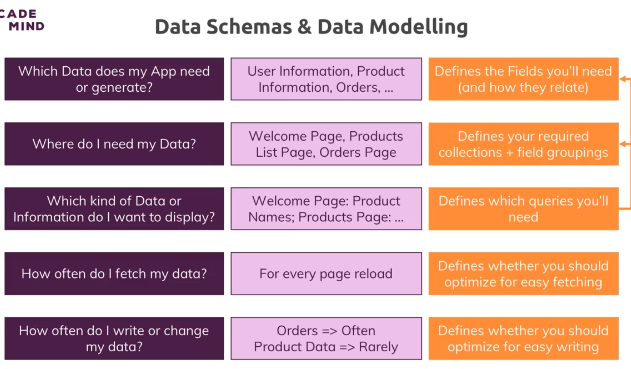
Designing schemas

* Design collections in such a way that u **should not end up in joining the tables** to fetch the records
* Don’t end up in creating duplicate data
* Design the collections in such a way that while retrieving u should not perform complex joins between tables to fetch the required data to display in the front end
* **Duplicate data issue :-**While designing make sure you should not have duplicate data , if u have duplicate data and if u want to update its painful to update every where
* **For tables having one- one relationship better choose embedded documents instead of storing them separately in separate tables**
* **While de**signing tables think of each field , in future are we going to change to object type from single field, analyse and decide to take it as either single or object type
* **Embedded document approach not suitable when**  1 person having 1 car/ 2 cars , if u have a situation if u want to fetch only cars for all persons, then don’t follow embedded documents methodology



2) **Duplicate data issue**

Collection user:-

{

Username : “santhoshi”,

Age: 29,

favBooks:[ {bookname:”java InAction”,AuthorName:”keith serra”},

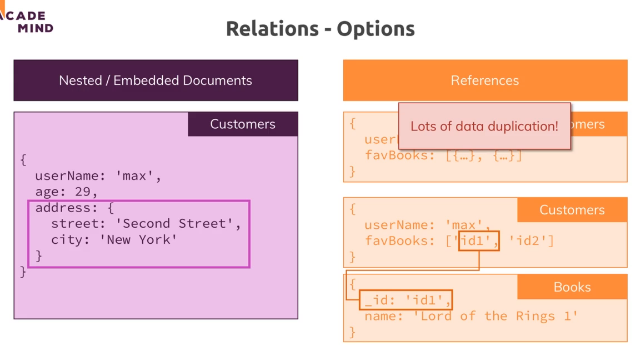
{bookname:”Spring In Action”,AuthorName:”Rod johnson”}

]

}

Think we have hundreds of user almost if all have same favourite book then tomorrow if we have some modifications its practically impossible to change in all places

So in that case use references concept



So as per above create separate collection and store them separately and refer them

Ex:- patient, discharge summary

Every patient will definitely have discharge summary so why to store separately and

join 2 tables while fetching , better store in a single table having embedded document

One to one relationship

1. 1 Person is having cars – this followed embedded documents methodology

{ personName: “Manideep”,

Cars: [ {carName: “BENZ”, horsepower : “2000RPM”} ]

}

In this case if any time if u wanted