***MONGO QUERIES***

**Tranformation of Nan values into “ “**

use Volleyball; *#selects the database*

const collections = db.getCollectionNames(); *#gets all the collections*

*#for each collection we ask to iterate each document, checks if any value is equal to “NaN” and eventually transform it into “ “*

collections.forEach((collectionName) => {

print(`working with collection n. : ${collectionName}`);

db[collectionName].find().forEach((doc) => {

let updates = {};

Object.keys(doc).forEach((key) => {

if (typeof doc[key] === "number" && doc[key] !== doc[key]) {

updates[key] = "";

}

});

*#if there are updates execute them*

if (Object.keys(updates).length > 0) {

db[collectionName].updateOne(

{ \_id: doc.\_id },

{ $set: updates }

);

print(`Updated id: ${doc.\_id}`);

}

});

});

**To see the last updates**

db.local.find().pretty();

**We wanted to change the column Height in Height\_in\_cm**

db.athletes.updateMany(

{},

{ $rename: { "Height": "Height\_in\_cm" } }

)

**Merge statistics and Italian players**

db.stat\_set\_2425.aggregate([ *#starting from the setter dataset*

{

$unionWith: {

coll: "stat\_opp\_2425" *#combine with the opposite hitter statistics*

}

},

{

$unionWith: {

coll: "stat\_mb\_2425" #*union with the middle blocker*

}

},

{

$unionWith: {

coll: "stat\_lib\_2425" *#union with the libero*

}

},

{

$unionWith: {

coll: "stat\_spi\_2425" *#union with the spiker*

}

},

*#consider the new collection*

{

$lookup: {

from: "athletes", *#collection to consider for join*

localField: "Athlete ", *#attribute from the first collection*

foreignField: "Athlete", *#attribute of the athlete collection*

as: "Athlete2425" *#final document*

}

},

{

$match: {

"Athlete2425": { $ne: [] } *#consider only the matching documents*

}

},

*#unwind in order to transform the array in a single document*

{

$unwind: "$Athlete2425"

},

{

$out: "stats2425" *#output file*

}

]);