**CRUD**

**2.14** Given a query with a sort and limit, identify the correct output.

**db.pizzas.find().sort({type: 1, price:1}).limit(1)**

2.15 Identify the incorrect projection among a set of expressions.

2.16 Identify how to get all results from a cursor.

db.myCollection.find().toArray()

2.17 Identify the expressions used to count the number of documents matching a query.

db.restaurants.countDocuments({borough: "Brooklyn"})

2.18 Given an indexing scenario, identify the correct command for defining a search index.

db.restaurants.createIndex({ "name": "text", "description": "text" })

2.19 Given a scenario, identify the correct search query:

db.stores.find( { $text: { $search: **"java shop -coffee"** } } )

db.restaurants.aggregate([{ "$search": { "text": { "path": "name", "query": "carvel"} } }])

2.20 Given an aggregation expression using $match, $group, identify the correct output:

db.films.aggregate(

[

    {

        $match: {

            disponibile\_in\_4k: true

            }

    },

    {

        $group: {

            \_id: "$regista",

            sommaFilm: {

                $sum: 1

                }

            }

        }

    ]

)

2.21 Given an aggregation expression using $lookup, identify the correct output.

db.ordini.aggregate(

[

    {

        $lookup: {

            from: "utenti",

            localField: "utente\_id",

            foreignField: "\_id",

            as: "dettagli\_utente"

        }

    }

]

)

2.22 Given an aggregation expression using $out, identify the correct output:

db.films.aggregate(

[

    {

        $match: {

            disponibile\_in\_4k: true

            }

    },

    {

        $group: {

            \_id: "$regista",

            sommaFilm: {

                $sum: 1

                }

            }

        },

    {

        $out: {

            db:"prova",

            coll: "prova"

            }

        }

    ]

)