Document #1

```
{
        “name”: “CarCar&Car” ,
        “address”: {
                “street”: “Montepaolo”
                “number”: 78
        },
        “cars”: [
                { “model”: “Tesla”
                  “year”: 2021
                  “price”: 45’000
                },
                { “model”: “Citroen”
                  “year”: 2003
                  “price”: 10’000
                },
                { “model”: “Fiat”
                  “year”: 1998
                  “price”: 5’300
                }
        ],
        owner: {
                “name”: “Michele”
                “surname”: “Fumagalli”
                “age”: 56
        }
}
```

Document #2

```
{
        “name”: “Deals&Co”,
        “address”: {
                “street”: “Pascani Luca”
                “number”: 89
        },
        “cars”: [
                { “model”: “Ferrari”
                  “year”: 2018
                  “price”: 320’000
                },
                { “model”: “Lamborghini”
                  “year”: 2020
                  “price”: 450’000
                },
                { “model”: “Citroen”
                  “year”: 2001
                  “price”: 12’000
                }
        ],
        owner: {
                “name”: “Filippo”
                “surname”: “Mercanti”
                “age”: 41
        }
}
```

Document #3

```
{
        “name”: “Buy&SellCars”,
        “address”: {
                “street”: “Papa Giovanni”
                “number”: 12
        },
        “cars”: [
                { “model”: “BMW”
                  “year”: 2013
                  “price”: 35’000
                },
                { “model”: “Audi”
                  “year”: 2019
                  “price”: 45’000
                },
                { “model”: “Mercedes”
                  “year”: 2016
                  “price”: 56’000
                }
        ],
        owner: {
                “name”: “Mattia”
                “surname”: “Scanzani”
                “age”: 48
        }
}
```

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.find({
                  “$or”: [{“address.number”: {“$gte”: 80} },
                         {“$and”: [{“cars.model”: “BWM”},
                                   {“owner.name”: “Mattia”}]
                  ]})
```

□ Document #1

■ Document #2

■ Document #3

□ None

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.aggregate([
                    {“$match”: {“cars.model”: “Citroen”} },
                    {“$match”: {“cars.price”: {“$gt”: 100’000} } }
               ])
```

□ Document #1
■ Document #2
□ Document #3
□ None

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
        {“$unwind”: {“path”: “$cars”} },
        {“$match”: {“cars.brand”: “Ferrari”} },
        {“$group”: {“_id”: “$cars.price”,
                “tot”: {“$sum”: “$cars.price”} }
])
```

_id	**tot**	… ATTR #3…
320’000	320’000	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
        {“$unwind”: {“path”: “$cars”} },
        {“$group”: {“_id”: “$cars.price”,
                “tot”: {“$sum”: 1} }
])
```

_id	**tot**	… ATTR #3…
45’000	2	…
10’000	1	…
5’300	1	…
320’000	1	…
450’000	1	…
12’000	1	…
35’000	1	…
56’000	1	…
…	…	…
…	…	…

Which of the following querie**s** returns (at least) Document #1?

```
■  db.car_dealers_collection.aggregate([
            {“$match”: {“cars.year”: 2021} },
            {“$match”: {“cars.model”: “Fiat”} },
     ])
```

```
■  db.car_dealers_collection.find({“cars.price”: {“$lt”: 10’000} })
```

```
■  db.car_dealers_collection.find({ }, {“cars”: 1, “owner”: 1})
```

```
□  db.car_dealers_collection.aggregate([
            {“$match”: {“cars”: {“$elemMatch”: {“year”: 2021, “model”: “Citroen”} } } }
     ])
```

□ None

Document #1

```
{
        “name”: “Car&Car” ,
        “address”: {
                “street”: “Montepaolo”
                “number”: 35
        },
        “cars”: [
                { “model”: “Tesla”
                  “year”: 2023
                  “price”: 85’000
                },
                { “model”: “Citroen”
                  “year”: 2003
                  “price”: 14’000
                },
                { “model”: “Fiat”
                  “year”: 1998
                  “price”: 9’300
                }
        ],
        owner: {
                “name”: “Michele”
                “surname”: “Remora”
                “age”: 39
        }
}
```

Document #2

```
{
        “name”: “Deals&Co”,
        “address”: {
                “street”: “Luccarelli”
                “number”: 19
        },
        “cars”: [
                { “model”: “Ferrari”
                  “year”: 2017
                  “price”: 320’000
                },
                { “model”: “Ferrari”
                  “year”: 2020
                  “price”: 450’000
                },
                { “model”: “Ferrari”
                  “year”: 2001
                  “price”: 152’000
                }
        ],
        owner: {
                “name”: “Fernando”
                “surname”: “Gianotti”
                “age”: 60
        }
}
```

Document #3

```
{
        “name”: “Buy&Sell”,
        “address”: {
                “street”: “Sant’Antonio”
                “number”: 27
        },
        “cars”: [
                { “model”: “BMW”
                  “year”: 2017
                  “price”: 150’000
                },
                { “model”: “Audi”
                  “year”: 2019
                  “price”: 25’000
                },
                { “model”: “Mercedes”
                  “year”: 2019
                  “price”: 260’000
                }
        ],
        owner: {
                “name”: “Mattia”
                “surname”: “Fernandi”
                “age”: 36
        }
}
```

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
        {“$unwind”: {“path”: “$cars”} },
        {“$match”: {“cars.model”: “Ferrari”} },
        {“$group”: {“_id”: true,
                        “tot”: {“$sum”: 10’000} }
])
```

_id	**tot**	… ATTR #3…
true	30’000	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…

Which of the following querie**s** returns (at least) Document #2?

□ db.car_dealers_collection.find({“cars.price”: {“$lt”: 10’000} })

■ db.car_dealers_collection.find({ }, {“cars”: 1, “owner”: 1})

□
```
db.car_dealers_collection.aggregate([
        {“$match”: {“cars”: {“$elemMatch”: {“year”: 2021, “model”: “Ferrari”} } } }
  ])
```

□
```
db.car_dealers_collection.aggregate([
        {“$match”: {“cars.year”: {“$lt”: 2021} } },
        {“$match”: {“cars.model”: {“$ne”: “Ferrari”} } },
  ])
```

□ None

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
        {“$unwind”: {“path”: “$cars”} },
        {“$group”: {“_id”: “$cars.model”,
                        “tot”: {“$sum”: “$cars.price”} }
])
```

_id	**tot**	… ATTR #3…
Tesla	85’000	…
Citroen	14’000	…
Fiat	9’300	…
Ferrari	922’000	…
BMW	150’000	…
Audi	25’000	…
Mercedes	260’000	…
…	…	…
…	…	…
…	…	…

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.aggregate([
                {“$match”: {“cars.model”: “Citroen”} },
                {“$match”: {“cars.price”: {“$gt”: 100’000} } }
            ])
```

- □ Document #1
- □ Document #2
- □ Document #3
- ■ None

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.find({
                “$or”: [{“address.number”: {“$gte”: 80} },
                      {“$and”: [{“cars.model”: “BWM”},
                                {“owner.name”: “Mattia”}]
                ]})
```

□ Document #1
□ Document #2
■ Document #3
□ None

Document #1

```
{
        “name”: “CarDeals”,
        “address”: {
                “street”: “Luccarelli”
                “number”: 1
        },
        “cars”: [
                { “model”: “Ferrari”
                  “year”: 2017
                  “price”: 320’000
                },
                { “model”: “Ferrari”
                  “year”: 2020
                  “price”: 450’000
                },
                { “model”: “Ferrari”
                  “year”: 2001
                  “price”: 152’000
                }
        ],
        owner: {
                “name”: “Fernando”
                “surname”: “Mandelli”
                “age”: 30
        }
}
```

Document #2

```
{
        “name”: “Deals&Co”,
        “address”: {
                “street”: “Santa Maria M.”
                “number”: 36
        },
        “cars”: [
                { “model”: “BMW”
                  “year”: 2018
                  “price”: 320’000
                },
                { “model”: “Mercedes”
                  “year”: 2020
                  “price”: 46’000
                },
                { “model”: “Ferrari”
                  “year”: 2001
                  “price”: 190’000
                }
        ],
        owner: {
                “name”: “Dennis”
                “surname”: “Bracchetti”
                “age”: 49
        }
}
```

Document #3

```
{
        “name”: “CarsCarsGo”,
        “address”: {
                “street”: “Giovanni Paolo”
                “number”: 18
        },
        “cars”: [
                { “model”: “BMW”
                  “year”: 2013
                  “price”: 35’000
                },
                { “model”: “Audi”
                  “year”: 2019
                  “price”: 45’000
                },
                { “model”: “Mercedes”
                  “year”: 2016
                  “price”: 56’000
                }
        ],
        owner: {
                “name”: “Simon”
                “surname”: “Pirelli”
                “age”: 48
        }
}
```

Which of the following querie**s** returns (at least) Document #3?

```
■ db.car_dealers_collection.find({“cars.price”: {“$gt”: 30’000} })
```

```
□ db.car_dealers_collection.aggregate([
          {“$match”: {“cars”: {“$elemMatch”: {“year”: 2019, “model”: “Ferrari”} } } }
   ])
```

```
■ db.car_dealers_collection.aggregate([
          {“$match”: {“cars.year”: {“$lt”: 2021} } },
          {“$match”: {“cars.model”: {“$ne”: “Mercedes”} } },
   ])
```

```
■ db.car_dealers_collection.find({ }, {“cars”: 1, “name”: 1})
```

□ None

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
        {“$unwind”: {“path”: “$cars”} },
        {“$match”: {“cars.model”: “Ferrari”} },
        {“$group”: {“_id”: “$cars.model”,
                        “tot”: {“$sum”: “$cars.year”} }
])
```

_id	**tot**	… ATTR #3…
Ferrari	8’039	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
        {“$unwind”: {“path”: “$cars”} },
        {“$group”: {“_id”: true,
                        “tot”: {“$sum”: 1} }
])
```

_id	**tot**	… ATTR #3…
true	9	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.aggregate([
                        {“$match”: {“cars.model”: “Ferrari”} },
                        {“$match”: {“cars.price”: {“$lt”: 100’000} } }
                  ])
```

□ Document #1
■ Document #2
□ Document #3
□ None

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.find({
                    “$or”: [{“address.number”: {“$gte”: 80} },
                            {“$and”: [{“cars.model”: “BWM”},
                                      {“owner.name”: “Mattia”}]
                    ]})
```

□ Document #1

□ Document #2

□ Document #3

■ None

Document #1

```
{
        “name”: “NiceCars” ,
        “address”: {
                “street”: “Paolo Sarca”
                “number”: 45
        },
        “cars”: [
                { “model”: “BMW”
                  “year”: 2021
                  “price”: 45’000
                },
                { “model”: “Fiat”
                  “year”: 2003
                  “price”: 23’000
                },
                { “model”: “Fiat”
                  “year”: 1998
                  “price”: 9’300
                }
        ],
        owner: {
                “name”: “Riccardo”
                “surname”: “Savesi”
                “age”: 39
        }
}
```

Document #2

```
{
        “name”: “CarDeals”,
        “address”: {
                “street”: “Lucarelli”
                “number”: 12
        },
        “cars”: [
                { “model”: “BMW”
                  “year”: 2018
                  “price”: 220’000
                },
                { “model”: “Mercedes”
                  “year”: 2020
                  “price”: 120’000
                },
                { “model”: “Bentley”
                  “year”: 2019
                  “price”: 170’000
                }
        ],
        owner: {
                “name”: “Laura”
                “surname”: “Mastro”
                “age”: 31
        }
}
```

Document #3

```
{
        “name”: “Deals&Co”,
        “address”: {
                “street”: “Santa Maria M.”
                “number”: 6
        },
        “cars”: [
                { “model”: “Audi”
                  “year”: 2018
                  “price”: 320’000
                },
                { “model”: “Audi”
                  “year”: 2020
                  “price”: 46’000
                },
                { “model”: “Ferrari”
                  “year”: 2001
                  “price”: 190’000
                }
        ],
        owner: {
                “name”: “Marina”
                “surname”: “De Bernardi”
                “age”: 51
        }
}
```

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
        {“$unwind”: {“path”: “$cars”} },
        {“$group”: {“_id”: “$cars.year”,
                        “tot”: {“$sum”: 1} }
])
```

_id	**tot**	… ATTR #3…
2021	1	…
2003	1	…
1998	1	…
2018	2	…
2020	2	…
2019	1	…
2001	1	…
…	…	…
…	…	…
…	…	…

Consider the following query. Complete the table with its output. It may not be necessary to fill all the columns and/or rows.

```
db.car_dealers_collection.aggregate([
      {“$unwind”: {“path”: “$cars”} },
      {“$match”: {“cars.brand”: “BMW”} },
      {“$group”: {“_id”: “$cars.year”,
                  “tot”: {“$sum”: “$cars.price”} }
])
```

_id	**tot**	… ATTR #3…
2021	45’000	…
2018	220’000	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…

Which of the following querie**s** returns (at least) Document #3?

■ db.car_dealers_collection.find({“cars.price”: {“$lt”: 300’000} })

■ db.car_dealers_collection.aggregate([
        {“$match”: {“cars”: {“$elemMatch”: {“year”: 2020, “model”: “Audi”} } } }
  ])

□ db.car_dealers_collection.aggregate([
        {“$match”: {“cars.year”: {“$lt”: 2001} } },
        {“$match”: {“cars.model”: {“$ne”: “Mercedes”} } },
  ])

■ db.car_dealers_collection.find({ }, {“cars”: 1, “name”: 1, “owner”: 1})

□ None

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.aggregate([
                {“$match”: {“cars.price”: {“$gt”: 150’000} } },
                {“$match”: {“cars.price”: {“$lt”: 100’000} } }
            ])
```

□ Document #1

□ Document #2

■ Document #3

□ None

Consider the following query. Which document**s** are returned?

```
db.car_dealers_collection.find({
                    “$or”: [{“address.number”: {“$gte”: 3} },
                            {“$and”: [{“cars.model”: “BWM”},
                                      {“owner.name”: “Mattia”}]
                    ]})
```

- ■ Document #1
- ■ Document #2
- ■ Document #3
- □ None