.kottayi-pilapprathodi@stud.uni-bamberg.de

Matriculation number: 1942205

Name: Sini Ann Padinjarekuttu

E-Mail: [sini-ann.padinjarekuttu@stud.uni-bamberg.de](mailto:sini-ann.padinjarekuttu@stud.uni-bamberg.de)

Matriculation number: 1923470

Name: Shashidhar Reddy Nimmagari

E-Mail: Shashidhar-reddy.nimmagari@stud.uni-bamberg.de

Matriculation number: 1985736

Exercises of document database

Exercise1 TASK 1 Use Case: Equi Join

* 1. List of people with their department

db.person.aggregate(

{

$lookup:

    {

        from: "department",

        localField: "works\_in",

        foreignField : "id",

        as: "department"

    }},

    {$project: {"firstname":1,"lastname":1,"department.name":1}}

)

1.2 Number of emails sent out per department

db.email.aggregate([

  {

    $lookup:

      {

        from: "person",

        localField: "MESSAGE\_FROM",

        foreignField: "email\_address",

        as: "messages"

      }

 },

 {

  "$match":{

        "messages.works\_in":{"$exists":true,"$ne":null}

      }

 },

  {"$group" : {\_id:"$messages.works\_in", count:{$sum:1}}}

]);

1.3 Number of emails received per department

Exercise1 TASK 2 Use Case: Theta Join

2.1 Correlation between salary and number of emails

db.email.aggregate([

  {

     $lookup:{

        from:"person",

        localField:"MESSAGE\_FROM",

        foreignField:"email\_address",

        as:"result"

     }

  },

    {

  "$match":{

        "result.email\_address":{"$exists":true,"$ne":null}

      }

   },

   {

       $lookup:{

        from:"depavgsal",

        localField:"result.works\_in",

        foreignField:"\_id",

        as:"res2"

     }

   },

    {

      "$project": {

          "\_id":"$\_id",

          "name":"$result.id",

          "emailid":"$MESSAGE\_FROM",

          "works\_in":"$result.works\_in",

          "salary":"$result.salary",

          "davg":"$res2.avg",

          "salgt":{$gt: ["$result.salary","$res2.avg"]},

          as:"res3"

      }

    },

   {

       "$group": {

           "\_id":{

               "email":"$emailid",

               "works\_in":"$works\_in",

               "salcheck":"$salgt"

           },

           count:{"$sum":1}

       }

   }

]);

db.email.aggregate([

  {

     $lookup:{

        from:"person",

        localField:"MESSAGE\_FROM",

        foreignField:"email\_address",

        as:"result"

     }

  },

    {

  "$match":{

        "result.email\_address":{"$exists":true,"$ne":null}

      }

   },

   {

       $lookup:{

        from:"depavgsal",

        localField:"result.works\_in",

        foreignField:"\_id",

        as:"res2"

     }

   },

    {

      "$project": {

          "\_id":"$\_id",

          "name":"$result.id",

          "emailid":"$MESSAGE\_FROM",

          "works\_in":"$result.works\_in",

          "salary":"$result.salary",

          "davg":"$res2.avg",

          "salgt":{$lt: ["$result.salary","$res2.avg"]},

          as:"res3"

      }

    },

   {

       "$group": {

           "\_id":{

               "email":"$emailid",

               "works\_in":"$works\_in",

               "salcheck":"$salgt"

           },

           count:{"$sum":1}

       }

   }

]);

Exercise1 TASK 3 Use Case: Schema Evolution

3.1 Create a copy of a collection

db.email.find().forEach((\_id)=> { db.email\_temp.insert(\_id)});

3.2 Add information for entity set

db.email\_temp.update({},{$set:{prority:1}},false,true);

3.3 Add information for entity set with default value

  db.email\_temp.update(

  { \_id:ObjectId("5b0d270a9587ce1de41e5bfe")},

  { $set:

     { prority: 3}});

3.4 Drop a table

db.email\_temp.drop()

Exercise1 TASK 4 Use Case: Missing values

4.1 Find missing values

 db.email.aggregate([

 {

  "$match":{

        "MESSAGE\_SUBJECT":""

      }

 },

  {"$group" : {\_id:"MESSAGE\_SUBJECT", count:{$sum:1}}}

]);

4.1 (method 2)

db.email.aggregate(

 [

   {

     $match: {

       MESSAGE\_SUBJECT: {

         $eq: ""

       }

     }

   },

   {

     $count: "MESSAGE\_SUBJECT"

   }

 ]

)

Exercise1 TASK 5 Use Case: Range queries

5.1 Emails between two dates

db.email.find(

{MESSAGE\_DATE:{

$gt:"2001-09-01",

$lt:"2001-10-31"}

},

{"ID":1,"MESSAGE\_DATE":1,"\_id":0})

5.2 Emails between two dates for Larry John May

db.email.aggregate([

    {

        "$match": {

             "MESSAGE\_DATE":

                  {

                   $lt:"2001-10-31",

                   $gt:"2001-09-01"

                  }

               },

    },

    {

    $lookup: {

           from: "person",

           localField: "MESSAGE\_FROM",

           foreignField: "email\_address",

           as: "person"

         }

    },

    {

        $match: {

                 "person.firstname": mb.regex.contains("larry"),

                 "person.lastname":mb.regex.contains("may")

    },

    }

]).pretty()

Exercise1 TASK 6 Use Case: Network analysis

6.1 Network size by e-mail

6.2 Network size by "knows" relation

6.3 2-hop email network

6.4 Employees who sent emails to exact 7 TO-recipients.

db.email.aggregate([

{

$group:{

\_id:"$MESSAGE\_FROM",count:{$sum:1}

}

},

{

$match: {

"count":7

}

}

])

Exercise1 TASK 7 Use Case: User defined functions

7.1 Words and number of occurrences in a certain email

var map = function() {

   var body= this.MESSAGE\_BODY;

   if (body) {

       body= body.toLowerCase().split(/[^A-Za-z]/);

       for (var i = body.length - 1; i >= 0; i--) {

           if (body[i])  {

              emit(body[i], 1);

           }

       }

   }

};

var reduce = function( key, values ) {

   var count = 0;

   values.forEach(function(v) {

       count +=v;

   });

   return count;

};

db.email.mapReduce(map, reduce,{out:"words",limit:1})

7.2 Words and number of occurrences in all emails

var map = function() {

   var body= this.MESSAGE\_BODY;

   if (body) {

       body= body.toLowerCase().split(/[^A-Za-z]/);

       for (var i = body.length - 1; i >= 0; i--) {

           if (body[i])  {

              emit(body[i], 1);

           }

       }

   }

};

var reduce = function( key, values ) {

   var count = 0;

   values.forEach(function(v) {

       count +=v;

   });

   return count;

};

db.email.mapReduce(map, reduce,{out:"totalwordcount")