



BDBT

Projekt

Paweł Denst 303927

Szymon Olejnik 297303

Politechnika Warszawska, Instytut Telekomunikacji

19 listopada 2020

Spis treści

1. Zakres i cel projektu	3
2. Definicja systemu	3
2.1. Perspektywy użytkowników	3
3. Model konceptualny	4
3.1. Definicja zbiorów encji określonych w projekcie oraz określenie atrybutów i ich dziedzin	4
3.2. Ustalenie związków między encjami ich typów	7
3.3. Klucze kandydujące i główne (decyzje projektowe)	8
3.4. Schemat ER na poziomie konceptualnym	8
4. Model logiczny	8
4.1. Charakterystyka modelu relacyjnego	8
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady	9
4.2.1. Poprawa związków wielu do wielu	9
4.2.2. Usunięcie specjalizacji	9
4.3. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady	10
4.4. Proces normalizacji – analiza i przykłady	10
4.5. Schemat ER na poziomie modelu logicznego	13
4.6. Więzy integralności	13
4.7. Proces denormalizacji	13
5. Faza fizyczna	14
5.1. Projekt transakcji i weryfikacja ich wykonalności	14
5.2. Strojanie bazy danych – dobór indeksów	14
5.2.1. Kluby - Adresy	14
5.2.2. Pracownicy - Kluby, Adresy, Stanowiska	14
5.2.3. Obiekty - Kluby, Adresy	14
5.2.4. Usługi - Kluby, Klienci	14
5.2.5. Klienci - Adresy	14
5.2.6. Właściciele - Adresy, Kluby	15
5.2.7. Wynagrodzenia - Pracownicy	15
5.3. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	15
5.3.1. Wyszukaj jakie obiekty są przypisane do poszczególnych usług	15
5.4. Odszukaj Imię i Nazwisko klienta, wynajmującego obiekt z podanym numerem obiektu	15
5.4.1. Wyszukaj wszystkie korty wraz z nazwami	15
5.4.2. Wyszukaj obiekty w Poznaniu	15
5.5. Skrypt SQL tworzący bazę danych	16
5.6. Skrypt SQL uzupełniający bazę danych	29
Literatura	33

1. Zakres i cel projektu

Celem projektu jest stworzenie relacyjnej bazy danych w modelu konceptualnym i logicznym oraz implementacja bazy w bazie SQL.

Realizowany projekt dotyczy przedsiębiorstwa z branży sportowej, a dokładniej klubu tenisowego. Klub ten zajmuje się różnorodną działalnością związaną z tenisem. Oferta klubu obejmuje wydarzenia związane z tenisem, takie jak turnieje, zajęcia z doświadczonymi trenerami, wynajem posiadanych przez nas obiektów sportowych. Turnieje przez nas organizowane mogą wymagać wpisowego, ale zapewniamy wtedy wartościowe tenisowe nagrody. W przypadku braku potrzebnego sprzętu istnieje możliwość wypożyczenia rakiet i piłeczek z naszych zasobów. Posiadamy szereg obiektów, w tym korty na wszystkich nawierzchniach oraz kilka zadaszonych kompleksów z boiskami wielofunkcyjnymi. Nasze placówki są otwarte dla kibiców, większość ma zapewnione trybuny. Wszystkie obiekty są utrzymane w idealnym stanie przez odpowiednio wyszkoloną kadrę. Korzystanie z naszych usług wymaga dołączenia do klubu, a więc podania swoich podstawowych danych personalnych - dzięki temu również klub tworzy i utrzymuje bazę klientów, którą zgłosiło do GIODO - nasza baza danych będzie bazą, w której dane o nich będą składowane. W bazie przechowywane są również informacje dotyczące wszystkich pracowników, jak również wypłacanych im comiesięcznie wynagrodzeń.

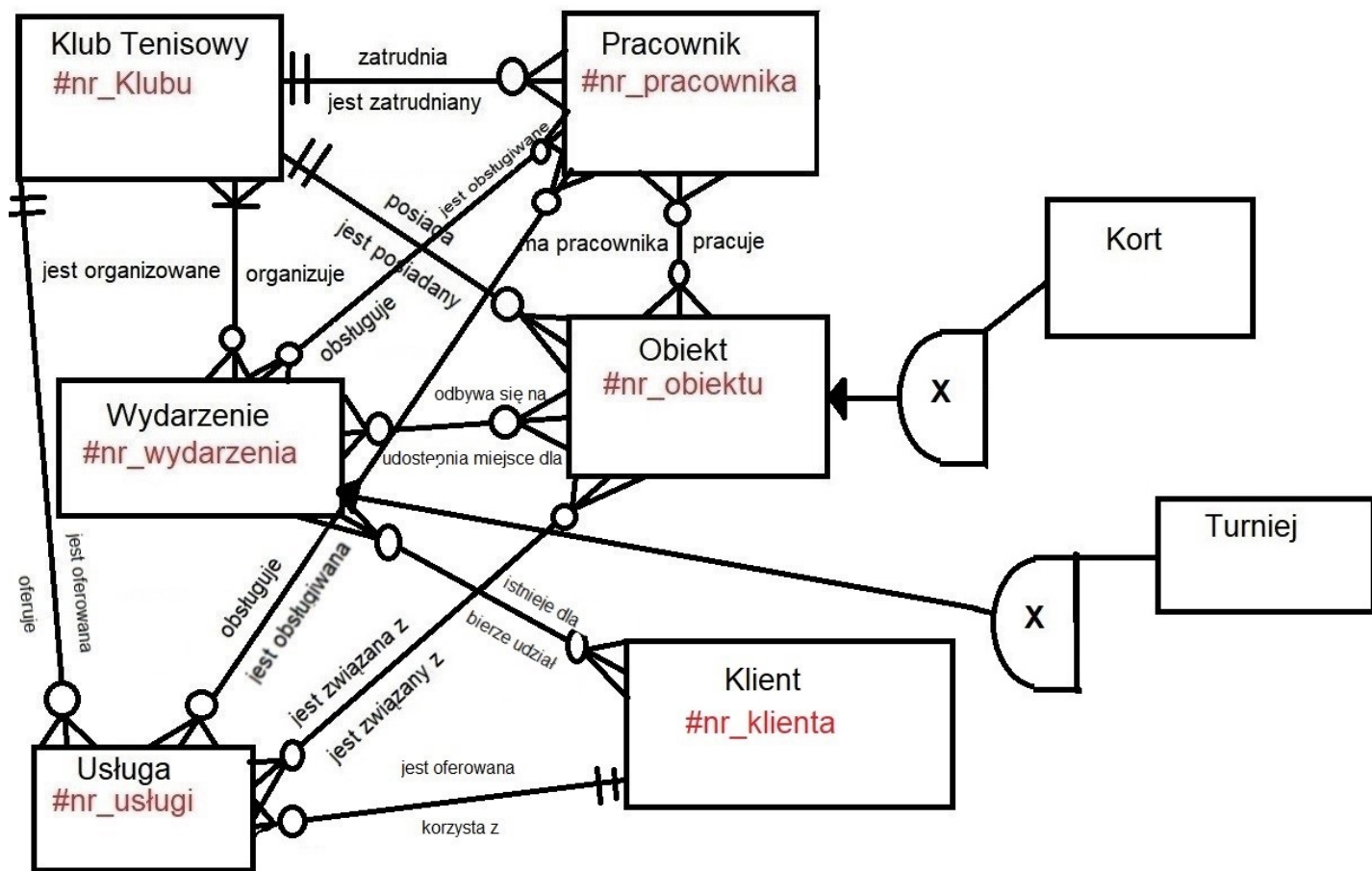
2. Definicja systemu

1. Dodawanie, usuwanie i modyfikacja Klubu Tenisowego
2. Podgląd danych Klubu Tenisowego
3. Dodawanie, usuwanie i modyfikacja danych pracownika klubu
4. Podgląd danych pracownika klubu
5. Dodawanie, usuwanie i modyfikacja obiektu posiadanego przez klub
6. Podgląd danych obiektu posiadanego przez klub
7. Dodawanie, usuwanie i modyfikacja wydarzenia jakie organizuje klub
8. Podgląd danych wydarzenia jakie organizuje klub
9. Dodawanie, usuwanie i modyfikacja usług świadczonych przez klub
10. Podgląd usług świadczonych przez klub
11. Dodawanie, usuwanie i modyfikacja danych klienta
12. Podgląd danych klienta
13. Dodawanie, usuwanie i modyfikacja usług z których korzysta klient
14. Podgląd usług z których korzysta klient

2.1. Perspektywy użytkowników

- Kierownik przedsiębiorstwa, właściciel, mający dostęp do wszystkich danych.
- Księgowa, mająca dostęp do wszystkich danych związanych z kosztorysem.
- Pracownik, widzący własne dane osobowe, historie wynagrodzenia oraz zadania, które wynikają z jego pracy. Przykładowo trener widzi swoich podopiecznych albo dostępność kortów, a konserwator obiekt, którym się "opiekuje".

3. Model konceptualny



Rys. 1. Pierwszy model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie oraz określenie atrybutów i ich dziedzin

Klub tenisowy - główna encja, która zawiera dane na temat naszego klubu.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_klubu	SmallInt	Obowiązkowy - klucz główny pozwala na identyfikację klubu	Unikatowy identyfikator klubu
Nazwa	VarChar(50)	Obowiązkowy - klub posiada nazwę	Nazwa klubu tenisowego
Adres	VarChar(300)	Obowiązkowy	Adres klubu, pole segmentowe, {miasto, ulica, numer lokalu, kod pocztowy, poczta}
Data_zalozenia	Date	Obowiązkowy	Data założenia klubu
Wlasciciel	VarChar(500)	Obowiązkowy	Właściciel klubu, pole segmentowe {imię, nazwisko}, wielowartościowe

Pracownik - encja, która zawiera dane na temat zatrudnionych przez nasz klub pracowników.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_pracownika	Integer	Obowiązkowy	Unikatowy identyfikator pracownika
Imie	VarChar(20)	Obowiązkowy	Imię pracownika
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko pracownika
Data_urodzenia	Date	Obowiązkowy	Data urodzenia pracownika
PESEL	Character(11)	Opcjonalny	Numer PESEL pracownika
Plec	PlecD Character(1) (K/M)	Obowiązkowy	Płeć pracownika
Adres	VarChar(400)	Obowiązkowy	Adres pracownika, pole segmentowe, {miasto, ulica, numer lokalu, kod pocztowy, poczta}
Stanowisko	VarChar(30)	Obowiązkowy	Stanowisko pracownika
Data_zatrudnienia	Date	Obowiązkowy	Data zatrudnienia pracownika
Data_zwolnienia	Date	Opcjonalny	Data zwolnienia pracownika
Wynagrodzenie	Money	Opcjonalny	Kwota wynagrodzenia pracownika, pole segmentowe, {data wypłaty, kwota wypłaty}, pole wielowartościowe
Wazne_badania_lekarskie	Boolean	Obowiązkowy	Pracownik posiada badania lekarskie
Kwalifikacje	VarChar(500)	Opcjonalny	Kwalifikacje pracownika

Obiekt - encja, która zawiera dane na temat posiadanych przez nasz klub placówek.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_objektu	Integer	Obowiązkowy	Unikatowy identyfikator obiektu
Status_objektu	Status_ObjektuD VarChar(30) (Otwarty/Zamknięty)	Obowiązkowy	Określa status obiektu
Nazwa	VarChar(100)	Obowiązkowy	Nazwa obiektu
Adres	VarChar(400)	Obowiązkowy	Adres obiektu, pole segmentowe, {miasto, ulica, numer lokalu, kod pocztowy, poczta}
Typ_objektu	VarChar(30)	Obowiązkowy	Typ obiektu

Wydarzenie - encja, która zawiera dane na temat organizowanych przez nasz klub wydarzeń.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_wydarzenia	Integer	Obowiązkowy	Unikatowy identyfikator wydarzenia
Nazwa	VarChar(100)	Obowiązkowy	Nazwa wydarzenia
Data_roz poczenia_wydarzenia	Date	Obowiązkowy	Data rozpoczęcia wydarzenia
Opis	VarChar(50)	Obowiązkowy	Opis wydarzenia
Data_zakonczenia_wydarzenia	Date	Opcjonalny	Data zakończenia wydarzenia
Wpisowe	Money	Opcjonalny	Oplata za wzięcie udziału w wydarzeniu

Usługa - encja, która zawiera dane na temat oferowanych przez nasz klub usług.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_uslugi	Integer	Obowiązkowy	Unikatowy identyfikator usługi
Nazwa	VarChar(200)	Obowiązkowy	Nazwa usługi
Data_oplacen ia_uslugi	DateTime	Obowiązkowy	Data zapłacenia za usługę
Data_roz poczenia_uslugi	DateTime	Opcjonalny	Data rozpoczęcia usługi
Data_zakonczenia_uslugi	DateTime	Opcjonalny	Data zakończenia usługi
Cena	Money	Obowiązkowy	Cena usługi
Sprzet	VarChar(300)	Opcjonalny	Jaki sprzęt wypożyczamy/ sprzedajemy klientowi
Typ_uslugi	Typ_uslugiD VarChar(15) (Wynajem/Wypożyczenie/ Zakup/Trening)	Obowiązkowy	Typ wykonywanej usługi

Klient - encja, która zawiera dane na temat obsługiwanych przez nasz klub klientów.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_klienta	Integer	Obowiązkowy	Unikatowy identyfikator klienta
Imie	VarChar(30)	Obowiązkowy	Imie Klienta
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko klienta
Adres	VarChar(300)	Obowiązkowy	Adres klienta, pole segmentowe, {miasto, ulica, numer lokalu, kod pocztowy, poczta}
Nr_telefonu	VarChar(15)	Opcjonalny	Numer telefonu klienta
E_mail	VarChar(45)	Opcjonalny	Adres e-mail klienta
PESEL	Character(11)	Opcjonalny	Numer PESEL klienta
Data_urodzenia	Date	Opcjonalny	Data urodzenia klienta
Plec	PlecD Character(1) (K/M)	Opcjonalny	Płeć klienta

Kort - encja, która zawiera dane na temat posiadanych przez nasz klub kortów, specjalizacja encji Obiekt.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nawierzchnia	NawierzchniaD VarChar(30) (Trawiasta/Twarda/ Ceglana)	Obowiązkowy	Nawierzchnia kortu
Trybuny	Boolean	Obowiązkowy	Kort posiada trybuny
Krzeselko_sedziowskie	Boolean	Obowiązkowy	Kort posiada krzeselko sędziowskie

Turniej - encja, która zawiera dane na temat organizowanych przez nasz klub turniejów, specjalizacja encji Wydarzenie.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Poziom_turnieju	Poziom_TurniejuD VarChar(15) (Junior/Amator/ Profesjonalista/Open)	Obowiązkowy	Poziom turnieju
Typ_rozgrywek	Typ_RozgrywekD VarChar(10) (singiel/debel/mikst)	Obowiązkowy	Określa typ turnieju
Nagroda	Money	Opcjonalny	Wysokość nagrody
System_rozgrywania_turnieju	VarChar(30)	Obowiązkowy	System rozgrywania turnieju (np. grupy, drabinka brazylijska, itp)

3.2. Ustalenie związków między encjami ich typów

- Klub tenisowy + Pracownik - Klub zatrudnia Pracownika, Pracownik jest zatrudniany przez Klub, związek jeden do wielu.
- Klub tenisowy + Wydarzenie - Klub organizuje Wydarzenie, Wydarzenie jest organizowane przez Klub, związek wiele do wielu.
- Klub tenisowy + Obiekt - Klub posiada Obiekt, Obiekt jest posiadany przez Klub, związek jeden do wielu.
- Klub tenisowy + Usługa - Klub oferuje Usługę, Usługa jest oferowana przez Klub, związek jeden do wielu.
- Pracownik + Wydarzenie - Pracownik obsługuje Wydarzenie, Wydarzenie jest obsługiwane przez Pracownika, związek wiele do wielu.
- Pracownik + Usługa - Pracownik obsługuje Usługę, Usługa jest obsługiwana przez Pracownika, związek wiele do wielu.
- Pracownik + Obiekt - Pracownik pracuje na Obiekcie, Obiekt ma Pracownika, związek wiele do wielu.

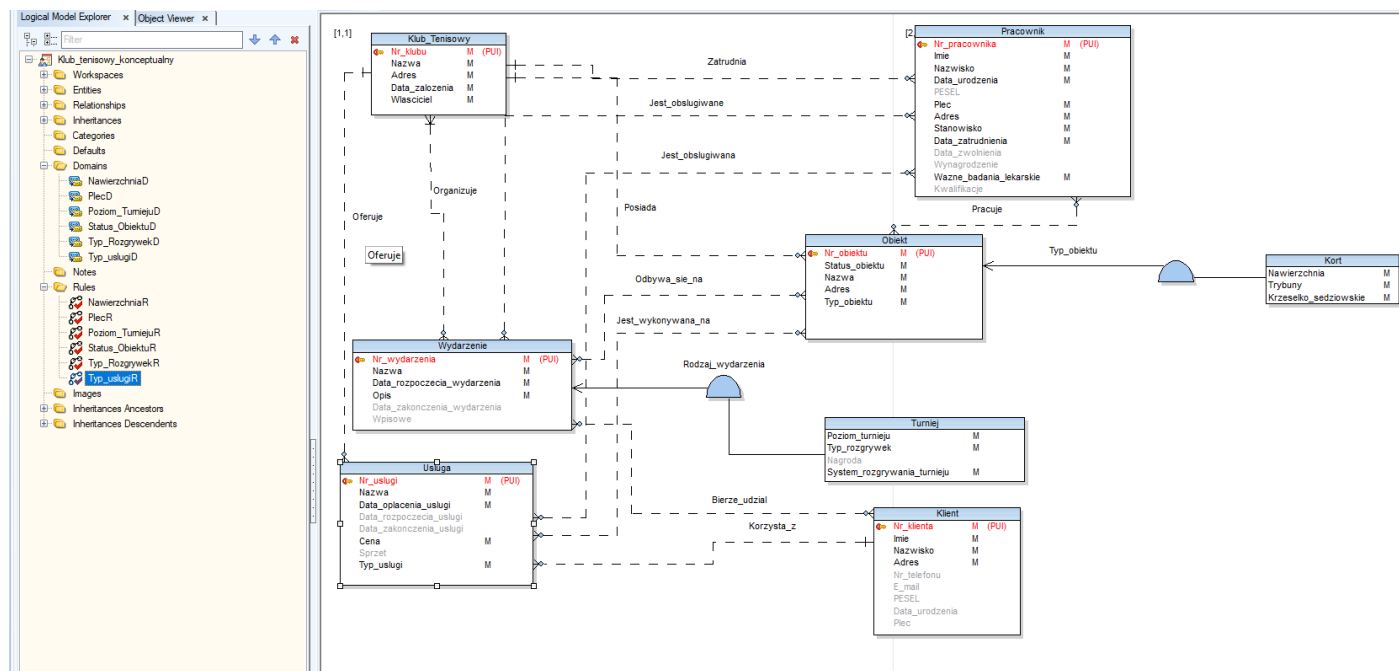
- Wydarzenie + Obiekt - Wydarzenie odbywa się na Obiekcie, Obiekt udostępnia miejsce dla Wydarzenia, związek wiele do wielu.
- Wydarzenie + Klient - Wydarzenie istnieje dla Klienta, Klient bierze udział w Wydarzeniu, związek wiele do wielu.
- Klient + Usługa - Klient korzysta z Usługi, Usługa jest oferowana Klientowi, związek jeden do wielu.

3.3. Klucze kandydujące i główne (decyzje projektowe)

Zdecydowaliśmy się na stworzenie sztucznych unikatowych kluczy identyfikacyjnych, co zagwarantowało nam jednoznaczność danych.

Nazwa encji	Potencjalny klucz kandydujący
Klub_Tenisowy	Nazwa
Pracownik	---
Obiekt	---
Wydarzenie	Nazwa+Data_roz poczenia_wydarzenia
Usługa	---
Klient	---

3.4. Schemat ER na poziomie konceptualnym



Rys. 2. Schemat ER na poziomie konceptualnym

4. Model logiczny

4.1. Charakterystyka modelu relacyjnego

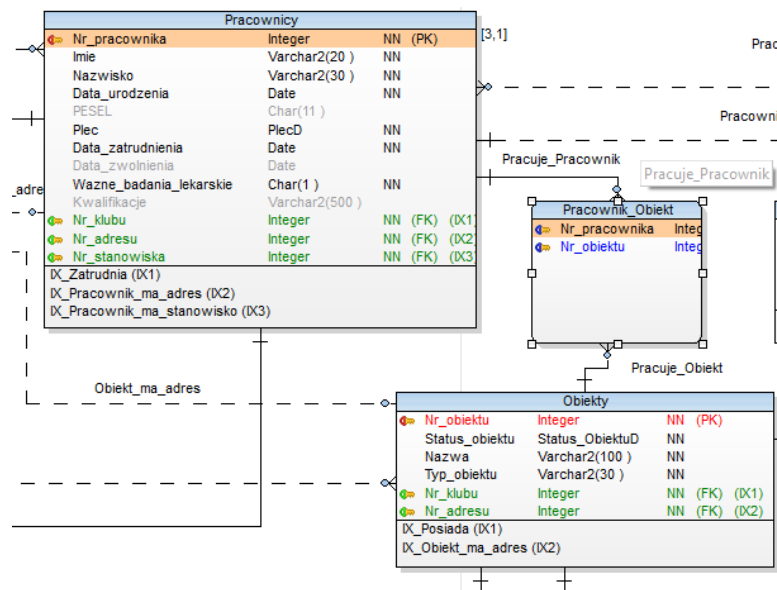
Model logiczny tworzyliśmy w programie Toad Data Modeler 7.3. Pierwszym krokiem było stworzenie tablic bridge'ujących odpowiedzialnych za naprawienie problemu związków wielu do wielu w modelu relacyjnym. Kolejną kwestią tego procesu była zmiana nazw relacji na liczbę mnogą - pozwoliło to zachować unikatowość w

modelu koncepcyjnym. Główną częścią tego etapu jest stworzenie dodatkowych relacji. Nowo tworzone relacje powstały z atrybutów, które w poprzedniej części były wielowartościowe, posiadały pola segmentowe lub z tych, w których mogła wystąpić redundancja danych. Były to Wynagrodzenia, Właściciele, Stanowiska oraz Adresy. Kluczowe było także dodanie relacji Poczty, która naprawiała problem kilku adresów z tym samym kodem pocztowym. Adresy połączyliśmy z wszystkimi relacjami, w których atrybutach było pole "Adres".

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

4.2.1. Poprawa związków wielu do wielu

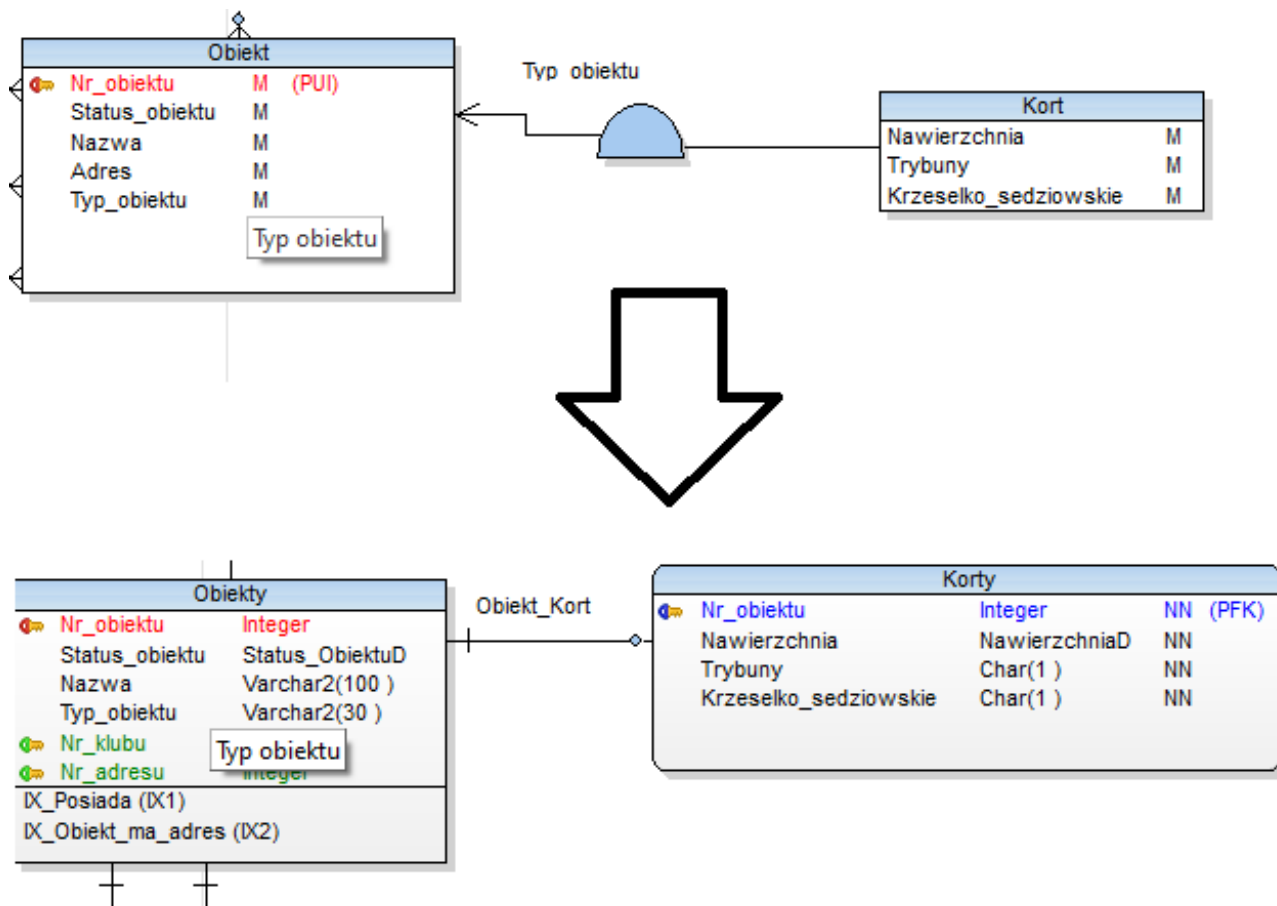
Jednym z poprawionych związków wielu do wielu, był ten łączący Pracowników oraz Obiekty. Dodanie Tablica Bridge'ującej oraz zmiana krotności związków z nią łączących (odznaczenie pola 'Mandatory') naprawia ten problem. Klucze, która posiada to: Nr_ pracownika oraz Nr_ obiektu.



Rys. 3. Poprawa związków wielu do wielu

4.2.2. Usunięcie specjalizacji

Jedną z usuniętych specjalizacji była specjalizacja Kort. Została ona zastąpiona relacją Korty.



Rys. 4. Usunięcie specjalizacji

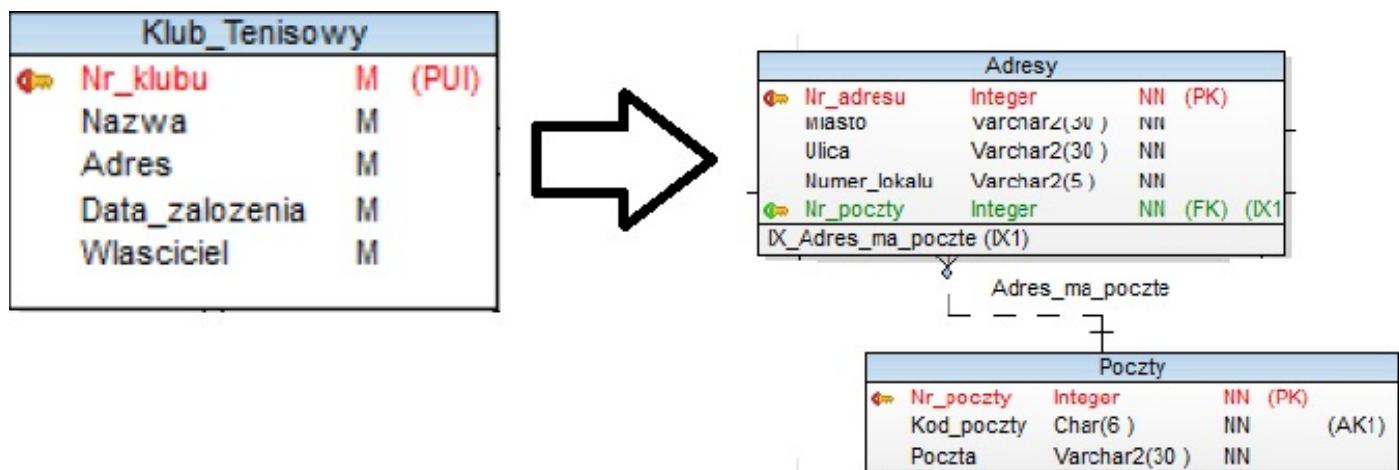
4.3. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

Jedną z możliwych pułapek szczelinowych występujących w naszym projekcie jest związek między Klientem, a Obiektem. W naszym rozumieniu bazy danych nie jest to problem, ponieważ Klient jest pośrednio połączony z obiektem przez usługę. Każde działanie na obiekcie jest wykonywane za pośrednictwem usługi. Jeśli przykładowo administrator chciałby sprawdzić kto aktualnie zajmuje kort, to przejście do usługi wynajmu wyświetli mu, kto obecnie je wynajmuje.

W projekcie nie doszło do problemu wachlarzowego. Unikaliśmy projektowania, które by do tego doprowadziło

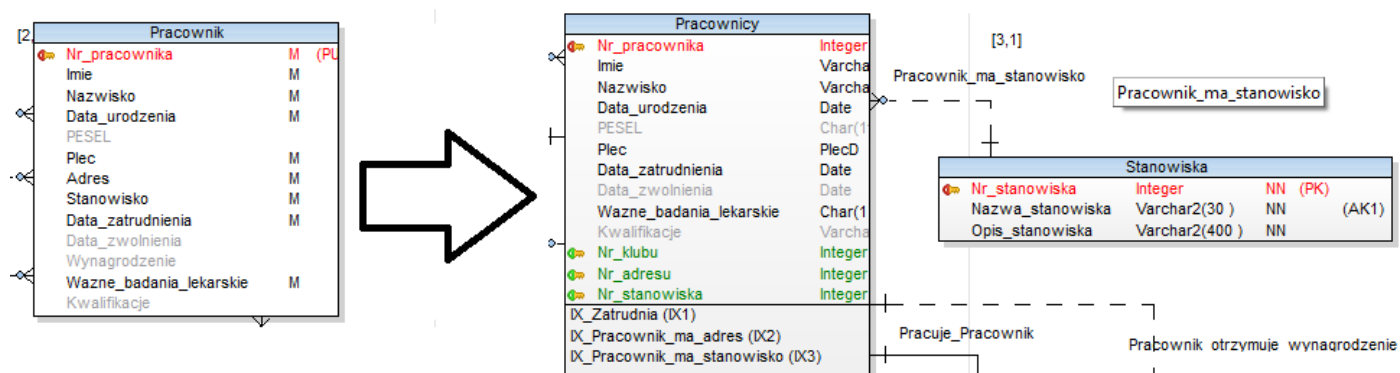
4.4. Proces normalizacji – analiza i przykłady

[4] W procesie normalizacji, na poziomie modelu logicznego definiowaliśmy nowe relacje dla atrybutów, które nie były polami prostymi. Przykładem może być pole Adresy, które było polem segmentowym oraz występowało w kilku relacjach - stworzyliśmy więc relację Adresy, połączoną z każdą relacją która zawierała Adres jako atrybut. Mając z tyłu głowy dążenie do trzeciej postaci normalnej, z relacji Adresy od razu wydzieliliśmy prostą relację słownikową Poczty, która odpowiadała za powtarzający się kod pocztowy (uwzględniając jednocześnie na powstanie problemu wachlarzowego).



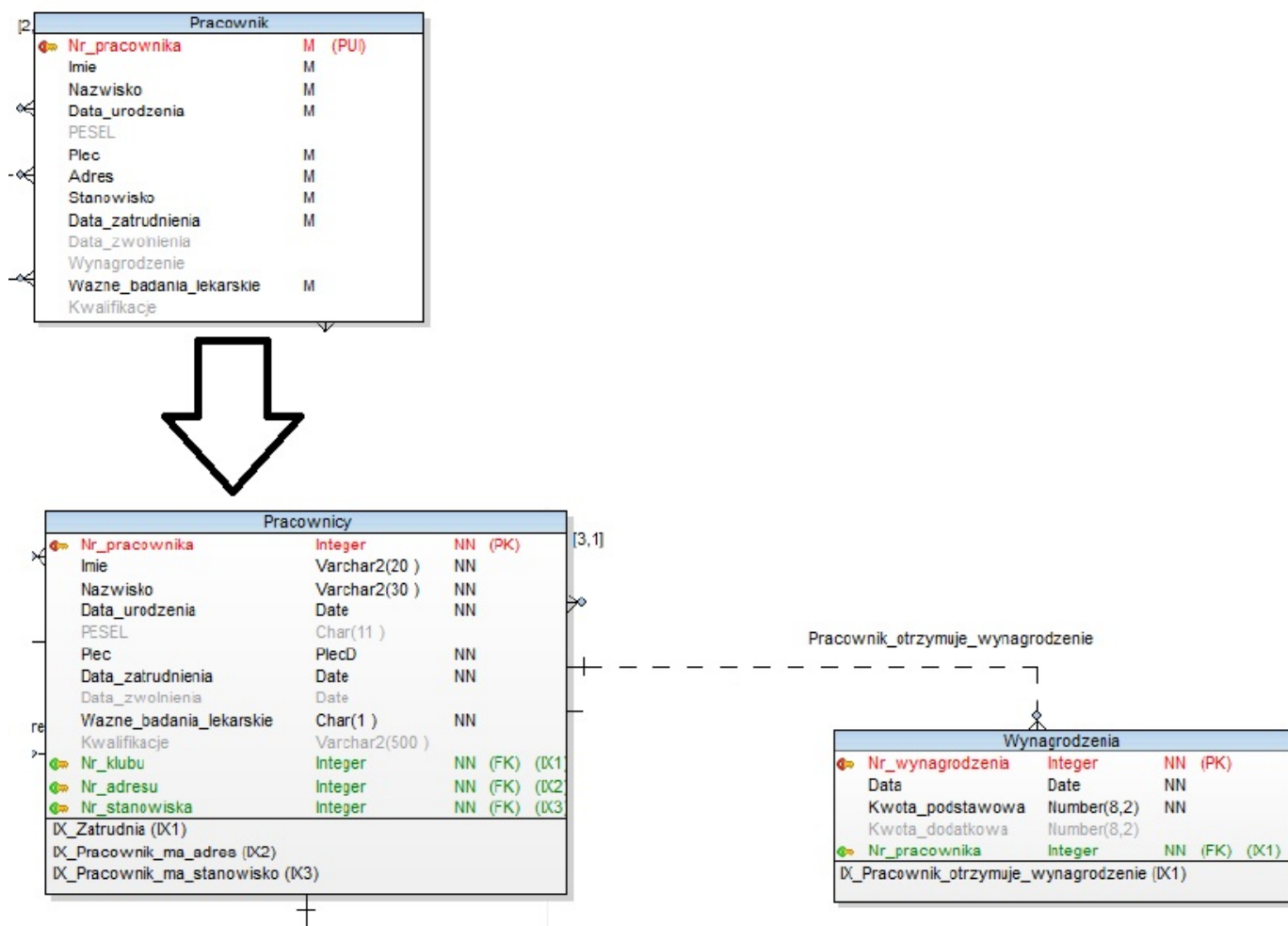
Rys. 5. Dodanie relacji Adresy i Poczty

Kolejnym z usuniętych atrybutów, który następnie został dodany jako Relacja było Stanowisko. Nową relacją były już Stanowiska. Utworzenie tej relacji wynikało z tego, że kilku pracowników miałyby to samo stanowisko, co prowadziłoby do redundancji danych i pojawiania się w bazach tego samego stanowiska wielokrotnie.

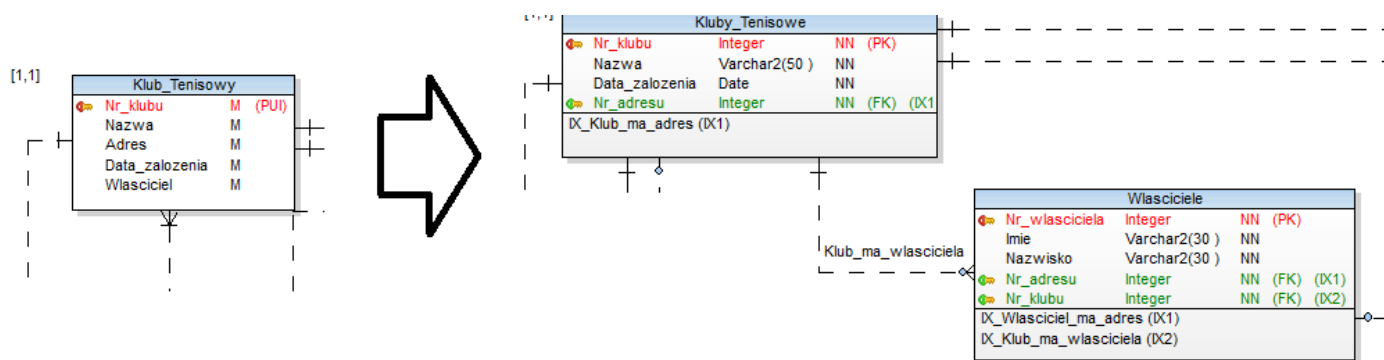


Rys. 6. Dodanie relacji Stanowiska

Innymi atrybutami zmienionymi w relacje były pola Wynagrodzenie oraz Właściciel.

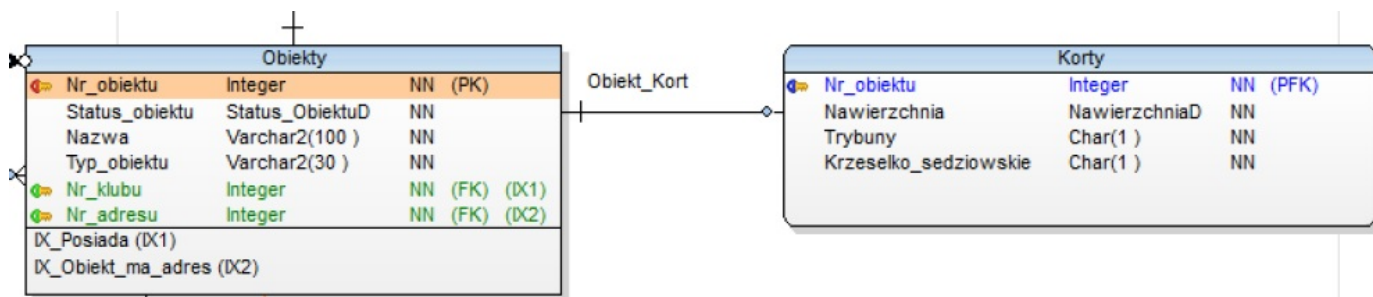


Rys. 7. Dodanie relacji Wynagrodzenia



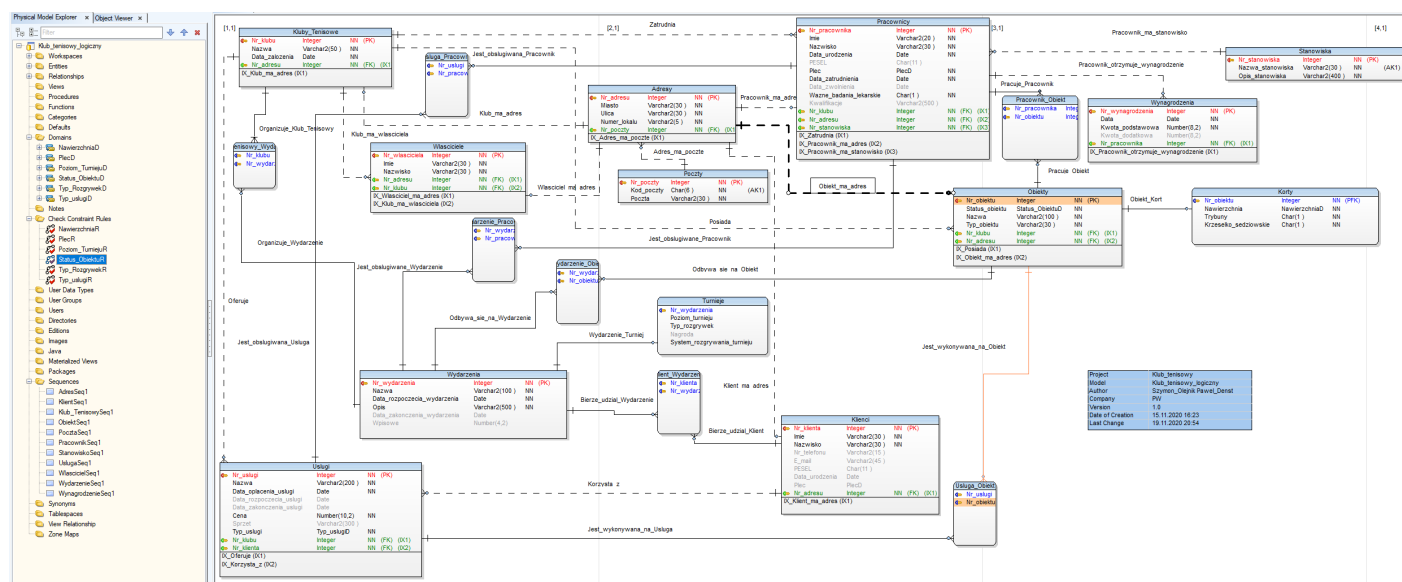
Rys. 8. Dodanie relacji Wlasciele

Zmienialiśmy też krotność związków wynikających ze specjalizacji na jeden do jeden (naprawiając problem szczelinowy). Również brak powtarzających się grup, powodował spełnienie pierwszej postaci normalnej. W relacjach posiadaliśmy same klucze proste, wszystkie klucze potencjalne również były kluczami prostymi oraz nie było zależności addytywne przechodnich, co skutkowało spełnieniem drugiej postaci normalnej. Nasze relacje specjalizujące od razu spełniały postać drugiej formy normalnej. Co więcej, żaden atrybut niekluczowy nie jest zależny od innych atrybutów niekluczowych, co skutkuje tym, że uzyskaliśmy trzecią postać normalną.



Rys. 9. Krotność związków jeden do jeden

4.5. Schemat ER na poziomie modelu logicznego



Rys. 10. Schemat ER na poziomie modelu logicznego

4.6. Więzy integralności

Na tym etapie projektowania nie pojawiają się już pola segmentowe, ani wielowartościowe. Wszystkie atrybuty są już polami atomowymi. Każdy klucz jest unikatowy i posiada jedynie niepowtarzalne wartości (UNIQUE). W niezbędnych atrybutach dodane zostało zastrzeżenie NOT NULL.

4.7. Proces denormalizacji

W naszym projekcie nie było potrzeby przeprowadzania procesu denormalizacji, ponieważ chcieliśmy zachować elastyczność naszego systemu. Analiza modelu logicznego utwierdziła nas w przekonaniu, że nie ma sensu mnożenia przechowywanych danych, ponieważ nie było miejsca w naszym projekcie, gdzie takie działanie przynosiłoby pozytywne czy optymalne skutki. Jedynym miejscem, w którym zastanawialiśmy się nad procesem denormalizacji była relacja Poczty. Nie wydawała się ona niezbędna do poprawnego działania bazy, jednak dodanie jej eliminowało problem posiadania takich samych kodów pocztowych przez kilka adresów. Niemniej usuwanie tej relacji nie wniosłoby nic pożytecznego w naszym projekcie.

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Potrzebne Relacje	Czy da się wykonać
Podgląd osobistych danych na temat pracownika	Pracownicy, Adresy, Wynagrodzenia	Tak
Podgląd osobistych danych na temat klienta	Klienci, Adresy	Tak
Edytowanie, kasowanie, dodawanie danych na temat pracowników	Pracownicy, Adresy, Wynagrodzenia	Tak
Edytowanie, kasowanie, dodawanie danych na temat klientów	Klienci, Adresy	Tak
Podgląd danych na temat wynajmowanych obiektów	Obiekty	Tak
Podgląd danych na temat wypożyczanego, sprzedawanego sprzętu	Usługi	Tak
Podgląd wypożyczonego sprzętu	Usługi	Tak
Podgląd obecnie trwających wydarzeń	Wydarzenia	Tak
Podgląd obecnie trwających trenigów	Usługi	Tak
Podgląd obecnie zajętych obiektów	Obiekty	Tak
Podgląd wynagrodzeń	Wynagrodzenia	Tak

Rys. 11. Dostępne transakcje i ich weryfikacja

5.2. Strojenie bazy danych – dobór indeksów

5.2.1. Kluby - Adresy

```
CREATE INDEX IX_Klub_ma_adres ON Kluby_Tenisowe (Nr_adresu)
```

5.2.2. Pracownicy - Kluby, Adresy, Stanowiska

```
CREATE INDEX IX_Zatrudnia ON Pracownicy (Nr_klubu)
```

```
CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (Nr_adresu)
```

```
CREATE INDEX IX_Pracownik_ma_stanowisko ON Pracownicy (Nr_stanowiska)
```

5.2.3. Obiekty - Kluby, Adresy

```
CREATE INDEX IX_Posiada ON Obiekty (Nr_klubu)
```

```
CREATE INDEX IX_Objekt_ma_adres ON Obiekty (Nr_adresu)
```

5.2.4. Usługi - Kluby, Klienci

```
CREATE INDEX IX_Oferuje ON Uslugi (Nr_klubu)
```

```
CREATE INDEX IX_Korzysta_z ON Uslugi (Nr_klienta)
```

5.2.5. Klienci - Adresy

```
CREATE INDEX IX_Klient_ma_adres ON Klienci (Nr_adresu)
```

\subsubsection{Adresy - Poczty}

```

\begin{minted}[breaklines]{SQL}/*
CREATE INDEX IX_Adres_ma_poczte ON Adresy (Nr_poczty)

```

5.2.6. Właściciele - Adresy, Kluby

```
CREATE INDEX IX_Wlasciciel_ma_adres ON Wlasciciele (Nr_adresu)
```

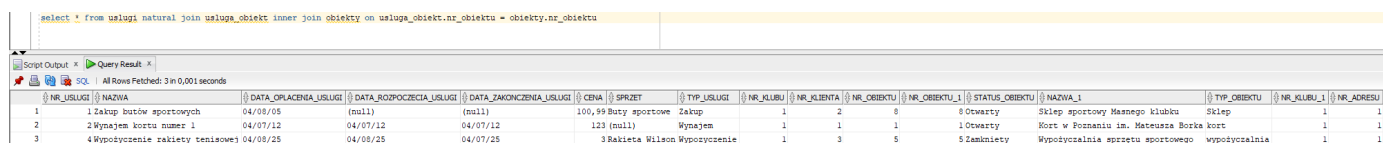
```
CREATE INDEX IX_Klub_ma_wlasciciela ON Wlasciciele (Nr_klubu)
```

5.2.7. Wynagrodzenia - Pracownicy

```
CREATE INDEX IX_Pracownik_otrzymuje_wynagrodzenie ON Wynagrodzenia (Nr_pracownika)
```

5.3. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

5.3.1. Wyszukaj jakie obiekty są przypisane do poszczególnych usług



```

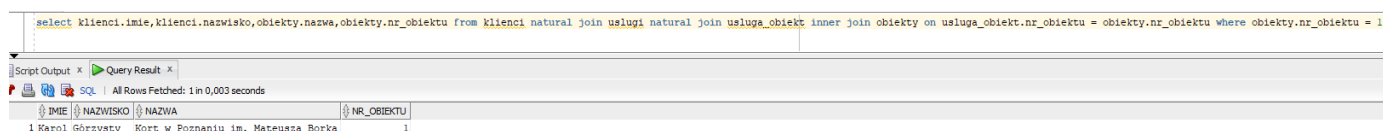
select * from uslugi natural join uslug_obiekt inner join obiekty on uslug_obiekt.nr_obiektu = obiekty.nr_obiektu

```

NR_USLUGI	NAMNA	DATA_OPLACENIA_USLUGI	DATA_ROZPOCZECIA_USLUGI	DATA_ZAKOŃCZENIA_USLUGI	CENA	SPRZET	TYP_USLUGI	NR_KLUBU	NR_KLIENTA	NR_OBIEKTU	NR_OBIEKTU_1	STATUS_OBIEKTU	NAMNA_1	TYP_OBIEKTU	NR_KLUBU_1	NR_ADRESU
1	1 Zakup butów sportowych	04/08/05	(null)	(null)	100,59	Buty sportowe	Zakup	1	2	8		8 Otwarty	Sklep sportowy Mamego klubu	Sklep	1	1
2	2 Wynajem kortu numer 1	04/07/12	04/07/12	04/07/12	123 (null)	Wynajem		1	1	1		1 Otwarty	Kort w Poznaniu im. Mateusza Borka	kort	1	1
3	4 Wypożyczenie rakietki tenisowej	04/08/25	04/08/25	04/07/25		3 Rakietka Wilson	Wypożyczenie	1	3	5		5 Zamknięty	Hypożyczalnia sprzętu sportowego	wypożyczalnia	1	1

Rys. 12. Zapytanie obiektu do usług

5.4. Odszukaj Imię i Nazwisko klienta, wynajmującego obiekt z podanym numerem obiektu



```

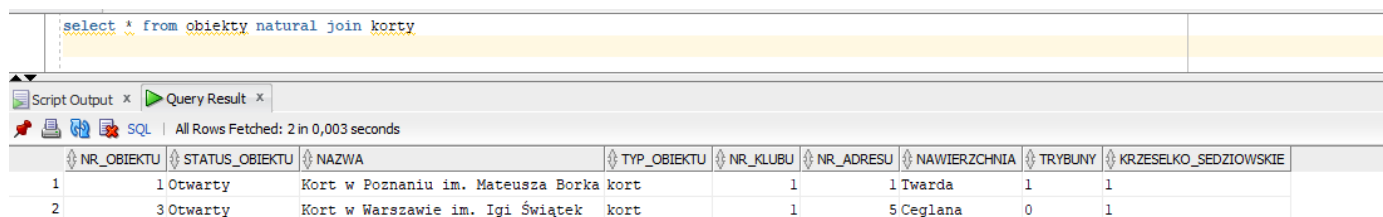
select klienci.imie,klienci.nazwisko,obiekty.nazwa,obiekty.nr_obiektu from klienci natural join uslugi natural join uslug_obiekt inner join obiekty on uslug_obiekt.nr_obiektu = obiekty.nr_obiektu where obiekty.nr_obiektu = 1

```

IMIE	NAMZWIKO	NAMZA	NR_OBIEKTU
1 Karol Górzysty	Kort w Poznaniu im. Mateusza Borka		1

Rys. 13. Zapytanie Klienta do obiektu

5.4.1. Wyszukaj wszystkie korty wraz z nazwami



```

select * from obiekty natural join korty

```

NR_OBIEKTU	STATUS_OBIEKTU	NAMZA	TYP_OBIEKTU	NR_KLUBU	NR_ADRESU	NAWIERZCHNIA	TRYBUNY	KRZESEŁKO_SEDZIOWSKIE
1	1 Otwarty	Kort w Poznaniu im. Mateusza Borka	kort	1	1 Twarda	1	1	
2	3 Otwarty	Kort w Warszawie im. Igi Świątek	kort	1	5 Ceglana	0	1	

Rys. 14. Zapytanie o korty z nazwami

5.4.2. Wyszukaj obiekty w Poznaniu

select * from obiekty natural join adresy where adresy.miasto = 'Poznań'										
Script Output x Query Result x										
All Rows Fetched: 5 in 0,003 seconds										
NR_ADRESU	NR_OBIKTU	STATUS_OBIKTU	NAZWA	TYP_OBIKTU	NR_KLUBU	MIASTO	ULICA	NUMER_LOKALU	NR_POCZTY	
1	1	1	Otwarty	Kort w Poznaniu im. Mateusza Borka	kort	1 Poznań	Kremówkowa 21			8
2	1	4	Zamknięty	Magazyn sportowy sprzętu	magazyn	1 Poznań	Kremówkowa 21			8
3	1	5	Zamknięty	Wypożyczalnia sprzętu sportowego	wypożyczalnia	1 Poznań	Kremówkowa 21			8
4	1	6	Zamknięty	Biuro przedsiębiorstwa Masny Klubik	biuro	1 Poznań	Kremówkowa 21			8
5	1	8	Otwarty	Sklep sportowy Masnego klubu	Sklep	1 Poznań	Kremówkowa 21			8

Rys. 15. Zapytanie obiektu do adresu

5.5. Skrypt SQL tworzący bazę danych

Created: 15.11.2020

Modified: 19.11.2020

Project: Klub_tenisowy

Model: Klub_tenisowy_logiczny

Company: PW

Author: Szymon Olejnik Pawel Denst

Version: 1.0

Database: Oracle 19c

*/

-- Create sequences section -----

CREATE SEQUENCE Klub_TenisowySeq1

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE PracownikSeq1

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE AdresSeq1

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE PocztaSeq1

INCREMENT BY 1

START WITH 1

NOMAXVALUE


```

NOMINVALUE
CACHE 20
/

CREATE SEQUENCE ObiektSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE WynagrodzenieSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE StanowiskoSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE WlascicielSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE UsługaSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE WydarzenieSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE KlientSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE

```

```

NOMINVALUE
CACHE 20
/

-- Create tables section -----

-- Table Kluby_Tenisowe

CREATE TABLE Kluby_Tenisowe(
    Nr_klubu Integer NOT NULL,
    Nazwa Varchar2(50 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Kluby_Tenisowe

CREATE INDEX IX_Klub_ma_adres ON Kluby_Tenisowe (Nr_adresu)
/

-- Add keys for table Kluby_Tenisowe

ALTER TABLE Kluby_Tenisowe ADD CONSTRAINT Klub_PK PRIMARY KEY (Nr_klubu)
/

-- Table Pracownicy

CREATE TABLE Pracownicy(
    Nr_pracownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('K','M')),
    Data_zatrudnienia Date NOT NULL,
    Data_zwolnienia Date,
    Wazne_badania_lekarskie Char(1 ) NOT NULL,
    Kwalifikacje Varchar2(500 ),
    Nr_klubu Integer NOT NULL,
    Nr_adresu Integer NOT NULL,
    Nr_stanowiska Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX IX_Zatrudnia ON Pracownicy (Nr_klubu)
/

CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (Nr_adresu)
/

CREATE INDEX IX_Pracownik_ma_stanowisko ON Pracownicy (Nr_stanowiska)
/

```

```

-- Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY (Nr_pracownika)
/

-- Table Obiekty

CREATE TABLE Obiekty(
    Nr_obiektu Integer NOT NULL,
    Status_obiektu Varchar2(15 ) NOT NULL
        CHECK (Status_obiektu IN ('Otwarty','Zamkniety')),
    Nazwa Varchar2(100 ) NOT NULL,
    Typ_obiektu Varchar2(30 ) NOT NULL,
    Nr_klubu Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Obiekty

CREATE INDEX IX_Posiada ON Obiekty (Nr_klubu)
/

CREATE INDEX IX_Objekt_ma_adres ON Obiekty (Nr_adresu)
/

-- Add keys for table Obiekty

ALTER TABLE Obiekty ADD CONSTRAINT Unique_Identifier6 PRIMARY KEY (Nr_obiektu)
/

-- Table Korty

CREATE TABLE Korty(
    Nr_obiektu Integer NOT NULL,
    Nawierzchnia Varchar2(10 ) NOT NULL
        CHECK (Nawierzchnia IN ('Trawiasta','Twarda','Ceglana')),
    Trybuny Char(1 ) NOT NULL,
    Krzeselko_sedziowskie Char(1 ) NOT NULL
)
/

-- Add keys for table Korty

ALTER TABLE Korty ADD CONSTRAINT Unique_Identifier7 PRIMARY KEY (Nr_obiektu)
/

-- Table Wydarzenia

CREATE TABLE Wydarzenia(
    Nr_wydarzenia Integer NOT NULL,
    Nazwa Varchar2(100 ) NOT NULL,
    Data_rozpoczecia_wydarzenia Date NOT NULL,
    Opis Varchar2(500 ) NOT NULL,
    Data_zakonczenia_wydarzenia Date,

```

```

    Wpisowe Number(4,2)
)
/

-- Add keys for table Wydarzenia

ALTER TABLE Wydarzenia ADD CONSTRAINT Unique_Identifier9 PRIMARY KEY (Nr_wydarzenia)
/

-- Table Turnieje

CREATE TABLE Turnieje(
    Nr_wydarzenia Integer NOT NULL,
    Poziom_turnieju Varchar2(15 ) NOT NULL
        CHECK (Poziom_turnieju IN ('Junior','Amator','Profesjonalista','Open')),
    Typ_rozgrywek Varchar2(10 ) NOT NULL
        CHECK (Typ_rozgrywek IN ('singiel','debel','mikst')),
    Nagroda Number(10,2),
    System_rozgrywania_turnieju Varchar2(30 ) NOT NULL
)
/

-- Add keys for table Turnieje

ALTER TABLE Turnieje ADD CONSTRAINT Unique_Identifier11 PRIMARY KEY (Nr_wydarzenia)
/

-- Table Usługi

CREATE TABLE Usługi(
    Nr_uslugi Integer NOT NULL,
    Nazwa Varchar2(200 ) NOT NULL,
    Data_oplacenia_uslugi Date NOT NULL,
    Data_rozpozeczenia_uslugi Date,
    Data_zakonczenia_uslugi Date,
    Cena Number(10,2) NOT NULL,
    Sprzet Varchar2(300 ),
    Typ_uslugi Varchar2(15 ) NOT NULL
        CHECK (Typ_uslugi IN ('Wynajem','Wypozyczenie','Zakup','Trening')),
    Nr_klubu Integer NOT NULL,
    Nr_klienta Integer NOT NULL
)
/

-- Create indexes for table Usługi

CREATE INDEX IX_Oferuje ON Usługi (Nr_klubu)
/

CREATE INDEX IX_Korzysta_z ON Usługi (Nr_klienta)
/

-- Add keys for table Usługi

ALTER TABLE Usługi ADD CONSTRAINT Unique_Identifier12 PRIMARY KEY (Nr_uslugi)
/

```

```

-- Table Klienci

CREATE TABLE Klienci(
    Nr_klienta Integer NOT NULL,
    Imie Varchar2(30 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Nr_telefonu Varchar2(15 ),
    E_mail Varchar2(45 ),
    PESEL Char(11 ),
    Data_urodzenia Date,
    Plec Char(1 )
        CHECK (Plec IN ('K','M')),
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Klienci

CREATE INDEX IX_Klient_ma_adres ON Klienci (Nr_adresu)
/

-- Add keys for table Klienci

ALTER TABLE Klienci ADD CONSTRAINT Unique_Identifier16 PRIMARY KEY (Nr_klienta)
/

-- Table Pracownik_Objekt

CREATE TABLE Pracownik_Objekt(
    Nr_pracownika Integer NOT NULL,
    Nr_objektu Integer NOT NULL
)
/

-- Table Klub_Tenisowy_Wydarzenie

CREATE TABLE Klub_Tenisowy_Wydarzenie(
    Nr_klubu Integer NOT NULL,
    Nr_wydarzenia Integer NOT NULL
)
/

-- Table Wydarzenie_Objekt

CREATE TABLE Wydarzenie_Objekt(
    Nr_wydarzenia Integer NOT NULL,
    Nr_objektu Integer NOT NULL
)
/

-- Table Usługa_Pracownik

CREATE TABLE Usługa_Pracownik(
    Nr_uslugi Integer NOT NULL,
    Nr_pracownika Integer NOT NULL
)
/

```

```

)
/

-- Table Wydarzenie_Pracownik

CREATE TABLE Wydarzenie_Pracownik(
    Nr_wydarzenia Integer NOT NULL,
    Nr_pracownika Integer NOT NULL
)
/

-- Table Usługa_Objekt

CREATE TABLE Usługa_Objekt(
    Nr_uslugi Integer NOT NULL,
    Nr_objektu Integer NOT NULL
)
/

-- Table Klient_Wydarzenie

CREATE TABLE Klient_Wydarzenie(
    Nr_klienta Integer NOT NULL,
    Nr_wydarzenia Integer NOT NULL
)
/

-- Table Adresy

CREATE TABLE Adresy(
    Nr_adresu Integer NOT NULL,
    Miasto Varchar2(30 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Numer_lokalu Varchar2(5 ) NOT NULL,
    Nr_poczty Integer NOT NULL
)
/

-- Create indexes for table Adresy

CREATE INDEX IX_Adres_ma_poczte ON Adresy (Nr_poczty)
/

-- Add keys for table Adresy

ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (Nr_adresu)
/

-- Table and Columns comments section

COMMENT ON COLUMN Adresy.Nr_adresu IS 'Unikatowy identyfikator adresu'
/
COMMENT ON COLUMN Adresy.Miasto IS 'Miasto'
/
COMMENT ON COLUMN Adresy.Ulica IS 'Ulica'
/

```

```

COMMENT ON COLUMN Adresy.Numer_lokalu IS 'Numer_lokalu'
/

-- Table Wlasciciele

CREATE TABLE Wlasciciele(
    Nr_wlasciciela Integer NOT NULL,
    Imie Varchar2(30 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Nr_adresu Integer NOT NULL,
    Nr_klubu Integer NOT NULL
)
/

-- Create indexes for table Wlasciciele

CREATE INDEX IX_Wlasciciel_ma_adres ON Wlasciciele (Nr_adresu)
/

CREATE INDEX IX_Klub_ma_wlasciciela ON Wlasciciele (Nr_klubu)
/

-- Add keys for table Wlasciciele

ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciele PRIMARY KEY (Nr_wlasciciela)
/

-- Table and Columns comments section

COMMENT ON COLUMN Wlasciciele.Nr_wlasciciela IS 'Unikatowy identyfikator właściciela'
/
COMMENT ON COLUMN Wlasciciele.Imie IS 'Imię właściciela'
/
COMMENT ON COLUMN Wlasciciele.Nazwisko IS 'Nazwisko właściciela'
/

-- Table Poczty

CREATE TABLE Poczty(
    Nr_poczty Integer NOT NULL,
    Kod_poczty Char(6 ) NOT NULL,
    Poczta Varchar2(30 ) NOT NULL
)
/

-- Add keys for table Poczty

ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (Nr_poczty)
/

ALTER TABLE Poczty ADD CONSTRAINT Kod_poczty UNIQUE (Kod_poczty)
/

-- Table and Columns comments section

COMMENT ON COLUMN Poczty.Nr_poczty IS 'Unikatowy identyfikator poczty'

```

```

/
COMMENT ON COLUMN Poczty.Kod_poczty IS 'Kod_poczty'
/
COMMENT ON COLUMN Poczty.Poczta IS 'Lokalizacja poczty'
/

-- Table Stanowiska

CREATE TABLE Stanowiska(
    Nr_stanowiska Integer NOT NULL,
    Nazwa_stanowiska Varchar2(30 ) NOT NULL,
    Opis_stanowiska Varchar2(400 ) NOT NULL
)
/

-- Add keys for table Stanowiska

ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stanowiska PRIMARY KEY (Nr_stanowiska)
/

ALTER TABLE Stanowiska ADD CONSTRAINT Nazwa_stanowiska UNIQUE (Nazwa_stanowiska)
/

-- Table and Columns comments section

COMMENT ON COLUMN Stanowiska.Nr_stanowiska IS 'Unikatowy identyfikator stanowiska'
/
COMMENT ON COLUMN Stanowiska.Nazwa_stanowiska IS 'Nazwa stanowiska'
/
COMMENT ON COLUMN Stanowiska.Opis_stanowiska IS 'Opis stanowiska'
/

-- Table Wynagrodzenia

CREATE TABLE Wynagrodzenia(
    Nr_wynagrodzenia Integer NOT NULL,
    Data Date NOT NULL,
    Kwota_podstawowa Number(8,2) NOT NULL,
    Kwota_dodatkowa Number(8,2),
    Nr_pracownika Integer NOT NULL
)
/

-- Create indexes for table Wynagrodzenia

CREATE INDEX IX_Pracownik_otrzymuje_wynagrodzenie ON Wynagrodzenia (Nr_pracownika)
/

-- Add keys for table Wynagrodzenia

ALTER TABLE Wynagrodzenia ADD CONSTRAINT PK_Wynagrodzenia PRIMARY KEY (Nr_wynagrodzenia)
/

-- Table and Columns comments section

COMMENT ON COLUMN Wynagrodzenia.Nr_wynagrodzenia IS 'Unikatowy identyfikator wynagrodzenia'

```



```

/
COMMENT ON COLUMN Wynagrodzenia.Data IS 'Data wypłaty wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Kwota_podstawowa IS 'Podstawowa kwota wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Kwota_dodatkowa IS 'Dodatkowa kwota wynagrodzenia'
/

-- Trigger for sequence Klub_TenisowySeq1 for column Nr_klubu in table Kluby_Tenisowe
-- -----
CREATE OR REPLACE TRIGGER ts_Kluby_Tenisowe_Klub_TenisowySeq1 BEFORE INSERT
ON Kluby_Tenisowe FOR EACH ROW
BEGIN
    :new.Nr_klubu := Klub_TenisowySeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Kluby_Tenisowe_Klub_TenisowySeq1 AFTER UPDATE OF Nr_klubu
ON Kluby_Tenisowe FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_klubu in table Kluby_Tenisowe as
    → it uses sequence. ');
END;
/

-- Trigger for sequence PracownikSeq1 for column Nr_pracownika in table Pracownicy
-- -----
CREATE OR REPLACE TRIGGER ts_Pracownicy_PracownikSeq1 BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN
    :new.Nr_pracownika := PracownikSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pracownicy_PracownikSeq1 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 ObiektSeq1 for column Nr_obiektu in table Obiekty -----
CREATE OR REPLACE TRIGGER ts_Obiekty_ObiektSeq1 BEFORE INSERT
ON Obiekty FOR EACH ROW
BEGIN
    :new.Nr_obiektu := ObiektSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Obiekty_ObiektSeq1 AFTER UPDATE OF Nr_obiektu
ON Obiekty FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_obiektu in table Obiekty as it
    → uses sequence. ');
END;
/

```

```

-- Trigger for sequence WydarzenieSeq1 for column Nr_wydarzenia in table Wydarzenia
↪ -----
CREATE OR REPLACE TRIGGER ts_Wydarzenia_WydarzenieSeq1 BEFORE INSERT
ON Wydarzenia FOR EACH ROW
BEGIN
    :new.Nr_wydarzenia := WydarzenieSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wydarzenia_WydarzenieSeq1 AFTER UPDATE OF Nr_wydarzenia
ON Wydarzenia FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wydarzenia in table Wydarzenia as
↪ it uses sequence. ');
END;
/

-- Trigger for sequence UslugaSeq1 for column Nr_uslugi in table Uslugi -----
CREATE OR REPLACE TRIGGER ts_Uslugi_UslugaSeq1 BEFORE INSERT
ON Uslugi FOR EACH ROW
BEGIN
    :new.Nr_uslugi := UslugaSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Uslugi_UslugaSeq1 AFTER UPDATE OF Nr_uslugi
ON Uslugi FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_uslugi in table Uslugi as it uses
↪ sequence. ');
END;
/

-- Trigger for sequence KlientSeq1 for column Nr_klienta in table Klienci -----
CREATE OR REPLACE TRIGGER ts_Klienci_KlientSeq1 BEFORE INSERT
ON Klienci FOR EACH ROW
BEGIN
    :new.Nr_klienta := KlientSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Klienci_KlientSeq1 AFTER UPDATE OF Nr_klienta
ON Klienci FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_klienta in table Klienci as it
↪ uses sequence. ');
END;
/

-- Trigger for sequence AdresSeq1 for column Nr_adresu in table Adresy -----
CREATE OR REPLACE TRIGGER ts_Adresy_AdresSeq1 BEFORE INSERT
ON Adresy FOR EACH ROW
BEGIN
    :new.Nr_adresu := AdresSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Adresy_AdresSeq1 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 WlascicielSeq1 for column Nr_wlasciciela in table Wlasciciele
↪ -----
CREATE OR REPLACE TRIGGER ts_Wlasciciele_WlascicielSeq1 BEFORE INSERT
ON Wlasciciele FOR EACH ROW
BEGIN
    :new.Nr_wlasciciela := WlascicielSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wlasciciele_WlascicielSeq1 AFTER UPDATE OF Nr_wlasciciela
ON Wlasciciele FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wlasciciela in table Wlasciciele
    ↪ as it uses sequence. ');
END;
/

-- Trigger for sequence PocztaSeq1 for column Nr_poczty in table Poczty -----
CREATE OR REPLACE TRIGGER ts_Poczty_PocztaSeq1 BEFORE INSERT
ON Poczty FOR EACH ROW
BEGIN
    :new.Nr_poczty := PocztaSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Poczty_PocztaSeq1 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 StanowiskoSeq1 for column Nr_stanowiska in table Stanowiska
↪ -----
CREATE OR REPLACE TRIGGER ts_Stalowiska_StalowiskoSeq1 BEFORE INSERT
ON Stanowiska FOR EACH ROW
BEGIN
    :new.Nr_stalowiska := StanowiskoSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Stalowiska_StalowiskoSeq1 AFTER UPDATE OF Nr_stalowiska
ON Stanowiska FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_stalowiska in table Stanowiska as
    ↪ it uses sequence. ');
END;
/

-- Trigger for sequence WynagrodzenieSeq1 for column Nr_wynagrodzenia in table Wynagrodzenia
↪ -----
CREATE OR REPLACE TRIGGER ts_Wynagrodzenia_WynagrodzenieSeq1 BEFORE INSERT
ON Wynagrodzenia FOR EACH ROW

```

```

BEGIN
    :new.Nr_wynagrodzenia := WynagrodzenieSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wynagrodzenia_WynagrodzenieSeq1 AFTER UPDATE OF
    → Nr_wynagrodzenia
ON Wynagrodzenia FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_wynagrodzenia in table
    → Wynagrodzenia as it uses sequence. ');
END;
/

-- Create foreign keys (relationships) section
→ -----

ALTER TABLE Pracownicy ADD CONSTRAINT Zatrudnia FOREIGN KEY (Nr_klubu) REFERENCES
    → Kluby_Tenisowe (Nr_klubu)
/

ALTER TABLE Obiekty ADD CONSTRAINT Posiada FOREIGN KEY (Nr_klubu) REFERENCES Kluby_Tenisowe
    → (Nr_klubu)
/

ALTER TABLE Uslugi ADD CONSTRAINT Oferuje FOREIGN KEY (Nr_klubu) REFERENCES Kluby_Tenisowe
    → (Nr_klubu)
/

ALTER TABLE Uslugi ADD CONSTRAINT Korzysta_z FOREIGN KEY (Nr_klienta) REFERENCES Klienci
    → (Nr_klienta)
/

ALTER TABLE Kluby_Tenisowe ADD CONSTRAINT Klub_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES
    → Adresy (Nr_adresu)
/

ALTER TABLE Wlasciciele ADD CONSTRAINT Wlasciciel_ma_adres FOREIGN KEY (Nr_adresu)
    → REFERENCES Adresy (Nr_adresu)
/

ALTER TABLE Wlasciciele ADD CONSTRAINT Klub_ma_wlasciciela FOREIGN KEY (Nr_klubu) REFERENCES
    → Kluby_Tenisowe (Nr_klubu)
/

```

```
ALTER TABLE Adresy ADD CONSTRAINT Adres_ma_poczta FOREIGN KEY (Nr_poczty) REFERENCES Poczty
↳ (Nr_poczty)
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES
↳ Adresy (Nr_adresu)
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_stanowisko FOREIGN KEY (Nr_stanowiska)
↳ REFERENCES Stanowiska (Nr_stanowiska)
/
```

```
ALTER TABLE Wynagrodzenia ADD CONSTRAINT Pracownik_otrzymuje_wynagrodzenie FOREIGN KEY
↳ (Nr_pracownika) REFERENCES Pracownicy (Nr_pracownika)
/
```

```
ALTER TABLE Obiekty ADD CONSTRAINT Obiekt_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy
↳ (Nr_adresu)
/
```

```
ALTER TABLE Klienci ADD CONSTRAINT Klient_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy
↳ (Nr_adresu)
/
```

5.6. Skrypt SQL uzupełniający bazę danych

```
INSERT INTO Poczty(kod_poczty,poczta)VALUES('05-320','Mrozy');
INSERT INTO Poczty(kod_poczty,poczta)VALUES('05-300','Mińsk Mazowiecki');
INSERT INTO Poczty (kod_poczty,poczta)VALUES('00-001','Warszawa');
INSERT INTO Poczty (kod_poczty,poczta)VALUES('00-002','Warszawa');
INSERT INTO Poczty (kod_poczty,poczta)VALUES('00-004','Warszawa');
INSERT INTO Poczty (kod_poczty,poczta)VALUES('04-218','Kraków');
INSERT INTO Poczty (kod_poczty,poczta)VALUES('30-000','Kraków');
INSERT INTO Poczty (kod_poczty,poczta)VALUES('60-001','Poznań');

INSERT INTO Adresy
↳ (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Poznań','Kremówkowa','21','8');
INSERT INTO Adresy (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Mińsk
↳ Mazowiecki','Warszawska','37','2');
INSERT INTO Adresy (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Mrozy','Willowa','55','1');
INSERT INTO Adresy (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Warszawa','Aleja Jana Pawła
↳ 2','2/13','3');
```

```

INSERT INTO Adresy
→ (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Warszawa','Grójecka','112','4');
INSERT INTO Adresy
→ (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Warszawa','Banacha','2/3','5');
INSERT INTO Adresy
→ (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Kraków','Wawelska','178','6');
INSERT INTO Adresy
→ (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Kraków','Podwawelska','213/7','7');
INSERT INTO Adresy (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Poznań','Mariacka','2','8');
INSERT INTO Adresy
→ (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Poznań','Koziolkowa','56','8');
INSERT INTO Adresy
→ (miasto,ulica,numer_lokalu,nr_poczty)VALUES('Warszawa','Marszałkowska','53','5');

INSERT INTO Kluby_Tenisowe(Nazwa,Data_zalozenia,Nr_adresu) VALUES('Masny
→ Klubik','1978-10-16','1');

INSERT INTO Wlasciciele(Imie,Nazwisko,Nr_adresu,Nr_klubu)
→ VALUES('Jerzy','Kornicki','2','1');
INSERT INTO Wlasciciele(Imie,Nazwisko,Nr_adresu,Nr_klubu)
→ VALUES('Daniel','Górnicki','3','1');
INSERT INTO Wlasciciele(Imie,Nazwisko,Nr_adresu,Nr_klubu)
→ VALUES('Daniel','Górzysty','5','1');

INSERT INTO Wydarzenia(Nazwa,Data_roz poczeczia_wydarzenia,Opis,Wpisowe) VALUES ('Turniej
→ Pucharu Sołtysa','2007-03-21','Turniej singlowy w formacie grupowym, zwycięzca zgarnia
→ puchar sołtysa, mecze rozgrywane będą od 21-03-2007 do momentu wyłonienia zwycięzcy,
→ mecze rozgrywane na różnych kortach, 5zł wpisowego ZAPRASZAMY',5.00);
INSERT INTO Wydarzenia(Nazwa,Data_roz poczeczia_wydarzenia,Opis,Data_zakonczenia_wydarzenia)
→ VALUES ('Turniej Pucharu Szkół Podstawowych w Janówku','2007-05-05','Turniej mikst w
→ formacie każdy z każdym, zwycięzca zgarnia puchar Szkół Podstawowych w Janówku, mecze
→ rozgrywane będą od 2007-05-05 do momentu wyłonienia zwycięzcy, mecze rozgrywane na
→ różnych kortach','2007-06-15');
INSERT INTO Wydarzenia(Nazwa,Data_roz poczeczia_wydarzenia,Opis,Wpisowe) VALUES ('Spotkanie z
→ Agnieszką Radwańską, nauka pokory w dążeniu do celu','2008-01-22','Spotkaj się z
→ Agnieszką Radwańską!, wypytaj ją o wszystko, naucz się grać jak profesjonalista, wstęp
→ płatny',65.00);
INSERT INTO Wydarzenia(Nazwa,Data_roz poczeczia_wydarzenia,Opis) VALUES ('Spotkanie z Michałem
→ Polem, nauka profesjonalnego dziennikarstwa sportowego','2007-06-15','Spotkaj się z
→ Michałem Polem, obecnie najbardziej rozpoznawalnym dziennikarzem sportowym!, przepytaj
→ go jak wyglądał jego wywiad z Ronaldinho lub innymi gwiazdami sportu światowego, wstęp
→ darmowy');
INSERT INTO Wydarzenia(Nazwa,Data_roz poczeczia_wydarzenia,Opis) VALUES ('Wspólny trening z
→ Jerzym Janowiczem na jednej z naszych hal','2007-08-24','Spotkaj się z topowym
→ tenisistą, naucz się jego sztuczek oraz zagrań, wstęp darmowy');

INSERT INTO Stanowiska(Nazwa_stanowiska,Opis_stanowiska) VALUES('Trener','Zajmuje się
→ trenowaniem swoich podopiecznych');
INSERT INTO Stanowiska(Nazwa_stanowiska,Opis_stanowiska) VALUES('Konserwator','Dogląda, aby
→ boiska były sprawne do użytku');
INSERT INTO Stanowiska(Nazwa_stanowiska,Opis_stanowiska) VALUES('Księgowy','Księguje
→ dokumenty i dogląda finansów klubu');
INSERT INTO Stanowiska(Nazwa_stanowiska,Opis_stanowiska) VALUES('Sędzia','Obsługuje mecze w
→ turnieju');

```

```

INSERT INTO Klienci(Imie,Nazwisko,Nr_Telefonu,E_mail,PESEL,Data_urodzenia,Plec,Nr_adresu)
→ VALUES('Karol','Górzysty','123456789','numer@wp.pl','07265726823','1998-02-24','K','6');
INSERT INTO Klienci(Imie,Nazwisko,Plec,Nr_adresu) VALUES('Karol','Górzysty','M','7');
INSERT INTO Klienci(Imie,Nazwisko,Nr_Telefonu,Plec,Nr_adresu)
→ VALUES('Karolina','Wojtyła','662812245','K','4');
INSERT INTO Klienci(Imie,Nazwisko,Nr_Telefonu,Data_urodzenia,Nr_adresu)
→ VALUES('Paweł','Pawelczak','665214823','1997-09-11','8');
INSERT INTO Klienci(Imie,Nazwisko,Nr_Telefonu,PESEL,Data_urodzenia,Nr_adresu)
→ VALUES('Szymon','Pawelczak','665214823','82462127582','1994-03-12','8');

INSERT INTO
→ Pracownicy(Imie,Nazwisko,Data_urodzenia,Plec,Data_zatrudnienia,Wazne_badania_lekarskie,
→ Nr_klubu,Nr_adresu,Nr_stanowiska)VALUES
→ ('Leszek','Wałęsa','1980-12-12','M','2000-01-15','1','1','9','1');
INSERT INTO
→ Pracownicy(Imie,Nazwisko,Data_urodzenia,PESEL,Plec,Data_zatrudnienia,Wazne_badania_lekarskie,
→ Nr_klubu,Nr_adresu,Nr_stanowiska)VALUES
→ ('Mateusz','Skowyrski','1978-04-25','12547257821','M','2002-03-05','1','1','9','2');
INSERT INTO
→ Pracownicy(Imie,Nazwisko,Data_urodzenia,PESEL,Plec,Data_zatrudnienia,Data_zwolnienia,
→ Wazne_badania_lekarskie,Nr_klubu,Nr_adresu,Nr_stanowiska)VALUES
→ ('Barbara','Krasucka','1977-03-22','87245812340','K','2002-03-05','2005-03-04','0','1','10','3');

INSERT INTO Wynagrodzenia(Data,Kwota_podstawowa,Kwota_dodatkowa,Nr_pracownika)
→ VALUES('2007-03-10',3000.25,300.45,'1');
INSERT INTO Wynagrodzenia(Data,Kwota_podstawowa,Nr_pracownika)
→ VALUES('2007-03-10',4500.54,'2');

INSERT INTO Turnieje
→ (Nr_wydarzenia,Poziom_turnieju,Typ_rozgrywek,Nagroda,System_rozgrywania_turnieju)
VALUES('1','Profesjonalista','singiel',200.00,'grupowy');
INSERT INTO Turnieje
→ (Nr_wydarzenia,Poziom_turnieju,Typ_rozgrywek,Nagroda,System_rozgrywania_turnieju)
VALUES('2','Open','mikst',500.00,'każdy z każdym');

INSERT INTO Obiekty(Status_obiektu,Nazwa,Typ_obiektu,Nr_klubu,Nr_adresu)
→ VALUES('Otwarty','Kort w Poznaniu im. Mateusza Borka','kort','1','1');
INSERT INTO Obiekty(Status_obiektu,Nazwa,Typ_obiektu,Nr_klubu,Nr_adresu)
→ VALUES('Otwarty','Hala w Warszawie im. Michała Pola','hala','1','5');
INSERT INTO Obiekty(Status_obiektu,Nazwa,Typ_obiektu,Nr_klubu,Nr_adresu)
→ VALUES('Otwarty','Kort w Warszawie im. Igi Świątek','kort','1','5');
INSERT INTO Obiekty(Status_obiektu,Nazwa,Typ_obiektu,Nr_klubu,Nr_adresu)
→ VALUES('Zamknięty','Magazyn sportowy sprzętu','magazyn','1','1');
INSERT INTO Obiekty(Status_obiektu,Nazwa,Typ_obiektu,Nr_klubu,Nr_adresu)
→ VALUES('Zamknięty','Wypożyczalnia sprzętu sportowego','wypożyczalnia','1','1');
INSERT INTO Obiekty(Status_obiektu,Nazwa,Typ_obiektu,Nr_klubu,Nr_adresu)
→ VALUES('Zamknięty','Biuro przedsiębiorstwa Masny Klubik','biuro','1','1');
INSERT INTO Obiekty(Status_obiektu,Nazwa,Typ_obiektu,Nr_klubu,Nr_adresu)
→ VALUES('Zamknięty','Hala w Warszawie im. Jerzego Janowicza','hala','1','11');
INSERT INTO Korty(Nr_obiektu,Nawierzchnia,Trybuny,Krzeselko_sedziowskie)
→ VALUES('1','Twarda','1','1');
INSERT INTO Korty(Nr_obiektu,Nawierzchnia,Trybuny,Krzeselko_sedziowskie)
→ VALUES('3','Ceglana','0','1');

```


INSERT INTO

```
→ Usługi(Nazwa,Data_opłacenia_usługi,Data_rozpoczęcia_usługi,Data_zakończenia_usługi,Cena,  
→ Typ_usługi,Nr_klubu,Nr_klienta) VALUES('Wynajem kortu numer  
→ 1','2004-07-12',TO_DATE('2004-07-12 10:25:14','yyyy-mm-dd  
→ hh24:mi:ss'),TO_DATE('2004-07-12 18:25:14','yyyy-mm-dd  
→ hh24:mi:ss'),123.00,'Wynajem','1','1');
```

INSERT INTO

```
→ Usługi(Nazwa,Data_opłacenia_usługi,Cena,Sprzet,Typ_usługi,Nr_klubu,Nr_klienta)VALUES('Zakup  
→ butów sportowych','2004-08-05',100.99,'Buty sportowe','Zakup','1','2');
```

INSERT INTO

```
→ Usługi(Nazwa,Data_opłacenia_usługi,Data_rozpoczęcia_usługi,Data_zakończenia_usługi,Cena,Sprzet,  
→ Typ_usługi,  
Nr_klubu,Nr_klienta) VALUES('Wypożyczenie rakiety  
→ tenisowej','2004-08-25',TO_DATE('2004-08-25 11:36:15','yyyy-mm-dd  
→ hh24:mi:ss'),TO_DATE('2004-07-25 12:36:15','yyyy-mm-dd hh24:mi:ss'),3.00,'Rakieta  
→ Wilson','Wypożyczenie','1','3');
```

INSERT INTO Usługa_Obiekt (Nr_usługi,Nr_obiektu)VALUES(1,8);

INSERT INTO Usługa_Obiekt (Nr_usługi,Nr_obiektu)VALUES(2,1);

INSERT INTO Usługa_Obiekt (Nr_usługi,Nr_obiektu)VALUES(3,5);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(1,1);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(2,2);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(3,3);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(4,4);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(5,5);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(2,1);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(3,1);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(4,1);

INSERT INTO Klient_Wydarzenie (Nr_klienta,Nr_wydarzenia)VALUES(5,1);

INSERT INTO Wydarzenie_Obiekt (Nr_wydarzenia,Nr_obiektu)VALUES(3,3);

INSERT INTO Wydarzenie_Obiekt (Nr_wydarzenia,Nr_obiektu)VALUES(4,2);

INSERT INTO Wydarzenie_Obiekt (Nr_wydarzenia,Nr_obiektu)VALUES(5,7);

INSERT INTO Wydarzenie_Pracownik (Nr_wydarzenia,Nr_pracownika)VALUES(1,1);

INSERT INTO Wydarzenie_Pracownik (Nr_wydarzenia,Nr_pracownika)VALUES(2,1);

INSERT INTO Wydarzenie_Pracownik (Nr_wydarzenia,Nr_pracownika)VALUES(3,2);

INSERT INTO Wydarzenie_Pracownik (Nr_wydarzenia,Nr_pracownika)VALUES(4,2);

INSERT INTO Wydarzenie_Pracownik (Nr_wydarzenia,Nr_pracownika)VALUES(5,1);

INSERT INTO Klub_Tenisowy_Wydarzenie (Nr_Klubu,Nr_wydarzenia)VALUES(1,1);

INSERT INTO Klub_Tenisowy_Wydarzenie (Nr_Klubu,Nr_wydarzenia)VALUES(1,2);

INSERT INTO Klub_Tenisowy_Wydarzenie (Nr_Klubu,Nr_wydarzenia)VALUES(1,3);

INSERT INTO Klub_Tenisowy_Wydarzenie (Nr_Klubu,Nr_wydarzenia)VALUES(1,4);

INSERT INTO Klub_Tenisowy_Wydarzenie (Nr_Klubu,Nr_wydarzenia)VALUES(1,5);

INSERT INTO Usługa_Pracownik (Nr_usługi,Nr_pracownika)VALUES(2,2);

INSERT INTO Pracownik_Obiekt (Nr_pracownika,Nr_obiektu)VALUES(2,2);

INSERT INTO Pracownik_Obiekt (Nr_pracownika,Nr_obiektu)VALUES(2,3);

INSERT INTO Pracownik_Obiekt (Nr_pracownika,Nr_obiektu)VALUES(2,7);

INSERT INTO Pracownik_Obiekt (Nr_pracownika,Nr_obiektu)VALUES(1,1);

[1]

[2]

[3]

Literatura

- [1] D. M. Kowalczyk. Projektowanie część 1, 2020. https://drive.google.com/file/d/1jA6d_T0uDswIyZjAxUYdupYyK9mVM_RG/view?usp=sharing).
- [2] D. M. Kowalczyk. Projektowanie część 2, 2020. <https://drive.google.com/file/d/1Ld13msESQxzX1H56svE4dZI7tZWiatrh/view?usp=sharing>).
- [3] D. M. Kowalczyk. Projektowanie część 3, 2020. <https://drive.google.com/file/d/1VJQrkpWymF9c9BwYJaSPtA6eGh6RZVhj/view?usp=sharing>).
- [4] Wikipedia. Normalizacja, 2018. [https://pl.wikipedia.org/wiki/PostaÅŹ_normalna_\(bazy_danych\)](https://pl.wikipedia.org/wiki/PostaÅŹ_normalna_(bazy_danych)).