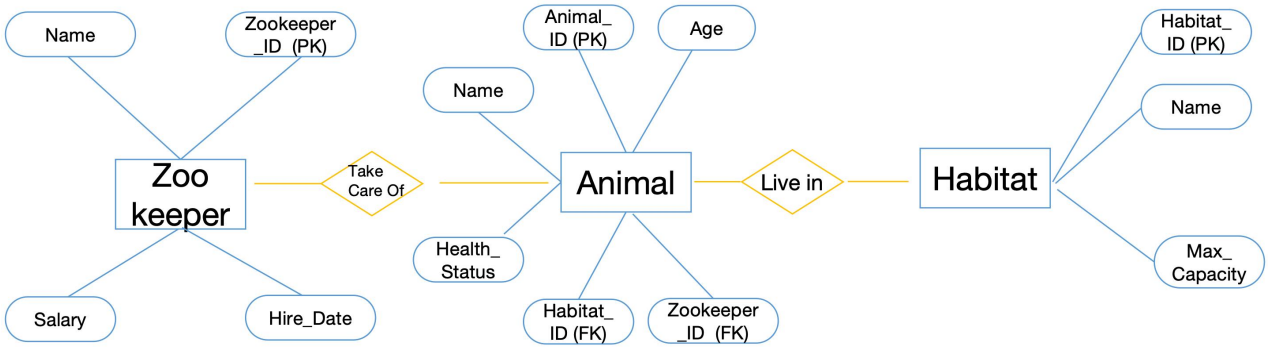
ER Diagram: Zoo Management System



1. Entities and Attributes

1.1 Animal

Animal_ID (Primary Key): This is a unique identifier for each animal.

Name: The name of the animal (e.g., "Leo" the lion).

Age: The age of the animal, stored as an integer.

Health_Status: The health condition of the animal (e.g., Healthy, Sick, Injured).

Habitat_ID (Foreign Key): This links the animal to its habitat.

Zookeeper_ID (Foreign Key, Unique): This links the animal to the zookeeper responsible for it.

1.2 Habitat

Habitat_ID (Primary Key): A unique identifier for each habitat.

Name: The name of the habitat.

Max_Capacity: The maximum number of animals that can live in the habitat. This is stored as an integer.

1.3 Zookeeper

Zookeeper_ID (Primary Key): A unique identifier for each zookeeper.

Name: The name of the zookeeper.

Salary: The salary of the zookeeper, stored as a float (e.g., 50000.50).

Hire_Date: The date the zookeeper was hired, stored as a date (e.g., 2023-01-15).

2. Relationships

- **Animal - Habitat (Many-to-One)**

Each animal lives in one habitat. Each habitat can house many animals.
The Habitat_ID in the Animal table acts as a Foreign Key to link animals to their habitats.

- **Animal - Zookeeper (Many-to-One)**

Each animal will be assigned one zookeeper to be taken care. Each zookeeper can take care of many animals.
The Zookeeper_ID in the Animal table acts as a Foreign Key to link animals to their zookeepers.

3. Constraints

- **Primary Keys (PK)**

Each entity has a unique identifier (Animal_ID, Habitat_ID, Zookeeper_ID) to prevent duplicate records and ensure data integrity.

- **Foreign Keys (FK)**

Habitat_ID in the Animal table ensures that every animal is assigned to a valid habitat.
Zookeeper_ID in the Animal table ensures that every animal is assigned to a valid zookeeper.

- **Max_Capacity Constraint**

The Max_Capacity attribute in the Habitat table ensures that the number of animals in a habitat does not exceed its limit. This helps maintain the well-being of the animals.

- **Data Types**

Age is an integer, Salary is a float, and Hire_Date is a date to ensure accurate and appropriate data storage.

4. Why These Decisions?

- **Why Entities?**

Animal: This is the core of the zoo. Each animal has unique attributes like name, species, and health status.

Habitat: Animals need a place to live, and habitats help organize them logically.

Zookeeper: Zookeepers are responsible for maintaining habitats and caring for animals.

- **Why Relationships?**

Animal - Habitat: Animals must live somewhere, and habitats need animals to be functional.

Animal - Zookeeper: Animals must be taken care by someone.

- **Why These Constraints?**

Primary Keys:
Primary keys like Animal_ID, Habitat_ID, and Zookeeper_ID ensure that each record in the table is unique. This prevents duplication and makes it easy to identify specific animals, habitats, or zookeepers. For example, without a primary key, two animals named "Leo" could cause confusion.

Foreign Keys:
Foreign keys like Habitat_ID and Zookeeper_ID in the Animal table maintain relationships between entities. They ensure that every animal is assigned to a valid habitat and cared for by a valid zookeeper. This prevents orphaned records, like an animal without a habitat or zookeeper, which would break the system's logic.

Max_Capacity:
The Max_Capacity in the Habitat table ensures that habitats don’t get overcrowded. Overcrowding can harm animals and make habitats harder to manage. For example, a habitat designed for 10 lions shouldn’t have 15.

Data Types:
Using the correct data types ensures data is stored accurately and efficiently. For example:
Age is an integer because age is a whole number.
Salary is a float to handle decimal values (e.g., $50,000.50).
Hire_Date is a date to store employment dates in a standard format.