

Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

## Wprowadzenie do baz danych (WBD)

### Sprawozdanie z projektu, część 1 Filharmonia

Katarzyna Dziewulska,  
Kinga Staszekiewicz

Warszawa, 2017

# Spis treści

<b>1. Zakres i cel projektu</b>	2
1.1. Założenia projektu	2
<b>2. Definicja systemu</b>	3
2.1. Możliwości systemu	3
2.2. Perspektywy użytkowników	3
<b>3. Model konceptualny</b>	4
3.1. Definicja zbiorów encji określonych w projekcie	4
3.2. Ustalenie związków i ich typów między encjami	5
3.3. Określenie atrybutów i ich dziedzin	5
3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)	8
3.5. Klucze kandydujące i główne	8
3.6. Problem pułapek szczelinowych i wachlarzowych	8
3.7. Schemat ER na poziomie konceptualnym	9
<b>4. Model relacyjny</b>	11
4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym	11
4.2. Proces normalizacji	11
4.3. Proces denormalizacji	13
4.4. Model fizyczny	13
<b>5. Model fizyczny</b>	15
5.1. Projekt transakcji i weryfikacja ich wykonalności	15
5.2. Skrypt SQL	16
5.3. Sekwencje	25
5.4. Strojanie bazy danych - dobór indeksów	26
5.5. Wypełnienie bazy danych danymi	27
5.6. Przykłady zapytań poleceń SQL odnoszących się do bazy	32

# 1. Zakres i cel projektu

Projekt ma na celu usprawnienie działania filharmonii dzięki zaprojektowaniu bazy danych na poziomie konceptualnym, logicznym oraz jej fizycznej implementacji. Baza danych będzie oparta na rozwiązaniach firmy Oracle, a zaimplementowana zostanie w języku SQL.

W celu realizacji projektu wykorzystane zostanie następujące oprogramowanie:

- TOAD Data Modeler - stworzenie modelu konceptualnego i logicznego
- Oracl Database 11g - silnik zarządzania bazą danych
- Oracle SQL Developer -implementacja bazy danych

## 1.1. Założenia projektu

Realizowany projekt dotyczy **filharmonii**. Filahrmonia zajmuje się organizowaniem **koncertów**, które odbywają się w **salach koncertowych** filharmonii. **Bilety** na koncerty sprzedawane są w **kasach biletowych**. Przed koncertem mogą być organizowane **próby** w **salach do ćwiczeń**, na których **muzycy** ćwiczą przed występem. Filharmonia posiada pewną liczbę **instrumentów**, które od czasu do czasu mogą przechodzić **naprawy**. W bazie przechowywane są dane **pracowników** zatrudnionych w filharmonii oraz **wynagrodzenie**, jakie otrzymują.

## 2. Definicja systemu

### 2.1. Możliwości systemu

System został zaprojektowany w taki sposób, aby zapewnić następujące możliwości:

- dodawanie, modyfikowanie, odczyt informacji o filharmonii
- dodawanie, modyfikowanie, odczyt informacji o pracownikach
- dodawanie, modyfikowanie, odczyt informacji o instrumentach
- dodawanie, modyfikowanie, odczyt informacji o koncertach
- dodawanie, modyfikowanie, odczyt informacji o salach
- dodawanie, modyfikowanie, odczyt informacji o próbach
- dodawanie, modyfikowanie, odczyt informacji o biletach
- dodawanie, modyfikowanie, odczyt informacji o wynagrodzeniach

Dostępne opcje zależą od uprawnień użytkownika.

### 2.2. Perspektywy użytkowników

System umożliwia dostęp do danych z poziomu pięciu perspektyw użytkownika:

- **administrator systemu** - posiada dostęp do wszystkich danych, może dodawać, usuwać, aktualizować i modyfikować dane;
- **kierownik artystyczny** - posiada możliwość modyfikacji danych związanych z encjami Koncert, Utwory, Muzycy, Próby, Sale, Instrumenty;
- **dyrektor** - posiada dostęp do wszystkich danych, lecz nie może nimi manipulować bezpośrednio, ale na polecenie dyrektora dane mogą być modyfikowane przez administratora;
- **muzyk** - ma wgląd do danych związanych z encjami Koncert, Instrument, Naprawa Instrumentu, Próba, Sala, Wynagrodzenie (tylko swoje);
- **księgowa** - posiada możliwość modyfikacji danych dotyczących Wynagrodzenia;
- **kasjer** - może modyfikować dane związane z encją Bilet;

### 3. Model konceptualny

#### 3.1. Definicja zbiorów encji określonych w projekcie

1. **Filharmonia** - instancja filharmonii, zawiera podstawowe dane o filharmonii, takie jak nazwa, adres, numer telefonu, adres email, dyrektor, rok założenia.
2. **Pracownik** - instancja pracownika filharmonii, określana przez jego imię, nazwisko, płeć, datę urodzenia, datę zatrudnienia, PESEL, numer telefonu oraz adres email.
3. **Muzyk** - specjalizacja encji Pracownik, charakteryzowana dodatkowo przez wykształcenie, specjalność główną, specjalność dodatkową oraz informację o tym, czy muzyk jest solistą, kameralistą, muzykiem orkiestrowym lub chórzystą.
4. **Wynagrodzenie** - instancja wynagrodzenia, określana poprzez jego wartość, datę wypłaty oraz wartość premii.
5. **Instrument** - instancja instrumentu będącego własnością filharmonii, charakteryzowana przez nazwę, rodzinę instrumentów, do której należy, rok produkcji, cenę, markę oraz model.
6. **Naprawa instrumentu** - instancja naprawy instrumentu, określana przez jej datę, koszt oraz firmę wykonującą naprawę.
7. **Koncert** - instancja koncertu, definiowana przez datę i godzinę jego rozpoczęcia, czas trwania oraz repertuar.
8. **Sala** - instancja sali, charakteryzowana przez jej numer oraz numer piętra, na którym się znajduje.
9. **Sala koncertowa** - specjalizacja encji Sala, określana dodatkowo przez liczbę miejsc, wielkość sceny oraz informację o obecności łoża VIP.
10. **Sala do ćwiczeń** - specjalizacja encji Sala, definiowana dodatkowo przez swoją wielkość oraz informację o obecności fortepianu.
11. **Próba** - instancja próby, charakteryzowana przez datę i godzinę rozpoczęcia, czas jej trwania oraz jej rodzaj.
12. **Bilet** - instancja biletu, określana przez jego cenę, numer miejsca, numer rzędu oraz informację o tym, czy został zarezerwowany i opłacony.
13. **Kasa biletowa** - instancja kasy, definiowana przez swój numer, model, producenta, markę oraz numer homologacji.

### 3.2. Ustalenie związków i ich typów między encjami

Encja nr 1	Nazwa związku	Encja nr 2	Stopień	Typ	Uczestnictwo
Filarmonia	organizuje	Koncert	binarny	1:n	obowiązkowy-opcjonalny
Filarmonia	zatrudnia	Pracownik	binarny	1:n	obowiązkowy-opcjonalny
Filarmonia	dysponuje	Sala	binarny	1:n	obowiązkowy-obowiązkowy
Filarmonia	jest właścicielem	Instrument	binarny	1:n	obowiązkowy-opcjonalny
Filarmonia	posiada	Kasa biletowa	binarny	1:n	obowiązkowy-opcjonalny
Pracownik	pobiera	Wynagrodzenie	binarny	1:n	obowiązkowy-opcjonalny
Pracownik	odbywa	Próba	binarny	n:m	obowiązkowy-opcjonalny
Pracownik	bierze udział w	Koncert	binarny	n:m	opcjonalny-opcjonalny
Pracownik	używa	Instrument	binarny	n:m	opcjonalny-opcjonalny
Koncert	odbywa się w	Sala	binarny	n:1	opcjonalny-obowiązkowy
Instrument	przechodzi	Naprawa instrumentu	binarny	1:n	obowiązkowy-opcjonalny
Sala	jest miejscem	Próba	binarny	1:n	obowiązkowy-opcjonalny
Bilet	na	Koncert	binarny	n:1	obowiązkowy-obowiązkowy
Kasa biletowa	sprzedaje	Bilet	binarny	1:n	obowiązkowy-opcjonalny
Kasa biletowa	jest obsługiwana przez	Pracownik	binarny	n:m	opcjonalny-opcjonalny

Filharmonia może organizować wiele koncertów, zatrudniać pewną liczbę pracowników, dysponować wieloma salami, być właścicielem wielu instrumentów oraz posiadać wiele kas biletowych. Każdy pracownik może pobierać wiele wynagrodzeń, odbywać wiele prób oraz używać wielu instrumentów. W każdej próbie może brać udział wielu pracowników, jednak aby próba się odbyła, musi wziąć w niej udział przynajmniej jeden pracownik. Każdy instrument może być używany przez wielu pracowników. W każdej sali może odbywać się wiele koncertów. Instrument może przechodzić wiele napraw. Sala może być miejscem wielu prób. Na koncert może być wiele biletów. Kasa biletowa może sprzedawać wiele biletów oraz być obsługiwana przez wielu pracowników. Każdy pracownik może obsługiwać wiele kas biletowych.

### 3.3. Określenie atrybutów i ich dziedzin

Tab. 3.1: Filharmonia

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id filharmonii	Integer	Tak	Tak	klucz główny encji
Nazwa	Varchar(20)	Tak	Tak	
Adres	Varchar(200)	Tak	Nie	pole segmentowe
Numer telefonu	Varchar(15)	Tak	Nie	
Email	Varchar(100)	Tak	Nie	
Dyrektor	Varchar(100)	Tak	Nie	pole wielowartościowe
Rok założenia	Char(4)	Tak	Nie	

Tab. 3.2: Pracownik

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id pracownika	Integer	Tak	Tak	klucz główny encji
Imie	Varchar(30)	Tak	Nie	
Nazwisko	Varchar(50)	Tak	Nie	
Plec	Char(1)	Tak	Nie	K/M
Data urodzenia	Date	Tak	Nie	
Data zatrudnienia	Date	Tak	Nie	
PESEL	Char(11)	Nie	Tak	
Numer telefonu	Varchar(15)	Nie	Nie	
Email	Varchar(100)	Nie	Nie	

Tab. 3.3: Wynagrodzenie

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id wynagrodzenia	Integer	Tak	Tak	klucz główny encji
Wartosc wynagrodzenia	Float(24)	Tak	Nie	w PLN
Data wypłaty	Date	Tak	Nie	
Premia	Float(24)	Nie	Nie	w PLN

Tab. 3.4: Instrument

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id instrumentu	Integer	Tak	Tak	klucz główny encji
Nazwa instrumentu	Varchar(30)	Tak	Nie	
Rodzina instrumentu	Varchar(20)	Tak	Nie	smyczkowe/dete blaszane/ dete drewniane/perkusyjne/ inne
Rok produkcji	Char(4)	Nie	Nie	
Cena instrumentu	Float(24)	Nie	Nie	w PLN
Marka	Varchar(30)	Nie	Nie	
Model	Varchar(30)	Nie	Nie	

Tab. 3.5: Naprawa instrumentu

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id naprawy instrumentu	Integer	Tak	Tak	klucz główny encji
Data naprawy	Date	Tak	Nie	
Koszt naprawy	Float(24)	Tak	Nie	w PLN
Firma wykonawcza	Varchar(30)	Tak	Nie	

Tab. 3.6: Koncert

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id koncertu	Integer	Tak	Tak	klucz główny encji
Data koncertu	Date	Tak	Nie	
Godzina koncertu	Char(5)	Tak	Nie	w formacie HH::MM
Czas trwania koncertu	SmallInt	Tak	Nie	w minutach
Repertuar koncertu	Varchar(200)	Tak	Nie	pole segmentowe i wielowartościowe

Tab. 3.7: Sala

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id sali	Integer	Tak	Tak	klucz główny encji
Numer sali	Varchar(4)	Tak	Tak	
Numer piętra	SmallInt	Nie	Nie	

Tab. 3.8: Sala koncertowa - specjalizacja Sali

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Liczba miejsc	SmallInt	Tak	Nie	
Wielkość sceny	Varchar(4)	Tak	Nie	M/S/D
Loża VIP	Char(1)	Tak	Nie	T/N

Tab. 3.9: Sala do ćwiczeń - specjalizacja Sali

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Wielkość sali	Char(1)	Tak	Nie	M/S/D
Obecność fortepianu	Char(1)	Nie	Nie	T/N

Tab. 3.10: Próba

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id próby	Integer	Tak	Tak	klucz główny encji
Data próby	Date	Tak	Nie	
Godzina próby	Char(5)	Tak	Nie	w formacie HH:MM
Czas trwania	SmallInt	Tak	Nie	w minutach
Rodzaj próby	Varchar(30)	Tak	Nie	indywidualna/sekcyjna całościowa/generalna

Tab. 3.11: Bilet

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id biletu	Integer	Tak	Tak	klucz główny encji
Cena biletu	Float(24)	Tak	Nie	w PLN
Numer miejsca	SmallInt	Tak	Nie	
Numer rzędu	SmallInt	Tak	Nie	
Zarezerwowany	Char(1)	Tak	Nie	T/N
Oplacony	Char(1)	Tak	Nie	T/N

Tab. 3.12: Kasa biletowa

Nazwa atrybutu	Dziedzina	Obowiązkowość	Unikatowość	Uwagi
Id kasy	Integer	Tak	Tak	klucz główny encji
Numer kasy	SmallInt	Tak	Tak	
Model kasy	VarChar(30)	Tak	Nie	
Marka kasy	VarChar(30)	Tak	Nie	
Numer homologacji kasy	VarChar(30)	Tak	Tak	



### 3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)

W zaprojektowanym systemie występują następujące dodatkowe reguły:

- dla atrybutu 'Plec' w encji 'Pracownik' występuje ograniczenie **K**(obieta)/**M**(ęczyzna)
- dla atrybutów 'Zarezerwowany' i 'Oplacony' w encji 'Bilet' oraz 'Loza VIP' i 'Obecnosc fortepianu' w specjalizacjach 'Sala koncertowa' i 'Sala do ćwiczeń' encji 'Sala' występuje ograniczenie **T**(ak)/**N**(ie)
- dla atrybutów 'Wielkosc sceny' i 'Wielkosc sali' w specjalizacjach 'Sala koncertowa' i 'Sala do ćwiczeń' encji 'Sala' występuje ograniczenie **M**(ała)/**S**(rednia)/**D**(uża).
- dla atrybutu 'Rodzina instrumentu' w encji 'Instrument' występuje ograniczenie na możliwość wpisania jednej z rodzin: **smyczkowe/dete blaszane/dete drewniane/perkusyjne/inne**
- dla atrybutu 'Rodzaj proby' w encji 'Proba' występuje ograniczenie na możliwość wpisania jednego z rodzajow: **indywidualna/sekcyjna/calosciowa/generalna**
- dla atrybutów 'Godzina koncertu' i 'Godzina proby' w encjach 'Koncert' i 'Proba' występuje ograniczenie na wpisanie godziny w formacie **HH:MM**

### 3.5. Klucze kandydujące i główne

Tabela (3.13) przedstawia potencjalne klucze kandydujące dla poszczególnych encji. Klucze główne nie zostały jednak wybrane spośród nich. W celu zapewnienia unikatowości kluczy, stworzono klucze sztuczne według reguły 'Id (nazwa encji)'. Dzięki takiemu zabiegowi baza danych została zabezpieczona przed zagrożeniami wynikającymi z braku integralności i unikalności danych, które mogłyby wynikać w przypadku wybrania innych atrybutów jako kluczy głównych.

Tab. 3.13: Klucze kandydujące

Nazwa encji	Klucz kandydujący
Filharmonia	Nazwa
Pracownik	PESEL
Wynagrodzenie	-
Instrument	-
Naprawa instrumentu	-
Koncert	-
Sala	Numer sali
Próba	-
Bilet	-
Kasa biletowa	Numer kasy, Numer homologacji kasy

### 3.6. Problem pułapek szczelinowych i wachlarzowych

Potencjalne pułapki szczelinowe:

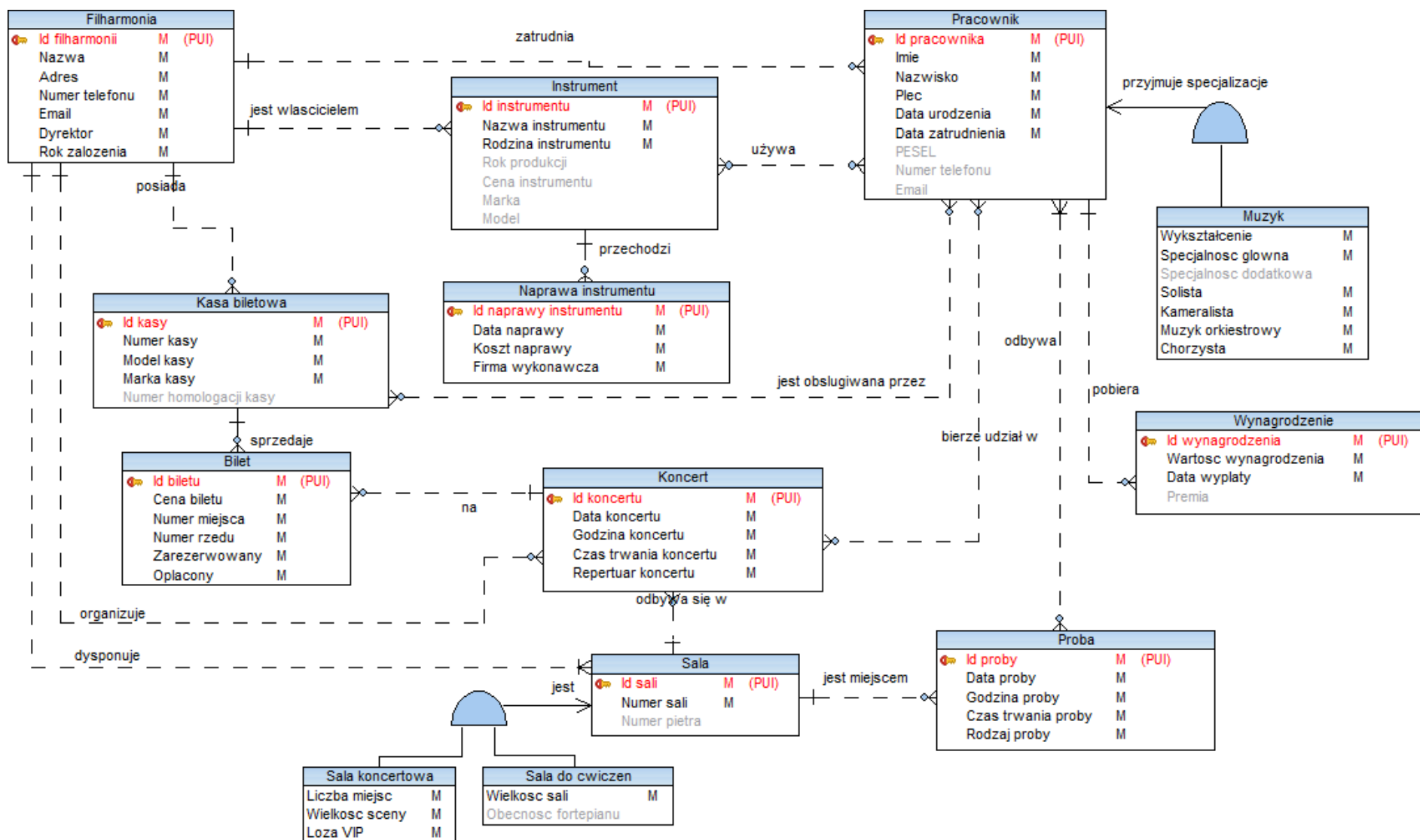
- Związek pomiędzy encją Pracownik a encją Koncert - można utworzyć koncert bez przypisania do niego żadnego pracownika, co może nie być działaniem pożądanym. Z drugiej strony w koncercie mogą brać udział wykonawcy gościnni.
- Związek pomiędzy encją Kasa biletowa a encją Pracownik - do danej kasy nie musi być przypisany żaden obsługujący ją pracownik, jednak kasa mogła być dopiero co zakupiona i nie musi być czynna.

Potencjalne pułapki wachlarzowe:

- Gdyby nie związek pomiędzy encją Instrument a encją Pracownik pojawiłaby się pułapka wachlarzowa, gdyż nie byłibyśmy w stanie powiedzieć jakiego instrumentu używa dany pracownik.

### **3.7. Schemat ER na poziomie konceptualnym**

Schemat ER na poziomie konceptualnym został przedstawiony na rysunku (3.1).



Rys. 3.1: Schemat ER na poziomie koncepcyjnym

## 4. Model relacyjny

W ramach przejścia z modelu konceptualnego do relacyjnego usunięto niekompatybilności związane z występowaniem związków wiele do wielu. W praktyce wiązało się to z zamianą każdego takiego związku na dwa związki jeden do wielu z wykorzystaniem tablicy brydzącej. Zmieniono także nazwy encji z liczby pojedynczej na mnogą. Dzięki wcześniejszemu zaplanowaniu wykorzystania kluczy sztucznych rozwiązano problem wyboru kluczy głównych. W związkach typu jeden do wielu dodano klucz obcy po stronie 'wiele'. W modelu konceptualnym nie występowały związki rekurencyjne ani o stopniu wyższym niż drugi.

### 4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym

W modelu konceptualnym występowały 4 związki wiele do wielu. Dla każdego z nich stworzono tablicę brydzącą. Jednak z powodu na ograniczenie liczby relacji w modelu logicznym zdecydowano się na usunięcie relacji 'Naprawa instrumentów' i 'Kasy biletowe' oraz powstałej tablicy brydzącej łączącej 'Kasy biletowe' z 'Pracownikami'. Dzięki takiemu zabiegowi liczba relacji nie przekracza 20. Powstałe tablice brydzące wraz z ich atrybutami przedstawiono poniżej.

Tab. 4.1: Uzytkowanie (tablica brydząca związku Pracownicy-Instrumenty)

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id instrumentu	Integer	Tak	Nie	PF KEY
Id pracownika	Integer	Tak	Nie	PF KEY

Tab. 4.2: Cwiczenie (tablica brydząca związku Pracownicy-Proby)

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id pracownika	Integer	Tak	Nie	PF KEY
Id proby	Integer	Tak	Nie	PF KEY

Tab. 4.3: Wystepowanie (tablica brydząca związku Pracownicy-Koncerty)

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id pracownika	Integer	Tak	Nie	PF KEY
Id koncertu	Integer	Tak	Nie	PF KEY

### 4.2. Proces normalizacji

W celu uzyskania III postaci normalnej modelu pola segmentowe zamieniono na kilka atrybutów w ramach relacji, natomiast pola wielowartościowe zamieniono na osobne relacje. Dodano także dwie relacje słownikowe. Stwierdzono także, że każdy atrybut relacji nie wchodzący w

skład żadnego klucza kandydującego jest w pełni funkcyjnie zależny od wszystkich kluczy kandydujących relacji. Ponadto wszystkie atrybuty niewchodzące w skład klucza głównego danej relacji są funkcyjnie zależne od klucza, całego klucza i tylko klucza.

Tab. 4.4: Relacja Filharmonie po pozbyciu się pól segmentowych i wielowartościowych

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id filharmonii	Integer	Tak	Tak	PK
Nazwa	Varchar2(20)	Tak	Tak	
Ulica	Varchar2(30)	Tak	Nie	
Nr lokalu	Varchar2(6)	Tak	Nie	
Miasto	Varchar2(30)	Tak	Nie	
Numer telefonu	Varchar2(15)	Tak	Nie	
Email	Varchar2(100)	Tak	Nie	
Rok założenia	Char(4)	Tak	Nie	
Id poczty	Integer	Tak	Nie	

Tab. 4.5: Nowopowstała relacja Poczty - relacja słownikowa

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id poczty	Integer	Tak	Tak	PK
Kod	Char(6)	Tak	Tak	
Miejscowosc	Varchar2(30)	Tak	Nie	

Tab. 4.6: Nowopowstała relacja Dyrektorzy

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id dyrektora	Integer	Tak	Nie	
Imie	Varchar2(30)	Tak	Tak	
Nazwisko	Varchar2(50)	Tak	Nie	

Tab. 4.7: Nowopowstała relacja Zarządzanie (tablica brydżująca związku Filharmonie-Dyrektorzy)

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id dyrektora	Integer	Tak	Nie	PF KEY
Id filharmonii	Integer	Tak	Nie	PF KEY
Data od	Date	Tak	Nie	PK
Data do	Date	Nie	Nie	

Tab. 4.8: Nowopowstała relacja Rodziny instrumentow - relacja słownikowa

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id rodziny	Integer	Tak	Tak	PK
Nazwa rodziny	Varchar2(30)	Tak	Nie	

Tab. 4.9: Nowopowstała relacja Utwory - relacja słownikowa

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id utworu	Integer	Tak	Tak	PK
Nazwa utworu	Varchar2(100)	Tak	Nie	
Imię kompozytora	Varchar2(30)	Tak	Nie	
Nazwisko kompozytora	Varchar2(50)	Tak	Nie	
Czas trwania utworu	Char(8)	Tak	Nie	HH:MM:SS

Tab. 4.10: Nowopowstała relacja Wykonywanie (tablica brydżująca związku Koncerty-Utwory)

Nazwa atrybutu	Dziedzina	Not Null	Unikatowość	Uwagi
Id koncertu	Integer	Tak	Nie	PF KEY
Id Utworu	Integer	Tak	Nie	PF KEY

### 4.3. Proces denormalizacji

Analizując otrzymany model, zauważono iż można by usprawnić działanie systemu, a konkretnie przyspieszyć wykonywanie niektórych zapytań poprzez częściową denormalizację. Przykładowe operacje denormalizujące model:

1. Wprowadzenie powtarzających się grup poprzez usunięcie relacji Poczty i zastąpienie jej atrybutami w relacji Filharmonie.
2. Usunięcie związku jeden do jeden, czyli specjalizacji Muzycy poprzez wcielenie jej atrybutów do relacji Pracownicy.
3. Wprowadzenie cyklu, tzn. dodanie związku jeden do wielu pomiędzy relacjami Filharmonie i Wynagrodzenia.

**Ad.1** Dzięki takiemu zabiegowi nie trzeba wykonywać złączenia relacji w celu uzyskania informacji.

**Ad.2** Połączenie tych dwóch relacji byłoby kłopotliwe, ponieważ trzeba by zrezygnować z obowiązkowości atrybutów należących wcześniej do specjalizacji Muzycy. W związku z tym można by dodać pracownika będącego muzykiem nie uzupełniając wszystkich wymaganych informacji. Takie zachowanie jest niepożądane dla działania systemu.

**Ad.3** Poprzez dodanie klucza obcego w tabeli Wynagrodzenia możliwe jest bezpośrednie uzyskanie informacji o wszystkich wynagrodzeniach (np. suma miesięcznych wypłat dla danej filharmonii) bez konieczności łączenia relacji Pracownicy.

Po przeanalizowaniu modelu stwierdzono, iż z powodu małej liczby rekordów denormalizacja nie przyczyni się do znacznego polepszenia działania systemu, a jej brak zapewni lepszą spójność danych.

### 4.4. Model fizyczny

Schemat modelu na poziomie logicznym został przedstawiony na rysunku (4.1).



## 5. Model fizyczny

### 5.1. Projekt transakcji i weryfikacja ich wykonalności

Poniższa tabela przedstawia przykładowe transakcje wraz z określeniem ich wykonalności.

<b>Id.</b>	<b>Nazwa transakcji</b>	<b>Wykorzystywane zasoby</b>	<b>Wyk.</b>	<b>Uwagi</b>
1	Dodawanie informacji o filharmonii	Filharmonie, Poczty	Tak	-
2	Modyfikowanie informacji o filharmonii	Filharmonie, Poczty	Tak	-
3	Odczyt informacji o filharmonii	Filharmonie, Poczty	Tak	-
4	Dodawanie informacji o pracownikach	Pracownicy, Filharmonie, Muzycy	Tak	-
5	Modyfikowanie informacji o pracownikach	Pracownicy, Filharmonie, Muzycy	Tak	-
6	Odczyt informacji o pracownikach	Pracownicy, Filharmonie, Muzycy	Tak	-
7	Dodawanie informacji o instrumentach	Instrumenty, Rodziny instrumentów, Naprawy instrumentów, Filharmonie	Tak/Nie	Z uwagi na ograniczenie liczby tabel w projekcie usunięto relację 'Naprawy instrumentów'.
8	Modyfikowanie informacji o instrumentach	Instrumenty, Rodziny instrumentów, Naprawy instrumentów, Filharmonie	Tak/Nie	Z uwagi na ograniczenie liczby tabel w projekcie usunięto relację 'Naprawy instrumentów'.
9	Odczyt informacji o instrumentach	Instrumenty, Rodziny instrumentów, Naprawy instrumentów, Filharmonie	Tak/Nie	Z uwagi na ograniczenie liczby tabel w projekcie usunięto relację 'Naprawy instrumentów'.
10	Dodawanie informacji o koncertach	Koncerty, Sale, Wystepowanie, Pracownicy, Filharmonie	Tak	-
11	Modyfikowanie informacji o koncertach	Koncerty, Sale, Wystepowanie, Pracownicy, Filharmonie	Tak	-
12	Odczyt informacji o koncertach	Koncerty, Sale, Wystepowanie, Pracownicy, Filharmonie	Tak	-
13	Dodawanie informacji o salach	Sale, Sale koncertowe, Sale do ćwiczeń, Filharmonie	Tak	-
14	Modyfikowanie informacji o salach	Sale, Sale koncertowe, Sale do ćwiczeń, Filharmonie	Tak	-
15	Odczyt informacji o salach	Sale, Sale koncertowe, Sale do ćwiczeń, Filharmonie	Tak	-
16	Dodawanie informacji o próbach	Próby, Sale, Ćwiczenie, Pracownicy	Tak	-



17	Modyfikowanie informacji o próbach	Próby, Sale, Ćwiczenie, Pracownicy	Tak	-
18	Odczyt informacji o próbach	Próby, Sale, Ćwiczenie, Pracownicy	Tak	-
19	Dodawanie informacji o biletach	Bilety, Koncerty, Kasy biletowe	Tak/Nie	Z uwagi na ograniczenie liczby tabel w projekcie usunięto relację 'Kasy biletowe'.
20	Modyfikowanie informacji o biletach	Bilety, Koncerty, Kasy biletowe	Tak/Nie	Z uwagi na ograniczenie liczby tabel w projekcie usunięto relację 'Kasy biletowe'.
21	Odczyt informacji o biletach	Bilety, Koncerty, Kasy biletowe	Tak/Nie	Z uwagi na ograniczenie liczby tabel w projekcie usunięto relację 'Kasy biletowe'.
22	Dodawanie informacji o wynagrodzeniach	Wynagrodzenia, Pracownicy	Tak	-
23	Modyfikowanie informacji o wynagrodzeniach	Wynagrodzenia, Pracownicy	Tak	-
23	Odczyt informacji o wynagrodzeniach	Wynagrodzenia, Pracownicy	Tak	-

## 5.2. Skrypt SQL

Poniżej przedstawiono podstawową wersję skryptu SQL bazy danych wygenerowaną na podstawie modelu logicznego z kilkoma poprawkami.

```

/*
Created: 2017-11-29
Modified: 2017-12-02
Model: Model_logiczny_v1
Database: Oracle 11g Release 2
*/

-- Create tables section --

-- Table Filharmonie

CREATE TABLE Filharmonie(
  Id_filharmonii Integer NOT NULL,
  Nazwa Varchar2(100 ) NOT NULL,
  Ulica Char(30 ) NOT NULL,
  Numer_lokalu Varchar2(6 ) NOT NULL,
  Miasto Varchar2(30 ) NOT NULL,
  Numer_telefonu Varchar2(15 ) NOT NULL,
  Email Varchar2(100 ) NOT NULL,
  Rok_zalozenia Char(4 ) NOT NULL,
  Id_poczty Integer NOT NULL

```

```
)  
/  
  
— Create indexes for table Filharmonie  
  
CREATE INDEX IX_Relationship6 ON Filharmonie (Id_poczty)  
/  
  
— Add keys for table Filharmonie  
  
ALTER TABLE Filharmonie ADD CONSTRAINT Filharmonia_PK  
                    PRIMARY KEY (Id_filharmonii)  
/  
  
— Table Pracownicy  
  
CREATE TABLE Pracownicy(  
  Id_pracownika Integer NOT NULL,  
  Imie Varchar2(30 ) NOT NULL,  
  Nazwisko Varchar2(50 ) NOT NULL,  
  Plec Char(1 ) CHECK(Plec IN( 'M', 'K')) NOT NULL,  
  Data_urodzenia Date NOT NULL,  
  Data_zatrudnienia Date NOT NULL,  
  PESEL Char(11 ),  
  Numer_telefonu Varchar2(15 ),  
  Email Varchar2(100 ),  
  Id_filharmonii Integer NOT NULL  
)  
/  
  
— Create indexes for table Pracownicy  
  
CREATE INDEX IX_zatrudnia ON Pracownicy (Id_filharmonii)  
/  
  
— Add keys for table Pracownicy  
  
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownicy_PK  
                    PRIMARY KEY (Id_pracownika)  
/  
  
— Table Instrumenty  
  
CREATE TABLE Instrumenty(  
  Id_instrumentu Integer NOT NULL,  
  Nazwa_instrumentu Varchar2(30 ) NOT NULL,  
  Rok_produkcji Char(4 ),  
  Cena_instrumentu Float(24),  
  Marka Varchar2(30 ),  
  Model Varchar2(30 ),  
  Id_filharmonii Integer NOT NULL,
```

```
Id_rodziny Integer NOT NULL
)
/

— Create indexes for table Instrumenty

CREATE INDEX IX_jest_wlascicielem ON Instrumenty (Id_filharmonii)
/

CREATE INDEX IX_Relationship7 ON Instrumenty (Id_rodziny)
/

— Add keys for table Instrumenty

ALTER TABLE Instrumenty ADD CONSTRAINT Instrumenty_PK
PRIMARY KEY (Id_instrumentu)
/

— Table Koncerty

CREATE TABLE Koncerty(
Id_koncertu Integer NOT NULL,
Data_koncertu Date NOT NULL,
Godzina_koncertu Char(5) Check(Godzina_koncertu LIKE '__:__')NOT NULL,
Czas_trwania_koncertu Integer NOT NULL,
Id_filharmonii Integer NOT NULL,
Id_sali Integer NOT NULL
)
/

— Create indexes for table Koncerty

CREATE INDEX IX_organizuje ON Koncerty (Id_filharmonii)
/

CREATE INDEX IX_odbywa_się_w ON Koncerty (Id_sali)
/

— Add keys for table Koncerty

ALTER TABLE Koncerty ADD CONSTRAINT Koncerty_PK
PRIMARY KEY (Id_koncertu)
/

— Table Sale

CREATE TABLE Sale(
Id_sali Integer NOT NULL,
Numer_sali Varchar2(4 ) NOT NULL,
Numer_pietra Integer ,
Id_filharmonii Integer NOT NULL
```

```
)  
/  
  
— Create indexes for table Sale  
  
CREATE INDEX IX_dysponuje ON Sale (Id_filharmonii)  
/  
  
— Add keys for table Sale  
  
ALTER TABLE Sale ADD CONSTRAINT Sale_PK PRIMARY KEY (Id_sali)  
/  
  
— Table Muzycy  
  
CREATE TABLE Muzycy(  
  Id_pracownika Integer NOT NULL,  
  Stopien_wyksztalcenia Char(1 )  
    CHECK(Stopien_wyksztalcenia IN( '1', '2', '3', '4')) NOT NULL,  
  Nazwa_uczelni Varchar2(100 ) NOT NULL,  
  Specjalnosc_glowna Varchar2(30 ) NOT NULL,  
  Specjalnosc_dodatkowa Varchar2(30 ),  
  Solista Varchar2(1 ) CHECK(Solista IN( 'T', 'N')) NOT NULL,  
  Kameralista Varchar2(1 ) CHECK(Kameralista IN( 'T', 'N')) NOT NULL,  
  Muzyk_orkiestrowy Varchar2(1 )  
    CHECK(Muzyk_orkiestrowy IN( 'T', 'N')) NOT NULL,  
  Chorzysty Varchar2(1 ) CHECK(Chorzysty IN( 'T', 'N')) NOT NULL  
)  
/  
  
— Add keys for table Muzycy  
  
ALTER TABLE Muzycy ADD CONSTRAINT Muzycy_PK PRIMARY KEY (Id_pracownika)  
/  
  
— Table Wynagrodzenia  
  
CREATE TABLE Wynagrodzenia(  
  Id_wynagrodzenia Integer NOT NULL,  
  Wartosc_wynagrodzenia Float(24) NOT NULL,  
  Data_wyplaty Date NOT NULL,  
  Premia Float(24),  
  Id_pracownika Integer NOT NULL  
)  
/  
  
— Create indexes for table Wynagrodzenia  
  
CREATE INDEX IX_pobiera ON Wynagrodzenia (Id_pracownika)  
/
```

— *Add keys for table Wynagrodzenia*

```
ALTER TABLE Wynagrodzenia ADD CONSTRAINT Wynagrodzenia_PK
        PRIMARY KEY (Id_wynagrodzenia)
```

/

— *Table Proby*

```
CREATE TABLE Proby(
    Id_proby Integer NOT NULL,
    Data_proby Date NOT NULL,
    Godzina_proby Char(5) Check(Godzina_proby LIKE '__: __')NOT NULL,
    Czas_trwania_proby Integer NOT NULL,
    Rodzaj_proby Varchar2(30 )
        CHECK(Rodzaj_proby IN( 'indywidualna', 'sekcyjna', 'calosciowa', 'generalna',
    Id_sali Integer NOT NULL
)
```

/

— *Create indexes for table Proby*

```
CREATE INDEX IX_jest_miejscem ON Proby (Id_sali)
```

/

— *Add keys for table Proby*

```
ALTER TABLE Proby ADD CONSTRAINT Proby_PK PRIMARY KEY (Id_proby)
```

/

— *Table Sale\_koncertowe*

```
CREATE TABLE Sale_koncertowe(
    Id_sali Integer NOT NULL,
    Liczba_miejsc Integer NOT NULL,
    Wielkosc_sceny Char(1 )
        CHECK(Wielkosc_sceny IN ( 'M', 'S', 'D')) NOT NULL,
    Loza_VIP Varchar2(1 ) CHECK(Loza_VIP IN( 'T', 'N')) NOT NULL
)
```

/

— *Add keys for table Sale\_koncertowe*

```
ALTER TABLE Sale_koncertowe ADD CONSTRAINT Sale_Koncertowe_PK
        PRIMARY KEY (Id_sali)
```

/

— *Table Sale\_do\_cwiczen*

```
CREATE TABLE Sale_do_cwiczen(
    Id_sali Integer NOT NULL,
```

```
Wielkosc_sali Char(1 ) CHECK(Wielkosc_sali IN ( 'M', 'S', 'D')) NOT NULL,
Obecnosc_fortepianu Varchar2(1 ) CHECK(Obecnosc_fortepianu IN( 'T', 'N'))
)
/

— Add keys for table Sale_do_cwiczen

ALTER TABLE Sale_do_cwiczen ADD CONSTRAINT Sale_do_cwiczen_PK
PRIMARY KEY (Id_sali)
/

— Table Bilety

CREATE TABLE Bilety(
Id_biletu Integer NOT NULL,
Cena_biletu Float(24) NOT NULL,
Numer_miejsca Integer NOT NULL,
Numer_rzedu Integer NOT NULL,
Zarezerwowany Varchar2(1 ) CHECK(Zarezerwowany IN( 'T', 'N')) NOT NULL,
Oplacony Varchar2(1 ) CHECK(Oplacony IN( 'T', 'N')) NOT NULL,
Id_koncertu Integer NOT NULL
)
/

— Create indexes for table Bilety

CREATE INDEX IX_na ON Bilety (Id_koncertu)
/

— Add keys for table Bilety

ALTER TABLE Bilety ADD CONSTRAINT Bilety_PK PRIMARY KEY (Id_biletu)
/

— Table Uzytkowanie

CREATE TABLE Uzytkowanie(
Id_instrumentu Integer NOT NULL,
Id_pracownika Integer NOT NULL
)
/

— Table Wystepowanie

CREATE TABLE Wystepowanie(
Id_pracownika Integer NOT NULL,
Id_koncertu Integer NOT NULL
)
/

— Table Cwiczenie
```

```
CREATE TABLE Cwiczenie(  
  Id_pracownika Integer NOT NULL,  
  Id_proby Integer NOT NULL  
)  
/  
  
— Table Dyrektorzy  
  
CREATE TABLE Dyrektorzy(  
  Id_dyrektora Integer NOT NULL,  
  Imie Varchar2(30 ) NOT NULL,  
  Nazwisko Varchar2(50 ) NOT NULL  
)  
/  
  
— Add keys for table Dyrektorzy  
  
ALTER TABLE Dyrektorzy ADD CONSTRAINT Dyrektorzy_PK  
  PRIMARY KEY (Id_dyrektora)  
/  
  
— Table Zarzadzanie  
  
CREATE TABLE Zarzadzanie(  
  Id_dyrektora Integer NOT NULL,  
  Id_filharmonii Integer NOT NULL,  
  Data_od Date NOT NULL,  
  Data_do Date  
)  
/  
  
— Add keys for table Zarzadzanie  
  
ALTER TABLE Zarzadzanie ADD CONSTRAINT Zarzadzanie_PK  
  PRIMARY KEY (Id_dyrektora ,Id_filharmonii ,Data_od)  
/  
  
— Table Poczty  
  
CREATE TABLE Poczty(  
  Id_poczty Integer NOT NULL,  
  Kod Char(6 ) NOT NULL,  
  Miejscowosc Varchar2(30 ) NOT NULL  
)  
/  
  
— Add keys for table Poczty  
  
ALTER TABLE Poczty ADD CONSTRAINT Poczty_PK PRIMARY KEY (Id_poczty)
```

```
/

— Table Rodziny_instrumentow

CREATE TABLE Rodziny_instrumentow(
  Id_rodziny Integer NOT NULL,
  Nazwa_rodziny Char(20 ) NOT NULL
)
/

— Add keys for table Rodziny_instrumentow

ALTER TABLE Rodziny_instrumentow ADD CONSTRAINT Rodziny_Instrumentow_PK
          PRIMARY KEY (Id_rodziny)
/

— Table Utwory

CREATE TABLE Utwory(
  Id_utworu Integer NOT NULL,
  Nazwa_utworu Varchar2(100 ) NOT NULL,
  Imie_kompozytora Varchar2(30 ) NOT NULL,
  Nazwisko_kompozytora Varchar2(50 ) NOT NULL,
  Czas_trwania_utworu Char(8)
          CHECK(Czas_trwania_utworu LIKE '__:__:__') NOT NULL
)
/

— Add keys for table Utwory

ALTER TABLE Utwory ADD CONSTRAINT Utwory_PK PRIMARY KEY (Id_utworu)
/

— Table Wykonywanie

CREATE TABLE Wykonywanie(
  Id_koncertu Integer NOT NULL,
  Id_utworu Integer NOT NULL
)
/

— Add keys for table Wykonywanie

ALTER TABLE Wykonywanie ADD CONSTRAINT Wykonywanie_PK
          PRIMARY KEY (Id_koncertu ,Id_utworu)
/

— Create foreign keys (relationships) section —————
```



```
ALTER TABLE Instrumenty ADD CONSTRAINT jest_wlascicielem  
FOREIGN KEY (Id_filharmonii) REFERENCES Filharmonie (Id_filharmonii)  
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT zatrudnia  
FOREIGN KEY (Id_filharmonii) REFERENCES Filharmonie (Id_filharmonii)  
/
```

```
ALTER TABLE Koncerty ADD CONSTRAINT organizuje  
FOREIGN KEY (Id_filharmonii) REFERENCES Filharmonie (Id_filharmonii)  
/
```

```
ALTER TABLE Koncerty ADD CONSTRAINT odbywa się w  
FOREIGN KEY (Id_sali) REFERENCES Sale (Id_sali)  
/
```

```
ALTER TABLE Wynagrodzenia ADD CONSTRAINT pobiera  
FOREIGN KEY (Id_pracownika) REFERENCES Pracownicy (Id_pracownika)  
/
```

```
ALTER TABLE Proby ADD CONSTRAINT jest_miejscem  
FOREIGN KEY (Id_sali) REFERENCES Sale (Id_sali)  
/
```

```
ALTER TABLE Sale ADD CONSTRAINT dysponuje  
FOREIGN KEY (Id_filharmonii) REFERENCES Filharmonie (Id_filharmonii)  
/
```

```
ALTER TABLE Bilety ADD CONSTRAINT na  
FOREIGN KEY (Id_koncertu) REFERENCES Koncerty (Id_koncertu)  
/
```

```
ALTER TABLE Zarzadzanie ADD CONSTRAINT przez  
FOREIGN KEY (Id_dyrektora) REFERENCES Dyrektorzy (Id_dyrektora)  
/
```

```
ALTER TABLE Zarzadzanie ADD CONSTRAINT pod  
FOREIGN KEY (Id_filharmonii) REFERENCES Filharmonie (Id_filharmonii)  
/
```

```
ALTER TABLE Filharmonie ADD CONSTRAINT nalezy
FOREIGN KEY (Id_poczty) REFERENCES Poczty (Id_poczty)
/

ALTER TABLE Instrumenty ADD CONSTRAINT zawiera_sie
FOREIGN KEY (Id_rodziny) REFERENCES Rodziny_instrumentow (Id_rodziny)
/

ALTER TABLE Wykonywanie ADD CONSTRAINT podczas
FOREIGN KEY (Id_koncertu) REFERENCES Koncerty (Id_koncertu)
/

ALTER TABLE Wykonywanie ADD CONSTRAINT jest_czescia
FOREIGN KEY (Id_utworu) REFERENCES Utwory (Id_utworu)
/
```

### 5.3. Sekwencje

W celu zapewnienia unikatowości kluczy głównych i ułatwienia ich wprowadzania, zdecydowano się na stworzenie odpowiednich sekwencji.

```
CREATE SEQUENCE Filharmonie_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Poczty_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Instrumenty_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Rodziny_instrumentow_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Dyrektorzy_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Bilety_SEQ
```

```
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Koncerty_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Sale_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Utwory_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Proby_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Pracownicy_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;

CREATE SEQUENCE Wynagrodzenia_SEQ
MINVALUE 0
START WITH 0
INCREMENT BY 1 ;
```

#### 5.4. Strojenie bazy danych - dobór indeksów

W celu zmniejszenia czasu wykonywania często występujących zapytań wprowadzono indeksy. Przygotowano następujące indeksowane listy:

1. Lista koncertów wraz z datą i godziną rozpoczęcia.

```
CREATE INDEX ix_koncert_info ON Koncerty
(Id_koncertu , Data_koncertu , Godzina_koncertu);
```

2. Lista biletów na dany koncert.

```
CREATE INDEX ix_bilet_na_koncert On Bilety
(Id_biletu , Id_koncertu);
```

3. Lista utworów wraz z informacją o kompozytorze.

```
CREATE INDEX ix_utwor_info on Utwory
(Id_utworu , Nazwa_utworu , Imie_kompozytora , Nazwisko_kompozytora);
```

### 5.5. Wypełnienie bazy danych danymi

Poniżej przedstawiono skrypt SQL, za pomocą którego wypełniono bazę danych przykładowymi danymi.

```
INSERT INTO Poczty VALUES
(POCZTY_SEQ.NEXTVAL, '00-950', 'Warszawa');

INSERT INTO Filharmonie VALUES
(Filharmonie_SEQ.NEXTVAL, 'Narodowa', 'Jasna', 5, 'Warszawa',
'225517111', 'filharmonia@filharmonia.pl', '1900', 1);

INSERT INTO Dyrektorzy VALUES
(Dyrektorzy_SEQ.NEXTVAL, 'Jacek', 'Kaspszyk');

INSERT INTO Dyrektorzy VALUES
(Dyrektorzy_SEQ.NEXTVAL, 'Antoni', 'Wit');

INSERT INTO Zarzadzanie VALUES
(1, 1, TO_DATE('01.09.2013', 'DD.MM.YYYY'), NULL);

INSERT INTO Zarzadzanie VALUES
(2, 1, TO_DATE('01.09.2002', 'DD.MM.YYYY'),
TO_DATE('31.08.2013', 'DD.MM.YYYY'));

INSERT INTO Pracownicy VALUES
(Pracownicy_SEQ.NEXTVAL, 'Janusz', 'Kowalczyk', 'M',
TO_DATE('14.07.1980', 'DD.MM.YYYY'), TO_DATE('05.06.2010', 'DD.MM.YYYY'),
'80071413579', '517607888', 'JK_puzon24@gmail.com', 1);

INSERT INTO Pracownicy VALUES
(Pracownicy_SEQ.NEXTVAL, 'Grażyna', 'Nowak', 'K',
TO_DATE('28.04.1986', 'DD.MM.YYYY'), TO_DATE('29.04.2011', 'DD.MM.YYYY'),
'86042824680', '607517778', 'Nowak_Graża@gmail.com', 1);

INSERT INTO Pracownicy VALUES
(Pracownicy_SEQ.NEXTVAL, 'Mateusz', 'Koroś', 'M',
TO_DATE('17.03.1955', 'DD.MM.YYYY'), TO_DATE('06.07.1980', 'DD.MM.YYYY'),
'55061757715', '755123464', 'super_mati555@gmail.com', 1);

INSERT INTO Pracownicy VALUES
(Pracownicy_SEQ.NEXTVAL, 'Antonina', 'Kucyk', 'K',
TO_DATE('14.02.1991', 'DD.MM.YYYY'), TO_DATE('04.11.2016', 'DD.MM.YYYY'),
'91021401802', '514141414', 'Tosia_Kucyk@wp.pl', 1);

INSERT INTO Rodziny_instrumentow VALUES
(Rodziny_instrumentow_SEQ.Nextval, 'dęte_drewnianie');

INSERT INTO Rodziny_instrumentow VALUES
(Rodziny_instrumentow_SEQ.Nextval, 'dęte_blaszane');

INSERT INTO Rodziny_instrumentow VALUES
```

```
(Rodziny_instrumentow_SEQ.Nextval, 'klawiszowe');
```

```
INSERT INTO Rodziny_instrumentow VALUES
```

```
(Rodziny_instrumentow_SEQ.Nextval, 'perkusyjne');
```

```
INSERT INTO Rodziny_instrumentow VALUES
```

```
(Rodziny_instrumentow_SEQ.Nextval, 'strunowe_smyczkowe');
```

```
INSERT INTO Rodziny_instrumentow VALUES
```

```
(Rodziny_instrumentow_SEQ.Nextval, 'strunowe_szarpane');
```

```
INSERT INTO Rodziny_instrumentow VALUES
```

```
(Rodziny_instrumentow_SEQ.Nextval, 'strunowe_uderzane');
```

```
INSERT INTO Rodziny_instrumentow VALUES
```

```
(Rodziny_instrumentow_SEQ.Nextval, 'inne');
```

```
INSERT INTO INSTRUMENTY Values
```

```
(Instrumenty_SEQ.Nextval, 'obój', '1985', 120200.00,  
'Marigaux', '2500x', 1, 1);
```

```
INSERT INTO INSTRUMENTY Values
```

```
(Instrumenty_SEQ.Nextval, 'puzon', '2005', 36000.00,  
'Yamaha', 'YSL-354E', 1, 2);
```

```
INSERT INTO INSTRUMENTY Values
```

```
(Instrumenty_SEQ.Nextval, 'skrzypce', '1701', 1000000.00,  
'Stradivarius', null, 1, 5);
```

```
INSERT INTO INSTRUMENTY Values
```

```
(Instrumenty_SEQ.Nextval, 'fortepian', '2005', 158000.00,  
'Bechstein', 'B.190', 1, 7);
```

```
INSERT INTO Uzytkowanie Values
```

```
(1,2);
```

```
INSERT INTO Uzytkowanie Values
```

```
(2,3);
```

```
INSERT INTO Uzytkowanie Values
```

```
(4,1);
```

```
INSERT INTO Utwory Values
```

```
(Utwory_SEQ.NEXTVAL, 'Koncert_nr_4_f-moll_Zima, _RV_297',  
'Antonio', 'Vivaldi', '00:08:40');
```

```
INSERT INTO Utwory Values
```

```
(Utwory_SEQ.NEXTVAL, 'Koncert_obojowy_C-dur, _KV_285D',  
'Wolfgang_Amadeusz', 'Mozart', '00:19:33');
```

**INSERT INTO** Utwory **Values**

(Utwory\_SEQ.NEXTVAL, 'Et\_exspecto\_resurrectionem\_mortuorum\_I/47',  
'Olivier', 'Messiaen', '00:33:19');

**INSERT INTO** Utwory **Values**

(Utwory\_SEQ.NEXTVAL, 'Nokturn\_e-moll\_Op.72\_nr1',  
'Fryderyk', 'Chopin', '00:04:44');

**INSERT INTO** Utwory **Values**

(Utwory\_SEQ.NEXTVAL, '4:33', 'John', 'Cage', '00:04:33');

**INSERT INTO** Sale **Values**

(Sale\_SEQ.NEXTVAL, '1452', 1, 1);

**INSERT INTO** Sale **Values**

(Sale\_SEQ.NEXTVAL, '2874', 2, 1);

**INSERT INTO** Sale **Values**

(Sale\_SEQ.NEXTVAL, '1451', 1, 1);

**INSERT INTO** Sale\_koncertowe **Values**

(2, 450, 'D', 'T');

**INSERT INTO** Sale\_do\_cwiczen **Values**

(1, 'M', 'T');

**INSERT INTO** Koncerty **Values**

(Koncerty\_SEQ.NEXTVAL, TO\_DATE('14.02.2018', 'DD.MM.YYYY'),  
'18:00', 120, 1, 2);

**INSERT INTO** Koncerty **Values**

(Koncerty\_SEQ.NEXTVAL, TO\_DATE('31.12.2017', 'DD.MM.YYYY'),  
'20:00', 95, 1, 2);

**INSERT INTO** Koncerty **Values**

(Koncerty\_SEQ.NEXTVAL, TO\_DATE('05.01.2018', 'DD.MM.YYYY'),  
'13:00', 60, 1, 2);

**INSERT INTO** Wykonywanie **Values**

(1,4);

**INSERT INTO** Wykonywanie **Values**

(2,1);

**INSERT INTO** Wykonywanie **Values**

(3,2);

```
INSERT INTO Wykonywanie Values  
(3,3);
```

```
INSERT INTO Wykonywanie Values  
(2,5);
```

```
INSERT INTO Bilety Values  
(Bilety_SEQ.NEXTVAL, 120, 45, 1, 'T', 'T', 1);
```

```
INSERT INTO Bilety Values  
(Bilety_SEQ.NEXTVAL, 120, 46, 1, 'T', 'N', 1);
```

```
INSERT INTO Bilety Values  
(Bilety_SEQ.NEXTVAL, 90, 1, 0, 'N', 'N', 2);
```

```
INSERT INTO Bilety Values  
(Bilety_SEQ.NEXTVAL, 100, 56, 2, 'T', 'T', 3);
```

```
INSERT INTO Bilety Values  
(Bilety_SEQ.NEXTVAL, 105, 78, 3, 'T', 'T', 1);
```

```
INSERT INTO Wynagrodzenia Values  
(WYNAGRODZENIA_SEQ.NEXTVAL, 5000,  
TO_DATE('10.10.2017', 'DD.MM.YYYY'), null, 1);
```

```
INSERT INTO Wynagrodzenia Values  
(WYNAGRODZENIA_SEQ.NEXTVAL, 10000,  
TO_DATE('10.10.2017', 'DD.MM.YYYY'), 1000, 4);
```

```
INSERT INTO Wynagrodzenia Values  
(WYNAGRODZENIA_SEQ.NEXTVAL, 2000,  
TO_DATE('10.10.2017', 'DD.MM.YYYY'), null, 3);
```

```
INSERT INTO Proby Values  
(PROBY_SEQ.NEXTVAL, TO_DATE('12.12.2017', 'DD.MM.YYYY'),  
'14:00', 120, 'sekcyjna', 1);
```

```
INSERT INTO Proby Values  
(PROBY_SEQ.NEXTVAL, TO_DATE('05.01.2018', 'DD.MM.YYYY'),  
'09:00', 180, 'generalna', 2);
```

```
INSERT INTO cwiczenie Values  
(1,1);
```

```
INSERT INTO cwiczenie Values  
(1,2);
```

```
INSERT INTO cwiczenie Values  
(3,2);
```

```
INSERT INTO cwiczenie Values
```

```

(2,2);

INSERT INTO cwiczenie Values
(4,2);

INSERT INTO wystepowanie Values
(1,2);

INSERT INTO wystepowanie Values
(2,2);

INSERT INTO wystepowanie Values
(3,2);

INSERT INTO wystepowanie Values
(4,2);

INSERT INTO Muzycy Values
(4, '4', 'Uniwersytet ŁMuzyczny Łim. ŁFryderyka ŁChopina ŁwŁWarszawie ',
'obój', null, 'T', 'T', 'T', 'N');

INSERT INTO Muzycy Values
(1, '4', 'Uniwersytet ŁMuzyczny Łim. ŁFryderyka ŁChopina ŁwŁWarszawie ',
'puzon', null, 'N', 'T', 'T', 'N');

```

Poniżej zamieszczone zostały niektóre z wypełnionych relacji.

ID_FILHARMONII	NAZWA	ULICA	NUMER_LOKALU	MIASTO	NUMER_TELEFONU	EMAIL	ROK_ZALOZENIA	ID_POCZTY
1	1 Narodowa Jasna ...	5		Warszawa	225517111	filharmonia@filharmonia.pl	1900	1

Rys. 5.1: Relacja Filharmonie

ID_PRACOWNIKA	IMIE	NAZWISKO	PLEC	DATA_URODZENIA	DATA_ZATRUDNIENIA	PESEL	NUMER_TELEFONU	EMAIL	ID_FILHARMONII
1	1 Janusz	Kowalczyk	M	80/07/14	10/06/05	80071413579	517607888	JK_puzon24@gmail.com	1
2	2 Grażyna	Nowak	K	86/04/28	11/04/29	86042824680	607517778	Nowak_Graza@gmail.com	1
3	3 Mateusz	Koroś	M	55/03/17	80/07/06	55061757715	755123464	super_mati555@gmail.com	1
4	4 Antonina	Kucyk	K	91/02/14	16/11/04	91021401802	514141414	Tosia_Kucyk@wp.pl	1

Rys. 5.2: Relacja Pracownicy

ID_KONCERTU	DATA_KONCERTU	GODZINA_KONCERTU	CZAS_TRWANIA_KONCERTU	ID_FILHARMONII	ID_SALI
1	1 18/02/14	18:00		120	1 2
2	2 17/12/31	20:00		95	1 2
3	3 18/01/05	13:00		60	1 2

Rys. 5.3: Relacja Koncerty



ID_SALI	NUMER_SALI	NUMER_PIETRA	ID_FILHARMONII
1	1 1452	1	1
2	2 2874	2	1
3	3 1451	1	1

Rys. 5.4: Relacja Sale

ID_UTWORU	NAZWA_UTWORU	IMIE_KOMPOZYTORA	NAZWISKO_KOMPOZYTORA	CZAS_TRWANIA_UTWORU
1	1 Koncert nr 4 f-moll Zima, RV 297	Antonio	Vivaldi	00:08:40
2	2 Koncert obojowy C-dur, KV 285D	Wolfgang Amadeusz	Mozart	00:19:33
3	3 Et exspecto resurrectionem mortuorum I/47	Olivier	Messiaen	00:33:19
4	4 Nokturn e-moll Op.72 nr1	Fryderyk	Chopin	00:04:44
5	5 4:33	John	Cage	00:04:33

Rys. 5.5: Relacja Utwory

ID_PRACOWNIKA	STOPIEN...	NAZWA_UCZELNI	SPECJALNOSC_GLOWNA	SPECJALN...	SOLISTA	KAMER...	MUZ...	CHORYSTA
1	4 4	Uniwersytet Muzyczny im. Fryderyka Chopina w Warszawie	obój	(null)	T	T	T	N
2	1 4	Uniwersytet Muzyczny im. Fryderyka Chopina w Warszawie	puzon	(null)	N	T	T	N

Rys. 5.6: Relacja Muzycy

ID_BILETU	CENA_BILETU	NUMER_MIEJSCA	NUMER_RZEDU	ZAREZERWOWANY	OPLACONY	ID_KONCERTU
1	1	120	45	1 T	T	1
2	2	120	46	1 T	N	1
3	3	90	1	0 N	N	2
4	4	100	56	2 T	T	3
5	5	105	78	3 T	T	1

Rys. 5.7: Relacja Bilety

ID_INSTRUMENTU	NAZWA_INSTRUMENTU	ROK_PRODUKCJI	CENA_INSTRUMENTU	MARKA	MODEL	ID_FILHARMONII	ID_RODZINY
1	1 obój	1985	120200	Marigaux	2500x	1	1
2	2 puzon	2005	36000	Yamaha	YSL-354E	1	2
3	3 skrzypce	1701	1000000	Stradivarius	(null)	1	5
4	4 fortepian	2005	158000	Bechstein	B.190	1	7

Rys. 5.8: Relacja Instrumenty

ID_PROBY	DATA_PROBY	GODZINA_PROBY	CZAS_TRWANIA_PROBY	RODZAJ_PROBY	ID_SALI
1	1 17/12/12	14:00	120	sekcyjna	1
2	2 18/01/05	09:00	180	generalna	2

Rys. 5.9: Relacja Próby

## 5.6. Przykłady zapytań poleceń SQL odnoszących się do bazy

1. Zapytanie listujące podstawowe informacje o koncercie tj. godzinę rozpoczęcia koncertu, czas jego trwania oraz numer sali, w której się odbywa na podstawie daty koncertu.

```

select k.Data_koncertu , k.Godzina_koncertu , k.Czas_trwania_koncertu ,
s.Numer_sali
from koncerty k natural join sale s
where k.Data_koncertu='17/12/31';

```

	DATA_KONCERTU	GODZINA_KONCERTU	CZAS_TRWANIA_KONCERTU	NUMER_SALI
1	17/12/31	20:00		95 2874

Rys. 5.10: Wynik zapytania

2. Zapytanie listujące podstawowe informacje o próbie tj. datę próby, rodzaj próby, wielkość sali, w której się odbywa na podstawie numeru sali.

```

select p.Data_proby , p.Rodzaj_proby , s_c.wielkosc_sali
from sale s natural join proby p natural join sale_do_cwiczen s_c
where s.numer_sali = '1452';

```

	DATA_PROBY	RODZAJ_PROBY	WIELKOSC_SALI
1	17/12/12	sekcyjna	M

Rys. 5.11: Wynik zapytania

3. Zapytanie listujące id oraz nazwę rodziny instrumentów, do której nie należą żadne z posiadanych przez filharmonię instrumentów.

```

select r.id_rodziny , r.nazwa_rodziny
from Rodziny_instrumentow r
where r.ID_RODZINY <> ALL (select i.id_rodziny from Instrumenty i);

```

	ID_RODZINY	NAZWA_RODZINY
1	3	klawiszowe
2	4	perkusyjne
3	6	strunowe szarpane
4	8	inne

Rys. 5.12: Wynik zapytania

4. Zapytanie listujące podstawowe informację o muzykach biorących udział w koncertach odbywających się danego dnia tj. imię, nazwisko oraz specjalność główną na podstawie daty.

```

select p.Imie , p.Nazwisko , m.specjalnosc_glowna
from Koncerty k natural Join wystepowanie Natural join
pracownicy p natural join muzycy m
where k.Data_koncertu = '17/12/31';

```

	IMIE	NAZWISKO	SPECJALNOSC_GLOWNA
1	Janusz	Kowalczyk	puzon
2	Antonina	Kucyk	obój

Rys. 5.13: Wynik zapytania

5. Zapytanie wypisujące liczbę opłaconych biletów na podstawie id koncertu.

```
select count(*) as liczba_oplaconych
from Koncerty Join Bilety using(id_koncertu)
where id_koncertu=1 and oplacony='T';
```

	LICZBA_OPLACONYCH
1	2

Rys. 5.14: Wynik zapytania

6. Zapytanie listujące nazwę instrumentu wraz z imieniem i nazwiskiem pracownika, który go używa/używał.

```
select nazwa_instrumentu, imie, nazwisko from instrumenty
natural join uzytkowanie natural join pracownicy
order by nazwa_instrumentu desc;
```

	NAZWA_INSTRUMENTU	IMIE	NAZWISKO
1	puzon	Mateusz	Koroś
2	obój	Grażyna	Nowak
3	fortepian	Janusz	Kowalczyk

Rys. 5.15: Wynik zapytania