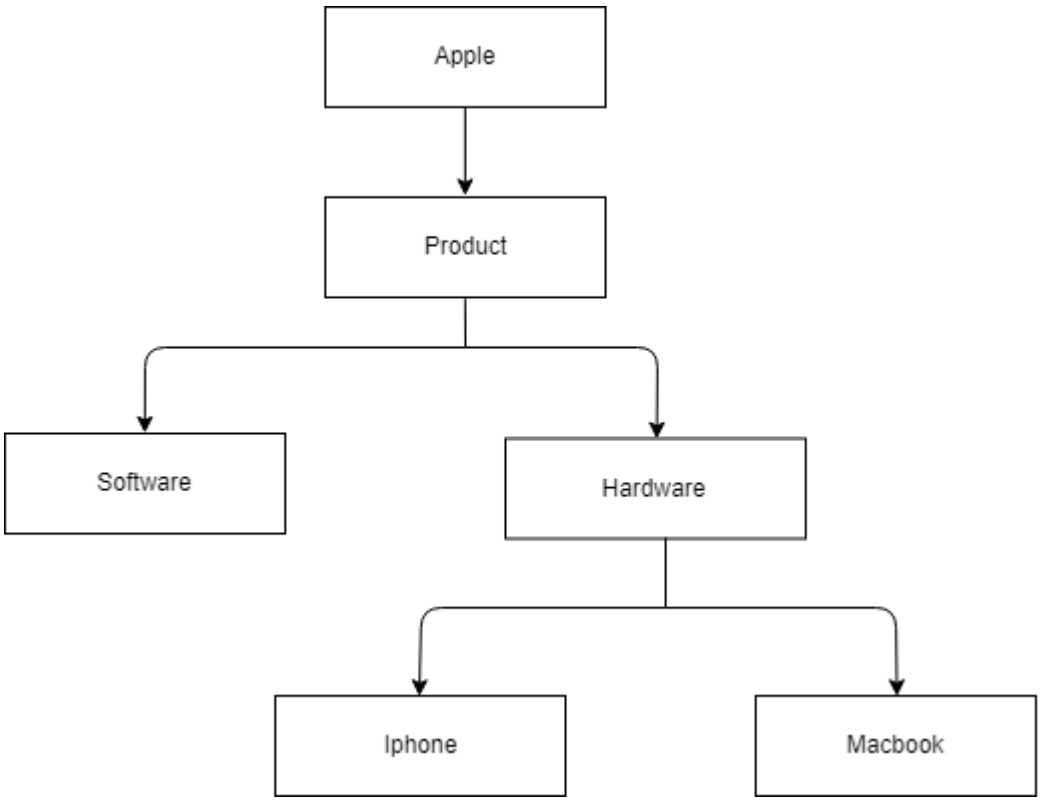
Ahmad miftahul khoiri

17030623023

TUGAS BASIS DATA LANJUT
MONGO DB TREE STRUCTURE

Tingkatan field dari collection “kategori”



1. **Parent References**

membuat database “parentReferences”

```
> use parentReferences
switched to db parentReferences
```

Memasukan data ke collection “kategori”

```
> db.kategori.insert({ _id: "iphone", parent: "hardware"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "macbook", parent: "hardware"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "hardware", parent: "product"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "software", parent: "product"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "product", parent: "apple"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "apple", parent: null})
WriteResult({ "nInserted" : 1 })
>
```

Perintah untuk menampilkan parent

```
>
> db.kategori.findOne ({ _id: "iphone"}).parent
hardware
>
> db.kategori.findOne ({ _id: "hardware"}).parent
product
```

membuat indeks pada field induk untuk mengaktifkan pencarian cepat oleh simpul induk

```
>
> db.kategori.createIndex ({parent:1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

Mencari field berdasarkan nama parent

```
> db.kategori.find({parent:"product"})
{ "_id" : "hardware", "parent" : "product" }
{ "_id" : "software", "parent" : "product" }
>
```

2. Child References

```
> use childReferences
switched to db childReferences
> db
childReferences
```

```
> db.kategori.insert({ _id: "iphone",children:[] } )
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "macbook",children:[] } )
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "hardware",children:["iphone","macbook"] } )
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "software",children:[] } )
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "product",children:["hardware","software"] } )
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "apple",children:["product"] } )
WriteResult({ "nInserted" : 1 })
>
```

```
>
> db.kategori.findOne ({ _id: "product"}).children
[ "hardware", "software" ]
>
```

```
> db.kategori.createIndex ({children:1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

```
>
> db.kategori.find({children:"product"})
{ "_id" : "apple", "children" : [ "product" ] }
>
> db.kategori.find({children:"iphone"})
{ "_id" : "hardware", "children" : [ "iphone", "macbook" ] }
>
```

3. Array Of Ancestors

```
> use arrayOfAncestors
switched to db arrayOfAncestors
> db
arrayOfAncestors
```

```
> db.kategori.insert ({ _id:"iphone", ancestors: ["apple","product","hardware"], parent: "hardware"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert ({ _id:"macbook", ancestors: ["apple","product","hardware"], parent: "hardware"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert ({ _id:"hardware", ancestors: ["apple","product"], parent: "product"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert ({ _id:"software", ancestors: ["apple","product"], parent: "product"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert ({ _id:"product", ancestors: ["apple"], parent: "apple"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert ({ _id:"apple", ancestors: [], parent: null})
WriteResult({ "nInserted" : 1 })
```

```
> db.kategori.findOne({ _id:"macbook"}).ancestors
[ "apple", "product", "hardware" ]
```

```
> db.kategori.createIndex ({ancestors:1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

```
> db.kategori.find({ancestors:"hardware"})
{ "_id" : "iphone", "ancestors" : [ "apple", "product", "hardware" ], "parent" : "hardware" }
{ "_id" : "macbook", "ancestors" : [ "apple", "product", "hardware" ], "parent" : "hardware" }
>
```

4. Materialized Paths

```
> use materializedPaths
switched to db materializedPaths
> db
materializedPaths
```

```
>
> db.kategori.insert({ _id: "apple", path: null })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "product", path: ",apple," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "software", path: ",apple,product," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "hardware", path: ",apple,product," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "iphone", path: ",apple,product,hardware," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id: "macbook", path: ",apple,product,hardware," })
WriteResult({ "nInserted" : 1 })
>
```

```
>
> db.kategori.find().sort ({ path: 1 })
{ "_id" : "apple", "path" : null }
{ "_id" : "product", "path" : ",apple," }
{ "_id" : "software", "path" : ",apple,product," }
{ "_id" : "hardware", "path" : ",apple,product," }
{ "_id" : "iphone", "path" : ",apple,product,hardware," }
{ "_id" : "macbook", "path" : ",apple,product,hardware," }
>
```

```
>
> db.kategori.find({ path: /,hardware,/ })
{ "_id" : "iphone", "path" : ",apple,product,hardware," }
{ "_id" : "macbook", "path" : ",apple,product,hardware," }
>
> db.kategori.find({ path: /^,apple,/ })
{ "_id" : "product", "path" : ",apple," }
{ "_id" : "software", "path" : ",apple,product," }
{ "_id" : "hardware", "path" : ",apple,product," }
{ "_id" : "iphone", "path" : ",apple,product,hardware," }
{ "_id" : "macbook", "path" : ",apple,product,hardware," }
>
```