

Building the Mongoose Schema



Mark Scott

AUTHOR

@tripletdad99



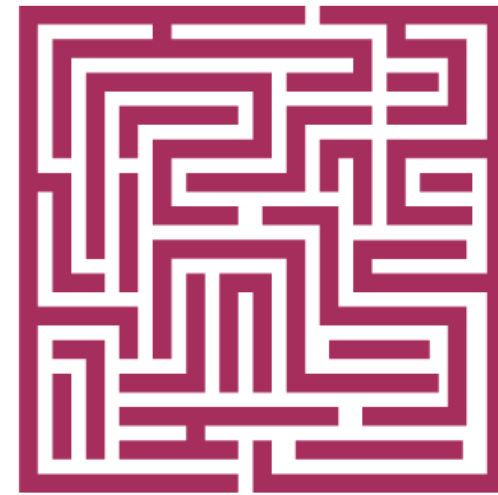
Everything in Mongoose
starts with a Schema.



Mongoose Schema Examples



Simple Schema



Complex Schema

Compare and Contrast



Traditional Relational Database Schema
Mongoose Schema

Comparison - Relational Database Schemas



Relational Database Schema Comparison



QUESTIONS

1- A B C D

2- A B C D

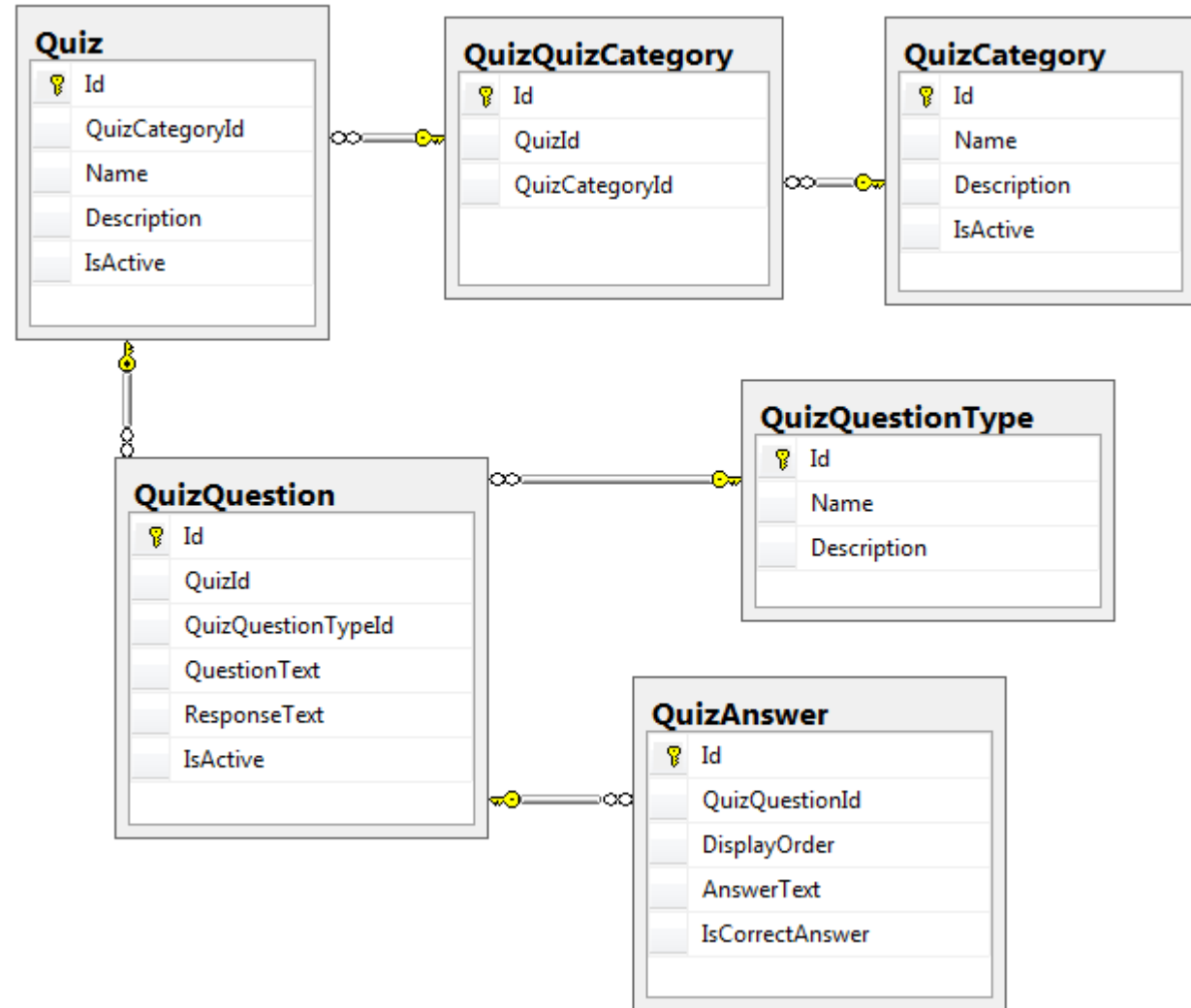
3- A B C D

4- A B C D

5- A B C D

6- A B C D

Relational Database Schema Comparison



```
name
description
isActive
```

```
categories [ ... ]
```

```
questions [
  questionType
  questionText
```

```
    answers [
      answerText
      isCorrect
      displayOrder
```

```
    ]
```

```
]
```

- ◀ Quiz root document
- ◀ Categories sub-document array
- ◀ Questions sub-document array
- ◀ Answers sub-document array



Simple Schema Example



Simple Schema Example

```
const mongoose = require('mongoose');
```

```
const customerSchema = new mongoose.Schema({  
  name: String,  
  address: String,  
  city: String,  
  state: String,  
  country: String,  
  zipCode: Number,  
  createdAt: Date,  
  isActive: Boolean  
});
```

```
const simpleSchema = new Schema({ fieldName: SchemaType });
```



Mongoose Data Types

Mongoose Schema Types

String
Number
Date
Buffer
Boolean
Mixed
ObjectId
Array

JavaScript Data Types

String
Number
Object
Object
Boolean
Object
Object
Array (Object)



Complex Schema Example



Complex Schema Example

```
const mongoose = require('mongoose');  
  
// child address schema  
// parent customer schema:..  
const addressSchema = new mongoose.Schema({  
  type: String,  
  name: {  
    street: String,  
    first: String,  
    city: String,  
    last: String,  
    state: String,  
    country: String,  
  },  
  address: [addressSchema ],  
  postalCode: Number,  
  createdAt: { type: Date, default: Date.now },  
});  
const customerSchema = new Schema({  
  isActive: { type: Boolean, default: true },  
});
```



```
const mongoose = require('mongoose');

// Pre-define sub-documents...
const subCategory = {
  name: String,
  description: String,
  isActive: Boolean };

const subAnswers = {
  answerText: String,
  isCorrect: Boolean,
  displayOrder: Number };

const subQuestions = {
  questionType: String,
  questionText: String,
  answers: [ subAnswers ] };

// Define main document schema...
const quizSchema = new mongoose.Schema({
  name: String,
  description: String,
  isActive: Boolean,
  categories: [ subCategory ],
  questions: [ subQuestions ]
});
```

The diagram illustrates a MongoDB document structure. It is a large object with the following fields:

- `name`
- `description`
- `isActive`
- `categories [...]` (highlighted with a red border)
- `questions [` (highlighted with a blue border)
 - `questionType`
 - `questionText`
 - `answers [` (highlighted with a purple border)
 - `answerText`
 - `isCorrect`
 - `displayOrder`
 - `]`
- `]`



```
const mongoose = require('mongoose');

// child address schema...
const addressSchema = new mongoose.Schema({
  type: String,
  street: String,
  city: String,
  state: String,
  country: String,
  postalCode: Number
});

// parent customer schema...
const customerSchema = new Schema({
  name: {
    first: String,
    last: String
  },
  address: [ addressSchema ],
  createdAt: { type: Date, default: Date.now },
  isActive: { type: Boolean, default: true },
});
```



```

const mongoose = require('mongoose');

// Pre-define sub-documents...
const subCategory = {
  name: String,
  description: String,
  isActive: Boolean };

const subAnswers = {
  answerText: String,
  isCorrect: Boolean,
  displayOrder: Number };

const subQuestions = {
  type: String,
  text: String,
  answers: [ subAnswers ] };

// Define main document schema...
const quizSchema = new mongoose.Schema({
  name: String,
  description: String,
  isActive: Boolean,
  categories: [ subCategory ],
  questions: [ subQuestions ]
});

```

```

{
  "_id" : ObjectId("5422cef53ba88f182229f450"),
  "name" : "Test Quiz",
  "description" : "Test Quiz long description...",
  "isActive" : true,
  "questions" : [
    "[object Object]",
    "[object Object]"
  ],
  "categories" : [
    {
      "name" : "programming",
      "description" : "programming category",
      "isActive" : true,
      "_id" : ObjectId("5422cef53ba88f182229f452")
    },
    {
      "name" : "Test2",
      "description" : "Test 2 category",
      "isActive" : true,
      "_id" : ObjectId("5422cef53ba88f182229f451")
    }
  ],
  "__v" : 0
}

```

 **Oops!**




```

const mongoose = require('mongoose');

// Pre-define sub-documents...
const subCategory = {
  name: String,
  description: String,
  isActive: Boolean };

const subAnswers = {
  answerText: String,
  isCorrect: Boolean,
  displayOrder: Number };

const subQuestions = {
  type: { type: String },
  text: String,
  answers: [ subAnswers ] };

// Define main document schema...
const quizSchema = new mongoose.Schema({
  name: String,
  description: String,
  isActive: Boolean,
  categories: [ subCategory ],
  questions: [ subQuestions ]
});

```

```

{
  "_id" : ObjectId("5422cfdb36663a0424d7a66b"),
  "name" : "Test Quiz",
  "description" : "Test Quiz long description...",
  "isActive" : true,
  "questions" : [
    {
      "type" : "Multiple-Choice",
      "text" : "What is your favorite color?",
      "_id" : ObjectId("5422cfdb36663a0424d7a670"),
      "answers" : [
        {
          "answerText" : "Red",
          "isCorrect" : false,
          "displayOrder" : 1,
          "_id" : ObjectId("5422cfdb36663a0424d7a673")
        },
        // abbreviated for brevity sake ...
      ],
      // abbreviated for brevity sake ...
    },
    // abbreviated for brevity sake ...
  ],
  "categories" : [
    // abbreviated for brevity sake ...
  ]
}

```



Demo



Time to start developing our schema

- standup.js
- teamMember.js
- project.js

Model folder

- Models are built from the schema

Review Demo Application



Next Steps



Build a model

- from the Mongoose schema

