**Coding Assessment – Mongodb (Relationships in MongoDB)**

**By Afreen Sultana A**

**One-to-One Relationships with Embedded Documents**

It can be modelled using embedded documents. This means one document contains another as a field. It is useful when related data is always accessed together.

db.users.insertOne({name:"Afreen", age:21, permanentaddress:{address:"13 Main street", city:"Chennai", pincode:"600088"}})

output:



**One-to-One Relationships with Referenced Documents**

In this model related data is stored in separate collections, and one document referes to another using an \_id. Useful when related data is large or not always needed.

db.address.insertOne({address:"13 Main street", city:"chennai",pincode:"600088"})

db.person.insertOne({name:"Afreen", age:21, address\_id:ObjectId('688309878cf42684d5e99d0d')})

db.person.find().pretty()

output:



**One-to-Many Relationships with Embedded Documents**

Here the embedded documents are used when the many side is not large. Useful when related data is small and always needed.

db.persons.insertOne({name:"Afreen", age:21, addresses:[{address:"13 Main street", city:"Chennai"},{address:"23 Main road", city:"Mumbai"}]})

Output:



**One-to-Many Relationships with Reference Documents**

When the many data is large or accessed separately. Each related document has a reference to the one document’s \_id.

db.user.insertOne({name:"Afreen", age:21})

db.addresses.insertMany([{address:"13 Main street", city:"Chennai", person\_id:ObjectId('688310a38cf42684d5e99d10')},{address:"23 Main road", city:"Mumbai", person\_id:ObjectId('688310a38cf42684d5e99d10')}])

Output:

