MongoDB

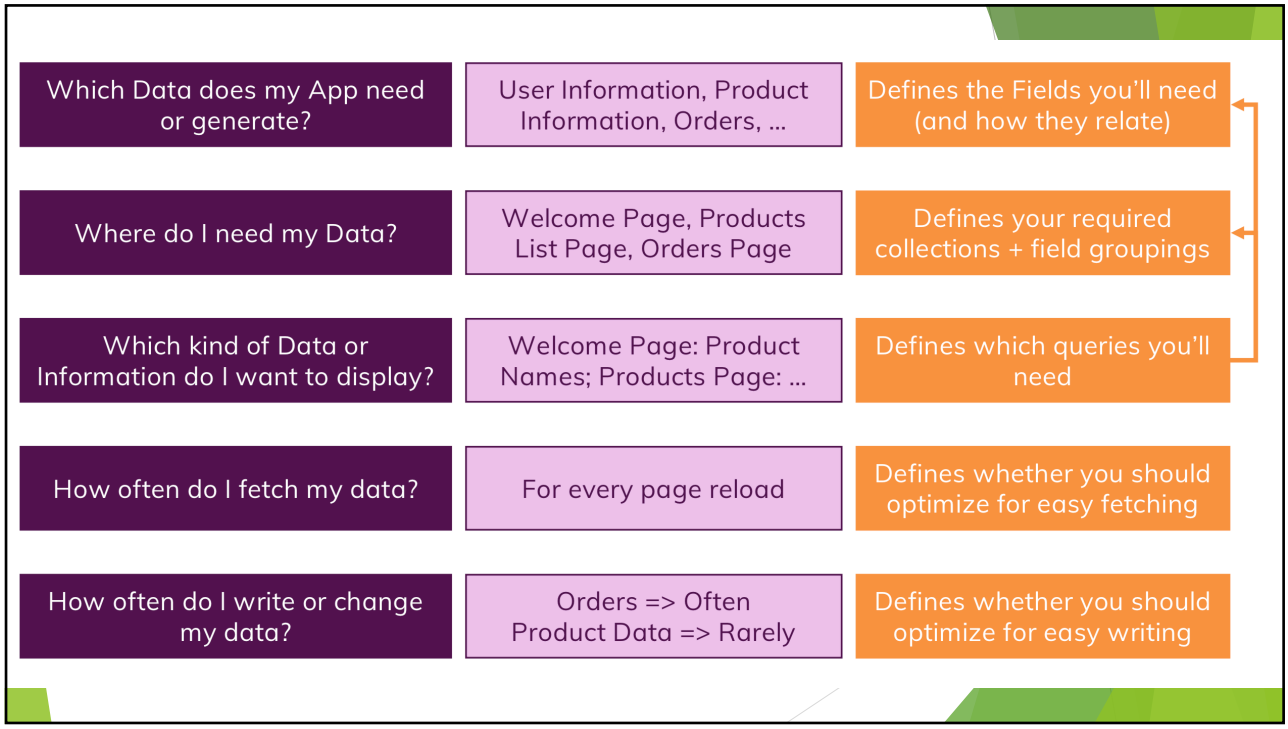
Scope: Data Modelling

Ramu RC

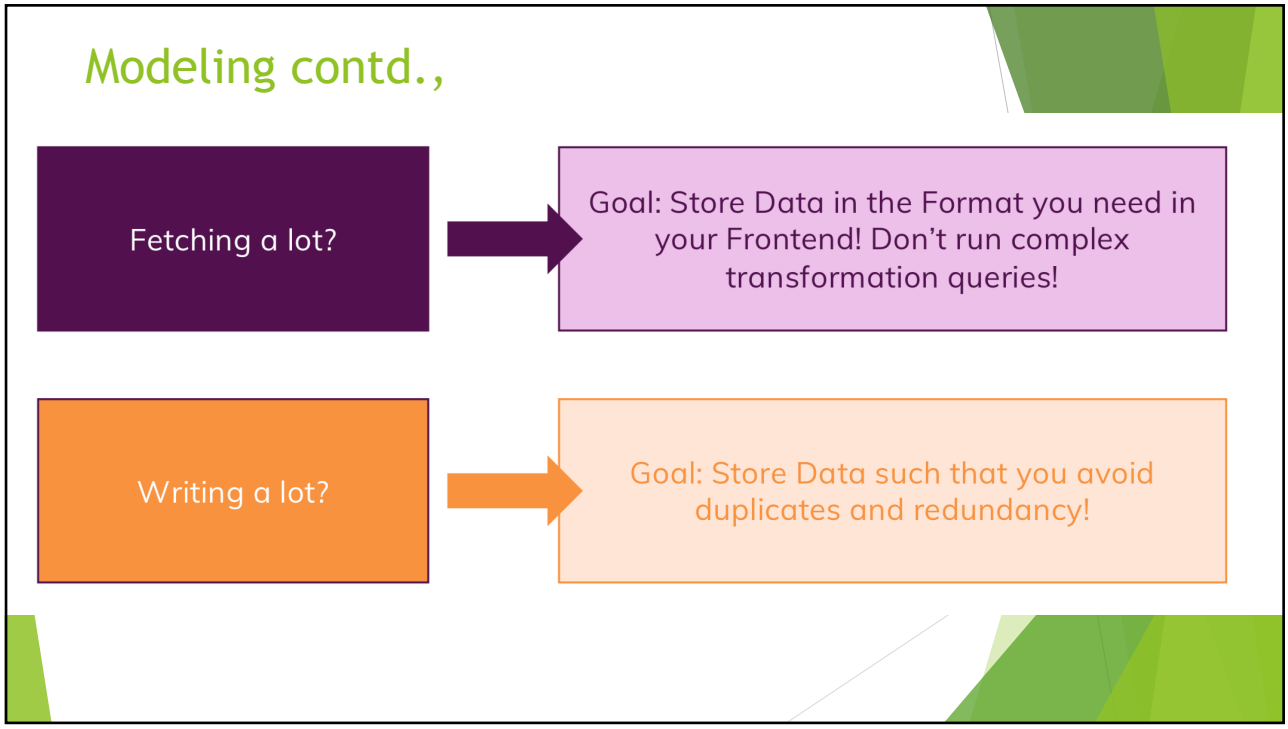
1

Data Modelling

2


Which Data does my App need or generate?
User Information, Product Information, Orders, ...
Defines the Fields you'll need (and how they relate)
Where do I need my Data?
Welcome Page, Products List Page, Orders Page
Defines your required collections + field groupings
Which kind of Data or Information do I want to display?
Welcome Page: Product Names; Products Page: ...
Defines which queries you'll need
How often do I fetch my data?
For every page reload
Defines whether you should optimize for easy fetching
How often do I write or change my data?
Orders => Often
Product Data => Rarely
Defines whether you should optimize for easy writing

3


Modeling contd.,
Fetching a lot?
Goal: Store Data in the Format you need in your Frontend! Don't run complex transformation queries!
Writing a lot?
Goal: Store Data such that you avoid duplicates and redundancy!

4

Checklist

- In which Format will you fetch your Data?
- How often will you fetch and change your Data?
- How much data will you save (and how big is it)?
- How is your Data related?
- Will Duplicates hurt you (=> many Updates)?
- Will you hit Data/ Storage Limits?

5

Modelling Schemas

- Schemas should be modelled based on your application needs
- Important factors are: Read and write frequency, relations, amount (and size) of data

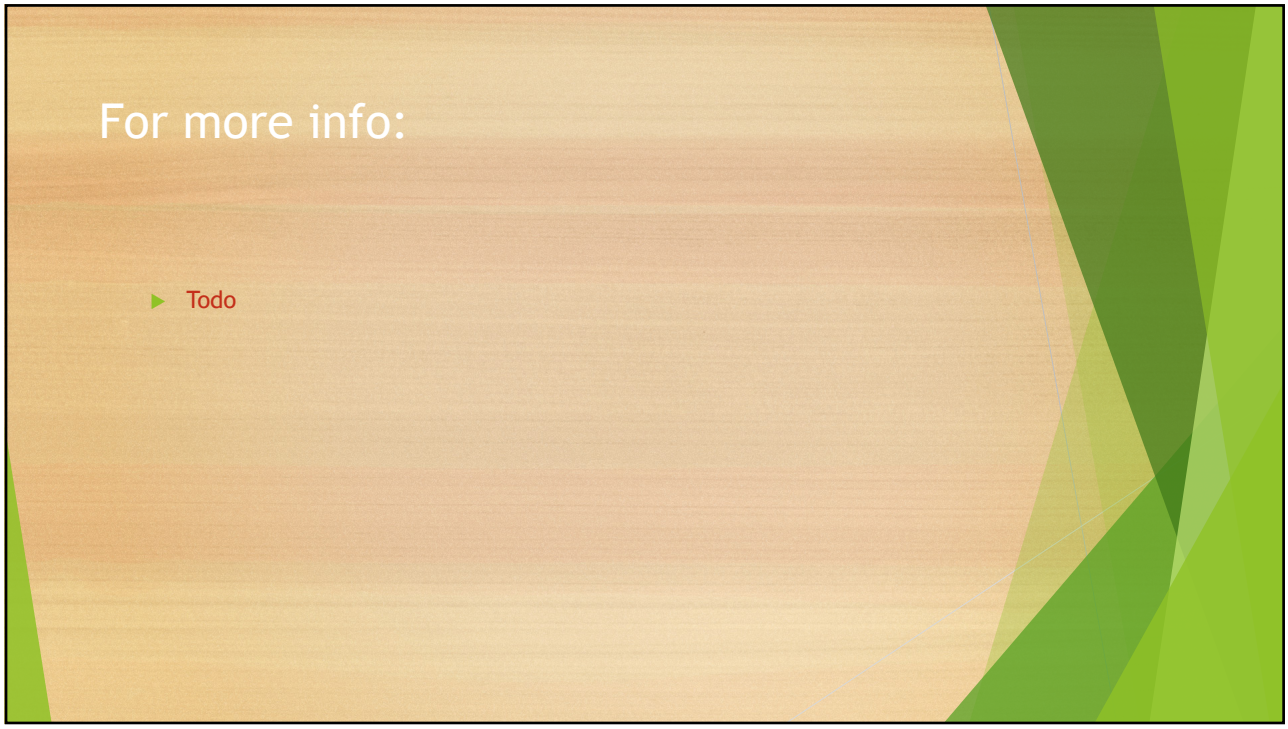
Schema Validation

- You can define rules to validate inserts and update before writing to the database
- Choose your validation level and action based on your application requirements

Modelling Relations

- Two options: Embedded documents or references
- Use embedded documents if you got one-to-one or one-to-many relationships and no app or data size reason to split
- Use references if data amount/ size or application needs require it or for many-to-many relations
- Exceptions are always possible => Keep your app requirements in mind!

6



7