Q1-a)

```
> db.reserves.update(
... {"marina.name": {$regex: /^Port/}},
... {$set: {"marina.name": "Port Nicholson"}},
... {multi:true}
... )
WriteResult({ "nMatched" : 7, "nUpserted" : 0, "nModified" : 6 })
```

Q1-b)

```
> db.reserves.update(
... ... {"_id":ObjectId("54f102de0b54b61a031776ed")},
... ... {$rename: {"reserves.boat.numbver": "reserves.boat.number"}}
... ... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Q1-c)

```
> db.reserves.insert({ "marina" : { "name" : "Port Nicholson", "location" :
"Wellington" }, "reserves" : { "boat" : { "name" : "Tarakihi", "number" : 717, "color" :
"red", "driven_by" : [ "row", "motor" ] }, "sailor" : { "name" : "Eileen", "sailorId" : 919,
"skills" : [ "sail", "motor", "swim" ], "address" : "Lower Hutt" }, "date" : "2017-03-
25" } })
WriteResult({ "nInserted" : 1 })
```

Q1-d)

```
> db.reserves.insert({"marina" : { "name" : "Port Nicholson", "location" :
"Wellington" }, "reserves" : { "boat" : { "name" : "Dolphin", "number" : 110, "color" :
"white", "driven_by" : [ ] }, "sailor" : { "name" : "James", "sailorId" : 707, "skills" :
[ "row", "sail", "motor", "fish" ], "address" : "Wellington" }, "date" : "2017-03-28" } })
WriteResult({ "nInserted" : 1 })
```

Q1-e)

```
> db.reserves.insert({"marina" : { "name" : "Sea View", "location" : "Petone" },
"reserves" : { "boat" : { name: "Night Breeze", number: 818, color: "black", driven_by:
["row"]}, "sailor" : { "name" : "Paul", "sailorId" : 110, "skills" : [ "row", "swim" ],
"address" : "Upper Hutt" }, "date" : "2017-03-29" } })

WriteResult({ "nInserted" : 1 })
```

Q1-f)

i. Create unique index by sailor's ID and date

```
> db.reserves.createIndex( { "reserves.sailor.sailorId": 1, "reserves.date":1 }, { unique:
true, name: "sailor_date_idx" } )

{

        "createdCollectionAutomatically" : false,

        "numIndexesBefore" : 2,

        "numIndexesAfter" : 3,

        "ok" : 1

}
```

ii. Create unique index by boat's number and date

```
> db.reserves.createIndex( { "reserves.boat.number": 1, "reserves.date":1 }, { unique:
true, name: "boat_date_idx" } )

{

        "createdCollectionAutomatically" : false,

        "numIndexesBefore" : 1,

        "numIndexesAfter" : 2,

        "ok" : 1

}
```

iii.

I used the following method to check whether the indexes perform as expected.

Insert a document whose key is already in the collection. If occur a " duplicate key error " message, it means the index perform as expected.

```
> db.reserves.insert({"marina" : { "name" : "Sea View", "location" : "Petone" },
"reserves" : { "boat" : { name: "Night Breeze", number: 818, color: "black", driven_by:
["row"]}, "sailor" : { "name" : "Milan", "sailorId" : 818, "skills" : [ "row", "sail", "motor",
"first aid" ], "address" : "Wellington" }, "date" : "2017-03-21" } })

WriteResult({

        "nInserted" : 0,

        "writeError" : {

                "code" : 11000,

                "errmsg" : "insertDocument :: caused by :: 11000 E11000 duplicate key error
index: test.reserves.$boat_date_idx  dup key: { : 818.0, : \"2017-03-21\" }"

        }

})


> db.reserves.insert({"marina" : { "name" : "Sea View", "location" : "Petone" },
"reserves" : { "boat" : { name: "Red Cod", number: 616, color: "yellow", driven_by:
["sail","motor"]}, "sailor" : { "name" : "James", "sailorId" : 707, "skills" : [ "row",
"swim" ], "address" : "Upper Hutt" }, "date" : "2017-03-28" } })

WriteResult({

        "nInserted" : 0,

        "writeError" : {

                "code" : 11000,

                "errmsg" : "insertDocument :: caused by :: 11000 E11000 duplicate key error
index: test.reserves.$sailor_date_idx  dup key: { : 707.0, : \"2017-03-28\" }"

        }

})
```

Q2-a)

```
> db.reserves.count()

17
```

Q2-b)

```
> db.reserves.count({"marina.name":"Port Nicholson"})

9
```

Q2-c)

```
> db.reserves.distinct("reserves.sailor.name")

[

        "James",

        "Peter",

        "Milan",

        "Eileen",

        "Charmain",

        "Gwendolynn",

        "Paul"

]
```

Q2-d)

```
> db.reserves.find({"reserves.date":"2017-03-16"},{ "marina.name": 1,
"reserves.boat.name": 1,"reserves.sailor.name":1, "_id":0 })

{ "marina" : { "name" : "Sea View" }, "reserves" : { "boat" : { "name" : "Flying Dutch" },
"sailor" : { "name" : "Peter" } } }

{ "marina" : { "name" : "Port Nicholson" }, "reserves" : { "boat" : { "name" :
"Mermaid" }, "sailor" : { "name" : "Milan" } } }
```

or

```
> db.reserves.distinct('marina.name', {'reserves.date': "2017-03-16" } );

[ "Sea View", "Port Nicholson" ]

> db.reserves.distinct('reserves.boat.name' ,  {'reserves.date': "2017-03-16" });

[ "Flying Dutch", "Mermaid" ]

> db.reserves.distinct('reserves.sailor.name',  {'reserves.date': "2017-03-16" });

[ "Peter", "Milan" ]
```

Q2-e)

```
> db.reserves.distinct("reserves.sailor.name", {"reserves.sailor.skills":"swim"})

[ "Eileen", "Paul" ]
```

Q2-f)

```
> db.reserves.distinct("reserves.sailor.name",
{"reserves.sailor.skills":{ $all:["row","sail","motor"], $size: 3}})

[ "Peter" ]
```

Create time_table

Q3-a)

```
> db.time_table.insert( { "date": "2017-03-28", "line_name": "Hutt Valley Line", "service_no": 2, "time": 1045, "distance": 34.3, "latitude": -41.2865, "longitude": 174.7762, "stop": "Wellington", "driver" : { "driver_name": "fred", "email": "fred@ecs.vuw.ac.nz", "password": "f00f", "mobile": "2799797", "current_position": "Wellington", "skill": [ "Ganz Mavag", "Guliver" ]} , "vehicle": { "vehicle_id": "KW3300", "status": "in_use", "type": "Matangi" }, "data_point":[ {"sequence": "2016-03-28 21:17:40+0000", "longitude": -41.2012, "latitude": 175.07, "speed": 70.1}, {"sequence": "2016-03-28 21:07:40+0000", "longitude": -41.2262, "latitude": 174.77, "speed": 69.2} ] })

WriteResult({ "nInserted" : 1 })

> db.time_table.createIndex( { date: 1, line_name: 1, service_no: 1, time: 1}, { unique: true, name: "time_table_idx" }  )

{

        "createdCollectionAutomatically" : false,

        "numIndexesBefore" : 1,

        "numIndexesAfter" : 2,

        "ok" : 1

}

> db.time_table.find()

{ "_id" : ObjectId("59165fb0f4b792eb4b0b8cbc"), "date" : "2017-03-28", "line_name" : "Hutt Valley Line", "service_no" : 2, "time" : 1045, "distance" : 34.3, "latitude" : -41.2865, "longitude" : 174.7762, "stop" : "Wellington", "driver" : { "driver_name" : "fred", "email" : "fred@ecs.vuw.ac.nz", "password" : "f00f", "mobile" : "2799797", "current_position" : "Wellington", "skill" : [ "Ganz Mavag", "Guliver" ] }, "vehicle" : { "vehicle_id" : "KW3300", "status" : "in_use", "type" : "Matangi" }, "data_point" : [ { "sequence" : "2016-03-28 21:17:40+0000", "longitude" : -41.2012, "latitude" : 175.07, "speed" : 70.1 }, { "sequence" : "2016-03-28 21:07:40+0000", "longitude" : -41.2262, "latitude" : 174.77, "speed" : 69.2 } ] }
```

Create index for time_table

Q3-b)

For example, from Wellington to Upper Hutt, it takes 50 minutes, and the time interval in the data point table is 10 seconds. So there will be 50*60/10=300 instances in the data point entity type.

Q4-a)

```
> db.sailor.distinct("name");
[
        "James",
        "Peter",
        "Milan",
        "Eileen",
        "Paul",
        "Charmain",
        "Gwendolynn"
]
```

Q4-b)

```
> db.sailor.distinct("name", {"skills":{ $all:["row","sail","motor"], $size: 3}})
[ "Peter" ]
```

Q5-a)

```
> db.res_ref.find({"reserves.date":"2017-03-16"}, { _id: 0, marina:1, "reserves.boat":1,
"reserves.sailor":1 })
{ "marina" : "Sea View", "reserves" : { "boat" : 313, "sailor" : 111 } }
{ "marina" : "Port Nicholson", "reserves" : { "boat" : 919, "sailor" : 818 } }
> db.boat.findOne({marina:"Sea View", number:313},{_id:0, name:1});
{ "name" : "Flying Dutch" }
> db.sailor.findOne({sailorId:111},{_id:0, name:1});
{ "name" : "Peter" }
> db.boat.findOne({marina:"Port Nicholson", number:919},{_id:0, name:1});
{ "name" : "Mermaid" }
> db.sailor.findOne({sailorId:818},{_id:0, name:1});
{ "name" : "Milan" }
```

Q5-b)

```
> var curs = db.res_ref.find({"reserves.date":"2017-03-16"}, { _id: 0, marina:1,
"reserves.boat":1, "reserves.sailor":1 })

> while (curs.hasNext()) {

... tmp_res=curs.next();

... tmp_boat=db.boat.findOne({marina:tmp_res.marina,
number:tmp_res.reserves.boat},{_id:0, name:1});

... tmp_sailor=db.sailor.findOne({sailorId:tmp_res.reserves.sailor},{_id:0, name:1});

... ret={marina_name:tmp_res.marina,
boat_name:tmp_boat.name,sailor_name:tmp_sailor.name}

... print(tojson(ret))

... }

{

        "marina_name" : "Sea View",

        "boat_name" : "Flying Dutch",

        "sailor_name" : "Peter"

}

{

        "marina_name" : "Port Nicholson",

        "boat_name" : "Mermaid",

        "sailor_name" : "Milan"

}
```