

Giacomo Corbett Englaro.

For the scenario below identify the entities, their attributes and appropriate keys

### Finsbury Happy Zoo

Finsbury Happy Zoo's concept is to show animals together in their habitats. They have a number of enclosures of different habitat types (such as forest or tundra), different sizes (square metres), each having a main feature (such as a stream or a cave). Animals of different species share the same enclosure. Each enclosure has a unique number and there can be several enclosures with the same habitat but with a different main feature or of a different size.

Each animal has a unique ID, and their name, date\_of\_birth, diet and description are stored. When an animal is put in an enclosure, the start date is recorded, and if they are transferred to another enclosure the end date is recorded.

Zoo keepers may need to make a note about a particular animal, for example "not eating well today" and this is recorded along with the date.

To make sure the animals don't eat each other a species compatibility table is maintained which has the following information; speciesA, speciesB, compatibility\_rating (5 for happy neighbours to 1 for bitter enemies).

Species are identified by their name, and a description of the species and their habitat type are recorded.

Species are matched against enclosures by Zoo staff, and if suitable the maximum number of animals of a particular species for a particular enclosure is recorded to prevent overcrowding.

A question: should animals and species be linked in any way?

Creating the enclosure table

```
CREATE TABLE enclosure (  
  Enclosure_ID INTEGER PRIMARY KEY  
  Habitat_type VARCHAR(30),  
    Size INTEGER,  
    Main_feature VARCHAR(30),  
  )
```

Now creating a table for animals

Enclosure to animals will be a MANY to MANY, we will need a link table.

Since they can move between enclosures, the link table helps to track their history.

```
CREATE TABLE animal (  
  Animal_ID INTEGER PRIMARYKEY  
  Name VARCHAR(20)  
  Date_birth DATE,  
  Diet VARCHAR (30),  
  Description VARCHAR(50)
```

```
CREATE TABLE animal_enclosure(
```

```

Animal_ID INTEGER
Enclosure_ID INTEGER
Start_date DATE,
End_date DATE,
PRIMARYKEY (animal_id, enclosure_id)
FOREIGN KEY animal_id REFERENCES animal(Animal_id)
FOREIGN KEY enclosure_id REFERENCES enclosure(enclosure_id)

```

For the zoo keeper notes...

```

CREATE TABLE zoonotes (
Note_ID INTEGER PRIMARYKEY
Animal_ID INTGER
Date_ID DATE,
Note VARCHAR (100)
FOREIGNKEY Animal_ID REFERENCES animal (animal_ID)
)

```

```

CREATE TABLE species (
Name VARCHAR(20) PRIMARYKEY,
Description VARCHAR(40)
Habitat VARCHAR(40)

```

Now we need to do the compatibility rate

This is gonna be a many to many between species,

Since species are identified by their name, we use the species\_name as primary keys.

```

CREATE TABLE compatibility(
Species_a VARCHAR(20)
Species_b VARCHAR(20)
Compatibility_rating INTEGER
PRIMARYKEY(species_a, species_b)
FOREIGN KEY species_a REFERENCES species(Name)
FOREIGN KEY species_b REFERENCES Species(Name)

```

Since the enclosure limit will be a many to many, it will need a link table with a composite key.

```

CREATE TABLE enclosure_limit_species(
Enclosure_ID INTEGER,
Species_name VARCHAR (20)
Max_number INTEGER
PRIMARY KEY (enclosure_id, species_name)
FOREIGN KEY enclosure_id REFERENCES animal_enclosure(enclosure_id)
FOREIGN KEY species name REFERENCES species(species_name)
)

```