## Week4 : Schema Design

Rich documents (array, key:value or doc of doc)

No join

No constraints like foreign key

No transactions but atomic within a collection

No declared Schema as such

Modeling a Blog in Docs

Posts

\_id

title

author

content

comments:[]

tags:[]

date

Author

\_id

name

email

password: hashed

Living without Constraints

No need of joins , all relevent information is already there in the collection

Living without Transactions

Atomic compliant : A single doc operation will be completed then only published to the user

entire Blog content can be updated in just one go instead of joins in case of RDBMS

Implement either of following

1. restructure your model
2. implement transactions in your software part
3. tolerate inconsistency (like visibility of your posts on fb walls)

One to One Relation

Employees and their resumes

Building and floor plan

Patient and medical history

Max allowed size of doc 16Mb

One to One Many

City to Person

People

\_id

name:

city:

City

\_id:city

Many to Many Relation

Books and authors

students and teachers

books

\_id:12

title

authors: [27]

authors

\_id: 27

author\_name:

books:[12,7,15]

Multikeys

db.coll.createIndex( { <field>: < 1 or -1 > } )

Benifits of embedding

improved read performance

one round trip to db

Trees

When to denormalize

1:1 embed

1:many , many to one : embed

many : many link through \_id,