Bada - schronisko

Maciej Lipski

Kacper Średnicki

30 listopada 2023

Spis treści

1. Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)

W ramach projektu przygotowano projekt bazy danych schroniska dla zwierząt. Baza danych ma na celu skuteczne zarządzanie procesami związanymi z przyjmowaniem, opieką, a także adopcją zwierząt. Schronisko to oferuje tymczasowe schronienie, opiekę weterynaryjną i generalną opiekę pracownika, oraz umożliwia adopcję zwierząt potrzebujących nowego domu. Przyjmuje tylko koty i psy. Schronisko składa się z wielu placówek. Wszystkie placówki posiadają boksy, w których mieszkają przyjmowane do schroniska zwierzęta. Każda placówka posiada swoje wyposażenie, w którego skład wchodzi też karma. Placówki zapewniają też podawanie zwierzętom odpowiednich leków. Organizacja zatrudnia pracowników na stanowiskach indywidualnego opiekuna zwierząt, opiekuna medycznego, kierownika, pracownika biurowego i pracownika utrzymującego czystość. Niektórzy pracownicy zatrudniani są bezpośrednio w placówce, a część pracuje na rzecz wszystkich placówek (np. niektórzy pracownicy biurowi). Część pracowników wykonuje pracę jako wolontariat a pozostała odpłatnie.

2. Definicja systemu

Zdefiniowano następujące transakcje będące wykonywane na bazie danych:
- W obszarze **obsługi placówki**:
  - modyfikacja informacji o godzinach otwarcia, maksymalnej ilości zwierząt w placówce,
  - rozbudowa schroniska o dodatkowe boksy.
- W obszarze **obsługi zwierząt**:
  - przyjęcie zwierzęcia do schroniska - dodanie do bazy danych,
  - adopcja zwierzęcia,
  - przypisanie zwierzęciu aktualnie zamieszkiwanego boksu,
  - przypisanie zwierzęciu informacji o przyjmowanych lekach i spożywanej karmie,
  - dodawanie i modyfikacja informacji o aktualnych opiekunach zwierzęcia.
- W obszarze **obsługi wyposażenia**:
  - wprowadzanie informacji o nowym wyposażeniu i ustalanie jego wykorzystania przez boks,
  - ewidencja lekarstw, karm oraz ich wykorzystania, wprowadzanie informacji o nowych lekach, karmach.
- W obszarze **obsługi pracowników**:
  - zatrudnianie nowych pracowników,
  - dodawanie informacji o płacach, wprowadzanie podwyżek,
  - zatrudnianie pracowników w ramach wolontariatu,
  - wprowadzanie informacji o funkcjach pełnionych przez pracownika.

2.1. Perspektywy użytkowników

Zidentyfikowane perspektywy użytkowników:
- **Dyrektor organizacji** - ma dostęp do wszystkich danych w bazie danych.
- **Kierownik placówki** - ma dostęp do wszystkich danych dotyczących danej placówki.
- **Pracownik organizacji** - ma dostęp do danych o zwierzętach, wyposażeniu, lekach i karmach. W zależności od stanowiska może wprowadzać i modyfikować te dane.
- **Potencjalny adoptujący** - ma dostęp do podstawowych danych dotyczących placówki (adres, godziny otwarcia, itp.) i danych dotyczących zwierząt adoptowanych przez siebie oraz zwierząt jeszcze nieadoptowanych.

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

Na początku projektowania bazy danych ustalono konieczność reprezentowania w bazie następujących encji:
- **placówka** - encja reprezentująca pojedynczą placówkę schroniska
- **boks** - encja reprezentująca boks w schronisku zamieszkiwany przez zwierzęta
- **zwierzę** - encja reprezentująca zwierzę mieszkające w schronisku (lub takie, które już je opuściło), zarówno psa jak i kota
- **adoptujący** - encja reprezentująca osobę adoptującą zwierzę/zwierzęta ze schroniska
- **karma** - encja reprezentująca karmę na stanie schroniska
- **wyposażenie** - encja reprezentująca pojedynczy element wyposażenia schroniska np. miskę, smycz
- **lekarstwo** - encja reprezentująca lekarstwo na stanie schroniska
- **pracownik** - encja reprezentująca pracownika schroniska, niezależnie od stanowiska oraz wykonywania pracy jako wolontariat.

3.2. Ustalenie związków między encjami i ich typów

Na podstawie założeń funkcjonalnych, dokonano ustaleń dotyczących związków między encjami oraz ich typów. Związki opisano w tablach 1 i 2.

Nazwa związku	Encja 1.	Encja 2.	Krotność związku (encja1 - encja2)	Opis związku
Adoptuje	Adoptujący	Zwierzę	0..1 - 1..n	Związek reprezentujący adoptowanie zwierzęcia. Adoptujący zawsze adoptuje co najmniej jedno zwierzę, ale może wiele. Zwierzę może być jeszcze nieadoptowane, ale może wrócić do schroniska i być adoptowane przez innego adoptującego lub od razu przez dwóch adoptujących np. małżeństwo.
Je	Zwierzę	Karma	0..n - 0..m	Związek reprezentujący informację o karmie jaką je zwierzę. Zwierzę nie musi mieć przypisanej karmy ale może być karmione wieloma karmami. Karmy mogą być nieużywane przez żadne zwierzę lub przez wiele zwierząt.
Jest_ zamieszkiwany	Boks	Zwierzę	1..n - 0..m	Związek wiążący zwierzę z zamieszkiwanym przez nie boksem. Boks może być pusty lub zamieszkiwany przez wiele zwierząt. Zwierzę w momencie przyjęcia do schroniska zawsze ma przypisany boks. Może zmieniać boksy więc istnieje potrzeba możliwości przypisania zwierzęciu wielu boksów i wprowadzenia związku pośredniego reprezentującego zamieszkiwanie na dalszym etapie.
Ma	Placówka	Karma	0..n - 0..m	Związek reprezentujący karmy posiadane w placówkach. Placówka może nie mieć przypisanych żadnych karm lub mieć ich wiele. Karmy mogą istnieć w wielu placówkach lub w żadnej (mogą jedynie być zarejestrowane w bazie danych)
Ma_na_stanie	Placówka	Wyposażenie	1..1 - 0..m	Związek reprezentujący informację o posiadanym przez placówkę wyposażeniu. Jako, że wyposażenie w projektowanej bazie to pojedynczy obiekt ma on zawsze przypisaną jedną placówkę. Placówka może nie mieć żadnego wyposażenia lub mieć ich wiele.
Opiekuje_się	Pracownik	Zwierzę	1..n - 0..m	Związek reprezentujący opiekę pracownika nad zwierzęciem. Pracownik może opiekować się wieloma zwierzętami lub żadnym (np. pracownik biurowy). Zwierzę musi posiadać min. jednego opiekuna. Może posiadać także wielu opiekunów (np. dodatkowego opiekuna medycznego).
Posiada	Placówka	Boks	1..1 - 1..m	Związek wiąże boks z placówką, w której się znajduje. Każda placówka musi mieć co najmniej jeden boks, ale może także posiadać ich wiele. Z drugiej strony, każdy boks musi być przypisany do konkretnej placówki, co zapewnia jednoznaczne określenie właściciela.

Tabela 1. Tabela zawierająca związki w modelu konceptualnym

Nazwa związku	Encja 1.	Encja 2.	Krotność związku (encja1 - encja2)	Opis związku
Posiada_na_stanie	Placówka	Lekarstwo	0..n - 0..m	Związek umożliwia placówce zarządzanie lekarstwami. Placówka może mieć od żadnego do wielu lekarstw na stanie. Równocześnie jedno lekarstwo może być na stanie w wielu placówkach, ale by istnieć w bazie danych nie musi być w żadnej placówce.
Przyjmuje	Zwierzę	Lekarstwo	0..n - 0..m	Związek reprezentuje przyjmowanie przez zwierzę lekarstw. Zwierzę może przyjmować od zera do wielu lekarstw. Równocześnie jedno lekarstwo może być przyjmowane przez wiele zwierząt, ale może być nie przyjmowane przez żadne zwierzę.
Zatrudnia	Placówka	Pracownik	0..n - 0..m	Związek reprezentuje związek placówki i pracowników. Placówka może zatrudniać od żadnego do wielu pracowników. Równocześnie jeden pracownik może być zatrudniony w wielu placówkach. Może być także pracownikiem całej organizacji, wtedy nie jest zatrudniony w żadnej placówce.

Tabela 2. Tabela zawierająca związki w modelu konceptualnym - cd.

3.3. Określenie atrybutów i ich dziedzin

Dla wszystkich encji ustalono również zestawy ich atrybutów. Każdy atrybut posiada określoną dziedzinę, a także informację czy jest atrybutem obowiązkowym oraz czy pełni rolę klucza głównego. Atrybuty poszczególnych encji zostały przedstawione w tabelach poniżej.

Encja Placowka

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiąz-kowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_placowki	Unikatowy numer placówki	SmallInt	T	T	T	
Adres	Adres placówki	VarChar(200)	T	N	N	Atrybut segmentowy, wielowarto-ściowy, obejmujący: miasto, ulicę, numer lokalizacji, kod pocztowy, pocztę
Nazwa_placowki	Nazwa placówki	VarChar(30)	T	N	T	
Nr_telefonu	Numer telefonu do placówki	VarChar(12)	T	N	T	Atrybut wielo-wartościowy
Adres_email	Adres email placówki	VarChar(20)	N	N	T	Atrybut wielo-wartościowy zgodny z maską '....@....'
Maks_liczba_zwierzat	Maksymalna liczba zwierząt w placówce	Integer	T	N	T	
Godziny_otwarcia	Godziny otwarcia placówki	VarChar(200)	T	N	N	Atrybut segmentowy, wielowarto-ściowy, składający się z dni tygodnia i godzin

Tabela 3. Tabela opisująca atrybuty encji Placowka

Encja Boks

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiąz-kowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_boksu	Unikatowy numer boksu	Integer	T	T	T	
Maks_liczba_zwierzat	Maksymalna liczba zwierząt w boksie	Integer	T	N	T	
Wymiary	Wymiary boksu	VarChar(20)	T	N	N	Atrybut segmentowy

Tabela 4. Tabela opisująca atrybuty encji Boks

Encja Zwierze

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiąz-kowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_zwierzecia	Unikatowy numer zwierzęcia	Integer	T	T	T	
Gatunek	Gatunek zwierzęcia	'pies', 'kot'	T	N	T	
Rasa	Rasa zwierzęcia	VarChar(30)	N	N	T	
Wiek	Wiek zwierzęcia	SmallInt	N	N	T	
Masa	Masa zwierzęcia	SmallInt	T	N	T	Wyrażona w kilogramach
Data_przyjecia	Data przyjęcia zwierzęcia do schroniska	Date	T	N	T	
Data_opuszczenia	Data opuszczenia schroniska	Date	N	N	T	
Czy_szczepiony	Flaga określająca czy zwierzę jest szczepione	Boolean	T	N	T	
Czy_kastrowany	Flaga określająca czy zwierzę jest kastrowane	Boolean	T	N	T	
Plec	Płeć zwierzęcia	'Samiec','Samica'	T	N	T	
Opis	Opis zwierzęcia	VarChar(100)	N	N	N	Opis charakteru, wyglądu, historii zwierzęcia

Tabela 5. Tabela opisująca atrybuty encji Zwierze

Encja Adoptujacy

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_adoptujacego	Unikatowy numer adoptującego	Integer	T	T	T	
Imie	Imię adoptującego	VarChar(20)	T	N	T	
Nazwisko	Nazwisko adoptującego	VarChar(30)	T	N	T	
Nr_dokumentu	Numer dokumentu adoptującego	VarChar(10)	T	N	T	
Pesel	Numer pesel adoptującego	Character(11)	N	N	T	
Nr_telefonu	Numer telefonu adoptującego	VarChar(12)	T	N	T	Atrybut wielowartościowy
Adres_email	Adres email adoptującego	VarChar(20)	N	N	T	Atrybut wielowartościowy zgodny z maską '....@....'
Adres	Adres zamieszkania adoptującego	VarChar(200)	T	N	N	Atrybut segmentowy, wielowartościowy, obejmujący: miasto, ulicę, numer lokalizacji, kod pocztowy, pocztę
Plec	Płeć adoptującego	'M','K'	T	N	T	

Tabela 6. Tabela opisująca atrybuty encji Adoptujacy

Encja Karma

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_karmy	Unikatowy numer karmy	Integer	T	T	T	
Nazwa	Nazwa karmy	VarChar(20)	T	N	T	
Producent	Producent karmy	VarChar(20)	T	N	T	
Ilosc_na_stanie	Ilość karmy na stanie placówki	Integer	T	N	T	Ilość w kilogramach
Data_waznosci	Data ważności karmy	Date	T	N	T	

Tabela 7. Tabela opisująca atrybuty encji Karma

Encja Wyposazenie

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiąz-kowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_wyposazenia	Unikatowy numer wyposażenia	Integer	T	T	T	
Typ	Typ wyposażenia	VarChar(20)	T	N	T	

Tabela 8. Tabela opisująca atrybuty encji Wyposazenie

Encja Lekarstwo

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiąz-kowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_lekarstwa	Unikatowy numer lekarstwa	Integer	T	T	T	
Ilosc_na_stanie	Ilość lekarstwa na stanie	Integer	T	N	T	Ilość w sztukach
Nazwa	Nazwa lekarstwa	VarChar(20)	T	N	T	
Producent	Producent lekarstwa	VarChar(20)	T	N	T	
Dawka	Dawka lekarstwa	VarChar(10)	T	N	T	
Przeciwskazania	Przeciwskazania do podania lekarstwa	VarChar(200)	N	N	N	Atrybut wielo-wartościowy

Tabela 9. Tabela opisująca atrybuty encji Lekarstwo

Encja Pracownik

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_pracownika	Unikatowy numer pracownika	Integer	T	T	T	
Imie	Imię pracownika	VarChar(20)	T	N	T	
Nazwisko	Nazwisko pracownika	VarChar(30)	T	N	T	
Nr_dokumentu	Numer dokumentu pracownika	VarChar(10)	T	N	N	
Pesel	Numer pesel pracownika	Character(11)	N	N	T	
Nr_telefonu	Numer telefonu pracownika	VarChar(12)	T	N	N	Atrybut wielowartościowy
Adres_email	Adres email pracownika	VarChar(20)	N	N	N	Atrybut wielowartościowy zgodny z maską '....@....'
Adres	Adres zamieszkania pracownika	VarChar(200)	T	N	N	Atrybut segmentowy, wielowartościowy, obejmujący: miasto, ulicę, numer lokalizacji, kod pocztowy, pocztę
Plec	Płeć pracownika	'M','K'	T	N	T	
Pensja	Pensja pracownika	Money	N	N	T	Wartość w złotych
Stanowisko	Stanowisko pracownika	'Opiekun_zwierzat', 'Pracownik_medyczny', 'Kierownik', 'Pracownik_biurowy', 'Sprzatajacy'	T	N	T	Atrybut wielowartościowy
Czy_wolontariusz	Flaga określająca czy pracownik jest wolontariuszem	Boolean	T	N	T	

Tabela 10. Tabela opisująca atrybuty encji Pracownik

3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)

W ramach projektowania konceptualnego zdecydowano się na zastosowanie kilku reguł integralnościowych, w celu ścisłego zdefiniowania dziedzin niektórych atrybutów. Przedstawiono je w tabeli 11.

Domena	Dziedzina	Zastosowanie w encja/atrybut	Uwagi
Gatunek	('Pies','Kot')	Zwierze/Gatunek	Zgodnie z początkowymi założeniami projektowymi, schronisko przyjmuje wyłącznie psy i koty.
PlecOsoby	('M','K')	Adoptujacy/Plec, Pracownik/Plec	Adoptujący i pracownik mogą być wyłącznie mężczyzną (płeć 'M') lub kobietą (płeć 'K').
PlecZwierzecia	('Samiec','Samica')	Zwierze/Plec	Zwierzę może być wyłącznie samcem lub samicą.
Stanowisko	('Opiekun_zwierzat', 'Pracownik_medyczny', 'Kierownik', 'Pracownik_biurowy', 'Sprzatajacy')	Pracownik/Stanowisko	Zgodnie z początkowymi założeniami projektowymi, schronisko zatrudnia pracowników wyłącznie na stanowiska takie jak: opiekun zwierząt, pracownik medyczny, kierownik, pracownik biurowy, sprzątający.

Tabela 11. Tabela zawierająca zastosowane reguły integralnościowe

3.5. Klucze kandydujące i główne (decyzje projektowe)

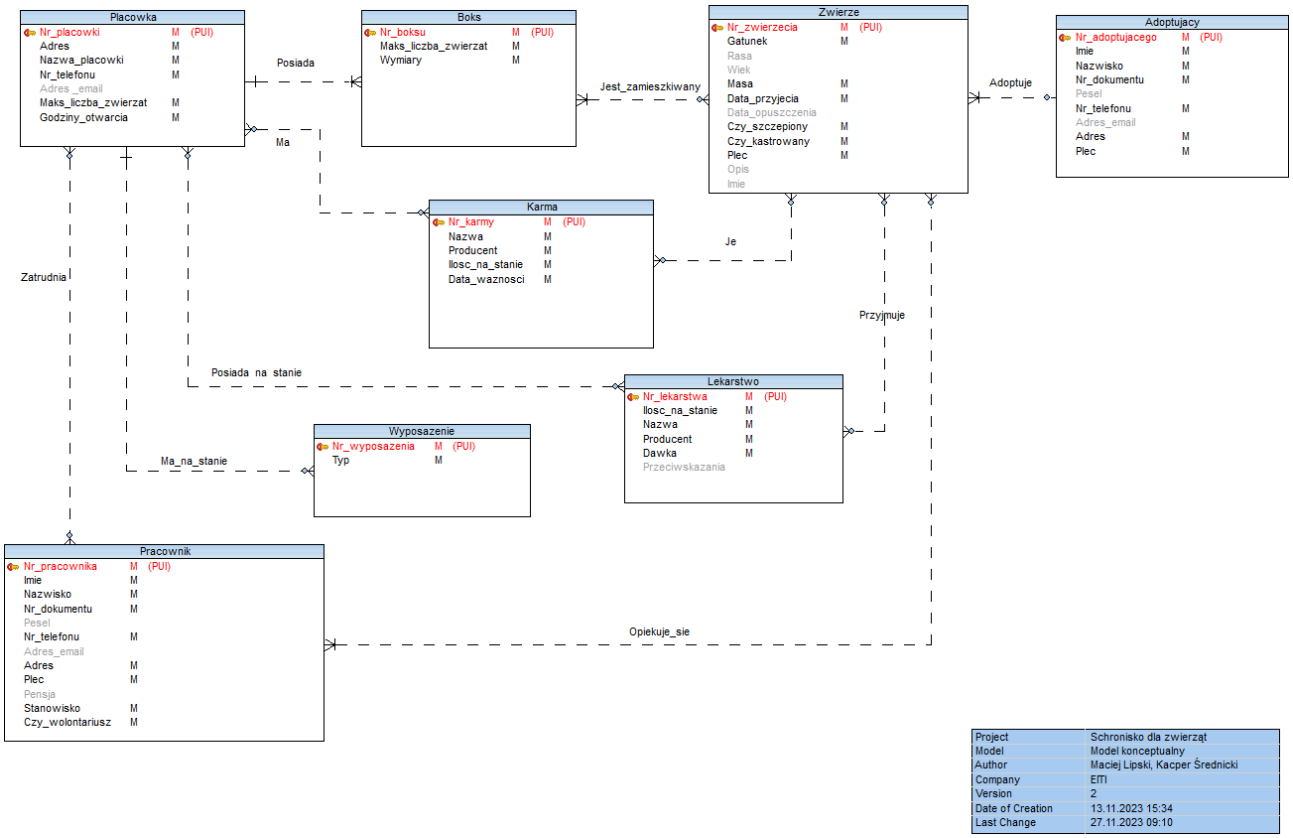
Poza wyborem kluczy głównych, dla każdej encji przeanalizowano także istnienie kluczy kandydujących. Decyzje projektowe podjęte w tym zakresie podsumowano w tabeli 12.

Encja	Klucz główny	Klucze kandydujące	Uwagi
Placowka	Nr_placowki		
Boks	Nr_boksu		
Zwierze	Nr_zwierzecia		
Adoptujacy	Nr_adoptujacego	Nr_dokumentu, Pesel	Numer dokumentu został odrzucony jako potencjalny klucz główny ze względu na brak jednolitości jego formatu, PESEL natomiast ze względu na fakt, że posiadają go jedynie obywatele Polski.
Karma	Nr_karmy	Klucz złożony: (Nazwa, Producent)	Zastosowanie klucza złożonego jako klucza głównego zostało odrzucone ze względu na jego skomplikowaną formę.
Wyposazenie	Nr_wyposazenia		
Lekarstwo	Nr_lekarstwa	Klucz złożony: (Nazwa, Producent, Dawka)	Zastosowanie klucza złożonego jako klucza głównego zostało odrzucone ze względu na jego skomplikowaną formę.
Pracownik	Nr_pracownika	Nr_dokumentu, Pesel	Numer dokumentu został odrzucony jako potencjalny klucz główny ze względu na brak jednolitości jego formatu, PESEL natomiast ze względu na fakt, że posiadają go jedynie obywatele Polski.

Tabela 12. Tabela zawierająca klucze główne oraz kandydujące

3.6. Schemat ER na poziomie konceptualnym

Po dokonaniu powyższych ustaleń, przystąpiono do przeniesienia modelu do narzędzia Toad Data Modeler. Wykonano w nim schemat ER na poziomie konceptualnym (nazywany logical model w narzędziu). Model załączono na rysunku 1.

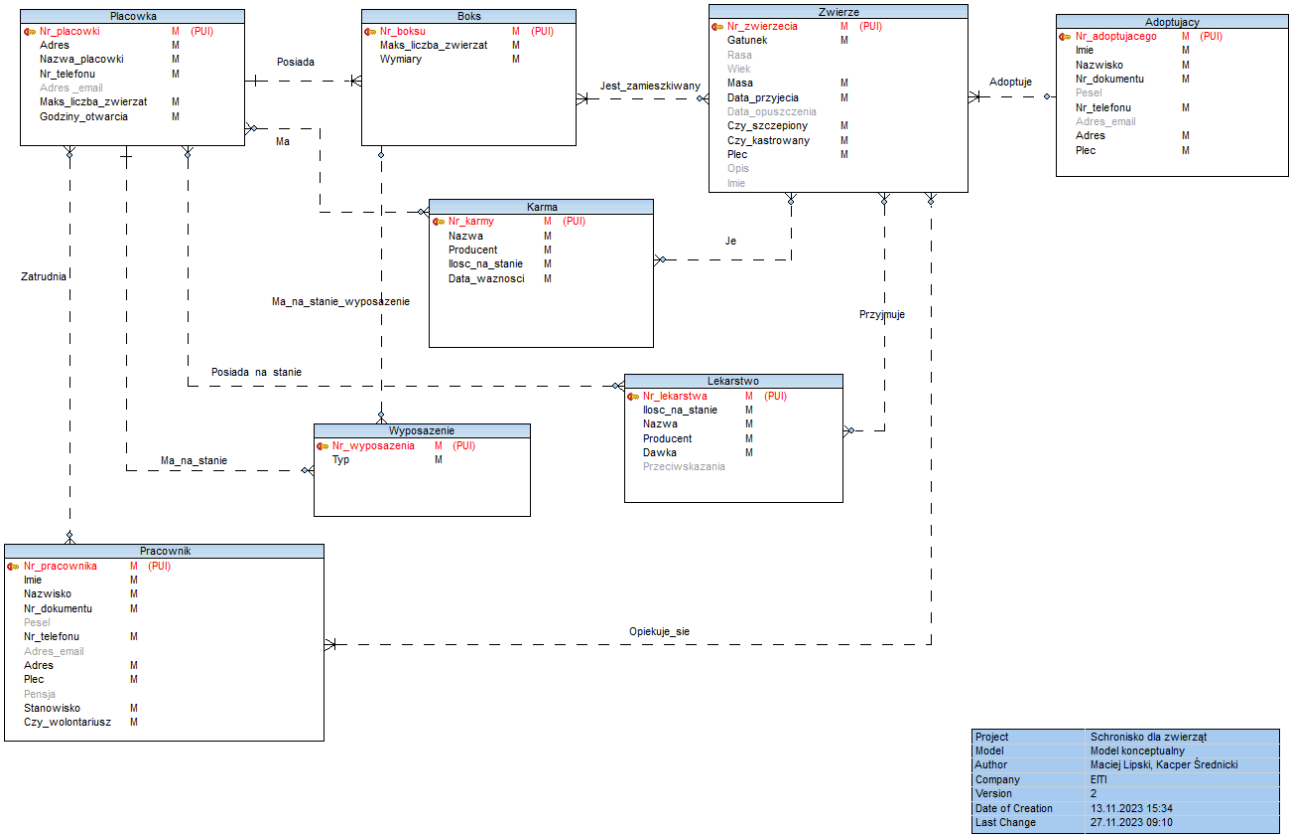


Rysunek 1. Schemat ER bazy danych schroniska dla zwierząt na etapie konceptualnym - wersja 1.

3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

W trakcie analizy przygotowanego modelu znaleziono pułapkę wachlarzową. W pierwszej wersji modelu (rys. 1) nie istniała możliwość jednoznacznej identyfikacji do którego boksu należy wyposażenie. Aby do rozwiązać, rozważono zmianę związku Ma_na_stanie (placówka - wyposażenie) na związek boks - wyposażenie. Zauważono, że taka zmiana spowodowałaby pojawienie się pułapki szczelinowej. Przestałaby wtedy istnieć możliwość reprezentacji wyposażenia, które jest na stanie placówki, ale nie przynależy do żadnego boksu np. urządzeń elektronicznych. W związku z tym zdecydowano się na dodanie związku Ma_na_stanie_wyposazenie między boksem a wyposażeniem (0..1 - 0..n), oraz pozostawienie bezpośredniego związku między wyposażeniem a placówką. Poprawiony schemat zamieszczono na rysunku 2.

Potencjalna pułapka szczelinowa istniałaby między zwierzęciem a placówką. Nie istnieje bezpośredni związek między tymi encjami. Ze względu na to, że zwierzęciu już na przyjęciu przypisuje się boks, jesteśmy w stanie jednoznacznie określić w jakiej placówce jest przyjęte zwierzę. W związku z tym można wykluczyć istnienie pułapki szczelinowej w tym wypadku.



Rysunek 2. Schemat ER bazy danych schroniska dla zwierząt na etapie konceptualnym w wersji po usunięciu pułapek szczelinowych i wachlarzowych i poprawie pozostałych błędów.

4. Model logiczny

4.1. Charakterystyka modelu relacyjnego

Po ukończeniu modelu konceptualnego przystąpiono do jego konwersji na model logiczny. Każdą encję zmieniono na relację. W celu łatwiejszego odróżnienia encji od relacji zmieniono nazwę encji na liczbę mnogą. Dokonano przekształcenia niekompatybilnych z modelem logicznym związków wiele do wielu. W tym celu stworzono odpowiednie tabele łączące (sekcja 4.2). Jeśli uczestnictwo w związku było opcjonalne, dopuszczono wartość NULL klucza obcego w relacji reprezentującej dawną encję. W przeciwnym wypadku klucz obcy stał się polem MANDATORY.

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

W celu usunięcia związków wiele do wielu, dokonano wprowadzenia tabel łączących. Dawne związki wiele do wielu przekształcono na związki jeden do wielu między dawnymi encjami a tabelą łączącą. Poniżej opisano wprowadzone relacje łączące.

Zamieszkiwania
Relacja opisująca zamieszkiwanie przez zwierzę boksu w danym czasie. Powstała w wyniku likwidacji związku wiele do wielu Jest_zamieszkiwany (Boks - zwierzę, 1..n - 0..n) Posiada atrybuty opisane w tabeli 13.

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_zamieszkiwania	Unikatowy numer zamieszkiwania	Integer	T	Klucz główny
Nr_boksu	Identyfikator boksu będącego zamieszkiwanym	Integer	T	Klucz obcy
Nr_zwierzęcia	Identyfikator zwierzęcia zamieszkującego w danym momencie boks	Integer	T	Klucz obcy
Data_rozpoczecia	Data rozpoczęcia zamieszkiwania boksu przez zwierzę	Date	T	
Data_zakonczenia	Data zakończenia zamieszkiwania boksu przez zwierzę	Date	N	

Tabela 13. Atrybuty relacji zamieszkiwania

Karmienia
Relacja opisująca podawanie zwierzęciu karmy. Powstała w wyniku likwidacji związku wiele do wielu Je (zwierzę - karma, 0..n - 0..n). Posiada atrybuty opisane w tabeli 14.

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_karmienia	Unikatowy numer karmienia	Integer	T	Klucz główny
Nr_karmy	Identyfikator używanej karmy	Integer	T	Klucz obcy
Nr_zwierzęcia	Identyfikator karmionego zwierzęcia	Integer	T	Klucz obcy
Data_rozpoczecia	Data rozpoczęcia danego karmienia	Date	T	
Data_zakonczenia	Data zakończenia danego karmienia	Date	N	

Tabela 14. Atrybuty relacji karmienia

Posiadania_karm
Relacja opisująca posiadanie przez placówkę karmy na stanie magazynowym. Powstała w wyniku likwidacji związku wiele do wielu Ma (placówka - karma, 0..n - 0..n). Posiada atrybuty opisane w tabeli 15.

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_posiadania _karmy	Unikatowy numer posiadania karmy	Integer	T	Klucz główny
Nr_karmy	Identyfikator posiadanej karmy	Integer	T	Klucz obcy
Nr_placowki	Identyfikator placówki posiadającej karmę	Integer	T	Klucz obcy
Ilosc_na _stanie	Ilość posiadanej karmy w kg, przeniesiony z tabeli karmy	Integer	T	

Tabela 15. Atrybuty relacji posiadania karm

Posiadania_lekarstw
Relacja opisująca posiadanie przez placówkę lekarstw na stanie magazynowym. Powstała w wyniku likwidacji związku wiele do wielu Posiada_na_stanie (placówka - lekarstwo, 0..n - 0..n). Posiada atrybuty opisane w tabeli 16.

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_posiadania _lekarstw	Unikatowy numer posiadania lekarstwa na stanie	Integer	T	Klucz główny
Nr_lekarstwa	Identyfikator posiadanego lekarstwa	Integer	T	Klucz obcy
Nr_placowki	Identyfikator placówki posiadającej lekarstwo	Integer	T	Klucz obcy
Ilosc_na _stanie	Ilość posiadanej karmy w kg, przeniesiony z tabeli lekarstwa	Integer	T	

Tabela 16. Atrybuty relacji posiadania lekarstw

Przyjmowania_lekarstw
Relacja opisująca przyjmowanie lekarstwa przez zwierzę. Powstała w wyniku likwidacji związku wiele do wielu Przyjmuje (zwierzę - lekarstwo, 0..n - 0..n). Posiada atrybuty opisane w tabeli 17.

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_przyjmowania _lekarstw	Unikatowy numer przyjmowania lekarstwa przez zwierzę	Integer	T	Klucz główny
Nr_lekarstwa	Identyfikator przyjmowanego lekarstwa	Integer	T	Klucz obcy
Nr_zwierzęcia	Identyfikator zwierzęcia przyjmującego lekarstwo	Integer	T	Klucz obcy
Poczatek _przyjmowania	Data rozpoczęcia przyjmowania lekarstwa	Date	T	
Koniec _przyjmowania	Data zakończenia przyjmowania lekarstwa	Date	N	

Tabela 17. Atrybuty relacji przyjmowania lekarstw

Przypisania pracowników
Relacja umożliwiająca przypisanie pracownika do placówki. Powstała w wyniku likwidacji związku wiele do wielu Zatrudnia (placówka- pracownik, 0..n - 0..n). Posiada atrybuty opisane w tabeli 18. Pracownik nieuczestniczący w związku z tą relacją jest zatrudniony na rzecz całej organizacji.

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_przypisania	Unikatowy numer przypisania pracownika do placówki	Integer	T	Klucz główny
Nr_pracownika	Identyfikator przypisywanego pracownika	Integer	T	Klucz obcy
Nr_placowki	Identyfikator placówki do której przypisany jest pracownik	Integer	T	Klucz obcy
Poczatek _przypisania	Data rozpoczęcia przypisania pracownika do placówki	Date	T	
Koniec _przypisania	Data zakończenia przypisania pracownika do placówki	Date	N	

Tabela 18. Atrybuty relacji przypisania pracowników

Opiekowania
Relacja opisująca opiekowanie się zwierzęciem przez pracownika. Powstała w wyniku likwidacji związku wiele do wielu Opiekuje_sie (pracownik- zwierzę, 1..n - 0..n). Posiada atrybuty opisane w tabeli 19.

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_opiekowania	Unikatowy numer opiekowania się zwierzęciem przez pracownika	Integer	T	Klucz główny
Nr_pracownika	Identyfikator opiekującego się zwierzęciem pracownika	Integer	T	Klucz obcy
Nr_zwierzecia	Identyfikator zwierzęcia będącego pod opieką pracownika	Integer	T	Klucz obcy
Poczatek _opiekowania	Data rozpoczęcia opiekowania się zwierzęciem przez pracownika	Date	T	
Koniec _opiekowania	Data zakończenia opiekowania się zwierzęciem przez pracownika	Date	N	

Tabela 19. Atrybuty relacji opiekowania

4.3. Proces normalizacji – analiza i przykłady

Po usunięciu właściwości niekompatybilnych z modelem relacyjnym przystąpiono do wykonania procesu normalizacji. Przeanalizowano, które z relacji należało poddać temu procesowi, aby doprowadzić wszystkie relacje w bazie danych do (wymaganej w ramach projektu) trzeciej postaci normalnej.

Pierwsza postać normalna zakłada, że wartość każdego z atrybutów relacji jest wartością atomową, a także nie występują w niej powtarzające się grupy. W celu doprowadzenia relacji do tej postaci, dokonano szeregu modyfikacji na poziomie modelu logicznego.

W pierwszej kolejności zwrócono uwagę na trybut segmentowy Adres występujący w relacjach: Adoptujący, Placówki oraz Pracownicy. Zdecydowano się na zastąpienie tego atrybutu osobną relacją Adresy, posiadającą atrybuty odpowiadające informacjom o adresie lokalizacji. Po usunięciu atrybutu Adres z trzech relacji wymienionych powyżej oraz połączeniu ich związkami (1:n) z nową relacją Adresy, wszystkie trzy relacje otrzymały dodatkowy atrybut Nr_adresu, pełniący funkcję klucza obcego. Na podobnej zasadzie stworzono także osobną relację Poczty, wchodzącą w związek (1:n) z relacją Adresy, w celu uniknięcia powtarzających się grup danych. Atrybuty relacji Adresy i Poczty zostały przedstawione w tabelach 20 i 21.
Adresy

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_adresu	Unikatowy numer adresu	Integer	T	Klucz główny
Nr_poczty	Identyfikator poczty	Integer	T	Klucz obcy
Ulica	Ulica	VarChar2(30)	T	
Nr_budynku	Numer budynku	VarChar2(5)	T	
Nr_mieszkania	Numer mieszkania	VarChar2(5)	N	
Miasto	Miasto	VarChar2(30)	T	

Tabela 20. Atrybuty relacji adresy

Poczty

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_poczty	Unikatowy numer poczty	Integer	T	Klucz główny
Kod_pocztowy	Kod pocztowy zgodny z formatem XX-XXX	Char(6)	T	
Miasto	Miasto urzędu pocztowego	VarChar2(30)	T	

Tabela 21. Atrybuty relacji poczty

Podobne modyfikacje, do tych opisanych powyżej, wykonano dla wielowartościowego atrybutu segmentowego Godziny otwarcia w relacji Placówki. Był to atrybut typu VarChar, odpowiadający liście dni i godzin, w których placówka jest otwarta. W celu pozbycia się tego atrybutu, nieprzyjmującego wartości atomowych, a także jednoczesnego uniknięcia redundancji danych, utworzono relacje Otwarcia oraz Czasy otwarcia. Pierwsza z nich, poza swoim kluczem głównym, posiada pole będące identyfikatorem placówki oraz pole będące identyfikatorem czasu otwarcia zdefiniowanego w relacji Otwarcia. Oba pola pełnią funkcję kluczy obcych. Relacja Czasy otwarcia, za pośrednictwem swoich atrybutów, określa godziny otwarcia schroniska w danych dniach. Atrybuty dwóch dodanych i opisanych wyżej relacji zostały przedstawione w tabelach 22 oraz 23.

Otwarcia

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_otwarcia	Unikatowy numer otwarcia	Integer	T	Klucz główny
Nr_placowki	Identyfikator placówki	Integer	T	Klucz obcy
Nr_czasu_otwarcia	Identyfikator czasu otwarcia	Integer	T	Klucz obcy

Tabela 22. Atrybuty relacji otwarcia

Czasy_otwarcia

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_czasu_otwarcia	Unikatowy numer otwarcia	Integer	T	Klucz główny
Godzina_otwarcia	Godzina otwarcia zgodna z formatem XX:XX	Char(5)	T	Klucz obcy
Godzina_zamkniecia	Godzina zamknięcia zgodna z formatem XX:XX	Char(5)	T	
Dzien	Dzień tygodnia	'PON', 'WT', 'SR', 'CZW', 'PT', 'SOB', 'NIEDZ'	T	

Tabela 23. Atrybuty relacji czasy otwarcia

Podobnych zmian dokonano także w przypadku atrybutu Stanowisko relacji Pracownicy. Zastąpiono go relacjami Zajmowanie stanowisk oraz Stanowiska. Modyfikacje wykonane zostały analogicznie do przypadków opisanych wyżej. Co istotne, zdecydowano się zrezygnować z ustalonej na etapie projektowania konceptualnego dziedziny atrybutu stanowisko. Decyzja ta została podjęta z myślą o potencjalnej możliwości zatrudnienia pracownika przez schronisko, na nowo powstałe stanowisko. Zatem po wykonaniu procesu normalizacji, typ atrybutu nazwa w relacji Stanowiska został ustawiony jako VarChar. Wszystkie atrybuty powstałych w wyniku normalizacji relacji Zajmowania stanowisk oraz Stanowiska zostały zebrane w tabelach 24 i 25.

Zajmowania_stanowisk

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_zajmowania_stanowisk	Unikatowy numer zajmowania stanowiska	Integer	T	Klucz główny
Nr_stanowiska	Identyfikator stanowiska	Integer	T	Klucz obcy
Nr_pracowinika	Identyfikator pracownika	Integer	T	Klucz obcy
Data_rozpoczecia	Data rozpoczęcia zajmowania stanowiska	Date	T	
Data_zakonczenia	Data zakończenia zajmowania stanowiska	Date	N	

Tabela 24. Atrybuty relacji zajmowania stanowisk

Stanowiska

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_stanowiska	Unikatowy numer stanowiska	Integer	T	Klucz główny
Nazwa	Nazwa stanowiska	VarChar2(20)	T	

Tabela 25. Atrybuty relacji stanowiska

Ze względu na powtarzanie się wartości atrybuty oraz jego brak dziedziny zdecydowano się na osobną reprezentację typu wyposażenia i egzemplarzy. W tym celu stworzono dwie relacje - wyposazenia_egzemplarze i typy_wyposazenia, związane ze sobą związkiem 0..n - 1..1. Relacja wyposazenia_egzemplarze przejęła związki dawnej encji wyposażenia. Atrybuty relacji wyposazenia_egzemplarze i typy_wyposazenia przedstawiono odpowiednio w tabelach 26 i 27.

Wyposazenia_egzemplarze

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_wyposazenia	Unikatowy numer egzemplarza wyposażenia	Integer	T	Klucz główny
Nr_placowki	Identyfikator placówki, w której znajduje się wyposażenie	Integer	T	Klucz obcy
Nr_boksu	Identyfikator boksu, w którym znajduje się wyposażenie	Integer	N	Klucz obcy
Nr_typu_wyposazenia	Identyfikator typu wyposażenia	Integer	T	Klucz obcy

Tabela 26. Atrybuty relacji wyposażenia egzemplarze

Typy_wyposazenia

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_typu_wyposazenia	Unikatowy numer typu wyposażenia	Integer	T	Klucz główny
Nr_producenta	Identyfikator producenta wyposażenia	Integer	T	Klucz obcy
Nazwa_typu	Nazwa typu wyposażenia	VarChar2(20)	T	
Model	Model wyposażenia	VarChar2(30)	N	

Tabela 27. Atrybuty relacji typy wyposażenia

Ze względu na powtarzanie się atrybutu producent w wielu relacjach oraz potencjalnego powtarzania się przez to grup w relacjach karmy i lekarstwa, zdecydowano się wydzielić atrybut producent do osobnej relacji. Wchodzi ona w związki 1..n, ze swoim obowiązkowym uczestnictwem, z relacjami karmy, lekarstwa oraz typy_wyposazenia. Atrybuty relacji producenci przedstawiono w tabeli 28.

Producenci

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_producenta	Unikatowy numer producenta	Integer	T	Klucz główny
Nazwa	Nazwa producenta	VarChar2(20)	T	

Tabela 28. Atrybuty relacji producenci

W encji boksy na etapie konceptualnym istniał atrybut segmentowy wymiary. W celu normalizacji zdecydowano się wydzielić ten atrybut do osobnej relacji wymiary. Atrybuty tej relacji przedstawiono w tabeli 29.

Wymiary

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_wymiarow	Unikatowy numer wymiarów	Integer	T	Klucz główny
Szerokosc	Szerokość wyrażona w centymetrach	Integer	T	
Wysokosc	Wysokość wyrażona w centymetrach	Integer	T	
Glebokosc	Głębokość wyrażona w centymetrach	Integer	T	

Tabela 29. Atrybuty relacji wymiary

Powtarzające się grupy istniały także w relacji zwierzęta. Były to atrybuty gatunek i rasa. Zdecydowano się wydzielić je do osobnej relacji gatunki_rasy. Atrybut rasa jest w niej nieobowiązkowy w celu możliwości uwzględnienia w bazie danych zwierząt nierasowych. Wszystkie atrybuty relacji gatunki_rasy przedstawiono w tabeli 30.

Gatunki_rasy

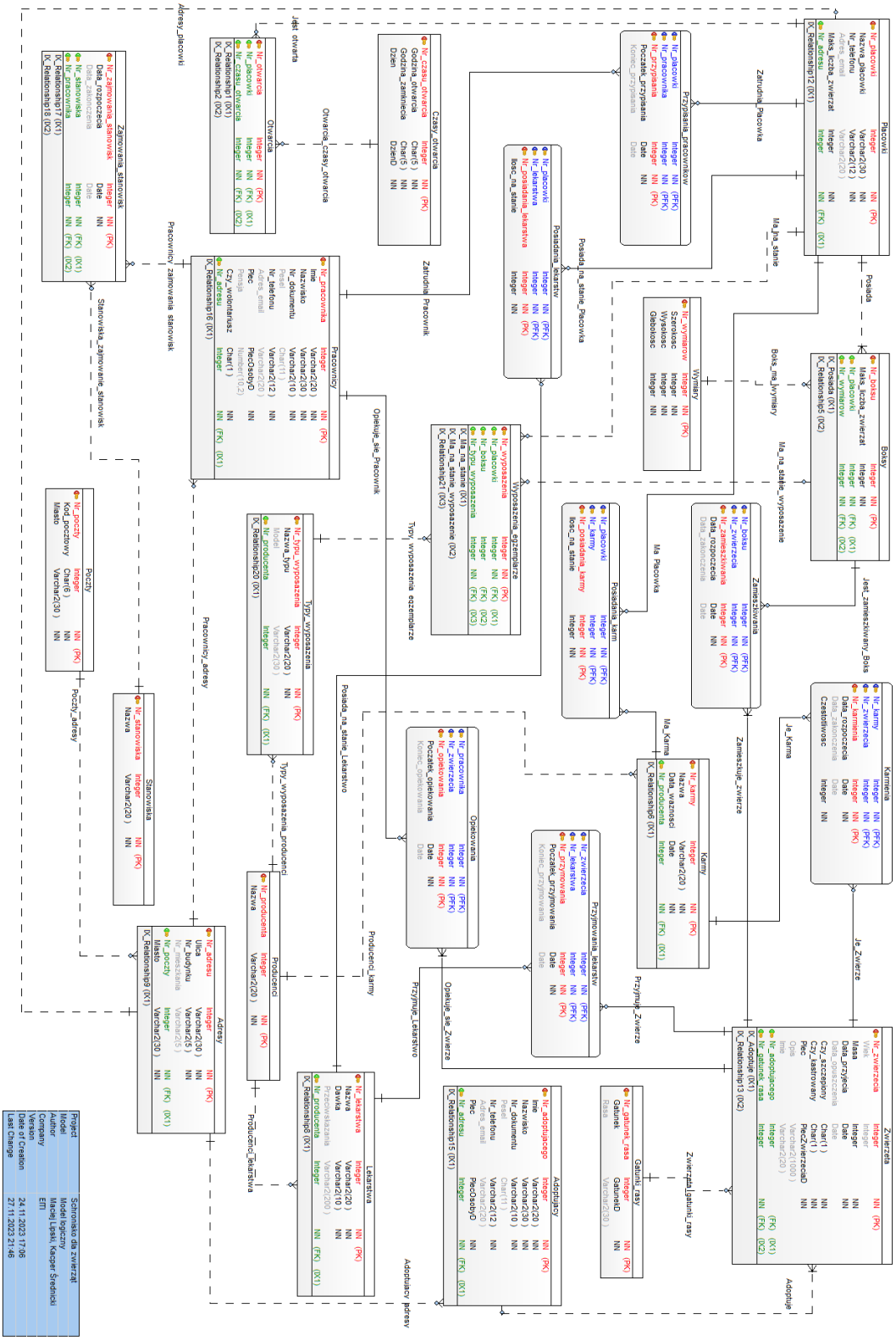
Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz
Nr_gatunek_rasa	Unikatowy numer kombinacji gatunek - rasa zwierzęcia	Integer	T	Klucz główny
Gatunek	Gatunek zwierzęcia	'pies','kot'	T	
Rasa	Rasa zwierzęcia	VarChar2(30)	N	

Tabela 30. Atrybuty relacji gatunki_rasy

Po sprowadzeniu relacji bazy danych do pierwszej postaci normalnej przeanalizowano je ponownie, tym razem pod kątem drugiej postaci normalnej. Zauważono, że wszystkie klucze kandydujące są kluczami prostymi i tym samym stwierdzono, że relacje są już w **drugiej postaci normalnej**.

Trzecia postać normalna zakłada dodatkowo, że każdy atrybut relacji, nie wchodzący w skład żadnego klucza kandydującego, nie jest przechodnio funkcyjnie zależny od żadnego klucza kandydującego tej samej relacji, czyli nie występują zależności tranzytywnie przechodnie. W przypadku opracowywanej bazy danych fakt ten jest spełniony. Baza danych w wyniku normalizacji została sprowadzona zatem do **trzeciej postaci normalnej**.

4.4. Schemat ER na poziomie modelu logicznego



Rysunek 3. Schemat ER bazy danych schroniska dla zwierząt na etapie logicznym

4.5. Więzy integralności

Przygotowany model logiczny bazy danych spełnia zbiór reguł zwanych więzami integralności. Zadbano, aby każda z relacji posiadała **unikatowy i obowiązkowy klucz główny**. W przypadku atrybutów występujących w więcej niż jednej relacji, ich typy zostały ustawione w sposób konsekwentny. Zdefiniowano także kilka specyficznych dziedzin, w celu jednoznacznego określenia wartości możliwych do przypisania konkretnemu atrybutowi. Każdemu kluczowi głównemu przypisano sekwencję, w celu jego automatycznej inkrementacji wraz z tworzeniem nowej krotki.

4.6. Proces denormalizacji – analiza i przykłady

Po doprowadzeniu relacji do trzeciej postaci normalnej, dokonano analizy potencjalnej denormalizacji - złagodzenia reguł normalizacji w celu szybszego dostępu do danych.

Możliwym przypadkiem denormalizacji dla projektu byłby powrót do połączenia relacji wyposazenia_egzemplarze i typy_wyposazenia. Takie działanie doprowadziłoby do powtarzania się grupy atrybutów model, producent, typ, jednak przyśpieszyłoby dostęp do informacji o danym egzemplarzu wyposażenia. Innym możliwym przejawem denormalizacji jest likwidacja relacji wymiary i scalenie jej z relacją boksy. Przyśpieszyłoby to dostęp do danych o wymiarach danego boksu jednak wprowadziłoby redundancję danych. Możliwe byłoby także scalenie tablicy poczty i adresy. Wprowadziłoby to powtarzanie się wartości kodów pocztowych, jednak dawałoby łatwiejszy dostęp do kodu danego adresu.

Choć widocznych jest wiele możliwości denormalizacji, zdecydowano się na pozostawienie bazy w trzeciej postaci normalnej. Taki stan pozwoli na zachowanie względnej prostoty i uporządkowania bazy. Pozwala też na łatwe uaktualnianie wartości, bez konieczności uaktualniania ich w wielu miejscach oraz na bezpieczne usuwanie krotek bez utraty innych danych.

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Po ukończeniu modelu na poziomie logicznym, przystąpiono do weryfikacji możliwości realizacji transakcji na etapie specyfikacji wymagań (sekcja 2). Raport z procesu weryfikacji wykonywalności transakcji umieszczono w tabelach 31, 32, 33, 34.

Transakcja	Czy wykonalna	Wykorzystywane relacje
Modyfikacja informacji o godzinach otwarcia	Tak	Placówki, Otwarcia, Czasy_otwarcia
Modyfikacja informacji o maksymalnej liczbie zwierząt w schronisku	Tak	Placówki
Rozbudowa schroniska o dodatkowe boksy	Tak	Placówki, Boksy

Tabela 31. Weryfikacja wykonywalności transakcji z perspektywy obsługi placówki

Transakcja	Czy wykonalna	Wykorzystywane relacje
Przyjęcie zwierzęcia do schroniska	Tak	Zamieszkiwania, Zwierzęta, Boksy, Opiekowania, Pracownicy, Gatunki_rasy
Adopcja zwierząt	Tak	Adoptujący, Zwierzęta
Przypisanie zwierzęciu aktualnie zamieszkiwanego bosku	Tak	Zwierzęta, Zamieszkiwania, Boksy
Przypisanie zwierzęciu informacji o przyjmowanych lekach	Tak	Zwierzęta, Lekarstwa, Przyjmowanie lekarstw
Przypisanie zwierzęciu informacji o spożywanych karmach	Tak	Zwierzęta, Karmienia, Karmy
Dodawanie i modyfikacja informacji o aktualnych opiekunach zwierzęcia	Tak	Pracownicy, Opiekowania, Pracownicy

Tabela 32. Weryfikacja wykonywalności transakcji z perspektywy obsługi zwierząt

Transakcja	Czy wykonalna	Wykorzystywane relacje
Wprowadzanie informacji o nowym wyposażeniu	TAK	Wyposazenia_egzemplarze, Wyposzenia_typy, Boksy, Placówki
Ewidencja lekarstw, ich wykorzystywania	TAK	Lekarstwa, Producenci, Leczenia
Ewidencja karm, ich wykorzystywania	TAK	Karmy, Karmienia, Producenci

Tabela 33. Weryfikacja wykonywalności transakcji z perspektywy obsługi wyposażenia

Transakcja	Czy wykonalna	Wykorzystywane relacje
Zatrudnianie nowych pracowników	Tak	Pracownicy, Przypisania pracowników, Zajmowania stanowisk
Dodawanie informacji o płacach	Tak	Pracownicy
Zatrudnianie pracowników w ramach wolontariatu	Tak	Pracownicy
Wprowadzanie informacji o funkcjach pełnionych przez pracownika	Tak	Pracownicy, Stanowiska

Tabela 34. Weryfikacja wykonywalności transakcji z perspektywy obsługi pracowników

5.2. Skrypt SQL zakładający bazę danych

```
/*
Created: 24.11.2023
Modified: 27.11.2023
Project: Schronisko dla zwierząt
Model: Model logiczny
Company: EITI
Author: Maciej Lipski, Kacper Średnicki
Database: Oracle 19c
*/

-- Create sequences section -------------------------------------------------
CREATE SEQUENCE Sequence_Nr_placowki
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_boksu
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_karmienia
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_zwierzecia
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_przypisania
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_gatunek_rasa
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_adoptujacego
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
```

```
/
CREATE SEQUENCE Sequence_Nr_lekarstwa
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_adresu
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_stanowiska
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_pracownika
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_typu_wyposazenia
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_otwarcia
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_zajmowania_stanowisk
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_posiadania_lekarstwa
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_czasu_otwarcia
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
```

```
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_poczty
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_opiekowania
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_przyjmowania
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_wyposazenia
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_posiadania_karmy
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_wymiarow
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_zamieszkiwania
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_producenta
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/
CREATE SEQUENCE Sequence_Nr_karmy
 INCREMENT BY 1
```

```
 START WITH 1
 NOMAXVALUE
 NOMINVALUE
 CACHE 20
/

-- Create tables section -------------------------------------------------

-- Table Placowki
CREATE TABLE Placowki(
  Nr_placowki Integer NOT NULL,
  Nazwa_placowki Varchar2(30 ) NOT NULL,
  Nr_telefonu Varchar2(12 ) NOT NULL,
  Adres_email Varchar2(20 ),
  Maks_liczba_zwierzat Integer NOT NULL,
  Nr_adresu Integer NOT NULL
)
/
-- Create indexes for table Placowki
CREATE INDEX IX_Relationship12 ON Placowki (Nr_adresu)
/
-- Add keys for table Placowki
ALTER TABLE Placowki ADD CONSTRAINT Unique_Identifier1 PRIMARY KEY (Nr_placowki)
/
-- Create triggers for table Placowki
CREATE TRIGGER Trigger1
  BEFORE
  ON Placowki
  BEGIN
    /*trigger_body*/
END;
/

-- Table Zwierzeta
CREATE TABLE Zwierzeta(
  Nr_zwierzecia Integer NOT NULL,
  Wiek Integer,
  Masa Integer NOT NULL,
  Data_przyjecia Date NOT NULL,
  Data_opuszczenia Date,
  Czy_szczepiony Char(1 ) NOT NULL,
  Czy_kastrowany Char(1 ) NOT NULL,
  Plec Varchar2(10 ) NOT NULL
        CHECK (Plec IN ('Samiec','Samica')),
  Opis Varchar2(1000 ),
  Imie Varchar2(20 ),
  Nr_adoptujacego Integer,
  Nr_gatunek_rasa Integer NOT NULL
)
/
-- Create indexes for table Zwierzeta
CREATE INDEX IX_Adoptuje ON Zwierzeta (Nr_adoptujacego)
/
CREATE INDEX IX_Relationship13 ON Zwierzeta (Nr_gatunek_rasa)
/
-- Add keys for table Zwierzeta
ALTER TABLE Zwierzeta ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY (Nr_zwierzecia)
/

-- Table Boksy
CREATE TABLE Boksy(
```

```
  Nr_boksu Integer NOT NULL,
  Maks_liczba_zwierzat Integer NOT NULL,
  Nr_placowki Integer NOT NULL,
  Nr_wymiarow Integer NOT NULL
)
/
-- Create indexes for table Boksy
CREATE INDEX IX_Posiada ON Boksy (Nr_placowki)
/
CREATE INDEX IX_Relationship5 ON Boksy (Nr_wymiarow)
/
-- Add keys for table Boksy
ALTER TABLE Boksy ADD CONSTRAINT Unique_Identifier3 PRIMARY KEY (Nr_boksu)
/

-- Table Adoptujacy
CREATE TABLE Adoptujacy(
  Nr_adoptujacego Integer NOT NULL,
  Imie Varchar2(20 ) NOT NULL,
  Nazwisko Varchar2(30 ) NOT NULL,
  Nr_dokumentu Varchar2(10 ) NOT NULL,
  Pesel Char(11 ),
  Nr_telefonu Varchar2(12 ) NOT NULL,
  Adres_email Varchar2(20 ),
  Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('M','K')),
  Nr_adresu Integer NOT NULL
)
/
-- Create indexes for table Adoptujacy
CREATE INDEX IX_Relationship15 ON Adoptujacy (Nr_adresu)
/
-- Add keys for table Adoptujacy
ALTER TABLE Adoptujacy ADD CONSTRAINT Unique_Identifier4 PRIMARY KEY (Nr_adoptujacego)
/
-- Table Lekarstwa
CREATE TABLE Lekarstwa(
  Nr_lekarstwa Integer NOT NULL,
  Nazwa Varchar2(20 ) NOT NULL,
  Dawka Varchar2(10 ) NOT NULL,
  Przeciwskazania Varchar2(200 ),
  Nr_producenta Integer NOT NULL
)
/
-- Create indexes for table Lekarstwa
CREATE INDEX IX_Relationship8 ON Lekarstwa (Nr_producenta)
/
-- Add keys for table Lekarstwa
ALTER TABLE Lekarstwa ADD CONSTRAINT Unique_Identifier6 PRIMARY KEY (Nr_lekarstwa)
/

-- Table Karmy
CREATE TABLE Karmy(
  Nr_karmy Integer NOT NULL,
  Nazwa Varchar2(20 ) NOT NULL,
  Data_waznosci Date NOT NULL,
  Nr_producenta Integer NOT NULL
)
/
-- Create indexes for table Karmy
CREATE INDEX IX_Relationship6 ON Karmy (Nr_producenta)
```

```
/
-- Add keys for table Karmy
ALTER TABLE Karmy ADD CONSTRAINT Unique_Identifier9 PRIMARY KEY (Nr_karmy)
/

-- Table Wyposazenia_egzemplarze
CREATE TABLE Wyposazenia_egzemplarze(
  Nr_wyposazenia Integer NOT NULL,
  Nr_placowki Integer NOT NULL,
  Nr_boksu Integer,
  Nr_typu_wyposazenia Integer NOT NULL
)
/
-- Create indexes for table Wyposazenia_egzemplarze
CREATE INDEX IX_Ma_na_stanie ON Wyposazenia_egzemplarze (Nr_placowki)
/
CREATE INDEX IX_Ma_na_stanie_wyposazenie ON Wyposazenia_egzemplarze (Nr_boksu)
/
CREATE INDEX IX_Relationship21 ON Wyposazenia_egzemplarze (Nr_typu_wyposazenia)
/
-- Add keys for table Wyposazenia_egzemplarze
ALTER TABLE Wyposazenia_egzemplarze ADD CONSTRAINT Unique_Identifier10 PRIMARY KEY (Nr_wyposazenia)
/

-- Table Pracownicy
CREATE TABLE Pracownicy(
  Nr_pracownika Integer NOT NULL,
  Imie Varchar2(20 ) NOT NULL,
  Nazwisko Varchar2(30 ) NOT NULL,
  Nr_dokumentu Varchar2(10 ) NOT NULL,
  Pesel Char(11 ),
  Nr_telefonu Varchar2(12 ) NOT NULL,
  Adres_email Varchar2(20 ),
  Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('M','K')),
  Pensja Number(10,2),
  Czy_wolontariusz Char(1 ) NOT NULL,
  Nr_adresu Integer NOT NULL
)
/
-- Create indexes for table Pracownicy
CREATE INDEX IX_Relationship16 ON Pracownicy (Nr_adresu)
/
-- Add keys for table Pracownicy
ALTER TABLE Pracownicy ADD CONSTRAINT Unique_Identifier11 PRIMARY KEY (Nr_pracownika)
/

-- Table Zamieszkiwania
CREATE TABLE Zamieszkiwania(
  Nr_boksu Integer NOT NULL,
  Nr_zwierzecia Integer NOT NULL,
  Nr_zamieszkiwania Integer NOT NULL,
  Data_rozpoczecia Date NOT NULL,
  Data_zakonczenia Date
)
/

-- Table Przypisania_pracownikow
CREATE TABLE Przypisania_pracownikow(
  Nr_placowki Integer NOT NULL,
  Nr_pracownika Integer NOT NULL,
```

```
  Nr_przypisania Integer NOT NULL,
  Poczatek_przypisania Date NOT NULL,
  Koniec_przypisania Date
)
/
-- Table Posiadania_lekarstw
CREATE TABLE Posiadania_lekarstw(
  Nr_placowki Integer NOT NULL,
  Nr_lekarstwa Integer NOT NULL,
  Nr_posiadania_lekarstwa Integer NOT NULL,
  Ilosc_na_stanie Integer NOT NULL
)
/

-- Table Przyjmowania_lekarstw
CREATE TABLE Przyjmowania_lekarstw(
  Nr_zwierzecia Integer NOT NULL,
  Nr_lekarstwa Integer NOT NULL,
  Nr_przymowania Integer NOT NULL,
  Poczatek_przyjmowania Date NOT NULL,
  Koniec_przyjmowania Date
)
/

-- Table Posiadania_karm
CREATE TABLE Posiadania_karm(
  Nr_placowki Integer NOT NULL,
  Nr_karmy Integer NOT NULL,
  Nr_posiadania_karmy Integer NOT NULL,
  Ilosc_na_stanie Integer NOT NULL
)
/

-- Table Opiekowania

CREATE TABLE Opiekowania(
  Nr_pracownika Integer NOT NULL,
  Nr_zwierzecia Integer NOT NULL,
  Nr_opiekowania Integer NOT NULL,
  Poczatek_opiekowania Date NOT NULL,
  Koniec_opiekowania Date
)
/

-- Table Karmienia
CREATE TABLE Karmienia(
  Nr_karmy Integer NOT NULL,
  Nr_zwierzecia Integer NOT NULL,
  Nr_karmienia Integer NOT NULL,
  Data_rozpoczecia Date NOT NULL,
  Data_zakonczenia Date,
  Czestotliwosc Integer NOT NULL
)
/

-- Table and Columns comments section
COMMENT ON COLUMN Karmienia.Czestotliwosc IS 'Co ile godzin dostaje karmę'
/

-- Table Adresy
CREATE TABLE Adresy(
```

```
  Nr_adresu Integer NOT NULL,
  Ulica Varchar2(30 ) NOT NULL,
  Nr_budynku Varchar2(5 ) NOT NULL,
  Nr_mieszkania Varchar2(5 ),
  Nr_poczty Integer NOT NULL,
  Miasto Varchar2(30 ) NOT NULL
)
/
-- Create indexes for table Adresy
CREATE INDEX IX_Relationship9 ON Adresy (Nr_poczty)
/
-- Add keys for table Adresy
ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (Nr_adresu)
/

-- Table Poczty
CREATE TABLE Poczty(
  Nr_poczty Integer NOT NULL,
  Kod_pocztowy Char(6 ) NOT NULL,
  Miasto Varchar2(30 ) NOT NULL
)
/
-- Add keys for table Poczty
ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (Nr_poczty)
/
-- Table and Columns comments section
COMMENT ON COLUMN Poczty.Kod_pocztowy IS 'Format XX-XXX'
/

-- Table Czasy_otwarcia
CREATE TABLE Czasy_otwarcia(
  Nr_czasu_otwarcia Integer NOT NULL,
  Godzina_otwarcia Char(5 ) NOT NULL,
  Godzina_zamkniecia Char(5 ) NOT NULL,
  Dzien Varchar2(5 ) NOT NULL
        CHECK (Dzien IN ('PON','WT','SR','CZW','PT','SOB','NIEDZ'))
)
/
-- Add keys for table Czasy_otwarcia
ALTER TABLE Czasy_otwarcia ADD CONSTRAINT PK_Czasy_otwarcia PRIMARY KEY (Nr_czasu_otwarcia)
/
-- Table and Columns comments section
COMMENT ON COLUMN Czasy_otwarcia.Godzina_otwarcia IS 'Format XX:XX'
/
COMMENT ON COLUMN Czasy_otwarcia.Godzina_zamkniecia IS 'Format XX:XX'
/

-- Table Otwarcia
CREATE TABLE Otwarcia(
  Nr_otwarcia Integer NOT NULL,
  Nr_placowki Integer NOT NULL,
  Nr_czasu_otwarcia Integer NOT NULL
)
/
-- Create indexes for table Otwarcia
CREATE INDEX IX_Relationship1 ON Otwarcia (Nr_placowki)
/
CREATE INDEX IX_Relationship2 ON Otwarcia (Nr_czasu_otwarcia)
/
-- Add keys for table Otwarcia
ALTER TABLE Otwarcia ADD CONSTRAINT PK_Otwarcia PRIMARY KEY (Nr_otwarcia)
```

```
/

-- Table Wymiary
CREATE TABLE Wymiary(
  Nr_wymiarow Integer NOT NULL,
  Szerokosc Integer NOT NULL,
  Wysokosc Integer NOT NULL,
  Glebokosc Integer NOT NULL
)
/
-- Add keys for table Wymiary
ALTER TABLE Wymiary ADD CONSTRAINT PK_Wymiary PRIMARY KEY (Nr_wymiarow)
/
-- Table and Columns comments section
COMMENT ON COLUMN Wymiary.Szerokosc IS 'Wartość w centymetrach'
/
COMMENT ON COLUMN Wymiary.Wysokosc IS 'Wartość w centymetrach'
/
COMMENT ON COLUMN Wymiary.Glebokosc IS 'Wartość w centymetrach'
/

-- Table Producenci
CREATE TABLE Producenci(
  Nr_producenta Integer NOT NULL,
  Nazwa Varchar2(20 ) NOT NULL
)
/
-- Add keys for table Producenci
ALTER TABLE Producenci ADD CONSTRAINT PK_Producenci PRIMARY KEY (Nr_producenta)
/

-- Table Gatunki_rasy
CREATE TABLE Gatunki_rasy(
  Nr_gatunek_rasa Integer NOT NULL,
  Gatunek Varchar2(30 ) NOT NULL
        CHECK (Gatunek IN ('Pies','Kot')),
  Rasa Varchar2(30 )
)
/
-- Add keys for table Gatunki_rasy
ALTER TABLE Gatunki_rasy ADD CONSTRAINT PK_Gatunki_rasy PRIMARY KEY (Nr_gatunek_rasa)
/

-- Table Stanowiska
CREATE TABLE Stanowiska(
  Nr_stanowiska Integer NOT NULL,
  Nazwa Varchar2(20 ) NOT NULL
)
/
-- Add keys for table Stanowiska
ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stanowiska PRIMARY KEY (Nr_stanowiska)
/

-- Table Zajmowania_stanowisk
CREATE TABLE Zajmowania_stanowisk(
  Nr_zajmowania_stanowisk Integer NOT NULL,
  Data_rozpoczecia Date NOT NULL,
  Data_zakonczenia Date,
  Nr_stanowiska Integer NOT NULL,
  Nr_pracownika Integer NOT NULL
)
```

```
/
-- Create indexes for table Zajmowania_stanowisk
CREATE INDEX IX_Relationship17 ON Zajmowania_stanowisk (Nr_stanowiska)
/
CREATE INDEX IX_Relationship18 ON Zajmowania_stanowisk (Nr_pracownika)
/

-- Add keys for table Zajmowania_stanowisk
ALTER TABLE Zajmowania_stanowisk ADD CONSTRAINT PK_Zajmowania_stanowisk PRIMARY KEY
↪  (Nr_zajmowania_stanowisk)
/

-- Table Typy_wyposazenia
CREATE TABLE Typy_wyposazenia(
  Nr_typu_wyposazenia Integer NOT NULL,
  Nazwa_typu Varchar2(20 ) NOT NULL,
  Model Varchar2(30 ),
  Nr_producenta Integer NOT NULL
)
/
-- Create indexes for table Typy_wyposazenia
CREATE INDEX IX_Relationship20 ON Typy_wyposazenia (Nr_producenta)
/
-- Add keys for table Typy_wyposazenia
ALTER TABLE Typy_wyposazenia ADD CONSTRAINT PK_Typy_wyposazenia PRIMARY KEY (Nr_typu_wyposazenia)
/

-- Trigger for sequence Sequence_Nr_placowki for column Nr_placowki in table Placowki ---------
CREATE OR REPLACE TRIGGER ts_Placowki_Sequence_Nr_placowki BEFORE INSERT
ON Placowki FOR EACH ROW
BEGIN
  :new.Nr_placowki := Sequence_Nr_placowki.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Placowki_Sequence_Nr_placowki AFTER UPDATE OF Nr_placowki
ON Placowki FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_placowki in table Placowki as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_zwierzecia for column Nr_zwierzecia in table Zwierzeta ---------
CREATE OR REPLACE TRIGGER ts_Zwierzeta_Sequence_Nr_zwierzecia BEFORE INSERT
ON Zwierzeta FOR EACH ROW
BEGIN
  :new.Nr_zwierzecia := Sequence_Nr_zwierzecia.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Zwierzeta_Sequence_Nr_zwierzecia AFTER UPDATE OF Nr_zwierzecia
ON Zwierzeta FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_zwierzecia in table Zwierzeta as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_boksu for column Nr_boksu in table Boksy ---------
CREATE OR REPLACE TRIGGER ts_Boksy_Sequence_Nr_boksu BEFORE INSERT
ON Boksy FOR EACH ROW
BEGIN
  :new.Nr_boksu := Sequence_Nr_boksu.nextval;
END;
```

```
/
CREATE OR REPLACE TRIGGER tsu_Boksy_Sequence_Nr_boksu AFTER UPDATE OF Nr_boksu
ON Boksy FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_boksu in table Boksy as it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_adoptujacego for column Nr_adoptujacego in table Adoptujacy
↪  ---------
CREATE OR REPLACE TRIGGER ts_Adoptujacy_Sequence_Nr_adoptujacego BEFORE INSERT
ON Adoptujacy FOR EACH ROW
BEGIN
  :new.Nr_adoptujacego := Sequence_Nr_adoptujacego.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Adoptujacy_Sequence_Nr_adoptujacego AFTER UPDATE OF Nr_adoptujacego
ON Adoptujacy FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_adoptujacego in table Adoptujacy as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_lekarstwa for column Nr_lekarstwa in table Lekarstwa ---------
CREATE OR REPLACE TRIGGER ts_Lekarstwa_Sequence_Nr_lekarstwa BEFORE INSERT
ON Lekarstwa FOR EACH ROW
BEGIN
  :new.Nr_lekarstwa := Sequence_Nr_lekarstwa.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Lekarstwa_Sequence_Nr_lekarstwa AFTER UPDATE OF Nr_lekarstwa
ON Lekarstwa FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_lekarstwa in table Lekarstwa as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_karmy for column Nr_karmy in table Karmy ---------
CREATE OR REPLACE TRIGGER ts_Karmy_Sequence_Nr_karmy BEFORE INSERT
ON Karmy FOR EACH ROW
BEGIN
  :new.Nr_karmy := Sequence_Nr_karmy.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Karmy_Sequence_Nr_karmy AFTER UPDATE OF Nr_karmy
ON Karmy FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_karmy in table Karmy as it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_wyposazenia for column Nr_wyposazenia in table
↪  Wyposazenia_egzemplarze ---------
CREATE OR REPLACE TRIGGER ts_Wyposazenia_egzemplarze_Sequence_Nr_wyposazenia BEFORE INSERT
ON Wyposazenia_egzemplarze FOR EACH ROW
BEGIN
  :new.Nr_wyposazenia := Sequence_Nr_wyposazenia.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wyposazenia_egzemplarze_Sequence_Nr_wyposazenia AFTER UPDATE OF
↪  Nr_wyposazenia
ON Wyposazenia_egzemplarze FOR EACH ROW
BEGIN
```

```
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wyposazenia in table Wyposazenia_egzemplarze
  ↪ as it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_pracownika for column Nr_pracownika in table Pracownicy ---------
CREATE OR REPLACE TRIGGER ts_Pracownicy_Sequence_Nr_pracownika BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN
  :new.Nr_pracownika := Sequence_Nr_pracownika.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pracownicy_Sequence_Nr_pracownika AFTER UPDATE OF Nr_pracownika
ON Pracownicy FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_pracownika in table Pracownicy as it uses
  ↪ sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_zamieszkiwania for column Nr_zamieszkiwania in table
↪ Zamieszkiwania ---------
CREATE OR REPLACE TRIGGER ts_Zamieszkiwania_Sequence_Nr_zamieszkiwania BEFORE INSERT
ON Zamieszkiwania FOR EACH ROW
BEGIN
  :new.Nr_zamieszkiwania := Sequence_Nr_zamieszkiwania.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Zamieszkiwania_Sequence_Nr_zamieszkiwania AFTER UPDATE OF
↪ Nr_zamieszkiwania
ON Zamieszkiwania FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_zamieszkiwania in table Zamieszkiwania as it
  ↪ uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_przypisania for column Nr_przypisania in table
↪ Przypisania_pracownikow ---------
CREATE OR REPLACE TRIGGER ts_Przypisania_pracownikow_Sequence_Nr_przypisania BEFORE INSERT
ON Przypisania_pracownikow FOR EACH ROW
BEGIN
  :new.Nr_przypisania := Sequence_Nr_przypisania.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Przypisania_pracownikow_Sequence_Nr_przypisania AFTER UPDATE OF
↪ Nr_przypisania
ON Przypisania_pracownikow FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_przypisania in table Przypisania_pracownikow
  ↪ as it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_posiadania_lekarstwa for column Nr_posiadania_lekarstwa in table
↪ Posiadania_lekarstw ---------
CREATE OR REPLACE TRIGGER ts_Posiadania_lekarstw_Sequence_Nr_posiadania_lekarstwa BEFORE INSERT
ON Posiadania_lekarstw FOR EACH ROW
BEGIN
  :new.Nr_posiadania_lekarstwa := Sequence_Nr_posiadania_lekarstwa.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Posiadania_lekarstw_Sequence_Nr_posiadania_lekarstwa AFTER UPDATE OF
↪ Nr_posiadania_lekarstwa
ON Posiadania_lekarstw FOR EACH ROW
```

```
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_posiadania_lekarstwa in table
  ↪  Posiadania_lekarstw as it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_przyjmowania for column Nr_przymowania in table
↪  Przyjmowania_lekarstw ---------
CREATE OR REPLACE TRIGGER ts_Przyjmowania_lekarstw_Sequence_Nr_przyjmowania BEFORE INSERT
ON Przyjmowania_lekarstw FOR EACH ROW
BEGIN
  :new.Nr_przymowania := Sequence_Nr_przyjmowania.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Przyjmowania_lekarstw_Sequence_Nr_przyjmowania AFTER UPDATE OF
↪  Nr_przymowania
ON Przyjmowania_lekarstw FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_przymowania in table Przyjmowania_lekarstw
  ↪  as it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_posiadania_karmy for column Nr_posiadania_karmy in table
↪  Posiadania_karm ---------
CREATE OR REPLACE TRIGGER ts_Posiadania_karm_Sequence_Nr_posiadania_karmy BEFORE INSERT
ON Posiadania_karm FOR EACH ROW
BEGIN
  :new.Nr_posiadania_karmy := Sequence_Nr_posiadania_karmy.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Posiadania_karm_Sequence_Nr_posiadania_karmy AFTER UPDATE OF
↪  Nr_posiadania_karmy
ON Posiadania_karm FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_posiadania_karmy in table Posiadania_karm as
  ↪  it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_opiekowania for column Nr_opiekowania in table Opiekowania
↪  ---------
CREATE OR REPLACE TRIGGER ts_Opiekowania_Sequence_Nr_opiekowania BEFORE INSERT
ON Opiekowania FOR EACH ROW
BEGIN
  :new.Nr_opiekowania := Sequence_Nr_opiekowania.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Opiekowania_Sequence_Nr_opiekowania AFTER UPDATE OF Nr_opiekowania
ON Opiekowania FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_opiekowania in table Opiekowania as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_karmienia for column Nr_karmienia in table Karmienia ---------
CREATE OR REPLACE TRIGGER ts_Karmienia_Sequence_Nr_karmienia BEFORE INSERT
ON Karmienia FOR EACH ROW
BEGIN
  :new.Nr_karmienia := Sequence_Nr_karmienia.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Karmienia_Sequence_Nr_karmienia AFTER UPDATE OF Nr_karmienia
ON Karmienia FOR EACH ROW
```

```
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_karmienia in table Karmienia as it uses
  ↪ sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_adresu for column Nr_adresu in table Adresy ---------
CREATE OR REPLACE TRIGGER ts_Adresy_Sequence_Nr_adresu BEFORE INSERT
ON Adresy FOR EACH ROW
BEGIN
  :new.Nr_adresu := Sequence_Nr_adresu.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Adresy_Sequence_Nr_adresu AFTER UPDATE OF Nr_adresu
ON Adresy FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_adresu in table Adresy as it uses
  ↪ sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_poczty for column Nr_poczty in table Poczty ---------
CREATE OR REPLACE TRIGGER ts_Poczty_Sequence_Nr_poczty BEFORE INSERT
ON Poczty FOR EACH ROW
BEGIN
  :new.Nr_poczty := Sequence_Nr_poczty.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Poczty_Sequence_Nr_poczty AFTER UPDATE OF Nr_poczty
ON Poczty FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_poczty in table Poczty as it uses
  ↪ sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_czasu_otwarcia for column Nr_czasu_otwarcia in table
↪ Czasy_otwarcia ---------
CREATE OR REPLACE TRIGGER ts_Czasy_otwarcia_Sequence_Nr_czasu_otwarcia BEFORE INSERT
ON Czasy_otwarcia FOR EACH ROW
BEGIN
  :new.Nr_czasu_otwarcia := Sequence_Nr_czasu_otwarcia.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Czasy_otwarcia_Sequence_Nr_czasu_otwarcia AFTER UPDATE OF
↪ Nr_czasu_otwarcia
ON Czasy_otwarcia FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_czasu_otwarcia in table Czasy_otwarcia as it
  ↪ uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_otwarcia for column Nr_otwarcia in table Otwarcia ---------
CREATE OR REPLACE TRIGGER ts_Otwarcia_Sequence_Nr_otwarcia BEFORE INSERT
ON Otwarcia FOR EACH ROW
BEGIN
  :new.Nr_otwarcia := Sequence_Nr_otwarcia.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Otwarcia_Sequence_Nr_otwarcia AFTER UPDATE OF Nr_otwarcia
ON Otwarcia FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_otwarcia in table Otwarcia as it uses
  ↪ sequence.');
```

```
END;
/
-- Trigger for sequence Sequence_Nr_wymiarow for column Nr_wymiarow in table Wymiary ---------
CREATE OR REPLACE TRIGGER ts_Wymiary_Sequence_Nr_wymiarow BEFORE INSERT
ON Wymiary FOR EACH ROW
BEGIN
  :new.Nr_wymiarow := Sequence_Nr_wymiarow.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wymiary_Sequence_Nr_wymiarow AFTER UPDATE OF Nr_wymiarow
ON Wymiary FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wymiarow in table Wymiary as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_producenta for column Nr_producenta in table Producenci ---------
CREATE OR REPLACE TRIGGER ts_Producenci_Sequence_Nr_producenta BEFORE INSERT
ON Producenci FOR EACH ROW
BEGIN
  :new.Nr_producenta := Sequence_Nr_producenta.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Producenci_Sequence_Nr_producenta AFTER UPDATE OF Nr_producenta
ON Producenci FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_producenta in table Producenci as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_gatunek_rasa for column Nr_gatunek_rasa in table Gatunki_rasy
↪  ---------
CREATE OR REPLACE TRIGGER ts_Gatunki_rasy_Sequence_Nr_gatunek_rasa BEFORE INSERT
ON Gatunki_rasy FOR EACH ROW
BEGIN
  :new.Nr_gatunek_rasa := Sequence_Nr_gatunek_rasa.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Gatunki_rasy_Sequence_Nr_gatunek_rasa AFTER UPDATE OF Nr_gatunek_rasa
ON Gatunki_rasy FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_gatunek_rasa in table Gatunki_rasy as it
  ↪  uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_stanowiska for column Nr_stanowiska in table Stanowiska ---------
CREATE OR REPLACE TRIGGER ts_Stanowiska_Sequence_Nr_stanowiska BEFORE INSERT
ON Stanowiska FOR EACH ROW
BEGIN
  :new.Nr_stanowiska := Sequence_Nr_stanowiska.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Stanowiska_Sequence_Nr_stanowiska AFTER UPDATE OF Nr_stanowiska
ON Stanowiska FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_stanowiska in table Stanowiska as it uses
  ↪  sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_zajmowania_stanowisk for column Nr_zajmowania_stanowisk in table
↪  Zajmowania_stanowisk ---------
```

```
CREATE OR REPLACE TRIGGER ts_Zajmowania_stanowisk_Sequence_Nr_zajmowania_stanowisk BEFORE INSERT
ON Zajmowania_stanowisk FOR EACH ROW
BEGIN
  :new.Nr_zajmowania_stanowisk := Sequence_Nr_zajmowania_stanowisk.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Zajmowania_stanowisk_Sequence_Nr_zajmowania_stanowisk AFTER UPDATE OF
↪  Nr_zajmowania_stanowisk
ON Zajmowania_stanowisk FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_zajmowania_stanowisk in table
  ↪  Zajmowania_stanowisk as it uses sequence.');
END;
/
-- Trigger for sequence Sequence_Nr_typu_wyposazenia for column Nr_typu_wyposazenia in table
↪  Typy_wyposazenia ---------
CREATE OR REPLACE TRIGGER ts_Typy_wyposazenia_Sequence_Nr_typu_wyposazenia BEFORE INSERT
ON Typy_wyposazenia FOR EACH ROW
BEGIN
  :new.Nr_typu_wyposazenia := Sequence_Nr_typu_wyposazenia.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Typy_wyposazenia_Sequence_Nr_typu_wyposazenia AFTER UPDATE OF
↪  Nr_typu_wyposazenia
ON Typy_wyposazenia FOR EACH ROW
BEGIN
  RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_typu_wyposazenia in table Typy_wyposazenia
  ↪  as it uses sequence.');
END;
/

-- Create foreign keys (relationships) section -------------------------------------------------
ALTER TABLE Boksy ADD CONSTRAINT Posiada FOREIGN KEY (Nr_placowki) REFERENCES Placowki (Nr_placowki)
/

ALTER TABLE Wyposazenia_egzemplarze ADD CONSTRAINT Ma_na_stanie FOREIGN KEY (Nr_placowki) REFERENCES
↪  Placowki (Nr_placowki)
/

ALTER TABLE Zwierzeta ADD CONSTRAINT Adoptuje FOREIGN KEY (Nr_adoptujacego) REFERENCES Adoptujacy
↪  (Nr_adoptujacego)
/

ALTER TABLE Wyposazenia_egzemplarze ADD CONSTRAINT Ma_na_stanie_wyposazenie FOREIGN KEY (Nr_boksu)
↪  REFERENCES Boksy (Nr_boksu)
/

ALTER TABLE Otwarcia ADD CONSTRAINT Jest_otwarta FOREIGN KEY (Nr_placowki) REFERENCES Placowki
↪  (Nr_placowki)
/

ALTER TABLE Otwarcia ADD CONSTRAINT Otwarcia_czasy_otwarcia FOREIGN KEY (Nr_czasu_otwarcia) REFERENCES
↪  Czasy_otwarcia (Nr_czasu_otwarcia)
/

ALTER TABLE Boksy ADD CONSTRAINT Boks_ma_wymiary FOREIGN KEY (Nr_wymiarow) REFERENCES Wymiary
↪  (Nr_wymiarow)
/

ALTER TABLE Karmy ADD CONSTRAINT Producenci_karmy FOREIGN KEY (Nr_producenta) REFERENCES Producenci
↪  (Nr_producenta)
```

```
/

ALTER TABLE Lekarstwa ADD CONSTRAINT Producenci_lekarstwa FOREIGN KEY (Nr_producenta) REFERENCES
↪  Producenci (Nr_producenta)
/

ALTER TABLE Adresy ADD CONSTRAINT Poczty_adresy FOREIGN KEY (Nr_poczty) REFERENCES Poczty (Nr_poczty)
/

ALTER TABLE Placowki ADD CONSTRAINT Adresy_placowki FOREIGN KEY (Nr_adresu) REFERENCES Adresy
↪  (Nr_adresu)
/

ALTER TABLE Zwierzeta ADD CONSTRAINT Zwierzeta_gatunki_rasy FOREIGN KEY (Nr_gatunek_rasa) REFERENCES
↪  Gatunki_rasy (Nr_gatunek_rasa)
/

ALTER TABLE Adoptujacy ADD CONSTRAINT Adoptujacy_adresy FOREIGN KEY (Nr_adresu) REFERENCES Adresy
↪  (Nr_adresu)
/

ALTER TABLE Pracownicy ADD CONSTRAINT Pracownicy_adresy FOREIGN KEY (Nr_adresu) REFERENCES Adresy
↪  (Nr_adresu)
/

ALTER TABLE Zajmowania_stanowisk ADD CONSTRAINT Stanowiska_zajmowanie_stanowisk FOREIGN KEY
↪  (Nr_stanowiska) REFERENCES Stanowiska (Nr_stanowiska)
/

ALTER TABLE Zajmowania_stanowisk ADD CONSTRAINT Pracownicy_zajmowania_stanowisk FOREIGN KEY
↪  (Nr_pracownika) REFERENCES Pracownicy (Nr_pracownika)
/

ALTER TABLE Typy_wyposazenia ADD CONSTRAINT Typy_wyposazenia_producenci FOREIGN KEY (Nr_producenta)
↪  REFERENCES Producenci (Nr_producenta)
/

ALTER TABLE Wyposazenia_egzemplarze ADD CONSTRAINT Typy_wyposazenia_egzemplarze FOREIGN KEY
↪  (Nr_typu_wyposazenia) REFERENCES Typy_wyposazenia (Nr_typu_wyposazenia)
```

5.3. Przykłady zapytań i poleceń odnoszących się do bazy danych

5.3.1. Uzupełnianie wartości w bazie

```
INSERT ALL
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('23-456', 'Wrocław')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('56-789', 'Szczecin')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('78-901', 'Katowice')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('34-567', 'Gdynia')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('90-123', 'Białystok')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('45-678', 'Bydgoszcz')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('67-890', 'Częstochowa')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('23-456', 'Radom')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('56-789', 'Sosnowiec')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('78-901', 'Toruń')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('34-567', 'Kielce')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('90-123', 'Rzeszów')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('45-678', 'Olsztyn')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('67-890', 'Bielsko-Biała')
    INTO Poczty(Kod_pocztowy, Miasto) VALUES('23-456', 'Opole')
SELECT 1 FROM DUAL;
```

```
/
INSERT ALL
    INTO Stanowiska(Nazwa) VALUES('Opiekun zwierzat')
    INTO Stanowiska(Nazwa) VALUES('Pracownik medyczny')
    INTO Stanowiska(Nazwa) VALUES('Kierownik')
    INTO Stanowiska(Nazwa) VALUES('Pracownik biurowy')
    INTO Stanowiska(Nazwa) VALUES('Sprzatajacy')
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Brzozowa', '2', '3', 1,
    ↪  'Wrocław')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Bukowa', '4', '5', 2,
    ↪  'Szczecin')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Jasna', '6', NULL, 3,
    ↪  'Katowice')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Cicha', '7', '9', 4,
    ↪  'Gdynia')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Szkolna', '10', '15', 5,
    ↪  'Białystok')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Leśna', '11', NULL, 6,
    ↪  'Bydgoszcz')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Akacjowa', '13', '17', 7,
    ↪  'Częstochowa')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Szkolna', '14', NULL, 8,
    ↪  'Radom')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Polna', '15', '19', 9,
    ↪  'Sosnowiec')
    INTO Adresy(Ulica, Nr_budynku, Nr_mieszkania, Nr_poczty, Miasto) VALUES('Sosnowa', '16', NULL, 10,
    ↪  'Toruń')
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Placowki(Nazwa_placowki, Nr_telefonu, Adres_email, Maks_liczba_zwierzat, Nr_adresu)
    ↪  VALUES('Słoneczko', '222379065', 'sloneczko@bada.pl', 57, 5)
    INTO Placowki(Nazwa_placowki, Nr_telefonu, Adres_email, Maks_liczba_zwierzat, Nr_adresu)
    ↪  VALUES('Promyk', '167350872', 'promyk@bada.pl', 112, 10)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Wymiary(Szerokosc, Wysokosc, Glebokosc) VALUES(500, 200, 300)
    INTO Wymiary(Szerokosc, Wysokosc, Glebokosc) VALUES(700, 180, 400)
    INTO Wymiary(Szerokosc, Wysokosc, Glebokosc) VALUES(300, 200, 200)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Boksy(Maks_liczba_zwierzat, Nr_placowki, Nr_wymiarow) VALUES(3, 1, 3)
    INTO Boksy(Maks_liczba_zwierzat, Nr_placowki, Nr_wymiarow) VALUES(4, 2, 2)
    INTO Boksy(Maks_liczba_zwierzat, Nr_placowki, Nr_wymiarow) VALUES(2, 2, 1)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Czasy_otwarcia(Godzina_otwarcia, Godzina_zamkniecia, Dzien) VALUES('10:00', '16:00', 'PON')
    INTO Czasy_otwarcia(Godzina_otwarcia, Godzina_zamkniecia, Dzien) VALUES('12:00', '16:00', 'PT')
    INTO Czasy_otwarcia(Godzina_otwarcia, Godzina_zamkniecia, Dzien) VALUES('12:00', '14:00', 'SOB')
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Otwarcia(Nr_placowki, Nr_czasu_otwarcia) VALUES(1, 1)
    INTO Otwarcia(Nr_placowki, Nr_czasu_otwarcia) VALUES(2, 2)
    INTO Otwarcia(Nr_placowki, Nr_czasu_otwarcia) VALUES(1, 3)
```

```
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Producenci(Nazwa) VALUES('Bayer')
    INTO Producenci(Nazwa) VALUES('Pedigree')
    INTO Producenci(Nazwa) VALUES('Gryzuś')
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Lekarstwa(Nazwa, Dawka, Nr_producenta) VALUES('Bravecto', '2mg', 1)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Typy_wyposazenia(Nazwa_typu, Nr_producenta) VALUES('Smycz', 3)
    INTO Typy_wyposazenia(Nazwa_typu, Nr_producenta, Model) VALUES('Obroza', 3, '12A')
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Karmy(Nazwa, Data_waznosci, Nr_producenta) VALUES('Diet Senior', TO_DATE('2023-12-27',
    ↪  'YYYY-MM-DD'), 2)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Pracownicy(Imie, Nazwisko, Nr_dokumentu, Pesel, Nr_telefonu, Plec, Czy_wolontariusz,
    ↪  Nr_adresu) VALUES('Andrzej', 'Kupiec', 'MGR12345', '03123298566', '509187144', 'M', 'T', 1)
    INTO Pracownicy(Imie, Nazwisko, Nr_dokumentu, Pesel, Nr_telefonu, Plec, Czy_wolontariusz, Pensja,
    ↪  Nr_adresu) VALUES('Kamil', 'Kasprzak', 'LIC98765', '92567398003', '639254087', 'M', 'N',
    ↪  10000, 3)
    INTO Pracownicy(Imie, Nazwisko, Nr_dokumentu, Pesel, Nr_telefonu, Plec, Czy_wolontariusz, Pensja,
    ↪  Adres_email, Nr_adresu) VALUES('Kamil', 'Pilarczyk', 'TWI72545', '72509849805', '964382670',
    ↪  'M', 'N', 5700, 'kasztan@op.pl', 7)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Przypisania_pracownikow(Nr_placowki, Nr_pracownika, Poczatek_przypisania) VALUES(1, 1,
    ↪  TO_DATE('2022-12-27', 'YYYY-MM-DD'))
    INTO Przypisania_pracownikow(Nr_placowki, Nr_pracownika, Poczatek_przypisania) VALUES(2, 2,
    ↪  TO_DATE('2020-06-20', 'YYYY-MM-DD'))
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Zajmowania_stanowisk(Data_rozpoczecia, Nr_pracownika, Nr_stanowiska)
    ↪  VALUES(TO_DATE('2022-12-27', 'YYYY-MM-DD'), 1, 1)
    INTO Zajmowania_stanowisk(Data_rozpoczecia, Nr_pracownika, Nr_stanowiska)
    ↪  VALUES(TO_DATE('2020-06-20', 'YYYY-MM-DD'), 2, 5)
    INTO Zajmowania_stanowisk(Data_rozpoczecia, Nr_pracownika, Nr_stanowiska)
    ↪  VALUES(TO_DATE('2017-11-26', 'YYYY-MM-DD'), 3, 1)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Posiadania_lekarstw(Nr_lekarstwa, Nr_placowki, Ilosc_na_stanie) VALUES(1, 1, 100)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Wyposazenia_egzemplarze(Nr_placowki, Nr_boksu, Nr_typu_wyposazenia) VALUES(1, 1, 1)
    INTO Wyposazenia_egzemplarze(Nr_placowki, Nr_boksu, Nr_typu_wyposazenia) VALUES(1, 1, 2)
    INTO Wyposazenia_egzemplarze(Nr_placowki, Nr_boksu, Nr_typu_wyposazenia) VALUES(2, 2, 2)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Posiadania_karm(Nr_placowki, Nr_karmy, Ilosc_na_stanie) VALUES(1, 1, 30)
```

```
    INTO Posiadania_karm(Nr_placowki, Nr_karmy, Ilosc_na_stanie) VALUES(2, 1, 37)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Gatunki_rasy(Gatunek, Rasa) VALUES('Pies', 'Maltańczyk')
    INTO Gatunki_rasy(Gatunek, Rasa) VALUES('Pies', 'Entlebucher')
    INTO Gatunki_rasy(Gatunek) VALUES('Pies')
    INTO Gatunki_rasy(Gatunek) VALUES('Kot')
    INTO Gatunki_rasy(Gatunek, Rasa) VALUES('Kot', 'Brytyjski')
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Zwierzeta(Wiek, Masa, Data_przyjecia, Czy_szczepiony, Czy_kastrowany, Plec, Nr_gatunek_rasa,
    ↪  Imie) VALUES(3, 4, TO_DATE('2021-07-10', 'YYYY-MM-DD'), 'T', 'T', 'Samica', 1, 'Bella')
    INTO Zwierzeta(Wiek, Masa, Data_przyjecia, Czy_szczepiony, Czy_kastrowany, Plec, Nr_gatunek_rasa,
    ↪  Imie) VALUES(7, 23, TO_DATE('2016-02-19', 'YYYY-MM-DD'), 'T', 'T', 'Samica', 2, 'Luna')
    INTO Zwierzeta(Wiek, Masa, Data_przyjecia, Czy_szczepiony, Czy_kastrowany, Plec, Nr_gatunek_rasa,
    ↪  Imie) VALUES(5, 5, TO_DATE('2020-04-10', 'YYYY-MM-DD'), 'T', 'N', 'Samiec', 4, 'Tobiasz')
    INTO Zwierzeta(Wiek, Masa, Data_przyjecia, Czy_szczepiony, Czy_kastrowany, Plec, Nr_gatunek_rasa,
    ↪  Imie) VALUES(10, 40, TO_DATE('2017-07-29', 'YYYY-MM-DD'), 'T', 'T', 'Samiec', 3, 'Hektor')
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Zamieszkiwania(Nr_boksu, Nr_zwierzecia, Data_rozpoczecia) VALUES(2, 1, TO_DATE('2021-07-10',
    ↪  'YYYY-MM-DD'))
    INTO Zamieszkiwania(Nr_boksu, Nr_zwierzecia, Data_rozpoczecia) VALUES(3, 2, TO_DATE('2019-03-11',
    ↪  'YYYY-MM-DD'))
    INTO Zamieszkiwania(Nr_boksu, Nr_zwierzecia, Data_rozpoczecia) VALUES(2, 3, TO_DATE('2021-04-10',
    ↪  'YYYY-MM-DD'))
    INTO Zamieszkiwania(Nr_boksu, Nr_zwierzecia, Data_rozpoczecia) VALUES(3, 4, TO_DATE('2017-07-29',
    ↪  'YYYY-MM-DD'))
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Karmienia(Nr_karmy, Data_rozpoczecia, Nr_zwierzecia, Czestotliwosc) VALUES(1,
    ↪  TO_DATE('2022-04-10', 'YYYY-MM-DD'), 1, 3)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Przyjmowania_lekarstw(Nr_lekarstwa, Poczatek_przyjmowania, Nr_zwierzecia) VALUES(1,
    ↪  TO_DATE('2023-10-19', 'YYYY-MM-DD'), 1)
    INTO Przyjmowania_lekarstw(Nr_lekarstwa, Poczatek_przyjmowania, Nr_zwierzecia,
    ↪  Koniec_przyjmowania) VALUES(1, TO_DATE('2023-07-03', 'YYYY-MM-DD'), 4, TO_DATE('2023-10-03'))
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Opiekowania(Nr_pracownika, Poczatek_opiekowania, Nr_zwierzecia) VALUES(1,
    ↪  TO_DATE('2023-11-07', 'YYYY-MM-DD'), 4)
SELECT 1 FROM DUAL;
/
INSERT ALL
    INTO Adoptujacy(Imie, Nazwisko, Nr_dokumentu, Pesel, Nr_telefonu, Plec, Nr_adresu)
    ↪  VALUES('Maciej', 'Lipski', 'INZ12345', '03675008672', '733804956', 'M', 9)
SELECT 1 FROM DUAL;
```

5.3.2. Inne zapytania

Wypisanie pracowników mieszkających we Wrocławiu

```
SELECT p.Imie, p.Nazwisko FROM Pracownicy p, Adresy a
WHERE p.Nr_adresu = a.Nr_adresu AND a.Miasto = 'Wrocław';
```

Wypisanie pracowników zarabiających pensję i stosunku ich płacy do średniej w procentach

```
SELECT Nazwisko, Pensja, ROUND(Pensja/(SELECT AVG(Pensja) FROM Pracownicy), 2)*100 AS
↪  Procent_sredniej_pensji FROM Pracownicy WHERE Pensja IS NOT NULL
```

Adopcja zwierzęcia

```
INSERT INTO Adoptujacy(Imie, Nazwisko, Nr_dokumentu, Pesel, Nr_telefonu, Plec, Nr_adresu)
↪  VALUES('Anna', 'Kowalska', 'ABC72345', '92675008672', '733034956', 'K', 4)

UPDATE Zwierzeta SET Nr_adoptujacego = (SELECT Nr_adoptujacego FROM Adoptujacy WHERE Imie = 'Anna' AND
↪  Nazwisko = 'Kowalska') WHERE Imie = 'Tobiasz'

UPDATE Zamieszkiwania SET Data_zakonczenia = (SELECT SYSDATE FROM DUAL) WHERE Nr_zwierzecia = (SELECT
↪  Nr_zwierzecia FROM Zwierzeta WHERE Imie = 'Tobiasz') AND Data_zakonczenia IS NULL
```

Wypisanie imion zwierząt i lekarstw które aktualnie przyjmują

```
SELECT z.Imie, l.Nazwa FROM Zwierzeta z, Przyjmowania_lekarstw p, Lekarstwa l WHERE p.Nr_zwierzecia =
↪  z.Nr_zwierzecia AND l.Nr_lekarstwa = p.Nr_lekarstwa AND p.Koniec_przyjmowania IS NULL
```

Wypisanie placówek otwartych w poniedziałek wraz z ich godzinami otwarcia

```
SELECT p.Nazwa_placowki, cz.Godzina_otwarcia, cz.Godzina_zamkniecia FROM Placowki p JOIN Otwarcia o ON
↪  p.Nr_placowki = o.Nr_placowki JOIN Czasy_otwarcia cz ON cz.Nr_czasu_otwarcia = o.Nr_czasu_otwarcia
↪  AND Dzien = 'PON'
```