**Name and surname: Kinga Chodorowska**
**Index: 184549**

# The topic of the project:

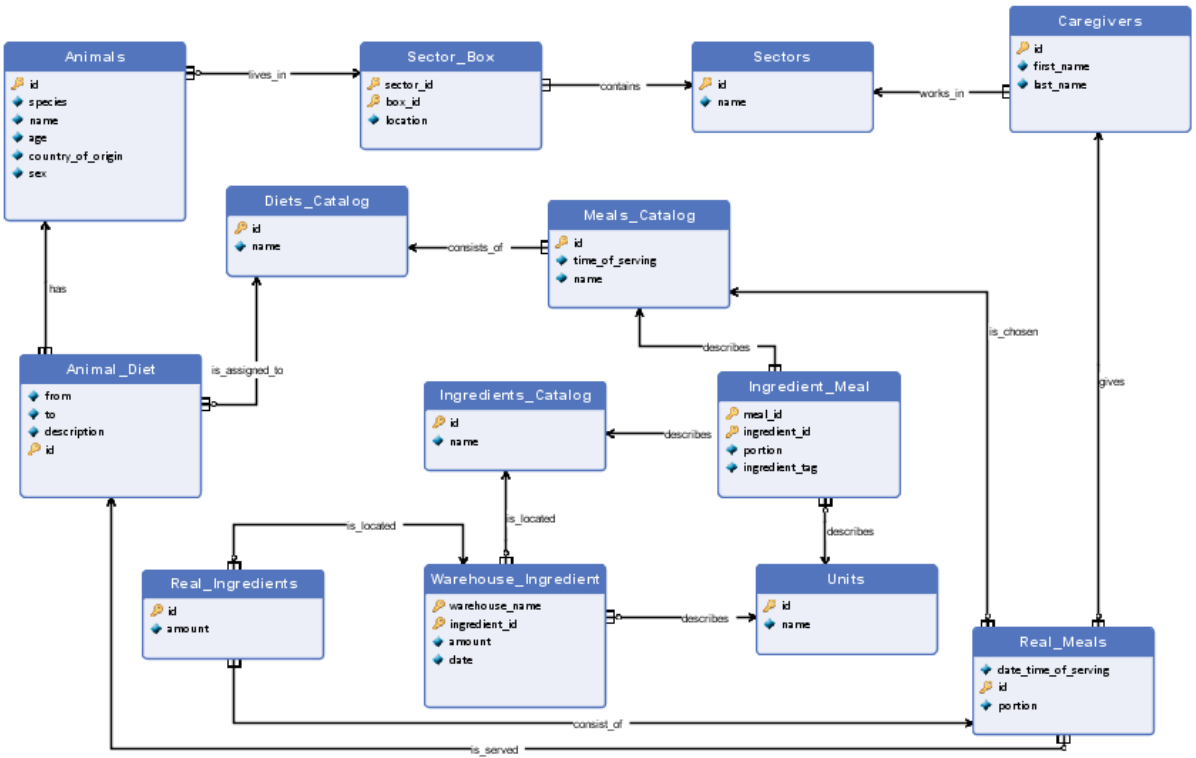
7. Zoo garden registers animals and their diets.

# Assumptions

**Client:**
The client is the owner/manager of the zoo.
**User:**
Users are animal caregivers.
**Purpose of the database:**
The purpose of the database is to hold information on diets and animals to improve zoo operations.
**Description:**
The database should allow us to search for animals, their diets, and the meals they need to be fed, and it should allow us to schedule meals and check inventory.
**Limitations:**
There is no information in the database about employees' salaries or how long they have worked.
**Possible scenarios of database use:**

- **caregiver feeds the animals** - to do this, the caregiver creates a new entity real_meals, gives the exact date, number of servings of food, chooses one of the stored diets and gives the ingredients used, (the system changes the number of components in the warehouse by the specified amount),
- **caregiver changes the diet -** to change the diet of a selected animal, the maintainer creates a new entity animal_diet and puts there the information how long the diet will last and a short description,
- **searching for diets and meals,**
- **checking the warehouse,**
- **finding where the chosen animal is (in which sector and box).**

# Diagram ERD:




# Description of the entity set

| **Animals** | | | |
|---|---|---|---|
| Animals is an entity set containing information about animals;<br>an entry is added when a new animal is in zoo, and might be deleted only if animal die or leave the zoo; ~ 400 entries; | | | |
| Name: | Primary key | Type/domain | Description |
| id | Yes | int, from 1 to 400, increases in 1 | a unique identifier for the animal |
| species | No | varchar, first letter is capital letter, a-ż and spaces are allowed, number of characters<=20 | species of animal |
| name | No | varchar, a-ż and spaces are allowed, first | name of animal |

| | | | |
|---|---|---|---|
| | | letter and first letter after space are capital letters, number of characters<=15 | |
| age | No | float, from 0 to 400, two decimal places | age of the animal |
| country_of_origin | No | varchar, a-ż and spaces are allowed, first letter and first letter after space are capital letters, number of characters<=20 | country of origin of the animal |
| sex | No | varchar, only female or male | sex of the animal |

| **Sector_Box** | | | |
|---|---|---|---|
| Sector_Box is an entity set of locations, it tells where the animal is located; an entry is added when a new box or sector is created, and might be deleted only if we want to delete box or sector from the zoo ~ 20 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| box_id | Yes | int, from 1 to 20, increases in 1 | a unique identifier for the box |
| sector_id | Yes | int, from 1 to 12, increases in 1 | a unique identifier for the sector |
| location | No | varchar, (world directions: north, south, east, west) | location of the box in a given sector |

| **Sectors** | | | |
|---|---|---|---|
| Sectors is an entity set of sectors; an entry is added when a new sector is created, and might be deleted only if we want to delete sector from the zoo ~ 12 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| id | Yes | int, from 1 to 12, increases in 1 | a unique identifier for the sector |

| | | | |
|---|---|---|---|
| name | No | varchar (only small letters a-ż), number of characters<=20 | name of the sector |

| **Caregivers** | | | |
|---|---|---|---|
| Caregivers is an entity set containing information about caregivers; an entry is added when a new caregiver is employed, and might be deleted only if caregiver stops working in the zoo ~ 50 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| id | Yes | int, from 1 to 50, increases in 1 | a unique identifier for the caregiver |
| first_name | No | varchar, a-ż and spaces are allowed, first letter and first letter after space are capital letters, number of characters<=20 | the first name of the caregiver |
| last_name | No | varchar, a word starting with a capital letter, a hyphen is allowed in the middle of the word, then the following letter is also capital, a-ż, number of characters<=20 | the last name of the caregiver |

| **Animal_Diet** | | | |
|---|---|---|---|
| Animal_Diet is an entity set containing information about current diet duration; an entry is added when an animal get new diet, and might be deleted only if animal runs out of diet~ 400 entries; | | | |
| Name: | Primary key | Type/domain | Description |
| from | No | date, format dd-mm-yyyy | date of starting the diet |
| to | No | date, format dd-mm-yyyy also NULL this means that the diet does not have an end | date of ending the diet |

| | | | |
|---|---|---|---|
| | | date | |
| description | No | text, it is allowed to use all letters of the Polish alphabet, punctuation marks, digits and capital letters, number of characters<=150 | unique recommendations for your animal's diet |
| id | Yes | int, from 1 to 400, increases in 1 | a unique identifier for the diet of animal |

| **Diets_Catalog** | | | |
|---|---|---|---|
| Diets_Catalog is an entity set containing diets' names; an entry is added when new diet is created, and might be deleted only if we want to delete a diet~ 50 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| id | Yes | int, from 1 to 50, increases in 1 | a unique identifier for the diet |
| name | No | varchar, a-ż, number of characters<=30 | name of the diet |

| **Meals_Catalog** | | | |
|---|---|---|---|
| Meals_Catalog is an entity set containing information about meals; an entry is added when new meal is created, and might be deleted if we want to delete meal from diet and when we delete a diet, all entries of Meals_Catalog of deleted diets will be also deleted~ 500 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| id | Yes | int, from 1 to 500, increases in 1 | a unique identifier for the meal |
| time_of_serving | No | varchar, (morning, noon, afternoon, evening, night, midnight) | recommended time of day for serving a meal |
| name | No | varchar, a-ż and spaces are allowed, number of characters<=20 | name of the meal |

| **Ingredient_Meal** | | | |
|---|---|---|---|
| Ingredient_Meal is an entity set containing information about specific ingredients in specific diets; an entry is added when a new ingredient proposition is added to a meal, and might be deleted if we don't want this ingredient in this meal and also if we delete an ingredient; ~ 500 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| meal_id | Yes | int, from 1 to 500, increases in 1 | a unique identifier for the meal |
| ingredient_id | Yes | int, from 1 to 500, increases in 1 | a unique identifier for the ingredient |
| portion | No | float, using dot, only two decimal places(ex. 1.70) | average portion of the ingredient in this meal |
| ingredient_tag | No | varchar, a-ż, number of characters=2 | a tag for a group of ingredients which can be replaced with one of them |

| **Ingredients_Catalog** | | | |
|---|---|---|---|
| Ingredients_Catalog is an entity set containing information about diets; an entry is added when a new ingredient is created, and might be deleted only if we don't need this ingredient any longer~ 500 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| id | Yes | int, from 1 to 500, increases in 1 | a unique identifier for the ingredient |
| name | No | varchar, a-ż and spaces are allowed, number of characters<=20 | name of the ingredient |

| **Units** | | | |
|---|---|---|---|
| Units is an entity set containing basic information about units; an entry is added when a new unit is created, and might be deleted only if it won't be needed any longer ~ 10 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| id | Yes | int, from 1 to 10, increases in 1 | a unique identifier for the unit |

| | | | |
|---|---|---|---|
| name | No | varchar, a-ż and 0-9, * / % ^, number of characters<=20 | name of the unit |

| **Warehouse_Ingredient** | | | |
|---|---|---|---|
| Warehouse_Ingredient is an entity set containing information about ingredients stored in warehouses; an entry is added when a new ingredient or warehouse is created, and might be deleted only if the ingredient or warehouse is deleted or if we change location of ingredients to other warehouse and conversely ~ 500 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| warehouse_name | Yes | varchar, a-ż and spaces are allowed, number of characters<=30 | name of the warehouse |
| ingredient_id | Yes | int, from 1 to 500, increases in 1 | a unique identifier for the ingredient |
| amount | No | float, using dot, only two decimal places(ex. 1.70) | amount of ingredients in the indicated unit stored in warehouses |
| date | No | date, format nn-mm-yyyy | expiry date of the ingredient |

| **Real_Ingredients** | | | |
|---|---|---|---|
| Real_Ingredient is an entity set containing information about ingredients used in Real_Meals; an entry is added when an ingredient in the real_meal is created, and might be deleted only if we want to delete an ingredient from real_meal because of error and every real_ingredients are deleted every day at midnight from the day before yesterday ~ 10000 entries; | | | |
| Name: | Primary key | Type/domain | Description: |
| id | Yes | int, from 1 to 10000, increases in 1 | a unique identifier for the real ingredient |
| amount | Yes | float, using dot, only two decimal places(ex. 1.70) | amount of real ingredients in the indicated unit that we will take away from warehouse |

| Real_Meals | | | |
|---|---|---|---|
| Real_Meals is an entity set containing information about meals that are currently served; an entry is added when register a meal for animal, and might be deleted only if we change mind and want to change meal for an animal and every real_ingredients are deleted every day at midnight from the day before yesterday~ 3600 entries; | | | |
| Name: | Primary key | Type/domain: | Description: |
| date_time_of_serving | No | datetime, format hh-min-nn-mm-yyyy | the date and time the meal was served |
| id | Yes | int, from 1 to 3600, increases in 1 | a unique identifier for the real meal |
| portion | No | float, using dot, only two decimal places(ex. 1.70) | is a part of average portion |

# Definition of relationships

| Name: | Entity set 1 | Entity set 2 | Type: | Description: |
|---|---|---|---|---|
| lives_in | Animals | Sector_Box | 0..n : 1 | relationship lives_in between Animals and Sector_Box entities – represents where an animal lives. |
| contains | Sectors | Sector_Box | 1 : 1..n | relationship contains between Sectors and Sector_Box entities – represents which boxes belong to which Sector_Box. |
| works_in | Caregivers | Sectors | 1..n : 1 | relationship works_in between Caregivers and Sectors entities – represents in which sector caregiver works in. |
| has | Animals | Animal_Diet | 1 : 1 | relationship has between Animals and Animal_Diet entities – represents what diet animal has. |
| is_assigned_to | Animal_Diet | Diets_Catalog | 0..n : 1 | relationship is_assigned_to between Animal_Diet and Diets_Catalog entities – represents in which animal diet from diets catalog is.. |
| consists_of | Diets_Catalog | Meals_Catalog | 1 : 1..n | relationship consists_of between Diets_Catalog and Meals_Catalog |

| | | | | |
|---|---|---|---|---|
| | | | | entities – represents which meal is in which diet. |
| describes | Meals_Catalog | Ingredient_Meal | 1 : 1..n | relationship describes between Animals and Animal_Diet entities – represents which ingredients we need to prepare a meal. |
| describes | Ingredient_Meal | Ingredients_Catalog | 1..n : 1 | relationship describes between Ingredient_Meal and Ingredients_Catalog entities – represents which ingredients can be replaced. |
| describes | Warehouse_Ingredient | Units | 0..n : 1 | relationship describes between Warehouse_Ingredient and Units entities – represents in which units the ingredient is stored. |
| is_located | Ingredients_Catalog | Warehouse_Ingredient | 1 : 0..n | relationship is_located between Ingredients_Catalog and Warehouse_Ingredient entities – represents which ingredients are stored in which warehouse. |
| describes | Ingredient_Meal | Units | 0..n : 1 | relationship describes between Ingredient_Meal and Units entities – represents in which units the ingredient is served. |
| consist_of | Real_Meals | Real_Ingredients | 1 : 1..n | relationship consist_of between Real_Meals and Real_Ingredients entities – represents of which ingredients are composed a meal. |
| is_served | Real_Meals | Animal_Diet | 0..n : 1 | relationship is_served between Real_Meals and Animal_Diet entities – represents which meal is served in animal diet. |
| is_chosen | Meals_Catalog | Real_Meals | 1 : 0..n | relationship is_chosen between Meals_Catalog and Real_Meals entities – represents which meal is chosen from meals catalog. |
| gives | Real_Meals | Caregivers | 0..n : 1 | relationship gives between Real_Meals and Caregivers entities – represents which caregiver creates a meal. |
| consist_of | Warehouse_Ingredient | Real_Ingredients | 1 : 0..n | relationship consist_of between Warehouse_Ingredient and Real_Ingredients entities – represents which ingredients are chosen from the warehouse. |

# Relational database schema

Animals(id, species, name, age, country_of_origin, sex, sector_id REF Sector_Box, box_id REF Sector_Box)
Sector_Box(sector_id REF Sectors, box_id , location)
Sectors(id, name)
Caregivers(id, first_name, last_name, sector_id REF Sectors)
Animal_Diet(from, to, description, id, diet_id REF Diets_Catalog, animal_id REF Animals)
Diets_Catalog(id, name)
Meals_Catalog(id, time_of_serving, name, diet_id REF Diets_Catalog)
Ingredient_Meal(meal_id, ingredient_id, portion, ingredient_tag, meal_id REF Meals_Catalog, ingredient_id REF Ingredients_Catalog, unit_id REF Units)
Ingredients_Catalog(id, name)
Units(id, name)
Warehouse_Ingredient(warehouse_name, ingredient_id, date, amount, ingredient_id REF Ingredients_Catalog, unit_id REF Units)
Real_Ingrdients(id, ingredient_id REF Warehouse_Ingredient, warehouse_name REF Warehouse_Ingredient, real_meal_id REF Real_Meals)
Real_Meals(date_time_of_serving, portion, id, caregiver_id REF Caregivers, meal_id REF Meals_Catalog, animal_diet_id REF Animal_Diet)