

### QUESTION:

Design DB model for Guvi Zen class Student Collection Batch Collection Task Collection Mentor Collection

### SOLUTION:

#### In SQL schema format:

1.> Student Collection

```
mysql> create table studentCollection(  
-> studentID int(10) unsigned not null primary key,  
-> studentName varchar(50) not null,  
-> email varchar(50) not null,  
-> courseID int(10) );  
Query OK, 0 rows affected, 2 warnings (0.04 sec)
```

```
mysql> describe studentCollection;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| studentID  | int unsigned  | NO   | PRI | NULL    |       |  
| studentName | varchar(50)   | NO   |     | NULL    |       |  
| email      | varchar(50)   | NO   |     | NULL    |       |  
| courseID   | int           | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

```

mysql> insert into studentCollection value(1205,"Michael Welsh","mwelsh@gmail.com",123458);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1206,"Meghan Curri","mcurri@gmail.com",123457);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1207,"John Baker","jbaker@gmail.com",123457);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1208,"Charlie Cooper","cooper@gmail.com",123457);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1209,"Felix Lopez","flopez@gmail.com",123457);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1210,"Cody Nolan","cnolan@gmail.com",123458);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1211,"baron Kurt","bkurt@gmail.com",123459);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1212,"Susan Sarandon","susans@gmail.com",123459);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1213,"Katie Leung","katiel@gmail.com",123459);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1214,"Katie Larkin","katiela@gmail.com",123460);
Query OK, 1 row affected (0.00 sec)

mysql> insert into studentCollection value(1215,"James Patterson","jamesp@gmail.com",123460);
Query OK, 1 row affected (0.00 sec)

```

```
mysql> select * from studentCollection;
```

studentID	studentName	email	courseID
1201	John Doe	jdoe@gmail.com	123456
1202	Tom Hessel	thessel@gmail.com	123456
1203	Faye Mclure	fayem@gmail.com	123456
1204	Toby Sawayn	tsawayn@gmail.com	123458
1205	Michael Welsh	mwelsh@gmail.com	123458
1206	Meghan Curri	mcurri@gmail.com	123457
1207	John Baker	jbaker@gmail.com	123457
1208	Charlie Cooper	cooper@gmail.com	123457
1209	Felix Lopez	flopez@gmail.com	123457
1210	Cody Nolan	cnolan@gmail.com	123458
1211	baron Kurt	bkurt@gmail.com	123459
1212	Susan Sarandon	susans@gmail.com	123459
1213	Katie Leung	katiel@gmail.com	123459
1214	Katie Larkin	katiela@gmail.com	123460
1215	James Patterson	jamesp@gmail.com	123460

```
15 rows in set (0.00 sec)
```

## 2.> Batch Collection

```
mysql> create table batchCollection(  
  -> batchSize int(10) not null,  
  -> time varchar(20) not null,  
  -> batchSlot varchar(20) not null,  
  -> language varchar(20) );  
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

```
mysql> describe batchCollection;
```

Field	Type	Null	Key	Default	Extra
batchName	varchar(20)	NO		NULL	
batchSize	int	NO		NULL	
time	varchar(20)	NO		NULL	
batchSlot	varchar(20)	NO		NULL	
language	varchar(20)	YES		NULL	

5 rows in set (0.01 sec)

```
mysql> insert into batchCollection value("A21",43,"morning","weekend","english");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into batchCollection value("A31",38,"morning","weekday","english");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into batchCollection value("A22",39,"evening","weekend","tamil");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into batchCollection value("A32",40,"evening","weekday","tamil");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into batchCollection value("B21",41,"evening","weekend","english");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into batchCollection value("B31",36,"evening","weekday","english");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into batchCollection value("B22",37,"morning","weekend","tamil");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into batchCollection value("B32",48,"morning","weekday","tamil");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from batchCollection;
```

batchName	batchSize	time	batchSlot	language
A21	43	morning	weekend	english
A31	38	morning	weekday	english
A22	39	evening	weekend	tamil
A32	40	evening	weekday	tamil
B21	41	evening	weekend	english
B31	36	evening	weekday	english
B22	37	morning	weekend	tamil
B32	48	morning	weekday	tamil

```
8 rows in set (0.00 sec)
```

### 3.> Task Collection

```
mysql> create table taskCollection(
-> taskID int(10) not null primary key,
-> type varchar(20) not null,
-> description varchar(250) not null,
-> batchAssigned varchar(50) not null,
-> course int(20) not null );
Query OK, 0 rows affected, 2 warnings (0.02 sec)
```

```
mysql> describe taskCollection;
```

Field	Type	Null	Key	Default	Extra
taskID	int	NO	PRI	NULL	
type	varchar(20)	NO		NULL	
description	varchar(250)	NO		NULL	
batchAssigned	varchar(50)	NO		NULL	
course	int	NO		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> insert into taskCollection value(8,"frontend","Create a simple form with submit functionality","A21,A32,B32",1201);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(9,"backend","Create a simple database in MongoDB schemaless format with express middleware","A21,A32,B32",1201);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(16,"frontend","Create a simple UI with a dynamic carousel","A*",1204);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(17,"backend","Create a database for a stack overflow clone app","A*,B*",1202);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(18,"backend","Create a database for a gmail clone app","B*",1204);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(19,"frontend","Create a simple UI similar to IMDB web application","A*,B*",1204);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(24,"frontend","Create a simple UI similar to flipkart using react","A*,B*",1207);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(25,"frontend","Create a simple UI similar to netflix using react","A*,B*",1208);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(26,"backend","Create a database for a netflix clone application","A*,B*",1209);
Query OK, 1 row affected (0.00 sec)

mysql> insert into taskCollection value(27,"backend","Create a database for a flipkart clone application","A*,B*",1209);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select *from taskCollection;
+-----+-----+-----+-----+-----+-----+
| taskID | type  | description                                     | batchAssigned | course |
+-----+-----+-----+-----+-----+
| 8      | frontend | Create a simple form with submit functionality | A21,A32,B32  | 1201  |
| 9      | backend  | Create a simple database in MongoDB schemaless format with express middleware | A21,A32,B32  | 1201  |
| 16     | frontend | Create a simple UI with a dynamic carousel    | A*           | 1204  |
| 17     | backend  | Create a database for a stack overflow clone app | A*,B*        | 1202  |
| 18     | backend  | Create a database for a gmail clone app         | B*           | 1204  |
| 19     | frontend | Create a simple UI similar to IMDB web application | A*,B*        | 1204  |
| 24     | frontend | Create a simple UI similar to flipkart using react | A*,B*        | 1207  |
| 25     | frontend | Create a simple UI similar to netflix using react | A*,B*        | 1208  |
| 26     | backend  | Create a database for a netflix clone application | A*,B*        | 1209  |
| 27     | backend  | Create a database for a flipkart clone application | A*,B*        | 1209  |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

#### 4.> Mentor Collection

```
mysql> create table mentorCollection(
-> mentorID int(20) not null primary key,
-> name varchar(50) not null,
-> email varchar(100) not null,
-> batchAssigned varchar(20) not null,
-> course int(20) not null );
Query OK, 0 rows affected, 2 warnings (0.02 sec)
```

```
mysql> describe mentorCollection;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| mentorID   | int           | NO   | PRI | NULL    |       |
| name       | varchar(50)   | NO   |     | NULL    |       |
| email      | varchar(100)  | NO   |     | NULL    |       |
| batchAssigned | varchar(20)   | NO   |     | NULL    |       |
| course     | int           | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into mentorCollection value(6201,"Harry Sunderson","harrys@gmail.com","A22",1206);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6202,"Emilia Anders","eanders@gmail.com","B22",1208);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6203,"Emma Granger","emmag@gmail.com","B22",1208);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6204,"Hannah Abbot","habbot@gmail.com","A22",1206);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6205,"Karina Simper","ksimper@gmail.com","A21",1207);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6207,"hadrian S","hadrian@gmail.com","B21",1207);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6206,"Ali Ahmed","alimd@gmail.com","B32",1209);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6208,"Samarth Karan","samarth@gmail.com","A31",1202);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6209,"Shobhit Kumar","shobhitk@gmail.com","A32",1203);
Query OK, 1 row affected (0.00 sec)

mysql> insert into mentorCollection value(6210,"Samar Kumar","samark@gmail.com","A32",1201);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from mentorCollection;
```

mentorID	name	email	batchAssigned	course
6201	Harry Sunderson	harrys@gmail.com	A22	1206
6202	Emilia Anders	eanders@gmail.com	B22	1208
6203	Emma Granger	emmag@gmail.com	B22	1208
6204	Hannah Abbot	habbot@gmail.com	A22	1206
6205	Karina Simper	ksimper@gmail.com	A21	1207
6206	Ali Ahmed	alimd@gmail.com	B32	1209
6207	hadrian S	hadrian@gmail.com	B21	1207
6208	Samarth Karan	samarth@gmail.com	A31	1202
6209	Shobhit Kumar	shobhitk@gmail.com	A32	1203
6210	Samar Kumar	samark@gmail.com	A32	1201

10 rows in set (0.00 sec)

## In MongoDB database format:

### 1.> Student Collection

```
switched to db guviDB
> db.studentCollection.insert(
... {
... studentID: 1201,
... studentName: "John Doe",
... email: "jdoe@gmail.com",
... courseId: 123456})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.find()
{ "_id" : ObjectId("6244a309ac066c6dfd2b3d"), "studentID" : 1201, "studentName" : "John Doe", "email" : "jdoe@gmail.com", "courseId" : 123456 }
```

```
> db.studentCollection.insert( { studentID: 1202, studentName: "Tom Hessel", email: "thessel@gmail.com", courseId: 123456})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1203, studentName: "Toby Sawayn", email: "tsawayn@gmail.com", courseId: 123458})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1204, studentName: "Faye Mclure", email: "fayem@gmail.com", courseId: 123456})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1205, studentName: "Michael welsh", email: "mwelsh@gmail.com", courseId: 123458})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1206, studentName: "Meghan Curri", email: "mcurri@gmail.com", courseId: 123457})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1207, studentName: "John Baker", email: "jbaker@gmail.com", courseId: 123457})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1208, studentName: "Charlie Cooper", email: "cooper@gmail.com", courseId: 123457})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1209, studentName: "Felix Lopez", email: "flopez@gmail.com", courseId: 123458})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1210, studentName: "Cody Nolan", email: "cnolan@gmail.com", courseId: 123458})
WriteResult({ "nInserted" : 1 })
```

```
> db.studentCollection.insert( { studentID: 1211, studentName: "Baron Kurt", email: "bkurt@gmail.com", courseId: 123459})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1212, studentName: "Susan Sarandon", email: "susans@gmail.com", courseId: 123459})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1213, studentName: "Katie Leung", email: "katiel@gmail.com", courseId: 123459})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1214, studentName: "Katie Larkin", email: "katiela@gmail.com", courseId: 123460})
WriteResult({ "nInserted" : 1 })
> db.studentCollection.insert( { studentID: 1215, studentName: "James Peterson", email: "jamesp@gmail.com", courseId: 123460})
WriteResult({ "nInserted" : 1 })
```

```
> db.studentCollection.find()
{ "_id" : ObjectId("6244a309ac066c6dfd2b3d"), "studentID" : 1201, "studentName" : "John Doe", "email" : "jdoe@gmail.com", "courseId" : 123456 }
{ "_id" : ObjectId("6244a362ac066c6dfd2b3e"), "studentID" : 1202, "studentName" : "Tom Hessel", "email" : "thessel@gmail.com", "courseId" : 123456 }
{ "_id" : ObjectId("6244a38fac066c6dfd2b3f"), "studentID" : 1203, "studentName" : "Toby Sawayn", "email" : "tsawayn@gmail.com", "courseId" : 123458 }
{ "_id" : ObjectId("6244a3e3ac066c6dfd2b40"), "studentID" : 1204, "studentName" : "Faye Mclure", "email" : "fayem@gmail.com", "courseId" : 123456 }
{ "_id" : ObjectId("6244a40dac066c6dfd2b41"), "studentID" : 1205, "studentName" : "Michael welsh", "email" : "mwelsh@gmail.com", "courseId" : 123458 }
{ "_id" : ObjectId("6244a433ac066c6dfd2b42"), "studentID" : 1206, "studentName" : "Meghan Curri", "email" : "mcurri@gmail.com", "courseId" : 123457 }
{ "_id" : ObjectId("6244a452ac066c6dfd2b43"), "studentID" : 1207, "studentName" : "John Baker", "email" : "jbaker@gmail.com", "courseId" : 123457 }
{ "_id" : ObjectId("6244a4f1ac066c6dfd2b44"), "studentID" : 1208, "studentName" : "Charlie Cooper", "email" : "cooper@gmail.com", "courseId" : 123457 }
{ "_id" : ObjectId("6244a52aac066c6dfd2b45"), "studentID" : 1209, "studentName" : "Felix Lopez", "email" : "flopez@gmail.com", "courseId" : 123458 }
{ "_id" : ObjectId("6244a54fac066c6dfd2b46"), "studentID" : 1210, "studentName" : "Cody Nolan", "email" : "cnolan@gmail.com", "courseId" : 123458 }
{ "_id" : ObjectId("6253ee85ac066c6dfd2b59"), "studentID" : 1211, "studentName" : "Baron Kurt", "email" : "bkurt@gmail.com", "courseId" : 123459 }
{ "_id" : ObjectId("6253ee82ac066c6dfd2b5a"), "studentID" : 1212, "studentName" : "Susan Sarandon", "email" : "susans@gmail.com", "courseId" : 123459 }
{ "_id" : ObjectId("6253eeb8ac066c6dfd2b5b"), "studentID" : 1213, "studentName" : "Katie Leung", "email" : "katiel@gmail.com", "courseId" : 123459 }
{ "_id" : ObjectId("6253eeddac066c6dfd2b5c"), "studentID" : 1214, "studentName" : "Katie Larkin", "email" : "katiela@gmail.com", "courseId" : 123460 }
{ "_id" : ObjectId("6253eef5ac066c6dfd2b5d"), "studentID" : 1215, "studentName" : "James Peterson", "email" : "jamesp@gmail.com", "courseId" : 123460 }
>
```

```
> db.studentCollection.find().pretty()
{
  "_id" : ObjectId("6244a309ac066c6dfdad2b3d"),
  "studentID" : 1201,
  "studentName" : "John Doe",
  "email" : "jdoe@gmail.com",
  "courseId" : 123456
}
{
  "_id" : ObjectId("6244a362ac066c6dfdad2b3e"),
  "studentID" : 1202,
  "studentName" : "Tom Hessel",
  "email" : "thessel@gmail.com",
  "courseId" : 123456
}
{
  "_id" : ObjectId("6244a38fac066c6dfdad2b3f"),
  "studentID" : 1203,
  "studentName" : "Toby Sawayn",
  "email" : "tsawayn@gmail.com",
  "courseId" : 123458
}
{
  "_id" : ObjectId("6244a3e3ac066c6dfdad2b40"),
  "studentID" : 1204,
  "studentName" : "Faye Mclure",
  "email" : "fayem@gmail.com",
  "courseId" : 123456
}
{
  "_id" : ObjectId("6244a40dac066c6dfdad2b41"),
  "studentID" : 1205,
  "studentName" : "Michael welsh",
  "email" : "mwelsh@gmail.com",
  "courseId" : 123458
}
{
  "_id" : ObjectId("6244a433ac066c6dfdad2b42"),
  "studentID" : 1206,
  "studentName" : "Meghan Curri",
  "email" : "mcurri@gmail.com",
  "courseId" : 123457
}
}
```



```
{
  "_id" : ObjectId("6244a40dac066c6dfdad2b41"),
  "studentID" : 1205,
  "studentName" : "Michael welsh",
  "email" : "mwelsh@gmail.com",
  "courseId" : 123458
}
{
  "_id" : ObjectId("6244a433ac066c6dfdad2b42"),
  "studentID" : 1206,
  "studentName" : "Meghan Curri",
  "email" : "mcurri@gmail.com",
  "courseId" : 123457
}
{
  "_id" : ObjectId("6244a452ac066c6dfdad2b43"),
  "studentID" : 1207,
  "studentName" : "John Baker",
  "email" : "jbaker@gmail.com",
  "courseId" : 123457
}
{
  "_id" : ObjectId("6244a4f1ac066c6dfdad2b44"),
  "studentID" : 1208,
  "studentName" : "Charlie Cooper",
  "email" : "cooper@gmail.com",
  "courseId" : 123457
}
{
  "_id" : ObjectId("6244a52aac066c6dfdad2b45"),
  "studentID" : 1209,
  "studentName" : "Felix Lopez",
  "email" : "flopez@gmail.com",
  "courseId" : 123458
}
{
  "_id" : ObjectId("6244a54fac066c6dfdad2b46"),
  "studentID" : 1210,
  "studentName" : "Cody Nolan",
  "email" : "cnolan@gmail.com",
  "courseId" : 123458
}
```

## 2.> Batch Collection

```
> db.batchCollection.insert(
... {
...  batchName: "A21",
...  batchSize: 43,
...  time: "morning",
...  batchSlot: "weekend",
...  language: "english"})
WriteResult({ "nInserted" : 1 })
```

```
> db.batchCollection.insert({ batchName: "A31", batchSize: 38, time: "morning", batchSlot: "weekday", language: "english"})
WriteResult({ "nInserted" : 1 })
> db.batchCollection.insert({ batchName: "A22", batchSize: 39, time: "evening", batchSlot: "weekend", language: "tamil"})
WriteResult({ "nInserted" : 1 })
> db.batchCollection.insert({ batchName: "A32", batchSize: 40, time: "evening", batchSlot: "weekday", language: "tamil"})
WriteResult({ "nInserted" : 1 })
> db.batchCollection.find()
{ "_id" : ObjectId("6244a647ac066c6dfdad2b47"), "batchName" : "A21", "batchSize" : 43, "time" : "morning", "batchSlot" : "weekend", "language" : "english" }
{ "_id" : ObjectId("6244a666ac066c6dfdad2b48"), "batchName" : "A31", "batchSize" : 38, "time" : "morning", "batchSlot" : "weekday", "language" : "english" }
{ "_id" : ObjectId("6244a699ac066c6dfdad2b49"), "batchName" : "A22", "batchSize" : 39, "time" : "evening", "batchSlot" : "weekend", "language" : "tamil" }
{ "_id" : ObjectId("6244a6a8ac066c6dfdad2b4a"), "batchName" : "A32", "batchSize" : 40, "time" : "evening", "batchSlot" : "weekday", "language" : "tamil" }
> db.batchCollection.insert({ batchName: "B21", batchSize: 41, time: "evening", batchSlot: "weekend", language: "english"})
WriteResult({ "nInserted" : 1 })
> db.batchCollection.insert({ batchName: "B31", batchSize: 36, time: "evening", batchSlot: "weekday", language: "english"})
WriteResult({ "nInserted" : 1 })
> db.batchCollection.insert({ batchName: "B22", batchSize: 37, time: "morning", batchSlot: "weekend", language: "tamil"})
WriteResult({ "nInserted" : 1 })
> db.batchCollection.insert({ batchName: "B32", batchSize: 48, time: "morning", batchSlot: "weekday", language: "tamil"})
WriteResult({ "nInserted" : 1 })
```

```
> db.batchCollection.find()
{ "_id" : ObjectId("6244a647ac066c6dfdad2b47"), "batchName" : "A21", "batchSize" : 43, "time" : "morning", "batchSlot" : "weekend", "language" : "english" }
{ "_id" : ObjectId("6244a666ac066c6dfdad2b48"), "batchName" : "A31", "batchSize" : 38, "time" : "morning", "batchSlot" : "weekday", "language" : "english" }
{ "_id" : ObjectId("6244a699ac066c6dfdad2b49"), "batchName" : "A22", "batchSize" : 39, "time" : "evening", "batchSlot" : "weekend", "language" : "tamil" }
{ "_id" : ObjectId("6244a6a8ac066c6dfdad2b4a"), "batchName" : "A32", "batchSize" : 40, "time" : "evening", "batchSlot" : "weekday", "language" : "tamil" }
{ "_id" : ObjectId("6244a6f4ac066c6dfdad2b4b"), "batchName" : "B21", "batchSize" : 41, "time" : "evening", "batchSlot" : "weekend", "language" : "english" }
{ "_id" : ObjectId("6244a70eac066c6dfdad2b4c"), "batchName" : "B31", "batchSize" : 36, "time" : "evening", "batchSlot" : "weekday", "language" : "english" }
{ "_id" : ObjectId("6244a728ac066c6dfdad2b4d"), "batchName" : "B22", "batchSize" : 37, "time" : "morning", "batchSlot" : "weekend", "language" : "tamil" }
{ "_id" : ObjectId("6244a73fac066c6dfdad2b4e"), "batchName" : "B32", "batchSize" : 48, "time" : "morning", "batchSlot" : "weekday", "language" : "tamil" }
```

```
> db.batchCollection.find().pretty()
{
  "_id" : ObjectId("6244a647ac066c6dfdad2b47"),
  "batchName" : "A21",
  "batchSize" : 43,
  "time" : "morning",
  "batchSlot" : "weekend",
  "language" : "english"
}
{
  "_id" : ObjectId("6244a666ac066c6dfdad2b48"),
  "batchName" : "A31",
  "batchSize" : 38,
  "time" : "morning",
  "batchSlot" : "weekday",
  "language" : "english"
}
{
  "_id" : ObjectId("6244a699ac066c6dfdad2b49"),
  "batchName" : "A22",
  "batchSize" : 39,
  "time" : "evening",
  "batchSlot" : "weekend",
  "language" : "tamil"
}
{
  "_id" : ObjectId("6244a6a8ac066c6dfdad2b4a"),
  "batchName" : "A32",
  "batchSize" : 40,
  "time" : "evening",
  "batchSlot" : "weekday",
  "language" : "tamil"
}
{
  "_id" : ObjectId("6244a6f4ac066c6dfdad2b4b"),
  "batchName" : "B21",
  "batchSize" : 41,
  "time" : "evening",
  "batchSlot" : "weekend",
  "language" : "english"
}
{
  "_id" : ObjectId("6244a70eac066c6dfdad2b4c"),
  "batchName" : "B31",
  "batchSize" : 36,
  "time" : "evening",
  "batchSlot" : "weekday",
  "language" : "english"
}
```

```
{
  "_id" : ObjectId("6244a6a8ac066c6dfdada2b4a"),
  "batchName" : "A32",
  "batchSize" : 40,
  "time" : "evening",
  "batchSlot" : "weekday",
  "language" : "tamil"
}
{
  "_id" : ObjectId("6244a6f4ac066c6dfdada2b4b"),
  "batchName" : "B21",
  "batchSize" : 41,
  "time" : "evening",
  "batchSlot" : "weekend",
  "language" : "english"
}
{
  "_id" : ObjectId("6244a70eac066c6dfdada2b4c"),
  "batchName" : "B31",
  "batchSize" : 36,
  "time" : "evening",
  "batchSlot" : "weekday",
  "language" : "english"
}
{
  "_id" : ObjectId("6244a728ac066c6dfdada2b4d"),
  "batchName" : "B22",
  "batchSize" : 37,
  "time" : "morning",
  "batchSlot" : "weekend",
  "language" : "tamil"
}
{
  "_id" : ObjectId("6244a73fac066c6dfdada2b4e"),
  "batchName" : "B32",
  "batchSize" : 48,
  "time" : "morning",
  "batchSlot" : "weekday",
  "language" : "tamil"
}
```

### 3.> Task Collection

```
> db.taskCollection.insert( { taskId: 010,type:"frontend", description: "Create a simple form with submit functionality", batchAssigned: "A21,A32,B32", course: 1201})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 011,type:"backend", description: "Create a simple database in MongoDB schemaless format with express middleware", batchAssigned: "A21,A32,B32", course: 1201})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 020,type:"frontend", description: "Create a simple UI with a moving carousel", batchAssigned: "A", "B", course: "1202,1203,1204" }
... ^C

> db.taskCollection.insert( { taskId: 020,type:"backend", description: "Create a simple UI with a moving carousel", batchAssigned: "A",B", course: "1201,1202,1204"})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 021,type:"backend", description: "Create a database for a stack overflow clone app", batchAssigned: "A", course: "1201,1202,1204"})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 022,type:"backend", description: "Create a database for a gmail clone app", batchAssigned: "B", course: "1201,1202,1204"})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 023,type:"frontend", description: "Create a simple UI similar to IMDb web application", batchAssigned: "A",B", course: "1201,1202,1204"})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 030,type:"frontend", description: "Create a simple UI similar to Flipkart using react", batchAssigned: "A",B", course: "1207,1208,1209"})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 031,type:"frontend", description: "Create a simple UI similar to netflix using react", batchAssigned: "A",B", course: "1207,1208,1209"})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 032,type:"backend", description: "Create a database fro a netflix clone application", batchAssigned: "A", course: "1207,1208,1209"})
WriteResult({ "nInserted" : 1 })
> db.taskCollection.insert( { taskId: 033,type:"backend", description: "Create a database for a flipkart clone application", batchAssigned: "B", course: "1207,1208,1209"})
WriteResult({ "nInserted" : 1 })

> db.taskCollection.find()
{ "_id" : ObjectId("6253f401ac066c6dfdad2b5e"), "taskId" : 8, "type" : "frontend", "description" : "Create a simple form with submit functionality", "batchAssigned" : "A21,A32,B32", "course" : 1201 }
{ "_id" : ObjectId("6253f44aac066c6dfdad2b5f"), "taskId" : 9, "type" : "backend", "description" : "Create a simple database in MongoDB schemaless format with express middleware", "batchAssigned" : "A21,A32,B32", "course" : 1201 }
{ "_id" : ObjectId("6253f6b7ac066c6dfdad2b60"), "taskId" : 16, "type" : "backend", "description" : "Create a simple UI with a moving carousel", "batchAssigned" : "A",B", "course" : "1201,1202,1204" }
{ "_id" : ObjectId("6253f831ac066c6dfdad2b61"), "taskId" : 17, "type" : "backend", "description" : "Create a database for a stack overflow clone app", "batchAssigned" : "A", "course" : "1201,1202,1204" }
{ "_id" : ObjectId("6253f851ac066c6dfdad2b62"), "taskId" : 18, "type" : "backend", "description" : "Create a database for a gmail clone app", "batchAssigned" : "B", "course" : "1201,1202,1204" }
{ "_id" : ObjectId("6253f87dac066c6dfdad2b63"), "taskId" : 19, "type" : "frontend", "description" : "Create a simple UI similar to IMDb web application", "batchAssigned" : "A",B", "course" : "1201,1202,1204" }
{ "_id" : ObjectId("6253f8b1ac066c6dfdad2b64"), "taskId" : 24, "type" : "frontend", "description" : "Create a simple UI similar to Flipkart using react", "batchAssigned" : "A",B", "course" : "1207,1208,1209" }
{ "_id" : ObjectId("6253f8c8ac066c6dfdad2b65"), "taskId" : 25, "type" : "frontend", "description" : "Create a simple UI similar to netflix using react", "batchAssigned" : "A",B", "course" : "1207,1208,1209" }
{ "_id" : ObjectId("6253f8edac066c6dfdad2b66"), "taskId" : 26, "type" : "backend", "description" : "Create a database fro a netflix clone application", "batchAssigned" : "A", "course" : "1207,1208,1209" }
{ "_id" : ObjectId("6253f912ac066c6dfdad2b67"), "taskId" : 27, "type" : "backend", "description" : "Create a database for a flipkart clone application", "batchAssigned" : "B", "course" : "1207,1208,1209" }
>
```

```
> db.taskCollection.find().pretty()
{
  "_id" : ObjectId("6253f401ac066c6dfdada2b5e"),
  "taskId" : 8,
  "type" : "frontend",
  "description" : "Create a simple form with submit functionality",
  "batchAssigned" : "A21,A32,B32",
  "course" : 1201
}
{
  "_id" : ObjectId("6253f44aac066c6dfdada2b5f"),
  "taskId" : 9,
  "type" : "backend",
  "description" : "Create a simple database in MongoDB schemaless format with express middleware",
  "batchAssigned" : "A21,A32,B32",
  "course" : 1201
}
{
  "_id" : ObjectId("6253f6b7ac066c6dfdada2b60"),
  "taskId" : 16,
  "type" : "backend",
  "description" : "Create a simple UI with a moving carousel",
  "batchAssigned" : "A*,B*",
  "course" : "1201,1202,1204"
}
{
  "_id" : ObjectId("6253f831ac066c6dfdada2b61"),
  "taskId" : 17,
  "type" : "backend",
  "description" : "Create a database for a stack overflow clone app",
  "batchAssigned" : "A*",
  "course" : "1201,1202,1204"
}
{
  "_id" : ObjectId("6253f851ac066c6dfdada2b62"),
  "taskId" : 18,
  "type" : "backend",
  "description" : "Create a database for a gmail clone app",
  "batchAssigned" : "B*",
  "course" : "1201,1202,1204"
}
{
  "_id" : ObjectId("6253f87dac066c6dfdada2b63"),
  "taskId" : 19,
  "type" : "frontend",
  "description" : "Create a simple UI similar to IMDB web application",
  "batchAssigned" : "A*,B*",
  "course" : "1201,1202,1204"
}
```

```

{
  "_id" : ObjectId("6253f831ac066c6dfdada2b61"),
  "taskId" : 17,
  "type" : "backend",
  "description" : "Create a database for a stack overflow clone app",
  "batchAssigned" : "A*",
  "course" : "1201,1202,1204"
}
{
  "_id" : ObjectId("6253f851ac066c6dfdada2b62"),
  "taskId" : 18,
  "type" : "backend",
  "description" : "Create a database for a gmail clone app",
  "batchAssigned" : "B*",
  "course" : "1201,1202,1204"
}
{
  "_id" : ObjectId("6253f87dac066c6dfdada2b63"),
  "taskId" : 19,
  "type" : "frontend",
  "description" : "Create a simple UI similar to IMDB web application",
  "batchAssigned" : "A*,B*",
  "course" : "1201,1202,1204"
}
{
  "_id" : ObjectId("6253f8b1ac066c6dfdada2b64"),
  "taskId" : 24,
  "type" : "frontend",
  "description" : "Create a simple UI similar to Flipkart using react",
  "batchAssigned" : "A*,B*",
  "course" : "1207,1208,1209"
}
{
  "_id" : ObjectId("6253f8c8ac066c6dfdada2b65"),
  "taskId" : 25,
  "type" : "frontend",
  "description" : "Create a simple UI similar to netflix using react",
  "batchAssigned" : "A*,B*",
  "course" : "1207,1208,1209"
}
{
  "_id" : ObjectId("6253f8edac066c6dfdada2b66"),
  "taskId" : 26,
  "type" : "backend",
  "description" : "Create a database fro a netflix clone application",
  "batchAssigned" : "A*",
  "course" : "1207,1208,1209"
}
{
  "_id" : ObjectId("6253f912ac066c6dfdada2b67"),
  "taskId" : 27,
  "type" : "backend",
  "description" : "Create a database for a flipkart clone application",
  "batchAssigned" : "B*",
  "course" : "1207,1208,1209"
}
>

```

## 4.> Mentor Collection

```
> db.mentorCollection.insert( { guviId: 6201, name: "Harry Sunderson", email: "harrys@gmail.com",
... batchAssigned: "A22",
... course: 1206 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.find(
{ "_id" : ObjectId("6244a9a0ac066c6dfdad2b4f"), "guviId" : 6201, "name" : "Harry Sunderson", "email" : "harrys@gmail.com", "batchAssigned" : "A22", "course" : 1206 }
)
{ "_id" : ObjectId("6244a9a0ac066c6dfdad2b4f"), "guviId" : 6201, "name" : "Harry Sunderson", "email" : "harrys@gmail.com", "batchAssigned" : "A22", "course" : 1206 }

> db.mentorCollection.insert( { guviId: 6202, name: "Emilia Anders", email: "eanders@gmail.com", batchAssigned: "B22", course: 1208 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6203, name: "Emma Granger", email: "emmag@gmail.com", batchAssigned: "B22", course: 1208 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6204, name: "Hannah Abbot", email: "habbot@gmail.com", batchAssigned: "A22", course: 1206 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6205, name: "Karina Simper", email: "ksimper@gmail.com", batchAssigned: "A21", course: 1207 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6206, name: "Hadrian S", email: "hadrian@gmail.com", batchAssigned: "B21", course: 1207 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6207, name: "Ali Ahmed", email: "alimd@gmail.com", batchAssigned: "B32", course: 1209 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6208, name: "Samarth Karan", email: "samarth@gmail.com", batchAssigned: "A31", course: 1202 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6209, name: "Shobit Kumar", email: "shobitk@gmail.com", batchAssigned: "A32", course: 1203 })
WriteResult({ "nInserted" : 1 })
> db.mentorCollection.insert( { guviId: 6210, name: "Samar Kumar", email: "samark@gmail.com", batchAssigned: "A32", course: 1201 })
WriteResult({ "nInserted" : 1 })

> db.mentorCollection.find(
{ "_id" : ObjectId("6244a9a0ac066c6dfdad2b4f"), "guviId" : 6201, "name" : "Harry Sunderson", "email" : "harrys@gmail.com", "batchAssigned" : "A22", "course" : 1206 }
{ "_id" : ObjectId("624543a6ac066c6dfdad2b50"), "guviId" : 6202, "name" : "Emilia Anders", "email" : "eanders@gmail.com", "batchAssigned" : "B22", "course" : 1208 }
{ "_id" : ObjectId("62454df7ac066c6dfdad2b51"), "guviId" : 6203, "name" : "Emma Granger", "email" : "emmag@gmail.com", "batchAssigned" : "B22", "course" : 1208 }
{ "_id" : ObjectId("62454e2dac066c6dfdad2b52"), "guviId" : 6204, "name" : "Hannah Abbot", "email" : "habbot@gmail.com", "batchAssigned" : "A22", "course" : 1206 }
{ "_id" : ObjectId("62454e64ac066c6dfdad2b53"), "guviId" : 6205, "name" : "Karina Simper", "email" : "ksimper@gmail.com", "batchAssigned" : "A21", "course" : 1207 }
{ "_id" : ObjectId("6245571bac066c6dfdad2b54"), "guviId" : 6206, "name" : "Hadrian S", "email" : "hadrian@gmail.com", "batchAssigned" : "B21", "course" : 1207 }
{ "_id" : ObjectId("6245635fac066c6dfdad2b55"), "guviId" : 6207, "name" : "Ali Ahmed", "email" : "alimd@gmail.com", "batchAssigned" : "B32", "course" : 1209 }
{ "_id" : ObjectId("624563f3ac066c6dfdad2b56"), "guviId" : 6208, "name" : "Samarth Karan", "email" : "samarth@gmail.com", "batchAssigned" : "A31", "course" : 1202 }
{ "_id" : ObjectId("6245641dac066c6dfdad2b57"), "guviId" : 6209, "name" : "Shobit Kumar", "email" : "shobitk@gmail.com", "batchAssigned" : "A32", "course" : 1203 }
{ "_id" : ObjectId("62456697ac066c6dfdad2b58"), "guviId" : 6210, "name" : "Samar Kumar", "email" : "samark@gmail.com", "batchAssigned" : "A32", "course" : 1201 }
}
```



```
> db.mentorCollection.find().pretty()
{
  "_id" : ObjectId("6244a9a0ac066c6dfdad2b4f"),
  "guviId" : 6201,
  "name" : "Harry Sunderson",
  "email" : "harrys@gmail.com",
  "batchAssigned" : "A22",
  "course" : 1206
}
{
  "_id" : ObjectId("624543a6ac066c6dfdad2b50"),
  "guviId" : 6202,
  "name" : "Emilia Anders",
  "email" : "eanders@gmail.com",
  "batchAssigned" : "B22",
  "course" : 1208
}
{
  "_id" : ObjectId("62454df7ac066c6dfdad2b51"),
  "guviId" : 6203,
  "name" : "Emma Granger",
  "email" : "emmag@gmail.com",
  "batchAssigned" : "B22",
  "course" : 1208
}
{
  "_id" : ObjectId("62454e2dac066c6dfdad2b52"),
  "guviId" : 6204,
  "name" : "Hannah Abbot",
  "email" : "habbot@gmail.com",
  "batchAssigned" : "A22",
  "course" : 1206
}
{
  "_id" : ObjectId("62454e64ac066c6dfdad2b53"),
  "guviId" : 6205,
  "name" : "Karina Simper",
  "email" : "ksimper@gmail.com",
  "batchAssigned" : "A21",
  "course" : 1207
}
{
  "_id" : ObjectId("6245571bac066c6dfdad2b54"),
  "guviId" : 6206,
  "name" : "Hadrian S",
  "email" : "hadrian@gmail.com",
  "batchAssigned" : "B21",
  "course" : 1207
}
{
  "_id" : ObjectId("6245635fac066c6dfdad2b55"),
  "guviId" : 6207,
  "name" : "Ali Ahmed",
  "email" : "alimd@gmail.com",
  "batchAssigned" : "B32",
  "course" : 1209
}
```

```
{
  "_id" : ObjectId("624563f3ac066c6dfdad2b56"),
  "guviId" : 6208,
  "name" : "Samarth Karan",
  "email" : "samarth@gmail.com",
  "batchAssigned" : "A31",
  "course" : 1202
}
{
  "_id" : ObjectId("6245641dac066c6dfdad2b57"),
  "guviId" : 6209,
  "name" : "Shobit Kumar",
  "email" : "shobitk@gmail.com",
  "batchAssigned" : "A32",
  "course" : 1203
}
{
  "_id" : ObjectId("62456697ac066c6dfdad2b58"),
  "guviId" : 6210,
  "name" : "Samar Kumar",
  "email" : "samark@gmail.com",
  "batchAssigned" : "A32",
  "course" : 1201
}
> _
```