

{ name: "John", age: 32, city: "New city" }

25/01/25

① Price 30 to 50.

Shipping: yes.

↳ Computers

↳ replace details

Data Modelling:

↳ id
name
age
city

3 records.

db.customers.find ({ hobbies : { \$in

↳ AND (both are true show result)

db.products.find ({ \$and : [{ price : { \$gt : 80000 } },
{ brand : 'Apple' }] })

↳ Or! any one is true show result.)

db.products.find ({ \$or : [{ price : { \$lt : 100000 } },
{ brand : 'Apple' }] }).

↳ exists. If it (exists : false) means it show
result.

db.customers.find ({ \$and : [{ hobbies : { \$exists :
false } }, { age : { \$gt : 40 } }] }).

Too access two files
aggregate
↳ group by

```
db.companies.aggregate([{$lookup: {from:
'Employees', localField: 'id', foreignField:
'company-id', as: 'Employees'}}
```

Employees → Should be Array name

②

Create a database called bank,
Create two collections under that 1st collection
name customer personal 2nd collection name
customer Account.

1st

Data Model

Name : String

Address : Array

phone no : object

Age : Int

2nd

3

Data

Account Number	Int
branch	String
bank name	String
IFC code	String
Current Balance	
Account type	
OD	

- 1) Filter only OD category Yes.
- 2) Display only customer address where the branch name are same.
- 3) Current balance 10,000 to 20,000 filter Only the phone number.

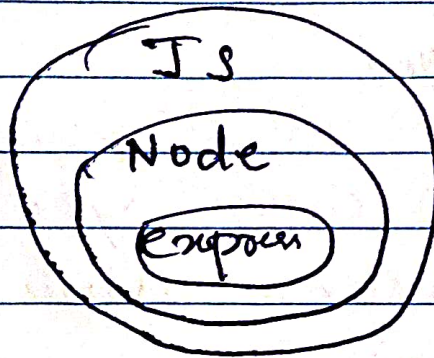
A) filter only saving Account Details

- 5) Add field Status : same
IFC code same as another.

Backend (or) Middleware Node.js

Frontend + Middleware + Database.

In node we can use express
js as Middleware.



Module, Third Party Module,
Custom Module.

```
const http = require('http');  
const express = require('express');  
const say = require('./greet');
```