```
/* What is Data-Association in mongodb:- Data association in MongoDB refers to the way
relationships between different pieces of data are managed within the database. Since
MongoDB is a NoSQL database, it doesn't use traditional relational database concepts
like foreign keys and joins directly. However, it still supports modeling relationships
between documents. There are two primary ways to model relationships in MongoDB: embedding
documents and referencing documents.
*/

/* 1. Embedding Documents :- In this approach, related data is stored within the same
document. This is useful when you have a "one-to-few" relationship or when the related
data is often retrieved together.
*/
/* Ex:-
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const addressSchema = new Schema({
  street: String,
  city: String,
  state: String,
  zip: String
});
const userSchema = new Schema({
  name: String,
  age: Number,
  addresses: [addressSchema]
});
const User = mongoose.model('User', userSchema);
*/

/*2. Referencing Documents :- In this approach, related data is stored in separate
documents, and references (usually ObjectIds) are used to link them. This is useful for
"one-to-many" or "many-to-many" relationships.
*/

/* Ex:-
User Schema:-
const userSchema = new Schema({
  name: String,
  age: Number
});
const User = mongoose.model('User', userSchema);

postSchema:-
const postSchema = new Schema({
  title: String,
  content: String,
  author: { type: Schema.Types.ObjectId, ref: 'User' }
});
const Post = mongoose.model('Post', postSchema);

To query and populate the references:-
Post.find()
  .populate('author')
  .exec((err, posts) => {
    if (err) return handleError(err);
```

```
    console.log(posts);
});
*/

```