**Bazy danych**

**Dokumentacja projektu**

**Temat projektu:**

System mający za zadanie przechowywać informacje o obiektach należących do sieci basenów oraz oferowanych przez nią usługach. Ma to ułatwić zarządzanie pracownikami (możliwość zmiany kadry i wynagrodzeń), obiektami (przeglądy basenów), transakcjami dokonywanymi przez klientów jedynie „na miejscu” (np. wykup karnetów, wejść do sauny) oraz transakcjami, które można dokonywać poprzez aplikację dostępną dla klienta (rezerwacja torów oraz lekcji pływackich). Pracownicy sieci basenów przy pomocy aplikacji (dla pracowników) mogą wykonywać operacje zmieniające stan bazy oraz wyświetlać odpowiednie informacje w postaci raportów właściwych dla obejmowanego przez nich stanowiska. Dostęp do bazy danych z poziomu pracownika jest chroniony loginem i hasłem. System polega na dynamicznie zmieniających się danych, które definiują kształt bazy.

Interfejs użytkownika będzie zrealizowany jako aplikacja okienkowa.

**Punkty kontrolne:**

|  |  |  |  |
| --- | --- | --- | --- |
| **LP.** | **Opis** | **Data oddania** | **Realizacja** |
| **1.** | Stworzenie modelu ER (z wykorzystaniem notacji Barkera) | Do **20.11** | Oracle Database |
| **2.** | Zaprojektowanie relacyjnego modelu danych | Do **27.11** | Oracle Database |
| **3.** | Zaprojektowanie fizycznego modelu danych | Do **11.12** |  |
| **4.** | Implementacja bazy danych | Do **11.12** | Oracle Database |
| **5.** | Implementacja aplikacji dostępowej | Do **08.01** | JDBC, JavaFX |
| **6.** | Implementacja aplikacji raportowej | Do **08.01** | JDBC, JavaFX |

Komentarz: każdy z punktów kontrolnych ma określony deadline i wykorzystywaną do jego realizacji technologię. Zamierzamy sukcesywnie oddawać kolejne etapy, w niektórych przypadkach planujemy prezentację kilku punktów w ramach jednego spotkania. Zakładamy realizację wszystkich wymagań maksymalnie do rozpoczęcia ferii świątecznych.

**Spis treści:**

[Wymagania funkcjonalne: 3](#_Toc531347584)

[Wymagania niefunkcjonalne: 4](#_Toc531347585)

[Słownik pojęć: 4](#_Toc531347586)

[Opis/specyfikacja: 6](#_Toc531347587)

[Technologie: 7](#_Toc531347588)

[Model konceptualny 8](#_Toc531347589)

[Model logiczny 9](#_Toc531347590)

[Model relacyjny 11](#_Toc531347591)

# Wymagania funkcjonalne:

|  |  |  |
| --- | --- | --- |
| **ID** | **Tytuł** | **Opis** |
| **1** | Zatrudnianie pracowników | **HR** wprowadza dane kontaktowe, stanowisko i wynagrodzenie nowo zatrudnionego pracownika. |
| **1.1** | Zwalnianie pracowników | **HR** archiwizuje wszystkie dane pracownika na kolejne 50 lat, zgodnie z kodeksem pracy, i usuwa go z ewidencji pracowniczej. |
| **1.2** | Edycja pracownika | **HR** edytuje dane pracownika, a w szczególności: |
| **1.2.1** | Zmiana stanowiska | Zmienia stanowisko pracownika. |
| **1.2.2** | Zmiana wynagrodzenia | Zmienia wynagrodzenie pracownika. |
| **1.3** | Sprawdzanie listy pracowników | **HR**, **audytor**, **kierownik** i **właściciel** mogą sprawdzić listę pracowników. |
| **2** | Dodanie klienta | **Kasjer** automatycznie dodaje nowego klienta, gdy nie figuruje w bazie danych, a takowy zakupił karnet lub zarezerwował tor. |
| **2.1** | Usuwanie klienta | Dane klientów kasowane są po określonym czasie po wygaśnięciu karnetu lub po dacie rezerwacji toru. |
| **2.2** | Edycja danych klienta | Dane klientów na życzenie **klienta** mogą być edytowane przez **kasjera**. Tyczy się to danych kontaktowych. |
| **2.3** | Sprawdzanie listy klientów | **Ekspert ds. marketingu,** a także kasjer,mogą przeglądać listę klientów w celu wykonania pewnych czynności. |
| **3** | Dodanie rezerwacji torów | **Kasjer** na życzenie **klienta** dodaje rezerwację toru do bazy danych. **Klient** może też dokonać rezerwacji samodzielnie przy pomocy aplikacji. |
| **3.1** | Usuwanie rezerwacji torów | **Kasjer** na życzenie **klienta** usuwa rezerwację toru z bazy danych. |
| **3.2** | Sprawdzanie rezerwacji torów | **Kasjer,** w celu wykonania akcji związanej z rezerwacjami torów, musi mieć dostęp do podglądu tychże rezerwacji. |
| **3.3** | Zmiana statusu rezerwacji | **Kasjer** może zmienić status rezerwacji toru (na wykorzystaną) po zgłoszeniu się **klienta**, co spowoduje też umieszczenie jej w zbiorze transakcji. |
| **4** | Dodawanie przeglądów | **Konserwator** może dodawać opis przeprowadzonych przeglądów. |
| **4.1** | Usuwanie przeglądów | **Konserwator** może usuwać przeprowadzone przeglądy ze względu na błędy, które w nich mogły wyniknąć. |
| **4.2** | Sprawdzanie przeglądów | **Konserwator** i **audytor** mogą sprawdzać przeprowadzone przeglądy. |
| **4.3** | Sprawdzanie daty przeglądu | **Konserwator** i **kierownik** mogą sprawdzić datę kolejnego przeglądu technicznego basenu. W przypadku przekroczenia daty wyświetlany jest odpowiedni komunikat. |
| **5** | Dodanie transakcji | **Kasjer** może na życzenie **klienta** dodać nową transakcję. |
| **5.1** | Sprawdzanie transakcji | **Audytor**, **ekspert ds. marketingu** oraz **kierownik** mogą sprawdzić listę przeprowadzonych transakcji. |
| **5.2** | Wyświetlanie raportów dot. transakcji | **Ekspert ds. marketingu** może wyświetlić informacje na temat klientów dokonujących najwięcej transakcji, ilości wykorzystanych rezerwacji w danym przedziale czasowym oraz rodzajów transakcji i ich wartości w danym przedziale czasowym. |
| **6** | Sprawdzanie przychodów | **Właściciel** może sprawdzić jakie przychody osiągnęła jego sieć basenów. |
| **6.1** | Sprawdzanie kosztów | **Właściciel** może sprawdzić jakie koszty poniosła jego sieć basenów. |
| **7** | Dodanie lekcji pływania | **Kasjer** na życzenie **klienta** dodaje rezerwację lekcji pływania do bazy danych. **Klient** może też dokonać rezerwacji samodzielnie przy pomocy aplikacji. |
| **7.1** | Usuwanie lekcji pływania | **Kasjer** na życzenie **klienta** usuwa rezerwację lekcji pływania z bazy danych. |
| **7.2** | Wyświetlenie informacji o lekcji pływania | **Kasjer** oraz **klient** mogą zobaczyć ile osób jest zapisanych na daną lekcję oraz kto ją prowadzi. |

# Wymagania niefunkcjonalne:

|  |  |  |
| --- | --- | --- |
| **ID** | **Wymaganie** | **Opis** |
| **N1** | Łatwość obsługi | Użytkownik systemu jest w stanie biegle posługiwać się systemem po 2h szkoleniu. |
| **N2** | Bezpieczeństwo | Zastosowanie podczas logowania kodowania haseł. Hasła w bazie danych przechowywane w formie zakodowanej wraz z solą [salt]. |

# Słownik pojęć:

|  |  |
| --- | --- |
| **Obiekt** | **Opis** |
| Basen | Posiada swój **identyfikator**, **nazwę** i **lokalizację.** Każdy z obiektów przechodzi **przeglądy**, ma **pracowników**, zbiór oferowanych **usług** oraz określoną liczbę **torów pływackich**, które mogą być rezerwowane. |
| Przegląd | Przeprowadzany przez **konserwatora** przypisanego do danego basenu, jest cyklicznym zdarzeniem mającym na celu identyfikację i naprawę ewentualnych usterek. Przeglądy będą mieć informację o **dacie** przeprowadzenia przeglądu oraz o **numerze identyfikacyjnym** basenu, wraz z **opisem** wykrytej usterki bądź jej braku. Każdy z basenów musi odbywać przegląd co dwa lata. Baza danych pozwoli konserwatorowi sprawdzić termin następnego przeglądu. |
| Rezerwacja torów | Zawierać będą **datę**, **godzinę**, **numer toru** i **status**. Atrybut „status” będzie miał na celu odnotowanie faktu, czy rezerwacja została wykorzystana. Wprowadzenie wyżej wspomnianego atrybutu pozwoli zbierać dane do statystyk użyteczne dla eksperta ds. marketingu. Będzie ona powiązana z **klientem**, który dokonał rezerwacji. |
| Lekcja pływania | Zawierać będą **datę**, **godzinę**, **numer ratownika** oraz **liczbę zapisanych osób**. Maksymalna liczba osób mogąca uczestniczyć w lekcji to 6. **Klient**, aby móc się zapisać na lekcję pływania, musi zapłacić za nią z góry (w kasie lub poprzez aplikację). |
| Pracownik | Będzie miał unikalny **identyfikator**, **imię**, **nazwisko**, **stanowisko**, **wynagrodzenie**, ewentualne **dodatki do pensji** oraz **login** i **hasło** do bazy danych (poszczególni pracownicy będą mieć zapewniony dostęp do typów danych powiązanych wyłącznie z wykonywaną przez nich pracą). |
| Usługa | Jest elementem ofert basenu. Klient opłacając daną usługę, uzyskuje dostęp do odpowiadającej jej atrakcji. Każda usługa będzie miała swoją **cenę.** Ponadto usługi dostępne na poszczególnych basenach są od siebie niezależne (przykładowo: karnet na basenie A jest inną usługą, niż karnet na basenie B). Opłacając pewien (wybrany) zestaw usług, klient będzie dokonywał transakcji – bytu będącego odwzorowaniem dokonania zakupu przez klienta. Usługi dzielą się na **lokalne i ogólne**. Lokalne to takie, które kupowane są „na miejscu”, a ogólne mogą być też kupowane poprzez aplikację dla klienta. |
| Transakcja | Będzie miała unikalny **identyfikator**, **datę** oraz **kwotę transakcji.** Aby powiązać ze sobą usługi i transakcje, wprowadzony będzie **koszyk,** który będzie zawierać informacje na temat tego, ile usług danego typu wchodziło w skład konkretnej transakcji. **Klient** może być zapisany do bazy danych w przypadku **rezerwacji toru** lub wykupienia **lekcji pływania**. |
| Klient | Osoba, która dokonuje transakcje „na miejscu” lub **rezerwuje tor** albo wykupuje **lekcję pływania** poprzez aplikację**.** Nie wszystkie osoby, które dokonują transakcji, znajdą się w bazie danych – personalia są potrzebne tylko w przypadku **rezerwowania toru** lub wykupu **lekcji pływania**. Klient będzie mieć unikalny **identyfikator**, **imię**, **nazwisko**, **numer telefonu** oraz **adres e-mail**. |
| Ekspert ds. marketingu | Pracownik, który będzie mógł sprawdzić, jakie usługi nabywał dany klient w danym przedziale czasowym, celem np. zaoferowania spersonalizowanej promocji. |

# Opis/specyfikacja:

System wraz z bazą danych opiera się na podziale ról użytkowników. Wyróżniamy kilka typów użytkowników:

* **audytor** – może **sprawdzać listę pracowników** (w tym ich wynagrodzenie), **przeglądy** oraz **transakcje** dokonywane przez poszczególne baseny
* **ekspert ds. marketingu** – może **przeglądać listę transakcji** oraz **klientów** bez możliwości modyfikowania; predefiniowane kwerendy: wyświetlenie listy dziesięciu najcenniejszych klientów, wyświetlenie zestawienia rezerwacji na pewien okres, wyświetlenie zestawienia transakcji i ich wartości na pewien okres, wyświetlenie procentu wykorzystanych rezerwacji
* **HR** – może **zwalniać/zatrudniać pracowników** oraz **zmieniać** ich **stanowiska**   
  i **wynagrodzenie**; predefiniowane kwerendy: zmiana pensji o pewien procent dla danego stanowiska
* **kasjer** – może **dodawać/usuwać rezerwacje torów** oraz **przeglądać je**; predefiniowane kwerendy: zaznaczenie wykorzystania danej rezerwacji, sprawdzenie dostępności poszczególnych torów
* **kierownik** – może **sprawdzać daty przeglądów**, **dane pracowników** i **transakcje** (wszystko z zarządzanego obiektu); predefiniowane kwerendy: wyświetlenie zestawienia zysków na miesiąc, wyświetlenie zestawienia wypłacanych pensji na dany miesiąc, sprawdzenie, ile dni zostało do następnego przeglądu
* **konserwator** – może **dodawać/usuwać przeglądy** z bazy danych oraz je **wyszukiwać**; może **sprawdzać termin następnego przeglądu** (data oraz ile dni zostało)
* **właściciel** – może **sprawdzać przychody** i **koszty** z poszczególnych obiektów oraz **przeglądać listę pracowników** (bez możliwości edytowania – od tego jest HR)

Przy czym dwa stanowiska pracowników nie mają dostępu do bazy danych:

* **ratownik** – pracownik ten nie potrzebuje znać żadnych danych wewnątrz bazy
* **sprzątaczka** – pracownik ten nie ma dostępu do systemu ze względu na fizyczny charakter jego pracy

Ostatnim aktorem bazy danych jest **klient**. Może on wpływać na bazę danych przy pomocy aplikacji (rezerwując tory lub wykupując lekcje pływania) lub poprzez kontakt z pracownikami sieci basenów (np. z **kasjerką**).

Baza danych składa się z obiektów i ich relacji (opisy tych podstawowych przedstawione w słowniku).

**Obiekty:**

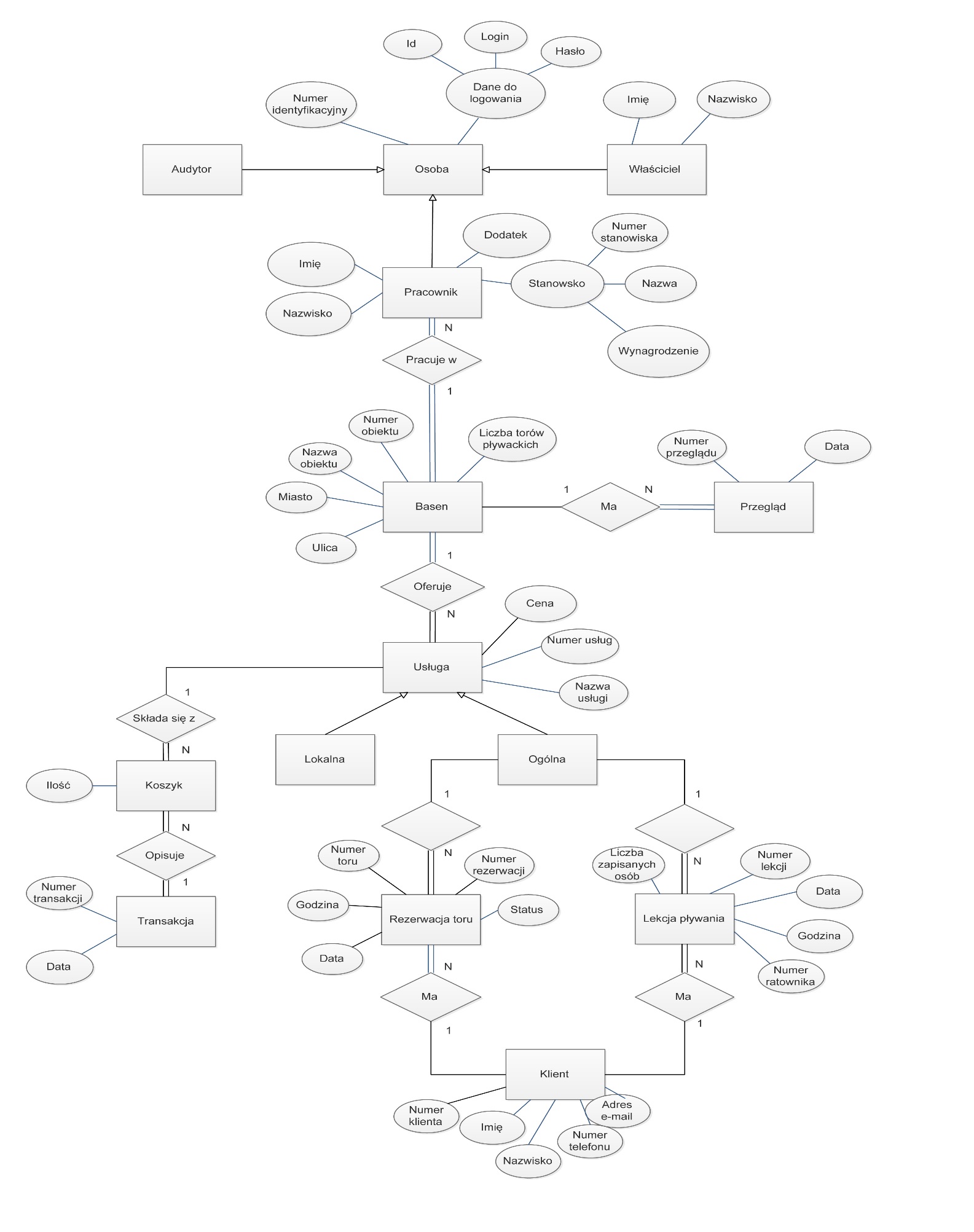
* basen
* klient
* koszyk
* osoba (podtypy: audytor, pracownik, właściciel)
* przegląd
* rezerwacja toru
* lekcja pływania
* transakcja
* usługa (podtypy: lokalna, ogólna)

# Technologie:

* Oracle Database – SZBD
* JDBC – komunikacja z bazą danych z poziomu aplikacji
* JavaFX – interfejs graficzny (okienkowy)

Komentarz: zdecydowaliśmy się na wybór Oracle Database jako używanego przez nas SZBD, ponieważ znamy ten system z przedmiotu Bazy danych 1 realizowanego w poprzednim semestrze.

# Model konceptualny



# Model logiczny

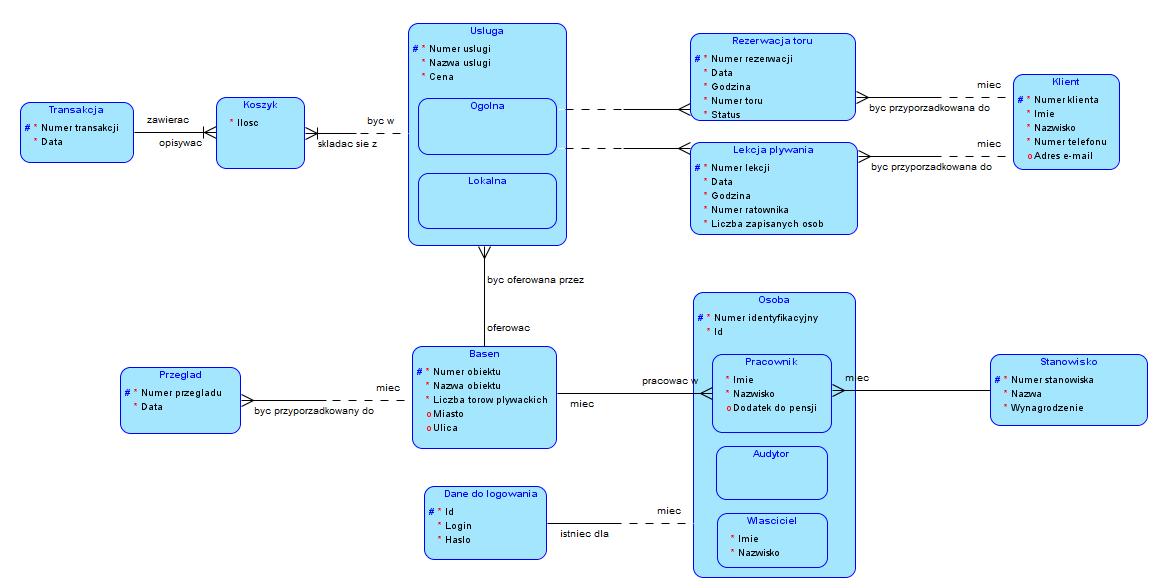
Podstawową encją w tym modelu jest **Basen**. Jest on połączony relacjami z aż czterema innymi encjami: **Usługa**, **Pracownik**, **Przegląd**, **Rezerwacja toru**. Zgodnie ze schematem, **Basen** musi mieć **Pracowników**i oferować **Usługi**. **Przeglądy** i **Rezerwacje toru** są natomiast opcjonalne.

Encja **Osoba** składa się z trzech podtypów: **Audytor**, **Pracownik**, **Właściciel**. Każda z osób może mieć **Dane do logowania**. Wyróżnionym podtypem jest **Pracownik**, który jest połączony relacją ze **Stanowiskiem** i **Basenem**.

W przypadku encji **Klient**, jest ona połączona relacjami z **Rezerwacją toru** i **Transakcją**, przy czym   
na szczególną uwagę zasługuje drugi ze związków – jest to relacja 1:N, obligatoryjna od strony **Transakcji**. Otóż zgodnie z dokumentacją, **Klient** może dokonać wielu **Transakcji**, ale nie każda **Transakcja** dotyczy **Klienta**. Wynika to z przyjętego założenia, że w bazie zapisywane są dane tylko niektórych **Klientów**.   
Stąd też opcjonalność związku od strony **Transakcji**.

Encja **Koszyk** jest encją słabą, połączoną relacjami identyfikującymi z encjami **Usługa** i **Transakcja**.   
Jej istnienie jest podyktowane potrzebą związania encji **Usługa** i **Transakcja** – istnieją atrybuty,   
które identyfikują łączącą je relację i muszą być gdzieś zawarte.

Zmianą, jaka zaszła w modelu logicznym względem modelu koncepcyjnego jest przede wszystkim wyróżnienie nowych encji: **Stanowisko** i **Dane do logowania**. Poszczególne **Stanowiska** mogą się powtarzać wśród **Pracowników**, więc rozsądniej jest utworzyć osobną encję, niż przechowywać (i powielać)   
te informacje w atrybutach **Pracownika**. Ponadto wielu **Pracowników** nie będzie miało dostępu do bazy danych – dlatego też utworzona została nowa encja, przechowującą **Dane do logowania** tylko tych **Osób**, które dostęp do systemu posiadają.



# Model relacyjny

Model relacyjny jest bardzo podobny do modelu er z poprzedniego etapu projektu, gdyż wszystkie relacje są odzwierciedleniem encji z modelu logicznego, a atrybuty relacji są odzwierciedleniem atrybutów encji. W modelu logicznym nie było związków typu M:N, więc nie było potrzeby tworzenia żadnych dodatkowych relacji. Jedynie encja „osoba” nie została transformowana na relację, gdyż z założenia miało nie być instancji tej encji nadrzędnej.

Związki jednoznaczne (1:N) zostały transformowane do klucza obcego w tabeli po stronie „wiele”. Związek wyłączny jedno-jednoznaczny (pomiędzy podtypami encji „Osoba”, a encją „Dane do logowania”) został transformowany do klucza obcego w tabeli „Dane do logowania”.

Relacje (spis kluczy głównych i obcych)

|  |  |  |
| --- | --- | --- |
| **Nazwa relacji** | **Klucz główny** | **Klucz obcy** |
| Basen | Numer\_obiektu |  |
| Pracownik | Numer\_identyfikacyjny | Basen\_Numer\_obiektu |
| Stanowisko | Numer\_stanowiska |  |
| Audytor | Numer\_identyfikacyjny |  |
| Wlasciciel | Numer\_identyfikacyjny |  |
| Przeglad | Numer\_przegladu | Basen\_Numer\_obiektu; Stanowisko\_Numer\_stanowiska |
| Usluga | Numer\_uslugi | Basen\_Numer\_obiektu |
| Koszyk | Usluga\_Numer\_uslugi, Transakcja\_Numer\_transakcji (klucz kompozytowy) | Usluga\_Numer\_uslugi; Transakcja\_Numer\_transakcji |
| Transakcja | Numer\_transakcji | Klient\_Numer\_klienta |
| Klient | Numer\_klienta |  |
| Rezerwacja toru | Numer\_rezerwacji | Basen\_Numer\_obiektu; Klient\_Numer\_klienta |
| Dane do logowania | Id | Numer\_identyfikacyjny (w tym indeks założony na tę kolumnę) |

