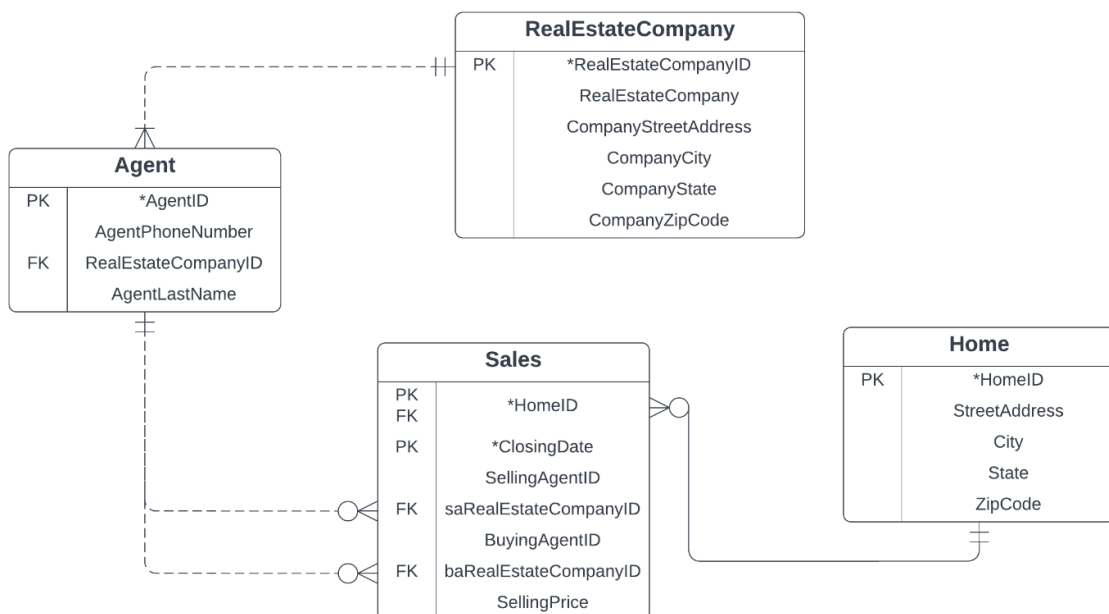


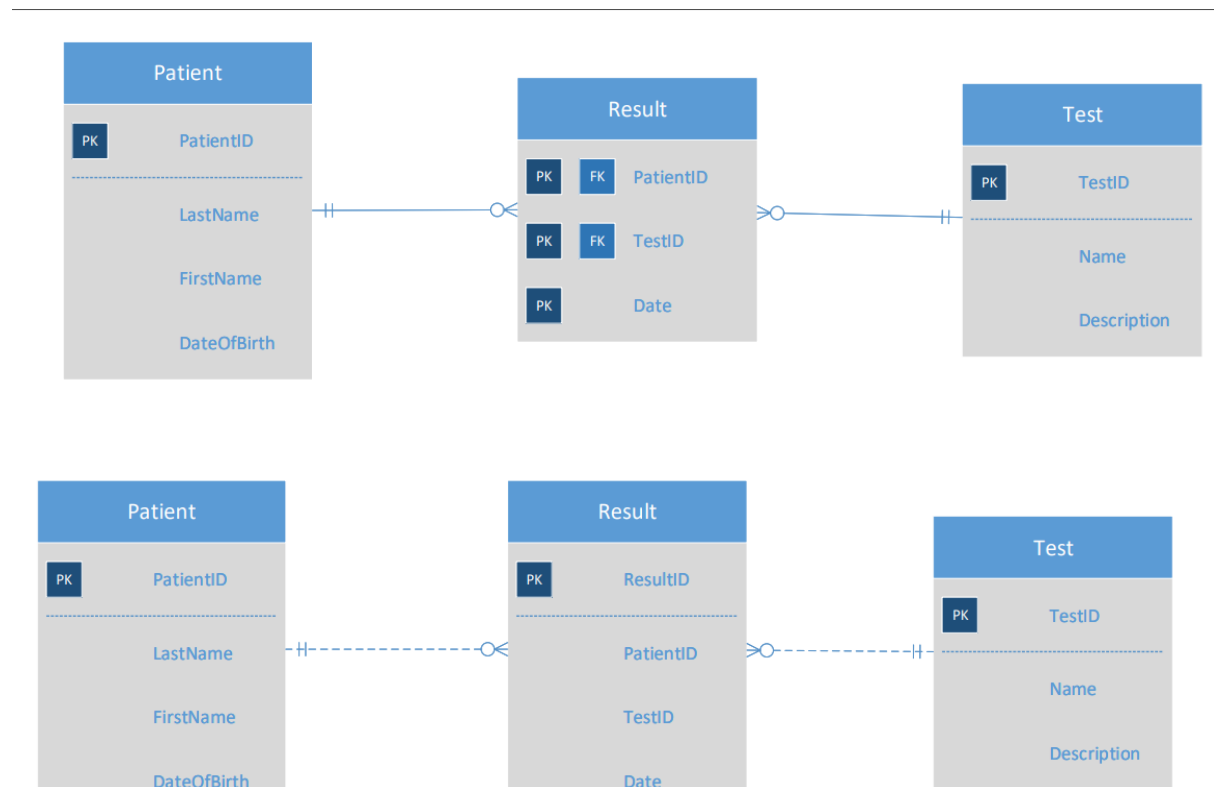
LAB 1

Ashi Tyagi, NUID: 002706544

Part 1:



Part 2:



To ensure data integrity & business rules certain conditions must be met:

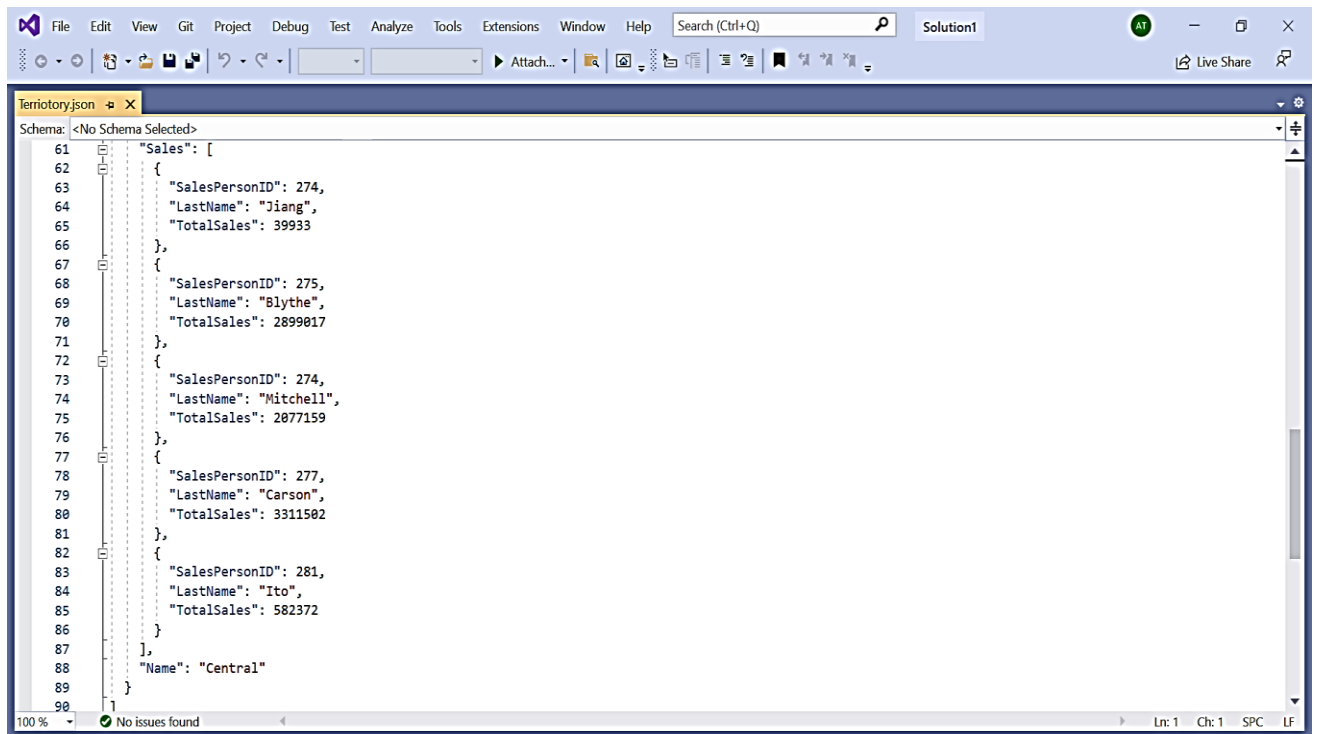
1. First, we must make sure that the intended primary key attributes of the parent entity are necessarily present in the child entity. This entails that the inherited attributes of the parent entity to the child entity must be mandatorily present. In the above example the primary key attributes of PatientID and the TestID (foreign keys for Result entity) must be necessarily present in the Result entity attributes.
2. Secondly, we must ensure that referential integrity constraint is also met for the foreign key attributes present. This implies, for the above example, that we must ensure the values added to the Result entity are correct and matches the original values entered in the Patient and Test entity.
3. Lastly, we must ensure that the combination of the foreign key attributes present in child entity is original or unique. For example, the combination of foreign key attributes present in the Result entity, if made, must be unique. We can enforce this by creating a unique index of the combination of original key attributes.

Part 3:

JSON file:

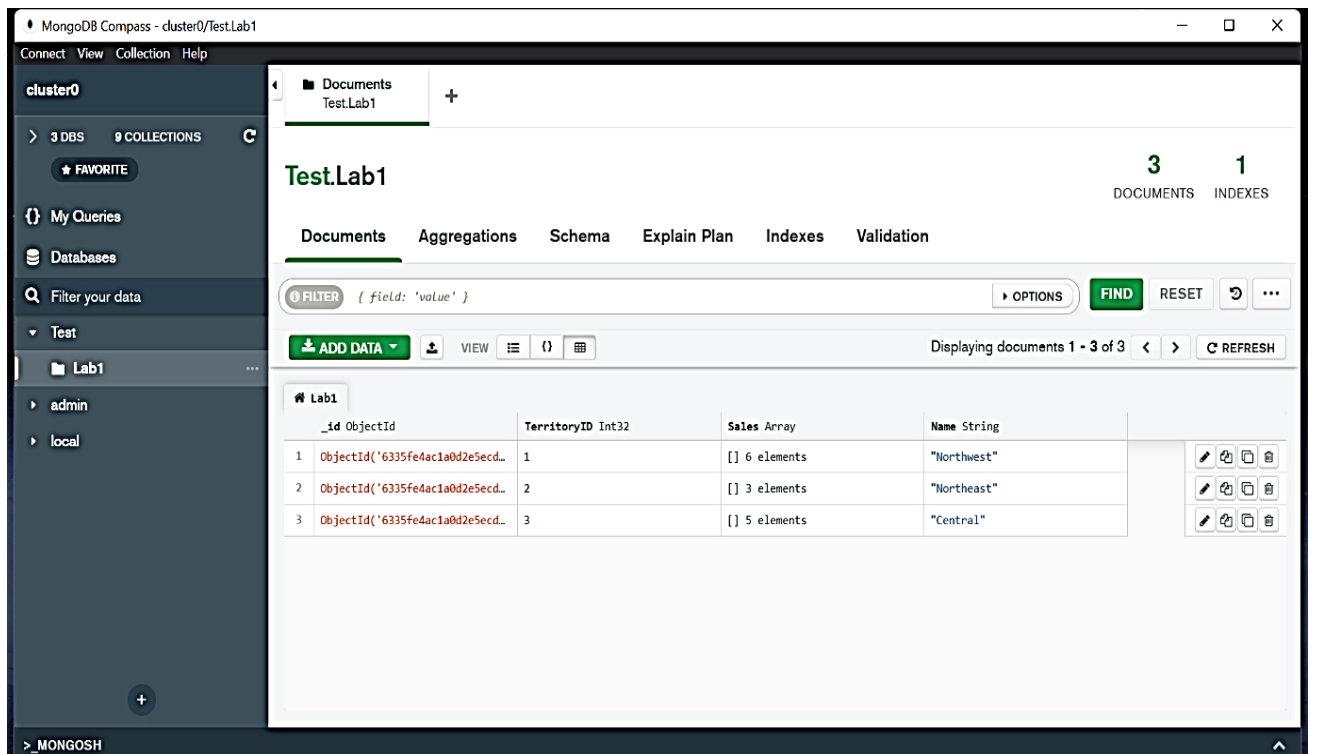
```
1 {
2   {
3     "TerritoryID": 1,
4     "Sales": [
5       {
6         "SalesPersonID": 274,
7         "LastName": "Jiang",
8         "TotalSales": 262466
9       },
10      {
11        "SalesPersonID": 276,
12        "LastName": "Mitchell",
13        "TotalSales": 2353659
14      },
15      {
16        "SalesPersonID": 280,
17        "LastName": "Ansan-Wolfe",
18        "TotalSales": 3748246
19      },
20      {
21        "SalesPersonID": 281,
22        "LastName": "Ito",
23        "TotalSales": 848175
24      },
25      {
26        "SalesPersonID": 283,
27        "LastName": "Campbell",
28        "TotalSales": 4207895
29      },
30    ]
31  }
32 }
```

```
31   "SalesPersonID": 284,
32   "LastName": "Mensa-Anan",
33   "TotalSales": 2608116
34 },
35 ],
36 "Name": "Northwest"
37 },
38 {
39   "TerritoryID": 2,
40   "Sales": [
41     {
42       "SalesPersonID": 274,
43       "LastName": "Jiang",
44       "TotalSales": 93733
45     },
46     {
47       "SalesPersonID": 275,
48       "LastName": "Blythe",
49       "TotalSales": 3301670
50     },
51     {
52       "SalesPersonID": 277,
53       "LastName": "Carson",
54       "TotalSales": 4417588
55     }
56   ],
57   "Name": "Northeast"
58 },
59 {
60   "TerritoryID": 3,
```

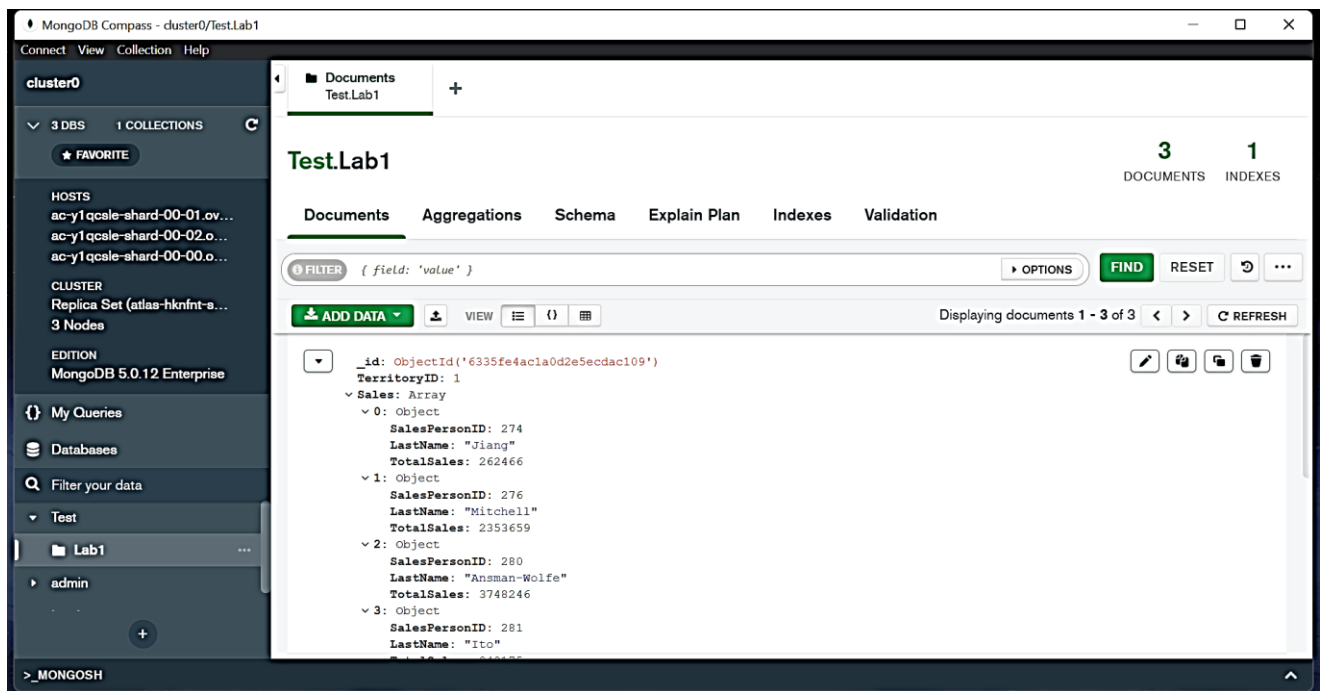


```
61  "Sales": [  
62    {  
63      "SalesPersonID": 274,  
64      "LastName": "Jiang",  
65      "TotalSales": 39933  
66    },  
67    {  
68      "SalesPersonID": 275,  
69      "LastName": "Blythe",  
70      "TotalSales": 2899817  
71    },  
72    {  
73      "SalesPersonID": 274,  
74      "LastName": "Mitchell",  
75      "TotalSales": 2877159  
76    },  
77    {  
78      "SalesPersonID": 277,  
79      "LastName": "Carson",  
80      "TotalSales": 3311582  
81    },  
82    {  
83      "SalesPersonID": 281,  
84      "LastName": "Ito",  
85      "TotalSales": 582372  
86    }  
87  ],  
88  "Name": "Central"  
89  }  
90  ]
```

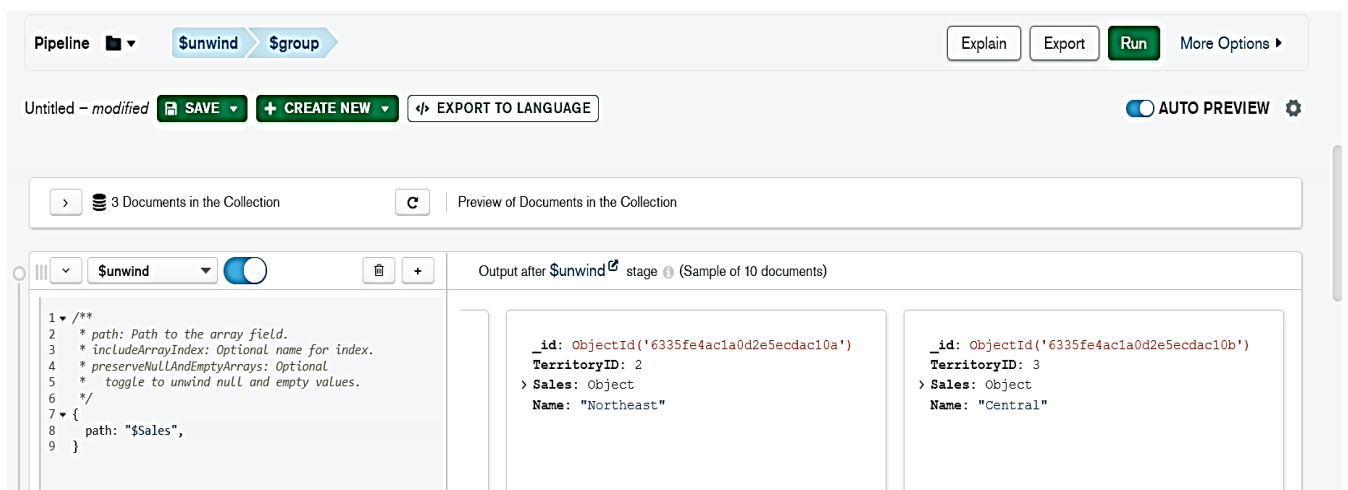
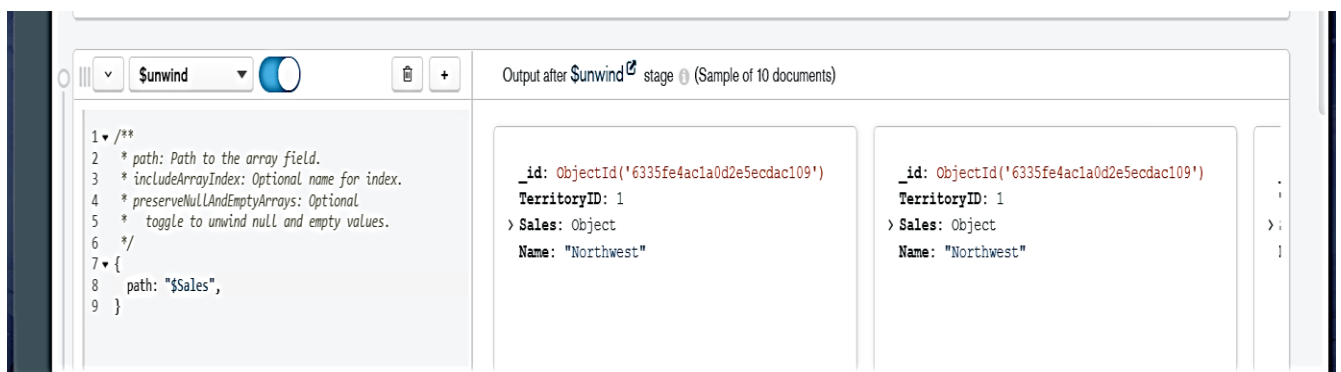
Imported Data



_id	TerritoryID	Sales	Name
ObjectId('6335fe4ac1a0d2e5ecd...')	1	[6 elements]	"Northwest"
ObjectId('6335fe4ac1a0d2e5ecd...')	2	[3 elements]	"Northeast"
ObjectId('6335fe4ac1a0d2e5ecd...')	3	[5 elements]	"Central"



Aggregations



Total Territories

Pipeline **\$unwind** **\$group** Explain Export Run More Options

Untitled - modified **SAVE** **CREATE NEW** EXPORT TO LANGUAGE **AUTO PREVIEW**

Output after **\$group** stage (Sample of 3 documents)

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   id: "$TerritoryID",
7   TotalTerritories: {
8     $sum: "$Sales.TotalSales"
9   }
10 }
```

_id: 1 TotalTerritories: 14028557	_id: 2 TotalTerritories: 7812991
--	---

Pipeline **\$unwind** **\$group** Explain Export Run More Options

Untitled - modified **SAVE** **CREATE NEW** EXPORT TO LANGUAGE **AUTO PREVIEW**

Output after **\$group** stage (Sample of 3 documents)

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   id: "$TerritoryID",
7   TotalTerritories: {
8     $sum: "$Sales.TotalSales"
9   }
10 }
```

_id: 2 TotalTerritories: 7812991	_id: 3 TotalTerritories: 8909983
---	---

Result

Documents **Aggregations** Schema Explain Plan Indexes Validation

Pipeline **\$unwind** **\$group** Edit Explain Export Run Less Options

COLLATION { locale: 'simple' } **MAX TIME MS** 60000

Showing 1 - 3 count results VIEW

_id: 1 TotalTerritories: 14028557
_id: 2 TotalTerritories: 7812991
_id: 3 TotalTerritories: 8909983

Code:

```
[[{
  $unwind: {
    path: '$Sales'
  }
}, {
  $group: {
    _id: '$TerritoryID',
    TotalTerritories: {
      $sum: '$Sales.TotalSales'
    }
  }
}]
```