

### Creating parent structure

```
db.categories.insert( { _id: "MongoDB", parent: "Databases" } )
db.categories.insert( { _id: "dbm", parent: "Databases" } )
db.categories.insert( { _id: "Databases", parent: "Programming" } )
db.categories.insert( { _id: "Languages", parent: "Programming" } )
db.categories.insert( { _id: "Programming", parent: "Books" } )
db.categories.insert( { _id: "Books", parent: null } )
```

### Finding Parent of a node

```
db.categories.findOne( { _id: "MongoDB" } ).parent
```

```
db.categories.findOne( { _id: "Books" } ).parent
```

Find the node for which "Parent" is "Databases"

```
db.categories.find( { parent: "Databases" } )
```

```
db.categories.find( { parent: "MongoDB" } )
```

### Delete Parent structure before creating Child Structure

```
db.categories.remove({})
```

### Creating Child structure

```
db.categories.insert( { _id: "MongoDB", children: [] } );
db.categories.insert( { _id: "dbm", children: [] } );
db.categories.insert( { _id: "Databases", children: [ "MongoDB", "dbm" ] } );
db.categories.insert( { _id: "Languages", children: [] } )
```

```
db.categories.insert( { _id: "Programming", children: [ "Databases", "Languages" ] } );
```

```
db.categories.insert( { _id: "Books", children: [ "Programming" ] } );
```

The query to retrieve the immediate child of a node

```
db.categories.findOne( { _id: "Databases" } ).children;
```

Query by the child field to find its parent nodes

```
db.categories.find( { children: "MongoDB" } );
```

```
db.categories.find( { children: "Books" } );
```