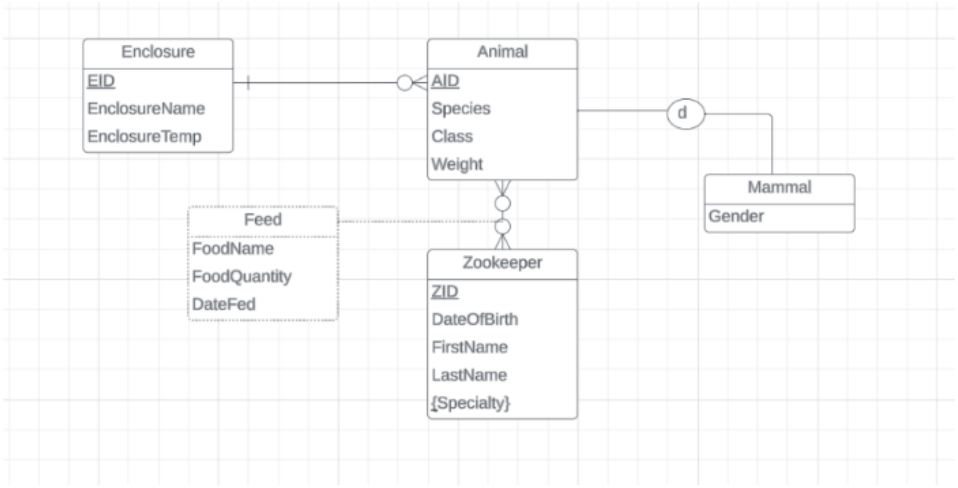
Karson Bastug

Zoo Database

ERD Diagram



Conversion

Enclosure (EID, EnclosureName, EnclosureTemp)

Animal (AID, Species, Class, Weight, EID)

Zookeeper (ZID, DateOfBirth, FirstName, LastName)

Feed (FoodName,FoodQuantity,DateFed, ZID, AID)

Specialty (Specialty, ZID)

Mammal (AID, Gender)

Zoo Database Tables

```
mysql> describe Animal;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| AID     | int         | NO   | PRI | NULL    |       |
| Species | varchar(10) | YES  |     | NULL    |       |
| Class   | varchar(10) | YES  |     | NULL    |       |
| Weight  | int         | NO   |     | NULL    |       |
| EID     | int         | YES  | MUL | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

```
mysql> select * from Animal;
+-----+-----------+---------+--------+------+
| AID | Species   | Class   | Weight | EID  |
+-----+-----------+---------+--------+------+
|   1 | Penguin   | Bird    |     30 |    1 |
|   2 | Penguin   | Bird    |     35 |    1 |
|   3 | Penguin   | Bird    |     40 |    1 |
|   4 | Polar Bear | Mammal |    900 |    1 |
|   5 | Polar Bear | Mammal |    600 |    1 |
|   6 | Seal      | Mammal  |    600 |    1 |
|   7 | Lion      | Mammal  |    420 |    2 |
|   8 | Lion      | Mammal  |    400 |    2 |
|   9 | Lion      | Mammal  |    300 |    2 |
|  10 | Elephant  | Mammal  |  13000 |    2 |
|  11 | Elephant  | Mammal  |  12500 |    2 |
|  12 | Elephant  | Mammal  |  12000 |    2 |
|  13 | Elephant  | Mammal  |  11000 |    2 |
|  14 | Elephant  | Mammal  |  12500 |    2 |
|  15 | Elephant  | Mammal  |  11500 |    2 |
|  16 | Elephant  | Mammal  |  13500 |    2 |
|  17 | Snake     | Reptile |     15 |    3 |
|  18 | Snake     | Reptile |     15 |    3 |
|  19 | Jaguar    | Mammal  |    150 |    3 |
|  20 | Monkey    | Mammal  |     30 |    3 |
|  21 | Monkey    | Mammal  |     35 |    3 |
|  22 | Toucan    | Bird    |      1 |    3 |
|  23 | Toucan    | Bird    |      2 |    3 |
+-----+-----------+---------+--------+------+
23 rows in set (0.00 sec)
```

```
mysql> describe Enclosure;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EID           | int         | NO   | PRI | NULL    |       |
| EnclosureName | varchar(10) | YES  |     | NULL    |       |
| EnclosureTemp | int         | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

```
mysql> select * from Enclosure;
+-----+---------------+---------------+
| EID | EnclosureName | EnclosureTemp |
+-----+---------------+---------------+
|   1 | Arctic        |             0 |
|   2 | Savannah      |            80 |
|   3 | Tropic        |            90 |
|   4 | Desert        |            90 |
|   5 | Aquatic       |            60 |
+-----+---------------+---------------+
```

```
mysql> describe Feed;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| AID          | int         | NO   | PRI | NULL    |       |
| ZID          | int         | NO   | PRI | NULL    |       |
| FoodName     | varchar(10) | YES  |     | NULL    |       |
| FoodQuantity | int         | NO   |     | NULL    |       |
| DateFed      | date        | NO   | PRI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
mysql> select * from Feed;
+-----+-----+----------+--------------+------------+
| AID | ZID | FoodName | FoodQuantity | DateFed    |
+-----+-----+----------+--------------+------------+
|   1 |   6 | Fish     |            2 | 2023-10-01 |
|   2 |   6 | Fish     |            2 | 2023-10-01 |
|   3 |   6 | Fish     |            2 | 2023-10-01 |
|   4 |   5 | Fish     |           20 | 2023-10-01 |
|   5 |   5 | Fish     |           15 | 2023-10-01 |
|   6 |   1 | Fish     |            2 | 2023-10-02 |
|   6 |   4 | Fish     |           10 | 2023-10-01 |
|   7 |   1 | Beef     |           25 | 2023-10-01 |
|   8 |   1 | Beef     |           27 | 2023-10-01 |
|   9 |   1 | Beef     |           20 | 2023-10-01 |
|  10 |   2 | PlantMix |          140 | 2023-10-01 |
|  11 |   2 | PlantMix |          150 | 2023-10-01 |
|  12 |   2 | PlantMix |          100 | 2023-10-01 |
|  13 |   2 | PlantMix |          110 | 2023-10-01 |
|  14 |   2 | PlantMix |           90 | 2023-10-01 |
|  15 |   2 | PlantMix |          110 | 2023-10-01 |
|  16 |   2 | PlantMix |          160 | 2023-10-01 |
|  17 |   3 | Rat      |            1 | 2023-10-01 |
|  18 |   3 | Rat      |            1 | 2023-10-01 |
|  19 |   2 | Beef     |           50 | 2023-10-01 |
|  20 |   3 | Banana   |            8 | 2023-10-01 |
|  21 |   3 | Banana   |           12 | 2023-10-01 |
|  22 |   1 | SeedMix  |            1 | 2023-10-01 |
|  23 |   1 | SeedMix  |            1 | 2023-10-01 |
+-----+-----+----------+--------------+------------+
```

```
mysql> describe Mammal;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| AID    | int         | YES  | MUL | NULL    |       |
| Gender | varchar(10) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

```
mysql> select * from Mammal;
+------+--------+
| AID  | Gender |
+------+--------+
|    4 | Male   |
|    5 | Female |
|    6 | Male   |
|    7 | Male   |
|    8 | Male   |
|    9 | Female |
|   10 | Male   |
|   11 | Female |
|   12 | Female |
|   11 | Female |
|   13 | Female |
|   14 | Female |
|   15 | Female |
|   16 | Male   |
|   19 | Female |
|   20 | Female |
|   21 | Male   |
+------+--------+
```

```
mysql> describe Specialty;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| ZID       | int         | NO   | PRI | NULL    |       |
| Specialty | varchar(10) | NO   | PRI | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

```
mysql> select * from Specialty;
+-----+-----------+
| ZID | Specialty |
+-----+-----------+
|   1 | Diet      |
|   2 | Health    |
|   3 | Health    |
|   4 | Behavior  |
|   5 | Behavior  |
|   6 | Climate   |
|   6 | Diet      |
+-----+-----------+
```

```
mysql> describe Zookeeper;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| ZID         | int         | NO   | PRI | NULL    |       |
| DateOfBirth | date        | YES  |     | NULL    |       |
| FirstName   | varchar(10) | YES  |     | NULL    |       |
| LastName    | varchar(10) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
mysql> select * from Zookeeper;
+-----+-------------+-----------+----------+
| ZID | DateOfBirth | FirstName | LastName |
+-----+-------------+-----------+----------+
|   1 | 2003-01-03  | Karson    | Bastug   |
|   2 | 2000-03-05  | Billy     | Smith    |
|   3 | 2001-02-25  | Kaleigh   | Ferland  |
|   4 | 1948-08-20  | Robert    | Plant    |
|   5 | 1946-09-05  | Freddie   | Mercury  |
|   6 | 1982-12-25  | Nick      | Claus    |
+-----+-------------+-----------+----------+
```

Seven Queries

Create a query that includes at least two INNER JOINS (i.e. spans at least three tables), a search criterion, and orders the resulting data.

During the summer, our Arctic enclosure becomes the most popular, as it's the coldest and people will go there to cool down. We need to know how many Zookeepers are assigned to Arctic animals so we know how many we can station there. From there, we need to order them based on the classification of the animals they are looking after so we know how many for each exhibit.
Query Results

```
mysql> SELECT Z.ZID, Z.FirstName, A.AID, A.Class
    -> from Enclosure as E
    -> inner join Animal as A on E.EID=A.EID
    -> inner join Feed as F on A.AID=F.AID
    -> inner join Zookeeper as Z on F.ZID=Z.ZID
    -> WHERE E.ENclosureName="Arctic"
    -> ORDER BY Z.FirstName, A.Class;
+-----+-----------+-----+---------+
| ZID | FirstName | AID | Class   |
+-----+-----------+-----+---------+
|   5 | Freddie   |   4 | Mammal  |
|   5 | Freddie   |   5 | Mammal  |
|   1 | Karson    |   6 | Mammal  |
|   6 | Nick      |   1 | Bird    |
|   6 | Nick      |   2 | Bird    |
|   6 | Nick      |   3 | Bird    |
|   4 | Robert    |   6 | Mammal  |
+-----+-----------+-----+---------+
7 rows in set (0.00 sec)
```

Create a query that includes a calculation (e.g. average, sum, etc.).
During a tour, a visitor asks the average amount of food the lions were fed today (assuming the day asked was October 1st)

Query Results

```
mysql> select AVG(F.FoodQuantity) as "Total Fed To Lions"
    -> from Animal as A
    -> inner join Feed as F on A.AID=F.AID
    -> WHERE A.Species = "Lion" AND F.DateFed="2023-10-01";
+--------------------+
| Total Fed To Lions |
+--------------------+
|            24.0000 |
+--------------------+
1 row in set (0.00 sec)
```

Create a query that includes at least one RIGHT or LEFT join.
The manager wants to know what animals are being treated by what Specialists

Query Results

```
mysql> SELECT Z.ZID, Z.Specialty, F.AID
    -> FROM Specialty as Z
    -> LEFT JOIN Feed as F
    -> ON F.ZID=Z.ZID;
+-----+-----------+------+
| ZID | Specialty | AID  |
+-----+-----------+------+
|   1 | Diet      |    6 |
|   1 | Diet      |    7 |
|   1 | Diet      |    8 |
|   1 | Diet      |    9 |
|   1 | Diet      |   22 |
|   1 | Diet      |   23 |
|   2 | Health    |   10 |
|   2 | Health    |   11 |
|   2 | Health    |   12 |
|   2 | Health    |   13 |
|   2 | Health    |   14 |
|   2 | Health    |   15 |
|   2 | Health    |   16 |
|   2 | Health    |   19 |
|   3 | Health    |   17 |
|   3 | Health    |   18 |
|   3 | Health    |   20 |
|   3 | Health    |   21 |
|   4 | Behavior  |    6 |
|   5 | Behavior  |    4 |
|   5 | Behavior  |    5 |
|   6 | Climate   |    1 |
|   6 | Climate   |    2 |
|   6 | Climate   |    3 |
|   6 | Diet      |    1 |
|   6 | Diet      |    2 |
|   6 | Diet      |    3 |
+-----+-----------+------+
27 rows in set (0.00 sec)
```

Create a query that properly uses a GROUP BY to perform a calculation on information spanning at least two tables.

Find the number of animals in each exhibit

Query Results

```
mysql> SELECT E.EID, E.EnclosureName, COUNT(A.AID) as NumberofAnimals
    -> from Enclosure as E
    -> inner join Animal as A on E.EID=A.EID
    -> GROUP BY E.EID, E.EnclosureName;
+-----+--------------+-----------------+
| EID | EnclosureName | NumberofAnimals |
+-----+--------------+-----------------+
|   1 | Arctic       |               6 |
|   2 | Savannah     |              10 |
|   3 | Tropic       |               7 |
+-----+--------------+-----------------+
3 rows in set (0.00 sec)
```

Create a query that includes a GROUP BY, but selects items from the results of the GROUP BY query and lists them in order (e.g. using our COOKIE example, list everyone who has spent over $100 in total).

Your manager wants to measure the animals that are consuming the most food to look for more ways to feed them. List every animal that was fed more than 50 pounds today (assuming the date is October 1st 2023)

Query Results

```
mysql> select SUM(F.FoodQuantity) as TotalEaten, A.Species, A.AID
    -> from Animal as A
    -> inner join Feed as F on A.AID=F.AID
    -> WHERE F.FoodQuantity > 50 AND F.DateFed="2023-10-01"
    -> GROUP BY A.Species, A.AID
    -> ORDER BY TotalEaten DESC;
+------------+----------+-----+
| TotalEaten | Species  | AID |
+------------+----------+-----+
|        160 | Elephant |  16 |
|        150 | Elephant |  11 |
|        140 | Elephant |  10 |
|        110 | Elephant |  13 |
|        110 | Elephant |  15 |
|        100 | Elephant |  12 |
|         90 | Elephant |  14 |
+------------+----------+-----+
7 rows in set (0.00 sec)
```

Create a query that includes a useful nested query.

A visitor has very little time to visit our Zoo, and so they want to visit the exhibit with the largest variety of animals. Find the amount of different animal's within every exhibit.

Query Results

```
mysql> SELECT E.EID, E.EnclosureName,
    -> (SELECT COUNT(Distinct A.Species) from Animal as A
    -> WHERE A.EID=E.EID) as Number
    -> FROM Enclosure as E;
+------+---------------+--------+
| EID  | EnclosureName | Number |
+------+---------------+--------+
|    1 | Arctic        |      3 |
|    2 | Savannah      |      2 |
|    3 | Tropic        |      4 |
|    4 | Desert        |      0 |
|    5 | Aquatic       |      0 |
+------+---------------+--------+
5 rows in set (0.00 sec)
```

ADDITIONAL: Queries used to create tables

create table **Animal** (AID int not null primary key, Species varchar(10), Class varchar(10), Weight int not null, EID int, foreign key (EID) references Enclosure(EID));

create table **Enclosure** (EID int not null primary key, EnclosureName varchar(10), EnclosureTemp int not null);

Create table **Zookeeper** (ZID int not null primary key, DateofBirth DATE, FirstName varchar(10), LastName varchar(10));

Create table **Feed** (AID INT, ZID INT, FoodName varchar(10),FoodQuantity int not null, DateFed DATE, primary key (AID, ZID, DateFed), foreign key (AID) references Animal(AID), foreign key (ZID) references Zookeeper(ZID));

Create table **Mammal** (AID INT, Gender varchar(10), foreign key (AID) references Animal(AID));

Create table **Specialty** (ZID int, Specialty varchar(10), primary key (ZID, Specialty), foreign key (ZID) references Zookeeper(ZID));