3. The Professor now checks with the registration team that there are only ECE, CSE, BP, CIVIL, ME students are allowed to enroll in Web Engineering Course. The Professor again asks Anamika to delete all documents except the above mentioned branch students in the collection "StuData".

4. Prof Gabriel wants to register the students of Fall-2019 batch into the Web Engineering Courses for monitoring the student activities. But he likes to store the documents only in alphabetical order. Prof Gabriel asks Anamika to sort the collection "StuData" so that he can get the requisite output. Using appropriate methods sort the documents in the collection "StuData" both in ascending and descending order.

5. Prof Mathew has the students' previous marks through which he wants to arrange some special training for the students. But the data is in two different collections. The first collection is StuData which contains the StudentID, Name, Dept, Address fields whereas the second collection StuMarks contains StudentID and marks in various subjects. Prof Mathew asks his favourite student Mr. Honulul to display all the details of the students from both the collections.

StudentID	Name	Dept	Address
1594	Zudumba	CSE	VZA
1607	Hidamba	CSE	GUN
1642	Jyoto	CSE	VZA
1542	Chapa	CSE	GUN
1668	Pronto	BP	GUN
1587	Geets	ECE	TEN
1654	Anjani	BP	VZA
1540	Miniyo	CIVIL	GUN
1610	Bharg	ME	TEN
1595	Varen	ME	GUN
1677	Soven	ME	VZA
1603	Varshita	ECE	VZA
1592	Chuck	BT	VZA
1500	Rischard	PE	VZA

StudentID	Sub-1	Sub-2	Sub-3	Sub-4
1594	87	100	91	83
1607	95	89	98	97
1642	97	83	87	94
1542	86	86	100	80
1668	95	91	98	89
1587	90	95	84	100
1654	89	100	89	83
1540	83	86	83	81
1610	97	87	87	88
1595	99	95	84	84
1677	98	92	80	94
1603	91	90	99	99
1592	83	90	92	94
1500	81	99	100	94

Quick Fix:

```
1) Var Mongoclient = require('mongodb').Mongoclient;
5) Var url = "mongodb://localhost:27017/";
Mongoclient.connect(url, function(err, db) {
    Var dbname = db.db("kl");
    Var dc = dbname.collection("stuData");
```

```
Const muldata=[{StudentID:1594, name:"ZUDUMBA", dept:"CSE"
 ADDRESS:"VZA"},
{StudentID: 1607, name:"Hidamba", dept:"CSE", address:"GUN"},
{StudentID: 1642, name:"Jyoto", dept:"CSE", address:"VZA"},
{StudentID: 1542, name:"Chapa", dept:"CSE", address:"GUN"
{StudentID: 1668, name:"pronto", dept:"Bp", address:"GUN"},
{StudentID: 1587, name:"Geets", dept:"ECE", address:"TEN"},
{StudentID: 1654, name:"Anjani", dept:"Bp", address:"VZA"},
{StudentID: 1540, name:"Miniyo", dept:"CIVIL", address:"GUN"
{StudentID: 1610, name:"Bharg", dept:"ME", address:"TEN"},
{StudentID: 1595, name:"Varen", dept:"ME", address:"GUN"},
{StudentID: 1677, name:"Soven", dept:"ME", address:"VZA"},
{StudentID: 1603, name:"Vanshita", dept:"ECE", address:"VZA"
{StudentID: 1592, name:"Chuck", dept:"BT", address:"VZA"},
{StudentID: 1500, name:"Rischard", dept:"PE", address:"VZA"
];
dc.insertMany(muldata, function(err, result){
    if(err) throw err;
    Console.log(result);
    db.close();
  });
});
```

2)a) use kl

Switched to db kl

```
db.StuData.findone();
{
    "_id": ObjectId("6162cb1fe49f31d856c581c3"),
    "StudentID":1594,
    "name": "ZUDUMBA",
    "dept": "CSE",
    "ADDRESS": "VZA"
```

b) show dbs

```
db.studata.find().pretty();
{
    "_id" : ObjectId("6162cb1fe49f31d856c581c3"),
    "studentID" : 1594,
    "name" : "ZUDUMDA",
    "dept" : "CSE",
    "ADDRESS" : "VIZA".
}
{
    "_id" : ObjectID("6162cb1fe49f31d856c581c4"),
    "studentID" : 1607,
    "name" : "Hidamba",
    "dept" : "CSE",
    "address" : "GUN"
}
{ "_id" : ObjectID("6162cb1fe49F31d856c5815"),
    "studentID" : 1642,
    "name" : "Jyoto",
    "dept" : "CSE",
    "address" : "VIZA",
}
{ "_id" : ObjectID("6162cb1fe49f31d856c5816"),
    "studentID" : 1542,
    "name" : "chapa",
    "dept" : "CSE",
    "address" : "GUN"
}
```

c) `db.studata.find({}, {studentID:1, dept:1, _id:0});`

```
{"studentID" : 1594, "dept" : "CSE"}
{"studentID" : 1607, "dept" : "CSE"}
{"studentID" : 1642, "dept" : "CSE"}
{"studentID" : 1542, "dept" : "CSE"}
```

{"StudentID" : 1668, "dept" : "CSE"}
{"StudentID": 1587, "dept" : "ECE"}
{"StudentID" : 1654, "dept" : "Bp"}
{"StudentID" : 1540, "dept": "ME"}

d) db.stuData.find{ }, {dept:0, _id:0});

{"StudentID":1594, "name": "ZUDUMBA", "ADDRESS" : "VZA"}
{"studentID": 1607, "name": "Hidamba", "address" : "GUN"}
{"studentID": 1642, "name": "Jyoto", "address": "GUN"}
{"StudentID": 1542, "name": "chapa", "address": "GUN"}
{"studentID": 1668, "name": "pronto", "address": "GUN"}
{"StudentID": 1587, "name": "Geets", "address": "VZA"}

3) db.stuData.remove({dept: {$nin:["ECE", "CSE", "Bp", "CNI"
"ME"]}});
WriteResult({"nRemoved": 2})
db.stuData.find().pretty();

{
"_id": ObjectId("61611c89dcc0d14cd7a66041"),
"StudentID":1594,
"name": "ZUDUMBA",
"dept": "CSE",
"ADDRESS": "VZA"
}
{
"_id": ObjectId("61611c89dcc0d14cd7a66042"),
"StudentID": 1607,
"name": "Hidamba",
"dept": "CSE",
"address": "GUN",
}

```
{
  "_id" : ObjectId("61611c89dcc0d14cd7a66043"),
   "studentID" : 1642,
   "name": "Jyoti",
   "dept" : "CSE",
   "address" : "VZA"
}
```

4) Ascending order

```
db.stuData.find().sort({name:1});
{"_id": ObjectId("61611c89dcc0d14cd7a66047"), "studentID"
:1654, "name": "Anjani", "dept": "Bp", "address" :"VZA"}
{"_id": ObjectId("61611c89dcc0d14cd7a66049"),"studentID"
:1610,"name": "Bharg", "dept": "ME", "address": "TEN"}
{"_id": ObjectId("61611c89dcc0d14cd7a66044"), "studentID"
:1542, "name": "Chapa", "dept": "CSE", "address": "GUN"}
```

5) Descending order.

```
db.stuData.find().sort({name:-1});
{"_id" : ObjectId("61611c89dcc0d14cd7a66041"),"studentID"
:1594,"name": "ZUDUMDA", "dept": "CSE","ADDRESS":"VZA"}
{"_id": ObjectId("61611c89dcc0d14cd7a66042"),"studentID"
: 1603,
```

Exercises:

Q1. Does MongoDB assist primary-key, foreign-key relationship?

MongoDB doesn't Support primary key - foreign key relationship. It has "id field, for each of the every document. MongoDB doesn't work with the Concept of foreign keys.

Q2. What are Indexes in MongoDB?

Indexes are the special data structures that store a small portion of the collections data [Structures] It is used to identify the right file wherever required.

Q3. How many indexes does MongoDB create by way of default for a new collection?

A collection in MongoDB can only have upto 64 Indexes.

Q4. How Does MongoDB database have tables for storing records?

MongoDB data stores the data in the form of collection instead of tables fields in the documents are similar to columns in the tables.

Q5. How does MongoDB offer concurrency?

MongoDB uses multi granularity locking that allows operation to lock at the global database or collection level and allows for individual engines to implement the concurrency.