**Superhero Sightings**

(Back End Application, using Spring Boot with JPA)

[DATABASE 1](#_Toc2043021334)

[Design Considerations / Relationships 1](#_Toc2009826433)

[SPRING BOOT 1](#_Toc1471342213)

[SPRING DATA JPA 1](#_Toc20530763)

[Entities 2](#_Toc625379464)

[DTOs 2](#_Toc1545305389)

[EXCEPTION HANDLING 2](#_Toc1644958923)

[LOMBOK 2](#_Toc902667234)

[SQL Considerations 3](#_Toc1511261352)

[User Stories: 3](#_Toc1124938104)

[Appendix 1: ERD Diagram 4](#_Toc1568098319)

[Appendix 2: SQL Script – DB 5](#_Toc1008919822)

[Appendix 3: SQL Script – Data 7](#_Toc679192266)

# **DATABASE**

## **Design Considerations / Relationships**

* Supertype (One) - (Many) Super
  + one Super can be only of a Supertype, but a Supertype can represent many Supers – heroes or villains)
* Supertype (One) - (Many) Organization
  + one Organization can only be of one type (superhero or supervillains), but the Supertype can represent many organizations.
* Super (Many) - (Many) Organization
  + a super can belong to many organizations, and an organization can have one or many supers
* Location (One) - (Many) Sighting
  + One location can have many sightings, while a sighting only has a location at a time
* Sighting (Many) – (Many) Super
  + A Sighting can contain one or more Supers, and a Super can be sighted many times

**Note: The Many to Many relationships will create extra table, pulling the PK of either table into the newly constituted 3rd table, where they also act as FK, building therefore new ManytoOne/OnetoMany relationships. (ie : the SuperOrganization and the SuperSightings tables).**

# **SPRING BOOT**

## **SPRING DATA JPA**

* This project uses **ORM to map the java entities objects to the DB SQL tables and columns**. Furthermore, the entities objects are corelated with the DTOs corresponding objects.
* By having the DAO layer of **each DTO extending JpaRepository** interface, Spring JPA is implemented, which in turn will result in no DAO implementation layer being created, as Spring JPA handles the CRUD operations for us. JpaRepository is an interface which takes 2 generics parameters: the entity and the data type of the PK.
* However, all the coding will be required on the Service layer now. The objects and entities will have to be carefully manipulated, called, mapped, copied, the relationship of DTO and corresponding Entity playing the most important role. Aside from this, the in-build methods will be used to perform the CRUD operations **(findAll, findById, saveAndFlush, deleteById).**
* **Object composition** will be reiterated throughout.

## **Entities**

* Super (contains Supertype Object Entity (@ManyToOne), collection of Sightings and collection of Organizations (@ManytoMany))
* Organization (contains SuperType Object (@ManyToOne), and a collection of Supers (@ManytoMany))
* Supertype (hero or villain – within Super and Organization Entities – object composition)
* Location (contains a collection of Sightings (@OnetoMany))
* Sighting ((contains Location Object Entity (@ManyToOne), and a collection of Supers (@ManytoMany)

In Spring JPA, aside from the one direction, the **bidirectional mapping** approach is also applied: an organization can fetch all supers, and all super can show all organizations they belong to. Similar to the location - sightings – supers’ relationship. To connect a location to all supers mapping is connected through sightings, and vice versa, to fetch all locations where a particular super has been seen.

## **DTOs**

* Super
* Organization
* Supertype (hero or villain – applies to both Super and Organization Entities – object composition)
* Location
* Sighting

Note: Entities will reflect DTOs, in terms of attributes and object composition.

## **EXCEPTION HANDLING**

Custom exceptions have been used, thrown in the Service layer (as lacking Dao Implementation class). These are runtime exceptions, meaning they are unchecked (the system will not check for them automatically). The developer is responsible to manage all the errors/ exceptions handling that might occur so that the program responds in a graceful way.

I.E - trying to access a super/organization/location/sighting/location which does not exist.

## **LOMBOK**

Lombok dependency has been added to the pom.xml file to facilitate the DTOs, as well as the Entity objects, being created. In this sense the following annotations have been used throughout the mentioned classes:

@NoArgsConstructor //empty constructor  
@AllArgsConstructor //a constructor using all attributes  
@Getter //for getters  
@Setter //for setters  
@ToString //to use ToString method  
@EqualsAndHashCode // equals and hash code

## **SQL Considerations**

In order to keep track of the SQL queries generated by the ORM framework, application.properties file contains:

spring.jpa.show-sql= true  
spring.jpa.properties.hibernate.format\_sql = true;

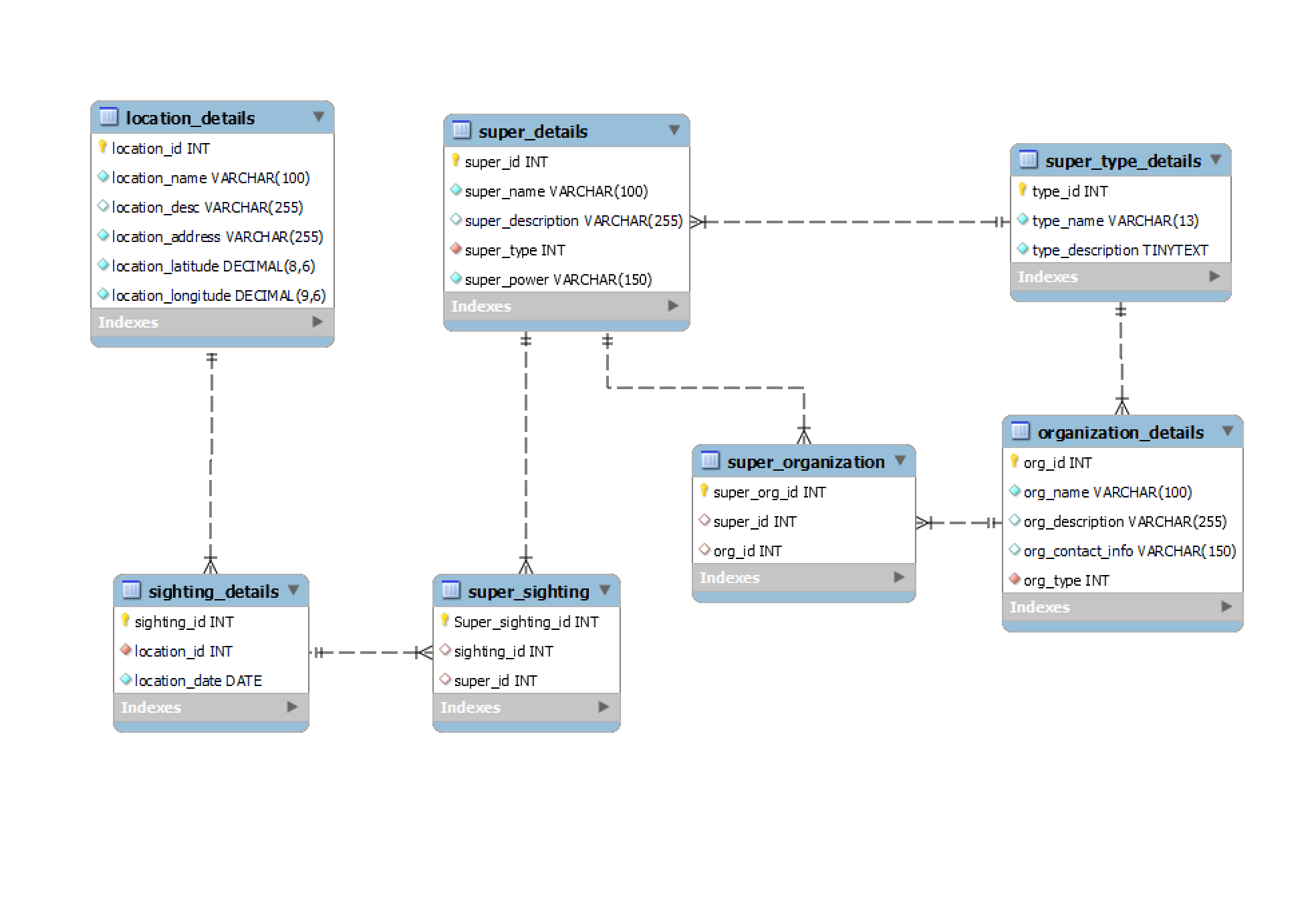
However, these have been commented out before “release” as they are not used in production. (were mainly used for learning purposes)

## **User Stories:**

1. The user can create a new SuperType.
2. The user can read / fetch a SuperType.
3. The user can read / fetch all SuperTypes.
4. The user can update a SuperType.
5. The user can delete a SuperType.
6. The user can create a new Super.
7. The user can read / fetch a Super.
8. The user can read / fetch all Supers.
9. The user can update a Super.
10. The user can delete a Super.
11. The user can create a new Organization.
12. The user can read / fetch an Organization.
13. The user can read / fetch all Organizations.
14. The user can update an Organization.
15. The user can delete an Organization.
16. The user can create a new Location.
17. The user can read / fetch a Location.
18. The user can read / fetch all Locations.
19. The user can update a Location.
20. The user can delete a Location.
21. The user can create a new Sighting.
22. The user can read / fetch a Sighting.
23. The user can read / fetch all Sightings.
24. The user can update a Sighting.
25. The user can delete a Sighting.
26. Record a superhero/supervillain sighting for a particular location and date.
27. Report all superheroes sighted at a particular location.
28. Report all locations where a particular superhero has been seen.
29. Report all sighting (hero and location) for a particular date.
30. Report all members of a particular organization
31. Report all organizations a particular superhero / supervillain belongs to.

**Note: all user stories have been successfully implemented and tested through Postman.**

# Appendix 1: ERD Diagram



# Appendix 2: SQL Script – DB

DROP DATABASE IF EXISTS superhero\_sightings;

CREATE DATABASE superhero\_sightings;

USE superhero\_sightings;

-- type\_details (for heroes/villains and organizations superheroes/supervillains)

CREATE TABLE super\_type\_details(

type\_id INT AUTO\_INCREMENT PRIMARY KEY,

type\_name VARCHAR (13) NOT NULL,

type\_description TINYTEXT NOT NULL);

-- super table (supers can be heros or villains - super\_type\_details)

CREATE TABLE super\_details(

super\_id INT AUTO\_INCREMENT PRIMARY KEY,

super\_name VARCHAR(100) NOT NULL,

super\_description VARCHAR(255),

super\_type INT NOT NULL,

super\_power VARCHAR(150) NOT NULL, -- 1 hero has 1 superpower, and at the same time, the superpower is unique to only 1 superhero

FOREIGN KEY (super\_type) REFERENCES super\_type\_details(type\_id));

-- organization table

CREATE TABLE organization\_details(

org\_id INT auto\_increment PRIMARY KEY,

org\_name VARCHAR(100) NOT NULL,

org\_description VARCHAR(255),

org\_contact\_info VARCHAR(150),

org\_type INT NOT NULL,

FOREIGN KEY (org\_type) REFERENCES super\_type\_details(type\_id));

-- because superhero - organization represents a many-to-many relationship

-- hero\_organization table

CREATE TABLE super\_organization(

super\_org\_id INT AUTO\_INCREMENT PRIMARY KEY,

super\_id INT,

org\_id INT,

FOREIGN KEY (super\_id) REFERENCES super\_details(super\_id),

FOREIGN KEY (org\_id) REFERENCES organization\_details(org\_id));

-- location table

CREATE TABLE location\_details(

location\_id INT AUTO\_INCREMENT PRIMARY KEY,

location\_name VARCHAR (100) NOT NULL,

location\_desc VARCHAR(255),

location\_address VARCHAR (255) NOT NULL,

location\_latitude DECIMAL (8,6) NOT NULL,

location\_longitude DECIMAL (9,6) NOT NULL);

-- sighting details table

CREATE TABLE sighting\_details(

sighting\_id INT AUTO\_INCREMENT PRIMARY KEY,

location\_id INT NOT NULL,

sighting\_date DATE NOT NULL,

FOREIGN KEY (location\_id) REFERENCES location\_details(location\_id));

-- because superhero - sighting represents a many-to-many relationship

-- superhero\_sighting table

DROP TABLE if exists super\_sighting;

CREATE TABLE super\_sighting(

super\_sighting\_id INT AUTO\_INCREMENT PRIMARY KEY,

sighting\_id INT,

super\_id INT,

-- FK CONSTRAINTS

FOREIGN KEY (super\_id) REFERENCES super\_details(super\_id),

FOREIGN KEY (sighting\_id) REFERENCES sighting\_details(sighting\_id));

# Appendix 3: SQL Script – Data

-- insert data in the super\_type\_details table

INSERT INTO super\_type\_details (type\_name, type\_description) VALUES

('Superhero', 'Superheroes are characterized by their moral code, strong sense of duty, and their willingness to use their powers for the greater good. They fight for justice, protect the innocent, and combat supervillains and other threats to society.'),

('Supervillain','Supervillains use their abilities for nefarious and criminal purposes, engaging in activities such as committing crimes, seeking world domination, or causing chaos and destruction.');

-- insert data in the super\_details table

INSERT INTO super\_details (super\_name, super\_description,super\_type,super\_power) VALUES

('Storm', 'As one of the X-Men, she uses her weather-controlling powers to fight against villains and protect both mutants and humans.',1, 'Weather manipulation'),

('Loki', 'Loki, Prince of Asgard, his tricks and schemes wreak havoc across the realms.', 2, 'God of Mischief') ,

('Spider-Man', 'Your friendly neighbourhood hero.', 1, 'Spider Powers'),

('Magneto', 'Magneto is an Omega Level Mutant. He is one of the most powerful mutant.', 2, ' Magnokinesis'),

('Wonder-Woman', ' Diana Prince is an Amazonian warrior with superhuman strength, agility, and the Lasso of Truth, which compels honesty.', 1, 'Superhuman'),

('Jean', 'Powerful level 5 mutant, with extraordinary telekinesis abilities',1,'Telekinesis'),

('Flash', 'Barry Allen can move at superhuman speeds, allowing him to outrun almost anything and travel through time.',1, 'Superspeed'),

('Thor', 'The God of Thunder can summon and control lightning, wield the enchanted Mjolnir hammer, and has superhuman strength.',1,'Control over Lightning and Thunder'),

('Mystique', 'Mystique has the ability to change her appearance, mimic anyones physical appearance and voice.', 2, 'Shape-Shifting'),

('Sandman', 'The Sandman (Flint Marko) can transform his body into sand, granting him incredible shapeshifting and sand-based attacks.',2,'Sand Manipulation'),

('Poison', 'Poison Ivy (Pamela Isley) has the ability to control and manipulate plants, using them as weapons or for various purposes.', 2,' Control over Plant Life');

-- insert data into the organization\_details table

INSERT INTO organization\_details(org\_name,org\_description,org\_contact\_info,org\_type) VALUES

('X-Men', ' The X-Men are known for their unique powers and their struggle for mutant rights.','Marvel Universe',1),

('Avengers','Earths Mightiest Heroes','Marvel Universe',1),

('The Brotherhood of Mutants', 'Brotherhood advocates for mutant supremacy over humans and often clashes with the X-Men','Comic Books',2),

('Justice League','The Justice League is dedicated to defending Earth from powerful supervillains and cosmic threats.','DC Universe',1),

('Sinister Six', 'This group is composed of six of Spider-Mans most formidable foes who come together to defeat their common enemy', 'Marvel Comics', 2),

('Legion of Doom', 'Legion of Doom is a gathering of some of DCs most notorious villains','DC Comics',2);

-- insert into superhero\_organization (after inserting in each separate table - to get the IDs)

INSERT INTO super\_organization(super\_id, org\_id) VALUES

(4,3),

(9,3),

(1,1),

(6,1),

(2,6),

(11,6),

(8,2),

(5,4),

(7,4),

(10,5),

(3,1),

(3,2);

-- insert data in the location\_details table

INSERT INTO location\_details(location\_name, location\_desc, location\_address,location\_latitude,location\_longitude) VALUES

('Mystic Falls','A small, picturesque town nestled in a valley surrounded by dense forests. Known for its mysterious history and supernatural occurrences.','123 Elm Street, Mystic Falls, USA', 40.724583,-74.180832 ),

('Ironclad Tower','A massive, imposing fortress made of iron and stone, guarding a strategic mountain pass.','Ironclad Road, Citadel Peak, Ironrealm',38.987654,-82.345678),

('Starlight Observatory', 'A state-of-the-art observatory situated on a remote mountain peak, offering breathtaking views of the night sky.','Observatory Road, Starlight Peak, Astronomy Valley',-32.345678,118.901234),

('Phoenix Plaza','The bustling heart of a modern metropolis, with towering skyscrapers, neon lights, and a vibrant nightlife.','789 Downtown Avenue, Phoenix City, Metropolis',34.567890 ,-112.345678),

('Emerald Bay','Seaside with pristine beaches, crystal-clear waters, and a lush tropical landscape.','456 Oceanview Drive, Emerald Bay, Paradise Island', -23.567891, 152.987654),

('Maplewood Park', 'A serene urban park with a scenic pond, walking trails, and a playground for families to enjoy.','123 Maple Avenue, Maplewood, USA',40.712345,-74.123456);

-- insert data in the sighting\_details table

INSERT INTO sighting\_details (location\_id, sighting\_date) VALUES

(1,'2020-10-10'),

(2,'2020-10-10'),

(3,'2022-01-15'),

(4,'2023-01-01'),

(5,'2022-12-12'),

(6,'2008-08-08'),

(1,'2018-05-05'),

(2,'2023-04-04'),

(3,'2009-09-09'),

(4,'2007-07-07');

-- insert data into superhero\_sighting

INSERT INTO super\_sighting(sighting\_id, super\_id) VALUES

(1,1),

(1,2),

(1,5),

(2,8),

(2,4),

(3,3),

(4,4),

(5,5),

(6,6),

(7,7),

(8,8),

(9,9),

(10,10);