L4T35 – Final Capstone Project V: Authentication.

Create the basic DB Structure for modeling [1]Users/Employees, [2]OUs, [3]Divisions, and [4]Credential Repositories.

Modeling a Credential Repository:

All employees of a division should have access to it.

[0]Division Shorthand:	[1]~~Organizational Unit:~~	[2]~~Division:~~	[3]Place:	[4]Username:	[5]Password:
1	News Management	Finances	WP Sites	newsfin_wp	hydrogen
1	News Management	Finances	Servers	newsfin_servers	helium
1	News Management	Finances	Financial Accounts	newsfin_finances	lithium
2	News Management	IT	WP Sites	newsit_wp	beryllium
2	News Management	IT	Servers	newsit_servers	boron
2	News Management	IT	Financial Accounts	newsit_finances	carbon
3	News Management	Writing	WP Sites	newswrit_wp	nitrogen
3	News Management	Writing	Servers	newswrit_servers	oxygen
3	News Management	Writing	Financial Accounts	newswrit_finances	fluorine
4	News Management	Development	WP Sites	newsdev_wp	neon
4	News Management	Development	Servers	newsdev_servers	sodium
4	News Management	Development	Financial Accounts	newsdev_finances	magnesium
5	Software Reviews	Finances	WP Sites	softfin_wp	aluminium
5	Software Reviews	Finances	Servers	softfin_servers	silicon
5	Software Reviews	Finances	Financial Accounts	softfin_finances	phosphorus
6	Software Reviews	IT	WP Sites	softit_wp	sulfur
6	Software Reviews	IT	Servers	softit_servers	chlorine
6	Software Reviews	IT	Financial Accounts	softit_finances	argon
7	Software Reviews	Writing	WP Sites	softwrit_wp	potassium
7	Software Reviews	Writing	Servers	softwrit_servers	calcium
7	Software Reviews	Writing	Financial Accounts	softwrit_finances	scandium
8	Software Reviews	Development	WP Sites	softdev_wp	titanium
8	Software Reviews	Development	Servers	softdev_servers	vanadium
8	Software Reviews	Development	Financial Accounts	softdev_finances	chromium
9	Hardware Reviews	Finances	WP Sites	hardfin_wp	manganese
9	Hardware Reviews	Finances	Servers	hardfin_servers	iron

9	Hardware Reviews	Finances	Financial Accounts	hardfin_finances	cobalt
10	Hardware Reviews	IT	WP Sites	hardit_wp	nickel
10	Hardware Reviews	IT	Servers	hardit_servers	copper
10	Hardware Reviews	IT	Financial Accounts	hardit_finances	zinc
11	Hardware Reviews	Writing	WP Sites	hardwrit_wp	gallium
11	Hardware Reviews	Writing	Servers	hardwrit_servers	germanium
11	Hardware Reviews	Writing	Financial Accounts	hardwrit_finances	arsenic
12	Hardware Reviews	Development	WP Sites	harddev_wp	selenium
12	Hardware Reviews	Development	Servers	harddev_servers	bromine
12	Hardware Reviews	Development	Financial Accounts	harddev_finances	krypton
13	Opinion Publishing	Finances	WP Sites	opfin_wp	rubidium
13	Opinion Publishing	Finances	Servers	opfin_servers	strontium
13	Opinion Publishing	Finances	Financial Accounts	opfin_finances	yttrium
14	Opinion Publishing	IT	WP Sites	opit_wp	zirconium
14	Opinion Publishing	IT	Servers	opit_servers	niobium
14	Opinion Publishing	IT	Financial Accounts	opit_finances	molybdenum
15	Opinion Publishing	Writing	WP Sites	opwrit_wp	technetium
15	Opinion Publishing	Writing	Servers	opwrit_servers	ruthenium
15	Opinion Publishing	Writing	Financial Accounts	opwrit_finances	rhodium
16	Opinion Publishing	Development	WP Sites	opdev_wp	palladium
16	Opinion Publishing	Development	Servers	opdev_servers	silver
16	Opinion Publishing	Development	Financial Accounts	opdev_finances	cadmium

Caveat:

- Most employees are only part of one OU and one Division in it.
- **BUT there are some employees that are part of more than one OU and Division.**

Modeling Divisions:

[1]**Organizational Unit:**	[2]**Division:**	[3]**Division_ID_Number** Unique Number for Every Row.	[4]**Shorthand:**
News Management	Finances	6504457da8416917276b05d5	1
News Management	IT	650445f8a8416917276b05d6	2
News Management	Writing	65044667a8416917276b05d7	3
News Management	Development	650446afa8416917276b05d8	4
Software Reviews	Finances	65044710a8416917276b05d9	5
Software Reviews	IT	6504473ea8416917276b05da	6
Software Reviews	Writing	6504475ba8416917276b05db	7
Software Reviews	Development	65044773a8416917276b05dc	8
Hardware Reviews	Finances	650447aba8416917276b05dd	9
Hardware Reviews	IT	650447eaa8416917276b05de	10
Hardware Reviews	Writing	65044809a8416917276b05df	11
Hardware Reviews	Development	6504481ca8416917276b05e0	12
Opinion Publishing	Finances	65044857a8416917276b05e1	13
Opinion Publishing	IT	6504488ba8416917276b05e2	14
Opinion Publishing	Writing	6504489aa8416917276b05e3	15
Opinion Publishing	Development	650448ada8416917276b05e4	16

Modeling a CoolTech Employee:

Assume the same role across all divisions.

[1]Username:	[2]Password:	[3]Division_no_1:	[4]Division_no_2	[5]Division_no_3	[6]Role:
baldomar	afghanistan	1	2	16	normal
karrie	albania	1	3		management
lenny	algeria	1			admin
tawnee	andorra	2	3	15	normal
vena	angola	2	4		admin
gerd	anguilla	2			management
nerva	antartica	3	4	14	management
richardis	argentina	3	5		admin
keeleigh	armenia	3			normal
xenophon	aruba	4	5	13	management
sloane	australia	4	6		normal
jemima	austria	4			admin
freyr	azerbaijan	5	6	12	admin
callahan	bahamas	5	7		normal
leon	bahrain	5			management
makarios	bangladesh	6	7	11	admin
hyacinthus	barbados	6	8		management
galenos	belarus	6			normal
trueman	belgium	7	8	10	normal
eumelia	belize	7	9		management
nolan	greenland	7			admin
kris	bermuda	8	9	11	normal
solveig	bhutan	8	10		admin
geroald	bolivia	8			management
felagi	botswana	9	10	8	management
dipaka	brazil	9	11		admin
baylee	bulgaria	9			normal
shelagh	burundi	10	11	7	management
pheme	cambodia	10	12		normal
essence	cameroon	10			admin
ekkebert	canada	11	12	6	admin
ealhstan	greece	11	13		normal
sierra	gibraltar	11			management
rusty	colombia	12	13	5	admin
anastasia	comoros	12	14		management
valeriana	germany	12			normal

valentina	croatia	13	14	4	normal
appius	georgia	13	15		management
meliton	cyprus	13			admin
christmas	denmark	14	15	3	normal
esme	djibouti	14	16		admin
eudoxos	dominica	14			management
jimmy	ecuador	16	15	2	management
judda	france	16	15		admin
wilmer	eritrea	16			normal

Does the employee have **one role** across all the divisions he belongs to? → Yes.

```
Mongoose.SchemaTypes.ObjectID  +  Ref
```

Admins:

1. They can **assign** and **un-assign** users from divisions and OUs.
2. The can **change the role** of any user.

~~Do they have access to ALL the employee-data in the *Employee-Collection*? []~~

Only have access to the employees of their divisions**.**

~~Do they have access to ALL *Divisions* of every *Organisational Unit*?~~

~~I.e. does an Admin have *'global access'* to **everything** on the credential management web app?~~

~~If that is the case, then it seems meaningless to assign a Division to the Admin user in the Employee Collection?~~

How to use:

```
bestFriend: {
  type: mongoose.SchemaTypes.ObjectId,
  ref: "User",
}
```

StackOverflow Link | Referencing another Schema in Mongoose:

```
const userSchema = new Schema({
    twittername: String,
    twitterID: Number,
    displayName: String,
    profilePic: String,
});

const postSchema = new Schema({
    name: String,
    postedBy: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
    dateCreated: Date,
});

const Post = mongoose.model('Post', postSchema);
const User = mongoose.model('User', userSchema);
```

Then, when you make your query, you can populate references like this:

```
Post.findOne({_id: 123})
.populate('postedBy')
.exec(function(err, post) {
    // do stuff with post
});
```

Maksym Strukov (23 January 2014), Embedded document vs reference in mongoose design model?, Accessed On: 14 September 2023, Retrieved From: https://stackoverflow.com/questions/21302279/embedded-document-vs-reference-in-mongoose-design-model

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

const eventSchema = Schema({
    title: String,
    location: String,
    startDate: Date,
    endDate: Date
});

const personSchema = Schema({
    firstname: String,
    lastname: String,
    email: String,
    gender: {type: String, enum: ["Male", "Female"]}
    dob: Date,
    city: String,
    eventsAttended: [{ type: Schema.Types.ObjectId, ref: 'Event' }]
});

const Event  = mongoose.model('Event', eventSchema);
const Person = mongoose.model('Person', personSchema);
```

```
aaron = new Person({firstname: 'Aaron'}) // and an event object,

event1 = new Event({title: 'Hackathon', location: 'foo'}):

aaron.eventsAttended.push(event1);
aaron.save(callback);
```

MongoDB and Mocha *(Hitesh Choudhary)* | 34 Videos

One-to-One Relation in Mongo Database with ID:

```
db.persons.insertOne({
  name: "hitesh",
  isVerified: true,
  earning: 3000
})  // "insertedId" : ObjectId("5c9b90361a0344a90724042b");

db.videos.insertOne({
  topic: "Fun One",
  length: 4,
  creator: ObjectId("5c9b90361a0344a90724042b")  // Enable the BSON functionality.
})
```

One-to-Many relation in Mongo DataBase:

One YouTuber is creating many videos: One-to-Many Relationship.

The reference ID approach:

```
db.comment.insertOne({
   videoname: "react js",
   comment: "How to install reactjs",
   replies: ["rep112", "rep223"]  // But why not [ObjectId("rep112"), ObjectId("rep223")]
})  // "insertedId" : ObjectId("5c9ba0cc1a0344a90724042e").


db.replies.insertMany([
   {
      _id: "rep112",
      text: "You can use node"
   },
   {
      _id: "rep223",
      text: "You can use homebrew"
   }
])
```

Many-to-Many Relation in MongoDB.

Example – User can Courses:

One user purchases many courses.
One course is purchased by many users.

```
db.courses.insertOne({
   name: "react native",
   price: 12.0
})  // "insertedId" : ObjectId("5ca24bf5e7949e55b8999735");

db.users.insertOne({
   name: "hitesh",
   isVerified: true
})  // "insertedId" : ObjectId("5ca24c2de7949e55b8999736");

db.purchase.insertOne({
   productId:  ObjectId("5ca24bf5e7949e55b8999735"),
   customerId: ObjectId("5ca24c2de7949e55b8999736")
})

db.purchase.drop();

db.users.updateOne({}, {$set: {purchaselist: [{purchaseId:  ObjectId("5ca24bf5e7949e55b8999735")}]}})
```