**Data engineering**

**Update using \_id field**

db.Employees.updateOne({\_id:1},{$set:{age:24}});

**update using other field apart from \_id**

db.Employees.updateMany({city:"Bangalore"},{$set:{city:"Bengaluru"}});

**delete using \_id field**

db.Employees.deleteOne({\_id:6});

delete using any other field apart from \_id

**db.Employees.deleteMany({city:”Bangalore”});**

if fields can be array

sid,sname,age,technologies

1,Raj,21,C,C++,Java

PK

FK

db.StudentInfo.insertOne({\_id:100,sname:"Raj",age:21,skillSet:["Java","Python"]});

adding new value to add fields

db.StudentInfo.updateOne({\_id:100},{$push:{skillSet:"ML"}});

to remove last property we use $pop

db.StudentInfo.updateOne({\_id:100},{$pop:{skillSet:1}});

to filter array values

db.StudentInfo.find({skillSet:"JS"});

In mongo db we can achieve relationship using

1. Embedded style : we are storing all document in single collection
2. Linking style : storing more than one collection.

db.EmployeeInfo.insertMany([

{\_id:100,name:"Steven",age:21,salary:45000,address:{city:"Bangalore",state:"Kar"},projects:[

{pid:1111,tech:"Java"},

{pid:222,tech:"Python"}

]},

{\_id:101,name:"john",age:24,salary:48000,address:[{city:"Bangalore",state:"Kar"},{city:"Mysore",state:"kar"}],

projects:[

{pid:1111,tech:"Java"}

]}

]);

Trainer

Student

Linked style : more than one collections

**Student hold trainer details**

db.Trainer1.insertMany([

{\_id:100,tname:"John",tech:"AI"},

{\_id:101,tname:"Lex",tech:"Python"}

]);

db.Trainer1.findOne({\_id:100});

db.Student1.insertMany(

[

{\_id:1,sname:"Reeta",age:21,tid:db.Trainer1.findOne({\_id:100})},

{\_id:2,sname:"Veeta",age:22,tid:db.Trainer1.findOne({\_id:100})},

{\_id:3,sname:"Meeta",age:23,tid:db.Trainer1.findOne({\_id:101})},

{\_id:4,sname:"Leeta",age:24,tid:[db.Trainer1.findOne({\_id:100}),db.Trainer1.findOne({\_id:101})]}

]);

**Trainer hold student details**

db.Student2.insertMany(

[

{\_id:1,sname:"Reeta",age:21},

{\_id:2,sname:"Veeta",age:22},

{\_id:3,sname:"Meeta",age:23},

{\_id:4,sname:"Leeta",age:24}

]);

db.Trainer2.insertMany([

{\_id:100,tname:"John",tech:"AI",students:[db.Student2.findOne({\_id:1}),db.Student2.findOne({\_id:2}),db.Student2.findOne({\_id:4})]},

{\_id:101,tname:"Lex",tech:"Python",students:[db.Student2.findOne({\_id:3}),db.Student2.findOne({\_id:4})]}

]);

**Aggregate function**

Which help to retrieve document from one or more than one collection and generate the result.

db.Trainer3.insertMany([

{\_id:100,tname:"John",tech:"AI"},

{\_id:101,tname:"Lex",tech:"Python"}

]);

db.Student3.insertMany(

[

{\_id:1,sname:"Reeta",age:21,tid:100},

{\_id:2,sname:"Veeta",age:22,tid:100},

{\_id:3,sname:"Meeta",age:23,tid:101},

{\_id:4,sname:"Leeta",age:24,tid:[100,101]}

]);

db.Student3.aggregate([{

$lookup:{

from:"Trainer3",

localField:"tid",

foreignField:"\_id",

as:"Trainer"

}

}]);

db.Trainer3.aggregate([{

$lookup:{

from:"Student3",

localField:"\_id",

foreignField:"tid",

as:"Student"

}

}]);

Creating group with repeating values

**db.Employees.aggregate([{$group:{\_id:"$city"}}]);**

creating group and apply sum function on those groups

db.Employees.aggregate([{$group:{\_id:"$city",totalSalary:{$sum:"$salary"}}}]);

db.Employees.aggregate([{$group:{\_id:"$city",maxSalary:{$max:"$salary"}}}]);

db.Employees.aggregate([{$group:{\_id:"$city",minSalary:{$min:"$salary"}}}]);

db.Employees.aggregate([{$group:{\_id:"$city",avgSalary:{$avg:"$salary"}}}]);