**Day 6: 30 Mar. 25**

**Mongo Db Database**

**Mongo DB Relationship**

In RDBMS database we use primary key and foreign key concept to connect the database.

In Mongo DB we create relationship on document level rather than collection level.

In Mongo DB we create the relationship using

1. Embedded style

We are storing records or document in single collection.

db.Employee.insertMany(

[

{\_id:100,name:"Steven",age:21,salary:45000,address:{city:"Bangalore",state:"Kar"},projects:[{pid:1,tech:"Python"},{pid:2,tech:"Node JS"}]},

{\_id:101,name:"Lex",age:22,salary:42000,address:[{city:"Bangalore",state:"Kar"},{city:"Mumbai",state:"Mh"}],

projects:[{pid:1,tech:"Python"},{pid:2,tech:"Node JS"},{pid:3,tech:"Java"}]},

{\_id:102,name:"John",age:28,salary:48000,address:{city:"Bangalore",state:"Kar"}}

]);

Condition apply for complex property

db.Employee.find({"projects.tech":"Java"});

1. Linking style

We can store the document in more than one collection and retrieve those document using aggregate function it is like a Join.

Trainer

PK

**\_id** TName tech

100 Steven Java

101 Lex Python

102 John Node JS

db.Trainer.insertOne({\_id:100,tname:"Steven",tech:"Java"});

db.Trainer.insertOne({\_id:101,tname:"Lex",tech:"Python"});

db.Trainer.insertOne({\_id:102,tname:"John",tech:"Node JS"});

**db.Trainer.findOne({\_id:100}).\_id**

**db.Trainer.findOne({\_id:100}).\_tname**

**db.Trainer.findOne({\_id:100}).\_tech**

**above query provide particular field value not key-value pairs.**

**Student1**

PK

**\_id** SName age **TId**

1 Reeta 21 100

2 Meeta 22 100

3 Leeta 23 101

4 Veeta 24

4 Teeta 25 102,100

db.Student1.insertOne({\_id:1,sname:"Reeta",age:21,tid:db.Trainer.findOne({\_id:100}).\_id});

db.Student1.insertOne({\_id:2,sname:"Meeta",age:222,tid:db.Trainer.findOne({\_id:100}).\_id});

db.Student1.insertOne({\_id:3,sname:"Leeta",age:24,tid:db.Trainer.findOne({\_id:101}).\_id});

db.Student1.insertOne({\_id:3,sname:"Leeta",age:24,tid:db.Trainer.findOne({\_id:101}).\_id}); error

db.Student1.insertOne({\_id:4,sname:"Veeta",age:25});

db.Student1.insertOne({\_id:5,sname:"Teeta",age:26,tid:[db.Trainer.findOne({\_id:100}).\_id,db.Trainer.findOne({\_id:102}).\_id]});

Student2 collection

db.Student2.insertOne({\_id:1,sname:"Reeta",age:21,tid:db.Trainer.findOne({\_id:100})});

db.Student2.insertOne({\_id:2,sname:"Meeta",age:222,tid:db.Trainer.findOne({\_id:100})});

db.Student2.insertOne({\_id:3,sname:"Leeta",age:24,tid:db.Trainer.findOne({\_id:101})});

db.Student2.insertOne({\_id:3,sname:"Leeta",age:24,tid:db.Trainer.findOne({\_id:101})}); error

db.Student2.insertOne({\_id:4,sname:"Veeta",age:25});

db.Student2.insertOne({\_id:5,sname:"Teeta",age:26,tid:[db.Trainer.findOne({\_id:100}),db.Trainer.findOne({\_id:102})]});

aggregate function which help to do maths operations.

Total sum of salary for all documents

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

total sum of salary for all document group by depit

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

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

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

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

**Aggregate with $lookup operator**

**This query is like a Join concept.**

$lookup operator use to retrieve more than one document information form a more than one collection with command field like join.

**db.Student1.aggregate([{$lookup:{from:"Trainer",localField:"tid",foreignField:"\_id",as:"Trainer"}}]);**

**db.Trainer.aggregate([{$lookup:{from:"Student1",localField:"\_id",foreignField:"tid",as:"Student"}}]);**