1.Opis i założenia

Nasz projekt pt. „ZOO” będzie bazą danych dla systemu zarządzania Zoo, które ma ułatwić starszym pracownikom codzienną pracę a nowym wdrożenie do ich obowiązków (np. lokalizacje zwierząt, ich godziny karmienia itp.) oraz ułatwiająca/umożliwiająca łatwiejsze „wyśledzenie” winowajcy jakiegoś błędu z fatalnymi skutkami. Główne założenia:

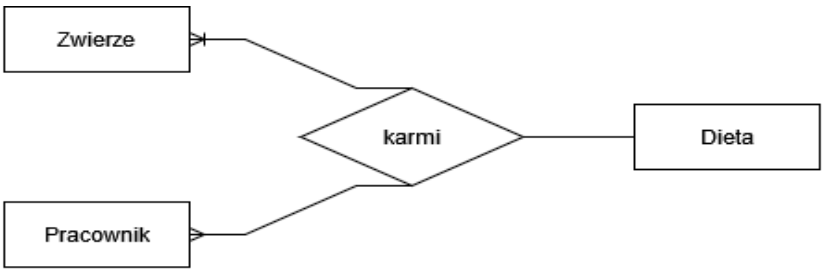
- zwierzę może być karmione przez kilku pracowników o różnych porach dnia
- każde zwierzę ma przypisanego opiekuna
- każde zwierze ma przypisane imię, lokalizacje, gatunek, wiek oraz stan jego zdrowia (w celu ułatwienia w np. izolowaniu lub hospitalizacji chorych zwierząt)

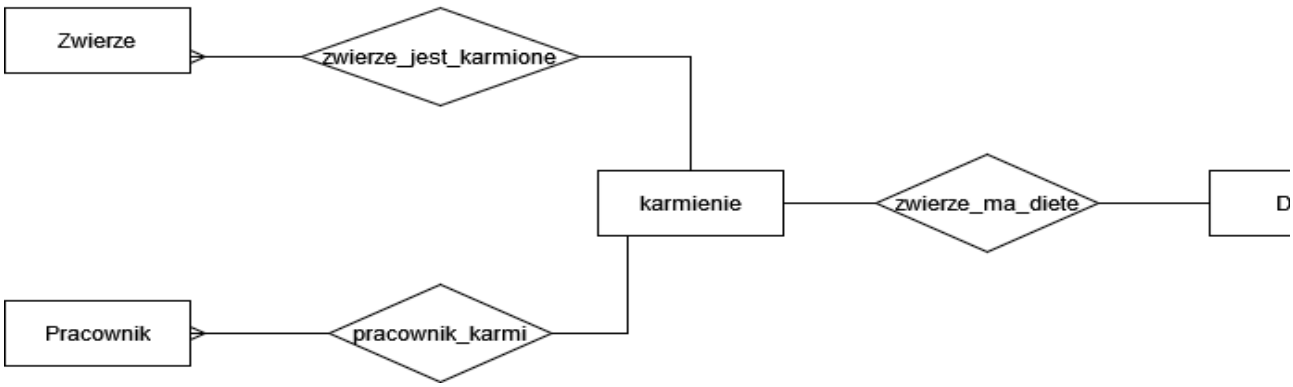
-lokalizacja możliwie szczegółowa a jednocześnie zwięzła (pawilon, sektor i klatka są jednocześnie szczegółowym opisem lokalizacji zwierzęcia jak i na tyle zwięzłym że w naszej bazie nie zajmie za wiele miejsca)

-uwzględniliśmy również dietę zwierząt aby ułatwić pracownikom organizacje pokarmów oraz czasy ich karmienia

-tabela z pracownikiem zawiera jak najmniej informacji gdyż potrzebne są tylko podstawowe które będą informować o tym kto czym się zajmuje/zajmował i w jakim czasie (co ułatwi np. dochodzenie w wypadku jakiejś pomyłki z fatalnymi skutkami)

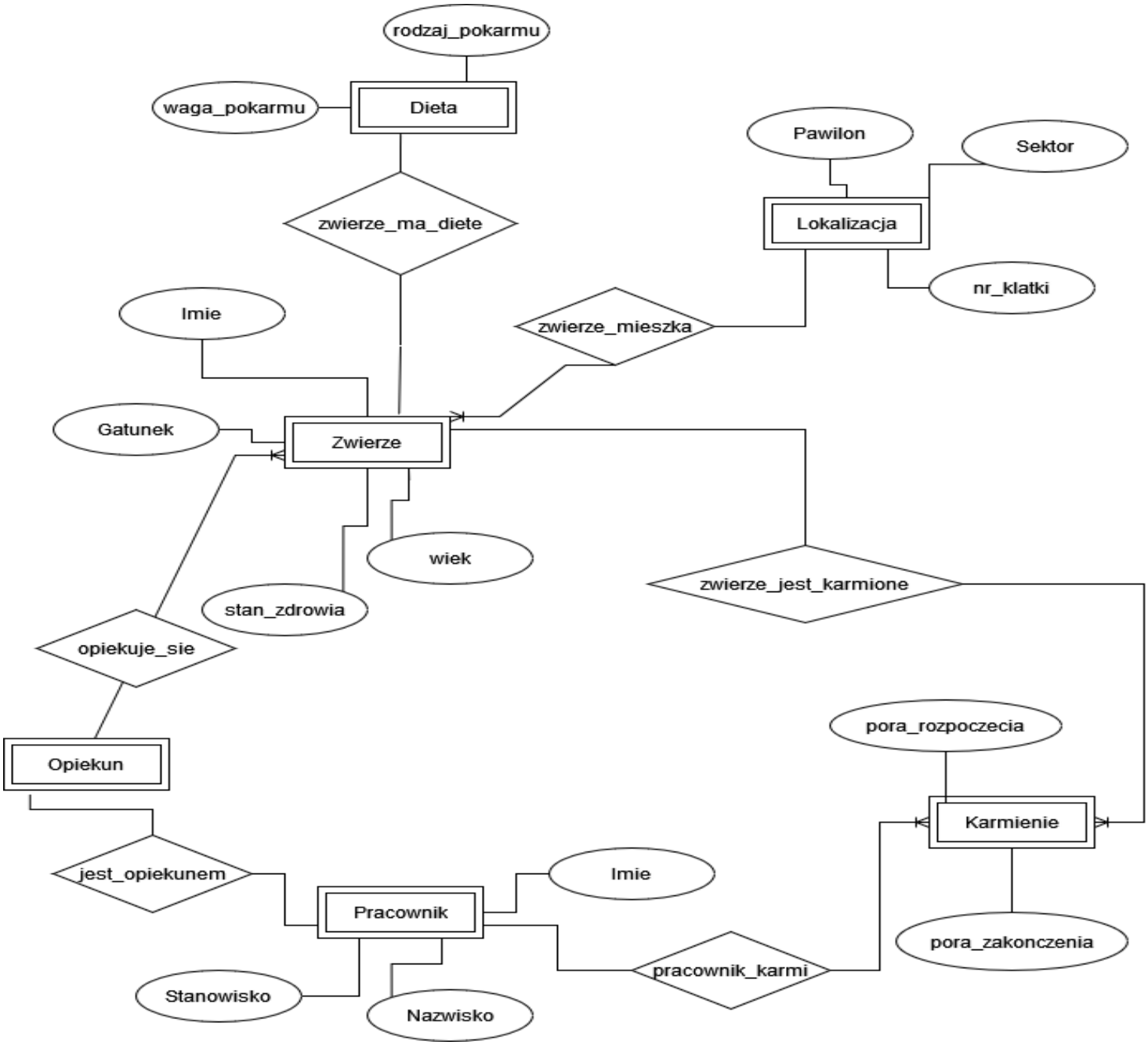
-podsumowując: baza w miarę szczegółowa aby lepiej zorganizować obowiązki pracowników oraz ułatwić ewentualne „śledztwo” ale na tyle prosta aby nie mącić niepotrzebnymi informacjami, które by tylko utrudniały prace.



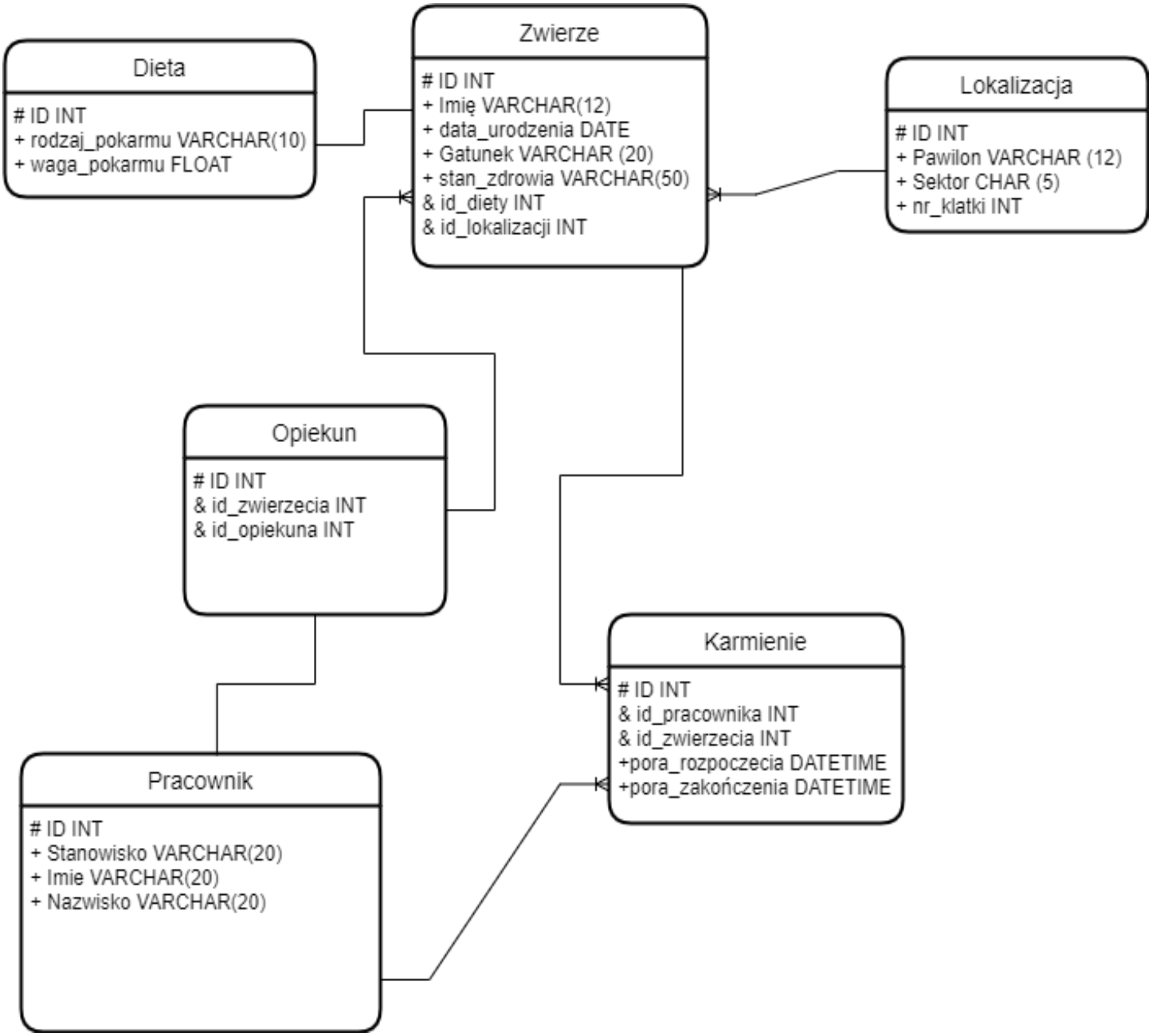


Rys.1 Uproszczenie związku tenarnego Karmi

2.Diagram ERD



3.Model fizyczny



4.Triggery

```
delimiter ;;
CREATE TRIGGER t_pracownik_del AFTER DELETE ON pracownik
        FOR EACH ROW BEGIN
                INSERT INTO log values (now(),'usunieto wiersz z tabeli pracownik');
        END;
;;
delimiter ;
```

```
delimiter ;;
CREATE TRIGGER t_zwierze_ins AFTER INSERT INTO zwierze
        FOR EACH ROW BEGIN
                INSERT INTO log values (now(),'dodano wiersz do tabeli zwierze');
        END;
;;
delimiter ;
```

```
delimiter ;;
DROP TRIGGER if exists t_zwierze_upd;
CREATE TRIGGER t_zwierze_upd AFTER UPDATE ON zwierze
        FOR EACH ROW BEGIN
                INSERT INTO log values (now(),concat('zmieniono wiersz w tabeli
zwierze stan_zdrowia z:',OLD.stan_zdrowia,' na :', NEW.stan_zdrowia));
        END;
;;
delimiter;

update zwierze set stan_zdrowia='chory' where id=12;
select * from log;
```

5.Store procedure

```
CREATE PROCEDURE `pokaz_zwierzeta`(x date,out count_mlodsze int,out
count_starsze int)
BEGIN
select count(*) from zwierze where data_urodzenia>=x into count_mlodsze;
select count(*) from zwierze where data_prod<x into count_starsze;
```

```
END;


CREATE PROCEDURE `pokaz_pracownika`(x char)
BEGIN
select * from pracownik where imie=x;
END;


CREATE PROCEDURE `pokaz_stan_zwierzecia`(x char)
BEGIN
select stan_zdrowia from zwierze where imie=x;
END;
```

6.Zapytania walidujące

```
// dieta niczyja
select d.id from dieta d
where d.id not in (
select z.id_diety from zwierze z
where z.id_diety is not null
);

// zwierze bez diety
select z.id from zwierze z
where z.id_diety not in (
select d.id from dieta d
);

// zwierze bez opiekuna
select z.id from zwierze z
where z.id not in (
select o.id_zwierzecia from opiekun o
);

// zwierzę bez poprawnej lokalizacji
select z.id from zwierze z
where z.id_lokalizacji not in (
select l.id from lokalizacja l
);

// błędna pora karmienia
select * from karmienie k
where k.pora_rozpoczecia > k.pora_zakonczenia;
```

```
// karmione nieistniejące zwierzę
select k.id from karmienie k
where k.id_zwierzecia not in (
select z.id from zwierze z
);
```

7.Agregaty

```
// sumaryczna masa paszy
select sum(x.waga_pokarmu) from (
        select d.waga_pokarmu, z.id
        from dieta d join zwierze z
        on d.id=z.id_diety
        where d.rodzaj_pokarmu='pasza'
) x;

// ilość opiekunów
select count(distinct id_pracownika) from opiekun;

// najstarsze zwierze
select * from zwierze where data_urodzenia in (
        select min(data_urodzenia) from zwierze
);

// ilość klatek w pawilonach
select pawilon, count(*)
from lokalizacja group by pawilon;

// przeciętny czas trwania karmienia
select avg(time_to_sec(timediff(pora_zakonczenia, pora_rozpoczecia)))/60
sredni_czas_w_min
from karmienie;
```

8.Podwójny join

```
// opiekuni + lokalizacje zwierząt, którymi się opiekują
select p.imie, p.nazwisko, l.pawilon, l.sektor, l.nr_klatki
from pracownik p join opiekun o on o.id_pracownika=p.id
join zwierze z on z.id=o.id_zwierzecia
join lokalizacja l on l.id=z.id_lokalizacji;
```

Twórcy: Jakub Styn (s22449) i Kasia Popieniuk (s22048)