

Create COMPANY DATABASE

Collections: 1. EMPLOYEE [id, First Name, DNumber, Age, Title]
2. DEPARTMENT [Dnumber, Name, Location]

use Company

db.createCollection("Employee");

db.createCollection("Department");

show collections

1. Insert 5 documents in each collection using all 3 methods of insertion.

db.Employee.insert({_id:1, FirstName:"Nam",
DNumber:1001, Age:21, Title:"Manager"});

db.Employee.update({_id:2, FirstName:"Amy",
DNumber:2001, Age:45, Title:"Clerk"},
{upsert:true});

db.Employee.update({_id:3, FirstName:"Penny",
DNumber:3001, Age:30, Title:"SE"},
{upsert:false});

db.Employee.save({FirstName:"Leo", DNumber:1001,
Age:25, Title:"SDET"});

db.Department.insert({Dnumber:1001, Name:"R&D"});

db.Department.update({Dnumber:2001, Name:"IT"},
{upsert:true});

db.Department.save({Dnumber:3001, Name:"HR"});

2. Update Employee document to add new field.

```
db.Employee.update ( { -id: 1 }, { $set: { Sex: "Female" } } );
```

3. Remove a field from an existing document

```
db.Employee.update ( { -id: 1 }, { $unset: { Sex: "Female" } } );
```

4. Select all documents

```
db.Employee.find ( {} );
```

```
db.Department.find ( {} );
```

5. Select only employee name & department no. whose department no. falls b/w 1001 to 1005

```
db.Employee.find ( { DNumber: { "$gt": 1000, "$lt": 1005 } }, { Name: true, DNumber: true } );
```

6. Select Employee whose name begins with "A"

```
db.Employee.find ( { Name: { $regex: "A" } } );
```

7. Select Employee document whose age is greater than 30

```
db.Employee.find ( { Age: { "$gt": 30 } } );
```