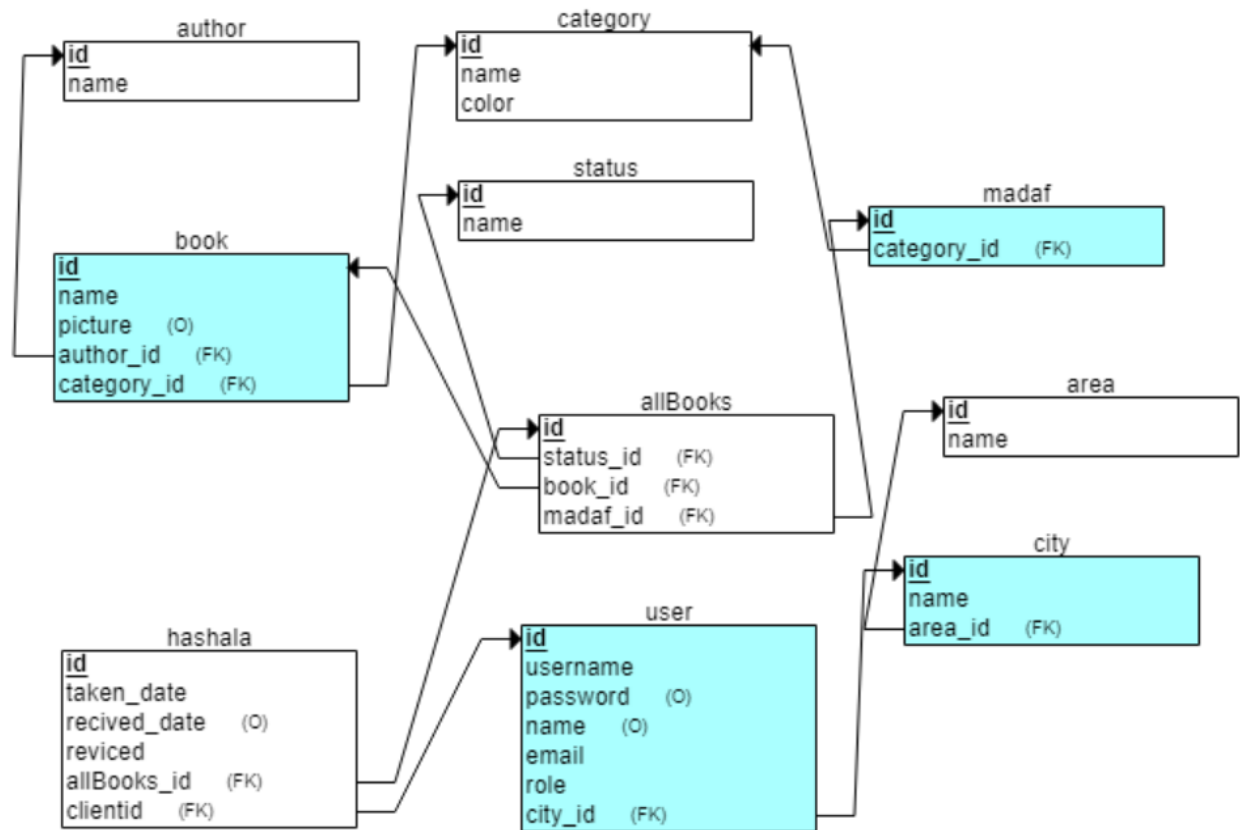


EX3: Sequelize Associations



טבלאות עם קשרים

נתחיל עם טבלת ספרים, כל ספר משויך למחבר (קשר יחיד לרבים)
וכל ספר משויך לקטגוריה (ספרי ילדים וכד')
נעבור על השלבים מפרק ב אך נוסיף עוד שלב של יצירת קשרים

1. יצירת המודל

שלב ראשון ניצור את האובייקט של הספר book.js

```
const { sequelize, DataTypes } = require("../sequelize");
const Book = sequelize.define("book",
{
  name: {
    type: DataTypes.STRING,
    allowNull: false,
```

```

    },
    picture:{
      type:DataTypes.STRING,
    },
  });
module.exports = Book;

```

2. עדכון המודל במסד הנתונים

ונוסיף אותו בקובץ הראשי של מסד הנתונים index.js

```
db.books = require('./book')
```

3. הוספת קשרים

הוספת קובץ קשרים

תחת תקיית models הmodels ניצור קובץ extra-setup.js - להגדרת הקשרים יש לשים לב שהמודלים זה השם שנתנו למודל בפונקציית DEFINE

```

> ex1
> ex2
> ex3
  > config
  > controllers
  > migrations
  > models
    JS area.js
    JS author.js
    JS book.js
    JS category.js
    JS extra-setup.js
    JS index.js
    JS note.js
    JS sequelize.js
    JS status.js
  > node_modules

```

```

1  const { sequelize } = require("../sequelize");
2
3  const applyExtraSetup = () => {
4    const { book, author, category } = sequelize.models;
5
6    book.belongsTo(category);
7    book.belongsTo(author);
8
9    author.hasMany(book);
10   category.hasMany(book);
11 };
12
13 module.exports = { applyExtraSetup };
14
15
16
17
18
19
20

```

```
const { sequelize } = require("../sequelize");

const applyExtraSetup = () => {
  const { book, author, category } = sequelize.models;

  book.belongsTo(category);
  book.belongsTo(author);

  author.hasMany(book);
  category.hasMany(book);
};

module.exports = { applyExtraSetup };
```

belongsTo

X.belongsTo(Y)

היחיד זה X שייך לY - [הספר שייך למחבר אחד]

hasMany

Y.hasMany(X)

היחיד זה X שייך לY - [למחבר מסויים יש הרבה ספרים]

בקובץ index.js נייבא את פונקציית הקשרים שיצרנו

```
const { applyExtraSetup } = require('../extra-setup');
```

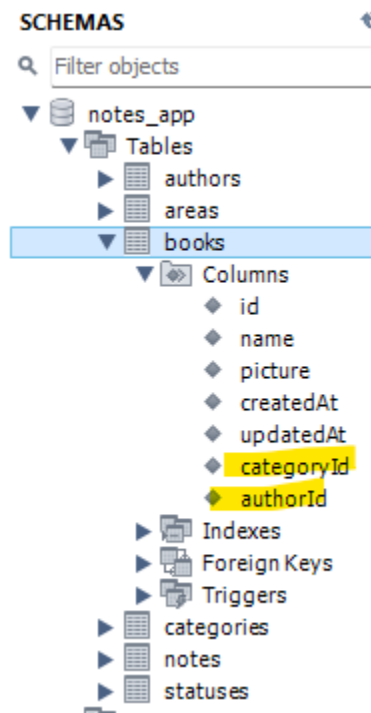
ונפעיל את הפונקציה עם אובייקט מסד הנתונים לאחר יצירת כל המודלים

```
applyExtraSetup();
```

```
SERVER
> ex1
> ex2
✓ ex3
  > config
  > controllers
  > migrations
  ✓ models
    JS area.js
    JS author.js
    JS book.js
    JS category.js
    JS extra-setup.js
    JS index.js
    JS note.js
    JS sequelize.js
    JS status.js
  > node_modules
  > public
  > routes
  > seeders
  > views

ex3 > models > JS index.js > ...
1  const {Sequelize} = require('sequelize');
2  const {sequelize} = require('./sequelize');
3  const { applyExtraSetup } = require('./extra-setup');
4
5  const db = {}
6
7  db.Sequelize = Sequelize
8  db.sequelize = sequelize
9  //*****MODELS*****//
10 db.notes = require('./note')
11 db.categories = require('./category')
12 db.statuses = require('./status')
13 db.areas = require('./area')
14 db.authors = require('./author')
15 db.books = require('./book')
16 //*****END MODELS*****//
17 applyExtraSetup();
18
19 db.sequelize.sync({ alter: true })
20 .then(() => {
21   console.log('yes re-sync done!')
22 })
23 module.exports = db
```

אם אין הגדרה נוספת אז המערכת אוטומטית מכניסה את שדה הFK בטבלת היחיד



ניתן גם להגדיר את השדה באופן ידני אך יש לשים לב להגדיר גם על טבלת היחיד את שם השדה וגם על טבלת הרבים

```
const { sequelize } = require("../sequelize");

const applyExtraSetup = () => {
  const { book, author, category } = sequelize.models;
  book.belongsTo(category, { foreignKey: "cateogry_id", as: "category" });
  book.belongsTo(author, { foreignKey: "author_id", as: "author" });
  author.hasMany(book, { foreignKey: "author_id", as: "books" });
  category.hasMany(book, { foreignKey: "cateogry_id", as: "books" });
};

module.exports = { applyExtraSetup };
```

```
const { sequelize } = require("../sequelize");

const applyExtraSetup = () => {
  const { book, author, category } = sequelize.models;
  book.belongsTo(category, { foreignKey: "cateogry_id", as: "category" });
  book.belongsTo(author, { foreignKey: "author_id", as: "author" });
  author.hasMany(book, { foreignKey: "author_id", as: "books" });
  category.hasMany(book, { foreignKey: "cateogry_id", as: "books" });
};

module.exports = { applyExtraSetup };
```

ובנוסף ניתן להגדיר את השדות גם בהגדרת המודל על מנת להחיל עליהם הגדרות של שדה חובה וכד

```
cateogry_id:{
  type:DataTypes.INTEGER,
  allowNull:false
},
author_id:{
  type:DataTypes.INTEGER,
  allowNull:false
}
```

```
const { sequelize, DataTypes } = require("../sequelize");
const Book = sequelize.define("book",
  {
    name:{
      type:DataTypes.STRING,
      allowNull: false,
    },
    picture:{
      type:DataTypes.STRING,
    },
    cateogry_id:{
      type:DataTypes.INTEGER,
      allowNull:false
    },
    author_id:{
      type:DataTypes.INTEGER,
      allowNull:false
    }
  }
));
module.exports = Book;
```