**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.

**Spis treści:**

[Wymagania funkcjonalne: 3](#_Toc532339605)

[Wymagania niefunkcjonalne: 4](#_Toc532339606)

[Słownik pojęć: 5](#_Toc532339607)

[Opis/specyfikacja: 7](#_Toc532339608)

[Technologie: 8](#_Toc532339609)

[Model konceptualny 9](#_Toc532339610)

[Model logiczny 11](#_Toc532339611)

[Model relacyjny 13](#_Toc532339612)

[Spis relacji, kluczy głównych i kluczy obcych 15](#_Toc532339613)

[Model fizyczny 16](#_Toc532339614)

# 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** dodaje nowego klienta, gdy nie figuruje w bazie danych, a takowy zakupił karnet lub zarezerwował tor. |
| **2.1** | Usuwanie klienta | Dane klientów na życzenie **klienta** mogą być usunięte przez **kasjera**. |
| **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.  **Klient** również może to zrobić za pomocą aplikacji. |
| **3.2** | Sprawdzanie rezerwacji torów | **Kasjer** oraz **Klient** 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 dodać kolejny przegląd. |
| **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** dodaje nową transakcję. Lekcje pływania (po ich opłaceniu) oraz rezerwacje (po wykorzystaniu) są dodawane automatycznie. |
| **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. **Klient** również może to zrobić za pomocą aplikacji. |
| **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 szyfrowania haseł podczas logowania. 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. Każdy z basenów musi odbywać przegląd co dwa lata. Baza danych pozwoli konserwatorowi sprawdzić termin następnego przeglądu i wprowadzić datę kolejnego 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. Ponadto zmiana statusu **rezerwacji** na „wykorzystaną” spowoduje powstanie nowej **transakcji** jej odpowiadającej (**rezerwacja** jest opłacana jedynie w przypadku jej wykorzystania). **Rezerwacja** będzie powiązana z **klientem**, który jej dokonał. **Rezerwacja** może być dokonana maksymalnie z tygodniowym wyprzedzeniem. |
| 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). W ciągu dnia na **basenie** mogą odbyć się maksymalnie 3 **lekcje**, które poprowadzi jeden, losowo wybrany ratownik. Na **lekcję** można się zapisać maksymalnie z tygodniowym wyprzedzeniem. |
| Pracownik | Będzie miał unikalny **identyfikator**, **imię**, **nazwisko**, **stanowisko**, **wynagrodzenie**, **datę zatrudnienia, datę zwolnienia** (kolumna pusta dopóki pracownik nie zostanie zwolniony)**,** 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). W przypadku zwolnienia danego pracownika, kolumna z datą zwolnienia zostaje wypełniona, a dane zamazane (poza numerem identyfikacyjnym). |
| 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. |
| Koszyk | Pokazuje ile **usług** danego typu wchodzi w skład danej **transakcji**. |
| 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**. |
| Osoba | **Właściciel, audytor** lub **pracownik**. |
| 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**, **adres e-mail** oraz login i hasło (w osobnej tabeli) przydatne do zalogowania się do aplikacji. |
| Dane do logowania | Login oraz hasło przyporządkowane do **osoby** lub **klienta.** |
| 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 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**;
* **kasjer** – może **dodawać/usuwać rezerwacje torów** oraz **przeglądać je,** może też **dodawać/usuwać/modyfikować informacje na temat klientów oraz dokonywanych przez nich transakcji**; 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 mieć dostępu do 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
* dane do logowania (oddzielne dla pracowników i klientów)
* koszyk
* osoba (podtypy: audytor, pracownik, właściciel)
* przegląd
* rezerwacja toru
* lekcja pływania
* transakcja
* usługa (podtypy: lokalna, ogólna)

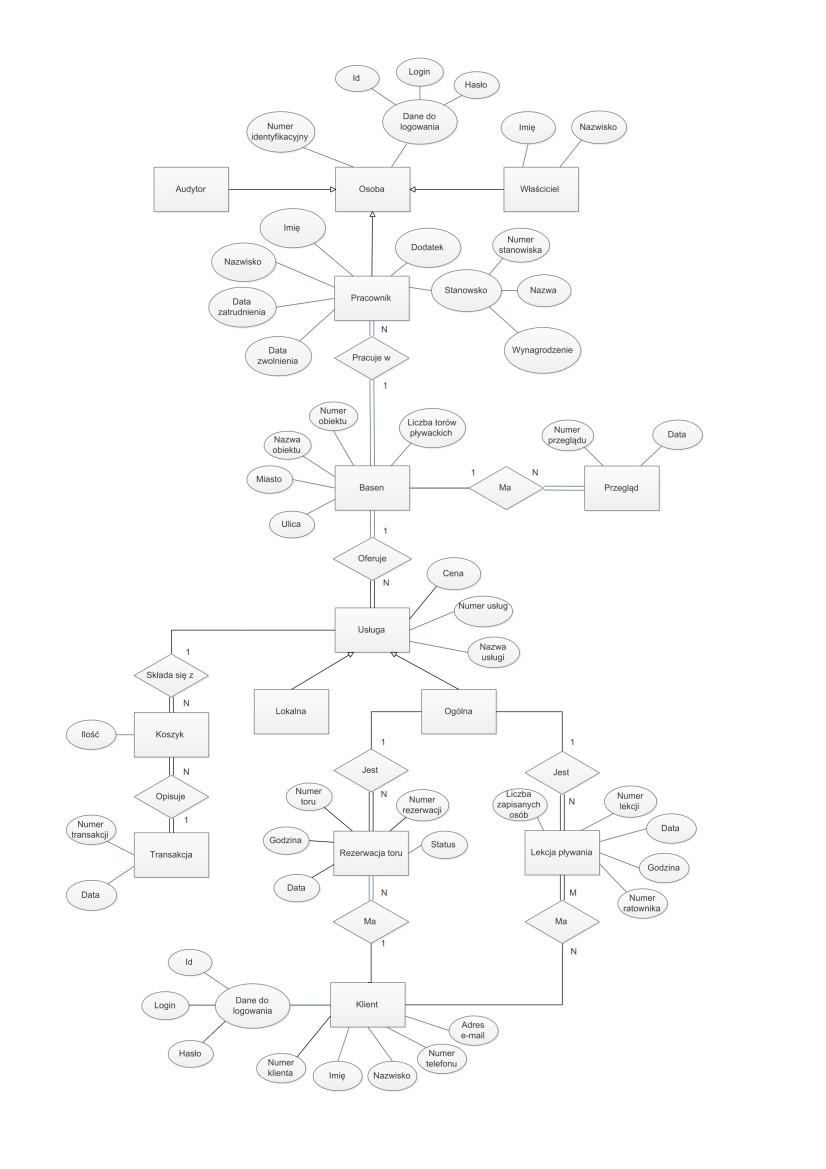
# Technologie:

* 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 konceptualny jest reprezentacją graficzną omówionych wyżej założeń: znalazły się w nim wszystkie wspomniane obiekty wraz z odpowiadającymi im atrybutami.



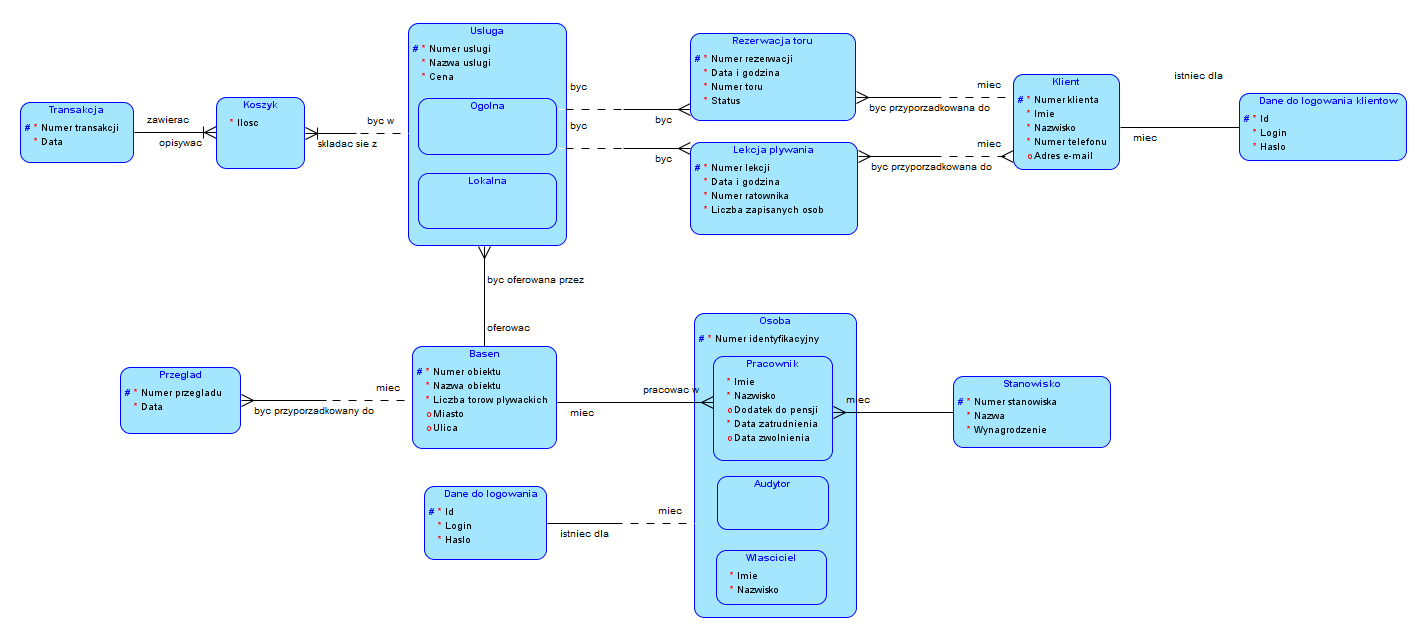
# Model logiczny

Podstawową encją w tym modelu jest **Basen**. Jest on połączony relacjami z trzema innymi encjami: **Usługa**, **Pracownik**, **Przegląd**. Zgodnie ze schematem, **Basen** musi mieć **Pracowników** i oferować **Usługi**. **Przeglądy** są natomiast opcjonalne (nowo utworzone obiekty mogą ich nie mieć).

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**.

Encja **Usługa** składa się natomiast z dwóch podtypów: **Lokalna** i **Ogólna**. W tym przypadku na szczególną uwagę zasługuje podtyp **Ogólna** połączony relacją z **Rezerwacją toru** i **Lekcją pływania**, gdzie są przechowywane szczegółowe informacje na temat tych rodzajówusług. **Rezerwacja toru** i **Lekcja pływania** są z kolei połączone związkiem z **Klientem**, jako że są to usługi personalizowane. Warto zwrócić uwagę na związek M:N między **Klientem** a **Lekcją pływania**. Jest to spowodowane tym, że w **Lekcji pływania** może brać udział kilku **Klientów**, a **Klient** może zapisać się na wiele **Lekcji pływania**.

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.

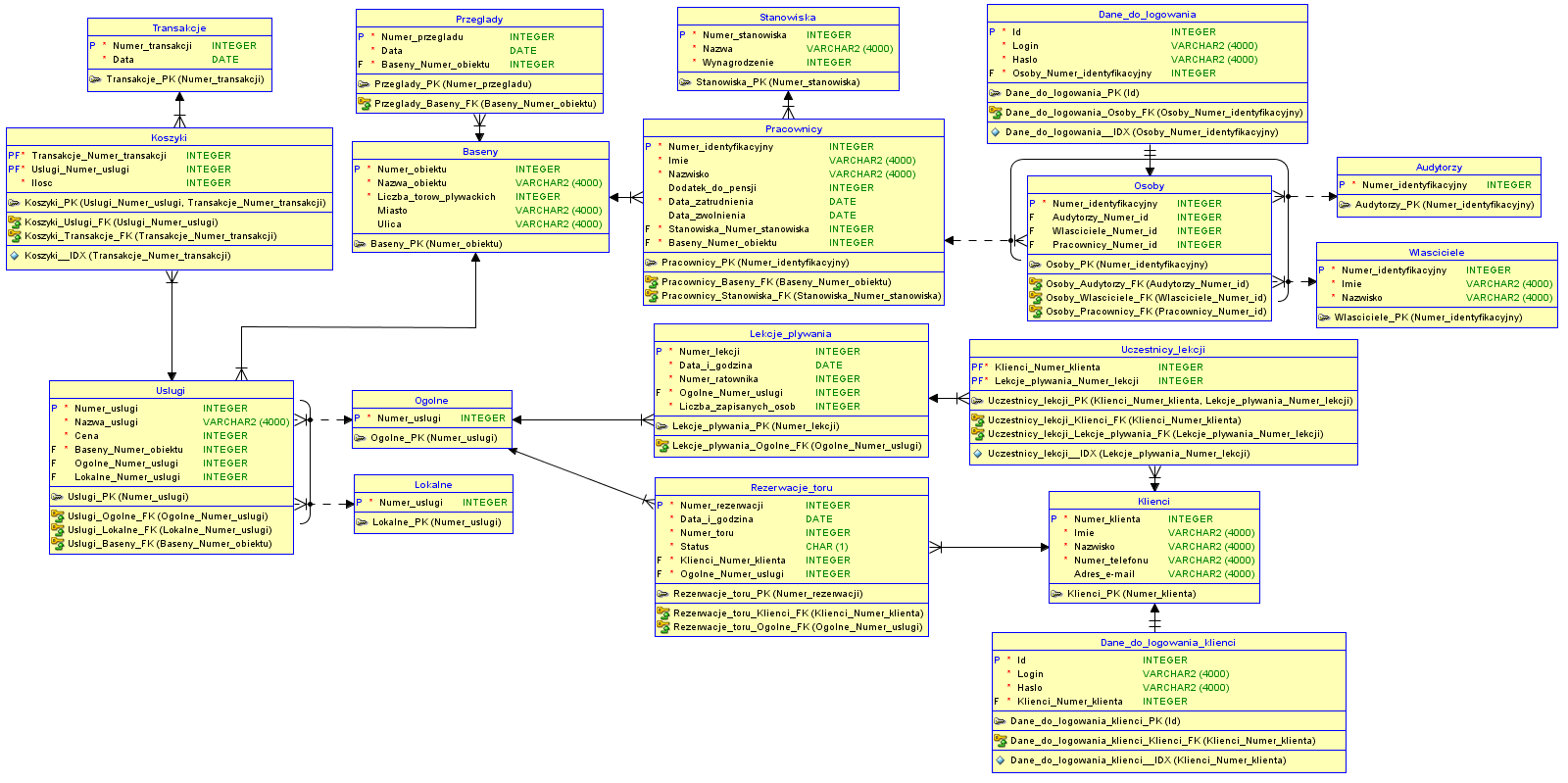


# Model relacyjny

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. Istotną różnicą jest jednak pojawienie się nowej relacji **Uczestnicy\_lekcji** powstałej w wyniku dekompozycji związku M:N między encjami **Klient** a **Lekcja pływania** z modelu logicznego na relacje 1:N i N:1. Dla encji **Osoba** i **Usluga** oraz każdego z ich podtypów zostały utworzone oddzielne relacje (Table for each entity). Do każdej relacji reprezentującej nadtyp zostały dodane opcjonalne klucze obce wiążące je z relacjami reprezentującymi podtypy. Został też dodany constraint wymuszający, aby jeden rekord w relacji reprezentującej nadtyp był powiązany tylko z jednym rekordem w relacji reprezentującej jeden z podtypów.

Związki jednoznaczne (1:N) zostały transformowane do klucza obcego w tabeli po stronie „wiele”. Związki jedno-jednoznaczne (pomiędzy encją **Osoba/Klient**, a encją **Dane do logowania/Dane do logowania klientow**) zostały transformowane do klucza obcego w tabeli **Dane\_do\_logowania/ Dane\_do\_logowania\_klienci** (przy jednoczesnym dodaniu unikalnego indexu do kolumny z kluczem obcym).

Model relacyjny przedstawia znormalizowaną bazę danych. Kolejne etapy jej normalizacji można prześledzić, obserwując poprzedzające relacyjny modele bazy danych. Najpierw została wprowadzona atomowość danych (przekształcenie atrybutów kompozytowych takich jak: **Dane do logowania** i **Stanowiska** na encje), co odpowiada pierwszej postaci normalnej.Przejście do drugiej postaci normalnej objawia się wyróżnieniem wcześniej wymienionych encji: **Stanowisko** (poszczególne stanowiska mogą się powtarzać wśród **Pracowników**) i **Dane do logowania** (nie wszyscy **Pracownicy** posiadają dostęp do bazy danych), do zaobserwowania w modelu logicznym. Przejście do trzeciej postaci normalnej okazało się natomiast zbyteczne, gdyż w naszym modelu każdy niekluczowy argument jest bezpośrednio zależny tylko od klucza głównego a nie od innej kolumny, co definiuje 3NF. Przejście do postaci BCNF jest w naszym przypadku bezcelowe – objawiłoby się ono wyróżnieniem nowej relacji **Adresy** powiązanej z **Basenami**, a sieć składa się z na tyle małej ilości **Basenów**, że każdy posiada inny adres. Ostatecznie więc zatrzymaliśmy się na trzecim etapie normalizacji.



# Spis relacji, kluczy głównych i kluczy obcych

|  |  |  |
| --- | --- | --- |
| **Nazwa relacji** | **Klucz główny** | **Klucz obcy** |
| Baseny | Numer\_obiektu |  |
| Osoby | Numer\_identyfikacyjny | Audytorzy\_Numer\_id; Wlasciciele\_Numer\_id; Pracownicy\_Numer\_id |
| Pracownicy | Numer\_identyfikacyjny | Baseny\_Numer\_obiektu;  Stanowiska\_Numer\_stanowiska |
| Stanowiska | Numer\_stanowiska |  |
| Audytorzy | Numer\_identyfikacyjny |  |
| Wlasciciele | Numer\_identyfikacyjny |  |
| Przeglady | Numer\_przegladu | Baseny\_Numer\_obiektu |
| Uslugi | Numer\_uslugi | Baseny\_Numer\_obiektu; Lokalne\_Numer\_uslugi; Ogolne\_Numer\_uslugi |
| Lokalne | Numer\_uslugi |  |
| Ogolne | Numer\_uslugi |  |
| Koszyki | Uslugi\_Numer\_uslugi, Transakcje\_Numer\_transakcji (klucz kompozytowy) | Uslugi\_Numer\_uslugi; Transakcje\_Numer\_transakcji |
| Transakcje | Numer\_transakcji |  |
| Klienci | Numer\_klienta |  |
| Rezerwacje\_toru | Numer\_rezerwacji | Klienci\_Numer\_klienta; Ogolne\_Numer\_uslugi |
| Lekcje\_plywania | Numer\_lekcji | Ogolne\_Numer\_uslugi |
| Dane\_do\_logowania | Id | Osoby\_Numer\_identyfikacyjny |
| Dane\_do\_logowania\_klienci | Id | Klienci\_Numer\_klienta |
| Uczestnicy\_lekcji | Klienci\_Numer\_klienta, Lekcje\_plywania\_Numer\_lekcji (klucz kompozytowy) | Klienci\_Numer\_klienta, Lekcje\_plywania\_Numer\_lekcji |

# Model fizyczny

W modelu fizycznym będą poruszone cztery zagadnienia: wolumetryka, partycjonowanie, indeksowanie i kopie zapasowe.

Wolumetryka, czyli wielkość baz danych jak i ich przyrost czasowy. W naszej bazie największymi tabelami są: **Transakcje**, **Koszyki**, **Lekcje\_plywania**, **Rezerwacje\_toru**, ewentualnie **Klienci** i **Dane\_do\_logowania** (aczkolwiek są one znacznie mniejsze, niż te wymienione w pierwszej kolejności). Każda z czterech pierwszych tabel charakteryzuje się wysokim przyrostem czasowym, z tego względu można rozważyć przechowywanie danych np. jedynie z dwóch ostatnich lat (takie rozwiązanie zastosowaliśmy). Można również usuwać dane klientów (i ich dane do logowania), którzy nie korzystali z usług przez dłuższy czas (np. rok). Pozostałe tabele są małe i praktycznie nie występuje w nich przyrost czasowy (a jeśli nawet, to znikomy).

W celach partycjonowania zostało utworzonych dziewięć przestrzeni tabel, odpowiadających poszczególnym kwartałom lat 2017-2019. Do partycjonowania wybraliśmy tabele: **Transakcje**, **Lekcje\_plywania**, **Rezerwacje\_toru**. Każda z nich została podzielona na dziewięć partycji (zgodnie z przyjętym kryterium podziału), które zostały przyporządkowane do odpowiednich przestrzeni tabel. Wybraliśmy takie właśnie tabele, ponieważ są one największymi w naszej bazie. Kryterium podziału również było dosyć oczywiste, ponieważ dostępy do tych tabel zwykle wiążą się z badaniem pewnego przedziału czasowego (np. kwartału). Tabela **Koszyki** nie była partycjonowana, ze względu na jej związek z tabelą **Transakcje**.

Indeksowaniu zostały poddane trzy tabele (a właściwie trzy kolumny z nich pochodzące). Jeden z indeksów, niezwiązany bezpośrednio z modelem fizycznym, został nałożony na pochodzącą z tabeli **Dane\_do\_logowania** kolumnę **Osoby\_Numer\_identyfikacyjny**, będący jej kluczem obcym, w celu zapewnienia unikalności danych do logowania dla każdego z klientów. Pozostałe dwa, jeden nałożony na kolumnę **Lekcje\_plywania\_Numer\_lekcji** w tabeli **Uczestnicy\_lekcji**, drugi – na kolumnę **Transakcje\_Numer\_transakcji** w tabeli **Koszyki**, dotyczą tabel „asocjacyjnych”, powstałych w wyniku dekompozycji związku M:N bądź w celu połączenia ze sobą dwóch encji. Dzięki tym indeksom szybciej zostaną odnalezione identyfikatory klientów biorących udział w lekcji pływania czy też numery usług zakupionych w danej transakcji.

Ostatni punkt dotyczy kopii zapasowych bazy danych. Jako że taka kopia jest tworzona domyślnie podczas instalacji bazy (i dotyczy ona później jej całości), zostaną podane jedynie tabele, dla których istotne jest utworzenie takiej kopii, ze względu na brak możliwości odtworzenia utraconych w razie awarii danych. Są to: **Transakcje**, **Lekcje\_plywania**, **Rezerwacje\_toru**, **Klienci**, **Dane\_do\_logowania**.